Internet of Things

Lab 1

Lesson 1 : Led Blink

Code

/\*

\* Author: Elecrow engle

\* Date:7/12/2017

\* IDE V1.8.2

\* Email:engle@elecrow.com

\* Function:

\*/

// Pin 13 has an LED cconnected on most Arduino boards.

// give it a name:

const int led = 13;

// the setup routine runs once when you press reset:

void setup() {

// initialize the digital pin as an output.

pinMode(led, OUTPUT);

}

// the loop routine runs over and over again forever:

void loop() {

digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)

delay(1000); // wait for a second

digitalWrite(led, LOW); // turn the LED off by making the voltage LOW

delay(1000); // wait for a second

}

Lesson 5 : Fade

Code

/\*

Fade

This example shows how to fade an LED on pin 9 using the analogWrite()

function.

The analogWrite() function uses PWM, so if you want to change the pin you're

using, be sure to use another PWM capable pin. On most Arduino, the PWM pins

are identified with a "~" sign, like ~3, ~5, ~6, ~9, ~10 and ~11.

This example code is in the public domain.

http://www.arduino.cc/en/Tutorial/Fade

\*/

int led = 11; // the PWM pin the LED is attached to

int brightness = 0; // how bright the LED is

int fadeAmount = 5; // how many points to fade the LED by

// the setup routine runs once when you press reset:

void setup() {

// declare pin 9 to be an output:

pinMode(led, OUTPUT);

}

// the loop routine runs over and over again forever:

void loop() {

// set the brightness of pin 9:

analogWrite(led, brightness);

// change the brightness for next time through the loop:

brightness = brightness + fadeAmount;

// reverse the direction of the fading at the ends of the fade:

if (brightness <= 0 || brightness >= 255) {

fadeAmount = -fadeAmount;

}

// wait for 30 milliseconds to see the dimming effect

delay(30);

}

Lesson 10 : RGB Led

Code

/\*

\* Author: Elecrow Engle

\* Date:7/12/2017

\* IDE V1.8.2

\* Email:engle@elecrow.com

\* Function:

\*/

const int redPin = 11;

const int greenPin = 10;

const int bluePin = 9;

//uncomment this line if using a Common Anode LED

//#define COMMON\_ANODE

void setup(){

pinMode(redPin, OUTPUT);

pinMode(greenPin, OUTPUT);

pinMode(bluePin, OUTPUT);

}

void setColor(int red, int green, int blue){

analogWrite(redPin, red);

analogWrite(greenPin, green);

analogWrite(bluePin, blue);

}

void loop(){

setColor(255, 0, 0); // red

delay(1000);

setColor(0, 255, 0); // green

delay(1000);

setColor(0, 0, 255); // blue

delay(1000);

setColor(255, 255, 0); // yellow

delay(1000);

setColor(80, 0, 80); // purple

delay(1000);

setColor(0, 255, 255); // aqua

delay(1000);

}

Lesson 4 : Analog Input

Code

/\*

Analog Input

Demonstrates analog input by reading an analog sensor on analog pin 0 and

turning on and off a light emitting diode(LED) connected to digital pin 13.

The amount of time the LED will be on and off depends on the value obtained

by analogRead().

The circuit:

- potentiometer

center pin of the potentiometer to the analog input 0

one side pin (either one) to ground

the other side pin to +5V

- LED

anode (long leg) attached to digital output 13

cathode (short leg) attached to ground

- Note: because most Arduinos have a built-in LED attached to pin 13 on the

board, the LED is optional.

created by David Cuartielles

modified 30 Aug 2011

By Tom Igoe

This example code is in the public domain.

http://www.arduino.cc/en/Tutorial/AnalogInput

\*/

int sensorPin = A0; // select the input pin for the potentiometer

int ledPin = 13; // select the pin for the LED

int sensorValue = 0; // variable to store the value coming from the sensor

void setup() {

// declare the ledPin as an OUTPUT:

pinMode(ledPin, OUTPUT);

}

void loop() {

// read the value from the sensor:

sensorValue = analogRead(sensorPin);

// turn the ledPin on

digitalWrite(ledPin, HIGH);

// stop the program for <sensorValue> milliseconds:

delay(sensorValue);

// turn the ledPin off:

digitalWrite(ledPin, LOW);

// stop the program for for <sensorValue> milliseconds:

delay(sensorValue);

}

Lesson 6 : Button

Code

/\*

\* Author: Elecrow Engle

\* Date:7/12/2017

\* IDE V1.8.2

\* Email:engle@elecrow.com

\* Function:

\*/

// constants won't change. They're used here to

// set pin numbers:

const int buttonPin = 2; // the number of the pushbutton pin

const int ledPin = 13; // the number of the LED pin

// variables will change:

int buttonState = 0; // variable for reading the pushbutton status

void setup() {

// initialize the LED pin as an output:

pinMode(ledPin, OUTPUT);

// initialize the pushbutton pin as an input:

pinMode(buttonPin, INPUT);

}

void loop() {

// read the state of the pushbutton value:

buttonState = digitalRead(buttonPin);

// check if the pushbutton is pressed.

// if it is, the buttonState is HIGH:

if (buttonState == HIGH) {

// turn LED on:

digitalWrite(ledPin, HIGH);

} else {

// turn LED off:

digitalWrite(ledPin, LOW);

}

}

Lesson 9 : buzzer

Code

/\*

\* Author: Elecrow Engle

\* Date:7/12/2017

\* IDE V1.8.2

\* Email:engle@elecrow.com

\* Function:

\*/

#define buzzer 5

void setup(){

//Null

}

void loop() {

// generates a 400Hz tone in output pin 8 with 2000ms of duration

tone(buzzer, 400, 2000);

}