ASSIGNMENT 3

PYTHON CODE TO BLINK AN LED AND TRAFFIC LIGHT CONTROL

Assignment Date	14-10-2022
Student Name	SOURAV SARDAR
Student Roll Number	410619104037
Maximum Marks	2 Marks

QUESTION:

Write python code for blinking LED and Traffic lights for Raspberry pi.Only python code is enough, no need to execute in raspberry pi.
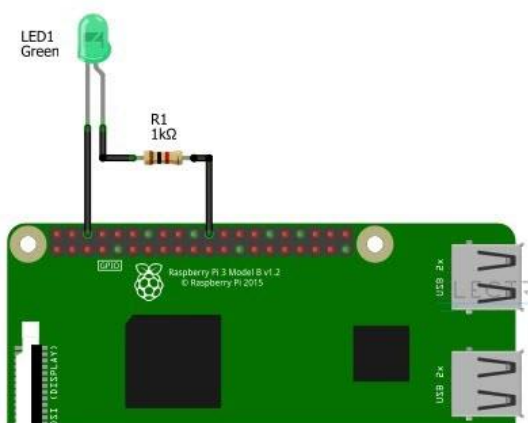
Solution: "BLINKING LED"

COMPONENTS REQUIRED:

- Raspberry Pi 3 Model B
- 5mm LED x 1
- 1KΩ Resistor (1/4 Watt) x 1
- Connecting wires
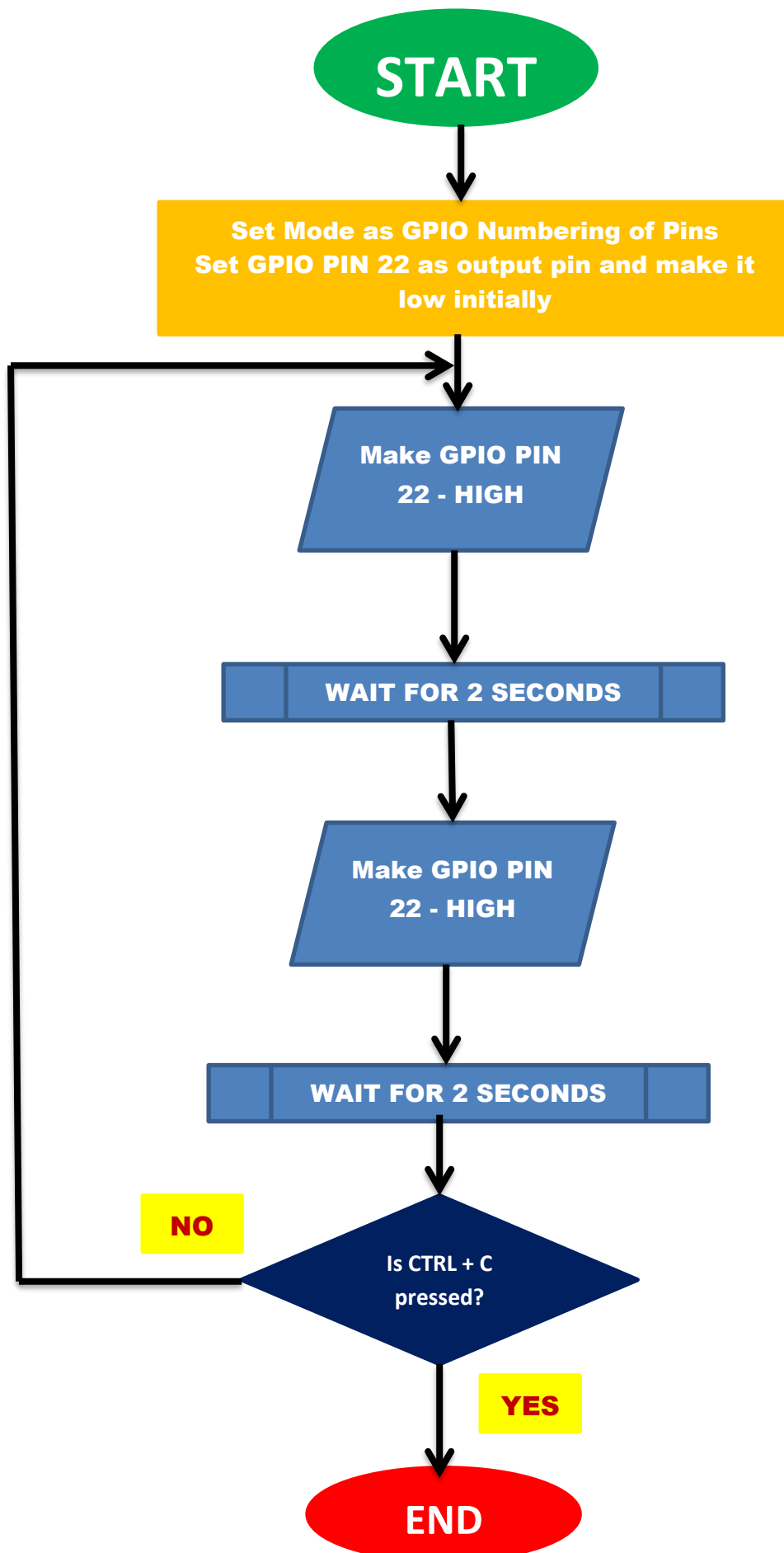- Miscellaneous (Computer, Ethernet cable, Power Supply for Raspberry Pi etc.)

CONNECTION:

The anode of the LED is connected to GPIO25 (Physical Pin 22) through a 1KΩ current limiting resistor. The cathode of the LED is connected to the GND Pin.

In this circuit, the GPIO Pins will act as source and provides with 3.3V when it is activated.

CIRCUIT DIAGRAM:

FLOW CHART:

```
                    ┌─────────────┐
                    │   START     │
                    └─────────────┘
                           │
                           ▼
        ┌──────────────────────────────────────────┐
        │  Set Mode as GPIO Numbering of Pins       │
        │  Set GPIO PIN 22 as output pin and make it │
        │  low initially                             │
        └──────────────────────────────────────────┘
                           │
                           ▼
                 ┌───────────────────┐
                 │  Make GPIO PIN    │
                 │  22 - HIGH        │
                 └───────────────────┘
                           │
                           ▼
                 ┌───────────────────┐
                 │ WAIT FOR 2 SECONDS │
                 └───────────────────┘
                           │
                           ▼
                 ┌───────────────────┐
                 │  Make GPIO PIN    │
                 │  22 - HIGH        │
                 └───────────────────┘
                           │
                           ▼
                 ┌───────────────────┐
                 │ WAIT FOR 2 SECONDS │
                 └───────────────────┘
                           │
                           ▼
            NO        ◇ Is CTRL + C ◇
         ◄────────────   pressed?
                           │
                          YES
                           │
                           ▼
                      ┌─────────┐
                      │   END   │
                      └─────────┘
```

PYTHON CODE:

```python
import RPi.GPIO as GPIO # RPi.GPIO can be referred as GPIO from now
import time
ledPin = 22    # pin22
def setup():
    GPIO.setmode(GPIO.BOARD)      # GPIO Numbering of Pins
    GPIO.setup(ledPin, GPIO.OUT)   # Set ledPin as output
    GPIO.output(ledPin, GPIO.LOW)  # Set ledPin to LOW to turn Off the LED
def loop():
    while True:
        GPIO.output(ledPin, GPIO.HIGH)   # LED On
        time.sleep(2.0)            # wait 2 sec
        GPIO.output(ledPin, GPIO.LOW)   # LED Off
        time.sleep(2.0)            # wait 2 sec
def endprogram():
    GPIO.output(ledPin, GPIO.LOW)     # LED Off
    GPIO.cleanup()            # Release resources


if __name__ == '__main__':       # Program starts from here
    setup()
    try:
        loop()
    except KeyboardInterrupt:  # When 'Ctrl+C' is pressed, the destroy() will be  executed.
        endprogram()
```

CONCLUSION:

Thus the Python code for LED Blinking is successfully executed.

"TRAFFIC LIGHTS"

SOLUTION:

COMPONENTS REQUIRED:

- Raspberry Pi 3 Model B
- Red 5mm LED x 1
- Green 5mm LED x 1
- Yellow 5mm LED x 1
- 75Ω Resistor x 3
- Connecting wires
- Miscellaneous (Computer, Ethernet cable, Power Supply for Raspberry Pi etc.)
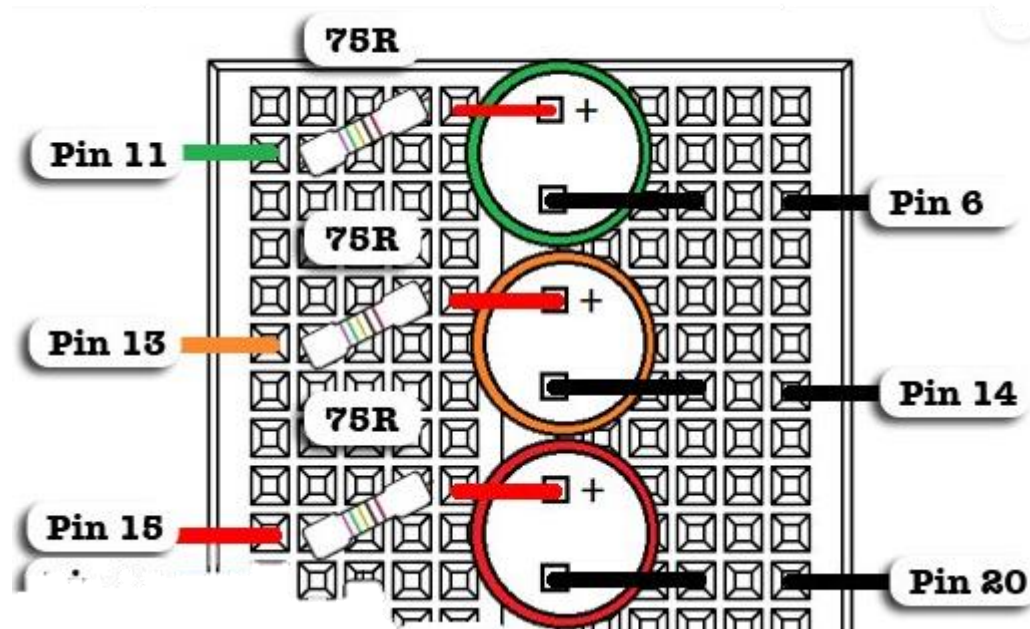
CONNECTION: 11 13 15

The anode of the Red LED is connected to Physical Pin 11 through a 75Ω current limiting resistor. The cathode of the LED is connected to the GND Pin 6.

The anode of the Green LED is connected to Physical Pin 13 through a 75Ω current limiting resistor. The cathode of the LED is connected to the GND Pin 14.
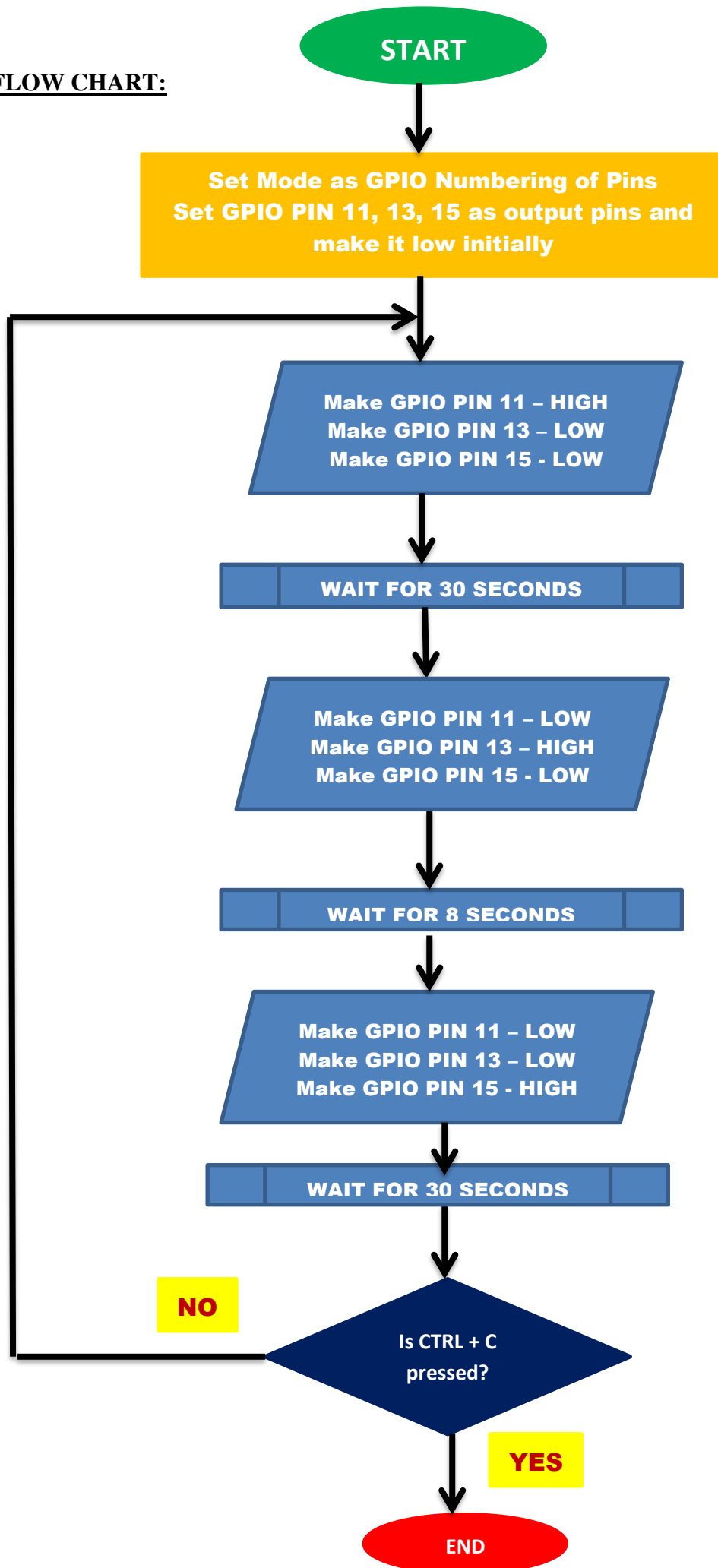
The anode of the Yellow LED is connected to Physical Pin 15 through a 75Ω current limiting resistor. The cathode of the LED is connected to the GND Pin 20.

In this circuit, the GPIO Pins will act as source and provides with 3.3V when it is activated.

CIRCUIT DIAGRAM:

FLOW CHART:

```
                    ┌─────────────┐
                    │    START    │
                    └──────┬──────┘
                           │
                           ▼
        ┌──────────────────────────────────────────┐
        │   Set Mode as GPIO Numbering of Pins       │
        │   Set GPIO PIN 11, 13, 15 as output pins   │
        │   and make it low initially                │
        └────────────────────┬───────────────────────┘
                             │
                             ▼
              ┌────────────────────────────┐
              │   Make GPIO PIN 11 – HIGH   │
              │   Make GPIO PIN 13 – LOW    │
              │   Make GPIO PIN 15 - LOW    │
              └──────────────┬─────────────┘
                             │
                             ▼
              ┌────────────────────────────┐
              │    WAIT FOR 30 SECONDS      │
              └──────────────┬─────────────┘
                             │
                             ▼
              ┌────────────────────────────┐
              │   Make GPIO PIN 11 – LOW    │
              │   Make GPIO PIN 13 – HIGH   │
              │   Make GPIO PIN 15 - LOW    │
              └──────────────┬─────────────┘
                             │
                             ▼
              ┌────────────────────────────┐
              │    WAIT FOR 8 SECONDS       │
              └──────────────┬─────────────┘
                             │
                             ▼
              ┌────────────────────────────┐
              │   Make GPIO PIN 11 – LOW    │
              │   Make GPIO PIN 13 – LOW    │
              │   Make GPIO PIN 15 - HIGH   │
              └──────────────┬─────────────┘
                             │
                             ▼
              ┌────────────────────────────┐
              │    WAIT FOR 30 SECONDS      │
              └──────────────┬─────────────┘
                             │
                             ▼
        NO               ◇─────────◇
       ◄─────────────────│ Is CTRL + C │
                          │  pressed?   │
                          ◇─────┬─────◇
                                │ YES
                                ▼
                         ┌─────────────┐
                         │     END     │
                         └─────────────┘
```

PYTHON CODE:

```python
import RPi.GPIO as GPIO # RPi.GPIO can be referred as GPIO from now
import time


red_led = 11    # pin11
yellow_led = 13    # pin13
green_led = 15    # pin15


def setup():
    GPIO.setmode(GPIO.BOARD)      # GPIO Numbering of Pins
    GPIO.setup(red_led, GPIO.OUT)   # Set red_led as output
    GPIO.setup(yellow_led, GPIO.OUT)   # Set yellow_led as output
    GPIO.setup(green_led, GPIO.OUT)   # Set green_led as output
    GPIO.output(red_led, GPIO.LOW) # Set red_led  to LOW to turn Off the Red LED
    GPIO.output(yellow_led, GPIO.LOW)  # Set yellow_led  to LOW to turn Off the
Yellow LED
    GPIO.output(green_led, GPIO.LOW)  # Set green_led  to LOW to turn Off the Green
LED


def loop():
    while True:
        #RED LED ON
        GPIO.output(red_led, GPIO.HIGH)   #RED LED On
        GPIO.output(yellow_led, GPIO.LOW)   # YELLOW LED Off
        GPIO.output(green_led, GPIO.LOW)   # Green LED Off
        time.sleep(30)              # wait 30 seconds
        #YELLOW LED ON
        GPIO.output(red_led, GPIO.LOW)   #RED LED Off
        GPIO.output(yellow_led, GPIO.HIGH)   # YELLOW LED On
        GPIO.output(green_led, GPIO.LOW)   # GREEN LED Off
```

```python
        time.sleep(8)          # wait 8 seconds
        #GREEN LED ON
        GPIO.output(red_led, GPIO.LOW)   #RED LED Off
        GPIO.output(yellow_led, GPIO.LOW)   # YELLOW LED Off
        GPIO.output(green_led, GPIO.HIGH)   # GREEN LED On
        time.sleep(30)            # wait 30 seconds


def endprogram():
    GPIO.output(red_led, GPIO.LOW)     # LED Off
    GPIO.output(yellow_led, GPIO.LOW)     # LED Off
    GPIO.output(green_led, GPIO.LOW)     # LED Off
    GPIO.cleanup()              # Release resources


if __name__ == '__main__':      # Program starts from here
    setup()
    try:
        loop()
    except KeyboardInterrupt:  # When 'Ctrl+C' is pressed, the destroy() will be  executed.
        endprogram()
```

CONCLUSION:

Thus the Python code for Traffic Light Control is successfully executed.