

## **Prototipo de tableros electrónicos de ajedrez**

**Rubén Serrano López**

**Tutor: Hermenegildo Gil Gómez**

Trabajo Fin de Grado presentado en la Escuela Técnica Superior de Ingenieros de Telecomunicación de la Universitat Politècnica de València, para la obtención del Título de Graduado en Ingeniería de Tecnologías y Servicios de Telecomunicación

Curso 2015-16

Valencia, 5 de noviembre de 2016

## Resumen

En el mundo del ajedrez, los tableros electrónicos tipo DGT© (Digital Game Technology) son una gran opción para monitorizar y retransmitir partidas, aunque su elevado precio los hace inaccesibles a los pequeños clubes de jugadores con escasos recursos. Este trabajo de fin de grado, pretende ofrecer una solución igual de funcional y más económica. Hemos desarrollado un tablero, que intenta mantener una estética tradicional, y a su vez, ofrece estas funciones sin alcanzar un precio excesivamente caro, alrededor de los 100€. Para ello hemos usado unos switches que junto a un microcontrolador envían los movimientos efectuados en el tablero de juego a un ordenador al que se encuentran conectados mediante USB. El usuario podrá, ejecutando un software, usar estos datos para retransmitir la partida o guardarla a posteriori en un formato reducido.

## Abstract

In chess world, electronic boards as DGT© (digital game technology) type are a great option for supervising and retransmitting games, although their high price is inaccessible to small clubs of players with limited resources. This end-of-degree project aims to offer an equally functional and more economical solution. We have developed a board, which tries to maintain a traditional aesthetic, and in turn, offer these functions without arriving at an excessively expensive price, around 100 €. For this we have used switches that together with a microcontroller sends the movements made on the game board to a computer connected via USB. The user can, executing a software, use this data to retransmit the game or to save it later in a reduced format.

## Resum

En el món dels escacs, els taulers electrònics tipus DGT © (tecnologia de joc digital) són una gran opció per supervisar i retransmetre partides, encara que el seu preu alt els fa inacessibles als petits clubs de jugadors amb escassos recursos. Aquest treball de fi de grau, pretén oferir una solució igual de funcional i més econòmica. Hem desenvolupat un tauler, que intenta mantenir una estètica tradicional, i al seu torn, oferir aquestes funcions sense arribar a un preu excessivament car, al voltant dels 100 €. Per això hem fet servir uns commutadors que junt amb un microcontrolador envien els moviments realitzats en el tauler de joc a un ordinador al qual es troben connectats per USB. L'usuari pot, executant un programari, utilitzar aquestes dades per retransmetre la partida o guardar-la a posteriori en un format reduït.

## Contenido

1.	Introducción .....	1
1.1.	Justificación y motivación .....	1
1.2.	Objetivos .....	2
2.	Marco teórico.....	3
2.1.	Estado del arte .....	3
2.1.1.	Tablero DGT.....	3
2.2.	Métodos de detección .....	4
2.2.1.	RFID .....	4
2.2.2.	Sensores Hall .....	5
2.2.3.	Galga extensiométrica.....	5
2.2.4.	Fotorresistor.....	6
2.2.5.	Interruptores Reed.....	7
2.3.	Microcontroladores.....	8
2.4.	Proyectos similares .....	9
3.	Aplicación Práctica .....	11
3.1.	Soluciones .....	11
3.1.1.	RFID .....	11
3.1.2.	Sensores Hall .....	12
3.1.3.	Galgas Extensiométricas.....	12
3.1.4.	Fotorresistores .....	13
3.1.5.	Interruptores reed.....	13
3.2.	Solución seleccionada .....	14
3.2.1.	Diseño del hardware .....	14
3.2.2.	Diseño del Software .....	19
4.	Conclusión .....	23
4.1.	Objetivos .....	23
4.2.	Limitaciones .....	24
4.3.	Futuras mejoras.....	24
5.	Bibliografía .....	26

# 1. Introducción

## 1.1. Justificación y motivación

El trabajo fin de grado (TFG para próximas referencias) nace de la voluntad del club de ajedrez de Quart de Poblet de perfeccionar su técnica de juego. En este afán de mejora querían analizar las diferentes partidas que tenían lugar en su club, y para poder hacerlo, comenzaron a investigar cuál era la mejor fórmula para registrar sus partidas.

La primera opción que pensaron fue la introducción de una cámara de video encima del tablero. Esta opción tenía una gran desventaja: implicaba la construcción de una estructura en forma de brazo que invadiría la mesa de juego. Además, para la creación de la misma, se necesitaría realizar un diseño personalizado que conllevaría un coste adicional, al que tendríamos que añadir la cámara de video, aunque las hay muy baratas hoy en día. Por otro lado, los ficheros de video ocupan mucho espacio de almacenaje, por lo que además de la inversión en cámaras y el armazón para sostenerlas habría que comprar unidades de almacenamiento adicionales. Dejando a un lado la inversión inicial, esta tampoco era una solución práctica pues las piezas de un tablero, vistas desde su posición cenital, no son fácilmente reconocibles.

Tras analizar esta opción, se fijaron en los programas de ajedrez para PCs. A diferencia de las cámaras de video, estos programas de ajedrez son capaces de almacenar las partidas en pequeños archivos que permiten su reproducción a posteriori. Trasladar una partida jugada en un tablero convencional a uno de estos programas, supondría la digitalización del juego. Así podríamos monitorizar y revisar las partidas de torneo y entrenamiento en cualquier momento y lugar, e incluso la emisión en streaming del juego. Pero, ¿Por qué no comprar un tablero que realice por sí mismo esta digitalización como lo hacen los DGT(c)?, La respuesta está en el precio: estos tableros tienen un precio medio de 600€, siendo el más barato de 360€, lo que los pone fuera del alcance de un club de jugadores con escasos recursos. Además, hay que tener en cuenta que no solo sería necesario uno de estos tableros, ya que los torneos se juegan en muchas mesas en paralelo. Y es en ese punto en el que realmente nace la propuesta de TFG: tratar de dar respuesta las necesidades del club de ajedrez de Quart de Poblet sin salirnos de su ajustado presupuesto.

Escogí este proyecto pues veía la posibilidad realizar un trabajo integrado a nivel de hardware y software, dos componentes que se destacan claramente en las especialidades de electrónica y telemática respectivamente. Por este motivo me pareció un buen proyecto para demostrar aptitudes adquiridas durante mi educación en la escuela. Además de ser una oportunidad para aplicar lo aprendido y manipular y testear circuitos de forma autónoma; era la ocasión idónea para realizar un software adaptado a un hardware diseñado por mi mismo.

Para realizar este trabajo, implementaremos una red de sensores que detectarán los movimientos, para lo que nos valdremos de los conocimientos adquiridos en la especialidad de electrónica; por otro lado, diseñaremos un software de comunicación con el ordenador, para lo que nos valdremos de las nociones aprendidas de Telemática. Si observamos con detenimiento el currículum de ambas especialidades, nos daremos cuenta que ambas se valen de distintos lenguajes de programación. En electrónica se utilizan aquellos orientados a hardware o simulación del mismo, tales como Verilog y Pspice, mientras que en Telemática usan otros lenguajes como, Java y C. A pesar de ello, en la asignatura Sistemas de microcontroladores, aprendimos a programar hardware, una placa Arduino, mediante lenguaje C.

## 1.2. Objetivos

El primer objetivo del trabajo y el más importante es crear un tablero de ajedrez digital económico. Como ya hemos descrito, existen tableros digitales que nos permiten digitalizar las partidas, pero son extremadamente caros. Este proyecto pretende ser un prototipo similar sin agredir seriamente la estética de un tablero convencional ni sobrepasar el presupuesto, cubra la necesidad más básica del usuario: la virtualización de las partidas.

Con este fin modificaremos un tablero común, introduciéndole un hardware capaz de captar el movimiento de las piezas. Además, diseñaremos un software capaz de procesar los movimientos de las fichas y transmitirlos a otro software alojado en un ordenador. Todo el hardware y el software desarrollado para la producción del proyecto se describe con detalle más adelante.

Es propósito también de este TFG dotar al diseño de una estética pulida que facilite el mantenimiento y posibilite, en caso de haberla, una ampliación de prestaciones. Este acabado perfeccionado, será también una prioridad en el software puesto que una programación ordenada facilitará la aportación de nuevas funcionalidades, así como la mejora de las mismas, una vez que el proyecto esté terminado.

## 2. Marco teórico

En un tablero digital encontraremos dos partes indispensables, el método de detección, y el microcontrolador.

El método de detección se encarga de determinar si una pieza se encuentra o no en una casilla. Los métodos de detección pueden ser interruptores mecánicos o magnéticos, sensores resistivos, inductivos, capacitivos o de efecto hall e incluso circuitos RFID.

Para poder procesar la información obtenida de estos elementos es necesaria la implementación de un microcontrolador que sea capaz de interactuar con ellos. Por ello, el software que se diseñará deberá ser capaz de enviar y recibir estímulos a/de estos elementos, para en consecuencia trasladar la información de los movimientos al ordenador.

### 2.1. Estado del arte

#### 2.1.1. Tablero DGT

Antes de embarcarnos en el diseño de un prototipo propio, analizamos las características de los tableros que hallamos en el mercado. ¿Cómo funcionan los tableros DGT? Estos tableros constan de dos partes, una es la superficie de juego, en la que encontramos un circuito impreso con dos matrices: una matriz de sensores y otra matriz de emisores. La otra la conforman las piezas que interactúan con las matrices de la superficie.

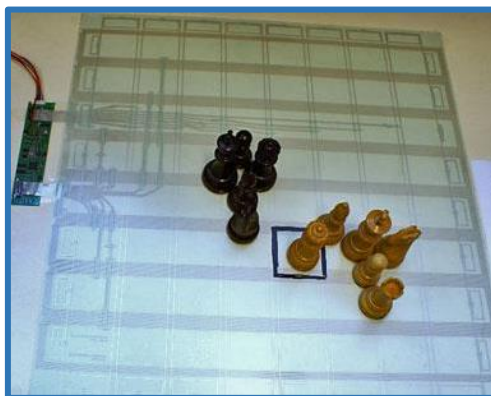


Imagen 1 Interior Tablero DGT [7]

Cada una de las piezas lleva dentro un núcleo de ferrita con una bobina que lo rodea. La bobina de cada pieza tiene una longitud diferente, esto las dotará de distintas frecuencias de resonancia, desde los 90 hasta los 350 KHz. Aprovecharemos estas diferentes frecuencias para codificar cada una de las piezas y clasificarlas por tipo y color, lo que hace que este tipo de tableros tengan una gran fiabilidad.

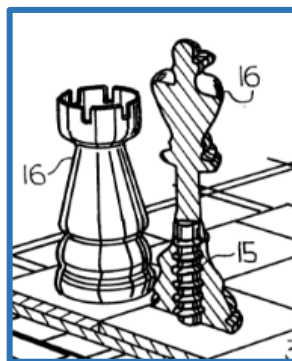


Imagen 2 Ilustración Pieza DGT [9]

Las matrices están formadas por espiras planas. La matriz emisora excita la pieza, que contiene un circuito LC formado por la bobina y el núcleo de ferrita. Al ser excitado, el circuito pasivo resonará generando corrientes parásitas en el núcleo de ferrita. Estas son las conocidas corrientes de Foucault, que perturban la frecuencia de resonancia de la bobina. La matriz sensora mide la interferencia creada en la resonancia, con este valor será posible identificar las piezas. Este tipo de detección se conoce como inductivo, en referencia al uso de bobinas.

Para evitar las posibles interferencias de las piezas vecinas, es necesario excitar solo uno de los cuadros cada vez. Mediante un multiplexor de señales analógico es posible excitar y sensor cuadro por cuadro. Una vez leído el valor resultante de la excitación de la casilla, este se analiza y se decide si hay o no pieza en esa casilla, y qué pieza es. Esta información se envía por el puerto serie (o bluetooth) al ordenador.

## 2.2. Métodos de detección

A parte del sistema utilizado por el tablero DGT, existen otros métodos para saber si una ficha se encuentra en el tablero de juego y qué ficha es. Antes de decidir qué método es el mejor para nuestro proyecto, revisaremos las características de algunos de estos.

### 2.2.1. RFID

Los tableros DGT usan una tecnología similar a la RFID (RadioFrecuency IDentification). Los primeros usan dos matrices: una que excita y otra que hace de sensor, mientras que los sistemas RFID emplean una bobina que hace las veces de excitador y detector, alternando de una función a otra.

Los sistemas RFID están formados por una parte activa y otra pasiva. La bobina antes mencionada será la parte activa del sistema, e interactuará con una parte pasiva que hará de emisor. Este emisor sólo enviará información al ser excitado por la bobina, es por ello que lo denominamos parte pasiva. Si a este sistema le añadimos un sensor de proximidad en la parte activa podemos reducir el consumo de energía considerablemente.

Comunicaciones como esta se usan día a día por la población, en especial en los transportes públicos como el metro o el autobús, ya que las tarjetas de viaje cuentan con esta tecnología. Al validar un viaje, el usuario acerca la tarjeta al lector, que se encuentra emitiendo y leyendo perpetuamente por si algún usuario acerca la tarjeta. Cuando el emisor por fin emite sobre una tarjeta, esta se alimenta de la emisión y emite entonces contra el lector, transmitiéndole la información que hay en ella.

Además, los elementos pasivos de esta comunicación pueden ser especialmente pequeños y almacenar información suficiente, como en nuestro caso el color y el tipo de ficha.

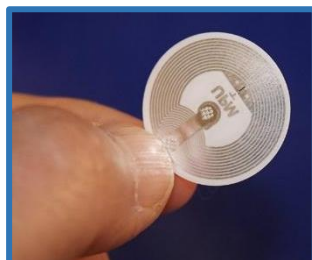


Imagen 3 Pegatina pasiva RFID [3]

Por otro lado, el elemento activo de este tipo de comunicación también se puede encontrar de un tamaño reducido, de hasta 4x4 cm<sup>2</sup>, siendo ideal para nuestro tablero.

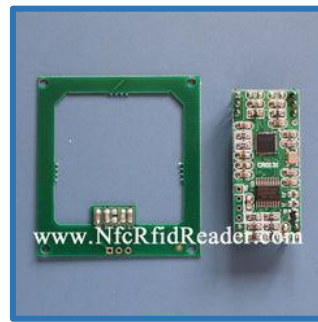


Imagen 4 Lector RFID 4x4cm [15]

### 2.2.2. Sensores Hall

Otro mecanismo de detección serían los sensores de efecto Hall. Estos sensores pueden detectar la presencia de un imán, o el movimiento de uno. Su mecanismo es sencillo, se induce una corriente sobre un metal plano en una dirección, y se mide la tensión en bornes de la dirección perpendicular a la corriente. Cuando un imán se acerca perpendicular al plano, los electrones de este se desplazan, generando así una tensión perpendicular a la corriente. A continuación, adjunto una imagen aclaratoria.

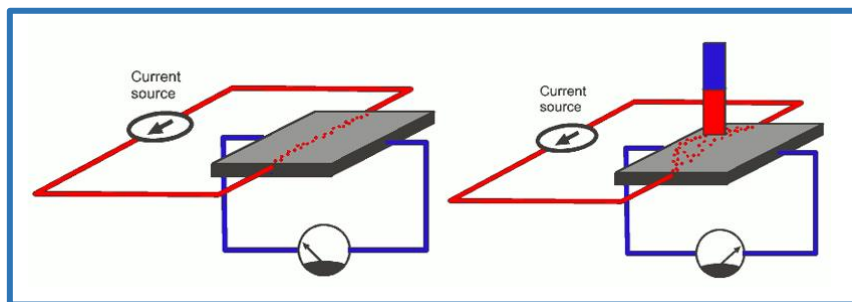


Imagen 5 Ejemplo funcionamiento sensor Hall

Este tipo de sensor es comúnmente utilizado en los frenos de los coches, motores, sensores de proximidad, discos duros, entre otros. En nuestro caso se utilizaría como detector de un imán, que se encontraría insertado en la pieza de ajedrez.

### 2.2.3. Galga extensiométrica

Por otro lado, encontramos un sensor en el que no intervienen campos magnéticos, sino presiones, las galgas extensiométricas. Son en esencia impedancias que se ven modificadas al deformarse, y tienen una gran sensibilidad. El diseño puede variar dependiendo de la dirección en la que queramos medir la presión ejercida, así como su posicionamiento en el material a medir. Son del tamaño de una uña aproximadamente.

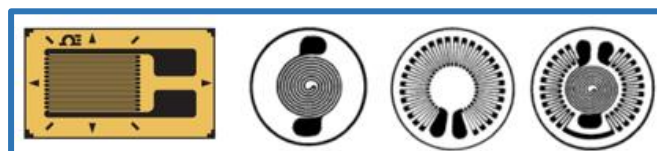


Imagen 6 Ejemplos de Galga Extensiométrica



Para la medida de las variaciones de impedancia, las galgas se suelen montar en sistemas de puente de Wheatstone, ya que las variaciones de impedancia al igual que las impedancias son muy pequeñas, del orden de los 100-300 ohm para esos tamaños. El puente de Wheatstone se puede configurar de muchas maneras con tal de minimizar los desfases por temperatura o desgaste. Este es en sí dos divisores de tensión. Midiendo en bornes del puente (V) podemos saber cómo se ha modificado la resistencia.

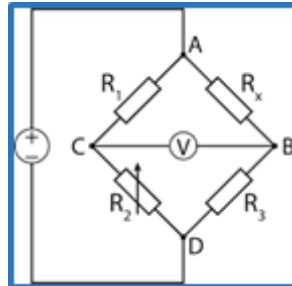


Imagen 7 Puente de Wheatstone general

Estos sensores se utilizan en las básculas domésticas, acelerómetros, deformación de estructuras. Para el caso particular del tablero se podría saber si una pieza se ha colocado en una casilla, ya que ejercería presión sobre la galga modificándose la impedancia.

#### 2.2.4. Fotorresistor

Al igual que las galgas, los fotorresistores son también impedancias que, en este caso, varían en función de la luz que incide en ellas. Los tamaños son equiparables, pero no así las resistencias, que varían de 100Ω con luz al 1MΩ en oscuridad.

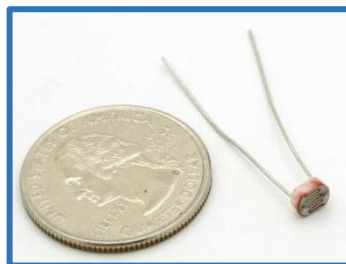


Imagen 8 Fotorresistor LDR [2]

Al tener dos resistencias tan dispares (100Ω y 1MΩ), se puede implementar un detector muy sencillo mediante un divisor de tensión, que nos dé una tensión aproximadamente de  $V/2(V)$  o cercana a 0(V) seleccionando una R de aproximadamente 1MΩ, aunque puede ser de hasta 50KΩ sin importar demasiado, lo que interesa es que sea grande para que cuando LDR sea pequeña por la incidencia de luz, la  $V_o$  sea cercana a 0(V).

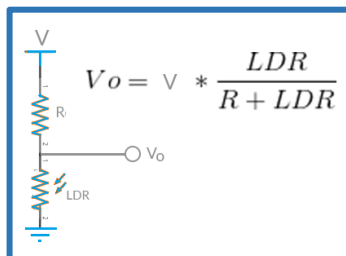


Imagen 9 Divisor de Tensión para el LDR

Tras revisar los diferentes sensores que podrían ser de utilidad en el proyecto, ahora es necesario saber cómo procesar la información que estos nos aportan. En mi caso opté por buscar un microcontrolador de tamaño reducido, ya que la intención es que todo el sistema pueda integrarse dentro del propio tablero, minimizando el impacto exterior.

### 2.2.5. Interruptores Reed



Imagen 10 Switches Reed [1]

Se trata de unos interruptores que se estimulan mediante un campo magnético. Los interruptores pueden ser de distintas longitudes y su respuesta dependerá tanto de la longitud de estos como de la posición y la polarización del imán que se le aproxime. Esto significa que pueden tener muchos comportamientos distintos, por ello, es necesario saber cuál nos sería de ayuda para nuestro tablero. A continuación se muestra dicho comportamiento.

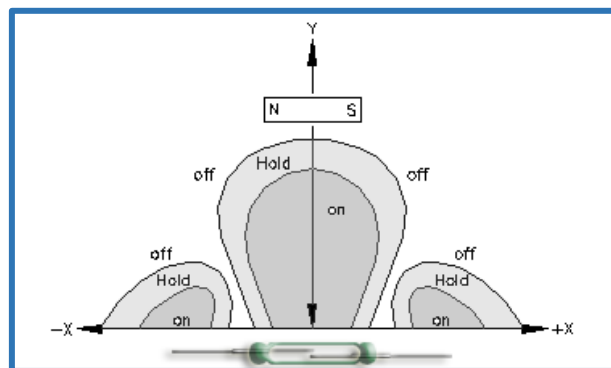


Imagen 11 Diagrama de comportamiento Switches Reed 1 [27]

Cuando el imán se coloca encima del interruptor este se enciende. Si nos fijamos esto ocurre solo cuando el imán está polarizado en paralelo al interruptor. Pero al utilizarse imanes de tipo botón esta no es la polarización utilizada, obteniendo un comportamiento como el que se ve a continuación:

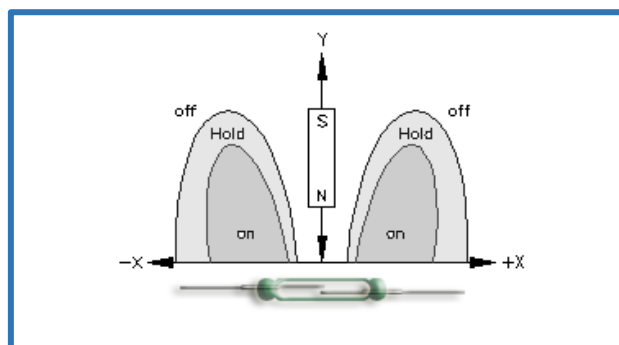


Imagen 12 Diagrama de comportamiento Switches Reed 2 [27]

Como se puede ver, el comportamiento ahora es bien distinto. El lóbulo de activación central ha desaparecido, generándose dos lóbulos en los vértices, que, además, son de igual rango. Estos lóbulos están influenciados a su vez por la fuerza del imán, cuanto más potente es el imán más grande se hacen los lóbulos.

### 2.3. Microcontroladores

Un microcontrolador es un circuito integrado programable, que puede ejecutar instrucciones almacenadas en su memoria. Al ser programable se puede diseñar un modelo general, fácil de fabricar en masa para abaratar su coste. Esto hace que sea una opción muy viable para el diseño de productos que necesitan un software concreto. Se ahorra en la inversión de un procesador personalizado, y se invierte en la creación de un software que haga que el microcontrolador se comporte como se desee.

Los microcontroladores están formados por un microprocesador, una o más memorias e interfaces de entrada y salida. Estos interfaces de entrada y salida pueden ser desde pines hasta puertos USB o interfaces bluetooth, por ejemplo.

Existen infinidad de microcontroladores en la actualidad, por ello, el primer criterio de filtro que apliqué fue conocer un poco el microcontrolador. Es por este motivo que opté por buscar uno de la marca Atmel. Estos microcontroladores son los integrados en las placas Arduino que utilizamos en la asignatura de aplicaciones de microcontroladores del grado. Buscando información sobre estos microcontroladores, vi que existían integrados en placas realmente pequeñas para multitud de usos, que además contenían una entrada/salida USB. La mejor forma para encontrarlos fue buscando en inglés en *google*: “usb development board”, la búsqueda arroja a primera vista imágenes de pequeñas pcb (printable circuit board) con un integrado.



Imagen 13 Ejemplos de Microcontroladores con USB integrado [26]

Encontré en uno de los primeros enlaces una página que vendía distintos modelos de microcontroladores, cada uno de un tamaño, y la cual contenía tutoriales, consejos, e incluso proyectos a modo de ejemplo. Así pues, decidí que era una buena base para programar, ya que tenía ayuda del propio vendedor. Los modelos se denominan Teensy, de los muchos que habían escogí el modelo 2.0, al ser el más económico y no necesitar más de 25 pines. En un cálculo rápido sabemos que la matriz será de 8x8 y con algún botón extra o led, usaremos aproximadamente 20-22 pines.

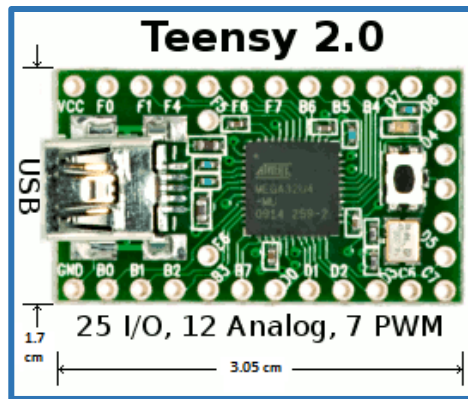


Imagen 14 Teensy 2.0 [23]

Tras indagar un poco vi que el microcontrolador elegido no era Arduino, aunque se podía usar el propio programa de Arduino mediante un parche que ellos mismos te proporcionaban y enseñaban a instalar. A partir de ahí solo necesitaba aprender a usar los distintos pines como entradas y salidas, por lo que empecé por lo básico, encender un led. Ellos mismos te dan un código base para esto, por lo que fue fácil aprender.

#### 2.4. Proyectos similares

Una vez seleccionado el microcontrolador, tenía que aprender a manejar estímulos de entrada y salida. Comencé por buscar ejemplos de su uso en teclados, botones, leds y otros. De este modo encontré tutoriales bastante completos sobre programación Arduino, que, al compartir microcontrolador, eran igual de útiles.

Comencé buscando guías básicas, y encontré una muy completa [12], con la que empecé a programar cosas básicas como sacar un 1 lógico o un 0, a través del comando `digitalWrite(pin, high/low)`, comando que permite poner un pin a 1 o 0. Más adelante aprendí a leerlos [`digitalRead(pin)`] también, y cómo determinar si un pin era entrada o salida.

Encontré en la web del distribuidor un proyecto muy sencillo de un botón [20]. El código de dicho proyecto usaba la librería con la que yo había empezado a programar, pues incluía la función `digitalRead(pin)`, función usada para leer la entrada de un pin. Este código simula la función de la barra espaciadora de un teclado, así pues, al pulsar el botón se envía un espacio en blanco a través del USB.

Al fijarme en el código, vi que el espacio en blanco no se mandaba como un carácter, sino, como una cadena de caracteres, por lo que podría aprovechar dicha función para mandar las posiciones de las piezas al software del ordenador con una sola instrucción.



Imagen 15 Botón para Teensy [20]

```

void setup() {
  Serial.begin(9600);
  pinMode(10, INPUT_PULLUP);
  delay(500);
}
void loop() {
  if (digitalRead(10) == HIGH) {
    delay(10);
  } else {
    Keyboard.print(" "); // we print a space
    delay(1000);
  }
  delay(10);
}

```

*Código 1 Botón Teensy que envía un espacio en blanco [20]*

También me sorprendí al ver que se había realizado un ajedrez mediante pulsadores. En el blog del autor [6] se ve la matriz trasera del tablero, y un video que muestra cómo funciona su tablero.



*Imagen 16 Perspectiva del Tablero Sish [6]*

Navegando un poco por el blog, encontré más gente que se había aventurado a realizar tableros con el mismo método o similares. Y encontré el que me aportó la idea final: el uso de interruptores reed. Este blog se llama: Los Proyectos de Berger – DIY [17], y su diseño se denomina SoluChess, además proporciona instrucciones detalladas, así como su propio software compilado, para las personas que no tienen conocimientos de programación o experiencia con microcontroladores.



*Imagen 17 Perspectiva del tablero de Berger [17]*

### 3. Aplicación Práctica

#### 3.1. Soluciones

Una vez decidido el microcontrolador que usaremos, queda decidir qué red de sensores se iba a utilizar, por lo que se analizan las distintas opciones descritas anteriormente en el apartado de sensores.

##### 3.1.1. RFID

Empezaremos con los RFID. Estos sensores nos permiten almacenar bastante información en los elementos pasivos. El bajo coste de los elementos pasivos, 0.20€/unidad, supone una gran ventaja, y su capacidad para almacenar información sería de gran ayuda si el software del microcontrolador se viera limitado en capacidad. Otra gran ventaja es que el elemento activo de este par de comunicación puede ser del tamaño perfecto para nuestras casillas, denominadas escaques en la jerga del ajedrez, de hasta 4x4 cm<sup>2</sup>.

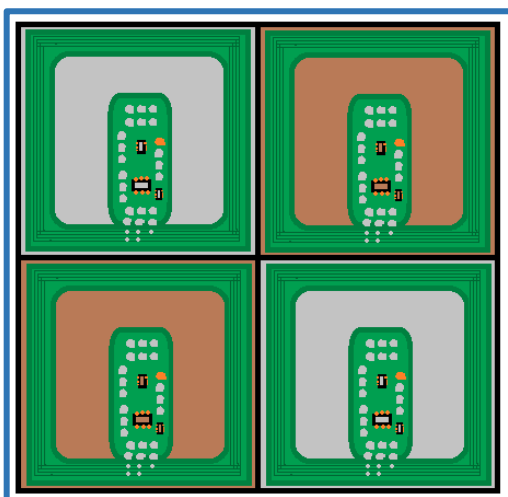


Imagen 18 RFID bajo el Escaque



Imagen 19 Pegatina NFC Bajo la Pieza

El primer inconveniente de este tipo de sensores sería el aumento de consumo energético. Sería necesario hacer uso de una matriz de alimentación, ya que los 64 módulos (uno por cada casilla) necesitarían alimentación. Por otra parte, los módulos tendrían que emitir y leer secuencialmente, ya que sino se pueden generar interferencias que afecten a la lectura de las celdas vecinas. Al ser muy complicada su sincronización y su radio de acción no limitarse solo a su escaque inmediatamente superior, sería necesario añadir un multiplexador y un retardo entre lecturas. Esto sumado a su precio del componente activo que es de unos 5€ la unidad, la más barata, hacen que fuese una opción a descartar, ya que el precio alcanzaría al de los tableros DGT, 320€.

### 3.1.2. Sensores Hall

Los sensores hall podrían proporcionarnos un comportamiento similar a los DGT. La forma de lograrlo sería colocando imanes en las bases de las piezas. Como los sensores de efecto Hall son capaces de medir el campo magnético, colocando cada imán a una profundidad distinta, dependiendo del tipo de ficha y color, crearíamos distintas fuerzas magnéticas, detectables por el sensor hall. Todo esto a un coste menor que los DGT ya que cada sensor cuesta 1€ y los imanes usados en las piezas tendrían un valor aproximado de 6€ (6€/50 unidades), por tanto, el coste total sería 70€.

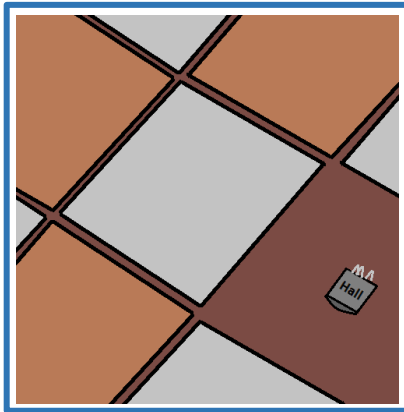


Imagen 20 Sensor Hall bajo el Escaque

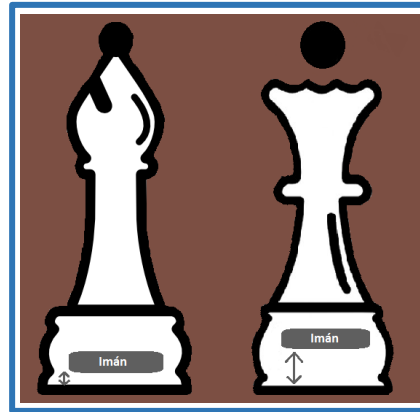


Imagen 21 Piezas con Imán

En mi caso, esta solución se me ocurrió a posteriori, por lo que queda propuesta como posible mejora de precisión del tablero, con su consecuente aumento de precio. Más detalle de los aspectos que mejorarían con este tipo de sensores al final del proyecto.

### 3.1.3. Galgas Extensiométricas

A priori la solución más sencilla que se me ocurrió, muy similar a la del proyecto Sish con pulsadores, pero sin botones. Además, el hecho de que fuesen galgas añade la función de identificación de las piezas por peso.

Los inconvenientes de este método aparecen con el uso, ya que la pérdida de peso por caída o desgaste de las piezas, el mantenimiento del fieltro de la base, etc. causarían una variación en el peso, con lo que podrían producirse errores de identificación.

Por otro lado, las galgas deberían situarse sobre la superficie del tablero, quedando expuestas. Para solucionar este problema se podrían crear escaques flotantes, como si de mini basculas se tratasen, y colocar las galgas debajo de estos, pero cuantas más piezas móviles en un diseño, más fallos pueden aparecer.

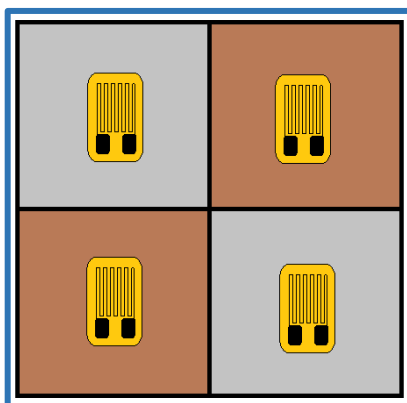


Imagen 22 Ejemplo Galgas Superficie

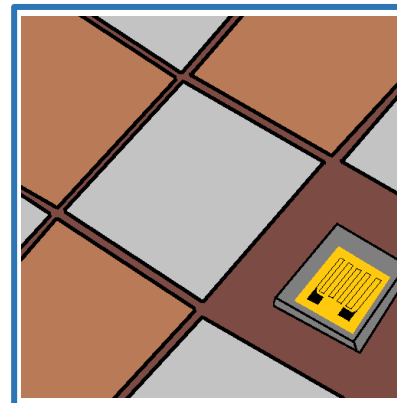


Imagen 23 Ejemplo Galgas bajo Escaques

Aunque sin ninguna duda el factor predominante del rechazo absoluto a esta solución fue su desmesurado precio, 7€ la unidad más barata, alcanzando un precio de 448€ solo los sensores.

#### 3.1.4. Fotorresistores

La mejor cualidad de estos sensores es su precio, 0.08€ la unidad (en packs de 100). Además, pueden ser usados como conmutadores de 1 o 0 lógicos debido a sus cambios drásticos de impedancia.

Al igual que las galgas, los fotorresistores deben colocarse en la superficie del tablero. Además, estos no pueden ser escondidos bajo el tablero, por lo que no solo quedan a la vista y rompen la visión de un tablero convencional, sino que están expuestos a la suciedad y el ambiente.

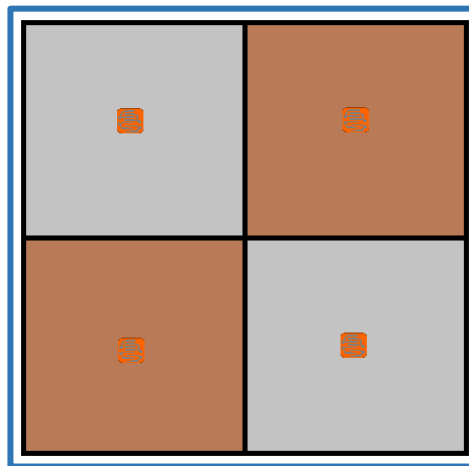


Imagen 24 LSR superficie

#### 3.1.5. Interruptores reed

Este tipo de interruptores son especialmente baratos, desde los 30 céntimos de euro, aunque son un poco más caros que los fotorresistores. Tienen un precio total aproximado de 23€, al que sumamos el valor de los imanes, que es de aproximadamente 6€/50 unidades, unos 29€ en total.

Para usar los interruptores reed en nuestro proyecto, nos serviremos del comportamiento que tienen estos interruptores frente a un imán plano de tipo botón: sabemos que con este tipo de imán el lóbulo de activación central desaparece y se generan dos lóbulos de igual rango en los vértices. Así pues, colocaremos los interruptores de forma que uno de los lóbulos se sitúe en el centro de la casilla, haciendo que el lóbulo opuesto quede colocado en una zona menos apreciable respecto al centro de otra casilla.

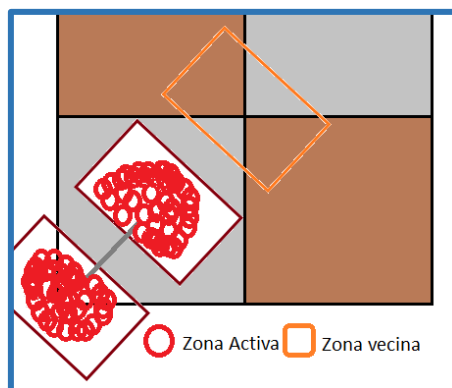


Imagen 25 Interruptor Reed y sus Lóbulos



Si bien es cierto que con este tipo de sensores no podemos saber qué ficha se encuentra en cada casilla, es un método barato y sencillo de saber si hay o no una ficha en una casilla.

Un aspecto negativo es, como se puede ver, que el lóbulo no utilizado puede ser estimulado por una pieza en otro escaque, creando falsas posiciones.

### 3.2. Solución seleccionada

Habiendo analizado todas las opciones con sus pros y contras, la solución seleccionada ha sido esta última, los interruptores reed, debido a su sencillez y su reducido precio.

El proyecto se divide en dos secciones, el hardware y el software. El primero conlleva el diseño del tablero, testeo de los componentes y su comportamiento, montaje de la matriz de interruptores, y su conexión al microcontrolador. En segundo lugar, el software, que deberá ser capaz de estimular las entradas de la matriz, leer las respuestas, analizarlas y enviar el comando del movimiento al software de juego del ordenador.

#### 3.2.1. Diseño del hardware

Como hemos visto, los interruptores reed tienen distintos comportamientos en función de su longitud y la fuerza del imán. Para estudiar el comportamiento de los interruptores reed y los imanes en el contexto de un escaque adquirí dos imanes distintos y un interruptor de 1.5cm. Elegí este tamaño del interruptor porque encaja perfectamente en la diagonal de los escaques seleccionados, puesto que el tamaño medio de un escaque de competición es de 4-4.5cm X 4-4.5cm. El proyecto se implementará en uno de 4.5x4.5 cm

Sabiendo que los lóbulos de activación del interruptor son simétricos, y el centro es no activo, el interruptor debe colocarse en un lateral: de forma que uno de los lóbulos de activación quede centrado en el escaque. Para realizar las pruebas elegí el tamaño de escaque más pequeño: de 4x4 cm. El objetivo de esta elección fue ponernos en la situación más desfavorable, aquella en la que, al tener los escaques vecinos más cerca los unos de los otros, se pueden crear más interferencias. Estas interferencias se deben a la colocación del interruptor reed, ya que al centrar uno de sus lóbulos de activación en el escaque, el otro queda desplazado pudiendo invadir los contiguos.

Antes de comenzar a estudiar las interacciones entre los imanes que compré y los interruptores reed debemos saber que los imanes siguen una nomenclatura estandarizada. Esta nomenclatura permite conocer las propiedades del imán mediante el uso de una o más letras y un número (Xyy-M), donde "X" y "M" detallan el material (en referencia a la temperatura soportada), e "yy" es la fuerza del imán.

Tras probar los dos imanes que había adquirido: imán de tipo N34-45 y otro N52, vi que no había una gran diferencia de respuesta. El motivo fue que a pesar de comprar dos imanes distintos no había tenido en cuenta que las dimensiones del imán afectan al campo generado. Resultó que el N34-46 era de 4 mm de grosor y 12 mm de diámetro, mientras que el N52 era de 2mm de grosor y 12 mm de diámetro.

Sabiendo que la posición del interruptor reed debe ser lateral, de forma que uno de los lóbulos quede centrado y el otro interactúe lo menos posible con los escaques contiguos, valoramos cuál es la posición idónea del interruptor. Como observamos en la imagen, colocándolo en la diagonal conseguimos marginar el lóbulo no deseado y evitar falsos positivos debidos a la interacción con otros escaques.

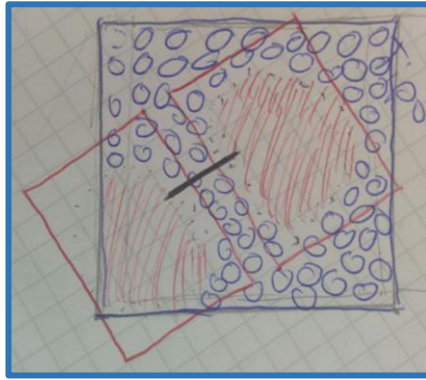


Imagen 26 Lóbulos en papel

Para comprobar la posición idónea y la reacción ante distintas posiciones de las piezas utilice un trozo de madera, que simulaba unos pocos escaques del tablero 4x4cm<sup>2</sup>, con un grosor de 0.5 cm.

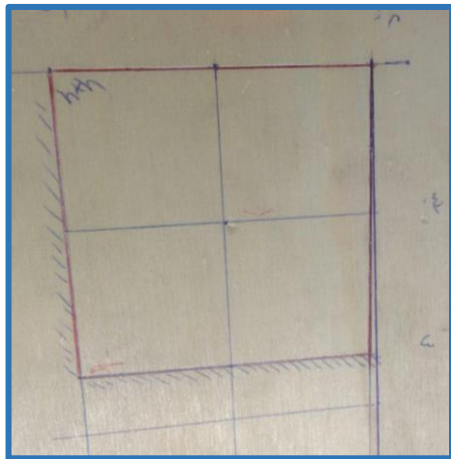


Imagen 27 Escaques test(top)

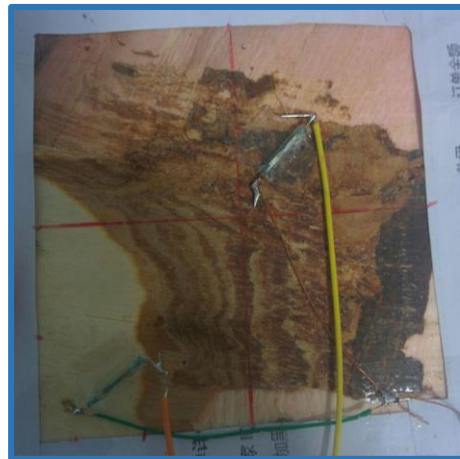


Imagen 28 Escaques test(bottom)

Las imágenes anteriores se encuentran rotadas de forma que los escaques coinciden en posición. Los interruptores se han colocado en escaques diagonalmente adyacentes, ya que es la peor condición, al tener el lóbulo de activación en esa dirección. De este modo es sencillo entender lo que ocurre al posicionar el imán en las distintas posiciones:



Imagen 29 Test Activación(Pos-1)



Imagen 30 Test Activación(Pos-2)

El circuito de test es muy simple, se trata de una pila de 3.3 Voltios en serie con un paralelo de dos ramas. Cada una de estas ramas contiene: un diodo led en serie con una resistencia de  $220\Omega$ , y un interruptor reed.

De este modo al situar el imán sobre el lóbulo del interruptor, este se cierra, cerrando así el circuito y encendiendo el led correspondiente. Como hemos comentado previamente, las pruebas las estamos realizando en la condición más desfavorable, pero, cuando se implemente en el tablero con escaque de  $4'5 \times 4,5$ , el tamaño de los escaques será medio centímetro más grande, por lo que el lóbulo de activación quedará completamente centrado, y el lóbulo no deseado no invadirá a penas los escaques vecinos, de modo que la posibilidad de que se active con una pieza situada en un escaque vecino es ínfima.

Una vez estudiado el posicionamiento de los interruptores, se ha de fresar un tablero convencional de madera, para dejar sitio a la matriz de interruptores. De este modo todo el hardware queda integrado en el tablero.

El tablero tendrá dos botones para controlar en qué lado se encuentran las blancas, y para comenzar una partida nueva. También habrá dos leds, uno a cada lado del tablero para indicar el turno. Por este motivo hay que dejar un hueco para dichos botones y leds, así como para el puerto USB del microcontrolador.



Imagen 31 Tablero perforado(top)



Imagen 32 Tablero perforado(bottom)

El siguiente paso es instalar la matriz junto al microcontrolador, los botones y los leds. Pero antes de comenzar la instalación, revisando bibliografía, leí de la existencia de un fenómeno a tener en cuenta: el "ghost key". Este fenómeno se produce cuando tenemos tres o más interruptores activos formando una "L" en la matriz, lo que provoca un falso positivo. Para esclarecer este caso utilizaré los siguientes esquemáticos:

En esta imagen podemos ver como se excita A, y al no estar activos los interruptores, no se excitan ni la salida 1 ni la 2.

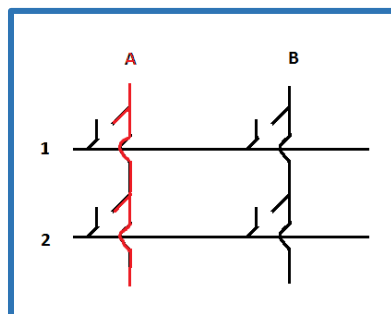


Imagen 33 Ghosting (1/5)

En esta ocasión se excita A y se cierra A1, leyendo así dicho valor.

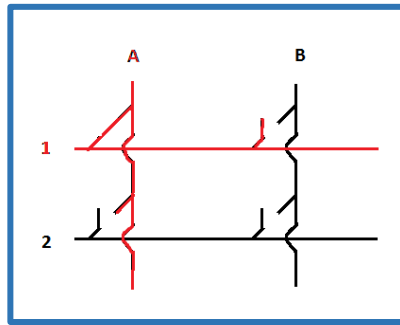


Imagen 34 Ghosting (2/5)

Ahora se estimula B y se cierra B2, obteniendo esa misma salida.

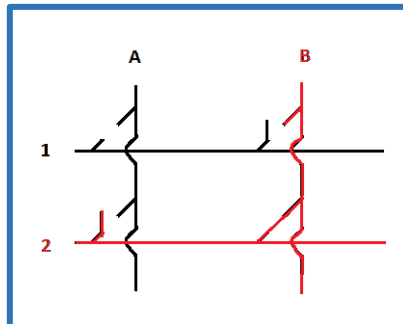


Imagen 35 Ghosting (3/5)

Pero si se excita A para leer A1, cuando B1 y B2 se encuentran a su vez cerrados se crea un camino alternativo entre A y 2, produciendo la lectura A1 y un falso positivo en A2.

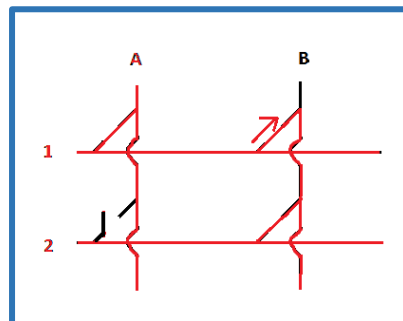


Imagen 36 Ghosting (4/5)

La solución es colocar unos diodos, que impidan el paso de la corriente, impidiendo que se activen los nodos vecinos por error. De este modo la lectura A1 será correcta.

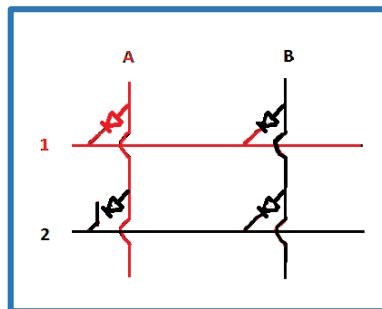


Imagen 37 Ghosting (5/5)

Tras solucionar este problema, y antes de realizar el montaje del hardware, es necesario comprender el funcionamiento del microcontrolador, por lo que el software se diseña previo a este paso. Una vez diseñado el software, se sueldan las entradas y salidas de la matriz a los pines seleccionados, al igual que los pines de control de los botones y los pines de información led.



*Imagen 38 Tablero con matriz*

Una vez finalizado el montaje del tablero hay que pegar los imanes a las piezas. La idea original era perforarlas e introducir el imán en el interior. Pero al quitar el fieltro descubrí que no eran de madera maciza, contenían un peso en el interior impidiendo introducir el imán. La alternativa fue pegar el imán en la superficie y rodearlo con un trozo de fieltro para igualar el nivel de la base y luego cubrirlo con otra capa.



*Imagen 39 Pieza sin fieltro*



*Imagen 40 Pieza con imán*



*Imagen 41 Pieza cubierta*

### 3.2.2. Diseño del Software

En cuanto al software, es necesario tener claro cómo queremos que funcione el tablero, y cómo se va a interactuar con él. Por ello presento un diagrama que define el comportamiento del software por estados:

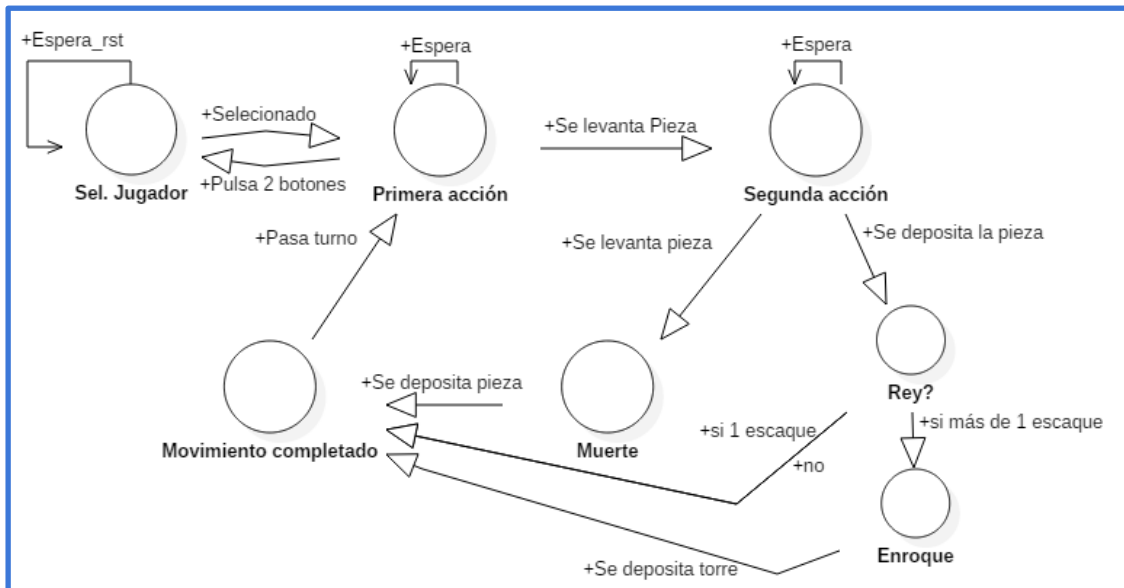


Imagen 42 Diagrama de estados

Como los interruptores reed solo nos informan de si hay o no una pieza sobre el escaque será el software el encargado de averiguar de qué ficha se trata y si es amiga o enemiga. Para ello el código tendrá diferentes variables de tipo “pieza”, y cada variable nos informa del tipo de pieza, el color, el estado (viva o muerta) y la posición que ocupa en el tablero. Esto será útil para determinar cuándo se captura una pieza enemiga o cuando se realiza un movimiento de enroque.

La programación del microcontrolador se realiza presuponiendo que los jugadores jugarán de forma lícita. Con estas bases establecidas procedo a detallar el funcionamiento del microcontrolador.

Lo primero que se plantea cuando se empieza una partida es: qué lado mueve primero. Esto es lo mismo que preguntar: ¿En qué lado se encuentran las blancas? Las blancas además de ser las primeras en mover, también son la base del tablero, esto quiere decir que siempre sitúan sus piezas en las dos primeras filas del tablero (filas 1 y 2). Como este no tiene las coordenadas pintadas en la superficie, las piezas se pueden colocar en cualquier lado, pudiendo no coincidir con el orden en que se han denominado a los sensores. Por este motivo, hay que otorgar al usuario la opción de decirle al tablero en qué lado se encuentran las blancas. Para ello se han colocado dos botones que permiten seleccionar el lado y dos leds que indican que lado se ha seleccionado, a modo de confirmación. Ambos leds se encuentran iluminados hasta que se pulsa uno de los dos botones, entonces queda iluminado el led del jugador que comienza.



Imagen 43 Botones Tablero

La selección de la posición de inicio solo afecta al comportamiento interno del tablero, ya que los programas de ajedrez sitúan las blancas siempre en las dos primeras filas. Lo que se traduce, en nuestro código, en un cambio de turno y de la correspondencia del tablero con los sensores. Esto quiere decir que el tablero ha de tener dos posiciones predeterminadas entre las que elegir. Estas dos posiciones son la plantilla que se sitúa sobre la matriz de sensores, haciendo coincidir el sensor (1,1) con la casilla A1 para una posición o la H8 para la posición opuesta.

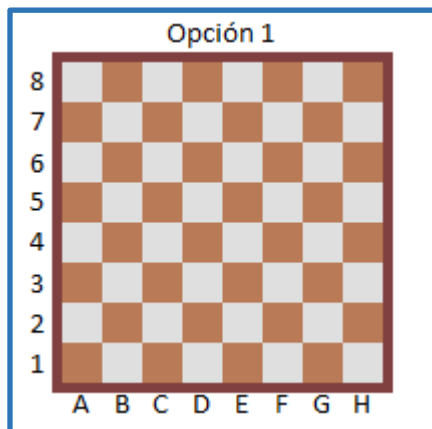


Imagen 44 Posición 1

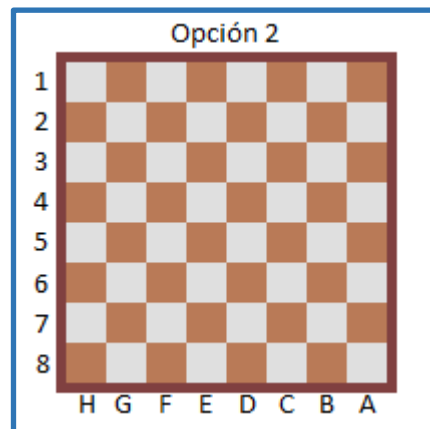


Imagen 45 Posición 2

Al iniciar el tablero este entra en un primer estado: esperar, mediante un bucle *while* y una variable bandera, a que el jugador pulse uno de los dos botones. Antes de proseguir, debemos aclarar que al iniciar el estado la variable bandera se inicializa a 0. Mientras esta sea 0 se ejecutará el bucle *while*. Al pulsar el botón, el tablero le otorga el turno al jugador que lo ha pulsado, selecciona la plantilla de tablero en que se va a jugar, se activa la bandera. Al activarse, adquiere el valor de 1 y el bucle *while* deja de ejecutarse. Con esto se inicia el juego.

Una vez la partida comienza, el tablero entra en un segundo estado. En ese momento el microcontrolador tiene una variable "tablero" de dimensiones 8x8 con la posición inicial de las piezas, que coincide con la posición real de las piezas en el primer instante de la partida. Aquellas casillas que tengan pieza, se les otorga un 0, aquellas que no un 1. Comparando esta información con la obtenida a raíz de recorrer la matriz de interruptores, el microcontrolador puede saber si una pieza ha sido levantada del tablero.

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Imagen 46 Variable "tablero [8][8]"

Antes de continuar, es necesario aclarar que la variable "tablero" no es la encargada de determinar la posición del tablero. Esta selección, que se produce cuando el jugador pulsa el botón indicando dónde se encuentran las blancas, afectará a dos vectores de tipo "char" a los que se recurrirá para enviar los movimientos al ordenador, como veremos en el último estado.

En este momento el microcontrolador espera a que ocurra la primera acción del jugador: mover una pieza. El modo en que el microcontrolador sabe si se ha producido el movimiento es analizando cada uno de los interruptores[i][j] y comparando su correspondencia con la variable "tablero[i][j]". "i" y "j" son las variables numéricas que utilizamos para identificar los diferentes interruptores. Así recorreremos el tablero (variando "i" y "j") hasta que encontremos aquel interruptor que ha variado su posición. Buscaremos sólo un cambio de 0 a 1, puesto que pretendemos averiguar qué ficha se ha levantado del tablero. Por tanto, mediante un bucle *while* analizaremos el valor de cada interruptor[i][j] hasta que encontremos uno que siendo antes 0 (tablero[i][j]==0), ahora no lo sea. Cuando lo encontremos, activaremos la variable bandera.

Es importante aclarar que, debido al método de detección y análisis de los cambios en el tablero, las piezas sólo pueden ser levantadas y depositadas de nuevo sobre el tablero, no arrastradas sobre él. La forma en la que el microcontrolador determina que una pieza ha comenzado el movimiento es mediante la detección de que dicha pieza se ha levantado del tablero y este finaliza al dejarla en otro escaque. Si se desliza una pieza, el tablero interpreta que se ha levantado del primer escaque y depositado sobre el siguiente. Con este movimiento se terminaría el turno del jugador, aunque su intención fuese realizar un movimiento de más de un escaque de distancia.

Cuando la pieza se levanta se guarda el valor de "i" y "j" en dos variables globales "i1" y "j1" (estas variables globales son accesible desde cualquier parte del código) y se pasa al tercer estado. De nuevo se espera con una variable bandera a que ocurra la siguiente acción del jugador. En este estado la bandera puede activarse mediante dos acciones distintas: depositar la pieza sobre otro escaque distinto, o levantar una segunda pieza. En esta situación el microcontrolador compara dos casos, uno en el que "tablero[i][j]==1" no coincida con el interruptor[i][j] (en caso que se deposite la pieza), y otro en el que "tablero[i][j]==0" no coincida con el interruptor[i][j] (en caso que se levante una segunda pieza).

En el primer caso pueden ocurrir dos cosas: un movimiento normal o un enroque. Como hemos explicado anteriormente, las piezas tienen como característica su propia posición, por lo que, conociendo su posición inicial, es posible saber si la pieza levantada en el estado anterior



era el rey. Si no lo es, se guardan las nuevas variables “i” y “j” en otras dos variables globales, “i2” y “j2”, y se pasa al siguiente estado. Si por el contrario se trata del rey se guardan “i” y “j” en “i2” y “j2” y se comprueba si el desplazamiento ha sido en horizontal y si ha sido de más de 1 escaque de distancia. Para hacer esta comprobación, comparamos el valor de “i1” anterior con el actual “i2”. Si no es mayor que 1 se pasa al siguiente estado. Si resulta que el desplazamiento es mayor que 1 escaque, se está realizando un enroque y habrá que esperar a que la torre se mueva para pasar al siguiente estado, por lo que esperamos a que un valor “tablero[i][j]==0” se diferencie de un interruptor. Una vez movida la torre, almacenamos en “ie1” y “je1” los valores “i” y “j” y esperamos a que “tablero[i][j]==1” se diferencie de un interruptor para almacenar el último movimiento del enroque en “ie2” y “je2”.

El segundo caso aquel en que se levanta una segunda pieza sin que la primera haya sido depositada. Esto solo ocurre cuando se realiza un movimiento de captura por lo que se busca la pieza que se encuentra en la segunda posición detectada y se establece su característica “estado” a muerta. El microcontrolador espera a que se deposite una pieza sobre la segunda posición detectada del tablero, entonces almacena las variables de posición “i” y “j” en “i2” y “j2” y pasa al siguiente estado.

En el siguiente estado los movimientos han terminado y por tanto ya conocemos “i1”, “i2”, “j1” y “j2” que conforman un movimiento completo. Es ahora cuando se comparan estos valores con los de los vectores tipo “char” que dependen de la posición del tablero. Los valores de “i” se corresponderán con el vector de columnas “char columna(8)=(a,b,c,d,e,f,g,h)|(h,g,f,e,d,c,b,a)” y los valores de “j” son para las filas “char fila(8)=(1,2,3,4,5,6,7,8)|(8,7,6,5,4,3,2,1)”. Con estas correspondencias se enviará al ordenador un comando de tipo teclado con las coordenadas del movimiento. El comando será `Keyboard.print(columna(i1)+fila(j1)+columna(i2)+fila(j2));` enviando por el USB mensajes del tipo a2a3 (peón de a2 → a3). En este mismo estado se actualiza la variable “tablero” poniendo un 1 en las viejas posiciones [(i1,j1) ,(ie1,je1)] y un 0 en las nuevas [(i2,j2),(ie2,je2)]. Tras esta actualización se cambia el turno y se pasa al segundo estado de nuevo.

Cuando termina la partida, o si se desea en algún momento de la misma, se puede reiniciar el tablero sin resetearlo. Cuando se encuentre en el segundo estado pueden pulsar ambos botones de selección de lado para pasar al estado 1 y comenzar una nueva partida. Puesto que este primer estado se queda en pausa, hasta que se selecciona uno de los dos lados, se pueden recolocar las piezas en ese momento sin desconectar el tablero.

Todo el código se encuentra subido en formato txt y pdf como anexo.

## 4. Conclusión

La realización de este proyecto me ha hecho comprender todos los aspectos del diseño de un producto desde su idea hasta su desarrollo final. Con esto me refiero a las ideas que van surgiendo conforme se va investigando los métodos de implementación de la versión original, hasta los problemas que surgen y que te hacen reconducir el proyecto por rutas alternativas.

Un ejemplo claro es el uso de circuitos RFID. En una primera instancia quería utilizar este tipo de circuitos pues son muy completos, toda la información de las piezas es siempre accesible, sin posibilidad de errores de posición. Pero estudiando en profundidad su comportamiento observé que tenían un radio de acción que abarcaba una zona más amplia de la que cubrían físicamente, por lo que podría detectar piezas vecinas. Por otra parte, no tardé en ser consciente del elevado coste económico que podían suponer: 5€ el circuito RFID. Realicé amplias búsquedas tratando de encontrar una venta al por mayor que redujese es coste a 1-2€ la unidad, pero no la encontré. Así que tuve que replantearme si el tablero sería factible sin conocer qué ficha estaba en qué casilla en cada momento. Al final opté por sensores más sencillos y mucho más económicos con los que la única forma de reconocer una ficha el diseño de un software con capacidad para ello.

### 4.1. Objetivos

En cuanto a los objetivos, considero que se han cumplido. Por un lado se ha conseguido digitalizar la partida enviando los movimientos como cadenas de caracteres al ordenador. Además se ha mantenido la imagen del tablero lo más intacta posible, teniendo en cuenta que el usuario debe poder saber en qué estado se encuentra el tablero y poder interactuar con él para seleccionar el lado de comienzo de la partida y para comenzar una nueva.

El objetivo económico de mantener el precio por debajo de los 360€ que cuesta el tablero más barato de tipo DGT, se ha cumplido con creces. El presupuesto total en componentes de este proyecto es de 100€. Por lo que con un precio de venta de 150€ se tendría unos beneficios del 50% y aún quedaría 210€ por debajo del DGT. La programación del microcontrolador y el tiempo invertido en el diseño del prototipo se verían disueltos ante una venta en masa.

Se ha conseguido también identificar el movimiento especial de enroque, ya sea corto o largo, aunque no se tiene en cuenta si el movimiento es legal o no. No se ha podido implementar por el contrario la captura al paso, ni la coronación. Ambas explicadas en la sección de posibles mejoras.



*Imagen 47 tablero terminado*

## 4.2. Limitaciones

El tablero sufre limitaciones provenientes de distintas fuentes. Encontramos en primer lugar la limitación física: el espacio disponible queda delimitado por el propio tablero. Esto significa que el número y tamaño de los componentes que se le pueden introducir es limitado, por ejemplo, el tamaño de los escaques, que impiden implementar sensores como el RFID.

En segundo lugar, encontramos limitaciones dentro del software del microcontrolador. En este caso ha habido un movimiento en concreto que no se ha programado, y es el de captura al paso. Este ocurre cuando un peón-1 avanza 2 escaques y un peón-2 enemigo lo captura en la posición intermedia entre el primer y último escaque del primer peón-1.

En tercer lugar, encontramos una limitación por el tipo de sensor que se ha utilizado, y es la identificación de las piezas, que al ser interruptores delegan dicha función de identificación al seguimiento por software.

## 4.3. Futuras mejoras

Todo proyecto puede, y debe, ser mejorado cuando las limitaciones en tiempo, recursos, conocimientos, tecnología, etc sean superadas. Las siguientes son posibles mejoras a implementar que, por algún motivo, no han podido ser objeto de este TFG.

Las primeras mejoras pasan por agregar los movimientos especiales y mejorar el enroque. Por ello la primera mejora consiste en agregar al microcontrolador la capacidad de reconocer una captura al paso. Esta función no se implementó porque dependía de muchos factores: que el movimiento hubiera sido en 1 solo turno y no 1+1, que no se hubiera movido a un tercer ataque y que el peón fuese el de su fila, que no se encontrase en la 4ª fila por haber capturado un peón enemigo, que hubiera ocurrido en el turno anterior del contrincante y que la pieza que captura sea un peón.

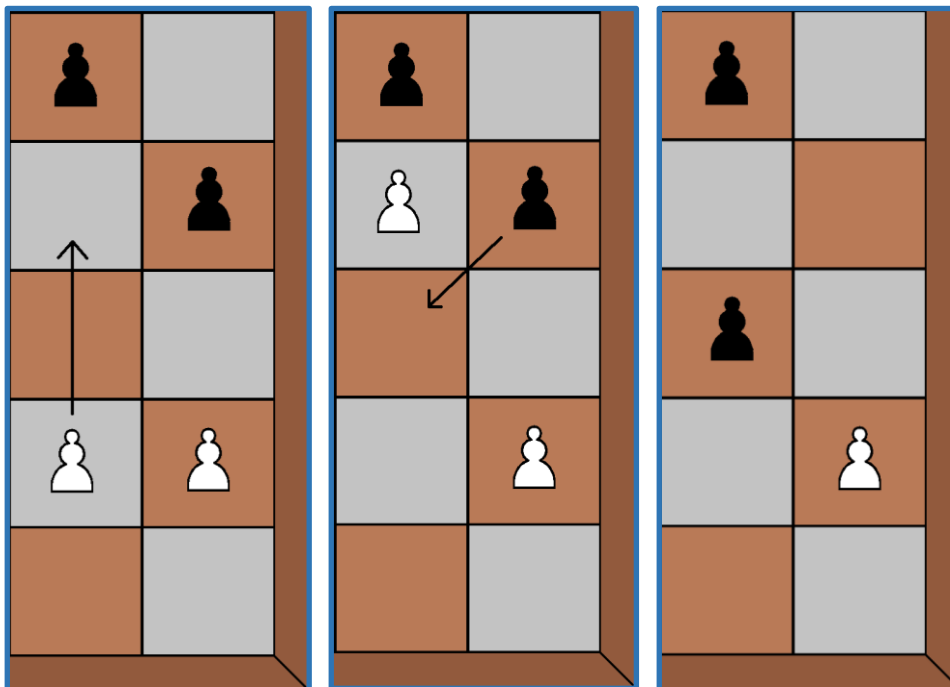


Imagen 48 Captura al paso(1/3) Imagen 49 Captura al paso(2/3) Imagen 50 Captura al paso(3/3)

Tras el desarrollo del código aprecié que se podían implementar algunos cambios que mejorarían la precisión de los movimientos. Utilizar el tipo de pieza para calcular a qué escaques puede desplazarse y así solo escanear dichos escaques y por tanto optimizar el escaneo de movimientos.

Por otro lado, el enroque es un movimiento en el que influyen muchos factores, como pasa con la captura al paso. Yo he optado por confiar en el cumplimiento de las normas, pero podría programarse el microcontrolador para que tenga en cuenta todas las limitaciones de este movimiento. Estos son los factores a tener en cuenta: si el rey se encuentra en jaque, si el camino al enroque está atacado por una pieza enemiga, si el resultado hará que el rey se encuentre en jaque o si el rey se ha movido anteriormente. La diferencia con la captura al paso es que este movimiento no implica la captura de ninguna pieza. Y es que este último movimiento necesita que el peón-2 capturado desaparezca, complicando así su implementación, ya que no muere en la posición final del peón-1 que captura, sino en la que se encontraba el peón-2 capturado.

Por último, tenemos la coronación de los peones. Para implementarla sería necesario identificar qué peón llega al extremo del tablero y de algún modo identificar a qué pieza promocionará el peón. Podría implementarse utilizando alguno de los botones integrados pulsando "X" veces para seleccionar en qué pieza promociona.

Dejando a un lado la parte del microcontrolador. Tras haber comenzado el máster MUIT (Máster Universitario en Ingeniería de Telecomunicación) y haber profundizado en los conocimientos de la asignatura de sensores, recordé que mediante un sensor de tipo hall podía identificar a qué distancia se encontraba un imán. además, este tipo de sensores pueden ser muy direccionales. Con este método de detección junto con la posición de los imanes dentro de las piezas podría haber detectado qué pieza se encontraba en cada escaque. De este modo se conseguiría un comportamiento idéntico al de los tableros DGT, que son capaces de identificar cada pieza. Con este método los movimientos son mucho más fáciles de identificar ya que se sabe qué piezas abandonan el tablero al ser capturadas, y que piezas se introducen al coronarse.

Estos sensores se podrían implementar igual que la matriz de interruptores reed, pero en lugar de introducir unos lógicos por las columnas, lo que se haría sería habilitar o no la alimentación de los sensores.

Además, independientemente del modelo (interruptores reed o sensores hall), se puede incorporar un módulo Bluetooth para comunicarse con el dispositivo que aloje el software de reproducción de partida, eliminando así el cable USB.

## 5. Bibliografía

1.

25PCS Reed Glass Magnetic Switches N/O SPST 300VDC 3X20MM [Internet]. eBay  
Disponible en: <http://www.ebay.com/itm/25PCS-Reed-Glass-Magnetic-Switches-N-O-SPST-300VDC-3X20MM-/231012529938>

2.

100 piezas del sensor fotoeléctrico de 5 mm LDR resistencia de 2 Pin Fotoresistencia.  
Disponible en: [https://www.amazon.es/piezas-sensor-fotoel%C3%A9ctrico-resistencia-Fotoresistencia/dp/B01EZZMLNE/ref=sr\\_1\\_3?ie=UTF8&qid=1480271334&sr=8-3&keywords=fotoresistencias](https://www.amazon.es/piezas-sensor-fotoel%C3%A9ctrico-resistencia-Fotoresistencia/dp/B01EZZMLNE/ref=sr_1_3?ie=UTF8&qid=1480271334&sr=8-3&keywords=fotoresistencias)

3.

NFC Pegatinas Etiqueta Protocolo ISO14443A aliexpress.com. Disponible en:  
[//es.aliexpress.com/store/product/10PCS-LOT-30mm-Ntag213-203-White-NFC-Stickers-Protocol-ISO14443A-RFID-Tags-For-Samsung-Galaxy/1727595\\_32420414569.html?src=ibdm\\_d03p0558e02r02](https://es.aliexpress.com/store/product/10PCS-LOT-30mm-Ntag213-203-White-NFC-Stickers-Protocol-ISO14443A-RFID-Tags-For-Samsung-Galaxy/1727595_32420414569.html?src=ibdm_d03p0558e02r02)

4.

Arduino - page 2 - SparkFun Electronico Disponible en:  
<https://www.sparkfun.com/categories/103?page=2>

5.

Arduino - Placas y Kits Arduino. Disponible en:  
<http://www.cetronic.es/sqlcommerce/disenos/plantilla1/seccion/Catalogo.jsp?idIdioma=&idTienda=93&cPath=1339>

6.

Building a Sish Board. Sish - Computer Interfacing DIY Chessboards. Disponible en:  
<http://sishchess.blogspot.com.es/p/building-sish-board-tutorial.html>

7.

chessprogramming - DGT Board. Disponible en:  
<https://chessprogramming.wikispaces.com/DGT+Board>

8.

McLoughlin I. Computer Peripherals. McGraw-Hill; 2011. Disponible en:  
<http://www.lintech.org/comp-per/04KB.pdf>

9.

Bulsink BJ. Device for detecting playing pieces on a board . US6168158 B1, 2001 .  
Disponible en: <http://www.google.tl/patents/US6168158>

10.

diodos de conmutación de Alta velocidad. Disponible en: [//es.aliexpress.com/item/Free-shipping-100PCS-do-35-1N4148-IN4148-High-speed-switching-diodes/32334444816.html?src=ibdm\\_d03p0558e02r02](http://es.aliexpress.com/item/Free-shipping-100PCS-do-35-1N4148-IN4148-High-speed-switching-diodes/32334444816.html?src=ibdm_d03p0558e02r02)

11.

Galga extensiométrica de precisión por catálogo. Disponible en:

[http://es.omega.com/pptst/SGD\\_LINEAR1-AXIS.html](http://es.omega.com/pptst/SGD_LINEAR1-AXIS.html)

12.

Getting started with Arduino!. tronixstuff. 2010 . Disponible en:

<http://tronixstuff.com/2010/04/04/getting-started-with-arduino-chapter-zero/>

13.

Grades of Neodymium. Disponible en: [http://www.ndfeb-](http://www.ndfeb-info.com/neodymium_grades.aspx)

[info.com/neodymium\\_grades.aspx](http://www.ndfeb-info.com/neodymium_grades.aspx)

14.

Iman de neodimio, 12x2mm de diámetro 2-100 UNIDADES. La tienda magnética.

Disponible en: <https://latiendamagnetica.com/products/iman-de-neodimio-12x2mm-de-diametro-2-100-unidades-agnet-envio-desde-espana>

15.

NFC Long Range RFID Reader SLI / TI2k With Extra Antenna. Disponible en:

<http://www.nfc-rfid-reader.com/sale-2881379-iso15693-nfc-long-range-rfid-reader-sli-ti2k-with-extra-antenna.html>

16.

Lector RFID IIC - SL030. [citado 4 de diciembre de 2016]. Disponible en:

<http://www.stronglink-rfid.com/es/rfid-modules/sl030.html>

17.

Los Proyectos de Berger - DIY. Disponible en:

<https://sites.google.com/site/proyectosdeberger/home>

18.

Galga de deformación RS Pro N11MA512023, 5mm. Disponible en: [http://es.rs-](http://es.rs-online.com/web/p/galgas-extensiometricas/0632180/)

[online.com/web/p/galgas-extensiometricas/0632180/](http://es.rs-online.com/web/p/galgas-extensiometricas/0632180/)

19.

Pro Micro - 5V/16MHz - DEV-12640 - SparkFun Electronics. Disponible en:

<https://www.sparkfun.com/products/12640>

20.

Prodoehl P. RasterWeb! » The Button. Disponible en:

<http://rasterweb.net/raster/2011/05/09/the-button/>

21.

Tablero electrónico Bluetooth con piezas modelo palo rosa/ébano. Disponible en:

<http://ajedrez.optyma.com/articulos/tablero-electronico-bluetooth-con-piezas-modelo-palo-rosaebano-codigo-10789/3138/>

22.

Tablero madera caoba 48 cm. Tienda EscaciMat. Disponible en:

<https://tiendaescacimat.es/es/tableros-ajedrez-madera/513-tablero-ajedrez-madera-caoba.html>

23.

Teensy USB Development Board. Disponible en: <https://www.pjrc.com/teensy/index.html>

24.

The DGT Electronic Chessboard USB. The Chess House. Disponible en:

<http://www.chesshouse.com//ProductDetails.asp?ProductCode=DGT128>

25.

Tienda especializada en productos de ajedrez - La Tienda de Ajedrez. Disponible en:

<http://latiendadeajedrez.com/>

26.

usb development board - Buscar con Google. Disponible en:

<https://www.google.es/webhp?sourceid=chrome-instant&ion=1&espv=2&ie=UTF-8#q=usb+development+board>

27.

Reed Switch operating characteristics from RRE! Magnetic Reed Switches and Reed

Sensors. 2013. Disponible en: <https://www.reed-sensor.com/characteristics-reed-switches/>