

Universidad de Costa Rica
Facultad de Ingeniería
Escuela de Ingeniería Eléctrica

**Diseño de un módulo de control para el
brazo robótico ARMDROID 2001 de
LabVolt**

Por:

José Ignacio Hidalgo Gómez

Ciudad Universitaria “Rodrigo Facio”, Costa Rica

Diciembre de 2013

Diseño de un módulo de control para el brazo robótico ARMDROID 2001 de LabVolt

Por:

José Ignacio Hidalgo Gómez

IE-0499 Proyecto eléctrico

Aprobado por el Tribunal:

Ing. Mauricio Espinoza Bolaños
Profesor guía

Dr. José David Rojas Fernández
Profesor lector

M.Sc. José Luis López Sauma
Profesor lector

Índice general

Índice de figuras	vi
1 Introducción	1
1.1 Alcance del proyecto	1
1.2 Objetivos	2
1.3 Metodología	2
2 Marco Teórico	5
2.1 Brazo robótico ARMDROID 2001	5
2.2 Modulación por ancho de pulso PWM	5
2.3 DC chopper	7
2.4 Plataforma de desarrollo	8
3 Implementación del módulo de control	11
3.1 Fase inicial de implementación	12
3.2 Implementación de hardware	14
3.3 Implementación de software	18
4 Validación del módulo de control	23
5 Conclusiones y recomendaciones	27
5.1 Conclusiones	27
5.2 Recomendaciones	27
Bibliografía	29
A Manual de usuario	31
A.1 Puesta en marcha	31
A.2 Operación del brazo robótico ARMDROID 2001	32
B Manual de programador	35
B.1 Operación del brazo robótico ARMDROID 2001	35
C Diagramas Esquemáticos	39

Índice de figuras

2.1	Brazo robótico ARMDROID 2001.	6
2.2	Funcionamiento del PWM.	7
2.3	Diagrama esquemático de un DC chopper	7
2.4	Circuito equivalente de un motor DC.	8
2.5	Plataforma de desarrollo Arduino Uno.	9
3.1	Esquema del módulo de control.	12
3.2	Motores DC utilizados por el ARMDROID 2001.	12
3.3	Pieza usada para el final de carrera del hombro.	13
3.4	Partes movibles del ARMDROID 2001.	14
3.5	Etapas de entrada para un único motor.	15
3.6	Etapas de salida para un único motor.	16
3.7	Módulo integrado del L298n.	16
3.8	Esquema de la unidad de hardware para un único motor.	18
3.9	Diagrama de bloques de la unidad de software.	19
3.10	Filtrado y normalización de la señal obtenida.	19
3.11	Obtención de la señal de posición.	20
3.12	Flip flop D implementado.	20
3.13	Interfaz gráfica vista del módulo de control.	21
3.14	Diagrama de bloques del módulo de control simplificado.	21
4.1	Respuesta de lazo abierto del brazo.	23
4.2	Respuesta de lazo abierto del hombro.	24
4.3	Respuesta en lazo cerrado del sistema de control de posición para el hombro.	24
4.4	Respuesta del control manual.	25
4.5	Respuesta de lazo cerrado con un controlador proporcional.	25
4.6	Mensaje de error desplegado por el programa.	26
A.1	Conexión del Arduino con las etapas de hardware para un sólo motor.	31
A.2	Interfaz gráfica del módulo de control.	32
B.1	Programa requerido para controlar un único motor.	35
B.2	Obtención de la señal de posición.	36
B.3	Bloque de control automático/manual.	37
B.4	Código utilizado para tomar la acción correctiva.	37

C.1	Diagrama esquemático completo de la etapa de entrada.	40
C.2	Diagrama esquemático completo de la etapa de salida.	41

1 Introducción

El Laboratorio de Automática, ubicado en la Escuela de Ingeniería Eléctrica de la Universidad de Costa Rica, posee una gran variedad de recursos, los cuales tienen el potencial para ser aprovechados en varios cursos que se imparten en la Escuela. Con el fin de aprovechar el brazo robótico ARMDROID 2001 de LabVolt se pretende diseñar un módulo de control completamente nuevo, debido a que el módulo de control con el que cuenta actualmente es poco versátil, es decir, el control manual resulta incómodo de utilizar, si se desea cargar rutinas de movimiento al brazo robótico se deben ingresar mediante un disquete de 3,5 pulgadas (actualmente esta tecnología es prácticamente obsoleta).

El módulo de control que se pretende diseñar comunicará el brazo robótico con una computadora de escritorio, al hacer esto se pretende facilitar el control, monitoreo, y programación del brazo robótico, debido a que se pretende diseñar una interfaz gráfica que facilite estas tareas.

1.1 Alcance del proyecto

El proyecto se centra en el diseño de un módulo de control para el brazo robótico ARMDROID 2001, con el cual sea posible manejar dicho robot desde una computadora, el módulo de control que se pretende diseñar contará con una interfaz gráfica que permita al usuario monitorear el estado del robot en todo momento. También se confeccionará un manual de usuario con la finalidad de que éste módulo de control pueda ser utilizado eventualmente en otros cursos impartidos por la Escuela de Ingeniería Eléctrica.

Se plantea el diseño de un nuevo módulo de control debido a que el módulo que utiliza el robot actualmente es poco práctico, debido a que el monitoreo del brazo robótico no es eficiente, el modo de control manual es engorroso y la programación del robot mediante un disquete es obsoleta, en síntesis, el proyecto girará en torno al diseño de un módulo de control para el brazo robótico ARMDROID 2001.

1.2 Objetivos

Objetivo general

Diseñar una interfaz gráfica y un sistema de control para el robot ARMDROID 2001.

Objetivos específicos

Para el desarrollo de este proyecto se establecieron los siguientes objetivos:

- Investigar el principio de funcionamiento mecánico del brazo ARMDROID 2001.
- Analizar el estado del brazo robótico ARMDROID 2001, y realizar reparaciones de ser necesario.
- Implementar la electrónica necesaria para el control de los motores del brazo robótico.
- Seleccionar una plataforma de desarrollo para la manipulación de la electrónica que controla los motores del brazo ARMDROID 2001.
- Desarrollar una interfaz gráfica tal que un usuario pueda operar el brazo robótico en forma segura.
- Crear un manual de usuario, para explicar los aspectos en el uso del brazo robótico.

1.3 Metodología

Por la forma en que están formulados los objetivos del proyecto, la metodología que se siguió fue la siguiente:

1. Se procedió a realizar una revisión general al estado del brazo, para investigar el mecanismo de funcionamiento que utiliza.
2. Se realizaron las reparaciones pertinentes con base en los resultados de la revisión realizada.
3. Se diseñó e implementó un circuito electrónico capaz de controlar los diversos motores que generan los movimientos del brazo robótico.
4. Se utilizó una plataforma de desarrollo que pudiera controlar el circuito de potencia diseñado anteriormente, y a su vez, fuera capaz de enviar las señales necesarias al programa principal ubicado en la computadora.

5. Se desarrolló una interfaz gráfica en la que se puedan recibir las señales enviadas por la plataforma de desarrollo, también debe ser capaz de enviar señales de control a la plataforma de desarrollo, con base en lo que el usuario o la rutina requieran.
6. Se confeccionó un manual de usuario y un manual de programador que pueden utilizarse como referencia para aquellos que requieran hacer uso del robot.

2 Marco Teórico

En esta sección se mencionan los conceptos teóricos necesarios para la síntesis del módulo de control. La sección 2.1 hace una breve definición de qué es un brazo robótico, sus aplicaciones y características de funcionamiento. La sección 2.2 discute sobre la modulación por ancho de pulso (PWM), la cual es una técnica ampliamente utilizada para el control de motores eléctricos. La sección 2.3 habla sobre las unidades de control de potencia conocidas como *DC chopper*, las cuales proporcionan la potencia que requieren las cargas que se conectan a estas unidades. Para finalizar, la sección 2.4 habla sobre las plataformas de desarrollo, en particular se habla sobre el Arduino UNO y su utilidad para este proyecto.

2.1 Brazo robótico ARMDROID 2001

Un robot es un dispositivo virtual o mecánico el cual lleva a cabo una función específica, en particular, se puede decir que los brazos robóticos son los robots que se fabrican con mayor frecuencia, debido a su capacidad para desempeñar labores repetitivas de manera rápida y eficiente, lo cual resulta de gran utilidad en las líneas de ensamblaje de grandes fábricas. La configuración de un brazo robótico depende directamente de la función que deba llevar a cabo, por lo general, estos robots utilizan motores eléctricos para realizar los diferentes movimientos que se requieren, tal y como mencionan Khatib y Siciliano (2008).

El brazo robótico ARMDROID 2001 (ver figura 2.1) fabricado por LabVolt se puede clasificar como un brazo esférico, es decir, el conjunto de todos los movimientos individuales que el robot es capaz de realizar forman una esfera. Éste robot utiliza seis motores DC de 24 V para realizar dichos movimientos, también posee cuatro interruptores, los cuales se activan cuando el brazo alcanza una posición no permitida, cuando alguno de estos interruptores se activa, el módulo de control detendrá por completo el movimiento que realizaba el robot en dicho instante, luego, invertirá la dirección del movimiento que el robot realizaba hasta que la señal enviada por el interruptor se desactive.

2.2 Modulación por ancho de pulso PWM

Según la Universidad Politécnica de Madrid (2013), la modulación por ancho de pulso o PWM (Pulse-Width Modulation) por sus siglas en inglés es una



Figura 2.1: Brazo robótico ARMROID 2001.

estrategia de conmutación, usualmente utilizada para manipular el ciclo de trabajo de una señal a una frecuencia de operación fija. Esto se realiza con la finalidad de modificar la potencia efectiva que alimenta a una carga determinada como por ejemplo un motor DC. El ciclo de trabajo puede definirse como el tiempo efectivo en el que una señal periódica determinada está activa con respecto a su período, con lo cual, el ciclo de trabajo puede expresarse como:

$$D = \frac{\tau}{T} \quad (2.1)$$

Donde T corresponde al período de la señal y τ corresponde al tiempo en alto de la señal cuadrada. En la figura 2.2 se muestra como el PWM se utiliza para modificar el valor DC promedio que se genera.

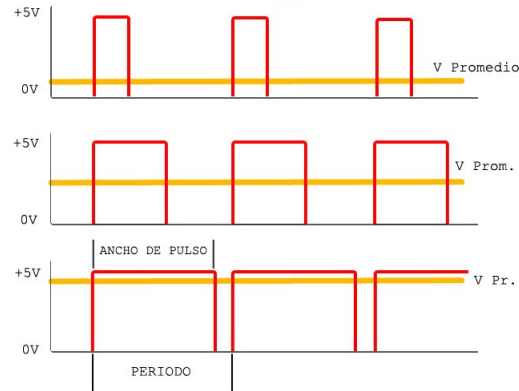


Figura 2.2: Funcionamiento del PWM.

2.3 DC chopper

El DC chopper es una unidad de control de potencia, la cual permite un flujo bidireccional de la corriente eléctrica utilizando sólo una fuente de tensión tal y como lo menciona Husain (2003). En la figura 2.3 se muestra la configuración

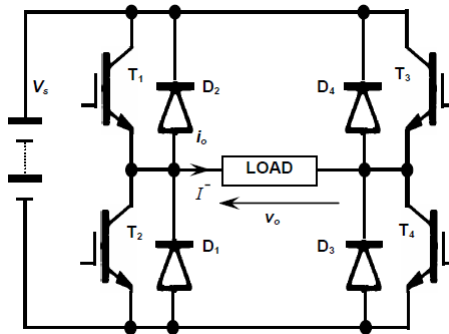


Figura 2.3: Diagrama esquemático de un DC chopper

de un DC chopper, los transistores se activan o desactivan dependiendo de la señal de control que se utiliza. En principio se puede decir que las partes que componen a un DC chopper son la fuente de alimentación de potencia, la unidad de control para los transistores, la unidad manejadora de potencia o puente h y de manera opcional puede considerarse un filtro pasivo el cual permita conmutaciones de tensión menos pronunciadas.

La señal de tensión V_s es la fuente de tensión de potencia, mientras que el arreglo de transistores controlan el flujo de corriente a través de la carga.

Puente H

El arreglo de transistores mostrado en la figura 2.3 es conocido como “*puente h*”. El puente h es un circuito electrónico utilizado generalmente para controlar motores de corriente continua. Con este esquema es posible controlar la dirección de giro de un motor de corriente continua, utilizando únicamente una sola señal de alimentación, la cual puede ser una señal PWM.

Motor DC

Se puede considerar a un motor DC o motor de corriente continua como una máquina que transforma una señal DC en par mecánico. El circuito equivalente considerado para el motor de corriente continua se muestra en la figura 2.4.

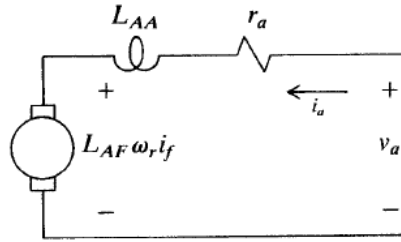


Figura 2.4: Circuito equivalente de un motor DC.

Donde L_{AA} representa la inductancia de armadura del motor y r_a la resistencia de armadura o resistencia del devanado, mientras que el factor $L_{AF}\omega_r i_f$ representa la fuerza contraelectromotriz que produce el motor. La tensión v_a y la corriente i_a corresponden a la tensión de alimentación de armadura y corriente de armadura que alimentan al motor respectivamente.

2.4 Plataforma de desarrollo

Se puede considerar que una plataforma de desarrollo es una unidad de hardware que permite el desarrollo de un proyecto determinado. Arduino es una plataforma de desarrollo en hardware libre, basada en una placa con un microcontrolador el cual posee diversas entradas y salidas (CENDITEL, 2013), existen diversas aplicaciones en las cuales se puede utilizar un Arduino, como conectarse al software de una computadora para utilizarse como una tarjeta de adquisición de datos. Existen diversos modelos de Arduino en el mercado, cada uno posee características diferentes, como el número de entradas o salidas, mayor capacidad de procesamiento entre otras.

En la figura 2.5 se muestra un Arduino Uno, el cual cuenta con 14 entradas/salidas digitales, de las cuales seis pueden proveer una señal PWM, también

cuenta con seis entradas analógicas y un procesador ATmega328 de 32 bits, con una frecuencia de 16 MHz.

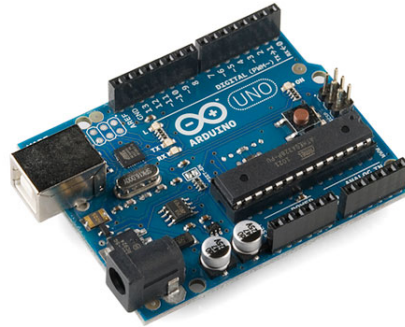


Figura 2.5: Plataforma de desarrollo Arduino Uno.

3 Implementación del módulo de control

En el presente capítulo se habla sobre el proceso de diseño del módulo de control, con los diferentes componentes tanto de software como de hardware que lo componen. En la sección 14.1 se discute sobre las condiciones iniciales con las que el proyecto dio inicio. La sección 14.2 menciona los elementos presentes en el sistema de adquisición de datos que se utiliza para realizar una comunicación apropiada entre el robot y el sistema de control. Por último, la sección 14.3 habla sobre el programa desarrollado para monitorear y controlar al brazo robótico.

El módulo de control debe ser capaz de realizar las siguientes tareas:

- Monitorear la posición de los motores que componen al brazo robótico en todo momento.
- Manejar la electrónica de potencia necesaria para alimentar los motores del robot.
- Manejar las interrupciones generadas por los interruptores del robot.
- Permitir al usuario la elección entre operación del robot, ya sea en forma automática o manual.
- Realizar rutinas definidas por el usuario.

En la figura 3.1 se muestra el esquema del módulo de control a implementar.

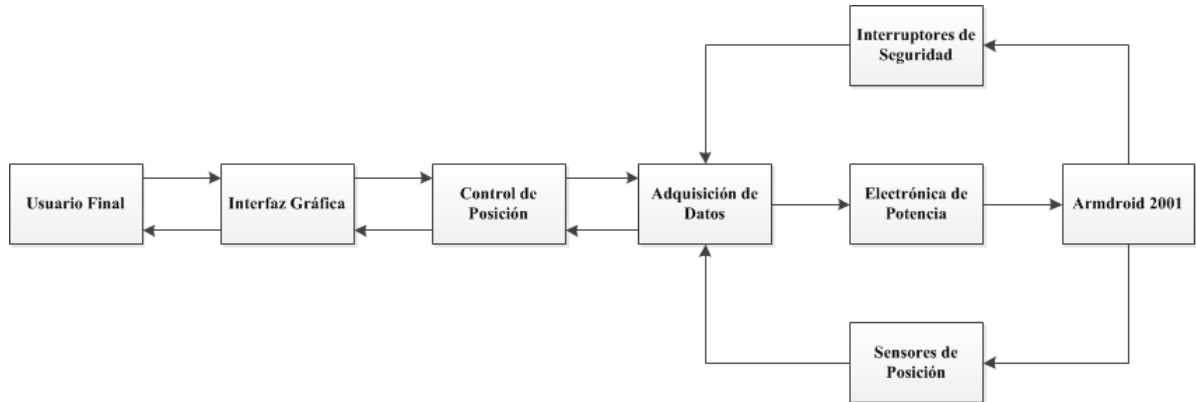


Figura 3.1: Esquema del módulo de control.

3.1 Fase inicial de implementación

En la fase inicial del proyecto, se examinó el funcionamiento del brazo robótico ARMDROID 2001, esto con el fin de detectar posibles fallas y determinar las causas de las mismas. Del examen realizado se pudo comprobar que dos de los motores DC pertenecientes al brazo no funcionaban correctamente, esto debido a que los engranes que conectan el eje del motor con el sistema de engranajes del brazo robótico no estaban acoplados adecuadamente al eje del motor, esto provocaba que el engrane se deslizara sobre el eje del motor cuando este se alimentaba, provocando que el movimiento deseado no se llevara a cabo.



Figura 3.2: Motores DC utilizados por el ARMDROID 2001.

Los motores DC que utiliza este brazo robótico son motores Pittman de 24 V sin escobillas, con imán permanente, en la figura 3.2 se muestra un motor similar a los utilizados por el ARMDROID 2001. Se logró observar que estos motores DC cuentan con un codificador HEDS 9100 con el cual se logra detectar la posición del robot, dicho codificador se ubica en la parte posterior

del motor, se encontró que uno de estos codificadores no funcionaba en uno de estos motores.

También se observó que la pieza mostrada en la figura 3.3 no funcionaba adecuadamente, esta pieza es la responsable de marcar el final de carrera del hombro en el brazo robótico.



Figura 3.3: Pieza usada para el final de carrera del hombro.

Una vez detectadas estas fallas se les atendió de inmediato, primero, se procedió a reemplazar los engranes de los motores, adicionalmente se reemplazó uno de estos motores por uno con una “zona muerta” menos pronunciada, finalmente se reemplazó el codificador dañado por uno completamente funcional. Es importante señalar que no se logró encontrar un reemplazo para la pieza mostrada en la figura 3.3, no obstante, la misma se puede construir utilizando una impresora tridimensional.

El brazo robótico ARMDROID 2001 es un brazo esférico, es decir, la combinación de todos los movimientos que el mismo realiza forman una esfera. En la figura 3.4 se detallan las partes que éste robot es capaz de mover, haciendo uso de los seis motores DC que posee.

Haciendo uso de esta figura, se definieron los movimientos del motor de la siguiente forma:

1. Base.
2. Hombro.
3. Brazo.
4. Mano.
5. Pinza.

Es importante mencionar que la denominada mano del brazo robótico es capaz de hacer dos movimientos, giro y flexión, de esta forma se obtienen los seis movimientos con los cuales el robot es capaz de formar una esfera.

En la parte final de esta fase se elige diseñar un sistema de control de posición, utilizando los codificadores como sensores de posición, un Arduino

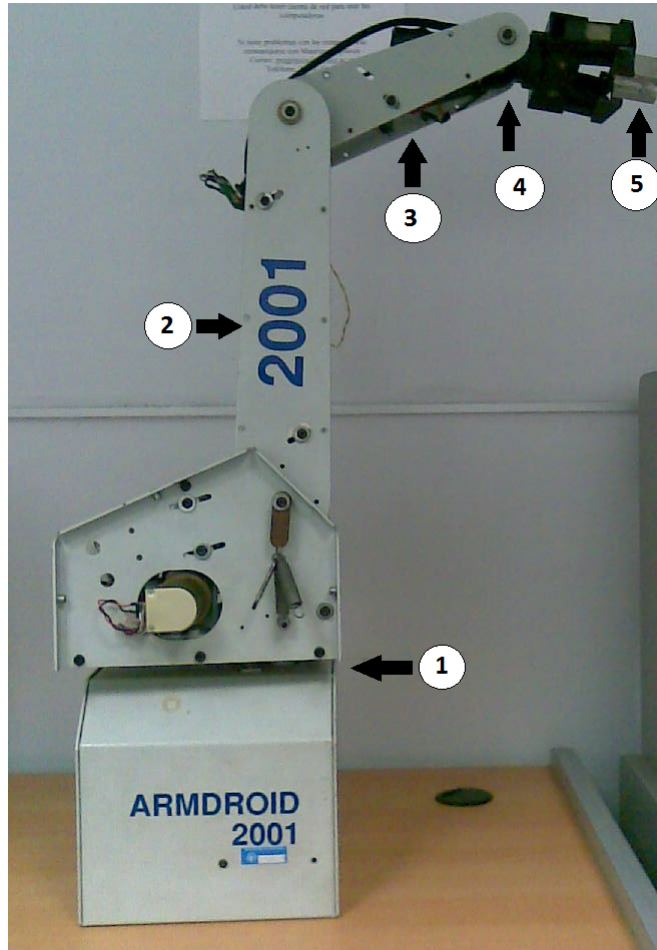


Figura 3.4: Partes movibles del ARMDROID 2001.

Uno como sistema de adquisición de datos y como la salida del controlador, para la interfaz gráfica se elige utilizar el software LabVIEW[®] debido a que el mismo facilita tanto la comunicación con el Arduino como la creación de la interfaz gráfica.

3.2 Implementación de hardware

La unidad de hardware que se pretende implementar debe cumplir las siguientes funciones:

- Proveer al brazo robótico de la potencia necesaria para que este logre moverse sin complicaciones, esto en función de lo que el lazo de control

de posición requiera.

- Traducir las señales provenientes de los distintos codificadores e interruptores del robot en señales que puedan leerse e interpretarse.

Es por este motivo que esta misma unidad de hardware se subdivide en dos etapas, etapa de entrada y etapa de salida, a continuación se detalla el funcionamiento de ambas etapas.

Etapa de entrada

En la etapa de entrada se encuentra toda la electrónica necesaria para transmitir las señales de posición y seguridad que envía el brazo robótico al computador, en la figura 3.5 se muestra el diagrama esquemático de dicho circuito.

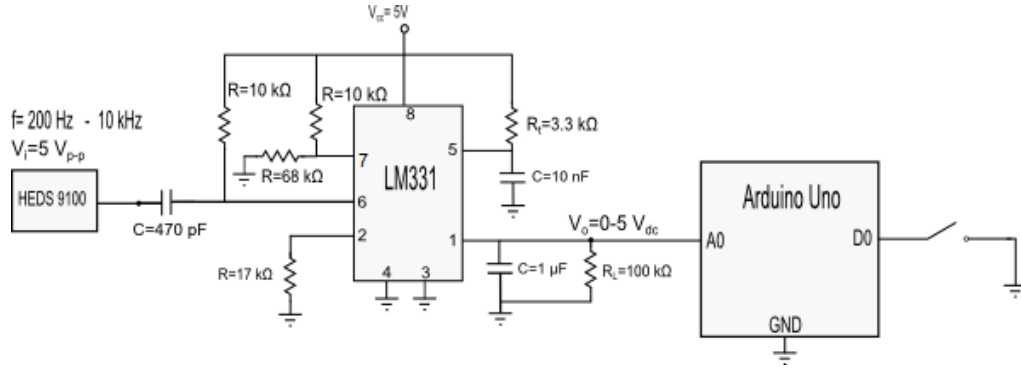


Figura 3.5: Etapa de entrada para un único motor.

Se utiliza un LM331 para convertir la señal de frecuencia variable proveniente del codificador HEDS-9100 en una señal de tensión DC variable, esto se hace debido a que el rango de frecuencias que genera el codificador para estos motores (de 200 a 10 kHz aproximadamente) supera la frecuencia de muestreo del Arduino. Se realiza la lectura de la señal de tensión V_o utilizando los puertos analógicos del Arduino. Es importante señalar que este circuito se debe utilizar para cada motor que posee el brazo robótico. La señal proveniente de los distintos interruptores se lee utilizando los puertos digitales del Arduino, tal y como se aprecia en la figura 3.5. Es importante señalar que la señal que lee el Arduino es una señal proporcional a la velocidad de giro del motor, por lo que es necesario obtener la posición a partir de ésta señal.

Etapa de salida

En la etapa de salida, se debe tomar la señal pwm del Arduino y utilizarla para alimentar los distintos motores del robot.

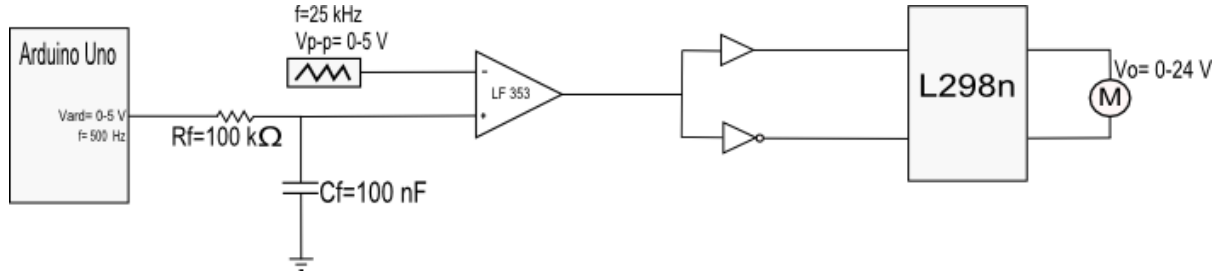


Figura 3.6: Etapa de salida para un único motor.

La figura 3.6 muestra la unidad de hardware implementada, la señal de tensión V_{ard} es una señal cuadrada de 5 V con ciclo de trabajo variable y una frecuencia aproximada de 500 Hz, el capacitor C_f se carga a un valor proporcional al ciclo de trabajo $D \cdot V_{ard}$, dicha señal entra a un comparador, junto con una señal triangular de 25 kHz, con amplitud de 5 V_{p-p} , esto se hace con el fin de obtener una señal de alta frecuencia, dado que la frecuencia original de la señal del Arduino es muy baja, generando problemas con el consumo de corriente debido a la impedancia del motor, también se producen vibraciones y sonido en los motores, estos efectos no deseados se eliminan incrementando la frecuencia de la señal de alimentación de los motores, reduciendo así los efectos de la impedancia del motor y eliminando por completo las vibraciones y el sonido. A la salida del comparador se tendrá una señal cuadrada, con un ancho de pulso igual al de la señal producida por el Arduino, y una frecuencia de 25 kHz, esta señal se hace pasar por un *buffer* y por un inversor.

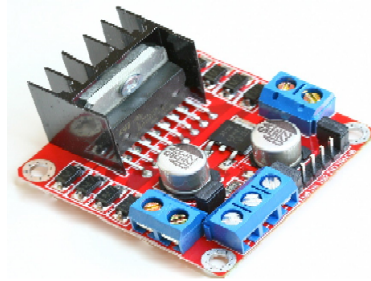


Figura 3.7: Módulo integrado del L298n.

El componente L298n es un circuito integrado contiene dos puentes H, por su configuración interna este componente requiere dos señales para controlar un único motor, para controlar el ciclo de trabajo de un único motor es preciso que el L298n reciba la señal de control, y la misma señal negada, de esta forma se logra el control bidireccional de un motor DC. En la figura 3.7 se muestra

uno de los tres módulos integrados que se requieren para implementar la etapa de salida de la unidad de hardware.

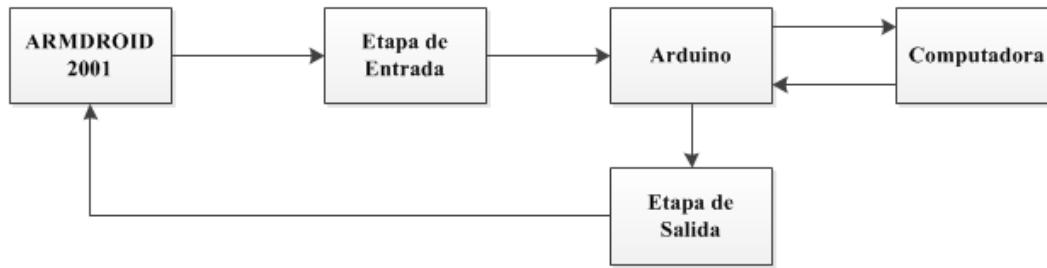


Figura 3.8: Esquema de la unidad de hardware para un único motor.

Al unir la etapa de salida y la etapa de entrada, se obtiene el esquema de funcionamiento mostrado en la figura 3.8. En síntesis, el módulo de control resultante se obtendrá al repetir el esquema mostrado en la figura 3.8 para cada uno de los motores, es importante señalar que el robot cuenta con cuatro interruptores, es decir, el brazo robótico sólo cuenta con cuatro posiciones de operación no permitidas correspondientes a los motores que las controlan.

3.3 Implementación de software

A partir de las señales que recibe el Arduino, la unidad de software debe llevar a cabo las siguientes funciones:

- Implementar el sistema de control de posición.
- Monitorear el estado del brazo robótico en todo momento.
- Tomar las medidas de seguridad requeridas para el correcto funcionamiento del robot.

Para la implementación de la unidad de software se utilizó el programa LabVIEW[®] 2010, el cual es un entorno de desarrollo creado por National Instruments para diseñar sistemas, con un lenguaje de programación visual gráfico. Este programa permite la creación de una interfaz gráfica para el usuario, así como la síntesis del sistema de control y la comunicación con el Arduino, ya sea para recibir las señales del brazo robótico o para enviar salidas de tensión con ancho de pulso variable.

En la figura 3.9 se muestra el diagrama de bloques de la unidad de software, la primer tarea del programa principal es obtener una señal de posición a partir de la señal de velocidad que se obtiene de la unidad de hardware. Una vez obtenida la señal de posición, se procede a implementar el lazo de control, para lo cual se utiliza un controlador PID con dos grados de libertad. Posteriormente, la unidad de software debe implementar la interfaz gráfica que permita el monitoreo y control del brazo robótico.

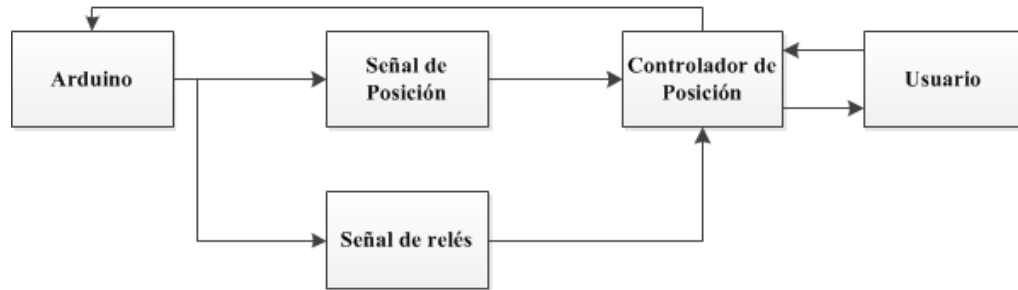


Figura 3.9: Diagrama de bloques de la unidad de software.

Controlador de posición

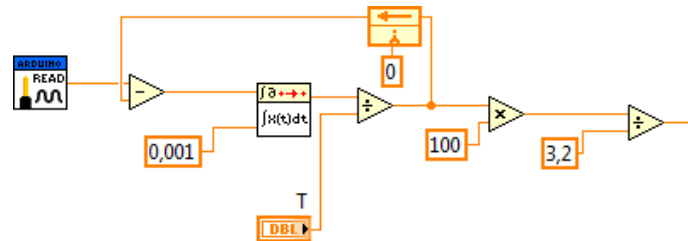


Figura 3.10: Filtrado y normalización de la señal obtenida.

El diagrama de bloques mostrado en la figura 3.10 muestra (en forma representativa) la manera en que se filtra y normaliza la señal de tensión obtenida, no obstante, hay que tomar en cuenta que esta señal siempre tendrá una pendiente positiva. Para solucionar este inconveniente hay que tener en cuenta que en primera instancia, un ciclo de trabajo del 50 % marca el punto en el cual un motor se encuentra en reposo, un ciclo de trabajo mayor implica que el motor gira en una dirección, un ciclo de trabajo menor al 50 % indica que el motor girará en sentido contrario.

A partir de este hecho, se implementan dos *flip-flops* D como los mostrados en la figura 3.12, los cuales recibirán como flanco de reloj un cruce por 50 % en el ciclo de trabajo (salida del controlador). En la figura 3.11 se muestra el diagrama de bloques a partir del cual se obtiene la señal de posición, las señales de control que reciben los *flip flops* corresponden a los cruces de la salida del controlador por 50 % además, en función de estas señales de control se elige entre la señal con pendiente positiva o la señal con pendiente negativa, los cuadros grises contienen un pequeño *script* el cual limita la señal de posición entre 0 y 100 %. Esta es la señal resultante que ingresa al controlador de dos grados de libertad, finalmente la salida del controlador es normalizada y será

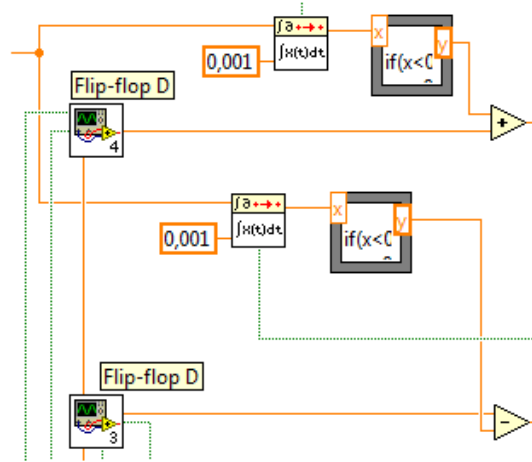


Figura 3.11: Obtención de la señal de posición.

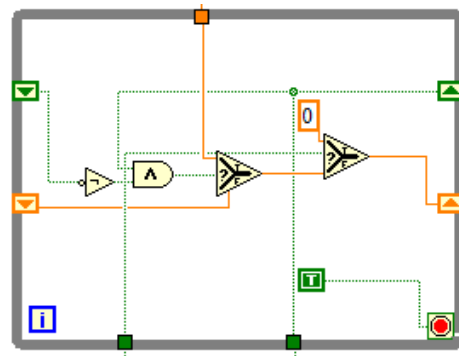


Figura 3.12: Flip flop D implementado.

enviada a la etapa de salida de hardware.

Para la etapa de seguridad, el programa principal monitorea constantemente los interruptores del brazo robótico, los cuales se encuentran normalmente abiertos, una vez que estos se activen, el programa principal detectará un cero, deteniendo por completo el funcionamiento normal del brazo, en esta condición, el programa no permitirá que se sobrepase la posición límite marcada por estos interruptores. Cabe señalar que el lazo de control manual realiza la manipulación sobre la salida del controlador PID directamente, es decir, el lazo de control manual manipula el ciclo de trabajo con el que los motores son alimentados.

Por último en la figura 3.13 se muestra la interfaz gráfica que se obtiene en LabVIEW®, el funcionamiento de la misma se detalla en el apéndice A.

En resumen, el modelo del módulo de control para el brazo robótico se

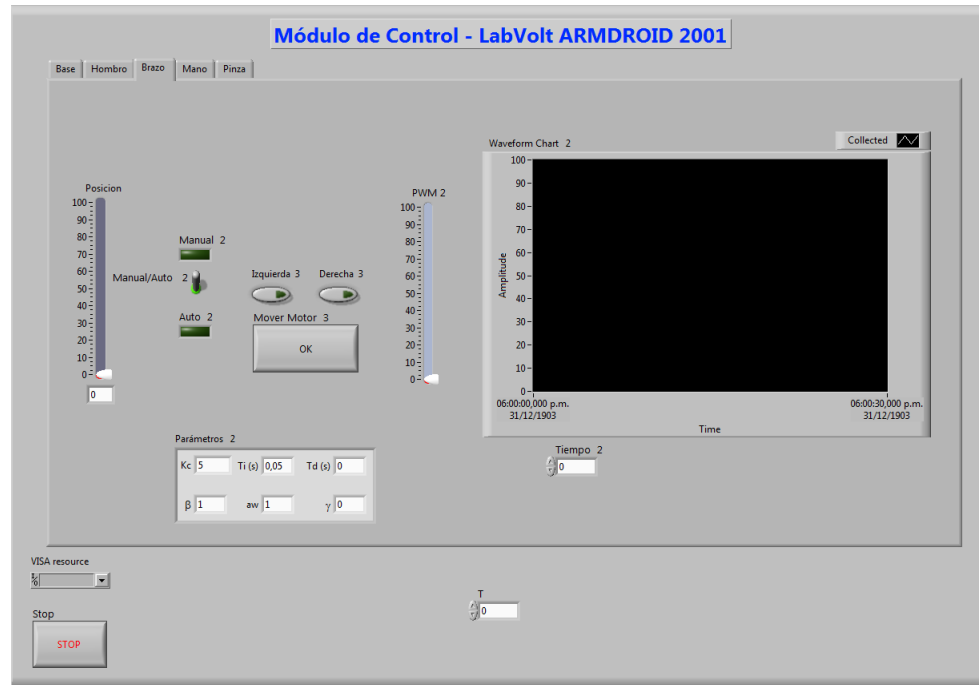


Figura 3.13: Interfaz gráfica vista del módulo de control.

puede simplificar de la forma en que se observa en la figura 3.14.

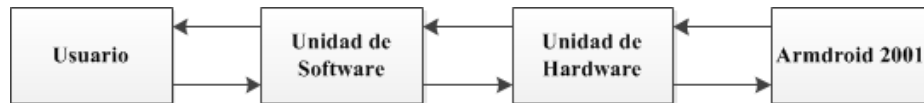


Figura 3.14: Diagrama de bloques del módulo de control simplificado.

4 Validación del módulo de control

En este capítulo se muestran los resultados que se obtuvieron al utilizar el módulo de control implementado para manejar el brazo robótico ARMDROID 2001.

En primera instancia se realizaron dos pruebas de lazo abierto con el fin de obtener un modelo del proceso, los resultados mostrados en las figuras 4.1 y 4.2 sugieren que el modelo es una planta integrante indistintamente de la parte del brazo que se requiere mover, se debe mencionar que para la prueba de lazo abierto realizada sobre el brazo se aplicó un 100 % a la salida del controlador.

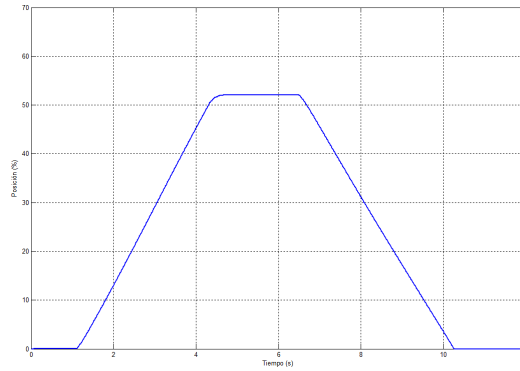


Figura 4.1: Respuesta de lazo abierto del brazo.

Es importante observar que las dos pendientes difieren, esto supone que la gravedad tiene cierta influencia sobre la respuesta del sistema, no obstante la diferencia en las respuestas se puede considerar despreciable. A partir de estas curvas de lazo abierto se obtuvieron modelos tanto para el hombro como para el brazo los cuales se muestran en las ecuaciones 4.1 y 4.2, los cuales corresponden a plantas integrantes sin tiempo muerto.

$$P_b(s) = \frac{12,36}{s} \quad (4.1)$$

$$P_h(s) = \frac{29,43}{s} \quad (4.2)$$

La figura 4.3 muestra la respuesta en lazo cerrado del motor que mueve el hombro del robot, se puede apreciar como la respuesta tiende a ser de

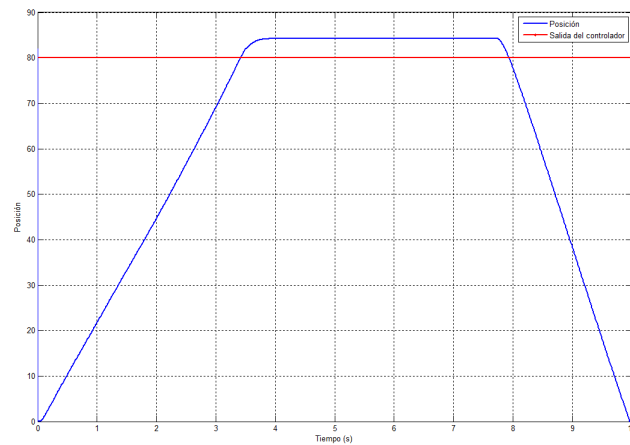


Figura 4.2: Respuesta de lazo abierto del hombro.

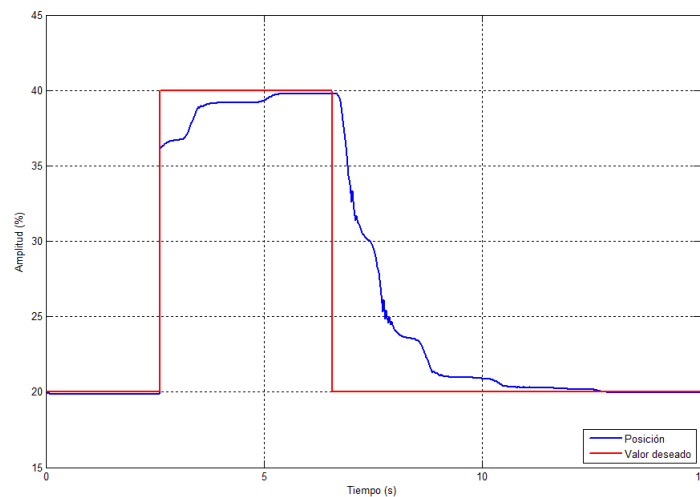


Figura 4.3: Respuesta en lazo cerrado del sistema de control de posición para el hombro.

primer orden, no obstante se logra apreciar que el sistema es muy poco lineal, debido en gran parte a que la fuerza que el motor debe ejercer para bajar el hombro del brazo robótico, no será la misma que debe ejercer para levantar el hombro. En esta figura, el escalón de amplitud positiva representa el descenso del hombro, se aprecia que esta respuesta es mas rápida a la obtenida para el escalón negativo. Dado que la dinámica de todos los motores es similar a la mostrada en la figura 4.3 se puede decir que los demás motores tendrán respuestas semejantes, claro esta, tomando en cuenta que en un sentido se

logrará un movimiento más rápido que en su sentido opuesto, esto en función de la carga que deben mover los motores.

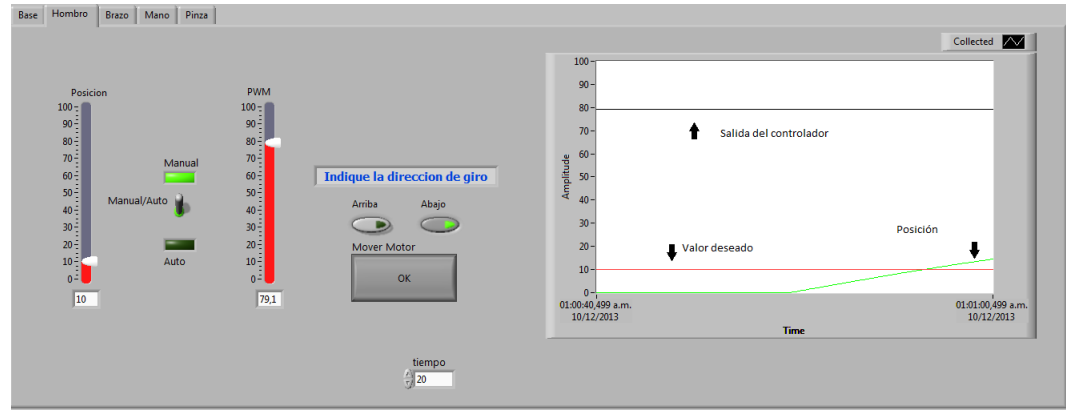


Figura 4.4: Respuesta del control manual.

En la figura 4.4 se muestra el comportamiento a lazo abierto del lazo de control de posición en conjunto con el funcionamiento de la interfaz. Para facilitar la lectura de la gráfica se cambia el color de fondo, no obstante en la interfaz real la salida del controlador será una línea blanca, el valor deseado se mantendrá en rojo y la señal de posición se mantendrá en verde, para mover el hombro hacia abajo es preciso seleccionar la dirección y posteriormente pulsar el botón “*Mover Motor*”, de lo contrario no se podrá realizar el movimiento deseado.

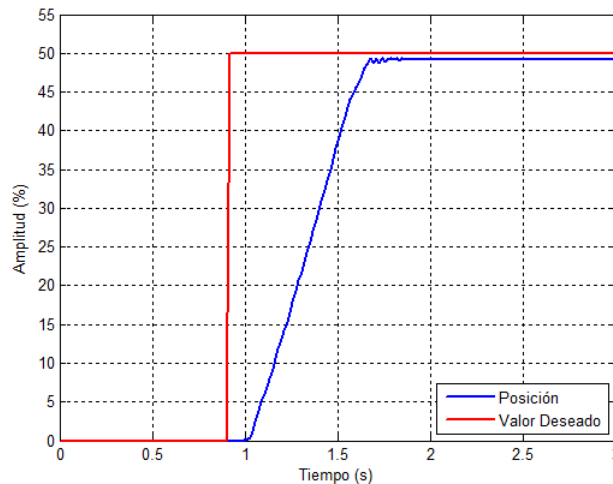


Figura 4.5: Respuesta de lazo cerrado con un controlador proporcional.

Posteriormente se sintonizó un controlador proporcional el cual garantiza un error permanente menor al 1% con $\beta = 1$, la función de transferencia de este controlador se muestra en la ecuación 4.3. La figura 4.5 muestra la respuesta obtenida.

$$C(s) = 3,36 \quad (4.3)$$

Se puede apreciar como la velocidad de respuesta se mejora considerablemente en cuanto a su velocidad y la reducción en las oscilaciones, no obstante se recomienda utilizar un controlador proporcional-integral que llegue a garantizar un error permanente nulo.

Por último se comprobó el funcionamiento del sistema de seguridad implementado, por lo que si se pretende llevar al brazo a una posición no permitida, los interruptores de final de carrera existentes se activarán, desplegando el mensaje mostrado en la figura 4.6, para realizar esta prueba se activó el interruptor de final de carrera en el hombro en forma manual.

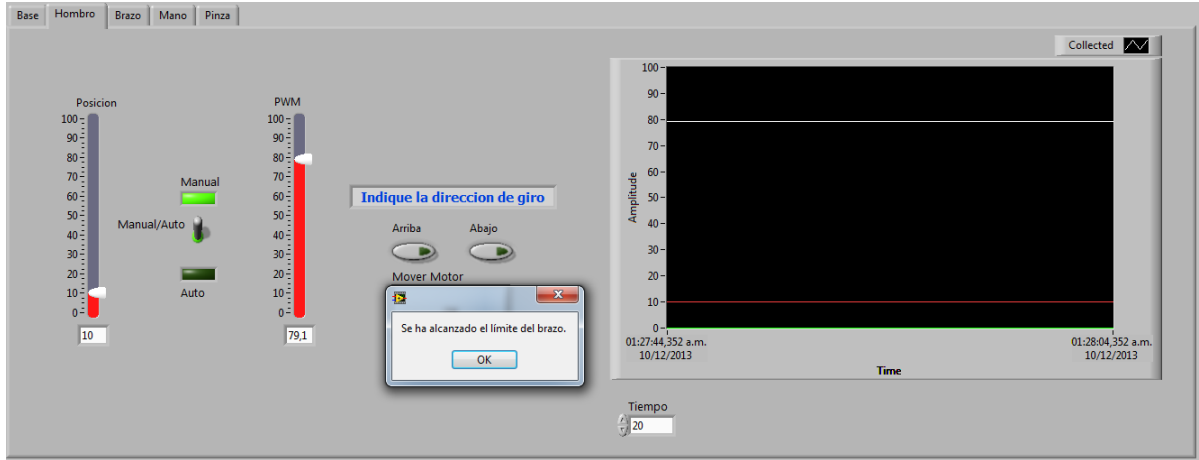


Figura 4.6: Mensaje de error desplegado por el programa.

5 Conclusiones y recomendaciones

5.1 Conclusiones

- Se lograron reparar y reemplazar las partes del brazo robótico que así lo requerían, esto con la finalidad de que el brazo pudiese operar adecuadamente, no obstante, la única pieza que no tiene reemplazo es la que se muestra en la figura 3.3.
- Se logró implementar la electrónica necesaria para poder realizar un control de posición para los distintos motores del brazo robótico.
- La frecuencia de muestreo del Arduino Uno es tan baja que se requiere implementar una etapa de electrónica que convierta las señales provenientes de los sensores de posición en señales leíbles por el Arduino.
- Se integraron las señales de seguridad provenientes de los interruptores en forma adecuada.
- Se obtuvo una interfaz gráfica con la cual se puede monitorear el estado del ARMDROID 2001 en todo momento así como manipular el mismo ya sea en modo automático o en modo manual.
- Se logró cumplir con los objetivos establecidos al inicio del proyecto.

5.2 Recomendaciones

- Para evitar el uso de una etapa de electrónica de entrada se puede hacer uso de una plataforma de adquisición de datos con una frecuencia de muestreo mayor.
- La señal pwm que alimente los motores debe tener una frecuencia igual o superior a los 20 kHz, esto garantizará que los efectos de la impedancia de los motores se minimice, con esta frecuencia de operación se reducirá el consumo de corriente en los motores cuando se encuentran en estado estacionario.

Bibliografía

CENDITEL (2013). *Plataforma de desarrollo en hardware libre*. Fundación Centro de Desarrollo e Investigación en Tecnologías Libres. <http://hl.cenditel.gob.ve/>.

Husain, I. (2003). *Electric and Hybrid Vehicles*. CRC Press.

Khatib, O. y Siciliano, B. (2008). *Springer Handbook of Robotics*. Springer, Berlin.

Universidad Politécnica de Madrid (2013). *Electrónica de excitación empleada en la actuación de motores de corriente continua de baja potencia*. http://www-app.etsit.upm.es/departamentos/teat/asignaturas/lab-ingel/modulacion_pwm_pmdc_v4.pdf.

A Manual de usuario

En este apéndice se detallan los aspectos a considerar para la puesta en marcha y operación del brazo robótico, para la puesta en marcha se discuten las distintas conexiones que deben realizarse, mientras que para la operación se detalla el modo de uso del programa, así como los distintos elementos que componen a la interfaz gráfica.

A.1 Puesta en marcha

En primera instancia, para poner en marcha el brazo robótico, ejecutar el programa principal, llamado módulo de control. Posteriormente se deben alimentar las unidades de hardware con 5 V y -5 V, en el caso del comparador, y con 5 V y tierra en el caso de los inversores y el LM331. Es importante señalar que se deben interconectar las referencias del Arduino, y la unidad de hardware, con el fin de obtener valores lógicos coherentes. Los comparadores deben alimentarse con la señal pwm proveniente del Arduino y con una señal triangular de 25 kHz con 5 V_{p-p} , es importante que esta señal no tenga un offset en la tensión, ya que esto puede alterar el resultado de la comparación. En la figura A.1 se muestra un esquema básico de la conexión del Arduino con las unidades de hardware.

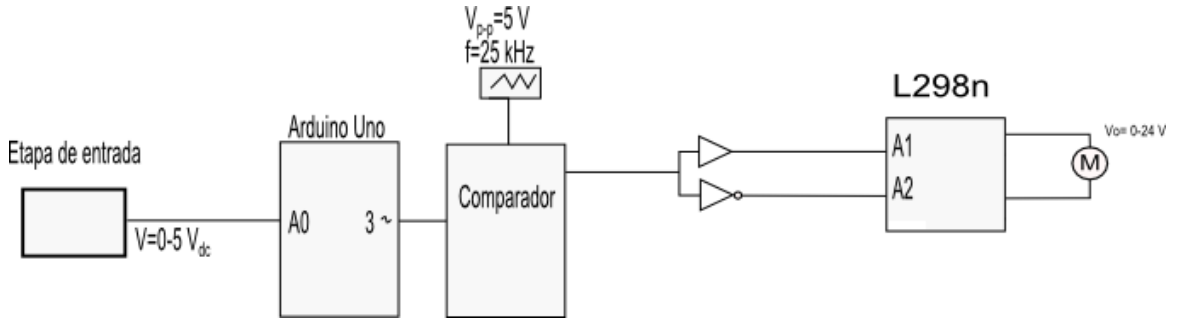


Figura A.1: Conexión del Arduino con las etapas de hardware para un sólo motor.

Nótese que la mano requiere dos entradas analógicas y dos salidas pwm, debido a los movimientos de flexión y rotación. En la tabla A.1 se muestra la relación de conexiones que se debe realizar para poner en marcha el brazo robótico.

Cuadro A.1: Relación de conexiones.

Movimiento	Entrada analógica	Salida PWM	Entrada digital
Brazo	A0	6	D8
Hombro	A1	9	D12
Pinza	A2	10	-
Mano (flexión)	A3	5	D7
Mano (giro)	A4	3	-
Base	A5	11	D4

A.2 Operación del brazo robótico ARMDROID 2001

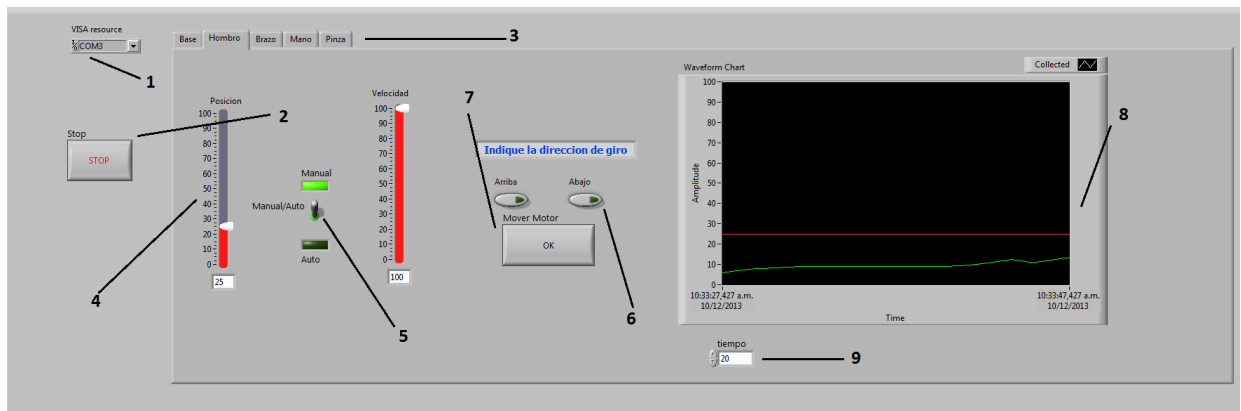


Figura A.2: Interfaz gráfica del módulo de control.

En esta figura se señalan todos los elementos necesarios para poder operar el programa en forma adecuada.

1. Selector de puerto: con este selector se elige el puerto en el cual está ubicado el Arduino.
2. Detener ejecución: detiene la ejecución del programa.
3. Pestañas de monitoreo: en estas pestañas se elige la parte del robot que se desea monitorear y controlar.
4. Selector de valor deseado: esta barra especifica la posición deseada a la que se quiere llevar cierta parte del brazo robótico
5. Selector automático o manual: permite cambiar el modo de operación de modo automático a modo manual o viceversa.

6. Selectores de dirección: estos botones solo se habilitan en modo manual, permiten elegir la dirección en que se desea mover la parte del robot que se está controlando.
7. Mover motor: al mantener pulsado este botón se habilita el movimiento del brazo, al soltarlo el robot se detendrá.
8. Gráfico de monitoreo: este gráfico monitorea el estado del brazo robótico, la línea roja indica la posición deseada (valor deseado), la línea blanca muestra la salida del controlador y finalmente la línea verde representa la posición de la parte del robot que se monitorea.
9. Tiempo de despliegue del gráfico: esta constante especifica el intervalo tiempo de muestreo en segundos que se desea mostrar en el gráfico de monitoreo.

B Manual de programador

En este apéndice se mencionan los bloques de mayor importancia para el funcionamiento del lazo de control de posición así como la estructura del programa en si.

B.1 Operación del brazo robótico ARMDROID 2001

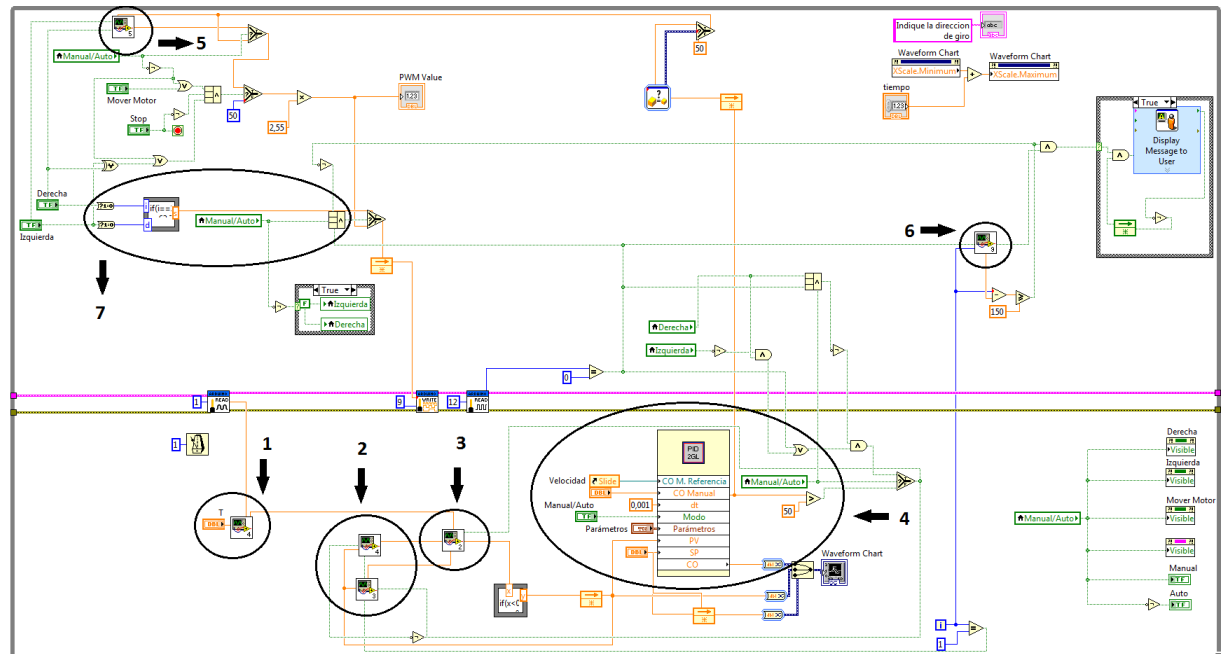


Figura B.1: Programa requerido para controlar un único motor.

En la figura B.1 se muestra el código requerido para controlar un único motor, esto con la finalidad de facilitar la lectura y comprensión del programa, para minimizar el tamaño del mismo, se utilizan diversos subbloques personalizados, los bloques de interés se señalan en la figura.

1. En este bloque se filtra y normaliza la señal de tensión que lee el Arduino, la constante T indica el período de muestreo para la integral interna (ver

figura 3.9).

2. Estos bloques son dos flip-flops D requeridos para la obtención de la señal de posición tal y como se mencionó en el capítulo 3.
3. La estructura de este bloque se observa en la figura B.2.

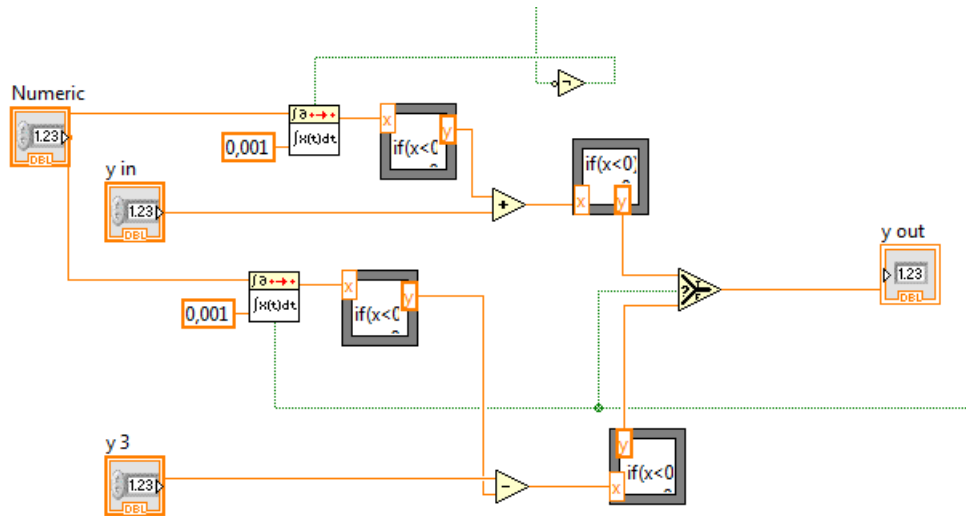


Figura B.2: Obtención de la señal de posición.

A partir de las señales de control que recibe este bloque se obtendrá una curva con pendiente positiva o negativa, según sea el caso, en este bloque se obtiene la señal de posición que recibe el controlador.

4. Este bloque es el controlador PID con dos grados de libertad, a la salida del mismo se generan distintas señales de control necesarias para el funcionamiento del bloque utilizado para obtener la señal de posición que recibe el controlador, por este motivo se recomienda manipular estas señales con cuidado.
5. La función de este bloque es elegir entre el control automático y el control manual, además, se encarga de tomar la señal porcentual proveniente del controlador y transformarla en una señal que pueda ser escrita en las salidas PWM del Arduino, la cual es un número entero entre 0 y 255. En la figura B.3 se muestra la estructura interna de este bloque.
6. Este bloque es un flip-flop D utilizado para generar el mensaje de error después de una determinada cantidad de iteraciones en caso de que el interruptor monitoreado se active. Se eligen 150 iteraciones (aproximadamente 2 segundos), no obstante la acción correctiva en la operación

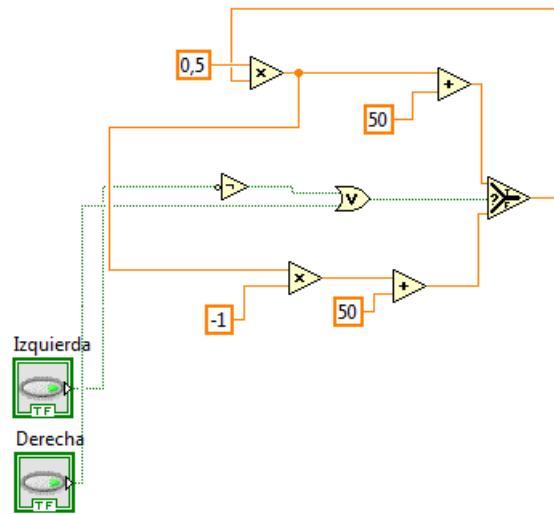


Figura B.3: Bloque de control automático/manual.

del robot se realizará tan pronto como se detecte que el interruptor se ha activado.

7. En esta sección del código se toma la acción correctiva una vez que se ha comprobado que el interruptor monitoreado se ha activado, en la figura C.2 se muestra la estructura interna del código ubicado en el cuadro gris.

```

i if(i==1&d==0)
  s=63,75;
else
  if(i==0&d==1)
    s=191,25;
  else
    s=127,5;
d

```

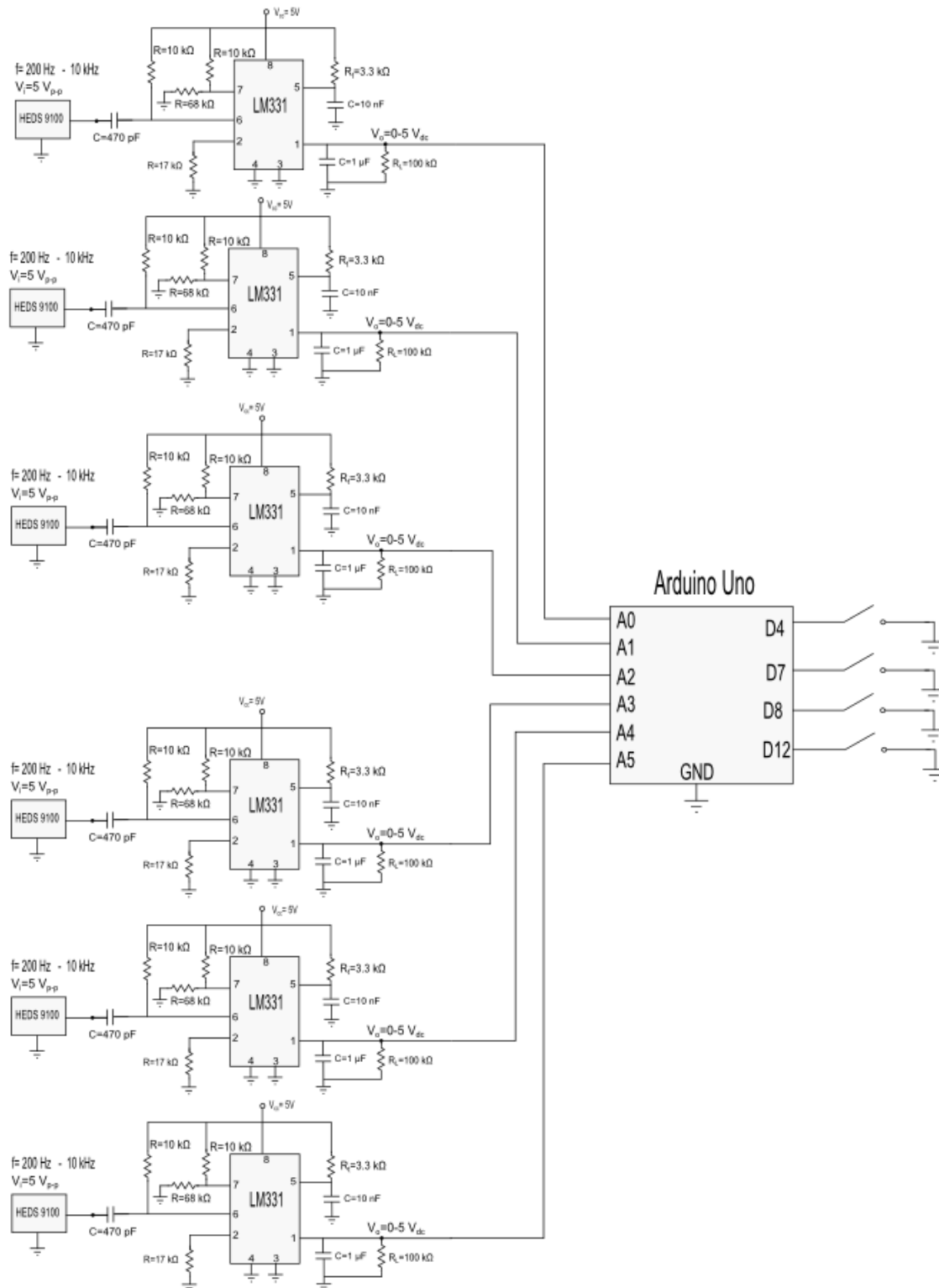
Figura B.4: Código utilizado para tomar la acción correctiva.

C Diagramas Esquemáticos

Tomando en consideración que toda la electrónica desarrollada para las etapas de entrada y salida se puede implementar en una placa de silicio, se presentan los diagramas esquemáticos completos de las etapas de entrada y salida. En el cuadro C.1 se muestran los componentes necesarios para implementar estos circuitos.

Cuadro C.1: Lista de componentes.

Cantidad	Componente	Valor
1	Arduino Uno	-
1	Adaptador USB A-B	-
6	LF353	-
3	Puentes H L298n	-
6	Compuertas NOT	-
6	Buffers	-
6	LM 331	-
6	Capacitores	470 pF
6	Capacitores	10 nF
6	Capacitores	100 nF
6	Capacitores	1 μF
12	Resistencias	10 $k\Omega$
6	Resistencias	68 $k\Omega$
6	Resistencias	17 $k\Omega$
6	Resistencias	3,3 $k\Omega$
12	Resistencias	100 $k\Omega$



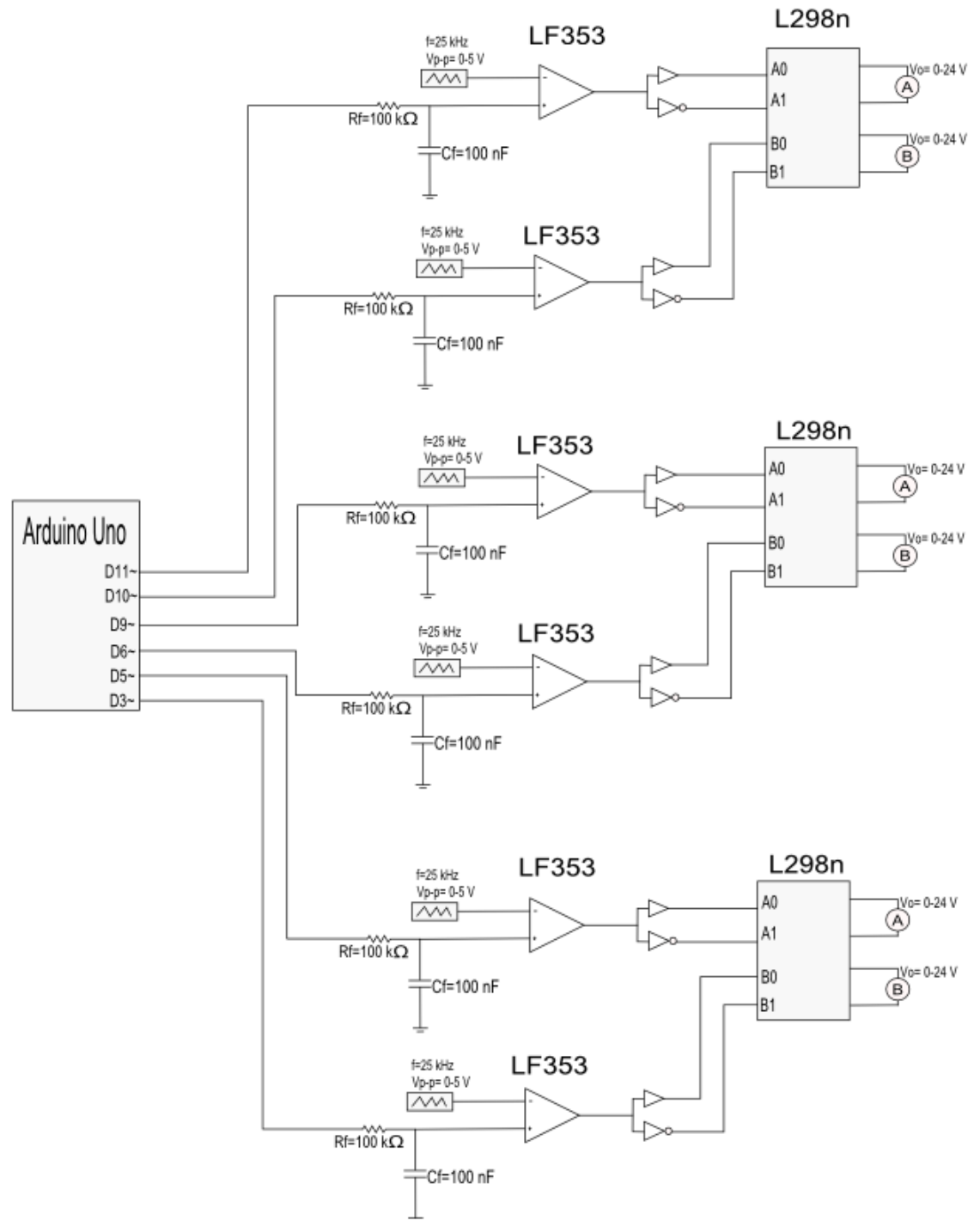


Figura C.2: Diagrama esquemático completo de la etapa de salida.