Al Hussein
Technical University
جامعة الحسين التقنية

Internet Of Things

PWM and Keypad

Prepared by:

Dr. Murad Yaghi
Eng. Malek Al-Louzi

School of Computing and Informatics – Al Hussein Technical University

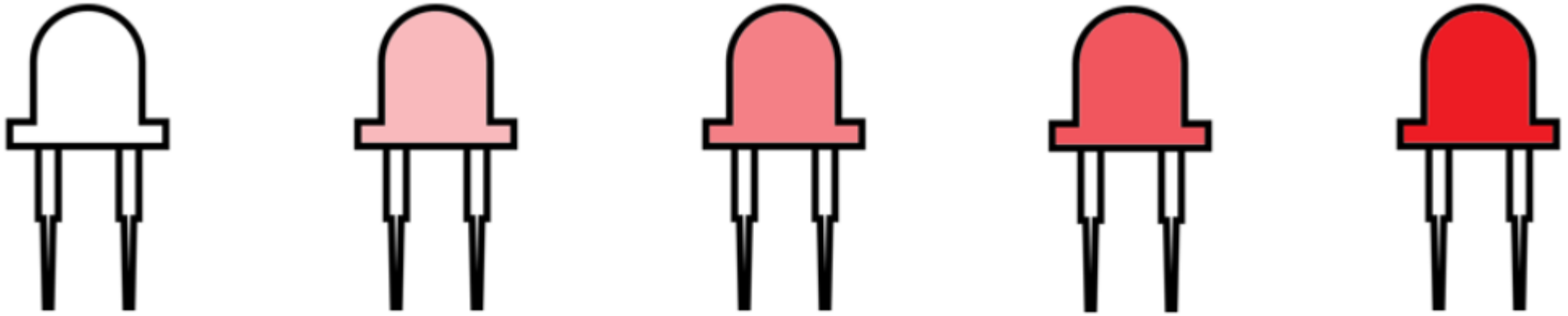
Fall 2024/2025

Introduction

- The Raspberry Pi GPIOs can be set either to output 0V or 3.3V, but they can’t output any voltages in between
- You can output “fake” mid-level voltages using pulse-width modulation (PWM)
- Which is how you’ll produce varying levels of LED brightness
- PWM is also useful for other applications like varying the speed of DC motors, setting the position of a servo motor, and much more

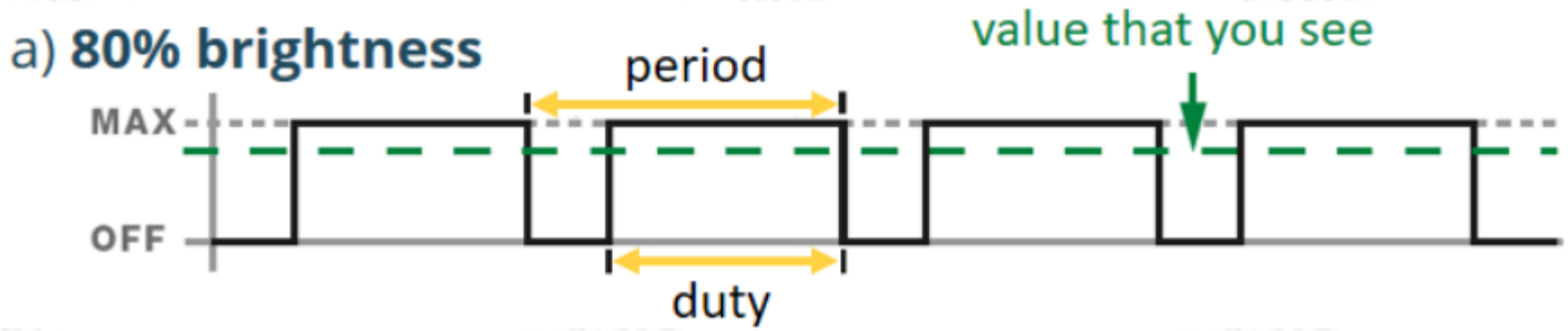
Introduction

- If you alternate an LED's voltage between HIGH and LOW very fast, your eyes can't keep up with the speed at which the LED switches on and off
- you'll simply see some gradations in brightness



How PWM works

- By producing an output that changes between HIGH and LOW at a very high frequency



- The duty cycle is the fraction of the time period at which LED is set to HIGH

How PWM works

b) **20% brightness**

c) **50% brightness**

d) **100% brightness**

e) **0% brightness**

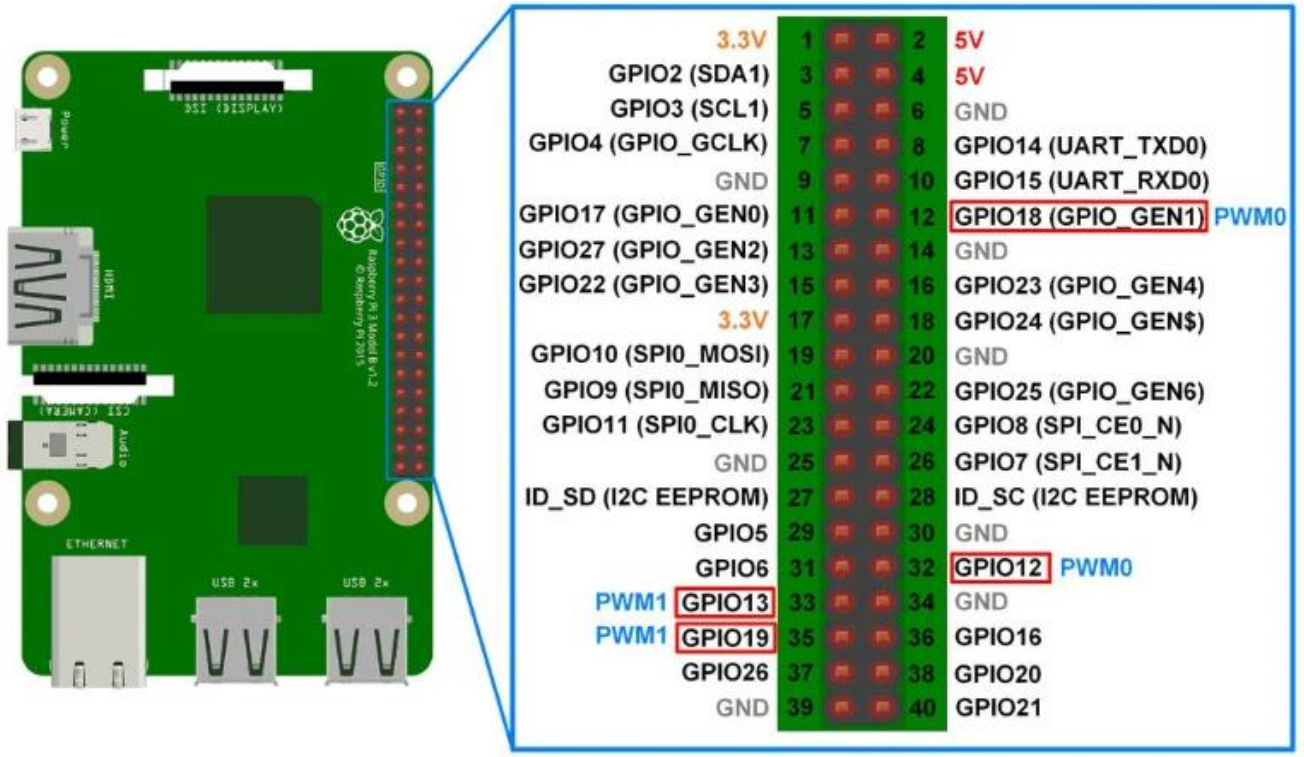
How PWM works

- A duty cycle of 50 percent results in 50 percent LED brightness, a duty cycle of 0 means the LED is fully off, and a duty cycle of 100 means the LED is fully on
- Changing the duty cycle is how you produce different levels of brightness

PWM Pins on the Raspberry Pi

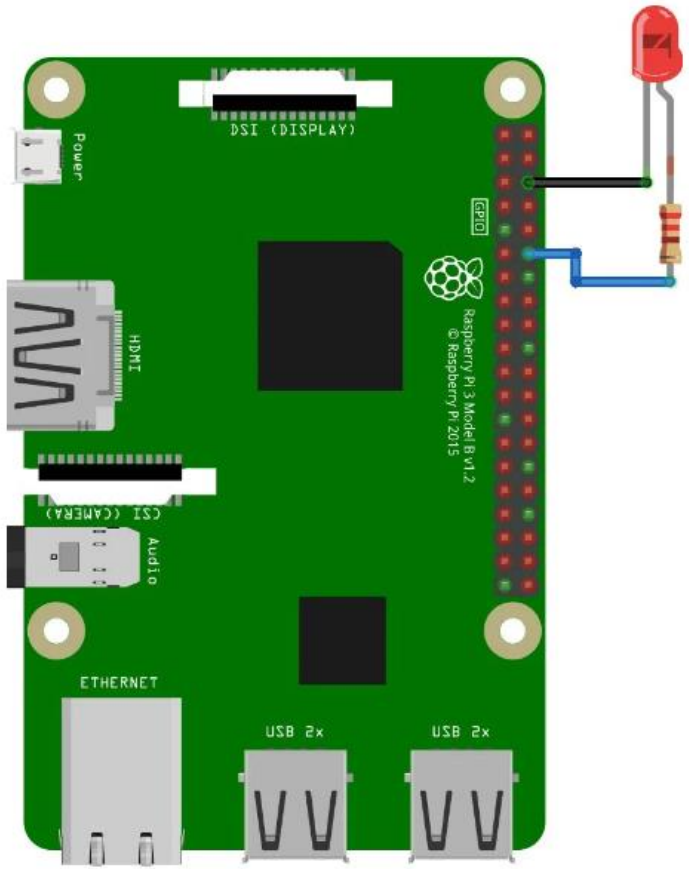
- The Raspberry Pi has 4 hardware PWM pins:
  - GPIO 12, GPIO 13, GPIO 18 and GPIO 19

GPIO Pin	PWM0/PWM1
GPIO12	PWM0
GPIO18	PWM0
GPIO13	PWM1
GPIO19	PWM1



Wiring the Circuit

- Wire an LED to one of the Raspberry Pi GPIOs. We'll connect one LED to GPIO 18 (pin 12)



Python Program

- **Create PWM Object:**
- Create an Object of class PWM which is a part of RPi.GPIO library
- In the full code in next slides, we have created Object of name pi_pwm, we can provide any name for Object
- **Syntax:**

```
pi_pwm = GPIO.PWM (Pin no., frequency)
```

Where,

Pin no. – PWM pin no on which PWM will be generated.

Frequency – frequency of PWM

Python Program

start (Duty Cycle)

It is used to start PWM generation of specified Duty Cycle.

ChangeDutyCycle(Duty Cycle)

This function is used to change the Duty Cycle of signal. We have to provide Duty Cycle in the range of 0-100.

ChangeFrequency(frequency)

This function is used to change the frequency (in Hz) of PWM.

stop()

This function is used to stop the PWM generation.

Python Full Program

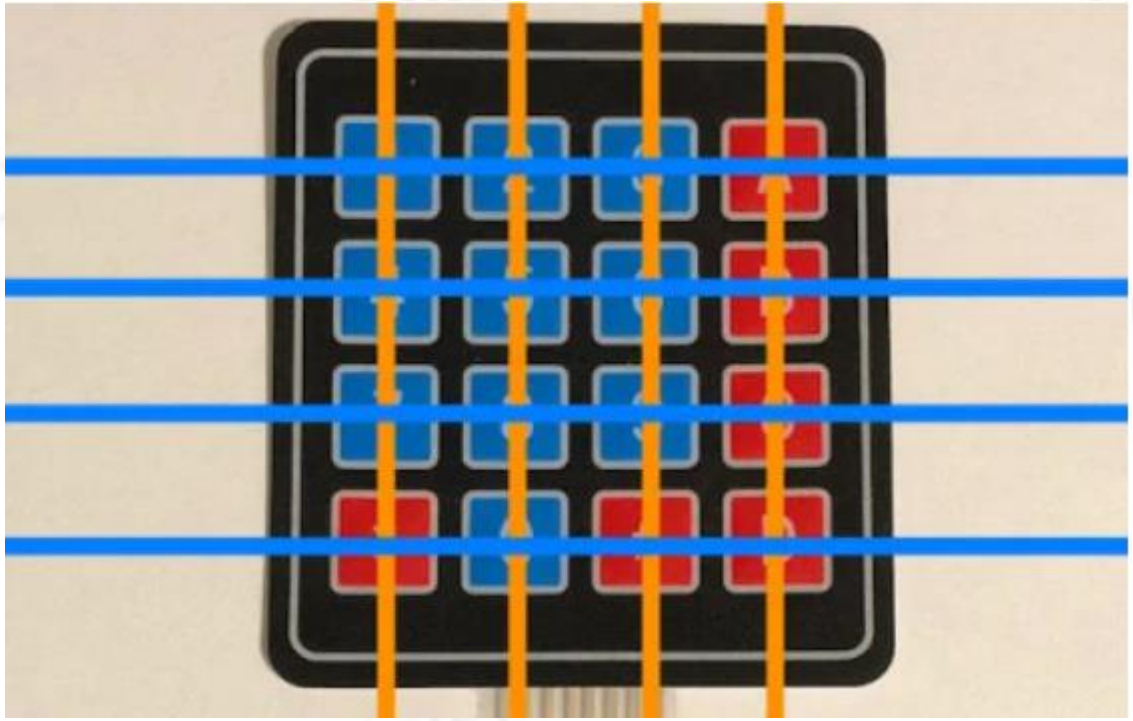
```
import RPi.GPIO as GPIO
from time import sleep

ledpin = 12                 # PWM pin connected to LED
GPIO.setmode(GPIO.BOARD)          #set pin numbering system
GPIO.setup(ledpin,GPIO.OUT)
pi_pwm = GPIO.PWM(ledpin,1000)       #create PWM instance with frequency
pi_pwm.start(0)                 #start PWM of required Duty Cycle
while True:
    for duty in range(0,101,1):
        pi_pwm.ChangeDutyCycle(duty) #provide duty cycle in the range 0-100
        sleep(0.01)
    sleep(0.5)

    for duty in range(100,-1,-1):
        pi_pwm.ChangeDutyCycle(duty)
        sleep(0.01)
    sleep(0.5)
```

Understanding the Keypad

- There are numerous ways for users to input data on a Raspberry Pi
- One of them is to use a 16-button keypad that contains the numbers from zero to nine as well as a few extra buttons

Understanding the Keypad

- The keypad that we want to be used can be divided into four rows and four columns like this:



Understanding the Keypad

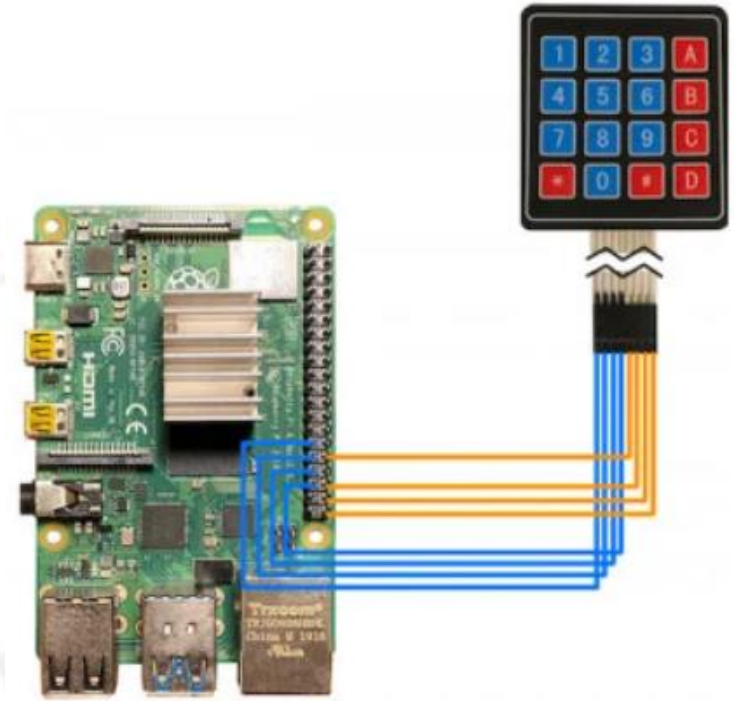
- To detect which button is pressed, the Raspberry Pi has to send a pulse to each of the four rows of the keyboard
- When a user presses a button that’s connected to the line which is currently pulled high, the corresponding column is also pulled high
- By decoding the combination of line and column, you can determine which button got pressed

Understanding the Keypad

- If a user, for example, presses the B button located on the second row in the fourth column, the Raspberry Pi detects this button press when it sends a pulse to the second line and then checks which of the four columns was pulled high

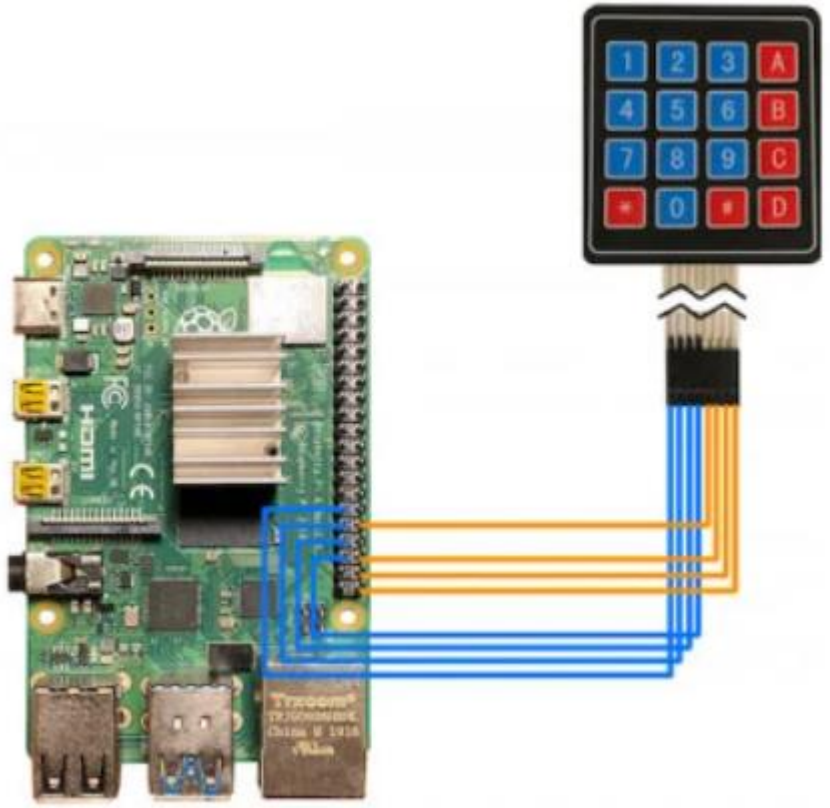
Connecting the Keypad to the Raspberry Pi

- Keypads that operate this way don’t need any power to work
- This means that you can simply connect all eight data lines of the keypad to any eight GPIO pins on your Raspberry Pi



Connecting the Keypad to the Raspberry Pi

- The blue connections correspond to the rows, and the orange ones represent the columns



A Simple Code Example

- Once you made the connections, it’s time to run a simple test program that will print the buttons, a user presses on the keypad, to the Raspberry Pi’s console

- The full code is on the next slide

```python
import RPi.GPIO as GPIO
import time

L1 = 5
L2 = 6
L3 = 13
L4 = 19

C1 = 12
C2 = 16
C3 = 20
C4 = 21

GPIO.setmode(GPIO.BCM)

GPIO.setup(L1, GPIO.OUT)
GPIO.setup(L2, GPIO.OUT)
GPIO.setup(L3, GPIO.OUT)
GPIO.setup(L4, GPIO.OUT)

GPIO.setup(C1, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
GPIO.setup(C2, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
GPIO.setup(C3, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
GPIO.setup(C4, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)

def readLine(line, characters):
    GPIO.output(line, GPIO.HIGH)
    if(GPIO.input(C1) == 1):
        print(characters[0])
    if(GPIO.input(C2) == 1):
        print(characters[1])
    if(GPIO.input(C3) == 1):
        print(characters[2])
    if(GPIO.input(C4) == 1):
        print(characters[3])
    GPIO.output(line, GPIO.LOW)

try:
    while True:
        readLine(L1, ["1","2","3","A"])
        readLine(L2, ["4","5","6","B"])
        readLine(L3, ["7","8","9","C"])
        readLine(L4, ["*","0","#","D"])
        time.sleep(0.1)
except KeyboardInterrupt:
    print("\nApplication stopped!")
```

A Simple Code Example

- As you can see, the test program contains a method called **readLine**
- The **readLine** method sends the pulse discussed above to a single line and then checks which of the buttons got pressed while the line is pulled high
- This is simply repeated for all four rows
- The method also takes a list of symbols that the buttons correspond to

Any Questions???