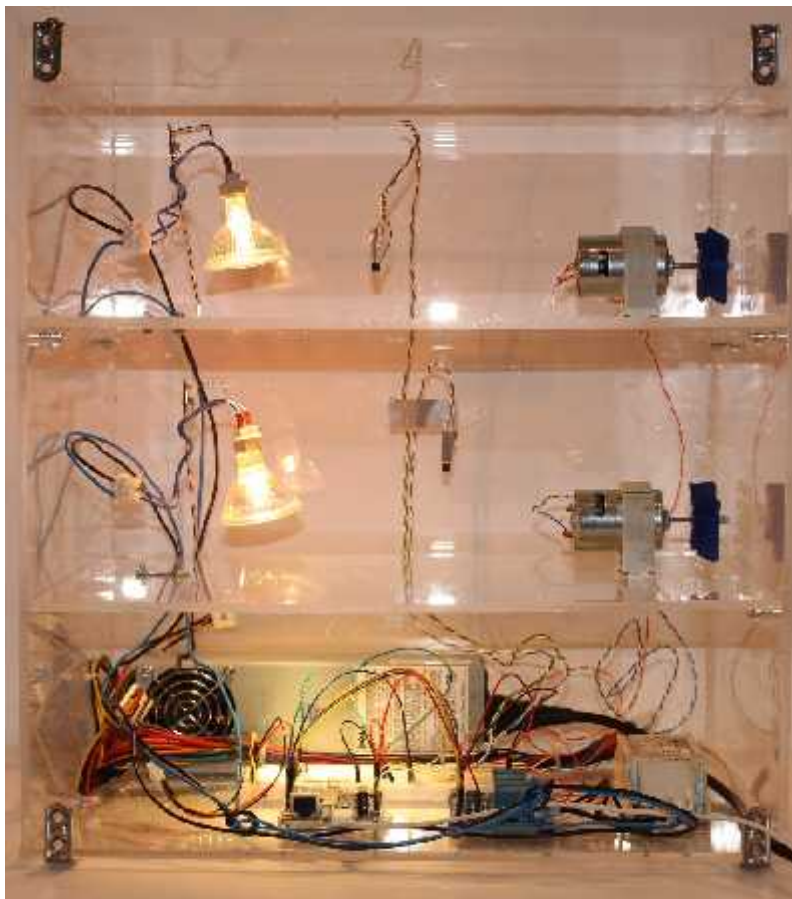


# CONTROL DE TEMPERATURA D'UN HABITATGE



Curs: 2n Batxillerat B

Tutor: Manel García

Autor: Albert Ferrer

IES Frederic Martí Carreras

Data de presentació: 9/12/2016



## ÍNDEX

Introducció.....	3
<b>MARC TEÒRIC</b> .....	4
1. QUÈ ÉS UN SISTEMA DE CALEFACCIÓ? .....	5
1.1 Classificació segons la font d'energia.....	5
1.2. Classificació dels sistemes de calefacció segons els aparells que fan servir.....	6
2. QUÈ ÉS UN CONTROLADOR LÒGIC PROGRAMABLE? TIPUS DE PLAQUES .	7
2.1. Tipus de pins .....	9
2.1.1. Pins digitals .....	9
2.1.2. Pins analògics .....	10
2.2. Què és un sensor? .....	10
2.3. Els actuadors .....	11
3. LENGUATGES DE PROGRAMACIÓ. ....	12
3.1. Com declarar una variable.....	14
3.2. Tipus de variables .....	15
3.3. Les constants .....	16
3.4. Ordres .....	16
4. PROGRAMES NECESSARIS.....	17
4.1. Sketchup .....	17
4.2. Arduino.....	17
4.3 Openscad.....	17
<b>MARC PRÀCTIC</b> .....	18
5. MATERIAL UTILITZAT PER A LA CONSTRUCCIÓ DE LA MAQUETA PER LA SIMULACIÓ DEL PROGRAMA DE CONTROL DE TEMPERATURA.....	19
6. PRESSUPOST PER A LA FABRICACIÓ DE LA MAQUETA .....	23
7. MUNTATGE PAS A PAS DE LA MAQUETA. ....	25
8. CODI DEL PROGRAMA DE CONTROL DE TEMPERATURA. ....	28
9. DISSENY DELS VENTILADORS AMB OPENSCAD. ....	33
10. PROBLEMES I SOLUCIONS AL LLARG DE LA REALITZACIÓ DEL TREBALL. ....	36
CONCLUSIÓ .....	37
AGRAÏMENTS .....	39
BIBLIOGRAFIA.....	40
BIBLIOGRAFIA FOTOGRÀFICA .....	41
ANNEX 1 .....	48
ANNEX 2.....	51
ANNEX 3 .....	54
ANNEX 4.....	56



## Introducció.

El meu treball es titula *control de temperatura d'un habitatge*, l'objectiu principal del treball és escriure un codi amb el que poder controlar el sistema de calefacció d'una casa.

En un principi tenia un treball en ment que era construir un "drone". Al final em vaig decidir per fer el control de temperatura d'una casa, atès que l'altre treball semblava més popular. A més, volia fer un projecte original i que pogués tenir una utilitat en la vida quotidiana.

Els diferents objectius que he proposat per el meu treball són els següents:

- Conèixer els tipus de PLCs <sup>1</sup> que existeixen.
- Fer un programa que controli un sistema de calefacció.
- Aplicar el programa en un habitatge real.
- Construir una maqueta per poder veure el funcionament del programa.
- Crear una aplicació per controlar la temperatura d'un immoble.

El primer que vaig fer un cop havia triat el tema del treball, va ser fer una recerca dels components més adients que havia d'utilitzar i entendre el seu funcionament per poder dur a terme el projecte. Aquesta recerca va ser llarga, ja que havia de comparar molts components, i cadascun tenia una característica que el diferenciava d'algun altre i el feia més adient. Un cop vaig tenir clar els components que havia de fer servir, els vaig comprar a través d'internet.

D'altre banda vaig estar aprenent el llenguatge de programació arduino, el qual necessitava per escriure el codi del programa. Vaig començar a elaborar diferents programes per tal de conèixer el funcionament dels relés (amb programes d'apagada i encesa de LEDs<sup>2</sup>), lectures de variables analògiques com la temperatura que captaven els sensors, però em vaig estancar un temps ja que conèixer bé un llenguatge de programació és molt difícil. Per això vaig anar a fer una consulta a un programador, perquè em donés algun consell i poder continuar amb el treball.

En resum, la finalitat del projecte era poder crear un programa que llegís la temperatura d'una habitació i accionés la font de calor o de fred que tingui l'habitatge per tal de regular a una temperatura de consigna aquesta sala.

---

<sup>1</sup>PLC: Controlador lògic programable.

<sup>2</sup>LED: Light Emitting Diode. Díode emissor de llum.



## MARC TEÒRIC



## 1. QUÈ ÉS UN SISTEMA DE CALEFACCIÓ?

Un sistema de calefacció té com a funció escalfar l'interior dels habitatges o edificis quan la temperatura és baixa. Actualment la major part de les cases en disposen d'un, tot i que també n'hi ha que fan servir el foc a la xemeneia per escalfar l'immoble.

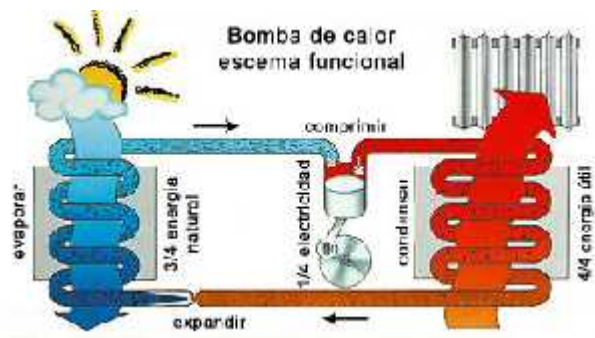
Els tipus de calefacció es poden dividir segons la font d'energia, ja sigui la biomassa, la geotèrmica, la solar, l'elèctrica o be el gas. També els podem classificar segons l'aparell o el sistema amb el qual es bescanvia l'escalfor entre l'equip i la sala, com per exemple radiadors, terra radiant, elèctric per acumuladors...

Un breu resum segons les classificacions fetes anteriorment serien:

### 1.1 Classificació segons la font d'energia

#### **Elèctrica:**

**Bomba de calor:** Aquest sistema de calefacció no produeix calor sinó que la mou d'un lloc a un altre. El seu funcionament es basa en la segona llei de la termodinàmica, segons els canvis d'estat, absorbeix o cedeix la calor. El líquid que hi ha dins el circuit, es converteix en gas de forma mecànica. El gas absorbeix la calor i mitjançant un compressor induïm al gas un canvi d'estat, cedint la seva calor per poder-la fer arribar on vulguem.



*Imatge (1) : Sistema calefacció d'una bomba de calor*

#### **Combustibles:**

**Calderes:** És un aparell que utilitza un combustible fòssil o la biomassa per escalfar l'aigua o un altre fluid que servirà per proporcionar l'escalfor que transmetrà l'aparell calefactor. (Radiador, terra radiant...)

#### **Energies renovables:**

Dins aquest apartat hi trobaríem les plaques solars que escalfen l'aigua que circula per un serpentí a la placa i acumula l'escalfor en un acumulador.



## 1.2. Classificació dels sistemes de calefacció segons els aparells que fan servir.



**Imatge (2):** Terra radiant

-Terra radiant: És un sistema de calefacció que utilitza l'aigua calenta que escalfa una xarxa de canonades dins del terra, la superfície d'aquest serà la que escalfarà l'ambient. Les xarxes de canonades podrien estar també dins les parets o inclús en el sostre, però el més lògic és que es trobin al terra. Un dels avantatges que té aquest sistema és que treballa a baixes temperatures

de l'aigua calefactors, pel que es possible la utilització de plaques solars o altres sistemes de baix consum.

-Radiadors: Els radiadors d'aigua o elèctrics són uns dispositius que tenen la funció d'emetre calor per pujar la temperatura de l'espai on estan situats. El seu funcionament és molt senzill, l'aigua calenta entra al radiador procedent d'una font de calor, i aquesta intercanvia l'energia procedent de l'aigua amb la de l'ambient, intercanvia la calor entre dos mitjans.



**Imatge (3):** Radiador d'aigua.

-Emissors termoelèctrics: Són radiadors que aconseguen la transmissió del calor a través d'oli tèrmic que s'escalfa a través d'una resistència elèctrica. Cada radiador és



**Imatge (4):** Emissor termoelèctric

independent i, no necessita ni caldera ni canonades. Els radiadors porten un termòstat i un programador. Després d'apagar els radiadors segueixen radiant calor durant hores. Acostumen a ser més cars i necessiten corrent.



## 2. QUÈ ÉS UN CONTROLADOR LÒGIC PROGRAMABLE? TIPUS DE PLAQUES

Un controlador lògic programable o PLC, és un dispositiu electrònic creat per controlar màquines i processos lògics. Aquest controlador utilitza un programa lògic que permet realitzar diferents funcions, com per exemple: recollir dades captades per un sensor connectat en una entrada analògica o digital, permet prendre decisions a partir d'uns criteris ja programats i actuar sobre dispositius externs.

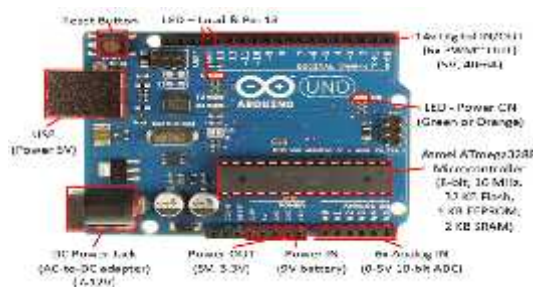
El controlador té unes entrades on es poden connectar els sensors per captar la informació del seu voltant i permet interpretar les dades captades. També té unes sortides on es poden connectar els actuadors de la màquina com per exemple uns motors. Gràcies a tot aquests components de la placa podem controlar o modificar el procés.

Els PLC serveixen per realitzar automatismes, es pot introduir un programa a la seva memòria i que el PLC executarà, es produeix un cicle d'SCAN<sup>3</sup>. Per tant hem de tenir en compte que hi ha una gran varietat de PLC, per exemple podem trobar el *siemens*, *omron*, *allen*, el *Netmedia's BX-24*, *arduino*...

Una de les marques més conegudes en aquest moment és arduino. Es una plataforma de prototips electrònics amb codi obert amb un hardware i software flexible i fàcil d'utilitzar. Arduino fa servir el llenguatge *processin* que és molt semblant al C++

Dins la marca de microcontroladors arduino hi ha diferents models de plaques.

Una de les plaques més conegudes que té arduino és la *uno*, ja que és una placa que et permet començar a introduir-te en el món de la robòtica, la defineixen com una placa amb la que començar els teus primers projectes. És una placa molt completa però alhora senzilla de fer funcionar. Disposa de 14 pins de sortida o entrada, 6 pins digitals.



**Imatge (5):** arduino Uno

<sup>3</sup>SCAN: "scan cycle" Seqüència d'operacions que realitza un autòmat de forma repetitiva.

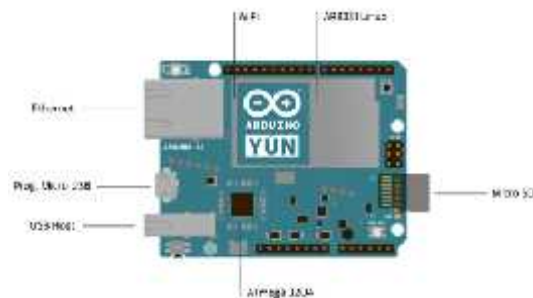


L'arduino *mega* és possiblement l'arduino més adequat per grans projectes, ja que conta amb 54 pins digitals que actuen com a entrades o sortides, 16 entrades analògiques, una connexió USB, un botó de RESET<sup>4</sup> i una entrada per alimentar la placa.



*Imatge (6): Arduino Mega*

La placa arduino *yun* és la utilitzada per fer el meu projecte, la principal característica d'aquesta placa és la capacitat que té per connectar-se a internet o a una xarxa wifi. Això és possible gràcies al port red que ve directament integrat a la placa o bé a través de wifi. La connexió amb la computadora es realitza a través d'un port micro USB, aquest micro USB també serveix per connectar la placa a una font d'alimentació de 5 volts. A part del port ethernet i l'entrada micro USB, l'arduino yun també conta amb un USB per poder connectar memòries flash, càmeres i altres dispositius per interactuar amb el processador Atheros que permet utilitzar Linux. Aquesta placa també conta amb un adaptador microSD.



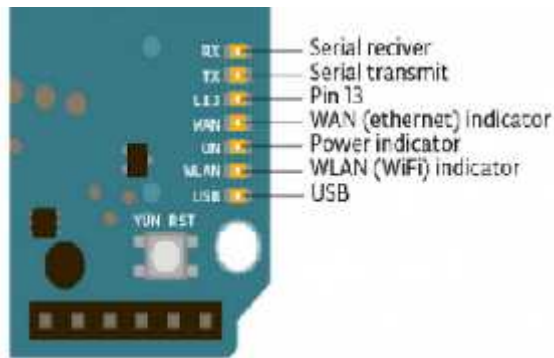
*Imatge (7): Connexions i entrades de l'arduino yun*

<sup>4</sup>RESET: Reiniciar. Inici de les condicions inicials del sistema.





L'arduino *yun* també té alguns LEDs integrats que indiquen l'estat del dispositiu en un moment determinat. La placa també porta 3 botons de RESET, un per el microcontrolador, un segon per el microprocessador i un tercer per el wifi.



*Imatge (8): LEDs integrats de la placa yun*



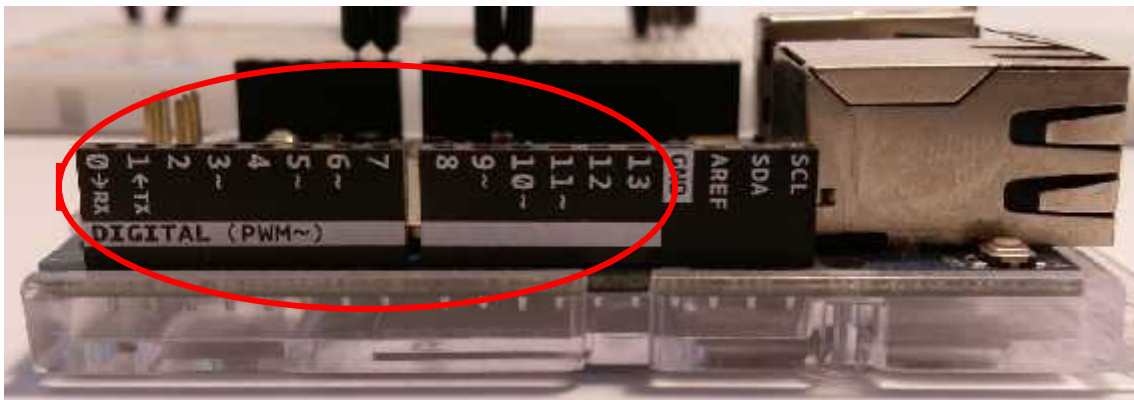
*Imatge (9): Botons reset de l'arduino yun*

## 2.1. Tipus de pins

Les plaques arduino disposen de diferents pins que poden ser d'entrada, és a dir, que poden rebre dades d'un dispositiu que estigui connectat com ara un sensor, o també disposa sortida, és a dir, que hi podem posar un aparell perquè realitzi una tasca, com per exemple hi podria ser un motor. Entre aquestes entrades i sortides s'hi diferencien dos tipus, les digitals i les analògiques.

### 2.1.1. Pins digitals

L'arduino *yun* té 13 pins digitals d'entrada o sortida. Un pin digital es caracteritza per tenir només dos valors, high (+5V), o low (+0V), així la variable associada, del programa, al pin serà 1 si el voltatge al pin és +5V, o serà 0 si el voltatge al pin és 0V.

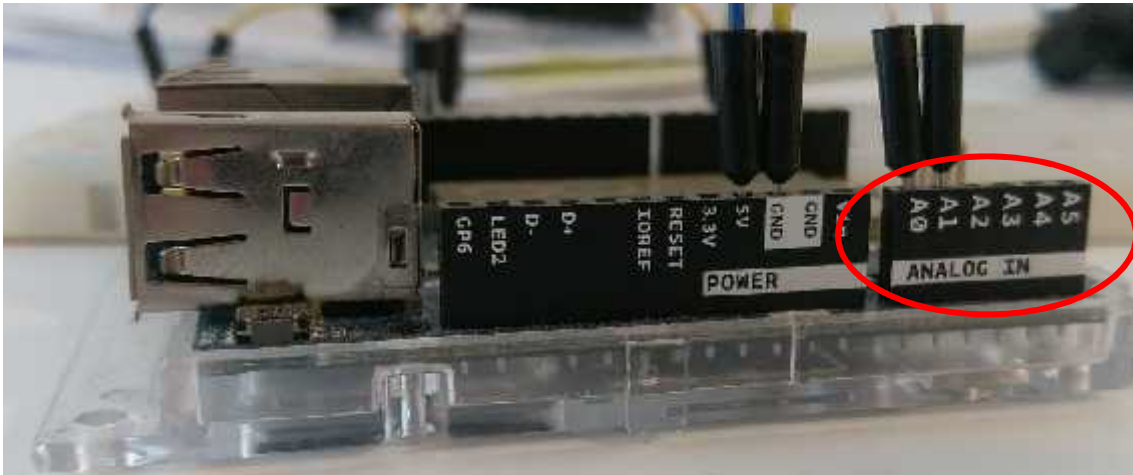


*Imatge (10): Entrades i sortides digitals de l'arduino yun.*



### 2.1.2. Pins analògics

La placa *yun* també consta de 5 pins analògics d'entrada o sortida. Les entrades reben tensions d'entre 0 i 5 volts, a diferència dels pins digitals, els analògics no s'han de declarar com entrades o sortides. En funció del voltatge a l'entrada la variable associada, del programa, tindrà un valor enter que van des del 0 fins al 1023. Els pins analògics són molt útils per connectar-hi sensors. Com és el cas d'aquest treball, on s'han utilitzat aquets pins per connectar-hi els sensors de temperatura.



*Imatge (11): Pins analògics*

## 2.2. Què és un sensor?

Un sensor és un dispositiu amb la capacitat d'interpretar magnituds tant físiques com químiques i transformar aquestes dades en variables elèctriques. Els sensors es poden classificar en analògics o digitals. Els sensors analògics són aquells que reben més informació, i el valor de voltatge que subministra és en funció de la magnitud que llegeix. En canvi els sensors digitals són aquells que donen la informació en dos estats (0 o 1).

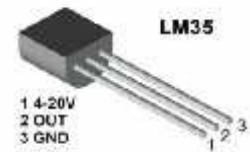
Les característiques dels sensors són els següents:

- Rang de mesura: Domini dins la magnitud de mesura que pot aplicar el sensor.
- Precisió: Error de mesura màxim esperat.
- Sensibilitat del sensor.
- Resolució: Qualitat del sensor mesurant la magnitud d'entrada.
- Rapidesa de resposta: Temps fix o dependent de la variació de la magnitud a mesurar.
- Errors esperats al repetir la mateixa mesura.

Existeixen de moltes classes: mecànics, ultrasònics, capacitius i fotoelèctrics.



En aquest treball el sensor que s'ha fet servir és un de temperatura, concretament un anomenat LM35 i el LM36. Aquest sensor envia la informació de la temperatura de casa.



**Imatge (12):** Sensor de temperatura

### 2.3. Els actuadors

Els actuadors són aparells capaços de transformar energia hidràulica, pneumàtica o elèctrica amb la finalitat d'activació d'un procés automatitzat.

Els actuadors elèctrics són els més simples de tots ja que només necessiten energia elèctrica com a font d'energia. Transformen l'energia elèctrica en mecànica, o en altres tipus com a calorífica (una resistència, una bombeta d'incandescència).

En el treball utilitzarem un motor per tal de refrigerar l'habitacle (construcció d'un ventilador). Podem trobar tres tipus de motors: els de corrent continua, els de corrent altern i el motors de pas a pas.

Els avantatges que tenen els motors elèctrics és que són fiables i precisos, silenciosos, fàcils de controlar i d'instal·lar. Un dels seus desavantatges és que tenen una potència limitada.



**Imatge (13):** Motor corrent



### 3. LENGUATGES DE PROGRAMACIÓ.

Els autòmats pretenen controlar un determinat procés repetitiu, és a dir, executar una sèrie d'accions mitjançant uns actuadors, per exemple encendre i apagar un llum o fer un moviment mitjançant un motor... en funció d'unes determinades condicions, per exemple una determinada temperatura, o un desplaçament d'una peça. Els autòmats sorgeixen principalment de la indústria de l'automoció substituint la mà d'obra humana en tasques repetitives i/o feixugues. En l'autòmat s'introdueix un programa amb les instruccions que volem que realitzi la màquina i resoldrà o executarà la feina en l'ordre indicat. Si introduïm un altre programa el mateix autòmat, la màquina resoldrà o executarà la feina de manera diferent.

Així doncs, quan volem realitzar una tasca amb un autòmat cal seguir dues etapes: la primera introduir les instruccions a la memòria; el programa haurà d'estar escrit en un llenguatge de programació que l'autòmat reconegui. A la segona fase, haurem de demanar a l'autòmat que executi el programa, el qual farà que la màquina es comporti com una persona executant una determinada feina, ja sigui de control, com d'execució.

Per tal de preparar un programa, cal seguir també dues etapes: la primera d'anàlisi, d'on partirem d'un problema o tasca que cal fer i obtindrem un algorisme, que és una descripció, en el nostre llenguatge, molt precisa dels passos a realitzar per resoldre el problema. Només es pot dissenyar un algorisme si es coneix la manera de resoldre el problema. A la segona etapa haurem de traduir l'algorisme a un programa que s'introduirà en el PLC. Aquesta fase és molt més senzilla que la primera, i consisteix únicament en la traducció del llenguatge algorísmic al llenguatge de programació.

La virtut dels programadors és el coneixement exhaustiu d'aquest llenguatge, les funcions predeterminades que existeixen en el mateix autòmat, les possibilitats de càlcul per tal de maximitzar el rendiment d'aquest, fer-lo més ràpid i precís.

Un llenguatge de programació és un llenguatge informàtic el qual s'utilitza per controlar un dispositiu electrònic. Cada llenguatge té unes regles semàntiques i sintàctiques per escriure un programa. El codi té unes regles que permeten especificar tant les classe de dades amb que ha de treballar el programa com les funcions que a d'executar el programa.

Alguns programes estan definits per un document i d'altres estan definits per un compilador.



Un compilador és un programa informàtic que tradueix el llenguatge escrit de programació en un altre llenguatge equivalent perquè la màquina sigui capaç d'interpretar-lo. Permet traduir el codi font d'un programa en un llenguatge d'alt nivell a un llenguatge de nivell més baix. Així d'aquesta forma un programador pot escriure un programa d'una forma més entenedora per l'ésser humà per després compilar-lo i fer-lo un programa adequat per a un ordinador.

En el cas del meu treball he utilitzat alguns programes que utilitzen diferents llenguatges de programació. Un d'ells era el programa *ARDUINO* i l'altre és l' *OPENCAD*

El programa arduino és el que he utilitzat més ja que és en el que centre el meu treball, aquest programa utilitza el llenguatge C que és un llenguatge de mig nivell però amb algunes característiques d'un d'alt nivell i d'altres de baix nivell. L'estructura del llenguatge és força simple i està composta per tres parts. Aquestes parts comprenen la declaració de les variables, el bloc de declaracions també anomenat *setup* i el de les instruccions o *loop*. Aquestes dos últimes parts en el programa arduino són obligatòries, per tant no serà possible escriure un algoritme sense aquestes funcions.

Al principi del codi hi trobem les variables, que en programació són espais reservats a la memòria que poden canviar de contingut al llarg de l'execució d'un programa. Poden ser de longitud fixa o variable. Les variables de longitud fixa són aquelles que el tamany no varia al llarg del funcionament del programa. En canvi les de longitud variable poden canviar la mida al llarg del funcionament del programa, normalment ho són aquelles que han d'emmagatzemar dades.

En segon lloc en el programa arduino hi trobem el *setup* és la part del programa on s'estableixen les funcions que s'han d'executar un únic cop. Com per exemple seria on definiríem el pin en que estar connectat qualsevol component i també definiríem si és una entrada o bé una sortida. Per exemple si es un component com ara un sensor el definiríem com una entrada ja que captaria alguna dada i ens l'enviaria al controlador, en canvi si fos un LED el definiríem com una sortida ja que no ens ha de donar cap informació sinó que seria qui rebria la funció que ha executat la placa. Dins d'aquesta funció també hi podríem definir que començaríem una comunicació amb un serial.

En tercer lloc hi trobem el *loop* que és el lloc on s'executen les accions del programa. Aquesta part del programa s'executa un cop i un altre, quan acaba el programa torna a començar. El programa arduino executa línia a línia.

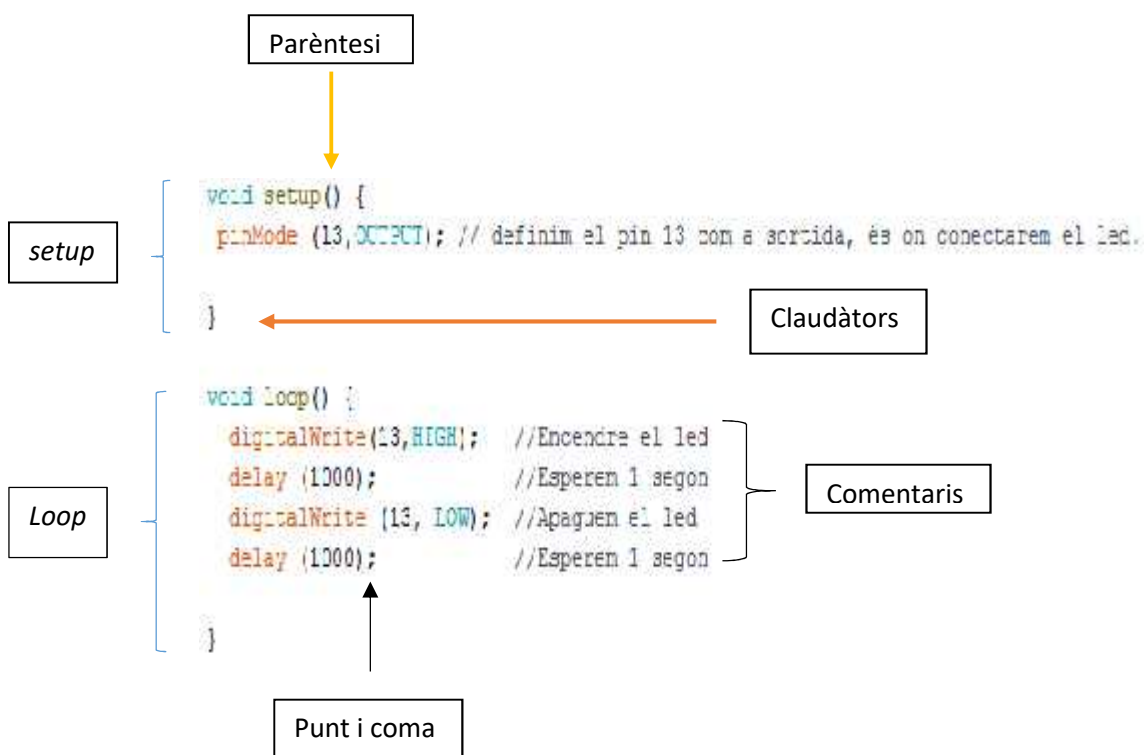
Dins el programa hi han un seguit marques de puntuacions que serveixen per definir el començament i el final d'algunes parts del codi. Hi ha dos tipus de marques en el codi



arduino. Els primers són els parèntesis ( ), aquets s'utilitzen per escriure funcions dins el nostre programa, també s'utilitzen per intercanviar alguna variable dins el programa. En alguns casos com en el del *setup* o en el *loop* s'hi posen un parèntesis buits perquè el programa no tingui un error a l'hora de compilar.

En segon lloc hi trobem les claus { }. Aquestes claus s'utilitzen per indicar el principi i el final d'una funció, sense aquestes clau el programa no seria capaç de saber on comença i on acaba la funció i què es considera la següent part del codi.

Un altre element molt important són els punts i comeses ;. S'utilitzen per separar les diferents línies de codi i indicar a l'arduino on acaba el seu comando.



**Imatge (14):** Programa arduino, codi de programació

### 3.1. Com declarar una variable

Una variable per declarar-la ha de ser d'un tipus, ha de tenir un nom i un valor. Per exemple podria ser:

```
int sensor = 1
```

Hi ha dos maneres de declarar una variable. Si la declarem al principi del programa, abans del *void setup*, serà una variable global, la qual és accessible des de tot el



programa. En canvi si la variable la declarem dins una funció serà una variable local, la qual només hi podrem accedir dins la funció on ha estat declarada.

```
int ledPin = 13;
void setup() {
  digitalWrite(ledPin, OUTPUT);
  digitalWrite(ledPin, HIGH);
  delay(1000);
  digitalWrite(ledPin, LOW);
  delay(1000);
}
```

*Imatge (15): Variable declarada al principi del programa.*

```
void setup() {
  int ledPin = 13;
  digitalWrite(ledPin, OUTPUT);
}
void loop() {
  digitalWrite(ledPin, HIGH);
  delay(1000);
  digitalWrite(ledPin, LOW);
  delay(1000);
}
```

*Imatge (16): Variable declarada dins el programa*

### 3.2. Tipus de variables

**-int:** És l'abreviatura del número enter, és el tipus de variable més comuna d'arduino. Aquesta variable es fa servir per declarar variables enters sense decimals. Guarda un valor de 16 bits, això vol dir que ocupa 16 bits de la memòria d'arduino, que pot guardar un número enter entre el -32767 i el 32767.

**-long:** És una variable com la int però que ens permet guardar una variable més gran només de números enters. Aquesta variable té un valor de 32 bits, això vol dir que ocuparà 32 bit de la memòria de l'arduino que pot guardar variables entre els números -2147483647 i el 2147483647.

**-byte:** És una variable que ens permet estalviar espai a la memòria de l'arduino. És un número enter de 8 bits, així que només ens ocuparà 8 bits de memòria i ens permetrà guardar una variable dels valors del 0 al 255.

**-float:** És la única variable que ens permet guardar números decimals, es guarden en un valor de 32 bits. Aquesta variable ocupa molt més espai que les altres variables que només guarden números enters. També és més lent al fer els càlculs ja que triga més temps en fer tots els càlculs.

**-arrays (matrius):** Aquesta variable ens permet guardar una col·lecció de valors. Tots els valors guardats dins la matriu es guarden amb un número índex, aquest número permet estudiar la variació d'una magnitud. Aquestes variables es declara de la mateixa forma que les altres, amb un nom, un tipus i un valor.



### 3.3. Les constants

Les constants són paraules que fa servir arduino i que tenen uns valors preestablerts.

Unes altres constants indispensables són les que definim un pin com una entrada *input* o una sortida *output*.

```
void setup() {
  pinMode (13,OUTPUT); // definim el pin 13 com a sortida, és on connectarem el led.
}
```

```
void loop() {
  digitalWrite(13,HIGH); //Encendre el led
```

*Imatge (17): Constants (output i high)*

Unes constants molt utilitzades son les de *high* (*on*) y *low* (*off*). Aquestes constants s'utilitzen per determinar l'estat d'un pin que només té dos estats. L'estat d'apagat, es a dir que hi ha 0 volts, o l'estat d'engegat, es a dir que l'arduino envia 5 volts en el pin.

### 3.4. Ordres

- *If*: Aquesta ordre traduïda al català és un *si* i serveix per donar una condició. Per exemple si tenim una temperatura massa baixa i volem que es posi en marxa la calefacció, li diríem a l'arduino:

```
If (Temperatura < 19°C) {
    Resistència1 , HIGH ;
}
```

- *for*: Aquesta ordre es pot traduir al català com si fos un *per* o un *durant*. Aquesta funció es pot utilitzar complementant-la amb un *if*.

Per exemple si volem que per una temperatura que estigui entre mig de la més baixa i la més alta la placa no faci funcionar res diríem:

```
For (Temperatura > 19°C i Temperatura <23) {
    Resistència1, LOW;
    Ventilador1,LOW;
}
```

Aquestes ordres són les utilitzades en el codi del programa que he escrit per controlar la temperatura. El programa arduino, en té d'altres, però aquestes dos són les més utilitzades en els programes.





## 4. PROGRAMES NECESSARIS

### 4.1 Sketchup

El primer per començar el treball ha sigut ajuntar els plànols de la casa (maqueta) amb un model de tres dimensions, per aquest motiu s'ha utilitzat el programa sketchup. Es tracta d'un programa molt senzill d'utilitzar i molt intuïtiu.

Aquest programa es pot descarregar des de la seva pàgina oficial on es poden trobar vídeos tutorials, on s'ensenya el programa d'una manera bàsica.



**Imatge (18):**  
Logotip sketchup.

### 4.2 Arduino

El segon programa utilitzat és l'arduino, un programa per poder escriure el codi de programació per el control de temperatura i poder-lo instal·lar a la placa. Utilitza un codi de programació amb el llenguatge C.

Aquest programa es troba a la pàgina oficial d'arduino on també hi podem trobar tutorials de com programar i ajuda en el moment en que tinguem problemes.

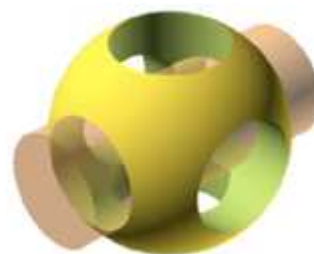


**Imatge (19):** Logotip Arduino.

### 4.3 Openscad

El tercer i últim programa que s'ha utilitzat pel treball ha estat l'openscad. És tracta d'un programa de disseny en tres dimensions. He utilitzat aquest programa per poder crear un ventilador que es pugui imprimir en 3d, per utilitzar-lo a la maqueta, particularment estava molt més familiaritzat amb aquest programa i per fer un disseny complicat com és el d'un ventilador ha estat molt útil.

El programa no és tant intuïtiu com l'sketchup, ja que fa servir una mena de codi de programació per poder crear una peça.



**Imatge (20):** Logotip openscad.



## MARC PRÀCTIC



## 5. MATERIAL UTILITZAT PER A LA CONSTRUCCIÓ DE LA MAQUETA PER LA SIMULACIÓ DEL PROGRAMA DE CONTROL DE TEMPERATURA.

### La placa:

D'acord amb el projecte portat a terme, la placa *arduino yun*, és una placa senzilla de programar, econòmica i disposa d'un sistema wifi per comunicar-se. La placa amb un sensor de temperatura podria ser un termòstat de calefacció intel·ligent.

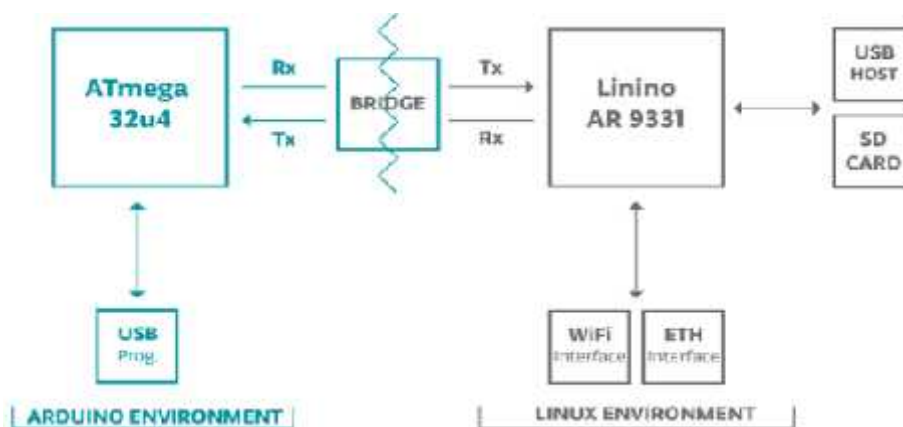
Característiques de la placa:

- ) Mides de la placa: 7'8 X 5'6 cm.
- ) Disposa de 5 pins analògics i de 13 pins digitals.
- ) Consta d'un port per una targeta de memòria *microSD*.



**Imatge (21):** Placa arduino yun

Aquesta placa també es caracteritza per combinar dos sistemes operatius, el propi d'arduino i un sistema operatiu LINUX. Un xip conté tot el sistema arduino, i un altre el sistema operatiu LINUX, basat en *OpenWRT* anomenada *linino*. El dispositiu es comunica amb tots dos alhora. És molt similar a un router wifi, per una banda té tota la connectivitat pròpia d'un arduino i per altre banda conta amb linino que pot executar funcions en l'apartat LINUX mentre utilitzem la seva connectivitat wifi.



**Imatge (22):** Explicació sistemes operatius de l'arduino yun

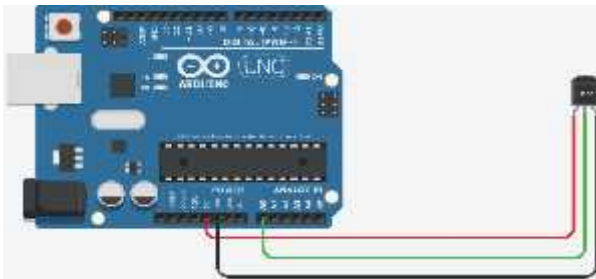


**Sensors de temperatura:**

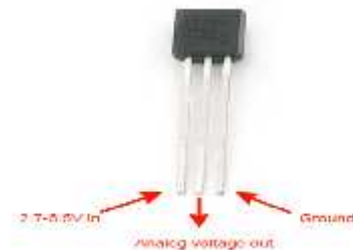
Un sensor de temperatura és un xip que ens dona una tensió de sortida en funció de la temperatura que capta. En la maqueta del projecte els sensors de temperatura utilitzats han sigut LM35 i LM36.

Els dos sensors mesuren la temperatura en graus centígrads, els dos poden mesurar temperatures que oscil·len entre els -55°C fins als 155°C. Aquets sensors poden tenir un marge d'error de  $\pm 1^\circ\text{C}$ .

El sensor té tres potes on cadascuna va connectada en llocs diferents. La pota esquerra va connectada a la corrent amb un voltatge al voltant de 5V. La pota del mig subministra la tensió en funció de la temperatura que llegirà la placa, l'interpretarà i ens donarà un valor numèric . I per últim la pota de la dreta va connectada a terra, és a dir, a 0V.



*Imatge (23): Connexió del sensor a la placa*

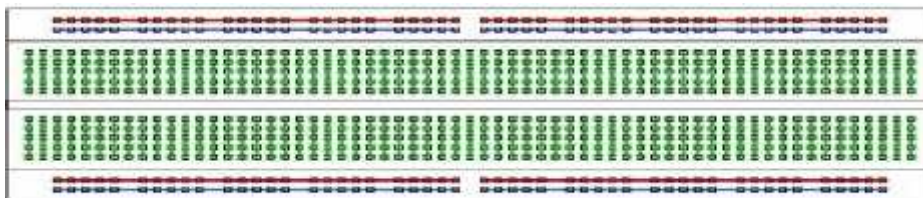


*Imatge (24): Sensor temperatura*

Per obtenir la temperatura, en el programa s'han de fer uns factors de conversió, ja que el sensor envia les dades captades amb milivolts i s'han de passar a graus centígrads.

**Protoboard:**

Una protoboard o també coneguda com a placa de proves, és utilitzada per construir prototips de circuits electrònics.



*Imatge (25): Protoboard*



**Cables:**

Per el projecte han estat utilitzat cables amb diferents formes, els cables dupont, que són cables mascle femella que serveixen per connectar motors a les plaques i components en els relés.



*Imatge (26): Cables dupont*

Cables mascle, mascle, per connectar els components a la protoboard i a l'arduino.



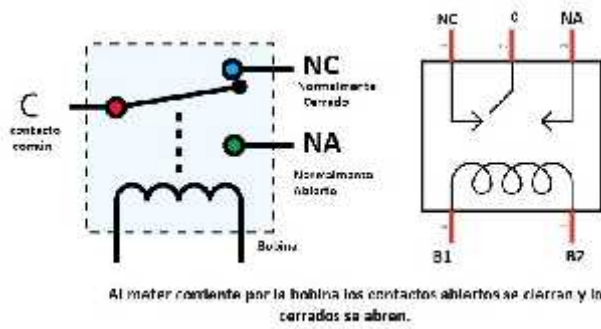
*Imatge (27): Cables mascle-mascle*

**Relé:**

Un relé ens permet modificar l'estat d'un commutador gràcies a l'electricitat. Està accionat per un electroimant que obre o tanca un o diversos contactes. Una de les principals característiques es que permet controlar voltatge o intensitat molt superiors a les que entren.



*Imatge (28): Relé*



*Imatge (29): Funcionament del relé*

**Motors:**

Un motor de corrent continua és un aparell que converteix energia elèctrica en energia mecànica. En aquest cas els motors han sigut utilitzats per fer uns ventiladors amb la finalitat de reduir la temperatura a la maqueta de la casa.



*Imatge (30): Motor de corrent continua*



**Metacrilat:**

És un plàstic també conegut amb el nom d'acrílic. És un material resistent, transparent i resistent a les condicions exteriors.

S'ha utilitzat per construir la maqueta per poder veure el funcionament del programa de control de temperatura. Simular les diferents sales d'una casa amb diferents condicions de temperatura



**Imatge (31):**  
Metacrilat

**Cable USB – MicroUSB:**

Cable per connectar la placa a l'ordinador.



**Imatge (32):** Cable USB-  
microUSB

**Transformador**

És un aparell que permet augmentar o disminuir la tensió en un circuit elèctric mantenint la potència.



**Imatge (33):** Transformador

**Font d'alimentació**

És un aparell que transforma el voltatge de 220v a 12v normalment, i que passa de corrent alterna a corrent continu. En el mercat existeixen diferents fonts per diferents voltatges d'entrada i sortida



**Imatge (34):** Font  
d'alimentació



## 6. PRESSUPOST PER A LA FABRICACIÓ DE LA MAQUETA

Aquesta taula mostra el material utilitzat i el cost per a la construcció de la maqueta.

Concepte	Lloc de compra o URL	Unitats	Preu	Total
<b>Placa arduino yun</b>	<a href="http://www.microlog.net/">http://www.microlog.net/</a>	1	76,25€	76,25€
<b>Sensor de temperatura LM35</b>	<a href="http://www.microlog.net/">http://www.microlog.net/</a>	3	2,65€	7,95€
<b>Motor dc 12v.</b>	<a href="https://www.amazon.es/">https://www.amazon.es/</a>	2	4,93€	9,86€
<b>Pack cables arduino</b>	<a href="https://www.amazon.es/">https://www.amazon.es/</a>	1	1,35€	1,35€
<b>Relé</b>	<a href="https://www.amazon.es/">https://www.amazon.es/</a>	2	2,68€	5,36€
<b>Pack LEDs</b>	<a href="http://tienda.bricogeek.com/">http://tienda.bricogeek.com/</a>	1	3,90€	3,90€
<b>Pack cables M/M</b>	<a href="http://tienda.bricogeek.com/">http://tienda.bricogeek.com/</a>	1	2,50€	2,50€
<b>Pack cables M/H</b>	<a href="http://tienda.bricogeek.com/">http://tienda.bricogeek.com/</a>	1	2,50€	2,50€
<b>Pack cables H/H</b>	<a href="http://tienda.bricogeek.com/">http://tienda.bricogeek.com/</a>	1	2,50€	2,50€
<b>resistències 1/4W</b>	<a href="http://tienda.bricogeek.com/">http://tienda.bricogeek.com/</a>	1	11,50€	11,50€

CONTROL DE TEMPERATURA D'UN HABITATGE



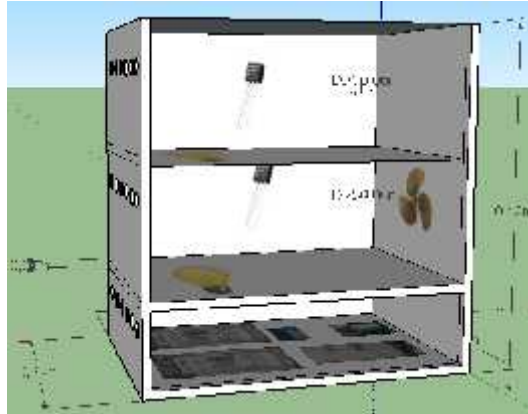
<b>Sensor de temperatura LM36</b>	<a href="http://tienda.bricogeek.com/">http://tienda.bricogeek.com/</a>	5	1,90€	9,50€
<b>Protoboard i mòdul adaptador d'alimentació</b>	<a href="https://www.amazon.es/">https://www.amazon.es/</a>	1	10,99€	10,99€
<b>transformadors</b>	Ferreteria	1	4,99€	4,99€
<b>Metacrilat</b>	Ferreteria	1m <sup>2</sup>	60€	60€
<b>Cargols</b>	Ferreteria	20	0,07€	1,40€
<b>Volanderes</b>	Ferreteria	20	0,041€	0,82€
<b>Femella</b>	Ferreteria	20	0,041€	0,82€
<b>Escaires</b>	Ferreteria	4	0,65€	2,60€
<b>Adhesiu</b>	Ferreteria	1	3,29€	3,29€
<b>TOTAL</b>				218,08€





## 7. MUNTATGE PAS A PAS DE LA MAQUETA.

**Pas 1:** El primer pas per poder muntar la maqueta és dissenyar, en aquest cas amb l'sketchup, un model en tres dimensions per poder veure com seria la maqueta, les mides que hauria de tenir i si s'hi podrien posar tots els elements necessaris per el funcionament.



**Imatge (35):** Disseny de la maqueta amb sketchup

**Pas 2:** El següent pas va ser comprar les planxes de metacrilat tallades a mida i enganxar-les amb una cola espacial. S'ha escollit el metacrilat per fer el treball per ser transparent i poder veure el funcionament, així com la complexitat del circuit elèctric.



**Imatge (36):** Plaques de metacrilat tallades.



**Imatge (37):** Muntatge de les parets i la base.



**Imatge (38):** Muntatge del sostre.



**Imatge (39):** Muntatge de les plantes.



**Pas 3:** El tercer pas va ser fer forats a les parets laterals per on el ventilador expulsa l'aire calent i refrigera la sala.



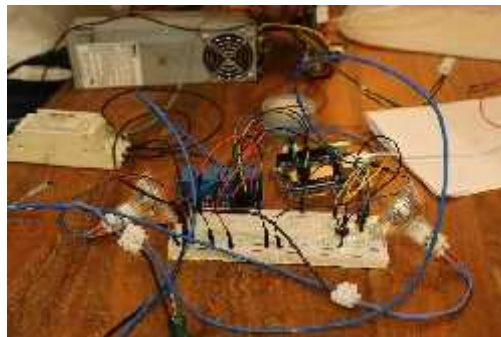
**Imatge (40):** Forats a la paret lateral esquerra



**Imatge (41):** Forats a la paret lateral dreta

**Pas 4:** El quart pas va ser connectar tots el components per veure si el programa funcionava correctament, per després poder-los instal·lar dins la maqueta.

El muntatge consta de la placa arduino *yun*, una placa de relés, un transformador de corrent , una font d'alimentació, dos bombetes i dos sensors de temperatura.

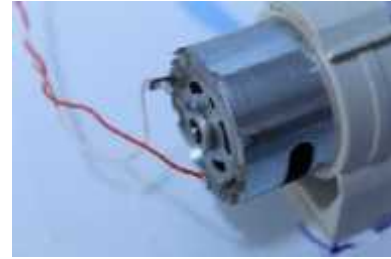


**Imatge (42):** Connexió provisional dels components.

**Pas 5:** El següent va ser connectar definitivament tots els components d'una forma que no es poguessin desconectar ni fer mal contacte, un cop instal·lats dins la maqueta. Per tal d'assegurar les connexions es van soldar els sensors de temperatura i els motors amb els cables corresponents. Es van enganxar i fixar els cables amb silicona per tal de no desconectar-se ni fer mal contacte. Per últim es va posar uns suports per aguantar els motors.



**Imatge (43):** Soldadura del sensor de temperatura



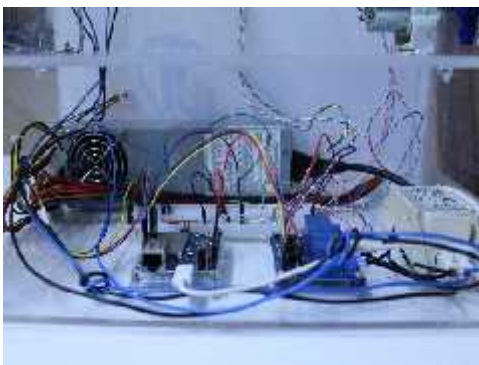
**Imatge (44):** Soldadura del motor

**Pas 6:** El sisè pas consistia en col·locar uns escaires de ferro al terra i al sostre de la maqueta, per poder-hi posar la paret extraïble i així accedir als components de la maqueta.



**Imatge (45):** Escaires per aguantar la paret abatible

**Pas 7:** L'últim pas consistia en col·locar tots els elements a la maqueta. A la planta baixa primer de tot s'hi vaig fixar la font d'alimentació, el transformador de corrent, la placa arduino, els relés i la protoboard. Tots els elements es van fixar amb cargols o amb un adhesiu. Al primer i segon pis es van posar els sensors de temperatura, les bombetes i els ventiladors.



**Imatge (46):** Planta baixa



**Imatge (47):** Primera i segona planta



## 8. CODI DEL PROGRAMA DE CONTROL DE TEMPERATURA.

El codi del programa del control de temperatura està escrit amb llenguatge C perquè la placa el pugui comprendre.

En primer lloc es va buscar com funcionaven els sensors de temperatura i com s'havien de programar. Els sensors de temperatura proporciona un voltatge en funció de la temperatura, el programa instal·lat a l'arduino interpreta aquest voltatge mitjançant una funció que s'ha programat, i ens subministra un valor real de temperatura en graus centígrads. Aquest primer programa era molt senzill i l'únic que feia era mostrar la temperatura que captaven els sensors en graus centígrads i els mostrava a la pantalla de l'ordinador.

Al principi del programa declarem una variable i on està connectat el sensor.

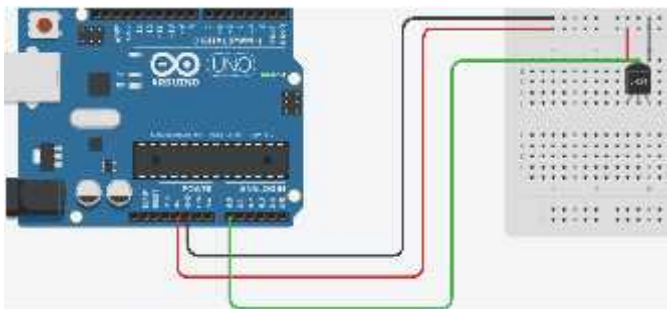
A la funció del *void setup* que només s'executa al iniciar el programa, iniciem una comunicació amb el serial.

A la funció del *void loop* que segueix un bucle, és a dir, un cop ha acabat de fer-ho tot, es torna a repetir. Aquí hi trobem que la placa llegeix els valors que envia el sensor de temperatura i un cop els ha rebut fa servir el factor de conversió per poder transformar la magnitud donada a graus centígrads. Un cop té la temperatura en graus, fa que es pugui llegir en el serial.

```

1 int LM35_PIN = 0; //definem el pin on està connectat el sensor
2
3 void setup() {
4   Serial.begin(9600); //Iniciem la comunicació amb el serial
5 }
6
7 void loop() {
8   int input = analogRead(LM35_PIN); // llegim el valor que ens dona el sensor LM35
9   float mv = (5000 / 1024.0) * input; // converteixo el valor llegit a mV
10  float cel = mv / 10; // converteixo els mV llegits a graus celsius
11
12  Serial.println(val); // Fem que es mostri la temperatura en el serial
13  delay(1000); // Esperem 1 segon a tornar a iniciar el programa
14 }
    
```

**Imatge (48):** Primer codi per llegir la temperatura del sensor.



**Imatge (49):** Esquema del circuit a la pàgina web [circuits.io](http://circuits.io)



**Imatge (50):** Monitor serial. Simulació de lectura de la temperatura



En segon lloc el que vaig fer va ser escriure un programa molt més complex i que fos molt més funcional. Per exemple una cosa que vaig fer va ser declarar una variable on posaria les quantitats de sales que hi vulgues i dins declarar els sensor, les temperatures màximes i mínimes de cada sala, les resistències que farien la simulació del sistema de calefacció per terra radiant i els ventiladors que faries la simulació d'un aire a condicionat o bé una corrent d'aire que refrigerés la casa. Totes aquestes funcions estan fetes i organitzades per si volguéssim afegir o treure sales del programa només modificant un número en comptes d'haver d'escriure un altre cop tot el programa per cada sala o bé modificar gran part del codi per un petit canvi.

En aquest programa també hi trobem unes resistències i uns ventiladors per poder simular com funcionaria el programa en una maqueta.

A cada sala hi ha un sensor que marca la temperatura de la sala, cada una disposa d'una bombeta al·lògena que fa la funció de terra radiant, i un ventilador per refrigerar la sala.

Al començament del programa hi declarem les variables que farem servir per escriure el codi

La primera variable és per definir les quantitats de sales.

Les dos següents són les variables amb valors decimals per definir les temperatures màximes i mínimes de les sales més endavant en el programa.

Les altres dos variables que venen a continuació, la 4 i la 5, són per definir les resistències i els ventiladors com a vectors.

L'última variables és la temperatura llegida a la sala.

```
int const QTT_SALES = 2;           //quantitat de sales que hi han
float tempMin[QTT_SALES];         //variable per la temperatura minima
float tempMax[QTT_SALES];         //variable per la temperatura maxima

int Resistencia[QTT_SALES];       //definim les variables com a vectors
int Ventiladors[QTT_SALES];       //definim variables com a vectors

float temp = 0;                   //temp=Temperatura llegida a la sala
```

**Imatge (51):** Declaració de les variables al principi del programa.



La segona part de la programació és on hi trobem el *void setup*, la part que només s'executa a l'inici del programa.

Iniciem la comunicació amb el serial i definim el voltatge de referència de les senyals analògiques.

Declarem els ventiladors i indiquem cadascun a quin pin està connectat, per exemple el ventilador 0 està connectat al pin 2 i correspon a la sala 0. El ventilador 1 està connectat al pin 3 i correspon a la sala 1.

Seguidament declarem les resistències igual que hem fet amb els ventiladors. La resistència 0 està connectada al pin 4 i correspon a la sala 0 i la resistència 1 està connectada al pin 5 i correspon a la sala 1.

Per una altre part escrivim una ordre per poder definir la temperatura mínima i màxima que volem a sala. Aquestes temperatures que hem definit seran les mateixes a les dos sales. Dintre aquesta ordre li diem a la placa que totes les variables on trobi ventilador o resistència seran sortides.

Per últim si volem que a les sales tinguin temperatures mínimes i màximes diferents, escriurem les variables de temperatura mínima i màxima.

```
void setup() {

  Serial.begin (9600);
  analogReference(EXTERNAL); //per definir el voltatge de referencia senyals analogiques
  Ventiladors[0] = 2; //ventilador connectat al pin 2 sala 0
  Ventiladors[1] = 3; //ventilador connectat al pin 3 sala 1
  Resistencia[0] = 4; //resistencia connectat al pin 4 sala 0
  Resistencia[1] = 5; //resistencia connectat al pin 5 sala 1 //si el led del rele esta engegat = resistencia parada

  for (int i = 0; i < QT_SALES; i++) {
    tempMin[i] = 18;
    tempMax[i] = 20;
    pinMode( Ventiladors[i], OUTPUT ); //definim els ventiladors com a sortides
    pinMode( Resistencia[i], OUTPUT ); //definim les resistencias com a sortides
  }

  //TEST
  tempMin[1] = 28; //prova per tal sala 2 tingui diferent temperatura
  tempMax[1] = 30;
}
```

**Imatge (52):** Ordres del *void setup* per explicar el codi.



La tercera part en que es divideix el programa per controlar la temperatura és la del *void loop*, és la part del programa que es repeteix sempre un cop i altre d'haver-se executat.

Aquesta és la part del programa on li donem la majoria de les ordres.

Primer de tot li diem que si la temperatura de la sala que capta el sensor és més baixa de la que hem definit prèviament, la resistència s'haurà d'engegar per escalfar la sala, però que el ventilador haurà d'estar parat. També definim un paràmetre perquè ens mostri a la pantalla si una resistència es posa en marxa ho mostri amb un missatge que posi *R ON* a la resistència que fa funcionar.

Seguidament li donem l'ordre que, si al llegir la temperatura de les sales la temperatura està dins dels valors que hem definit, apagui els ventiladors i les resistències de la sala. I també li fem mostrar el missatge de *OFF R&V*

Més endavant li definim que si el sensor de temperatura capta una temperatura superior a la definida prèviament, el què haurà de fer la placa serà posar en marxa el ventilador de la sala. I li farem mostrar al serial que el ventilador està engegat amb el missatge: *Von*.

```
void loop () {
  for (int i = 0; i < QT_SALES; i++) {
    temp = mestraTemperatura(i);
    if (temp <= tempMin[i]) {
      //si la temperatura és més baixa que els valors declarats la resistència s'encen
      Serial.println(" R ON Sala:" + String(i) );
      digitalWrite(Resistencia[i], HIGH);
      digitalWrite(Ventiladors[i], LOW);
    }
    if (tempMin[i] < temp & temp < tempMax[i]) { //si la temperatura està entre els dos valors declarats el ventilador i la resistència estaran parats
      Serial.println(" OFF R&V Sala:" + String(i) );
      digitalWrite(Resistencia[i], LOW);
      digitalWrite(Ventiladors[i], LOW);
    }
    if (tempMax[i] <= temp) { //si la temperatura és més alta que el valor mínim declarat s'encen el ventilador
      Serial.println(" V ON Sala:" + String(i) );
      digitalWrite(Ventiladors[i], HIGH);
      digitalWrite(Resistencia[i], LOW);
    }
  }
}
```

**Imatge (53):** Ordres dels *void loop* per explicar el programa.



Seguidament donarem unes ordres per tal de mostrar en el monitor serial una separació entre les dades que prèviament hem fet mostrar i separar les dades captades pel sensors de temperatura cada 5 segons.

Amb la ordre de *serial.print* podem fer que el programa mostri el que vulguem en el serial, en aquest cas farà mostrar-hi els estats dels ventiladors i les resistències, la temperatura que detecta el sensor, al costat de la temperatura farem que hi escrigui *grados celsius*

Seguidament donarem una ordre de pausa de 5 segons i que ens escrigui una sèrie de guions seguits per separar les dades que ens va mostrant.

```

Serial.print("Analogics ");
Serial.print( i );
Serial.print( " ");
Serial.print(temp);
Serial.print( " grados Celsius"); // escriu graus celsius al costat de la temperatura
}
delay (5000); //esperar 5 segons per repetir el codi
Serial.print ("-----\n"); //guions de separació entre cada lectura en el monitor sèrie
}

float mostraTemperatura(int analog) { //mostra les temperatures de cada sensor
float tempC;

tempC = analogRead (analog); //llegir valor del sensor
tempC = (1 * tempC * 100.0) / 1024.0; //convertir el valor donat a temperatura

return tempC;
}

```

**Imatge (54):** Ordres dels void loop per explicar el programa.

Per últim escrivim un factor de conversió perquè ens mostri la temperatura amb graus centígrads i no amb la magnitud que ens dona el sensor que seria un voltatge. Cada sensor té un factor de conversió diferent en funció del model, les gràfiques dels sensors segons el voltatge que proporciona està a l'annex 1 del treball.

Hi ha una llibreria que conté un codi per tal de mostrar la pantalla en *html*, en aquesta pantalla es poden assignar les temperatures de consigna màxima i mínima de la sala, així com les temperatures llegides a cada sala, aquest codi també porta els gràfics que es poden veure en el sistema operatiu d'*android*. El codi està en l'annex 2 del treball.





## 9. DISSENY DELS VENTILADORS AMB OPENS CAD.

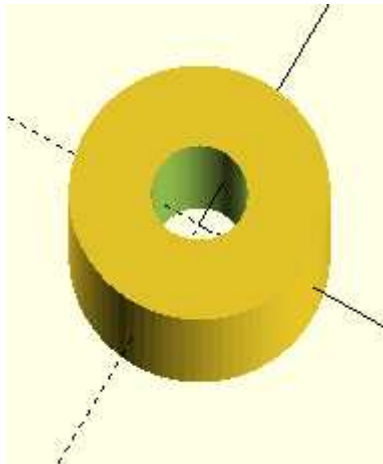
El ventilador serveix per refrigerar la maqueta un cop passi de la temperatura declarada.

Aquest és el codi del programa per construir el ventilador.

El primer pas és escriure la ordre *union*, que uneix totes les peces per poder-les imprimir sense cap problema.

El segon pas és crear un cilindre (*cylinder*) del radi que vulguem en aquest cas és de radi 8mm ( $r=8$ ) i una altura de 16mm ( $h=16$ ), després li diem el número de cares que ha de tenir el cilindre, com més cares més resolució, en aquest cas li hem dit que en tingués 80 ( $fn=80$ ) i el situem al centre de l'eix (*center=true*).

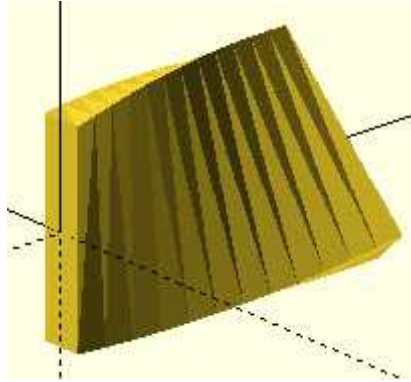
En tercer lloc per poder fer el forat en el cilindre sòlid, utilitzem un cilindre de radi amb les dimensions del forat. En el cas del ventilador, el cilindre per fer el forat és de 3mm, l'altura li diem que sigui més que la del cilindre anterior per assegurar que faci el forat. Per fer el forat utilitzem una ordre anomenada *difference*.



**imatge (55):** Cilindre foradat

A continuació crearem la les pales del ventilador:

El primer pas és declarar un rectangle *square* amb les seves dimensions, en aquest cas totes les pales són de les mateixes mesures 15x3mm i el posem al centre dels eixos. Després fem servir una funció per declarar l'altura *hight=22*, també li diem que ha de rotar sobre ell mateix  $45^\circ$  *twist=45* i per últim definim el numero de cares  $fn=100$



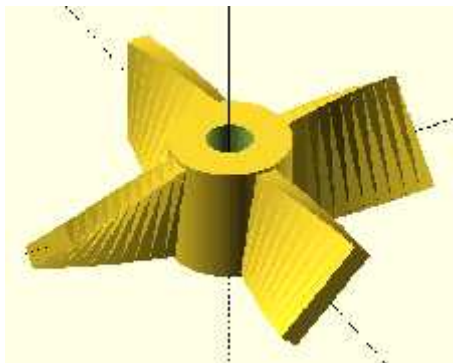
**Imatge (56):** Pala ventilador

Per crear les altres pales hem de seguir el mateix procediment que el que hem vist anteriorment.

Un cop creades les pales les hem de traslladar i posar sobre el cilindre, per això utilitzem l'ordre de traslladar (translate) on li donem les coordenades on volem situar la pala respecte l'eix.

Per poder enganxar les pales i que quedin en forma de creu també hem d'utilitzar una ordre de rotació (rotate) de les peces. En aquest cas la peça la fem rotar 90° perquè quedi perpendicular a les altres dos.

Un cop hem fet tot això, aquest és el resultat de com queda la peça.



**Imatge (57):** Resultat final de la peça

Cilindre	Radi	Altura	Número de cares	Posició
↙	↙	↙	↙	↘
<pre>cylinder (r=3,h=30,\$fn=80 ,center=true);</pre>				

**Imatge (58):** Noms tècnics del programa



```

Trasllat
    translate ([4,0,0])
Rotació → rotate ([0,90,0])
Altura → linear_extrude(height=22
Rotació sobre ell mateix → , twist = 45, $fn=
    100)
Número de cares
Quadrat → square ([15,3],center=
    true);
dimensions del quadrat
    
```

Imatge (59): Llenguatge Openscad

```

1 //ventilador maqueta TDR
2 union () {
3 difference () {
4     //forat al cilindre
5     //incial
6     cylinder(r=3, h=16, $fn=
7     80, center=true);
8
9     cylinder (r=3, h=30, $fn
10    =80, center=true);
11    //cilindre que li
12    restem
13 }
14 //pales ventilador
15 //pala 1
16 translate ([4,0,0])
17 rotate ([0,90,0])
18 linear_extrude(height=22
19     , twist = 45, $fn=
20     100)
21 square ([15,3], center=
22     true);
23
24 //pala 2
25 translate ([0,-4,0])
26 rotate ([90,90,0])
27 linear_extrude(height=
28     22, twist=45, $fn=100)
29 square ([15,3], center=
30     true);
31
32 //pala 3
33 translate ([0,4,0])
34 rotate ([-90,-90,0])
35 linear_extrude(height=
36     22, twist=45, $fn=100)
37 square ([15,3], center=
38     true);
39
40 //pala 4
41 translate ([-25,0,0])
42 rotate ([90,-90,90])
43 linear_extrude(height=
44     22, twist=45, $fn=100)
45 square ([15,3], center=
46     true);
47 }
    
```

Imatge (60): Codi de disseny de l'openscad per crear el ventilador.



## 10. PROBLEMES I SOLUCIONS AL LLARG DE LA REALITZACIÓ DEL TREBALL.

El primer problema que vaig tenir al fer el treball un cop programada per primer cop la placa amb un codi de lectura de temperatura, vaig connectar el sensor de temperatura, però el vaig connectar al revés, és a dir, el positiu del sensor el vaig connectar al negatiu de la placa i el negatiu amb el positiu. Això va provocar que actués com una resistència i es va cremar el sensor.

El segon problema que vaig tenir al poc temps de connectar un segon sensor a la placa que funcionés correctament, al veure la temperatura que marcava el sensor sortia una temperatura superior als 100 graus en una temperatura ambient. Vaig estar dies buscant perquè passava això i no trobava cap mena d'informació referent aquest problema. Més tard vaig descobrir que era culpa d'un cable que feia mal contacte i per això marcava temperatures desorbitades. Un cop canviat el cable ja podia veure la temperatura real que captava el sensor.

El tercer problema amb que hem vaig trobar va ser que al fondre un sensor de temperatura havia de comprar-ne més per poder fer el muntatge de la maqueta. Però vaig comprar un altre model de sensors (I<sub>m</sub>36) que era un model més nou i funcionava amb uns factors de conversió diferents. Amb aquest problema vaig estar buscant per internet i a moltes pàgines es diu que funciona de la mateixa forma. Però finalment vaig trobar unes gràfiques de com funcionaven els diferents sensors de temperatura que feia servir i vaig veure que interpretava d'una altra forma la temperatura i vaig haver de canviar els factors de conversió perquè em mostrés la temperatura real.

El quart problema que vaig tenir va ser que vaig cremar un relé ja que el vaig connectar malament i pensava que no funcionava correctament fins que hem vaig adonar que una part estava desfeta ja que es va cremar i era de plàstic.

Aquest són uns dels problemes amb que he passat més temps intentant solucionar perquè no troba allò que funcionava malament.



## Conclusió

Un cop finalitzat el treball puc dir que m'ha aportat molts coneixements nous gràcies a les recerques que he fet durant el treball i també m'ha permès experimentar amb tot el muntatge de la maqueta. Per això considero que la realització d'aquest treball ha estat molt positiu per mi.

Al principi del treball vaig marcar un objectiu que havia d'assolir, ara un cop acabat puc dir que he assolit la majoria dels que m'havia marcat.

### Objectius assolits:

- Conèixer els tipus de PLC que existeixen. Aquest objectiu el puc considerar com a assolit ja que havia de comparar tota mena de controladors per triar el més adequat per el meu treball.
- Fer un programa que controli un sistema de calefacció. Aquest objectiu el considero com el principal, ja que el meu treball tracta del control de temperatura.
- Construir una maqueta per poder veure el funcionament del programa. Aquest objectiu ha sigut un dels més difícils d'assolir ja que és el que requeria més recerca per totes les connexions que s'havien de fer per no cremar res.
- Crear una aplicació per controlar la temperatura d'un immoble. Aquest objectiu l'he complert parcialment, ja que no he creat una aplicació per controlar la temperatura, sinó que he utilitzat una pàgina web per comunicar-me amb l'arduino i poder regular la temperatura.

### Objectius no assolits:

- Aplicar el programa en un habitatge real. Aquest objectiu no l'he assolit ja que si connectava el controlador a la casa no podria haver fet la maqueta per ensenyar el funcionament. Però es podria haver complert ja que només hauria de connectar la placa al relé de la caldera per controlar el funcionament del sistema de calefacció.

L'extensió del treball és degut a que el projecte necessita una explicació molt detallada per poder fer entendre el lector tots els procediments per fer funcionar la maqueta.



Si hagués d'ampliar el meu treball primer de tot el connectaria en un habitatge real, ja que era un dels objectius que no he complert, i també m'agradaria crear una aplicació per *smartphone* per poder controlar la temperatura dels habitatges sense haver de canviar de xarxa wifi. Aquets són dos exemples que podria fer per ampliar el projecte però aquest treball té infinitat d'opcions per poder-lo ampliar.

Com a conclusió general puc dir que tot i no haver assolit del tot algun objectiu que m'havia proposat al començament del treball, he tingut la oportunitat de treballar en un projecte que em sembla molt interessant. Això m'ha ajudat a fer front en tot moment per tirar endavant el treball fins al final.

Aquest treball m'ha semblat molt satisfactori de fer. M'ha fet aprendre molt sobre un tema que em sembla molt interessant i m'ha fet gaudir mentre l'estava fent. Així que ho recomanaria perquè és experiència que s'ha de viure personalment.



## Agraïments

Agraeixo en primer lloc tot el suport donat per part del meu tutor, Manel García, que m'ha ajudat a tirar endavant el treball i també m'ha donat idees per solucionar els problemes que han anat sorgint al llarg del projecte. També voldria agrair especialment tota l'ajuda i el suport dels meus pares i familiars que m'han donat en tot moment. I per últim agrair a Josep Antoni Rey tota l'ajuda proporcionada en l'àmbit de programació que m'ha ensenyat noves tècniques i m'ha solucionat tots els dubtes que m'anaven sorgint.



## Bibliografia

Pàgines web consultades:

<https://www.youtube.com/watch?v=9-hPhWHWnvs> - getting started with arduino yún

<https://www.youtube.com/watch?v=-fNiG3f0288> - Arduino yun tutorial

<https://www.youtube.com/watch?v=QEIZjCVI2NQ> - curso arduino 7: temperatura - codigo facilito

(<https://www.youtube.com/watch?v=WqF8UJv9RpY>) -arduino yun tutoria: basics

<https://www.youtube.com/watch?v=v6y8DkS6epU> - sensor de temperatura con arduino. subir datos a thinkspeak

<http://www.luisllamas.es/2015/07/medir-temperatura-con-arduino-y-sensor-lm35/> - web sensor de temperatura!

<http://www.tecnosalva.com/pr%C3%A1cticas-3-y-4-arduino-control-motor-y-servomotor> - programar motor cc arduino pag web

<http://www.hell-desk.com/termostato-arduino-yun-conexion-rele-sensor-temperatura-arduino-yun/> - web amb sistema de control de calefacció

[https://ca.wikipedia.org/wiki/Llenguatge\\_de\\_programaci%C3%B3](https://ca.wikipedia.org/wiki/Llenguatge_de_programaci%C3%B3) - llenguatge de programació wikipedia marc teoric

<http://www.digits.cat/colaboracions/llenguatges-de-programacio> - llenguatge de programació (digits)

<http://playground.arduino.cc/Es/OSW06> - llenguatge arduino Marc teoric

<http://playground.arduino.cc/ArduinoNotebookTraduccion/Structure> - llenguatge arduino Marc teoric

[https://ca.wikipedia.org/wiki/Variable\\_\(programaci%C3%B3\)](https://ca.wikipedia.org/wiki/Variable_(programaci%C3%B3)) - que es una variable Marc teoric.

<http://panamahitek.com/el-setup-y-el-loop-en-arduino/> - setup/ loop Marc teoric.

<http://panamahitek.com/arduino-yun-caracteristicas-y-capacidades/> - arduino yun especificacions





## Bibliografia fotogràfica

Imatge 1:

[https://www.google.es/search?q=bomba+calor&espv=2&biw=1242&bih=602&source=lnms&tbm=isch&sa=X&ved=0ahUKEwiL2JK66tPQAhUoCsAKHYWnCA8Q\\_AUIBigB#imgsrc=8IFgvfslGLpyjM%3A](https://www.google.es/search?q=bomba+calor&espv=2&biw=1242&bih=602&source=lnms&tbm=isch&sa=X&ved=0ahUKEwiL2JK66tPQAhUoCsAKHYWnCA8Q_AUIBigB#imgsrc=8IFgvfslGLpyjM%3A)

Imatge 2:

[https://www.google.es/search?q=terra+radiant&espv=2&biw=1242&bih=602&source=lnms&tbm=isch&sa=X&ved=0ahUKEwjM8PuO6tPQAhVE2xoKHQpZBvIQ\\_AUIBigB#imgsrc=iaZYgprznifU8M%3A](https://www.google.es/search?q=terra+radiant&espv=2&biw=1242&bih=602&source=lnms&tbm=isch&sa=X&ved=0ahUKEwjM8PuO6tPQAhVE2xoKHQpZBvIQ_AUIBigB#imgsrc=iaZYgprznifU8M%3A)

Imatge 3:

[https://www.google.es/search?q=sistema+de+calefacci%C3%B3+per+radiador&espv=2&biw=1242&bih=602&source=lnms&tbm=isch&sa=X&sqi=2&ved=0ahUKEwjOhp\\_X6tPQAhUDDxoKHxO5A\\_IQ\\_AUIBigB&dpr=1.1#imgsrc=AHo0qHOPCB1OQM%3A](https://www.google.es/search?q=sistema+de+calefacci%C3%B3+per+radiador&espv=2&biw=1242&bih=602&source=lnms&tbm=isch&sa=X&sqi=2&ved=0ahUKEwjOhp_X6tPQAhUDDxoKHxO5A_IQ_AUIBigB&dpr=1.1#imgsrc=AHo0qHOPCB1OQM%3A)

Imatge 4:

[https://www.google.es/search?q=sistema+de+calefacci%C3%B3+per+radiador&espv=2&biw=1242&bih=602&source=lnms&tbm=isch&sa=X&sqi=2&ved=0ahUKEwjOhp\\_X6tPQAhUDDxoKHxO5A\\_IQ\\_AUIBigB&dpr=1.1#tbn=isch&q=emissor+termoel%C3%A8ctric&imgsrc=3M9JcW-iGf\\_imM%3A](https://www.google.es/search?q=sistema+de+calefacci%C3%B3+per+radiador&espv=2&biw=1242&bih=602&source=lnms&tbm=isch&sa=X&sqi=2&ved=0ahUKEwjOhp_X6tPQAhUDDxoKHxO5A_IQ_AUIBigB&dpr=1.1#tbn=isch&q=emissor+termoel%C3%A8ctric&imgsrc=3M9JcW-iGf_imM%3A)

Imatge 5:

[https://www.google.es/search?q=arduino+uno&espv=2&biw=1242&bih=602&source=lnms&tbm=isch&sa=X&ved=0ahUKEwiRs6mm69PQAhUJbBoKHUOBCIQQ\\_AUIBigB#imgsrc=kIRvIILMaKcWHM%3A](https://www.google.es/search?q=arduino+uno&espv=2&biw=1242&bih=602&source=lnms&tbm=isch&sa=X&ved=0ahUKEwiRs6mm69PQAhUJbBoKHUOBCIQQ_AUIBigB#imgsrc=kIRvIILMaKcWHM%3A)

Imatge 6:

[https://www.google.es/search?q=arduino+mega&espv=2&biw=1242&bih=602&source=lnms&tbm=isch&sa=X&ved=0ahUKEwiA-Mq969PQAhXKnBoKHx6uCPiQ\\_AUIBigB#imgsrc=W0gSOO2BdVbQpM%3A](https://www.google.es/search?q=arduino+mega&espv=2&biw=1242&bih=602&source=lnms&tbm=isch&sa=X&ved=0ahUKEwiA-Mq969PQAhXKnBoKHx6uCPiQ_AUIBigB#imgsrc=W0gSOO2BdVbQpM%3A)



Imatge 7:

[https://www.google.es/search?q=arduino+yun&espv=2&biw=1242&bih=602&source=lnms&tbm=isch&sa=X&ved=0ahUKEwiOqojd69PQAhWF5xoKHbSdCfMQ\\_AUIBigB#imgdii=ellbXCJ-zxzNZM%3A%3BellbXCJ-zxzNZM%3A%3BqD18vp4q4gWprM%3A&imgrc=ellbXCJ-zxzNZM%3A](https://www.google.es/search?q=arduino+yun&espv=2&biw=1242&bih=602&source=lnms&tbm=isch&sa=X&ved=0ahUKEwiOqojd69PQAhWF5xoKHbSdCfMQ_AUIBigB#imgdii=ellbXCJ-zxzNZM%3A%3BellbXCJ-zxzNZM%3A%3BqD18vp4q4gWprM%3A&imgrc=ellbXCJ-zxzNZM%3A)

Imatge 8:

[https://www.google.es/search?q=arduino+yun&espv=2&biw=1242&bih=602&source=lnms&tbm=isch&sa=X&ved=0ahUKEwiOqojd69PQAhWF5xoKHbSdCfMQ\\_AUIBigB#tbnm=isch&q=arduino+yun+reset&imgdii=61C0sBd1Fi1fuM%3A%3B61C0sBd1Fi1fuM%3A%3BellbXCJ-zxzNZM%3A&imgrc=61C0sBd1Fi1fuM%3A](https://www.google.es/search?q=arduino+yun&espv=2&biw=1242&bih=602&source=lnms&tbm=isch&sa=X&ved=0ahUKEwiOqojd69PQAhWF5xoKHbSdCfMQ_AUIBigB#tbnm=isch&q=arduino+yun+reset&imgdii=61C0sBd1Fi1fuM%3A%3B61C0sBd1Fi1fuM%3A%3BellbXCJ-zxzNZM%3A&imgrc=61C0sBd1Fi1fuM%3A)

Imatge 9:

[https://www.google.es/search?q=arduino+yun&espv=2&biw=1242&bih=602&source=lnms&tbm=isch&sa=X&ved=0ahUKEwiOqojd69PQAhWF5xoKHbSdCfMQ\\_AUIBigB#tbnm=isch&q=arduino+yun+reset&imgrc=61C0sBd1Fi1fuM%3A](https://www.google.es/search?q=arduino+yun&espv=2&biw=1242&bih=602&source=lnms&tbm=isch&sa=X&ved=0ahUKEwiOqojd69PQAhWF5xoKHbSdCfMQ_AUIBigB#tbnm=isch&q=arduino+yun+reset&imgrc=61C0sBd1Fi1fuM%3A)

Imatge 10:

Elaboració pròpia.

Imatge 11:

Elaboració pròpia.

Imatge 12:

[https://www.google.es/search?q=lm+35&espv=2&biw=1242&bih=602&source=lnms&tbnm=isch&sa=X&ved=0ahUKEwj09Zap-9PQAhUCsxQKHd2NAewQ\\_AUIBigB#imgrc=qBtWtVHKSQHnZM%3A](https://www.google.es/search?q=lm+35&espv=2&biw=1242&bih=602&source=lnms&tbnm=isch&sa=X&ved=0ahUKEwj09Zap-9PQAhUCsxQKHd2NAewQ_AUIBigB#imgrc=qBtWtVHKSQHnZM%3A)

Imatge 13:

[https://www.google.es/search?q=motor+cc&espv=2&biw=1242&bih=602&source=lnms&tbnm=isch&sa=X&ved=0ahUKEwjFz9u77NPQAhVEXhoKHcmiDSMQ\\_AUIBigB#tbnm=isch&q=motor+cc+arduino&imgrc=j2n\\_xlCDUtZz-M%3A](https://www.google.es/search?q=motor+cc&espv=2&biw=1242&bih=602&source=lnms&tbnm=isch&sa=X&ved=0ahUKEwjFz9u77NPQAhVEXhoKHcmiDSMQ_AUIBigB#tbnm=isch&q=motor+cc+arduino&imgrc=j2n_xlCDUtZz-M%3A)

Imatge 14:

Elaboració pròpia.

Imatge 15:

Elaboració pròpia.



Imatge 16:

Elaboració pròpia.

Imatge 17:

Elaboració pròpia

Imatge 18:

[https://www.google.es/search?q=sketchup&biw=1242&bih=602&source=Inms&tbm=isch&sa=X&ved=0ahUKEwigxarv7NPQAhUFMhoKHWgwBvIQ\\_AUIBigB#tbm=isch&q=sketchup+logo&imgsrc=JO4YopQgiVggNM%3A](https://www.google.es/search?q=sketchup&biw=1242&bih=602&source=Inms&tbm=isch&sa=X&ved=0ahUKEwigxarv7NPQAhUFMhoKHWgwBvIQ_AUIBigB#tbm=isch&q=sketchup+logo&imgsrc=JO4YopQgiVggNM%3A)

Imatge 19:

[https://www.google.es/search?q=arduino&espv=2&biw=1242&bih=602&source=Inms&tbm=isch&sa=X&ved=0ahUKEwipy9H67NPQAhWEzRoKHb\\_EBvMQ\\_AUIBigB#imgsrc=-ylmVRV3v6I2qM%3A](https://www.google.es/search?q=arduino&espv=2&biw=1242&bih=602&source=Inms&tbm=isch&sa=X&ved=0ahUKEwipy9H67NPQAhWEzRoKHb_EBvMQ_AUIBigB#imgsrc=-ylmVRV3v6I2qM%3A)

Imatge 20:

[https://www.google.es/search?q=opencad&espv=2&biw=1242&bih=602&source=Inms&tbm=isch&sa=X&ved=0ahUKEwiGyYWM7dPQAhXBXRoKHRoiBggQ\\_AUIBigB#tbm=isch&q=logo+opencad&imgsrc=D9mqWsT81FW0KM%3A](https://www.google.es/search?q=opencad&espv=2&biw=1242&bih=602&source=Inms&tbm=isch&sa=X&ved=0ahUKEwiGyYWM7dPQAhXBXRoKHRoiBggQ_AUIBigB#tbm=isch&q=logo+opencad&imgsrc=D9mqWsT81FW0KM%3A)

Imatge 21:

Elaboració pròpia.

Imatge 22:

[https://www.google.es/search?q=arduino+yun&espv=2&biw=1242&bih=602&source=Inms&tbm=isch&sa=X&ved=0ahUKEwiPsqjW-9PQAhVF6xQKHem3AekQ\\_AUIBigB#imgsrc=ZQmP2iRHRHGL5M%3A](https://www.google.es/search?q=arduino+yun&espv=2&biw=1242&bih=602&source=Inms&tbm=isch&sa=X&ved=0ahUKEwiPsqjW-9PQAhVF6xQKHem3AekQ_AUIBigB#imgsrc=ZQmP2iRHRHGL5M%3A)

Imatge 23:

[https://www.google.es/search?q=lm+35&espv=2&biw=1242&bih=602&source=Inms&tbm=isch&sa=X&ved=0ahUKEwjo9Zap-9PQAhUCsxQKHd2NAewQ\\_AUIBigB#imgsrc=qBtWtVHKSQHnZM%3A](https://www.google.es/search?q=lm+35&espv=2&biw=1242&bih=602&source=Inms&tbm=isch&sa=X&ved=0ahUKEwjo9Zap-9PQAhUCsxQKHd2NAewQ_AUIBigB#imgsrc=qBtWtVHKSQHnZM%3A)

Imatge 24:

Elaboració pròpia.

Imatge 25:



[https://www.google.es/search?q=protoboard&espv=2&biw=1242&bih=602&source=Inms&tbm=isch&sa=X&ved=0ahUKEwjPytWN-9PQAhVGOBQKHbNDBe8Q\\_AUIBigB#imgrc=wC0ONPSOkByriM%3A](https://www.google.es/search?q=protoboard&espv=2&biw=1242&bih=602&source=Inms&tbm=isch&sa=X&ved=0ahUKEwjPytWN-9PQAhVGOBQKHbNDBe8Q_AUIBigB#imgrc=wC0ONPSOkByriM%3A)

Imatge 26:

Elaboració pròpia.

Imatge 27:

Elaboració pròpia.

Imatge 28:

Elaboració pròpia.

Imatge 29:

[https://www.google.es/search?q=rele&espv=2&biw=1242&bih=602&source=Inms&tbm=isch&sa=X&ved=0ahUKEwjZv\\_Hw-tPQAhVEwBQKHUfmDu4Q\\_AUIBigB#imgrc=jkmTh\\_zd3RE9AM%3A](https://www.google.es/search?q=rele&espv=2&biw=1242&bih=602&source=Inms&tbm=isch&sa=X&ved=0ahUKEwjZv_Hw-tPQAhVEwBQKHUfmDu4Q_AUIBigB#imgrc=jkmTh_zd3RE9AM%3A)

Imatge 30:

Elaboració pròpia.

Imatge 31:

[https://www.google.es/search?q=metracrilat&espv=2&biw=1242&bih=602&source=Inms&tbm=isch&sa=X&ved=0ahUKEwjg6\\_Ki-tPQAhWDVBQKHfXcDesQ\\_AUIBygC#tbm=isch&q=metracrilat&imgrc=3vbzeA5mMvoQEM%3A](https://www.google.es/search?q=metracrilat&espv=2&biw=1242&bih=602&source=Inms&tbm=isch&sa=X&ved=0ahUKEwjg6_Ki-tPQAhWDVBQKHfXcDesQ_AUIBygC#tbm=isch&q=metracrilat&imgrc=3vbzeA5mMvoQEM%3A)

Imatge 32:

[https://www.google.es/search?q=calble+usb+micro+usb&espv=2&biw=1242&bih=602&tbm=isch&imgil=tFWQ6gDHxKhB0M%253A%253BNd4txvozTMSv7M%253Bhttp%25253A%25252F%25252Fwww.l-link.es%25252Fproductos%25252Flistado.asp%25253FtxtFamilia%2525253DAccesorios%25252526txtSubFamilia%2525253DCables%2525252520USB%2525252520y%25252520FIREWARE%25252526txtMarca%2525253DL-Link&source=iu&pf=m&fir=tFWQ6gDHxKhB0M%253A%252CNd4txvozTMSv7M%252C\\_&usg=\\_\\_Mf6rCINr3RJYi5pdtFXDw0YRTOw%3D&ved=0ahUKEwi2pryD-tPQAhWC1xQKHVCxAewQyjcliAE&ei=EphAWLbVGyKvU9DihuAO](https://www.google.es/search?q=calble+usb+micro+usb&espv=2&biw=1242&bih=602&tbm=isch&imgil=tFWQ6gDHxKhB0M%253A%253BNd4txvozTMSv7M%253Bhttp%25253A%25252F%25252Fwww.l-link.es%25252Fproductos%25252Flistado.asp%25253FtxtFamilia%2525253DAccesorios%25252526txtSubFamilia%2525253DCables%2525252520USB%2525252520y%25252520FIREWARE%25252526txtMarca%2525253DL-Link&source=iu&pf=m&fir=tFWQ6gDHxKhB0M%253A%252CNd4txvozTMSv7M%252C_&usg=__Mf6rCINr3RJYi5pdtFXDw0YRTOw%3D&ved=0ahUKEwi2pryD-tPQAhWC1xQKHVCxAewQyjcliAE&ei=EphAWLbVGyKvU9DihuAO)



Imatge 33:

Elaboració pròpia.

Imatge 34:

[https://www.google.es/search?q=font+alimentaci%C3%B3+dell&espv=2&biw=1242&bih=602&source=lnms&tbm=isch&sa=X&ved=0ahUKEwjriZqf-dPQAhXItxQKHUNGcugQ\\_AUIBigB#tbm=isch&q=dell+fuente+alimentacion&imgref=fB9gGwDlrAixxM%3A](https://www.google.es/search?q=font+alimentaci%C3%B3+dell&espv=2&biw=1242&bih=602&source=lnms&tbm=isch&sa=X&ved=0ahUKEwjriZqf-dPQAhXItxQKHUNGcugQ_AUIBigB#tbm=isch&q=dell+fuente+alimentacion&imgref=fB9gGwDlrAixxM%3A)

Imatge 35:

Elaboració pròpia.

Imatge 36:

Elaboració pròpia.

Imatge 37:

Elaboració pròpia.

Imatge 38:

Elaboració pròpia.

Imatge 39:

Elaboració pròpia.

Imatge 40:

Elaboració pròpia.

Imatge 41:

Elaboració pròpia.

Imatge 42:

Elaboració pròpia.

Imatge 43:

Elaboració pròpia.

Imatge 44:

Elaboració pròpia.



Imatge 45:

Elaboració pròpia.

Imatge 46:

Elaboració pròpia.

Imatge 47:

Elaboració pròpia.

Imatge 48:

Elaboració pròpia.

Imatge 49:

Elaboració pròpia.

Imatge 50:

Elaboració pròpia.

Imatge 51:

Elaboració pròpia.

Imatge 52:

Elaboració pròpia.

Imatge 53:

Elaboració pròpia.

Imatge 54:

Elaboració pròpia.

Imatge 55:

Elaboració pròpia.

Imatge 56:

Elaboració pròpia.

Imatge 57:

Elaboració pròpia.



Imatge 58:

Elaboració pròpia.

Imatge 59:

Elaboració pròpia.

Imatge 60:

Elaboració pròpia.



## **ANNEX 1.**





Data Sheet

TMP35/TMP36/TMP37

**SPECIFICATIONS**

$V_S = 2.7\text{ V to } 5.5\text{ V}$ ,  $-40^\circ\text{C} \leq T_A \leq +125^\circ\text{C}$ , unless otherwise noted.

Table 1.

Parameter <sup>1</sup>	Symbol	Test Conditions/Comments	Min	Typ	Max	Unit
<b>ACCURACY</b>						
TMP35/TMP36/TMP37 (F Grade)		$T_A = 25^\circ\text{C}$		±1	±2	°C
TMP35/TMP36/TMP37 (G Grade)		$T_A = 25^\circ\text{C}$		±1	±3	°C
TMP35/TMP36/TMP37 (F Grade)		Over rated temperature		±2	±3	°C
TMP35/TMP36/TMP37 (G Grade)		Over rated temperature		±2	±4	°C
Scale Factor, TMP35		$10^\circ\text{C} \leq T_A \leq 125^\circ\text{C}$		10		mV/°C
Scale Factor, TMP36		$-40^\circ\text{C} \leq T_A \leq +125^\circ\text{C}$		10		mV/°C
Scale Factor, TMP37		$5^\circ\text{C} \leq T_A \leq 85^\circ\text{C}$		20		mV/°C
		$5^\circ\text{C} \leq T_A \leq 100^\circ\text{C}$		20		mV/°C
Load Regulation		$3.0\text{ V} \leq V_S \leq 5.5\text{ V}$ $0\ \mu\text{A} \leq I_L \leq 50\ \mu\text{A}$				
		$-40^\circ\text{C} \leq T_A \leq +105^\circ\text{C}$		6	20	m°C/μA
		$-105^\circ\text{C} \leq T_A \leq +125^\circ\text{C}$		25	60	m°C/μA
Power Supply Rejection Ratio	PSRR	$T_A = 25^\circ\text{C}$		30	100	m°C/V
		$3.0\text{ V} \leq V_S \leq 5.5\text{ V}$		50		m°C/V
Linearity				0.5		°C
Long-Term Stability		$T_A = 150^\circ\text{C}$ for 1000 hours		0.4		°C
<b>SHUTDOWN</b>						
Logic High Input Voltage	$V_{IH}$	$V_S = 2.7\text{ V}$	1.8			V
Logic Low Input Voltage	$V_{IL}$	$V_S = 5.5\text{ V}$			400	mV
<b>OUTPUT</b>						
TMP35 Output Voltage		$T_A = 25^\circ\text{C}$		250		mV
TMP36 Output Voltage		$T_A = 25^\circ\text{C}$		750		mV
TMP37 Output Voltage		$T_A = 25^\circ\text{C}$		500		mV
Output Voltage Range			100		2000	mV
Output Load Current	$I_L$		0		50	μA
Short-Circuit Current	$I_{SC}$	Note 2			250	μA
Capacitive Load Driving	$C_L$	No oscillations <sup>2</sup>	1000	10000		pF
Device Turn-On Time		Output within ±1°C, 100 kΩ  100 pF load <sup>2</sup>		0.5	1	ms
<b>POWER SUPPLY</b>						
Supply Range	$V_S$		2.7		5.5	V
Supply Current	$I_{SV}$ (ON)	Unloaded			50	μA
Supply Current (Shutdown)	$I_{SV}$ (OFF)	Unloaded		0.01	0.5	μA

<sup>1</sup> Does not consider errors caused by self-heating.

<sup>2</sup> Guaranteed but not tested.



TYPICAL PERFORMANCE CHARACTERISTICS

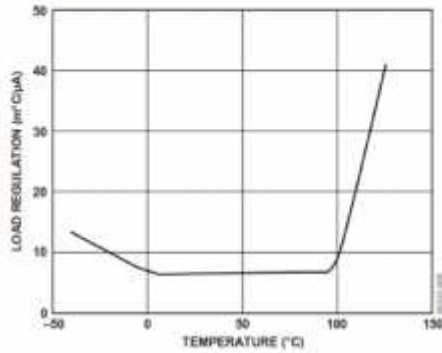


Figure 5. Load Regulation vs. Temperature (mV/µA)

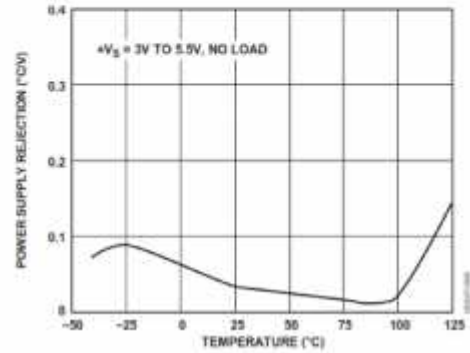


Figure 8. Power Supply Rejection vs. Temperature

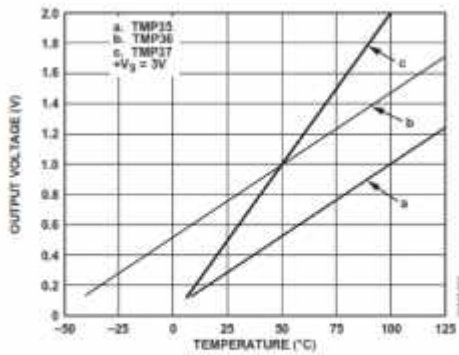


Figure 6. Output Voltage vs. Temperature

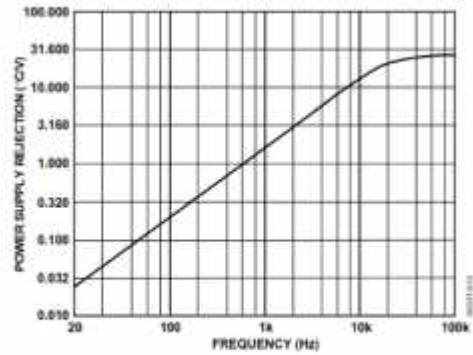


Figure 9. Power Supply Rejection vs. Frequency

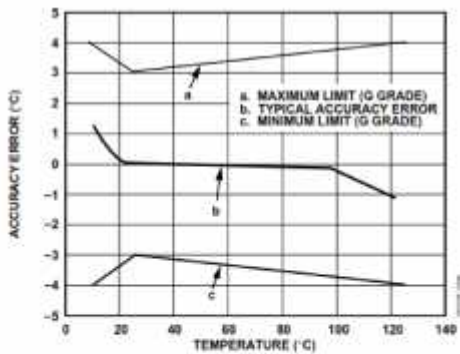


Figure 7. Accuracy Error vs. Temperature

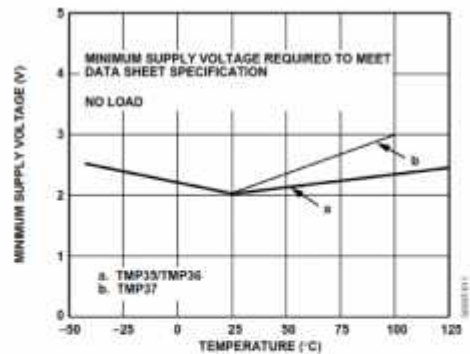


Figure 10. Minimum Supply Voltage vs. Temperature



## **ANNEX 2.**

**CODI:**

Llibreria utilitzada per la comunicació de la placa i la pàgina web htm per controlar la temperatura des d'una pàgina web.

IP: 192.168.240.1/sd/

```
#include <Bridge.h>;
#include <YunServer.h>
#include <YunClient.h>
```

```
YunServer server;
YunClient client;
```

```
unsigned long TEMPS_AGAFANT_MOSTRA = 10000; //Temps agafant mostra temp.
unsigned long iniciAgafantMostra;
```

```
void setup() {
  Bridge.begin();
  server.listenOnLocalhost();
  server.begin();
}
```

```
String llegirDades() {
  String rspJSON = "";
  for (int i =0;i<QTT_SALES;i++){
    if (i == 0 )
      rspJSON = "\"MinH"+String(i)+"\":\\"" + String(tempMin[i]) + "\",\"MaxH"+String(i)+"\":\\""
+String(tempMax[i]) + "\",\"TmpH"+String(i)+"\":\\"" + String(mostraTemperatura(i)) + "\"";
    else
      rspJSON = rspJSON + "," + "\"MinH"+String(i)+"\":\\"" + String(tempMin[i])
+ "\",\"MaxH"+String(i)+"\":\\"" +String(tempMax[i]) + "\",\"TmpH"+String(i)+"\":\\"" +
String(mostraTemperatura(i)) + "\"";
  }
}
```

```
return "{" + rspJSON + "}";
```

```
// return "{\"MinH1\":\\"" + String(tempMin[0]) + "\",\"MaxH1\":\\"" +String(tempMax[0]) +
+ "\",\"TmpH1\":\\"" + String(mostraTemperatura(0)) + "\",\"MinH2\":\\"" +String(tempMin[1]) +
+ "\",\"MaxH2\":\\"" + String(tempMax[1]) + "\",\"TmpH2\":\\"" + String(mostraTemperatura(1)) + "\"}";
```

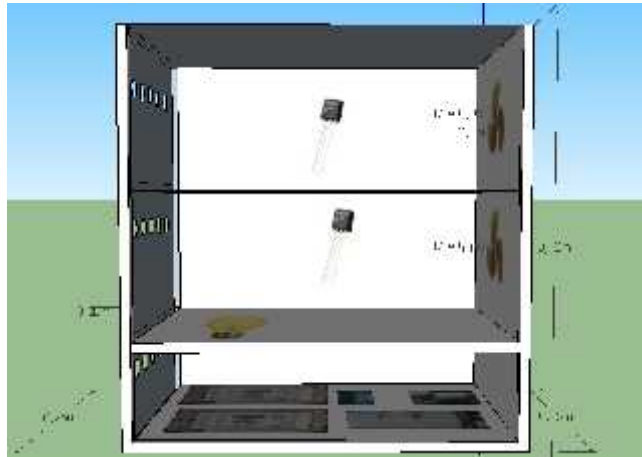
```
void afegirCapcaleraJSON() {
  client.println("Status: 200");
  client.println("Content-type: application/json");
  client.println();
}
```



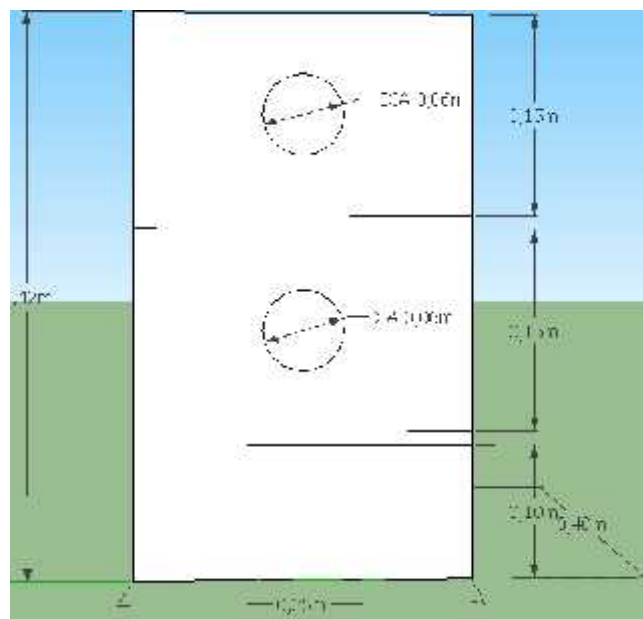
## **ANNEX 3.**



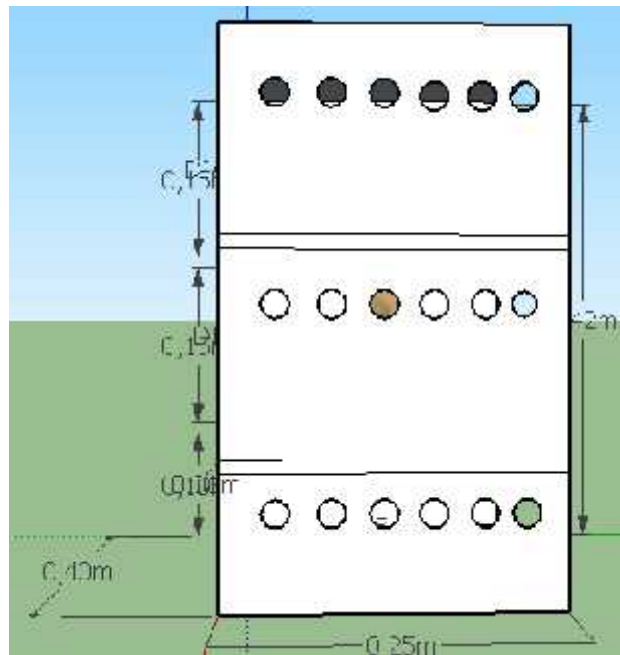
Disseny en tres dimensions dels plànols de la maqueta amb *sketchup*.



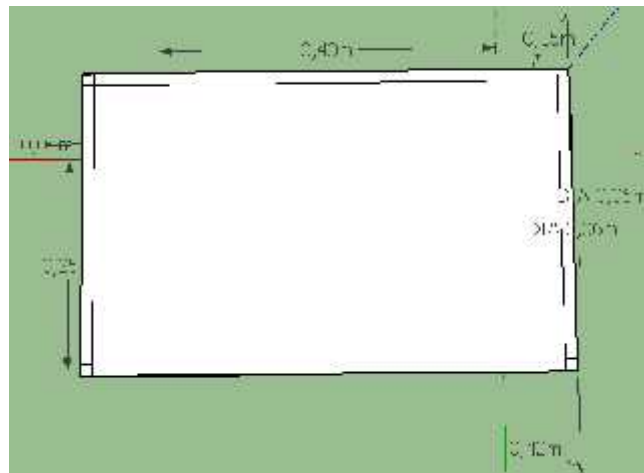
Disseny de la maqueta (Vista: Perfil).



Disseny de la maqueta (Vista: Alçat).



Disseny de la maqueta (Vista: Alçat).



Disseny de la maqueta (Vista: Planta).

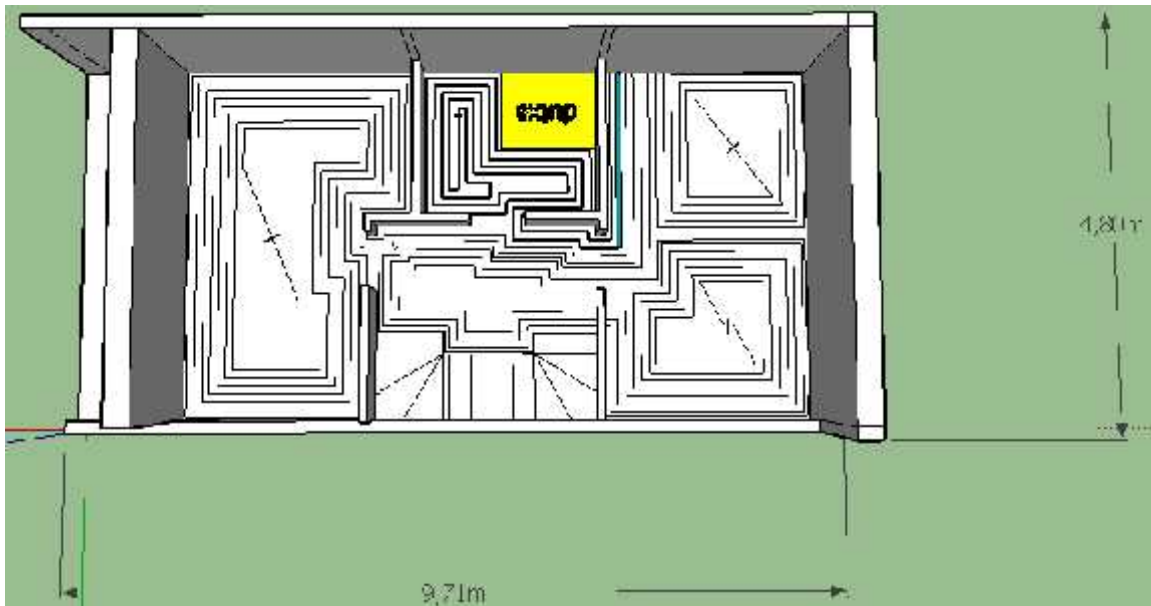


## **ANNEX 4.**





Disseny en tres dimensions del sistema de calefacció per terra radiant de l'habitatge on es volia implementar la placa amb el codi del control del sistema de calefacció.



Disseny en tres dimensions del sistema de calefacció per terra radiant.



Canonades del sistema de calefacció per terra radiant per on circula l'aigua.



### 3. OPTIMIZACIÓN DE LAS BOBINAS DE TUBO

Para optimizar al máximo la utilización de las bobinas, se recomienda utilizar la siguiente distribución. No cortar nunca las bobinas a medida.

<b>Coil n.:</b> 1	<b>Tubo:</b> Tubo PEX-A 16x1,8	<b>Long. [m]:</b> 200	<b>Resto <sup>3</sup> [m]:</b>
-------------------	--------------------------------	-----------------------	--------------------------------

Espacio	Circuito n.	Long [m]	VT	[m] inicio	[m] fin	[m] pond
P.P-2 - Dormitorio	1	68				
P.P-4 - Baño	4	38				
P.S-2 - Baño	3	91				

<b>Coil n.:</b> 2	<b>Tubo:</b> Tubo PEX-A 16x1,8	<b>Long. [m]:</b> 200	<b>Resto <sup>15</sup> [m]:</b>
-------------------	--------------------------------	-----------------------	---------------------------------

Espacio	Circuito n.	Long [m]	VT	[m] inicio	[m] fin	[m] pond
P.B.-2 - Salón	4	85				
P.P-3 - Dormitorio	3	100				

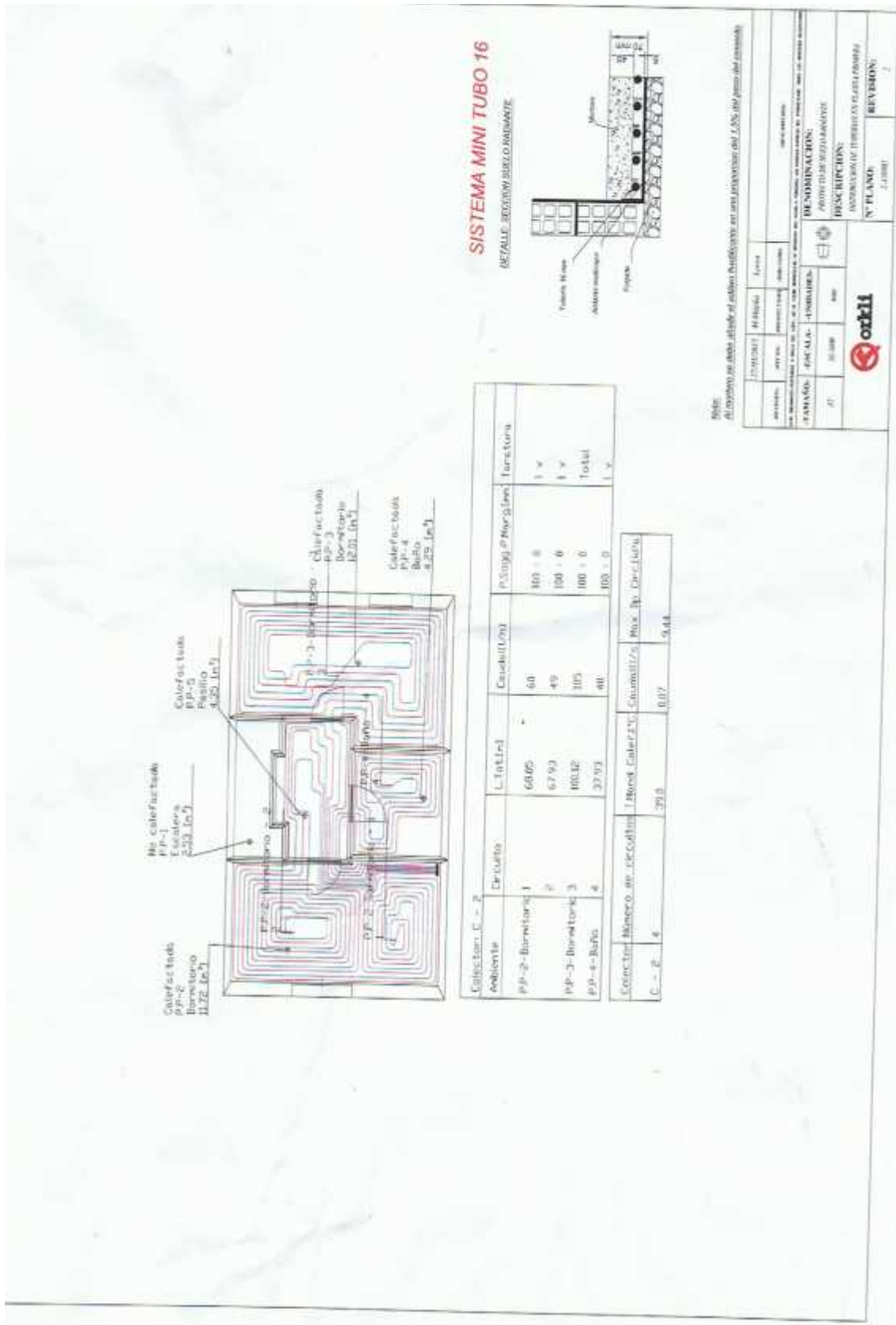
<b>Coil n.:</b> 3	<b>Tubo:</b> Tubo PEX-A 16x1,8	<b>Long. [m]:</b> 200	<b>Resto <sup>6</sup> [m]:</b>
-------------------	--------------------------------	-----------------------	--------------------------------

Espacio	Circuito n.	Long [m]	VT	[m] inicio	[m] fin	[m] pond
P.B.-1 - Cocina	2	42				
P.B.-2 - Salón	3	84				
P.P-2 - Dormitorio	2	68				

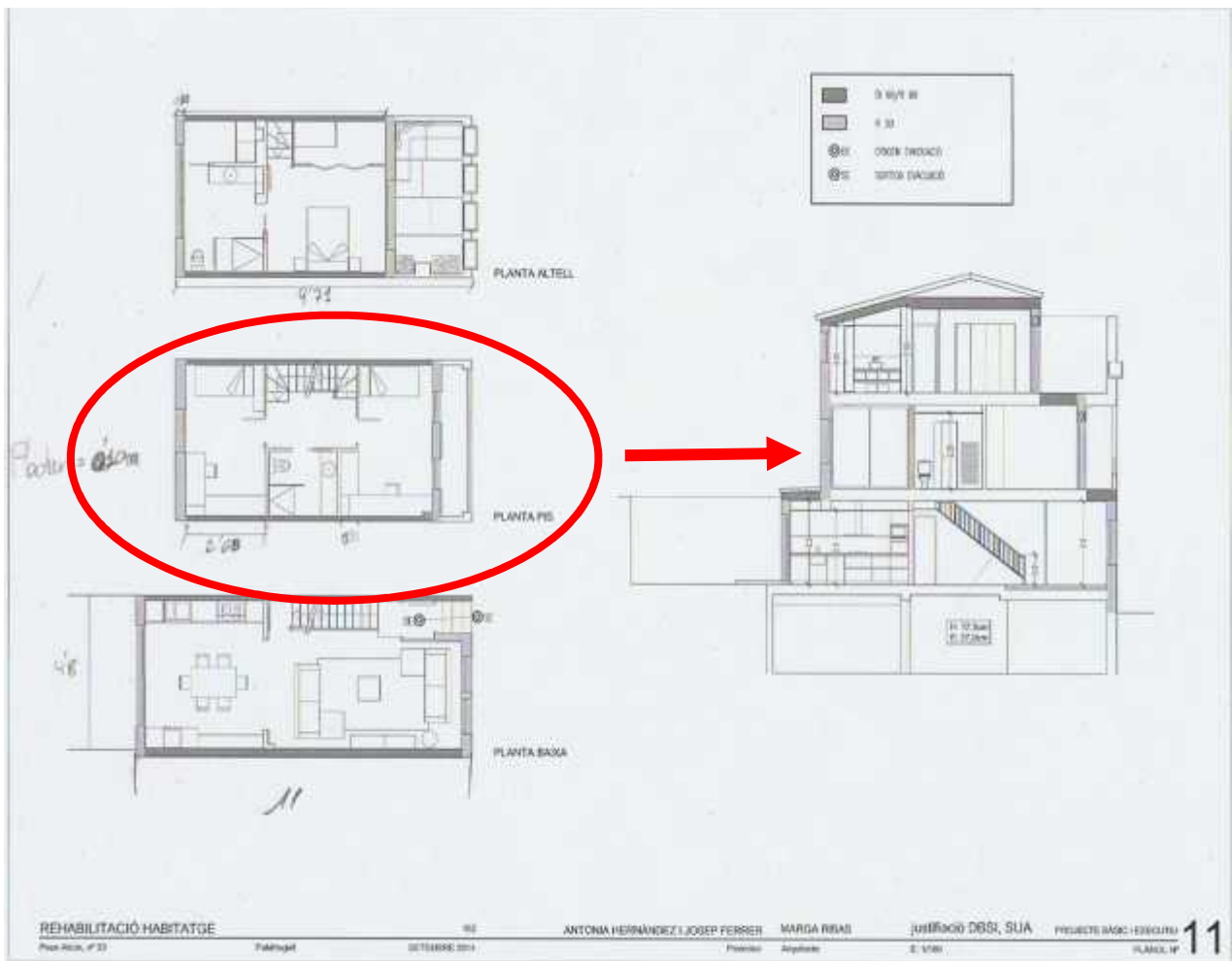
<b>Coil n.:</b> 4	<b>Tubo:</b> Tubo PEX-A 16x1,8	<b>Long. [m]:</b> 200	<b>Resto <sup>11</sup> [m]:</b>
-------------------	--------------------------------	-----------------------	---------------------------------

Espacio	Circuito n.	Long [m]	VT	[m] inicio	[m] fin	[m] pond
P.B.-1 - Cocina	1	44				
P.S-1 - Dormitorio	1	74				
P.S-1 - Dormitorio	2	71				

Quantitat en metres de les canonades utilitzades.



Distribució de les canonades de sistema de calefacció per terra radiant



Plànol de l'habitatge dissenyat amb *AutoCAD*.