San Francisco Bay University

CE450 Fundamentals of Embedded Engineering
Lab 11 LED Dot Matrix Display

Objectives:

In this week lab, the students will design the program to display the patterns in LED dot matrix through GPIO ports on Raspberry Pi bord and do hands-on exercise through lab assignments

Introduction:

As the name suggests, an LED dot matrix is a matrix composed of LEDs. The lighting up and dimming of the LEDs formulate different characters and patterns.

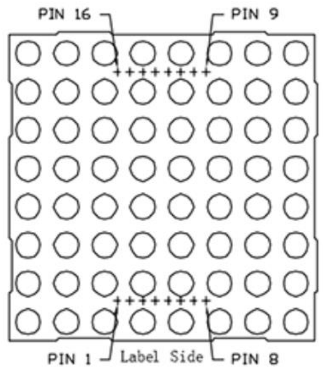
Equipment:

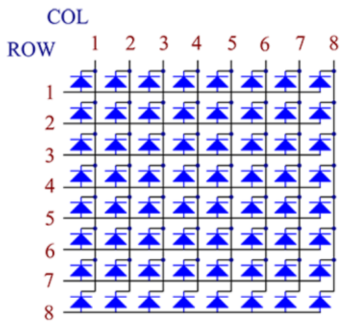
The equipment you require is as follows:

- Laptop & Raspberry Pi 3 model Board
- SunFounder Super Starter Kit V2.0 for Raspberry Pi
- LED dot matrix

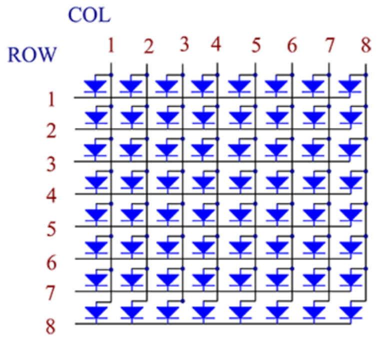
The Laboratory Procedure:

1. **Hardware connection**

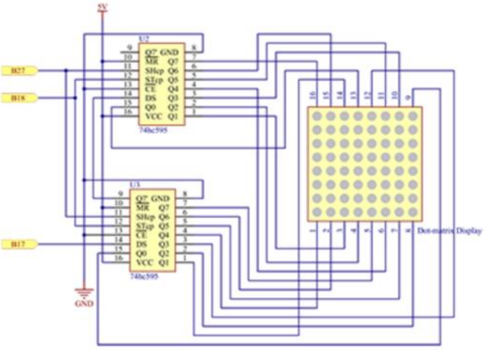


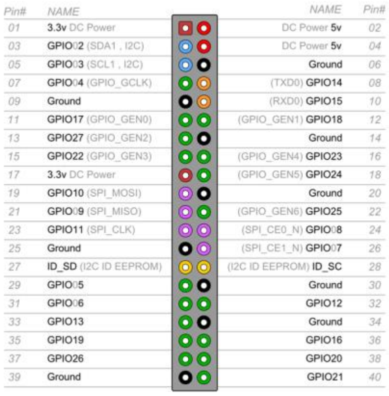


Common Anode

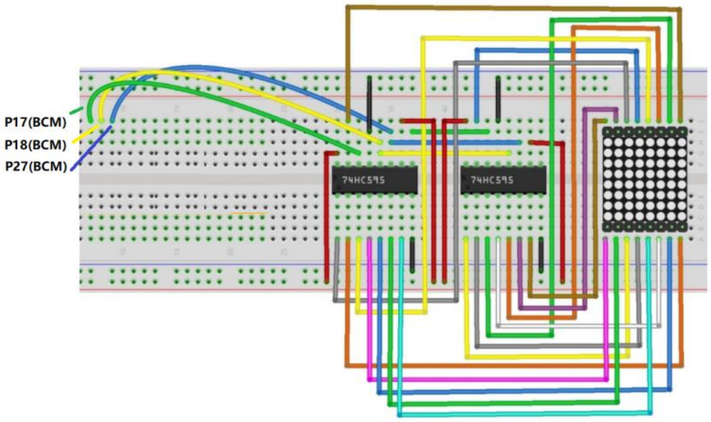


Common Cathode





Pin#	NAME	NAME	Pin#
01	3.3v DC Power	DC Power 5v	02
03	GPIO02 (SDA1 , I2C)	DC Power 5v	04
05	GPIO03 (SCL1 , I2C)	Ground	06
07	GPIO04 (GPIO_GCLK)	(TXD0) GPIO14	08
09	Ground	(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)	(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)	Ground	14
15	GPIO22 (GPIO_GEN3)	(GPIO_GEN4) GPIO23	16
17	3.3v DC Power	(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)	Ground	20
21	GPIO09 (SPI_MISO)	(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)	(SPI_CE0_N) GPIO08	24
25	Ground	(SPI_CE1_N) GPIO07	26
27	ID_SD (I2C ID EEPROM)	(I2C ID EEPROM) ID_SC	28
29	GPIO05	Ground	30
31	GPIO06	GPIO12	32
33	GPIO13	Ground	34
35	GPIO19	GPIO16	36
37	GPIO26	GPIO20	38
39	Ground	GPIO21	40



code_H = [0x01,0xff,0x80,0xff,0x01,0x02,0x04,0x08,0x10,0x20,0x40,0x80,0xff,0xff,0xff,0xff,0xff,0xff,0xff,0xff]
code_L = [0x00,0x7f,0x00,0xfe,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xfe,0xfd,0xfb,0xf7,0xef,0xdf,0xbf,0x7f]

code_H	code_L
0000_0001	0000_0000
1111_1111	0111_1111
1000_0000	0000_0000
1111_1111	1111_1110
0000_0001	0000_0000
0000_0010	0000_0000

2. Control program in Python

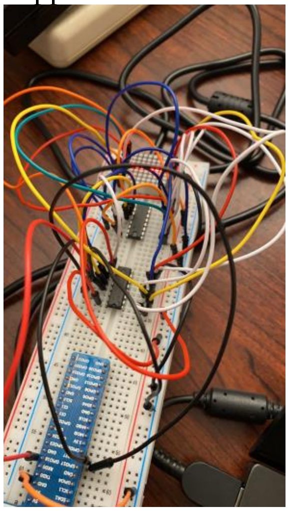
```
# Python Program

import RPi.GPIO as GPIO
import time

SDI   = 17
RCLK  = 18
SRCLK = 27

# we use BX matrix, ROW for anode, and COL for cathode
# ROW  ++++

code_H = [0x01,0xff,0x80,0xff,0x01,0x02,0x04,0x08,0x10,0x20,0x40,0x80,0xff,0xff,0xff,0xff,0xff,0xff,0xff,0xff]
# COL  ----
```

```
code_L = [0x00,0x7f,0x00,0xfe,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xfe,0xfd,0xfb,0xf7,0xef,0xdf,0xbf,0x7f]

def print_msg():
    print ("=========================================")
    print ("|      Dot matrix with two 74HC595      |")
    print ("|    ------------------------------     |")
    print ("|        SDI connect to GPIO 0          |")
    print ("|        RCLK connect to GPIO 1         |")
    print ("|        SRCLK connect to GPIO 2        |")
    print ("|                                       |")
    print ("|   Control Dot matrix with 74HC595     |")
    print ("|                                       |")
    print ("|                             SunFounder|")
    print ("=========================================\n")
    print 'Program is running...'
    print 'Please press Ctrl+C to end the program...'
    raw_input ("Press Enter to begin\n")

def print_matrix(matrix):
    for i in xrange(0,len(matrix)):
        print matrix[i]

def get_matrix(row_buffer, col_buffer, max_row=8, max_col=8):
    matrix_msg = [[0 for i in range(max_row)] for i in range(max_col)]

    print "row_buffer = 0x%02x , col_buffer = 0x%02x"%(row_buffer, col_buffer)
    for row_num in xrange(0,8):
        for col_num in xrange(0,8):
            #print (row_num, col_num), '-->', (((row_buffer >> row_num) & 0x01), ((col_buffer >> col_num) & 0x01))
            if (((row_buffer >> row_num) & 0x01) - ((col_buffer >> col_num) & 0x01)):
                matrix_msg[7-row_num][col_num] = 1
    print_matrix(matrix_msg)
    matrix_msg = [[0 for i in range(max_row)] for i in range(max_col)]

def setup():
    GPIO.setmode(GPIO.BCM)    # Number GPIOs by its BCM location
    GPIO.setup(SDI, GPIO.OUT)
    GPIO.setup(RCLK, GPIO.OUT)
    GPIO.setup(SRCLK, GPIO.OUT)
    GPIO.output(SDI, GPIO.LOW)
    GPIO.output(RCLK, GPIO.LOW)
    GPIO.output(SRCLK, GPIO.LOW)

# Shift the data to 74HC595
def hc595_shift(dat):
    for bit in range(0, 8):
        GPIO.output(SDI, 0x80 & (dat << bit))
        GPIO.output(SRCLK, GPIO.HIGH)
        time.sleep(0.001)
        GPIO.output(SRCLK, GPIO.LOW)
    GPIO.output(RCLK, GPIO.HIGH)
    time.sleep(0.001)
    GPIO.output(RCLK, GPIO.LOW)

def main():
    print_msg()
    while True:
        for i in range(0, len(code_H)):
            hc595_shift(code_L[i])
            hc595_shift(code_H[i])
            get_matrix(code_L[i], code_H[i])
            time.sleep(0.1)

        for i in range(len(code_H)-1, -1, -1):
            hc595_shift(code_L[i])
            hc595_shift(code_H[i])
            get_matrix(code_L[i], code_H[i])
            time.sleep(0.1)

def destroy():
    GPIO.cleanup()


setup()
try:
     main()
except KeyboardInterrupt:
     destroy()
```

**Note: Hardware connection reference and running command*
https://learn.sunfounder.com/lesson-15-driving-dot-matrix-by-74hc595/
https://learn.sunfounder.com/category/super-kit-v3-0-for-raspberry-pi/

The Laboratory Assignments:

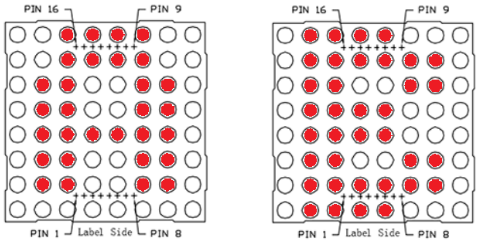
1. Build up the hardware circuit and run the example program to observe what will happen



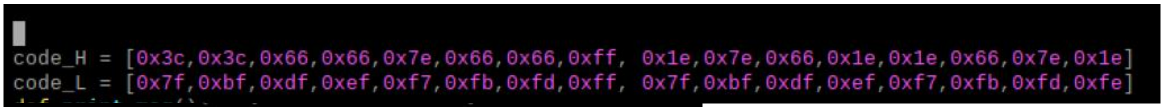
Youtube Link:

https://youtube.com/shorts/vFiEhPEpX-U?feature=share

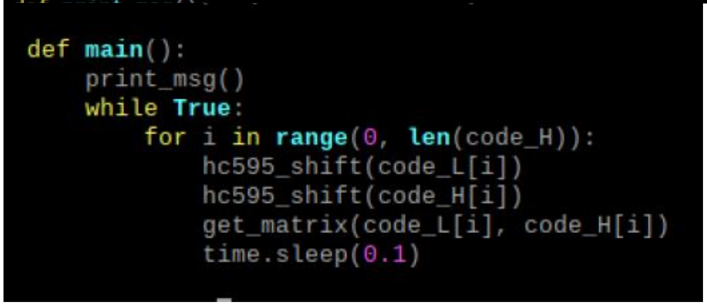
2. Design the program to periodically display "A" and "B" in one LED dot matrix as follows



Code Snippet:



```
code_H = [0x3c,0x3c,0x66,0x66,0x7e,0x66,0x66,0xff, 0x1e,0x7e,0x66,0x1e,0x1e,0x66,0x7e,0x1e]
code_L = [0x7f,0xbf,0xdf,0xef,0xf7,0xfb,0xfd,0xff, 0x7f,0xbf,0xdf,0xef,0xf7,0xfb,0xfd,0xfe]
```



```
def main():
    print_msg()
    while True:
        for i in range(0, len(code_H)):
            hc595_shift(code_L[i])
            hc595_shift(code_H[i])
            get_matrix(code_L[i], code_H[i])
            time.sleep(0.1)
```

Youtube Link:
https://youtube.com/shorts/q4xJudYptGY?feature=share