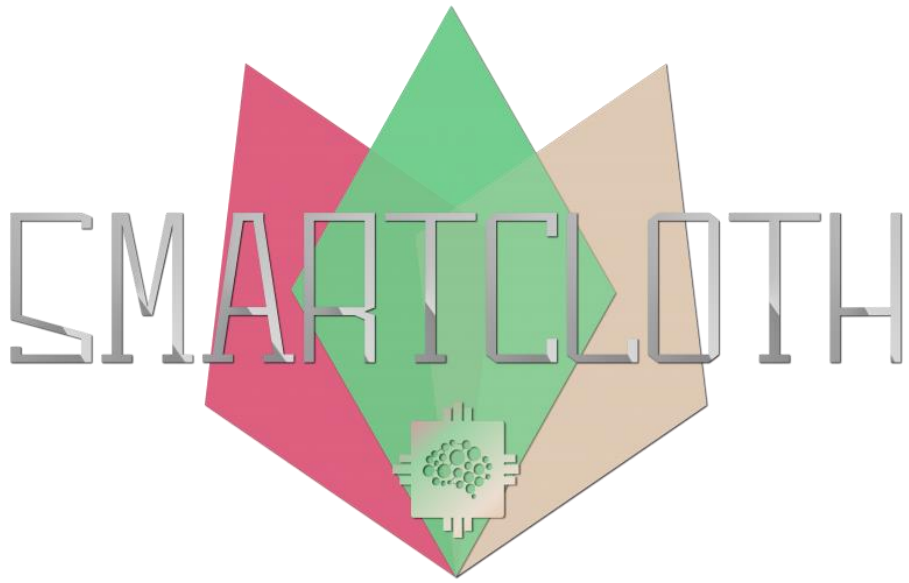
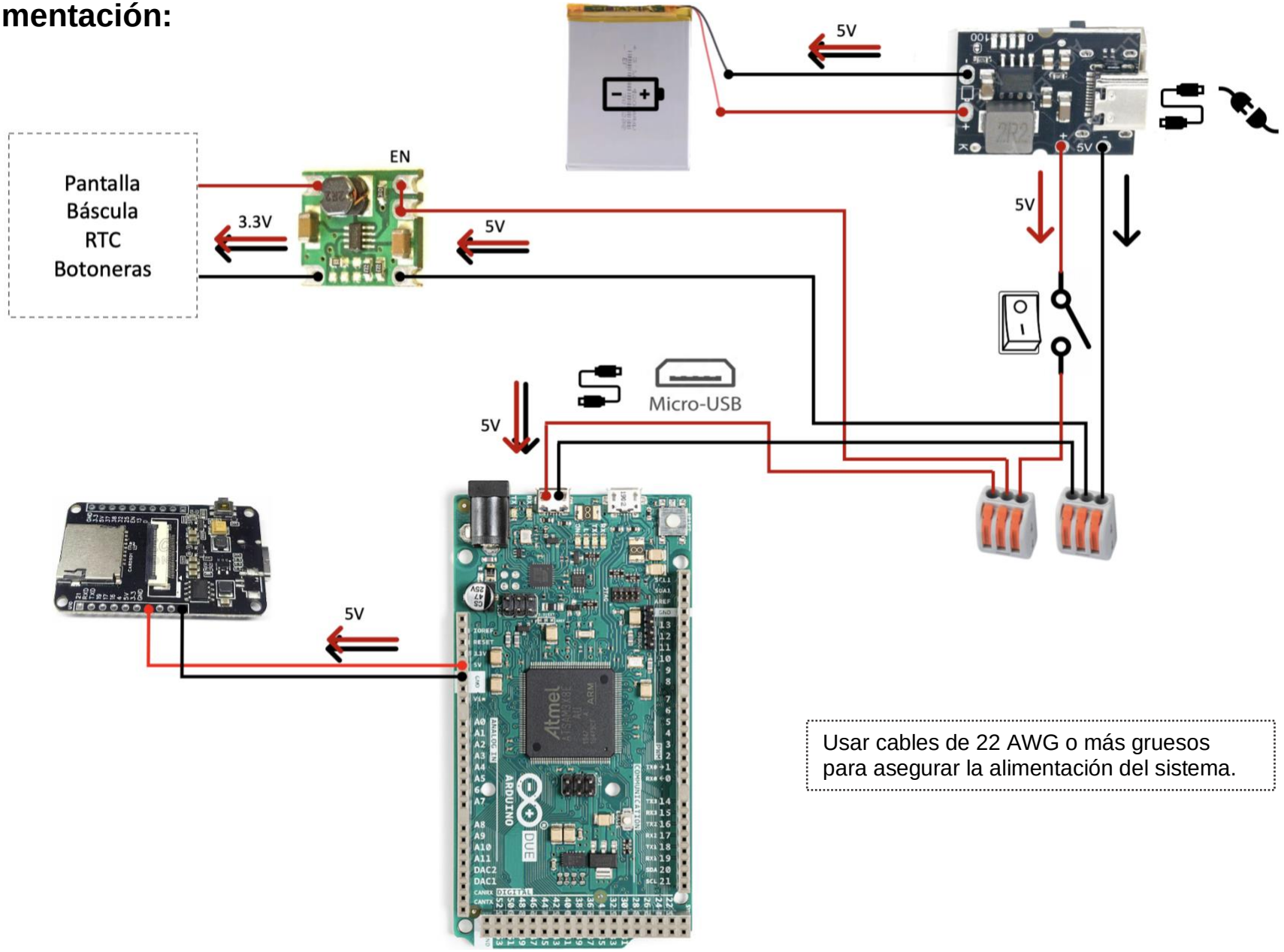
DOCUMENTO DE MONTAJE

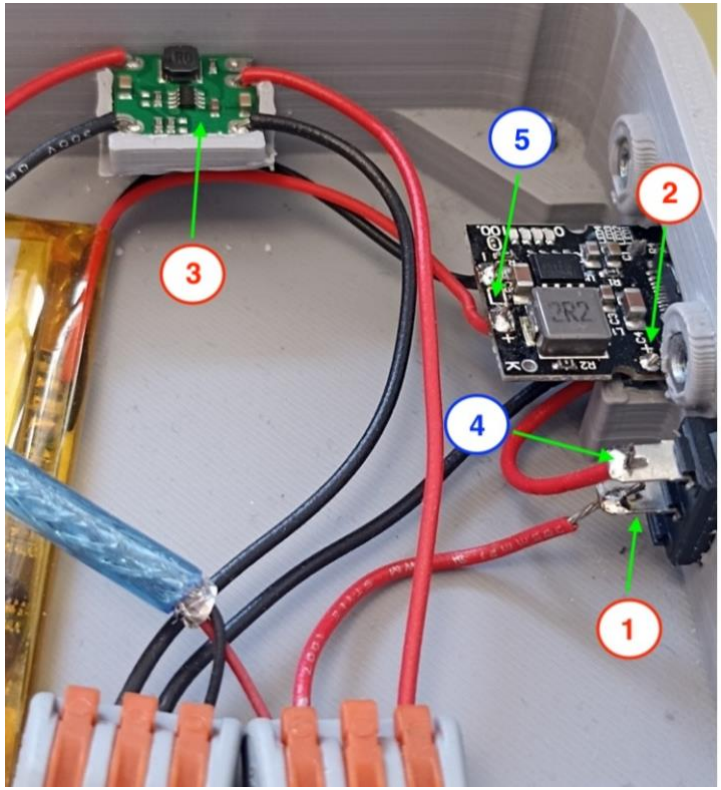


COMPONENTES PRINCIPALES	
Pantalla ER-TFM070-6	Batería
Tarjeta SD	Interruptor basculante
Arduino Due	Módulo Step-Down
Screw Shield	Botonera A (grupos de alimentos)
RTC	Botonera B (acciones)
Celda de carga	Placas de adaptación de botoneras
Conversor AC/DC HX711	ESP32-CAM
Módulo USB cargador de batería	Botón código de barras
COMPONENTES AUXILIARES	
Cable USB-microUSB	4 fichas de 5 cables
4 pegatinas con bridas	5 fichas de 3 cables

PROCEDIMIENTO DE MONTAJE:

1. Alimentación:





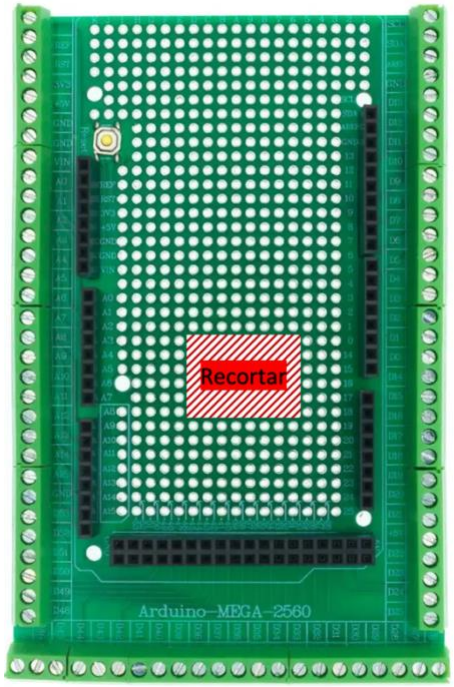
CABLES 22 AWG (soldados o conectados a Fichas) (**R: rojo**; **N: negro**)
Interruptor → Ficha de 3: **1R x 6cm**
Cargador USB → Interruptor: **1R x 4cm**
Cargador USB → Ficha de 3: **1N x 6cm**
Step-down (5V) → Fichas de 3: **1R x 10cm** y **1N x 10cm**
Step-down (3.3V) → Fichas de 5: **1R x 20cm** y **1N x 20cm**
Fichas de 5 (arriba) → Fichas de 5 (abajo): **1R x 35cm** y **1N x 35cm**
CABLE USB
Arduino (puerto junto a conector jack) → Fichas de 3: **cable USB** sin cabezal USB-A

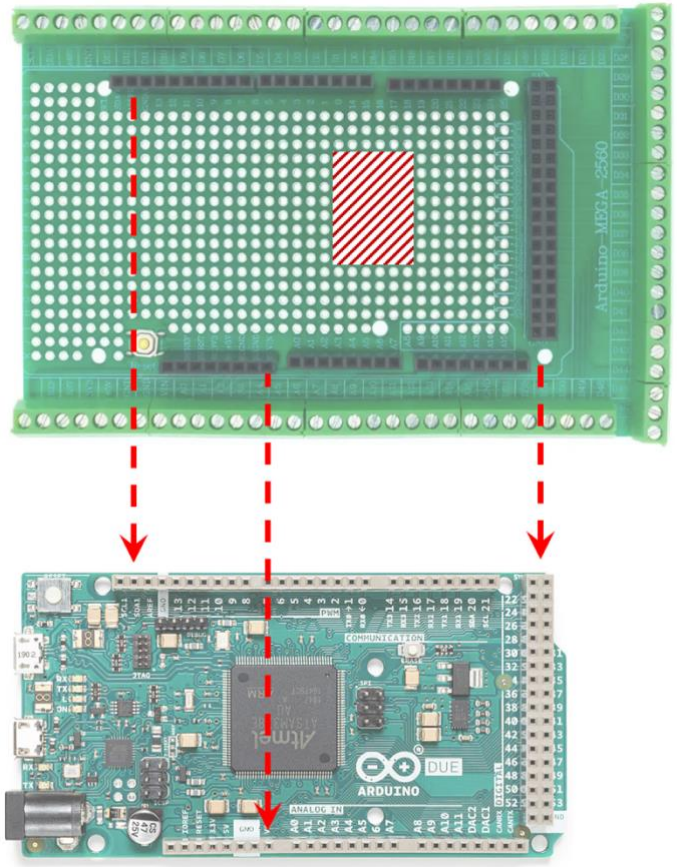
- Soldar **antes** de montar:
  - **(1)** Cable **R** del interruptor a la ficha de 3, en el borne del **|** para que el **O** quede arriba y sea más fácil soldar después.
  - **(2)** Cables **R** y **N** del cargador USB (donde pone 5V) al interruptor (R) y a la ficha de 3 (N), pero solo soldar el extremo del cargador USB para poderlo colocar desde fuera de la carcasa.
  - **(3)** Regulador step-down completo (**R**-**N** en entrada de 5V y en salida de 3.3V). Soldar **EN** con **Vi+** en la entrada.

- Soldar **después** de montar:
  - **(4)** Cable **R** del cargador USB al interruptor, el extremo que va al borne con **O** del interruptor.
  - **(5)** Cables **R** y **N** de la batería en el cargador USB (donde el símbolo de batería).

2. Conexiones:

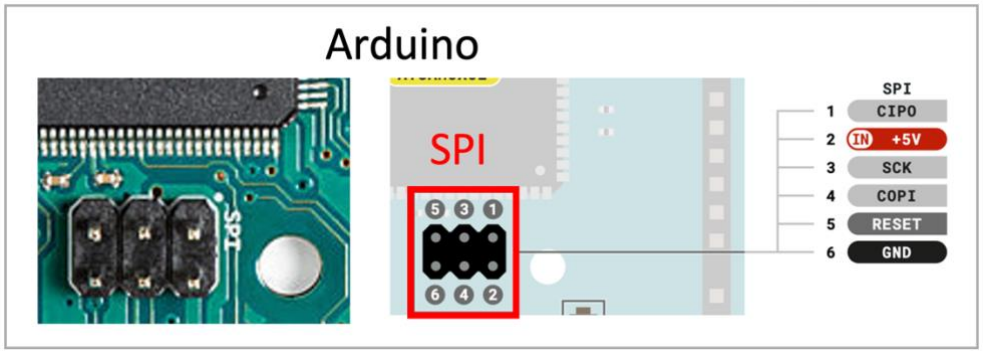
2.1 Shield de tornillos



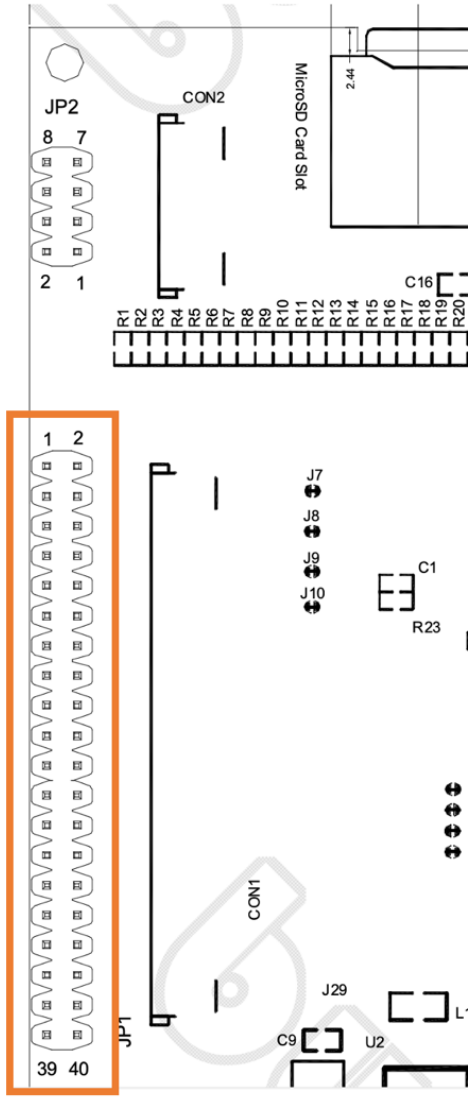


2.2 Pantalla:

CABLES DUPONT HEMBRA (R: rojo; N: negro; AM: amarillo; NA: naranja; V: verde; AZ: azul; B: blanco; M: morado)	
Pantalla (JP1) - VCC (3.3 V)	**Pantalla (JP1) - GND**
Pines 3 y 4 → Ficha de 5 (3.3V): **2R x 15cm**	Pines 1, 2 y 13 → Ficha de 5 (GND): **3N x 13cm**
Pines 37 y 38 → Ficha de 5 (3.3V): **2R x 17cm**	Pines 31, 39 y 40 → Ficha de 5 (GND): **3N x 15cm**
Pantalla (JP1) - SPI	**Arduino Due - SPI**
Pin 7 (MOSI) → Ficha de 3 (SPI): **1AM x 17cm**	Ficha de 3 (SPI) → Pin 4 (SPI): **1AM x 20cm**
Pin 6 (MISO) → Ficha de 3 (SPI): **1NA x 17cm**	Ficha de 3 (SPI) → Pin 1 (SPI): **1NA x 20cm**
Pin 8 (CLK) → Ficha de 3 (SPI): **1V x 17cm**	Ficha de 3 (SPI) → Pin 3 (SPI): **1V x 20cm**
Pantalla (JP1) - Señales	**Arduino Due - Señales**
Pin 5 (SS): **1AZ x 20cm**	Pin 12 (PWM)
Pin 11 (Reset): **1B x 20cm**	Pin 11 (PWM)
Pin 14 (Backlight): **1M x 20cm**	Pin 10 (PWM)

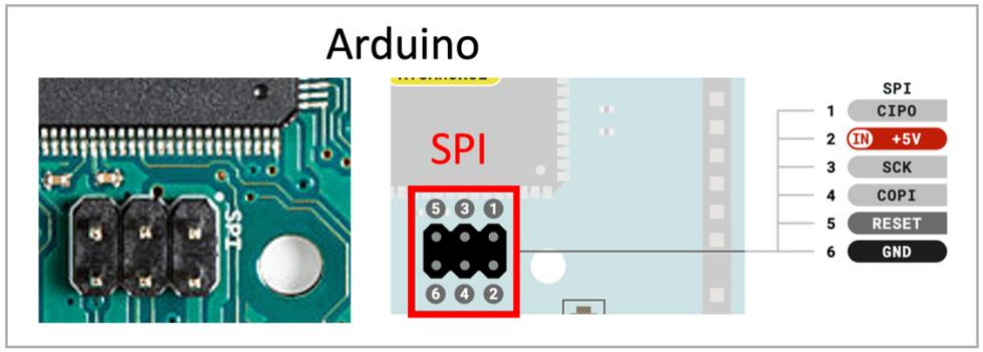


Pantalla (JP1)

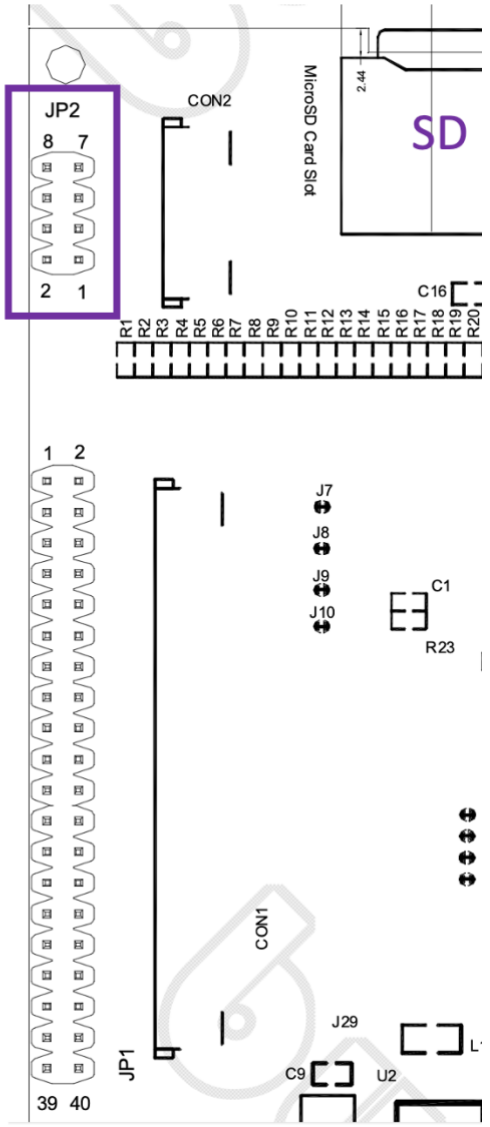


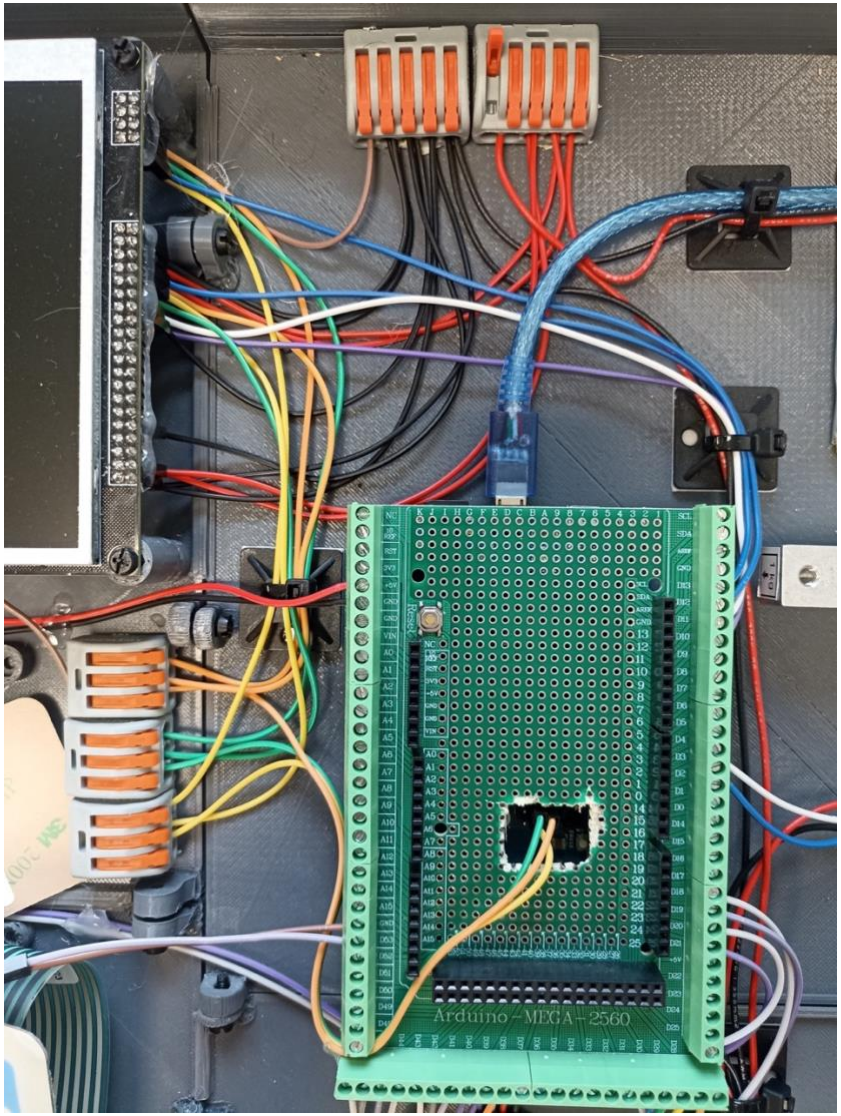
2.3 Tarjeta microSD:

CABLES DUPONT HEMBRA **(N: negro; AM: amarillo; NA: naranja; V: verde; AZ: azul)**	
microSD (JP2) - GND	
Pin 5 → Ficha de 5 (GND): **1N x 10cm**	
microSD (JP2) - SPI	**Arduino Due - SPI**
Pin 3 (MOSI) → Ficha de 3 (SPI): **1AM x 20cm**	Ficha de 3 (SPI) → Pin 4 (SPI): **1AM x 20cm**
Pin 6 (MISO) → Ficha de 3 (SPI): **1NA x 20cm**	Ficha de 3 (SPI) → Pin 1 (SPI): **1NA x 20cm**
Pin 4 (CLK) → Ficha de 3 (SPI): **1V x 20cm**	Ficha de 3 (SPI) → Pin 3 (SPI): **1V x 20cm**
microSD (JP1) - Señales	**Arduino Due - Señales**
Pin 2 (SS): **1AZ x 20cm**	Pin 13 (PWM)



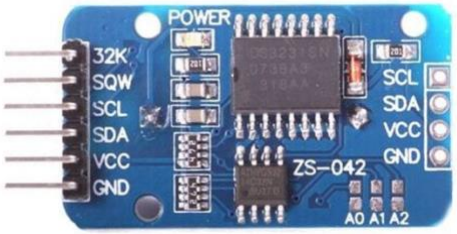
SD (JP2)




Arduino MEGA 2560

2.4 RTC:

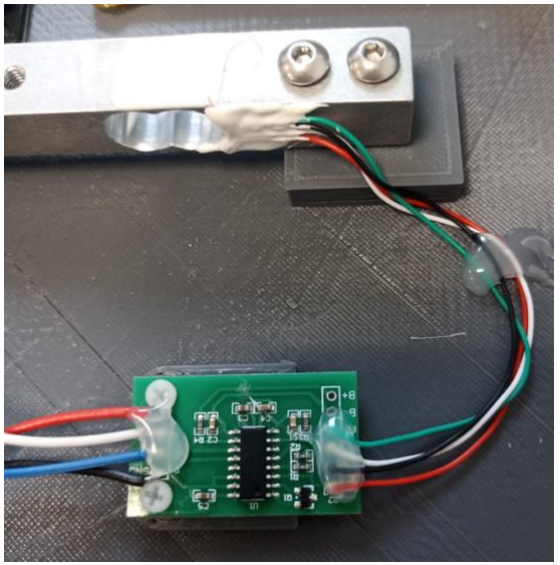
CABLES DUPONT HEMBRA (R: rojo; N: negro; B: blanco; M: morado)	
RTC - Alimentación	
VCC → Ficha de 5 (3.3V): **1R x 18cm**	
GND → Ficha de 5 (GND): **1N x 18cm**	
RTC – I2C	**Arduino Due – I2C**
SDA: **1B x 12cm**	Pin 20 (SDA)
SCL: **1M x 12cm**	Pin 21 (SCL)



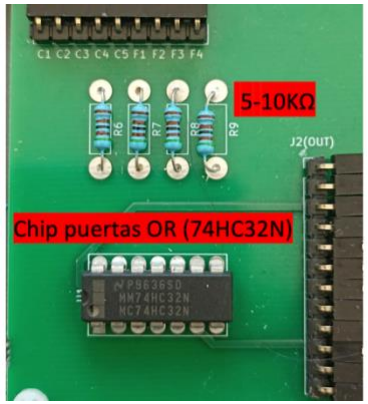
2.5 Celda de carga + HX711:

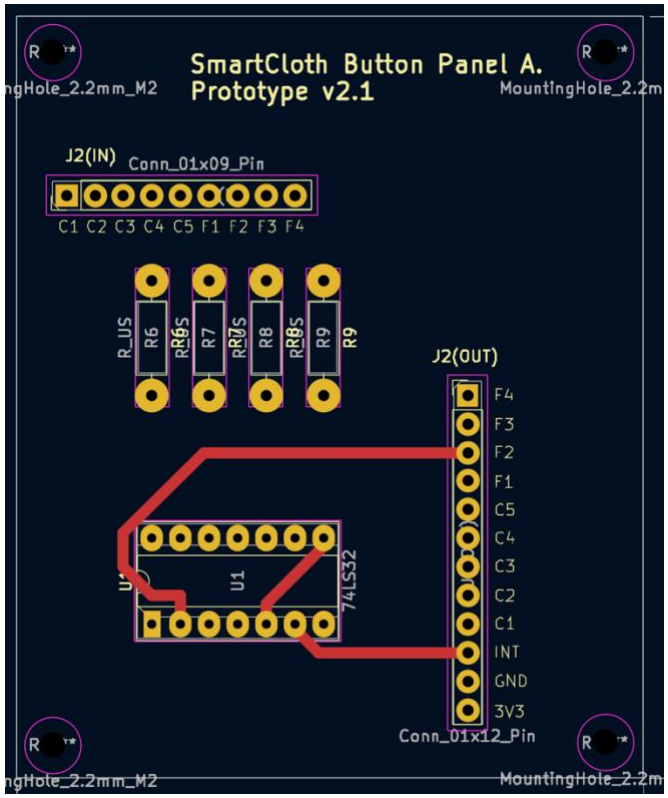
Célula/celda de carga	HX711
Cable **rojo**: VCC (E+)	E+
Cable **negro**: GND (E-)	E-
Cable **blanco**: Output – (S-)	A-
Cable **verde**: Output + (S+)	A+

CABLES 22 AWG (soldados) (R: rojo; N: negro; B: blanco; AZ: azul)	
HX711 - Alimentación	
VCC → Ficha de 5 (3.3V): **1R x 20cm**	
GND → Ficha de 5 (GND): **1N x 20cm**	
HX711 – Señales	**Arduino Due – Señales**
SCK: **1B x 7cm**	Pin 3 (PWM)
SCL: **1AZ x 7cm**	Pin 2 (PWM)



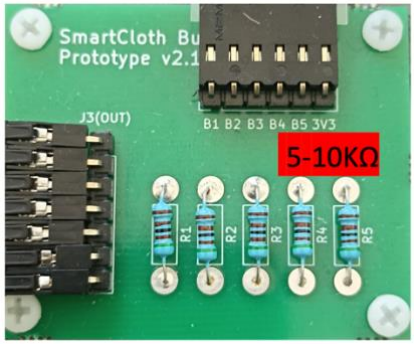
2.6 Botonera A + placa adaptación:

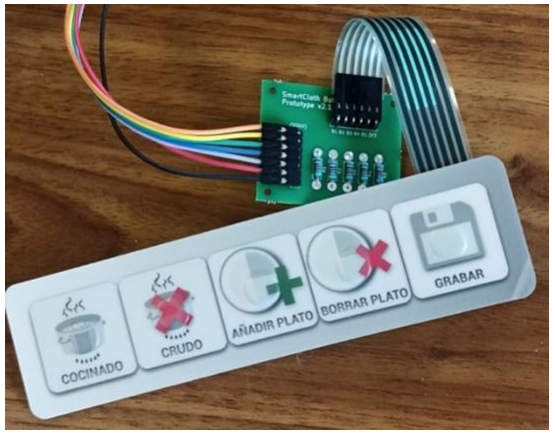


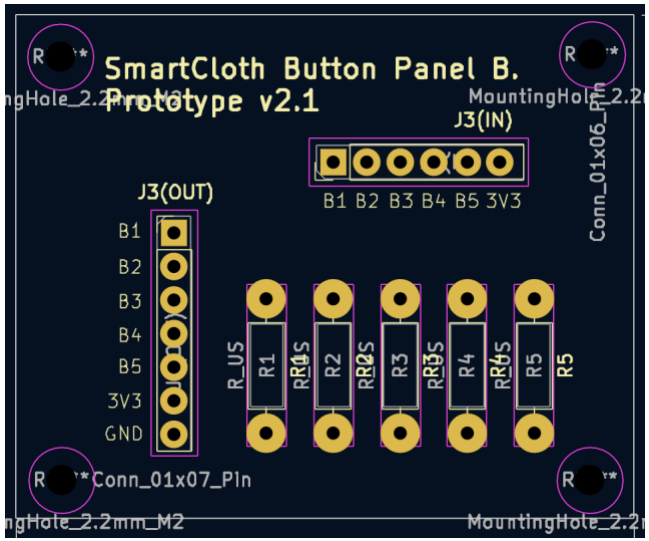


CABLES DUPONT HEMBRA (R: rojo; N: negro; C: cualquiera excepto R o N)	
Botonera A (J2 – OUT) - Señales	**Arduino Due - Señales**
F4: 1C x 11cm	Pin 47 (digital)
F3: 1C x 11cm	Pin 46 (digital)
F2: 1C x 11cm	Pin 45 (digital)
F1: 1C x 11cm	Pin 44 (digital)
C5: 1C x 14cm	Pin 43 (digital)
C4: 1C x 14cm	Pin 42 (digital)
C3: 1C x 14cm	Pin 41 (digital)
C2: 1C x 14cm	Pin 40 (digital)
C1: 1C x 14cm	Pin 39 (digital)
INT: 1C x 14cm	Pin 38 (digital)
Botonera A (J2 – OUT) - Alimentación	
3V3 → Ficha de 5 (3.3V): **1R x 15cm**	
GND → Ficha de 5 (GND): **1N x 18cm**	

2.7 Botonera B + placa adaptación:



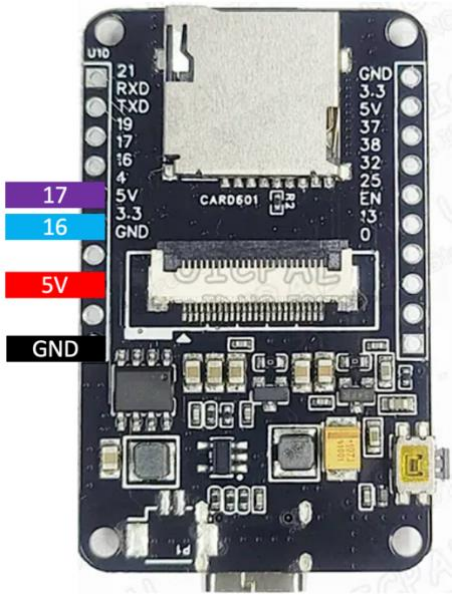


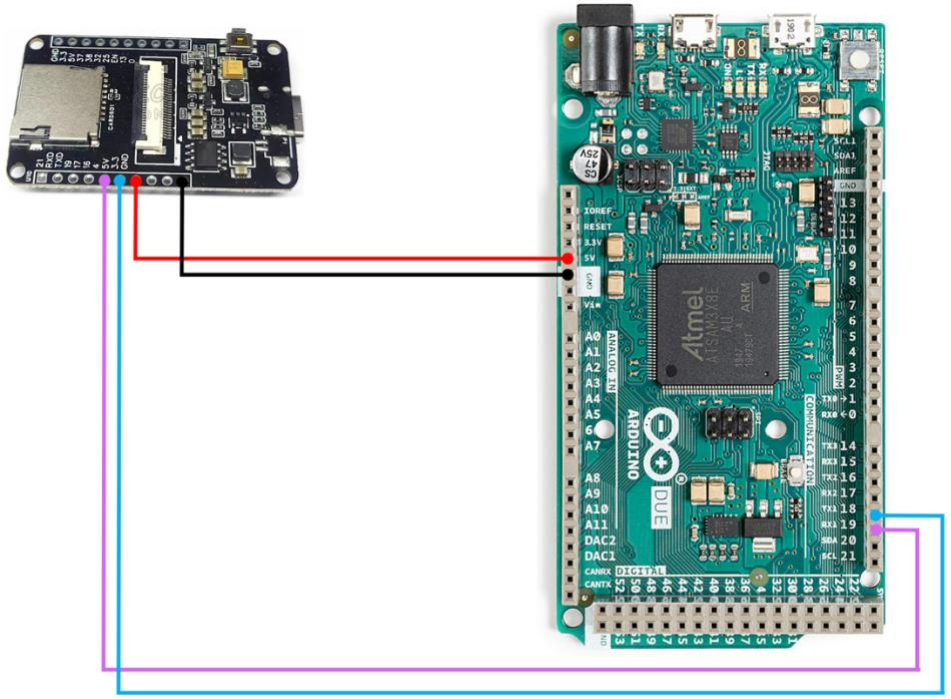


CABLES DUPONT HEMBRA (R: rojo; N: negro; C: cualquiera excepto R o N)	
Botonera B (J3 – OUT) - Señales	**Arduino Due - Señales**
B1: 1C x 13cm	Pin 26 (digital)
B2: 1C x 13cm	Pin 27 (digital)
B3: 1C x 13cm	Pin 28 (digital)
B4: 1C x 13cm	Pin 29 (digital)
B5: 1C x 13cm	Pin 30 (digital)
Botonera B (J3 – OUT) - Alimentación	
3V3 → Ficha de 5 (3.3V): 1R x 10cm	
GND → Ficha de 5 (GND): 1N x 7cm	

2.8 ESP32-CAM:

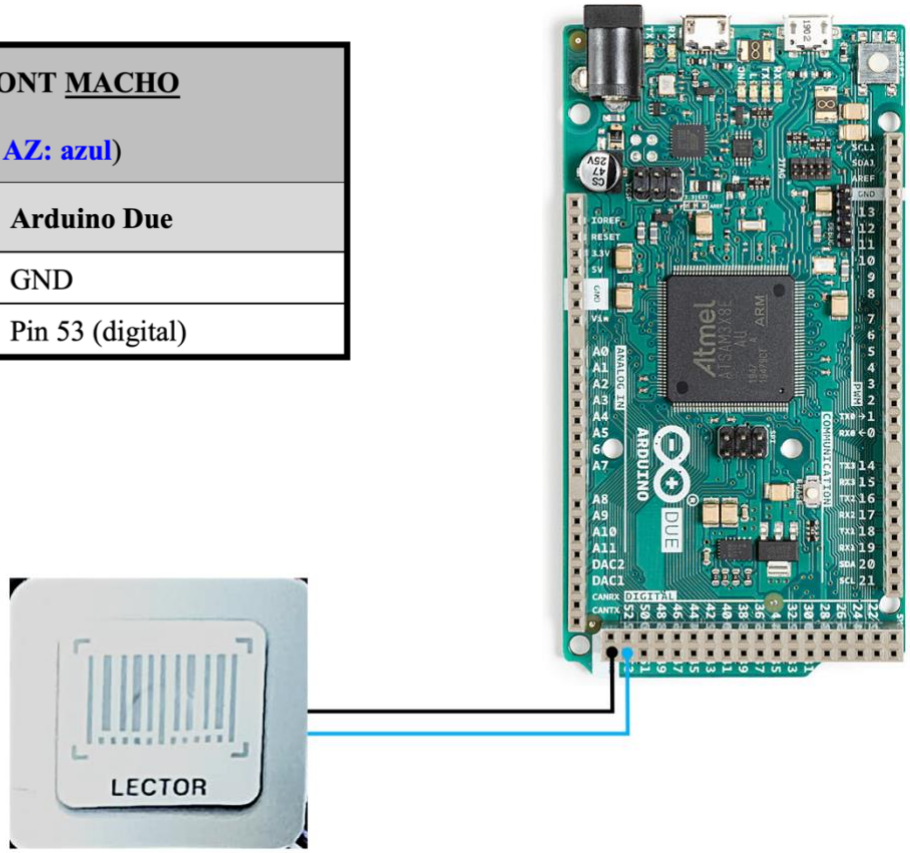
CABLES DUPONT HEMBRA (R: rojo; N: negro; M: morado; AZ: azul)	
ESP32-CAM	**Arduino Due (Serial1)**
IO17 (Tx1): **1M x 34cm**	Pin 19 (Rx1)
IO16 (Rx1): **1AZ x 34cm**	Pin 18 (Tx1)
VCC: **1R x 20cm**	5 V
GND: **1N x 20cm**	GND

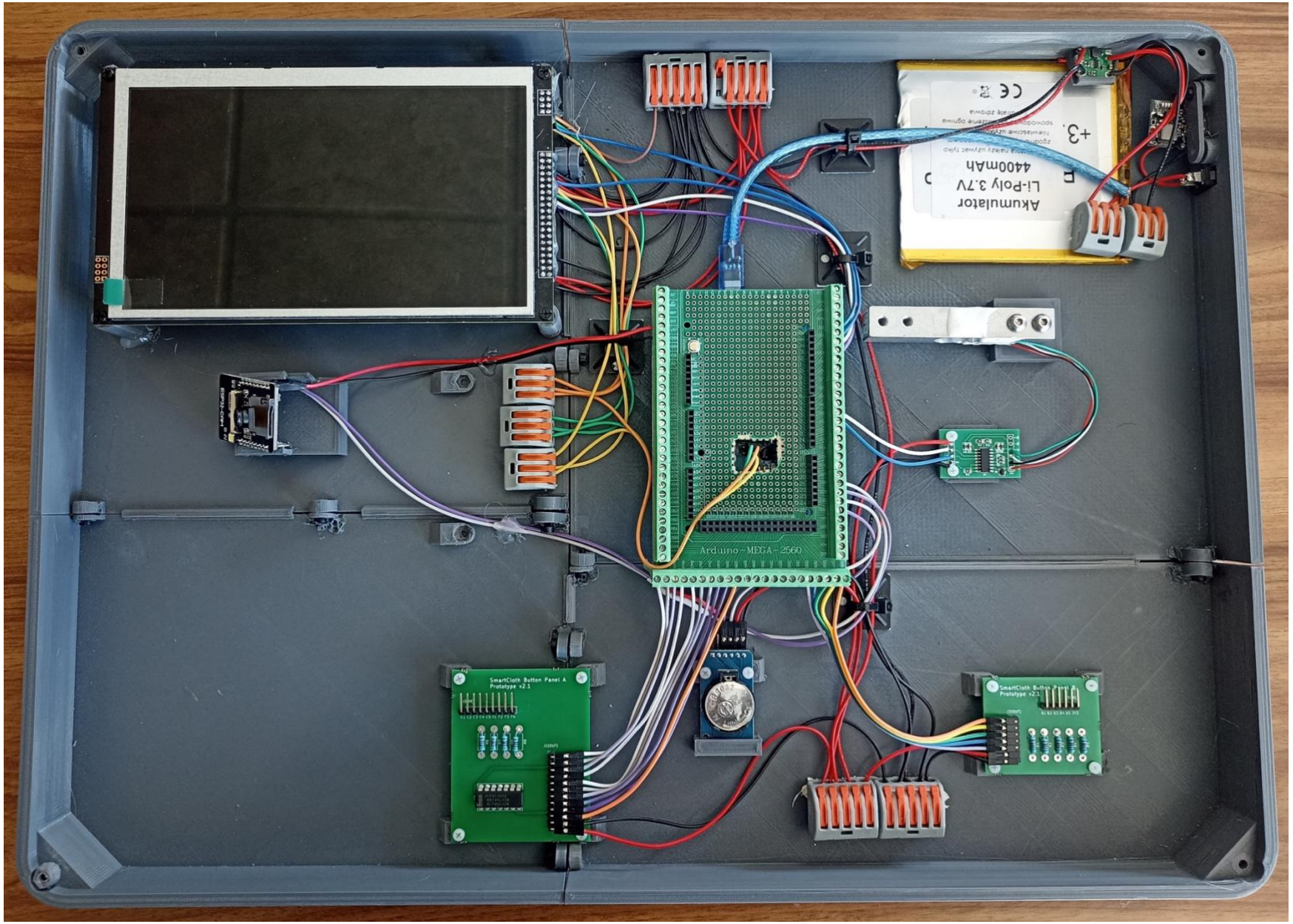




2.9 Botón código de barras:

CABLES DUPONT MACHO (N: negro; AZ: azul)	
Botón código de barras	**Arduino Due**
GND: **1N x 10cm**	GND
Señal: **1AZ x 10cm**	Pin 53 (digital)

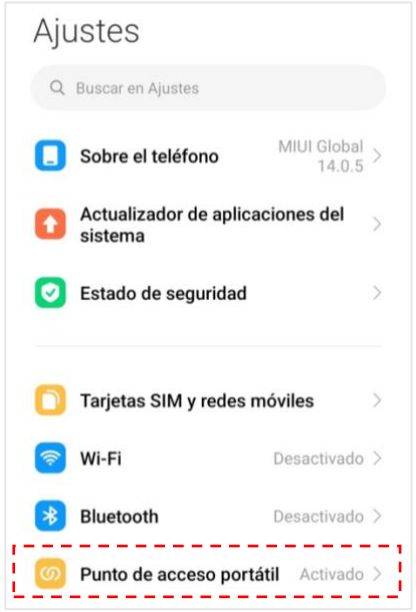


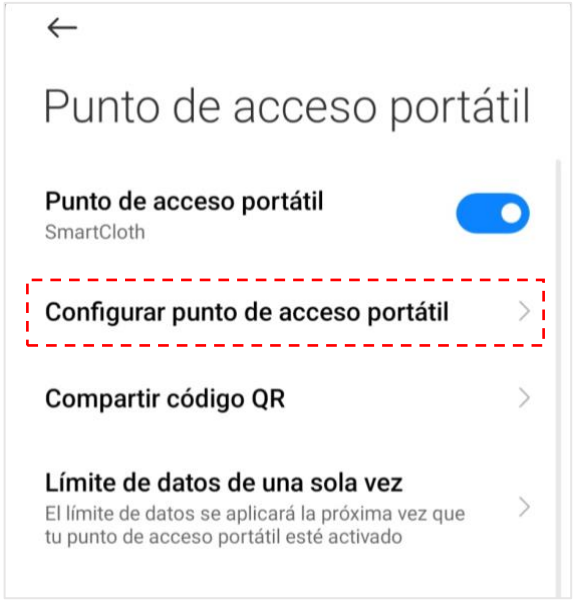

Akumulator
Li-Poly 3.7V
4400mAh
Arduino-MEGA-2560
SmartCloth Button Panel A
Prototype v2.1
SmartCloth Button Panel B
Prototype v2.1

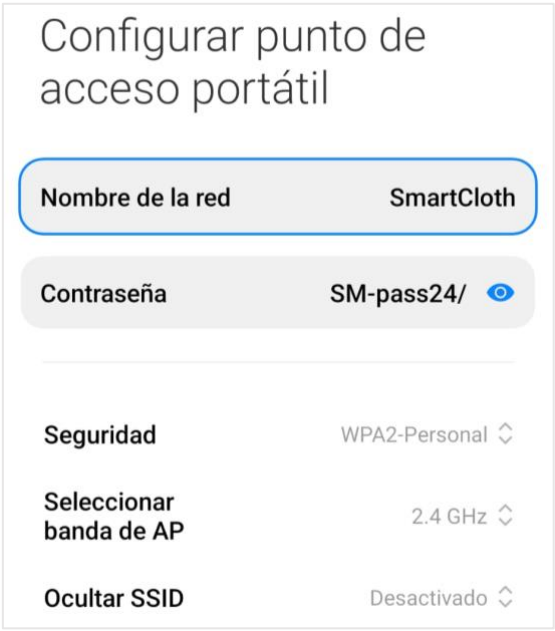
Akumulator
Li-Poly 3.7V
4400mAh
LECTOR
Arduino MEGA 2560
SMARTCLOTH
COCINADO
CRUDO
AÑADIR PLATO
BORRAR PLATO
GRABAR

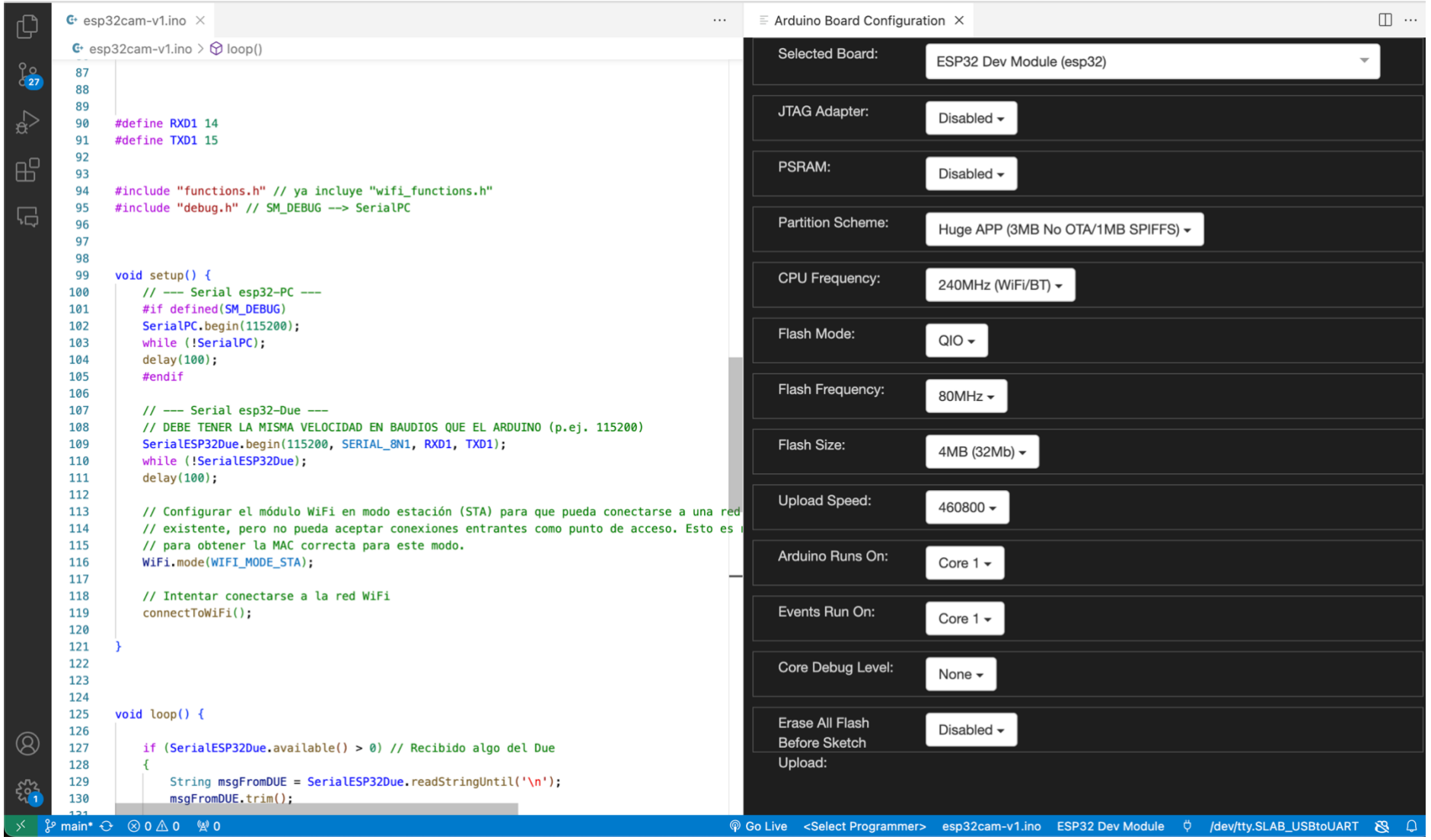
SOFTWARE

1. Guardar imágenes en SD manteniendo la estructura de ficheros
2. Poner RTC en hora (v2.1 - src/RTC_set_time)
3. Calibrar báscula (v2.1 - src/scale)
    a. Poner peso conocido (p.ej. celda de carga que pesa 30gr)
    b. $escala\ inicial * gr\ obtenidos\ al\ pesar = X$
    c. ${}^{X}/_{30\ gr} = nueva\ escala\ (bascula\ calibrada)$
    d. Modificar programa Arduino con nueva escala calculada (*setupScale()* → *scale.set_scale(escala)*)
4. Comentar *#define SM_DEBUG* del *debug.h* en los programas de Arduino y ESP32
5. Programar Arduino (v2.1 - src/smartcloth_v2)
6. Cambiar credenciales WiFi (nombre y contraseña) en programa ESP32
7. Programar ESP32 (v2.1 - src/esp32cam-v1)








esp32cam-v1.ino
esp32cam-v1.ino > loop()
#define RXD1 14
#define TXD1 15
#include "functions.h" // ya incluye "wifi_functions.h"
#include "debug.h" // SM_DEBUG --> SerialPC
void setup() {
// --- Serial esp32-PC ---
#if defined(SM_DEBUG)
SerialPC.begin(115200);
while (!SerialPC);
delay(100);
#endif
// --- Serial esp32-Due ---
// DEBE TENER LA MISMA VELOCIDAD EN BAUDIOS QUE EL ARDUINO (p.ej. 115200)
SerialESP32Due.begin(115200, SERIAL_8N1, RXD1, TXD1);
while (!SerialESP32Due);
delay(100);
// Configurar el módulo WiFi en modo estación (STA) para que pueda conectarse a una red
// existente, pero no pueda aceptar conexiones entrantes como punto de acceso. Esto es
// para obtener la MAC correcta para este modo.
WiFi.mode(WIFI_MODE_STA);
// Intentar conectarse a la red WiFi
connectToWiFi();
}
void loop() {
if (SerialESP32Due.available() > 0) // Recibido algo del Due
{
String msgFromDUE = SerialESP32Due.readStringUntil('\n');
msgFromDUE.trim();
Arduino Board Configuration
Selected Board:
ESP32 Dev Module (esp32)
JTAG Adapter:
Disabled
PSRAM:
Disabled
Partition Scheme:
Huge APP (3MB No OTA/1MB SPIFFS)
CPU Frequency:
240MHz (WiFi/BT)
Flash Mode:
QIO
Flash Frequency:
80MHz
Flash Size:
4MB (32Mb)
Upload Speed:
460800
Arduino Runs On:
Core 1
Events Run On:
Core 1
Core Debug Level:
None
Erase All Flash Before Sketch Upload:
Disabled
main*
0 0
0
Go Live
<Select Programmer>
esp32cam-v1.ino
ESP32 Dev Module
/dev/tty.SLAB_USBtoUART