

# 1.Blink LEDs using Push Button

```
import time

import sys

sys.path.append('/home/pi/Adafruit-Raspberry-Pi-Python-Code-legacy/Adafruit_MCP230xx')

from Adafruit_MCP230XX import Adafruit_MCP230XX

mcp = Adafruit_MCP230XX(busnum = 1, address = 0x20, num_gpios = 16)

mcp.config(9, mcp.INPUT)

mcp.pullup(9, 1)

for i in range(0,9):
    mcp.config(i,mcp.OUTPUT)

while (True):
    x = mcp.input(9)
    if x==1:
        mcp.output(0,1)  #Red light of LED 1
        mcp.output(4,1)  #Green light of LED 2
        mcp.output(8,1)  #Blue light of LED 3
        time.sleep(1)
    else:
        mcp.output(0,0)
        mcp.output(4,0)
        mcp.output(8,0)
    time.sleep(1)
```

## 2. Buzzer using Push Button

```
import time
import sys

sys.path.append('/home/pi/Adafruit-Raspberry-Pi-Python-Code-
legacy/Adafruit_MCP230xx')

from Adafruit_MCP230XX import Adafruit_MCP230XX

mcp = Adafruit_MCP230XX(busnum = 1, address = 0x20, num_gpios = 16)

mcp.config(9, mcp.INPUT)

mcp.pullup(9, 1)

mcp.config(11,mcp.OUTPUT)

while (True):
    x = mcp.input(9)
    if x==1:
        mcp.output(11,1)    #Buzzer is blown
        time.sleep(1)
        mcp.output(11,0)
    else:
        mcp.output(11,0)
    time.sleep(1)
```

### 3. Temperature, Pressure, Humidity using BME280 sensor

```
import sys
import time
import BME280lib as bme
from Adafruit_MCP230XX import Adafruit_MCP230XX
mcp = Adafruit_MCP230XX(busnum=1,address=0x20,num_gpios = 16)
DEVICE = 0x76
mcp.config(1,mcp.OUTPUT)
mcp.config(11,mcp.OUTPUT)
while True:
    t,p,h = bme.readBME280All()
    print "Temperature " , t , "C"
    print "Pressure " , p , "hPa"
    print "Humidity " , h , "%"
    if t>28 and t<30:
        mcp.output(1,1)
        time.sleep(1)
        mcp.output(1,0)
    else:
        mcp.output(11,1)
        time.sleep(1)
        mcp.output(11,0)

    time.sleep(2)
```

## 4. Ultrasonic Sensor

```
import sys
import time
import RPi.GPIO as GPIO

from Adafruit_MCP230XX import Adafruit_MCP230XX
mcp=Adafruit_MCP230XX(busnum=1, address =0x20, num_gpios=16)
mcp.config(11,mcp.OUTPUT)
mcp.output(11,0)

GPIO.setwarnings(False)
GPIO.setmode(GPIO.BOARD)

GPIO_TRIGGER = 16

GPIO_ECHO = 18

GPIO.setup(GPIO_TRIGGER,GPIO.OUT)

GPIO.setup(GPIO_ECHO,GPIO.IN)

GPIO.output(GPIO_TRIGGER, False)

while True:

    GPIO.output(GPIO_TRIGGER, True)
```

```
time.sleep(0.1)
GPIO.output(GPIO_TRIGGER, False)

while GPIO.input(GPIO_ECHO)==0:
    start = time.time()

while GPIO.input(GPIO_ECHO)==1:
    stop = time.time()

elapsed = stop-start

distance = elapsed * 34300

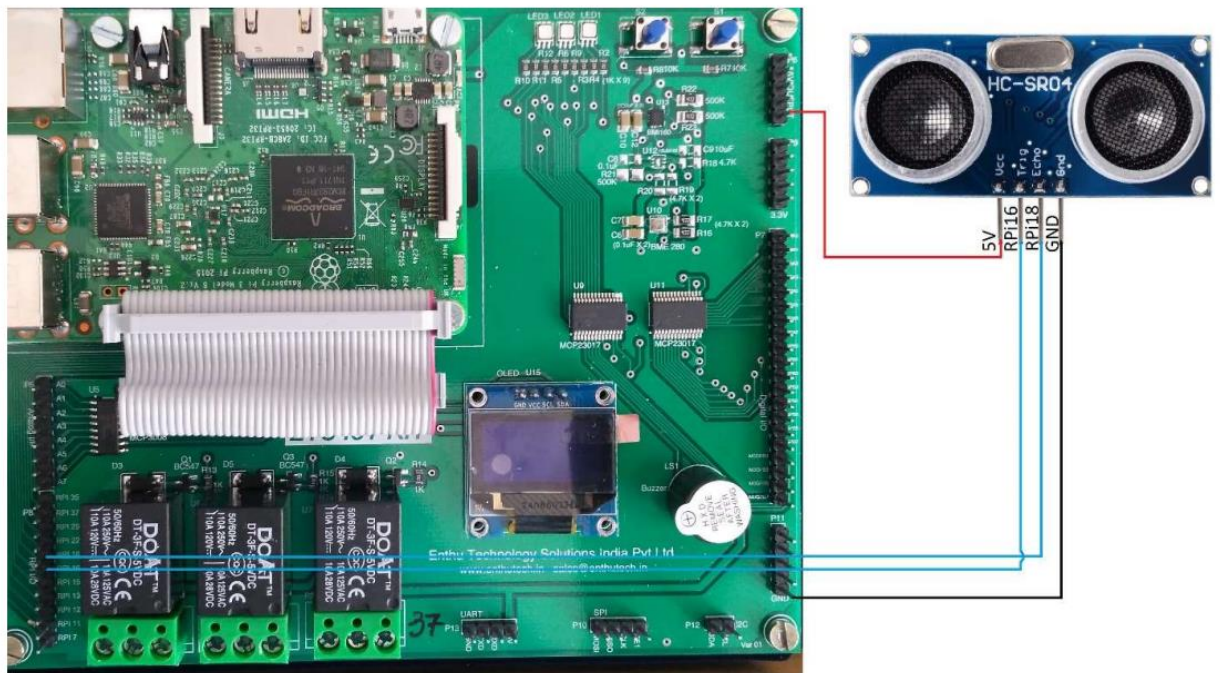
distance = distance / 2
print "Distance : %.1f" % distance

if(distance<70):
    mcp.output(11,1)
else:
    mcp.output(11,0)

time.sleep(1)

GPIO.cleanup()
sys.exit()
```

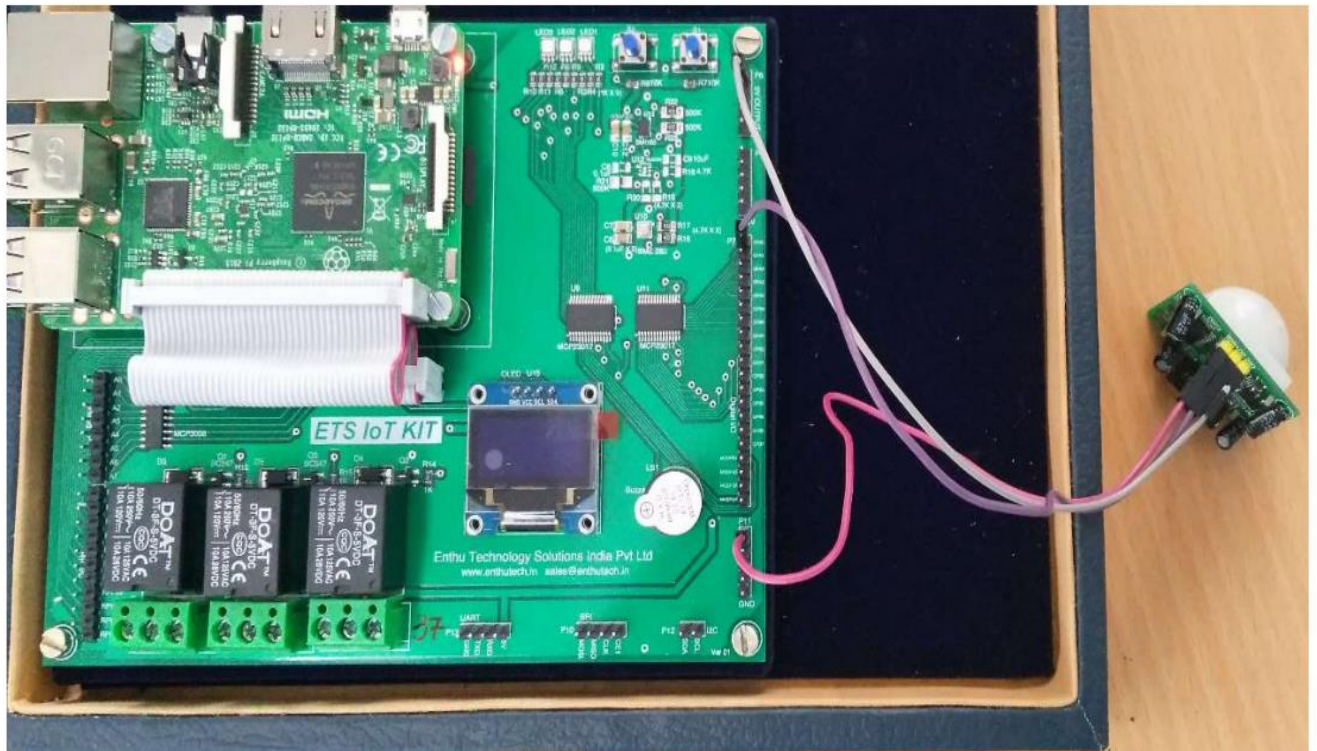
## Connection diagram of ULTRASONIC Sensor:



## Pin mapping:

ETS-IoT Kit	SENSOR
5V	VCC
RPi16	TRIG
RPi18	ECHO
GND	GND

## 5. PIR Sensor



```
from Adafruit_MCP230XX import Adafruit_MCP230XX
mcp = Adafruit_MCP230XX(busnum=1, address=0x21, num_gpios=16)
mcp1 = Adafruit_MCP230XX(busnum=1, address=0x20, num_gpios=16)
import time
mcp.config(0,mcp.INPUT)
mcp1.config(1,mcp.OUTPUT)
mcp.pullup(0,1)
while True:
    i = mcp.input(0)
    time.sleep(1)
    if i==1:
        mcp1.output(1,1)
        print "person is detected"
        mcp1.output(1,0)
```

else:

print "person is not detected"

## 6. Relay

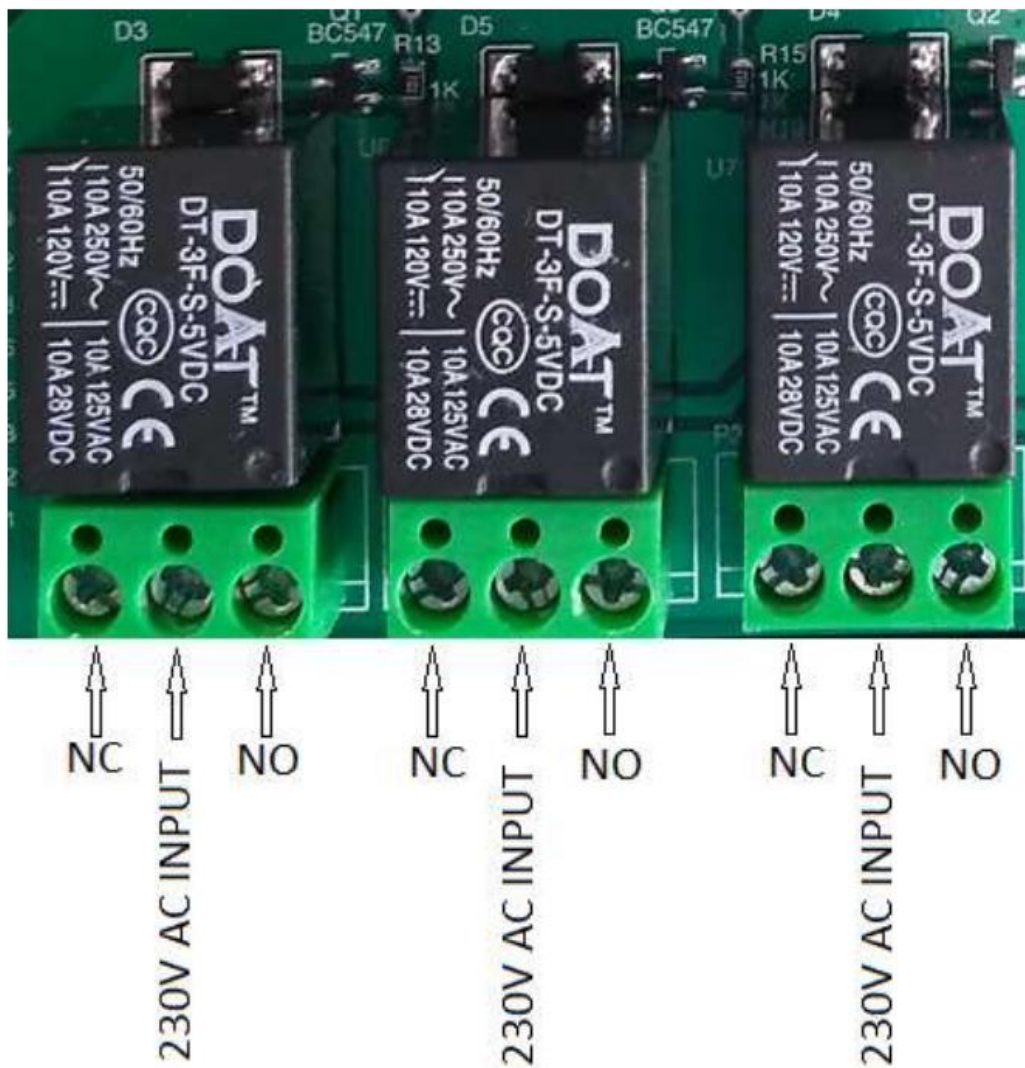


Figure: RELAY Image in ETS-IoT Trainer Kit



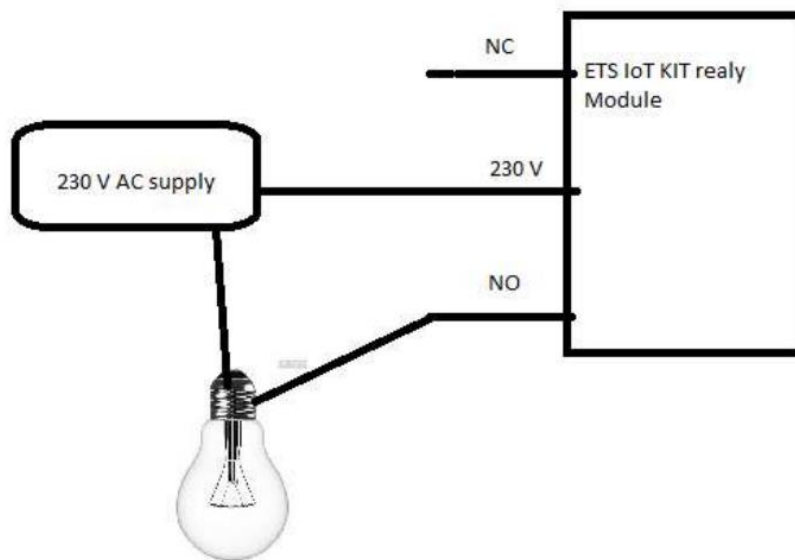


Figure: Connection diagram of RELAY with Bulb

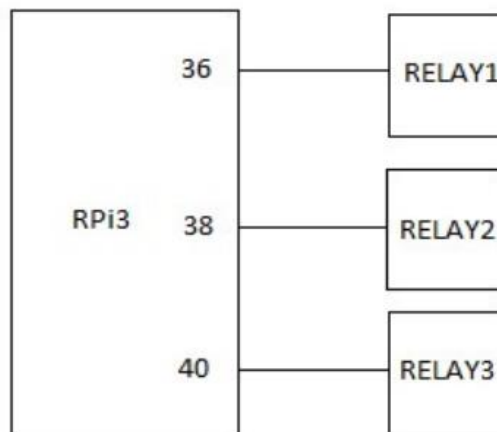


Figure: Connection diagram of RELAY

```
import RPi.GPIO as GPIO
import time
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BOARD)
GPIO.setup(36, GPIO.OUT)
while True:
```

```
    #Relay1 ON
    GPIO.output(36, 1)
    time.sleep(2)
    #Relay1 OFF
    GPIO.output(36, 0)
    time.sleep(2)
```