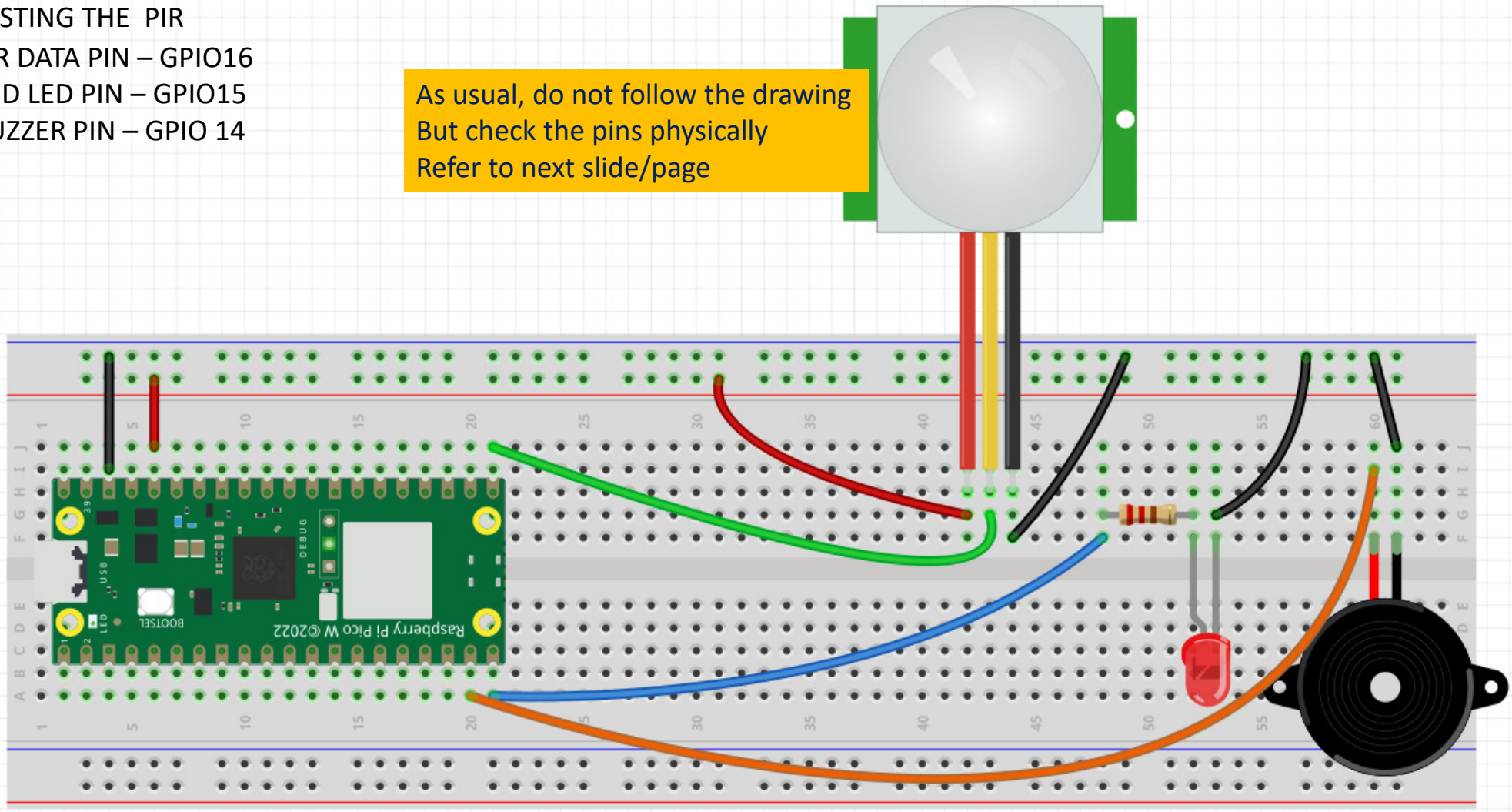
TESTING THE PIR

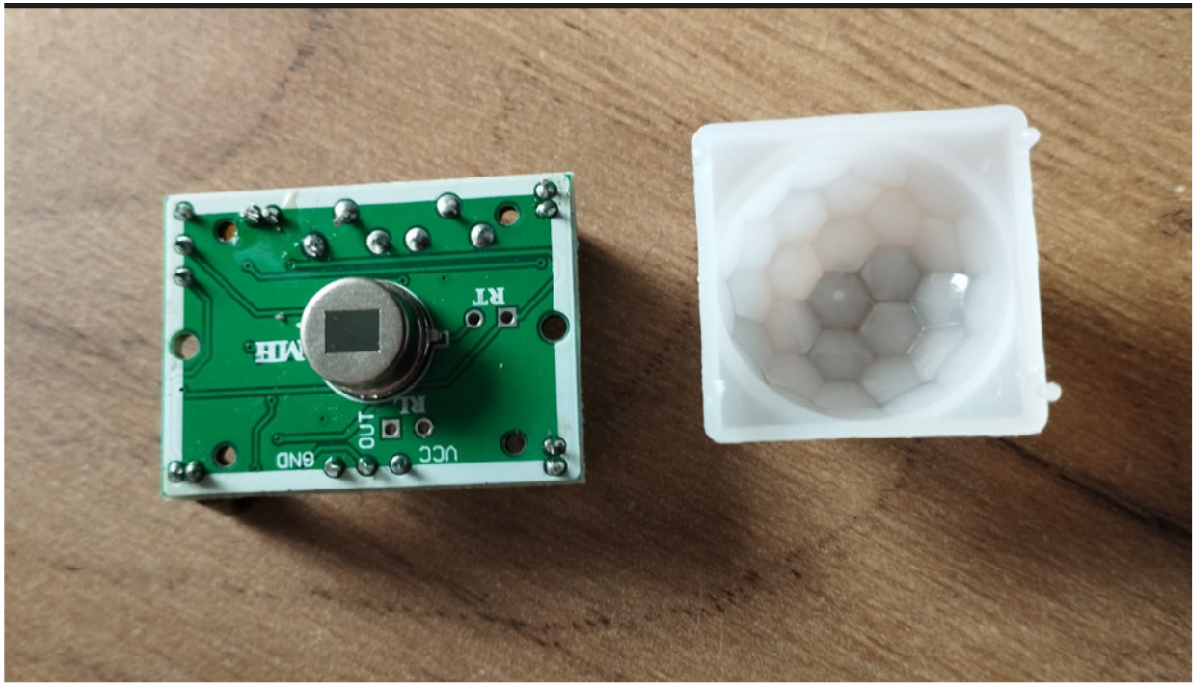
PIR DATA PIN – GPIO16

RED LED PIN – GPIO15

BUZZER PIN – GPIO 14

As usual, do not follow the drawing
But check the pins physically
Refer to next slide/page





Gently remove the plastic cap from the PIR
The pins are marked on the device itself.
Follow these pin markings and not the drawing

Program to Test the PIR

When motion is detected
The LED and Buzzer
Will be activated.

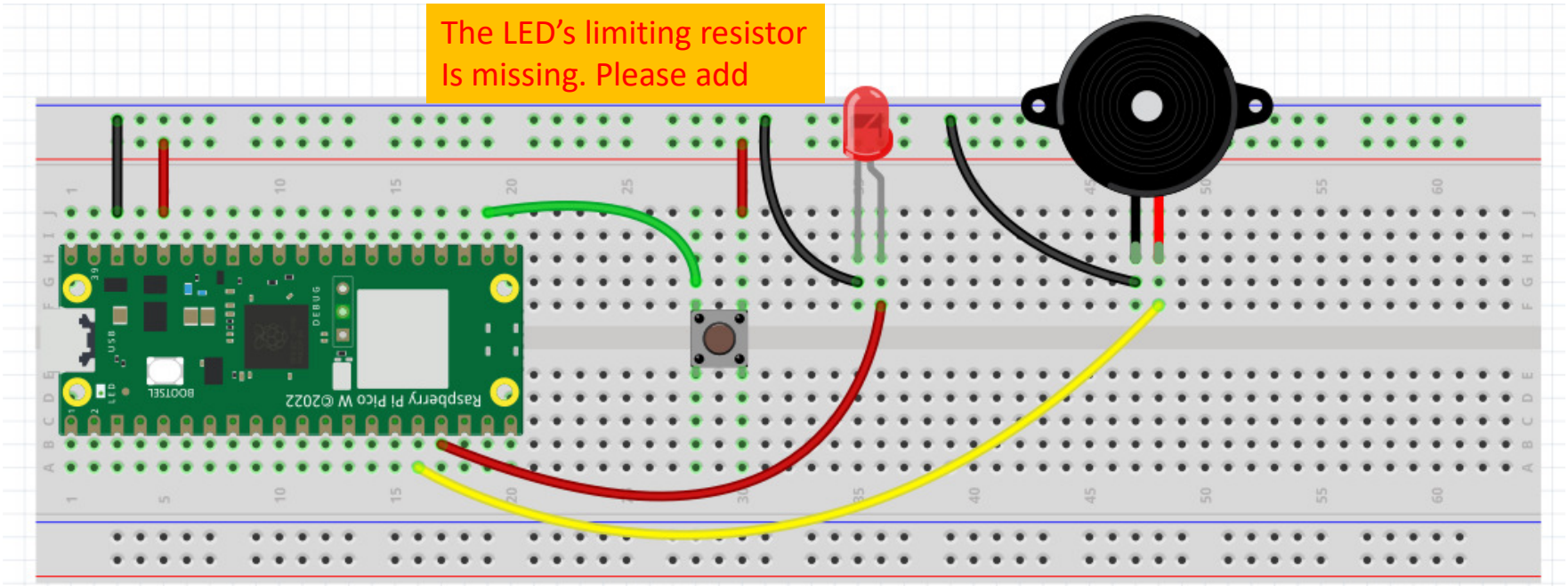
```
from machine import Pin
import time

pir_pin = Pin(16, Pin.IN)
led_pin = Pin(15, Pin.OUT)
buzzer_pin = Pin(14, Pin.OUT)

while True:
    if pir_pin.value() == 1:
        # Motion detected
        led_pin.on()  # Turn on the LED
        buzzer_pin.on()  # Turn on the buzzer
        time.sleep(1)  # Wait for 1 second
        led_pin.off()  # Turn off the LED
        buzzer_pin.off()  # Turn off the buzzer
```

In this code, we first import the necessary modules and initialize the pins for the PIR sensor, LED, and buzzer. We then enter an infinite loop and check if the PIR sensor's value is equal to 1, indicating that motion has been detected. If motion is detected, we turn on the LED and buzzer, wait for 1 second, and then turn them off. The loop will continue running, and the code will keep checking for motion.

Testing the Button



Button is on GPIO17 - PULL-DOWN (meaning pin is connected to 3V3 instead of Ground
Red LED is on GPIO13
Buzzer is on GPIO12

Program to Test the Button

When button is pressed
The LED and Buzzer
Will be activated.

python Copy code

```python
from machine import Pin
import time

button_pin = Pin(17, Pin.IN, Pin.PULL_DOWN)
led_pin = Pin(13, Pin.OUT)
buzzer_pin = Pin(12, Pin.OUT)

while True:
    if button_pin.value() == 1:
        # Button pressed
        led_pin.on()  # Turn on the LED
        buzzer_pin.on()  # Turn on the buzzer
        time.sleep(1)  # Wait for 1 second
        led_pin.off()  # Turn off the LED
        buzzer_pin.off()  # Turn off the buzzer
```

In this code, we first import the necessary modules and initialize the pins for the button, LED, and buzzer. We then enter an infinite loop and check if the button's value is equal to 1, indicating that the button has been pressed. If the button is pressed, we turn on the LED and buzzer, wait for 1 second, and then turn them off. The loop will continue running, and the code will keep checking for button presses.

Carefully Examine the code for both the PIR and Button
With the knowledge gain, construct a Burglar Alarm
Burglar Alarm is active when armed (use button and led)
Burglar Alarm is deactivated when disarmed with password
Refer to Raspberry Pi 4 course notes (Lesson 4) for Burglar Alarm
Drawing is attached.

https://github.com/ssgoh/Intro_To_Raspberry_Pi/blob/main/Pi%20Python%20Course%20Material%20-%20Rev%202.pdf