### 1. Assignment 1: TinkerCad LED Blink:

### // C++ code

//

int led1Pin = 4; // LED1 connected to digital pin 4

void setup()

{

pinMode(led1Pin, OUTPUT); // sets the digital pin as output

}

void loop()

{

digitalWrite(led1Pin, HIGH); // sets the LED on

delay(1000); // waits for a second

digitalWrite(led1Pin, LOW); // sets the LED off

delay(1000); // waits for a second

}

### 2. Assignment 2: Arduino Traffic Light:

void setup()

{

  pinMode(3, OUTPUT); // Red LED is HIGH

  pinMode(4, OUTPUT); // Green LED is HIGH

  pinMode(2, OUTPUT); // Yellow LED is HIGH

}

void loop()

{

  // Turn On red signal

  digitalWrite(2, LOW);

  digitalWrite(4, HIGH);

  delay(5000); // Wait for 5000 millisecond(s)

  digitalWrite(4, LOW);

  digitalWrite(3, HIGH);

  delay(5000); // Wait for 5000 millisecond(s)

  digitalWrite(3, LOW);

  digitalWrite(2, HIGH);

  delay(5000); // Wait for 5000 millisecond(s)

}

### 2.1. Assignment 2: Arduino Traffic Light:

// C++ code

//

int led1Pin = 4; // LED1 connected to digital pin 13

int led2Pin = 3; // LED2 connected to digital pin 12

int led3Pin = 2; // LED3 connected to digital pin 11

void setup()

{

pinMode(led1Pin, OUTPUT); // sets the digital pin as output

pinMode(led2Pin, OUTPUT); // sets the digital pin as output

pinMode(led3Pin, OUTPUT); // sets the digital pin as output

}

void loop()

{

digitalWrite(led1Pin, HIGH); // sets the LED on

delay(1000); // waits for a second

digitalWrite(led1Pin, LOW); // sets the LED off

delay(1000); // waits for a second

digitalWrite(led2Pin, HIGH); // sets the LED on

delay(1000); // waits for a second

digitalWrite(led2Pin, LOW); // sets the LED off

delay(1000); // waits for a second

digitalWrite(led3Pin, HIGH); // sets the LED on

delay(1000); // waits for a second

digitalWrite(led3Pin, LOW); // sets the LED off

delay(1000); // waits for a second

}

### 2.2. Assignment 2: Arduino Traffic Light:

**int red1 = 13;**

**int yellow1 = 12;**

**int green1 = 11;**

**int red2 = 10;**

**int yellow2 = 9;**

**int green2 = 8;**

**int red3 = 7;**

**int yellow3 = 6;**

**int green3 = 5;**

**int red4 = 4;**

**int yellow4 = 3;**

**int green4 = 2;**

void setup() {

  pinMode(red1, OUTPUT);

  pinMode(yellow1, OUTPUT);

  pinMode(green1, OUTPUT);

  pinMode(red2, OUTPUT);

  pinMode(yellow2, OUTPUT);

  pinMode(green2, OUTPUT);

  pinMode(red3, OUTPUT);

  pinMode(yellow3, OUTPUT);

  pinMode(green3, OUTPUT);

  pinMode(red4, OUTPUT);

  pinMode(yellow4, OUTPUT);

  pinMode(green4, OUTPUT);

}

void loop() {

  //North-South

  digitalWrite(green1, HIGH);

  digitalWrite(red2, HIGH);

  digitalWrite(yellow3, HIGH);

  digitalWrite(yellow4, HIGH);

  delay(5000);

  digitalWrite(green1, LOW);

  digitalWrite(yellow1, HIGH);

  delay(2000);

  digitalWrite(yellow1, LOW);

  digitalWrite(red1, HIGH);

  digitalWrite(green2, HIGH);

  delay(5000);

  digitalWrite(red2, LOW);

  digitalWrite(yellow4, LOW);

  digitalWrite(yellow2, HIGH);

  delay(2000);

  digitalWrite(yellow2, LOW);

  digitalWrite(red2, HIGH);

  digitalWrite(red3, HIGH);

  digitalWrite(green3, HIGH);

  delay(5000);

  digitalWrite(green3, LOW);

  digitalWrite(yellow3, HIGH);

  delay(2000);

  digitalWrite(yellow3, LOW);

  digitalWrite(red3, HIGH);

  digitalWrite(green4, HIGH);

  digitalWrite(red4, LOW);

  delay(5000);

  digitalWrite(green4, LOW);

  digitalWrite(yellow4, HIGH);

  delay(2000);

  digitalWrite(yellow4, LOW);

  digitalWrite(red4, HIGH);

### 3. Lab Assignment 3: Arduino Temp. Sensor (DHT11):

#include<dht.h>  // Include library

#define outPin 8  // Defines pin number to which the sensor is connected

dht DHT;      // Creates a DHT object

void setup() {

  Serial.begin(9600);

}

void loop() {

  int readData = DHT.read11(outPin);

  float t = DHT.temperature;  // Read temperature

  float h = DHT.humidity;   // Read humidity

  Serial.print("Temperature = ");

Serial.print(t);

  Serial.print("°C | ");

  Serial0.0print0((0t\*09.00)0/05.00+032.00)0;0 // Convert celsius to fahrenheit

  Serial.println("°F ");

Serial.print("Humidity = ");

  Serial.print(h);

  Serial.println("% ");

  Serial.println("");

  delay(2000); // wait two seconds

}

### 3.1. Lab Assignment 3: Arduino Temp. Sensor (LM35):

int val;

int tempPin = 1;

int ledPin = 7;                // LED connected to digital pin 7

void setup()

{

  pinMode(ledPin, OUTPUT);      // sets the digital pin as output

  Serial.begin(9600);

}

void loop()

{

val = analogRead(tempPin);

//Serial.print("TEMPRATURE = ");

Serial.print(val);

//Serial.print("\*C");

delay(1000);

Serial.println();

}

/\* uncomment this to get temperature in farenhite

Serial.print("TEMPRATURE = ");

Serial.print(farh);

Serial.print("\*F");

Serial.println();  
\*/

### 4. Lab Assignment 4: Arduino Servo Motor:

#include <Servo.h>

int pos = 0;

Servo servo\_9;

void setup()

{

servo\_9.attach(9);

}

void loop()

{

// sweep the servo from 0 to 180 degrees in steps

// of 1 degrees

for (pos = 0; pos <= 180; pos += 1) {

// tell servo to go to position in variable 'pos'

servo\_9.write(pos);

// wait 15 ms for servo to reach the position

delay(30); // Wait for 15 millisecond(s)

}

for (pos = 180; pos >= 0; pos -= 1) {

// tell servo to go to position in variable 'pos'

servo\_9.write(pos);

// wait 15 ms for servo to reach the position

delay(30); // Wait for 15 millisecond(s)

}

**}**

### 5. Lab Assignment 5: Arduino Ultrasonic Sensor:

int trig =6;

int echo =5;

int led1 = 11;

int buzzpin = 10;

long dur ;

int dis\_mm ;

float dis\_cm;

void setup()

{

pinMode(trig, OUTPUT); // sets the digital pin as output

pinMode(echo, INPUT); // sets the digital pin as output

pinMode(led1, OUTPUT); // sets the digital pin as output

pinMode(buzzpin, OUTPUT);

Serial.begin(9600);

}

void loop()

{

digitalWrite(trig, LOW); // sets the LED on

delayMicroseconds(2); // waits for a second

digitalWrite(trig, HIGH); // sets the LED off

delayMicroseconds(10);

digitalWrite(trig, LOW); // sets the LED on

dur = pulseIn(echo, HIGH);

dis\_mm = dur\*0.344/2 ;

dis\_cm = dis\_mm/10;

if (dis\_cm <50)

{

digitalWrite(led1, HIGH);

digitalWrite(buzzpin, HIGH);

}

els

{

digitalWrite (led1,LOW);

digitalWrite(buzzpin, LOW);

}

Serial.print("distance: ");

Serial.print(dis\_cm);

Serial.println(" cm");

delay(500);

}

### 6. Lab Assignment 6: Raspberry Pi DHT22:

import Adafruit\_DHT

import time

DHT\_SENSOR = Adafruit\_DHT.DHT22

DHT\_PIN =23

while True:

humidity, temperature = Adafruit\_DHT.read(DHT\_SENSOR, DHT\_PIN)

if humidity is not None and temperature is not None:

print("Temp={0:0.1f}C Humidity={1:0.1f}%".format(temperature, humidity))

else:

print("Sensor failure. Check wiring.");

time.sleep(3);

### 6.1 Lab Assignment 6: Raspberry Pi Ultrasonic Sensor:

import RPi.GPIO as GPIO

import time

try:

GPIO.setmode(GPIO.BOARD)

PIN\_TRIGGER = 7

PIN\_ECHO = 11

GPIO.setup(PIN\_TRIGGER, GPIO.OUT)

GPIO.setup(PIN\_ECHO, GPIO.IN)

GPIO.output(PIN\_TRIGGER, GPIO.LOW)

print "Waiting for sensor to settle"

time.sleep(2)

print "Calculating distance"

GPIO.output(PIN\_TRIGGER, GPIO.HIGH)

time.sleep(0.00001)

GPIO.output(PIN\_TRIGGER, GPIO.LOW)

### 7. Lab Assignment 7: Thingspeak:

**Steps for Creating Cloud Account**

Goto https://thingspeak.com/

Create your login

Select Tab Channel

Select Create new Channel

Click on API key and Copy Write API Key

Steps on Client side (RPi)

Include Write API key in the code in front of **myAPI**

Execute the code on Client

Click – Private option in My Channel

Observe output – temperature and humidity in the graphical format.

My Channel – ADD Widgets – select any appropriate option

--------------

Go to APP- MATLAB Analysis and select NEW

In Templates select Custom (no starter code)

In Examples Sample code to analyze and transform data, Select Calculate and display average humidity

Click on CREATE

It will display the MATLAB code

Click on SAVE AND RUN

Go to APP- MATLAB Visualization and select NEW and follow the same steps

-------------------------

# Temperature and humidity data logging to cloud system by BNJagdale

#! /usr/bin/env python

import sys

import urllib3

import random

import time

import Adafruit\_DHT as dht

# Enter Your API key here

#myAPI= 'YTO65N2C4JEDKXMI'

myAPI= 'PU9CNPOBWV7HHNZ4'

# URL where we will send the data, Don't change it

baseURL = 'https://api.thingspeak.com/update?api\_key=%s' % myAPI

def DHT22\_data():

# Dummey data of temperature and humidity

humi, temp = dht.read\_retry(dht.DHT22,23)

return humi, temp

print("Exit data logging with control+c")

while True:

try:

humi,temp = DHT22\_data()

print('Temp={0:0.1f}\*C Humidity={1:0.1f}%'.format(temp,humi))

#Sending the data to thingspeak

con = urllib3.PoolManager()

response = con.request('GET', baseURL + '&field1=%s&field2=%s' % (temp, humi))

print(response.status)

print("Sample transfer OK")

# DHT22 requires 5 seconds to give a reading

time.sleep(5)

except KeyboardInterrupt:

exit()

### 7.1 Lab Assignment 7: UbiDots:

#Python Program to read temperature sample from Pi/dht22 and send it to ubiDots cloud dashboard

import time

import requests

import math

import random

import Adafruit\_DHT as dht

#TOKEN = "BBTR-CajzYXTtXV9iG8vXTJrWXsQDgbLAyi"

TOKEN = "BBUS-mNHbITwMItBA4202n2NnumjAdpjiAa"

DEVICE\_LABEL = "mydevice1" # Put your device label here

VARIABLE\_LABEL\_1 = "temperature" # Put your first variable label here

def build\_payload(variable\_1):

# REading sensor data

humi, temp = dht.read\_retry(dht.DHT22,14)

payload = {variable\_1: temp}

return payload

def post\_request(payload):

# Creates the headers for the HTTP requests

url = "http://industrial.api.ubidots.com"

url = "{}/api/v1.6/devices/{}".format(url, DEVICE\_LABEL)

headers = {"X-Auth-Token": TOKEN, "Content-Type": "application/json"}

# Makes the HTTP requests

status = 400

attempts = 0

while status >= 400 and attempts <= 5:

req = requests.post(url=url, headers=headers, json=payload)

status = req.status\_code

attempts += 1

time.sleep(4)

# Processes results

print(req.status\_code, req.json())

if status >= 400:

print("[ERROR] Could not send data after 5 attempts, please check \

your token credentials and internet connection")

return False

print("[INFO] request made properly, Samples uploaded successfully")

return True

def main():

payload = build\_payload(VARIABLE\_LABEL\_1)

print("[INFO] Attemping to send data")

post\_request(payload)

print("[INFO] finished")

if \_\_name\_\_ == '\_\_main\_\_':

while (True):

main()

time.sleep(2)