EXPERIMENT NUMBER - 9

EXPERIMENT NAME - LCD INTERFACING USING RASPBERRY PI

DATE - 22/12/2022, THURSDAY

---

* AIM:

To interface LCD module with Raspberry Pi.

* INTEGRATED DEVELOPMENT ENVIRONMENT: (IDE)

Name - Thonny 4.0.1

Publisher - Aivar Annamaa

Support link - https://thonny.org

* IMPORT NECESSARY LIBRARIES:

(sudo pip install adafruit-circuitpython-charlcd)

① Import time, provides various time-related functions.

② Import board, implements a general-purpose board structure which has the functionality needed for a range of purposes.

③ Import digitalio, contains classes to provide access to basic digital IO.

④ Import adafruit_character_lcd.character_lcd as characterlcd, module for interfacing with monochromatic character LCDs.

(a) Hello World Display on LCD (GPIO):

→ Algorithm -

① Metro NO/MA Pin config.

② Initialise the LCD class.

③ Turn backlight ON.

④ Print a two line message.

⑤ Wait for some time.

⑥ Clear the LCD.

→ PYTHON CODE -

```
import time
import board
import digitalio
import adafruit_character_lcd.character_lcd as characterlcd

# modify this if you have a different sized character LCD :-
lcd_columns = 16 ; lcd_rows = 2

lcd_rs = digitalio.DigitalInOut(board.D7)
lcd_en = digitalio.DigitalInOut(board.D8)
lcd_d4 = digitalio.DigitalInOut(board.D9)
lcd_d5 = digitalio.DigitalInOut(board.D10)
lcd_d6 = digitalio.DigitalInOut(board.D11)
lcd_d7 = digitalio.DigitalInOut(board.D12)
lcd_backlight = digitalio.DigitalInOut(board.D13)

lcd = characterlcd.character_LCD_Mono(lcd_rs, lcd_en, lcd_d4, lcd_d5,
lcd_d6, lcd_d7, lcd_columns, lcd_rows, lcd_backlight)
lcd.backlight = True
lcd.message = "Hello in CircuitPython"
time.sleep(5)
lcd.clear()
```

(b) Display Scrolling Text on LCD :

→ ALGORITHM -

① Metro M0/M4 Pin Config.
② Initialize the LCD class.
③ Turn backlight ON.
④ Print a two line message.
⑤ Wait for some time.
⑥ Clear the LCD.
⑦ Turn backlight off.

→ FUNCTIONALITIES -

① Print message right to left.
② Print message left to right.
③ True if cursor is visible.
④ True to blink the cursor.
⑤ Create message to scroll.
- Scroll message to left
- Move displayed text left one column.

→ PYTHON CODE -

```
import time
import board
import digitalio
import adafruit_character_lcd.character_lcd as characterlcd

# Modify this if you have a different sized character LCD:-
lcd_columns = 16; lcd_rows = 2

lcd_rs = digitalio.DigitalInOut(board.D7)
lcd_en = digitalio.DigitalInOut(board.D8)
lcd_d4 = digitalio.DigitalInOut(board.D9)
lcd_d5 = digitalio.DigitalInOut(board.D10)
lcd_d6 = digitalio.DigitalInOut(board.D11)
lcd_d7 = digitalio.DigitalInOut(board.D12)
lcd_backlight = digitalio.DigitalInOut(board.D13)

lcd = characterlcd.character_LCD_Mono(lcd_rs, lcd_en, lcd_d4, lcd_d5,
lcd_d6, lcd_d7, lcd_columns, lcd_rows, lcd_backlight)
lcd.backlight = True
lcd.message = "Hello\n CircuitPython"
time.sleep(5)
lcd.clear()

lcd.text_direction = lcd.RIGHT_TO_LEFT
lcd.message = "Hello\n CircuitPython"
time.sleep(5)
lcd.clear()

lcd.text_direction = lcd.LEFT_TO_RIGHT
lcd.message = "Hello\n CircuitPython"
time.sleep(5)
lcd.clear()
```

```
lcd.cursor = True
lcd.message = "Cursor!"
time.sleep(5)
lcd.clear()

lcd.blink = True
lcd.message = "Blinky Cursor!"
time.sleep(5)
lcd.blink = False
lcd.clear()

scroll_msg = "<-- Scroll"
lcd.message = scroll_msg

for i in range(len(scroll_msg)):
    time.sleep(0.5)
    lcd.move_left()
lcd.clear()

lcd.message = "Going to sleep\n CYa later!"
time.sleep(3)
lcd.backlight = False
time.sleep(2)
```

(c) Assignment - There is a digital board installed in a smart restaurant. Read in numbers of today's special food items as input through keyboard. Scroll through the read foods one by one on LCD for customer's view.

→ ALGORITHM -

① metro M0/M4 Pin config.
② Initialise the LCD class.
③ Turn backlight ON.
④ Print a message.
⑤ Wait for some time.
⑥ scroll message to the left.
- wait for half a second.
- move displayed text left one column.

→ Python Code -

```
import time
import board
import digitalio
import adafruit_character_lcd.character_lcd as characterlcd

# Modify this if you have a different sized character LCD :-
lcd_columns = 16 ; lcd_rows = 2

lcd_rs = digitalio.DigitalInOut(board.D7)
lcd_en = digitalio.DigitalInOut(board.D8)
lcd_d4 = digitalio.DigitalInOut(board.D9)
lcd_d5 = digitalio.DigitalInOut(board.D10)
lcd_d6 = digitalio.DigitalInOut(board.D11)
lcd_d7 = digitalio.DigitalInOut(board.D12)
lcd_backlight = digitalio.DigitalInOut(board.D13)

lcd = characterlcd.character_LCD_Mono(lcd_rs, lcd_en, lcd_d4,
lcd_d5, lcd_d6, lcd_d7, lcd_columns, lcd_rows, lcd_backlight)
lcd.backlight = True
lcd.message = "Welcome!"
time.sleep(5)
lcd.clear()

n = int(input("Enter n value: "))
food_items = []

for i in range(n):

    item = str(input("Enter food item: "))
    food_items.append(item)
    time.sleep(1)
    print(food_items[i])
```

```
    # Scroll message to the left :-
    for j in range (len (food. items [i])) :
        time. sleep (0.5)
        lcd. move_ left ()
    lcd. clear ()

time. sleep (3)
lcd. backlight = False
time. sleep (2)
```

* **RESULT :**

Thus, interfaced LCD module with Raspberry Pi. All the simulation results were verified successfully.

Circuit diagram: