

Memoria Puzzle 1 LCD-2004

1- Connexió a internet amb configuracions prèvies

- Per fer la connectar a internet al raspberrypi primer s'ha de instal·lar el sistema operatiu oficial raspberryOS, gravant a una tarjeta microSD que després de fer les configuracions necessàries es posa a la placa.(Important al moment de grabar, activar funcio de SSH y VNC ja que el farem servir més tard.)
- Per trobar la ip de la RP al que està connectat a la wifi corresponent vaig provar 2 mètodes:
 - Amb un cable Ethernet conecto la RP amb un ordinador i amb l'ajuda de Advanced IP Scanner, te com a funció cercar els ip que està connectat al mateix wifi que al ordinador. Però personalment no em va sortir la ip de la RP així que tenir que buscar un altre solucio.
 - El segon mètode que vaig provar va ser posar li una ip fixa a la RP, un arxiu situat a la microSD, **cmdline.txt** , vaig posar li una ip fixe que en aquest cas va ser ip = 192.168.137.2 , i al ordinador hauria de tenir una ip al mateix rang o segment per fer la connexió, vaig posar la ip 192.168.137.1. Llavors per comprovar en el terminal amb el codi ipconfig em surt dels ip de l'ordinador on surt també els ip que he posat. D'aquesta manera va ser com he pogut trobar la ip de la RP.

```
C:\Users\huank>arp -a

Interfaz: 192.168.137.1 --- 0x8
Dirección de Internet      Dirección física      Tipo
192.168.137.2             d8-3a-dd-aa-20-3c    dinámico
192.168.137.255           ff-ff-ff-ff-ff-ff    estático
224.0.0.22                01-00-5e-00-00-16    estático
224.0.0.251               01-00-5e-00-00-fb    estático

Interfaz: 192.168.56.1 --- 0xc
Dirección de Internet      Dirección física      Tipo
192.168.56.255            ff-ff-ff-ff-ff-ff    estático
224.0.0.22                01-00-5e-00-00-16    estático
224.0.0.251               01-00-5e-00-00-fb    estático
224.0.0.252               01-00-5e-00-00-fc    estático

Interfaz: 192.168.1.140 --- 0x10
Dirección de Internet      Dirección física      Tipo
192.168.1.1               e4-ca-12-d4-83-4e    dinámico
192.168.1.255             ff-ff-ff-ff-ff-ff    estático
224.0.0.22                01-00-5e-00-00-16    estático
224.0.0.251               01-00-5e-00-00-fb    estático
224.0.0.252               01-00-5e-00-00-fc    estático
255.255.255.255           ff-ff-ff-ff-ff-ff    estático
```

2- Configuració dins la RP

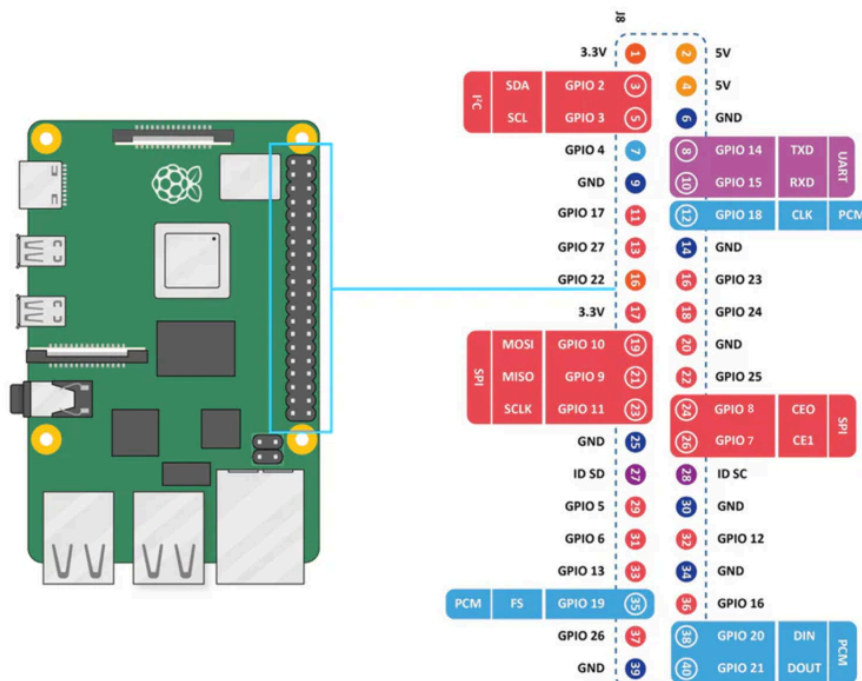
- Les configuracions necessàries son:
 - Actualitzar els paquets amb: **sudo apt update.**
 - Instalem el llenguatge de programació Ruby: **sudo apt-get install -y ruby-full.**
 - Com que programarem al bus i2c de la RP, la activem amb:
sudo raspi-config -> seleccionar interface options -> i2c enable.
- Verificació del correcte conexion de la placa lcd amb el RP.

```

juanka24@raspberrypi1:~$ i2cdetect -y 1
    0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
10:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
20:  --  --  --  --  --  --  27  --  --  --  --  --  --  --  --  --
30:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
40:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
50:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
60:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
70:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
juanka24@raspberrypi1:~$

```

- Pins a conectar a la RP:



3- Llibreries emprades

- Després de cercar informació he decidit utilitzar la biblioteca:
 - <https://github.com/daisukesuzuki/i2c-lcd>
- Per descarregar i instal·lar:
 - **sudo gem install i2c-lcd**
 - **sudo apt-get install i2c-tools**

4- Codi implementat

```
require 'i2c/drivers/lcd'

# Nueva funcion para inicializar el LCD y procesar el texto
def escribir_en_lcd(texto)
  display = I2C::Drivers::LCD::Display.new('/dev/i2c-1', 0x27, rows = 20, cols = 4)
  display.clear

  if texto.empty?
    puts "El texto esta vacio."
  elsif texto.length <= 20
    display.text(texto, 0)
  else
    display.text(texto[0, 20], 0)
    display.text(texto[20, 40], 1)
    display.text(texto[40, 60], 2)
    display.text(texto[60..-1], 3)
  end
end

# Lee el texto de la consola y escribe en el LCD
puts 'Ingrese un texto y presione ENTER:'
input_texto = gets.chomp # Leer input hasta que se presione ENTER

escribir_en_lcd(input_texto)
```

display.text.multi -> a biblioteca
fa falta mes control de error

- Execució al terminal:
 - Va sortir aquest error que encara estic trobant manera de solucionar-lo.

```
$ juanka24@raspberrypi:~ $ ruby lcd2004.rb
Ingrese un texto y presione ENTER:
esto es prueba de la placa lcd
/var/lib/gems/3.1.0/gems/i2c-lcd-0.0.4/lib/i2c/drivers/lcd/display.rb:54:in `text': undefined method `each_char'
for nil:NilClass (NoMethodError)

      string.each_char do |c|
        ^^^^^^^^^^^^^
from lcd2004.rb:19:in `escribir_en_lcd'
from lcd2004.rb:28:in `<main>'
```

- Exemple a la placa lcd-2004

