#define TRIGGER\_PIN\_1 13

#define ECHO\_PIN\_1 12

#define TRIGGER\_PIN\_2 11

#define ECHO\_PIN\_2 10

#define BUZZER\_PIN 9

#define TONE\_PIN 8

#define USONIC\_DIV 58.0

#define MEASURE\_SAMPLE\_DELAY 5

#define MEASURE\_SAMPLES 25

#define MEASURE\_DELAY 250

#define ALARM\_TARGET\_1 10.0 // Target Distance

#define ALARM\_TARGET\_2 10.0 // Target Distance

#define ALARM\_DEADBAND 1.0 // Deadband

#define ALARM\_COUNT\_SETPOINT 4 // Number of loop to start alarming at

#define ALARM\_COUNT\_RESET 8 // Maximum Loops (counts) before reset

#define ALARM\_ON\_HIGH\_DISTANCE 0 // ALARM ON HIGH (1) / LOW (0)

int ALARM\_COUNT\_1 = 0; // Current Loop Count

int ALARM\_COUNT\_2 = 0; // Current Loop Count

int CURRENT\_ALARM\_COUNT = 0;

float CURRENT\_ALARM\_TARGET = 10.0; // // Target Distance

// int mode = 1;

// int mode = 2;

int Current\_Sensor = 2;

int debug = 1;

int TRIGGER\_PIN = 0;

int ECHO\_PIN = 0;

float DISTANCE\_INCHES = 0.0;

void setup()

{

// Serial monitoring

Serial.begin(9600);

print\_config();

// Initializing Trigger and Buzzer Output and Echo Input

pinMode(TRIGGER\_PIN\_1, OUTPUT);

pinMode(ECHO\_PIN\_1, INPUT);

pinMode(BUZZER\_PIN, OUTPUT);

pinMode(TONE\_PIN,INPUT);

// Reset the trigger pin and wait a half a second

digitalWrite(TRIGGER\_PIN\_1, LOW);

delayMicroseconds(500);

// Initializing Trigger Output and Echo Input

pinMode(TRIGGER\_PIN\_2, OUTPUT);

pinMode(ECHO\_PIN\_2, INPUT);

// Reset the trigger pin and wait a half a second

digitalWrite(TRIGGER\_PIN\_2, LOW);

delayMicroseconds(500);

// Turn Alarm Buzzer output off

digitalWrite(BUZZER\_PIN, LOW);

delayMicroseconds(500);

}

void loop()

{

if (Current\_Sensor== 1)

{

Current\_Sensor = 2;

TRIGGER\_PIN = TRIGGER\_PIN\_2;

ECHO\_PIN = ECHO\_PIN\_2;

CURRENT\_ALARM\_COUNT = ALARM\_COUNT\_2 +=1;

CURRENT\_ALARM\_TARGET = ALARM\_TARGET\_2;

}

else if (Current\_Sensor== 2)

{

Current\_Sensor= 1;

TRIGGER\_PIN = TRIGGER\_PIN\_1;

ECHO\_PIN = ECHO\_PIN\_1;

CURRENT\_ALARM\_COUNT = ALARM\_COUNT\_1 += 1;

CURRENT\_ALARM\_TARGET = ALARM\_TARGET\_1;

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

}

// CURRENT\_ALARM\_COUNT += 1;

delay(MEASURE\_DELAY);

// CURRENT\_ALARM\_COUNT += 1;

long distance = measure();

DISTANCE\_INCHES = distance/25.4;

// if (debug == 999)DISTANCE\_INCHES = 2222.2; //Force a value in each read

Serial.println(" ");

Serial.print("(");

Serial.print(CURRENT\_ALARM\_COUNT);

Serial.print(")");

Serial.print("- SENSOR ");

Serial.print(Current\_Sensor);

Serial.print(" --- DISTANCE: ");

Serial.print(DISTANCE\_INCHES);

Serial.print(" ---------------- ");

Serial.print(ECHO\_PIN);

Serial.print(" ");

Serial.println(TRIGGER\_PIN);

check\_alarm();

}

long measure()

{

long measureSum = 0;

for (int i = 0; i < MEASURE\_SAMPLES; i++)

{

delay(MEASURE\_SAMPLE\_DELAY);

measureSum += singleMeasurement();

}

return measureSum / MEASURE\_SAMPLES;

}

long singleMeasurement()

{

long duration = 0;

// Measure: Put up Trigger...

digitalWrite(TRIGGER\_PIN, HIGH);

// ... wait for 11 µs ...

delayMicroseconds(11);

// ... put the trigger down ...

digitalWrite(TRIGGER\_PIN, LOW);

// ... and wait for the echo ...

duration = pulseIn(ECHO\_PIN, HIGH);

return (long) (((float) duration / USONIC\_DIV) \* 10.0);

}

int check\_alarm()

{

// if (Current\_Sensor== 1)

// {

// CURRENT\_ALARM\_COUNT += 1;

// Serial.println("-----------------Count Incremented------------------: ");

// Serial.print(ALARM\_COUNT\_1);

// Serial.println(" ");

if (CURRENT\_ALARM\_COUNT > ALARM\_COUNT\_SETPOINT)

{

Serial.println(" ");

Serial.print("(");

Serial.print(CURRENT\_ALARM\_COUNT);

Serial.print(")");

Serial.print(" <1> Alarm Count ");

Serial.print(CURRENT\_ALARM\_COUNT);

Serial.print(" Greater than Setpoint ");

Serial.print(ALARM\_COUNT\_SETPOINT);

Serial.println(" ");

}

if (ALARM\_ON\_HIGH\_DISTANCE == 1)

{

if (DISTANCE\_INCHES > (CURRENT\_ALARM\_TARGET + ALARM\_DEADBAND))

{

Serial.println(" ");

Serial.print("(");

Serial.print(CURRENT\_ALARM\_COUNT);

Serial.print(")");

Serial.print(" <2> DISTANCE\_INCHES ");

Serial.print(DISTANCE\_INCHES);

Serial.print(" GT ALARM\_TARGET ");

Serial.print(CURRENT\_ALARM\_TARGET);

Serial.print(" + DEADBAND ");

Serial.print(ALARM\_DEADBAND);

Serial.print(" (OK 2 ALM)------");

Serial.println(" ");

if (CURRENT\_ALARM\_COUNT > ALARM\_COUNT\_SETPOINT)

{

digitalWrite(BUZZER\_PIN, HIGH);

if (debug == 1)

{

Serial.println(" ");

Serial.print("(");

Serial.print(CURRENT\_ALARM\_COUNT);

Serial.print(")");

Serial.print(" \*\*\* <3> \*\*\*\*\*\* BUZZER ON ALARM \*\*\*\*\*\* SENSOR ");

Serial.print(Current\_Sensor);

Serial.print(" Distance: ");

Serial.println(DISTANCE\_INCHES);

}

}

}

}

else

if (DISTANCE\_INCHES < (CURRENT\_ALARM\_TARGET - ALARM\_DEADBAND))

{

Serial.println(" ");

Serial.print("(");

Serial.print(CURRENT\_ALARM\_COUNT);

Serial.print(")");

Serial.print(" <2> DISTANCE\_INCHES ");

Serial.print(DISTANCE\_INCHES);

Serial.print(" LT ALARM\_TARGET ");

Serial.print(CURRENT\_ALARM\_TARGET);

Serial.print(" - DEADBAND ");

Serial.print(ALARM\_DEADBAND);

Serial.print(" (OK 2 ALM)------");

Serial.println(" ");

if (CURRENT\_ALARM\_COUNT > ALARM\_COUNT\_SETPOINT)

{

digitalWrite(BUZZER\_PIN, HIGH);

if (debug == 1)

{

Serial.println(" ");

Serial.print("(");

Serial.print(CURRENT\_ALARM\_COUNT);

Serial.print(")");

Serial.print(" \*\*\* <3> \*\*\*\*\*\* BUZZER ON ALARM \*\*\*\*\*\* SENSOR ");

Serial.print(Current\_Sensor);

Serial.print(" Distance: ");

Serial.println(DISTANCE\_INCHES);

}

}

}

if (CURRENT\_ALARM\_COUNT > ALARM\_COUNT\_RESET)

{

Serial.println(" ");

Serial.print("(");

Serial.print(CURRENT\_ALARM\_COUNT);

Serial.print(")");

Serial.print(" <4> ");

Serial.print(" ALARM COUNT ---- RESET ----- ---- RESET ----- ---- RESET -----: ");

Serial.print(CURRENT\_ALARM\_COUNT);

Serial.println(" ");

if (Current\_Sensor == 1)

{

ALARM\_COUNT\_1 = 0;

}

if (Current\_Sensor == 2)

{

ALARM\_COUNT\_2 = 0;

}

// CURRENT\_ALARM\_COUNT = 0;

digitalWrite(BUZZER\_PIN, LOW);

}

}

void print\_config()

{

Serial.println(" ");

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

Serial.print("( Alarm Target Sensor 1: ");

Serial.println(ALARM\_TARGET\_1);

Serial.print("( Alarm Target Sensor 2: ");

Serial.println(ALARM\_TARGET\_2);

Serial.print("( Alarm Target Deadband: ");

Serial.println(ALARM\_DEADBAND);

Serial.print("( Alarm Count Setpoint: ");

Serial.println(ALARM\_COUNT\_SETPOINT);

Serial.print("( Alarm Count Reset : ");

Serial.println(ALARM\_COUNT\_RESET);

Serial.print("( Alarm on Hi or Lo : ");

Serial.print(ALARM\_ON\_HIGH\_DISTANCE);

Serial.println(" Alarm on Hi (1) or Lo (0)");

delayMicroseconds(500);

}