

“...pero, que es esto del **Open Source**?”

Juan C. Inostroza

jci@codemonkey.cl

Twitter : @jci

<https://github.com/jci>



# **Vamos a hablar del Open Source?**

(otra vez?!)

No es una charla teorica.  
Mas bien una charla tecnica.  
Con monton de referencias.

jci@marvin:~> whoami

**Ingeniero Ejecucion Informatica**  
**Trabajando como Sysadmin desde 2000**  
**Conocer Unix/Linux en 1996**  
**Conocer Internet en 1994**  
**Conocer un MOS6502 (Atari!) en 1985**

(yes, I am that old)

```
Sql> select * from jci where jci.status="employed" and  
      jci.datetime < now();
```

(DBA, baby!)

“The competition is getting younger,  
Tougher broncs, You know I can't recall”

– Garth Brooks

Licencias no importan mucho en este momento.  
Aunque debieran importar, ya no es parte de la  
discusion.

Open Source.  
Codigo Abierto.

Source.  
Codigo.

Codigo fuente.  
Accesible.  
Barato.  
Y de calidad.  
A-HO-RA.

Código fuente es la forma de mostrar como algo está hecho.

Mejor que un manual de instrucciones.

“Let me show you how it's done”

# Comunidad de Makers

## Mentalidad de Makers

# a.k.a. Hackers

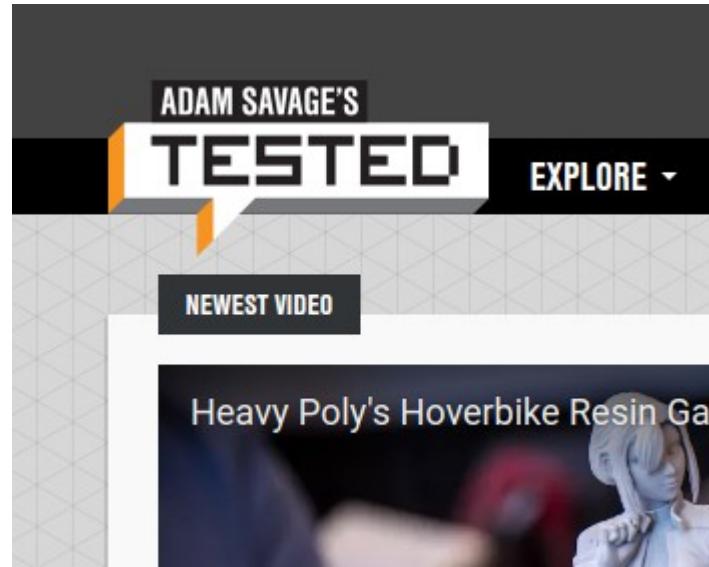
(en el sentido de modificar algo fuera de los parametros de los que  
fue hecho para realizar cualquier otra tarea o bien mejorar lo que  
hace)

<mindset>



Adam Savage.  
Former Mythbuster.

# Tested.com





### Adam Savage's One Day Builds: Cave Bathroom Upgrade

644,658 views

11K 475 SHARE ...

### Up next

AUTOPLAY



Adam Savage's One Day Builds:  
A Better Tape Dispenser!

Tested   
1M views



Adam Savage's One Day Builds:  
Foam Cosplay Sword!

Tested   
469K views



Adam Savage and Vsauce's  
Michael Stevens Build a

Tested   
1.5M views



Adam Savage's One Day Builds:  
Chewbacca and C-3PO!

Tested   
1M views



Adam Savage's One Day Builds:  
Ping Pong Machine Gun!

Tested   
1.9M views



element14

# THE BEN HECK SHOW



Youtube tiene cientos de miles de videos de comunidades de “makers”...

<

>

Aprendan algo fuera de su zona de comodidad.

Idioma. Lenguaje. Tecnologia. Tecnica.

# Código

It's all about the code, baby!

Leer código propio es estar dentro de la zona de comodidad.

Leer código ajeno es salirse de esa zona y aprender como otra persona resuelve un problema particular.

Miles de repos en Github en lenguajes irreconocibles.

Cual es la idea?

Código accesible.

Leible. Copiable. Usable. Para otros.  
O para ustedes mismos.

Publiquen lo que hacen en algun repo publico en Github.  
Puede ser su mejor CV.

<

>



La herramienta correcta para el trabajo correcto.

# Git

# GIT como se debe usar (o al menos como lo uso)

- \$ git init
- \$ git clone
- \$ git checkout
- \$ git pull
- \$ git branch
- \$ git remote
- \$ git push
- \$ git log
- \$ git status
- \$ git diff
- \$ git merge
- \$ git stash
- \$ git clean

Aprender a usar la herramienta correcta para el trabajo correcto.

Código? Git.

O cualquier cosa para control de versiones.

(aunque yo sugiero git)

<

>

<coolstorybro.jpg>

C

<kernighan, ritchie et al>



# Daily routine

- Ruby (the fuck is dis)
- Rails (the fuck is dat)
- Bundle (what kind of black magic is dis)
  - Rack (hell, no!)
  - Npm (node.js? Now? Now? Now?)
- Ruby (what kind of language is dat?!)
- Deployment made for humans?!?!

Fuck this shit.









<Aun debo el documento de instructables para hacer  
esto>



Unemployed two days later.

<sadviolin.mp3>

So...what?!

# Arduino!

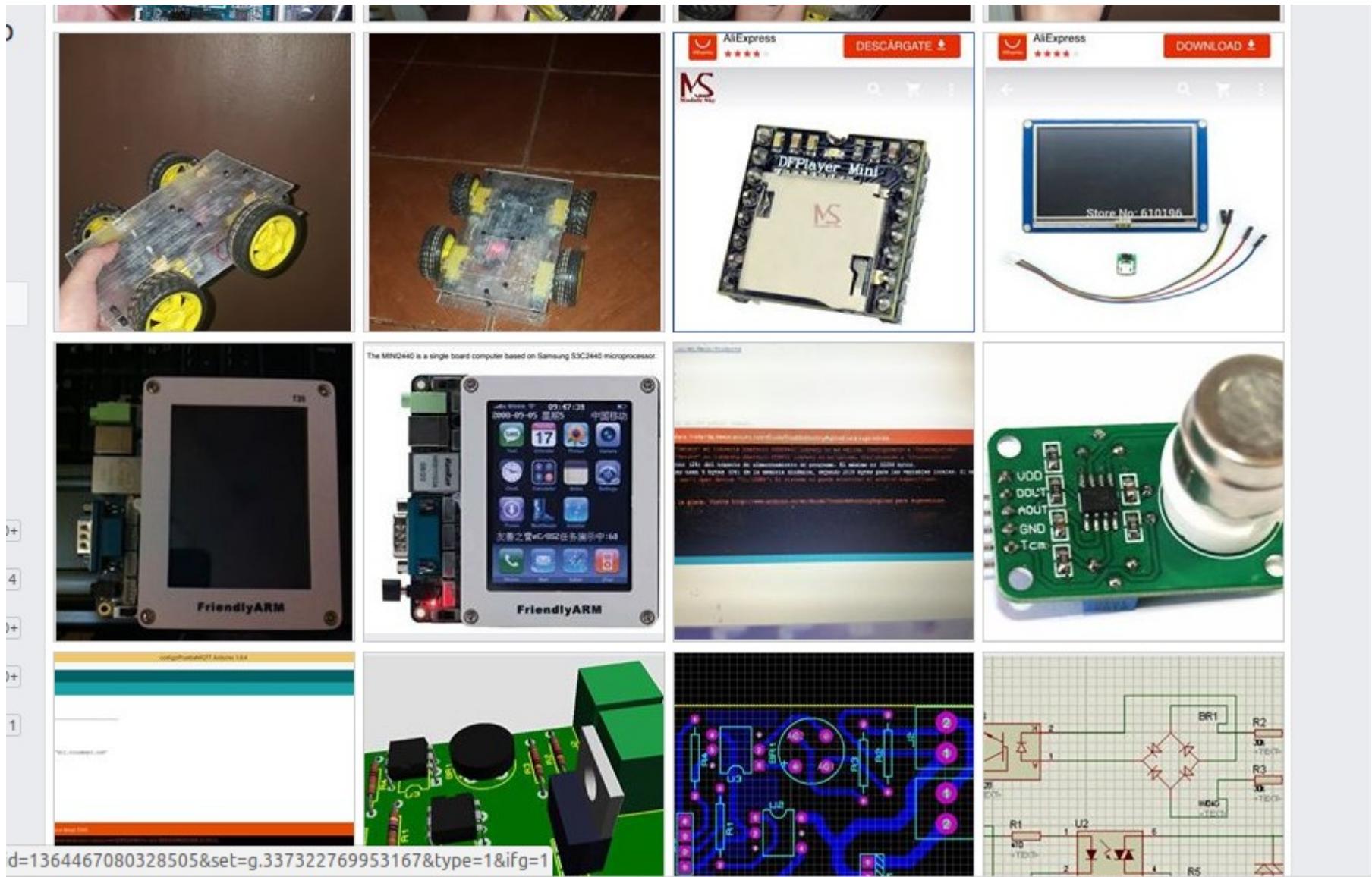
(open source hardware FTW)

## Notas:

- No se mucho de electronica
- No se mucho de piezas de robotica
- No soy muy cuidadoso usando VDC

Arduino es  
Open Source Software

....donde empiezo?



led | Arduino 1.6.9

File Edit Sketch Tools Help

led

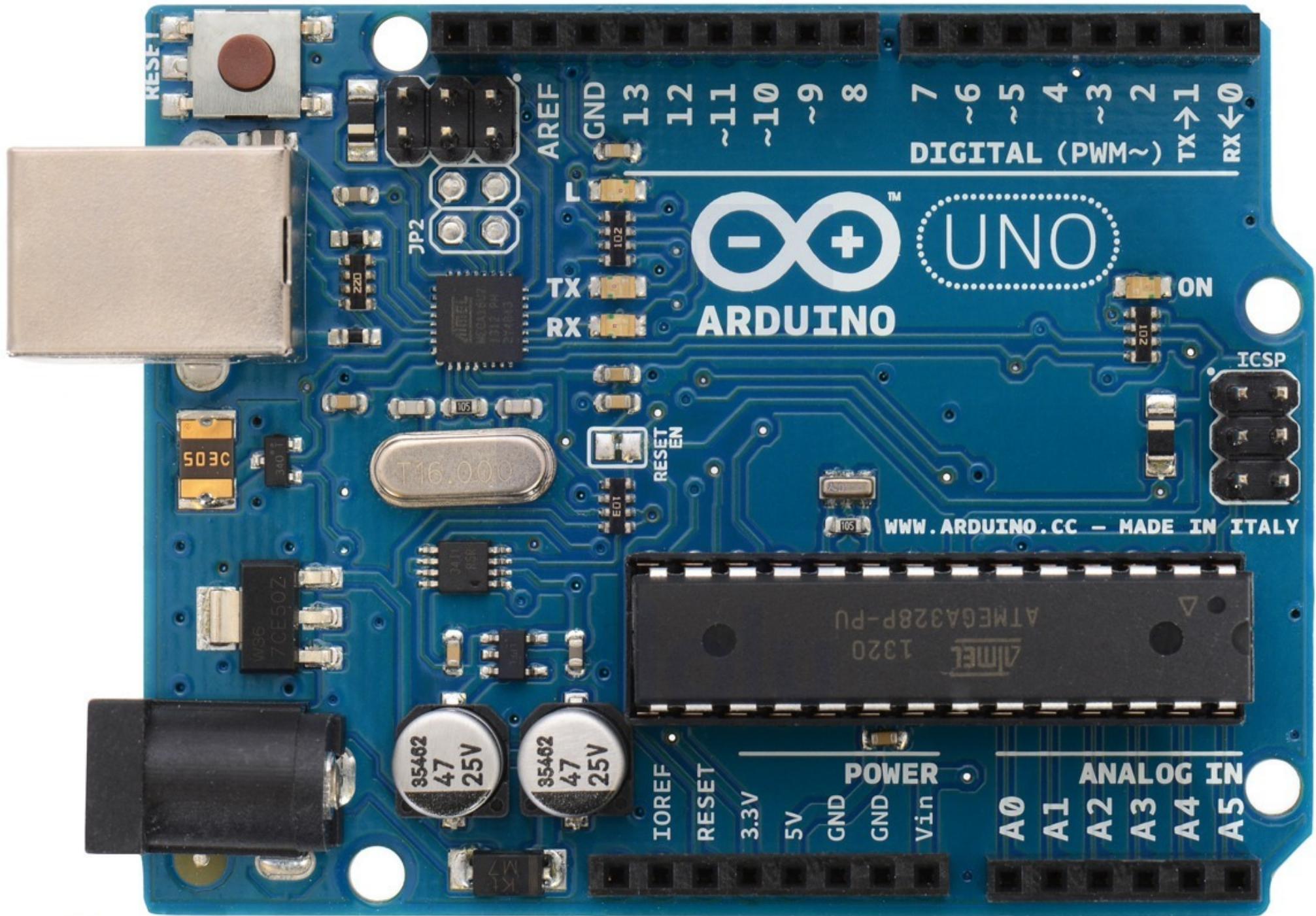
```
8 #define BUTTON_1 A1
9 #define BUTTON_2 A2
10 #define BUTTON_3 A3
11
12 static int LEDS[] = { LED_1, LED_2, LED_3, LED_4 };
13
14 void setup() {
15     // Initialize the digital pin as an output and turn off all LEDs.
16     // Note: For whatever weird reasons, the shield uses HIGH for off.
17     for(int i = 0; i < TOTAL_LEDS; i++)
18     {
19         pinMode(LEDS[i], OUTPUT);
20         digitalWrite(LEDS[i], HIGH);
21     }
22 }
23
24 void loop() {
25     int pause = 200;
26     if (!digitalRead(BUTTON_1)) pause -= 50;
27     if (!digitalRead(BUTTON_2)) pause -= 50;
28     if (!digitalRead(BUTTON_3)) pause -= 50;
29
30     if (!digitalRead(BUTTON_1) || !digitalRead(BUTTON_2) || !digitalRead(BUTTON_3)) {
31         rollLeds(pause);
32     }
}
Done compiling.

Sketch uses 1,368 bytes (4%) of program storage space. Maximum is 32,256 bytes.
Global variables use 17 bytes (0%) of dynamic memory, leaving 2,031 bytes for local variables. Maximum is 16,384 bytes.
```

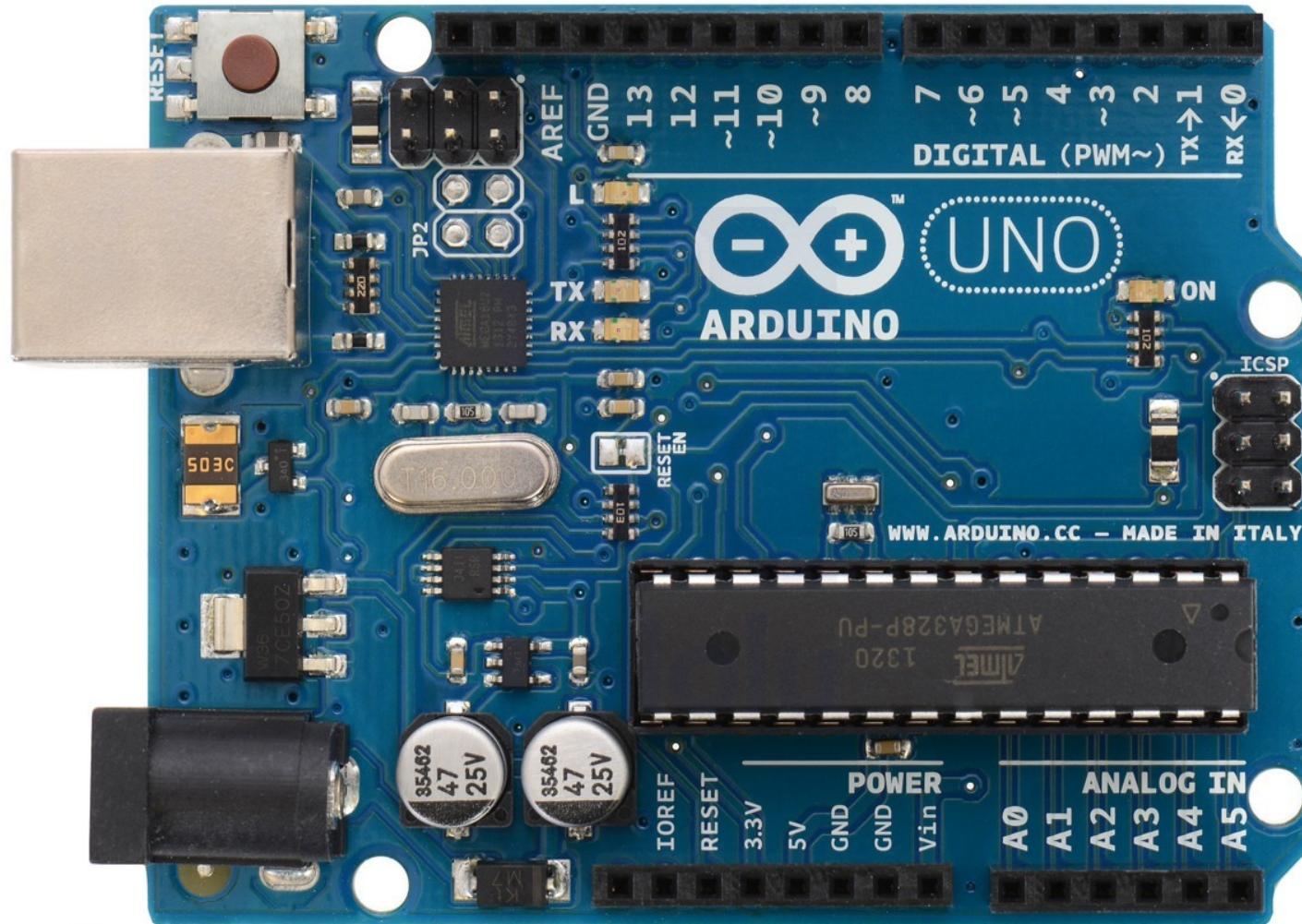
Arduino/Genuino Uno on COM3

<THIS IS C>

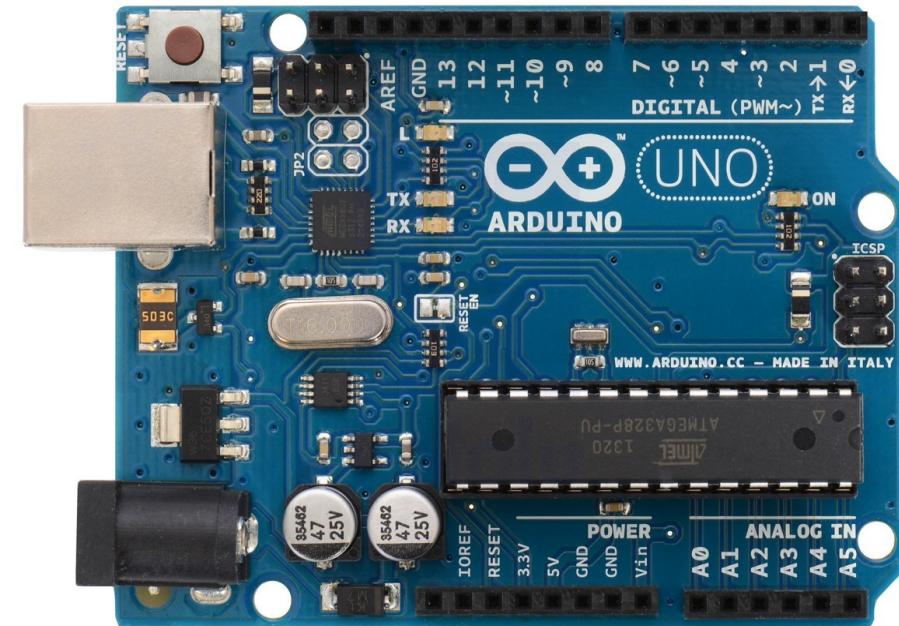




# DIGITAL PINS



# ANALOG PINS



## DIGITAL PINS

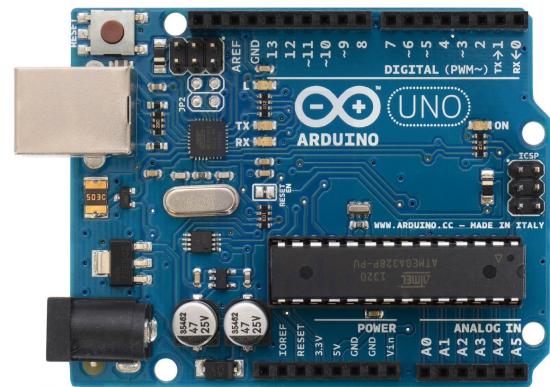
Usables para leer un valor LOW o HIGH

## ANALOG PINS

Usables para leer un valor entre 0v y 5v, devolviendo un valor entre 0 (0v) o 1023 (5v)

## ARDUINO

Manejado por un procesador Atmel AVR de 8 bits. Viene en distintos sabores, como ATmega168, ATmega328, ATmega1280 y ATmega2560



**Los primeros modelos (como UNO) vienen con un puerto USB B hembra. Los nuevos con microUSB. Aunque tambien la alimentacion puede hacerse via transformador externo o por la pata Vin.**

## **ARDUINO**

**Tiene la posibilidad de conectar “shields” o placas para extender funcionalidades, como Ethernet, WiFi o bien protoboards.**

**Las placas pueden ser stackeadas gracias a que usa I2C.**

**Arduino fue la marca registrada original. Debido a limitantes de exportacion de EEUU, los modelos Arduino son solamente distribuidos en EEUU. En el resto del mundo, los modelos por lo general son llamados “Genuino”.**



← “hey, aun soy importante o que? :( ”

## Que es I2C?

**Es una implementacion de bus serial dise~ada para comunicar varios dispositivos de manera master-slave. Es multimaster, multislave. Es usado tipicamente par conectar perifericos con circuitos integrados de baja velocidad, como LCDs. Tambien para conectarlos a poca distancia.**

**Cada uno de los dispositivos cuenta con una direccion de hardware asignada entre 01 (master) y 0xff (el ultimo slave) permitiendo la comunicacion de hasta 127 dispositivos esclavos.**

**Se usan dos canales, SDA (Serial Data) y SCL (Serial Clock). Todos los esclavos van conectados al mismo “bus” en paralelo.**

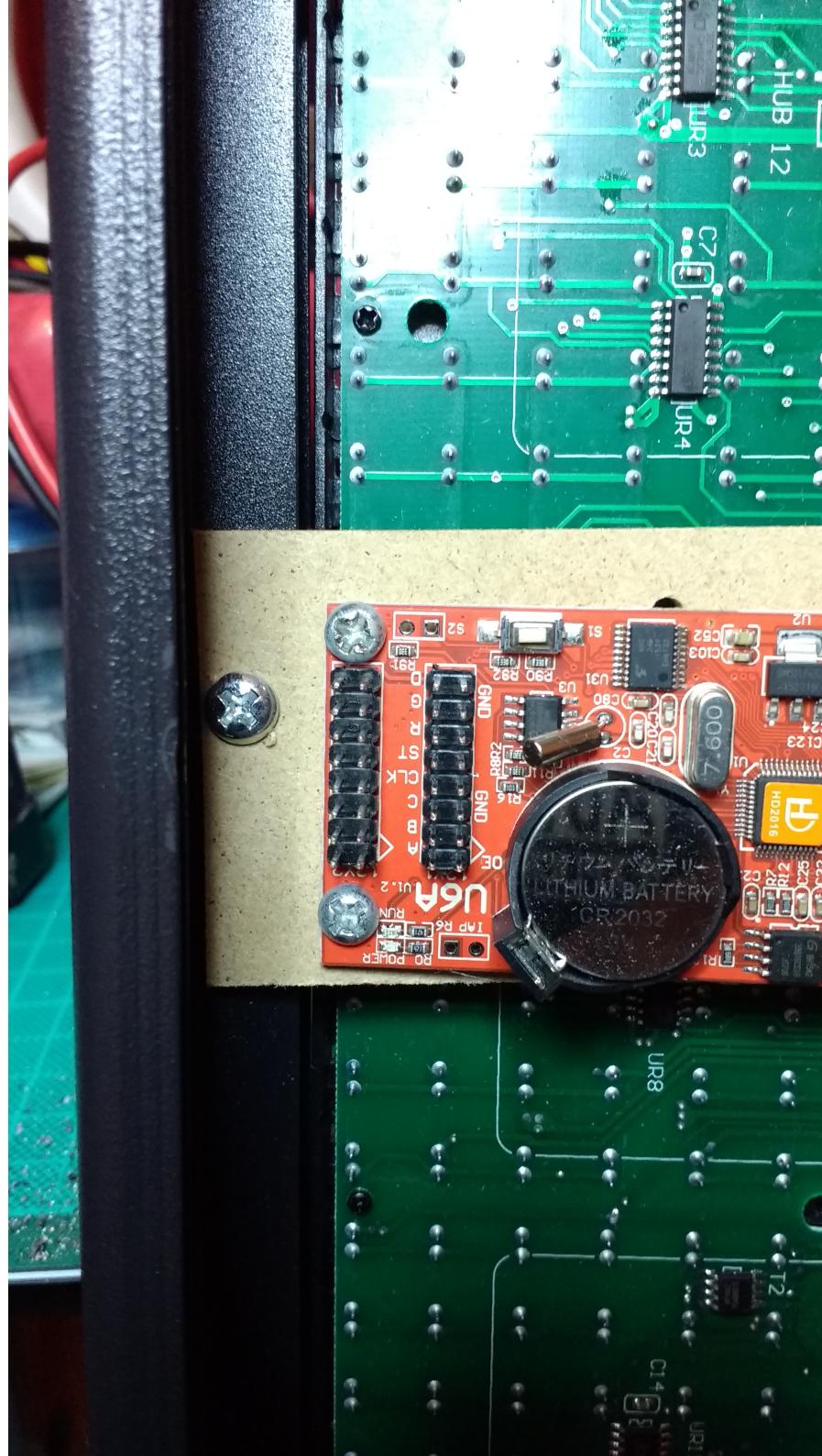
**<el viaje es mas interesante que el destino>**

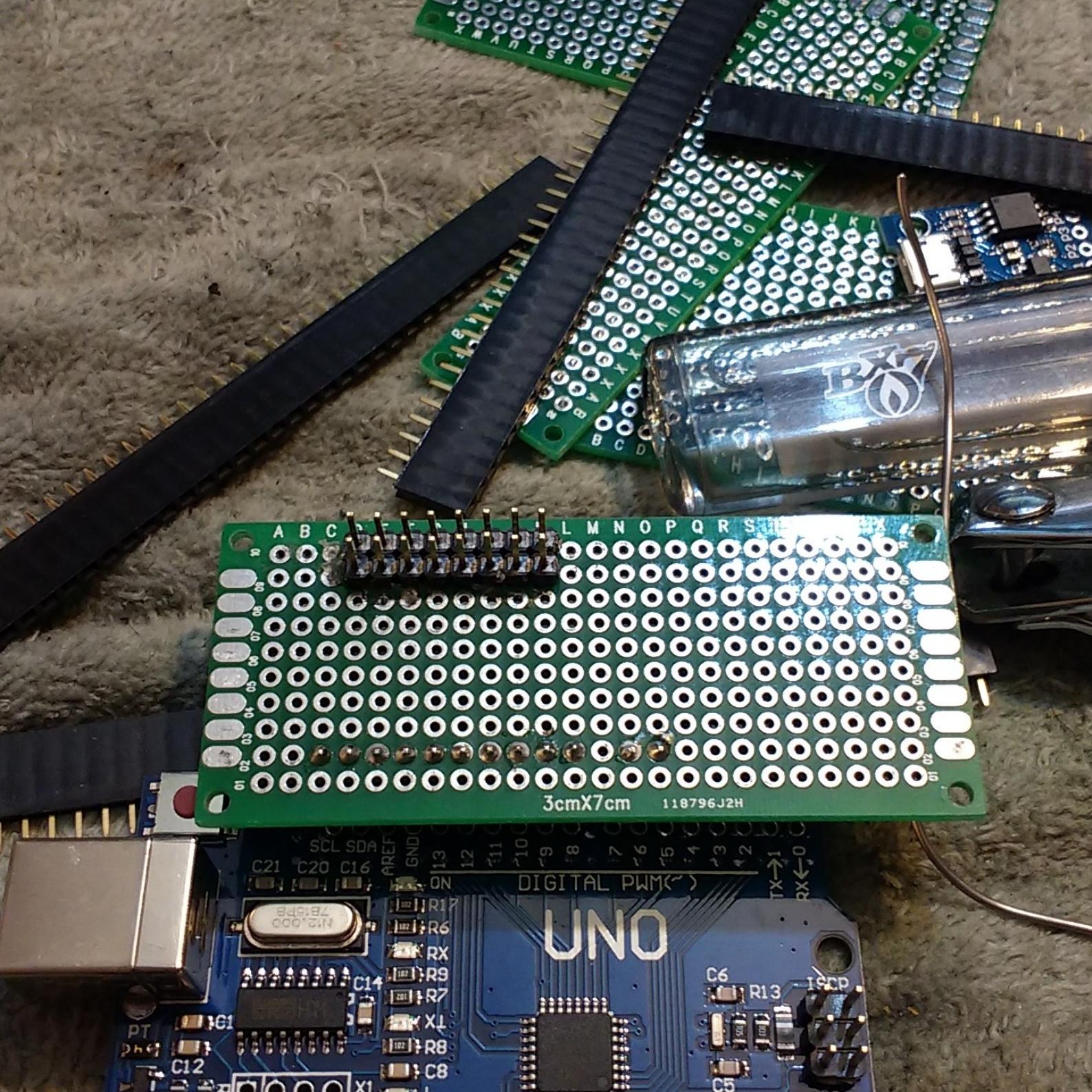


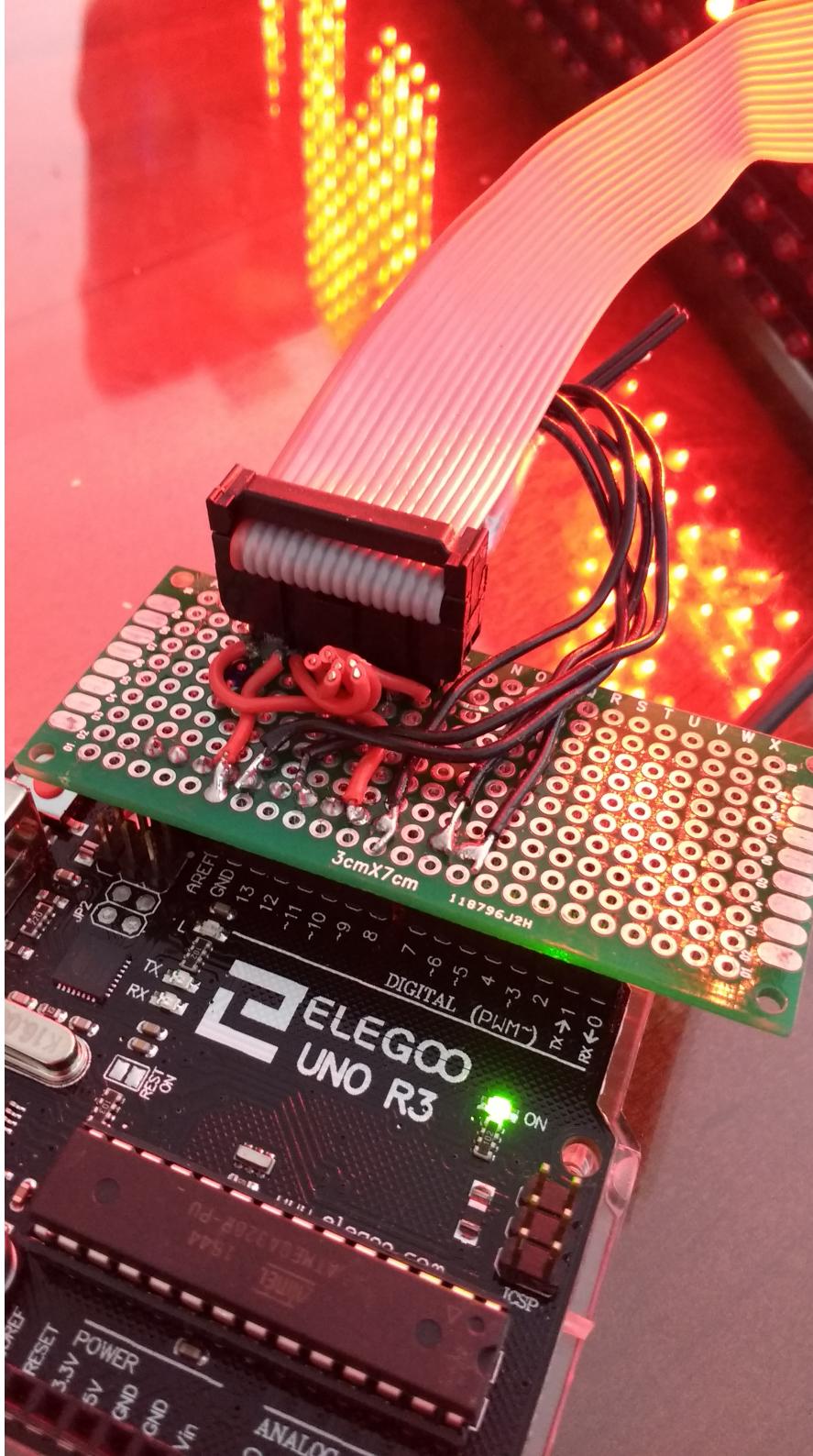
<new cool job a month later>

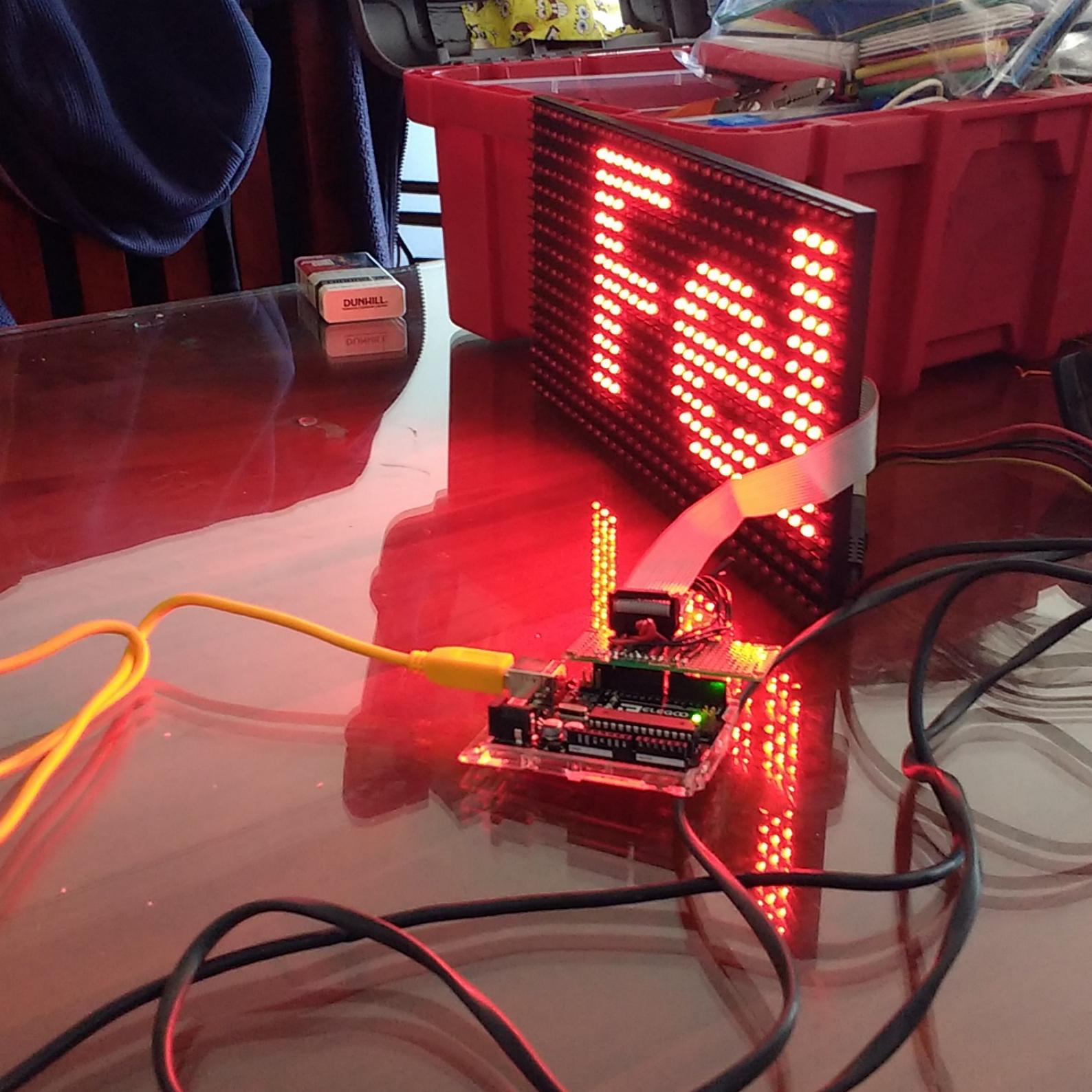
<endstory.txt?>



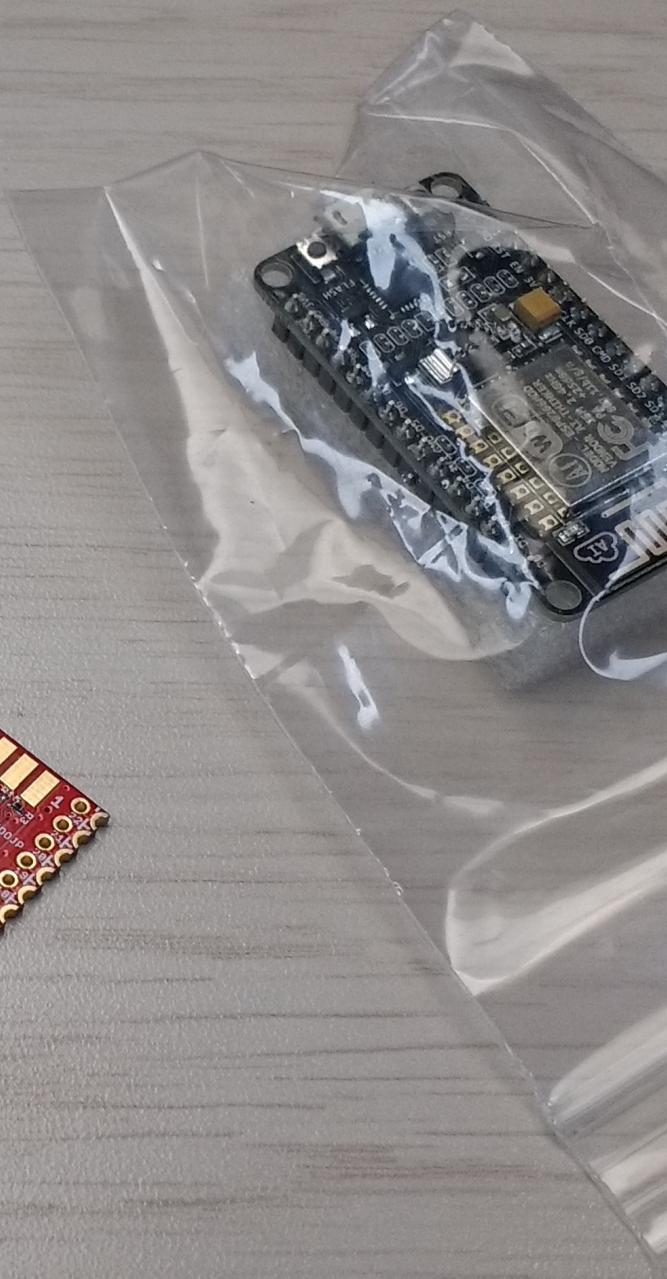












2017-10-14

- light-dmd-row-1-5-9-13  
 - light-dmd-row-2-6-10-14  
 - light-dmd-row-3-7-11-15  
 - light-dmd-row-4-8-12-16

Shift reg → latch then shift  
 HIGH → LOW (PW-DMD-SCLC)

hurtaplexion  
 (4 ciclos de 4MHz) = 16MHz

DMD2

Noe = 9

a = 6

b = 7

SCK = 8

CLK = 13?

r-DATA = 11?

B	A	0	1
0	1, 5, 9, 13	2, 6, 10, 14	2
1	3, 7, 11, 15	4, 8, 12, 16	-
2	-	-	-
3	-	-	-

AB NOE

BA (00) = 1, 5, 9, 13

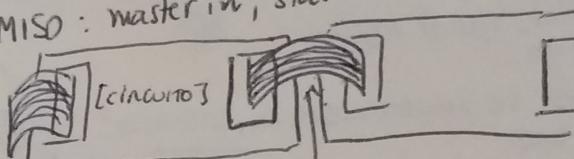
BA (01) = 2, 6, 10, 14

BA (10) = 3, 7, 11, 15

BA (11) = 4, 8, 12, 16

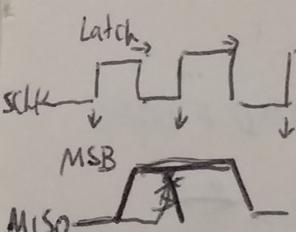
### DMD

- SPI ← for scanning - connection
- Pin 11 ← D11 MOSI (8 Master-slave)
- tiene un clock divider → SPI-CLOCK-DIV
- → 4MHz
- pin 11 está en HIGH → otras patas en gnd
- MISO : master in, slave out



4 filas?  
 (8 bits alto)

daisy chain



cPOL = φ  
 cPHA = φ  
 fast switching (4MHz)



- CPOL : clock POLARITY ← φ → idles at φ

- CPHA : clock PHASE ← half cycle with clock idle  
 half cycle with clock asserted

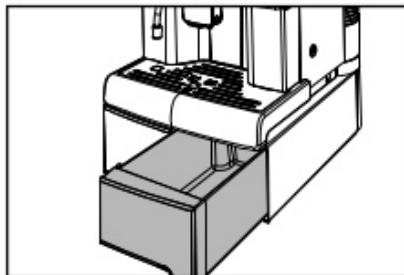


< Conclusiones >

Nada de esto seria minimamente posible con elementos cerrados o de codigo cerrado.

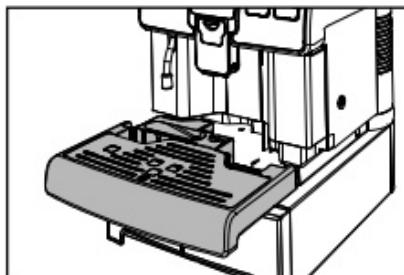
## 5.2 Cleaning the Coffee Grounds Drawer

The coffee grounds drawer must be emptied and cleaned whenever indicated by the machine. To carry out this operation, refer to section 3.7.



## 5.3 Cleaning the Drip Tray

Empty and clean the drip tray every day. Perform this operation also when the float is raised.



## 5.4 Cleaning the Cappuccinatore (After Each Use)

 **Warning!** Failure to clean the Cappuccinatore after use with milk may cause proliferation of bacteria which may be harmful for the user.

The Cappuccinatore must be washed and cleaned every time it is used.

The lack of cleaning before and after each use can affect the correct operation of the machine.

 **Danger of burns!** Dispensing may be preceded by small jets of hot water.

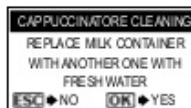
Perform a quick cleaning procedure as follows:

1 Ensure that the Cappuccinatore, complete with all its parts, is correctly inserted (Fig.10, page 3).

2 Insert the suction hose in a carafe containing clean water (Fig.11, page 3).

3 Place a container under the dispensing spout (Fig.12, page 3).

4 With the machine on, press the "CLEAN" button (24).



5 Press the "CUP" button (18) to start the Cappuccinatore cleaning cycle.

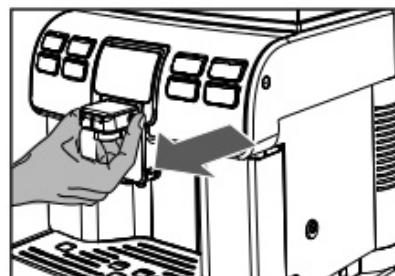
 **Note:** To exit the cycle at any moment, press the "CUP" button (23).

6 The machine heats up the system.



Once the warm-up is complete, the machine performs an assisted wash cycle of the Cappuccinatore. Wait until the cycle is complete.

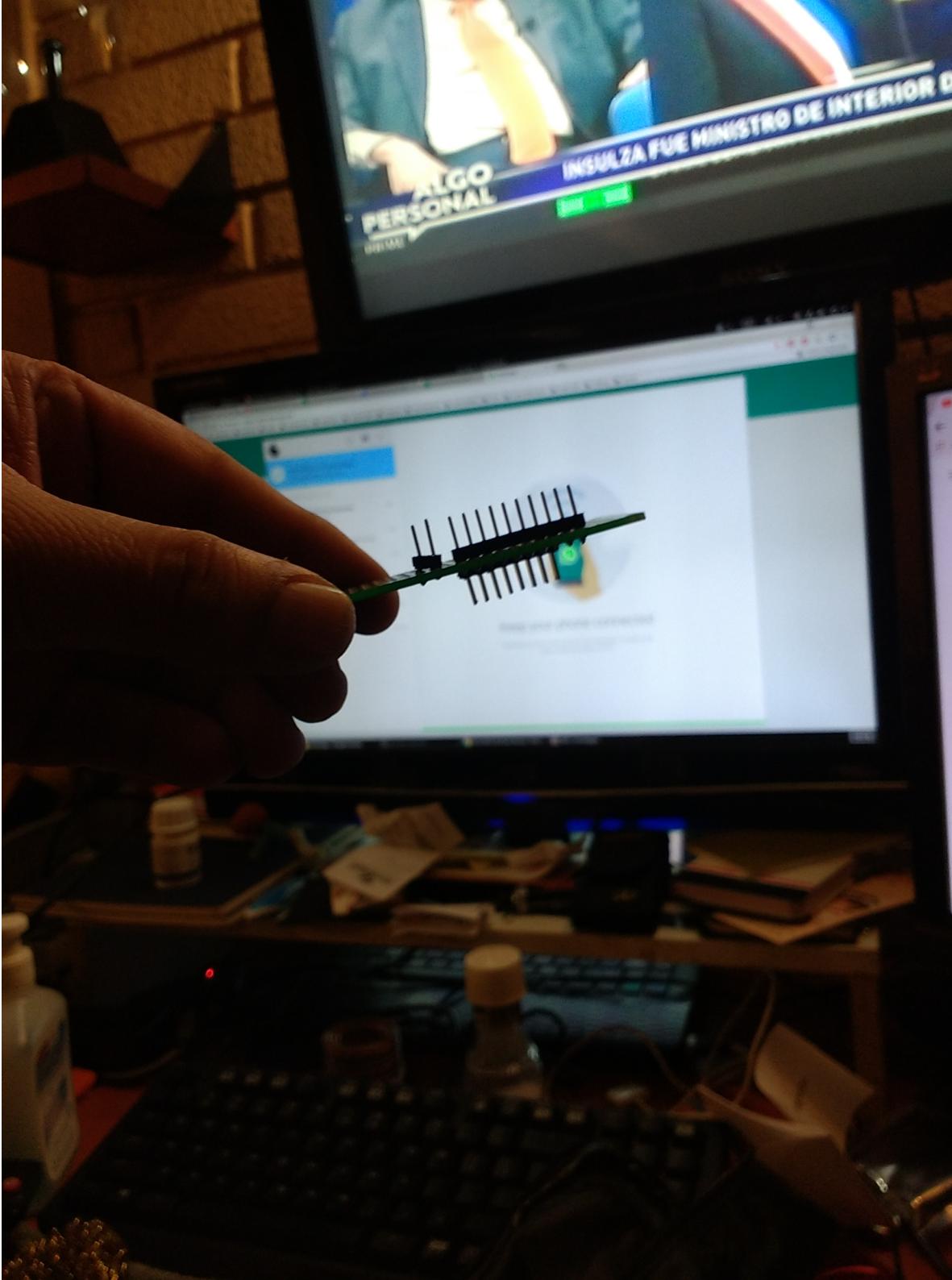
7 Only once the cycle is complete press the side buttons and remove the Cappuccinatore.



0...



Curiosidad como motor.  
Ingenuidad como direccion.  
Ingenieria como construccion.  
Conocimiento como el camino.







<mindset>

**Gracias por su atencion!**

--j

<https://github.com/jci>

Twitter : @jci