



Universidad de Costa Rica

VICERRECTORÍA DE ACCIÓN SOCIAL

TRABAJO COMUNAL UNIVERSITARIO

TC-691 TROPICALIZACIÓN DE LA TECNOLOGÍA

DOCUMENTACIÓN DE HARDWARE Y SOFTWARE PARA LOS BOTONES DE LA CAJITA DE SATISFACCIÓN

Creador:

Andrés Chaves Vargas

B92198

29 de febrero de 2024

1. Información Necesaria de Fio V3 Sparkfun.

A continuación se muestra la distribución de pines del Fio V3 Sparkfun:

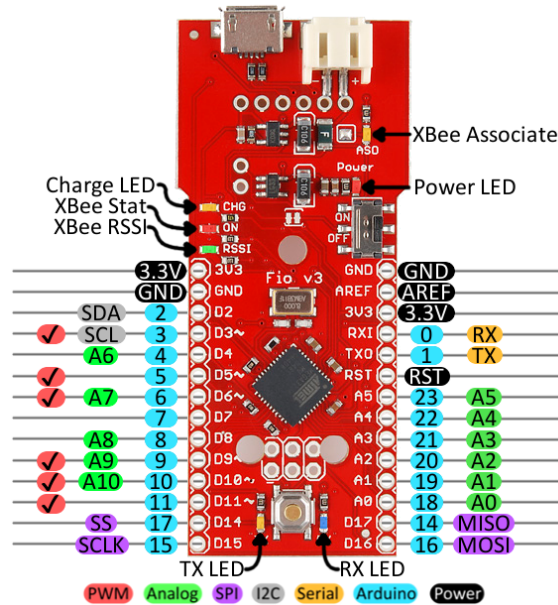


Figura 1: Distribución de Pines Fio V3 Sparkfun.

Para comprender mejor el funcionamiento de cada pin se puede ingresar al siguiente link: <https://learn.sparkfun.com/tutorials/pro-micro--fio-v3-hookup-guide/hardware-overview-fio-v3>

1.1. Proceso de Instalación de Sparkfun ARDUINO Boards

Para poder instalar la placa del arduino Fio V3 Sparkfun se debe seguir el siguiente procedimiento:

1. Ingresar a: https://github.com/sparkfun/Arduino_Boards/blob/main/README.md y buscar los pasos de instalación copiando el url dado en ese sitio. Y buscar el modelo de instalación de AVR boards.
2. Luego de instalar las placas tal y como se muestra en el procedimiento anterior. se debe seleccionar el Sparkfun Fio V3.

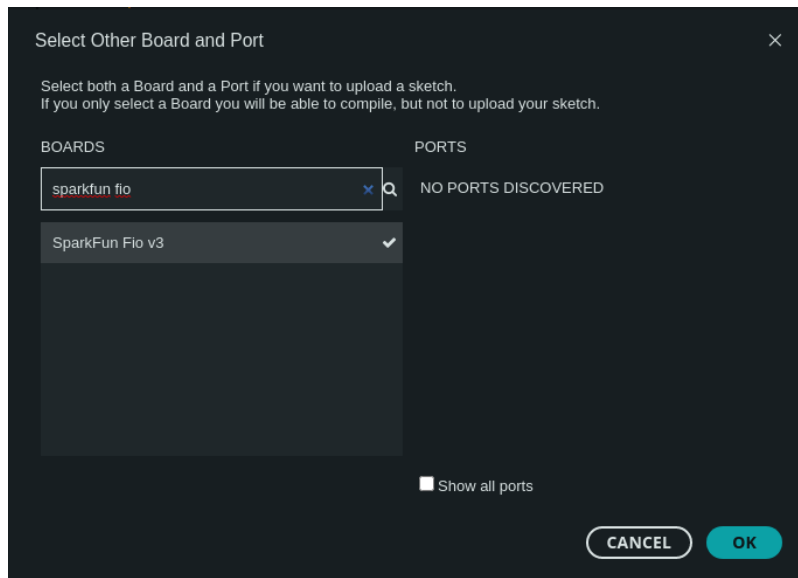


Figura 2: Muestra de la Placa Fio V3.

1.2. Proceso de Intalación de GFButtons

Para facilitar la implementación de los botones se utilizará la librería "GFButton.h". La documentación de dicha librería se puede encontrar en <https://bitbucket.org/geekfactory/gfbutton/src/master/README.md>

Para instalarla la librería se debe seguir un proceso sumamente sencillo:

1. Ir a *Sketch/Include Library/Manage Libraries*.
2. Una vez en el Library Manager buscar "*GFButton*". A continuación aparecerá una única librería que se debe instalar.

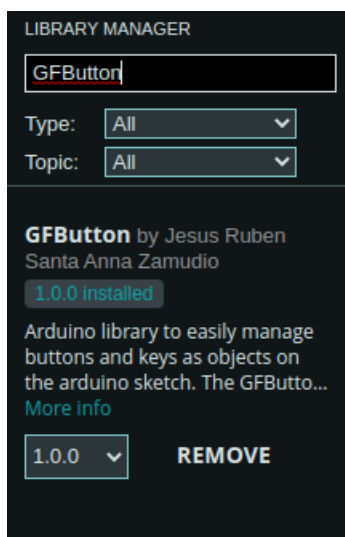


Figura 3: Muestra de la Librería GFButton.

3. Una vez instalada la librería ya es posible utilizarla. Solamente se debe incluir al inicio del código la librería como "GFButton.h".
4. Es importante destacar que, según la documentación, esta librería solamente ha sido probada en Arduino UNO. Pero nunca en una placa Fio V3. Y que además, las pruebas realizadas para crear el código serán implementadas en la aplicación SimulIde específicamente en una placa de Arduino UNO. Debido a que la placa Fio V3 no se encuentra disponible para simular. Es por este motivo, que la persona encargada de realizar las pruebas en forma física del correcto uso de los botones en la placa Fio V3, debe tomar en consideración que puede no funcionar correctamente.

1.3. Creación de Código

En el siguiente artículo se mencionan aspectos importantes a tomar en consideración para la programación de los botones, así como el hardware necesario para evitar la existencia de valores desconocidos e incluso el control de rebotes: <https://www.geekfactory.mx/tutoriales-arduino/boton-o-pulsador-con-arduino/>

En caso de tener alguna duda con el código creado, dicho artículo puede ser sumamente útil para comprender mejor el funcionamiento.

1.4. Hardware

A continuación se muestra un ejemplo del arduino UNO simulado con el hardware necesario para la implementación del funcionamiento de los botones:

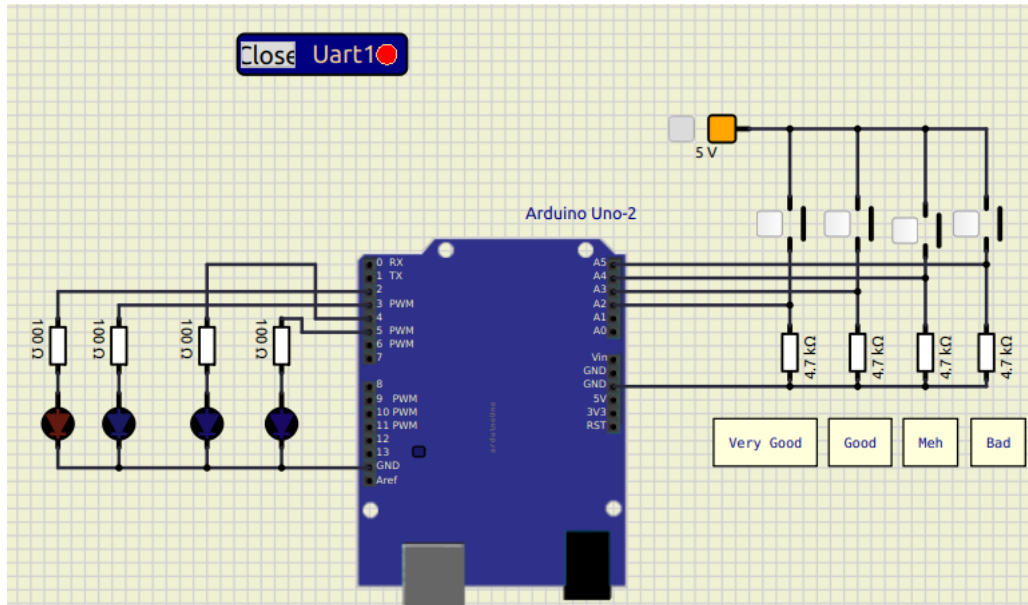


Figura 4: Hardware de los botones usando Arduino UNO.

Una vez que se vaya a implementar el circuito físico con el Fio V3, se puede realizar un diseño similar al anterior, solamente cambiando un microcontrolador por otro, cambiando la placa en el IDE de Arduino y los pines que conectan tanto a los botones como los LED's.

En este caso, ya que no se tiene una pantalla para mostrar salidas seriales, se optó por añadir led's que se enciendan dependiendo de la votación realizada, y una vez que se apaguen, quiere decir que la siguiente persona puede votar. Esto para dar un poco más de tiempo entre cada votación para tener un mejor control de rebotes a demás del ya implementado mediante software.

Provisionalmente se implementaron cuatro botones, según se muestra en la figura anterior. Cada botón se puede usar en cualquier orden.

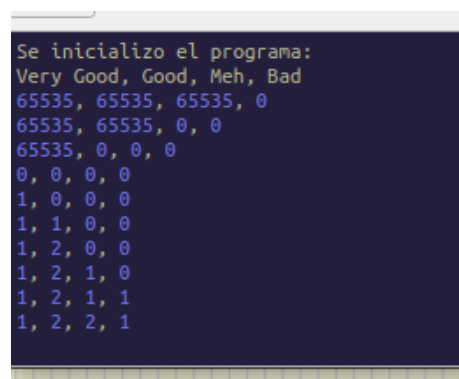
1.5. Software

Tal y como se mencionó anteriormente se utilizó la libreria GFButton para implementar los botones. La función `button1_on_press` nos permite determinar la cantidad de veces que se ha tocado un botón. Las demás funciones son exactamente iguales. Cada función imprime el total de veces que se han tocado todos los botones, para obtener siempre un valor actualizado.

Es importante destacar que debido a que el Fio V3 no cuenta con entrada SD, se vió en la obligación de implementar el envío de los datos de los botones mediante el puerto serial. Por lo tanto, además del archivo *buttons.ino* es necesario utilizar otro archivo llamado *datasaver.py*. El cual se encarga de crear un archivo de tipo csv y guardar los datos enviados por el arduino a dicho archivo.

Una vez conectado el arduino físico, se debe revisar que la velocidad de envío de bits sea de 9600 bauds mientras que el puerto al cual se están enviando los datos sea el correcto. Solamente el puerto se debe cambiar en caso de tener problemas de envío. Esto se puede revisar desde el código implementado en *datasaver.py*. Una vez que se hayan hecho los cambios necesarios, y el arduino esté encendido, se debe ejecutar el código *datasaver.py* para guardar la información respectiva.

A continuación se muestra un ejemplo de como debería verse la salida del arduino, que también debería imprimirse en el archivo csv:



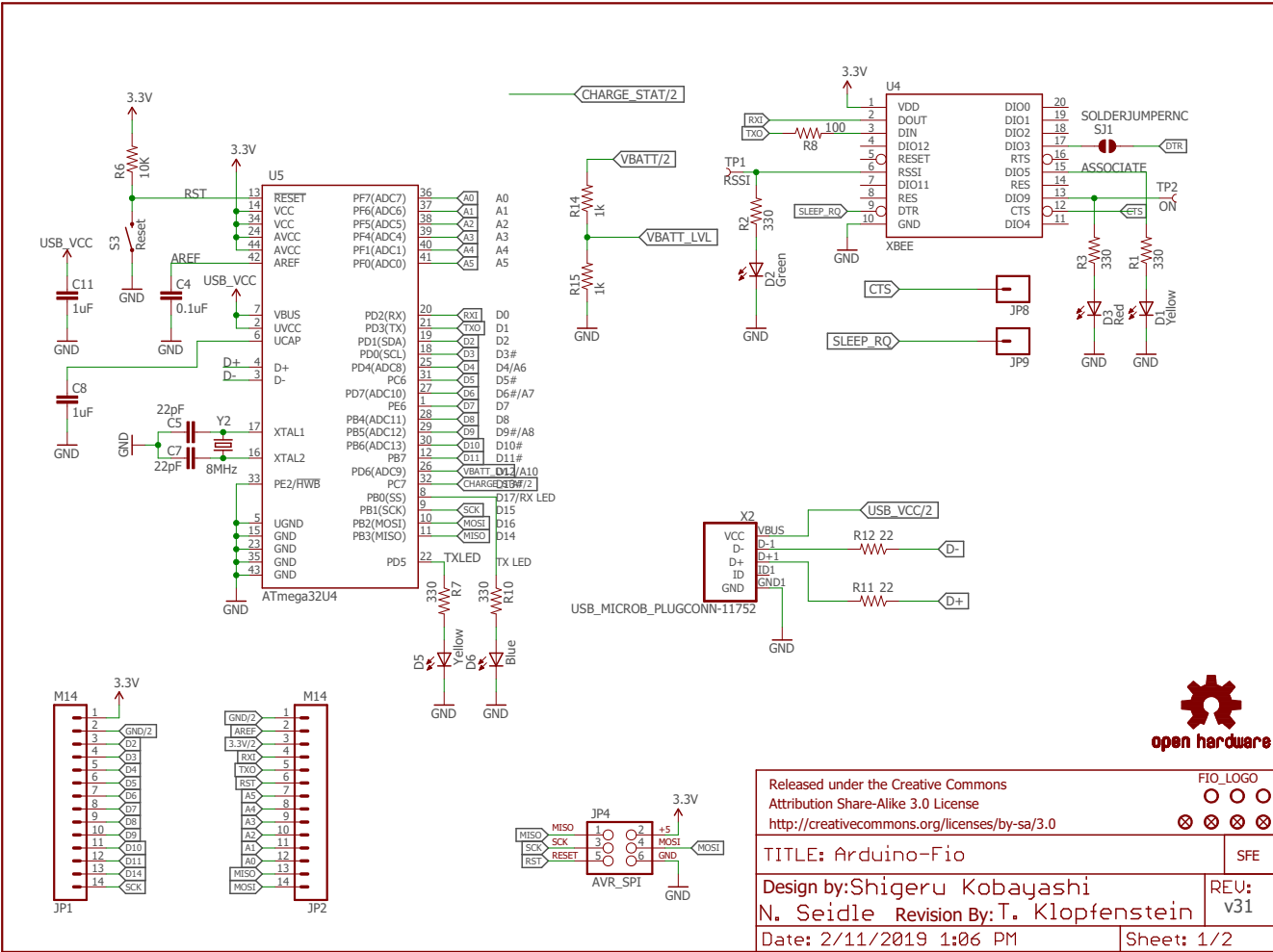
```
Se inicializo el programa:
Very Good, Good, Meh, Bad
65535, 65535, 65535, 0
65535, 65535, 0, 0
65535, 0, 0, 0
0, 0, 0, 0
1, 0, 0, 0
1, 1, 0, 0
1, 2, 0, 0
1, 2, 1, 0
1, 2, 1, 1
1, 2, 2, 1
```

Figura 5: Ejemplo de Salida del Arduino.

Siempre se tienen valores de inicialización, sin embargo, luego de que se pasen todas las funciones de los botones por primera vez, los valores vuelven a cero por si solos. Una vez que todos los valores sean ceros en el puerto serial, o bien, que todos los leds conectados se apaguen, se puede iniciar a utilizar los botones.

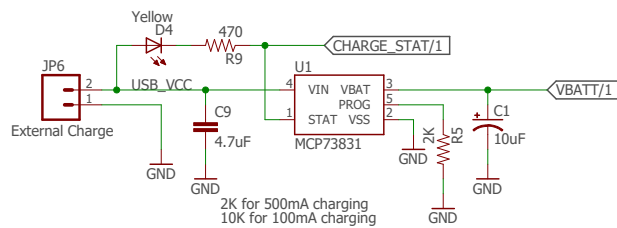
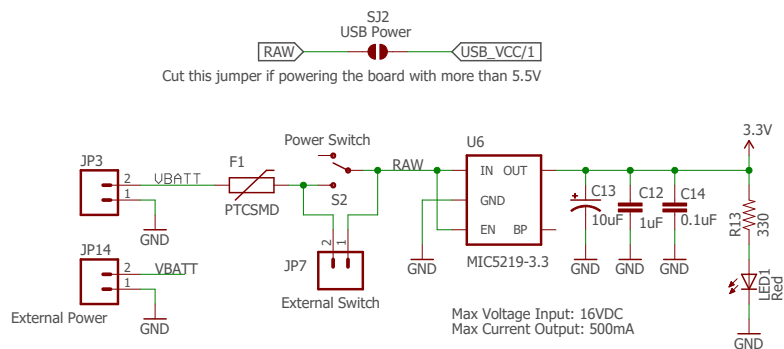
2. Anexos

A continuación se muestra información importante sobre el Fio V3 Sparkfun:



New on v31:
 Change to ATmega32U4
 Switch to microUSB
 Change to 8MHz Xtal
 Add battery voltage detect
 Add Charge Status to ATmega
 Update reset switch
 Removed FTDI Connection
 Re-arranged and re-named SPI pins
 Re-ordered Analog pins
 Changed PWM pin designator to ~
 Changed VCC to 3.3V

New on v31:



Released under the Creative Commons
Attribution Share-Alike 3.0 License
<http://creativecommons.org/licenses/by-sa/3.0>

TITLE: Arduino-Fio

Design by:

REV:

Date: 2/11/2019 1:06 PM

Sheet: 2/2