

MEMÒRIA



SENSORS BÀSICS EN UN HORT URBÀ

Nom:	Xavier Martinez Escolar
Data:	03.12.2024

Taula de continguts

1. **Introducció**
 - 1.1. Objectiu del projecte
2. **Materials i Components**
 - 2.1. Raspberry Pi (model utilitzat)
 - 2.2. Sensors i perifèrics
 - 2.2.1. Sensor d'Aigua
 - 2.2.2. Pantalla LCD 16x2 (i2C)
 - 2.2.3. Sensor de llum BH1750
 - 2.2.4. Sensor de temperatura DHT11
 - 2.2.5. Sensor d'humitat de terra YI-69
 - 2.2.6. Pi Camera
 - 2.2.7. Protoboard i cables
 - 2.3. Fonts d'alimentació i connectivitat
 - 2.3.1. Font d'alimentació de 5V
 - 2.3.2. Cable USB per a connexió a la xarxa
3. **Desenvolupament del Sistema**
 - 3.1. Configuració de la Raspberry Pi
 - 3.1.1. Instal·lació del sistema operatiu
 - 3.1.2. Configuració de la xarxa i accés remot (SSH)
 - 3.2. Connexió dels sensors a Raspberry Pi
 - 3.2.1. Connexió del sensor d'aigua
 - 3.2.2. Connexió de Pantalla LCD 16x2 (i2C)
 - 3.2.3. Connexió del sensor de llum BH1750
 - 3.2.4. Connexió del sensor de temperatura DHT11
 - 3.2.5. Connexió del sensor d'humitat de terra YI-69
 - 3.2.6. Connexió Pi Camera
 - 3.3. Configuració de la Protoboard
 - 3.4. Connexió de l'alimentació
4. **Programació del Sistema**
 - 4.1. Instal·lació de biblioteques i eines necessàries
 - 4.1.1. Instal·lació de Python i GPIO
 - 4.1.2. Instal·lació de llibreries per a LCD i2C
 - 4.2. Esquema bàsic de programació
 - 4.2.1. Control del sensor d'aigua
 - 4.2.2. Control de Pantalla LCD 16x2 (i2C)
 - 4.2.3. Control del sensor de llum BH1750
 - 4.2.4. Control del sensor de temperatura DHT11
 - 4.2.5. Control del sensor d'humitat de terra YI-69
 - 4.2.6. Control de Pi Camera
5. **Conclusió**
 - 5.1. Resum dels resultats obtinguts
 - 5.2. Avaluació del sistema
 - 5.3. Limitacions del projecte
 - 5.4. Millors futures
6. **Annexos**
 - 6.1. Codi font complet

- 6.2. Diagrama de connexions
- 6.3. Enllaços a documentació i recursos

7. **Referències**

1. Introducció

1.1. Objectiu del projecte

El Mercat Municipal de la Vall d'Hebron-Taxonera disposa d'un hort urbà a la planta primera. Aquest hort urbà es un projecte d'agricultura urbana singular, amb una superfície de 1.750 m², on conrear espècies vegetals de secà (debat al període de sequera), i experimentar nous cultius adaptables al clima mediterrani.

L'hort urbà en transformació:

<https://youtu.be/LroyCEZPqQw>



Els responsables de la gestió de l'hort urbà volen presentar un projecte pilot d'implantació de tecnologia raspberry amb sensors d'aigua, d'humitat, temperatura ambient i d'altres.

La proliferació d'horts urbans a l'àrea metropolitana de Barcelona necessita solucions tecnològiques per fer front als canvis climatològics que pateix el planeta els darrers anys o decades amb episodis de sequera extrema, temperatures elevades o altes concentracions de CO₂. La gestió de l'aigua és una prioritat per aquestes cultius en entorn urbà.

La Internet de les coses i la implantació d'estructures físiques tan populars com Raspberry (o Arduino) ens obre una infinitat de possibilitats per digitalitzar tot el sistema de detecció i control de les mesures meteorològiques en temps real de l'hort urbà. Tot això, ens permet monitoritzar i actuar ràpidament davant afectacions al cultiu de plantes.

2. Materials i Components

2.1. Raspberry Pi (3B+)

En aquest projecte s'ha optat per utilitzar targetes Raspberry PI 3. La Raspberry es pot utilitzar directament com a ordinador domèstic. Disposen de connectivitat WiFi i Bluetooth.

La Raspberry Pi 3 Model B+ és un microordinador compacte i potent, dissenyat per a aplicacions d'aprenentatge, desenvolupament i projectes IoT. Equipada amb un procesador de quatre nuclis Broadcom BCM2837B0 a 1.4 GHz i 1 GB de RAM, ofereix un rendiment millorat respecte a models anteriors. Inclou connectivitat sense fils integrada amb Wi-Fi de doble banda (2.4 GHz i 5 GHz) i Bluetooth 4.2, a més de port Ethernet Gigabit amb velocitat limitada a 300 Mbps.

Disposa de 4 ports USB 2.0, un port HDMI, àudio analògic i interfícies GPIO per connectar perifèrics. És compatible amb sistemes operatius com Raspberry Pi OS i altres basats en Linux. Perfecta per domòtica, servidors lleugers, educació i prototips electrònics avançats. Funciona amb una font d'alimentació de 5V i 2.5A a través de micro-USB.

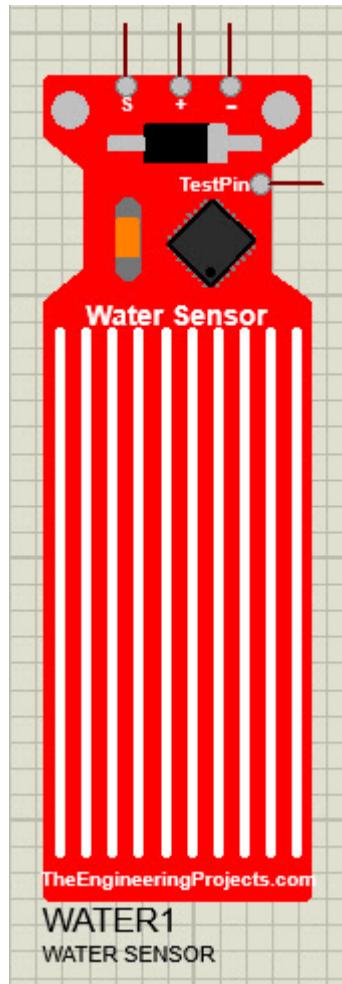


En etapes posteriors del projecte d'ús de Raspberry de l'Hort Urbà del Mercat de la Vall d'Hebron-Taxonera es poden implementar funcionalitats de seguretat i videovigilància, per detectar petits rosegadors.

Els rosegadors poden menjar les arrels, les fulles, els fruits i els brots joves de les plantes. Això pot perjudicar el creixement dels cultius, reduint la seva producció o fins i tot eliminant-les completament en casos més greus. També poden ser vectors de malalties que afecten tant les plantes com els animals i humans.

2.2. Sensors i perifèrics

2.2.1. Sensor d'Aigua (Water Sensor)



Un **water sensor** o sensor d'aigua és un dispositiu que detecta la presència, absència o nivell d'aigua en una ubicació específica. Es fa servir en nombroses aplicacions, com ara la gestió d'aigua en sistemes de reg, la detecció de fuites, el control de tancs d'aigua i dispositius domèstics.

Els sensors d'aigua habitualment funcionen utilitzant una sèrie de contactes conductors que generen un senyal elèctric quan estan en contacte amb aigua. Alguns models més avançats poden mesurar la conductivitat o fins i tot la puresa de l'aigua.

Tipus de Water Sensors

1. **Resistius:** Detecten canvis en la resistència elèctrica entre els seus contactes.
2. **Capacitius:** Funcionen mesurant canvis en la capacitat elèctrica deguts a la presència d'aigua.
3. **Ultrasònics:** Mesuren el nivell d'aigua basant-se en el temps de reflexió d'un senyal d'ultrasons.

Aplicacions

- Gestió d'horts i reg automàtic.
- Prevenció de danys per fuites d'aigua en entorns domèstics.
- Monitors de dipòsits d'aigua per a ús industrial i agrícola.

2.2.2. Pantalla LCD 16x2 (I2C)

	<p>La pantalla LCD 16x2 amb interfície I2C és un mòdul que permet mostrar fins a 32 caràcters en dues línies de 16 columnes. Gràcies al seu adaptador I2C integrat, només requereix 4 connexions (VCC, GND, SDA, SCL), simplificant el cablejat i estalvant pins del microcontrolador. Ideal per mostrar informació en projectes d'electrònica amb espai limitat.</p>
---	--

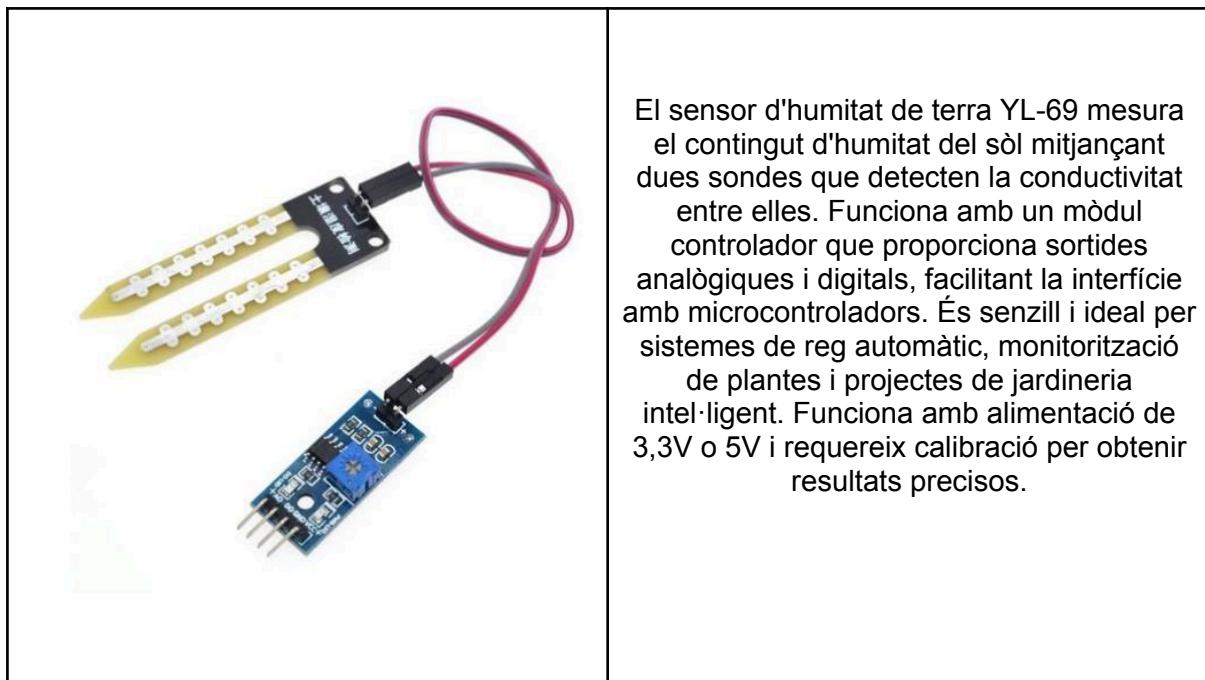
2.2.3. Sensor de llum BH1750

	<p>El sensor BH1750 és un sensor digital de llum ambiental que mesura la intensitat de la llum en lux, utilitzant interfície I2C per a la comunicació amb microcontroladors. Ofereix alta precisió i un rang de mesura ampli (1 a 65.535 lux), ideal per aplicacions d'automatització, fotografia i il·luminació intel·ligent. És compacte, de baix consum energètic i no requereix calibració externa. Funciona amb tensions de 3,3V o 5V, sent fàcil d'integrar en projectes.</p>
--	--

2.2.4. Sensor de temperatura DHT11

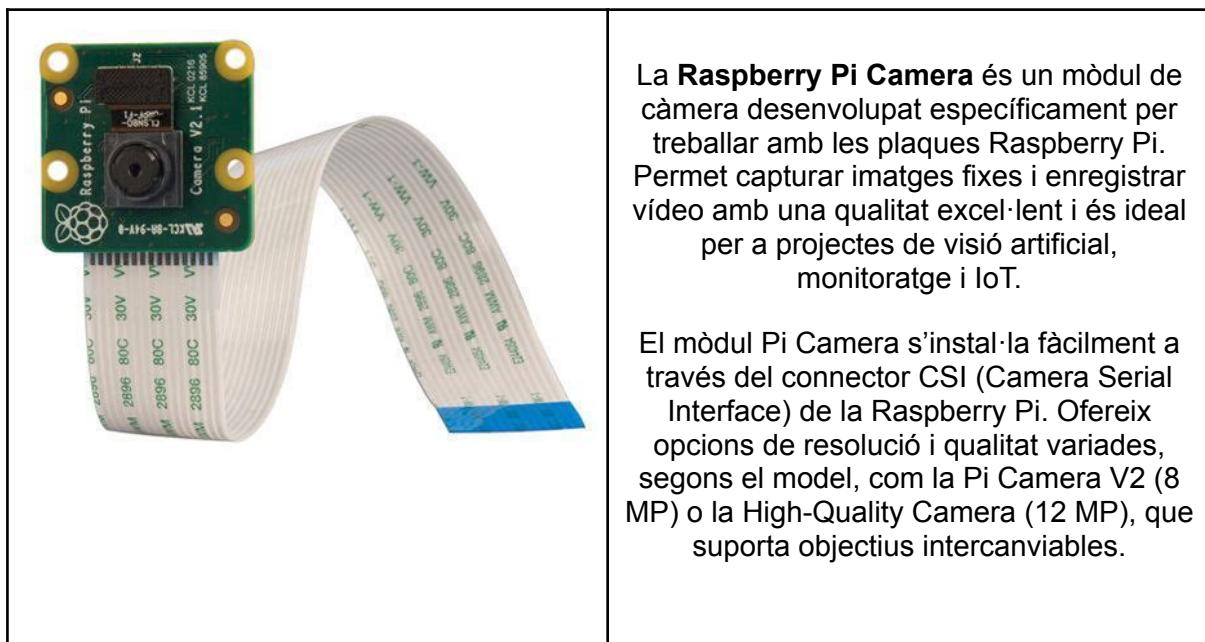
	<p>El sensor DHT11 és un sensor digital de temperatura i humitat d'alta fiabilitat i fàcil ús. Pot mesurar temperatures entre 0°C i 50°C amb una precisió de $\pm 2^\circ\text{C}$, i humitat relativa entre el 20% i el 90% amb precisió de $\pm 5\%$. Proporciona lectures digitals mitjançant un únic pin de dades, simplificant la interfície amb microcontroladores. És econòmic i ideal per aplicacions domòtiques, meteorològiques i projectes educatius.</p>
---	--

2.2.5. Sensor d'humitat de terra YL-69



El sensor d'humitat de terra YL-69 mesura el contingut d'humitat del sòl mitjançant dues sondes que detecten la conductivitat entre elles. Funciona amb un mòdul controlador que proporciona sortides analògiques i digitals, facilitant la interfície amb microcontroladors. És senzill i ideal per sistemes de reg automàtic, monitorització de plantes i projectes de jardineria intel·ligent. Funciona amb alimentació de 3,3V o 5V i requereix calibració per obtenir resultats precisos.

2.2.6. Pi Camera

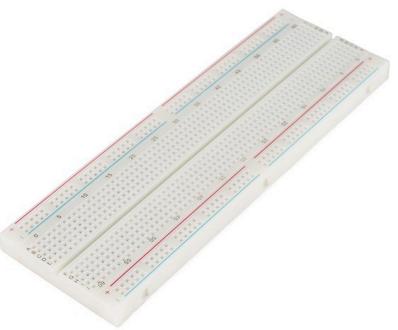
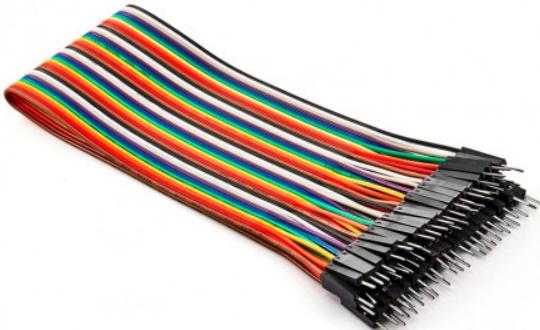


La **Raspberry Pi Camera** és un mòdul de càmera desenvolupat específicament per treballar amb les plaques Raspberry Pi. Permet capturar imatges fixes i enregistrar vídeo amb una qualitat excel·lent i és ideal per a projectes de visió artificial, monitoratge i IoT.

El mòdul Pi Camera s'instal·la fàcilment a través del connector CSI (Camera Serial Interface) de la Raspberry Pi. Ofereix opcions de resolució i qualitat variades, segons el model, com la Pi Camera V2 (8 MP) o la High-Quality Camera (12 MP), que suporta objectius intercanviables.

2.2.7. Protoboard i cables

Escollim un model de placa Protoboard d'alta qualitat de 16x5 cm i 830 punts.

	<p>Entre els molts avantatges de les plaques Protoboard destaquem la facilitat per canviar connexions amb línies de distribució d'alimentació i terra que faciliten la creació de circuits nets i ordenats. Les Protoboards són àmpliament utilitzades en molts entorns. I finalment, es tracta d'unes plaques relativament econòmiques.</p>
	<p>Malla plana de 40 cables individuals de diferents colors de tipus Dupont (20 cm). Són flexibles i separables, permetent l'ús individual o en grups. Es fan servir habitualment amb plaques de prototipatge, sensors i microcontroladors com Raspberry Pi.</p>

2.3. Fonts d'alimentació i connectivitat

2.3.1. Font d'alimentació de 5V



La **font d'alimentació de 5V** és un dispositiu dissenyat per subministrar energia constant de 5 volts a components electrònics com microcontroladors, sensors o pantalles. S'utilitza habitualment en projectes basats en Arduino, Raspberry Pi i altres dispositius de baixa potència. Aquest tipus de font pot tenir diferents connectors d'entrada, com USB o connectors de tipus jack. Les fonts de 5V són essencials per garantir una alimentació estable i segura en circuits amb requisits específics de tensió.

2.3.2. Cable USB per a connexió a la xarxa



El **cable USB per a connexió a la xarxa** permet establir una connexió Ethernet a dispositius com ordinadors, Raspberry Pi o altres aparells mitjançant un port USB. Aquest cable és útil per dispositius que no tenen un port Ethernet dedicat o quan es vol afegir una connexió de xarxa a dispositius sense la funcionalitat integrada. S'utilitza amb adaptadors USB a Ethernet, que proporcionen una connexió ràpida i fiable per a l'accés a internet o xarxes locals. La seva instal·lació és plug-and-play en la majoria de sistemes operatius.

3. Desenvolupament del Sistema

3.1. Configuració de la Raspberry Pi

La configuració de la Raspberry Pi és un procés essencial per posar en marxa aquesta petita computadora i començar a utilitzar-la per a diferents projectes. Seguint els passos bàsics, es pot configurar de manera ràpida i eficient:

1. **Preparació de la targeta microSD:** Abans de començar, necessites una targeta microSD (generalment de 8 GB o més) en la qual instal·lar el sistema operatiu. El sistema més utilitzat és Raspberry Pi OS, que es pot descarregar des del lloc web oficial i instal·lar-se mitjançant eines com Raspberry Pi Imager.
2. **Connexió de la Raspberry Pi:** Un cop el sistema operatiu estigui carregat a la targeta microSD, la pots inserir a la Raspberry Pi. Connecta un monitor mitjançant un cable HDMI, una font d'alimentació mitjançant un adaptador micro-USB (o USB-C, depenent del model), un teclat i un mouse mitjançant els ports USB.
3. **Arrencada i configuració inicial:** Quan posis en marxa la Raspberry Pi, es mostrarà una pantalla de configuració inicial, que inclou la selecció de l'idioma, la configuració de la xarxa Wi-Fi (si esculls una connexió sense fils), la zona horària i la configuració de contrasenya. També és possible actualitzar el sistema durant aquesta fase.
4. **Actualitzacions i seguretat:** Després de la configuració inicial, és important actualitzar el sistema per assegurar-te que tens les últimes versions dels paquets i les actualitzacions de seguretat. Això es pot fer fàcilment mitjançant la línia de comandes o utilitzant l'aplicació Raspberry Pi Configuration.
5. **Ús i personalització:** Un cop la Raspberry Pi estigui configurada, es pot començar a utilitzar per a projectes de programació, control d'electrònica, o fins i tot convertir-la en un centre multimèdia o un servidor web. També pots personalitzar la interfeïcie i les funcionalitats segons les teves necessitats.

Aquest procés bàsic et permetrà començar a utilitzar la Raspberry Pi de manera efectiva i aprofitar tot el seu potencial per a una gran varietat de projectes.

3.1.1. **Instal·lació del sistema operatiu**

La instal·lació del sistema operatiu en una Raspberry Pi és un dels primers passos per posar en marxa la petita computadora i començar a utilitzar-la en els teus projectes. A continuació, et descriu els passos bàsics per fer-ho:

Descarregar el sistema operatiu: el sistema operatiu més comú per a la Raspberry Pi és Raspberry Pi OS (abans conegut com Raspbian), que és una versió lleugera de Linux basada en Debian. El sistema operatiu es pot descarregar de manera gratuita des de la pàgina oficial de Raspberry Pi.

Preparar la targeta microSD: la Raspberry Pi necessita una targeta microSD (mínim 8 GB de capacitat) per emmagatzemar el sistema operatiu i els fitxers. Necessitaràs una eina com Raspberry Pi Imager o Etcher per escriure el sistema operatiu descarregat a la targeta microSD. Amb Raspberry Pi Imager, simplement selecciona el sistema operatiu desitjat, tria la targeta microSD com a destinació i fes clic a "Escriure".

Inserir la targeta microSD a la Raspberry Pi: Un cop la targeta microSD estigui preparada amb el sistema operatiu, insereix-la a la Raspberry Pi. A continuació, connecta la Raspberry Pi a un monitor, un teclat i un ratolí, i finalment connecta el cable d'alimentació.

Engregar la Raspberry Pi: quan connectis l'alimentació, la Raspberry Pi començarà a arrencar. En la primera arrencada, es mostrarà una pantalla de configuració inicial que et permet configurar la teva connexió Wi-Fi, l'idioma, la zona horària i altres opcions bàsiques.

Actualitzar el sistema operatiu: Un cop completada la configuració inicial, és recomanable actualitzar el sistema operatiu i els paquets per garantir que tens les versions més recents i segures. Això es pot fer fàcilment mitjançant la línia de comandes amb l'ordre `sudo apt update && sudo apt upgrade`.

Un cop realitzats aquests passos, el sistema operatiu estarà instal·lat i la Raspberry Pi 3 estarà llesta per començar a fer servir en els teus projectes.

3.1.2. Configuració de la xarxa i accés remot (SSH)

La configuració de la xarxa i l'accés remot (SSH) a la Raspberry Pi permet utilitzar-la de manera més flexible i còmoda, ja que no es necessita conectar un monitor, keyboard o mouse de manera permanent. Aquí tens els passos per configurar la xarxa i activar l'accés remot a través de SSH:

Configuració de la xarxa

1. Connexió Wi-Fi (sense fils):

- Si vols conectar la Raspberry Pi a una xarxa Wi-Fi, durant el procés de configuració inicial (després de la instal·lació del sistema operatiu), se't demana que seleccionis una xarxa Wi-Fi disponible. Simplement selecciona la xarxa desitjada, introduceix la contrasenya i confirma la connexió.
- Si necessites canviar la configuració de la xarxa Wi-Fi més tard, pots accedir a la configuració de xarxa des de la interfície gràfica de Raspberry Pi OS, a través del menú "Wi-Fi" a la barra de tasques o bé mitjançant la línia de comandes amb l'ordre `sudo raspi-config` i seleccionant les opcions de xarxa.

2. Connexió per cable Ethernet:

- Si prefereixes una connexió més estable i ràpida, pots connectar la Raspberry Pi a la teva xarxa mitjançant un cable Ethernet. Un cop conectada al cable, la Raspberry Pi obtindrà automàticament una adreça IP mitjançant el protocol DHCP (si el router està configurat per a això). No caldrà configurar res de manera manual en aquest cas.

Activació de l'accés remot (SSH)

1. Activar SSH: L'accés remot a través de SSH permet controlar la Raspberry Pi des d'una altra màquina mitjançant la línia de comandes, sense necessitat de connectar-se a la pantalla, teclat i mouse. Per activar SSH:

- Si estàs utilitzant Raspberry Pi OS des de la configuració inicial, hi haurà una opció per activar SSH. Només cal seleccionar l'opció "Enable SSH" i guardar els canvis.
- Si ja tens la Raspberry Pi configurada, pots activar SSH a través de la línia de comandes. Obre una terminal i escriu `sudo systemctl enable ssh` seguit de `sudo systemctl start ssh`.

2. Accedir via SSH: Un cop activat SSH, pots connectar-te de manera remota a la **Raspberry Pi** des d'un altre ordinador (com a Linux, macOS o Windows amb PuTTY). Necessites conèixer l'adreça IP de la **Raspberry Pi**, que pots obtenir utilitzant l'ordre `hostname -I` a la Raspberry Pi o des del router. Des de la línia de comandes d'un ordinador amb Linux: `ssh pi@<adreça_IP>`.

Substitueix <adreça_IP> per la IP real de la Raspberry Pi.

En el cas de Windows, pots utilitzar una aplicació com **PuTTY** per establir una connexió SSH.

3. **Seguretat SSH:** Per millorar la seguretat, és recomanable canviar la contrasenya per defecte de l'usuari "pi" utilitzant l'ordre **passwd**, i desactivar l'inici de sessió amb contrasenya per poder utilitzar claus SSH. Això farà que l'accés a la Raspberry Pi sigui més segur i evitar atacs de força bruta.

Amb aquests passos, tindràs la Raspberry Pi configurada per accedir-hi remotament mitjançant SSH i amb connexió a la xarxa, cosa que facilita el seu ús en entorns de servidor o per a projectes a distància.

3.2. Connexió dels sensors a Raspberry Pi

3.2.1. Connexió del sensor d'Aigua

Els sensors d'aigua són dispositius útils per detectar presència o nivells d'aigua. Aquests sensors s'utilitzen en aplicacions com sistemes de reg automàtic, detecció de fuites o monitorització de dipòsits d'aigua. A continuació, veurem com connectar un sensor d'aigua a una Raspberry Pi i recollir dades mitjançant Python.

Materials necessaris:

1. **Raspberry Pi** (model 3B, 4 o superior és recomanable).
2. **Sensor d'aigua** (per exemple, un sensor de nivell de resistència o capacitat).
3. **Cables jumper** (mascle-femella o mascle-mascle segons necessitis).
4. **Resistència** (opcional, depèn del sensor específic emprat).
5. **Protoboard** (opcional, però molt útil per fer connexions temporals).
6. **Font d'alimentació** per a la Raspberry Pi.
7. **Software**: Raspberry Pi OS i una biblioteca de Python com *RPi.GPIO*.

Funcionament del Sensor:

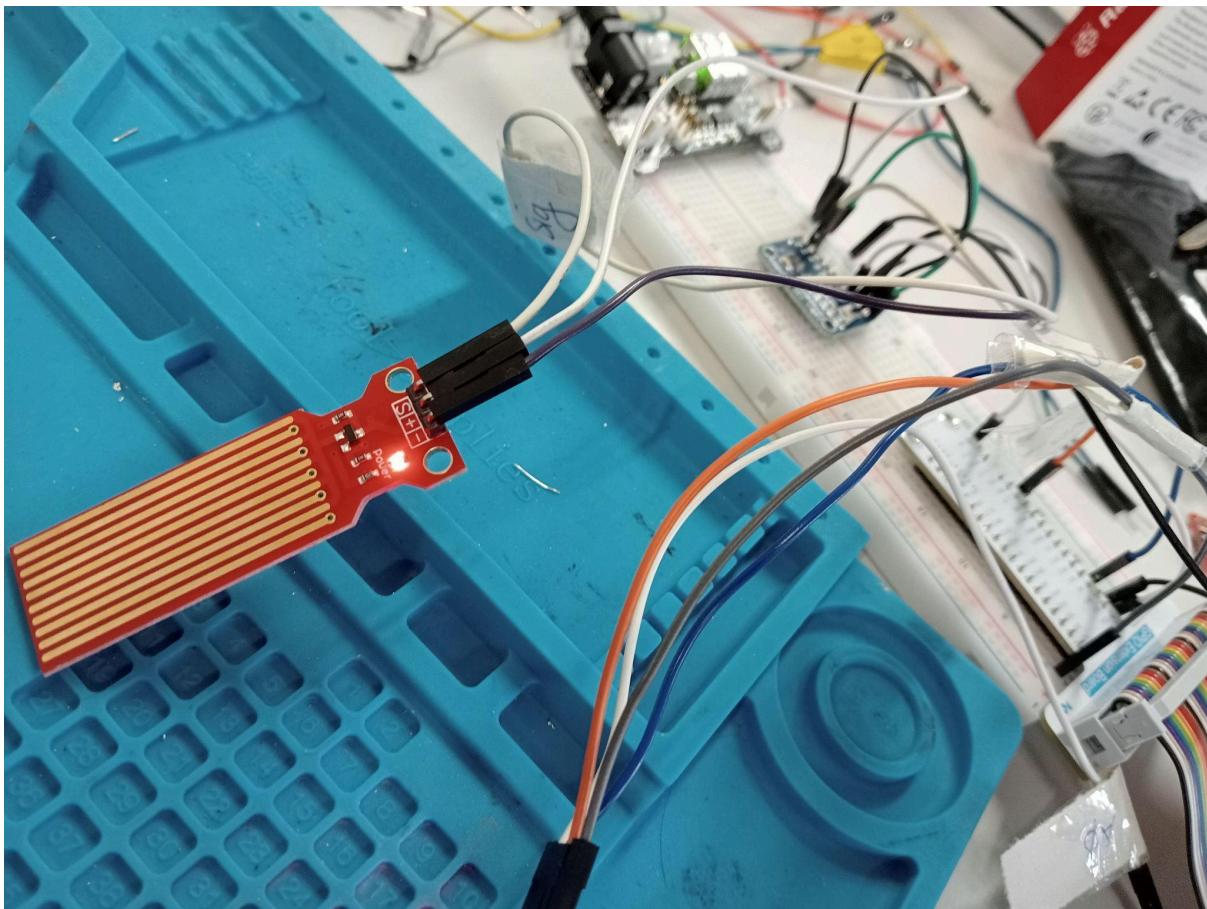
El sensor d'aigua detecta la presència o absència d'aigua mitjançant una placa que té pistes conductores. Quan l'aigua està present, aquestes pistes es connecten i permeten el pas del corrent elèctric, generant un senyal que pot ser llegit per la Raspberry Pi.

- **Sensors digitals:** Retornen un valor "HIGH" (1) o "LOW" (0) basat en la presència d'aigua.
- **Sensors analògics:** Proporcionen una lectura en un rang de valors que depèn de la quantitat d'aigua (valor mínim = 0, valor màxim a especificar).

Connexions:

1. Identifica els pins del sensor d'aigua:
 - **VCC (+)**: Alimentació positiva. Subministra energia al sensor. Es recomana utilitzar una tensió entre 3.3V i 5V.
 - **GND (-)**: És una connexió a Terra (ground).
 - **OUT (S)**: Sortida de dades (pot ser digital o analògica). S'ha de connectar a una de les entrades analògiques de Raspberry Pi.
2. Connecta el sensor a la Raspberry Pi:
 - **VCC (+)** → Pin de 3.3V o 5V de la Raspberry Pi (depèn del sensor).
 - **GND (-)** → Pin de terra (GND) de la Raspberry Pi.

- OUT (S) → Pin GPIO (entrada digital o analògica).
3. Nota: Si el sensor és analògic, caldrà un convertidor ADC (convertidor analògic-digital), com l'ADS1115, perquè la Raspberry Pi només suporta entrades digitals.



Instal.lació:

Opció 1: Utilitzar un entorn virtual

Es recomana per aïllar les llibreries de Python:

Obrir entorn bash

1. Crear un entorn virtual:

```
python3 -m venv ~/myenv
```

2. Activar l'entorn virtual:

a. Linux/macOS:

```
source ~/myenv/bin/activate
```

b. Windows:

- ```
~/myenv/Scripts/activate
```
3. Instal.lar la llibreria en un entorn virtual:

```
pip install adafruit-ads1x15
```

  4. Per sortir de l'entorn virtual:

```
deactivate
```

#### Opció 2: Instal·lació via apt

S'ha de comprovar si la llibreria es troba disponible al repositori. Per això, tenim aquestes dues comandes::

```
sudo apt update
sudo apt install python3-adafruit-ads1x15
```

#### Opció 3: Utilitzar --break-system-packages (No Recomanada)

Tot i que aquesta opció no es recomana, si les altres dues opcions donen errors a l'instal.lació, es pot executar aquesta línia de comanda :

```
sudo pip3 install adafruit-ads1x15 --break-system-packages
```

## i2C

### 1. Comprova si l'i2C està activat

Primer, has de comprovar si l'I2C està habilitat al teu sistema. Si estàs treballant amb Raspberry Pi o una altra placa similar, segueix aquests passos:

- a. Habilitar I2C en Raspberry Pi. Ho farem amb la comanda:

```
sudo raspi-config
```

- b. Navega a Interfacing Options > I2C i selecciona Yes per habilitar l'I2C.
- c. Després de reiniciar la Raspberry Pi, comprova si el bus I2C està disponible amb aquesta comanda:

```
i2cdetect -y 1
```

### 2. Instal·la les biblioteques I2C necessàries

Si tens un sistema com Raspberry Pi i has activat I2C, però l'error persisteix, assegura't que tens instal·lades les biblioteques de suport d'I2C:

```
sudo apt-get update
sudo apt-get install python3-smbus i2c-tools
```

Aquestes biblioteques són essencials perquè Adafruit ADS1x15 pugui comunicar-se amb el dispositiu I2C.

### Com funciona el sensor de nivell d'aigua

En resum, la tensió de sortida al pin de senyal augmenta a mesura que augmenta la quantitat d'aigua en què el sensor està submergit.

El sensor té deu pistes de coure exposades, cinc de les quals són pistes d'alimentació i les altres cinc són pistes de detecció. Aquestes pistes estan disposades en paral·lel, amb una pista de detecció entre cada dues pistes d'alimentació. A menys que l'aigua les uneixi quan estan submergides, aquestes pistes romanen desconnectades.

Les pistes actuen com una resistència variable, similar a un potenciómetre, i la seva resistència depèn del nivell d'aigua:

- La resistència es determina per la distància entre la part superior del sensor i la superfície de l'aigua.
- La resistència és inversament proporcional a la quantitat d'aigua present:
  - Com més aigua submergeix el sensor, més augmenta la conductivitat i menys resistència hi ha.
  - Com menys aigua submergeix el sensor, més disminueix la conductivitat i més resistència hi ha.

La tensió de sortida del sensor es basa en la resistència. Es pot determinar el nivell d'aigua mesurant la tensió.

### Calibració del sensor de nivell d'aigua

La sortida del sensor no només es veu afectada pel nivell d'aigua, sinó també per la conductivitat de l'aigua. L'aigua pura no és conductiva, mentre que l'aigua amb minerals i impureses sí que ho és. Com més alta és la conductivitat de l'aigua, més sensible és el sensor. A més, el valor de sortida també depèn de la tensió subministrada al pin VCC del sensor. Per garantir l'exactitud en la lectura del sensor d'aigua, recomanem calibrar el sensor per al tipus específic d'aigua que s'ha de monitoritzar.

### Instruccions per a la calibració

- Utilitza l'esbós proporcionat anteriorment per llegir el valor del sensor.
- Submergeix el sensor a l'aigua al nivell desitjat per establir el llindar.
- Anota el valor que mostra el sensor al terminal.
- Fes servir aquest valor com a llindar per activar una acció.

La prova també pot servir per determinar:

El valor 'MIN\_ADC\_VALUE', quan el sensor no està submergit en el líquid.

El valor 'MAX\_ADC\_VALUE', quan el sensor està completament submergit en el líquid.

### 3.2.2. Connexió de la pantalla LCD 16x2 (i2C)

La connexió d'una pantalla LCD 16x2 amb interfície i2C a una Raspberry Pi és una tasca senzilla i pràctica per afegir una interfície visual als teus projectes. La pantalla LCD 16x2 és capaç de mostrar 16 caràcters per 2 línies, i l'ús de l'i2C permet reduir el nombre de cables necessaris per controlar-la. A continuació, es descriu com connectar i configurar aquesta pantalla amb la Raspberry Pi.

Materials necessaris:

1. **Pantalla LCD 16x2 amb mòdul i2C:** Aquest mòdul inclou un controlador i2C que simplifica la connexió, ja que només necessita 4 cables per comunicar-se amb la Raspberry Pi.
2. **Raspberry Pi:** Pot ser qualsevol model amb pins GPIO disponibles.
3. **Cables de connexió:** Per realitzar les connexions entre la pantalla i la Raspberry Pi.

La connexió es fa a través dels pins GPIO de la Raspberry Pi i el mòdul i2C de la pantalla. Els pins necessaris són:

1. **VCC a 5V:** Connecta el pin **VCC** del mòdul i2C de la pantalla LCD al pin de **5V** de la **Raspberry Pi**.
2. **GND a GND:** Connecta el pin **GND** del mòdul LCD al pin de **terra** (GND) de la **Raspberry Pi**.
3. **SDA a SDA (GPIO 2):** Connecta el pin **SDA** del mòdul i2C al pin **GPIO 2** de la **Raspberry Pi** (que és el pin de dades i2C).
4. **SCL a SCL (GPIO 3):** Connecta el pin **SCL** del mòdul I2C al pin **GPIO 3** de la **Raspberry Pi** (que és el pin de rellotge i2C).

### 3.2.3. Connexió del sensor de llum BH1750

El sensor de llum BH1750 és un sensor digital de mesura d'intensitat lumínica (llum ambiental) basat en i2C. Es fa servir sovint en projectes de mesura de llum en automàtiques, sistemes de control de llums o altres aplicacions d'automatització. A continuació et descriu com connectar el sensor BH1750 a una Raspberry Pi i com llegir els seus valors de manera senzilla.

Materials necessaris:

1. **Sensor BH1750:** Aquest sensor utilitza una interfície **i2C** per comunicar-se amb la **Raspberry Pi**.
2. **Raspberry Pi:** Amb pins GPIO disponibles i2C habilitat.
3. **Cables de connexió:** Per fer les connexions entre el sensor BH1750 i la **Raspberry Pi**.

El BH1750 disposa de 4 pins:

1. **VCC (Alimentació):** Connecta aquest pin al pin **5V** de la **Raspberry Pi** (els sensors BH1750 solen funcionar a 3.3V o 5V).
2. **GND (Terra):** Connecta aquest pin al pin **GND** de la **Raspberry Pi**.
3. **SDA (Dades):** Connecta aquest pin al pin **GPIO 2 (SDA)** de la **Raspberry Pi**.
4. **SCL (Rellotge):** Connecta aquest pin al pin **GPIO 3 (SCL)** de la **Raspberry Pi**.

**Configuració de la Raspberry Pi:**

1. **Habilitar i2C:**

- Obre una terminal i executa:  

```
sudo raspi-config
```

    - A l'opció **Interfacing Options**, selecciona **i2C** i actíva-ho.
    - Reinicia la **Raspberry Pi** perquè els canvis tinguin efecte.
2. **Instal·lar les biblioteques necessàries:** Per poder llegir els valors del sensor, necessitaràs instal·lar les biblioteques que permeten utilitzar i2C i accedir a les dades del sensor.

Executa els següents comandaments a la terminal per instal·lar les biblioteques:

```
sudo apt update
sudo apt install python-smbus i2c-tools
sudo apt install python3-smbus
```

**Comprovar que el sensor estigui connectat correctament:** Un cop hagis connectat el sensor, pots comprovar si el dispositiu i2C està reconegut amb la comanda següent:

```
sudo i2cdetect -y 1
```

3. Si tot està correcte, hauries de veure l'adreça i2C del sensor, normalment **0x23** o **0x5C**.

### 3.2.4. Connexió del sensor de temperatura DHT11

El **sensor de temperatura i humitat DHT11** és un sensor digital molt utilitzat en projectes amb la Raspberry Pi per mesurar la temperatura i la humitat de l'entorn. El DHT11 és senzill d'utilitzar i només necessita un pin per transmetre les dades, ja que utilitza un protocol de comunicació digital.

Materials necessaris:

1. **Sensor DHT11:** Aquest sensor té 3 pins: **VCC**, **GND** i **DATA**.
2. **Raspberry Pi:** Qualsevol model amb pins GPIO disponibles per realitzar les connexions.
3. **Resistència de 10kΩ:** Aquesta resistència es posa entre el pin **VCC** i el pin **DATA** per garantir una lectura fiable del sensor.
4. **Cables de connexió:** Per realitzar les connexions entre el sensor DHT11 i la **Raspberry Pi**.

El sensor DHT11 té 3 pins:

- **VCC:** Pin d'alimentació (connectat a 5V o 3.3V de la Raspberry Pi).
- **GND:** Pin de terra (connectat a GND de la Raspberry Pi).
- **DATA:** Pin de sortida de dades (connectat a un pin GPIO de la Raspberry Pi per llegir les dades de temperatura i humitat).

Les connexions es fan de la següent manera:

1. **VCC a 5V:** Connecta el pin **VCC** del DHT11 al pin **5V** de la **Raspberry Pi**.

2. **GND a GND:** Connecta el pin **GND** del DHT11 al pin de **terra** (GND) de la **Raspberry Pi**.
3. **DATA a GPIO:** Connecta el pin **DATA** del DHT11 a un pin GPIO de la **Raspberry Pi**. Per exemple, pots utilitzar el **GPIO 17**.
4. **Resistència de 10kΩ:** Col·loca una resistència de 10kΩ entre el pin **VCC** i el pin **DATA** per garantir una bona estabilitat a l'hora de llegir les dades.

### Configuració de la Raspberry Pi:

**Instal·lar les biblioteques necessàries:** Per poder llegir les dades del DHT11 des de la Raspberry Pi, necessitaràs una biblioteca com **Adafruit\_DHT**. La pots instal·lar amb els següents comandaments:

```
sudo apt update
```

```
sudo apt install python3-pip
```

```
sudo pip3 install Adafruit_Python_DHT
```

si aquesta comanda dona error d'execució, es pot forçar la instal·lació amb aquesta altra comanda:

```
sudo pip3 install --break-system-packages Adafruit_Python_DHT
```

**Comprovar les connexions i la lectura de dades:** Un cop instal·lada la biblioteca, pots escriure un petit programa en Python per llegir la temperatura i la humitat del DHT11.

Passos per a la instal·lació i ús de la biblioteca adafruit-circuitpython-dht:

1. Actualitza el sistema i instal·la les dependències necessàries:

```
sudo apt-get update
```

```
sudo apt-get install python3-pip libgpiod2
```

2. Instal·la la biblioteca adafruit-circuitpython-dht:

```
pip3 install adafruit-circuitpython-dht
```

si aquesta comanda dona error d'execució, es pot forçar la instal·lació amb aquesta altra comanda:

```
sudo pip3 install --break-system-packages adafruit-circuitpython-dht
```

3. Instal·la la biblioteca Adafruit Blinka, que proporciona la compatibilitat amb CircuitPython a la Raspberry Pi:

```
pip3 install adafruit-blinka
```

si aquesta comanda dona error d'execució, es pot forçar la instal·lació amb aquesta altra comanda:

```
sudo pip3 install --break-system-packages adafruit-blinka
```

4. Crea un fitxer Python, per exemple dht11\_test.py, amb el següent codi:

```
import time
import board
import adafruit_dht
Inicialitza el dispositiu DHT11 al pin GPIO4
dhtDevice = adafruit_dht.DHT11(board.D4)
try:
 while True:
 # Llegeix la temperatura i la humitat
 temperature_c = dhtDevice.temperature
 humidity = dhtDevice.humidity
 if temperature_c is not None and humidity is not None:
 print(f"Temperatura: {temperature_c:.1f}°C")
 print(f"Humitat: {humidity:.1f}%")
 else:
 print("Error en la lectura del sensor. Intentant de nou...")
 time.sleep(2.0)
except KeyboardInterrupt:
 print("\nLectura del sensor aturada.")
except Exception as error:
 dhtDevice.exit()
 raise error
```

5. Executa el codi amb Python 3:

```
python3 dht11_test.py
```

### 3.2.5. Connexió del sensor d'humitat de terra YI-69

El **sensor d'humitat de terra YI-69** és un sensor molt utilitzat per mesurar la humitat del sòl en projectes d'agricultura automatitzada o sistemes de reg intel·ligent. Aquest sensor mesura la humitat de la terra a través de la resistència de la humitat en el sòl, de manera que, com més humit estigui el sòl, menor serà la resistència i més alt serà el valor de lectura del sensor.

Materials necessaris:

1. **Sensor YI-69 (Sensor d'humitat de terra)**: Aquest sensor té tres pins: **VCC**, **GND** i **DO** (Digital Output) o **AO** (Analog Output).
2. **Raspberry Pi**: Qualsevol model amb pins GPIO disponibles.
3. **Cables de connexió**: Per fer les connexions entre el sensor YI-69 i la **Raspberry Pi**.

El sensor YI-69 es pot utilitzar en mode digital (per enviar una senyal **HIGH/LOW**) o en mode analògic (per llegir un valor continu de resistència). Aquí et descriu com connectar-lo en ambdós modes:

**Connexió en mode Digital:**

1. **VCC a 3.3V o 5V**: Connecta el pin **VCC** del sensor al pin **3.3V** o **5V** de la **Raspberry Pi** (el sensor pot funcionar amb ambdues tensions, però 3.3V és més segur per la **Raspberry Pi**).
2. **GND a GND**: Connecta el pin **GND** del sensor a un pin de **terra** (GND) de la **Raspberry Pi**.
3. **DO (Digital Output) a GPIO**: Connecta el pin **DO** (digital) del sensor a un pin GPIO de la Raspberry Pi. Per exemple, pots utilitzar el **GPIO 17**.

**Connexió en mode Analògic (requereix un convertidor ADC):**

El sensor YI-69 també pot treballar en mode analògic, però la **Raspberry Pi** no té un convertidor analògic a digital (ADC) integrat. Per llegir valors analògics, caldrà utilitzar un mòdul ADC com el **MCP3008**. Així, els passos serien:

1. **VCC a 3.3V o 5V**: Connecta el pin **VCC** al pin **3.3V** o **5V** de la **Raspberry Pi**.
2. **GND a GND**: Connecta el pin **GND** a un pin de **terra** (GND) de la **Raspberry Pi**.
3. **AO (Analog Output) a un canal ADC (per exemple, canal CH0 del MCP3008)**: Connecta el pin **AO** (analògic) a un dels canals del mòdul ADC MCP3008.

**Configuració de la Raspberry Pi:**

1. **Instal·lar les biblioteques necessàries**: Per llegir el sensor YI-69 en mode digital, no cal instal·lar cap biblioteca addicional si utilitzes la biblioteca **RPi.GPIO**. Si utilitzes el mòdul MCP3008 per llegir valors analògics, hauràs d'instal·lar la biblioteca **spidev**.

Executa els següents comandaments per instal·lar **spidev** si utilitzes un ADC:

```
sudo apt update
```

```
sudo apt install python3-spidev
```

#### 1. Habilita la interfície SPI

SPI és una interfície que ha de ser activada explícitament a la Raspberry Pi.

#### Utilitzant `raspi-config`:

Obre el terminal i executa:

```
sudo raspi-config
```

1. Ves a l'opció "**Interfacing Options**" i selecciona "**SPI**".
2. Quan et pregunti si vols activar SPI, selecciona **Yes**.
3. Reinicia la Raspberry Pi per aplicar els canvis:

```
sudo reboot
```

#### PINOUT DEL SENSOR D'HUMITAT DEL SÒL CAPACITIU

El sensor d'humitat del sòl capacitiu té quatre pins:

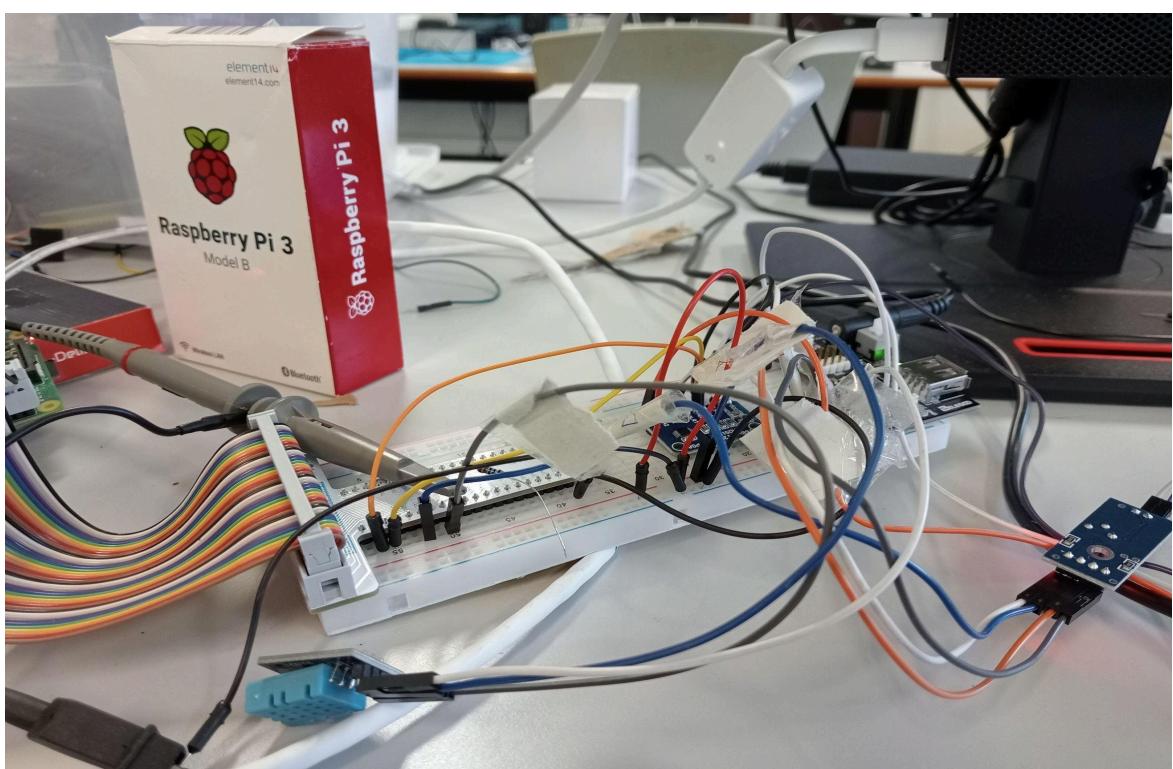
Pin GND: Aquest pin s'ha de connectar al GND (0V).

Pin VCC: Aquest pin s'ha de connectar al VCC (5V o 3,3V).

Pin DOUT: Pin de sortida de senyal digital que produeix un voltatge proporcional al nivell d'humitat del terra. S'ha de connectar a un pin d'entrada digital d'una Raspberry Pi.

Pin AOUT: Pin de sortida de senyal analògic que produeix un voltatge proporcional al nivell d'humitat del terra. S'ha de connectar a un pin d'entrada analògica d'una Raspberry Pi.

La quantitat d'aigua al sòl afecta el voltatge al pin AOUT; com més gran sigui el contingut d'aigua, més baix serà el voltatge.



```

Sensor Aigua (A0): 26037 (100.00%)
Sensor Humitat Terra (A0): 26037 (100.00%)
Sensor Aigua (A3): 0 (0.00%)
Sensor Humitat Terra (A0): 26036 (100.00%)
Sensor Aigua (A3): 0 (0.00%)
Sensor Humitat Terra (A0): 26034 (100.00%)
Sensor Aigua (A3): 1 (0.00%)
Sensor Humitat Terra (A0): 26032 (100.00%)
Sensor Aigua (A3): 1 (0.00%)
Sensor Humitat Terra (A0): 26032 (100.00%)
Sensor Aigua (A3): 0 (0.00%)
Sensor Humitat Terra (A0): 26033 (100.00%)
Sensor Aigua (A3): 1 (0.00%)
Sensor Humitat Terra (A0): 26032 (100.00%)
Sensor Aigua (A3): 0 (0.00%)
Sensor Humitat Terra (A0): 26031 (100.00%)
Sensor Aigua (A3): 1 (0.00%)
Sensor Humitat Terra (A0): 26034 (100.00%)
Sensor Aigua (A3): 1 (0.00%)
Sensor Humitat Terra (A0): 26019 (100.00%)
Sensor Aigua (A3): 1 (0.00%)
Sensor Humitat Terra (A0): 26024 (100.00%)
Sensor Aigua (A3): 1 (0.00%)
Sensor Humitat Terra (A0): 26030 (100.00%)
Sensor Aigua (A3): 0 (0.00%)
Sensor Humitat Terra (A0): 26034 (100.00%)
Sensor Aigua (A3): 1 (0.00%)
Sensor Humitat Terra (A0): 26031 (100.00%)
Sensor Aigua (A3): 0 (0.00%)
Sensor Humitat Terra (A0): 26024 (100.00%)
Sensor Aigua (A3): 0 (0.00%)
Sensor Humitat Terra (A0): 26019 (100.00%)
Sensor Aigua (A3): 0 (0.00%)
Sensor Humitat Terra (A0): 26025 (100.00%)

```

### Com calibrar el sensor d'humitat (Soil Moisture)

1. Col·loca el sensor a la terra seca.
2. Premsa la terra al voltant del sensor.
3. Executa el codi capacitive-soil-pico.py des de la Raspberry Pi Pico connectada al teu ordinador i comprova el monitor sèrie (terminal) per comparar els valors llegits.

4. Pren el valor més alt i assigna'l a la variable max\_moisture dins del fitxer capacitive-soil-pico.py.

Fitxer capacitive-soil.pico.py

```
import time
from Adafruit_ADS1x15 import ADS1115
Crea una instància de l'ADS1115
adc = ADS1115(busnum=1)
Configura el guany (ajusta segons el rang del sensor, aquí 4.096V)
GAIN = 1

def read_sensors():
 # Llegeix els valors dels canals A0 i A3
 water_level = adc.read_adc(0, gain=GAIN)
 soil_moisture = adc.read_adc(3, gain=GAIN)

 # Converteix a volts (16 bits, 4.096V màxim)
 water_voltage = water_level * 4.096 / 32768.0
 soil_voltage = soil_moisture * 4.096 / 32768.0
 return water_voltage, soil_voltage

def main():
 while True:
 # Llegeix els sensors
 soil_voltage, water_voltage = read_sensors()
 # Mostra els resultats
 print(f"Water Level Voltage: {water_voltage:.2f} V")
 print(f"Soil Moisture Voltage: {soil_voltage:.2f} V")
 print("=" * 30)
 # Petita pausa per alternar
 time.sleep(1)

if __name__ == "__main__":
 main()
```

```
Water Level Voltage: 0.53 V
=====
Water Level Voltage: 1.92 V
Soil Moisture Voltage: 0.54 V
=====
Water Level Voltage: 1.97 V
Soil Moisture Voltage: 0.54 V
=====
Water Level Voltage: 1.98 V
Soil Moisture Voltage: 0.54 V
=====
Water Level Voltage: 1.99 V
Soil Moisture Voltage: 0.54 V
=====
Water Level Voltage: 1.99 V
Soil Moisture Voltage: 0.52 V
=====
Water Level Voltage: 2.00 V
Soil Moisture Voltage: 0.52 V
=====
Water Level Voltage: 2.00 V
Soil Moisture Voltage: 0.53 V
=====
Water Level Voltage: 2.00 V
Soil Moisture Voltage: 0.53 V
=====
Water Level Voltage: 2.01 V
Soil Moisture Voltage: 0.53 V
=====
Water Level Voltage: 2.01 V
Soil Moisture Voltage: 0.53 V
=====
Water Level Voltage: 2.01 V
Soil Moisture Voltage: 0.53 V
=====
Water Level Voltage: 2.01 V
Soil Moisture Voltage: 0.53 V
=====
Water Level Voltage: 2.01 V
Soil Moisture Voltage: 0.53 V
=====
Water Level Voltage: 2.01 V
Soil Moisture Voltage: 0.54 V
=====
Water Level Voltage: 2.01 V
Soil Moisture Voltage: 0.54 V
=====
Water Level Voltage: 2.01 V
Soil Moisture Voltage: 0.54 V
=====
Water Level Voltage: 2.01 V
Soil Moisture Voltage: 0.54 V
```

### 3.2.6. Connexió Pi Camera

La connexió de la Pi Camera a una Raspberry Pi és un procés senzill que permet integrar funcions de captura d'imatges i vídeos d'alta qualitat en els teus projectes. El mòdul de càmera es connecta mitjançant el port CSI (Camera Serial Interface), situat directament a la placa Raspberry Pi.

Per començar, assegura't que la Raspberry Pi estigui apagada per evitar possibles danys al connectar el dispositiu. Localitza el port CSI, que es troba a prop del connector HDMI o dels GPIO, depenent del model de la Raspberry Pi. Obre amb cura el pestell del connector i introduceix la cinta plana de la càmera amb els contactes metàl·lics orientats cap al processador. Tanca el pestell per assegurar la connexió.

Un cop connectada físicament, engega la Raspberry Pi i obre el terminal per habilitar la càmera. Executa l'ordre sudo raspi-config i accedeix al menú "Interface Options". Activa l'opció "Camera" i reinicia la Raspberry Pi perquè els canvis tinguin efecte.

Amb la càmera configurada, pots començar a utilitzar-la amb eines com raspistill per capturar imatges o raspivid per enregistrar vídeo. Si treballeres amb programació, les llibreries com picamera2 per a Python o OpenCV ofereixen un control avançat sobre les funcionalitats de la càmera.

És important assegurar-te que tens la versió correcta de la càmera (1, 2 o HQ) i que és compatible amb el model de la teva Raspberry Pi. A més, verifica que el cable estigui correctament connectat i sense doblegar-se per garantir una comunicació estable.

Aquest procés de connexió és ràpid i, gràcies a la gran comunitat de Raspberry Pi, pots trobar suport en línia per resoldre problemes o explorar funcionalitats addicionals. Amb la Pi Camera connectada, podràs desenvolupar projectes de visió per computador, vigilància, seguiment d'objectes i molt més.

#### Materials necessaris:

1. **Raspberry Pi** (model 3B, 4 o superior és recomanable).
2. **Pi Camera**:
3. **Cable de connexió** compatible amb el port CSI de la Raspberry Pi.
4. **Font d'alimentació per a Raspberry Pi**: una font de 5V amb un corrent mínim de 2.5A (per assegurar una alimentació estable).
5. **Targeta microSD**: amb el sistema operatiu Raspberry Pi OS instal·lat.
6. **Cables jumper** (mascle-femella o mascle-mascle segons necessitis).
7. **Protoboard** (opcional, però molt útil per fer connexions temporals).
8. **Monitor o pantalla HDMI** (opcional): per comprovar la visualització en temps real de la càmera durant les proves.
9. **Software**: Raspberry Pi OS i una biblioteca de Python com *RPi.GPIO*.
10. **Cables GPIO i Protoboard** (opcional): si vols integrar el control de la càmera amb altres components o sensors.
11. **Carcassa compatible amb Pi Camera** (opcional): per protegir la càmera i la Raspberry Pi durant el muntatge.

### 3.3. Configuració de la Protoboard

Una placa Protoboard de 830 punts és una eina útil per a la creació de circuits electrònics de manera provisional i sense soldadura. Aquesta placa està dissenyada per permetre connectar components electrònics (com resistències, LEDs, transistors, etc.) de forma fàcil i ràpida.

La configuració típica d'una Protoboard de 830 punts es divideix en tres seccions:

- > Secció central (filament de connexió): Aquesta part ocupa la major part de la placa i està formada per una sèrie de línies de connexió. Hi ha dues files de 24 punts (generalment marcades amb números) que es poden connectar a components i cables. Les files estan connectades internament en línies horizontals, el que permet la connexió dels pins dels components entre ells sense necessitat de soldadura.
- > Files d'alimentació (línies laterals): A cada costat de la Protoboard, hi ha dues línies d'alimentació. Aquestes línies són generalment marcades amb una línia vermella per al

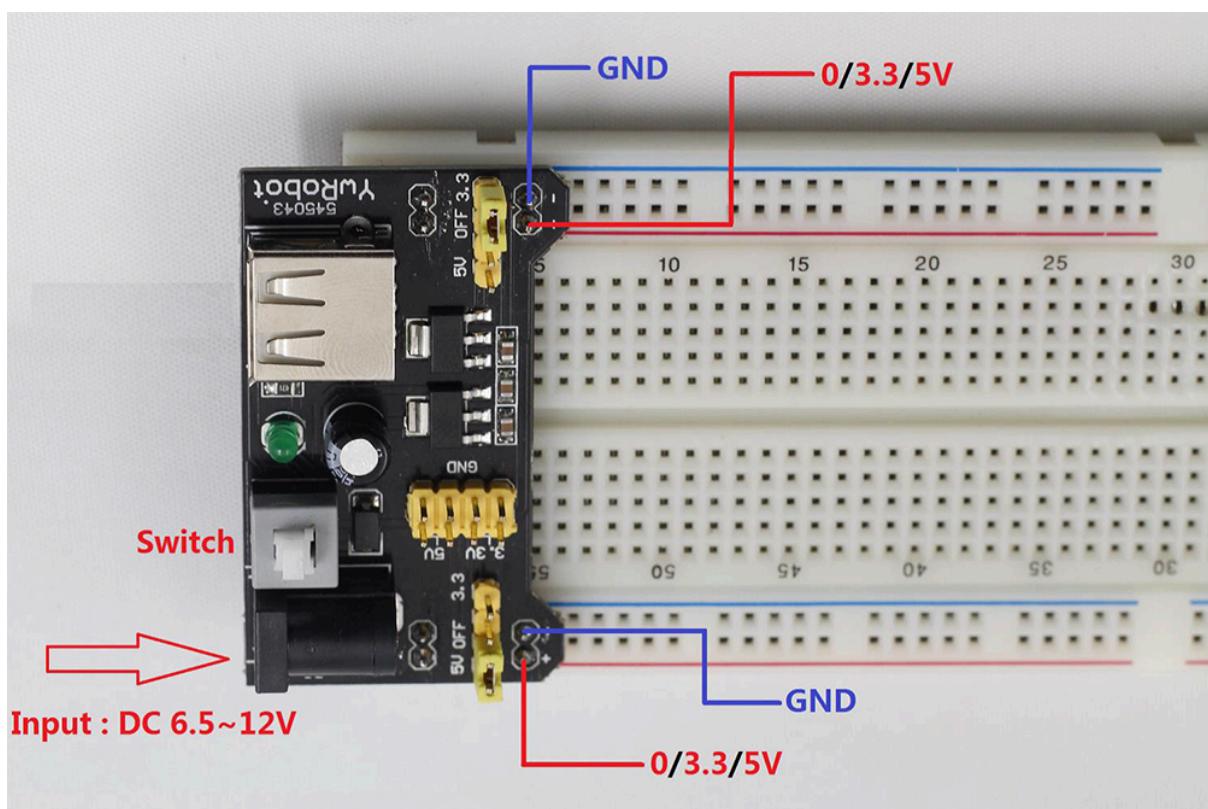
voltatge positiu (+) i una línia blava o negra per a la terra (-). Aquestes línies s'estenen per tota la longitud de la Protoboard i permeten distribuir el voltatge i la terra als components del circuit.

> Distribució de punts: La Protoboard té un total de 456 punts de connexió disponibles. Aquests punts estan organitzats de manera que faciliten la connexió dels components, amb les files centrals dedicades a les connexions de components i les línies laterals destinades a l'alimentació i la terra.

Amb aquesta configuració, pots muntar circuits complexos, experimentar amb components electrònics, i fer connexions ràpides i netes sense la necessitat de soldar res. La versatilitat de la Protoboard fa que sigui una eina ideal per a prototips i aprendre sobre l'electrònica.

### 3.4. Connexió de l'alimentació

Utilitzarem com a font d'alimentació, el YwRobot Power Supply Properly (Power MB U2). Es tracta d'una font d'alimentació utilitzada principalment per a Protobards i circuits electrònics. Aquesta font proporciona una manera senzilla i eficient d'alimentar els projectes electrònics amb voltatges estables i regulats.



Algunes característiques principals d'aquest dispositiu inclouen:

> Voltages regulats: El Power MB U2 ofereix diverses opcions de voltatge de sortida, com 5V i 3.3V, que són comuns per a molts components electrònics.

Protecció de sobrecàrrega: Disposa de circuits de protecció que impedeixen danys en cas de sobrecàrrega o curtcircuit.

- > Connexions fàcils: S'integra fàcilment amb una Protoboard per proporcionar alimentació estable a diversos components sense necessitat de soldadura.
- Indicadors LED: Els LEDs indiquen l'estat de l'alimentació, amb senyals clars que mostren quan la sortida de tensió és activa.
- > Muntatge senzill: Aquest dispositiu és compacte i fàcil de muntar en projectes de prototipat, facilitant la creació de circuits ràpidament.

El YwRobot Power Supply MB U2 és molt utilitzat en entorns educatius i prototipatge gràcies a la seva simplicitat i fiabilitat.

## 4. Programació del Sistema

### 4.1. Instal·lació de biblioteques i eines necessàries

#### 4.1.1. Instal·lació de Python i GPIO

Per a començar a treballar amb Python i controlar els pins GPIO de la Raspberry Pi, primer cal assegurar-se que Python estigui instal·lat. La majoria de les distribucions de Raspberry Pi ja vénen amb Python preinstal·lat. Per verificar-ho, simplement obre un terminal i executa:

```
python3 --version
```

per comprovar la versió de Python 3 disponible. Si no tens Python, el pots instal·lar amb aquesta comanda:

```
sudo apt-get install python3
```

Pel que fa a la biblioteca GPIO, la Raspberry Pi utilitza la biblioteca RPi.GPIO per controlar els pins de la placa. Aquesta biblioteca també sol estar preinstal·lada en la majoria de les imatges del sistema operatiu. Si no la tens, la pots instal·lar executant:

```
sudo apt-get install python3-rpi.gpio
```

Un cop instal·lades les biblioteques, pots començar a escriure el teu codi Python per controlar els pins GPIO. Per començar, cal importar la biblioteca mitjançant aquesta línia de codi:

```
import RPi.GPIO as GPIO
```

Després, cal configurar el mode de numeració dels pins, que es fa amb una d'aquestes dues comandes:

```
GPIO.setmode(GPIO.BCM) # per utilitzar la numeració de la BCM.
```

```
GPIO.setmode(GPIO.BOARD) # per utilitzar la numeració de la placa.
```

Un cop configurat el mode de numeració, pots configurar un pin com a sortida o entrada amb les comandes:

```
GPIO.setup(pin, GPIO.OUT) # canal de sortida.
```

```
GPIO.setup(pin, GPIO.IN) # canal d'entrada.
```

Per exemple, per encendre un LED connectat a un pin de la Raspberry Pi, es podria fer amb aquestes altres comandes:

```
GPIO.output(pin, GPIO.HIGH) # per encendre el pin
```

```
GPIO.output(pin, GPIO.LOW) # per apagar el pin.
```

Finalment, és recomanable netejar la configuració de les biblioteques amb la línia de codi:

```
GPIO.cleanup()
```

un cop acabat el codi per evitar conflictes amb futurs programes que utilitzin els pins GPIO.

#### 4.1.2. Instal·lació de llibreries per a LCD i2C

Per a utilitzar una pantalla LCD amb un controlador i2C a la Raspberry Pi, és necessari instal·lar biblioteques específiques. Una de les més utilitzades és Adafruit\_CharLCD. Primer, cal instal·lar les biblioteques requerides per i2C mitjançant la línia de comandes. Això es fa executant:

```
sudo apt-get install python-smbus i2c-tools
```

Després, per instal·lar la biblioteca Adafruit\_CharLCD, s'ha d'executar:

```
pip install adafruit-circuitpython-charlcd
```

Un cop instal·lades les biblioteques, es pot accedir a la pantalla mitjançant Python amb les ordres corresponents per a configurar i2C, la mida de la pantalla i escriure text.

També és important habilitar, abans de començar a programar, la interfície i2C a la Raspberry Pi mitjançant

```
raspi-config
```

Així, podràs controlar la teva pantalla LCD i2C de manera senzilla.

## 4.2. Esquema bàsic de programació

### 4.2.1. Control del sensor d'aigua.

El control d'un sensor d'aigua amb la microcontroladora Raspberry Pi es pot realitzar mitjançant un sensor de tipus digital o analògic. Un dels sensors més comuns és el sensor de nivell d'aigua, que detecta la presència d'aigua. Per a la connexió del sensor, si és digital, es connecta a un pin GPIO de la Raspberry Pi, i es pot llegir l'estat (ON/OFF) mitjançant un simple script en Python.

```
import RPi.GPIO as GPIO
import time
```

```

GPIO.setmode(GPIO.BCM)
GPIO.setup(17, GPIO.IN) # El sensor està connectat al pin GPIO 17

while True:
 if GPIO.input(17): # Si el sensor detecta aigua
 print("Aigua detectada!")
 else:
 print("No s'ha detectat aigua.")
 time.sleep(1)

```

Si el sensor és analògic, serà necessari un convertidor analògic-digital (ADC) com el ADS1115, ja que la Raspberry Pi no té entrades analògiques. El valor llegeix la tensió del sensor per determinar el nivell d'aigua o la humitat. Un cop llegit el valor, es pot utilitzar per activar o desactivar actuadors com una bomba d'aigua o per mostrar alertes.

#### CODI PER MESURAR EL NIVELL D'AIGUA:

```

import time
import Adafruit_ADS1x15

Create an ADS1115 ADC instance
ADC = Adafruit_ADS1x15.ADS1115(busnum = 1)

Specify the ADC channel (0-3) based on your connection
ADC_CHANNEL = 3 # A3 of ADS1115 module

Set the gain (input voltage range) for your application
GAIN = 1 # Gain of 1 corresponds to +/-4.096V

Define the conversion factor for water level calculation
MIN_ADC_VALUE = 14134 # Replace with the minimum ADC value for your sensor
MAX_ADC_VALUE = 25470 # Replace with the maximum ADC value for your sensor

try:
 while True:
 # Read the raw ADC value
 adc_value = ADC.read_adc(ADC_CHANNEL, gain=GAIN)

 # Convert the raw ADC value to a water level percentage
 water_level = (adc_value - MIN_ADC_VALUE) / (MAX_ADC_VALUE - MIN_ADC_VALUE) * 100

 print(f"ADC Value: {adc_value} | Water Level: {water_level:.2f}%")

 time.sleep(1) # Wait for a second before the next reading

except KeyboardInterrupt:
 print("\nScript terminated by user.")

```



Python. Primer, cal importar la biblioteca i configurar la connexió I2C amb la pantalla, indicant l'adreça I2C i la mida de la pantalla, per exemple, 16x2 (16 caràcters per 2 línies).

Per exemple, es podria escriure el codi següent:

```
import board
import digitalio
import adafruit_character_lcd.characterlcd as characterlcd
from time import sleep

lcd = characterlcd.Character_LCD_I2C(board.I2C(), 16, 2)
lcd.clear()
lcd.message = "Projecte Hort Urbà!"
```

Amb aquest codi, el missatge "Projecte Hort Urbà!" es mostrerà a la pantalla LCD. També es poden utilitzar funcions per controlar el cursor, netejar la pantalla (`lcd.clear()`), o ajustar la posició del text utilitzant el mètode `lcd.set_cursor(fila, columna)`.

El control de la pantalla LCD també permet utilitzar caràcters especials, fer que el text es mogui o fins i tot mostrar informació dinàmica. D'aquesta manera, la Raspberry Pi pot comunicar-se visualment amb l'usuari de manera senzilla i eficaç.

#### 4.2.3. Control del sensor de llum BH1750

El sensor de llum BH1750 és un dispositiu digital de gran precisió utilitzat per mesurar la intensitat de llum ambiental. Funciona mitjançant interfície I2C, cosa que permet una comunicació senzilla amb microcontroladors com Arduino o Raspberry Pi. Aquest sensor és molt eficient en termes de consum d'energia, ideal per a aplicacions com sistemes d'il·luminació automàtica, dispositius portàtils o estacions meteorològiques.

El BH1750 pot operar en diferents modes, com ara la mesura contínua o la mesura única, amb diferents resolucions per ajustar-se a les necessitats del projecte. Per utilitzar-lo, es configuren les adreces I2C i es llegeixen les dades mitjançant registres interns. Gràcies a la seva precisió i facilitat d'ús, és una solució popular per a sistemes que requereixen control automàtic basat en la il·luminació ambiental.

#### 4.2.4. Control del sensor de temperatura DHT11

El sensor de temperatura i humitat DHT11 és un dispositiu senzill i econòmic utilitzat per mesurar la temperatura i la humitat relativa de l'aire. El seu control es realitza mitjançant una comunicació digital a través d'un sol pin, el que simplifica la seva connexió amb microcontroladors com Arduino o Raspberry Pi.

Per controlar el DHT11, primer es necessita inicialitzar el sensor enviant una senyal d'arrencada a través del pin de dades. El sensor respon amb un paquet de dades que conté la temperatura i la humitat en format digital. Aquestes dades s'han de llegir i processar mitjançant un codi específic que decodifica el senyal de resposta.

## CODI PER MESURAR LA TEMPERATURA I LA HUMITAT

```
import time
import board
import adafruit_dht

Inicialitza el dispositiu DHT11 al pin GPIO22
dhtDevice = adafruit_dht.DHT11(board.D4)

try:
 while True:
 # Llegeix la temperatura i la humitat
 temperature_c = dhtDevice.temperature
 humidity = dhtDevice.humidity
 if temperature_c is not None and humidity is not None:
 print("Temperatura: {:.1f}°C")
 print("Humitat: {:.1f}%")
 else:
 print("Error en la lectura del sensor. Intentant de nou...")
 time.sleep(2.0)
except KeyboardInterrupt:
 print("\nLectura del sensor aturada.")
except Exception as error:
 dhtDevice.exit()
 raise error
```

Un cop rebuda la informació, els valors de temperatura i humitat poden ser utilitzats per controlar altres dispositius o per mostrar-los en una pantalla. El DHT11 té una precisió limitada, amb un rang de temperatura de 0 a 50 °C i una humitat relativa del 20% al 80%, amb una precisió d'uns 5%. És ideal per a aplicacions senzilles de control ambiental o d'aprenentatge.

### 4.2.5. Control del sensor d'humitat de terra YI-69

El sensor d'humitat del terra s'utilitza per mesurar la quantitat d'aigua present en el sòl, ajudant a controlar el reg en aplicacions d'agricultura automatitzada i jardins intel·ligents. El sensor funciona mesurant la resistència elèctrica entre dos elèctrodes que es col·loquen al terra. Com més humit és el sòl, més baixa serà la resistència, ja que l'aigua condueix l'electricitat.

Per controlar el sensor, es connecta a un microcontrolador com un Arduino o Raspberry Pi. El sensor envia un valor analògic o digital (segons el model) que es pot llegir per determinar el nivell d'humitat del terra. Aquest valor es pot processar per activar un sistema de reg automàtic o per alertar l'usuari si el sòl necessita aigua.

A través de programació, el microcontrolador pot establir límits específics d'humitat, de manera que quan el valor mesurat és inferior a un cert punt, el sistema activa el reg. Aquest tipus de control ajuda a optimitzar el consum d'aigua i mantenir unes condicions òptimes per al creixement de les plantes.

## CODI PER MESURAR LA HUMITAT DE TERRA

```
import spidev
import time

class SensorHumitatTerraYI69:

 def __init__(self, canal=0, bus=0, device=0):
 self.canal = canal
 self.bus = bus
 self.device = device

 # Inicialitzacio de l'ADC MCP3008
 self.spi = spidev.SpiDev()
 self.spi.open(self.bus, self.device)
 self.spi.max_speed_hz = 1350000 # Configuració de la velocitat SPI

 def llegir_analogic(self):

 # Enviar la comanda per llegir del canal seleccionat
 request = [1, (8 + self.canal) << 4, 0]
 resposta = self.spi.xfer2(request)

 # Convertir la resposta de l'ADC (10 bits)
 resultat = ((resposta[1] & 3) << 8) + resposta[2]
 return resultat

 def llegir_humitat(self):
 # Llegeix el valor analogic
 valor_analogic = self.llegir_analogic()
 print(f"Valor analogic: {valor_analogic:.2f}%")
 # Convertir el valor a percentatge de humitat (0-100%)
 # Considerant que el sensor té un rang de 0 (sec) a 1023 (molt humit).
 humitat = (valor_analogic / 1023.0) * 100
 return humitat

 def tancar(self):
 self.spi.close()

Prova de la classe
if __name__ == "__main__":
 try:
 # Crear una instancia del sensor
 print("Prova obrir ")
 sensor = SensorHumitatTerraYI69()

 # Llegir i mostrar valors d'humitat cada segon
 print("Llegeixo valors d'humitat del sensor YI-69 (Ctrl+C per aturar).")
 while True:
 humitat = sensor.llegir_humitat()
 print(f"Humitat del sòl: {humitat:.2f}%")
 time.sleep(1)
 except KeyboardInterrupt:
 print("\nLectura aturada per l'usuari.")
```

```
finally:
 # Tancar la connexió SPI abans de sortir
 sensor.tancar()
```

#### 4.2.6. Control de Pi Camera

El control de la Pi Camera és essencial per maximitzar les seves capacitats en projectes basats en Raspberry Pi. Aquesta càmera es connecta mitjançant el port CSI (Camera Serial Interface) i es configura fàcilment mitjançant l'eina raspi-config. Un cop habilitada, es pot gestionar amb llibreries com picamera2 per a Python, que ofereix un accés senzill i potent a les seves funcionalitats.

Amb la llibreria picamera2, pots capturar imatges i vídeos amb només unes poques línies de codi. La llibreria permet ajustar paràmetres com la resolució, la qualitat de compressió i la velocitat de fotogrames. També proporciona control manual sobre configuracions avançades com l'exposició, el balanç de blancs, l'ISO i el focus, cosa que és útil per a aplicacions personalitzades.

El control es pot estendre a funcions més sofisticades, com la detecció de moviment, la captura de timelapse o l'ús en projectes d'intel·ligència artificial. Per exemple, combinant la Pi Camera amb OpenCV o TensorFlow, pots implementar sistemes de reconeixement facial, seguiment d'objectes o classificació d'imatges.

També és possible utilitzar ordres bàsiques al terminal, com raspistill per capturar fotos o raspivid per enregistrar vídeo. Aquestes eines permeten configurar opcions com el temps d'exposició, l'angle de rotació o aplicar filtres visuals.

El control remot de la càmera és factible mitjançant connexions SSH, cosa que permet administrar projectes des d'un ordinador diferent. Amb una àmplia comunitat de suport i exemples disponibles en línia, controlar la Pi Camera és una experiència accessible per a principiants i una eina potent per a desenvolupadors avançats.

### 5. Conclusió

#### 5.1. Resum dels resultats obtinguts

Inicialment, la meva aspiració era poder implementar i connectar un total de 6 sensors a la placa Raspberry Pi. La falta d'experiència en el món Maker i els coneixements limitats en electrònica i microcontroladors Raspberry Pi ha sigut cabdal per no reeixir la totalitat del projecte. D'altra banda, els objectius en aquest projecte eren, d'entrada, massa elevats.

En el termini donat per realitzar el projecte només s'ha pogut activar 3 sensors:

- a. Sensor d'Aigua
- b. Sensor de temperatura DHT11
- c. Sensor d'humitat de terra YI-69

La resta de sensors (pantalla LCD - sensor de llum BH1750 i Pi Camera) restan pendents de posar en pràctica fora del curs CREACIÓ DE PROTOTIPS IOT AMB RASPBERRY.

## 5.2. Avaluació del sistema

Connectivitat i muntatge:

- Els sensors es connecten a la Raspberry Pi mitjançant GPIO i el bus i2C (per al BH1750 i la pantalla LCD).
- L'ús d'un Protoboard facilita les connexions inicials i simplifica el prototipatge.
- Es podria considerar un PCB personalitzat per a la versió final.

Integració de sensors:

- El sensor d'humitat de terra YI-69 activa automàticament una bomba d'aigua quan el sòl està sec.
- El sensor de llum BH1750 optimitza l'ús de llum artificial en absència de llum solar suficient.
- El DHT11 ajuda a prevenir condicions extremes d'humitat o calor, activant alertes o sistemes de ventilació.

Interacció amb l'usuari:

- La pantalla LCD ofereix informació útil en temps real al jardiner, com els valors de llum, humitat i temperatura.
- Opcionalment, es pot afegir un sistema d'avars (via missatges i/o notificacions) en cas de paràmetres crítics.

Documentació i seguiment:

- La Pi Camera permet capturar imatges diàries per fer un timelapse o analitzar el creixement de les plantes. També pot detectar anomalies visuals com plagues o malalties.

Automatització i eficiència energètica:

- La Raspberry Pi controla els sensors i actuadors mitjançant programació en Python, optimitzant els recursos del jardí.
- Es podria integrar amb una font d'energia solar per fer el sistema més sostenible.

Punts forts:

- Solució modular i escalable, fàcil d'ampliar amb més sensors o funcionalitats.
- Muntatge senzill gràcies al Protoboard i a la compatibilitat dels sensors amb la Raspberry Pi.
- Capacitat de monitorització en temps real i documentació detallada.

Limitacions:

- La precisió del DHT11 és limitada en comparació amb altres sensors més avançats com el DHT22.
- Dependència d'una bona configuració inicial del sistema d'irrigació i il·luminació.
- L'ús prolongat del Protoboard podria provocar inestabilitat elèctrica; es recomana un PCB per a la versió final.

### 5.3. Limitacions del projecte

#### 1. Durabilitat en l'exterior:

El microcontrolador Raspberry Pi no està dissenyat per a condicions ambientals extremes: la pluja, la humitat, la calor i el fred poden danyar els components electrònics si no es protegeixen correctament.

#### 2. Dependència d'energia

Necessita una font d'alimentació constant. Si el projecte s'implementa en un hort urbà remot, pot requerir bateries o panells solars, que poden ser costosos o poc fiables en condicions meteorològiques adverses.

#### 3. Cobertura Wi-Fi

En horts grans o allunyats d'un entorn urbà, la connexió Wi-Fi pot ser insuficient per transmetre dades de manera fiable. Això limita la capacitat de monitoratge en remot.

#### 4. Complexitat en la configuració

Configurar sensors, càmeres, o altres perifèrics amb el Raspberry Pi pot ser tècnicament complicat, especialment per a persones amb poca experiència en programació o electrònica.

#### 5. Limitacions de maquinari

Els pins GPIO tenen un nombre limitat, cosa que restringeix el nombre de sensors connectats simultàniament. Això podria requerir anar ampliant el sistema amb components addicionals.

#### 6. Interferències ambientals

Les condicions ambientals com la pols, insectes, o la llum solar directa poden interferir amb sensors òptics o d'humitat, afectant la precisió en la medició de les dades.

#### 7. Cost addicional dels perifèrics

Els sensors, càmeres, mòduls de reg, i altres components necessaris poden augmentar significativament el cost total del projecte, especialment si es busquen perifèrics de qualitat.

#### 8. Manquen opcions d'alimentació robustes

La majoria dels sistemes Raspberry Pi depenen d'alimentació USB. Aquesta mena d'alimentació no sempre és fiable en entorns exteriors.

#### 9. Seguretat del dispositiu

El Raspberry Pi pot ser susceptible a robatoris o vandalisme si es deixa en zones accessibles o poc vigilades.

#### 10. Manteniment continu

Requereix manteniment periòdic per comprovar les connexions, netejar els sensors, actualitzar el programari, i reparar possibles danys. Això pot ser laboriós si es tracta d'horts grans.

## 5.4. Millores futures

El projecte de sensors en un hort urbà es podria anar ampliant amb d'altres sensors més complexos i que gestionin altres mètriques funcionals en la detecció de variables o paràmetres crítics com podria ser la visualització de l'entorn mitjançant una xarxa de càmeres (més sensors Raspberry Camera Pi) ubicades en llocs estratègics, que captarien la presència de petits rosegadors o indicis de possibles plagues com erugues o insectes. També seria molt interessant mesurar la pressió atmosfèrica al mode d'un baròmetre (Raspberry Pi Barometer Weather Clock). I vincular leds o buzzers en determinats sensors. D'aquesta manera, quan es sobrepassin valors màxims o mínims en el sensor podem rebre advertències visuals o sonores.

## 6. Annexos

### 6.1. Codi font complet

```
import time
import board
import Adafruit_ADS1x15
import adafruit_dht

Crear una instància del ADC ADS1115
ADC = Adafruit_ADS1x15.ADS1115(busnum = 1)

Adreça I2C del ADS1115 (per defecte 0x48)
ADS1115_ADDRESS = 0x48

Canals del ADC
WATER_SENSOR_CHANNEL = 3 # A3 per al sensor d'aigua
SOIL_SENSOR_CHANNEL = 0 # A0 per al sensor d'humitat de terra

Guany del ADC (rang de tensió d'entrada)
GAIN = 1 # ±4.096V

Calibració del sensor d'aigua
MIN_ADC_VALUE_WATER = 0 # Valor màxim en condicions seques
MAX_ADC_VALUE_WATER = 26068 # Valor màxim submergit en aigua

Calibració del sensor d'humitat de terra
MIN_ADC_VALUE_SOIL = 0 # Valor màxim en condicions seques
MAX_ADC_VALUE_SOIL = 25500 # Valor màxim en condicions humides

Inicialitza el dispositiu d'humitat i temperatura DHT11 al pin GPIO4
dhtDevice = adafruit_dht.DHT11(board.D4)
```

```

def llegir_percentatgeadc_value, min_val, max_val):
 """
 Converteix el valor digitalitzat a un percentatge.
 :param adc_value: Valor digital del ADC.
 :param min_val: Valor mínim del sensor.
 :param max_val: Valor màxim del sensor.
 :return: Percentatge calculat (0-100%).
 """
 percentatge = (adc_value - min_val) / (max_val - min_val) * 100
 return max(0, min(100, percentatge)) # Assegurar que el valor està entre 0 i
100

try:
 while True:
 # Llegir el canal A3 (sensor d'aigua)
 water_value = ADC.read_adc(WATER_SENSOR_CHANNEL, gain=GAIN)
 water_percentage = llegir_percentatge(water_value,
MIN_ADC_VALUE_WATER, MAX_ADC_VALUE_WATER)

 # Llegir el canal A0 (sensor d'humitat de terra)
 soil_value = ADC.read_adc(SOIL_SENSOR_CHANNEL, gain=GAIN)
 soil_percentage = llegir_percentatge(soil_value, MIN_ADC_VALUE_SOIL,
MAX_ADC_VALUE_SOIL)

 # Llegit el pin 4 (sensor d'humitat i temperatura)
 temperature_c = dhtDevice.temperature
 humidity = dhtDevice.humidity
 if temperature_c is not None and humidity is not None:
 print(f"Temperatura: {temperature_c:.1f}°C")
 print(f"Humitat: {humidity:.1f}%")
 else:
 print("Error en la lectura del sensor. Intentant de nou...")
 time.sleep(2.0)

 # Mostrar els resultats
 print(f"Sensor Aigua (A3): {water_value} ({water_percentage:.2f}%)")
 print(f"Sensor Humitat Terra (A0): {soil_value} ({soil_percentage:.2f}%)")

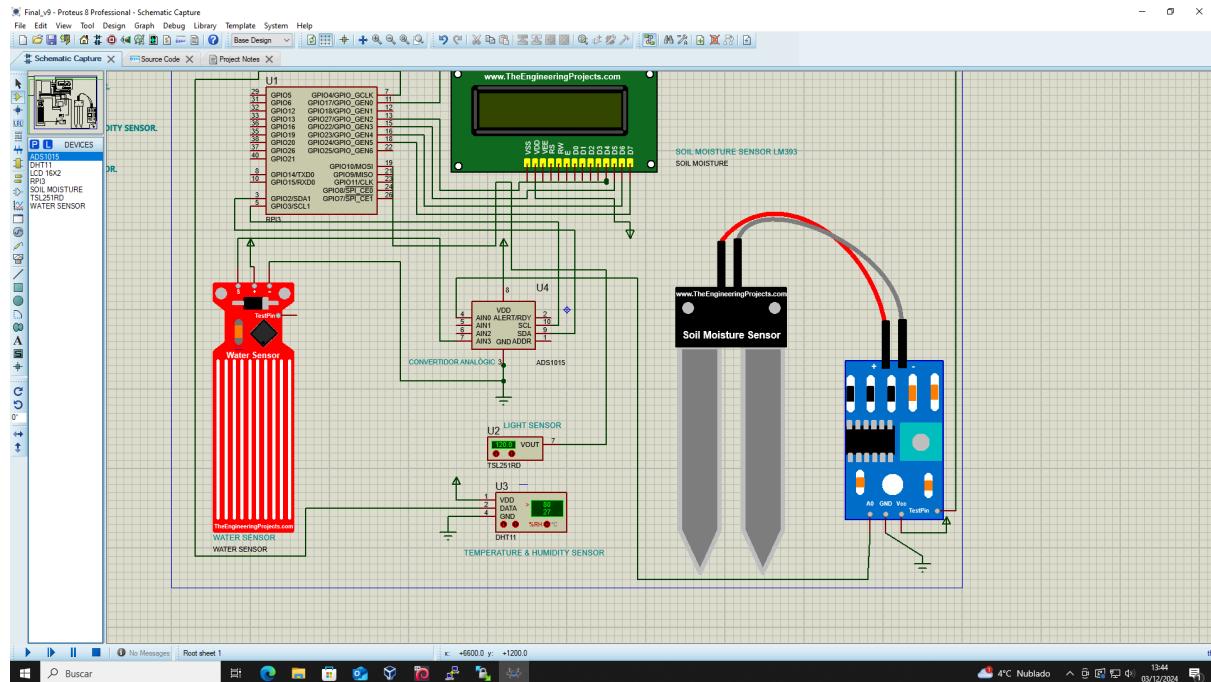
 # Pausa d'1 segon entre lectures
 time.sleep(1)

except KeyboardInterrupt:
 print("Lectura aturada per l'usuari.")

```

## 6.2. Diagrama de connexions

El Proteus 8.0 és una eina de disseny ampliament utilitzada per fer les simulacions de connexions de Raspberry Pi. A més a més, té un servei de suport molt acreditat per resoldre incidències o dubtes.



Tot seguit, es detalla els aspectes més destacats d'aquest software.

### Avantatges.

**Simulació avançada de circuits:** Proteus permet dissenyar i simular circuits abans de construir-los físicament. Això és especialment útil per validar configuracions de connexions per a projectes amb Raspberry Pi.

**Biblioteca de components extensa:** Conté una àmplia gamma de components electrònics, incloent-hi mòduls comuns com resistències, condensadors, sensors i fins i tot models genèrics de Raspberry Pi, ajudant a dissenyar un diagrama complet.

**Entorn visual intuitiu:** L'eina té una interfície gràfica fàcil d'utilitzar, cosa que facilita el procés de dibuixar el diagrama de connexions, especialment per a aquells amb menys experiència en disseny de circuits.

**Simulació de microcontroladors:** Tot i que la Raspberry Pi no és exactament un microcontrolador, Proteus permet integrar microcontroladors i mòduls perifèrics en el disseny, proporcionant una visió general del projecte.

**Compatibilitat amb altres entorns:** Proteus permet importar i utilitzar dissenys i esquemes d'altres eines, cosa que el fa útil per a integracions en projectes més complexos.

**Detecció d'errors en el disseny:** L'eina pot simular el comportament del circuit i detectar possibles errors com curts circuits o components mal connectats abans de muntar-lo físicament.

*Documentació visual professional:* Genera esquemes professionals que poden ser útils per documentar i compartir el projecte amb altres desenvolupadors o per presentar-lo en entorns educatius o comercials.

*Estalvi de temps i costos:* La simulació evita errors en la implementació física, estalvant materials i temps en la comprovació manual del circuit.

*Funcions avançades:* Proteus permet afegir elements com LEDs, pantalles i altres components interactius, cosa que és ideal per a projectes educatius o demostratius amb Raspberry Pi.

*Suport i comunitat activa:* Tot i que Proteus no és gratuït, té un gran suport en línia i una comunitat activa que proporciona exemples i tutorials per facilitar l'ús.

### **Limitacions a considerar.**

Proteus no és específic per a Raspberry Pi, per la qual cosa pot faltar compatibilitat directa amb algunes característiques específiques de la placa. També pot ser menys eficient per a dissenys molt complexos que combinin hardware i software altament personalitzat. Tot i això, segueix sent una opció excel·lent per al disseny de diagrames bàsics i mitjans.

D'altra banda, el fet que és una eina de pagament pot limitar el seu ús en les petites i mitjanes empreses. Aquests clients acostumen a preferir programari gratuït per abaratir costos.

### **6.3. Enllaços a documentació i recursos**

- a. [Raspberry Pi - Water Sensor | Raspberry Pi Tutorial](#)
- b. [How To Interface DHT 11 With Raspberry Pi | Temperature Measurement](#)
- c. [Soil Moisture Sensor \(Raspberry Pi\)](#)
- d. [LCD Display with a Raspberry Pi](#)
- e. [How to Connect ADS1115 to Raspberry Pi - Measure Analog Signals with MQ135 \(Part 1\)](#)

## **7. Referències**

Raspberry Pi a fondo para desarrolladores / Derek Molloy (Marcombo, 2019)