

Enginyeria inversa d'un aparell d'aire condicionat per connectar-lo al WiFi

Albert Vaca Cintora <albertvaka@gmail.com>

Projecte original: Albert Vaca Cintora, Joan Montserrat, Marce Coll

Mitsubishi Heavy Industries, model FDURA301



Controlador de paret



Aparell central amagat al sostre

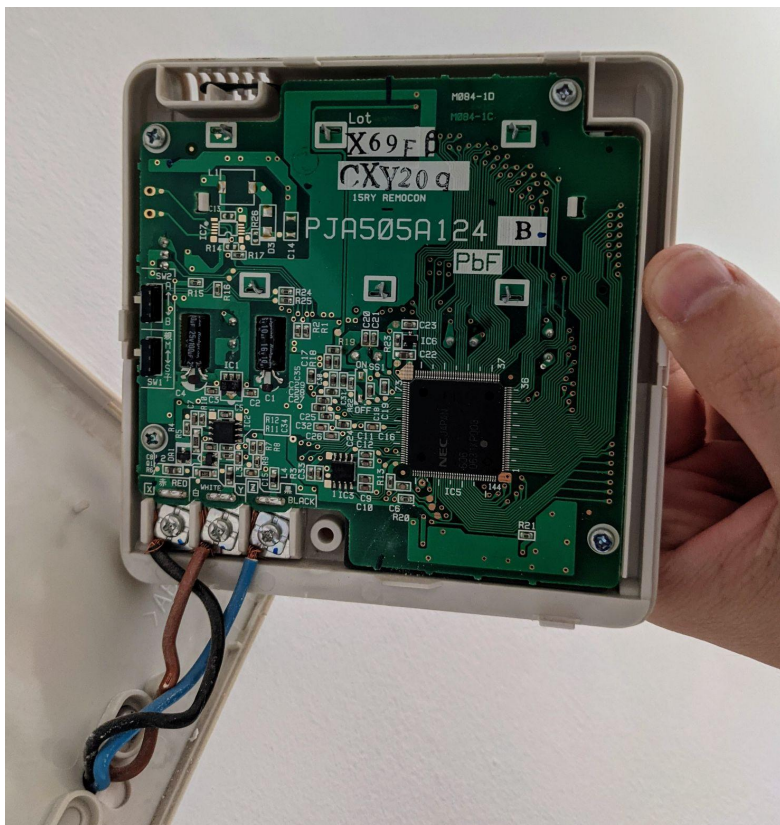
Mitsubishi Heavy Industries, model FDURA301



Per canviar la temperatura m'he d'aixecar del sofà.

L'única automatització possible és configurar l'hora d'encendre/apagar.

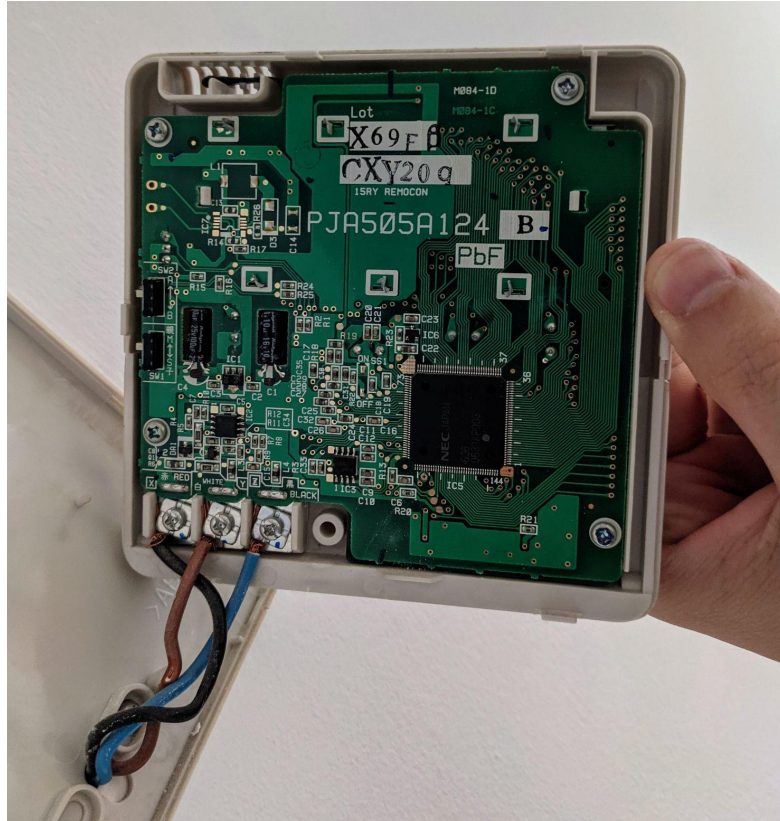
Podem fer-ho millor...



El controlador i la màquina "parlen"
d'alguna manera a través de 3 cables.

El nostre objectiu és trobar com parlen...

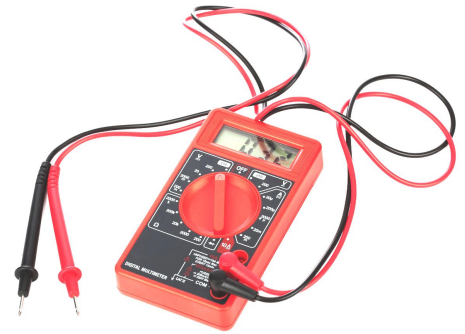




Un dels cables és facil d'identificar: GND.

Podem saber quin és mirant components amb polaritat, p.e.: condensadors.

Havent trobat GND, amb un tester veiem que els altres dos cables van a +12V.



Com trobem què passa pels cables?



Analitzador logic barato

+



sigrok

Software lliure per a fer servir
oscil·loscopis, analitzadors lògics...

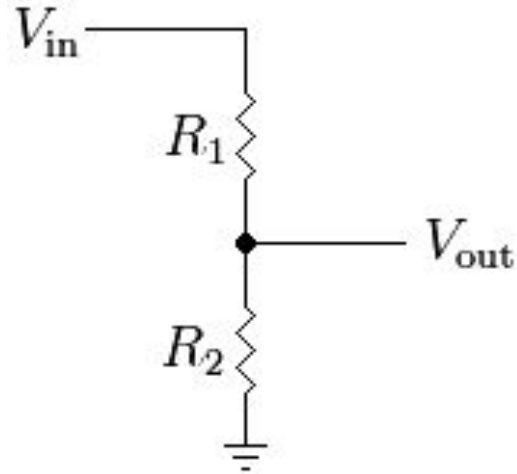
<http://sigrok.org>



PulseView, la interfície gràfica de Sigrok

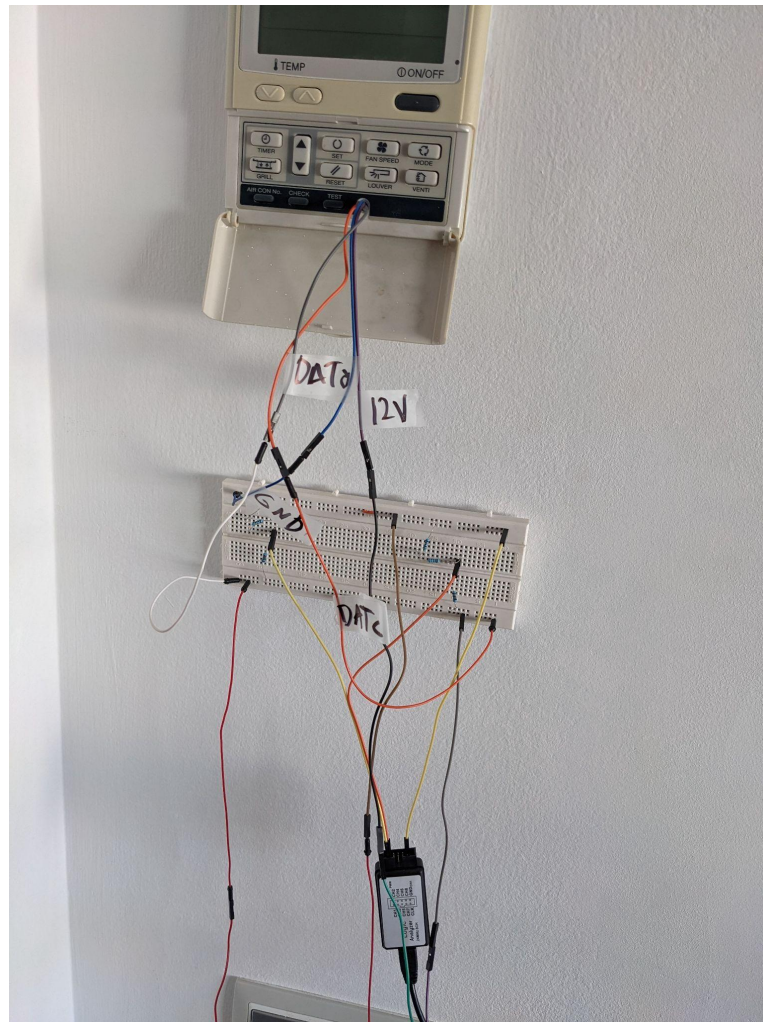
Oju! No podem connectar directament 12V a l'analitzador
El fregiríem.

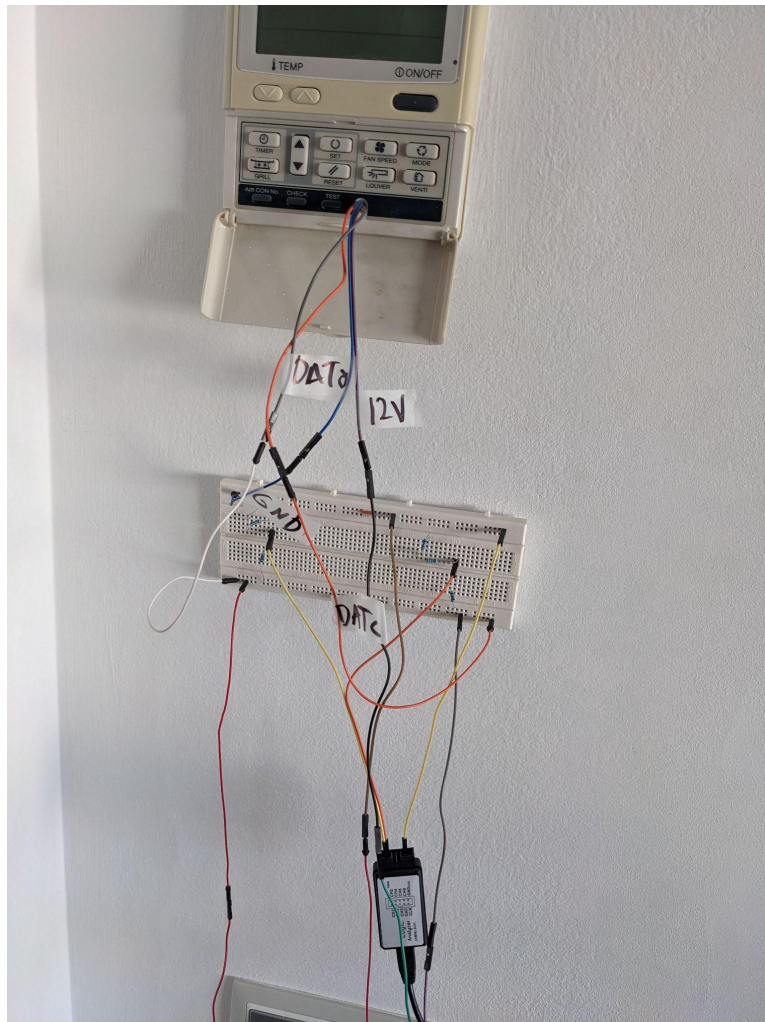
Oju! No podem connectar directament 12V a l'analitzador



$$V_{out} = \frac{R_2}{R_1 + R_2} V_{in}$$

Esquema i fòrmula d'un divisor de tensió.





Dos divisores de tensió entre els dos cables de 12V i l'analitzador.

Els divisores de tensió passen de 12V a ~3V.

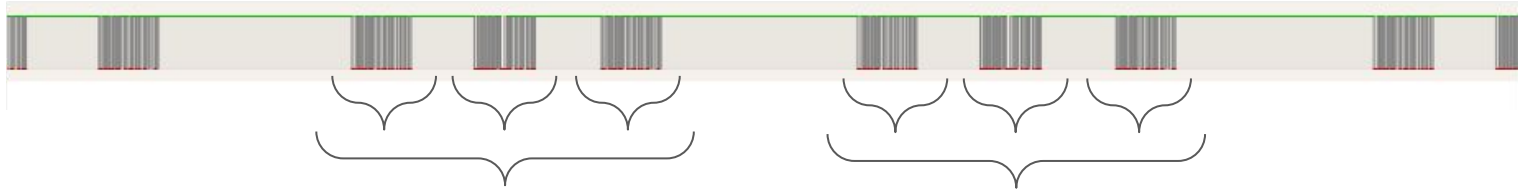
El GND també el connectem a l'analitzador, és la referència.



Veiem cosetes i fem els primers descobriments.

Un dels dos cables de +12V és simplement corrent, no té dades.

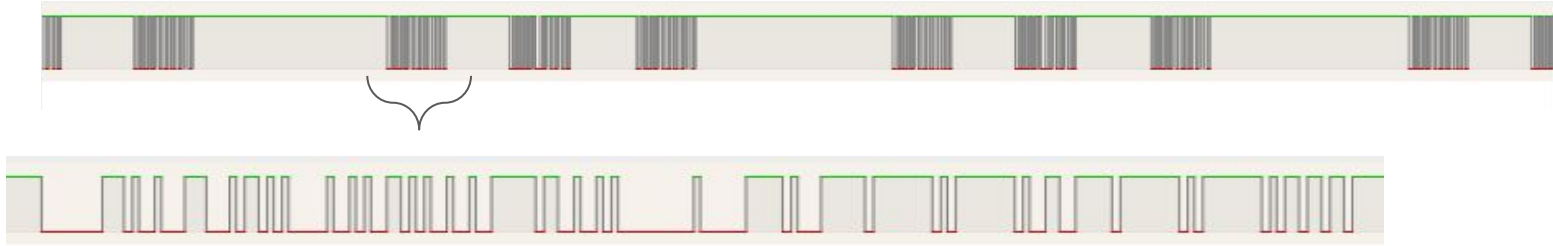
Estructura general de la senyal



Al cable que té dades, veiem grups de paquetets separats per silencis.

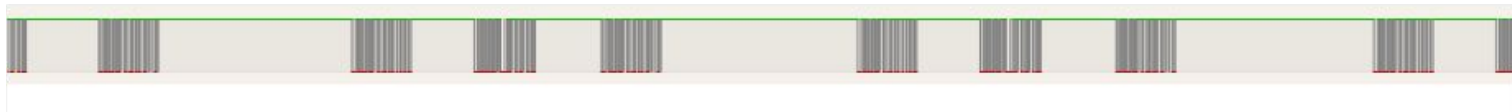
Cada 3 paquets hi ha un silenci més llarg.

Estructura general de la senyal



Si fem zoom a un dels paquets, veiem que la senyal és binària.

Estructura general de la senyal



S'envien dades tota l'estona, encara que no faci cap canvi al controlador.

També amb l'aire apagat: la senyal d'on/off és digital.

Només hi ha un cable de dades -> la comunicació és unidireccional (??)

Comunicació bidireccional

Si desconnectem la màquina de l'aire del cable de dades, de tal manera que al Sigrok només veiem les dades que envia el controlador, desapareix el 2n paquet de cada grup de 3.



Comunicació bidireccional

Si connectem la màquina però desconnectem el controlador, desapareixen el 1r i el 3r paquets!



Comunicació bidireccional

Així doncs, deduim:

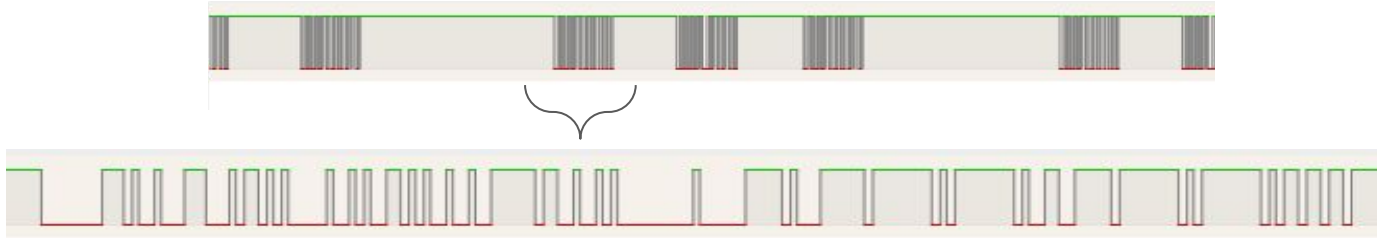


I així en constant repetició.

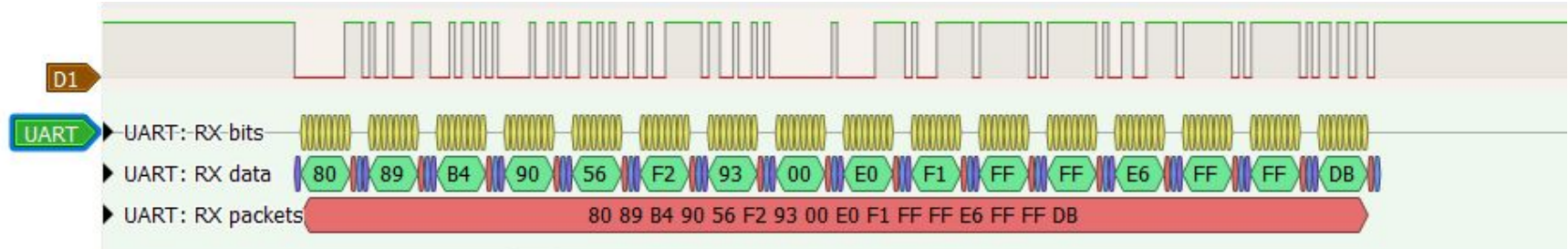
Descodificant la senyal

Per poder entendre la comunicació entre els dos aparells, hem de traduir la senyal elèctrica a bits i bytes.

Centrem-nos en un dels paquets...



Decodificant la senyal

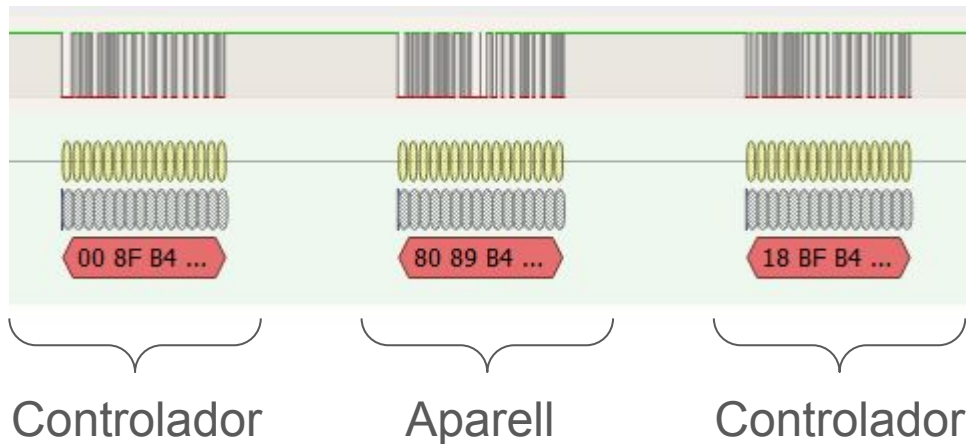


Tenim sort i el Sigrok sap decodificar el protocol que fan servir:

- UART a 1200 bauds
- Amb bit de paritat senar cada 8 bits
- Agrupats en paquets de 16 bytes

Descodificant la senyal

Així podem descodificar cadascún dels paquets:



Agafem-ne una mostra en format hexadecimal i anem a un editor de text!

00 8F B4 90 50 F0 90 FF F0 FF 3F FF FF FF FF CC
80 89 B4 90 50 F0 90 00 E0 F0 FF FF E4 FF FF CD
29 BF B4 90 50 F0 90 FF F0 FF 3F FF F1 F7 FF 0F

00 8F B4 90 50 F0 90 FF F0 FF 3F FF FF FF FF CC
80 89 B4 90 50 F0 90 00 E0 F0 FF FF E4 FF FF CD
2A 8F B4 90 50 F0 90 FF F0 FF 3F FF F1 F7 FF E0

00 8F B4 90 50 F0 90 FF F0 FF 3F FF FF FF FF CC
80 89 B4 90 50 F0 90 00 E0 F0 FF FF E4 FF FF CD
2A 9F B4 90 50 F0 90 FF F0 FF 3F FF F1 F7 FF F0

00 8F B4 90 50 F0 90 FF F0 FF 3F FF FF FF FF CC
80 89 B4 90 50 F0 90 00 E0 F0 FF FF E4 FF FF CD
2A AF B4 90 50 F0 90 FF F0 FF 3F FF F1 F7 FF 00

00 8F B4 90 50 F0 90 FF F0 FF 3F FF FF FF FF CC
80 89 B4 90 50 F0 90 00 E0 F0 FF FF E4 FF FF CD
2A BF B4 90 50 F0 90 FF F0 FF 3F FF F1 F7 FF 10

00 8F B4 90 50 F0 90 FF F0 FF 3F FF FF FF FF CC
80 89 B4 90 50 F0 90 00 E0 F0 FF FF E4 FF FF CD
2B 8F B4 90 50 F0 90 FF F0 FF 3F FF F1 F7 FF E1

00 8F B4 90 50 F0 90 FF F0 FF 3F FF FF FF FF CC
80 89 B4 90 50 F0 90 00 E0 F0 FF FF E4 FF FF CD
2B 9F B4 90 50 F0 90 FF F0 FF 3F FF F1 F7 FF F1


```
00 8F B4 90 50 F0 90 FF F0 FF 3F FF FF FF FF CC
80 89 B4 90 50 F0 90 00 E0 F0 FF FF E4 FF FF CD
29 BF B4 90 50 F0 90 FF F0 FF 3F FF F1 F7 FF 0F
```

```
00 8F B4 90 50 F0 90 FF F0 FF 3F FF FF FF FF CC
80 89 B4 90 50 F0 90 00 E0 F0 FF FF E4 FF FF CD
2A 8F B4 90 50 F0 90 FF F0 FF 3F FF F1 F7 FF E0
```

```
00 8F B4 90 50 F0 90 FF F0 FF 3F FF FF FF FF CC
80 89 B4 90 50 F0 90 00 E0 F0 FF FF E4 FF FF CD
2A 9F B4 90 50 F0 90 FF F0 FF 3F FF F1 F7 FF F0
```

```
00 8F B4 90 50 F0 90 FF F0 FF 3F FF FF FF FF CC
80 89 B4 90 50 F0 90 00 E0 F0 FF FF E4 FF FF CD
2A AF B4 90 50 F0 90 FF F0 FF 3F FF F1 F7 FF 00
```

```
00 8F B4 90 50 F0 90 FF F0 FF 3F FF FF FF FF CC
80 89 B4 90 50 F0 90 00 E0 F0 FF FF E4 FF FF CD
2A BF B4 90 50 F0 90 FF F0 FF 3F FF F1 F7 FF 10
```

```
00 8F B4 90 50 F0 90 FF F0 FF 3F FF FF FF FF CC
80 89 B4 90 50 F0 90 00 E0 F0 FF FF E4 FF FF CD
2B 8F B4 90 50 F0 90 FF F0 FF 3F FF F1 F7 FF E1
```

```
00 8F B4 90 50 F0 90 FF F0 FF 3F FF FF FF FF CC
80 89 B4 90 50 F0 90 00 E0 F0 FF FF E4 FF FF CD
2B 9F B4 90 50 F0 90 FF F0 FF 3F FF F1 F7 FF F1
```

Si descodifiquem uns quants grups de 3 paquets, veiem que entre un grup i el següent hi ha pocs canvis.

```
00 8F B4 90 50 F0 90 FF F0 FF 3F FF FF FF FF CC
80 89 B4 90 50 F0 90 00 E0 F0 FF FF E4 FF FF CD
29 BF B4 90 50 F0 90 FF F0 FF 3F FF F1 F7 FF 0F
```

```
00 8F B4 90 50 F0 90 FF F0 FF 3F FF FF FF FF CC
80 89 B4 90 50 F0 90 00 E0 F0 FF FF E4 FF FF CD
2A 8F B4 90 50 F0 90 FF F0 FF 3F FF F1 F7 FF E0
```

```
00 8F B4 90 50 F0 90 FF F0 FF 3F FF FF FF FF CC
80 89 B4 90 50 F0 90 00 E0 F0 FF FF E4 FF FF CD
2A 9F B4 90 50 F0 90 FF F0 FF 3F FF F1 F7 FF F0
```

```
00 8F B4 90 50 F0 90 FF F0 FF 3F FF FF FF FF CC
80 89 B4 90 50 F0 90 00 E0 F0 FF FF E4 FF FF CD
2A AF B4 90 50 F0 90 FF F0 FF 3F FF F1 F7 FF 00
```

```
00 8F B4 90 50 F0 90 FF F0 FF 3F FF FF FF FF CC
80 89 B4 90 50 F0 90 00 E0 F0 FF FF E4 FF FF CD
2A BF B4 90 50 F0 90 FF F0 FF 3F FF F1 F7 FF 10
```

```
00 8F B4 90 50 F0 90 FF F0 FF 3F FF FF FF FF CC
80 89 B4 90 50 F0 90 00 E0 F0 FF FF E4 FF FF CD
2B 8F B4 90 50 F0 90 FF F0 FF 3F FF F1 F7 FF E1
```

```
00 8F B4 90 50 F0 90 FF F0 FF 3F FF FF FF FF CC
80 89 B4 90 50 F0 90 00 E0 F0 FF FF E4 FF FF CD
2B 9F B4 90 50 F0 90 FF F0 FF 3F FF F1 F7 FF F1
```

Si descodifiquem uns quants grups de 3 paquets, veiem que entre un grup i el següent hi ha pocs canvis.

Els primers 2 bytes de cada 3r paquet canvien en seqüència ascendent, segurament són una ID.

```
00 8F B4 90 50 F0 90 FF F0 FF 3F FF FF FF FF CC
80 89 B4 90 50 F0 90 00 E0 F0 FF FF E4 FF FF CD
29 BF B4 90 50 F0 90 FF F0 FF 3F FF F1 F7 FF 0F
```

```
00 8F B4 90 50 F0 90 FF F0 FF 3F FF FF FF FF CC
80 89 B4 90 50 F0 90 00 E0 F0 FF FF E4 FF FF CD
2A 8F B4 90 50 F0 90 FF F0 FF 3F FF F1 F7 FF E0
```

```
00 8F B4 90 50 F0 90 FF F0 FF 3F FF FF FF FF CC
80 89 B4 90 50 F0 90 00 E0 F0 FF FF E4 FF FF CD
2A 9F B4 90 50 F0 90 FF F0 FF 3F FF F1 F7 FF F0
```

```
00 8F B4 90 50 F0 90 FF F0 FF 3F FF FF FF FF CC
80 89 B4 90 50 F0 90 00 E0 F0 FF FF E4 FF FF CD
2A AF B4 90 50 F0 90 FF F0 FF 3F FF F1 F7 FF 00
```

```
00 8F B4 90 50 F0 90 FF F0 FF 3F FF FF FF FF CC
80 89 B4 90 50 F0 90 00 E0 F0 FF FF E4 FF FF CD
2A BF B4 90 50 F0 90 FF F0 FF 3F FF F1 F7 FF 10
```

```
00 8F B4 90 50 F0 90 FF F0 FF 3F FF FF FF FF CC
80 89 B4 90 50 F0 90 00 E0 F0 FF FF E4 FF FF CD
2B 8F B4 90 50 F0 90 FF F0 FF 3F FF F1 F7 FF E1
```

```
00 8F B4 90 50 F0 90 FF F0 FF 3F FF FF FF FF CC
80 89 B4 90 50 F0 90 00 E0 F0 FF FF E4 FF FF CD
2B 9F B4 90 50 F0 90 FF F0 FF 3F FF F1 F7 FF F1
```

Si descodifiquem uns quants grups de 3 paquets, veiem que entre un grup i el següent hi ha pocs canvis.

Els primers 2 bytes de cada 3r paquet canvien en seqüència ascendent, segurament són una ID.

L'últim byte té molta més entropia que la resta, segurament és un checksum.

```
00 8F B4 90 50 F0 90 FF F0 FF 3F FF FF FF FF CC
80 89 B4 90 50 F0 90 00 E0 F0 FF FF E4 FF FF CD
29 BF B4 90 50 F0 90 FF F0 FF 3F FF F1 F7 FF 0F
```

```
00 8F B4 90 50 F0 90 FF F0 FF 3F FF FF FF FF CC
80 89 B4 90 50 F0 90 00 E0 F0 FF FF E4 FF FF CD
2A 8F B4 90 50 F0 90 FF F0 FF 3F FF F1 F7 FF E0
```

```
00 8F B4 90 50 F0 90 FF F0 FF 3F FF FF FF FF CC
80 89 B4 90 50 F0 90 00 E0 F0 FF FF E4 FF FF CD
2A 9F B4 90 50 F0 90 FF F0 FF 3F FF F1 F7 FF F0
```

```
00 8F B4 90 50 F0 90 FF F0 FF 3F FF FF FF FF CC
80 89 B4 90 50 F0 90 00 E0 F0 FF FF E4 FF FF CD
2A AF B4 90 50 F0 90 FF F0 FF 3F FF F1 F7 FF 00
```

```
00 8F B4 90 50 F0 90 FF F0 FF 3F FF FF FF FF CC
80 89 B4 90 50 F0 90 00 E0 F0 FF FF E4 FF FF CD
2A BF B4 90 50 F0 90 FF F0 FF 3F FF F1 F7 FF 10
```

```
00 8F B4 90 50 F0 90 FF F0 FF 3F FF FF FF FF CC
80 89 B4 90 50 F0 90 00 E0 F0 FF FF E4 FF FF CD
2B 8F B4 90 50 F0 90 FF F0 FF 3F FF F1 F7 FF E1
```

```
00 8F B4 90 50 F0 90 FF F0 FF 3F FF FF FF FF CC
80 89 B4 90 50 F0 90 00 E0 F0 FF FF E4 FF FF CD
2B 9F B4 90 50 F0 90 FF F0 FF 3F FF F1 F7 FF F1
```

Si descodifiquem uns quants grups de 3 paquets, veiem que entre un grup i el següent hi ha pocs canvis.

Els primers 2 bytes de cada 3r paquet canvien en seqüència ascendent, segurament són una ID.

L'últim byte té molta més entropia que la resta, segurament és un checksum.

- En efecte, fent servir eines online que calculen els checksums més comuns veiem que és una suma de bytes mòdul 256.


```
00 8F B4 90 50 F0 90 FF F0 FF 3F FF FF FF FF CC
80 89 B4 90 50 F0 90 00 E0 F0 FF FF E4 FF FF CD
29 BF B4 90 50 F0 90 FF F0 FF 3F FF F1 F7 FF 0F
```

```
00 8F B4 90 50 F0 90 FF F0 FF 3F FF FF FF FF CC
80 89 B4 90 50 F0 90 00 E0 F0 FF FF E4 FF FF CD
2A 8F B4 90 50 F0 90 FF F0 FF 3F FF F1 F7 FF E0
```

```
00 8F B4 90 50 F0 90 FF F0 FF 3F FF FF FF FF CC
80 89 B4 90 50 F0 90 00 E0 F0 FF FF E4 FF FF CD
2A 9F B4 90 50 F0 90 FF F0 FF 3F FF F1 F7 FF F0
```

```
00 8F B4 90 90 F0 90 FF F0 FF 3F FF FF FF FF CC
80 89 B4 90 90 F0 90 00 E0 F0 FF FF E4 FF FF CD
2A AF B4 90 90 F0 90 FF F0 FF 3F FF F1 F7 FF 00
```

```
00 8F B4 90 90 F0 90 FF F0 FF 3F FF FF FF FF CC
80 89 B4 90 90 F0 90 00 E0 F0 FF FF E4 FF FF CD
2A BF B4 90 90 F0 90 FF F0 FF 3F FF F1 F7 FF 10
```

```
00 8F B4 90 90 F0 90 FF F0 FF 3F FF FF FF FF CC
80 89 B4 90 90 F0 90 00 E0 F0 FF FF E4 FF FF CD
2B 8F B4 90 90 F0 90 FF F0 FF 3F FF F1 F7 FF E1
```

```
00 8F B4 90 90 F0 90 FF F0 FF 3F FF FF FF FF CC
80 89 B4 90 90 F0 90 00 E0 F0 FF FF E4 FF FF CD
2B 9F B4 90 90 F0 90 FF F0 FF 3F FF F1 F7 FF F1
```

També podem veure quin efecte té en les dades fer un canvi de temperatura al controlador:

- En un dels dos paquets que envia el controlador, canvia el byte número 4.
- L'aire respón canviant el mateix byte al següent paquet.

00	8F	B4	90	50	F0	90	FF	F0	FF	3F	FF	FF	FF	FF	CC
80	89	B4	90	50	F0	90	00	E0	F0	FF	FF	E4	FF	FF	CD
29	BF	B4	90	50	F0	90	FF	F0	FF	3F	FF	F1	F7	FF	0F
00	8F	B4	90	50	F0	90	FF	F0	FF	3F	FF	FF	FF	FF	CC
80	89	B4	90	50	F0	90	00	E0	F0	FF	FF	E4	FF	FF	CD
2A	8F	B4	90	50	F0	90	FF	F0	FF	3F	FF	F1	F7	FF	E0
00	8F	B4	90	50	F0	90	FF	F0	FF	3F	FF	FF	FF	FF	CC
80	89	B4	90	50	F0	90	00	E0	F0	FF	FF	E4	FF	FF	CD
2A	9F	B4	90	50	F0	90	FF	F0	FF	3F	FF	F1	F7	FF	F0
00	8F	B4	90	90	F0	90	FF	F0	FF	3F	FF	FF	FF	FF	CC
80	89	B4	90	90	F0	90	00	E0	F0	FF	FF	E4	FF	FF	CD
2A	AF	B4	90	90	F0	90	FF	F0	FF	3F	FF	F1	F7	FF	00
00	8F	B4	90	90	F0	90	FF	F0	FF	3F	FF	FF	FF	FF	CC
80	89	B4	90	90	F0	90	00	E0	F0	FF	FF	E4	FF	FF	CD
2A	BF	B4	90	90	F0	90	FF	F0	FF	3F	FF	F1	F7	FF	10
00	8F	B4	90	90	F0	90	FF	F0	FF	3F	FF	FF	FF	FF	CC
80	89	B4	90	90	F0	90	00	E0	F0	FF	FF	E4	FF	FF	CD
2B	8F	B4	90	90	F0	90	FF	F0	FF	3F	FF	F1	F7	FF	E1
00	8F	B4	90	90	F0	90	FF	F0	FF	3F	FF	FF	FF	FF	CC
80	89	B4	90	90	F0	90	00	E0	F0	FF	FF	E4	FF	FF	CD
2B	9F	B4	90	90	F0	90	FF	F0	FF	3F	FF	F1	F7	FF	F1

← canvi temp

També podem veure quin efecte té en les dades fer un canvi de temperatura al controlador:

- En un dels dos paquets que envia el controlador, canvia el byte número 4.
- L'aire respón canviant el mateix byte al següent paquet.

La codificació sembla senzilla.

L'estat s'envia sencer cada cop, encara que no fem canvis.

El Person-in-the-Middle

Si aconseguim modificar aquest estat que el controlador envia, podem donar ordres a la màquina de l'aire.

El Person-in-the-Middle

Si aconseguim modificar aquest estat que el controlador envia, podem donar ordres a la màquina de l'aire.

Si aprofitem que l'estat s'envia constantment i el modifiquem, ens estalviem "inventar" paquets des de zero: partim d'un paquet valid.

- Ens estalviem d'entendre el 100% del paquet, generar les IDs, etc.
- Podem mantenir el controlador de paret i no substituir-lo.

El Person-in-the-Middle

Si aconseguim modificar aquest estat que el controlador envia, podem donar ordres a la màquina de l'aire.

Si aprofitem que l'estat s'envia constantment i el modifiquem, ens estalviem "inventar" paquets des de zero: partim d'un paquet valid.

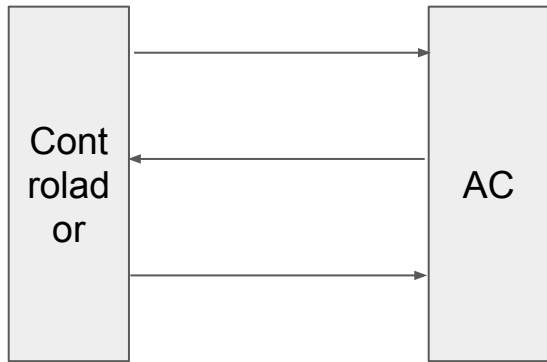
- Ens estalviem d'entendre el 100% del paquet, generar les IDs, etc.
- Podem mantenir el controlador de paret i no substituir-lo.

El Person-in-the-Middle

Per poder modificar els paquets necessitem intervenir la comunicació.

El Person-in-the-Middle

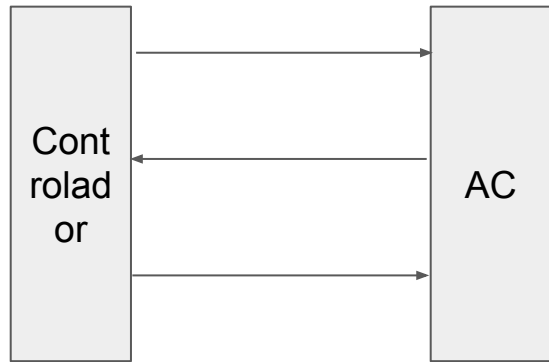
Per poder modificar els paquets necessitem intervenir la comunicació.



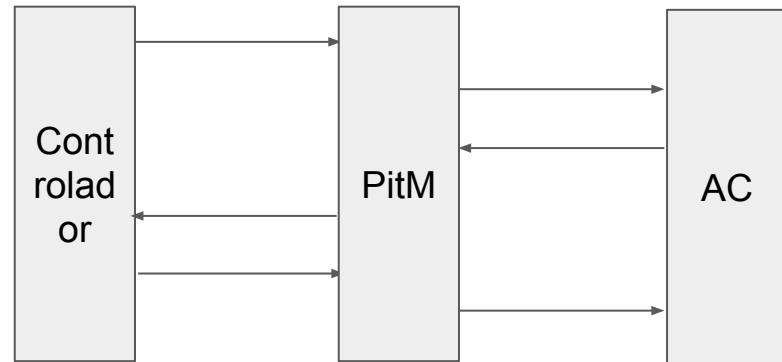
3 paquets abans del PitM

El Person-in-the-Middle

Per poder modificar els paquets necessitem intervenir la comunicació.

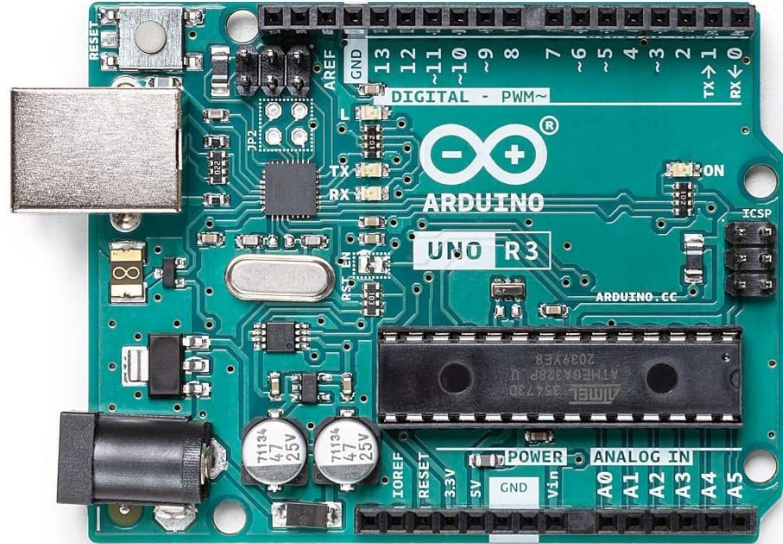


3 paquets abans del PitM



3 paquets després del PitM

L'Arduino

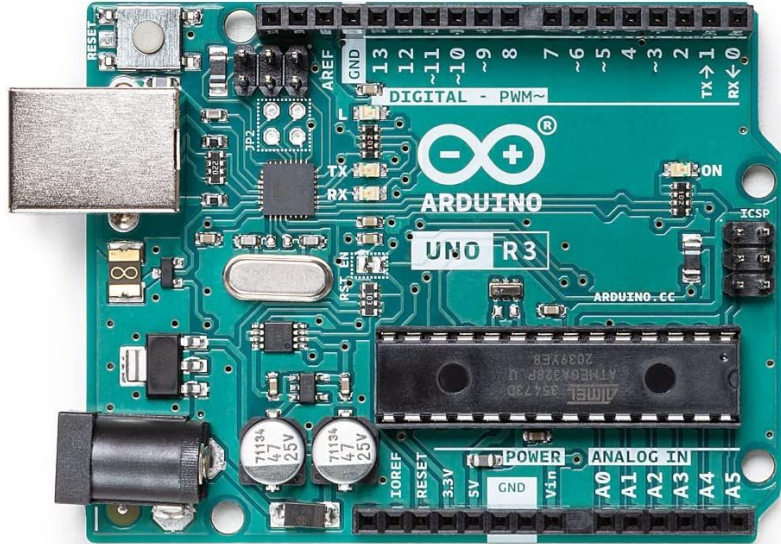


Projecte de software lliure i hardware lliure.

Fa molt fàcil la interacció entre electrònica i software.

<http://ardunio.cc>

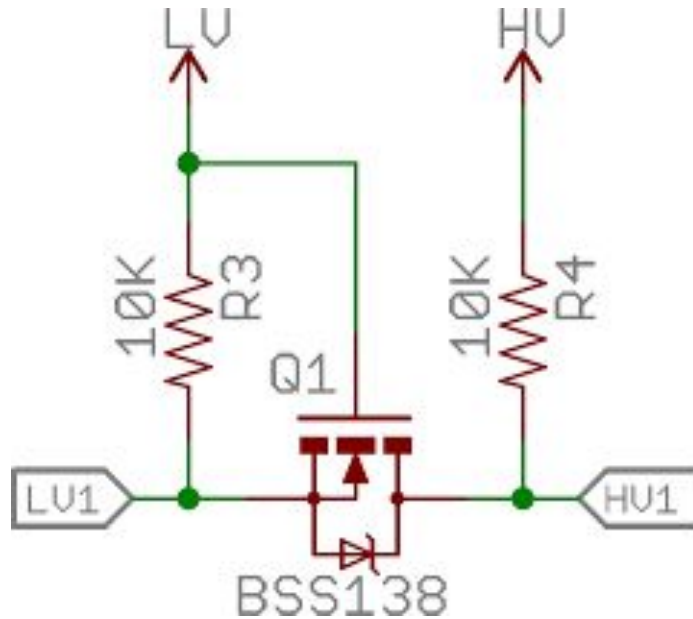
Problemes de nou amb els 12V



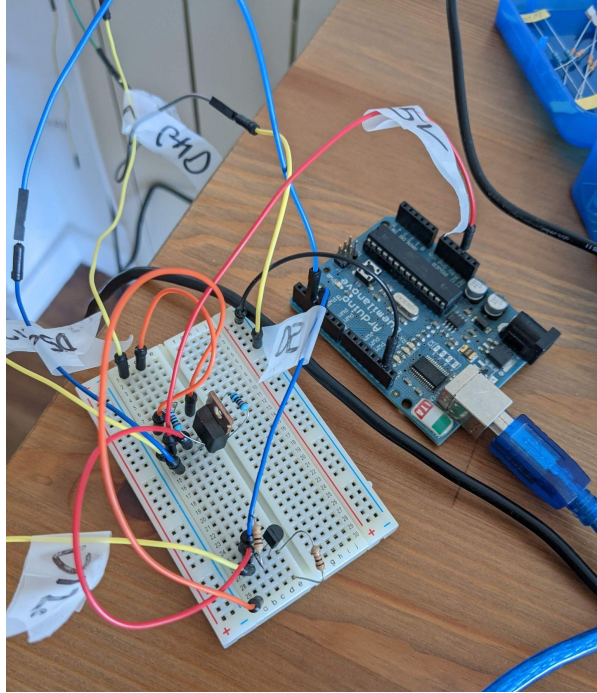
L'arduino "parla" a 5V i no 12V.

El divisor de tensió no és suficient:
Necessitem reduir de 12V a 5V
per a poder llegir la senyal, però
també apujar-la de 5V a 12V quan
la enviem.

El "bidirectional logic level converter"



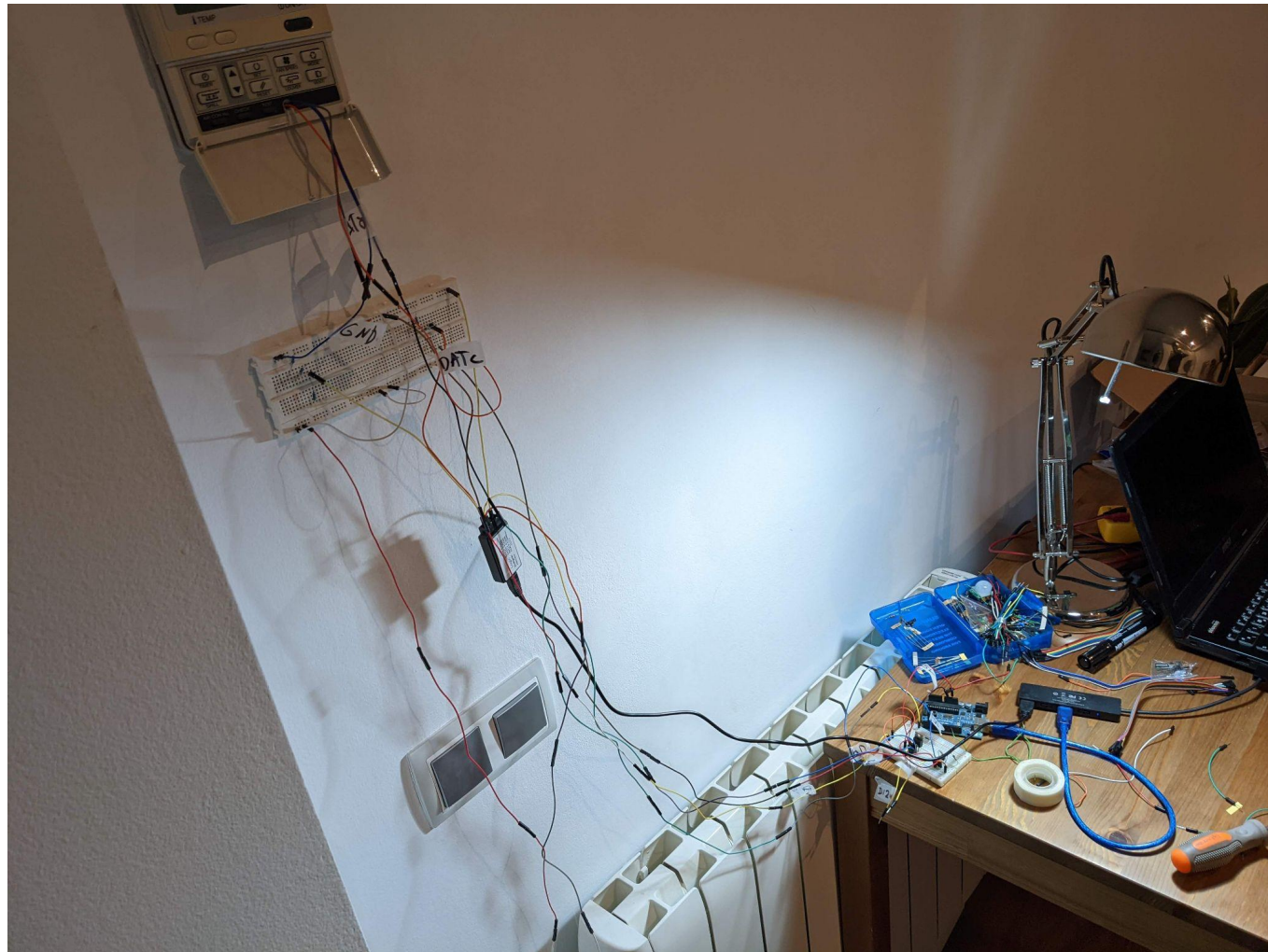
Internet opina que aquest circuit amb un transistor mosfet i dues resistències fa el que volem.



Arduino i amb dos "bidirectional logic level converters".

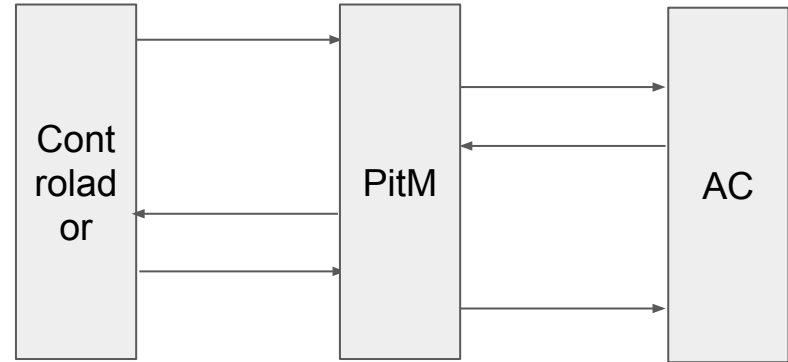
Per una banda, connectat a l'aparell de l'aire. Per l'altra, al controlador.

Ja no hi ha un cable de dades directe entre el controlador i la maquina d'aire.



A programar

Primer objectiu: passar les dades a través nostre i que la màquina i el controlador no s'adonin que som al mig.



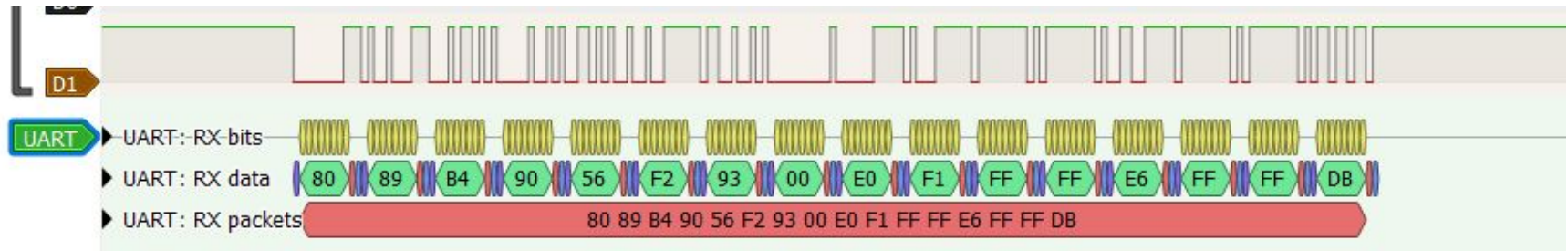
A programar

Primer objectiu: passar les dades a través nostre i que la màquina i el controlador no s'adonin que som al mig.

Hi ha llibreries d'Arduino per a UART, però no ens seveixen:

- Et fan llegir tot un byte abans de poder-lo escriure
- El delay fa enfadar la maquina d'aire

Cal llegir bit a bit per una banda i escriure'l immediatament per l'altra.



A programar

Primer objectiu: passar les dades a través nostre i que la màquina i el controlador no s'adonin que som al mig.

Hi ha llibreries d'Arduino per a UART, però no ens seveixen:

- Et fan llegir tot un byte abans de poder-lo escriure
- El delay fa enfadar la maquina d'aire

Cal llegir bit a bit per una banda i escriure'l immediatament per l'altra.

~200 línies de codi, i l'aire torna a funcionar!

<https://github.com/albertvaka/MHI-WifiController/blob/69b3c11de/MHI-WifiController.ino>

A programar

Segon objectiu: modificar l'estat que passa a través nostre.

El bit d'on/off és senzill de trobar i fàcil comprovar si ha canviat.

A programar

Segon objectiu: modificar l'estat que passa a través nostre.

El bit d'on/off és senzill de trobar i fàcil comprovar si ha canviat.

Però ens cal recalcular els parity bits i el checksum!

```
00 8F B4 90 50 F0 90 FF F0 FF 3F FF FF FF FF CC
```


A programar

Segon objectiu: modificar l'estat que passa a través nostre.

El bit d'on/off és senzill de trobar i fàcil comprovar si ha canviat.

Però ens cal recalcular els parity bits i el checksum!

~260 línies de codi

<https://github.com/albertvaka/MHI-WifiController/blob/e3155bb0c/MHI-WifiController.ino>

A programar

Tercer objectiu: trobar com es representa la resta de l'estat i poder modificar-lo.

2A	8F	B4	90	50	F0	90	FF	F0	FF	3F	FF	F1	F7	FF	E0
00	8F	B4	90	50	F0	90	FF	F0	FF	3F	FF	FF	FF	FF	CC
80	89	B4	90	50	F0	90	00	E0	F0	FF	FF	E4	FF	FF	CD
2A	9F	B4	90	50	F0	90	FF	F0	FF	3F	FF	F1	F7	FF	F0
00	8F	B4	90	90	F0	90	FF	F0	FF	3F	FF	FF	FF	FF	CC
80	89	B4	90	90	F0	90	00	E0	F0	FF	FF	E4	FF	FF	CD
2A	AF	B4	90	90	F0	90	FF	F0	FF	3F	FF	F1	F7	FF	00
00	8F	B4	90	90	F0	90	FF	F0	FF	3F	FF	FF	FF	FF	CC
80	89	B4	90	90	F0	90	00	E0	F0	FF	FF	E4	FF	FF	CD

← canvi temp

A programar

Tercer objectiu: trobar com es representa la resta de l'estat i poder modificar-lo.

Alguns valors són fàcils: velocitat del ventilador, mode, etc.

Hi ha dues temperatures: la desitjada i la actual. Codificades diferent!

A programar

Tercer objectiu: trobar com es representa la resta de l'estat i poder modificar-lo.

Alguns valors són fàcils: velocitat del ventilador, mode, etc.

Hi ha dues temperatures: la desitjada i la actual. Codificades diferent!

- La temperatura desitjada la podem canviar al valor que vulguem

A programar

Tercer objectiu: trobar com es representa la resta de l'estat i poder modificar-lo.

Alguns valors són fàcils: velocitat del ventilador, mode, etc.

Hi ha dues temperatures: la desitjada i la actual. Codificades diferent!

- La temperatura desitjada la podem canviar al valor que vulguem
- La temperatura actual la podem canviar amb un encenedor i mirar el valor.



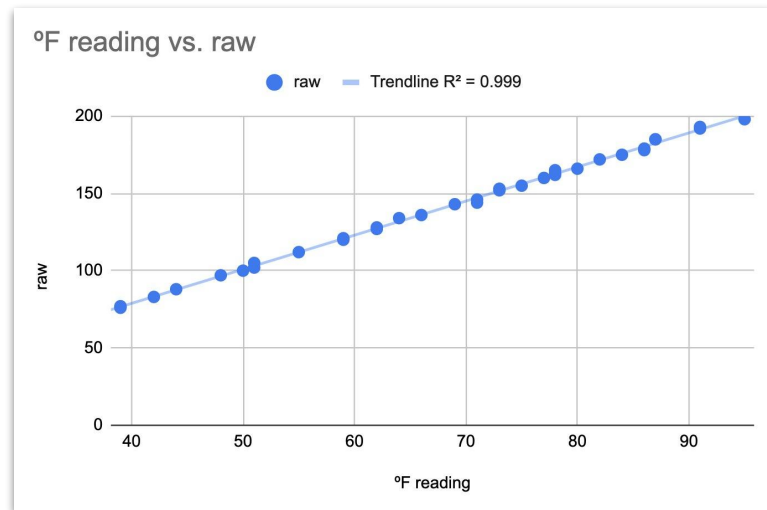
Descodificant la temperatura

Amb alguns punts i una regressió lineal podem trobar la fórmula.

El controlador no mostra els decimals de la temperatura...Si configurem Fahrenheit tenim salts més petits :)

Resultat:

$$F = (0.4527 * \text{raw}) + 4.353$$



Descodificant la temperatura

$$F = (0.4527 * raw) + 4.353$$

És una merda de fórmula. Algú ha programat això?

Descodificant la temperatura

$$F = (0.4527 * raw) + 4.353$$

És una merda de fórmula. Algú ha programat això?

Passem a Celsius:

$$C = (0.2515 * raw) - 15.3594$$

Descodificant la temperatura

$$F = (0.4527 * raw) + 4.353$$

És una merda de fórmula. Algú ha programat això?

Passem a Celsius:

$$C = (0.2515 * raw) - 15.3594$$

Ep! Això s'assembla molt a:

$$C = (0.25 * raw) - 15$$

Descodificant la temperatura

$$F = (0.4527 * raw) + 4.353$$

És una merda de fórmula. Algú ha programat això?

Passem a Celsius:

$$C = (0.2515 * raw) - 15.3594$$

Ep! Això s'assembla molt a:

$$C = (0.25 * raw) - 15$$

És molt més raonable que algú hagi programat això!

- Increments de quarts de grau, a partir de -15 graus.

A programar

Ja podem llegir i escriure qualsevol part de l'estat!

~350 linies de codi

<https://github.com/albertvaka/MHI-WifiController/blob/c4098a604/MHI-WifiController.ino>

A programar

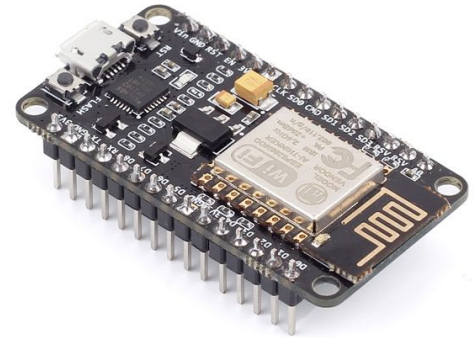
Quart objectiu: connectar-ho a internet.

A programar

Quart objectiu: connectar-ho a internet.

Passem d'Arduino a ESP8266 que té WiFi incorporat.

Pero no és HW lliure :(



A programar

Quart objectiu: connectar-ho a internet.

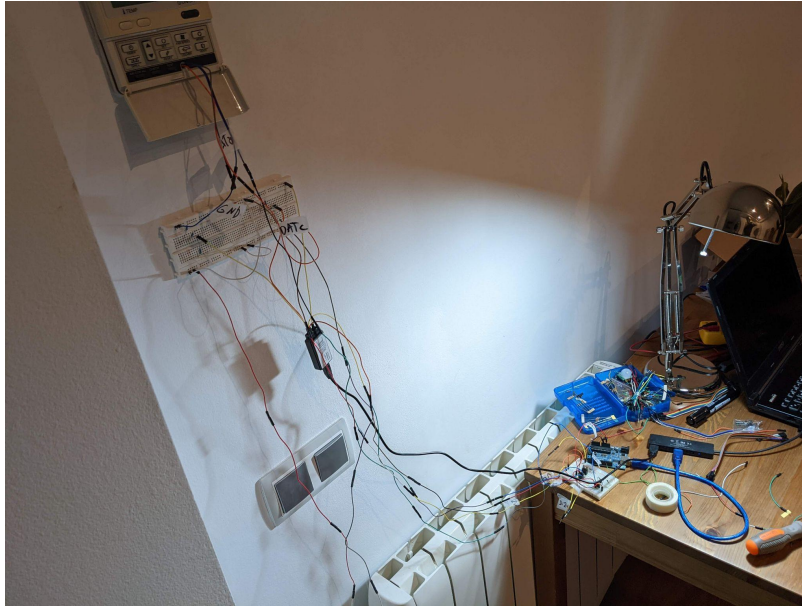
Passem d'Arduino a ESP8266 que té WiFi incorporat.

Per a interactuar-hi des del mòbil amb una app o via Google Home/Alexa/Siri, hi ha diferents serveis i APIs.

Tampoc son lliures :(

~650 lines de codi

Prototip funcionant

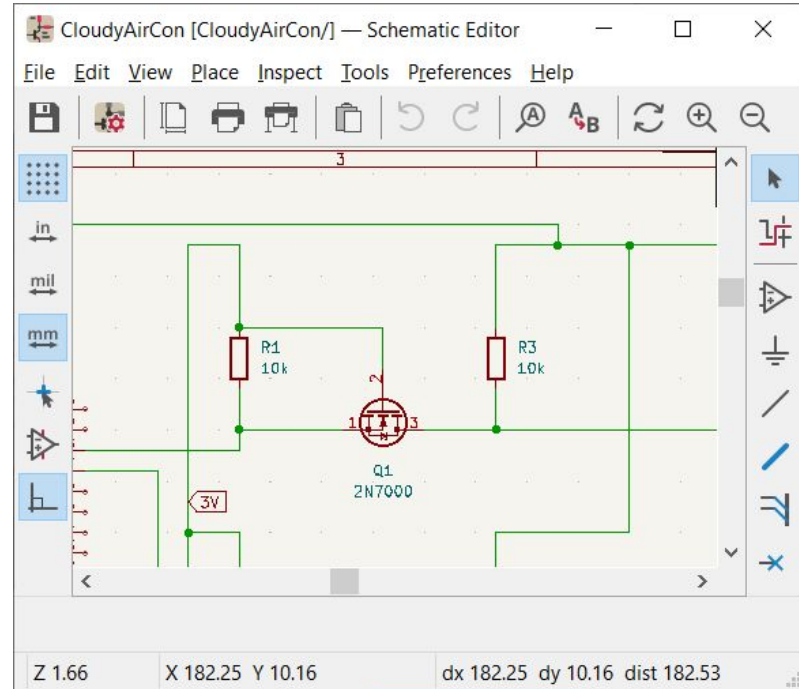


Pero tenir cables penjant al menjador no mola

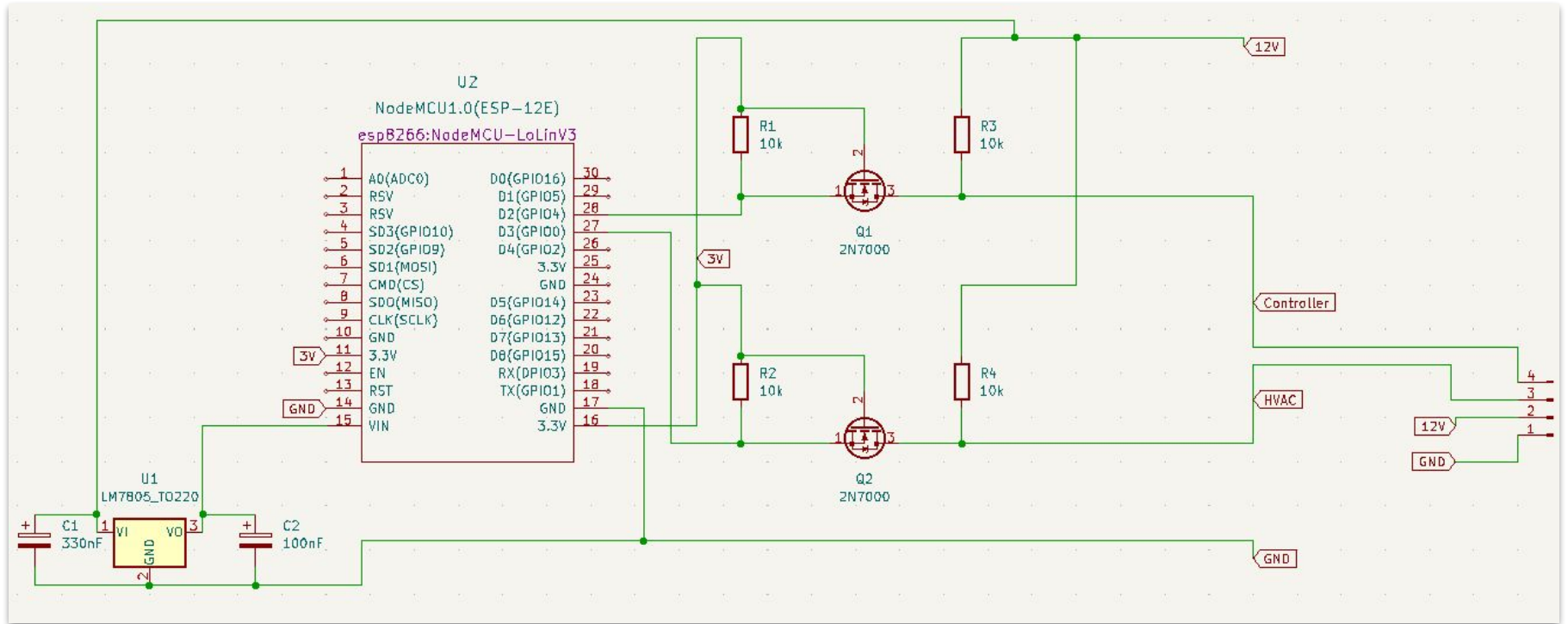
Dissenyant una PCB



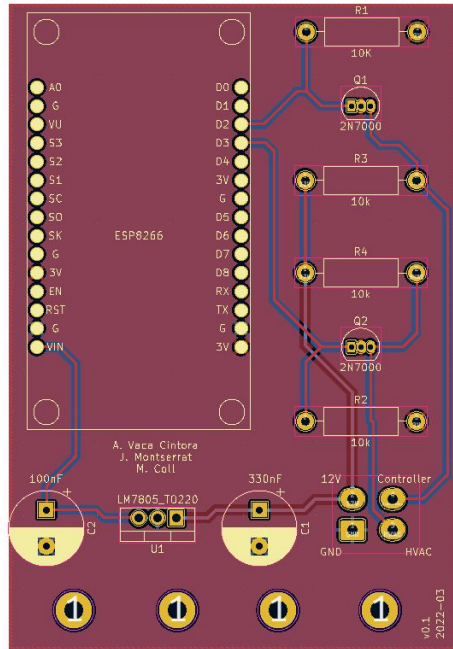
Software lliure per dissenyar esquemes electrònics i la seva conversió a circuits impresos.



Dissenyant una PCB



Dissenyant una PCB

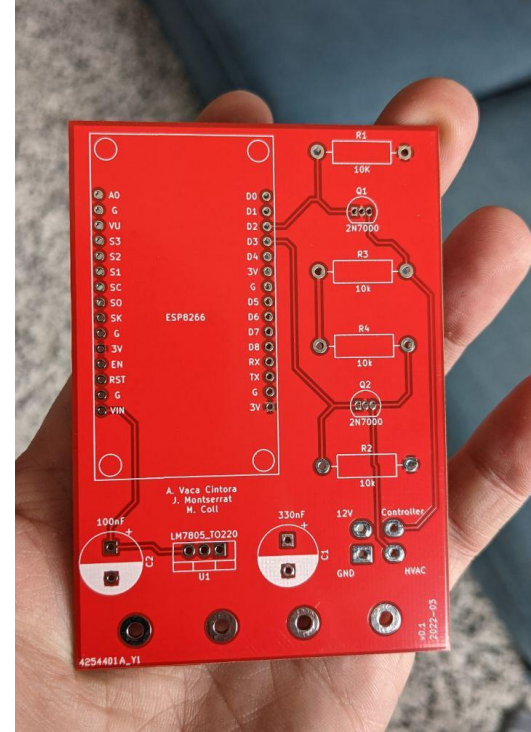
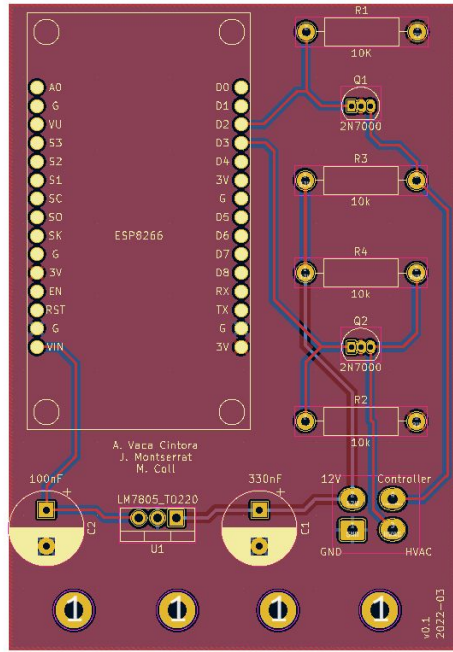


Un cop dissenyada, KiCad permet exportar en format Gerber per enviar a fabricar.

Costa ~10€ fabricar 5 PCBs.

Cal comprar els components per separat i i soldar-los.

Dissenyant una PCB



El resultat final

- Encendre/apagar l'aire i canviar la temperatura des del mòbil
- També per veu amb Google Home
- Històric de la temperatura de casa
- Puc programar canvis de temperatura quan es pon el sol, quan sona el despertador, etc.

