Experiment No 1: Arduino basic setup, how to install it and use it, shields to extend the functionality of an Arduino based system.

Objectives: To study IOT, their characteristics of components and basic awareness of Arduino
Hardware Requirement: Arduino basic kit
Software Requirement: Can be installed on LINUX and Windows

Experiment Setup :

Programming with Arduino UNO:
Required Materials: To follow along with Arduino we will need the following materials.

- A computer (Windows, Mac, or Linux)
- An Arduino-compatible microcontroller
- a USB A-to-B cable, or another appropriate way to connect your Arduino-compatible microcontroller to your computer.

In our lab :

- Windows System
- IoT Universal Development Kit provided by Techgraylogix

Procedure:

1. Download the Arduino IDE from their website. They have installation instructions. Make sure you download the version that matches your operating system.

2. Installation in Windows:

- The Windows version of Arduino is offered in two options: an installer or a zip file.
- The installer is the easier of the two options, just download that, and run the executablefile to begin the installation.
- In Lab system : the installer is available in C:/IoTDevelopmentKit/Arduino Software/arduino installer .exe file is available.
- When you're prompted to install a driver during installation, select "Install". This will install drivers for Arduino specific boards (like the Uno, Nano, etc.) that you may use in the future.

3. Connecting Arduino:

- Power up your Arduino by connecting your Arduino board to your computer with a USB cable. You should see an LED labeled 'ON' light up.
- After the installation process is complete, there will be an Arduino icon on the Desktop. Or check on the search icon and write "arduino". If you have found the arduino icon, runthe application.
- Once the sketch is written it should be uploaded into the arduino board. This process needs to ensure that the port is enabled in the Arduino IDE. If not then troubleshooting needs to be done. Either by installing the required drivers or by changing the cable.
- Select the type of Arduino board you're using: **Tools** > **Board** > **Arduino Uno**.
- Select the serial/COM port that your Arduino is attached to: **Tools** > **Port** > **COMxx.**
- ESP8266 module can be connected to the universal development board for wifi adaption.

Conclusion: Studied Arduino basic setup, how to install it and use it, shields to extend thefunctionality of an Arduino based system.

Experiment No 2 : Blinking LED sketch with Arduino.

Objectives: To study how to write a sketch for blinking LED with Arduino

Hardware Requirement: Arduino basic kit

Software Requirement: Arduino IDE

Experimental Setup:

1. This experiment requires us to use arduino universal development kit from Techgraylogix.
2. Connect the kit to the system.

Procedure:

- Open the Blink example sketch by going to: **File** > **Examples** > **01.Basics** > **Blink**.
- OR choose from the sample code from IoT Developemnt Kit folder in C drive.
- With your Arduino board connected, and the Blink sketch open, press the "**Upload**"button.

  After a second, you should see some LEDs flashing on your Arduino, followed by themessage "**Done Uploading**" in the status bar of the Blink sketch.
- If everything worked, the onboard LED on your Arduino should now be blinking! Youjust programmed your first Arduino!

Conclusion: Studied how to write and upload sketch onto arduino board and observed the desiredresult.

Experiment No 3: Spinning a Stepper Motor and Motor Speed Control Sketch

Objective:

1. Learn the technicality of DC motor
2. Developing an arduino sketch to run and control the DC motor.
3. Designing the circuit connection using arduino and DC motor.

Hardware Required:

1. Arduino Universal Development Kit
2. DC motor

Experimental Setup and Procedure:

1. Connect DC motor to motor 1 (shown in figure 1) slot of arduino universal developmentkit as shown in figure 2.
2. Choose the Motor Control sketch available under Sample codes folder in IoT Development Kit folder in C drive.
3. Connect the arduino kit to PC.
4. Compile and upload the motor control sketch to the arduino board.
5. You will see the motor rotating in clockwise and anticlockwise directions.
6. In the sketch the pin details are mentioned as for motor 1 slot connects to pin number 10and 11 of arduino and motor slot 2 connects to pin number 7 and 9 of arduino.

Conclusion : We learnt the working of a DC motor controlled by Arduino.

Experiment 4: Getting started with Raspberry Pi, Install Raspbian on your SD card.

Objectives:

1. To study basics of Raspberry Pi
2. To study how to install Raspbian on an SD card for booting Raspberry Pi.

Hardware Requirement: Raspberry Pi Kit: SD card and card reader

Software Requirement: Raspbian imager

Experiment Setup:

1. Insert SD card to card reader
2. Plug in the SD card to the system
3. Download Raspberry Pi imager to proceed with the installation of OS

Procedure:

Install Raspberry Pi OS using Raspberry Pi Imager

Step 1: Raspberry Pi imager can be downloaded for windows, MAC, Linux. The following website has the corresponding links: Raspberry Pi downloads page

Step 2: When the download finishes, click it to launch the installer
Step 3: Format the SD card. Anything that’s stored on the SD card will be overwritten during formatting.You should back up the files first to prevent you from permanently losing them. Also chose to format SD card as FAT32 before writing OS to it.
Step 4: Follow the instructions to install and run the Raspberry Pi Imager

- Insert your SD card into the computer or laptop SD card slot
- In the Raspberry Pi Imager, select the OS you want to install and the SD card you want to install. NOTE: In the lab we chooseto install lite version.

Step 5: You can now insert the SD card into the Raspberry Pi and power it up. When your Raspberry Pi boots for the first time a configuration wizard will run that allows you to set upyour Raspberry Pi.

Conclusion : Studied the basics of Raspberry Pi and the procedure for installing Raspbian OS and boot Raspberry Pi.

Experiment No 5: Connect an LED to GPIO pin 24 and a Switch to GPIO 25 and controlthe LED with the switch.

Objectives:

1. **Learn GPIO connectivity**
2. **Learn to program switch controlled LED**

Hardware Required: For the circuit of controlling the LED with a button on the Raspberry Pi 4, we need Raspberry Pi 4,LED, a resistor of 220 ohms, Push-button, Connecting wires

Experimental Setup and Procedure:

1. Step 1: For hardware configuration of the above circuit diagram, we will first connect theRaspberry Pi 4 and a LED on the breadboard.
2. Step 2: After this, we will connect the push-button(either of two legs or four legs) on thebreadboard.
3. Step 3: Then we will connect a cathode terminal of LED and any leg terminal of push-button to the series short terminal of the breadboard, and connect this short terminalof the breadboard with the ground pin of Raspberry Pi 4
4. Step 4: Connect the anode terminal of the LED with the GPIO pin 24 of Raspberry Pi 4
5. Step 5: In the last, connect the remaining second terminal of the push-button with theGPIO 25 pin of the Raspberry Pi 4.

The Python code for controlling the LED with the button on Raspberry Pi 4 is simple,we will first make a file with the name of “switch.py” and open it with nano editor:

```
from gpiozero import LED
import Button
led = LED(24)
button = Button(25)
while True:
            button.wait_for_press()
                             led.on()
            button.wait_for_release()
                              led.off()
```

Conclusion: We learnt programming switch controlled LED system using RPi.

Experiment No 6: Understanding the connectivity of Arduino with IR sensor. Write an application todetect obstacle and notify user using LEDs.

Objective:

1. **To understand the connectivity of Arduino circuit with IR sensor**
2. **To Understanding how to read the Interference of object**
3. **To Understanding how to switch on the Buzzer and LED on detecting interference**

Hardware Requirements : IoT Universal Development Kit, IR sensor and buzzer modules.

Software Requirements : Arduino IDE

Experimental setup and Procedure:

STEPS to setup the circuit:

1. Use the TECHGRAYLOGIX universal development board.
2. Connect IR sensor to DENSE 1 slot
3. Connect Buzzer to the Buzzer slot
4. Place the jumper wires as per the specification mentioned in the sketch.
5. Choose the IR sensor obstacle detection sketch from the sample code.
6. Compile and upload the code to arduino uno
7. Test the circuit by placing obstacle in front of IR sensor and buzzer must turn on when there is obstacle.

Conclusion: IR sensor will sense the obstacle and buzzer will be turned ON on detection of obstacle. Working of IR sensor and buzzer in arduino kit is observed.

Experiment No 7: How to use Python-based IDE for the Raspberry Pi and how to trace and debug Python code on the device.

Objective:

1. **To understand Connectivity and configuration of Raspberry-Pi**
2. **To Understanding GPIO and its use in program**
3. **To understand board circuit with basic peripherals, LEDS**
4. **To trace and debug Python code on the device**

Programming the pins: LED Blink Programming using Raspberry Pi:

- In Thonny Python IDE or nano(in lite version), create a python script. For example : blink.py
- There are two ways to do the blink program.
  - Using gpiozero library
    - This library is easiest way to blink LED. At the same time it is simple to understand and code.
  - RPi.GPIO library
    - This is old library and similar to arduino.
- blink.py will contain the following code (nano blink.py # for creating the file):

```
#blinking with gpiozero library
from gpiozero import LED
from time import sleep
led = LED(14)
while True:
   led.on()
   print('LED ON')
   sleep(1)
   led.off()
   print('LED OFF')
   sleep(1)
```

To execute : python3 blink.py #execute the file

1. First we will import LED from gpiozero library and sleep from timelibrary.
2. create an object called led that refers to GPIO 14. You can see this numberon the pinout chart.
3. In the loop, call on() and off() functions to turn the led on and off. A delayof 1 second is introduced.

Conclusion: We Studied Connectivity and configuration of Raspberry-Pi circuit with basic peripherals, LEDS and to write python code for blinking LED. Also studied how to trace and debug the python code in Raspberry Pi.

Experiment No 8: Understanding connectivity of Raspberry-Pi /Beagle board with camera

Objectives:

1. **To understand the configuration of Pi camera with RaspberryPi**
2. **To understand the working of Pi camera**
3. **To learn commands and code to interface and control Pi camera**

Experimental setup:

Connect the Camera Module

Ensure your Raspberry Pi is turned off.

1. Locate the Camera Module port
2. Gently pull up on the edges of the port's plastic clip
3. Insert the Camera Module ribbon cable; make sure the connectors at the bottom of the ribbon cable are facing the contacts in the port.
4. Push the plastic clip back into place.

Turn on RaspberryPi. Now, we can use Pi Camera for capturing images and videos using Raspberry Pi. Before using the Pi Camera, we need to enable the camera for its working.

Procedure:

Enable Camera functionality on Raspberry Pi:

For enabling camera in Raspberry Pi, open raspberry pi configuration using following command,

```
$ sudo raspi-config
```

then select **Interface options** in which select **legacy camera** option to enable its functionality.Also enable I2C option in the interfaces.

Then reboot Raspberry Pi to apply the changes.

The following command captures an image and saves in the specified folder.

```
$ raspistill -o image.jpg
```

This will capture the image and save it in image.jpg

We can record a video with the Camera Module by using the following raspivid command:

```
$ raspivid -o video.mp4
```

We can capture images using Python.

The Python picamera library allows you to control your Camera Module and create amazingprojects.

Python Program for Image Capture

```
from picamera import

PiCamerafrom time import

sleep
```

```
camera = PiCamera()

camera.start_preview(

)sleep(5)

camera.stop_preview(

)
```

Functions Used :

To use picamera python based library we have to include it in our program as given

belowimport picamera

This picamera library has PiCamera class for camera module. So, we have to create object for **PiCamera** class.

PiCamera Class

To use Pi Camera in Python on Raspberry Pi, we can use PiCamera class which has different APIs for camera functionality. We need to create object for the PiCamera class.

The above PiCamera class has different member variables and functions which we can access bysimply inserting a dot (.) in between object name and member name.

It has functions like :

1. **capture() :** It is used to capture images using Pi Camera.The capture() function has different parameters which we can pass for different operations likeresize, format, use_video_port, etc.
2. **resolution= (width,height):** It sets the resolution of camera at which image captures, video records and preview will display. The resolution can be specified as (width, height) tuple, as a string formatted **WIDTHxHEIGHT**
3. **Annotate_text = "Text" :** It is used to add text on image, video, etc.
4. **start_preview():** It displays the preview overlay of the default or specified resolution.
5. **stop_preview():** It is used to close the preview overlay.

Conclusion : Learned to connect RPi camera with Raspberry Pi and capture images and videos.

Experiment No 9: Interfacing Arduino with Cloud (Thingspeak API).

Objectives:

1. **Understand the need of cloud services in IoT.**
2. **Understand the procedures of creating channels and writing data to Thingspeak.**
3. **Understand the Arduino configurations for wifi connectivity**
4. **Learn to send temperature data sensed by DHT11 sensor to Thingspeak channel.**

Hardware & Software Requirements: Arduino Uno + ESP8266 Connectivity with Cloud, ThingSpeak.

Experimental Setup and Procedure:

Step 1: In Techgraylogix Universal Development Kit, we use DHT11 sensor and ESP8266 module.
Step 2: Place the DHT11 sensor on LM35 slot in the board and place ESP8266 module into wifi slot.

Step 3: Open IoT Development Kit folder in C drive of lab PC. Got to IoT ThingSpeak folder.Open the the sketch. Copy and paste the following code into the sketch area (Replace the entirecode with this new code:

Step 4: Compile and upload the code. Make sure that your personal hotspot is ON and esp8266 module connects to it. Once it is connected, the Temperature will be written into the channel.

Conclusion: We learned how to interface Arduino with Cloud (Thingspeak API)