

Instituto Tecnológico de Costa Rica

Carrera de Ingeniería Mecatrónica



**Diseño e Implementación de la Plataforma Móvil del robot basado en Smartphone,
ROBOBO**

**Informe de Proyecto de Graduación para optar por el título de Ingeniero en Mecatrónica
con el grado académico de Licenciatura**

Moisés Elías Bautista Briceño

Cartago, Febrero de 2016

INSTITUTO TECNOLÓGICO DE COSTA RICA
CARRERA DE INGENIERÍA MECATRÓNICA
PROYECTO DE GRADUACIÓN
ACTA DE APROBACIÓN

Proyecto de Graduación defendido ante el presente Tribunal Evaluador como requisito para optar por el título de Ingeniero en Mecatrónica con el grado académico de Licenciatura, del Instituto Tecnológico de Costa Rica.

Miembros del Tribunal



M. Sc. Marta Vilchez Monge.

Profesor lector



Ing. William Marín Moreno.

Profesor lector

TEC | Tecnológico
de Costa Rica
Ingeniería Mecatrónica



Dr. Ing. Juan Luis Crespo Mariño

Profesor asesor

Los miembros de este Tribunal dan fe de que el presente trabajo de graduación ha sido aprobado y cumple con las normas establecidas por la Carrera de Ingeniería Mecatrónica

Cartago, 24 de Febrero de 2016

Declaro que el presente Proyecto de Graduación ha sido realizado enteramente por mi persona, utilizando y aplicando literatura referente al tema e introduciendo conocimientos propios.

En los casos en que he utilizado bibliografía, he procedido a indicar las fuentes mediante las respectivas citas bibliográficas.

En consecuencia, asumo la responsabilidad total por el trabajo de graduación realizado y por el contenido del correspondiente informe final.

Cartago, 24 de Febrero de 2016



Moisés Eñas Bautista Briceño

Cédula: 800860658

TEC | Tecnológico
de Costa Rica
Ingeniería Mecatrónica

Resumen.

En el presente documento se presenta el diseño e implementación de una plataforma móvil llamada ROBOBO 2.0, realizado en el Grupo Integrado de Ingeniería de la Universidade da Coruña. Esta plataforma robótica se utilizará en el proyecto con fondos europeos llamado DREAM (Deferred Restructuring of Experience in Autonomous Machines), el proyecto DREAM consiste en aplicar el proceso cognitivo del sueño en la arquitectura de los robots autónomos.

En este proyecto se desarrolló el control de velocidad y posición para el motor CD, el módulo de comunicación por medio de Bluetooth, el sistema de movimiento Pan/Tilt del Smartphone, el funcionamiento de los driver para LED y motores y el firmware para el ROBOBO 2.0.

Capítulo 1: se presenta la introducción del proyecto, el en cual se muestra el entorno del proyecto, definición del problema y el enfoque de la solución; este apartado se enfoca en dar una perspectiva general del proyecto. Capítulo 2: se definen los objetivos del proyecto. Capítulo 3: en este apartado se hace un repaso de los conceptos de robótica y se presentan algunos tipos de robot. Capítulo 4: aparece algunas características el primer prototipo del ROBOBO. Capítulo 5: en este capítulo se explica el proceso con el que se desarrolló las soluciones. Capítulo 6: en este capítulo se realiza la explicación de los conceptos teóricos utilizados para el desarrollo de la solución. Capítulo 7: en este apartado se desarrolla las soluciones del proyecto, se muestra los resultados de la configuración del microcontrolador, el diseño del controlador de velocidad, módulo de comunicación Bluetooth, el desarrollo del sistema Pan/Tilt, el driver para LED TLC5941 y finalmente la explicación del firmware del ROBOBO 2.0. Capítulo 8: se presentan las conclusiones del proyecto. Capítulo 9: se presentan las recomendaciones y el posible trabajo futuro que mejorarán el producto final.

Palabras claves: Robótica móvil, robótica en investigación, robótica educativa, ROBOBO, Smartphone, UART, Bluetooth, Controlador PI, movimiento Pan/Tilt.

Abstract.

This document shows the design and implementation of a mobile platform called ROBOBO 2.0, made in the Integrated Engineering Group of the University of Corunna. This robotic platform will be used in the project with European funds called DREAM (Deferred Restructuring of Experience in Autonomous Machines). The DREAM project will apply the cognitive process of sleep in the architecture of autonomous robots.

In this project the speed control and position for the DC motor, the communication module through Bluetooth, the movement system of Smartphone Pan/Tilt, the operation of the driver for LED and motors and finally the firmware for ROBOBO 2.0 were developed.

Chapter 1: It shows the introduction of the project, which shows the project environment, it also defines the problem and focus of the solution; this section focuses on giving an overview of the project. Chapter 2: The objectives of the project were defined. Chapter 3: This section reviews about the concepts of robotics and some types of robots that are presented. Chapter 4: In this chapter some characteristics of the first prototype of the ROBOBO are presented. Chapter 5: In this chapter process about obtaining solutions is explained. Chapter 6: In this chapter the explanation of the theoretical concepts used for the development of the solution is made. Chapter 7: This section develops the solutions of the project, it shows the results for the configuration of the microcontroller, and also another configurations including the design of the speed controller, Bluetooth module communication, the development of Pan/Tilt system, the driver for LED TLC5941 and the development of the firmware for the ROBOBO 2.0. Chapter 8: The project conclusions are presented. Chapter 9: the recommendations and possible future work that will enhance the final product are presented.

Keywords: Mobile robotics, investigational robotics, educational robotics, ROBOBO, Smartphone, UART, Bluetooth, PI controller, motion system Pan/Tilt.

Dedicatoria.

Dedico este trabajo especialmente a mi madre Maritza Briceño Morales y a mi hermana Giselle Bautista Briceño, quienes me apoyaron durante todo mi proceso formativo y en todas las decisiones que he tomado. Así como a todos mis amigos que han estado junto a mí durante todos los momentos importantes de mi vida. Y a todos mis profesores y personas externas al sector académico que han contribuido con sus conocimientos y experiencia a mi desarrollo profesional y personal.

Agradecimiento.

Agradezco al Tecnológico de Costa Rica por la excelente formación académica, además de permitirme la oportunidad de realizar aportes en el área científico tecnológico. Al Grupo Integrado de Ingeniería de la Universidad de Coruña por permitirme realizar este proyecto de graduación, así mismo a los miembros del equipo de trabajo por el apoyo, consejos y aportes para lograr la finalización de este proyecto. A los profesores de Tecnológico de Costa Rica: Dr. Ing. Juan Luis Crespo Mariño, M. Sc. Marta Vilchez Monge e Ing. William Marín Moreno, por formar parte del Tribunal Evaluador y la asesoría para la elaboración del Informe Final.

Tabla de contenido

Resumen.....	iii
Abstract.....	iv
Dedicatoria.....	v
Agradecimiento.....	vi
Capítulo 1: Introducción.....	1
1.1. Entorno del proyecto.....	1
1.2. Definición del problema.....	2
1.2.1. Generalidades del problema.....	2
1.2.2. Síntesis del problema.....	3
1.3. Enfoque de la solución.....	5
Capítulo 2: Objetivos.....	8
2.1. Objetivo General.....	8
2.2. Objetivos específicos.....	8
Capítulo 3: Estado del arte.....	9
3.1. Robótica.....	9
3.2. Robots de docencia e investigación.....	14
Capítulo 4: Antecedentes del ROBOBO.....	21
4.1. Sensores utilizados.....	21
4.2. Sistema Motriz.....	22
4.3. Sistema de comunicación.....	22
4.4. Sistema LED.....	22
4.5. Sistema de alimentación.....	23
Capítulo 5: Enfoque metodológico.....	24

5.1.	Investigación y recopilación de datos.....	24
5.2.	Definición de los problemas y mejoras a implementar.	25
5.3.	Diseño e implementación de las soluciones.	25
5.4.	Verificación del comportamiento.	26
5.5.	Rediseño y cambio de componentes.....	26
Capítulo 6:	Marco teórico.	28
6.1.	Microcontrolador.	28
6.1.1.	Arquitecturas.	29
6.1.2.	PIC32MX534F064H.	30
6.1.3.	Características del PIC32MX534F064H.....	31
6.2.	PicKit3.....	32
6.3.	Motor.	33
6.3.1.	Principio de funcionamiento del motor CD.	33
6.3.2.	Micro metal gearmotor.....	34
6.4.	Driver del motor	35
6.4.1.	Puente H.	35
6.4.2.	Driver DRV8833.	36
6.5.	Encoder.....	39
6.5.1.	Sensor efecto Hall.	39
6.5.2.	Encoder Magnético.	39
6.6.	Controlador automático.....	40
6.6.1.	Controlador PI.....	42
6.7.	Protocolos de comunicación.....	43
6.7.1.	UART.....	43
6.7.2.	SPI.....	44

6.8.	Bluetooth.....	45
6.9.	Driver LED.....	46
6.9.1.	Modos de funcionamiento.....	48
Capítulo 7:	Descripción detallada de la solución.....	50
7.1	Configuración PWM.....	50
7.1.1.	Consideraciones para el PWM.....	51
7.2.	Funcionamiento del driver para motor DRV8833.....	53
7.2.1.	Modos de funcionamiento del driver DRV8833.....	55
7.3.	Pruebas con los motores.....	58
7.3.1.	Caracterización de los motores.....	58
7.3.2.	Pruebas en vacío.....	62
7.3.3.	Prueba en el suelo en línea recta.....	63
7.4.	Funcionamiento y configuración del encoder.....	67
7.4.1.	Implementación del encoder.....	67
7.5.	Diseño del controlador.....	68
7.5.1.	Modelado Teórico.....	69
7.5.2.	Modelado empírico.....	70
7.5.3.	Controladores.....	80
7.5.4.	Implementación del controlador.....	84
7.6.	Funcionamiento del Bluetooth.....	86
7.7.	Sistema Pan/Tilt.....	88
7.8.	Funcionamiento del driver para LED TLC5941.....	92
7.9.	Firmware del ROBOBO 2.0.....	93
Capítulo 8:	Conclusiones.....	95
Capítulo 9:	Recomendaciones.....	96

Bibliografía.....	97
Apéndices.....	100
Apéndice A.1: Glosario, abreviaturas y simbologías.....	100
Apéndice A.2: Resultados del PWM.....	102
Apéndice A.3: Comandos AT.....	107
Apéndice A.4: Esquemáticos y PCB.....	108
Anexos.....	111
Anexo B.1: Datos técnicos de los componentes utilizados.....	111
Anexo B.2: Esquemáticos de los componentes utilizados.....	116
Anexo B.3: Planos del gearmotor HP.....	118

Índice de Figuras.

Figura 1.1.	Concepto desarrollado por GII del ROBOBO. (GII, 2015).....	3
Figura 1.2.	Capacidad de sensorización del ROBOBO. (GII, 2015).	4
Figura 1.3.	Actuadores del ROBOBO. (GII, 2015).	4
Figura 1.4.	Funcionalidades de conectividad del ROBOBO. (GII, 2015).	5
Figura 3.1.	Robots industriales típicos. (Springer, 2008).	10
Figura 3.2.	Disposición de algunos sensores de entorno en los vehículos inteligentes. (Springer, 2008).	11
Figura 3.3.	Algunos ejemplos de procesamiento de imagen de los vehículos inteligentes. (a) Detección de vía. (b) Detección de vehículos. (Springer, 2008).	12
Figura 3.4.	Algunas funcionalidades de los vehículos inteligentes. (a) Parqueo asistido. (b) Cambio de vía. (Springer, 2008).	12
Figura 3.5.	Robots humanoides. (a) ASIMO. (b) NAO. (c) Actroid.	13
Figura 3.6.	Plataformas similares al ROBOBO en el sector educativo. (a)WHEELPHONE. (b) TYCHE. (c)MICROROVER. (d) ROMO. (e) SMARTBOT.	17
Figura 3.7.	Plataformas similares al ROBOBO en el sector de investigación. (a) ELISA-3.	19
Figura 4.1.	Primer prototipo realizado, ROBOBO 1.0. (Naya, 2013).	21
Figura 4.2.	Sensor infrarrojo VCNL4000. (Naya, 2013).	22
Figura 4.3.	Reguladores utilizados en el ROBOBO 1.0. (a) DC-DC TSR1-2450 (b) MCP1700.	23
Figura 5.1.	Enfoque metodológico utilizado para el desarrollo del proyecto.	27
Figura 6.1.	Estructura básica de un microcontrolador. (Valdes & Pallás, 2007).	29
Figura 6.2.	Arquitecturas de organización de memoria. (a) von Neumann. (b) Harvard.	29
Figura 6.3.	MINI32 utilizado en las pruebas. (MiKroEleKtroniKa, 2015).....	30
Figura 6.4.	Esquemático del circuito impreso del MINI32. (MiKroEleKtroniKa, 2015).....	31
Figura 6.5.	Diagrama de conexión del PicKit3.	32
Figura 6.6.	Representación de un motor CD. (Wildi, 2007).	33
Figura 6.7.	Motor reductor utilizado en el ROBOBO para las pruebas. (POLOLU, 2015).	35
Figura 6.8.	Escobillas utilizadas como conmutador en el gearmotor. (POLOLU, 2015).	35
Figura 6.9.	Esquemático de un puente H. (Texas Instruments, 2015).	36

Figura 6.10.	Esquemático simplificado del driver DRV8833. (Texas Instruments, 2015).....	38
Figura 6.11.	Encoder por cuadratura utilizado en el prototipo del ROBOBO.....	40
Figura 6.12.	Esquemático del encoder.....	40
Figura 6.13.	Diagrama de bloques de un sistema de control. (Ogata, 2010).	41
Figura 6.14.	Estructura de datos de la comunicación UART. (a) 8 bits. (b) 9 bits.	44
Figura 6.15.	Conexión utilizando el protocolo de comunicación SPI.....	44
Figura 6.16.	Módulo Bluetooth HC-06.....	46
Figura 6.17.	Pines del Driver TLC5941.....	47
Figura 6.18.	Estructura de los datos en el modo DC.....	48
Figura 6.19.	Estructura de los datos en el modo GS.....	49
Figura 7.1.	Diagrama de bloques de los osciladores.....	51
Figura 7.2.	Proceso de la formación de la señal del PWM.....	52
Figura 7.3.	Pines del driver DRV8833.....	53
Figura 7.4.	Señal obtenida al bloquear el motor.....	54
Figura 7.5.	Esquemático de la configuración utilizado para el giro de los motores.....	55
Figura 7.6.	Señal de salida en modo <i>fast decay</i>	56
Figura 7.7.	Señal de salida en modo <i>slow decay</i>	57
Figura 7.8.	Motor CD y caja reductora utilizada en el ROBOBO 2.0.....	59
Figura 7.9.	Montaje de los motores, driver, MINI32 y ruedas para las pruebas en vacío.....	62
Figura 7.10.	Diagrama implementado en la prueba de los motores en el suelo.....	64
Figura 7.11.	Rampa utilizada en la prueba de los motores.....	65
Figura 7.12.	Modelo en 3D de la disposición del eje del motor y el eje de la rueda.....	66
Figura 7.13.	Implementación real del sistema de transmisión.....	66
Figura 7.14.	Forma de onda de la señal del encoder utilizado en los motores 75:1.....	67
Figura 7.15.	Representación del motor DC.....	69
Figura 7.16.	Forma de onda obtenida como salida del encoder en la primera prueba.....	71
Figura 7.17.	Diagrama empleado para procesar los datos del encoder.....	72
Figura 7.18.	Forma de onda obtenida como salida del encoder en la segunda prueba.....	74
Figura 7.19.	Forma de onda obtenida a la salida del encoder en la tercera prueba.....	75
Figura 7.20.	Respuesta ante un escalón con valor de 6, las pruebas 1 y 2.....	78
Figura 7.21.	Respuesta ante un escalón con valor de 6, las pruebas 3 y 4.....	78

Figura 7.22.	Respuesta ante un escalón con valor de 6, las pruebas 5 y 6.....	78
Figura 7.23.	Coincidencia de las funciones de transferencia con los datos de la primera prueba.	79
Figura 7.24.	Lugar de las raíces de la función de transferencia del motor.	80
Figura 7.25.	Región donde se deben ubicar los polos dominantes.	81
Figura 7.26.	Respuesta del sistema con un controlador Klag.	82
Figura 7.27.	Respuesta del sistema con un controlador Proporcional e Integral.	83
Figura 7.28.	Respuesta del sistema con un controlador Integral.....	83
Figura 7.29.	Sistema Pan/Tilt.....	89
Figura 7.30.	Vista explosionada del sistema Pan/Tilt.	89
Figura 7.31.	Modo de operación del Pan.	90
Figura 7.32.	Modo de operación del Tilt.....	91
Figura 7.33.	Corte transversal del sistema Pan/Tilt.	91
Figura 7.34.	Diagrama de flujo en el programa principal.	93
Figura 7.35.	Aspecto físico del ROBOBO 2.0.....	94
Figura A.1.	PWM a 50% de ciclo de trabajo y a 50 kHz.....	102
Figura A.2.	PWM a 60% de ciclo de trabajo y a 100 kHz.....	103
Figura A.3.	PWM a 99% de ciclo de trabajo y a 100 kHz.....	103
Figura A.4.	PWM a 1% de ciclo de trabajo y a 100 kHz.....	104
Figura A.5.	Dos PWM a 100 kHz y con ciclo de trabajo de 40% y 60%.	104
Figura A.6.	Dos PWM a 1 kHz y con ambos ciclo de trabajo de 50%.	105
Figura A.7.	Dos PWM a 10 kHz y con ciclo de trabajo de 40% y 60%.	105
Figura A.8.	Dos PWM a 100 kHz, ambos con ciclo de trabajo de 74%.	106
Figura A.9.	Esquemático utilizado para las pruebas con los motores, utilizando solo el PWM.	108
Figura A.10.	Esquemático utilizado en las pruebas de los motores, controlados por Bluetooth.	109
Figura A.11.	Esquemático utilizado en las pruebas del driver TLC5941.	110
Figura A.12.	PCB fabricado para las pruebas de los LED.....	110

Figura B.1.	Información técnica del PIC32MX534F064H. (Microchip, 2015).....	111
Figura B.2.	Información técnica del driver DRV8833. (Texas Instruments, 2015).....	112
Figura B.3.	Información técnica del módulo Bluetooth HC-06. (OLIMEX, 2016).....	113
Figura B.4.	Información técnica del driver TLC5941. (Texas Instruments, 2015)	114
Figura B.5.	Información técnica de los motores CD. (POLOLU, 2015).	115
Figura B.6.	Esquemático del driver DRV8833. (Texas Instruments, 2015).	116
Figura B.7.	Esquemático de la conexión recomendada para el driver TLC5941. (Texas Intruments, 2015)	117
Figura B.8.	Pines del PICMX534F064H. (Microchip, 2015).	117
Figura B.9.	Planos del gearmotor HP. (POLOLU, 2015).	118

Índice de Tabla.

Tabla 3.1.	Comparación del ROBOBO con otras plataformas educativas.	18
Tabla 3.2.	Comparación del ROBOBO con otras plataformas educativas.	18
Tabla 3.3.	Comparación de plataformas de investigación similares al ROBOBO.....	20
Tabla 6.1.	Funcionalidades de salida del driver DRV8833.....	37
Tabla 6.2.	Protecciones del circuito en el driver DRV8833. (Texas Instruments, 2015).....	38
Tabla 7.1.	Verificación de la tensión de salida del driver sin motor.	57
Tabla 7.2.	Prueba de linealidad de la tensión de salida del driver.	58
Tabla 7.3.	Prueba del motor 75:1 en vacío y bloqueado.	60
Tabla 7.4.	Prueba del motor 150:1 en vacío y bloqueado.	61
Tabla 7.5.	Prueba de los motores en vacío con el acople de las llantas.	63
Tabla 7.6.	Funciones de transferencia empíricas.	77
Tabla 7.7.	Prueba de los controladores.....	85

Capítulo 1: Introducción.

1.1. Entorno del proyecto.

El Grupo Integrado de Ingeniería (GII) de la Universidade da Coruña (UDC) es un grupo interdisciplinar de investigación aplicada en ingeniería orientado a la transferencia de conocimiento y a la generación de nuevos productos en el entorno industrial. El GII se centra en dos ejes estratégicos, los cuales son: la generación de conocimiento científico y la transferencia de tecnología a la industria. La investigación científica del GII se estructura a través de las siguientes líneas: Ingeniería Naval, Dinámica de Fluidos, Inteligencia Computacional, Robótica, Sistemas de Medida y Sensorización y Organización Industrial. (GII, 2014a).

La línea de investigación de Robótica y Cognición tiene como objetivo general el de desarrollar sistemas reales que puedan responder con el mayor grado de autonomía posible a las condiciones cambiantes del medio, sin tener que precisar de la ayuda de un operador o programador para establecer las nuevas estrategias necesarias para la consecución de la tarea encomendada. De esta línea de investigación se derivada otra que se denomina Mecanismos Cognitivos, en la cual se centra en el estudio e implementación de funcionalidades cognitivas de alto nivel en robots autónomos, utilizando los procesos cognitivos humanos como inspiración para desarrollar un mecanismo cognitivo completo para un robot. (GII, 2014b).

Uno de los proyectos asociados a la línea de Mecanismos Cognitivos es DREAM (Deferred Restructuring of Experience in Autonomous Machines), financiado por la Unión Europea, y que se centra en la integración de procesos cognitivos basados en el sueño dentro de una arquitectura cognitiva para robots autónomos. El objetivo de utilizar estos procesos reside en la capacidad de consolidar el aprendizaje adquirido durante la operación en "tiempo de vida" que se ha visto que existe en los procesos asociados al sueño en el cerebro humano. De esta forma, la experiencia adquirida por el robot se analiza y procesa en otra escala temporal que proporciona nuevas representaciones de más alto nivel. (GII, 2014c) (DREAM, 2015).

Uno de los principales problemas a solucionar en el proyecto DREAM se centra en el aprendizaje a partir de la interacción con humanos. En este sentido, se requieren numerosos datos de interacción

en tiempo real en diversas situaciones para poder mejorar la arquitectura, y para lograrlos se ha diseñado la iniciativa "adopt a robot". Esta iniciativa se centra en proporcionar a los centros educativos un robot de bajo costo que contenga la arquitectura cognitiva desarrollada en el DREAM, de modo que los alumnos puedan interactuar con el robot libremente y se obtengan gran variedad de datos para mejorar dicha arquitectura. Este robot se va a desarrollar en el GII bajo el nombre de ROBOBO, y su diseño y fabricación son el principal objetivo del presente proyecto.

1.2. Definición del problema.

1.2.1. Generalidades del problema.

Desde hace varios años, el GII está desarrollando un robot autónomo denominado ROBOBO. Una versión mejorada de este robot se utilizará dentro de la iniciativa "adopt a robot" del proyecto DREAM, en la que dicho robot debe interactuar con los humanos y extraer información útil para su aprendizaje a largo plazo. El ROBOBO consta de una plataforma móvil sobre la que se conecta un Smartphone que hace las veces de unidad de procesado, sensorización y comunicación. Ya existe un primer prototipo del ROBOBO en la actualidad, pero este debe ser mejorado para dar soporte a la iniciativa "adopt a robot", que dará lugar al ROBOBO 2, cuya idea conceptual inicial se muestra en la Figura 1.1. (GII, 2015).

Se debe, por tanto, fabricar un prototipo del ROBOBO 2 que incluya las mejoras mecánicas y electrónicas del mismo, tales como: nuevo diseño mecánico con atril móvil, nueva ubicación de los elementos electrónicos y de alimentación, comunicación con Smartphone basada en Bluetooth, entre otras. Todas estas mejoras están condicionadas a que el prototipo debe tener un precio final muy ajustado.



Figura 1.1. Concepto desarrollado por GII del ROBOBO. (GII, 2015).

1.2.2. Síntesis del problema.

El problema a resolver se puede sintetizar en el diseño y fabricación de la unidad de procesamiento, sistema de comunicación y sistema mecánico para un prototipo mejorado de una plataforma robótica móvil de bajo coste a la que se le pueda conectar un Smartphone constituyendo así un robot autónomo con gran capacidad de sensorización, cálculo y comunicación, así como de actualización. En la Figura 1.2, Figura 1.3 y Figura 1.4 se muestran las características esperadas del ROBOBO en: Capacidad de sensorización, actuadores y conectividad respectivamente.

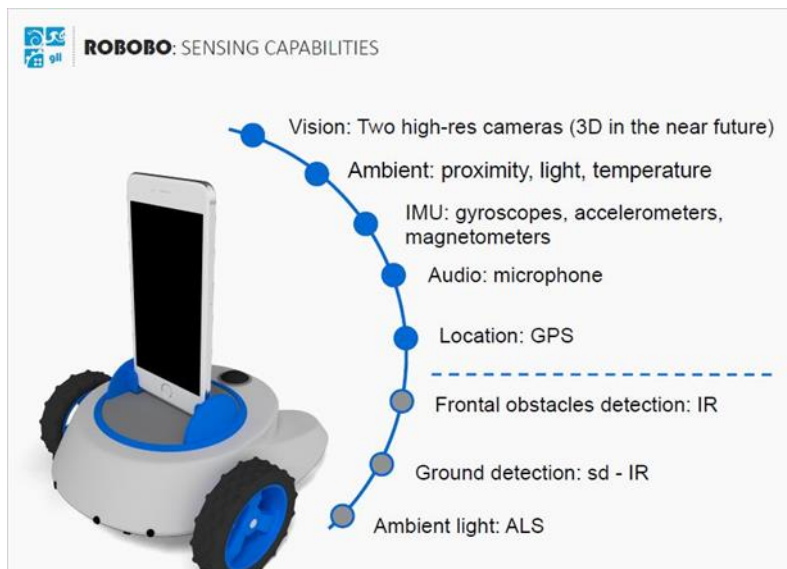


Figura 1.2. Capacidad de sensorización del ROBOBO. (GII, 2015).

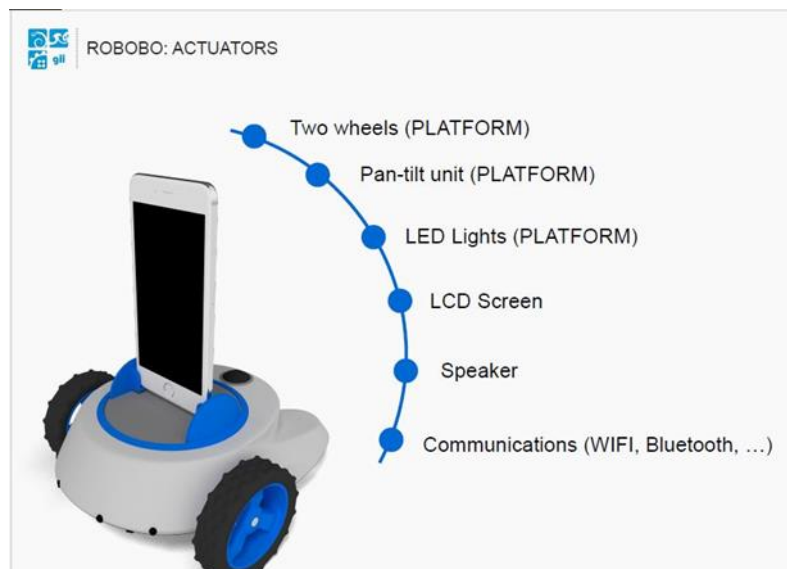


Figura 1.3. Actuadores del ROBOBO. (GII, 2015).

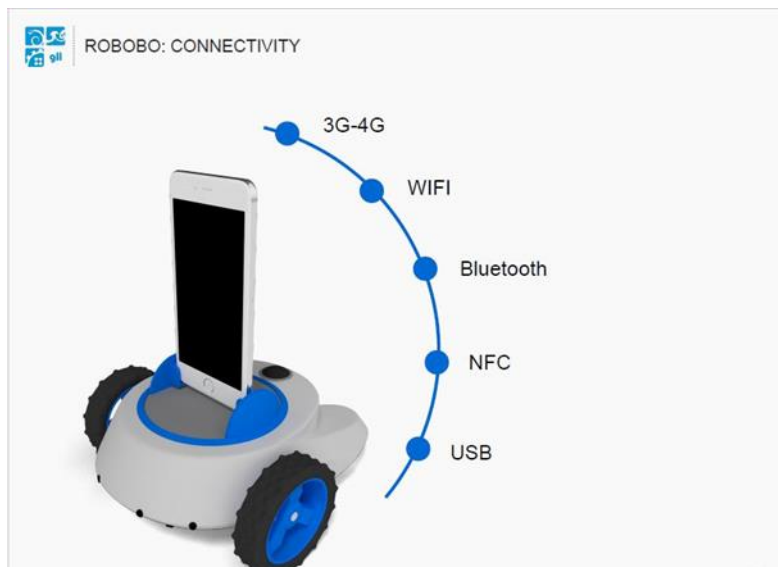


Figura 1.4. Funcionalidades de conectividad del ROBOBO. (GII, 2015).

1.3. Enfoque de la solución.

Para solucionar el problema se diseñará una plataforma que permita alojar y transportar un teléfono inteligente (smartphone en adelante), de modo que este proporciona al robot su sensorización (amplia, actualizada y en aumento), su gran capacidad de procesamiento. Con este planteamiento, el problema de la actualización del hardware se ve solucionado, ya que una vez un smartphone quede desfasado, simplemente se renueva con otro más actual. Además, este planteamiento permite hacer uso de la gran potencialidad de los sistemas operativos que se utilizan en los smartphones, en continua mejora y con gran cantidad de aplicaciones disponibles.

En el Grupo Integrado de Ingeniería de la Universidad de Coruña se comenzó el desarrollo de una nueva plataforma de este tipo, bautizada como ROBOBO, que tiene las siguientes características:

- Sensores en la plataforma robótica (independientes del Smartphone).
- Compatible con Android.
- Desarrollado en ROS (software en código libre).
- Hardware abierto (planos de fabricación de la electrónica disponible).

Se pretende fabricar un prototipo mejorado del ROBOBO para poder ser utilizado en el proyecto europeo DREAM. Para ello se parte de un nuevo diseño, ya en fase avanzada de conceptualización, basado en el diseño actual. Trabajando en colaboración con el equipo de mecatrónica y diseño del GII, se deberá manufacturar la estructura y carcasa de dicho prototipo mediante un proceso de manufacturación aditiva, seleccionar los componentes de terceros necesarios, programar la placa de control y realizar el montaje de todos los elementos electrónicos dentro de la estructura.

La solución se puede dividir en tres partes:

1. Parte mecánica: en esta se trabajó con los componentes móviles de la plataforma, así como los que mantienen fija la estructura, con esto se buscó que la plataforma pueda reaccionar tanto a estímulos externos como además comprobar los efectos de sus decisiones. Se puede dividir en los siguiente módulos:
 - Módulo de movimiento de traslación del ROBOBO: en este módulo, el robot podrá moverse sobre la superficie, podrá esquivar o mover objetos, desplazarse de un punto A hacia un punto B, entre otras acciones.
 - Módulo de movimiento del Smartphone: este módulo tiene la función de darle libertad de movimiento al Smartphone independientemente del movimiento de traslación de la plataforma. Los movimientos que puede realizar serían de rotación y variación del ángulo de inclinación del Smartphone (*Pan/Tilt*).
 - Módulo de sujeción del Smartphone: este módulo pretende asegurar que el Smartphone no se caiga o mueva debido a la interacción con el medio ambiente.
2. Parte Electrónica: esta comprende todos los componentes electrónicos que permiten a la plataforma tomar decisiones, así como comunicarse con el Smartphone. Se puede dividir en los siguientes módulos:
 - Módulo de procesado del sistema: en este se analiza la información de los distintos parámetros y posteriormente toma las decisiones correspondientes.
 - Módulo de comunicación: este módulo permitirá la transferencia de información entre la plataforma y el Smartphone y viceversa. Esta transferencia de datos se realizará por medio de Bluetooth.
 - Módulo de alimentación: este módulo comprende toda la parte de potencia que permitirá a la plataforma realizar las acciones deseadas.

3. Parte visual: en esta parte se evalúa la distribución de todos los componentes electrónicos y mecánicos, la carcasa que contiene todas las partes y además el aspecto visual final, con el objetivo de ser agradable para el usuario.

Capítulo 2: Objetivos

2.1. Objetivo General.

Diseñar e implementar los elementos mecatrónicos de una plataforma robótica móvil basada en Smartphone ROBOBO 2.0.

2.2. Objetivos específicos.

- Seleccionar nuevas soluciones de componentes electrónicos y mecánicos que puedan suponer una mejora con respecto a los empleados en la primera versión del ROBOBO.
- Crear un módulo de comunicación para la transferencia de datos entre la plataforma y el robot que permita el empleo de cualquier Smartphone con el sistema operativo Android.
- Fabricar un prototipo del PCB para la validación del diseño y la realización de pruebas.
- Adaptar la disposición de los componentes electrónicos y mecánicos al diseño de la plataforma robótica.
- Diseñar el módulo para conseguir el movimiento *Pan/Tilt* del soporte sobre el que se encuentra posicionado el Smartphone independiente del movimiento de la plataforma.

Capítulo 3: Estado del arte.

3.1. Robótica.

El concepto de robot fue claramente establecido por medio de numerosas realizaciones históricas dentro de las cuales sobresalen: el Gallo de la catedral de Estrasburgo en el año 1352, el León mecánico de Leonardo Da Vinci en el año 1499, la Muñeca mecánica capaz de dibujar de H. Maillardet en el año 1805, entre otros (Barrientos Cruz, 2007). Sin embargo la apariencia del robot que se conoce actualmente tuvo que esperar hasta los avances de la tecnología del siglo veinte. El término “robot”, que se deriva de la palabra eslava *robota*, que se refiere al trabajo subordinado, se utilizó por primera vez en la obra de teatro del escritor checo Karel Capek llamada *Rossum's Universal Robots (R.U.R)*. Uno de los mayores impulsores de la palabra robot fue el escritor ruso Isaac Asimov, quien en el año 1940 en su novela “Círculo vicioso” se propone la ética en la interacción entre robots y humanos por medio de las tres leyes de la robótica (Springer, 2008).

En la década de 1980, se definió a la robótica como la ciencia que estudia la conexión inteligente entre la percepción y la acción. Dentro de esta definición se utiliza la palabra acción para expresar ya sea la locomoción que le permite al robot moverse por el entorno como a la manipulación de los objetos o herramientas; de igual forma la percepción se refiere a los sensores que le permiten extraer información tanto del estado del robot como del medio donde este se encuentra; y finalmente el término conexión inteligente hace referencia a la arquitectura de programación, planificación y control (Springer, 2008). Por lo tanto un robot se define como un agente artificial mecánico guiado por un programa de control o circuito electrónico que posee sensores que le proporcionan información de su entorno y actuadores que le permiten modificar dicho entorno (Arkin, 1998).

Los robots pueden ser o no autónomos. Robots no autónomos serían, por ejemplo, los robots teledirigidos o los exoesqueletos, totalmente dependientes de un operador. En contraposición a estos encontramos los robots autónomos, cuyo sistema de control no requiere de supervisión humana.

Podemos distinguir distintas categorías de los robots según su campo de aplicación, algunas de estas son:

- Robots Industriales:** Los robots industriales combinan alta productividad, calidad y adaptabilidad con un costo mínimo, lo que ha permitido incrementar la competitividad en el área de manufactura, logrando grandes volúmenes de producción. Algunas aplicaciones industriales donde se emplean robots son: soldadura, ensamblaje de carros, pintura, transferencia automática de materiales, maquinado, asistencia a humanos en tareas manuales, entre otros. Los tipos de robots industriales más conocidos son los cartesianos, SCARA, articulado y paralelos, en la Figura 3.1 se muestran las estructuras básicas de cada uno así como el área de trabajo típica (Springer, 2008).

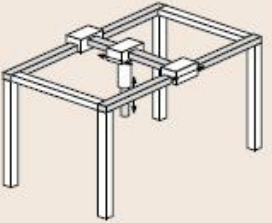
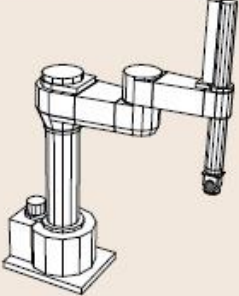
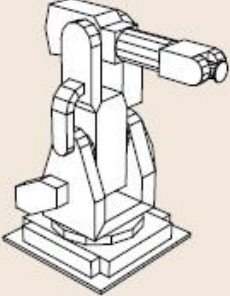
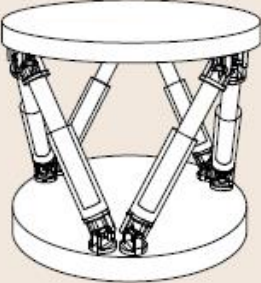
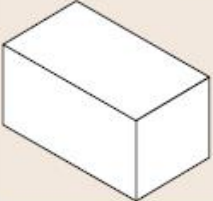
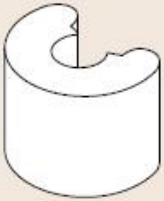






Gantry	SCARA	Serial articulated	Parallel
			
Form of workspace			
			
Product example			
			

Figura 3.1. Robots industriales típicos. (Springer, 2008).

- **Robots aéreos:** el término *aerial robotics* se le atribuye a Robert Michelson, quien buscó la manera de expresar la alta tecnología en pequeñas máquinas voladoras, sin embargo un robots aéreo se define como un sistema capaz de mantener el vuelo sin control humano directo y siendo capaz de realizar una tarea específica. Las aplicaciones de los robots aéreos en la actualidad son muy limitadas debido a las rigurosas regulaciones del espacio aéreo, en la actualidad se utilizan para: observación aérea (mapeo, identificación de objetivos, monitoreo de cultivos), operaciones militares y aplicaciones privadas (uso a pequeña escala, experimentos científicos) (Springer, 2008).
- **Vehículos inteligentes:** Los automóviles han sido uno de los productos más importantes del siglo 21, en los últimos 25 años ha sido un importante campo de aplicación de robótica. Un vehículo inteligente se define como un vehículo mejorado con dispositivos de percepción, razonamiento y de actuación que permite a la automatización de las tareas de conducción, tales como cambio de carril, evitar obstáculos, adelantar a otro vehículo, siguiendo a un carro que va delante, evaluar y evitando situaciones peligrosas y escoger el recorrido. En la Figura 3.2 se muestra algunas de las cámaras que emplean los vehículos inteligentes para monitorear el entorno, además en las Figura 3.3 y Figura 3.4 aparecen algunas funcionalidades que presentan estos modernos automóviles (Springer, 2008).

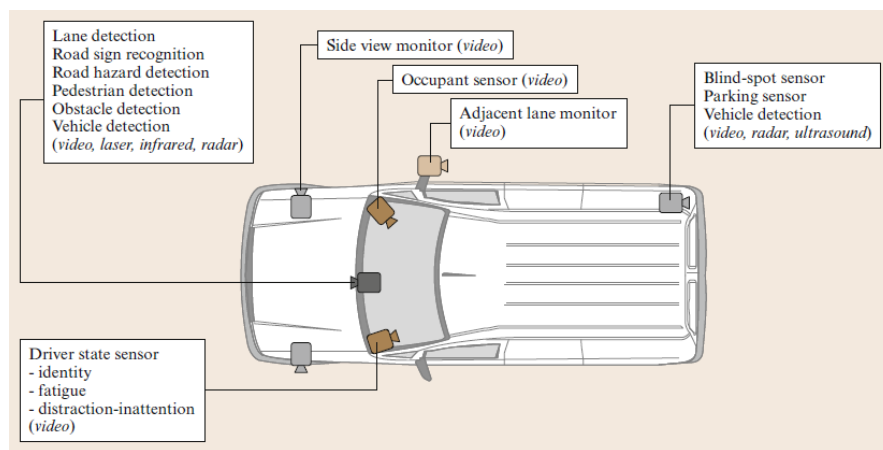
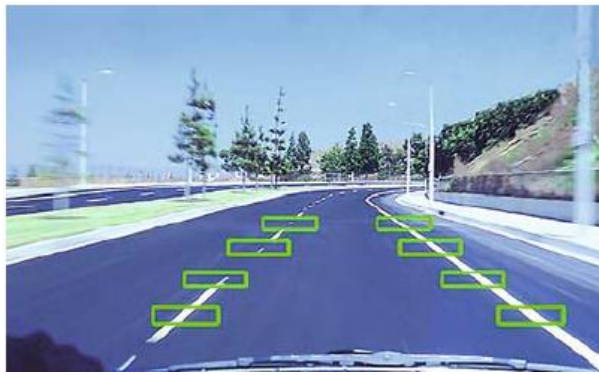
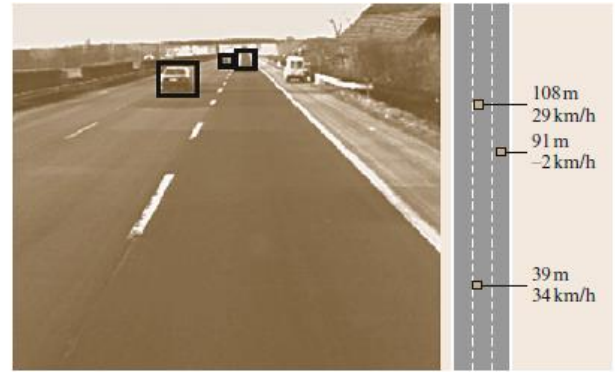


Figura 3.2. Disposición de algunos sensores de entorno en los vehículos inteligentes. (Springer, 2008).

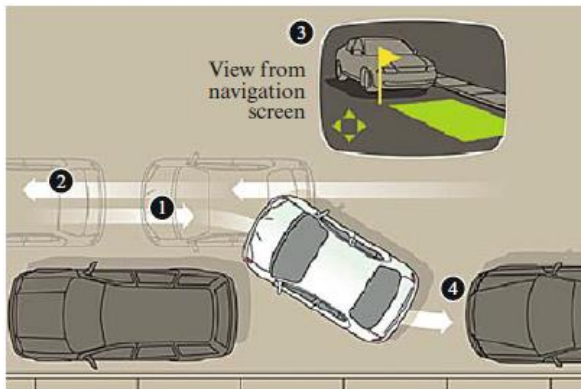


a)

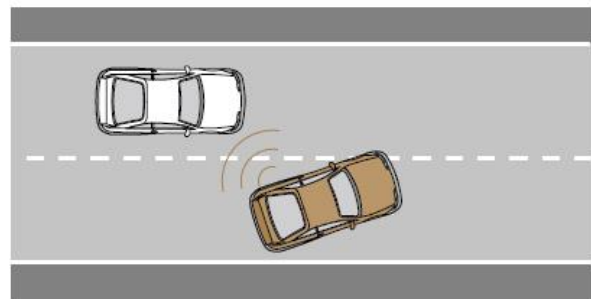


b)

Figura 3.3. Algunos ejemplos de procesamiento de imagen de los vehículos inteligentes. (a) Detección de vía. (b) Detección de vehículos. (Springer, 2008).



a)



b)

Figura 3.4. Algunas funcionalidades de los vehículos inteligentes. (a) Parqueo asistido. (b) Cambio de vía. (Springer, 2008).

- **Robots domésticos:** desde la misma aparición de la robótica, uno de los objetivos con lo que se ha soñado es tener en cada casa y asequible para todos los ciudadanos un robot doméstico, encargado de realizar todas las tareas cotidianas del hogar. Sin embargo los avances en esta rama de la robótica no ha avanzado tan rápido como se hubiera querido Actualmente existen muchos robots domésticos que realizan labores como: limpieza de pisos, ventanas, piscinas, entre otros; aplicaciones inteligentes como planchar, refrigeradoras inteligentes, roperos digitales. Esta área de la robótica converge al concepto deseado de *Smart homes*, que se define como una residencia equipada con la tecnología computacional y la información que le permite anticipar y responder a las necesidades de

los ocupantes, para promover su confort, comodidad, seguridad y entretenimiento a través de la gestión de la tecnología en el hogar y las conexiones con el exterior (Springer, 2008).

- **Humanoides:** En muchas regiones del orbe, los robots humanoides son la representación mental del concepto de robots. El campo de los robots humanoides se enfoca en la creación de robots inspirados en las capacidades humanas, copiando la cinemática de los humanos, así como los sentidos y el comportamiento. El objetivo principal de estos desarrollos es poder entender la inteligencia humana, la interacción del ser humano con el mundo y otras personas, entretenimiento, cultura, entre otras. El reto de estos mecanismos radica en la complejidad de los sistemas biológicos que tratan de simular, ya que se debe abarcar la locomoción (caminar, sensibilidad de balance, evitar obstáculos, entre otros), manipulación (brazos, manos, sensores de manipulación, aprendizaje, entre otros), comunicación (morfologías de expresión, comportamiento, interpretación de expresiones humanas, entre otros). Algunos de los robots representativos son los que se muestran en la Figura 3.5 (Springer, 2008).



a)



b)



c)

Figura 3.5. Robots humanoides. (a) ASIMO. (b) NAO. (c) Actroid.

3.2. Robots de docencia e investigación.

El ROBOBO es un robot autónomo para tareas de docencia e investigación. Los robots de este campo tienen dos características principales: son fácilmente programables y su costo es reducido. Es clave que sean de bajo coste ya que es un campo donde, a diferencia de la industria, el retorno de la inversión no suele ser directo. Otro motivo para abaratar costos es que puede ser necesario el uso de un número amplio de robots a la vez, para la investigación en sistemas multirobot o para evitar que los integrantes de los grupos de alumnos a los que se asigna cada robot sean demasiado grandes. La facilidad al programar un robot de investigación o docencia es una característica fundamental, ya que en estos campos el robot se utiliza como plataforma hardware para el desarrollo de sistemas de control, de sensorización, entre otros. En este punto hay que aclarar que una mayor facilidad de programación para fines docentes, como pueden ser lenguajes visuales o más simples, no tiene necesariamente que traducirse en una mayor facilidad para los investigadores. Esto se debe a que la robótica con fines de investigación puede depender de algoritmos complejos, difíciles de implementar utilizando como base conceptos que pueden ser excesivamente simples. En cualquier caso, los robots para docencia e investigación no pueden ser sistemas cerrados a nivel software.

Estos dos requisitos de bajo costo y código abierto delimitan un mercado para este tipo de robots que se encuentra en pleno auge en la actualidad. No son robots que requieran de un hardware muy sofisticado ni de una gran robustez mecánica, pero sí deben ser versátiles, contener un gran número de sensores y actuadores, y como ya se ha destacado, un entorno de desarrollo abierto y compatible con los lenguajes de programación estándar. Tras haber revisado los robots comerciales más representativos en robótica docente y de investigación (LEGO Mindstorms EV3, mOway, Khepera IV, e-puck, Thymio II, Pioneer 2, AIBO o NAO) se pueden identificar 4 problemas en este campo:

1. Baja calidad de los sensores: estos robots suelen montar sensores de bajo coste, desfasados del nivel actual del mercado y que complican el desarrollo de sistemas de control.

2. Baja capacidad de procesado: igual que en el caso anterior, suelen montar procesadores de bajo coste y con baja capacidad de cómputo, que impide ejecutar algoritmos complejos a bordo en tiempo real.
3. Poca capacidad de actualización del hardware: este es el principal problema que presentan, ya que el hardware se queda desfasado con rapidez, lo que implica renovar el robot por completo, aumentando el coste de amortización de los mismos.
4. Baja capacidad de comunicación: los robots más baratos que los actuales tienen una capacidad de comunicación limitada.

La robótica está introduciéndose rápidamente en el segmento educativo, incorporándose como disciplina en los planes de estudio. Esta irrupción abarca no solo el entorno tecnológico universitario sino que aparece con fuerza en centros de enseñanza primaria y secundaria. Además el aumento de distintas competiciones a nivel mundial patrocinadas por marcas líderes fomenta la implantación de robótica en los entornos de los más jóvenes, lo cual provoca la aparición de una relevante comunidad educativa.

Los educadores precisan de productos adaptados pedagógicamente a sus necesidades, que sean fáciles de implantar y con un *back office* que les apoye en la implantación del sistema. El educador o profesor, no demanda un Robot, demanda un sistema educativo integral que les facilite, sin una excesiva carga de tiempo, la asimilación de la tecnología correspondiente y el método de enseñanza adecuado a sus necesidades.

Tomando como punto de partida el segmento educativo, se identifican dos segmentos diferenciados y posicionados a ambos extremos del mismo.

Dichos segmentos corresponden a:

- **La robótica de investigación**
 - Es un sector que necesita de productos complejos y de altas prestaciones, dirigidos a centros de investigación de alto nivel, especializados en el área de robótica.
 - Es un sector limitado en cuanto a tamaño, estimándose en unos pocos centenares los centros existentes en todo el mundo, catalogados bajo esta denominación.

- **La robótica dirigida a actividades de entretenimiento**

- Este sector, está caracterizado por una serie de productos dirigidos al consumidor final, que generalmente combinan el concepto de “juguete robotizado”.
- Se encuentran presentes empresas multinacionales con una gran capacidad de innovación, distribución y marketing.
- En este segmento los productos están entrando con mucha rapidez en el mundo de la enseñanza, posicionándose con fuerza como instrumentos educativos dentro de las disciplinas de la robótica.
- Los productos actualmente presentan grandes limitaciones en el terreno educativo por ofrecer pocas opciones a esta comunidad.
- Algunos ejemplos de las compañías presentes en este mercado meta son LEGO Y Fischertechnik.

En el sector educativo se han identificado cinco productos que presentan similitudes técnicas y de funcionamiento con el ROBOBO, es decir, se componen de un Smartphone y una Plataforma móvil, en general, han sido promovidos desde centros de investigación o empresas StartUp y presentan capacidades inferiores al deseado en el ROBOBO. Los productos son los siguientes:

- WHEELPHONE (RoadNarrows, 2015).
- TYCHE (AIBRAIN, 2015).
- MICROROVER (Network, 2015).
- ROMO (APPLE, 2015).
- SMARTBOT (overdriverobotics, 2015).

En las figura 3.6 se muestra los robot mencionados anteriormente, como se puede observar en mucho de los casos presentan similitud en apariencia física, sin embargo el único que tiene prestaciones parecidas al ROBOBO es el WHEELPHONE que fue ha sido promovido por Swiss Federal Institute of Technology (EPFL-Suiza), los otros productos tiene carencias en sensorización (es el caso de TYCHE, MICROROVER y SMARTBOT) o no poseen la capacidad de crear una comunidad fuerte basada en sistemas libres (es el caso de ROMO) y en disponibilidad de aplicaciones (es el caso de TYCHE, MICROROVER, ROMO y SMARTBOT). En las Tabla 3.1 y

Tabla 3.2 se realiza una comparación que ejemplifica mejor las características de las plataformas similares al ROBOBO.



Figura 3.6. Plataformas similares al ROBOBO en el sector educativo. (a)WHEELPHONE. (b) TYCHE. (c)MICROROVER. (d) ROMO. (e) SMARTBOT.

Tabla 3.1. Comparación del ROBOBO con otras plataformas educativas.

ROBOTS EDUCATIVOS							
ASPECTOS BÁSICOS DEL PRODUCTO		ROBOT- NOMBRE COMERCIAL					
		ROBOBO	WHEELPHONE	TYCHE	MICROROVER	ROMO	SMARTBOT
Sistema	Android	X	X	X	X	-	X
	iPhone	-	X	-	-	X	X
Plataforma	Sensorizada	Sensorizada	Inactiva (estéril)	Solo movimiento	Sensorizada	Solo movimiento	
Precios (€)	Plataforma	180	175	-	109	160	
	Cargador	37					
	Movil	157					
	Punto de venta	SI	-	-	-	-	
Empresa	Fabricante	GII CORP	GCTRONIC	AI BRAIN	QUALCOMM TECH.	ROMOTIVE	OVERDRIVES ROBOTICS
	País	ESP	SUIZA	USA	USA	USA	USA
	Origen	Universidad	Universidad	Kick starter	-	Kick starter	High School Team
Comentarios		Diseño pobre	Es una aplicación para movil	Es un movil + base imprimible	Marketing enfocado segmento bajo	En desarrollo	
		Entidad con portafolio de productos	Poca credibilidad	kit para construcción 3D	Niños, telepresencia	Poca credibilidad	
Potencial competidor		SI	-	-	-	-	

Tabla 3.2. Comparación del ROBOBO con otras plataformas educativas.

ROBOTS EDUCATIVOS							
ASPECTOS BÁSICOS DEL PRODUCTO		ROBOT- NOMBRE COMERCIAL					
		ROBOBO	WHEELPHONE	TYCHE	MICROROVER	ROMO	SMARTBOT
Creación de Comunidad							
Sistemas abiertos		X	X	X	X	-	X
Aplicaciones disponibles		X	X	-	-	-	-
Valoración		alta	alta	media	media	nula	media
Calidad Pedagógica							
Estructuración por proyectos		X	-	-	-	-	-
Guías didácticas		X	-	-	-	-	-
Formación presencial a educadores		X	-	-	-	-	-
Diversos niveles de dificultad		X	-	-	-	-	-
Colectivos necesidades especiales		X	-	-	-	-	-
Eventos y competiciones		X	-	-	-	-	-
Valoración		alta	nula	nula	nula	nula	nula

En el sector de investigación también se han utilizado diversos robots, dentro del área de aplicación y funcionalidad del ROBOBO existen pocos productos y los que se pueden encontrar tienen la desventaja de que ciclos de vida son cortos y rápidamente quedan obsoletos. Se han identificado cinco productos similares al ROBOBO, promovidos por solamente dos entidades, la más destacada

es el Swiss Federal Institute of Technology (EPFL) presenta tres de los productos; el otro lo promueve una empresa canadiense el cual no se identifican como producto posicionados en el mercado, sino como producto en fase de desarrollo. Los productos similares son los siguientes:

- ELISA-3 (GCtronic, 2015).
- E-PUCK (e-puck, 2015).
- CCSR (Robots, 2015).
- THYMIO (Org, 2015).

En la Tabla 3.3, se muestra la comparación de los productos similares al ROBOBO, donde se hace énfasis al rango de precios que es elevado, a excepción del THYMIO, pero ofrece bajas prestaciones.

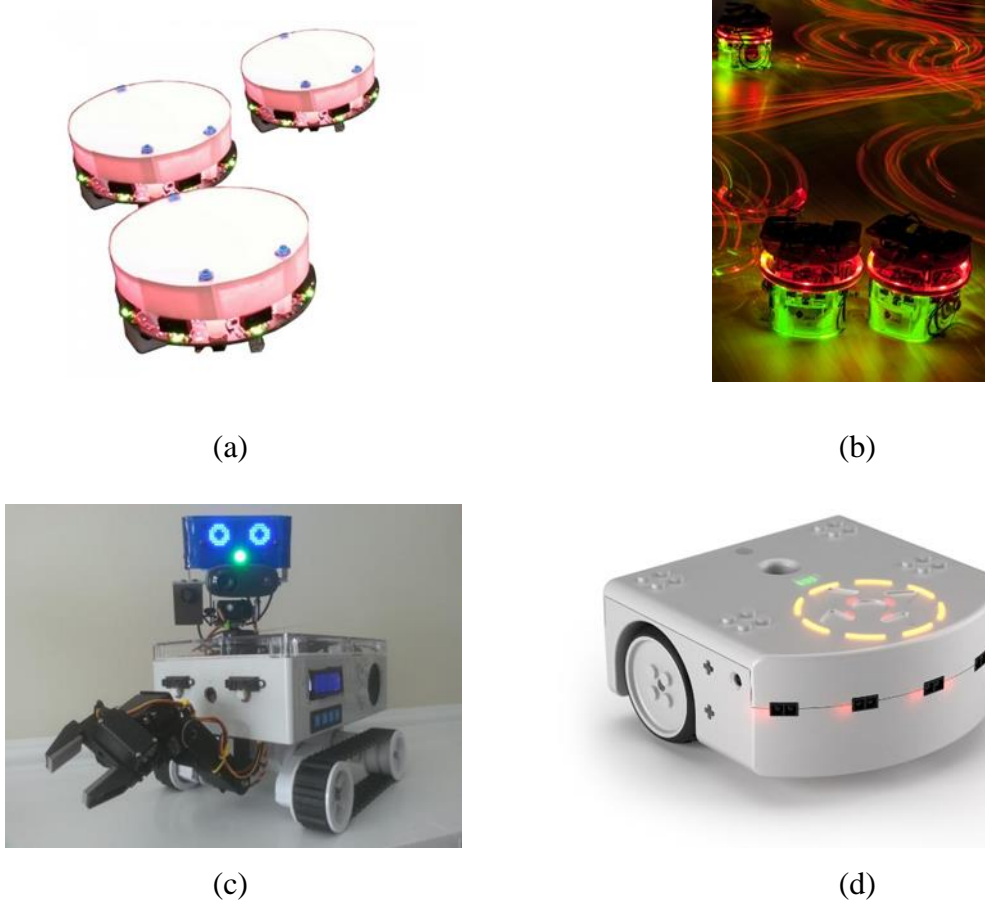


Figura 3.7. Plataformas similares al ROBOBO en el sector de investigación. (a) ELISA-3. (b) E-PUCK. (c) CCSR. (d) THYMIO.

Tabla 3.3. Comparación de plataformas de investigación similares al ROBOBO.

ROBOTS PARA INVESTIGACIÓN					
ASPECTOS BÁSICOS DEL PRODUCTO		ROBOT- NOMBRE COMERCIAL			
		ELISA-3	E-PUCK	CCSR	THYMIO
Precios (€)	Producto	300	738	-	115
	Cargador		92		
	Punto de venta	SI	SI		SI
Empresa	Fabricante	GCTRONIC	GCTRONIC	ROBOTSHOP	-
	Pais	SUIZA	SUIZA	CANADA	SUIZA
	Origen	Centro Investigación	Centro Investigación	Centro Investigación	Centro Investigación

Capítulo 4: Antecedentes del ROBOBO.

El primer prototipo del ROBOBO, fue desarrollado por el Ing. Martín Naya Varela, algunas de las características del primer prototipo se abarcan en este apartado. En la Figura 4.1 se muestra el primer prototipo del ROBOBO.

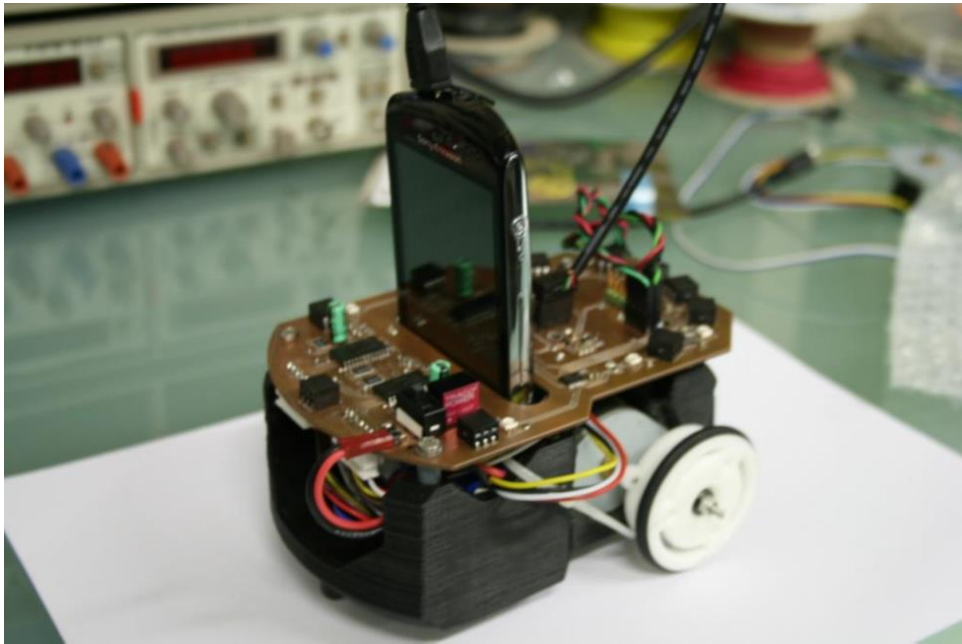


Figura 4.1. Primer prototipo realizado, ROBOBO 1.0. (Naya, 2013).

4.1. Sensores utilizados.

Los sensores con los que contaba el ROBOBO 1.0 eran sensores infrarrojos de proximidad. Se utilizó los sensores VCNL4000 de la marca VISHAY, se utiliza como sensor de choque, los cuales permiten, además de evitar que el dispositivo impacte contra obstáculos, obtener información del medio. Las características principales del sensor infrarrojo VCNL4000 son: sensor de luz ambiental, integrado con un procesador de señales con ADC de 16 bits, alcance de 20 cm y permite la conexión con el microcontrolador por medio del protocolo de comunicación I2C, lo que facilita conectar gran cantidad de sensores en cascada y ser controlados solamente utilizando 2 pines del microcontrolador (Naya, 2013). En la Figura 4.2 se muestra la apariencia física del sensor infrarrojo.



Figura 4.2. Sensor infrarrojo VCNL4000. (Naya, 2013).

4.2. Sistema Motriz.

Para el sistema motriz del ROBOBO 1 se había escogido motores a paso, debido a la facilidad para controlarlos usando una secuencia de pulsos. Debido a que se buscaba que el sistema motriz tuviera movimientos precisos, suaves y sin vibraciones. (Naya, 2013).

4.3. Sistema de comunicación.

Los sistemas de comunicación utilizados en el ROBOBO 1.0 fueron I2C para los sensores infrarrojos, SPI para los LED y USB para la comunicación con el Smartphone.

4.4. Sistema LED.

Los LED utilizados en el ROBOBO 1.0 fueron los ASMT-QTB0-0AA02, los cuales son LED RGB con ánodo común, la función de estos era servir como indicadores de los estados de los sensores infrarrojos, así como indicar el funcionamiento y los estados del ROBOBO por ejemplo, batería baja, alerta, entre otros. Para controlar la mayor cantidad de LED posibles utilizando las mínimas entradas del microcontrolador se utilizó un driver para LED el TLC5971, este dispositivo permite controlar la intensidad de corriente que circula por 12 canales (para 4 LED RGB) utilizando un PWM por cada salida además de permitir ajuste el brillo global. EL driver TLC5971 fue controlado por el microcontrolador utilizando el protocolo de comunicación SPI.

4.5. Sistema de alimentación.

Para la alimentación del ROBOBO 1.0 se utilizó una batería de Lipoly de 1000 mAh y 14.8 V, con la cual abastece la electrónica y la parte de potencia, para lograr las tensiones deseadas se utilizaron 2 reguladores, el convertidor DC-DC TSR1-2450 que reduce la tensión de 14.8 V a 5 V y seguido a este el regulador lineal MCP1700 que regula la tensión de 5 V a 3.3 V. (Naya, 2013).

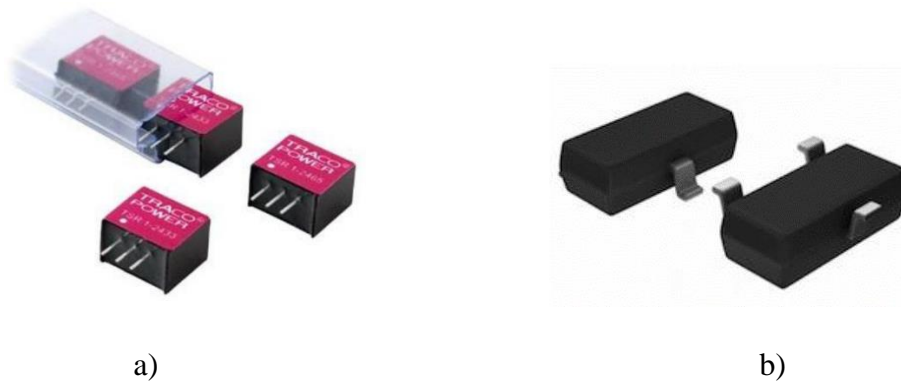


Figura 4.3. Reguladores utilizados en el ROBOBO 1.0. (a) DC-DC TSR1-2450 (b) MCP1700.

Capítulo 5: Enfoque metodológico.

El método de trabajo realizado se basó en el establecimiento de metas claras y concisas a corto plazo por parte de los directores que supervisaban la consecución de las mismas. Esta organización se realizó con un planteamiento temporal conservador que permitió cumplir con la mayoría de los objetivos finales del proyecto.

Las etapas se ordenaron secuencialmente para facilitar la comprensión del lector, sin embargo en la realidad se tuvo que ir alternando entre etapas futuras y etapas pasadas, con esta interacción dinámica nos permitió solucionar muchos de los problemas que surgieron.

5.1. Investigación y recopilación de datos.

Dentro de esta etapa se realizó la lectura de la propuesta desarrolladas por uno de los investigadores del GII, formaba parte de la memoria del primer prototipo del ROBOBO 1.0 realizado por el Ing. Martín Naya Varela como parte de su Proyecto Final de Carrera el día 8 de febrero de 2013.

En este documento se explica los antecedentes, requisitos del proyecto tanto a alto como a bajo nivel, los componentes utilizados junto con algunas dificultades encontradas y las soluciones, diseño de la placa PCB, resultados y finalmente los anexos junto con algunos cálculos realizado.

El objetivo principal de la lectura de este documento es conocer lo que se había hecho y cómo se había hecho, además de dar una perspectiva general de proyecto (aunque se han cambiado algunas cosas desde esa fecha hasta la actualidad), ya que el objetivo principal del proyecto que se está desarrollando con el ROBOBO 2.0 es que se optimice lo que ya se hizo en el ROBOBO, así como cambiar y adaptar los componentes del primer prototipo para que el costo de producción sea el menor posible sin afectar esto el rendimiento ni las prestaciones deseadas.

5.2. Definición de los problemas y mejoras a implementar.

En esta etapa se definieron las mejoras que se debían implementar así como los problemas que tenía la antigua plataforma. La mayoría de esta información fue facilitada por los organizadores del proyecto, sin embargo con el desarrollo del ROBOBO 2.0 y al realizar las pruebas respectivas se fueron encontrando nuevos problemas.

Los problemas y mejoras se definieron con base en la funcionalidad deseada para el ROBOBO y con el fin de llegar a satisfacer al usuario final, por lo que las decisiones se tomaron a partir de información de mercado y desde un punto de vista empresarial. A partir de lo anterior se fueron definiendo prioridades en los problemas a solucionar.

5.3. Diseño e implementación de las soluciones.

Las soluciones se definían siguiendo el proceso de justificación de la misma, así como un consenso entre los miembros del grupo de trabajo que estuvieran encargados del área en particular (mecánica, electrónica o de diseño). Dentro de la justificación se incluía la debida información técnica de la nueva solución, explicación debidamente justificada del porqué del cambio y además se debían tomar en cuenta las implicaciones desde el punto de vista económico, ya que uno de los objetivos principales del ROBOBO es que sea una solución de bajo coste y con grandes prestaciones.

Con el desarrollo de la plataforma, al componerse de varios módulos, se iba alternando en cada uno de ellos para encontrar los problemas y solucionarlos, una vez que se solucionaba o surgía una nueva mejora, se definía la solución y si esta implicaba nuevos componentes o un rediseño a cargo de otro miembro del grupo, mientras se realizaba el proceso administrativo correspondiente, se iba trabajando en otro módulo.

Una vez definida la solución se proseguía a implementar la solución, en donde se realizaban los esquemáticos eléctricos o modelos en 3D, se maquinaba las placas PCB, se soldaban los componentes electrónicos, se realizaba la programación correspondiente y finalmente se probaba la solución.

5.4. Verificación del comportamiento.

Al realizar las pruebas de las soluciones se comprobaba si esta funcionaba como se esperaba, además de verificar la respuesta ante perturbaciones o interferencia del medio. Cada uno de los módulos tiene en común el microcontrolador, por lo que en cada verificación estaba presente verificar si la configuración del microcontrolador era la correcta.

Primero se realizaba la verificación del componente diferencial de la solución, comprobando funcionalidad, comportamiento, comprobar las detecciones de error y llevar al componente al límite de funcionamiento. Con estas pruebas se podía descartar o confirmar la utilización de la solución en el ROBOBO final.

Posteriormente a las pruebas de verificación del módulo individual, se integraba con otro de los módulos para así comprobar si las funcionalidades encontradas anteriormente se mantenían o entraban en conflicto con el nuevo módulo. Además con esta verificación se reconfiguraba el microcontrolador para que realizara ambas tareas simultáneamente, con lo cual se trataba de evitar conflictos entre los protocolos de comunicación o con las interrupciones.

5.5. Rediseño y cambio de componentes.

Al realizar la verificación de la solución y con los datos correspondientes se realizaba el rediseño de la solución, esta podría ser desde el cambio en la configuración del microcontrolador hasta el cambio de uno o varios componentes de la solución o incluso el rediseño de la estructura mecánica del ROBOBO.

Para poder realizar este cambio de debía justificar técnicamente el motivo del rediseño o cambio del componente y proponer una nueva solución que fuera mejor que la anterior, en el caso de un rediseño solamente se realizaban los cambios y se volvía a realizar las pruebas para verificar si el problema encontrado persistía, sin embargo para el cambio de algún componente, el procedimiento es más extenso, que conlleva desde la selección del nuevo componente, la búsqueda de proveedores que puedan satisfacer la demanda para la producción en masa, verificación del precio del componente para muchas unidades, si el costo para la producción con el nuevo componente no

elevaba significativamente el precio del ROBOBO, hasta entonces se podría considerar como un nuevo sustituto para el antiguo componente, se realizaban pedidos de pocas unidades para realizar nuevas pruebas y todo el proceso se volvía a repetir.

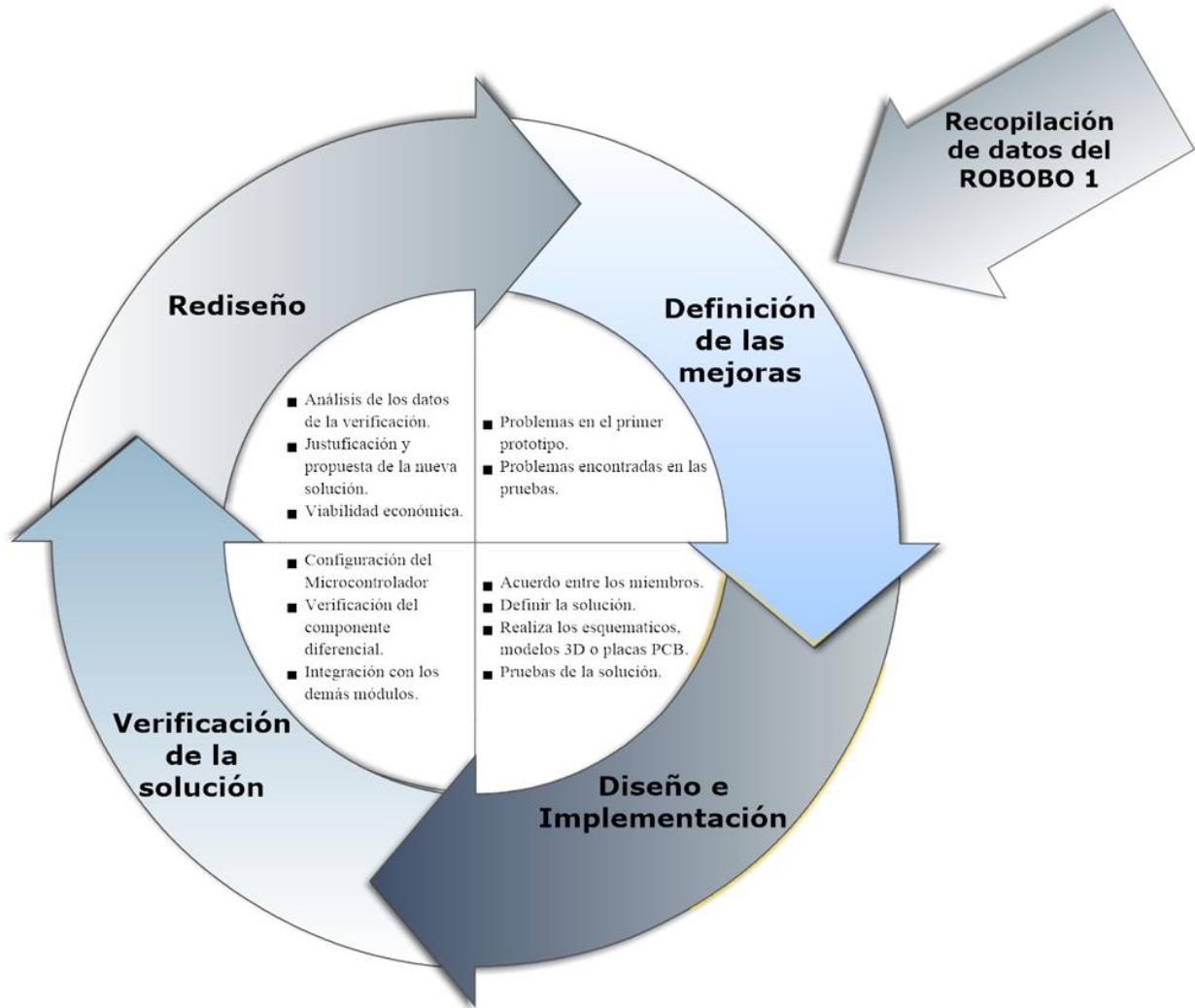


Figura 5.1. Enfoque metodológico utilizado para el desarrollo del proyecto.

Capítulo 6: Marco teórico.

6.1. Microcontrolador.

Un microcomputador está compuesto básicamente por el CPU (*Central Processing Unit*), memoria y entradas y salidas, cada parte intercambia información por medio de los buses (datos y/o direcciones). “Un microcontrolador es una microcomputador realizado en un circuito integrado” (Valdes & Pallás, 2007). Los microcontroladores se utilizan en aplicaciones puntuales, realizando pequeños números de tareas, donde no se requiere gran capacidad de procesamiento. Ejecuta solamente un programa previamente almacenado, guarda datos temporales e interactúa con el medio utilizando los periféricos, en la Figura 6.1 se muestra los componentes y la estructura básica de un microcontrolador. Los microcontroladores poseen las siguientes ventajas:

- Optimizar el espacio, debido a que se requiere que la solución sea de menor tamaño sin aumentar excesivamente los costos.
- A la medida, debido a la gran cantidad de gamas de microcontroladores se puede seleccionar uno que satisfaga con los requerimientos deseados para la aplicación.
- Seguridad de funcionamiento, ya que permiten saber cuándo no se logra cumplir con lo establecido en la programación o cuando lo que se ejecuta no corresponde a lo deseado. Una de esas herramientas es el *watchdog timer*.
- Bajo consumo, pueden funcionar utilizando una batería como fuente de alimentación, además permite la opción de entrar en estado de *sleep mode* mientras espera una respuesta externa y de esta forma consumir menos energía.
- Protección ante copias, cuentan con sistemas que evitan que se extraiga información almacenada en las memorias.

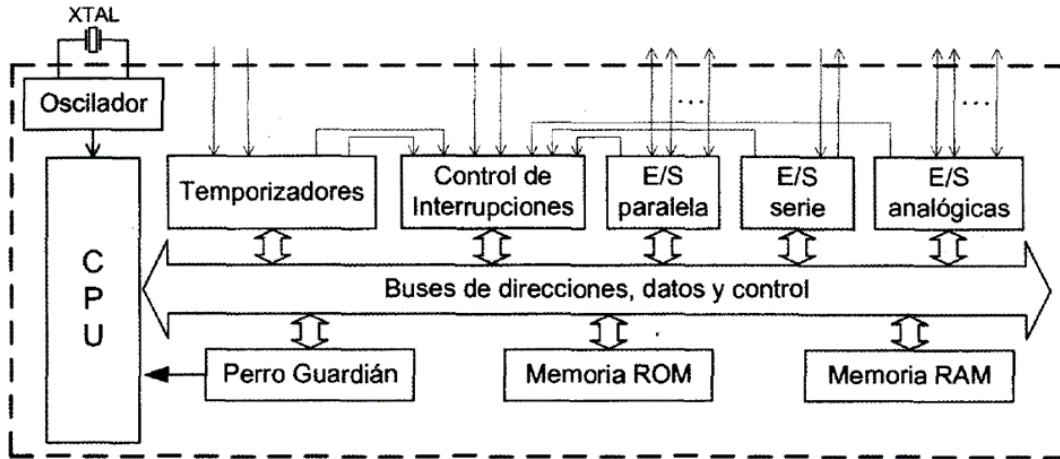


Figura 6.1. Estructura básica de un microcontrolador. (Valdes & Pallás, 2007).

6.1.1. Arquitecturas.

En lo que refiere a la organización de la memoria y la comunicación con el CPU existen dos arquitecturas: von Neumann, recibió este nombre debido al matemático John von Neumann, quien propuso que se utilizara solamente una memoria para datos y direcciones, lo cual implica que solamente se utiliza un bus para el tránsito de datos e instrucciones; y Harvard llamada así debido al lugar donde se diseñaron los ordenadores con esta estructura, realizado por Howard Aiken, esta arquitectura consiste en separar las memorias de datos y de instrucciones, por lo tanto cuenta con un bus para datos y otro bus para instrucciones, la memoria del programa es solo de lectura y el de datos se puede leer y escribir. (Valdes & Pallás, 2007). En la Figura 6.2 se muestra la diferencia entre la arquitectura von Neumann y Harvard.

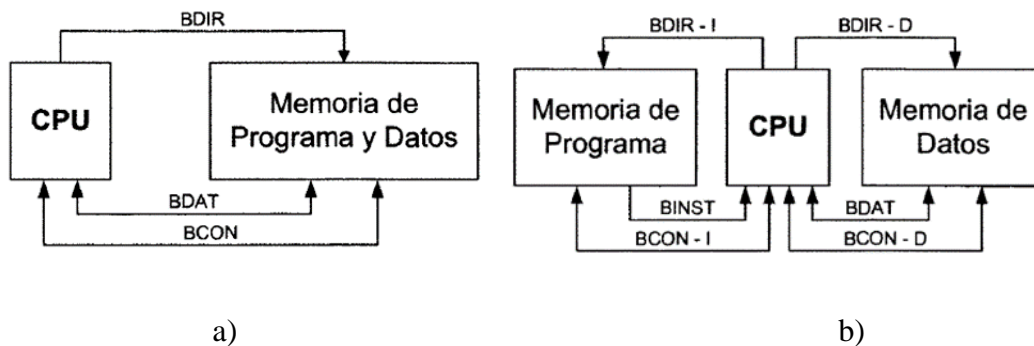


Figura 6.2. Arquitecturas de organización de memoria. (a) von Neumann. (b) Harvard.

Existen 2 arquitecturas para las instrucciones: CISC (*Complex Instruction Set Computer*) y RISC (*Reduced Instruction Set Computer*). la arquitectura CISC tiene las siguientes características: efectúa acciones muy complejas, las instrucciones pueden tener distintas longitudes, complejos modos de direccionamiento, lo que implica tener complejo hardware en la CPU por lo tanto gran espacio del circuito integrado; la arquitectura RISC tiene las siguientes características: el CPU dispone de instrucciones cortas y sencillas, las instrucciones tienen el mismo tamaño, pocos modos de direccionamiento, se ejecutan las instrucciones a alta velocidad y es más barato de diseñar. (Valdes & Pallás, 2007).

6.1.2. PIC32MX534F064H.

Para el desarrollo del ROBOBO 2.0 se empleó un microcontrolador, el cual posee la arquitectura Harvard y como arquitectura de instrucciones utiliza RISC. Como microcontrolador para las funciones de bajo nivel se utilizó el PIC32MX534F064H, aunque fue solamente para realizar las pruebas de los componentes periféricos y para las pruebas de funcionamiento, para el producto final se espera utilizar uno de la familia PIC32MZ.

Para facilitar la ejecución y la puesta en marcha de las pruebas se utilizó el MINI32, el cual consiste en un módulo que cuenta con todos los componentes integrados para hacer funcionar el PIC32MX (osciladores, reguladores, así como led indicadores y puerto USB), en la Figura 6.3 se muestra la apariencia física del MINI32 y en la Figura 6.4 se muestra el esquemático del circuito del MINI32. El uso de esta herramienta nos permite realizar las pruebas más rápidamente de los demás componentes como motores, led, drives y sistema de comunicación. (MiKroEleKtroniKa, 2015).

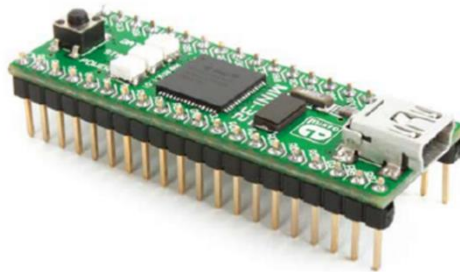


Figura 6.3. MINI32 utilizado en las pruebas. (MiKroEleKtroniKa, 2015).

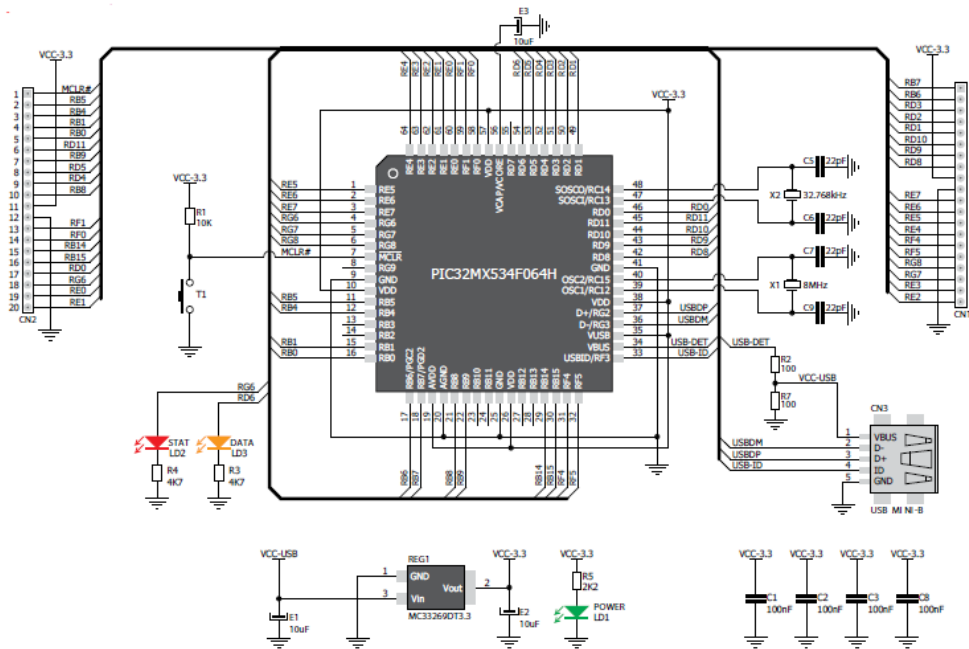


Figura 6.4. Esquemático del circuito impreso del MINI32. (MiKroEleKtroniKa, 2015).

6.1.3. Características del PIC32MX534F064H.

Las características del MINI32 son las siguientes:

- Alimentación puede ser por medio de la terminal USB, la cual es a 5V.
- Alimentación directa al PIC32MX534F064H, la cual puede ser de 2.3V a 3.3V.
- El consumo es de 300mA máximo cuando se alimenta a 3.3V.
- Osciladores de 8MHz y 32.768kHz.
- Memoria de programa de 64kB.
- Memoria RAM de 16kB.
- Memoria auxiliar Flash de 12kB.
- Temporizadores/contadores (Timers/Counters): cuenta con 5 de 16-bit, además al cambiar la configuración se puede utilizar 1 de 16-bit y 2 de 32-bit.
- Comparador de salida (output compare), cuenta con 5 que corresponden a los puertos: RD0, RD1, RD2, RD3 y RD4.

- Captura de entrada (input capture), cuenta con 4 input capture en los puertos: RD8, RD9, RD10 y RD11.
- 16 Entradas analógicas de 10-bit.
- Interface de comunicación, incluye USB 2.0, Ethernet MAC, modulo CAN, módulos UART, módulos SPI, módulos I²C y PMP.

6.2. PicKit3.

Para programar el Pic32MX se utiliza el software MPLAB X IDE v3.1, al cual se le instaló el compilador XC32 v1.40, La programación se realiza utilizando el PicKit3, esto debido a que con el PicKit3 se puede realizar depuraciones, así como verificación de los datos que está ejecutando el microprocesador en tiempo real. La configuración de la conexión del PicKit3 se muestra en la Figura 6.5.

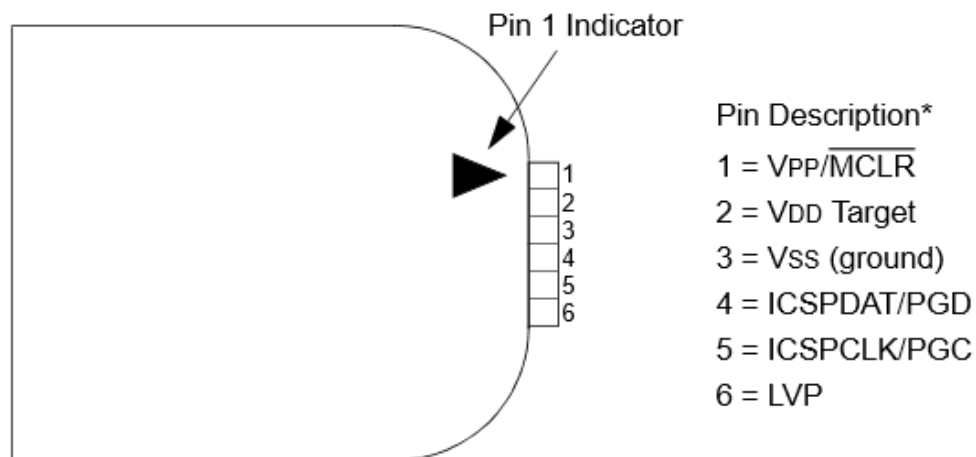


Figura 6.5. Diagrama de conexión del PicKit3.

6.3. Motor.

En este apartado se explicará los conceptos básicos del funcionamiento de un motor de corriente directa (CD). Un motor es un dispositivo que transforma la energía eléctrica en energía mecánica, los motores se pueden clasificar en motor de corriente directa (CD) o motor de corriente alterna (CA). Para el desarrollo del ROBOBO 2.0 se utilizó motores CD debido a que la fuente de alimentación con la que se va energizar proviene de una batería que entrega corriente directa.

6.3.1. Principio de funcionamiento del motor CD.

Si bien es cierto que el motor funciona con corriente directa, sin embargo internamente tiene un comportamiento alterno, el cual es dado por un rectificador mecánico llamado conmutador. Cuando se le aplica tensión en las terminales del motor, esto genera una corriente que pasa por las bobinas del rotor generando un campo magnético que interacciona con el campo magnético del estator (generalmente los campos magnético tendrían la misma polaridad), cuando el rotor gira buscando el polo opuesto, gracias a la intervención del conmutador el rotor cambia de polaridad, y de esta forma mantener el movimiento rotatorio continuo. En la Figura 6.6 se muestra la representación eléctrica de un motor CD, donde E_s es la alimentación de entrada, I es la corriente que fluye por las bobinas del motor, R es la resistencia producida por los cables del motor (de valor muy pequeño), E_o es la tensión en las bobinas, n es la velocidad angular y Φ es el flujo magnético efectivo por cada polo. (Wildi, 2007).

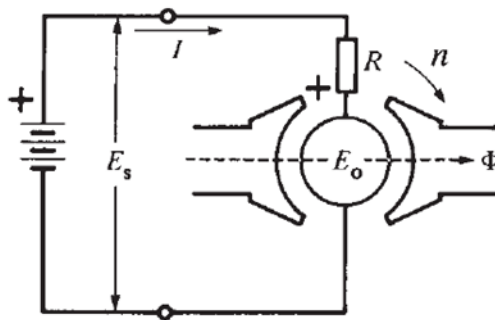


Figura 6.6. Representación de un motor CD. (Wildi, 2007).

Utilizando el circuito de la Figura 6.6, cuando se aplica la tensión E_s en el instante posterior se produce una gran corriente que fluye por la armadura debido al valor muy bajo de la resistencia R , inmediatamente en las bobinas del rotor se produce una fuerza la debido al campo magnético en el que se encuentra inmerso el rotor provocado por los imanes permanente del estator, generando el par de rotación. Una vez que el eje del motor comienza a girar se produce el efecto “generador” en el cual consiste en que comienza a fluir una corriente en dirección opuesta al que se le está aplicando por medio del E_s , esta tensión inducida es el E_o y es proporcional a la velocidad angular del motor y al Φ . A este voltaje inducido E_o se le conoce como fuerza contraelectromotiz (fcem) debido a que se opone al sentido del voltaje de la fuente. (Wildi, 2007).

6.3.2. Micro metal gearmotor

El motor que se utilizó para estas pruebas es el motor DC reductor con eje extendido (para colocar el encoder) como el que se muestra en la Figura 6.7, con escobillas como conmutador iguales a las que se muestran en la Figura 6.8. Se escogió este tipo de motor debido a que permite poder cambiar los requerimientos de diseño (velocidad y potencia) del motor sin necesidad de cambiar la estructura mecánica del ROBOBO, ya que este motor puede venir con reducciones de 5:1, 30:1, 50:1, 75:1, 100:1, 150:1, 210:1, 250:1, 298:1 y 1000:1. En general estos motores están pensado para ser usados a 6V, sin embargo pueden funcionar aceptablemente entre los rangos de 3V-9V, el giro comienza a un poco menos de 0.5 V sin embargo no posee suficiente torque para ser utilizados correctamente, y con tensiones mayores a 9V se reduce la vida útil del motor. (POLOLU, 2015).

Para estas pruebas se utilizó el motor 100:1, las características del motor a 6V son las siguientes:

- Tamaño: 10 x 12 x 26 mm (sin el eje).
- Peso: 9.5 g.
- Diámetro del eje: 3 mm.
- Velocidad en vacío: 320 rpm.
- Corriente en vacío: 120 mA.
- Corriente motor bloqueado: 1600 mA.
- Torque en motor bloqueado: 2.1602 kg-cm (30 oz-in).



Figura 6.7. Motor reductor utilizado en el ROBOBO para las pruebas. (POLOLU, 2015).



Figura 6.8. Escobillas utilizadas como conmutador en el gearmotor. (POLOLU, 2015).

6.4. Driver del motor

Para poder realizar el cambio de giro en el motor CD y regular la velocidad de funcionamiento se utilizó un driver, el cual consta de un puente H y un regulador de corriente.

6.4.1. Puente H.

El puente H recibe su nombre debido a la forma con la que se representa (Figura 6.9), utiliza cuatro MOSFETs, dependiendo de la dirección a la que se desea que gire el motor se activan lo MOSFET correspondiente, en el caso de puente H de la Figura 6.9, para la dirección *forward* se activarían los MOSFETs (A) y (D), mientras que para la dirección *reverse* se activarían los MOSFETs (B) y (C). Otra ventaja de utilizar los puentes H es que permiten que el motor se pueda detener mediante dos modos distintos, *fast decay*: en esta modalidad, al apagarse el motor, la corriente almacenada

en las bobinas se descarga desde tierra hacia la fuente de alimentación, protegiendo al motor de cambios bruscos de corriente pero genera un cambio de polaridad brusco en las terminales del motor, haciendo que este se detenga rápidamente; *slow decay*: en esta modalidad, ambas terminales del motor quedan conectadas a tierra, por lo que la energía almacenada en las bobinas del motor se descargarán según la constante de tiempo del circuito RL, lo que es más lento que la modalidad anterior. (Texas Instruments, 2015).

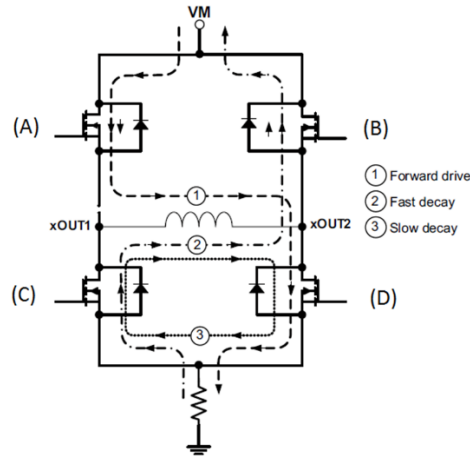


Figura 6.9. Esquemático de un puente H. (Texas Instruments, 2015).

6.4.2. Driver DRV8833.

El driver utilizado para los gearmotor fue el DRV8833 de la empresa Texas Instruments, dentro de las principales características de este driver son:

- Cuenta con dos puentes H para el control de la corriente de los motores.
- Tiene la capacidad de controlar dos motores CD.
- Capacidad de entregar hasta 1.5 A RMS por cada puente H.
- Rango de tensión de alimentación de 2.7 V a 10.8 V.
- Capacidad de limitar la corriente máxima individualmente de cada puente H.
- Función interna de apagado con una salida de fallos, se activa en las siguientes circunstancias: exceso de corriente, protección ante corto circuito, voltaje bajo y sobrecalentamiento.

Este driver permite variar la velocidad del motor utilizando PWM (*Pulse Width Modulation*) en una de las dos entradas de control, en la Tabla 6.1 se muestra las funcionalidades dependiendo de la configuración en los pines de control. En la Figura 6.10 se muestra las conexiones simplificadas del driver. Una de las funcionalidades del driver es el control de la corriente de salida, lo cual es utilizado para limitar la corriente de arranque y la corriente durante el bloqueo del motor, cuando la intensidad de corriente de salida supera el umbral definido, inmediatamente el puente H se deshabilita hasta el inicio del próximo ciclo del PWM, la corriente de corte es definida por un comparador el cual compara la tensión entre de la resistencia R_{ISENSE} y una tensión de referencia definida en 200 mV, la corriente de corte se calcula según la Ecuación 6.1.

$$I_{CHOP} = \frac{200 \text{ mV}}{R_{ISENSE}} \tag{6.1}$$

Donde:

I_{CHOP} = Corriente de corte.

200 mV = Tensión de referencia.

R_{ISENSE} = Resistencia para medir la corriente de salida.

Tabla 6.1. Funcionalidades de salida del driver DRV8833.

xIN1	xIN2	FUNCTION
PWM	0	Forward PWM, fast decay
1	PWM	Forward PWM, slow decay
0	PWM	Reverse PWM, fast decay
PWM	1	Reverse PWM, slow decay

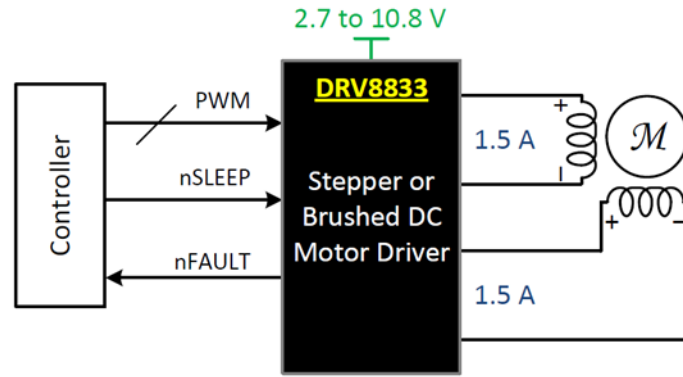


Figura 6.10. Esquemático simplificado del driver DRV8833. (Texas Instruments, 2015).

Las protecciones del circuito integrado son: ante exceso de corriente, si la corriente en las bobinas de los motores o en la alimentación del driver sobrepasa el valor de 3.3 A, se detectará un evento de exceso de corriente, las acciones en este caso es la inhabilitación de los puentes H, cabe mencionar que esta funcionalidad es independiente del control de la corriente de salida; ante sobrecalentamiento, este fallo sucede cuando la temperatura del integrado alcanza los 160 °C (llegando al límite de la temperatura de funcionamiento), al suceder este evento se deshabilitan los puente H y solamente volverá a funcionar el driver una vez la temperatura este dentro del rango de funcionamiento; y la última protección es ante baja tensión, esto sucede cuando la tensión de alimentación del driver DRV8833 es inferior a 2.7 V, al suceder este evento se deshabilitan los puentes H y el circuito interno del driver, el funcionamiento vuelve a la normalidad una vez la tensión de alimentación supere los 2.7 V (Texas Instruments, 2015). En la Tabla 6.2 se resume las protecciones del circuito.

Tabla 6.2. Protecciones del circuito en el driver DRV8833. (Texas Instruments, 2015).

FAULT	CONDITION	ERROR REPORT	H-BRIDGE	INTERNAL CIRCUITS	RECOVERY
VM undervoltage (UVLO)	$V_M < 2.5 \text{ V}$	None	Disabled	Disabled	$V_M > 2.7 \text{ V}$
Overcurrent (OCP)	$I_{OUT} > I_{OCP}$	FAULTn	Disabled	Operating	OCP
Thermal Shutdown (TSD)	$T_J > T_{TSD}$	FAULTn	Disabled	Operating	$T_J < T_{TSD} - T_{HYS}$

6.5. Encoder.

6.5.1. Sensor efecto Hall.

Los sensores efecto Hall son utilizados para detectar campos magnéticos, pueden ser analógicos o digitales. En el ROBOBO 2.0 se utilizó el sensor digital TLE4946-2K, el cual posee umbral e histéresis muy bajos, lo que le permite tener buena precisión en determinar la posición del rotor del motor. Las características del sensor efecto Hall TLE4946-2K son:

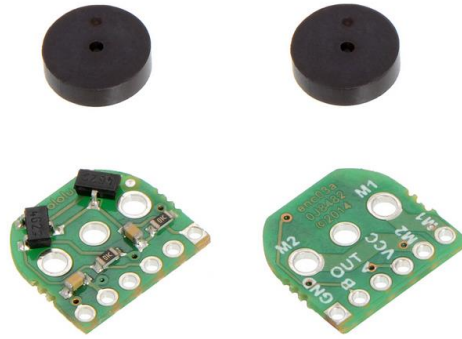
- Tensión de alimentación entre 2.7 V a 24 V.
- Alta sensibilidad y alta estabilidad.
- Protección ante tensiones reversas de -18 V.
- Temperatura de funcionamiento hasta de 150 °C.
- Salida digital.

6.5.2. Encoder Magnético.

Los encoders son sensores utilizados para determinar el desplazamiento angular de sistemas rotatorios, el en ROBOBO el encoder utilizado consta de dos sensores efecto Hall (TLE4946-2K) y un disco de 6 polos magnéticos, por lo que le permite tener 12 cuentas por revolución, generalmente se utilizan dos sensores efecto Hall, ya que al ponerlos en cuadratura se puede determinar la dirección de giro del motor. En la Figura 6.11 se muestra el encoder que se utilizó para determinar tanto la posición angular como la velocidad del motor.

Las características del encoder son:

- Tamaño: 10.6mm x 11.6mm.
- Tensión de operación mínimo: 2.7V.
- Tensión máxima de operación: 18V



Magnetic Encoder Pair Kit for Micro Metal Gearmotors, 12 CPR, 2.7-18V (old version).

Figura 6.11. Encoder por cuadratura utilizado en el prototipo del ROBOBO.

En la Figura 6.12 aparece el esquemático del encoder, donde U1 y U2 representan los sensores efecto hall.

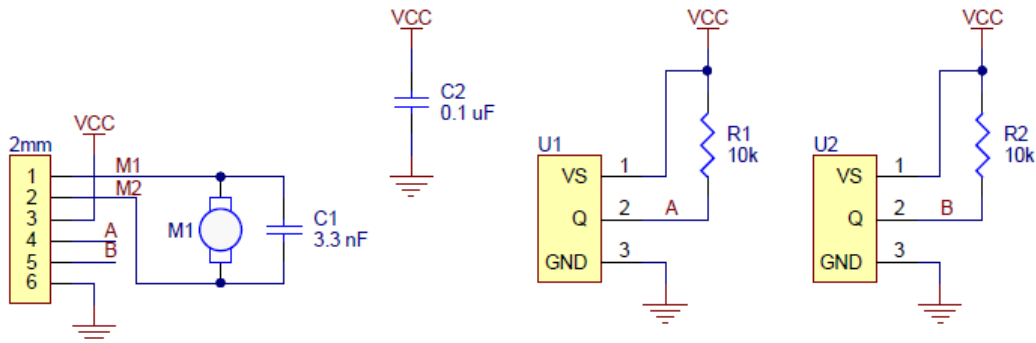


Figura 6.12. Esquemático del encoder.

6.6. Controlador automático.

“Un controlador automático compara el valor real de la salida de una planta con la entrada de referencia (el valor deseado), determina la desviación y produce una señal de control que reduce la desviación a cero o a un valor pequeño. La manera en la cual el controlador automático produce la señal de control se denomina acción de control” (Ogata, 2010). En la Figura 6.13 se muestra el diagrama básico de un sistema que utiliza un controlador automático.

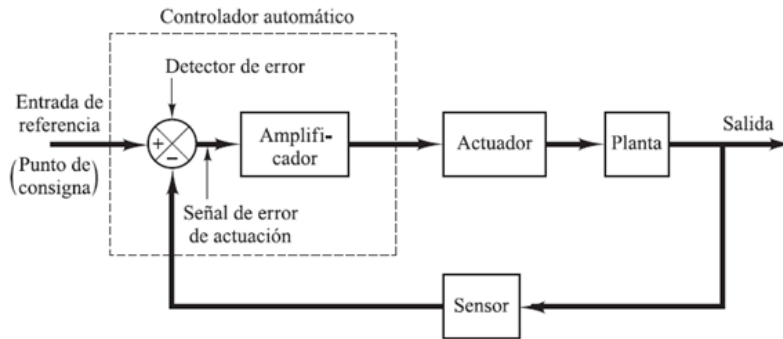


Figura 6.13. Diagrama de bloques de un sistema de control. (Ogata, 2010).

El control se puede realizar ya sea a lazo cerrado (retroalimentado) o a lazo abierto. Un control a lazo cerrado consiste en un sistema que mantiene una relación entre la salida y la entrada de referencia, realizando una comparación y usando esta como medio de control; Un control a lazo abierto son sistemas en la que la salida no tiene efecto sobre la acción de control, en otras palabras, no existe una comparación entre las salidas y las entradas. La ventaja de utilizar el control a lazo cerrado es que gracias a la retroalimentación permite que la salida del sistema no se vea afectada a pesar de las perturbaciones externas e internas (Ogata, 2010), el diagrama de la Figura 6.13 es un ejemplo de un sistema a lazo cerrado, y es con el que se va realizar el control de velocidad del ROBOBO 2.0.

La función de transferencia es lo que se utiliza para caracterizar la relación de entrada y salida de un sistema, esta descrito por medio de ecuaciones diferenciales lineales e invariantes en el tiempo, es el cociente de la transformada de Laplace de la salida y de la entrada, suponiendo condiciones iniciales cero, en la Ecuación 6.2 se muestra el cociente de la función de transferencia.

$$G(s) = \frac{\mathcal{L} [Salida]}{\mathcal{L}[Entrada]} \quad 6.2$$

6.6.1. Controlador PI.

Los controladores se pueden clasificar según sus acciones de control en:

- Controladores *on-off*.
- Controladores proporcionales (P).
- Controladores integrales (I).
- Controladores Proporcionales integrales (PI).
- Controladores Proporcionales derivativos (PD).
- Controladores Proporcionales integrales derivativos (PID).

En este apartado solamente se va a explicar el funcionamiento de los controladores integrales y proporcionales integrales. En la acción de control integral, el valor de salida del controlador cambia a una razón proporcional a la señal de error, la Ecuación 6.3, se muestra la función de transferencia del controlador integral. En la acción de control proporcional integral, mezcla las características del controlador proporcional y del controlador integral, en el controlador proporcional existe una relación directa entre la salida del controlador y la señal de error, esta relación es definida por una constante K_p , al incluirle el controlador integral, el comportamiento además de mantener la proporcionalidad entre salida y entrada, la señal cambia a una razón proporcional a la señal de error, en la Ecuación 6.4 se muestra la expresión matemática de la salida del controlador en función del tiempo, mientras que en la Ecuación 6.5 aparece la función de transferencia del controlador proporcional integral. (Ogata, 2010).

$$I = \frac{K_I}{s} \quad 6.3$$

$$u(t) = K_p e(t) + \frac{K_p}{T_I} \int_0^t e(t) dt \quad 6.4$$

$$PI(s) = K_p \frac{\left(s + \frac{K_I}{K_p}\right)}{s} \quad 6.5$$

6.7. Protocolos de comunicación.

Para poder comunicar el microcontrolador con otros dispositivos, se debe hacer uso de los protocolos de comunicación, el microcontrolador PIC32MX534F064H tiene la capacidad de utilizar los siguientes protocolos de comunicación: UART, SPI, I2C, USB 2.0, Ethernet MAC, módulo CAN y PMP (*Parallel Master Port*). En la implementación del ROBOBO 2.0 se utilizará solamente los protocolos UART, SPI y I2C, sin embargo para el desarrollo de la parte correspondiente a esta documentación solamente se utilizó los dos primeros.

6.7.1. UART.

El microcontrolador PIC32MX534F064H tiene un puerto serie para comunicación llamado UART (*Universal Asynchronous Receiver Transmitter*), bidireccional simultanea (*full dúplex*), permite la comunicación con dispositivos periféricos y computadora personal por medio de los protocolos como RS-232, RS-485, LIN 1.2 y IrDA®.

Utiliza dos terminales UxTX (*transmitter*) y UxRX (*receiver*), además de estos el PIC32MX534F064H cuentan con dos pines adicionales UxCTS (*clear to send*) y UxRTS (*request to send*). El UxCTS funciona como una entrada que permite controlar la transmisión y es controlado por otro dispositivo (usualmente una PC); El UxRTS indica cuando el dispositivo está listo para recibir datos. (Microchip, 2015).

La ventaja de protocolo UART es que al funcionar en modo asincrónico permite que ambos dispositivos puedan enviar y recibir datos a la vez. La señal que se envía consta de 8 bits precedidos del bits de inicio (valor de 0) y sucedidos por el bits de parada (valor de 1), adicionalmente se puede configurar el puerto para que envíe o reciba 9 bits en lugar de 8. En la Figura 6.14 se muestra la estructura de los datos en el protocolo de comunicación UART. (Valdes & Pallás, 2007).

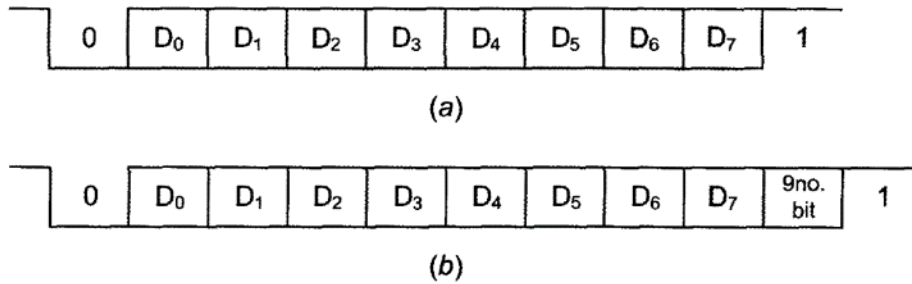


Figura 6.14. Estructura de datos de la comunicación UART. (a) 8 bits. (b) 9 bits.

6.7.2. SPI.

El protocolo de comunicación SPI (*Serial Peripheral Interface*) es un módulo serial síncrono muy útil para la comunicación entre periféricos externos y otros microcontroladores. Puede realizar transmisión y recepción de datos de 8 bits, además puede funcionar como servidor (*master*) o cliente (*slave*). El protocolo SPI utiliza cuatro terminales, las cuales son; SDO, para la salida de datos; SDI, para la entrada de datos; SCK, es la señal síncrona que funciona como reloj, es generada por el servidor; y la terminal SSx (*Slave Select*), la cual sirve para seleccionar al cliente con el que quiere establecer la comunicación. En la Figura 6.15 se muestra la conexión entre dos PIC utilizando el protocolo de comunicación SPI.

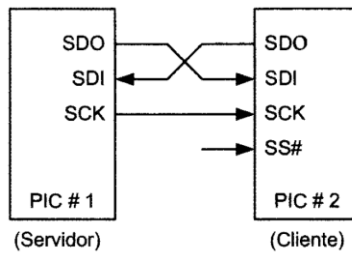


Figura 6.15. Conexión utilizando el protocolo de comunicación SPI.

6.8. Bluetooth.

En 1994 se la tecnología Bluetooth se concibió como una alternativa de comunicación inalámbrica que intercambiara datos usando transmisión de radio. El nombre Bluetooth viene del nombre del rey danés del siglo decimo llamado Harald Blatand que en ingles se traduce a Harold Bluetooth, este rey ayudo a unir a los bandos enfrentados en lugares que ahora son Noruega, Suecia y Dinamarca, al igual que el rey, la tecnología Bluetooth permite la conectividad y la colaboración entre productos e industrias distintas. La comunicación se da a corta distancia, cuando dos dispositivos desean conectarse primero deben emparejarse, después se crea una red llamada *Piconet* y puede ser de dos a ocho dispositivos conectados, cuando de crea la red uno de los dispositivos debe asumir el papel de master y los demás funcionaran como esclavos, además una vez que los dispositivos ya están emparejados, la conexión se realiza automáticamente al entrar en el radio de acción. (Bluetooth, 2016).

Para el ROBOBO 2.0 se utilizará el módulo de Bluetooth HC-06 (OLIMEX, 2016). Algunas de las características de este módulo son:

- Sensibilidad puede alcanzar los -80dBm.
- Antena de 2.4GHz.
- Memoria externa Flash de 8Mbit.
- Tensión de alimentación entre 3.1 V a 4.2 V.
- Consumo mientras esta emparejando entre 30 mA y 40 mA.
- Consumo una vez establecida la conexión de 8 mA, sin importar si transmite datos o no.

El módulo bluetooth HC-06 permite transmitir datos serial a las siguientes velocidades: 1200, 2400, 4800, 9600 (valor por defecto), 19200, 38400, 57600, 115200, 230400, 460800, 921600 y 1382400 b/s. Sin embargo muchos dispositivos no admiten velocidades superiores a 115200 b/s

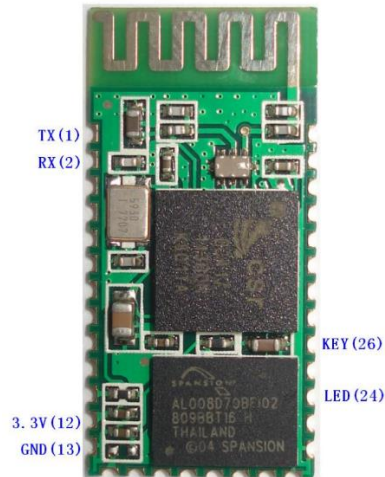


Figura 6.16. Módulo Bluetooth HC-06.

6.9. Driver LED.

Para poder controlar varios LED simultáneamente se utilizó el driver para LED TLC5941, de la empresa Texas Instruments. Tiene las siguientes características:

- Permite controlar 16 LED (cinco LED RGB).
- Dos modalidades, escala de grises por PWM o por control de corriente por cada canal.
- PWM de 12 bits (4096 valores).
- Control de corriente de 6 bits (64 valores).
- Tensión de alimentación para LED de hasta 17 V.
- Tensión de alimentación para el driver entre 3 V hasta 5.5 V.
- Interfaz de datos serial.
- Transferencia de datos de hasta 30 MHz.
- Salida CMOS.
- Información de error, LOD (*LED Open Detection*) y TEF (*Thermal Error Flag*).

La conexión serial del driver solamente necesita tres pines: SCLK el cual funciona como reloj, en cada flanco de subida almacena el bits de entrada en el registro interno correspondiente; SIN pin

6.9.1. Modos de funcionamiento.

Existen dos modos de funcionamiento, por escala de grises definido por un PWM (modo GS) y ajuste de la corriente de salida (modo DC). Cabe resaltar que al energizar el driver, este asigna valores aleatorios en los registros, por lo que se debe configurar los valores iniciales antes de activar las salidas, primero se asignan los valores de GS y posteriormente los de DC en la configuración inicial. (Texas Instruments, 2015).

En el modo DC (*Dot Correction*), permite un ajuste fino de la corriente que fluye por cada salida de forma independiente, con esto se puede ajustar el brillo de cada LED de los 16 canales, para definir el brillo se programa con una palabra de 6 bits, lo que permite tener 64 valores que van de 0% hasta 100% de brillo respecto a la corriente máxima definida por la resistencia R_{REF} , la corriente de salida se calcula utilizando la Ecuación 6.7. En la Figura 6.18 se muestra como debe ser la secuencia de datos enviados por el protocolo SPI. (Texas Instruments, 2015).

$$I_{OUT} = I_{max} \frac{DC}{63} \tag{6.7}$$

Donde:

I_{max} = corriente máxima definida por la resistencia R_{REF} .

DC = valor programable entre el rango de 0 a 63.



Figura 6.18. Estructura de los datos en el modo DC.

Modo GS (Grayscale PWM), al igual que el modo DC, permite ajustar el brillo de cada salida para LED, sin embargo en esta modalidad, la regulación se da a partir de un control por PWM, el cual

utiliza 12 bits lo que permite tener 4096 valores que van desde 0% a 100% de brillo. El porcentaje de brillo se pueda calcular utilizando la Ecuación 6.8, la estructura de datos que se envía por el protocolo de comunicación SPI es la que se muestra en la Figura 6.19. (Texas Instruments, 2015).

$$Brillo\% = \frac{GS}{4095} * 100 \tag{6.8}$$



Figura 6.19. Estructura de los datos en el modo GS.

Capítulo 7: Descripción detallada de la solución.

En este apartado se explicará en detalle todos los componentes de la solución seleccionada, cabe resaltar que durante el desarrollo de todo el proyecto se rediseñaron varios componentes, sin embargo para efectos de simplificar la comprensión de este apartado, solamente se hará la explicación de la solución final.

7.1 Configuración PWM.

Los PWM se pueden generar a partir de la configuración de los output compare (OC), sin embargo solo se puede usar el timer2 o el timer3 para definir la frecuencia del PWM lo cual es una gran limitante cuando se trabaja con varios PWM, para la primer prueba de los motores se utilizaron el OC5 y el OC1 que corresponden a los puertos RD4 y RD0 del microcontrolador, esto debido a la facilidad para cablear junto a las demás salidas que irán conectadas al driver.

Para poder configurar el PWM se debe tomar en cuenta principalmente los siguientes registros: PRx, este es el registro donde se guarda el periodo deseado; OCxRS, este registro determina el tiempo en alto de cada periodo del PWM; y finalmente el registro OCxCON que tiene la función de configurar el output compare.

La secuencia recomendada para la configuración del PWM es la siguiente:

- Desactivar el OC que se desea configurar.
- Seleccionar en los bit del OCxCON que corresponden a OCM y asignarles el valor de 110 ó 111 (binario).
- Asignar el periodo en el registro PRx (correspondiente al Timer que se va utilizar).
- Asignar el ciclo de trabajo en el registro OCxRS.
- Asignar el ciclo de trabajo inicial en el registro OCxR.
- Habilitar las interrupciones del Timer y del módulo OC.
- Asignar el valor de prescala del Timer.
- Activar el Timer y el modulo OC.

7.1.1. Consideraciones para el PWM.

Para implementar correctamente el PWM se debe tener en cuenta la frecuencia del oscilador en los periféricos, debido a que con ésta es que funcionan los temporizadores. Como se puede apreciar en la Figura 7.1, se puede seleccionar distintos valores de frecuencia para los periféricos, este valor depende de las distintas escalas que se seleccionen, si no se realiza ningún cambio el PIC32MX534F064H selecciona por defecto la división de 2 en el registro FRCDIV y la división de 8 en el registro de PBDIV, al tener el oscilador primario con el valor de 8MHz, la frecuencia de los periféricos es de 500kHz. Para las pruebas se utilizó este valor, sin embargo para la prueba con los motores se utilizó la frecuencia máxima del oscilador para los periféricos, para lo cual basta con configurar en las opciones del FNOSC y seleccionar el valor del PRI, además de esto se debe poner el registro PBDIV con un valor de 1, con estos cambios se obtiene en los periféricos la frecuencia de 40MHz.

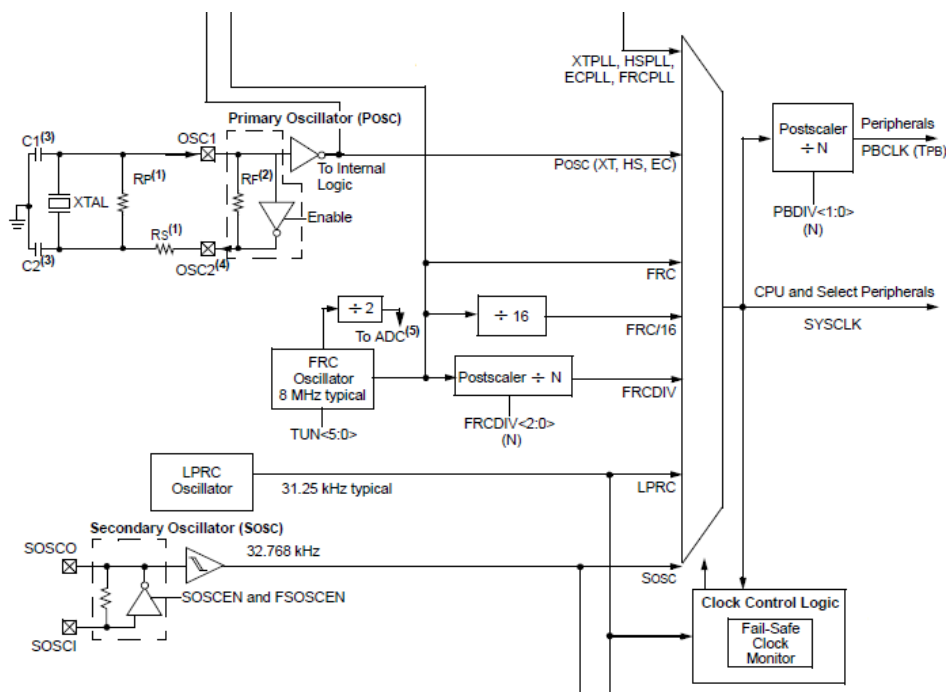


Figura 7.1. Diagrama de bloques de los osciladores.

Otra de las consideraciones es el cálculo de los valores de los registros, para lo cual como se explicó anteriormente, se necesita calcular los valores para dos registros, el PRx que corresponde a la Ecuación 7.1 y el OCxR que corresponde a la Ecuación 7.2.

$$PRx = \frac{1}{FrecuenciaPWM * Tpb * TimerPrescale} - 1 \quad 7.1$$

$$OCxR = \frac{Ciclo\ de\ Trabajo\ (\%)}{100} * PRx \quad 7.2$$

Donde Tpb es el periodo del oscilador en el bus de periféricos.

Cabe destacar que el “- 1” presente al final de Ecuación 7.1 viene dado por el comportamiento de la onda del PWM, en la Figura 7.2 se muestra como está distribuido los valores de los registros PRx y OCxRS.

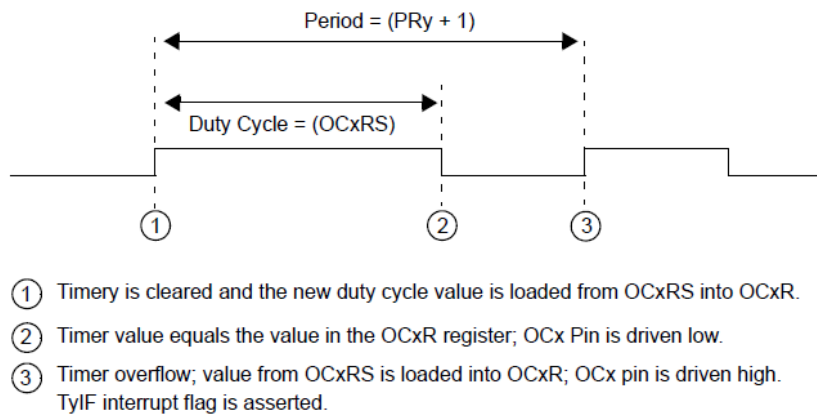


Figura 7.2. Proceso de la formación de la señal del PWM.

En el Apéndice A.2: aparecen algunas pruebas con los módulos OC, además de las comparaciones entre estos resultados.

7.2. Funcionamiento del driver para motor DRV8833

Para probar el funcionamiento del driver DRV8833, se utilizó el módulo que aparece en la Figura 7.3, el cual ya viene con un regulador de tensión que protege al integrado así como al circuito de control de los picos de corriente que generan los motores. Para la primera prueba no se utilizó el PWM del microcontrolador, sino que se alimentó con una tensión constante de 6.5 V. Los pines de la Figura 7.3 se configuraron de la siguiente forma:

- GND, AIN2, BIN2 = 0 V.
- VMM, nFAULT, AISEN, BISEN = no conectado (NC).
- VIN, AIN2, BIN2, nSLEEP = 6.5 V.
- AOUT1, AOUT2, BOUT1, BOUT2 = terminales hacia los motores.

Adicionalmente se le soldó los espacios correspondiente las resistencias que limitan la corriente de salida de cada motor, se definió una corriente máxima permitida de 1.3 A y utilizando la Ecuación 6.1 se obtuvo el valor de la resistencia $R_{ISENSE} \approx 0.15 \Omega$.

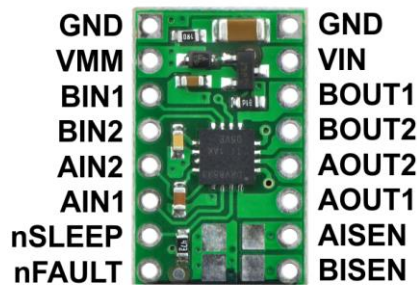


Figura 7.3. Pines del driver DRV8833.

En la primera prueba del driver para motor se buscaba determinar los límites de funcionamiento del driver y con esto verificar si podría ser útil para utilizarlo en el ROBOBO 2.0, las condiciones en esta prueba era la conexión de los pines con la configuración mencionada anteriormente, además se utilizó los motores 150:1 de reducción, se alimentó lo motores a 6.5 V y se bloquearon para que la corriente aumentara y así verificar si el driver apagaba el puente H correspondiente. En la Figura 7.4 se muestra la señal obtenida en la primera prueba. La señal en amarillo es la tensión en el motor

que fue bloqueado, la señal en cian es el otro motor funcionando en vacío y la señal de color morado es la entrada al pin BIN2 que corresponde al motor bloqueado.

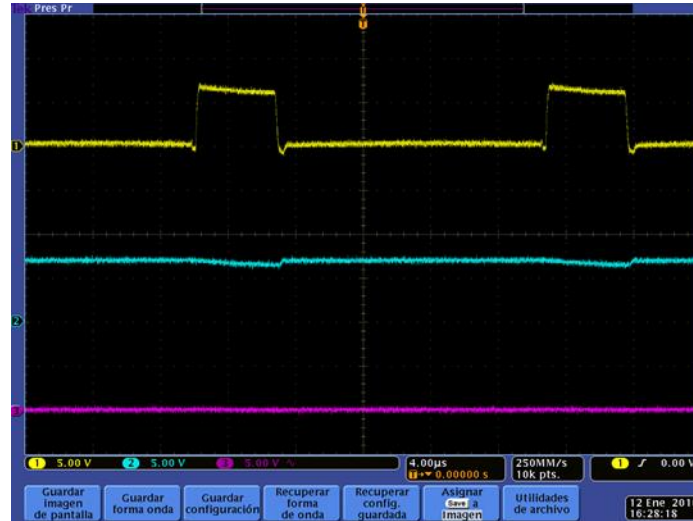


Figura 7.4. Señal obtenida al bloquear el motor.

Para el funcionamiento en el ROBOBO 2.0 no se puede utilizar la configuración de los pines xIN1 y xIN2, debido a que el motor tiene que ser capaz de cambiar de dirección, por lo que se utilizó la configuración que se muestra en la Figura 7.5, se realizó esta conexión debido a que según la configuración de la Tabla 6.1, el PWM debe alternarse de la entrada xIN1 a la entrada xIN2 y viceversa y con esto cambiar el sentido de giro del motor. Debido a que el microcontrolador solamente tiene 5 pines con la capacidad de generar PWM y se necesita controlar cuatro motores, solamente se puede utilizar un PWM por motor. Se discutieron varias opciones para este inconveniente, como por ejemplo, utilizar transistores para alternar el PWM, pero debido al coste se prefirió utilizar solamente resistencias y con las propiedades de los pines del microcontrolador (pasar de entrada a salida y viceversa) se logró solucionar el problema.

Para poder hacer que solamente una de las entradas del driver recibiera el PWM, considerando lo siguiente de la Figura 7.5: los pines RE0 y RE1 son terminales del microcontrolador; OC1 es la señal PWM; y BIN1 y BIN2 son las entradas del driver DRV8833 que controlan la salida del motor B, teniendo claro lo anterior, para hacer que la señal PWM llegue solamente a la terminal

BIN1 por ejemplo se debe configurar los pines del microcontrolador de la siguiente manera: RE0 se configura como entrada, debido a que los puertos de entrada y salidas del PIC32MX534F064H funcionan mediante *flip flop* tipo D, y al configurarse como entrada, la alta impedancia del puerto D hace que la señal del PWM pueda ser adquirida también por el pin BIN1 del integrado DRV8833; el pin RE1 se configura como salida, el valor puede ser *high* o *low* dependiendo de la funcionalidad deseada (*fast decay* o *slow decay*), se puede consultar la Tabla 6.1 para determinar el valor y la orientación de giro, en el caso de que RE1 tenga el valor de 0, para evitar que la terminal OC1 quedara conectada directamente a 0 V y esto pudiera dañar el puerto OC1, se conectó una resistencia entre las terminales OC1 y RE1, de esta forma la tensión de 3.3 V que provienen de la señal PWM cae solamente en la resistencia R1, mientras que en la terminal BIN2 estaría a 0 V y no percibiría la señal PWM, en el caso de que RE1 tenga el valor de 1, cuando el PWM esté en alto, estaría a la misma referencia que RE1 por lo tanto la tensión en la resistencia R1 sería de 0 V, pero cuando el PWM esté en bajo, la tensión de la resistencia R1 sería de 3.3 V, en ambos casos la terminal BIN2 solamente detectaría 3.3 V. El valor de la resistencia R1 tiene que ser mayor a 330 Ω para asegurar que la corriente sea inferior a 10 mA.

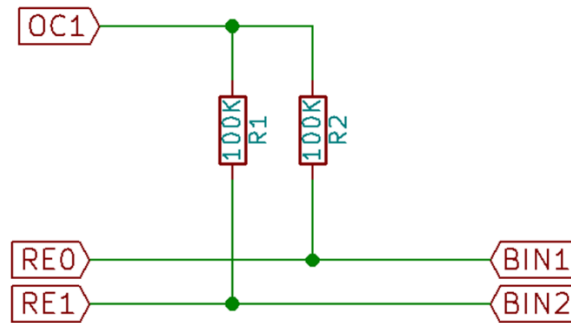


Figura 7.5. Esquemático de la configuración utilizado para el giro de los motores.

7.2.1. Modos de funcionamiento del driver DRV8833.

Para recircular la corriente del motor se puede realizar de dos formas, *fast decay* y *slow decay*, se realizaron pruebas con el motor 100:1. En modo *fast decay*, en la Figura 7.6 se muestra la señal de una de las terminales del motor a tierra, en este modo de funcionamiento, se puede apreciar que la

tensión pasa de 9 V a -9 V en un tiempo muy corto, esto se debe al cambio de polaridad que se realiza cuando el PWM pasa al valor de cero, en la Figura 6.9 se puede apreciar la trayectoria de la corriente, la cual no cambia de dirección, pero deja de fluir hacia tierra y fluye hacia VM, con esto la corriente almacenada en la bobina se descarga rápidamente, además en la Figura 7.6 se aprecia que después de que la tensión cae a -9 V el sistema presenta un comportamiento subamortiguado, esto se debe a la interacción entre el capacitor de 3.3 nF en las terminales del motor (dado por el encoder) y las bobinas del motor. En modo *slow decay*, en la Figura 7.7 se muestra el comportamiento de la señal medido desde una de las terminales del motor y tierra, a diferencia del modo anterior, en este caso no existen valores de tensión negativos, esto se debe a que al pasar el PWM a cero, ambas terminales del motor pasan a tierra, por lo que la descarga de la energía almacenada es más lenta, en la Figura 6.9 se muestra la trayectoria de la corriente. Comparando ambos modos de funcionamiento, en el caso del *slow decay*, el motor genera más torque en el modo *fast decay* esto debido a que la bobina del motor todavía tiene energía almacenada para el siguiente ciclo del PWM, lo que le permite mantener el torque mientras el PWM está en cero.



Figura 7.6. Señal de salida en modo *fast decay*.

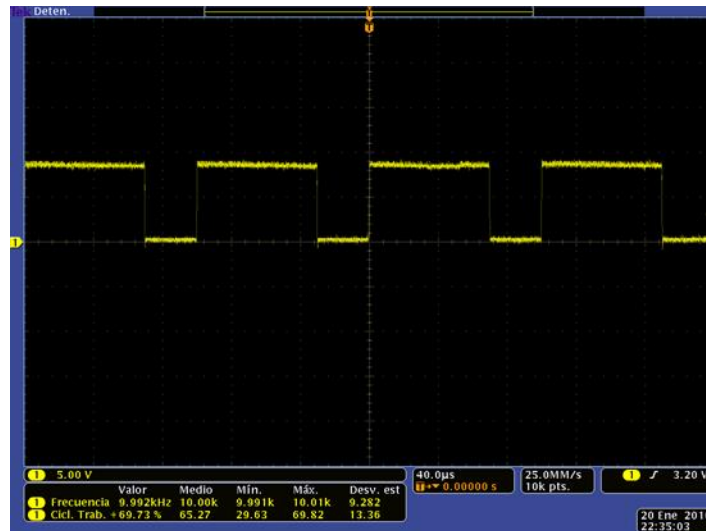


Figura 7.7. Señal de salida en modo *slow decay*.

Antes de realizar las pruebas con los motores, se realizó las pruebas del driver sin motores en las terminales, esto con el fin de verificar la funcionalidad del driver DRV8833, en la Tabla 7.1 se realizó las pruebas de verificación de la tensión de salida fuera congruente con el valor teórico, para estas pruebas se tomó como valor teórico la Ecuación 7.3, la finalidad de estas pruebas era comprobar si existe algún cambio significativo en la tensión final cuando se varía la frecuencia del PWM. La siguiente prueba consistía en verificar la linealidad de la tensión de salida respecto al PWM de entrada, Tabla 7.2 en la se muestran los datos obtenidos.

Tabla 7.1. Verificación de la tensión de salida del driver sin motor.

<i>Ciclo de trabajo</i> (%)	<i>Tensión de entrada</i> (V)	<i>Frecuencia</i> (kHz)	<i>Salida</i> (V)	<i>Salida teórica</i> (V)	<i>Error</i> (%)
40	3.3	100	1.31	1.32	0.75
60	3.3	100	1.97	1.98	0.50
50	3.3	1	1.64	1.65	0.61

$$V_{final} = \frac{V_{entrada} * CicloTrabajo(\%)}{100} \quad 7.3$$

Tabla 7.2. Prueba de linealidad de la tensión de salida del driver.

<i>Ciclo de Trabajo</i> (%)	<i>Tensión empírica</i> (V)	<i>Tensión teórica</i> (V)	<i>Error</i> (%)
30	0.98	0.99	1.01
40	1.31	1.32	0.76
60	1.97	1.98	0.51
80	2.64	2.64	0

Como se muestra en la Tabla 7.1 y en la Tabla 7.2, el error en la mayoría de los datos no supera el 1% por lo que se puede asegurar que se comportó como se esperaba, por lo menos a bajas tensiones y sin conectar los motores. Al conectar los motores se espera que la relación no se cumpla con un error menor al 1%, esto debido a que con la utilización de los motores, estos inducen caídas de tensión debido al aumento de la corriente de consumo y a las pérdidas características de los motores.

7.3. Pruebas con los motores.

7.3.1. Caracterización de los motores.

Para las pruebas con los motores primero se realizó la caracterización del motor, como se explicó en el apartado 6 correspondiente a los motores, se escogieron los mismos debido a que estos permiten gran variedad de cajas reductoras y manteniendo las mismas características eléctricas del motor, por lo cual se puede cambiar las prestaciones mecánicas deseadas (como velocidad y torque) sin que esto resulte en modificaciones significativas en el ensamblaje ni en la programación del microcontrolador, en la Figura 7.8 se muestran las partes del gearmotor, compuestas por el motor CD y la caja reductora.

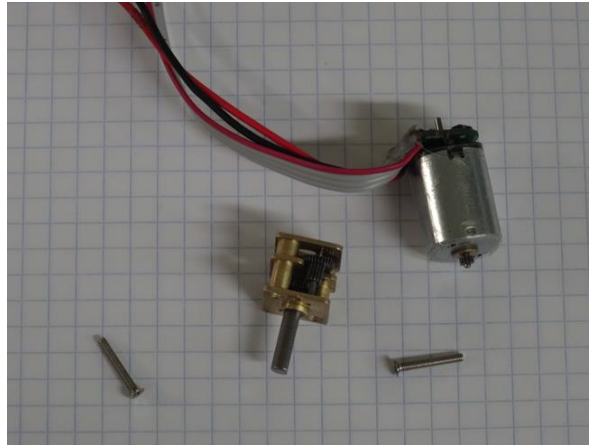


Figura 7.8. Motor CD y caja reductora utilizada en el ROBOBO 2.0.

Se realizó las pruebas a los motores 75:1 y 150:1, el objetivo de la prueba era realizar las gráficas que describan el comportamiento del motor dentro del rango de funcionamiento. Se realizaron dos tipos de pruebas, en vacío y a motor bloqueado. Con la prueba en vacío se obtuvo las velocidades nominales a distintos rangos de tensión, con esto se puede determinar cuáles son las velocidades que se pueden poner como referencia para el controlador, se busca que el rango de tensión del motor sea entre 2 V y 7 V. En las pruebas con el motor bloqueado (0 RPM) se obtuvo la corriente máxima que consumiría el motor a diferente tensiones de alimentación, la realización de esta prueba es importante para determinar los límites de corriente que se deben de configurar en el driver del motor. En la Tabla 7.3 y en la se muestran los datos obtenidos de los motores 75:1 y 150:1 respectivamente, la prueba se realizó cambiando el ciclo de trabajo en el driver del motor, a cada ciclo de trabajo se medía la corriente tanto en vacío como a motor bloqueado, se medía la velocidad angular, posteriormente se bloqueaba el motor para registrar la máxima corriente que fluía por los motores.

Tabla 7.3. Prueba del motor 75:1 en vacío y bloqueado.

<i>Ciclo de trabajo</i> (%)	<i>Tensión</i> (V)	<i>En Vacío</i>		<i>Bloqueado</i>
		<i>Corriente (mA)</i>	<i>Velocidad (RPM)</i>	<i>Corriente (mA)</i>
20	1.88	90	126	180
25	2.3763	100	162	250
30	2.968	120	200	330
35	3.3690	120	235	400
40	3.8626	120	265	490
45	4.3537	140	300	590
50	4.8494	140	333	690
55	5.3450	150	367	750
60	5.837	150	397	890
65	6.353	170	422	1000
70	6.854	160	452	1080
75	7.349	150	488	1240

Gráfica 7.1. Tensión vs Corriente del motor bloqueado.

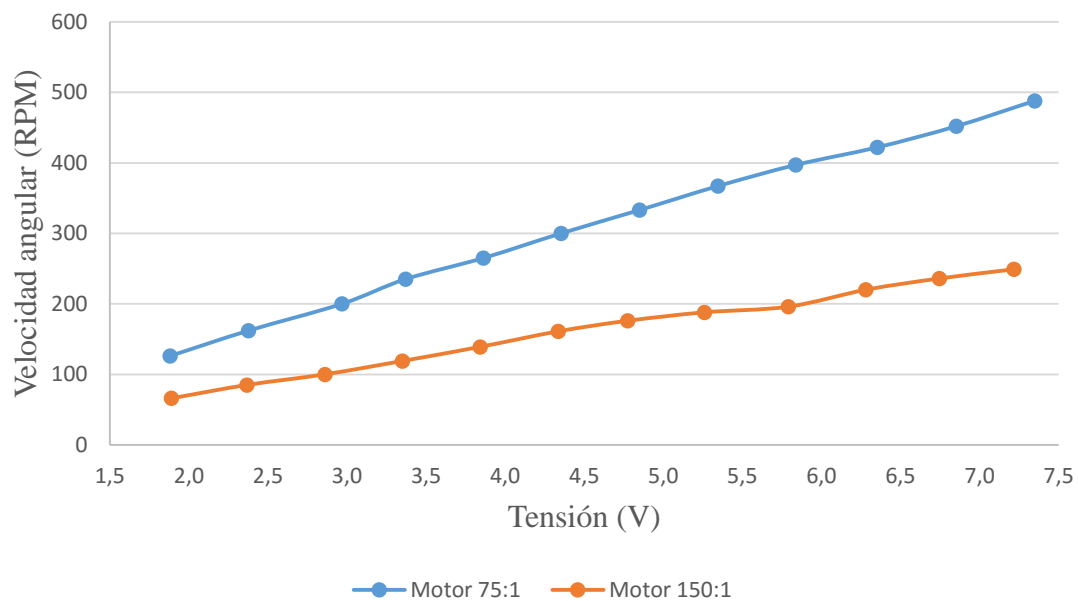
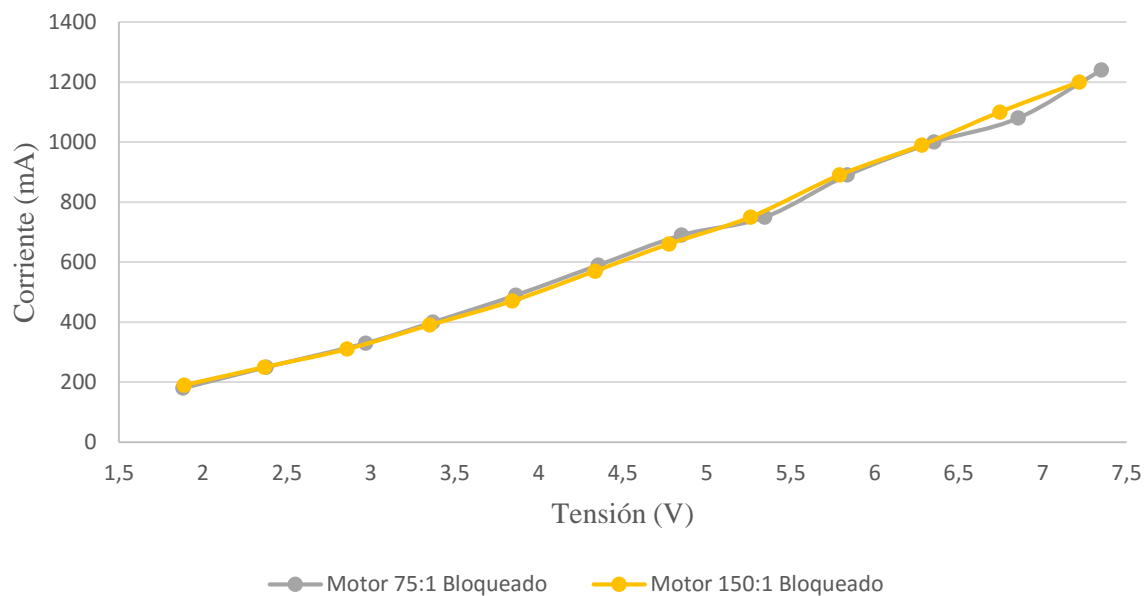


Tabla 7.4. Prueba del motor 150:1 en vacío y bloqueado.

<i>Ciclo de Trabajo</i> (%)	<i>Tensión</i> (V)	<i>En vacío</i>		<i>Bloqueado</i>
		Corriente (mA)	Velocidad (RPM)	Corriente (mA)
20	1,8883	100	66	190
25	2,3669	100	85	250
30	2,859	110	100	310
35	3,3502	120	119	390
40	3,8415	120	139	470
45	4,3352	130	161	570
50	4,7744	140	176	660
55	5,2607	140	188	750
60	5,791	140	196	890
65	6,281	150	220	990
70	6,746	150	236	1100
75	7,218	160	249	1200

Gráfica 7.2. Tensión vs Velocidad angular en vacío.



Como se pueden apreciar en la Gráfica 7.2, las características eléctricas del motor son las mismas sin importar la reducción, sin embargo como se puede ver en la Gráfica 7.1 las características mecánicas cambian, por lo que existe una disyuntiva entre tener velocidad o torque en los motores.

7.3.2. Pruebas en vacío.

Para las prueba de los motores en vacío y en el suelo, los ejes de las ruedas eran acoplados directamente en el eje del motor así como se muestra en la Figura 7.9, para los dos PWM se utilizó como frecuencia de operación 10 kHz, debido a que según las pruebas realizadas en la Tabla 7.1, la tensión promedio de salida del driver es indiferente de la frecuencia de operación.

Para la prueba en vacío se alimentó aproximadamente a 10.92V utilizando la fuente de alimentación que está en el laboratorio, la cual puede entregar como máximo 10 A, el acople del motor junto a las ruedas es como se muestra en la Figura 7.9.

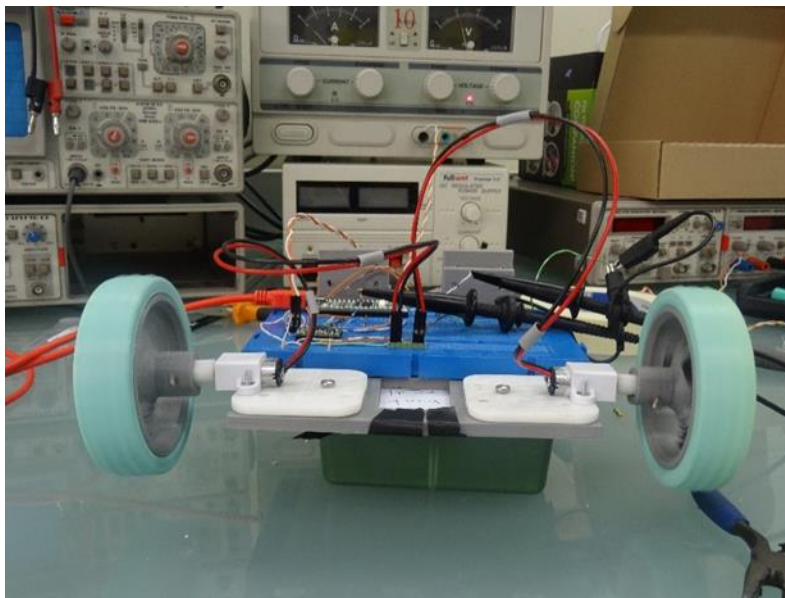


Figura 7.9. Montaje de los motores, driver, MINI32 y ruedas para las pruebas en vacío.

En la Tabla 7.5 se muestran las cinco pruebas realizadas a distintos ciclos de trabajo, al compararlo con la Tabla 7.2, se puede apreciar que el error aumentó, esto se debe a que al conectar los motores, los componentes eléctricos consumen potencia, además los componentes pasivos del motor distorsionan la señal a razón de las constantes de tiempo.

Tabla 7.5. Prueba de los motores en vacío con el acople de las llantas.

<i>Ciclo de Trabajo</i> (%)	<i>Entrada</i> (V)	<i>Salida Empírica</i> (V)	<i>Salida Teórica</i> (V)	<i>Error</i> (%)
10	10.92	1.04	1.092	4.76
14	10.91	1.47	1.5274	3.76
22	10.91	2.32	2.4002	3.34
54	10.92	5.76	5.8968	2.32
74	10.86	7.93	8.0364	1.32

7.3.3. Prueba en el suelo en línea recta.

Para la prueba en el suelo, se utilizó el diagrama que se muestra en la Figura 7.10. El objetivo de esta prueba fue verificar el funcionamiento de los motores tanto en velocidad como en potencia mecánica. Se utilizaron los motores 100:1, se utilizó el oscilador externo de 8MHz del microcontrolador y la alimentación se realizó con una batería de 11.1V y de 1000mAh.

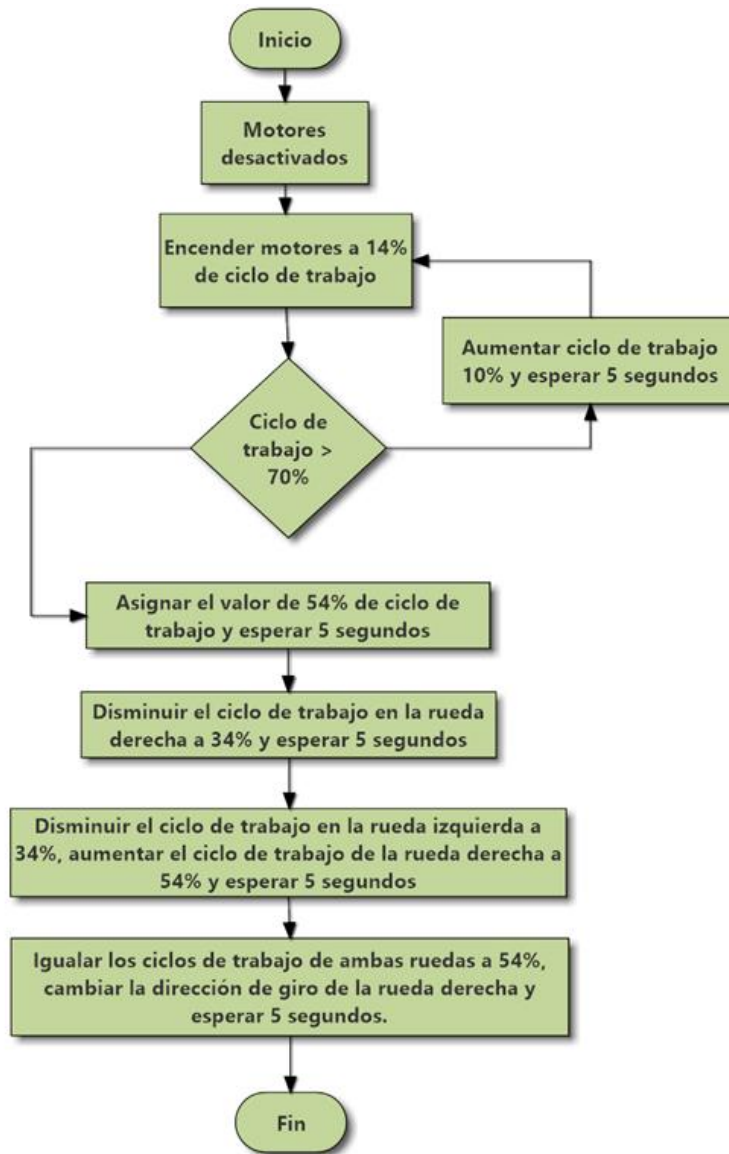


Figura 7.10. Diagrama implementado en la prueba de los motores en el suelo.

La prueba se realizó en el suelo del laboratorio del GII, uno de los retos de la prueba era que los motores pudieran subir una rampa que se improvisó en el lugar, los motores debían ser capaces de poder subir la inclinación a una velocidad razonable (la definición exacta de la velocidad se realizó más adelante, en el apartado del controlador) aun cuando el movimiento se iniciara sin impulso previo. La rampa utilizada se muestra en la Figura 7.11.

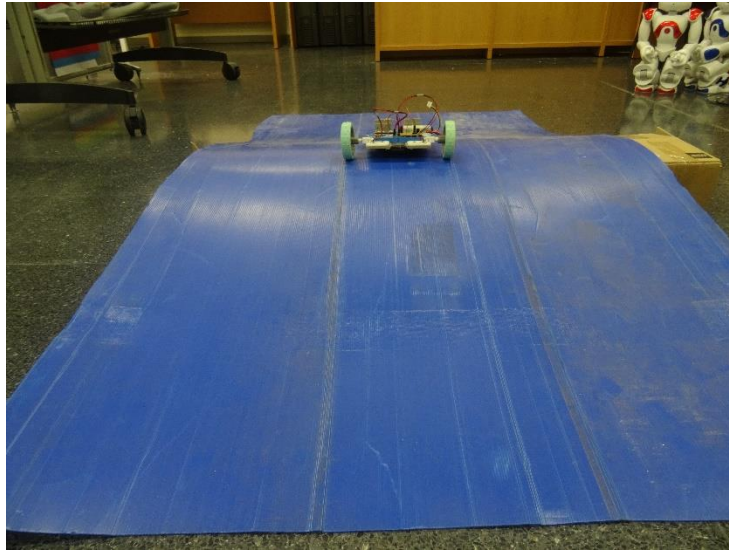


Figura 7.11. Rampa utilizada en la prueba de los motores.

Los motores fueron capaces de mover satisfactoriamente todos los componentes dentro del carro, sin embargo el movimiento se dio a partir de los 24% de ciclo de trabajo (lo que equivale aproximadamente a 2.8V) aunque al estar funcionando bajo carga, la tensión debió disminuir y la corriente aumentar. Para poder subir la rampa, con impulso era capaz de subirla a partir de los 44% de ciclo de trabajo, sin embargo para poder subir la rampa sin impulso tuvo que llegar hasta los 54% de ciclo de trabajo para superar el torque inicial para comenzar el movimiento.

Uno de los problemas que se encontraron en la prueba fue que el acople entre el motor y la rueda no es la correcta, debido a que cuando esta giraba tenía un movimiento irregular causado debido a que los centro del eje y de las ruedas no estaba alineado, las piezas fueron realizadas con una impresora 3D y por problemas de precisión las piezas no habían quedado con el acabado correcto. Este inconveniente se reflejaba en la trayectoria del ROBOBO en el suelo, ya que el movimiento de traslación no era en dirección recta sino que tenía una trayectoria en “zigzag”.

Después de las pruebas los motores comenzaron a generar ruidos extraños al girar, aun estando en vacío, este comportamiento fue causado por el movimiento irregular de las ruedas, las cuales desgastaron los dientes de la caja reductora que traen los motores. Esto es respaldado por los siguientes dos hechos: si se volvía a encender los motores, el ruido solamente se escuchaba cuando giraban en el mismo sentido que al momento de la prueba, al girar en sentido contrario el sonido

desaparecía; además el desgaste era evidente al examinar la caja reductora más detalladamente utilizando una lupa.

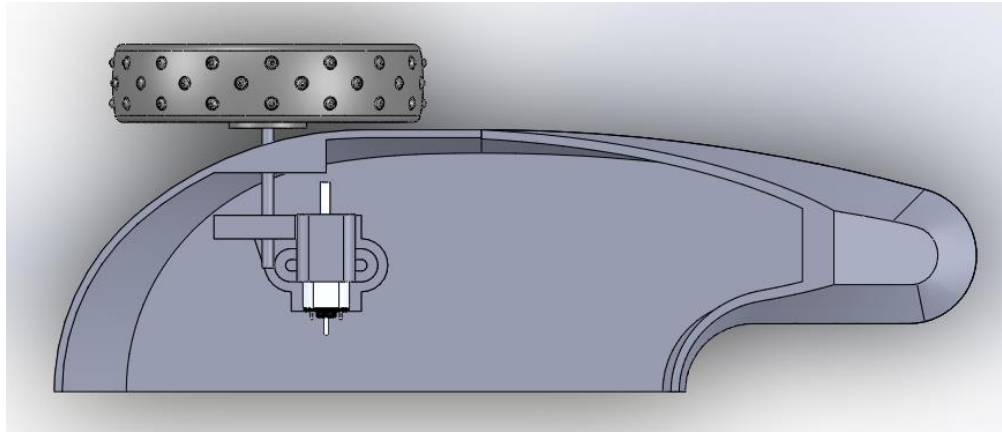


Figura 7.12. Modelo en 3D de la disposición del eje del motor y el eje de la rueda.

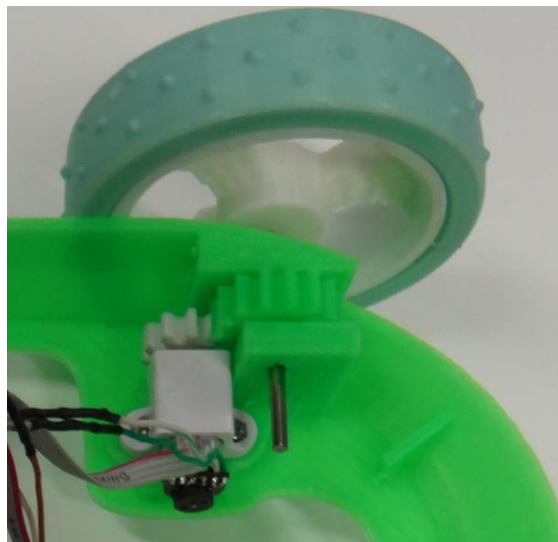


Figura 7.13. Implementación real del sistema de transmisión.

La solución a este problema fue cambiar la forma en se acopla el motor al eje, se realizó como se muestra en la Figura 7.12 y en la Figura 7.13, ya que de esta forma los esfuerzos mecánicos provocados por el movimiento de las ruedas, no afectan al eje del motor.

7.4. Funcionamiento y configuración del encoder.

Después de conectar el encoder al motor se realizó la prueba de funcionamiento, para verificar que no existiera ningún problema en la instalación y para comprobar que funcionara tal cual lo explica la hoja de datos. En la Figura 7.14 se muestra las formas de ondas obtenidas en el osciloscopio, para esta prueba se utilizó el motor 75:1 y se le aplicó tensión de 5V, las gráficas corresponden a los canales A y B del encoder. El comportamiento es como se esperaba y coincide con lo expresado en las hojas de datos. Sin embargo para efectos de la implementación real del ROBOBO 2.0, solamente se va utilizar un canal para el encoder, lo que limita la detección del sentido de giro, no obstante se va determinar la dirección del giro a partir del software y de los parámetros de configuración.

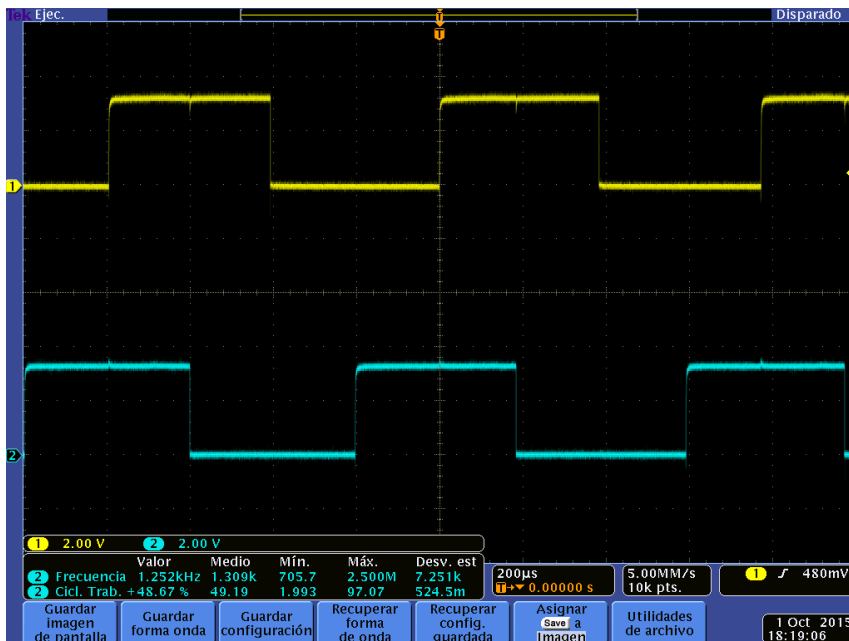


Figura 7.14. Forma de onda de la señal del encoder utilizado en los motores 75:1.

7.4.1. Implementación del encoder.

Para poder implementar el encoder se utilizó los pines del microcontrolador con la función de medir el tiempo que dura una señal en alto, por lo que se utilizó los *Input Capture (IC)*, esta configuración

permite capturar el valor de los *timer* en base en un evento en la entrada del pin del microcontrolador. Se puede configurar para que capture el valor del temporizador cada vez que pase lo siguiente:

- Cada flanco de bajada.
- Cada flanco de subida.
- Cada cuatro flancos de subida.
- Cada 16 flancos de subida.
- Cada vez que pase un flanco de subida y uno de bajada.

El IC puede almacenar hasta cuatro valores en el búfer de salida, la estructura de datos es FIFO (*First In First Out*), la interrupción se puede configurar para que se active cada cierto número de eventos.

El búfer almacena el valor del registro del temporizador, en la rutina de interrupción correspondiente al IC, primero se debe extraer el dato del búfer del IC correspondiente (ICxBUF), como el dato que almacena el búfer es el contador del temporizador y este no es una medida directa de tiempo, sino que es el valor de los pulsos del temporizador (valor entre 0 y 65 535), de acuerdo a la configuración de reloj en el bus de periféricas y en los escalares del temporizador se puede realizar la conversión a segundos.

Para calcular la velocidad angular, se sabe que cada interrupción del IC proveniente del encoder sucede cada 2.094 radianes de giro del motor (sin la caja reductora), este valor se escala de acuerdo a la reducción que tenga la caja reductora. Definida esta constante y con el valor en el ICxBUF se hace la conversión de [rad/s] a RPM.

7.5. Diseño del controlador.

Uno de los objetivos es que el ROBOBO sea capaz de mantener una velocidad constante sin importar si está subiendo/bajando por una rampa o se desplaza por una superficie plana. Además debía de tener como parámetros de entrada la velocidad a la que se desea mover y los segundos o

desplazamiento angular, esto último como condición de parada, y como parámetros de salida grados desplazados por la rueda. Por lo que se decidió realizar un controlador.

Para realizar el controlador se siguieron los siguientes pasos:

- Realizar el modelo teórico del motor.
- Realizar las pruebas al motor para obtener la relación entrada/salida.
- Aproximar varias funciones de transferencia.
- Obtener el modelo empírico.
- Diseñar un controlador para que cumpla con los requisitos deseados.
- Comparar los resultados de los distintos controladores.
- Implementar en el PIC32MX534F064H.
- Comprobar el comportamiento del motor y verificar que cumple con los requisitos.

7.5.1. Modelado Teórico.

Para realizar el modelo teórico del motor, se asumió que es un motor CD controlado por el voltaje inducido en las bobinas del estator y de imanes permanentes. La modelización del motor se realiza para tener un panorama general de cómo debería ser la ecuación empírica, pero en este caso no se van a buscar los valores numéricos del motor, debido a la falta de tiempo para realizar todas las pruebas necesarias. El modelo teórico se va a realizar a partir de la representación de la Figura 7.15.

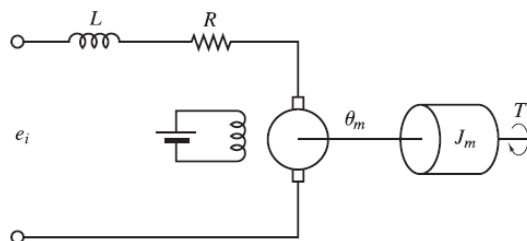


Figura 7.15. Representación del motor DC.

Primero se determinan las ecuaciones que representan la parte mecánica y la parte eléctrica:

$$ei(t) = L * \frac{dia(t)}{dt} + R * ia(t) + Kb * \omega(t)$$

$$T(t) = B * \omega(t) + Jm * \frac{d\omega(t)}{dt}$$

$$T(t) = Ki * ia(t)$$

Aplicando transformada de Laplace.

$$Ei(s) = L * Ia(s) * s + R * Ia(s) + Kb * \Omega(s) \quad 7.4$$

$$T(s) = B * \Omega(s) + Jm * \Omega(s) * s \quad 7.5$$

$$T(s) = Ki * Ia(s) \quad 7.6$$

Despejando $Ia(s)$ de la Ecuación 7.4 y sustituyéndola en la ecuación 7.6 se obtiene:

$$\frac{Ei(s) - Kb * \Omega(s)}{L * s + R} * Ki = B * \Omega(s) + Jm * \Omega(s) * s \quad 7.7$$

Se obtiene que:

$$\frac{\Omega(s)}{Ei(s)} = \frac{Ki}{JmLs^2 + (JmRa + BL)s + BR + KbKi} \quad 7.8$$

La Ecuación 7.8 es la función de transferencia teórica que representa el comportamiento del motor, con este resultado se puede afirmar que la función de transferencia empírica debe tener dos polos y ningún cero.

7.5.2. Modelado empírico.

Para el modelado empírico se realizaron tres muestras, para todas ellas se utilizó el motor de reducción 75:1, como hubo problemas en medir la tensión directamente de las terminales del motor, se decidió que se iba medir con un multímetro y se tomaría como pulso de disparo la señal al pin SLEEP del driver DRV8833. Las pruebas se realizaron a 14% de ciclo de trabajo y a 50% de ciclo de trabajo, en cada una de estas a 10ms/div, 20ms/div y 40ms/div del osciloscopio, el cual tenía un muestreo de 10 000 datos.

La lógica utilizada para encontrar los datos necesarios para realizar el modelo se muestra en la Figura 7.17, esta secuencia se programó en Excel para obtener las gráficas de entrada y RPM ambas en función del tiempo. Sin embargo dependiendo de los ms/div del osciloscopio el valor en la casilla en la que se calcula el tiempo que dura en girar 0.4° puede variar a: 0.00001 para 10ms/div, 0.00002 para 20ms/div y a 0.00004 para 40ms/div.

En las pruebas se obtuvieron las formas de ondas que se muestra en la Figura 7.16, en la Figura 7.18 y en la Figura 7.19, a partir de los datos obtenidos del osciloscopio, en un archivo con formato csv, se realizaron las gráficas de posición y velocidad angular.

En la primera prueba se realizó con un ciclo de trabajo del 14%, se le aplicó una entrada escalón de 1.57V, el osciloscopio estaba a 10 ms/div con un muestreo de 10 000 datos, lo que equivale a que cada dato era muestreado cada 0.00001 segundo. En la Figura 7.16 se muestra las señales captadas por el osciloscopio y en la Gráfica 7.3 y en la Gráfica 7.4 se muestran la posición y la velocidad angular respectivamente.



Figura 7.16. Forma de onda obtenida como salida del encoder en la primera prueba.

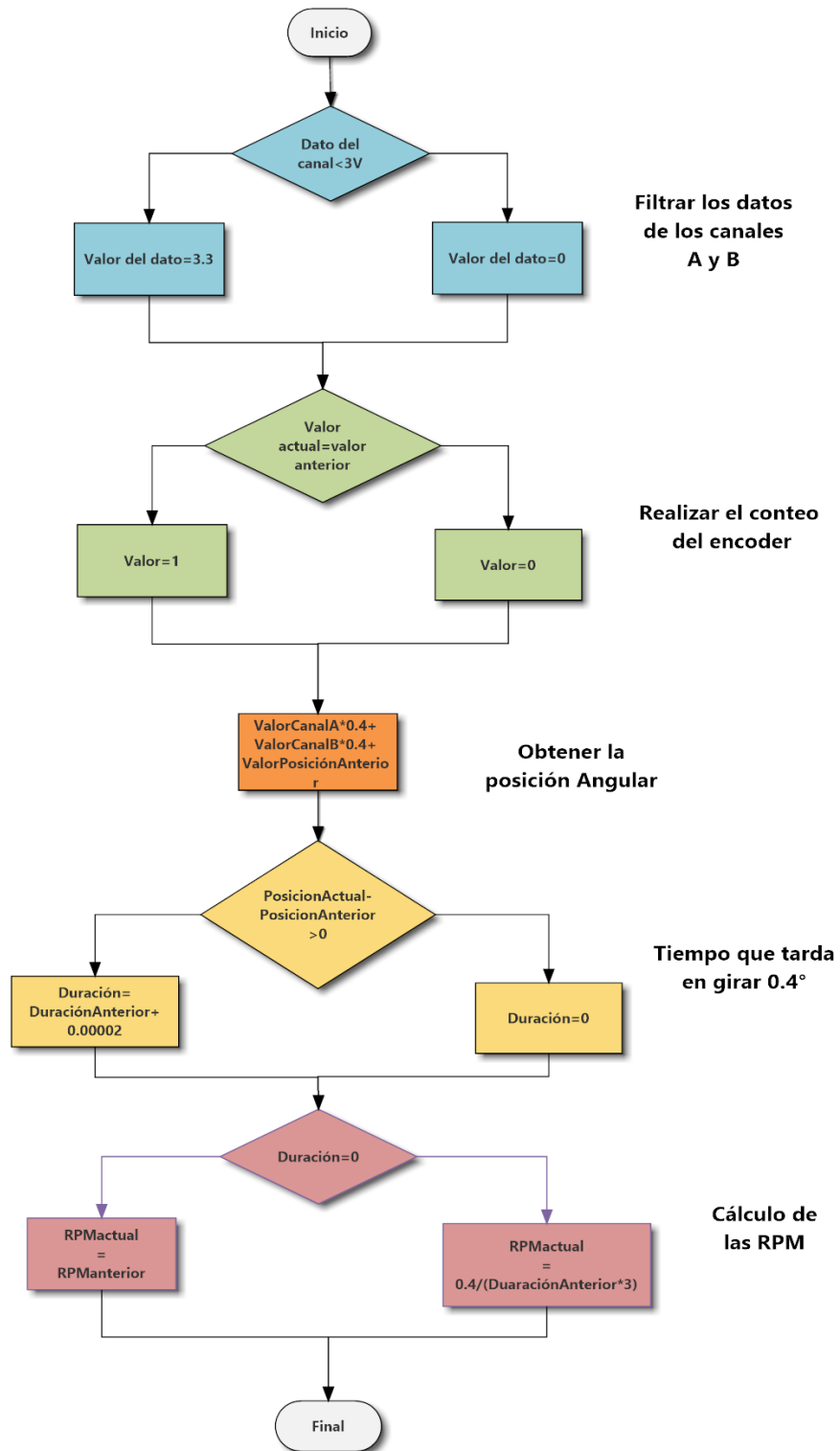
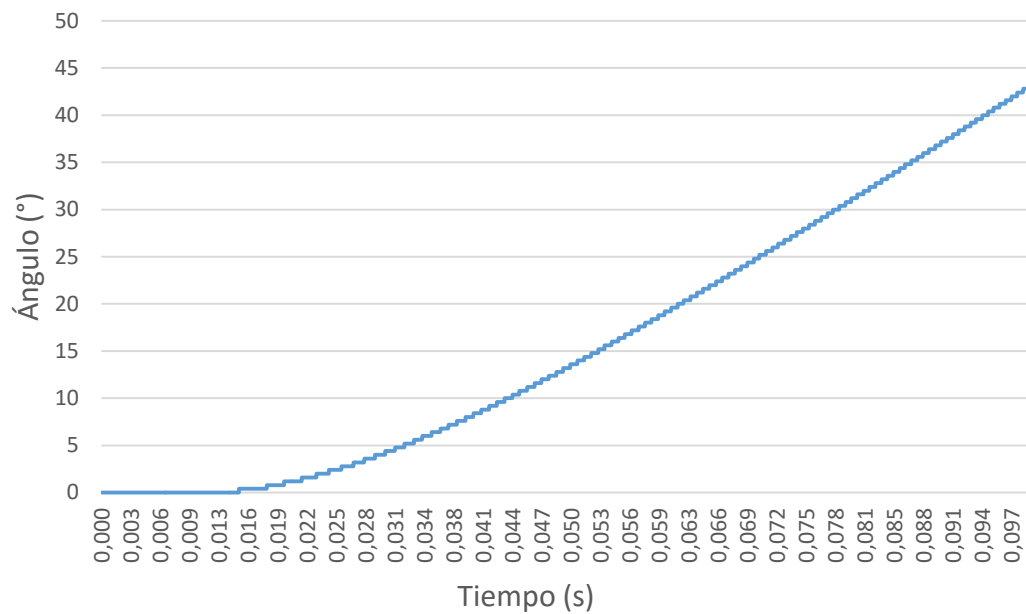
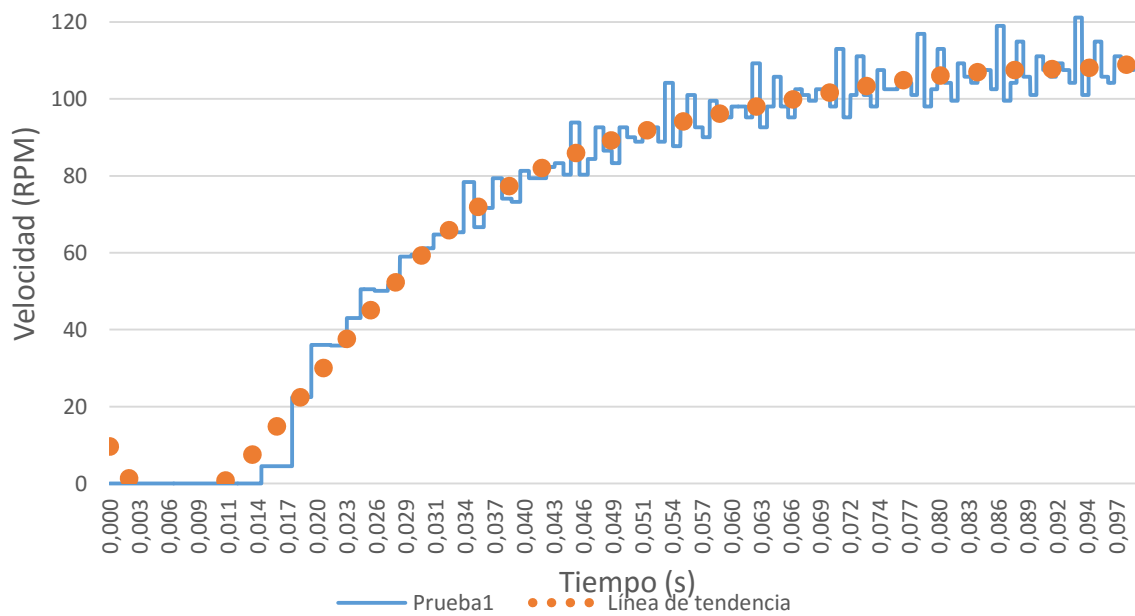


Figura 7.17. Diagrama empleado para procesar los datos del encoder.

Gráfica 7.3 Posición angular del motor 75:1 en la primera prueba.



Gráfica 7.4. Velocidad angular del motor 75:1 en la primera prueba.



En la segunda prueba se realizó con un ciclo de trabajo del 14% al igual que en la primera prueba, se le aplicó una entrada escalón de 1.56V, el osciloscopio estaba a 40 ms/div con un muestreo de 10 000 datos, lo que equivale a que cada dato era muestreado cada 0.00004 segundo. En la Figura 7.18 se muestra las señales captadas por el osciloscopio y en las Gráfica 7.5 y en la Gráfica 7.6 se muestran la posición y la velocidad angular respectivamente.

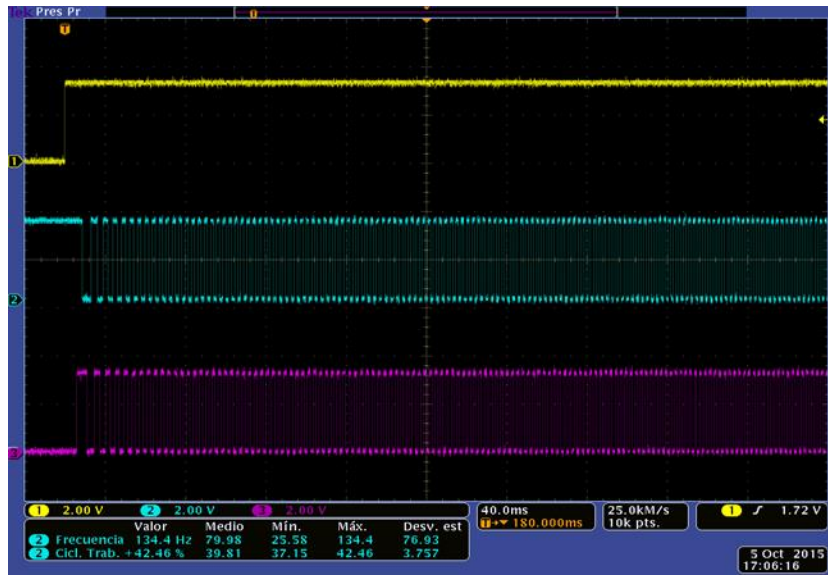
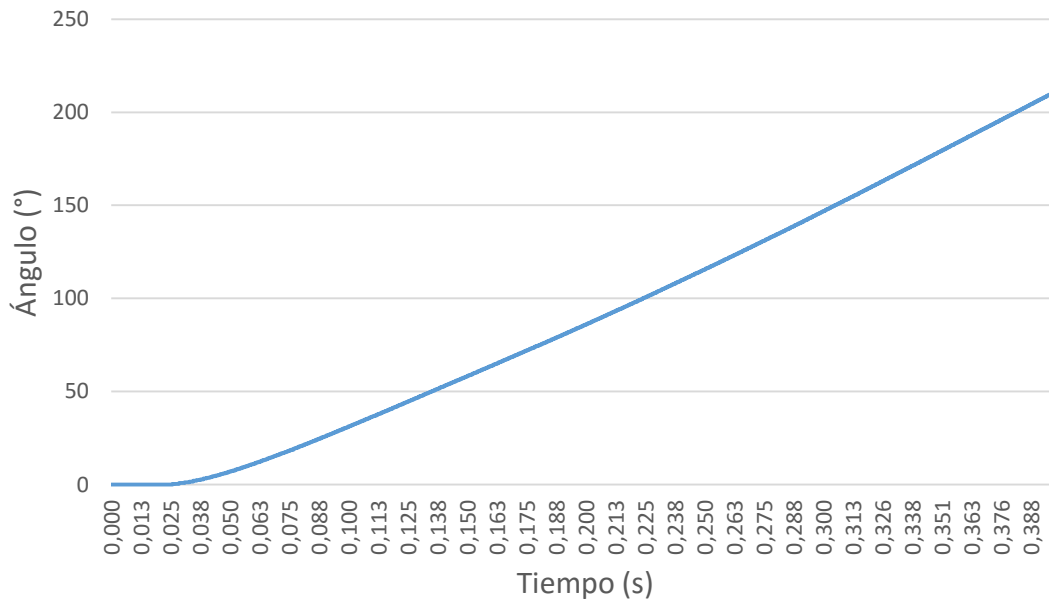
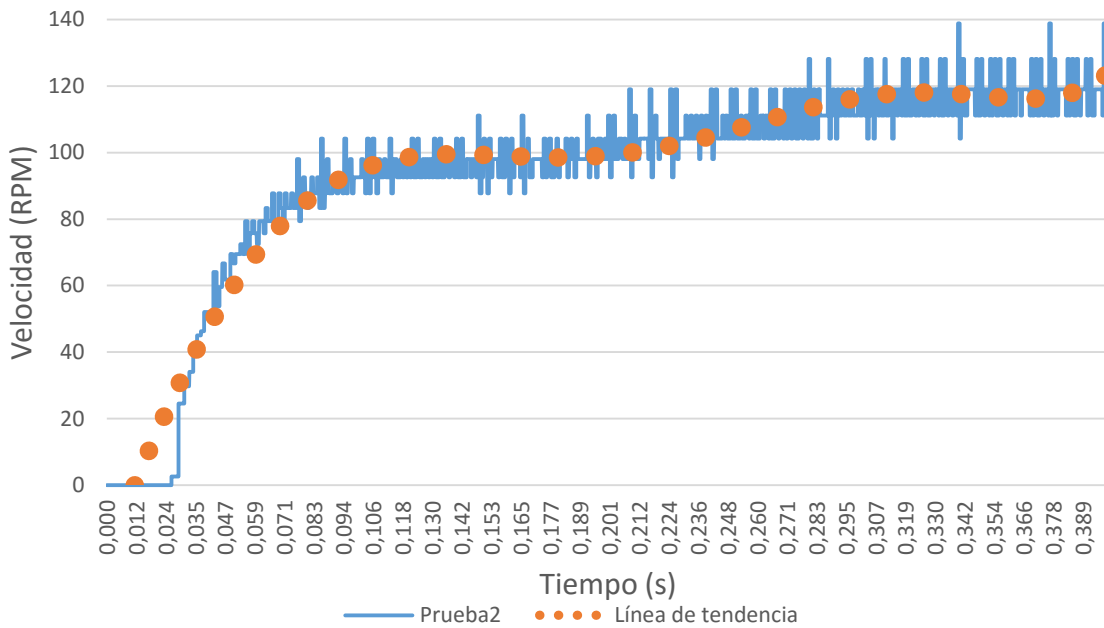


Figura 7.18. Forma de onda obtenida como salida del encoder en la segunda prueba.

Gráfica 7.5. Posición angular del motor 75:1 en la segunda prueba.



Gráfica 7.6. Velocidad angular del motor 75:1 en la segunda prueba.



En la tercera prueba se realizó con un ciclo de trabajo del 50%, se le aplicó una entrada escalón de 5.66V, el osciloscopio estaba a 20 ms/div con un muestreo de 10 000 datos, lo que equivale a que cada dato era muestreado cada 0.00002 segundo. En la Figura 7.19 se muestra las señales captadas por el osciloscopio y en la Gráfica 7.7 y en la Gráfica 7.8 se muestran la posición y la velocidad angular respectivamente.

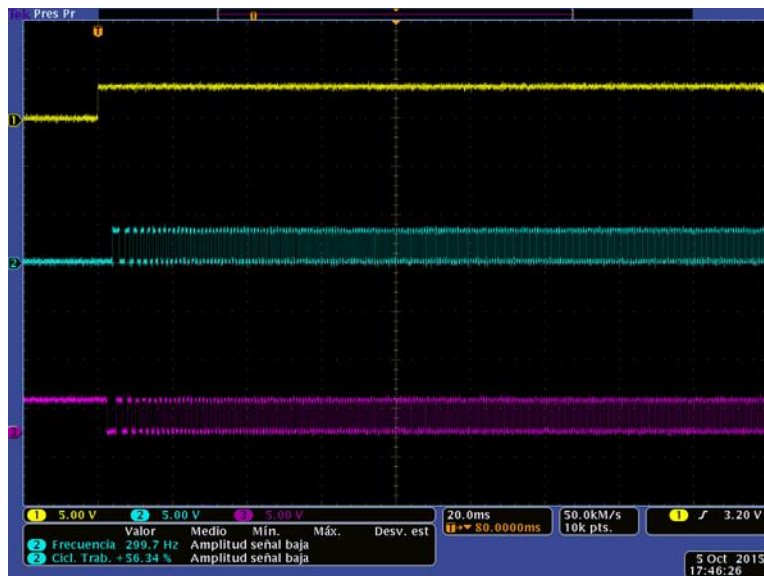
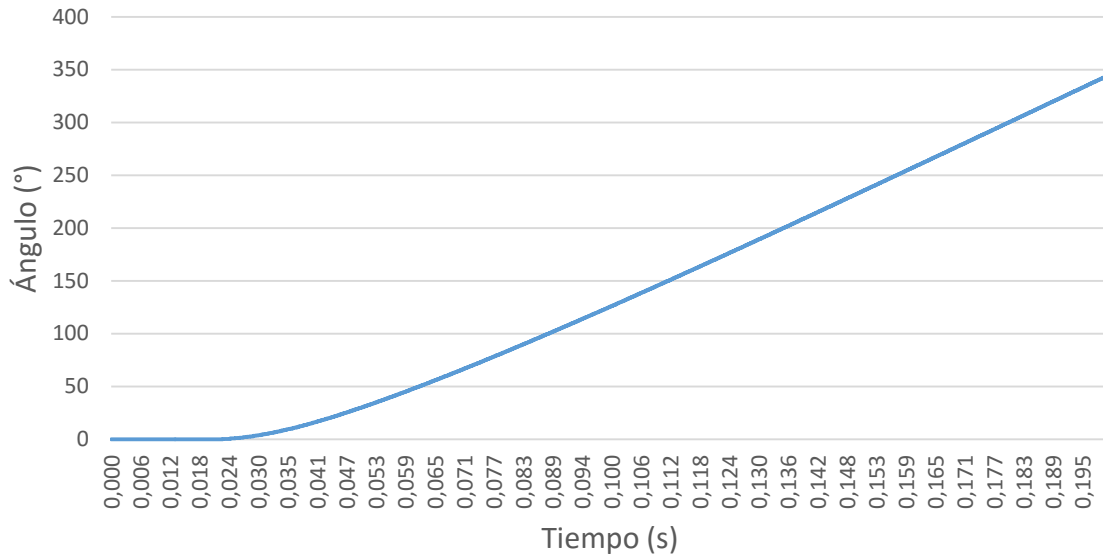
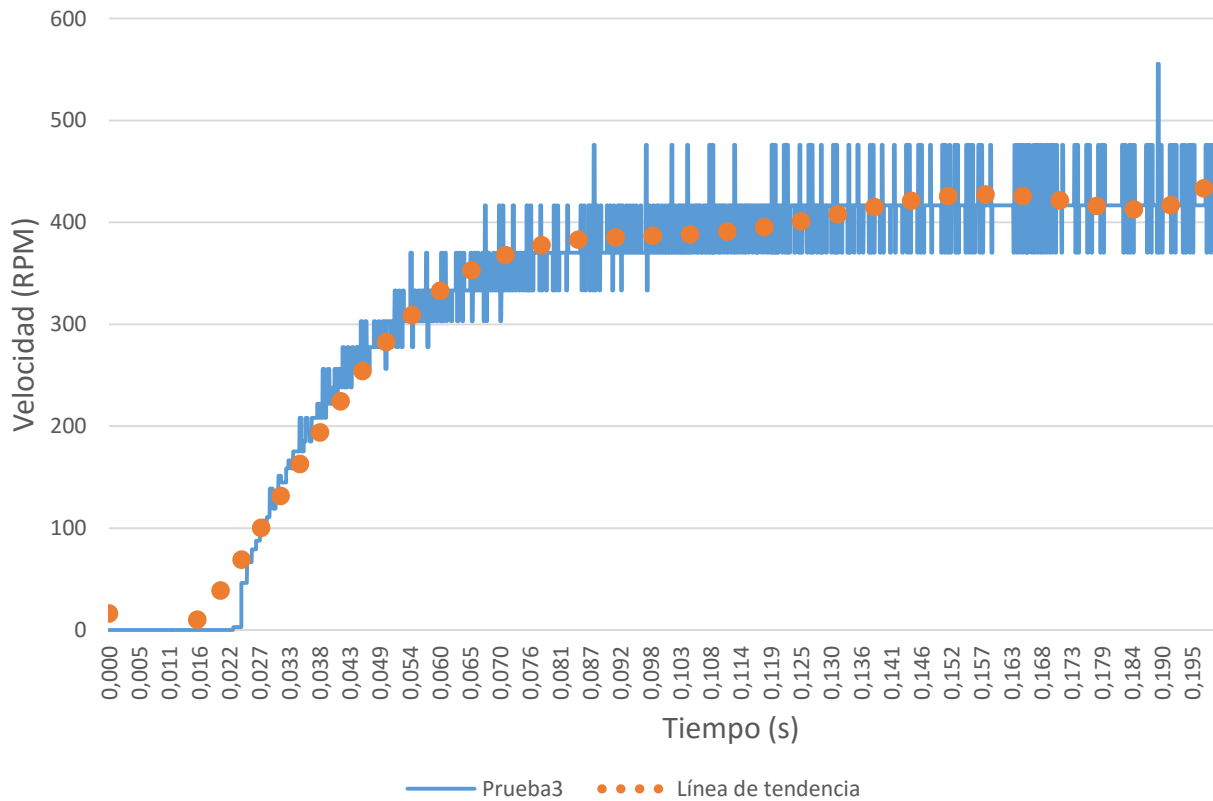


Figura 7.19. Forma de onda obtenida a la salida del encoder en la tercera prueba.

Gráfica 7.7. Posición angular del motor 75:1 en la tercera prueba.



Gráfica 7.8. Velocidad angular del motor 75:1 en la tercera prueba.



Como se puede observar en las gráficas anteriores, el sistema es estable respecto a la velocidad, sin embargo no mantiene la velocidad constante, sino que con cualquier perturbación en la salida se afecta directamente a la velocidad final, esto se debe a que el sistema no está retroalimentado.

Para obtener la función de transferencia se utilizó el software Matlab, con la función `ident` se realizó la aproximación de la función de transferencia. Para cada una de las pruebas se escogieron dos aproximaciones (una para los datos de velocidad usando los dos canales y otros solo usando un canal del encoder). En la Tabla 7.6 se muestra un resumen de las funciones de transferencia que dieron mayor coincidencia con los datos empíricos por cada prueba.

Tabla 7.6. Funciones de transferencia empíricas.

<i>Prueba</i>	<i>Función de transferencia</i>	<i>Coincidencia (%)</i>
1	$\frac{5.068 \times 10^5}{s^2 + 216.7 * s + 7687}$	90.42
2	$\frac{6.006 \times 10^5}{s^2 + 208.3 * s + 8796}$	90.16
3	$\frac{1.242 \times 10^6}{s^2 + 736.3 * s + 18250}$	77.68
4	$\frac{1.257 \times 10^6}{s^2 + 418.9 * s + 19490}$	93.33
5	$\frac{1.914 \times 10^6}{s^2 + 699.9 * s + 27840}$	89.53
6	$\frac{1.844 \times 10^6}{s^2 + 656.9 * s + 24930}$	80.55

Para comprobar cuáles de las funciones de transferencias se aproximaba más al comportamiento real del motor, se graficó cada función de transferencia junto con los datos correspondiente a las pruebas, sin embargo eso no generó ningún resultado contundente debido a que los datos originales tienen mucho ruido, por lo que se decidió aplicarle una entrada escalón de un valor de 6 a cada función, según los datos del fabricante el motor debe alcanzar una velocidad de 400 RPM, las gráficas obtenidas se muestran en las figuras a continuación:

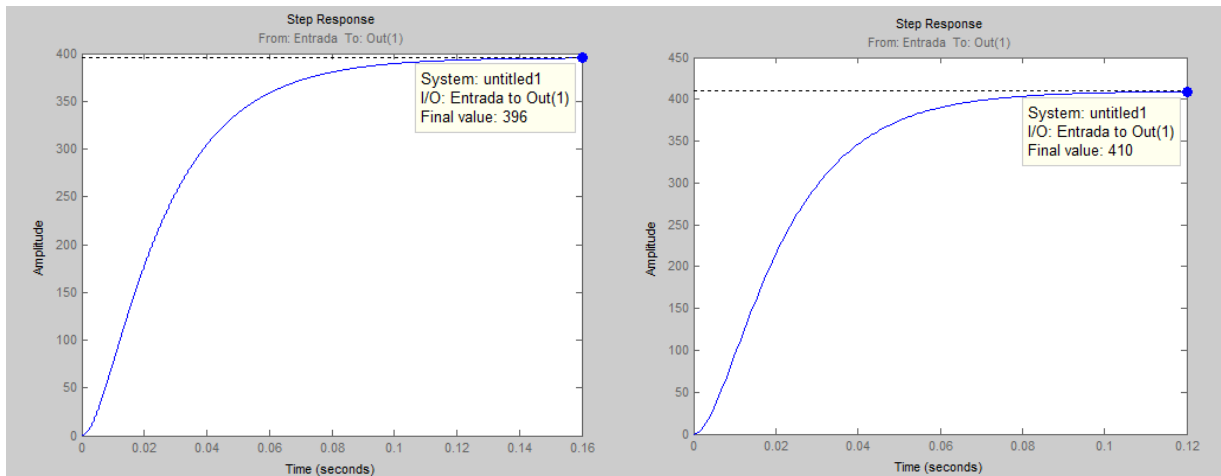


Figura 7.20. Respuesta ante un escalón con valor de 6, las pruebas 1 y 2.

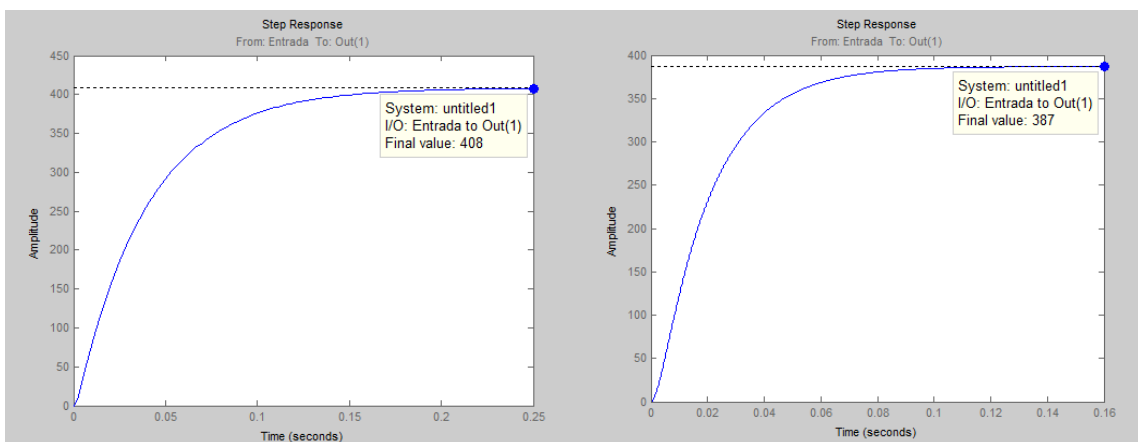


Figura 7.21. Respuesta ante un escalón con valor de 6, las pruebas 3 y 4.

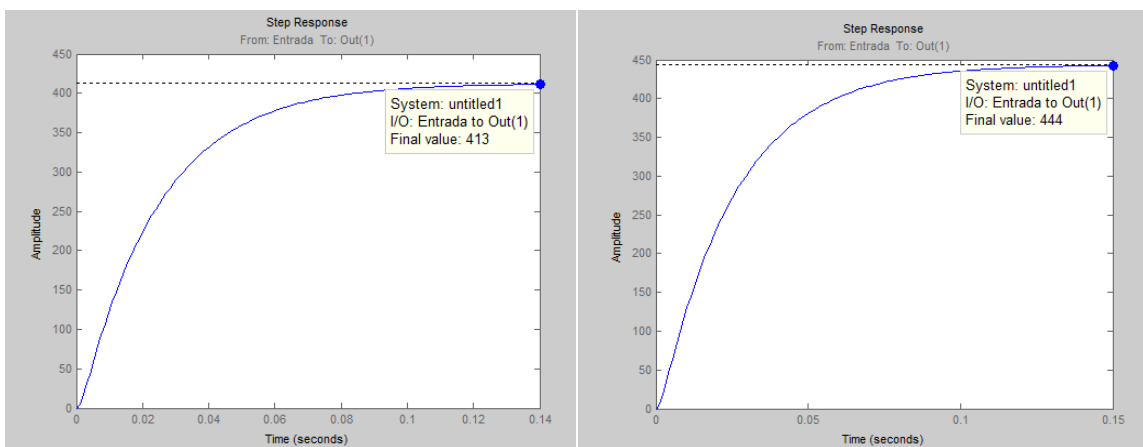


Figura 7.22. Respuesta ante un escalón con valor de 6, las pruebas 5 y 6.

Para poder seleccionar la función de transferencia que mejor se ajuste al modelo real del motor se recurrió a comparar todas las funciones encontradas con la gráfica de los datos de la prueba 2. En la Figura 7.23 se muestran cómo se ajustan a la curva las funciones de transferencia, debido a la velocidad de subida que tienen las funciones de las prueba 1 y 3 (colores verde y cian respectivamente) no se escogerán para cálculos posteriores, aun cuando estas funciones dieron los menores errores en la Figura 7.20 y en la Figura 7.21 respectivamente, las otra función que se elimina es la de la prueba 6 ya que es la que presenta el mayor error en la Figura 7.22. Se puede escoger cualquiera de las funciones de las pruebas 2, 4 y 5. La función con la que se va trabajar de aquí en adelante es la de la prueba 5 (color azul), se muestra en la ecuación 7.9.

$$G(s) = \frac{1.914 \times 10^6}{s^2 + 699.9 * s + 27840} \tag{7.9}$$

Esta función de transferencia también se puede escribir en términos de zpk (*zero-pole-gain*):

$$G(s) = \frac{1.914 \times 10^6}{(s + 657.59) * (s + 42.34)} \tag{7.10}$$

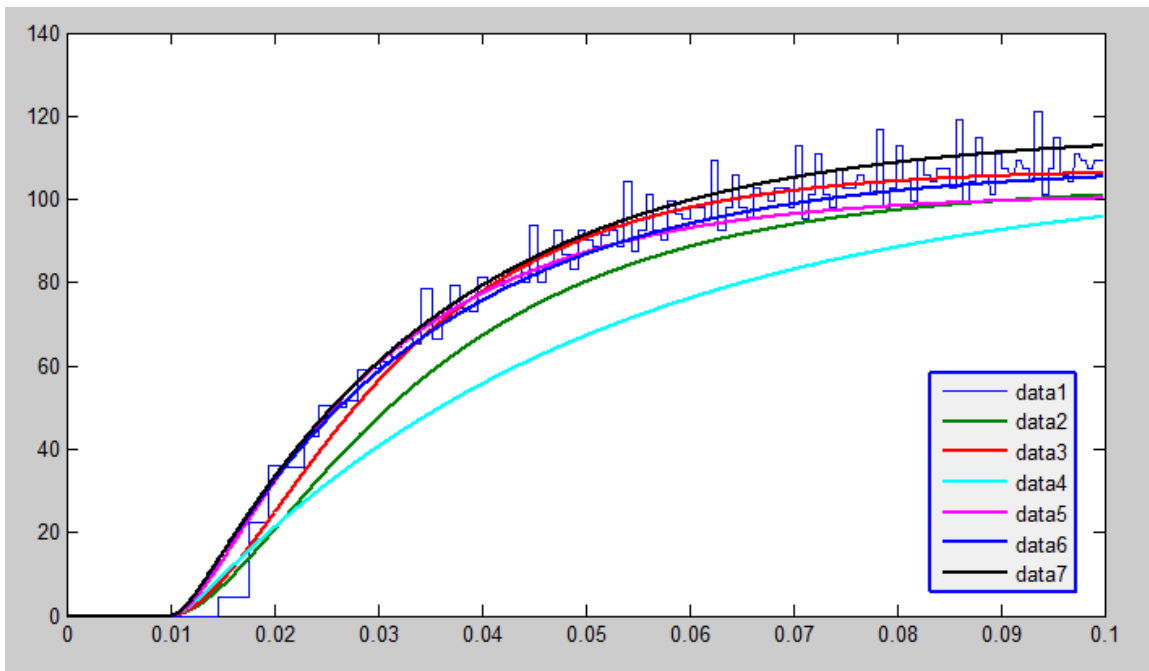


Figura 7.23. Coincidencia de las funciones de transferencia con los datos de la primera prueba.

En la Figura 7.23 los datos se asocian a las pruebas de la Tabla 7.6 de la siguiente forma:

- *Data 1*: Datos de la primera prueba del motor.
- *Data 2*: Ecuación empírica prueba 1.
- *Data 3*: Ecuación empírica prueba 2.
- *Data 4*: Ecuación empírica prueba 3.
- *Data 5*: Ecuación empírica prueba 4.
- *Data 6*: Ecuación empírica prueba 5.
- *Data 7*: Ecuación empírica prueba 6.

7.5.3. Controladores.

El lugar de las raíces de la función de transferencia de la Ecuación 7.10 se muestra en la Figura 7.24, debido a la ubicación de los polos el sistema responde rápido, además es estable (como ya se había concluido anteriormente).

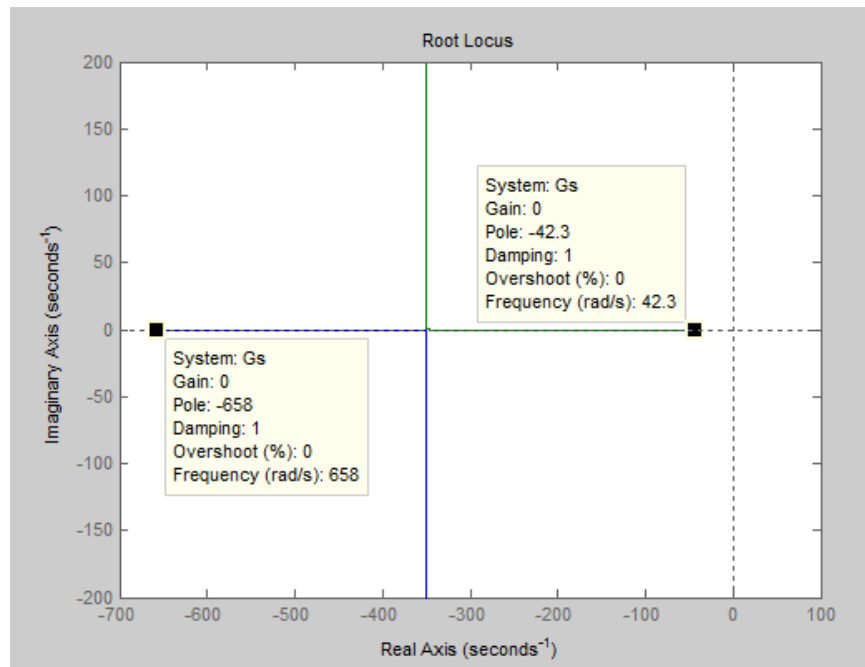


Figura 7.24. Lugar de las raíces de la función de transferencia del motor.

Para diseñar el controlador se definió que el motor debe cumplir con las siguientes condiciones:

- Tiempo de estabilización al 5% = 0.5 s.
- Sobreimpulso <5%.
- Error en estado estable = 0.

Con las características anteriores, los polos dominantes del sistema se deben ubicar en la región que se muestra en la Figura 7.25.

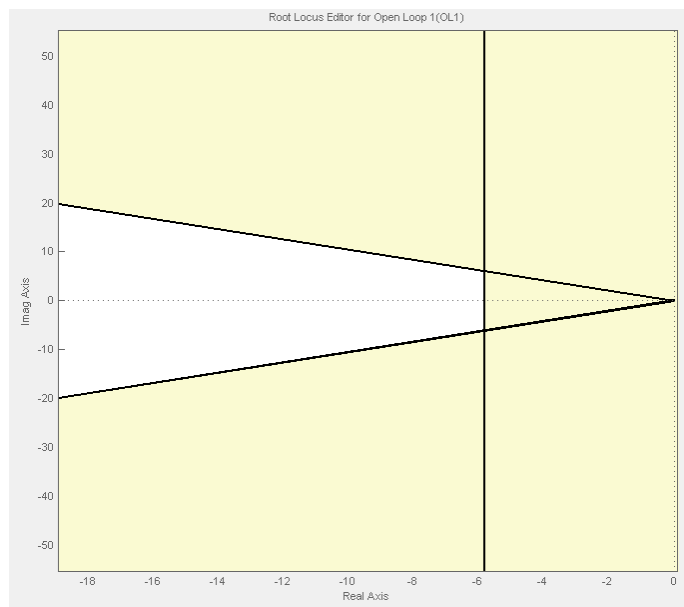


Figura 7.25. Región donde se deben ubicar los polos dominantes.

Se escogió los polos deseados en:

$$-6 \pm j * 0.05$$

Como el sistema es estable, sin sobreimpulso, además por la posición de los polos reacciona rápidamente, se decidió realizar controladores sencillos, se hizo un compensador de atraso (Ecuación 7.11) y controladores PI y I (Ecuaciones 7.12 y 7.13 respectivamente).

$$Klag(s) = \frac{0.1 * s + 0.1}{s + 0.1} \quad 7.11$$

$$PI(s) = \frac{0.1 * s + 0.1}{s} \quad 7.12$$

$$I(s) = \frac{0.08}{s} \quad 7.13$$

En la Figura 7.26, en la Figura 7.27 y en la Figura 7.28 se muestra el comportamiento del sistema, compuesto por el motor y el controlador ante una entrada de escalón unitario (color azul) y la tensión entregada por el controlador necesaria para alcanzar la referencia (color verde). Las curvas están normalizadas por esa razón la escala del eje de amplitud es distinta para la curva verde que para la curva azul. Usando el controlador Proporcional-Integral y el controlador Integral, el error en estado estable es 0% (Figura 7.27 y la Figura 7.28 respectivamente), mientras que con el controlador de atraso el error es de 1,4% (Figura 7.26) y la tensión necesaria para alcanzar 1 RPM es de 0.0145 V, que si escalamos este valor para alcanzar la referencia de 400 RPM, la tensión sería de 5.8 V, lo cual es un valor que está dentro del rango de funcionamiento del motor y además coincide con los datos de la Tabla 7.3, con lo que se puede concluir que cualquiera de los controladores mencionados en este apartado pueden ser implementados sin exigir mayor potencia eléctrica al motor.

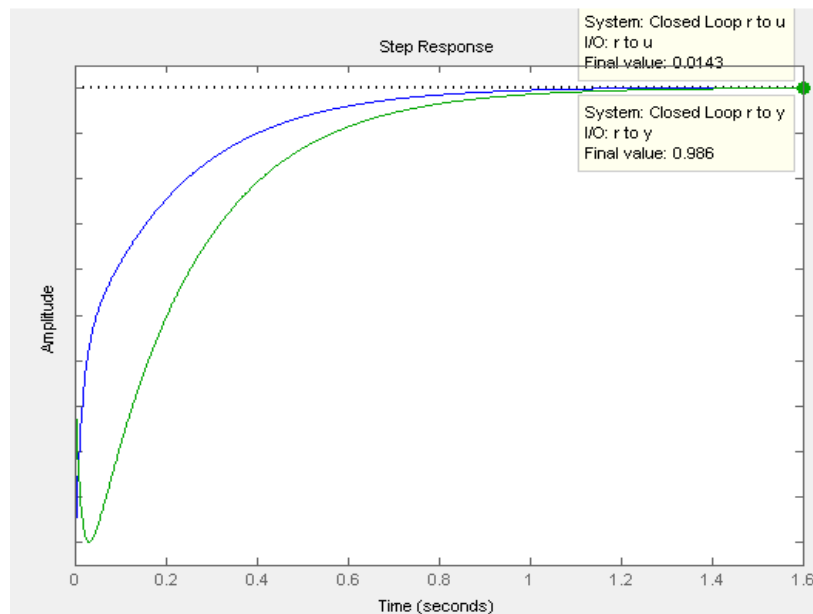


Figura 7.26. Respuesta del sistema con un controlador Klag.

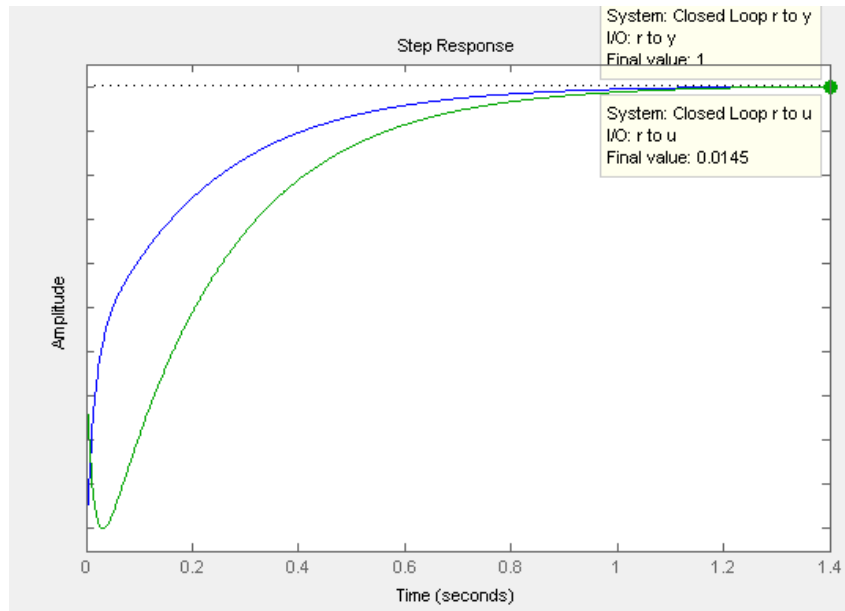


Figura 7.27. Respuesta del sistema con un controlador Proporcional e Integral.

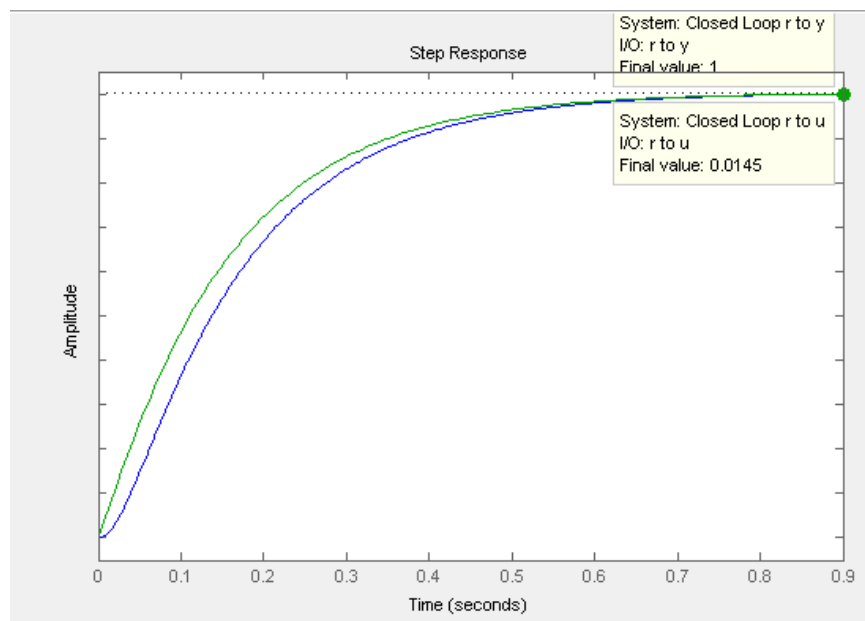


Figura 7.28. Respuesta del sistema con un controlador Integral.

7.5.4. Implementación del controlador.

Los controladores diseñados en el apartado 7.5.3 se realizaron para implementarse en el tiempo continuo, sin embargo el sistema que se está utilizando funciona en tiempo discreto, por lo que se debe discretizar los controladores. Para discretizar se utilizó el método de Tustin con un tiempo de muestreo de 0.001 s, para resolver esta transformada se utilizó el software Matlab. El resultado se muestra en las Ecuaciones 7.14, 7.15 y 7.16, aunque se muestran con exponentes positivos, en la implementación se debe cambiar por su equivalente con exponentes negativos, esto debido a que se busca que el controlador sea causal (reaccione a eventos pasados y no a eventos futuros).

$$Klag(z) = \frac{0.01005 z - 0.00995}{z - 0.9999} \quad 7.14$$

$$PI(z) = \frac{0.01005 z - 0.00995}{z - 1} \quad 7.15$$

$$I(z) = \frac{0.00004 z + 0.00004}{z - 1} \quad 7.16$$

Sin embargo para poder ser programados en el microcontrolador se transformó las ecuaciones del plano z a ecuaciones de diferencia, por lo tanto para ser implementado el controlador se tiene que expresar de la siguiente forma: la Ecuación 7.14 que corresponde al compensador de atraso, se transforma en las Ecuaciones de diferencia 7.17 y 7.18; la Ecuación 7.15 que corresponde al controlador PI se transforma en las Ecuaciones de diferencia 7.19 y 7.20; y la Ecuación 7.16 que corresponde al controlador I se transforma en las Ecuaciones de diferencia 7.21 y 7.22.

$$h(k) = u(k) + 0.9999 h(k - 1) \quad 7.17$$

$$y(k) = 0.01005 h(k) - 0.00995 h(k - 1) \quad 7.18$$

$$h(k) = u(k) + h(k - 1) \quad 7.19$$

$$y(k) = 0.01005 h(k) - 0.00995 h(k - 1) \quad 7.20$$

$$h(k) = u(k) + h(k - 1) \quad 7.21$$

$$y(k) = 0.00004 h(k) + 0.00004 h(k - 1) \quad 7.22$$

Donde:

- $u(k)$ es la señal de error que entra al controlador.
- $h(k)$ es el valor que determina el estado actual del sistema.
- $h(k-1)$ es el estado del sistema en el muestreo anterior.
- $y(k)$ es la salida del controlador en voltios, que se procesa y se transforma en PWM.

Se realizó las pruebas del funcionamiento de los controladores, para la cual se utilizó el motor con reducción de 75:1 y para medir la velocidad angular se utilizó un tacómetro láser digital, la prueba se realizó con los motores en vacío. En la Tabla 7.7 se muestran los resultados obtenidos.

Tabla 7.7. Prueba de los controladores.

<i>Controlador</i>	<i>Tensión de alimentación</i> (V)	<i>Referencia</i> <i>a</i> (RPM)	<i>Ciclo de trabajo</i> (%)	<i>Tensión en el motor</i> (V)	<i>Velocidad del motor</i> (RPM)	<i>Error</i> (%)
<i>Klag</i>	3.3	100	55-56	1.41	92.4	7.6
<i>PI</i>	3.3	100	59-60	1.52	99.5	0.5
<i>I</i>	3.3	100	59-60	1.51	99.4	0.6
<i>Klag</i>	10.10	100	14-15	1.43	97.5	2.5
<i>PI</i>	10.10	100	15	1.46	99.6	0.4
<i>I</i>	10.10	100	15	1.47	99.5	0.5
<i>Klag</i>	10.09	400	61-63	6.20	389.1	2.725
<i>PI</i>	10.09	400	62-64	6.27	396.4	0.9
<i>I</i>	10.09	400	61-62	6.10	396.2	0.95

Se hicieron tres pruebas, las dos primeras tenían la misma velocidad de referencia pero a distintas tensiones de alimentación para los motores, se logró observar cómo el ciclo de trabajo cambió para mantener la velocidad de referencia y la tensión en el motor fue similar sin importar la tensión de alimentación; en la penúltima y última prueba, se mantuvo constante la tensión de alimentación pero se cambió la velocidad de referencia de 100 RPM a 400 RPM. Como se puede observar en la Tabla 7.7 el error es menor a 1% en la mayoría de los casos, el controlador que sobrepasa ese 1% es el Klag, sin embargo todos los controladores cumplen con la condición de diseño ($\text{error} < 5\%$). Las causas del error se deben a la medición de la velocidad con el tacómetro, ya que para realizar la medida se debía colocar un adhesivo reflectante en el eje del motor y el láser del tacómetro debía estar perpendicular a este adhesivo, lo anterior puede provocar error humano en la medición.

Para la implementación del ROBOBO 2.0 se utilizaran los controladores PI e I, debido a que fueron los que generaron menos error en estado estacionario. Solamente se tomó como condición determinante el error en estado estacionario, debido a que el tiempo de subida no es una variable que afecte el funcionamiento del ROBOBO 2.0, en el peor de los casos puede durar dos segundos en alcanzar la velocidad máxima, pero en las pruebas realizadas, lo lograba en menos de un segundo.

7.6. Funcionamiento del Bluetooth.

El módulo Bluetooth se comunica con el PIC32MX534F064H por medio del puerto UART, por lo que en este apartado se va mencionar sobre el desarrollo de la comunicación serial y de los comandos utilizados.

La configuración del puerto UART así como las funciones para recibir, enviar y atender a las interrupciones, me fueron facilitadas por el grupo de trabajo del GII, esto debido a que en otras ocasiones ellos ya habían utilizado el puerto serial UART (no para comunicarse con un módulo Bluetooth), por lo que esta programación ya estaba depurada y su funcionalidad estaba asegurada. Por lo tanto lo que se desarrolló en esta parte fue la puesta en marcha del Bluetooth, integrarlo al resto de dispositivos del ROBOBO 2.0 y se crearon algunas funciones propias del módulo (como los comandos AT).

El puerto UART se configuró para que funcionara a 9600 b/s, con paquetes de datos de 8 bits sin bit de paridad y con un bit de parada. Las funciones que me facilitaron se estructuraron para que los datos fueran administrados en listas (para transmitir y otra para recibir), donde cada vez que recibía una interrupción se verificaba si era causado para transmitir o para recibir, y a partir de los datos en las listas así hacía lo solicitado (guardar si se estaba recibiendo y enviar si se estaba transmitiendo); además se contaba con funciones para enviar datos, el cual consistía en almacenar los datos en la lista dedicada a transmitir; la función para obtener datos, donde simplemente retornaba los valores dentro de la lista dedicada a recibir; adicionalmente contaba con funciones para obtener un dato específico de las listas y para descartar datos en la lista, sin embargo esas dos últimas no fueron utilizadas hasta el momento en la implementación del ROBOBO 2.0.

Las funciones que se realizaron adicionales a las ya explicadas fueron: una dedicada a la verificación de la conexión con el módulo Bluetooth, por medio de los comandos AT, en el Apéndice A.3: se explica en qué consisten estos comandos; y una función que limpie una arreglo de caracteres, la cual recibe como entrada el puntero que direcciona la variable correspondiente y con esto se limpia todas direcciones de memoria asociadas a este puntero.

Para las pruebas de funcionamiento del ROBOBO 2.0 se definió unos comandos que serían enviados desde el Smartphone, el objetivo es que cada motor se pueda mover independientemente, incluso los motores de la rueda, ya que de esta forma se puede realizar giros, entre otras maniobras; se puede escoger tres opciones para detener el motor, una definiendo cuantos grados se desea que gire el motor (se decidió trabajar con grados en lugar de radianes debido a que permite manejar números enteros, lo que disminuye el tiempo de cálculo en el microcontrolador), otra por medio de definir cuanto tiempo se desea que este encendido y la última condición es que nunca se detenga.

Los comandos implementados fueron los siguientes:

- TILT(RPM,t/p/c);
- PAN(RPM,t/p/c);
- WHEEL(RPM1, t/p/c, RPM2. t/p/c);
- STOP(T/P/W1/W2)

Donde:

- El encabezado hace referencia al motor que se desea mover.

- RPM = es la velocidad a la que se desea mover.
- t/p/c = es la condición de parada, por ejemplo: t2000, se detendrá en 2 segundos; p90, se detendrá al girar 90°; y c, que significa que no se detenga.
- En el caso del comando STOP, el parámetro que recibe es el motor que se desea detener, T=TILT, P=PAN, W1=rueda1 y W2=rueda2.

7.7. Sistema Pan/Tilt.

El sistema Pan/Tilt recibió este nombre debido a los movimientos que realiza el Smartphone, similar al de las cámaras de grabación, Pan se refiere al movimiento rotatorio sobre el plano horizontal de la base, y Tilt que en español significa inclinación, se refiere al movimiento de inclinación o cabeceo del Smartphone.

En la Figura 7.29 se muestra la apariencia del sistema Pan/Tilt, el cual será controlado utilizando dos motores, uno de reducción 1000:1 para el tilt, se eligió de esta reducción debido a que por el modo de funcionamiento no se necesita que el movimiento sea rápido, sin embargo sí debe ser capaz de soportar el torque generado por el peso del Smartphone aun cuando no se le esté aplicando tensión al motor; para el Tilt se utilizó un motor de reducción 150:1, en este caso se necesita que sea rápido debido a que la relación de transmisión entre el engrane mayor y el engrane utilizado en el motor es de aproximadamente 14.

La vista explosionada que se muestra en la Figura 7.30 aparecen los componentes básicos del sistema Pan/Tilt, los cuales son:

1. Soporte para el Smartphone, donde también lleva el engrane que mueve el Tilt.
2. Estructura de soporte del Tilt, en esta pieza se sujetan los componentes del Tilt, además funciona como engrane para el sistema Pan.
3. Insertos roscados para colocar los tornillos que sujetan los componentes.
4. Motor de reducción 1000:1.
5. Cubierta para el engrane del motor, evitando que los dientes del engrane entre en contacto con cables u otros componentes.
6. Tapa que cubre todo el sistema Tilt.

7. Motor de reducción 75:1.
8. Tapa que cubre el sistema Pan, la cual se sujeta con la carcasa del ROBOBO 2.0.

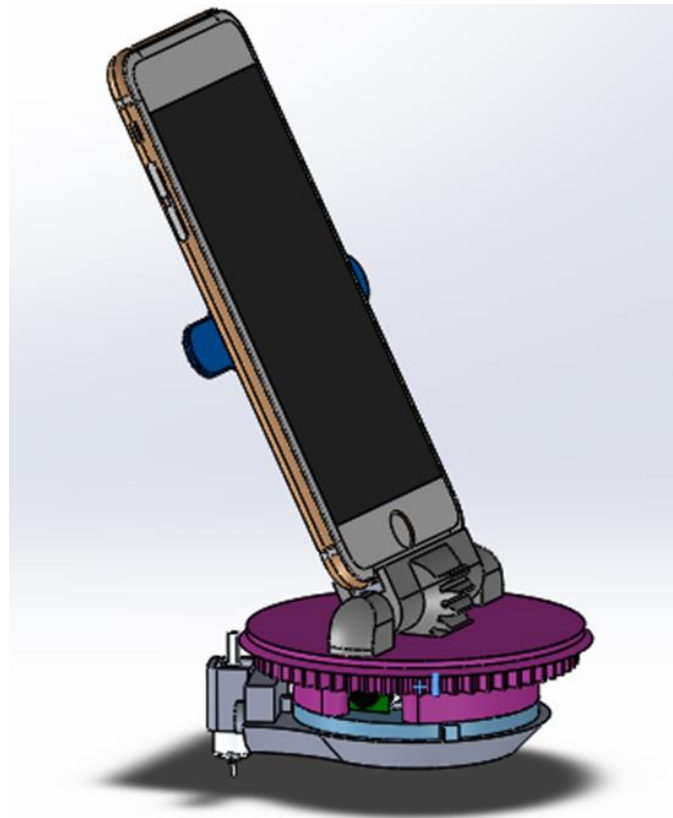


Figura 7.29. Sistema Pan/Tilt.

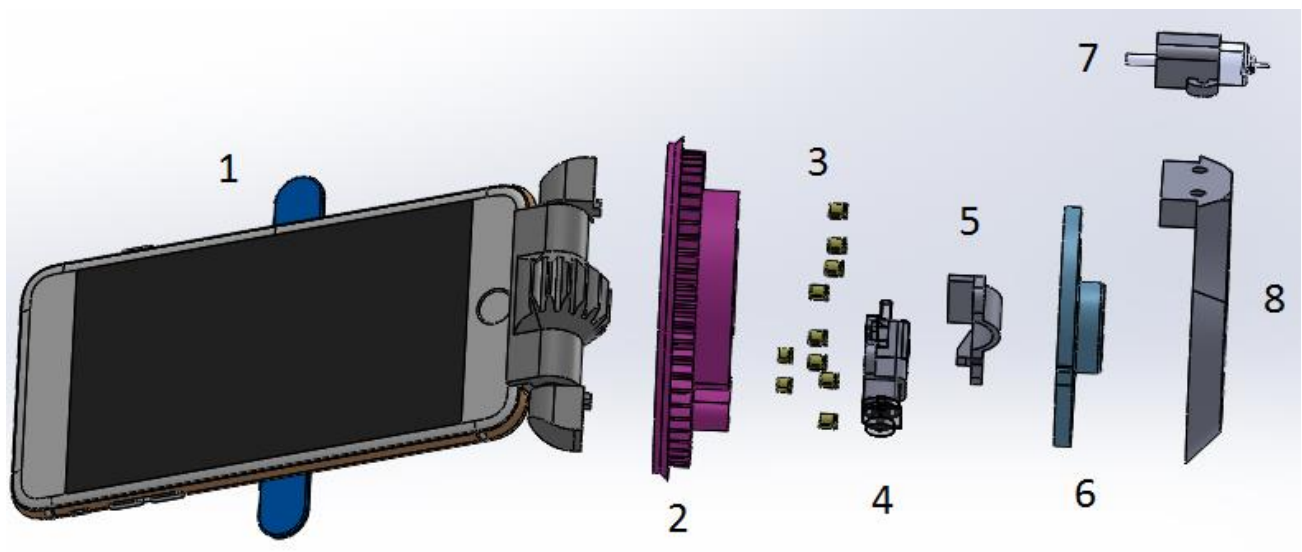


Figura 7.30. Vista explosionada del sistema Pan/Tilt.

El movimiento Pan es originado por la rotación del disco que aparece en la Figura 7.30 como la pieza número 2, se utiliza un motor de reducción 150:1, dispuesto de la forma como aparece en la Figura 7.31, cabe resaltar que en esta figura no aparece el piñón que va en el eje del motor, el motor se colocó en esta posición debido a que el espacio dentro de ROBOBO 2.0 es reducido, así nos permite aprovechar el espacio al máximo. En la Figura 7.32 se aprecia que la pieza número 2 está construida como una rueda dentada, para permitir que el giro sea aproximado a 360°. El sistema Tilt está compuesto por un motor de reducción 1000:1, ya que este es el que genera más torque que las otras opciones disponibles, en la Figura B.5 aparecen las especificaciones técnicas de los motores disponibles en el mercado. Como se puede apreciar en la Figura 7.32, hay una cubierta sobre el piñón del motor, esta pieza adicional se agregó debido a que en las pruebas realizadas mientras el Tilt giraba, el cable del encoder del motor 1000:1 fue prensado por el piñón, provocando la ruptura del cable. En la Figura 7.33 se muestra el corte transversal del sistema Pan/Tilt, en la cual se logra apreciar el espacio reducido dentro de la cavidad del encapsulado del Tilt, además permite apreciar el funcionamiento del sistema Tilt, cabe resaltar que en esta imagen no aparece el piñón que va en el eje del motor.

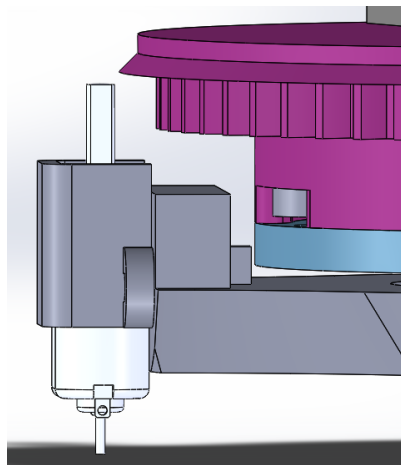


Figura 7.31. Modo de operación del Pan.

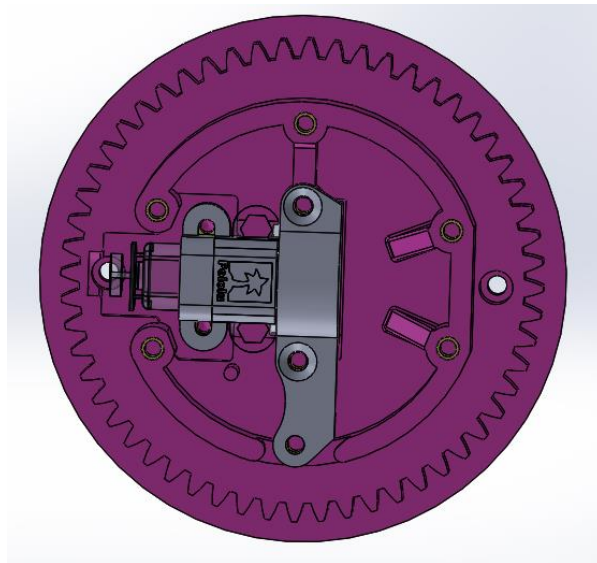


Figura 7.32. Modo de operación del Tilt.

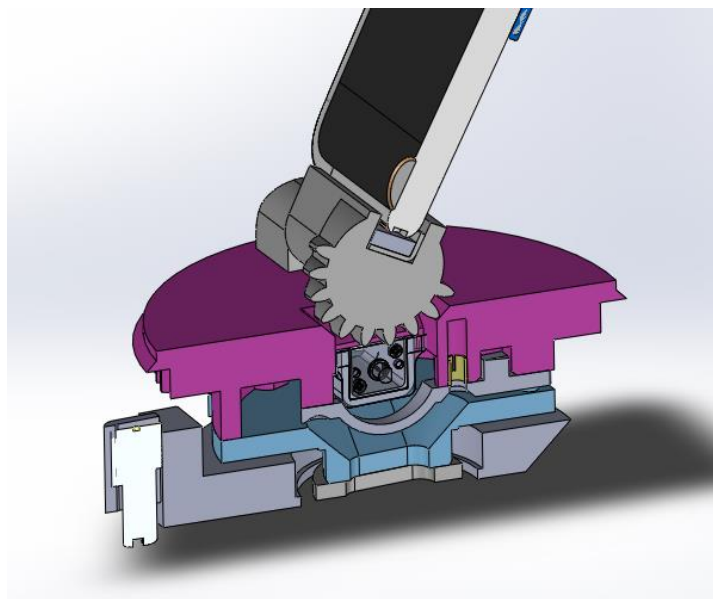


Figura 7.33. Corte transversal del sistema Pan/Tilt.

Para la programación del sistema Pan/Tilt se realizó similar a la ya existente de los motores de las ruedas, solamente se hizo variaciones en las constantes del encoder (debido a que las reducciones eran distintas), y en la referencia de posición. Esto último se refiere a que en el caso de los motores de las ruedas, la posición es relativa y se administra a partir de cambios de posición; mientras que

en el sistema Pan/Tilt la posición es absoluta, esto significa que los valores de posición angular es la medición definida desde un punto de referencia fijo dentro de la carcasa del ROBOBO 2.0. El controlador utilizado es el mismo que en los motores, debido a que el comportamiento del motor es el mismo (mismo tiempo de respuesta, misma velocidad si la caja reductora).

Uno de los inconvenientes encontrados en la implementación de este sistema, fue la definición de la referencia, debido a que los encoders usados (solo un canal del encoder) no permiten detectar cambios de dirección y además si alguien mueve manualmente el sistema, estando el ROBOBO 2.0 apagado, la referencia se perdería y los cálculos posteriores estarían erróneos, por lo que se decidió realizar una función de autoajuste, la cual permite al sistema llegar a una posición definida, para lograr esto se aprovechó de los datos del controlador, primero se colocaron limitadores de giro, de tal forma que al llegar a una posición el sistema dejara de girar, esto provocaría que el controlador aumentara la tensión de los motores, definiendo un umbral permisible de la tensión en los motores y verificando que el motor no está girando durante un periodo de tiempo definido, se concluye que se llegó al final de carrera por consiguiente a la posición de 0° .

7.8. Funcionamiento del driver para LED TLC5941.

Los LED en el ROBOBO 2.0 se utilizarán como una interfaz a bajo nivel de algunas funcionalidades básicas (indicador de encendido, batería cargada, conexión con el Smartphone) así como para indicar estados especiales definidos en el alto nivel (expresar emociones, indicadores luminosos de funciones programadas desde el Smartphone, entre otras). Se estima que el ROBOBO 2.0 tendrá aproximadamente 21 LED.

Este apartado fue el que menos tiempo se le dedicó en la actualidad no está finalizado del todo, debido a que para su finalización se requiere que tanto el alto como el bajo nivel tengan definido todas sus funcionalidades.

Las pruebas realizadas en este apartado solamente consistieron en encender, apagar y cambiar de color los LED, controlados a partir del driver TLC5941 y por medio del protocolo de comunicación SPI.

7.9. Firmware del ROBOBO 2.0.

Las rutinas descritas en apartados anteriores del ROBOBO 2.0, se implementan en su mayoría como interrupciones del sistema, en la Figura 7.34 se muestra la lógica del programa principal, donde la función primordial de este programa es realizar la configuración inicial y esperar la orden del Smartphone. Una vez que recibe una orden, por medio de funciones que se ejecutan en otro archivo del programa, se cambian las configuraciones iniciales, mientras que el programa principal regresa a su estado natural de esperar la orden del Smartphone. Caso excepcionales sacan al microcontrolador de este estado, como lo son las interrupciones, las interrupciones que se programaron son las siguientes: el tiempo de muestreo del controlador, IC de los encoders, una base de tiempo de 1 ms que se implementó para llevar el conteo en el plano temporal de los distintos eventos (tiempo encendido de los motores, envíos periódicos de información al Smartphone, entre otros), interrupción por protocolo de comunicación.

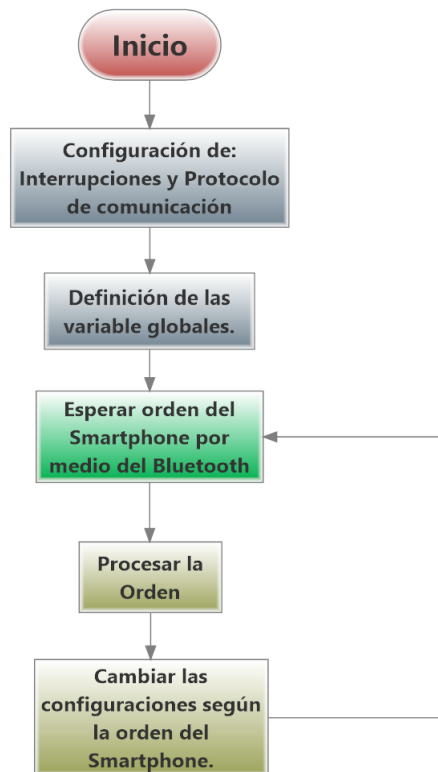


Figura 7.34. Diagrama de flujo en el programa principal.

En la Figura 7.35 se muestra la apariencia física deseada del ROBOBO 2.0, la cual está diseñada para poder ser impresa en 3D, cabe recalcar que todavía está en fase de desarrollo por lo que el producto final puede diferir con respecto al de la figura.

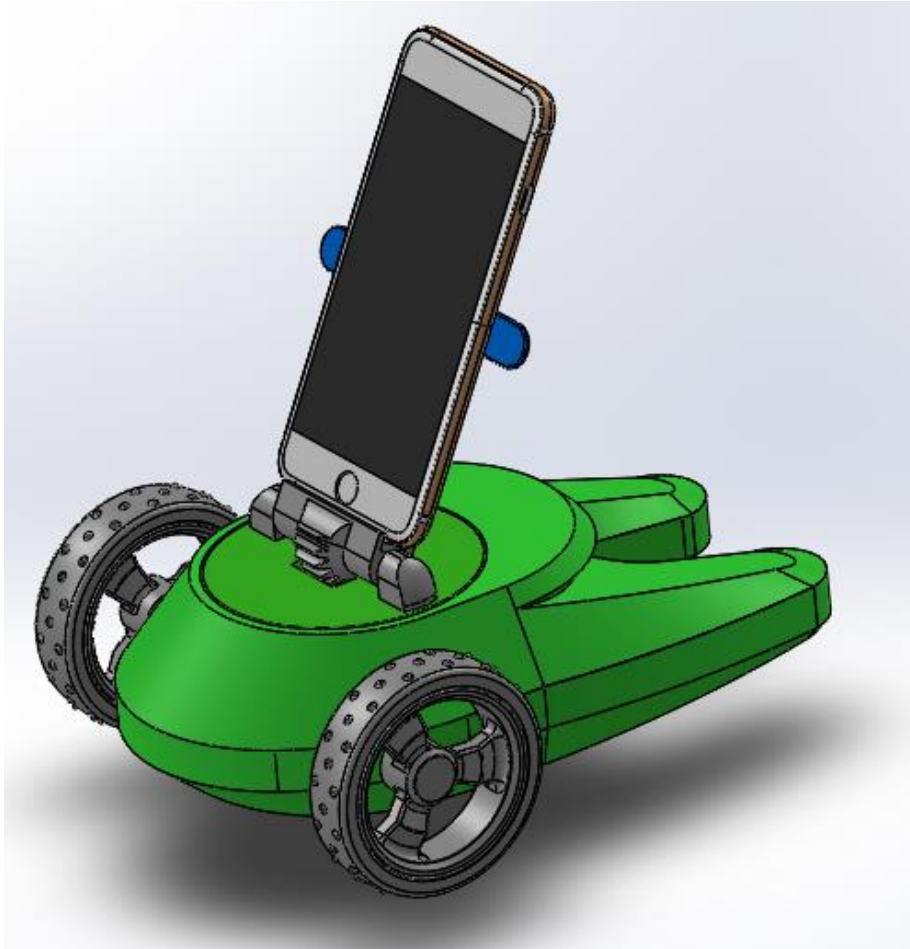


Figura 7.35. Aspecto físico del ROBOBO 2.0.

Capítulo 8: Conclusiones.

Se seleccionaron nuevas soluciones de componentes electrónicos y mecánicos respecto a la primera versión de ROBOBO, esto se evidencia en el cambio de los motores a paso por motores CD, mejoraron la velocidad y potencia mecánica del ROBOBO; el encoder permite tener precisión de movimiento y junto con el controlador permite movimientos suaves y con alta confiabilidad; el driver DRV8833 permite limitar la corriente de los motores, lo que protege los circuitos de la parte de potencia; todos los driver (para motor y para LED) cuentan con detección de errores, lo que le permite al ROBOBO 2.0 protegerse de daños; el sistema Pan/Tilt mejoró el rango de movimiento del ROBOBO y permitió que la interacción con humanos fuera dinámica; la comunicación por Bluetooth permite que el Smartphone pueda tener aplicaciones que no necesiten que se coloque sobre la plataforma, mejorando la experiencia de usuario.

El módulo de comunicación se logró crear, logrando la comunicación entre un Smartphone con sistema operativo Android y la plataforma del ROBOBO 2.0 por medio de Bluetooth; el sistema de comunicación consume solamente 8 mA una vez que se realiza el emparejamiento sin importar si está o no transfiriendo datos; la plataforma del ROBOBO 2.0 fue capaz de reconocer las funciones enviadas desde el Smartphone y realizar la tarea solicitada.

Aunque no fue necesario la creación del PCB, se logró realizar pruebas que validaron el diseño mecánico y electrónico, se realizaron los esquemáticos de las conexiones del sistema para la futura fabricación del PCB.

Se reacomodaron los componentes mecánicos para mejorar las prestaciones del sistema y aprovechar al máximo el espacio disponible.

El diseño del módulo con el movimiento Pan/Tilt permitió el movimiento del Smartphone independiente del movimiento de la plataforma, los motores utilizados en el sistema Pan/Tilt permiten tener alto torque y bajas velocidades para que el movimiento sea suave y preciso, se logró realizar una secuencia de autoajuste del sistema Pan/Tilt para definir la referencia del sistema, se implementó la interfaz de usuario básica compuesta por LED RGB.

Capítulo 9: Recomendaciones.

- Utilizar el driver DRV8833 en modo *Slow decay*, debido a que en este modo el motor tiene mayor potencia, ya que no se descarga completamente para el próximo ciclo de trabajo del PWM.
- Utilizar los motores de reducción de 150:1 en las ruedas, debido a que con estas se gana potencia mecánica aunque disminuya la velocidad de desplazamiento, sin embargo para las aplicaciones deseadas, la velocidad no es un factor determinante.
- Repetir las pruebas del controlador, pero esta vez con todos los sistemas de ROBOBO 2.0, para tener una estimación del peso del sistema más cercana al modo de operación en la aplicación final.
- Aumentar la velocidad de comunicación UART entre el microcontrolador y el módulo Bluetooth, debido a que al ser más rápida la transmisión de dato, la lista de espera para envíos o recepción se transmite más rápido, evitando que la interrupción del puerto UART entre en conflicto con las demás interrupciones del microcontrolador.
- Cambiar los comandos utilizados para controlar los motores por medio del Bluetooth, por unos más sencillos de implementar y con el consenso de los encargados de la programación a alto nivel.
- Verificar los límites de tensión permisible y el tiempo sin que los motores giren para el autoajuste del sistema Pan/Tilt, evitando exceder la carga mecánica del motor y de la estructura del bloqueo de giro.
- Colocar los límites de giro en posiciones que se semejen al movimiento natural de los sistemas Pan/Tilt, con esto el rango de visión del Smartphone no se ve afectado por los límites de giro.
- Realizar pruebas para definir los rangos de los colores con los que se quiere trabajar en los LED RGB, y así delimitar el funcionamiento de los LED y normalizarlo.
- El procesamiento de las ordenes que recibe la plataforma proveniente del Smartphone, realizarlas como máquinas de estado de esta forma se puede optimizar la ejecución del programa.

Bibliografía

- [1] AIBRAIN, I. T. (2015). *TYCHE: The Ultimate Family Companion*. Recuperado el 20 de Enero de 2016, de <http://aibrain.com/products/tyche/>
- [2] APPLE. (2015). *Romotive Romo: Your Robot Friend*. Recuperado el 20 de Enero de 2016, de <http://www.apple.com/au/shop/product/HE516LL/A/romotive-romo-your-robot-friend>
- [3] Arkin, R. (1998). *BEHAVIOR-BASED ROBOTICS*. Cambridge, Massachusetts, U.S.A.: MIT Press.
- [4] Barrientos Cruz, A. (2007). *Fundamentos de Robótica* (Segunda ed.). McGraw-Hill.
- [5] Bluetooth. (2016). *Bluetooth*. Recuperado el 10 de Febrero de 2016, de <https://www.bluetooth.com/>
- [6] DREAM. (2015). *Deferred Restructuring of Experience in Autonomous Machines*. Recuperado el 6 de Julio de 2015, de <http://www.robotsthatdream.eu/>
- [7] e-puck. (2015). Recuperado el 20 de Enero de 2016, de e-puck education robot: <http://www.e-puck.org/>
- [8] GCtronic. (2015). *GCtronic*. Recuperado el 20 de Enero de 2016, de <http://www.gctronic.com/>
- [9] GII. (2014a). *Presentación*. Recuperado el 6 de Julio de 2015, de Grupo Integrado de Ingeniería: <http://www.gii.udc.es/presentacion>
- [10] GII. (2014b). *Robótica y Cognición*. Recuperado el 6 de Julio de 2015, de Grupo Integrado de Ingeniería: <http://www.gii.udc.es/investigacion/detalle/5>
- [11] GII. (2014c). *DREAM: Deferred Restructuring of Experience in Autonomous Machines*. Recuperado el 6 de Julio de 2015, de Grupo Integrado de Ingeniería: <http://www.gii.udc.es/proyectos/detalle/342>
- [12] GII. (2015). Adopt a robot initiative. *ROBOBO*. Ferrol, A Coruña, España.
- [13] Microchip. (2015). *Microchip's 32-bit Microcontrollers*. Recuperado el 15 de Setiembre de 2015, de <http://www.microchip.com/pagehandler/en-us/family/32bit/>

- [14] MiKroEleKtroniKa. (2015). *MINI-32 Board*. Recuperado el 20 de Setiembre de 2015, de <http://www.mikroe.com/mini/pic32/>
- [15] Naya, M. (2013). *Proyecto Final de Carrera*. Ferrol, A Coruña.
- [16] Network, Q. D. (2015). *Snapdragon Micro Rover*. Recuperado el 20 de Enero de 2016, de <https://developer.qualcomm.com/hardware/micro-rover>
- [17] Ogata, K. (2010). *Ingeniería de Control Moderno* (5 ed.). (Pearson, Ed.) Madrid: Prentice Hall.
- [18] OLIMEX. (2016). *BLUETOOTH-SERIAL-HC-06*. Recuperado el 10 de Febrero de 2016, de <https://www.olimex.com/Products/Components/RF/BLUETOOTH-SERIAL-HC-06/>
- [19] Org, t. (2015). Recuperado el 20 de Enero de 2016, de thymio: <https://www.thymio.org/es:thymio>
- [20] overdriverobotics. (2015). *SmartBot*. Recuperado el 20 de Enero de 2016, de <http://www.overdriverobotics.com/SmartBot/>
- [21] POLOLU. (2015). *Robotics & Electronic*. Recuperado el 22 de Setiembre de 2015, de <https://www.pololu.com/>
- [22] RoadNarrows. (2015). *WHEELED*. Recuperado el 20 de Enero de 2016, de RoadNarrows Store: Robotics and Intelligent: <http://www.roadnarrows-store.com/products/robots/wheeled/wheelphone-robot.html>
- [23] Robots, L. M. (2015). Recuperado el 20 de Enero de 2016, de CCSR: <http://letsmakerobots.com/node/42397>
- [24] Springer. (2008). *HandBook of Robotics*. (B. Siciliano, & O. Khatib, Edits.) Berlin: Springer.
- [25] Texas Instruments. (2015). *Stepper Drivers with FETs*. Recuperado el 10 de Enero de 2016, de Texas Instruments products: <http://www.ti.com/product/DRV8833/technicaldocuments?dcmp=dsproject&hqs=td&#doctype2>

- [26] Texas Instruments. (2015). *TLC5941*. Recuperado el 10 de Febrero de 2016, de Texas Instruments products: <http://www.ti.com/product/tlc5941>
- [27] Valdes, F., & Pallás, R. (2007). *Microcontroladores: Fundamentos y aplicaciones con PIC*. (MARCOMBO, & ALFAOMEGA, Edits.) México.
- [28] Wildi, T. (2007). *Máquinas eléctricas y sistemas de potencia* (Sexta ed.). México: PEARSON EDUCATION.

Apéndices.

Apéndice A.1: Glosario, abreviaturas y simbologías.

Actuador: Dispositivo mecánico, eléctrico, neumático u hidráulico que modifica o actúa sobre un sistema.

ADC: Siglas en inglés de *Analog to Digital Converter*.

Bluetooth: Comunicación inalámbrica que intercambia datos utilizando transmisión de radio.

CISC: Siglas en inglés de *Complex Instruction Set Computer*.

CPU: Siglas en inglés de *Central Processing Unit*.

DREAM: *Deferred Restructuring of Experience in Autonomous Machines*, proyecto de investigación financiado por la Unión Europea.

Encoder: Dispositivo que convierte la posición angular en una señal eléctrica analógica o digital.

Estator: Componente fijo que genera campo magnético en el motor.

GII: Grupo Integrado de Ingeniería, grupo de investigación de la Universidade da Coruña.

LED: Siglas en inglés de *Light Emitting Diode*.

Microcontrolador: Microcomputadora realizado en un circuito integrado.

Motor CD: Dispositivo que transforma la energía eléctrica proveniente de una fuente de corriente continua en energía mecánica.

Pan: Movimiento de rotación sobre un plano horizontal, utilizado ampliamente en aplicaciones con cámaras de video.

PCB: siglas en inglés de *Printed Circuit Board*.

PWM: Siglas en inglés de *Pulse Width Modulation*.

RISC: Siglas en inglés de *Reduced Instruction Set Computer*.

ROBOBO: Robot autónomo móvil desarrollado por el GII.

Robot: Agente artificial mecánico guiado por un programa de control o circuito electrónico que posee sensores que le proporcionan información de su entorno y actuadores que le permiten modificar dicho entorno.

Robótica: Ciencia que estudia la conexión inteligente entre la percepción y la acción.

Rotor: Componente giratorio de un motor.

RPM: Revoluciones por minuto.

Sensor: Dispositivo que transforma las variables físicas en señales eléctricas para ser interpretadas y/o procesadas.

SPI: Siglas en inglés de *Serial Peripheral Interface*, módulo serial síncrono utilizado para la comunicación entre periféricos y otros microcontroladores.

Tilt: Movimiento de inclinación sobre un plano vertical, utilizado ampliamente en aplicaciones con cámaras de video.

UART: Siglas en inglés de *Universal Asynchronous Receiver Transmitter*, Puerto serial para comunicación bidireccional.

UDC: Universidade da coruña.

Apéndice A.2: Resultados del PWM.

Para ejecutar el PWM se realizaron distintas pruebas, con el objetivo de explotar las posibilidades del controlador para utilizar los puertos OC, la primera parte de las pruebas consistió en un solo PWM que se le variaba el ciclo de trabajo y las frecuencia de operación, el objetivo principal en esta parte era comprobar si los resultados eran congruentes con la configuración realizada.

Los resultados se pueden observar en la Figura A.1 se asignó la frecuencia de 50kHz y el ciclo de trabajo de 50%, como se puede observar en la medición realizada por el osciloscopio, existe un error de 0.6% en la frecuencia de operación, esta se pudo ocasionar por el ruido generado en la señal.



Figura A.1. PWM a 50% de ciclo de trabajo y a 50 kHz.

En la Figura A.2, se cambió tanto el ciclo de trabajo como la frecuencia, por los valores de 60% y 100kHz respectivamente. En esta prueba al igual que en la anterior existe un error de 0.6% en la frecuencia de operación pero además esta vez apareció un error de 0.017% en el ciclo de trabajo, esto se pudo originar por el ruido de la señal, pero aun así el error no es significativo ya que está por debajo del 1%.

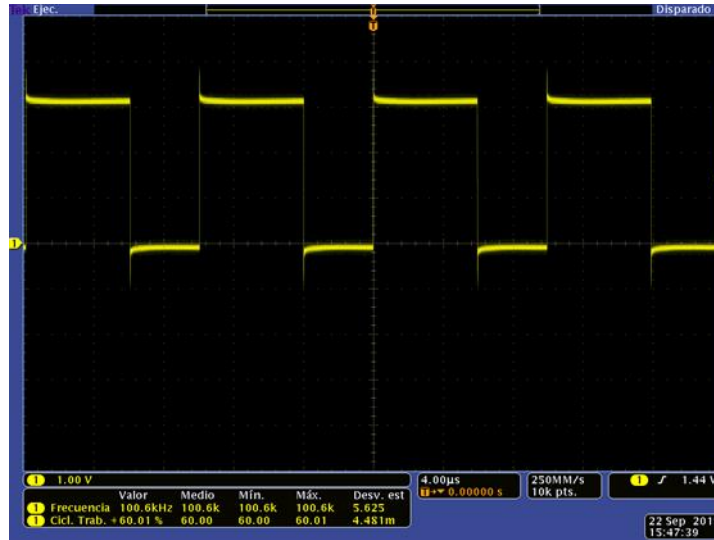


Figura A.2. PWM a 60% de ciclo de trabajo y a 100 kHz.

En las Figura A.3 y en la Figura A.4 se mantuvo constante la frecuencia de operación pero se cambió el ciclo de trabajo con el objetivo de registrar el error que se tenía al trabajar con el mínimo y el máximo ciclo de trabajo (1% y 99% respectivamente). Como se puede observar en ambas imágenes el error en la frecuencia es de 0.6% al igual que en las pruebas anteriores, sin embargo el error para el ciclo de trabajo de 99% es de 0.03% y para el ciclo de trabajo de 1% y es de 0.7%. Esta diferencia se debe a que a menor ciclo de trabajo la medición puede verse afectada en mayor medida por el ruido, sin embargo en ambos caso es menor al 1% por lo que la configuración y la secuencia programada en el microcontrolador cumplen con los requisitos.

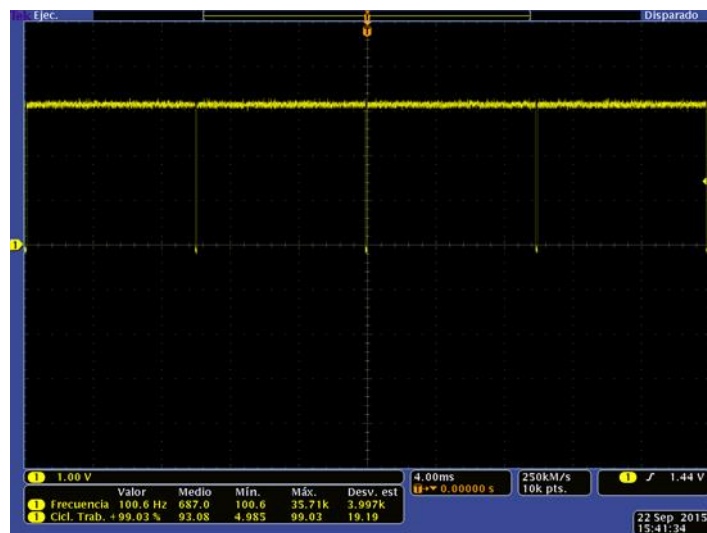


Figura A.3. PWM a 99% de ciclo de trabajo y a 100 kHz.

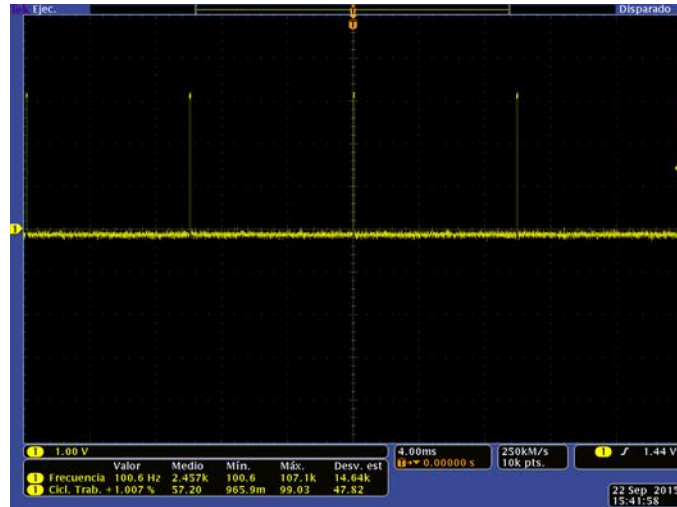


Figura A.4. PWM a 1% de ciclo de trabajo y a 100 kHz.

En las siguientes pruebas se implementó dos PWM funcionando simultáneamente debido a que eso se acerca al funcionamiento final (cuatro PWM ejecutándose al mismo tiempo). En estas pruebas se utilizó el mismo temporizador (Timer2) para ambos PWM, por lo que ambos funcionarán a la misma frecuencia, en las Figura A.5 se puede observar que el error en la frecuencia sigue siendo del 0.6% al igual que en las pruebas anteriores, además en este caso en particular el error en el ciclo de trabajo en ambos PWM es de 0%. En la Figura A.6 se cambió la frecuencia de operación por 1 kHz y se puso ambos ciclos de trabajo con un valor de 50%, en esta prueba se obtuvo que el error de la frecuencia de operación fue de 0.8% mientras que para el ciclo de trabajo fue de 0.2% en ambos casos.



Figura A.5. Dos PWM a 100 kHz y con ciclo de trabajo de 40% y 60%.



Figura A.6. Dos PWM a 1 kHz y con ambos ciclo de trabajo de 50%.

En las pruebas que se muestran en la Figura A.7 y en la Figura A.8, en ambas se utilizó la misma frecuencia de operación de 10kHz y en ambas el error fue de 0.6%. Para la prueba de la Figura A.7, los ciclos de trabajo fueron de 40% y 60% en ambos casos con error de 0%. En la prueba de la Figura A.8 los ciclos de trabajo fueron ambos de 74% sin embargo en este caso el error fue de 0.014%.



Figura A.7. Dos PWM a 10 kHz y con ciclo de trabajo de 40% y 60%.

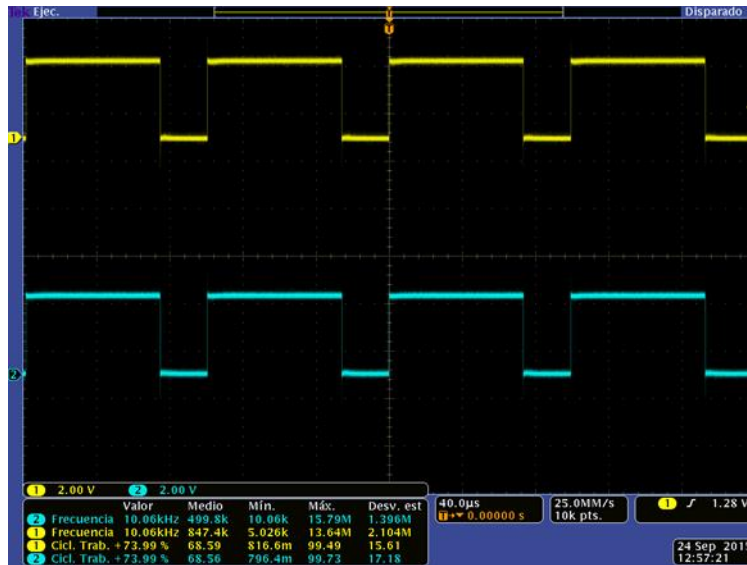


Figura A.8. Dos PWM a 100 kHz, ambos con ciclo de trabajo de 74%.

Apéndice A.3: Comandos AT.

Los comandos AT son los que se utilizan para configurar el módulo Bluetooth por medio del puerto serial UART, cabe resaltar que para que los comandos AT puedan ser utilizados, es necesario que el módulo Bluetooth no esté emparejado con ningún dispositivo. Los parámetros por defecto del HC-06 son: velocidad de transmisión de 9600 b/s, paquete de datos de 8 bit, sin bit de paridad, con un bit de parada, ID de HC-06 y contraseña: 1234. Los comandos para configuración AT son los siguientes:

- Prueba de comunicación:

Enviar: AT

Respuesta: OK

- Definir la velocidad de transmisión:

Enviar: AT+BAUD1

Respuesta: OK1200

- Definir el nombre del módulo Bluetooth:

Enviar: AT+NAMExxxxx (escribir el nombre deseado).

Respuesta: OKname.

- Cambiar la contraseña:

Enviar: AT+PINxxxx

Respuesta: OKsetpin.

Apéndice A.4: Esquemáticos y PCB.

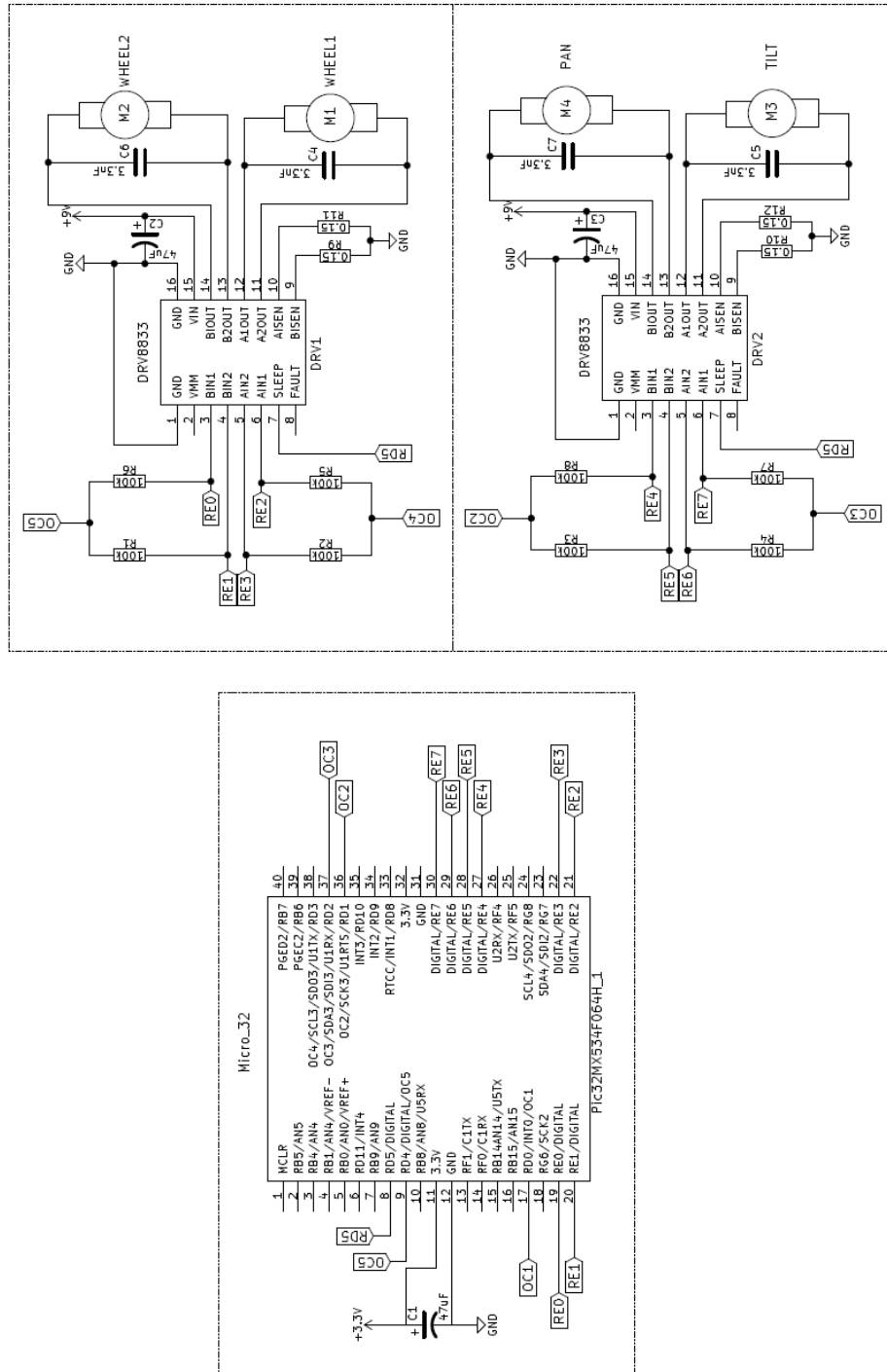


Figura A.9. Esquemático utilizado para las pruebas con los motores, utilizando solo el PWM.

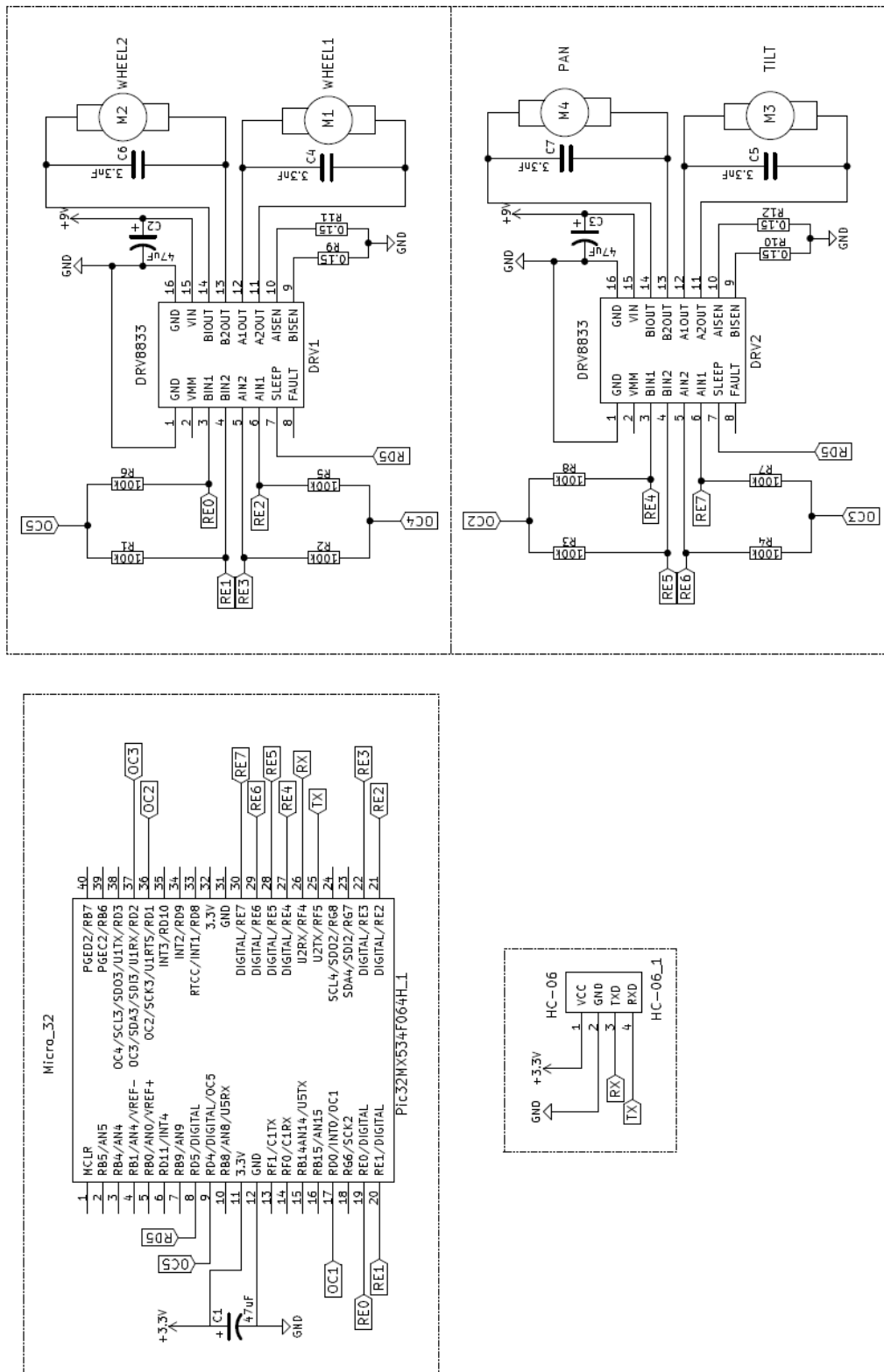


Figura A.10. Esquemático utilizado en las pruebas de los motores, controlados por Bluetooth.

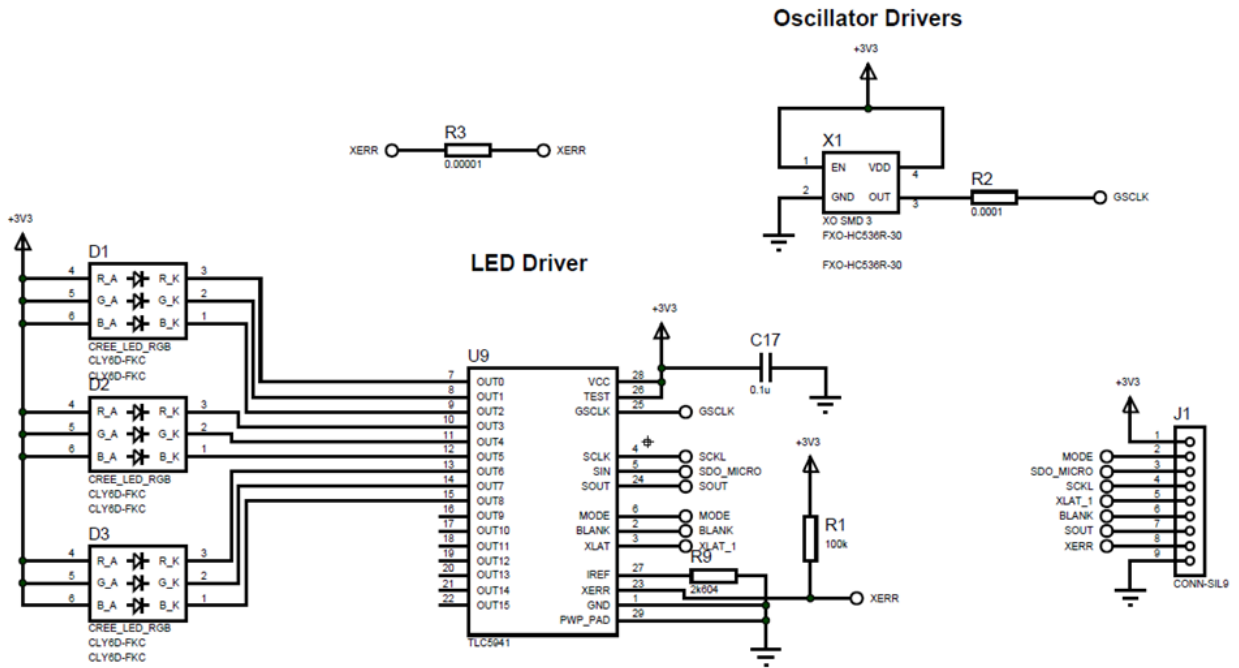


Figura A.11. Esquemático utilizado en las pruebas del driver TLC5941.

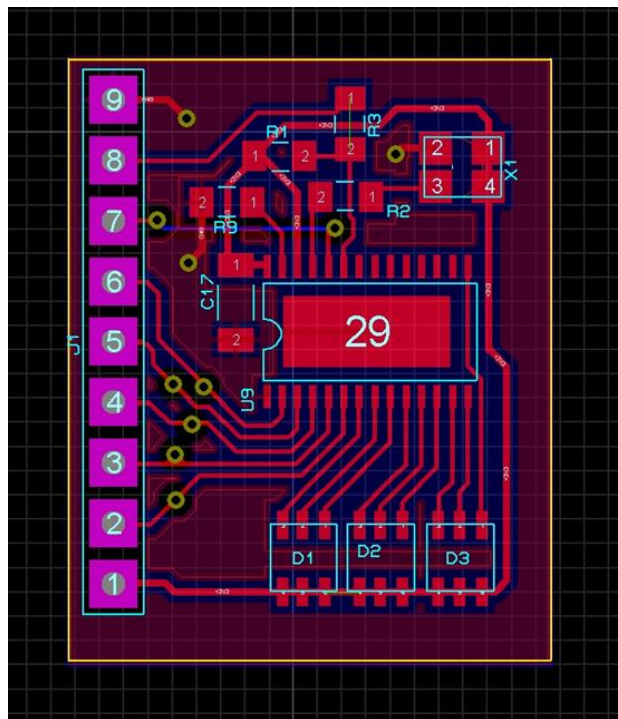


Figura A.12. PCB fabricada para las pruebas de los LED.

Anexos

Anexo B.1: Datos técnicos de los componentes utilizados.



PIC32MX5XX/6XX/7XX

32-bit Microcontrollers (up to 512 KB Flash and 128 KB SRAM) with Graphics Interface, USB, CAN, and Ethernet

Operating Conditions

- 2.3V to 3.6V, -40°C to +105°C, DC to 80 MHz

Core: 80 MHz/105 DMIPS MIPS32® M4K®

- MIPS16e® mode for up to 40% smaller code size
- Code-efficient (C and Assembly) architecture
- Single-cycle (MAC) 32x16 and two-cycle 32x32 multiply

Clock Management

- 0.9% internal oscillator
- Programmable PLLs and oscillator clock sources
- Fail-Safe Clock Monitor (FSCM)
- Independent Watchdog Timer
- Fast wake-up and start-up

Power Management

- Low-power management modes (Sleep and Idle)
- Integrated Power-on Reset, Brown-out Reset
- 0.5 mA/MHz dynamic current (typical)
- 41 µA IPD current (typical)

Graphics Features

- External graphics interface with up to 34 Parallel Master Port (PMP) pins:
 - Interface to external graphics controller
 - Capable of driving LCD directly with DMA and internal or external memory

Analog Features

- ADC Module:
 - 10-bit 1 Msps rate with one Sample and Hold (S&H)
 - 16 analog inputs
 - Can operate during Sleep mode
- Flexible and independent ADC trigger sources
- Comparators:
 - Two dual-input Comparator modules
 - Programmable references with 32 voltage points

Timers/Output Compare/Input Capture

- Five General Purpose Timers:
 - Five 16-bit and up to two 32-bit Timers/Counters
- Five Output Compare (OC) modules
- Five Input Capture (IC) modules
- Real-Time Clock and Calendar (RTCC) module

Communication Interfaces

- USB 2.0-compliant Full-Speed OTG controller
- 10/100 Mbps Ethernet MAC with MII and RMII interface
- CAN module:
 - 2.0B Active with DeviceNet™ addressing support
- Six UART modules (20 Mbps):
 - Supports LIN 1.2 protocols and IrDA® support
- Up to four 4-wire SPI modules (25 Mbps)
- Up to five I²C modules (up to 1 Mbaud) with SMBus support
- Parallel Master Port (PMP)

Direct Memory Access (DMA)

- Up to eight channels of hardware DMA with automatic data size detection
- 32-bit Programmable Cyclic Redundancy Check (CRC)
- Six additional channels dedicated to USB, Ethernet and CAN modules

Input/Output

- 15 mA or 10 mA source/sink for standard V_{OH}/V_{OL} and up to 22 mA for non-standard V_{OH1}
- 5V-tolerant pins
- Selectable open drain and pull-ups
- External interrupts

Qualification and Class B Support

- AEC-Q100 REV G (Grade 2 -40°C to +105°C) planned
- Class B Safety Library, IEC 60730

Debugger Development Support

- In-circuit and in-application programming
- 4-wire MIPS® Enhanced JTAG interface
- Unlimited program and six complex data breakpoints
- IEEE 1149.2-compatible (JTAG) boundary scan

Packages

Type	QFN	TQFP	TFBGA	VTLA
Pin Count	64	64	100	100
I/O Pins (up to)	51	51	83	83
Contact/Lead Pitch	0.50	0.50	0.40	0.50
Dimensions	9x9x0.9	10x10x1	12x12x1	14x14x1
				10x10x1.1
				9x9x0.9

Note: All dimensions are in millimeters (mm) unless specified.

Figura B.1. Información técnica del PIC32MX534F064H. (Microchip, 2015)



DRV8833

SLVSAR1E – JANUARY 2011 – REVISED JULY 2015

DRV8833 Dual H-Bridge Motor Driver

1 Features

- Dual-H-Bridge Current-Control Motor Driver
 - Can Drive Two DC Motors or One Stepper Motor
 - Low MOSFET ON-Resistance: HS + LS 360 mΩ
- Output Current (at $V_M = 5\text{ V}$, 25°C)
 - 1.5-A RMS, 2-A Peak per H-Bridge in PWP and RTY Package Options
 - 500-mA RMS, 2-A Peak per H-Bridge in PW Package Option
- Outputs can be in Parallel for
 - 3-A RMS, 4-A Peak (PWP and RTY)
 - 1-A RMS, 4-A Peak (PW)
- Wide Power Supply Voltage Range: 2.7 to 10.8 V
- PWM Winding Current Regulation and Current Limiting
- Thermally Enhanced Surface-Mount Packages

2 Applications

- Battery-Powered Toys
- POS Printers
- Video Security Cameras
- Office Automation Machines
- Gaming Machines
- Robotics

3 Description

The DRV8833 device provides a dual bridge motor driver solution for toys, printers, and other mechatronic applications.

The device has two H-bridge drivers, and can drive two DC brush motors, a bipolar stepper motor, solenoids, or other inductive loads.

The output driver block of each H-bridge consists of N-channel power MOSFETs configured as an H-bridge to drive the motor windings. Each H-bridge includes circuitry to regulate or limit the winding current.

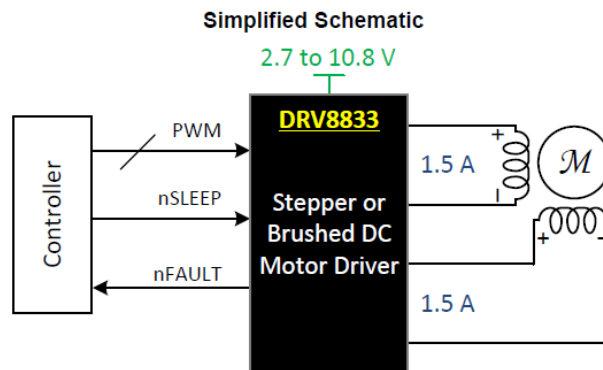
Internal shutdown functions with a fault output pin are provided for overcurrent protection, short-circuit protection, undervoltage lockout, and overtemperature. A low-power sleep mode is also provided.

The DRV8833 is packaged in a 16-pin WQFN package with PowerPAD™ (Eco-friendly: RoHS & no Sb/Br).

Device Information⁽¹⁾

PART NUMBER	PACKAGE	BODY SIZE (NOM)
DRV8833	TSSOP (16)	5.00 mm × 4.40 mm
	HTSSOP (16)	5.00 mm × 4.40 mm
	WQFN (16)	4.00 mm × 4.00 mm

(1) For all available packages, see the orderable addendum at the end of the data sheet.




 An IMPORTANT NOTICE at the end of this data sheet addresses availability, warranty, changes, use in safety-critical applications, intellectual property matters and other important disclaimers. PRODUCTION DATA.

Figura B.2. Información técnica del driver DRV8833. (Texas Instruments, 2015).

2. Feature

- Wireless transceiver
 - Sensitivity (Bit error rate) can reach -80dBm.
 - The change range of output's power: -4 - +6dBm.
- Function description (perfect Bluetooth solution)
 - Has an EDR module; and the change range of modulation depth: 2Mbps - 3Mbps.
 - Has a build-in 2.4GHz antenna; user needn't test antenna.
 - Has the external 8Mbit FLASH
 - Can work at the low voltage (3.1V~4.2V). The current in pairing is in the range of 30~40mA. The current in communication is 8mA.
 - Standard HCI Port (UART or USB)
 - USB Protocol: Full Speed USB1.1, Compliant With 2.0
 - This module can be used in the SMD.
 - It's made through RoHS process.
 - The board PIN is half hole size.
 - Has a 2.4GHz digital wireless transceiver.
 - Bases at CSR BC04 Bluetooth technology.
 - Has the function of adaptive frequency hopping.
 - Small (27mm×13mm×2mm)
 - Peripherals circuit is simple.
 - It's at the Bluetooth class 2 power level.
 - Storage temperature range: -40 °C - 85°C, work temperature range: -25 °C - +75°C
 - Any wave inter Interference: 2.4MHz, the power of emitting: 3 dBm.
 - Bit error rate: 0. Only the signal decays at the transmission link, bit error may be produced. For example, when RS232 or TTL is being processed, some signals may decay.
- Low power consumption
- Has high-performance wireless transceiver system
- Low Cost

www.wavesen.com Phone: 020-84083341 Fax: 020-84332079 QQ:1043073574
 Address: Room 527, No.13, Jiangong Road, Tianhe software park, Tianhe district, Guangzhou Post: 510660
 Technology consultant: support@wavesen.com Business consultant: sales@wavesen.com
 Complaint and suggestion: sunbirdit@hotmail.com

Figura B.3. Información técnica del módulo Bluetooth HC-06. (OLIMEX, 2016).



TLC5941

SLVS589D–JULY 2005–REVISED JANUARY 2008

16-CHANNEL LED DRIVER WITH DOT CORRECTION AND GRAYSCALE PWM CONTROL

FEATURES

- 16 Channels
- 12-Bit (4096 Steps) Grayscale PWM Control
- Dot Correction
 - 6 Bit (64 Steps)
- Drive Capability (Constant-Current Sink)
 - 0 mA to 80 mA
- LED Power Supply Voltage up to 17 V
- $V_{CC} = 3.0\text{ V to }5.5\text{ V}$
- Serial Data Interface
- Controlled In-Rush Current
- 30-MHz Data Transfer Rate
- CMOS Level I/O
- Error Information
 - LOD: LED Open Detection
 - TEF: Thermal Error Flag

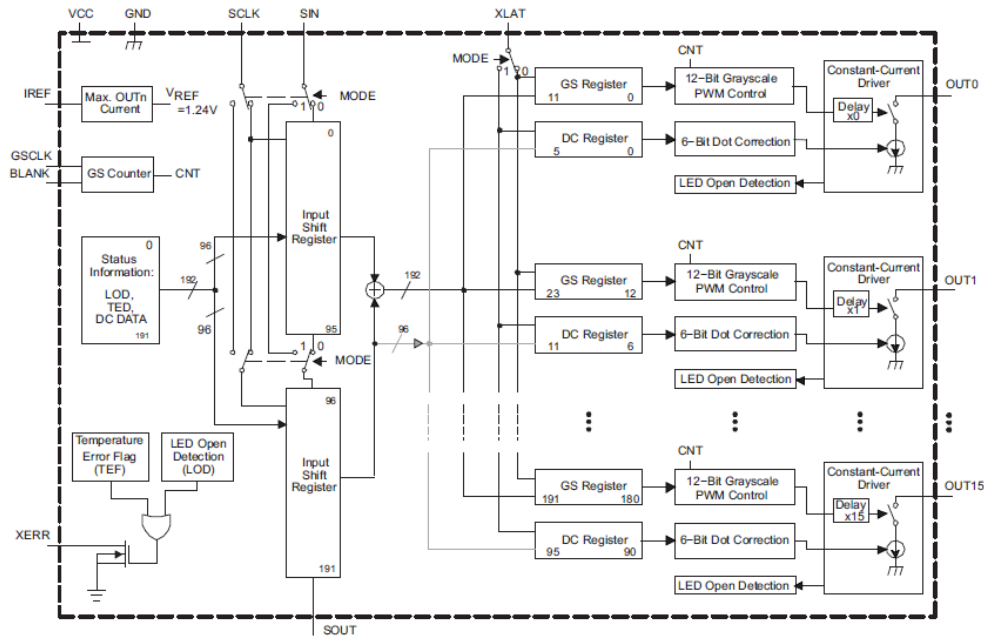
APPLICATIONS


- Monocolor, Multicolor, Full-Color LED Displays
- LED Signboards
- Display Back-Lighting

DESCRIPTION

The TLC5941 is a 16-channel, constant-current sink, LED driver. Each channel has an individually adjustable 4096-step grayscale PWM brightness control and a 64-step constant-current sink (dot correction). The dot correction adjusts the brightness variations between LED channels and other LED drivers. Both grayscale control and dot correction are accessible via a serial interface. A single external resistor sets the maximum current value of all 16 channels.

The TLC5941 features two error information circuits. The LED open detection (LOD) indicates a broken or disconnected LED at an output terminal. The thermal error flag (TEF) indicates an overtemperature condition.



 Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet. PowerPAD is a trademark of Texas Instruments.

PRODUCTION DATA information is current as of publication date. Products conform to specifications per the terms of the Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

Copyright © 2005–2008, Texas Instruments Incorporated

Figura B.4. Información técnica del driver TLC5941. (Texas Instruments, 2015).



Motor Type	Stall Current @ 6 V	No-Load Speed @ 6 V	Approximate Stall Torque @ 6 V		
				Single-Shaft (Gearbox Only)	Dual-Shaft (Gearbox & Motor)
high-power, carbon brushes (HPCB)	1600 mA	3000 RPM	4 oz-in	10:1 HPCB	10:1 HPCB dual-shaft
		1000 RPM	9 oz-in	30:1 HPCB	30:1 HPCB dual-shaft
		625 RPM	15 oz-in	50:1 HPCB	50:1 HPCB dual-shaft
		400 RPM	22 oz-in	75:1 HPCB	75:1 HPCB dual-shaft
		320 RPM	30 oz-in	100:1 HPCB	100:1 HPCB dual-shaft
		200 RPM	40 oz-in	150:1 HPCB	150:1 HPCB dual-shaft
		140 RPM	50 oz-in	210:1 HPCB	210:1 HPCB dual-shaft
		120 RPM	60 oz-in	250:1 HPCB	250:1 HPCB dual-shaft
		100 RPM	70 oz-in	298:1 HPCB	298:1 HPCB dual-shaft
		32 RPM	125 oz-in	1000:1 HPCB	1000:1 HPCB dual-shaft
high-power (HP) <i>(same specs as HPCB above)</i>	1600 mA	6000 RPM	2 oz-in	5:1 HP	
		3000 RPM	4 oz-in	10:1 HP	10:1 HP dual-shaft
		1000 RPM	9 oz-in	30:1 HP	30:1 HP dual-shaft
		625 RPM	15 oz-in	50:1 HP	50:1 HP dual-shaft
		400 RPM	22 oz-in	75:1 HP	75:1 HP dual-shaft
		320 RPM	30 oz-in	100:1 HP	100:1 HP dual-shaft
		200 RPM	40 oz-in	150:1 HP	150:1 HP dual-shaft
		140 RPM	50 oz-in	210:1 HP	
		120 RPM	60 oz-in	250:1 HP	
		100 RPM	70 oz-in	298:1 HP	298:1 HP dual-shaft
32 RPM	125 oz-in	1000:1 HP	1000:1 HP dual-shaft		
medium-power (MP)	700 mA	2200 RPM	3 oz-in	10:1 MP	10:1 MP dual-shaft
		730 RPM	8 oz-in	30:1 MP	
		420 RPM	13 oz-in	50:1 MP	
		290 RPM	17 oz-in	75:1 MP	75:1 MP dual-shaft
		220 RPM	19 oz-in	100:1 MP	100:1 MP dual-shaft
		150 RPM	24 oz-in	150:1 MP	
		75 RPM	46 oz-in	298:1 MP	298:1 MP dual-shaft
		25 RPM	80 oz-in	1000:1 MP	1000:1 MP dual-shaft
low-power	360 mA	2500 RPM	1 oz-in	5:1	
		1300 RPM	2 oz-in	10:1	
		440 RPM	4 oz-in	30:1	30:1 dual-shaft
		250 RPM	7 oz-in	50:1	50:1 dual-shaft
		170 RPM	9 oz-in	75:1	
		120 RPM	12 oz-in	100:1	100:1 dual-shaft
		85 RPM	17 oz-in	150:1	
		60 RPM	27 oz-in	210:1	
		50 RPM	32 oz-in	250:1	
		45 RPM	40 oz-in	298:1	298:1 dual-shaft
14 RPM	70 oz-in	1000:1	1000:1 dual-shaft		

Figura B.5. Información técnica de los motores CD. (POLOLU, 2015).

Anexo B.2: Esquemáticos de los componentes utilizados.

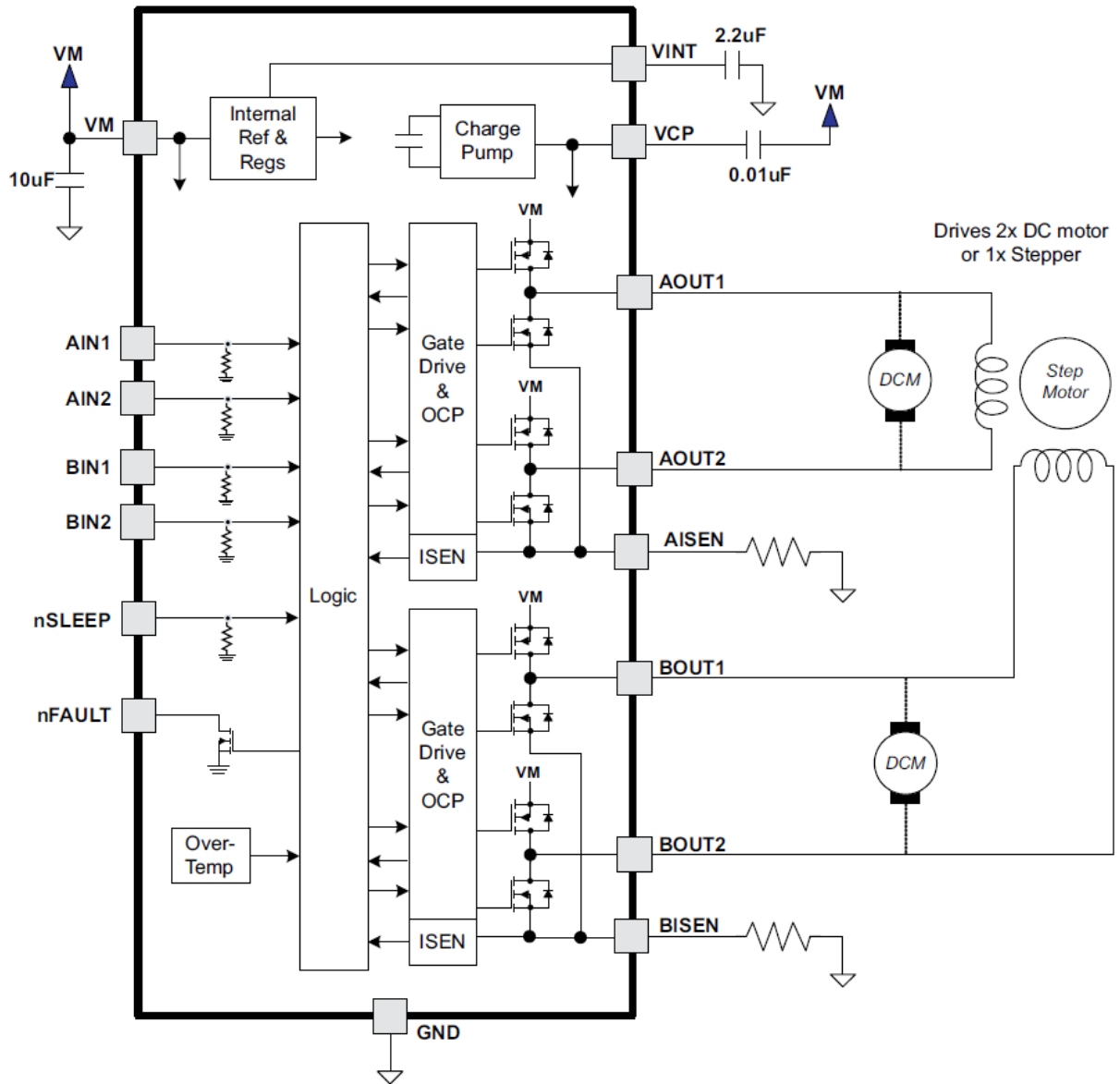


Figura B.6. Esquemático del driver DRV8833. (Texas Instruments, 2015).

Anexo B.3: Planos del garmotor HP.

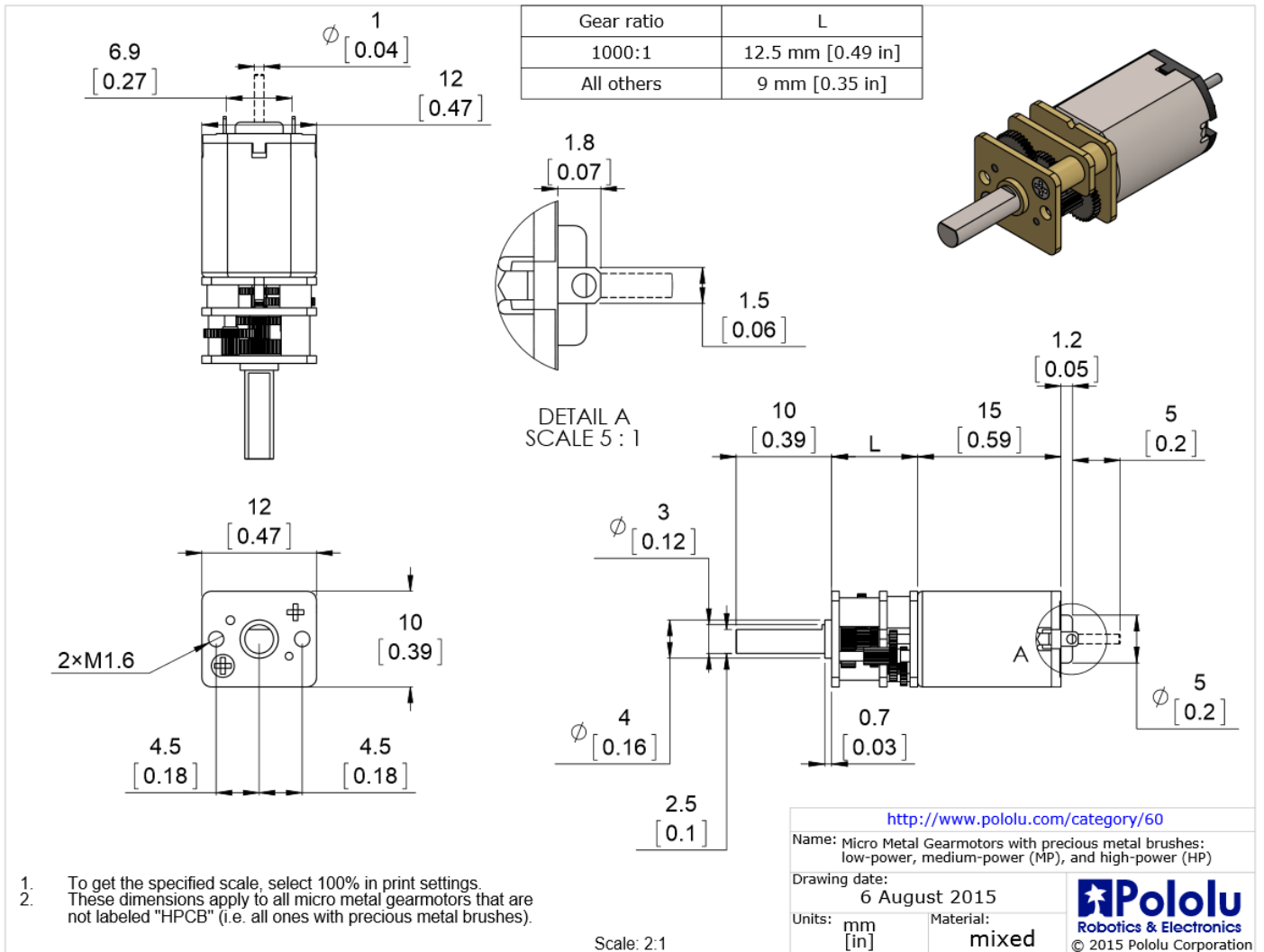


Figura B.9. Planos del garmotor HP. (POLOLU, 2015).