

ESCUELA DE INGENIERÍA MINERA E INDUSTRIAL DE
ALMADÉN



TRABAJO FIN DE GRADO
DISEÑO DE BARRERAS
LUMINOSAS E INTELIGENTES
PARA MEJORAR LA SEGURIDAD
VIAL EN LOS PASOS DE PEATONES

UNIVERSIDAD DE CASTILLA-LA MANCHA
GRADO EN INGENIERÍA ELÉCTRICA

Autor: Luis Biko Nzang
Director: Dr. Javier A. Albusac Jiménez
Curso: 2015-2016

AGRADECIMIENTOS Y DEDICATORIAS

En primer lugar me gustaría agradecer a mi tutor asignado para la realización de este proyecto *Doctor y Profesor D. Javier Alonso Albusac*, por ofrecerme esa posibilidad de realizar dicho trabajo y por depositarme tanta confianza con la esperanza de que salga perfecto conforme a nuestros objetivos marcados.

Agradecer a mi familia y seres queridos, en especial amigos y compañeros cercanos porque ellos siempre estuvieron ahí incluso cuando todavía no tenía claro lo que iba hacer para cerrar este ciclo de mi vida. A lo largo de mi carrera recibí muchos golpes como todos pero dichos golpes los podía ir superando gracias a esa gente que estaban ahí que a pesar de no ser familia me dieron la posibilidad de crear otra familia y sentirme como si los conociera de toda la vida.

Dedicar tiempo para realizar este proyecto para mí ha sido importante porque ha marcado una etapa en mi vida y me ha permitido tener más visión acerca de otros posibles proyectos que ya tengo pensado realizar con otros fines benéficos. Al principio de todo no me sentía capaz de realizar un proyecto de estas dimensiones y mucho menos realizarlo prácticamente solo pero he de confesar que el hecho de decidir continuar con dicho trabajo era porque detrás de todo reto siempre se esconde un estado de satisfacción, seguridad y confianza hacia uno mismo y además se adquiere experiencia y te acerca un poco más hacia el nuevo mundo en el que estas a punto de entrar, que es el mundo laboral y todos estos conocimientos desarrollados te permiten ser mas competente.

Haciendo una dedicatoria muy especial para mi familia destacarían figuras muy importantes como la de mi madre *Catalina Micha Biko Ayaba*, mi tio padre *Jesús Asu Micha* y mi tia *Marta Avaro Biko*, cuya responsabilidad de cubrirme los estudios recaía sobre ellos. Sería imposible olvidar la participación de todas mis hermanas (*Engracia, Francisca, Estela, Priscila, Balbina, Marta*) que nunca dejaron de confiar en mí y me apoyaban hasta tal extremo en que yo a veces ya dudaba de mi potencial pero que a la vez me decía “ellas han vivido más que yo y por lo tanto si ven algo en mi ellas sabrán” por lo tanto mi única obligación era hacer todo cuanto podía porque lo consideraba como mi única moneda de cambio.

Vivir alejado del hogar nos permite tener otra percepción de la vida y a la vez que nos hace más fuertes y nos forma, no solo educativamente sino moralmente y nos deja ser más responsables con nuestra obligaciones (opinión personal).

No podría depositarle todo el peso de este sufrimiento en este proyecto fin de grado a pesar de ser muy importante olvidando los casi cinco años que ha durado la carrera, por lo tanto, estoy sumamente agradecido por toda esa gente que me ayudaron a llevar este buque de trabajo, dedicación, esfuerzo y sacrificio desde el comienzo hasta el final ya sea de manera presencia o no.

Gracias.

RESUMEN

En este presente trabajo fin de grado se ha diseñado e implementado un sistema de control de tráfico basado en un controlador Arduino para la mejorar y refuerzo de la seguridad vial en los cruces para peatones.

El principal objetivo marcado en la creación de este sistema es agregarle un complemento a las soluciones adoptadas para la reducción de los atropellos que se producen en los pasos de peatones, es decir, aquellos en donde se registran mayor índice de atropellos ya sea debido a la escasa visibilidad o a la poca iluminación que se pueda dar en el ambiente. Para ello lo primero ha sido diseñar un entorno de pruebas con toda la tecnología pensada para el sistema.

Para el control total del sistema se ha utilizado un microcontrolador Arduino ATmega2560 a través de cual se le fue conectando diversos componentes eléctricos ya sean sensores y actuadores tales como: sensores infrarrojos, sensores LDR o fotorresistencias, sensores PIR o de movimiento, placa LCD, diodos LED, etc. Entre las funciones que debe desempeñar en el sistema, las dos más importantes son:

- Comprobación de la ausencia o presencia de peatones en medio del cruce. Este control se consigue directamente con el controlador gracias a los sensores piroeléctricos o sensores de movimientos instalados sobre el paso y cuya función no es otra que informar de la situación que se está produciendo en todo momento independientemente del estado en que se encuentre el semáforo que regula el tráfico en esta zona.
- Estimar la velocidad a la que viaja un coche una vez esté dentro del área de control formada por una serie de barreras infrarrojas y dividida en tramos, de tal manera que una vez que el automóvil haya cortado dos barreras consecutivas (*lo que constituye un tramo*), se pueda calcular mediante un regulador PID su velocidad y error producido debido a la variación de esta velocidad con respecto a la velocidad permitida en todo el recorrido de dicha zona de control.

Todo el comportamiento modelado en el sistema se realizó mediante el lenguaje de programación predeterminado de Arduino (*conocimientos adquiridos en los grados de la EIMIA*) y el resultado final que se pretende conseguir con este nuevo sistema es proporcionarle a los conductores de automóviles un **aviso prolongado de iluminación** gracias a la tecnología de **barreras lumínicas incrustadas en el asfalto** con el fin de que puedan conocer de manera intuitiva su comportamiento o la evolución de su automóvil (*nivel de velocidad*), independientemente de la información que le proporcione el velocímetro.

Dicho aviso les ayudará a obtener un control más seguro en caso de que éste estuviera distraído sobre el volante, regular su velocidad y poder llegar de forma segura a la línea de parada del paso de peatones o evitar alcanzar a ciertos peatones que suelen cruzar las intersecciones de manera legal o ilegal poniendo el peligro sus vidas y las de otra gente.

En función de los datos proporcionados por los diferentes sensores y de los cálculos realizados, el controlador adoptará una salida conjunta activando o desactivando las líneas de iluminación de aviso prolongado de manera progresiva en función a las condiciones

preestablecidas en el sistema, las cuales deben ser comprobadas o comparadas con los datos proporcionados anteriormente.

Se espera y se confía con toda sinceridad que este nuevo componente ayudará a resolver o a reducir este problema que durante décadas ya viene cobrando vidas humanas.

Palabras clave: seguridad, atropellos, peatones, regulador PID, velocidad.

ABSTRACT

In this present work order degree it is designed and implemented a traffic control system based on an Arduino controller to improve and enhance road safety in crosswalks.

The main objective of the implementation of this system is to add a complement to the solutions adopted to reduce the huge number vast majority of abuses that occur in crosswalks, in areas highly frequented pedestrian and higher rates of abuses carried out due to poor visibility that can be given by weather accordingly and thus provide a safe traffic. For that, the first thing that has been designed an environmental test and then implement all the technology designed into the system.

For the full control of the system, an Arduino microcontroller ATmega2560 has been used through which it'd been connected to various electrical components either sensors or actuators such as: infrared sensors, LDR sensors, PIR or motion sensors, LCD board, LED diodes, etc. Among the functions to be played in the system, the most important two are:

- Testing the absence or presence of pedestrians in the middle of the crossroad. This control is achieved directly to the controller thanks to the pyroelectric sensors or motion sensors installed on the step and whose function is simply to report the situation that is occurring at all times regardless of the state of the traffic light that regulates the traffic in that area.
- To estimate the speed at which the car travels once it is within the control area consisted of a series of infrared barriers and divided into sections, so that once the car has cut two consecutive barriers (which is a stretch), that can be calculated using a PID controller and speed error produced by this speed in relation to the speed limit throughout the route of this control zone.

All modelling system behavior was performed using the default programming language Arduino (knowledge acquired in the degrees of the EIMIA) and the final result to be achieved with this new system is to provide car drivers an extended lighting notice by lighting technology barriers embedded in the asphalt so that they could know intuitively their behavior or evolution of your car (velocity level) regardless of the information provided by the speedometer.

Such notice would help them to get a secure control if drivers were distracted at the steering wheel, adjust the speed and reach the traffic light stop line in a safe way or avoid hitting pedestrians who often cross certain intersections legally or illegally putting in danger their lives and those of other people.

Based on data provided by the various sensors and calculations, the controller will adopt a joint output by activating or deactivating the lighting lines notice progressively extended according to the conditions preset in the system, which must be checked or compared with the data provided above.

It is expected and sincerely hoped that in this new component helping to solve or reduce this problem which for decades has been taking human lives.

Key words: security, pedestrians, outrage, PID controller, speed

Índice de contenido

1.	INTRODUCCIÓN.....	14
1.1.	Planteamiento de problema.....	15
1.2.	Formulación del problema.....	16
2.	OBJETIVOS.....	20
2.1.	Implementación de unas barreras infrarrojas.....	21
2.2.	Implementación de unas barreras luminosas y visibles sobre el asfalto.....	21
2.3.	Programación y modelado del comportamiento del sistema.....	22
3.	ESTADO DEL CONOCIMIENTO.....	24
3.1.	Etimología.....	24
3.2.	Historia de los semáforos.....	24
3.2.1.	Origen.....	24
3.2.2.	Pasado.....	25
3.2.3.	Evolución.....	26
3.2.4.	Presente.....	27
3.3.	Constitución y funcionamiento de los semáforos convencionales.....	29
3.3.1.	Componentes de un semáforo convencional.....	29
3.3.2.	Funcionamiento de un semáforo convencional.....	30
3.3.3.	Equipo de control de un semáforo.....	30
3.3.4.	Tipos de semáforos: semáforos para funciones especiales.....	31
3.3.5.	Empresas Fabricantes y Distribuidora.....	34
3.4.	Semáforos Inteligentes y los sistemas aplicados.....	35
3.4.1.	Sistemas.....	35
3.4.2.	Semáforos inteligentes.....	36
3.4.2.1.	Tecnologías Aplicadas.....	37
3.4.3.	Otros Sistemas usados en la actualidad.....	42
3.5.	Inteligencia de un semáforo o señales de tráfico programables.....	44
3.5.1.	Lenguaje ensamblador.....	44
3.5.2.	Microcontrolador.....	46
3.5.3.	Microprocesador.....	47
3.6.	Sistema de Control.....	49
3.6.1.	Clasificación de los sistemas de control.....	50
3.6.2.	Tipos de sistemas de control.....	51
3.6.2.2.	Control proporcional (P).....	51
3.6.3.	Control Integral.....	52
3.6.4.	Control proporcional-integral (PI).....	53

3.6.5.	Control proporcional-Derivativo (PD)	54
3.6.6.	Control proporcional-Integral Derivativo (PID).....	56
3.7.	Sensores	58
3.7.1.	Características de un sensor.....	59
3.7.2.	Tipos de sensores	60
3.8.	Controladores programables-Arduino.....	62
3.8.1.	Lenguajes de programación para Arduino.....	63
3.8.2.	Placas de Arduino	64
3.8.3.	Shield Arduino	71
3.8.4.	Sensores Arduino	77
4.	Diseño de barreras luminosas e inteligentes para mejorar la seguridad en los pasos de peatones.....	79
4.1.	Introducción	79
4.2.	Componentes del sistema.....	80
4.2.1.	Arduino ATMmega2560	81
4.2.2.	Modulo de sensor emisor y receptor de infrarrojos	83
4.2.3.	Módulo detector de movimiento sensor PIR HCSR501.....	85
4.2.4.	Diodos LED's	86
4.2.5.	Sensor LDR (CE-C2795).....	88
4.3.	Montaje del entorno de pruebas	89
4.3.1.	Material de montaje	89
4.3.2.	Instalación de los componentes electrónicos.....	92
4.4.	Esquemas de conexiones de los componentes del sistema	99
4.4.1.	Tipo de cable.....	99
4.4.2.	Conexión de los diodos emisores y receptores infrarrojos	100
4.4.3.	Conexión de los diodos LED para la iluminación de las barreras.....	104
4.4.4.	Conexión de sensor LDR.....	109
4.4.5.	Conexión del sensor Piroeléctrico PIR o de presencia	111
4.4.6.	Conexión de la pantalla LCD con un potenciómetro	114
4.5.	Regulador PID y funcionamiento del sistema.....	116
4.5.1.	Regulador PID para el control de la velocidad.....	117
4.5.2.	Funcionamiento del sistema	126
4.5.3.	Problemas encontrados tras realizar las primeras pruebas en el sistema.....	134
4.5.3.	Ejemplo Práctico del funcionamiento de todo el sistema.....	137
4.6.	Programación con Arduino de sistema.....	139
4.6.1.	Reglas definidas en el sistema	139
4.6.1.	Condigo fuente programado	141

4.6.	Suposición sobre un entorno real	173
4.6.1.	Zonas con mayor índice de atropellos en España	174
4.6.2.	Relés	175
4.6.3.	Cables jumper Arduino	176
4.6.4.	Adaptador de tensión y toma de corriente	176
4.6.5.	Interruptores diferenciales	177
4.6.6.	Caja de conexiones eléctricas	177
4.6.7.	Control de las barreras luminosas.....	178
4.6.8.	Control del sensor de movimiento.....	181
4.6.9.	Control de sensor de luz	182
4.6.10.	Control de las barreras infrarrojas.....	184
5.	RESULTADOS.....	188
5.1	Comprobación de la señales los sensores	188
5.2	Hipótesis y resultados de las simulaciones realizadas	189
5.2.1	Hipótesis 1	189
5.2.2.	Hipótesis 2	190
5.2.3.	Hipótesis 3	192
5.2.4.	Hipótesis 4	193
5.2.5.	Hipótesis 5	195
6.	ESTUDIO ECONÓMICO.....	201
6.1.	Presupuesto del entorno de pruebas	201
6.1.1.	Componentes electrónicos	201
6.1.2.	Herramientas, y material decorativo para entorno.....	203
6.1.3.	Horas de trabajo y planificación del proyecto	204
6.2.	Presupuesto del entorno real	206
6.2.1.	Componentes electrónicos	206
7.	CONCLUSIONES.....	210
8.	Bibliografía.....	214

Tablas

Tabla 1:	Empresas destacadas fabricantes de semáforos	34
Tabla 2:	sensores con magnitudes y características	61
Tabla 3:	sensores con magnitudes y características	61
Tabla 4:	pinos utilizados en el Arduino	83
Tabla 5:	Funciones de los pines de la pantalla LCD.....	99
Tabla 6:	Configuración del sensor PIR	112
Tabla 7:	valores de la ganancia K para diferentes zonas	120

Tabla 8: salidas del regulador con respecto a la variación de la velocidad entre tramos...	124
Tabla 9: Interruptores de protección	177
Tabla 10: hoja de características del módulo infrarrojo.....	185
Tabla 11: señales recibidas por los receptores infrarrojos en <i>Bits</i>	188
Tabla 12: Rango de señales del LDR y señales del PIR	189
Tabla 13: Resultados de la simulación uno.....	189
Tabla 14: Resultados de la simulación uno, hipótesis 2	191
Tabla 15: Resultados de la simulación dos, hipótesis 2.....	192
Tabla 16: Resultados de la simulación uno, hipótesis 3	193
Tabla 17: Resultados de la simulación uno, hipótesis 4	194
Tabla 18: Resultados de la simulación dos, hipótesis 1	195
Tabla 19: Resultados de la simulación uno, hipótesis 5	196
Tabla 20: Resultados de la simulación dos, hipótesis 5.....	196
Tabla 21: Resultados de la simulación tres, hipótesis 5.....	197
Tabla 22: Resultados de la simulación cuatro, hipótesis 5	198
Tabla 23: presupuesto del material empleado en el prototipo.....	204
Tabla 24: horas de trabajo invertido y precio.....	206

Ilustraciones

Ilustración 1: comportamiento del sistema	17
Ilustración 2: Monitor de niveles de tono de un equipo de música Blackmagic	17
Ilustración 3: Monitor de tonos de Sony Walkman	18
Ilustración 4: problemas registrados (A) solución diseñada (B).....	20
Ilustración 5: Modelos de los primeros semáforos	25
Ilustración 6: Luces Led de los semáforos actuales.....	28
Ilustración 7: semáforo para el control de carril	31
Ilustración 8: semáforo peatonal.....	32
Ilustración 9: semáforo para ciclistas.....	33
Ilustración 10: Esquema de un circuito con procesamiento de imágenes.....	40
Ilustración 11: semáforo con contador.....	42
Ilustración 12: semáforo onda verde o sincronizada	42
Ilustración 13: ejemplo para el computador virtual (POCA).....	45
Ilustración 14: ejemplo para el microchip PIC16F84	45
Ilustración 15: microcontrolador RISC.....	47
Ilustración 16: Microcontrolador CISC	47
Ilustración 17: aspecto interno de un microprocesador	48
Ilustración 18: Diagrama de bloques simples	49
Ilustración 19: Bloque de representación de operaciones matemáticas.....	49
Ilustración 20: Representación de un sistema de control.....	50
Ilustración 21: Esquema de control de lazo abierto	50
Ilustración 22: Esquema de control de lazo cerrado	51
Ilustración 23: Respuesta temporal de un controlador proporcional	52
Ilustración 24: Respuesta temporal de un controlador integral.	53
Ilustración 25: Respuesta temporal de un controlador PI.	54
Ilustración 26: Respuesta temporal de un controlador PD.	55
Ilustración 27: Diagrama de bloques de un regulador PID.....	57
Ilustración 28: Respuesta temporal de un controlador PID	58
Ilustración 29: grafico analógico de un sensor	59
Ilustración 30: grafico digital de un sensor.....	59

Ilustración 31: placa de Arduino uno	64
Ilustración 32: Placa Arduino Due	65
Ilustración 33: Arduino ADK.....	66
Ilustración 34:Placa ArduinoMega2560	67
Ilustración 35: Placa Arduino nano.....	67
Ilustración 36: Placa Arduino Leonardo	68
Ilustración 37: Placa Arduino YUN	69
Ilustración 38: Placa Arduino Micro.....	70
Ilustración 39: Placa Arduino Ethernet.....	71
Ilustración 40: Placa Arduino wifi	72
Ilustración 41: Placa shield Arduino GSM	72
Ilustración 42:shield Arduino GSM.....	72
Ilustración 43: Placa shield Arduino motor	73
Ilustración 44: shield controlador Puente H	74
Ilustración 45: Módulos RF transmisor y RF receptor	74
Ilustración 46: Módulo HC-06 delantero y Trasero.....	75
Ilustración 47: Conector USB del Módulo APC220 y antena emisor-receptor.....	76
Ilustración 48: Shield de Arduino con Módulo XBee.....	76
Ilustración 49: Módulos nRF24LU1 y nRF201	77
Ilustración 50: Vistas 3D del entono de pruebas.....	80
Ilustración 51: Arquitectura del sistema	81
Ilustración 52: placa Arduino ATMmega2560 con especificaciones.....	82
Ilustración 53: Diodo emisor infrarrojos	84
Ilustración 54: Diodo receptor de infrarrojos	84
Ilustración 55: sensor PIR HCSR501	86
Ilustración 56: Componentes físicos de un LED	87
Ilustración 57: sensor LDR CE-C2795	88
Ilustración 58: Comportamiento del sensor LDR CE-C2795 en de la iluminaria	88
Ilustración 59: tabla con trazado de medidas.....	89
Ilustración 60: herramienta de corte y pegamento.....	90
Ilustración 61: Recorte y encaje de piezas.....	90
Ilustración 62: Tornillo utilizado como taladro	91
Ilustración 63: pintura y señalización	91
Ilustración 64: material de pintura y señalización	92
Ilustración 65: herramientas utilizadas en la instalación	93
Ilustración 66: Distribución Emisor-receptor en el modelado 3D	93
Ilustración 67: distribución de los diodos emisores y receptores de los infrarrojos	94
Ilustración 68: distribución de las barreras luminosas.....	95
Ilustración 69: Estados del semáforo	95
Ilustración 70: Modo de instalación del sensor PIR, modelado 3D.....	96
Ilustración 71: Modo de instalación del sensor PIR	97
Ilustración 72: disposición del sensor LDR.....	97
Ilustración 73: pantalla LCD.....	98
Ilustración 74: ubicación de la pantalla LCD	99
Ilustración 75: cable trenzado	100
Ilustración 76: esquema eléctrico de una única barrera de infrarrojos	100
Ilustración 77: aspecto físico de conexión emisor-receptor.....	101
Ilustración 78: Aspecto físico y completo del circuito de los módulos emisor-receptor ...	101
Ilustración 79: Esquema eléctrico de los módulos Emisor-receptor de infrarrojos	102
Ilustración 80: conexión de un diodo LED al Arduino	104

Ilustración 81: Esquema eléctrico de las barreras de iluminación.....	105
Ilustración 82: esquema eléctrico de una sola barrera de iluminación	106
Ilustración 83: aspecto físico de una única barrera luminosa	106
Ilustración 84: aspecto físico de Circuito completo de todas las barreras luminosas	107
Ilustración 85: configuración pull-up y pull-down	109
Ilustración 86: Conexión Pull-up del sensor LDR con Arduino ATmega2560	110
Ilustración 87: configuración pull-up en el prototipo	110
Ilustración 88: esquema de conexión de un sensor PIR a placa Arduino	111
Ilustración 89: medidas mecánicas de un sensor PIR	112
Ilustración 90: Esquema eléctrico de los sensores PIR, LDR y estados del semáforo	113
Ilustración 91: conexión de una pantalla LCD	114
Ilustración 92: Esquema eléctrico de conexión de una pantalla de display LCD.....	115
Ilustración 93: pantalla LCD con un potenciómetro regulador	115
Ilustración 94: diagrama de bloques del sistema	117
Ilustración 95: nueva distribución con 12 barreras de iluminación	135
Ilustración 96: Mensaje informativo de estado Nocturno.....	137
Ilustración 97: Información sobre el comportamiento en el sistema	138
Ilustración 98: cruce de la C/ Dr Manuel candela-valencia.....	174
Ilustración 99: Avd. La Paz y calle Vara de Rey- Logroño.....	174
Ilustración 100: Cruce de la c/ Isable II- Santander.....	175
Ilustración 101: Módulo relé de 4 canales	175
Ilustración 102: Cables jumper con conectores macho-hembra	176
Ilustración 103: toma de corriendo 10 A	176
Ilustración 104: Adaptador de tensión a 9v	176
Ilustración 105: Interruptor diferencial de 63A	184
Ilustración 106: Interruptor diferencial 25A	177
Ilustración 107: caja de conexiones	178
Ilustración 108: aspecto modular de un LightGuard	178
Ilustración 109: conexión de un lightguard al Arduino	179
Ilustración 110: configuración con barreras de iluminación utilizadas en el prototipo 1 ..	180
Ilustración 111: configuración con barreras de iluminación retrasadas.....	180
Ilustración 112: distribución lineal de las <i>lightsguard</i>	180
Ilustración 113: Sensor de movimiento de 12v.....	181
Ilustración 114: Esquema de conexión de un sensor de movimiento con relé	182
Ilustración 115: Sensor de luz.....	183
Ilustración 117: Módulos infrarrojos E3JM-10M4-N.....	184
Ilustración 116: Esquema de conexión de un sensor de luz con relé.....	183
Ilustración 118: circuito eléctrico de emisor-receptor conectado a unos relés	186
Ilustración 119: simulación uno	190
Ilustración 120: simulación uno de la hipótesis dos	191
Ilustración 121: simulación dos de la hipótesis dos.....	192
Ilustración 122: simulación de semáforo en verde o ámbar y con peatones cruzando	193
Ilustración 123: simulación uno de la hipótesis cuatro	194
Ilustración 124: simulación dos de la hipótesis cuatro	195
Ilustración 125: simulación uno de la hipótesis cinco	196
Ilustración 126: simulación dos de la hipótesis cinco.....	197
Ilustración 127: simulación tres de la hipótesis cinco	198
Ilustración 128: simulación cuarto de la hipótesis cinco	199

Ecuaciones

Ecuación 1: salida de un control proporcional (P).....	51
Ecuación 2: salida de un control integral (I).....	52
Ecuación 3: salida de un control integral en Laplace	52
Ecuación 4: Función de transferencia de un control Integral en Laplace.....	53
Ecuación 5: función de transferencia de un control PI en Laplace.....	53
Ecuación 6: salida de un control PD.....	54
Ecuación 7: salida de un controlador PD en Laplace	55
Ecuación 8: Función de transferencia de un control PD.....	55
Ecuación 9: suma de las tres acciones de un control PID.....	56
Ecuación 10: salida de un control PID.....	57
Ecuación 11: salida de un control PID en Laplace	57
Ecuación 12: función de transferencia de un control PID	57
Ecuación 13: velocidad con aceleración constante.....	83
Ecuación 14: fórmula cinemática del espacio.....	119
Ecuación 15: fórmula de la velocidad con aceleración constante	119
Ecuación 16: formula de la velocidad normal de un tramo de control.....	119
Ecuación 17: ganancia para el incremento de la velocidad calculada	120
Ecuación 18: Error absoluto	120
Ecuación 19: Error absoluto en tanto porciento.....	120
Ecuación 20: límite del error permitido k_p	121
Ecuación 21: límite del error con evolución lenta k_p, k_i	121
Ecuación 22: límite del error con evolución rápida, k_d	122
Ecuación 23: velocidad ficticia.....	123
Ecuación 24: Distancia de corte.....	124

CAPÍTULO 1: *INTRODUCCIÓN*

1. INTRODUCCIÓN

Visto el aumento considerable en la producción y venta de los medios de transporte terrestres, la seguridad vial tanto para los peatones como los propios conductores va siendo cada vez más ineficiente y menos segura.

Este gran problema surge debido a un aumento considerable en el tráfico. La acumulación y/o congestión de los vehículos en casi todas las ciudades del mundo es muy pronunciado, de tal manera que siguiendo con el método de control con los semáforos convencionales, es casi imposible proporcionarles una mayor seguridad a toda esta gente que teme por su vida y que velan por tener una comodidad a la hora de utilizar un medio de transporte personal, público o aquellos que optan desplazarse a pie para llegar a sus destinos.

Observando claramente como este problema crece, muchos expertos e investigadores de la seguridad vial o incluso ingenieros entendidos en la materia están buscando otras soluciones con el fin de acabar con esto. Unas de las posibles soluciones por las que se ha optado ha sido el perfeccionamiento de las señales de tráfico. Para este trabajo fin de grado se ha optado por hacer uso de la arquitectura de los semáforos convencionales y acoplarles nuevos componentes tales como, la *instalación de barreras luminosas visibles y un sistema de sensorización* que producirían un aviso a los conductores sobre el estado de su velocidad, aplicando diversas áreas de la ingeniería y tecnologías implementadas para hacer posible su existencia y funcionalidad.

Es necesario mencionar que los semáforos convencionales son dispositivos que se han venido utilizando desde hace tiempo y que cuya función esencial es la de regular el tráfico evitando la congestión de vehículos y al mismo tiempo permitiendo el paso de los peatones en las vías donde se encuentren instalados. Pero también se diría que no solo es importante obtener un flujo flexible en las vías e intersecciones, sino tener más en consideración las vidas de los peatones dado que en algunas situaciones corren peligro a la hora de cruzar dichas intersecciones y que la mejor opción sería encontrar un funcionamiento coordinado de estas barreras tanto para los peatones como para los conductores.

Basándose en el último estudio realizado por la **fundación Mutua Madrileña** sobre la *siniestralidad vial y atropellos* [29] en el que se asegura que de todos los atropellos que se producen en el mundo entero casi el **70%** de ellos se dan en los pasos de peatones y que dentro de este porcentaje, casi el **45%** de ellos son nocturnos y eso es debido a ciertos factores como los cambios meteorológicos, la poca visibilidad o falta de iluminación, el deslumbramiento producido por las luces adyacentes perturbando la vista. Ante tal situación y teniendo en cuenta que los semáforos convencionales están sometidos a una incapacidad de tomar sus propias decisiones o conseguir un control óptimo atendiendo a la situación del tráfico en un determinado momento, es necesario desarrollar un mecanismo que de algún modo podría contribuir en la reducción de estos incidentes.

Siguiendo esa idea se supone que con la incorporación de las *barreras luminosas* en toda la zona de control será un complemento esencial para reforzar la seguridad. En tanto que se podría resolver diversos problemas que hasta la fecha se siguen dando tales como, poder detectar con antelación la existencia de un semáforo a una determinada distancia debido a la escasa visibilidad que a veces se producen por razones de las condiciones ambientales o debido a las distracciones que se pueden producir por otras razones. Situaciones que originan accidentes causando pérdidas humanas y de bienes materiales.

Partiendo de este gran problema es importante plantearse una idea clara de modificar o reforzar estos métodos, es decir, poder dotar de dispositivos que no solamente seguirían una determinada secuencia para su funcionamiento (*cambio de estados: rojo, ámbar, verde*), sino que también gracias a las innovaciones introducidas en su sistema tecnológico, tendrán la habilidad de tomar ciertas decisiones atendiendo a las circunstancias que se le presente, como puede ser por ejemplo: *actuar ante un exceso de velocidad del vehículo*.

Con el avance de estas nuevas tecnologías que se están desarrollando, el ser humano espera y confía en que haya dado en el clavo, de tal modo que, el funcionamiento de estos nuevos sistemas sean excelentes y proporcionen las respuestas esperadas con respecto al antiguo sistema.

1.1. Planteamiento de problema.

Teniendo en cuenta los avances tecnológicos y su incorporación en la vida cotidiana que, por mencionar algunos se destacaría los siguientes: la aparición de vehículos de alta gama y que alcanzan velocidades extremas en un breve espacio de tiempo; el uso de los *SmartPhones* que va siendo cada vez más el principal compañero del hombre y por consiguiente nos lleva a tener grandes distracciones, motivo por el cual ciertos conductores son menos prudentes sobre el volante o se sienten muy confiados a la hora de conducir. Otra de las causas que generarán atropellos pueden ser, la imprudencia de los propios peatones, los infinitos objetos presentes en zonas urbanas que, desviando un segundo la vista también podrían producir distracciones.

Ante estos problemas, la circulación va volviéndose insegura y las vidas de los peatones cada vez más corren peligro y/o aumentan los índices de riesgo, hasta tal extremo que se registran cada día grandes cifras de accidentes, por lo tanto, sería muy útil hacer uso de alguna herramienta que ayude a reducir estos puntos negros donde el índice de atropellos es mayor y proporcionar de alguna forma esta seguridad que se hace cada vez más escasa.

Recapitulando las conclusiones obtenidas en el estudio realizado por la *fundación Mutual Madrileña para la siniestralidad y atropellos*, sobresalen las siguientes:

- ✓ Uno de cada cinco accidentes de tráfico producidos en España es por atropello, a pesar de ser el accidente de circulación más fácil de prevenir.
- ✓ Casa año más de diez mil personas son atropellados en España, cerca d 400 fallecen y casi cerca de 3500 terminan con heridas graves.
- ✓ Un tercio de las víctimas por atropello son mayores de 60 años.
- ✓ Nueve de cada diez atropellos se producen en zonas urbanas.
- ✓ Las causas principales de los atropellos son las imprudencias de los peatones y el exceso de velocidad de los automóviles.
- ✓ El 70% de los atropellos se registran cuando el peatón está cruzando la vía de un lado para otro.

- ✓ En función de género, las mujeres sufren un 6% más de atropellos que los hombres.
- ✓ Otoño es la estación crítica del año para los atropellos, siendo así octubre el mes en el que se registran más atropellos.

Haciendo memoria de las épocas pasadas se puede decir que el tráfico vehicular nació específicamente en Europa a finales de la década de los 60 (**1960**) con el fin de promover un cambio en el comportamiento que adoptaban los conductores sobre el volante todo con el fin de reforzar la seguridad para las los niños ciclistas y todos los peatones en general.

Los ciudadanos frustrados por el paso del tráfico en sus calles y que las hacían inseguras, ciudades como Delft (Holanda) convirtieron sus calles como obstáculo, para evitar el paso de los vehículos motorizados y proporcionando un sitio seguro para sus niños. Esta fue una de las primeras soluciones adoptadas para controlar el tráfico y se denominó como “*yardas vivientes*” que hacían uso de de material casero, bancas, mesas... para crear barreras y obstaculizar el camino.

Haciendo un pequeño avance con el paso del tiempo, se implementaron los semáforos tradiciones definidos como dispositivos eléctricos para controlar el tráfico en las diferentes intersecciones de nuestras ciudades mediante una configuración de tres luces que sin embargo, a pesar de su instalación en casi todas las ciudades no han sido capaces de resolver todos los problemas del tráfico debido a su inadecuada sincronización entre semáforos y que estos no toman el flujo de vehículos en las calles y avenidas.

El único problema preocupando que se pretende resolver con el diseño de esta herramienta son las *pérdidas de vidas humanas debido a los atropellos producidos por los conductores*.

1.2. Formulación del problema.

Atendiendo al planteamiento expuesto anteriormente, este proyecto está basado en la investigación y en el desarrollo de un sistema que sirva de apoyo y refuerzo para proporcionar mayor seguridad en las señalizaciones del tráfico. Este mecanismo consiste en implantar un sistema de tecnología lumínica incrustada en el asfalto cuya función principal sería la de actuar como unas barreras visibles tanto para los peatones como para los conductores avisándoles sobre el estado de un **exceso velocidad** a medida que este se viene acercando desde exterior hacia el entorno físico o escenario de control, con el fin de evitar numerosas accidentes involuntarios.

En la siguiente *ilustración 1* se muestra un ejemplo de lo que se pretende conseguir con las barreras luminosas y la manera en la que estarán distribuidas.

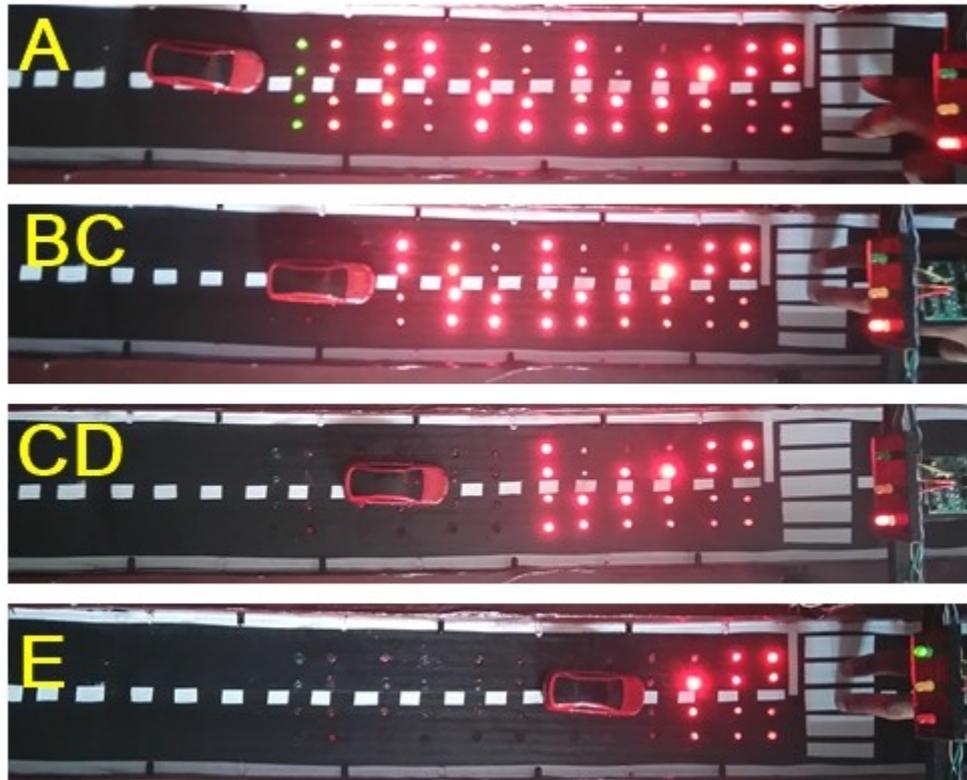


Ilustración 1: comportamiento del sistema

- ✚ **Nota:** la codificación A, BC, CD, E se refiere al tramo en que se encuentra el coche, más adelante se explica de manera detallada.

Con fin de hacerse una idea más clara sobre las **barreras luminosas**, su modo de funcionamiento es prácticamente similar a la tecnología utilizada en algunos electrodomésticos como los **equipos de música** para monitorizar mediante barritas de luz los decibelios o los distintos tonos de voz que se están reproduciendo, de tal manera que, a mayor tono de voz mayor serán los decibelios y mayor será la cantidad de barritas iluminadas y a medida que ese tono se va reduciendo, las barritas también se irán apagando secuencialmente. La única diferencia que tienen con las barreras luminosas es que, estas en vez de representar los niveles de tono, representa el **exceso de velocidad** de un automóvil, es decir, a medida que aumente la velocidad, mas barreras se irán iluminando y a medida que la reduzca, se irán apagando las barreras de un modo gradual.

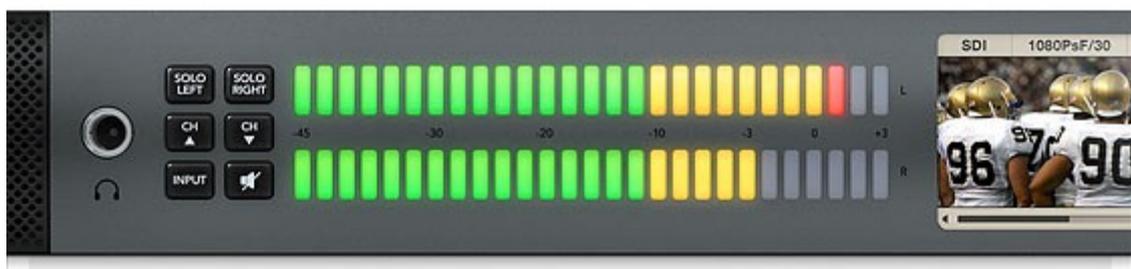


Ilustración 2: Monitor de niveles de tono de un equipo de música Blackmagic

Siendo esta una tecnología cuya utilización ya lleva décadas, una visión más familiar de esta representación sería la que se muestra en la *ilustración 3*.

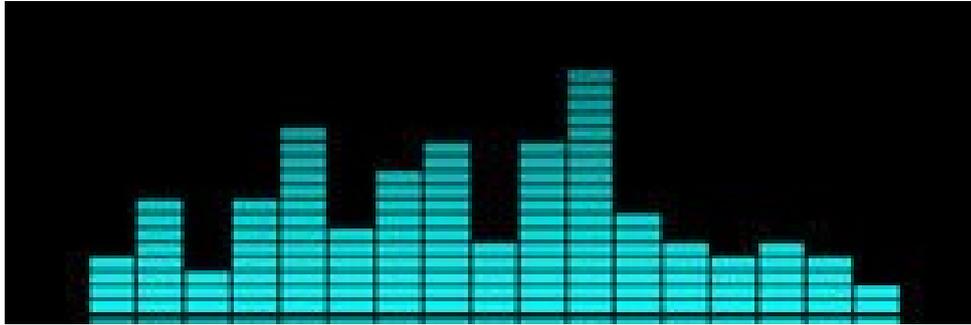


Ilustración 3: Monitor de tonos de Sony Walkman

El proyecto está pensado para hacer uso de estas barreras luminosas en momentos de poca luminosidad ya sea en días con mucha niebla, pero sobre todo de noche ya que es el momento en que más peligro puede correr las personas.

Este nuevo sistema no solo consistiría en instalar barreras luminosas y que actúasen sin estar regidas bajo ninguna norma. El proyecto está pensado para proporcionarle al controlador de estos una cierta inteligencia a base de los siguientes **aspectos fundamentales**:

- ❖ Detección de la presencia de los vehículos en el escenario con la ayuda de unas barreras infrarrojas
- ❖ Determinación de la evolución de estos vehículos, es decir, conocer su velocidad gracias a la utilización de un regulador PID que genere una salida de iluminación de las barreras luminosas en función de la velocidad del automóvil a lo largo del tiempo
- ❖ Determinar en todo momento la ausencia o presencia de los peatones en caso de que se esté cruzando las vías con la ayuda de un sensor hábil para esa función.

Teniendo ya una idea general sobre lo que se pretende hacer en este proyecto surgen preguntas como *¿Cómo desarrollar un sistema de barreras luminosas?*, *¿Cuáles son los requisitos para este sistema?*, *¿Cómo estará estructurado el sistema?* Y cuyas respuestas se irán concretizando a lo largo de su desarrollo.

CAPÍTULO 2: *OBJETIVOS*

2. OBJETIVOS

La realización de este proyecto viene enfocada principalmente en el **diseño de un sistema de barreras luminosas y visibles e inteligentes** para su posible uso general en todo el mundo con el fin de mejorar y reforzar aspectos que hasta ahora siguen generando incidentes desagradables en el tráfico y proporcionar mayor seguridad en los pasos de peatones.

Basándose en este enfoque general, lo que se pretende hacer es incorporar en los pasos de peatones actuales o en los del futuro, unas “**barreras luminosas incrustadas en el asfalto en una determinada zona**”, las cuales serán el componente que ayudará a los conductores a evitar o a reducir sobre todo el mayor número de accidentes que se producen en las intersecciones, ya sean colisiones y/o atropellos.



Ilustración 4: problemas registrados (A) solución diseñada (B)

Fijándose en la ilustración 2 se puede ver dos escenarios completamente diferentes pero directamente relacionados. En la **imagen A** se puede ver como se está atendiendo a un hombre que ha sido alcanzado por un coche mientras cruzaba la vía; mientras que en la **imagen B** se presenta una posible solución que resolvería este problema.

Este sistema será utilizado como un modo de alerta prolongada iluminando las barreras y así avisará a un conductor del nivel de riesgo en que se encuentra una vez entrado en la zona de control, permitiéndole poder reducir o regular su velocidad antes de llegar al final

del tramo, de tal manera que la cantidad de barreras iluminadas serán directamente proporcional a velocidad del automóvil, es decir, **a mayor velocidad, mayor será el número de barreras iluminadas** y a medida que este vaya reduciendo la velocidad las barreras también se irán apagando gradualmente.

Generalmente este diseño está pensado para que se pueda elaborar aplicando conocimientos de la Ingeniería Eléctrica, Electrónica y conocimientos informáticos, mediante la incorporación de **Micro-controladores** tales como el Arduino. Este micro-controladores será programado a través de su propio lenguaje de programación.

Según el diseño final, se procederá a la instalación de una serie de **sensores** necesarios y específicos que ayudarían a obtener o proporcionar información para el correcto funcionamiento del diseño, que sean capaces de realizar las funciones específicas según las cuales hayan sido elegidos.

Sin embargo, para realizar o conseguir el objetivo u objetivos expuestos anteriores y poder hacer realidad este proyecto es necesario satisfacer ciertos puntos tales como:

2.1. Implementación de unas barreras infrarrojas.

La implementación de estos módulos en el sistema contribuye en la detección de vehículos y en el cálculo de la velocidad de los automóviles gracias al tipo regulador utilizado. Sin la utilización de esos dispositivos no sería posible la creación de este nuevo sistema.

2.2. Implementación de unas barreras luminosas y visibles sobre el asfalto

Para este proyecto se considera este punto como el principal para que se pueda llevar a cabo. El motivo por el que es tan importante diseñar unas barreras luminosas e instalarlas en el semáforo, se debe a la necesidad de querer contribuir en el control del tráfico pero sobre todo para poder evitar o reducir en número de atropellos que se registran cada día en dichas intersecciones.

Lo que se pretende conseguir con esas barreras es que se vayan iluminando de manera gradual o progresiva desde la primera hasta la más cercana al automóvil, a medida que este aumenta su velocidad y se irán apagando gradualmente a medida que éste vaya desacelerando. Solo de este modo el conductor tendría constancia de su estado sobre el volante. Así mismo como, alertar a los conductores de la presencia o proximidad de un **paso de peatones** tras iluminarse las luces que constituyen las barreras lumínicas.

2.3. Implementación de un sensor de movimiento

Como ya se mencionó anteriormente, la utilización de este dispositivo era una de las funciones fundamentales que debía realizar el sistema ya que, casi el 80% del las decisiones tomadas por el controlador se realizaban en base a la información que proporcionaría ese sensor.

Este sensor está instalado en el soporte horizontal del semáforo y su campo de trabajo será sobre el paso de peatones, en el cual estará comprobando en todo momento si se detecta presencia de seres vivos cruzando las vías, ya sea de manera legal o ilegal y enviar dicha información al controlador.

Se necesita un pin digital para enviar la señal al controlador y su modo de operación es binario, es decir, cuando detecta presencia es un **1** y en caso contrario un **0**.

2.4. Programación y modelado del comportamiento del sistema

Este vendrá a ser unas de las cosas más importantes de este proyecto ya que se hará uso de un lenguaje de programación para modelar el comportamiento artificial del sistema, así como el control de sensores y actuadores. Esa programación se centrará en detectar vehículos, calcular parámetros como la velocidad, la detección de peatones y así determinar la cantidad de barreras que se habría que iluminar en función a la cantidad de barreras infrarrojas cortadas y a los resultados de los parámetros ya calculados.

Esta secuencia llevará consigo todas las funciones que se necesitaría que se realicen y que pueda ser bien interpretado por el micro-controlador y este a su vez poder ejecutar dichas funciones.

Dependiendo del tipo de micro-controlador elegido, este lenguaje de programación podrá variar o incluso hay micro-controladores que se adaptan a dos o más lenguajes de programación. En este caso el lenguaje de programación usado será **C** que es el lenguaje predeterminado e integrado en nuestro micro-controlador.

CAPÍTULO 3: *ESTADO DEL* *CONOCIMIENTO*

3. ESTADO DEL CONOCIMIENTO

En este capítulo del proyecto se partirá hablando sobre el origen de los semáforos, es decir, su primera aparición también se mencionaría su etimología hasta llegar a los semáforos actuales y ya a finales del capítulo se comentará acerca de las nuevas tecnologías que se están desarrollando para reforzar el tráfico como *por ejemplo (la inserción de las barreras luminosas e inteligentes para reforzar la seguridad en los pasos de peatones.)*

3.1. Etimología.

Normalmente la palabra semáforo procede del griego: «σημα (sema)», que significa señal, y «φόρος (foros)», que significa portador, es decir, semáforo es lo que “lleva las señales” [1].

Pero desde hacía siglos atrás el castellano consideraba semáforos a unas enormes torres de señales que se construían por todo el ámbito territorial de modo que, por medio de señales ópticas tales como *las luces de la noche o banderas durante el día*, se comunicaban las noticias importantes y mucho más deprisa que la vieja usanza del *caballo al galope*.

Otro de los métodos arcaicos usados en esa época fueron implantados en estaciones y desde los cuales se transmitían las señales del telégrafo óptico establecido en las costas y en los puertos teniendo como objetivo la llegada y las maniobras de los buques que navegaban a la vista y procedentes de alta mar.

Éste semáforos rudimentarios no eran nada más que y nada menos que un elevado mástil en el cual los *vigías* efectuaban las señales por medio del travesaño con unas bolas y con las banderas y si era de noche la mejor manera de hacerlo era con la luminaria.

Actualmente el término semáforo es considerado como un aparato mecánico y eléctrico que regula el tráfico en los vehículos en los caminos y viene definido para **Real Academia Española** como: *aparato eléctrico de señales luminosas para regular la circulación*

3.2. Historia de los semáforos.

Como bien se sabe, los semáforos llamados también señales para controlar el tráfico son dispositivos situados en intersecciones viales o en otros lugares para regular el tráfico y no solo el tráfico peatonal sino también en otros sectores tales como los lugares portuarios o en ferrocarriles.

3.2.1. Origen

La historia de los semáforos viene marcada por una evolución constante originada como muchas de las otras invenciones de hombre para conseguir una comodidad. Este tiene como esencia la regulación de la circulación que se viene registrando desde tiempo remotamente Romano hasta la actualidad.

El semáforo fue creado a principios XX con la idea de solventar las aglomeraciones de vehículos que se registraba en las ciudades, problema que se venía arrastrando desde la época romana y que no se obtenía una solución eficaz y segura ya que utilizaban métodos

poco ortodoxos basados en la utilización de antorchas en las calzadas de las ciudades.

La idea de los semáforos actuales está basada en los sistemas que se crearon siglos después para el control de tráfico ferrovial y fluvial que eran señales lumínicas que controlaban tanto barcos como ferrocarriles [2].

3.2.2. Pasado

La aparición de primer semáforo que hacía posible el control del tráfico en la calle se registró el **10 de diciembre en Londres de 1868** y estaba colocado en el exterior del parlamento y constaba de solo dos luces una verde y otra roja y funcionaban con gas.

Este gran avance en la seguridad vial fue posible gracias al Ingeniero Británico **John Peake Knight** especializado en señalización ferroviaria e inventor de las luces de tráfico o semáforo.

Este primer semáforo era manual constaba de dos brazos móviles para indicar el sentido en que tenían que detenerse y requerían de la presencia de un policía o un agente de tráfico que lo controlase todo el tiempo. Desafortunadamente pasados dos meses este primer semáforo explotó el **2 de enero 1869** hiriendo de gravedad al agente de policía. Tras este desastroso accidente esta gran invención fue retirada de las calles lo que provocó nuevamente el caos total en la aglomeración de vehículos [3].

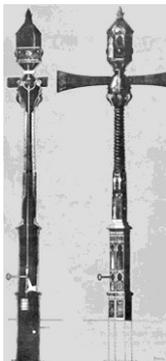


Ilustración 5: Modelos de los primeros semáforos

3.2.2.1. Elección de los colores

La principal razón por la que esas luces emitían un color verde y rojo con el fin de regular el tráfico era debido a una técnica heredada del mundo del ferrocarril y que esos a su vez la heredaron del transporte marítimo. Retrocediendo varios siglos atrás se puede afirmar que los barcos o buques de transporte ya venían utilizando un determinado código de colores para señalar el derecho de paso la cual no sido cambiado hasta hoy día y se ha trasladado a las alas de los aviones (**rojo a babor y verde al estribor**) y de modo que si un barco se acercaba en sentido perpendicular al otro, uno de ellos tenía la posibilidad de ver la luz roja en el babor del otro, es decir que se acercaba por la derecha y el barco que se acercaba en este lado derecho veía la luz verde en el estribor del otro barco y por lo tanto, el timonel que veía la luz roja sabía que estaba en la obligación de ceder el paso al otro barco permitiendo que este pudiera continuar su trayecto [4].

3.2.3. Evolución

Debido a la poca efectividad que tenían los semáforos manuales, nace la necesidad de saber y conocer cuando un semáforo sería capaz de cambiar de estado sin la necesidad de la presencia de un agente. En 1910 **Ernest Surrine** mejora el semáforo y dio un gran avance en convertirlos en automáticos y cuyo sistema de Surrine usó las palabras no iluminadas «*stop*» (detenerse) y «*proceed*» (proceder).

En 1912 A pesar de este gran paso otros hombres entendidos en la materia como **Lester Wire**, un oficial de la policía optó por volver al antiguo sistema manual a que le iba agregar ciertas modificaciones como cambian las luces que funcionaban a base de gas con luces eléctricas y posteriormente incorporarle un zumbador que tenía como misión advertir el cambio de estado. Otras de las cosas que añadió a esta modificación tecnológica fue permitir a las estaciones de bomberos y policía cambiar el estado de semáforo en caso de emergencia. Pero a pesar de todas las opciones presentadas para esta mejora su invención nunca fue patentada debido a que era un empleado del gobierno estadounidense.

Conforme al avance y la innovación del anterior modelo T se registraron ciertos avances en la ingeniería eléctrica ya que efectuaron mejoras en las señalizaciones posibles. Todo esto ocurrió a consecuencia de que en aquella época en Estados Unidos se abarcaba la más alta tasa de propiedades de automóviles en sus ciudades ya que el coche se convirtió en el principal juguete de hombre rico hacia el año 1910. Todos esos *hobbies* hacia el hombre no traían más que congestiones en las ciudades y situaciones amenazadoras ya que las antiguas señales de tráfico implantadas no proporcionaban la seguridad del ciudadano, razón por la cual era de vital importancia crear una mejoras en el antiguo sistema de control de vehículos.

Vista la problemática que se creaban en las ciudades estadounidenses los ingeniosos no tardaron mucho el presentar una posible solución y por tanto el primer semáforo automático y que utilizaban luces rojas y verdades automáticas fue patentado por **William Ghiglieri** en San Francisco, California en 1917 [3][4].

3.2.3.1. Problemática en la incorporación de semáforos automáticos.

La inserción de los semáforos automáticos en las ciudades estadounidenses no fue nada fácil de modo que se llegó a registrar problemas tales como:

- **Adaptación peatonal.**

A pesar de todos los esfuerzos invertidos para conseguir una posible regulación de tráfico, en países como EE.UU. hubo ciertos problemas por parte de los ciudadanos relacionados con el nuevo sistema incorporado en el tráfico. De modo que a los peatones les resultaba muy difícil acostumbrarse a las nuevas luces automáticas debido al cambio del comportamiento de regulación ya establecido en el antiguo sistema. Presentaban un ajuste más exigente y tenían que acatarse a las nuevas reglas tales como la manera de conducir. [3]

- **Discriminación en la ciudadanía.**

A falta de una organización entre los peatones y de los automovilistas y debido a la falta de un equivalente como la Asociación de Peatones en Gran Bretaña o la Asociación de la seguridad vial de Alemania, todos se influían en la ciudad de San Luis Misuri para bloquear una ordenanza contra el cruce de carretera en el punto didáctico en 1936.

Basándose en la impotencia policial que se daba en esta época, los ingenieros de tráfico trataban a los peatones como ciudadanos de segunda clase hasta tal extremo de que en un artículo de un periódico de 1921 estos fueron definidos “*como un obstáculo muy grave*”.

Ante la insistencia de los agentes de tráfico, los ayuntamientos remplazan esta regla antigua con nuevas ordenanzas que daban a los automóviles todo el derecho de vía y en varios libros de la ingeniería del tráfico se recalca que era más importante la protección vital que la velocidad de transporte. Pero a su vez estos libros afirmaban que la seguridad y prevención de accidentes parecía ser la última preocupación de los funcionarios.

Ante el problema de la inseguridad vial, en 1920 **William Potts** un oficial de la policía de Detroit e inventor de otros modelos de señales de tráfico propuso la incorporación de la actual de la luz amarilla que sirviera de señal para avisar a los conductores y peatones del inminente cambio de luz roja. A pesar de esta posible solución su idea no fue patentada por la misma razones por las que se negó patentar la idea de **Wire**, trabajaban para el gobierno

En 1923 **Garrett Morgan** en 1923 fue la primera persona en lograr la patente por un semáforo de tres etapas eléctrico aunque era diferente a los actuales ya que también llevaban dos brazos y palabras iluminadas.

Otro inventor **Charles Marshall** en 1936 crea una señal rotatoria con marcadores mecánicos con una estructura similar a la de un reloj pero que indicaba el tiempo restante ante el cambio de estado de un semáforo [3].

3.2.4. Presente.

Al igual que otras tecnologías, la tecnología de los semáforos no ha parado de crecer y ha producido una profunda evolución con el tiempo y debido a su gran rentabilidad y utilidad es necesario tener en cuenta el gran ahorro energético que estos producirían a la hora de fabricarlos.

3.2.4.1. Cambio de lámparas incandescentes por LED

Teniendo en cuenta el gran consumo energético de las bobillas incandescentes y visto en la necesidad de estos de tenerlos funcionando en todo el día sin disponer de la posibilidad de ser apagados salvo por mantenimiento, las antiguas bobillas usadas han ido perdiendo terreno y se han ido implantando el uso de luz **LED** debido al gran ahorro energético que estos brindan así como la gran luminosidad que estos proporcionan y su gran duración vital.

Estas lámparas LED utilizan solo el 10% de toda la energía que les proporcionan y responden una vida media superior a 50 veces mayor que la de las lámparas incandescentes. Vista estas ventajas entre otras que más abajo se citaran, los fabricantes han optado por estas luces debido a su fiabilidad y por la seguridad que estos pueden

proporcionar [5].



Ilustración 6: Luces Led de los semáforos actuales

3.2.4.2. Ventajas de un LED

- **Proporciona mayor vida útil de las lámparas:** mientras las lámparas halógenas tienen una duración de 8000 horas los leds ofrecen unas 100.000 horas
- **Muy bajo consumo energético:** debido a que son unos diodos por lo que no necesitan tensiones elevadas para conseguir que se enciendan
- **Mayor ahorro energético:** solo consumen un 10% de toda la energía entregada
- **Respeto por el medio ambiente:** no emite ningún tipo de gases ni algún otro producto tóxico en su fundición
- **Simple recambio en caso de avería:** no es necesario cambiar toda la matriz de leds, solo el o los que se han fundido en todo el tolde lo cual origina un gran ahorro en el coste.
- **Mantenimiento muy flexible y mínimo:** gracias a su duración y a la fiabilidad, los costes del mantenimiento se reducen enormemente.
- **Distribución luminosa uniforme:** de la luminosidad en los semáforos de leds, en lugar del único punto de luz que existe en los semáforos halógenos.
- **Control de la luminosidad:** conocido también como ‘dimming’, es posible variar la tensión del controlador de tráfico para que, durante la noche, se baje la luminosidad del mismo, ajustándola a la luz solar existente y permitiendo un ahorro energético mayor.
- **Bajo nivel del deslumbramiento:** debido a que tiene una luminosidad regulable eso evita crear confusiones al conductor
- **Unidad óptica** a prueba de la luz solar y alto contraste de la luz solar.
- **Mayor seguridad vial.**

- **Evita la fundición** de los diodos al estar formado por una matriz de diodos independientes.
- **Menor emisión de calor:** Los semáforos de leds emiten mucho menos calor en su operación que los semáforos halógenos
- **Eliminación del efecto fantasma:** los semáforos halógenos, al utilizar una óptica diferente, llegan a producir un efecto por el cual, según el reflejo del sol, pueden llegar a confundir al conductor al dar la impresión de estar encendido cuando no es así. Con los semáforos de led's, este efecto no está presente.
- Se pueden cambiar la imágenes fácilmente (*por ejemplo.*, en Madrid durante el día de la mujer añadieron falda a los muñequitos.)
- Animación de peatones móviles.
- Incorporación de baterías adicionales: Su bajo consumo permite su funcionamiento automático mediante una batería en un cierto tiempo.
- Incorporación de sonidos intermitentes: para cambiar de estado.

3.3. Constitución y funcionamiento de los semáforos convencionales

Dado que en la mayoría de los países hasta la fecha siguen haciendo uso de los semáforos automáticos no inteligentes, este apartado se centra en explicar detalladamente los semáforos convencionales o actuales. Como bien ya se había dicho anteriormente los semáforos son señales de tráfico que nos permiten controlar y poder regular el tráfico de vehículos y otros medios de transportes ya sean terrestres y o navales.

3.3.1. Componentes de un semáforo convencional.

Los semáforos suelen estar formados de los siguientes elementos esenciales:

- a) **Cabeza:** Normalmente suele ser la armadura que contiene las partes visibles del semáforo. Atendiendo a su modelo, cada cabeza suele tener un número determinado de caras orientadas en diferentes direcciones.
- b) **Soportes:** Los soportes suelen ser la estructura metálica que se utiliza para sustentar a la cabeza de manera que les permitan algunos ajustes angulares verticales y horizontales y en cuyo interior suele llevar el cableado que suministra energía a los led instalados en la cabeza.

Atendiendo a la ubicación de los semáforos los soportes pueden clasificarse en:

- ❖ ubicados al lado de las vías:
 - Postes.
 - Ménsulas cortas
- ❖ ubicados en las vías:

- Ménsulas largas sujetos a postes laterales
- cables de suspensión.
- Postes y pedestales en islas

c) **Caras:** No son nada más que las distintas luces que constituyen el semáforo y suelen variar desde dos o más de tres luces siendo las más comúnmente usadas las de tres luces.

d) **Lentes:** Es uno de los componentes muy importantes ya que es la parte de ella unidad óptica que por refracción cumple la misión de dirigir la luz proveniente de las lámparas y de su reflector en la dirección deseada.

e) **Visera:** Es un elemento de protección colocado alrededor o encima de cada una de las unidades ópticas y cuya función es la de evitar la incidencia de los rayos solares que podrían crear esa impresión de estar siendo iluminadas. Otra de sus funciones es la de evitar que la señal emitida por la unidad óptica sea vista desde otros lugares hacia el cual este orientado y enfocando. Como el caso de las lentes, esta parte está desapareciendo ya que los nuevos semáforos de leds iluminan de mejor forma que los antiguos.

f) **Placa de contraste:** Elemento utilizado como apoyo para incrementar la visibilidad del semáforo y ser más destacable con el fin de evitar que otras fuentes lumínicas confundan al piloto.

3.3.2. Funcionamiento de un semáforo convencional

Los semáforos de tres etapas, es de decir los de tres luces son actualmente los más usados en todo el mundo, y la función de cada estado viene descrita a continuación

- **Rojo:** Es la señalización que indica a los vehículos que están obligados a detenerse para cederles el paso a los peatones. Esta señalización puede ser de inmediato o parpadeando
- **Amarillo o Ámbar:** es una señal de tránsito que produce antes de pasar de esta rojo a verde y suele significar precaución. Evita a que se produzca un cambio brusco entre ambos estados y permite alertar tanto los conductores como a los peatones.
- **verde:** esta es la señalización que permite a los conductores seguir el recorrido sin necesidad de parar y advierte a la vez a los peatones que no tienen acceso a cruzar la vía debido a que corren un gran riesgo de ser alcanzados por algún vehículo.

3.3.3. Equipo de control de un semáforo.

Para conseguir el funcionamiento de un semáforo es necesario disponer de un equipo que lleve instalado un circuito automático y secuencial a través del cual las salidas irán cambiando de estado.

Este equipo de control normalmente suele ser un mecanismo electromecánico o electrónico cuya misión es la ordenar los cambios de estados en los semáforos y que adicionalmente pueden realizar funciones como:

- ✓ Procesar información generada por los detectores para ajustar los tiempos a las necesidades de la intersección.
- ✓ Recibir y enviar información a un centro de control o controlador maestro con el fin de operar de forma coordinada.
- ✓ Proveer los elementos que garanticen la seguridad de los usuarios evitando señalizaciones conflictivas y reportar al centro de control el tipo de falla que pueda presentar.

3.3.4. Tipos de semáforos: semáforos para funciones especiales.

Actualmente no solo está uso los semáforos de tres estados para el control del tránsito, es decir, se han ido introduciendo o instalando otros tipos de semáforos que colaboran de igual forma la regulación del tránsito en aspectos secundarios. De entre estos podemos destacar los siguientes:

3.3.4.1. Control de direcciones.

Los semáforos para el cambio de direcciones son utilizados en determinadas direcciones especificadas. Siendo su uso habitual cuando el tráfico proveniente de la dirección contraria evita el correcto flujo de los vehículos que giran, o cuando los vehículos que viran impiden el correcto flujo de los vehículos que vienen en la dirección contraria. Esto puede crear congestión vehicular y aumentar el riesgo de accidentes.

3.3.4.2. Control del carril.

Se pueden usar semáforos para controlar el flujo vehicular de cada carril. Esto puede ser usado para advertir a los conductores que uno de los carriles se encuentra cerrado o que no es seguro circular por este. Este tipo de semáforos también es usado cuando existe una vía reversible, que cambia la dirección del tráfico, por lo que es necesario advertir a los conductores sobre que carriles son utilizables.



Ilustración 7: semáforo para el control de carril

Además de las típicas luces, la mayoría de estos semáforos también utilizan flechas combinadas con una “X” para indicar el estado de cada carril.

3.3.4.3. Peatonales

En muchas intersecciones se usan además semáforos peatonales para indicar al peatón el momento seguro para que pueda cruzar la intersección. También se pueden usar para dar preferencia a los peatones sobre el resto del tráfico de la vía.

La mayoría de estos semáforos no cuenta con una luz intermedia entre el verde y el rojo, por lo que normalmente la luz verde o roja parpadea dos o tres veces para anunciar el próximo cambio al rojo.



Ilustración 8: semáforo peatonal

En algunos casos los semáforos peatonales pueden tener contadores de tiempo para que el peatón pueda juzgar si tiene tiempo suficiente para cruzar la vía, en el momento en que el contador llega a cero inmediatamente el semáforo peatonal cambia a rojo.

En otros casos el semáforo puede incluir un botón para que el peatón pueda pedir manualmente el paso. Esto puede servir para evitar la detención innecesaria de los vehículos cuando en realidad no existe ningún peatón queriendo cruzar la vía, o para disminuir el tiempo de espera de los peatones.

Tipos.

Existen Diferentes tipos de semáforos peatonales; entre los más destacados tenemos:

- Semáforo peatonal con contador para la luz verde.
- Semáforo peatonal combinado de 3 etapas. En la tercera etapa la luz verde parpadea para indicar el próximo cambio a rojo.
- Semáforo peatonal con la luz ámbar intermedia en Suiza
- Animación de uno de los tipos de semáforos peatonales usado en Estados Unidos

3.3.4.4. Para ciclistas.

En caso de existir una cicloavía o de que la vía sea compartida por ciclistas se pueden usar semáforos especiales para ellos, logrando así mejorar su seguridad y la de los que los rodean. Normalmente se los usa para dar prioridad a los ciclistas o para mantenerlos alejados de otros usuarios de la vía, como peatones o vehículos.



Ilustración 9: semáforo para ciclistas

3.3.4.5. Para transporte público.

En algunos casos se usan semáforos para dar prioridad o controlar independientemente el flujo de vehículos del **transporte público**. En algunos casos cuando la vía es compartida con un tipo diferente de transporte como los tranvías se usan señales diferentes para estos. Estos semáforos son más comunes en Europa, aunque pueden ser vistos alrededor del mundo. En muchos casos no utilizan las típicas tres luces (rojo, amarillo y verde) a cambio de sistemas más complejos para dar mejores indicaciones a los conductores.

3.3.4.6. Evolución y datos globales de los semáforos.

El 4 de Agosto de 1914 se pudo instalar el primer semáforo moderno en los **Estados Unidos** y constaba de con luces rojas y verdes, colocadas sobre unos soportes con forma de brazo y también llevaba un zumbador para avisar el cambio de estados [6].

En 1930 en la **Cd. de México**, los capitalinos se encontraron con los primeros indicadores de señalamientos de tránsito vehicular (semáforos). Eran manuales y los operaba un agente moviendo una palanca para sacar un letrero que decía alto o siga, según el caso [7].

En 1926, se instaló el primero en **Madrid** convirtiéndose en el primer semáforo de España, entre las calles del Barquillo y Alcalá.

Buenos Aires, es la ciudad con más semáforos por habitante. Más semáforos pero hace falta rapidez para cruzarlos. La ciudad de Buenos Aires es una de las que dispone de mayor número de intersecciones.

El 5 de abril de 2008, el Ayuntamiento de Jaén (España) instaló el primer semáforo feminista un «primer paso» igualitario en la señalización, pues la tradicional silueta 'andante' cambia el pantalón por la falda.

El alegre hombrecillo verde o rojo de los semáforos, conocido en Alemania como '*Ampelmann*', fue introducido por las autoridades comunistas en la Alemania del Este y desde la caída del muro de Berlín hace 15 años se ha convertido en un fenómeno de marketing [8].

Palma de Mallorca se ha convertido en la primera ciudad de España en contar en sus calles con un semáforo dotado con un mando a distancia que permite que se encienda solo cuando lo necesita una persona ciega.

El día 6 de marzo de 2006, el tráfico de la Ronda de la Muralla de **Lugo** protagonizó una iniciativa de la artista uruguaya Luz Darriba. Los 'muñequitos' de los 72 semáforos de la Ronda de la Muralla que regulan el paso de peatones vistieron falda en homenaje a la mujer trabajadora.

El primer semáforo de Argentina alimentado por energía solar y controlada en forma inalámbrica a través de Internet funciona desde el 15 de diciembre de 2008.

En Akureyri, Islandia, las luces de los semáforos tienen forma de corazón.

3.3.5. Empresas Fabricantes y Distribuidora

Teniendo en cuenta que son incontables la cantidad de empresas fabricantes de señales de tráfico en todo el mundo, la siguiente tabla muestra datos recogidos de ciertas empresas operantes y en España que son unas de las más destacadas.

Tabla 1: Empresas destacadas fabricantes de semáforos

Nombre	Localidad	Características
BENITO URBAN	Barcelona	Diseño Fabrica y distribución
LINEAS Y CABLES S.A	Madrid	Empresa de semáforos y señalizaciones luminosas. Proveedor, distribuidor, subcontratista.
CONFUSO S.A	Barcelona	Proveedor y Fabricante
ECOYMA S.A	Barcelona	Proveedor y Fabricante
VALMONT IBERICA	Madrid	Empresa de semáforos y señalizaciones luminosas. Diseño Fabrica y distribución
DITRONIC	Mérida	Fabrica y distribuidor
TRAFFIC FUTURA	Madrid	Fabrica y distribuidor
SCHLETTER GmbH	Almería	Fabrica y distribuidor

3.4. Semáforos Inteligentes y los sistemas aplicados

En este apartado se empieza a entrar poco a poco en la idea para la cual se quiere llevar a cabo este proyecto y por lo tanto es necesario empezar hablando sobre lo que viene a constituir el sistema que hace posible que diversos dispositivos electrónicos puedan funcionar.

3.4.1. Sistemas

El término ‘sistema’ puede ser considerado como un conjunto de mecanismos y herramientas que hacen posible o permiten la creación así mismo como la interconexión entre diferentes componentes al software, junto con una colección de servicios para facilitar las funciones de los componentes que están instalados en él y ayudarles a ejecutar dichas funciones.

Actualmente existen diversos sistemas y los más comunes y usados son sistemas electrónicos y los sistemas inteligentes.

3.4.1.1. Sistemas electrónicos.

Se consideran como sistemas electrónicos aquellos que están formados por un conjunto de circuitos cuyos componentes interactúan entre sí con el fin de obtener un resultado común.

Una manera más clara para entender los sistemas electrónicos consiste en dividirlos en las siguientes partes tales como vienen a ser los:

a) Entradas Inputs.

Se les denomina entradas a los sensores o transductores electrónicos o mecánicos cuya función es la de tomar la señal ya sea en forma de temperatura, presión, intensidad, etc del entorno físico y convertirla en señales de corriente y tensión. *Por Ejemplo: el termopar, la foto resistencia para medir la intensidad de la luz, etc.*

b) Circuitos de procesamiento de señales.

Son los elementos conectados al circuito y cuya función es la de manipular, interpretar y transformar las señales de intensidad de corriente o como la tensión provenientes o tomadas de los transductores, es decir, de las entradas.

c) Salidas Outputs.

Suelen ser considerados como los actuadores u otros dispositivos como algunos sensores, que convierten las señales de corriente y la tensión en señales físicamente útiles. *Por ejemplo: una luz que se enciende automáticamente si se oscurece, una pantalla LCD que nos muestra presión medida en una tubería.*

3.4.1.2. Sistemas inteligentes.

Un sistema inteligente es aquel que puede ser definido como un programa de computación que reúne características y comportamientos asimilables a la inteligencia humana o animal.

El ‘concepto sistema inteligente’ es un concepto muy complejo y se usa incluso en sistemas inteligentes incompletos tal como es el caso de casas inteligentes o un sistema experto.

En realidad un sistema inteligente, es aquel que incluye ‘sentidos’ que le permiten recibir información atendiendo a su entorno. Dispone de la capacidad de actuar ya que trae consigo una memoria en la cual va almacenada los datos tomados de sus acciones y a través de la cual puede aprender cómo lograr su eficiencia y su rendimiento.

3.4.2. Semáforos inteligentes

Al día de hoy el hombre no ha dejado de investigar sobre la mejora de las señales de regulación de tráfico, de modo que, visto sometidos en la problemática de las grandes aglomeraciones de los vehículos en las vías y a la producción de accidentes por un fallo de regulación o por la misma congestión de los vehículos en los semáforos convencionales. Que ante todo han servido y siguen sirviendo de gran ayuda en casi todos los países pero que no resultan lo suficientemente eficientes y que carecen de inteligencia para reducir el índice de accidentes producidos cada año en todo el mundo.

Ante esta problemática nace la necesidad de crear un nuevo estilo de controlar el tráfico y que resulte a su vez muy eficiente y pueda resolver el 90% o 100% de los incidentes que se producen en las vías.

Basándose en esta teoría nace el nombre de **Semáforos Inteligentes** que han sido definidos como señales de tráfico capaces de tomar sus propias decisiones y actuar atendiendo a las circunstancias presenciadas en cada momento para regular y controlar el tráfico de manera correcta.

Dicho de otro modo son señales capaces de tomar decisiones atendiendo a los parámetros de una serie de parámetros de entrada como por ejemplo (*flujo del vehículos, velocidad media, identificar la calle, entre otros*) y que tenga un comportamiento dinámico.

Estos semáforos tienen como misión solucionar varios problemas de tránsito comunes en todo el mundo y por citar los más importantes tendremos:

- ✓ Congestión vehicular.
- ✓ Tiempos excesivos de viaje.
- ✓ Esperas innecesarias.
- ✓ Mayor gasto de combustible
- ✓ Mayor contaminación en el medio ambiente.

En la actualidad existen diferentes tecnologías que se hayan aplicado en esta invención pero se puede decir que todavía no se ha dado con la fórmula perfecta para la solución de los problemas de las ciudades. En este apartado se irá hablando de las diferentes tecnologías que van siendo implantadas y la forma de funcionamiento del mismo.

3.4.2.1. Tecnologías Aplicadas.

Dentro del marco de las tecnologías implementadas en las señales de tráfico, éstas pueden variar dependiendo de las funciones que se necesita realizar. Pueden ser decisiones sencillas por ejemplo (permanecer en luz rojo o verde) y en decisiones de mayor jerarquía como la de tomar una decisión a causa de un accidente, cambiar el flujo del tráfico, etc. Entre las tecnologías aplicadas se pueden destacar las siguientes.

3.4.2.1.1. Semáforos inteligentes con tarjetas RFID

La finalidad de esta tecnología se base en tener el control inteligente del tráfico mediante una variación del tiempo de espera en los distintos puntos de intersección atendiendo al grado de congestión de los vehículos.

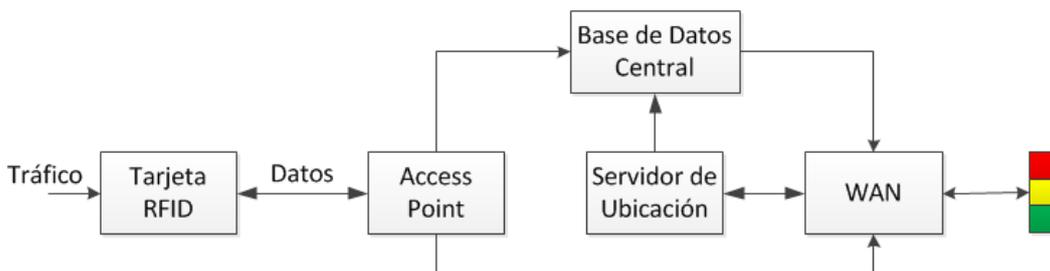


Ilustración 1: Esquema de un circuito usando tecnología RFID

Este tipo de sistemas suele constar de cuatro partes esenciales tales como:

- ❖ La tarjeta RFID
- ❖ Punto de acceso.
- ❖ Ubicación de servidores de Redes WAN
- ❖ Base centralizada de almacenar datos para la elección de alternativas.

La toma de decisiones parte de la información en tiempo real proporcionada por los sistemas RFID. Los datos captados son guardados en la base centralizada; Las Tarjetas RFID son sensores que captan la ubicación y el tiempo de cada vehículo y se guardan como una etiqueta que puede ser referenciada a través de un identificador único.

La situación de espera de un semáforo en cada uno de los cruces depende principalmente de la congestión, es decir, la longitud y de otros parámetros de entrada. La sincronización y el algoritmo de comunicaciones entre los semáforos utilizada en esta tecnología sea hace a través de internet de modo que todos los sensores ubicados en los diferentes puntos toman los tiempos, los tipos de vehículos y todos esos datos se van cargando en una tabla

de almacenamiento. Gracias a esos datos, es posible determinar la velocidad media de los automóviles que están transitando en un determinado momento o la velocidad después de estar un periodo de tiempo parado.

En el supuesto caso de que ocurriese una situación anómala, sería posible seguir controlando el tráfico ya que todos los datos acumulados previamente en la base de datos pueden ser utilizados. El sistema está capacitado para aprender las decisiones y podría producir un escenario general del flujo de tráfico mediante la identificación de una variedad de situaciones tales como:

- ❖ Determinar el patrón de circulación de vehículos en un determinado día.
- ❖ Determinar la secuencia más eficaz.
- ❖ Indicar la ruta más activa

El gran desafío es diseñar un sistema de gestión de tráfico electrónico sofisticado que sea capaz comprender e identificar el movimiento del tráfico y los destinos de rutina para toda una ciudad.

El mayor problema que presenta este tipo de tecnologías es que cada vehículo de tener su propia etiqueta RFID [9].

3.4.2.1.2. Tecnología Basada en Redes de Sensores Inalámbricos.

Esta tecnología comparte puntos en común con la de tarjetas RFID ya que cuenta con un sistema capaz de manipular de forma dinámica los tiempos de espera. Pero a diferencia de los Tecnología RFID estos destacan esencialmente dos partes principales que pueden ser:

- ❖ La red de sensores inalámbricos (Wireless Sensor Network o WSN)
- ❖ La estación Base (Base Station) cuya función a desempeñar es la de ejecutar los algoritmos de control.

Entrando en detalles sobre la red de sensores inalámbricos se puede decir que es un conjunto o un grupo de sensores diseñados atendiendo a una infraestructura capaz de proporcionar una comunicación del tráfico y al mismo tiempo regulando el flujo del tránsito.

Ante todo el proceso de este sistema, los sensores son los encargados de proporcionar la información o generar los datos del tráfico como por ejemplo: número de vehículos, los procesos de salida, la velocidad de cada vehículo o su longitud, Tras esta toma de información, estos son enviados a la base de datos en tiempo real.

El principal método utilizado para la comunicación suele ser el TDMA y con este se puede aprovechar de manera eficiente la energía ya que permitirá a los nodos de la red entrar en estados inactivos hasta que sean asignados sus espacios de tiempo.

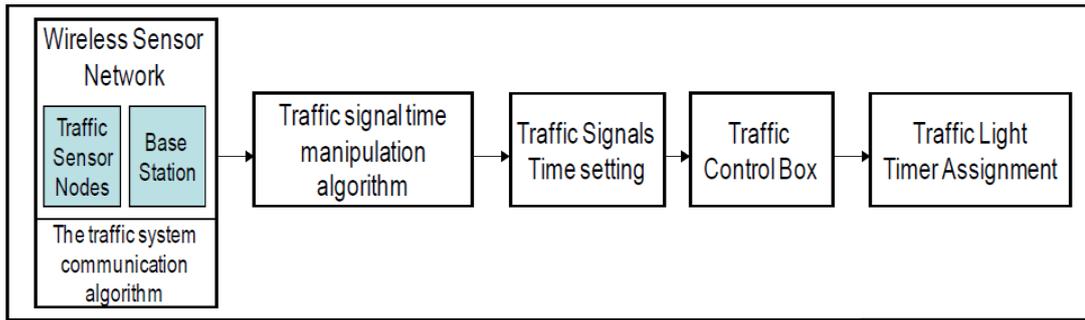


Ilustración 2: Esquema usando en tecnología de Red de sensores inalámbricos

Haciendo uso de circuito que arriba se muestra se puede decir que el encargado de detectar y contar los vehículos es el TSN (*Trac Sensor Nodes*), posteriormente se transfiere esta información periódicamente a la base de datos.

El **TSN** es un sistema de detección y cuenta con cuatro componentes en su composición:

- ✓ Sensor para detectar señales generadas por lo vehículos
- ✓ Procesador para los datos detectados
- ✓ unidad de comunicación para transferir los datos procesados a la base.
- ✓ Fuente de energía

El **TSCA** (*Trac System Communication Algorithm*) es otro de los componentes que las destacan en el circuito y estos han sido desarrollados básicamente para controlar las rutas de comunicación y evitando la desviación de señales entre todos los TSN y la base así como la interfaz con la caja de control del tráfico **TCB** (*tráfico control box*)

Una vez guiados los datos de tráfico a la base de datos por e TSCA estos son recogidos por los TSTMA (*Trac Signal Time manipulation algorithm*) para fijar duraciones de tiempo de señales de manera dinámica conforme a los parámetros recibidos. Para poder realizar estos cálculos se ha hecho uso de la **Matriz de conflictos** que es un matriz que determina los posibles casos que se pueden dar cuando se habilita cierto sentido del semáforo.

Este tipo de tecnología tiene la idea de presentar algo eficiente, adaptable y rentable que garantice el flujo de tráfico rápido y en forma eficiente [10].

3.4.2.1.3. Tecnología Mediante Procesamiento de Imágenes.

La tecnología de procesamiento de imágenes ha sido una de las que más puntos a favor han obtenido ya que permite controlar el tráfico utilizando un procesador de imagen permitiendo así poder medir la densidad de tráfico de acuerdo a los datos obtenidos en sus respectivos receptores y con ellos poder modificar los tiempos de espera en los semáforos.

■ Bloque Observar:

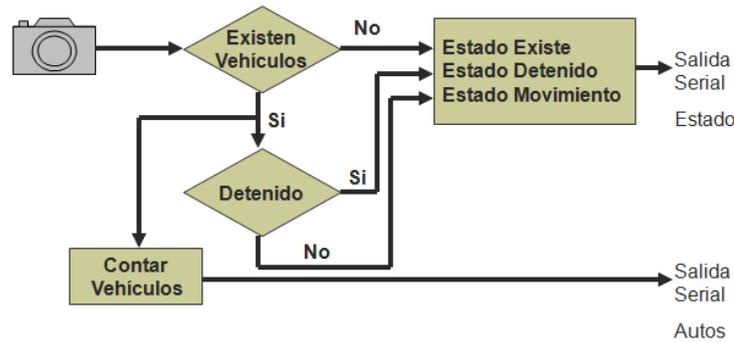


Ilustración 10: Esquema de un circuito con procesamiento de imágenes

La idea fundamental para este tipo de tecnologías puede subrayarse en los siguientes puntos principales:

- ✓ Adquisición de dos imágenes, una con la carretera vacía y la otra con el tráfico actual pero tomando como imagen de referencia la de la carretera vacía.
- ✓ Mediante un proceso de tratamiento de imágenes, ambas capturas son convertidas en RGB a la escala grises.
- ✓ Mejora de la imagen.
- ✓ Verificar coincidencia entre la imagen de referencia y la imagen con el tráfico actual.
- ✓ Después de un procedimiento de detección de bordes de ambas imágenes, los semáforos se pueden controlar basándose únicamente en el porcentaje de coincidencia.

Para esta misma tecnología se puede plantear otra Hipótesis enfocada desde otro punto de vista distinto al anterior y obtendríamos otro modo de funcionamiento que se puede detallar de la manera siguiente:

- **Observar**
 - ✓ Ver todas las direcciones donde se mueve el tránsito.
 - ✓ Ver los estados de los vehículos: movimiento, parado, vacío.
 - ✓ Hay vehículos ¿cuántos?
- **Procesar datos**
 - ✓ Averiguar las propiedades de las avenidas y dar paso al de mayor preferencia.
 - ✓ Determinar la mejor secuencia.
 - ✓ Si n hay transito utilizar el algoritmo tradicional.
- **Caso en que haya transito.**
 - ✓ No dejar atascada una dirección del tránsito.
 - ✓ Si las siguientes cuadras están congestionadas, no congestionarlas más.
- **Comunicación con los vecinos**

- ✓ Preguntar si sus calles están congestionadas.
- ✓ El estado del semáforo en dicha calle.

A pesar de la efectividad que presenta esta tecnología, ésta presenta problemas relacionados con el elevado coste de inversión. Pero ante todo destaca una ventaja muy importante que consiste en que pueden ser utilizados como sistemas de seguridad y control monitoreo de las calles 24 horas, entre otros aspectos. Todo esto con la misma cámara [11].

3.4.2.1.4. Tecnología basada en Inteligencia Artificial

Haciendo énfasis a este tipo tecnología se puede abarcar diversos temas tales como es el caso de la lógica difusa, algoritmos genéricos y refuerzo de aprendizaje [12].

- **Lógica Difusa**

La lógica difusa es una de las técnicas más comunes usadas en diseños robóticos y también en semáforos inteligentes. El sistema de Lógica difusa permite la aplicación de las reglas de la vida cotidiana y real similar a la manera de pensar de los seres humanos

El objetivo principal en el uso de este sistema de control se podría resumir en:

- ✓ Determinar la presencia y ausencia de vehículos a través de la imágenes tomadas en las rutas.
- ✓ Mantener la luz verde por más tiempo en caso de mayor flujo de vehículos.
- ✓ Si un carril está vacío, buscar un carril con vehículos y colocar dicho carril en verde.

La técnica de uso de la lógica difusa tiene como idea contar utilizando dos sensores electromagnéticos, colocados en el pavimento de modo que, el primero cuente la cantidad de vehículos que cruzan el semáforo y el otro cuenta la cantidad que llega a la intersección. Este método nos permite controlar la densidad del tráfico en los carriles y permite así mismo una mejor evolución en los cambios de los patrones de tráfico.

- **Refuerzo de Aprendizaje.**

La utilización de esta tecnología está basada en establecer una comunicación inalámbrica entre el vehículo y el dispositivo inteligente con el fin de que éste vaya tomando conocimiento de cuando los vehículos están parados esperando el cambio de la señal.

Básicamente la idea consiste en que mediante una comunicación *wireless* todos los coches indicaran su dirección así como su posición en la cola y así el micro controlador instalado en el semáforo será capaz de tomar decisión con el fin de optimizar el tiempo de espera mínimo de cada vehículo.

Gracias a la técnica del Refuerzo de aprendizaje es posible realizar este problema haciendo un seguimiento de los tiempos de espera de los automóviles y utilizar de forma inteligente su programación para calcular los promedios a largo plazo en los tiempos de espera utilizando algoritmos de programación dinámica.

3.4.3. Otros Sistemas usados en la actualidad

Actualmente los semáforos más usados en la mayoría de los países son los que cuenta con un temporizador cuya función es la de indicar tanto a los peatones como a los pilotos de los vehículos el tiempo de espera restante para que se produzca el cambio de estado en el momento exacto. También son usados para indicar a los peatones el tiempo estima para cruzar.



Ilustración 11: semáforo con contador

Otros de los semáforos muy utilizados son los sincronizados o los llamados ‘Onda Verde’ y cuyo misión es la de ofrecer a los conductores un tráfico fluido a través de varios cruces ya que están diseñados para tener activa la luz verde a una determinada velocidad dada y en una trayectoria determinada.



Ilustración 12: semáforo onda verde o sincronizada

También existen semáforos que deciden por sí mismo de acuerdo a unos parámetros de entrada de tiempo que va a permanecer en un estado determinado, ya sea rojo o verde. Estos nos permiten regular la circulación en zonas con mayor congestión.

Normalmente los sistemas mencionados son sistemas centralizados y por lo tanto dotan de una gran ventaja en cuando al mantenimiento ya que nos permitiría saber fácilmente si está fundida alguna de las luces, también se puede controlar la velocidad con cámaras instaladas o que no se cruce la luz roja.

En caso de un accidente el semáforo debe ser capaz de identificar tal situación, evitando que los vehículos sigan circulando en esa dirección, entonces es necesario que desvíe el tráfico en calles anteriores para evitar el congestionamiento.

- **TrafiRadar.** Esta viene a ser una de nuevas tecnologías implementadas con el objetivo de decidir cuánto tiempo debe permanecer el estado “*Ambar*” hasta que el vehículo haya cruzado la intersección, es decir, informa por si algún vehículo necesita más tiempo para pasar la intersección antes de que se cambie de estado y mantener al resto del tráfico detenido hasta que el coche haya pasado [13].

- **LYZUN.** es un sistema de Software Inteligente desarrollado a base de toras teorías entre la que se puede destacar como mas importante la Teoría de Colas. Este software desarrollado suele estar conectado a sensores ubicados en un determinado pavimento, los cuales miden la cantidad de automóviles que circulan por cada carril y ya en función de esta información proporcionada por el sensor se determinaría los retrasos de los vehículos en acceder a la intersección la relación entre la demanda y la capacidad del acceso, longitud media de las colas, la probabilidad de detención etc [14].

Todos estos parámetros son tomados por el Software para decidir de forma inteligente que hacer para manejar el tránsito de forma optimizada, por lo tanto un de las posibles soluciones es alargar la duración del estado verde o desviar el tránsito a otras calles para descongestionar la carga y facilitar la circulación vehicular, proceso que contribuiría también a la disminución de la contaminación del medio ambiente.

Esta tecnología está diseñada en un nuevo concepto de semaforización llamada Semáforos **IP**, con lo cual permite programar y controlar de manera rápida, inteligente y eficaz las posibles sincronizaciones de los semáforos mediante la utilización de un navegador WEB.

También cabe mencionar que estos semáforos disponen de tecnología suficiente como para poder ser conectados a la red de datos WIFI, de Fibra Óptica, o cableada que tenga el municipio en la actualidad para centralizar su manejo desde un centro de operaciones. Permitiendo manejar el tránsito adecuadamente ya que se interconectan con cámaras

- **Scats.** utiliza el abordaje adaptativo de tráfico en tiempo real para controlar el tráfico urbano, midiendo las condiciones de tráfico actual y, enseguida, hace el ajuste del: Ciclo Length (ciclo semafórico completo), *Splits* (duración del tiempo de los verdes) y Off Set (para garantizar un sincronismo entre los semáforos, la conocida “ola verde”) [15].

Su respuesta en tiempo real garantiza las condiciones más apropiadas y seguras a las fases de la señal de tránsito en los cruces.

Por otro lado, algunas ciudades cuenta con un avanzado sistema que permite la comunicación de los vehículos con los semáforos, de tal forma a que interactúen de acuerdo a la necesidad. Por el momento, sólo se aplica con vehículos especiales (Ambulancias, policías). *Por ejemplo: cuando ocurre un accidente, la ambulancia busca una ruta para llegar al punto del accidente, a medida que se acerca a algún cruce semafórico, éstos, se ponen en verde, de tal forma a facilitar el avance de la ambulancia.*

Otra forma de conexión entre los semáforos y los vehículos se hace a través de *bluetooth*. Los sistemas bluetooth permitirán:

- ✓ Conocer las rutas más habituales de los conductores, ya que se podrá determinar por qué semáforos (y el orden) pasa cada coche, y de esta forma se sabe la ruta seguida.
- ✓ Conocer el tiempo que le ha costado al coche hacer dicha ruta, es decir, se podrá saber con exactitud aquellas zonas donde hay atascos.

El uso de la “tecnología *bluetooth*” para conocer las rutas de los vehículos es discutible, ya que depende de que tantos vehículos cuenten con dicha tecnología.

3.5. Inteligencia de un semáforo o señales de tráfico programables

Cuando el ingeniero de ferrocarriles *John Peake Knight* inventó el primero semáforo que apareció con el fin de promover la regulación del tráfico, estos no eran más que un mecanismo mecánico que constaba de dos farolas colocadas en forma de cruz y cuya iluminación se conseguía por medio de un componente rotativo que producía una combustión con la gasolina, es decir, carecían de un sistema eléctrico y mucho menos de computadores.

Tras las modificación sufridas por diversos inventores e investigadores y gracias a la aparición de sistemas eléctricos, electromecánicos y automáticos, el funcionamiento de estos varió bastante ya que dicho funcionamiento ya se realizaba mediante autómatas no programables y que eran una combinación de diferentes componentes electromecánicos instalados en el llamado ‘*sistema de control*’. Posteriormente aparecieron los semáforos con sistemas combinatoriales y/o secuenciales casi semiprogramables y estos ya eran dispositivos eléctricos y o electrónicos.

Actualmente ya existen sistemas más complejos y más precisos como ya se mencionó en capítulos anteriores y cuyo funcionamiento se obtiene a base de computadores y micro-controladores programables mediante lenguajes más básicos u otros más altos, como se irán describiendo a continuación.

3.5.1. Lenguaje ensamblador

Se considera como lenguaje ensamblador a aquellos lenguajes de programación de bajo nivel para los computadores, microprocesadores, micro-controladores y otros circuitos integrados programables como pueden ser algunos autómatas programables. Estos lenguajes suelen implementar una representación simbólica de los códigos de máquina binarios así como otras constantes necesarias para realizar la programación de una determinada arquitectura dada de CPU y constituye la representación más directa del código máquina específico para cada arquitectura legible por un programador.

Muchos fabricantes de *hardware* han definido este modo de representación, la cual está basada en los nemónicos que simboliza los pasos de procesamiento, los registros del procesador, las posiciones de la memoria y otras características del lenguaje. Por lo tanto,

un lenguaje ensamblador es específico en cierta arquitectura de computador física o virtual y el cual difiere con los lenguajes de alto nivel ya que estos son los portables.

El ensamblador es un programa utilitario usado estrictamente para interpretar sentencias del lenguaje ensamblador al código máquina del computador objetivo. El programa realiza una traducción isomorfa, es decir, un mapeo de uno en uno partiendo desde las sentencias mnemónicas hasta las instrucciones y datos de la máquina. Este modo de interpretar las instrucciones es diferente al de los lenguajes de alto nivel ya que esos una sola declaración genera múltiples instrucciones de máquina.

Cabe mencionar que también existen otros métodos o mecanismos adicionales para facilitar el desarrollo del programa así como controlar en ensamblaje y ayudar a la depuración. Estos mecanismos adicionales son los llamados **macros ensambladores**

Este lenguaje sirvió de gran ayuda ya que fue la utilizada mientras aun se desarrollaba el software ya que no se disponía de lenguajes muy potentes y eran muy limitados los recursos. Actualmente se sigue haciendo uso de este lenguaje en algunos ámbitos tales como en ambientes académicos y de investigación que requieren la manipulación directa del hardware. También existen algunos microcontroladores que aun cuentan con el ensamblador como la única manera de manipularlos.

Dir.	Etiqueta	Instrucción	Código máquina ²³
		.begin	
		.org 2048	
	a_start	.equ 3000	
2048		ld length,%	
2064	be done		00000010 10000000 00000000 00000110
2068		addcc %r1,-4,%r1	10000010 10000000 01111111 11111100
2072		addcc %r1,%r2,%r4	10001000 10000000 01000000 00000010
2076		ld %r4,%r5	11001010 00000001 00000000 00000000
2080	ba loop		00010000 10111111 11111111 11111011
2084		addcc %r3,%r5,%r3	10000110 10000000 11000000 00000101
2088	done:	jmp %r15+4,%r0	10000001 11000011 11100000 00000100
2092	length:	20	00000000 00000000 00000000 00010100
2096	address:	a_start	00000000 00000000 00001011 10111000
		.org a_start	
3000		a:</tt	

Ilustración 13: ejemplo para el computador virtual (POCA)

```

ORG 0
Inicio
    bsf    STATUS,RP0
    clrf  PORTB
    movlw 0xFF
    movwf PORTA
    bcf   STATUS,RP0

Principal
    movf  PORTA,W
    movwf Contador
    movf  Contador,F
    btfsc STATUS,Z
    goto  PuntoDecimal
    sublw d'9'
    btfss STATUS,C

END
    
```

Ilustración 14: ejemplo para el microchip PIC16F84

3.5.1.1. Programa ensamblador

De define como un programa ensamblador [16] aquel que genera un código objeto que puede ser traducido nemónicamente por un lenguaje ensamblador en ‘*opcodes*’ y que sea capaz de resolver los nombres simbólicos para las localizaciones de memoria y otras entidades. Este lenguaje tiene como habitual el uso de referencias simbólicas ya que es una de sus propiedades más importantes. Dicho uso evita infinitos cálculos y diversas actualizaciones que se haría de modo manual de las direcciones después de cada modificación del programa. Otra de las herramientas de este potente programa, es la incorporación de macros para realizar sustituciones textuales.

Una de las ventajas que se puede mencionar en estos tipos de programas es la facilidad o la

simplicidad que presentan a la hora de escribir, en comparación con los compiladores para los lenguajes de nivel alto.

Los ensambladores modernos, especialmente para las arquitecturas basadas en RISC, tales como MIPS, Sun SPARC, y HP PA-RISC, así como también para el x86 (-64), optimiza la planificación de instrucciones para explotar la segmentación del CPU eficientemente.

En los compiladores para lenguajes de alto nivel, son el último paso antes de generar el código ejecutable.

3.5.2. Microcontrolador.

El termino Microcontrolador se define o se hace referencia a un circuito integrado programable o chip, capaz de ejecutar las órdenes grabadas o guardadas en su memoria. Todo Microcontrolador trae consigo en su interior las tres unidades fundamentales y funcionales de una computadora tales como: la unidad central de procesamiento (CPU), memoria y unidades de E/S [17].

Todos esos componentes citados, dan a entender que básicamente se trata de un computador completo integrado en único circuito pero con prestaciones limitadas y que se caracteriza principalmente por su alto nivel de especificaciones. Difieren de una CPU normal en el sentido de que es más fácil convertirlo en controlador en funcionamiento con un mínimo de chips externos de apoyo.

La idea fundamental de la tecnología desarrollada en estos pequeños chips, es que solo sea necesario conectar los dispositivos a una fuente de energía y de información necesaria con el fin de que un programa sea guardado en la memoria del Microcontrolador, es necesario que esté codificado en un sistema numérico hexadecimal, ya que con este sistema el Microcontrolador trabaja cuando está alimentado a una determinada tensión y está conectado al dispositivo analógico y/o discreto para su funcionamiento y que se quiera controlar.

Por otra parte las unidades de entrada y salida (E/S) suelen estar constituidos por puertos de 8 bits de longitud, de tal manera que permite leer datos del exterior. Pero ante todo lo dicho a priori el objetivo para el que están hechos los microcontroladores es que sean capaces de trabajar con dispositivos simples como: relés, LED, fotocélulas, sensores, motores, etc.

Haciendo un breve recorrido en la historia de estos pequeños chips, se puede decir que el primer microcontrolador **TMS 1000** fue creado en 1971 y comercializado en 1974, hecho por los ingenieros de Texas instruments (Gary Boone y Michael Cocharan). Combina memoria ROM, memoria RAM y microprocesador y reloj en un chip.

Dentro de grupo de la clasificaciones de los diferentes tipos de microprocesadores, los microprocesadores con memoria EPROM reprogramable son considerados más caros que la variantes PROM debido a que este solo se podía programar una vez. Para borrar la EPROM se necesita exponer a la luz ultravioleta la tapa de cuarzo transparente. Los microprocesadores con memoria EEPROM permiten que el borrado sea eléctrico y rápido sin la necesidad de un paquete costoso como en la EPROM.

Otros componentes que integrados en la arquitectura de los microcontroladores a parte de los ya mencionados pueden ser los siguientes:

- Temporizadores
- Convertidores A/D y D/A
- Perro guardián (WatchDog)
- Oscilador externo
- Comparador analógico
- Unidad lógica aritmética

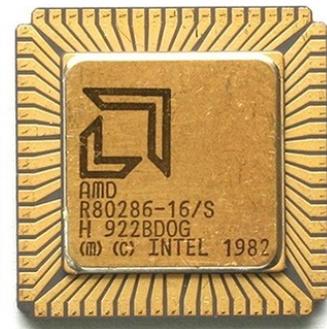
La arquitectura de un Microcontrolador puede ser RISC (reducido) o CISC (complejo), atendiendo a la cantidad de instrucciones que se quiera manejar y siendo definido este conjunto de instrucciones por el por el modelo y por el fabricante.



Ilustración 15:
microcontrolador RISC



Ilustración 16: Microcontrolador CISC



En la realización de este proyecto utilizaremos placas de Arduino, que incorpora que dependiendo de la placa que sea utilizara un tipo de microprocesador u otro. Más adelante se verán las diferentes placas y microprocesadores de Arduino.

3.5.3. Microprocesador.

El microprocesador conocido también como procesador, es un circuito central y más complejo de un sistema informático; a modo de ilustración, se le suele llamar por analogía el “*cerebro*” del computador. Arquitectónicamente hablando se puede decir que es un circuito integrado compuesto de millones de componentes electrónicos y constituye la unidad central de procesamiento de un PC apodado como microprocesador.

Este microprocesador es el encargado de ejecutar los programas, desde el sistema operativo hasta las aplicaciones de los usuarios; solo ejecuta instrucciones programadas en lenguajes de bajo nivel mediante la realización de operaciones aritméticas y lógicas simples tales como, sumar, restar, multiplicar, dividir, las lógicas binarias y accesos a memorias.

Estas unidades centrales de procesamiento esta constituidas esencialmente por los siguientes componentes:

- Registros
- Unidades de control
- Una unidad aritmética lógica (ALU)
- Unidad de cálculo de coma flotante

La conexión de este microprocesador se consigue con la ayuda de un zócalo específico de la placa base de la computadora; y para su correcto y estable funcionamiento, se le instala un sistema de refrigeración que consta de un disparador de calor fabricado con material de alta conductividad térmica como el cobre o aluminio; también lleva incorporado uno o más ventiladores que eliminan el exceso de calor absorbido por el disparador. Entre la capsula y el ventilador, usualmente se coloca pasta térmica para mejorar la conductividad del calor.

Existen otros métodos más eficaces como la refrigeración líquida o el uso de células peltier para la refrigeración externa, aunque estas técnicas se utilizan casi exclusivamente para aplicaciones especiales, tales como en las prácticas de *overclocking*

La medición del rendimiento de un microprocesador es una tarea realmente compleja, dado que existen diferentes tipos de “cargas” que pueden ser procesadas con diferente efectividad por procesadores de la misma gama. Una métrica del rendimiento es la frecuencia de reloj que permite comparar procesadores con núcleos de la misma familia, siendo este un indicador muy limitado dada la gran variedad de diseños con los cuales se comercializan los procesadores de una misma marca y referencia. Un sistema informático de alto rendimiento puede estar equipado con varios microprocesadores trabajando en paralelo y un microprocesador puede a su vez, estar constituido por varios núcleos físicos o lógicos. Un núcleo físico se refiere a una porción interna del microprocesador causi-independiente que realiza todas las actividades de una CPU solitaria, un núcleo lógico es la simulación de un núcleo físico de repartir de manera más eficiente el procesamiento. Existe una tendencia de integrar el mayor número de elementos dentro del propio procesador, aumentado así la eficiencia energética y la miniaturización. Entre los elementos integrados están las unidades de punto flotante, controladores de la memoria RAM, controladores de buses u procesadores dedicados de video.

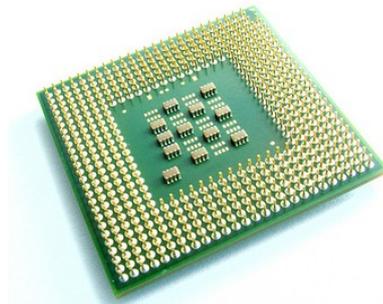


Ilustración 17: aspecto interno de un microprocesador

3.6. Sistema de Control

Un sistema de control [30] se define como todo conjunto de dispositivos encargados de administrar, dirigir, regular u ordenar el comportamiento de un determinado sistema, con el fin de reducir las probabilidades de fallos y obtener con eso los resultados deseados. En términos generales se utilizan sistemas de control en procesos de producción industrial para controlar ciertos equipos y maquinas. Para nuestro caso lo que se pretende regular es la velocidad de los automóviles una vez estén dentro del área de control.

Los sistemas de control pueden ser representados en forma de diagramas de bloques, en los cuales se proporciona una expresión visual y simplificada de la relación entre la entrada y la salida de un sistema físico y a cada componente integrado en el sistema de control se le denomina elemento, y suele estar representado mediante un rectángulo.

Dentro de la gran variedad de los diagramas de bloques se definen como los más sencillos o simples aquellos que constan de una sola entrada y de una sola salida y la interacción entre los bloques se representa por medio de flechas que indican el sentido de flujo de la información.

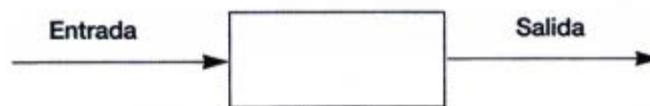


Ilustración 18: Diagrama de bloques simples

Dentro de los diagramas de bloques es posible realizar diversas operaciones algebraicas tales como la adición, sustracción, división y multiplicación. La forma de representar estas operaciones es la que se muestra en la siguiente *ilustración 17*

:

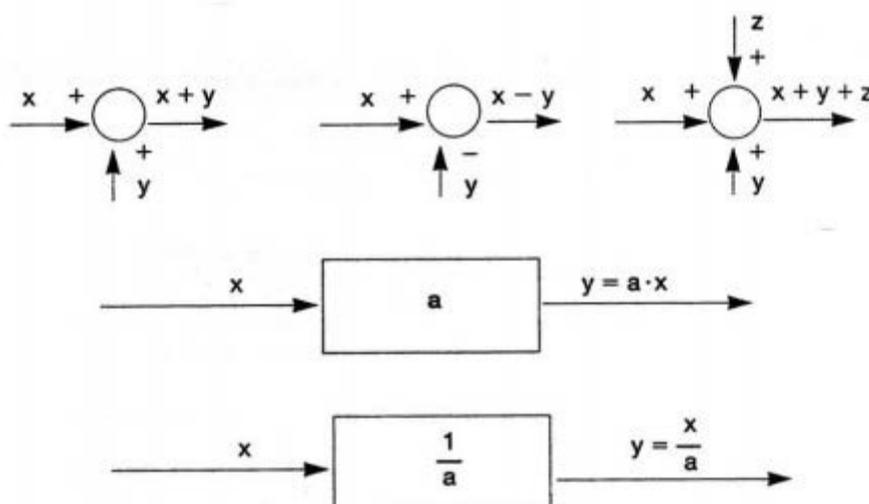


Ilustración 19: Bloque de representación de operaciones matemáticas

Un sistema de control produce una salida o respuesta para una entrada o un estímulo dado, como se muestra en la ilustración 16.

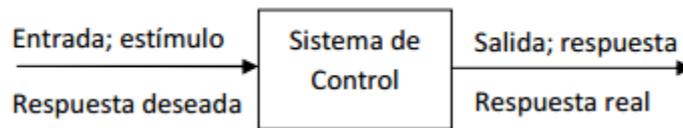


Ilustración 20: Representación de un sistema de control

La entrada representa una respuesta deseada y la salida representa una respuesta real. Dos de los principales factores que hacen que la salida difiera de la entrada son:

- El primero se conoce como respuesta transitoria, el cual está presente cuando existe un cambio instantáneo de la entrada con respecto al cambio gradual de la salida. Después de la respuesta transitoria, un sistema físico aproxima su respuesta en estado estable, en donde trata de aproximarse a la respuesta deseada.
- La precisión es el segundo factor que puede ocasionar que la salida sea diferente a la entrada. Esta diferencia es conocida como error en estado estable y en ocasiones depende de las características físicas del dispositivo

3.6.1. Clasificación de los sistemas de control

Existen dos clases comunes de sistemas de control, sistemas de **lazo abierto** y sistemas de **lazo cerrado**. En los sistemas de control de lazo abierto la salida se genera dependiendo de la entrada; mientras que en los sistemas de lazo cerrado la salida depende de las consideraciones y correcciones realizadas por la retroalimentación. Un sistema de lazo cerrado es llamado también sistema de control con realimentación. Los sistemas de control más modernos en ingeniería automatizan procesos sobre la base de muchos parámetros y reciben el nombre de controladores de automatización programables (PAC) [30]

Un ejemplo práctico es una lavadora automática: el remojo, el centrifugado y el lavado operan con una base de tiempo. La máquina no mide la señal de salida, la limpieza de la ropa, etc.



Ilustración 21: Esquema de control de lazo abierto

Un ejemplo de un sistema de control en lazo cerrado es el control de temperatura en una incubadora

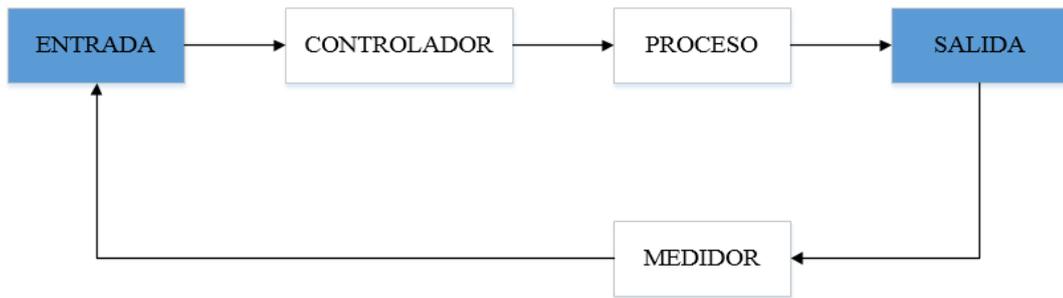


Ilustración 22: Esquema de control de lazo cerrado

3.6.2. Tipos de sistemas de control

Actualmente existen diferentes tipos de sistemas de control, estos pueden ser de acción todo-nada, de acción proporcional, acción integral, acción proporcional-integral, acción proporcional-derivativa y acción proporcional-integral-derivativa

3.6.2.1. Control Todo-Nada

Los controladores Todo-Nada se consideran como los sistemas los más básicos ya que estos envían una señal de activación siempre que en la entrada se detecte una señal menor que el valor de consigna asignado previamente. En casi contrario se desactivaría la salida.

Los controladores Todo-Nada son utilizados en termostatos de aire acondicionado. Estos activan el aire frío cuando la temperatura es mayor que el valor de consigna y la desactivan cuando la temperatura es menor que el valor de consigna [30].

3.6.2.2. Control proporcional (P)

Normalmente los sistemas de control proporcional son los sistemas con realimentación lineal. En la mayoría de los casos son utilizados para regular la velocidad de los automóviles

Los controlares proporcionales se caracterizan por tener una salida proporcional a la señal de error y ese error suele ser definido como la diferencia entre el valor real o medido y el valor de consigna o set point, es decir, la salida de un controlador proporcional es el resultado del producto entre la señal de error y la ganancia proporcional.

Esto puede ser expresado matemáticamente como.

$$P_{out} = K_p * e(t) \quad (1)$$

Ecuación 1: salida de un control proporcional (P)

Donde:

Pout: Salida del controlador proporcional.

Kp: Ganancia proporcional.

e(t): Error de proceso instantáneo en el tiempo t. $e(t) = SP - PV$.

SP: valor de consigna.

PV: valor del proceso.

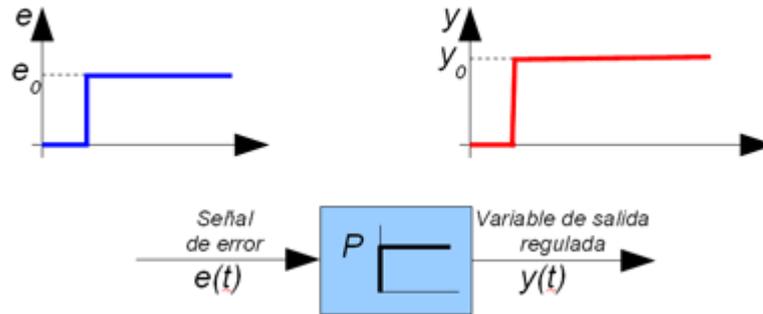


Ilustración 23: Respuesta temporal de un controlador proporcional

En general los controladores proporcionales siempre presentan una respuesta con un cierto error remanente que el sistema es incapaz de compensar. [30].

3.6.3. Control Integral

En estos controladores el valor de la acción de control es proporcional a la integral de la señal de error, por lo que en este tipo de control la acción varía en función de la desviación de la salida y del tiempo en el que se mantiene esta desviación.

Si consideramos que:

$y(t)$ = salida integral.

$e(t)$ = error.

T_i = tiempo integral.

La salida proporcionada por este controlador es:

$$y(t) = \frac{1}{T_i} \int e(t) dt \quad (2)$$

Ecuación 2: salida de un control integral (I)

Que en el dominio de Laplace, se expresaría como:

$$Y(s) = \frac{1}{T_i s} E(s) \quad (3)$$

Ecuación 3: salida de un control integral en Laplace

Por lo que la función de transferencia será:

$$G(s) = \frac{Y(s)}{E(s)} = \frac{1}{Tis} \quad (4)$$

Ecuación 4: Función de transferencia de un control Integral en Laplace

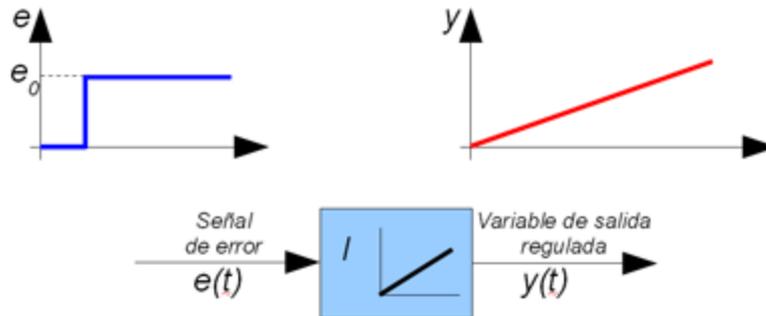


Ilustración 24: Respuesta temporal de un controlador integral.

Los controladores proporcionan el inconveniente de que la respuesta inicial es muy lenta y por lo tanto, este no empieza a ser efectivo hasta haber transcurrido un determinado periodo de tiempo pero que en cambio anula los posibles errores remanentes que puede proporcionar dicho controlador [30].

3.6.4. Control proporcional-integral (PI)

Normalmente no existen controladores que actúen exclusivamente con acción integral, por lo tanto para que estos puedan ser utilizados deben actuar en combinación con el regulador de acción proporcional de manera instantánea mientras que el regulador integral actúa durante un periodo de tiempo (T_i =tiempo integral)

La función de transferencia del bloque de control PI responde a la ecuación:

$$G(s) = \frac{Y(s)}{E(s)} = K_p \left(\frac{1}{Tis} + 1 \right) \quad (5)$$

Ecuación 5: función de transferencia de un control PI en Laplace

Donde K_p y T_i , son parámetros modificables de acuerdo a las necesidades que requiere el sistema. En el caso de que T_i fuera grande, la pendiente de la rampa correspondiente a la acción rampa será pequeña y en el caso de que la T_i fuera pequeño, la acción correspondiente la acción rampa sería pequeña grande.

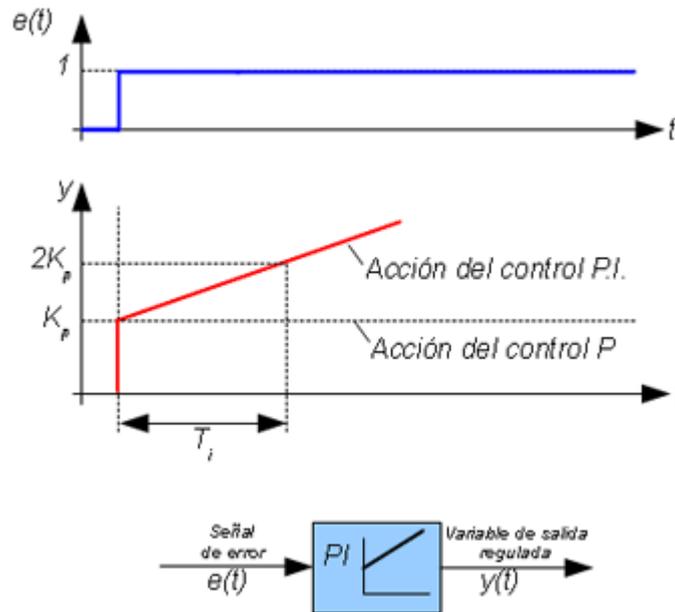


Ilustración 25: Respuesta temporal de un control PI.

Por lo tanto la respuesta de un regulador PI, será la suma de las respuestas de un controlado proporcional y de un controlador integral, el regulador proporcional detectara la señal de error y el regulador integral anulara esa señal de error [30].

3.6.5. Control proporcional-Derivativo (PD)

El controlador derivativo se opone a desviaciones de la señal de entrada, con una respuesta que es proporcional a la rapidez con que se producen estas.

Si consideramos que:

$y(t)$ = salida diferencial.

$e(t)$ = Error.

T_d = Tiempo diferencial, se usa para dar mayor o menor trascendencia a la acción derivativa.

La salida de este regulador es:

$$y(t) = K_p \cdot e(t) + K_p \cdot T_d \frac{de(t)}{dt} \quad (6)$$

Ecuación 6: salida de un control PD

Que en el dominio de Laplace será:

$$Y(s) = K_p E(s) + K_p T_d s Y(s) \quad (7)$$

Ecuación 7: salida de un controlador PD en Laplace

Por lo que su función de transferencia será:

$$G(s) = \frac{Y(s)}{E(s)} = K_p(1 + sT_d) \quad (8)$$

Ecuación 8: Función de transferencia de un control PD

Al igual que pasaba con un regulador integrador, en un regulador diferencial tampoco actúa de forma individual, sino que siempre debe estar asociada a la actuación de un regulador proporcional.

En caso de que se le pudieran aplicar modificaciones instantáneas a la entrada de un regulador diferencial, implicaría que la variación de la velocidad será muy elevada y por lo tanto, la respuesta del regulador diferencial adoptará un comportamiento brusco, razón por la cual no se aconsejaría su utilización

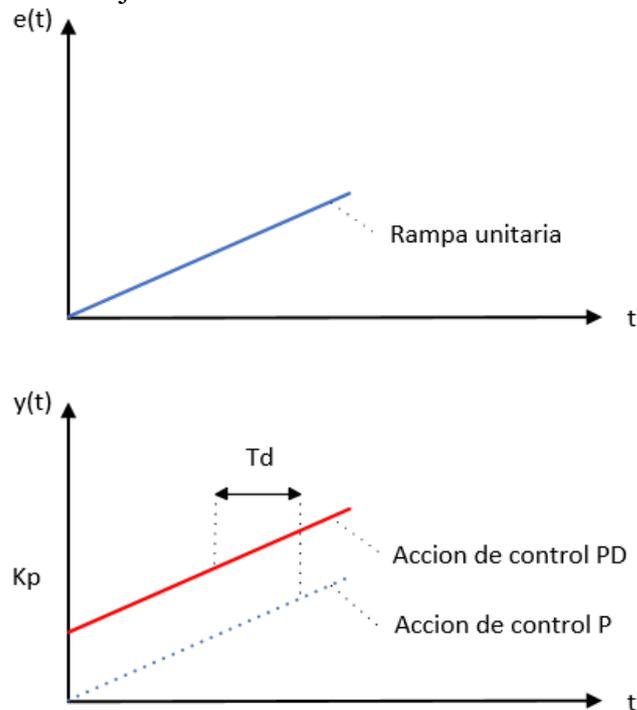


Ilustración 26: Respuesta temporal de un control PD.

En los controladores diferenciales, al ser la derivada de una constante igual a cero, el control derivativo no ejerce ningún efecto, siendo únicamente práctico en aquellos casos en los que la señal de error varía en el tiempo de forma continua.

Por lo que, el análisis de este controlador ante una señal de error tipo escalón no tiene sentido, por ello representamos la salida del controlador en respuesta a una señal de entrada en forma de rampa unitaria.

En la anterior ilustración se observa la respuesta que ofrece el controlador, que se anticipa a la propia señal de error. Este tipo de controlador se utiliza en sistemas que deben actuar muy rápidamente, ofreciendo una respuesta tal que provoca que la salida continuamente este cambiando de valor.

El regulador derivativo no se emplea aisladamente, ya que para señales lentas, el error producido en salida en régimen permanente sería muy grande y si la señal de mando dejase de actuar durante un tiempo largo la salida tendería hacia cero, con lo que no se realizaría ninguna acción de control.

La ventaja de este tipo de controlador es que aumenta la velocidad de respuesta del sistema de control.

Al actuar conjuntamente con un controlador proporcional, las características de un controlador derivativo provocan una apreciable mejora de la velocidad de respuesta del sistema, aunque pierde precisión en la salida durante el tiempo de funcionamiento del control derivativo [30].

3.6.6. Control proporcional-Integral Derivativo (PID)

Un regulador o controlador PID (proporcional, integral, derivativo) es un mecanismo de control retroalimentado en bucle cerrado ampliamente usado en sistemas de control industrial. Un PID calcula un error como la diferencia entre el valor actual del sistema y el valor al que se desea llegar. Además intenta minimizar el error mediante el ajuste de las entradas del proceso.

Es un sistema de regulación que trata de aprovechar las ventajas de cada uno de los controladores de acciones básicas, de manera que si la señal de error varía lentamente en el tiempo, predomina la acción proporcional e integral, mientras que si la señal de error varía rápidamente, predomina la acción derivativa. Tiene la ventaja de ofrecer una respuesta muy rápida y una compensación de la señal de error inmediata en el caso de perturbaciones. Presenta el inconveniente de que este sistema es muy propenso a oscilar y los ajustes de los parámetros son mucho más difíciles de realizar.

En el cálculo del regulador PID intervienen tres parámetros distintos. Estos valores se pueden interpretar en función del tiempo. El proporcional P depende del error actual, el integral I depende de la suma de todos los errores pasados, y el derivativo D es la predicción de errores futuros, basándose en la tasa de cambio actual. La suma ponderada de los tres términos permiten ajustar un proceso mediante un elemento de control como por ejemplo la velocidad que debe alcanzar o circular un automóvil en un periodo de tiempo determinado [30] [31].

$$U = P + I + D \quad (9)$$

Ecuación 9: suma de las tres acciones de un control PID

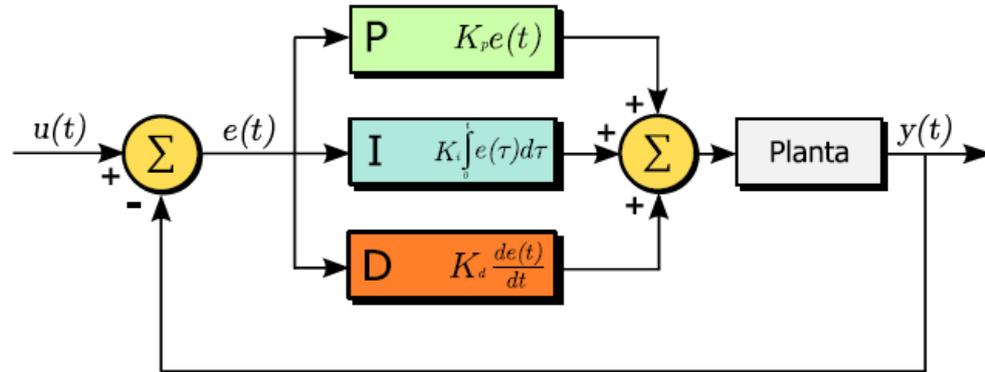


Ilustración 27: Diagrama de bloques de un regulador PID

La salida del regulador viene dada por la siguiente ecuación:

$$y(t) = Kp e(t) + \frac{Kp}{Ti} \int_0^t e(\tau) d\tau + Kp Td \frac{de(t)}{dt} \quad (10)$$

Ecuación 10: salida de un control PID

Que en el dominio de Laplace, será:

$$Y(s) = Kp E(s) + \frac{Kp}{Tis} E(s) + Kp Tds E(s) \quad (11)$$

Ecuación 11: salida de un control PID en Laplace

Y por tanto la función de transferencia del bloque de control PID será:

$$\frac{Y(s)}{E(s)} = Kp \left(1 + \frac{1}{Tis} + Tds \right) \quad (12)$$

Ecuación 12: función de transferencia de un control PID

Donde Kp , Ti y Td son parámetros ajustables del sistema.

La respuesta temporal de un regulador PID sería la mostrada en la ilustración siguiente:

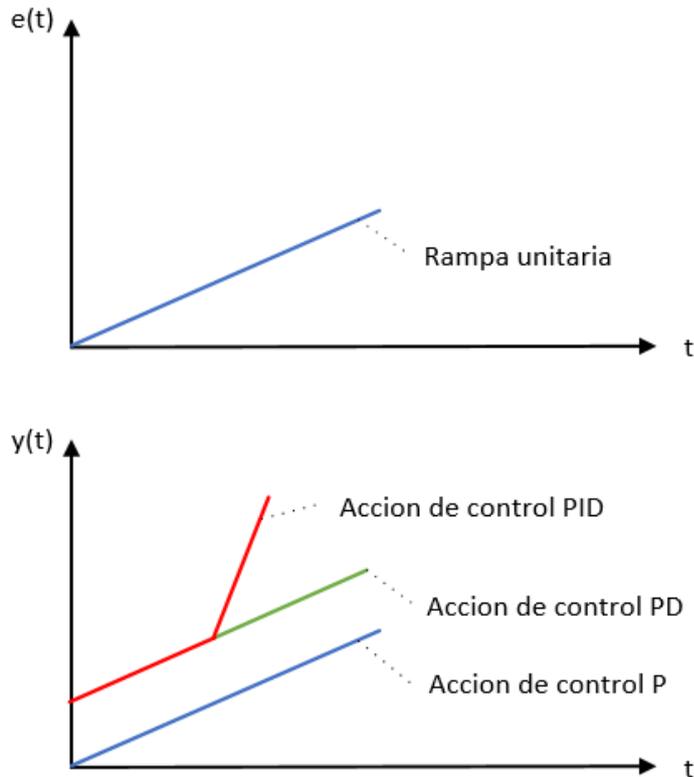


Ilustración 28: Respuesta temporal de un controlador PID

3.7. Sensores

Los dispositivos denominados sensores, son aquellos capaces de detectar magnitudes químicas y físicas conocidas como **variables de instrumentación** y las transforman en **variables eléctricas**, es decir, miden las características o cambios de estado del ambiente que los rodea.

Las variables de instrumentación pueden ser por ejemplo: temperatura, humedad, distancia, intensidad lumínica, aceleración, inclinación, desplazamiento, torsión, fuerza, pH, etc. Las variables eléctricas a diferencia de las variables de instrumentación, puede ser: una resistencia eléctrica (como en una RTD), una capacidad eléctrica (como un sensor de humedad), una corriente eléctrica (como una fotorresistencia), una tensión eléctrica (como un termopar), etc.

La precisión y la rapidez del sensor son aspectos que dependen de la utilización de dicho sensor, de tal manera que, hay veces que se necesita que sean más precisos y otras veces interesa más que su precisión a la hora de tomar datos o proporcionar alguna información. Ante estos dos puntos, también es muy importante tener en cuenta el tipo de controlador que en algunos casos se tiene que intercalar otro circuito entre el sensor y el controlador con el fin de que se pueda adaptar las señales del sensor a la salida o entrada del controlador ya que a veces no suelen estar en la misma magnitud.

Los sensores a diferencia de los transductores es que los sensores, están siempre en contacto con las variables de instrumentación, por lo tanto, también se puede decir que son dispositivos que aprovechan una de sus propiedades con el fin de adaptar las señales que miden para que las puedan interpretar otros dispositivos. Un ejemplo sería: *el termómetro*

que aprovecha las propiedades que tiene el mercurio de dilatarse o contraerse dependiendo de la temperatura.

Otra definición que se les daría a los sensores sería que, los consideran como dispositivos que transforman una determinada clase de energía en otra con propiedades diferentes y propias de dicha energía. Normalmente los sensores se clasifican en **digitales** y **analógicos**.

Se llaman analógicos a aquellos que proporcionan valores dentro de un rango determinado y continuo (por ejemplo: la fotorresistencia que mide la intensidad de la luz); los sensores digitales, son aquellos que entregan una señal discreta dentro de un conjunto de posibles valores (por ejemplo: una fotocélula que entrega 0 si detecta y un 1 si no detecta).

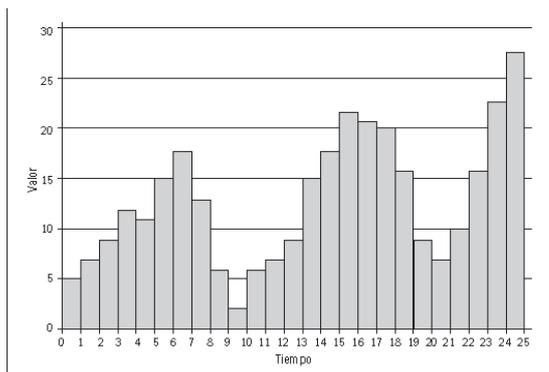


Ilustración 29: gráfico analógico de un sensor

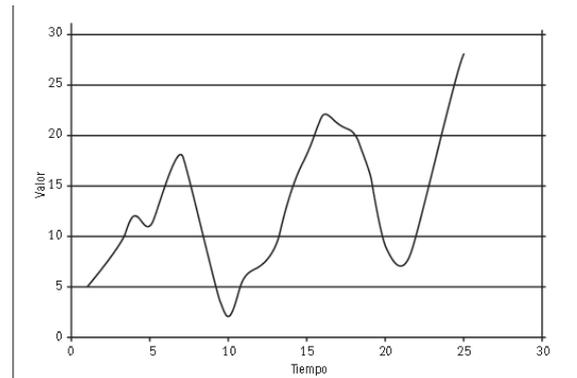


Ilustración 30: gráfico digital de un sensor

Otra de las clasificaciones que se pueden dar en los sensores son: **externos** e **internos**, **pasivos** y **activos**. Los sensores internos son los que proporcionan información al dispositivo en que está instalado, mientras que los externos son los que proporcionan datos del medio que les rodea. Los sensores activos son los que están preparados para enviar señales al ambiente para luego recibir la información en el rebote.

La utilización de los sensores actualmente ocupa casi todo el ámbito del diseño, ya sea en: industria automotriz, domótica, robótica, industrias aeroespaciales, industria de manufactura, medicina, etc.

3.7.1. Características de un sensor

- **Rango de medida:** dominio de una magnitud de medida en el que puede aplicarse el sensor.
- **Precisión:** es el error de medida máximo esperado.
- **Offset o derivación de cero:** valor de la variable de salida cuando la variable de entrada es nula.
- **Linealidad o correlación lineal.**
- **Sensibilidad:** suponiendo que es de entrada a salida y la variación de la magnitud es la entrada.

- **Resolución:** mínima variación de la magnitud en la entrada que puede apreciarse en la salida.
- **Rapidez de respuesta:** puede ser un tiempo fijo o depender de cuanto varíe la magnitud a medir.
- **Derivadas:** son otras magnitudes, aparte de la medida como magnitud de entrada, que influyen en la variable de salida.
- **Repetitividad:** error esperado al repetir varias veces la misma medida.

3.7.2. Tipos de sensores

Debido al gran número de sensores que existen actualmente, a continuación se elaborará una lista de casi todos y otra parte se definirá los más utilizados por los diseñadores.

- **Sensores infrarrojos:** su función consiste en enviar señales luminosas infrarrojas a un fotoreceptor y dependiendo del tipo de sensor y de su potencia puede ser utilizados para medir distancias o para la seguridad.
- **Sensores de luz o brillo:** su función esencial es la detección de la luz, aunque los hay que incluso pueden distinguir los colores. Su principal aplicación se da en instalaciones domóticas o enmóticas.
- **Sensores de posición:** estos sensores están hechos para identificar la posición del objeto al que van instalado, tal objetividad se podría conseguir con la ayuda de un potenciómetro, de modo que al variar su valor de resistencia se puede determinar su posición actual.
- **Sensores efecto hall:** tienen la finalidad de detectar la presencia de metales, gracias a la variación de la conductividad que se registrar al acercarse a un objeto metálico.
- **Sensores para la utilización de captación de vídeos:** consiste en la incorporación de cámaras en los dispositivos tales como los semáforos entre otros, ya que proporciona mayor información.
- **Sensores detectores de humo:** Un detector de humo es una alarma que detecta la presencia de humo en el aire y emite una señal acústica avisando del peligro de incendio. Atendiendo al método de detección que usan pueden ser de varios tipos: - **Detectores iónicos:** Utilizados para la detección de gases y humos de combustión que no son visibles a simple vista. - **Detectores ópticos:** Detectan los humos visibles mediante la absorción o difusión de la luz.
- **Sensores detectores de movimiento:** son aquellos que responden a la detección de un movimiento físico. En ese proyecto la incorporación de este tipo de sensor es una de las ideas fundamental con el fin de detectar la presencia de peatones el medio de la calzada en caso de que ya no tengan tiempo suficiente para terminar de cruzar la calzada.

- **Sensores laser:** tiene la misma función que los sensores infrarrojos pero con la ventaja de que emiten una luz visible

Haciendo un recorrido general sobre los sensores obtendremos los siguientes:

Tabla 2: sensores con magnitudes y características

Magnitud	Traductor	Característica
Posición línea o angular	Potenciómetro	Analógica
	Encoder	Digital
	Hall	Digital
Desplazamiento y deformación	Transformador diferencial	Analógica
	Galga extensiométrica	Analógica
	Magnetostrictivos	A/D
	Magnetorestrictivos	Analógica
Velocidad lineal y angular	LVDT	Analógica
	Dinamo tacométrica	Analógica
	Encoder	Digital
	Detector inductivo	Digital
	Servo-inclinómetros	A/D
	RVDT	Analógica
Aceleración	Giroscopio	Analógico
	Acelerómetro	Analógico
Fuerza y par	Servo-acelerómetro	Analógico
	Galga- extensiométrica	analógica
Presión	Triaxiales	A/D
	Membranas	Analógicas
	Piezoeléctricos	Analógicos
Caudal	Manómetros digitales	Digital
	Turbina	Analógica
Temperatura	Magnético	Analógica
	Termopar	Analógica
	RTD	Analógica
	Termistor NTC	Analógica
	Termistor PTC	Analógica
	Bimetal-termostato	I/O

Tabla 3: sensores con magnitudes y características

Magnitud	Traductor	Característica
Sensores de presencia	Inductivo	I/O
	Capacitivo	I/O
	Óptico	I/O
Sensores táctiles	Matriz de contacto	I/O y analógico
	Piel artificial	Analógico
Visión artificial	Cámara de video	Procesamiento img

	Cámara CCD o CMOS	Procesamiento img
Sensor de proximidad	Sensor final de carrera	
	Sensor capacitivo	
	Fotoinductivo	Analógica
	Fotoeléctrico	Analógica
Sensor acústico	Micrófono	Analógico
Sensor de acidez	IsFET	
Sensor de luz	fotoresistencia	Analógica
	fotodiodo	Analógica
	fototransistor	Analógica
	Células fotelectricas	Analógica
Sensor de captura de movimiento	Sensores inerciales	

3.8. Controladores programables-Arduino

Arduino [26] es una plataforma de hardware libre, basada en una placa con un Microcontrolador y un entorno de desarrollo, diseñada para facilitar el uso de la electrónica en proyectos multidisciplinarios. La creación de esta plataforma fue para aficionados, artistas en crear entornos u objetos interactivos.

El hardware consiste en una placa con un microcontrolador Atmel AVR y puertos de entrada y salida. Los microcontroladores más usados son los Atmega168, Atmega328, Atmega1280 Atmega8, por su sencillez y bajo coste permite el desarrollo de múltiples diseños. Por otra parte, el microcontrolador incorpora un software que consiste en un entorno de desarrollo que implementa el lenguaje de programación *processing/wiring* y el cargador de arranque que corre en la placa.

Arduino se utiliza para desarrollar objetos interactivos autónomos o puede ser conectado a un software del ordenador. Las placas se puede desarrollar a mano o adquirirse y su entorno de desarrollo programable se puede descargar gratuitamente. Al ser un open-hardware, tanto su diseño como su distribución son libres. Es decir, son utilizables para cualquier sin la necesidad de adquirir alguna licencia.

Arduino recibe información del entorno a través de sus pines de entrada, mediante una gama amplia de sensores disponibles, pudiendo controlar el entorno que nos rodea, como puede ser motores, servos, luces, etc. Para trabajar con Arduino se necesita tener conocimientos de electrónica y programación. Los proyectos hechos con Arduino se pueden ejecutar sin necesidad de conectarlos a un PC.

Desde octubre de 2012, Arduino se utiliza también con microcontroladoras CortexM3 de ARM de 32 bits, que coexistirán con las más limitadas y más económicas AVR de 8 bits. Tanto ARM como AVR no son plataformas compatibles con el sistema binario. Pero se

pueden programar en el mismo IDE de Arduino y hacerse programas que compilen sin cambios en las dos plataformas. A diferencia de las microcontroladoras ARM que usan una tensión de 3.3V las mayorías de las placas AVR usan 5V, aunque hace tiempo atrás se lanzaron Atmel AVR que usaban 3.3V y actualmente ya hay versiones clonadas.

3.8.1. Lenguajes de programación para Arduino

La plataforma Arduino, se programa mediante el uso de un lenguaje propio basado en el lenguaje programable de alto nivel *processing*. Pero sin embargo, también es posible usar otros lenguajes de programación y aplicaciones populares en Arduino, tales como los citados a continuación.

- Java
- Flash: mediante ActionScript
- Processing
- Pure Data
- MaxMSP: entorno grafico de programación para aplicaciones musicales, audio, multimedia
- VVVV: síntesis de videos en tiempo real
- Adobe Director
- PyThon
- Ruby
- C
- C++: mediante libSerial o en Windows
- C#
- Cocoa/Objective-C : para Mac OS X
- Linux TTY: para terminales de Linux
- 3DVDIA Virtools: para aplicaciones interactivas y del tiempo real
- SuperCollider: para síntesis de audio en tiempo real
- Instant Reality : para X3D
- Liberlab: software de medición y experimentaciones
- BlitzMax: con acceso restringido
- Squeak: implemetacion libre de SmalTalk
- Mathematica
- Matlab
- Minobloq: entorno gráfico de programación y corre también en OLPC
- Isadora: interactividad audiovisual en tiempo real
- Perl
- Physical Etoy: entorno gráfico de programación para proyectos de robótica educativa
- Scratch for Arduino S4A: entorno gráfico de programación, modifcaicion del entorno para niños Scratch del MIT
- Visual Basic: NET
- VBScript
- Gambas
- Php

Esto es posible debido a que Arduino se comunica a través de transmisión de datos en formato serie y es algo que casi todos los lenguajes citados soportan. Si existiera alguno que no se comunicara de forma directa, hay diversos software que actúan de intermediario y traducen los mensajes enviados por ambas partes para permitir una comunicación fluida.

✚ **Nota:** Para este proyecto se haría uso del entorno de programación C ya que es el lenguaje que viene integrado dentro del microcontrolador que se pretende en dicho proyecto.

3.8.2. Placas de Arduino

Actualmente existen diversas placas de Arduino, dependiendo de la cantidad de entradas y salidas que se necesita tener. Cada modelo de placa tiene un procesador con características propias según para lo que se quiere utilizar. De modo que, a la hora de elegir una placa de Arduino es necesario tener claro las funciones que queremos realizar. En los siguientes puntos se va a detallar las diferentes placas de Arduino que se pueden encontrar

3.8.2.1. Arduino uno

Este modelo, está basado en ATmega328. Tiene 14 pines digitales de entrada-salida (de las cuales 6 se pueden utilizar como salidas PWM), 6 entradas analógicas. Lleva incluido una conexión USB y un conector de alimentación y conector ICSP. Se puede conectar al PC mediante un cable USB. Esta placa se diferencia de las otras porque no utiliza el chip controlador FTDI USB-to-serial, en cambio tiene el ATMEGA16U2 programado como convertidor USB a serie.

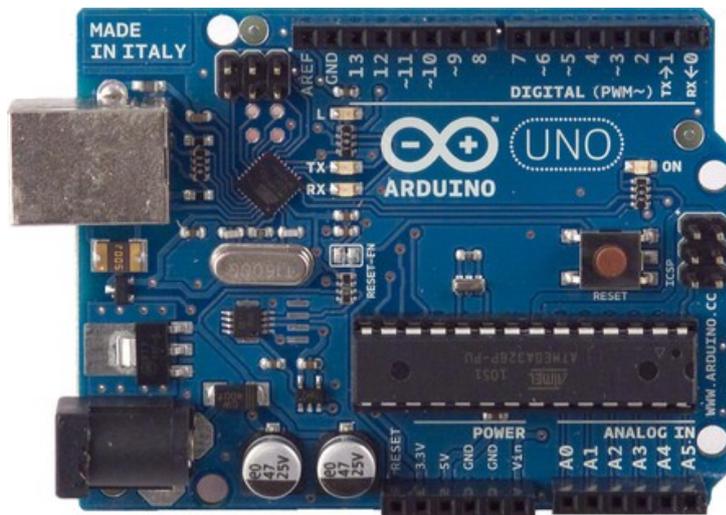


Ilustración 31: placa de Arduino uno

Características más importantes de la placa

- Microcontrolador ATmega328, la placa funciona con 5 V.
- La tensión recomendada de entrada 7-12 V, siendo la

- Tensión de entrada límites de 6-20 V.
- Corriente de la C.C. por el pin de entrada-salida 40 mA..
- Corriente de la C.C. para el pin 3.3 V 50 mA..
- Memoria flash de 32 Kb.
- SRAM 2 Kb y EEPROM 1 Kb.

3.8.2.2. Arduino Due

Está basado en Atmel SAM3X8E CPU Tiene 54 pernos digitales de entrada-salida (de las cuales 12 se pueden utilizar como salidas PWM), 12 entradas analógicas. 4 UARTs (puertos serie de hardware) 2 salidas DAC (convertidores de analógico a digital) Contiene 2 conectores USB, un conector de alimentación, un conector ICSP, un conector JTAG En uno de los USB se puede conectar otros dispositivos USB a la tarjeta (ratones, teclados,etc)

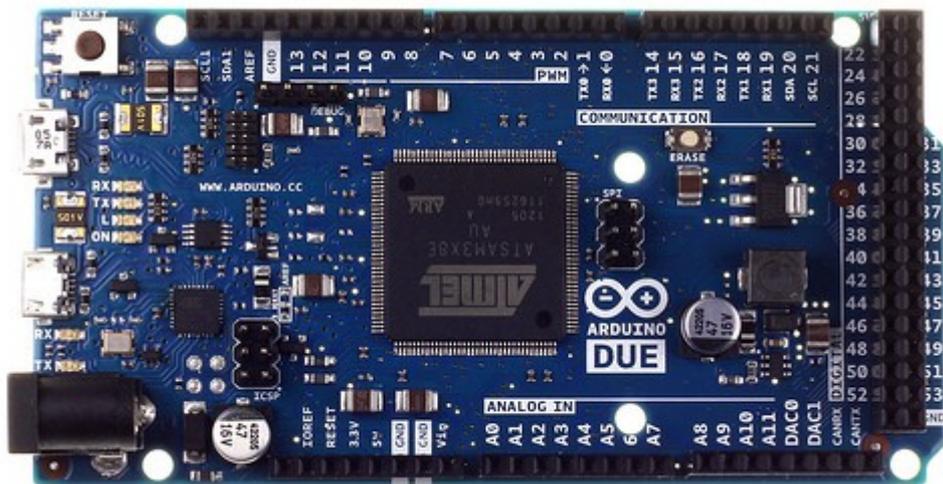


Ilustración 32: Placa Arduino Due

Características más importantes de la placa:

- Microcontrolador AT91SAM3X8E, la placa funciona con 3.3 V.
- La tensión recomendada de entrada 7-12 V, siendo la tensión de entrada límites de 6- 20 V.
- Corriente máxima para todas entradas-salidas 130 mA.
- La Corriente máxima de los pines de Entrada/Salida de 3.3 V 800 mA.
- La Corriente máxima de los pines de Entrada/Salida de 5 V 800 mA.
- Max. 5 V en un pin de entrada.
- Memoria flash de 32 Kb.
- SRAM 96 Kb y velocidad de reloj 84 MHz.

3.8.2.3. Arduino ADK

Está basado en ATmega 2560. Lleva una USB para conectar los teléfonos que están basados en Android. Tiene 54 pines digitales de entrada-salida (de las cuales 14 se pueden uti-

lizar como salidas PWM), 16 entradas analógicas. 4 UARTs (puertos serie de hardware) Un oscilador de cristal de 16 MHz Contiene una conexión USB, un conector de alimentación, un conector ICSP Lleva un circuito USB que permite comunicarse con los dispositivos USB y suministrarle alimentación.



Ilustración 33: Arduino ADK

Características más importantes de la placa:

- Microcontrolador ATmega2560, la placa funciona con 5 V.
- La tensión recomendada de entrada 7-12 V, siendo la tensión de entrada limitada de 6- 20 V.
- Corriente máxima para todas entradas-salidas 40 mA.
- La Corriente máxima de los pines de Entrada/Salida de 3.3 V 50 mA.
- Memoria flash de 256 Kb
- SRAM 8 Kb, EEPROM 4 Kb y velocidad de reloj 16 MHz.

3.8.2.4. Arduino Mega2560

Este modelo ATmega2560 consta de 54 pines digitales de entrada-salida (de las cuales 14 se pueden utilizar como salidas PWM), 16 entradas analógicas. 4 UARTs (puertos serie de hardware) Un oscilador de cristal de 16 MHz Contiene una conexión USB, un conector de alimentación, un conector ICSP.



Ilustración 34: Placa Arduino Mega 2560

Características más importantes de la placa:

- Microcontrolador ATmega2560, la placa funciona con 5 V.
- La tensión recomendada de entrada 7-12 V, siendo la tensión de entrada límites de 6-20 V.
- Corriente máxima para todas entradas-salidas 40 mA.
- La Corriente máxima de los pines de Entrada/Salida de 3.3 V 50 mA.
- Memoria flash de 256 Kb
- SRAM 8 Kb, EEPROM 4 Kb y velocidad de reloj 16 MHz

✚ **Nota:** Para la realización de este proyecto se haría uso de esta placa de Arduino y con ella intentar controlar todas las funciones para cuales está pensado.

3.8.2.5. Arduino Nano

El Arduino Nano, está basado en ATmega 328. Normalmente no tiene toma de alimentación y se programa a través de un cable mini-USB. Consta de 14 pines digitales de entrada-salida (de las cuales 6 se pueden utilizar como salidas PWM), 8 entradas analógicas

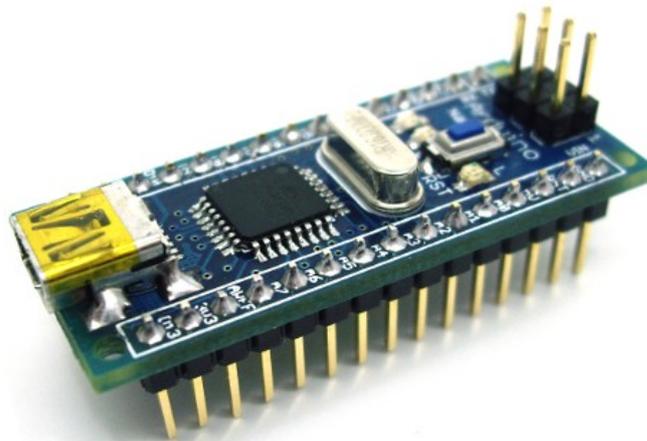


Ilustración 35: Placa Arduino nano

Características más importantes de la placa:

- Microcontrolador ATmega328, la placa funciona con 5 V.
- La tensión recomendada de entrada 7-12 V, siendo la tensión de entrada límites de 6-20 V.
- Corriente máxima para todas entradas-salidas 40 mA.
- Memoria flash de 32 Kb.
- SRAM 2 Kb, EEPROM 512 bytes y velocidad de reloj 16 MHz.

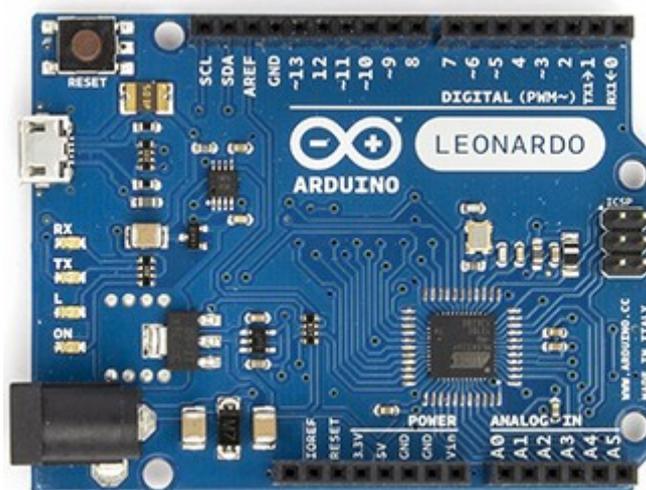
3.8.2.6. Arduino Leonardo

Ilustración 36: Placa Arduino Leonardo

Está basado en ATmega 32u4. Tiene 20 pines digitales de entrada-salida (de las cuales 7 se pueden utilizar como salidas PWM), 12 entradas analógicas. Un oscilador de cristal de 16 MHz. Contiene una conexión USB, un conector de alimentación, un conector ICSP. Esta placa es diferente a todas las demás, porque incorpora en su interior la parte de comunicación USB, y no necesita un microprocesador secundario, por lo tanto cuando se conecta al ordenador se reconoce como un ratón o un teclado.

Características más importantes de la placa:

- Microcontrolador ATmega32u4, la placa funciona con 5 V.
- La tensión recomendada de entrada 7-12 V, siendo la tensión de entrada límites de 6-20 V.
- Corriente máxima para todas entradas-salidas 40 mA.
- La Corriente máxima de los pines de Entrada/Salida de 3.3 V 50 mA.
- Memoria flash de 32 Kb.
- SRAM 2.5 Kb, EEPROM 1 Kb y velocidad de reloj 16 MHz.

3.8.2.7. Arduino YUN

Es la combinación de un Arduino Leonardo, con un chip Wifi que utiliza Linino (un MIPS GNU/Linux basada en OpenWRT). Incorpora Linux en la PCB de la placa Arduino Leonardo y están conectados los dos para ejecutar comandos en Linux, y se utiliza como una interfaz Ethernet y Wifi. Está basado en ATmega 32u4 Tiene 20 pines digitales de entrada-salida (de las cuales 7 se pueden utilizar como salidas PWM), 12 entradas analógicas. Un oscilador de cristal de 16 MHz Contiene una conexión USB, un conector de alimentación, un conector ICSP Esta placa es diferente a todas las demás, porque incorpora en su interior la parte de comunicación USB, y no necesita un microprocesador secundario, por lo tanto cuando se conecta al ordenador se reconoce como un ratón o un teclado.

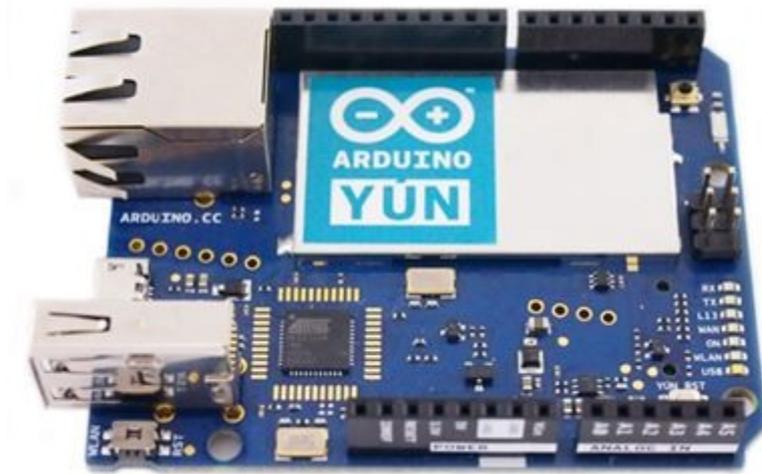


Ilustración 37: Placa Arduino YUN

Características más importantes de la placa:

- Microcontrolador ATmega32u4, la placa funciona con 5 V.
- La tensión recomendada de entrada 7-12 V, siendo la tensión de entrada límites de 6-20 V.
- Corriente máxima para todas entradas-salidas 40 mA.
- La Corriente máxima de los pines de Entrada/Salida de 3.3 V 50 mA.
- Memoria flash de 32 Kb
- SRAM 2.5 Kb, EEPROM 1 Kb y velocidad de reloj 16 MHz

3.8.2.8. Arduino Micro

La placa del Arduino Micro, es similar que placa Arduino Leonardo, pero en formato más pequeña. Está basado en ATmega32u4 Tiene 20 pines digitales de entrada-salida (de las cuales 7 se pueden utilizar como salidas PWM), 12 entradas analógicas. Un oscilador de

cristal de 16 MHz. Contiene una conexión micro USB, un conector ICSP. Esta placa es diferente a todas las demás, porque incorpora en su interior la parte de comunicación USB, y no necesita un microprocesador secundario, por lo tanto cuando se conecta al ordenador se reconoce como un ratón o un teclado.

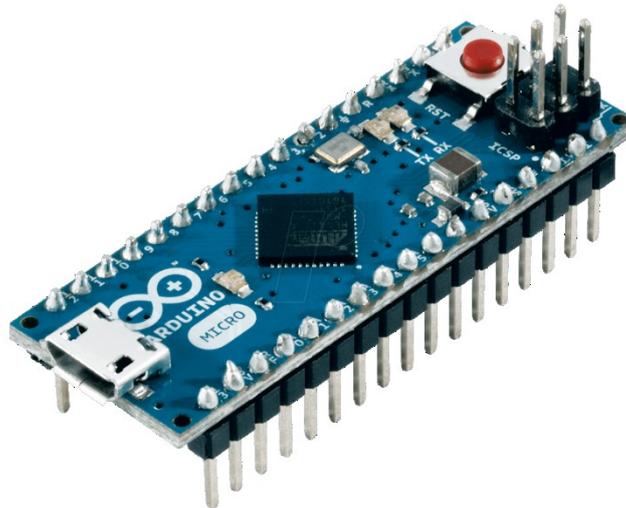


Ilustración 38: Placa Arduino Micro

Características más importantes de la placa:

- Microcontrolador ATmega32u4, la placa funciona con 5 V.
- La tensión recomendada de entrada 7-12 V, siendo la tensión de entrada límites de 6-20 V.
- Corriente máxima para todas entradas-salidas 40 mA.
- La Corriente máxima de los pines de Entrada/Salida de 3.3 V 50 mA.
- Memoria flash de 32 Kb.
- SRAM 2.5 Kb, EEPROM 1 Kb y velocidad de reloj 16 MHz.

3.8.2.9. Arduino Ethernet W5100 escudo

La importancia de esta placa, es que puede conectar Arduino a internet. Se basa en el chip W5100 WIZnet Ethernet, que proporciona una red (IP) de pila capaz de TCP y UDP, que soporta hasta conexiones de socket simultáneas. Se puede añadir una ranura de tarjeta micro-SD, para almacenar archivos para servir a través de la red. Se adapta a todas las versiones de Arduino tarjeta principal de 2009, la ONU, mega 1280, Mega 2560. Tiene 14 entradas/salidas digitales (de las cuales 4 se pueden utilizar como PWM) y 6 entradas analógicas. Tiene reservado desde 10 hasta 13 pines por SPI, 4 utilizado por la placa SD, y 2 desde W5100.

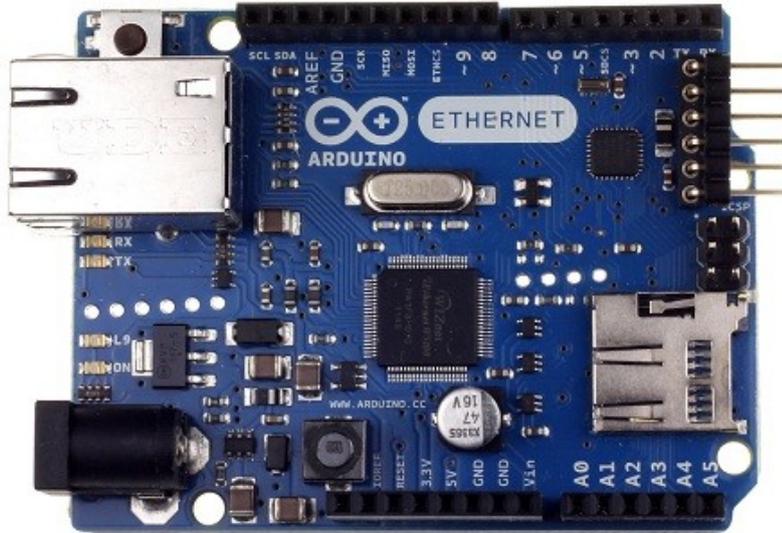


Ilustración 39: Placa Arduino Ethernet

Características más importantes de la placa:

- Microcontrolador ATmega328, la placa funciona con 5 V.
- La tensión recomendada de entrada 7-12 V, siendo la tensión de entrada límites de 6-20 V.
- Corriente máxima para todas entradas-salidas 40 mA.
- La Corriente máxima de los pines de Entrada/Salida de 3.3 V 50 mA.
- Memoria flash de 32 Kb.
- SRAM 2 Kb, EEPROM 1 Kb y velocidad de reloj 16 MHz.

3.8.3. Shield Arduino

Un *shield* es una placa impresa que se pueden conectar en la parte superior de la placa Arduino para ampliar sus capacidades, pudiendo ser apilada una encima de la otra. Suelen ser diseños bastante simples y en general de código abierto, a su vez suelen ser publicados libremente. A continuación se va a mostrar algunos shield usados en algunos proyectos para la expandir su capacidad.

3.8.3.1. Arduino Wifi

Este Shield permite la conexión a internet sin cables, a una red Wifi. Lleva Incorporada una ranura micro SD y una conexión Micro-USB para poder llevar a cabo las actualizaciones de firmware Wifi. También permite realizar conexiones al exterior mediante el conector de antena R-SMA que trae incorporado en su sistema.

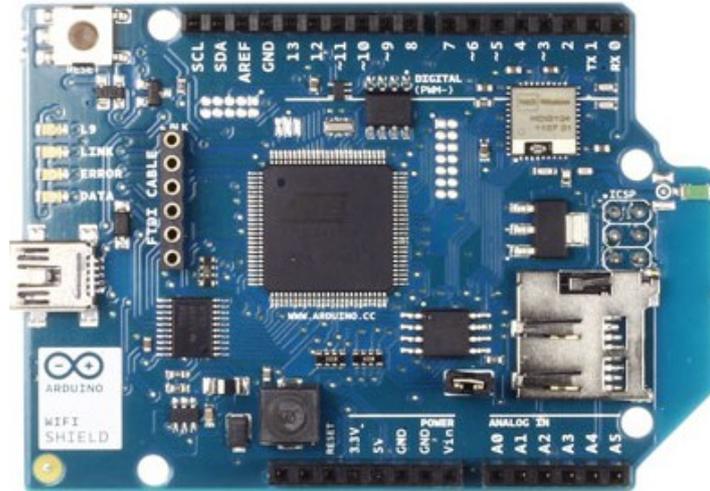


Ilustración 40: Placa Arduino wifi

Características más importantes de la placa:

- Tensión de trabajo 5 V, suministrados desde la placa Arduino
- Conexión a través de redes 802.11b/g.
- Tipos de cifrados WEP y WPA2.
- Conexión con Arduino a través del puerto SPI a través de ICSP, por lo que es compatible con cualquier Arduino que tenga este puerto.
- Requiere una placa Arduino, y la comunicación con este es mediante serial Software.

3.8.3.2. Arduino GSM

Las placas shield Arduino GSM, permiten que pueda conectar Arduino por vía internet independientemente del lugar en el que esté, enviar y recibir información, así mismo como poder realizar llamadas de voz y de datos. Esta placa es compatible con Arduino UNO, Mega, ADK, mediante algunas modificaciones puede ser compatible con la placa Leonardo. También otra de las características de este shield es que son compatibles con cualquier SIM de cualquier operador.



Ilustración 41: Placa shield Arduino GSM

Ilustración 42: shield Arduino GSM

Características más importantes de la placa:

- Tensión de trabajo 5 V, suministrado desde la placa Arduino.
- Conexión a través de GSM y GPRS.
- Requiere una placa Arduino, y la comunicación con este es mediante serial Software.

3.8.3.3. Arduino motor

La placa motor normalmente es un controlador de puente completo, diseñado para manejar cargas inductivas, en este caso para los motores. Se puede controlar con facilidad 2 motores DC la velocidad y dirección de forma independiente de cada uno. También se puede medir el consumo de corriente de los motores. Contiene 2 canales separados (A y B) que usa 4 pines de la placa Arduino para variar la velocidad, dirección de rotación, freno rápido, haciendo un total de 8 pines de utilización, pudiendo utilizar cada canal por separado para controlar 2 motores por separado.

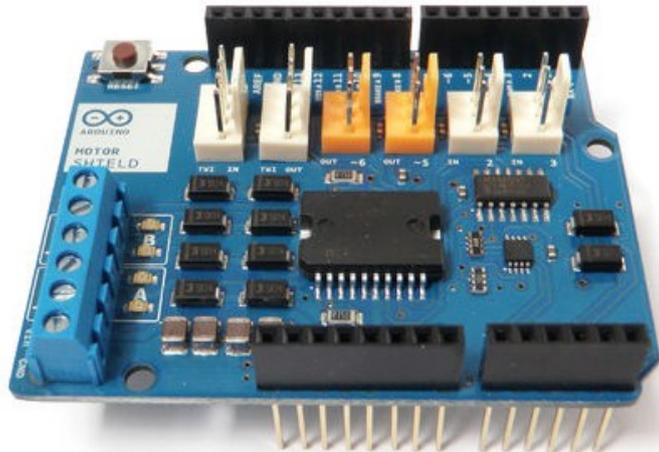


Ilustración 43: Placa shield Arduino motor

Características más importantes de la placa:

- Tensión de trabajo 5 V, suministrado desde la placa Arduino.
- Corriente máxima por canal de 2 A, con un total de 4 amperios.

3.8.3.4. Controlador puente H L298N

La función esencial para esta placa es la de ayudar a controlar la velocidad, giros de motores y frenados rápidos en DC. Su diseño se centra a la utilización de coches inteligentes para Arduino.

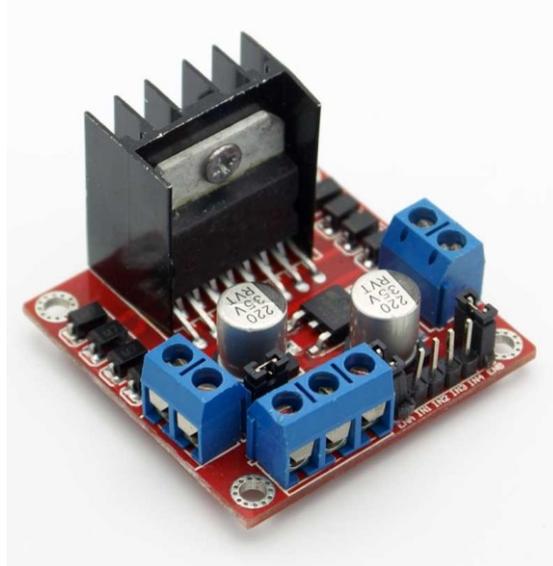


Ilustración 44: shield controlador Puento H

Características más importantes de la placa:

- Tensión nominal: 3-15V.
- Corriente nominal: 30 A
- Corriente de pico: 70 A.
- Resistencia de los contactos: 0.005 ohm
- Frecuencia de resonancia: 1 k a 200 kHz

3.8.3.5. Módulos de Largo alcance RF Transmisor y Receptor

Normalmente estas placas trabajan en la frecuencia 315MHz. Para la utilización de estos módulos harían falta dos Arduinos para que se comunicaran entre sí, de tal manera que, uno esté conectado al PC y otro conectado al dispositivo que se quiere controlar.

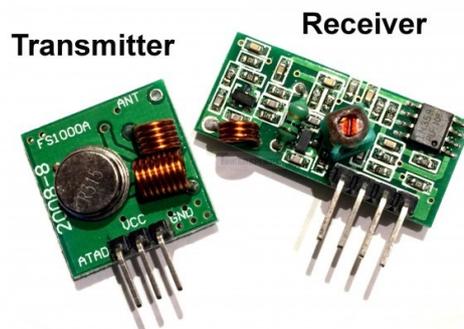


Ilustración 45: Módulos RF transmisor y RF receptor

3.8.3.6. Módulo puerto serie HC-06 Bluetooth

Básicamente se Incorpora este módulo al Arduino, gracias a este mecanismo se consigue la conexión entre dispositivos Bluetooth, sin necesidad de cables, es decir, de manera inalámbrica como puede ser comunicación con PC. El alcance óptimo aconsejado son máximo 10 metros.

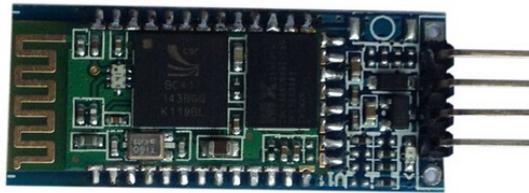


Ilustración 46: Módulo HC-06 delantero y Trasero

3.8.3.7. Módulos APC220 PC Kit-138

Sirven principalmente para la transmisión de datos entre PC y Arduino. Para que se consiga dicho objetivo, es necesario instalar en el Arduino el modulo APC220 para que se comuniquen entre ellos. Tienen una distancia de transmisión de hasta 1200m aproximadamente.



Ilustración 47: Conector USB del Módulo APC220 y antena emisor-receptor

3.8.3.8. Shield y módulos XBee

La utilización de estos módulos se realiza para obtener conexiones inalámbricas. Se pueden ser utilizados para conectarse entre sí o crear redes de ordenadores, transmitir datos de un Arduino a PC, o para comunicar un Arduino a otro Arduino.



Ilustración 48: Shield de Arduino con Módulo XBee

3.8.3.9. Módulos nRF24LU1 y nRF201

Sirven para conectar el PC y Arduino sin cables y permitir la transferencia de datos entre ambos equipos. Usualmente trabaja en la Radio de 2.4GHz con una distancia de transmisión de datos de hasta 100m aproximadamente.



Ilustración 49: Módulos nRF24LU1 y nRF201

3.8.4. Sensores Arduino

Los sensores Arduino normalmente son aquellos que ya vienen preparados para poder ser utilizados con algún tipo de las diversas placas de Arduino y que se pueden adaptar a las características de la placa. En ciertas condiciones esos sensores no requieren de la necesidad de ser soldados, aunque a veces se realiza para conseguir que no se muevan de donde los hayan instalado y tener mejor organizado el área de trabajo. Existen multitud de sensores ya hechos para Arduino, como pueden ser de temperatura, humedad, detección de gases, luminosidad, etc.

✚ **Nota:** Para la realización de este proyecto, se haría uso de algunos sensores como: detectores de presencia o de movimiento, sensores infrarrojos, fotorresistencias, etc. En los posteriores capítulos se irá detallando cada uno de estos elementos y las funciones que deberían realizar para la realización de este trabajo.

CAPÍTULO 4: **BARRERAS LUMIOSAS E INTELIGENTES**

4. Diseño de barreras luminosas e inteligentes para mejorar la seguridad en los pasos de peatones

4.1. Introducción

En este capítulo y como ya se vino mencionando en capítulos anteriores, se procederá a la realización o diseño del sistema propuesto cuyos objetivos son los que se detallaron en el **capítulo 2** de este documento. Haciendo una pequeña memoria sobre el enfoque de este proyecto, se trata de diseñar un sistema que actúe como refuerzo en el control del tráfico para la reducción de atropellos que se puedan producir en los pasos de peatones.

El proyecto consiste en implementar un sistema a base de sensores que permitan conocer en todo momento tanto la situación de los peatones como la situación del automóvil (*predecir la velocidad*) una vez esté dentro del área de control y poder informarle de algún modo del grado de peligro que está generando, con el fin de poder reaccionar a tiempo.

Cuando se habla de un modo de aviso, se refiere a que dentro del sistema se implantarán una serie de barreras luminosas enterradas que se irían ***iluminando de manera gradual***, dependiendo de la velocidad del vehículo calculada con el subsistema de sensorización implantado.

La idea de este modo de iluminación gradual y progresiva, tiene como objetivo informarle de manera intuitiva al conductor (*lejos de mirar su velocidad en el propio auto*) su situación en un periodo de tiempo, con el fin de que este pueda regular su velocidad antes de llegar al paso de peatones de manera segura.

Para la construcción de este dispositivo se opta por hacer primero un prototipo a pequeña escala partiendo desde cero y justificando todos los pasos. Así mismo como adoptar el uso de una serie de sensores y/o todos los demás componentes que hagan falta para llevar a cabo dicha construcción, con la esperanza que estos cumplan con las características o especificaciones requeridas. En los próximos apartados se explicará con más detalles las funciones que desempeñarán todos los componentes que se integrarán en el sistema.

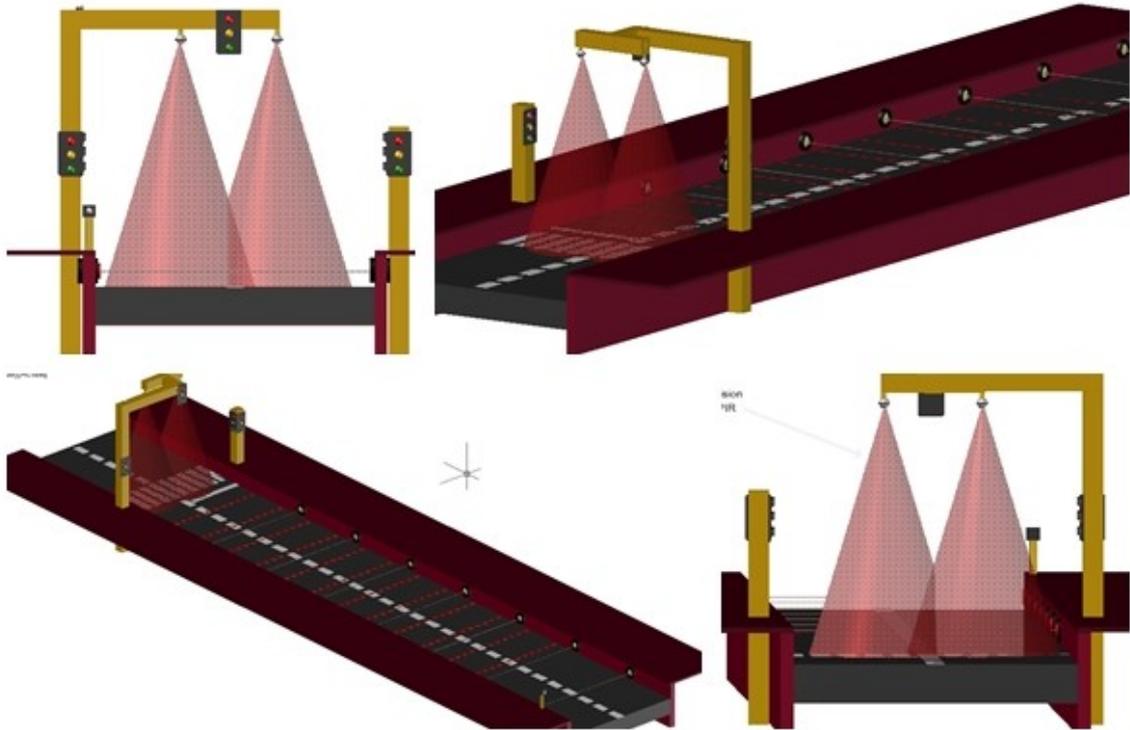


Ilustración 50: Vistas 3D del entono de pruebas

Una vez finalizado el prototipo y comprobado que funciona, se propondrá un modelo a escala real y que tenga las características o que realice funciones similares a las del prototipo. Teniendo en cuenta que controlador y el código de programación realizado no cambia, se realizará un modelado virtual con las medidas y disposiciones de los componentes, así como determinar el presupuesto de los materiales y los posibles lugares de adquisición. También se indicarán los lugares o puntos negros de una ciudad donde se podría poner a prueba el funcionamiento dicho sistema. Los lugares elegidos serán zonas de mucha concurrencia vehicular y peatonal y que a la vez son zonas donde el índice de atropellos según una serie de estadísticas realizadas previamente es mayor.

Dichas zonas serán aquellas que necesiten más seguridad y en los lugares con menos tráfico por lo tanto, un menor o cero índice de atropellos se cree que los semáforos convencionales seguirían rindiendo de igual forma y no hará falta implementar esta nueva tecnología en estos puntos.

4.2. Componentes del sistema

En la siguiente *ilustración 51* se muestra la arquitectura que llevara el sistema, tanto del prototipo como del modelo real.

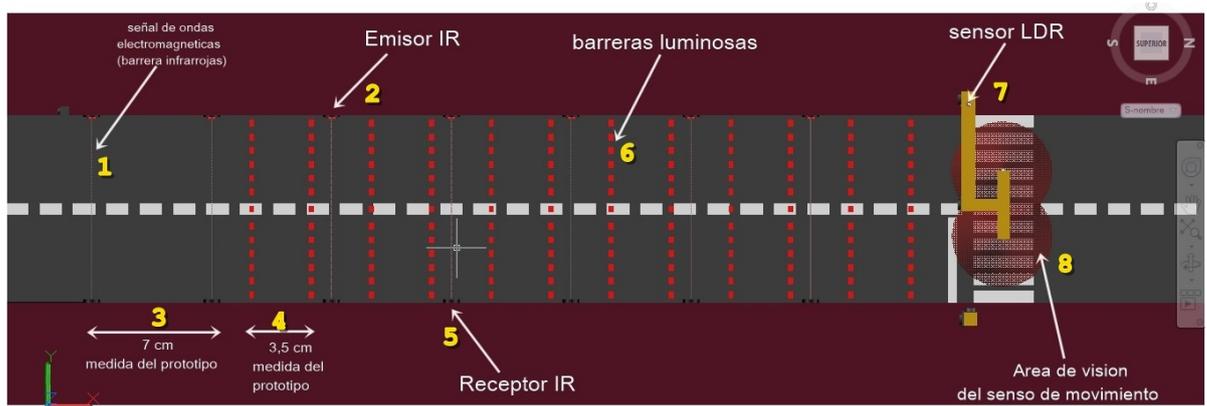


Ilustración 51: Arquitectura del sistema

1. **Señales infrarrojas.** Son las señales en forma de ondas electromagnéticas que emiten los emisores de infrarrojos y que llegan a los receptores.
2. **Emisor IR.** Es el módulo emisor de las señales infrarrojas.
3. **Distancia entre barreras IR.** Es la separación entre dos barreras infrarrojas con un valor de **7 cm** en el prototipo.
4. **Distancia entre barreras luminosas.** Es la separación entre las barreras luminosas que simularán el aviso prolongado a medida que se vayan iluminando. La separación es de **3.5 cm** para el prototipo.
5. **Receptor IR.** Es el módulo receptor de señales infrarrojas.
6. **Barreras luminosas.** Son las líneas de iluminación enterradas en el asfalto y que *simularán el aviso prolongado* sobre la situación vehicular presente en cada instante.
7. **Ubicación del Sensor LDR.** Sensor que medirá la cantidad de luz presente en el ambiente con el fin de activar de manera automática el sistema en cuanto detecte poca visibilidad. Está situado en la parte posterior con el fin de que la luz incida directamente sobre el módulo sin que esta sea perturbada por otro factor externo (*por ejemplo: las sombras.*)
8. **Área de trabajo del PIR.** Es el área de detección del sensor de movimiento cuando está en funcionamiento.

Una vez realizado un estudio del comportamiento y de las características que deseamos que se desempeñen en nuestro sistema diseñado para un prototipo y de una manera económica, se ha llegado a desarrollar una lista de todos los componentes necesarios que harían falta para poder comenzar con la construcción del dispositivo. A continuación se irá detallando la utilidad de cada componente integrado en este sistema.

4.2.1. Arduino ATMmega2560

Como ya se mencionó en el apartado 3.7.2.4. Para este trabajo se ha optado por utilizar la placa de **Arduino ATmega2560** porque atendiendo a sus características, podría realizar todas las funciones que se desea incorporar en el sistema y sobre todo porque tiene mayor número de pines.

El sistema a diseñar dispone de muchos sensores que deben estar cada uno conectado a su propio pin para captar información u obedecer las funciones de la placa, por lo tanto se requiere un total de 42 pines para conectar todo el sistema.

En la siguiente ilustración se mostrará una placa Arduino ATmega2560 y en el cual se va describiendo punto a punto cada uno de sus integrantes.

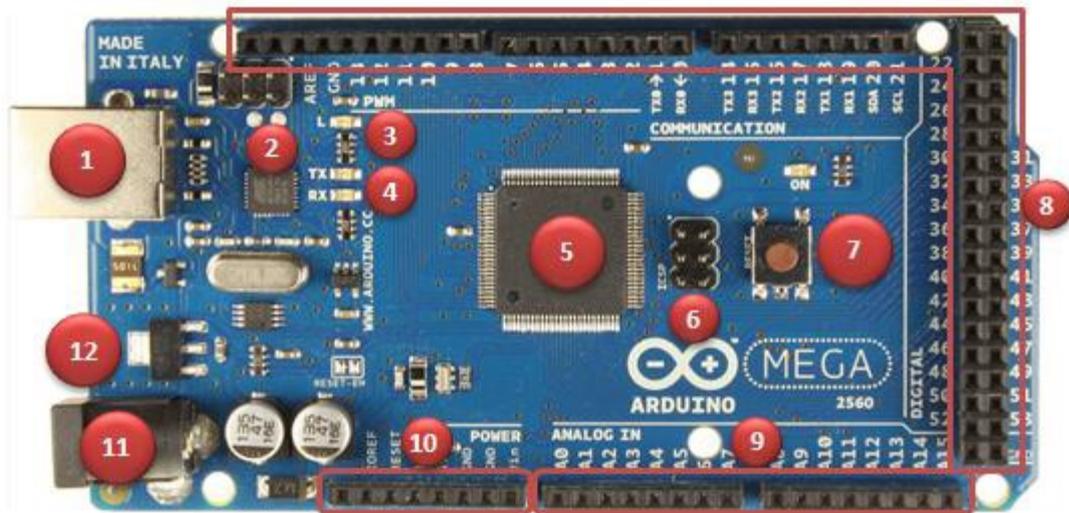


Ilustración 52: placa Arduino ATmega2560 con especificaciones

- 1) Conector USB para el cable tipo AB.
- 2) ATmega 2560 encargado de la comunicación.
- 3) Led naranja conectado al pin13.
- 4) Led TX (Transmisor) y RX (Receptor) de la comunicación serial.
- 5) Microcontrolador ATmega 2560.
- 6) Puerto ICSP para programación serial.
- 7) Botón para resetear el microcontrolador.
- 8) Pines E/S digitales, PWM y de comunicación.
- 9) Entradas analógicas.
- 10) Pines de voltaje y tierra.
- 11) Conector hembra 2.1 mm con centro positivo.
- 12) Regulador de voltaje.

Tabla 4: pines utilizados en el Arduino

Componentes	Tipo de entrada/salida	Numero de pines
Barreras luminosas	Digitales	12
Módulos emisores de infrarrojos	Digitales	7
Módulos receptores de infrarrojos	Analógicas	7
Estados semáforo	Digitales	3
Sensor de movimiento	Digital	1
Sensor de luz foto resistivo	Analógica	1
Indicador de corte de barrera	Digital	1
Pantalla LCD	Digitales	8
Alimentación 5V	Tensión	1
Tierra o GND	Tensión	1
Total pines		42

4.2.2. Modulo de sensor emisor y receptor de infrarrojos

Teniendo en cuenta que nuestro sistema no puede acceder o tomar el control sobre la situación del vehículo y regular su velocidad, la utilización de estos módulos infrarrojos en el sistema permitirá detectar los vehículos y calcular su velocidad en cada tramo.

Para hacer posible esto, se ha pensado en instalar al menos siete barreras (el **emisor** a un lado de la calzada y el **receptor** al otro lado, formando una barrera invisible) separados a una cierta distancia un par del otro (*en centímetros*). Lo que se pretende en realidad es conocer el tiempo que transcurre desde que un vehículo corta dos barreras consecutivas que componen un tramo (*por ejemplo A-B*).

Conociendo la separación entre los pares de sensores y conociendo el tiempo que ha transcurrido en cortar las dos barreras consecutivas, se aplicaría conocimientos básicos de la física y se calcularía la velocidad a la que viaja el vehículo suponiendo que va con aceleración constante.

(13)

$$V = \frac{s}{t} \text{ con aceleración constante}$$

Ecuación 13: velocidad con aceleración constante

De donde:

S: es la separación entre barreras

t: el tiempo que transcurre desde que un vehículo corta dos barreras consecutivas

Siendo esta una velocidad estimada, lo único que se pretende conocer en todo momento es si el conductor ha excedido de la velocidad máxima permitida, ya que la verdadera velocidad es la que le proporciona el velocímetro.

a) **Módulo emisor IR383:** el emisor de infrarrojos se utiliza para transmitir señales infrarrojas a través de un LED infrarrojo hacia el receptor. Este LED es como cualquier

otro LED con una longitud de onda de 940nm. Es utilizado para transmitir datos o emular mandos a distancia.



Ilustración 53: Diodo emisor infrarrojos

Características técnicas

- Tipo: Modulo emisor TSOP438.
- Longitud de onda: 940nm.
- Voltaje: 2.7-5V DC.
- Temperatura de trabajo: -20°C a +85°C.
- Frecuencia de trabajo: 38 KHz.

b) **Modulo receptor IR VS 1838:** el receptor de infrarrojos es utilizado para recibir señales de infrarrojos y también se utiliza para detectar un mando a distancia mediante un control remoto. El detector IR tiene un demodulador que busca IR modulada a 38 KHz. Este receptor puede recibir señales de buena calidad hasta una longitud de 10 metros, por lo que fuera de ese rango se puede perder la conexión.



Ilustración 54: Diodo receptor de infrarrojos

Características técnicas

- Tipo: Modulo receptor VS 1838.
- Voltaje: 2.7-5V DC.
- Temperatura de trabajo: -20°C a +85°C.
- Frecuencia de trabajo: 38 KHz.

4.2.3. Módulo detector de movimiento sensor PIR HCSR501

Al igual que los sensores IR (*sensores infrarrojos*), los módulos detectores de movimiento o sensores PIR formarán parte de la inteligencia de nuestro sistema. Su uso está pensado para ser instalados sobre los cruces de peatones (*o paso de cebra*) a una determinada altura con respecto del suelo y cuya función será detectar o tomar datos (*suponiendo que el semáforo esta en rojo, verde o ámbar*) y comprobar si existe algún tipo de movimiento, es decir, comprobar si todavía queda gente o algún ser vivo cruzando la vía.

La toma de información de estos sensores es fundamental ya que dependiendo de los datos que nos proporcione durante su periodo de funcionamiento y junto con la información proporcionada por los sensores infrarrojos, nuestro sistema adoptaría un determinado comportamiento, es decir, si no es necesario activar las barreras luminosas, si es necesario activar y especificar cuántas se deben activar exactamente dependiendo de la posición o distancia a la que se encontraba el vehículo tras realizar el último corte.

Es importante resaltar dos aspectos fundamentales:

- El detector de movimiento elegido tiene que ser capaz de detectar cuerpos en movimiento y que sean capaces de emitir radiaciones infrarrojas debido a la variación de temperatura corporal.
- El sensor debe actúa sólo y solo si el sistema este operativo, es decir, si ya se registra poca iluminación en el ambiente e independientemente del estado del semáforo (rojo, ámbar, verde).

a) Sensor PIR: un sensor infrarrojo pasivo (o sensor PIR) es un sensor electrónico o dispositivo piroeléctrico que mide cambios en los niveles de radiación infrarroja (IR) emitida por los objetos situados en su campo de visión. Se utilizan principalmente en los detectores de movimiento basados en PIR. Como respuesta al movimiento, el sensor cambia de nivel lógico de un 'pin', por lo cual su uso es extremadamente simple.

Todos los objetos con una temperatura por encima del cero absoluto emiten calor y produciendo energía en forma de radiación. Por lo general, esta radiación es invisible para el ojo humano, ya que irradia en longitudes de onda infrarrojas, pero puede ser detectado por dispositivos electrónicos diseñados para tal propósito. Es importante tener en cuenta que los sensores PIR no detectan o miden "calor", sino que detectan la radiación infrarroja emitida por un objeto, que es diferente pero que a menudo está asociado/correlacionado con la temperatura del objeto.

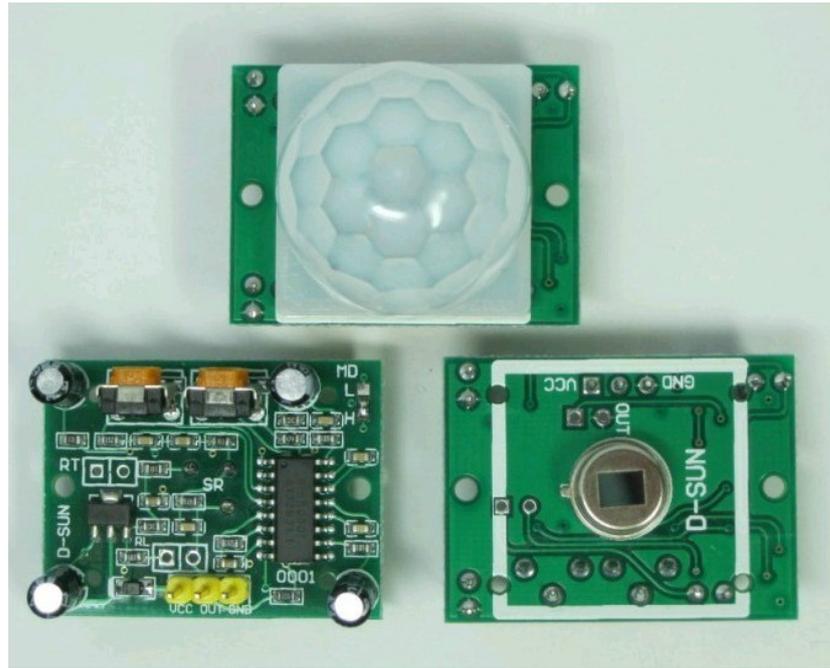


Ilustración 55: sensor PIR HCSR501

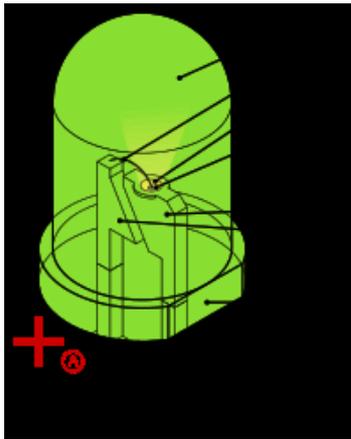
Características técnicas

- Tiempo de bloqueo: 0.2 s (estándar).
- Rango Detector: aproximadamente 7m.
- Ángulo de detección: menos de 120°.
- Gama del Voltaje: 4.5 – 20V DC.
- Corriente en reposo: menos de 50uA,
- Tensión nivel de salida: 3V alta / baja 0V.
- Temperatura de trabajo: -15°C a 70°C.
- Método de activación: L disparador irrepitable / H disparador repetible.

4.2.4.Diodos LED's

Actualmente en casi todas las tecnologías donde se desarrollan dispositivos que tengan una emisión de luz (*iluminación*) con la necesidad de hacer uso de muy poca energía utilizan la tecnología de diodos LED. Como bien se viene mencionando en este proyecto se está llevando a cabo la construcción del prototipo con barreras luminosas, por lo tanto, se hace uso de esta misma tecnología para crear barreras de iluminación (*color rojo*), así como otros usos tales como un indicador de corte de barrera (*luce azul*), etc.

a) **LED:** Un LED (*Light Emitting Diode*) consiste en un dispositivo que en su interior contiene un material semiconductor que al aplicarle una pequeña corriente eléctrica (*entre el cátodo y el ánodo*) produce luz. Esta luz emitida adopta un determinado color que depende de los materiales semiconductores con los que ha sido fabricado, no produce calor, por lo tanto no se produce aumento de temperatura [25].



- A: Ánodo.
- B: Cátodo.
- 1: lente/ encapsulado epóxico (capsula de plástico).
- 2: contacto metálico (hilo conductor).
- 3: cavidad reflectora (copa reflectora).
- 4: terminación del semiconductor.
- 5: yunque.
- 6: poste.
- 7: Marco conductor.
- 8: Borde plano.

Ilustración 56: Componentes físicos de un LED

Características técnicas y mariales de construcción [26]

Composición del chip	Nombre del compuesto	Color de la luz emitida	Tensión de trabajo en volt (V)	Frecuencia en hertz (Hz)	Longitud de onda en nm
GaAs	Arseniuro de galio	Infrarrojo	< 1,9	< 4,0 x 10 ¹⁴	> 760
GaAlAs	Arseniuro de galio y aluminio				
GaP	Fosfuro de galio				
GaAlAs	Arseniuro de galio y aluminio	± 1,8	4,8 - 4,0 x 10 ¹⁴	610 - 760	
AlInGaP	Fosfuro de aluminio indio y galio				
GaAsP/GaP	Fosfuro de galio y arsénico / Fosfuro de galio				
AlInGaP	Fosfuro de aluminio indio y galio	± 2,0	5,1 - 4,8 x 10 ¹⁴	590 - 610	
GaAsP/GaP	Fosfuro de galio y arsénico / Fosfuro de galio				
AlInGaP	Fosfuro de aluminio indio y galio				
GaP	Fosfuro de galio	± 3,0	5,3 - 5,1 x 10 ¹⁴	570 - 590	
InGaN	Nitruro de indio y galio				
GaN	Nitruro de galio				
InGaN / Zafiro	Nitruro de indio y galio / Zafiro	± 3,3	5,8 - 5,3 x 10 ¹⁴	500 - 570	
SiC	Carburo de silicio				
InGaN / Zafiro	Nitruro de indio y galio / Zafiro				
GaN	Nitruro de galio	± 3,3	6,7 - 6,0 x 10 ¹⁴	450 - 500	
InGaN / Zafiro	Nitruro de indio y galio / Zafiro				
InGaN	Nitruro de indio y galio				
InGaN	Nitruro de indio y galio	± 3,4	7,9 - 6,7 x 10 ¹⁴	450 - 500	
GaN	Nitruro de galio	± 3,7	>7,9 x 10 ¹⁴		
Ce:YAG	Granate-aluminio-itrío, dopado de cerio	± 3,4	Espectro completo		

4.2.5. Sensor LDR (CE-C2795).

La utilización del sensor LDR en este proyecto consiste en poder captar la cantidad de iluminación que se detecta en el exterior.

La razón de su utilización es debido a que, el sistema que se pretende diseñar, sirve como refuerzo para la **reducción de atropellos** que se producen de noche o en tramos con poca iluminación, por lo tanto, el sistema implantado solo entraría en funcionamiento si se detecta que hay poca iluminación ya sea que el día este muy nublado, sea tarde o en el peor de los casos sea de noche. De este modo se consigue un **ahorro de energía** en horas con mucha iluminación y no sería necesario que entre en funcionamiento el sistema.

Las **fotorresistencias o LDR** dependen de la luz y está formado por una célula de sulfuro de cadmio, altamente estable, encapsulada con una resina epoxi transparente y resistente a la humedad. La respuesta espectral es similar a la del ojo humano.

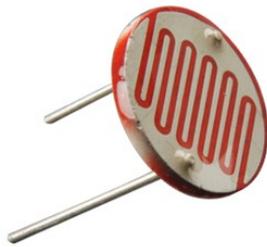


Ilustración 57: sensor LDR CE-C2795

Como podemos observar en la siguiente gráfica, Su nivel de resistencia aumenta cuando el nivel de luz disminuye y viceversa.

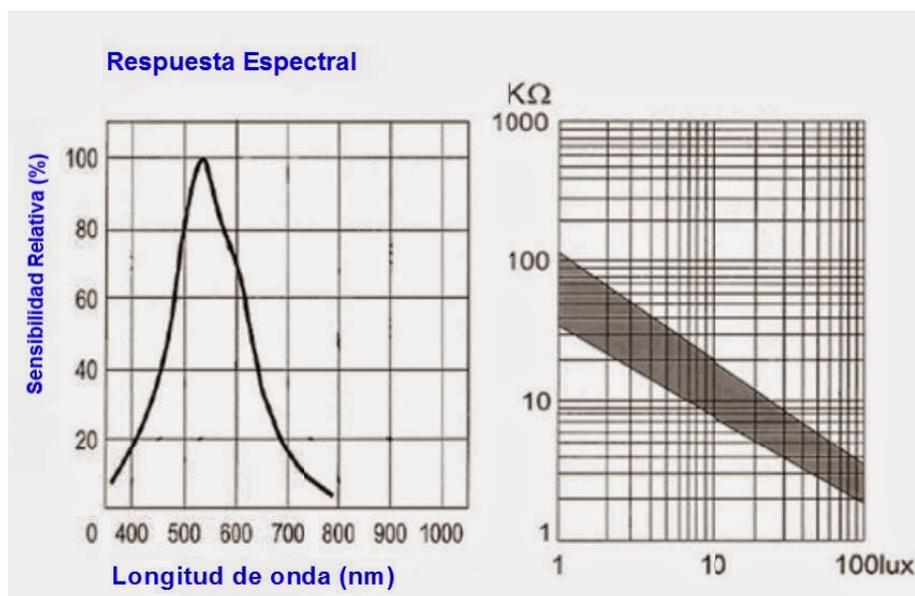


Ilustración 58: Comportamiento del sensor LDR CE-C2795 en de la iluminaria

4.3. Montaje del entorno de pruebas

Como ya se viene mencionando sobre la construcción de un entorno de pruebas para nuestro sistema, en este apartado se va a detallar todos los pasos realizados para obtener tanto la construcción de nuestro prototipo, como la instalación que se llevará a cabo sobre dicha maqueta.

4.3.1. Material de montaje

Actualmente en el mercado existen diferentes maquetas ya realizadas, pero teniendo en cuenta que es difícil encontrar una maqueta que cumpla con los requisitos y todas las mediciones que uno pretende utilizar en su proyecto, se prefirió buscar material casero y realizar un diseño que se adapte mejor a las especificaciones.

Para la realización de lo que es la estructura del prototipo se ha optado por utilizar material de cartón de 6mm de grosor debido a que es muy fácil de tratar en lo que se refiere a cortar, agujerear y unificar las piezas ya sea con pegamento o grapando.

- **Justificación del tipo de material utilizado**

Se podía haber utilizado el contrachapado o encargar una maqueta en una carpintería para facilitar el trabajo, fue una de mis primeras opciones pero el precio me salía fuera del presupuesto, así como adquirir herramientas como la cinta de corte o una mini taladradora, etc. Ante todas estas dificultades preferí hacer uno con un material más fácil de tratar con herramientas caseras de casa.

- **Pasos del diseño**

a) **Trazados de las medidas:** este viene a ser el primer paso y consistía en trazar sobre la tabla de cartón las dimensiones de lo que será el asfalto, las paredes que lo sujetarían, las calzadas y la distancia de separación a la que irían las barreras tanto las luminosas y las de infrarrojo.

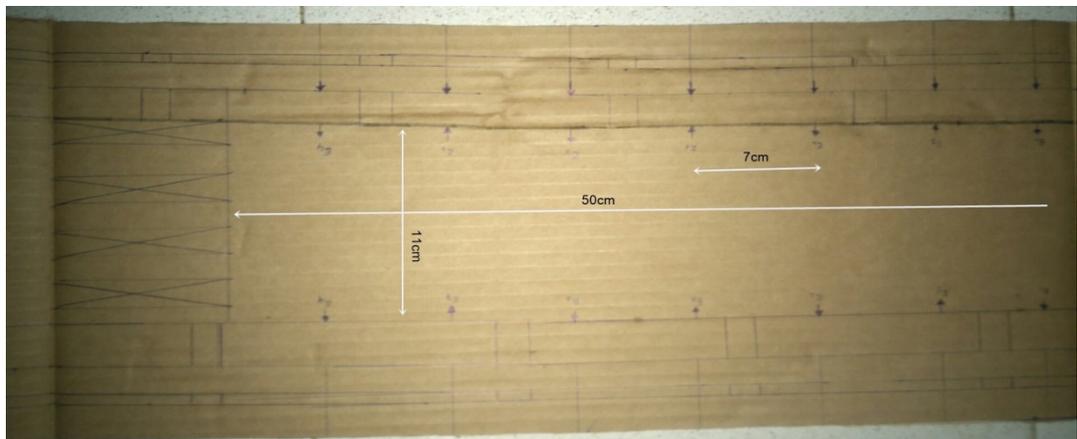


Ilustración 59: tabla con trazado de medidas

- **50cm** es la distancia comprendida desde el paso de peatones hasta última barrera infrarroja, esta distancia determina todo el escenario sobre el que se realizará el control.
- **11cm** determina la anchura del asfalto determinado en función al tamaño del cochecito de juguete que se utilizará en la simulación.
- **7cm** es la distancia entre barreras infrarrojas que determinan un tramo.

b) **Recorte y primer encaje:** siguiendo con el diseño se procedió a recortar la tabla mediante una herramienta de corte que se muestra en la *ilustración 60* y posteriormente realizar el encaje de las piezas recortadas. Una vez encajada las piezas se debía asegurarlas para que no se suelten. Para ellos se hizo uso de de un pegamento de cartón o de madera casero y sin tener que utilizar tornillos.



Ilustración 60: herramienta de corte y pegamento

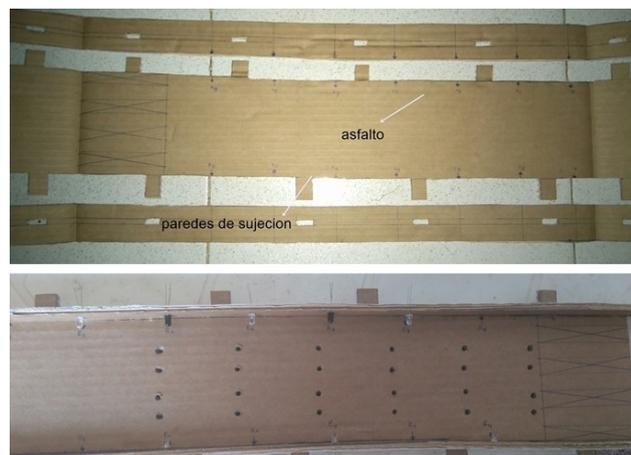


Ilustración 61: Recorte y encaje de piezas

Otra de las cosas que se realizó durante esta fase fue agujerear la pista como se muestra en la *ilustración 61* los cuales servirán para instalar las barreras luminosas incrustadas. Al no disponer de una mini taladradora utilicé un tornillo con un diámetro significativo como se muestra en la *ilustración 62*.



Ilustración 62: Tornillo utilizado como taladro

c) **Mano de pintura y trazado de la señalización:** en este paso lo que se quería conseguir era darle una estructura similar a un escenario real, por lo tanto, se procedió a pasarle una mano de pintura a la estructura y posteriormente marcar la señalización del tráfico.

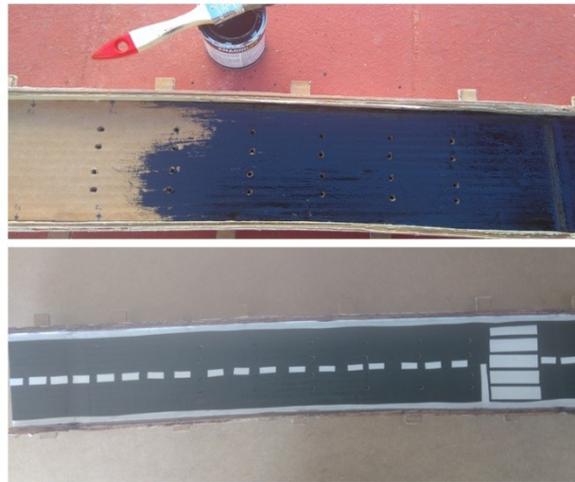


Ilustración 63: pintura y señalización

Se utilizó pintura plástica fácil de secar sobre papel y para la señalización se utilizó trozos de cinta aislante de color blanco, ya que salía más barata que un bote de pintura blanca y además se iba a utilizar para otros fines dentro del proyecto.



Ilustración 64: material de pintura y señalización

4.3.2. Instalación de los componentes electrónicos

Acabada la construcción de la maqueta del entorno de pruebas, se procedió con la instalación de los distintos componentes electrónicos elegidos para nuestro sistema y justificando su modo de instalación o distribución.

4.3.2.1. Materiales y herramientas de utilización:

Para llevar a cabo nuestra instalación se utilizó los siguientes materiales:

- **La Placas PCB:** es la superficie constituida por caminos, pistas o buses de material conductor laminadas sobre una base no conductora. El circuito impreso se utiliza para conectar eléctricamente a través de las pistas conductoras, y sostener mecánicamente, por medio de la base, un conjunto de componentes electrónicos. Las pistas son generalmente de cobre mientras que la base se fabrica de resinas de fibra de vidrio reforzada, cerámica, plástico, teflón o polímeros como la baquelita.
- **Soldador:** es una herramienta electrónica que se utiliza principalmente para soldar los distintos componentes que constituyen un circuito sobre la placa PCB con el fin de mantenerlos fijos y evitar que estas puedan moverse e interrumpir el correcto funcionamiento de un circuito electrónico.
- **Estaño:** es el material que acompaña al soldador con el fin de poder llevar a cabo el posicionamiento fijado de los distintos componentes dentro del circuito.
- **Mini alicate de corte:** como su nombre indica, ese se utilizó para realizar cortes sobre el material conductor (*cables*).
- **Cinta adhesiva:** esta cinta tiene dos principales funciones. La primera es su utilización es la de aferrar, es decir, tapar los cerramientos, puntos convergentes o de unión entre dos piezas dentro de la estructura. Otra función secundaria es la de utilizarla para crear etiquetas en los diferentes conductos procedentes de los dispositivos electrónicos con el fin de no tenerlos todos entre mezclados y poder facilitar la disposición de cada uno en

su pin correspondiente.



Ilustración 65: herramientas utilizadas en la instalación

4.3.2.2. Distribución de los módulos de infrarrojos

Haciendo una pequeña observación en la ilustración que abajo se muestra, se puede ver el modelado 3D realizado con autoCAD, se puede ver que se ha optado por instalar los diodos IR emisores por un lado de la calzada y en el otro lado los diodos IR receptores. Esta distribución está diseñada para un montaje a escala real del proyecto, ya que se podrá ver que en el entorno de pruebas esta distribución no es así y se justificara las razones.

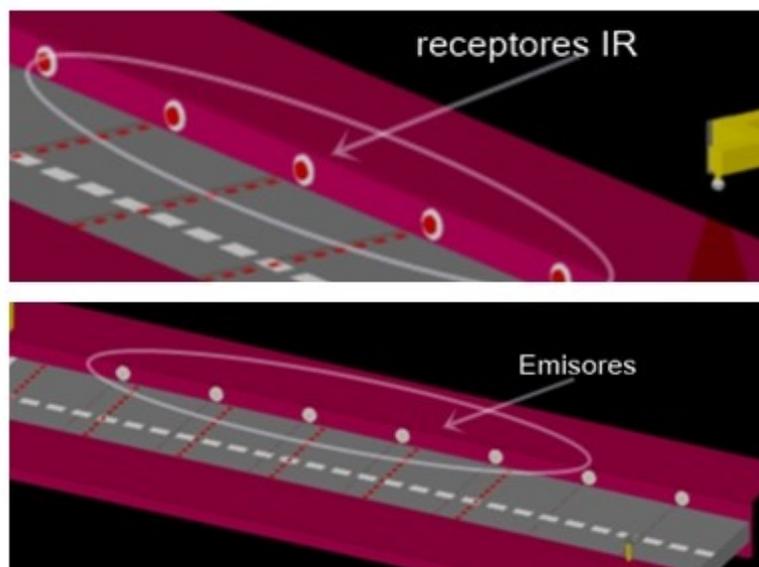


Ilustración 66: Distribución Emisor-receptor en el modelado 3D

A continuación se va mostrar una ilustración del montaje en el prototipo

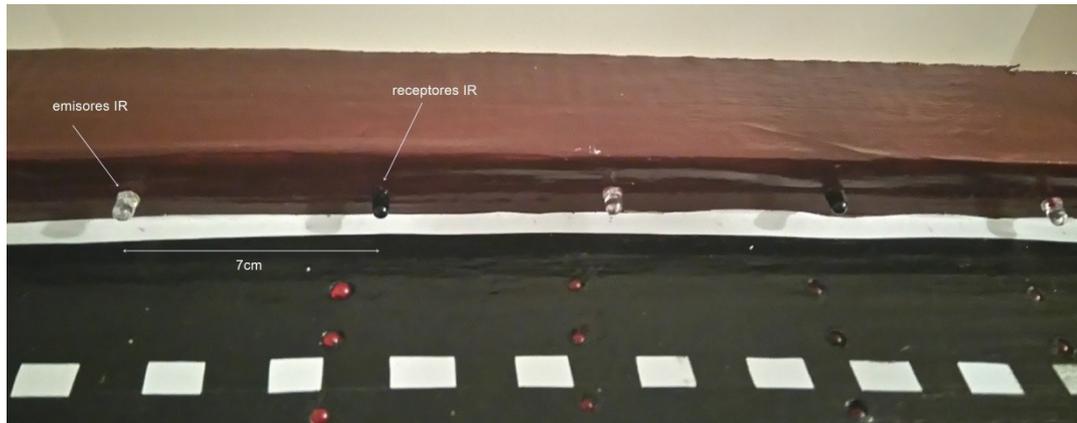


Ilustración 67: distribución de los diodos emisores y receptores de los infrarrojos

Justificación de la distribución: el principal motivo por el que se ha optado por esta distribución se fundamenta por los tipos de diodos receptores adquiridos, es decir, teniendo en cuenta su bajo coste, tienen la pequeña desventaja de que son *módulos bidireccionales*, por lo tanto, son capaces de recibir las señales electromagnéticas procedentes de los emisores en varias direcciones y no solo del emisor que tienen en frente.

Problemática: ante esta desventaja presenta la problemática de que si se instalasen de la misma manera que se ilustra en el modelado 3D, se producirían interferencias de otros emisores a la hora de realizar una determinada lectura y de este modo será imposible saber si la señal recibida procede del emisor correspondiente o es de un emisor adyacente.

Otro de los problemas que han hecho llevar a cabo esta distribución ha sido por la escasa separación que se presenta en el prototipo, al ser distancias muy reducidas es posible que algún emisor no esté correctamente alineado con el receptor y puede producirte alguna fuga de señal que apunte a otro receptor y ocasionar un mal funcionamiento en el sistema.

4.3.2.3. Distribución de las barreras luminarias

La distribución de la luminaria adoptada en este sistema fue la distribución de iluminación incrustada (*un poco similar a la iluminación de las pistas de aterrizaje*) con el fin de que estos pueden ser atravesados por los automóviles sin necesidad de romperse. Para ello se instalará barreras de seis LED cada una, aunque podría poner más para que sea más visible pero debido el ancho del asfalto esta cantidad de luminosidad sería suficiente. También cabe mencionar que se podría haber utilizado la iluminación con diodos láser solo porque presentan un montaje más sencillo, pero teniendo en cuenta el coste que supone adquirir esos módulos el presupuesto se dispararía excesivamente.

Ventajas: La gran ventaja que supone utilizar los diodos LED no solo resalta en su bajo consumo energético, sino en la posibilidad de poder regular su intensidad con el fin de

evitar deslumbramientos y así como su facilidad de mantenimiento.

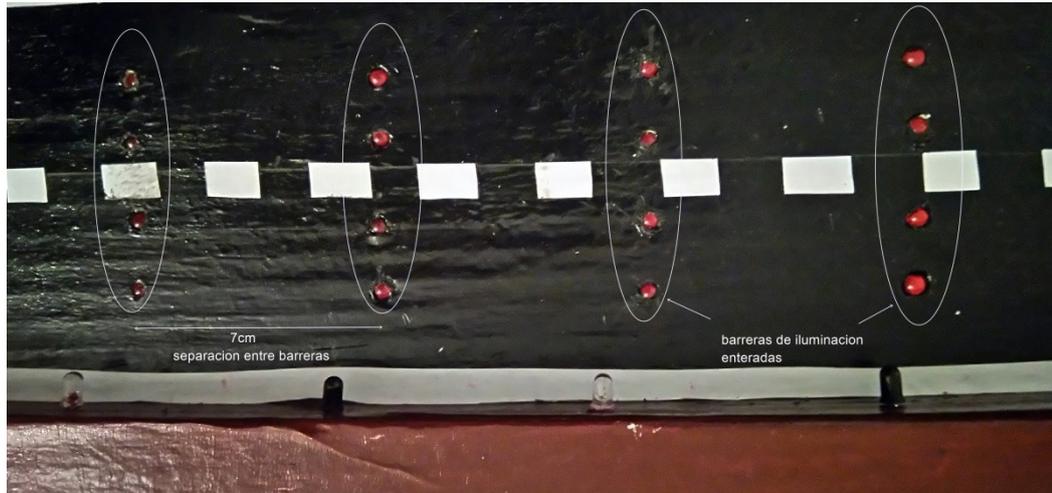


Ilustración 68: distribución de las barreras luminosas

Otra de las cosas realizadas en este apartado fue la creación del semáforo, a través del cual se realizarán diversas hipótesis con el fin de proporcionar soluciones a los diversos problemas presente en cada uno de sus estados.



Ilustración 69: Estados del semáforo

4.3.2.4. Modo de instalación de sensor PIR

Como ya se menciona la funcionalidad de este dispositivo electrónico en el apartado 4.2.3, ahora se procedería a explicar los pasos en su montaje dentro de nuestro sistema.

Haciendo una pequeña observación en el modo de instalación en la *ilustración 70*, se puede ver que en el modelado 3D se incorporaron dos sensores PIR dependiendo de la ancho de la carrera y calculando que el rango de visión de uno es de 120° lo cual no cubriría todo el área de trabajo, por lo tanto, instalando dos sensores y que trabajen conjuntamente no solo optimiza el trabajo sino que también evita pérdidas de información. Otra de las ideas por las que se adoptó este modo de distribución fue para evitar que en caso de que se estropease uno de los dispositivos, que el otro siga trabajando al menos para evitar que se produzcan caídas en el sistema.

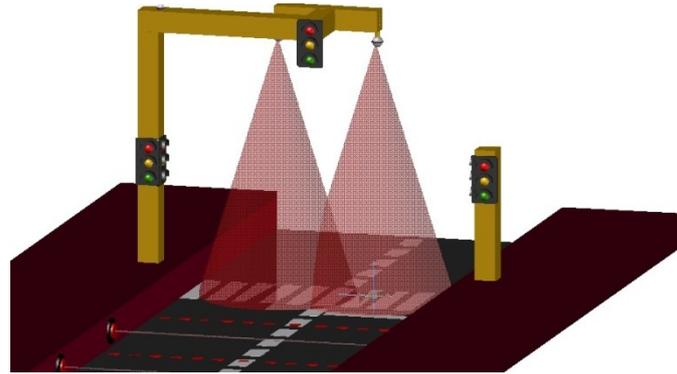


Ilustración 70: Modo de instalación del sensor PIR, modelado 3D

Teniendo en cuenta las especificación de estos pequeños dispositivos pero potentes, en este prototipo se optó por instalar solo uno debido a que al tratarse de un entorno de pruebas con medidas muy reducidas y teniendo en cuenta también que se trata de un dispositivo que se utiliza o se utilizaría en un proyecto real de Domótica, no era necesario disponer de dos sensores porque uno solo dos bastaría para cubrir todo el área de trabajo.

Cabe decir que la introducción de este dispositivo en la señalización del tráfico para reforzar la seguridad vial, no solo sería la primera vez sino que también ayudaría a resolver muchos problemas o minimizar algunos riesgos que se pasaba de largo y por lo tanto, tales riesgos podrían haber sido consecuentes de varios accidentes que se hayan producido en los cruces de peatones.

Es posible que la detección de peatones a lo largo del cruce es un problema que se solucionaría de muchas maneras y con tecnologías diferentes (*barrido por sensores láser, barrido por infrarrojos, etc.*) pero que en nuestro estudio se opta por usar la manera más rápida, sencilla, intuitiva y económicamente posible.

Justificación: al tratarse de un estudio en el que solo se pretende detectar presencia o no presencia de movimiento de seres vivos, en concreto seres humanos, la utilización de estos dispositivos y no las otras tecnologías viene a ser sumamente importante ya que como bien se dijo en su descripción, los sensores piroeléctricos (PIR) solo detectan cuerpos que irradian calor dependiendo de la variación de la temperatura corporal. Por lo tanto, solo se detectarían cuerpos vivos cuya temperatura corporal no es constante, cosa que no se podrían decir de los cuerpos inertes que desprenden calor a una temperatura constante.

A continuación se ilustra imágenes de cómo quedaría la instalación de nuestro dispositivo dentro del sistema que se está diseñando.

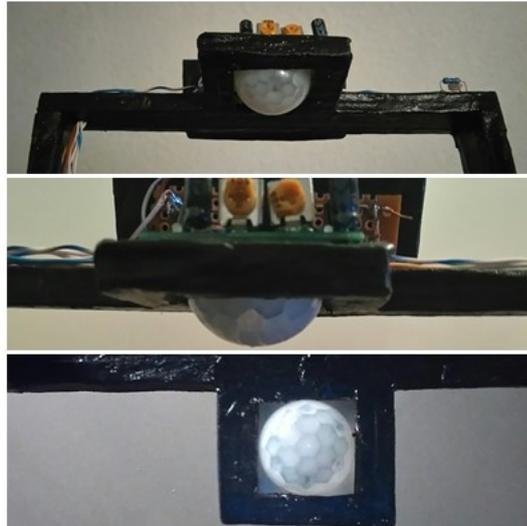


Ilustración 71: Modo de instalación del sensor PIR

4.3.2.5. Modo de instalación del sensor LDR

A pesar de su función descrita en el *apartado 4.2.5* y teniendo en cuenta que, el diseño está preparado y pensado para operar en campo abierto, la instalación de ese sensor es una de las tareas más fáciles, ya que su modo de funcionamiento depende de la cantidad de luz exterior existente, por lo tanto, se podría instalar este dispositivo en cualquier lugar dentro del sistema.

En el prototipo se opta por incorporarlo encima del travesaño para proporcionar seguridad y dificultar su acceso al personal no cualificado. Así mismo como permitir que la luz del día se refleje o incida directamente sobre las resistencias integradas en su circuito y evitando a la vez la interrupción de alguna sombra.

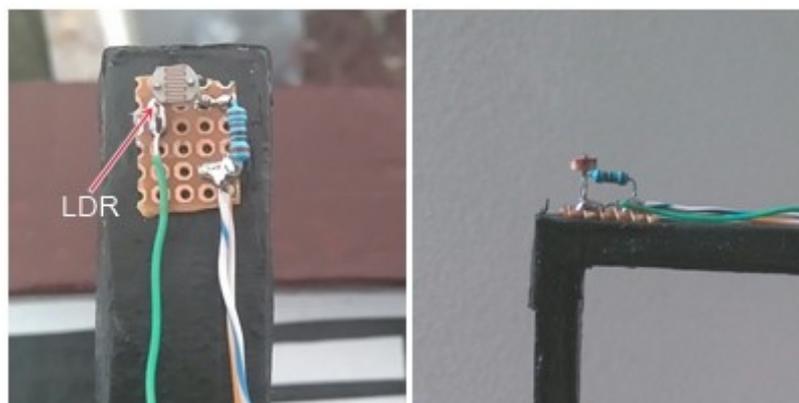


Ilustración 72: disposición del sensor LDR

En el *apartado 4.4* se explicará cómo están colocados los cables y la utilidad que tiene la resistencia de 1ohm que se puede apreciar en la imagen.

4.3.2.6. Pantalla LCD

La única finalidad de utilizar una pantalla LCD en este proyecto, no es otra que mostrar información suficiente para tener constancia del comportamiento de todo el sistema durante el proceso de control y regulación, con el fin de proporcionar una respuesta a nuestra salida y así conocer en todo momento como han ido variando todos esos parámetros que se pretende controlar.

La **LCD** (*pantalla de cristal líquido* o liquid crystal display) es una pantalla delgada y plana, que disponen de un gel líquido cristalino que se intercala entre dos paneles de vidrio o plástico de visualización. El gel se divide en píxeles por una rejilla de transistores de metal, que suministran a cada píxel con una corriente eléctrica diferente, lo que les permite iluminar en diferentes colores.

Las pantallas o display LCD, tienen la capacidad de mostrar cualquier carácter alfanumérico, permitiendo representar la información que genera cualquier equipo electrónico de una forma fácil y económica. El proceso de visualización es gobernado por un microcontrolador incorporado a la pantalla, siendo el Hitachi 44780 el modelo de controlador más utilizado.

Existen muchos tipos de LCD (distintos tamaños, colores, iluminados, sin iluminar..) pero el funcionamiento general de todos es el mismo.

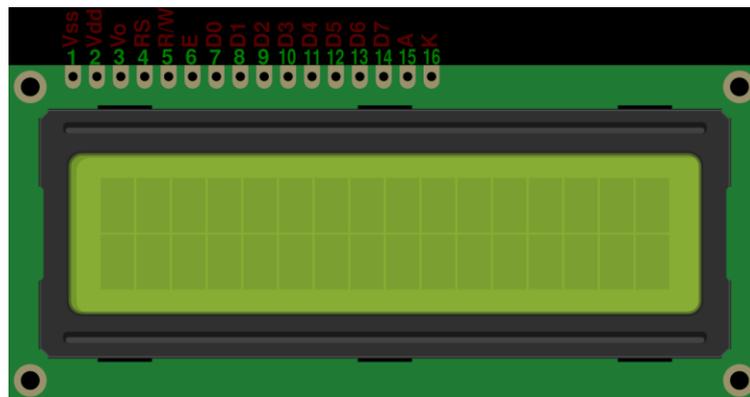


Ilustración 73: pantalla LCD

Para este proyecto se ha utilizado una pantalla LCD de 16 píxeles de largo por 2 de alto, decir, dispondremos de 2 filas y 16 columnas, las cuales presentan los 16 caracteres diferentes que se pueden ilustrar en cada fila.

Observan la imagen ilustrada anteriormente se puede ver que una serie de pines de conexión y continuación se va a describir las funcionalidades de cada una de ellas.

Tabla 5: Funciones de los pines de la pantalla LCD

PIN	NOMBRE	FUNCIÓN
1	GND	Tierra
2	Vcc	5v
3	Vo	Regula el contraste
4	Rs	Selección de registro de escritura o lectura
5	R/w	Selección de opción de lectura o escritura
6	E	Datos
7	DB0	Datos
8	DB1	Datos
9	DB2	Datos
10	DB3	Datos
11	DB4	Datos
12	DB5	Datos
13	DB6	Datos
14	DB7	Datos
15	LED A	Fuente de alimentación para la
16	LED K	Retroiluminación

A continuación se mostrara la ubicación de la pantalla LCD en este proyecto. Ya que no interviene en el funcionamiento del sistema su instalación se puede hacer en cualquier parte de nuestro entorno de pruebas siempre que no esté dentro de la zona de control.

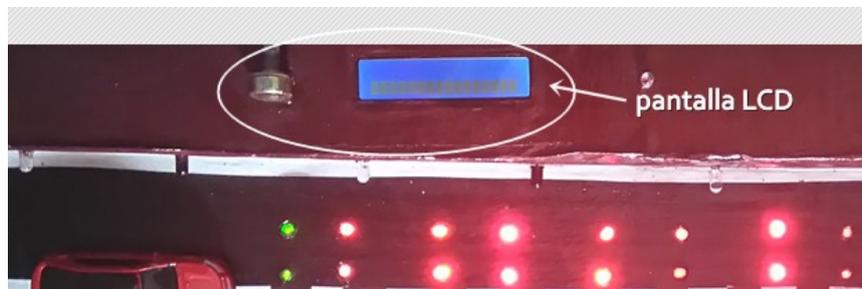


Ilustración 74: ubicación de la pantalla LCD

4.4. Esquemas de conexiones de los componentes del sistema

Este va a ser el apartado en que se explica cómo estarán conectados eléctricamente los distintos componentes electrónicos del sistema, tanto de manera individual o de manera colectiva (*barreras luminosas y barreras de infrarrojos*).

4.4.1. Tipo de cable

Para cubrir toda la instalación se ha utilizado un mismo tipo de cable, que es un cable coaxial (*normalmente viene trenzado pero se puede desenrollar*) de un único conductor recubierto con un único envoltente de plástico de un solo color o multicolor. Este conductor ha sido sacado de los cables de datos de conexión de internet o de conexión entre ordenadores debido a su reducido diámetro (1.5mm) y a su facilidad de manejo. Es un cable

UTP (*par trenzado sin blindaje*) utilizado en una de las prácticas realizadas en la asignatura de **sistemas de comunicaciones en edificios**.



Ilustración 75: cable trenzado

4.4.2. Conexión de los diodos emisores y receptores infrarrojos

- **Emisor:** la conexión de un diodo emisor con dos pines, es similar a la de cualquier diodo LED con dos terminales (*ánodo y cátodo*) siendo el ánodo, pin largo o pin positivo (+) el cual suele ir acompañado de una resistencia eléctrica para evitar que se funda. Desde esta resistencia conectada en serie con el ánodo se conectaría el conductor que lo lleva directamente al pin de datos correspondiente. Por otro lado tenemos el terminal cátodo, pin corto o negativo (-) que suele ir conectado a tierra (GND) directamente.
- **Receptor:** la conexión de un diodo receptor con dos pines lleva una configuración diferente a la del receptor en el sentido de que, el cátodo, pin corto es la que suele ir conectado directamente al pin de alimentación (5V) del Arduino mientras que, la pin largo o ánodo suele llevar incorporado una resistencia en serie que lo lleva directamente a tierra (GND). Pero para la adquisición de datos se debe conectar un conductor perpendicular justo en el punto de unión entre el ánodo y el pin de la resistencia conectada a este ánodo. Este conductor conectado en el punto de unión será llevado hasta el pin de datos correspondiente en el controlador.
 - Observando de manera independiente una única barrera infrarroja se puede ver que lleva el siguiente circuito eléctrico que será el mismo para el resto de barreras

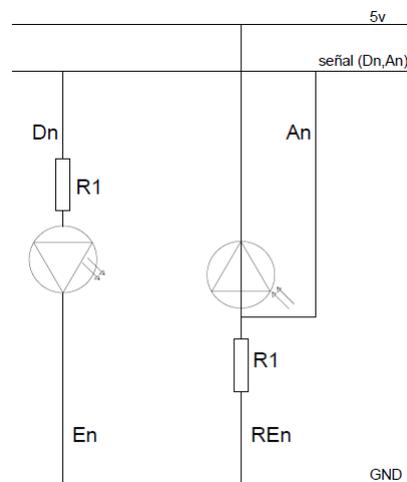


Ilustración 76: esquema eléctrico de una única barrera de infrarrojos

Donde:

- **En:** n diodo emisor de infrarrojos
 - **REn:** n diodo receptor de infrarrojos
 - **Dn:** n pin digital
 - **An:** n pin analógico
 - **5v:** tensión de alimentación 5v
- En la siguiente ilustración, se muestra cómo ha sido instalado nuestros módulos emisor y receptor utilizando resistencias de un $1k\Omega$ para cada diodo.

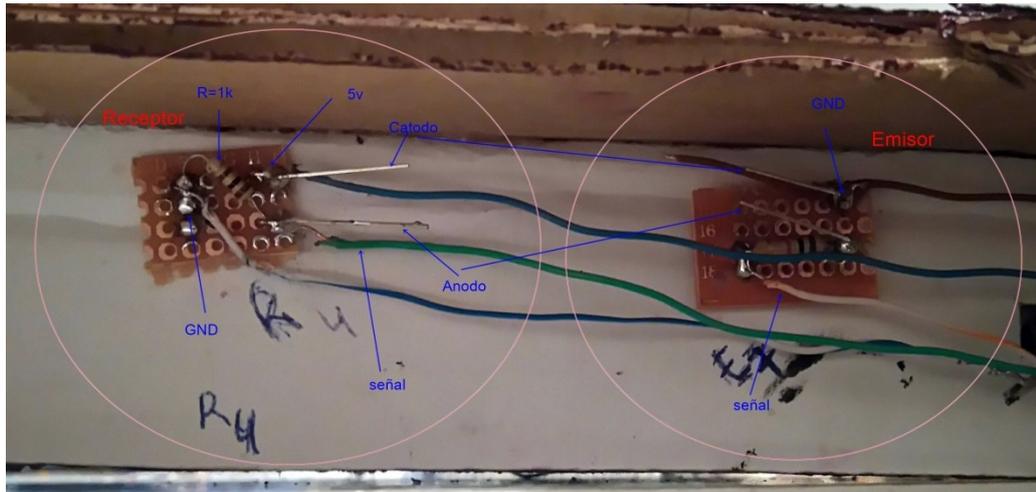


Ilustración 77: aspecto físico de conexión emisor-receptor

- Posteriormente se muestra la ilustración de todos los módulos emisores y receptores conectados en un lado de la calzada.



Ilustración 78: Aspecto físico y completo del circuito de los módulos emisor-receptor

- Todo el circuito eléctrico general correspondiente a las barreras infrarrojas se muestra en la siguiente imagen y ha sido diseñado con autoCAD.

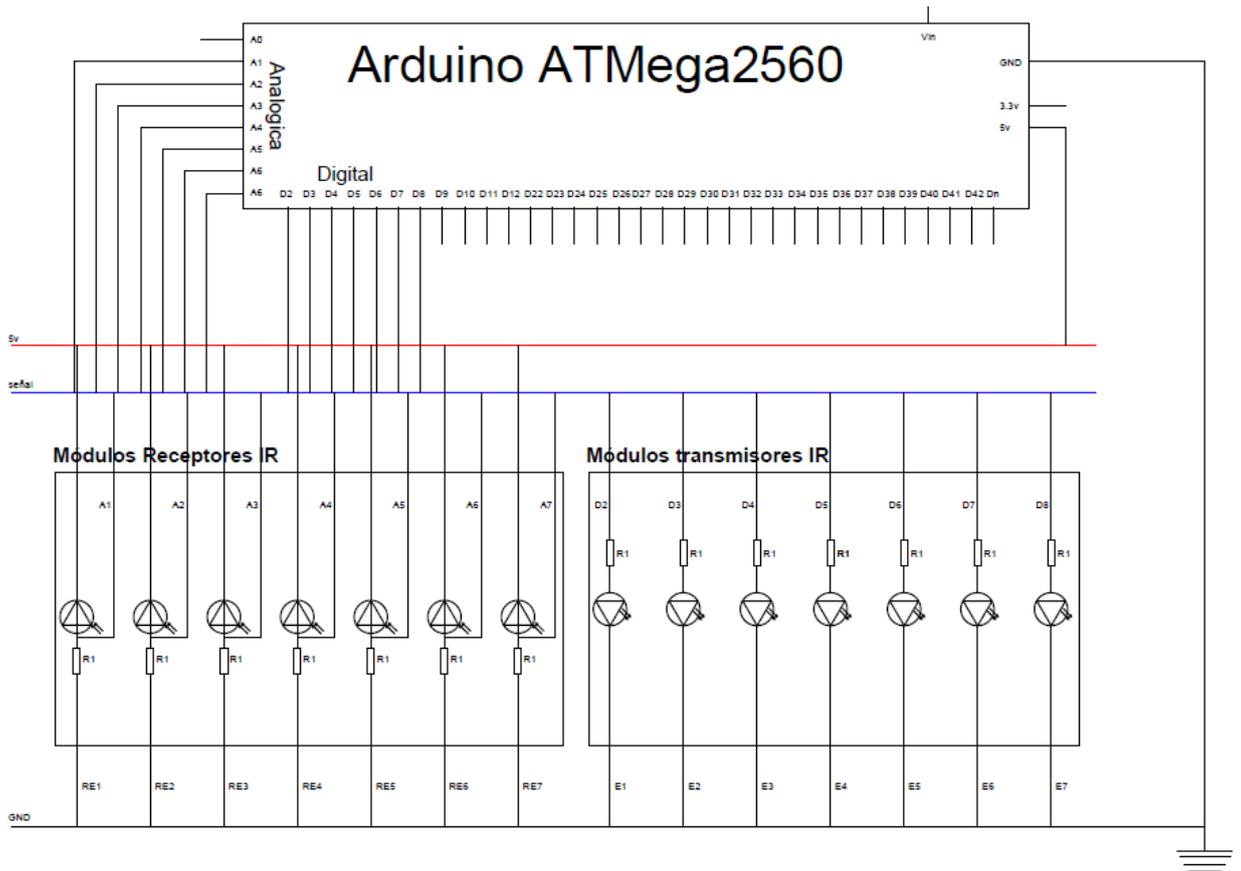


Ilustración 79: Esquema eléctrico de los módulos Emisor-receptor de infrarrojos

De donde:

- **Rn**: resistencias
- **An**: pines analógicos del Arduino
- **Dn**: pines digitales del Arduino
- **Línea roja**: tensión 5V
- **Línea azul**: señal conectada a los pines
- **Línea negra inferior**: tierra
- **REn**: módulo receptor
- **En**: módulo emisor

4.4.2.1. Fragmento del código realizado Para las barreras infrarrojas

Lo primero fue declarar las variables y asignarles sus pines correspondientes y especificar si iban a ser variables de entrada o salida.

```
// receptores infrarrojos
const int receptorIR1=A1;
const int receptorIR2=A2;
const int receptorIR3=A3;
const int receptorIR4=A4;
const int receptorIR5=A5;
const int receptorIR6=A6;
```

```

const int receptorIR7=A7;

// emisores infrarrojos
int emisorIR1=2;
int emisorIR2=3;
int emisorIR3=4;
int emisorIR4=5;
int emisorIR5=6;
int emisorIR6=7;
int emisorIR7=8;

// indicador de corte de barrera IR
int acceso_sistema=38;

// lectores IR para guardar las lecturas realizadas
int lectorIR1;
int lectorIR2;
int lectorIR3;

void setup() {
  lcd.begin(16,2);
  Serial.begin(9600);

  pinMode(emisorIR1,OUTPUT);
  pinMode(emisorIR2,OUTPUT);
  pinMode(emisorIR3,OUTPUT);
  pinMode(emisorIR4,OUTPUT);
  pinMode(emisorIR5,OUTPUT);
  pinMode(emisorIR6,OUTPUT);
  pinMode(emisorIR7,OUTPUT);

  pinMode(receptorIR1,INPUT);
  pinMode(receptorIR2,INPUT);
  pinMode(receptorIR3,INPUT);
  pinMode(receptorIR4,INPUT);
  pinMode(receptorIR5,INPUT);
  pinMode(receptorIR6,INPUT);
  pinMode(receptorIR7,INPUT);

  // indicador de corte de barrera
  pinMode(acceso_sistema,OUTPUT);
}

```

Para realizar las lecturas se creó una función que permita optimizar este proceso ya que se iba a realizar repetidas veces dentro del programa.

```

void lecturasBarreras(int receptorIR1, int receptorIR2, int
receptorIR3) {

int lectorIR1=analogRead(receptorIR1);
int lectorIR2=analogRead(receptorIR2);
int lectorIR3=analogRead(receptorIR3);

}

```

4.4.3. Conexión de los diodos LED para la iluminación de las barreras

El modo de conexión de los diodos LED en este proyecto será exactamente igual que el modo de conexión explicado en los diodos emisores infrarrojos del apartado anterior.

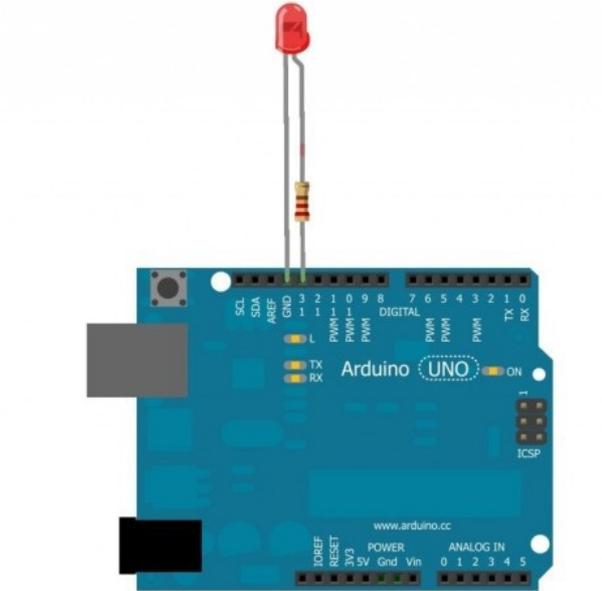


Ilustración 80: conexión de un diodo LED al Arduino

Esquema eléctrico

- Todo el circuito eléctrico general correspondiente a las barreras de iluminación se muestra en la siguiente imagen y ha sido diseñado con AutoCAD.

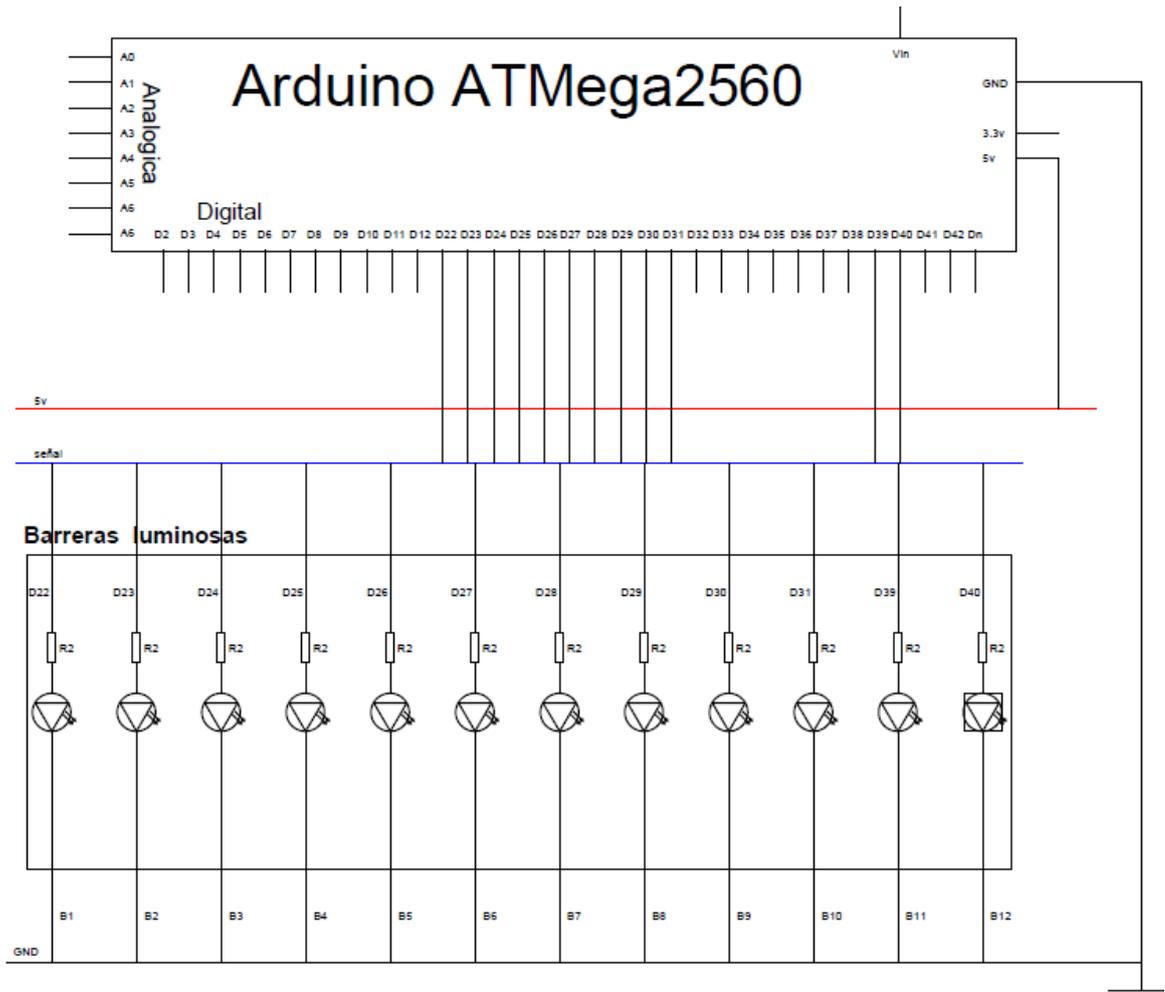


Ilustración 81: Esquema eléctrico de las barreras de iluminación

De donde:

- **R_n**: resistencias
- **A_n**: pines analógicos del Arduino
- **D_n**: pines digitales del Arduino
- **B_n**: número de barrera
- **Línea roja**: tensión 5V
- **Línea azul**: señal conectada a los pines
- **Línea negra inferior**: tierra

✚ **Nota:** La resistencia en serie se conecta a la patita del ánodo para no fundir el diodo

- Observando de manera independiente una única barrera luminosa se puede ver que lleva el siguiente circuito eléctrico que será el mismo para el resto de barreras luminosas.

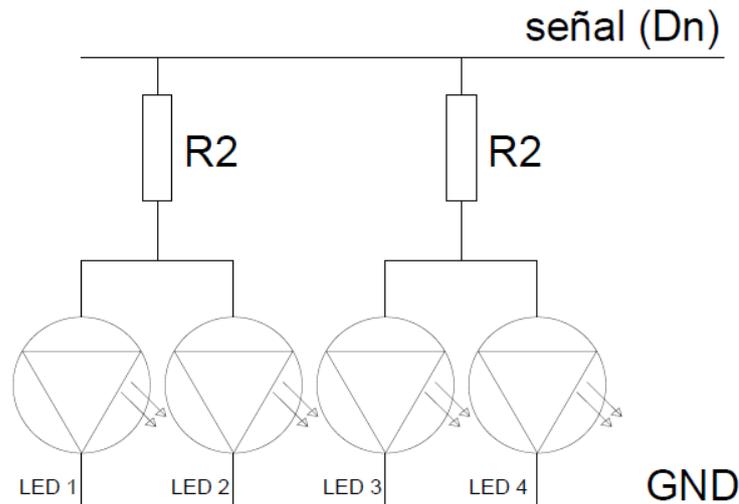


Ilustración 82: esquema eléctrico de una sola barrera de iluminación

- A continuación se va a ilustrar el circuito montado en el entorno de pruebas del proyecto.

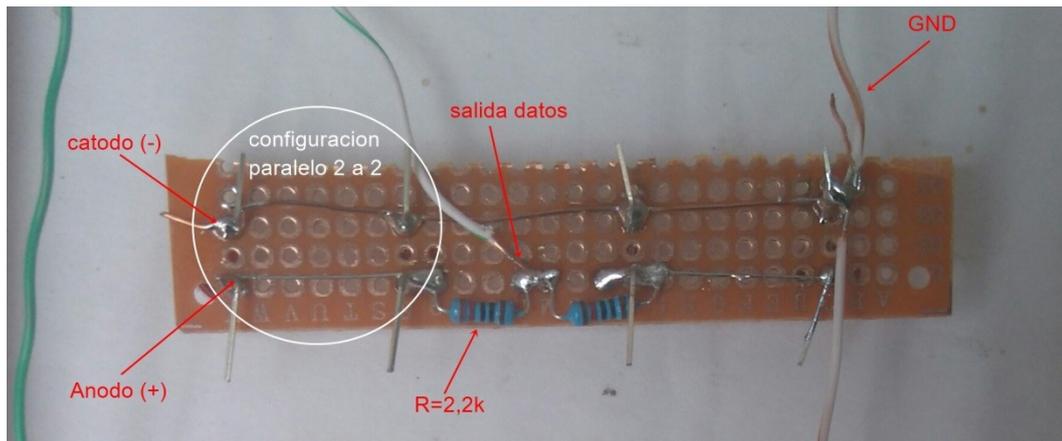


Ilustración 83: aspecto físico de una única barrera luminosa

La configuración paralelo dos a dos es una configuración que consiste en conectar dos diodos LED en paralelo cuyos ánodos están conectados a una misma resistencia común y los cátodos conectados a tierra.

Justificación de la configuración: independientemente de que se adoptara esta configuración se podía haber conectado los cuatro LED que componen la barrera en paralelo y conectar los ánodos a una misma resistencia, pero, si se hiciera uso de esa configuración y teniendo en cuenta que el sistema está formado por diversas barreras, a todos los LED de una determinada barrera no les llegaría la misma intensidad e incluso habría algunos que no podrían iluminarse.

Dicho esto, con la configuración paralelo dos a dos, lo que se consigue es que la intensidad que llega a los dos LED conectados en paralelo sea básicamente la misma intensidad ya que se producirían menos **perdidas en el cobre** entre un LED y otro en comparación con las pérdidas que se producían entre el primer LED y el cuarto en caso de usarla otra confi-

guración. Ambos pares comparten la misma GND y las resistencias conectadas en la salida de datos tienen un punto común.

Es configuración también ayuda a que en caso de que se haya fundido una resistencia en el circuito no se apague toda la barrera ya que una parte de ella quedaría funcionando.



Ilustración 84: aspecto físico de Circuito completo de todas las barreras luminosas

4.4.3.1. Fragmento del código realizado Para barreras luminosas

Primero se declararon las variables y pines correspondientes a las barreras. Todas las salidas serán especificadas como *output*

```
// Barreras de iluminacion

int iluminaBarrera1=22;
int iluminaBarrera2=23;
int iluminaBarrera3=24;
int iluminaBarrera4=25;
int iluminaBarrera5=26;
int iluminaBarrera6=27;
int iluminaBarrera7=28;
int iluminaBarrera8=29;
int iluminaBarrera9=30;
int iluminaBarrera10=31;
int iluminaBarrera11=39;
int iluminaBarrera12=40;

// luces semaforos
int verde=10;
int ambar=11;
int rojo=12;

void setup() {
  lcd.begin(16,2);
```

```

Serial.begin(9600);
// luces semaforos
pinMode(verde,OUTPUT);
pinMode(ambar,OUTPUT);
pinMode(rojo,OUTPUT);
// iluminacion de las barreras
pinMode(iluminaBarrera1,OUTPUT);
pinMode(iluminaBarrera2,OUTPUT);
pinMode(iluminaBarrera3,OUTPUT);
pinMode(iluminaBarrera4,OUTPUT);
pinMode(iluminaBarrera5,OUTPUT);
pinMode(iluminaBarrera6,OUTPUT);
pinMode(iluminaBarrera7,OUTPUT);
pinMode(iluminaBarrera8,OUTPUT);
pinMode(iluminaBarrera9,OUTPUT);
pinMode(iluminaBarrera10,OUTPUT);
pinMode(iluminaBarrera11,OUTPUT);
pinMode(iluminaBarrera12,OUTPUT);

}

```

A continuación se mostrará algunas funciones creadas para realizar la activación y desactivaciones de las barreras. El total de funciones que se debía realizar eran doce y todas ellas se podrán ver en el código completo del programa *apartado 4.6.1*.

```

void iluminacionBarrera0(){
    digitalWrite(iluminaBarrera12,LOW);
    delay(20);
    digitalWrite(iluminaBarrera11,LOW);
    delay(20);
    digitalWrite(iluminaBarrera10,LOW);
    delay(20);
    digitalWrite(iluminaBarrera9,LOW);
    delay(20);
    digitalWrite(iluminaBarrera8,LOW);
    delay(20);
    digitalWrite(iluminaBarrera7,LOW);
    delay(20);
    digitalWrite(iluminaBarrera6,LOW);
    delay(20);
    digitalWrite(iluminaBarrera5,LOW);
    delay(20);
    digitalWrite(iluminaBarrera4,LOW);
    delay(20);
    digitalWrite(iluminaBarrera3,LOW);
    delay(20);
    digitalWrite(iluminaBarrera2,LOW);
    delay(20);
    digitalWrite(iluminaBarrera1,LOW);

}
void iluminacionBarrera1(){

    digitalWrite(iluminaBarrera1,HIGH);
    delay(retardol);
    digitalWrite(iluminaBarrera6,LOW);
    delay(retardol);
    digitalWrite(iluminaBarrera5,LOW);
    delay(retardol);
    digitalWrite(iluminaBarrera4,LOW);

```

```

        delay(retardo1);
        digitalWrite(iluminaBarrera3,LOW);
        delay(retardo1);
        digitalWrite(iluminaBarrera2,LOW);
    }

void iluminacionBarrera2() {
    digitalWrite(iluminaBarrera1,HIGH);
    delay(retardo1);
    digitalWrite(iluminaBarrera2,HIGH);
    delay(retardo1);
    digitalWrite(iluminaBarrera6,LOW);
    delay(retardo1);
    digitalWrite(iluminaBarrera5,LOW);
    delay(retardo1);
    digitalWrite(iluminaBarrera4,LOW);
    delay(retardo1);
    digitalWrite(iluminaBarrera3,LOW);
    delay(retardo2);
    digitalWrite(iluminaBarrera2,LOW);
}

```

4.4.4. Conexión de sensor LDR

Cuando se trata de medir valores del LDR se puede tener en cuenta dos configuraciones fundamentales, tal como se puede observar en la siguiente ilustración.

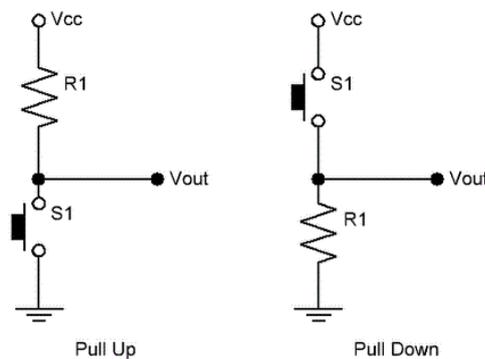


Ilustración 85: configuración pull-up y pull-down

Utilizando la configuración **Pull-up**, lo que se consigue es que la tensión de salida **Vout**, vaya aumentando a medida que el sensor capte más luz y por el contrario, la **Vout** disminuirá cuando haya falta de luz. En el caso contrario, con la configuración **Pull-down**, **Vout** aumentará cuando menos luz capte, y disminuirá cuanto más luz haya.

En la siguiente *Ilustración* se puede observar cómo se conecta un LDR a Arduino usando la forma de conexión **Pull-up**, debido a que es que preferimos introducir en nuestro sistema. Ya sea en un tipo de conexión u otra, la **Vout** siempre se conecta a una de las *entradas analógicas*, ya que el sensor LDR nos va a producir señales analógicas.

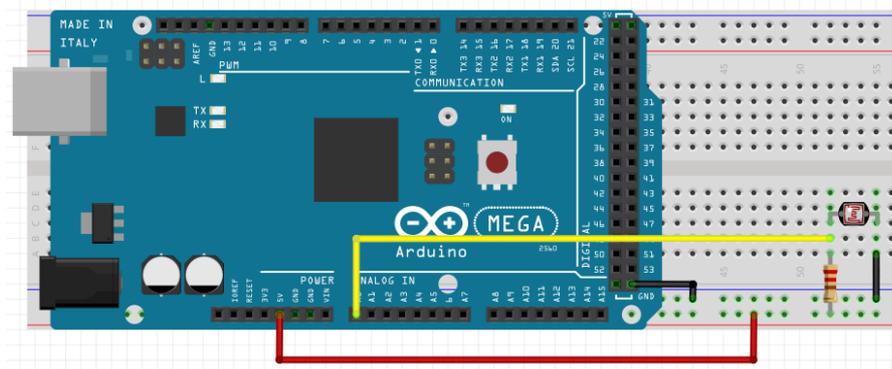


Ilustración 86: Conexión Pull-up del sensor LDR con Arduino ATmega2560

A continuación se ilustrará el circuito montado en prototipo siguiendo la configuración *pull-up* (activación del sistema a medida que oscurece.)

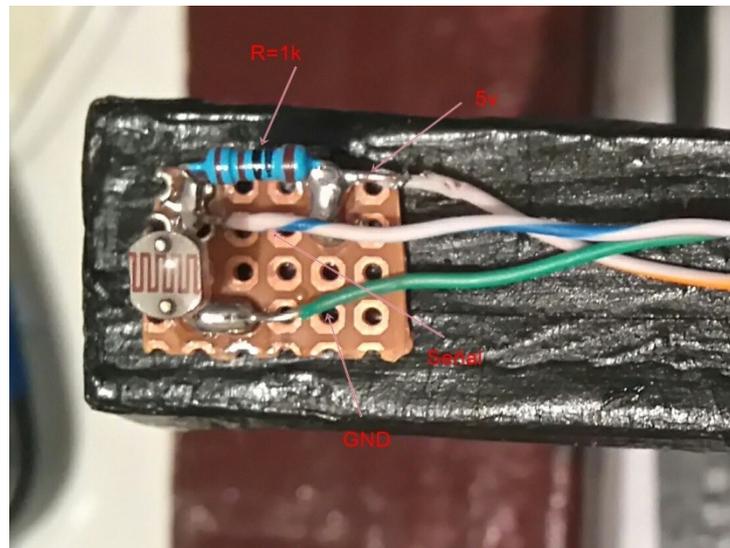


Ilustración 87: configuración pull-up en el prototipo

4.4.4.1. Fragmento del código realizado Para el sensor de luz

Primero se declaró la variable y su pin correspondiente. Las lecturas de este sensor se realizarán directamente en el *void loop*.

```
const int sensorLDR=A0;
int valorLDR;

void loop(){
  // se comprueba el estado del ambiente
  valorLDR=analogRead(sensorLDR);
  Serial.print("valor LDR: ");
  Serial.println(valorLDR);

  if (valorLDR<900){//mucho iluminación
    cambio_estados_dia(); // el semaforo funciona de manera
    tradicional.
  }
}
```

```

        lcd.setCursor(0,0);
        lcd.print("ESTADO: DIA");
        lcd.setCursor(0,1);
        lcd.print("VISIBILIDAD 100%");

    } // fin DIA

    else if(valorLDR>900){ // implicaría poca iluminación

        lcd.setCursor(0,0);
        lcd.print("ESTADO NOCTURNO");
        lcd.setCursor(0,1);
        lcd.print("POCA VISIBILIDAD");

        inicioDelControl();

        lcd.clear();
        situacionEnEstadoRojo();
        lcd.clear();

    } // fin sistema nocturno

} // fin del loop

```

Este fragmento que arriba se muestra es de toda la programación del sistema, pero solo hay que fijarse en las líneas donde aparece *sensorLDR* y *valorLDR* y ver como ha sido codificado

4.4.5. Conexión del sensor piroeléctrico PIR o de presencia

Al igual que se ha hecho con los demás componentes electrónicos en este proyecto, toca explicar ahora el modo de conexión del sensor PIR HC-SR501 al sistema, como una de las partes fundamentales para la toma de información en el exterior y que pueda garantizar un correcto funcionamiento del sistema según los objetivos previamente ya expuestos.

El sensor PIR se suele conectar de la manera digital al controlador, ya que consta de dos estados de funcionamiento binario 0 y 1 que equivaldrían a **No presencia** (0) y **Si presencia** (1). Cuenta con solo 3 terminales, uno de los cuales se utilizan para alimentación (5v) otro para tierra (GND) y el tercero que normalmente suele ser el pin del medio para la salida o señal (3.3v) como se muestra en la siguiente *ilustración*.

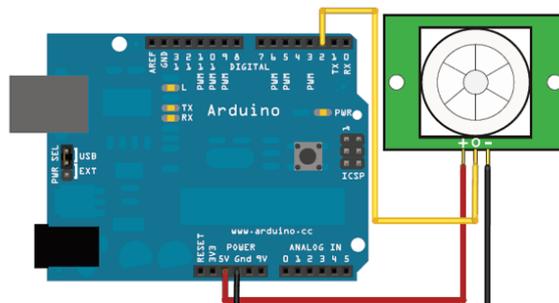


Ilustración 88: esquema de conexión de un sensor PIR a placa Arduino

✚ **Nota:** no es necesario conectarle una resistencia en serie con el pin de 5V porque a que ya trae una resistencia integrada dentro de su circuito

También dispone de dos resistencias variables de calibración, CH1 y RL2. Con *CHI* se puede controlar el tiempo que está activa la salida del sensor, y con RL2, permite ajustar la distancia de detección.

Configuración de puentes del sensor PIR

Tabla 6: Configuración del sensor PIR

Símbolo	Descripción
L	La salida pasa alto y luego bajo cuando se activa. Resultados de movimiento continuo en repetidos Pulsos de alta / baja. El caudal es bajo cuando está inactivo.
H	La salida permanece alta cuando el sensor se reactiva repetidas veces. El caudal es bajo cuando está inactivo (no activado)

Calibración del detector

El sensor PIR requiere un tiempo de 'warm-up' para poder funcionar correctamente al energizarse. Esto es debido al tiempo de establecimiento involucrado en el "aprendizaje" de su entorno. El tiempo que puede transcurrir en el aprendizaje de su entorno oscila entre 10 a 60 segundos. Durante este tiempo la sensibilidad de detector va aumentando a medida que el campo de visión del módulo va tomando datos del entorno y es recomendable la ausencia de personas durante la calibración.

Dimensiones del módulo

En la siguiente imagen se ilustra las dimensiones mecánicas que pueden tener sensores de este tipo. Se refleja la altura, el ancho y largo medidos en mm.

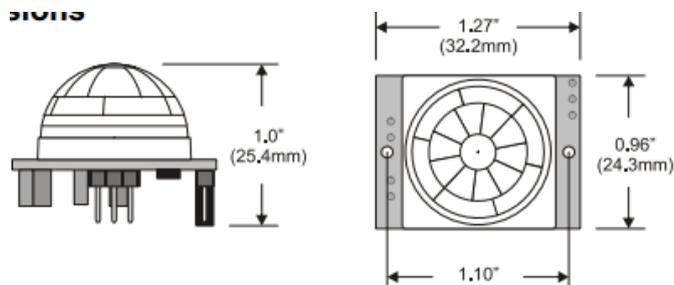


Ilustración 89: medidas mecánicas de un sensor PIR

Modo de operación

Los dispositivos piroeléctricos como el PIR, poseen elementos fabricados de un material cristalino que genera una carga eléctrica cuando se expone a la radiación infrarroja. Los cambios en la cantidad de radiación producen cambios de voltaje los cuales son medidos

por el amplificador. El PIR trae consigo unos filtros especiales llamados **lentes de Fresnel** que enfocan las señales infrarrojas sobre el elemento sensor. Cuando las señales infrarrojas del ambiente cambian rápidamente, el amplificador activa la salida para indicar el movimiento y esta salida permanece activada durante unos segundos permitiendo al microcontrolador saber si hubo movimiento.

Esquema eléctrico

Todo el circuito eléctrico correspondiente al sensor LDR, sensor PIR y a los estados del semáforo se muestra en la siguiente imagen.

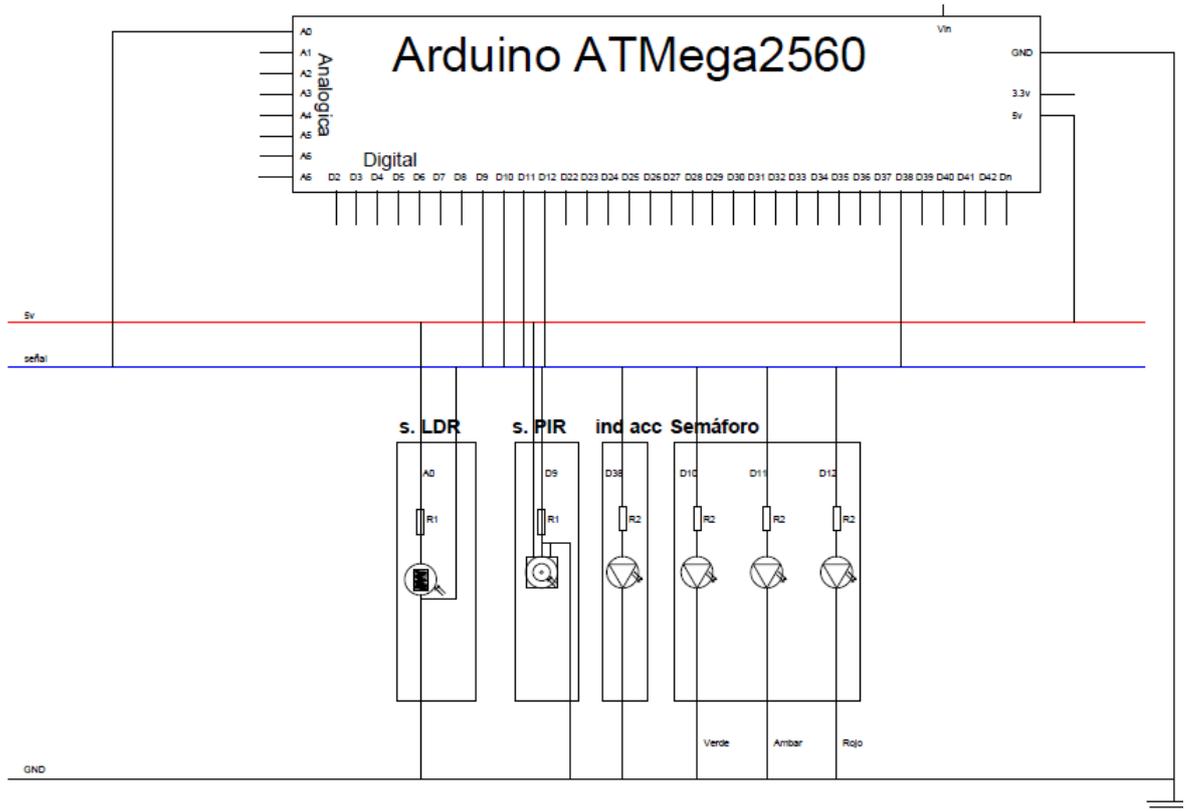


Ilustración 90: Esquema eléctrico de los sensores PIR, LDR y estados del semáforo

De donde:

- **R_n**: resistencias
- **An**: pines analógicos del Arduino
- **D_n**: pines digitales del Arduino
- **Línea roja**: tensión 5V
- **Línea azul**: señal conectada a los pines
- **Línea negra inferior**: tierra

4.4.5.1. Fragmento del código realizado para el sensor de movimiento

Primero se declaró las variables y asignarles sus pines correspondientes y especificarlo como una entrada *input* en el *void setup*. Con el fin de poder realizar las lecturas se creó una función ya que esta operación se iba a realizar constantemente en todo el sistema.

```

// sensor de presencia
const int sensorPIR=9;
int estadoPIR;

void setup() {
  lcd.begin(16,2);
  Serial.begin(9600);

// sensor de presencia
  pinMode(sensorPIR, INPUT);
}

void informacionPir() {
  estadoPIR=digitalRead(sensorPIR);
  if(estadoPIR==HIGH) {
    lcd.setCursor(0,1);
    lcd.print("PIR:");
    lcd.print("SI");
  }
  else{
    lcd.setCursor(0,1);
    lcd.print("PIR:");
    lcd.print("NO");
  }
}
}

```

4.4.6. Conexión de la pantalla LCD con un potenciómetro

Gracias a la utilización de una placa Arduino con muchos pines y al no disponer de un **modulo I2C** que se suele acoplar a la pantalla para ahorrar pines, la instalación de la pantalla LCD se ha realizado de la manera más conocida y directa. Utilizando los respectivos pines con una conexión de 4 bits ya que eran suficientes para mostrar toda la información necesaria.

En la siguiente ilustración se muestra el esquema eléctrico utilizado para su conexión.

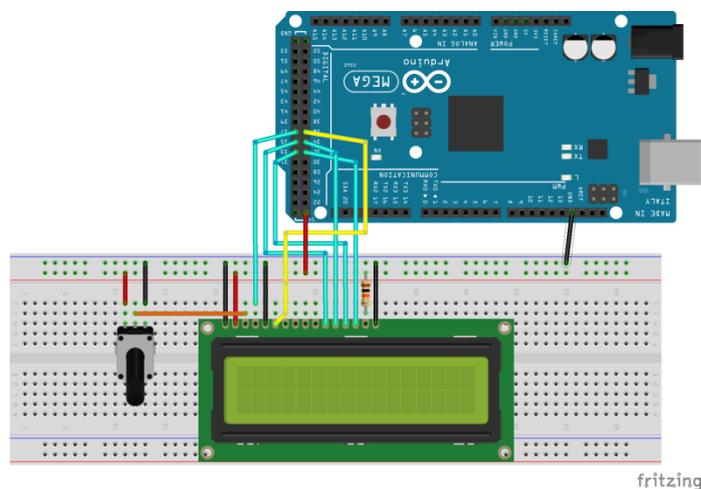


Ilustración 91: conexión de una pantalla LCD

La utilización de un potenciómetro en este esquema, tiene como finalidad regular el contraste en la pantalla, ya que sin él sería imposible apreciar los caracteres enviados a la pantalla para poder ser visualizados.

En el código de la programación se mostrara la declaración de todos los pines utilizados.

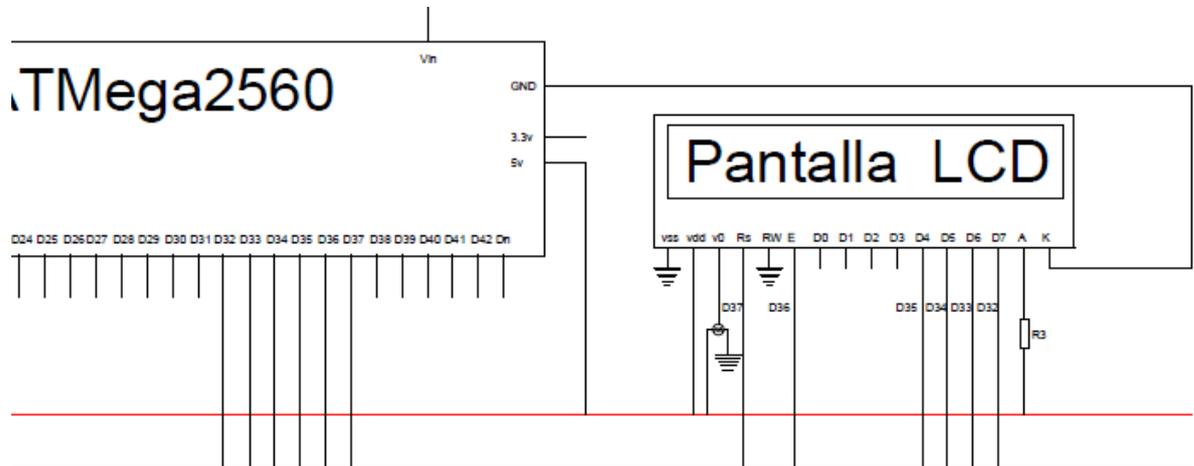


Ilustración 92: Esquema eléctrico de conexión de una pantalla de display LCD

De donde:

- **Rn**: resistencias
- **An**: pines analógicos del Arduino
- **Dn**: pines digitales del Arduino
- **Línea roja**: tensión 5V
- **Línea azul**: señal conectada a los pines
- **Línea negra inferior**: tierra

A continuación se muestra una imagen de la pantalla LCD ubicado en una de las calzadas con su respectivo potenciómetro.



Ilustración 93: pantalla LCD con un potenciómetro regulador

4.4.6.1. Fragmento del código realizado Para la pantalla LCD

Lo primero que se hace es llamar a la librería creada para la utilización de una pantalla display y posteriormente se declara los pines que serán utilizados y las dimensiones o el tipo de pantalla que se va a utilizar en el *void setup*.

```
#include <LiquidCrystal.h>

// declaración de los pines a utilizar
LiquidCrystal lcd(37,36,35,34,33,32);

void setup() {
  lcd.begin(16,2); // pantalla de de dos filas de 16 columnas

  lcd.print("SIST. D REDUCCION");
  lcd.setCursor(0,1);
  lcd.print("  DE ATROPELLOS");
  delay(2500);
  lcd.clear();
}
}
```

En este sistema se creó una función que mostrarían toda la información que se pretendía visualizar en la pantalla y que era de suma importancia.

```
void informacionOtrosParametros() {

  lcd.setCursor(0,0);
  lcd.print("V:");
  lcd.print(velocidadCalculada1);
  lcd.print("m/s");

  lcd.setCursor(11,0);
  lcd.print("D:");
  lcd.print(distanciaPuntual);
  lcd.print("cm");

  lcd.setCursor(8,1);
  lcd.print("Pt:");
  lcd.print(tramo);
  lcd.print("s");
}
}
```

✚ **Nota:** la codificación para mostrar la información del sensor de presencia viene ya dentro de la función *informacionPir()* en el apartado 4.4.5.1

4.5. Regulador PID y funcionamiento del sistema

Llegado a este punto, se va entrar en detalle de lo que viene a ser la descripción del funcionamiento de de todo el sistema, recorriendo todos los aspectos y reglas que hacen posible dicho proceso. Para ello se partirá diseñando un modelo matemático sencillo y fácil de interpretar ya que abarcará en gran mayoría todo los pasos que debe seguir el sistema para cumplir con los objetivos marcados en el *capítulo 2* de este proyecto.

Lo primero que se hará en este proceso va a ser diseñar un diagrama de bloques simplificado y que refleja de manera superficial todos los componentes que intervienen en el sistema. Ya más abajo se presentará un diagrama de flujo que reflejará toda la complejidad del sistema.

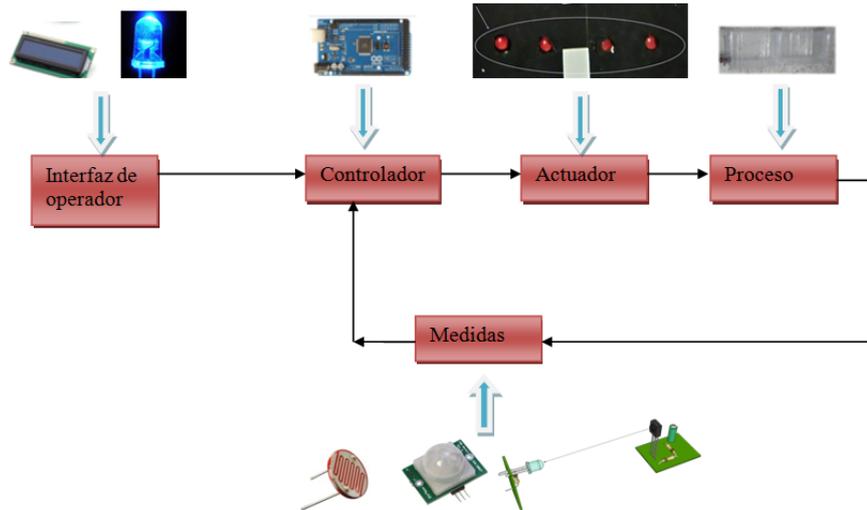


Ilustración 94: diagrama de bloques del sistema

- **Interfaz de operador.** Para el interfaz de operador se dispone de una pantalla *display* LCD para visualizar la información proporcionado por el controlador. También se dispone de un LED indicador (color azul) que luce cuando una de las barreras infrarrojas ha sido cortada. Con ese indicador se sabrá si alguna barrera está funcionando correctamente o no.
- **Controlador.** Como ya es sabido este será el cerebro que procesará todas las funciones del sistema, ya sean de cálculo, toma o envío de información.
- **Actuador.** El único actuador existente en este sistema son las líneas de iluminación o barreras luminosas que se deberían activar en función de las decisiones tomadas por el controlador.
- **Proceso.** El proceso va a ser todo el código fuente realizado en la programación del controlador. En él se establecerán todos los cálculos y decisiones a partir de las cuales se deberá tomar una salida adecuada en el sistema.
- **Medidas.** Son los instrumentos utilizados para la toma de información en el sistema. Entre ellos tenemos: sensor PIR, sensor LDR, emisor-receptor de infrarrojos

4.5.1. Regulador PID para el control de la velocidad

No se trata de elaborar un sistema que realice funciones básicas e ideales, es decir, libre de errores sino de tener un sistema que abarque todos los problemas que se puedan dar durante el proceso. No se pretende elaborar un sistema condicionado ni secuencial sino uno que estudie, aprenda y analice el comportamiento del vehículo en cada tramo en que se haya hecho una toma de datos con el fin de proporcionar una salida adecuada. Para este proyecto se ha decidido utilizar un regulador proporcional integral derivativo PID, posteriormente se justifica los motivos.

Justificación: la razón por la que se ha optado utilizar un regulador PID es el sistema viene descrita a continuación.

Al tratarse de un sistema cuyo factor más importante es la de proporcionarle a los conductores la información suficiente (*iluminación de las barreras*) para regular su velocidad una vez esté dentro del área de control y teniendo en cuenta que nuestro sistema no tiene acceso directo sobre los automóviles, es decir, no puede acceder sobre el manejo y control de esos para regular su velocidad, por lo tanto, la función principal de este regulador PID será *calcular y/o estimar la velocidad* del vehículo y *el error producido* debido a esa variación de esta velocidad.

Es posible que esta serie de cálculos se pueda hacer de manera directa o utilizando otros métodos, pero lo que se conseguirá con el PID es conocer el error producido en un determinado tramo y a partir de este error calcular una respuesta mediante su expresión matemática. Este dato y junto con otros parámetros secundarios (*mirar la evolución de la velocidad entre dos tramos, comprobar la presencia o ausencia peatonal en el paso de peatones*) tomar decisiones.

4.5.1.1. Comportamiento del PID

El funcionamiento de un regulador PID se podría dividir en dos bloques: por una parte se tiene que si la señal de error varía lentamente teniendo como punto de partida el valor permitido de la velocidad predominará la acción **proporcional e integral (PI)** y la respuesta que se obtenga se adaptará a esta variación; pero si la señal de error varía bruscamente predominará la acción **derivativa (D)** y la respuesta que se obtenga en la salida también adoptará un comportamiento rápido.

Es necesario aclarar que en este proyecto, no se pretende elaborar un sistema que ayude a regular la velocidad a valores permitidos de manera constante hasta el final del recorrido, sino de elaborar un sistema que permita a los conductores tener una visión sobre su estado y desacelerar de manera gradual y controlada hasta el final del trayecto (*línea de parada en el semáforo*) gracias a la información que el sistema les proporciona.

4.5.1.2. Elaboración del PID

Para la elaboración de éste controlador se empezará definiendo los parámetros que hay que tener en cuenta.

- **Distancia entre tramos (Dt):** se define como distancia entre tramos a la separación entre dos barreras infrarrojas. Con la ayuda de esta distancia y gracias a los conocimientos adquiridos en física se calculará la velocidad en cada tramo. Es necesario que esta **Dt** tenga un valor fijo en todos los tramos con el fin de optimizar los cálculos.
- **Periodo entre cortes (Tc):** el periodo de cortes, viene a ser definido como el periodo de tiempo transcurrido desde que el automóvil atravesó dos barreras consecutivas. Al igual que la **Dt**, el **Tc** también servirá como dato para el cálculo de la velocidad del automóvil.
- **Velocidad normal (Vn):** la velocidad normal será definida como el valor de la

velocidad permitida en un tramo determinado. Partiendo de esta velocidad se estudiará la evolución de la velocidad de automóviles y a partir de ella se sacarán las conclusiones adecuadas.

Es necesario destacar que, dentro del sistema diseñado este valor puede ser modificado dependiendo de la zona que se quiere controlar teniendo en cuenta velocidad permitida en esta zona.

***Por ejemplo:** si en este proyecto se está realizando el estudio en una zona donde la velocidad máxima permitida es de 50km/h, en caso de realizar un estudio en una zona diferente, bastaría con asignar dentro del programa el nuevo valor de la velocidad permitida en dicha zona a partir de cual se realizarían las futuras comparaciones en función de los valores calculados por el regulador.*

➤ **Velocidad calculada (V_c):** la velocidad calculada V_c será la velocidad estimada o viajaba un determinado automóvil una vez que haya atravesado un determinado tramo.

Para llegar a conocer este valor de la velocidad se basará en las leyes de la física y se aplicará conceptos de la cinemática, tomando como datos la D_t y el T_c .

Partiendo de que

$$S = V \cdot t \quad (14)$$

Ecuación 14: fórmula cinemática del espacio

Al desconocer la aceleración del automóvil se considera que es **constante** y se obtiene que

$$V_c = \frac{D_t}{T_c} \quad (15)$$

Ecuación 15: fórmula de la velocidad con aceleración constante

Siendo $\begin{cases} D_t : \text{la distancia entre tramos} \\ T_c : \text{el periodo de corte entre barreras} \end{cases}$

Dado que la velocidad calculada en el prototipo viene expresada en centímetros y es muy pequeña, la solución aplicada fue transformar esta velocidad a un valor de simulación real. Para ello se multiplicará esa velocidad por una constante K que normalmente será una ganancia calculada de la siguiente forma.

Si definimos que

$$V_n = V_c \cdot k \quad (16)$$

Ecuación 16: fórmula de la velocidad normal de un tramo de control

Eso implicará que

$$k = \frac{V_N}{V_c \cdot 100} \quad (17)$$

Ecuación 17: ganancia para el incremento de la velocidad calculada

ejemplo practico: si se supone que en un tramo de control

$$V_n=50\text{km/h} \text{ o } 14\text{m/s} \text{ tendríamos que } V_c = \frac{0.07\text{m}}{2\text{s}} = 0.035\text{m/s}$$

$$T_c=2\text{s}$$

$$\text{Y por lo tanto para transformarlo tendríamos que } \gg k = \frac{14\text{m/s}}{0.035\text{m/s} \cdot 100} = 4$$

✚ **Nota** Esa ganancia calculada nos elevará el valor de la velocidad calculada a un valor mayor. Pero en el entorno real no haría falta realizar este cálculo y la velocidad real será la calculada. Pero para ello hay que establecer que la $K=1$

Tabla 7: valores de la ganancia K para diferentes zonas

Vn (km/h)	Tc (s)	Vc (m/s)	K
30	2	0.035	2.5
40	2	0.035	3
50	2	0.035	4
60	2	0.035	4.5
70	2	0.035	5

Gracias a esta V_c se pretenderá conocer la evolución de la velocidad en todo el tramo y ya después calcular el error producido durante dicha variación.

➤ **Error (Err):** se define el error producido como la diferencia entre el valor permitido y el valor calculado. Este viene a ser el parámetro fundamental a partir del cual el regulador PID adoptará una respuesta u otra dependiendo del valor del error calculado.

$$E_{rr} = |V_N - V_c| \quad (18)$$

Ecuación 18: Error absoluto

Para obtener dicho error en tanto por ciento se puede utilizar la siguiente expresión

$$E_{rr}(\%) = \frac{|V_N - V_c|}{V_N} \cdot 100 \quad (19)$$

Ecuación 19: Error absoluto en tanto porciento

Con el fin de proporcionar un mejor control en la regulación y ofrecer unas respuestas óptimas en el sistema, se ha dividido el valor del error en una serie de rango o intervalos mediante los cuales se irá comparando dichos valores respecto del error calculado.

- **Error permitido:** el error permitido será un intervalo comprendido entre **0 y 20%**. Siempre que el valor del error calculado este comprendido dentro de este intervalo no será necesario que el regulador proporcione una respuesta significativa (iluminación de las barreras.) En caso de que el error es negativo se considerará cero ya que esta situación se producirá siempre que la velocidad sea muy baja en comparación con la permitida.

Para un sistema de **tipo cero** el valor máximo del rango se puede calcular mediante la siguiente expresión.

$$e_p = \frac{1}{1 + K_p} \quad (20)$$

Ecuación 20: límite del error permitido k_p

Donde **K_p** es el valor de la constante proporcional y cuanto más grande sea su valor, menor será el límite del error y más sensible será el sistema para la detección de errores permitidos.

- **Error con evolución lenta:** el error con evolución lenta hace alusión a la parte proporcional integral (PI) del regulador y abarcará el intervalo de **20% a 50%**. Siempre que el valor de error calculado esté dentro de este intervalo significará que la velocidad calculada está variando de manera significativa con respecto a la velocidad permitida, por lo tanto, ha sido una evolución lenta se deberá proporcionar una salida que se adapte a este cambio.

Para un sistema de **tipo cero** el valor máximo del rango se puede calcular mediante la siguiente expresión.

$$e_{PI} = \frac{1}{1 + K_p \cdot K_i} \quad (21)$$

Ecuación 21: límite del error con evolución lenta k_p, k_i

Donde **K_p** es el valor de la constante proporcional y la **K_i** la constante velocidad cuanto más grande sean sus valores, menor será el límite del error y más sensible será el sistema para la detección de errores con evolución lenta.

- **Error con evolución rápida:** este va a ser error producido superior a **50%** e decir que se ha producido una evolución muy rápida de velocidad calculada con respecto de la velocidad normal, por lo tanto, el regular deberá adoptar un comportamiento Derivativo (D) para proporcionar una respuesta que se adapte a este cambio brusco.

Para un sistema de **tipo cero** el valor máximo del rango se puede calcular mediante la siguiente expresión.

$$e_D = \frac{1}{K_d} \quad (22)$$

Ecuación 22: límite del error con evolución rápida, kd

Donde K_d es la constante aceleración y cuanto más grande sea su valor, menor será el límite del error y más sensible será el sistema para la detección de errores con evolución rápida.

Ejemplo práctico: para un sistema cuyos parámetros son $K_p=3$, $K_i=1$ y $K_d=2$ Tendremos que:

$$e_p = \frac{1}{1 + 4} = 0.2 \text{ ó } 20\%$$

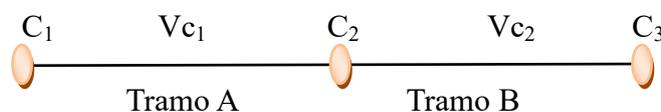
$$e_{PI} = \frac{1}{1 + 4 * 1} = 0.2 \text{ ó } 20\%$$

$$e_D = \frac{1}{2} = 0.5 \text{ ó } 50\%$$

Esos valores establecidos para el cálculo de los límites de errores, son modificables según el grado de sensibilidad que se desea integrar en el sistema, por lo tanto, cuando más grandes sean esos valores mayores será el grado de sensibilidad que puede adoptar el regulador para la toma de decisiones.

Otro de los factores muy importantes a tener en cuenta en este sistema consiste en que, para conseguir una regulación óptima y fluida no solo pueden basarse en valores del error calculado y compararlos con los intervalos preestablecidos. Es necesario tener en cuenta otros parámetros para la toma de decisiones en el sistema. Estos parámetros podrían ser las siguientes:

- **Detector de movimiento:** la toma de información en este sensor es fundamental ya que la ubicación de este dispositivo se encuentra sobre la zona de cruce peatonal y es necesario tener constancia en todo momento de los movimientos que se pueden producir o no sobre esta zona y proporciona dicha información al controlador. Así junto con el error y los otros parámetros tomar una decisión definitiva que proporcione una salida adecuada.
- **Variación de la velocidad entre tramos ΔV :** no solo se puede tener en cuenta la variación de la velocidad en un tramo respecto de la permitida, sino que también se puede hacer un análisis de la evolución de la velocidad del automóvil al pasar entre dos tramos continuos. Esta información permite conocer si éste está acelerando o desacelerando su velocidad con respecto a la velocidad del tramo anterior. Gracias a este nuevo dato y junto con otros parámetros ya mencionados, la salida es a cada vez más precisa y compleja.



Siendo:

V_{c1} la velocidad calculada en el primer tramo o tramo A al pasar por los cortes C_1 y C_2 .

V_{c2} es la velocidad calculada o estima a lo largo del recorrido del tramo 2 o tramo B justo antes de hacerse el corte C_3 .

Normalmente esta velocidad V_{c2} es una velocidad ficticia, ya que la verdadera velocidad real del tramo B se obtiene una vez realizado los cortes C_2 y C_3 . La obtención de esta velocidad se realiza por suposición o por regla de tres simple, teniendo como datos la velocidad V_{c1} , el tiempo de corte T_{c1} y tiempo transcurrido (T_{c-2}) desde que se realizó el corte C_2 .

La expresión para obtener esta velocidad ficticia estimada es:

$$V_{c2} = \frac{V_{c1} \cdot T_{c1}}{T_{c-2}} \quad (23)$$

Ecuación 23: velocidad ficticia

Siendo:

T_{c1} : el periodo transcurrido en el primer tramo.

V_{c1} : velocidad real calculada en el primer tramo

T_{c-2} : tiempo transcurrido desde que se inició el segundo corte. Normalmente al ser un valor que varía constantemente hasta que se realiza el otro corte, se suele obtener mediante la utilización de un contador que vaya tomando el tiempo y con ello realizar las operaciones.

Ejemplo práctico: si se considera que la velocidad normal permitida en una zona de control es de 50km/h y la velocidad calculada en un tramo A es de 70 km/h, calcular a lo largo del segundo tramo B justo antes de finalizarlo si el tiempo T_{c-2} es de 2.5 segundos y T_{c1} es de 2 segundos. Hacer el mismo cálculo para V_{c1} igual a 45km/h, T_{c-2} es de 1.9 segundo y T_{c1} es de 3 segundos.

Datos

$V_n=50\text{km/h}$

$V_{c1}=70\text{km/h}$, $T_{c1}=2\text{s}$, $T_{c-2}=2.5\text{s}$

$V_{c1}=45\text{km/h}$, $T_{c1}=3\text{s}$, $T_{c-2}=1.9\text{s}$

Solución

a)

$$V_{c2} = \frac{70\text{km/h} \cdot 2\text{s}}{2.5\text{s}} = 56\text{km/h}$$

b)

$$V_{c2} = \frac{45\text{km/h} \cdot 3\text{s}}{1.9\text{s}} = 47.36\text{km/h}$$

Conclusión: viendo ambos resultados se puede ver que la velocidad **Vc2** calculada desde que el vehículo entro en tramo B hasta que se produjo el siguiente corte iba disminuyendo saliendo de 70km/h a 56km/h y en el segundo tramo sucede casi lo contrario, e decir, la velocidad va aumentando saliendo de 45km/h a 47,36km/h.

Como se dijo anteriormente esas velocidades Vc2 son ficticias ya que solo se estiman para evitar desconocer el comportamiento del vehículo en un tramo que sigue recorriendo. Por lo tanto la velocidad real de este tramo se obtendrá una vez realizada el siguiente corte.

Tabla 8: salidas del regulador con respecto a la variación de la velocidad entre tramos

Comportamiento presente	Comportamiento pasado	Tipo de Error	Regulador PID	Salida
V_{c2}>V_{c1}	V _{c1} <V _N	Error con evolución rápida	D	Mucha iluminación
	V _{c1} >V _N	Error con evolución lenta	PI	Iluminación media
V_{c2}<V_{c1}	V _{c1} <V _N	Error permitido	-	Sin iluminación
	V _{c1} >V _N	Error permitido	-	Sin iluminación

➤ **Distancia del último corte (D_c):** este es un parámetro permite conocer el punto exacto donde se realizó el último corte con el fin de determinar la distancia a la que se sitúa el automóvil antes de llegar al semáforo. Este parámetro junto con el resto de parámetros ya mencionados se conseguiría un control óptimo sobre todo el sistema.

La expresión para calcular la distancia del último corte es la siguiente:

$$D_c = D_T - (n - 1) \cdot D_t \tag{24}$$

Ecuación 24: Distancia de corte

Siendo:

D_T: la distancia total desde la primera barrera hasta la última barrera de corte.

D_t: la distancia entre tramos que normalmente suele ser constante.

n: el numero de barrera, que varía desde la primera barrera o barrera de acceso hasta la última.

Ejemplo práctico: para un zona de control cuya distancia máxima es de 50m si la separación entre barreras es de 7 metros, calcular la distancia a la que estará el vehículo si se producen cortes 2, 4, 7 etc.

Datos

$D_T=50m$, $D_t=7$ y $n=2, 4, 7$

Por lo tanto aplicando la fórmula para el cálculo de distancias $D_c = D_T - (n - 1) \cdot D_t$

$$D_{c2} = 50 - (2 - 1) \cdot 7 = 43m$$

$$D_{c4} = 50 - (4 - 1) \cdot 7 = 29m$$

$$D_{c7} = 50 - (7 - 1) \cdot 7 = 8m$$

Conclusión: las tres distancias calculadas son las distancias a las que se sitúan las barreras infrarrojas dos, cuatro y siete.

4.5.1.2.1. Fragmento del código del regulador y otras operaciones relacionadas

El primer lugar se declaró todos los parámetros y variables necesarios en el sistema.

```
// variables que captura el tiempo de corte en cada barrera infrarroja

double T1=0;
double T2=0;
double T3=0;
double T4=0;
double T5=0;
double T6=0;
double T7=0;

// tiempo transcurrido en cada tramo,
double tramo=0;

// velocidad preestablecida del tramo de control
int velocidadNormal=14; //m/s o 50km/h
int K=2; // Ganancia para simular la velocidad real de un coche
K=Vn/Vp

// velocidades calculadas
double velocidadCalculada1;
double velocidadCalculada2;

// Variables del error
double Error;
double Kp=4;
double Ki=1.5;
double Kd=2;
double ErrorMin;
double ErrorMax;

// los valores de Kp , Ki, Kd son modificables en caso de cambiar las
condiciones del regulador

int distanciaDefinida=42; //cm
int distanciaPuntual;;
int distanciaTramo=7;
```

```
// duración de los estados del semaforos

int DuracionMaximaRojo=6000;
int DuracionMaximaVerde=8000;
int retardo2=300;// milisegundos para el blink
int retardo1=40;// milisegundos para encender barres
```

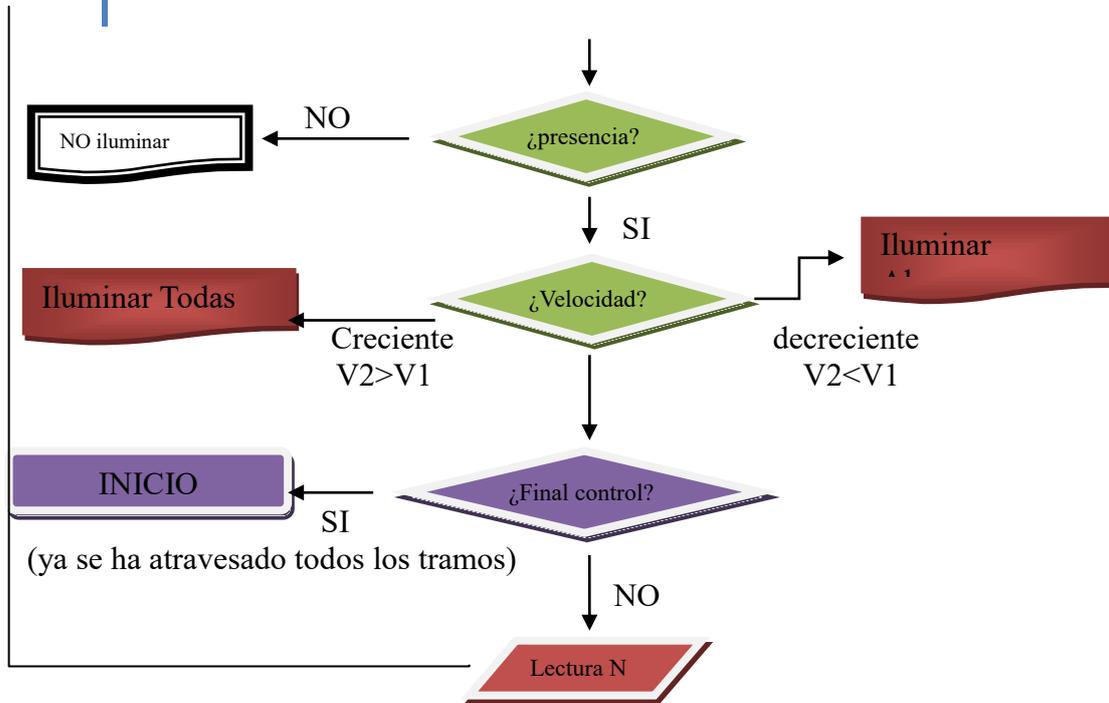
Posteriormente se creó una función que determina el *error* producido.

```
void calculoError() {

    /*Intervalos preestablecidos sobre el que actuara el regulador
    Error permitido=<=0,2 o 20%
    Error actuación PI: 0.201<=Error<0.6 ó 20,01%-60%
    Error actuacion D: Error>0.6 ó 60%
    */
    int ouput=0;
    ErrorMin=1/(1+Kp*Ki);
    ErrorMax=1/Kd;
    Error=(-1*(velocidadNormal-velocidadCalculada1)/velocidadNormal);
    Output=velocidadNormal*Error*(1/(1+Kp*Ki))+VelcidadCalculada1*Error
*(1/Kd);
}
```

4.5.2. Funcionamiento del sistema

Antes de explicar verbalmente como funciona conjuntamente el sistema diseñado, se mostrara primero un diagrama de flujo orientativo a partir del cual se irá detallando paso a paso cada proceso.



4.5.2.1 Descripción funcional del regulador de la velocidad en el sistema.

Como ya se ha mencionado en varias ocasiones, el funcionamiento del control de la velocidad de los automóviles, está pensado en un control PID. Recordando como viene caracterizada la expresión de un regulador PID en el *apartado 3.6.6*

$$y(t) = Kp e(t) + \frac{Kp}{Ti} \int_0^t e(t) dt + Kp Td \frac{de(t)}{dt}$$

Con el fin de regular los procesos en el sistema, se utilizaron principalmente tres parámetros, que pueden ser **Kp** (constante proporcional, que en otros controladores suele estar expresado en Banda proporcional $BP=100/KP$), **Ki** (constante integral) que es igual a $\frac{Kp}{ti}$ y **Kd** (constante derivativa) y por consiguiente tres acciones, que son la acción proporcional Kp que es igual a $Kp \cdot e(t)$, la acción integral que equivale a $\frac{Kp}{Ti} \int_0^t e(t)$ y la acción derivativa que equivale a $Kp \cdot Td \frac{de(t)}{dt}$.

A continuación se va a explicar el control y o regulación de la velocidad en el sistema. Pero antes es necesario aclarar que el regulador PID no controla todo el sistema para proporcionar una salida única sino que también existen otros parámetros que intervienen en la decisión final.

En función del error calculado por el controlador PID que en nuestro caso será el Arduino ATMmega2560, a partir del valor permitido de la velocidad V_N en el proceso, la respuesta

del sistema será una o u otra. Siempre que error sea muy pequeño o este comprendido dentro del rango de valores de error permitido (*apartado 4.5.1*) no será necesario que el regulador actúe sobre las barreras, independientemente de las otras condiciones declaradas en el sistema debido a que se calificaría el comportamiento del automóvil como funcionamiento normal y fuera de peligro.

Cuando el error sea grande, la variable calculada de la velocidad estará muy lejos de la variable permitida, la acción proporcional calculará una señal proporcional a la señal de error y en ese instante el controlador enviará una señal de salida analógica proporcional a ese cálculo al regulador para que actúe sobre las barreras luminosas, pero esperando que previamente se comprueben las otras variables que intervienen en la toma de decisión final para la activación de una salida definitiva.

Si se comprueba que la velocidad calculada se asemeja al valor permitido o que está ligeramente por debajo o por encima, en ese momento entrará en funcionamiento la acción integral calculando de nuevo una nueva señal proporcional al tiempo en que el automóvil ha tardado en variar su velocidad y por lo tanto, el controlador volverá a enviar una salida proporcional a la suma de la acción proporcional y la acción integral al regulador para que actúe sobre las barreras luminosas y como siempre esperando a que se comprueben las otras condiciones del sistema a fin de tomar una decisión conjunta.

Es importante mencionar que la acción derivativa solo funcionará cuando haya un cambio brusco en el sistema que haga que la señal de error cambie de ser pequeña a ser muy grande, y dejará de funcionar cuando el sistema vuelva a estar regido por la zona del error permitido o error con evolución lenta (PI). En este caso la acción derivativa calculará una señal proporcional al tiempo en el que la velocidad ha cambiado respecto de la V_N (*velocidad normal*) o valor permitido. Una vez realizado este cálculo, el controlador enviará una señal proporcional a la suma de las tres acciones al regulador para que actúe sobre las barreras.

4.5.2.2 Descripción funcional de la detección de peatones en el sistema.

Haciendo memoria de lo que era un sensor piroeléctrico en el *apartado 4.2.3* se dijo que era un sensor de infrarrojo pasivo que mide cambios en los niveles de radiación infrarroja (IR) emitida por los objetos situados en su campo de visión.

La utilización de este dispositivo entra en funcionamiento como consecuencia de la segunda condición, que debe ser comprobada en el sistema antes de tomar una decisión para la salida. La comprobación de ausencia y/o presencia peatonal es fundamental e incluso se podría decir que es la fase más importante en nuestro sistema, que esta información proporcionada por esos dispositivos evitaría que se produzcan incidentes y salvar vidas humanas y optimizar el sistema.

El sensor PIR es un dispositivo de medida cuyo funcionamiento no depende ni está vinculado con el regulador PID para el cálculo de la velocidad pero teniendo en cuenta que este es un dispositivo con modo de funcionamiento binario (**no presencia 0 y presencia 1**) se puede controlar directamente desde el controlador Arduino y solo entrará en modo de operación una vez que el regulador PID ya haya calculado previamente el error en la

velocidad e independientemente del estado de semáforo. Es necesario que el sensor este bien situado y bien calibrado con el fin de evitar falsas lecturas y poner el peligro las vidas humanas.

A la hora de instalar un sensor de infrarrojos pasivo o piroeléctrico y se energiza por vez primera, este empezará a acostumbrarse a todas las posibles radiaciones infrarrojas del entorno que le rodea ya que todos los cuerpos que están a una temperatura superior a 0 grados kelvin emiten radiación infrarroja negativa, por lo tanto, esta radiación aumente a medida que aumenta la temperatura del cuerpo situado en el entorno.

Una vez realizado el estudio del entorno se mantendrá estable sin embargo, si un peatón intenta cruzar la intersección se producirá una variación en la radiación infrarroja del ambiente y por lo tanto el sensor detectará ese cambio y avisará al controlador.

4.5.2.3. Fragmento del código que describe el funcionamiento del sistema

Para programar todo el sistema se crearon dos grandes funciones que abarcaban todas las condiciones, cálculos y subfunciones creadas previamente, para determinar el comportamiento del sistema.

La primera función era la *situacionEnEstadoRojo()* y abarcaba todas las situaciones que se pueden presentar en el escenario.

La segunda función era *situacionDePeligro()* que es aquella en la que el semáforo esta en verde o ámbar. La única diferencia que tiene esta función respecto de la primera es que el sistema actuaría si se detecta peatones cruzando el paso de manera ilegal.

En las siguientes líneas de código se muestra cómo funciona el sistema en los 2 primeros tramos ya que a partir del tercer tramo todos los cálculos o condiciones establecidas en el sistema serán las mismas que en el tramo dos.

El *Void loop* que contiene el todo el programa es el que se muestra a continuación.

```
void loop() {
// se comprueba el estado del dia con el sensor LDR
  valorLDR=analogRead(sensorLDR);
  Serial.print("valor LDR: ");
  Serial.println(valorLDR);

  if (valorLDR<900){// mucha iluminación en el ambiente
    cambio_estados_dia(); // el semáforo funciona de manera tradicional.

    lcd.setCursor(0,0);
    lcd.print("ESTADO: DIA");
    lcd.setCursor(0,1);
    lcd.print("VISIBILIDAD 100%");

  }// fin DIA
}
```

```

else if(valorLDR>900){//poca iluminación en el ambiente o nocturno

    lcd.setCursor(0,0);
    lcd.print("ESTADO NOCTURNO");
    lcd.setCursor(0,1);
    lcd.print("POCA VISIBILIDAD");

    inicioDelControl();

    lcd.clear(); // dentro de esta función está la situaciónDeRiesgo
    situacionEnEstadoRojo();
    lcd.clear();
} //fin sistema nocturno

} // fin del loop

```

El modo de cómo esta programa unas de las dos funciones fundamentales del sistema es el que se muestra a continuación. La codificación es referida a la **barrera 1-2** que constituye el **tramo A** y a la **barrera 2-3** que constituyen al **tramo B**.

```

void situacionEnEstadoRojo() {
    int tiempoEsperaEnRojo=0; // contador de tiempo del estado
    digitalWrite(emisorIR1,HIGH);
    digitalWrite(emisorIR2,HIGH);
    digitalWrite(emisorIR3,HIGH)

do{
    tiempoEsperaEnRojo=tiempoEsperaEnRojo+50;
    lecturasBarreras(receptorIR1,receptorIR2,receptorIR3);
    if(lectorIR1<100) {
        T1=millis(); // tiempo de corte lector1
        digitalWrite(acceso_sistema,HIGH);
        delay(50);
        digitalWrite(acceso_sistema,LOW);
        distanciaPuntual=distanciaDefinida;//distancia de corte

        break;
    }
    else if(lectorIR2<50){ // corte de lector2
        T2=millis();
        velocidadCalculada1=0;
        digitalWrite(emisorIR1,LOW);

        Serial.print("lector2: ");
        Serial.println(lectorIR2);

        digitalWrite(acceso_sistema,HIGH);
        delay(50);
        digitalWrite(acceso_sistema,LOW);

        break;
    }
    else if(lectorIR3<50){ // corte de lector 3
        T3=millis();
        velocidadCalculada1=0;
        digitalWrite(emisorIR2,LOW);
        digitalWrite(emisorIR1,LOW);

```

```

        Serial.print("lector3: ");
        Serial.println(lectorIR3);

        digitalWrite(acceso_sistema,HIGH);
        delay(50);
        digitalWrite(acceso_sistema,LOW);

        break;
    }

}while(rojo&&tiempoEsperaEnRojo<DuracionMaximaRojo); // fin cortes
1,2,3

```

Las lecturas se realizan de tres en tres para suponer una situación en la que una de las barreras puede fallar o una de ellas puede no ser cortadas entonces habría más posibilidades de que se corte la siguiente barrera.

```

//***** CORTES 2,3, 4*****

lcd.clear();
double tiempoIntermedio=0;
digitalWrite(emisorIR3,HIGH);
digitalWrite(emisorIR4,HIGH);
do{
    tiempoEsperaEnRojo=tiempoEsperaEnRojo+50;
    Serial.print("tiempo espera: ");
    Serial.println(tiempoEsperaEnRojo);
    lecturasBarreras (receptorIR2, receptorIR3, receptorIR4);
    if(lectorIR1<50){ // se ha producido corte 2 y comparamos 1y2
        T2=millis(); //
        Serial.print("lector2: ");
        Serial.println(lectorIR1);
        digitalWrite(acceso_sistema,HIGH);
        delay(50);
        digitalWrite(acceso_sistema,LOW);
        distanciaPuntual=distanciaPuntual-distanciaTramo; //
        digitalWrite(emisorIR1,LOW);

// CALCULO DE TIEMPO Y VELOCIDAD
        tramo=(T2-T1)/1000; // duración entre barreras 1y2
        velocidadCalculada1=distanciaTramo*K/tramo;
        velocidadCalculada2=0;

// CALCULAMOS EL ERROR y COMPROBACION DEL PIR
        informacionOtrosParametros();
        informacionPir();
        calculoError();

// COMPARACIONES DE LAS CONDICIONES TRAMO "A"

//condición 0
        if (Error<=ErrorMin){
            iluminacionBarrera0();
        }

//condición 1
        if (Error>ErrorMin&&Error<=ErrorMax && estadoPIR==HIGH){
            iluminacionBarrera9();
        }

//condición 2

```

```

        else if (Error>ErrorMin&&Error<=ErrorMax&& estadoPIR==LOW) {
            iluminacionBarrera8();
        }
//condición 3
        else if (Error>ErrorMax && estadoPIR==LOW) {
            iluminacionBarrera11();
        }
//condición 4
        else if (Error>ErrorMax && estadoPIR==HIGH) {
            iluminacionBarrera12();
        }
        break;
    }

else if (lectorIR1<50) { // corte lector 3
    T3=millis();
    Serial.print("lector3: ");
    Serial.println(lectorIR1);
    digitalWrite(acceso_sistema,HIGH);
    delay(100);
    digitalWrite(acceso_sistema,LOW);
    distanciaPuntual=distanciaPuntual-distanciaTramo;
    digitalWrite(emisorIR2,LOW);
    digitalWrite(emisorIR3,LOW);

// CALCULO DE TIEMPO Y VELOCIDAD
    tramo=(T3-T2)/1000; // duración entre barreras 2y3

    velocidadCalculada2=velocidadCalculada1;
    velocidadCalculada1=distanciaTramo*K/tramo;

// CALCULAMOS EL ERROR y COMPROBACION DEL PIR
    informacionPir();
    calculoError();
    informacionOtrosParametros();

// CONDICIONES SALIDAS ADOPTADAS EN EL TRAMO "B"

//condición 0
    if (Error<=ErrorMin) {
        iluminacionBarrera0();
    }

//condición 5
    if (Error>ErrorMin&&Error<=ErrorMax&& estadoPIR==HIGH &&
    velocidadCalculada2>velocidadCalculada1) {
        iluminacionBarrera7();
    }

//condición 6
    if (Error>ErrorMin&&Error<=ErrorMax&& estadoPIR==HIGH &&
    velocidadCalculada2<velocidadCalculada1) {
        iluminacionBarrera8();
    }
//condición 7
    else if (Error>ErrorMin&&Error<=ErrorMax&& estadoPIR==LOW ) {
        iluminacionBarrera7();
    }
//condición 3

    else if (Error>ErrorMax && estadoPIR==LOW) {

```

```

        iluminacionBarrera9();
    }

    //condición 9
    else if(Error>ErrorMax && estadoPIR==HIGH
    &&velocidadCalculada2>velocidadCalculada1){
        iluminacionBarrera9();
    }

    //condición 10
    else if(Error>ErrorMax && estadoPIR==HIGH
    &&velocidadCalculada2<velocidadCalculada1){
        iluminacionBarrera10();
    }

    break;
}
}

```

4.5.3. Problemas encontrados tras realizar las primeras pruebas en el sistema

A continuación se mencionará algunos problemas encontrados a la hora de realizar las primeras pruebas en el sistema y el modo en el que fueron solucionados de manera parcial o totalmente.

4.5.3.1. Falsas lecturas y lecturas no producidas en las barreras infrarrojas

El primer problema encontrado en el sistema, fue cuando se empezó hacer lecturas en las barreras infrarrojas. Habían barreras que daban falsas lecturas aunque su emisor correspondiendo no emitía ninguna señal y eso era debido a dos grandes motivos.

Por un lado era debido a la mala alineación entre emisor-receptor puestos en cada barrera, por lo tanto, la señal analógica que llegaba era muy baja (entre **0 y 5 bits**) y las lecturas no se realizaban correctamente. Otro de problemas encontrados fue la mala instalación en algunos sensores receptores de modo que, el conductor de toma de datos estaba posicionado en el lugar equivocado y por lo tanto las lecturas eran muy elevadas llegando incluso al valor máximo (**1024 bits**) cosa que era imposible, ya que la separación entre el emisor y el receptor tenía que producir algunas pérdidas en la señal. En síntesis con respecto al segundo problema no se realizaba ninguna lectura y el sensor tomaba el valor máximo como valor medido.

4.5.3.2. Solución a los problemas de las lecturas.

La primera solución adoptada para el primer problema era muy sencilla, consistía en alinear los dos módulos (emisor-receptor) con una buena precisión que permitía que la señal aumentase hasta alcanzar o superar los valores mínimos requeridos para un correcto funcionamiento.

Para el segundo problema también hubo una solución fácil, al tratarse de un fallo técnico en la instalación, solo se tenía que revisar y modificar las conexiones de los módulos que estaban mal instalados. Una vez realizada esta modificación se podía observar como la señal captada por el receptor variaba de valores máximos a valores que sí proporcionaba el emisor.

Y para asegurar y tener controladas todas las lecturas se optó por instalar un indicador LED (*azul*) que nos informaba de cada una de las lecturas producidas mediante un parpadeo (*blink*) y de este modo resultaba más fácil localizar y reparar las futuras averías relacionadas con las barreras infrarrojas.

4.5.3.3. Pocas barreras de iluminación en el sistema

La primera disposición adoptada en el sistema solo constaba de seis barreras luminosas, colocadas cada una justo detrás de cada barrera infrarroja y separadas siete centímetros, es decir, a cada barrera infrarroja se le asignada una única barrera luminosa. Esta primera disposición funcionaba perfectamente, pero al tratarse de querer crear un impacto visual de alerta prolongada, una vez que el automóvil haya pasado por los tres primeros tramos del circuito, se apagaban las barreras luminosas atravesadas de manera gradual y ya no quedaban muchas barreras que iluminar a pesar de las múltiples opciones que en el sistema se podrían tomar según comportamientos diferentes.

4.5.3.4. Solución del problema de las barreras luminosas

La única solución viable a este problema fue agregar más barreras luminosas al sistema, seis fueron las barreras agregadas alcanzando una cantidad de doce barreras luminosas en total, que a pesar de ser un trabajo costoso ya que se trataba de retocar tanto el entorno de pruebas, la instalación y el código de programación, iba a ser una mejora y ampliaría el rango de opciones o comportamientos que se podría tomar en la salida y que por tanto no se podía haber notado en la distribución primera.

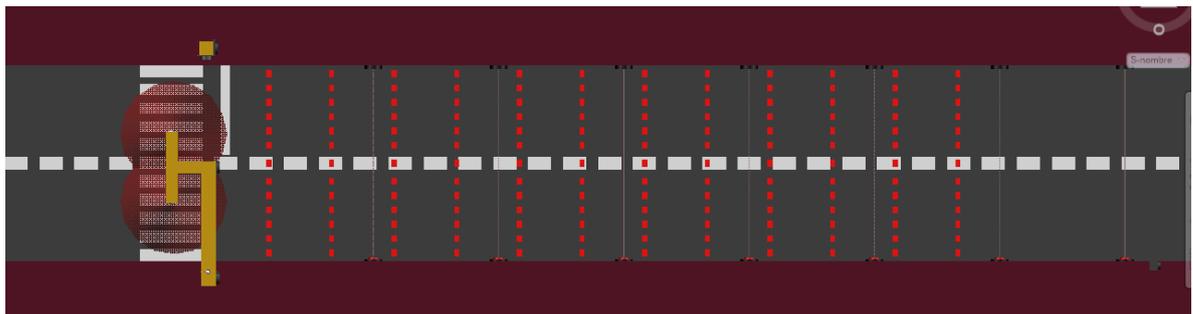


Ilustración 95: nueva distribución con 12 barreras de iluminación

Es necesario mencionar que en un modelo real se podría disponer de más barreras para crear un mejor impacto visual y por lo tanto los conductores estarían alertados en un periodo de tiempo más prolongado.

4.5.3.5. Poco periodo de tiempo entre puntos de corte que limitan un tramo

Partiendo del periodo de tiempo entre tramos establecido (*T_c normal igual a un segundo*) para una velocidad normal V_N y al tratarse de un entorno de pruebas con medidas muy reducidas, la variación de la velocidad entre un tramo y otro tramo era difícil de detectar y/o diferenciar aunque en el interfaz del operador (pantalla LCD) se mostrase que en realidad las dos velocidades eran diferentes. Eso era debido a que todos los tiempos de corte calculados en los tramos tomaban como punto de referencia T_c normal y tenían que ser menores a un segundo para simular que el vehículo iba a gran velocidad y por lo tanto la escasa separación entre barreras infrarrojas hacían imposible notar el frenado y /o aceleración en este pequeño tramo (*siente centímetros de distancia*) sin que afectara la siguiente barrera y por lo tanto no se podía apreciar la evolución del vehículo con total normalidad y claridad.

4.5.3.6. Solución del problema de poco periodo de tiempo predefinido en cada tramo.

La solución adoptaba para este problema fue aumentar el periodo de tiempo normal entre tramos (*a dos con cinco segundos*) con fin de poder apreciar como variaba la velocidad calculada respecto a los tiempos tomados y así poder realizar maniobras de frenado y aceleración y ver como varía la salidas para cada comportamiento adoptado por el vehículo.

En la simulación real no haría falta esta modificación ya que la separación entre barreras sería mayor y se podrían apreciar a ojo cómo evoluciona el automóvil.

4.5.3.7. Retardo en la activación y toma de información del sensor de movimiento

Este fue uno de los problemas más difíciles de detectar en el sistema por dos grandes factores:

Al tratarse de un proceso que se debía comprobar en cada tramo, una vez que el regulador PID haya realizado sus operaciones. La activación de este sensor y puesta en modo de operación requiere al menos tres o cuatro segundos para empezar a tomar datos. Pero debido a este retardo, habían tramos que no se llegaba a comprobar la presencia o ausencia de peatones y el sistema toma como correcta los datos tomados del tramo anterior.

El otro factor a tener en cuenta fue que, al tratarse de un dispositivo aplicable en instalaciones reales, el entorno de detección en el prototipo era tan pequeño que a veces hacia lecturas en zonas fuera del entorno (*por ejemplo: movimientos realizados por una persona alejada un par de metros del prototipo*) y por lo tanto el controlador las tomaba como buenas y con ellas trabajaba para tomar una salida conjunta junto con la información del regulador.

4.5.3.8. Solución del problema de retardo en la activación del sensor de movimiento

La solución adoptada a este problema no proporcionó resultados 100% correctos pero sí que redujo de manera considerable este problema. Se cree que para obtener una solución total, se debía escalar las medidas de prototipo con el fin de que el sensor tenga más área de trabajo.

Lo que se hizo para conseguir una solución parcial fue calibrar el sensor gracias a dos potenciómetros que trae consigo dentro del circuito. Uno de ellos se utilizó para ajustando la sensibilidad del sensor con fin de que no detectará todo movimiento cercano sino solo aquellos que se producían muy cerca de la lente. El otro potenciómetro se utilizó para ajustar el *trigger* el cual ayuda a reducir tiempo de espera para su activación y puesta en modo de operación a casi medio segundo o menos.

4.5.3. Ejemplo Práctico del funcionamiento de todo el sistema

En este ejemplo se utilizará como datos de partida los siguientes:

Velocidad normal de 50km/h equivalente a 14m/s y la duración normal en atravesar un tramo sería de 2,5 segundos y el funcionamiento del sistema partirá con la detección del sensor LDR de la escasa visibilidad presente en el escenario con valores comprendidos entre 800 y 1024 *bits* según rango programado.

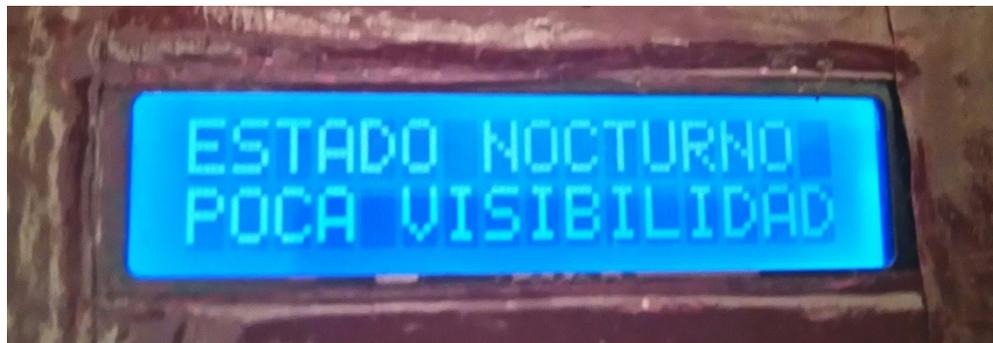


Ilustración 96: Mensaje informativo de estado Nocturno

Supongamos que se ha detectado escasa visibilidad en el escenario, el sistema entrará en funcionamiento. Una vez comenzado el cambio de estados de semáforo (*verde, ámbar, rojo*) se activarán los emisores de las barreras infrarrojas suponiendo el caso de que todas funcionan correctamente. Si el receptor infrarrojo detecta un bajo nivel en la señal recibida, enviará esa información al controlador, avisándole de que se ha producido un primer corte. En el supuesto caso de que haya producido un segundo corte detectado por el emisor dos, el controlador calculará el tiempo que ha tardado un automóvil en recorrer todo el tramo, con ese dato y la separación entre barreras se calculará la velocidad a la que iba $Vc1$.

Tras calcular este valor el regulador calculará el error producido a partir del valor permitido de la velocidad ($Error = Vn - Vc1$). Si el error obtenido está dentro de *rango de*

error permitido, es decir, que la $Vc1$ sea prácticamente igual o por debajo de velocidad permitida (por ejemplo $Vc1=53\text{km/h}$ o $Vc1=42\text{km/h}$), no será necesario comprobar los otros parámetros del sistema y el controlador adoptará un salida de *No Iluminación* debido a que la calificaría como progreso fuera de peligro.

En el supuesto caso de que el error calculado es considerablemente grande (20%-50%) o muy grande (*mayor que el 50%*), es decir, que la $Vc1$ ha superado de manera considera a la velocidad permitida (por ejemplo $Vc1=70\text{km/h}$ o $Vc1=95\text{km/h}$) y por lo tanto una de las acciones del regulador PID (*proporcional integral PI o derivativo D*) entra en acción y adoptará una posible salida parcial en función de este error. Posteriormente el controlador calculará la distancia a la que se produjo el último corte y activaría el sensor de movimiento instalado sobre paso de peatones. Una vez que el sensor haya capturado la información en su área de trabajo (*Si presencia 1 o No presencia 0*) la enviaría al controlador y en función de la respuesta (*comportamiento peligroso*) del error, a la distancia del último corte y a la información del sensor PIR, el controlador adoptaría una salida un otra para activar una determinada cantidad de barreras.

Toda esa información procesa por el controlador será mostrada en una pantalla *display* LCD y podrá ser visualizada por la persona que esté realizando las simulaciones tal como se puede ver en la *ilustración 94*.



Ilustración 97: Información sobre el comportamiento en el sistema

- **V:** velocidad transformada en m/s
- **PIR:** sensor de movimiento NO presencia
- **Pt:** periodo de tiempo en recorrer el tramo
- **D:** distancia a la que se produjo el último corte hasta la línea de parada

El comportamiento del sistema cambiaría una vez que el automóvil haya atravesado el primer tramo, en el sentido de que ahora no solo se tendrá en cuenta los parámetros explicados anteriormente para analizar el comportamiento del vehículo en los siguientes tramos hasta el final, sino que también se tendrá en cuenta la evolución de éste a lo largo del nuevo tramo antes de realizar el siguiente corte.

Este nuevo parámetro consistirá en conocer el comportamiento de frenado y/o aceleración gracias a un contador de tiempo (*TiempoIntermedio*) que arrancaría una vez realizado el último corte y supondrá que si la velocidad del coche es constante, el automóvil deberá

tardar lo mismo que tardó en atravesar el tramo anterior. Si se registra el caso de que ese contador ya ha superado el periodo de tiempo que se tardó en el tramo anterior, las barreras activadas en la última salida proporcionada por el controlador se irán desactivando progresivamente desde la más cercana al vehículo hasta la última si se diera el caso de que el coche tarde más tiempo o este en posición de parada.

Una vez analizado este comportamiento intermedio y se haya producido finalmente el siguiente corte, el sistema volverá a repetir todo el proceso explicado al principio e ir adoptando diferentes salidas, pero cada vez más con menos barreras que iluminar ya que todas las que están en tramos ya atravesados estarán apagadas ya definitivamente hasta el nuevo proceso.

Ante todo es necesario mencionar que sistema no solo está diseñado para analizar el comportamiento desde el principio sino que se ha supuesto casos como:

- Que alguna de las barreras no funcione, principalmente las cuatro primeras ya que a partir de la quinta se supone que vehículo ya estará muy cercano al semáforo y podrá llegar hasta la parada sin la ayuda del sistema.
- Que cuando el semáforo se puso en rojo el vehículo ya estaba a mitad del recorrido y por lo tanto las barreras infrarrojas atravesadas ya no deben entrar en el funcionamiento del sistema.
- Que con el semáforo en verde o ámbar se detecte presencia peatonal cruzando la vía de circulación.

Ante estos tres grandes inconvenientes, el sistema empezará a tomar información del entorno a partir de la siguiente barrera y actuar.

4.6. Programación con Arduino de sistema

4.6.1. Reglas definidas en el sistema

Antes de realizar la programación del sistema se establecieron una serie de reglas o condiciones a través de las cuales el controlador iba a decidir conforme a todos los parámetros calculados y poder adoptar una respuesta adecuada y proceder a la activación de la salida correspondiente. Estas series de reglas preestablecidas son las que a continuación se citan.

Regla uno: la regla uno se da cuando el error calculado en un determinado tramo es relativamente pequeño, es decir corresponde a rango de errores permitidos (error menor al 20%) predefinidos en el regulador.

En este caso, se supondrá que el automóvil viaja a una velocidad normal o menos que la permitida y por lo tanto la salida adoptada por el controlador debería ser *no activar ninguna de las barreras luminosas*.

Regla dos: en esta segunda regla se establece el caso en que la error calculado es relativamente grande, es decir, que corresponde al rango de errores con evolución lenta ($20\% < \text{error} < 50\%$) predefinidos en el regulador. Si en este momento se registra el caso de que SI existe presencia (binario 1 o *high*) de peatones cruzando la vía y la velocidad de automóvil sigue aumentando con respecto a la velocidad del tramo anterior, *el controlador activaría una salida de iluminación relativamente mayor en función de la cantidad de barreras luminosas restante.*

Regla tres: en esta tercera regla se establece el caso en que la error calculado es relativamente grande, es decir, que corresponde al rango de errores con evolución lenta ($20\% < \text{error} < 50\%$) predefinidos en el regulador. Si en este momento se registra el caso de que NO existe presencia (binario 0 o *Low*) de peatones cruzando la vía y la velocidad de automóvil sigue aumentando con respecto a la velocidad del tramo anterior, *el controlador comprobaría a qué distancia se realizó el último corte y activaría una salida de iluminación relativamente mayor/mediano en función de la cantidad de barreras luminosas restante.*

Regla cuatro: en esta cuarta regla se establece el caso en que la error calculado es relativamente grande, es decir, que corresponde al rango de errores con evolución lenta ($20\% < \text{error} < 50\%$) predefinidos en el regulador. Independientemente de que se compruebe el estado del sensor de movimiento, si la velocidad de automóvil está disminuyendo con respecto a la velocidad del tramo anterior, *el controlador activaría una salida de iluminación que se iría apagando progresivamente en relación a la salida adoptada en el tramo anterior o se activarían un número reducido de barreras de iluminación en caso de de estuvieran apagadas posteriormente.*

Regla cinco: para establecer esta quinta regla, se supondrá el caso en que error calculado es excesivamente muy grande, es decir, que el automóvil viaja a una velocidad excesivamente grande y por lo tanto, corresponde al rango de errores con evolución rápida (mayor que el 50%) pre establecidos en el regulador. Si en este momento se confirma la existencia de peatones cruzando, independiente de la distancia a la que se produjo el último corte o de la evolución de la velocidad con respecto al tramo anterior, *el controlador adoptará una salida del alto nivel activando casi todas las barreras luminosas restantes desde la primera hasta la más cercana al automóvil.*

Regla seis: en esta sexta condición es igual que la anterior pero con la diferencia de que si el estado del sensor de movimiento es un No presencia (binario 0 o *low*) se debería comprobar la distancia a la que está el vehículo antes de llegar a la línea de parada. Si la distancia es larga, el controlador debería esperar a que realice el siguiente corte y volver a comparar los nuevos parámetros y adoptar en este instante una salida de iluminación *media/baja*. Pero si se da el caso de que distancia es corta o muy corta el controlador debería proporcionar una salida de iluminación *medio/alto*

Regla siete: el enfoque de la regla siete se fundamenta en la distancia a la que estaría el automóvil de la línea de parada de modo que, si resulta que está cerca o muy cerca y el error calculado está por encima de lo permitido no será necesario comprobar el estado del sensor de movimiento debido a la proximidad y el controlador deberá adoptar una salida de activación de las barreras en función de la cantidad de barreras restantes que podrían ser *medio/bajo*.

Regla ocho: la octava condición hace referencia a la situación en la que hay un individuo cruzando el paso con el semáforo en verde o ámbar, por lo tanto, solo se debería tener en cuenta el estado del sensor de movimiento de modo que si se registra presencia (binario 1 o *high*) el controlador deberá adoptar una salida inmediata activar una cierta cantidad de barreras en función de la distancia a la que está el automóvil.

4.6.1. Condigo fuente programado

Lo primero que se hizo fue llamar a las librerías utilizadas para la pantalla LCD. Estas eran las únicas utilizadas ya que las librerías del regulador PID que trae Arduino regulaban la velocidad a valores permitidos independiente del valor calculado de la velocidad. En síntesis estas librerías del PID creo que están pensados para proyectos como regular la temperatura en un proceso a uno o unos valores ya fijados.

```

/* CODIGO DE PROGRAMACION PARA LA REDUCCION DE ATROPELOS EN LOS PASOS DE
PEATONES*/
#include <LiquidCrystal.h>

// DECLARAMOS LOS PIN DE LAS VARIABLES A UTILIZAR
LiquidCrystal lcd(37, 36, 35, 34, 33, 32);

// receptores infrarrojos
const int receptorIR1=A1;
const int receptorIR2=A2;
const int receptorIR3=A3;
const int receptorIR4=A4;
const int receptorIR5=A5;
const int receptorIR6=A6;
const int receptorIR7=A7;

// emisores infrarrojos
int emisorIR1=2;
int emisorIR2=3;
int emisorIR3=4;
int emisorIR4=5;
int emisorIR5=6;
int emisorIR6=7;
int emisorIR7=8;

// lectores IR
int lectorIR1;
int lectorIR2;
int lectorIR3;

// Barreras de iluminacion

int iluminaBarrera1=22;
int iluminaBarrera2=23;
int iluminaBarrera3=24;
int iluminaBarrera4=25;
int iluminaBarrera5=26;
int iluminaBarrera6=27;
int iluminaBarrera7=28;

```

```

int iluminaBarrera8=29;
int iluminaBarrera9=30;
int iluminaBarrera10=31;
int iluminaBarrera11=39;
int iluminaBarrera12=40;

// sensor de presencia

const int sensorPIR=9;
int estadoPIR;

// sensor de iluminacion

const int sensorLDR=A0;
int valorLDR;

// capturas de tiempo de corte en cada barrera infrarroja

double T1=0;
double T2=0;
double T3=0;
double T4=0;
double T5=0;
double T6=0;
double T7=0;

// tiempo transcurido en cada tramo,
double tramo=0;

// luces semaforos

int verde=10;
int ambar=11;
int rojo=12;

// indicador de corte de barrera infrarroja
int acceso_sistema=38;

// velocidad preestablecida del tramo de control
int velocidadNormal=14;//m/s o 50km/h
int K=2; // Ganancia  $K=V_n/V_p$ 

// distrancia desde el corte hasta la parada
double velocidadCalculada1;
double velocidadCalculada2;

// Variables del error
double Error;
double Kp=4;
double Ki=1.5;
double Kd=2;
double ErrorMin;
double ErrorMax;

int distanciaDefinida=42; //cm
int distanciaPuntual;;
int distanciaTramo=7;//cm

// duración de los estados

int DuracionMaximaRojo=6000;

```

```

int DuracionMaximaVerde=8000;
int retardo2=300;// retardo entre luces
int retardo1=40;// retardo entre luces

// DECLARACION DE LAS ENTRADAS Y SALIDAS EN L VOID SETUP

void setup(){
  lcd.begin(16,2);
  Serial.begin(9600);

  pinMode(verde,OUTPUT);
  pinMode(ambar,OUTPUT);
  pinMode(rojo,OUTPUT);

  pinMode(sensorLDR,INPUT);

  pinMode(sensorPIR,INPUT);

  pinMode(emisorIR1,OUTPUT);
  pinMode(emisorIR2,OUTPUT);
  pinMode(emisorIR3,OUTPUT);
  pinMode(emisorIR4,OUTPUT);
  pinMode(emisorIR5,OUTPUT);
  pinMode(emisorIR6,OUTPUT);
  pinMode(emisorIR7,OUTPUT);

  pinMode(receptorIR1,INPUT);
  pinMode(receptorIR2,INPUT);
  pinMode(receptorIR3,INPUT);
  pinMode(receptorIR4,INPUT);
  pinMode(receptorIR5,INPUT);
  pinMode(receptorIR6,INPUT);
  pinMode(receptorIR7,INPUT);

  pinMode(acceso_sistema,OUTPUT);

  pinMode(iluminaBarrera1,OUTPUT);
  pinMode(iluminaBarrera2,OUTPUT);
  pinMode(iluminaBarrera3,OUTPUT);
  pinMode(iluminaBarrera4,OUTPUT);
  pinMode(iluminaBarrera5,OUTPUT);
  pinMode(iluminaBarrera6,OUTPUT);
  pinMode(iluminaBarrera7,OUTPUT);
  pinMode(iluminaBarrera8,OUTPUT);
  pinMode(iluminaBarrera9,OUTPUT);
  pinMode(iluminaBarrera10,OUTPUT);
  pinMode(iluminaBarrera11,OUTPUT);
  pinMode(iluminaBarrera12,OUTPUT);

  lcd.print("SIST. D REDUCCION");
  lcd.setCursor(0,1);
  lcd.print("  DE ATROPELLOS");
  delay(2500);
  lcd.clear();
}

// funciones del sistema diurno

void cambio_estados_dia(){
  digitalWrite(ambar,HIGH);

```

```

    delay(1000);
    digitalWrite(ambar, LOW);
    delay(10);
    digitalWrite(verde, HIGH);
    delay(1500);
    digitalWrite(verde, LOW);
    delay(10);
    digitalWrite(ambar, HIGH);
    delay(1000);
    digitalWrite(ambar, LOW);
    delay(10);
    digitalWrite(rojo, HIGH);
    delay(2000);
    digitalWrite(rojo, LOW);
    delay(10);
}

void calculoError() {

    /*Intervalos preestablecidos sobre el que actuara el regulador
    Error permitido=<=0,2 o 20%
    Error actuación PI: 0.201<=Error<0.6 ó 20,01%-60%
    Error actuación D: Error>0.6 ó 60%
    */
    int ouput=0;
    ErrorMin=1/(1+Kp*Ki);
    ErrorMax=1/Kd;
    Error=(-1*(velocidadNormal-velocidadCalculada1)/velocidadNormal);
    Output=velocidadNormal*Error*(1/(1+Kp*Ki))+VelcidadCalculada1*Error
*(1/Kd);

}

// funcion para el sensor PIR
void informacionPir() {
    estadoPIR=digitalRead(sensorPIR);
    if(estadoPIR==HIGH) {
        lcd.setCursor(0,1);
        lcd.print("PIR:");
        lcd.print("SI");
    }
    else{
        lcd.setCursor(0,1);
        lcd.print("PIR:");
        lcd.print("NO");
    }
}

// funciones para iluminar barreras

void iluminacionBarrera0() {
    digitalWrite(iluminaBarrera12, LOW);
    delay(20);
    digitalWrite(iluminaBarrera11, LOW);
    delay(20);
    digitalWrite(iluminaBarrera10, LOW);
    delay(20);
    digitalWrite(iluminaBarrera9, LOW);
    delay(20);
}

```

```

        digitalWrite(iluminaBarrera8,LOW);
        delay(20);
        digitalWrite(iluminaBarrera7,LOW);
        delay(20);
        digitalWrite(iluminaBarrera6,LOW);
        delay(20);
        digitalWrite(iluminaBarrera5,LOW);
        delay(20);
        digitalWrite(iluminaBarrera4,LOW);
        delay(20);
        digitalWrite(iluminaBarrera3,LOW);
        delay(20);
        digitalWrite(iluminaBarrera2,LOW);
        delay(20);
        digitalWrite(iluminaBarrera1,LOW);
    }
    void iluminacionBarrera1(){
        digitalWrite(iluminaBarrera1,HIGH);
        delay(retardo1);
        digitalWrite(iluminaBarrera6,LOW);
        delay(retardo1);
        digitalWrite(iluminaBarrera5,LOW);
        delay(retardo1);
        digitalWrite(iluminaBarrera4,LOW);
        delay(retardo1);
        digitalWrite(iluminaBarrera3,LOW);
        delay(retardo1);
        digitalWrite(iluminaBarrera2,LOW);
    }
    void iluminacionBarrera2(){
        digitalWrite(iluminaBarrera1,HIGH);
        delay(retardo1);
        digitalWrite(iluminaBarrera2,HIGH);
        delay(retardo1);
        digitalWrite(iluminaBarrera6,LOW);
        delay(retardo1);
        digitalWrite(iluminaBarrera5,LOW);
        delay(retardo1);
        digitalWrite(iluminaBarrera4,LOW);
        delay(retardo1);
        digitalWrite(iluminaBarrera3,LOW);
        delay(retardo2);
        digitalWrite(iluminaBarrera2,LOW);
    }
    void iluminacionBarrera3(){
        digitalWrite(iluminaBarrera1,HIGH);
        delay(retardo1);
        digitalWrite(iluminaBarrera2,HIGH);
        delay(retardo1);
        digitalWrite(iluminaBarrera3,HIGH);
        delay(retardo1);
        digitalWrite(iluminaBarrera4,LOW);
        delay(retardo1);
        digitalWrite(iluminaBarrera5,LOW);
        delay(retardo1);
        digitalWrite(iluminaBarrera6,LOW);
        digitalWrite(iluminaBarrera7,LOW);
    }

```

```

        digitalWrite(iluminaBarrera8, LOW);
        digitalWrite(iluminaBarrera9, LOW);
        digitalWrite(iluminaBarrera10, LOW);
        digitalWrite(iluminaBarrera11, LOW);
        delay(retardo2);
        digitalWrite(iluminaBarrera3, LOW);
    }
    void iluminacionBarrera4() {
        digitalWrite(iluminaBarrera1, HIGH);
        delay(retardo1);
        digitalWrite(iluminaBarrera2, HIGH);
        delay(retardo1);
        digitalWrite(iluminaBarrera3, HIGH);
        delay(retardo1);
        digitalWrite(iluminaBarrera4, HIGH);
        delay(retardo1);
        digitalWrite(iluminaBarrera5, LOW);
        delay(retardo1);
        digitalWrite(iluminaBarrera6, LOW);
        digitalWrite(iluminaBarrera7, LOW);
        digitalWrite(iluminaBarrera8, LOW);
        digitalWrite(iluminaBarrera9, LOW);
        digitalWrite(iluminaBarrera10, LOW);
        digitalWrite(iluminaBarrera11, LOW);
        delay(retardo2);
        digitalWrite(iluminaBarrera4, LOW);
    }

    void iluminacionBarrera5() {
        digitalWrite(iluminaBarrera1, HIGH);
        delay(retardo1);
        digitalWrite(iluminaBarrera2, HIGH);
        delay(retardo1);
        digitalWrite(iluminaBarrera3, HIGH);
        delay(retardo1);
        digitalWrite(iluminaBarrera4, HIGH);
        delay(retardo1);
        digitalWrite(iluminaBarrera5, HIGH);
        delay(retardo1);
        digitalWrite(iluminaBarrera6, LOW);
        digitalWrite(iluminaBarrera7, LOW);
        digitalWrite(iluminaBarrera8, LOW);
        digitalWrite(iluminaBarrera9, LOW);
        digitalWrite(iluminaBarrera10, LOW);
        digitalWrite(iluminaBarrera11, LOW);
        digitalWrite(iluminaBarrera12, LOW);
        delay(retardo2);
        digitalWrite(iluminaBarrera5, LOW);
    }

    void iluminacionBarrera6() {
        digitalWrite(iluminaBarrera1, HIGH);
        delay(retardo1);
        digitalWrite(iluminaBarrera2, HIGH);
        delay(retardo1);
        digitalWrite(iluminaBarrera3, HIGH);
        delay(retardo1);
        digitalWrite(iluminaBarrera4, HIGH);
        delay(retardo1);
    }

```

```

        digitalWrite( iluminaBarrera5, HIGH );
        delay( retardo1 );
        digitalWrite( iluminaBarrera6, HIGH );
        digitalWrite( iluminaBarrera7, LOW );
        digitalWrite( iluminaBarrera8, LOW );
        digitalWrite( iluminaBarrera9, LOW );
        digitalWrite( iluminaBarrera10, LOW );
        digitalWrite( iluminaBarrera11, LOW );
        digitalWrite( iluminaBarrera12, LOW );
        delay( retardo2 );
        digitalWrite( iluminaBarrera6, LOW );
        delay( retardo2 );
        digitalWrite( iluminaBarrera5, LOW );
    }
    void iluminacionBarrera7() {
        digitalWrite( iluminaBarrera1, HIGH );
        delay( retardo1 );
        digitalWrite( iluminaBarrera2, HIGH );
        delay( retardo1 );
        digitalWrite( iluminaBarrera3, HIGH );
        delay( retardo1 );
        digitalWrite( iluminaBarrera4, HIGH );
        delay( retardo1 );
        digitalWrite( iluminaBarrera5, HIGH );
        delay( retardo1 );
        digitalWrite( iluminaBarrera6, HIGH );
        delay( retardo1 );
        digitalWrite( iluminaBarrera7, HIGH );
        digitalWrite( iluminaBarrera8, LOW );
        digitalWrite( iluminaBarrera9, LOW );
        digitalWrite( iluminaBarrera10, LOW );
        digitalWrite( iluminaBarrera11, LOW );
        digitalWrite( iluminaBarrera12, LOW );
        delay( retardo2 );
        digitalWrite( iluminaBarrera7, LOW );
    }
    void iluminacionBarrera8() {

        digitalWrite( iluminaBarrera1, HIGH );
        delay( retardo1 );
        digitalWrite( iluminaBarrera2, HIGH );
        delay( retardo1 );
        digitalWrite( iluminaBarrera3, HIGH );
        delay( retardo1 );
        digitalWrite( iluminaBarrera4, HIGH );
        delay( retardo1 );
        digitalWrite( iluminaBarrera5, HIGH );
        delay( retardo1 );
        digitalWrite( iluminaBarrera6, HIGH );
        delay( retardo1 );
        digitalWrite( iluminaBarrera7, HIGH );
        delay( retardo1 );
        digitalWrite( iluminaBarrera8, HIGH );
        digitalWrite( iluminaBarrera9, LOW );
        digitalWrite( iluminaBarrera10, LOW );
        digitalWrite( iluminaBarrera11, LOW );
        digitalWrite( iluminaBarrera12, LOW );
        delay( retardo2 );
        digitalWrite( iluminaBarrera8, LOW );
        delay( retardo2 );
    }

```

```

        digitalWrite(iluminaBarrera7,LOW);
    }
    void iluminacionBarrera9() {

        digitalWrite(iluminaBarrera1,HIGH);
        delay(retardol);
        digitalWrite(iluminaBarrera2,HIGH);
        delay(retardol);
        digitalWrite(iluminaBarrera3,HIGH);
        delay(retardol);
        digitalWrite(iluminaBarrera4,HIGH);
        delay(retardol);
        digitalWrite(iluminaBarrera5,HIGH);
        delay(retardol);
        digitalWrite(iluminaBarrera6,HIGH);
        delay(retardol);
        digitalWrite(iluminaBarrera7,HIGH);
        delay(retardol);
        digitalWrite(iluminaBarrera8,HIGH);
        delay(retardol);
        digitalWrite(iluminaBarrera9,HIGH);
        digitalWrite(iluminaBarrera10,LOW);
        digitalWrite(iluminaBarrera11,LOW);
        digitalWrite(iluminaBarrera12,LOW);
        delay (retardo2);
        digitalWrite(iluminaBarrera10,LOW);
        delay(retardo2);
        digitalWrite(iluminaBarrera9,LOW);
    }

    void iluminacionBarrera10() {
        digitalWrite(iluminaBarrera1,HIGH);
        delay(retardol);
        digitalWrite(iluminaBarrera2,HIGH);
        delay(retardol);
        digitalWrite(iluminaBarrera3,HIGH);
        delay(retardol);
        digitalWrite(iluminaBarrera4,HIGH);
        delay(retardol);
        digitalWrite(iluminaBarrera5,HIGH);
        delay(retardol);
        digitalWrite(iluminaBarrera6,HIGH);
        delay(retardol);
        digitalWrite(iluminaBarrera7,HIGH);
        delay(retardol);
        digitalWrite(iluminaBarrera8,HIGH);
        delay(retardol);;
        digitalWrite(iluminaBarrera9,HIGH);
        delay(retardol);
        digitalWrite(iluminaBarrera10,HIGH);
        delay(retardo2);
        digitalWrite(iluminaBarrera10,LOW);
        delay(retardo2);
        digitalWrite(iluminaBarrera9,LOW);
    }
    void iluminacionBarrera11() {

        digitalWrite(iluminaBarrera1,HIGH);
        delay(retardol);
        digitalWrite(iluminaBarrera2,HIGH);
    }

```

```

        delay (retardol);
        digitalWrite (iluminaBarrera3, HIGH);
        delay (retardol);
        digitalWrite (iluminaBarrera4, HIGH);
        delay (retardol);
        digitalWrite (iluminaBarrera5, HIGH);
        delay (retardol);
        digitalWrite (iluminaBarrera6, HIGH);
        delay (retardol);
        digitalWrite (iluminaBarrera7, HIGH);
        delay (retardol);
        digitalWrite (iluminaBarrera8, HIGH);
        delay (retardol);
        digitalWrite (iluminaBarrera9, HIGH);
        delay (retardol);
        digitalWrite (iluminaBarrera10, HIGH);
        delay (retardol);
        digitalWrite (iluminaBarrera11, HIGH);
        digitalWrite (iluminaBarrera12, LOW);
        delay (retardo2);
        digitalWrite (iluminaBarrera11, LOW);
        delay (retardo2);
        digitalWrite (iluminaBarrera10, LOW);
    }
    void iluminacionBarrera12 () {
        digitalWrite (iluminaBarrera1, HIGH);
        delay (retardol);
        digitalWrite (iluminaBarrera2, HIGH);
        delay (retardol);
        digitalWrite (iluminaBarrera3, HIGH);
        delay (retardol);
        digitalWrite (iluminaBarrera4, HIGH);
        delay (retardol);
        digitalWrite (iluminaBarrera5, HIGH);
        delay (retardol);
        digitalWrite (iluminaBarrera6, HIGH);
        delay (retardol);
        digitalWrite (iluminaBarrera7, HIGH);
        delay (retardol);
        digitalWrite (iluminaBarrera8, HIGH);
        delay (retardol);
        digitalWrite (iluminaBarrera9, HIGH);
        delay (retardol);
        digitalWrite (iluminaBarrera10, HIGH);
        delay (retardol);
        digitalWrite (iluminaBarrera11, HIGH);
        delay (retardol);
        digitalWrite (iluminaBarrera12, HIGH);
        delay (retardo2);
        digitalWrite (iluminaBarrera12, LOW);
        delay (retardo2);
        digitalWrite (iluminaBarrera11, LOW);
    }
    void iluminacionBlink1 () {

        digitalWrite (iluminaBarrera1, LOW);
        delay (retardol);
        digitalWrite (iluminaBarrera1, HIGH);
        delay (retardol);
        digitalWrite (iluminaBarrera1, LOW);
    }

```

```

        delay(retardol);
        digitalWrite(iluminaBarrera1,HIGH);
        delay(retardol);
        digitalWrite(iluminaBarrera1,LOW);
        delay(retardol);
        digitalWrite(iluminaBarrera1,HIGH);
        delay(retardol);
        digitalWrite(iluminaBarrera1,LOW);
        delay(retardol);
        digitalWrite(iluminaBarrera1,HIGH);
        delay(retardol);
        digitalWrite(iluminaBarrera1,LOW);
        delay(retardol);
        digitalWrite(iluminaBarrera1,HIGH);
        delay(retardol);
        digitalWrite(iluminaBarrera1,LOW);
        delay(retardol);
        digitalWrite(iluminaBarrera1,HIGH);
    }
    void iluminacionBlink2() {
    //blink
        digitalWrite(iluminaBarrera1,HIGH);
        digitalWrite(iluminaBarrera2,HIGH);
        delay(retardol);
        digitalWrite(iluminaBarrera1,LOW);
        digitalWrite(iluminaBarrera2,LOW);
        digitalWrite(iluminaBarrera3,LOW);
        delay(retardol);
        digitalWrite(iluminaBarrera1,HIGH);
        digitalWrite(iluminaBarrera2,HIGH);
        delay(retardol);
        digitalWrite(iluminaBarrera1,LOW);
        digitalWrite(iluminaBarrera2,LOW);
        delay(retardol);
        digitalWrite(iluminaBarrera1,HIGH);
        digitalWrite(iluminaBarrera2,HIGH);
    }
    void iluminacionBlink3() {
    //blink
        digitalWrite(iluminaBarrera1,HIGH);
        digitalWrite(iluminaBarrera2,HIGH);
        digitalWrite(iluminaBarrera3,HIGH);
        delay(retardol);
        digitalWrite(iluminaBarrera1,LOW);
        digitalWrite(iluminaBarrera2,LOW);
        digitalWrite(iluminaBarrera3,LOW);
        delay(retardol);
        digitalWrite(iluminaBarrera1,HIGH);
        digitalWrite(iluminaBarrera2,HIGH);
        digitalWrite(iluminaBarrera3,HIGH);
        delay(retardol);
        digitalWrite(iluminaBarrera1,LOW);
        digitalWrite(iluminaBarrera2,LOW);
        digitalWrite(iluminaBarrera3,LOW);
        delay(retardol);
        digitalWrite(iluminaBarrera1,HIGH);
        digitalWrite(iluminaBarrera2,HIGH);
        digitalWrite(iluminaBarrera3,HIGH);
    }
}

```

```

// función para mostrar datos en la pantalla
void informacionOtrosParametros() {

    lcd.setCursor(0,0);
    lcd.print("V:");
    lcd.print(velocidadCalculada1);
    lcd.print("m/s");

    lcd.setCursor(11,0);
    lcd.print("D:");
    lcd.print(distanciaPuntual);
    lcd.print("cm");

    lcd.setCursor(8,1);
    lcd.print("Pt:");
    lcd.print(tramo);
    lcd.print("s");

}

// POGRAMACION DEL FUNCIONAMIENTO DEL SISTEMA

void loop() {
    // comprobación del estado del ambiente
    valorLDR=analogRead(sensorLDR);
    Serial.print("valor LDR: ");
    Serial.println(valorLDR);

    if (valorLDR<900){//mucho iluminación
    cambio_estados_dia
        lcd.setCursor(0,0);
        lcd.print("ESTADO: DIA");
        lcd.setCursor(0,1);
        lcd.print("VISIBILIDAD 100%");

    }// fin DIA

    else if(valorLDR>900){// poca iluminación o situación nocturna

        lcd.setCursor(0,0);
        lcd.print("ESTADO NOCTURNO");
        lcd.setCursor(0,1);
        lcd.print("POCA VISIBILIDAD");

        inicioDelControl();
        lcd.clear();
        situacionEnEstadoRojo();
        lcd.clear(); //
    }//fin sistema nocturno

}

} // Fin del loop

//FUNCION PARA LAS LECTURAS
void lecturasBarreras(int receptorIR1, int receptorIR2, int
receptorIR3) {

int lectorIR1=analogRead(receptorIR1);
int lectorIR2=analogRead(receptorIR2);
int lectorIR3=analogRead(receptorIR3);

```

```

}

// FUNCIONES PARA EL COMPOTAMIENTO EN ESTADO ROJO Y EN ESTADO VERDE CON
PEATONES CRUZANDO
void situacionEnEstadoRojo() {
    int tiempoEsperaEnRojo=0; // contador de tiempo en rojo
    digitalWrite(emisorIR1,HIGH);
    digitalWrite(emisorIR2,HIGH);
    digitalWrite(emisorIR3,HIGH)

do{

    tiempoEsperaEnRojo=tiempoEsperaEnRojo+50;
    lecturasBarreras(receptorIR1,receptorIR2,receptorIR3);
    if(lectorIR1<100) {
        T1=millis(); // tiempo de corte lector1
        digitalWrite(acceso_sistema,HIGH);
        delay(100);
        digitalWrite(acceso_sistema,LOW);
        distanciaPuntual=distanciaDefinida;//distancia de corte

        break;
    }
    else if(lectorIR2<50) { // corte de lector2
        T2=millis();
        velocidadCalculada1=0;
        digitalWrite(emisorIR1,LOW);

        Serial.print("lector2: ");
        Serial.println(lectorIR2);

        digitalWrite(acceso_sistema,HIGH);
        delay(50);
        digitalWrite(acceso_sistema,LOW);

        break;
    }
    else if(lectorIR3<50) { // corte de lector 3
        T3=millis();
        velocidadCalculada1=0;
        digitalWrite(emisorIR2,LOW);
        digitalWrite(emisorIR1,LOW);

        Serial.print("lector3: ");
        Serial.println(lectorIR3);

        digitalWrite(acceso_sistema,HIGH);
        delay(50);
        digitalWrite(acceso_sistema,LOW);

        break;
    }

}while(rojo&&tiempoEsperaEnRojo<DuracionMaximaRojo); // fin cortes
1,2,3

//***** CORTES 2,3, 4*****

    lcd.clear();
    double tiempoIntermedio=0;
    digitalWrite(emisorIR3,HIGH);
    digitalWrite(emisorIR4,HIGH);

```

```

do{
  tiempoEsperaEnRojo=tiempoEsperaEnRojo+50;
  Serial.print("tiempo espera: ");
  Serial.println(tiempoEsperaEnRojo);
  lecturasBarreras (receptorIR2, receptorIR3, receptorIR4);
  if(lectorIR1<50){ // se ha producido corte 2 y comparamos 1y2
    T2=millis(); //
    Serial.print("lector2: ");
    Serial.println(lectorIR1);
    digitalWrite(acceso_sistema,HIGH);
    delay(100);
    digitalWrite(acceso_sistema,LOW);
    distanciaPuntual=distanciaPuntual-distanciaTramo; //
    digitalWrite(emisorIR1,LOW);

// CALCULO DE TIEMPO Y VELOCIDAD
    tramo=(T2-T1)/1000; // duración entre barreras 1y2
    velocidadCalculada1=distanciaTramo*K/tramo;
    velocidadCalculada2=0;

// CALCULAMOS EL ERROR y COMPROBACION DEL PIR
    informacionOtrosParametros();
    informacionPir();
    calculoError();

// COMPARACIONES DE LAS CONDICIONES

    if (Error<=ErrorMin){
      iluminacionBarrera0();
    }

    if (Error>ErrorMin&&Error<=ErrorMax && estadoPIR==HIGH){
      iluminacionBarrera9();
    }

    else if(Error>ErrorMin&&Error<=ErrorMax&& estadoPIR==LOW){
      iluminacionBarrera8();
    }

    else if(Error>ErrorMax && estadoPIR==LOW){
      iluminacionBarrera11();
    }

    else if(Error>ErrorMax && estadoPIR==HIGH){
      iluminacionBarrera12();
    }
    break;
  }
  else if(lectorIR2<50){ // corte lector 3 tramo 1-3

    T3=millis();
    Serial.print("lector3: ");
    Serial.println(lectorIR1);
    digitalWrite(acceso_sistema,HIGH);
    delay(100);
    digitalWrite(acceso_sistema,LOW);
    distanciaPuntual=distanciaDefinida-2*distanciaTramo;
    digitalWrite(emisorIR2,LOW);
    digitalWrite(emisorIR1,LOW);

    tramo=(T3-T1)/1000; // duración entre barreras 1y3

```

```

    velocidadCalculada1=2*distanciaTramo*K/tramo;
    velocidadCalculada2=0;

// CALCULAMOS EL ERROR Y COMPROBACION DEL PIR
    informacionPir();
    calculoError();
    informacionOtrosParametros();

// CONDICIONES Y SALIDAS ADOPTADAS
    if (Error<=ErrorMin){
        iluminacionBarrera0();
    }

    if (Error>ErrorMin&&Error<=ErrorMax&& estadoPIR==HIGH &&
velocidadCalculada2>velocidadCalculada1){
        iluminacionBarrera7();
    }

    if (Error>ErrorMin&&Error<=ErrorMax && estadoPIR==HIGH &&
velocidadCalculada2<velocidadCalculada1){
        iluminacionBarrera8();
    }

    else if(Error>ErrorMin&&Error<=ErrorMax&& estadoPIR==LOW ){
        iluminacionBarrera7();

    }

    else if(Error>ErrorMax&& estadoPIR==LOW){
        iluminacionBarrera9();

    }

    else if(Error>ErrorMax
&&estadoPIR==HIGH&&velocidadCalculada2>velocidadCalculada1){
        iluminacionBarrera9();
    }

    else if(Error>ErrorMax&&
estadoPIR==HIGH&&velocidadCalculada2<velocidadCalculada1){
        iluminacionBarrera10();

    }

    break;
}
else if(lectorIR3<100){ // corte del lector 4
    T4=millis();
    Serial.print("lector4: ");
    Serial.println(lectorIR3);
    digitalWrite(acceso_sistema,HIGH);
    delay(100);
    digitalWrite(acceso_sistema,LOW);
    distanciaPuntual=distanciaDefinida-3*distanciaTramo;
    digitalWrite(emisorIR3,LOW);
    digitalWrite(emisorIR2,LOW);

//***** CALCULO DE TIEMPO Y VELOCIDAD
    tramo=(T4-T1)/1000; // duración entre barreras 1y4

    velocidadCalculada2=velocidadCalculada1;

```

```

    velocidadCalculada1=3*distanciaTramo*K/tramo;

// CALCULAMOS EL ERROR Y COMPROBACION DEL PIR
    informacionPir();
    calculoError();
    informacionOtrosParametros();

// CONDINES Y SALIDAS ADOPTADAS

    if (Error<=ErrorMin){
        iluminacionBarrera0();
    }

    if (Error>ErrorMin&&Error<=ErrorMax&& estadoPIR==HIGH
&&velocidadCalculada2>velocidadCalculada1){
        iluminacionBarrera6();
    }

    if (Error>ErrorMin&&Error<=ErrorMax&& estadoPIR==HIGH
&&velocidadCalculada2<velocidadCalculada1){
        iluminacionBarrera7();
    }

    else if(Error>ErrorMin&&Error<=ErrorMax&& estadoPIR==LOW
&&velocidadCalculada2>velocidadCalculada1){
        iluminacionBarrera5();
    }

    else if(Error>ErrorMin&&Error<=ErrorMax&& estadoPIR==LOW
&&velocidadCalculada2<velocidadCalculada1){
        iluminacionBarrera6();
    }

    else if(Error>ErrorMax&& estadoPIR==LOW ){
        iluminacionBarrera5();
    }

    else if(Error>ErrorMax&& estadoPIR==HIGH
&&velocidadCalculada2<velocidadCalculada1){
        iluminacionBarrera8();
    }

    else if(Error>ErrorMax&&
estadoPIR==HIGH&&velocidadCalculada2>velocidadCalculada1){
        iluminacionBarrera7();
    }

    break;
}

}while(rojo &&tiempoEsperaEnRojo<DuracionMaximaRojo);//fin cortes
2,3,4

// *****CORTES 3,4, 5*****

    digitalWrite(emisorIR4,HIGH);
    digitalWrite(emisorIR5,HIGH);
    do{

        tiempoEsperaEnRojo=tiempoEsperaEnRojo+50;

```

```

lecturasBarreras (receptorIR3, receptorIR4, receptorIR5);
if(lectorIR1<50){ // corte lector 3
  T3=millis();
  Serial.print("lector3: ");
  Serial.println(lectorIR1);
  digitalWrite(acceso_sistema,HIGH);
  delay(100);
  digitalWrite(acceso_sistema,LOW);
  distanciaPuntual=distanciaPuntual-distanciaTramo;
  digitalWrite(emisorIR2,LOW);
  digitalWrite(emisorIR3,LOW);

  /****** CALCULO DE TIEMPO Y VELOCIDAD
  tramo=(T3-T2)/1000; // duración entre barreras 2y3

  velocidadCalculada2=velocidadCalculada1;
  velocidadCalculada1=distanciaTramo*K/tramo;

  // CALCULAMOS EL ERROR y COMPROBACION DEL PIR
  informacionPir();
  calculoError();
  informacionOtrosParametros();

  // CONDICIONES SALIDAS ADOPTADAS

  if (Error<=ErrorMin){
    iluminacionBarrera0();
  }

  if (Error>ErrorMin&&Error<=ErrorMax&& estadoPIR==HIGH &&
  velocidadCalculada2>velocidadCalculada1){
    iluminacionBarrera7();
  }

  if (Error>ErrorMin&&Error<=ErrorMax&& estadoPIR==HIGH &&
  velocidadCalculada2<velocidadCalculada1){
    iluminacionBarrera8();
  }

  else if(Error>ErrorMin&&Error<=ErrorMax&& estadoPIR==LOW ){
    iluminacionBarrera7();
  }

  else if(Error>ErrorMax && estadoPIR==LOW){
    iluminacionBarrera9();
  }

  else if(Error>ErrorMax && estadoPIR==HIGH
  &&velocidadCalculada2>velocidadCalculada1){
    iluminacionBarrera9();
  }

  else if(Error>ErrorMax && estadoPIR==HIGH
  &&velocidadCalculada2<velocidadCalculada1){
    iluminacionBarrera10();
  }

  break;
}
else if(lectorIR2<100){ // corte lector 4

```

```

T4=millis();
Serial.print("lector4: ");
Serial.println(lectorIR2);
digitalWrite(acceso_sistema,HIGH);
delay(100);
digitalWrite(acceso_sistema,LOW);
distanciaPuntual=distanciaPuntual-2*distanciaTramo;
digitalWrite(emisorIR3,LOW);
digitalWrite(emisorIR4,LOW);

//***** CALCULO DE TIEMPO Y VELOCIDAD
tramo=(T4-T2)/1000; // duración entre barreras 2y4

velocidadCalculada2=velocidadCalculada1;
velocidadCalculada1=2*distanciaTramo*K/tramo;

// CALCULAMOS EL ERROR Y COMPROBACION DEL PIR
informacionPir();
calculoError();
informacionOtrosParametros();

// CONDICIONES SALIDAS ADOPTADAS

if (Error<=ErrorMin){
    iluminacionBarrera0();
}

if (Error>ErrorMin&&Error<=ErrorMax && estadoPIR==HIGH
&&velocidadCalculada2>velocidadCalculada1){
    iluminacionBarrera6();
}

if (Error>ErrorMin&&Error<=ErrorMax&& estadoPIR==HIGH
&&velocidadCalculada2<velocidadCalculada1){
    iluminacionBarrera7();
}

else if(Error>ErrorMin&&Error<=ErrorMax&& estadoPIR==LOW
&&velocidadCalculada2>velocidadCalculada1){
    iluminacionBarrera5();
}

else if(Error>ErrorMin&&Error<=ErrorMax && estadoPIR==LOW
&&velocidadCalculada2<velocidadCalculada1){
    iluminacionBarrera6();
}

else if(Error>=ErrorMax&& estadoPIR==LOW ){
    iluminacionBarrera5();
}

else if(Error>=ErrorMax && estadoPIR==HIGH
&&velocidadCalculada2<velocidadCalculada1){
    iluminacionBarrera8();
}

else if(Error>=ErrorMax &&
estadoPIR==HIGH&&velocidadCalculada2>velocidadCalculada1){
    iluminacionBarrera7();
}
break;

```

```

}

else if(lectorIR3<100){ // corte lector 5
  T5=millis();
  Serial.print("lector5: ");
  Serial.println(lectorIR3);
  digitalWrite(acceso_sistema,HIGH);
  delay(100);
  digitalWrite(acceso_sistema,LOW);
  distanciaPuntual=distanciaPuntual-3*distanciaTramo
  digitalWrite(emisorIR4,LOW);
  digitalWrite(emisorIR3,LOW);
  digitalWrite(emisorIR5,LOW);

//***** CALCULO DE TIEMPO Y VELOCIDAD
  tramo=(T5-T2)/1000; // duración entre barreras 2y5

  velocidadCalculada2=velocidadCalculada1;
  velocidadCalculada1=3*distanciaTramo*K/tramo;

// CALCULAMOS EL ERROR y COMPROBACION DEL PIR
  calculoError();
  informacionPir();
  informacionOtrosParametros();

// CONDICIONES Y SALIDAS ADOPTADAS
  if (Error<=ErrorMin){
    iluminacionBarrera0();
  }

  if (Error>ErrorMin&&Error<=ErrorMax&& estadoPIR==HIGH
&&velocidadCalculada2>velocidadCalculada1){
    iluminacionBarrera3();
  }

  if (Error>ErrorMin&&Error<=ErrorMax&& estadoPIR==HIGH
&&velocidadCalculada2<velocidadCalculada1){
    iluminacionBarrera5();
  }

  }

  else if(Error>ErrorMin&&Error<=ErrorMax&& estadoPIR==LOW){
    iluminacionBarrera4();
  }

  else
if(Error>=ErrorMax&&velocidadCalculada2>velocidadCalculada1)
{
  iluminacionBarrera5();
}

  else
if(Error>=ErrorMax&&velocidadCalculada2<velocidadCalculada1)
{
  iluminacionBarrera6();
}

  }
  break;
}
}while(rojo&& tiempoEsperaEnRojo<DuracionMaximaRojo); // fin cortes 3,
4, 5

```

```

//***** CORTES 4, 5, 6*****

digitalWrite(emisorIR5,HIGH);
digitalWrite(emisorIR6,HIGH);

tiempoIntermedio=0; // es el tiempo que transcurre una vez
realizada una lectura hasta espera k se realice la siguiente.
do{
tiempoEsperaEnRojo=tiempoEsperaEnRojo+50;
lecturasBarreras (receptorIR4,receptorIR5,receptorIR6);
if(lectorIR1<100){ // corte lector 4
T4=millis
Serial.print("lector4: ");
Serial.println(lectorIR1);
digitalWrite(acceso_sistema,HIGH);
delay(100);
digitalWrite(acceso_sistema,LOW);
distanciaPuntual=distanciaPuntual-distanciaTramo;
digitalWrite(emisorIR3,LOW);
digitalWrite(emisorIR4,LOW);

//***** CALCULO DE TIEMPO Y VELOCIDAD
tramo=(T4-T3)/1000; // se obtiene tiempo entre barreras 2y3

velocidadCalculada2=velocidadCalculada1;
velocidadCalculada1=distanciaTramo*K/tramo;

// CALCULAMOS EL ERROR y COMPROBACION DEL PIR
calculoError();
informacionPir();
informacionOtrosParametros();

// COMPARACIONES DE LAS CONDICIONES BARRERA 4

if (Error<=ErrorMin){
iluminacionBarrera0();
}

if (Error>ErrorMin&&Error<=ErrorMax&& estadoPIR==HIGH
&&velocidadCalculada2>velocidadCalculada1){
iluminacionBarrera6();
}

if (Error>ErrorMin&&Error<=ErrorMax&& estadoPIR==HIGH
&&velocidadCalculada2<velocidadCalculada1){
iluminacionBarrera7();
}

else if(Error>ErrorMin&&Error<=ErrorMax&& estadoPIR==LOW
&&velocidadCalculada2>velocidadCalculada1){
iluminacionBarrera5();
}

else if(Error>ErrorMin&&Error<=ErrorMax&& estadoPIR==LOW
&&velocidadCalculada2<velocidadCalculada1){
iluminacionBarrera6();
}

else if(Error>=ErrorMax && estadoPIR==LOW ){

```

```

        iluminacionBarrera5();
    }

    else if(Error>=ErrorMax&& estadoPIR==HIGH
&&velocidadCalculada2<velocidadCalculada1){
        iluminacionBarrera8();
    }

    else if(Error>=ErrorMax&&
estadoPIR==HIGH&&velocidadCalculada2>velocidadCalculada1){

        iluminacionBarrera7();
    }
    break;
}
else if(lectorIR2<100){ // corte lector 5
    T5=millis
    Serial.print("lector5: ");
    Serial.println(lectorIR2);
    digitalWrite(acceso_sistema,HIGH);
    delay(100);
    digitalWrite(acceso_sistema,LOW);
    distanciaPuntual=distanciaPuntual-2*distanciaTramo
    digitalWrite(emisorIR4,LOW);
    digitalWrite(emisorIR5,LOW);

//***** CALCULO DE TIEMPO Y VELOCIDAD
    tramo=(T5-T3)/1000; duración entre barreras 3y5

    velocidadCalculada2=velocidadCalculada1;
    velocidadCalculada1=2*distanciaTramo*K/tramo;

// CALCULAMOS EL ERROR y COMPROBACION DEL PIR
    calculoError();
    informacionPir();
    informacionOtrosParametros();

// COMPARACIONES DE LAS CONDICIONES BARRERA 5

    if (Error<=ErrorMin){
        iluminacionBarrera0();
    }

    if (Error>ErrorMin&&Error<=ErrorMax && estadoPIR==HIGH
&&velocidadCalculada2>velocidadCalculada1){
        iluminacionBarrera3();
    }

    if (Error>ErrorMin&&Error<=ErrorMax&& estadoPIR==HIGH
&&velocidadCalculada2<velocidadCalculada1){
        iluminacionBarrera5();
    }

    }

    else if(Error>ErrorMin&&Error<=ErrorMax&& estadoPIR==LOW){
        iluminacionBarrera4();
    }

    }

    elseif(Error>=ErrorMax&&velocidadCalculada2>velocidadCalcul
ada1){
        iluminacionBarrera5();
    }

```

```

    }

    elseif(Error>=ErrorMax&&velocidadCalculada2<velocidadCalculada1){
        iluminacionBarrera6();

    }
    break;
}
else if(lectorIR3<100){ // corte lector 6
    T6=millis();
    Serial.print("lector6: ");
    Serial.println(lectorIR3);
    digitalWrite(acceso_sistema,HIGH);
    delay(100);
    digitalWrite(acceso_sistema,LOW);
    distanciaPuntual=distanciaPuntual-3*distanciaTramo
    digitalWrite(emisorIR5,LOW);
    digitalWrite(emisorIR4,LOW);
    digitalWrite(emisorIR6,LOW);

//***** CALCULO DE TIEMPO Y VELOCIDAD
    tramo=(T6-T3)/1000; // duración entre barreras 4y6

    velocidadCalculada2=velocidadCalculada1;
    velocidadCalculada1=3*distanciaTramo*K/tramo;

// CALCULAMOS EL ERROR y COMPROBACION DEL PIR
    calculoError();
    informacionPir();
    informacionOtrosParametros();

//SALIDAS Y CONDICIONES ADOPTADAS
    if (Error<=ErrorMin){
        iluminacionBarrera0();
    }

    else if(Error>ErrorMin && estadoPIR==LOW){
        iluminacionBarrera3();
        iluminacionBlink2();

    }

    else if(Error>ErrorMin&& estadoPIR==HIGH){
        iluminacionBarrera4();
        iluminacionBlink3();

    }
    break;
}

}while(rojo&&tiempoEsperaEnRojo<DuracionMaximaRojo); //fin cortes
4,5,6

// *****CORTE 5 Y 6*****

    tiempoIntermedio=0;
    digitalWrite(emisorIR6,HIGH);
    do{

```

```

tiempoEsperaEnRojo=tiempoEsperaEnRojo+50;
lecturasBarreras (receptorIR5, receptorIR6, receptorIR7);
if(lectorIR1<100){ // corte lector 5
    T5=millis();
    Serial.print("lector5: ");
    Serial.println(lectorIR1);
    digitalWrite(acceso_sistema, HIGH);
    delay(100);
    digitalWrite(acceso_sistema, LOW);
    distanciaPuntual=distanciaPuntual-distanciaTramo;
    digitalWrite(emisorIR4, LOW);
    digitalWrite(emisorIR5, LOW);

//***** CALCULO DE TIEMPO Y VELOCIDAD
    tramo=(T5-T4)/1000; // duración entre barreras 4y5

    velocidadCalculada2=velocidadCalculada1;
    velocidadCalculada1=distanciaTramo*K/tramo;

// CALCULAMOS EL ERROR y COMPROBACION DEL PIR
    calculoError();
    informacionPir();
    informacionOtrosParametros();

// CONDICIONES Y SALIDA ADOPTADAS

    if (Error<=ErrorMin){
        iluminacionBarrera0;
    }

    if (Error>ErrorMin&&Error<=ErrorMax&& estadoPIR==HIGH
&&velocidadCalculada2>velocidadCalculada1){
        iluminacionBarrera3();
    }

    if (Error>ErrorMin&&Error<=ErrorMax&& estadoPIR==HIGH
&&velocidadCalculada2<velocidadCalculada1){
        iluminacionBarrera5();
    }

    else if(Error>ErrorMin&&Error<=ErrorMax&& estadoPIR==LOW){
        iluminacionBarrera4();
    }

    else
if(Error>=ErrorMax&&velocidadCalculada2>velocidadCalculada1)
{
    iluminacionBarrera5();
}

    else
if(Error>=ErrorMax&&velocidadCalculada2<velocidadCalculada1)
{
    iluminacionBarrera6();
}

    break;
}
else if(lectorIR2<100){ // corte lector 6

```

```

        T6=millis();
        Serial.print("lector6: ");
        Serial.println(lectorIR2);
        digitalWrite(acceso_sistema,HIGH);
        delay(100);
        digitalWrite(acceso_sistema,LOW);
        distanciaPuntual=distanciaPuntual-2*distanciaTramo;
        digitalWrite(emisorIR5,LOW);
        digitalWrite(emisorIR6,LOW);

//***** CALCULO DE TIEMPO Y VELOCIDAD
        tramo=(T6-T4)/1000; // duración entre barreras 4y6

        velocidadCalculada2=velocidadCalculada1;
        velocidadCalculada1=2*distanciaTramo*K/tramo;

// CALCULAMOS EL ERROR Y COMPROBACION DEL PIR
        calculoError();
        informacionPir();
        informacionOtrosParametros();

// CONDICIONES Y SALIDAS ADOPTADAS
        if (Error<=ErrorMin){
            iluminacionBarrera0();

        }

        else if(Error>ErrorMin && estadoPIR==LOW){
            iluminacionBarrera3();
            iluminacionBlink2();
        }

        else if(Error>ErrorMin && estadoPIR==HIGH){
            iluminacionBarrera4();
            iluminacionBlink3();
        }
        break;
    }

}while(rojo&& tiempoEsperaEnRojo<DuracionMaximaRojo); // fin
cortes 5Y6

// *****CORTES 6 Y 7*****

        tiempoIntermedio=0;
        digitalWrite(emisorIR7,HIGH);
        do{

            tiempoEsperaEnRojo=tiempoEsperaEnRojo+50;
            lecturasBarreras(receptorIR6,receptorIR7,receptorIR3);
            if(lectorIR1<100){ // corte lector 6
                T6=millis();
                Serial.print("lector6: ");
                Serial.println(lectorIR1);
                digitalWrite(acceso_sistema,HIGH);
                delay(100);
                digitalWrite(acceso_sistema,LOW);
                distanciaPuntual=distanciaPuntual-distanciaTramo
                digitalWrite(emisorIR6,LOW);

// CALCULO DE TIEMPO Y VELOCIDAD

```

```

    tramo=(T6-T5)/1000; // duración entre barreras 5y6

    velocidadCalculada2=velocidadCalculada1;
    velocidadCalculada1=distanciaTramo*K/tramo;

// CALCULAMOS EL ERROR y COMPROBACION DEL PIR
    calculoError();
    informacionPir();
    informacionOtrosParametros();

// CONDICIONES Y SALIDAS ADOPTADAS

    if (Error<=ErrorMin){

        iluminacionBarrera0();
    }

    else if(Error>ErrorMin && estadoPIR==LOW){
        iluminacionBarrera3();
        iluminacionBlink2();
    }

    else if(Error>ErrorMin && estadoPIR==HIGH){
        iluminacionBarrera4();
        iluminacionBlink3();
    }

    break;
}
else if(lectorIR2<75){ // corte lector 7
    T7=millis();
    Serial.print("lector7: ");
    Serial.println(lectorIR2);
    digitalWrite(acceso_sistema,HIGH);
    delay(100);
    digitalWrite(acceso_sistema,LOW);
    distanciaPuntual=distanciaPuntual-2*distanciaTramo;
    digitalWrite(emisorIR5,LOW);
    digitalWrite(emisorIR6,LOW);

// CALCULO DE TIEMPO Y VELOCIDAD
    tramo=(T7-T5)/1000; // duración entre barreras 5y7

    velocidadCalculada2=velocidadCalculada1;
    velocidadCalculada1=2*distanciaTramo*K/tramo;

// CALCULAMOS EL ERROR y COMPROBACION DEL PIR
    calculoError();
    informacionPir();
    informacionOtrosParametros();

// CONDICIONES Y SALIDAS ADOPTADAS

    if (Error<=ErrorMin){
        iluminacionBarrera0();
    }

    else if(Error>ErrorMin&& estadoPIR==LOW){
        iluminacionBarrera1();
        iluminacionBlink1();

```

```

    }

    else if(Error>ErrorMin && estadoPIR==HIGH) {

        iluminacionBarrera1();
        iluminacionBlink2();
    }
    break;
}

}while(rojo&& tiempoEsperaEnRojo<DuracionMaximaRojo); // fin cortes
6Y7

//*****CORTE 7 FIN TRAMO*****
digitalWrite(emisorIR6,LOW);
tiempoIntermedio=0;
do{

    tiempoEsperaEnRojo=tiempoEsperaEnRojo+50;
    lecturasBarreras (receptorIR7, receptorIR1, receptorIR2);
    if(lectorIR1<50){ // se ha producido corte 7 y comparamos 6y7
        T7=millis(); // tomamos el tiempo en que se realizo el
corte desde que se inicio el programa
        Serial.print("lector7: ");
        Serial.println(lectorIR1);
        digitalWrite(acceso_sistema,HIGH);
        delay(100);
        digitalWrite(acceso_sistema,LOW);
        distanciaPuntual=distanciaPuntual-distanciaTramo; // punto
exacto donde se realizo el primer corte
        digitalWrite(emisorIR7,LOW);
        digitalWrite(emisorIR6,LOW);

// CALCULO DE TIEMPO Y VELOCIDAD
        tramo=(T7-T6)/1000; // se obtiene tiempo entre barreras 6y7

        velocidadCalculada2=velocidadCalculada1;
        velocidadCalculada1=distanciaTramo*K/tramo;

// CALCULAMOS EL ERROR y COMPROBACION DEL PIR
        calculoError();
        informacionPir();
        informacionOtrosParametros();

        if (Error<=ErrorMin) {
            iluminacionBarrera0();
        }

        else if(Error>ErrorMin && estadoPIR==LOW) {
            iluminacionBarrera1();
            iluminacionBlink1();
        }

        else if(Error>ErrorMin && estadoPIR==HIGH) {

            iluminacionBarrera1();
            iluminacionBlink2();
        }
        break;
}

```

```

    }
}while(rojo&& tiempoEsperaEnRojo<DuracionMaximaRojo); // fin
cortes 7

    delay(500 );
    iluminacionBarrera0();

int restante=DuracionMaximaRojo-tiempoEsperaEnRojo;

if (tiempoEsperaEnRojo<DuracionMaximaRojo){
    delay(restante);
    digitalWrite(rojo,LOW);
}
else{
    delay(1000);
    digitalWrite(rojo,LOW);
}
digitalWrite(emisorIR7,LOW);

lcd.clear();
}

void situacionDePeligro(){
    digitalWrite(emisorIR3,HIGH);
    digitalWrite(emisorIR4,HIGH);
    digitalWrite(emisorIR5,HIGH);
    int tiempoEsperaEnVerde=0;
do{
    tiempoEsperaEnVerde=tiempoEsperaEnVerde+50;
    lecturasBarreras(receptorIR3,receptorIR4,receptorIR5);
    if(lectorIR1<50){
        T3=millis();
        digitalWrite(acceso_sistema,HIGH);
        delay(retardol);
        digitalWrite(acceso_sistema,LOW);
        distanciaPuntual=distanciaDefinida-2*distanciaTramo

        lcd.setCursor(11,0);
        lcd.print("D:");
        lcd.print(distanciaPuntual);
        lcd.print("cm");

        break;
    }
    else if(lectorIR2<80
        T4=millis();
        velocidadCalculada1=0;
        digitalWrite(emisorIR3,LOW);
        Serial.print("lector4: ");
        Serial.println(lectorIR2);
        digitalWrite(acceso_sistema,HIGH);
        delay(retardol);
        digitalWrite(acceso_sistema,LOW);

        break;
    }
    else if(lectorIR3<50
        T5=millis();
        velocidadCalculada1=0;
        digitalWrite(emisorIR3,LOW);

```

```

        digitalWrite(emisorIR4, LOW);

        Serial.print("lector5: ");
        Serial.println(lectorIR3);

        digitalWrite(acceso_sistema, HIGH);
        delay(retardol);
        digitalWrite(acceso_sistema, LOW);

    break;
}
}while(tiempoEsperaEnVerde<DuracionMaximaVerde); // fin cortes
3,4,5

//***** CORTES 4,5 6*****
digitalWrite(emisorIR5, HIGH);
digitalWrite(emisorIR6, HIGH);
do{
    tiempoEsperaEnVerde=tiempoEsperaEnVerde+50;
    lecturasBarreras (receptorIR4, receptorIR5, receptorIR5);

    if(lectorIR1<100) {
        T4=millis
        Serial.print("lector4: ");
        Serial.println(lectorIR1);
        digitalWrite(acceso_sistema, HIGH);
        delay(100);
        digitalWrite(acceso_sistema, LOW);
        distanciaPuntual=distanciaPuntual-distanciaTramo;
        digitalWrite(emisorIR3, LOW);
        digitalWrite(emisorIR4, LOW);
        tramo=(T4-T3)/1000;
        velocidadCalculada2=velocidadCalculada1;
        velocidadCalculada1=distanciaTramo*K/tramo;

// CALCULAMOS EL ERROR y COMPROBACION DEL PIR
        calculoError();
        informacionPir();
        informacionOtrosParametros();

// CONDICIONES Y SALIDA ADOPTADAS

        if (Error<=ErrorMin) {
            iluminacionBarrera0();
        }

        if (Error>ErrorMin&&Error<=ErrorMax&& estadoPIR==HIGH) {
            iluminacionBarrera6();
        }

        if (Error>ErrorMin&&Error<=ErrorMax&&
estadoPIR==HIGH&&velocidadCalculada2<velocidadCalculada1) {
            iluminacionBarrera7();
        }

        else if(Error>=ErrorMax&&
estadoPIR==HIGH&&velocidadCalculada2<velocidadCalculada1) {
            iluminacionBarrera8();
        }

        else if(Error>=ErrorMax&& estadoPIR==HIGH) {

```

```

        iluminacionBarrera7();
    }
    break;
}
else if(lectorIR2<100) {
    T5=millis();
    Serial.print("lector5: ");
    Serial.println(lectorIR2);
    digitalWrite(acceso_sistema,HIGH);
    delay(100);
    digitalWrite(acceso_sistema,LOW);
    distanciaPuntual=distanciaPuntual-2*distanciaTramo
    digitalWrite(emisorIR4,LOW);
    digitalWrite(emisorIR5,LOW);

    //***** CALCULO DE TIEMPO Y VELOCIDAD
    tramo=(T5-T3)/1000; // se obtiene tiempo entre barreras

    velocidadCalculada2=velocidadCalculada1;
    velocidadCalculada1=2*distanciaTramo*K/tramo;

    // CALCULAMOS EL ERROR y COMPROBACION DEL PIR
    calculoError();
    informacionPir();
    informacionOtrosParametros();
    // COMPARACIONES DE LAS CONDICIONES BARRERA 5

    if (Error<=ErrorMin) {
        iluminacionBarrera0();
    }

    if (Error>ErrorMin&&Error<=ErrorMax && estadoPIR==HIGH) {
        iluminacionBarrera3();
    }

    if (Error>ErrorMin&&Error<=ErrorMax&& estadoPIR==HIGH
    &&velocidadCalculada2<velocidadCalculada1) {
        iluminacionBarrera5();
    }

    elseif(Error>=ErrorMax&&velocidadCalculada2>velocidadCalcul
    ada1) {
        iluminacionBarrera5();
    }

    elseif(Error>=ErrorMax&&velocidadCalculada2<velocidadCalcul
    ada1) {
        iluminacionBarrera6();
    }
    break;
}
else if(lectorIR3<100) {
    T6=millis();
    Serial.print("lector6: ");
    Serial.println(lectorIR3);
    digitalWrite(acceso_sistema,HIGH);

```

3y5

```

        delay(100);
        digitalWrite(acceso_sistema,LOW);
        distanciaPuntual=distanciaPuntual-3*distanciaTramo
        digitalWrite(emisorIR5,LOW);
        digitalWrite(emisorIR4,LOW);
        digitalWrite(emisorIR6,LOW);

//***** CALCULO DE TIEMPO Y VELOCIDAD
        tramo=(T6-T3)/1000;
        velocidadCalculada2=velocidadCalculada1;
        velocidadCalculada1=3*distanciaTramo*K/tramo;

// CALCULAMOS EL ERROR Y COMPROBACION DEL PIR
        calculoError();
        informacionPir();
        informacionOtrosParametros();

        if (Error<=ErrorMin){

                iluminacionBarrera0();
        }

        else if(Error>ErrorMin&& estadoPIR==HIGH){
                iluminacionBarrera4();
                iluminacionBlink3();
        }
        break;
}

}while(tiempoEsperaEnVerde<DuracionMaximaVerde); // fin cortes
4,5,6

//*****CORTES 5 Y 6 7*****

digitalWrite(emisorIR6,HIGH);
do{

        tiempoEsperaEnVerde=tiempoEsperaEnVerde+50;
        lecturasBarreras(receptorIR5,receptorIR6,receptorIR1);
        if(lectorIR1<100){
                T5=millis();
                Serial.print("lector5: ");
                Serial.println(lectorIR1);
                digitalWrite(acceso_sistema,HIGH);
                delay(100);
                digitalWrite(acceso_sistema,LOW);
                distanciaPuntual=distanciaPuntual-distanciaTramo
                digitalWrite(emisorIR4,LOW);
                digitalWrite(emisorIR5,LOW);

                //***** CALCULO DE TIEMPO Y VELOCIDAD
                tramo=(T5-T4)/1000
                velocidadCalculada2=velocidadCalculada1;
                velocidadCalculada1=distanciaTramo*K/tramo;

// CALCULAMOS EL ERROR Y COMPROBACION DEL PIR
                calculoError();
                informacionPir();
                informacionOtrosParametros();

```

```

// CONDICIONES Y SALIDA ADOPTADAS
    if (Error<=ErrorMin) {
        iluminacionBarrera0;
    }

    if (Error>ErrorMin&&Error<=ErrorMax&& estadoPIR==HIGH) {
        iluminacionBarrera3();
    }

    if (Error>ErrorMin&&Error<=ErrorMax&& estadoPIR==HIGH
&&velocidadCalculada2<velocidadCalculada1) {
        iluminacionBarrera5();
    }

    else if(Error>=ErrorMax&&estadoPIR==HIGH) {
        iluminacionBarrera5();
    }

    else if(Error>=ErrorMax&&estadoPIR==HIGH) {
        iluminacionBarrera6();
    }

    break;
}
else if(lectorIR2<100) {
    T6=millis();
    Serial.print("lector6: ");
    Serial.println(lectorIR2);
    digitalWrite(acceso_sistema,HIGH);
    delay(100);
    digitalWrite(acceso_sistema,LOW);
    distanciaPuntual=distanciaPuntual-2*distanciaTramo
    digitalWrite(emisorIR5,LOW);
    digitalWrite(emisorIR6,LOW);

    /******* CALCULO DE TIEMPO Y VELOCIDAD
    tramo=(T6-T4)/1000;
    velocidadCalculada2=velocidadCalculada1;
    velocidadCalculada1=2*distanciaTramo*K/tramo;

// CALCULAMOS EL ERROR y COMPROBACION DEL PIR
    calculoError();
    informacionPir();
    informacionOtrosParametros();

// CONDICIONES Y SALIDA ADOPTADAS
    if (Error<=ErrorMin){
        iluminacionBarrera0();
    }

    else if(Error>ErrorMin && estadoPIR==HIGH) {
        iluminacionBarrera4();
        iluminacionBlink3();
    }
    break;
}

```

5Y6 7

```

}while(tiempoEsperaEnVerde<DuracionMaximaVerde); // fin cortes

// *****CORTES 6 Y 7*****

digitalWrite(emisorIR7,HIGH);
do{

    tiempoEsperaEnVerde=tiempoEsperaEnVerde+50;
    lecturasBarreras (receptorIR6, receptorIR7, receptorIR2);
    if(lectorIR1<100){
        T6=millis();
        Serial.print("lector6: ");
        Serial.println(lectorIR1);
        digitalWrite(acceso_sistema,HIGH);
        delay(100);
        digitalWrite(acceso_sistema,LOW);
        distanciaPuntual=distanciaPuntual-distanciaTramo
        digitalWrite(emisorIR6,LOW);

        //***** CALCULO DE TIEMPO Y VELOCIDAD
        tramo=(T6-T5)/1000
        velocidadCalculada2=velocidadCalculada1;
        velocidadCalculada1=distanciaTramo*K/tramo;

// CALCULAMOS EL ERROR y COMPROBACION DEL PIR
        calculoError();
        informacionPir();
        informacionOtrosParametros();

// CONDICIONES Y SALIDA ADOPTADAS

        if (Error<=ErrorMin){

            iluminacionBarrera0();
        }

        else if(Error>ErrorMin && estadoPIR==HIGH){
            iluminacionBarrera4();
            iluminacionBlink3();

        }
        break;
    }
else if(lectorIR2<75
    T7=millis();
    Serial.print("lector7: ");
    Serial.println(lectorIR2);
    digitalWrite(acceso_sistema,HIGH);
    delay(100);
    digitalWrite(acceso_sistema,LOW);
    distanciaPuntual=distanciaPuntual-2*distanciaTramo
    digitalWrite(emisorIR5,LOW);
    digitalWrite(emisorIR6,LOW);

// CALCULO DE TIEMPO Y VELOCIDAD
    tramo=(T7-T5)/1000;
    velocidadCalculada2=velocidadCalculada1;
    velocidadCalculada1=2*distanciaTramo*K/tramo;

// CALCULAMOS EL ERROR y COMPROBACION DEL PIR

```

```

        calculoError();
        informacionPir();
        informacionOtrosParametros();

// CONDICIONES Y SALIDA ADOPTADAS
    if (Error<=ErrorMin) {
        iluminacionBarrera0();
    }

    else if(Error>ErrorMin&& estadoPIR==HIGH) {
        iluminacionBarrera1();
        iluminacionBlink1();
    }

    else if(Error>ErrorMin && estadoPIR==HIGH) {
        iluminacionBarrera1();
        iluminacionBlink2();
    }
    break;
}

}while(rojo&& tiempoEsperaEnVerde<DuracionMaximaVerde); // fin cortes
6Y7

//*****CORTES 7 FIN TRAMO*****

digitalWrite(emisorIR6,LOW);

do{

    tiempoEsperaEnVerde=tiempoEsperaEnVerde+50;
    lecturasBarreras (receptorIR7, receptorIR1, receptorIR2);
    if(lectorIR1<50) {
        T7=millis();
        Serial.print("lector7: ");
        Serial.println(lectorIR1);
        digitalWrite(acceso_sistema,HIGH);
        delay(100);
        digitalWrite(acceso_sistema,LOW);
        distanciaPuntual=distanciaPuntual-distanciaTramo; // punto
exacto donde se realizo el primer corte
        digitalWrite(emisorIR7,LOW);
        digitalWrite(emisorIR6,LOW);

        //***** CALCULO DE TIEMPO Y VELOCIDAD
        tramo=(T7-T6)/1000
        velocidadCalculada2=velocidadCalculada1;
        velocidadCalculada1=distanciaTramo*K/tramo;

// CALCULAMOS EL ERROR y COMPROBACION DEL PIR
    calculoError();
    informacionPir();
    informacionOtrosParametros();

    if (Error<=ErrorMin) {
        iluminacionBarrera0();
    }

    else if(Error>ErrorMin && estadoPIR==HIGH) {

```

```

        iluminacionBarrera1();
        iluminacionBlink2();
    }
    break;
}
}while(tiempoEsperaEnVerde<DuracionMaximaVerde); // fin corte 7

    lcd.clear();
    iluminacionBarrera0();

} // fin funcion situacion de peligro

void inicioDelControl(){

    digitalWrite(ambar,HIGH);
    delay(1000);
    digitalWrite(ambar,LOW);
    delay(10);
    digitalWrite(verde,HIGH);
    int tiempoEsperaEnVerde=0;
    situacionDePeligro();
    digitalWrite(verde,LOW);
    delay(10);
    digitalWrite(ambar,HIGH);
    delay(1000);
    digitalWrite(ambar,LOW);
    delay(10);
    digitalWrite(rojo,HIGH);
}

```

4.6. Suposición sobre un entorno real

Este será el apartado en que se explicará todo lo que será necesario para llevar el proyecto en un entorno real y zonas en las que sería útil su funcionalidad. Para ello se va a buscar componentes electrónicos que podrán sustituir o adaptarse a las funciones que desarrollan los que están instalados en el entorno de pruebas. No será necesario cambiar el controlador ya que este puede accionar todo el sistema gracias a la ayuda de los relés conectados entre placa controladora y los dispositivos electrónicos. Se calcula que el código de programación utilizado en el entorno de pruebas será el mismo, aunque puede sufrir pequeñas modificaciones a la hora de calibrar algunos dispositivos ya que se estará trabajando con medidas reales de un entorno real y por lo tanto, esas distancias pueden afectar a la cantidad de señal que se puede recibir o enviar al controlador para la toma de decisiones. Una vez realizado esa suposición se hará un estudio económico de todos esos materiales para su posible adquisición.

- ✚ **Nota:** teniendo en cuenta que el sistema diseñado en este proyecto es un sistema novedoso, lo normal sería que todos los componentes fuesen diseñados de fábrica con unas medidas adecuadas lo que incrementaría altamente el presupuesto. Pero viendo el abanico de opciones presentes en el mercado actual se pueden conseguir componentes que realicen funciones similares aunque estos

tengan que ser reprogramados o calibrados para adaptarlos al sistema.

4.6.1. Zonas con mayor índice de atropellos en España

En todas las ciudades del mundo existen zonas determinadas en las que se registran grandes cifras de atropellos y en base al estudio realizado por la fundación **Mutua madrileña para la seguridad vial** se ha concluido que el **70%** de esos atropellos se producen en los cruces de peatones ya sea debido al mal tiempo meteorológico o a la escasa visibilidad que se pueda presentar en el ambiente. Dicho eso, en este apartado se va a mostrar algunas zonas de España donde sería conveniente instalar este sistema diseñado con el fin de intentar reducir el índice de incidentes producidos en estas zonas.

Existen cientos de calles en toda España donde sería útil esta tecnología pero por mencionar algunas en función a los últimos estudios realizados recientemente

Valencia, Cruce peatonal de la calle Doctor Manuel candela, esta es una calle con mucha circulación ya sea de día o de noche, presenta poca iluminación debido a la elevada altura de los edificios o la vegetación. En esta calle se ha llegado a registrar hasta dos o 3 atropellos en menos de 72h en el 2015 [27].



Ilustración 98: cruce de la C/ Dr Manuel candela-valencia

Logroño, cruce de la Avenida la Paz y el cruce de la calle Vara de Rey: ambas zonas ya han llegado a registrar un índice de 30 atropellos en menos de 5 meses y la mayoría de ellos nocturnos. Como se muestra en la ilustración son calles muy estrechas y con una visión nefasta por lo tanto el riesgo es mayor además de la escasa visibilidad que se presenta debido a la poca iluminación [28].



Ilustración 99: Avd. La Paz y calle Vara de Rey- Logroño

Santander, Cruce de la calle Isabel II: esta es una zona que a pesa tener mucha circulación y concurrencia peatonal apenas tiene semáforo y por lo tanto, no solo sería necesario instalar uno sino que se sea que disponga de esta tecnología de reducción de atropellos



Ilustración 100: Cruce de la c/ Isable II- Santander

4.6.2. Relés

El principal elemento que hará posible accionar todos los componentes con la placa controladora serán los *relés* o relevadores, que son unos dispositivos electromecánicos. Funcionan como unos interruptores controlados por un circuito eléctrico en el que por medio de una bobina y un electroimán se acciona un juego de uno o varios contactos que permiten abrir o cerrar otros circuitos eléctricos independientes.

Dado que el relé es capaz de controlar un circuito de salida de mayor potencia que el de entrada, puede considerarse en un amplio sentido como un amplificador eléctrico. La cantidad de dispositivos que se pretende controlar será directamente proporcional a la cantidad de relés que se deberían instalar, ya sean de manera independientes o en módulos de diversos canales.



Ilustración 101: Módulo relé de 4 canales

4.6.3. Cables jumper Arduino

Estos son unos cables con conectores macho-hembra y se utilizarían para conectar los pines del controlador Arduino a los distintos pines de los relés. La utilización de esos cables proporciona un modo muy sencillo de mantenimiento ya que si se funde uno, bastaría con cambiar ese en concreto con uno nuevo y sin la necesidad ir soldados para su sujeción.



Ilustración 102: Cables jumper con conectores macho-hembra

4.6.4. Adaptador de tensión y toma de corriente

Con el fin de no tener que utilizar una batería para alimentar el controlador a una tensión de 5-9v debido a que esta podría agotarse e interrumpiría el funcionamiento del sistema y como consecuencia poniendo en peligro la vida muchos seres vivos que estén ya acostumbrados al funcionamiento de esta nueva tecnología, se ha optado por incorporar una toma de corriente de 10A dentro del cuadro eléctrico y así alimentar al controlador mediante un adaptador de tensión de 9v DC. Eso permitiría que el sistema estuviera funcionando en todo momento y sin producirse interrupciones salvo por otras causas no previstas.



Ilustración 104: Adaptador de tensión a 9v



Ilustración 103: toma de corriente 10 A

4.6.5. Interruptores diferenciales

Un interruptor diferencial (ID), también llamado dispositivo diferencial residual (DDR), es un dispositivo electromecánico que se coloca en las instalaciones eléctricas de corriente alterna con el fin de proteger a las personas de los contactos directos e indirectos provocados por el contacto con partes activas de la instalación (contacto directo) o con elementos sometidos al potencial debido. También protegen contra los incendios que pudieran provocar dichas derivaciones

En este proyecto se ha optado por colocar una serie de interruptores que protejan a los técnicos de maniobra o a las personas no cualificadas en caso de producirse un incidente en la manipulación de la instalación. Y la distribución diseñada para ofrecer mayor seguridad es la que se ilustra en la siguiente tabla.

Tabla 9: Interruptores de protección

Bloque de protección	Intensidad (A) bipolar
Sistema General	63
Barreras luminosas	25
Barreras infrarrojas	25
Sensor PIR y sensor LDR	10
Estados del semáforos	10



Ilustración 105: Interruptor diferencial de 63A



Ilustración 106: Interruptor diferencial 25A

4.6.6. Caja de conexiones eléctricas

La única finalidad que tiene utilizar este cuadro es para concentrar en una única zona común todas las conexiones de los distintos componentes del sistema al controlador. Permitiendo que sea más fácil de manipular en caso de revisiones o para realizar el

mantenimiento en el sistema.



Ilustración 107: caja de conexiones

4.6.7. Control de las barreras luminosas

Para controlar las barreras luminosas incrustadas sobre el asfalto se utilizaría la tecnología *LightsGuard-system* que son unos dispositivos emisores de luz LED instalados de forma línea y que sobre salen 1,2 centímetros sobre el asfalto casi indetectables por los neumáticos al atravesarlos, pero visibles a casi 450 metros. Los *LightsGuard* son económicos y normalmente son utilizados para la señalización del cruce de peatones pero su disposición y tecnología se adapta perfectamente a las barreras luminosas de este proyecto. Son altamente resistentes al peso de un camión debido al duro envoltorio de acero que lo cubre y al requerir una tensión de alimentación de 9-30Vcc deben ser conectadas a unos relés para accionarlos con el Arduino.



Ilustración 108: aspecto modular de un LightGuard

Su modo de conexión al Arduino es exactamente igual a que se utilizaría para controlar cualquier foco luminoso. En la siguiente imagen se muestra el esquema de conexión.

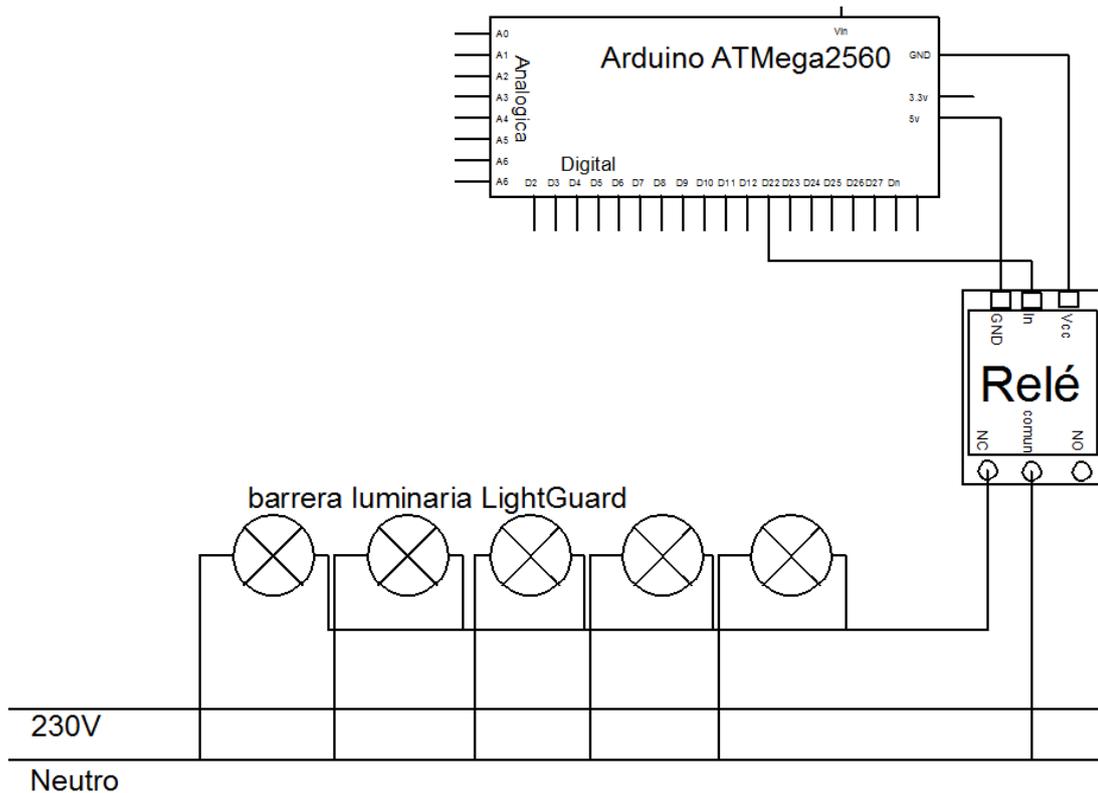


Ilustración 109: conexión de un lightguard al Arduino

4.6.6.1 Configuración de la distribución de iluminación

A la hora de diseñar todo el sistema en un entorno real se realizó un pequeño estudio sobre la posibilidad de que una mala distribución de las líneas de iluminación podría acarrear desventajas como crear una posible distracción a la hora de activarse en caso de que se encuentren muy cercar del vehículo. Dos de las posibles distribuciones adoptadas fueron las siguientes.

Solución 1: haciendo uso de la misma configuración adoptada en el entorno de pruebas solo bastaría con ampliar la separación entre las barreras infrarrojas y así el piloto tendrían espacio y tiempo necesario para una posible reacción. Teniendo en cuenta que la longitud media de los coches más rápidos o deportivos es de 4 metros se optó por utilizar una separación de **7 metros** entre barreras infrarrojas y **3,5 metros** de separación entre las barreras o líneas de iluminación *LightGuard*.

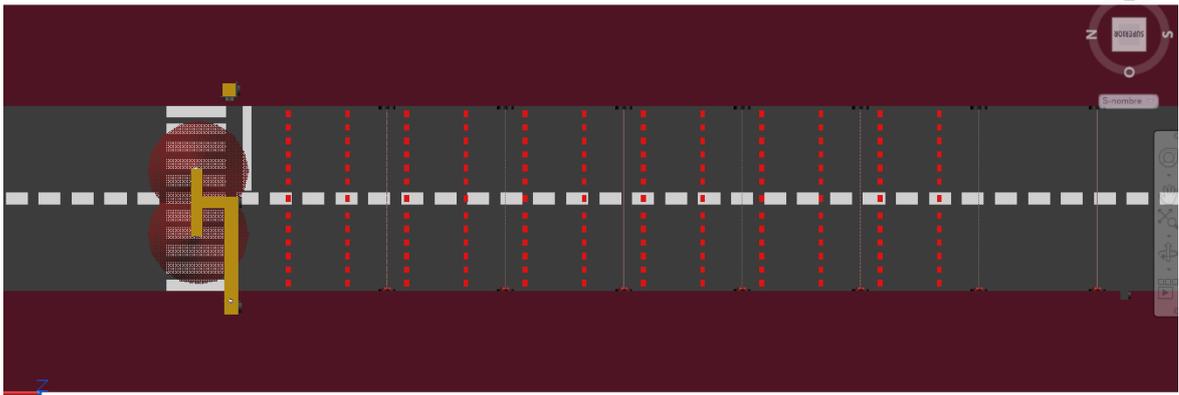


Ilustración 110: configuración con barreras de iluminación utilizadas en el prototipo 1

Solución 2: esta solución tenía un enfoque distinto y consiste en retrasar las barreras de iluminación un par de metros hacia el paso de peatones pero con la misma separación entre barreras infrarrojas adoptadas en la solución uno (7 metros y 3,5 metros entre barreras luminosas) y de este modo el piloto tendrá una visión más completa y amplia de todas las líneas de iluminación restantes delante de él y ampliando el espacio y tiempo de reacción.

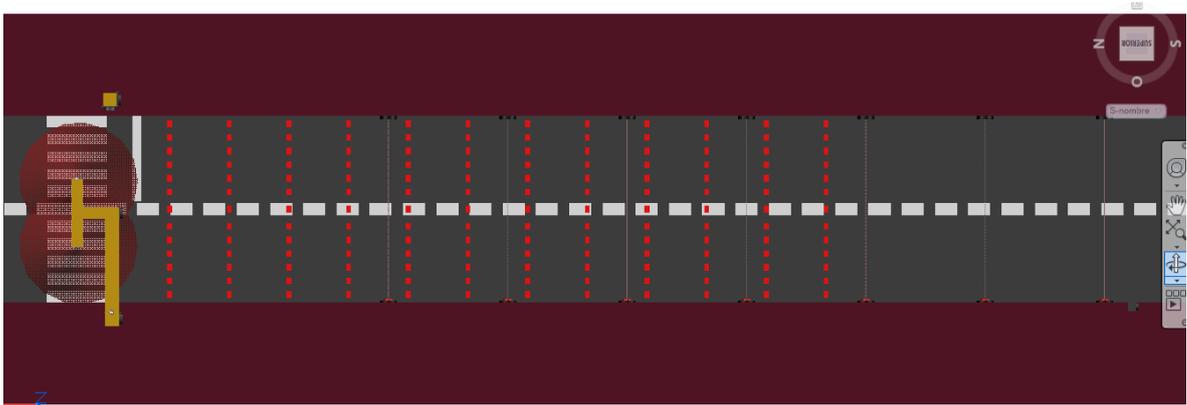


Ilustración 111: configuración con barreras de iluminación retrasadas

- **Nota:** En el apartado de programación del microcontrolador se incluirá tanto el código de programación para la *solución uno* y el código programado para la *solución dos*, aunque dentro del chip del controlador estará compilado el código uno.

Y la disposición después del montaje sería como se ilustra en la siguiente imagen.



Ilustración 112: distribución lineal de las *lightsguard*

4.6.8. Control del sensor de movimiento

Teniendo en cuenta la gran variedad de sensores de movimiento existentes para instalaciones domóticas, se ha optado por elegir un sensor que cumpla con las condiciones de operar en exteriores, es decir, que sea capaz de adaptarse a las condiciones medioambientales.

El sensor elegido es el **sensor de movimiento PIR BK-P02C**, funciona exactamente igual que el modulo PIR utilizado en el prototipo pero con la única diferencia de que este tiene que ser alimentado con una tensión de 12v en continua.

Especificaciones técnicas

- Fuente de alimentación: 110-220V AC.
- Carga de energía: 5 W ~ 100 W.
- Potencia normal: 1200W.
- Distancia del sensor: 5 ~ 10 m.
- Temperatura de trabajo: -10-+ 50°C.
- Protección del medio ambiente: IP44.
- Grados sensor: ángulo del cono de 140°.
- Tiempo de retardo: 2-600 segundos.



Ilustración 113: Sensor de movimiento de 12v

Para poder accionarlo con el Arduino también necesitará estar conectado a un relé para poder ser alimentado con una tensión de 12v. El esquema de conexión es el que se muestra en la siguiente ilustración.

Código de colores

- marón: 12Vcc
- Rojo: señal
- azul: neutro

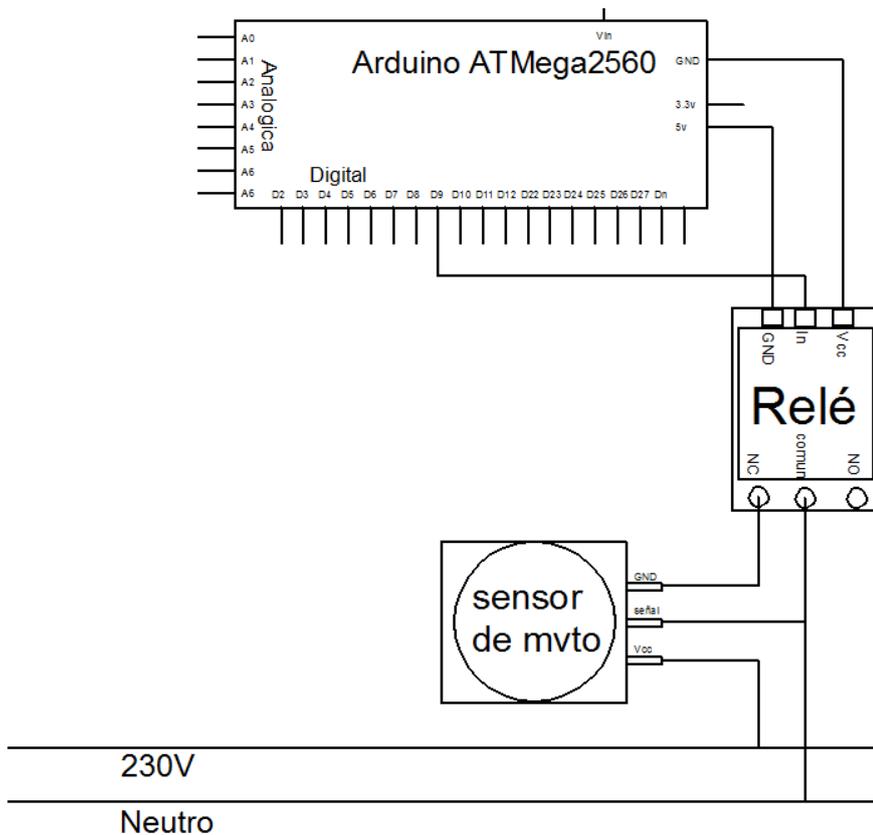


Ilustración 114: Esquema de conexión de un sensor de movimiento con relé

4.6.9. Control de sensor de luz

Para la adaptación del sensor de luz al sistema se ha optado por utilizar un **interruptor de control de luz fotoeléctrico**, ya que es utilizado habitualmente en demótica para la conmutación automática de las farolas sin necesidad de ser accionadas manualmente. Este módulo realiza la misma función que desempeña el sensor LDR utilizado en el entorno de prueba y existe una gran variedad de modelos pero todos actúan de la misma forma.

Especificaciones técnicas

- Tensión: 230v
- Potencia: 2200 W
- Intensidad: 10 A
- Consumo: 2W
- Sensibilidad de encendido: 30 Lux
- Sensibilidad de apagado: 150 Lux
- Ajuste Encendido Apagado: <5 lux-a 15lux



Ilustración 115: Sensor de luz

Visto que requiere una tensión de alimentación de 230v, para poder accionarlos con el Arduino también necesitará estar conectado a un relé como el resto de dispositivos. El esquema de conexión es el que se muestra en la siguiente ilustración

Código de colores

- **Negro:** neutro
- **Rojo:** 230Vcc
- **Blanco:** señal

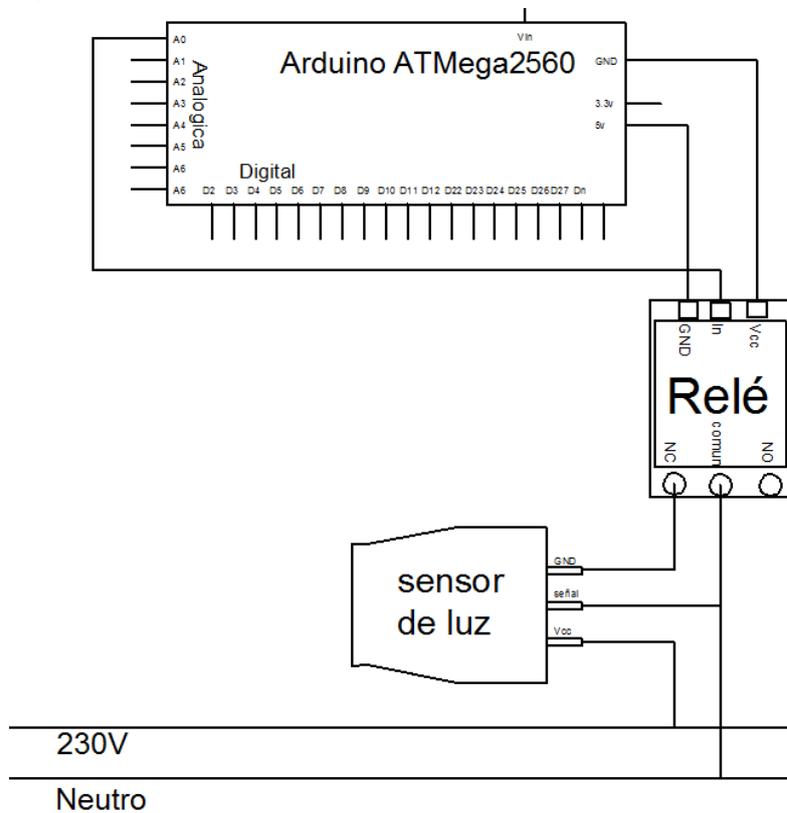


Ilustración 116: Esquema de conexión de un sensor de luz con relé

4.6.10. Control de las barreras infrarrojas

Para el control de las barreras infrarrojas se pueden encontrar diversos módulos debido a que es una tecnología muy utilizada en sistemas de seguridad. Con el fin de proporcionar un sistema seguro y con un bajo porcentaje de fallos en las lecturas, se ha elegido módulos **E3JM-10M4-N**, que son unos Sensor Fotoeléctricos para Sistema de Barrera infrarrojas tienen un alcance de 10 m y una precisión de hasta 100%. Ya por sí traen incorporados una Salida Relé para su conmutación.



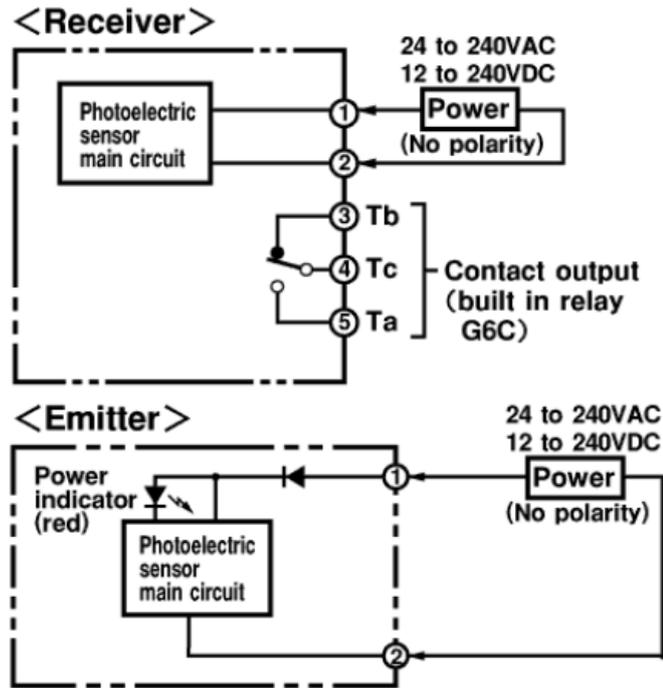
Ilustración 117: Módulos infrarrojos E3JM-10M4-N

Tabla 10: hoja de características del módulo infrarrojo

Specifications

Ratings/Characteristics						
Item	Through-beam		Retro-reflective with M.S.R.		Diffuse-reflective	
	E3JM-10□□4	E3JM-10□□4T	E3JM-R4□□4	E3JM-R4□□4T	E3JM-DS70□□4	E3JM-DS70□□4T
Sensing distance	10 m		4 m (When using E39-R1)		White paper (200 × 200 mm): 700 mm	
Standard sensing object	Opaque: 14.8-mm dia. min.		Opaque: 75-mm dia.min.		---	
Differential travel	---		---		20% max. of sensing distance	
Directional angle	Both Emitter and Receiver 3° to 20°		1° to 5°		---	
Light source (wavelength)	Infrared LED (950 nm)		Red LED (660 nm)		Infrared LED (950 nm)	
Power supply voltage	12 to 240 VDC±10%, ripple (p-p): 10% max. 24 to 240 VAC±10%, 50/60 Hz					
Power consumption	3 W max.		2 W max.			
Control output	Relay output (M Models): SPDT 250 VAC, 3 A max. (cosφ = 1) 5 VDC, 10 mA min. DC SSR output (S, R Models):48 VDC, 100 mA max. (residual voltage: 2 V max.) Light-ON/Dark-ON selectable					
Life expectancy	Mechanical	50,000,000 times min. (switching frequency: 18,000 times/h)				
	Electrical	100,000 times min. (switching frequency: 1,800 times/h)				
Response time	Relay output	Operation or reset: 30 ms max.				
	DC SSR output	Operation or reset: 5 ms max.				
Sensitivity adjustment	---				One-turn adjuster	
Timer function (See note.)	ON-delay/OFF-delay/One-shot delay switch selectable Delay time: 0.1 to 5 s (adjustable), only for E3JM-□□□4T					
Ambient illumination (Receiver side)	Incandescent lamp: 3,000 lx max.					
Ambient temperature	Operating: -25°C to 55°C (with no icing or condensation) Storage: -30°C to 70°C (with no icing or condensation)					
Ambient humidity	Operating: 45% to 85% (with no condensation) Storage: 35% to 95% (with no condensation)					
Insulation resistance	20 MΩ min. at 500 VDC between current-carrying parts and case					
Dielectric strength	2,000 VAC, 50/60 Hz for 1 min. between current-carrying parts and case					
Vibration resistance	Destruction	10 to 55 Hz, 1.5-mm double amplitude for 2 hours each in X, Y, and Z directions				
	Malfunction	10 to 55 Hz, 1.5-mm double amplitude for 2 hours each in X, Y, and Z directions				
Shock resistance	Destruction	500 m/s ² 3 times each in X, Y, and Z directions				
	Malfunction	100 m/s ² 3 times each in X, Y, and Z directions				
Degree of protection	IEC 60529: IP66					
Connection method	Terminal block					
Indicator	Light indicator (red), power indicator (red)	Operation indicator (red), power indicator (red)	Light indicator (red)	Operation indicator (red)	Light indicator (red)	Operation indicator (red)
Weight (packed state)	Approx. 270 g		Approx. 160 g		Approx. 160 g	
Material	Case	ABS				
	Lens	Methacrylic resin				
	Cover	Polycarbonate				
	Mounting Bracket	Iron				
Accessories	Mounting Bracket (with screw), nut, terminal protection cover, one set of cable connection nuts, reflector (E39-R1: only for retro-reflective models), instruction manual					

El circuito eléctrico de estos dispositivos es el que se muestra a continuación



Visto que requiere una tensión de alimentación de 230v, para poder accionarlos con el Arduino también necesitará estar conectado a dos relés, uno para el emisor y otro para el receptor. El esquema de conexión es el que se muestra en la siguiente ilustración

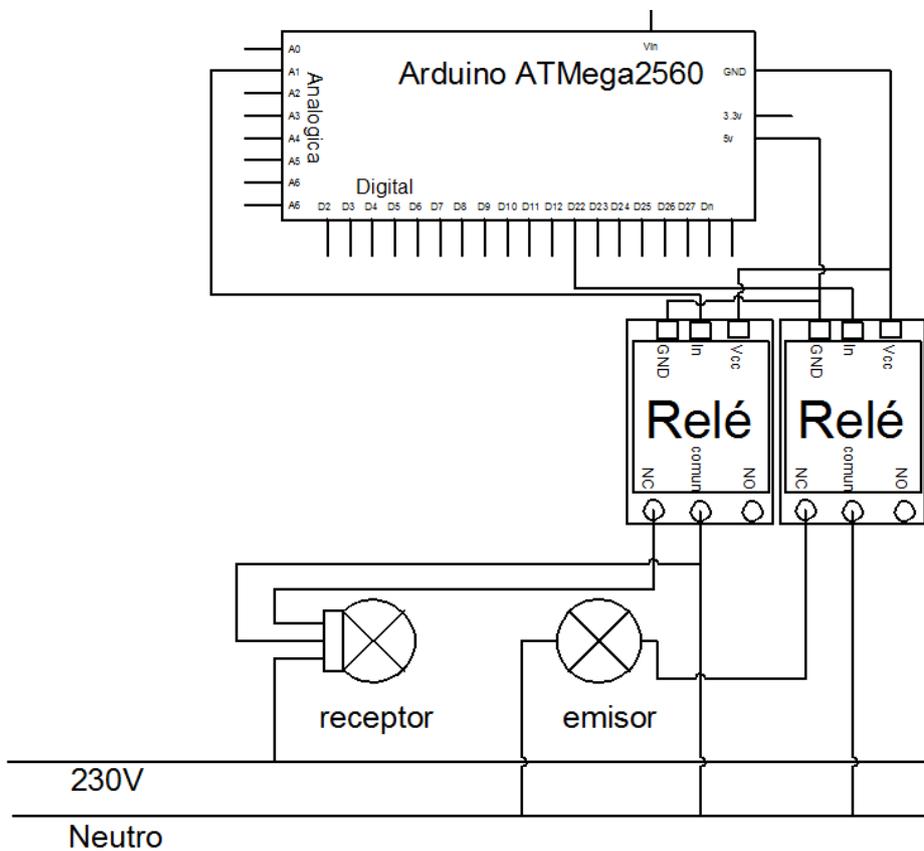


Ilustración 118: circuito eléctrico de emisor-receptor conectado a unos relés

CAPÍTULO 5: *RESULTADOS*

5. RESULTADOS

Como es de esperar en este quinto capítulo se recogerán todos los resultados de todas las pruebas realizadas en el entorno de pruebas, ya sean tanto los datos recogidos en la comprobación de los distintos componentes como los datos producidos en las diversas simulaciones realizadas en el sistema. En cada una de las hipótesis simuladas se realizará un comentario explicándolo a fondo con el fin de proporcionar una mayor comprensión del funcionamiento del sistema.

5.1 Comprobación de la señales los sensores

La primera prueba realizada fue comprobar el correcto funcionamiento de los distintos sensores integrados en el sistema, en especial los sensores que constituyen las barreras infrarrojas obteniendo así los siguientes datos o señales analógicas.

Tabla 11: señales recibidas por los receptores infrarrojos en Bits

Barrera	Señal en Activación (bits)	Señal en desactivación (bits)
1	189-198	0-3
2	968-970	0-6
3	80 - 85	0-4
4	963-967	0-20
5	113-151	0-9
6	154-158	0-16
7	95-102	0-10

Observando la cantidad de señal recibida por los distintos receptores infrarrojos se puede ver que hay una clara variación en cada rango de señales y eso es producido principalmente por dos razones:

- **La correcta o no correcta alineación entre el emisor y el receptor** de cada barrera ya que al ser un prototipo hecho a mano y al no disponer de una herramienta de nivelación pues se producen ligeras desviación en la orientación de la señal.
- Al ser un circuito eléctrico real con resistencias y aplicando los conceptos de teoría de circuitos (**asignatura impartida en los grados de la EIMIA**), todas las intensidad que circulan en las ramas no pueden ser iguales y por lo tanto esa **ligera diferencia de intensidades** que llegan a los dispositivos puede influir en la cantidad de señal que se puede emitir.

Haciendo una valoración general sobre estas cantidades de señal recibida se podría dar como valores buenísimos ya que al ser señales analógicas y mayores que cero proporcionan un correcto funcionamiento y por lo tanto no se producirían lecturas falsas.

Para la suposición de un entorno real ese pequeño problema no se produciría ya que los dispositivos utilizados se podrían alinear perfectamente, calibrar y establecer una misma cantidad de señal recibida en todas las barreras.

Otra de las comprobaciones realizadas fue el sensor de movimiento para la detección de personas y el sensor de luz o fotorresistencia con el fin de saber la cantidad de luz a que debería empezar a funcionar el sistema de manera automática sin necesidad de ser conmutado por un agente de tráfico. A continuación se muestra una tabla de datos preestablecidos teniendo en cuenta solo tres situaciones de riesgo y una situación fuera de peligro.

Tabla 12: Rango de señales del LDR y señales del PIR

Estado LDR		Estado PIR	Señal	T. comprobación
Mucha luz	0-749	Presencia	1 (5v)	0,4 seg
Nublado/ niebla	750-860	No presencia	0 (0v)	0,4 seg
Tarde-oscura	861-900			
Noche	901-1024			
Señales analógicas		Señales digitales		

Observando la *tabla 11* se puede ver todos los valores establecidos, los cuales fueron tomados tras realizar un estudio del tiempo real en diversos días. Se diseñó un pequeño circuito con el sensor LDR y se realizaron la toma de datos conforme iba cambiando el estado del día desde la terraza de mi piso y anotando los valores proporcionados por el sensor.

5.2 Hipótesis y resultados de las simulaciones realizadas

Como ya se anticipó al principio, en este apartado se realizará una serie de hipótesis para distintos casos y se mostrarán todos los resultados obtenidos en las distintas simulaciones con el fin de determinar los distintos comportamientos que adopta el sistema.

✚ **Nota:** las simulaciones se realizarán teniendo solo en cuenta los tres estados de riegos preestablecidos; *nublado, tarde oscura, noche*.

5.2.1 Hipótesis 1

El coche viaja a una velocidad normal y/o con desaceleración progresiva. En este caso independientemente del estado del semáforo no se crea peligro alguno y los datos recogidos en la simulación son los siguientes:

Tabla 13: Resultados de la simulación uno

Barreras	Tramo	Vnormal (km/h)	Vc2 (km/h)	Vc1 (km/h)	Evolución	Error	PIR	Salida activadas
1-2	A	50	50	52,3	acelerando	0.046	NO	No iluminación
2-3	B	50	52,3	49,7	desacelerando	-0.006	NO	No iluminación

3-4	C	50	49,7	45,5	desacelerando	-0.09	SI	No iluminación
4-5	D	50	45,5	31,5	desacelerando	-0.37	NO	No iluminación
5-6	E	50	31,5	10,8	desacelerando	-0.78	NO	No iluminación
6-7	F	50	10,8	6,8	parada	-1.08	NO	No iluminación

✚ **Nota:** cuando el Error es negativo es considerado prácticamente **cero** ya que la velocidad es menor que la velocidad permitida.

Observaciones: mirando la tabla de valores recogidos se puede ver que la velocidad del coche partía de una ligera aceleración y poco a poco ha ido reduciendo la velocidad. Otra cosa que se puede observar en la tabla es la información proporcionada por el sensor de PIR y muestra que prácticamente no hubo gente cruzando y por lo tanto no se producía peligro alguno aunque el piloto haya reducido la velocidad en todo el recorrido por ser prudente obteniendo como respuesta en la salida **No iluminación**.



Ilustración 119: simulación uno

5.2.2. Hipótesis 2

Con el semáforo en rojo se simulará la situación en la que el automóvil entra en la zona de control a mucha velocidad y se podrá ver la rápida respuesta que producirá el sistema ante tal situación. Los datos recogidos en esta simulación son los que a continuación se puede ver en la siguiente tabla.

Tabla 14: Resultados de la simulación uno, hipótesis 2

Barreras	Tramo	Vnormal (km/h)	Vc2 (km/h)	Vc1 (km/h)	Evolución	Error	PIR	Salida activadas
1-2	A	50	50	95.6	Acelerando	0.912	SI	Iluminar 12
2-3	B	50	95.6	100	Acelerando	1	SI	Iluminar 8
3-4	C	50	100	120	Acelerando	1.4	NO	Iluminar 7
4-5	D	50	120	65	Desacelerando	0.3	SI	Iluminar 6
5-6	E	50	65	50	Desacelerando	0	SI	Iluminar 3
6-7	F	50	50	28.4	Prec/parada	-0.43	NO	Iluminar 1



Ilustración 120: simulación uno de la hipótesis dos

Observaciones: haciendo un análisis a los resultados recogidos en la simulación se puede ver que el automóvil iba a mucha velocidad en los tres primeros tramos y a pesar de que se registraban presencia humana seguía acelerando hasta que se dio cuenta del peligro y empezó a frenar poco a poco. En la *ilustración 120* se puede ver las salidas que tomó el controlador ante tales situaciones.

- Otra situación que se incluiría en esta hipótesis es aquella en la que **el automóvil entra en la zona de control a una velocidad normal** pero sin darse cuenta de la su proximidad a la intersección acelera aumentando considerablemente su velocidad y por lo tanto, la respuesta producida en la salida también tendría que ser rápida. Los datos recogidos en esta simulación son las que a continuación se muestran en la tabla.

Tabla 15: Resultados de la simulación dos, hipótesis 2

Barreras	Tramo	Vnormal (km/h)	Vc2 (km/h)	Vc1 (km/h)	Evolución	Error	PIR	Salida activadas
1-2	A	50	50	45	Desacelerando	0.1	NO	NO iluminación
2-3	B	50	45	56	Acelerando	0.12	NO	Iluminar 10
3-4	C	50	56	79.5	Acelerando	0.59	SI	Iluminar 9
4-5	D	50	79.5	90	Acelerando	0.8	NO	Iluminar 6
5-6	E	50	90	71	Desacelerando	0.42	SI	Iluminar 3
6-7	F	50	71	66	Des/ riesgo	0.32	SI	Iluminar 2

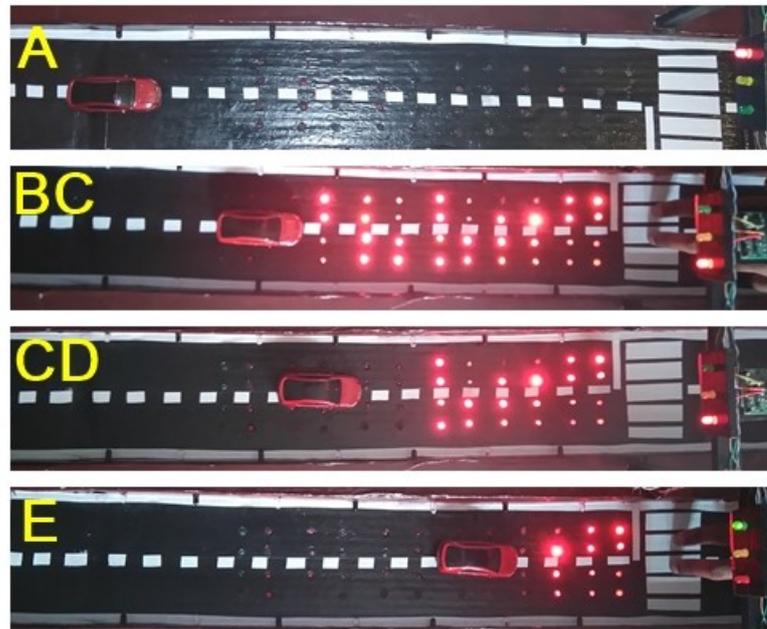


Ilustración 121: simulación dos de la hipótesis dos

Observaciones: fijándose en la *ilustración 121* se puede ver que cuando el vehículo entro en la zona de control iba a una velocidad normal pero en un determinado momento empezó a acelerar e independientemente de que se detecten peatones cruzando, el comportamiento del conductor no es normal teniendo en cuenta que semáforo ha pasado al estado rojo y debería desacelerar para llegar seguro a la parada. Por lo tanto el sistema ha decidido avisarle ante este comportamiento brusco.

5.2.3. Hipótesis 3

Con semáforo en verde o ámbar y teniendo en cuenta que los automóviles tienen toda la prioridad para transitar independientemente de la velocidad a la que va, se ha simulado la situación en la que se ha detectado **peatones vía andantes cruzado la intersección de forma ilegal** arriesgando sus vidas. Los datos en la simulación son los siguientes.

Tabla 16: Resultados de la simulación uno, hipótesis 3

Barreras	Tramo	Vnormal (km/h)	Vc2 (km/h)	Vc1 (km/h)	Evolución	Error	PIR	Salida activadas
1-2	A	50	50	70.4	acelerando	0.408	-	No iluminación
2-3	B	50	70.4	74.5	acelerando	0.49	-	No iluminación
3-4	C	50	74.5	80	acelerando	0.6	NO	Iluminar 6
4-5	D	50	80	60	desacelerando	0.1	SI	Iluminar 6
5-6	E	50	60	50	desacelerando	0	SI	Iluminar 3
6-7	F	50	50	25.9	Des /Prec	-0.482	SI	Iluminar 2

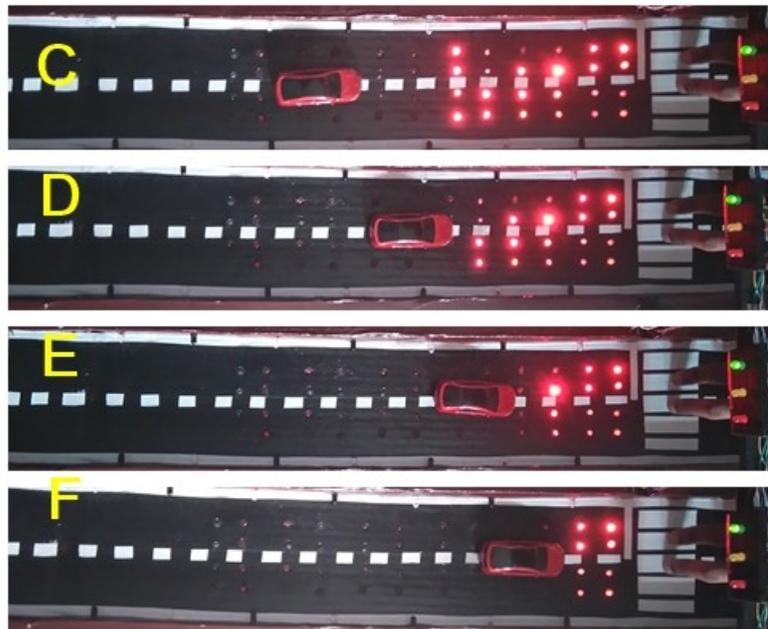


Ilustración 122: simulación de semáforo en verde o ámbar y con peatones cruzando

Observaciones: haciendo una observación en la tabla de datos recogidos se puede ver que las lecturas del sensor de movimiento producidos en los cuatro últimos tramos (C,D,E,F) mostraban que se detectaba personas cruzando de modo arriesgado pero independientemente de que el piloto tenga prioridad absoluta es necesario que el sistema actúe ante tal comportamiento y avisa al piloto de que es necesario e importante acercarse con mucha precaución con el fin de evitar incidentes (*principal problema que se pretende solucionar en este proyecto*) y la respuesta adoptada por el controlador es la de activar en la salida una serie de barreras que se van iluminando a medida que el vehículo se acerca.

5.2.4. Hipótesis 4

La situación presentada en esta hipótesis es aquella en la una vez que el sistema este funcionando coincide con que el automóvil o automóviles ya han atravesado primeros tramos (*tramo uno, dos o tres*) situados a una mayor distancia con respecto al punto de parada del semáforo y por lo tanto ya no es necesario realizar lecturas en las barreras infrarrojas en estos tramos y se deberá tener solo en cuenta los tramos siguientes para

analizar el comportamientos en esos tramos restantes. Los datos recogidos en la simulación realizada son las que se muestran a continuación.

Tabla 17: Resultados de la simulación uno, hipótesis 4

Barreras	Tramo	Vnormal (km/h)	Vc2 (km/h)	Vc1 (km/h)	Evolución	Error	PIR	Salida activadas
1-2	A	50	-	-	-	-	-	No iluminación
2-3	B	50	-	-	-	-	-	No iluminación
3-4	C	50	-	68.7	Acelerando	0.374	NO	No iluminación
4-5	D	50	68.7	51.8	desacelerando	0.036	SI	Iluminar 7
5-6	E	50	51.8	36.8	desacelerando	-0.264	SI	Iluminar 4
6-7	F	50	36.8	15.7	Des/ parada	-0.686	NO	Iluminar 2



Ilustración 123: simulación uno de la hipótesis cuatro

Observaciones: acorde a los resultados recogidos en esta simulación se observa que en los dos primeros tramos no se tiene información de ellos debido a que el sistema entro funcionamiento cuando el vehículo ya había recorrido una parte del escenario de control y por lo tanto solo se ha tenido en cuenta los tramos restantes.

- En esta hipótesis también se podrían presentar la situación en la que el automóvil acelera continuamente y por lo tanto la respuesta en la salida sería diferente a la de la primera simulación.

Tabla 18: Resultados de la simulación dos, hipótesis 1

Barreras	Tramo	Vnormal (km/h)	Vc2 (km/h)	Vc1 (km/h)	Evolución	Error	PIR	Salida activadas
1-2	A	50	-	-	-	-	-	No iluminación
2-3	B	50	-	58	Acelerando	0.16	SI	Iluminar 11
3-4	C	50	58	67	Acelerando	0.34	NO	Iluminar 9
4-5	D	50	67	72.5	Acelerando	0.45	NO	Iluminar 5
5-6	E	50	72.5	80	Acelerando	0.6	SI	Iluminar 3
6-7	F	50	80	82	Prec/riesgo	0.64	SI	Iluminar 2

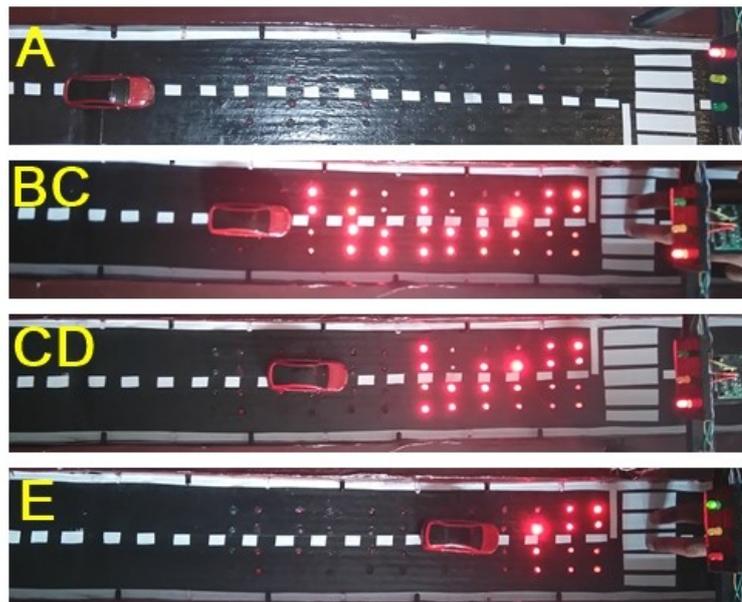


Ilustración 124: simulación dos de la hipótesis cuatro

Observaciones: recapitulando la situación descrita en esta hipótesis y mirado salidas adoptadas por el controlador en *ilustración 124* y la evolución del vehículo según los datos mostrados en la tabla se puede ver que se produjo un cambio brusco en la velocidad y por lo tanto se tuvieron que activar algunas barreras de iluminación ya que también se detectaban patones cruzando.

5.2.5. Hipótesis 5

Teniendo en cuenta que los sistemas eléctricos/ electrónicos están propensos a fallos técnicos, pues en esta hipótesis se simulará la situación en la que **una o varias barreras infrarrojas no funcionan** y por lo tanto, solo se tendrá en cuenta las que si funcionan para analizar el comportamiento del sistema.

- ✚ **Nota:** de acuerdo a la programación realizada se estableció que se tendría en cuenta esos fallos técnicos en las seis primeras barreras ya que las última restante está ya muy próxima al paso de peatones y el sistema no tendría información para una posible actuación en caso de que la séptima también fallase.

- En el primer caso de fallos simulado en esta hipótesis se recogieron los siguientes datos:

Tabla 19: Resultados de la simulación uno, hipótesis 5

Barreras	Tramo	Fallos (B _n)	Vc2 (km/h)	Vc1 (km/h)	Evolución	Error	PIR	Salida activadas
1-3	A	B2	50	60	Acelerando	0.2	NO	Iluminación 8
3-4	B	-	60	75	Acelerando	0.5	NO	Iluminar 7
4-6	C	B5	75	52	Desacelerando	0.04	SI	Iluminar 3
6-7	D	-	52	69	Ac/riesgo	0.38	SI	Iluminar 1

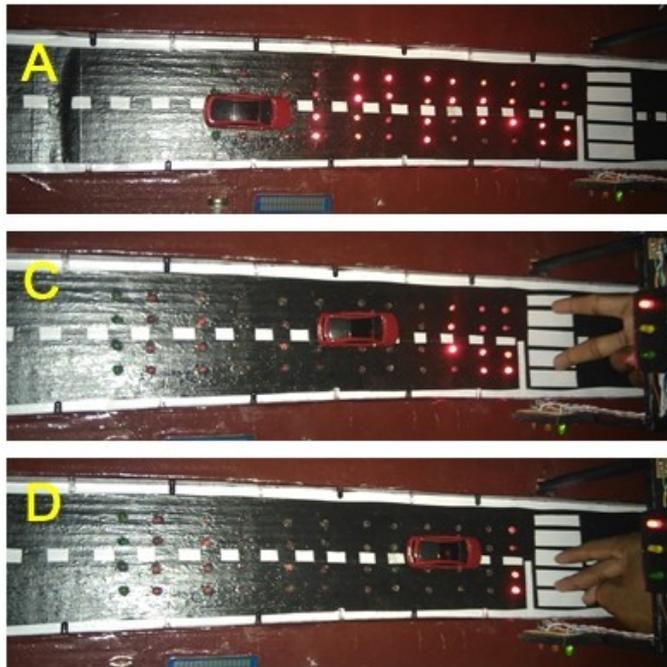


Ilustración 125: simulación uno de la hipótesis cinco

- En el segundo caso de fallos simulado en esta hipótesis se recogieron los siguientes datos:

Tabla 20: Resultados de la simulación dos, hipótesis 5

Barreras	Tramo	Fallos (B _n)	Vc2 (km/h)	Vc1 (km/h)	Evolución	Error	PIR	Salida activadas
1-4	A	B2 B3	50	84	Acelerando	0.68	NO	Iluminar 4
4-5	B	-	84	62	Desacelerando	0.24	NO	Iluminar 3
5-6	C	-	62	40	Desacelerando	-0.2	SI	Iluminar 2
6-7	D	-	40	26	Des/parada	-0.48	NO	Iluminar 1

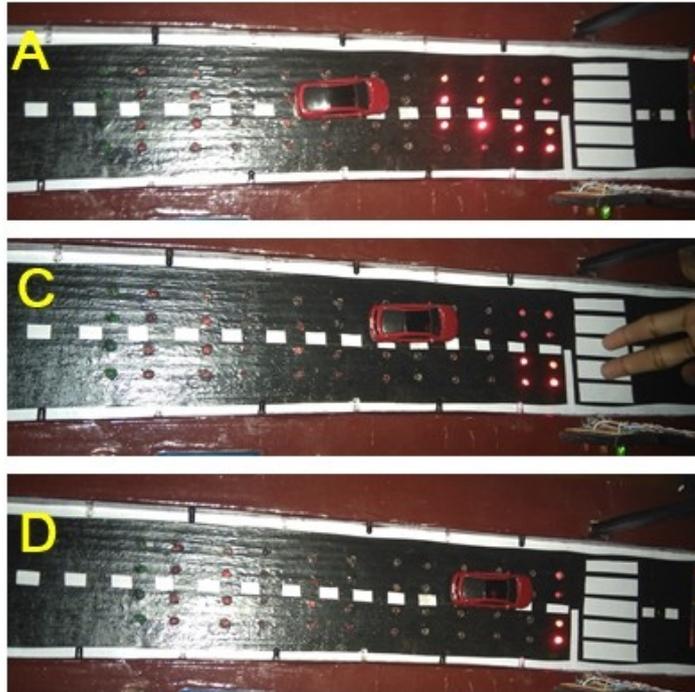


Ilustración 126: simulación dos de la hipótesis cinco

- En el tercer caso de fallos simulado en esta hipótesis se recogieron los siguientes datos:

Tabla 21: Resultados de la simulación tres, hipótesis 5

Barreras	Tramo	Fallos (B _n)	Vc2 (km/h)	Vc1 (km/h)	Evolución	Error	PIR	Salida activadas
2-3	A	B1	50	35	Desacelerando	-0.3	SI	NO iluminación
3-4	B	-	35	68	Acelerando	0.36	NO	Iluminar 6
4-6	C	B3	68	53	Desacelerando	0.06	NO	Iluminar 3
6-7	D	-	53	21	Des/parada	-0.58	NO	Iluminar 2

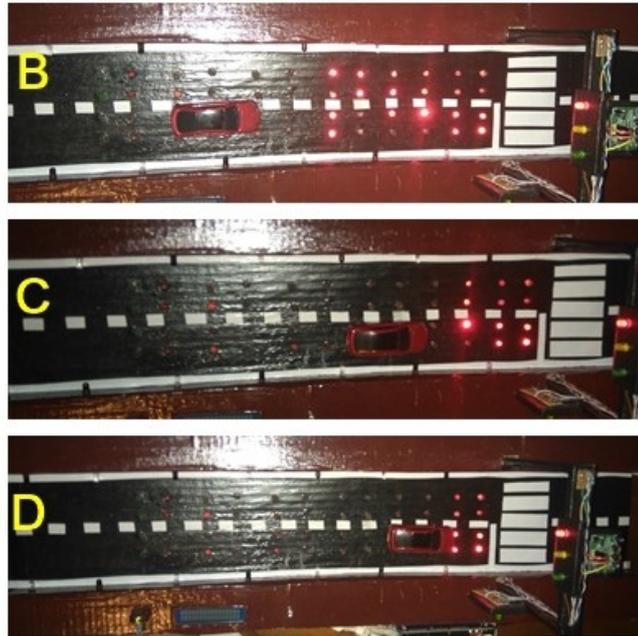


Ilustración 127: simulación tres de la hipótesis cinco

- En el cuarto caso de fallos simulado en esta hipótesis se recogieron los siguientes datos:

Tabla 22: Resultados de la simulación cuatro, hipótesis 5

Barreras	Tramo	Fallos (B _n)	Vc2 (km/h)	Vc1 (km/h)	Evolución	Error	PIR	Salida activadas
1-2	A	-	50	50	Constante	0	NO	Iluminar 10
2-5	B	B3 B4	50	55	Acelerando	0.1	NO	Iluminar 4
5-6	C	-	55	75	Acelerando	0.5	NO	Iluminar 3
6-7	D	-	75	60	Des/^ riesgo	0.2	SI	Iluminar 2



Ilustración 128: simulación cuarto de la hipótesis cinco

- **Nota:** La codificación *B1, B2, B3, B4, B5* es referida a las barreras que no están funcionando cuando son atravesados por el automóvil.

Observaciones: como se explicó en la hipótesis el funcionamiento del sistema es idénticamente igual que en las primeras hipótesis planteadas con la única diferencia de que se ha tenido en cuenta que algunas barreras pueden fallar y la distancia entre tramos ya no será la misma lo que da a entender que algunos tramos abarcarían tres o más barreras hasta que se haga el próximo corte.

Atendiendo a los resultados obtenidos en las diversas hipótesis y simulación realizadas se puede confirmar que el sistema responde de manera rápida, eficaz y satisfactoria conforme a los diversos comportamientos que pueden ir adquiriendo los automóviles en todo el recorrido que compone el escenario de control, desde situaciones más complejas a situaciones más sencillas.

CAPÍTULO 6: *ESTUDIO* *ECONÓMICO*

6. ESTUDIO ECONÓMICO

Llegado a este punto y teniendo en cuenta que este es un proyecto que abarca dos partes, el estudio económico realizado estará dividido también en dos partes. En la primera parte se mostrará el coste total que ha supuesto la elaboración del *prototipo o el entorno de pruebas*, incluyendo ciertos aspectos fundamentales tales como, el tiempo empleado y todos los materiales utilizados con el fin de diseñarlo. En la segunda parte del estudio se reflejará cual podría ser el presupuesto para una implementación en *un entorno real*. Para ello se buscará precios de los materiales en el mercado que podrían ser compatibles y utilizados en dicho sistema.

6.1. Presupuesto del entorno de pruebas

Con el objetivo de realizar un proyecto lo más económico posible, se realizó un previo estudio de todos los componentes necesarios y compatibles que se utilizarían para la elaboración del prototipo asegurando que esos tengan todas las características deseadas. Una vez obtenido un listado con todos los componentes, se proseguía a su búsqueda y consulta de precios en las diferentes tiendas de informática *online* debido a que en las tiendas físicas esos componentes resultaban ser más caras y todo el presupuesto corría a cuenta del proyectista.

6.1.1. Componentes electrónicos

Todos los componentes presentes en el escenario son los que a continuación se mencionan y con sus respectivos precios y puntos de adquisición. Aunque es necesario mencionar que la mayoría los componentes fueron adquiridos en la página de compra venta eBay: <http://www.ebay.es/>

➤ Placas Arduino

Kit Arduino uno Básico 30€ + gastos de envío 0.79€ (5-10 días laborales) placa fundida tras varias pruebas y las conexiones.

Sensores LDR incluidos en el Kit Básico

Potenciómetro

<http://www.ebay.es/itm/Basic-Starter-Learning-Kit-UNO-For-Arduino-Basics-/131572829679?hash=item1ea25a0def:g:HpEAAOSwrkIVUnav>

Placa Arduino ATmega2560 13,40€ con envío gratis (5-10 días laborales) ubicación España

<http://www.ebay.es/itm/MEGA-2560-R3-TIPO-ARDUINO-100-COMPATIBLE-ATMEGA-16U2-REV3-V3-UNO->

[/141478130050?hash=item20f0c0f182:g:iqwAAOSwAYtWHx1A](#)

➤ **Diodos LED**

Paquete de 50 pieza de color rojo + 25 Resistencias de 220 ohm 3,45€ envío gratis (5-10 días laborales) ubicación España

<http://www.ebay.es/itm/x50-Diodos-Led-3mm-Rojo-Red-difuso-Alta-Calidad-Resistencias-Arduino-/151850178572?hash=item235af9d80c:g:30QAAOSwbqpT0DcF>

Paquete de 15 piezas (verde, rojo, amarillo) 5mm super brillante 1€ + gastos de envío 0.19€ (5-7 días laborales) ubicación España

<http://www.ebay.es/itm/15x-kits-Diodo-led-verde-rojo-amarillo-5mm-Super-Bright-Arduino-electronica-/301798278144?hash=item4644944400:g:KKkAAOSw-W5UvqFz>

➤ **Módulos Emisor-Receptor de infrarrojos**

Kit de 10 módulos emisor y 10 módulos receptor 2x1,60€ + gastos de envío 0,60€ (3-5 días laborales) ubicación España

<http://www.ebay.es/itm/Kit-5-Diodo-IR-emisor-y-5-Sensor-IR-receptor-3mm-infrarrojo-infrared-diode-photo-/321838018564?hash=item4aef0a7004:g:U8cAAOSwu4BV1Rzw>

➤ **Sensor de movimiento PIR**

Módulo detector de movimiento o sensor de infrarrojo pasivo 2,99€ gastos de envío gratis (7-15 días laborales) ubicación España.

<http://www.ebay.es/itm/Modulo-Sensor-detector-movimiento-PIR-HC-SR501-Arduino-Nano-Uno-Mega-Raspberry-/252084295428?hash=item3ab1651704:g:FbwAAOSwFnFV7fQ9>

➤ **Módulos láser**

3 módulos transmisor láser KY-008 4,80€ + gastos de envío 1,50€ (10-20 días laborales) ubicación España.

Estos módulos se utilizaron al principio pero debido a ciertas desventajas respecto a su posible mal utilización, se cambio parte del enfoque del proyecto.

<http://www.ebay.es/itm/Modulo-transmisor-Laser-KY-008-650nm-5V-para-arduino-/121493700517?hash=item1c4996bfa5:g:saUAAOSw2XFUbx6s>

➤ **Pantalla display LCD**

Pantalla display LCD de 16x2 caracteres 3,99€ + gastos de envío 1€ (5-7 días laborales)

ubicación España

<http://www.ebay.es/itm/PANTALLA-LCD-16x2-CARACTERES-ARDUINO-RASPBERRY-IC-ALFANUMERICO-DISPLAY-LUZ-AZUL-/121808421588?hash=item1c5c5902d4:g:ARgAAOSwT6pVwzew>

➤ **Placas PCB**

Kit de 10 Placas PCB para soldar circuitos 3,79€ con envío gratis (5-7 días laborales) ubicación España.

<http://www.ebay.es/itm/10-X-Nuevo-Placa-Prototipo-PCB-para-kit-Circuito-PROTOBOARD-ARDUINO-Breadboard-/271850344220?hash=item3f4b8b0f1c:g:4WYAAOSwymxVPAH1>

➤ **Resistencias**

Lote de 50 resistencias de 1k 1€+ gastos de envío 2,01€ (10-15 días laborales) ubicación Francia

http://www.ebay.es/sch/i.html?_from=R40&_trksid=m570.11313&_nkw=resistencias+de+1K&_sacat=0

6.1.2. Herramientas, y material decorativo para entorno

Mini alicate de corte 2,50€ tienda Bazar Chinos, Ubicación España: Almadén Plaza de Toros

Dos botes de pintura plástica 3,98€ tienda Bazar Chinos, Ubicación España: Almadén Plaza de Toros

Brocha de pintura 0,80 € tienda Bazar Chinos, Ubicación España: Almadén Plaza de Toros

Pegamento de cartón o madera 0,75€ tienda Bazar Chinos, Ubicación España: Almadén Plaza de Toros

Pack destornilladores mino 6€ tienda Bazar Chinos, Ubicación España: Almadén Plaza de Toros

2 tiras de cinta adhesiva 1,20€ tienda Bazar Chinos, Ubicación España: Almadén Plaza de Toros

Tabla 23: presupuesto del material empleado en el prototipo

Materiales	Precio
Kit Arduino Uno básico	30,79€
Arduino ATMMega2560	13,40€
Diodos LED	4,64€
Diodos Emisor-receptor Infrarrojos	3,80€
Placa PCB	3,79€
Sensor de Movimiento	2,99€
Módulos transmisor Laser	6,30€
Pantalla LCD display	3,99€
Resistencias	3,01€
Mini alicata de corte	2,50€
Pintura	3,98€
Brocha de pintura	0,80€
Pack destornilladores	6€
Cinta adhesiva	1,20€
Pegamento de cartón o madera	0,75€
Total (IVA incluido)	87,94€

6.1.3. Horas de trabajo y planificación del proyecto

- El proyecto arrancó a mediados de **octubre** y a lo largo de esos primeros días se estuvo buscando información sobre los problemas registrados en la seguridad vial (*incremento del índice de atropellos en los pasos de peatones*) y planteando una serie de Hipótesis sobre cómo podrían ser resueltos tales problemas y estudiar si era necesario o no intentar reforzar esos puntos negativos encontrados. El tiempo empleado en realizar estos estudios eran 4 días a la semana y 2 horas diariamente haciendo un total de **16 horas** en las 2 semanas.
- En el mes de **noviembre** y **diciembre** se planteó un primer enfoque sobre el proyecto y se investigó a fondo la historia sobre los semáforos, su evolución y los avances tecnológicos que se han ido implementado, así mismo como los problemas que todavía no han sido resueltos. A lo largo de este tiempo también se empezó a pensar sobre cómo sería el modelado del prototipo y los posibles componentes que pueden ser utilizados sin olvidar estudiar sus características. Toda la información encontrada se fue documentando con fin de crear o tener una idea general sobre lo que se quería conseguir con la realización de este proyecto. Al ser un periodo de mucho estudio, el tiempo invertido en esos dos meses fue de 4 días a la semana y 3 horas diarias, haciendo un total de **96 horas** aproximadamente.
- En el mes de **enero** y **febrero** teniendo ya claro lo que se iba a realizar se diseñó el primer modelado en 3D para obtener una visión intuitiva de cómo podría ser el modelo final adoptado y se comenzó a comprar los componentes electrónicos para la construcción de entorno de pruebas, a la vez que se seguía investigando todo sobre semáforos y estudiando cómo se comportaría el sistema y las principales funciones que iba a desempeñar. Ya para finales de febrero se empezó a investigar sobre la existencia de proyectos similares o proyectos cuya finalidad era la de reforzar o mejorar la seguridad vial.

Cosa que no tuvo mucho éxito debido a que la idea de este proyecto era novedoso y poca información se podía encontrar. Durante este periodo de tiempo se invirtieron unos 5 días a la semana y 4 horas diarias, haciendo un total de **200 horas** aproximadamente.

- En el mes de **marzo** se empezó a estudiar a fondo la programación de microcontroladores (Arduino) y realizando programas para ir comprobando el correcto funcionamiento de los diferentes compones comprados. Para no tener que construir un entorno de pruebas definitivo se optó por realizar un mini entorno para simular el funcionamiento del sistema y ver si se realizaban las funciones expuestas, examinar su comportamiento para luego adaptarlo al entorno definitivo. Como ya se menciona anteriormente, toda información encontrada y todos los avances o modificaciones eran documentados. Al ser un mes de trabajo teórico-práctico se invirtieron unos 5 días a la semana y 4,5 horas diarias, haciendo un total de **90 horas**.
- En el mes de **abril** y **mayo** tras realizar varias pruebas y visto el comportamientos del sistema, se decidió realizar varias modificaciones con el fin de hacerlo más complejo con el fin de proporcionar unas respuestas más precisas. Luego en todo ese tiempo se dieron cuenta de que una de las implementaciones en el sistema traería más problemas o sería un sistema vulnerable y con posibles críticas y que al final se decidió cambiar parte del enfoque del problema que se quería resolver y adoptar otra tecnología similar (*barreras luminosas enterradas en el asfalto*) y que también ofrecería los mismos resultados e incluso mejores que los que podía haber ofrecido la primera idea planteada (*barreras de iluminación con laser en forma de pantalla instalada en el propio semáforo que se desplegaba cuando se detectaba comportamientos anómalos*). El tiempo empleado en todo ese análisis fue de 4 días a la semana y 4 horas diarias haciendo un total de **128 horas**.
- Ante los problemas surgidos en el sistema, en el mes de **junio** y **julio** se empezó a estudiar sobre la posibilidad de implementar la tecnología de inteligencia artificial conocida como lógica difusa o en utilizar algún tipo de regulador para conseguir un mejor control en el sistema ya que también sería más fácil adaptar este nuevo sistema a un entorno real, sin necesidad de realizar muchas modificaciones. En este mes también se realizaron pedidos de los nuevos componentes que iban ser introducidos en el nuevo sistema. Siendo este un periodo de investigación se invirtieron unos 4 días a la semana y 3,5 horas diarias, sumando así un total de **112 horas** aproximadamente.
- Una parte del mes de **septiembre** y todo **octubre** se decidió por utilizar un regulador PID y se dedicó todo ese tiempo en realizar y finalizar el entorno de pruebas definitivo, ya que se tenía todos los componentes y ya se conocía de sobre todas las funciones que se iban a realizar y las disposiciones de cada uno de esos componentes. Durante 5 horas se avanzaba con la maqueta y luego se tomaba otras 2 horas para ir documentando todos los avances. También se seguía estudiando sobre los reguladores y las diferentes utilidades que pueden desempeñar con el fin de saber diseñarlos correctamente. Este fue un periodo de duro trabajo y por tanto, se dedicaba 7 horas diarias durante 6 días a la semana, haciendo un total de **252 horas** aproximadamente.
- en el mes de **noviembre** acabada ya la maqueta, las dos únicas tareas pendientes que quedaban eran: por una parte completar la memoria del proyecto y por otra parte se tomó el tiempo para programar todo el sistema, realizar cientos de pruebas, analizando comportamientos y corrigiendo cientos de errores que se registraban con el fin de conseguir un sistema perfecto y que se adaptase a las necesidades que se pretendía resolver.

Teniendo en cuenta ese duro trabajo y abultado se invertía unas 9 horas diarias durante los 7 días de la semana consiguiendo un total de **270 horas** con la posibilidad de haber trabajado algunos días más horas de las justificadas.

- En el mes de **diciembre y enero**, se dedicó a completar la memoria, a realizar las revisiones corrigiendo errores y modificando conceptos que o estaban claros o bien explicados. También se utilizó ese tiempo para formalizar el documento y darle un aspecto más enriquecido y a la vez que se seguía investigando sobre algunas mejoras que se podrían volver agregar al sistema o mejorar lo que ya se tenía hecho ya. Al ser un trabajo menos intensivo se dedicaba 4 horas al día y durante 5 días de la semana consiguiendo un total de **140 horas**.

Tabla 24: horas de trabajo invertido y precio

Mes	Horas trabajadas	Precio*hora (8,75€)
Octubre	16	140
Noviembre y diciembre	96	840
Enero y Febrero	200	1750
Marzo	90	787,5
Abril y Mayo	128	1120
Junio y Julio	112	980
Septiembre y octubre	252	2205
Noviembre	270	2362,5
Diciembre y enero	140	1225
Total	1304	11410
Meses totales 15	Componentes 87,94€	11497,94€

6.2. Presupuesto del entorno real

6.2.1. Componentes electrónicos

Los posibles componentes electrónicos que se utilizarían en un entorno real se pueden conseguir en diferentes tiendas de informática o electrónicas online y físicas situadas en diferentes puntos del país.

➤ **Relés**

4 módulos relés 5V de 8 canales cada uno para Arduino 4x5,85 € con gastos de envío gratos (15-20 laborales) ubicación Hong Kong

<http://www.ebay.com/itm/8-Channel-5V-Relay-Interface-Module-Board-w-Optocoupler-Isolationby-PIC-Arduino-/271738487715?&trksid=p2056016.m2518.14276>

➤ **Interruptores diferenciales**

Un interruptor diferencial de 63A de 25€ con gastos de envío 6€ (entrega 5-7 días laborales) ubicación España

http://www.ebay.es/sch/i.html?_odkw=interruptor+diferencial+25A&_osacat=0&_from=R40&_trksid=m570.11313&_nkw=interruptor+diferencial+63a&_sacat=0

Dos interruptores diferenciales de 15A de 2x15€ con gastos de envío 6€ (entrega 5-7 días laborales) ubicación España

<http://www.ebay.es/itm/INTERRUPTOR-DIFERENCIAL-MERLIN-GUERIN-2-POLOS-25-A-0-030mA-/231775974625?hash=item35f6ecb4e1:g:lzAAAOSw8-tWZHCB>

Dos interruptores diferenciales de 10A de 2x6,85€ con gastos de envío 6€ (entrega 10-15 días laborales) ubicación China.

http://www.lightinthebox.com/es/1p-10a-chnt-rccb-230v-interruptor-diferencial-interruptor-explosion-magnetica-proteccion-diferencial_p2719707.html

➤ **Cables jumper Arduino**

40 piezas de 10 cm con conexión macho-hembra 1,51€ con gastos de envío gratis (entrega 10-20 días laborales) ubicación Hong Kong

<http://www.ebay.es/itm/40x-10cm-Macho-a-Hembra-Breadboard-Protoboard-Jump-Wire-Cable-Jumper-for-Arduino/381391506754?hash=item58ccb48d42:g:hFUAOSw0JpV7VSf>

➤ **Caja de conexiones eléctricas**

Caja De Conexiones De 300 X 220 X 120 mm resistente al agua de plástico gris ip56 Pvc Adaptable 21,81€ con gastos de envío 3,95 (entrega 4-10 días laborales) ubicación Reino Unido

<http://www.ebay.es/itm/Junction-Box-300-x-220-x-120-mm-Weatherproof-Plastic-Grey-IP56PVCAdaptableNEW/300997753652?hash=item4614dd3b34:g:3wAAAOSw8IBTpF8H>

➤ **Barreras infrarrojas**

Para el control de las barreras infrarrojas se utilizaron los módulos fotoeléctricos E3JM-10M4-N 7x121,44 € gastos de envío gratis para ventas al por mayor (entrega en 48 horas laborales) se pueden adquirir en la tienda online RS.es

<http://es.rs-online.com/web/p/sensores-fotoelectricos/0256427/>

Con el fin de obtener toda la información sobre estos dispositivos se puede visitar la página del fabricante

<http://www.ia.omron.com/product/item/e3jm7087g/>

➤ **Sensor de movimiento exterior**

Dos módulos de sensor de movimiento exterior 2x6,5 € con gastos de envío 5,50€ (entrega entre 5-10 días laborales) ubicación España.

<http://www.ebay.es/itm/Maclean-Detector-de-movimiento-de-pared-180-Exterior-Interior-Sensor-oscuridad-/121545458226?hash=item1c4cac8232:g:z3YAAOSwxYxUuPjX>

➤ **Interruptor fotoeléctrico para el control de luminosidad**

Un interruptor fotoeléctrico o sensor LDR 7,50 € con gastos de envío 2,40€ (entrega entre 5-7 días laborales) ubicación España

<http://www.ebay.com/itm/INTERRUPTOR-FOTOELECTRICO-DIA-Y-NOCHE-REGULABLE-LUZ-AMBIENTE-30-150-LUX-2W/280940105733?hash=item416955d005:g:Q4sAAOSw7I5Tw7D2>

➤ **Barreras de iluminación**

Doce barreras de iluminación *LightGuardSystem* de 5 módulos cada una. Pero el precio de adquisición no ha podido ser encontrado, pesé a que les escribí a la empresa fabricante pero no respondieron.

Si se asemeja le precio de estos módulos con el de las balizas automáticas utilizadas ya en el tráfico para la señalización se podrían estimar un precio de 90€ por barrera ya que cada uno constad de seis módulos, lo que hace un total de 1080€ las 12 barreras distribuidas en todo el sistema.

Toda la información relacionada con esos módulos se puede encontrar en su página oficial.

<http://www.lightguardsystems.com/>

Materiales	Precio
Arduino ATMmega2560	13,40€
Relés de ocho canales	23,€
Interruptores diferenciales 63 , 15, 10A	68,7€
Cables jumper	1,51€
Caja de conexiones	25,76€
Módulos de barreras luminosas <i>LightGuard</i>	1080€
Módulos de barreras infrarrojas	850,08€
Señor de movimiento	18,5€
Sensor de luz fotoresistivo	10€
Total (IVA incluido)	2090,95€

CAPÍTULO 7: **CONCLUSIONES**

7. CONCLUSIONES

Partiendo de que para llevar a cabo el desarrollo de este trabajo fin de grado se ha invertido tiempo excesivo en la investigación, ya sea tanto para la búsqueda de información como el desarrollo de diversos temas tratados a lo largo de su evolución. Dentro de esos se pueden destacar como más importantes los siguientes: los reguladores PID, domótica, regulación del tráfico.

Valorando todas las horas de trabajo invertido, puedo decir que me han servido para adquirir y ampliar mis conocimientos sobre la utilización y el funcionamiento de los reguladores PID, así como las posibles áreas de aplicación y las competencias sobre su uso. Un factor fundamental que he tenido en cuenta es la importancia que tiene usar la programación para crear sistemas autosuficientes y dotados de cierta inteligencia artificial para controlar un proceso que tal vez utilizando otros métodos serían más laboriosos o menos eficientes.

Otro punto esencial tenido en cuenta fue la aplicación de las matemáticas, las leyes de electricidad y/o electrónica y los fundamentos básicos de la física para resolver los diversos problemas de ingeniería surgidos a lo largo del desarrollo de este proyecto. Gracias a todos esos conocimientos adquiridos pude completar con resultados satisfactorios y útiles para incrementar mi formación en el grado de ingeniera cursado durante cuatro años.

Como ya se ha repetido en varias ocasiones, la realización de este proyecto tenía como objetivo construir un entorno de pruebas desde cero y dotarlo de un sistema electrónico de la manera más económica posible con el fin del conseguir realizar pruebas que resuelvan todos los problemas de tráfico expuestos anteriormente y así incrementar la seguridad y evitar incidentes desagradables.

A la vez que se fue desarrollando el proyecto se pensó en aplicar técnicas de uso de un regulador PID, lo que supuso un plus de mejora para el proyecto, ya que gracias a ello se lograron conseguir muchas mejoras, entre ellas, que se pudiera calcular la velocidad simulada de los vehículos y con esos datos involucrados en gran mayoría y juntos a otros parámetros calculados independientes del regulador proporcionar una salida apropiada. En un principio resultó muy difícil llegar a entender y poner en práctica los reguladores PID y su aplicación a dicho entorno, pero gracias a los conocimientos adquiridos en regulación automática me facilitó mucho su comprensión y poder elaborarlo.

Con el fin de poder realizar todo lo anterior con garantías, en la primera fase del proyecto, se realizó un estudio del estado del conocimiento de los problemas registrados y no resueltos en la seguridad vial, con el objetivo de poder llegar a comprender mejor el funcionamiento de los distintos actuadores y sensores que se iban a incorporar en nuestro sistema. Durante esta fase pudimos ver los posibles problemas que más adelante tendríamos que afrontar con algunas soluciones propuestas. Tras estas soluciones propuestas y el estudio propio realizado sirvió para plantear nuestras propias soluciones y cumplir con los objetivos marcados inicialmente.

Con el fin de llevar un estudio más organizado se decidió dividir el proyecto en una serie de bloques como pueden ser:

- Diseño y construcción desde cero de un entorno de pruebas.
- Implementación de unas barreras de iluminación que sirvan de apoyo para simular la regulación del tráfico con el fin de reducir el número de atropellos producidos en zonas de poca visibilidad y con mucha concurrencia peatonal
- Modelación de un sistema con regulador PID y a base de sensores y actuadores que defina el comportamiento general del sistema.
- Suposición de la utilización de este sistema en un entorno real.

Todos los objetivos marcados se han cumplido. En primer lugar se diseñó y construyó una representación virtual con AutoCAD y posteriormente una maqueta física de una vía con cruce de peatones en que se simulara el tráfico de vehículos.

En segundo lugar se diseñaron e implementaron mediante dispositivo Arduino una técnica de control binario 0 y 1 para, detectar la ausencia o presencia de peatones en las intersecciones.

En tercer lugar se diseñaron e implementaron un regulador PID para el cálculo de la velocidad gracias a un subsistema de barreras infrarrojas integradas en el circuito y aplicando una técnica de control con señales analógicas que varían entre 0 y 1024 bits.

En cuarto lugar se implementaron mediante el dispositivo Arduino un subsistema utilizando un sensor LDR con el fin de activar el sistema de manera automática en función del grado de luminosidad presente en el ambiente. La técnica de control utiliza una configuración *pull-up* y con señales analógicas que varían entre 0 y 1024 bits.

En función a toda la tecnología implementada se obtiene como salida la activación o desactivación de los actuadores que en este caso eran las barreras luminosas con el fin de crear un aviso prolongado sobre la situación o comportamiento del vehículo una vez se encuentre dentro del área de control.

Y por último, una vez diseñados e implementados todos los controles del sistema, se definió el comportamiento general del sistema de control que permita reforzar el tráfico mediante la programación.

En relación a todos los objetivos marcados y visto los resultados positivos obtenidos en las simulaciones realizadas, que por recordar la misión que este sistema debía desempeñar era crear un alerta a los conductores sobre su situación a decenas de metros con el fin de que estos pudieran evitar incidentes o siniestralidad y llegar de forma segura al paso de peatones, se puede decir con toda certeza que este sistema cumple con tales objetivos y puede ser utilizado para intentar resolver este problema que se viene arrastrando desde hace décadas.

Una vez finalizado el entorno de pruebas y comprobado que funciona correctamente se hizo un estudio, para saber que se necesitaría para llevarlo a cabo sobre un **entorno real**. El programa creado podría valer casi en su totalidad, pero habría que realizar algunas modificaciones, ya que se dejaría de utilizar unos diodos infrarrojos en las barreras y se uti-

lizarían dispositivos altamente cualificados para generar barreras infrarrojas; o el cambio del sensor LDR con el interruptor de control de Luz funcionando a 230v con corriente continua.

Me gustaría resaltar que la realización de este proyecto ha sido una de las mejores experiencias, con un final muy satisfactorio durante el desarrollo formativo en el grado de ingeniería eléctrica, puesto que se puede ver en él, gran parte de los conocimientos adquiridos durante estos 4 años de carrera. Para cualquier estudiante, es de gran satisfacción que un proyecto donde se inicia con ideas, suposiciones, colaboración del profesorado dándome sugerencias, y afrontar retos de temas a tratar sobre los que no dispones de conocimientos iniciales; se logre el funcionamiento correcto.

Para terminar, mencionar mi total disposición y cedo dicho entorno de pruebas a la Universidad, para que sirva tanto como muestrario para la atracción de nuevos alumnos o para el uso de futuros alumnos, donde poder coger ideas o incluso para reutilizarlo como un nuevo TFG, donde puedan añadir nuevos elementos de control, como por ejemplo implementar o modificar el sistema para vías de doble sentido.

CAPÍTULO 8: BIBLIOGRAFÍA

8. Bibliografía

- [1] Diccionario enciclopédico popular ilustrado Salvat (1906-1914)
- [2] *The Origins and Globalization of Traffic Control Signals* - Los orígenes y la globalización de las señales de control de tráfico - en inglés
- [3] Sites.Tufts.edu *The Origins and Globalization of Traffic Control Signals* - Los orígenes y la globalización de las señales de control de tráfico, Clay McShane, Northeastern University, Pag. 381 - en inglés - Consultado el 08 de diciembre 2013
- [4] Artículo de documentación de Pedro Gómez Esteban
- [5] Explicación de *Luis Camilla*, director de las carreras de Electricidad y Electrónica del DuocUC de Antonio Varas
- [6] Artículo Crónica Viva- ciencias y tecnología. Romina Cannoni
- [7] Artículo EL MUNDO.es: Los semáforos de Lugo se visten hoy de mujer
- [8] Heckhausen, Markus y otros (1997). *Das Buch vom Ampelmännchen* (en alemán)
- [9] A. S. Khalid, A. Jaiz, and F. Wajdi, Dynamic traffic light sequence algorithm using rfid, *Computer Science*, vol. 4, pag. 517-524, 2008.
- [10] K. Yousef, J. Al-Karaki, and A. M. Shatnawi, Intelligent traffic light control system using wireless sensors networks, *Information Science and Engineering*, vol. 26. pag. 753-768, 2010.
- [11] A. Goel, S. Ray, and N. Chandra, Intelligent traffic light system to prioritized emergency purpose vehicles based on wireless sensor network, *Computer Applications*, vol. 40, pag. 36-39, 2012.
- [12] M. Arora and V. Banga, Intelligent traffic light control system using morphological edge detection and fuzzy logic, *Intelligent Computational Systems*, vol. 7, pag. 151-154, 2012.
- [13] Coches: noticias.coches.com/noticias-motor/tra_radar-el-semaforointeligente-con-el-ambar/55406.
- [14] S. Lizun: www.lyzunsistemas.com.ar/microsite/transito.php
- [15] Digicon: www.digicon.com.br/site/es/transito/menutransitoscats.html.
- [16] David Salomon (1993). *Assemblers and Loaders*

- [17] **L. Tokheim Roger** Fundamentos de los microprocesadores [Libro] Mexico: Mc Graw Hill, 1991.
- [18] tecnología Laser: Enciclopedia de fotónica y tecnología láser (en inglés).
- [19] Michael De Podesta (2002). *Understanding the Properties of Matter*. CRC Press. p. 131. ISBN 0415257883.
- [20] F. K. Kneubühl, M. W. Sigrist: *Laser*. 3. Auflage. Teubner, 1991, p. 4.
- [21] «Teletransporte en un rayo láser». <http://www.elmundo.es/>. 23 de junio de 2002. Consultado el 21 de noviembre de 2007.
- [22] «El Museo Británico abre al público el centro de realidad virtual de Silicon Graphics». <http://www.sgi.com/>. Consultado el 25 de agosto de 2012.
- [23] **Ruiz Gutiérrez José Manuel** Manual de Programación Arduino. [Libro]. - [s.l.]: Autoedición, 2008.
- [24] <http://www.lightguardsystems.com/>
- [25] <https://es.wikipedia.org/wiki/Led>
- [26] <http://www.asifunciona.com/tablas/leds/leds.htm>
- [27] http://ccaa.elpais.com/ccaa/2015/12/28/valencia/1451317921_876098.html
- [28] <http://www.elcorreo.com/alava/20090719/rioja/logrono-registra-medio-centenar-20090719.html>
- [29] Fundación Mutua Madrileña: estudio de la siniestralidad y atropellos 2014
- [30] Sistema de control basado en Arduino de una piscina climatizada: *Eduardo Valero Alonso*. EIMIA 2014-2015