

Informe del Proyecto

The Bomb Game

Integrantes:

Diaz, Maximiliano

Torres Baldi, Tehuel

Zaracho, Rosalí

Objetivos

El objetivo inicial del proyecto fue desarrollar un juego de ingenio/puzzle en el que se debe descifrar un código secreto con una limitada cantidad de intentos, tanto de manera offline, como online contra otro jugador.

Además el juego se conecta a través de WiFi a un servidor dedicado que permitirá llevar un registro de los puntajes más altos, así como también publicar en Twitter los resultados de las partidas.

Desarrollo

El desarrollo del proyecto pasó por distintas etapas, en las que fuimos construyendo y conectando las diversas partes de las que está compuesto el juego.

Lógica del Juego

Comenzamos desarrollando la base del juego en sí mismo, que es la lógica de adivinación del código, así como las condiciones de victoria y derrota.

La idea del juego es una adaptación de un juego de mesa conocido como *MasterMind*¹. Aún más específicamente, nos basamos en una adaptación de dicho juego, pero con una temática de desactivación de bomba, que existe para Android, llamada Bomb Simulator², de un desarrollador llamado Alexey Dvortsov.

El juego consiste en adivinar el código de desactivación de la bomba con una limitada cantidad de intentos. el código está compuesto de 4 dígitos distintos. Por cada posible solución que se arriesga se consume un intento, y el juego informa para esa posible solución dos datos:

1. Cuántos de esos números están en el código solución, pero en alguna posición distinta a la que fueron ingresados.
2. Cuántos de esos números están en el código solución, y exactamente en la posición que fueron ingresados

¹ [https://en.wikipedia.org/wiki/Mastermind_\(board_game\)](https://en.wikipedia.org/wiki/Mastermind_(board_game))

² <https://play.google.com/store/apps/details?id=dvortsov.alexey.bomb2digits>

Ejemplo de una Partida

El código que se debe adivinar es 1234.

Código Ingresado	Respuesta del Juego	Detalle
2143	C:4 P:0	Hay 4 números que son correctos, pero ninguno en su posición correcta
1258	C:2 P:2	Hay 2 numeros que son los correctos, y de esos números hay 2 que están en sus posiciones correctas

Manejo de Pantalla + Shift Register

Para reducir la cantidad de pines que ocupa la pantalla en el arduino, se optó por usar un shift register. Y para mejorar la interconexión de la pantalla con el shift, ensamblamos el chip sobre una pequeña placa perforada la conexión necesaria para todos los componentes.

Para la interconexión del módulo con la pantalla armamos un cable personalizado usando un cable plano de un IDE. Esto nos permitió que para el armado final queden las conexiones mucho más simples de manejar, en contraste con manejar cada cable por separado.

Esto permitió realizar la conexión ocupando tan solo 3 pines del arduino, y usando un solo cable desde la pantalla.

Se puede ver en la sección de diagramas como resultó la conexión final.

Conexión WiFi

Para permitir las partidas multijugador y también publicar los puntajes de las partidas, es necesaria una conexión a internet. Con tal fin decidimos usar un módulo wifi.

El módulo que conseguimos es específicamente el ESP8266.

Debido al alto consumo del módulo, la conexión final de alimentación no pudimos hacerlo directamente sobre el arduino, sino que necesitamos hacerlo de forma externa. Para esto tuvimos que armar un pequeño circuito que nos permita alimentar al módulo con los 3.3V que éste requiere.

Para la conexión física con el módulo también decidimos ensamblar una placa perforada (al igual que con el wifi) para facilitar las interconexiones. En la sección diagramas se puede ver el resultado final.

A nivel código el manejo de las conexiones wifi se hizo en primera instancia directamente comunicándose por serial con el módulo wifi.

Más tarde encontramos una librería que nos permitió realizar el manejo del con algo más de abstracción.

La librería se utiliza para listar las redes dentro del alcance, establecer conexiones, y mandar datos a través de TCP.

Manejo de Teclado

El teclado que usamos es un teclado de matriz de 4x3. Para el control del mismo se usó una librería.

La librería permite el ingreso de los numeros, y tambien de letras. Por cada caracter que se ingresa, se recibe el valor del código ASCII de la letra que se presiono. Para conseguir el valor en decimal (que es el valor que hay que adivinar) se hace un pequeño calculo al valor asci obtenido.

Ensamblaje

Fabricamos una caja de madera para que contenga todos los componentes que conforman nuestro proyecto.

Las conexiones de la pantalla y teclado decidimos dejarlos expuestas para darle una estética improvisada, similar a la bomba del Counter Strike³.

Server Side

el servidor al que se conecta el juego está programado en NodeJS.

Para establecer la comunicación de la forma más simple posible, establecimos un pequeño protocolo. Mediante el protocolo, usando los siguientes comandos podíamos coordinar las partidas.

Protocolo

Comando	Descripción
E	El cliente quiere unirse a una partida Online.
G	El cliente avisa que ganó la partida en la que está participando.
N	El cliente informa su número de usuario, que es el id en la base de datos.
P	El cliente informa su puntaje en la última partida jugada.

³ <http://files.gamebanana.com/img/ss/srends/54ad6dc2e7220.jpg>

<i>default</i>	El comando es enviado directamente al otro arduino en la partida online.
----------------	--------------------------------------------------------------------------

El servidor escucha en 2 puertos específicos, cada uno sirviendo a un propósito distinto

- Puerto **4000**: es para coordinar partidas en red.
- Puerto **8080**: Es para publicar el puntaje y el número de jugador de las partidas en solitario. Al finalizar una partida el cliente envía a este puerto el número de usuario y el puntaje obtenido. Si el usuario existe en la base de datos, se hace una publicación en twitter referenciando al usuario. Si no existe se publica como anónimo en el Twitter del juego⁴.

Algunos Inconvenientes

librerías wifi, websockets

Uno de los inconvenientes más grandes que tuvimos fue con la utilización de websockets, que era nuestro plan inicial para el control de las partidas en red.

La forma en que estábamos usando el modulo wifi (comunicandonos por serie, enviando directamente los comandos en bajo nivel) no nos permitía establecer el tipo de conexión que estábamos buscando. Investigamos bastante, pero no pudimos conseguir una librería que nos permita establecer websockets.

Algo destacable es que encontramos librerías para hacer correr código arduino directamente dentro del microcontrolador que es usado por el módulo wifi, es decir, usar solo el modulo wifi como si fuese un arduino. Interesante, pero no era para nada lo que estábamos buscando.

Finalmente decidimos usar una librería que nos permita el manejo de conexiones wifi, pero sin poder establecer sockets. Hubo un cambio de planes, y finalmente las comunicaciones se hacen directamente con conexiones TCP.

librerías shift register + lcd

Al decidir conectar la pantalla LCD mediante un shift register al arduino conseguimos reducir la cantidad de pines necesarios para la conexión, pero surgió el problema de encontrar una librería que sea compatible con este tipo de conexión.

Luego de pelearnos con varias librerías, y no tener éxito con ninguna, usamos una que nos recomendó el profe, lo que nos aseguró que la librería funcionaba.

⁴ <https://twitter.com/ArduBombGame>

A partir de esta librería el siguiente problema fue "descifrar" los pines que se usaban para la conexión. Esto lo resolvimos por "fuerza bruta" probando todas las conexiones posibles. Lo bueno es que eran solo 3 cables!

alimentacion del modulo wifi

Uno de los inconvenientes que se nos presentó a último momento del proyecto, en la etapa de armado, fue la alimentación del módulo wifi. El módulo se alimenta a 3.3V, con un consumo que puede llegar a los 200 mA, mientras que la salida de 3.3V de Arduino está limitada a 50 mA, lo que generaba que la aplicación arranque, pero no se puedan establecer conexiones wifi.

Para solucionarlo se decidió alimentar de forma separada al módulo wifi. Inicialmente pensamos hacerlo usando un regulador de voltaje. al no conseguir un regulador directo (por ejemplo el LM7833⁵) tuvimos que usar un regulador variable, y mediante los valores de resistencias que le soldamos logramos fijar la salida a 3.3V, y soportando hasta 1.5A de consumo.⁶

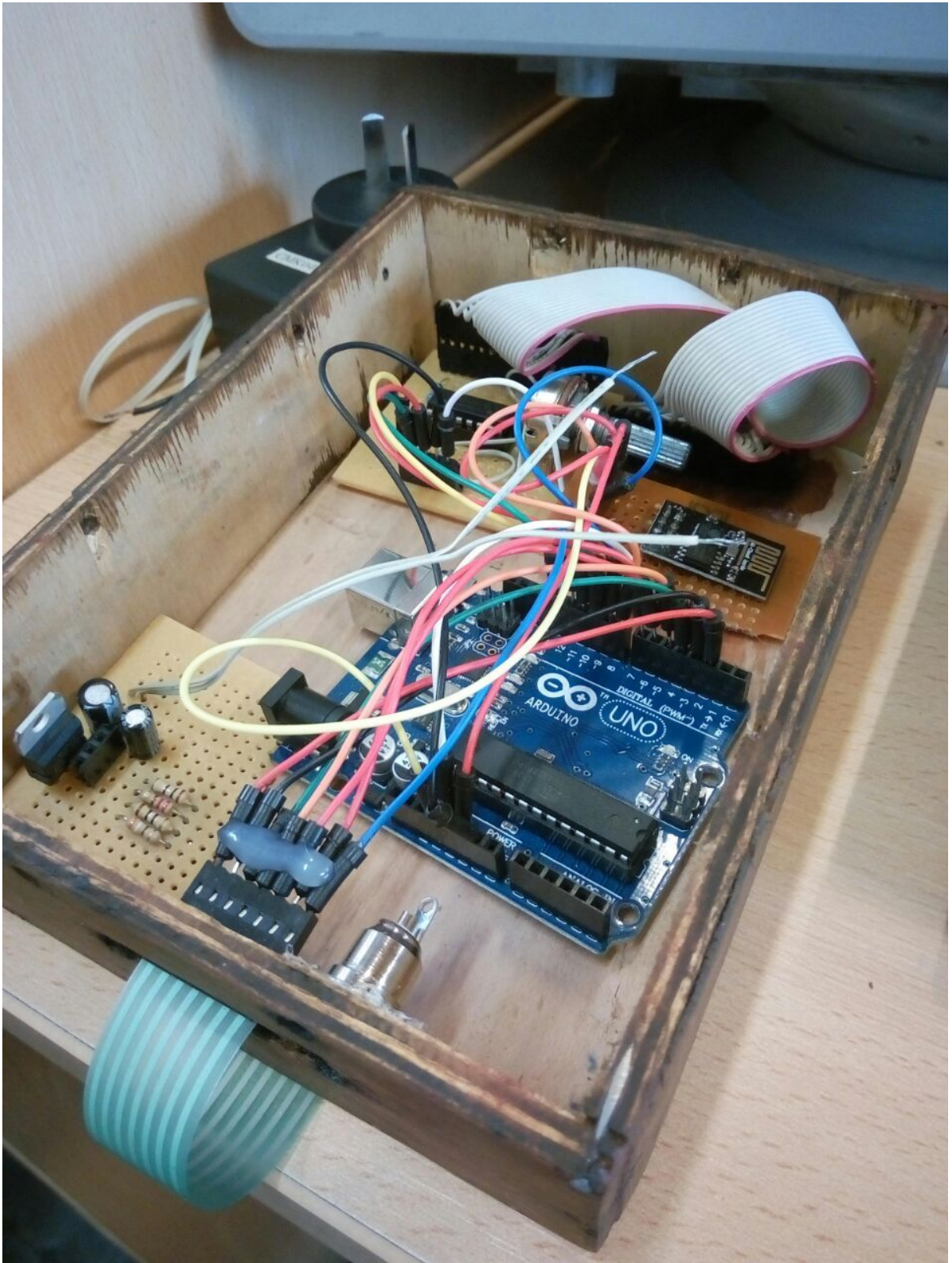
Tuvimos que diseñar y ensamblar una pequeña placa que contenga todos los componentes necesarios para la conversión. Esta placa es la que recibe la alimentación desde el exterior, y por un lado alimenta directamente al Arduino con la misma tensión de entrada, y por otro lado alimenta a la placa wifi, con los 3.3V que eran requeridos.

⁵ <http://www.alldatasheet.com/datasheet-pdf/pdf/237960/TAITRON/LM7833.html>

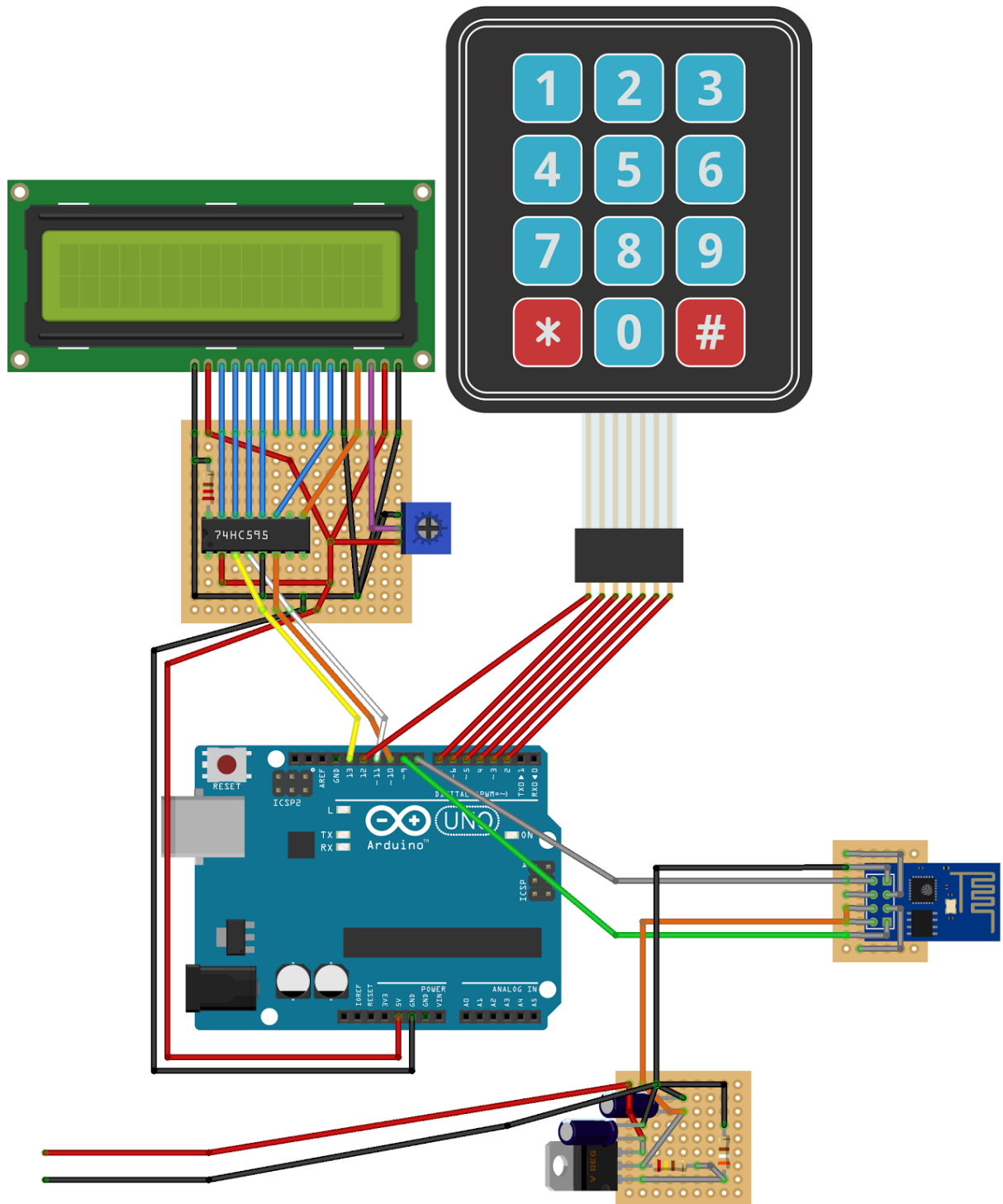
⁶ <http://kb.kaminskiengineering.com/node/171>

Resultado Final





Diagramas de Conexiones



fritzing

Diagrama visual

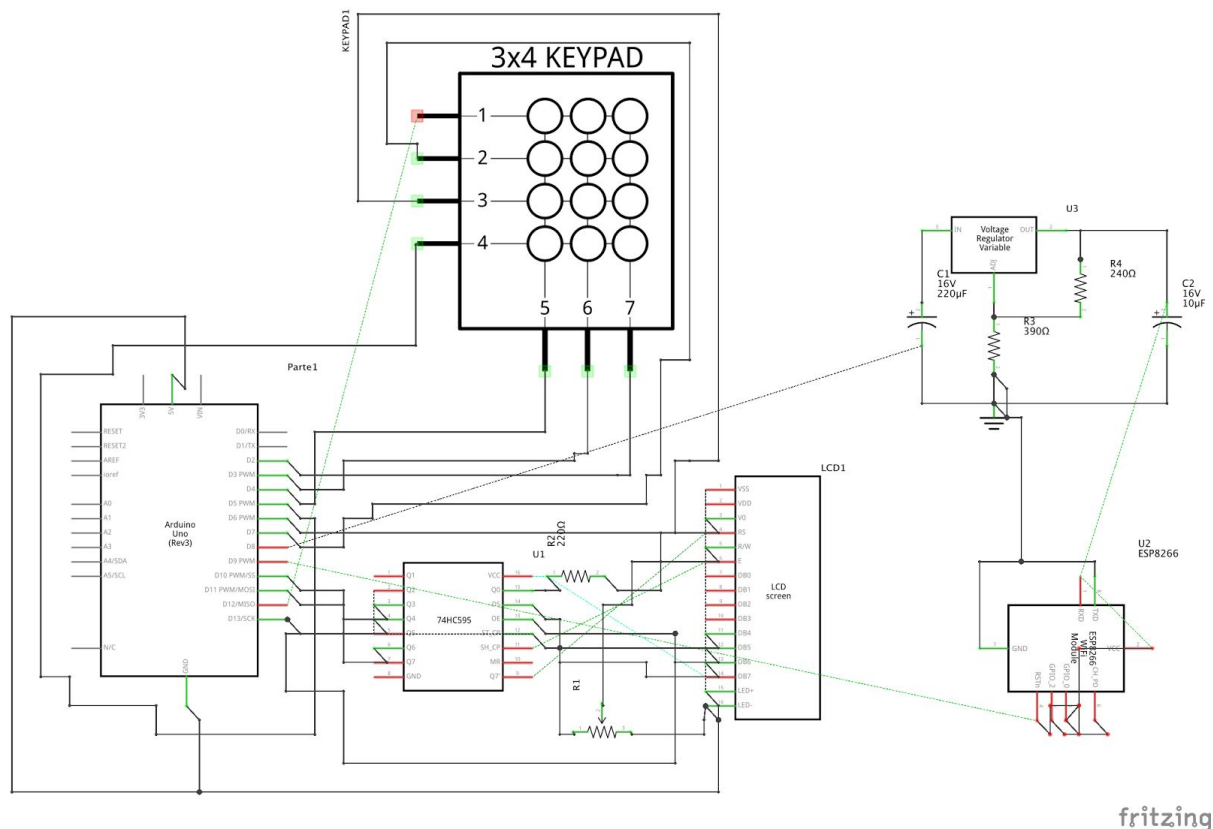


Diagrama Esquemático

Conclusiones

Fue un desafío bastante interesante y ambicioso. Creemos haber cumplido satisfactoriamente con todo lo que nos propusimos.

Por un lado nos llevamos un montón de buenas impresiones con el arduino, y con la facilidad con la que se pueden hacer y controlar un montón de dispositivos y sensores.

Por otro lado tuvimos bastantes problemas cuando nos íbamos metiendo en terreno más específico. Cuando ya existe la librería apropiada para lo que quieres hacer, es sencillo y divertido, pero si te alejas un poco de ese camino empiezan a aparecer problemas por todos lados.

En su mayoría pudimos resolver todos los inconvenientes que nos surgieron, y en algunos casos aprendiendo bastante de las soluciones, pero en otros fue cuestión de dar exactamente con la librería (caso de LCD + Shift), y en otros nos fue imposible de resolver (caso websockets + wifi).