

Name: Manjiri Cole

Class: TE-IT Div.: A Roll No.: 12

Subject: SI

Topic:

Page No.: 2

Date: 4/3/24

### Assignment-1

1. Apply the Analog Read Serial in Potentiometer to print its state out to the Arduino Serial Monitor in JupyterLab.

```
→ void setup()
{
  // initialize serial communication at 9600 bits per sec
  Serial.begin(9600);
}
// loop routine runs over and over again forever.
void loop() {
  // read the input on analog pin 0:
  int sensorValue = analogRead(A0);
  // print out the value you read:
  Serial.println(sensorValue);
  delay(1); // delay in between reads for stability
}
```

2. Apply the LED Blink example in Arduino IDE.

```
→ void setup()
{
  pinMode pinMode(LED_BULLETIN, OUTPUT);
}
void loop()
{
  digitalWrite(LED_BULLETIN, HIGH);
  delay(1000);
  digitalWrite(LED_BULLETIN, LOW);
  delay(1000);
}
```

3.) Identify the components required to take readings from DHT11 Temperature/Humidity Sensors and specify the Arduino IDE Serial Monitor.

→ Arduino Uno, DHT11 Temperature/Humidity, Jumper Wires, Breadboards, Cable (USB 2.0)

```
#include <dht.h>
```

```
#define dht_apin A0 // Analog Pin sensor is connected to  
dht DHT;
```

```
void setup()
```

```
{
```

```
Serial.begin(9600);
```

```
delay(500); // Delay to let system boot
```

```
Serial.println("DHT11 Humidity & temperature Sensors  
  \n\n");
```

```
delay(1000); // wait before accessing sensor
```

```
} // end "setup()"
```

```
void loop() {
```

```
// Start of Program
```

```
DHT.read1(dht_apin);
```

```
Serial.println("Current humidity = ");
```

```
Serial.println("DHT humidity = ");
```

```
Serial.print("%");
```

```
Serial.print("temperature = ");
```

```
Serial.print("DHT temperature");
```

```
Serial.print("C");
```

```
delay(5000) // wait 5 seconds before accessing  
sensor again
```

```
// Fastest should be once every two seconds.
```

```
} // end loop.
```



Name : \_\_\_\_\_

Subject : \_\_\_\_\_

Topic : \_\_\_\_\_

Class : \_\_\_\_\_

Div : \_\_\_\_\_

Roll No. : \_\_\_\_\_

Page No. : \_\_\_\_\_

Date : \_\_\_\_\_

4. Identify the components required for ultrasonic sensor with Arduino?

→ To set up an ultrasonic sensor with an Arduino, you'll need the following components:

Arduino board (eg. Arduino Uno, Arduino Nano, etc.)

Ultrasonic sensor module (commonly HC-SR04).

Jumper wires

Breadboard (optional but recommended for prototyping)

Once you have these components, here's how you can connect them:

Connect the VCC pin of the ultrasonic sensor module to the 5V pin on the Arduino.

Connect the GND pin of the ultrasonic sensor module to the GND pin on the Arduino.

Connect the Trig pin of the ultrasonic sensor module to any digital pin on the Arduino (eg. pin 7).

Connect the Echo pin of the ultrasonic sensor module to any digital pin on the Arduino (eg. pin 6).

If you're using a breadboard, use jumper wires to make the connections between the components.

If not, you can directly connect them using jumper wires.

5. Compare IEEE 802.11, IEEE 802.15.1 and IEEE 802.15.4.

→ IEEE 802.11 - WLAN / Wifi

Wireless ~~WAN~~ LAN (WLAN also known as Wifi) is a set of low freq, terrestrial, network technologies for data communications. The WLAN standard

operates on the 2.4GHz and 5GHz Industrial, Science and Medical (ISM) frequency bands. It is specified by the IEEE 802.11 standard and it comes in many different variations like IEEE 802.11 a/b/g/n. The application of WLAN has ~~not~~ been most visible in consumer market where most portable computers support at least one of the variations.

**IEEE 802.15.1 - Bluetooth:** The IEEE 802.15.1 standard is the basis for the Bluetooth wireless communication technology. Bluetooth is a low tier, ad hoc, terrestrial wireless standard for short range communication. The technology operates with three different classes of devices: class 1, class 2, class 3 where the range is about 100 meters, 10 meters and 1 meter respectively. Wireless LAN operates in the same 2.4GHz frequency band as Bluetooth, but the two technologies use different signaling methods which should prevent ~~interference~~ interference.

**IEEE 802.15.4 - ZigBee:** ZigBee is a low tier, ad hoc, terrestrial, wireless standard in some ways similar to Bluetooth. The IEEE 802.15.4 standard is commonly known as ZigBee, but ZigBee has some features in addition to 802.15.4. It operates in the 868MHz, 915MHz, and 2.4GHz ISM Bands.

Standard	Ad hoc	Infrastructure
802.11a/b/g/n	Yes	Yes
802.15.1	Yes	No
802.15.4	Yes	Yes/No