

CONSTRUCCIÓ D'UNA POLSERA INTEL·LIGENT AMB ARDUINO

Nora Bolívar López

TREBALL DE RECERCA

Dirigit per: Jose Toro i Jordi Orts

Any 2019-2020

Institut Príncep de Viana

2n de Batxillerat A

SUMARI

1 RESUM.....	5
2 INTRODUCCIÓ.....	7
3 QUÈ SÓN ELS <i>SMARTWATCHES</i> I LES <i>SMARTBANDS</i> ?.....	9
4 ARDUINO.....	11
4.1 Què és?.....	11
4.2 Com funciona?.....	12
5 COMPONENTS.....	14
5.1 Wemos D32 PRO.....	14
5.2 Sensor de freqüència cardíaca, MAX30102.....	16
5.3 Sensor de pressió baromètrica, BMP180.....	17
5.4 Sensor d'UV, VEML6070.....	18
5.5 Sensor d'humitat i temperatura, SI7021.....	19
5.6 Triple eix brúixola-magnetòmetre, HMC5883L.....	20
5.7 Pantalla OLED.....	21
5.8 Sensor de gestos, PAJ7620.....	24
5.9 Relotge, RTC DS1307.....	25
5.10 Brunzidor.....	25
5.11 Preus.....	26
6 PROCÉS CODI ARDUINO.....	27
6.1 Relotge.....	28
6.2 Sensor de freqüència cardíaca.....	29
6.3 Brúixola.....	30
6.4 Comptador de plantes pujades.....	31
6.5 Sensor de temperatura i humitat.....	32
6.6 UV.....	32
6.7 Cronòmetre.....	33
6.8 Notificacions.....	34
6.9 Alarma.....	37

6.10 Brunzidor.....	38
6.11 Programa final.....	39
6.12 Diagrama de flux.....	41
7 APP INVENTOR.....	45
8 CONCLUSIÓ I VALORACIÓ.....	49
AGRAÏMENTS.....	50
BIBLIOGRAFIA.....	51
9 ANNEXOS.....	56
9.1 ANNEX 1: instal·lar Arduino i configurar l'entorn.....	56
9.2 ANNEX 2: Què és l' I ² C i com ho apliquem a Arduino.....	59
9.3 ANNEX 3: exemple sensor MAX30105.....	63
9.4 ANNEX 4: altímetre amb el BMP180.....	64
9.5 ANNEX 5: exemple sensor VEML6070.....	65
9.6 ANNEX 6: exemple sensor HMC5883L.....	66
9.7 ANNEX 7: exemple sensor SI7021.....	67
9.8 ANNEX 8: exemple OLED amb els botons.....	68
9.9 ANNEX 9: exemple OLED amb imatges.....	70
9.10 ANNEX 10: exemple brunzidor.....	71
9.11 ANNEX 11: exemple sensor PAJ7620.....	72

1 RESUM

CASTELLANO

En este trabajo he construido un prototipo de una pulsera inteligente con Arduino. Basándome en las pulseras que hay actualmente en el mercado, he creado un reloj que muestra la información en diferentes pantallas, por las que puedes moverte gracias a los botones incorporados o a un sensor de gestos. Las pantallas se pueden dividir en dos: las que necesitan un sensor (brújula, pulsómetro, humedad relativa, temperatura ambiental, pisos subidos y nivel de UV) y las que no (reloj, cronómetro, alarma y, a través de una aplicación externa, el número de notificaciones de WhatsApp y de Gmail que lleguen al móvil). Adicionalmente, también he creado una aplicación para el móvil que recibe varios parámetros a través del Bluetooth.

En primer lugar, me informé sobre el funcionamiento básico de Arduino: las diferentes partes del programa, las funciones y las variables. Después, probé cada sensor de forma individual: me descargué las librerías y miré los ejemplos básicos, editándolos hasta llegar al código que necesitaba. Seguidamente, pasé a programar aquello que no requería un sensor. Mientras editaba el código, creé el diseño de las pantallas de la OLED donde se mostraría la información junto a una imagen. Finalmente, obtuve el programa final combinando todo el código.

Estoy muy contenta con este trabajo porque, a pesar de las limitaciones de tiempo y de conocimiento, he conseguido crear un prototipo de reloj por un precio competitivo que funciona correctamente, aprendiendo a la vez los fundamentos de un lenguaje de programación.

ENGLISH

In this project, I have built a prototype of a smartband with Arduino. Based on the bands that are on the market, I have created a watch that shows the information on different screens, where you can browse thanks to the built-in buttons or a gesture sensor. The screens can be classified into those that need a sensor (compass, heart rate, relative humidity and ambient temperature, floors climbed and UV level) and those that do not need it (clock, stopwatch, alarm and, through an external application, the number of WhatsApp and Gmail notifications that receive the phone). Additionally, I have created an app for smartphones that receives several parameters via Bluetooth.

First of all, I found out information about the basic running of Arduino: the different sections of the program, functions and variables. Then, I tested each sensor individually: I downloaded the libraries and examined the basic examples, editing them until I reached the code I wanted. Afterwards, I programmed all the things that didn't need a sensor. While I was editing the code, I created the design of the OLED screens where the information is displayed together with a picture. Eventually, I got the definitive program by combining all the code.

I am highly satisfied with this project because, despite the limitations of time and knowledge, I have managed to create a band prototype for a competitive price that works correctly as well as I have learned the basics of a programming language.

2 INTRODUCCIÓ

Amb aquest treball de recerca pretenc crear una polsera intel·ligent, o *smartband*, amb Arduino. M'he basat en el meu rellotge Fitbit per intentar aproximar-la en funcionalitats als dispositius disponibles al mercat. El meu dispositiu ha de poder mostrar l'hora i la data, detectar la freqüència cardíaca de l'usuari, comptar els pisos pujats, mesurar la temperatura i la humitat relativa de l'ambient, mostrar una brúixola amb el punt cardinal, funcionar com a cronòmetre, mostrar el risc que suposa el nivell de llum ultraviolada a cada moment, programar alarmes i, si se sincronitza amb una aplicació, visualitzar el nombre de notifikacions de determinades aplicacions.

He triat l'àmbit tecnològic perquè estic enfocant els meus estudis cap a aquesta branca del coneixement. He escollit treballar amb Arduino per així aprendre els fonaments del llenguatge de programació C++.

El C++ és un llenguatge derivat del C amb la intenció de fer més fàcil manipular objectes. Les seves característiques més destacables són que permet l'agrupació d'instruccions i que és un llenguatge d'alt nivell, que significa que és més semblant al llenguatge humà i és més fàcil d'interpretar.

Aquest tema en específic em sembla interessant, ja que en els darrers anys s'han posat molt de moda aquests rellotges, sobretot en l'àmbit esportiu. A més, gràcies al ventall de dispositius i de preus, moltes persones ho fan servir com un complement més.

Primerament, vaig pensar en tots els sensors que necessitaria, i una vegada els vaig obtenir, vaig descarregar-me totes les llibreries¹ necessàries. Seguidament, vaig comprovar un per un que tots els perifèrics funcionaven correctament a través dels exemples proporcionats per les llibreries. Després, a partir dels exemples, vaig anar construint el meu codi segons les meves necessitats i alhora, vaig crear les pantalles individuals fusionant aquest codi amb imatges i text per l'OLED. Finalment, vaig aglutinar totes les pantalles per aconseguir el codi final. Per complementar aquest projecte, vaig crear una aplicació amb App Inventor i faig servir l'aplicació Notiduino Arduino IoT Platform.

En aquesta memòria primer trobareu un petit resum en castellà i anglès d'aquest treball de recerca. Després de forma breu explicaré que és l'Arduino i el seu funcionament. Tot seguit, hi ha informació sobre tots els materials que vaig fer servir, en ordre en el qual els vaig provar. Per cada sensor, he explicat com connectar-lo i com funciona amb un exemple de codi. Per finalitzar aquest apartat, indico quin és el preu aproximat de cada mòdul i el total. Al següent capítol, explico quin ha sigut el procés de creació del codi seguint l'ordre de les pantalles en el codi final. Per facilitar la comprensió del programa, he dissenyat un diagrama de flux. Després explico el programa que he creat amb App Inventor i finalment exposo les meves conclusions i la meva valoració final.

Per a aquest projecte he consultat diverses pàgines. A Arduino he consultat al seu fòrum i les anomenades *Language Reference*, on per exemple he buscat les diferències entre les múltiples variables. A Github he trobat gairebé totes les llibreries emprades en aquest treball. També m'he documentat a la web "<https://www.luisllamas.es/>", quan encara no tenia nocions bàsiques d'Arduino. Allà vaig trobar l'explicació del codi de molts dels sensors.

¹ Una llibreria és una eina que facilita la programació. Permet establir funcions per realitzar processos i simplificar el codi. Hi ha diverses incloses a l'Arduino IDE però també es poden aconseguir a Github. En aquest cas s'ha de descarregar un arxiu Zip i incloure'l a Arduino a través d'*Esbós > inclou la biblioteca > Afegeix una biblioteca .ZIP*. Així i tot, no és necessari fer-les servir.

3 QUÈ SÓN ELS *SMARTWATCHES* I LES *SMARTBANDS*?

Els *smartwatches* són rellotges intel·ligents amb una pantalla que permeten a l'usuari veure les notificacions del mòbil, comptar passes, freqüència cardíaca, calories, etc. També permet interactuar i controlar la música, veure el temps i moltes altres funcions que depenen del model.

Les *smartbands*, o polseres intel·ligents, són més senzilles i moltes no tenen pantalla, i si la tenen, només t'informen de les dades recollides i potser t'informen que tens una notificació.

L'*smartwatch* més venut el 2018 va ser l'Apple Watch 4, el qual es pot adquirir per 300 euros, encara que la seva millor configuració val més de 500. Així i tot, la seva autonomia no supera els dos dies. Altres models populars són els rellotges de Xiaomi. Avui dia, la més recent és la Mi Smart Band 4, la qual té una petita pantalla a color i una bateria que promet arribar als vint dies d'autonomia. Aquesta és més assequible i està a l'abast de molta gent, ja que el seu preu oficial és de 35 euros. Altres marques reconegudes són Fitbit, Samsung i Huawei.

Figura 1. Apple Watch 4



Extreta de la web
"www.amazon.es"

Figura 2. Mi Smart Band 4



Extreta de la web
"www.bestbuy.com"

Figura 3. Fitbit Charge 3



Extreta de la web
"www.amazon.es"

CONSTRUCCIÓ D'UNA POLSERA INTEL·LIGENT AMB ARDUINO

Nora Bolívar López

La Fitbit Charge 3 és una de les polseres intel·ligents més completes al mercat. Promet cinc dies de funcionament sense carregar-la i té una molt bona resolució de pantalla. A més, es pot submergir. L'aplicació amb la qual es poden configurar les esferes és molt intuïtiva i va recollint les dades de cada dia per mostrar-les en gràfiques.

Figura 4. Pantalla inicial aplicació Fitbit



Imatge de l'autora

Figura 5. Pantalla on es mostra les passes fetes diàriament



Imatge de l'autora

Figura 6. Pantalla on es mostra la qualitat i duració del son



Imatge de l'autora

En aquest treball he realitzat una polsera intel·ligent amb una aplicació que només permet visualitzar les dades en temps real.

4 ARDUINO

4.1 Què és?

L'Arduino és una plataforma de hardware i software lliure² multiplataforma³. L'idea va néixer l'any 2005, a Itàlia, amb la finalitat d'abaratir el material utilitzat pels estudiants de l'Institut IVREA. Però aquell mateix any la universitat va tancar i van decidir compartir el projecte amb el món.

A nivell d'usuari ha tingut molt d'èxit per diversos motius. El primer és que les plaques i els perifèrics són econòmics. Per exemple, l'Arduino Uno i la Nano valen 20 € cadascuna a la pàgina oficial. També hi ha la possibilitat d'adquirir versions no-oficials a pàgines com AliExpress, on valen menys de 5 €. Un altre motiu és el llenguatge de programació, el C++, molt intuïtiu, sobretot per a la gent que acaba d'entrar al món de la programació. A més, hi ha una gran comunitat que penja contingut diàriament i a través dels fòrums, ajuden a resoldre problemes i dubtes.

Figura 7. Logo d'Arduino



Extreta de la web "es.wikipedia.org"

Figura 8. L'Arduino Uno, la placa més coneguda i usada de la marca



Extreta de la web "store.arduino.cc"

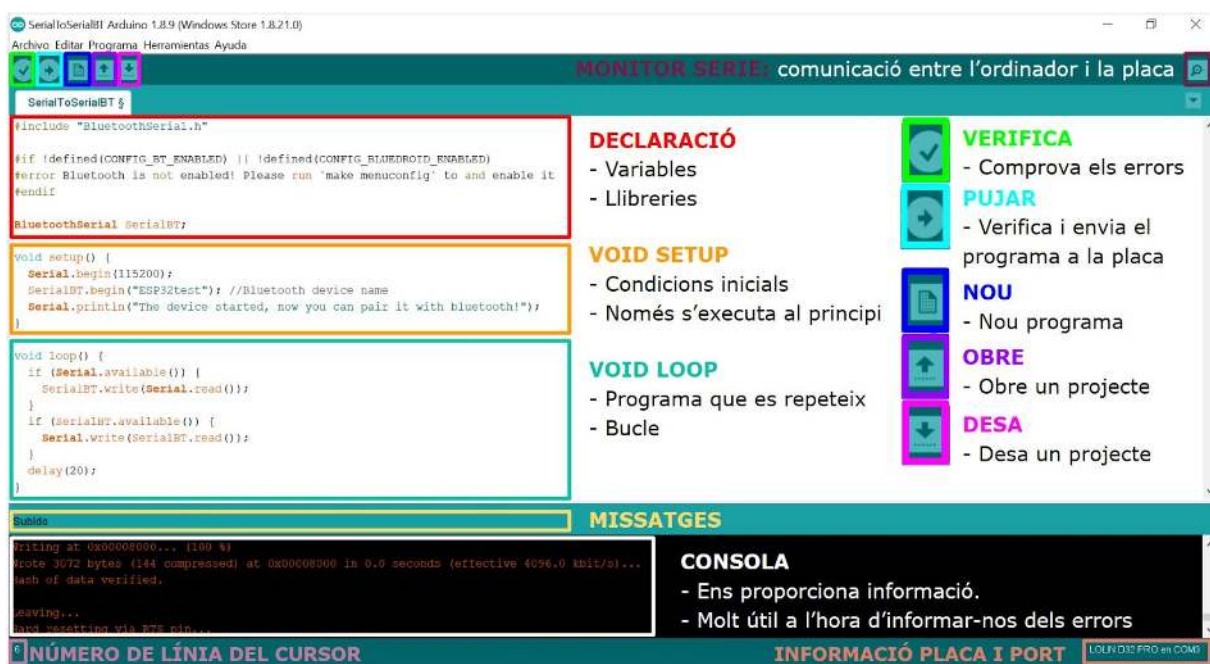
² Els usuaris tenen llibertat per a executar, copiar, distribuir, estudiar, canviar i millorar el programari.

³ Funciona a Windows, Mac OS i Linux.

4.2 Com funciona?

En la següent imatge podem veure l'Arduino IDE i les seves característiques amb l'exemple del Bluetooth:

Figura 9. L'IDE d'Arduino i les seves característiques



Imatge de l'autora

- La declaració, que va abans del "void setup", és on s'han d'incloure totes les llibreries que farem servir i també les variables, tot i que aquestes també es poden declarar al principi d'una funció. En aquest exemple s'inclou la llibreria per fer servir el Bluetooth, un missatge d'error i el nom amb el qual ens referirem al Bluetooth durant el programa.
- El "void setup" és una funció que només s'executa una vegada al principi. En aquest cas, s'inicia la comunicació entre l'ordinador i la placa a través del Monitor Sèrie, s'inicia el Bluetooth, escollim el nom que apareixerà al llistat de dispositius Bluetooth i ho informa el Monitor Sèrie.

- El "void loop" és la part del programa que s'executa constantment en bucle, només es detindrà si hi ha un "delay"⁴. En aquest cas, hi haurà una pausa de 20 mil·lisegons. En aquest simple programa estableix una mena de xat entre el Monitor Sèrie i un altre dispositiu amb Bluetooth que tingui instal·lat una aplicació que permeti enviar i rebre dades a través del Bluetooth.
- El Monitor Serie ens permet llegir o enviar dades a la placa, amb les funcions ".write" i ".read". La velocitat d'aquest procés la definim en el parèntesi de "Serial.begin()". Les més comunes són 9600 i 115200 bauds.
- Una vegada tinguem el codi, podem veure si és correcte amb l'opció de verificar, o bé podem verificar i enviar el programa a la placa alhora amb el botó de pujar. Si ens detecta un error, ens ho indicarà a la zona dels missatges i a la consola.

⁴ Un "delay" és una funció que pausa el programa durant els mil·lisegons (1000 mil·lisegons equival a 1 segon) que hi ha especificat entre els parèntesis.

5 COMPONENTS

5.1 Wemos D32 PRO

La placa que he utilitzat és la "WEMOS D32 PRO", una placa amb wifi i Bluetooth basada en l'"ESP32", d'Espressif. Les seves especificacions són les següents:

Microcontrolador	ESP-32
Alimentació subministrada via USB	5V
Bateries admeses	Bateria de liti 3.7V
Voltatge operatiu	3,3 V
Pins digitals ⁵	22
Pins d'entrada analògics ⁶	6 (VP, VN, 32, 33, 34, 35)
Pins de sortida analògics	2 (25, 26)
LED_BUILTIN ⁷	5
Velocitat màxima del rellotge	240 MHz
Flash ⁸	4M bytes
Longitud	65 mm
Amplada	25.4 mm
Pes	7.5 g

Els pins que utilitzen gairebé tots els sensors són el GND, el 3V, el SDA i el SCL, ja que fan servir l'"I²C"⁹. El brunzidor també fa servir el pin "0". A més, també utilitzo el port "I²C" per connectar la pantalla OLED.

5 Els pins digitals són aquells que només tenen dos valors: -Vcc o +Vcc. Aquests dos estats són anomenats LOW (0) i HIGH (1).

6 Els pins analògics llegeixen valors de tensió d'entre 0V a 5V.

7 El LED_BUILTIN és el número del pin del LED integrat a la placa.

8 La memòria FLASH és on es desa el programa.

9 Veure annex 2.

CONSTRUCCIÓ D'UNA POLSERA INTEL·LIGENT AMB ARDUINO

Nora Bolívar López

En comparació amb l'Arduino Uno, la placa més coneguda d'Arduino, és més ràpida, disposa de més memòria i té *Dual-Core*¹⁰. Això permet un millor rendiment a l'hora de treballar amb projectes que requereixen més processos. A més, és més petita i lleugera.

Figura 10. LOLIN D32 PRO. Part posterior



Imatge de l'autora

Figura 11. LOLIN D32 PRO. Part davantera



Imatge de l'autora

¹⁰ Dos nuclis. Cada nucli extra és una unitat central de processament addicional, que permet a una placa fer diverses coses alhora.

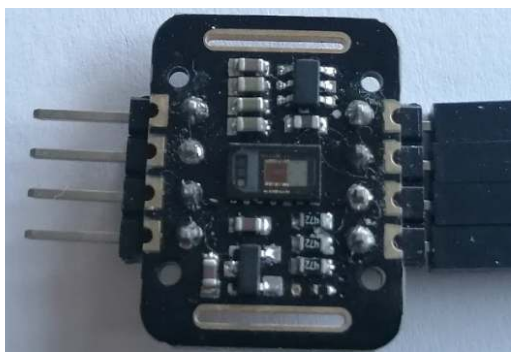
5.2 Sensor de freqüència cardíaca, MAX30102

El MAX30102 és capaç de detectar les pulsacions d'una persona, la temperatura corporal i el percentatge d'oxigen en sang. Fa servir la llibreria "MAX30105.h"¹¹, on hi ha diversos exemples on s'explica com obtenir aquestes dades. Fa servir el "bus I²C" i 3,3V.

A l'annex 3 podem trobar un exemple de la llibreria modificat. Primer declarem les llibreries i les funcions necessàries. Al "void setup" iniciem la comunicació serial i el sensor. També configurem l'"I²C" perquè funcioni a alta velocitat i establim la configuració predeterminada del pulsòmetre. Al "void loop" recollim tres dades:

- L'"irValue": aquest ens interessa per detectar la proximitat d'un objecte. Utilitza un filtre anomenat "PBA" (*Peripheral Beat Amplitude*), que detecta "tocs" i reconeix el temps entre les irregularitats, que serien els batecs, i calcula la freqüència cardíaca. Si aquest valor és inferior a 50.000, és a dir, no detecta res a un centímetre de distància aproximadament, al Monitor Sèrie apareixerà que no s'ha detectat el pols.
- "BeatsPerMinute": és la freqüència cardíaca instantània.
- "BeatAvg": és la freqüència cardíaca mitjana (BPM). És la mesura que apareix al Monitor Sèrie i és el la mitjana de "BeatsPerMinut".

Figura 12. MAX30102. Part davantera



Imatge de l'autora

Figura 13. MAX30102. Part posterior



Imatge de l'autora

¹¹ Llibreria obtinguda a "https://github.com/sparkfun/SparkFun_MAX3010x_Sensor_Library".

5.3 Sensor de pressió baromètrica, BMP180

El BMP180 és capaç de detectar la temperatura, la pressió atmosfèrica i calcular l'altitud amb una referència. La llibreria que fa servir s'anomena "SFE_BMP180.h"¹². El sensor funciona amb un voltatge que va dels 1,8 a 3,3 V.

A l'annex 4 es pot veure com he programat un comptador de plantes basant-me en un codi extret de "<https://www.luisllamas.es>".

El sensor fa una primera lectura de la temperatura abans d'obtenir la pressió, ja que la temperatura afecta la densitat de l'aire. Per això, tindrà en compte aquesta dada per mesurar la pressió.

La pressió atmosfèrica és la força que exerceix l'aire sobre la superfície terrestre i és creada pel pes de l'aire. Aquesta força és mesurada pel BMP180 i és el que fa servir per determinar l'altura a través de la següent equació, on " p " és la pressió mesurada i " p_0 " és la pressió del nivell del mar:

$$altitude = 44330 * (1 - (\frac{p}{p_0})^{\frac{1}{5.255}})$$

La pressió és donada pel BMP180 en pascals (Pa), que és la unitat en Sistema Internacional. El rang és de 300 hPa a 1110 hPa: una altitud de -500 m a 9000 m sobre el nivell del mar.

Figura 14. BMP180. Part davantera



Imatge de l'autora

Figura 15. BMP180. Part posterior



Imatge de l'autora

¹² Llibreria obtinguda a "https://github.com/sparkfun/BMP180_Breakout_Arduino_Library".

5.4 Sensor d'UV, VEML6070

El VEML6070 és un sensor que utilitza la llibreria "Sseed_VEML6070.h"¹³. Aquest perifèric és capaç de detectar la intensitat de la llum ultraviolada. Funciona a través de l'PC i funciona tant en 3,3 com en 5 V.

Figura 16.
BMP180. Part
davantera



Imatge de l'autora

Figura 17.
BMP180. Part
posterior



Imatge de l'autora

La llum ultraviolada és la radiació electromagnètica emesa pel Sol amb una longitud d'ona que va dels 100 fins als 400 nm. L'UV és necessari per a la fotosíntesi de les plantes però també per les persones, ja que sintetitzem la vitamina D. Així i tot, necessitem una quantitat determinada d'UV, ja que si ens quedem curts, podem tenir problemes als ossos, però si ens excedim, podem desenvolupar càncer de pell, cataractes, etc. La següent taula ens indica les mesures que hem de prendre segons el nivell d'UV:

1	Nivell baix	- Utilitzar ulleres de Sol.
2		
3	Nivell moderat	- Utilitzar crema solar.
4		
5		
6	Nivell alt	- No exposar-se al Sol en les hores de més intensitat.
7		
8	Nivell molt alt	- Utilitzar crema solar sempre.
9		
10		
11+	Nivell extrem	- Evitar sortir a l'exterior.

A l'annex 5 fa una conversió del nivell d'intensitat al risc que suposa l'exposició en aquell moment.

¹³ Llibreria obtinguda a

"https://github.com/Seeed-Studio/Seeed_VEML6070/blob/master/examples/basic_demo_basic_demo.ino".

5.5 Sensor d'humitat i temperatura, SI7021

El sensor SI7021 pot mesurar la humitat relativa¹⁴ en un rang del 0 al 80% amb un marge d'error de $\pm 3\%$. També és capaç de saber la temperatura que va dels -10 al $85\text{ }^{\circ}\text{C}$, amb una precisió de $\pm 0,4\text{ }^{\circ}\text{C}$. Aquest perifèric només es pot fer servir amb I²C i admet 3,3 V, però no 5V.

A l'annex 7 es pot veure que aquest sensor és molt fàcil de fer servir, ja que amb les funcions "sensor.readHumidity()" i "sensor.readTemperature()" incorporades a la llibreria "Adafruit_Si7021.h"¹⁵, podem obtenir les dades molt fàcilment.

Figura 18. SI7021. Part davantera



Imatge de l'autora

Figura 19. SI7021. Part posterior



Imatge de l'autora

¹⁴ "La humitat relativa és la relació entre la quantitat de vapor d'aigua que conté un espai determinat (humitat absoluta, expressada en $\text{g}\cdot\text{m}^3$) i la quantitat total que podria arribar a contenir en cas de saturació a una mateixa temperatura."(MUSEU NACIONAL D'ART DE CATALUNYA. Humitat relativa. [en línia]. <<https://www.museunacional.cat/ca/humitat-relativa>> [Consulta: 09/10/19]).

¹⁵ Llibreria obtinguda a "https://github.com/adafruit/Adafruit_Si7021".

5.6 Triple eix brúixola-magnetòmetre, HMC5883L

He utilitzat el HMC5883L com a una brúixola digital, ja que obté els valors del camp magnètic en tres eixos. Proporciona una precisió de $\pm 2^\circ$. Utilitza I²C i un voltatge d'entre 1,8 a 3,3 V. Fa servir dues llibreries: l'"HMC5883L.h" i l'"I2Cdev.h"¹⁶.

Com es pot veure a l'annex 6, per poder obtenir l'orientació primer mesurem els components magnètics. Tot i que el nord geogràfic no està alineat amb el nord magnètic, la diferència a Barcelona segons "<https://www.ign.es>" és de menys d'1 °C.

Després calcula l'angle de l'eix X respecte al nord amb l'arctangent de dos paràmetres, "my" i "mx", com es pot veure a la Figura 23. Aquests graus són passats de radians a graus sexagesimals i se li resta la declinació. Depenent de l'angle, es mostrarà el punt cardinal corresponent o les zones intermèdies entre aquests.

Figura 20.
HMC5883L. Part
davantera



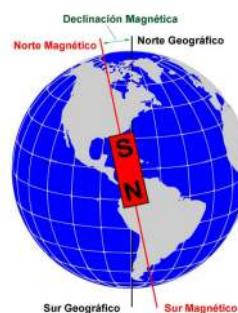
Imatge de l'autora

Figura 21.
HMC5883L. Part
posterior



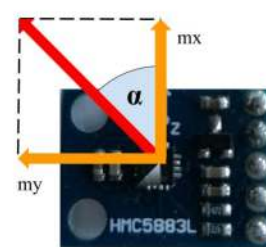
Imatge de l'autora

Figura 22.
Declinació
magnètica



Extreta de la web
"Sail and Trip"

Figura 23. Obtenir
angle amb
l'arctangent



$$\alpha = \text{angle} = \arctan \frac{my}{mx}$$

Imatge de l'autora

¹⁶ Llibreries obtingudes a "<https://github.com/jrowberg/i2cdevlib>".

5.7 Pantalla OLED

La pantalla OLED que he fet servir té 64*48 píxels, que són 0,66 polzades de diagonal (1,68 centímetres). Està connectada a la placa a través del port "I²C" i incorpora dos botons "I²C" ("A" i "B"). He utilitzat tres llibreries: l'<Adafruit_GFX.h>¹⁷, que proporciona funcions gràfiques (fonts, formes, etc.); l'<Adafruit_SSD1306.h>¹⁸, que és la llibreria per la pantalla, i la <LOLIN_I2C_BUTTON.h>¹⁹, per així poder fer servir els botons que incorpora l'OLED.

Figura 24. OLED shield



Imatge de l'autora

A l'annex 8 podem trobar un exemple de la llibreria <LOLIN_I2C_BUTTON.h> que també incorpora les altres dues esmentades. El codi es basa en un comptador que va del 0 fins al 255. Si es prem l'"A", resta, i si es prem el "B", suma. Es detecten quatre tipus de pressions: una curta, que suma o resta 1; dues seguides, que suma o resta 2; una mantinguda durant dos segons aproximadament, que suma o resta 10; una mantinguda durant més temps, que en aquest cas no fa res.

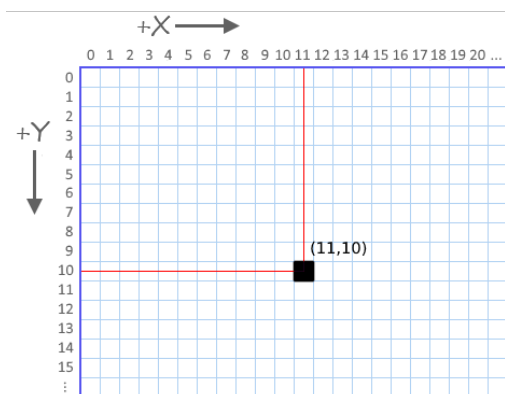
A l'annex 9 podem veure com cada 5 segons va mostrant dues pantalles: a la primera es mostra una imatge d'una gota d'aigua i la lectura d'humitat del sensor acompanyada del símbol "%"; a la segona apareix un dibuix d'un termòmetre i la lectura de temperatura del sensor amb el símbol "°C".

En aquests exemples el primer que fem és netejar la pantalla amb "display.clearDisplay()". Si volem escriure un text, primer declarem la mida de la lletra, entre els parèntesis hem d'escriure un número d'entre l'1 a 8, amb "display.setTextSize()". Després escollim el color del text, tot i que només tenim l'opció de què sigui blanc, amb

¹⁷ Llibreria preinstal·lada a l'IDE d'Arduino.

¹⁸ Llibreria obtinguda a "https://github.com/mcauser/Adafruit_SSD1306/tree/esp8266-64x48".

¹⁹ Llibreria obtinguda a "https://github.com/wemos/LOLIN_OLED_I2C_Button_Library".

Figura 25. Eixos en la pantalla OLED

Extreta de la web "www.esploradores.com"

Figura 26. Captura de pantalla de la web
<http://jav1.github.io/image2cpp/>

1. Select image



"display.setTextColor(WHITE)". Tot seguit indiquem la localització amb "display.setCursor(x,y)" seguint la Figura 25. Finalment, per escriure farem servir "display.println()". A l'annex 9 podem veure que gràcies a la llibreria <Adafruit_GFX.h>, és molt fàcil fer formes tals com un cercle amb "display.drawCircle(45, 21, 1, WHITE)".

Per mostrar una imatge primer la descarreguem i la pugem a la pàgina "<http://jav1.github.io/image2cpp/>".

Després ajustem la imatge com es pot veure a la Figura 27:

- Escollim la mesura segons la mida desitjada, en el meu cas 24*48 píxels. S'ha de tenir en compte la mesura de la pantalla perquè no ens mostri coses estranyes.
- Després triem el color de fons, en el meu cas el negre. El termòmetre és de color negre i per tant invertirem els colors perquè la imatge es mostri de color blanc.
- Seleccionem adaptar, proporcionar i centrar la imatge a les mesures.

Tot seguit, ens mostra com quedaria. Finalment, ens facilita inserir-ho a Arduino si seleccionem "Arduino code" en el format de sortida del codi. Escrivim l'identificador, en el meu cas "temp", i cliquem a generar el codi. Copiem el codi i ho enganxem a l'Arduino abans del "void setup". Quan vulguem que aparegui la imatge només hem d'escriure `display.drawBitmap(0,0,temp,24,48,1)`.

És molt important que al final incloguem "display.display()" per mostrar tots els canvis.

Figura 27. Captura de pantalla de la web "http://javl.github.io/image2cpp/"

2. Image Settings

Canvas size/s: Download-Temperature-PNG-Clipart.png (file resolution: 340 x 584)
 24 x 48 glyph remove

Background color: ☐ White ☒ Black

Invert image colors ☒

Brightness threshold: 128
 0 - 255; pixels with brightness above become white, below become black.

Scaling scale to fit, keeping proportions v

Center: ☒ horizontally ☒ vertically

NOTE: Centering the image only works when using a canvas larger than the selected image.

Figura 28. Captura de pantalla de la web "http://javl.github.io/image2cpp/"

3. Preview



4. Output

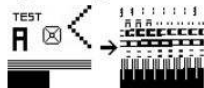
Code output format Arduino code v

Adds some extra Arduino code around the output for easy copy-paste into [this example](#). If multiple images are loaded, generates a byte array for each and appends a counter to the identifier.

Identifier: temp

Draw mode: ☒ Horizontal ☐ Vertical

If your image looks all messed up on your display, like the image below, try the other mode.



Generate code

```
// 'Download-Temperature-PNG-Clipart', 24x48px
const unsigned char myBitmap [] PROGMEM = {
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x03, 0x18, 0x00, 0x04,
  0x04, 0x00, 0x08, 0x02, 0x00, 0x08, 0x02, 0x00, 0x08, 0x02, 0x00, 0x08, 0x02, 0x00, 0x08, 0x02,
  0x00, 0x08, 0x02, 0x00, 0x08, 0x02, 0x7f, 0x08, 0x02, 0x00, 0x08, 0x02, 0x00, 0x08, 0x02, 0x70,
  0x08, 0x42, 0x00, 0x08, 0xe2, 0x00, 0x08, 0xe2, 0x7f, 0x08, 0xe2, 0x00, 0x08, 0xe2, 0x00, 0x08,
  0xe2, 0x70, 0x08, 0xe2, 0x00, 0x08, 0xe2, 0x00, 0x08, 0xe2, 0x7f, 0x08, 0xe2, 0x00, 0x08, 0xe2,
  0x00, 0x08, 0xe2, 0x00, 0x08, 0xe2, 0x00, 0x10, 0xe1, 0x00, 0x20, 0xe0, 0x80, 0x21, 0xf0, 0xc0,
  0x43, 0xf8, 0x40, 0x47, 0xfc, 0x40, 0x47, 0xfc, 0x40, 0x47, 0xfc, 0x40, 0x47, 0xfc, 0x40, 0x47,
  0xfc, 0x40, 0x43, 0xf8, 0x40, 0x20, 0xe0, 0x80, 0x30, 0x01, 0x80, 0x18, 0x03, 0x00, 0x0e, 0x0e,
  0x00, 0x03, 0xf8, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
};
```

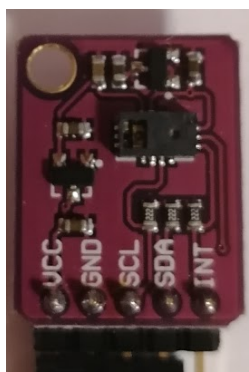

5.8 Sensor de gestos, PAJ7620

El sensor PAJ7620 és capaç d'identificar fins a 9 gestos diferents: amunt, avall, dreta, esquerra, enrere, cap endavant, onejant, en sentit horari i antihorari. En el meu projecte només he utilitzat dos moviments per passar les pantalles, com si fossin els botons: a l'esquerra equivaldria al botó A, però aquest botó només es fa servir per moure's dins una pantalla, i a la dreta correspon al botó B.

Per detectar el moviment, que ha de realitzar-se a una distància d'entre 5 i 30 centímetres, fa servir un LED infraroig i un sensor d'imatge. Utilitza l'"I²C" i baix voltatge, 3,3V.

A l'annex 11 podem trobar l'exemple de la llibreria "`paj7620.h`"²⁰. Al principi s'inclouen les llibreries i es defineixen els temps de reacció. Després podem veure com en fer un dels moviments, la variable "data" serà l'encarregada de dir quin gest s'ha fet.

Figura 29. PAJ620.
Part davantera



Imatge de l'autora

Figura 30. PAJ620.
Part posterior



Imatge de l'autora

²⁰ Llibreria obtinguda a "https://github.com/Seeed-Studio/Gesture_PAJ7620".

5.9 Rellotge, RTC DS1307

Per crear un rellotge que guardi l'hora necessitem un rellotge de temps real (RTC). Incorpora una petita pila per si no li arribés energia de la placa.

He tingut problemes amb el rellotge. Primer vaig provar el codi "ds1307" de la llibreria "RTClib.h"²¹ i no funcionava: al Monitor Serie indicava que la data era 2165/165/165 165:165:85. Pensava que ho havia arreglat amb el codi "softrtc", però al final em vaig adonar que utilitzava la funció millis²² i no el rellotge. A més, per comprovar-ho vaig utilitzar l'"I2C Scanner"²³ i no apareixia cap dispositiu. Llavors, em van suggerir que utilitzés 5V i va funcionar amb el codi "ds1307". El problema és que només proporciona 5 V l'ordinador i no una bateria, que és com funcionaria el rellotge. Per això, al final he descartat el seu ús. Així i tot, he aprofitat l'exemple de la llibreria que funciona amb "millis".

5.10 Brunzidor

El brunzidor és un petit aparell que converteix un senyal elèctric en so. El so només s'emetrà quan hi hagi una notificació nova i quan soni l'alarma. Per fer-ho funcionar, només s'ha de connectar el VCC a 3,3V, el GND i l'I/O al pin 0. A l'annex 10 podem trobar un petit exemple del funcionament d'un brunzidor: definim el pin 0 com una sortida i cada 0,1 segons encenem i apaguem el brunzidor.

Figura 31. Brunzidor. Part davantera



Imatge de l'autora

Figura 32. Brunzidor. Part posterior



Imatge de l'autora

21 Llibreria obtinguda a "<https://github.com/adafruit/RTClib>".

22 La funció "millis" compta els mil·lisegons que han passat des que el programa es va executar. Es pot configurar com un "delay", amb la diferència que pots decidir quina part del codi parar.

23 Codi d'autor desconegut que podem obtenir a la pàgina "<https://playground.arduino.cc/Main/I2cScanner/>". Només s'ha de pujar i s'hi ha un dispositiu I2C connectat, al Monitor Serie apareixerà la seva adreça.

5.11 Preus

COMPONENT	PROVEÏDOR	PREU (€)
Sensor de freqüència cardíaca (MAX30102)	AliExpress	1,69
Sensor pressió baromètrica (BMP180)	AliExpress	0,90
Sensor UV (VEML6070)	AliExpress	1,89
Sensor d'humitat i temperatura (Si7021)	AliExpress	1,57
Triple eix brúixola-magnetòmetre (HMC5883L)	AliExpress	1,20
Sensor de gestos (PAJ7620)	AliExpress	3,89
Brunzidor	AliExpress	0,60
LOLIN D32 Pro	AliExpress	8,05
Cables "dupont" femella-femella	AliExpress	0,21
PREU TOTAL		20

El preu total dels components és de 20 euros, deixant de banda les despeses d'entrega o cupons. També s'hauria de tenir en compte el salari de qui ho ha programat i que si es fes en sèrie, els materials es podrien adquirir per un preu inferior.

6 PROCÉS CODI ARDUINO

Quan vaig començar el projecte no tenia nocions bàsiques d'Arduino. Per això, vaig començar buscant les coses bàsiques i a la pàgina "<https://programarfacil.com/blog/arduino-blog/curso-de-arduino/>" vaig trobar una guia molt ben explicada i estructurada.

Després de navegar per diverses pàgines, vaig començar per instal·lar l'Arduino IDE i el paquet per l'"ESP32"²⁴. Tot seguit, vaig començar a provar cada perifèric: primer el connectava tenint en compte el voltatge, tots funcionaven a 3,3 V, excepte l'RTC; a continuació vaig descarregar les llibreries necessàries; després vaig provar els exemples inclosos a les llibreries per veure si els sensors funcionaven. Tots funcionaven i donaven resultats correctes, excepte l'RTC. A continuació, vaig programar allò que no requeria un sensor.

Després de comprovar-ho vaig passar a editar cada exemple segons què necessitava per la meva polsera i també aprofitava i feia la pantalla per l'OLED.

Finalment vaig pensar la manera d'aglutinar tots els codis amb variables que anessin canviant el seu valor depenent del botó per així canviar la pantalla. L'únic sensor que no funcionava correctament en el codi final va ser el sensor de freqüència cardíaca.

²⁴ Veure annex 1.

6.1 Rellotge

Pel rellotge, vaig basar-me en l'exemple "softrtc". Només he conservat la part de la data, l'hora i el dia de la setmana. En la part superior de l'OLED hi ha l'hora, en format 24 hores i amb dues xifres (HH:MM). A sota, hi ha la data en lletra més petita (DD/MM/AAAA). A la part inferior hi ha el dia de la setmana en anglès.

Figura 33. Primera pantalla.
Rellotge



Imatge de l'autora

La part més complicada ha sigut quadrar tots els números. El programa proporciona la següent data: 6:5, 9/1/2019. Però perquè quedi més estètic, si el número és inferior a 10, afegirem un 0 davant, quedant 06:05, 09/01/2019.

```
if((now.minute())< 10) // Si el minut és inferior a 10, s'escriurà un 0 en (39,5) i el número en (51,5). Un número en lletra gran ocupa aproximadament 10 píxels d'amplada
{
    display.setCursor(39,5);
    display.setTextSize(2);
    display.setTextColor(WHITE);
    display.println("0");
    display.setCursor(51,5);
    display.println(now.minute(), DEC);
}
if((now.minute())>= 10)// En canvi, si és igual o superior a 10, el número s'escriu directament a (39,5)
{
    display.setCursor(39,5);
    display.setTextSize(2);
    display.setTextColor(WHITE);
    display.println(now.minute(), DEC);
}
```

6.2 Sensor de freqüència cardíaca

A l'exemple, al Monitor Sèrie apareixia l'"IR Value", els "BPM" i l'"AVG BPM". Però vaig decidir que només volia que aparegués les pulsacions mitjanes (AVG BPM) quan detectés un objecte a menys de 2 centímetres, ja que no tenia sentit que calculés les pulsacions si això no es complia. Quan encara s'està iniciant, els resultats que dona no són coherents i pot aparèixer que una persona té 20 ppm²⁵. Per això, si les pulsacions són inferiors a 39 ppm i/o no detecta un objecte en menys de 2 centímetres, apareixerà "- PPM" al Monitor Sèrie.

Després vaig passar a fer el disseny de la pantalla. Com es pot veure a la Figura 34, em vaig decantar per la imatge d'un electrocardiograma a la part superior i a la part inferior les pulsacions, que apareixeran en lletres petites.

Aquest sensor va bé només amb poques línies de codi, perquè quan ho he combinat amb un altre programa, o bé triga molt a donar un resultat correcte, o bé no n'arriba a detectar cap pulsació. He buscat altres llibreries i codis, però no he trobat cap que funcionés.

Figura 34. Segona pantalla. Freqüència cardíaca



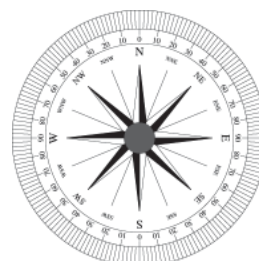
Imatge de l'autora

²⁵ Pulsacions per minut.

6.3 Brúixola

Per fer la brúixola, em vaig fixar en l'exemple de la llibreria, que indicava el nord magnètic i no el terrestre. Per obtenir un resultat més fidel a la realitat, he restat la declinació de Barcelona al nord magnètic per obtenir el nord terrestre. Segons el resultat i basant-me en la imatge dels punts cardinals, he fixat l'orientació:

Figura 35. Punts cardinals



Extreta de la web
"www.sitographics.com"

```
if(angulo >= 20 && angulo <70){
  Serial.print("NE:");
  Serial.println(angulo,0);
}
```

Aquestes línies de codi extretes de l'annex 6, signifiquen que si l'angle resultant està entre els 20 i els 70, al Monitor Sèrie apareixerà "NE" i l'angle.

Per a la pantalla OLED, simplement vaig dissenyar una petita brúixola i al mig, apareix el punt cardinal:

Figura 36. Tercera pantalla. Punts cardinals



Imatge de l'autora

Només vaig necessitar declarar les imatges i afegir al codi que aparegui la imatge a la pantalla, com es pot veure en el codi final:

```
if(angulo >= 20 && angulo <70){
  Serial.print("NE:");
  Serial.println(angulo,0);
  display.drawBitmap(0,0,ne,64,48,1); // Declarada abans del "void setup"
  display.display();
}
```

6.4 Comptador de plantes pujades

Després de fer la brúixola, vaig fer el comptador de pisos pujats amb el sensor de pressió baromètrica que torna a 0 a mitjanit. Primer vaig intentar calcular l'altura a partir de la pressió a nivell del mar, però com la pressió obtinguda era major que la pressió a nivell del mar, posava que estava per sota d'aquest nivell. Després vaig optar per calcular l'altura entre dos punts.

En ell, primer s'obté la temperatura i la pressió. Aquesta primera mesura de la pressió la desa a "AA". Després de 15 segons, torna a calcular-ho tot i calcula la diferència d'altura respecte "AA" i ho desa a "A". Llavors si "A", és més gran o igual que 2,5 però més petit o igual que 6, suma una planta a la variable "PlantesPujades". Ho vaig provar a l'escala del meu pis diverses vegades i totes les plantes que pujava les comptava.

Per dissenyar la pantalla, vaig optar pel dibuix d'una persona pujant les escales a la part superior esquerra. En la part inferior dreta apareixerà el nombre de plantes, en lletra gran, acompanyat pel text "PLANTES", en lletra petita.

Figura 37. Quarta pantalla.
Plantes pujades



Imatge de l'autora

6.5 Sensor de temperatura i humitat

Aquest sensor és el més simple de tots, ja que només hi ha dues funcions: "sensor.readHumidity()" i "sensor.readTemperature()". No vaig necessitar editar el codi.

Vaig decidir dissenyar dues pantalles:

- La de la humitat relativa, amb una imatge d'una gota d'aigua amb un percentatge dins, i en la part dreta, la humitat relativa en lletra gran, i en lletra petita, “ % RH”.
- La de la temperatura, on en la part esquerra hi ha un termòmetre i en l'esquerra la temperatura en lletra gran, i el text “ °C TEMP ” en lletra més petita.

Figura 38. Cinquena pantalla.
Humitat relativa



Imatge de l'autora

Figura 39. Sisena pantalla.
Temperatura



Imatge de l'autora

6.6 UV

Primer vaig veure els exemples d'Adafruit, però el resultat de la mesura era el nivell d'intensitat. Vaig buscar si hi havia una relació entre aquest nivell i l'índex, però no la vaig trobar. Llavors, als exemples de SeedStudio vaig veure un exemple on indicava el risc d'exposició a l'UV en cada moment.

Per fer la pantalla, simplement buscar imatges relacionades amb la llum ultraviolada i em vaig decidir per un Sol amb les lletres UV. A sota apareix el "nivell UV_str", però traduït en català i amb lletra petita.

Figura 40. Setena pantalla. Nivell UV



Imatge de l'autora

6.7 Cronòmetre

Em vaig proposar fer un cronòmetre amb la funció "millis". Funciona amb el botó "A": quan el prems durant un segon, el comptador es posa en marxa o es pausa; quan es prem durant dos segons, el comptador torna a 0. Aquest mecanisme funciona amb la variable "pausacrono", la qual té dos valors: 0 i 1. En 0 es pausa i en 1 s'activa.

El codi funciona de la següent manera: quan estàs situat en la pantalla del cronòmetre i polses del botó A, començarà a sumar un 1 a la variable "segons". Quan hagin passat 60 segons, els segons tornaran a 0 i se sumarà un minut a la variable "minuts". Quan hagin passat 60 minuts, els minuts tornaran a 0 i se sumarà un 1 a la variable "hores".

El cronòmetre al principi només funcionava en aquesta pantalla, és a dir, si començaves a comptar i canviaves de pantalla, quan hi tornaves, apareixia el mateix temps que quan canviaves la pantalla del cronòmetre. Per solucionar-ho vaig crear una altra funció que fa únicament de cronòmetre i s'activa quan "pausacrono" equival a 1, tant en aquesta pantalla com en d'altres.

Figura 41. Vuitena pantalla. Cronòmetre



Imatge de l'autora

6.8 Notificacions

Una de les funcions que més m'agrada de les polseres intel·ligents és la possibilitat de rebre les notificacions del mòbil i fins i tot respondre amb missatges predeterminats. En el meu projecte, gràcies a l'aplicació Notiduino, la qual es pot descarregar a Google Play²⁶, es mostra a la pantalla el nombre de notificacions de WhatsApp i de Gmail que han arribat al nostre mòbil. A Arduino vaig crear diverses variables:

```
int WHATSAPP = 0;
int GMAIL = 0;
int incoming;
int periode2 = 3000;
unsigned long tempsara2 = 0;
```

Per fer-ho només vaig haver de seguir els passos següents a l'aplicació Notiduino:

- Primerament vaig prémer al botó rosa amb el signe “+” situat a la part inferior dreta. Apareix una llista amb totes les aplicacions del mòbil. Vaig escollir només WhatsApp i Gmail i prémer *Save*. Apareix la llista de la Figura 42.
- Després vaig clicar a *Gmail > Values* i vaig escollir el valor 0 (Figura 43).
- Igualment, vaig clicar a *WhatsApp > Values* i vaig indicar el valor 1. (Figura 44).
- Finalment, cada vegada que connectava la polsera havia de prémer la icona del Bluetooth i seleccionar el rellotge, com podem veure a la Figura 45.

26 <https://play.google.com/store/apps/details?id=notiduino.geny.code.main&hl=es>

A Notiduino vaig programar que un WhatsApp equivalgui a un 1 i un correu a un 0. Depenent de quin rebi, s'activarà el codi de pantalla emergent del WhatsApp o de Gmail:

```
void whatsappcode(){
  WHATSAPP++;
  Serial.print("whatsapp ");
  Serial.println(WHATSAPP);

  {if(millis() <= (tempsara2 + periode2)){ //Pantalla es mostrarà tres segons
    desar = 1; //Condicció que no apareguin les pantalles dels sensors

    display.clearDisplay();
    display.setCursor(1,5);
    display.setTextSize(1);
    display.setTextColor(WHITE);
    display.println("TENS");
    display.setCursor(28,5);
    display.println(WHATSAPP);
    display.setCursor(1,15);
    if (WHATSAPP == 1) {
      display.println("WHATSAPP");
    }
    else {display.println("WHATSAPPS");}
    display.drawBitmap(2,28,WHATSAPPICON,27,48,1);
    display.display();
  }else{desar = 0;} //Condicció que torni a aparèixer la pantalla d'abans
  }
}
```

```
if (ESP_BT.available()){ //Comprovem si rebem alguna cosa per Bluetooth
  incoming = ESP_BT.read(); //Llegim el que ha arribat i ho guardem a la variable "incoming"
  tempsara2 = millis();
  if(incoming == 1){
    whatsappcode();}
  if(incoming == 0){
    gmailcode();}
}
```

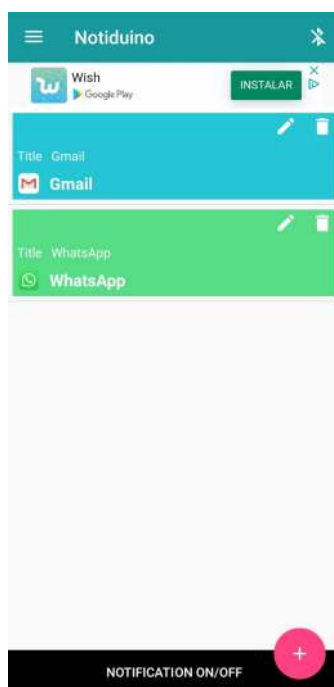
CONSTRUCCIÓ D'UNA POLSERA INTEL·LIGENT AMB ARDUINO

Nora Bolívar López

A la pantalla de les notificacions, Figura 46, apareixen el nombre de WhatsApps i de correus de Gmail que hem rebut. Si es pressiona durant dos segons el botó A, tornen a 0.

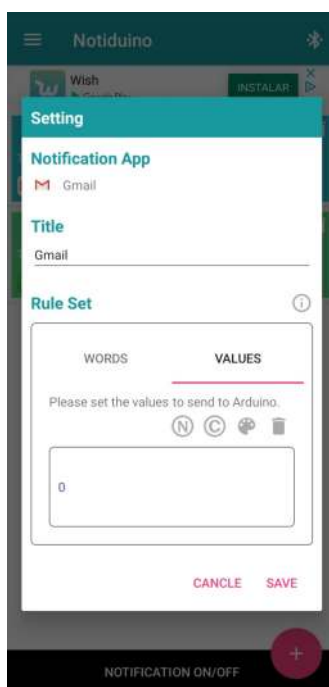
Quan es rebí un missatge, apareixerà una pantalla emergent durant tres segons i el brunzidor s'activarà. Si és de Gmail apareixerà a l'OLED la Figura 47 i si és un WhatsApp apareixerà la Figura 48.

Figura 42. Notiduino. Llista de les nostres aplicacions



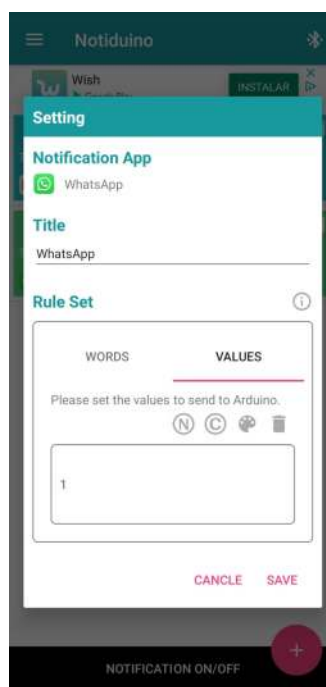
Imatge de l'autora

Figura 43. Notiduino. Gmail



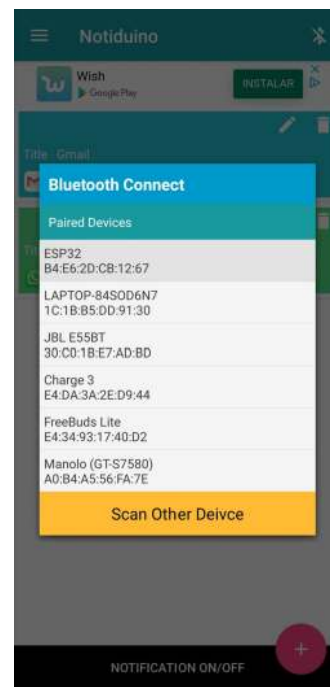
Imatge de l'autora

Figura 44. Notiduino. WhatsApp



Imatge de l'autora

Figura 45. Notiduino. Llistat dispositius Bluetooth



Imatge de l'autora

Figura 46. Novena pantalla. Llistat de notificacions



Imatge de l'autora

Figura 47. Pantalla emergent de correus de Gmail



Imatge de l'autora

Figura 48. Pantalla emergent de WhatsApp



Imatge de l'autora

6.9 Alarma

Per fer l'alarma vaig crear 6 noves variables:

```
int horesalarma = 0;    //Hora en la qual s'activarà l'alarma
int minutsalarma = 0;   //Minut en el qual s'activarà l'alarma
int segonsalarma = 00;  //Sempre s'activarà en el segon 0
int num = 0;            //Hi ha tres posicions en la pantalla de l'alarma. Amb el botó A canviarà.
                        //Valors: 0 (on/off), 1 (hores) i 2 (minuts).
bool alarma = false;    //Condicció d'estar l'alarma activada (true) o desactivada (false)
bool alarmaonoff = false; //Condicció de saltar l'alarma
```

Per saber en quin menú estem apareixerà una línia a sota. Si estem en l'alarma *off* / *on* hem de clicar breument al botó A per activar o desactivar l'alarma, com indica a la Figura 49 i 50. Com es pot veure al següent codi, canviarà el valor de la variable "alarma":

```
if (alarma == false){display.println("ALARMA OFF");}
if (alarma == true){display.println("ALARMA ON");}
```

Per ajustar l'hora hem de clicar dues vegades seguides al botó "A". Llavors apareixerà una línia sota les hores. Per avançar hem de clicar breument al botó "A" per afegir una hora, o bé, mantenir premut el botó A per afegir-ne 3. Per canviar els minuts, hem de tornar a prémer dues vegades seguides el botó A i per ajustar-los s'ha de seguir les mateixes instruccions que per les hores.

Quan l'alarma estigui activada i coincideixin les hores, els minuts i els segons de l'hora amb les variables de l'alarma, el brunzidor s'activarà i apareixerà la finestra emergent de la Figura 51, que no desapareixerà fins que no es premi el botó "A":

```
if (alarma == true){ //Alarma activada
if (((now.hour())==horesalarma)&& ((now.minute())==minutsalarma) && ((now.second())==((segonsalarma))){
    alarmaonoff = true; //S'activarà la pantalla emergent
    while(alarmaonoff == true){
        display.clearDisplay();
        display.drawBitmap (0,0,ALARMICON,64,48,1);
        display.display();
```

```

if (alarma == true){ //Alarma activada
if (((now.hour()) == horesalarma) && ((now.minute()) == minutsalarma) && ((now.second()) == (segonsalarma))){
    alarmaonoff = true; //S'activarà la pantalla emergent
    while(alarmaonoff == true){display.clearDisplay();
    display.drawBitmap (0,0,ALARMICON,64,48,1);display.display();

    if (button.get() == 0){
        switch (button.BUTTON_A){
        case KEY_VALUE_SHORT_PRESS:
            if (alarmaonoff == true){alarmaonoff = false;} //Fins que no es premi el
            botó A no desapareixerà i la variable "alarmaonoff" seguirà sent "true"
            break;}}}}

```

Figura 49. Pantalla desena.
Alarma desactivada



Imatge de l'autora

Figura 50. Pantalla desena.
Alarma activada



Imatge de l'autora

Figura 51. Pantalla emergent
de l'alarma



Imatge de l'autora

6.10 Brunzidor

Quan vaig acabar de programar de l'alarma i les notificacions, vaig crear el del brunzidor. Per fer-ho més senzill, vaig fer servir una variable anomenada "desar". Aquesta només pot ser 0 o 1. Serà 1 quan l'alarma soni, i quan es desactivi passarà a ser 0. També equivaldrà a 1 quan la notificació de WhatsApp o Gmail aparegui.

```

if (desar == 0){digitalWrite(pinbuzzer,LOW);}
if (desar == 1){beep();}

```

```

void beep(){
    digitalWrite(pinbuzzer, HIGH);
    Serial.println("HIGH");
    if(millis() > tempsarabuzzer + periodbuzzer){
        tempsarabuzzer = millis();
        digitalWrite(pinbuzzer, LOW);
        Serial.println("LOW");}}

```

6.11 Programa final

Una vegada vaig tenir tots els programes per separat, vaig muntar-ho tot. Primer vaig posar totes les variables i llibreries, incloses totes les imatges. Al "void setup" també vaig inserir totes les funcions que iniciaven la comunicació serial, el "wire" i tots els sensors que faig servir. Al "void loop" no s'executa cap programa, només crida altres "voids" si es compleix una condició. Per passar les pantalles i activar el codi corresponent, faig servir una variable anomenada "number", la qual té un valor entre el 0 i el 10:

```
if (number == 11) number = number - 11; // Com hi ha 10 pantalles, quan estigui a l'última i es vulgui avançar,
tornarà a la primera
if (number == 255) number = number - 245; // Quan es vulgui retrocedir i estigui a la primera pantalla, anirà al
final
if (number == 0) { // La pantalla està apagada
  display.clearDisplay();
  display.display();
}
if ((number == 1) && (desar == 0)) { // S'activarà el void del rellotge.
  rellotge();
}
if ((number == 2) && (desar == 0)) { // S'activarà el void de la freqüència cardíaca
  cor();
}
if ((number == 3) && (desar == 0)) { // S'activarà el void de la brúixola
  bruixola();
}
if ((number == 4) && (desar == 0)) { // S'activarà el void de les plantes pujades
  altimetre();
}
if ((number == 5) && (desar == 0)) { // S'activarà el void de la humitat relativa
  humi();
}
if ((number == 6) && (desar == 0)) { // S'activarà el void de la temperatura
  temper();
}
```

```
if ((number == 7) && (desar == 0)) { // S'activarà el void de l'índex d'UV
  nivellradiacio();
}
if ((number == 8) && (desar == 0)) { // S'activarà el void del cronòmetre
  crono();
}
if ((number == 9) && (desar == 0)) { // S'activarà el void de les notificacions
  notificacions();
}
if ((number == 10) && (desar == 0)) { // S'activarà el void de l'alarma
  alarmacode();
}
```

Per canviar aquesta variable, al "void loop" hi ha col·locat una primera part del codi del sensor de gestos, que quan detecti un gest, activarà un altre codi que detectarà si la mà passa de dreta a esquerra (es restarà 1) o si passa d'esquerra a dreta (se sumarà 1). A més, també activarà un altre codi quan detecti un botó. El botó B funcionarà per avançar, però el botó A només funcionarà al cronòmetre, a les notificacions i a l'alarma.

Figura 52. Imatge de tots els components connectats a la placa PCB



Imatge de l'autora

Per poder connectar tots els components he fet servir una placa de coure PCB perforada en línia²⁷, tal com es pot veure a la Figura 52.




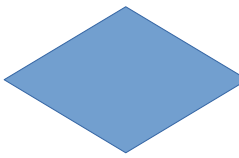


El codi final està disponible a

"https://drive.google.com/drive/folders/15hAiGT7p_001NczTIN9Gy5SgqaiKWSb?usp=sharing"

²⁷ M'ha servit per connectar alhora tots els sensors, ja que tots requerien els mateixos pins i no hi havia tants a la placa com necessitava.

6.12 Diagrama de flux

Per comprendre millor el codi final, he construït un diagrama de flux, que seria una representació gràfica del meu programa. M'he basat en la següent simbologia:

SÍMBOL	SIGNIFICAT
	Inici: iniciem el programa
	Entrada o sortida: rebem o enviem dades a un perifèric o via Bluetooth. Exemple: rebem la temperatura del sensor corresponent.
	Activitat: són els processos que es duen a terme. Exemple: convertim les mesures del magnetòmetre en un angle a través de l'arctangent.
	Decisió: la decisió es pot fer segons si es compleix una condició sí o no, o també depenent del valor d'una variable. Exemple: s'ha pressionat algun botó?
	Visualització a la pantalla: les dades o imatges que es mostren a l'OLED. Exemple: la temperatura i la imatge d'un termòmetre.
	Línia: indica la direcció i el sentit de les diferents parts del programa

Al programa hi ha diferents processos que es duen a terme alhora i ho he representat amb diferents colors per aclarir-ho:

- NEGRE: condicions inicials.
- VERMELL: comptador de plantes pujades.
- TARONJA: processos que van amb Bluetooth (notificacions i aplicació).
- VERD: el cronòmetre va comptant encara que hi hagi una altra pantalla.
- BLAU CEL: salta l'alarma quan sigui l'hora.
- BLAU FOSC: comprova els botons per canviar de pantalla.
- LILA: comprova el sensor de gestos per canviar de pantalla.
- ROSA: comprova la variable que controla les pantalles.

```

graph TD
    INICI([INICI]) --> Encenem[Encenem els sensors]
    Encenem --> PantallaNegra([Pantalla negra])
    PantallaNegra --> ArribaInfo{Arriba informació pel Bluetooth?}
    ArribaInfo -- No --> ArribaInfo
    ArribaInfo -- Sí --> Llegim[Llegim la informació]
    Llegim --> D2{2}
    Llegim --> D1{1}
    Llegim --> D0{0}
    Llegim --> Altres{Altres}
    Altres --> Exit(( ))
    D2 --> EnviamPulsacions[/Enviem per Bluetooth les pulsacions i "|"/]
    D1 --> AfegimWhatsApp[Afegim un WhatsApp al comptador]
    D0 --> AfegimCorreu[Afegim un correu al comptador]
    EnviamPulsacions --> EnviamPlantes[/Enviem per Bluetooth les plantes pujades i "|"/]
    AfegimWhatsApp --> PantallaEmergent1{{Pantalla emergent i vibració}}
    AfegimCorreu --> PantallaEmergent2{{Pantalla emergent i vibració}}
    PantallaEmergent1 -- 3s --> Desapareix1{{Desapareix aquesta pantalla i la vibració cessa}}
    PantallaEmergent2 -- 3s --> Desapareix2{{Desapareix aquesta pantalla i la vibració cessa}}
    EnviamPlantes --> EnviamTemperatura[/Enviem per Bluetooth la temperatura i "|"/]
    EnviamTemperatura --> EnviamHumiditat[/Enviem per Bluetooth la humitat relativa i "|"/]
    EnviamHumiditat --> Exit
    ObtenimA[/Obtenim la pressió (A)/] -- 15s --> ObtenimAA[/Obtenim la pressió (AA)/]
    ObtenimAA --> CalcAlçada[Calculem l'altura pujada]
    CalcAlçada --> S'haPujat{S'ha pujat entre 2,5 i 6 m?}
    S'haPujat -- Sí --> SumemPlanta[Sumem una planta pujada al comptador]
    SumemPlanta --> ObtenimA
    S'haPujat -- No --> ArribaInfo

```

Figura 54. Diagrama de flux 2

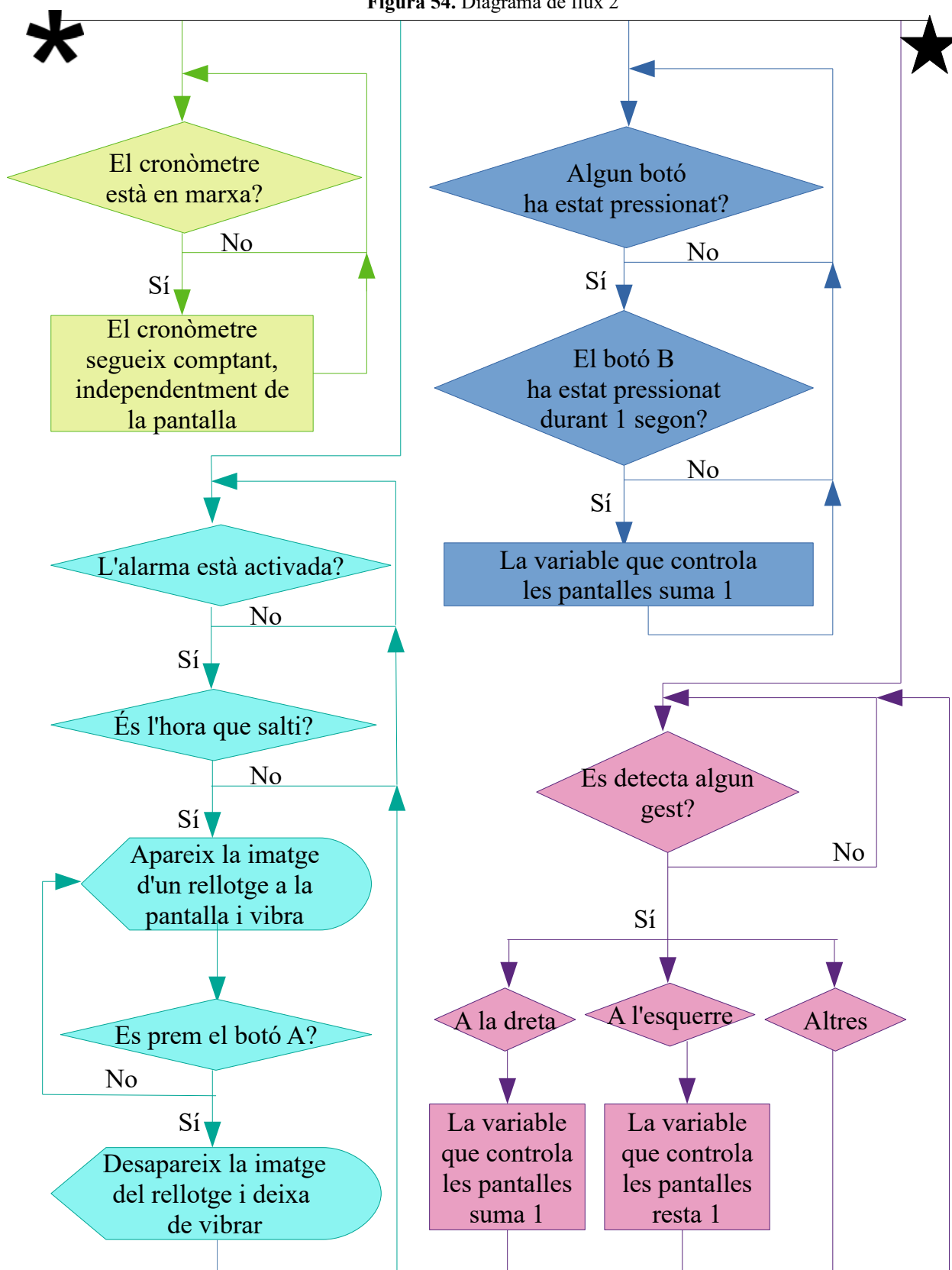
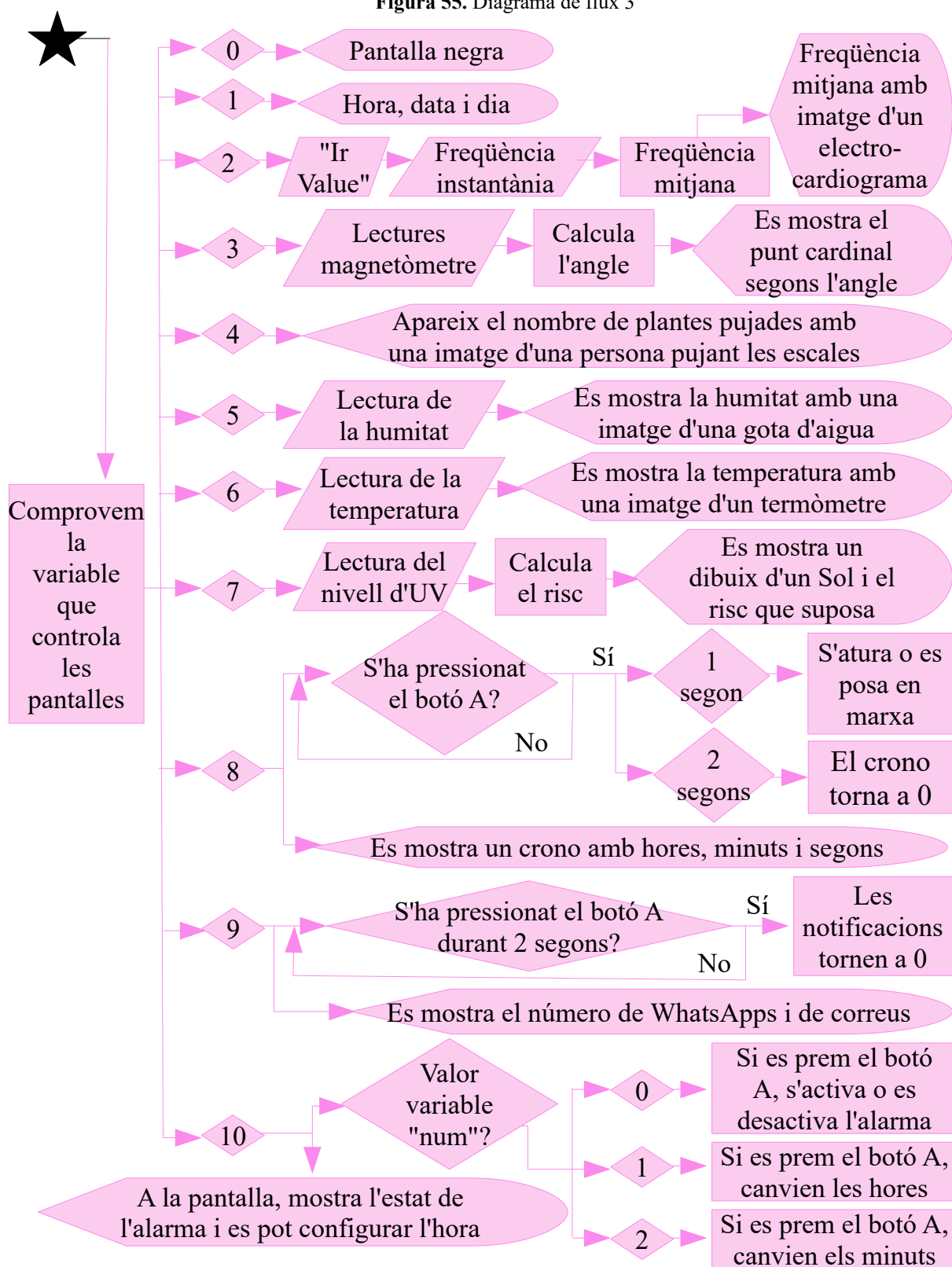


Figura 55. Diagrama de flux 3



Imatge de l'autora

7 APP INVENTOR

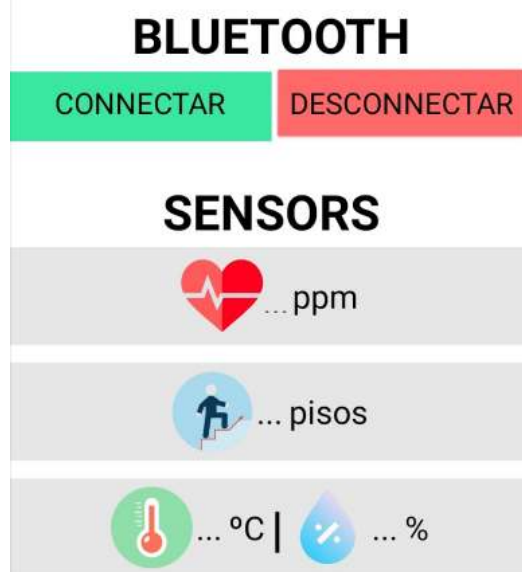
App Inventor és una web creada per Google Labs on desenvolupar aplicacions per Android. Està pensada per a persones sense gaires coneixements en informàtica, ja que no necessites programar, sinó que has d'unir blocs.

Per complementar el rellotge, he dissenyat una aplicació a través d'aquesta web. A la part superior de l'aplicació, hi ha l'apartat del Bluetooth. Hi ha dos botons: "CONNECTAR", en el qual apareixerà un llistat de dispositius Bluetooth disponibles, i "DESCONNECTAR", on en prémer es desconnectarà del rellotge.

Les lectures dels sensors apareixeran més avall: freqüència cardíaca, els pisos pujats, la temperatura i la humitat relativa.

L'aplicació és molt simple però el disseny és bonic i és molt fàcil de fer servir.

Figura 56. Aplicació pel mòbil

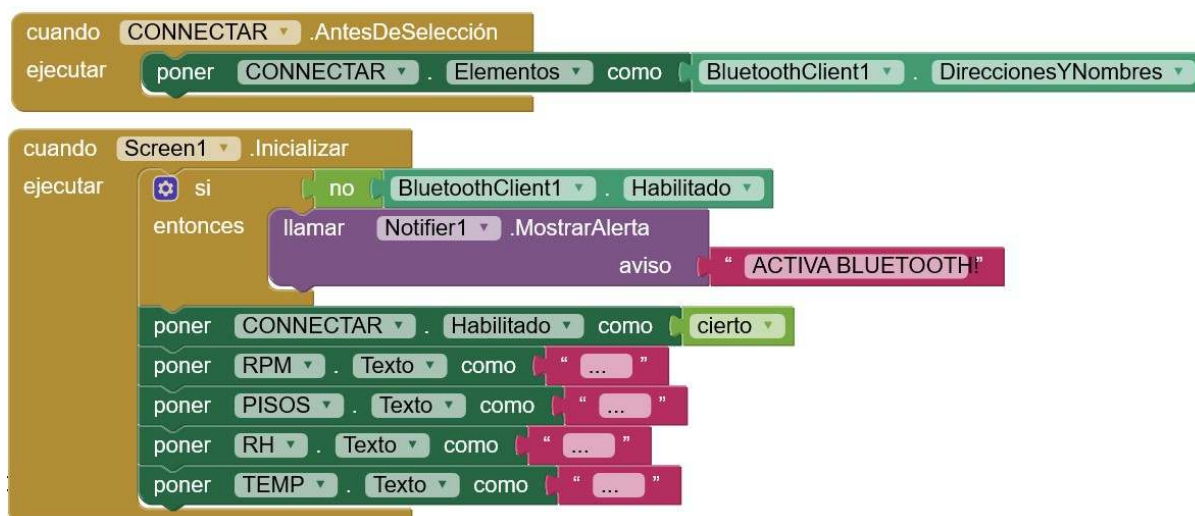


Imatge de l'autora

Per crear una aplicació a App Inventor hi ha dues parts:

- El disseny, on es van afegint els components, com per exemple els botons²⁸ i les etiquetes²⁹. És molt fàcil de fer servir perquè només s'ha d'anar arrossegant des de la paleta fins al visor. Aquests elements s'afegiran als components, des d'on es podran modificar les seves propietats tals com el color o el nom.
- Els blocs, on es pot programar l'aplicació. Primer vaig configurar el Bluetooth. Si el Bluetooth del mòbil està desactivat, es mostrarà un missatge d'alerta: "ACTIVA BLUETOOTH". Si està activat, el botó "CONNECTAR" es podrà prémer però no el de "DESCONNECTAR". Quan es premi "CONNECTAR" apareixerà el llistat de dispositius Bluetooth.

Figura 57. App Inventor. Bloc 1



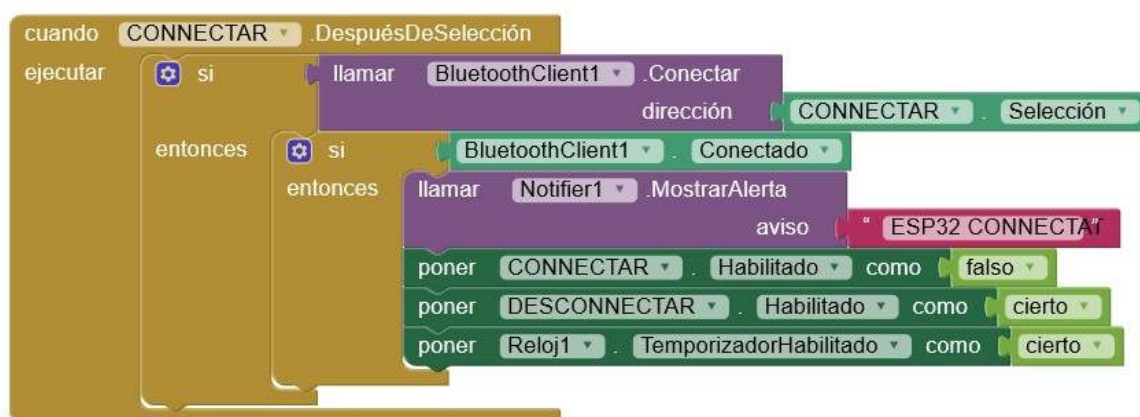
Imatge de l'autora

28 Els botons detecten quan es fa clic sobre ells. Poden canviar la seva aparença i es poden deshabilitar.

29 A les etiquetes es mostra text. Poden anar canviant depenent de la configuració.

Quan s'esculli el rellotge, apareixerà l'alerta "ESP32 CONNECTAT". A més, ja no es podrà prémer el botó "CONNECTAR" i sí el de "DESCONNECTAR". També s'habilitarà el rellotge, el qual és un temporitzador que salta cada 20 segons i enviarà el número "2" al rellotge.

Figura 58. App Inventor. Bloc 2



Imatge de l'autora

- Quan es desconnecti el rellotge, el botó "DESCONNECTAR" no es podrà polsar i el botó "CONNECTAR" sí.

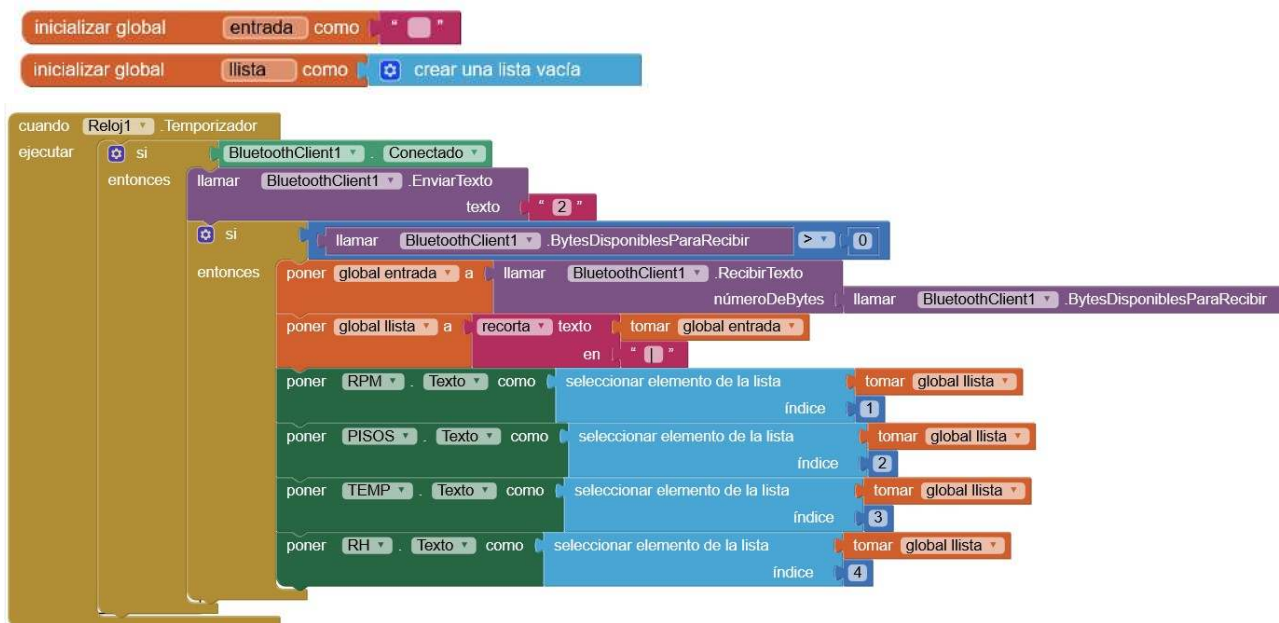
Figura 59. App Inventor. Bloc 3



Imatge de l'autora

- L'"ESP32", quan rebí per Bluetooth el número "2", enviarà la freqüència cardíaca, les plantes pujades, la temperatura i la humitat relativa. Les dades estaran separades per una barra, "|". He creat dues variables a l'aplicació: "entrada", la qual primer rep totes les dades, i "llista", la qual processa la variable anterior i detecta les barres. El primer element seran les pulsacions, després els pisos pujats, i així successivament.

Figura 60. App Inventor. Bloc 4



Imatge de l'autora

8 CONCLUSIÓ I VALORACIÓ

La hipòtesi d'aquest treball era saber si es pot realitzar una polsera intel·ligent amb l'entorn d'Arduino. De la part de software estic molt satisfeta perquè he avançat molt i he adquirit coneixements que em seran molt útils. Al principi anava molt perduda perquè no tenia nocions bàsiques i necessitava molt de temps per comprendre el codi, però al final del projecte, ja podia escriure el codi sense ajuda.

El rellotge funciona amb codi però no amb l'RTC, ja que necessita 5 V i una bateria no proporciona aquest voltatge. En una futura versió necessitaria un rellotge que funcionés amb 3,3 V. El codi del pulsòmetre funciona de forma individual, però no de forma conjunta. He intentat buscar alternatives però no les he trobades. La brúixola funciona bé, com també el sensor de temperatura i humitat relativa, el comptador de plantes pujades, el risc d'UV, el cronòmetre i l'alarma. Possibles millores són: que es pugui configurar més d'una alarma i poder configurar notificacions d'altres aplicacions.

La part de hardware l'he deixada de banda i és un aspecte a millorar, ja que es podria fer una versió molt més compacta que permetés posar una corretja. Però no era un objectiu del meu treball, ja que al principi ho vaig intentar, però el Jordi Orts em va fer veure que no era factible i que m'havia d'enfocar en el codi, ja que era un prototip. També vaig intentar programar una aplicació que enviés les notificacions del telèfon i que aquesta rebés les dades del rellotge, però no la vaig arribar a fer per manca de temps.

La intenció era que s'assemblés a un model comercial. Per una banda, crec que les pantalles s'assemblen als models més senzills. Però d'altra banda, els components eren massa grans per un prototip més compacte.

Com a valoració final, estic molt contenta del resultat i de tot el que he après. Ha valgut la pena, tot i els moments més difícils on tenia la sensació d'estar en un carreró sense sortida.

AGRAÏMENTS

En primer lloc, m'agradaria donar les gràcies al Jordi Orts per proporcionar-me tot el material necessari i per ajudar-me a resoldre els meus dubtes.

També vull agrair al Jose Toro per haver revisat i corregit la memòria escrita.

BIBLIOGRAFIA

ADAFRUIT. A fork of Jeelab's fantastic RTC library [en línia]. <<https://github.com/adafruit/RTClib>> [Consulta: 02/09/19].

AEMET. Interpretación: Radiación Ultravioleta (UVI) [en línia]. <<http://www.aemet.es/es/eltiempo/prediccion/radiacionuv/ayuda>> [Consulta: 06/10/19].

ARDUINO PARA TODOS. Tutorial MIT App Inventor con Arduino y Bluetooth. Varios ejemplos [en línia]. 09/07/17. <<https://www.youtube.com/watch?v=OO7vKKuJ9a0>> [Consulta: 10/08/19].

ARDUINO. Download the Arduino IDE [en línia]. <<https://www.arduino.cc/en/Main/Software>> [Consulta: 22/06/19].

ARDUINO. Wire Library [en línia]. <<https://www.arduino.cc/en/reference/wire>> [Consulta: 27/08/19].

GNU. What is free software?. [en línia]. <<https://www.gnu.org/philosophy/free-sw.es.html>> [Consulta: 30/09/19].

JAVL. image2cpp. [en línia]. 22/01/16 . <<http://javl.github.io/image2cpp>> [Consulta: 01/09/19].

JOEL_E_B. Si7021 Humidity and Temperature Sensor Hookup Guide. [en línia]. <https://learn.sparkfun.com/tutorials/si7021-humidity-and-temperature-sensor-hookup-guide?_ga=2.35813344.1953479569.1570642066-1367301285.1566812262> [Consulta: 09/10/19].

LADY ADA. Adafruit Si7021 Temperature + Humidity Sensor. [en línia]. 22/09/16. <<https://learn.adafruit.com/adafruit-si7021-temperature-plus-humidity-sensor/overview>> [Consulta: 09/10/19].

LLAMAS, LUIS. Medir presión del aire y altitud con Arduino y barómetro BMP180 [en línia]. 13/09/16. <<https://www.arduino.cc/en/Main/Software>> [Consulta: 22/07/19].

MIKEGRUSIN. BMP180 Barometric Pressure Sensor Hookup. [en línia].
<[https://learn.sparkfun.com/tutorials/bmp180-barometric-pressure-sensor-hookup-?_ga=2.219340176.809589358.1570284787-1367301285.1566812262](https://learn.sparkfun.com/tutorials/bmp180-barometric-pressure-sensor-hookup-_ga=2.219340176.809589358.1570284787-1367301285.1566812262)> [Consulta: 06/10/19].

MUSEU NACIONAL D'ART DE CATALUNYA. Humitat relativa. [en línia].
<<https://www.museunacional.cat/ca/humitat-relativa>> [Consulta: 09/10/19].

NATE. MAX30105 Particle and Pulse Ox Sensor Hookup Guide. [en línia].
<<https://learn.sparkfun.com/tutorials/max30105-particle-and-pulse-ox-sensor-hookup-guide>>
[Consulta: 05/10/19].

NO, DANI. PRÁCTICA 5: Funcionamiento de una pantalla OLED con el procesador ESP8266. [en línia]. 14/11/16. <<https://www.esploradores.com/practica-5-funcionamiento-de-una-pantalla-oled-con-el-procesador-esp8266/>> [Consulta: 01/09/19].

PÉREZ, GUILLERMO. Tutorial Arduino: Entradas Analógicas y Digitales. [en línia]. 22/04/15.
<<https://openwebinars.net/blog/tutorial-arduino-entradas-analogicas-y-digitales/>> [Consulta: 30/09/19].

RAJ, ASWINTH. How to Use Serial Bluetooth in ESP32. [en línia]. 16/10/18. <<https://circuitdigest.com/microcontroller-projects/using-classic-bluetooth-in-esp32-and-toogle-an-led>> [Consulta: 02/08/19].

ROBOT ELECTRONICS. Using the I2C Bus [en línia]. <<https://robot-electronics.co.uk/i2c-tutorial>>
[Consulta: 27/08/19].

ROWBERG, JEFF. I2C device library collection for AVR/Arduino or other C++-based MCUs [en línia].
<<https://github.com/jrowberg/i2cdevlib>> [Consulta: 07/09/19].

SANTOS, RUI. Guide for I2C OLED Display with Arduino [en línia].
<<https://randomnerdtutorials.com/guide-for-oled-display-with-arduino/>> [Consulta: 01/09/19].

CONSTRUCCIÓ D'UNA POLSERA INTEL·LIGENT AMB ARDUINO

Nora Bolívar López

SOTO, MARGARET. ¿Qué significa “Dual Core” y Quad Core”? Para qué te sirve saberlo?. [en línia]. 22/09/16. <<https://tecnobitt.com/que-significa-dual-core-y-quad-core-para-que-te-sirve-saberlo/>> [Consulta: 05/11/19].

SPARKFUN. An Arduino Library for the MAX3015 particle sensor and MAX30102 Pulse Ox sensor. <https://github.com/sparkfun/SparkFun_MAX3010x_Sensor_Library> [Consulta: 17/07/19].

SPARKFUN. Arduino libraries for the BMP180 pressure sensor breakout board [en línia]. <https://github.com/sparkfun/BMP180_Breakout_Arduino_Library> [Consulta: 22/07/19].

SPARKFUN. Arduino library for Adafruit Si7021. [en línia]. <https://github.com/adafruit/Adafruit_Si7021> [Consulta: 02/09/19].

SPARKFUN. UV(Ultraviolet light) sensor. [en línia]. <https://github.com/Seeed-Studio/Seeed_VEML6070> [Consulta: 24/07/19].

TESLABEM. Fundamentos del Protocolo I2C – Aprende. [en línia]. 04/02/17. <<https://teslabem.com/nivel-intermedio/fundamentos-del-protocolo-i2c-aprende/#>> [Consulta: 27/08/19].

WEMOS. Arduino core for ESP32 WiFi chip [en línia]. 06/10/16. <https://github.com/wemos/Arduino_ESP32/blob/master/docs/arduino-ide/boards_manager.md> [Consulta: 22/06/19].

WEMOS. Installation instructions using Arduino IDE Boards Manager [en línia]. 26/07/18. <https://github.com/wemos/Arduino_ESP32/blob/master/docs/arduino-ide/boards_manager.md> [Consulta: 22/06/19].

WEBS DE COMPRA

ALIEXPRESS. Sensor de freqüència cardíaca (MAX30102) [en línia] <https://es.aliexpress.com/item/32902336311.html?spm=sns_none&aff_platform=default&cpt=15554299>

CONSTRUCCIÓ D'UNA POLSERA INTEL·LIGENT AMB ARDUINO

Nora Bolívar López

12269&sk=MoSiTPL&aff_trace_key=e78a7d7f54314de3bce947ca05bdd72a-1555429912269-03787-MoSiTPL&terminal_id=bc8057ef5ea343e6b2630c4b02bfe9a0>

ALIEXPRESS. SENSOR PRESSIÓ BAROMÈTRICA (BMP180) [en línia]

<[https://es.aliexpress.com/item/32519558222.html?](https://es.aliexpress.com/item/32519558222.html?spm=a2g0o.productlist.0.0.489d3818tcAiN6&algo_pvid=ed0a4858-e421-4c1c-ab09-ff98be2e43de&algo_expid=ed0a4858-e421-4c1c-ab09-ff98be2e43de-8&btsid=a85abf00-cf2b-4fb8-9968-643cf5f62bf8&ws_ab_test=searchweb0_0,searchweb201602_4,searchweb201603_60)

spm=a2g0o.productlist.0.0.489d3818tcAiN6&algo_pvid=ed0a4858-e421-4c1c-ab09-ff98be2e43de&algo_expid=ed0a4858-e421-4c1c-ab09-ff98be2e43de-8&btsid=a85abf00-cf2b-4fb8-9968-643cf5f62bf8&ws_ab_test=searchweb0_0,searchweb201602_4,searchweb201603_60>

ALIEXPRESS. SENSOR PRESSIÓ BAROMÈTRICA (BMP180) [en línia]

<[https://es.aliexpress.com/item/32519558222.html?](https://es.aliexpress.com/item/32519558222.html?spm=a2g0o.productlist.0.0.489d3818tcAiN6&algo_pvid=ed0a4858-e421-4c1c-ab09-ff98be2e43de&algo_expid=ed0a4858-e421-4c1c-ab09-ff98be2e43de-8&btsid=a85abf00-cf2b-4fb8-9968-643cf5f62bf8&ws_ab_test=searchweb0_0,searchweb201602_4,searchweb201603_60)

spm=a2g0o.productlist.0.0.489d3818tcAiN6&algo_pvid=ed0a4858-e421-4c1c-ab09-ff98be2e43de&algo_expid=ed0a4858-e421-4c1c-ab09-ff98be2e43de-8&btsid=a85abf00-cf2b-4fb8-9968-643cf5f62bf8&ws_ab_test=searchweb0_0,searchweb201602_4,searchweb201603_60>

ALIEXPRESS. SENSOR UV (VEML6070) [en línia] <[https://es.aliexpress.com/item/32700670830.html?](https://es.aliexpress.com/item/32700670830.html?spm=a2g0o.productlist.0.0.339e462cJxaMri&algo_pvid=26a2f0e8-a6f6-4223-b168-ea6d9dff3eea&algo_expid=26a2f0e8-a6f6-4223-b168-ea6d9dff3eea-8&btsid=62c538c9-f4b8-4ad9-a0f6-80c703735a7d&ws_ab_test=searchweb0_0,searchweb201602_4,searchweb201603_60)

spm=a2g0o.productlist.0.0.339e462cJxaMri&algo_pvid=26a2f0e8-a6f6-4223-b168-ea6d9dff3eea&algo_expid=26a2f0e8-a6f6-4223-b168-ea6d9dff3eea-8&btsid=62c538c9-f4b8-4ad9-a0f6-80c703735a7d&ws_ab_test=searchweb0_0,searchweb201602_4,searchweb201603_60>

ALIEXPRESS. SENSOR D'HUMITAT I TEMPERATURA (Si7021) [en línia]

<[https://es.aliexpress.com/item/33031108713.html?](https://es.aliexpress.com/item/33031108713.html?spm=a2g0o.productlist.0.0.dfd24abcohmjTu&algo_pvid=98b67c2a-85e9-45d4-8925-de4b550e3c02&algo_expid=98b67c2a-85e9-45d4-8925-de4b550e3c02-0&btsid=fb1b0169-2094-4f1b-b2b4-a9f68a4485f7&ws_ab_test=searchweb0_0,searchweb201602_5,searchweb201603_53)

spm=a2g0o.productlist.0.0.dfd24abcohmjTu&algo_pvid=98b67c2a-85e9-45d4-8925-de4b550e3c02&algo_expid=98b67c2a-85e9-45d4-8925-de4b550e3c02-0&btsid=fb1b0169-2094-4f1b-b2b4-a9f68a4485f7&ws_ab_test=searchweb0_0,searchweb201602_5,searchweb201603_53>

ALIEXPRESS. Triple eix brúixola-magnetòmetre (HMC5883L) [en línia]

<[https://es.aliexpress.com/item/32826264150.html?](https://es.aliexpress.com/item/32826264150.html?spm=a2g0o.productlist.0.0.161d36c3wKKWqH&algo_pvid=54cd3698-d83f-48df-9b87-7270f3b3f59f&algo_expid=54cd3698-d83f-48df-9b87-7270f3b3f59f-0&btsid=08949444-0a32-403e-b6ef-a80347f2bbb8&ws_ab_test=searchweb0_0,searchweb201602_5,searchweb201603_53)

spm=a2g0o.productlist.0.0.161d36c3wKKWqH&algo_pvid=54cd3698-d83f-48df-9b87-7270f3b3f59f&algo_expid=54cd3698-d83f-48df-9b87-7270f3b3f59f-0&btsid=08949444-0a32-403e-b6ef-a80347f2bbb8&ws_ab_test=searchweb0_0,searchweb201602_5,searchweb201603_53>

ALIEXPRESS. Sensor de gestos (PAJ7620) [en línia]

<[https://es.aliexpress.com/item/32969936474.html?](https://es.aliexpress.com/item/32969936474.html?spm=a2g0o.productlist.0.0.dfb9acjf4sZxe&algo_pvid=c9c87610-e32a-44f4-b9b5-)

spm=a2g0o.productlist.0.0.dfb9acjf4sZxe&algo_pvid=c9c87610-e32a-44f4-b9b5-

CONSTRUCCIÓ D'UNA POLSERA INTEL·LIGENT AMB ARDUINO

Nora Bolívar López

304473ba08a0&algo_expid=c9c87610-e32a-44f4-b9b5-304473ba08a0-17&btsid=90607535-b6b6-4517-ade3-e556e7e6d44e&ws_ab_test=searchweb0_0,searchweb201602_5,searchweb201603_53>

ALIEXPRESS. LOLIN D32 Pro [en línia] <<https://es.aliexpress.com/item/32883116057.html>>

ALIEXPRESS. Brunzidor [en línia] <https://es.aliexpress.com/item/32973874228.html?spm=a2g0o.productlist.0.0.63e53ddd3yZKKQ&algo_pvid=42bb98b5-ce3f-40e4-a2b6-dd5d685d1788&algo_expid=42bb98b5-ce3f-40e4-a2b6-dd5d685d1788-13&btsid=62993930-0347-4dcc-b240-22ef94cea351&ws_ab_test=searchweb0_0,searchweb201602_5,searchweb201603_53>

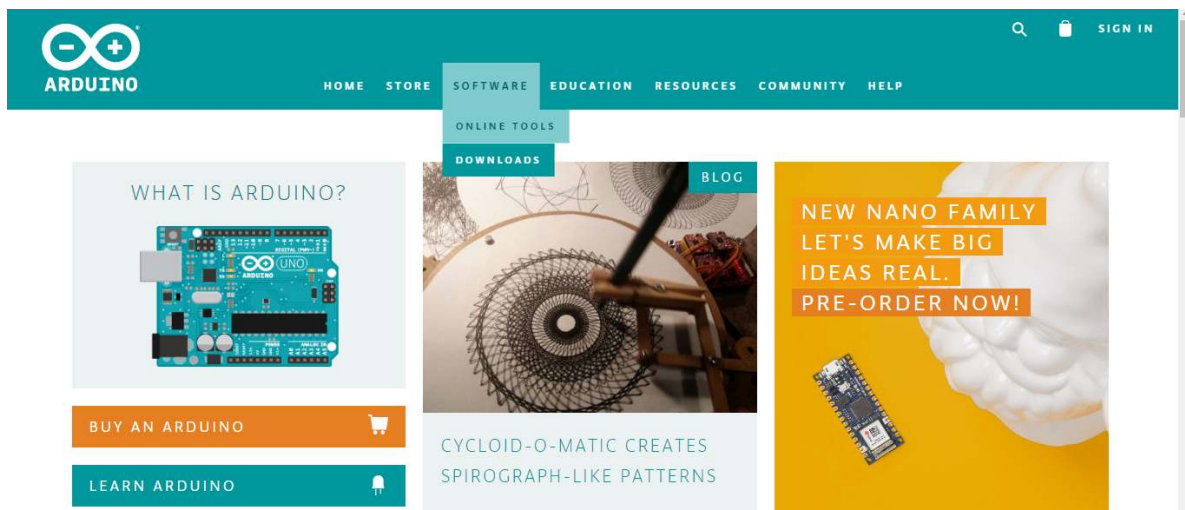
ALIEXPRESS. Cables "dupont" femella-femella [en línia] <https://es.aliexpress.com/item/33047998854.html?spm=a2g0o.productlist.0.0.77753337RirRKe&algo_pvid=3a43ae5e-a7d0-4bf8-89ec-c008d3f8aafd&algo_expid=3a43ae5e-a7d0-4bf8-89ec-c008d3f8aafd-1&btsid=665bc792-df9e-47bf-acf5-fb7d3d01d450&ws_ab_test=searchweb0_0,searchweb201602_5,searchweb201603_53>

9 ANNEXOS

9.1 ANNEX 1: instal·lar Arduino i configurar l'entorn

- Des de la pàgina web d'Arduino, descarreguem l'aplicació des de l'apartat de *Software* > *Downloads*.

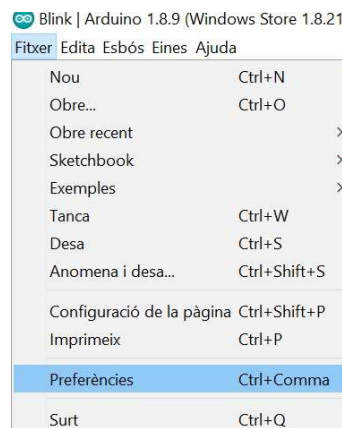
Figura 61. Pàgina web d'Arduino



Imatge de l'autora

- Obrim les preferències d'Arduino: *Fitxer* > *Preferències*.

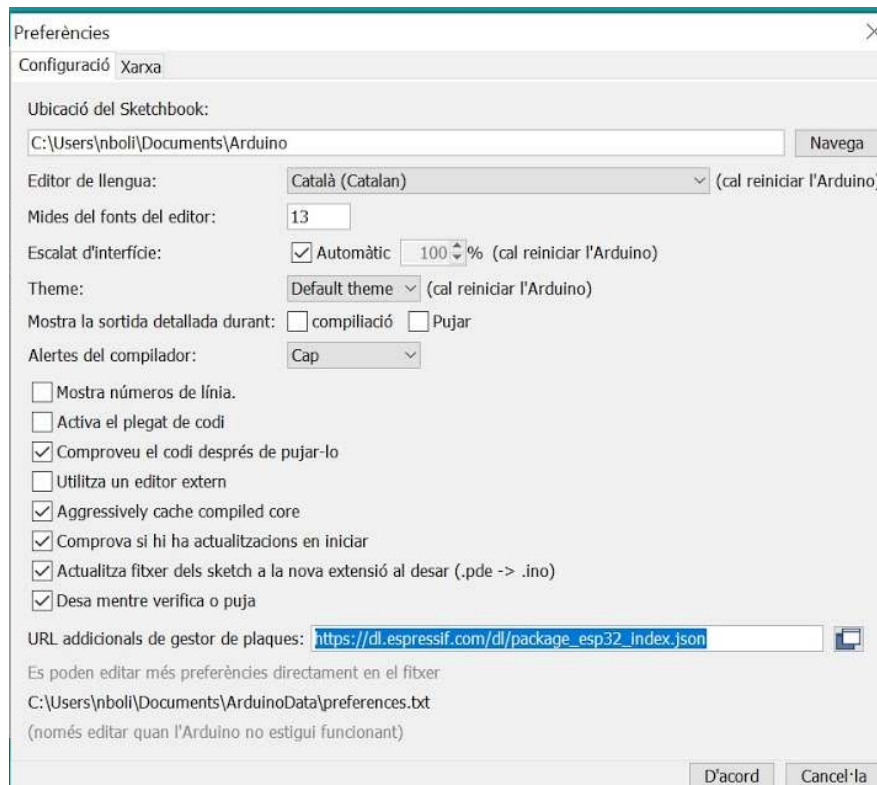
Figura 62. Preferències d'Arduino



Imatge de l'autora

- Enganxem l'URL proporcionat per la pàgina de Github³⁰ (https://dl.espressif.com/dl/package_esp32_index.json) a l'apartat de gestió de plaques.

Figura 63. URL addicionals de gestor de plaques



Imatge de l'autora

- Configurarem la placa instal·lant el paquet de plaques d'esp32 proporcionat per Espressif Systems: *Eines > Placa > Gestor de plaques > esp32*.

Figura 64. Gestor de plaques

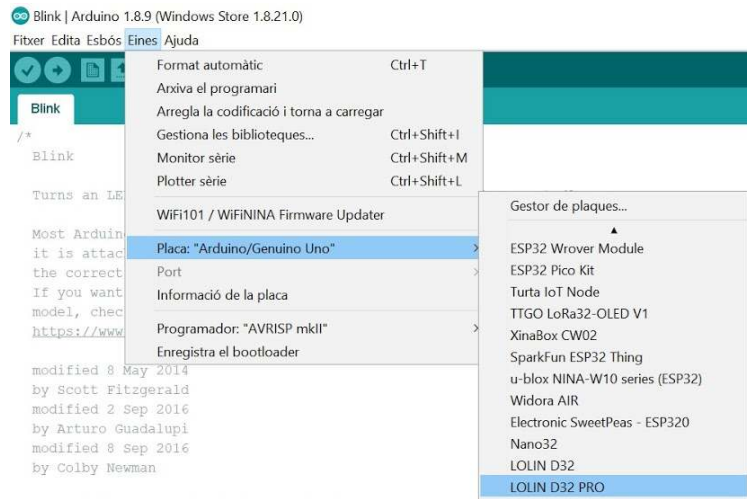


Imatge de l'autora

³⁰ "https://github.com/wemos/Arduino_ESP32/blob/master/docs/arduino-ide/board_manager.md"

- Ara ens apareixen noves plaques de l'"ESP32" i escollirem la "LOLIN D32 PRO":
Eines > Placa > ESP32 Arduino: LOLIN D32 PRO.

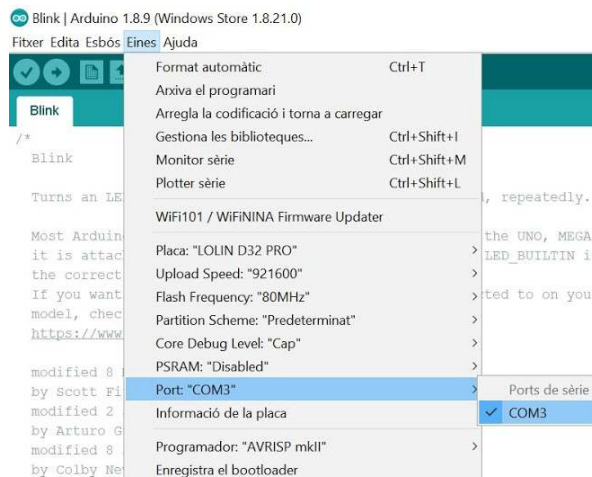
Figura 65. Escollir LOLIN D32 PRO



Imatge de l'autora

- També haurem de configurar el port. En el meu cas és el COM3: *Eines > Port > COM3.*

Figura 66. Escollir COM3 com a port



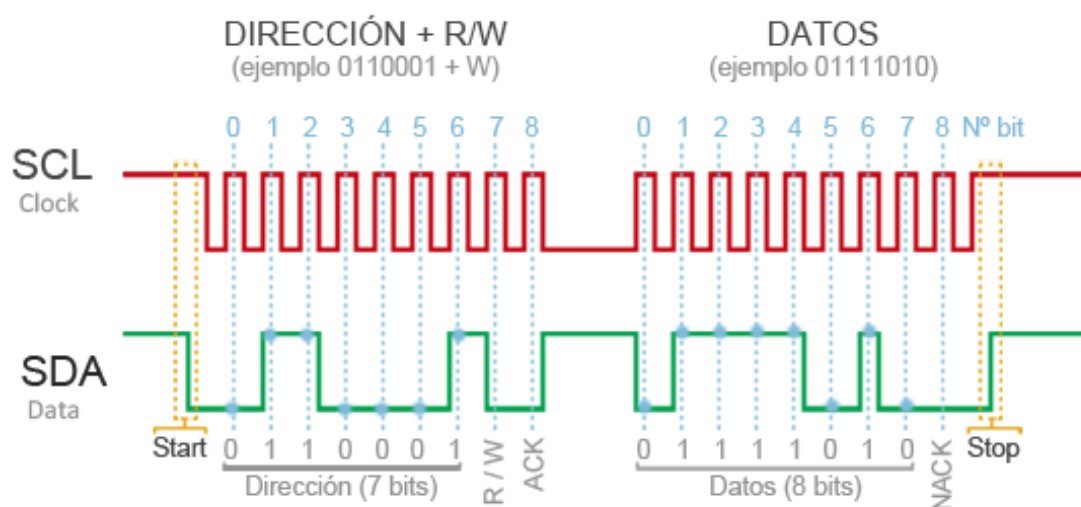
Imatge de l'autora

9.2 ANNEX 2: Què és l' I²C i com ho apliquem a Arduino

El I²C és un protocol de comunicació creat per Philips a principis dels vuitanta per poder comunicar múltiples xips entre si en els seus televisors. Serveix per comunicar un o més mestres amb un o diversos esclaus. Es caracteritza per ser serial i síncron, és a dir, que les dades només circulen per una via i està sincronitzat per un rellotge compartit.

El protocol fa servir dues vies per comunicar-se: el SDA i el SCL. El *Serial Data*, SDA, és la via per on es transmeten les dades. El *Serial Clock*, SCL, és la via per on viatja el senyal del rellotge.

Figura 67. SDA i SCL. Primer envia la direcció i informa que enviarà dades. Després es mostra com envia dades

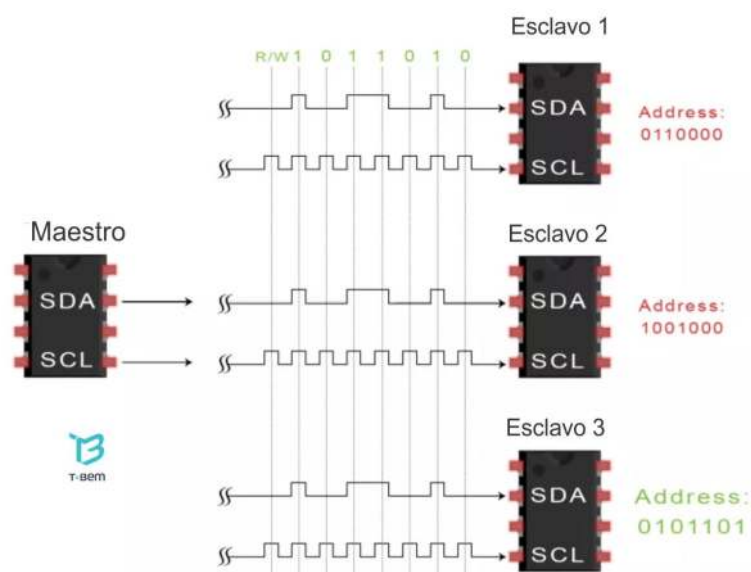


Extreta de la web "www.luisllamas.es/"

El protocol comença amb el mestre volent comunicar-se amb un esclau. Per fer-ho s'emet la condició d'inici, on l'SDA canvia el voltatge a baix i tot seguit també ho fa l'SCL. Aquest canvi indica als esclaus que el mestre vol començar una transmissió.

Per saber amb qui es vol comunicar, el mestre envia una adreça de 7 bits. A més també envia un bit R/W (*read* o *write*) indica si volen enviar (1) o rebre (0) informació. Els esclaus comparen la seva direcció amb la rebuda i si coincideix, envien un bit AFK (0) i baixen el voltatge de l'SDA. Si no coincideix, envien un bit NAFK (1) i deixen l'SDA alt.

Figura 68. Els esclaus comparen la direcció enviada amb la seva



Extreta de la web "Teslabem"

A continuació, s'envia la informació al mestre/esclau a través de l'SDA. Quan ja s'hagin enviat totes les dades, el receptor enviarà un bit AFK per notificar que les ha rebut correctament. Tot seguit es produirà la condició d'aturada pel mestre. L'SDL pujarà el voltatge i tot seguit també ho farà l'SDA.

En el meu rellotge, la WEMOS LOLIN D32 Pro és el mestre que controla als altres components, els sensors i l'OLED, que funcionaran com a esclaus. Per poder funcionar el protocol, haurem d'incloure la llibreria <Wire.h>.

Per controlar les fases, Arduino consta de les següents funcions:

- Amb la funció "`begin()`" s'inicia la llibreria i s'estableix com a mestre/esclau.
- Quan el mestre vol rebre informació d'un esclau, enviarà "`requestFrom()`". Per rebre la informació, envia un "`available()`" per comprovar si s'ha enviat un byte, i si és així, s'envia un "`write()`". Aquesta informació es pot programar perquè aparegui al Monitor Serie. Si el mestre vol enviar informació, enviarà "`read()`".
- Amb "`beginTransaction(address)`" comença la transmissió amb l'esclau amb el qual coincideix l'adreça.
- La funció "`onReceive()`" es necessita quan les dades siguin rebudes per l'esclau. En el cas que sigui el mestre qui rebi les dades, es necessita la funció "`onRequest()`".
- La funció "`SetClock()`" permet modificar la freqüència del rellotge, és a dir, la velocitat de la transmissió. Hi ha 5 valors que van des dels 10 kHz fins als 3.400 kHz. El mode estàndard (100 kHz) és el predeterminat.
- Per finalitzar la transmissió, s'envia "`endTransmission()`".

A les plaques Arduino, com per exemple a l'Arduino Uno i Nano, l'SDA correspon al pin A4 I l'SCL a l'A5. Però cada placa és diferent, pel que hauríem de consultar les especificacions. A la meua placa, en canvi, els pins s'anomenen SCL i SDA.

Per poder saber la direcció de cada esclau, podem veure les especificacions del component. Però en cas que no ho especifiqui o no puguem accedir-ne, l'"I2C Scanner" és un programa que ens informa de la direcció o direccions dels components connectats. El codi, que està disponible a "<https://playground.arduino.cc/Main/I2cScanner/>", només ens indica que hi ha algú que respon a l'adreça indicada al Monitor Serie.

Per fer una prova, només es necessita connectar un component a la placa, en el meu cas he utilitzat el sensor baromètric BMP180 perquè en les pàgines que vaig consultar no vaig trobar l'adreça, i pujar el programa. Una vegada obrim el Monitor Serie, ens apareix la direcció del BMP180 en sistema hexadecimal.

Figura 69. Al Monitor Sèrie s'observa que hi ha un perifèric connectat amb la direcció 0x77



Imatge de l'autora

9.3 ANNEX 3: exemple sensor MAX30105

```
#include <Wire.h>                // Llibreria per l'I²C
#include "MAX30105.h"
#include "heartRate.h"
MAX30105 particleSensor;
const byte RATE_SIZE = 4;        // Aquestes 7 variables es fan servir per prendre el pols
byte rates[RATE_SIZE];
byte rateSpot = 0;
long lastBeat = 0;
float beatsPerMinute;
int beatAvg;

void setup(){
  Serial.begin(115200);
  particleSensor.begin(Wire, I2C_SPEED_FAST); // Encenc el sensor de freqüència cardíaca
  particleSensor.setup();                    // Configuro el sensor amb la configuració predeterminada
  particleSensor.setPulseAmplitudeRed(0x0A); //Encenc el LED vermell per indicar que el sensor està
                                              funcionant
  particleSensor.setPulseAmplitudeGreen(0);  // Apagar el LED verd

  void loop(){
    long irValue = particleSensor.getIR();
    if (checkForBeat(irValue) == true)
    {
      long delta = millis() - lastBeat;
      lastBeat = millis();
      beatsPerMinute = 60 / (delta / 1000.0);
      if (beatsPerMinute < 255 && beatsPerMinute > 20)
      {rates[ratesSpot++] = (byte)beatsPerMinute; //Store this reading in the array
        rateSpot %= RATE_SIZE; //Wrap variable
        //Take average of readings
        beatAvg = 0;
        for (byte x = 0 ; x < RATE_SIZE ; x++)
          beatAvg += rates[x];
        beatAvg /= RATE_SIZE;}
    }
    if (irValue >= 50000){ //Si detecta un objecte a menys de 2 centímetres aprox.
      Serial.print(beatAvg);
      Serial.print(" SÓN LES TEVES PULSACIONS PER MINUT");
      Serial.println();
    }
    if (irValue < 50000){ //Si no detecta un objecte a menys de 2 centímetres aprox.
      Serial.print("NO S'HA DETECTAT EL POLS");
      Serial.println();}}
```

9.4 ANNEX 4: altímetre amb el BMP180

```
#include <Wire.h>
int periode = 15000; // Temps d'espera entre les dues mesures de l'altura
int tempsara = 0; // Una vegada han passat els 15 segons entre ambdues mesures, el comptador torna a 0
#include <SFE_BMP180.h>
SFE_BMP180 alt;
int PlantesPujades=0;
char status;
double T,P,A,AA;

void setup() {
  Serial.begin(9600);
  display.begin(SSD1306_SWITCHCAPVCC, 0x3C); }

void loop() {
  if(millis() > tempsara + periode){ // Fins que no passin 15 segons, no tornarà a calcular l'altura
    tempsara = millis();
    alt1();alt2();}}

void alt2(){
  status = alt.startTemperature(); //Demanem la temperatura
  if (status != 0){delay(status);
    status = alt.getTemperature(T); //Obtenim la temperatura
    if (status != 0){status = alt.startPressure(3); //Demanen la pressió
      if (status != 0){delay(status);
        status = alt.getPressure(P,T); //Obtenim la pressió
        if (status != 0){
          AA=P; //AA serà el punt amb el qual es compararà A}}}}

void alt1(){
  status = alt.startTemperature();
  if (status != 0){delay(status);
    status = alt.getTemperature(T);
    if (status != 0){ status = alt.startPressure(3);
      if (status != 0){delay(status);
        status = alt.getPressure(P,T);
        if (status != 0){
          A= alt.altitude(P,AA); //A serà l'altura que hem pujat o baixat des de AA
          Serial.print("S'ha pujat o baixat=");
          Serial.print(A);
          Serial.println(" m.");
          if ((A >= 2.5) && (A <=6)){ //Només es comptaran els pisos pujats, no els baixats
            PlantesPujades++;}
          Serial.print ("Plantes pujades =");
          Serial.println (PlantesPujades);}}}}}
```


9.5 ANNEX 5: exemple sensor VEML6070

```
#include "Sseed_VEML6070.h"
#ifdef ARDUINO_SAMD_VARIANT_COMPLIANCE
  #define SERIAL SerialUSB
#else
  #define SERIAL Serial
#endif
VEML6070 sensor;
char *UV_str[]={ "low level", "moderate level", "high_level", "very high", "extreme" };

err_t read_UV() {
  err_t ret=NO_ERROR;
  u16 step;
  sensor.wait_for_ready();
  CHECK_RESULT(ret,sensor.read_step(step));
  SERIAL.print("UV step = ");
  SERIAL.println(step);
  RISK_LEVEL level=sensor.convert_to_risk_level(step);

  if (UV_str[level] == "low level"){
    Serial.println("NIVELL BAIX D'UV");}

  if (UV_str[level] == "moderate level"){
    Serial.println("NIVELL MODERAT D'UV");}

  if (UV_str[level] == "high_level"){
    Serial.println("NIVELL ALT D'UV");}

  if (UV_str[level] == "very high"){
    Serial.println("NIVELL MOLT ALT D'UV");}

  if (UV_str[level] == "extreme"){
    Serial.println("NIVELL EXTREM D'UV");}
  return ret;}

void setup(){
  SERIAL.begin(115200);
  SERIAL.println("serial start!!!");
  sensor.set_interrupt(INT_145_STEP,ENABLE);}

void loop(){
  if(read_UV()) {
    SERIAL.print("read UV sensor failed!!!");
  }
}
```

9.6 ANNEX 6: exemple sensor HMC5883L

```
#include "Wire.h"
#include "I2Cdev.h"
#include "HMC5883L.h"
HMC5883L mag;
int16_t mx, my, mz;
const float declinacion = 0.55;

void setup(){
  Serial.begin(9600);
  Wire.begin();
  mag .initialize();}

void loop() {
  mag .getHeading(&mx, &my, &mz);
  float angulo = atan2(my, mx);
  angulo = angulo * RAD_TO_DEG;
  angulo = angulo - declinacion;
  if(angulo < 0) angulo = angulo + 360;

  if(angulo >= 340 && angulo < 20){
    Serial.print("N:");
    Serial.println(angulo,0);}

  if(angulo >= 20 && angulo <70){
    Serial.print("NE:");
    Serial.println(angulo,0)}

  if(angulo >= 70 && angulo <110){
    Serial.print("E:");
    Serial.println(angulo,0);}

  if(angulo >= 110 && angulo <160){
    Serial.print("SE:");
    Serial.println(angulo,0);}

  if(angulo >= 160 && angulo <200){
    Serial.print("S:");
    Serial.println(angulo,0);
  }

  if(angulo >= 200 && angulo <240){
    Serial.print("SO:");
    Serial.println(angulo,0);}
```

```
if(angulo >= 240 && angulo <300){  
  Serial.print("O:");  
  Serial.println(angulo,0);}
```

```
if(angulo >= 300 && angulo <340){  
  Serial.print("NO:");  
  Serial.println(angulo,0);}}
```

9.7 ANNEX 7: exemple sensor SI7021

```
#include "Adafruit_Si7021.h"  
Adafruit_Si7021 sensor = Adafruit_Si7021();  
  
void setup() {  
  Serial.begin(115200);  
  if (!sensor.begin()) {  
    Serial.println("Did not find Si7021 sensor!"); } }  
  
void loop() {  
  Serial.print("RH: ");  
  Serial.print(sensor.readHumidity(), 0);  
  Serial.print("\tTEMP: ");  
  Serial.println(sensor.readTemperature(), 0);  
  delay(1000);  
}
```

9.8 ANNEX 8: exemple OLED amb els botons

```

/*
BOTÓ A:
  PRÉMER number-1
  DOBLE PRESSIÓ number-2
  MANTENIR PRESSIÓ number-10
BOTÓ B:
  PRÉMER number+1
  DOBLEPRESS: number+2
  MANTENIR PRESSIÓ number+10
PRÉMER BOTÓ A & B:
  number=0
*/
#include <Wire.h>
#include <LOLIN_I2C_BUTTON.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#define OLED_RESET -1
Adafruit_SSD1306 display(OLED_RESET);
I2C_BUTTON button(DEFAULT_I2C_BUTTON_ADDRESS); //I2C Adreça 0x31
byte number = 0;

void setup(){
  Serial.begin(115200);
  display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
  display.clearDisplay();
  display.setTextSize(2);
  display.setTextColor(WHITE);
  display.println(number);
  display.println();
  display.print("- +");
  display.display();
}

void loop(){
  if (button.get() == 0) // Si es pressiona el botó
  {
    display.clearDisplay();
    display.setCursor(0, 0);
    if ((button.BUTTON_A == KEY_VALUE_HOLD) && (button.BUTTON_B == KEY_VALUE_HOLD)) //
Si es pressionen els dos a la vegada
    {
      number = 0;
    }
    else

```

```
{
  switch (button.BUTTON_B)
  {
    case KEY_VALUE_NONE:
      break;
    case KEY_VALUE_SHORT_PRESS:
      number++;
      break;
    case KEY_VALUE_DOUBLE_PRESS:
      number += 2;
      break;
    case KEY_VALUE_HOLD:
      number += 10;
      break;
    case KEY_VALUE_LONG_PRESS:
      break;
  }
  switch (button.BUTTON_A)
  {
    case KEY_VALUE_NONE:
      break;
    case KEY_VALUE_SHORT_PRESS:
      number--;
      break;
    case KEY_VALUE_DOUBLE_PRESS:
      number -= 2;
      break;
    case KEY_VALUE_HOLD:
      number -= 10;
      break;
    case KEY_VALUE_LONG_PRESS:
      break;}}

display.println(number);
display.println();
display.print("- +");
display.display();}}
```

9.9 ANNEX 9: exemple OLED amb imatges

```
#include "Adafruit_Si7021.h"
Adafruit_Si7021 sensor = Adafruit_Si7021();
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#define OLED_RESET 0
Adafruit_SSD1306 display(OLED_RESET);

const unsigned char temp [] PROGMEM = { //24x48px
0x01, 0xe0, 0x00, 0x03, 0xb8, 0x00, 0x06, 0x0c, 0x00, 0x0c, 0x04, 0x00, 0x08, 0x02, 0x00, 0x08, 0x02, 0x00,
0x08, 0x02, 0x00, 0x08, 0x02, 0x00, 0x08, 0x02, 0x00, 0x08, 0x02, 0x00, 0x08, 0x02, 0x7f, 0x08, 0x02, 0x7f,
0x08, 0x02, 0x00, 0x08, 0x02, 0x00, 0x08, 0x02, 0x70, 0x08, 0x42, 0x00, 0x08, 0xe2, 0x00, 0x09, 0xe2, 0x00,
0x09, 0xe2, 0x7f, 0x09, 0xe2, 0x00, 0x09, 0xe2, 0x00, 0x09, 0xe2, 0x70, 0x09, 0xe2, 0x00, 0x09, 0xe2, 0x00,
0x09, 0xe2, 0x00, 0x09, 0xe2, 0x7f, 0x09, 0xe2, 0x00, 0x09, 0xe2, 0x00, 0x09, 0xe2, 0x00, 0x09, 0xe2, 0x00,
0x11, 0xe3, 0x00, 0x31, 0xe1, 0x80, 0x21, 0xe0, 0x80, 0x43, 0xf0, 0xc0, 0x47, 0xf8, 0x40, 0x47, 0xfc, 0x40,
0xc7, 0xfc, 0x40, 0x87, 0xfc, 0x40, 0xc7, 0xfc, 0x40, 0x47, 0xfc, 0x40, 0x47, 0xf8, 0x40, 0x43, 0xf8, 0xc0,
0x61, 0xe0, 0x80, 0x20, 0x01, 0x80, 0x10, 0x03, 0x00, 0x0c, 0x06, 0x00, 0x07, 0xbc, 0x00, 0x01, 0xf0, 0x00
};

const unsigned char hum [] PROGMEM = { //28x48px
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x06, 0x00, 0x00, 0x00, 0x0c, 0x00, 0x00, 0x18, 0x00, 0x00, 0x00, 0x10,
0x00, 0x00, 0x00, 0x20, 0x40, 0x00, 0x00, 0x60, 0x60, 0x00, 0x00, 0xc0, 0x20, 0x00, 0x00, 0x80, 0x30, 0x00,
0x01, 0x80, 0x18, 0x00, 0x01, 0x00, 0x08, 0x00, 0x03, 0x00, 0x0c, 0x00, 0x02, 0x00, 0x04, 0x00, 0x06, 0x20,
0x46, 0x00, 0x04, 0x70, 0xc2, 0x00, 0x04, 0x59, 0x82, 0x00, 0x04, 0x73, 0x02, 0x00, 0x0c, 0x26, 0x03, 0x00,
0x0c, 0x0c, 0x03, 0x00, 0x0c, 0x19, 0xc3, 0x00, 0x0c, 0x33, 0x43, 0x00, 0x04, 0x63, 0x42, 0x00, 0x04, 0x41,
0x82, 0x00, 0x06, 0x00, 0x06, 0x00, 0x02, 0x00, 0x04, 0x00, 0x03, 0x00, 0x0c, 0x00, 0x01, 0x80, 0x18, 0x00,
0x00, 0xf0, 0xf0, 0x00, 0x00, 0x1f, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
};

void setup() {
  Serial.begin(115200);
  display.begin(SSD1306_SWITCHCAPVCC, 0x3C);}

void loop() {
  display.clearDisplay();
  display.drawBitmap(0,0,hum,28,48,1);
  display.setTextSize(1);
  display.setTextColor(WHITE);
  display.setCursor(30,10);
  display.println("HUM:");
```

```
display.setCursor(30,20);
display.println(sensor.readHumidity(), 0);
display.setCursor(45,20);
display.println("%");
display.display();

Serial.print("Humidity: ");
Serial.print(sensor.readHumidity(), 0);
delay(5000);

display.clearDisplay();
display.drawBitmap(0,0,temp,24,48,1);
display.setTextSize(1);
display.setTextColor(WHITE);
display.setCursor(30,10);
display.println("TEMP:");
display.setCursor(30,20);
display.println(sensor.readTemperature(), 0);
display.drawCircle(45, 21, 1, WHITE);
display.setCursor(50,20);
display.println("C");
display.display();

Serial.print("Temp: ");
Serial.print(sensor.readTemperature(), 0);
delay(5000);
}
```

9.10 ANNEX 10: exemple brunzidor

```
const int pinbuzzer = 0;

void setup() {
  pinMode(pinbuzzer, OUTPUT);} //el pin del buzzer és una sortida

void loop(){
  digitalWrite(pinbuzzer, HIGH); //encenem
  delay(100);
  digitalWrite(pinbuzzer, LOW); //apaguem
  delay(100);}
```

9.11 ANNEX 11: exemple sensor PAJ7620

```
#include <Wire.h>
#include "paj7620.h"
#define GES_REACTION_TIME      500
#define GES_ENTRY_TIME        800
#define GES_QUIT_TIME          1000

void setup(){
    uint8_t error = 0;
    Serial.begin(9600);
    error = paj7620Init();

void loop(){
    uint8_t data = 0, data1 = 0, error;

    error = paj7620ReadReg(0x43, 1, &data);
    if (!error) {
        switch (data){
            case GES_RIGHT_FLAG:
                delay(GES_ENTRY_TIME);
                paj7620ReadReg(0x43, 1, &data);
                if(data == GES_FORWARD_FLAG){
                    Serial.println("Forward");
                    delay(GES_QUIT_TIME);}
                else if(data == GES_BACKWARD_FLAG){
                    Serial.println("Backward");
                    delay(GES_QUIT_TIME);}
                else{
                    Serial.println("Right");}
                break;
            case GES_LEFT_FLAG:
                delay(GES_ENTRY_TIME);
                paj7620ReadReg(0x43, 1, &data);
                if(data == GES_FORWARD_FLAG){
                    Serial.println("Forward");
                    delay(GES_QUIT_TIME);}
                else if(data == GES_BACKWARD_FLAG){
                    Serial.println("Backward");
                    delay(GES_QUIT_TIME);}
                else{
                    Serial.println("Left");}
                break;
            case GES_UP_FLAG:
                delay(GES_ENTRY_TIME);
```



```
paj7620ReadReg(0x43, 1, &data);
if(data == GES_FORWARD_FLAG){
    Serial.println("Forward");
    delay(GES_QUIT_TIME);}
else if(data == GES_BACKWARD_FLAG){
    Serial.println("Backward");
    delay(GES_QUIT_TIME);}
else{
    Serial.println("Up");}
break;
case GES_DOWN_FLAG:
    delay(GES_ENTRY_TIME);
    paj7620ReadReg(0x43, 1, &data);
    if(data == GES_FORWARD_FLAG){
        Serial.println("Forward");
        delay(GES_QUIT_TIME);}
    else if(data == GES_BACKWARD_FLAG){
        Serial.println("Backward");
        delay(GES_QUIT_TIME);}
    else{
        Serial.println("Down");}
    break;
case GES_FORWARD_FLAG:
    Serial.println("Forward");
    delay(GES_QUIT_TIME);
    break;
case GES_BACKWARD_FLAG:
    Serial.println("Backward");
    delay(GES_QUIT_TIME);
    break;
case GES_CLOCKWISE_FLAG:
    Serial.println("Clockwise");
    break;
case GES_COUNT_CLOCKWISE_FLAG:
    Serial.println("anti-clockwise");
    break;
default:
    paj7620ReadReg(0x44, 1, &data1);
    if (data1 == GES_WAVE_FLAG) {
        Serial.println("wave");}
    break;}}
delay(100);}
```