



Escuela Técnica N°1 “Otto Krause”

Especialidad: Electrónica

Proyecto: TETOK-BOT



Taller de Electrónica

Docentes:

Basile, Leandro

Potenza, Nicolás

Integrantes:

Púa, Gonzalo Nicolás

Masó, Gabriel Andrés

Año y División:

6º ET 1ª

Ciclo Lectivo:

2018

Introducción

La robótica, entendida como el diseño, construcción, operación y aplicación de robots, es una de las ramas más interesantes dentro de la ingeniería electrónica no sólo por su interdisciplinaridad (abarca áreas de la informática, mecánica, ingeniería de control, inteligencia artificial, física, etc.) sino también por su dinamismo. En la actualidad, el uso masificado de robots tanto en el ámbito industrial como así también en lo cotidiano sumado al avance en la investigación en inteligencia artificial, ha dado un gran impulso al desarrollo de autómatas cada vez más potentes y precisos.

Dentro del ámbito educativo, la enseñanza de la robótica en las escuelas técnicas de especialidad electrónica presenta grandes desafíos precisamente por su interdisciplinaridad. Sin embargo, es dicha característica la que hace de la robótica uno de los ámbitos más potentes para la enseñanza. De esta manera, la elección de una metodología apropiada y el desarrollo de prácticas pertinentes se vuelven vitales para lograr un aprendizaje holístico sobre la materia.

En este marco, la construcción y puesta en marcha de un brazo robot bajo la metodología de trabajos por proyectos resulta una solución adecuada para abarcar y profundizar en una gran cantidad de contenidos curriculares de la especialidad electrónica.

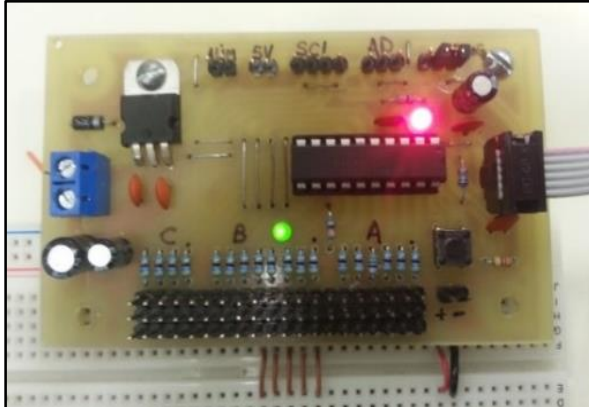
Objetivo

Desarrollar e implementar el control electrónico de un brazo robot que posee 3 ejes de libertad y una pinza como actuador con el fin de estudiar y aplicar los contenidos de la cinemática directa (uso de ecuaciones cinemáticas para calcular la posición final del actuador a partir de valores específicos) y de la cinemática inversa (cálculo del conjunto de parámetros a partir de una posición específica del actuador). El diseño del brazo fue realizado por la empresa Zortrax para su impresión en 3D y consiste en un TETOK-BOT a escala basado en el robot industrial KUKA.

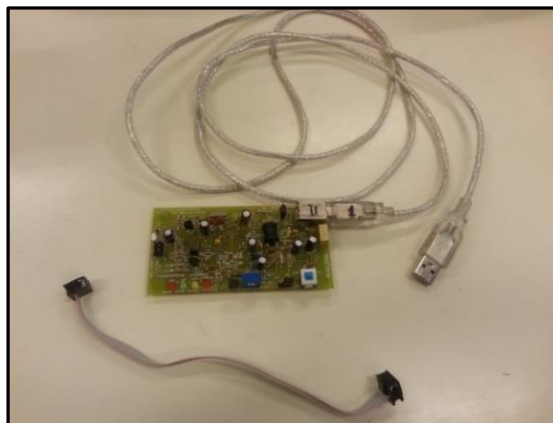


Desarrollo y funcionamiento general:

Este proyecto trabaja con un microcontrolador en particular (MC9S08SH8), que es un circuito integrado programable que contiene una Unidad de Control de Procesamiento, unidades de memoria y, puertos de entrada y salida. Se fabricó una placa para más comodidad a la hora de programar dicho microcontrolador y el lenguaje de programación utilizado es del tipo "C".



Para la programación de este microcontrolador dentro de la placa ya realizada utilizamos una placa programadora. Esta placa es un dispositivo portátil, que como su nombre indica, sirve para programar. Este dispositivo transmite información programada desde la computadora hacia la placa a programar. Estos datos que le envía la placa programadora, a la placa a programar, es creada mediante un programa llamado "CodeWarrior IDE". Básicamente la función de este software es designar funciones para varios tipos de microcontrolador, en este caso, FREESCALE MC9S08SH8.

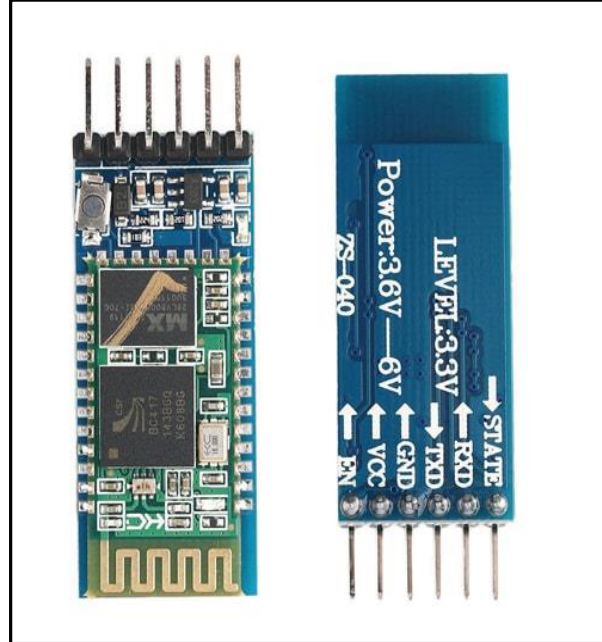
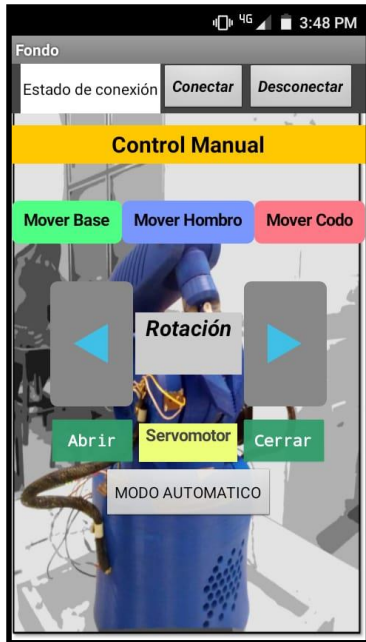


El funcionamiento de este proyecto consta de un control manual mediante una aplicación de un dispositivo móvil para Android, creada desde una página web llamada "ApplInventor" mediante un tipo de programación en bloques. Se encarga de enviar distintos tipos de datos dependiendo de la función del trabajo a realizar. Dentro de la aplicación se encuentran distintos botones, entre ellos:

- "Conectar": Muestra una lista con los enlaces establecidos con otros dispositivos Bluetooth y, al seleccionar el necesario, logra una conexión entre la aplicación y el módulo.
- "Desconectar": Desvincula la aplicación del módulo Bluetooth.
- "Motor Base": Selecciono el motor 'Base' para poder utilizarlo.
- "Motor Hombro": Selecciono el motor 'Hombro' para poder utilizarlo.
- "Motor Codo": Selecciono el motor 'Codo' para poder utilizarlo.
- Flechas: Logro controlar la rotación de cada uno de los motores.

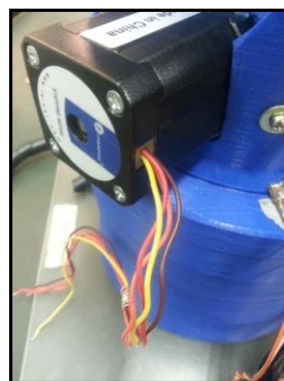
-“Abrir”: Logro que el efector final que, en este caso es una pinza, pueda aumentar el grado de separación entre sus mordazas.

-“Cerrar”: Logro que el efector final disminuya su grado de separación entre sus mordazas a cero.



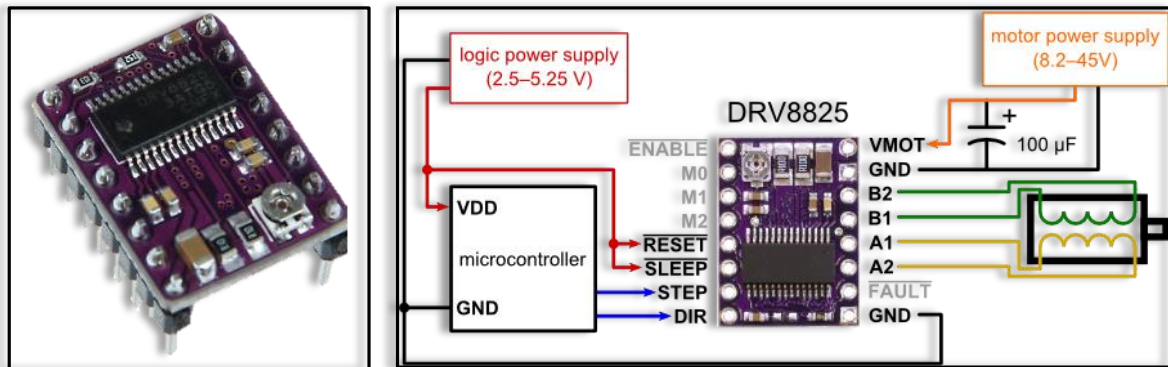
Cada uno de estos botones envía un tipo de dato al microcontrolador para distinguirlos entre sí dentro de la programación y, de este modo, asignarles distintos tipos de funciones. Esta aplicación se vincula y logra una transmisión y recepción de datos, junto con el microcontrolador, a través de un módulo Bluetooth.

Luego de lograr la transmisión de datos desde la aplicación, el microcontrolador, con la ayuda del módulo Bluetooth, se encarga de recibir cada uno de estos para lograr generar un tipo de movimiento en el TETOK-BOT. El tipo de motores a utilizar para controlar la rotación de cada uno de los ejes, son los llamados motores bipolares “Paso a paso”.



Cada uno de estos motores logra que el TETOK-BOT pueda maniobrar dentro de distintos ejes de libertad. Los motores bipolares poseen dos bobinas en su interior sin ningún punto medio donde salga un cable, por lo que se obtienen cuatro cables y cada par corresponde a las terminales de una bobina. Basan su funcionamiento en las fuerzas aplicadas por un campo electromagnético y son creadas al hacer circular una corriente eléctrica a través de sus bobinas. Dentro del motor, los dientes del rotor son atraídos por las bobinas activadas para así generar una rotación o giro del rotor.

Todos los motores bipolares a utilizar poseen un *driver* llamado “DRV8825” (Pololu) que fueron diseñados para el control de los mismos. El DRV8825 cuenta con límite de corriente ajustable, picos de corriente y exceso de temperatura. Funciona desde 8,2 hasta 45 V y puede entregar hasta aproximadamente 1,5 A (2,2 A con disipador). El control de corriente ajustable permite regular la salida de corriente máxima con un potenciómetro, que le permite utilizar tensiones superiores a la tensión nominal del motor paso a paso para lograr mayores tasas de paso.



Esta placa posee tres pines (“M0”, “M1” y “M2”) en donde se pueden configurar los micropasos, enviándoles distintos datos, según el tipo de micropasos con el que se trabaja. Estos, según el tipo de configuración previa, cuentan con diferentes tipos de micropasos, entre ellos:

-Un paso completo: para hacer que el rotor da una vuelta completa (360°), el driver debe enviarle 200 pasos (en este caso cada paso tendrá un giro de 1,8°). Obtiene un mayor torque, realiza movimientos bruscos y mayor ruido.

-Medio paso: con el driver configurado, para dar un giro completo de 360° el driver deberá enviarle 400 pasos ya que cada paso tendrá un ángulo de 0,9°. Genera un menor torque, movimientos suaves y menor ruido.

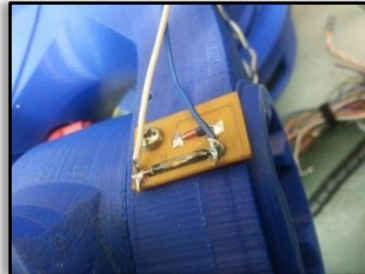
-¼-step: para este caso, ocurrirá lo mismo, pero dividido a la mitad a comparación del medio paso. Es decir, se enviarán 800 pasos porque cada paso tendrá un ángulo de 0,45°. Genera un torque muchísimo menor, movimientos muchos más suaves y mucho menos ruido.

Y continúa funcionando de la misma manera hasta llegar no más allá de los 1/32-step. La finalidad del micro-paso es poder generar un movimiento más fluido al momento de que gire el motor.

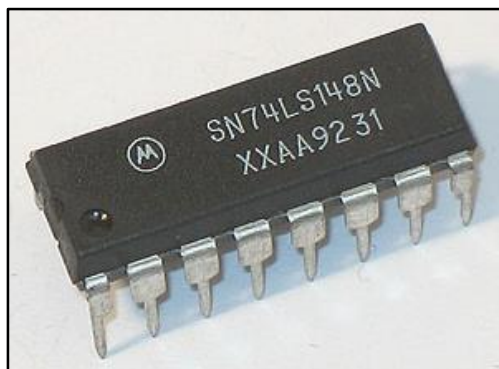
MODE0	MODE1	MODE2	Microstep Resolution
Low	Low	Low	Full step
High	Low	Low	Half step
Low	High	Low	1/4 step
High	High	Low	1/8 step
Low	Low	High	1/16 step
High	Low	High	1/32 step
Low	High	High	1/32 step
High	High	High	1/32 step

Además tiene dos pines en particular (“Dir” y “Step”) que controlan a dirección del motor (“Dir”) y envía un pulso para que el motor avance un paso (“Step”).

Cada eje posee un grado aproximado de libertad de 0° a 180° y, para lograr esta limitación de movimiento, se utilizaron Finales de carrera. Estas son unas placas que se encuentran en los extremos del área de trabajo, cuentan con una especie de interruptor que se activa o desactiva y envían un dato al microcontrolador cuando un imán se encuentra en sus superficies. La cantidad de finales de carreras utilizados se distribuye en dos por motor, es decir, un total de seis.



Partiendo de que la cantidad de finales de carrera como entradas es inmensa, utilizamos un codificador con prioridad con el que los identificamos con mayor facilidad y que reduce las ocho entradas a tres, estas entradas se activan con un valor lógico bajo (0V) y se desactivan con un valor lógico alto (5V). Es decir, reducimos ocho entradas (finales de carrera y dos sin usar conectadas en las entradas del codificador) a tres salidas que, estas a su vez, van conectadas en las entradas del microcontrolador.

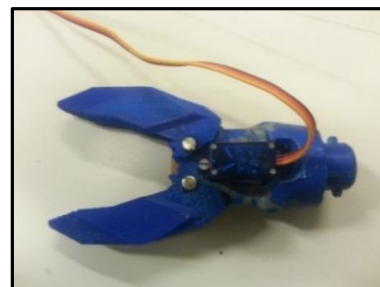


SN54/74LS148
SN54/74LS748
FUNCTION TABLE

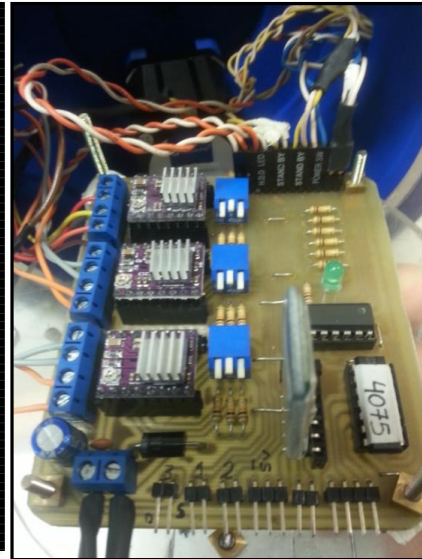
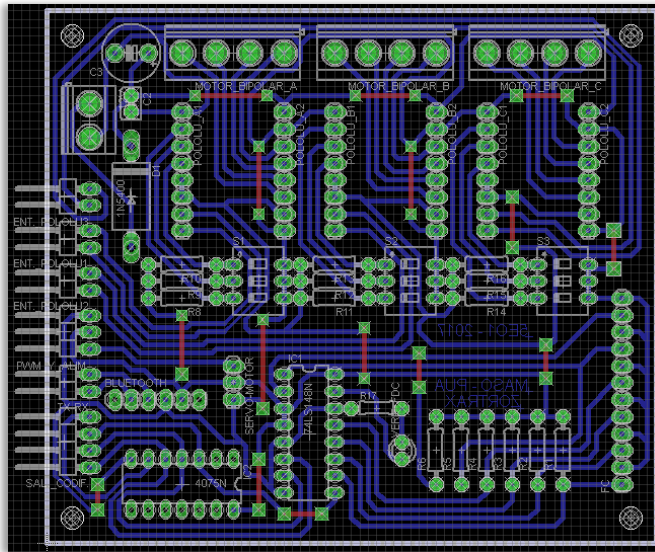
INPUTS									OUTPUTS				
EI	0	1	2	3	4	5	6	7	A2	A1	A0	GS	EO
H	X	X	X	X	X	X	X	X	H	H	H	H	H
L	H	H	H	H	H	H	H	H	H	H	H	H	L
L	X	X	X	X	X	X	X	L	L	L	L	L	H
L	X	X	X	X	X	X	L	H	L	L	H	L	H
L	X	X	X	X	X	L	H	H	L	H	L	L	H
L	X	X	X	X	L	H	H	H	L	H	H	L	H
L	X	X	X	L	H	H	H	H	H	L	L	L	H
L	X	X	L	H	H	H	H	H	H	L	H	L	H
L	X	L	H	H	H	H	H	H	H	H	L	L	H
L	L	H	H	H	H	H	H	H	H	H	H	L	H

Al ser un codificador con prioridad, se refiere a que hay un orden jerárquico en cuanto a la activación de cada una de las entradas y que, si se activan más de una, se ignoren por completo las entradas de menor importancia. Dependiendo del final de carrera activado se envía por la salida del codificador, un número en binario de 3 bits. Además el codificador tiene la función de enviar, mediante un pin en particular (GS), la información sobre si alguna entrada se encuentra en un valor lógico bajo.

Como efector final, en este caso, utilizamos una pinza que es manejada por un servomotor que, a su vez, se controla mediante una señal PWM (Modulación por ancho de pulso) enviada desde el microcontrolador. El tipo de señal varía según el botón pulsando desde la aplicación, ya sean: "Abrir" o "Cerrar".

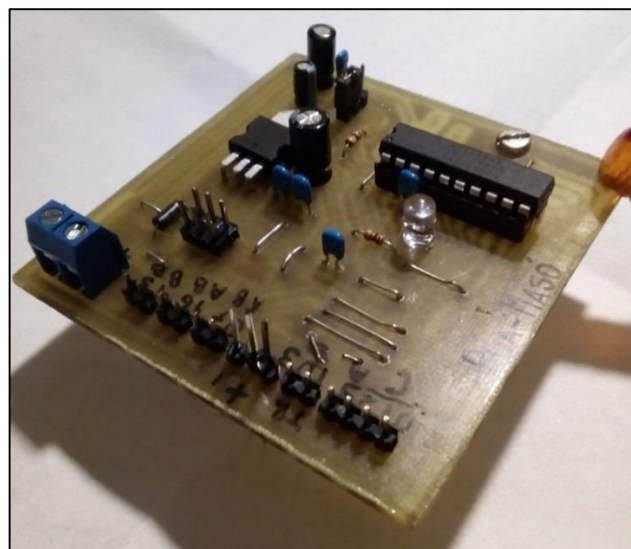
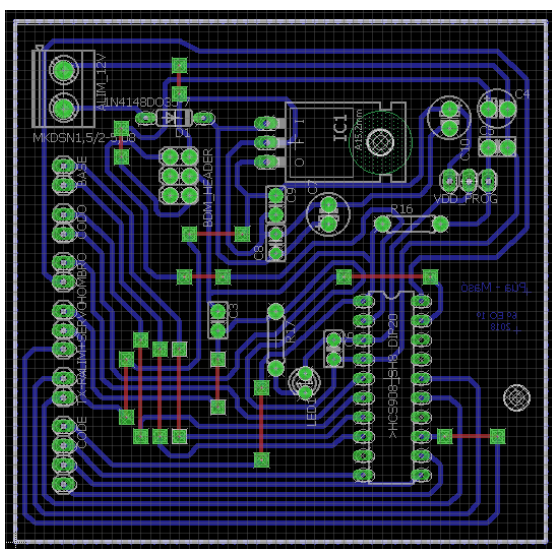


Fabricamos una placa de potencia para poder controlar cada una de las tres placas Pololu, poder configurar las mismas, programarlas y poder alimentar los motores (12V) junto con otros componentes (5V).



Esta placa fue diseñada a través de un programa llamado "Eagle" con el fin de obtener comodidad al programar cada uno de estos componentes en simultáneo, entre ellos: codificador, Pololu's y módulo Bluetooth. Dentro de esta placa también se encuentran y se conectan, los finales de carrera, los interruptores para la configuración de cada Pololu, también borneras para alimentar cada motor. La placa se encuentra dentro de la base del TETOK-BOT.

Para evitar que las conexiones entre las placas multipropósito y de potencia estén a la vista, diseñamos y fabricamos otra placa multipropósito pero que esta sea dedicada únicamente para la placa de potencia. Esta placa contiene exactamente lo mismo que la anterior solo que esta roteada (es decir, "que las pistas están puestas") de una manera para que coincida con las entradas de la placa de potencia. Esta placa funciona como un *shield*, ya que va montado por encima de la placa de potencia.



Además, para brindar todavía más comodidad a la hora de trabajar con el dispositivo, implementamos una base hecho de acrílico bordeado con madera en la parte inferior del TETOK-BOT y agregamos mallas a cada cableado para que este se encuentre presentable.



Observaciones

Estas observaciones son aquellas conclusiones que se vieron e hicieron durante realización del proyecto. Constan de dar a conocer qué cosas resultaron difíciles, sus soluciones y las formas correctas de hacer cosas tales que en un futuro se puedan mejorar.

Anotaciones y observaciones

Hubo que determinar el inicio y el final del giro en el servomotor en la programación ya que no se trabaja con la rotación entera.

Se tuvo en duda la forma con la que la pinza debe sujetar los objetos entre dos formas: que la pinza se vaya cerrando lentamente o que se cierre de manera instantánea.

El cableado realizado se aplicó nuevamente para las entradas del servomotor.

Se fabricó una placa para controlar directamente tres motores bipolares, que logran los movimientos del TETOK-BOT.

Se cambiaron los motores unipolares por motores bipolares, no se utilizaron ya que la placa de motores funciona con motores bipolares.

Se mejoró el barrido del motor bipolar usando micro-pasos.

El codificador utilizado no estaba previsto en un principio, y gracias a su funcionamiento se redujo la cantidad de salidas a 3 en la placa (ya que eran 12 salidas del robot y 12 entradas al microcontrolador).

No se logró obtener un codificador que funcione con niveles lógicos altos (+5V), por consiguiente, se consiguió uno que lo haga con valores lógicos bajos (0V).

Se agregaron 3 switch para la libre configuración del DRV8825 de manera externa, para no establecer una configuración fija.

Para más comodidad se modificó la base del TETOK-BOT para la placa de motores ya que, al haber un error, debe revisarse de una manera más rápida.

Debajo de la base se le agregaron pedazos de madera de forma rectangular para no dañar la placa de motores (cuando se tenga que realizar una revisión) y al mismo tiempo mantener estable el TETOK-BOT.

No se fabricará una placa para el control manual ya que el objetivo final es controlarlo mediante una aplicación desde el celular y porque, para lo económico, no conviene hacerlo. Al ser simplemente cuatro pulsadores, se tomó la decisión de utilizarlos en un protoboard.

Errores generales

Nunca realizar modificaciones en el cableado del driver cuando el ENABLE esté activado, es decir, si por ejemplo se desconecta el motor con el driver estando todavía alimentado, de seguro este se quemará.

El codificador, al tener prioridad, no dejará que se activen más de dos finales de carrera. De lo contrario, ignoraría uno de los dos, dependiendo de qué final de carrera se haya activado.

Al principio, se utilizó una configuración determinada para enviar los pasos mediante el Pololu, lo cual no es correcto.

Enviarle al servomotor un porcentaje de ciclo útil más grande o más pequeño de lo que necesitaría, ya que este se encuentra limitado por la pinza.

En principio, la cantidad de entradas y salidas al microcontrolador era inmensa, superaba los veinte, cosa que a la placa multipropósito no le alcanza ya que posee dieciocho pines para entradas y salidas.

Al realizar el diseño de la placa de motores para tener todo más cómodo, la cantidad de salidas seguía siendo mayor por los finales de carrera.

Conclusiones

Como precaución, si se necesita modificar alguna conexión en el *driver*, hay que asegurarse previamente de quitarle la energía y para evitarse problemas.

Se tomó en cuenta y se decidió activar un solo final de carrera a la vez, para que el codificador pueda detectarlo correctamente y enviar en la salida sus respectivos datos sin tener que ignorar otra activación de final de carrera.

Cada tipo de paso a enviar tiene su configuración establecida. Siempre se configura en la parte de hardware y no en la programación.

Se encontraron las limitaciones de cada extremo del eje del servomotor para la variación del ciclo útil adecuada.

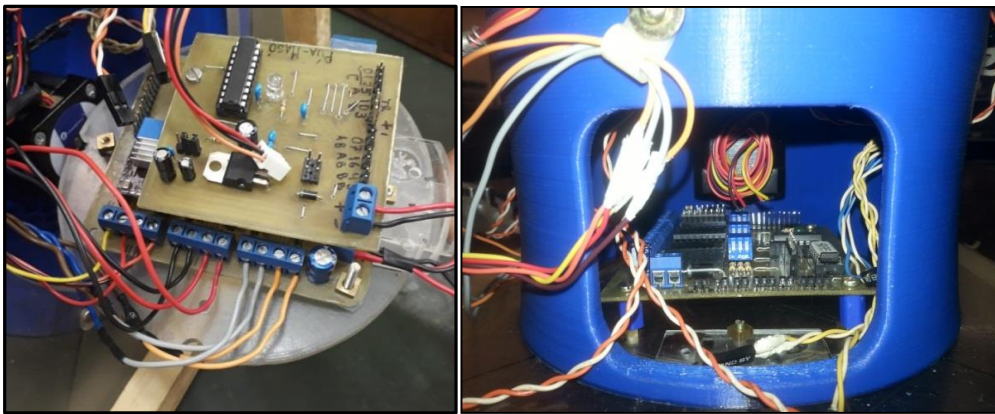
Se decidió realizar una placa de control de potencia para ubicar todo en un lugar más apropiado y para que en la programación sea más eficiente.

Se redujo bastante la cantidad de salidas al utilizar el codificador 74LS148 por su funcionamiento. De tener doce salidas, se redujeron a tres.

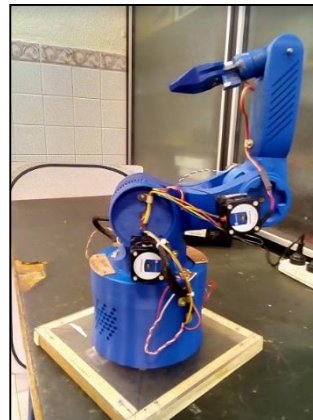
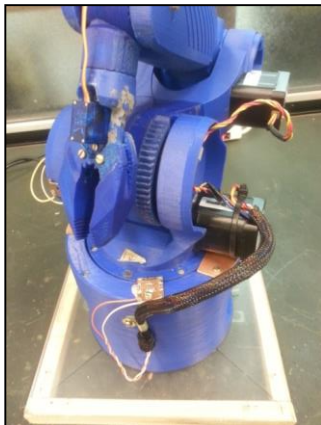
La idea final del proyecto es implementar dos modos de control de trabajo: el control manual y automático. Hasta ahora el estado actual del proyecto se encuentra con un control manual mediante una aplicación de un dispositivo inteligente y se está desarrollando la automatización.

Vistas adicionales del TETOK-BOT en general:

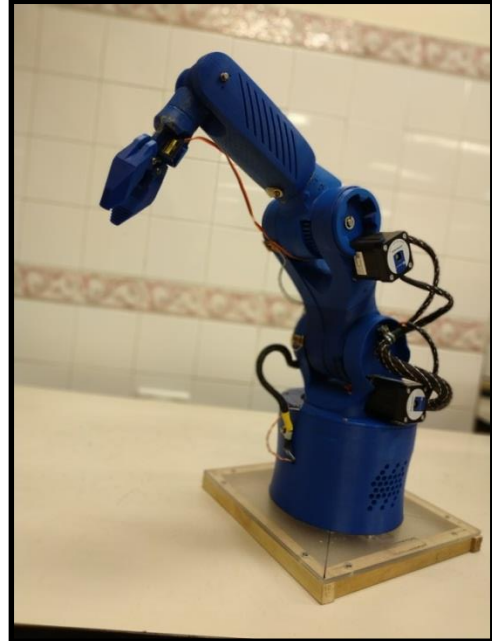
-Montura de placa multipropósito dedicada sobre la placa de potencia, que estas a su vez, se encuentran dentro de la base del TETOK-BOT.



-Diferentes vista del TETOK-BOT y visualización de la base construida.



Proyecto TETOK-BOT: Vistas



Video del TETOK-BOT:

-Plataforma Youtube:

<https://youtu.be/Ou1mqjB87FQ>

Agradecimientos:

Principalmente al taller de electrónica con todos sus docentes por darnos la facilidad de obtener las herramientas, materiales y lo más importante el conocimiento.

- Leandro Basile
- Nicolás Potenza
- Rodolfo Gramont
- Adrián Durante
- Hugo Campoy