#include <MultiFuncShield.h>

//Modified from Hacktronics Pat McMahon 9/6/2022

//MFS10

// Heart Monitor

void initializeSensorReading();

int data[4];

byte dataIdx=0;

byte pulseDetected = false;

int lastPulseTime = -1;

void setup() {

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

MFS.initialize();

initializeSensorReading();

//Serial.begin(9600);

}

void loop()

{

if (MFS.getTimer() == 0)

{

MFS.setTimer(10000); // reset millisecond countdown timer.

if (lastPulseTime != -1)

{

lastPulseTime = 10000 + lastPulseTime;

}

}

int sensorValue = analogRead(A5); // read the sensor.

if (sensorValue < 20 || sensorValue > 970)

{

// Sensor hasn't normalized, check how long for in milliseconds.

if (lastPulseTime != -1 && lastPulseTime - MFS.getTimer() > 700)

{

initializeSensorReading();

}

}

else if (sensorValue > (3 \* 1024) / 5) // value is rising, so must be start of a pulse.

{

if (!pulseDetected)

{

pulseDetected = true;

if (lastPulseTime == -1)

{

lastPulseTime = MFS.getTimer();

}

else

{

int pulsePeriod = lastPulseTime - MFS.getTimer(); // calculate time between pulses in millseconds.

lastPulseTime = MFS.getTimer();

int bpm = 60000 / pulsePeriod; // calculate beats per minute.

if (bpm < 45 || bpm > 230) // bpm is outside acceptible range, so clear the data buffer.

{

initializeSensorReading();

}

else

{

// bpm is within range, but still need to calculate average.

data[dataIdx++] = bpm;

if (dataIdx >= 4)

{

dataIdx = 0;

}

if (data[0] && data[1] && data[2] && data[3]) // check if data buffer is full before calculating avg bpm.

{

int avgBpm = (data[0] + data[1] + data[2] + data[3]) / 4;

MFS.blinkDisplay(DIGIT\_ALL, OFF);

MFS.write(avgBpm);

MFS.beep();

}

else

{

// buffer not full, so blink the display.

MFS.blinkDisplay(DIGIT\_ALL, ON);

}

}

}

}

}

else if (sensorValue < (1024 / 2)) // value is falling, so must be end of pulse.

{

pulseDetected = false;

}

//Serial.println(sensorValue);

//delay(10);

}

// Initialize the read buffer and display.

void initializeSensorReading()

{

lastPulseTime = 0;

dataIdx = 0;

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

{

data[i] = 0;

}

MFS.write(0);

MFS.blinkDisplay(DIGIT\_ALL, OFF);

}