Press the Buttons & LED pattern

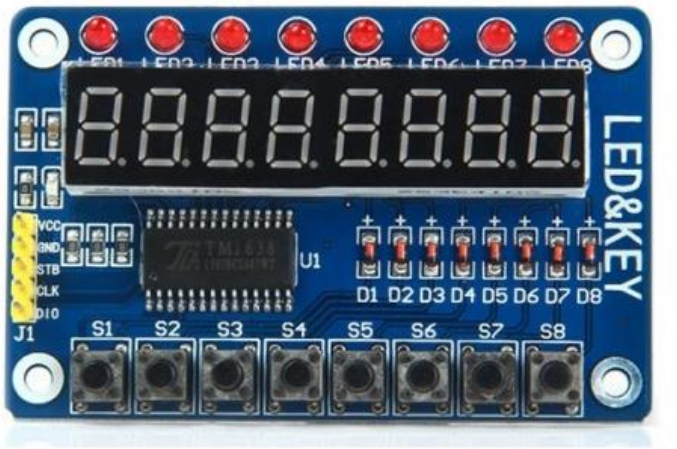
RASPBERRY JAM

3.3V	1	2	5V	
2 SDA1	3	4	5V	
3 SCL1	5	6	GND	
4 GCLK	7	8	14 TXD0	
GND	9	10	15 RXD0	
17	11	12	18	
27	13	14	GND	
22	15	16	23	
3.3V	17	18	24	
10 MOSI	19	20	GND	
9 MISO	21	22	25	
11 SCLK	23	24	8 CE0_N	
GND	25	26	7 CE1_N	
ID_SD	27	28	ID_SC	
5	29	30	GND	
6	31	32	12	
13	33	34	GND	
19	35	36	16	
26	37	38	20	
GND	25	40	21	

Getting started

1. Before turning on the Raspberry Pi wire up the board using the details below.

LED&KEY	Raspberry Pi
VCC	3.3V (Pin 1)
GND	GND (Pin 6)
STB	GPIO 22 (Pin15)
CLK	GPIO 21 (Pin 40)
DIO	GPIO 17 (Pin11)



2. Time to connect the power to boot up the Raspberry Pi.

Code to read the buttons

1. Let's start coding:

   First we're going to read the buttons and learn a bit about binary.

2. Run Python 3 and open a new file [File]-[New].

   Save the file and call it **`pressbuttons.py`**

   You could use any name but it makes it easier for the workshop if everyone is the same

Enter the code below.

```
#!/usr/bin/env python3
# Output the value for the buttons.

# TM1638

import TM1638

# These are the pins the display is connected to. Adjust accordingly.
# In addition to these you need to connect to 3.3V and ground.

DIO = 17
CLK = 21
STB = 22

display = TM1638.TM1638(DIO, CLK, STB)

display.enable(1)
display.set_led(0, True)

count = 0
active = True
while active == True:
····keys = display.get_buttons()
····display.set_text(str(keys))
····print(str(keys))
····if keys == 128:
········active = False
```

Run your program and press the buttons.

NOTE: If you press the button on the right the program will exit (where is this in the code)

If all is working well then different numbers will appear on the 7 segment display.

Press each button one by one. Can you see a pattern in the numbers.

Try pressing more than one button at a time. Do they still make sense.

The numbers are decimal version of binary (except kind of backwards)

Binary	Decimal
00000001	1
00000010	2
00000100	4
00001000	8
00010000	16
00100000	32
01000000	64
10000000	128

Binary numbers are where each digit can only be zero (0) or one (1). This is how computers store data/information. For computers each 0/1 is called a Bit. Eight Bits is called a Byte. Computer store their data in Byte sized pieces. So, this board having 8 LEDs, 8 7-Segment display and 8 buttons is no coincident, It is the most that can be handled with Bytes. If there were 9 then another set of Bytes would be needed.

For this board the buttons are mapped on the opposite way to the bits.
The button on the left is the bit on the right.

Light up some LEDs

1. A fun pattern for LEDs is to make them light up one by one from left to right and then back again:

2. In Python 3 open a new file [File]-[New].
   Save the file and call it **`ledpattern.py`**
   Enter the code below.

```
#!/usr/bin/env python3
# LEDs pattern

# TM1638

import TM1638
import time

# These are the pins the display is connected to. Adjust accordingly.
# In addition to these you need to connect to 5V and ground.

DIO = 17
CLK = 21
STB = 22

pause = 0.01

display = TM1638.TM1638(DIO, CLK, STB)

display.enable(1)
display.set_led(0, True)

for x in range(5):
····for i in range(7):
········display.set_led(i,True)
········time.sleep(pause)
········display.set_led(i,False)
········time.sleep(pause)

····for i in range (7,0,-1):
········display.set_led(i,True)
········time.sleep(pause)
········display.set_led(i,False)
········time.sleep(pause)
```

Run the code and enjoy the pattern.