

Design and Implementation of a Selective Compliant Articulated Robot Arm with Level One of Artificial Intelligence to Develop a Chess Game for training purposes.

Universidad San Carlos de Guatemala, Facultad de Ingeniería, Escuela de mecánica eléctrica, Laboratorio de Robotica
Oscar Gabriel Fuentes Lanfur, Student Member 94300573. Juan Diego Arias Rossa, Student Member 94300523. Rodrigo Otoniel Méndez
Canú, Student Member 94976087 .
 hbkgabriel@ieee.org
 diego.jd610@ieee.org
 mendezrod57@gmail.com

Resumen—Diseño e implementación de un brazo robótico de cumplimiento selectivo (SCARA) capaz de interpretar, analizar y desarrollar una partida de ajedrez completa contra una persona en tiempo real, presentando cierto nivel de dificultad según lo desee el usuario en entrenamiento. El robot se encuentra programado con algoritmos de inteligencia artificial nivel 1 con el fin de interpretar y ejecutar un movimiento en un tablero de ajedrez físico previamente modificado. Se expondrá acerca del sistema óptico del robot y sus algoritmos programados, los cuales hacen posible identificar las piezas y las jugadas realizadas por el usuario en base a la librería OPEN CV. El robot posee cuatro grados de libertad, cada uno con motores paso a paso, logrando una mayor precisión utilizando las técnicas de cinemática inversa, cinemática directa y parámetros Denavit-Hatemberg. Se implemento el microcontrolador de Texas instruments Tiva C TM4C123GH6PM para el sistema de control del brazo robótico, Python 2.7 sobre Ubuntu 16.04 y una cámara PS3 EYE como sistema óptico.

I. INTRODUCCIÓN

El ajedrez es uno de los juegos de estrategia más entrenados que existen con cierta antigüedad, en su versión de competición es reconocido mundialmente como un deporte siendo un juego para dos personas. Una partida de ajedrez puede ser jugada contra una persona o contra un programa de ajedrez ejecutado desde una computadora, el problema de la segunda opción es que el usuario debe estar sentado con la vista fija en la pantalla. Observar la pantalla de un ordenador o dispositivo móvil conlleva a daños oculares y daños a la visión entre otros que afectarán a largo plazo la salud del usuario. Es requisito indispensable el constante entrenamiento para el jugador profesional de ajedrez por ende dañaran su vista si practican con un software dedicado, además las partidas de ajedrez tienen la peculiaridad de durar varias horas en juego. La otra opción es entrenar con un segundo jugador lo cual depende de la disponibilidad de tiempo de la segunda persona. El ajedrez se juega sobre un tablero cuadrado de 64 casillas alternadas en color y con dieciséis piezas cada oponente; así mismo las partidas oficiales en torneos se juegan de la misma forma. La tecnología avanza rápidamente y la solución a dicho problema es el desarrollo de un prototipo robot que juegue

ajedrez, proporcionando al usuario en entrenamiento niveles de dificultad, disponibilidad de tiempo y entrenamiento continuo sin alterar su salud física ocular.

II. DESARROLLO DE CONTENIDOS

II-A. Inteligencia artificial

Existen 3 niveles de inteligencia Artificial, son aplicables en todos los seres inteligentes reales o hipotéticos que puedan existir, esto incluye, robots, Aliens y Monos súper Dotados.

- * Nivel 1, inteligencia Matemática: Esta inteligencia se refiere a la habilidad para hacer cálculos y algoritmos matematicos dados con una serie de variables.
- * Nivel 2, inteligencia creativa: Las computadoras muy avanzadas con inteligencia matemática bien desarrollada a lo sumo pueden simular cierto nivel de creatividad, pero en realidad no saben hacer nada para lo que no estaban programadas.
- * Nivel 3, inteligencia filosófica: Se refiere a un nivel de inteligencia muy superior. Es el tipo de inteligencia que nos hace hacernos preguntas filosóficas del tipo ¿estaremos sólo en el universo? o ¿Cuál es el propósito de nuestra vida? es difícil imaginar a una máquina haciéndose estas preguntas.

II-B. Brazo Robótico

Un brazo robótico es un tipo de brazo mecánico, normalmente programable, con funciones parecidas a las de un brazo humano. Las partes de estos manipuladores o brazos son interconectadas a través de articulaciones que permiten tanto un movimiento rotacional (tales como los de un robot articulado), movimiento traslacional o desplazamiento lineal.

II-C. Brazo Robótico SCARA

Un brazo robotico SCARA (acrónimo que responde por sus siglas en inglés a Selective Compliant Assembly Robot Arm o Selective Compliant Articulated Robot Arm) es un robot con posicionamiento horizontal. Los Robots SCARA se conocen por sus rápidos ciclos de trabajo, excelente repetitividad, gran capacidad de carga y su amplio campo de aplicación.

II-D. Ajedrez

El ajedrez es un juego, entre dos personas, cada una dispone de 16 piezas móviles que se colocan sobre un tablero dividido en 64 casillas o también llamados escaques. En su versión de competición el ajedrez está considerado como un deporte. Al principio del juego cada jugador tiene dieciséis piezas: un rey, una dama, dos alfiles, dos caballos, dos torres y ocho peones. Se trata de un juego de estrategia en el que el objetivo es derrocar al rey del oponente. Esto se hace amenazando la casilla que ocupa el rey con alguna de las piezas propias sin que el otro jugador pueda proteger a su lo que trae como resultado el jaque mate y este es el fin de la partida.

II-E. Metodología Denavit-Hatenberg

La convención o metodología de Denavit-Hartenberg (DH) permite establecer la ubicación de los sistemas de referencia de los eslabones en los sistemas robóticos articulados, ya sean prismáticas o de revolución, con cadenas cinemáticas abiertas. Dichos parámetros facilitan el cálculo de la posición del actuador final del robot en base a los elementos del robot y también los ángulos de los elementos en base a las coordenadas del actuador final del robot.

II-F. Cinemática directa

La cinemática de un robot es el estudio de los movimientos de este. En un análisis cinemático la posición de cada uno de los elementos y el efector final del robot son calculadas sin considerar las fuerzas que causan el movimiento en base a los parámetros Denavit-Hatenberg del mismo.

II-G. Cinemática Inversa

En Robótica, la Cinemática inversa (IK) es la técnica que permite determinar el movimiento de una cadena de articulaciones para lograr que un actuador final se ubique en una posición concreta. El cálculo de la cinemática inversa es un problema complejo que consiste en la resolución de una serie de ecuaciones cuya solución normalmente no es única.

II-H. Cinematica Inversa: Método Geométrico

El procedimiento se basa en encontrar un número suficiente de relaciones geométricas en las que intervendrán las coordenadas del extremo del robot, sus coordenadas articulares y las dimensiones físicas de sus elementos.

II-I. Pychess

PyChess es un programa de ajedrez (o cliente de ajedrez) por computadora publicado como software libre con licencia GPL. Pychess incluye su propio motor de ajedrez, permitiendo así al usuario jugar contra la máquina, contra otro jugador y también jugar en internet utilizando el Servidor de Ajedrez Gratuito por Internet (FICS).

II-J. Open CV

Open CV es multiplataforma, existiendo versiones para GNU/Linux, Mac OS X y Windows. Contiene más de 500 funciones que abarcan una gran gama de áreas en el proceso de visión, como reconocimiento de objetos (reconocimiento facial), calibración de cámaras, visión estéreo y visión robótica

II-K. Motor Stepper

El motor paso a paso conocido también como motor de pasos es un dispositivo electromecánico que convierte una serie de impulsos eléctricos en desplazamientos angulares discretos, lo que significa que es capaz de girar una cantidad de grados (paso o medio paso) dependiendo de sus entradas de control.

III. MARCO METODOLÓGICO

III-A. Brazo Robotico SCARA

El prototipo consta de 3 grados de libertad articulados con motores paso a paso girando alrededor del eje Z sobre el plano XY más un grado de libertad articulado con un servo motor, este se encarga de realizar el movimiento que eleva la pieza de ajedrez alrededor del eje Y sobre el plano XZ. Para el movimiento paralelo de los motores paso a paso se utilizó lógica de algoritmos contadores programada en el microcontrolador Tiva c 123GH6PM. El microcontrolador se comunica por medio del protocolo de comunicación serial UART a la computadora la cual envía los valores de la cantidad de pasos a moverse (motor paso a paso) y un Angulo en específico (servomotor).

III-B. Parámetros Denavit-Hatenberg

Se obtuvieron los parámetros Denavit-Hatenberg con las dimensiones físicas del brazo robótico, esto con el objetivo calcular la cinemática directa del brazo y poder transformar coordenadas del sistema de referencia cero al sistema de referencia del actuador final.

Country List				
No. Articulación	θ_i (grados)	d_i	a_i	α_i (grados)
1	θ_1	13.25cm	14.40cm	0
2	θ_2	2.30cm	13.60cm	0
3	θ_3	-3.75cm	2.40cm	90
4	θ_4	-4.75cm	15.20cm	-90

Tabla I: Tabla con los parámetros Denavit-Hatenberg

III-C. Cinemática directa del brazo SCARA de 4 grados de libertad

Matriz de transformación Denavit-Hatenberg para la cinemática directa:

$$\begin{bmatrix} \cos \theta & -\cos \alpha \sin \theta & -\sin \alpha \sin \theta & a_i * \cos \theta \\ \sin \theta & \cos \theta \cos \alpha & -\cos \theta \sin \alpha & a_i * \sin \theta \\ 0 & \sin \alpha & \cos \alpha & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

III-D. Transformación directa del sistema de referencia 0 del brazo robótico con el sistema de referencia de la cámara

Para realizar esta transformación recurrimos a la metodología de la cinemática directa diseñando una matriz de transformación M para convertir coordenadas del marco de referencia de la cámara al marco de referencia cero del brazo SCARA, se utilizaron milímetros como unidad de medida. Antes de realizar dicha transformación es necesario realizar la conversión de pixeles de la cámara al sistema de medición del brazo SCARA. Se utilizó la siguiente formula:

$$X = 670 * \frac{P_x}{640} \quad (1)$$

Donde X es la posición en el eje X en milímetros y P_x es la posición en el eje X en pixeles de la cámara. Se conoce el desplazamiento del punto de referencia (0,0) de la cámara con el punto de referencia del brazo (0,0), dicho desplazamiento es 320 milímetros sobre el eje X y 30 milímetros sobre el eje Y. También existe una rotación sobre el eje Y de -90 grados. Realizada la conversión de pixeles a milímetros el siguiente paso es ingresar los datos en la matriz de transformación M.

$$M = \begin{bmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 1 & \cos \beta \end{bmatrix} \quad (2)$$

Sustituyendo β por -90 grados la matriz de transformación queda de la siguiente manera.

$$M = \begin{bmatrix} -1 & 0 & 0 & 335 \\ 0 & 1 & 0 & 25 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

III-E. Cinemática inversa

Haciendo uso del método geométrico para la cinemática inversa del brazo robótico, se estableció un conjunto de triángulos imaginarios con el propósito de aplicar la ley del coseno y encontrar los ángulos q_1 , q_2 y q_3 respectivamente de cada articulación. El brazo Robótico cuenta con tres grados de libertad con el eje de acción sobre el plano cartesiano XY y ángulo de rotación alrededor del eje Z θ grados. La cinemática inversa tiende a encontrar infinitos resultados o posiciones de las articulaciones para la solución de un mismo punto. Para escoger la mejor solución posible de la cinemática inversa, se parametrizó un nuevo vector R en función de las coordenadas del punto final de la siguiente manera:

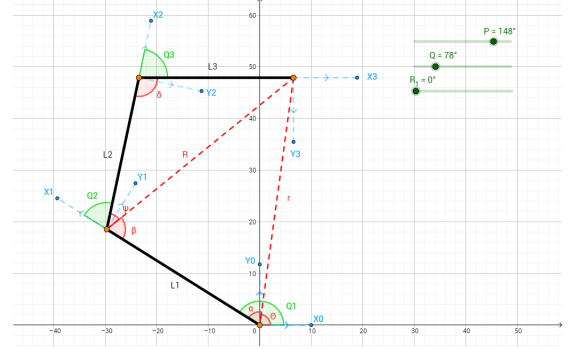


Figura 1: Representación gráfica de los grados de libertad sobre el plano "X", "Y"

$$R = 312 * \frac{r}{456} \quad (4)$$

Donde el largo total de L2 y L3 es 312 milímetros, r es un vector distancia desde el origen del sistema de referencia 0 del brazo al punto (x,y) donde se requiere que el efector final se posicione y 456 representa el largo total del brazo.

La solución a los ángulos q_1 , q_2 y q_3 se definen como: El primer ángulo q_1 de la primera articulación se define como la suma de θ y α . para determinar los valores de dichos ángulos procedemos a desarrollar la ley de cosenos sobre el primer triángulo de la siguiente manera:

$$q_1 = \theta + \alpha \quad (5)$$

$$q_2 = 180 - \Psi - \beta \quad (6)$$

$$q_3 = 180 - \gamma \quad (7)$$

III-F. Conversión de grados a pasos del motor stepper

$$ST1 = \frac{q_1}{1.8} * \frac{90}{16} \quad (8)$$

Donde ST1 define la cantidad de pasos de cada motor respectivamente; q_1 definen el ángulo resultante de la cinemática inversa; $\frac{90}{16}$ define la relación de los engranajes del brazo con el motor y para finalizar 1.8 es la cantidad en grados por paso ejecutado.

III-G. Sistema óptico del Robot

El robot es capaz de interpretar la posición del tablero permitiendo que él mismo pueda girar o rotar en cualquier posición sin alterar la partida de ajedrez y sus movimientos. El robot es capaz de visualizar la jugada del usuario para luego interpretarla y realizar una jugada en respuesta en tiempo real. Para la detección de la rotación del tablero y las jugadas ejecutadas por el usuario se implementó una cámara PS3 EYE, dicha cámara se conecta a la computadora para enviarle imágenes del tablero en tiempo real para su análisis respectivo. se utilizó la librería OpenCV para el tratamiento y análisis de imágenes.

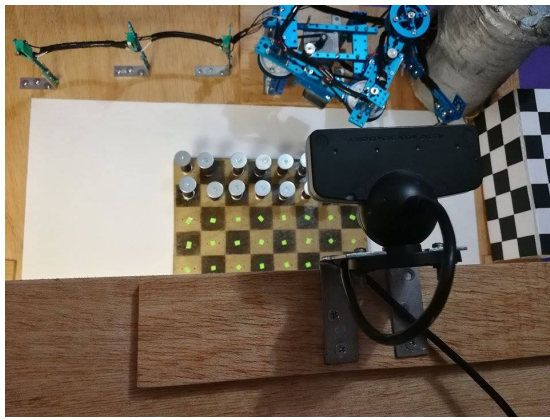


Figura 2: Cámara del brazo robotico

III-H. Inteligencia Artificial del brazo

Toda la programación del robot esta realizada en lenguaje Python 2.7 a su vez todos los algoritmos de inteligencia artificial que se mostraran a continuación están programadas en el lenguaje orientado a objetivos. Se utilizaron varios algoritmos de inteligencia artificial de nivel 1 en el robot para las siguientes funciones:

1. Rotación del Tablero: Para permitir que el tablero rotara y se pudiera encontrar en cualquier posición dentro del área diestra del robot, se tomó una foto del tablero en una posición de referencia. Después, por medio de la búsqueda de las características se comparaba la imagen de referencia con una imagen de la posición actual del tablero. Aplicando el algoritmo SIFT (Scale-Invariant Feature Transform), con la ayuda de OpenCV, se podía determinar la rotación del tablero y su nueva posición. Este algoritmo permite encontrar la matriz de transformación de la posición actual del tablero respecto a la posición de referencia con la cámara. De esta forma pueden relacionarse los píxeles con las posiciones reales en el tablero.
2. Interpretación y posición de las pizas: Cada casilla del tablero posee un recuadro de color verde fosforescente colocado en el centro. Al tomar la imagen del tablero para conocer su posición, se procesa para encontrar el centro de cada casilla. Para este proceso, el primer paso es cambiar el espacio de colores de BGR (usado por defecto en OpenCV) y convertirlo a HSV. Este permite un mejor procesamiento del color para binarizar la imagen, mostrando resultados de mucha mejor calidad. Después de tener la imagen binarizada, se aplica la operación de OpenCV `cv2.findContours` para encontrar cada uno de los recuadros de color. Una vez localizados se encuentra el centro, y así se obtiene la posición exacta de cada casilla. Por medio de la matriz de transformación encontrada en el paso anterior, se puede actualizar la información de la posición cada vez que se inicia el juego.
3. Memoria del robot: Se genera un diccionario, que relaciona el nombre de cada casilla (a1, e2, g3, etc) con su coordenada. A partir de este punto se pueden realizar

jugadas que se escriben en el formato estándar de un juego de ajedrez, y el programa es capaz de relacionar las casillas con las posiciones de cada una.

4. Movimientos de Ajedrez: Luego de interpretar los movimientos del usuario y llevar el registro de las posiciones de las piezas procedemos a utilizar la librería Pychess. Pychess posee una función que permite ingresar un movimiento en términos de coordenadas de ajedrez (fila,columna) y devuelve una coordenada en respuesta (fila,columna) utilizando una serie de algoritmos matemáticos para escoger el movimiento más inteligente.
5. Cinemática inversa: La cinemática inversa se programó utilizando las funciones matemáticas de Python y la librería MATH la cual permite realizar cualquier función trigonométrica.

IV. RESULTADOS

Aplicando y combinando todas las técnicas previamente explicadas en el marco metodológico obtenemos un robot capaz de identificar la posición del tablero con sus piezas, interpretar los movimientos del oponente, guardar y analizar los movimientos de la partida de ajedrez y responder con un movimiento inteligente en respuesta al movimiento del oponente.

1. Diseño de un brazo robótico SCARA con 4 grados de libertad, cada uno programado con algoritmos de cinemática inversa para el movimiento de las piezas físicas de ajedrez, implementado con motores paso a paso para obtener la mayor precisión posible sobre el espacio de trabajo definido.
2. Diseño de un programa capaz de identificar el ángulo de rotación del tablero medio de los algoritmos implementados haciendo uso de Open CV y una cámara digital PS3 EYE.
3. Interacción robot humano gracias los algoritmos de inteligencia artificial nivel 1 que le permiten llevar acabo y culminar exitosamente una partida de ajedrez completa con el usuario en entrenamiento.
4. Link del video interactivo del robot en accion: <https://youtu.be/WJu9PnL8gPE>



Figura 3: Brazo robótico

V. CONCLUSIONES

- Los brazos robóticos SCARA tienen la característica de ser muy precisos dentro del espacio de trabajo sin importar la aplicación que se otorga.
- Uno de los principales problemas es la luz, que al modificarse cambian los parámetros de binarización, lo que puede provocar que algunas casillas no se detecten, o se den falsos positivos. Otra de las desventajas de este método es la resolución de la cámara, que si tiene poca resolución tendrá mayor error.
- La inteligencia artificial hoy en día es una herramienta muy útil y novedosa que nos permite realizar robots inteligentes que puedan tomar decisiones en base al entorno o ecosistema en el que se encuentre. Gracias a la inteligencia artificial podemos realizar prototipos de entrenamiento para jugadores profesionales de ajedrez.

REFERENCIAS

- [1] MediaWiki.*Open CV* [En línea]. Disponible en: <https://es.wikipedia.org/wiki/OpenCV>
- [2] MediaWiki.*Ajedrez* [En línea]. Disponible en: <https://es.wikipedia.org/wiki/Ajedrez>



Oscar Gabriel Fuentes Lanfur Nacido el 15 de noviembre de 1994. Actualmente ha cerrado los de estudios de Bachiller Industrial y Perito en Electrónica con Orientación en Microprocesadores, cursando el noveno semestre de la carrera Ingeniería en Electrónica en la Universidad de San Carlos de Guatemala. Miembro y voluntario activo IEEE número 94300573 desde principios de 2017, se ha desempeñado en la Rama Estudiantil de la Universidad de San Carlos. Ha participado en competencias de electrónica tanto a nivel nacional como internacional.



Rodrigo Otoniel Méndez Canú. Nacido el 14 de septiembre de 1995. Bachiller en Ciencias y Letras, y actualmente cursando en décimo semestre de ingeniería en la Universidad de San Carlos de Guatemala. Miembro y voluntario activo IEEE número 94976087 desde principios de 2017. Ha participado en competencias de electrónica a nivel nacional e internacional. Actualmente trabaja en el campo de las telecomunicaciones, en telefonía IP y redes locales.



Juan Diego Arias Rossa. Nacido el 6 de octubre de 1995. Estudiante de ingeniería electrónica en la Universidad de San Carlos de Guatemala. Miembro IEEE número 94300523, desde principios de 2017, perteneciente a la Rama Estudiantil de la Universidad de San Carlos. Ha participado en competencias en el ámbito de electrónica tanto a nivel nacional como internacional. Los temas hacia los que se inclina su interés son robótica, comunicaciones, tecnología humanitaria, entre otros.

VI. ANEXOS

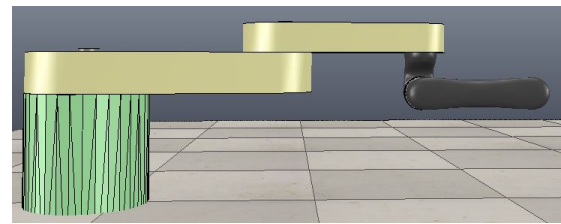


Figura 4: Ilustración de la simulación del brazo SCARA

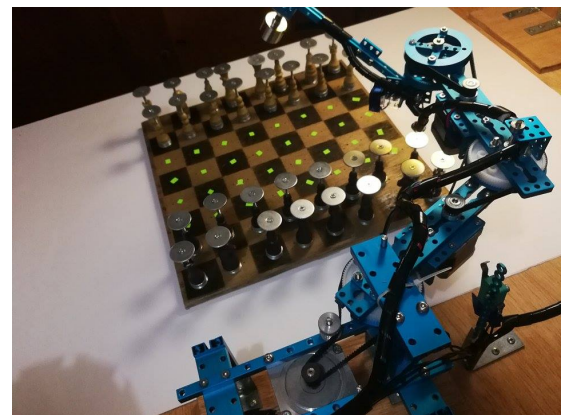


Figura 5: Brazo robótico

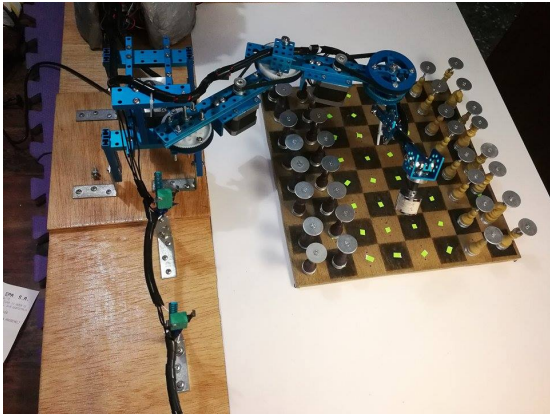


Figura 6: Brazo robótico

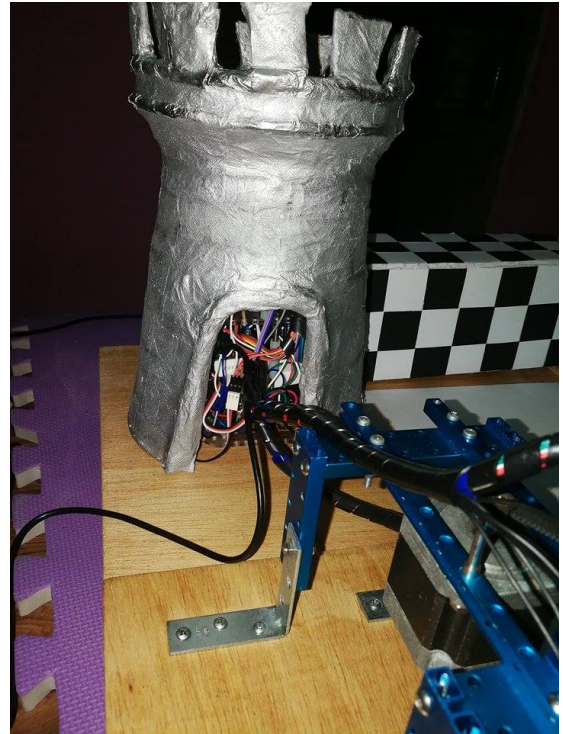


Figura 9: Torre donde se almacena en su interior los circuitos de potencia para mover los motores paso a paso

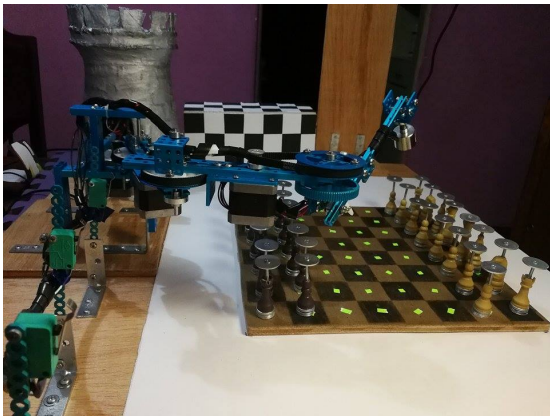


Figura 7: Fuentes de alimentación del brazo Robotico

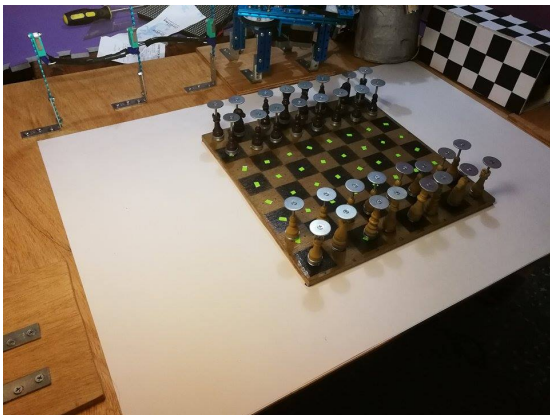


Figura 8: Tablero de Juego