ENSC 351

How To Guide:

Grove Ear-Clip Heart Rate Sensor on BeagleBone Green Guide

Jerome Ignacio - jeromei@sfu.ca
Jia Ming Li (James) - jml44@sfu.ca
John Alvarado - jpalvara@sfu.ca
Ruoyi Zhao - ruoyiz@sfu.ca

Table of Contents

1. Introduction

Heart pulse is the most important aspect in being able to track the vitals of one's body. It is used to measure heart beats per minute amongst many uses such as checking if someone is still alive via ECG or via calories expended in an exercise. For our purposes, we use the heart pulse to calculate a person's heart rate per minute in order to extract data such as basal metabolic rate and calories burned. This how-to-guide provides an in-depth guide on how to interface with a GPIO heart rate sensor using BeagleBone Green via the Linux terminal and via C code; this guide talks through how to measure heart rate using a GPIO heart pulse sensor and how to use the data extracted from the sensor to be able to make calculations on them. Other guides exist for interfacing with the ear-clip heart rate sensor but they all use Arduino in order to accomplish this.

Brief background is needed in order to understand how the sensor works in detecting pulses; knowing what a pulse is. A pulse can be measured in multiple ways but amongst them is via photoplethysmography (PPG). PPG is an optical measurement method that detects changes in concentration/volume of hemoglobin according to the changes in the absorption of infrared light (IR). Accordingly, running an IR light through a thin part of the body where light can be reliably absorbed on the other side will show variations in the absorption of its original signal strength. As such, by sandwiching the ear between the photo sensor and the IR light, we can detect pulses.

2. Requirements

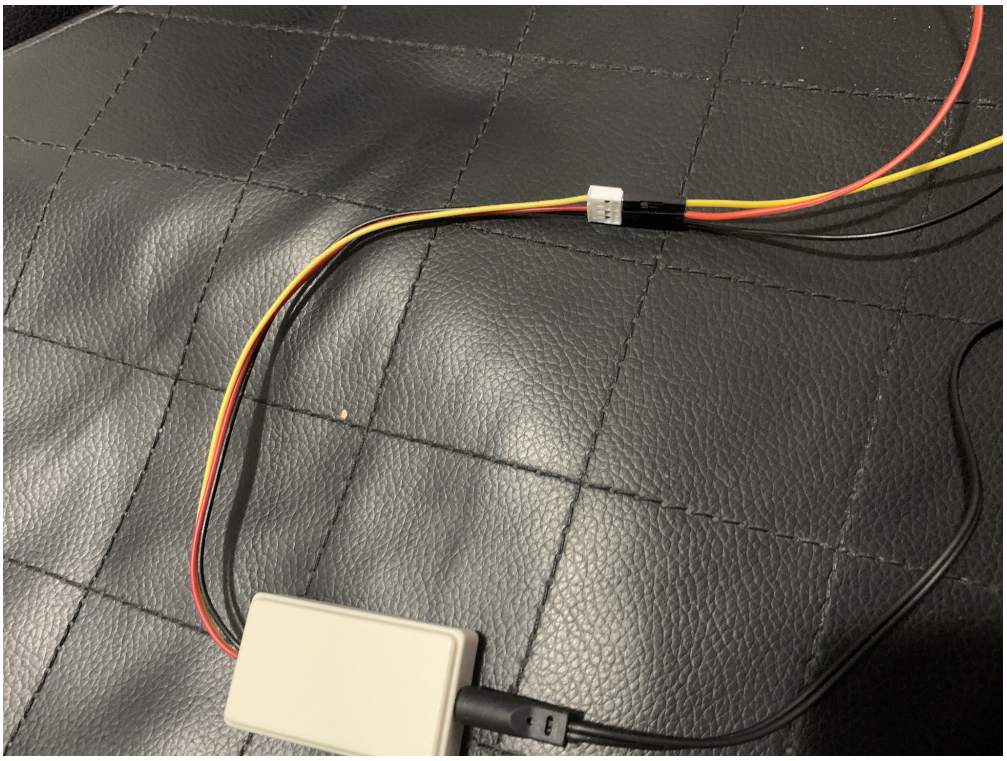
Required Hardware

- **BeagleBone Green**
- **Seeed Grove Ear-Clip Heart Rate Sensor**

1. Seeed Grove Ear-Clip Heart Rate Sensor

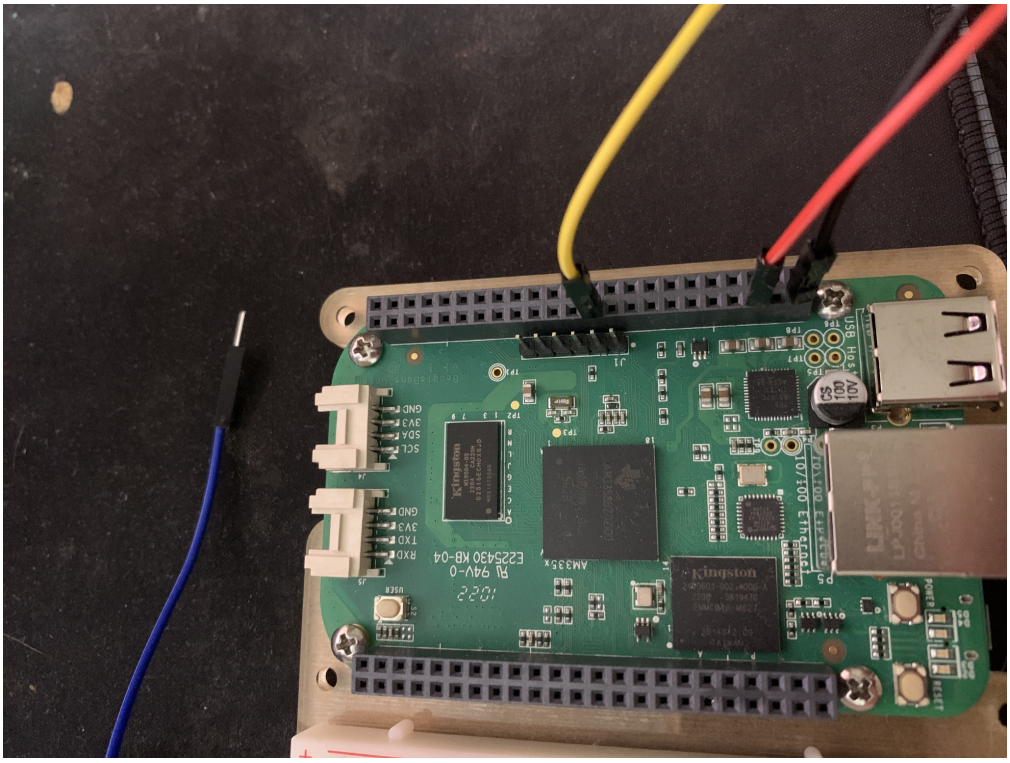
1.1. Setting up Heart Rate Sensor for connection

Connect three male-male jumper wires to the ends of the Grove Cable as shown below:



1.2. Connecting Heart Rate Sensor to the Beaglebone

Connect the yellow jumper wire to the P9 pin 22 on the BeagleBone Green. Connect also the power (red) and ground (black) ends of the male-male jumper wires to the P9 pins 1 (black) and 3 (red) on the BeagleBone Green:



2. BeagleBone Green

2.1. Enabling the GPIO pins on the BeagleBone Green

2.1.1. Setting up the pin for GPIO use

On the BeagleBone, execute the following command to setup P9 pin 22 on the board:

$(bbg) **config-pin p9.22 gpio**

2.1.2. Checking the pin is set

On the BeagleBone , execute the following command to check that the pin is set to GPIO:

$(bbg) **config-pin -q p9.22 gpio**

2.2. Enabling the GPIO for interaction with Linux

Proceed to the sysfs directory on the BeagleBone:

$(bbg) **config-pin -q p9.22 gpio**

Then tell Linux that the pin will be used as GPIO via:

$(bbg) **echo 30 > export**

3. C Code

1. **Configuring/ Setting Up**
   The following code was taken from Dr. Brian for running Linux commands via C.

```c
static void runCommand(char *command)
{
    // Execute the shell command (output into pipe)
    FILE *pipe = popen(command, "r");
    // Ignore output of the command; but consume it
    // so we don't get an error when closing the pipe.
    char buffer[1024];
    while (!feof(pipe) && !ferror(pipe)) {
        if (fgets(buffer, sizeof(buffer), pipe) == NULL)
        break;
        // printf("--> %s", buffer); // Uncomment for debugging
        }
        // Get the exit code from the pipe; non-zero is an error:
    int exitCode = WEXITSTATUS(pclose(pipe));
    if (exitCode != 0) {
        perror("Unable to execute command:");
        printf(" command: %s\n", command);
        printf(" exit code: %d\n", exitCode);
    }
}
```

Using this function, we can setup the pin to be used for GPIO in C as follows:

```c
char *configPin = "config-pin p9.22 gpio";
runCommand(configPin);
```

2. **Reading**
   Modified from Dr. Brian's GPIO Guide for reading a pin.
   Reading from the heart rate sensor can be done via C using the following code, where the file location is set to the Linux GPIO number. Value may hold 48 or 49 depending on whether a pulse has been detected by the sensor according to preliminary knowledge that when the heart pumps blood, a higher concentration

of hemoglobin is present. Accordingly, the IR light produced by the IR led is absorbed by the hemoglobin which causes modulation in the amount of light that can be detected by the photosensor. This modulation to value 49 is a heartbeat.

```
char readHeartBeats()
{
    FILE *pFile = fopen("/sys/class/gpio/gpio2/value", "r");
    if (pFile == NULL)
    {
        printf("ERROR: Unable to open file.\n");

    }
    const int MAX_LENGTH = 1024;
    char buff[MAX_LENGTH];
    fgets(buff, MAX_LENGTH, pFile);

    fclose(pFile);
    return buff[0];
}
```

3. **Calculating**
   Using the data that we read using the heartbeat function and collecting enough data for a minute, we calculate the BMR, Total Calories and Calories Burned Per Minute Biking as follows:

   a. **BMR**

```
void calculateBMR(int kg, int cm, int yrs, char *sex)
{
    if (strcasecmp(male, sex))
    {
        BMR = 88.362 + (13.397 * kg) + (4.799 * cm) - (5.677 * yrs);
    }
    else if (strcasecmp(female, sex))
    {
        BMR = 447.593 + (9.247 * kg) + (3.098 * cm) - (4.330 * yrs);
    }
}
```

b. **Total Calories**

```
void caloriesBurnedBiking()
{



    if (strcasecmp(male, getSex()))
    {
        calories = calories + ((getAge() * 0.2017) - (getWeight() *
0.1988) + (6*getHeartBeat() * 0.6309) - 55.0969)/0.697;
    }
    else if (strcasecmp(female, getSex()))
    {
        calories = calories + ((getAge() * 0.074) - (getWeight() *
0.1263) + (6*getHeartBeat() * 0.4472) - 20.4022)/0.697;
    }
}
```

c. **Calories**

```
void caloriesBurnedBikingPerMin()
{


    //char * sex = getSex();

    if (strcasecmp(male, getSex()))
    {
        calories = calories + ((getAge() * 0.2017) - (getWeight() *
0.1988) + (getHeartBeat() * 0.6309) - 55.0969)*(1.0/4.184);
    }
    else if (strcasecmp(female, getSex()))
    {
```

```
        calories = calories + ((getAge() * 0.074) - (getWeight() *
0.1263) + (getHeartBeat() * 0.4472) - 20.4022)*(1.0/4.184);
    }
}
```

4. **Putting Together**

   The following code for main shows how to use the functions in action. First we obtain the necessary information to be able to calculate the BMR and calories such as age, weight, height and sex. Then, we run the function for BMR before we run the thread that extracts the data from the sensor and calculates them.

```
#include <stdio.h>
#include <ctype.h>
#include <time.h>
#include <stdint.h>
#include <stdlib.h>
#include <unistd.h>
#include <stdbool.h>
#include <strings.h>
#include "caloriesRead.h"
#include "printTerminal.h"

int main()
{

    printf("Welcome!!!\n");

    printf("Please enter your age: ");
    setAge();
    printf("Age: %d\n", getAge());

    printf("Please enter your weight: ");
    setWeight();
    printf("Weight: %d\n", getWeight());

    printf("Please enter your height: ");
    setHeight();
    printf("Height: %d\n", getHeight());
```

```
    printf("Please enter your sex: ");
    setSex();
    printf("Sex: %s\n\n", getSex());

    calculateBMR(getWeight(), getHeight(), getAge(), getSex());

    printf("Your basal metabolic rate is (Calories Per Day): %d\n",
(int)getBMR());

    printf("\nScanning for heartbeats per min!!!!!\n\n");

    startCaloriesReadThread();

    freeCaloriesReadThread();
    return 0;
}
```

References:

https://www.adafruit.com/product/1782?gclid=Cj0KCQiAkMGcBhCSARIsAIW6d0C1yURVb4tos4NC8Jd9M_sHzufuUHUOsc0R_GRiWt57IRWJtdx4oSUaAqufEALw_wcB

https://files.seeedstudio.com/wiki/Grove-Ear-clip_Heart_Rate_Sensor/res/PPG%20Sensor%20System.pdf

https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1748-1716.1949.tb00147.x

https://seermedical.com/blog/calculate-heart-rate-ecg
https://www.analog.com/media/en/technical-documentation/data-sheets/max30100.pdf
https://www.analog.com/media/en/technical-documentation/data-sheets/max30102.pdf
https://medicine.uiowa.edu/iowaprotocols/pulse-oximetry-basic-principles-and-interpretation
https://lastminuteengineers.com/max30102-pulse-oximeter-heart-rate-sensor-arduino-tutorial/
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6426305/