Elen 123: Mechatronics

Lab 1: Introduction

Matt Russell, Joshua Ramayrat

4/11/17

**Objectives**

-To acquire an understanding of how to work with the Arduino Mega and write programs for digital input/output.

-To become familiar with LED circuits, the pitfalls you can find yourself in when working with them, and the importance of reading the datasheets of the components you are using.

-Learn how to use the Serial Monitor to display program output and debug programs with the Serial.println() function.

**Part 1: Assignment 1**

Code

/\*

Matt Russell, Joshua Ramayrat

Lab 1 Assignment 1

4/11/17

This program is a simple modification of Blink. Instead of blinking the onboard LED a signal is sent

to pin 12 that blinks an LED connected to it.

\*/

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

void setup() {

// initialize digital pin LED\_BUILTIN as an output.

pinMode(12, OUTPUT);

}

// the loop function runs over and over again forever

void loop() {

digitalWrite(12, HIGH); // turn LED on (HIGH=voltage level)

delay(1000); // wait for a second

digitalWrite(12, LOW); // turn the LED off by making the voltage LOW

delay(1000); // wait for a second

}

Observations

The shorter wire on the LED is the cathode (- terminal) and the longer wire is the anode (+ terminal). The placement of the resistor with respect to the diode (i.e, between +5V and the anode or the cathode and GND) has no effect on the behavior of the simple circuit.

**Part 1: Assignment 2**

Code

/\*Lab1 Part 2

Matt Russell, Joshua Ramayrat

\*/

#define DELAY 1000

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

void setup() {

//Initialize 30-37 as outputs

for (int i = 30; i < 38; i++){

pinMode(i,OUTPUT);

}

}

// the loop function runs over and over again forever

void loop() {

for(int j = 30; j < 38; j++){

digitalWrite(j,HIGH);

delay(DELAY);

digitalWrite(j,LOW);

delay(DELAY);

}

for(int k = 30; k < 38; k++){

digitalWrite(k,HIGH);

}

delay(3\*DELAY);

for(int l = 30; l < 38; l++){

digitalWrite(l,LOW);

}

//Pattern 2: Blink from LED 9 to LED 1

for(int i = 37; i > 29; i--){

digitalWrite(i,HIGH);

delay(DELAY);

digitalWrite(i,LOW);

delay(DELAY);

}

/\*Pattern 3: Blink the lights attached to the even pins then

the odd ones\*/

for(int h = 30; h < 38; h=h+2){

digitalWrite(h,HIGH);

delay(DELAY);

digitalWrite(h,LOW);

delay(DELAY);

}

for(int h = 31; h < 39;h=h+2){

digitalWrite(h,HIGH);

delay(DELAY);

digitalWrite(h,LOW);

delay(DELAY);

}

}

Observations

To avoid blowing out LEDs, it is important to always have a resistor in series to prevent destructive levels of current from flowing through the junction. This is especially important when trying to figure out which side of an LED is the anode or cathode as a blown LED won’t turn on no matter the polarity of the voltage across it and you will be left none the wiser for your efforts.

**Part 2**

Code

const int ledPin = LED\_BUILTIN;// the number of the LED pin

int ledState = LOW; // ledState used to set the LED

unsigned long previousMillis = 0; //last time LED stores.

const long interval = 1000; // (ms) blink interval.

void setup() {

pinMode(ledPin, OUTPUT); //setting output pin.

}

void loop() {

// check to see if it's time to blink the LED; that is, if the

// difference between the current time and last time you blinked

// the LED is bigger than the interval at which you want to

// blink the LED.

unsigned long currentMillis = millis();

if (currentMillis - previousMillis >= interval) {

// save the last time you blinked the LED

previousMillis = currentMillis;

// if the LED is off turn it on and vice-versa:

if (ledState == LOW) {

ledState = HIGH;

} else {

ledState = LOW;

}

// set the LED with the ledState of the variable:

digitalWrite(ledPin, ledState);

}

}

Observations

The code provides the same output as the one using delays.

**Part 2: Assignment 3**

Code

/\*Lab1 Part 2

Matt Russell, Joshua Ramayrat

\*/

long interval = 1000

previousMillis = 0

currentMillis = 0;

int ledstate = LOW

j = 30

happen = 0;

void setup() {

Serial.begin(9600);

for (int i = 30; i < 38; i++) {

pinMode(i, OUTPUT); //Initialize 30-37 as outputs

}

}

void loop() {

//pulses all in sequence

Serial.println("Loop start");

while (j < 38) {

currentMillis = millis();

if (currentMillis - previousMillis > interval) {

previousMillis = currentMillis;

if (ledstate == LOW)

ledstate = HIGH;

else

ledstate = LOW;

digitalWrite(j, ledstate);

happen++;

}

if (happen == 2 ) {

j++;

happen = 0;

}

}

for (int k = 30; k < 38; k++) {

digitalWrite(k, HIGH);

}

currentMillis = millis();

previousMillis = currentMillis;

while (currentMillis - previousMillis < 3\*interval) {

currentMillis = millis();

}

for (int l = 30; l < 38; l++) {

digitalWrite(l, LOW);

}

currentMillis = millis();

previousMillis = currentMillis;

while (currentMillis - previousMillis < 3\*interval) {

currentMillis = millis();

}

j = 30;

}

Observations

Although it is much easier to program timing operations with delay() than it is with millis() and while-loops delay() has the unfortunate side effect of halting all processor functions when it is called. This isn’t a problem in programs whose function is something simple, like blinking an LED on and off at regular intervals, but in more complicated programs, like one that guides a robot through an obstacle course, where sections of the code need to be turned off or on while others run it becomes necessary to use millis().

**Part 3: Assignment 4**

Code

/\*Matt Russell, Joshua Ramayrat

Lab 1 Part 3 Assignment 4

4/11/17

Program reads in the settings of the DIP switches to pins 9-12

and then uses the bar graph LED to show the settings

\*/

#define DELAY 500

int pinval, dip[4], dtime;

void setup() {

for(int i = 30; i < 38; i++){

pinMode(i,OUTPUT); //initialize 30-37 output pins.

}

for(int j = 9; j < 13; j++){

pinMode(j,INPUT); //initialize 9-12 input pins.

}

dtime = DELAY;

}

void loop()

{

for(int k = 9; k < 13; k++){

pinval = digitalRead(k); //signal read from pin.

digitalWrite(-k+42, !pinval); //switch-state LED.

dip[k-9] = !pinval; //store LED-state in array.

}

for(int l = 34; l < 38; l++){

digitalWrite(l,dip[37-l]);

}

robdelay(dtime);

for(int l = 34; l < 38; l++){

digitalWrite(l,LOW);

}

robdelay(dtime);

}

//robust delay function that doesn't halt processor operation

void robdelay(int dtime)

{

int prevtime, curtime;

curtime = millis();

prevtime = curtime;

while(curtime - prevtime < dtime){

curtime = millis();

}

}

Observations

Writing a delay function that you can just slap anywhere is very helpful organizationally. The switch state is the opposite of the voltage state (S.HIGH -> 0V, S.LOW -> 5V). Due to how the board was wired a little bit of funky math had to be done in the for-loops to make sure the iterative variable matched up with the correct pin.

For some reason, removing the power from the switch circuit changes all the LED states to high if even one is on but keeps all of them low if they are all closed.

**Part 3: Assignment 5**

Code

/\*Matt Russell, Joshua Ramayrat

Lab 1 Part 3 Assignment 5

4/11/17

Reads in and then outputs the dip-switch states to the serial monitor

\*/

#define DELAY 1000

int pinval, dtime;

void setup() {

//initialize state pins as inputs

for(int i = 9; i < 13; i++){

pinMode(i,INPUT);

}

dtime = DELAY;

Serial.begin(9600);

}

void loop() {

for(int i = 12; i > 8; i--){

pinval = digitalRead(i);

Serial.print(!pinval,DEC);

Serial.print(" ");

}

rodelay(dtime);

Serial.println("");

}

void rodelay(int dtime){

int pt, ct;

pt = millis();

ct = pt;

while (ct - pt < dtime){

ct = millis();

}

}

Observations

The delay function was added so that the Serial Monitor was easier to read as it scrolled by.

**Conclusion**

This lab was mainly an exercise in programming the Arduino. We learned how to use the millis() function to write a custom delay function that does not halt the functioning of the entire processor. Previously foreign components, like the SIP resistor, became known to us and we learned firsthand the importance of connecting an LED to a resistor in series to avoid blowing anything out.

Parts of the lab, like Assignment 5, required us to think about the way we wired our breadboard when we were writing our code.