9. SOC Experiments using Raspberry PI or Ordroid Xu4: 02

a. Touch sensor

b. Tracking sensor

1. Touch sensor

Components Required:

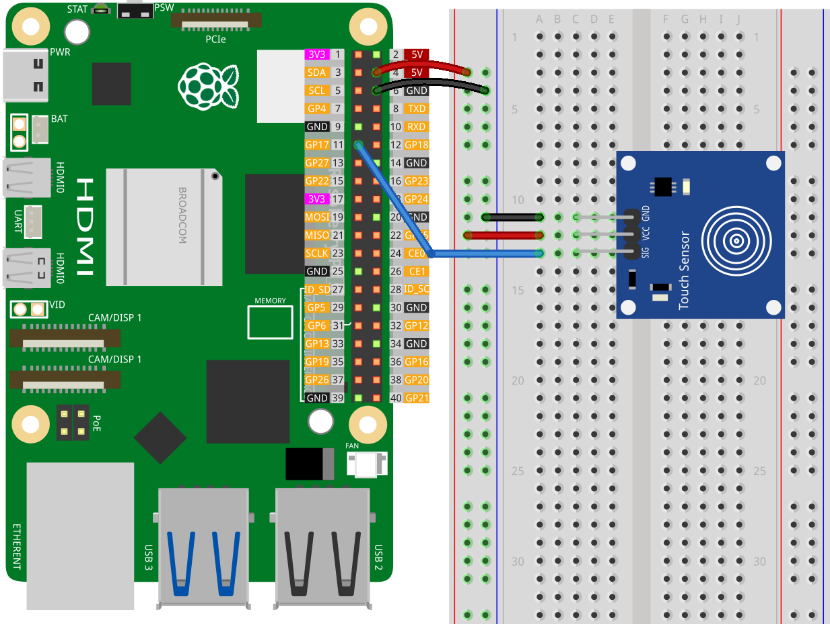
Rasberry Pi

Touch Sensor

Breadboard

Jumper Wires

**Circuit Diagram:**



**Program:**

from gpiozero import Button

from signal import pause

# Function called when the sensor is touched

def touched():

# Print a message indicating the sensor is touched

print("Touched!")

# Function called when the sensor is not touched

def not\_touched():

# Print a message indicating the sensor is not touched

print("Not touched!")

# Initialize a Button object for the touch sensor

# GPIO 17: pin connected to the sensor

# pull\_up=None: disable internal pull-up/pull-down resistors

# active\_state=True: high voltage is considered the active state

touch\_sensor = Button(17, pull\_up=None, active\_state=True)

# Assign functions to sensor events

touch\_sensor.when\_pressed = touched

touch\_sensor.when\_released = not\_touched

pause() # Keep the program running to detect touch events

1. Tracking Sensor

Components Required:

Rasberry Pi

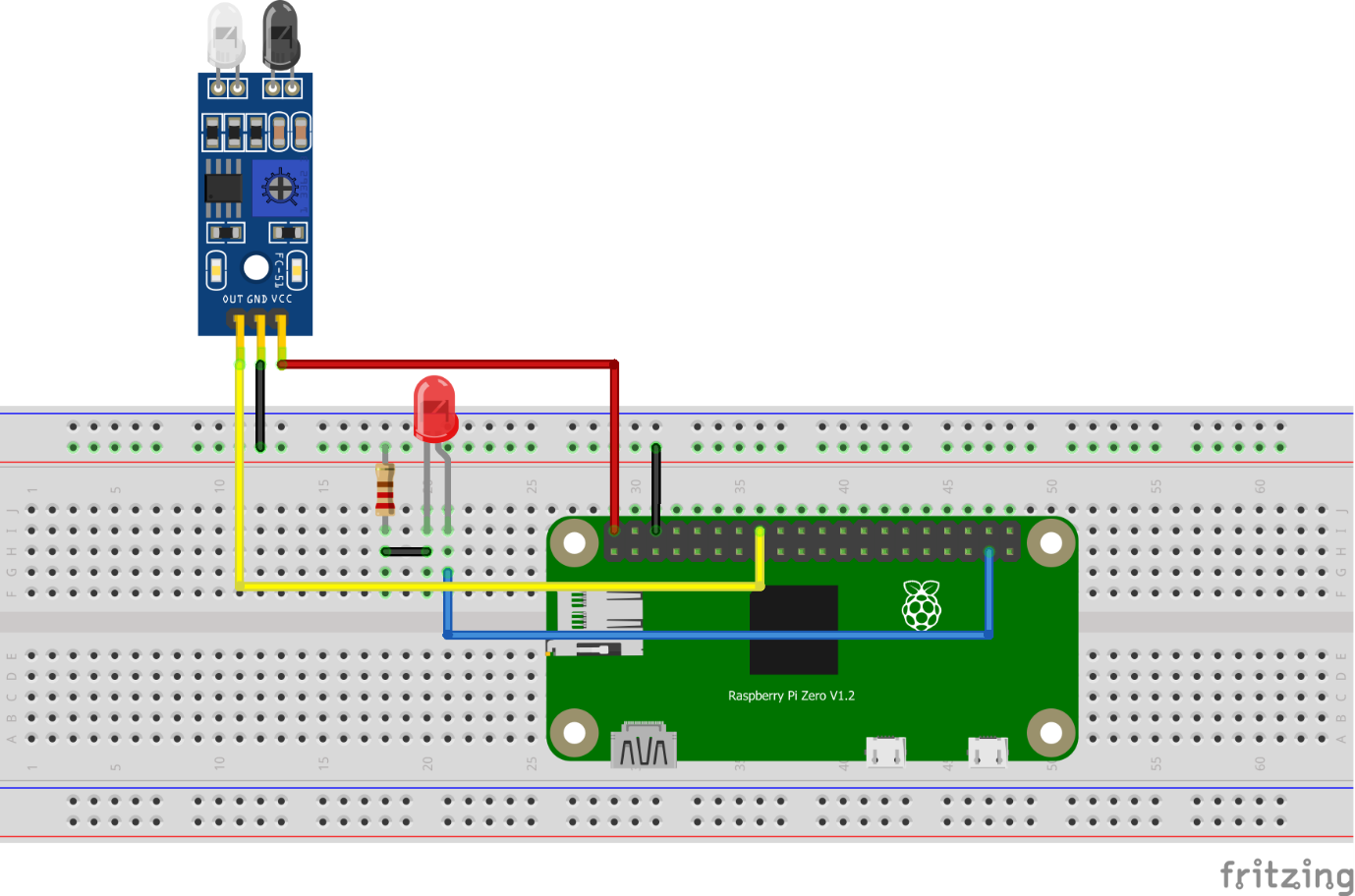
Tracking Sensor

LED

10 Ohm Resistor

Jumper Wires

Circuit Diagram:



Program:

import RPi.GPIO as GPIO

import time

# declare the sensor and led pin

sensor\_pin = 23

led\_pin = 26

# GPIO setup

GPIO.setwarnings(False)

GPIO.setmode(GPIO.BCM)

GPIO.setup(sensor\_pin, GPIO.IN)

GPIO.setup(led\_pin, GPIO.OUT)

try:

while True:

if GPIO.input(sensor\_pin):

# If no object is near

GPIO.output(led\_pin, False)

while GPIO.input(sensor\_pin):

time.sleep(0.2)

else:

# If an object is detected

GPIO.output(led\_pin, True)

except KeyboardInterrupt:

GPIO.cleanup()

10. SOC Experiments using Raspberry PI or Ordroid Xu4: Control and communication Experiments

a. Mercury tilt switch

b. Laser emitter

a. Mercury tilt Switch

Components Required:

Rasberry Pi

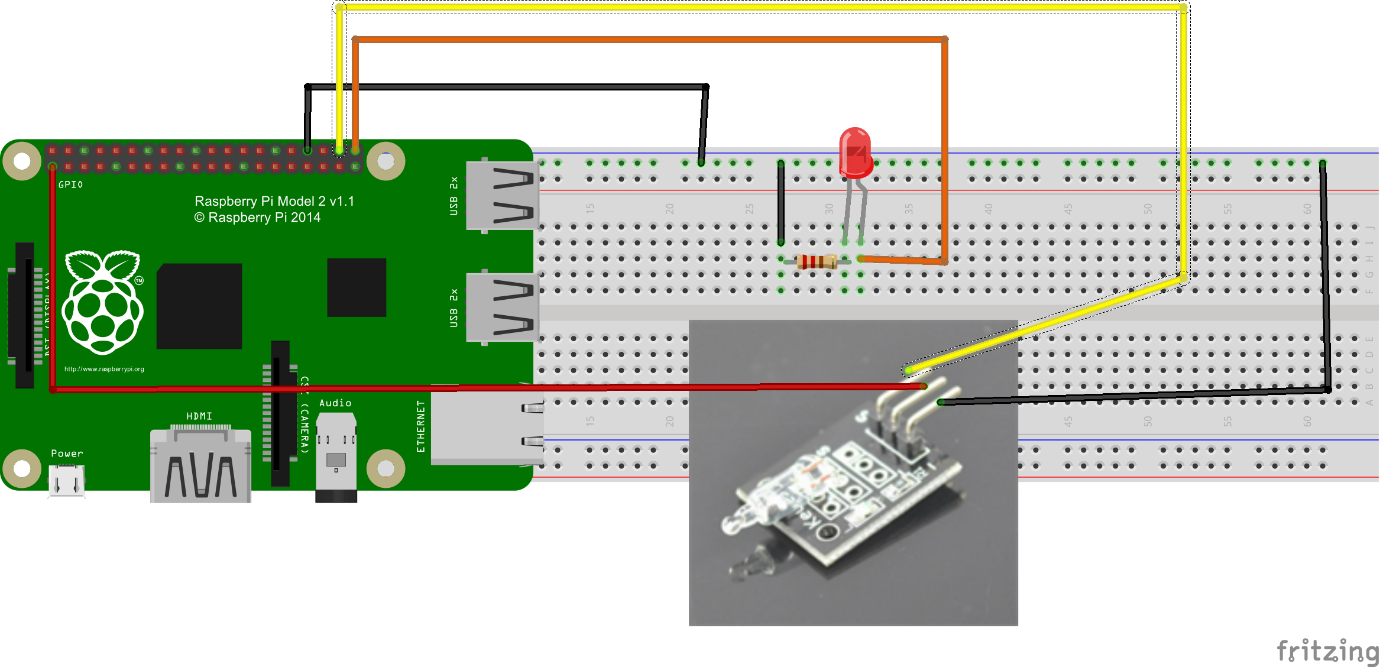
Tilt Sensor

LED

10 Ohm resistor

Jumper Wires

Circuit Diagram:



Program:

#Project tutorial URL http://osoyoo.com/?p=804

#Copyright Osoyoo.com

import RPi.GPIO as GPIO

import time

sensor\_pin = 38

led\_pin = 40

GPIO.setmode(GPIO.BOARD)

GPIO.setup(led\_pin,GPIO.OUT)

GPIO.setup(sensor\_pin, GPIO.IN)

current\_state = 0

try:

while True:

time.sleep(0.1)

current\_state = GPIO.input(sensor\_pin)

if current\_state == 1:

print("tilt sensor value is %s" % (current\_state))

GPIO.output(led\_pin,True)

else:

print("tilt sensor value is %s" % (current\_state))

GPIO.output(led\_pin,False)

except KeyboardInterrupt:

pass

finally:

GPIO.cleanup()

1. Laser Emitter

Components Required:

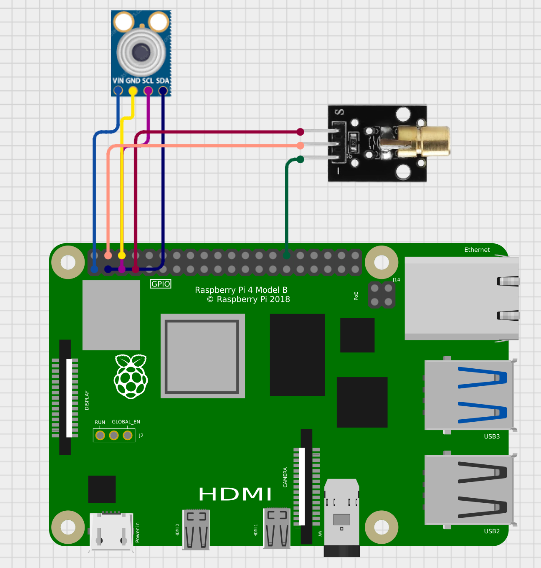
Laser Emitter

Rasberry Pi

Jumper Wires

Breadboard

Circuit Diagram:



Program:

const int laserPin = 4; // GPIO4 on Raspberry Pi connected to SIG of KY-008

const int delayTime = 1000; // Delay time in milliseconds

void setup() {

  pinMode(laserPin, OUTPUT); // Set laserPin as an OUTPUT

}

void loop() {

  digitalWrite(laserPin, HIGH); // Turn the laser on

  delay(delayTime); // Wait for a second

  digitalWrite(laserPin, LOW); // Turn the laser off

  delay(delayTime); // Wait for a second

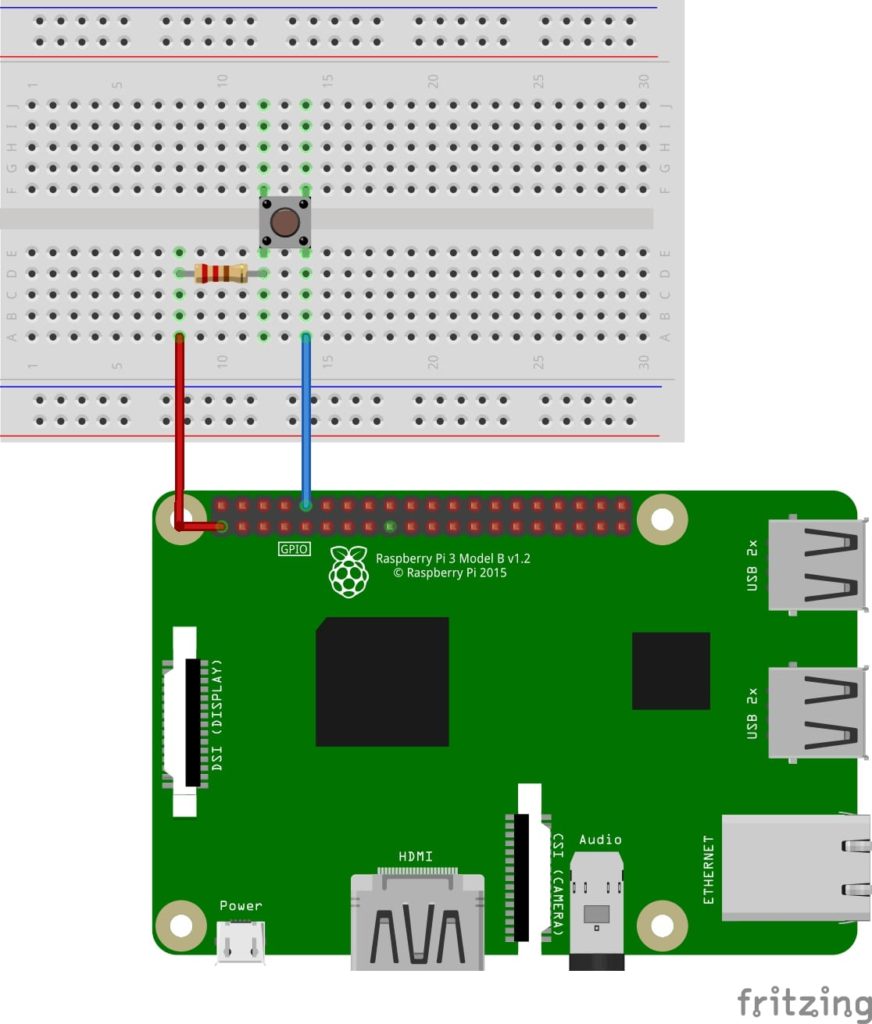
}

11. SOC Experiments using Raspberry PI or Ordroid Xu4:

a. Button

b. IR emitter

a. Button



Program:

import RPi.GPIO as GPIO # Import Raspberry Pi GPIO library

def button\_callback(channel):

print("Button was pushed!")

GPIO.setwarnings(False) # Ignore warning for now

GPIO.setmode(GPIO.BOARD) # Use physical pin numbering

GPIO.setup(10, GPIO.IN, pull\_up\_down=GPIO.PUD\_DOWN) # Set pin 10 to be an input pin and set initial value to be pulled low (off)

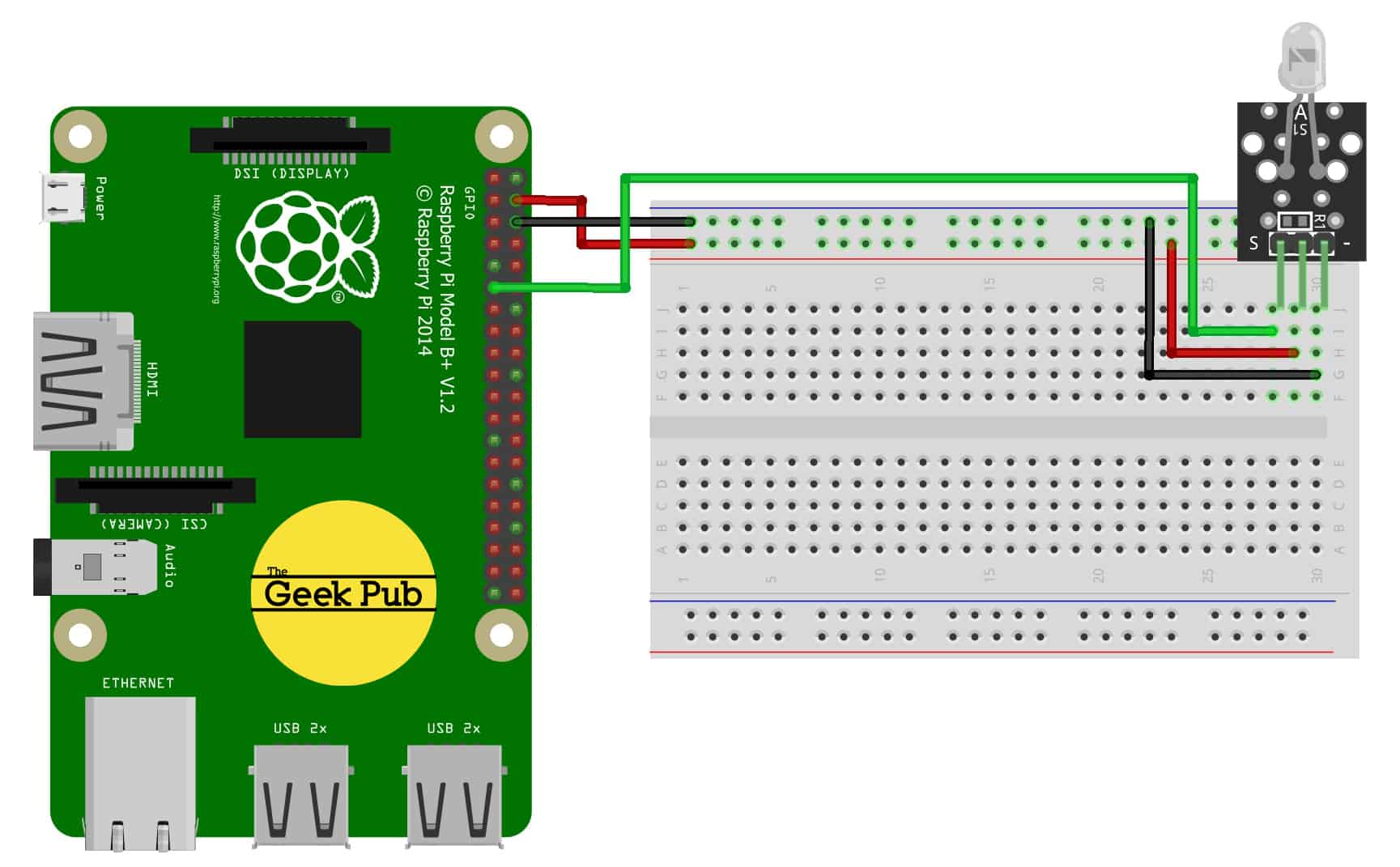
GPIO.add\_event\_detect(10,GPIO.RISING,callback=button\_callback) # Setup event on pin 10 rising edge

message = input("Press enter to quit\n\n") # Run until someone presses enter

GPIO.cleanup() # Clean up

1. IR Emitter

Circuit Diagram:



Program:

import os

import time

def send\_ir\_command(command):

os.system(f"irsend SEND\_ONCE myremote {command}")

print(f"Sent command: {command}")

# Example usage

send\_ir\_command("KEY\_POWER")

time.sleep(1)

send\_ir\_command("KEY\_VOLUMEUP")

Assignment:

12. SOC Experiments using Raspberry PI or Ordroid Xu4:

a. Ball Switch

b. Tap Sensor