**Introducción**

La tarjeta UPCH es un sistema electrónico wearable pensado en detectar el patrón de ejercicios que hace un paciente en terapia física. Este sistema cuenta con sensores giroscopio y acelerómetro, los cuales permiten obtener los datos de movimiento y orientación del circuito en tiempo real. El microcontrolador cuenta con WiFi incorporado, de esta manera los datos son enviados de forma inmediata a cualquier dispositivo que se conecte a la red via MQTT, protocolo utilizado en aplicaciones de Internet of Things que es implementado encima del protocolo TCP/IP. El circuito está pensado en ser ubicado en una pulsera, la cual será usada por el paciente.

El sistema electrónico diseñado e implementado cuenta con las siguientes características generales:

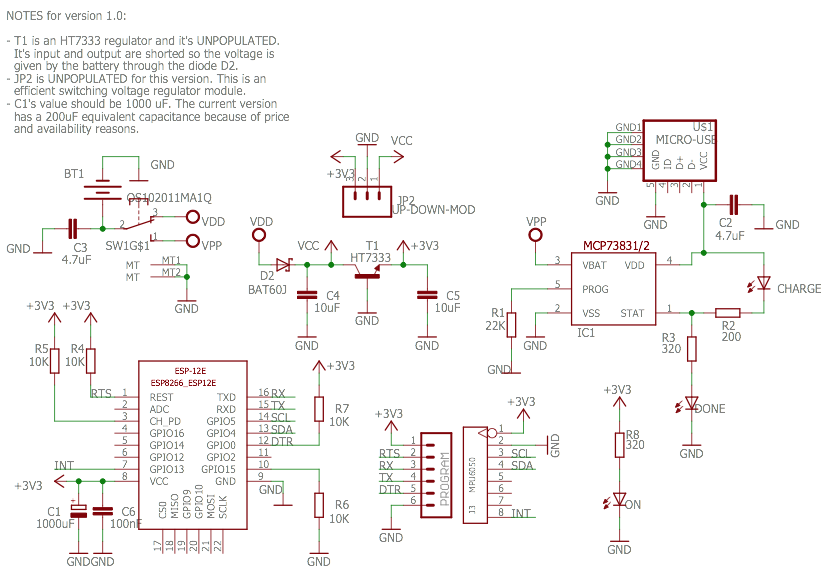
* WiFi incorporado, modo host y modo access point implementados.
* Envío de datos por medio del protocolo MQTT.
* Reconexión automática.
* Batería y cargador de batería con conector micro USB incorporados.
* Frecuencia de muestreo de los sensores está configurada a 20Hz.
* Filtro pasabajos de los sensores configurado a 42Hz.
* Sensibilidad del giroscopio configurada a +-250 grados/segundo.
* Sensibilidad del acelerómetro configurada a +-2g.
* Duración de energía a operación continua sin recarga de 30 minutos.
* Switch mecánico de encendio y apagado.
* Led indicador de encendido.
* Led indicador de carga en progreso y led indicador de carga finalizada.
* Monitoreo de batería (valor enviado por MQTT).

**Diseño electrónico**

El circuito cuenta con los siguientes componentes principales:

* Microcontrolador ESP8266  
  Microncontrolador de 32 bits operando a 80MHz con 96KB de memoria RAM y 4MB de memoria de programa y WiFi incorporado. Utiliza la interfaz I2C para comunicarse con el circuito integrado que aloja los sensores. También se utiliza un puerto serial para depuración y programación del firmware del circuito. Existen varias versiones de este microcontrolador, en el presente proyecto se ha utilizado el ESP12E.
* MPU6050  
  Circuito integrado que incorpora dos sensores inerciales, un acelerómetro y un giroscopio, en un solo circuito integrado. Es de bajo consumo y se comunica con el microcontrolador principal por el protocolo I2C.
* MCP73831  
  Circuito integrado controlador de administración de carga de baterías Li-Ion o Li-Polymer para batería LIR2450. Se encarga de administrar la carga de la batería incorporada a partir de una fuente de voltaje de 5VDC, administrada por el puerto micro USB. El switch de encendido desabilita este cargador mientras el circuito se encuentra operativo, esto es necesario para evitar un mal manejo de la carga que puede conllevar al mal funcionamiento de la batería.
* Batería LIR2450  
  Batería de Iones de litio recargable que proporciona 120 mAh de energía nominal. Totalmente recargada llega a medir 4.1VDC y operando nominalmente llega a 3.6VDC.

A continuación se puede observar el circuito esquemático del sistema:

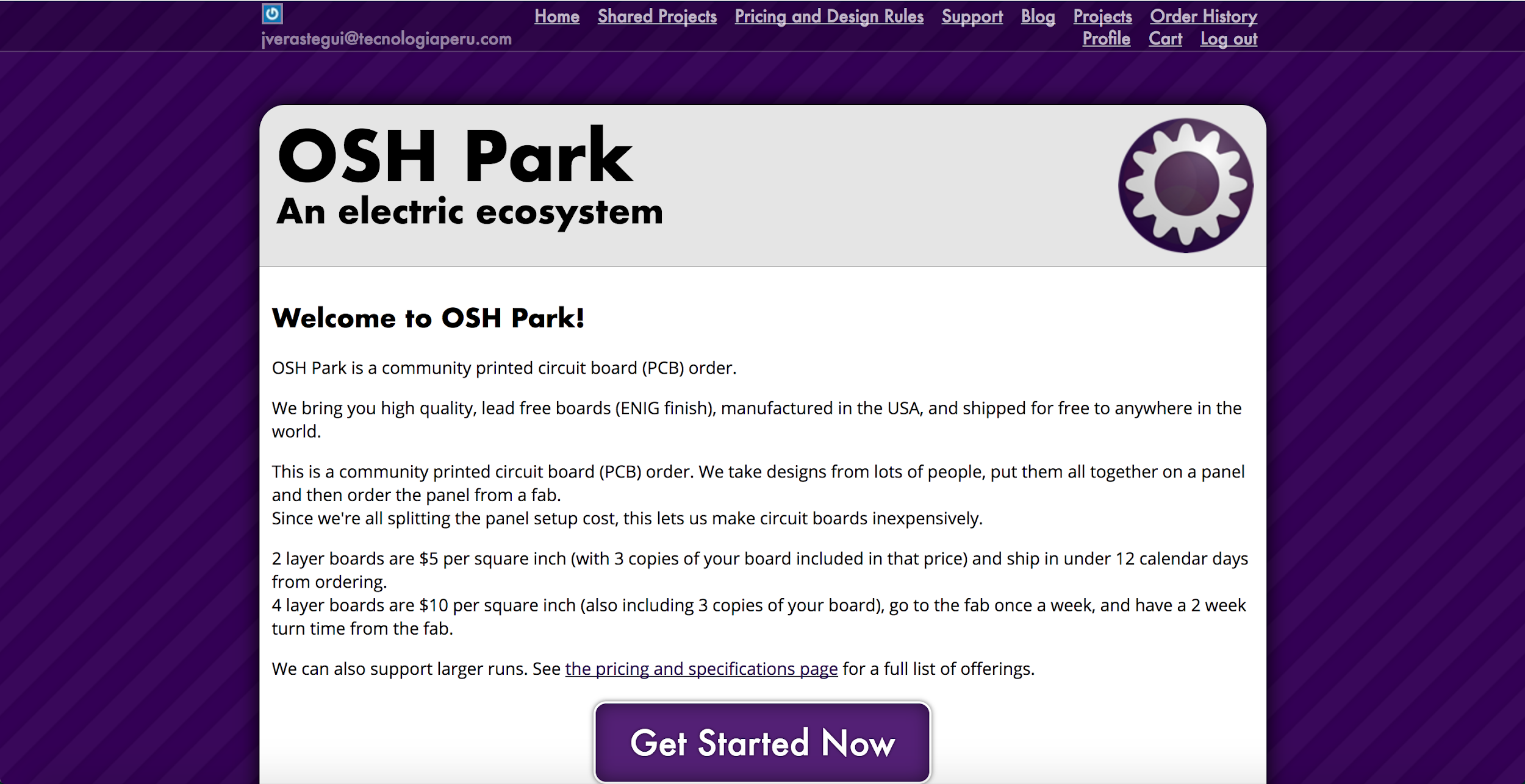


Se puede observar que el circuito cuenta con todos los componentes principales mencionados anteriormente. Adicionalmente se puede observar:

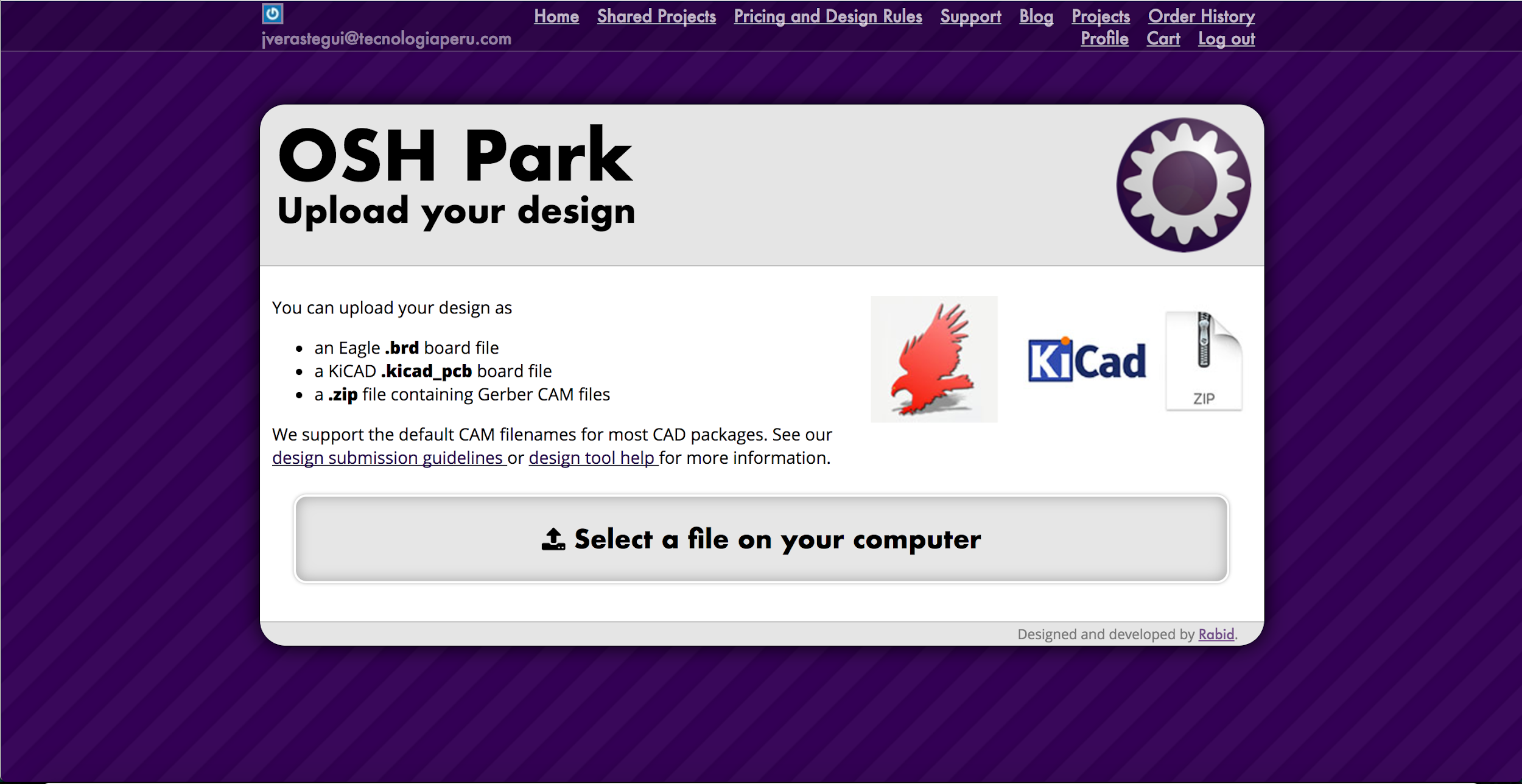
* Un puerto de programación llamado PROGRAM, este puerto cuenta con la interfaz serial (RX,TX), además de dos pines adiconales, el pin de control de programación, DTR, y el pin de reset del microcontrolador ESP8266, RTS.
* Un regulador de voltaje HT7333. **NOTA IMPORTANTE: Este regulador de voltaje no se encuentra soldado en el circuito. Para esta versión de hardware se debe soldar juntos los puntos VCC con +3V3 para que el circuito opere de manera habitual.**
* JP2 es un regulador de voltaje de tipo switching. **NOTA IMPORTANTE: Este regulador de voltaje no se encuentra soldado en el circuito. No se han hecho pruebas si la performance del circuito mejora con este regulador o no.**
* Leds indicadores de encendido (ON), carga en progreso (CHARGE) y carga finalizada (DONE).
* Es muy importante la presencia del diodo D2 BAT60, ya que el circuito no cuenta con regulador de voltaje, este diodo cumple dos funciones, proteger el circuito de cualquier posible retorno de corriente hacia la batería, lo cual podría dañarla y también adaptar ligeramente el voltaje de la batería.
* También es muy importante la presencia del condensador C1, este condensador es necesario para el correcto funcionamiento del microcontrolador ESP8266. Durante su operación este microcontrolador puede exigir picos de corriente que la batería no necesariamente podría proporcionar, esto genera dos consecuencias, que el voltaje de la batería caiga bastante, y que al no recibir la corriente necesaria, el microcontrolador no funcione correctamente y entre en un estado de inestabilidad. En la primera versión del circuito en vez de un condensador de 1000uF se ha colocado un condensador de 200uF que cumple con los requerimientos. A futuro se sugiere colocar uno de 1000uF, esto inclusive podría prolongar el tiempo de operación a más de 30 minutos además de prolongar el tiempo de vida de la batería.

**Pedido de compra de tarjetas**

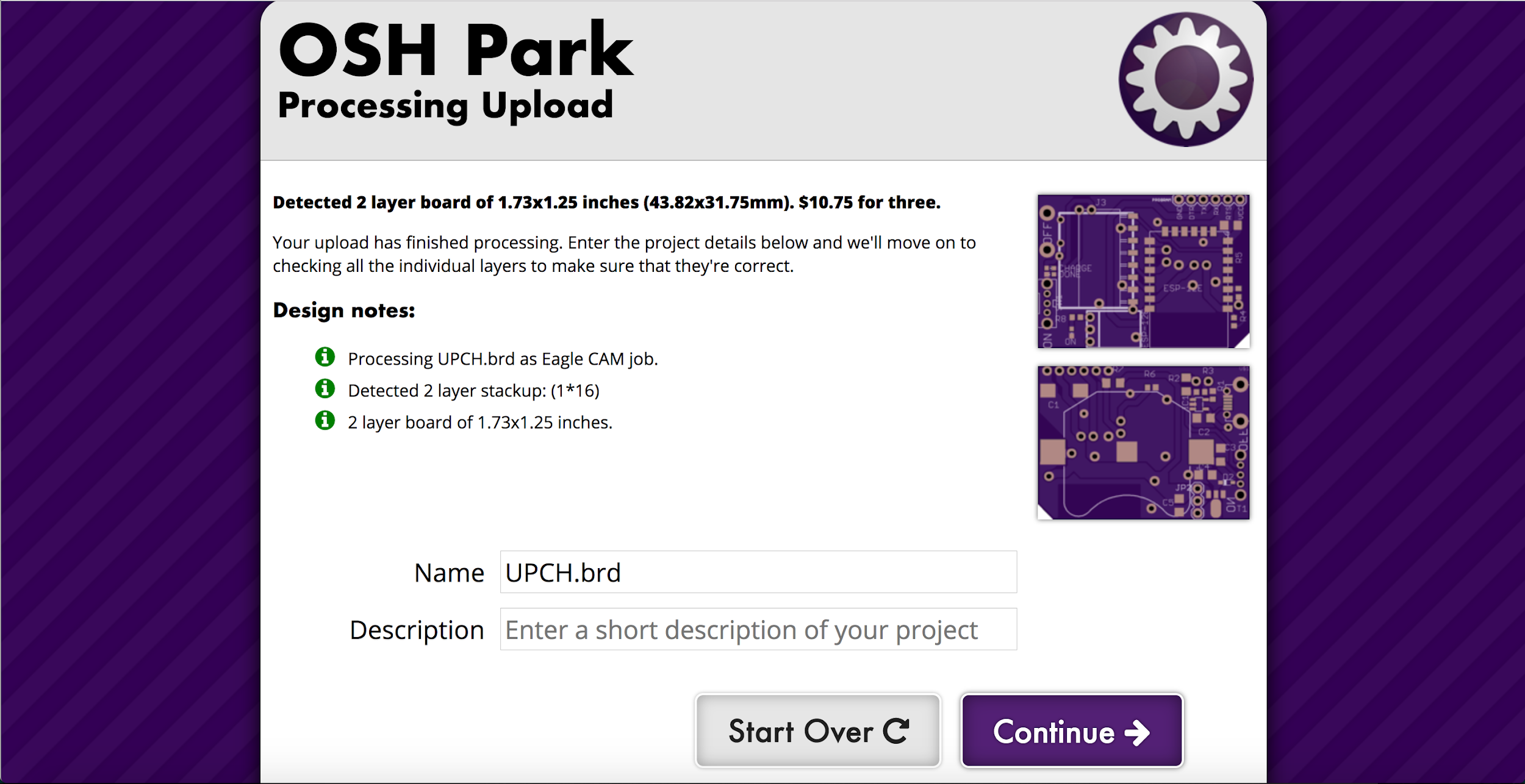
Los diseños validos creados en el software CAD Eagle se encuentran en los archivos **UPCH.sch** (circuito esquemático) y **UPCH.brd** (circuito impreso). Para ordenar uno o más circuitos se puede ingresar a la dirección web <https://oshpark.com/>.



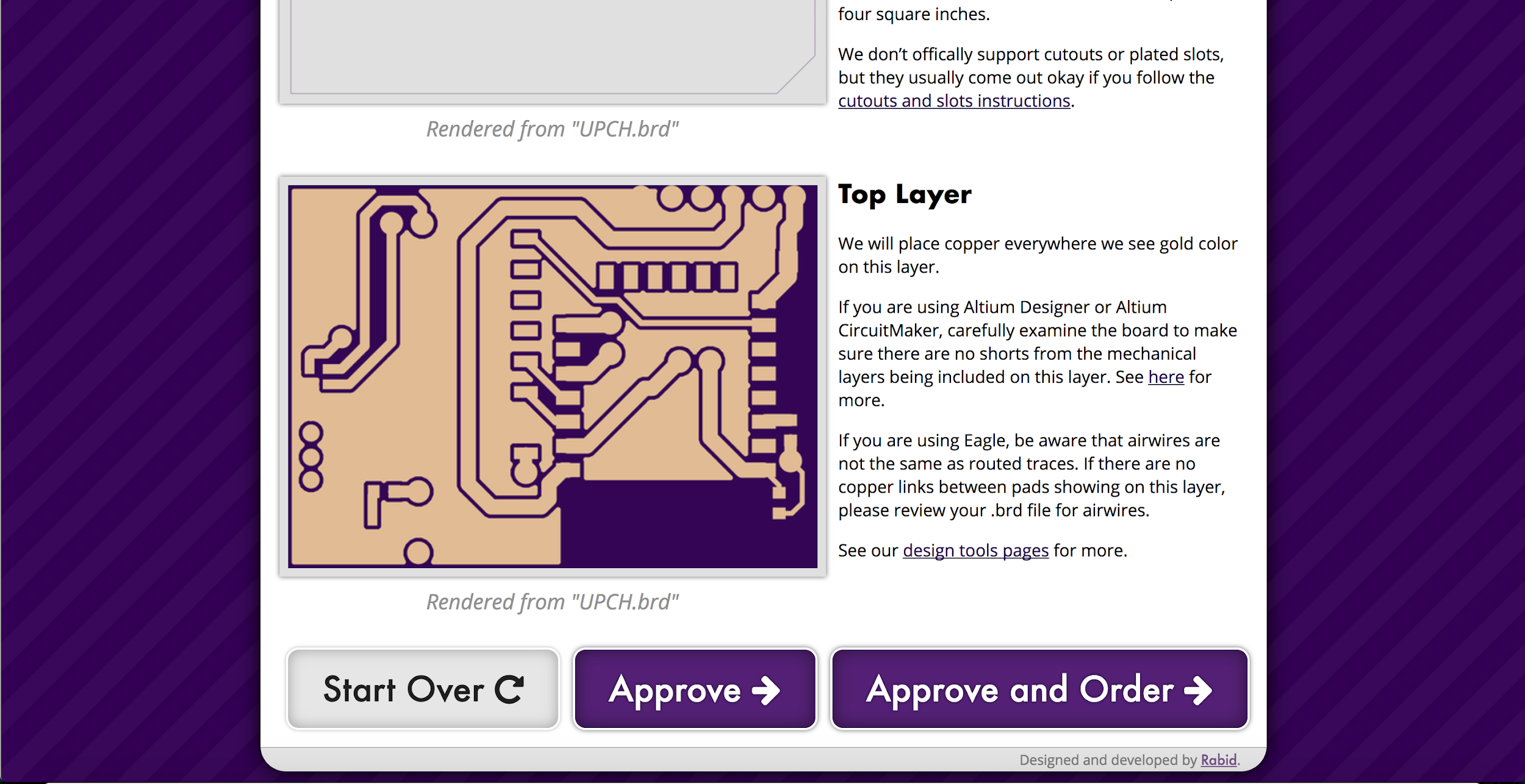
Luego ir al link “Get Started Now”, esto lo llevará a otra vista en donde podrá subir su archivo de Eagle:



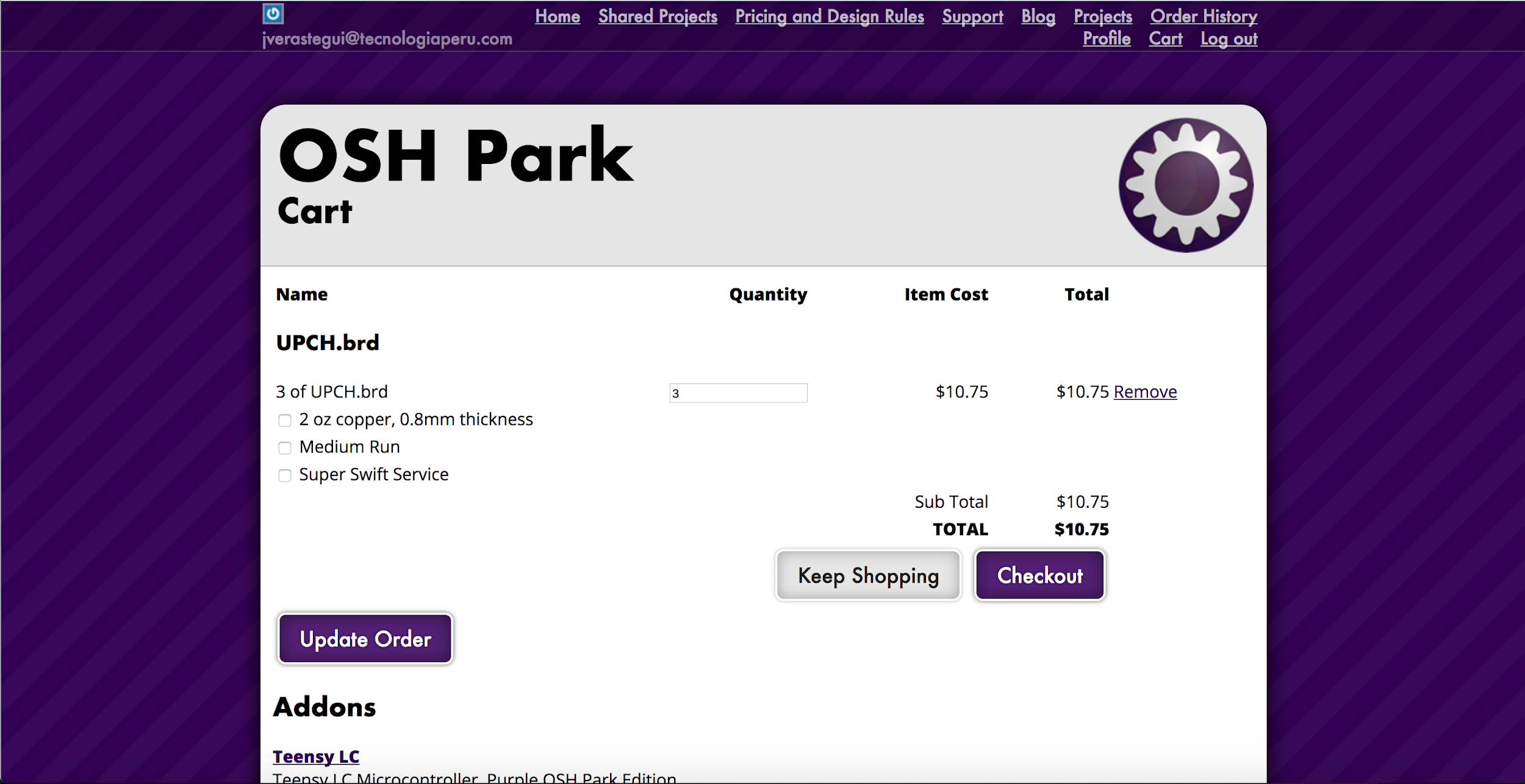
Seleccionar el archivo de Eagle llamado “UPCH.brd” y proseguir con las instrucciones de la página:



Una vez cargado el archivo, el sitio web les dirá si pueden proseguir, caso contrario revisar si han cargado el archivo correcto.



Luego de haber proseguido, bajar al final de la siguiente vista y presionar “Approve and Order” para poder ordenar el circuito.



Si todo es correcto, dar click en “Checkout” y proseguir con la compra.

**Firmware**

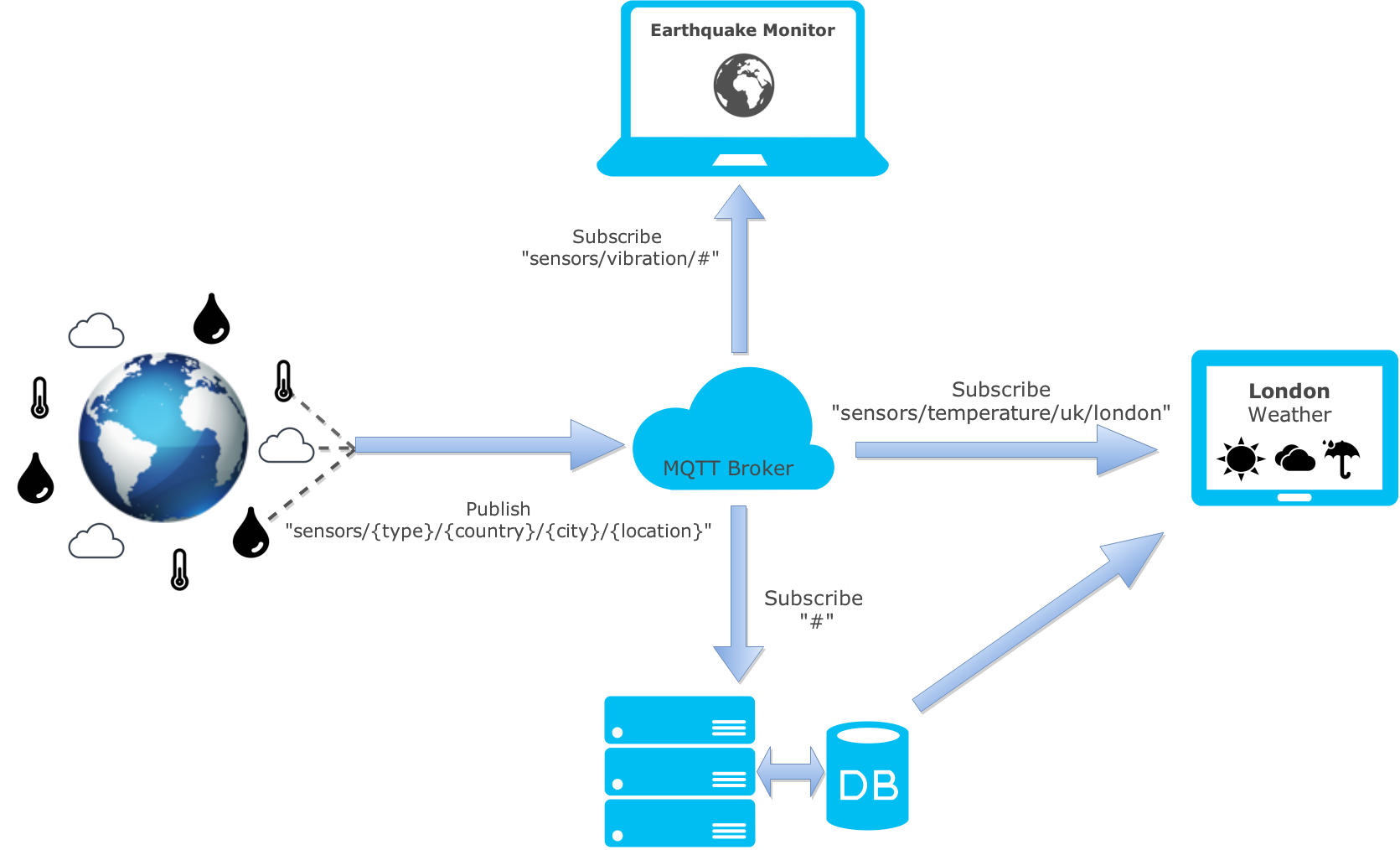
El código del microcontrolador ESP8266 ha sido escrito utilizando la plataforma de programación Arduino y todas sus librerías. Existen dos códigos diferentes para los circuitos. Como se conoce, el producto está pensado para contener dos circuitos en él. Esto para tener dos puntos de sensado y detectar las flexiones de los pacientes de mejor manera. Estos circuitos son identificados como MPU1 y MPU2, y se diferencian por algunas características:

* MPU1 es configurado como access point. Esto quiere decir que MPU1 crea una red WiFi para que el otro circuito y el servidor MQTT se conecten.
* MPU2 es configurado como host WiFi. Esto quiere decir que MPU2 se conecta a una red ya existente.
* Ambos circuitos se conectan a la misma red “HR XX”, las dos X representan dos números. Para evitar conflictos de comunicación cada pareja tiene configurada una red única empezando de la 01.
* MPU1 publica sus mensajes por MQTT en el topic MPU1.
* MPU2 publica sus mensajes por MQTT en el topic MPU2.

Dentro del microcontrolador ESP8266 ocurren muchas cosas en simultáneo, este microcontrolador se encarga de:

* Crear una red WiFi (MPU1).
* Conectarse a una red WiFi (MPU2).
* El circuito crea un servidor de sockets TCP/IP y se queda a la espera del envío de la IP del broker. Esto es necesario para poder saber en que IP se encuentra el broker ya que no siempre será la misma IP si es que existen otros dispositivos conectados a la misma red. La IP es enviado por sockets al circuito en el puerto 8000 y luego de esto el circuito elimina el servidor.
* El circuito crea un cliente y conecta el cliente al servidor MQTT cuya IP ya es conocida.
* Configurar el circuito MPU6050 via protocolo I2C.
* Recibir datos cada vez que el circuito MPU6050 tiene datos listos. Esto lo hace por medio de un pin de interrupción (INT en el esquemático), el ESP8266 está configurado para recibir una interrupción externa por este pin.
* Leer datos de acelerómetro y giroscopio uno por uno y convertirlos a binario.
* Empaquetar los datos binarios del acelerómetro y el giroscopio. Estos datos se empaquetan junto con dos datos más, el valor de voltaje de la batería y el tiempo exacto (tiempo relativo al encendido del circuito) de recepción del dato del MPU6050.
* Luego que el paquete está listo es publicado por mqtt bajo el topic “MPU1” o “MPU2” dependiendo del caso.
* Después de esto se queda a la espera de otro dato del MPU6050 y se repite el lazo.

MQTT es un protocolo de comunicación basado en mensajería simple, un estándar que corre encima del protocolo TCP/IP. Se utiliza ampliamente en aplicaciones de Internet of Things para conectar dispositivos de mensajería simple, típicamente sensores. A continuación se muestra un diagrama de una red MQTT:



Todos los dispositivos se conectan al broker que es el dispositivo encargado de administrar los mensajes y reenviarlos a los suscriptores. El publicador envia datos por medio de un topic y los suscriptores que se conecten a este topic recibirán estos datos.

**Programación del firmware**

El firmware utiliza varias librerias:

* PubSubClient: es encargada de manejar todo el protocolo MQTT.
* ESP8266WiFi: Se encarga de administrar todas las conexiones WiFi del circuito.
* DNSServer: \_\_\_
* ESP8266WebServer: Se utiliza para crear el servidor de sockets que recibe la IP del broker.
* I2Cdev: Librería customizada para comunicación con los sensores.
* MPU6050\_6Axis\_MotionApps20: Librería customizada para comunicación con los sensores.
* Wire: Librería utilizada para comunicación I2C.

Todas estas librerías se encuentran de manera pública en el IDE de Arduino, excepto por la del MPU6050. Esta librería se encuentra originalmente en <https://github.com/jrowberg/i2cdevlib/tree/master/Arduino/MPU6050>. Sin embargo, ha sido editada para el uso en este proyecto. Es por eso que esta librería se encuentra dentro de la documentación en el directorio con nombre “**MPU6050\_library**” y debe ser instalada por separado. En esta guía no se explicará el uso del IDE de Arduino, ni cómo instalar librerías, estos temas se encuentran ampliamente documentados en la web. Cabe resaltar que para utilizar el IDE de Arduino con el microcontrolador ESP8266 es necesario configurar el IDE de Arduino previamente. Esto tampoco se explicará en esta guía.

La programación del circuito UPCH es realizada con un programador externo. Se puede utilizar cualquier adaptador serial a usb del mercado que tenga la señal **DTR,** particularmente en este proyecto se utilizó el programador de la siguiente figura:



Este programador es basado en el chip FT232RL y es conocido como adaptador serial a USB, se consigue fácilmente en tiendas de la calle Paruro en el centro de Lima. Lo único que se debe hacer es conectar cada pin correspondiente al circuito (RX, TX, DTR y RTS, además de 3.3V y GND) y luego programar los circuitos con el IDE de Arduino. Hay que tener en cuenta que los pines de conexión de la tarjeta UPCH deben ser pequeños de tal manera que puedan entrar en su respectivo case. Se sugiere utilizar clips o pinzas para conectarse a estos pines.

**IMPORTANTE: El circuito UPCH funciona a 3.3 voltios, tener en cuenta esto al momento de utilizar cualquier programador, en el caso del de la figura tiene unos pines selectores para 5 voltios y 3.3 voltios.**

Los códigos a programar son **MPU6050\_DMP6\_ESP8266\_1.ino** y **MPU6050\_DMP6\_ESP8266\_2.ino** que corresponden a los circuitos MPU1 y MPU2 respectivamente. Anteriormente se mencionó la diferencia de estos dos circuitos. Se debe utilizar el IDE de Arduino para la programación, en el presente proyecto se utilizó la versión 1.8.1 del software.