

P15-P16: Backlight  
P3: LCD Contrast Adjustment

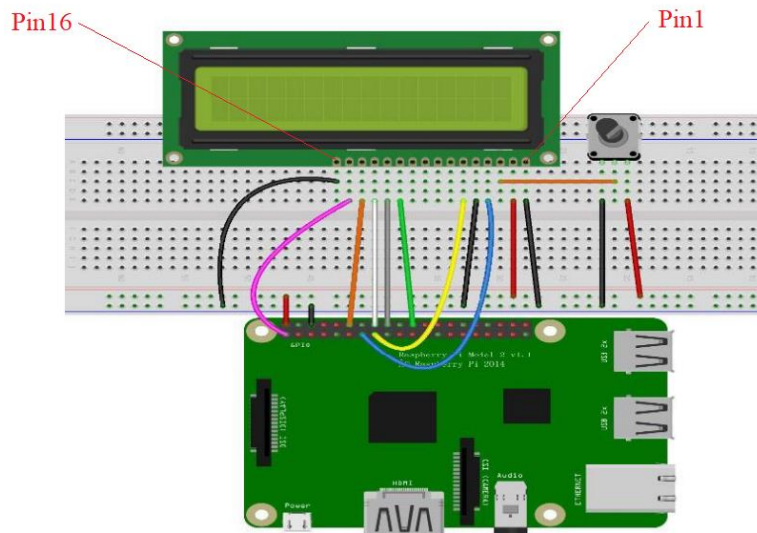
DB0-DB7: Data & Command inputs

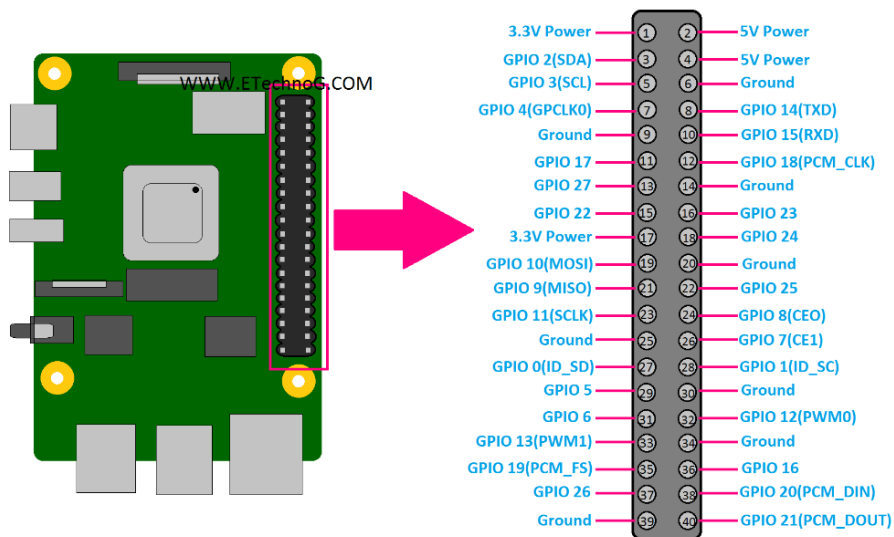
RS--- Instruction/Data Selection

E--- Enable

R/W=0 Always in Write Mode

PIN NO.	SYMBOL	DESCRIPTION	FUNCTION
1	VSS	GROUND	0V (GND)
2	VCC	POWER SUPPLY FOR LOGIC CIRCUIT	+5V
3	VEE	LCD CONTRAST ADJUSTMENT	
4	RS	INSTRUCTION/DATA REGISTER SELECTION	RS = 0 : INSTRUCTION REGISTER RS = 1 : DATA REGISTER
5	R/W	READ/WRITE SELECTION	R/W = 0 : REGISTER WRITE R/W = 1 : REGISTER READ
6	E	ENABLE SIGNAL	
7	DB0	DATA INPUT/OUTPUT LINES	8 BIT: DB0-DB7
8	DB1		
9	DB2		
10	DB3		
11	DB4		
12	DB5		
13	DB6		
14	DB7		
15	LED+	SUPPLY VOLTAGE FOR LED+	+5V
16	LED-	SUPPLY VOLTAGE FOR LED-	0V





## Raspberry Pi 4 Pinout Diagram

Question:

1. Display “you did good job” in LCD by left shifting from right.

Python Code:

```
from time import sleep

class Adafruit_CharLCD(object):

    # commands
    LCD_CLEARDISPLAY = 0x01
    LCD_RETURNHOME = 0x02
    LCD_ENTRYMODESET = 0x04
    LCD_DISPLAYCONTROL = 0x08
    LCD_CURSORSHIFT = 0x10
    LCD_FUNCTIONSET = 0x20
    LCD_SETCGRAMADDR = 0x40
    LCD_SETDDRAMADDR = 0x80

    # flags for display entry mode
    LCD_ENTRYRIGHT = 0x00
    LCD_ENTRYLEFT = 0x02
    LCD_ENTRYSHIFTINCREMENT = 0x01
    LCD_ENTRYSHIFTDECREMENT = 0x00

    # flags for display on/off control
    LCD_DISPLAYON = 0x04
    LCD_DISPLAYOFF = 0x00
    LCD_CURSORON = 0x02
    LCD_CURSOROFF = 0x00
    LCD_BLINKON = 0x01
    LCD_BLINKOFF = 0x00

    # flags for display/cursor shift
```

```

LCD_DISPLAYMOVE      = 0x08
LCD_CURSORMOVE       = 0x00

# flags for display/cursor shift
LCD_DISPLAYMOVE      = 0x08
LCD_CURSORMOVE       = 0x00
LCD_MOVERIGHT        = 0x04
LCD_MOVELEFT         = 0x00
# flags for function set
LCD_8BITMODE          = 0x10
LCD_4BITMODE          = 0x00
LCD_2LINE             = 0x08
LCD_1LINE             = 0x00
LCD_5x10DOTS          = 0x04
LCD_5x8DOTS           = 0x00

def __init__(self, pin_rs=25, pin_e=24, pins_db=[23, 17, 21, 22], GPIO=None):
    # Emulate the old behavior of using RPi.GPIO if we haven't been given
    # an explicit GPIO interface to use
    if not GPIO:
        import RPi.GPIO as GPIO
        GPIO.setwarnings(False)
        self.GPIO = GPIO
        self.pin_rs = pin_rs
        self.pin_e = pin_e
        self.pins_db = pins_db

    self.GPIO.setmode(GPIO.BCM) #GPIO=None use Raspi PIN in BCM mode
    self.GPIO.setup(self.pin_e, GPIO.OUT)
    self.GPIO.setup(self.pin_rs, GPIO.OUT)

    for pin in self.pins_db:
        self.GPIO.setup(pin, GPIO.OUT)

    self.write4bits(0x33) # initialization
    self.write4bits(0x32) # initialization
    self.write4bits(0x28) # 2 line 5x7 matrix
    self.write4bits(0x0C) # turn cursor off 0x0E to enable cursor
    self.write4bits(0x06) # shift cursor right

    self.displaycontrol = self.LCD_DISPLAYON | self.LCD_CURSOROFF | self.LCD_BLINKOFF

    self.displayfunction = self.LCD_4BITMODE | self.LCD_1LINE | self.LCD_5x8DOTS
    self.displayfunction |= self.LCD_2LINE

    # Initialize to default text direction (for romance languages)
    self.displaymode = self.LCD_ENTRYLEFT | self.LCD_ENTRYSHIFTDECREMENT
    self.write4bits(self.LCD_ENTRYMODESET | self.displaymode) # set the entry mode

    self.clear()

def clear(self):
    self.write4bits(self.LCD_CLEARDISPLAY) # command to clear display
    self.delayMicroseconds(3000) # 3000 microsecond sleep, clearing the display takes a long time

def setCursor(self, col, row):
    self.row_offsets = [0x00, 0x40, 0x14, 0x54]
    if row > self.numlines:
        row = self.numlines - 1 # we count rows starting w/0
    self.write4bits(self.LCD_SETDRAMADDR | (col + self.row_offsets[row]))

```

```
#####
from PCF8574 import PCF8574_GPIO
from Adafruit_LCD1602 import Adafruit_CharLCD

from time import sleep, strftime
from datetime import datetime

# Create PCF8574 GPIO adapter.
try:
    mcp = PCF8574_GPIO(PCF8574A_address)
except:
    print ('I2C Address Error !')
    exit(1)
# Create LCD, passing in MCP GPIO adapter.
try:
    mcp = PCF8574_GPIO(PCF8574_address)
except:
    try:
        mcp = PCF8574_GPIO(PCF8574A_address)
    except:
        print ('I2C Address Error !')
        exit(1)
# Create LCD, passing in MCP GPIO adapter.
lcd = Adafruit_CharLCD(pin_rs=0, pin_e=2, pins_db=[4,5,6,7], GPIO=mcp)

def loop():
    mcp.output(3,1) # turn on LCD backlight
    lcd.begin(16,2) # set number of LCD lines and columns
    while(True):
        lcd.clear()
        for i in range(15, -1,-1):
            lcd.clear()
            lcd.setCursor(i,0) # set cursor position
            lcd.message( 'you did good job\n' )# display
            sleep(0.1)
def destroy():
    lcd.clear()
if __name__ == '__main__':
    print ('Program is starting ... ')
    try:
        loop()
        #main()
    except KeyboardInterrupt:
        destroy()
```

**Youtube link:**

<https://youtu.be/FyNeGbojp6s>