



UNIVERSIDAD de VALLADOLID



ESCUELA de INGENIERÍAS INDUSTRIALES

**«INGENIERO TÉCNICO DE TELECOMUNICACIONES, ESPECIALIDAD EN
SISTEMAS ELECTRÓNICOS»**

PROYECTO FIN DE CARRERA

ROBOT MATRICIAL

Autor:

Mena Benito, Igor

Tutor:

Díez, Pedro Luis

Tecnología Electrónica

Junio 2013

CONTENIDO

1. INTRODUCCIÓN.....	9
1.1. Motivación del Proyecto	9
1.2. Introducción a la robótica.....	10
1.2.1. Definiciones Básicas.....	10
1.2.2. Clasificación de los robots	12
1.2.2.1. En base a su arquitectura	12
1.2.2.2. En base a su aplicación:	15
1.2.2.3. En base a su evolución:.....	15
1.2.3. Componentes de un robot	16
1.2.4. Impacto de la robótica.....	16
1.2.4.1. Impacto en la educación/aprendizaje	16
1.2.4.2. Impacto en la automatización industrial	17
1.2.4.3. Impacto en la competitividad	17
1.2.4.4. Impacto sociolaboral	17
1.3. OpenSource.....	18
1.3.1. OpenSoftware.....	19
1.3.1.1. Libertades del software libre	20
1.3.1.2. Motivaciones del Software Libre	20
1.3.2. OpenHardware	21
1.3.2.1. Problemática.....	21
1.3.2.2. Ventajas y desventajas	22
1.4. Plan de trabajo	23
1.5. Objetivos.....	24
2. DESCRIPCIÓN DEL SISTEMA.....	25
2.1. Descripción general del proyecto	25
2.2. Descripción del hardware	25
2.2.1. ARDUINO	25
2.2.1.1. ¿Qué es Arduino?	25
2.2.1.2. Lenguaje de Programación	26
2.2.1.3. ¿Por qué Arduino?	27
2.2.1.4. Hardware	28
2.2.1.5. Arduino Mega	31
2.2.2. Motor de Corriente Continua	38
2.2.2.1. Partes de un motor CC	38
2.2.2.2. Principio básico de funcionamiento de un motor CC	38
2.2.3. Servomotor	39

2.2.3.1.	Definición	39
2.2.3.2.	Estructura	39
2.2.3.3.	Funcionamiento	40
2.2.4.	Sensor de Ultrasonidos.....	44
2.2.4.1.	Qué son los ultrasonidos	44
2.2.4.2.	Sensores de distancia por ultrasonidos	44
2.2.4.3.	Generación de ultrasonidos.....	46
2.2.4.4.	Principales efectos utilizados para implementar sensores de ultrasonidos.....	47
2.2.5.	LCD.....	52
2.2.5.1.	Funcionamiento	53
2.2.5.2.	Especificaciones	54
2.2.5.3.	Tecnologías de Matriz Activa	55
2.2.5.4.	Inconvenientes	56
2.2.6.	Lector de tarjetas.....	57
2.2.6.1.	Características	57
2.2.6.2.	Tipos	58
2.2.7.	Reloj en tiempo real	59
2.2.7.1.	Ventajas de un RTC.....	59
2.2.7.2.	Ejemplos	60
2.2.8.	Baterías.....	60
2.2.8.1.	Definición y Clasificación	60
2.2.8.2.	Funcionamiento Básico.....	60
2.2.8.3.	Características que definen a una batería	61
2.2.8.4.	Tipos de baterías:	62
2.2.8.5.	Comparativa entre baterías.....	65
2.3.	Descripción Software.....	66
2.3.1.	Arduino.....	66
2.3.1.1.	Comunicación Arduino-PC.....	66
2.3.1.2.	Instalación Software Arduino	67
2.3.1.3.	Programación Arduino	71
2.3.2.	Proteus	73
2.3.2.1.	Instalación del Software Proteus.....	73
2.3.2.2.	Isis (Simulador de Circuitos)	73
2.3.2.3.	ARES (Realizar PCB's).....	76
3.	DESARROLLO/DISEÑO DEL SISTEMA	80
3.1.	Diseño mecánico	80
3.2.	Diseño Hardware.....	80
3.2.1.	Motores CC.....	80
3.2.1.1.	Problema surgido.....	86
3.2.1.2.	Posibles soluciones	87
3.2.1.3.	Replanteamiento de los objetivos	88
3.2.2.	Baterías.....	88
3.2.3.	Lector De Tarjetas.....	89

3.2.4.	Sensores de ultrasonidos.....	90
3.2.5.	Servomotor.....	93
3.2.6.	Reloj Real.....	94
3.2.7.	LCD.....	96
3.3.	Diseño Software.....	99
3.3.1.	Inclusión de librerías.....	99
3.3.2.	Configuración de pines.....	100
3.3.3.	Definición de variables.....	100
3.3.4.	Configuración inicial de Arduino.....	101
3.3.5.	Bloques de funciones.....	101
3.3.5.1.	Funciones de control de motores.....	102
3.3.5.2.	Funciones de movimiento y decisión.....	113
3.3.5.3.	Funciones de control de baterías.....	122
3.3.5.4.	Funciones de Sensor ultrasonidos.....	124
3.3.5.5.	Función pantalla.....	126
3.3.6.	Interrupciones.....	127
3.4.	Protocolos de comunicación.....	131
3.4.1.	Comunicación serie.....	131
3.4.2.	USB (Universal Serial Bus).....	131
3.4.3.	Comunicación I2C.....	133
3.4.3.1.	Introducción.....	133
3.4.3.2.	Protocolo I2C.....	134
3.4.3.3.	Definición de términos.....	135
3.4.4.	Comunicación SPI.....	136
3.5.	Otras consideraciones.....	137
3.5.1.	Ámbitos de utilización.....	137
3.5.1.1.	Industrial.....	137
3.5.1.2.	Doméstico.....	138
3.5.1.3.	Educativo.....	138
3.5.2.	Consejos de utilización.....	138
4.	CÁLCULOS.....	140
4.1.	Pistas.....	140
4.2.	Consumos.....	148
4.3.	Rectas baterías.....	149
4.4.	Rectas PWM.....	151
5.	PLANOS.....	153
6.	PRESUPUESTO.....	155

6.1. Lista de Materiales	155
6.1.1. Parte mecánica.....	155
6.1.2. Parte Hardware.....	156
6.1.3. Parte Software.....	157
6.2. Presupuesto Económico	158
6.2.1. Parte mecánica.....	158
6.2.2. Parte hardware.....	159
7. PLIEGO DE CONDICIONES	161
7.1. Objetivo del pliego	161
7.1.1. Descripción general del montaje.....	161
7.2. Normativa de obligado cumplimiento	162
7.3. Condiciones generales de los materiales	163
7.3.1. Especificaciones eléctricas.....	164
7.3.2. Especificaciones mecánicas.....	165
7.3.3. Componentes electrónicos.....	165
7.4. Condiciones del proceso de fabricación	166
7.4.1. Preparación de los componentes.....	166
7.4.2. Material del circuito impreso.....	166
7.4.3. Soldadura y montaje de los componentes.....	166
7.4.4. Condiciones del proceso de prueba.....	166
7.5. Condiciones facultativas	167
7.6. Solicitud de homologación de tipo ce	168
7.6.1. Expediente técnico de construcción.....	168
7.6.2. Declaración de conformidad del producto.....	169
7.6.3. Marcado CE sobre el producto.....	170
7.7. Marca de reciclado de aparatos eléctricos y electrónicos	171
7.7.1. Marca de aparatos eléctricos y electrónicos.....	171
7.8. Cláusulas de índole legal	172
7.9. Conclusiones del pliego de condiciones	173
8. CONCLUSIONES	174
8.1. Objetivos logrados	174
8.2. Conclusiones Finales	174
8.3. Futuras Mejoras	175
8.3.1. Giroscopio/Brújula electrónica.....	175

8.3.2.	Acelerómetro	175
8.3.3.	Sensores láser.....	176
8.3.4.	Acoplar cámara web.....	176
8.3.5.	Acoplar módulo WiFi o GSM	176
8.3.6.	Control mediante página web o dispositivos móviles	177
8.3.7.	Sustitución de Arduino Mega por dos Arduinos Duemilanove	177
9. ANEXOS.....		178
9.1. Enlaces.....		178
9.2. Esquemas y Conexionado.....		178
9.2.1.	Despiece mecánico	178
9.2.2.	Conexionado con Arduino	181
9.2.3.	Dibujo de la batería en la LCD.....	185
9.3. Contenido del CD		186
9.3.1.	Arduino (software)	186
9.3.2.	Demo proteus (software)	186
9.3.3.	Datasheet componentes	186
9.3.4.	Códigos	187
9.3.5.	Lecturas SD	187
9.3.6.	Fotos	187
9.3.7.	Imágenes	187
9.3.8.	Vídeos	187
9.3.9.	Fritzing (software)	187
9.3.10.	Planos	187
10. BIBLIOGRAFÍA		188

ÍNDICE DE FIGURAS

FIGURA 1: ROBOT POLIARTICULADO	12
FIGURA 2: ROBOT MÓVIL	13
FIGURA 3: ROBOT ANDROIDE.....	13
FIGURA 4: ROBOT ZOOMÓRFICO	14
FIGURA 5: ROBOT HÍBRIDO	14
FIGURA 6: COMPONENTES DE UN ROBOT	16
FIGURA 7: ARDUINO MEGA.....	31
FIGURA 8: PINES ARDUINO MEGA.....	34
FIGURA 9: DESPIECE DE UN SERVOMOTOR.....	39
FIGURA 10: PARTES DE UN SERVOMOTOR.....	40
FIGURA 11: COLORES DE LOS CABLES DE LOS PRINCIPALES FABRICANTES DE SERVOMOTORES	41
FIGURA 12: PWM Y RANGO DE MOVIMIENTO DE SERVOMOTORES	41
FIGURA 13: RANGO DE FRECUENCIAS	44
FIGURA 14: MATERIALES PIEZOELÉCTRICOS	46
FIGURA 15: EFECTO PIEZOELÉCTRICO	47
FIGURA 16: CONO DE RADIACIÓN FUNDAMENTAL.....	47
FIGURA 17: CÁLCULO DE LA DISTANCIA EN OBJETOS INMÓVILES	48
FIGURA 18: MÁRGENES DE MEDIDA DE LOS SENSORES DE ULTRASONIDOS.....	48
FIGURA 19: DIAGRAMA DE BLOQUES DE UN SENSOR DE ULTRASONIDOS BASADO EN EL EFECTO DOPPLER.....	51
FIGURA 20: FUNCIONAMIENTO DE UN CAUDALÍMETRO	52
FIGURA 21: DESPIECE DE UNA PANTALLA LCD	52
FIGURA 22: EJEMPLO DE BATERÍA.....	61
FIGURA 23: COMPARATIVA ENTRE BATERÍAS	65
FIGURA 24: ENERGÍA Y DENSIDAD ESPECÍFICA DE VARIAS BATERÍAS	65
FIGURA 25: PUERTOS DE COMUNICACIÓN DEL ARDUINO MEGA.....	66
FIGURA 26: CAPTURA DE PANTALLA DE LA OPCIÓN SERIAL MONITOR	67
FIGURA 27: CAPTURA DE PANTALLA DE LA CARPETA ARDUINO DESPUÉS DE LA INSTALACIÓN.....	68
FIGURA 28: CAPTURA DE PANTALLA DE LA INSTALACIÓN DEL SOFTWARE DE ARDUINO.....	68
FIGURA 29: PANTALLA PRINCIPAL DEL SOFTWARE ARDUINO	69
FIGURA 30: COMPILACIÓN EN ARDUINO	70
FIGURA 31: CARGAR PROGRAMA EN ARDUINO.....	70
FIGURA 32: INCLUSIÓN DE LIBRERÍAS EN ARDUINO	71
FIGURA 33: DEFINICIÓN DE VARIABLES COMPLEJAS.....	71
FIGURA 34: DEFINICIÓN DE VARIABLES.....	71
FIGURA 35: CONFIGURACIÓN INICIAL EN ARDUINO	72
FIGURA 36: COMIENZO DE LA FUNCIÓN PRINCIPAL DEL PROGRAMA EN ARDUINO	72
FIGURA 37: CREACIÓN DE FUNCIONES EN ARDUINO.....	72
FIGURA 38: CAPTURA DE PANTALLA DEL SOFTWARE PROTEUS	73
FIGURA 39: PANTALLA INICIAL DE PROTEUS	74
FIGURA 40: SELECCIÓN DE COMPONENTES EN PROTEUS.....	74
FIGURA 41: POSICIONAMIENTO DE COMPONENTES EN PROTEUS	74

FIGURA 42: EDICIÓN DE COMPONENTES EN PROTEUS.....	75
FIGURA 43: SIMULACIÓN EN PROTEUS	75
FIGURA 44: PANTALLA PRINCIPAL DE ARES.....	76
FIGURA 45: CAMBIAR DE ISIS A ARES EN PROTEUS.....	76
FIGURA 46: PANTALLA DE DISEÑO DE ARES.....	77
FIGURA 47: SELECCIÓN DE ENCAPSULADOS EN ARES	77
FIGURA 48: SELECCIÓN DE TAMAÑO PCB EN ARES.....	78
FIGURA 49: DISTRIBUCIÓN DE ENCAPSULADOS EN ARES	78
FIGURA 50: ROUTING DE PISTAS EN ARES.....	79
FIGURA 51: VISTA 3D EN ARES	79
FIGURA 52: INTEGRADO L293D	80
FIGURA 53: DIAGRAMA DE PINES L293D.....	81
FIGURA 54: FUNCIONAMIENTO DEL CIRCUITO	82
FIGURA 55: CONEXIONADO PARA DOS MOTORES CON GIRO EN AMBOS SENTIDOS.....	82
FIGURA 56: DISEÑO DE CONEXIONADO DE MOTORES Y ARDUINO	84
FIGURA 57: CONFIGURACIÓN DE LOS PINES DE ARDUINO PARA EL CONTROL DE MOTORES.....	85
FIGURA 58: FUNCIONES DE CONTROL DE LOS MOTORES	86
FIGURA 59: LECTOR DE TARJETAS SD MODULE.....	89
FIGURA 60: FUNCIÓN PARA LA ESCRITURA EN UNA TARJETA SD	90
FIGURA 61: SENSOR DE ULTRASONIDOS HC-SR04	90
FIGURA 62: CÓDIGO DE PRUEBA PARA SENSOR DE ULTRASONIDOS.....	91
FIGURA 63: DISTANCIA A UN OBSTÁCULO MEDIANTE SERIAL MONITOR.....	92
FIGURA 64: CÓDIGO DE PRUEBA PARA SERVOMOTOR.....	94
FIGURA 65: RELOJ REAL TINY RTC DS1307	94
FIGURA 66: CONFIGURACIÓN INICIAL DE RELOJ REAL	95
FIGURA 67: CÓDIGO DE PRUEBA PARA RELOJ REAL.....	96
FIGURA 68: PANTALLA LCD 2X16.....	97
FIGURA 69: CÓDIGO DE PRUEBA PARA PANTALLA LCD	97
FIGURA 70: MENSAJE DE BIENVENIDA EN EL ROBOT.....	98
FIGURA 71: MENSAJE DE ALERTA POR FALTA DE TARJETA EN ROBOT	99
FIGURA 72: LOGOTIPO DE USB.....	132
FIGURA 73: CABLEADO DE UN USB	132
FIGURA 74: ESTRUCTURA DE UN BUS SDA Y SCL.....	133
FIGURA 75: SECUENCIA DE INICIO EN I2C	134
FIGURA 76: INTERCAMBIO DE INFORMACIÓN EN I2C.....	134
FIGURA 77: : SECUENCIA DE PARADA EN I2C	135
FIGURA 78: COMUNICACIÓN SPI.....	137
FIGURA 79: SIMULACIÓN DEL CIRCUITO DE MOTORES EN ISIS.....	140
FIGURA 80: PICOS DE CORRIENTE EN LOS MOTORES I.....	142
FIGURA 81: PICOS DE CORRIENTE EN LOS MOTORES II.....	142
FIGURA 82: DISEÑO FINAL DE LA PCB.....	145
FIGURA 83: CARACTERÍSTICAS FINALES DE LA PCB	145
FIGURA 84: VISIÓN DE LA PCB EN 3D	146
FIGURA 85: PCB INSOLADA.....	147
FIGURA 86: PCB CONECTADA A LOS MOTORES PARA PRUEBAS.....	147
FIGURA 87: CIRCUITO DIVISOR DE TENSIÓN	149
FIGURA 88: CIRCUITOS DIVISORES DE TENSIÓN EN NUESTRO ROBOT	149

FIGURA 89: RECTA DE CONVERSIÓN DE VOLTIOS A BITS	150
FIGURA 90: MENSAJES DE ALERTA POR LA FALTA DE BATERÍA DE LOS MOTORES Y ARDUINO.....	150
FIGURA 91: GRÁFICA DE DESCARGA DE BATERÍA	151
FIGURA 92: GRÁFICA DE CONVERTIDOR DE BITS A PWM (MOTOR DERECHO).....	151
FIGURA 93: GRÁFICA DE CONVERTIDOR DE BITS A PWM (MOTOR IZQUIERDO)	152
FIGURA 94: LOGOTIPO CE.....	170
FIGURA 95- LOGOTIPO DE MARCADO DE APARATOS ELÉCTRICOS Y ELECTRÓNICOS.....	171
FIGURA 96: AMPLIACIÓN DE PINES GND Y VCC	184

ÍNDICE DE TABLAS

TABLA 1: LIBERTADES DEL SOFTWARE LIBRE	20
TABLA 2: CARACTERÍSTICAS DEL ARDUINO MEGA	32
TABLA 3: GIRO DE LOS MOTORES.....	83
TABLA 4: MEDIDA DE CORRIENTE SEGÚN VELOCIDAD DE LOS MOTORES.....	141
TABLA 5: CONSUMOS DE LOS DISPOSITIVOS	148
TABLA 6: PARTE MECÁNICA DEL ROBOT	155
TABLA 7: PARTE HARDWARE DEL ROBOT.....	156
TABLA 8: PRESUPUESTO PARTE MECÁNICA	158
TABLA 9: PRESUPUESTO PARTE HARDWARE.....	159
TABLA 10: PRESUPUESTO PCB Y DIVISORES DE TENSIÓN	160

1. INTRODUCCIÓN

Considerando el rápido desarrollo de la ciencia y la tecnología en un mundo globalizado, dónde existen cambios e innovaciones, se puede profundizar en un tema tan actual e interesante como es la robótica. Una rama de la tecnología relativamente nueva, pero, a pesar de ello, un motor que ha impulsado el avance de la humanidad. Esta trata sobre el diseño, desarrollo y creación de máquinas capaces de realizar tareas que para el ser humano son difíciles y hasta peligrosas de efectuar.

Para el desarrollo de la robótica también ha sido necesario la utilización de otras ramas de la tecnología que ayuden a su desarrollo, como son las matemáticas, la física, la electrónica, la informática, la mecánica, etc...

Día a día podemos observar la increíble evolución que la tecnología ha tenido, haciendo posible crear máquinas con una gran autonomía, ayudando a reducir así muchas tareas que para el ser humano serían difíciles de efectuar o, al menos, tardaría mucho en realizar como son las actividades repetitivas. Pero la principal función de éstas es reemplazar la mano de obra del hombre en aquellos trabajos donde las condiciones no son las adecuadas. Para ello, el hombre se ha puesto como objetivo el construir máquinas capaces de imitar a algunas partes del cuerpo humano. La industria, ha sido la gran beneficiada de ésta tecnología, caracterizada por grandes innovaciones, adelantos y avances tecnológicos.

A pesar de todos los avances realizados en el ámbito de la robótica, todavía queda un largo camino hasta lograr máquinas que consigan reproducir movimientos y pensamientos humanos casi a la perfección. Estamos seguros de que el hombre conseguirá sus propósitos de llegar a crear inteligencia artificial.

1.1. MOTIVACIÓN DEL PROYECTO

Este proyecto ha sido desarrollado de forma conjunta con el alumno Alberto Rodríguez Campos de la Especialidad I.T.I. Electrónica Industrial.

Partiendo de nuestros conocimientos adquiridos durante nuestro proceso de formación en esta universidad, hemos decidido realizar un proyecto basado en la robótica.

Como herramienta para llegar al objetivo de comprender el control de sensores, periféricos, baterías, motores y la programación de microcontroladores, hemos decidido la utilización de OpenSource y OpenHardware, debido a su fácil manejo y bajo coste.

1.2. INTRODUCCIÓN A LA ROBÓTICA

1.2.1. DEFINICIONES BÁSICAS

¿QUÉ ES UN ROBOT?

Existen ciertas dificultades a la hora de establecer una definición formal de lo que es un robot.

Las definiciones son muy dispares:

“Es un dispositivo reprogramable y multifuncional diseñado para mover materiales, piezas, herramientas o dispositivos especializados a través de movimientos programados”

- Robot Institute of America, 1979 –

“Un dispositivo automático que realiza funciones que normalmente se considera que son o debieran ser realizadas por humanos”

- Diccionario Webster –

“Un sistema que existe en el mundo físico y que autónomamente sensa su medio ambiente y actúa sobre él”

- Maja Mataric/USC –

“Un mecanismo reprogramable con un mínimo de cuatro grados de libertad diseñado para manipular y transportar partes, herramientas o implementar manufactura especializada a través de movimientos programados para la ejecución de la una tarea específica de manufactura”

- Asociación Británica de Robótica (BRA) –

Máquina o ingenio electrónico programable, capaz de manipular objetos y realizar operaciones antes reservadas sólo a las personas.

- Diccionario Real Academia –

"Puede que no sea capaz de definir qué es un robot, pero sé cuándo veo uno"

– *Joseph Engelberg (padre de la robótica industrial)* –

ROBÓTICA

El término 'Robótica' fue acuñado por Isaac Asimov para describir la tecnología de los robots.

Procede de las palabras checas **robot** (trabajo forzado) y **robotnik** (sirviente). Usadas por primera vez en 1921 por el escritor checo Karel Capek en su obra de teatro Rossum's Universal Robot (R.U.R.) que hacía referencia a un humanoide mecánico.

Podemos definir la **Robótica** como una disciplina que combina todas aquellas actividades relacionados con el estudio, diseño, construcción, operación y mantenimiento de robots. Es un campo de trabajo que combina diferentes disciplinas como Ingeniería Eléctrica, Ingeniería Electrónica, Ingeniería Mecánica, Telecomunicaciones, Ciencias de la Computación, Matemáticas, Física, Biología, Neurociencias, etc.

Disciplinas Relacionadas:

VISIÓN COMPUTACIONAL

INTELIGENCIA ARTIFICIAL/COMPUTACIONAL

CIBERNÉTICA (CONTROL + CIENCIAS DE LA INFORMACIÓN + BIOLOGÍA)

MECATRÓNICA (MECÁNICA + ELECTRÓNICA + COMPUTACIÓN)

Los primeros robots industriales modernos fueron denominados "Unimates" y fueron desarrollados a finales de la década de los 50 y principios de los 60 por George Devol y Joe Engelberger. Debido a desarrollos posteriores Engelberger es considerado como el padre de la robótica.

A finales de la década de los 70 y principios de los 80 el desarrollo de los robots industriales tuvo una rápida evolución, debido principalmente a grandes inversiones desarrolladas por la industria automotriz.

La palabra **robótica** fue utilizada por primera vez por el científico y escritor de ciencia ficción *Isaac Asimov* en 1942. Él propuso las llamadas leyes de la robótica:

Ley 0: Un robot no puede realizar ninguna acción, ni por inacción permitir que nadie la realice, que resulte perjudicial para la humanidad, aun cuando ello entre en conflicto con las otras leyes.

Ley 1: Un robot no puede dañar a un ser humano ni, por inacción, permitir que éste sea dañado.

Ley 2: Un robot debe obedecer las órdenes dadas por los seres humanos excepto cuando estas órdenes entren en conflicto con las leyes anteriores.

Ley 3: Un robot debe proteger su propia existencia hasta donde esta protección no entre en conflicto con las leyes anteriores.¹

1.2.2. CLASIFICACIÓN DE LOS ROBOTS

Es difícil establecer una clasificación rigurosa de los robots. Vamos a hacer diferentes clasificaciones:

1.2.2.1. EN BASE A SU ARQUITECTURA

POLIARTICULADOS

Son sedentarios y están estructurados para mover sus elementos terminales en un determinado espacio de trabajo con un número limitado de grados de libertad (manipuladores y algunos robots industriales).



FIGURA 1: ROBOT POLIARTICULADO

¹ *Introducción a la Robótica – Universidad de Chile*

MÓVILES

Cuentan con gran capacidad de desplazamiento, y están basados en carros o plataformas y dotados de un sistema locomotor de tipo rodante. Siguen su camino por telemando o guiándose por la información recibida de su entorno a través de sus sensores



FIGURA 2: ROBOT MÓVIL

ANDROIDES

Intentan reproducir total o parcialmente la forma y el comportamiento cinemático del ser humano. Uno de los aspectos más complejos de estos robots es la locomoción bípeda (control dinámico del movimiento y mantenimiento del equilibrio del robot).



FIGURA 3: ROBOT ANDROIDE

ZOOMÓRFICOS

Imitan los sistemas de locomoción de los diversos seres vivos.

NO CAMINADORES: basados en segmentos cilíndricos biselados acoplados axialmente entre sí y dotados de un movimiento relativo de rotación

CAMINADORES: múltiplos capaces de evolucionar en superficies muy accidentadas.

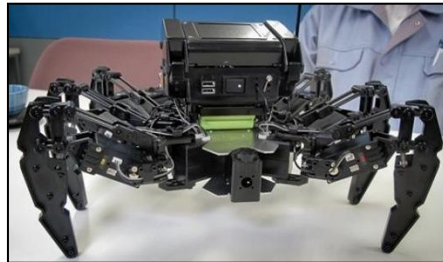


FIGURA 4: ROBOT ZOOMÓRFICO

HÍBRIDOS

Aquellos de difícil clasificación cuya estructura se sitúa en combinación con alguna de las anteriores (un carro móvil con un brazo, robot personal antropomorfo, etc.)



FIGURA 5: ROBOT HÍBRIDO

1.2.2.2. EN BASE A SU APLICACIÓN:

INDUSTRIALES

Son artilugios mecánicos y electrónicos destinados a realizar de forma automática determinados procesos de fabricación o manipulación.

SEGURIDAD Y ESPACIO

Nos referimos al uso de robots en tierra, mar y aire en misiones de seguridad civil o militar, así como a su utilización en misiones espaciales.

DE SERVICIOS

Sistemas aplicados en los dominios de la vida: entornos domésticos y de ocio, en salud y rehabilitación, en servicios profesionales y en ambientes peligrosos; que reproducen acciones de ayuda a los humanos.

1.2.2.3. EN BASE A SU EVOLUCIÓN:

1º GENERACIÓN

Sistema de control basado en “paradas fijadas mecánicamente” (mecanismos de relojería que mueven las cajas musicales o los juguetes de cuerda).

2º GENERACIÓN

El movimiento se controla a través de una secuencia numérica almacenada en disco o cinta magnética (industria automotriz).

3º GENERACIÓN

Utilizan las computadoras para su control y tienen cierta percepción de su entorno a través del uso de sensores. Con esta generación se inicia la era de los robots inteligentes y aparecen los lenguajes de programación.

4º GENERACIÓN

Robots altamente inteligentes con más y mejores extensiones sensoriales, para entender sus acciones y captar el mundo que los rodea. Incorporan conceptos “modélicos” de conducta.

5º GENERACIÓN

Actualmente se encuentran en desarrollo. Basarán su acción principalmente en modelos conductuales establecidos.

1.2.3. COMPONENTES DE UN ROBOT

ESTRUCTURA MECÁNICA

ACTUADORES

SENSORES (PERCEPCIÓN DEL ENTORNO)

PROCESADORES (SISTEMA DE CONTROL)



FIGURA 6: COMPONENTES DE UN ROBOT

1.2.4. IMPACTO DE LA ROBÓTICA

La robótica es una nueva tecnología multidisciplinar que hace uso de recursos de otras ciencias afines:

MECÁNICA; CINEMÁTICA; DINÁMICA; MATEMÁTICAS; AUTOMÁTICA; ELECTRÓNICA; INFORMÁTICA; ENERGÍA Y ACTUADORES ELÉCTRICOS, NEUMÁTICOS E HIDRÁULICOS; VISIÓN ARTIFICIAL; INTELIGENCIA ARTIFICIAL

1.2.4.1. IMPACTO EN LA EDUCACIÓN/APRENDIZAJE

- Formación de especialistas dada su gran demanda en el mundo industrial.
- Formación de especialistas en el uso de equipos médicos: cirugía robótica.

- Introducción de nuevas asignaturas en carreras de Ingeniería Superior y Técnica, facultades de informática y centros de formación profesional.
- La abundante oferta de robots educativos en el mercado y sus precios competitivos, permiten a los centros de enseñanza complementar el estudio teórico de la Robótica con prácticas.

1.2.4.2. IMPACTO EN LA AUTOMATIZACIÓN INDUSTRIAL

- Los robots permiten sistemas de fabricación flexibles que se adaptan a las diferentes tareas de producción.
- Las células flexibles disminuyen el ciclo de trabajo de un producto y liberan a las personas de trabajos desagradables y monótonos.
- La interrelación de las diferentes células flexibles a través de potentes computadores da lugar a la factoría totalmente automatizada.

1.2.4.3. IMPACTO EN LA COMPETITIVIDAD

- La adopción de la automatización en la fabricación de las poderosas compañías multinacionales obliga a todas las demás a seguir sus pasos para mantener su supervivencia.
- La automatización provoca un aumento de productividad y de calidad del producto, lo que hace que la empresa sea más competitiva.

1.2.4.4. IMPACTO SOCIOLABORAL

- La inversión económica de la automatización industrial se lleva a cabo a costa de la reducción de puestos de trabajo.
- El desempleo generado queda compensado por los nuevos puestos de trabajo que surgen en otros sectores: enseñanza, servicios, instalación, mantenimiento y fabricación de robots.
- También supone el mantenimiento del resto de puestos de trabajo, dado que la automatización de la misma provocará la revitalización y la salvación de empresa.²

² *Introducción a la Robótica – Rafael Barea*

1.3. OPENSOURCE

Código abierto (o *fuentes abiertas*) es el término con el que se conoce al software distribuido y desarrollado libremente. El código abierto tiene un punto de vista más orientado a los beneficios prácticos de poder acceder al código.

La idea del código abierto se centra en la premisa de que al compartir el código, el programa resultante tiende a ser de calidad superior al software propietario; es una visión técnica. Por otro lado, el software libre tiene tendencias filosóficas e incluso morales: el software propietario, al no poder compartirse, es "*antiético*", dado que prohibir compartir entre seres humanos va en contra del sentido común.

Al igual que el software libre, el *código abierto* tiene una serie de requisitos necesarios para que un programa pueda considerarse dentro de este movimiento; éstos son:

- Libre redistribución: el software debe poder ser regalado o vendido libremente.
- Código fuente: el código fuente debe estar incluido u obtenerse libremente.
- Trabajos derivados: la redistribución de modificaciones debe estar permitida.
- Integridad del código fuente del autor: las licencias pueden requerir que las modificaciones sean redistribuidas sólo como parches.
- Sin discriminación de personas o grupos: no se puede dejar fuera a nadie.
- Sin discriminación de áreas de iniciativa: los usuarios comerciales no pueden ser excluidos.
- Distribución de la licencia: deben aplicarse los mismos derechos a todo el que reciba el programa
- La licencia no debe ser específica de un producto: el programa no puede licenciarse sólo como parte de una distribución mayor.
- La licencia no debe restringir otro software: la licencia no puede obligar a que algún otro software que sea distribuido con el software abierto deba también ser de código abierto.
- La licencia debe ser tecnológicamente neutral: no debe requerirse la aceptación de la licencia por medio de un acceso por clic de ratón o de otra forma específica del medio de soporte del software.

1.3.1. OPENSOFTWARE

El software libre (en inglés free software, aunque esta denominación también se confunde a veces con "gratis" por la ambigüedad del término "free" en inglés, por lo que también se usan "libre software" y "logical libre") es la denominación del software que respeta la libertad de todos los usuarios que adquirieron el producto y, por tanto, una vez obtenido el mismo puede ser usado, copiado, estudiado, modificado, y redistribuido libremente de varias formas. Según la Free Software Foundation, el software libre se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, y estudiar el mismo, e incluso modificar el software y distribuirlo modificado.

El software libre suele estar disponible gratuitamente, o al precio de costo de la distribución a través de otros medios; sin embargo no es obligatorio que sea así, por lo tanto no hay que asociar software libre a "software gratuito", (denominado usualmente freeware), ya que, conservando su carácter de libre, puede ser distribuido comercialmente ("software comercial"). Análogamente, el "software gratis" o "gratuito" incluye en ocasiones el código fuente; no obstante, este tipo de software no es libre en el mismo sentido que el software libre, a menos que se garanticen los derechos de modificación y redistribución de dichas versiones modificadas del programa.

Tampoco debe confundirse software libre con "software de dominio público". Este último es aquel software que no requiere de licencia, pues sus derechos de explotación son para toda la humanidad, porque pertenece a todos por igual. Cualquiera puede hacer uso de él, siempre con fines legales y consignando su autoría original. Este software sería aquel cuyo autor lo dona a la humanidad o cuyos derechos de autor han expirado, tras un plazo contado desde la muerte de éste, habitualmente 70 años. Si un autor condiciona su uso bajo una licencia, por muy débil que sea, ya no es del dominio público.

1.3.1.1. LIBERTADES DEL SOFTWARE LIBRE

De acuerdo con tal definición, un software es "libre" cuando garantiza las siguientes libertades:

Libertad	Descripción
0	La libertad de usar el programa, con cualquier propósito.
1	La libertad de estudiar cómo funciona el programa y modificarlo, adaptándolo a tus necesidades.
2	La libertad de distribuir copias del programa, con lo cual puedes ayudar a tu prójimo.
3	La libertad de mejorar el programa y hacer públicas esas mejoras a los demás, de modo que toda la comunidad se beneficie.

Las libertades 1 y 3 requieren acceso al código fuente porque estudiar y modificar software sin su código fuente es muy poco viable.

TABLA 1: LIBERTADES DEL SOFTWARE LIBRE

1.3.1.2. MOTIVACIONES DEL SOFTWARE LIBRE

- Hay una motivación ética, abanderada por la Free Software Foundation, heredera de la cultura hacker y partidaria del apelativo *libre*, que argumenta que el software es conocimiento y debe poderse difundir sin trabas. Su ocultación es una actitud antisocial y la posibilidad de modificar programas es una forma de libertad de expresión, aunque sin olvidar una estructura jerarquizada por la meritocracia.
- La motivación pragmática, abanderada por la Open Source Initiative y partidaria del apelativo *abierto*, que argumenta ventajas técnicas y económicas, con respecto a evitar una tragedia de los anticomunes mejorando los incentivos.

Aparte de estas dos grandes motivaciones, la gente que trabaja en software libre suele hacerlo por muchas otras razones, que van desde la diversión a la mera retribución económica, que es posible debido a modelos de negocio sustentables.

1.3.2. OPENHARDWARE

Se llama hardware libre a los dispositivos de hardware cuyas especificaciones y diagramas esquemáticos son de acceso público, ya sea bajo algún tipo de pago o de forma gratuita. La filosofía del software libre (las ideas sobre la libertad del conocimiento) es aplicable a la del hardware libre. Se debe recordar en todo momento que libre *no* es sinónimo de gratis. El hardware libre forma parte de la cultura libre.

1.3.2.1. PROBLEMÁTICA

DEPENDENCIA TECNOLÓGICA EXTRANJERA DE LOS COMPONENTES

Al intentar fabricar un diseño, es posible encontrarse con el problema de la falta de material. En un país puede no darse este problema, pero en otros puede que no se encuentren los materiales necesarios y está demostrado que las TIC son herramientas indispensables para el desarrollo de las naciones, por lo cual es de importancia vital, a la vez que estratégica, el que cada nación no dependa de otra para su desarrollo tecnológico.

ALTOS COSTOS DE PRODUCCIÓN

La persona que desea utilizar el hardware que un tercero ha diseñado, primero lo tiene que fabricar, para lo cual tendrá que comprobar los componentes necesarios, construir el diseño y verificar que se ha hecho correctamente. Todo esto tiene un costo.

EL CONOCIMIENTO LO POSEEN POCAS EMPRESAS

Se sigue reteniendo el conocimiento en las grandes industrias productoras; como resultado el consumidor del producto tiene que adecuarse al producto que ofrece el mercado, que es por lo general un producto genérico que no cumple con las necesidades muy específicas de un determinado consumidor; allí es cuando se ata al usuario a las decisiones de las empresas productoras, y no se le da la libertad de elegir.

GRAN INVERSIÓN DE TIEMPO EN TRABAJOS DE DISEÑO REDUNDANTES

Tanto en el hardware como en el software propietario existe mucho diseño redundante, es decir, se "reinventa la rueda" en vez de usar ese conocimiento previo e innovar en nuevas áreas de investigación y producción.

1.3.2.2. VENTAJAS Y DESVENTAJAS

VENTAJAS

- Protege y defiende la soberanía, permitiendo a las naciones no depender de ninguna otra que le provea los recursos necesarios para su desarrollo e independencia tecnológica.
- Fomenta que el hardware pueda ser de calidad y los estándares, abiertos y más económicos.
- La reutilización y la adaptación de diseños, permitiendo así innovar y mejorar los diseños de forma colaborativa a nivel mundial.
- Ayudaría a las compañías a ahorrar costes y tiempos de diseño en sus trabajos.
- Existen comunidades de diseño, programación, pruebas y soporte que día a día crecen de forma dinámica y participativa.
- Evita la alianza Trusted Computing y la gestión digital de los derechos (DRM), que imponen restricciones a los dispositivos electrónicos, como por ejemplo electrodomésticos y computadoras, entre otras más.

DESVENTAJAS

No se pueden aplicar directamente las cuatro libertades del software libre al hardware, dada su naturaleza diferente. Uno tiene existencia física, el otro no. Esto hace que surja una serie de problemas:

- Un diseño físico es único. La compartición depende de la facilidad de reproducción que éste posea.
- La compartición tiene asociado un coste. La persona que quiera utilizar el hardware que otra haya diseñado primero lo tiene que fabricar, para lo cual tendrá que comprar los componentes necesarios, construir el diseño y verificar que se ha hecho correctamente. Todo esto tiene un coste.
- Disponibilidad de los componentes. ¿Están disponibles los chips? Al intentar fabricar un diseño nos podemos encontrar con el problema de la falta de material. En un país puede no haber problema, pero en otro puede que no se encuentren.
- El mundo del hardware está plagado de patentes; es una realidad, por eso muchas de las motivaciones de los autores de este concepto son que no se libere el código o el diseño si no se desea, pero se anima a que

nuevas empresas desarrollen y liberen nuevo hardware, en pro de crear estándares públicos y libres en los cuales todos puedan colaborar.

- Modelo de producción; no cualquiera podrá realizar hardware, debido a las implicaciones que conlleva toda la infraestructura de diseño, simulación, producción e implementación del hardware, al contrario de lo que se da en el software libre.³

1.4. PLAN DE TRABAJO

ELECCIÓN DEL MICROCONTROLADOR ADECUADO

Elegir entre los diferentes microcontroladores que se adaptan a nuestros requerimientos: gran número de pines de E/S, suficiente memoria para gran cantidad de operaciones, facilidad de programación, etc...

FAMILIARIZACIÓN CON EL SOFTWARE Y EL LENGUAJE DE PROGRAMACIÓN

Familiarizarse con el software elegido, además de comprender todas las opciones que nos ofrece el lenguaje de programación del dispositivo.

DISEÑO MECÁNICO

Diseño de la estructura y las partes mecánicas del robot móvil.

CONTROL DE MOTORES

Estudio y control de los motores de CC del robot móvil

ELECCIÓN DE DISPOSITIVOS DE E/S

Estudio y control de los diferentes dispositivos del robot móvil: sensores, servo-motores, LCD, lector SD, reloj real y baterías.

PROGRAMACIÓN Y MONTAJE

Programación conjunta de todos los componentes del robot, además de su ubicación física en el mismo.

PRUEBAS DE FUNCIONAMIENTO

Pruebas finales del funcionamiento del robot.

³ OpenSource, OpenSoftware, OpenHardware - Wikipedia

1.5. OBJETIVOS

El primer objetivo del proyecto es el estudio y comprensión del funcionamiento de la robótica, así como de las partes que componen ésta: control de motores, sensores, dispositivos de salida, baterías, etc...

El segundo objetivo es el desarrollo de un robot móvil que vaya recorriendo el suelo de una habitación. El robot deberá evitar todos sus obstáculos además de mostrar/grabar toda la información que vaya obteniendo, incluyendo una matriz en 2D de la superficie de la habitación, en la que se representarían los obstáculos esquivados y los muros de ésta.

2. DESCRIPCIÓN DEL SISTEMA

2.1. DESCRIPCIÓN GENERAL DEL PROYECTO

Nuestra idea principal está basada en el estudio y comprensión del funcionamiento de la robótica, desde la elección de un microprocesador adecuado hasta el control de los diferentes dispositivos de E/S, pasando por el diseño mecánico.

A partir de aquí trataremos de desarrollar un robot móvil que vaya recorriendo el suelo de una habitación. Éste deberá evitar todos los obstáculos además de mostrar/grabar toda la información que vaya obteniendo, incluyendo una matriz en 2D de la superficie de la habitación.

Haremos un especial énfasis en los problemas que nos surgirán durante el proyecto, puesto que a partir de estos trataremos de encontrar soluciones y ver cuáles de éstas son viables o no.

2.2. DESCRIPCIÓN DEL HARDWARE

Lo primero que hemos tenido que hacer ha sido decidir entre las diferentes opciones de OpenHardware que se nos ofrecían. Hay una amplia gama de productos, como Arduino, Raspberry, Tensy, Seeduino Series, STM32 Discovery, Econotag, SolderCore, cada uno con sus diferentes microprocesadores y características.

Necesitábamos una placa con multitud de entradas/salidas para poder conectar una gran cantidad de dispositivos, además de un microprocesador que pudiera soportar una gran cantidad de operaciones. También para nosotros era importante tener conocimiento del entorno de programación, sin olvidarnos del coste de la placa.

Puesto que ya habíamos trabajado con Arduino y conocíamos su entorno de programación, decidimos buscar una placa en su catálogo que se adaptara a nuestras necesidades.

2.2.1. ARDUINO

2.2.1.1 ¿QUÉ ES ARDUINO?

Arduino es una plataforma de electrónica abierta para la creación de prototipos basada en software y hardware flexibles y fáciles de usar. Se creó para artistas, diseñadores, aficionados y cualquiera interesado en crear entornos u objetos interactivos (Ver 9.1 Enlaces).

Arduino puede tomar información del entorno, a través de sus pines de entrada, de toda una gama de sensores y puede afectar aquello que le rodea controlando

luces, motores y otros actuadores. Los proyectos hechos con Arduino pueden ejecutarse sin necesidad de conectar a un ordenador, si bien tienen la posibilidad de hacerlo y comunicar con diferentes tipos de software (p.ej. Flash, Processing, MaxMSP). La placa puede montarse por uno mismo o comprarla ya lista para usar, el software de desarrollo es abierto y se puede descargar gratuitamente.

El lenguaje de programación de Arduino es una implementación de Wiring, una plataforma de computación física parecida, que a su vez se basa en Processing, un entorno de programación multimedia.

2.2.1.2. LENGUAJE DE PROGRAMACIÓN

Los programas hechos con Arduino se dividen en tres partes principales: *estructura*, *valores* (variables y constantes), y *funciones*. El Lenguaje de programación Arduino se basa en C/C++.

WIRING

Wiring es un marco de programación de código abierto para microcontroladores.

Wiring permite escribir software multiplataforma para controlar los dispositivos conectados a una amplia gama de placas microcontroladoras para crear todo tipo de códigos creativos, objetos interactivos, espacios o experiencias físicas. El marco está cuidadosamente creado por diseñadores y artistas para fomentar una comunidad donde los principiantes y expertos de todo el mundo compartan las ideas, el conocimiento y la experiencia colectiva. Hay miles de estudiantes, artistas, diseñadores, investigadores y aficionados que utilizan Wiring para el aprendizaje, creación de prototipos y terminar en la producción profesional.

- » Hoja de ruta incluye soporte para múltiples arquitecturas de hardware "Cores "
- » El actual AVR8 Core soporta el hardware de Wiring y cualquier hardware basado en los procesadores AVR Xmega, AVR Tiny, TI MSP430, Microchip PIC24/32 Series y STM M3 ARM Cores .
- » Descarga gratuita, código y hardware abierto (Ver en 9.1 Enlaces).
- » Para GNU / Linux, Mac OS X y Windows

PROCESSING

Processing es un lenguaje de programación de código abierto y ambientado para las personas que desean crear imágenes, animaciones e interacciones. Inicialmente desarrollado para servir como un cuaderno de bocetos de software y enseñar los fundamentos de la programación de computadoras dentro de un contexto visual, Processing también se ha convertido en una herramienta profesional de trabajo. En la actualidad, hay decenas de miles de estudiantes, artistas, diseñadores, investigadores y aficionados que utilizan Processing para el aprendizaje, creación de prototipos y producción.

Características:

- » Descarga gratuita y de código abierto (Ver en 9.1 enlaces)
- » Programas interactivos en 2D, 3D o PDF
- » OpenGL integrado para aceleración 3D
- » Para GNU / Linux, Mac OS X y Windows
- » Los Proyectos se pueden ejecutar en línea o como aplicaciones

2.2.1.3. ¿POR QUÉ ARDUINO?

Hay muchos otros microcontroladores y plataformas con microcontroladores disponibles para la computación física: Parallax Basic Stamp, BX-24 de Netmedia, Phidgets, Handyboard del MIT, y otros ofrecen funcionalidades similares. Todas estas herramientas organizan el complicado trabajo de programar un microcontrolador en paquetes fáciles de usar. Arduino, además de simplificar el proceso de trabajar con microcontroladores, ofrece algunas ventajas respecto a otros sistemas a profesores, estudiantes y principiantes:

ASEQUIBLE

Las placas Arduino son más asequibles comparadas con otras plataformas de microcontroladores. La versión más cara de un módulo de Arduino puede ser montada a mano, e incluso ya montada cuesta bastante menos de 60€.

MULTI-PLATAFORMA

El software de Arduino funciona en los sistemas operativos Windows, Macintosh OSX y Linux. La mayoría de los entornos para microcontroladores está limitada a Windows.

ENTORNO DE PROGRAMACIÓN SIMPLE Y DIRECTO

El entorno de programación de Arduino es fácil de usar para principiantes y lo suficientemente flexible para los usuarios avanzados. Pensando en los profesores, Arduino está basado en el entorno de programación de Processing con lo que el estudiante que aprenda a programar en este entorno se sentirá familiarizado con el entorno de desarrollo de Arduino.

SOFTWARE AMPLIABLE Y DE CÓDIGO ABIERTO

El software Arduino está publicado bajo una licencia libre y preparado para ser ampliado por programadores experimentados. El lenguaje puede ampliarse a través de librerías de C++, y si se está interesado en profundizar en los detalles técnicos, se puede dar el salto a la programación en el lenguaje AVR C en el que está basado. De igual modo se puede añadir directamente código en AVR C en tus programas si así lo deseas.

HARDWARE AMPLIABLE Y DE CÓDIGO ABIERTO

Arduino está basado en los microcontroladores ATMEGA168, ATMEGA328, ATMEGA1280 y ATMEGA2560. Los planos de los módulos están publicados bajo licencia Creative Commons, por lo que diseñadores de circuitos con experiencia pueden hacer su propia versión del módulo, ampliándolo u optimizándolo. Incluso usuarios relativamente inexpertos pueden construir la versión para placa de desarrollo, para entender cómo funciona y ahorrar algo de dinero.

2.2.1.4. HARDWARE

Hay multitud de diferentes versiones de placas Arduino. La actual placa básica, el Duemilanove, usa Atmel ATmega32. La anterior Diecimila y las primeras unidades de Duemilanove usaban el Atmel ATmega168, mientras que las placas más antiguas usan el ATmega8. El Arduino Mega está basado en el ATmega1280 y

ATmega2560.

Los diseños y esquemas de Arduino se distribuyen bajo licencia Creative Commons Attribution-ShareAlike 2.5.

También es posible crear tu propia placa Arduino, para más información para crear tu propio hardware (Ver 9.1 enlaces).

2.2.1.4.1. PLACAS DE E/S

..... DUEMILANOVE

Ésta es la última revisión de la placa Arduino USB básica. Se conecta al ordenador con un cable USB estándar y contiene todo lo necesario para programar la placa. Se puede ampliar con gran variedad de shields: placas de extensión con funcionalidades específicas.

..... DIECIMILA

Ésta es la revisión anterior de la placa USB básica.

..... NANO

Es una placa compacta diseñada para usar directamente en placas de desarrollo. El Nano se conecta al ordenador con un cable Mini-B USB.

..... MEGA

Es la más grande y potente placa Arduino, compatible con los shields de Duemilanove y Diecimila.

..... BLUETOOTH

El Arduino BT contiene un módulo bluetooth que permite comunicarse y programarse sin cables. Es compatible con los shields de Arduino.

..... LILYPAD

Diseñado para aplicaciones sobre prendas, esta placa puede ser cosida a la ropa y es de color púrpura y con un diseño con estilo.

FIO

Está diseñada para aplicaciones inalámbricas. Incluye un zócalo para XBee, un conector para baterías LiPo y electrónica para cargar baterías.

MINI

Es la placa Arduino más pequeña. Funciona perfectamente en una placa de desarrollo o en aplicaciones donde el espacio es primordial. Se conecta al ordenador usando el adaptador Mini USB.

ADAPTADOR MINI USB

Esta placa convierte una conexión USB en 5 voltios, tiene toma de tierra y líneas TX y RX que puedes conectar al Arduino Mini o a otro micro controlador.

PRO

Esta placa está diseñada para aquellos que quieren dejar la placa incrustada en el proyecto: es más barata que la Diecimila y se puede alimentar fácilmente con baterías. Pero requiere de componentes extras y de montaje.

PRO MINI

Como la Pro, la Pro Mini está diseñada para usuarios avanzados que requieren de bajo coste, menor tamaño y estén dispuestos a un poco de trabajo extra.

SERIAL

Es una placa básica que utiliza una interfaz RS232 como comunicación con el ordenador para programar o intercambiar datos. El montaje de esta placa es fácil incluso como ejercicio didáctico.

SERIAL A UNA CARA (SINGLE SIDED)

Esta placa está diseñada para ser trazada y montada a mano. Es un poco más grande que la Diecimila, pero compatible con los shields.

2.2.1.4.2. SHIELDS

Los Shields son dispositivos que se colocan encima de la placa Arduino y que amplían las funciones que serán controladas desde Arduino, para manejar diferentes aparatos, adquirir datos, etc.

SHIELD XBEE

Este shield permite conectar inalámbricamente varios Arduino a distancias de 30 metros en locales interiores de edificios y de 90 metros en el exterior usando el módulo Maxstream Xbee Zigbee.

SHIELD MOTORES

Este shield permite a Arduino controlar motores eléctricos de corriente continua, servos, motores paso a paso y leer encoders.

SHIELD ETHERNET

Este shield permite a una placa Arduino conectarse a una red Ethernet y tener acceso a y desde Internet.

2.2.1.5. ARDUINO MEGA

Es la elección de placa para nuestro prototipo, puesto que tiene una gran cantidad de pines de E/S, además de ser más potente y tener más memoria que las otras versiones de Arduino más pequeñas (Ver Enlaces 9.1).

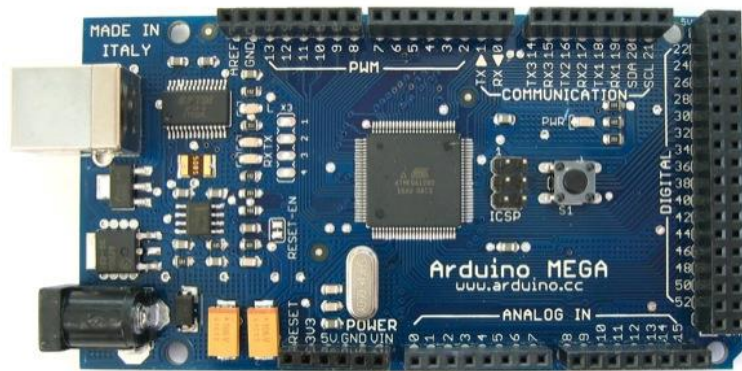


FIGURA 7: ARDUINO MEGA

2.2.1.5.1. VISIÓN GENERAL

El Arduino Mega es una placa microcontroladora basada en ATmega1280. Tiene 54 entradas/salidas digitales (de las cuales 14 proporcionan salida PWM), 16 entradas digitales, 4 UARTS (puertos serie por hardware), un cristal oscilador de 16MHz, conexión USB, entrada de corriente, conector ICSP y botón de reset. Contiene todo lo necesario para hacer funcionar el microcontrolador; se puede conectar al ordenador con un cable USB o alimentándolo con un transformador o batería para empezar. El Mega es compatible con la mayoría de shields diseñados para el Arduino Duemilanove o Diecimila.

2.2.1.5.2. ESQUEMAS

Ver 9.3 Esquema de pines del Arduino (Contenido del CD).

2.2.1.5.3. PRINCIPALES CARACTERÍSTICAS

Microcontrolador	ATmega1280
Voltaje de funcionamiento	5V
Voltaje de entrada (recomendado)	7-12V
Voltaje de entrada (limite)	6-20V
Pines E/S digitales	54 (14 proporcionan salida PWM)
Pines de entrada analógica	16
Intensidad por pin	40 mA
Intensidad en pin 3.3V	50 mA
Memoria Flash	128 KB de las cuales 4 KB las usa el gestor de arranque (bootloader)
SRAM	8 KB
EEPROM	4 KB
Velocidad de reloj	16 MHz

TABLA 2: CARACTERÍSTICAS DEL ARDUINO MEGA

2.2.1.5.4. ALIMENTACIÓN

El Arduino Mega puede ser alimentado vía USB o con una fuente de alimentación externa. El origen de la alimentación se selecciona automáticamente.

Las fuentes de alimentación externas (no-USB) pueden ser tanto un transformador como una batería. El transformador se puede conectar usando un conector macho de 2.1mm con centro positivo en el conector hembra de la placa. Los cables de la batería pueden conectarse a los pines Gnd y Vin en los conectores de alimentación (POWER)

La placa puede trabajar con una alimentación externa de entre 6 a 20 voltios. Si el voltaje suministrado es inferior a 7V el pin de 5V puede proporcionar menos de 5 Voltios y la placa puede volverse inestable; si se usan más de 12V los reguladores de voltaje se pueden sobrecalentar y dañar la placa. El rango recomendado es de 7 a 12 voltios.

Los pines de alimentación son los siguientes:

VIN

La entrada de voltaje a la placa Arduino cuando se está usando una fuente externa de alimentación (en opuesto a los 5 voltios de la conexión USB). Se puede proporcionar voltaje a través de este pin o, si se está alimentado a través de la conexión de 2.1mm, acceder a ella a través de este pin.

5V

La fuente de voltaje estabilizado usado para alimentar el microcontrolador y otros componentes de la placa. Ésta puede provenir de VIN a través de un regulador integrado en la placa, o proporcionada directamente por el USB u otra fuente estabilizada de 5V.

3V3

Una fuente de voltaje a 3.3 voltios generada en el chip FTDI integrado en la placa. La corriente máxima soportada 50mA.

GND

Pines de toma de tierra.

2.2.1.5.5. MEMORIA

El ATmega1280 tiene 128KB de memoria flash para almacenar código (4KB son usados para el arranque del sistema (bootloader). El ATmega1280 tiene 8 KB de memoria SRAM. El ATmega1280 tiene 4KB de EEPROM, a la cual se puede acceder para leer o escribir con la [Reference/EEPROM |librería EEPROM]].

2.2.1.5.6. ENTRADAS Y SALIDAS

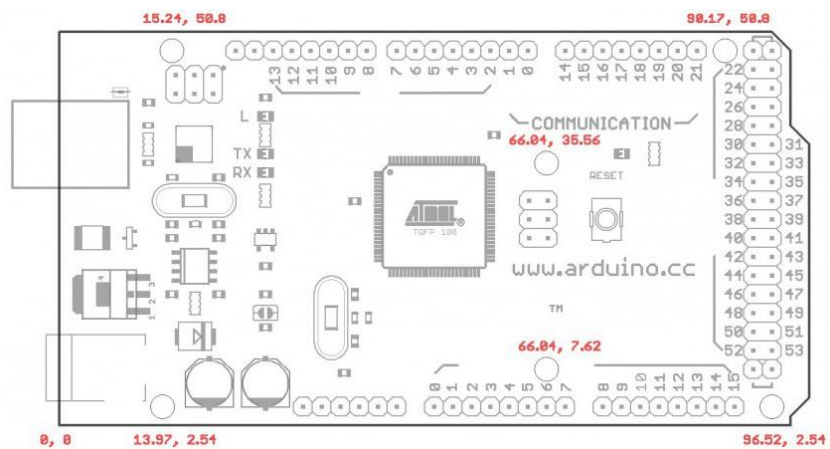


FIGURA 8: PINES ARDUINO MEGA

Cada uno de los 54 pines digitales en el Arduino Mega puede utilizarse como entrada o como salida usando las funciones `pinMode()`, `digitalWrite()`, y `digitalRead()`. Las E/S operan a 5 voltios. Cada pin puede proporcionar o recibir una intensidad máxima de 40mA y tiene una resistencia interna (desconectada por defecto) de 20-50kOhms. Además, algunos pines tienen funciones especializadas:

SERIE: 0 (RX) Y 1 (TX); SERIE 1: 19 (RX) Y 18 (TX); SERIE 2: 17 (RX) Y 16 (TX); SERIE 3: 15 (RX) Y 14 (TX).

Usado para recibir (RX) transmitir (TX) datos a través de puerto serie TTL. Los pines Serie 0 (RX) y 1 (TX) están conectados a los pines correspondientes del chip FTDI USB-to-TTL.

INTERRUPCIONES EXTERNAS: 2 (INTERRUPCIÓN 0), 3 (INTERRUPCIÓN 1), 18 (INTERRUPCIÓN 5), 19 (INTERRUPCIÓN 4), 20 (INTERRUPCIÓN 3), Y 21 (INTERRUPCIÓN 2).

Estos pines se pueden configurar para lanzar una interrupción en un

valor LOW(0V), en flancos de subida o bajada (cambio de LOW a HIGH(5V) o viceversa), o en cambios de valor. Véase la función [*attachInterrupt\(\)*](#) para más detalles.

PWM: DE 0 A 13

Proporciona una salida PWM (Pulse Wave Modulation, modulación de onda por pulsos) de 8 bits de resolución (valores de 0 a 255) a través de la función [*analogWrite\(\)*](#).

SPI: 50 (SS), 51 (MOSI), 52 (MISO), 53 (SCK).

Estos pines proporcionan comunicación SPI, que a pesar de que el hardware la proporcione, actualmente no está incluida en el lenguaje Arduino.

LED: 13

Hay un LED integrado en la placa conectado al pin digital 13. Cuando este pin tiene un valor HIGH(5V) el LED se enciende y cuando este tiene un valor LOW(0V) este se apaga.

El Mega tiene 16 entradas analógicas, y cada una de ellas proporciona una resolución de 10bits (1024 valores). Por defecto se mide de tierra a 5 voltios, aunque es posible cambiar la cota superior de este rango usando el pin AREF y la función [*analogReference\(\)*](#). Además algunos pines tienen funciones especializadas:

I2C: 20 (SDA) Y 21 (SCL)

Soporte del protocolo de comunicaciones I²C (TWI) usando la [*librería Wire*](#).

AREF

Voltaje de referencia para las entradas analógicas. Es usado por [*analogReference\(\)*](#).

RESET

Suministra un valor LOW(0V) para reiniciar el microcontrolador. Es típicamente usado para añadir un botón de reset a los shields que no dejan acceso a este botón en la placa.

2.2.1.5.7. COMUNICACIONES

El Arduino Mega facilita en varios aspectos la comunicación con el ordenador, otro Arduino u otros microcontroladores. El ATmega1280 proporciona cuatro puertos de comunicación vía serie UART TTL (5V). Un chip FTDI FT232RL integrado en la placa canaliza esta comunicación serie a través del USB, y los drivers FTDI (incluidos en el software de Arduino)

proporcionan un puerto serie virtual en el ordenador. El software incluye un monitor de puerto serie que permite enviar y recibir información textual de la placa Arduino. Los LEDs RX y TX de la placa parpadearán cuando se detecte comunicación transmitida a través del chip FTDI y la conexión USB (no parpadearán si se usa la comunicación serie a través de los pines 0 y 1).

La librería *SoftwareSerial* permite comunicación serie por cualquier par de pines digitales del Arduino Mega.

El ATmega1280 también soporta la comunicación I2C (TWI) y SPI. El software de Arduino incluye una librería *Wire* para simplificar el uso del bus I2C. Para el uso de la comunicación SPI, mírese en la hoja de especificaciones (datasheet) del ATmega1280 que adjuntamos en el contenido del CD.

2.2.1.5.8. PROGRAMACIÓN

El Arduino Mega se puede programar con el software Arduino.

El ATmega1280 en el Arduino Mega viene precargado con *un gestor de arranque (bootloader)* que permite cargar nuevo código sin necesidad de un programador por hardware externo. Se comunica utilizando el protocolo STK500 original.

También te puedes saltar el gestor de arranque y programar directamente el microcontrolador a través del puerto ISCP (In Circuit Serial Programming).

2.2.1.5.9. REINICIO AUTOMÁTICO POR SOFTWARE

En vez de necesitarse reiniciar presionando físicamente el botón de reset antes de la carga, el Arduino Mega está diseñado de manera que es posible reiniciar por software desde el ordenador donde esté conectado. Una de las líneas de control de flujo (DTR) del FT232RL está conectada a la línea de reinicio del ATmega1280 a través de un condensador de 100 nanofaradios. Cuando la línea se pone a LOW (0V), la línea de reinicio también se pone a LOW el tiempo suficiente para reiniciar el chip. El software de Arduino utiliza esta característica para permitir cargar los sketches con sólo apretar un botón del entorno. Dado que el gestor de arranque tiene un lapso de tiempo para ello, la activación del DTR y la carga del sketch se coordinan perfectamente.

Esta configuración tiene otras implicaciones. Cuando el Mega se conecta

a un ordenador con Mac OS X o Linux, esto reinicia la placa cada vez que se realiza una conexión desde el software (vía USB). Durante aproximadamente medio segundo posterior, el gestor de arranque se está ejecutando. A pesar de estar programado para ignorar datos mal formateados por ejemplo, cualquier cosa como la carga de un programa nuevo, intercepta los primeros bytes que se envían a la placa justo después de que se abra la conexión. Si un sketch ejecutándose en la placa recibe algún tipo de configuración inicial u otro tipo de información al inicio del programa, asegúrate de que el software con el cual se comunica espera un segundo después de abrir la conexión antes de enviar los datos.

El Mega contiene una pista que puede ser cortada para deshabilitar el auto-reset. Las terminaciones a cada lado pueden ser soldadas entre ellas para rehabilitarlo. Están etiquetadas con "RESET-EN". También podéis deshabilitar el auto-reset conectando una resistencia de 110 ohms desde el pin 5V al pin de reset.

2.2.1.5.10. PROTECCIÓN CONTRA SOBRETENSIONES EN USB

El Arduino Mega tiene un multifusible reinicializable que protege la conexión USB de tu ordenador de cortocircuitos y sobretensiones. Aparte de que la mayoría de ordenadores proporcionan su propia protección interna, el fusible proporciona una capa extra de protección. Si más de 500mA son detectados en el puerto USB, el fusible automáticamente corta la conexión hasta que el cortocircuito o la sobretensión desaparezca.

2.2.1.5.11. CARACTERÍSTICAS FÍSICAS Y COMPATIBILIDAD DE SHIELDS

La longitud y amplitud máxima de la placa Duemilanove es de 10 y 5.3 cm respectivamente, sobresaliendo de estas dimensiones, el conector USB y la conexión de alimentación. Tres agujeros para fijación con tornillos permiten colocar la placa en superficies y cajas. Ten en cuenta que la distancia entre los pines digitales 7 y 8 es de 160 mil (0,16"), no es múltiple de la separación de 100 mil entre los otros pines.

El Arduino Mega está diseñado para ser compatible con la mayoría de shields diseñados para el Diecimila o Duemilanove. Los pines digitales de 0 a 23 (y los pines AREF y GND adyacentes), las entradas analógicas de 0 a 5, los conectores de alimentación y los conectores ICPS están todos ubicados en posiciones equivalentes. Además, el puerto serie principal está ubicado en los mismos pines (0 y 1), así como las interrupciones 0 y

1 (pines 2 y 3 respectivamente).

SPI está disponible en los conectores ICSP tanto en el Arduino Mega como en el Duemilanove/Diecimila.⁴

2.2.2. MOTOR DE CORRIENTE CONTINUA

El motor de corriente continua es una máquina que convierte la energía eléctrica continua en mecánica, provocando un movimiento rotatorio.

Esta máquina de corriente continua es una de las más versátiles en la industria. Su fácil control de posición, par y velocidad la han convertido en una de las mejores opciones en aplicaciones de control y automatización de procesos. Pero con la llegada de la electrónica su uso ha disminuido en gran medida, pues los motores de corriente alterna, del tipo asíncrono, pueden ser controlados de igual forma a precios más accesibles para el consumidor medio de la industria. A pesar de esto los motores de corriente continua se siguen utilizando en muchas aplicaciones de potencia (trenes y tranvías) o de precisión (máquinas, micromotores, robots, etc.)

La principal característica del motor de corriente continua es la posibilidad de regular la velocidad desde vacío a plena carga. Su principal inconveniente, el mantenimiento, muy caro y laborioso.

2.2.2.1. PARTES DE UN MOTOR CC

Un motor CC está compuesto de un estator y un rotor. En muchos motores c.c., generalmente los más pequeños, el estator está compuesto de imanes para crear un campo magnético.

En motores de corriente continua más grandes este campo magnético se logra con devanados de excitación de campo.

El rotor es el dispositivo que gira en el centro del motor cc y está compuesto de arrollados de cable conductores de corriente continua. Esta corriente continua es suministrada al rotor por medio de las "escobillas", generalmente fabricadas de carbón.

2.2.2.2. PRINCIPIO BÁSICO DE FUNCIONAMIENTO DE UN MOTOR CC

Cuando un conductor por el que fluye una corriente continua es colocado bajo la influencia de un campo magnético, se induce sobre él (el conductor) una

⁴ Arduino Website

fuerza que es perpendicular tanto a las líneas de campo magnético como al sentido del flujo de la corriente.

Para controlar el sentido del flujo de la corriente en los conductores, se usa un conmutador que realiza la inversión del sentido de la corriente cuando el conductor pasa por la línea muerta del campo magnético.

La fuerza con la que el motor gira (el par motor) es proporcional a la corriente que hay por los conductores. A mayor tensión, mayor corriente y mayor par motor.

2.2.3. SERVOMOTOR

2.2.3.1. DEFINICIÓN

Un servomotor (o servo) es un motor de corriente continua que tiene la capacidad de ser controlado en posición. Es capaz de ubicarse en cualquier posición dentro de un rango de operación (generalmente de 180º) y mantenerse estable en dicha posición. Los servos se suelen utilizar en robótica, automática y modelismo (vehículos por radio-control, RC) debido a su gran precisión en el posicionamiento.



FIGURA 9: DESPIECE DE UN SERVOMOTOR

2.2.3.2. ESTRUCTURA

En general, los servos suelen estar compuestos por 4 elementos fundamentales:

MOTOR DE CORRIENTE CONTINUA (CC)

Es el elemento que le brinda movilidad al servo. Cuando se aplica un potencial a sus dos terminales, este motor gira en un sentido a su velocidad máxima. Si el voltaje aplicado a sus dos terminales es inverso, el sentido de giro también se invierte.

ENGRANAJES REDUCTORES

Tren de engranajes que se encarga de reducir la alta velocidad de giro del motor para acrecentar su capacidad de torque (o par-motor).

SENSOR DE DESPLAZAMIENTO

Suele ser un potenciómetro colocado en el eje de salida del servo que se utiliza para conocer la posición angular del motor.

CIRCUITO DE CONTROL

Es una placa electrónica que implementa una estrategia de control de la posición por realimentación. Para ello, este circuito compara la señal de entrada de referencia (posición deseada) con la posición actual medida por el potenciómetro. La diferencia entre la posición actual y la deseada es amplificada y utilizada para mover el motor en la dirección necesaria para reducir el error.

Además se pueden diferenciar más partes, que se pueden observar en la siguiente imagen:

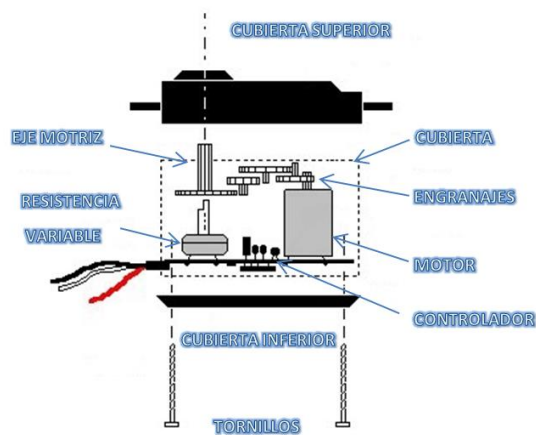


FIGURA 10: PARTES DE UN SERVOMOTOR

2.2.3.3. FUNCIONAMIENTO

Los servos disponen de tres cables: dos cables de alimentación (positivo y negativo/masa) que suministran un voltaje 4.8-6V y un cable de control que indica la posición deseada al circuito de control mediante señales PWM ("Pulse Width Modulation").

Colores de los cables de los principales fabricantes de servos:

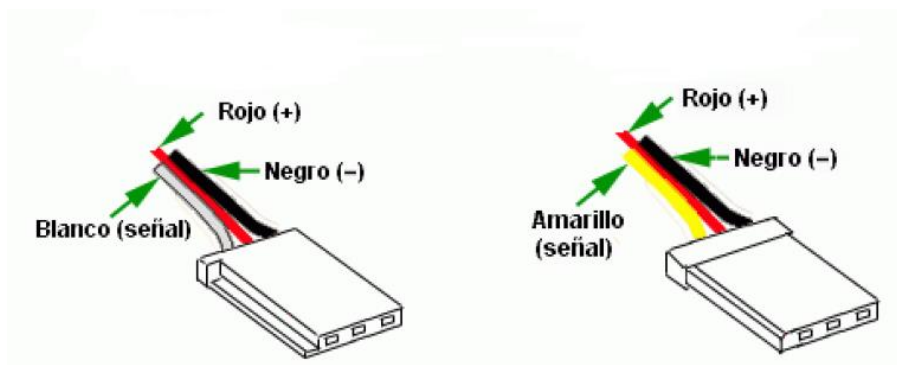


FIGURA 11: COLORES DE LOS CABLES DE LOS PRINCIPALES FABRICANTES DE SERVOMOTORES

Las señales PWM utilizadas para controlar los servos están formadas por pulsos positivos cuya duración es proporcional a la posición deseada del servo y que se repiten cada 20ms (50Hz). Todos los servos pueden funcionar correctamente en un rango de movimiento de 90°, que se corresponde con pulsos PWM comprendidos entre 0.9 y 2.1ms. Sin embargo, también existen servos que se pueden mover en un rango extendido de 180° y sus pulsos de control varían entre 0.5 y 2.5ms (Figura 12). Antes de utilizar un servo habrá que comprobar experimentalmente su rango de movimiento para no dañarlo. Para mantener fijo un servo en una posición habrá que enviar periódicamente el pulso correspondiente; ya que si no recibe señales, el eje del servo quedará libre y se podrá mover ejerciendo una leve presión.

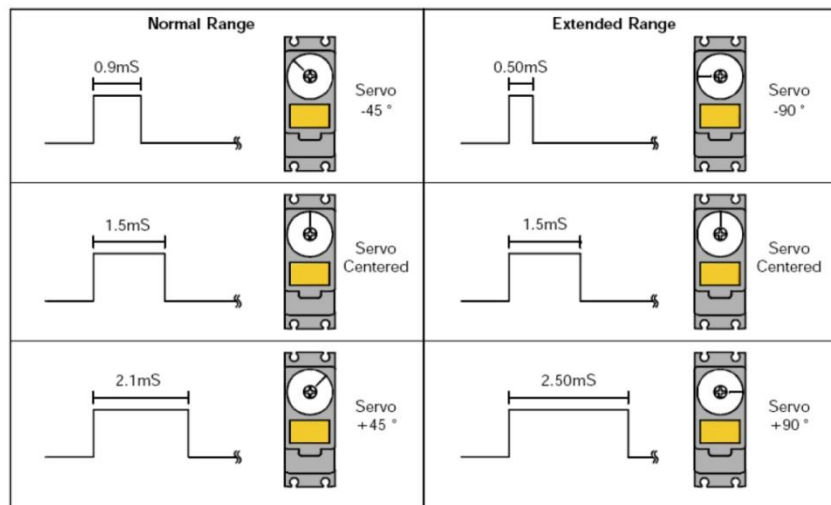


FIGURA 12: PWM Y RANGO DE MOVIMIENTO DE SERVOMOTORES

TIPOS

Existen dos tipos de servos: *analógicos* y *digitales*. Ambos tipos de servos son iguales a nivel de usuario: tienen la misma estructura (motor DC, engranajes reductores, potenciómetro y placa de control) y se controlan con las mismas señales PWM. La principal diferencia entre ellos radica en la adición de un microprocesador en el circuito de control de los servos digitales. Este microprocesador se encarga de procesar la señal PWM de entrada y de controlar el motor mediante pulsos con una frecuencia 10 veces superior a los servos analógicos.

El aumento en la frecuencia de excitación del motor en los servos digitales permite disminuir su tiempo de respuesta (menor *deadband*), aumentar su resolución de movimiento y suavizar su aceleración/deceleración. El uso de un microprocesador permite también a los servos digitales programar distintos parámetros de configuración que son fijos en los analógicos: sentido de giro, posición central inicial, topes en el recorrido del servo, velocidad de respuesta del servo y resolución. Para establecer estos parámetros se deben utilizar aparatos específicos de cada marca. El principal inconveniente de los servos digitales es que consumen más energía que los analógicos al tener que generar más pulsos de control para el motor.

CONTROL DE SERVOS

Para controlar los servos se les debe enviar pulsos PWM a través del cable de control.

En los sistemas de modelismo, se utilizan dos componentes para controlar los servos: un receptor y una emisora. El receptor es el componente que se encarga de recibir los comandos de la emisora inalámbricamente y transformarlos en los pulsos PWM correspondientes que son enviados a los servos. La emisora es un mando que transmite las órdenes al receptor a través de señales inalámbricas con modulación AM, FM o PCM.

Sin embargo, estos sistemas no se pueden conectar a un PC y la única forma de controlar los servos es con el mando. Existen otros circuitos controladores especialmente diseñados para controlar servos a través del PC. Suelen utilizar la interfaz serie (RS-232) para recibir los comandos de control. Estos comandos suelen ser diferentes dependiendo del modelo.

CONTROLADORES SERIE DE SERVOS

Un SSC (*Serial Servo Controller*, Controlador Serie de Servos) es un dispositivo utilizado para controlar servos desde un PC a través del puerto serie. Los SSC aceptan comandos con un determinado formato desde el puerto serie del PC y los transforman en pulsos PWM que son enviados a los servos que se desea controlar.

MINI SSC II

Mini SSC II es un controlador de la empresa *Scott Edwards Electronics Inc.* que puede controlar hasta un máximo de 8 servos. Se conecta al PC a través de un cable serie (con conector RJ11 en el extremo del controlador y conector DB9 en el extremo del PC) y puede comunicarse con el puerto serie a dos velocidades: 2400bps ó 9600bps. Este controlador requiere dos fuentes de alimentación: una de 9-12V para la circuitería y otra de 5V para los servos.

Las principales ventajas de este controlador son su reducido tamaño y la gran cantidad de software disponible debido a su amplia utilización en estos últimos años. Sus principales inconvenientes son la discontinuidad en su desarrollo (ya no se desarrollan nuevos modelos) y la necesidad de utilizar dos fuentes de alimentación distintas.

SSC-32

SSC-32 es un controlador de la empresa *Lynxmotion* que permite controlar hasta un máximo de 32 servos. Se trata de un controlador más completo que el Mini SSC, ya que dispone de un conjunto de funcionalidades adicionales: control de servos por tiempo/velocidad/posición, movimiento síncrono de varios servos, consulta de posición de los servos y utilización de los pines de control de los servos como salidas digitales TTL. Además, dispone de 4 entradas (A, B, C y D) que pueden ser leídas de manera digital (bits) o de manera analógica (voltajes).

Este controlador presenta un gran número de ventajas respecto al Mini SSC: control más complejo de los servos, disponibilidad de 4 entradas, diversas configuraciones de alimentación (fuente única para placa y servos o fuentes separadas), mayor rango de velocidades del puerto serie (2400bps, 9600bps, 38.4 Kbps y 115.2Kbps), posibilidad de conectar el SSC-32 a un micro-controlador a través de comunicación serie TTL... Su único inconveniente es su mayor tamaño al disponer de un conector DB9 estándar.

ARDUINO

Este dispositivo no es un controlador específico para servos, como los dos anteriores, sino que es una placa de I/O que permite el control de servos.

Esta placa presenta ciertos inconvenientes para el control de servos al no ser un controlador específico. No permite opciones de control tan complejas sobre los servos como el SSC-32. Además, el cableado de los servos es más aparatoso, ya que hay que conectar el cable de control a una salida digital y los cables de alimentación a otros pines que se encuentran en el lado opuesto de la placa.⁵

2.2.4. SENSOR DE ULTRASONIDOS

2.2.4.1. QUÉ SON LOS ULTRASONIDOS

Se denomina ultrasonidos a las vibraciones de frecuencia superior a las audibles por el ser humano que se producen en un medio elástico. La frecuencia audible por el ser humano comienza en unos 16 Hz y tiene un límite superior de aproximadamente 20 KHz.

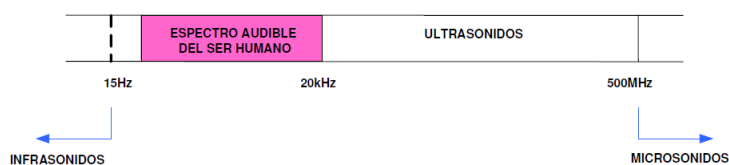


FIGURA 13: RANGO DE FRECUENCIAS

2.2.4.2. SENSORES DE DISTANCIA POR ULTRASONIDOS

La medida de diferentes variables físicas mediante los ultrasonidos están relacionadas normalmente con su velocidad, su tiempo de propagación y, en algunos casos, con la atenuación o interrupción del haz propagado.

Hay tres principios físicos en los que se basa el funcionamiento de los sensores de ultrasonidos:

- Propagación de los ultrasonidos en medios homogéneos y no homogéneos
- Reflexión de los ultrasonidos en objetos inmóviles o fijos.
- Efecto Doppler.

⁵ Servomotores – Francisco A. Candelas Herías. Juan A. Corrales Ramón.

PROPAGACIÓN EN MEDIOS HOMOGÉNEOS

Las perturbaciones sonoras que se producen en un punto de un medio elástico se propagan a través de él con una velocidad c , que depende de la densidad del medio ρ y de su módulo de elasticidad E , de acuerdo con la ecuación:

$$c = \sqrt{\frac{E}{\rho}}$$

Se define la impedancia acústica Z del medio como el producto de la densidad ρ por la velocidad del sonido c :

$$Z = \rho \cdot c$$

Como resultado de la perturbación, la presión varía con respecto a un valor medio, y la diferencia entre el valor instantáneo y el valor medio se denomina presión acústica p .

La intensidad I de la onda es la potencia por unidad de superficie en W/m^2 . La impedancia Z , la presión acústica p y la intensidad I están relacionadas mediante la ecuación:

$$I = \frac{p^2}{Z} = \frac{p^2}{\rho \cdot c}$$

Los ultrasonidos se pueden propagar a través de un medio homogéneo o no homogéneo.

Al propagarse la radiación en un medio homogéneo, su intensidad sufre una atenuación exponencial donde intervienen la intensidad incidente I_o , el coeficiente de atenuación α y la distancia recorrida x conforme la ecuación:

$$I = I_o \cdot e^{-2\alpha x}$$

Si la onda pasa de un medio de impedancia Z_1 a otro de impedancia Z_2 , además de ser absorbida, se refleja. Podemos diferenciar tres tipos diferentes de intensidad de radiación: incidente I_I , reflejada I_R y transmitida I_T . En este caso los coeficientes de reflexión y transmisión son:

$$R = \frac{I_R}{I_I} = \left(\frac{Z_1 - Z_2}{Z_1 + Z_2} \right)^2$$

$$T = \frac{I_T}{I_I} = \left(\frac{4Z_1 Z_2}{Z_1 + Z_2} \right)^2$$

$$R + T = 1$$

INFLUENCIA DE LOS FACTORES AMBIENTALES

La temperatura puede influir significativamente en la propagación de los ultrasonidos. La densidad del aire depende de la temperatura, que a su vez influye sobre la velocidad de propagación de la onda de acuerdo con la siguiente ecuación:

$$V_s = V_{so} \sqrt{1 + \frac{T}{273}}$$

Donde **V_{so}** es la velocidad de propagación de la onda sonora a 0 °C, y **T** es la temperatura absoluta (grados Kelvin).

2.2.4.3. GENERACIÓN DE ULTRASONIDOS

Para generar ultrasonidos se utilizan habitualmente materiales piezoeléctricos en los que se generan tensiones eléctricas al aplicarles una presión mecánica. El efecto piezoeléctrico se debe a que la presión provoca una deformación de la red cristalina que da lugar, a su vez, a un desplazamiento de las cargas eléctricas moleculares, lo que hace que aparezcan diferencias de potencial entre las caras del material. El signo de esta diferencia de potencial se invierte cuando se invierte el sentido de la presión, que puede ser de tracción o de compresión.

El cuarzo y la turmalina son materiales piezoeléctricos naturales, pero, debido al valor reducido de sus parámetros característicos, fueron sustituidos por otros materiales piezoeléctricos sintéticos implementados con titanatos y circonatos de plomo (PZV). Estos materiales son más estables que los naturales, aunque presentan el inconveniente de su dependencia de la temperatura y la pérdida de sus propiedades piezoeléctricas a medida que su temperatura se acerca al valor de Curie.

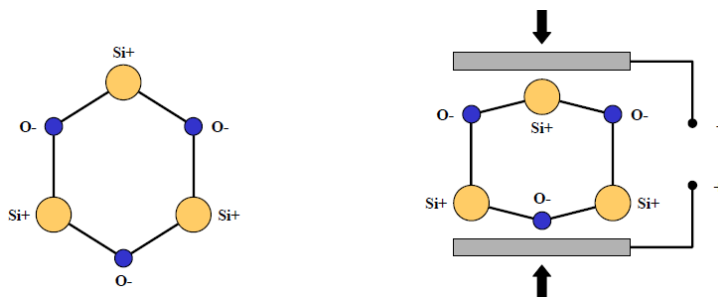


FIGURA 14: MATERIALES PIEZOELÉCTRICOS

Los sensores electrónicos de ultrasonidos se basan en la reversibilidad del principio de funcionamiento de los materiales piezoeléctricos.

Dichos materiales se caracterizan por generar una señal eléctrica al aplicarles una onda de presión (en este caso el sonido) y por ser capaces de vibrar y de generar ondas de presión cuando están inmersos en un medio elástico, como por ejemplo el aire, cualquier otro fluido o un sólido, al aplicarles una excitación eléctrica.

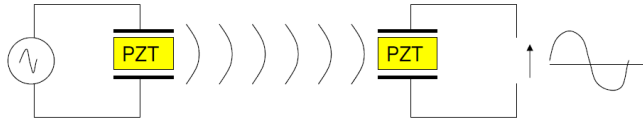


FIGURA 15: EFECTO PIEZOELÉCTRICO

En general se necesitan haces estrechos a fin de evitar reflexiones indeseadas, para lo cual la superficie de emisión debe ser grande con respecto a la longitud de onda del ultrasonido emitido.

En este caso, además del cono de radiación fundamental se genera un conjunto de lóbulos secundarios en forma de anillos.

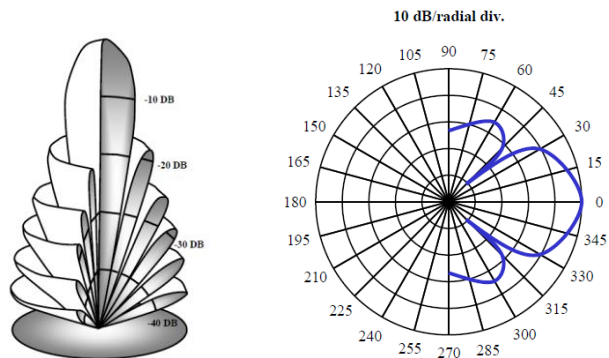


FIGURA 16: CONO DE RADIACIÓN FUNDAMENTAL

2.2.4.4. PRINCIPALES EFECTOS UTILIZADOS PARA IMPLEMENTAR SENSORES DE ULTRASONIDOS

LA INTERRUPCIÓN DEL HAZ ULTRASÓNICO

Funciona como detector de un objeto que interrumpe el haz ultrasónico

LA REFLEXIÓN EN OBJETOS INMÓVILES O FIJOS

Es debida a la propagación en medios no homogéneos, que no se mueven uno con respecto al otro, descrita anteriormente.

LA REFLEXIÓN EN OBJETOS MÓVILES (EFECTO DOPPLER)

Es debida a la propagación en medios no homogéneos que tienen un

movimiento relativo entre ellos

2.2.4.4.1. SENSOR ULTRASÓNICO DE REFLEXIÓN EN OBJETOS INMÓVILES O DE TIPO ECO

Se mide el tiempo que tarda en recibirse el eco de un impulso emitido, debido a la reflexión sobre un objeto presente en el camino de propagación de la radiación.

El objeto puede ser un líquido, un sólido, granular o polvo, con la única restricción de que debe tener una impedancia acústica muy diferente de la del medio en el que se propagan los ultrasonidos, para que la mayor parte de la radiación se refleje.

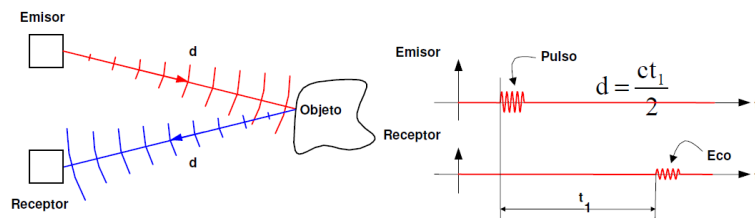


FIGURA 17: CÁLCULO DE LA DISTANCIA EN OBJETOS INMÓVILES

En los sensores de ultrasonidos de bajo coste se utiliza el mismo transductor como emisor y receptor. Tras la emisión del ultrasonido se espera un determinado tiempo a que las vibraciones en el sensor desaparezcan y a que esté preparado para recibir el eco producido por el obstáculo.

Esto implica que existe una distancia mínima d (proporcional al tiempo de relajación del transductor) a partir de la cual el sensor mide con precisión.

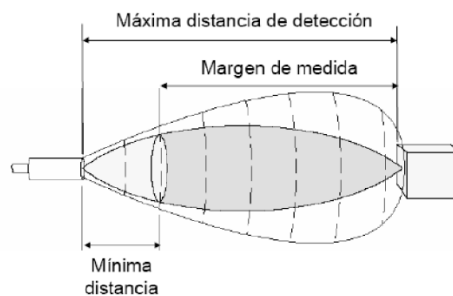


FIGURA 18: MÁRGENES DE MEDIDA DE LOS SENSORES DE ULTRASONIDOS

CONSIDERACIONES A TENER EN CUENTA

- Para lograr un gran alcance y reducir las interferencias acústicas es importante que el haz emitido sea estrecho.
- Si el tamaño del objeto es menor que la longitud de onda, la radiación reflejada es débil, por lo que para objetos pequeños se debe trabajar a alta frecuencia.
- El efecto de las interferencias acústicas es menor a altas frecuencias pero la atenuación de la onda es mayor.
- Si la temperatura ambiente no es constante hay que compensar la variación de la velocidad c de propagación (a menor temperatura mayor tiempo de recepción). La velocidad también depende de la presión, la densidad o la presencia de sustancias disueltas. Por ello, se suele calcular primero el valor de c mediante la reflexión en un objeto a una distancia conocida.
- Un factor de error muy común es el conocido como falsos ecos. Estos falsos ecos se pueden producir por razones diferentes: puede darse el caso en que la onda emitida por el transductor se refleje varias veces en diversas superficies antes de que vuelva a incidir en el transductor (si es que incide). Este fenómeno, conocido como reflexiones múltiples, implica que la lectura del sensor evidencia la presencia de un obstáculo a una distancia proporcional al tiempo transcurrido en el viaje de la onda; es decir, una distancia mucho mayor que a la que está en realidad el obstáculo más cercano, que pudo producir la primera reflexión de la onda.
- Otra fuente más común de falsos ecos, conocida como ***crosstalk***, se produce cuando se emplea un cinturón de ultrasonidos donde una serie de sensores están trabajando al mismo tiempo. En este caso puede ocurrir (y ocurre con una frecuencia relativamente alta) que un sensor emita un pulso y sea recibido por otro sensor que estuviese esperando el eco del pulso que él había enviado con anterioridad (o viceversa).
- Las ondas de ultrasonido obedecen a las leyes de reflexión de las ondas, por lo que una onda de ultrasonido tiene el mismo ángulo de incidencia y reflexión respecto a la normal a la superficie. Esto implica que si la orientación relativa de la superficie reflectora con respecto al eje del sensor de ultrasonido es mayor que un cierto umbral, el sensor nunca

reciba el pulso de sonido que emitió.

- Puede existir un problema de falsa detección cuando varios robots trabajan con ultrasonidos (si los sensores trabajan en la misma frecuencia).

APLICACIONES

SISTEMAS DE PERCEPCIÓN Y CONTROL DE ROBOTS MÓVILES

Los sensores de ultrasonidos se suelen utilizar en sistemas de reconocimiento del entorno y medición de distancias en robótica. En este caso se utilizan varios sensores y se diseña una estrategia de control en función de las mediciones de los mismos.

Los sensores se colocan de una manera determinada en la superficie exterior del robot y el resultado de las medidas se utiliza para establecer la estrategia de movimiento del mismo.

SISTEMA DE PREVENCIÓN DE COLISIONES

Hay un sistema de prevención de colisiones para vehículos de la industria de la construcción (grúas, camiones, topadoras, tractores, montacargas, etc.).

Es muy común durante el trabajo en obra que estos vehículos colisionen con el entorno debido a la gran cantidad de maniobras que deben realizar dentro de un ambiente en constante cambio y a la escasa visibilidad ocasionada por el propio vehículo.

El coste de reparación de estos vehículos y las demoras ocasionadas justifican la instalación de un sistema para la prevención de colisiones.

SONAR (SOUND NAVIGATION AND RANGING)

Utilizan la reflexión de los ultrasonidos en objetos situados en el agua. Se usó con fines militares en la Segunda Guerra Mundial.

Hoy en día su aplicación militar está en desuso, ya que la llegada del impulso alerta al objetivo de la presencia del emisor antes de que éste escuche el eco.

Actualmente se utiliza para detectar bancos de peces.

2.2.4.4.2. SENSOR ULTRASÓNICO DE REFLEXIÓN EN OBJETOS MÓVILES

EFFECTO DOPPLER

El efecto Doppler, llamado así por Christian Andreas Doppler en 1842, consiste en la variación de la frecuencia de cualquier tipo de onda emitida o recibida por un objeto en movimiento.

La frecuencia percibida por un observador (o) de una fuente (s) es:

$$f' = f \frac{V (+/-) V_0}{V (-/+) V_s}$$

V_0 es la velocidad del observador

V_s es la velocidad de la fuente

V es la velocidad de la onda en el medio

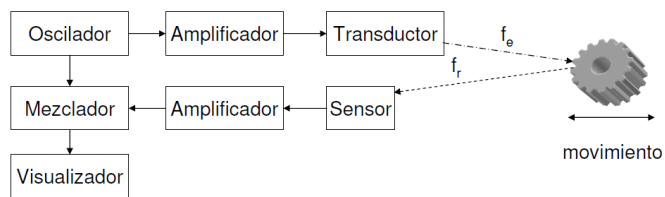


FIGURA 19: DIAGRAMA DE BLOQUES DE UN SENSOR DE ULTRASONIDOS BASADO EN EL EFECTO DOPPLER

APLICACIONES

CAUDALÍMETRO

Se basa en la diferencia entre el valor del tiempo que tardan dos ondas en recorrer el mismo camino en sentidos opuestos en el seno de un fluido en movimiento. Debido al efecto Doppler la velocidad del sonido se suma a la del fluido, lo que da como resultado valores diferentes

para los recorridos A-B y B-A. El emisor y el receptor se sitúan en las paredes opuestas de la tubería y forman un ángulo con el eje.⁶

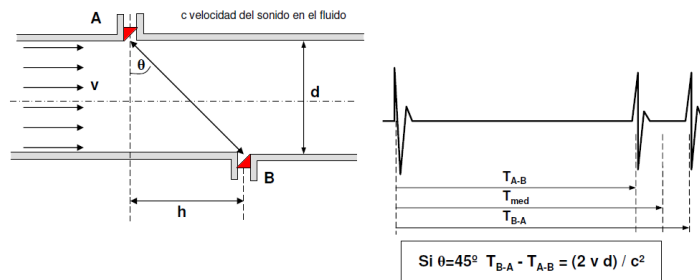


FIGURA 20: FUNCIONAMIENTO DE UN CAUDALÍMETRO

2.2.5. LCD

Una pantalla de cristal líquido o LCD (acrónimo del inglés *liquid crystal display*) es una pantalla delgada y plana formada por un número de píxeles en color o monocromos colocados delante de una fuente de luz o reflectora.

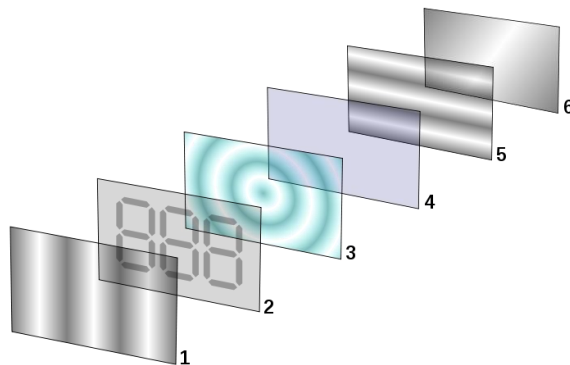


FIGURA 21: DESPIECE DE UNA PANTALLA LCD

1. Film de filtro vertical para polarizar la luz que entra.
2. Sustrato de vidrio con electrodos de Óxido de Indio ITO. Las formas de los electrodos determinan las formas negras que aparecen cuando la pantalla se enciende y apaga. Los cantos verticales de la superficie son suaves.
3. Cristales líquidos "Twisted Nematic" (TN).
4. Sustrato de vidrio con film electrodo común (ITO) con los cantos horizontales para alinearse con el filtro horizontal.
5. Film de filtro horizontal para bloquear/permitir el paso de luz.
6. Superficie reflectante para devolver la luz al espectador. En un LCD retroiluminado, esta capa es reemplazada por una fuente luminosa.

⁶ Sensores y acondicionadores – Enrique Mandado Pérez. Antonio Murillo Roldán.

2.2.5.1. FUNCIONAMIENTO

Cada píxel de un LCD consiste típicamente en una capa de moléculas alineadas entre dos electrodos transparentes y dos filtros de polarización; los ejes de transmisión de cada uno están (en la mayoría de los casos) perpendiculares entre sí. Sin cristal líquido entre el filtro polarizante, la luz que pasa por el primer filtro sería bloqueada por el segundo (cruzando) polarizador.

La superficie de los electrodos que están en contacto con los materiales de cristal líquido es tratada a fin de ajustar las moléculas de cristal líquido en una dirección en particular. Este tratamiento suele ser normalmente aplicable en una fina capa de polímero que es frotada unidireccionalmente utilizando, por ejemplo, un paño. La dirección de la alineación de cristal líquido se define por la dirección de frotación.

Antes de la aplicación de un campo eléctrico, la orientación de las moléculas de cristal líquido está determinada por la adaptación a las superficies. En un dispositivo twisted nematic, TN (uno de los dispositivos más comunes entre los de cristal líquido), las direcciones de alineación de la superficie de los dos electrodos son perpendiculares entre sí, y así se organizan las moléculas en una estructura helicoidal, o retorcida. Debido a que el material es de cristal líquido birrefringente, la luz que pasa a través de un filtro polarizante se gira por la hélice de cristal líquido que pasa a través de la capa de cristal líquido, lo que le permite pasar por el segundo filtro polarizado. La mitad de la luz incidente es absorbida por el primer filtro polarizante, pero por lo demás todo el montaje es transparente.

Cuando se aplica un voltaje a través de los electrodos, una fuerza de giro orienta las moléculas de cristal líquido paralelas al campo eléctrico, que distorsiona la estructura helicoidal (esto se puede resistir gracias a las fuerzas elásticas desde que las moléculas están limitadas a las superficies). Esto reduce la rotación de la polarización de la luz incidente, y el dispositivo aparece gris. Si la tensión aplicada es lo suficientemente grande, las moléculas de cristal líquido en el centro de la capa son casi completamente desenrolladas y la polarización de la luz incidente no es rotada, ya que pasa a través de la capa de cristal líquido. Esta luz será principalmente polarizada perpendicularmente al segundo filtro, y por eso será bloqueada y el píxel aparecerá negro. Por el control de la tensión aplicada a través de la capa de cristal líquido en cada píxel, la luz se puede permitir pasar a través de distintas cantidades, constituyéndose los diferentes tonos de gris.

El efecto óptico de un dispositivo twisted nematic (TN) en el estado del voltaje es mucho menos dependiente de las variaciones de espesor del dispositivo que en el estado del voltaje de compensación. Debido a esto, estos dispositivos suelen usarse entre polarizadores cruzados de tal manera que parecen brillantes sin tensión (el ojo es mucho más sensible a las variaciones en el estado oscuro que en el brillante). Estos dispositivos también pueden funcionar

en paralelo entre polarizadores, en cuyo caso la luz y la oscuridad son estados invertidos. La tensión de compensación en el estado oscuro de esta configuración aparece enrojecida debido a las pequeñas variaciones de espesor en todo el dispositivo. Tanto el material del cristal líquido como el de la capa de alineación contienen compuestos iónicos. Si un campo eléctrico de una determinada polaridad se aplica durante un período prolongado, este material iónico es atraído hacia la superficie y se degrada el rendimiento del dispositivo. Esto se intenta evitar, ya sea mediante la aplicación de una corriente alterna, ya por inversión de la polaridad del campo eléctrico que está dirigida al dispositivo (la respuesta de la capa de cristal líquido es idéntica, independientemente de la polaridad de los campos aplicados)

Cuando un dispositivo requiere un gran número de píxeles, no es viable conducir cada dispositivo directamente, así que cada píxel requiere un número de electrodos independiente. En cambio, la pantalla es multiplexada. En una pantalla multiplexada, los electrodos de la parte lateral de la pantalla se agrupan junto con los cables (normalmente en columnas), y cada grupo tiene su propia fuente de voltaje. Por otro lado, los electrodos también se agrupan (normalmente en filas), de modo que cada grupo obtiene una tensión de sumidero. Los grupos se han diseñado de manera que cada píxel tiene una combinación única y dedicada de fuentes y sumideros. Los circuitos electrónicos o el software que los controla, activa los sumideros en secuencia y controla las fuentes de los píxeles de cada sumidero.

2.2.5.2. ESPECIFICACIONES

RESOLUCIÓN

Las dimensiones horizontal y vertical son expresadas en píxeles. Las pantallas HD tienen una resolución nativa desde 1280x720 píxeles (720p), hasta 1920x1080 píxeles (1080p)

ANCHO DE PUNTO

La distancia entre los centros de dos píxeles adyacentes. Cuanto menor sea el ancho de punto, tanto menor granularidad tendrá la imagen. El ancho de punto suele ser el mismo en sentido vertical y horizontal, pero puede ser diferente en algunos casos.

TAMAÑO

El tamaño de un panel LCD se mide a lo largo de su diagonal, generalmente expresado en pulgadas (coloquialmente se llama "área de visualización activa").

TIEMPO DE RESPUESTA

Es el tiempo que demora un píxel en cambiar de un color a otro.

TIPO DE MATRIZ

Activa, pasiva y reactiva.

ÁNGULO DE VISIÓN

Es el máximo ángulo desde el que un usuario puede mirar el LCD, estando desplazado de su centro, sin que se pierda calidad de imagen. Las nuevas pantallas vienen con un ángulo de visión de 178 grados.

SOPORTE DE COLOR

Cantidad de colores soportados, coloquialmente conocida como “gama de colores”.

BRILLO

La cantidad de luz emitida desde la pantalla; también se conoce como luminosidad.

CONTRASTE

La relación entre la intensidad más brillante y la más oscura.

ASPECTO

La proporción de la anchura y la altura

2.2.5.3. TECNOLOGÍAS DE MATRIZ ACTIVA

TWISTED NEMATIC (TN)

Las pantallas *twisted nematic* contienen elementos de cristal líquido con desenrollado y enrollado en diversos grados para permitir que la luz pase a través de ellos. Cuando no se aplica voltaje a una celda de cristal líquido TN, la luz se polariza para pasar a través de la célula. En proporción a la tensión aplicada, las células LC giran hasta 90 grados cambiando la polarización y bloqueando el camino de la luz, para ajustar correctamente el nivel de la tensión de casi cualquier nivel de gris o la transmisión que se desee lograr.

IN-PLANE SWITCHING (IPS)

In-plane switching es una tecnología LCD que alinea las celdas de cristal líquido en una dirección horizontal. En este método, el campo eléctrico se aplica a través de cada uno de los extremos del cristal, pero esto requiere dos transistores por cada píxel en vez de un transistor, como era necesario para una pantalla estándar TFT. Esto hace que se produzca un mayor bloqueo del área de transmisión. También requiere un mayor brillo de fondo, el cual consumirá más energía, haciendo este tipo de pantalla menos deseable para los ordenadores portátiles.

VERTICAL ALIGNMENT (VA)

Las pantallas *vertical alignment*, VA, son una forma de pantallas LCD en las que el material de cristal líquido se encuentra en un estado vertical, eliminando así la necesidad de los transistores extras (como en el IPS). Cuando no se aplica voltaje, la celda de cristal líquido sigue siendo perpendicular al sustrato, creando una pantalla negra.

2.2.5.4. INCONVENIENTES

RESOLUCIÓN

Aunque los CRTs sean capaces de mostrar múltiples resoluciones de vídeo sin introducir artefactos, los LCD producen imágenes nítidas sólo en su "resolución nativa" y, a veces, en las fracciones de la resolución original. Al intentar ejecutar paneles LCD a resoluciones no nativas, por lo general los resultados en el panel de la escala de la imagen introducen emborronamientos de la imagen o bloqueos y, en general, es susceptible a varios tipos de HDTV borrosa. Muchos LCD no son capaces de mostrar modos de pantalla de baja resolución (por ejemplo, 320x200), debido a estas limitaciones de escala.

CONTRASTE

Aunque los LCD suelen tener más imágenes vibrantes y mejor contraste "del mundo real" (la capacidad de mantener el contraste y la variación de color en ambientes luminosos) que los CRT, tienen menor contraste que los CRTs en términos de la profundidad de los negros. El contraste es la diferencia entre un encendido completo (en blanco) y la desactivación de píxeles (negro), y los LCD pueden tener "sangrado de luz de fondo" donde la luz (por lo general, visto desde de las esquinas de la pantalla) se filtra y las fugas de negro se convierten en gris. En diciembre de 2007, los mejores LCD pueden acercarse al contraste de las pantallas de plasma en términos de entrega de profundidad de negro, pero la mayoría de los LCD siguen a la zaga.

TIEMPO DE RESPUESTA

Los LCD suelen tener tiempos de respuesta más lentos que sus correspondientes de plasma y CRT, en especial las viejas pantallas, creando imágenes fantasma cuando las imágenes se cargaban rápidamente. Por ejemplo, cuando se desplaza el ratón rápidamente en una pantalla LCD, se pueden ver múltiples cursores.

Algunas pantallas LCD tienen importantes aportaciones de retraso. Si el retraso es lo suficientemente grande, esa pantalla puede ser inadecuada para operaciones con el ratón rápidas y precisas (diseño asistido por ordenador, videojuegos de disparos en primera persona), en comparación con los monitores CRT o LCD pequeños y con insignificantes cantidades de retraso de

entrada. Los retrasos pequeños son a veces puestos de relieve en la comercialización.

ÁNGULO DE VISIÓN

Los paneles LCD tienden a tener un ángulo de visión limitado en relación con las CRT y las pantallas de plasma. Esto reduce el número de personas que pueden ver cómodamente la misma imagen - las pantallas de ordenadores portátiles son un excelente ejemplo. Así, esta falta de radiación es lo que da a las LCD su reducido consumo de energía en comparación con las pantallas de plasma y CRT. Si bien los ángulos de visión han mejorado al punto de que es poco frecuente que los colores sean totalmente incorrectos en el uso normal, a distancias típicas de uso de una computadora los LCD todavía permiten pequeños cambios en la postura del usuario, e incluso diferentes posiciones de los ojos producen una notable distorsión de colores, incluso para los mejores LCD del mercado.

DURABILIDAD

Los monitores LCD tienden a ser más frágiles que sus correspondientes CRT. La pantalla puede ser especialmente vulnerable debido a la falta de un grueso cristal protector como en los monitores CRT. Su durabilidad depende de su frecuencia de uso. Los fabricantes suministran en el manual del usuario un tiempo de durabilidad de la pantalla, regularmente expresado en horas de uso. Pero se puede extender este tiempo disminuyendo los niveles de brillo de la imagen (aún está en estudio).⁷

2.2.6. LECTOR DE TARJETAS

Los lectores de tarjetas son dispositivos electrónicos que se utilizan para leer y escribir datos en todo tipo de tarjetas, como las de memoria, las magnéticas, las SIM, y las inteligentes o RFID. Los lectores de tarjetas son el medio de comunicación entre la tarjeta y el anfitrión, por ejemplo un ordenador, una cámara o una impresora, y pueden ser lectores o grabadores, internos o externos, y específicos para un único tipo de tarjetas o compatibles con varios tipos de tarjetas.

2.2.6.1. CARACTERÍSTICAS

Los lectores de tarjetas pueden ser:

- Lectores o grabadores: los lectores sólo leen, los grabadores además tienen capacidad de escritura.

⁷ LCD - Wikipedia

- Internos o externos: los internos van en el interior del ordenador, en una ranura de 3,5 pulgadas. Los externos suelen tener conexión RS-232, USB o inalámbrica.
- Específicos o multi-tarjeta: pueden tener soporte para un único tipo de tarjetas o ser compatibles con varios tipos. El número de tarjetas que un lector multi-tarjeta acepta se expresa "x en 1", por ejemplo, "20 en 1".

2.2.6.2. TIPOS

LECTORES DE TARJETAS DE MEMORIA

Una tarjeta de memoria es un dispositivo de almacenamiento de memoria flash no volátil. Existen numerosos formatos como CompactFlash, SmartMedia, SD, MiniSD o MicroSD, y su uso está ampliamente extendido en ordenadores, cámaras fotográficas, teléfonos móviles, consolas o PDAs.

La mayoría de dispositivos electrónicos cuentan con lectores de tarjetas de memoria, incluyendo a menudo ranuras para distintos formatos para asegurar la compatibilidad.

LECTORES DE TARJETAS MAGNÉTICAS

Las tarjetas magnéticas son tarjetas dotadas de una banda magnética donde almacenan información. Se utilizan ampliamente como tarjetas de crédito, tarjetas de identificación, de control de accesos o tarjetas monedero.

Los lectores de tarjetas magnéticas leen los datos que contienen mediante contacto físico por inducción.

LECTORES DE TARJETAS INTELIGENTES

Las tarjetas inteligentes (smartcard) incluyen un chip con circuitos integrados que permiten la ejecución de cierta lógica programada. Se asemejan a un ordenador miniaturizado, ya que el chip contiene procesador, sistema operativo con cifrado de datos, memoria que hace las veces de disco duro y memoria RAM. Su uso se está implantando en tarjetas de crédito, tarjetas de identidad, tarjetas monedero, licencias de conducir, tarjetas sanitarias, etc.

Las tarjetas inteligentes no contienen baterías y son los lectores de tarjetas los encargados de suministrar energía eléctrica. Las tarjetas inteligentes pueden ser de contacto o sin contacto, según la comunicación con el lector sea mediante contacto físico o bien mediante inducción.

LECTORES DE TARJETAS SIM

Las tarjetas SIM son un tipo de tarjeta inteligente usada en teléfonos móviles para almacenar información específica de la red para autenticar e identificar a los suscriptores, de forma que cambiando la tarjeta es posible cambiar la línea de un terminal a otro.

Los lectores de tarjetas SIM permiten acceder a la información que contiene la tarjeta sin tener que insertarla en un teléfono.⁸

2.2.7. RELOJ EN TIEMPO REAL

En inglés, real-time clock, RTC), es un reloj de un ordenador, incluido en un circuito integrado, que mantiene la hora actual. Aunque el término normalmente se refiere a dispositivos en ordenadores personales, servidores y sistemas embebidos, los RTCs están presentes en la mayoría de los aparatos electrónicos que necesitan guardar el tiempo exacto.

El término se usa para evitar la confusión con los relojes hardware ordinarios, que sólo son señales comandadas por circuitos digitales, y no cuentan el tiempo en unidades humanas. Los RTC no deben ser confundidos con la computación en tiempo real (en inglés, real-time computing), que comparte su acrónimo de tres letras, pero que no se refiere directamente al tiempo del día.

Los RTCs a menudo tienen una fuente de alimentación alternativa, por lo que pueden seguir midiendo el tiempo mientras la fuente de alimentación principal está apagada o no está disponible. Esta fuente de alimentación alternativa es normalmente una batería de litio en los sistemas antiguos, pero algunos sistemas nuevos usan un condensador de alta capacidad, porque son recargables y pueden ser soldados. La fuente de alimentación alternativa también puede suministrar energía a una memoria no volátil.

La mayoría de los RTCs usan un oscilador de cristal, pero algunos usan la frecuencia de la fuente de alimentación. En muchos casos la frecuencia del oscilador es 32.768 kHz. Ésta es la misma frecuencia usada en los relojes de cuarzo, razón de que la frecuencia sea exactamente 2^{15} ciclos por segundo, es un ratio muy práctico para usar con circuitos de contadores binarios simples.

2.2.7.1. VENTAJAS DE UN RTC

- Bajo consumo de energía
- Libera de trabajo al sistema principal para que pueda dedicarse a tareas más críticas
- Algunas veces es más preciso que otros métodos

⁸ Lectores de tarjetas - Fransberns

2.2.7.2. EJEMPLOS

Muchos fabricantes de circuitos integrados fabrican RTCs, por ejemplo Intersil, Maxim, Philips, Texas Instruments y STMicroelectronics. El RTC fue introducido en los PC compatibles por IBM PC/AT en 1984, cuando usó un RTC MC146818. Posteriormente Dallas fabricó RTCs compatibles que fueron muy usados en ordenadores personales viejos, y que se pueden encontrar fácilmente en sus placas base por su distintiva batería negra y por su logo serigrafiado. En los sistemas nuevos el RTC está integrado en el chip southbridge.⁹

2.2.8. BATERÍAS

2.2.8.1. DEFINICIÓN Y CLASIFICACIÓN

La batería es un dispositivo que almacena energía en forma electroquímica y es el más ampliamente usado para almacenar energía en una variedad de aplicaciones.

BATERÍA PRIMARIA

Su reacción electroquímica es irreversible, es decir, después de que la batería se ha descargado no puede volver a cargarse.

BATERÍA SECUNDARIA

Su reacción electroquímica es reversible, es decir después de que la batería se ha descargado puede ser cargada inyectándole corriente continua desde una fuente externa. Su eficiencia en un ciclo de carga y descarga está entre el 70% y el 80%.

2.2.8.2. FUNCIONAMIENTO BÁSICO

En general el funcionamiento de una batería, se basa en una celda electroquímica.

Las celdas electroquímicas tienen dos electrodos: el ánodo y el cátodo.

El ánodo se define como el electrodo en el que se lleva a cabo la oxidación y el cátodo donde se efectúa la reducción.

Los electrodos pueden ser de cualquier material que sea un conductor eléctrico, como metales semiconductores.

⁹ Reloj en tiempo real - Wikipedia

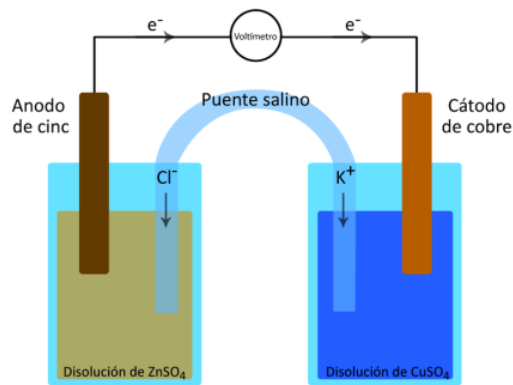


FIGURA 22: EJEMPLO DE BATERÍA

Para completar el circuito eléctrico, las disoluciones se conectan mediante un conductor por el que pasan los cationes y aniones, conocido como puente de sal (o como puente salino).

Los cationes disueltos (K^+) se mueven hacia el Cátodo y los aniones (Cl^-) hacia el Ánodo para que las disoluciones se neutralicen.

La corriente eléctrica fluye del ánodo al cátodo porque existe una diferencia de potencial eléctrico entre ambos electrolitos.

2.2.8.3. CARACTERÍSTICAS QUE DEFINEN A UNA BATERÍA

LA CANTIDAD DE ENERGÍA QUE PUEDE ALMACENAR

El número de Wh puede calcularse multiplicando el valor del voltaje nominal por el número de Ah.

LA MÁXIMA CORRIENTE QUE PUEDE ENTREGAR (DESCARGA)

Se especifica como un número fraccionario, por ejemplo para $C=200[Ah]$ una de tipo $C/20=10A$ quiere decir que la batería puede entregar 10A por 20 horas.

LA PROFUNDIDAD DE DESCARGA QUE PUEDE SOSTENER

Representa la cantidad de energía que puede extraerse de una batería. Este valor está dado en forma porcentual.

2.2.8.4. TIPOS DE BATERÍAS:

BATERÍA DE PLOMO ÁCIDO

Es el tipo de batería recargable más común por su buena relación de desempeño-costo aunque es la de menor densidad de energía por peso y volumen.

Esta batería cuenta con varias versiones:

SHALLOW-CYCLE O DE CICLO CORTO

Usada en automóviles, en los cuales se necesita una corta explosión de energía que es forzada desde la batería para encender el motor.

DEEP-CYCLE O DE CICLO PROFUNDO

Diseñada para repetidos ciclos de carga y descarga. La mayoría de las aplicaciones requiere este tipo de baterías.

LA VERSIÓN SELLADA "GEL-CELL" CON ADITIVOS

Estos aditivos vuelven el electrolito en un gel anti-derrames, está pensada para ser montada de lado, pero su alto costo la limita a aplicaciones en aviones militares.

La batería plomo-ácido es, en el momento actual, imprescindible e insustituible, especialmente en automoción, pero también en muchas otras aplicaciones que exigen continuidad en el suministro de energía eléctrica.

- **AUTOMOCIÓN:** batería de arranque (SLI)
- **TRACCIÓN:** carretillas, locomotoras de mina, vehículos industriales, embarcaciones...
- **INDUSTRIALES:** servicios continuos, regulación de cargas, energía fotovoltaica...

BATERÍA DE NÍQUEL-CADMIO (NI-CD)

Se caracteriza por sus celdas selladas, por tener la mitad del peso y por ser más tolerante a altas temperaturas que una batería de plomo-ácido convencional.

Tiene una tasa muy baja de autodescarga.

Debido a regulaciones ambientales ha sido reemplazada por las de NiMH e Ion-litio en notebooks y en otros tipos de electrónica de alto precio.

Tiene el efecto de memoria, lo cual acelera su proceso de descarga.

Alguno de los usos más frecuentes son en juguetes, equipos estéreos y máquinas fotográficas.

Existen diseños especializados, como el de baterías para aviones sin sellar, lo que permite expulsar el oxígeno y el hidrogeno cuando la batería se carga o se descarga rápidamente

BATERÍA DE NICKEL-HIDRURO METÁLICO (NI-MH)

Es una extensión de la tecnología de NiCd, que ofrece una mayor densidad de energía y cuyo ánodo está hecho de metal hidruro, evitando así los problemas ambientales de la NiCd.

Además su efecto memoria es casi despreciable.

No es capaz de entregar altos picos de potencia, tiene un alto grado de autodescarga y es muy peligrosa si se sobrecarga.

Aún es de precio elevado, aunque se estima que su costo disminuirá al producirse vehículos eléctricos a gran escala.

Entre las aplicaciones de esta batería podemos destacar:

VEHÍCULOS DE PROPULSIÓN TOTALMENTE ELÉCTRICA:

General Motors EV1, Honda EV Plus, Ford Ranger EV, Scooter Vectrix.

VEHÍCULOS HÍBRIDOS:

Toyota Prius, Honda Insight o las versiones híbridas de los Ford Escape, Chevrolet Malibu y Honda Civic Hybrid también las utilizan.

BATERÍA DE ION-LITIO

Es de una nueva tecnología, la cual ofrece una densidad de energía de 3 veces la de una batería plomo-ácido. Esta gran mejora viene dada por su bajo peso atómico (6,9 vs 209 para la de plomo).

Además cuenta con el voltaje más alto por celda (3.5v), lo cual reduce el número de celdas en serie para alcanzar cierto voltaje, lo que reduce su costo de manufactura.

Tiene una tasa muy baja de autodescarga.

Tiene rápida degradación y sensibilidad a las temperaturas elevadas, que pueden dar como resultado su destrucción por inflamación o incluso explosión.

Requieren en su configuración como producto de consumo, la inclusión de dispositivos adicionales de seguridad, derivando en un coste superior que ha limitado la extensión de su uso a otras aplicaciones.

Su utilización se ha popularizado en aparatos como teléfonos móviles,

agendas electrónicas, ordenadores portátiles y reproductores de música.

Las baterías de Ion-Litio, al ser baterías más compactas, permiten manejar más carga, lo que hay que tener en cuenta para lograr automóviles eléctricos prácticos.

BATERÍA DE POLÍMERO-LITIO

Es una batería de litio con un polímero sólido como electrolítico.

Estas baterías tienen una densidad de energía de entre 5 y 12 veces las de Ni-Cd o Ni-MH, a igualdad de peso. A igualdad de capacidad, las baterías de Li-Po son, típicamente, cuatro veces más ligeras que las de Ni-Cd de la misma capacidad.

La gran desventaja de estas baterías es que requieren un trato mucho más delicado, bajo riesgo de deteriorarlas irreversiblemente o, incluso, llegar a producir su ignición o explosión.

Un elemento de Li-Po tiene un voltaje nominal, cargado, de 3.7V. Nunca se debe descargar una batería por debajo de 3.0V por celda; nunca se la debe cargar más allá de 4.3V por celda.

Hyundai ha adoptado un enfoque ligeramente diferente en cuanto al almacenamiento de la energía, ya que utiliza baterías de polímero de litio en lugar de usar un compuesto de níquel con iones de litio.

La empresa Apple usa actualmente la tecnología de las baterías de polímero litio en iPod o iPhone. También se encuentra en dispositivos como teléfonos móviles y PDAs.

BATERÍA DE AIRE-ZINC

Con una fabricación más barata y capacidades que pueden superar en 3 veces a las populares de Ion-Litio.

Las nuevas baterías de Zinc-Aire funcionan utilizando el oxígeno almacenado en un cuarto como electrodo, mientras la batería contiene un electrolito y el electrodo de Zinc permite que el aire circule dentro de una caja porosa, logrando el milagro de la electricidad.

La compañía ReVolt se encuentra trabajando en llevar el Zinc-Aire a vehículos eléctricos; para esto será necesario incrementar el número de ciclos de carga en 10.000, algo un poco lejano todavía para los prototipos que sólo alcanzan las 300-500 cargas y descargas.

Las pilas a base de zinc tienen como principal ventaja la posibilidad de ser recicladas sin límite, sin perder ni sus cualidades químicas, ni sus cualidades físicas.

Entre las aplicaciones de esta batería podemos destacar:

- La tecnología zinc-aire, respetuosa con el medio ambiente, encuentra su mejor aplicación en prótesis de oído, aparatos electrónicos portátiles y en el sector de la automoción.
- Es probable que el futuro del coche eléctrico pase por el desarrollo de baterías más potentes de zinc-aire, que sustituyan a las de iones de litio.
- ReVolt Technology ha solicitado 30 millones de dólares dentro del programa del gobierno de Estados Unidos para acelerar el desarrollo tecnológico de las baterías de zinc-aire, para posibilitar vehículos eléctricos con mayor autonomía.¹⁰

2.2.8.5. COMPARATIVA ENTRE BATERÍAS

Tipo	Energía/peso	Tensión por elemento (V)	Duración (número de recargas)	Tiempo de carga	Auto-descarga por mes (% del total)
Plomo	30-50 Wh/kg	2 V	1000	8-16h	5 %
Ni-Cd	48-80 Wh/kg	1,25 V	500	10-14h *	30%
Ni-Mh	60-120 Wh/kg	1,25 V	1000	2h-4h *	20 %
Li-ion	110-160 Wh/kg	3,16 V	4000	2h-4h	25 %
Li-Po	100-130 Wh/kg	3,7 V	5000	1h-1,5h	10%

FIGURA 23: COMPARATIVA ENTRE BATERÍAS

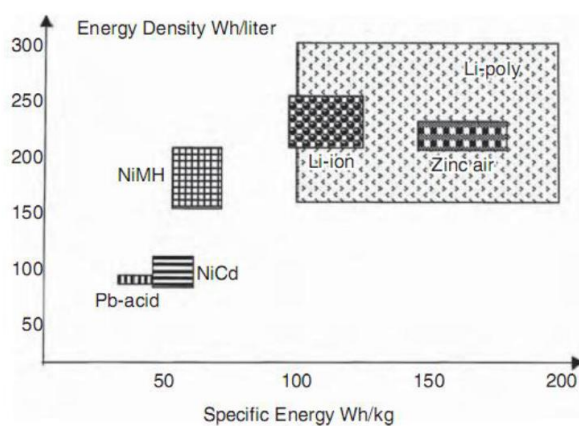


FIGURA 24: ENERGÍA Y DENSIDAD ESPECÍFICA DE VARIAS BATERÍAS

¹⁰ Tecnología de las baterías

2.3. DESCRIPCIÓN SOFTWARE

2.3.1. ARDUINO

El hardware Arduino que hemos elegido para realizar nuestro proyecto, nos proporciona un software de código abierto, desde el que podemos programar en alto nivel basado en programación C/C++ que hace fácil escribir código y cargarlo a nuestra placa Arduino Mega. El entorno está escrito en Java y basado en Processing (como anteriormente hemos explicado en la introducción), avr-gcc y otros programas de código abierto. Funciona en diferentes sistemas operativos como Windows, Mac OS X y Linux. Nosotros hemos trabajado con el entorno de Windows y Linux.

2.3.1.1. COMUNICACIÓN ARDUINO-PC

La comunicación entre el Arduino Mega y el PC, la realizamos mediante un cable USB, mediante el cual, gracias a la configuración Serial que nos proporciona el Arduino, podremos transmitir datos entre PC y Arduino, y viceversa.

Todos los Arduinos tiene al menos un puerto serie (UART o USART), que se comunica a través de los pines (TX) y (RX); para el caso del Arduino Mega, que es el que utilizamos nosotros, disponemos de 3 puertos Serials, además del utilizado para la comunicación Serial a través del USB.

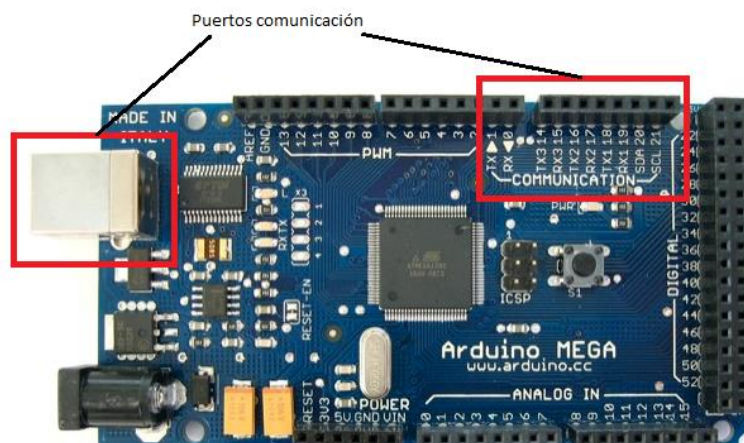


FIGURA 25: PUERTOS DE COMUNICACIÓN DEL ARDUINO MEGA

La comunicación de Arduino con nuestro PC la realizamos mediante USB. En primer lugar hay que decir que con esta placa no es necesaria la instalación de Drivers para los sistemas operativos anteriormente mencionados, que si sería necesaria si la placa utilizada hubiera sido Arduino UNO.

Nuestra interacción con Arduino se realiza con el SerialMonitor que nos proporciona el software de Arduino, que sería el siguiente:

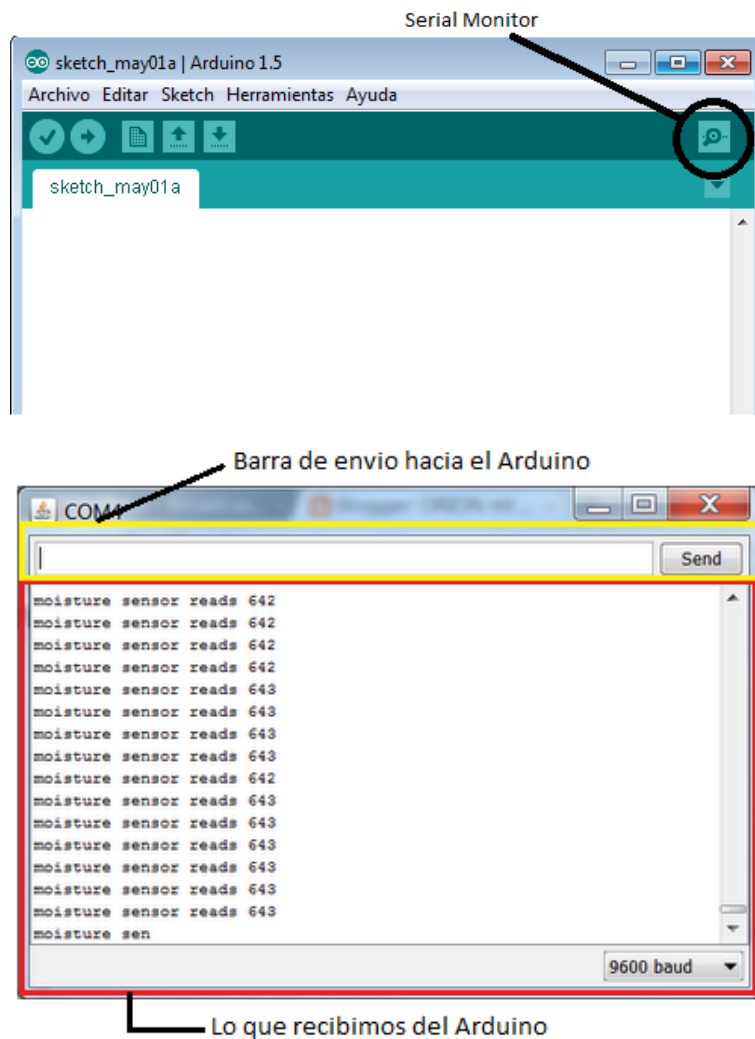


FIGURA 26: CAPTURA DE PANTALLA DE LA OPCIÓN SERIAL MONITOR

Con esta herramienta podemos ver, en tiempo real, las medidas de nuestros sensores, las funciones que está realizando, tiempo, valores de variables que nos gustaría ver...; además, si fuera necesario introducir algún valor al Arduino para que hiciera alguna función específica, sería posible desde la barra de envío.

2.3.1.2. INSTALACIÓN SOFTWARE ARDUINO

La instalación del software de Arduino es sencilla. En primer lugar necesitamos descargarnos la última versión del programa, que podremos encontrar en la página oficial de Arduino (arduino.cc), siempre teniendo en cuenta el sistema operativo instalado en nuestro PC, ya sea Windows, Mac OS X o Linux. (Véase contenido del CD 9.3.1 Arduino (software)).

Una vez descargados los datos comprimidos en un Zip, en nuestro caso para Windows, lo descomprimiríamos y nos encontraríamos los siguientes archivos:

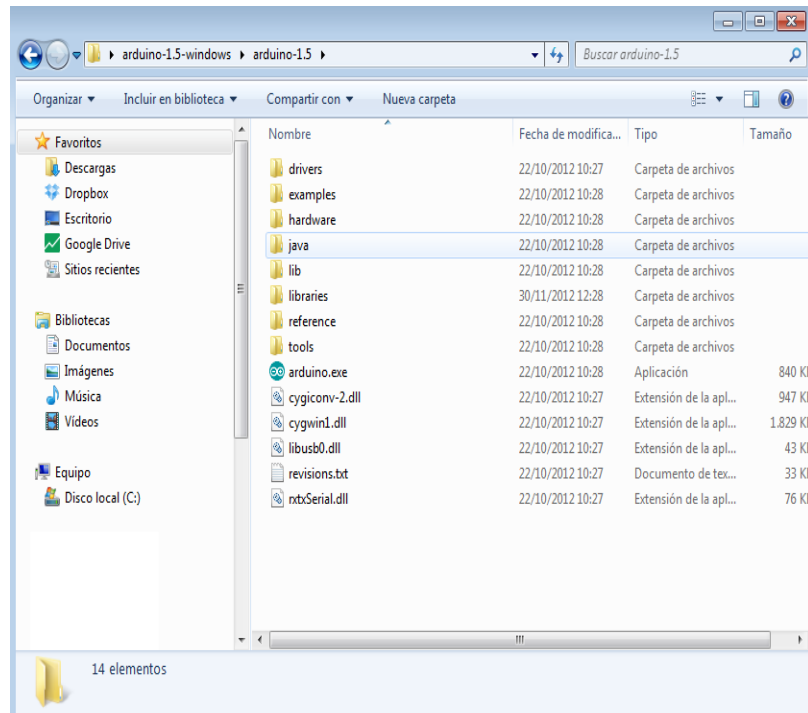


FIGURA 27: CAPTURA DE PANTALLA DE LA CARPETA ARDUINO DESPUÉS DE LA INSTALACIÓN

Si usáramos una placa Arduino UNO necesitaríamos ir a la carpeta *drivers* para instalar los drivers de comunicación; como utilizamos Arduino Mega no es necesario.

No es imprescindible una instalación; con hacer doble click sobre el archivo que tiene el nombre de *arduino.exe*, se ejecutara el programa y nos aparecerá lo siguiente:

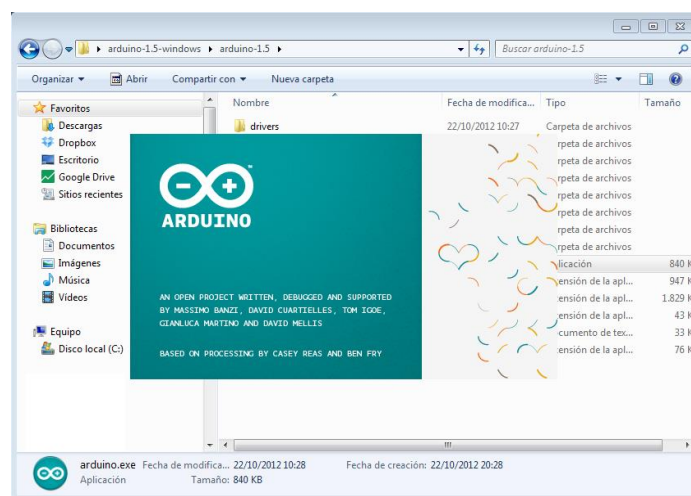


FIGURA 28: CAPTURA DE PANTALLA DE LA INSTALACIÓN DEL SOFTWARE DE ARDUINO

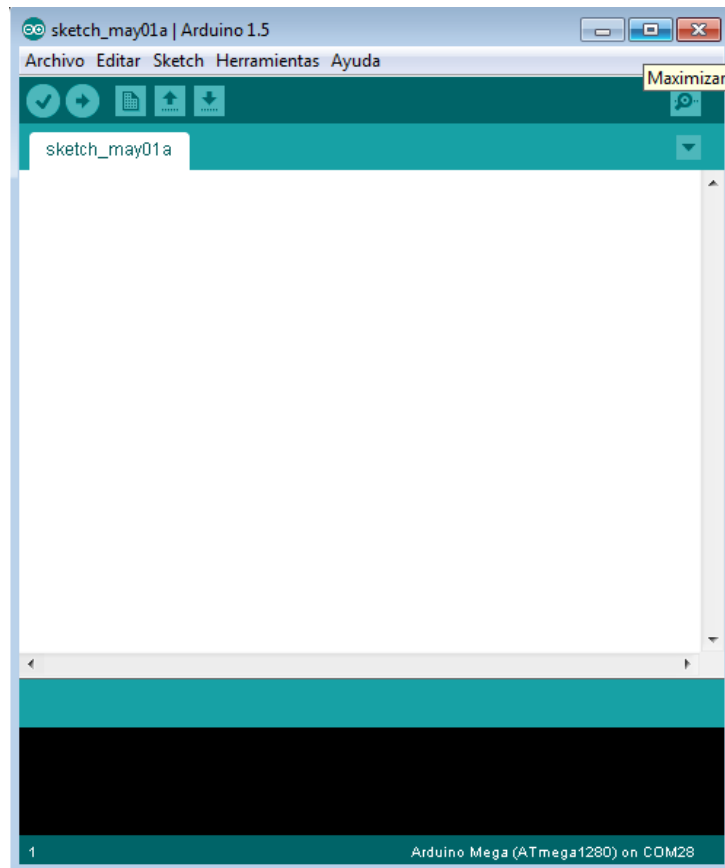


FIGURA 29: PANTALLA PRINCIPAL DEL SOFTWARE ARDUINO

A continuación seleccionamos la placa Arduino que usaremos, así como el puerto serie en el que está conectada la placa; vamos a la barra *Herramientas>Tarjeta>ArduinoMega*. A continuación seleccionaremos el procesador de nuestra placa en *Herramientas>Processor>ATmega1280*, y por último elegiremos el puerto serie que nos ha detectado el ordenador, en el cual está conectada nuestra placa: *Herramientas>Puerto Serial>COM...*

Ya tendríamos configurado nuestro programa de Arduino para empezar a programar en él.

Una vez que tengamos el programa escrito, sólo nos quedará compilarlo y subirlo a nuestra placa. Podemos verificar si el programa tiene algún problema de sintaxis en la programación C, pulsando el siguiente botón:

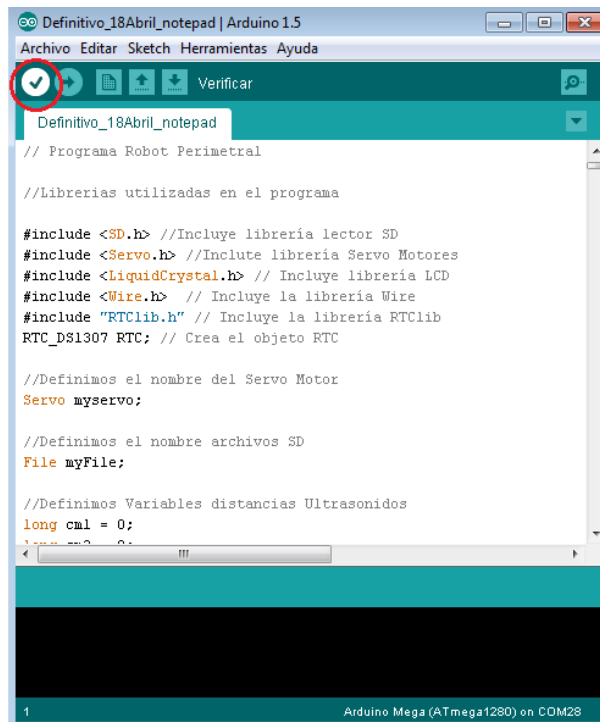


FIGURA 30: COMPILACIÓN EN ARDUINO

Si queremos subir el programa a la placa, con pulsar el siguiente botón nos verificaría el programa y, si es correcto, lo subiría directamente:

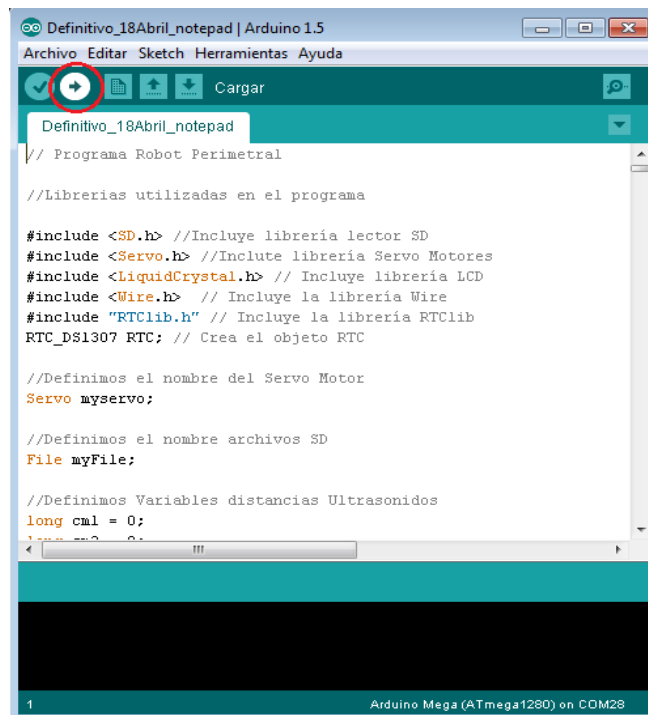


FIGURA 31: CARGAR PROGRAMA EN ARDUINO

Con estos pasos ya tendríamos cargado en la placa Arduino Mega nuestro programa, listo para poder probarlo.

2.3.1.3. PROGRAMACIÓN ARDUINO

Los programas en Arduino se estructuran de la siguiente manera: Librerías y funciones (la principal, *loop*).

En nuestro programa lo hemos estructurado de la siguiente manera: para empezar, hemos declarado todas las librerías que vamos a necesitar, desde el control de la LCD hasta el control de la SD.

```
// Programa Robot Perimetral

//Librerias utilizadas en el programa

#include <SD.h> //Incluye librería lector SD
#include <Servo.h> //Inclute librería Servo Motores
#include <LiquidCrystal.h> // Incluye librería LCD
#include <Wire.h> // Incluye la librería Wire
#include "RTClib.h" // Incluye la librería RTClib
RTC_DS1307 RTC; // Crea el objeto RTC
```

FIGURA 32: INCLUSIÓN DE LIBRERÍAS EN ARDUINO

Después configuramos los motores y la SD para su escritura.

```
//Definimos el nombre del Servo Motor
Servo myservo;

//Definimos el nombre archivos SD
File myFile;
```

FIGURA 33: DEFINICIÓN DE VARIABLES COMPLEJAS

Tras las configuraciones, declaramos todas las variables globales del programa (mostramos variables iniciales).

```
//Definimos Variables distancias Ultrasonidos
long cm1 = 0;
long cm2 = 0;
long cm3 = 0;
long cm4 = 0;

//Definimos los Pines Arduino Mega para los Ultrasonidos
const int pinTrig1 = 46;
const int pinEchol = 47;
const int pinTrig2 = 26;
const int pinEcho2 = 27;
const int pinTrig3 = 42;
const int pinEcho3 = 43;

//Definimos los Pines Arduino Mega para el control de los motores
int motorIzq1Pin = 30; // pin 1 del L293 7
int motorIzq2Pin = 31; // pin 2 del L293 8
int motorDchalPin= 32; // pin 3 del L293 10
```

FIGURA 34: DEFINICIÓN DE VARIABLES

Después de definir las variables, realizamos la configuración inicial que queremos cuando encendamos el Arduino (*Void Setup()*).

```
//Configuración inicial Arduino
void setup()
{
    //Velocidad en Baudios de la comunicación serie al ordenador
    Serial.begin(9600);

    //Configuración comunicación I2C
    Wire.begin(); // Establece la velocidad de datos del bus I2C
    RTC.begin(); // Establece la velocidad de datos del RTC

    // Configuración inicial LCD
    lcd.begin(16,2); //(N° de columnas,N°de filas)
    lcd.setCursor(0,0); //Donde empieza a escribir el cursor

    //Configuramos las interrupciones
    digitalWrite(pin, state);
    pinMode(pin, OUTPUT);
    digitalWrite(pinA, state);
    pinMode(pinA, OUTPUT);
}
```

FIGURA 35: CONFIGURACIÓN INICIAL EN ARDUINO

Tras el configuración inicial que queremos del Arduino, empieza la función principal o bucle (*Void Loop()*). Es donde escribimos todo lo que tiene que hacer nuestro robot paso por paso; es decir, mediante llamadas a funciones va realizando las diferentes operaciones.

```
//Funcion Principal Arduino
void loop()
{
    // Abrimos el archivo de la SD. Solo un archivo puede ser abierto a la vez.
    // Tenemos que cerrar uno antes de abrir otro.
    myFile=SD.open("seguidor.txt",FILE_WRITE);

    if(myFile)
    {
        // Inicializamos las banderas de decision
        flag1=0;
        flag2=0;

        estadobaterias();
        delay(50);

        pantalla(); //Función Pantalla que nos escribe en la LCD

        if(contpared==0)
    }
}
```

FIGURA 36: COMIENZO DE LA FUNCIÓN PRINCIPAL DEL PROGRAMA EN ARDUINO

Por último, creamos las funciones necesarias para que el robot perimetral realice sus operaciones.

```
// Bloque de Funciones necesarias para el Robot

//Función Robot gira las ruedas hacia adelante
int delante()
{
    //Escribimos el nombre de la función en la SD
    myFile.print("Delante");
    myFile.println();

    //Configuramos la velocidad de los Motores y la dirección de giro de las ruedas
    analogWrite(speedPin1, 90); // PWM Motor Izquierdo
    analogWrite(speedPin2, 105); // PWM Motor Derecho
    digitalWrite(motorIzqPin, LOW); // Pone el pin 1 del Puente en H del motor izquierdo a LOW
    digitalWrite(motorIzq2Pin, HIGH); // Pone el pin 2 del Puente en H del motor izquierdo a HIGH
    digitalWrite(motorDchaPin, HIGH); // Pone el pin 1 del Puente en H del motor derecho a HIGH
    digitalWrite(motorDcha2Pin, LOW); // Pone el pin 2 del Puente en H del motor derecho a LOW

    //Mostramos en la LCD la función que esta realizando el robot
    lcd.clear();
    lcd.setCursor(4,0);
}
```

FIGURA 37: CREACIÓN DE FUNCIONES EN ARDUINO

Con esta organización en el programa, podemos seguir más fácilmente su ejecución y poder revisar errores y fallos ocurran donde ocurran.

2.3.2. PROTEUS

El software Proteus es una compilación de programas de diseño y simulación electrónica. Está desarrollado por Labcenter Electronics y consta de dos programas principales, Ares e Isis, y módulos VSM y Electra.

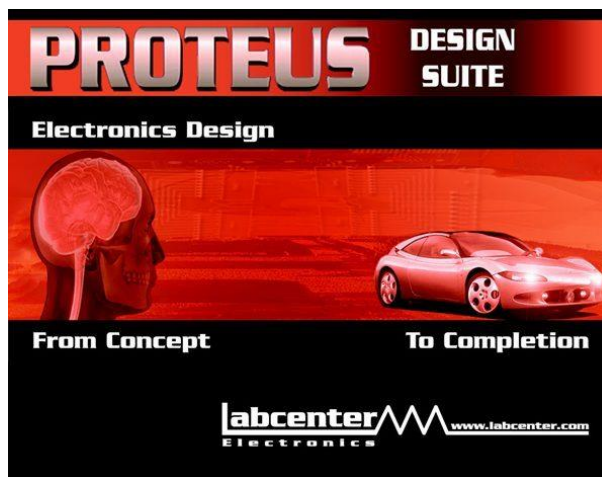


FIGURA 38: CAPTURA DE PANTALLA DEL SOFTWARE PROTEUS

En el proyecto hemos utilizado dos de las herramientas que nos proporciona este software de electrónica, que son Isis y Ares.

2.3.2.1. INSTALACIÓN DEL SOFTWARE PROTEUS

El software Proteus se puede conseguir a través de la página web del desarrollador Labcenter (ver 9.1 Enlaces).

Puedes descargar una versión Demo de Proteus 8 (ver 9.1 Enlaces), o si no, realizar una compra con la licencia educativa para conseguir el programa completo, con un coste que va desde los 200€ hasta casi unos 5000€. Puedes ver los precios en los enlaces (ver 9.1 Enlaces).

Nosotros hemos utilizado una versión Demo de Proteus 7.7. La instalación de este software es muy sencilla, tiene un archivo ejecutable “.exe”. Tras ejecutarlo veremos que nos instala una serie de aplicaciones. De entre todas nosotros sólo usaremos Ares (para diseño de PCB’s) e Isis (para la simulación de circuitos electrónicos, microcontroladores...).

2.3.2.2. ISIS (SIMULADOR DE CIRCUITOS)

Este programa nos va a ayudar a simular los circuitos eléctricos y electrónicos, para después montarlos físicamente y poder realizar un prototipo y una aplicación real.

La pantalla inicial de esta aplicación es la siguiente:

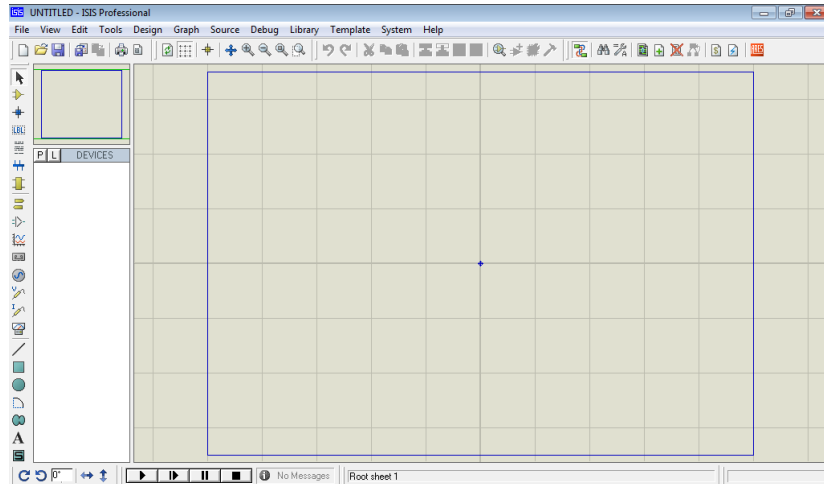


FIGURA 39: PANTALLA INICIAL DE PROTEUS

Desde esta pantalla podremos ir seleccionando los componentes en el lateral izquierdo:

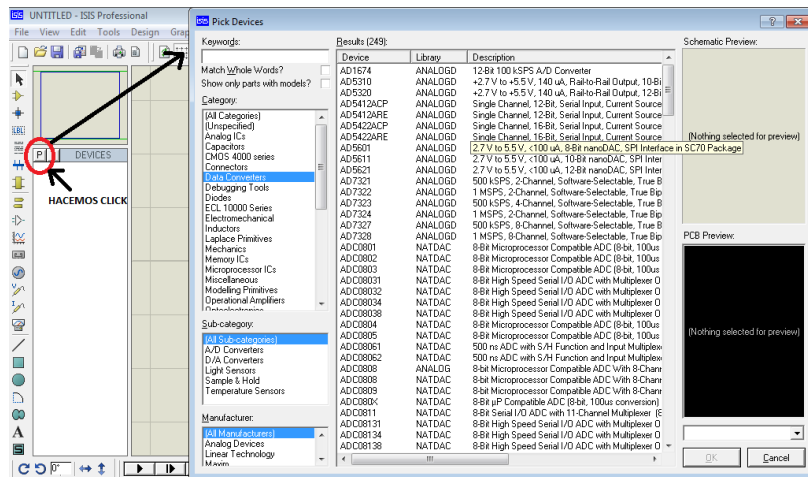


FIGURA 40: SELECCIÓN DE COMPONENTES EN PROTEUS

Una vez que tenemos seleccionados los componentes, sólo tendremos que seleccionarlos y hacer click sobre la pantalla principal para colocarlos. Quedaría de la siguiente manera:

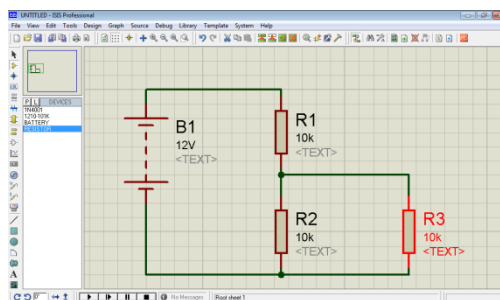


FIGURA 41: POSICIONAMIENTO DE COMPONENTES EN PROTEUS

Hemos realizado un divisor de tensión sencillo con 2 resistencias y una carga; ahora configuraremos los diferentes elementos, haciendo dobleclick sobre ellos para cambiar sus propiedades:

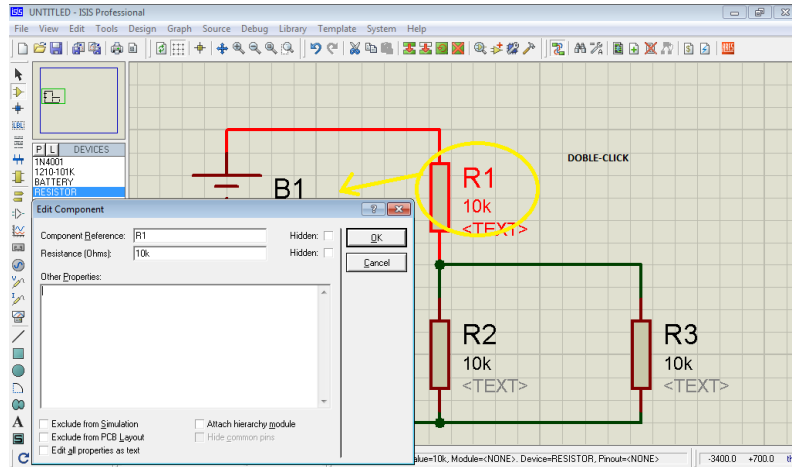


FIGURA 42: EDICIÓN DE COMPONENTES EN PROTEUS

Después de realizar los cambios deseados, podemos simular el circuito y comprobar la tensión, corriente y otros parámetros más, que (pasan/circulan) por un determinado sitio, por ejemplo:

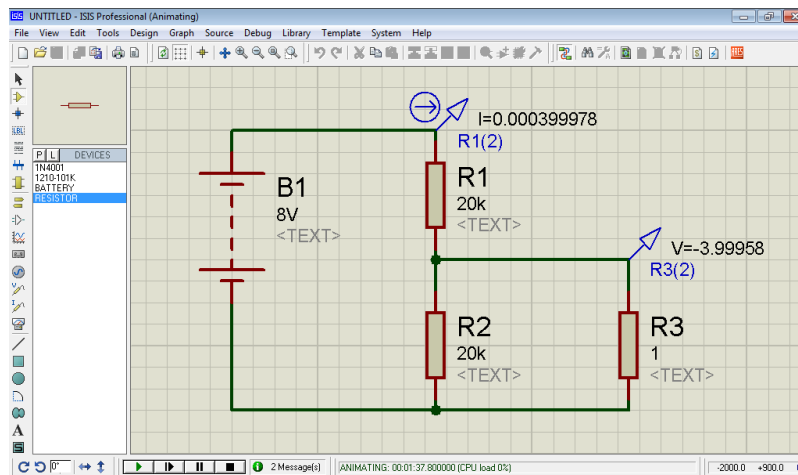


FIGURA 43: SIMULACIÓN EN PROTEUS

Con este simulador, podremos realizar más medidas usando osciloscopios, medidor de señales analógicas o digitales, amperímetros de corriente continua y corriente alterna, voltímetros...

También hay que resaltar la funcionalidad de poder realizar simulaciones con microcontroladores, pues podemos insertar un programa realizado en C, con la extensión “.hex”, en el microcontrolador deseado, y de esta manera poder simular la programación de éstos con los dispositivos conectados que queramos manejar.

Es una herramienta muy versátil y cómoda para poder realizar nuestros prototipos y probar su funcionamiento.

2.3.2.3. ARES (REALIZAR PCB'S)

Es una herramienta para el diseño de placas de circuito impreso. Ésta dentro del paquete de aplicaciones de Proteus.

La pantalla inicial de esta herramienta es la siguiente:

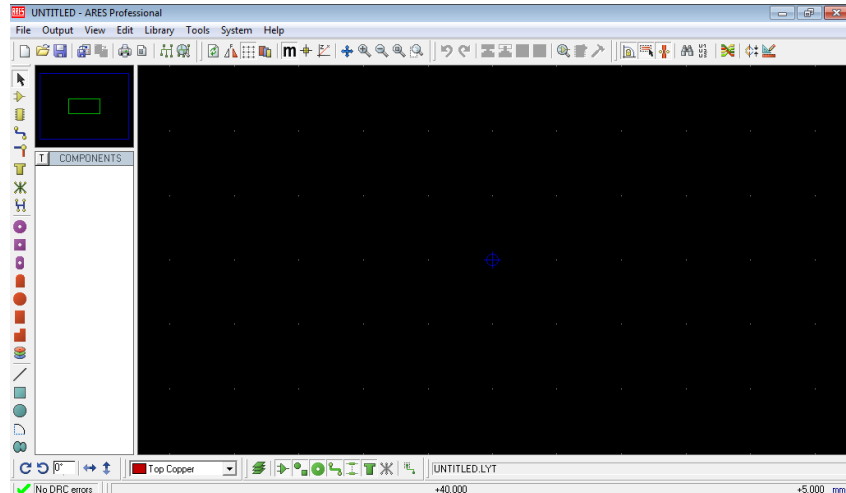


FIGURA 44: PANTALLA PRINCIPAL DE ARES

Como se puede observar es muy parecida a la pantalla de trabajo de ISIS, fondo negro.

Nos ayudará a realizar nuestro diseño de placa de circuito impreso de dos formas: introduciendo los elementos individualmente uno a uno, o a través de la aplicación ISIS, en la que tendremos ya nuestro esquema eléctrico realizado y probado.

Para hacer la placa de circuito impreso desde la herramienta ISIS, el primer paso lo damos desde esta, donde nos aparece un icono de fondo rojo que pone ARES:

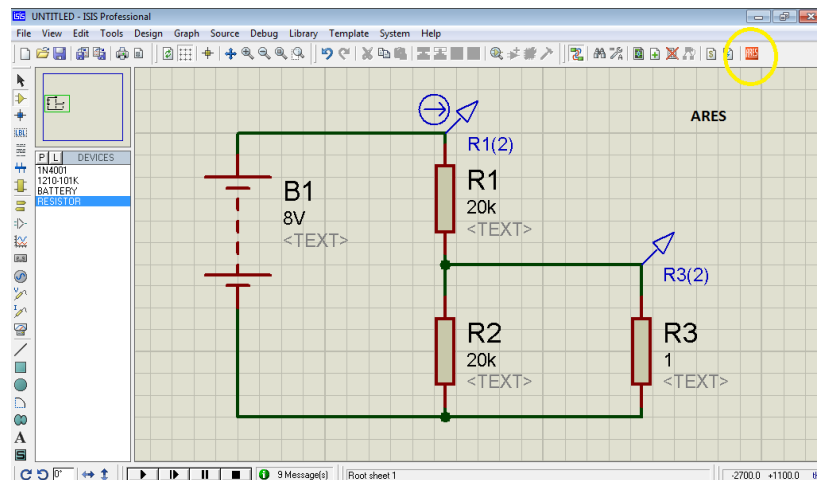


FIGURA 45: CAMBIAR DE ISIS A ARES EN PROTEUS

Después de seleccionar todos los encapsulados, realizaremos un recuadro, seleccionando como capa *board edge*, con el tamaño deseado de la PCB:

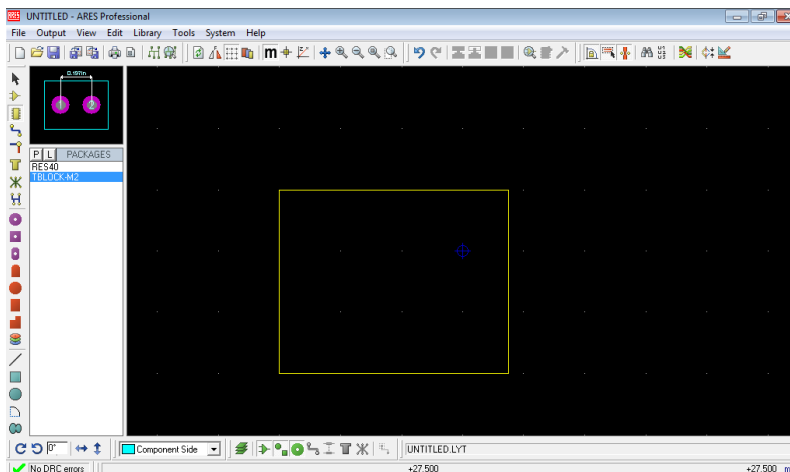


FIGURA 48: SELECCIÓN DE TAMAÑO PCB EN ARES

El siguiente paso sería colocar los encapsulados dentro del rectángulo que hemos dibujado anteriormente, de la mejor manera posible para conseguir una PCB reducida y funcional:

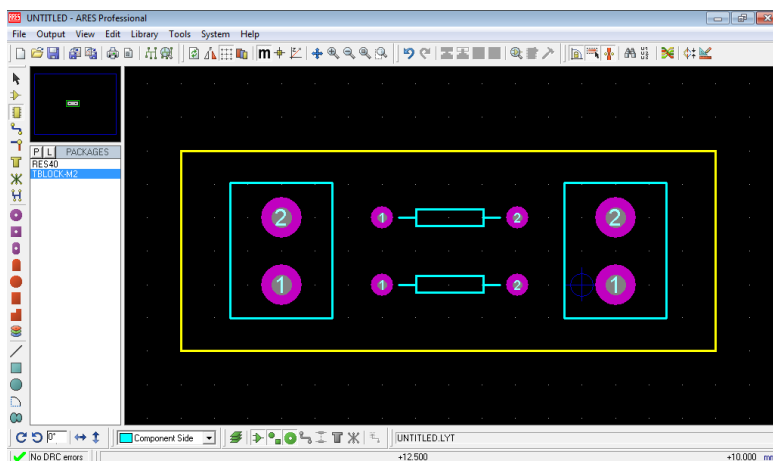


FIGURA 49: DISTRIBUCIÓN DE ENCAPSULADOS EN ARES

Por ultimo quedaría enrutar las pistas lo, que se podría hacer de forma automática (con una configuración previa) o manual, seleccionando el ancho de pistas y la separación entre estas.

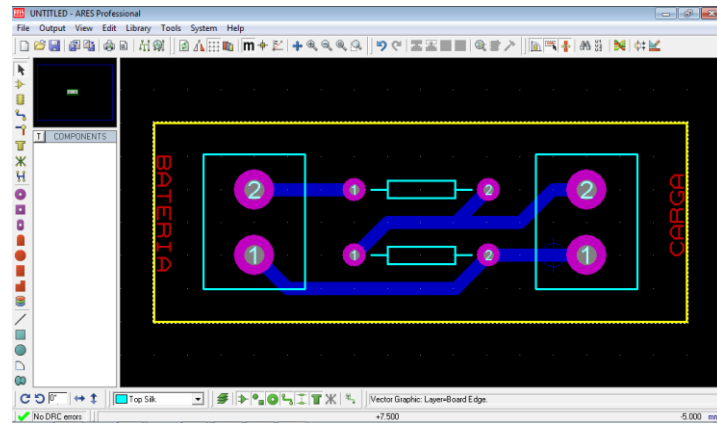


FIGURA 50: ROUTING DE PISTAS EN ARES

Ya estaría lista para imprimir e insolar.

Podemos ver una simulación 3D de cómo quedaría nuestra PCB fabricada:

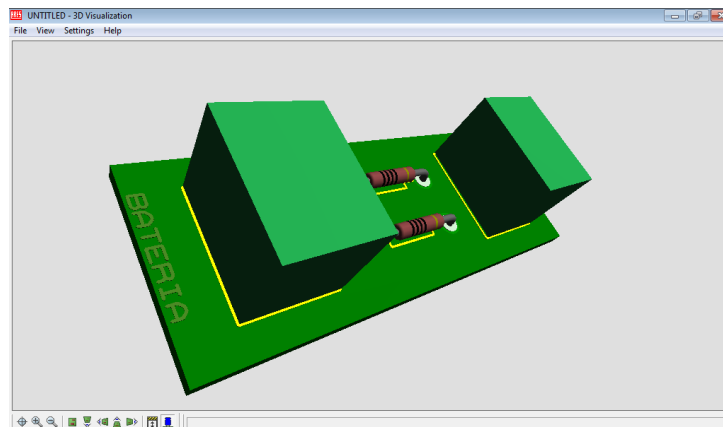


FIGURA 51: VISTA 3D EN ARES

3. DESARROLLO/DISEÑO DEL SISTEMA

3.1. DISEÑO MECÁNICO

Después de la elección de la placa, nos centramos en el diseño del robot. Lo primero de todo, es la parte mecánica. Queríamos una plataforma amplia donde poder poner la placa y todos los demás componentes necesarios, además necesitábamos dos ruedas conectadas a sendos motores desde donde pudiéramos controlar los giros y la velocidad del robot móvil. Como rueda trasera, utilizamos una rueda omnidireccional. Puesto que el diseño mecánico de estos componentes nos hubiera llevado mucho tiempo, nos decidimos por comprar un prototipo que nos ofrecía todo lo que queríamos, donde solamente fue necesario el montaje de las diferentes partes.

Se puede ver el despiece general en 9.2.1 Despiece mecánico.

3.2. DISEÑO HARDWARE

3.2.1. MOTORES CC

A continuación nos centramos en los motores. Necesitábamos controlar el giro y velocidad de los motores. Para ello nos decidimos a hacerlo mediante el integrado L293D.



FIGURA 52: INTEGRADO L293D

El integrado **L293D** incluye cuatro circuitos para manejar cargas de potencia media, con la capacidad de controlar corriente de hasta 600 mA en cada circuito y una tensión entre 4,5 V y 36 V.

Los circuitos individuales se pueden usar de manera independiente para controlar cargas de todo tipo y, en el caso de ser motores, manejar un único sentido de giro. Pero, además, cualquiera de estos cuatro circuitos sirve para configurar la mitad de un **punte H**.

El integrado permite formar, entonces, dos puentes H completos, con los que se

puede realizar el manejo de dos motores. En este caso el manejo será bidireccional, con frenado rápido y con posibilidad de implementar fácilmente el control de velocidad.

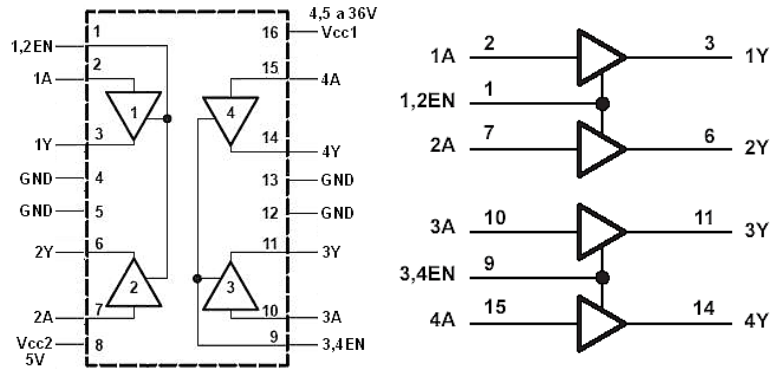


FIGURA 53: DIAGRAMA DE PINES L293D

Las salidas tienen un diseño que permite el manejo directo de cargas inductivas tales como relés, solenoides, motores de corriente continua y motores por pasos, ya que incorpora internamente los diodos de protección de contracorriente para cargas inductivas.

Las entradas son compatibles con niveles de lógica TTL. Para lograr esto, incluso cuando se manejen motores de voltajes no compatibles con los niveles TTL, el chip tiene patas de alimentación separadas para la lógica (**VCC2**, que debe ser de 5V) y para la alimentación de la carga (**VCC1**, que puede ser entre 4,5V y 36V).

Las salidas poseen un circuito de manejo en configuración "totem-pole" (término en inglés que se traduce como "poste de tótem", nombre que, gráficamente, nos remite a un "apilamiento" de transistores, como las figuras en los famosos totems indígenas).

En esta estructura, unos transistores en configuración Darlington conducen la pata de salida a tierra y otro par de transistores en conexión seudo Darlington aporta la corriente de alimentación desde **VCC2**. Las salidas tienen diodos incorporados en el interior del chip para proteger el circuito de manejo de potencia de las contracorrientes de una carga inductiva.

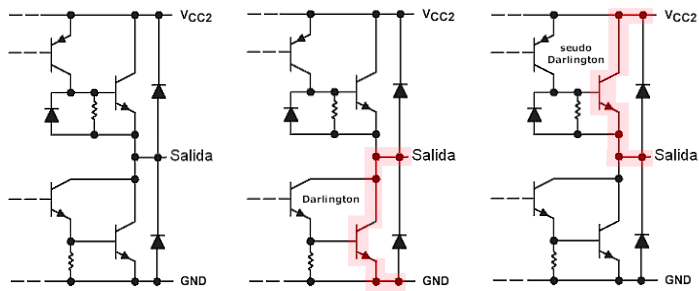


FIGURA 54: FUNCIONAMIENTO DEL CIRCUITO

Estos circuitos de salida se pueden habilitar en pares por medio de una señal TTL. Los circuitos de manejo de potencia 1 y 2 se habilitan con la señal **1,2EN**, y los circuitos 3 y 4 con la señal **3,4EN**.

Las entradas de habilitación permiten controlar con facilidad el circuito, lo que facilita la regulación de velocidad de los motores por medio de una **modulación de ancho de pulso (PWM)**. En ese caso, las señales de habilitación, en lugar de ser estáticas, se controlarían por medio de pulsos de ancho variable.

Las salidas actúan cuando su correspondiente señal de habilitación está en alto. En estas condiciones, las salidas están activas y su nivel varía en relación con las entradas. Cuando la señal de habilitación del par de circuitos de manejo está en bajo, las salidas están desconectadas y en un estado de alta impedancia.

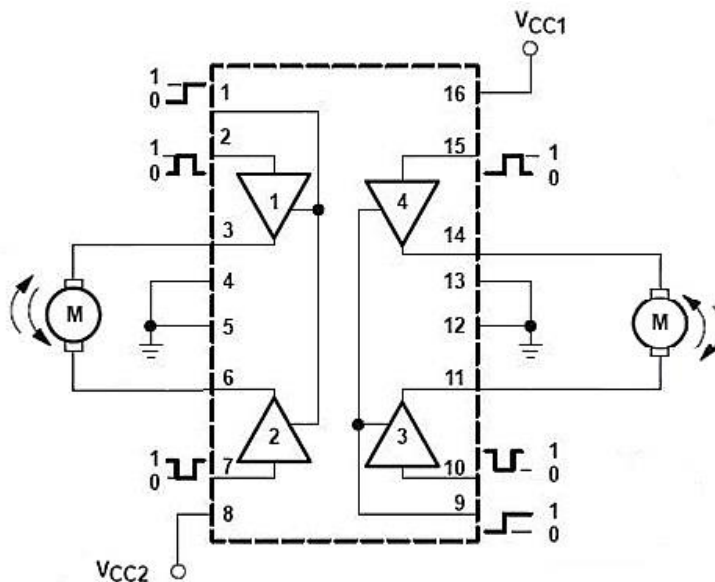


FIGURA 55: CONEXIONADO PARA DOS MOTORES CON GIRO EN AMBOS SENTIDOS

Por medio de un control apropiado de las señales de entrada y conectando el motor a sendas salidas de potencia, cada par de circuito de manejo de potencia

conforma un **punto H** completo, como se ve en el diagrama de arriba.

En la tabla de funcionamiento que sigue se pueden observar los niveles TTL que corresponden a cada situación de trabajo:

ENTRADAS			SALIDAS
ENABLE (pin 1)	Pin (2)	Pin (7)	
L	-	-	L
H	H	L	H (giro sentido horario)
H	L	H	H(giro sentido anti-horario)

ENTRADAS			SALIDAS
ENABLE (pin 9)	Pin (10)	Pin (15)	
L	-	-	L
H	L	H	H (giro sentido horario)
H	H	L	H(giro sentido anti-horario)

TABLA 3: GIRO DE LOS MOTORES

Después de comprobar el funcionamiento del integrado L293D, diseñamos el conexionado de éste y su programación para Arduino.

A continuación se muestra el conexionado del L293D a la placa Arduino Mega,

diseñado con Fritzing (ver en 9.1 Enlaces):

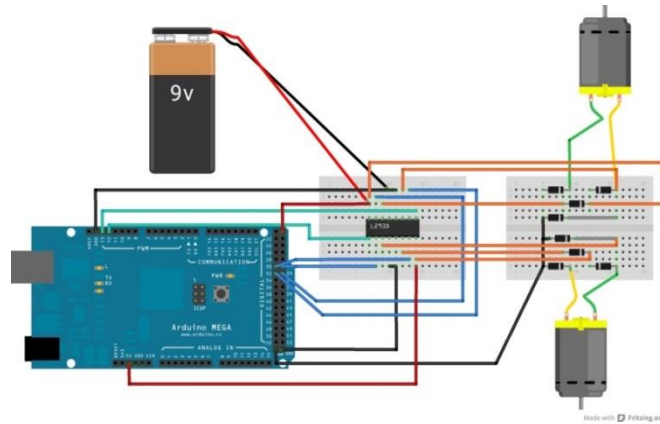


FIGURA 56: DISEÑO DE CONEXIONADO DE MOTORES Y ARDUINO

Y ésta sería la programación del L293D mediante Arduino.

- Configuración de los pines:



```

Control_de_dos_motores_mediante_L293D | Arduino 1.0.3
Archivo Editar Sketch Herramientas Ayuda
Control_de_dos_motores_mediante_L293D $
/* Control de dos motor DC con puente H */
/* Igor Mena Benito & Alberto Rodriguez Campos */
/* Visto el coche desde arriba */

//Definiamos los Pines Arduino Mega para el control de los motores
int motorIzqPin = 30; // pin 1 del L293 7
int motorIzq2Pin = 31; // pin 2 del L293 8
int motorDcha1Pin= 32; // pin 3 del L293 10
int motorDcha2Pin= 33; // pin 4 del L293 11
int speedPin1 = 12; // pin7 enable del L293 que controla pin1 y pin2 (PWM)
int speedPin2 = 13; // pin5 enable del L293 que controla pin3 y pin4 (PWM)
int valorVeloc = 250; //velocidad de los motores ( se puede regular mayor o menor )

void setup() {
// pone todos los otros pines usados como salidas
pinMode(motorIzqPin, OUTPUT);
pinMode(motorIzq2Pin, OUTPUT);
pinMode(speedPin1, OUTPUT);
pinMode(motorDcha1Pin, OUTPUT);
pinMode(motorDcha2Pin, OUTPUT);
pinMode(speedPin2, OUTPUT);

// activa la RPA del pin speedPin para activar los motores
digitalWrite(speedPin1, HIGH);
digitalWrite(speedPin2, HIGH);

// para leer por pantalla lo que escribamos en los bucles
Serial.begin(9600);

// establecemos la velocidad de los motores con ese valor
analogWrite(speedPin1, valorVeloc);
analogWrite(speedPin2, valorVeloc);
}

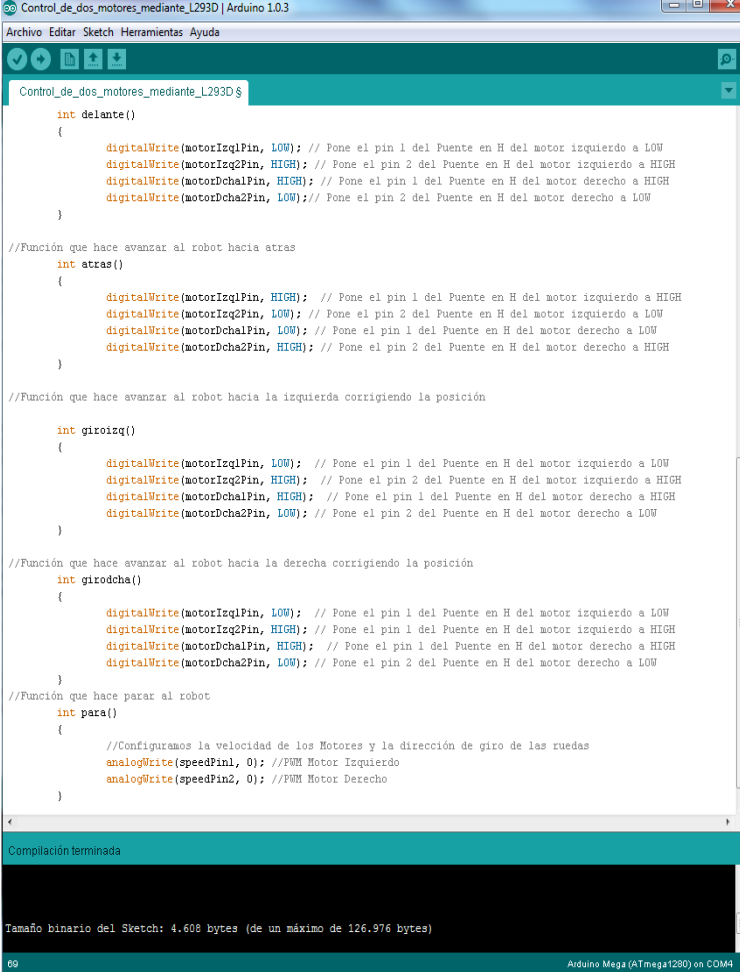
void loop() {
}
}

Compilación terminada
Tamaño binario del Sketch: 4.608 bytes (de un máximo de 126.976 bytes)
43 Arduino Mega (ATmega1280) on COM4

```

FIGURA 57: CONFIGURACIÓN DE LOS PINES DE ARDUINO PARA EL CONTROL DE MOTORES

- Funciones de movimiento de los motores:



```

Control_de_dos_motores_mediante_L293D | Arduino 1.0.3
Archivo Editar Sketch Herramientas Ayuda

Control_de_dos_motores_mediante_L293D $
int delante()
{
    digitalWrite(motorIzq1Pin, LOW); // Pone el pin 1 del Puente en H del motor izquierdo a LOW
    digitalWrite(motorIzq2Pin, HIGH); // Pone el pin 2 del Puente en H del motor izquierdo a HIGH
    digitalWrite(motorDcha1Pin, HIGH); // Pone el pin 1 del Puente en H del motor derecho a HIGH
    digitalWrite(motorDcha2Pin, LOW); // Pone el pin 2 del Puente en H del motor derecho a LOW
}

//Función que hace avanzar al robot hacia atras
int atras()
{
    digitalWrite(motorIzq1Pin, HIGH); // Pone el pin 1 del Puente en H del motor izquierdo a HIGH
    digitalWrite(motorIzq2Pin, LOW); // Pone el pin 2 del Puente en H del motor izquierdo a LOW
    digitalWrite(motorDcha1Pin, LOW); // Pone el pin 1 del Puente en H del motor derecho a LOW
    digitalWrite(motorDcha2Pin, HIGH); // Pone el pin 2 del Puente en H del motor derecho a HIGH
}

//Función que hace avanzar al robot hacia la izquierda corrigiendo la posición
int giroizq()
{
    digitalWrite(motorIzq1Pin, LOW); // Pone el pin 1 del Puente en H del motor izquierdo a LOW
    digitalWrite(motorIzq2Pin, HIGH); // Pone el pin 2 del Puente en H del motor izquierdo a HIGH
    digitalWrite(motorDcha1Pin, HIGH); // Pone el pin 1 del Puente en H del motor derecho a HIGH
    digitalWrite(motorDcha2Pin, LOW); // Pone el pin 2 del Puente en H del motor derecho a LOW
}

//Función que hace avanzar al robot hacia la derecha corrigiendo la posición
int girodcha()
{
    digitalWrite(motorIzq1Pin, LOW); // Pone el pin 1 del Puente en H del motor izquierdo a LOW
    digitalWrite(motorIzq2Pin, HIGH); // Pone el pin 2 del Puente en H del motor izquierdo a HIGH
    digitalWrite(motorDcha1Pin, HIGH); // Pone el pin 1 del Puente en H del motor derecho a HIGH
    digitalWrite(motorDcha2Pin, LOW); // Pone el pin 2 del Puente en H del motor derecho a LOW
}

//Función que hace parar al robot
int para()
{
    //Configuramos la velocidad de los Motores y la dirección de giro de las ruedas
    analogWrite(speedPin1, 0); //PWM Motor Izquierdo
    analogWrite(speedPin2, 0); //PWM Motor Derecho
}

Compilación terminada

Tamaño binario del Sketch: 4.608 bytes (de un máximo de 126.976 bytes)

69 Arduino Mega (ATmega1280) on COM4

```

FIGURA 58: FUNCIONES DE CONTROL DE LOS MOTORES

Después de comprobar el correcto funcionamiento de las funciones, nos disponemos a hacer un circuito impreso para realizar el control de los motores; esto nos ayudará a eliminar cables y a poder probar el robot móvil en el suelo.

Para ello hacemos el diseño. Todo el desarrollo del diseño de la placa de circuito impreso se puede ver en el apartado 4.1 Pistas.

3.2.1.1. PROBLEMA SURGIDO

Después de insolar la placa y hacer el conexionado con Arduino, probamos el robot en el suelo con un programa bien sencillo. Intentamos hacer al robot ir hacia adelante. (Véase código en CD)

El problema llega cuando, aun con las funciones programadas exactamente igual y mandando la misma velocidad de giro a las ruedas, el robot no consigue ir recto. Siempre se va torciendo hacia uno de los lados. He aquí el gran problema de la robótica móvil, el hacer que un robot vaya recto. Esto es

prácticamente imposible a causa de las ruedas, las llantas y los motores. Éstos, por muy bien fabricados que estén, nunca llegarán a ser iguales por completo. Por lo tanto, el diámetro de las ruedas y de las llantas no será el mismo en ambas ruedas, lo que creará una deriva (mayor o menor), al igual que los motores nunca transmitirán la misma potencia de giro a las ruedas.

Ante este problema, nos es imposible hacer que el robot vaya recto. Por lo tanto tendremos que buscar soluciones. Intentaremos solucionarlo con la utilización de sensores, los cuales guiarán al robot e intentarán hacer que vaya lo más recto posible (siempre con una referencia física).

3.2.1.2. POSIBLES SOLUCIONES

TACÓMETRO

El tacómetro es un dispositivo que mide la velocidad de giro de un eje, normalmente la velocidad de giro de un motor. Se mide en revoluciones por minuto (RPM). Actualmente se utilizan con mayor frecuencia los tacómetros digitales, por su mayor precisión.¹¹

Con un tacómetro podríamos controlar los giros que realiza cada motor, pero tendríamos el problema de que el motor gire y que la rueda no lo haga, o que ésta resbale en la superficie que se encuentra, por lo que nuevamente no sería posible hacer que el robot vaya recto.

ENCODERS

El encoder es un transductor rotativo, que mediante una señal eléctrica (normalmente un pulso o una señal senoidal) nos indica el ángulo girado. Si este sensor rotatorio lo conectáramos mecánicamente con una rueda o un husillo, también nos permitiría medir distancias lineales.¹²

Con la utilización de encoders (ya sean ópticos, de efecto Hall, etc.) tenemos el mismo problema que con el tacómetro. Podremos controlar cuántos giros da cada rueda, pero no podremos saber si ésta ha resbalado o no. Por lo tanto seguimos sin solucionar el problema de que el robot vaya recto.

GPS

¹¹ Tacómetro - Wikipedia

¹² Encoders – Blog www.mecatronic.co

Las siglas GPS se corresponden con "*Global Positioning System*", que significa Sistema de Posicionamiento Global (aunque sus siglas GPS han popularizado el producto en el mundo comercial).

En síntesis podemos definir el GPS como un Sistema Global de Navegación por Satélite (GNSS) que nos permite fijar a escala mundial la posición de un objeto, una persona, un vehículo o una nave.

La precisión del GPS puede llegar a determinar los puntos de posición con errores mínimos, aunque en la práctica hablemos de metros.¹³

Por lo tanto no es posible la utilización de esta tecnología dentro de una habitación, puesto que los errores son muy grandes para lo que nosotros queremos.

3.2.1.3. REPLANTEAMIENTO DE LOS OBJETIVOS

Nos hemos encontrado con uno de los grandes problemas de la robótica, hacer que un robot vaya recto. Con el tiempo y los recursos que tenemos, nos vemos incapacitados de realizar la segunda parte de nuestros objetivos. Por lo que, después de tomar una decisión conjunta con nuestro tutor, decidimos reorientar esta parte del proyecto.

El robot que construiremos será un "**Robot Perimetral**", que irá evitando obstáculos y podrá recorrer el perímetro de una habitación. Además, almacenará y mostrará información, tal como habíamos planeado.

3.2.2. BATERÍAS

Puesto que queríamos hacer nuestro robot autónomo, tuvimos que decidir qué baterías tenía que llevar. Necesitábamos unas baterías de duración media y que no ocuparan mucho espacio físico, además de que fueran recargables. Nos decidimos por unas baterías de LI-PO.

Puesto que los motores eran la parte del robot que más iban a consumir, decidimos hacer dos paquetes de baterías. Uno alimenta a los motores, y el otro alimenta a Arduino y demás componentes del robot.

Estos paquetes son iguales. Dos baterías de 3.7v y 1.300mAh en serie. Por lo tanto, 7.5v y 1300mAh (el voltaje de las baterías puede variar dependiendo de la memoria que tengan, desde 8.2v a 7.5v)

¹³ ¿Qué es el GPS? – Mateo P

En el apartado de Cálculos (ver 4.2 Consumos) se muestra el consumo de cada parte independientemente.

En ese mismo apartado se puede ver cómo dependiendo de la carga de las baterías de los motores, se ha calculado el PWM de cada uno de éstos (véase 4.4 Rectas PWM).

Uno de los problemas detectados con las baterías ocurre a la hora de la recarga, pues la tenemos que realizar de manera individual.

3.2.3. LECTOR DE TARJETAS

Cuando el robot no está autónomo y queremos hacer pruebas podemos leer por pantalla todo lo que le pasa mediante SerialMonitor. El problema está cuando lo queremos dejar de forma autónoma. Necesitamos un dispositivo que grabe toda la información para luego tratarla y corregir nuestros errores.

Para ello elegimos grabar la información en una tarjeta SD. Con tal propósito utilizaremos un grabador-lector de tarjetas SD. También utilizaremos la tarjeta SD para grabar toda tipo de información y mostrarla al usuario, el cual podrá leer la tarjeta en su ordenador.

Hemos elegido el dispositivo SD Module compatible con Arduino, que nos permitirá la escritura/lectura de la información que necesitemos registrar sobre las distintas funciones que realiza el robot.

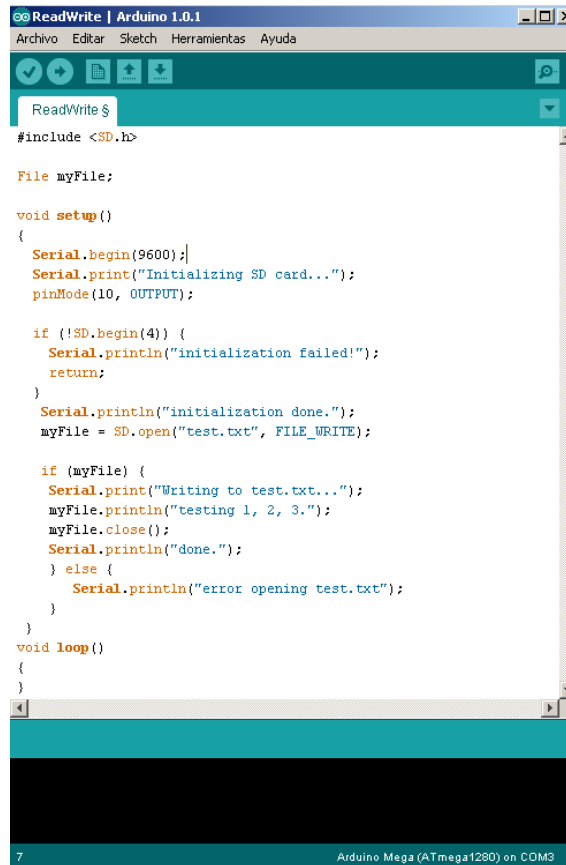
La comunicación entre el microcontrolador del Arduino y el modulo SD se realiza mediante el protocolo de comunicación SPI (ver 3.4.4. Comunicación SPI).



FIGURA 59: LECTOR DE TARJETAS SD MODULE

Para poder utilizar el SD Module es necesario incluir la librería <SD.h>. Aquí se

puede ver un ejemplo de escritura en la tarjeta SD:



```

ReadWrite | Arduino 1.0.1
Archivo  Editar  Sketch  Herramientas  Ayuda
ReadWrite $
#include <SD.h>

File myFile;

void setup()
{
  Serial.begin(9600);
  Serial.print("Initializing SD card...");
  pinMode(10, OUTPUT);

  if (!SD.begin(4)) {
    Serial.println("initialization failed!");
    return;
  }
  Serial.println("initialization done.");
  myFile = SD.open("test.txt", FILE_WRITE);

  if (myFile) {
    Serial.print("Writing to test.txt...");
    myFile.println("testing 1, 2, 3.");
    myFile.close();
    Serial.println("done.");
  } else {
    Serial.println("error opening test.txt");
  }
}

void loop()
{
}
7 Arduino Mega (ATmega1280) on COM3

```

FIGURA 60: FUNCIÓN PARA LA ESCRITURA EN UNA TARJETA SD

Se pueden ver las grabaciones de la SD durante diferentes pruebas en el contenido del CD.

3.2.4. SENSORES DE ULTRASONIDOS

Lo primero de todo será la elección de nuestros sensores de ultrasonidos. Para ello hemos elegido el sensor **HC - SR04**.



FIGURA 61: SENSOR DE ULTRASONIDOS HC-SR04

Es un sensor que se adapta perfectamente a las características deseadas:

- Tiene un rango de detección que va desde 2cm hasta 4m (perfecto para la detección de obstáculos dentro de una habitación).
- La frecuencia de funcionamiento es de 40Hz.
- Dispone de 4pins (Vcc, GND, Trig y Echo). Al tener la señal *Trig* y *Echo* en diferentes canales hace que el dispositivo se abarate.

Primeramente hacemos varias pruebas para ver cómo funciona este sensor con Arduino, aquí podemos ver un ejemplo:

```

Prueba_sensor_ultrasonidos | Arduino 1.0.3
Archivo Editar Sketch Herramientas Ayuda
Prueba_sensor_ultrasonidos $
long cm1 = 0;
const int pinTrig1 = 46;
const int pinEcho1 = 47;

void setup()
{
  Serial.begin(9600); //Velocidad en Baudios de la comunicación serie al ordenador
}

void loop()
{
  cm1 = ping(pinTrig1, pinEcho1);
  delay(5);

  Serial.print("Sensor 1 cm:");
  Serial.print(cm1);
  Serial.println();
}

//Función que calcula la distancia de los Ultrasonidos
long ping(int pinTrig, int pinEcho)
{
  //Configuración Pines Arduino Mega para que ultrasonidos envíen y reciban el ECHO
  pinMode(pinTrig, OUTPUT);
  digitalWrite(pinTrig, LOW);
  delayMicroseconds(2);
  digitalWrite(pinTrig, HIGH);
  delayMicroseconds(10);
  digitalWrite(pinTrig, LOW);
  pinMode(pinEcho, INPUT);

  long duration = pulseIn(pinEcho, HIGH); // Guarda el tiempo en recibir el ECHO

  return (microsecondsToCentimeters(duration)); //Llama función transforma el tiempo en distancia CM
}

//Función transforma tiempo en distancia (CM)
long microsecondsToCentimeters(long microseconds)
{
  return microseconds / 29 / 2;
  //La velocidad del sonido son 340m/s o 29 microsegundos por cm
  //Como el ultrasonidos tiene que ir y volver, para encontrar la distancia del
  //objeto tenemos que tomar la mitad de la distancia que ha recorrido
}

Compilación terminada

Tamaño binario del Sketch: 5.322 bytes (de un máximo de 126.976 bytes)

7 Arduino Mega (ATmega1280) on COM4
  
```

FIGURA 62: CÓDIGO DE PRUEBA PARA SENSOR DE ULTRASONIDOS

Aquí se puede apreciar una captura de Serial Monitor a la hora de capturar las distancias:

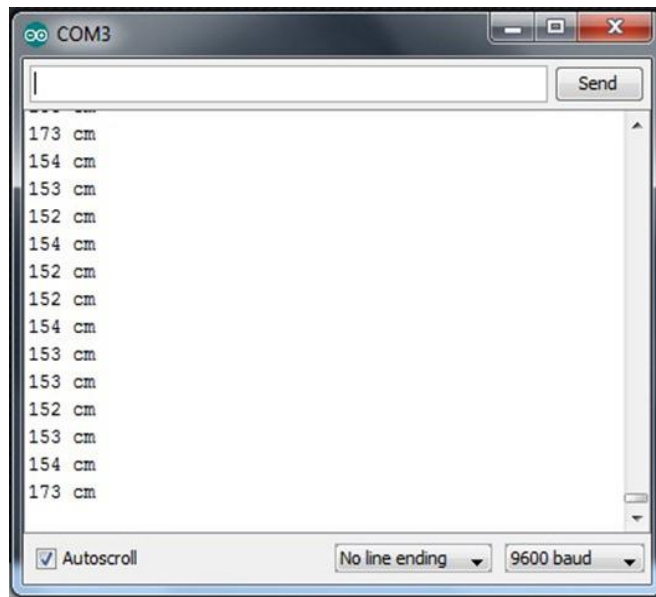


FIGURA 63: DISTANCIA A UN OBSTÁCULO MEDIANTE SERIAL MONITOR

Después de ver cómo funcionan nuestros sensores decidimos cómo enfocar nuestro objetivo. Puesto que nuestro robot tiene que seguir el perímetro de una habitación, el sensor lo ubicaremos a la izquierda del robot, para que éste vaya siguiendo el perímetro de la habitación desde la izquierda.

Después de colocado a la izquierda hacemos un pequeño programa con el cual el sensor va calculando la distancia a la pared y haciendo que rectifique según varias funciones (si se aleja más de 20 cm el robot gira a la izquierda y si se acerca más de 15 cm el robot gira a la derecha). Tras probarlo físicamente vemos que el robot se nos mueve mucho de la parte trasera e incluso pierde su rumbo.

Con los problemas surgidos decidimos poner dos sensores en la parte izquierda del robot, uno en la parte delantera y otro en la trasera. Con esto conseguimos hacer una comparación de la distancia de ambos sensores y que no se desvíe mucho y pierda su referencia, (ver diagrama de flujo Función Seguir izquierda 3.3.5.2 Funciones de movimiento y decisión).

Con este código conseguimos que el robot vaya siguiendo la pared de la izquierda; ahora intentaremos solucionar otros problemas.

Primeramente, que el robot no encuentre pared a la izquierda. Como no tenemos una referencia (puesto que los sensores sólo llegan hasta 3m.), tenemos que calcular que, cuando no haya pared a la izquierda, el robot avance hasta que el segundo sensor tenga una distancia mayor de 30 cm. Y después tenemos que hacer que éste gire 90°. Es imposible calcular exactamente el giro, como ya hemos comentado por el problema de los motores. Por lo tanto según la carga de las baterías, hacemos que éste tenga un PWM adecuado para un tiempo de giro estimado. De esta manera el robot girará muy cerca de los 90° (por exceso o por defecto). (Ver diagrama de flujo Función Giro izquierda Total 3.3.5.1 Funciones de Control de motores). Después de este giro el robot avanzará hasta encontrar otra vez una referencia a la izquierda. A partir de entonces volverá a hacer la función

Seguir Izquierda.

3.2.5. SERVOMOTOR

El siguiente problema sería tomar una decisión cuando tenemos un obstáculo o muro enfrente. Para ello solamente utilizaremos un sensor, pero haremos que éste gire para poder ver lo que el robot tiene a su derecha. Para ello utilizaremos un mini servomotor.

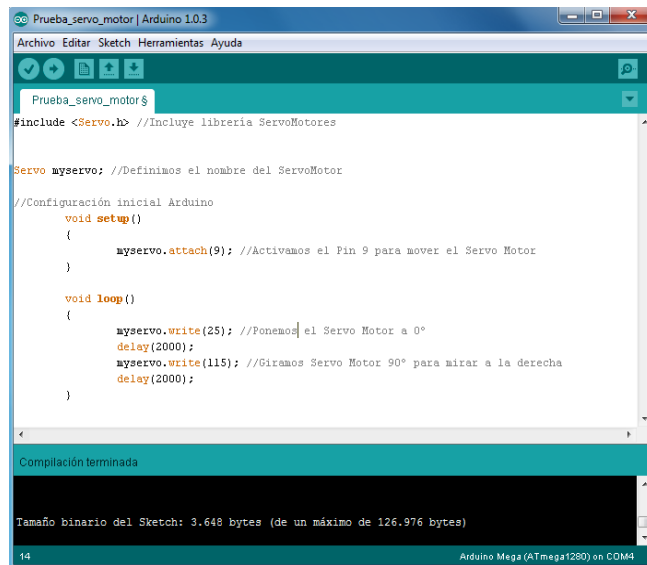
El servomotor elegido es el Micro Servo 9G de, cuyas características principales son:

- Velocidad sin carga: 0.12 segundos / 60 grados (4.8V)
- Rango de rotación: 180º
- Par de arranque: 1.6 kg / cm (4.8V)
- Rango de funcionamiento: 4.8V-6V
- Corriente de funcionamiento: menos de 500mA
- Tamaño: 22mmx12.5mmx29.5mm
- Peso: 9 gramos

Este servomotor se adapta perfectamente a lo que queremos. Tiene reducidas dimensiones, poco peso, gran par de arranque y un rango de rotación de al menos 90º para realizar nuestro giro.

Como hemos visto el servomotor funciona con PWM. Después de utilizar la librería <servo.h> y programar qué pin de Arduino está conectado a éste, tenemos que calibrar el giro.

De la misma manera que la calibración de los motores de continua no es perfecta, nos pasa lo mismo con los servomotores; por lo tanto tenemos que calibrar el ángulo de giro que nosotros queremos. En este caso necesitamos que gire desde 0º (sensor mirando enfrente) hasta 90º a la derecha (sensor mirando derecha). Aquí podemos ver la configuración y el programa para que el servomotor haga los movimientos deseados:



```

Prueba_servo_motor | Arduino 1.0.3
Archivo Editar Sketch Herramientas Ayuda
Prueba_servo_motor $
#include <Servo.h> //Incluye libreria ServoMotores

Servo myservo; //Definimos el nombre del ServoMotor

//Configuración inicial Arduino
void setup()
{
  myservo.attach(9); //Activamos el Pin 9 para mover el Servo Motor
}

void loop()
{
  myservo.write(25); //Ponemos el Servo Motor a 0°
  delay(2000);
  myservo.write(115); //Giramos Servo Motor 90° para mirar a la derecha
  delay(2000);
}

Compilación terminada

Tamaño binario del Sketch: 3.648 bytes (de un máximo de 126.976 bytes)

14 Arduino Mega (ATmega1280) on COM4

```

FIGURA 64: CÓDIGO DE PRUEBA PARA SERVOMOTOR

Entonces, encajando el sensor en el servomotor, podemos hacer que éste pueda ver si hay obstáculos enfrente y a la derecha, véase diagrama de flujo Función Mirar Enfrente 3.3.5.2 Funciones de movimiento y decisión. Y a partir de aquí tomaremos una decisión, ver diagrama de flujo Función Decisión de giro 3.3.5.2 Funciones de movimiento y decisión.

3.2.6. RELOJ REAL

Hemos incorporado un reloj real a nuestro robot, para que éste mantenga el día y la hora después de apagado.

El reloj real que hemos utilizado es el *Tiny RTC DS1307 Shield V2.0*.



FIGURA 65: RELOJ REAL TINY RTC DS1307

El DS1307 es un chip RTC de la empresa Dallas, que usa el protocolo I²C (ver apartado 3.4.3 Comunicación I2C) para comunicarse con SMC, esta interfaz es usada también por Arduino, por lo que es la mejor manera de comunicarlos entre sí. Tiene salidas programables, que pueden ser utilizadas para accionar LEDs, o para

accionar una interrupción, pero al usarlo es necesario utilizar algún tipo de fuente de alta potencia.

Por lo general sólo se utiliza la interfaz I2C para realizar la configuración básica del reloj / función de lectura. La batería es de pilas de botón (tipo CR1220).

Este módulo abarca el diminuto DS1307 módulo RTC, módulo de temperatura y AT24C32 módulo de almacenamiento. DS1307 y AT24C32 están conectados mediante bus I²C.

Las principales características de este módulo son:

- Fuente de alimentación: 5V DC y pila de botón.
- Interfaz I²C
- Tamaño: 16mm x 22mm x 23mm (tamaño casi igual que la pila de botón CR2032)
- Capacidad de memoria: 32 K bit
- IC: DS1307, AT24C32; DS1307 se puede utilizar por separado.
- Tiene la función de ajuste automático de segundo, minuto, día, mes y año.

Para la configuración del reloj con Arduino, primeramente tenemos que cargar un programa para que éste capte la fecha y la hora del ordenador desde donde se carga, además de utilizar las librerías <Wire.h> y <RTClib.h>



```
prueba_configuracion_reloj_real
#include <Wire.h>
#include "RTClib.h"
RTC_DS1307 RTC;

void setup () {
  Wire.begin(); // Inicia el puerto I2C
  RTC.begin(); // Inicia la comunicación con el RTC
  RTC.adjust(DateTime(__DATE__, __TIME__)); // Establece la fecha y hora
}

void loop () {
}
```

Guardado Terminado.

14 Arduino Mega (ATmega1280) on COM4

FIGURA 66: CONFIGURACIÓN INICIAL DE RELOJ REAL

Y aquí podemos ver el programa:

```

prueba_reloj_real | Arduino 1.0.3
Archivo Editor Sketch Herramientas Ayuda

prueba_reloj_real

#include <Wire.h> // Incluye la libreria Wire
#include "RTClib.h" // Incluye la libreria RTClib
RTC_DS1307 RTC; // Crea el objeto RTC

void setup () {

  Serial.begin(9600); // Establece la velocidad de datos del puerto serie
  Wire.begin(); // Establece la velocidad de datos del bus I2C
  RTC.begin(); // Establece la velocidad de datos del RTC
}

void loop () {

  DateTime now = RTC.now(); // Obtiene la fecha y hora del RTC

  Serial.print(now.year(), DEC);
  Serial.print('/');
  Serial.print(now.month(), DEC);
  Serial.print('/');
  Serial.print(now.day(), DEC);
  Serial.print(' ');
  Serial.print(now.hour(), DEC);
  Serial.print(':');
  Serial.print(now.minute(), DEC);
  Serial.print(':');
  Serial.print(now.second(), DEC);
  Serial.println();

  delay(1000); // La información se actualiza cada 1 seg.

}

Guardado Terminado.

10 Arduino Mega (ATmega1280) on COM4

```

FIGURA 67: CÓDIGO DE PRUEBA PARA RELOJ REAL

3.2.7. LCD

También queremos mostrar información al usuario mediante una pantalla LCD. Para ello elegimos una pantalla de LCD de 2x16, cuyas características principales son:

- Voltaje requerido: +5v
- Interface de 8 bits de bus de datos
- Retroiluminación con LEDs blancos
- Alto contraste
- 16 pines de conexión



FIGURA 68: PANTALLA LCD 2x16

Para el ajuste de contraste hemos puesto una resistencia de 2.2k Ω en serie a la salida del pin V0 de la LCD a GND.

Para la retroiluminación es necesario no alimentar directamente a 5v porque el LED se puede dañar; por lo tanto hemos conectado una resistencia de 10k Ω al ánodo de la LCD a Vcc.

Se puede ajustar el contraste y la retroiluminación según se quiera. Manipulando un potenciómetro, el usuario puede elegir qué resistencias quiere poner según el contraste y la retroiluminación que desee.

Cargamos un simple programa con Arduino, donde podemos ver cómo se realiza la configuración de la LCD. En este programa mostramos el mensaje de “Hola Mundo” además del tiempo transcurrido:

```

prueba_Hola_Mundo | Arduino 1.0.3
Archivo Editar Sketch Herramientas Ayuda
prueba_Hola_Mundo
Conexionado del circuito:
* LCD RS pin to digital pin 12
* LCD Enable pin to digital pin 11
* LCD D4 pin to digital pin 5
* LCD D5 pin to digital pin 4
* LCD D6 pin to digital pin 3
* LCD D7 pin to digital pin 2
* LCD R/W pin to ground
* 10K resistor:
* ends to +5V and ground
* wiper to LCD V0 pin (pin 3)

// incluimos la libreria
#include <LiquidCrystal.h>

// inicializamos la libreria con los números de los pins
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {
  //inicializamos el LCD con el numero de columnas y filas
  lcd.begin(16, 2);
  // imprimimos el mensaje en el LCD
  lcd.print("Hola Mundo");
}

void loop() {
  // seleccionamos la columna 0 y fila 1
  lcd.setCursor(0, 1);
  // imprimimos el número de segundos desde el reset
  lcd.print(millis()/1000);
}

Guardado Terminado.
Arduino Mega (ATmega1280) via COM4
  
```

FIGURA 69: CÓDIGO DE PRUEBA PARA PANTALLA LCD

Como podemos observar, la programación del LCD no es complicada, simplemente tenemos que elegir las columnas y las filas donde queremos que aparezca el mensaje, y después mediante `lcd.print()`, podemos imprimir el mensaje que queramos.

También se pueden hacer dibujos mediante la LCD. Nosotros hemos creado un dibujo que simula una batería. Dependiendo de la carga que tenga la batería de los motores ésta se dibujará de una u otra manera.

En nuestra LCD vamos a mostrar la siguiente información:

MENSAJE DE BIENVENIDA

El mensaje mostrado al principio del programa: DEP.ELECTRONICA UVA.

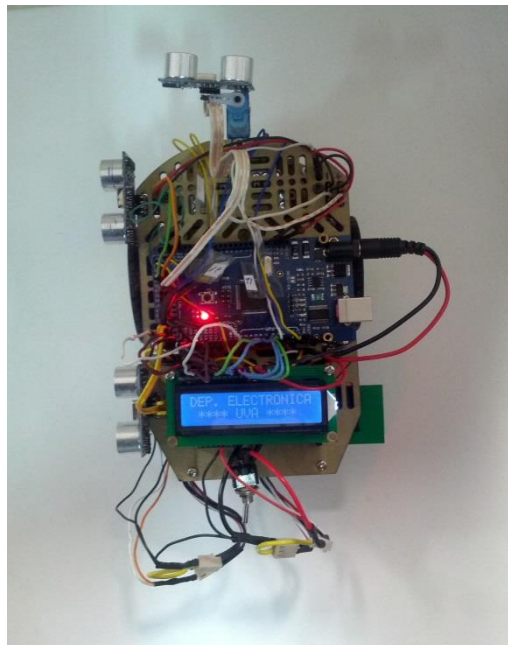


FIGURA 70: MENSAJE DE BIENVENIDA EN EL ROBOT

MENSAJES DE ERROR

Se muestran diferentes errores, al no insertar la tarjeta SD al inicio del programa y ante la falta de batería de motores o de Arduino.

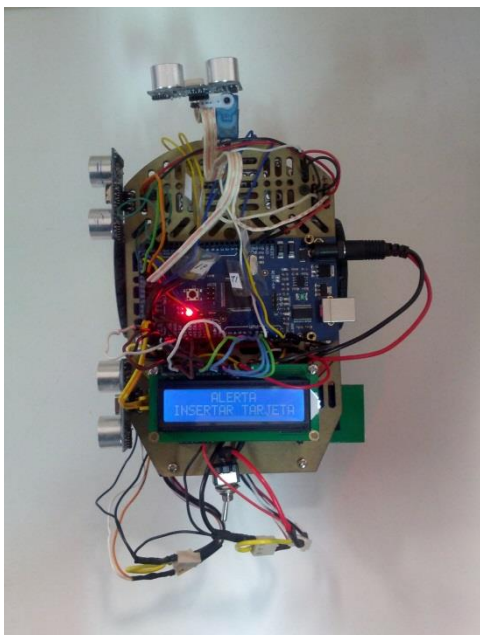


FIGURA 71: MENSAJE DE ALERTA POR FALTA DE TARJETA EN ROBOT

MENSAJES DE INFORMACIÓN

Se muestra información sobre lo que está haciendo el programa: los diferentes giros, ajustes, funciones, etc...

INFORMACIÓN SOBRE LA BATERÍA

Se muestra información sobre el estado de la batería mediante un dibujo de ésta, además de avisar cuando está baja.

FECHA Y HORA

Se muestra el día de la semana, la fecha y la hora.

3.3. DISEÑO SOFTWARE

En este apartado se va a explicar cómo se ha realizado la programación de cada una de las partes del programa.

3.3.1. INCLUSIÓN DE LIBRERÍAS

Lo primero de todo es incluir en el programa las diferentes librerías que se van a utilizar a lo largo de él. En nuestro programa incluimos las siguientes:

- `#include <SD.h> //Incluye librería lector SD`
- `#include <Servo.h> //Incluye librería Servo Motores`
- `#include <LiquidCrystal.h> // Incluye librería LCD`

- `#include <Wire.h> // Incluye la librería Wire`
- `#include "RTCLib.h" // Incluye la librería RTCLib`

3.3.2. CONFIGURACIÓN DE PINES

Se realiza la configuración de todos los pines del Arduino. Dependiendo de la necesidad de cada dispositivo, éstos irán conectados a un pin o a otro. Como ya sabemos Arduino tiene pines especializados para diferentes acciones. En nuestro caso los más importantes son:

Control de motores

Tanto, para el control de los motores de CC como para el control del servomotor es necesario, la introducción de PWM. En este caso hemos utilizado los siguientes pines:

- `int speedPin1 = 12; // Enable del L293D Control motor CC (PWM)`
- `int speedPin2 = 13; // Enable del L293D Control motor CC (PWM)`
- `myservo.attach(9); // Control servomotor (PWM)`

Lector de tarjetas

Para la placa Arduino Mega los pines de comunicación SPI son los siguientes:

Pin 50 → MISO

Pin 51 → MOSI

Pin 52 → SCK

Pin 53 → SS

Reloj Real

Para la placa Arduino Mega los pines de comunicación I2C son los siguientes:

Pin 20 → SDA

Pin 21 → SCL

3.3.3. DEFINICIÓN DE VARIABLES

Se realiza la definición de las variables que componen el programa. También se definen los dibujos de la batería para después imprimirlos en la LCD. En los Anexos se puede ver la figura realizada para mostrar la batería en la LCD.

3.3.4. CONFIGURACIÓN INICIAL DE ARDUINO

Se realiza la configuración inicial de Arduino. Esta configuración sólo se realizará una vez cuando comience el programa.

CONFIGURACIÓN DE LA COMUNICACIÓN SERIE

Se configura la velocidad de la comunicación serie con el ordenador.

CONFIGURACIÓN I2C

Se configura la velocidad de datos del bus I2C y del RTC.

CONFIGURACIÓN INICIAL DE LA LCD

Se configura la LCD, se crean los caracteres para mostrar el dibujo de la batería y se muestra el mensaje de bienvenida.

CONFIGURACIÓN DE LAS INTERRUPCIONES

Se configuran las interrupciones.

COMPROBACIÓN ESTADO DE BATERÍAS

Se comprueba el estado de las baterías por si alguna de ellas no tiene la carga adecuada.

CONFIGURACIÓN DEL SERVOMOTOR

Se configura el servomotor y se pone en posición inicial.

INICIALIZACIÓN DE LA TARJETA SD

Se inicializa la tarjeta SD y se muestra un mensaje de error si ésta no se encuentra.

CONFIGURACIÓN DE LOS MOTORES

Se configuran los pines de los motores y su velocidad, y se activa el RPA para el control de éstos.

3.3.5. BLOQUES DE FUNCIONES

A continuación se muestran los diagramas de flujo de las diferentes funciones del programa.

3.3.5.1. FUNCIONES DE CONTROL DE MOTORES

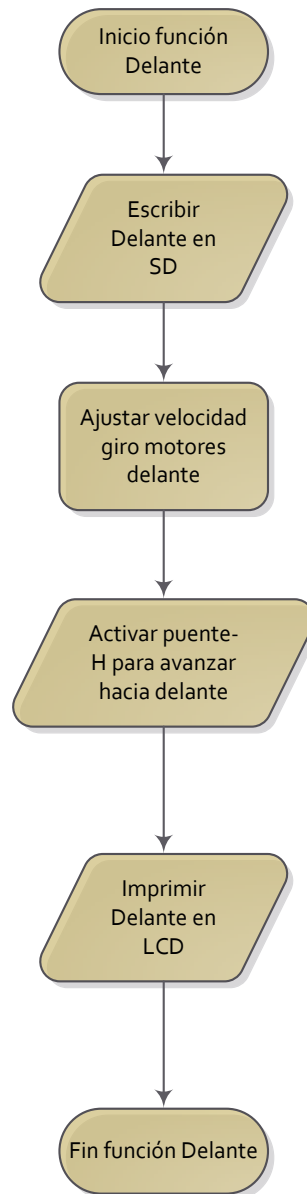
Función Delante

El robot gira las ruedas hacia delante.

Entradas:

Salidas: Escribir en SD, Activar puente H para avanzar hacia adelante, Imprimir 'Delante' en LCD.

Proceso: Después de escribir 'Delante' en la SD se ajusta la velocidad de giro de los motores para avanzar hacia adelante, se activa el puente H para avanzar hacia adelante y se imprime 'Delante' en la LCD.



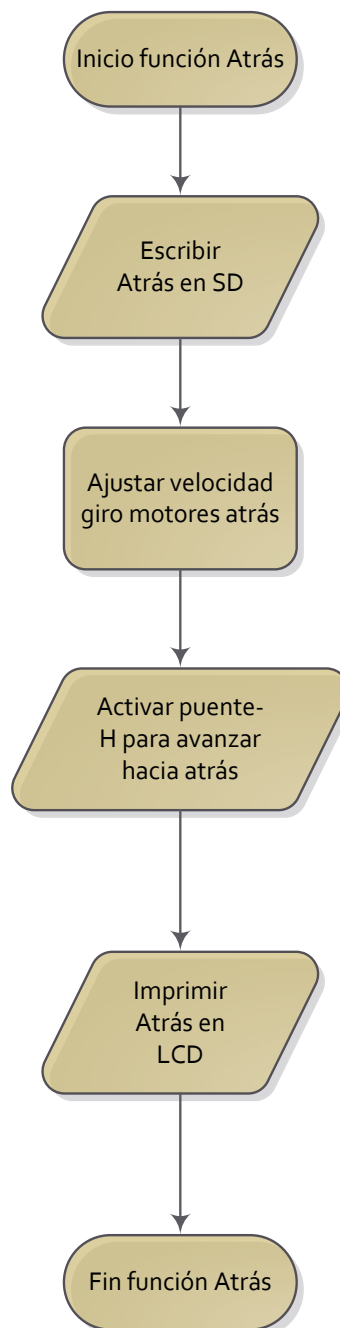
Función Atrás

El robot gira las ruedas hacia atrás

Entradas:

Salidas: Escribir 'Atrás' en SD, Activar puente H para avanzar hacia atrás, Imprimir 'Atrás' en LCD.

Proceso: Después de escribir 'Atrás' en la SD se ajusta la velocidad de giro de los motores para avanzar hacia atrás, se activa el puente H para avanzar hacia atrás y se imprime 'Atrás' en la LCD.



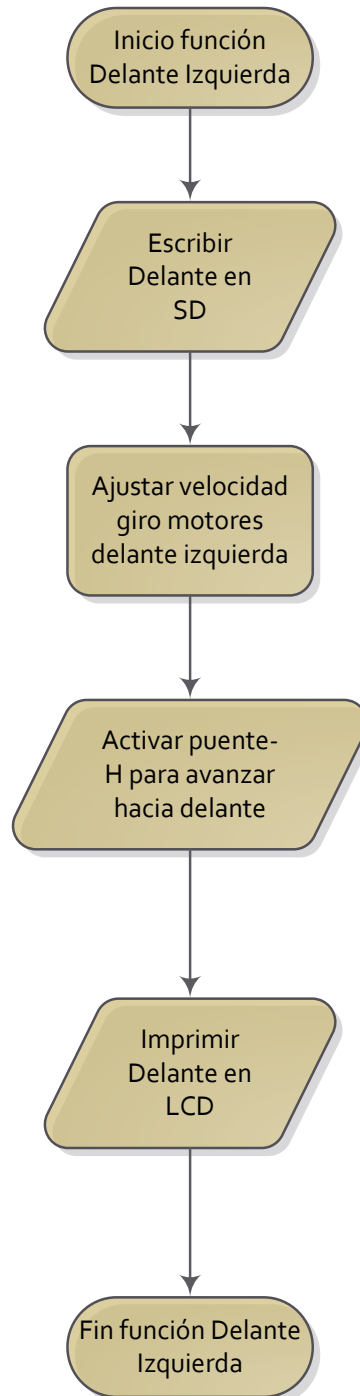
Función Corregir Izquierda

El robot avanza hacia adelante, ajustándose a la izquierda.

Entradas:

Salidas: Escribir 'Delante' en SD, Activar puente H para avanzar hacia adelante, Imprimir 'Delante' en LCD.

Proceso: Después de escribir 'Delante' en la SD se ajusta la velocidad de giro de los motores para corregir hacia adelante izquierda, se activa el puente H para avanzar hacia adelante y se imprime 'Delante' en la LCD.



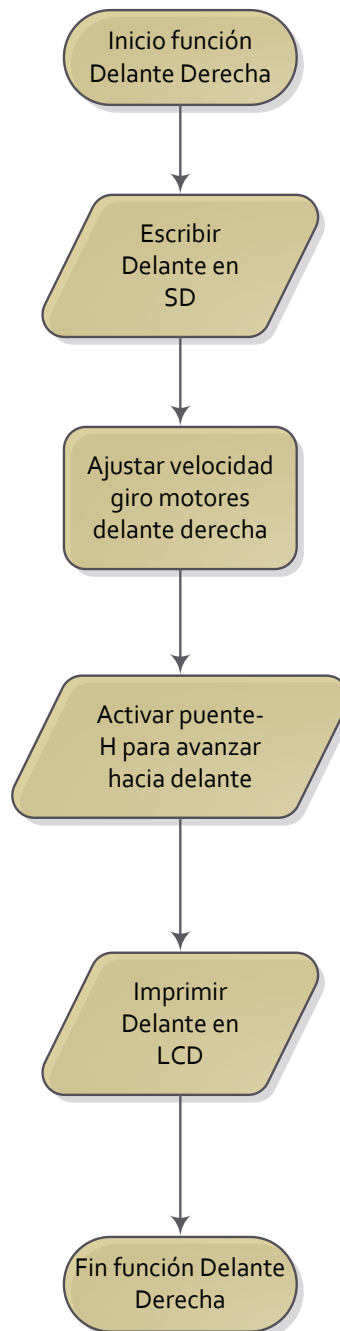
Función Corregir Derecha

El robot avanza hacia adelante, ajustándose a la derecha.

Entradas:

Salidas: Escribir 'Delante' en SD, Activar puente H para avanzar hacia adelante, imprimir 'Delante' en LCD.

Proceso: Después de escribir 'Delante' en la SD se ajusta la velocidad de giro de los motores para corregir hacia adelante a la derecha, se activa el puente H para avanzar hacia adelante y se imprime 'Delante' en la LCD.



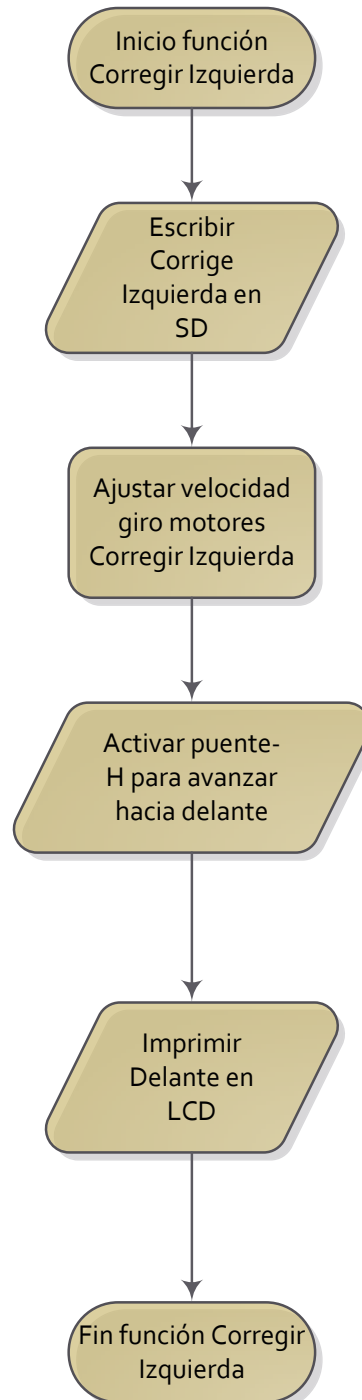
Función Giro Izquierda

El robot avanza hacia la izquierda corrigiendo su posición.

Entradas:

Salidas: Escribir 'Corrige Izquierda' en SD, activar puente H para avanzar hacia adelante, imprimir 'Delante' en LCD.

Proceso: Después de escribir 'Corrige Izquierda' en la SD se ajusta la velocidad de giro de los motores para girar a la izquierda, se activa el puente H para avanzar hacia adelante y se imprime 'Delante' en la LCD.



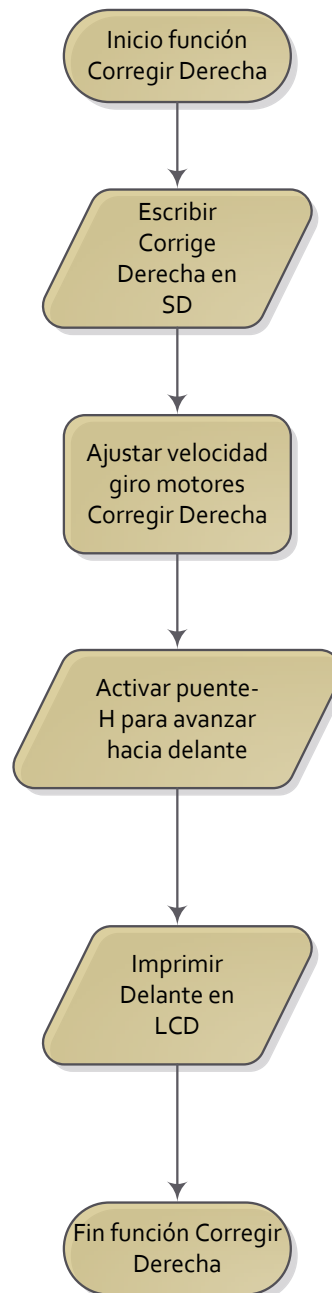
Función Giro Derecha

El robot avanza hacia la derecha corrigiendo su posición.

Entradas:

Salidas: Escribir 'Corrige Derecha' en SD, Activar puente H para avanzar hacia adelante, Imprimir 'Delante' en LCD.

Proceso: Después de escribir 'Corrige Derecha' en la SD se ajusta la velocidad de giro de los motores para girar a la derecha, se activa el puente H para avanzar hacia adelante y se imprime 'Delante' en la LCD.



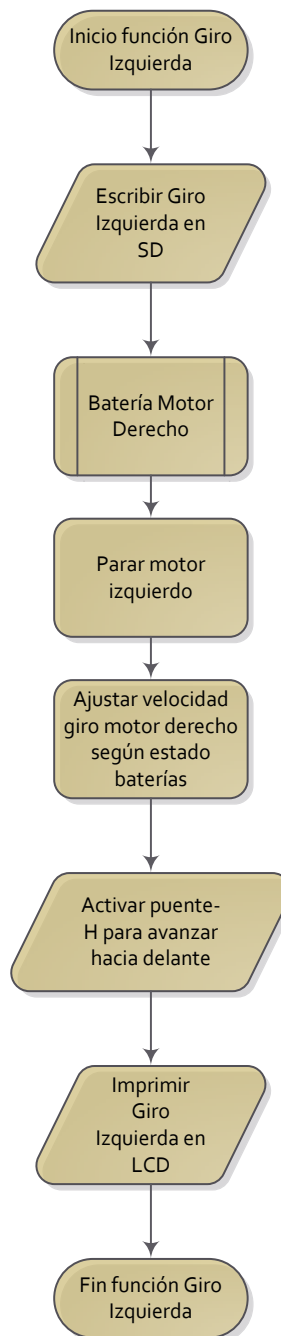
Función Giro Izquierda Total

El robot hace un giro total a la izquierda.

Entradas:

Salidas: Escribe 'Giro Izquierda' en SD, para el motor derecho, activa puente H para avanzar hacia adelante, imprime 'Giro Izquierda' en LCD

Proceso: Se escribe 'Giro Izquierda' en SD. Para el motor izquierdo. Después de comprobar el estado de las baterías ajusta la velocidad del motor derecho. Activa este motor para avanzar hacia adelante. Imprime 'Giro izquierda' en LCD.



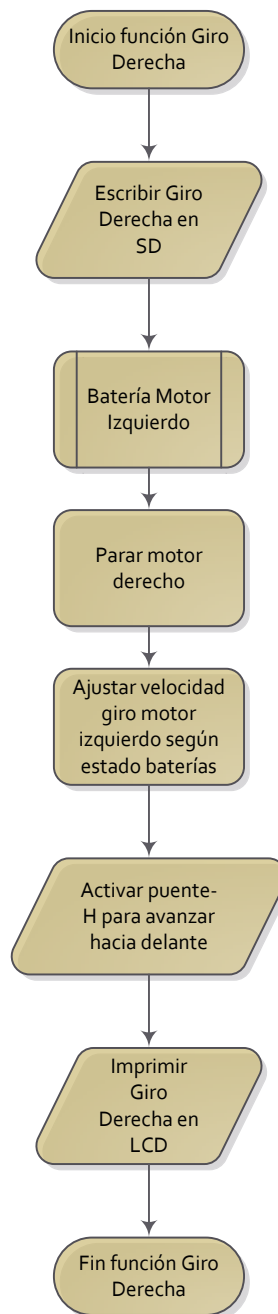
Función Giro Derecha Total

El robot hace un giro total a la derecha.

Entradas:

Salidas: Escribe 'Giro Derecha' en SD, para el motor derecho, activa puente H para avanzar hacia adelante, imprime 'Giro Derecha' en LCD

Proceso: Se escribe 'Giro Derecha' en SD. Para el motor derecho. Después de comprobar el estado de las baterías ajusta la velocidad del motor izquierdo. Activa este motor para avanzar hacia adelante. Imprime 'Giro derecha' en LCD.



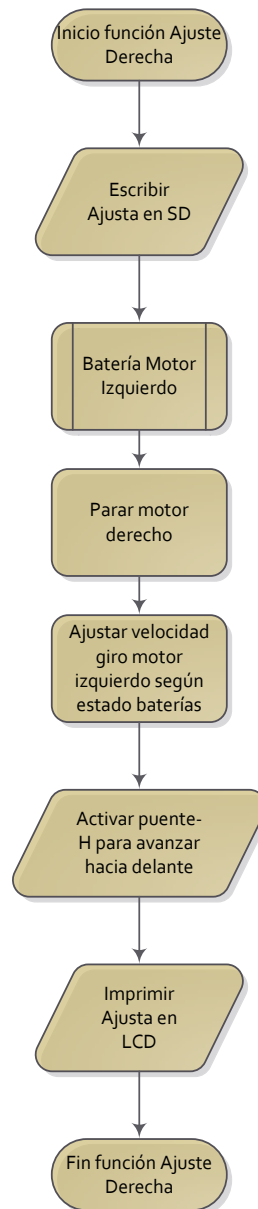
Función Ajuste Derecha

El robot se ajusta hacia la derecha, de forma paralela a la pared.

Entradas: Recibe información de la función Batería motor izquierdo.

Salidas: Escribe 'Ajusta' en SD, para el motor derecho, activa puente H para avanzar hacia adelante, imprime 'Ajusta' en LCD

Proceso: Se escribe 'Ajusta' en SD. Para el motor derecho. Después de comprobar el estado de las baterías ajusta la velocidad del motor izquierdo. Activa este motor para avanzar hacia adelante. Imprime 'Ajusta' en LCD.



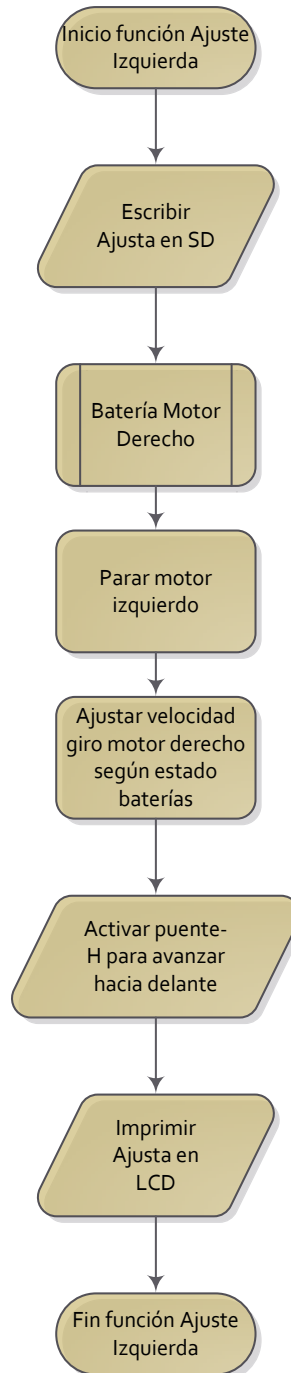
Función Ajuste Izquierda

El robot se ajusta hacia la izquierda, de forma paralela a la pared.

Entradas: Recibe información de la función Batería motor derecho.

Salidas: Escribe 'Ajusta' en SD, para el motor izquierdo, activa puente H para avanzar hacia adelante, imprime 'Ajusta' en LCD

Proceso: Se escribe 'Ajusta' en SD. Para el motor izquierdo. Después de comprobar el estado de las baterías ajusta la velocidad del motor derecho. Activa este motor para avanzar hacia adelante. Imprime 'Ajusta' en LCD.



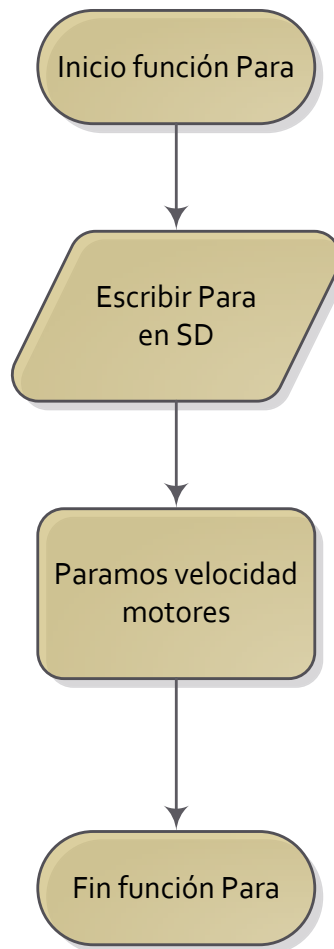
Función Para

El robot para sus motores.

Entradas:

Salidas: Grabar 'Para' en SD, activar los pines de control de los motores para que éstos se queden parados.

Proceso: Se graba 'Para' en la tarjeta SD y se activan los pines de control de los motores para que éstos se queden parados.



3.3.5.2. FUNCIONES DE MOVIMIENTO Y DECISIÓN

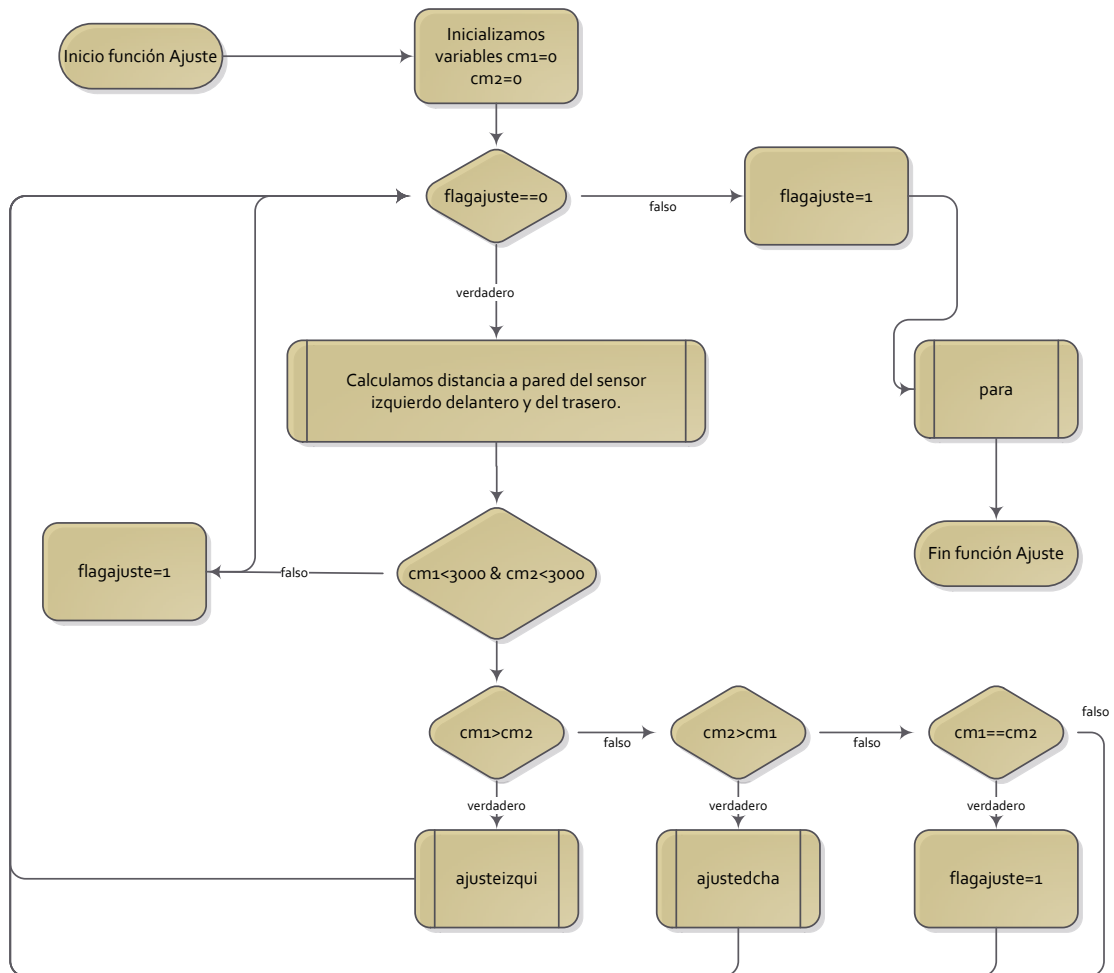
Función ajuste

Esta función hace que el robot se ajuste con la pared mediante sus dos sensores izquierdos

Entradas: Distancia del sensor izquierdo delantero (cm1), distancia del sensor izquierdo trasero (cm2), flagajuste.

Salidas: ajusteizquierda, ajustederecha, flagajuste, para

Proceso: El robot ajusta a la izquierda o a la derecha para ponerse paralelo a la pared; esto lo hace comprobando con los sensores de la izquierda (cm1 y cm2). Cuando los sensores marcan lo mismo, el robot se para y pone el flagajuste a 1.



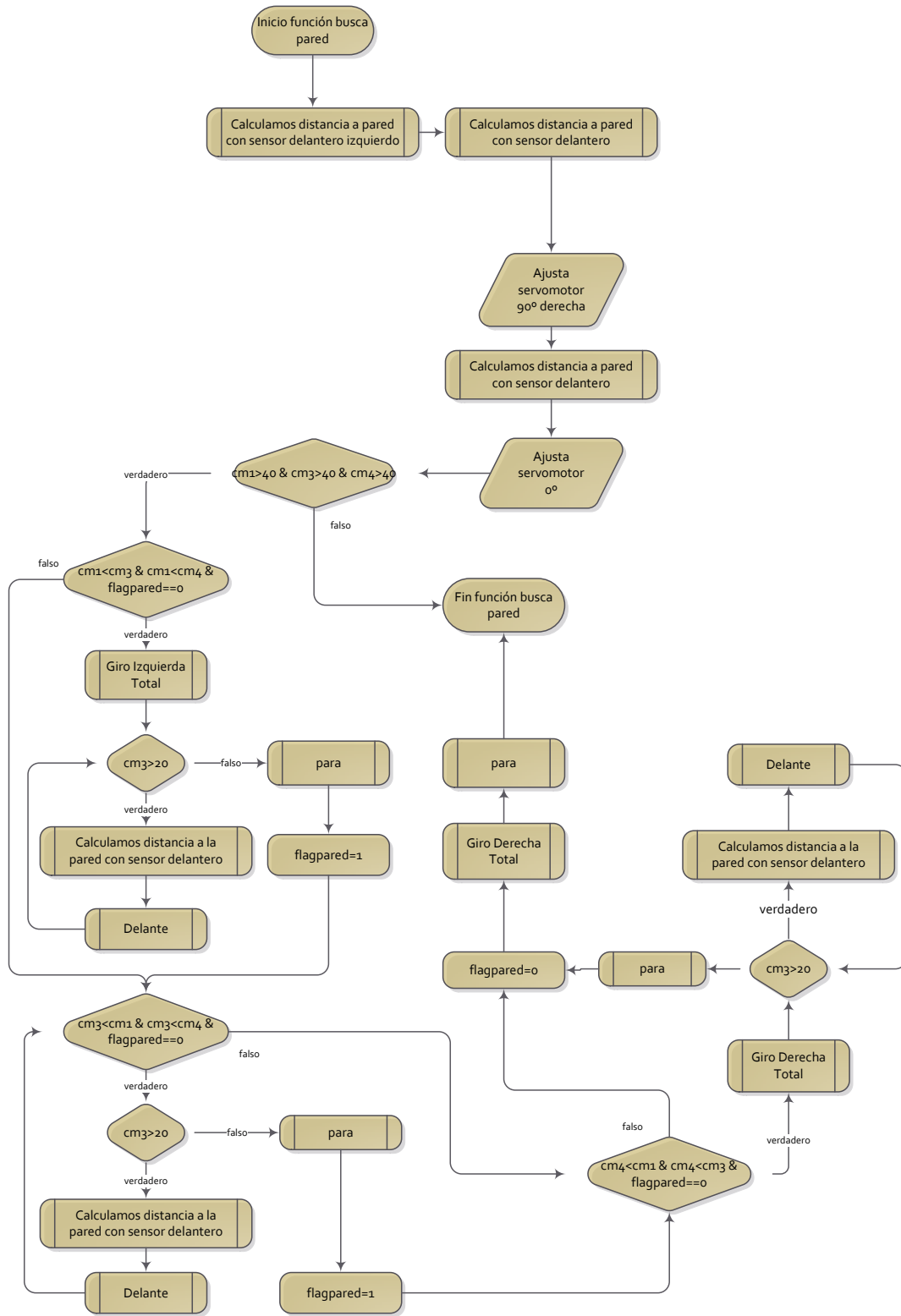
Función Busca Pared

Cuando el robot está a más de 40cm de cualquier pared, éste busca la más cercana, se aproxima a ella y se pone en posición de 'Seguir Izquierda'.

Entradas: Distancia del sensor izquierdo delantero (cm1), distancia del sensor delantero (cm3), distancia del sensor delantero mirando a la derecha (cm4).

Salidas: Ajusta servo a 0º y 90º a la derecha. Giro izquierda total, para, delante, giro derecha total.

Proceso: El robot mira si hay alguna pared más cerca de 40cm en alguna de las direcciones. Si no es así mira cuál de todas (frente, derecha e izquierda) está más cerca de su posición. Según la que esté más cerca toma la decisión de acercarse a ella (girando si es necesario) hasta que se encuentra con ella de frente, después girará a la derecha y así estará con sus sensores en posición de seguir la pared izquierda.



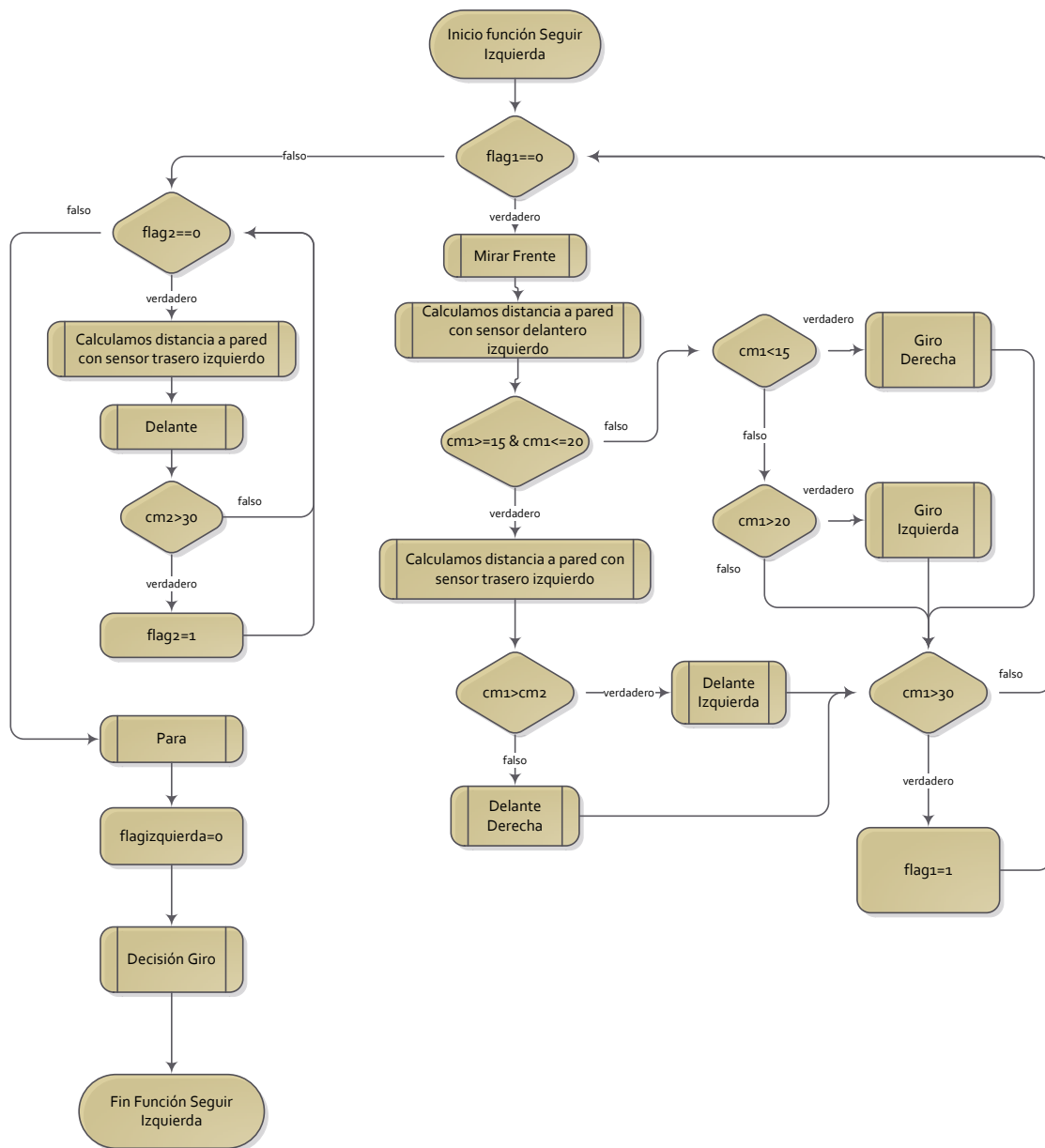
Función Seguir Izquierda

Esta función hace que el robot siga una pared que se encuentra a la izquierda.

Entradas: flag1, flag2, distancia del sensor izquierdo delantero (cm1), distancia del sensor izquierdo trasero (cm2).

Salidas: delante, para, giro derecha, giro izquierda, delante izquierda, delante derecha, flagizquierda.

Proceso: Dependiendo del flag1 y del flag2 el robot sigue la pared de la izquierda; dependiendo de su distancia a la pared el robot avanza hacia adelante, corrigiendo a la izquierda, corrigiendo a la derecha, girando a la izquierda o a la derecha. Además, siempre va mirando si tiene un obstáculo enfrente.



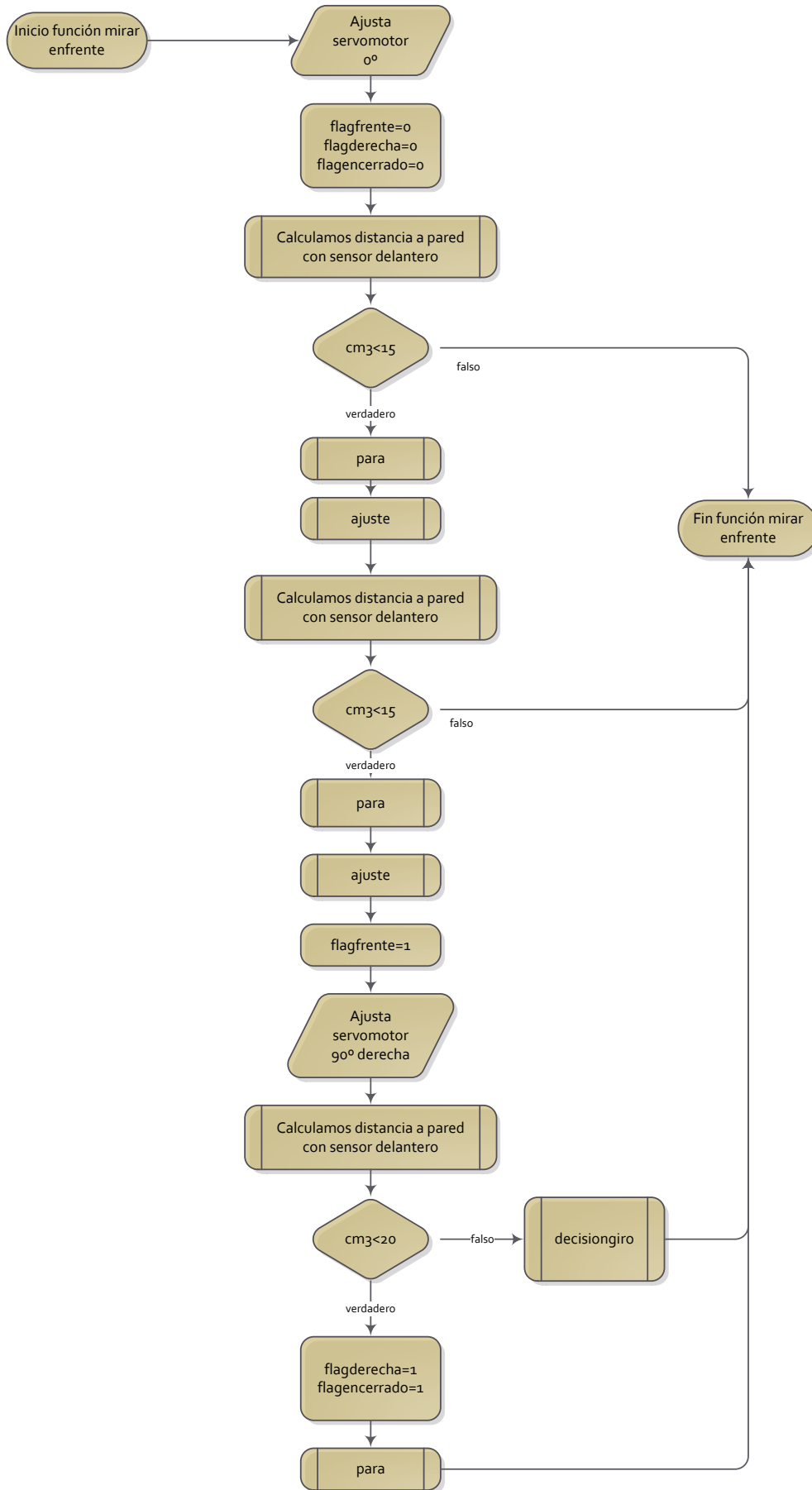
Función Mirar Enfrente

Esta función hace que el robot mire para detectar un posible obstáculo enfrente. Si es así, también mira a la derecha. Después de tener toda la información toma una decisión.

Entradas: Distancia del sensor delantero (cm3)

Salidas: Ajusta el servomotor a 0º y a 90º a la derecha, flagfrente, flagderecha, flagencerrado

Proceso: Se ajusta el servomotor a 0º, se inicializan los flags, se comprueba por 2 veces que no hay obstáculo enfrente (por posibles derivas del robot). Se activa el flagfrente si hay obstáculo enfrente. Después de parar y ajustar el robot, el servo giro 90º a la derecha y se comprueba si hay obstáculo en ese lugar. Se llama a la función 'Decisión de Giro' y si el robot está encerrado se activan los flags y se llama a la interrupción 'Encerrado'



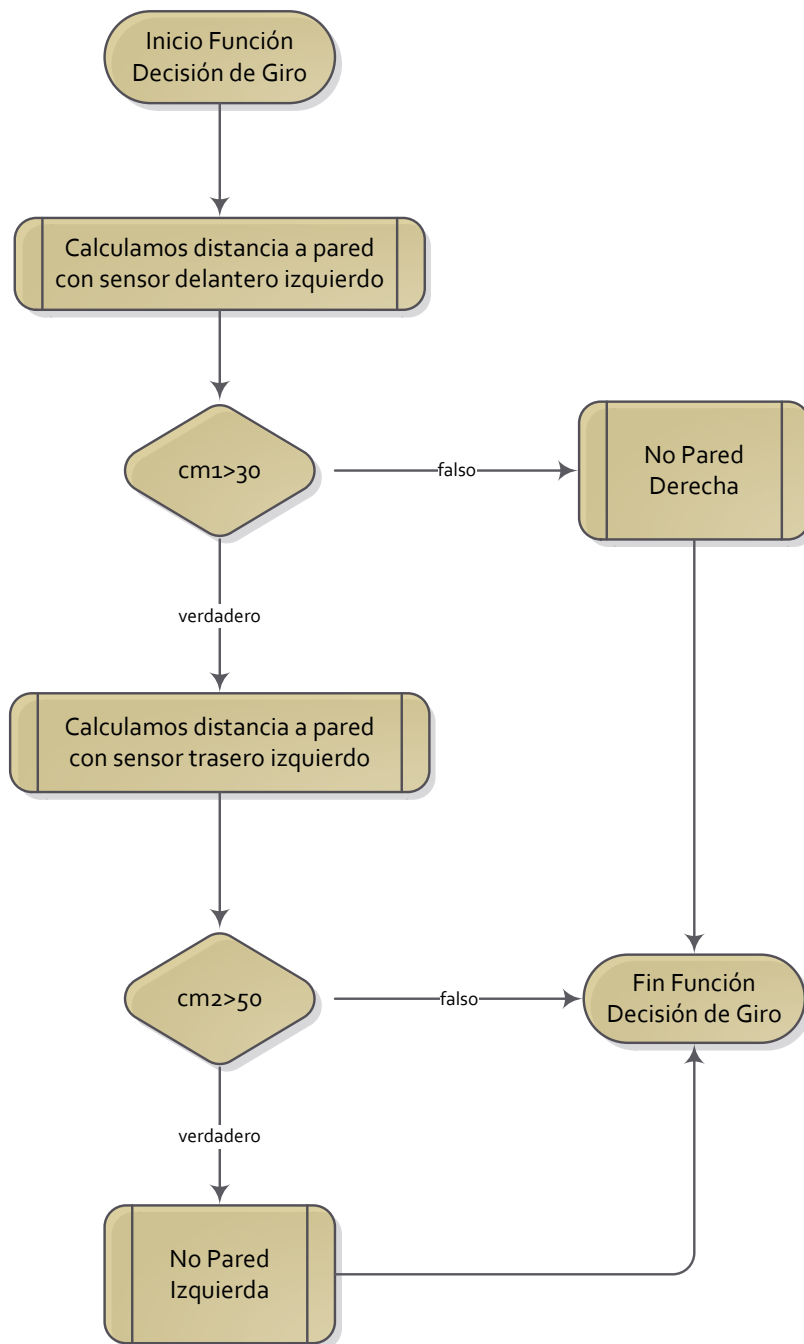
Función Decisión Giro

Toma una decisión para el giro.

Entradas: Distancia del sensor izquierdo delantero (cm1), distancia del sensor izquierdo trasero (cm2).

Salidas:

Proceso: Según la información de los ultrasonidos con cm1 y cm2, se llama a la función 'No Pared Derecha' o a 'No Pared Izquierda'.



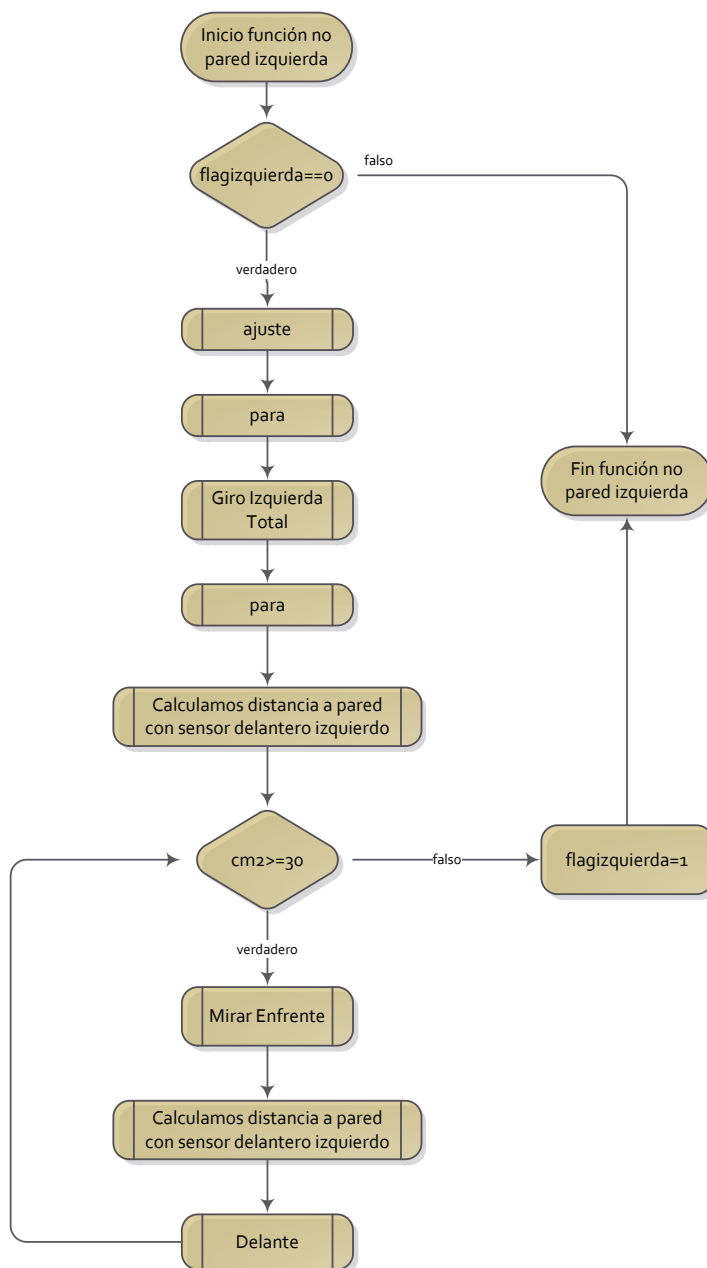
Función No Pared Izquierda

El robot realiza un giro a la izquierda cuando éste no encuentra obstáculo en ese lugar.

Entradas: Flagizquierda, distancia del sensor izquierdo trasero (cm2).

Salidas: El robot ajusta, para, gira total a la izquierda, para. Se activa 'flagizquierda' si es necesario.

Proceso: Se comprueba que 'flagizquierda' está a 0, si es así el robot ajusta, para, gira total a la izquierda, para, comprueba la distancia que hay enfrente y avanza hacia adelante para ver si encuentra un obstáculo; cuando acaba activa 'flagizquierda' a 1.



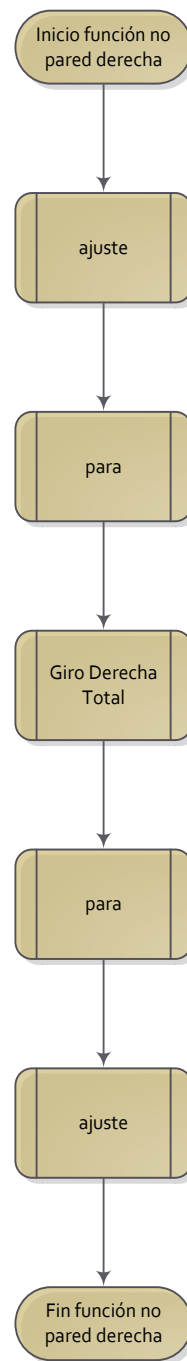
Función No Pared Derecha

El robot realiza un giro a la derecha al no encontrar obstáculo a la derecha y sí encontrarlo a la izquierda o enfrente.

Entradas:

Salidas: El robot se ajusta, para, hace un giro total a la derecha, para y vuelve a ajustar.

Proceso: Se entra en la función debido a que no hay pared a la derecha y hay obstáculos enfrente y a la izquierda. Por tanto el robot se ajusta, para, hace un giro total a la derecha, para y vuelve a ajustar.



3.3.5.3. FUNCIONES DE CONTROL DE BATERÍAS

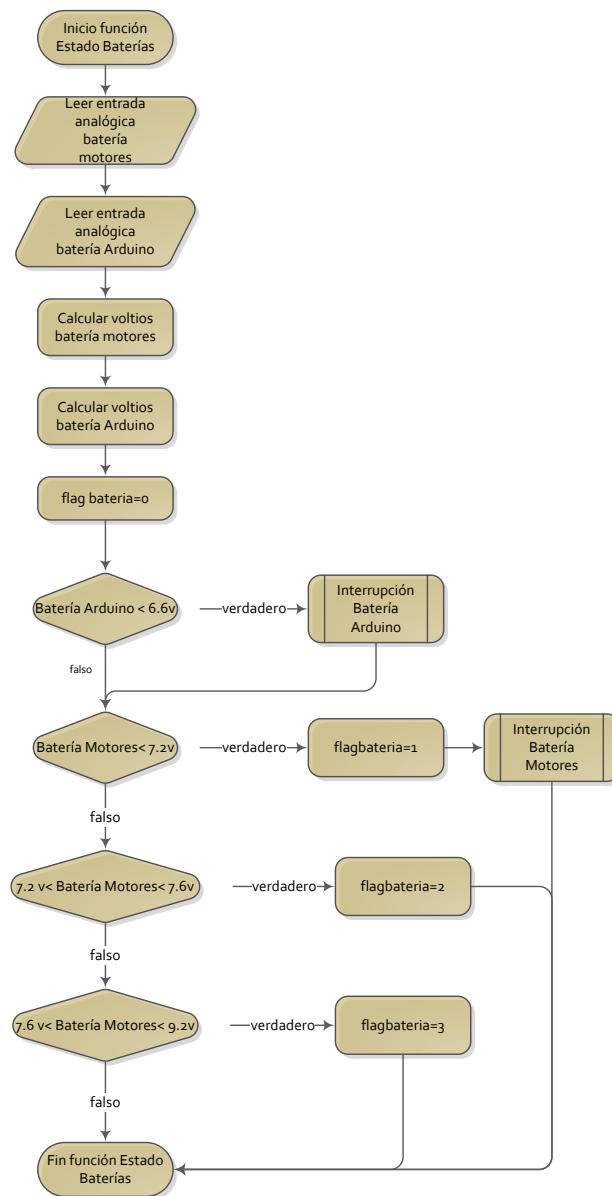
Función Estado Baterías

Comprueba el estado de las baterías y convierte los voltios a bits.

Entradas: Lee desde la entrada analógica de Arduino el estado de la batería de los motores y de la batería de Arduino.

Salidas: Llamadas a interrupciones por falta de batería. Activación de “flag batería” dependiendo del estado de ésta.

Proceso: A partir de las lecturas analógicas del estado de las baterías, se calculan los voltios de cada una. Si están por debajo del nivel óptimo se llama a las interrupciones. Si no dependiendo de la carga, se activa el “flag batería” de forma diferente.



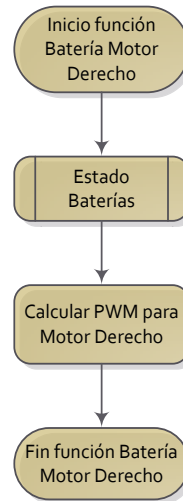
Función Batería Motor Derecho

Convierte los bits en PWM para el motor derecho.

Entradas: Utiliza la función estado baterías, para ver el estado de éstas.

Salidas:

Proceso: Utiliza la función estado baterías, para ver el estado de éstas. A partir de aquí se calcula el PWM para el Motor Derecho.

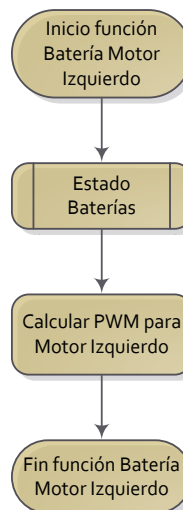
**Función Batería Motor Izquierdo**

Convierte los bits en PWM para el motor izquierdo.

Entradas: Utiliza la función estado baterías, para ver el estado de éstas.

Salidas:

Proceso: Utiliza la función estado baterías, para ver el estado de éstas. A partir de aquí se calcula el PWM para el Motor Izquierdo.



3.3.5.4. FUNCIONES DE SENSOR ULTRASONIDOS

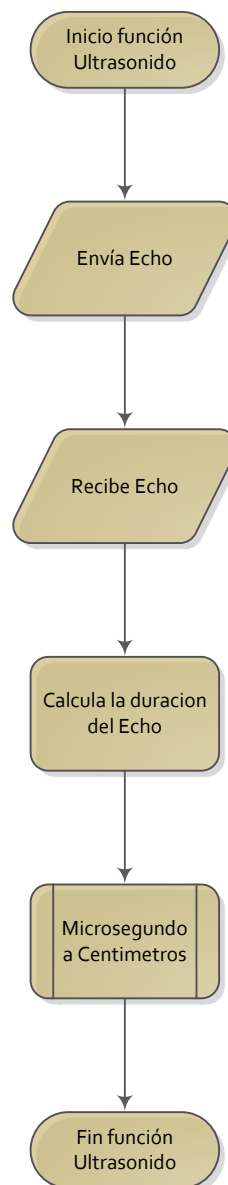
Función Ultrasonidos

Calcula la distancia del obstáculo.

Entradas: Señal Echo del ultrasonidos.

Salidas: La distancia al obstáculo devuelta por la función Microsegundos a Centímetros.

Proceso: El Ultrasonidos envía una señal Echo y espera a que vuelva; a partir de aquí se calcula la duración de este Echo. Esta duración la trata la función Microsegundos a Centímetros, que devuelve la distancia al obstáculo más próximo.



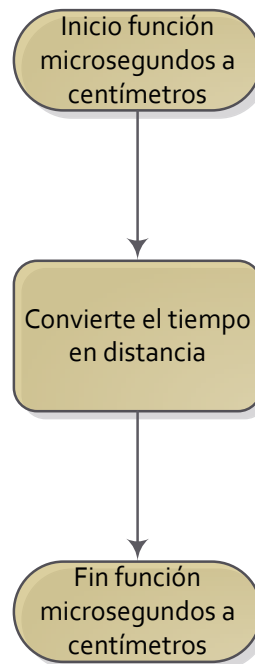
Función Microsegundos a Centímetros

Convierte el tiempo en distancia.

Entradas: La información recibida por el sensor de ultrasonidos.

Salidas: Distancia al obstáculo más próximo.

Proceso: Convierte el tiempo en distancia y lo devuelve.



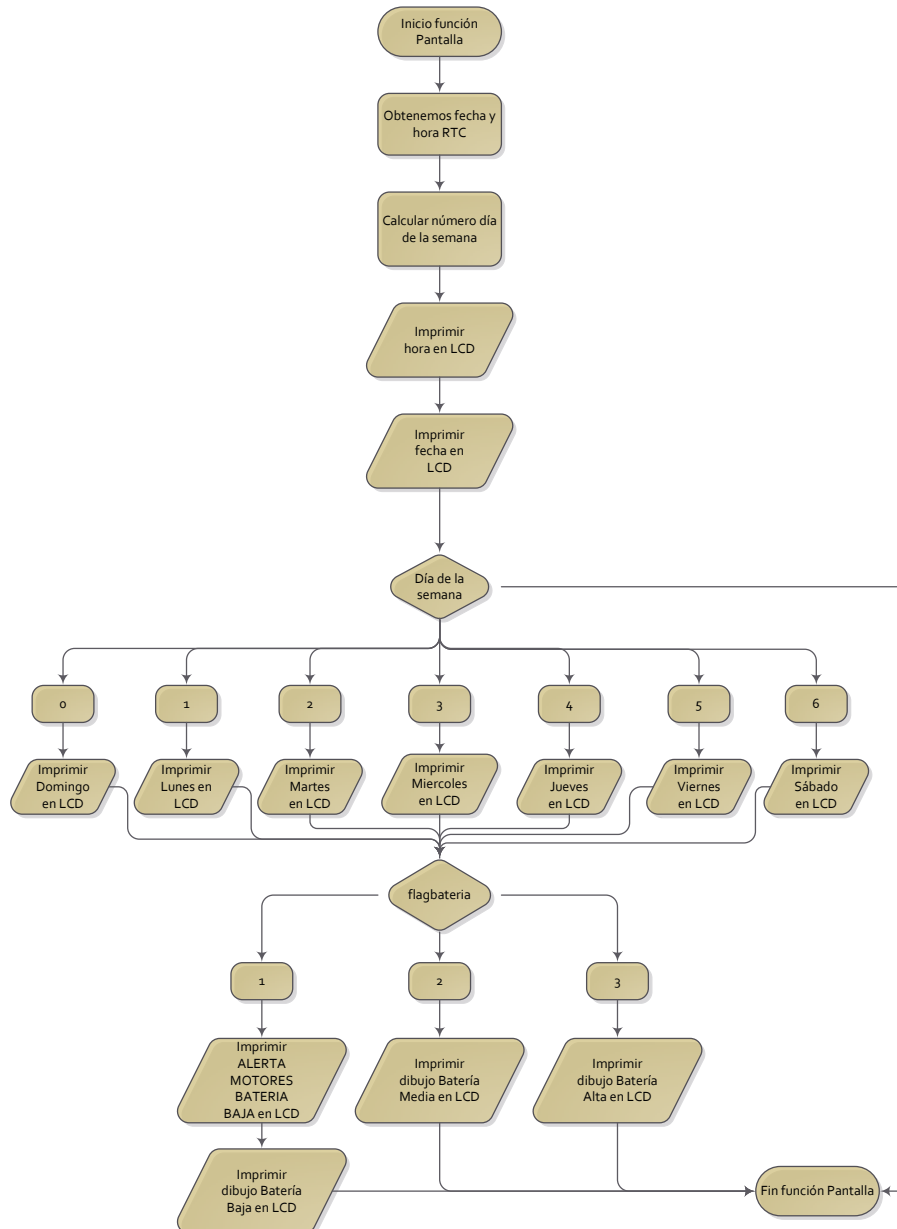
3.3.5.5. FUNCIÓN PANTALLA

Configura lo mostrado en la LCD. Fecha, hora, batería y mensajes de error.

Entradas: Fecha y hora desde la RTC. Flag batería.

Salidas: Mostrar fecha, hora y día de la semana en LCD. Mostrar dibujos de la batería en LCD y si fuera preciso mensaje de error por falta de batería.

Proceso: Se obtiene la fecha y la hora desde la RTC, a partir de aquí se calcula el día de la semana en que estamos y se imprime todo por pantalla. Además se comprueba el nivel de las batería mediante flagbatería, a partir de este flag se imprime por la pantalla el dibujo de la batería y si es necesario un mensaje de error por falta de batería de los motores.



3.3.6. INTERRUPCIONES

En nuestro programa tenemos 3 interrupciones diferenciadas.

Dos de estas interrupciones controlan las baterías de los motores y de Arduino. En cuanto una de estas baterías está por debajo de su nivel óptimo para el correcto funcionamiento de los dispositivos, se crea una interrupción. Esta interrupción mostrara el problema en la LCD, lo grabará en la SD y parará el programa y por tanto el robot.

Una tercera interrupción se ocupa del problema de que el robot se encuentre encerrado (obstáculos enfrente, izquierda y derecha). En este caso la interrupción mostrará el problema en la LCD, lo grabará en la SD y parará el programa y por tanto el robot.

A continuación se muestran los diagramas de flujo de las 3 interrupciones de nuestro programa:

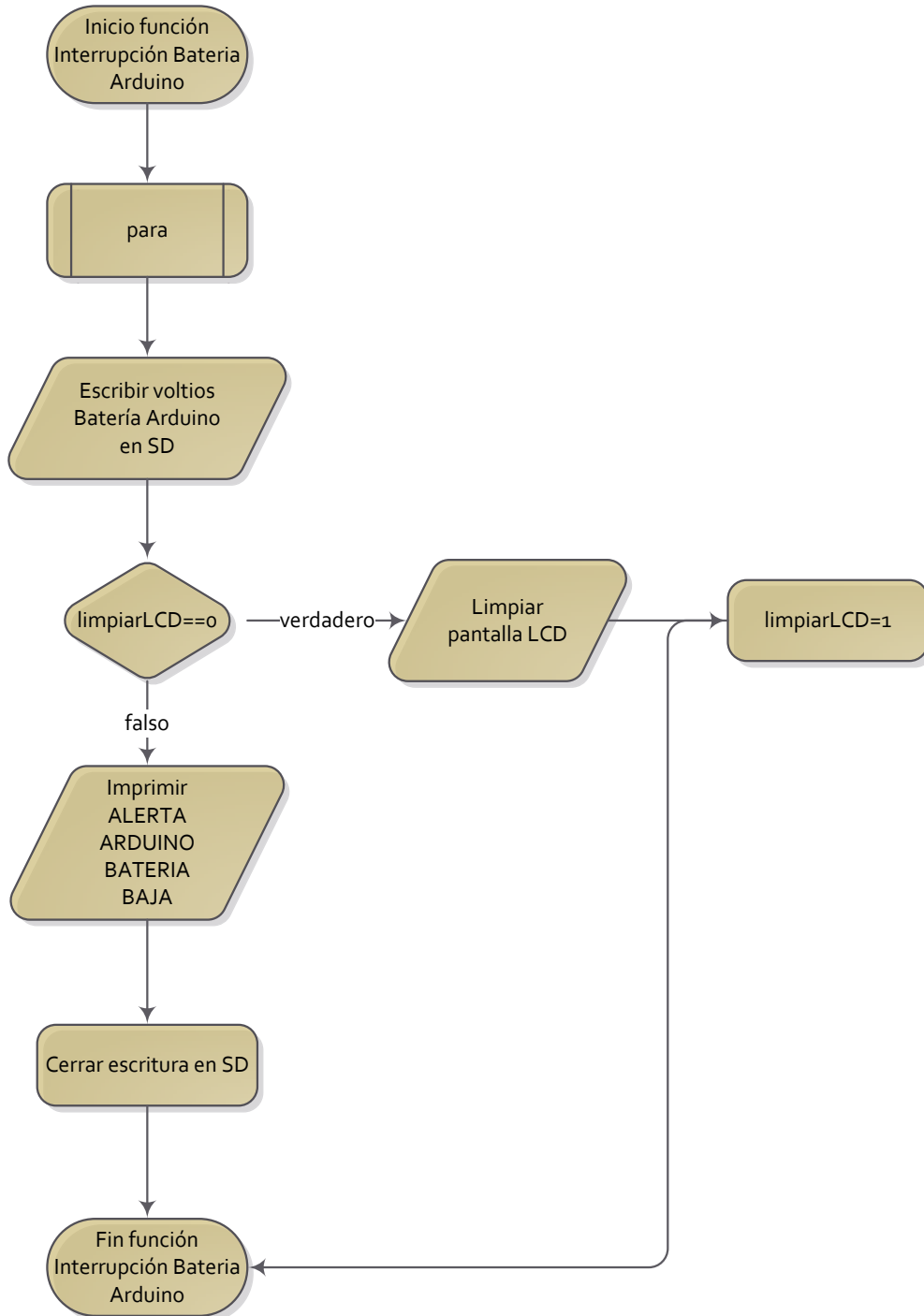
Interrupción Batería Arduino

Llama a la interrupción ante la falta de batería de Arduino y sus componentes.

Entradas:

Salidas: Graba en SD el nivel de las baterías del Arduino. Mensaje de alerta en SD y LCD. Parar el robot.

Proceso: La función para el robot. Graba en SD el nivel de las baterías del Arduino y muestra un mensaje de alerta en LCD.



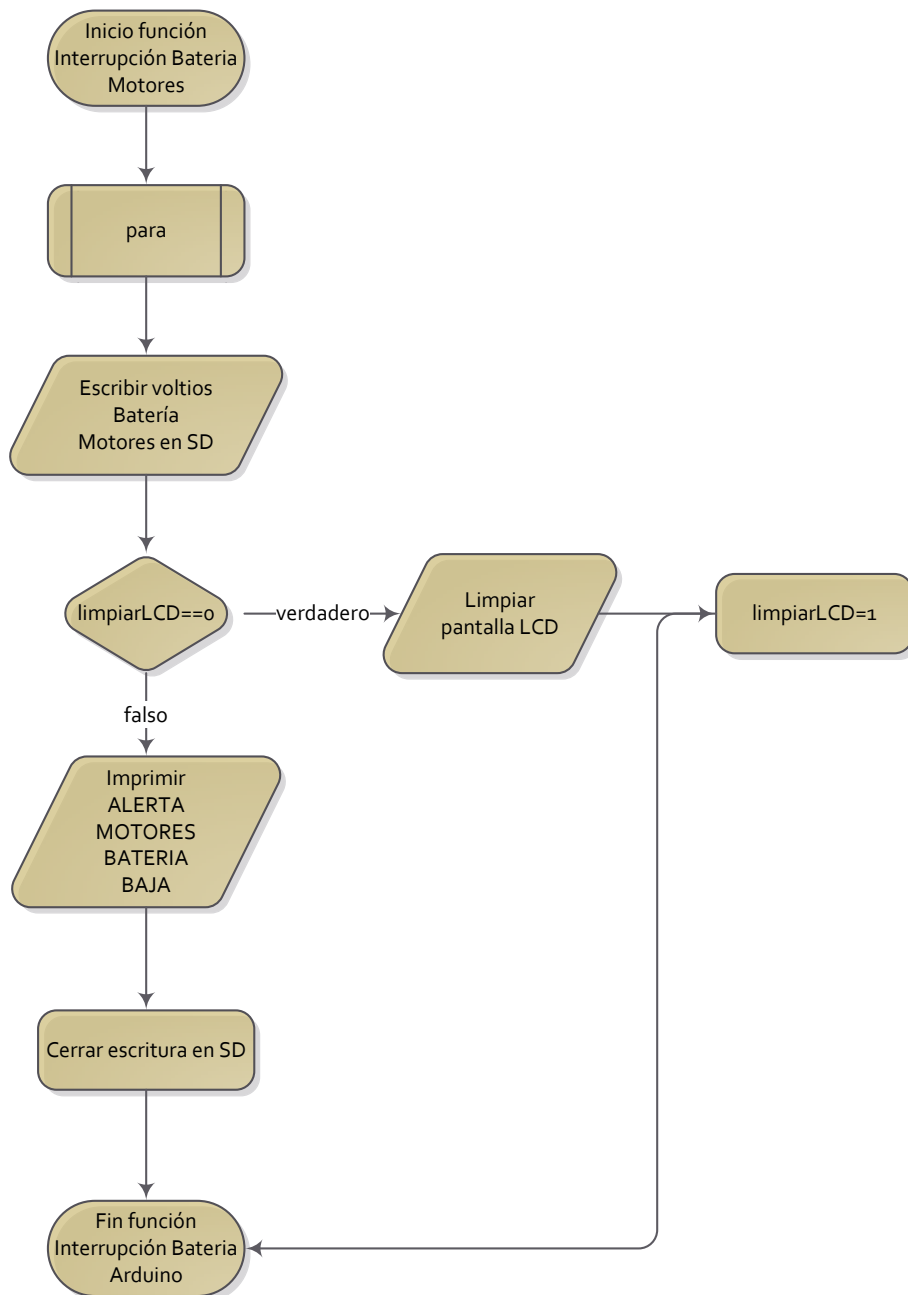
Interrupción Batería Motores

Llama a la interrupción ante la falta de batería de los motores.

Entradas:

Salidas: Graba en SD el nivel de las baterías de los motores. Mensaje de alerta en SD y LCD. Parar el robot.

Proceso: La función para el robot. Graba en SD el nivel de las baterías de los motores y muestra un mensaje de alerta en LCD.



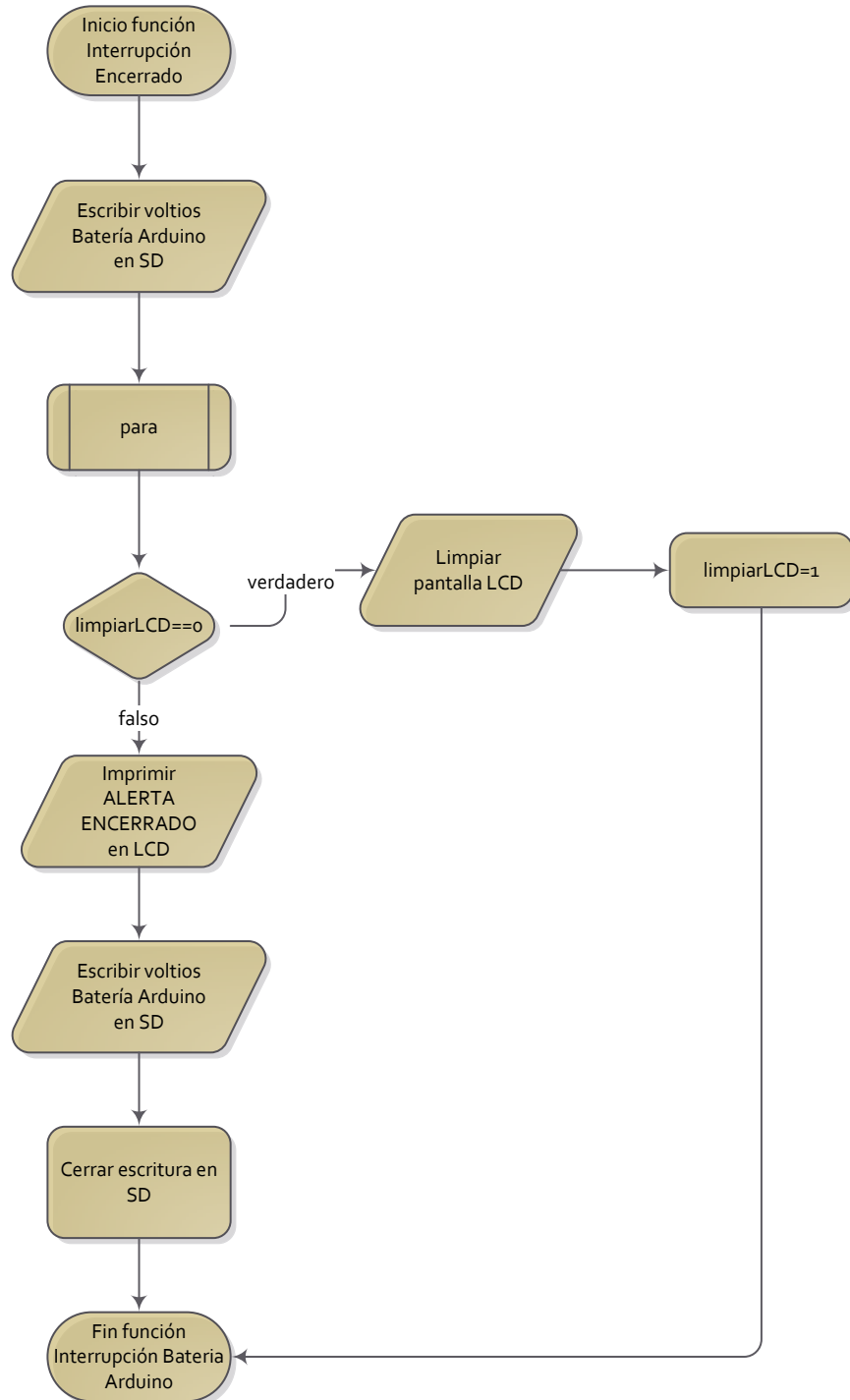
Interrupción Encerrado

Llama a la interrupción al encontrarse el robot encerrado.

Entradas:

Salidas: Grabar en SD el nivel de las baterías. Mensaje de alerta en SD y LCD. Parar el robot.

Proceso: La función para el robot. Graba en SD el nivel de las baterías además de un mensaje de alerta. Muestra un mensaje de alerta en LCD.



3.4. PROTOCOLOS DE COMUNICACIÓN

3.4.1. COMUNICACIÓN SERIE

La **comunicación serie** se basa en 2 cables que llamaremos Rx y Tx, que todo dispositivo de **comunicación serie** lleva. Éstos permiten entablar una conversación entre varios microprocesadores.

El microcontrolador manda las instrucciones mediante Tx y otro microcontrolador, que las interpreta, las recibe con el receptor Rx, que las interpreta.

La **comunicación** que se realiza es bidireccional, es decir, tanto un microcontrolador puede mandar instrucciones al otro como puede escucharlas; es más, dependiendo de la tecnología que uses, puedes trabajar desde ciertos terminales u otros.

Este método es conocido como *Comunicación Serial*, e implica cambiar el voltaje de la conexión eléctrica entre el *Transmisor* y el *Receptor* a una velocidad específica. Cada Intervalo de tiempo representa un *bit* de información. El transmisor cambia el voltaje para enviar un valor de entre 0 y 1, el cual representa el bit en cuestión, y el receptor lee si el voltaje es *high* o *low*.

Este ejemplo se basa en el *Asynchronous Serial Communication*, donde la velocidad de intercambio de bits es acordada de forma mutua y cronometrada de manera independiente entre el Transmisor y el Receptor.¹⁴

3.4.2. USB (UNIVERSAL SERIAL BUS)

El Bus en Serie Universal, más conocido por USB, es probablemente con diferencia la tecnología cableada más extendida para conectar entre sí toda clase de dispositivos.

Fue diseñado inicialmente para **estandarizar todo tipo de periféricos**, como ratones, teclados, cámaras digitales, smartphones, reproductores multimedia, etc. Hoy en día su campo de aplicación se extiende a cualquier dispositivo electrónico, desde automóviles a reproductores Blue-ray disc.

¹⁴ Blog DiegoTechnology.es - Diego Romano Ubalde

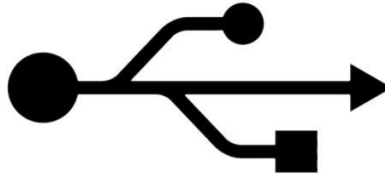


FIGURA 72: LOGOTIPO DE USB

El USB es un interfaz de tipo cable, que sirve a todas las necesidades de interconexión, permitiendo, conectar hasta 127 dispositivos por puerto, y que trabaja con tecnología de tipo Plug&Play y Hot-Plug, por lo que los periféricos que utilizan esta tecnología no necesitan de ninguna clase de configuración a la hora de ser conectados a, por ejemplo, un ordenador o una Smart TV. Además, al igual que con FireWire, no es necesario reiniciar el ordenador o apagar la televisión para hacer conexiones y desconexiones -Hot-Plug-.

El USB utiliza un conector de 4 polos, siendo la transferencia de datos llevada a cabo por medio de dos líneas de señales diferentes (D+, D-). Existe también una línea de tierra y una línea para la tensión de alimentación (VBus = 5V). La señal diferencial en el USB es al menos de 1V. No existe señal de frecuencia, sino que está se genera a partir del propio flujo de datos.

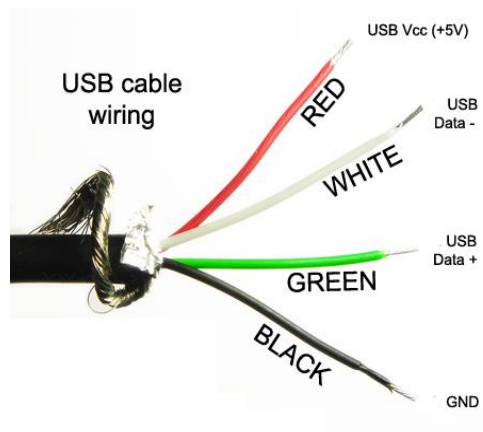


FIGURA 73: CABLEADO DE UN USB

La entrada máxima de corriente a través del bus es de 500 mA, de modo que algunos dispositivos requieren una alimentación adicional para su funcionamiento. Esto no afecta a dispositivos estándar, como puede ser un ratón o un teclado, ya que estos reciben la tensión de servicio necesaria del propio ordenador a través del cable USB.

Cada unidad USB recibe una dirección propia, la cual es asignada por el host USB, siendo posible un máximo de 127.¹⁵

¹⁵ Tecnomatica: <http://lochnessh.wordpress.com/>

3.4.3. COMUNICACIÓN I2C

3.4.3.1. INTRODUCCIÓN

El **bus I2C**, un estándar que facilita la comunicación entre microcontroladores, memorias y otros dispositivos con cierto nivel de "inteligencia", sólo requiere de dos líneas de señal y un común o masa. Fue diseñado a este efecto por Philips y permite el intercambio de información entre muchos dispositivos a una velocidad aceptable de unos 100 Kbits por segundo, aunque hay casos especiales en los que el reloj llega hasta los 3,4 MHz.

La metodología de comunicación de datos del bus I2C es en serie y sincrónica. Una de las señales del bus marca el tiempo (pulsos de reloj) y la otra se utiliza para intercambiar datos.

Descripción de las señales

- **SCL** (System Clock) es la línea de los pulsos de reloj que sincronizan el sistema.
- **SDA** (System Data) es la línea por la que se mueven los datos entre los dispositivos.
- **GND** (Masa) es el común de la interconexión entre todos los dispositivos "enganchados" al bus.

Las líneas **SDA** y **SCL** son del tipo 'drenaje abierto', es decir, de un estado similar al de colector abierto, pero asociadas a un transistor de efecto de campo (o FET). Se deben polarizar en estado alto (conectando a la alimentación por medio de resistores "pull-up"), lo que define una estructura de bus que permite conectar en paralelo múltiples entradas y salidas.

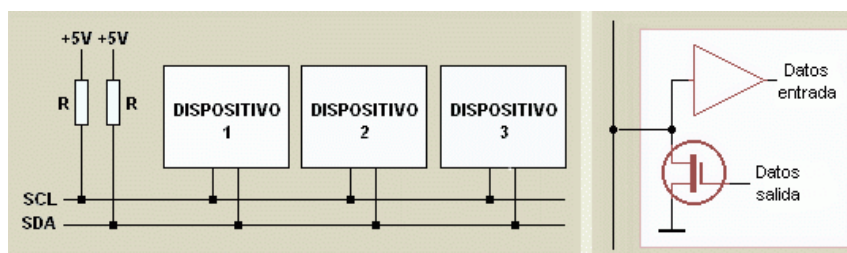


FIGURA 74: ESTRUCTURA DE UN BUS SDA Y SCL

El diagrama es suficientemente autoexplicativo. Las dos líneas del bus están en un nivel lógico alto cuando están inactivas. En principio, el número de dispositivos que se puede conectar al bus no tiene límites, aunque hay que observar que la capacidad máxima sumada de todos los dispositivos no supere los 400 pF. El valor de los resistores de polarización no es muy crítico,

y puede ir desde 1K8 (1.800 ohms) a 47K (47.000 ohms). Un valor menor de resistencia incrementa el consumo de los integrados, pero disminuye la sensibilidad al ruido y mejora el tiempo de los flancos de subida y bajada de las señales. Los valores más comunes en uso están entre 1K8 y 10K.

3.4.3.2. PROTOCOLO I2C

Habiendo varios dispositivos conectados sobre el bus, es lógico que para establecer una comunicación a través de él se deba respetar un protocolo. Digamos, en primer lugar, lo más importante: existen dispositivos **maestros** y dispositivos **esclavos**. Sólo los dispositivos maestros pueden iniciar una comunicación.

La condición inicial, de **bus libre**, es cuando ambas señales están en estado lógico alto. En este estado cualquier dispositivo maestro puede ocuparlo, estableciendo la condición de **inicio** (start). Esta condición se presenta cuando un dispositivo maestro pone en estado bajo la línea de datos (SDA), pero dejando en alto la línea de reloj (SCL).

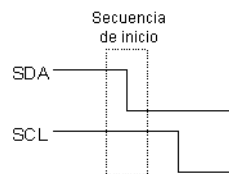


FIGURA 75: SECUENCIA DE INICIO EN I2C

El primer byte que se transmite luego de la condición de inicio contiene siete bits que componen la dirección del dispositivo que se desea seleccionar, y un octavo bit que corresponde a la operación que se quiere realizar con él (lectura o escritura).

Si el dispositivo cuya dirección corresponde a la que se indica en los siete bits (A0-A6) está presente en el bus, éste contesta con un bit en bajo, ubicado inmediatamente luego del octavo bit que ha enviado el dispositivo maestro. Este bit de **reconocimiento** (ACK) en bajo le indica al dispositivo maestro que el esclavo **reconoce** la solicitud y está en condiciones de comunicarse. Aquí la comunicación se establece en firme y comienza el intercambio de información entre los dispositivos.

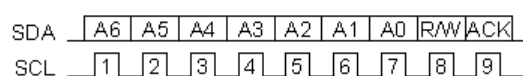


FIGURA 76: INTERCAMBIO DE INFORMACIÓN EN I2C

Si el bit de **lectura/escritura** (R/W) fue puesto en esta comunicación a nivel

lógico bajo (escritura), el dispositivo maestro envía datos al dispositivo esclavo. Esto se mantiene mientras continúe recibiendo señales de reconocimiento, y el contacto concluye cuando se hayan transmitido todos los datos.

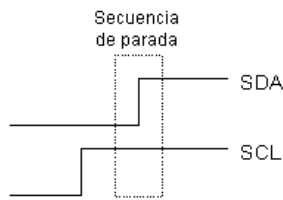


FIGURA 77: : SECUENCIA DE PARADA EN I2C

3.4.3.3. DEFINICIÓN DE TÉRMINOS

- **Maestro (Master):** Dispositivo que determina los tiempos y la dirección del tráfico en el bus. Es el único que aplica los pulsos de reloj en la línea SCL. Cuando se conectan varios dispositivos maestros a un mismo bus la configuración obtenida se denomina "multi-maestro".
- **Esclavo (Slave):** Todo dispositivo conectado al bus que no tiene la capacidad de generar pulsos de reloj. Los dispositivos esclavos reciben señales de comando y de reloj generados desde el maestro.
- **Bus libre (Bus Free):** Estado en el que ambas líneas (SDA y SCL) están inactivas, presentando un estado lógico alto. Es el único momento en que un dispositivo maestro puede comenzar a hacer uso del bus.
- **Comienzo (Start):** Se produce cuando un dispositivo maestro ocupa el bus, generando la condición. La línea de datos (SDA) toma un estado bajo mientras que la línea de reloj (SCL) permanece alta.
- **Parada (Stop):** Un dispositivo maestro puede generar esta condición, dejando libre el bus. La línea de datos y la de reloj toman un estado lógico alto.
- **Dato válido (Valid Data):** Situación presente cuando un dato presente en la línea SDA es estable al tiempo que la línea SCL está a nivel lógico alto.
- **Formato de Datos (Data Format):** La transmisión de un dato a través de este bus consiste de 8 bits de dato (1 byte). A cada byte transmitido al bus le sigue un noveno pulso de reloj durante el cual el dispositivo receptor del byte debe generar un pulso de reconocimiento.
- **Reconocimiento (Acknowledge):** El pulso de reconocimiento, conocido como ACK (del inglés Acknowledge), se logra colocando la línea de datos a un nivel lógico bajo durante el transcurso del noveno pulso de reloj.
- **Dirección (Address):** Todo dispositivo diseñado para funcionar en este

bus posee su propia y única dirección de acceso, preestablecida por el fabricante. Hay dispositivos que permiten definir externamente parte de la dirección de acceso, lo que habilita que se pueda conectar en un mismo bus un conjunto de dispositivos del mismo tipo, sin problemas de identificación. La dirección 00 es la denominada "de acceso general"; a ésta responden todos los dispositivos conectados al bus.

- **Lectura/Escritura** (Bit R/W): Cada dispositivo tiene una dirección de 7 bits. El octavo bit (el menos significativo), que se envía durante la operación de direccionamiento, completando el byte, indica el tipo de operación que hay que realizar. Si este bit es alto el dispositivo maestro lee información proveniente de un dispositivo esclavo. Si este bit es bajo, el dispositivo maestro escribe información en un dispositivo esclavo.¹⁶

3.4.4. COMUNICACIÓN SPI

El **Bus SPI** (*Serial Peripheral Interface*) es un estándar de comunicaciones, usado principalmente para la transferencia de información entre circuitos integrados en equipos electrónicos. El bus de interfaz de periféricos serie o bus SPI es un estándar para controlar casi cualquier dispositivo electrónico digital que acepte un flujo de bits serie regulado por un reloj.

Incluye una línea de reloj, dato entrante, dato saliente y un pin de *chip select*, que conecta o desconecta la operación del dispositivo con el que uno desea comunicarse. De esta forma, este estándar permite multiplexar las líneas de reloj.¹⁷

El SPI es un protocolo síncrono. La sincronización y la transmisión de datos se realizan por medio de 4 señales:

- **SCLK** (*Clock*): Es el pulso que marca la sincronización. Con cada pulso de este reloj, se lee o se envía un bit. También es llamado TAKT (en alemán).
- **MOSI** (*Master Output Slave Input*): Salida de datos del Master y entrada de datos al Slave. También es llamada SIMO.
- **MISO** (*Master Input Slave Output*): Salida de datos del Slave y entrada al Master. También es conocida por SOMI.
- **SS/Select**: Para seleccionar un Slave, o para que el Master le diga al Slave que se active. También es llamada SSTE.

La Cadena de bits es enviada de manera síncrona con los pulsos del reloj, es decir, con cada pulso el Master envía un bit. Para que empiece la transmisión el Master baja la señal SSTE ó SS/Select a cero; con esto el Slave se activa y empieza la transmisión, con un pulso de reloj al mismo tiempo que el primer bit es leído.

¹⁶ Comunicación Bus I2C – Eduardo J.Carletti

¹⁷ Serial Peripheral Interface (SPI) – Universidad Fermín Toro

Nótese que los pulsos de reloj pueden estar programados de manera que la transmisión del bit se realice en 4 modos diferentes; a esto se llama polaridad y fase de la transmisión:

1. Con el flanco de subida sin retraso.
2. Con el flanco de subida con retraso.
3. Con el flanco de bajada sin retraso.
4. Con el flanco de bajada con retraso.

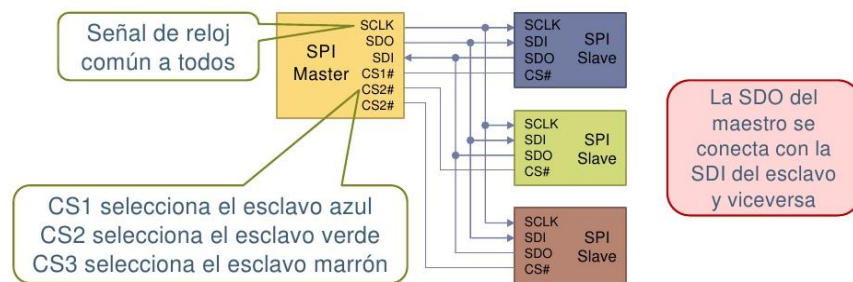


FIGURA 78: COMUNICACIÓN SPI

3.5. OTRAS CONSIDERACIONES

3.5.1. ÁMBITOS DE UTILIZACIÓN

3.5.1.1. INDUSTRIAL

El robot puede ser utilizado en un ámbito industrial, como mapeador de diferentes lugares donde sea necesario tener un control, como por ejemplo:

- Controlando la temperatura de servidores introduciendo un sensor de temperatura.
- Mostrando la cobertura móvil en una nave industrial mediante el acoplamiento un módulo GSM
- Mostrando la cobertura WiFi en una nave industrial acoplando un módulo WiFi
- Mostrando la radioactividad de una sala acoplándole un medidor Geiger.

- Mostrando el nivel de gases, ya sean necesarios o no, de una sala introduciendo un medidor de gases.
- Mostrando el nivel de humedad en una nave industrial acoplando un sensor de humedad.

3.5.1.2. DOMÉSTICO

El robot puede ser utilizado en un ámbito doméstico.

- Acoplándole sistemas de limpieza, puede convertirse en un robot de lavado.
- Acoplándole sistemas de corte, serviría en un entorno exterior como cortador de césped.

3.5.1.3. EDUCATIVO

El robot puede ser utilizado en un ámbito educativo.

- Puede mostrar a estudiantes de programación y electrónica cómo se pueden utilizar y programar los diferentes dispositivos.
- Creando problemas reales, que deben resolver con el robot y utilizando los diferentes sensores e incluso incluyendo más.

3.5.2. CONSEJOS DE UTILIZACIÓN

Para la puesta en marcha del robot es necesario que la tarjeta SD esté en su ranura; si no, se mostrará un mensaje de error. También las baterías del Arduino y de los motores tienen que estar cargadas; sino, se mostrará también un mensaje de error en la LCD. Todos estos mensajes son también grabados en la tarjeta SD.

Después de accionar el interruptor el robot rastreará, y si hay una distancia superior a 40cm de cualquier pared, éste buscará la más cercana y se aproximará a ella, de manera que se situará con los sensores laterales mirando hacia la pared. A partir de este momento el robot empezará a seguir el muro, evitando así los obstáculos que se encuentre.

El robot irá mostrando cada una de las acciones que va realizando mediante mensajes en la pantalla LCD, que además se graban en la tarjeta SD.

Si por cualquier motivo, el robot se encontrara encerrado. Parará sus motores y mostrará un mensaje de error en la LCD; que además grabará en la tarjeta SD.

CAMBIO DE BATERÍAS

Si las baterías se descargan, se pueden extraer fácilmente quitando solamente dos tuercas. De esta manera se pueden cargar sin dificultad y se pueden hacer todos los cambios deseados.

4. CÁLCULOS

Para la realización del proyecto hemos tenido que realizar diferentes cálculos, sea a la hora de realizar la placa de circuito impreso, sea a la hora de la selección de baterías.

4.1. PISTAS

Para realizar el control de motores, necesitamos diseñar un circuito que fuera capaz de controlar el movimiento de las ruedas. Para ello diseñamos un control de motores en puente en H.

Antes de realizar la placa de circuito impreso, realizamos la simulación del circuito en el simulador Isis de Proteus.

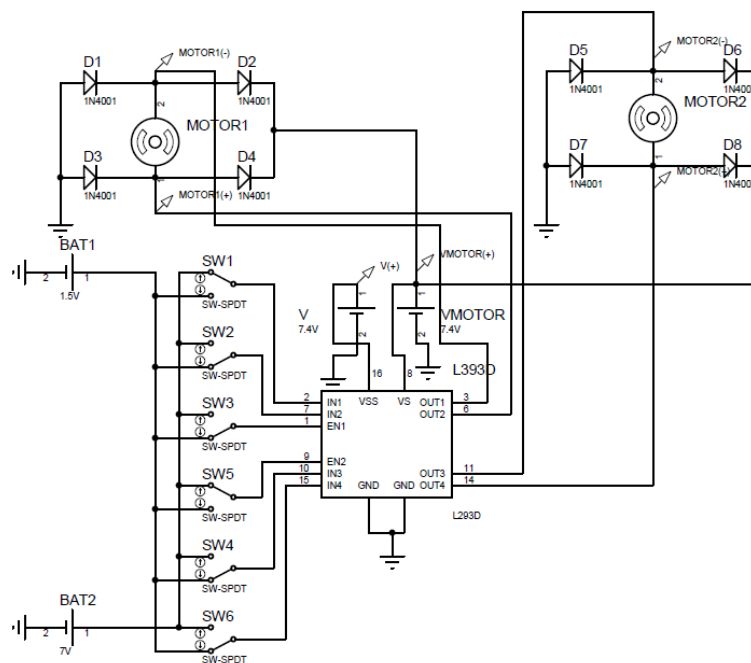


FIGURA 79: SIMULACIÓN DEL CIRCUITO DE MOTORES EN ISIS

Una vez probado y sabiendo que funciona, realizamos el montaje en una Board de pruebas, que conectamos al Arduino y a los motores para realizar las pruebas necesarias.

Una vez conectado, el siguiente paso fue realizar la medida de corriente que pasa por las pistas que van hacia los motores. Esto es importante, pues es donde más problemas nos puede dar la corriente al pasar por una pista, debido a que es donde mayor pico de corriente se produciría de toda la placa, ya que los motores, al iniciar la marcha, requieren un incremento de corriente para aumentar el par y poder vencer el rozamiento de las ruedas contra el suelo.

En primer lugar medimos la corriente media que pasa por esta pista, abriendo el circuito e introduciendo un polímetro en serie. De esta manera obteníamos sólo la corriente media, no la de pico, que quizás fuera la más importante, pero nos podíamos hacer una idea de la cantidad de corriente que circula por esas pistas. Los valores obtenidos fueron los siguientes:

CORRIENTE			
L293D			
	VELOCIDAD MOTOR (PWM)	MAXIMA (mA)	MEDIA (mA)
VELOCIDAD	250	300	200
MAXIMA			
VELOCIDAD	150	200	170
MEDIA			
VELOCIDAD	100	140	140
MINIMA			

TABLA 4: MEDIDA DE CORRIENTE SEGÚN VELOCIDAD DE LOS MOTORES

Tras realizar estas medidas, vimos necesario observar el pico de corriente que se produce en la pista de los motores cuando arranca o cuando lo paramos para que cambie de dirección.

Para realizar esta medida, necesitamos un osciloscopio digital, que es mucho más rápido que uno analógico y con el cual además podríamos congelar la imagen justo cuando se produce el pico de corriente. Además del osciloscopio, como tenemos que medir una caída de tensión, lo realizamos en paralelo con una resistencia muy pequeña o Shunt, que es una carga resistiva a través de la cual se deriva una corriente eléctrica. Al Shunt se le conoce como resistencia de precisión y se utiliza para medir corrientes eléctricas que fluyen a través de la carga, mediante las mediciones de diferentes tensiones, valiéndonos de la ley de Ohm para obtener la intensidad ($I=V/R$).

Hemos utilizado un Shunt de 1Ω en serie con la carga, y mediante el osciloscopio realizando una medida de la caída de tensión en el Shunt, determinamos la corriente.

(Podremos ver unos videos de las medidas con el polímetro en el 9.3.8 Videos en contenido del CD).

Los picos de corrientes obtenidos en el osciloscopio serían los siguientes:



FIGURA 80: PICOS DE CORRIENTE EN LOS MOTORES I

Podemos ver en la gráfica anterior, que el pico de corriente que obtenemos es elevado, caen 4.16V en la resistencia *shunt* de 1Ω , por lo tanto la corriente instantánea que tenemos es de 4.16V, lo que nos preocupa, pues es un pico de corriente elevado a la hora de realizar una pista en una placa de circuito impreso.

Por lo tanto nos decidimos a mirar cuanto tiempo ocurre este pico, a ver si verdaderamente es importante:



FIGURA 81: PICOS DE CORRIENTE EN LOS MOTORES II

Los picos de corriente se producen durante un tiempo de 64nS, por lo que pensamos que no es demasiado tiempo para que la pista se caliente y pueda sufrir un desperfecto. De

todas formas calculamos el grosor de la pista de la siguiente manera:

- Vamos a seguir el estándar general para diseños de circuitos impresos ANSI-IPC 2221, que podréis encontrar en los anexos.
- Para conocer el ancho de la pista en una PCB, necesitamos tres datos de entrada:
 - Corriente máxima que puede circular por la pista (Calculado en unos 4.16A).
 - El incremento máximo de temperatura que puede soportar la pista (depende de las especificaciones de uso que deseemos, 25°C ambientales y 60°C para temperatura de la pista).
 - El grosor de la pista (viene fijada por la PCB que compremos, en nuestro caso el grosor de la pista (alto) es de 35μ).
- Una vez que ya tenemos las especificaciones deseadas, ya podremos aplicar las fórmulas para el cálculo del ancho de pista.
 - Fórmula 1

$$Ancho = \frac{\text{Área}}{L * 1.378}$$

- Donde L representa al grosor de la pista, en nuestro caso 35μ.
- El ancho resultante se expresará en *mils al cuadrado* (Milésima parte de una pulgada).

Para calcular el área utilizaremos la siguiente fórmula:

- Fórmula 2

$$\text{Área} = \left[\frac{I}{k1 * \Delta T^{k2}} \right]^{\frac{1}{k3}}$$

- Donde I representa la corriente máxima.
- K1 es la constante definida por el estándar que estamos aplicando que es 0.0647 para nuestra pista exterior; si fuera interior (más de 2 caras), el coeficiente sería distinto.
- K2 es la constante que vale 0,4281 en nuestra pista exterior, otro coeficiente para caras internas.
- K3 es la última constante, que es de 0.6732 en nuestra PCB y de otro valor con pistas internas.

Si sustituimos la fórmula 2 en la fórmula 1, obtendríamos la siguiente ecuación:

$$Ancho = \frac{\left[\frac{I}{k1 * \Delta T^{k2}} \right]^{\frac{1}{k3}}}{L * 1.378}$$

Una vez que tenemos esta ecuación, ya podemos resolverla y obtener nuestro valor de ancho de pista, para nuestra PCB es de 35μ.

$$Ancho = \frac{\left[\frac{4.62A}{0.0647 * 35^{\circ} C^{0.4281}} \right]^{\frac{1}{0.6732}}}{0.0002mm * 1.378} = 203.66mils$$

Es un ancho demasiado elevado, porque hemos seleccionado los 4.62A que tenemos en los picos, pero no sería tan elevado en promedio; y al tener estos picos tan poco tiempo, decidimos bajar los amperios a una medida media de 1A, aunque nuestro promedio de corriente es de 300mA.

De esta manera el ancho que obtenemos es de:

$$Ancho = 20.970mils$$

Ahora con el ancho de la pista calculado, ya podremos realizar el diseño de la PCB en el programa Ares de Proteus.

Una vez que hemos realizado el esquema del sistema en la herramienta Isis de Proteus, nos disponemos a realizar la PCB en Ares.

Teniendo los componentes seleccionados, automáticamente se seleccionan los encapsulados en Ares, de tal forma que colocándolos en las medidas deseadas de la placa (sería routearlo directamente) el ancho de pista elegido ha sido de 25mils.

Uno de los problemas que hemos tenido a la hora de realizar la PCB en el Ares, es que queríamos realizar un diseño pequeño y funcional, y así poder colocarlo en el robot ocupando el menor espacio posible.

El problema es que de esta manera las pistas se encontrarían muy juntas y no podríamos realizar los taladros para soldar el encapsulado del L293D. Por lo que finalmente, ampliamos la placa para poder mejorar el diseño.

Tras colocar los componentes y routear la placa con las condiciones básicas, de no crear ángulos rectos y dejar una medida mínima tanto entre pistas como entre pista y componente, realizamos un enrejado GND y unos taladros, tanto en las esquinas como en lugares interiores, para poder tener más opciones de acoplamiento con nuestro chasis.

El diseño final es el siguiente:

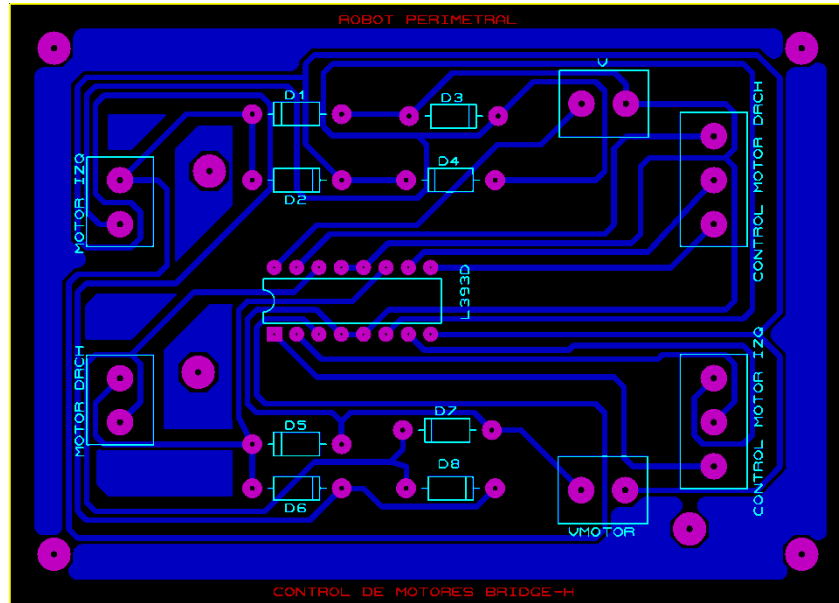


FIGURA 82: DISEÑO FINAL DE LA PCB

Las características finales de la PCB son las siguientes:

Board (Layout) Information

Filename: C:\...\Control de motores\Diseño\ControlmotoresPuenteH.LYT
 Dimensions: 3.76in x 2.68in / 9.5504cm x 6.8072cm

Objects: 33	Breakdown of hole diameters:
Pins: 53	
Vias: 0	
Holes: 53	
Traces: 41	
Traces Length: 63.7in / 161.798cm	
Nets: 15	
Missings: 0	
Missings Length: 0	

Diameter	Number Of Holes
20th	16
40th	16
60th	21

OK

FIGURA 83: CARACTERÍSTICAS FINALES DE LA PCB

En 3D, la PCB quedara de esta manera:

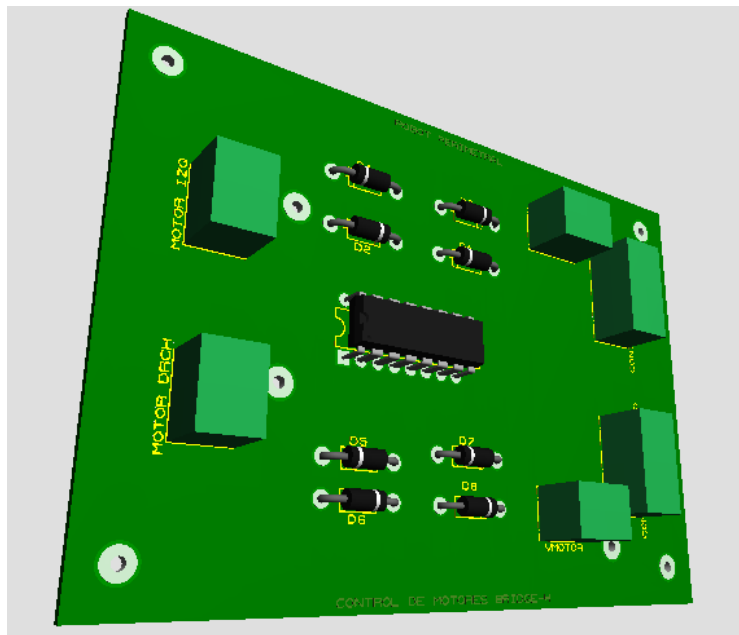


FIGURA 84: VISIÓN DE LA PCB EN 3D

Tras realizar el diseño, falta insolar la PCB, primero tendremos que imprimir en papel cebolla las pistas y los Pads para después poder realizar los taladros.

Tenemos que tener muy en cuenta cómo realizamos la impresión, sabiendo que tenemos que imprimir la cara de las pistas y pads y realizarlo en Mirror (espejo), pues si no a la hora de insolar la placa quedara al revés, y los pads no coincidirán con los pines de los componentes.

Después necesitamos insolar la placa en una insoladora entre 3 y 5 minutos.

Tras insolar la placa, la lavamos bien bajo el agua, para a continuación atacarla con el ácido.

Cuando la atacamos con el ácido, tenemos muchas precauciones, tanto para no atacarla demasiado y que las pistas se queden incompletas, como atacarla poco y que no se quite el cobre sobrante.

Una vez terminado el atacado, la lavamos de nuevo con agua y con alcohol vamos frotando bien toda la superficie para limpiarla de toda impureza que haya podido quedar.

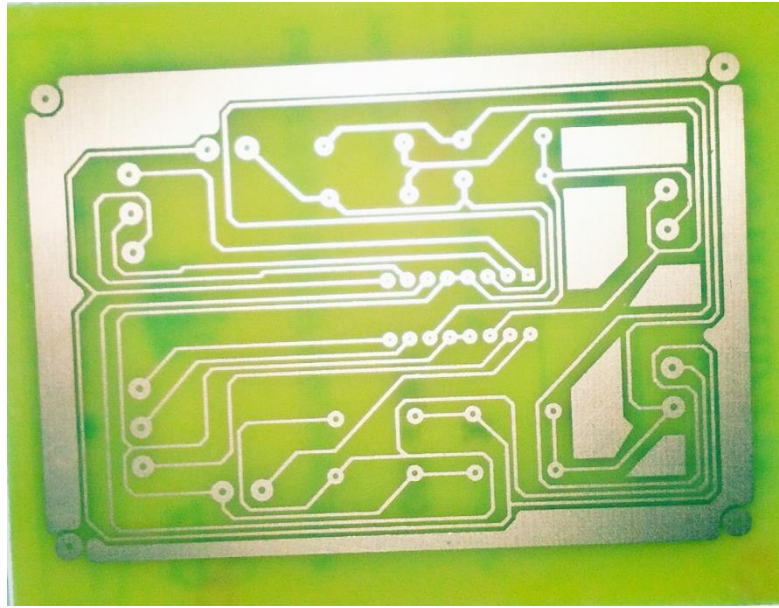


FIGURA 85: PCB INSOLADA

Ya insolada y atacada, nos disponemos a realizar los taladros, utilizando diferentes diámetros de broca según el Pad taladrado.

Sólo nos queda soldar los componentes, teniendo en cuenta la colocación correcta, pues los diodos y el encapsulado tienen su colocación determinada.

La apariencia final de la PCB conectada a los motores es ésta:

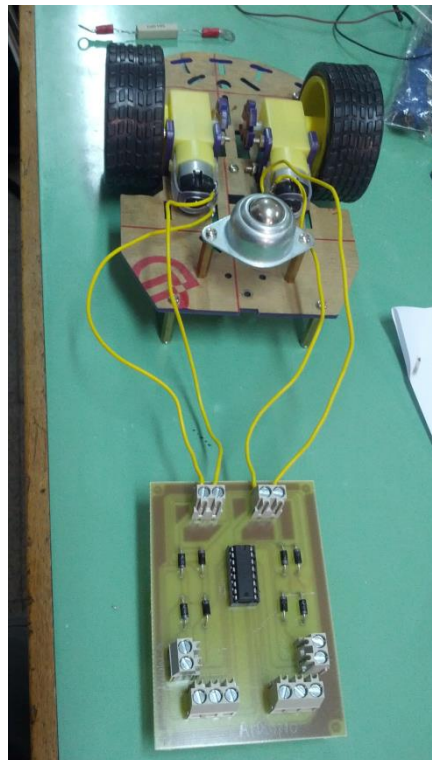


FIGURA 86: PCB CONECTADA A LOS MOTORES PARA PRUEBAS

4.2. CONSUMOS

A la hora de realizar el cálculo de los consumos, primero realizamos la conexión por partes de los elementos para poder saber lo que consumiría cada dispositivo individualmente.

Los consumos los calculamos mediante la medida en serie de la corriente con el dispositivo.

Los consumos son los siguientes:

DISPOSITIVO	CANTIDAD	CONSUMO (mA)		CONSUMO TOTAL (mA)	
LCD	1	36		36	
RTC	1	4		4	
ULTRASONIDOS	3	10.8		32.4	
SERVO MOTOR	1	PICO	30	PICO	30
		PARADO	4	PARADO	4
LECTOR SD	1	PICO	30	PICO	30
		GRABAR	6	GRABAR	6
		STANDBY	4	STANDBY	4
ARDUINO	1	6		6	
CONTROL DE MOTORES	1	160		160	
TOTAL PEOR CASO				298.4	
TOTAL STANDBY				86.4	

TABLA 5: CONSUMOS DE LOS DISPOSITIVOS

El consumo de los motores está anteriormente calculado en el punto 3.4.1. Pistas, donde tenemos una tabla con las corrientes de los motores en cada situación de funcionamiento.

Como vemos en la tabla, en el peor de los casos el consumo es de 298.4 mA. Esto nunca se mantendrá constante, pues no siempre tenemos los motores a la máxima velocidad, ni los dispositivos están en el funcionamiento extremo.

Nuestras baterías son de 1300mAH, por lo que podríamos trabajar a máxima potencia en torno a 4 horas.

4.3. RECTAS BATERÍAS

Los packs de baterías que alimentan los motores y Arduino pueden llegar a 8.2v, puesto que la máxima lectura analógica que puede realizar Arduino es de 5v. Hemos tenido que bajar esta tensión para que Arduino pueda leerla.

De esta manera hemos realizado dos divisores de tensión idénticos para bajar esta tensión y que Arduino pueda leerla. A continuación se muestra un esquema de uno de estos divisores de tensión:

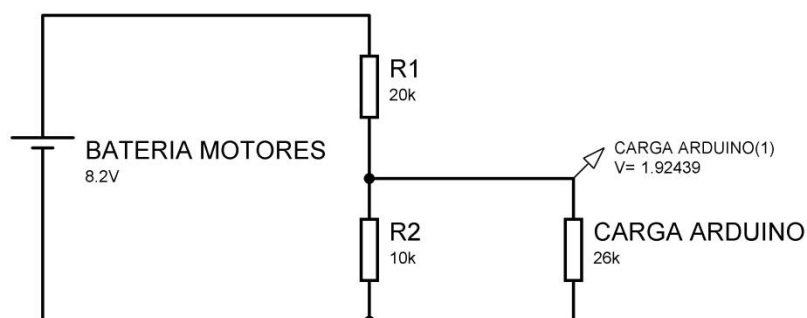


FIGURA 87: CIRCUITO DIVISOR DE TENSIÓN

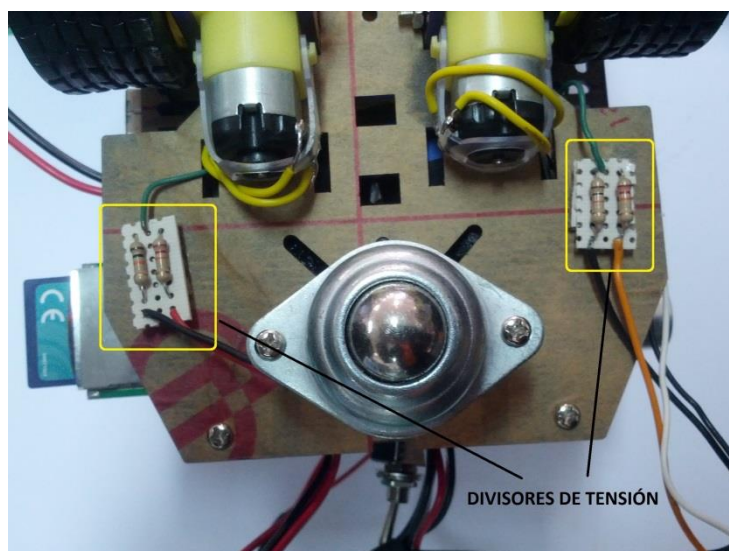


FIGURA 88: CIRCUITOS DIVISORES DE TENSIÓN EN NUESTRO ROBOT

Hemos utilizado un margen de seguridad amplio para no tener problemas con la lectura analógica de Arduino. La máxima tensión que llega a la entrada analógica de Arduino es de 1.92v cuando se le alimenta con 8.2v.

El convertidor que utiliza Arduino es de 1023 bits cuando a éste le llegan 5v. Como nosotros le vamos a meter 1.92v esto nos da un total de 533 bits. Por lo tanto podemos decir que con una tensión de 8.2v nuestro Arduino tendría 533 bits. De esta manera obtenemos también otra segunda medida, donde 7v equivaldrían a 455 bits. A partir de estos dos puntos y aplicando punto pendiente, podemos llegar a la ecuación de la recta:

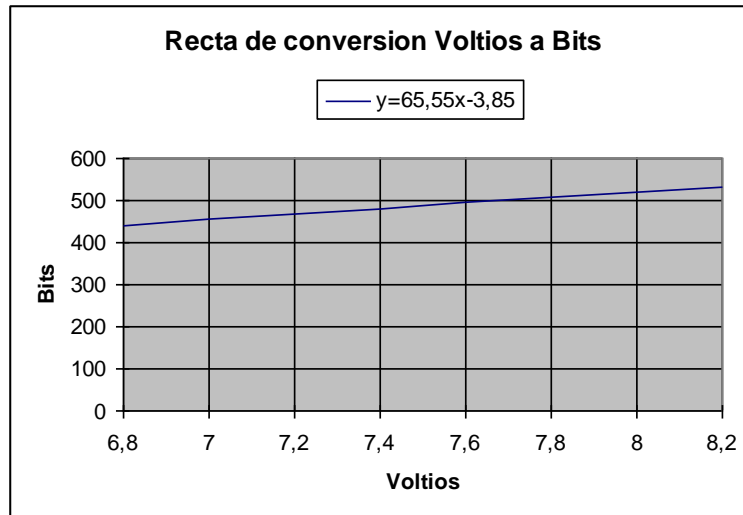


FIGURA 89: RECTA DE CONVERSIÓN DE VOLTIOS A BITS

Mediante esta ecuación controlamos las baterías de ambos dispositivos, y hacemos que no bajen de un nivel óptimo para su funcionamiento. En el caso de los motores, este voltaje límite es 7.2v, y para el caso de Arduino junto con sus componentes es de 6.6v. Si alguno de estos voltajes bajara de su nivel óptimo de funcionamiento, se mostraría un mensaje de error en la pantalla LCD y además se grabaría en la tarjeta SD.

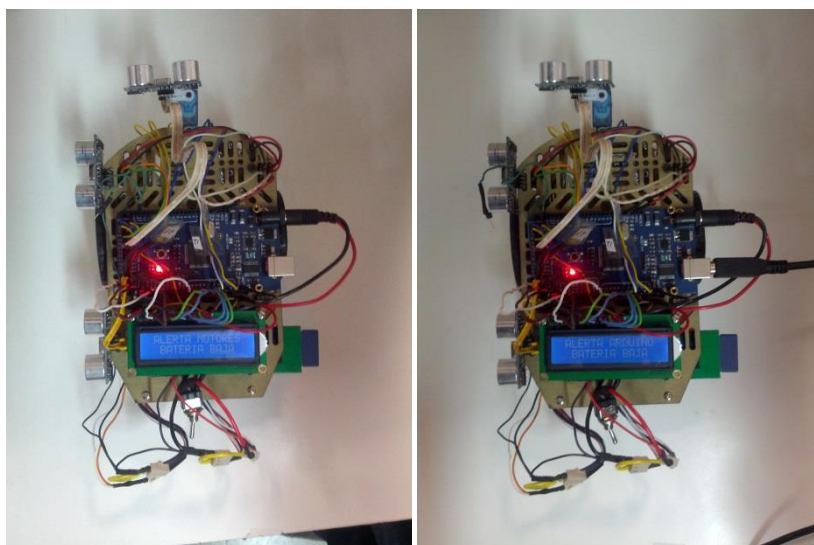


FIGURA 90: MENSAJES DE ALERTA POR LA FALTA DE BATERÍA DE LOS MOTORES Y ARDUINO

4.4. RECTAS PWM

El par de los motores va variando según la carga que tengan las baterías. Entonces, para controlar el PWM hemos realizado dos ecuaciones diferenciadas.

Primeramente hemos visto que la descarga de nuestra batería en el rango óptimo de funcionamiento de motores y Arduino se comporta de manera casi lineal:

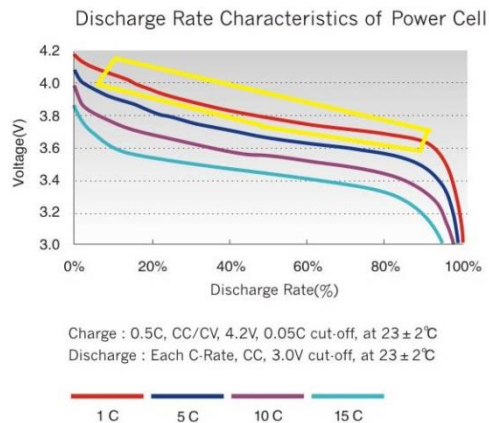


FIGURA 91: GRÁFICA DE DESCARGA DE BATERÍA

Sabiendo que su descarga es lineal y conociendo la carga de las baterías, calculamos el PWM necesario para el giro de cada motor.

Como ya habíamos comentado los motores no tienen el mismo par, y por lo tanto tenemos que crear ecuaciones y gráficas diferentes para cada motor.

Motor derecho:

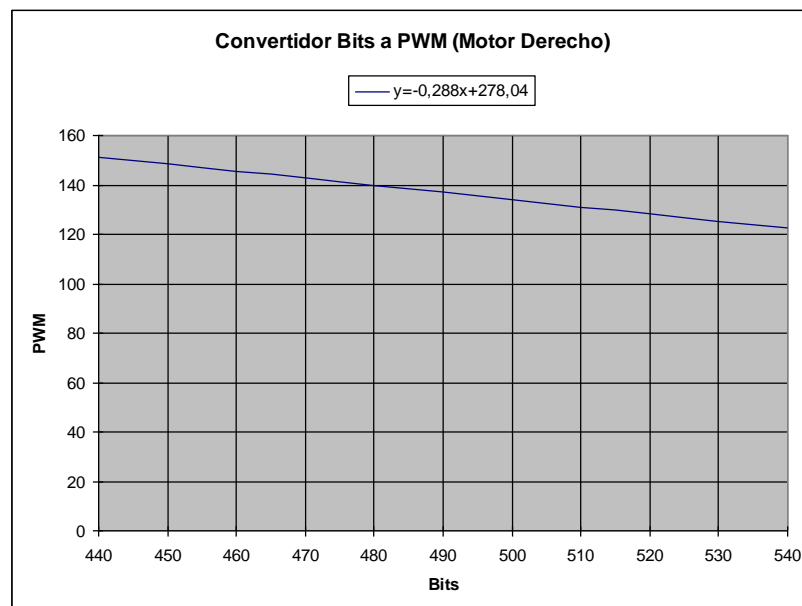


FIGURA 92: GRÁFICA DE CONVERTIDOR DE BITS A PWM (MOTOR DERECHO)

Motor Izquierdo:

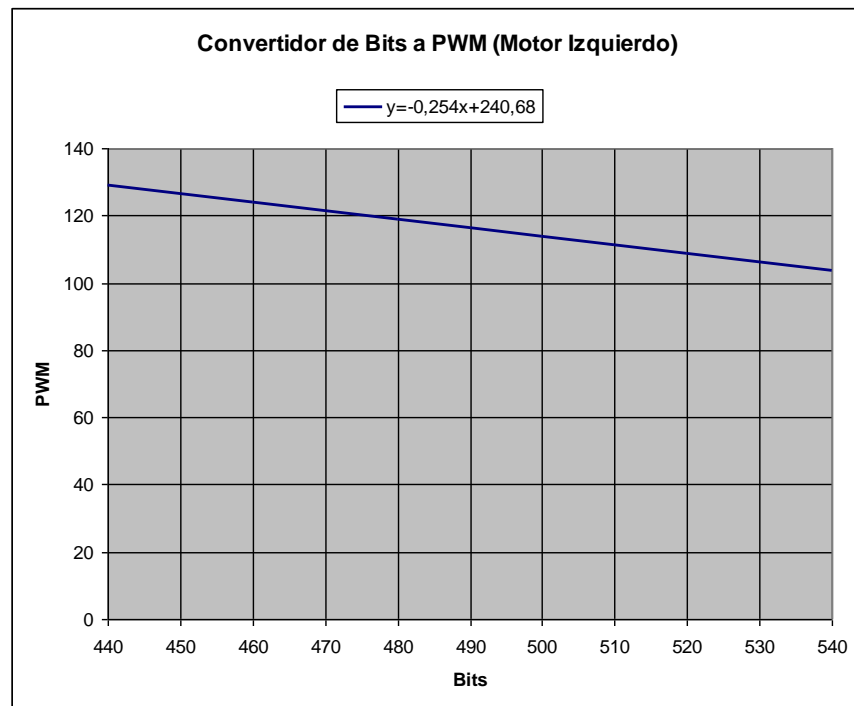
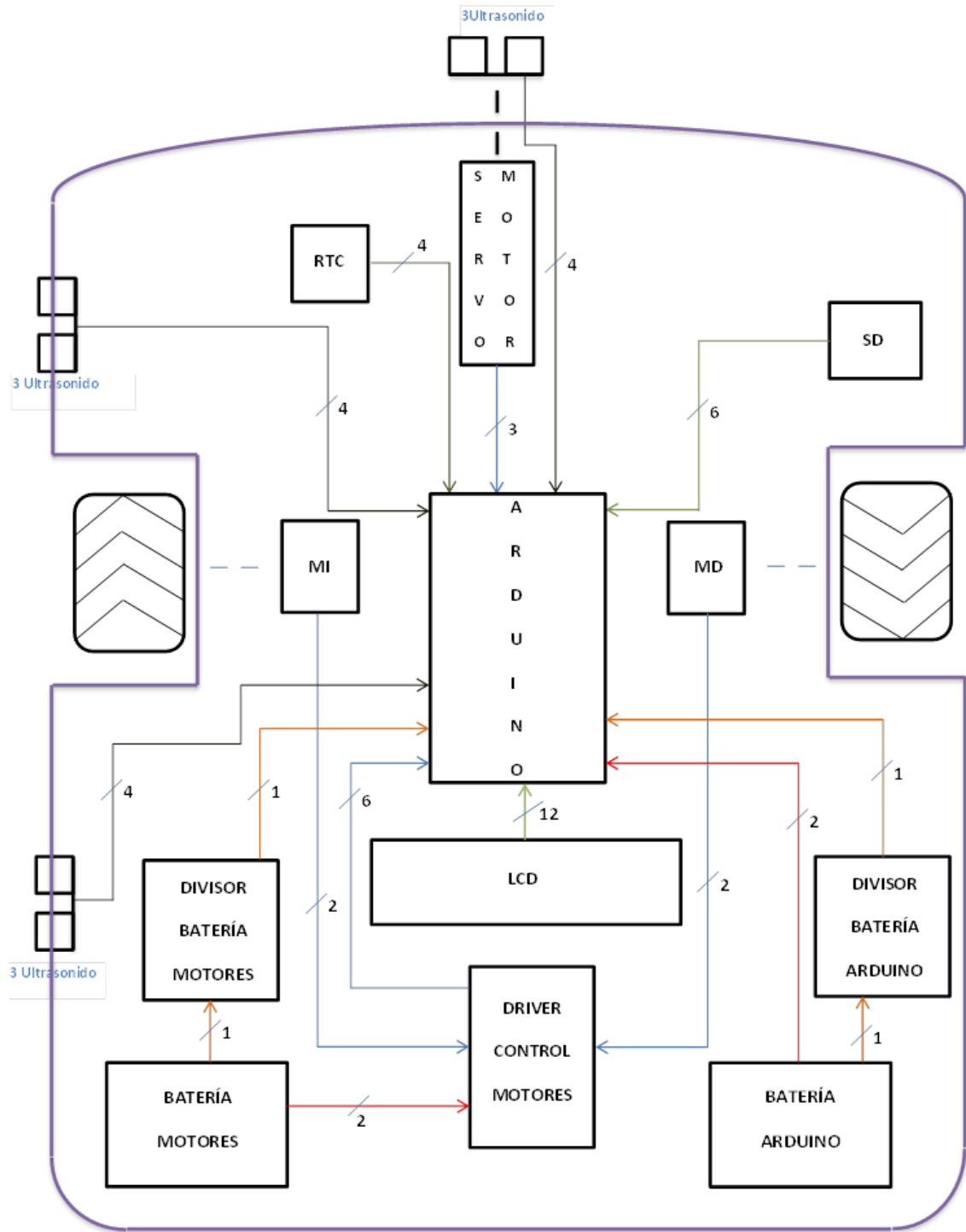


FIGURA 93: GRÁFICA DE CONVERTIDOR DE BITS A PWM (MOTOR IZQUIERDO)

5. PLANOS

Bloque de los sistemas del robot:



Índice de planos del circuito del control de los motores o puente en H:

- Hoja 1: Esquema del control de los motores.

Planos de fabricación:

- Hoja 2: PCB con bordes, pistas y componentes.
- Hoja 3: PCB con bordes y pistas.
- Hoja 4: PCB con serigrafía y bordes.

6. PRESUPUESTO

6.1. LISTA DE MATERIALES

6.1.1. PARTE MECÁNICA

Para la parte mecánica se compró el chasis “Magician Chasis DG007”. Es muy utilizado para hacer prototipos de robot móviles. Éstos son los componentes del pack:

Nombre	Cantidad	Descripción
Chasis superior	1	Parte superior del robot
Chasis inferior	1	Pare inferior del robot
Sujeción de motor	4	Para sujetar los motores al chasis
Rueda omnidireccional	1	Rueda omnidireccional que gira en todas las direcciones
Ruedas 65mm 20mm ancho	2	Ruedas de x diámetro
Espaciador L25	8	Para la sujeción de chasis y dispositivos
Espaciador L10	4	Para la sujeción de chasis y dispositivos
Tornillo M3*30	4	Para la sujeción de chasis y dispositivos
Tornillo M3*10	2	Para la sujeción de chasis y dispositivos
Tornillo M3*6	22	Para la sujeción de chasis y dispositivos
Tuerca M3	2	Para enroscar los tornillos

TABLA 6: PARTE MECÁNICA DEL ROBOT

6.1.2. PARTE HARDWARE

Todos los componentes se han comprado por Internet. En el precio están incluidos los portes y el IVA.

Nombre	Cantidad	Descripción
Arduino Mega + cable USB	1	Arduino Mega + cable USB
Motor CC - DG01D	2	Motor
Batería LIP603562-1C	6	Batería Li-Po 3.5v 1300mAh
Sensor de ultrasonidos HC-SR04	3	Sensor de ultrasonidos
Servomotor 9g HS-55 TREX 450	1	Servomotor
Lector Tarjetas SD Module R/W	1	Lector tarjetas SD
Reloj Real Tiny RTC + CR2032	1	Reloj Real + Pila CR2032 3V
Pantalla LCD 16x2 HD44780	1	Pantalla LCD
Tarjeta SD 512MB	1	Tarjeta SD 512MB
Tira tubo termorretáctil B11-2.4 1.22m	1	
Tira tubo termorretáctil B11-4.5 1.22m	1	
Pines macho 2.54mm tira 20	3	
Pines hembra 2.54mm tira 20	2	
Módulo Board	2	Modulo para conexionado
Jack macho	1	
Interruptor	1	
Cable 1mm tira diferentes colores 1.00m	3	
Cable 2mm tira diferentes colores 1.00m	1	
PCB 150mmx100mm	1	Para insolar el circuito controlador de los motores

TABLA 7: PARTE HARDWARE DEL ROBOT

6.1.3. PARTE SOFTWARE

SOFTWARE ARDUINO

Para la programación de los diferentes dispositivos

SOFTWARE PROTEUS

Para el diseño de los circuitos

SOFTWARE FRITZING

Para el diseño de la protoboard del circuito de motores (véase 9.3.9 Contenido del CD)

MICROSOFT WORD

Para la creación de la memoria

MICROSOFT EXCEL

Para la creación de las gráficas

MICROSOFT VISIO

Para la creación de los diagramas de flujo

WINDOWS MOVIE MAKER

Para la edición de los videos

6.2. PRESUPUESTO ECONÓMICO

6.2.1. PARTE MECÁNICA

Para la parte mecánica se compró el chasis “Magician Chasis DG007”. Es muy utilizado para hacer prototipos de robot móviles. El precio de todos sus componentes está en conjunto. En el precio están incluidos los portes y el IVA.

Nombre	Cantidad	Precio(€)
Chasis superior	1	
Chasis inferior	1	
Sujeción de motor	4	
Rueda omnidireccional	1	
Ruedas 65mm 20mm ancho	2	
Espaciador L25	8	
Espaciador L10	4	
Tornillo M3*30	4	
Tornillo M3*10	2	
Tornillo M3*6	22	
Tuerca M3	2	
Total		28.70

TABLA 8: PRESUPUESTO PARTE MECÁNICA

6.2.2. PARTE HARDWARE

Todos los componentes se han comprado por Internet. En el precio está incluidos los portes y el IVA.

Nombre	Precio Unitario (€)	Cantidad	Precio Total (€)
Arduino Mega+cable USB	44.95	1	44.95
Motor CC – DG01D	2.47	2	4.94
Batería LIP603562-1C	2.29	6	13.74
Sensor de ultrasonidos HC-SR04	3.54	3	10.62
Servomotor 9g HS-55 TREX 450	2.00	1	2.00
Lector Tarjetas SD Module R/W	5.72	1	5.72
Reloj Real Tiny RTC + CR2032	3.80	1	3.80
Pantalla LCD 16x2 HD44780	2.40	1	2.40
Tarjeta SD 512Mb	2.93	1	2.93
Tira tubo termorretáctil B11-2.4 1.22m	0.50	1	0.50
Tira tubo termorretáctil B11-4.5 1.22m	0.60	1	0.60
Pines macho 2.54mm tira 40	0.40	2	0.80
Pines hembra 2.54mm tira 40	1.50	1	1.50
Módulo Board	8.75	2	17.50
Jack macho	0.27	1	0.27
Interruptor	0.59	1	0.59
Cable 1mm tira diferentes colores 1.00m	0.60	6	3.60
Cable 2mm tira diferentes colores 1.00m	0.70	2	1.40
PCB 150mmx100mm	2.00	1	2.00
Total			119.86

TABLA 9: PRESUPUESTO PARTE HARDWARE

Ahora presentamos el presupuesto para la PCB del control de motores y los divisores de tensión necesarios para el control de las baterías:

Nombre	Precio Unitario (€)	Cantidad	Precio Total (€)
PCB una cara	1.90	1	1.90
Integrado L293D	4.00	1	4.00
Diodo 1N4001	0.10	8	0.80
Resistencia 2.2K	0.07	1	0.07
Resistencia 10K	0.07	3	0.21
Resistencia 20K	0.07	2	0.14
Total			7.12

TABLA 10: PRESUPUESTO PCB Y DIVISORES DE TENSIÓN

7. PLIEGO DE CONDICIONES

7.1. OBJETIVO DEL PLIEGO

El objetivo de este proyecto es el diseño de un robot móvil perimetral. Este proyecto está orientado a la posible industrialización del prototipo fabricado. Esto implica que el diseño haya tenido en cuenta la accesibilidad y la fiabilidad sin omitir su desarrollo industrial. En caso de una futura aplicación industrial se debería tener presente el pliego de condiciones, que tiene como principal función regular las condiciones entre las partes contratantes considerando los aspectos técnicos, facultativos, económicos y legales.

El pliego de condiciones define, entre otros, los siguientes aspectos:

- Obras que componen el proyecto.
- Características exigibles a los materiales y componentes.
- Detalles de la ejecución.
- Programa de obras.

Dado el gran número de detalles tratados si se presentan dudas a la hora de iniciar el proyecto lo más recomendable es ponerse en contacto con el proyectista.

7.1.1. DESCRIPCIÓN GENERAL DEL MONTAJE

Las diferentes partes que componen la obra a realizar por parte del instalador, poniendo especial énfasis en el orden establecido, no efectuando una actividad concreta sin haber realizado previamente la anterior:

- Encargo y compra de los componentes necesarios.
- Fabricación de la placa de circuito impreso.
- Montaje de los componentes en la placa.
- Conexión de la placa y los diferentes dispositivos.
- Montaje de la placa en el chasis.
- Puesta en marcha del prototipo.
- Controles de calidad y fiabilidad.
- Mantenimiento para el correcto funcionamiento del sistema.

Todas las partes que en conjunto forman la obra de este proyecto, tendrán que ser ejecutadas por montadores cualificados, sometiéndose a las normas de la Comunidad Europea, países o incluso comunidades internacionales que se tengan previstas para este tipo de montajes no haciéndose responsable el proyectista de los desperfectos ocasionados por su incumplimiento.

7.2. NORMATIVA DE OBLIGADO CUMPLIMIENTO

En la lista que veremos a continuación se enumeran las normativas más relevantes que regulan el diseño, montaje y fabricación del calibrador de cronómetros. No son las únicas y, en todo caso, siempre se seguirán las instrucciones expuestas en el Reglamento Electrotécnico de Baja Tensión de 2002:

- Directiva 1999/5/CE: Equipos radioeléctricos y equipos terminales de telecomunicación.
- Directiva 2001/95/CE: Seguridad General de Productos.
- Directiva 2002/95/CE: Restricción de sustancias peligrosas en aparatos eléctricos y electrónicos.
- Directiva 2002/96/CE: Residuos de aparatos eléctricos y electrónicos.
- Directiva 2003/108/CE: Modifica la Directiva 2002/96/CE sobre residuos de aparatos eléctricos y electrónicos.
- Directiva 2004/108/CE: Legislación común en los estados miembros sobre la compatibilidad electromagnética.
- Directiva 2006/95/CE: Material de Baja Tensión.
- R. D. 1580/2006, por el que se regula la compatibilidad electromagnética en aparatos eléctricos y electrónicos.

Respecto al desarrollo de productos electrónicos, se pueden encontrar en AENOR (Asociación Española de Normalización y Certificación) las siguientes normativas:

- EN 61010.1:2001: Requisitos de seguridad de equipos eléctricos de medida, control y uso en laboratorio. Parte 1: Requisitos generales.
- EN 61326-2-5:2013: Equipos eléctricos para medida, control y uso en laboratorio. Requisitos de compatibilidad electromagnética (CEM).
- UNE 20-050-74 (I). Código para las marcas de resistencias y condensadores. Valores y tolerancias.

- UNE 20-524-75 (I). Técnica circuitos impresos. Parámetros fundamentales. Sistemas de cuadrícula.
- UNE 20-524. Equipos electrónicos y sus componentes. Soldabilidad de circuitos impresos.
- UNE 20-524-77 (II). Técnica de circuitos impresos. Terminología
- UNE 20-531-73. Series de valores nominales para resistencias y condensadores.
- UNE 20-543-85 (I). Condensadores fijos en equipos electrónicos.
- UNE 20-545-89. Resistencias fijas para equipos electrónicos.
- UNE 20916: 1995: Estructuras mecánicas para equipos electrónicos. Terminología.
- UNE 21352: 1976: explicación de las cualidades y funcionamiento de equipos de media electrónicos.
- UNE-EN61000-4-3-1998: Compatibilidad electromagnética.
- EN61021-1: 1997: Núcleos de chapas laminadas para transformadores e inductancias destinadas a ser utilizadas en equipos electrónicos y de telecomunicaciones.
- EN123500: 1992: Especificación intermedia: placas de circuitos impresos flexibles con taladros para la inserción de componentes.

Normas DIN:

- DIN 40801. Parte 1. Circuitos impresos, fundamentos, retículos.
- DIN 40801. Parte 2. Circuitos impresos, fundamentos, orificios y espesores nominales.
- DIN 40803. Parte 1. Circuitos impresos, placas de circuito impreso, requisitos generales y comprobaciones, tablas de tolerancias.
- DIN 40803. Parte 2. Circuitos impresos, placas de circuito impreso, documentación.
- DIN 40804. Circuitos impresos, conceptos.
- DIN 41494. Formas de construcción para dispositivos electrónicos, placas de circuito impreso, medidas.

7.3. CONDICIONES GENERALES DE LOS MATERIALES

Todos los componentes utilizados en el proyecto cumplen las especificaciones técnicas que aparecen descritas tanto en la memoria, como en los planos, estando presente en estos últimos las particularidades técnicas referentes a valores, referencias y demás especificaciones relevantes utilizadas en resistencias, circuitos integrados, etc.

7.3.1. ESPECIFICACIONES ELÉCTRICAS

REGLAMENTO ELECTROTÉCNICO DE BAJA TENSIÓN

Todos los aspectos técnicos de la instalación que, directa o indirectamente, estén incluidos en el Reglamento Electrotécnico de Baja Tensión, tendrán que cumplir lo que se disponga en las respectivas normas.

Las instrucciones más importantes relacionadas con la realización del proyecto son las siguientes:

- I.T.C.B.T.002: Normas de referencia en el RBT.
- I.T.C.B.T.019 a la B.T.024: Instalaciones interiores o receptoras.
- I.T.C.B.T.036: Instalaciones a muy baja tensiones.
- I.T.C.B.T.037: Instalaciones a tensiones especiales.
- I.T.C.B.T.043: Instalaciones de receptores. Prescripciones generales.
- I.T.C.B.T.048: Instalaciones de Receptores. Transformadores y autotransformadores. Reactancias y rectificadores. Condensadores.

PLACAS DE CIRCUITO IMPRESO

El circuito diseñado se realizara sobre placa de fibra de vidrio, utilizándose de simple cara según el diseño, con un espesor mínimo de 1,7mm.

CONDUCTORES ELÉCTRICOS

Los conductores utilizados serán externos con cableados de 1mm y 2mm. Éstos conectarán la placa de circuito impreso con el resto de componentes.

COMPONENTES PASIVOS

Los componentes pasivos utilizados en el proyecto son los disponibles tecnológicamente en el momento de la realización del proyecto.

COMPONENTES ACTIVOS

Los componentes activos utilizados en el proyecto son los disponibles tecnológicamente en el momento de la realización del proyecto.

RESISTENCIAS

Existen resistencias con una gran precisión en el valor, lo que implica fijar tolerancias muy bajas, pero se tendrá en cuenta que su precio aumenta considerablemente y serán necesarias en aplicaciones muy específicas como en nuestro caso. Estando normalmente destinadas a usos generales las tolerancias estandarizadas de 5%, 10% y 20%.

CIRCUITOS INTEGRADOS Y SEMICONDUCTORES

Para el control de los motores utilizamos como circuito integrado el L293D.

Se tendrá que alimentar a una tensión adecuada, las características de tensión y corriente de entrada-salida, se encuentran en las hojas del fabricante del [Anexo](#).

7.3.2. ESPECIFICACIONES MECÁNICAS

Todos los materiales escogidos son de una calidad que se adapta al objetivo del proyecto, no obstante si no se pudiera encontrar en el mercado algún producto por estar agotado, el instalador encargado del montaje tendrá que estar capacitado para su sustitución por otro similar o equivalente.

La placa de circuito impreso se realizará en fibra de vidrio. Se recomienda el uso de zócalos torneados, para la inserción de componentes de agujero pasante. De esta forma se reduce el tiempo de reparación y además se disminuye el calentamiento de los pines de los componentes electrónicos en el proceso de soldadura que podría producir su deterioro.

7.3.3. COMPONENTES ELECTRÓNICOS

Todos los componentes electrónicos empleados en la elaboración del control de motores deben atender a los requerimientos de potencia, tensión y corriente demandados por el sistema. Todos los elementos deben cumplir al menos con las especificaciones del sistema, incluso podrán mejorar si eso no afecta al aumento del coste final del proyecto.

Vendrá convenientemente especificado en la Lista de Materiales el valor de los componentes, así como su tipo de encapsulado utilizado para la placa de circuito impreso.

7.4. CONDICIONES DEL PROCESO DE FABRICACIÓN

7.4.1. PREPARACIÓN DE LOS COMPONENTES

La adquisición de los componentes para la fabricación de este prototipo debe realizarse teniendo en cuenta sus especificaciones técnicas, además de cumplir con las pautas exigidas que hemos visto anteriormente.

Todos los materiales y elementos necesarios para la fabricación del prototipo deben estar comprados antes de iniciar el proceso de fabricación. Se debe tener todo preparado para poder comenzar sin problemas ni retrasos.

7.4.2. MATERIAL DEL CIRCUITO IMPRESO

La placa elegida para la realización del control de los motores es una placa de fibra de vidrio de simple cara, de un espesor mínimo de 1.7mm, siendo así más resistente a los cambios climáticos y mecánicos.

Esta placa se ha realizado a través del método de insolación, para lo cual se necesitaría una insoladora (ó lámpara de luz actínica), revelador y un atacador rápido. Para este último proceso necesitaremos que nuestra placa de circuito impreso sea de material fotosensible positivo de una cara y fibra de vidrio.

7.4.3. SOLDADURA Y MONTAJE DE LOS COMPONENTES

El montaje de todos los componentes eléctricos seguirán las pautas marcadas en el diseño del controlador de motores.

El circuito integrado de nuestro proyecto debe ser instalado en zócalos, para su instalación, también debemos prever el lugar y la indumentaria del personal de montaje, ya que estos pueden acumular cargas electrostáticas.

Se debe tener muy en cuenta la manipulación de los componentes, ya que este material es susceptible a la hora de su transporte e instalación en circuito impreso.

7.4.4. CONDICIONES DEL PROCESO DE PRUEBA

Una vez terminada la fase de montaje del dispositivo se pasará a realizar al 100% de los dispositivos un test del correcto funcionamiento del prototipo según las

especificaciones.

Se someterá al equipo a ensayos de compatibilidad electromagnética tanto radiados como inducidos para comprobar que el dispositivo es inmune a las radiaciones procedentes de elementos ajenos, comprobando que no se produce variación alguna con respecto a su modo de funcionamiento normal.

En todo caso se seguirá la normativa vigente sobre compatibilidad electromagnética (Directiva 2004/108/CE y su transposición R.D.1580/2006 por el que se regula la compatibilidad electromagnética en aparatos eléctricos y electrónicos).

Antes de la comercialización del producto, y por tanto del certificado CE, se deberá realizar un proceso de evaluación de conformidad con las distintas normativas que son de aplicación al regulador.

7.5. CONDICIONES FACULTATIVAS

Los permisos de carácter obligatorio necesarios para realizar el proyecto o la utilización de la misma tendrán que obtenerse por parte de la empresa contratante, quedando la empresa contratista al margen de todas las consecuencias derivadas de la misma.

Cualquier retardo producido en el proceso de fabricación por causas debidamente justificadas, siendo estas ajenas a la empresa contratista, será aceptada por el contratante, no teniendo este último derecho a reclamación por daños o perjuicios.

Cualquier demora no justificada supondrá el pago de una multa por valor del 6% del importe total de fabricación, para cada fracción del retardo temporal (acordado en el contrato).

La empresa contratista se compromete a proporcionar las mayores facilidades al contratista para que la obra se realice de una forma rápida y adecuada.

El aparato cumplirá los requisitos mínimos respecto al proyecto encargado, cualquier variación o mejora sustancial en el contenido del mismo tendrá que ser consultada con el técnico diseñador (proyectista).

Durante el tiempo que se haya estimado la instalación, el técnico proyectistas podrá anunciar la suspensión momentánea si así lo estimase oportuno.

Las características de los elementos y componentes serán los especificados en la memoria y el pliego de condiciones, teniendo en cuenta su perfecta colocación y posterior uso.

La contratación de este proyecto se considerará válida una vez que las dos partes implicadas, propiedad y contratista, se comprometan a concluir las cláusulas del contrato, por el cual tendrán que ser firmados los documentos adecuados en una reunión conjunta en haber llegado a un acuerdo.

Los servicios de la empresa contratista se consideran finalizados desde el mismo momento en que el aparato se ponga en funcionamiento, después de la previa comprobación de su correcto funcionamiento.

El presupuesto no incluye los gastos de tipo energético ocasionados por el proceso de instalación, ni las obras que fuesen necesarias, que irán a cargo de la empresa contratante.

El cumplimiento de las elementales comprobaciones por parte de la empresa instaladora, no será competencia del proyectista, el cual queda fuera de toda responsabilidad derivada del incorrecto funcionamiento del equipo como consecuencia de esta omisión.

7.6. SOLICITUD DE HOMOLOGACIÓN DE TIPO CE

El marcado CE indica que un producto es presuntamente conforme con todas las disposiciones de las directivas que son de aplicación al equipo en cuestión. Igualmente, garantiza que el fabricante ha tomado todas las medidas oportunas para garantizar el cumplimiento de las mismas en cada uno de los productos comercializados. Por lo tanto, tanto el fabricante como el producto cumplen con los requisitos esenciales de las directivas de aplicación.

Es totalmente indispensable que todo producto comercializado o puesto en servicio posea el correspondiente marcado CE. Esto no implica que todo producto deba llevar el marcado CE, ya que sólo es obligatorio que lo posean únicamente aquellos productos que estén regulados por directivas comunitarias de marcado CE.

El caso que nos ocupa se rige principalmente por la Directiva 2006/95/CE sobre material de baja tensión, la Directiva 2004/108/CE sobre la compatibilidad electromagnética y la Directiva 2002/96/CE sobre residuos de aparatos eléctricos y electrónicos, pero siempre en línea con todas las normativas que le son de aplicación. No es aceptable la conformidad parcial, es decir, la conformidad con sólo algunas de las directivas aplicables. Cumpliendo los requisitos esenciales de estas normativas.

El fabricante es el responsable de los procedimientos de certificación y, en su caso, certificación de la conformidad de un producto. Básicamente tiene que:

- Garantizar el cumplimiento del producto con los requisitos esenciales de las Directivas de aplicación.
- Firmar la Declaración "CE" de conformidad.
- Elaborar la documentación o expediente técnico.
- Fijar el marcado "CE".

7.6.1. EXPEDIENTE TÉCNICO DE CONSTRUCCIÓN

El marcado CE lo debe poner siempre el fabricante o su representante legal autorizado, ya que éste es principal responsable de la comercialización o puesta en servicio del producto y de la garantía de su seguridad. Para ello debe realizar un ETC (Expediente Técnico de Construcción) que contará con la siguiente relación:

- Descripción general del producto.

- Análisis de los requisitos esenciales de la/s directivas aplicables.
- Análisis de riesgos. Descripción de las soluciones adoptadas para prevenir los riesgos presentados por el producto.
- Lista de las normas aplicadas total o parcialmente, y la descripción de las soluciones adoptadas para cumplir los aspectos de seguridad de la Directiva en cuestión, en los casos en que no hayan sido aplicadas las normas.
- Informes técnicos con los resultados de los ensayos efectuados o certificados obtenidos de un organismo o laboratorio competente. Tales informes de ensayo serán necesarios si el fabricante declara conformidad con una norma armonizada y podrán ser efectuados por él mismo o bien por un organismo o laboratorio competente. Resultados de los cálculos efectuados en el diseño, de los controles realizados, etc.
- Planos de diseño y de fabricación, y esquemas de los componentes, subconjuntos, circuitos, etc. Explicaciones y descripciones necesarias para la comprensión de los mencionados planos y esquemas, y del funcionamiento del producto.
- Homogeneidad de la producción. Todas las medidas necesarias adoptadas por el fabricante para que el proceso de fabricación garantice la conformidad de los productos manufacturados.

7.6.2. DECLARACIÓN DE CONFORMIDAD DEL PRODUCTO

Para certificar la conformidad del producto, el fabricante o su representante establecido en la Comunidad, deberá elaborar una Declaración de Conformidad.

- Nombre y dirección del fabricante o de su representante establecido en la Comunidad. En caso de productos fabricados fuera de la Comunidad, se deberá indicar tanto el nombre del fabricante como el nombre del representante legal. Se debe hacer constar la dirección completa de la sede o de una de las fábricas o la de uno de los establecimientos del país destino.
- Descripción del producto.
- Todas las disposiciones pertinentes a las que se ajuste el producto.
- Referencia a las Directivas de aplicación. Aunque no es obligatorio, también se puede incluir las referencias a las transposiciones nacionales, es decir, referencia a los Reales Decretos que transponen las Directivas de aplicación.
- Referencia a las normas armonizadas. Aunque las normas armonizadas no son de obligatorio cumplimiento, al fabricante le interesa que se indiquen

dichas normas, ya que dichas normas proporcionan al producto una presunción de conformidad con los requisitos esenciales de la Directiva. Se podrá hacer referencia a la norma europea o directamente a la norma nacional. Debido a que el estado normativo avanza continuamente, se debería indicar la edición y fecha de publicación de la norma en cuestión. Por otro lado, en caso de no utilizar dichas normas armonizadas, se deberá especificar el procedimiento alternativo empleado para satisfacer los requisitos esenciales.

- Identificación del signatario apoderado para vincular al fabricante o a su representante. Es necesario mencionar el nombre del signatario, ya que es una señal de autenticidad. Aunque no es obligatorio, también se suele incluir el lugar y fecha de la firma.
- Nombre y dirección del organismo notificado y número de certificación CE de tipo, si procede. Necesario para aquellas máquinas que hayan obtenido un examen CE de tipo de un organismo notificado.
- Nombre y dirección del organismo notificado al que se haya comunicado o que haya efectuado la comprobación del ETC, si procede.

7.6.3. MARCADO CE SOBRE EL PRODUCTO

Una vez finalizado el proceso completo, el fabricante o representante legal puede proceder a poner el marcado CE sobre el producto. El marcado CE debe colocarse de modo visible, legible e indeleble sobre el equipo o su placa de características. En determinados casos, es aceptable que el marcado CE se ponga sobre el embalaje del producto.

Los diferentes elementos del marcado CE deberán tener una dimensión vertical apreciablemente igual, que no será inferior a 5mm. En caso de reducirse o aumentarse el tamaño del marcado CE, siempre deberán conservarse las proporciones del logotipo de la figura 87.

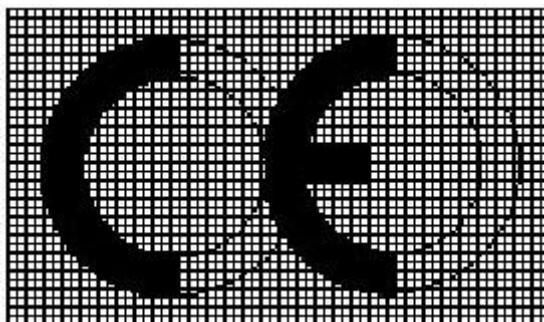


FIGURA 94: LOGOTIPO CE

7.7. MARCA DE RECICLADO DE APARATOS ELÉCTRICOS Y ELECTRÓNICOS

A partir de la Directiva 75/442 se regula a nivel europeo la recogida de residuos. Por lo que se refiere al proyecto que nos atañe, la regulación la marca la Directiva 2002/96 sobre reciclado de aparatos eléctricos y electrónicos.

Este proyecto cumple la Directiva presente en lo que se refiere al diseño como se explica en el artículo 4 de la misma y se ha tenido en cuenta en la realización del diseño.

Por lo que se refiere al proceso de producción, se exige que los aparatos eléctricos y electrónicos se marquen con el símbolo específico.

También se estipula que sean los productores o distribuidores los encargados de la recuperación de estos aparatos para valorarlos. Este punto puede ser de interés a la hora de reducir gastos, ya que la obligatoriedad existe y se debería estudiar un posible beneficio.

A partir de este proyecto será el fabricante el encargado de seguir debidamente el cumplimiento de la normativa.

7.7.1. MARCA DE APARATOS ELÉCTRICOS Y ELECTRÓNICOS

El símbolo indica la recogida selectiva de aparatos eléctricos y electrónicos es el contenedor de basura tachado, tal como aparece representado a continuación este símbolo se estampará de manera visible, legible e indeleble.

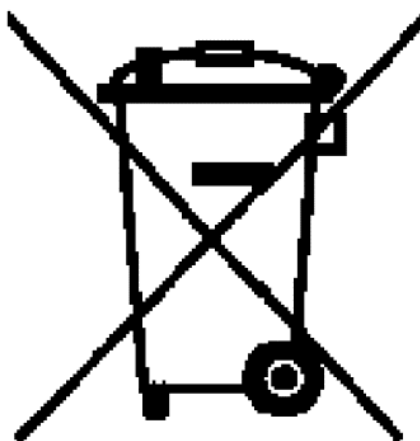


FIGURA 95- LOGOTIPO DE MARCADO DE APARATOS ELÉCTRICOS Y ELECTRÓNICOS.

7.8. CLÁUSULAS DE ÍNDOLE LEGAL

En estas cláusulas se delimitan las condiciones en las que ambas partes podrán rescindir el contrato de construcción del equipo electrónico objeto del proyecto.

Modificaciones de obra. El diseño del equipo electrónico podrá ser cambiado total o parcialmente por la empresa contratista, no obstante si la empresa proyectista se considera perjudicada en sus intereses, solicitará la indemnización a que se considere acreedora, y cuya estimación someterán las partes a la decisión de la comisión arbitral. En los casos de suspensión no correrá el plazo.

Derecho de rescisión. La empresa proyectista podrá rescindir el contrato en los siguientes casos:

1. Cuando las variaciones introducidas en el equipo electrónico aumenten o disminuyan el importe total de ésta de un 20%.
2. Cuando por razones ajenas a la empresa proyectista pase más de un año sin poder trabajar en el equipo electrónico.
3. Cuando se retrase más de seis meses el pago de alguno de los pagos valorados estipulados.

Rescisión por incumplimiento del contrato. En el caso de retraso injustificado sobre los plazos fijados se impondrá a la empresa proyectista una multa de 1,5% del presupuesto asignado como pago valorado.

Liquidación en caso de rescisión. Se hará una liquidación única que será la definitiva con arreglo a lo estipulado en este pliego.

Cuestiones no previstas o reclamaciones. Todas las cuestiones que pudieran surgir sobre interpretación, perfeccionamiento y cumplimiento de las condiciones del contrato entre ambas partes serán resueltas por la comisión arbitral.

La comisión arbitral deberá dictar resolución después de oídas las partes dentro de los quince días siguientes al planteamiento del asunto ante la misma. Durante este plazo, la empresa proyectista deberá acatar las órdenes de trabajo indicadas por la empresa contratista sin perjuicio de proclamar las indemnizaciones correspondientes si la resolución le fuese favorable. Entre las resoluciones dictadas por la comisión arbitral figurará en todo caso la proposición en que cada una de las partes deberá participar en el abono de los honorarios de las personas que forman la comisión y de los peritos cuyo informe haya sido solicitado por ella.

7.9. CONCLUSIONES DEL PLIEGO DE CONDICIONES

Las partes interesadas manifiestan que conociendo los términos de este Pliego de Condiciones y del proyecto adjunto, y están de acuerdo con lo que en él se manifiesta.¹⁸

¹⁸ Regulador para un sistema de climatización de suelo radiante.- Fausto J. de Lucas

8. CONCLUSIONES

8.1. OBJETIVOS LOGRADOS

La primera parte de nuestros objetivos se ha conseguido realizar. Hemos aprendido a trabajar con una plataforma OpenSource y a partir de ella controlar motores y todo tipo de sensores.

La segunda parte de nuestros objetivos ha encontrado un gran problema como ya hemos comentado a lo largo de este proyecto, la problemática de que un robot vaya recto. Hemos buscado varias soluciones sin conseguir los resultados deseados. A continuación se ofrecen varias ideas para intentar conseguir este propósito.

8.2. CONCLUSIONES FINALES

Como primera y más importante conclusión, podemos decir que la robótica móvil tiene una gran problemática, el hecho de que un robot vaya recto de forma independiente y sin control humano.

Como segunda conclusión podemos decir que con un pequeño microprocesador, es posible controlar una gran cantidad de componentes, desde motores hasta una LCD.

A pesar de los recursos materiales y económicos de los que hemos dispuesto se ha podido realizar un robot cuyas características principales son:

- La velocidad de los motores está controlada mediante PWM.
- El robot puede seguir una pared situada a la izquierda sin problemas, en trayecto recto respecto a la pared.
- Puede realizar giros a la derecha de manera precisa, gracias a los sensores situados a la izquierda del chasis.
- Puede realizar giros a la izquierda. Los giros son siempre de alrededor de 90° gracias al control que se realiza de los motores dependiendo de la carga de las baterías.
- Puede buscar la pared más cercana y después seguirla.
- Puede evitar obstáculos gracias a los sensores. El robot es capaz de mirar a la derecha gracias al giro de uno de los sensores acoplado a un servo-motor.
- Se muestran mensajes de error dependiendo del estado de las baterías.
- Se muestran mensajes de funcionamiento y error mediante una pantalla LCD.
- El robot graba mensajes de funcionamiento y error en una tarjeta SD.

- Dispone de un reloj real que almacena la fecha y la hora actual.

8.3. FUTURAS MEJORAS

En este apartado enunciamos posibles mejoras para solucionar los problemas surgidos. Estas mejoras deberían ser estudiadas detenidamente para ver su viabilidad.

A continuación citamos varias ideas para mejorar el posible problema de la orientación sobre el plano:

8.3.1. GIROSCOPIO/BRÚJULA ELECTRÓNICA

El giróscopo o giroscopio es un dispositivo mecánico que sirve para medir, mantener o cambiar la orientación en el espacio de algún aparato o vehículo.

Está formado esencialmente por un cuerpo con simetría de rotación que gira alrededor del eje de dicha simetría. Cuando el giróscopo se somete a un momento de fuerza que tiende a cambiar la orientación de su eje de rotación, tiene un comportamiento aparentemente paradójico, ya que cambia de orientación (o experimenta un momento angular en todo caso, si está restringido) girando respecto de un tercer eje, perpendicular tanto a aquel respecto del cual se lo ha empujado a girar, como a su eje de rotación inicial. Si está montado sobre un soporte Cardano, que minimiza cualquier momento angular externo, o si simplemente gira libre en el espacio, el giróscopo conserva la orientación de su eje de rotación ante fuerzas externas que tiendan a desviarlo, mejor que un objeto no giratorio; se desvía mucho menos, y en una dirección diferente.¹⁹

8.3.2. ACELERÓMETRO

Se denomina acelerómetro a cualquier instrumento destinado a medir aceleraciones. Éstas no son necesariamente las mismas que las aceleraciones de coordenadas (cambio de la velocidad del dispositivo en el espacio), sino que es el tipo de aceleración asociada con el fenómeno de peso experimentada por una masa de prueba que se encuentra en el marco de referencia del dispositivo. Un ejemplo en el que este tipo de aceleraciones son diferentes es cuando un acelerómetro medirá un valor sentado en el suelo, ya que las masas tienen un peso, a pesar de que no hay cambio de velocidad. Sin embargo, un acelerómetro en caída gravitacional libre hacia el centro de la Tierra mida un valor de cero, ya que, a pesar de que su velocidad es cada vez mayor, está en un marco de referencia en el que no tiene peso

¹⁹ Giróscopo – Wikipedia.

Actualmente es posible construir acelerómetros de tres ejes (X,Y,Z) en un solo chip de silicio, incluyendo en el mismo la parte electrónica que se encarga de procesar las señales.

El principio de operación de los dispositivos, acelerómetros e inclinómetros de tecnología MEMS, están basados en el traspaso térmico por convección natural.

Estos dispositivos miden cambios internos de la transferencia de calor causada por la aceleración, ofreciendo ventajas significativas sobre el empleo de una estructura tradicional sólida de masas de prueba.

Ya que la masa de prueba en el diseño de los sensores MEMS son moléculas de gas, las estructuras móviles mecánicas son eliminadas dentro del acelerómetro.²⁰

8.3.3. SENSORES LÁSER

El láser es el único de los sensores que realiza la medida sin contacto, y en consecuencia, sin rozamiento.

Los láseres de triangulación proyectan un haz de luz sobre la superficie a medir. Este haz es reflejado en el fotodetector del aparato con un cierto ángulo de inclinación, que variará en función de la distancia medida.²¹

8.3.4. ACOPLAR CÁMARA WEB

Mediante una cámara web o VGA se podrían controlar los movimientos del robot, pero dejaría de ser autónomo.

8.3.5. ACOPLAR MÓDULO WiFi o GSM

Mediante la inclusión de un módulo WiFi o GSM podríamos tener el control (si es mediante cámara web) o tratar toda la información que el robot está manejando.

²⁰ Acelerómetro – Wikipedia.

²¹ Sensores de distancia.

Otras posibles mejoras:

8.3.6. CONTROL MEDIANTE PÁGINA WEB O DISPOSITIVOS MÓVILES

Este control podría realizarse mediante una aplicación web o para dispositivo móvil. La aplicación recibiría toda la información del robot, la mostraría al usuario y éste podría tomar decisiones dependiendo de los datos recibidos.

8.3.7. SUSTITUCIÓN DE ARDUINO MEGA POR DOS ARDUINOS DUEMILANOVE

Para mejorar la velocidad de procesamiento se puede sustituir el Arduino Mega por dos Arduinos Duemilanove. De esta manera se tendrían dos microprocesadores (comunicación en serie - Rx Tx-) que podrían funcionar en paralelo y por lo tanto realizar las operaciones de forma más rápida.

9. ANEXOS

9.1. ENLACES

Arduino: <http://arduino.cc/>

Arduino Mega: <http://arduino.cc/es/Main/ArduinoBoardMega>

Wiring: <http://wiring.org.co/>

Processing: <http://www.processing.org/>

Crear tu propia placa arduino: <http://arduino.cc/es/Main/Policy>

Página Web Labcenter: <http://www.labcenter.com/index.cfm>

Descarga Demo Proteus:

http://www.labcenter.com/download/prodemo_download.cfm#professional

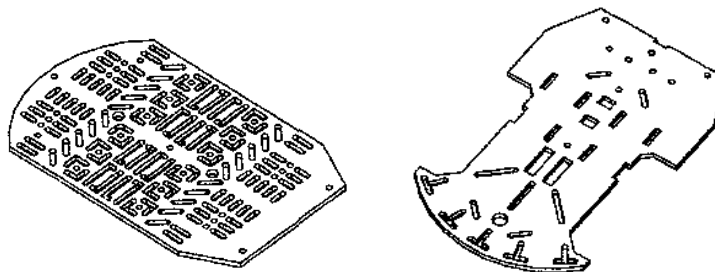
Precios Proteus: <http://www.labcenter.com/ordering/cprices.cfm>

Fritzing: <http://www.fritzing.org>

9.2. ESQUEMAS Y CONEXIONADO

9.2.1. DESPIECE MECÁNICO

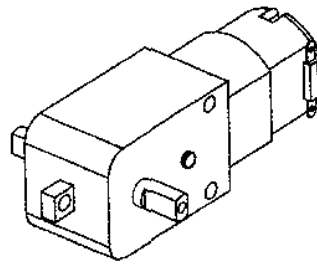
CHASIS



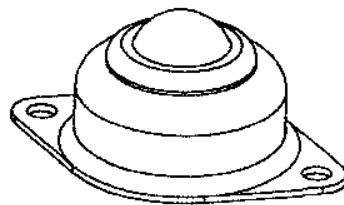
SUJECIÓN DE MOTOR



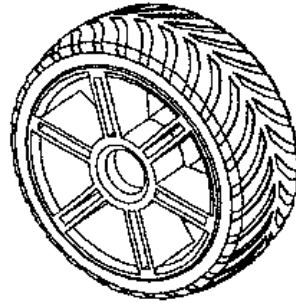
MOTOR



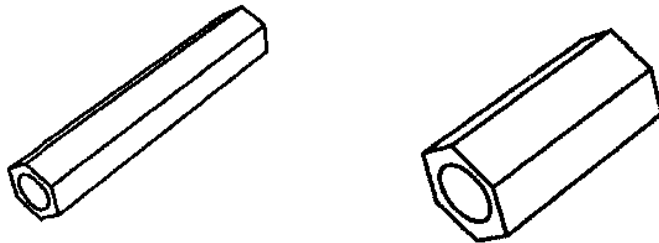
RUEDA OMNIDIRECCIONAL



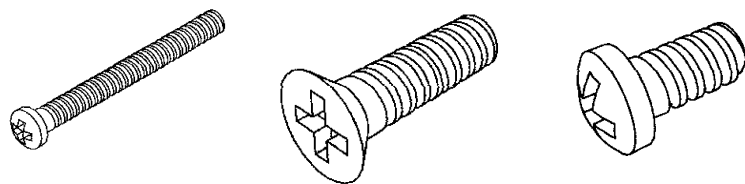
RUEDA

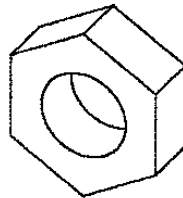


ESPACIADORES (L25, L10)



TORNILLOS (M3*30, M3*10, M3*6)



TUERCA M3

9.2.2. CONEXIONADO CON ARDUINO

MOTORES CC

<i>ARDUINO</i>	<i>PLACA CONTROL MOTORES</i>
PIN 12 (PWM)	PIN 1 MOTOR IZQUIERDO
PIN 31	PIN 2 MOTOR IZQUIERDO
PIN 30	PIN 3 MOTOR IZQUIERDO
PIN 13 (PWM)	PIN 1 MOTOR DERECHO
PIN 32	PIN 2 MOTOR DERECHO
PIN 33	PIN 3 MOTOR IZQUIERDO

ULTRASONIDOS DELANTERO

<i>ARDUINO</i>	<i>PING 3</i>
PIN 42	PIN TRIG
PIN 43	PIN ECHO
GND	GND
VCC	VCC

ULTRASONIDOS IZQUIERDO DELANTERO

<i>ARDUINO</i>	<i>PING 1</i>
PIN 46	PIN TRIG
PIN 47	PIN ECHO
GND	GND
VCC	VCC

ULTRASONIDOS IZQUIERDO TRASERO

<i>ARDUINO</i>	<i>PING 2</i>
PIN 26	PIN TRIG
PIN 27	PIN ECHO
GND	GND
VCC	VCC

SERVOMOTOR

<i>ARDUINO</i>	<i>MYSERVO</i>
PIN 9 (PWM)	CONTROL (CABLE BLANCO)
GND	GND (CABLE NEGRO)
VCC	VCC (CABLE ROJO)

LECTOR DE TARJETAS

<u>ARDUINO</u>	<u>LECTOR DE TARJETAS</u>
PIN 50	MISO
PIN 51	MOSI
PIN 52	SCK
PIN 53	SS
GND	GND
VCC	VCC

RELOJ REAL

<u>ARDUINO</u>	<u>RELOJ REAL</u>
PIN 20	SDA
PIN 21	SCL
GND	GND
VCC	VCC

LCD

<i>ARDUINO</i>	<i>LCD</i>
PIN 0	PIN D7
PIN 1	PIN D6
PIN 4	PIN D5
PIN 5	PIN D4
PIN 6	ENABLE
PIN 7	RS
GND	RW
	VSS
	V0
	CATODO (K)
VCC	VDD
	ÁNODO (A)

Como Arduino Mega sólo nos proporciona 3 pines de GND y VCC, hemos tenido que ampliar el número de pines para conectar todos los dispositivos:

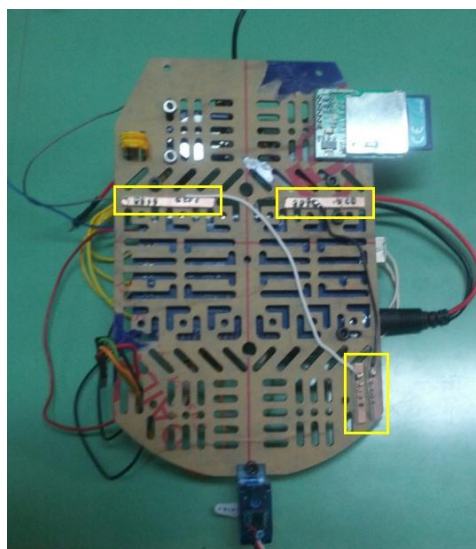


FIGURA 96: AMPLIACIÓN DE PINES GND Y VCC

9.2.3. DIBUJO DE LA BATERÍA EN LA LCD

Hemos creado los diferentes dibujos de la batería mediante código ASCII. Estos dibujos aparecen en la parte inferior derecha de la pantalla LCD. Aquí podemos ver los utilizados en nuestro programa:

```
byte bajo [8]=
{
    B00000,
    B01111,
    B01111,
    B01111,
    B01111,
    B01111,
    B01111,
    B01111,
    B00000
```

```
};
byte centro [8]=
{
    B00000,
    B11111,
    B11111,
    B11111,
    B11111,
    B11111,
    B11111,
    B11111,
    B00000
```

```
};
byte alto [8]=
{
    B00000,
    B11100,
    B11100,
    B11111,
    B11111,
    B11100,
    B11100,
    B11100,
    B00000
```

```
};
byte bajovacio [8]=
{
    B00000,
    B01111,
    B01000,
    B01000,
    B01000,
    B01000,
    B01000,
    B01111,
    B00000
```

```
byte centrovacio [8]=
```

```
{
    B00000,
    B11111,
    B00000,
    B00000,
    B00000,
    B00000,
    B11111,
    B00000
};
```

```
byte altovacio [8]=
```

```
{
    B00000,
    B11100,
    B00100,
    B00111,
    B00111,
    B00100,
    B11100,
    B00000
};
```

9.3. CONTENIDO DEL CD

9.3.1. ARDUINO (SOFTWARE)

Arduino-1.5.2-windows.exe

9.3.2. DEMO PROTEUS (SOFTWARE)

Prodemo.exe

9.3.3. DATASHEET COMPONENTES

Atmel 1280.pdf

L293D.pdf

Lcd: dip162-de.pdf

Motores CC: DG01D datasheet.pdf

Normativa diseño: IPC-2221.pdf

Lector de tarjetas SD: cd74hc4067_texas.pdf

Servomotor: AAS-309BB.pdf

Ultrasonidos: HC-SR04.pdf SRF_04(ultrasonido).pdf

Reloj Real: DS1307 datasheet.pdf

9.3.4. CÓDIGOS

Código Robot Perimetral.txt

Configuración Inicial Reloj RTC.txt

9.3.5. LECTURAS SD

6 lecturas de la Tarjeta SD durante el funcionamiento del Robot.

9.3.6. FOTOS

34 Fotos de diferentes ángulos del robot, así como de su proceso de construcción y de diferentes detalles.

9.3.7. IMÁGENES

Control de baterías.

Descarga baterías.

Esquema de Pines del Arduino Mega.

Picos de corriente con el osciloscopio digital.

9.3.8. VÍDEOS

Diferentes videos del funcionamiento del robot.

9.3.9. FRITZING (SOFTWARE)

fritzing.2013.02.25.pc.exe

9.3.10. PLANOS

Hoja 1: Esquema del control de los motores.

Hoja 2: PCB con bordes, pistas y componentes.

Hoja 3: PCB con bordes y pistas.

Hoja 4: PCB con serigrafía y bordes.

10. BIBLIOGRAFÍA

- [1] Introducción a la Robótica – Universidad de Chile
- [2] Introducción a la Robótica – Rafael Barea
- [3] OpenSource, OpenSoftware, OpenHardware – Wikipedia
- [4] Arduino website
- [5] Servomotores – Francisco A. Candelas Herías. Juan A. Corrales Ramón.
- [6] Sensores y acondicionadores – Enrique Mandado Pérez. Antonio Murillo Roldán.
- [7] LCD – Wikipedia
- [8] Lectores de tarjetas – Fransberns
- [9] Reloj en tiempo real – Wikipedia
- [10] Tecnología de las baterías
- [11] Tacómetro - Wikipedia
- [12] Encoders – Blog www.mecatronic.co
- [13] ¿Qué es el GPS? – Mateo P
- [14] Blog DiegoTechnology.es - Diego Romano Ubalde
- [15] Blog Tecnomatica: <http://lochnesh.wordpress.com/>
- [16] Comunicación Bus I2C – Eduardo J.Carletti
- [17] Serial Peripheral Interface (SPI) – Universidad Fermín Toro
- [18] Regulador para un sistema de climatización de suelo radiante.- Fausto J. de Lucas
- [19] Giróscopo – Wikipedia
- [20] Acelerómetro – Wikipedia
- [21] Sensores de distancia