**Lab\_1**

//First one

#include <ee108.h>

#include <ee108\_constants.h>

#include <ee108\_printf.h>

#include <ee108\_switches.h>

const int The\_Pin = BAR\_LED\_5\_PIN; // the pin to use

const int On\_Time = 1; // the on-time in ms during each blink

const int Off\_Time= 99; // the off-time in ms during each blink

void setup() {

// put your setup code here, to run once:

pinMode(LED\_BUILTIN, OUTPUT);

}

void loop() {

// put your main code here, to run repeatedly:

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

delay(0);

delayMicroseconds(50); // wait for 50 microseconds

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

delay(500); // wait for a half second

}

//Second one

#include <ee108.h>

#include <ee108\_constants.h>

#include <ee108\_printf.h>

#include <ee108\_switches.h>

/\*\*

\* TODO: explanation of this file goes here...

\*/

const unsigned long SUPERLOOP\_MS = 1000;

long gGlobal\_Time = 0;

void setup() {

Serial.begin(9600);

}

void loop() {

static long Static\_Time = 0;

long Local\_Time = 1;

Serial.print("1000 ms: ");

Serial.print("gGlobal\_Time = ");

Serial.print(gGlobal\_Time);

gGlobal\_Time++;

Serial.print(" Static\_Time = ");

Serial.print(Static\_Time);

Static\_Time++;

Serial.print(" Local\_Time = ");

Serial.println(Local\_Time);

Local\_Time++;

Serial.println("hello");

delay(SUPERLOOP\_MS);

}

**Lab\_2**

//Hanlin Cai

//832002117

//Lab2\_1\_NestedIf

/\*\*

\* demonstrate the use of nested if statements.

\* The sketch just lights different LEDs for short and long presses

\* of the button

\*/

#include <ee108.h>

// TODO - define constants for the superloop duration and the number of ticks

// (i.e. superloop iterations) corresponding to a very long click

//

const int Red\_LED = LED1\_PIN;

const int Green\_LED = 2;

const int Button = SW1\_PIN;

int count = 0;

void setup() {

// TODO - initialise the LED pins that you'll be using

pinMode(Red\_LED,OUTPUT);

pinMode(Green\_LED,OUTPUT);

pinMode(Button,INPUT);

}

void loop() {

int sw1Event; // variable to save the current switch event

if(digitalRead(Button)==SW\_ACTIVE){

digitalWrite(Red\_LED, HIGH);

count++;

if(count == 40){

digitalWrite(Green\_LED, HIGH);

}

}else{

count = 0;

digitalWrite(Red\_LED, LOW);

digitalWrite(Green\_LED, LOW);

}

delay(50);

// check the current switch event

sw1Event = readSwitchEvent(sw1Object);

// TODO: check if the switch is currently pressed and light the LEDs

// as per the lab instruction

//

// You can check if the SW1 is currently pressed using \*one\* of the following

// expression options in your if-statement:

// expression 1: if ((sw1Event == SW\_ON\_START) || (sw1Event == SW\_ON\_CONTINUED)) ...

// expression 2: if (sw1Event & SW\_ON\_ANY) ...

// TODO: delay for the appropriate amount of time

}

//Lab2\_2\_WrapTutorial

#include <ee108.h>

const int SUPERLOOP\_MS = 1000;

const int LED\_ON\_MS = 50;

const int NUM\_LEDS = 5;

void setup() {

// initialize 5 bar LEDs

pinMode(BAR\_LED\_1\_PIN, OUTPUT);

pinMode(BAR\_LED\_2\_PIN, OUTPUT);

pinMode(BAR\_LED\_3\_PIN, OUTPUT);

pinMode(BAR\_LED\_4\_PIN, OUTPUT);

pinMode(BAR\_LED\_5\_PIN, OUTPUT);

}

void loop() {

// this variable tracks which Bar LED we want to light

static int ledNum = 0;

// flash the LED on briefly and then switch it off again

digitalWrite(BAR\_LED\_1\_PIN + ledNum, HIGH);

delay(LED\_ON\_MS);

digitalWrite(BAR\_LED\_1\_PIN + ledNum, LOW);

// TODO - increment the counter and wrap if necessary

// ledNum++;

// if(ledNum>=5){

// ledNum=0;

// }

ledNum = (ledNum + 1) % 5;

// pause between LED flashes

delay(SUPERLOOP\_MS);

}

//Lab2\_3\_CascadedIf

#include <ee108.h>

const int RED = LED1\_PIN;

const int GREEN=LED2\_PIN;

static int count=0;

const int delay\_time=1000;

void setup() {

// put your setup code here, to run once:

pinMode(RED, OUTPUT);

pinMode(GREEN, OUTPUT);

}

void loop() {

// put your main code here, to run repeatedly:

if(count==0||count==1||count==8||count==9){

digitalWrite(RED, LOW);

digitalWrite(GREEN, LOW);

}else if(count==2||count==7){

digitalWrite(RED, HIGH);

digitalWrite(GREEN, LOW);

}else if(count==3||count==6){

digitalWrite(RED, LOW);

digitalWrite(GREEN, HIGH);

}else{

digitalWrite(RED, HIGH);

digitalWrite(GREEN, HIGH);

}

count++;

count=count%10;

delay(delay\_time);

}

//OVER

**Lab\_3**

/\*\*

\* Starter sketch for Lab3 - iteration

\*/

//Hanlin Cai

//832002117 in FZU NUMBER

//20122161 in MU NUMBER

//sketch1 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#include <ee108.h>

const int BUTTON\_POLL\_MS = 50;

const int NUM\_REPS = 5;

const int BETWEEN\_REPS\_MS = 500;

const int LED\_ON\_MS = 100;

const int LED\_OFF\_MS = 50;

void setup() {

int ledIndex;

Serial.begin(9600);

Serial.println("============== starting Lab3\_BarLedScan\_Starter ============");

// using a loop, we iterate over all Bar LED pins, enabling each one as an OUTPUT.

// NOTE the idiom BAR\_LED\_1\_PIN + ledIndex to get the pin number for any bar led

// according to it's zero based index, e.g. where the second led has index 1

for (ledIndex = 0; ledIndex < NUM\_BAR\_LEDS; ledIndex++)

pinMode(BAR\_LED\_1\_PIN + ledIndex, OUTPUT);

}

void loop() {

int sw1Event; // used when checking SW1

int sw2Event; // user when checking SW2

Serial.println("\nloop function starting -- click SW1 to begin");

// busy wait at this point until SW1 is clicked (i.e. wait while sw1Event is not a click)

while (readSwitchEvent(sw1Object) != SW\_CLICK)

delay(BUTTON\_POLL\_MS);

// TODO - at this point button SW1 has definitely been clicked

// implement the rest of the lab as specified in the lab doc.

// You will need a loop (a while-loop or for-loop) that repeats NUM\_REPS times

// and in each iteration/repetition you'll have to do some things as specifi ed

// in the lab doc.

for (int ledIndex = 0; ledIndex < NUM\_BAR\_LEDS; ledIndex++)

{

Serial.print("Rep ");

Serial.print(ledIndex+1);

digitalWrite(BAR\_LED\_1\_PIN, HIGH);

delay(LED\_ON\_MS);

digitalWrite(BAR\_LED\_1\_PIN, LOW);

delay(LED\_OFF\_MS);

if(readSwitchEvent(sw2Object) == SW\_ON\_CONTINUED)

{

while (readSwitchEvent(sw2Object) != SW\_OFF\_CONTINUED)

delay(BUTTON\_POLL\_MS);

Serial.println();

ledIndex=ledIndex-1;

continue;

}

Serial.print(", Placeholder – bar LED scan would go here");

Serial.println();

delay(BETWEEN\_REPS\_MS);

}

Serial.println("end of repetitions loop");

}

//sketch2 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#include <ee108.h>

const int BUTTON\_POLL\_MS = 50;

const int BUTTON = A5;

const int NUM\_REPS = 5;

const int BETWEEN\_REPS\_MS = 500;

const int LED\_ON\_MS = 100;

const int LED\_OFF\_MS = 50;

void setup() {

int ledIndex;

Serial.begin(9600);

Serial.println("============== starting Lab3\_BarLedScan\_Starter ============");

pinMode(BUTTON, INPUT);

for (ledIndex = 0; ledIndex < NUM\_BAR\_LEDS; ledIndex++)

pinMode(BAR\_LED\_1\_PIN + ledIndex, OUTPUT);

}

void loop() {

int sw1Event = 0; // used when checking SW1

int sw2Event = 0 ; // user when checking SW2

while (readSwitchEvent(sw1Object) != SW\_CLICK){

delay(BUTTON\_POLL\_MS);

}

sw1Event = 1;

Serial.print("SW1 clicked, starting repetitions loop \n");

for(int i = 0; i < 10 ; i++){

if (readSwitchEvent(sw2Object)& SW\_ON\_ANY){

continue;

}

Serial.print("Rep ");

Serial.print(i+1);

Serial.println(", Placeholder – bar LED scan would go here");

digitalWrite(BAR\_LED\_1\_PIN, HIGH);

delay(LED\_ON\_MS);

digitalWrite(BAR\_LED\_1\_PIN, LOW);

delay(LED\_OFF\_MS);

}

Serial.println("end of repetitions loop");

delay(BETWEEN\_REPS\_MS);

}

//sketch3 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

/\*\*

\* Starter sketch for Lab3 - iteration

\*/

#include <ee108.h>

const int BUTTON\_POLL\_MS = 50;

const int NUM\_REPS = 5;

const int BETWEEN\_REPS\_MS = 500;

const int LED\_ON\_MS = 100;

const int LED\_OFF\_MS = 50;

void setup() {

int ledIndex;

Serial.begin(9600);

Serial.println("============== starting Lab3\_BarLedScan\_Starter ============");

// using a loop, we iterate over all Bar LED pins, enabling each one as an OUTPUT.

// NOTE the idiom BAR\_LED\_1\_PIN + ledIndex to get the pin number for any bar led

// according to it's zero based index, e.g. where the second led has index 1

for (ledIndex = 0; ledIndex < NUM\_BAR\_LEDS; ledIndex++)

pinMode(BAR\_LED\_1\_PIN + ledIndex, OUTPUT);

}

void loop() {

int sw1Event; // used when checking SW1

int sw2Event; // user when checking SW2

Serial.println("\nloop function starting -- click SW1 to begin");

// busy wait at this point until SW1 is clicked (i.e. wait while sw1Event is not a click)

while (readSwitchEvent(sw1Object) != SW\_CLICK)

delay(BUTTON\_POLL\_MS);

// TODO - at this point button SW1 has definitely been clicked

// implement the rest of the lab as specified in the lab doc.

// You will need a loop (a while-loop or for-loop) that repeats NUM\_REPS times

// and in each iteration/repetition you'll have to do some things as specifi ed

// in the lab doc.

for (int ledIndex = 0; ledIndex < NUM\_BAR\_LEDS; ledIndex++)

{

Serial.print("Rep ");

Serial.print(ledIndex+1);

digitalWrite(BAR\_LED\_1\_PIN, HIGH);

delay(LED\_ON\_MS);

digitalWrite(BAR\_LED\_1\_PIN, LOW);

delay(LED\_OFF\_MS);

if(readSwitchEvent(sw2Object) == SW\_ON\_CONTINUED)

{

while (readSwitchEvent(sw2Object) != SW\_OFF\_CONTINUED)

delay(BUTTON\_POLL\_MS);

Serial.println();

ledIndex=ledIndex-1;

continue;

}

for(int i =2;i<=5;i++)

{

digitalWrite(BAR\_LED\_1\_PIN+i-1, HIGH);

delay(LED\_ON\_MS);

digitalWrite(BAR\_LED\_1\_PIN+i-1, LOW);

delay(LED\_OFF\_MS);

}

Serial.println();

delay(BETWEEN\_REPS\_MS);

}

Serial.println("end of repetitions loop");

}

//sketch4 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#include <ee108.h>

const int BUTTON\_POLL\_MS = 50;

const int BUTTON = A5;

const int BUTTON1 = A4;

const int NUM\_REPS = 5;

const int BETWEEN\_REPS\_MS = 500;

const int LED\_ON\_MS = 100;

const int LED\_OFF\_MS = 50;

void setup() {

int ledIndex;

Serial.begin(9600);

Serial.println("============== starting Lab3\_BarLedScan\_Starter ============");

pinMode(BUTTON, INPUT); //Additional questions

pinMode(BUTTON1, INPUT);

for (ledIndex = 0; ledIndex < NUM\_BAR\_LEDS; ledIndex++)

pinMode(BAR\_LED\_1\_PIN + ledIndex, OUTPUT);

}

void loop() {

int state = 0;

int sw1Event = 0; // used when checking SW1

int sw2Event = 0 ; // user when checking SW2

// busy wait at this point until SW1 is clicked (i.e. wait while sw1Event is not a click)

while (readSwitchEvent(sw1Object) != SW\_CLICK){

delay(BUTTON\_POLL\_MS);

}

Serial.print("SW1 clicked, starting repetitions loop \n");

sw1Event = 1;

for(int i = 0; i < 10 ; i++){

Serial.print("Rep ");

Serial.println(i+1);

for(int lednum= 0; lednum < 5 ; lednum++){

if (readSwitchEvent(sw2Object)& SW\_ON\_ANY){

lednum= 0;

continue;

}

if (digitalRead(BUTTON)==SW\_ACTIVE){

state = 1;

break;

}

digitalWrite(BAR\_LED\_1\_PIN + lednum, HIGH);

delay(LED\_ON\_MS);

digitalWrite(BAR\_LED\_1\_PIN + lednum, LOW);

delay(LED\_OFF\_MS);

}

if(state == 1){

Serial.println("release SW1!!");

while(digitalRead(BUTTON)==SW\_ACTIVE){

delay(BUTTON\_POLL\_MS);

}

break;

}

}

Serial.println("end of repetitions loop");

state = 0;

}

//sketch5 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#include <ee108.h>

const int BUTTON\_POLL\_MS = 50;

const int BUTTON = A5;

const int BUTTON1 = A4;

const int PT = A0;

const int NUM\_REPS = 5;

const int BETWEEN\_REPS\_MS = 500;

const int LED\_ON\_MS = 100;

const int LED\_OFF\_MS = 50;

int val = 0;

void setup() {

int ledIndex;

Serial.begin(9600);

Serial.println("============== starting Lab3\_BarLedScan\_Starter ============");

pinMode(BUTTON, INPUT); //Additional questions

pinMode(BUTTON1, INPUT);

pinMode(PT, INPUT); //I used a potentiometer to get a simulated input from 0 to 1200 by potential change, and divide by 300 to get a good number to start with

// using a loop, we iterate over all Bar LED pins, enabling each one as an OUTPUT.

// NOTE the idiom BAR\_LED\_1\_PIN + ledIndex to get the pin number for any bar led

// according to it's zero based index, e.g. where the second led has index 1

for (ledIndex = 0; ledIndex < NUM\_BAR\_LEDS; ledIndex++)

pinMode(BAR\_LED\_1\_PIN + ledIndex, OUTPUT);

}

void loop() {

int state = 0;

int sw1Event = 0; // used when checking SW1

int sw2Event = 0 ; // user when checking SW2

// busy wait at this point until SW1 is clicked (i.e. wait while sw1Event is not a click)

while (readSwitchEvent(sw1Object) != SW\_CLICK){

delay(BUTTON\_POLL\_MS);

}

Serial.print("SW1 clicked, starting repetitions loop \n");

sw1Event = 1;

for(int i = 0; i < 10 ; i++){

Serial.print("Rep ");

Serial.println(i+1);

val = analogRead(A0);

int start = val / 300;

for(int lednum= start; lednum < start+5 ; lednum++){

if (readSwitchEvent(sw2Object)& SW\_ON\_ANY){

lednum= 0;

continue;

}

if (digitalRead(BUTTON)==SW\_ACTIVE){

state = 1;

break;

}

digitalWrite(BAR\_LED\_1\_PIN + lednum, HIGH);

delay(LED\_ON\_MS);

digitalWrite(BAR\_LED\_1\_PIN + lednum, LOW);

delay(LED\_OFF\_MS);

}

if(state == 1){

Serial.println("release SW1!!");

while(digitalRead(BUTTON)==SW\_ACTIVE){

delay(BUTTON\_POLL\_MS);

}

break;

}

}

// TODO - at this point button SW1 has definitely been clicked

// implement the rest of the lab as specified in the lab doc.

// You will need a loop (a while-loop or for-loop) that repeats NUM\_REPS times

// and in each iteration/repetition you'll have to do some things as specified

// in the lab doc.

Serial.println("end of repetitions loop");

state = 0;

}

**Lab\_4**

//Lab4-832002117-HanlinCai-20122161

//Lab4-832002117-HanlinCai-20122161

int absoluteDiiference();

void setup() {

// put your setup code here, to run once:

Serial.begin(9600);

}

void loop() {

// put your main code here, to run repeatedly:

int p=random(100);

Serial.print("p is ");

Serial.print(p);

int q=random(100);

Serial.print(",q is ");

Serial.print(q);

int r=random(100);

Serial.print(",r is ");

Serial.print(r);

int s=random(100);

Serial.print(",s is ");

Serial.println(s);

int result=absoluteDiiference(q,r);

Serial.print("abs diff q and r is ");

Serial.println(result);

if(absoluteDiiference(q,r)>q){

Serial.println("abs diff p and r > q is true");

}else{

Serial.println("abs diff p and r > q is false");

}

Serial.print("abs diff of abs diff p and q and abs diff r and s is ");

Serial.println(absoluteDiiference(absoluteDiiference(p,q),absoluteDiiference(r,s)));

Serial.println();

}

int absoluteDiiference(int q,int r){

int result=abs(q-r);

return result;

}

#include <ee108.h>

void setupBarLeds();

//This initializes all output pins

void blinkLed(int pin,int times,int ontime,int offtime);

//can blink any BarLED number you specify some specified number of times and with specified blink on/off durations

// \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

// constants

const int SHORT\_ON\_MS = 100;

const int SHORT\_OFF\_MS = 500;

const int MED\_ON\_MS = 500;

const int MED\_OFF\_MS = 700;

const int LONG\_ON\_MS = 1200;

const int LONG\_OFF\_MS = 600;

// ===================================================================

// Top level functions

// ===================================================================

/\*\*

\* set up the sketch to use LEDs and serial

\*/

void setup() {

setupBarLeds();

// serial output

Serial.begin(9600);

Serial.println("====== Lab4\_BlinkNoFunctions starting ========");

}

/\*\*

\* Blink the LEDs in accordance with the lab requirements each

\* time the loop function is called

\*/

void loop() {

static int loopNum = 0;

Serial.print("Loop ");

Serial.println(loopNum);

// BarLED1

Serial.println(" Blink BarLED1 on/off long pattern x 1");

blinkLed(1,1,LONG\_ON\_MS,LONG\_OFF\_MS);

// BarLED4

Serial.println(" Blink BarLED4 on/off short pattern x 2 times");

blinkLed(4,2,MED\_ON\_MS,MED\_OFF\_MS);

// BarLED8

Serial.println(" Blink BarLED8 on/off short pattern x 3 times");

blinkLed(8,3,SHORT\_ON\_MS,SHORT\_OFF\_MS);

loopNum++;

// there are plenty of delays in the blinking code so we don't

// need to add any extra delay at the end of the loop function

}

void blinkLed(int pin,int times,int ontime,int offtime){

for (int i=0; i < times; i++) {

digitalWrite(BAR\_LED\_1\_PIN+pin-1, HIGH);

delay(ontime);

digitalWrite(BAR\_LED\_1\_PIN+pin-1, LOW);

delay(offtime);

}

}

void setupBarLeds(){

int ledIndex;

// set up the LEDs

pinMode(LED1\_PIN, OUTPUT);

pinMode(LED2\_PIN, OUTPUT);

// set up the Bar LEDs

for (ledIndex = 0; ledIndex < NUM\_BAR\_LEDS; ledIndex++)

pinMode(BAR\_LED\_1\_PIN + ledIndex, OUTPUT);

}

**Lab\_5**

/\*

Lab05

Hanlin Cai

20122161

832002117

\*/

#include <ee108\_switches.h>

#include <ee108\_printf.h>

#include <ee108\_constants.h>

#include <ee108.h>

/\*\*

\* This is possibly the simplest demonstration of

\* click detection using the ee108 library

\*/

#include <ee108.h>

#define arrSize 8

const int SUPERLOOP\_MS = 20;

// ==============================================================

// top level functions

// ==============================================================

void setup() {

// serial output

Serial.begin(9600);

Serial.println("====== Lab5\_ClickDetection starting ========");

}

void loop() {

static int arr[arrSize] = {0};

int sw1Event;

// Input

sw1Event = readSwitchEvent(sw1Object);

// processing and output

if (sw1Event == SW\_CLICK) {

Serial.print("[Click] array is {");

// output 1

for(int i=0;i<arrSize;i++){

Serial.print(arr[i]);

if(i==arrSize-1){

}else{

Serial.print(",");

}

}

Serial.print("}");

Serial.println();

} else if (sw1Event == SW\_CLICK\_LONG) {

// output 2

Serial.print("[LongClick] modified array {");

for(int i=0;i<arrSize;i++){

arr[i]=random(100);

}

for(int i=0;i<arrSize;i++){

Serial.print(arr[i]);

if(i==arrSize-1){

}else{

Serial.print(",");

}

}

Serial.print("}");

Serial.println();

delay(SUPERLOOP\_MS);

}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#include <ee108\_switches.h>

#include <ee108\_printf.h>

#include <ee108\_constants.h>

#include <ee108.h>

/\*\*

\* This is possibly the simplest demonstration of

\* click detection using the ee108 library

\*/

#include <ee108.h>

#define arrSize 8

const int SUPERLOOP\_MS = 20;

// ==============================================================

// top level functions

// ==============================================================

void setup() {

// serial output

Serial.begin(9600);

Serial.println("====== Lab5\_ClickDetection starting ========");

}

void loop() {

static int arr[arrSize] = {0};

int sw1Event;

// Input

sw1Event = readSwitchEvent(sw1Object);

// processing and output

if (sw1Event == SW\_CLICK) {

Serial.print("[Click] array is {");

printArray(arr,arrSize);

}

else if (sw1Event == SW\_CLICK\_LONG) {

Serial.print("[LongClick] modified array {");

// output 2

randomizeArrayValus(arr,arrSize);

printArray(arr,arrSize);

delay(SUPERLOOP\_MS);

}

}

void printArray(int arr[],int len){

// output 1

for(int i=0;i<len;i++){

Serial.print(arr[i]);

if(i==len-1){

}else{

Serial.print(",");

}

}

Serial.print("}");

Serial.println();

}

//function

//function

void randomizeArrayValus(int arr[],int len){

int p;

for(p=0;p<len;p++){

arr[p]=random(100);

}

}

**Lab\_6**

//Lab6 Hanlin Cai

#include <ee108.h>

const int SUPERLOOP\_MS = 20;

//

// function declarations

//

int\* rectangle(int length, int width); // TODO - will need to be modified

// ==============================================================

// top level functions

// ==============================================================

void setup() {

Serial.begin(9600);

Serial.println("========= Lab6\_RectOutputParams starting ===========");

Serial.println("Press SW1 to start...");

}

void loop() {

int sw1Event;

int length;

int width;

sw1Event = readSwitchEvent(sw1Object);

if (sw1Event == SW\_CLICK) {

// TODO: generate random values for length and width

length = rand() % 99;

width = rand() % 99;

// TODO: you will need to modify this call to the rectangle function

// to take account of your output parameters

int\* ans = new int[2];

ans = rectangle(length, width);

// TODO: you will need to modify this printing code to include output values

Serial.print("Rectange: length=");

Serial.print(length);

Serial.print(", width=");

Serial.print(width);

Serial.print(" => area ");

Serial.print(\*ans);

Serial.print(", perimeter ");

Serial.print(\*(ans + 1));

Serial.println(""); // newline on its own

Serial.println("\nPress SW1 to do it again...");

}

delay(SUPERLOOP\_MS);

}

// ==============================================================

// helper functions

// ==============================================================

// TODO - this function will need to be modified

int\* rectangle(int length, int width) {

// TODO - you need to add code here

int\* ret = new int[2];

ret[0] = length + width << 1;

ret[1] = length \* width;

return ret;

}

#include <ee108.h>

#include <limits.h>

const int SAMPLE\_MS = 250;

const int ANALOG\_PIN = LDR\_PIN;

const int NUM\_SAMPLES = 8;

const int FIFO\_LEN = 4;

//

// function declarations

//

int sampleAnalogIn(int pin, int\* val, int\* fifo); // TODO - will need to be modified

// ==============================================================

// top level functions

// ==============================================================

void setup() {

Serial.begin(9600);

Serial.println("========= Lab6\_OptionalInOut starting ===========");

}

void loop() {

int i;

int fifo[4];

int val[2];

fifo[0] = fifo[1] = fifo[2] = fifo[3] = 0;

val[0] = 0x3f3f3f3f;

val[1] = 0;

Serial.println("-----------");

Serial.println("sample without the optional params...");

for (i = 0; i < NUM\_SAMPLES; i++) {

int value = sampleAnalogIn(ANALOG\_PIN, val, fifo); // TODO - will need to be modified

Serial.print("Sample ");

Serial.print(i+1);

Serial.print(", value=");

Serial.print(value);

Serial.print(", min=");

Serial.print(val[0]);

Serial.print(", max=");

Serial.print(val[1]);

Serial.print(", Fifo = {");

int avg = 0;

for (int i = 0; i < 4; i++) {

if (i != 0) {

Serial.print(", ");

}

Serial.print(fifo[i]);

avg += fifo[i];

}

Serial.print("}, average = ");

Serial.println(avg / 4);

delay(SAMPLE\_MS);

}

// TODO - at the correct point int the lab (the description will tell you when)

// uncomment the following block and edit as necessary to handle udpating the

// minVal and maxVal variables

// Serial.println("sample using the optional params...");

// for (i = 0; i < NUM\_SAMPLES; i++) {

// int value = sampleAnalogIn(ANALOG\_PIN); // TODO - will need to be modified

//

// TODO printing and FIFO

//

// delay(SAMPLE\_MS);

// }

delay(8000);

}

// ==============================================================

// helper functions

// ==============================================================

// TODO - this function will need to be modified so that it can update

// the minVal or maxVal seen to date

int sampleAnalogIn(int pin, int\* val, int\* fifo) {

int value = analogRead(pin);

// TODO - extra code to go here

val[0] = min(val[0], value);

val[1] = max(val[0], value);

for (int i = 4; i >= 1; i--) {

fifo[i] = fifo[i - 1];

}

fifo[0] = value;

return value;

}

OVER