

Este proyecto está basado en el código abierto de [este enlace](#), todo lo expuesto en este post está basado en él y la idea expuesta.

El precio del proyecto es inferior a 5€ en materiales como dice el autor. Si alguien te la vende mas cara, es por negocio.

Lo primero es comprar los componentes básicos:

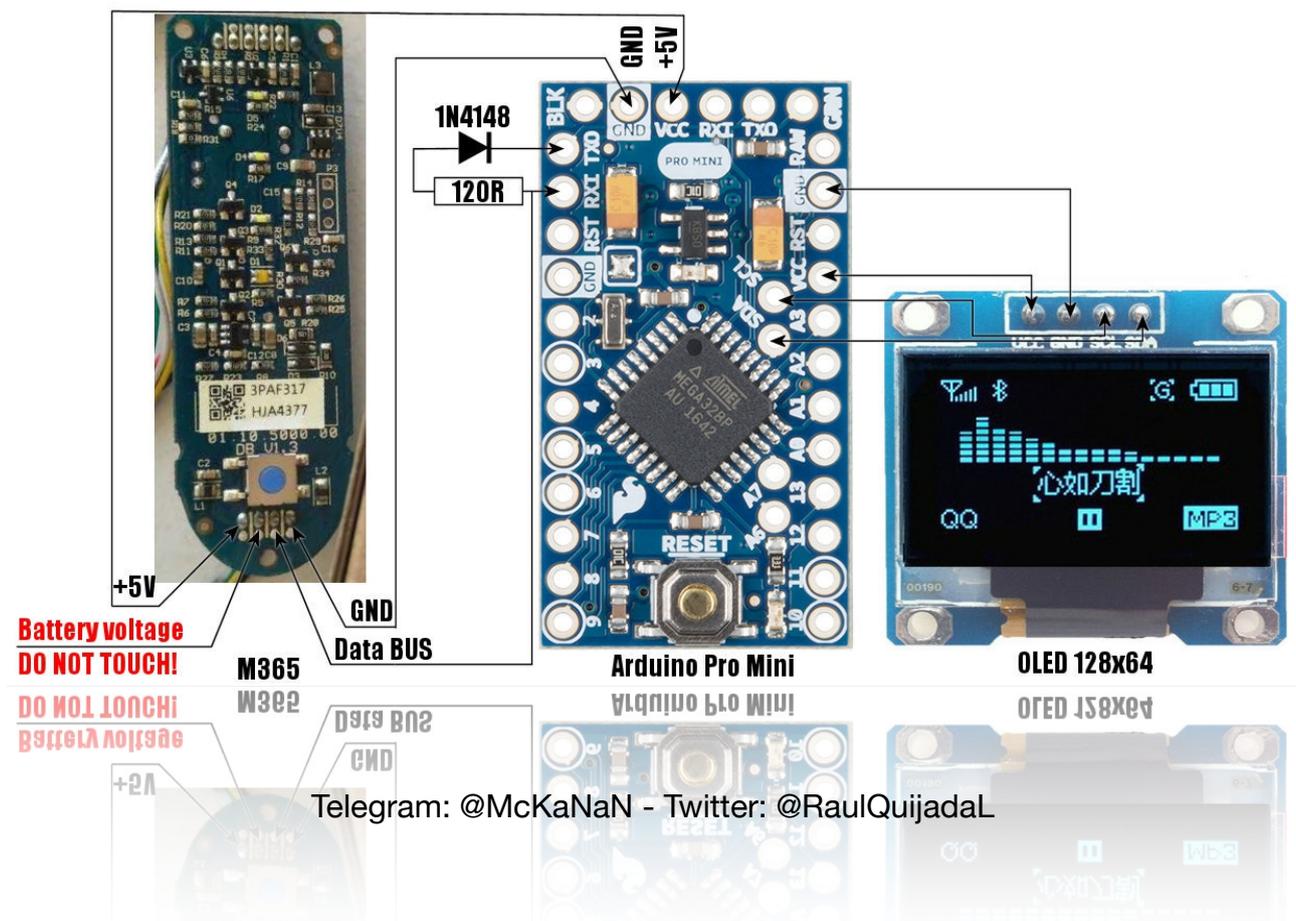
- Pantalla OLED: [Aliexpress](#) - [Amazon](#)
- Arduino Pro Mini: [Aliexpress](#) - [Amazon](#)
- Programador Arduino: [Aliexpress](#) - [Amazon](#)
- Cables para las conexiones: [Aliexpress](#) - [Amazon](#)
- Resistencia 120 Ohm y 0,25 vatios (en la tienda de electrónica de tu ciudad. Cuesta céntimos)
- Diodo 1N4148 (en la tienda de electrónica de tu ciudad. Cuesta céntimos)
- Estañador potente (el mio es de 80W y funciona muy bien)
- Estaño con flux (facilita la tarea muchísimo)

Empezamos por el principio:

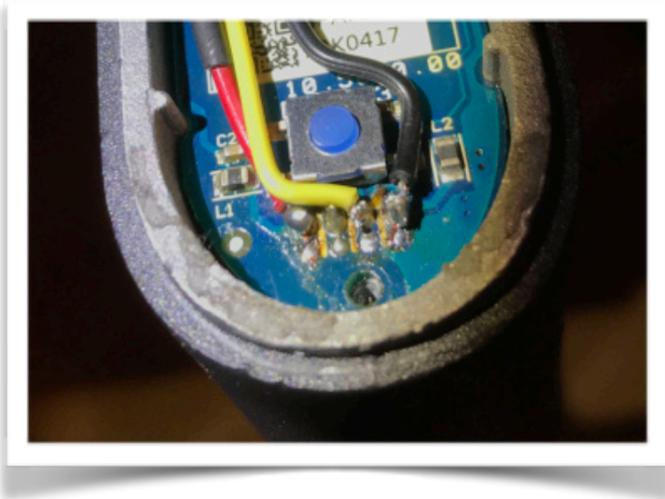
Lo mas importante a tener en cuenta es que este proyecto necesita de un conocimiento minimo de manejo del estañador pues requiere de conexiones pequeñas y en zonas mínimas.

Si no te ves con la capacidad necesaria, busca a alguien de tu ciudad que te puedo echar una mano en este aspecto.

Lo primero es conocer que tenemos que estañar (que al fin y al cabo, es lo mas complicado):



Como indica el esquema superior, hay que tener mucho cuidado de no estañar el 2º pin empezando por la izquierda de la placa Bluetooth del patín. Respecto a realizar los puntos de estaño recomiendo ir con cuidado, utilizar un estañador potente, de punta fina y un pulso de 10 jeje. Sobre todo **cuidar de que las conexiones no toquen otros pines para evitar fallos o problemas.**

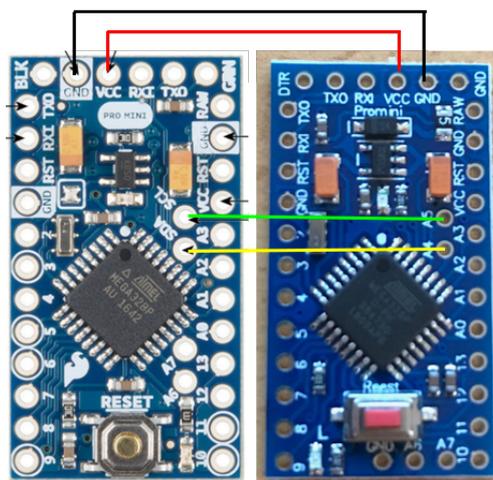


Los cables que he utilizado son:

- Rojo: VCC (positivo)
- Amarillo: Data BUS (datos del patín)
- Negro: GND (negativo)

A continuación vamos con la placa Arduino Mini PRO que a simple vista, es la más "complicada" de preparar por sus pequeños orificios.

A pesar de que el artículo original indica las conexiones debemos tener cuidado con la placa que compremos puesto que la imagen de la pagina web de GitHub es de una placa algo más antigua y algunas conexiones varían:



Placa antigua

Placa nueva

Las conexiones que cambian son las indicadas en el esquema superior: GND, VCC, SCL y SDA.

No debéis preocuparos si la placa viene con otro nombre u otra nomenclatura ya que conexiones van al mismo lugar en el caso de SCL y SDA. En cambio el positivo y negativo que salen de la placa del patín si que hay que ubicarlas donde toca del Arduino (VCC = positivo. GND = negativo).

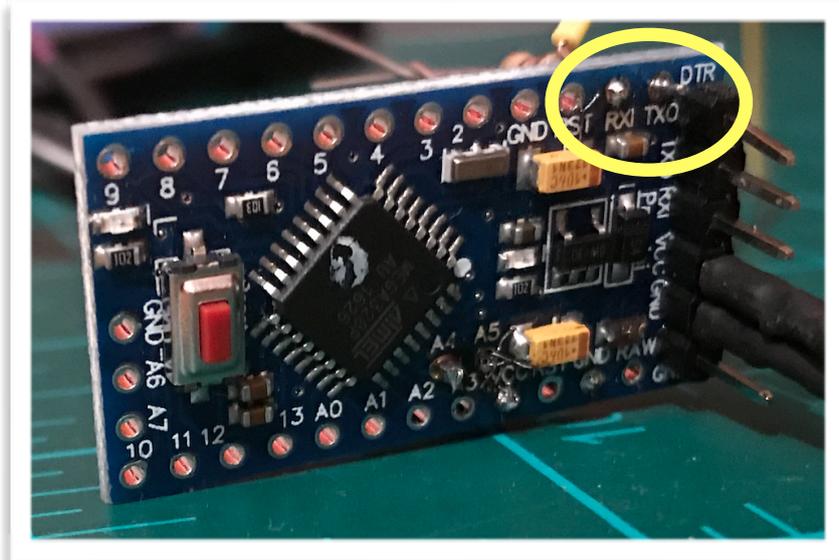
¿Como estañamos? Intentare ser conciso y breve, pero sin dejarme nada importante: para realizar un buen punto de soldadura entre componentes es importante que ambos estén bien calientes, pero teniendo en cuenta que trabajamos con componentes muy delicados no podemos dejar el estañador tocando las superficies demasiado tiempo porque acabarían dañándose (en el caso de la electrónica) o derritiéndose (en el caso de los cables, ya que son de una sección muy estrecha).

Consejo: utilizar flux liquido o en pasta. Esto facilita que el estaño se reparta más rápidamente y asi evitamos una larga exposición al calor de los distintos componentes.

Para estañar los cables a la placa arduino recomiendo seguir los siguientes pasos:

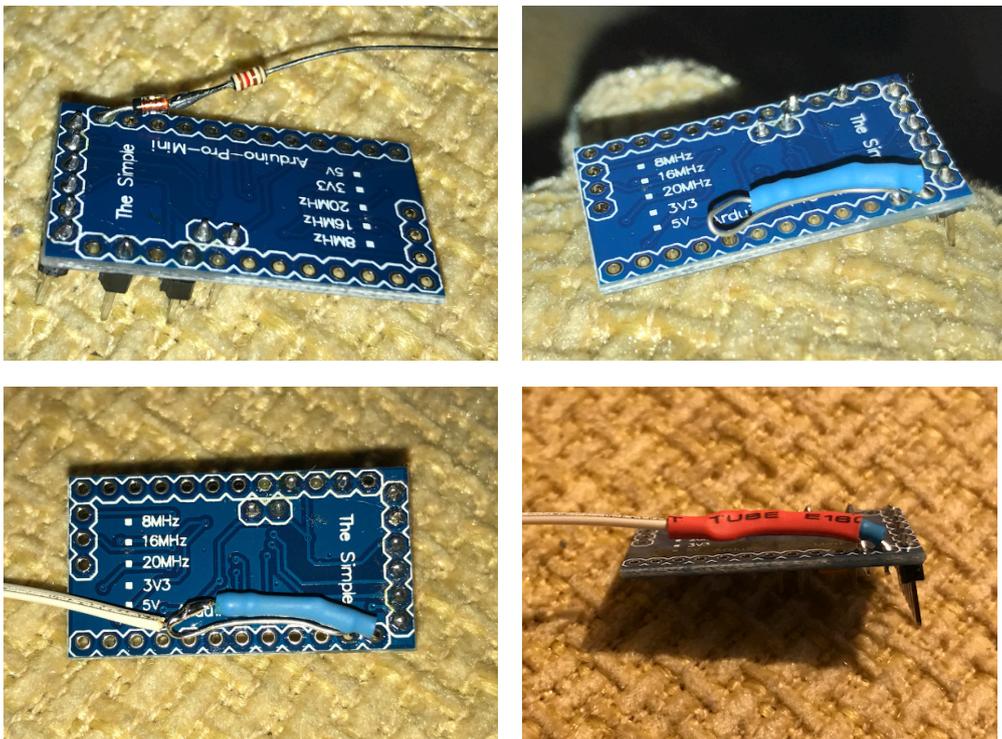
1. Pelar los cables por ambos lados
2. Rociar en flux
3. (Con el estañador totalmente caliente) añadir un poco de estaño. Veremos como este se esparce por la punta del cable rápidamente. Una vez hecho, apartar el estañador.
4. Meter el cable estañado en el agujero de la placa Arduino que queramos estañar
5. Coger con la punta del estañador un poco de estaño
6. Aplicar sobre el cable del Arduino que introducimos en la placa en el punto 4
7. Dejar que el estaño "agarre" la placa y el cable y retiramos el estañador

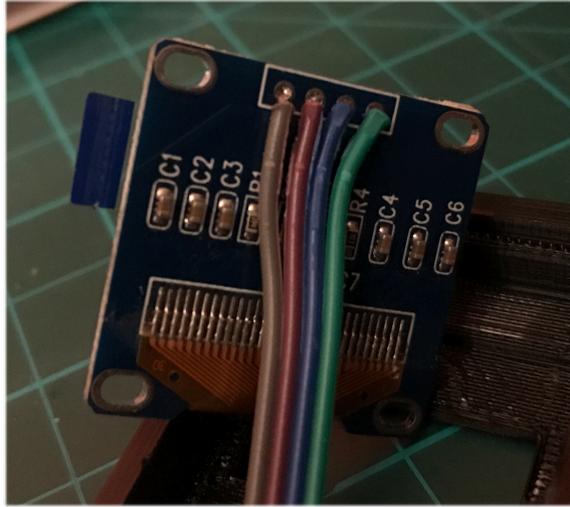
Aquí podéis ver un ejemplo



Requiere de cierta maña, pero si podéis y tenéis paciencia, sale. Es MUY importante que el material que utilizéis sea de calidad. Olvidaros del chino para este tipo de cosas jeje.

Para estañar el diodo, resistencia y cable de datos yo lo hice así:





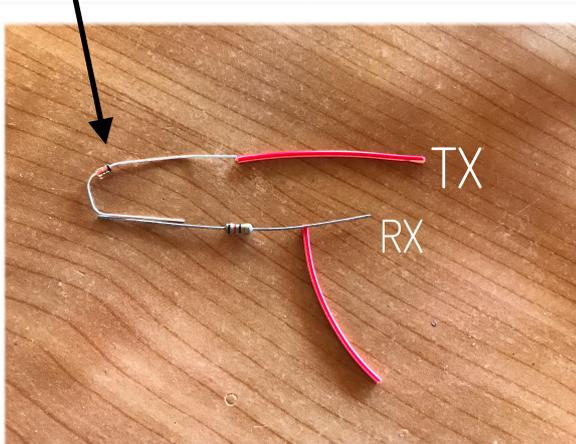
El siguiente paso es realizar las conexiones de la pantalla. Para ello podéis aprovechar el poco estaño que trae la propia pantalla y añadir una gotita más para que agarre como dios manda.

Es importante que en este paso no dejéis el estañador mucho tiempo pegado a la pantalla para evitar dañarla.

En mi caso utilice conectores Dupont ya que la pantalla que me llegó era de color azul y la prefiero en color blanco, que esta más iluminada.

Una vez estañados a la pantalla ya podemos estañar estos a la placa Arduino Pro Mini. RECORDAR que cada pin tiene su conexión. Para ello fijaros en la imagen que esta puesta arriba.

A continuación pasamos a estañar el diodo y resistencia. ¿Como? Mi esquema fue asi:



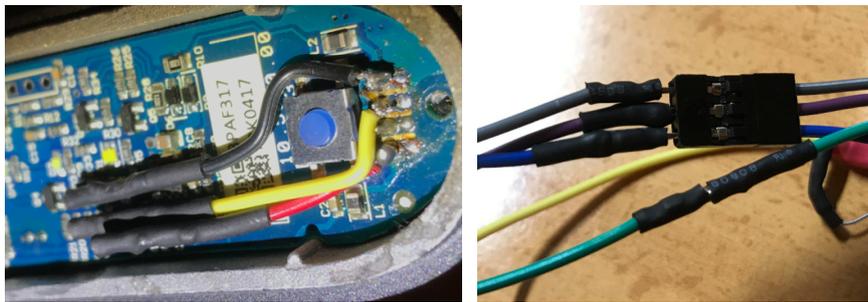
Los orificios TX y RX de la placa Arduino serán ocupados por el diodo y la resistencia. ¿Como? Mi consejo es que primero estañéis el diodo a la resistencia (**OJO!! El diodo tiene posición! fijaros en la imagen: el lado con la linea negra, va al TX de Arduino. Lo señalo con la flecha negra en la imagen**). A continuación el cable de datos de la placa Bluetooth del patín al otro extremo de la resistencia y por ultimo, la

punta de la resistencia al RX y la del diodo al TX de la placa Arduino.

Una vez tenemos estas conexiones hechas, debemos conectar el GND y VCC que sale de la placa bluetooth del patín a la placa de Arduino.

Mi consejo: yo utilice conectores dupont. Primero; porque son fáciles de poner y quitar para poder actualizar la placa cuando sea necesario. Segundo; porque permite probar todo antes de instalarlo en su sitio y en caso de que algo falle, se arregla fácilmente, sin exponer los materiales a largos periodos de calentamiento por parte del estañador.

Aquí podéis ver un ejemplo de los conectores que comento.



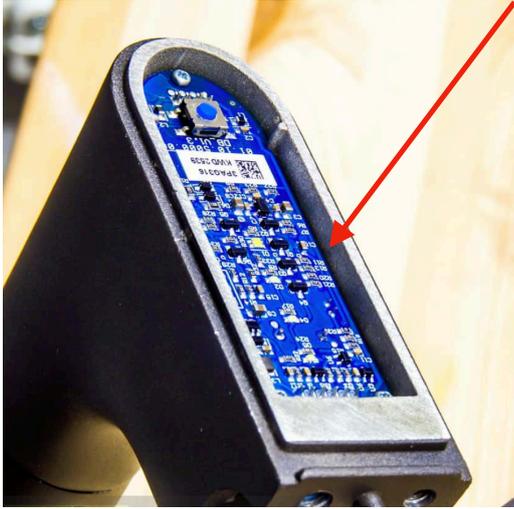
Por último, una vez tengamos la pieza 3D para el montaje, presentamos todo en el patín. Para ello tenemos que quitar la tapa de la botonera y los LED.



Para ello podemos ayudarnos de una pua de guitarra dura o alguna especie de material flexible que nos ayude a hacer palanca para no romper dicha tapa.

Yo lo que he hecho es, con la dremel y una bronca, rebajar el lateral del patín (es un material que se rebaja muy bien) 1mm aproximadamente para que así, al ubicar los cables por ese lateral, la placa bluetooth no me los corte.

Yo he rebajado por donde indica la flecha roja. De esta manera los cables quedan justo en la zona donde ubique la placa de Arduino, justo debajo de esta placa.



La mejor manera de calcular la situación de este rebaje, es presentar todo y viendo lo largo que hayáis dejado los cables y como queráis que vaya ubicada la placa, rebajáis. No necesitáis más de 1mm si usáis cables de baja sección.

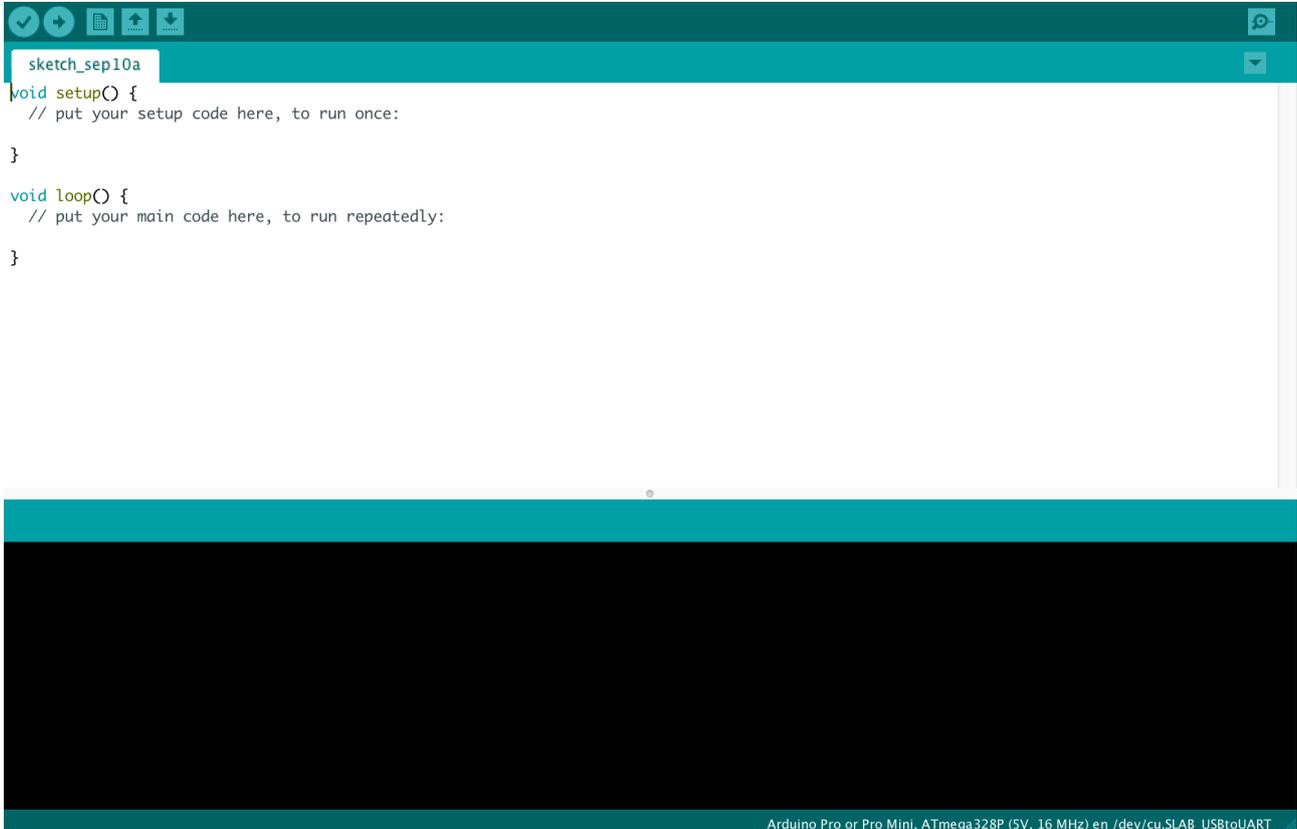
Para finalizar, colocamos la pantalla en la pieza 3D, pegándola a la misma pieza con 4 puntos de silicona en las esquinas de la pantalla y después colocando la pieza 3D en el patín. Recomiendo echarle un par de puntos de silicona para evitar

que con las vibraciones se caiga. No debería, pero más vale prevenir.

Configuración de la placa Arduino Mini PRO

Descargamos el programa de Arduino [pinchando AQUI](#)

Una vez descargado abrimos el programa y nos saldrá la siguiente pantalla:



```
sketch_sep10a
void setup() {
  // put your setup code here, to run once:
}

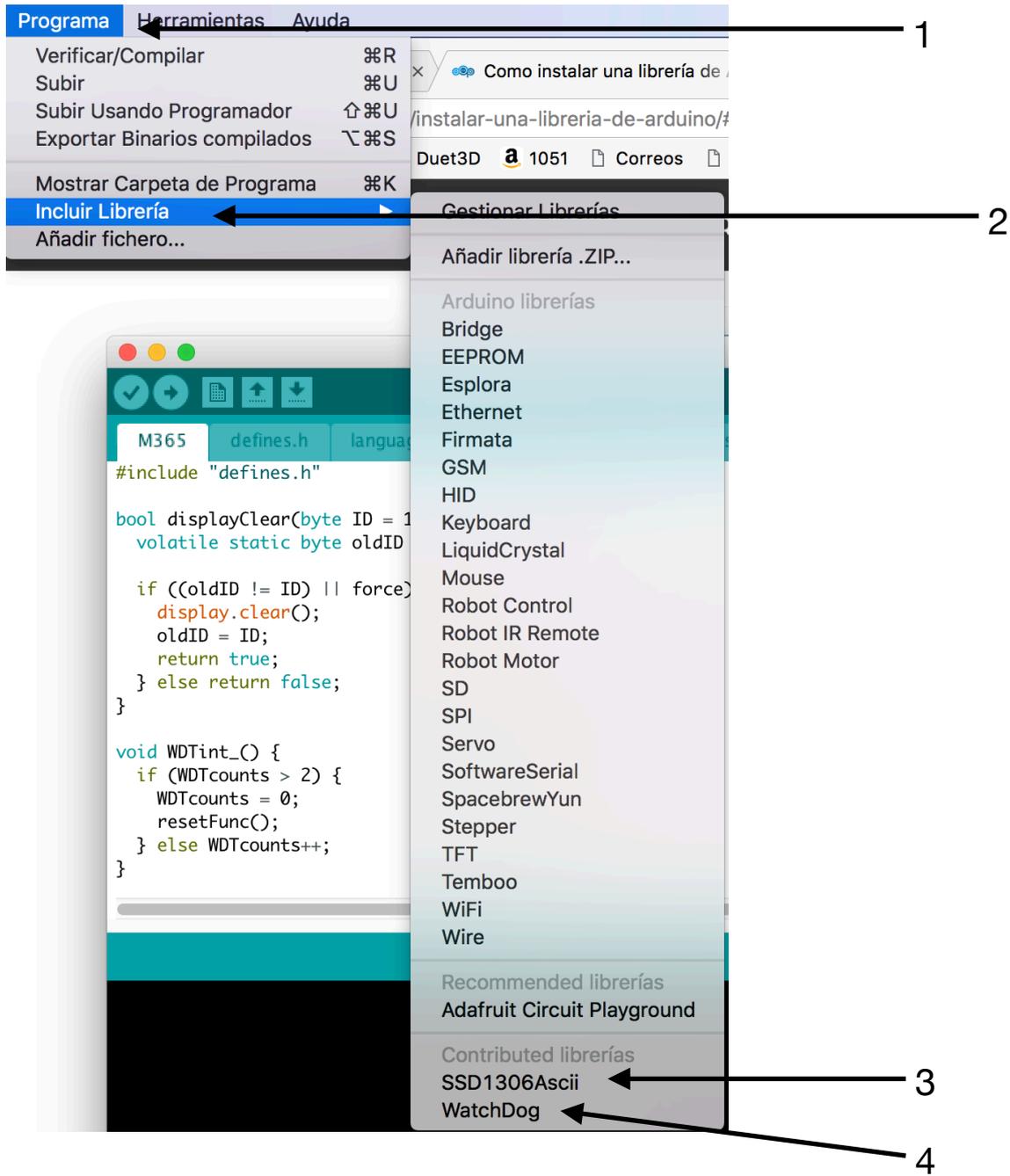
void loop() {
  // put your main code here, to run repeatedly:
}
```

Arduino Pro or Pro Mini, ATmega328P (5V, 16 MHz) en /dev/cu.SLAB_USBtoUART

A continuación descargamos el código del GitHub para la instalación en la placa Arduino -> [DESCARGAR](#)

Descomprimir el archivo RAR e ir a la carpeta "libraries", descomprimir los archivos "SSD1306Ascii-master" y "WatchDog-1.2.0" y copiar esas carpetas en la carpeta de vuestro ordenador: "C:\Usuarios\\Documentos\Arduino\libraries". Las pegamos y abrimos el archivo "M365.ino" ubicado en la carpeta "M365" del RAR descargado anteriormente.

Una vez cargado vamos a la barra de tareas y seleccionamos las librerías copiadas anteriormente:



¿Como sabemos que todo ha sido realizado correctamente? Si al terminar estos pasos nos sale los siguiente en la ventana del programa de Arduino:

```

M365 $ defines.h language.h messages.cpp messages.h
#include <WatchDog.h>

#include <SSD1306Ascii.h>
#include <SSD1306AsciiAvrI2c.h>
#include <SSD1306AsciiSoftSpi.h>
#include <SSD1306AsciiSpi.h>
#include <SSD1306AsciiWire.h>
#include <SSD1306init.h>

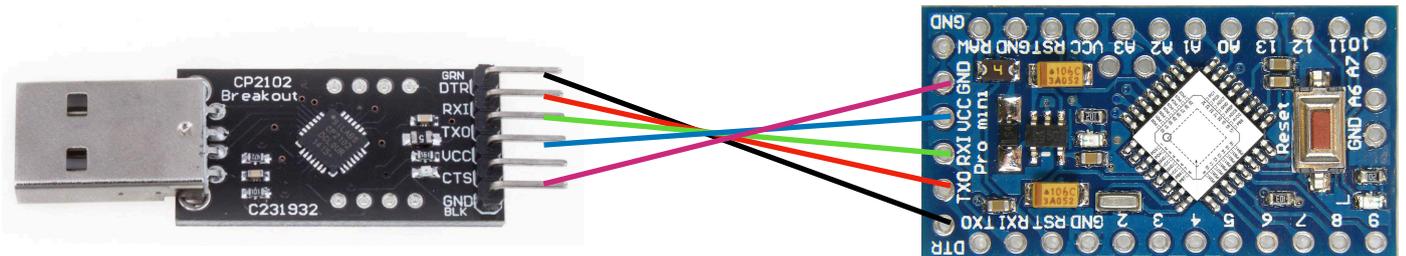
#include "defines.h"

bool displayClear(byte ID = 1, bool force = false) {
  volatile static byte oldID = 0;

  if ((oldID != ID) || force) {
    display.clear();
    oldID = ID;
    return true;
  } else return false;
}
    
```

Estas líneas indican que las librerías han sido incluidas correctamente

Una vez hecho, ya tenemos el código listo para meterlo en la placa. Para ello debemos conectarla al PC mediante un programador tipo "CP2102 Break out" (que es el que yo usé). Las conexiones entre este programador y la placa Arduino Mini PRO son:



El esquema principal de conexión es:

CP2102 BreakOut	Arduino Mini PRO
GRN o DTR	DTR
RXI	TX0
TX0	RXI
VCC	VCC
CTS	¡No conectar!
GND	GND

La placa Arduino Mini PRO funciona con 5V y a la hora de conectarlo al pc será ese voltaje el que le demos desde el USB TTL.

Las otras conexiones (Rx y Tx) ¿qué son? Uno es el transmisor de datos y el otro el receptor. Tenemos que conectarlos cruzados. De esta forma el USB manda datos por el TX y el Arduino los recibe por el Rx y viceversa.

Independientemente de las siglas que use el fabricante las conexiones para pasar los programas por USB son:

- GND a GND
- 5 V ó VCC a VCC ó 5 V
- R... a T...
- T... a R...

Una vez las conexiones están hechas y la placa conectada al ordenador, nos aseguramos de que los siguientes parámetros estén configurados:



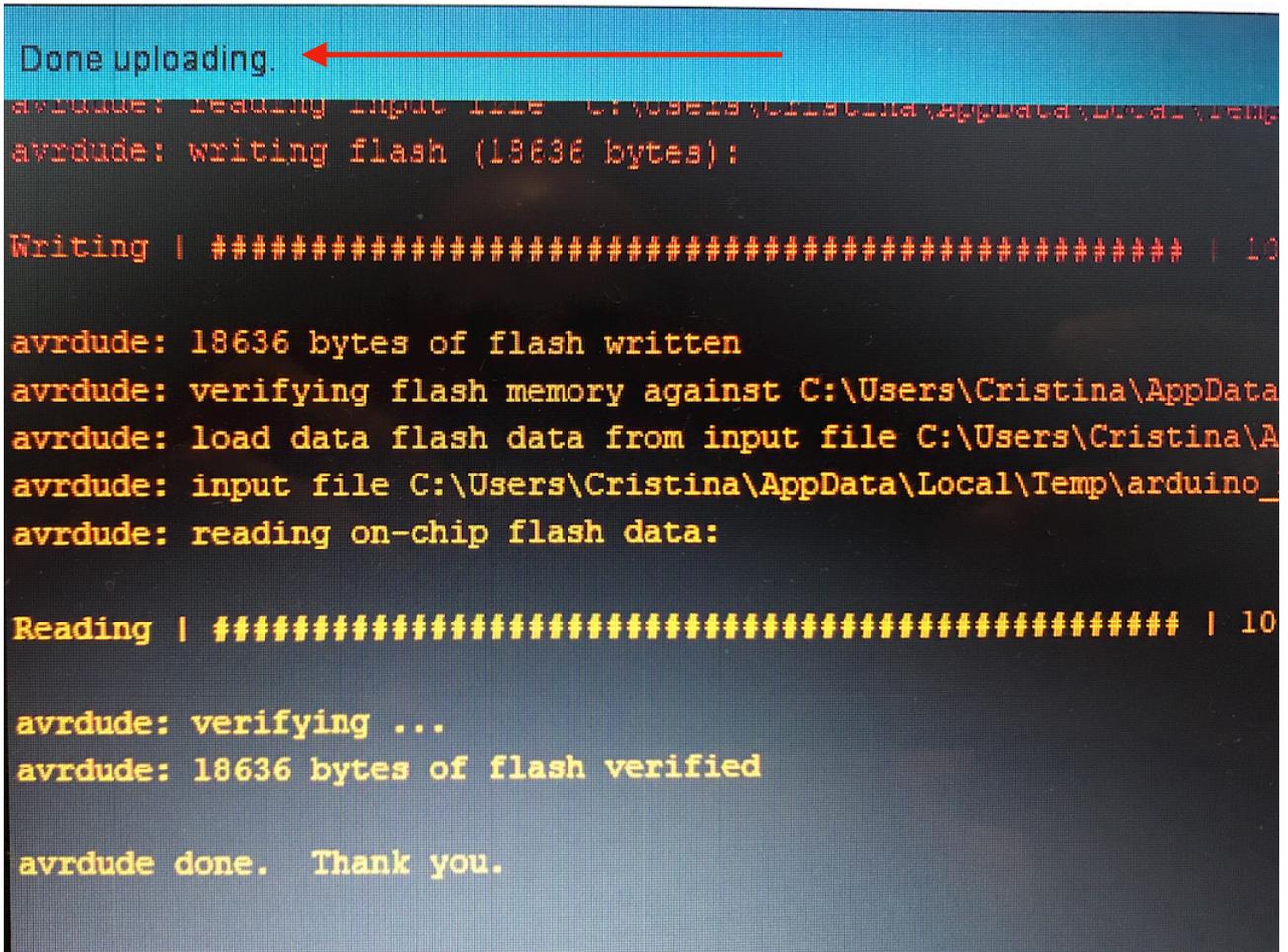
Seleccionar el puerto USB al que tengáis conectada la placa

Por último le damos al botón de "Subir" el código a la placa:



Seleccionar el puerto USB al que tengáis conectada la placa

Cuando el código haya sido cargado correctamente, el programa de Arduino nos mostrará un mensaje que pone "Subido" en la barra verde que esta encima de la consola de comandos.



```
Done uploading.
avrduide: reading input file C:\Users\Cristina\AppData\Local\Temp\arduino_
avrduide: writing flash (18636 bytes):

Writing | ##### | 10

avrduide: 18636 bytes of flash written
avrduide: verifying flash memory against C:\Users\Cristina\AppData
avrduide: load data flash data from input file C:\Users\Cristina\A
avrduide: input file C:\Users\Cristina\AppData\Local\Temp\arduino_
avrduide: reading on-chip flash data:

Reading | ##### | 10

avrduide: verifying ...
avrduide: 18636 bytes of flash verified

avrduide done. Thank you.
```

Resolución de dudas:

- *¿Es mejor soldar los cables directamente o utilizar conectores Dupont?*

Yo lo hice con conectores Dupont por lo que expuse más arriba: facilidad de conexión y en caso de querer actualizar, facilidad de montaje/desmontaje.

Por otro lado, los conectores Dupont para los cables ayudan a que estos no se quemem con el estañador. Recomiendo utilizar una crimpadora para realizar la instalación con este tipo de conectores.

- *Al conectar la pantalla me da un error que pone: "BUS not connected! No data to display!"*

Revisar que todas las conexiones sean correctas: diodo, resistencia, cable de datos de la placa bluetooth...

A mi me pasó algo curioso: dos resistencias (aparentemente) iguales no me dieron el mismo resultado. Una de ellas es más grande que la otra, pero tienen el mismo código de líneas y color. La que me funcionó fue la más pequeña. Quizás entran en juego un tema de tolerancias...

Asegúrate que la resistencia es de 120 y 0,25W. Este último dato es vital para corroborar que los componentes son los adecuados.

Otra solución aportada por "Jam":

En la pestaña "Define.h" de Arduino cambiamos el código que trae por el que contiene el siguiente enlace: <https://paste.ofcode.org/ehi9hPmg2rRYZTWZyYSdML>

Ahora tenemos que tener en cuenta que lo que antes iba en el orificio Tx ahora va en el 4 y lo que iba en el Rx ahora va en la 3.

- Si sale el error *"exit status 1 'm365' was not declared in this scope"* al compilar asegúrense que las librerías no se encuentran descomprimidas en ninguna otra carpeta del ordenador ya que de ser así, al intentar compilar el código y subirlo a Arduino el programa busca en todo el ordenador las librerías y puede coger la que no debe de donde no debe. (Gracias a @Mostar_jadro por el reporte)