

MEMÒRIA PUZZLE 1

·Configuracions realitzades

Abans de començar a fer el puzzle, vaig d'haver de descarregar el sistema operatiu de la raspberry dins de la targeta sd.

Per fer-ho vaig iniciar el programa de raspberry imager. Aquest sistema operatiu permet planxar el sistema dins la targeta i a més poder fer les configuracions inicials, és a dir, habilitar el ssh i el vnc sense la necessitat de fer servir un monitor amb teclat.

Un cop amb el sistema planxat, vaig començar amb les configuracions de la raspberry per habilitar el funcionament del LCD.

Com el meu grup li va tocar programar en python, vaig comprovar si la raspberry té instal·lat el python3 i un programa de editor de textos anomenat vim.

També perquè la meua raspberry trobés la pantalla LCD vaig d'haver de habilitar aquestes configuracions:

En la terminal de la raspberry vaig posar aquest comando:

raspi-config

Seleccionem "Advanced Options"

Seleccionem "Enable/Disable automatic loading"

Escollim "Yes"

I per tant ja tenim tot per començar a fer el puzzle



·Biblioteques i paquets instal·lats

Ara hem d'instal·lar un programa anomenat I2C-tools, que ens indicarà l'adreça I2C del LCD quan estigui connectat al Pi. Per tant, a l'indicador d'ordres es:

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

A continuació, hem d'instal·lar SMBUS, que proporciona a la biblioteca Python que farem servir l'accés al bus I2C al Pi

```
sudo apt-get install python-smbus.
```

Ara vaig reiniciar la raspberry i vaig tornar a iniciar sessió. Amb la pantalla LCD connectada, introduïu

```
i2cdetect -y 1 .
```

Això em va mostrar una taula d'adreces per a cada dispositiu I2C connectat.

He trobat una biblioteca Python I2C que té un bon conjunt de funcions i funciona força bé. Aquesta biblioteca es va publicar originalment aquí, després l'ha ampliat i millorat l'usuari de GitHub DenisFromHR

(LINK:<https://gist.github.com/DenisFromHR/cc863375a6e19dce359d>)

A més vaig necessitar també una altre llibreria que es deia lcddriver m'el vaig descarregar i amb això vaig començar a fer el puzle 1.

·Problemes trobats

·A l'hora de connectar la raspberry amb la pantalla LCD, no sabia com connectar els pins de la raspberry amb els pins del LCD.

·No sabia com programar amb python i vaig haver de estudiar com es programava

·No sé com fer que el codi agafi les llibreries sense la necessitat de que aquestes estiguin en la mateixa carpeta.

·Codi ben formatat

```
import lcddriver
```

```
import sys
```

```
import time
```

```
def read_multiline_input():
```

```
    """
```

Lee un string multilínea de la entrada estándar en una sola llamada.

Finaliza la entrada con Ctrl+D.

```
"""
```

```
print("Introduce el texto Ctrl+D para finalizar")
```

```
return sys.stdin.read()
```

```
def center_text(text, width=20):
```

```
    """Centra el texto en una línea de ancho especificado."""
```

```
    padding = (width - len(text)) // 2
```

```
    return " " * padding + text
```

```
def display_text(lcd, text):
```

```
    """
```

Muestra el texto en la pantalla LCD dividiéndolo en líneas.

Si supera el límite de caracteres por línea o el número de líneas, solicita al usuario que reescriba.

```
    """
```

```
    while True:
```

```
        lcd.lcd_clear()
```

```
        lines = text.split('\n')
```

```
        if len(lines) > 4:
```

```
            print(f"\nVaya! Has superado el número máximo de líneas ({len(lines)}/4).  
Vuelve a introducir el texto.")
```

```
            text = read_multiline_input()
```

```
            continue
```

```
        invalid_length = False
```

```
        for i, line in enumerate(lines):
```

```
            if len(line) > 20:
```

```
        print(f"\nVaya! Has superado el máximo de caracteres en la línea {i+1}
({len(line)}/20). Vuelve a introducir el texto.")

        invalid_length = True

        break

    if invalid_length:

        text = read_multiline_input()

        continue

    for i, line in enumerate(lines[:4]):

        lcd lcd_display_string(center_text(line[:20]), i + 1) # Máximo 20 caracteres por
línea

        break

def main():

    lcd = lcd_driver.Lcd()

    lcd lcd_clear()

    lcd lcd_display_string(center_text("Bienvenido, Pau"), 1)

    time.sleep(2)

    lcd lcd_clear()

    lcd lcd_display_string(center_text("Inserte su texto"), 1)

    lcd lcd_display_string(center_text("en la terminal."), 2)

    time.sleep(3)

    lcd lcd_clear()

    lcd lcd_display_string(center_text("Esperando..."), 2)
```

```
text = read_multiline_input()

display_text(lcd, text)

print("\nMostrando el texto insertado en la pantalla LCD.")

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

Com es veu el meu codi pot ser que sigui una mica llarg perquè he volgut que hi hagi un corrector de control d'errors perquè així l'usuari no s'equivoqui posant el text, però si vull que sigui molt més curt només hauria de treure tot el corrector d'errors.