

GRADO EN INGENIERÍA EN TECNOLOGÍA
INDUSTRIAL

TRABAJO FIN DE GRADO

***PLACA ELECTRÓNICA Y SOFTWARE
DE CONTROL PARA UN BANCO DE
POTENCIA***

Alumno/Alumna: Lizarralde Bilbao Markel

Director/Directora: Zuloaga Izaguirre Aitzol

Curso: 2017-2018

Fecha: lunes, 23 de Julio de 2018

RESUMEN TRILINGÜE

RESUMEN

El propósito de este documento es presentar un proyecto que podría ser de utilidad para el equipo MotoStudent. Este proyecto consiste en el desarrollo de una placa electrónica y su correspondiente software para controlar un banco de potencia. Se pretende presentar la posibilidad de diseñar un banco de potencia de manera económica y así poder usarlo en la competición de MotoStudent. Se trabajan apartados tales como los objetivos, el alcance, estudio de alternativas y soluciones, todo el desarrollo técnico para la consecución del trabajo y las conclusiones.

ABSTRACT

The purpose of this document is to introduce a project that could be useful for the MotoStudent team. The project consists in the development of an electronic board and its corresponding software to be able to control a dyno test. The aim of this document is to present the possibility of designing a dyno test in an economical way to be able to use it in the MotoStudent competition. The proposed study is described in the following text. Related to the document, we will find out that it is divided in different parts, including introduction, context, goals and scope, working plan, resources, and conclusions, where we will find all the important information about the Project.

LABURPENA

Dokumentu honen helburua MotoStudent taldearentzat erabilgarria izan daitekeen proiektu bat aurkeztea da. Lan hau potentzia banku bat kontrolatzeko plaka elektronikoren baten eta honi dagokion software-aren garapenean datza. MotoStudent lehiaketan erabili ahal izango den potentzia banku ekonomiko bat diseinatzeko aukera aurkezteko asmoa dago. Garatuko diren atalen artean helburua, irispena, aukeren eta konponbideen azterketa, lana aurrera eramateko garapen tekniko osoa eta ondorioak aurkitzen dira.

PALABRAS CLAVE

Banco de potencia, banco de pruebas, placa electrónica, software, hardware, sensor, potencia, par motor, revoluciones, MotoStudent

1. Contenido

1. MEMORIA.....	8
1.1 Introducción.....	8
1.2 Contexto.....	8
1.3 Descripción del banco de potencia.....	9
1.4 Objetivos y alcance del trabajo.....	15
1.5 Beneficios que aporta el trabajo	15
1.6 Análisis de alternativas	16
Tipo de banco	16
Tipo de freno	17
Adquisición de datos	18
Software del banco.....	19
Medida de potencia y el par	21
Medida de las revoluciones en el rodillo del banco.....	21
1.7 Descripción de la solución propuesta	22
Tipo de banco.....	22
Tipo de freno	23
Adquisición de datos	25
Software del banco.....	25
Medida de la potencia y del par	26
Medida de las revoluciones en el rodillo del banco.....	27
Fuente de alimentación.....	27
2. METODOLOGÍA SEGUIDA EN EL DESARROLLO DEL TRABAJO	29
2.1 Descripción de tareas o fases y equipos.....	29
Fases del trabajo	29

Hitos	31
Equipo y recursos utilizados.....	32
2.2 Diagrama de Gantt.....	33
2.3 Cálculos.....	33
Frecuencia de muestreo de las revoluciones	33
Cálculo del par	35
Cálculo de la potencia	36
2.4 Esquema eléctrico de la placa de control.....	36
Control del freno de Foucault.....	37
Celda de carga y amplificación.....	40
Fuentes de alimentación.....	45
Sensor Hall.....	46
Pantalla LCD.....	46
Consigna de revoluciones deseadas.....	48
2.5 Diseño del circuito impreso	49
2.6 Diseño del Software de control.....	54
Programación del Arduino	55
Programación del ordenador.....	57
2.7 Puesta en marcha. Primeras pruebas.....	59
2.8 Futuras mejoras. Segunda versión	63
3. ASPECTOS ECONÓMICOS. PRESUPUESTO.....	67
4. CONCLUSIONES.....	68
5. REFERENCIAS	69
ANEXO 1. Diagrama de Gantt.....	71
ANEXO 2. Esquema eléctrico.....	72

Markel Lizarralde

ANEXO 4. Programa completo del arduino.....	75
ANEXO 5. Programa completo del Matlab.....	83
ANEXO 6. Presupuesto.....	95

TABLA DE ILUSTRACIONES

Ilustración 1. Tabla del mapa de inyección (4)	10
Ilustración 2. Representación gráfica del mapa de inyección	11
Ilustración 3. Representación del PMS y el ángulo de avance.....	12
Ilustración 4. Tabla del mapa de encendido	13
Ilustración 5. Representación gráfica del mapa de encendido	13
Ilustración 6. Diagrama de flujos con tarjeta de adquisición de datos.....	19
Ilustración 7. Diagrama de flujos usando un microcontrolador.....	20
Ilustración 8. Diseño mecánico del banco de potencia.....	23
Ilustración 9. Freno de Foucault Frenelsa (11).....	24
Ilustración 10. Fuente de alimentación del freno VAR-10-40F (12)	25
Ilustración 11. Celda de carga RSCC C3	26
Ilustración 12. Sensor Hall Hamlin 55075	27
Ilustración 13. Fuente de alimentación Traco Power	28
Ilustración 14. Diagrama de Gantt.....	33
Ilustración 15. Piñón y cremallera de la moto (17).....	34
Ilustración 16. Esquema representativo del funcionamiento de la celda	36
Ilustración 17. Esquema eléctrico del control del freno de Foucault.....	37
Ilustración 18. Señales simuladas en el osciloscopio.....	38
Ilustración 19. Señales simuladas en el osciloscopio.....	39
Ilustración 20. Señales simuladas en el osciloscopio.....	40
Ilustración 21. Ciclo de combustión de un motor de 4 tiempos (19).....	41
Ilustración 22. Par motor en función de la fase del ciclo del motor (20)	42
Ilustración 23. Puente de Wheatstone	42
Ilustración 24. Circuito de obtención de la señal de la celda de carga.....	43

Ilustración 25. Amplificador operacional en modo diferencial (21).....	44
Ilustración 26. Amplificador operacional no inversor	44
Ilustración 27. Esquema eléctrico de las fuentes de alimentación.....	45
Ilustración 28. Esquema eléctrico del sensor Hall	46
Ilustración 29. Esquema eléctrico de la conexión del LCD (22)	47
Ilustración 30. Relación de conexiones del LCD	48
Ilustración 31. Esquema eléctrico de la conexión del potenciómetro para la consigna.....	49
Ilustración 32. Diseño del PCB sin colocar los componentes	50
Ilustración 33. Diseño del PCB con los componentes colocados.....	51
Ilustración 34. Diseño del PCB con las pistas colocadas.....	53
Ilustración 35. Representación tridimensional del PCB.....	54
Ilustración 36. Diagrama de flujos del programa del arduino	55
Ilustración 37. Pantalla LCD	57
Ilustración 38. Entorno gráfico del programa del ordenador	57
Ilustración 39. Diagrama de flujo del programa del ordenador	58
Ilustración 40. Placa de control soldada	60
Ilustración 41. Montaje provisional del sensor Hall.....	61
Ilustración 42. Resultados experimentales del sensor Hall	61
Ilustración 43. Resultados experimentales de la celda de carga.....	62
Ilustración 44. Representación gráfica de los resultados experimentales de la celda de carga	63
Ilustración 45. Presupuesto resumido.....	67

1. MEMORIA

1.1 Introducción

Este documento contiene un proyecto de diseño e implantación de una placa de control, así como el software necesario para la utilización de un banco de potencia fabricado por el equipo de MotoStudent Bizkaia. En una primera parte se presenta el contexto del proyecto, la oportunidad del mismo y los beneficios técnicos y económicos que se esperan conseguir. Para ello, se estudiarán distintas alternativas y se valorarán ambas mediante diferentes criterios, eligiendo aquella opción que más se ajuste a las necesidades.

Una vez seleccionada la alternativa, se realiza una descripción en la que se muestran las diferentes partes de las que constará la placa electrónica y se da una breve explicación de cada una de ellas. También se detallarán todos los detalles del Software diseñado para su control.

A continuación, se planifican todos los trabajos a realizar, viendo las diferentes relaciones entre las tareas a llevar a cabo. Principalmente hay dos etapas, una en la que se realiza el diseño y fabricación de la placa electrónica y otra en la que se programa el Software necesario para controlarla.

Este proyecto cuenta también con un presupuesto detallado de todos los costes. En este último se enumeran las ventajas obtenidas, dando una explicación de las consecuencias de las mismas.

1.2 Contexto

Como alumno de la escuela participo voluntariamente en la quinta edición de la competición de MotoStudent (2017-2018) (1). MotoStudent es un desafío que nace de la Ciudad del motor de Aragón, consistente en una competición entre universidades de ingeniería de todo el mundo. El objetivo consiste en el diseño estructural de una motocicleta de competición de 250cc y su posterior construcción. Las distintas universidades se enfrentarán en la competición a unas pruebas técnicas durante unas jornadas que se llevan a cabo en las instalaciones de Motorland Aragón durante un fin de semana en el mes de octubre 2018, teniendo como prueba de fuego la carrera final de MotoStudent. La competición consiste en una serie de pruebas planteadas para evaluar el

Markel Lizarralde

comportamiento dinámico y las prestaciones de la moto fabricada, para comprobar la seguridad y funcionalidad, así como la destreza de los equipos en mecánica y organización.

No cabe duda en que la potencia de la moto sea un punto clave para obtener unos mejores resultados en la competición. La organización suministra a todos los equipos un motor idéntico para partir de una situación de igualdad. Queda prohibido por el reglamento de la competición la manipulación mecánica del motor, con lo que las únicas modificaciones posibles solamente serán electrónicas, esto es, la parte relacionada con la unidad de control del motor o ECU (siglas en inglés “engine control unit”) (2). La unidad de control regula, entre otras muchas cosas, el punto de ignición y la apertura/cierre de válvulas del motor. Estos conceptos serán abordados en los siguientes apartados.

Conocidos los datos proporcionados por la ECU y combinados con otros obtenidos por distintos sensores existe la posibilidad de mejorar la potencia del motor. Estos datos se pueden conseguir llevando la moto a cualquier circuito y se aumentaría la potencia variando algunos valores relevantes de la unidad de control hasta cumplir el objetivo. Sin embargo, este proceso resulta muy laborioso y casi una pérdida de tiempo. La solución a ello ha sido montar la moto en un banco de potencia y calcular estos valores de una manera mucho más rápida y a tiempo real. Actualmente, la universidad tiene un banco de potencia, pero no es lo suficientemente preciso como para realizar el trabajo necesario. A causa de ello, el equipo MotoStudent Bizkaia ha decidido diseñar y montar su propio banco de potencia. Esto requiere un gran trabajo tanto mecánico como electrónico y de programación. Este es el origen de este trabajo de fin de grado.

1.3 Descripción del banco de potencia

En este apartado se explica los conocimientos necesarios para el aumento de la potencia del motor, así como la descripción estructural y funcional del banco de potencia. Gracias a esto, en los siguientes apartados se podrá estudiar las alternativas posibles, llegar sus respectivas conclusiones y elegir la solución correcta. Para la correcta comprensión del funcionamiento del banco primeramente se explican los distintos métodos electrónicos de aumentar la potencia del motor. Estos son el ajuste del mapa de inyección y el ajuste del mapa de encendido (3). Ambos consisten en la obtención de unos parámetros de funcionamiento utilizando la interpolación. Se obtiene el parámetro a medir en distintos rangos de funcionamiento del motor y se interpola para todos los puntos posibles.

Markel Lizarralde

El ajuste del mapa de inyección dosifica la cantidad de combustible inyectado variando el tiempo de inyección en cada ciclo del motor. Este tiempo está ajustado en milisegundos. Para poder ajustarlo se parte del valor del TPS (sensor que marca el porcentaje de apertura de mariposa) y de las revoluciones del motor. Con estos dos parámetros se forma una tabla que recoge dicho tiempo en cada punto de funcionamiento. A continuación se muestra una tabla de inyección como ejemplo y su posterior representación gráfica. La representación resultante es tridimensional, ya que representa el tiempo de inyección a distintas revoluciones y aperturas de mariposa.

Fuel Main (% of IJPU)		RPM	0	250	500	1000	1500	2000	2500	3000	3500
Effcy %	35,0		32,9	33,0	33,1	33,2	33,4	33,6	33,7	33,9	34,1
	30,0		32,9	33,0	33,1	33,3	33,5	33,6	33,8	34,0	34,2
	25,0		31,9	31,7	31,6	31,2	30,9	30,5	30,2	29,9	29,5
	20,0		30,0	29,8	29,6	29,2	28,8	28,4	28,0	27,5	27,1
	15,0		28,0	27,8	27,6	27,3	26,9	26,5	26,1	25,8	25,4
	10,0		26,0	25,7	25,3	24,7	24,0	23,4	22,7	22,1	21,4
	7,0		23,2	22,4	21,7	20,2	18,7	17,2	15,7	14,2	15,0
	4,0		22,0	21,8	21,5	21,0	20,5	20,0	19,5	19,1	18,6
	2,0		20,3	20,1	19,9	19,5	19,1	18,7	18,3	17,9	17,5
	0,0	19,0	18,8	18,7	18,3	18,0	17,6	17,3	17,0	16,6	
Fuel Overall Trim (%)											
Fuel Trims (ms)											
La1 Aim Value											
Lambda Diff											

Ilustración 1. Tabla del mapa de inyección (4)

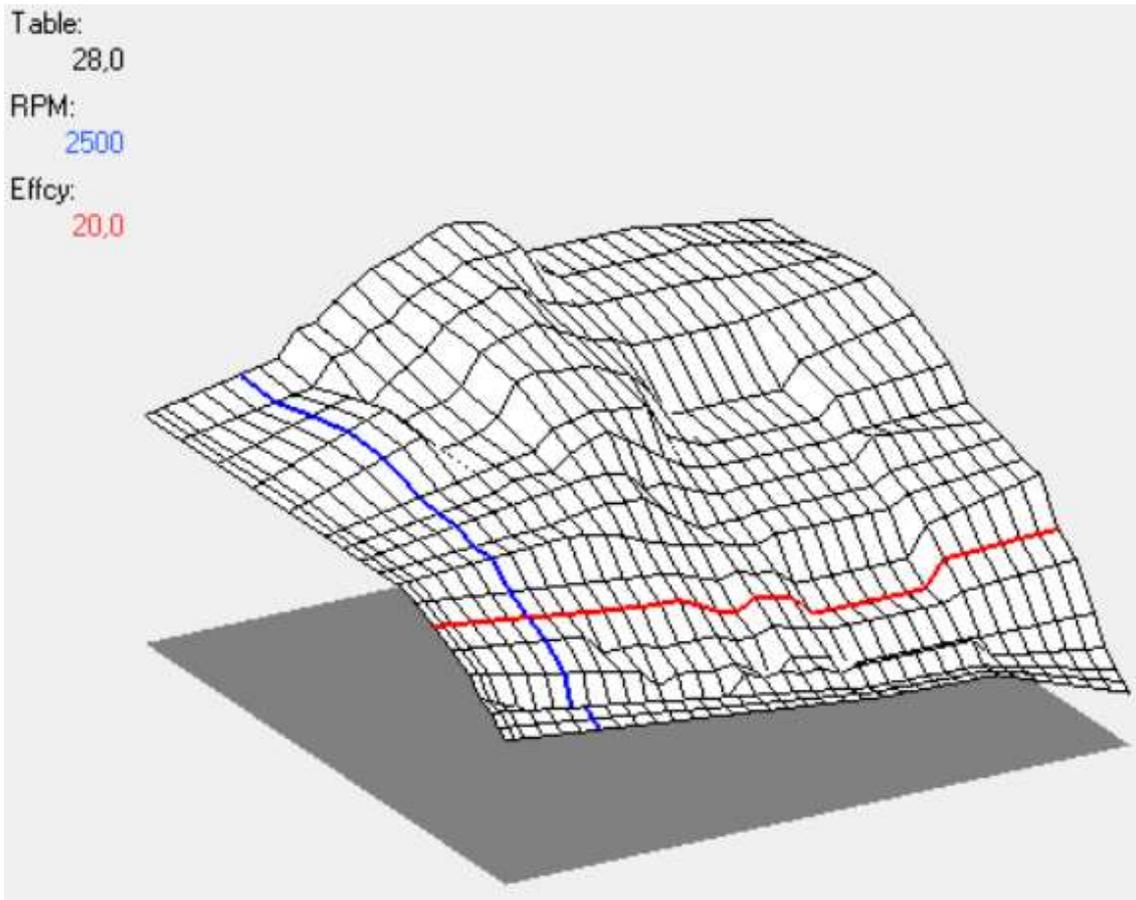


Ilustración 2. Representación gráfica del mapa de inyección

El ajuste del mapa de encendido depende, al igual que en el caso anterior, del régimen de giro y de la apertura de mariposa o carga del motor. También se puede representar gráficamente en un diagrama de tres ejes siendo el valor de cada punto de la gráfica el avance de encendido (5). El concepto de avance de encendido se explica a continuación.

Los motores de combustión encienden la mezcla de gasolina y aire a través de la chispa que lanza la bujía. En teoría, el momento ideal para lanzar la chispa es cuando el pistón se sitúa en la parte superior, con la mezcla completamente comprimida para lanzarlo con la mayor fuerza posible hacia abajo. Ese punto donde el pistón está lo más arriba posible es el PMS (punto muerto superior). El ángulo de avance de encendido se mide en grados de giro del cigüeñal antes del punto muerto superior.

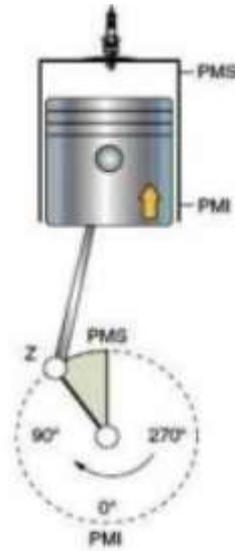


Ilustración 3.
Representación del
PMS y el ángulo de
avance

El problema surge en que el proceso de combustión lleva un determinado tiempo y, por lo tanto, si salta la chispa en el PMS, la explosión empujará el pistón hacia abajo cuando esté ya bajando, lo cual no es muy útil, y se desperdicia parte de la fuerza de la explosión.

Por ello se utiliza el avance de encendido, es decir, hacer saltar la chispa antes de que el pistón llegue al PMS. Hace falta una alta precisión ya que, si la chispa salta demasiado pronto la explosión empujará el pistón hacia abajo antes de alcanzar el PMS e intentará que el motor gire en dirección contraria a la habitual. Esta situación donde el avance de encendido es demasiado grande recibe el conocido nombre de “picar biela”. Las siguientes dos imágenes nos muestran la forma de un mapa de encendido cualquiera:

Ign Main (*BTDC)		RPM	0	250	500	1000	1500	2000	2500	3000	3500
Load %	70,0		11,5	13,1	14,7	17,8	21,0	24,1	27,3	30,4	33,6
	60,0		12,0	13,6	15,1	18,2	21,3	24,4	27,5	30,6	33,7
	50,0		12,5	14,0	15,5	18,4	21,4	24,3	27,3	30,2	33,2
	45,0		12,5	13,9	15,4	18,2	21,1	23,9	26,8	29,6	32,5
	40,0		13,0	14,4	15,8	18,5	21,3	24,0	26,8	29,5	32,3
	35,0		13,0	14,4	15,7	18,4	21,1	23,8	26,5	29,2	31,9
	30,0		13,5	14,8	16,2	18,8	21,5	24,1	26,8	29,4	32,5
	25,0		13,5	13,5	13,5	16,0	20,5	27,0	30,0	34,0	37,0
	20,0		14,0	14,0	14,0	16,5	21,5	28,0	30,0	34,0	37,0
	15,0		15,0	15,0	15,0	17,5	22,5	28,0	30,0	32,5	35,0
Ign Overall Trim Ignition Trims (dB TDC)											

Ilustración 4. Tabla del mapa de encendido

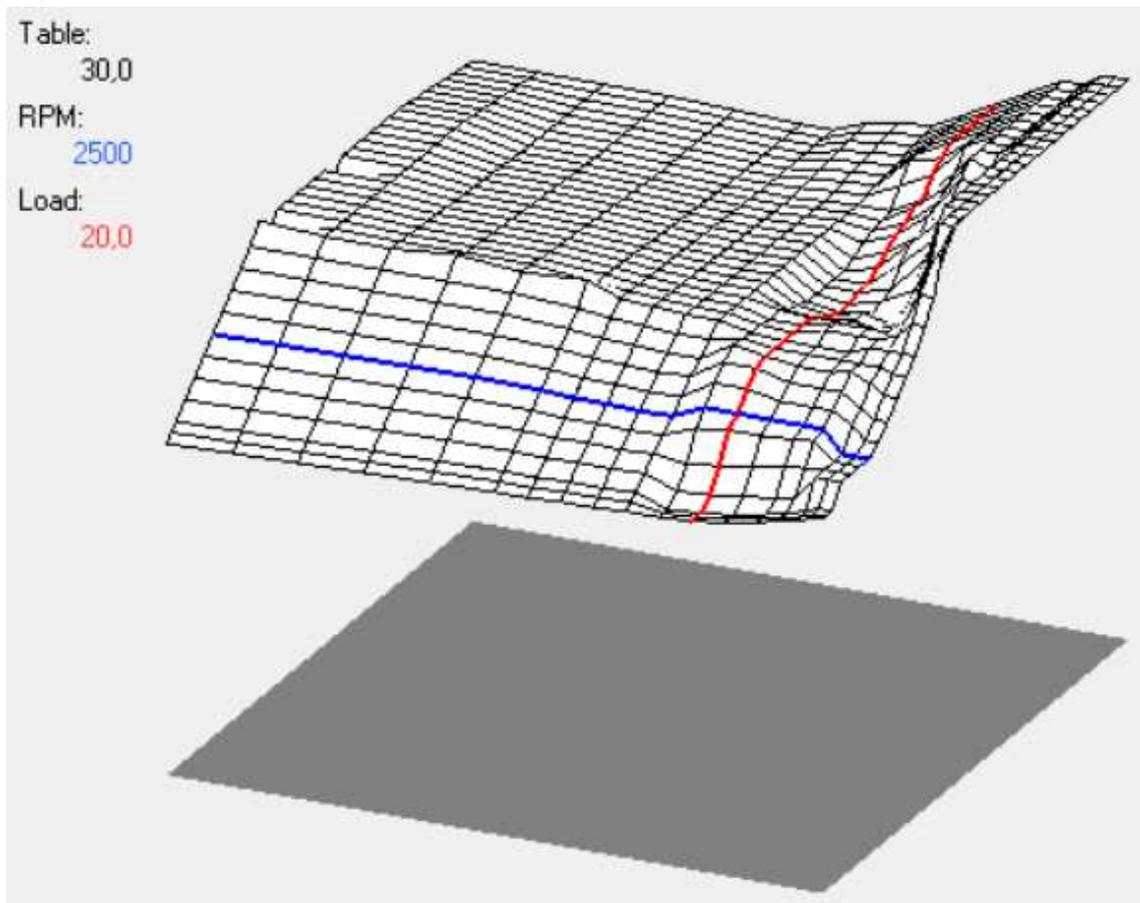


Ilustración 5. Representación gráfica del mapa de encendido

Ambas tablas son memorizadas por la centralita electrónica y son clave para gestionar el comportamiento ideal en cada punto de funcionamiento. Existen otras variables que se pueden modificar para optimizar el funcionamiento, pero las dos mencionadas anteriormente son las más importantes. Ahora que se conocen los datos que se deben optimizar se describirá el funcionamiento y la composición del banco de potencia y la utilidad que tiene.

Un banco de pruebas, o también llamado banco de potencia, es una estructura mecánica combinada con unos sensores capaces de medir variables importantes de los motores de combustión. Es decir, es una plataforma que permite medir de forma experimental magnitudes como el par motor, la potencia o las revoluciones del vehículo utilizado.

Esta plataforma está formada principalmente por una estructura que le da rigidez al banco, uno o varios rodillos (en función del número de ruedas motrices que tenga el vehículo) y los distintos sensores que permita medir las magnitudes necesarias. El vehículo se coloca sobre la estructura apoyando las ruedas motrices sobre los rodillos y se somete a una serie de pruebas en las que los sensores recogen continuamente los datos. Generalmente existen dos tipos diferentes de bancos de potencia: Los bancos inerciales y los bancos de freno. Se explicarán sus diferencias en el apartado de análisis de alternativas.

Ambos bancos son capaces de medir las tres magnitudes mencionadas: Las revoluciones, el par motor y la potencia. En los anteriores párrafos se ha explicado que los parámetros de ajuste de la centralita son el avance de encendido y el tiempo de inyección. Para poder relacionar estas variables, el modo de operar con el banco consiste en visualizar todas las variables a tiempo real mientras que se varía el avance de encendido y la inyección hasta que no logremos aumentar la potencia y el par motor. Una vez alcanzado el límite se sitúa la moto en otro punto de funcionamiento (apertura de mariposa y revoluciones diferentes) para seguir calculando nuevos puntos. Se deberá hacer esto con los máximos posibles y la centralita interpolará el resto de datos para asemejarlos lo mayor posible a la realidad. De esta manera se logra un motor funcionando a su máxima potencia en todos los casos posibles.

1.4 Objetivos y alcance del trabajo

A continuación se muestra tanto los objetivos de la fabricación del banco de potencia como como el alcance de dicho proyecto. El objetivo principal de este proyecto es el desarrollo de un sistema electrónico para la medición electrónica del par motor y la potencia de una moto en un banco de pruebas. En cuanto a objetivos secundarios podemos recalcar los siguientes:

- Desarrollar un sistema que permita regular la carga aplicada a la moto
- Medir el par motor y la potencia que está desarrollando la moto en cada instante con un error menor al 5%.
- Medir las revoluciones actuales de los rodillos del banco en contacto con la rueda de la moto con un error menor al 5%.
- Desarrollar el Software necesario para gestionar los datos recogidos por los sensores.
- Mostrar los valores obtenidos para poder contrastar las variaciones en la potencia del motor.
- Graficar los datos recogidos en un ordenador

El alcance de este proyecto parte desde el estudio de los distintos componentes de un banco de potencia hasta el montaje completo del mismo. Se va a diseñar una placa electrónica y un programa que permita utilizar el banco diseñado por otro compañero del equipo MotoStudent. Con el trabajo de las dos personas se fabricará un banco de potencia útil para el aumento de la potencia de las motos a utilizar en las futuras competiciones.

1.5 Beneficios que aporta el trabajo

Los beneficios a lograr con la realización de este proyecto son los siguientes:

El primero de todos, sin duda, obtener un motor funcionando a máxima potencia para conseguir una mejor posición dentro de la competición. La organización proporciona un motor KTM (es el motor del modelo KTM rc250), de una cilindrada de 250cc y una potencia de 23kW o, dicho de otra manera, casi 31 caballos de potencia (6) . Con un buen diseño y uso del banco de potencia se pretende obtener alrededor de 6 caballos extra. Esto significa un aumento de la potencia del 20% aproximadamente, cifra más que considerable ya que la potencia es una de las variables más importantes en competición.

Mirando el lado de aprendizaje, se estudiará y profundizará el funcionamiento de un banco de potencia. No se puede olvidar que, aunque sea una competición, el objetivo principal es que se trata de un proyecto educativo entre universidades. El planteamiento de fabricar un banco de potencia dentro de la universidad brindará de unos conocimientos importantes a los encargados de realizarlo, recurso valorado para el futuro de dichos estudiantes. Servirá de base para adquirir profundos conocimientos en el sector de los motores de combustión, las centralitas de los vehículos y, sobre todo, los bancos de potencia. De la misma manera, serán necesarios aprender muchos conceptos de electrónica y programación. Gracias a este y al resto de proyectos llevados a cabo dentro del equipo se logrará una experiencia laboral trabajando en algo real antes de haber finalizado el grado.

Con una buena gestión económica se podría ahorrar una cantidad elevada de dinero, ya que un banco de potencia que cumpla las necesidades planteadas puede llegar a costar unos 15000€. Como se detallará en el presupuesto, se estima que el coste del banco completo no supere los 9000€, siendo este un ahorro importante. Este dinero sobrante podría ser invertido en otras necesidades que surjan dentro del equipo siendo de ayuda para futuras inversiones u obtener un resultado en competición aún mejor.

1.6 Análisis de alternativas

A lo largo de este apartado se estudiará todas las alternativas posibles para diseñar y fabricar un banco de potencia. Se analizarán todos los aspectos funcionales, tanto mecánicos como electrónicos. Aunque el objetivo de este trabajo de fin de grado sea la parte electrónica y programación, la parte mecánica es tan importante como el resto ya que condiciona los sensores y cálculos a utilizar. Se comenzará analizando las alternativas en cuanto al diseño mecánico como el tipo de banco, inercial o de freno. Posteriormente se analizará la parte electrónica, sensores, programas, etc.

Tipo de banco

En la descripción del banco de potencia se han marcado dos tipos de bancos como los dos más importantes: Los bancos inerciales y los de freno (7).

Los bancos inerciales consisten en unos rodillos de masa e inercia conocidas que tienen la posibilidad de girar libremente sobre su propio eje. El programa informático interpreta las aceleraciones y velocidades del rodillo y convierte estos datos en gráficas de potencia o

Markel Lizarralde

par en función de las revoluciones. Es un sistema muy sencillo y de fácil cálculo, con lo que son muy prácticos. Su gran ventaja respecto a los bancos de freno (se explicarán a continuación) es que no es necesario calibrarlos. Esto se debe a que las características físicas del rodillo como la masa y la inercia siempre tendrán el mismo valor. Sin embargo, como el cálculo de la potencia y del par depende de la aceleración del rodillo, este no puede circular a una velocidad constante. Esto implica que el acelerador del vehículo casi siempre tendrá que estar al máximo y no se tiene tanto margen de cálculo como en los bancos de freno. No se pueden utilizar para medir en todos los rangos de funcionamiento del motor.

El banco de freno consta de un rodillo de mínima inercia adosado a un sistema de absorción de energía (un freno que puede ser hidráulico o electromagnético). Este freno, controlado mediante software, permite estabilizar la moto al régimen de giro que deseemos durante un largo tiempo. Independientemente de ese freno que estabiliza el motor, un banco puede montar otro freno sobre el rodillo, con el exclusivo fin de detener el giro del rodillo y de la rueda de la moto. Al poseer una fuerza de frenado variable son capaces de simular condiciones prácticamente reales. En los bancos inerciales solo se pueden medir las variables a máxima aceleración, mientras que en los de freno se puede medir a distintas cargas. Requieren una calibración cada cierto tiempo y son menos precisos que los anteriores, pero tienen un margen de maniobra mucho más amplio. Esto permite realizar mapeos de motores mucho más completos y con una comodidad mayor.

Tipo de freno

En el caso de que el banco escogido sea inercial, el único freno necesario sería de emergencia en caso de querer parar el vehículo. Como ya se ha explicado antes, el inercial no tiene un freno que le permita regular la carga.

Si el banco es del tipo de freno se debe escoger el freno a utilizar (8). La función de estos frenos es proporcionar un par resistente variable al motor, de manera que se ensayen distintas condiciones operativas. Este puede ser de fricción, hidráulico, o eléctrico.

El freno de fricción mecánico por zapata y tambor fue el primero utilizado. Sin embargo, debido a su inestabilidad y dificultad de regulación y refrigeración, actualmente se opta por otras opciones.

Markel Lizarralde

El freno hidráulico se compone de un rotor y una carcasa o estator llena de agua que sirve tanto de elemento frenante como refrigerante. La potencia del motor absorbida por el freno se transforma en calor necesitándose una alimentación continua de agua fría.

El par de frenado de los frenos dinamométricos hidráulicos es aproximadamente proporcional al cuadrado del número de revoluciones (curva característica de respuesta aproximadamente cuadrática), lo que les hace muy estables.

Las ventajas de este tipo de freno son el bajo costo para potencias absorbidas importantes, gran duración y reparación rápida y poco costosa. Como inconvenientes se tiene que ofrece poca versatilidad de las curvas de par resistente y un par de frenado fuertemente dependiente de la presión de la red hidráulica, lo que puede producir inestabilidad.

En cuanto a los frenos eléctricos, si se acopla un motor térmico a una dinamo conectada a una resistencia eléctrica, se puede determinar con un voltímetro y un amperímetro la potencia eléctrica suministrada por la dinamo. En este método debe tenerse en cuenta que existirán pérdidas por rozamiento, por efecto del aire y pérdidas eléctricas dependientes de la carga en el generador, llevando a que la medida no sea muy precisa. Esto hace que sea mucho más común medir la potencia del motor indirectamente a través del par motor.

Igual que en los frenos hidráulicos, el estator posee un montaje basculante y está unido a un sistema de medida de fuerza. El par motor se transmite del rotor (inducido) al estator (inductor en anillo) por medio del campo magnético.

La regulación de la carga cuando las variaciones no son demasiado grandes puede hacerse variando la excitación del estator. De esta manera crece la reacción electromagnética entre el rotor y el estator, aumentando la carga resistente. La corriente producida puede ser disipada en forma de calor en unas resistencias eléctricas.

Adquisición de datos

Los datos obtenidos del sensor de revoluciones y el de potencia deben ser recogidos por alguna tarjeta para después analizarlos. Esta tarjeta debe ser capaz de muestrear todos los datos a una frecuencia superior a la máxima demandada por el banco. Existen dos opciones posibles: Utilizar una tarjeta de adquisición de datos conectada al ordenador o un microprocesador.

La tarjeta de adquisición de datos solo se encarga de muestrear los valores de los sensores a la frecuencia programada y enviar las señales obtenidas al ordenador. Posteriormente, el ordenador realiza los cálculos necesarios y muestra los valores en pantalla. De esta manera se aprovecha la potencia de cálculo disponible en un ordenador

En el caso del microprocesador, la diferencia reside en que es este quien hace los cálculos y no el ordenador. Una vez calculadas las revoluciones, potencia y par motor los muestra en una pantalla LCD o en un ordenador comunicado por USB. No es obligatorio el uso de un ordenador ya que este solo se utiliza para mostrar los datos.

Actualmente, los microprocesadores más utilizados son Raspberry Pi y Arduino. Una Raspberry Pi es una mini computadora con una alta capacidad de procesamiento comparándola con el Arduino. Sin embargo, Arduino posee una mayor sencillez a la hora de programarlo y conectarle sensores. Arduino tiene, en algunos modelos, muchísimos más pines que las Raspberry Pi. Los dos tipos de microprocesadores cuentan con pines digitales de entrada y salida, pero solo el Arduino tiene pines analógicos. En cuanto al precio, generalmente las Raspberry Pi son más caras que los Arduino.

Software del banco

El software del banco depende plenamente del elemento elegido para la adquisición de datos. En el caso de escoger una tarjeta de adquisición de datos, el software que controla el banco será un programa de ordenador. Esta tarjeta podría controlarse mediante Matlab, Simulink, Labview o algún otro Software. Tienen distintos entornos de programación, siendo unos de puro texto (Matlab) y otros gráficos mediante diagramas de flujo (Simulink y Labview). Las tarjetas de adquisición de datos se podrían programar de las dos maneras en el programa que se eligiera. A continuación se puede ver un diagrama de flujos del funcionamiento:

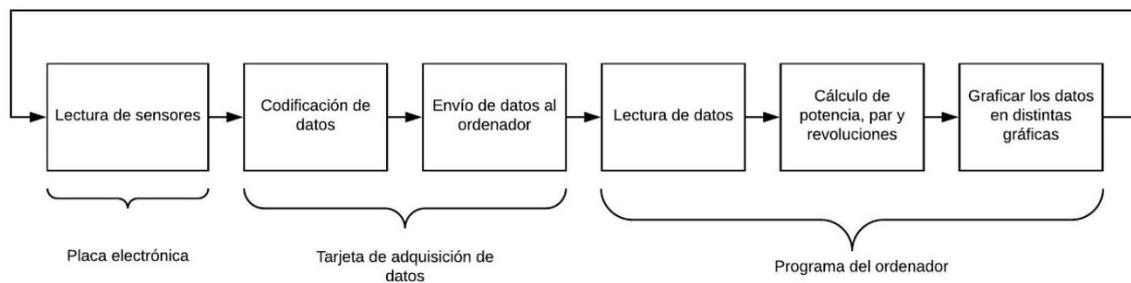


Ilustración 6. Diagrama de flujos con tarjeta de adquisición de datos

Si se escoge un Arduino o una Raspberry Pi se deberá programar el microprocesador de manera que ejecute el algoritmo que se le programe. Existe la posibilidad de mostrar los valores de tres maneras distintas: Únicamente con una pantalla LCD, únicamente graficándolo en el ordenador o una combinación de ambas. En el caso de que se utilice el ordenador para graficar los datos también habrá que diseñar un programa que se ejecute en el ordenador. La única diferencia que habrá respecto a utilizar una tarjeta de adquisición de datos estará en que el cálculo de la potencia, par y revoluciones lo hará el microprocesador y no el ordenador. En el siguiente diagrama se puede ver las diferencias entre el microprocesador y la tarjeta de adquisición de datos:

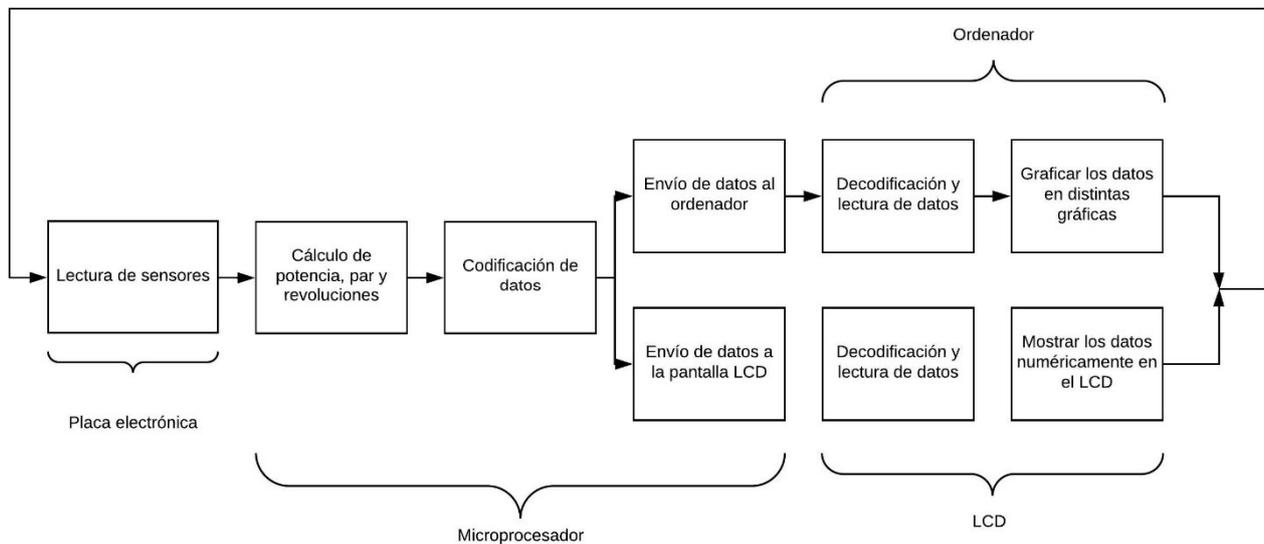


Ilustración 7. Diagrama de flujos usando un microcontrolador

Al final del diagrama hay dos ramas paralelas. Este es el caso en el que se envían los datos a una pantalla LCD y al ordenador para que sean graficados simultáneamente. De esta manera se podrá elegir el modo de visualización de las variables ya que se tiene dos métodos distintos. En caso de no conectar el ordenador seguirá estando la pantalla LCD y viceversa. Lo único que hace el ordenador es graficar los datos que ya se tiene numéricamente, de manera que las variaciones de par motor, potencia y revoluciones quede más visual y claro.

Medida de potencia y el par

La elección del sensor para la medida de la potencia y par motor depende principalmente del tipo de banco utilizado. En el caso del banco inercial no es necesaria la utilización de un sensor para dicho fin ya que la potencia y el par se calculan a partir de las revoluciones, masa e inercia del rodillo.

Si se opta por un banco de freno se necesita un sensor que mida la potencia y par. Para ello se utilizan las celdas de carga (9). El motivo por el que se utilizan este tipo de sensores se explica en el apartado de cálculo de la potencia y par.

Una celda de carga convierte una fuerza aplicada sobre ella en una señal eléctrica. Son el tipo de sensor de fuerza más usado en la actualidad. La señal generada puede ser neumática, hidráulica o eléctrica. También se pueden diferenciar en función del tipo de carga al que estén sometidas: flexión, tracción, compresión, cizalladura, etc.

Las celdas de carga eléctricas dan una señal de un voltaje proporcional a la carga aplicada, sea del tipo que sea. Son las celdas más utilizadas ya que permiten medir de una manera muy precisa las cargas aplicadas sobre ella. Generalmente la señal proporcionada por estos sensores suele ser del rango de unos mV, con lo que suelen necesitar una etapa de amplificación de la señal para la correcta lectura de la misma.

Medida de las revoluciones en el rodillo del banco

La manera más típica de medir las revoluciones de cualquier eje es utilizando una rueda fónica y un sensor; La rueda fónica es un disco dentado metálico que sabiendo el número de dientes que tiene permite saber a qué revoluciones se encuentra el banco en función de la señal del sensor. Este sensor será un captador que deberá detectar los dientes cada vez que pase uno a su lado. En función de la frecuencia a la que detecte los dientes se calculan las revoluciones. Se puede elegir entre dos tipos de captadores: Capacitivos e inductivos.

Los sensores capacitivos son un tipo de sensor eléctrico; reaccionan ante metales y no metales que al aproximarse a la superficie activa sobrepasan una determinada capacidad. La distancia de conexión respecto a un determinado material es tanto mayor cuanto más elevada sea su constante dieléctrica.

Un sensor inductivo (10) tiene internamente un generador de campo magnético y una bobina inductora que detecta el campo generado por el propio sensor. La presencia de un

objeto metálico modifica el campo inducido. Este campo es detectado por la electrónica del sensor, cuya salida se activa cuando un metal es detectado.

Un factor a tener en cuenta es el rango de revoluciones en el que girará el rodillo del banco. Si las revoluciones son mayores que la capacidad de muestreo del propio sensor este no será capaz de medir un valor real. Se deberá escoger un sensor con una frecuencia de muestreo mayor a la del régimen de uso independientemente del tipo de sensor que sea.

1.7 Descripción de la solución propuesta

En los siguientes puntos se detallan las opciones elegidas entre las alternativas mostradas en el apartado anterior. También se detallan los modelos y especificaciones técnicas de cada uno. El coste de los componentes y el total del proyecto se mostrará en el apartado de presupuesto.

Tipo de banco

Entre los dos tipos de bancos disponibles, el inercial o el de freno, se ha optado por el de freno. Como ya se ha explicado antes, el banco de freno tiene la opción de regular la carga a la que se somete el motor. De esta manera se podrá mapear todos los puntos de funcionamiento del motor, que es lo que se necesita para la competición. El diseño mecánico queda a cargo de un compañero de equipo y es el siguiente:

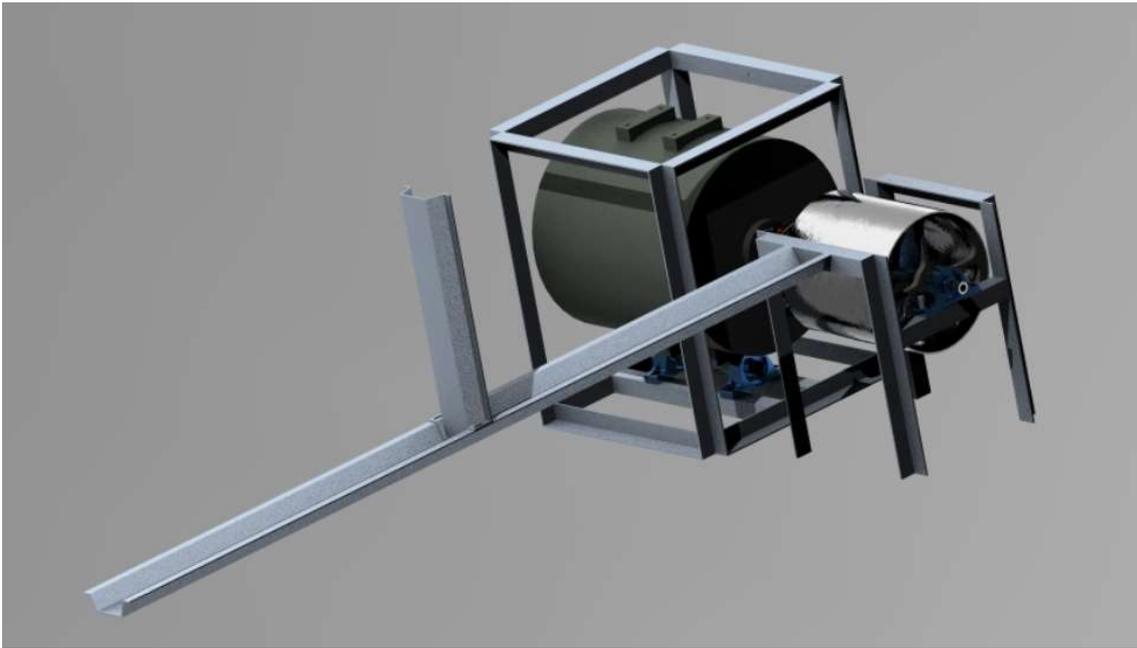


Ilustración 8. Diseño mecánico del banco de potencia

Como se puede ver, la estructura se puede dividir en tres partes importantes: La estructura que sujeta el freno (el cuadrado que tiene el freno en el interior), el soporte del rodillo y el de la moto. El freno y el rodillo están acoplados con un eje y un acoplamiento flexible. En ese eje es donde se colocará la rueda fónica para la medida de las revoluciones del rodillo. La celda de carga irá amarrada a la parte superior del cubo por un lado y a la parte superior del freno por el otro.

Las estructuras del freno y del rodillo serán tapadas con unas chapas de aluminio para una correcta protección de las personas que estén cerca. Se aprovechará una de las chapas para montar todo el cuadro eléctrico de la placa de electrónica, fuente de alimentación, etc.

En cuanto a la parte de amarre de la moto queda trabajo por hacer, pero no se entrará en detalle ya que es la parte que le corresponde al compañero.

Tipo de freno

Se ha optado por un freno eléctrico de Foucault de 100cV de potencia. Utiliza la inducción electromagnética para crear el fenómeno conocido como corriente de Foucault. Las corrientes de Foucault crean una resistencia que es aprovechada para frenar el vehículo. Durante el proceso de frenado un electroimán genera un campo electromagnético alrededor de las ruedas de metal que se encuentra en rotación generando las mencionadas corrientes. Estas corrientes encuentran resistencia a su paso por el metal disipando la energía en forma de calor y haciendo que las ruedas disminuyan de velocidad. Para controlar la fuerza de frenado y estabilizar las revoluciones se controla el campo electromagnético creado por el electroimán.



**Ilustración 9. Freno de Foucault Frenelsa
(11)**

La fuerza de frenado depende linealmente de la tensión alimentada. El freno funciona en un rango de voltaje entre 0 y 193Vdc y con un consumo de hasta 30A. Para alimentar este freno se necesita una fuente de alimentación regulable. Se ha optado por el modelo VAR-10-40F, una fuente regulable con un rango de voltaje de salida entre 10 y 330Vdc proporcional a una señal control entre 0 y 10Vdc. La fuente se alimenta desde una toma de red. No está galvánicamente aislada, con lo que habrá que tenerlo en cuenta a la hora de diseñar la parte electrónica del control del freno.



**Ilustración 10. Fuente de
alimentación del freno VAR-10-40F
(12)**

Adquisición de datos

Se ha optado por elegir un microprocesador Arduino, principalmente porque es el sistema que mejor conozco de anteriores proyectos que he realizado. Además, para este proyecto resulta mucho más útil que una Raspberry Pi ya que se necesita al menos un pin analógico para la medida del voltaje de la celda de carga. Esto no es posible con las Raspberry ya que, como se ha dicho en el apartado de análisis de alternativas, no tienen pines analógicos.

A la hora de elegir el microprocesador se pueden encontrar diversos modelos dentro de la empresa Arduino. Se necesita que el micro tenga la mayor capacidad de cálculo y precisión. Por ese motivo se ha optado por la placa Arduino "Due" (13), siendo la más potente de la empresa. Tiene un reloj que funciona a una frecuencia de 84MHz cuando el resto de modelos suelen funcionar a unos 16MHz, dando lugar a que el Due funciona unas 5 veces más rápido. En cuanto a los pines PWM y analógicos puede trabajar con una resolución de 12bits, a diferencia del resto que trabaja a 8bits o 10bits. Esto mejora en 4 o 16 veces la resolución obtenida en las mediciones.

Software del banco

Como el microprocesador elegido es de la marca Arduino, la parte de código del Arduino se hará con su software oficial (arduino IDE) ya que cumple las necesidades. En cuanto al programa que muestre los datos gráficamente en el ordenador se ha elegido Matlab con su herramienta AppDesigner, ya que se aprende a usar Matlab en algunas asignaturas de la

escuela y AppDesigner es una herramienta visual del mismo. Además, existe todo tipo de tutoriales en internet que sirven de ayuda para adquirir los conocimientos necesarios.

Medida de la potencia y del par

Para la medida de la potencia y el par se ha optado por una celda de carga con una carga máxima de 100kg. Su nombre es RSCC C3 (14). Esta celda estará amarrada a la estructura del banco y cuando la moto esté en funcionamiento empujará de ella de manera que, con unos cálculos sencillos, podremos calcular el par motor y la potencia. Según su Datasheet, esta celda da una señal proporcional a la carga aplicada.



Ilustración 11. Celda de carga RSCC C3

Tiene una sensibilidad de 2mV/V, con lo que alimentado a 5V se obtiene un rango de medición de 10mV. Este voltaje conviene que sea el máximo, siempre y cuando se respete el voltaje nominal, para que el rango medible sea el mayor posible. Arduino Due tiene una resolución de 12 bits en las entradas analógicas y opera a 3.3V. Esto quiere decir que tiene una sensibilidad de medición de:

$$\text{Sensibilidad} = \frac{3300mV}{4096} = 0.806mV$$

Una sensibilidad de 0.806mV trabajando entre 0-24mV no es lo suficientemente preciso como para que sea aceptable. Por lo tanto, tal y como se explicará en el apartado del esquema eléctrico, la medición de la celda de carga necesitará un circuito que amplifique la señal para que el rango de la medición en el Arduino sea mayor.

Medida de las revoluciones en el rodillo del banco

La elección ha sido un sensor de Hall. Este sensor irá conectado a una entrada digital del Arduino y dará como señal LOW cuando no esté detectando nada y HIGH cuando esté detectando un diente metálico.

El modelo elegido es un HAMLIN 55075 (15). Se puede alimentar de -25 a 25Vdc y tiene una frecuencia de muestreo máxima de 15kHz. Estos datos se han obtenido de su correspondiente datasheet.



Ilustración 12. Sensor Hall Hamlin 55075

Fuente de alimentación

Como la fuente regulable del freno no está galvánicamente aislada, todo el sistema tiene que ser alimentado por dos fuentes distintas. Una fuente de alimentación deberá suministrar energía a la parte del control del freno y la otra fuente se encarga del resto del circuito. En el caso de alimentar todo el sistema con una sola tierra las diferentes tensiones de masa se conectarán entre ellas y crearán un cortocircuito que probablemente quemará los componentes electrónicos. Utilizando las dos fuentes, la parte de control del freno queda conectado a la tierra de la red mientras que el resto del circuito queda flotante.

El circuito completo tiene que ser alimentado con tensiones positiva y negativas. El motivo de estas tensiones de doble polaridad se explica en el apartado del esquema eléctrico. En consecuencia, cada fuente necesita 3 terminales de alimentación: +12V, -12V y su tierra. Estas fuentes no pueden tener una tensión menor ya que el freno se controla con una señal

de 0 a 10V, con lo que, como mínimo, la fuente que alimente la parte del freno debe tener 12V. Por simplicidad se utiliza otra fuente idéntica para la otra parte del circuito.

El modelo de fuente elegido para los dos casos es TRACO POWER TMM 24 Series (16). Tienen una potencia de 24W, o lo que es lo mismo, una intensidad de 2A a 12V. Potencia suficiente ya que el Arduino con toda su electrónica no consumirá más de medio amperio. Este consumo es tan bajo porque la parte que alimenta el freno de Foucault con corrientes de hasta 30A es el antes comentado VAR-10-40F. Las fuentes son las siguientes:



Ilustración 13. Fuente de alimentación Traco Power

2. METODOLOGÍA SEGUIDA EN EL DESARROLLO DEL TRABAJO

2.1 Descripción de tareas o fases y equipos

En el presente capítulo se describe la planificación del proyecto que comienza el 8 de septiembre de 2017 y se estima que finalice el 20 de Julio de 2018 (es el tiempo total disponible en el cuarto curso de la carrera). En primer lugar, se realiza un desglose de las distintas actividades o fases y hitos que componen el trabajo. Después se detallan los equipos o recursos técnicos utilizados.

Fases del trabajo

A continuación se describen los distintos paquetes de trabajo en los que se descompone el proyecto.

Búsqueda de información

En esta fase o plan de trabajo se realizará la búsqueda de información necesaria para adquirir los conocimientos en las diferentes fuentes bibliográficas disponibles. Es el primer paso para el desarrollo del proyecto.

Análisis de alternativas

Una vez encontrada la información y obtenida una base sólida de conocimientos en el tema se procede a estudiar las distintas alternativas existentes para la realización del proyecto. Se busca información sobre los distintos sensores, tipos de banco, tarjeta de adquisición de datos, etc. Es un trabajo en equipo con el compañero encargado de la parte mecánica.

Desarrollo del esquema eléctrico

Una vez elegidas las soluciones entre todas las alternativas se sabe cuáles son los sensores electrónicos que se van a utilizar. Se comienza a diseñar el esquema eléctrico de la placa de potencia en el programa Multisim para posteriormente diseñar el circuito impreso. En esta etapa se trata de dejar el esquema lo más invariable posible para no tener que volver atrás en los siguientes apartados.

Desarrollo del circuito impreso

Cuando el esquema eléctrico está completamente acabado se empieza con la tarea del desarrollo del circuito impreso. Es la etapa en la que se pasa del diseño esquemático al diseño real de la placa. Todos los componentes se colocan sobre la placa electrónica del tamaño deseado y se hacen todas las pistas que los conectan.

Envío del circuito impreso

Tras haber terminado de diseñar el circuito completo se encarga a un fabricante de circuitos impresos para que nos traiga el prototipo. Es una fase en la que habrá que esperar a que la placa llegue y el tiempo depende de los fabricantes contratados.

Desarrollo del programa de arduino

Se diseña el programa necesario para controlar el arduino. De esta manera el microcontrolador será capaz de leer los valores de los distintos sensores de la manera que nosotros lo programemos. Este programa se encargará también de calcular la potencia, par motor, revoluciones y enviárselo tanto a la pantalla LCD como al ordenador.

Desarrollo del programa de Matlab

Para mostrar los valores en el ordenador diseñamos el programa con AppDesigner que permita leer los datos procedentes por USB del arduino. Estos datos son decodificados y graficados en sus correspondientes gráficas.

Búsqueda de los materiales y componentes

Tras terminado el diseño completo de los programas y el esquema eléctrico no habrá cambios en los componentes que se vayan a pedir. Por lo tanto, se buscan los componentes electrónicos, materiales y sensores necesarios para realizar todo el montaje. Se buscan distribuidores para obtener los materiales al mejor precio posible. Es una tarea que requiere una comunicación con distintas empresas para conseguir el mejor trato.

Envío del material

En esta fase toca esperar a que los componentes lleguen al taller. Al igual que con el encargo del circuito impreso, la duración de esta fase depende de la velocidad a la que trabajen los distribuidores con los que se contacte.

Pruebas de sensores por separado

Cuando todo el material haya llegado al taller se comienza a probar todos y cada uno de los sensores por separado para garantizar el correcto funcionamiento. En caso de haber fallos en alguno de ellos podrá ser muchísimo más fácil detectarlo que montándolo todo a la vez.

Puesta en marcha del banco

Tras haber comprobado que todos los componentes funcionan y que la placa mide las variables como debería se hace el montaje completo del banco. La primera versión completa del banco de potencia es la que se monte en esta fase.

Primeras pruebas

Se hacen las pruebas reales de funcionamiento utilizando una de las motos que haya en el taller. Se comprueba que el arduino recoge los datos, los envía y el ordenador los muestra correctamente. Se comparan los resultados con los esperados y se busca solución a los fallos en caso de que los haya. Las dos fases de puesta en marcha del banco y las primeras pruebas dependen del trabajo que haya hecho el otro compañero del equipo, ya que si no ha acabado su parte estas no se podrán cumplir en el plazo.

Redacción del trabajo

Esta fase se irá trabajando a lo largo de todo el año, a medida que el proyecto vaya avanzando. Se anotará las cosas más importantes del trabajo que se vaya realizando para una mejor redacción final del trabajo.

Hitos

El cumplimiento de estos hitos es fundamental para poder llevar a cabo el proyecto de fin de grado. Los hitos marcados son los siguientes:

Elección del trabajo de fin de grado

Es el primer hito y el más importante. Simboliza que para esa fecha el tema del trabajo de fin de grado ya habrá sido escogido y que se comienza a trabajarlo.

Elección de la solución

Es el momento en que tras profundizar en los conceptos teóricos se ha escogido entre cada una de las alternativas estudiadas. Llegado a este punto se empieza a diseñar tanto el esquema eléctrico como el Software del Arduino. Se deja la teoría y el estudio un poco de lado para pasar a la práctica.

Encargo del circuito impreso

Es el hito que marca que el circuito completo ya ha sido diseñado y puede encargarse a una empresa para su fabricación. Es el ecuador del proyecto y el trabajo que sigue a continuación será buscar los componentes eléctricos, pruebas reales, puesta en marcha, etc.

Compra del material

Tras haber contactado con distintos distribuidores se hace la compra para conseguir los componentes electrónicos, sensores y demás cosas necesarias. Todo el trabajo que queda después de este hito es mano de obra pura montando todo el banco.

Entrega del trabajo

Es el último hito que marca que el trabajo ha sido terminado cumpliendo cada una de las fases planificadas. Significa que el proyecto ha sido redactado completamente y está listo para ser entregado.

Equipo y recursos utilizados

Para la completa realización del programa se han utilizado 4 softwares distintos: Arduino IDE para la programación del Arduino, la herramienta AppDesigner de Matlab para la programación del ordenador, el programa Multisim junto con la extensión del diseño de circuitos impresos Ultiboard para la placa de control y el MS Project para la planificación y el Gantt.

A excepción del Arduino IDE, que es gratuito, y el Matlab y MS Project que está a disponibilidad de los alumnos en los ordenadores de la escuela, Multisim pide una licencia de pago. La escuela no dispone de licencias y no existe ninguna licencia gratuita para estudiantes. El coste de esta queda reflejado en el presupuesto.

2.2 Diagrama de Gantt

Una vez explicadas todas las tareas a realizar e hitos a alcanzar se representan en un Gantt para tener una visión más gráfica de toda la planificación del trabajo. Aunque se muestre a continuación, se puede ver el Gantt con mayor detalle en el apartado de anexos:

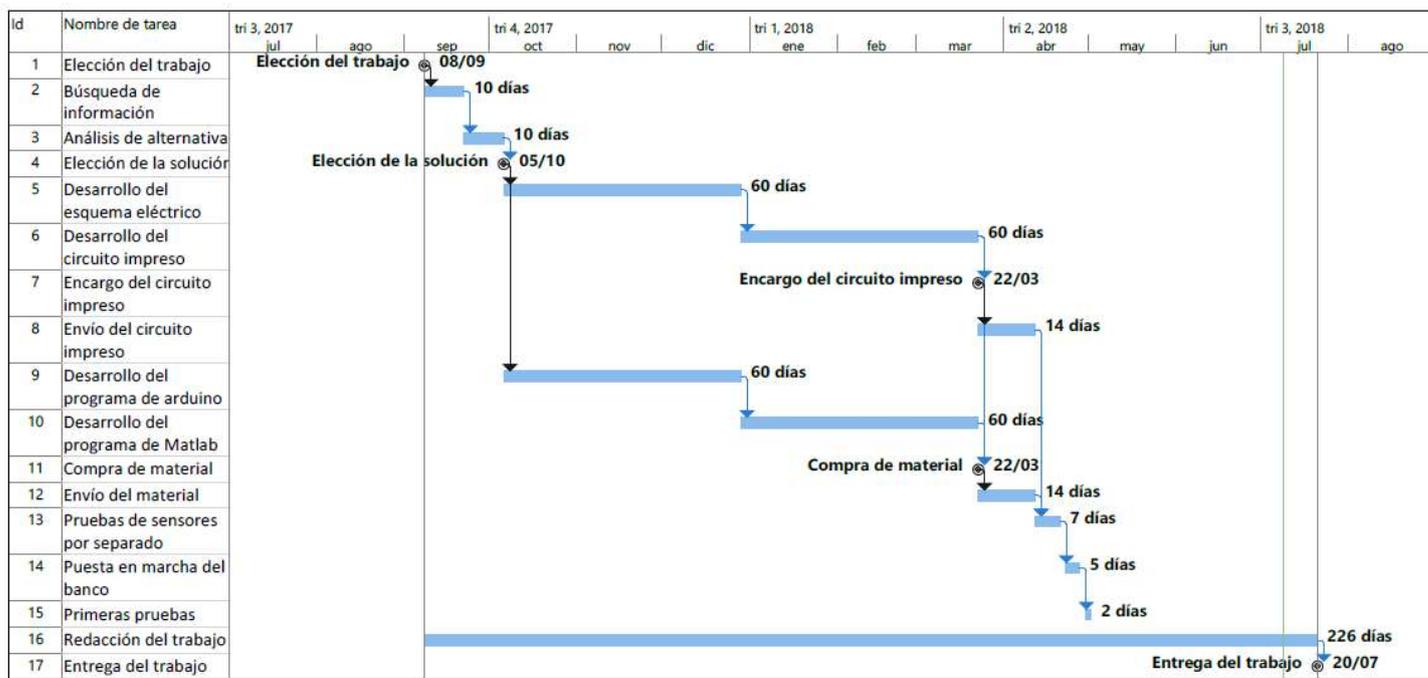


Ilustración 14. Diagrama de Gantt

2.3 Cálculos

En este apartado se muestran los cálculos realizados para la obtención de las frecuencias de muestreo necesarias para el sensor Hall y el Arduino. También se analiza el modo en el que se calculará el par motor y la potencia a partir de la señal de la celda de carga.

Frecuencia de muestreo de las revoluciones

Los cálculos necesarios para hallar las revoluciones máximas del rodillo se basan en el principio de la relación de transmisión entre engranajes. Para ello se necesita saber las revoluciones máximas del motor, la marcha con un desarrollo más largo, el tamaño de los engranajes de transmisión y los diámetros de la rueda y el rodillo.

La relación de transmisión es una relación entre las velocidades de rotación de dos engranajes conectados entre sí. Sabiendo dicha relación y la velocidad angular de uno

de los dos engranajes se puede obtener la velocidad angular del segundo mediante la siguiente fórmula:

$$\text{Relacion de transmisión} = \tau = \frac{\omega_2}{\omega_1} = \frac{Z_1}{Z_2} = \frac{\phi_1}{\phi_2}$$

Donde ω_1 , Z_1 y ϕ_1 son la velocidad angular, número de dientes y diámetro del engranaje 1 y ω_2 , Z_2 y ϕ_2 los del engranaje 2.

El rodillo girará a la máxima velocidad cuando lo haga el motor también. Esto ocurrirá a las revoluciones máximas del motor en la marcha más alta. La relación de transmisión de la última marcha respecto al eje de salida es de 25:21 según los datos proporcionados por la organización. El eje de salida está conectado al piñón de la moto. El piñón transmite la rotación a la corona y la corona a la rueda. Se puede ver en la siguiente imagen:

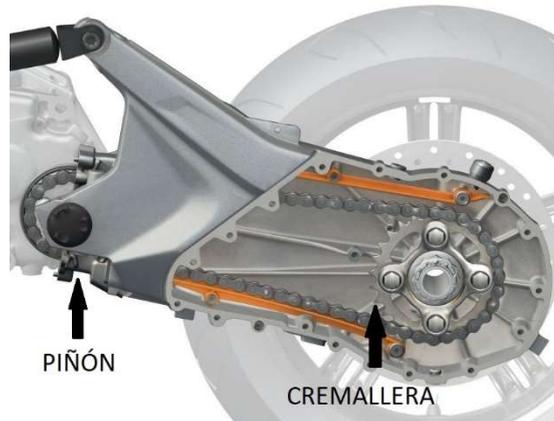


Ilustración 15. Piñón y cremallera de la moto (17)

Sabiendo que las revoluciones máximas del motor son 13500rpm y que internamente el motor tiene una relación interna de 0.375, el número de dientes del piñón son 15 y el de la cremallera 48, se tiene la siguiente velocidad de transmisión:

$$\tau_1 = 0.375 = \frac{\omega_2}{13500} \rightarrow \omega_2 = 5062.5 \text{ rpm}$$

$$\tau_2 = \frac{25}{21} = \frac{\omega_3}{5062.5} \rightarrow \omega_3 = 6026.78 \text{ rpm}$$

$$\tau_3 = \frac{15}{48} = \frac{\omega_3}{6026.78} \rightarrow \omega_3 = 1883.37 \text{ rpm}$$

Markel Lizarralde

Para los cálculos se ha tomado como τ_1 a la relación de transmisión interna, τ_2 la de la última marcha, τ_3 la relación del piñón con la cremallera, ω_2 la velocidad angular interna antes del cambio de marcha, ω_3 la velocidad angular del piñón y ω_4 la de la cremallera. La rueda trasera gira a la misma velocidad angular que la cremallera. Siendo la velocidad angular de la rueda 1883.37 rpm, su diámetro 601.3mm y el del rodillo 298 mm, las revoluciones máximas del rodillo son:

$$\tau_3 = \frac{601.3}{298} = \frac{\omega_4}{1883.37} \rightarrow \omega_4 = \mathbf{3800.24 \text{ rpm}}$$

Para obtener la frecuencia de muestreo del sensor de Hall se debe establecer un número de mediciones por revolución. Un margen de seguridad puede ser suponer que se harán 40 mediciones por revolución. Con estos datos la frecuencia de muestreo es:

$$3800.24 \frac{\text{revoluciones}}{\text{minuto}} * \frac{1 \text{ minuto}}{60 \text{ segundos}} * \frac{40 \text{ muestras}}{1 \text{ revolución}} = 2533.49 \frac{\text{muestras}}{\text{segundo}} \sim \mathbf{2.5 \text{ kHz}}$$

Cálculo del par

El cálculo del par motor se basa en los principios básicos de la mecánica. Las unidades internacionales del par son Nm, con lo que se obtiene a partir de multiplicar una fuerza por una distancia. En nuestro caso, el par motor se puede calcular a partir de una fuerza conocida, proveniente de la celda, y la distancia a ella desde el punto desde el que gira el freno. Esto es, multiplicando el resultado de la celda y la distancia normal desde el apoyo a ella. Se puede apreciar en la siguiente imagen:

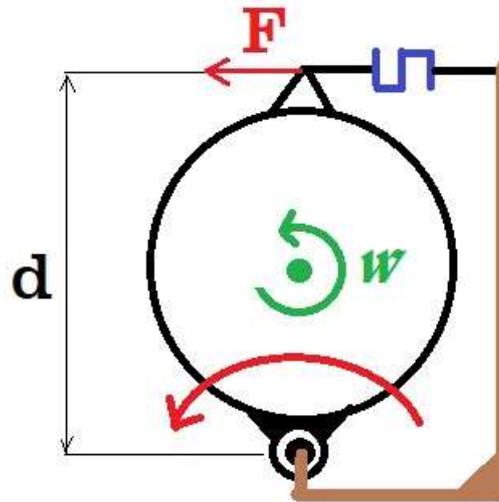


Ilustración 16. Esquema representativo del funcionamiento de la celda

Por lo tanto, el par motor desarrollado se calcula de la siguiente manera:

$$M = F * d \text{ (Nm)}$$

Cálculo de la potencia

El cálculo de la potencia se obtiene a partir del valor anterior del par (18). Se puede relacionar la potencia y el par motor como se muestra a continuación:

$$P = M * \omega \text{ (W)}$$

La ω es la velocidad angular del rodillo en rad/s, dato que se consigue con las mediciones del sensor de Hall. Para conseguir las revoluciones en esas unidades solamente habrá que hacer un cambio de unidades. Estos cálculos, los de las revoluciones, par motor y potencia los hará el arduino cada vez que recoja los datos.

2.4 Esquema eléctrico de la placa de control

En este apartado se detalla el esquema eléctrico completo. Para una mejor comprensión se ha separado en cuatro apartados distintos en función de lo que hacen. Estos apartados son las dos alimentaciones de la placa, el sensor de Hall, la celda de carga y su amplificación de señal y la pantalla LCD.

Control del freno de Foucault

A la hora de diseñar la parte del control del freno se debe recordar que la fuente de alimentación regulable del freno no está aislada galvánicamente. Esto significa que la tierra del freno está directamente conectada a la tierra del edificio. Como la tierra del Arduino es flotante, ambas tierras tienen diferente tensión. Por lo tanto, para controlar el freno se debe aislar la señal que se envía desde el Arduino y la que llega al freno. El circuito completo del control del freno es el siguiente:

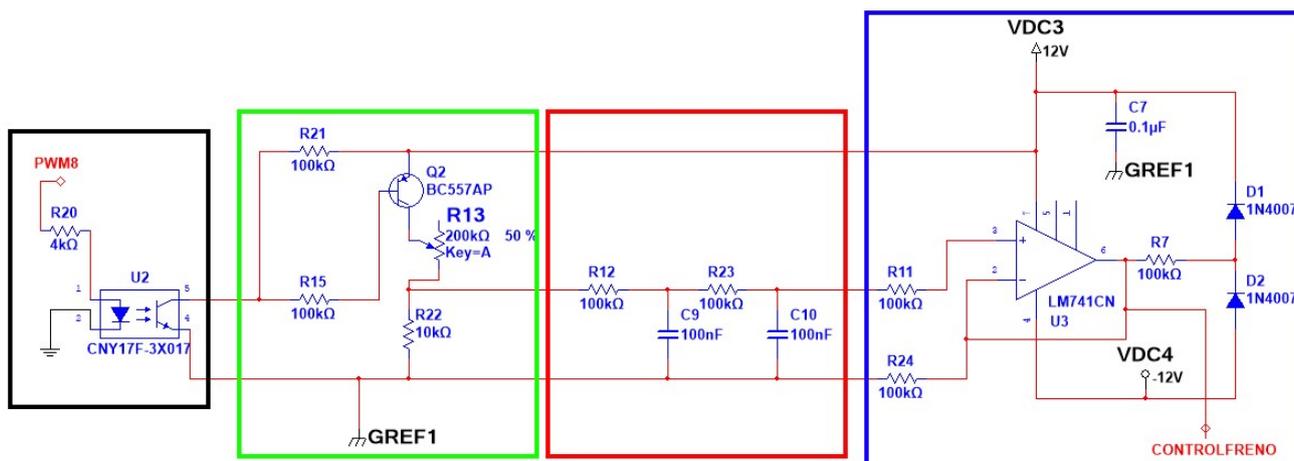


Ilustración 17. Esquema eléctrico del control del freno de Foucault

Una manera segura de aislar dos señales es utilizando un optoacoplador. Para ello, por un lado se conecta la señal de salida del Arduino. Cuando esta señal pase un determinado voltaje al otro lado del optoacoplador se genera una señal de distinta masa. De esta manera se pueden crear dos señales idénticas, pero con masas diferentes.

El optoacoplador no tiene un funcionamiento lineal, con lo que no da una señal de voltaje proporcional al voltaje aplicado en el otro lado. El Arduino Due puede tener como salida dos tipos de voltajes variables: con una salida analógica y con PWM. La salida analógica no sirve ya que el optoacoplador no es lineal. Por lo tanto, se opta por utilizar una de las salidas PWM. Esta salida tendrá un ciclo de trabajo en función de la carga que haya que aplicar al freno de Foucault.

La señal saliente del optoacoplador será una señal PWM opuesta a la de la entrada. Esto es, cuando la de la entrada esté a HIGH la de la salida a LOW y viceversa. Esto ocurre porque, cuando se activa la entrada del optoacoplador este conduce y hace que la salida se conecte a la masa. Cuando el PWM está en LOW el voltaje de la salida está en HIGH porque el

optoacoplador no conduce y estará al voltaje de la fuente de alimentación. Si se simula la salida del opto siendo la entrada una señal PWM de 3.3V de pico se obtiene lo siguiente:

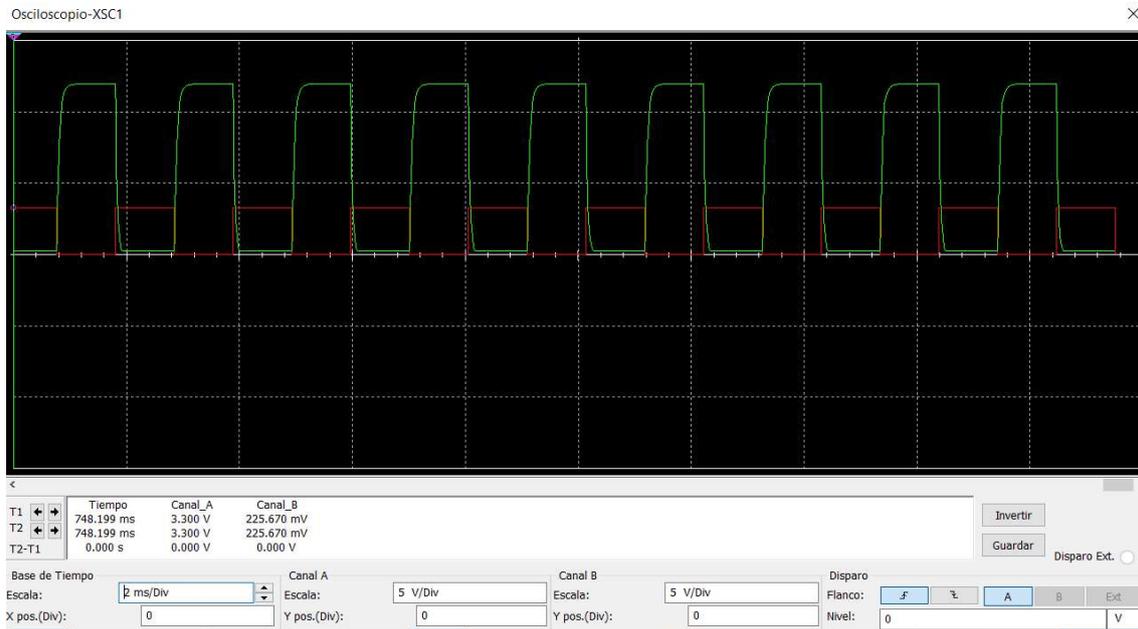


Ilustración 18. Señales simuladas en el osciloscopio

La señal roja es la de entrada y la verde de salida. Se puede ver como la señal de salida tiene 12V cuando se encuentra en HIGH. La segunda parte del circuito (el cuadrado verde) es una etapa en la que la señal se vuelve a invertir para que tenga la misma modulación que la de la salida del pin del arduino. El transistor hará que cambie el sentido de la corriente y tenga la siguiente forma:



Ilustración 19. Señales simuladas en el osciloscopio

La señal a entre el potenciómetro R13 y la resistencia R12 es la azul del osciloscopio. Como ya se ha explicado se trata de una señal invertida respecto a la verde. El voltaje de esta se debe a la caída de tensión que hay en el potenciómetro R13. Este potenciómetro se utiliza para ajustar la ganancia máxima de la señal que le llega al freno. Se ajustará de una manera a la similar a la mostrada en el osciloscopio ya que el voltaje máximo que le tiene que llegar al freno es de 10V.

El siguiente paso es filtrar la señal para que esta sea de un valor continuo y no PWM. Se han añadido un par de resistencias y condensadores para filtrar la señal (cuadrado rojo en el esquema). En la última parte del circuito, el cuadrado azul, se ha puesto un amplificador operacional con dos resistencias altas para que la salida a la fuente de alimentación del freno tenga una alta impedancia y no afecte al resto del circuito. La salida de la señal a la fuente es el punto del esquema en el que se denomina “CONTROLFRENO” y tiene el siguiente aspecto:

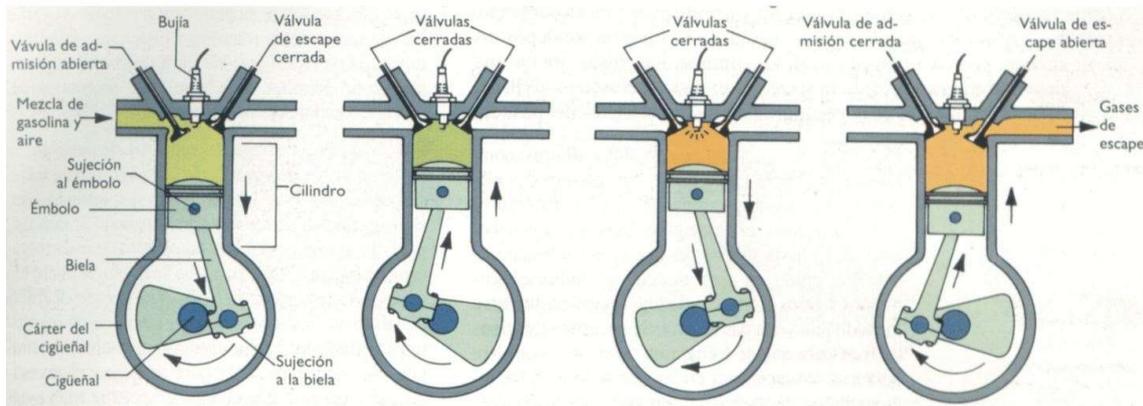


Ilustración 20. Señales simuladas en el osciloscopio

La señal de salida es la rosa. Se puede ver como se cumple que es la señal azul filtrada, ya que esta era de forma cuadrada y un ciclo de trabajo de 50% y la señal de salida tiene aproximadamente la mitad del valor pico de la azul.

Celda de carga y amplificación

El par motor desarrollado por el motor no tiene un valor constante. Un motor de cuatro tiempos (utilizado en la competición) necesita dos giros completos en el cigüeñal para poder realizar el ciclo de combustión completo. En la siguiente ilustración se puede ver cada fase del ciclo de combustión:



ADMISIÓN

COMPRESIÓN

EXPLOSIÓN

ESCAPE

Ilustración 21. Ciclo de combustión de un motor de 4 tiempos (19)

- En la admisión el pistón baja y entra el combustible por la válvula de admisión. El cigüeñal habrá dado media revolución.
- En la compresión el pistón sube dando otra media revolución y comprime la mezcla de aire y combustible. Durante este periodo las válvulas permanecen cerradas.
- La explosión o expansión hace que la mezcla se queme y produzca energía. De esta manera la explosión empuja nuevamente el pistón hacia abajo.
- Finalmente, se abre la válvula de escape y mientras el pistón vuelve a subir por segunda vez los gases de explosión son extraídos.

El único tramo en el que el motor realmente está desarrollando par motor es en la expansión. En los otros tres ciclos el cigüeñal sigue girando por la propia inercia del motor. En consecuencia, el empuje desarrollado a la celda por la rotación del freno no será constante. Se puede apreciar en la siguiente imagen:

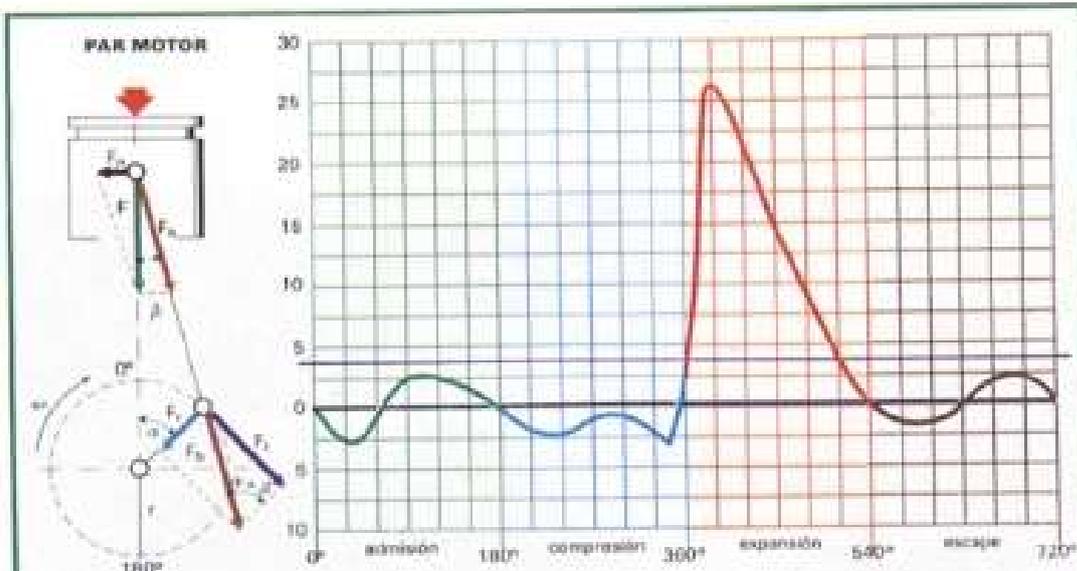


Ilustración 22. Par motor en función de la fase del ciclo del motor (20)

Por lo tanto, la señal a obtener de la celda de carga tendrá la forma mostrada. Es un factor a tener en cuenta, ya que esta forma se repetirá cada ciclo de combustión y el circuito electrónico debe ser diseñado para leerlo sin perder el más mínimo detalle. Para ello habrá que filtrar la señal obtenida.

La estructura interna de la celda de carga es un puente de Wheatstone. Se trata de unas resistencias colocadas de la siguiente manera:

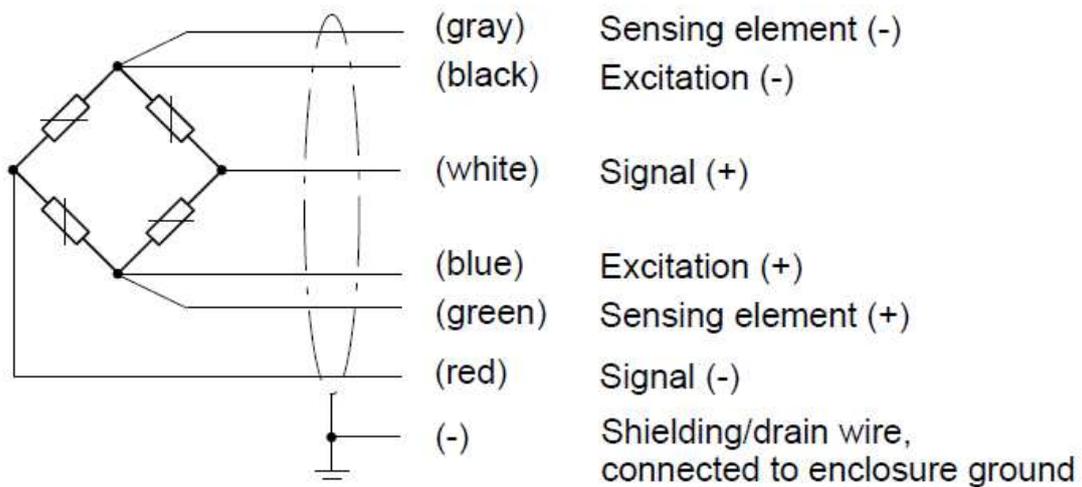


Ilustración 23. Puente de Wheatstone

Estas resistencias hacen que cuando la carga aplicada en la celda varíe, también lo haga el voltaje de la señal. El voltaje máximo de la señal serán 10mV, con lo que la señal a amplificar tendrá que tener una ganancia de:

$$Ganancia = \frac{3.3V}{0.01V} = 330$$

De esta manera el voltaje de salida amplificado tendrá un valor entre 0 y 3.3V (rango de medición del arduino). Esta ganancia se logrará a lo largo de dos etapas diferentes. El circuito completo de la celda de carga es el siguiente:

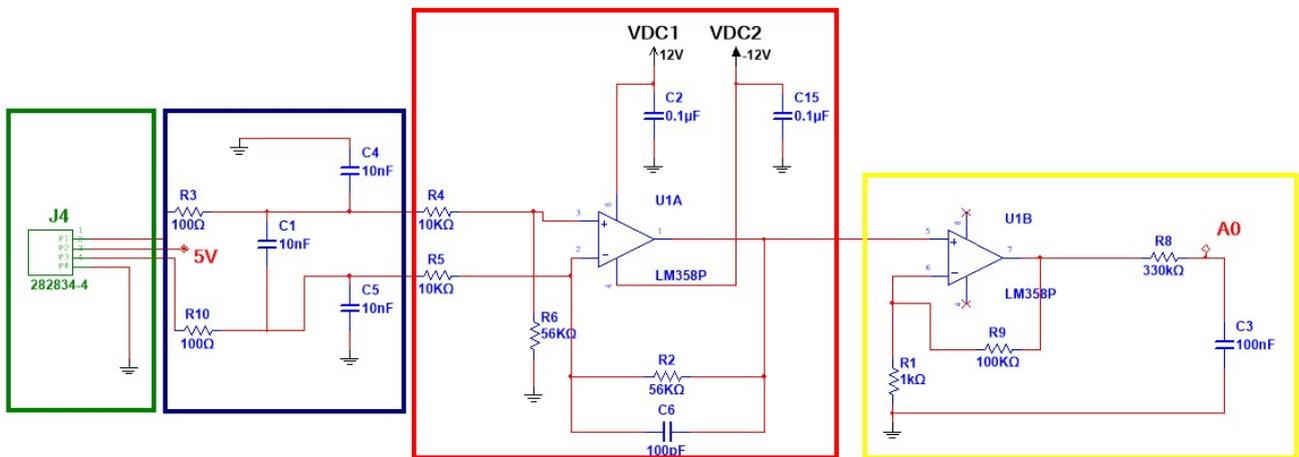


Ilustración 24. Circuito de obtención de la señal de la celda de carga

Como se puede ver el circuito está dividido en 4 partes diferentes. El circuito se explicará de izquierda a derecha. La zona verde no es más que el terminal de conexiones de la celda al circuito. Dos cables son los de la señal y los otros dos de alimentación; 5V y masa.

El cuadrado azul es un simple filtro para prevenir ruido eléctrico exterior que pueda afectar a la medición. Solamente queremos medir la señal procedente de la celda de carga.

El cuadrado rojo es la primera etapa de amplificación. Se trata de un amplificador operacional en modo diferencial con un condensador C6 al final de la salida para filtrar la señal. Funciona de la siguiente manera:

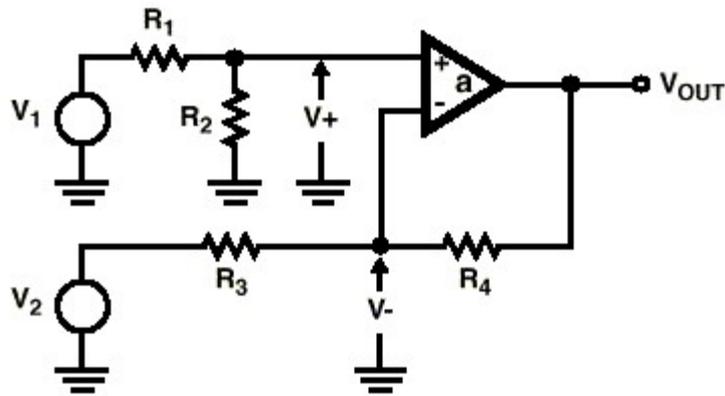


Ilustración 25. Amplificador operacional en modo diferencial (21)

El voltaje de salida V_{out} en función de las resistencias y los voltajes de entrada V_1 y V_2 es el siguiente:

$$V_{out} = (V_1 - V_2) * \frac{R_2}{R_1}$$

La ganancia del amplificador es la división de ambas resistencias. En el circuito diseñado se han escogido resistencias de $56 \text{ k}\Omega$ y $10 \text{ k}\Omega$. La ganancia es:

$$Ganancia_1 = \frac{R_2}{R_1} = \frac{56 \text{ k}\Omega}{10 \text{ k}\Omega} = 5.6$$

La última etapa de este circuito es la segunda etapa de amplificación. En este caso se ha utilizado otro amplificador operacional en modo no inversor. Su funcionamiento es similar al del diferencial y es el siguiente:

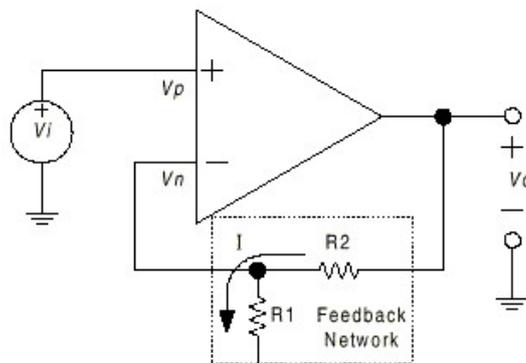


Ilustración 26. Amplificador operacional no inversor

Su ganancia en función de las resistencias y voltaje aplicado es:

$$V_{out} = V_i * \frac{R_1 + R_2}{R_1} \rightarrow \text{Ganancia}_2 = \frac{1k\Omega + 100k\Omega}{1k\Omega} = 101$$

El circuito tiene al final un filtro de paso bajo para filtrar la señal de salida. La ganancia total del circuito será el producto de las dos ganancias:

$$\text{Ganancia total} = \text{Ganancia}_1 * \text{Ganancia}_2 = 5.6 * 101 = 565.6$$

Se trata de una ganancia superior a la teórica calculada anteriormente. El motivo es que las motos que se utilizan en la competición son de una potencia que no hacen que la carga aplicada sobrepase los 50kg de fuerza aproximadamente. Por lo tanto, se ha decidido aumentar la ganancia para que el rango de medición sea el más adecuado a su uso. La carga máxima medible sin sobrepasar el voltaje de arduino con la ganancia diseñada será el siguiente:

$$\text{Carga máxima medible} = 100kg * \frac{330}{565.6} \approx 55 kg$$

Fuentes de alimentación

En el esquema eléctrico se integran de la siguiente manera:

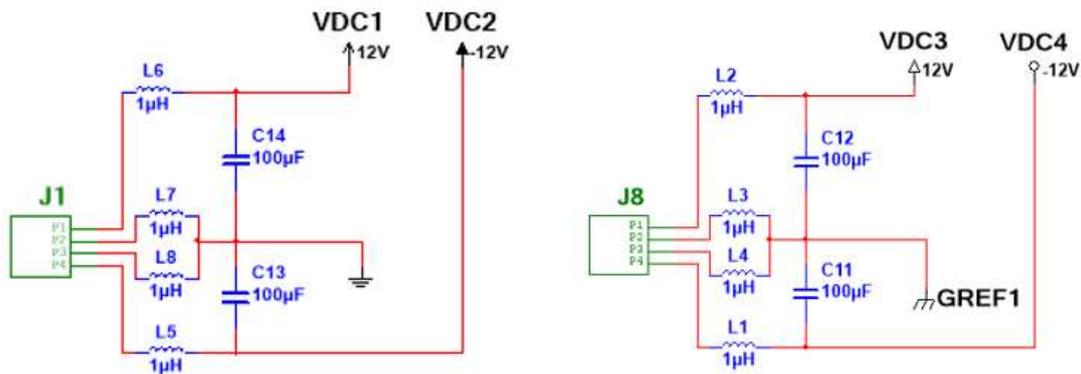


Ilustración 27. Esquema eléctrico de las fuentes de alimentación

Las dos fuentes de alimentación se conectan de la misma manera. La fuente que alimenta a todo el circuito salvo a la parte del control del freno es la que se muestra a la izquierda. VDC1 y VDC2 son los voltajes a los que se conecta la parte de amplificación de la celda,

Markel Lizarralde

mientras que el Arduino se conecta entre VDC1 y GND. Un voltaje mayor aplicado al Arduino podría dañarlo permanentemente. A la derecha se puede ver que la segunda fuente se conecta de la misma manera. VDC3 y VDC4 son los voltajes aplicados al control del freno. La tierra de esta parte se ha nombrado como GREF1.

En las dos partes se ha añadido inductancias de $1\mu\text{H}$ y condensadores de $100\mu\text{F}$. Estas se utilizan para evitar que cambios de tensión y corriente producidos por factores externos a la placa afecten a toda la electrónica que contiene.

Sensor Hall

El sensor Hall es una de las partes más simples del circuito. Tiene 3 pines, siendo 2 los de alimentación y 1 de señal. La señal es una onda cuadrada de pulsos de valores LOW y HIGH al voltaje aplicado. Como los pines digitales del Arduino Due operan a 3,3V alimentamos el sensor con este voltaje. Se ha conectado al pin digital 22, pero podría ir conectado a otro cualquiera. Su conexión queda de la siguiente manera:

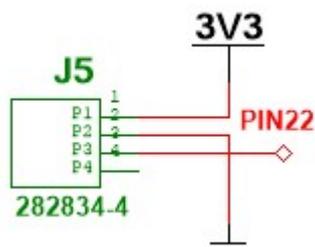


Ilustración 28. Esquema eléctrico del sensor Hall

Pantalla LCD

La pantalla LCD tiene 16 pines de los cuales 12 hay que conectarlos. Los 4 restantes se dejan sin conectar. A continuación se puede ver las conexiones necesarias para utilizar una pantalla LCD en Arduino:

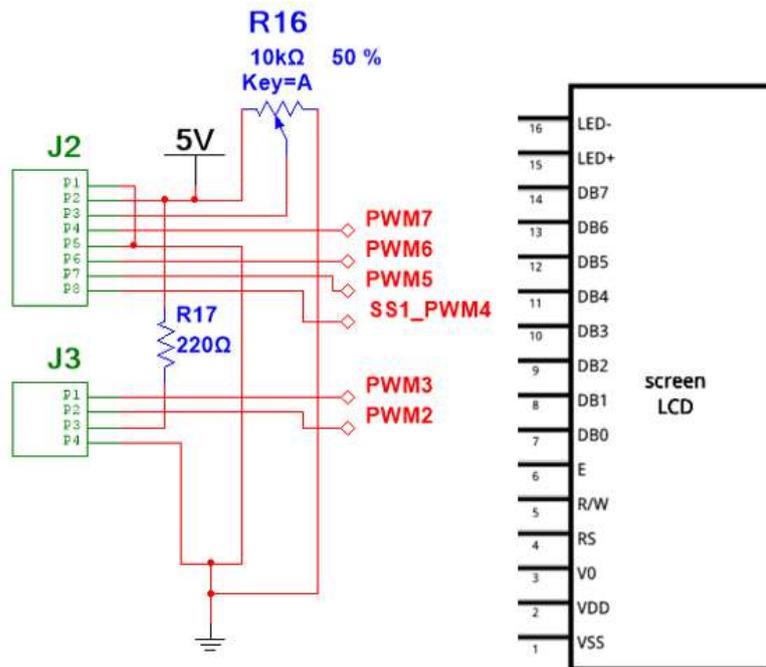


Ilustración 29. Esquema eléctrico de la conexión del LCD (22)

6 pines digitales son necesarios para el uso de la pantalla. Los pines escogidos son los mostrados en el esquema (desde el PWM2 hasta el PWM7). No es necesario que estos pines tengan la función de PWM. Se han escogido esos por comodidad a la hora de colocar los componentes en la placa. Se utiliza un potenciómetro de 10kΩ para poder regular el brillo de la pantalla cuando sea necesario.

Como la numeración de los pines del LCD y el de los terminales de conexión de J2 y J3 son distintos, se muestra en una tabla qué pin se conecta con cual para una mejor comprensión del esquema:

PIN DEL LCD	TERMINAL DE CONEXIÓN	PIN DEL TERMINAL EN EL ESQUEMA
1	J2	P1
2	J2	P2
3	J2	P3
4	J2	P4

5	J2	P5
6	J2	P6
7	No conectado	No conectado
8	No conectado	No conectado
9	No conectado	No conectado
10	No conectado	No conectado
11	J2	P7
12	J2	P8
13	J3	P1
14	J3	P2
15	J3	P3
16	J3	P4

Ilustración 30. Relación de conexiones del LCD

Consigna de revoluciones deseadas

Se conecta un potenciómetro que convierte la posición de su cursor a un voltaje proporcional al alimentado. Este voltaje a su vez es proporcional a la consigna de revoluciones que se quiere controlar. El potenciómetro tiene 3 pines, dos de alimentación y el de señal.

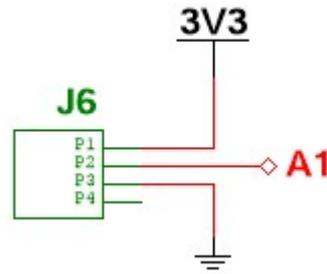


Ilustración 31. Esquema eléctrico de la conexión del potenciómetro para la consigna

Al igual que el sensor de Hall el potenciómetro debe operar a 3.3V para no dañar el Arduino, con lo que es alimentado desde la salida de 3.3V que tiene el Arduino. El pin intermedio de la señal se conecta al pin analógico 1.

2.5 Diseño del circuito impreso

Para el diseño del circuito impreso se ha utilizado el programa Ultiboard 14.0. Es una extensión del Multisim que permite realizar diseños completos de placas electrónicas. La función del Multisim es hacer el esquema y todas las conexiones de manera virtual, mientras que Ultiboard es la plataforma en la que todos los componentes se sitúan sobre la placa electrónica y se hace el ruteado completo. Para diseñar un circuito hay que importar todos los componentes del esquema. Estos aparecen en Ultiboard en una posición al azar con todas las conexiones de manera virtual. Cuando se importa el esquema completo del banco de potencia aparece lo siguiente:

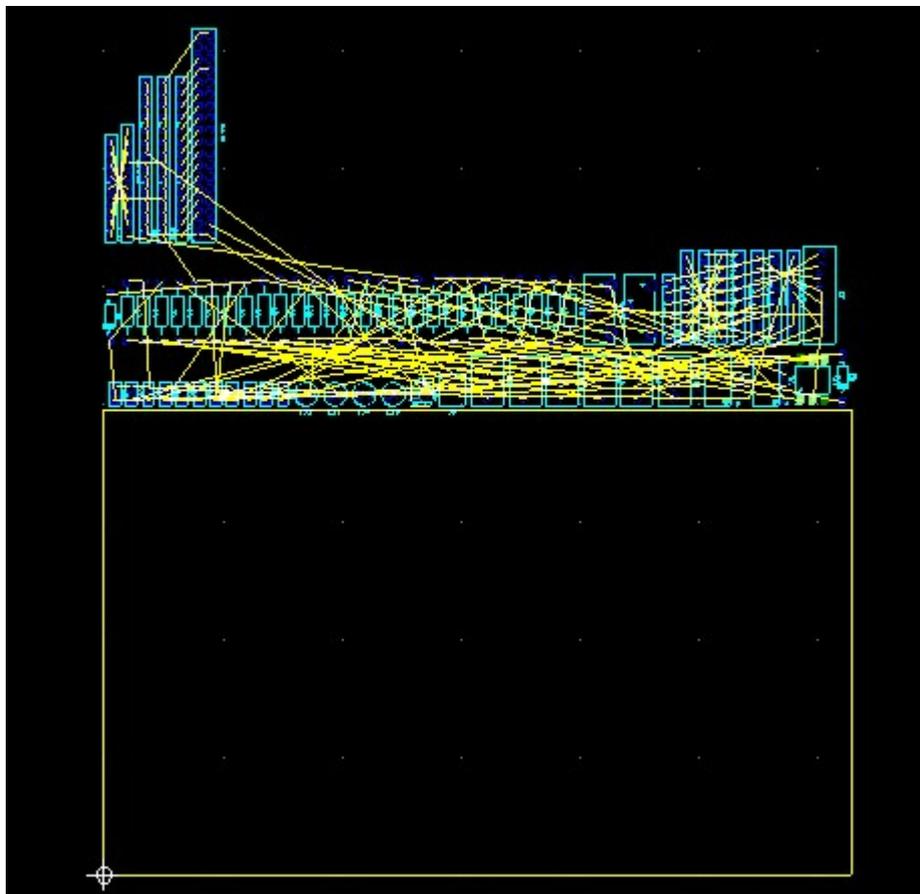


Ilustración 32. Diseño del PCB sin colocar los componentes

El rectángulo amarillo simboliza la placa electrónica, los rectángulos azules son todos los componentes electrónicos del esquema y las líneas amarillas simbolizan todas las conexiones que hay entre dichos componentes.

El siguiente paso es colocar todos los componentes en su sitio. Algunos pueden ir colocados donde más le convenga al conjunto, pero otros deben situarse en una posición obligatoria para que puedan conectarse ciertos componentes. Estos son los pines hembra a los que se conecta el Arduino. Si estos pines se conectan en el lugar equivocado el Arduino no podrá ser conectado y la placa completa no servirá para nada.

Una vez colocados estos pines hembra se procede a colocar el resto de componentes. Se intenta buscar un equilibrio entre que los componentes conectados entre sí queden lo más cerca posible y que los componentes que pertenecen a diferentes tierras queden

separados para evitar riesgos. De la misma manera, se intenta mantener las entradas de las dos fuentes de alimentación separadas. Una vez colocados todos los componentes el circuito queda de la siguiente manera:

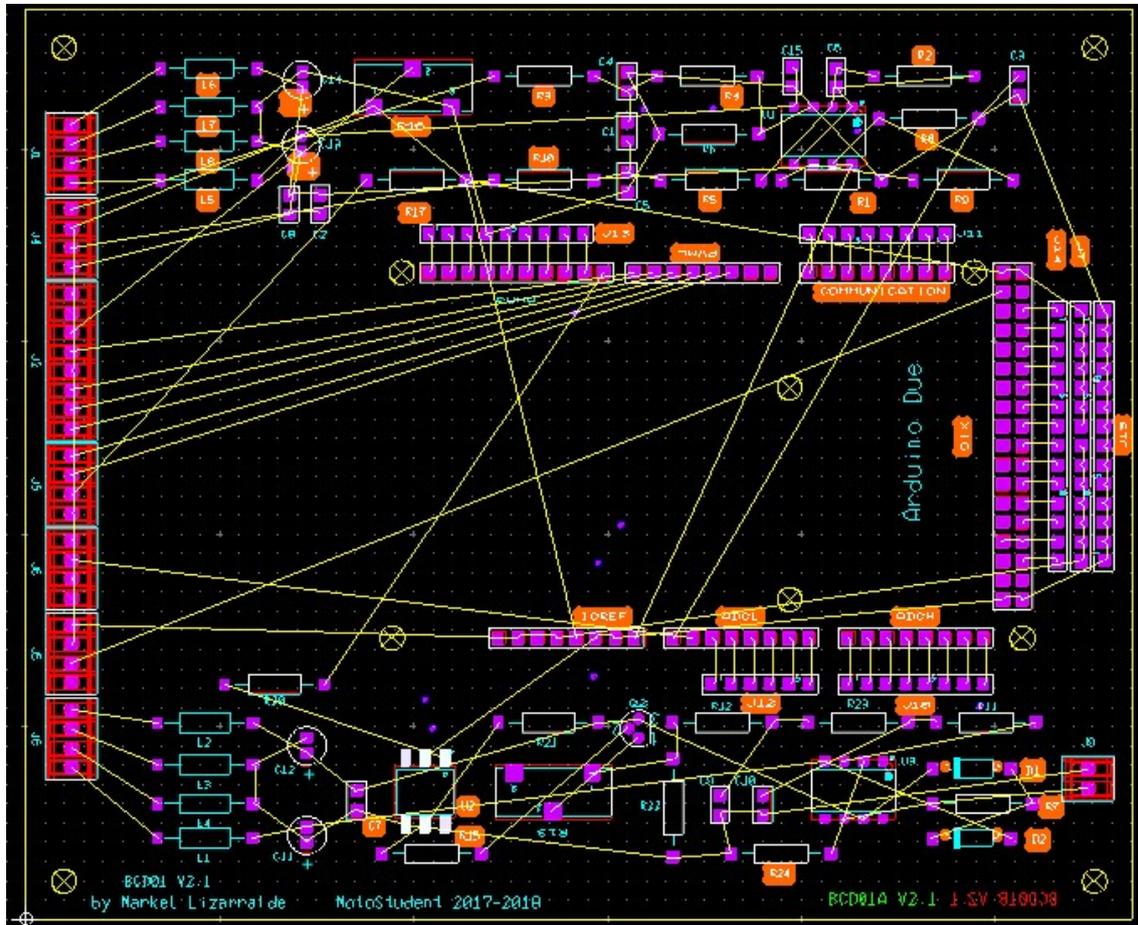


Ilustración 33. Diseño del PCB con los componentes colocados

Se puede distinguir 4 partes diferentes en el circuito: La primera de todas es la parte central, zona reservada para conectar el Arduino. Los pines extra de ampliación están colocados lo más cerca posible para que el circuito quede lo más sencillo y claro posible. En la parte inferior se encuentra toda la parte de control del freno. La parte superior tiene todo lo correspondiente a la amplificación y estabilización de la celda de carga. Finalmente se ha colocado a la izquierda todos los conectores de los componentes externos como la pantalla LCD, la celda de carga, el sensor de Hall y las fuentes de alimentación.

Como se ha dicho anteriormente, se ha respetado el criterio de mantener los componentes conectados lo más cerca posible. También se ha respetado las diferentes tierras colocando toda la parte del freno abajo y el resto arriba. Como se puede ver en la imagen, se han añadido algunas serigrafías en la parte inferior y algunos agujeros en los bordes y en la parte central de la placa para poder fijar todo el circuito en un panel en el banco de potencia.

Las líneas amarillas son las conexiones todavía sin conectar. Este es el último paso a realizar. Una vez asegurados y bloqueados todos los componentes en su posición se conectan todos los componentes entre ellos. Existen dos maneras de hacerlo: De manera automática o manual. Se ha optado por hacer una fusión de los dos. Primero se ha hecho las conexiones con la función automática del programa y todas las que se ha visto que no quedaban de la mejor manera se han cambiado manualmente.

Una vez hecho todo el ruteo de las pistas ha quedado como en la imagen que se muestra a continuación:

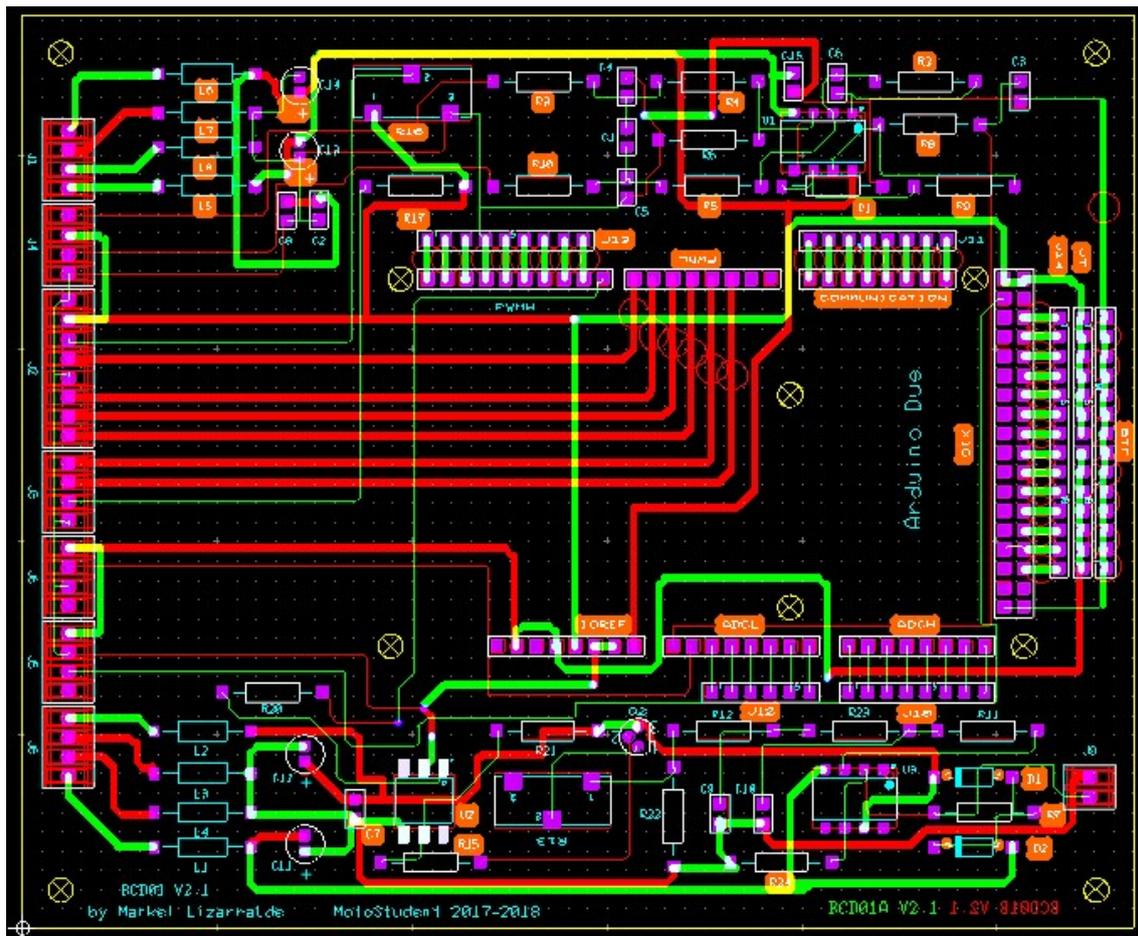


Ilustración 34. Diseño del PCB con las pistas colocadas

Hay pistas que tienen una anchura mayor que otras. Son pistas que necesitan mayor

espesor ya que circulará una corriente mayor que en otras pistas o que simplemente no tiene componentes cerca y puede cubrir más espacio. Se aprecian dos colores de pistas: Rojos y verdes. La placa es de doble capa, lo que quiere decir que los componentes y pistas se pueden colocar en dos capas distintas, superior e inferior. Todos los componentes están colocados en la capa superior junto con las pistas verdes. La capa inferior solamente contiene las pistas verdes. Los circuitos típicamente tienen dos capas ya que permite hacer conexiones mucho más complejas comparándolas con las de una capa.

Ultiboard ofrece la posibilidad de representar en 3D el diseño realizado. Es una herramienta útil ya que permite visualizar la separación real de los componentes. El diseño final es el siguiente:

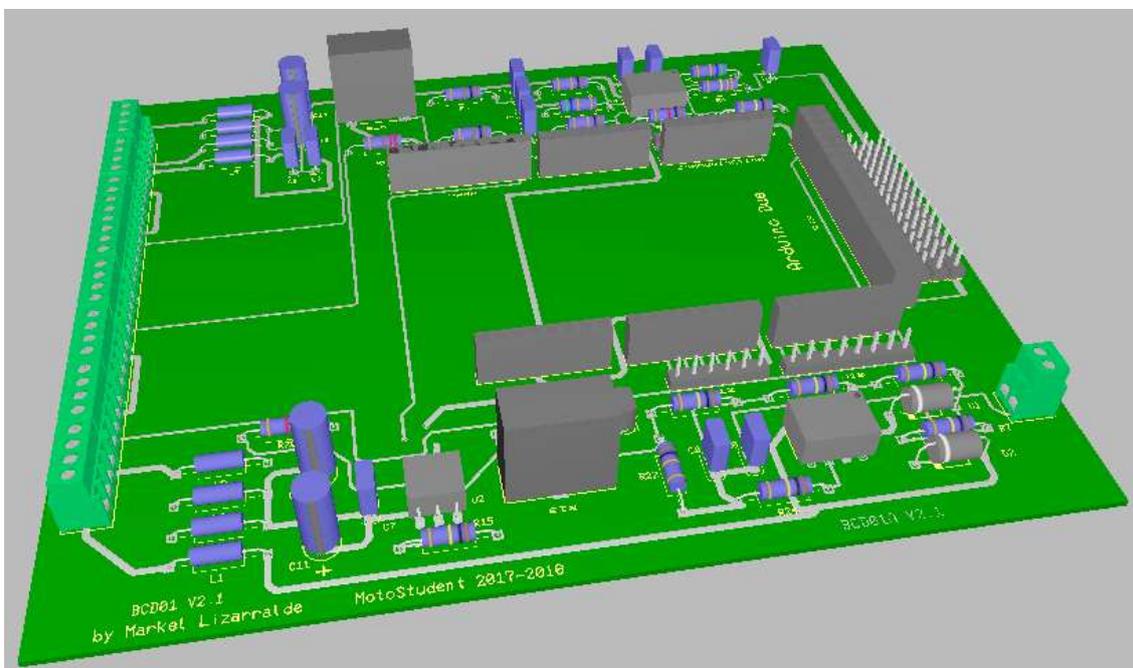


Ilustración 35. Representación tridimensional del PCB

2.6 Diseño del Software de control

A continuación se explica la programación requerida para el funcionamiento de la placa electrónica y el programa del ordenador. Como ya se ha explicado en apartados anteriores, el Arduino recibe los datos de los sensores, calcula la potencia, el par motor y las revoluciones y envía los datos tanto a la pantalla LCD y el ordenador. Primeramente se explica el programa hecho para el arduino y después la parte de mostrar los datos en Matlab. El código completo de los dos programas se puede encontrar en sus correspondientes apartados de anexos.

Programación del Arduino

Arduino utiliza un lenguaje de programación que combina los fundamentos de C y librerías propias de Arduino. Para una correcta comprensión, antes de explicar cada una de las partes del programa, se muestra el diagrama de flujos del programa completo:

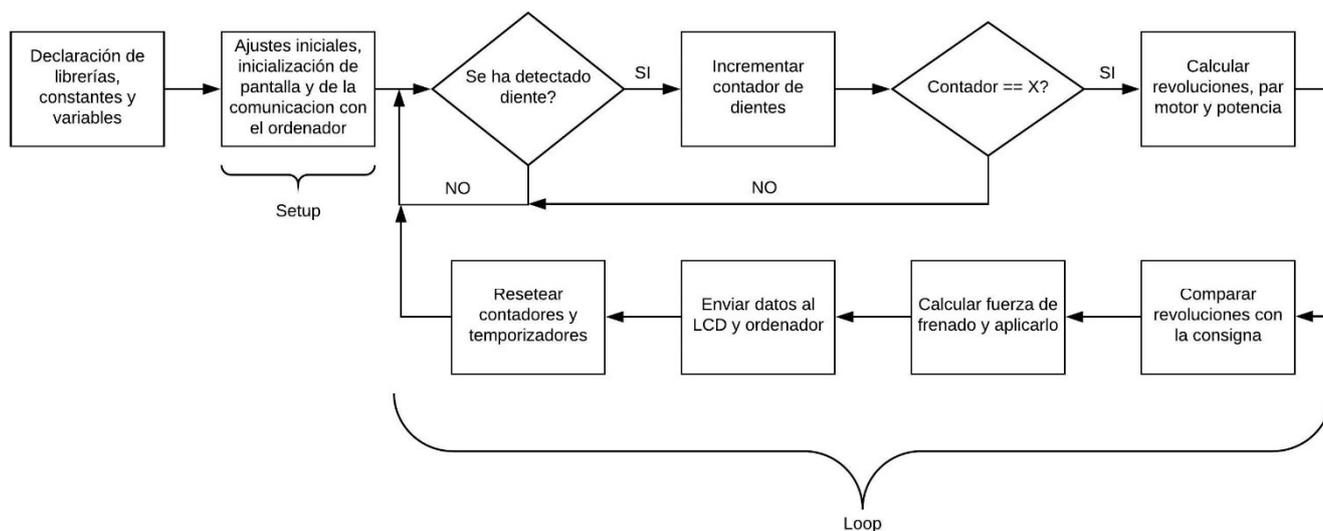


Ilustración 36. Diagrama de flujos del programa del arduino

La estructura con la que se programa en arduino puede dividirse en 4 partes fundamentales: La declaración de librerías, constantes y variables, la función setup, función loop y las funciones creadas personalmente. Se explican a continuación:

En la primera parte se declaran las librerías, constantes y variables necesarias para controlar todo el montaje eléctrico. Como se puede ver es la primera etapa del diagrama. La única librería necesaria es la de "LiquidCrystal" y sirve para poder controlar la pantalla LCD con unas funciones muy sencillas. Las constantes son los nombres que se le da a los pines utilizados del arduino y las variables son datos que se almacenan como el muestreo del tiempo, la potencia, el par, etc. Son datos que cambian de valor en cada ciclo del programa.

La segunda parte del programa es la función "setup". Es la primera función que se ejecuta tras iniciar el arduino cada vez que se vaya a usar el banco. En él se inicializa la pantalla para que muestre el texto que debe mostrar, se declaran los pines del arduino que se van a

utilizar y se inicia la comunicación serie para enviarle los datos por USB al ordenador. Es la segunda etapa del diagrama.

Terminada la función setup el programa entra a la función “loop”. Es una función cíclica que se ejecuta constantemente. Esta función engloba a todo el bucle representado en el diagrama. En él se realiza las mediciones de los sensores, se calcula la potencia, par motor... y se envía tanto al LCD como al ordenador. El modo de funcionamiento es el siguiente:

El arduino comprueba constantemente si se detecta algún diente de la rueda fónica y, en caso de detectarlo, lo compara con una constante que nosotros hemos declarado al principio del programa. Si su valor no coincide vuelve a la situación inicial, en la que comprueba si el sensor detecta un diente. Cuando se haya detectado el número de dientes estipulado en la constante se pasa a realizar el cálculo de todas las magnitudes importantes. Calculada las revoluciones del rodillo se compara con la consigna marcada con el potenciómetro y se hacen los cálculos del par de frenado a aplicar desde el freno. Finalmente envía todos los datos a la pantalla LCD y al ordenador y resetea algunos contadores como los muestreos de tiempo para las revoluciones y el contador de dientes. Tras esto vuelve a la situación inicial en la que comprueba si detecta dientes o no.

El método utilizado para detectar los dientes es con interrupciones Hardware. El pin al que se conecta el Hall detecta los flancos positivos de la señal (cuando llega a un diente) y el arduino salta a la función que aumenta el contador de dientes. Una vez aumentado el contador vuelve a la línea de código en la que estaba antes de la interrupción.

La última parte del programa del arduino tiene las funciones creadas personalmente para que el resto del programa quede más organizado. Estas funciones que se han creado son la de la interrupción del Hall que aumenta el contador, la que inicializa el texto que estará en la pantalla LCD y la que actualiza los valores en la pantalla. El LCD mostrará los valores de la siguiente manera:

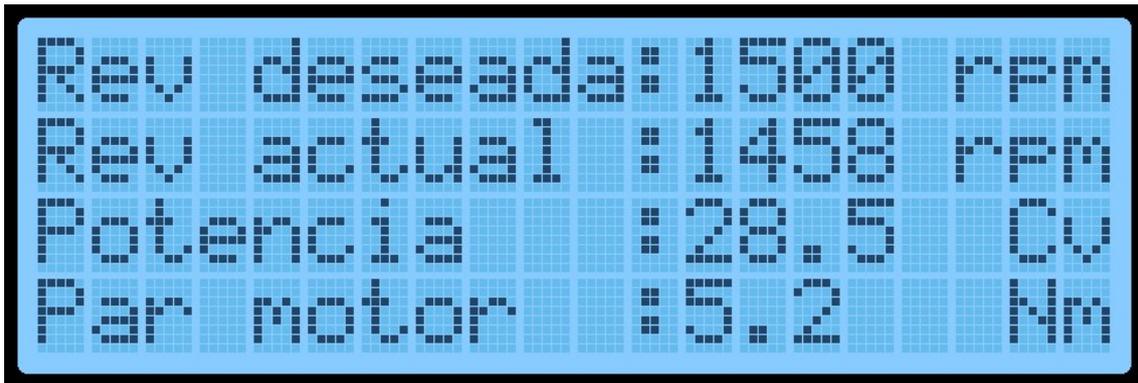


Ilustración 37. Pantalla LCD

Programación del ordenador

El diseño del programa de la parte del ordenador se divide en dos partes: La parte gráfica y la parte de programación. El lenguaje utilizado es similar al de Arduino ya que solamente cambia un poco la sintaxis a la hora de escribir, pero los conceptos son los mismos. Primero se muestra el entorno gráfico del programa y después se detalla explica el funcionamiento del mismo. El programa tiene la siguiente apariencia:

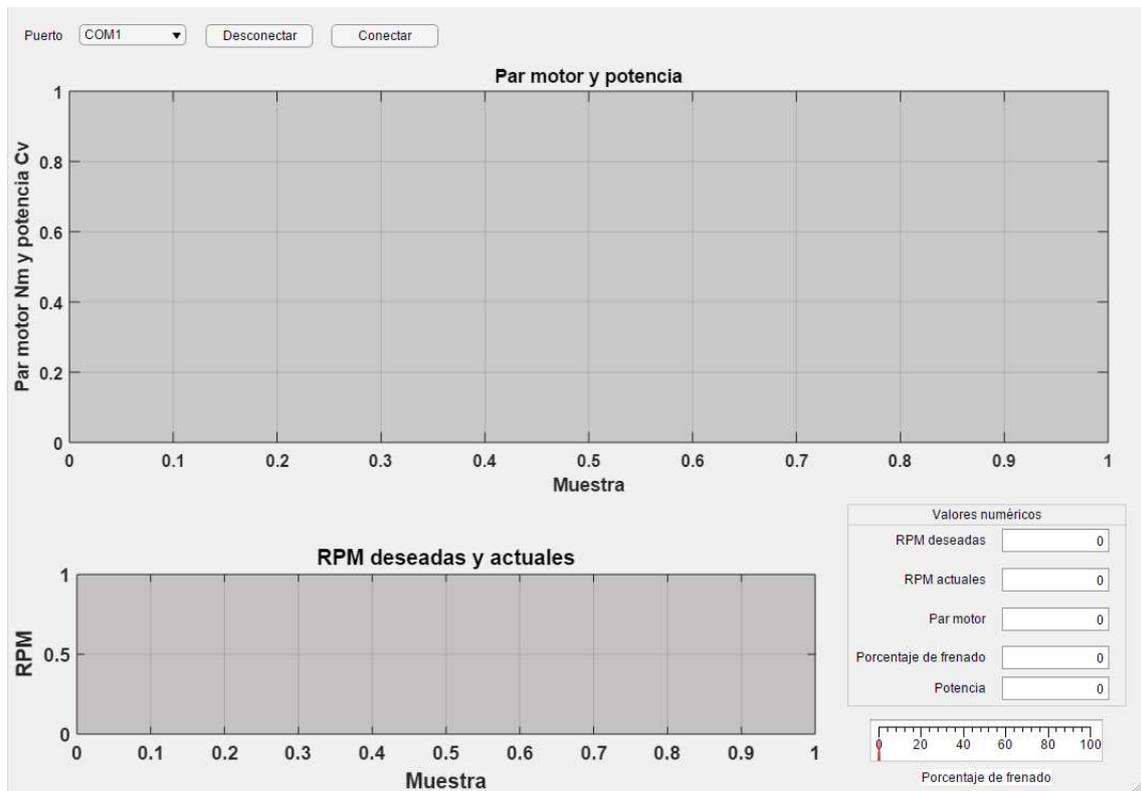


Ilustración 38. Entorno gráfico del programa del ordenador

En la parte superior está la lista de puertos USB disponibles en el ordenador. Para comenzar la comunicación con el arduino se debe escoger el puerto al que se conecta y darle a conectar. Al presionar el botón se establece la comunicación y los datos empiezan a llegar. La potencia y par motor son graficados en la gráfica más grande ya que son las variables que se vigilarán con más detalle para ver si el motor está mejorando o no. Las revoluciones del rodillo son graficadas junto con la consigna para poder ver de una manera sencilla la regulación del freno del banco. El freno debe ser capaz de regular la carga para que ambas revoluciones coincidan. A la derecha se ven los mismos datos que se grafican, pero mostrando solamente el valor numérico. También se incluye una barra que indica el porcentaje de frenado del freno, esto es, el porcentaje de toda la carga que puede aplicar el freno.

El modo de funcionamiento es que las gráficas se irán actualizando a tiempo real. Para aumentar la potencia del motor se irá variando los parámetros de la centralita de manera que se note un aumento en la gráfica de la potencia. Cuando no haya una mejora en las condiciones en las que se esté probando se cambia la consigna de revoluciones y se sitúa la prueba en otro punto. De esta manera se calculan los parámetros que ofrezcan la mayor potencia posible en todos los puntos de funcionamiento.

El programa que hace que funcione la parte gráfica del ordenador sigue el siguiente diagrama de flujo:

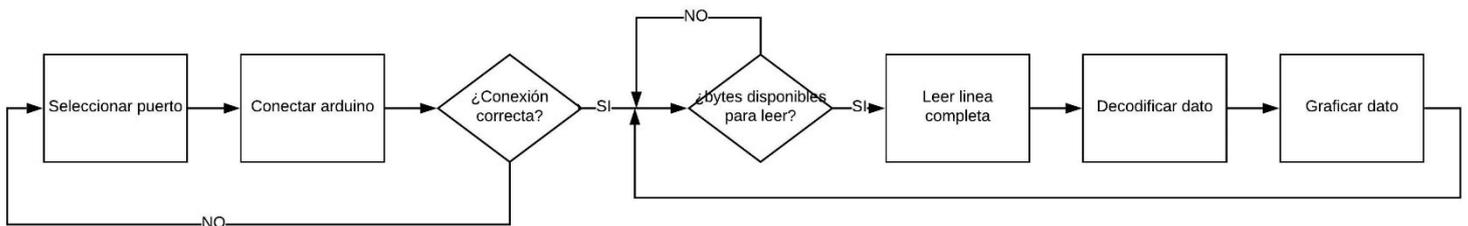


Ilustración 39. Diagrama de flujo del programa del ordenador

Inicialmente el programa permanece en reposo esperando a que se inicie la comunicación con el Arduino. Una vez establecida la conexión de manera correcta comienza la lectura de datos. El programa comprueba constantemente si hay bytes en el buffer serie para leer y, en caso de que sea correcto, lee bytes hasta que encuentre un salto de línea. Cuando llega un salto de línea significa que se ha enviado un dato completo.

Los datos vienen codificados con un identificador previamente. Todos los identificadores y números se envían en variables de tipo char. Un mensaje procedente del Arduino tiene el siguiente aspecto:

```
"R2789\nP0034\nF0099\nC1529\nM0123\n"
```

Hay 5 tipos de identificadores: R para las revoluciones, P para la potencia, F para el par de frenado, C como consigna y M para el par motor. "\n" simboliza un salto de línea y como cada vez que se lee un mensaje lee hasta que se detecta el salto, en cada lectura se consigue un dato completo con su identificador. En función de cual sea el identificador que llegue al inicio del mensaje se identifica el dato que ha llegado. Se obtiene el número del mensaje y se representa en la gráfica que le corresponda. Una vez graficado el dato vuelve a comprobar si hay bytes para leer y así sucesivamente, hasta que se cierre el programa o se pulse el botón de desconectar para parar la comunicación. El código completo que ejecuta todo lo explicado en este apartado se muestra en el apartado de anexos.

2.7 Puesta en marcha. Primeras pruebas

Una vez terminado el diseño tanto de la placa como de los programas que lo controlan hay que montar todo el sistema. La placa electrónica viene sin los componentes eléctricos, con lo que hay que soldarlos manualmente. El resultado es el siguiente:

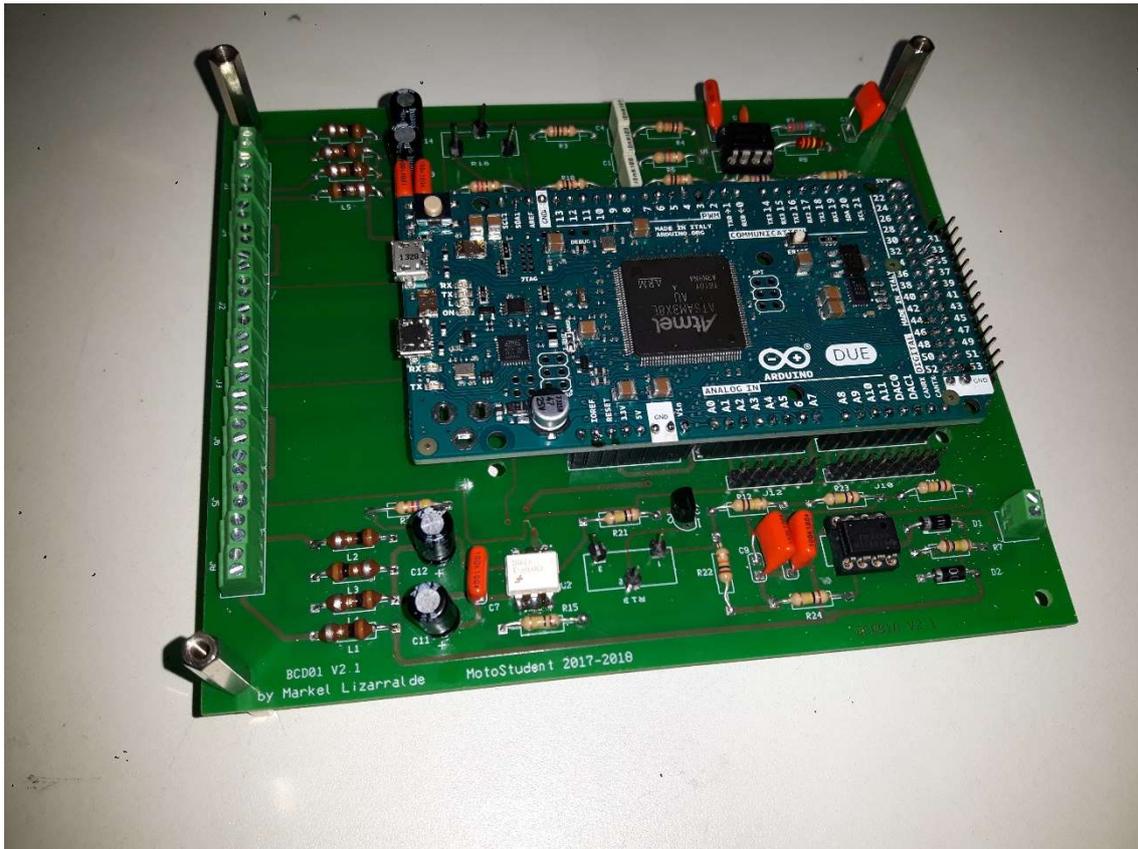


Ilustración 40. Placa de control soldada

Para comprobar que todos los componentes funcionan como deberían se han probado por separado antes de montar en el banco. Se ha hecho pruebas con el sensor Hall y la celda de carga para comprobar que las revoluciones y la carga se miden de la manera adecuada.

Para probar el sensor de Hall se ha utilizado un motor de corriente alterna disponible en el laboratorio de motores térmicos. Se ha acoplado al eje de este motor unos tornillos para que hagan la función de dientes. El montaje provisional ha quedado de la siguiente manera:



Ilustración 41. Montaje provisional del sensor Hall

Para asegurar que el sensor y el programa miden las revoluciones de la manera correcta se ha utilizado un tacómetro óptico y se han comparado los resultados. Se han hecho mediciones a distintas revoluciones y se han conseguido los siguientes resultados:

Sensor Hall	Tacómetro óptico	Error relativo	Error relativo medio
125	126	0,80%	0,45%
354	352	0,56%	
568	565	0,53%	
834	830	0,48%	
1023	1028	0,49%	
1256	1263	0,56%	
1412	1420	0,57%	
1754	1750	0,23%	
1963	1967	0,20%	
2103	2110	0,33%	
2336	2350	0,60%	
2587	2595	0,31%	
2854	2862	0,28%	
2953	2965	0,41%	

Ilustración 42. Resultados experimentales del sensor Hall

Como se puede observar los resultados son los deseados. Se ha obtenido un error relativo menor al 1% en todas las mediciones y el error medio es del 0.45%. No se ha podido medir a velocidades superiores ya que el motor no tenía capacidad de ir más rápido. De todas

maneras, se ha medido cerca del límite teórico de las revoluciones del rodillo, con lo que los resultados son válidos.

Para probar la celda de carga y su circuito de amplificación se ha colgado la celda de un amarre disponible del taller y se han colgado cargas de peso conocidas. Se ha comparado las mediciones con los valores reales de la carga y los resultados son los siguientes:

Peso colgado	Valor analógico	Valor analógico teórico	Error relativo	Error relativo medio
0	644	639	0,83%	0,77%
2,832	810	816	0,76%	
4,066	897	894	0,39%	
7,198	1100	1090	0,94%	
10,246	1293	1281	0,95%	
13,098	1445	1460	0,99%	
16,23	1646	1656	0,59%	
17,464	1745	1733	0,69%	

Ilustración 43. Resultados experimentales de la celda de carga

Al igual que con el sensor de Hall, todas las mediciones están por debajo del error relativo de 1%. El error relativo medio es 0.77%. Para calcular el valor analógico teórico se han hecho las mediciones obteniendo el valor analógico y se ha hecho una aproximación lineal de los resultados. Obtenida la ecuación de la recta se han calculado los valores teóricos y comparados con los reales. Se puede ver en la siguiente gráfica:

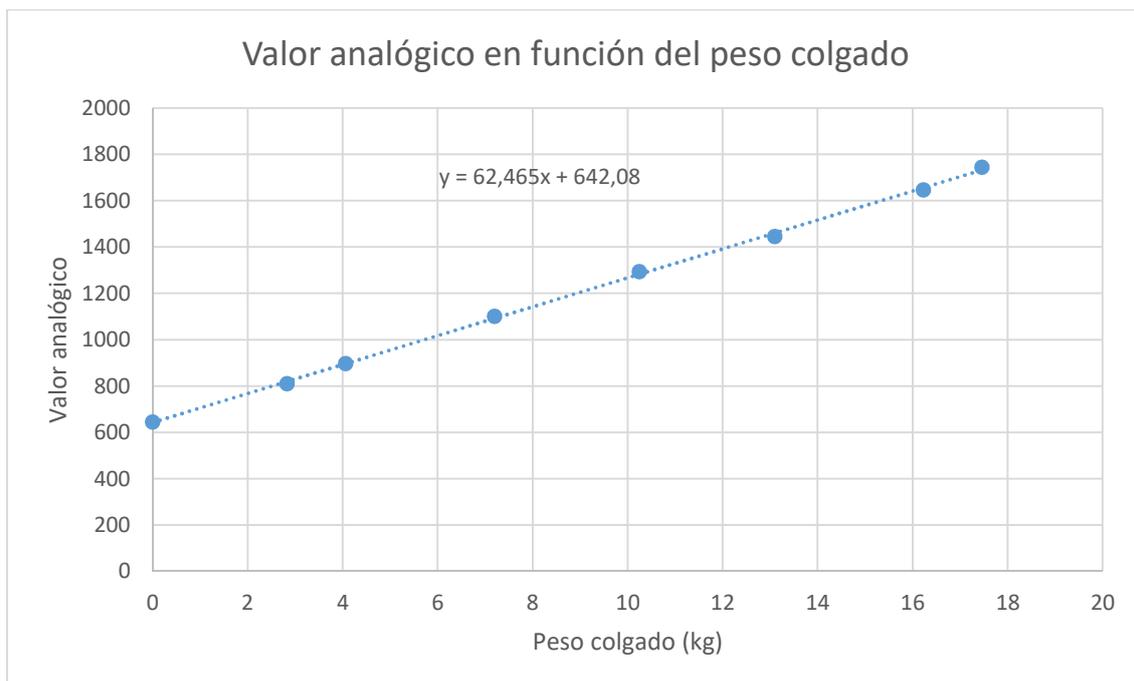


Ilustración 44. Representación gráfica de los resultados experimentales de la celda de carga

Como se puede observar los resultados cumplen satisfactoriamente los objetivos planteados. Las mediciones anteriores se han hecho por separado, pero también se ha probado a hacerlo con todo el conjunto, obteniendo resultados similares. El siguiente paso es realizar el montaje completo en el banco y hacer las pruebas finales con una moto en condiciones reales. Para ello es necesario montar todos los sensores en su posición definitiva.

No se ha podido realizar todo el montaje del banco aún ya que este año ha habido contratiempos en el equipo. Se ha roto el motor de la moto actual que se estaba montando y se ha perdido más tiempo del planificado, con lo que no se ha podido dedicar tanto tiempo a la construcción del banco. Sin embargo, aunque el banco no esté listo para obtener los resultados reales en este trabajo, se espera que no dará problemas ya que las condiciones a las que estará sometidas serán las que se ha probado.

2.8 Futuras mejoras. Segunda versión

Al igual que ocurre en todos los proyectos, al de un tiempo de haber tomado unas decisiones se descubre que había otras alternativas. En todos los proyectos hay fallos o simplemente mejoras que concluirían en un mejor resultado del mismo. Por ello, a medida

que se ha ido haciendo este proyecto se ha ido anotando las mejoras posibles que se ha detectado.

Estas mejoras pueden ser aprovechadas para diseñar una segunda placa a partir de la primera. Se ha visto que hay una lista de mejoras posible tanto en el diseño de la placa como en el software. Aquí se listarán todos los aspectos que se pueden mejorar:

En cuanto al diseño del circuito electrónico se han encontrado las siguientes mejoras:

- El arduino ocupa la parte central de la placa en uno de los dos lados, y el otro lado casi no está aprovechado. Todo ese sitio podría aprovecharse colocando algunos componentes en esa zona y el tamaño de la placa se reduciría considerablemente.
- Escoger un tamaño adecuado para los potenciómetros R16 y R13 ya que el tipo de potenciómetro que se ha colocado en la placa es difícil de encontrar.
- Escoger el optoacoplador U2 del tipo THT ya que, por un fallo a la hora de elegir el tipo de optoacoplador, se escogió del tipo SMD. Esto complica el proceso de soldar el componente
- Duplicar el tamaño de los terminales a los que se conectan los sensores por el lateral izquierdo de la placa ya que son demasiado estrechos y complica la conexión de los cables
- Aumentar el tamaño de los condensadores salvo los electrolíticos porque la huella puesta en la placa no encaja con los condensadores montados.
- En las alimentaciones no hace falta dos tierras por fuente ya que solo tiene 1. Solamente hay 3 terminales por fuente de alimentación y no 4. Por lo tanto, sobra 1 terminal de conexión a tierra y la inductancia que se ha colocado después
- Conectar el LCD mediante el protocolo de comunicación I2C ya que ofrece esa posibilidad acoplándole un pequeño módulo a la pantalla. De esta manera el número de conexiones al arduino serían 4 en vez de 12. La conexión sería mucho más simple porque se reduce en 8 el número de cables utilizados. Haría falta menos espacio para terminales de conexión, menos riesgo a que un cable se desconecte y menos pines de arduino utilizados. La programación de la pantalla sería la misma, pero añadiendo una librería diferente.
- Orientar todos los nombres de los componentes en la placa para que sea más sencillo el montaje. Alguna etiqueta está orientada en otro sentido en comparación al resto.

Markel Lizarralde

- Desplazar ligeramente los pines de expansión de la derecha todavía más porque el arduino roza con su borde alguno de esos pines y no se podrían utilizar
- Alejar un poco más algunos componentes entre ellos como los componentes R1 y U1 y los condensadores C1, C4 y C5.
- Aumentar el grosor todavía más de las pistas que estén separadas para reducir el calentamiento de las pistas y aumentar la eficiencia de las mismas.
- Poner etiquetas a los pines de expansión del Arduino y a las alimentaciones para facilitar las conexiones
- Poner pines de expansión para los diferentes tipos de comunicación que ofrece el arduino como el I2C, SPI, CAN y SERIAL. Estos pines son muy utilizados en diferentes módulos que se pueden conectar al arduino.

En cuanto al Software programado se ha descubierto lo siguiente:

- La consigna de revoluciones solamente se puede cambiar con el potenciómetro de la placa. Sería más cómodo colocar una casilla en el programa del ordenador para poder introducir ahí las revoluciones exactas y este le envíe esa consigna al arduino. Para ello haría falta que la comunicación fuera bidireccional ya que se enviarían datos del arduino al Matlab y viceversa.
- Configurar otro modo de operación en el banco. El funcionamiento se ha preparado de manera que se establezca una consigna de revoluciones y apertura de mariposa y variando parámetros de la ECU se observa si la potencia del motor aumenta o no. Otra forma de trabajar sería haciéndolo al revés. Se establece una consigna de potencia y apertura de mariposa, de manera que es el freno el que mantiene la carga constante y se observa si aumenta o no las revoluciones del motor en vez de la potencia al variar los parámetros de la ECU. El ajuste final debería ser el mismo, solo que se ha hecho de la manera inversa.
- Generar un archivo de texto con todos los datos llegados del arduino para después representarlos en un Excel o cualquier otro programa. De esta manera habría un registro de todo el trabajo realizado.

Como se puede ver hay una gran lista de mejoras posibles. No obstante, todos estos puntos solamente son mejoras y no fallos, lo que es muy satisfactorio. Aunque se mejoraran todos los puntos comentados el banco seguiría funcionando correctamente y de la misma

manera. Todas las mejoras anteriores son cosa de estética, comodidad a la hora de soldar los componentes y a la hora de utilizar el banco.

3. ASPECTOS ECONÓMICOS. PRESUPUESTO

En este apartado se valora económicamente todos los elementos necesarios para la consecución del proyecto. También se muestra el presupuesto realizado para indicar la cuantía de la inversión necesaria para poder llevar a cabo la instalación completa del banco. En resumen, se valoran todos los costes necesarios para diseñar y montar el banco de potencia. A continuación se muestra una tabla en la que figura la inversión necesaria desglosada por partidas. Este presupuesto se encuentra detallado en el Anexo 4.

PRESUPUESTO					
Concepto	Nº unidades		Coste unitario	Coste	% del coste total
Horas internas				4.500,00 €	49%
Ingeniería	150	h	30,00 €	4.500,00 €	49%
Gastos				3.727,66 €	41%
Parte mecánica				2.750,00 €	30%
Parte electrónica				977,66 €	11%
COSTES DIRECTOS				8.227,66 €	90%
Imprevistos	10%			914,18 €	10%
TOTAL	-	-	-	9.141,84 €	100%

Ilustración 45. Presupuesto resumido

Para el coste de las horas internas se ha tenido en cuenta que el TFG tiene un valor de 6 créditos. Como cada crédito equivale a 25 horas de trabajo del alumno suma un total de 150 horas. Se ha puesto un coste unitario de 30€/h del ingeniero para estimar su coste. Juntando las horas internas con los gastos tanto mecánicos como electrónicos el coste total del proyecto supera ligeramente los 9000€. A priori puede parecer caro, pero bancos de potencia comerciales con las mismas características pueden llegar a costar hasta 15000€. De todo el coste prácticamente la mitad son las horas de ingeniería, coste que realmente no ha habido ya que este trabajo se ha hecho de manera voluntaria en el equipo de MotoStudent. Por lo tanto, realmente ese coste no debería aparecer en el presupuesto siendo el coste total del proyecto casi 5000€. Se ha puesto un 10% de imprevistos como margen de seguridad. El presupuesto detallado con todos los costes mecánicos y electrónicos se puede encontrar en el apartado de anexos correspondiente.

4. CONCLUSIONES

Diseñar y fabricar un banco de potencia completo ha sido un gran reto en todos los sentidos. Han sido necesarios conocimientos mecánicos, electrónicos, programación y muchos más para la consecución del proyecto.

El objetivo principal era el desarrollo de un sistema electrónico para la medición electrónica del par motor y la potencia de una moto en un banco de pruebas y así ha sido. Todavía quedan cosas por mejorar, pero el plazo para terminarlo es hasta octubre, antes de la competición, ya que es para lo que se ha trabajado. De todas maneras, todos los detalles que quedan para terminarlo son cosa del montaje mecánico, aspecto que no entraba en este trabajo de fin de grado. La parte que se presenta en este trabajo es el diseño de la placa y software de control y se ha terminado en los plazos establecidos.

Los objetivos secundarios como medir las revoluciones, potencia y par motor también se han podido cumplir con un margen de error menor al establecido. El objetivo era un error menor al 5% y se ha bajado del 1%.

Se ha conseguido controlar con eficacia un freno de Foucault para regular la carga del banco desde el microcontrolador arduino. El tiempo de respuesta de la parte de control del freno es del orden de milisegundos y tiene un voltaje estable.

El entorno gráfico diseñado en Matlab permite ver con claridad los datos procedentes de los sensores a tiempo real, de manera que se puede observar fácilmente si se está aumentando la potencia del motor o no. Aunque también se haya implementado la opción de ver los datos en un LCD, gráficamente se aprecian mucho mejor los cambios.

El coste total del proyecto ha sido mucho más barato que un banco de potencia comercial y solamente se ha fabricado 1 unidad, aspecto que siempre encarece los proyectos. Se ha podido llegar a ahorrar hasta 10000€, en función de que banco se comprara si no se hubiera fabricado uno.

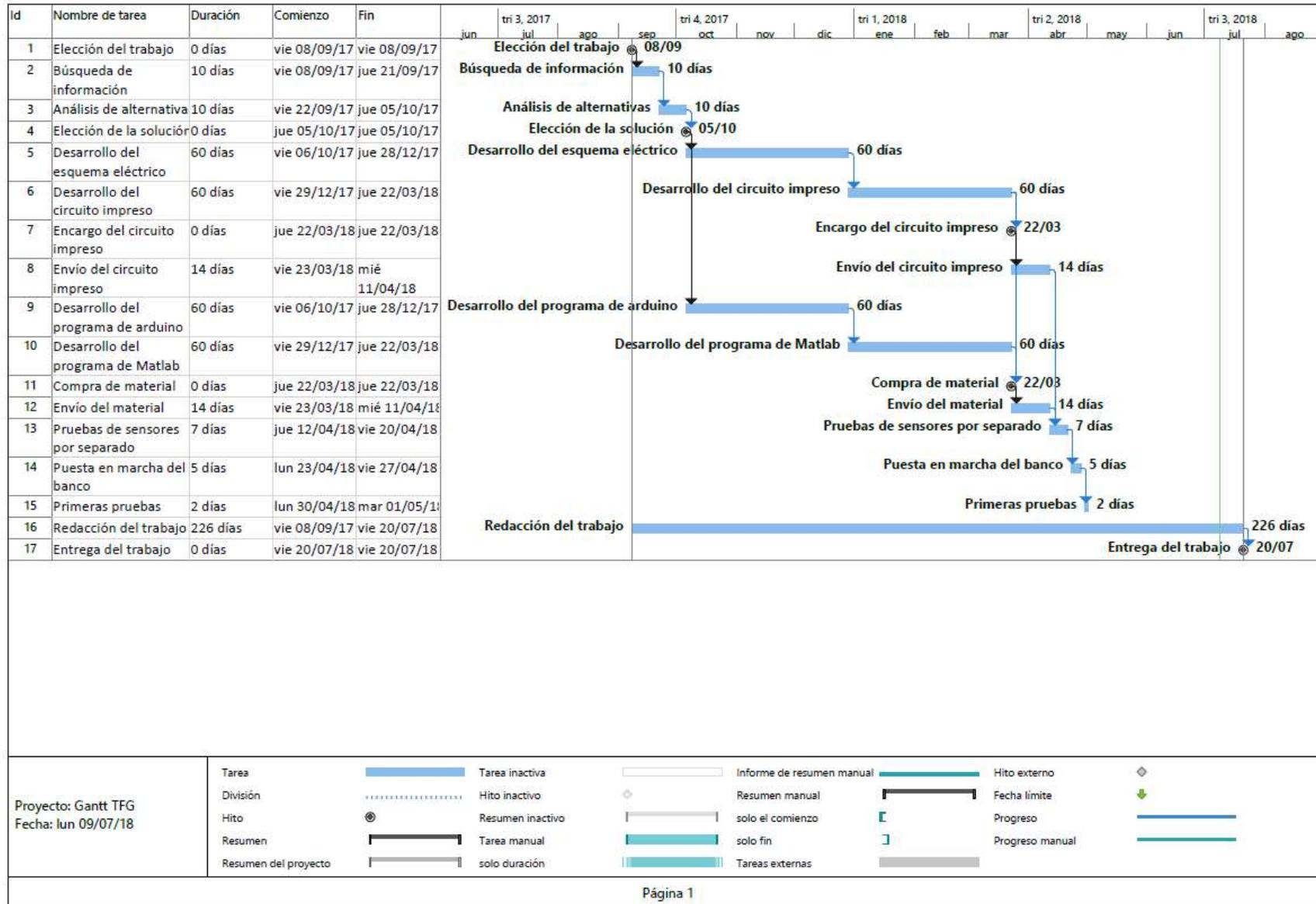
En conclusión, el trabajo propuesto se ha llevado a cabo con éxito cumpliendo los objetivos planteados, tanto los principales como los secundarios. Además, no solo se ha hecho el trabajo propuesto, sino que también se plantean mejoras para un futuro banco o una segunda versión mejorando la actual.

5. REFERENCIAS

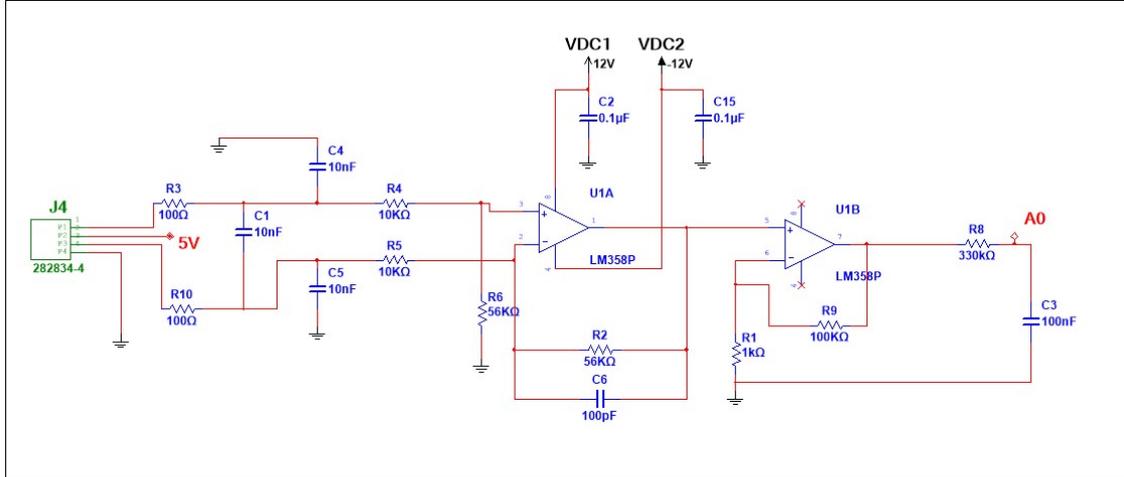
1. (s.f.). *TechnoPark MotorLand*. Obtenido de Moto Engineering Foundation:
<http://www.motostudent.com/about-motostudent.html>
2. (24 de Abril de 2018). *Wikipedia*. Obtenido de
https://es.wikipedia.org/wiki/Unidad_de_control_de_motor
3. (s.f.). *Orozco, Gilbert Mauricio García*. Obtenido de Prueba de ruta:
<https://www.pruebaderuta.com/mapeo-de-inyeccion-y-encendido.php>
4. (s.f.). m400, Software Motec.
5. (1 de abril de 2008). *Taller Virtual*. Obtenido de
<https://www.actualidadmotor.com/que-es-el-avance-de-encendido/>
6. (2017-2018). *KTM RC250 technical specs*.
7. (26 de enero de 2012). *La máquina de la verdad - Tipos de Banco de Potencia*. Obtenido de
<http://super7moto.com/directorio/revista/reportaje/14-la-maquina-de-la-verdad.html?showall=&start=1>
8. (4 de octubre de 2008). *Makina eta Motore Termikoetako Saila*. Obtenido de Banco de ensayo de Motores: <http://www.ehu.eus/mmtde/bancomot.htm>
9. (s.f.). *¿Qué es una célula de carga y cómo funciona?* Obtenido de Fundamentos de las células de carga: <https://es.omega.com/prodinfo/celulas-de-carga.html>
10. (23 de octubre de 2016). *Luis Llamas*. Obtenido de DETECTOR DE METALES CON ARDUINO Y SENSOR INDUCTIVO: <https://www.luisllamas.es/detector-de-metales-con-arduino-y-sensor-inductivo/>
11. (s.f.). *Frenelsa*. Obtenido de Bancos de ensayos: <http://www.frenelsa.es/es/producto-aplicaciones/be-gama.asp>
12. (s.f.). *iedelectronics*. Obtenido de Variador de corriente continua:
[http://www.iedelectronics.com/assets/files/otros-prod/9.CATALOGO-VAR10-40F-Preliminar\(ES\).pdf](http://www.iedelectronics.com/assets/files/otros-prod/9.CATALOGO-VAR10-40F-Preliminar(ES).pdf)
13. (s.f.). *Arduino*. Obtenido de Due: <https://store.arduino.cc/usa/arduino-due>

14. (s.f.). *RSCC*. Obtenido de Celdas de carga, Datasheet:
<http://www.cestus.net/attaches/20/RSCC.pdf>
15. (s.f.). *Hamlin*. Obtenido de Hall Effect Sensors:
https://www.mouser.es/datasheet/2/240/Littelfuse_Hall_Effect_Sensors_55075_Datasheet.pdf-938853.pdf
16. (s.f.). *Traco Power*. Obtenido de TMM 24 Series:
<https://www.tracopower.com/products/browse-by-category/find/tmm-24/3/>
17. (28 de abril de 2016). *Daniel Navarro*. Obtenido de Transmisiones de moto: ¿Cadena, correa o cardan?:
<https://www.formulamoto.es/reportajes/2016/04/28/transmisiones-moto-cadena-correa-o-cardan/14961.html>
18. (12 de noviembre de 2012). *Lasheras*. Obtenido de BANCOS DE POTENCIA, UNA APROXIMACIÓN TEÓRICA: <http://8000vueltas.com/2012/11/12/bancos-de-potencia>
19. (s.f.). *MOTOR DE EXPLOSIÓN DE 4 TIEMPOS – QUÉ ES, CÓMO FUNCIONA UN MOTOR DE EXPLOSIÓN DE 4 TIEMPOS*. Obtenido de <https://espaciocoche.com/motor-de-explosion-de-4-tiempos/>
20. (s.f.). *PAR MOTOR - Definición - Significado*. Obtenido de <https://diccionario.motorgiga.com/diccionario/par-motor-definicion-significado/gmx-niv15-con195062.htm>
21. (s.f.). *AMPLIFICADORES OPERACIONALES*. Obtenido de http://www.ifent.org/temas/amplificadores_operacionales.asp
22. (s.f.). *Arduino tutorials*. Obtenido de Display:
<https://www.arduino.cc/en/Tutorial/LiquidCrystalDisplay>

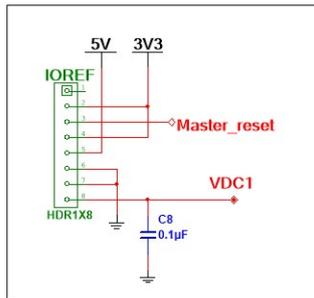
ANEXO 1. Diagrama de Gantt



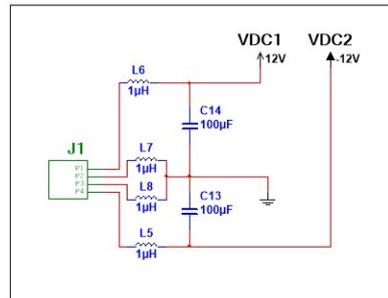
ANEXO 2. Esquema eléctrico



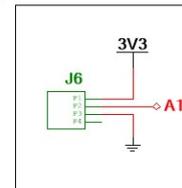
Parte que se encarga de amplificar la señal de la celda de carga y leerla con un pin analógico del Arduino



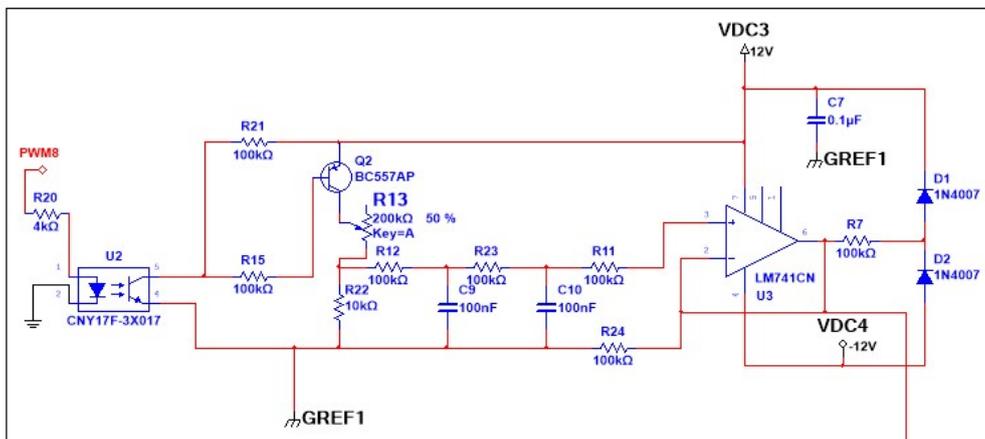
Pines del arduino



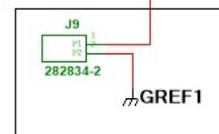
Alimentación de la parte general del circuito



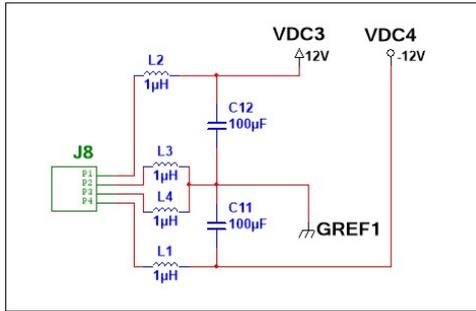
Potenciómetro para regular las revoluciones a las que queremos frenar la moto



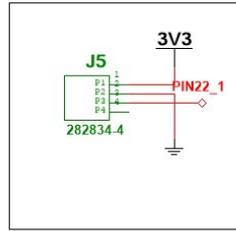
Parte aislada encargada de controlar el freno de Foucault



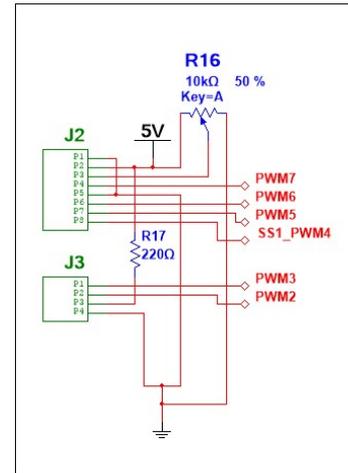
Salida de control a la fuente del freno de Foucault



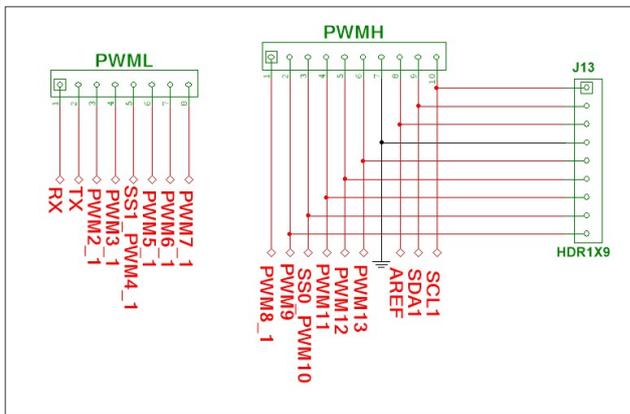
Alimentación de la parte del circuito que va aislada para controlar el freno de Foucault



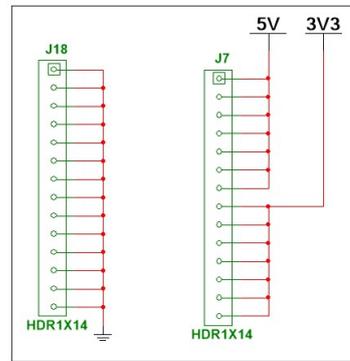
Hall para el tacómetro



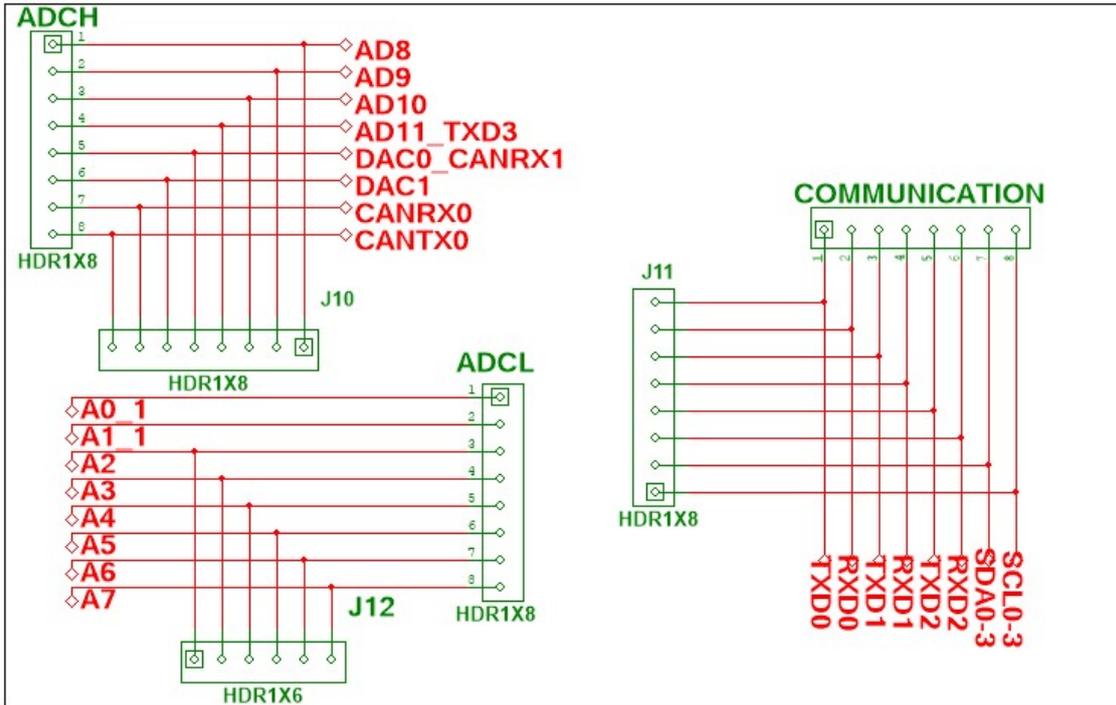
LCD para mostrar los valores medidos



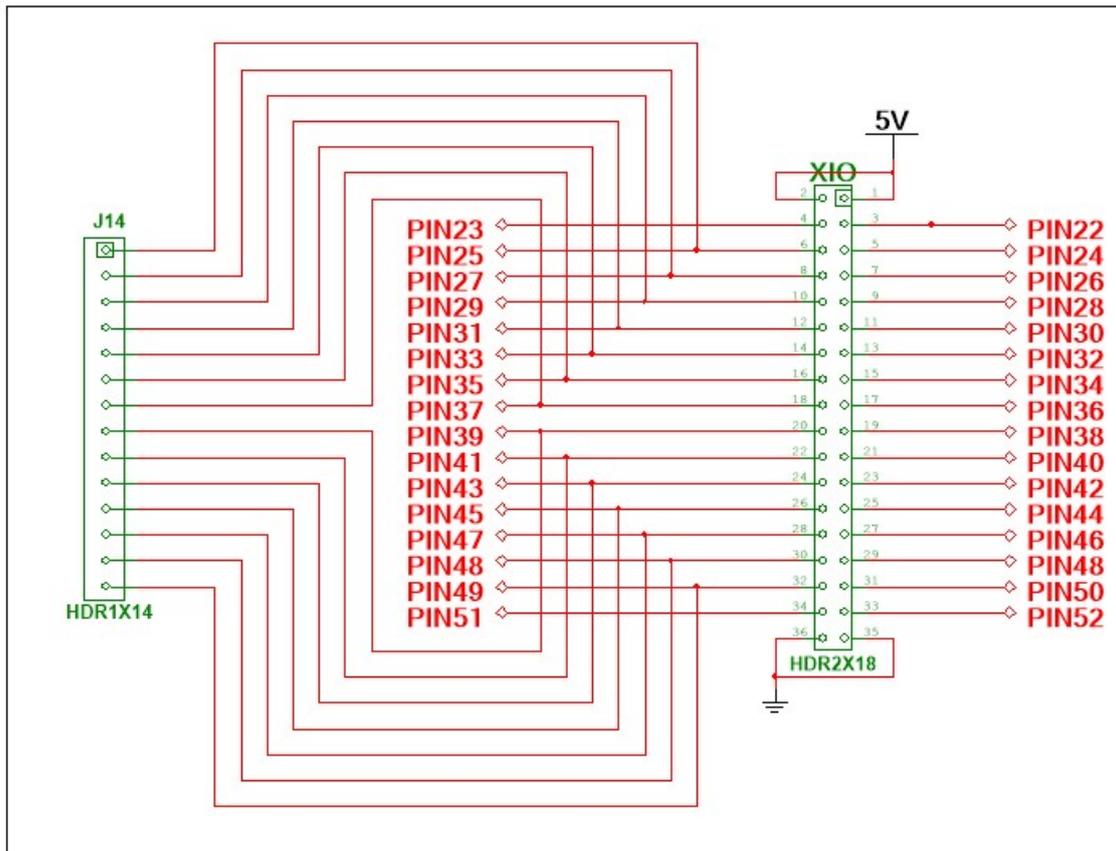
Pines de expansión del Arduino



Alimentaciones de salida extra a diferentes voltajes



Pines de expansión del Arduino



Pines de expansión del Arduino

ANEXO 4. Programa completo del arduino

```
#include <LiquidCrystal.h>

//Constantes que no cambian
const int NumeroDientes=3; //Dientes de la rueda fonica
const int VueltasMedidas=5;
const int PinHall=22;
const int PinPotenciometro=A1;
const int PinVoltajeFrenado=8;
const int PinCelda=A0;

const int FilaRPMDesead=0; //Posiciones de los valores en el LCD
const int FilaRPMActual=1;
const int FilaPorcentajeFrenado=2;
const int FilaParMotor=3;
const int ColumnaLCD=15;

//Variables que cambian de valor
```

```
int ContadorHall=0; //Variable que cuenta los dientes de la rueda fonica hasta que haya que refrescar y
calcular todo

unsigned long RPMActual=0; //rpm
unsigned long TiempoTotal=0; //Tiempo transcurrido hasta que haya que refrescar los datos y calcular las
rpm
unsigned long MuestraTiempo1=0; //Muestras que se toman para calcular el tiempo total
unsigned long MuestraTiempo2=0;

int ValorPotenciometro=0; //Valor con el que decimos a las RPM que queremos poner el motor
int RPMDeseada=0;

int ContadorCelda=0;
int VoltajeCelda=0;
int VoltajeActual=0; //Voltaje teorico que se esta aplicando para estar en las RPMActual
int VoltajeDeseado=0; //Voltaje que deberíamos aplicar para llegar a las rpm deseadas (suponiendo una
relacion de frenado/rpm lineal)
int VoltajeaAplicar=0; //Voltaje que se aplicará en el ciclo actual
int PorcentajeVoltajeFrenado=0; //Porcentaje que muestra cuanta parte del freno esta haciendo fuerza

//Declaramos el lcd
LiquidCrystal lcd(7,6,5,4,3,2);
```

```
void setup()  
{  
    //Iniciamos la comunicación serie  
    Serial.begin(57600);  
  
    //Configura la resolución de los pines a 12 bits. Por defecto esta a 10  
    analogWriteResolution(12);  
    analogReadResolution(12);  
  
    //Funcion creada (al final del programa)  
    IniciarPantallaLCD();  
  
    //Declara la función interrupción0 para que se ejecute cuando haya un cambio ascendente en la entrada  
    PinHall  
    attachInterrupt(PinHall, interrupcion0, RISING);  
}  
  
void loop()  
{  
    //Comprueba si ha dado las vueltas necesarias para calcular las RPM  
    //Si las ha dado resetea calcula el resto de variables y resetea contadores
```

```
if (ContadorHall==(VueltasMedidas*NumeroDientes))
{
    MuestraTiempo2=micros();
    TiempoTotal=(MuestraTiempo2-MuestraTiempo1);

    //Parte que se encarga de medir el voltaje que le llega de la celda de carga
    for (int i=1;i<=10;i++)
    {
        VoltajeCelda=VoltajeCelda+analogRead(PinCelda);
        ContadorCelda++;
    }
    VoltajeCelda=VoltajeCelda/ContadorCelda;
    ContadorCelda=0;

    //Calcula las RPM actuales
    RPMActual=((VueltasMedidas*60000000)/TiempoTotal);

    //Calcula la consigna de las RPM deseadas
    for (int i=1;i<=10;i++)
    {
        ValorPotenciometro=ValorPotenciometro+analogRead(PinPotenciometro);
    }
}
```

```
}  
ValorPotenciometro=ValorPotenciometro*0.1;  
RPMDeseada=map(ValorPotenciometro,0,4095,1,3000);  
  
//Calcula el frenado del motor y lo envia a la fuente que alimenta el freno  
VoltajeActual=map(RPMActual,0,3000,0,4095);  
VoltajeDeseado=map(RPMDeseada,0,3000,0,4095);  
VoltajeaAplicar=VoltajeaAplicar+VoltajeActual-VoltajeDeseado;  
//Establece los límites superiores e inferiores del voltaje del freno  
if(VoltajeaAplicar>4095)  
{  
    VoltajeaAplicar=4095;  
}  
else if(VoltajeaAplicar<0)  
{  
    VoltajeaAplicar=0;  
}  
//Pasa el voltaje de frenado a porciento de la capacidad de freno  
PorcentajeVoltajeFrenado=map(VoltajeaAplicar,0,4095,0,100);  
analogWrite(PinVoltajeFrenado,VoltajeaAplicar);
```

Markel Lizarralde

```
//Muestra todos los datos con el metodo que hayamos elegido: Puede ser por serial, graficamente o
por el LCD

//En esta parte lo mostraríamos con el puerto serial
/*Serial.print("RPMActual: ");
Serial.print(RPMActual);
Serial.print("  RPMDeseada: ");
Serial.print(RPMDeseada);
Serial.print("  PorcentajeVoltajeFrenado: ");
Serial.print(PorcentajeVoltajeFrenado);
Serial.print("  VoltajeCelda: ");
Serial.println(VoltajeCelda);*/

//En esta parte lo mostraríamos con el programa de Matlab
Serial.print("R"+String(RPMActual)+"\n");
Serial.print("C"+String(RPMDeseada)+"\n");
Serial.print("F"+String(PorcentajeVoltajeFrenado)+"\n");
Serial.print("P"+String(VoltajeCelda)+"\n");

//En esta parte lo mostraríamos con el LCD
ActualizarLCD(RPMActual,ColumnaLCD,FilaRPMActual); //Funcion creada (al final del programa)
ActualizarLCD(RPMDeseada,ColumnaLCD,FilaRPMDeseada); //Funcion creada (al final del programa)
```

```
    RefrescarLCD(PorcentajeVoltajeFrenado,ColumnaLCD,FilaPorcentajeFrenado); //Funcion creada (al
final del programa)
    RefrescarLCD(VoltajeCelda,ColumnaLCD,FilaParMotor); //Funcion creada (al final del programa)

    //Resetea algunos valores para el siguiente ciclo
    TiempoTotal=0;
    ContadorHall=0;

    //Recalcula el tiempo de muestreo para que tenga en cuenta el tiempo que ha perdido mostrando los
datos
    MuestraTiempo1=micros();
}
}
//Interrupción que se ejecuta cada vez que haya un cambio ascendente en la señal del sensor Hall
void interrupcion0()
{
    ContadorHall++;
}
//Funcion que se encarga de inicializar el LCD con todas las palabras salvo los valores a refrescar
void IniciarPantallaLCD()
{
    lcd.begin(20,4);
```

```
lcd.clear();  
lcd.noCursor();  
lcd.noBlink();  
  
lcd.setCursor(0,0);  
lcd.print("Rev deseada:      rpm");  
lcd.setCursor(0,1);  
lcd.print("Rev actual :      rpm");  
lcd.setCursor(0,2);  
lcd.print("Potencia   :      Cv");  
lcd.setCursor(0,3);  
lcd.print("Par motor   :      Nm");  
}  
//Funcion que se encarga de refrescar cualquier valor del LCD  
void RefrescarLCD(int Valor,int Columna,int Fila)  
{  
    lcd.setCursor(Columna,Fila);  
    lcd.print("      ");  
    lcd.setCursor(Columna,Fila);  
    lcd.print(Valor);  
}
```

ANEXO 5. Programa completo del Matlab

```

classdef BancoDePotenciaTFG < matlab.apps.AppBase
    % Properties that correspond to app components
    properties (Access = public)
        UIFigure          matlab.ui.Figure          % UI Figure
        LabelDropDown     matlab.ui.control.Label   % Puerto
        ListaPuertos     matlab.ui.control.DropDown % COM1, CO...
        BotonDesconectar  matlab.ui.control.Button  % Desconectar
        BotonConectar     matlab.ui.control.Button  % Conectar
        GraficaRPM        matlab.ui.control.UIAxes  % RPM dese...
        GraficaPar        matlab.ui.control.UIAxes  % Par moto...
        Panel             matlab.ui.container.Panel  % Valores ...
        LabelNumericEditField matlab.ui.control.Label % RPM actu...
        MostradorRPMActuales matlab.ui.control.NumericEditField % [-Inf Inf]
        Label             matlab.ui.control.Label   % Par motor
        MostradorParMotor matlab.ui.control.NumericEditField % [-Inf Inf]
        Label2            matlab.ui.control.Label   % Porcenta...
        MostradorPorcentajeFrenado matlab.ui.control.NumericEditField % [-I...
        Label3            matlab.ui.control.Label   % Potencia
        MostradorPotencia matlab.ui.control.NumericEditField % [-Inf Inf]
        Label4            matlab.ui.control.Label   % RPM dese...
        MostradorRPMDeseadas matlab.ui.control.NumericEditField % [-Inf Inf]
        LabelLinearGauge  matlab.ui.control.Label   % Porcenta...
        IndicadorFrenado  matlab.ui.control.LinearGauge % [0 100]
    end
    properties (Access = public)
        % Creamos las variables del programa
  
```

```
puerto_serial
puertoSeleccionado
tramaRecibida
mensajeProcesado
Par
RPMDeseadas
RPMActuales
PorcentajeFrenado
indicePar
indiceRPMActuales
indiceRPMDeseadas
indicePorcentajeFrenado
indicePotencia
end

methods (Access = public)
% Función que se ejecuta constantemente cuando el Arduino está conectado
function Conectado(app)
% Se crea un bucle que solo terminará si se corta la conexión
while 1
% Lee los bytes disponibles para leer
bytesDisponibles=app.puerto_serial.BytesAvailable;
% Si hay bytes lee hasta que haya llegado una línea
if bytesDisponibles>0
app.tramaRecibida=fgetl(app.puerto_serial);
app.mensajeProcesado=0;
end

% Si ya tiene una línea para procesar decodificamos el dato
% Clasifica las variables en función del identificador con el que llega el mensaje
```

```
if app.mensajeProcesado==0;
    switch app.tramaRecibida(1)
        % RPM actuales
        case 'R'

app.RPMActuales(app.indiceRPMDeseadas)=str2double(app.tramaRecibida(2:length(app.tramaRecibida)));
app.MostradorRPMActuales.Value=app.RPMActuales(app.indiceRPMDeseadas);

        if app.indiceRPM==50
            app.RPMActuales(1)=app.RPMActuales(app.indiceRPMDeseadas);
            app.indiceRPMDeseadas=1;
            app.RPMActuales(2:50)=zeros(1,49);
        end

plot(app.GraficaRPM,1:app.indiceRPMActuales,app.RPMDeseadas(1:app.indiceRPMDeseadas),1:app.indiceRPMDeseadas,app.RPMActuales(1:app.indiceRPMActuales));

        app.indiceRPM=app.indiceRPM+1;

        % Consigna de RPMs
        case 'C'

app.RPMDeseadas(app.indiceRPMDeseadas)=str2double(app.tramaRecibida(2:length(app.tramaRecibida)));
app.MostradorRPMDeseadas.Value=app.RPMDeseadas(app.indiceRPMDeseadas);

        if app.indiceRPMDeseadas==50
            app.RPMDeseadas(1)=app.RPMDeseadas(app.indiceRPMDeseadas);
            app.indiceRPMDeseadas=1;
            app.RPMDeseadas(2:50)=zeros(1,49);
        end
end
```

```
plot(app.GraficaRPM,1:app.indiceRPMDeseadas,app.RPMDeseadas(1:app.indiceRPMDeseadas),1:app.indiceRPMDeseadas,app.RPMAActuales(1:app.indiceRPMActuales));

    app.indiceRPMDeseadas=app.indiceRPMDeseadas+1;

    % Potencia
    case 'P'

app.Potencia(app.indicePotencia)=str2double(app.tramaRecibida(2:length(app.tramaRecibida)));
    app.mostradorPotencia.Value=app.Potencia(app.indicePotencia);

    if app.indicePotencia==50
        app.indicePotencia=1;
        app.indicePotencia=zeros(1,50);
    end

    plot(app.GraficaPar,1:app.indicePotencia,app.Potencia(1:app.indicePotencia));

    app.indicePar=app.indicePar+1;

    % Porcentaje de frenado
    case 'F'

app.PorcentajeFrenado(app.indicePorcentajeFrenado)=str2double(app.tramaRecibida(2:length(app.tramaRecibida)));

app.MostradorPorcentajeFrenado.Value=app.PorcentajeFrenado(app.indicePorcentajeFrenado);

    if app.indicePorcentajeFrenado==50
        app.indicePorcentajeFrenado=1;
```

```
        app.PorcentajeFrenado=zeros(1,50);
    end

    app.IndicadorFrenado.Value=app.PorcentajeFrenado(app.indicePorcentajeFrenado);

    app.indicePorcentajeFrenado=app.indicePorcentajeFrenado+1;

    %    Par motor
    case 'M'

app.ParMotor(app.indiceParMotor)=str2double(app.tramaRecibida(2:length(app.tramaRecibida)));
        app.MostradorParMotor.Value=app.ParMotor(app.indiceParMotor);

        if app.indiceParMotor==50
            app.indiceParMotor=1;
            app.indiceParMotor=zeros(1,50);
        end
        app.indiceParMotor=app.indiceParMotor+1;
    end

    %    Cambia la variable para indicar que ya se ha decodificado el mensaje
    app.mensajeProcesado=1;
end

    %    Dejamos un breve periodo para que el programa compruebe si se ha llamado a otras funciones
    %    Una función puede ser la de presionar el botón de desconectar el Arduino
    pause(0.001);
end
end

%    Función que se ejecuta cuando presionamos el botón de conectar el Arduino
```

```
function IniciarComunicacion(app)
    %Inicializo el puerto serial que utilizaré
    delete(instrfind({'Port'},{app.puertoSeleccionado}));
    app.puerto_serial=serial(app.puertoSeleccionado);
    app.puerto_serial.BaudRate=9600;
    warning('off','MATLAB:serial:fscanf:unsuccessfulRead');

    %Abro el puerto serial
    fopen(app.puerto_serial);

    %Espero 1 segundo para que se inicie la comunicación
    pause(1);

    Conectado(app);
end

% Función que se ejecuta cuando presionamos el botón de desconectar el Arduino
function PararComunicacion(app)
    %Cierro la conexión con el puerto serial y elimino las variables
    fclose(app.puerto_serial);
    %delete(app.puerto_serial);
    %clear(app.puerto_serial);
    while 1
        pause(0.01);
    end
end
end

methods (Access = private)
    % Code that executes after component creation
    function startupFcn(app)
```

```
% Es lo primero que se ejecuta del programa
% Iniciamos las variables necesarias
app.mensajeProcesado=1;
app.MostradorRPMDeseadas.Value='0';
app.Par=zeros(1,50);
app.RPMDeseadas=zeros(1,50);
app.RPMActuales=zeros(1,50);
app.PorcentajeFrenado=zeros(1,50);
app.indicePar=1;
app.indiceRPM=1;
app.indicePorcentajeFrenado=1;

% Establezco los límites de los ejes de las gráficas
xlim(app.GraficaPar,[0,50]);
xlim(app.GraficaRPM,[0,50]);
ylim(app.GraficaPar,[0,4095]);
ylim(app.GraficaRPM,[0,5000]);
end
% ListaPuertos value changed function
function ListaPuertosValueChanged(app, event)
    app.puertoSeleccionado = app.ListaPuertos.Value;
end
% BotonDesconectar button pushed function
function BotonDesconectarButtonPushed(app)
    PararComunicacion(app);
end
% BotonConectar button pushed function
function BotonConectarButtonPushed(app)
    IniciarComunicacion(app);
end
```

```
end
% App initialization and construction
methods (Access = private)
    % Create UIFigure and components
    function createComponents(app)
        % Create UIFigure
        app.UIFigure = uifigure;
        app.UIFigure.Position = [100 100 1062 740];
        app.UIFigure.Name = 'UI Figure';
        setAutoResize(app, app.UIFigure, true)
        % Create LabelDropDown
        app.LabelDropDown = uilabel(app.UIFigure);
        app.LabelDropDown.HorizontalAlignment = 'right';
        app.LabelDropDown.Position = [15 706 36 15];
        app.LabelDropDown.Text = 'Puerto';
        % Create ListaPuertos
        app.ListaPuertos = uidropdown(app.UIFigure);
        app.ListaPuertos.Items = {'COM1', 'COM2', 'COM3', 'COM4', 'COM5', 'COM6', 'COM7', 'COM8', 'COM9',
'COM10', 'COM11', 'COM12', 'COM13', 'COM14'};
        app.ListaPuertos.ValueChangedFcn = createCallbackFcn(app, @ListaPuertosValueChanged, true);
        app.ListaPuertos.Position = [66 704 100 20];
        app.ListaPuertos.Value = 'COM1';
        % Create BotonDesconectar
        app.BotonDesconectar = uibutton(app.UIFigure, 'push');
        app.BotonDesconectar.ButtonPushedFcn = createCallbackFcn(app, @BotonDesconectarButtonPushed);
        app.BotonDesconectar.Position = [184 702 100 22];
        app.BotonDesconectar.Text = 'Desconectar';
        % Create BotonConectar
        app.BotonConectar = uibutton(app.UIFigure, 'push');
        app.BotonConectar.ButtonPushedFcn = createCallbackFcn(app, @BotonConectarButtonPushed);
```

```
app.BotonConectar.Position = [301 702 100 22];
app.BotonConectar.Text = 'Conectar';
% Create GraficaRPM
app.GraficaRPM = uiaxes(app.UIFigure);
title(app.GraficaRPM, 'RPM deseadas y actuales');
xlabel(app.GraficaRPM, 'Muestra');
ylabel(app.GraficaRPM, 'RPM');
app.GraficaRPM.FontSize = 18;
app.GraficaRPM.FontWeight = 'bold';
app.GraficaRPM.Box = 'on';
app.GraficaRPM.Color = [0.7686 0.7608 0.7608];
app.GraficaRPM.XGrid = 'on';
app.GraficaRPM.YGrid = 'on';
app.GraficaRPM.Position = [0 0 758 240];
% Create GraficaPar
app.GraficaPar = uiaxes(app.UIFigure);
title(app.GraficaPar, 'Par motor y potencia');
xlabel(app.GraficaPar, 'Muestra');
ylabel(app.GraficaPar, 'Par motor Nm y potencia Cv');
app.GraficaPar.FontSize = 16;
app.GraficaPar.FontWeight = 'bold';
app.GraficaPar.Box = 'on';
app.GraficaPar.Color = [0.7882 0.7882 0.7882];
app.GraficaPar.XGrid = 'on';
app.GraficaPar.YGrid = 'on';
app.GraficaPar.Position = [0 280 1030 410];
% Create Panel
app.Panel = uipanel(app.UIFigure);
app.Panel.BorderType = 'line';
app.Panel.TitlePosition = 'centertop';
```

```
app.Panel.Title = 'Valores numéricos';
app.Panel.FontName = 'Helvetica';
app.Panel.FontUnits = 'pixels';
app.Panel.FontSize = 12;
app.Panel.Units = 'pixels';
app.Panel.Position = [782 83 260 190];
% Create LabelNumericEditField
app.LabelNumericEditField = uilabel(app.Panel);
app.LabelNumericEditField.HorizontalAlignment = 'right';
app.LabelNumericEditField.Position = [52 111 76 15];
app.LabelNumericEditField.Text = 'RPM actuales';
% Create MostradorRPMActuales
app.MostradorRPMActuales = uieditfield(app.Panel, 'numeric');
app.MostradorRPMActuales.Position = [143 107 100 22];
% Create Label
app.Label = uilabel(app.Panel);
app.Label.HorizontalAlignment = 'right';
app.Label.Position = [75 74 53 15];
app.Label.Text = 'Par motor';
% Create MostradorParMotor
app.MostradorParMotor = uieditfield(app.Panel, 'numeric');
app.MostradorParMotor.Position = [143 70 100 22];
% Create Label2
app.Label2 = uilabel(app.Panel);
app.Label2.HorizontalAlignment = 'right';
app.Label2.Position = [7 38 121 15];
app.Label2.Text = 'Porcentaje de frenado';
% Create MostradorPorcentajeFrenado
app.MostradorPorcentajeFrenado = uieditfield(app.Panel, 'numeric');
app.MostradorPorcentajeFrenado.Position = [143 34 100 22];
```

```
% Create Label3
app.Label3 = uilabel(app.Panel);
app.Label3.HorizontalAlignment = 'right';
app.Label3.Position = [80 9 48 15];
app.Label3.Text = 'Potencia';
% Create MostradorPotencia
app.MostradorPotencia = uieditfield(app.Panel, 'numeric');
app.MostradorPotencia.Position = [143 5 100 22];
% Create Label4
app.Label4 = uilabel(app.Panel);
app.Label4.HorizontalAlignment = 'right';
app.Label4.Position = [44 148 84 15];
app.Label4.Text = 'RPM deseadas';
% Create MostradorRPMDeseadas
app.MostradorRPMDeseadas = uieditfield(app.Panel, 'numeric');
app.MostradorRPMDeseadas.Position = [143 144 100 22];
% Create LabelLinearGauge
app.LabelLinearGauge = uilabel(app.UIFigure);
app.LabelLinearGauge.BackgroundColor = [1 1 1];
app.LabelLinearGauge.HorizontalAlignment = 'center';
app.LabelLinearGauge.Position = [851.5 9 121 15];
app.LabelLinearGauge.Text = 'Porcentaje de frenado';
% Create IndicadorFrenado
app.IndicadorFrenado = uigauge(app.UIFigure, 'linear');
app.IndicadorFrenado.Position = [803 31 217 40];
end
end
methods (Access = public)
% Construct app
function app = BancoDePotenciaTFG()
```

```
% Create and configure components
createComponents(app)
% Register the app with App Designer
registerApp(app, app.UIFigure)
% Execute the startup function
runStartupFcn(app, @startupFcn)
if nargin == 0
    clear app
end
end
% Code that executes before app deletion
function delete(app)
    % Delete UIFigure when app is deleted
    delete(app.UIFigure)
end
end
end
```

ANEXO 6. Presupuesto

PRESUPUESTO					
Concepto	Nº unidades		Coste unitario	Coste	% del coste total
Horas internas				4.500,00 €	49%
Ingeniería	150	h	30,00 €	4.500,00 €	49%
Gastos				3.727,66 €	41%
Parte mecánica				2.750,00 €	30%
Parte electrónica				977,66 €	11%
COSTES DIRECTOS				8.227,66 €	90%
Imprevistos	10 %			914,18 €	10%
TOTAL	-	-	-	9.141,84 €	100%

PRESUPUESTO PARTE ELECTRÓNICA					
Concepto	Nº unidades		Coste unitario	Coste	% del coste total
Freno de Foucault	1	unidades	300,00 €	300,00 €	3%
Fuente del freno	1	unidades	300,00 €	300,00 €	3%
Fuente de alimentación	2	unidades	72,48 €	144,96 €	2%
Placa electrónica	3	unidades	35,00 €	105,00 €	1%
Cables de alimentación	1	unidades	10,00 €	10,00 €	0%
Sensor Hall	1	unidades	25,00 €	25,00 €	0%
Celda de carga	1	unidades	50,00 €	50,00 €	1%
Pantalla LCD	1	unidades	12,00 €	12,00 €	0%
Terminal de conexión	8	unidades	0,50 €	4,00 €	0%
Pines de expansión	169	pinos	0,10 €	16,90 €	0%
Resistencia	19	unidades	0,10 €	1,90 €	0%
Condensador	15	unidades	0,20 €	3,00 €	0%
Inductor	8	unidades	0,20 €	1,60 €	0%
Potenciómetro	2	unidades	0,50 €	1,00 €	0%
Diodo	2	unidades	0,20 €	0,40 €	0%

Markel Lizarralde

Amplificador operacional	2	s unidade s	0,40 €	0,80 €	0%
Optoacoplador	1	unidade s	0,60 €	0,60 €	0%
Transistor	1	unidade s	0,50 €	0,50 €	0%