307 portfolio

lab 1

activity 1.1

For this activity you need to modify the Blink demo application to:

1: Make the LED flash every 1/2 second

2: Make the LED flash every 2 seconds

BONUS: Modify the code so the blink rate is controlled by a global variable

For this task the first thing that was completed was creating the global variable to make the other tasks easier. This was done by adding the following to the top of the code:

int time = 1000;

Then the delays were changed so instead of calling an integer they called the time variable like so:

delay(time)

As time is a global variable it is the only thing that needs to be changed to change the flash rate. The time variable is done in milliseconds, so 1000 is one second.

To complete the first task, making the LED flash every half second, the time global variable was changed from 1000 to 500 giving the following code:

// create the time variable

int time = 500;

// the setup function runs once when you press reset or power the board

void setup() {

// initialize digital pin 13 as an output.

pinMode(13, OUTPUT);

}

// the loop function runs over and over again forever

void loop() {

digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)

delay(time); // wait for a second

digitalWrite(13, LOW); // turn the LED off by making the voltage LOW

delay(time); // wait for a second

}

For the second task to make it flash once every 2 seconds the time variable just needed to be changed to 2000, like so:

// create the time variable

int time = 2000;

// the setup function runs once when you press reset or power the board

void setup() {

// initialize digital pin 13 as an output.

pinMode(13, OUTPUT);

}

// the loop function runs over and over again forever

void loop() {

digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)

delay(time); // wait for a second

digitalWrite(13, LOW); // turn the LED off by making the voltage LOW

delay(time); // wait for a second

}

Activity 1.2

For this activity modify the Blink demo application to use an external LED (Connected on **pin 10**) instead of the on-board LED

For this task the code needed to be changed to use an external LED on pin 10. This is done by changing the pinMode from 13 to 10 shown in the following code:

// create the time variable

int time = 1000;

// the setup function runs once when you press reset or power the board

void setup() {

// initialize digital pin 10 as an output.

pinMode(10, OUTPUT);

}

// the loop function runs over and over again forever

void loop() {

digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)

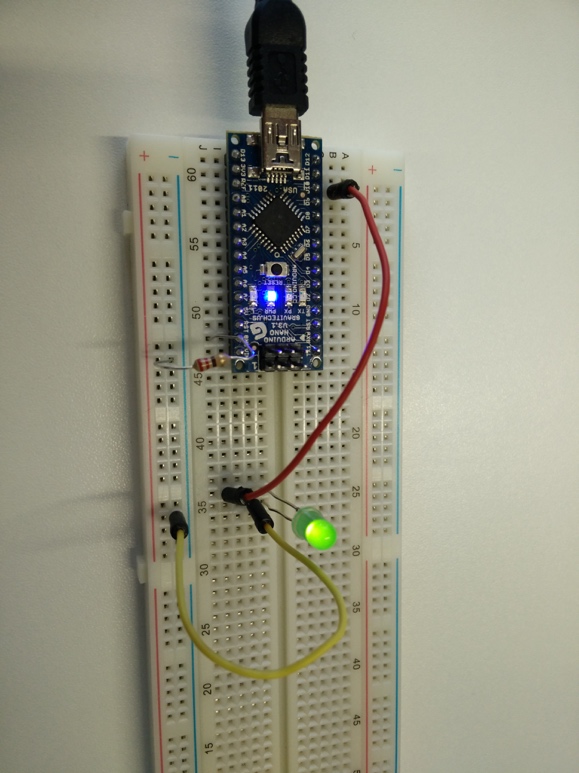
delay(time); // wait for a second

digitalWrite(13, LOW); // turn the LED off by making the voltage LOW

delay(time); // wait for a second

}

With this complete the circuitry needed to be created to accommodate this. This circuitry is shown in the following picture:



Activity 1.3

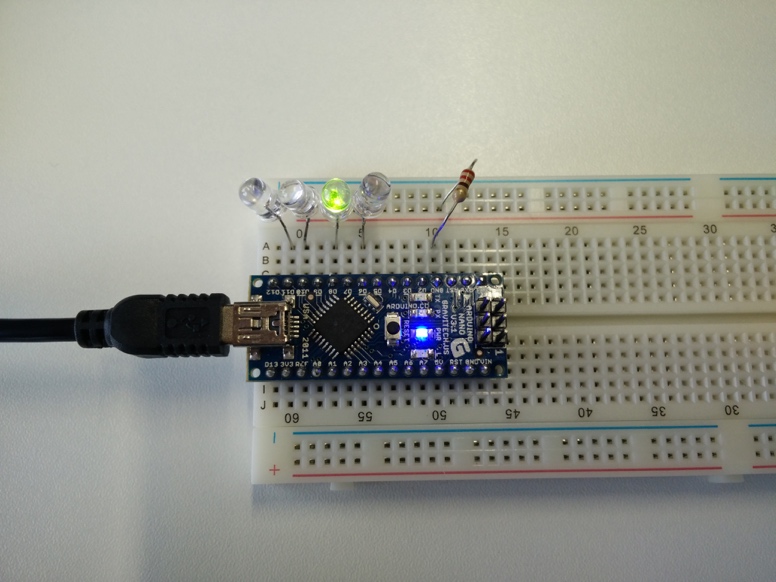
For this activity you are going to make your own Knight Rider light bar (camp voice of kit to be provided by Mike):

1: Connect 4 LED’s to the Arduino (one digital output each)

2: Develop Code to make the LED’s flash in sequence (IE LED1, then LED2, then LED3, etc..) once the sequence has ended start again with LED1

BONUS: Develop code to make the LED’S reverse direction at the end of a sequence

The first thing that needed to be completed was creating a new circuit to accommodate the 4 LEDs and have them all connected to separate pins. This was done like so:



The code to make the lights light up in the correct order is as follows:

// create the time variable

int time = 100;

int myPins[] = {6, 8, 10, 11};

// the setup function runs once when you press reset or power the board

void setup() {

// initialize digital pins as outputs.

for(int i = 0; i < sizeof(myPins); i++) {

pinMode(myPins[i], OUTPUT);

}

}

// the loop function runs over and over again forever

void loop() {

for(int i = 0; i < sizeof(myPins); i++) {

digitalWrite(myPins[i-1], LOW); // turn the LED off by making the voltage LOW

digitalWrite(myPins[i], HIGH); // turn the LED on (HIGH is the voltage level)

delay(time); // wait

}

for(int i = sizeof(myPins); i >= 0; i--) {

digitalWrite(myPins[i+1], LOW); // turn the LED off by making the voltage LOW

digitalWrite(myPins[i], HIGH); // turn the LED on (HIGH is the voltage level)

delay(time); // wait

}

}

This code also makes the LEDs reverse direction at the end.

Lab 2

2.2

void setup() {

Serial.begin(9600);

}

void loop() {

int sensorValue = analogRead(A0);

Serial.println(sensorValue);

delay(2000);

}

2.3

int buttonPin = 13;

int buttonState = 0;

long lastDebounceTime = 0;

long debounceDelay = 50;

int lastButtonState = LOW;

int buttonData;

void setup() {

Serial.begin(9600);

pinMode(buttonPin, INPUT);

}

void loop() {

int sensorValue = analogRead(A0);

buttonState = digitalRead(buttonPin);

if (buttonState != lastButtonState) {

lastDebounceTime = millis();

}

if ((millis() - lastDebounceTime) > debounceDelay) {

if (buttonState != buttonData) {

buttonData = buttonState;

if (buttonState == HIGH) {

Serial.println(sensorValue);

}

}

}

lastButtonState = buttonState;

}

lab 3

3.2

int buttonPin = 13;

int buttonState = 0;

long lastDebounceTime = 0;

long debounceDelay = 50;

int lastButtonState = LOW;

int buttonData;

void setup() {

Serial.begin(9600);

pinMode(buttonPin, INPUT);

}

void loop() {

int sensor1Value = analogRead(A0);

int sensor2Value = analogRead(A1);

buttonState = digitalRead(buttonPin);

float voltage = sensor1Value \* 5.0;

voltage = voltage / 1024.0;

float temperature = (voltage - 0.5) \* 100;

if (buttonState != lastButtonState) {

lastDebounceTime = millis();

}

if ((millis() - lastDebounceTime) > debounceDelay) {

if (buttonState != buttonData) {

buttonData = buttonState;

if (buttonState == HIGH) {

String outString = "Good ";

if (sensor2Value < 150) {

outString = outString + "day, the current temperature is ";

} else {

outString = outString + "night, the current temperature is ";

}

outString = outString + temperature;

Serial.println(outString);

}

}

}

lastButtonState = buttonState;

}

3.3

String tsense = "Temperature senor reads: ";

tsense = tsense + sensor1Value;

String lsense = "LDR reads: ";

lsense = lsense + sensor2Value;

Serial.println(tsense);

Serial.println(lsense);

3.4

import serial

ser = serial.Serial("/dev/cu.usbserial-14P00778", 9600)

while True:

print ser.readline()