

```
#!/usr/bin/python # # HD44780 LCD Test Script for # Raspberry Pi # # Author : Matt Hawkins # Site : http://www.raspberrypi-spy.co.uk # #
Date : 03/08/2012 #
```

```
# The wiring for the LCD is as follows: # 1 : GND # 2 : 5V # 3 : Contrast (0-5V)* # 4 : RS (Register Select) # 5 : R/W (Read Write) - GROUND
THIS PIN # 6 : Enable or Strobe # 7 : Data Bit 0 - NOT USED # 8 : Data Bit 1 - NOT USED # 9 : Data Bit 2 - NOT USED # 10: Data Bit 3 - NOT
USED # 11: Data Bit 4 # 12: Data Bit 5 # 13: Data Bit 6 # 14: Data Bit 7 # 15: LCD Backlight +5V** # 16: LCD Backlight GND
```

```
#import import RPi.GPIO as GPIO import time
```

```
# Define GPIO to LCD mapping LCD_RS = 26 LCD_E = 19 LCD_D4 = 13 LCD_D5 = 6 LCD_D6 = 5 LCD_D7 = 11 LED_ON = 15
```

```
# Define some device constants LCD_WIDTH = 16 # Maximum characters per line LCD_CHR = True LCD_CMD = False
```

```
LCD_LINE_1 = 0x80 # LCD RAM address for the 1st line LCD_LINE_2 = 0xC0 # LCD RAM address for the 2nd line
```

```
# Timing constants E_PULSE = 0.00005 E_DELAY = 0.00005
```

```
def main():
```

```
    # Main program block
```

```
    # Initialise display
    lcd_init()
```

```
    # Toggle backlight on-off-on
    GPIO.output(LED_ON, True)
    time.sleep(1)
    GPIO.output(LED_ON, False)
    time.sleep(1)
    GPIO.output(LED_ON, True)
    time.sleep(1)
```

```
    # Send some centred test
    lcd_byte(LCD_LINE_1, LCD_CMD)
    lcd_string("Raspberry Pi",2)
    lcd_byte(LCD_LINE_2, LCD_CMD)
    lcd_string("Model B",2)
```

```
    time.sleep(3) # 3 second delay
```

```
    # Send some left justified text
    lcd_byte(LCD_LINE_1, LCD_CMD)
    lcd_string("1234567890123456",1)
    lcd_byte(LCD_LINE_2, LCD_CMD)
    lcd_string("abcdefghijklmnop",1)
```

```
    time.sleep(3) # 3 second delay
```

```
    # Send some right justified text
    lcd_byte(LCD_LINE_1, LCD_CMD)
    lcd_string("Raspberrypi-spy",3)
    lcd_byte(LCD_LINE_2, LCD_CMD)
    lcd_string(".co.uk",3)
```

```
    time.sleep(30)
```

```
    # Turn off backlight
    GPIO.output(LED_ON, False)
```

```
def lcd_init():
```

```
    GPIO.setmode(GPIO.BCM)      # Use BCM GPIO numbers
    GPIO.setup(LCD_E, GPIO.OUT) # E
    GPIO.setup(LCD_RS, GPIO.OUT) # RS
    GPIO.setup(LCD_D4, GPIO.OUT) # DB4
    GPIO.setup(LCD_D5, GPIO.OUT) # DB5
    GPIO.setup(LCD_D6, GPIO.OUT) # DB6
    GPIO.setup(LCD_D7, GPIO.OUT) # DB7
    GPIO.setup(LED_ON, GPIO.OUT) # Backlight enable
    # Initialise display
```

```
lcd_byte(0x33,LCD_CMD)
lcd_byte(0x32,LCD_CMD)
lcd_byte(0x28,LCD_CMD)
lcd_byte(0x0C,LCD_CMD)
lcd_byte(0x06,LCD_CMD)
lcd_byte(0x01,LCD_CMD)
```

```
def lcd_string(message,style):
```

```
# Send string to display
# style=1 Left justified
# style=2 Centred
# style=3 Right justified
```

```
if style==1:
    message = message.ljust(LCD_WIDTH," ")
elif style==2:
    message = message.center(LCD_WIDTH," ")
elif style==3:
    message = message.rjust(LCD_WIDTH," ")
```

```
for i in range(LCD_WIDTH):
    lcd_byte(ord(message[i]),LCD_CHR)
```

```
def lcd_byte(bits, mode):
```

```
# Send byte to data pins
# bits = data
# mode = True  for character
#         False for command
```

```
GPIO.output(LCD_RS, mode) # RS
```

```
# High bits
GPIO.output(LCD_D4, False)
GPIO.output(LCD_D5, False)
GPIO.output(LCD_D6, False)
GPIO.output(LCD_D7, False)
if bits&0x10==0x10:
    GPIO.output(LCD_D4, True)
if bits&0x20==0x20:
    GPIO.output(LCD_D5, True)
if bits&0x40==0x40:
    GPIO.output(LCD_D6, True)
if bits&0x80==0x80:
    GPIO.output(LCD_D7, True)
```

```
# Toggle 'Enable' pin
time.sleep(E_DELAY)
GPIO.output(LCD_E, True)
time.sleep(E_PULSE)
GPIO.output(LCD_E, False)
time.sleep(E_DELAY)
```

```
# Low bits
GPIO.output(LCD_D4, False)
GPIO.output(LCD_D5, False)
GPIO.output(LCD_D6, False)
GPIO.output(LCD_D7, False)
if bits&0x01==0x01:
    GPIO.output(LCD_D4, True)
if bits&0x02==0x02:
    GPIO.output(LCD_D5, True)
if bits&0x04==0x04:
    GPIO.output(LCD_D6, True)
if bits&0x08==0x08:
    GPIO.output(LCD_D7, True)
```

```
# Toggle 'Enable' pin
```

```
time.sleep(E_DELAY)
GPIO.output(LCD_E, True)
time.sleep(E_PULSE)
GPIO.output(LCD_E, False)
time.sleep(E_DELAY)
```

```
if name == 'main':
```

```
    main()
```

From:

<http://cameraangle.co.uk/> - WalkerWiki - wiki.alanwalker.uk

Permanent link:

<http://cameraangle.co.uk/doku.php?id=lcd.py&rev=1469462918>

Last update: **2023/03/09 22:35**

