SWITCH PROGRAM:

import sys

sys.path.append('/home/pi/Adafruit-Raspberry-Pi-Python-Codelegacy/Adafruit\_CircuitPython\_MCP230xx-main') # LIBRARY

import time

import board

import busio

from digitalio import Direction, Pull

from adafruit\_mcp230xx.mcp23017 import MCP23017

# Initialize the I2C bus:

i2c = busio.I2C(board.SCL, board.SDA)

# Initialize the MCP23017 chip on the bonnet

mcp = MCP23017(i2c)

# Optionally change the address of the device if you set any of the A0, A1, A2

# pins. Specify the new address with a keyword parameter:

mcp = MCP23017(i2c, address=0x20) # MCP23017 chip address

# Make a list of all the port A pins

PortA = [ ]

for pin in range(0, 8):

PortA.append(mcp.get\_pin(pin))

# Make a list of all the port B pins

PortB = [ ]

for pin in range(8, 16):

PortB.append(mcp.get\_pin(pin))

for pin in range(0,8):

PortA[pin].value=True

for pin in range(0,8):

PortB[pin].value=False

#Configuring RGB LEDs connected pins as OUTPUT

PortA[0].direction = Direction.OUTPUT

PortA[1].direction = Direction.OUTPUT

PortA[2].direction = Direction.OUTPUT

PortA[3].direction = Direction.OUTPUT

PortA[4].direction = Direction.OUTPUT

PortA[5].direction = Direction.OUTPUT

PortA[6].direction = Direction.OUTPUT

PortA[7].direction = Direction.OUTPUT

PortB[0].direction = Direction.OUTPUT

#Configuring Switch connected pins as INPUT

PortB[1].direction = Direction.INPUT # Switch

PortB[1].pull = Pull.UP # Enable Internal Pull UP

PortB[2].direction = Direction.INPUT #Switch

PortB[2].pull = Pull.UP # Enable Internal Pull UP

try:

while True:

pinData9 =PortB[1].value

pinData10 =PortB[2].value

if (pinData10==False): #If the Switch Presses the pin goes to LOW / False

print("S1 Pressed")

PortA[0].value = True

PortA[3].value = True

PortA[6].value = True

if (pinData9==False): #If the Switch Presses the pin goes to LOW / False

print("S2 Pressed")

PortA[1].value = True

PortA[4].value = True

PortA[7].value = True

if(pinData10 == True and pinData9==True): #If the switches are in released

condition the Pins will be in HIGH or True

print("Both Switches are OFF ")

PortA[0].value = False

PortA[3].value = False

PortA[6].value = False

PortA[1].value = False

PortA[4].value = False

PortA[7].value = False

PortA[2].value = False

PortA[5].value = False

PortA[0].value = False

time.sleep(2)

except KeyboardInterrupt: # CLEAR THE PINS press CTRL+C

for pin in range(0,8):

PortA[pin].value=False

for pin in range(0,8):

PortB[pin].value=False