

ROBOT N6 Max

Diagramas de bloques y de flujo



Integrantes:

Astrada, Juan Ignacio

Lui, Santino

Martinez, Julieta

Palkovic, Micaela

Remedi, Valentina

Ríos, Franco

Grupo N°3

JTP: Eugenio Borzone

Introducción:

En un mundo cada vez más avasallado por tecnologías complejas, nace la necesidad de permitir a los jóvenes el poder interactuar con estas piezas de electrónica tan básicas en el campo, pero a la vez, tan intimidantes para cualquiera que carezca de una buena perspectiva previa del asunto. Fue entonces que nació el proyecto arduino, uno que buscaría permitir a numerosos internautas adentrarse en el mundo de la electrónica, y con ello, comenzaron las iniciativas para la introducción a temas más emocionantes todavía, como lo es la que nos trae aquí: la robótica. Ahora bien, ¿qué nos atañe específicamente hoy? Se trata del modelado de un gemelo digital de un robot proveniente de RobotGroup, conocido como el modelo Múltiplo N6-máx. El contenido del siguiente informe consistirá en expresar el procedimiento llevado a cabo para el modelado, partiendo desde el diseño en la aplicación simul8de, para expresar el hardware usado, e IDE Arduino, para diseñar el código que, tras investigar, intuimos que tiene el robot en realidad.

Software

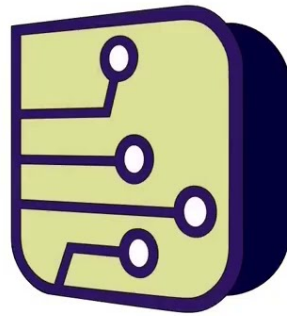
En tanto a lo que al software respecta, hemos utilizado una serie de herramientas que nos permitieron organizar el trabajo en un repositorio, presentar la idea detrás del robot mediante diagramas, bien sea de flujo o de bloques, y programar el código computacional que llevaría adelante el comportamiento del robot.

Comencemos hablando de uno de los grandes:



GitHub es un servicio basado en la nube que aloja un sistema de control de versiones (VCS) llamado Git. Éste permite a los desarrolladores colaborar y realizar cambios en proyectos compartidos, a la vez que mantienen un seguimiento detallado de su progreso. Fue gracias a su increíble versatilidad y utilidad que pudimos mantener una organización amena.

Luego tenemos nuestra principal herramienta a la hora de plantear las relaciones entre los componentes en un entorno físico:



SimulIDE

SimulIDE es un simple simulador de circuitos electrónicos en tiempo real, destinado a aquellos que tienen el hobby de la electrónica o estudiantes, para que así puedan experimentar y aprender mediante circuitos y microcontroladores simples. Este simulador no está destinado a ser utilizado para análisis de circuitos, dado a que apunta a ser rápido, simple y fácil de usar, con lo cual se explicita que tendrá modelos no tan precisos y una cantidad de herramientas limitada.

Lo siguiente, ya que el concepto de Arduino se tratará en la categoría de Hardware presente más adelante, simplemente dará por hecho el conocimiento de los conceptos que luego veremos.



El entorno de desarrollo integrado (IDE) de Arduino es una aplicación multiplataforma (para Windows, macOS, Linux) que está escrita en el lenguaje de programación Java. Se utiliza para escribir y cargar programas en placas compatibles con Arduino, pero también, con la ayuda de núcleos de terceros, se puede usar con placas de desarrollo de otros proveedores. El IDE de Arduino admite los lenguajes C y C++ utilizando reglas especiales de estructuración de códigos. El código escrito por el usuario solo requiere dos funciones básicas, para iniciar el boceto y el ciclo principal del programa, que se compilan y vinculan con un apéndice de programa `main()` en un ciclo. El IDE de Arduino emplea el programa `avrdude` para convertir el código ejecutable en un archivo de texto en codificación hexadecimal que se carga en la placa Arduino mediante un programa.



En tanto a planeamiento de código respecta, los diagramas de flujo se han llevado a cabo en la plataforma la cual cuenta con múltiples herramientas para el desarrollo y diseño de diagramas de flujo, diagramas de bloques, etc. La plataforma permite sincronizarse con Google Drive y otras plataformas permitiendo la exportación de los diagramas fácilmente. Es un software en línea y de escritorio con código abierto. Es un software de diagramas de flujo y diagramas construido para las obligaciones y sensibilidades contemporáneas de los profesionales. La versión sin conexión de Draw.io tiene todas las funciones de la versión en línea, excepto guardar en la nube y compartir en línea. Está desarrollada principalmente en el lenguaje de programación JavaScript y cuenta con la Apache License en su versión 2.0.

Hardware

Se planteó un gemelo digital en la plataforma simulIDE, con los componentes acordes a la versión real en 3 dimensiones.

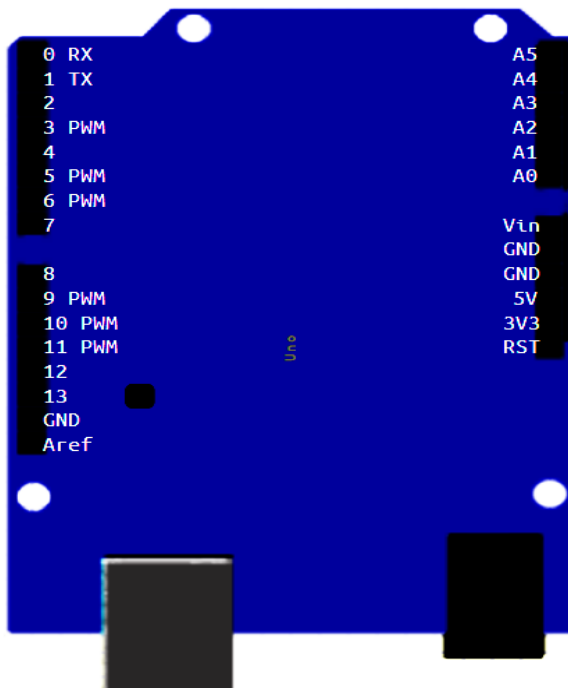
Los componentes utilizados fueron:

- Placa ArduinoUno (Similar a la placa DuinoBot 2.3)
- Sensor de ultrasonido HC-SR04
- Puente H (L298D)
- Dos Motores Dc-14
- Tres baterías AAA
- Componentes varios tales como sensor LDR, un puerto USB para distintos usos, 3 ruedas (2 de ellas conectadas a los motores, la tercera de apoyo), etc.

El robot de aprendizaje Múltiple N6-Max es un sistema ciber-físico cuyo hardware comienza, como casi todo robot, en una pieza fundamental, la placa Arduino. Para este caso, usamos la placa Arduino. El motivo detrás de la utilización de solamente uno se debe a que, entre las diferentes opciones de arduinos, proporcionadas por SimulIDE, todas nos resultaban útiles en igual medida para el proyecto. Habiendo ampliado conocimientos, podemos profundizar más en los diferentes componentes:

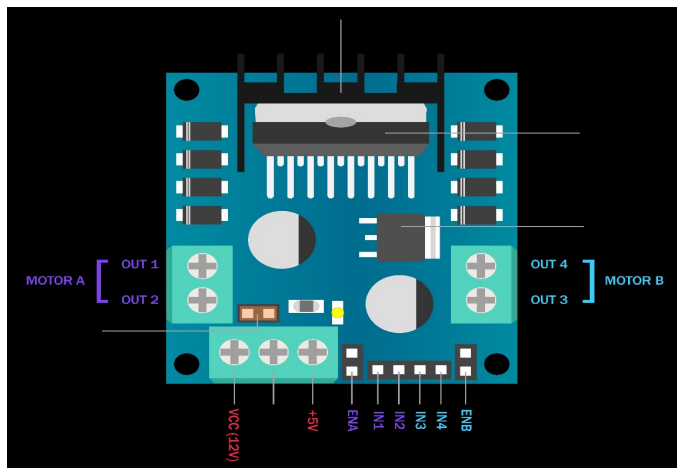
El Arduino:

Uno-1



- Los conectores con denominación A que van desde el cero hasta el cinco se denominan conexiones analógicas, estos pines permiten leer entradas más complejas, tales como la frecuencia o la potencia de una onda.
- Del otro lado de la placa, se encuentran los pines dedicados a tomar valor de variables y se identifican con los números desde el 2 hasta el 13.
- El regulador de voltaje se identifica con la entrada 5V. Lo que significa que proporciona una tensión de 5 voltios constantemente para el funcionamiento de la placa.
- Por último, otra parte importante del arduino son los puertos de entrada y salida. Se utilizan para conectar sensores y demás. Algunos pines tienen funciones específicas como el PWM (Pulse width modulation) o por su traducción al español, Modulación de Ancho de Pulso o comunicación serial. La comunicación serial se refiere a un protocolo de comunicación entre dos o más dispositivos (en este caso, la placa y otro dispositivo cualquiera siempre y cuando sea compatible). Se resume en un proceso de envío de datos de a un bit a la vez, de forma secuencial, mediante un canal de comunicación.

El Puente H:



La placa, si bien es la pieza con más relevancia del robot puesto que permite llevar a cabo la unión entre el software y el hardware entre otras varias funciones, el robot cuenta con un puente H, el cual permite al motor eléctrico Dc girar en ambos sentidos, avanzar y retroceder. El puente se

construye, principalmente y resumidamente, con 4 transistores los cuales son controladores de motor y mediante la información que ingresan a los pines manejan el sentido. También se utiliza para hacer que el motor se detenga abruptamente al hacer un corte en los bornes del motor cuando desconectamos el motor de la fuente que lo alimenta. El relé es un aparato que trabaja en la mayoría de casos con el puente H ya que, mediante el relé la persona es capaz de activar o desactivar la alimentación principal del motor.

Sensor Ultrasonido:



El sensor HC-SR04 es un sensor de distancia de bajo costo, su uso es muy frecuente en la robótica, utiliza transductores de ultrasonido para detectar objetos.

Su funcionamiento consiste en emitir un sonido ultrasónico por uno de sus transductores, y esperar que el sonido rebote de algún objeto presente, el eco es captado por el segundo transductor. La distancia es proporcional al tiempo que demora en llegar el eco. Destaca por su pequeño tamaño, bajo consumo energético, buena precisión y excelente precio. El sensor HC-SR04 es el más utilizado dentro de los sensores de tipo ultrasonido, principalmente por la cantidad de información y proyectos disponibles en la web. De igual forma es el más empleado en proyectos de robótica como robots laberinto o sumo, y en proyectos de automatización como sistemas de medición de nivel o distancia.

Motor Dc-14:



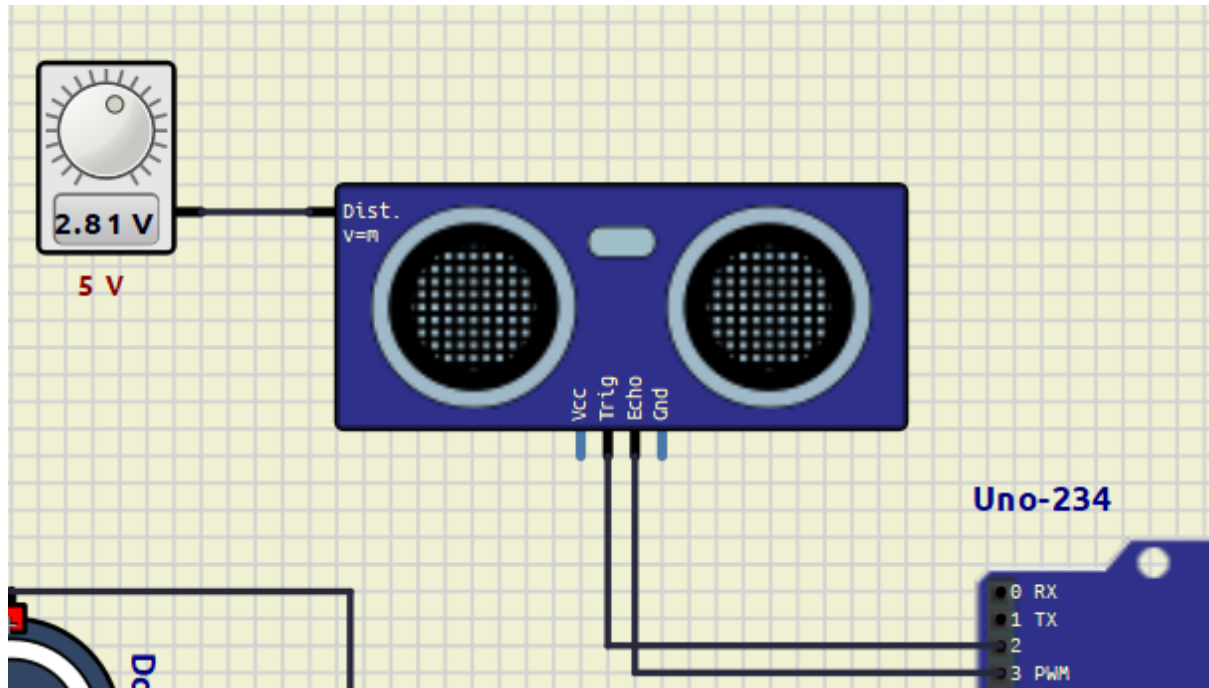
El motor DC, también llamado motor de corriente continua, pertenece a la clase de los electromotores y sirve principalmente para transformar la energía eléctrica en energía mecánica. La mayoría de las formas de construcción del motor DC se basa en fuerzas

magnéticas y dispone de mecanismos internos de tipo electrónico o electromecánico. También característico de los motores de corriente continua convencionales es el acumulador, que cambia periódicamente la dirección del flujo de corriente dentro del motor. Una modificación del motor DC clásico que resulta más potente en numerosos ámbitos de aplicación es el motor DC sin escobillas, que prescinde de contactos de roce y ofrece así, entre otras cosas, una vida útil más larga.

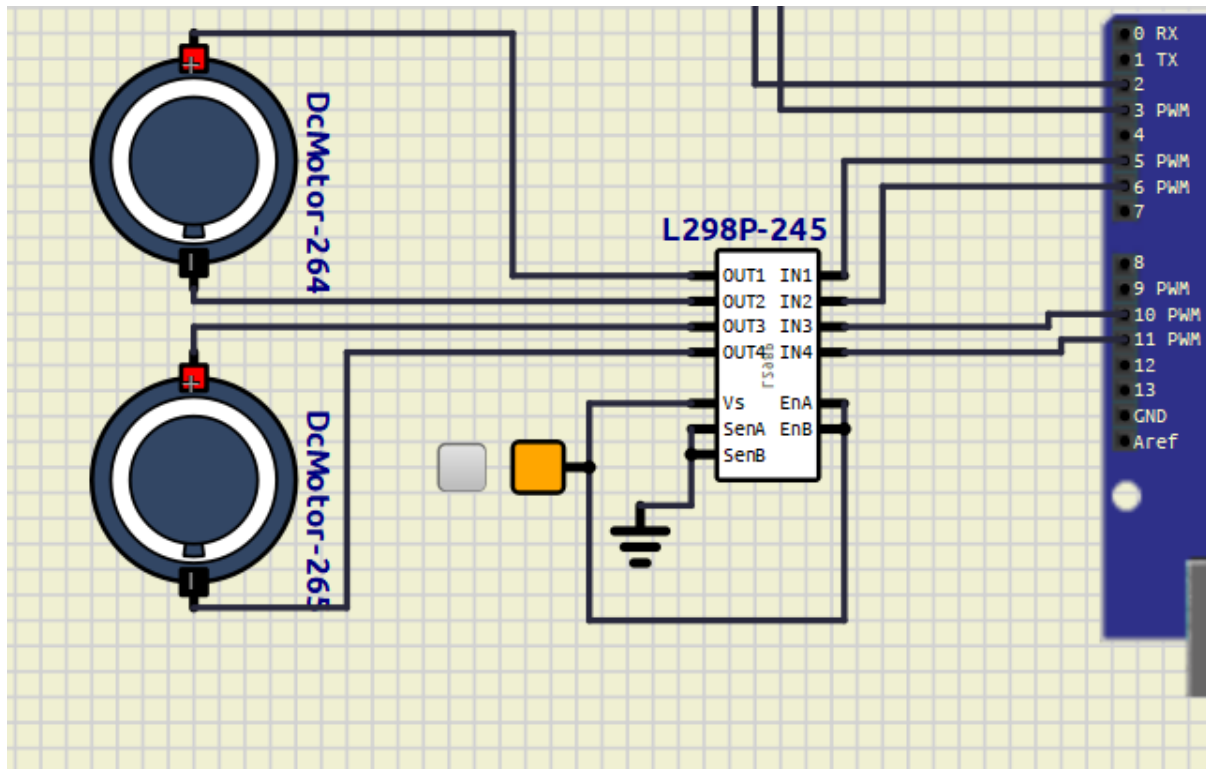
A pesar de la creciente utilización de motores AC y de la gran rivalidad con ellos, que, como motores trifásicos, también se consideran electromotores, los motores DC todavía tienen una importancia enorme a día de hoy. Gracias a sus características, (especialmente la regulabilidad de revoluciones y par motor, que es precisa y extremadamente variable), se utilizan en diversas aplicaciones industriales.

Desarrollo:

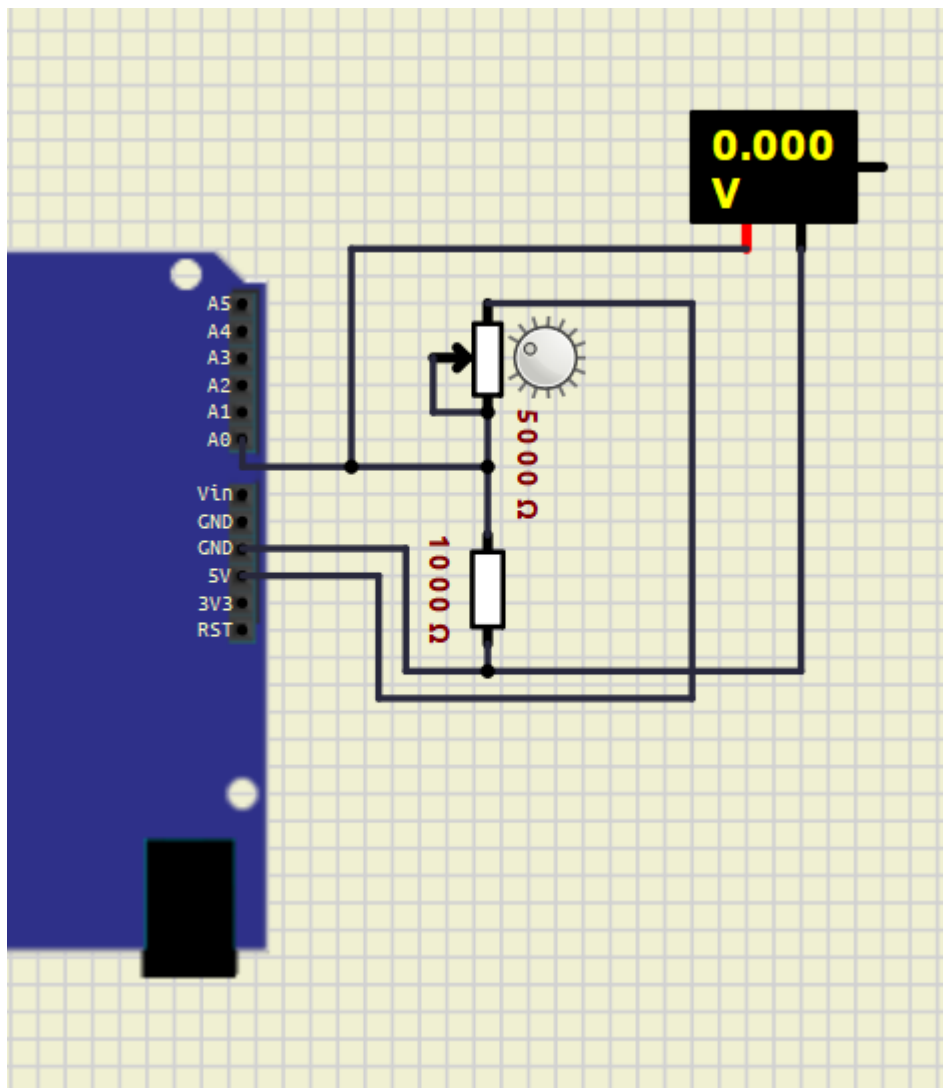
Habiendo hablado en abundancia acerca de los recursos utilizados, por fin nos vemos en la potestad de proseguir con el desarrollo de la idea, el cómo lo hicimos.



Para empezar, ingresamos un Arduino Uno dentro del Simulde, junto con un sensor HC-SR04 y una fuente de voltaje de hasta 5 voltios. Tras esto, conectamos el Trig del sensor, encargado de emitir la señal de sonido, al puerto número 2 del Arduino, y el Echo del sensor al puerto 3 PWM del Arduino. La idea detrás consiste en permitir que el HC-SR04 envíe ondas ultrasónicas hacia delante las cuales, al chocar contra algún objeto, rebotarán de vuelta hacia el Echo, el cual enviará la señal recibida al Arduino a través de la modulación PWM, que dividirá la señal en varios intervalos regulares de pulsos dependiendo de su intensidad. La fuente de voltaje será utilizada para simular la distancia entre el robot y un supuesto objeto.



El siguiente paso consiste en añadir los dos motores Dc-14 junto con un driver L298P, siendo poseedor de un puente H que permitirá al Arduino comunicarse con los dos motores. Lo siguiente es conectar los inputs del driver a los puertos 5, 6, 10 y 11, todos teniendo modulación PWM para interpretar las señales recibidas. Luego unimos los respectivos outputs a ambos extremos de ambos motores, de forma tal que avance a la hora de activarse y no retroceda en su lugar. Debido a que el robot está diseñado de tal forma que su lista de acciones a realizar está dada por las señales enviadas del Arduino, colocamos una fuente fija que mantuviese activados ambos motores con los puertos ENA y ENB, junto a su respectivo ground a partir de SENA y SENB.



Este fragmento está dedicado a ser un regulador de la velocidad del robot, haciendo uso de un potenciómetro con resistencia variable para modificar la corriente de 5 voltios, la cual proviene del propio Arduino. Este mecanismo está a su vez conectado al puerto ground y al A0 de la dichosa placa, esto último para hacer uso de esa frecuencia proporcionada para regular la velocidad a través del código.

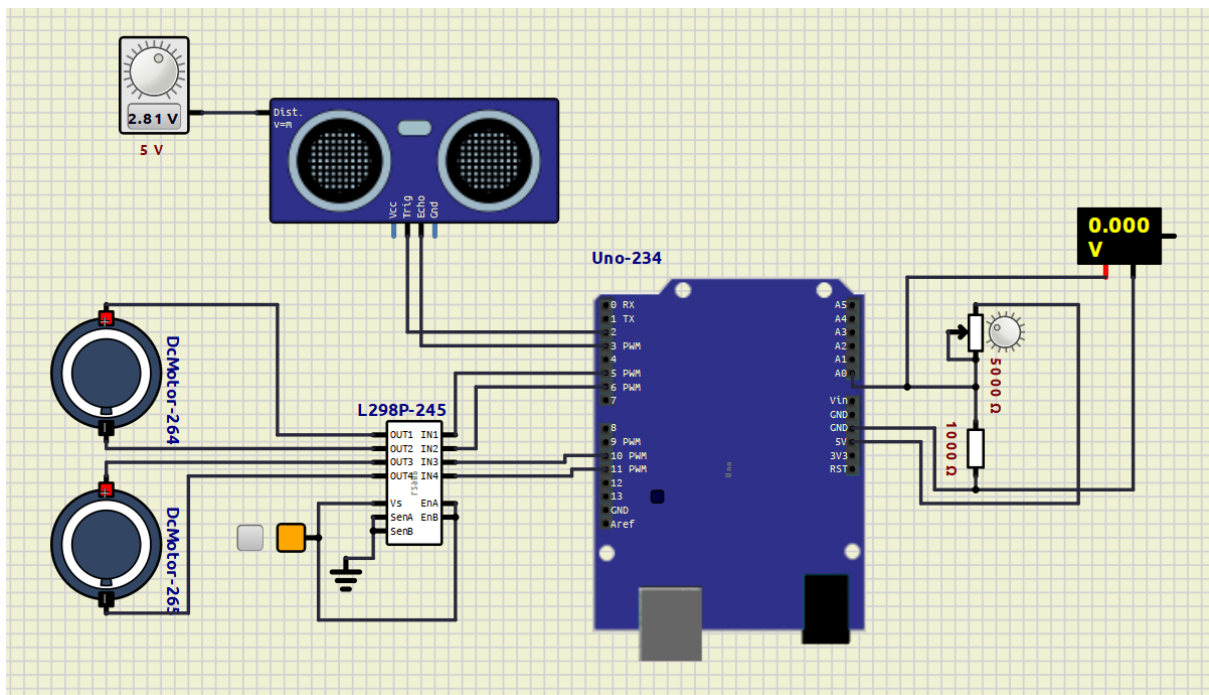
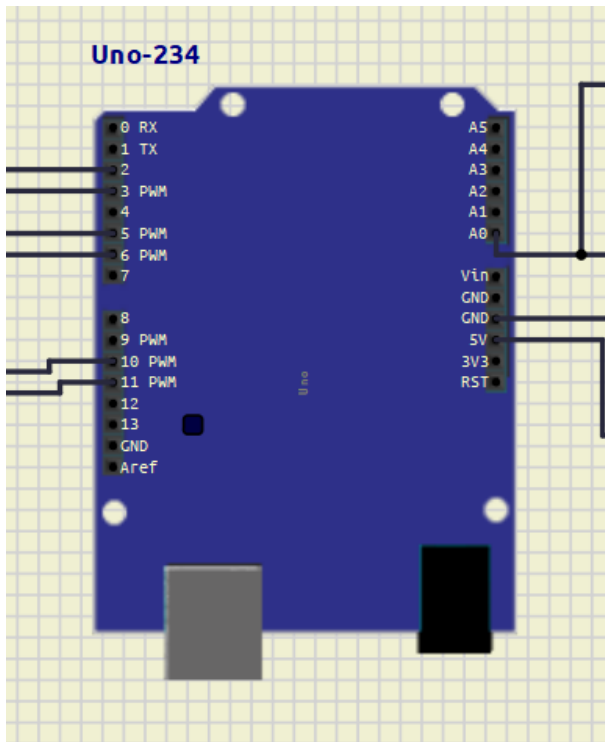
SimulIDE permite generar un gemelo digital y el código trabaja el conjunto para permitir darle movilidad al hardware, permite que las conexiones entre las entradas y los cables cobren utilidad. En tanto a lo que respecta al código, fue realizado en la plataforma IDE Arduino con el lenguaje C++.

En la primera parte del algoritmo se definen los pines Arduino a usar. Así, por ejemplo, `trick` y `echo` son los dos pines correspondientes al HC SR04; los pines de los motores son los M11, M12, M21, M22. Los pines 5, 6, 10, 11 son los PWM (pulse With Modulation) del Arduino los cuales regulan la velocidad. La entrada A0 se otorga al potenciómetro, ya que al ser un pin analógico del Arduino el cual nos permite medir el voltaje del potenciómetro, entre otros.

Luego en la función `void setup`, se definió con los `pinmode` el tipo de función que se quiere que cada pin tenga. Ejemplo, los motores como `outputs`, etc. Posteriormente, mediante `digitalWrite` se establecen en LOW los motores para evitar los falsos positivos.

En el `do-while` se “pide” al sensor que mida, lo que se hace de la siguiente manera: primero, se inicializa el pin en cero para evitar los falsos positivos, luego se establece un `delay` de 100ms para asegurarnos de que se reciba el siguiente pulso a tiempo. Acto seguido, el `trigger` se cambia a High para dar el pulso de disparo y se envía el pulso de 10 ms. y se regresa el `trigger` a cero. Luego se obtiene, mediante la variable `Dist`, el ancho del pulso del pin `echo` y se divide por 59 para convertir a cm. Este bucle se repite hasta que la condición del `while` se cumpla. Esta condición hace que la distancia sea mayor a cero. En otras palabras, el bucle se mantiene hasta que se acredite que los sensores están funcionando.

A modo de cierre, podemos decir que este código es el encargado de asignar una función y/o valor a los pines del Arduino. Éste, sumado a un `loop`, sería todo el código necesario para que el robot pueda reconocer el arduino y llevar a cabo las funciones impresas en el mismo. Cabe destacar que el segmento del `loop` puede inspirarse en el diagrama de flujo realizado para entregas anteriores.



Conclusión:

Al final, la meta de lograr una implementación digital del robot perteneciente a la extinta marca RobotGroup resultó ser un éxito rotundo, no sólo a nivel técnico, sino que también a nivel grupal y en lo referente al desarrollo personal de cada uno de nuestros integrantes. Por ello, vemos relevante el explayarnos en torno a esto.

Para este TP pudimos lograr una mejor organización con respecto al trabajo anterior, las tareas fueron mejor distribuidas en base a las capacidades y conocimientos de cada uno. Además realizamos una escucha activa de los integrantes que tenían un conocimiento más elevado sobre la robótica y pudimos aprender más para así poder aportar al trabajo práctico todos juntos y no de forma tan individual.

De igual forma, para seguir mejorando, estamos conversando, la productividad y la mejor organización del tiempo para poder ser más óptimos y enfocarnos activamente en otras materias.

Nos parece importante mencionar, en adición, qué cosas nos gustaron y llamaron la atención al realizar este trabajo práctico, de forma grupal e individual:

- Utilizar el programa simulIDE, ya que, con las herramientas que tiene permite realizar los circuitos de forma 100% digital sin la necesidad de utilizar los componentes físicos. Por lo cual simplifica mucho, a la hora de crear proyectos nuevos. Además es simple de usar, porque es fácil de entender.
- Descomponer algo complejo para nosotros, como es el robot de la marca RobotGroup, a algo tan simple como un diagrama de bloques/ flujo. Creemos que no solo nos puede servir para esta materia, sino que en general nos permite poder “simplificar” los problemas.
- Hubo multitud de novedades y enriquecedoras experiencias, que nos cautivaron con su majestuosidad a lo largo y ancho del desarrollo de este modelaje, y eso resultó en una gran motivación.

Ciertamente podríamos seguir incesantemente, detallando diferentes perspectivas, vivencias y aprendizajes, pero no es ese el motivo de ser de este proyecto grupal, por lo que opinamos de forma honesta que con lo ya mencionado es suficiente.

Links

- . [RobotGroup Argentina | Facebook](#)
- . [Guía del Arduinomaníaco: todo lo que necesitas saber sobre Arduino \(xataka.com\)](#)
- .
- .