Internet of Things Laboratory

Assignment 7

Prajakta Deokule

C22019221332

4329

Problem Statement: Write an application for Blinking LED and play a Buzzer using Raspberry Pi.

1.Differentiate between Raspberry Pi and Arduino.

	Raspberry Pi	**Arduino**
License	Hardware and software of Raspberry Pi is closed source.	Arduino is an open source project. Both its hardware and software design are open source.
Control Unit	From ARM family	From Atmega family
RAM	Requires large RAM(more than 1GB)	Requires less RAM(2KB)
Clock Frequency	Up to 1.4GHz	16MHz (Arduino UNO)
Logic Level	Raspberry Pi's logic level is 3V	Arduino's logic level is 5V
Power Consumption	Raspberry Pi consumes about	Arduino consumes about 200MW of

	700MW of power	power
Based On	Raspberry Pi is based on a microprocessor	Arduino is a microcontroller
Hardware Structure	Complex hardware structure	Simple hardware structure
Software	Raspberry Pi supports its own Linux based operating system Raspberry Pi OS. You can also install the OS you like.	Arduino boards are programmable using C/C++ languages.
Internet	Raspberry Pi has built in Ethernet port and wifi support	Arduino does not have internet support. You need additional modules or shields to connect it to the Internet.
CPU Architecture	64 bit	8 bit
Cost	Raspberry pi boards are expensive	Arduino boards are cheaper
Current Drive strength	Lower current drive strength	Higher current drive strength
How they handle power drop	Raspberry Pi requires the same care as a PC. You have to shut down the OS properly.	Arduino devices begin executing code when they are turned on. Hence when power is turned off abruptly the OS does not become corrupt.
Capability	Raspberry Pi can perform multiple tasks simultaneously	Arduino is generally used to perform single and simple

		tasks repeatedly
Wireless Connectivity	Raspberry Pi supports Bluetooth and wifi	Arduino does not support Bluetooth or wifi
Applications	Robot controllers, game servers, stop motion cameras etc	Traffic light countdown timer, Parking iot counter, Weighing machines, etc

2. Differentiate between Raspberry Pi and Beagle Bone Black.

Parameter	Raspberry Pi	Beagle Bone Black
Model Tested	It uses Model B version.	It uses Rev A5 version
Processor Type	Uses ARM11 processor	Uses ARM Cortex-A8 processor
RAM	It uses 700 MHz for processing	It uses 1 GHz for processing
Min Power	It requires power supply of 700mA (3.5W)	It uses minimum power of 210mA(1.05W) for its functioning.
GPIO Pins	It has 12 GPIO pins	It has 69 GPIO pins
Flash	It has dedicated SD Card socket for loading operating system.	It uses 4GB (microSD) for loading OS and data storage
USB Master	It has 2 USB 2.0 on board.	It has 1 USB 2.0 on board.
Dev IDE	It uses IDLE, Scratch,	It uses Python,

	Squeak, Linux to perform tasks.	Scratch, Squeak, Cloud9/Linux to perform a particular task.
Audio Output	Supports HDMI, Analog audio output	Uses Analog output for audio
Video Output	It uses HDMI and composite output for video.	No specific video output
UART	It uses 1 UART to transmit and receive serial data	It uses 5 UART to transmit and receive serial data.
No of I/O pins	8 digital and 0 analog pins	65 digital and 7 analog pins

3.What is BCM mode in Raspberry Pi?

BCM means Broadcom chip specific pin numbers. These pin numbers follow the lower-level numbering system defined by the Raspberry Pi's Broadcom-chip. The BCM pin mapping refers to the GPIO pins that have been directly connected to the System on a Chip (SoC) of the Raspberry Pi.We have direct links to the brain of Raspberry Pi to connect sensors and components for use.

4.How is the Raspberry Pi board different from BBB in programming?

- In Raspberry Pi, we use the BCM mode for programming. Such a mode setting is not applicable for Beaglebone Black.
- The python library used for Raspberry Pi programming is RPi.GPIO, whereas the library used in case of BBB is Adafruit_BBIO.GPIO.

5. Write a code for blinking LED using Raspberry Pi.

import RPi.GPIO as GPIO

import time

led = 8

```
GPIO.setmode(GPIO.BCM)

GPIO.setup(led, GPIO.OUT)

try:

        while True:

                GPIO.output(led, GPIO.HIGH)

                time.sleep(2)

                GPIO.output(led, GPIO.LOW)

`               time.sleep(2)

except:

        print ("Error Occurred")
```

Code

```
import time

import RPi.GPIO as GPIO

pin=25

buzzer=3

GPIO.setwarnings(False)

 GPIO.setmode(GPIO.BCM)

GPIO.setup(buzzer,GPIO.OUT)

GPIO.setup(pin,GPIO.OUT)

 while True:

         GPIO.output(pin,1)

        GPIO.output(buzzer,1)

        time.sleep(2)

        GPIO.output(pin,0)

        GPIO.output(buzzer,0)

        time.sleep(2)

GPIO.cleanup()
```