eman ta zabal zazu
Universidad del País Vasco
Euskal Herriko Unibertsitatea
BILBOKO INGENIARITZA ESKOLA
ESCUELA DE INGENIERÍA DE BILBAO

GRADO EN INGENIERÍA EN TECNOLOGÍA INDUSTRIAL

TRABAJO FIN DE GRADO

PLATAFORMA DE BAJO COSTE BASADO EN MATLAB Y RASPBERRY PI PARA EL DESARROLLO DE CONTROLADORES

Alumno: Lorenzo Vidarte, Raul

Director: Artaza Fano, Fernando

Curso: 2017-2018

Fecha: Julio de 2018

- *Resumen*

El propósito de este documento es detallar el diseño e implantación de una plataforma de bajo coste basado en Matlab y Raspberry Pi para que futuros alumnos puedan desarrollar algoritmos de control sobre sistemas diferentes. La plataforma debe ser capa de captar la señales provenientes del sistema *Feedback MS150*, procesarlas y ejecutar el programa de control, generando un señal de control que excite el actuador el sistema y mantenga el sistema en las condiciones deseadas de funcionamiento.

Palabras clave: *Raspberry Pi, Matlab, Control, Sistemas empotrados de bajo coste*

- *Abstract*

The purpose of this document is to detail the design and implementation of a low cost platform based on Matlab and Raspberry Pi so that future students can develop control algorithms on different systems. The platform must be capable of capturing the signals coming from the MS150 Feedback system, processing them and executing the control program, generating a control signal that excites the actuator of the system and maintains the system in the desired operating conditions.

Key Words: *Raspberry Pi, Matlab, Control, Low cost embedded systems*

- *Laburpena*

Hurrengo txostenaren helburua, Matlab eta Raspebry Pi bidez, kostu baxuko plataforma baten diseinu eta ezarpena da. Honen bidez, etortzeko dauden ikasleek algoritmo desberdin bidez, sistema anitz kontrola ahal izango ditutzte. Plataforma honek *Feedback MS150* sistemako seinaleak jaso eta prozesatzeko gaitasuna izan behar du, honen bidez eragilea exzitatu eta gure sistema fisikoa operazio kondizio egokietan mantentzeko.

Hitz Gakoak: *Raspberry Pi, Matlab, Controla, Kostu baxuko sistemak*

ÍNDICE DE CONTENIDO

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS Y GRÁFICOS

ÍNDICE DE GRÁFICOS

LISTA DE ACRÓNIMOS

DDC	Digital Direct Control
PC	Personal Computer
SBC	Single-Board Computer
WBS	Work Breakdown Structure
GPIO	General Purpose Input/Output
SPI	Serial Peripheral Interface
I2C	Inter Integrates Circuits
UART	Universal Asynchronous Receiver/Transmitter
SCK	Serial Clock
MOSI	Master Output Slave Input
MISO	Master Input Slave Output
SS	Select Slave
ADC	Analog Digital Converter
PWM	Pulse-Width Modulation

1. Introducción

Este documento contiene la información recogida tras el diseño, desarrollo e implantación de una plataforma de bajo coste para el desarrollo de algoritmos de control.

En primer lugar, se explica el contexto bajo el cual se realiza el proyecto y las razones por las qué llevarlo a cabo. Posteriormente, se exponen los objetivos para obtener un exitoso proyecto y se valora el alcance que puede tener y cuales son los límites del mismo. A su vez, se estiman los beneficios que ofrece.

A continuación, se detalla el estado del arte que engloba el proyecto, describiendo la situación en la que se encuentra la Raspberry Pi y el método de comunicación utilizado.

El siguiente apartado muestra un visión general del proyecto, con el fin de dar una perspectiva global y permita una comprensión rápida del mismo. Después, se detalla la metodología seguida para el desarrollo de la plataforma, describiendo cada una de las fases tomadas.

Todas las fases se ordenan y se muestran de forma cronológica mediante un diagrama de Gantt en el partido de la planificación. Para el desarrollo de cada fases han sido necesarios unos cálculos que se recogen en el siguiente apartado y van acompañados de los resultados del proyecto.

En lo referente al aspecto económico, este proyecto cuenta con un descargo de gastos de todo el proyecto y muestra el impacto económico de todas las partes implicadas.

Por último, se resumen los aspectos mas relevantes del proyecto en el apartado de conclusiones, con la intención de crear una visión global del mismo. También se adjuntan la bibliografía necesaria para la elaboración del proyecto y anexos.

2. Contexto

Un sistema de Control Digital Directo (DDC Digital Direct Control), es aquel en el que el ordenador es parte fundamental del lazo de control y se comunica con el proceso a controlar mediante la utilización de convertidores A/D y D/A que permiten leer los valores de las variables físicas, los primeros de ellos y actuar sobre los actuadores del proceso a los segundos.

Estos convertidores A/D y D/A, junto con otros elementos, tales como contadores, temporizadores, entradas/salidas digitales, se puede encontrar agrupadas en las denominadas tarjetas de adquisición y control.

Estas tarjetas de adquisición y control se conectan directamente al bus del ordenador. Las ventas de uso de las tarjetas son, la velocidad (debido a que están conectadas directamente al bus) y el coste.

FIGURA 2.1 TARJETAS DE ADQUISICIÓN DE DATOS

Limitándonos, únicamente al uso del ordenador personal como elemento de control, el mercado ha ofrecido una gran variedad de estas tarjetas, pudiendo sus características proporcionadas por las tarjetas pueden variar dependiendo del, numero y tipo de entradas (tensión termopar, digital), salidas, velocidad y otras funciones previstas.

Una de las características fundamentales a l hora de elegir estas tarjetas era el tipo de bus disponible en el ordenador al que se podía conectar. A lo largo de la historia del PC, estos han utilizado diferentes buses internos, ISA, EISA, Vesa, PCI, PC-e. Además con diferentes versiones de cada uno de ellos. Esto origina que al cambiar el parque de PC's de un laboratorio docente, podrían quedar obsoletas tarjetas de adquisición y control adquiridas anteriormente.

Desde hace unos años han ido apareciendo en el mercado los denominados **computador de placa reducida,** computador de placa única o computador de placa simple (SBC) de bajo costo.

En particular la **Raspberry Pi**. Un mini PC originalmente concebido como una solución orientada a entornos educativos aunque sus posibilidades y prestaciones pronto han logrado convertirla en base de **todo tipo de proyectos** hardware.

El fundamento de este proyecto se basa n utilizar este sistema para realizar el control de un servomotor, sustituyendo aun un ordenador personal dotado con tarjeta de adquisición y control.

3. Objetivos y Alcance

El objetivo principal de este proyecto se enfoca en el **diseño e implantación** del hardware y software de una **plataforma** de bajo coste basado en Raspberry Pi y Matlab para el **desarrollo de controladores**. El objeto del proyecto tiene un fin didáctico para que futuros alumnos aprendan a desarrollar controladores sobre esta herramienta.

La plataforma debe ser capaz de tomar lecturas analógicas del sistema a controlar, convertirlas a digital para el procesamiento y ejecución del algoritmo de control por parte de la Raspberry Pi y generar la señal pertinente de control para obtener el funcionamiento deseado del sistema.

Para lograr el objetivo principal se deberán cumplir ciertos objetivos secundarios o actividades prefijadas al inicio del proyecto los cuales se sintetizan a continuación y se ilustran en el WBS (Work Breakdown Structure):

En primer lugar, se analiza la información sobre el hardware que se tiene a disposición para realizar el proyecto, una Raspberry Pi y un servomotor *Feedback MS150,* con el fin de identificar el hardware necesario para la conexión entre ambos dispositivos.

Una vez estudiada la información, el segundo objetivo es la gestión de componentes necesarios para llevar a cabo el proyecto. Consistirá en la obtención de cada una de las partes físicas de la plataforma y la descarga de los paquetes software necesarios. Se debe tener en cuenta que una parte del circuito electrónico queda fuera del alcance del proyecto y será un técnico el encargado se llevarlo a cabo.

El siguiente objetivo, crucial para el desarrollo exitoso del proyecto, consiste en el diseño e implantación del hardware para la conexión entre dispositivos y del software capaz desarrollar algoritmo de control. El hardware, por una parte, debe conectar todos los diapositivas y ajustar los valores de tensión y intensidad óptimos para el bueno funcionamiento del conjunto. Por otro lado, el software deberá garantizar la comunicación entre los dispositivos y proporcionar un herramienta de programación para el control.

Por último, se programarán un algoritmo de control para garantizar el buen funcionamiento de la plataforma desarrollada y se evaluarán los resultados obtenidos. El algoritmo consistirá en un control proporcional de ganancia K_c.

Cabe destacar que para llevar a cabo el proyecto será necesaria una buena gestión del mismo y seguimiento del cumplimiento de cada uno de los objetivos.

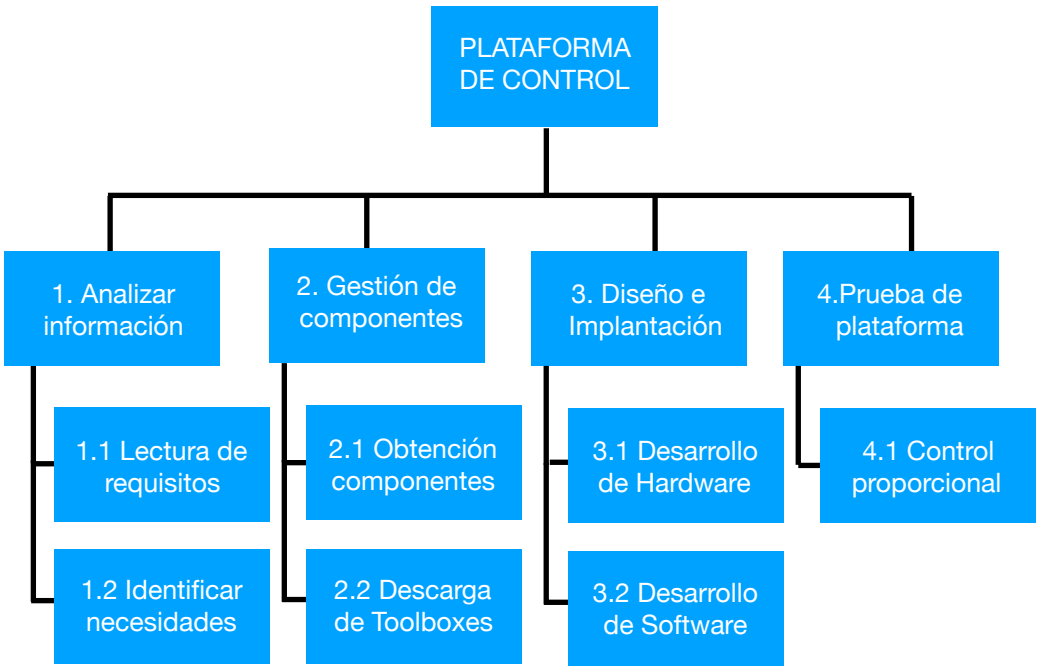


FIGURA 3.1 WBS PLATAFORMA DE CONTROL

4. Beneficios del proyecto

Este proyecto ofrece una gran variedad de beneficios, todos ellos ligados al ámbito del control y de la digitalización. Aunque la cantidad de proyectos de esta índole sea elevado, es el único que pretende relacionar dispositivos que en la actualidad son comúnmente usados como la Raspberry Pi y sistemas como el motor *Feedback MS150* que hasta hace poco se consideraban como obsoletos.

4.1 Beneficios tecnológicos

El beneficio principal que ofrece el proyecto es proporcionar un plataforma o soporte para que futuros usuarios o alumnos puedan desarrollar algoritmos de control sobre la Raspberry Pi y les permita enfocar su proyecto al control de diferentes sistema y dejar de lado la implantación necesaria para la comunicación entre dispositivos.

Además, la plataforma desarrollada es un herramienta de bajo coste totalmente modificable que aporta una base de trabajo a futuros alumnos para que puedan aprender lo desarrollado en el proyecto e incluso mejoren lo realizado.

Hasta ahora, aquellos sistema orientados al uso didáctico existentes en el laboratorio que se habían apartado por la obsolescencia de los ordenadores capaces de incorporar las tarjetas de adquisición de datos, se pueden reutilizar y aprovecharlos. .

Otro de los beneficios de proyecto, es publicar los códigos desarrollados durante el proyecto para que le puedan ser útil a cualquier miembro de la comunidad de Raspberry Pi, ya que ha resultado de gran ayuda disponer del trabajo realizado por la comunidad para la realización del proyecto.

4.2 Beneficios en el participante

Al tratarse un trabajo de fin de grado, el participante encargado del desarrollo del proyecto también es partícipe de algunos de los beneficios. Por un lado, la posibilidad de que el participante amplíe sus conocimientos en el ámbito de la programación, electrónica y control, y por otro, la disponibilidad de las herramientas necesarias para llevar a cabo ese aprendizaje. Cabe destacar, que muchos de los conocimientos adquiridos durante el proyecto no se ven reflejados en el mismo, es por ello, que los beneficios aportados van más allá del proyecto.

5. Estado del arte

En este apartado se analiza la situación de los computadores de placa reducida como es la Raspberry Pi, así como se detallan los desarrollos tecnológicos que ha sufrido a la lo largo de su utilización. Se considera interesante detallar el marco tecnológico que rodea la plataforma escogida para situar el proyecto y las posibles aplicaciones.

La Raspberry Pi es un pequeño ordenador que es muy funcional y que es capaz de ejecutar un sistema operativo como Ubuntu. Es por ello que la Raspberry Pi es la plataforma del proyecto, ya que se encarga de adquirir datos, procesarlos, mostrarlos e incluso generar otros.

5.1 Historia de la Raspberry PI

En el año 2006 el Dr. Eben Upton y sus colaboradores del laboratorio de informática de la Universidad de Cambridge, desarrollaron un equipo barato, pensando en que estuviera al alcance de cualquiera sin preocupación de posibles averías, con el fin de incentivar a los estudiantes en el ámbito de la programación. De esta idea, nació la Raspberry Pi.

Posteriormente, nació la Fundación Raspberry PI, una institución benéfica del Reino Unido que fomenta la alfabetización y el entusiasmo informático entre los niños. A parte de hardware, también se ha creado una plataforma web https://www.raspberrypi.org la cual esta compuesta por foros, preguntas frecuentes, noticias de actualidad y información sobre la placa que ayuda a los usuarios más recientes y a la comunidad en su conjunto.

La gran ventaja que posee este computador son las reducidas dimensiones que posee. Esto se debe al uso de un chip de tipo SoC, siendo una tecnología que fisicamente coloca la memoria, el microprocesador y el procesador de gráficos en una especie de sandwich de silicio provocando la reducción costes y de tamaño. En un principio, aparecieron dos tipo de placas al mercado: el modelo A, destinado para desarrolladores y el modelo B para uso doméstico.

El primer prototipo se monto en una placa con dimensiones idénticas a las de una memoria USB, y disponía de puerto HDMI y de un puerto USB. En agosto de 2011 consiguieron el verdadero hito fabricando 150 placas de los modelos A y B. Debido al gran exento que tuvieron, la fundación decidió fabricar un primer lote de 10.000 placas en Taiwány (China), en vez de fabricarlas en Reino Unido y así poder abaratar costes e invertir en nuevos desarrollos tecnológicos . El 29 de febrero de 2012 se puso a la venta el Modelo B en dos tiendas muy reconocidas “Farrel” y “Rs Components” obteniendo grandes resueltos en la venta de los mismos.

A partir de estos hechos, la fundación ha ido creando nuevos modelos, los cuales mejoran las prestaciones de los anteriores, los cuales se repasaran a continuación.

5.2 Modelos de la Raspberry PI

En este proyecto, el modelo de Raspberry Pi que se ha utilizo es el modelo 3. Es el último del mercado pero aún así en este apartado vamos ilustrar los modelos más relevantes que han aparecido en el mercado con el fin de observar cómo ha evolucionado la tecnología que abarca este computador de placa reducida.

En la siguiente tabla se muestra varios de los modelos de Raspberry Pi:

	Modelo A 	**Modelo B** 	**RPi V2 Modelo B** 	**RPi zero** 
Procesador	ARM1176JFZ-S 700 MHz	ARM1176JFZ-S 700 MHz	ARMv7 900MHz	BCM2835 1Ghz ARM 11
Memoria RAM	256 Mb	512 Mb	1Gb	512 Mb
Dimensiones	85,6 x 53,98 mm	85,6 x 53,98 mm	85 x 56 x 17 mm	65 x 30 x 5 mm
Precio	18,30 €	18,30 €	34,00 €	4,00 €
Conectividad	1x USB 2.0	2x USB 2.0	4x USB 2.0 / Ethernet	2x micro-USB
GPIO	26 pines	26 pines	40 pines	26 pines

TABLA 1. MODELOS DE RASPBERRY PI

5.3 Raspberry Pi 3

En este caso, se va a describen las características del modelo utilizado para la implantación de una plataforma para el desarrollo de controladores y, a continuación, se mostrará el protocolo de comunicación utilizado para la comunicación entre dispositivos.

La novedad que introdujo frente a otro modelos, se debe al nuevo procesador compuesto por 4 núcleos, a 1,2 GHz y soportando procesamiento de 64 bits, además de integrar la conectividad Wifi y la Bluetooth en la misma placa. Este tipo de conectividad permite la programación en remoto sin la necesidad de conectar el cable Ethernet o USB. Las características principales se encuentran en el Anexo 1.

La Raspberry Pi 3 permite tres protocolos de comunicación en serie para la trasferencia de información entre dispositivos mediante los pines GPIO (General Purpose Input/Output): SPI, I2C y UART. En este proyecto, se va a utilizar el protocolo SPI, por ello, a continuación de detalla el bus de comunicaciones SPI.

5.4 El bus de comunicaciones SPI

El método de comunicación SPI (Serial Peripherial Interface) es un método estándar desarrollado por Motorola con el fin de comunicar dispositivos electrónicos. Esto método ha sido el utilizado para conexión de la Raspberry Pi y el convertidor analógico digital. SPI establece una comunicación bidireccional, en modo full dúplex, y configuración Master/Slave. El master es el encargado de iniciar la comunicación y se permite la existencia de varios Slaves. El bus SPI es un bus de 4 líneas, sobre el cual se transmiten paquetes de información de 8 bits. Una de las características de este bus es que cada dispositivo puede actuar como transmisor y receptor al mismo tiempo, por lo tanto una comunicación dúplex.

El bus SPI utiliza un registro de desplazamientos circular y de 8 bits, por lo tanto cuando el Master quiere enviar información usará este registro de desplazamientos y enviara un bit por cada flanco de reloj. A su vez, cada bit enviado por el Master que recibe el esclavo debe de ser almacenado, desplazando el registro. En la siguiente figura se puede apreciar el desplazamiento circular del registro:

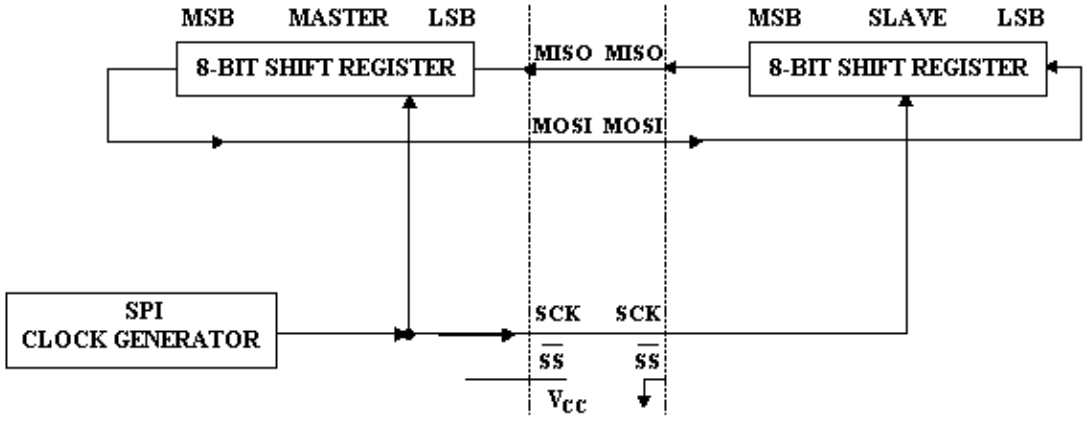


FIGURA 5.1 REGISTRO DE LA COMUNICACIÓN SPI

Como se ha dicho, la comunicación SPI requiere de 4 líneas de conexión entre Maestro y Esclavo. También es necesario la conexión de un masa común para un correcto funcionamiento. Estas líneas son las siguientes:

- **SCK** (Serial Clock): Es la señal de reloj, su función es la sincronización de los diferentes dispositivos que pueda haber conectados en el bus.
- **MOSI (**Master Output Slave Input): Línea encargada de transmitir el flujo de información de Master a Slave.
- **MISO (**Master Input Slave Output): Línea encargada de transmitir el flujo de información de Slave a Master.
- **SS** (Slave Select): Entrada de selección de esclavo.

El inicio de la transferencia de información solo la puede llevar a cabo el Master, ya que es el responsable de la generación de la señal de reloj. Una vez haya configurado la señal de reloj, el Master coloca nivel 0 la línea SS del Slave con el cual desea establecer comunicación. El siguiente paso es que el Master comience a transmitir y recibir bits de datos por cada pulso de reloj, los bits son empujados por la línea MOSI uno a uno en el registro de desplazamiento del esclavo y el mismo proceso pero de forma inversa por la línea MISO. Por lo tanto, el Slave envía por la línea MISO recibe por la línea MOSI, el Master por el contrario envía por la línea MOSI recibe por la línea MISO. Cuando el Maestro ha terminado la comunicación, la señal SS volverá a situación inicial.

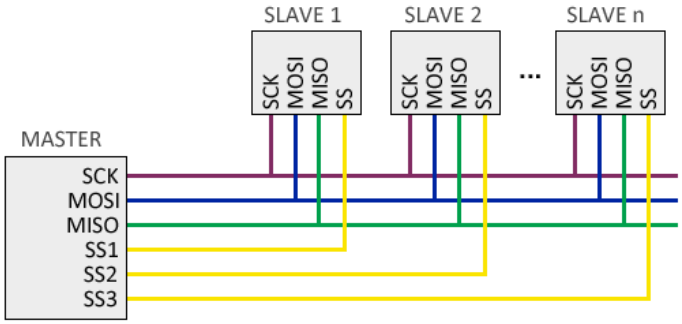


FIGURA 5.2 CONEXIONES MASTER-SLAVE

La Raspberry Pi dispone de 9 puertos GPIO dedicados a la comunicación SPI, concretamente a través de estos pines se establece la comunicación entre los dispositivos Master/Slave. La localización de dicho pines GPIO se encuentra en el Anexo 2.

La utilización del bus de comunicaciones SPI se facilita mucho al utilizar las diferentes librerías SPI para la Raspberry Pi que se pueden encontrar en multitud de sitios webs, desarrollados por la comunidad de Raspberry Pi. Aún así, hay que tener en cuenta que el bus destinado a este tipo de comunicación esta deshabitado por defecto, por ello, será necesaria su activación para una posterior utilización.

6. Diseño de alto nivel

A continuación, se va a proceder a la descripción general del sistema en su conjunto. Se tratará de mostrar la comunicación entre las diferentes partes del sistema, además de concretar lo que realizan cada de ellas. La figura que se muestra describe el flujo de información entre todos los dispositivos:

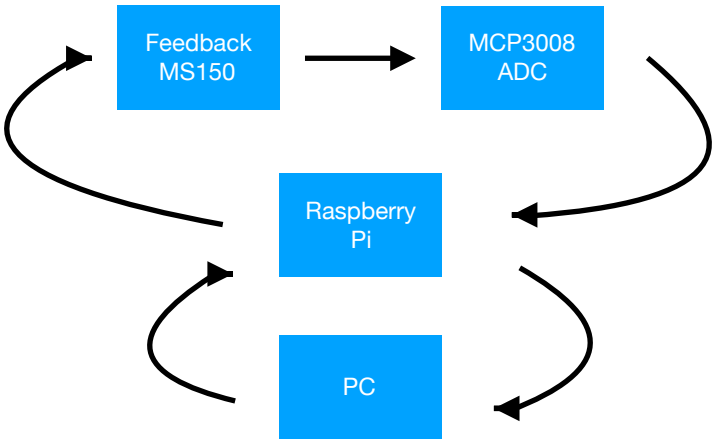


FIGURA 6.1 DIAGRAMA DE ALTO NIVEL

Como punto de partida, se encuentra el sistema **Feedback MS150** que consiste de la etapa de potencia, un motor excitado por inducido, un freno encargado de introducir perturbaciones, un tacómetro y un módulo que permite conocer la posición en la que se encuentra el rotor. Estos dos últimos módulos generan una diferencia de potencial en su bornes en función del valor de la magnitud a medir. Dichas tensiones son las señales de entrada al sistema de control.

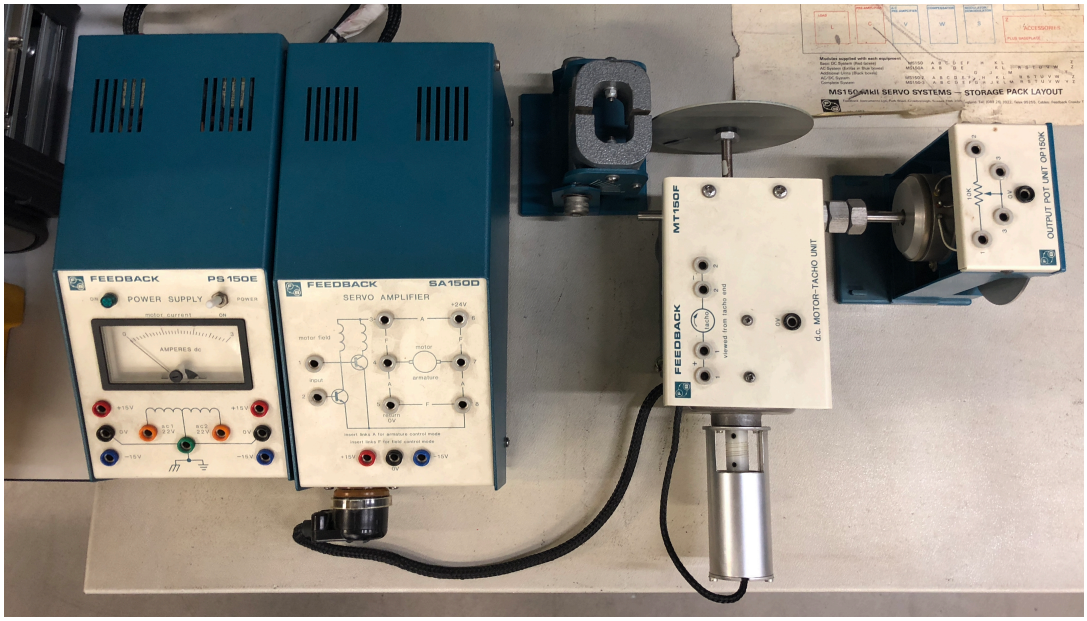


FIGURA 6.2 SISTEMA FEEDBACK MS150

Una de las características más significantes de las señales proporcionadas por los sensores es que son entradas analógicas. Por otro lado, el computador de placa reducida disponible para este proyecto carece de entradas analógicas, por ello, se ha procedido a la incorporación de un convertidor analógico digital llamado **MCP3008 ADC** (Analog Digital Converter) que efectué la conversión de las señales y proporcione una señal digital al computador. Gracias a este conversor de 10 bits y al circuito electrónico de reducción de tensión, permitirá al computador la lectura digital de las señales mediante el lenguaje de comunicación SPI entre ambos dispositivos.

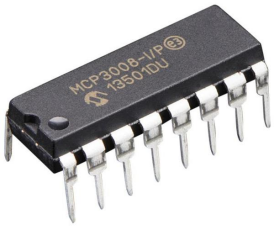


FIGURA 6.3 MCP3008 ADC

En este proyecto, como se ha explicado previamente, el computador de placa reducida utilizada ha sido un **Raspberry Pi 3 B**. Se decidió realizar el proyecto escogiendo este dispositivo como plataforma ya que se encontraba disponible en el laboratorio, y además, de ser la última versión y poseer conectividad *wireless.* Se encargará de la captación de señales, procesamiento y monitorización de la mismas y generación de la señal de control.

FIGURA 6.4 RASPBERRY PI 3 B

Dicho esto, la Raspberry Pi es el soporte en el cual se incorpora el algoritmo de control. Con ayuda de un **PC** y del software de programación **Matlab & Simulink**, se monitoriza las señales de entrada a las Raspberry Pi para poder conocer la respuesta del sistema para una señal de excitación conocida. De esta manera, se caracteriza la planta para, posteriormente, diseñar y sintonizar el algoritmo de control.

En este método de funcionamiento, se dice que la Raspberry Pi trabaja como *Target* y el PC trabaja como *host.* El PC es el sistema responsable de la creación del algoritmo de control y la Raspberry es el soporte al cual se vuelca dicho algoritmo. Una vez desarrollado el algoritmo de control se compila generando un código en lenguaje "c" que permita a ejecución en el dispositivo *Target*. La conexión ambos computadores se realizará mediante un cable Ethernet.

Por último, la señal de actuación proporcionada por el algoritmo de control necesariamente tiene que ser analógica. Tal y como se ha explicado previamente, los pines tanto de entrada como de salida son digitales. En el caso de las salidas, no es necesario la incorporación de un circuito integrado que proporcione una conversión de señales, ya que, mediante una modulación de ancho de pulso **PWM (**Pulse-Width Modulation) se puede obtener una señal analógica a partir de una digital. Simulink en este caso, dispone de un Toolbox que permite dicha modulación.

El sistema de control en su conjunto queda de esta de forma:

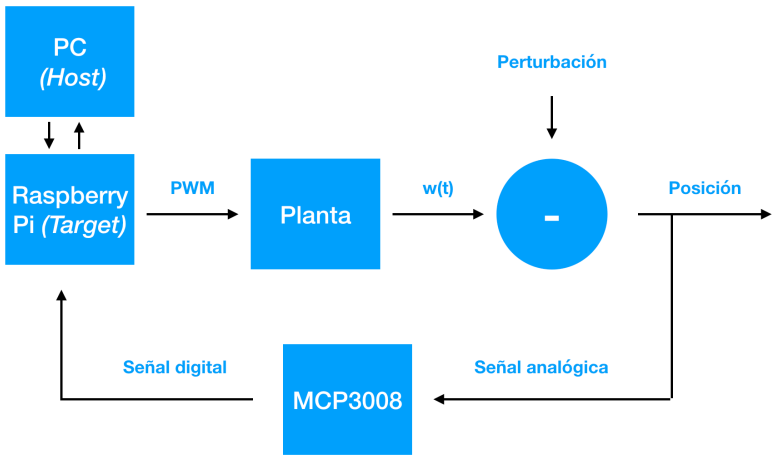


FIGURA 6.5 DIAGRAMA DE CONTROL

7. Metodología

Con el fin de implantar un sistema de control por computador sobre una sistema, se ha llevado acabo una serie de pasos preestablecidos. Desde la conversión del carácter de la entrada hasta la comunicación entre diferentes dispositivos. A continuación se muestra un diagrama de flujo que representa cada una de las etapas de implantación y verificación del sistema de control.

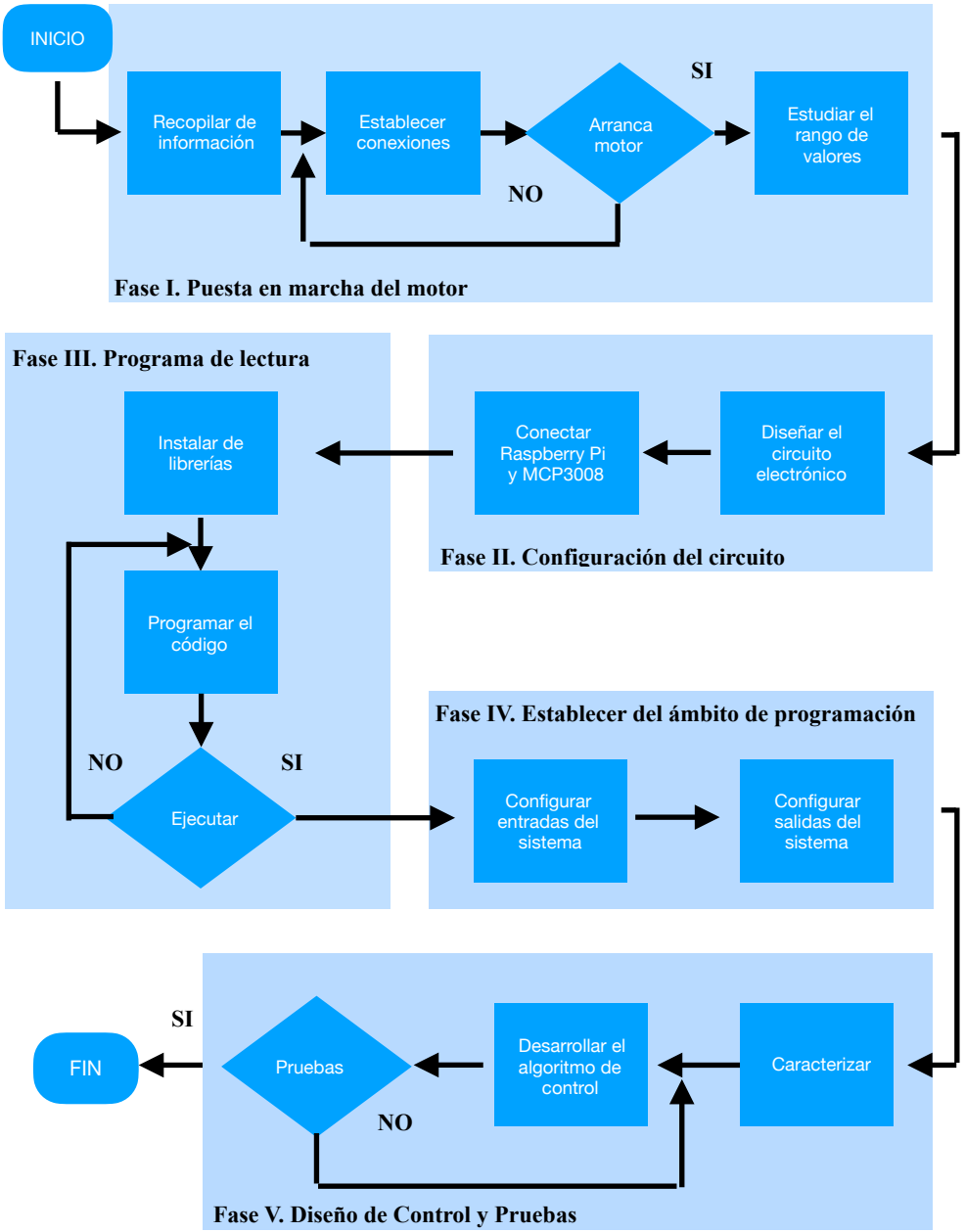


FIGURA 7.1 DIAGRAMA DE FLUJO

7.1 Fase I. Puesta en marcha del motor

El primer paso de todos es la puesta en marcha del motor para comprobar que el funcionamiento del sistema sea óptimo, dado que es necesario que el motor funcione para poder implantar un sistema de control. Por ello, se han realizado los siguientes pasos:

7.1.1. Recopilación de información

El motor admite dos modos de control: control por inducido o control por campo. Dependiendo de tipo de control que se desee, la conexión en bornes del amplificador es diferente. La decisión tomada en este proyecto, es realizar un control por inducido. La señal de actuación se generará mediante capos magnéticos, es decir, la señal de control que se mande al motor generará un campo magnético encargado de excitar el rotor del motor. La información necesaria para conocer las conexiones del amplificador, además de las conexiones del tacómetro y de la salida de la posición se encuentra en la hoja de especificaciones del sistema.

7.1.2. Establecer las conexiones

Antes de realizar las conexiones necesarias para el control, se ha alimentado el módulo de potencia del sistema y se ha conectado ha dicho módulo el amplificador y el motor mediante el cable de alimentación.

Una vez hecho esto, se ha conectado por una parte el amplificador. Con trazo rojo se muestra las conexiones pertinentes para el control por inducido. En el caso de querer realizar un control por campo se conectarían las bornes unidos por la letra F. Por otro lado, se encuentran los bornes 1 y 2 en los cuales se conecta la señal de control. Existen dos bornes para cada uno de las dos opciones de giro. Si la señal de control se conecta en la borda nº 1, el rotor girará en las agujas del reloj. Por el contrario, si se conecta en la borda nº 2 el rotor girará al contrario de las agujas del reloj. La señal control siempre debe ser positiva.

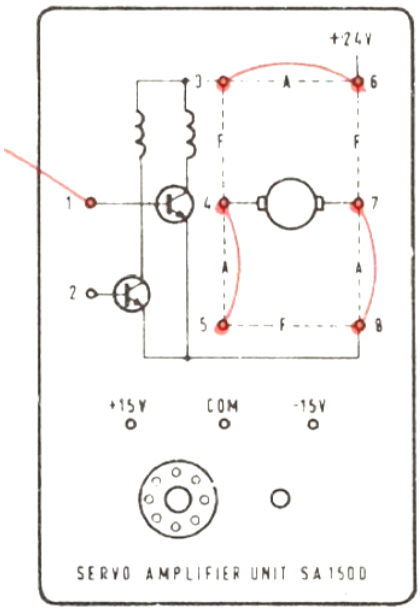


FIGURA 7.2 AMPLIFICADOR

En relación al tacómetro, la tensión entre las bornas 1 y 2 representa la velocidad a la que se encuentra el rotor. A mayor velocidad mayor será la tensión generada entre bornes. Para establecer una referencia uno de los pines, en este caso el 2, se conectara a tierra (COM). Cabe recalcar que dependiendo del sentido al que gire el rotor, la tensión inducida en las bornas puede ser de valor negativo o positivo. Debido a esto, se extraerá al circuito electrónico dos señales y cada una de ellas se conectará de forma inversa en

el circuito electrónico, con el fin de que una de ellas siempre sea positiva dado que el conversor MCP3008 solo realiza lecturas de valores positivos.

En el caso del módulo de posición del rotor, se establecerá una tensión de referencia entre los bornes 1 y 2, estableciendo el rango de valores de la tensión generada por la posición del rotor en la borna 3. Si la posición del rotor es de 0º, el valor de la tensión será 0V. En el momento en que el rotor de encuentre a 180º la tensión en la borna 3 valdrá la mitad de la tensión de referencia. Es decir, a medida que aumente el ángulo del rotor, aumentará la tensión en la borna. Llegando como límite a la tensión de referencia. En este caso, la tensión de referencia escogida son 15V. Esta tensión la proporciona el módulo amplificador.

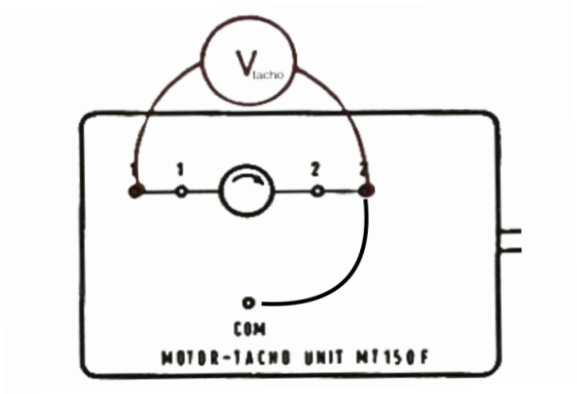


FIGURA 7.3 TACÓMETRO

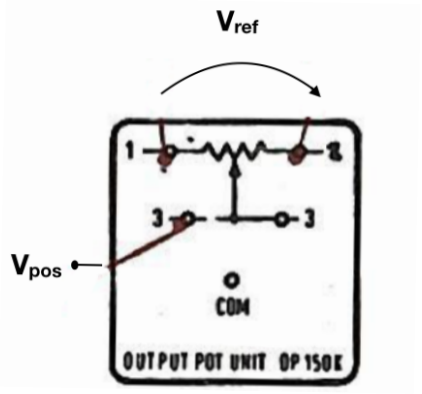


FIGURA 7.4 SENSOR DE POSICIÓN

7.1.3. Arrancar el motor. Comprobación de funcionamiento

Con ayuda de un fuente de alimentación, se genera un tensión en unos de las bornas del amplificador previamente descritos. Como resultado se obtiene un movimiento del rotor, verificando el funcionamiento del dispositivo. Por otro lado, se introducen las sondas de un polímetro en las bornas por las que se pretende obtener información sobre el comportamiento del motor para garantizar que los módulos tacómetro y de posición funcionan también correctamente.

FIGURA 7.5 FUENTE DE TENSIÓN

7.1.4. Estudiar el rango de valores

Por último será necesario conocer los rango de valores de la señal de control y de la señal del tacómetro. En el caso del motor, teóricamente se le puede excitar con una señal de entre 0-15V, pero debido a sus condiciones actuales y a la edad que tiene, su rango de tensiones real es de 4,2-10V. El rotor tiene un gran zona muerta debida a la fuerza de rozamiento que hace que hasta los 4,2V el rotor no comience su movimiento. Y por otro lado, a partir de los 10V genera unas perturbaciones que incomodan al usurario y generan mucho ruido.

En el caso del tacómetro, el rango de tensión es de 0-9,6V. Cuando se excita el motor con una tensión límite establecida de 10V, el tacómetro muestra una lectura de 9,6V exactamente.

7.2 Fase II. Configuración el circuito electrónico

A la hora de configurar el circuito electrónico para conectar las señales de entrada al convertidor MCP3008, se debe tener en cuenta los valores límite de tensión e intensidad que puede soportar el dispositivo con el fin de no dañarlo durante el funcionamiento. El rango de tensiones en las cuales trabaja el MCP3008 es de 0-5V y trabajando a 5V la intensidad no debe de superar un límite de 2mA.

7.2.1. Diseño del circuito electrónico

Por una parte, se encuentran las señales provenientes del tacómetro, y por otra, la señal del sensor de posición. Cada una de las señales tiene un rango de valores diferente como se ha explicado previamente.

El diseño de los circuitos de las señales del tacómetro se hace del tal forma que siempre una de las dos señales salientes hacia el MCP3008 sea positiva. Ambas señales provenientes del tacómetro son de igual magnitud y sentido, por ello, el circuito electrónico deberá ser el mismo para obtener la misma reducción de tensión (de 10V a 5V) pero la conexión del MCP3008 al circuito debe de ser de forma inversa.

En la primera figura, se toma una de las dos señales del tacómetro y se considera que esa señal siempre tendrá un valor positivo de tensión. La reducción de la tensión a la mitad se consigue con un divisor de tensiones mediante dos resistencias del mismo valor y que a su vez limiten la corriente a 2mA. En este caso, las resistencias escogidas han sido de 3,32K ohmios para la primera entrada. Los cálculos necesarios para la obtención de dicha resistencias se encuentran en el apartado 9.

En la segunda figura, se ha tomado la otra señal y se ha considera que su valor siempre va a ser negativo. El cálculo de las resistencias es exactamente el mismo que en caso anterior. En cambio, la conexión como se puede apreciar, se hace de forma inversa. De esta manera, cuando el motor gire en el sentido de las agujas del reloj, el primer circuito proporcionará la lectura correcta. Cuando gire en el sentido contrario de las agujas del reloj, las señal proveniente del motor será negativa, siendo el segundo circuito el que proporcione la lectura correcta.

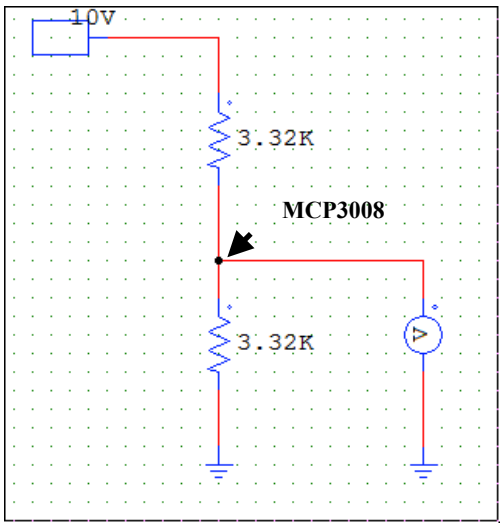


FIGURA 7.6 CIRCUITO SEÑAL POSITIVA

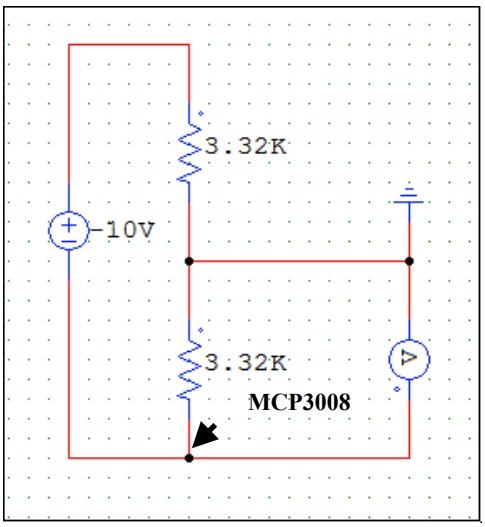


FIGURA 7.7 CIRCUITO SEÑAL NEGATIVA

En el caso de la señal proporcionada por el sensor de posición, el divisor de tensiones debe reducir la tensión de 15V a 5V, para ello, será necesario dos resistencias, una con un valor en ohmios el doble que la otra. Las resistencias escogidas han sido de 10K y 3.32K. El circuito sería el siguiente:

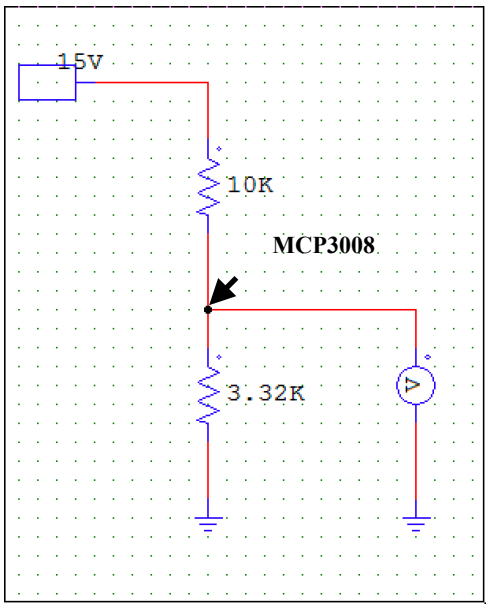


FIGURA 7.8 CIRCUITO SEÑAL DE POSICIÓN

La salida de cada uno de estos circuitos se han conectado a tres canales del MCP3008 diferentes. La lectura del taco positiva se ha introducido en el canal 1, la negativa en el canal 3 y la del sensor de posición se ha introducido en el canal 2.

7.2.2. Conexión entre Raspberry Pi y MCP3008

El MCP3008 y la Raspberry Pi se conectan usando la conexión en serie SPI. Esta conexión se puede realizar por el bus SPI o utilizando cuatro pines GPIO. Aprovechando que existe un canal predefinido para este tipo de comunicación entre dispositivos se ha conectado los dispositivos mediante esta conexiones. En caso de querer hacer la conexión mediante los pines GPIO, habría que programar el software SPI para que los pines proporcionen la información entre dispositivos de la forma correcta.

La conexión de los pines para una comunicación en serie SPI entre Raspberry Pi y MCP3008 es la siguiente:

MCP3008	Raspberry Pi
VDD	5V
VREF	5V
AGND	GND
DGND	GND
CLK	SCLK
DOUT	MISO
DIN	MOSI
CS/SHDN	CE0

TABLA 2. CONEXIONES ENTRE RASPBERRY Y MCP3008

El soporte donde se ha situado todo el circuito electrónico, el circuito integrado y la extensión de los pines de la Raspberry Pi ha sido una ProtoBoard, ya que permite flexibilidad a la hora de cambiar los componentes y realizar nuevas conexiones. El resultado de todo el circuito en su conjunto es el siguiente:

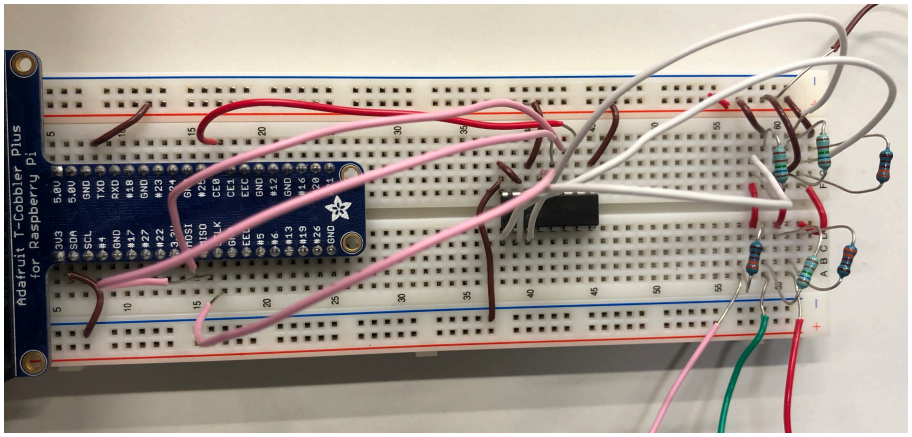


FIGURA 7.9 CIRCUITO REAL

El último paso para finalizar con la conexión del hardware implicado en el proyecto, es conectar la Raspberry Pi a un monitor mediante HDMI, un teclado mediante USB, el cable del Ethernet y la alimentación por el puerto micro-USB.

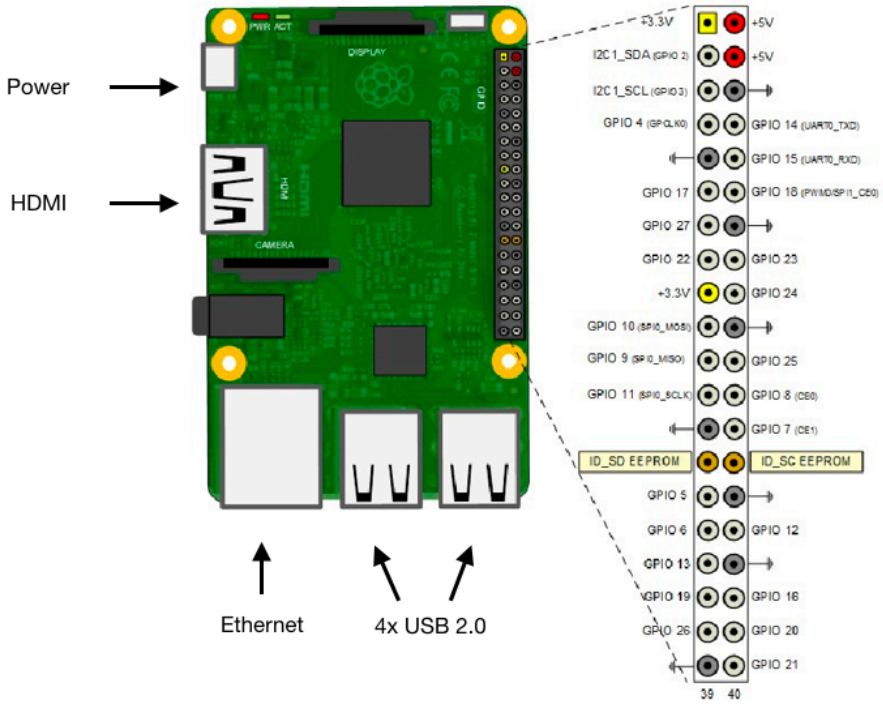


FIGURA 7.10 CONEXIONES RASPBERRY PI

7.3 Fase III. Desarrollo de programa de lectura

Una vez realizada la conexión hardware de todos los dispositivos, se ha desarrollado un programa en código "c" para que se ejecute en la Raspberry Pi con la capacidad de leer las lecturas de los diferentes canales del MCP3008. Con la programación en la Raspberry Pi se pretende obtener un programa sin fallos que posteriormente se pueda volcar a una S-Function de Simuink. De esta manera, se asegura que cualquier fallo durante la programación en Simuink, no es debido al código del programa sino algo relacionado con el propio Simulink.

7.3.1. Instalación de librerías WiringPi

Para la programación de un código capaz de leer las entradas del convertidor analógico digital mediante una comunicación SPI, es de gran ayuda acudir a la librería **WiringPi**. Esta librería permiten el acceso a los pines GPIO e incluye un linea de comandos para programar e inicializar los pines GPIO.

WiringPi soporta la lectura y escritura analógica, con lo cual, es relativamente sencillo y recomendable la implantación de convertidores analógicos digitales con la Raspberry Pi mediante le uso de esta librería.

En primer lugar, cabe descartar que para realizar la instalación de la librería es necesario la conexión de la Raspberry Pi a internet mediante Wifi. No se puede realizar una conexión mediante el cable Ethernet, ya que, se ha de utilizar para conectarla al PC. Suponiendo que la Raspberry Pi ya esta conectada. El enlace en el cual está disponible la descarga de la librería es el siguiente: https://git.drogon.net/ . Después de descargar el archivo, estos son los comandos que se han utilizado para la instalación de la librería:

```
$ gpio -v
```

```
$ sudo apt-get purge wiringpi
$ hash -r
```

```
$ sudo apt-get install git-core
```

```
$ sudo apt-get update
$ sudo apt-get upgrade
```

```
$ cd
$ git clone git://git.drogon.net/wiringPi
```

```
$ cd ~/wiringPi
$ git pull origin
```

```
$ cd ~/wiringPi
$ ./build
```

FIGURA 7.11 COMANDOS INSTALACIÓN DE LIBRERÍAS

Para poder incluir la librería en el código del programa es imprescindible que dicha librería este instalada, en caso contrario, el programa no entenderá los comandos utilizados. También es imprescindible la inclusión en el código de la librería *mcp3004.h* contenida en el archivo descargado previamente. La librería corresponde al MCP3004 pero también sirve para el MCP3008 ya que la única diferencia es la cantidad de canales en entrada disponibles.

Por defecto, la Raspberry Pi tiene inhabilitados ciertos pines GPIO, por ello se ha acudido a la interfaces de configuración, escribiendo *sudo raspi-config,* desde la cual se puede habilitar o inhabilitar cada uno de los pines. En este caso, se han habilitado los destinados a la comunicación SPI, y de esta manera, la librería wiringPi puede acceder y hacer uso de esos pines.

7.3.2. Programación del código

Una vez instalada la librería, se ha procedido a la programación del código encargado de las lecturas del MCP3008 al que se le ha dado el nombre de *Lecturas_MCP3008.c.* Los comandos utilizados en la programa son: *wiringPiSetup*, *mcp3004Setup* y *analogRead*. El código del programa en "c" se ha subido a la plataforma GitHub con el fin de que sea disponible para todo el mundo y también se adjunta en el Anexo 3. El enlace a dicho código es el siguiente: https://goo.gl/roJLPR.

La información proporcionada por el MCP3008 a la raspberry Pi, tiene un rango de valores de 0-1023, es decir, la tensión analógica conectada en el canal la convierte en un numero entero del 0 al 1023. Si la entrada son 5V (máximo) le corresponderá el valor 1023. Si la entrada son 2.5V le corresponderá el valor 512. Por esto, a la variable en la que se introduce la información aportada por el MCP3008 se le ha aplicado un cambio de escala, y así, el valor mostrado en pantalla es el valor real de la tensión en los canales del MCP3008. A continuación se muestra la escala, donde V_{ref} en este caso es 5V.

$$LSB\ Size = \frac{V_{REF}}{1024}$$

FIGURA 6.12 RELACIÓN DE CONVERSIÓN

7.3.3. Ejecución del Programa

El siguiente paso ha sido la verificación del funcionamiento del programa *Lecturas_MCP3008.c.* Se ha alimentado el sistema y se ha inducido una tensión en bornas del motor mediante la fuente de tensión. Al ejecutar el programa, el valor que mostrado en pantalla ha sido idéntico al mostrado por el polímetro conectado e los canales de entrada del MCP3008. Verificando que la programación del código ha sido la correcta.

7.4 Fase IV. Establecimiento del ámbito de programación

En este proyecto el software en el cual se va a diseñar el sistema de control es MATLAB & Simulink. El diagrama de bloques del sistema de control se va a alojar en Simulink concretamente. Se ha preferido recurrir a Simulink ya que la programación es más visual y además, permite simular y descargar el algoritmo en la Raspberry Pi y trabajar como *host* en el ordenador, a diferencia del Matlab que no lo permite.

Nada más arrancar el programa, ha sido necesario la instalación de paquetes hardware para poder trabajar con Raspberry pi en Matlab & Simulink. Dichos paquetes son: ***Matlab Support Package for Raspberry Pi Hardware*** y ***Simulink Support Package for Raspberry Pi Hardware.*** Con ambos paquetes se puede comunicar el PC de forma remota con la Raspberry Pi y utilizarla para controlar dispositivos periféricos. Estos soportes permiten adquirir y enviar datos a la Raspberry Pi, es decir, permite la lectura y la escritura de los mismos. El paquete de Simulink es el encargado de descargar el algoritmo completo para su ejecución autónoma en el dispositivo. Además de ofrecer la capacidad de ajustar los parámetros en tiempo real del modelo mientras el algoritmo se ejecuta en el hardware.

La descargar e instalación de ambos paquetes se llevan acabo desde el propio programa de Matlab. Acudiendo a la pestaña de **Add-Ons > Get Hardware Support Packages.** El propio programa va proporcionando las instrucciones necesarias a seguir para la instalación de dichos paquetes.

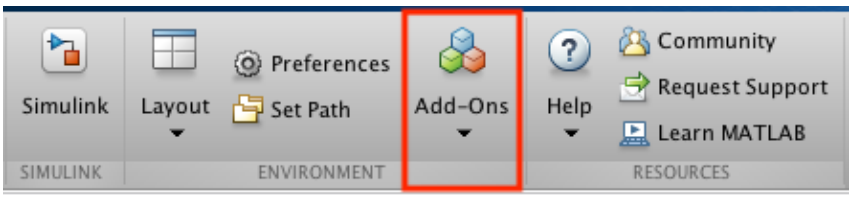


FIGURA 7.13 ADD-ONS

Para introducir la señal provenientes de los sensores en el programa de control hay dos alternativas para elección de los **drivers** a incorporar:

S_FUCTION

Esta alternativa consiste en programar un S-Function a gusto del usuario. Posee un gran flexibilidad ya que permite introducir cualquier código en "c" desarrollado por el usuario además de incorporar librerías y las entradas y salidas necesarias. Es un método muy polivalente que requiere de conocimientos previos y que posiblemente no sea sencilla su implantación.

SPI REGISTER READ

La segunda alternativa es incorporar un bloque predefinido del Simulink Support Package llamado **SPI Register.** Este bloque se encarga de la lectura información proporcionado por el canal de comunicación SPI. Es una alternativa muy cómoda y sencilla de utilizar ya que el bloque esta definido por defecto. La desventaja que muestra es que muy poco flexible y la lectura que proporciona es directamente la información proveniente del MCP3008 sin poder hacer un cambio de escala dentro del bloque

7.4.1. Entradas del sistema

Después de haber desarrollado el código de lectura *Lecturas_MCP3008.c,* se ha procedido a incorporar dicho código a bloque S-Function de tal forma que función como Driver vinculando la parte Hardware y la parte Software.

El primer paso ha sido ajustar las entradas y salidas del bloque a las necesidades del proyecto. Se han eliminado todas las entradas al bloque y se han incorporado dos salidas, una de ellas la salida del tacómetro y otra la posición del rotor. Quedando de esta forma:

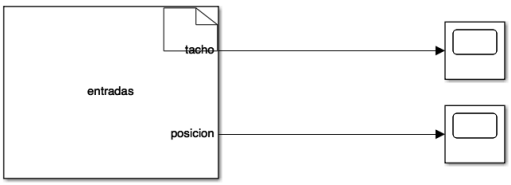


FIGURA 7.14 S-FUNCTION DE ENTRADAS

En el apartado de librerías, se ha incorporado la librerías utilizadas en el caso de *Lecturas_MCP3008.c,* y en el apartado de outputs, se ha introducido el código completo desarrolla en el programa. También se han introducido los comandos a utilizar para la lectura de los sensores y se tomado un tiempo de muestreo de 0.1s y un solo estado discreto.

Una vez llegado a este punto, se ha tomado la decisión de seguir adelante con la otra alternativa debido a su simplicidad y para probar con un método de programación diferente. Lo realizado hasta sigue siendo una herramienta muy útil que puede ser aprovechada y mejorada en cualquier otro proyecto .

En esta alternativa, simplemente se ha pinchado y arrastrado el driver predeterminado por el paquete y se ha abierto para establecer los parámetros. Los parámetros incluyen el modelo de la Raspberry pi a utilizar, el canal por el que se conecta el esclavo (MCP3008), el registro de la entrada a leer y el tipo de dato. El registro representa el canal del esclavo del cual se quiere obtener información. El tipo de dato que aporta el MCP3008 a la Raspberry Pi es de tipo *uint16*, establecido por el protocolo de comunicación SPI.

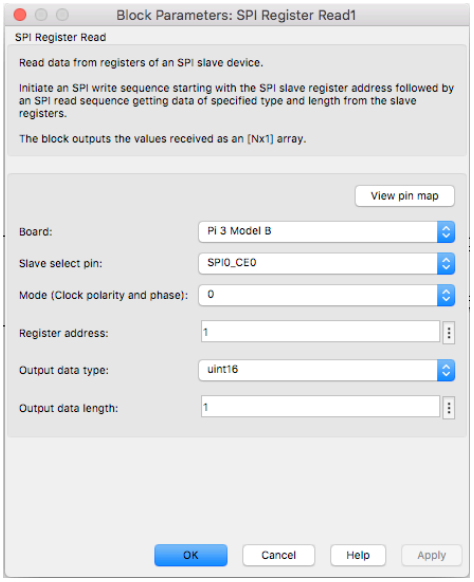


FIGURA 7.15 VENTANA DE PARAMETROS DE SPI REGISTER

La gran desventaja de esta alternativa se debe a que el bloque predefinido por Simulink no reconoce los diferentes canales del MCP3008. Simplemente, trata el integrado como si fuera un único canal. Debido a este handicap, a partir de aquí solamente se ha tenido que conectar al canal 1 aquella señal de la que quisiera tener información. Cabe destacar, que solamente se podría hacer un control de velocidad en un solo sentido de giro, ya que no se pueden disponer de las dos señales de salida del tacómetro.

7.4.2. Salidas del sistema

Para cerrar el ciclo de control, la Raspberry Pi debe proporcionar la señal de control al motor generando un señal de referencia en bornes del motor que haga que gire el rotor. Dicha señal de control se puede proporcionar por cualquiera de los pines libres de la placa. Aquellos pines son los GPIO numerados y que no están destinados para cualquier modelo de comunicación predeterminado.

Cabe destacar que la señal proporcionada por estos pines es una señal digital que pasa de 0V a 3,3V, cuando la señal de referencia a inducir en las bornas del motor es debe de ser analógica que oscile de 4,2V a 10V. Se han tomado dos medidas para **la conversión digital analógica** del carácter de la salida y el **aumento de la tensión de 3,3V a 10V.**

CONVERSIÓN ANALÓGICA DIGITAL

El Simulink hardware support package incluye un bloque predefinido llamado PWM que mediante una modulación de ancho de pulso de la señal digital de los GPIO permite extraer una señal analógica de 0V a 3,3V. La entrada al bloque es el ***Duty Cycle*** o **ciclo de trabajo** que es la relación existente entre el tiempo en que la señal de encuentra en estado activo (3,3V) con el periodo de dicha señal. El valor del *Duty Cycle* oscila de 0 a 1.

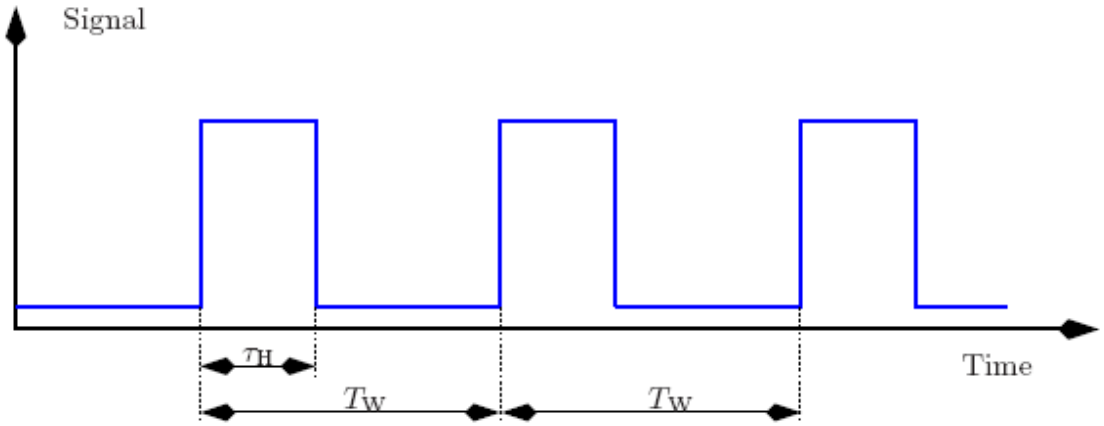


FIGURA 7.16 DUTY CYCLE O CICLO DE TRABAJO

A modo de ejemplo, estos son posibles situaciones de trabajo de la salida mediante PWM:

- 0% Duty: Cuando el *Duty Cycle* es igual a cero, la señal de la salida nunca estará activa, entonces la señal será de 0V.
- 50% Duty: Cuando el *Duty Cycle* es igual del 50%, significa que la mitad del periodo de la señal se encuentra activa y la otra mitad la señal sería 0V, obteniendo como resultado una señal analógica de 1,65V.
- 100% Duty: Cuando el *Duty Cycle* es igual al 100%, la señal simpre estará activa y la señal resultante será de 3,3V

El bloque de salida PWM es el siguiente:

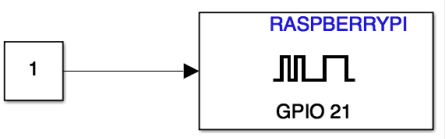


FIGURA 7.17 BLOQUE PWM

De esta forma, se ha obtenido disponer de una tensión analógica en los pines GPIO. Se han habilitado dos señales de salida de la Raspberry (GPIO 20 y GPIO 21) para hacer girar el motor en los dos sentidos. En la propiedades del bloque se necesita especificar el modelo de la Raspberry Pi, el pin de salida y la frecuencia del ciclo.

ELEVAR LA TENSIÓN

Con el fin de elevar la tensión de salida del rango de valores 0-3,3V a 0-10V. Para llevarlo a cabo, se ha recurrido a un técnico electrónico del departamento para el desarrollo de la circuiteria necesaria dado que no es objeto de estudio en este proyecto.

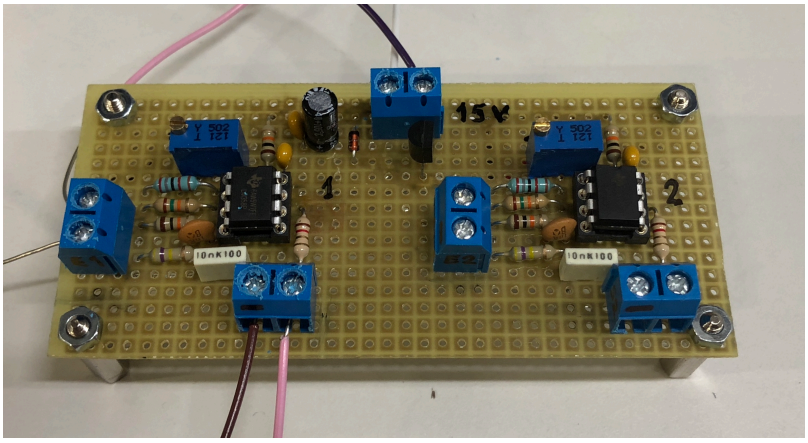


FIGURA 7.18 CIRCUITO ELEVADOR DE TENSIÓN

7.5 Fase V. Diseño de Control y Pruebas

Una vez conectadas la señales de los sensores y la señal de control de la Raspberry Pi se ha procedido al diseño de sistema de control de bloques en Simulink. La magnitud escogida para controlar es la posición del rotor, es decir, se ha realizado un control de posición. Para ello, se ha caracterizado la planta y posteriormente se desarrollado en sistema de control en bucle cerrado para su posterior control. Es importante que para que el programa se ejecute en la Raspberry Pi, el Matlab la debe reconocer, para ello, se hará el uso del comando *raspi()*. Además, se necesita configurar la ventana de Simulink de tal forma que el programa se ejecute en la RPI y no en el propio ordenador. Se configura de la siguiente manera: se accede a ***Tools > Run on Target Hardware > Options…> Hardware Implementation*** y se configura la placa con la que se va a trabajar. Y por último, se selecciona el modo "***Extern"*** para que el programa se ejecute en el *Target.*

7.5.1. Caracterización

Para la caracterización de la planta, se ha excitado en bucle abierto observando la respuesta a la excitación y de esta forma, se ha obtenido el orden de la función de transferencia de la misma. El control que se desea realiza es un control de la posición del rotor pero como la respuesta de la velocidad es mucho más sencilla de analizar, se ha decidido, obtener la función de transferencia de la planta según la magnitud de la velocidad y, posteriormente, se ha multiplicado por un integrador, obteniendo como resultado la función de transferencia según la posición. En la siguiente figura se observa lo explicado, donde $G_1(s)$ es la función de transferencia de la velocidad.

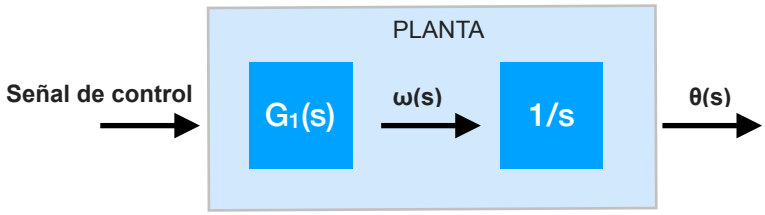


FIGURA 7.19 ESQUEMA FUNCIÓN DE TRANSFERENCIA DE LA PLANTA

A continuación, se ha procedido a excitar el sistema con una entrada escalón de 10V en bornas del motor. Para ello, la tensión en bornas en la entrada del elevador de tensión tiene que ser de 3,3V, es decir, que el *Duty Cycle* del bucle de PWM tiene que ser 1. Se ha programado los bloques de simulink de tal forma que el usuario introduzca directamente la tensión en bornas del motor que desee, sin tener que pensar cual es el *Duty Cycle* necesario para el valor deseado. Como se ha explicado previamente, con el método utilizado para la lectura de los sensores, solamente puede tomarse las lecturas del canal 1 del MCP3008, por ello, se ha conectado la señal del tacho a dicho canal.

La respuesta ante la entrada escalón de 10V es la siguiente:

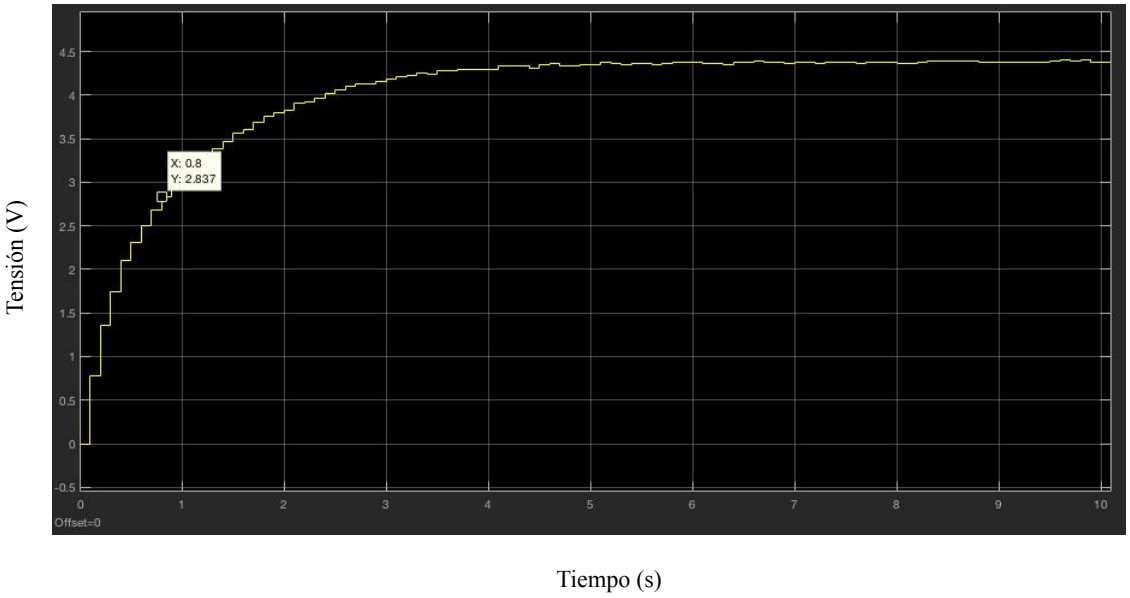


FIGURA 7.20 RESPUESTA ANTE ENTRADA ESCALÓN

Se puede verificar como la respuesta obtenida es característica un sistema de orden 1. Mediante un cálculos, especificados en el apartado 9, se ha concluido obteniendo una ganancia K del sistema de valor $K = 0{,}4375$ y una constante de tiempo, de valor de $\tau = 0.8$s. Con lo cual, la función transferencia $G_1(s)$queda de la siguiente forma:

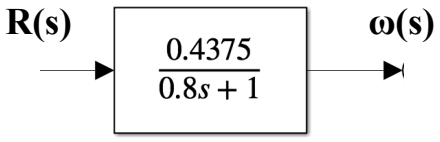


FIGURA 7.21 FUNCIÓN DE TRANSFERENCIA G1(S)

Por lo tanto, la función de transferencia del sistema en función de la posición obtenida es la anterior multiplicada por un integrador: $G(s) = G1(s) * 1/s$. Siendo la función de transferencia obtenida G(s):

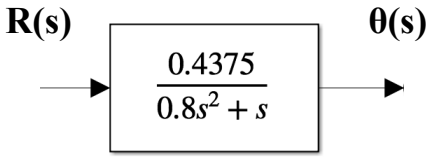


FIGURA 7.22 FUNCIÓN DE TRANSFERENCIA G(S)

A continuación, para facilitar el uso del programa de control al usuario, se ha caracterizado los sensores y el actuador del motor para poder introducir el valor del ángulo o velocidad en la que desea situar el rotor del motor. Se considera que los sensores y el actuador no tienen dinámica y que la función de transferencia que los representa es una ganancia K. Se ha extraído el voltaje generado en las bornas de los diferentes módulo para las diferentes valores de la magnitud a medir, y de esta forma, se ha podido relacionar la relación de la magnitud con el voltaje.

En la siguiente figura se representa el voltaje obtenido en el sensor de posición para los diferentes ángulos del rotor:

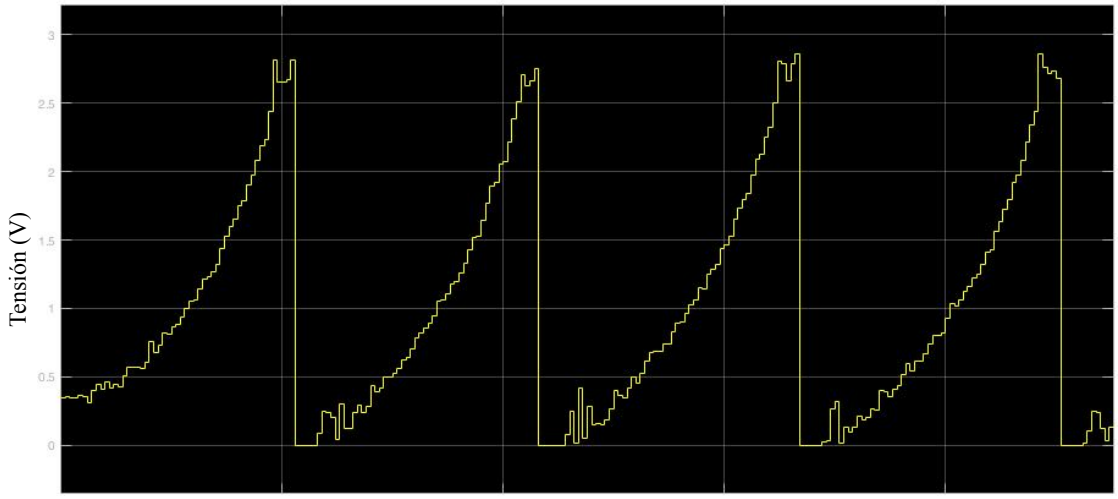


FIGURA 7.23 SALIDA DEL SENSOR DE POSICIÓN

La relación de voltaje y posición se puede aproximar a una relación lineal con pendiente igual a $m = 3{,}8/360 = 0.0105$. Por otro lado también se ha caracterizado el actuador del motor y el tacómetro con ayuda de un tacómetro externo que proporcione la velocidad a la que gira el rotor. A continuación se ven los valores obtenidos:

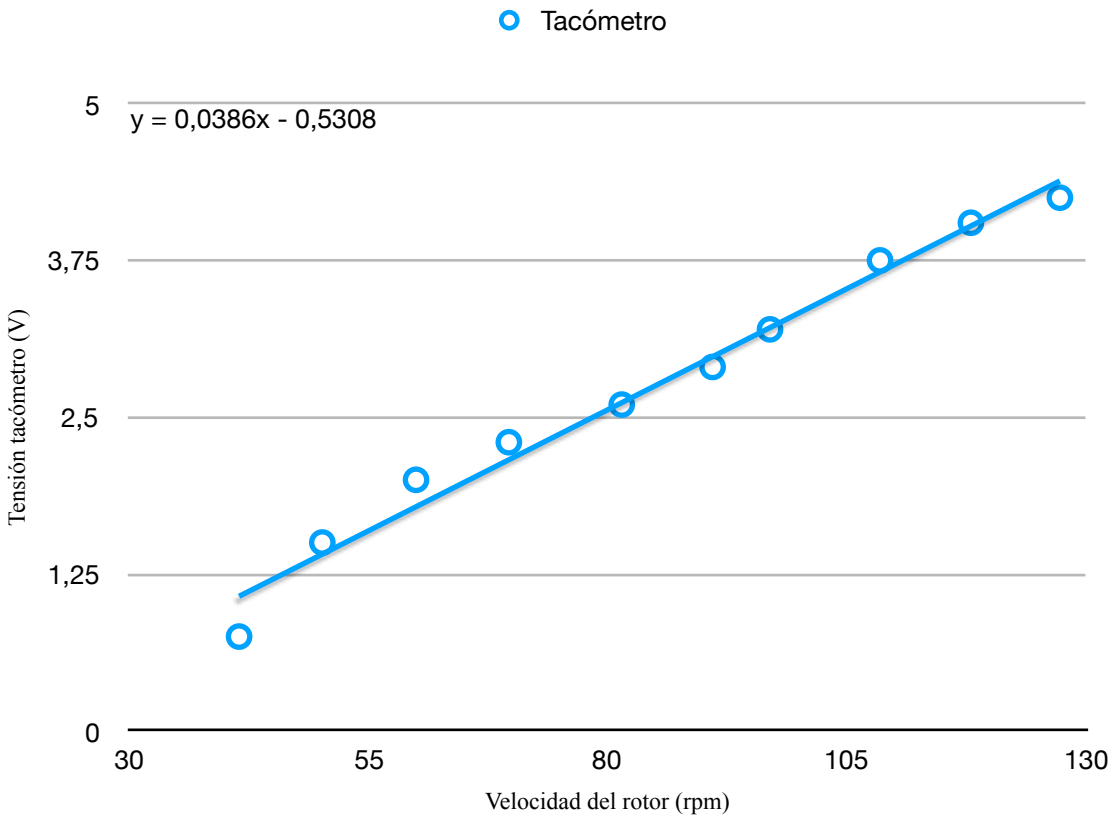


GRÁFICO 1. RELACIÓN DE TENSIÓN Y VELOCIDAD DEL TACÓMETRO

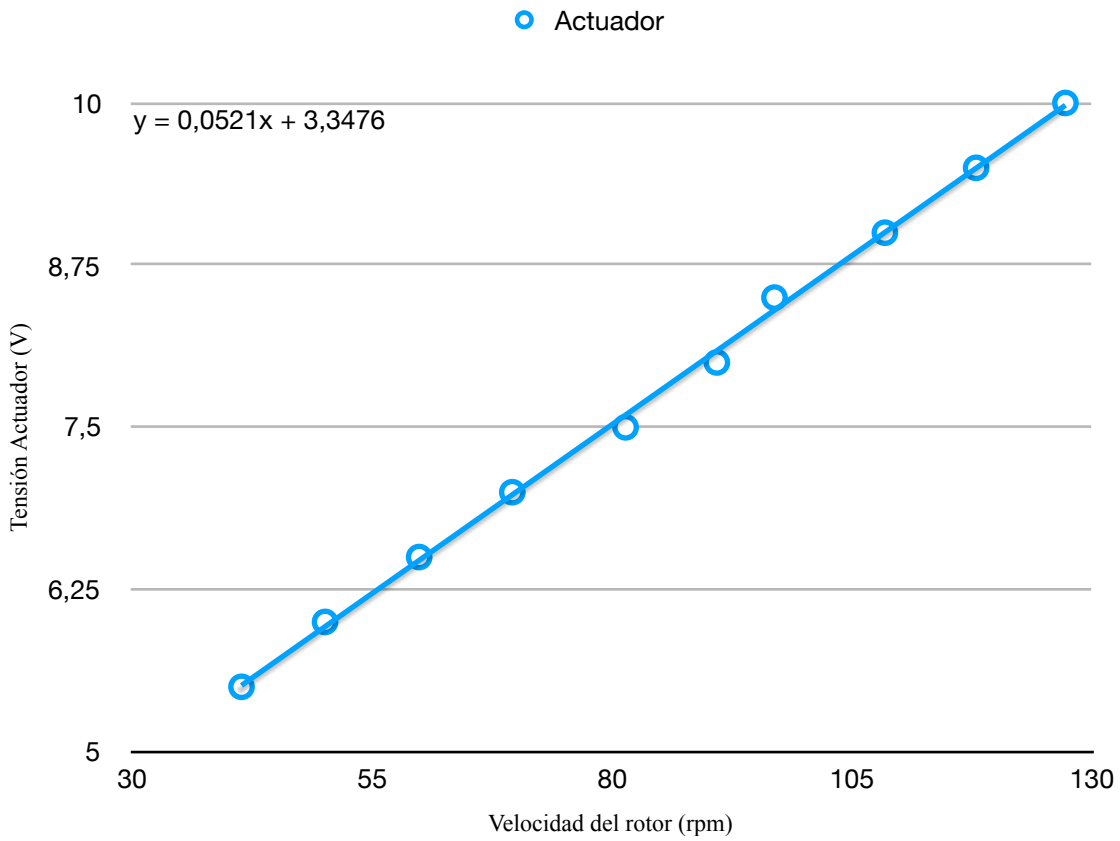


GRÁFICO 2. RELACIÓN DE TENSIÓN Y VELOCIDAD DEL ACTUADOR

Se puede considerar que ambas tensiones tiene una relación lineal respecto a la velocidad que gira el rotor. Por un lado, el actuador actuador posee una relación con una pendiente igual a 0,0521 y, por otro, el tacómetro tiene una relación de pendiente igual a 0,0386.

7.5.2 Desarrollo del algoritmo de control

El siguiente paso ha sido el diseño y desarrollo del algoritmo de control de la posición de rotor. El diseño del algoritmo ha consistido en un bucle cerrado como el siguiente:

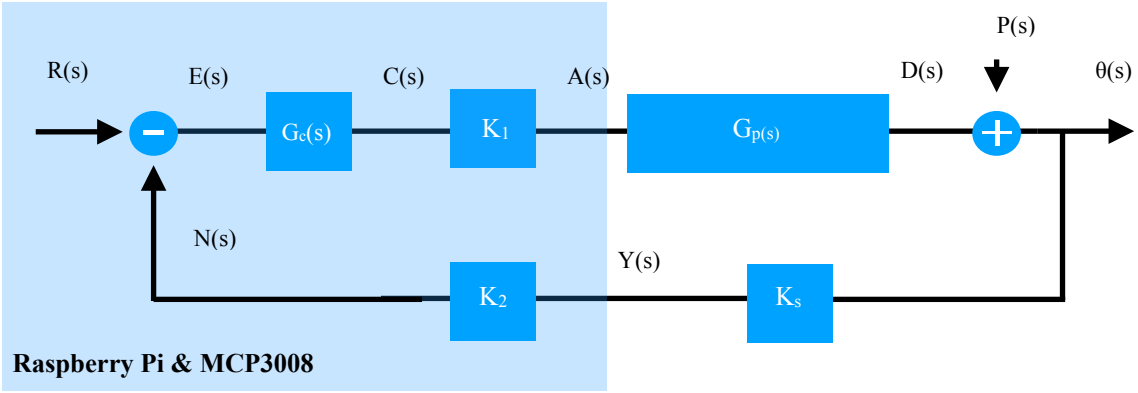


FIGURA 7.24 DIAGRAMA DE BLOQUES

- *R(s): Señal de entrada de la posición deseada (grados)*
- *E(s): Señal del error (grados)*
- *C(s): Señal de control (grados)*
- *A(s): Señal de control (V)*
- *P(s): Perturbación (grados)*
- *θ(s): Posición del rotor (grados)*
- *Y(s): posición del rotor (V)*
- *G_p(s): Función de transferencia de la planta más el actuador*
- *G_c(s): Función de transferencia del controlador*
- *K1: Ganancia de conversión de grados a voltios del actuador*
- *K2: Ganancia de conversión de voltios a grados del sensor*
- *Ksen: Ganancia del sensor*

En el diagrama se puede apreciar cuales son los bloques que pertenecen al software desarrollado en el *host* y cuales son bloques hardware reales del sistema completo.

En primer lugar, el algoritmo de control escogido ha sido un control proporcional sencillo con una ganancia Kc, simplemente para verificar que la plataforma desarrollada ha sido implantada con éxito. Los diagramas desarrollados en Simulink se incorporarán al Anexo 4.

Ha sido relevante desarrollar un programa que no saturase el actuador y, para conseguir esto, se ha programado de tal forma que si las señal de control superan el límite permitido, el algoritmo genera como resultado una señal igual al límite.

La ganancia para el control proporcional se ha escogido de tal forma que cuando el ángulo de error sea de 2.5º, la señal de control sea la máxima posible, es decir, 10V. Por lo tanto la ganancia K_c:

$$Kc = C(s)/E(s) = 10V/2{,}5 = 4$$

8. Diagrama de Gantt

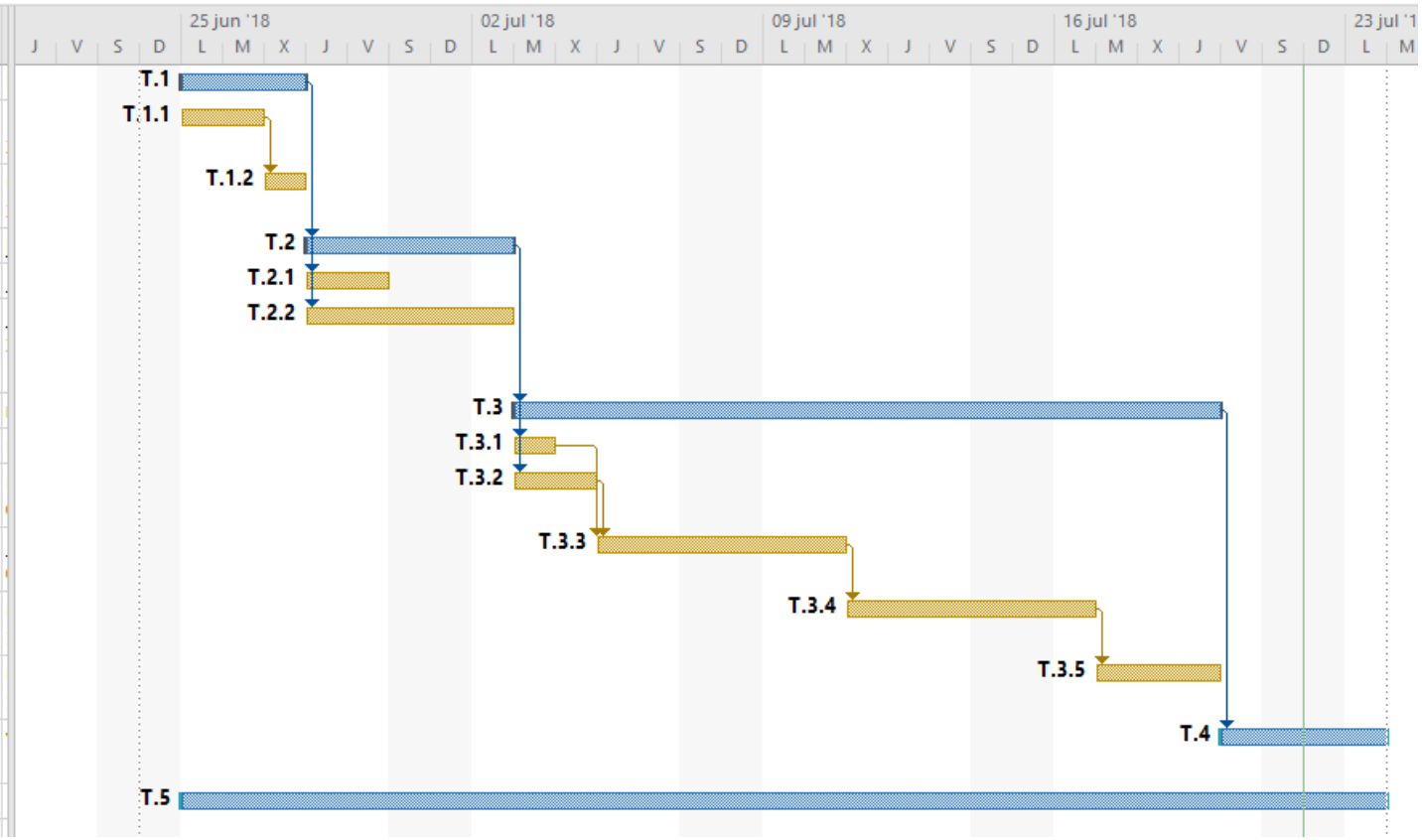
La planificación del proyecto se ha dividido en 5 grandes fases como se ha explicado previamente. El desarrollo del proyecto comenzó el día 25 de Junio de 2018 y se ha dado por finalizado el día 23 de Julio de 2018. Sin tener en cuenta los fines de semana, el proyecto a tenido una carga de trabajo de 21 días. Si es estima que se le han dedicado 7,5 horas diarias al proyecto, se obtiene una duración de 158h de trabajo.

Los paquetes de trabajo principales en lo que se ha dividido el proyecto son los siguientes:

- T.1 Inicio (3 días)
  - T.1.1 Análisis del objeto del proyecto (2 días)
  - T.1.2 Documentación del hardware disponible y análisis de la información (1días)
- T.2 Gestión de componentes (3 días)
  - T.2.1 Obtención de componentes (2 día)
  - T2.2 Desarrollo del circuito elevador por parte del técnico (3 días)
- T.3 Implantación (13 dias)
  - T3.1 Puesta en marcha del motor (1 día)
  - T3.2 Configuración del circuito electrónico (2 día)
  - T3.3 Desarrollo del programa de lectura (4 dias)
  - T3.4 Establecimiento del ámbito de programación (4 diás)
  - T3.5 Diseño de Control (3 días)
- T.4 Prueba de la plataforma de Control (2 día)
- T.5 Gestión del proyecto (21 días)

En el diagrama que Gantt que se incluye a continuación, se muestra detalladamente cuales la distribución de las tareas y la relación entre ellas.

EDT	Nombre de tarea	Duración
T.1	**Inicio**	**3 días**
T.1.1	Análisis del objeto del proyecto	2 días
T.1.2	Documentación y Análisis de la documentación	1 día
T.2	**Gestión del Hardware**	**3 días**
T.2.1	Obtención de componentes	2 días
T.2.2	Desarrollo del circuito elevador por parte del técnico	3 días
T.3	**Implantación**	**13 días**
T.3.1	Puesta en marcha del motor	1 día
T.3.2	Configuración del circuito electrónico	2 días
T.3.3	Desarrollo del programa de lectura	4 días
T.3.4	Establecimiento del ámbito de programación	4 días
T.3.5	Diseño de Control	3 días
T.4	**Pruebas de la plataforma de control**	**2 días**
T.5	**Gestión del proyecto**	**21 días**



9. Cálculo y algoritmos

A continuación se muestran los cálculos que han sido necesarios para la realización de proyecto. En primer lugar, se han hecho unos cálculos para la reducción de la tensión mediante el divisor de tensiones y, posteriormente, se han hecho unos calculo para la obtención de la función de transferencia del sistema *FeedBack MS150.*

9.1 Cálculo divisor de tensiones

Por un lado, se ha calculado el divisor de tensiones para las señales del tacómetro y por otro se ha calculado la del sensor de posición.

DIVISOR DE TENSIONES TACÓMETRO

La señal proveniente del tacómetro tiene un valor máximo de 10V. Para dicho valor, la señal entrante al MCP3008 tiene que tener un valor de 5V, con lo cual, el divisor de tensiones debe reducir la tensión a la mitad. Por otro lado, las especificaciones de MCP3008 señalan que la intensidad máxima que puede recorrer el dispositivo en las condiciones anteriores no debe superar un valor de 2mA. Con estos datos se han obtenido las resistencias necesarias para completar el circuito divisor de tensiones. Los cálculos han sido los siguientes:

$$\Delta V = I * (R1 + R2)$$

$$(R1 + R2) = \Delta V / I = 10V / 2mA = 5K$$

$$R1 = 5V / I = 5V * (R1 + R2) / AV$$

$$R1 / (R1 + R2) = 0.5$$

$$R1 = R2 = 2.5K$$

Como no es sencillo disponer de resistencias con valores exactos a los requeridos, se ha optado por recurrir a resistencias que garanticen la división de la tensión, es decir, que mantengan la misma relación de valor entre ambas resistencias, y que garanticen que la corriente no supere el valor límite. Las resistencias escogidas han sido de **3.65K.**

DIVISOR DE TENSIONES DEL SENSOR DE POSICIÓN

En cuanto a la señal del sensor de posición, tiene un valor máximo de 15V. En este caso, la división de la tensión debe ser 1/3. Los cálculos para obtener dicha división han sido los siguientes:

$$(R1 + R2) = \Delta V/I = 15V/2mA = 7.5K$$

$$R1 = 5V/I = 10V * (R1 + R2)/AV$$

$$R1/(R1 + R2) = 2/3$$

$$R1 = 5K; R2 = 2.5K$$

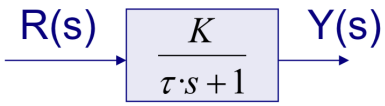
Al igual que ha hecho en el divisor de tensiones del tacómetro, se ha optado por tomar dos resistencias de más ohmios. Las resistencias son:

$$R1 = 3.32K; R2 = 10K$$

9.2 Cálculo de la función de transferencia

Para el cálculo de las funciones de transferencia del motor, se ha recurrido a las fórmulas estudiadas en Automática y Control.

Una vez obtenida la respuesta del sistema ante una entrada escalón conocida e identificando a que orden pertenece dicha respuesta, se ha recurrido a las ecuaciones que rigen el sistema, en este caso a un sistema de orden 1. A continuación se muestra la función de transferencia de un sistema de primer orden:



$$G(s) = \frac{Y(s)}{R(s)} = \frac{K}{\tau \cdot s + 1}$$

FIGURA 9.1 FUNCIÓN DE TRANSFERENCIA DE PRIMER ORDEN

Los dos parámetros para definir la función de transferencias son la ganancia estática [uds. Y/uds. U] y la constante de tipo del sistema [Uds. de tiempo], K y τ respectivamente. A continuación se muestra una figura que muestra las ecuaciones para

la obtención de ambos parámetros. El valor de la ganancia K se obtiene como cociente de la señal de salida del sistema entre la señal de entrada del sistema cuando el tiempo tiende a infinito o cuando s tiende a 0. El valor de la constante de tiempo es el tiempo en el que la salida alcanza el 63% de su valor final.

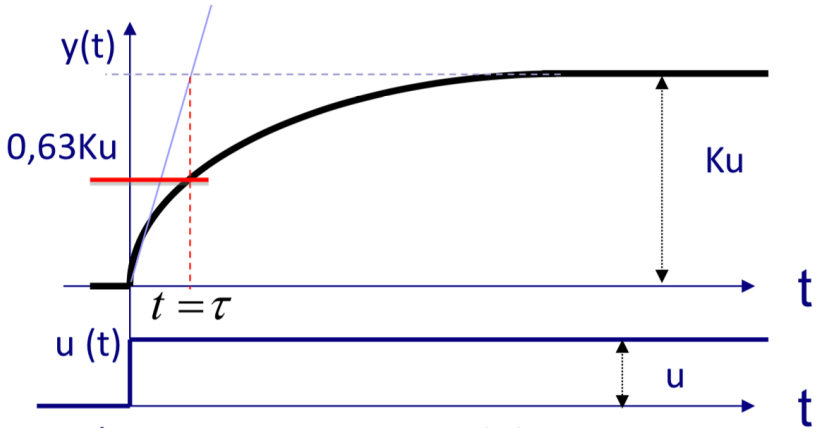


FIGURA 9.2 RESPUESTA TEMPORAL DE SISTEMA DE PRIMER ORDEN

La respuesta obtenida a la una entrada escalón de 10V en el motor ha sido la siguiente:

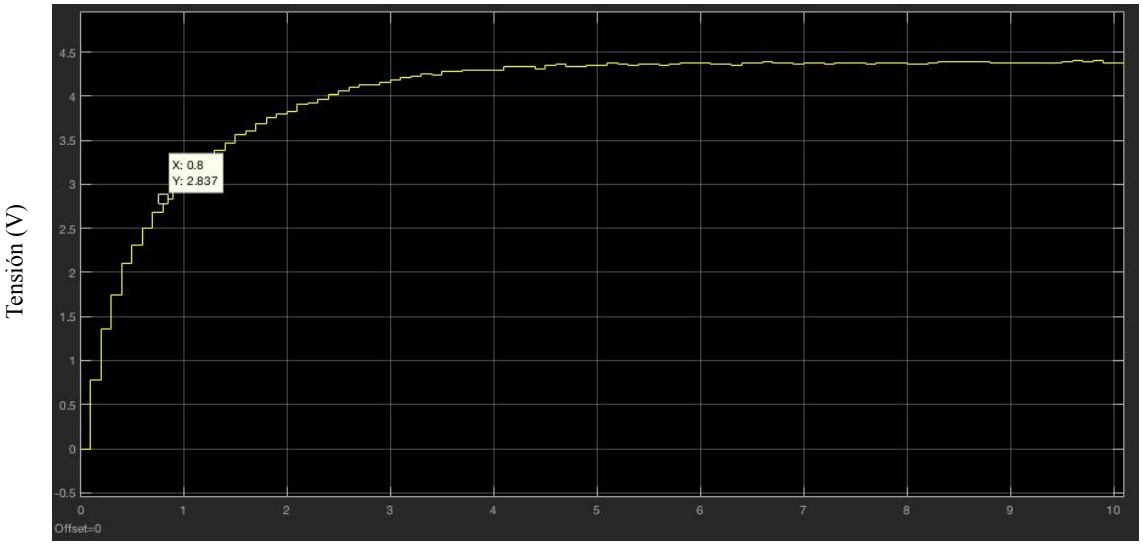


FIGURA 9.3 RESPUESTA ANTE ENTRADA ESCALÓN

El valor de la tensión de salida ha sido de 4,375V, por lo tanto:

$$yss = Ku$$

$$K = yss/u = 4{,}375V/10V = 0{,}4375$$

$$y(\tau) = 0{,}6321Ku = 2{,}8 \longrightarrow \tau = 0.8s$$

Por tanto, la función de transferencia obtenida para velocidad del rotor es:

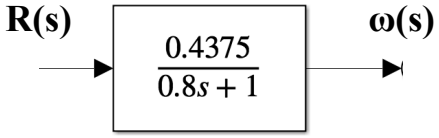


FIGURA 9.4 FUNCIÓN DE TRANSFERENCIA DE LA PLANTA

10. Resultados

Cabe destacar que la comunicación existente entre los diferentes dispositivos que completan la plataforma es la idónea y que se ha conseguido el principal objetivo del proyecto. A continuación se va a mostrar cuales han sido los resultados obtenidos del control de posición desarrollado en la plataforma implantada, con el objetivo de demostrar a su vez el cumplimiento del principal objetivo.

10.1 Control de la posición

Como se ha dicho previamente, se ha desarrollado un controlador proporcional y se ha tomado como señal de entrada un escalón de amplitud 20º y la respuesta del sistema ha sido la siguiente:

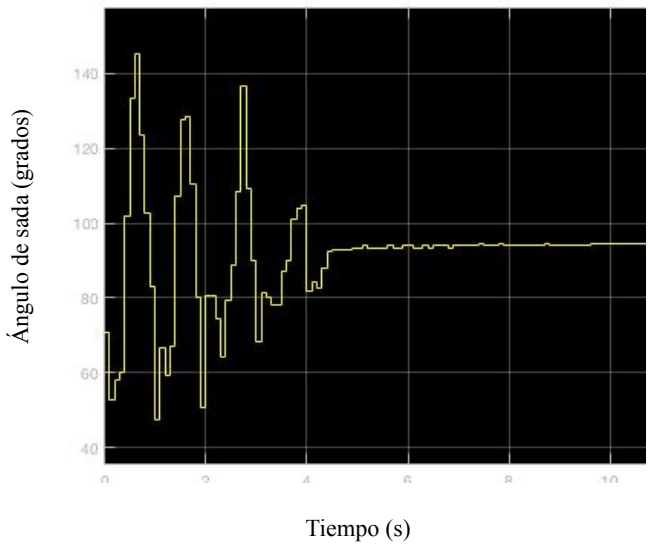


FIGURA 10.1 POSICIÓN DEL ROTOR

Como se puede observar en la gráfica, se trata de un sistema de orden dos debido a su carácter oscilatorio, verificando la función de transferencia obtenido en la caracterización de la planta. Por otro lado, la posición en régimen estacionario tiene un valor de 93.44º, cuando la respuesta esperada era 90º. El tipo de la planta con la que se trata en este proyecto es de tipo 1, con lo cual, en régimen estacionario no debería de tener un error, por ello, este error se debe a la gran fuerza de rozamiento que posee el rotor y las aproximaciones tomadas para el calculo de la ganancias del actuador y de los sensores, entre tantas. Cabe destacar que las condiciones en las que se encuentra el motor no son nada favorables, posee un gran zona muerta y además la potencia generada en la etapa de potencia del motor no es apenas estable. Por ello, lo resultado obtenidos en cuanto al control no son muy buenos pero ha servido para verificar que la implantación ha sido correcta y que la plataforma funciona perfectamente.

11. Descargo de gastos

En este apartado se presentan los gastos que se han originado en la realización del proyecto. Los gatos se desglosan en diferentes partidas, y se analiza por separado el impacto de cada una de ellas.

11.1 Amortizaciones:

Se han considerado como amortizaciones el uso del ordenador, el sistema Feedback MS150, la fuente de tensión y la Raspberry Pi 3. Cabe destacar que mientras el uso del MCP3008 y la electrónica es exclusivo para este proyecto, el resto de componentes, dada su versatilidad serán utilizados en proyectos ajenos a este. El cálculo de las amortizaciones se ha hecho de las siguiente manera:

Amortizaciones				
Sistema	**Coste Total**	**Años de Utilización**	**Uso Anual**	**Tasa Horaria**
Sistema Feedback MS150	12000,00 €	35	100h	3,43 €
Raspberry Pi 3 B	37,50 €	3	185h	0,07 €
Ordenador	1208,79 €	8	300h	0,50 €
Fuente de tensión	177,45 €	5	200h	0,18 €

TABLA 3. AMORTIZACIONES

Se incluye a continuación el descargo de gastos detallado por partidas. Se ha considerado un porcentaje de 7% para los costes derivados del proyecto (limpieza del laboratorio, luz,…). Se ha considerado el programa de Matlab y SimUlink como un gasto ya que se ha tomado en cuenta solamente la licencia de estudiante valorada en 80 euros.

Descargo de Gastos				
Concepto	**Unidades**	**Coste Unitario**	**Nº de Unidades**	**Coste**
Horas Internas				3915,00 €
Ingeniero	h	25,00 €	150	3750,00 €
Director	h	55,00 €	3	165,00 €
Amortizaciones				**291,02 €**
Ordenador	h	0,50 €	100	50,37 €
Feedback MS150	h	2,86 €	80	228,57 €
Fuente de tensión	h	0,18 €	30	5,32 €
Raspberry Pi	h	0,07 €	100	6,76 €
Gastos				**95,40 €**
MCP3008	-	2,40 €	1	2,40 €
Electrónica	-	5,00 €	1	5,00 €
Protoboard	-	8,00 €	1	8,00 €
Matlab & Simulink	-	80,00 €	1	80,00 €
Costes directos			4301,42 €	
Indirectos		7 %		301,10 €
Total			4602,52 €	

TABLA 4. DESCARGO DE GASTOS

Como se puede observar, la gran mayoría de los recursos invertidos en el proyecto con un 85% están destinados a las horas internas del ingeniero encargado de llevar a cabo el proyecto. El restos de gastos se han repartido en gran medida en amortizaciones debido al gran coste del sistema *Feedback MS150*. En cambio, el resto de elementos y dispositivos utilizados apenas suponen un gasto elevado respecto al proyecto en sus conjunto.

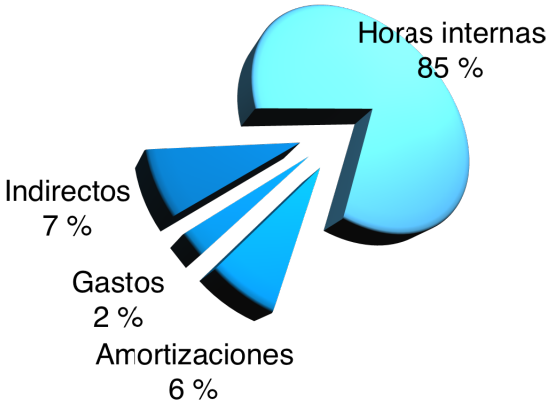


GRÁFICO 3. DESGLOSE DE GASTOS

12. Conclusiones

Se ha implementado un **plataforma de bajo coste** sobre el computador de placa reducida **Raspberry Pi** mediante la herramienta de **Software Matlab & Simulink** para el desarrollo de algoritmos de control. Un soporte con un fin didáctico para futuros alumnos tanto para la mejora de proyecto realizado como para el aprendizaje en sistemas de control.

La plataforma tiene la capacidad de soporta tanto señales analógicas como digitales y esta orientado para el control de sistemas existentes en el laboratorio. Además, permite la programación de cualquier algoritmo de control.

El hardware utilizado esta compuesto por un convertidor analógico digital llamado MCP3008, una Raspberry Pi y un circuito electrónico. La comunicación entre ambos dispositivos se realiza mediante el método de comunicación SPI. Por otro lado, el software se desarrolla PC (*host*) mediante el programa Simulink y posteriormente se vuelca a la Raspberry Pi (*Target*). El control desarrollado para verificación de la implantación ha sido un control proporcional.

El desarrollo del proyecto a supuesto una carga de trabajo de 158h repartidas en 21 días laborables. El proyecto comenzó el día 25 de Junio de 2018 y se dio por finalizado el día 23 de Julio de 2018.

El coste total del proyecto asciende a 4231,74 € que sumándole los porcentajes de gastos indirectos del 296,22 € llega a 4527,96 €. EL 85% de los gastos totales corresponden a las horas internas destinadas al desarrollo del proyecto y un 7% a las amortizaciones de los dispositivos utilizados.

Por último, se puede finalizar recalcando que para el desarrollo del proyecto en su conjunto ha requerido de conocimientos en el ámbito de la electrónica, de la programación y del control.

13. Fuentes de Información. Bibliografía

[1] Wiring Pi (2018). *GPIO Interface library for Raspberry Pi.* Autor. Consultado el 10 de Julio de 2018.
http://wiringpi.com

[2] Youtube (24 de Diciembre de 2016). *How to acquire sensor value from MPU6050 with Simulink external mode simulation.* Shoshei.
https://www.youtube.com/watch?v=PXtXcsiJbGw

[3] Raspberry PI ++ (7 de Junio de 2013). *Librería wiring Pi.* Miguel Alvarez. Consultado el 11 de Julio de 2018.
http://rpiplus.blogspot.com/2013/06/libreria-wiring-pi.html

[4] Adafruit (9 de Febrero de 2016). *Raspberry Pi Analog to Digital Converters, MCP3008.* Tony DiCola. Consultado el 25de Junio de 2018.
https://learn.adafruit.com/raspberry-pi-analog-to-digital-converters/mcp3008

[5] Observatorio Tecnológico, Gobierno de España (7 de Junio de 2011). *Tipos de datos en C.* Cristina Villoria. Consultado el 5 de Julio de 2018.
http://recursostic.educacion.es/observatorio/web/en/software/programacion/972-tipos-de-datos

[6] Raspberry Pi Foro. *C code for MCP3008.* Consultado el 4 de Julio de 2018.
https://lb.raspberrypi.org/forums/viewtopic.php?t=54366

[7] MathWorks (2018). *Simulink Support Package for Raspberry Pi Hardware.* Autor. Consultado el 27 de Junio de 2018.
https://es.mathworks.com/help/supportpkg/raspberrypi/index.html

[8] MathWorks (2018). *Matlab Support Package for Raspberry Pi Hardware.* Autor. Consultado el 27 de Junio de 2018.
https://es.mathworks.com/matlabcentral/fileexchange/45145-matlab-support-package-for-raspberry-pi-hardware

[9] Automática y Control (2017-2018). *Análisis en el dominio del tiempo.* Marga Marcos, Nagore Iriondo, Itziar Cabanes, Asier Zubizarreta.

[10] Control por Computador (2018). *Control directo de Servomotor.* Xabier Basogain.

[11] MathWorks (2018). *Build a Digital Voltmeter.* Autor. Consultado el 25 de Junio de 2018.
https://es.mathworks.com/help/supportpkg/raspberrypiio/examples/build-a-digital-voltmeter.html

[12] Gordons Projects (Marzo de 2018). *wiringPi.* Autor. Consultado el 2 de Julio de 2018.
https://git.drogon.net

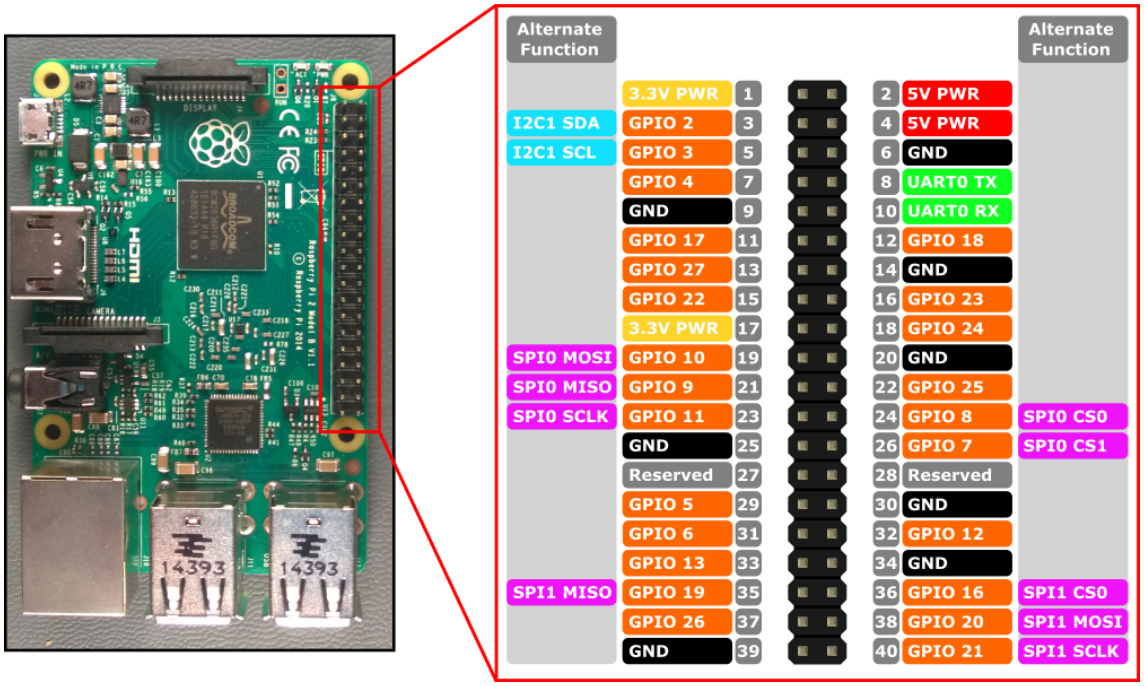
[13] PC Componentes (23 de Julio de 2018). *Especificaciones de Raspberry Pi 3 B.* Autor. Consultado el 23 de Julio de 2018.
https://www.pccomponentes.com/raspberry-pi-3-modelo-b

ANEXOS

Anexo 1: Especificaciones Raspberry Pi 3

- Procesador:
  - Chipset Broadcom BCM2387
  - 1,2GHz de cuatro núcleos ARM Cortex-A53
- GPU
  - Dual Core VideoCore IV Multimedia Co-procesador. Proporciona Open GL ES 2.0. OpenVG acelerado por hardware, y 1080p30 H.264 de alto perfil de descodificación
  - Capaz de 1 Gpixel/s, 1.5Gtexel/s o 24 GFLOPscon el filtrado de texturas y infraestructura DMA
- RAM: 1GB LPDDR2.
- Conectividad
  - Ethernet socket Ethernet 10/100 BaseT
  - 802.11 b / g / n LAN inalámbrica y Bluetooth 4.1 (Classic Bluetooth y LE)
  - Salida de Video
    - HDMI rev 1.3 y 1.4
    - RCA compuesto (PAL y NTSC)
  - Salida de audio
    - Jack de 3,5mm de salida de audio, HDMI
    - USB 4x conector USB2.0
  - Conector GPIO
    - 4-clavijas de 2,54mm (100 milésimas de pulgada) de expansión: 2x20 tira
    - Proporcionar 27 pines GPIO, así como 3,3V, +5V y GND líneas de suministro
  - Conector de la cámara de 15 pines cámara MIPI interfaz en serie (CSI-2)
  - Pantalla de visualización Conector de la interfaz de serie (DSI) Conector de 15 vías plana flex cable con de carriles de datos y u carril de reloj
  - Ranura de tarjeta de memoria Empuje / tire Micro SDIO

Anexo 2: Localización de SPI GPIO

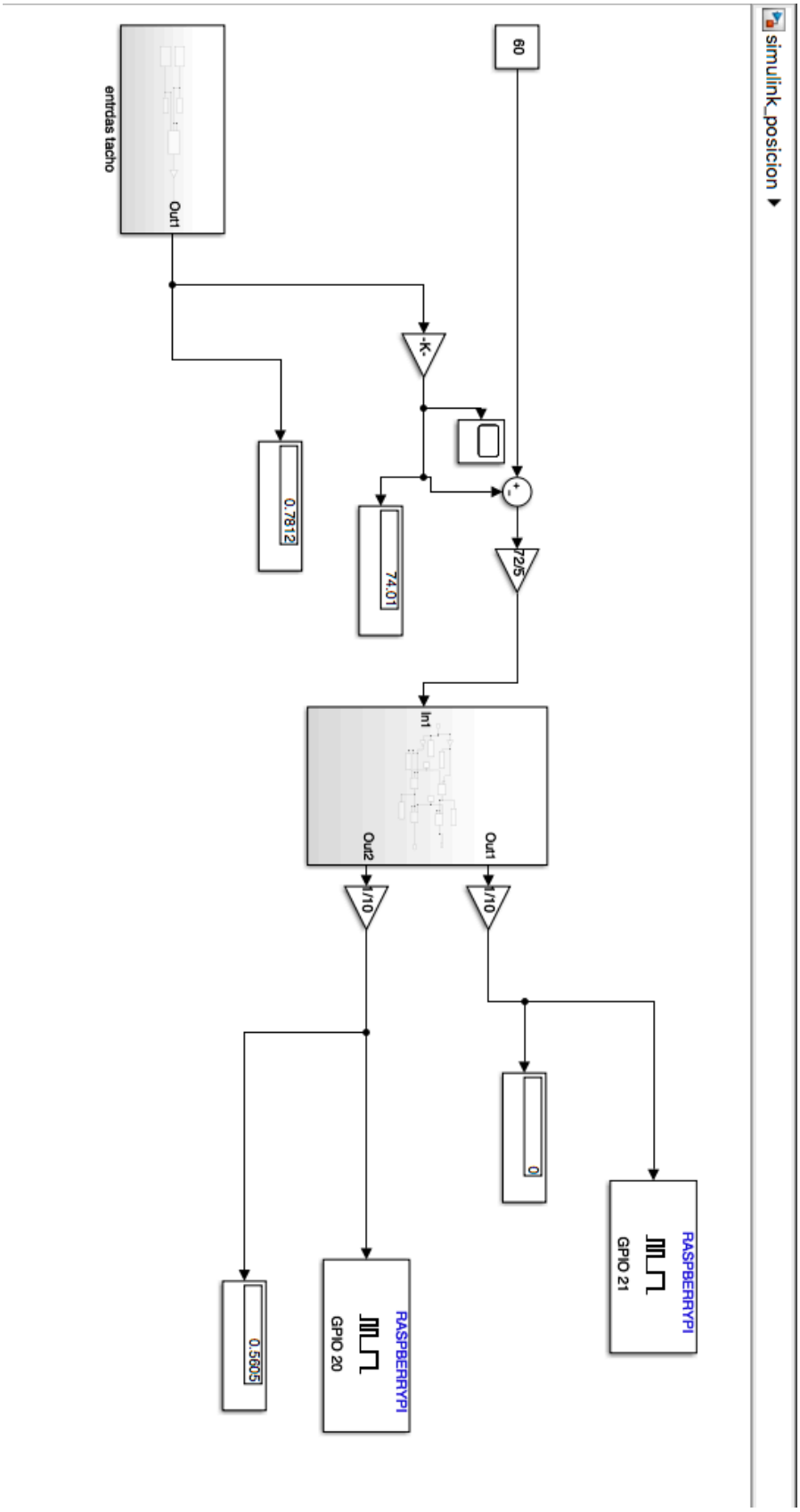


Anexo 3: Código en c del programa de lectura de señales

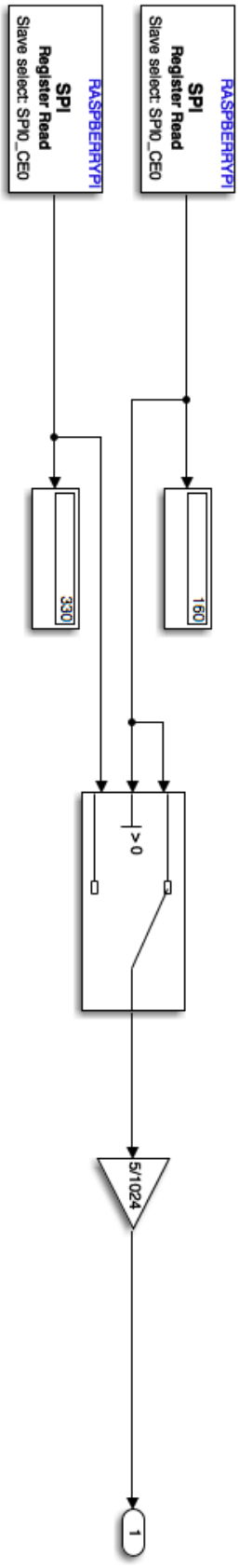
```
#include <wiringPi.h>
#include <mcp3004.h>
#include <stdio.h>
#include <stdlib.h>

#define BASE 100
#define SPI_CHAN 0

int main (void) {
    int chan;
    int x;
    int y;
    float tacho;
    float posicion;

    printf("Lectura de la posicion y velocidad del
servomotor\n");

    if(wiringPiSetup() == -1)
        exit(1);

    mcp3004Setup(BASE, SPI_CHAN);
    chan 0;

    for(;;) {
        x = analogRead(BASE + chan);
        if(x == 0)
            x = analogRead(BASE + chan + 2);

        tacho = x + (5.0 / 1024);

        y = analogRead(BASE + chan + 1);
        posicion = y * (5.0 / 1024);
        printf("%f\n", tacho);
        printf("%f\n", posicion);
        delay(1000);
    }

    return 0;

}
```

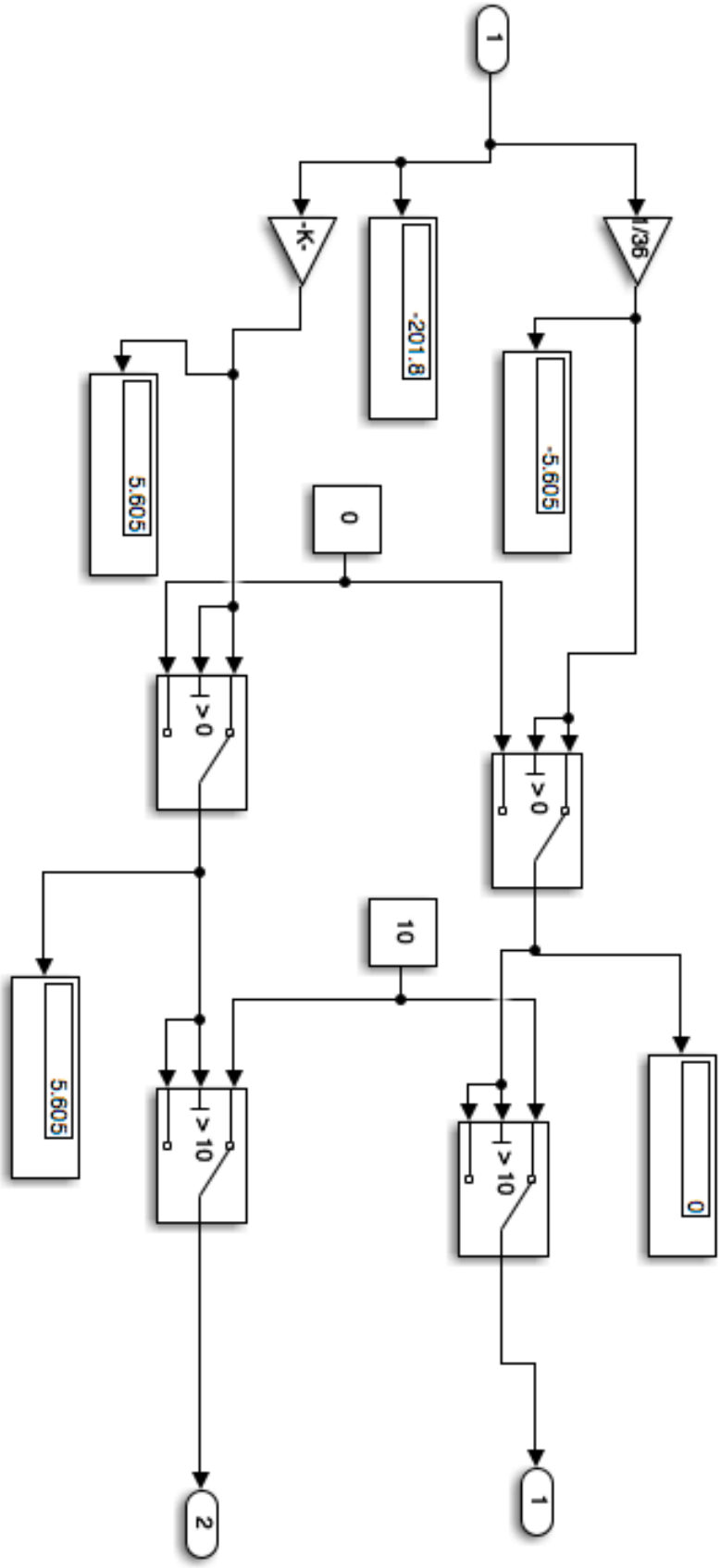
Anexo 4: Programas de Simulink

- Programa Principal



- Subprogramas




1
-K-
-201.8
1/36
5.605
-5.605
0
> 0
> 0
10
5.605
> 10
> 10
0
2
1