**Arduino group project**

**Cpr feedback System**

Group members:

Arsh Sood

Anmol Toor

Apurva Pandoh

Anurag

Deepanshu

**Overview**

The purpose of this work is to implement a cardiopulmonary resuscitation feedback device to recover a patient from cardiac arrest. This feedback device is implemented using AT Mega 328P microcontroller. The force sensitive resistor is used to obtain the pressure applied to the patient’s chest while performing chest compressions. Light Emitting Diode is used as an indicator to understand whether the chest compression performed is good or bad. 30 chest compressions make a single chest compression cycle. For each cycle, parameter like heart rate, beats per minute count and the time rate for a particular cycle are calculated. The prototype is initially implemented in Arduino integrated development environment. The hardware is then extended to MATLAB software. For MATLAB implementation, the parameter found are computed and various graphs are plotted like the heart rate versus cardiopulmonary resuscitation count, beats per minute count versus time and heart rate versus time. From these, it is observed that BPM count decreases with increase in CPR cycle time.

**What Is Cpr?**

Cardiopulmonary resuscitation (CPR) is an emergency technique to replenish the blood flow to the heart and brain at the time of heart failure. CPR includes chest compression with artificial respiration. The general procedure of performing a CPR includes giving two full breaths to the mouth of the patient i.e. mouth to mouth respiration. It is then followed by giving chest compressions to the patient. According to American Health Association (AHA), 5 to 6 mm and a heart rate of 100 to 120 beats per minute(bpm) is the standards taken for chest compression. CPR can also be performed in animals apart from humans.Recent statistics from the WHO (World Health Organization) shows that cardiovascular diseases have become the world’s leading global killer. As per WHO, 17.9 million people die every year due to cardiovascular diseases. This number corresponds to 31% of all deaths worldwide. Cardiovascular deaths are increasing year by year. 80% of cardiovascular deaths occur due to tobacco use, high blood pressure, and consumption of food with high salt content. Also, 75% of cardiovascular deaths occur in developing countries due to premature deaths from heart attacks and strokes. To counter this problem, there are many CPR feedback devices available in the market. Some of them include portable devices such as smartphones, smartwatches etc. These devices can continuously monitor the heart rate and can give continuous feedback to the patient. Mechanical CPR devices provide high quality CPR. If the high-quality manual chest compressions are challenging, automatic devices can meet the need for improving the outcome of CPR. But the AHA guidelines do not recommend the routine use of automatic devices. Recent studies make use of Arduino board for giving CPR.The cross platform, simple and clear programming environment of Arduino are the major benefits of using the kit.Arduino integrated development environment (IDE) is the platform used for the Arduino board.

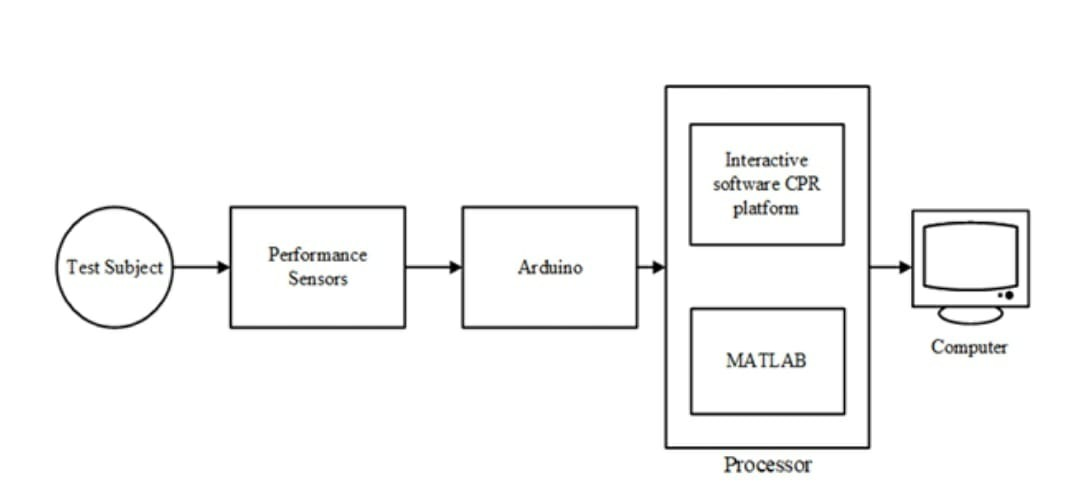
**Components Used**

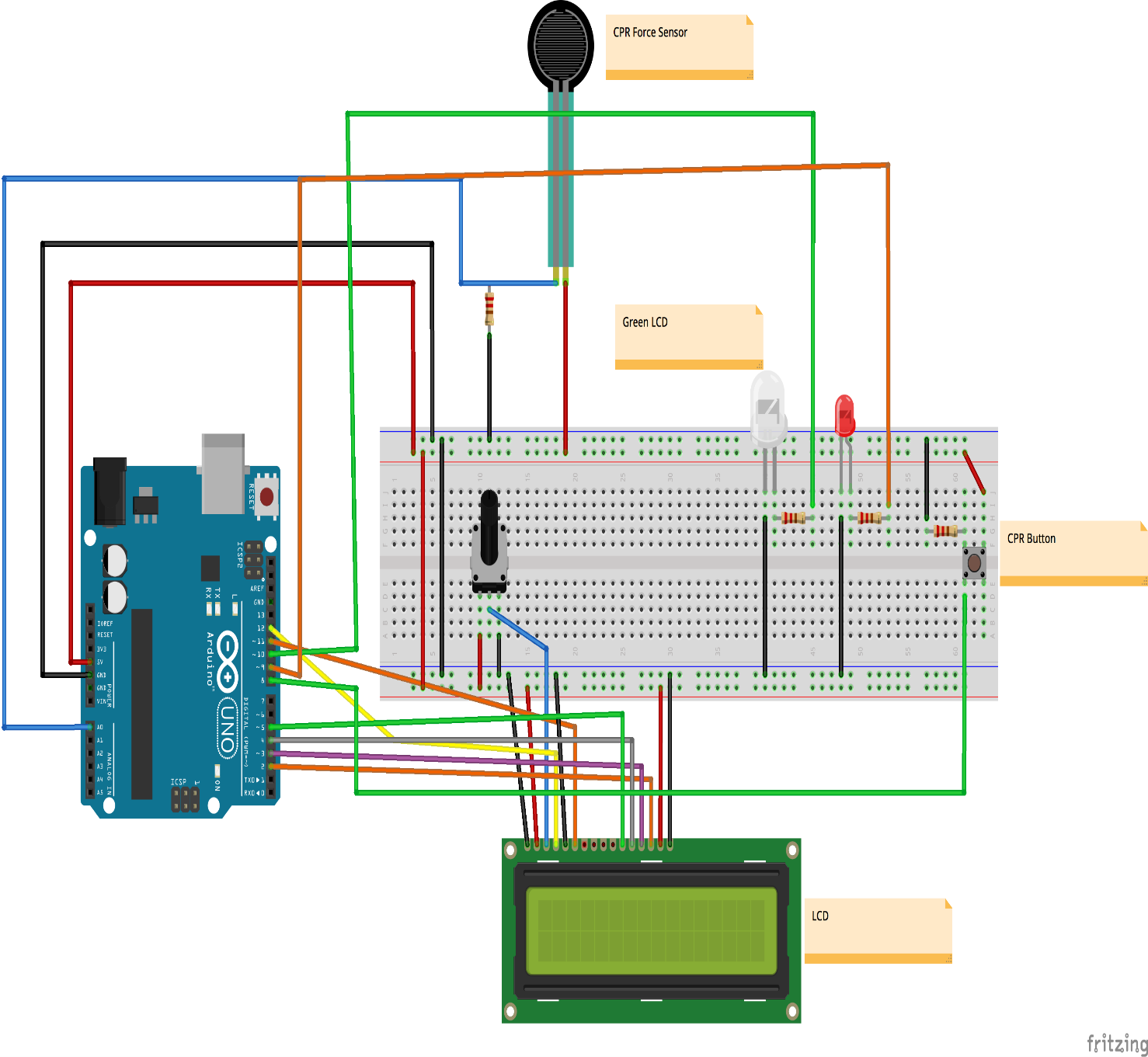
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | |
| Arduino UNO | |  | | --- | | [Arduino UNO](https://www.hackster.io/arduino/products/arduino-uno1?ref=project-58e6bd) | |  | |  | |  | |  |  |
| Pushbutton switch 12mm | |  | | --- | | [SparkFun Pushbutton switch 12mm](https://www.hackster.io/sparkfun/products/pushbutton-switch-12mm?ref=project-58e6bd) | |  | |  | |  | |  |  |
| Standard LCD - 16x2 White on Blue | |  | | --- | | [Adafruit Standard LCD - 16x2 White on Blue](https://www.hackster.io/adafruit/products/standard-lcd-16x2-white-on-blue?ref=project-58e6bd) | |  | |  | |  | |  |  |
| Breadboard (generic) | |  | | --- | | Breadboard (generic) | |  | |  | |  | |  |  |
|  | |  | | --- | | Adafruit Force Sensitive Resistor | |  | |  |  | |  |
| LED (generic) | |  | | --- | | LED (generic) | |  | |  | |  | |  |  |
| Rotary potentiometer (generic) | |  | | --- | | Rotary potentiometer (generic) | |  | |  | |  | |  |  |
| Alligator Clips | |  | | --- | | Alligator Clips | |  | |  | |  | |  |  |
| Jumper wires (generic) | |  | | --- | | Jumper wires (generic) | |  | |  | |  | |  |  |
| Perma-Proto Breadboard Half Size | |  | | --- | | Perma-Proto Breadboard Half Size | |  | |  | |  | |  |  |
|  | | | | | | | |
| Arduino IDE | |  | | --- | | [Arduino IDE](https://www.hackster.io/arduino/products/arduino-ide?ref=project-58e6bd)(Software used) | |  | |  | | | |  |  |
|  | | | | | | | |
| Soldering iron (generic) | |  | | --- | | Soldering iron (generic) | |  | |  | | | |  |  |

**Features Available**

As most of us working in health care training know, we sometimes have to rely on creating or hacking trainers to better suit the needs of the program and the learners. One of the newest projects I have been working on includes upgrading a CPR manikin to include a CPR feedback device made with Arduino. There are plenty of CPR feedback devices available on the market, some use a cell-phone, a portable device, or a CPR manikin with these features included. This project was not meant to create something entirely new but to evaluate more cost-effective ways to build upon existing trainers.

The CPR feedback device consists of the following features:

* Real Time Rate Measurement
* Depth Indicator
* Time
* Chest Recoil Indicator
* CPR Cycle Counter
* 9v Battery Powered
* **Block Diagram**
* ****
* **Circuit and Wiring**



**Code**

/\*

CPR feedback Test sketch

-Using LCD to display results in real time

\*/

#include <LiquidCrystal.h>

int pressurePin = 0; //pressure reading pin

int pressureRead; //for reading pressure pin

const int buttonPin = 8; // pin for CPR count

unsigned long time; //to display time for compressions

const int pressGood = 10;

const int pressBad = 9;

int BPM; //final output calculated beats per min

float bpmCount; // to calculate BPM

float cprCount = 0; // counter for the number of button presses, changed from int, changed from float

int buttonState = 0; // current state of the button

int lastButtonState = 0; // previous state of the button

int cprTotal = 0; //total of the 5 stored values

int cprCycle = 0; //keeps track of CPR cycle

int timeSec, timeCPR, BPM\_1; //calculating cpr time

float totalTime;

unsigned long timeBegin, timeEnd;

LiquidCrystal lcd(12, 11, 5, 4, 3, 2); //LCD pins

void setup() {

pinMode(buttonPin, INPUT); //CPR button

pinMode(pressGood, OUTPUT); //for good release, led on Green

pinMode(pressBad, OUTPUT); // for bad release, led on Red

Serial.begin(9600); // start serial communication / uncomment to debug

Serial.println("Begin CPR");

lcd.begin(16, 2); //start LCD

lcd.home();

lcd.print("Begin CPR");

}

void loop() {

pressureRead = analogRead(pressurePin);

time = millis(); //start time

buttonState = digitalRead(buttonPin);

if (buttonState != lastButtonState) {

if (buttonState == HIGH) { //depth of at least 2in per AHA guidelines

lcd.clear();

lcd.home();

cprCount++; //add 1 to the running total

Serial.println(cprCount);

lcd.setCursor(0,0);

lcd.print("Count: ");

lcd.setCursor(7,0);

lcd.print(cprCount);

lcd.setCursor(10,0);

lcd.print("R: "); //rate BPM

if(cprCount == 1){

timeBegin = millis(); //beginning time

}

}

}

//pressureRead = analogRead(pressurePin);

if (pressureRead < 500){

digitalWrite(pressGood, HIGH);

digitalWrite(pressBad, LOW);

}

else if (pressureRead > 501){

digitalWrite(pressBad, HIGH);

digitalWrite(pressGood, LOW);

}

if (cprCount == 30){ // 30:2 ratio for CPR, 30 total

cprCount = 0; //restart count

timeEnd = millis(); //capture end time for set

//Serial.println(timeEnd);

totalTime = (timeEnd - timeBegin) / 1000; //convert Ms to seconds

Serial.print("Total Time: ");

Serial.println(totalTime);

bpmCount = ( 30.0 / totalTime); //bpmCount = float

Serial.print("BPM Count: ");

Serial.println(bpmCount);

BPM = (bpmCount \* 60.0);

Serial.print("Total time in sec: ");

Serial.println(totalTime);

lcd.home();

lcd.print("Rate (BPM): ");

lcd.setCursor(12,0);

lcd.print(BPM);

Serial.print("BPM is: ");

Serial.println(BPM);

lcd.setCursor(0,1);

lcd.print("Total Time: ");

lcd.setCursor(12,1);

lcd.print(totalTime);

cprCycle++;

Serial.print("CPR Cycle #: ");

Serial.println(cprCycle);

if (cprCycle == 5){

cprCycle = 0;

}

}

if (cprCount > 5 && cprCount <= 30) {

float rateTime = (millis() - timeBegin) / 1000; //changed int rateTime to unsigned long

//Serial.println(rateTime);

int rate = (cprCount / rateTime) \* 60; //changed int to unsigned int

//Serial.print("BPM is: ");

//Serial.println(rate);

if (rate >= 100 && rate <= 120 ){ //AHA guidelines 100-120 CPM

lcd.setCursor(12,0);

lcd.print("Good");

}

else {

lcd.setCursor(12,0);

lcd.print("----");

}

}

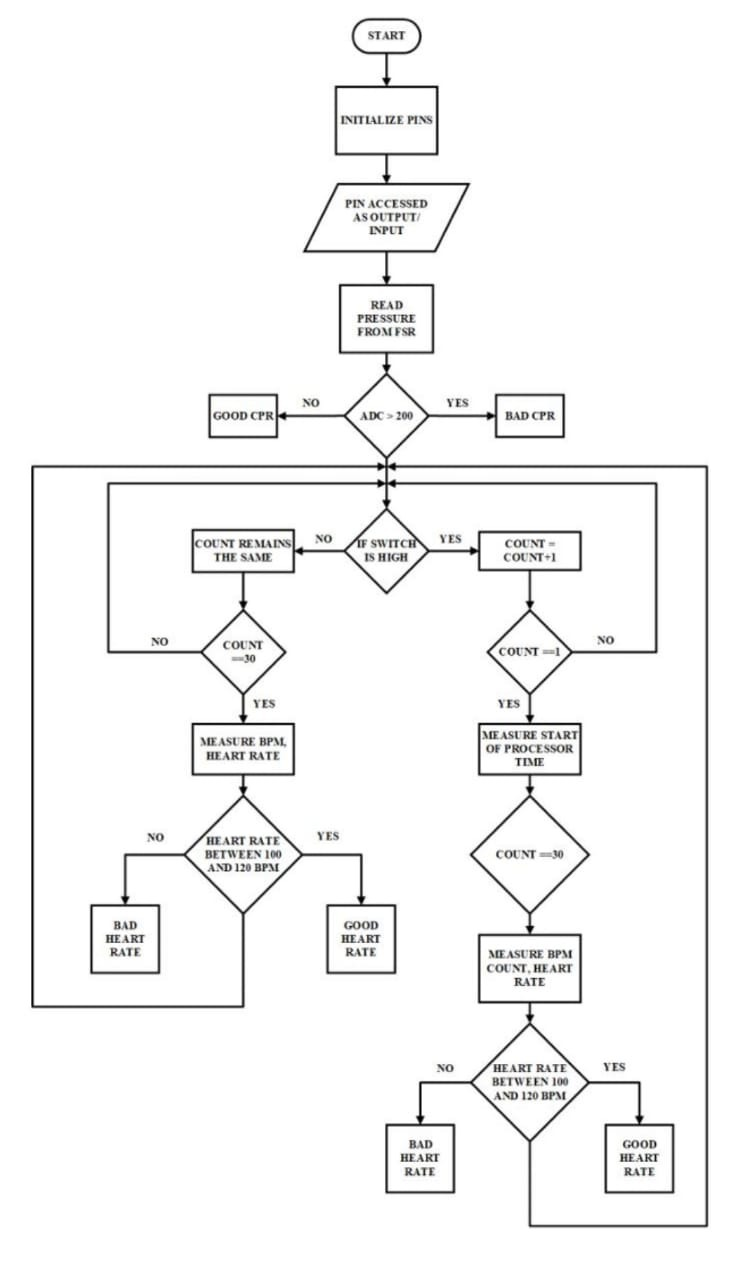
// Delay

delay(25); //changed from 50

lastButtonState = buttonState;

}

**Flowchart of all work done**



**Calculations**

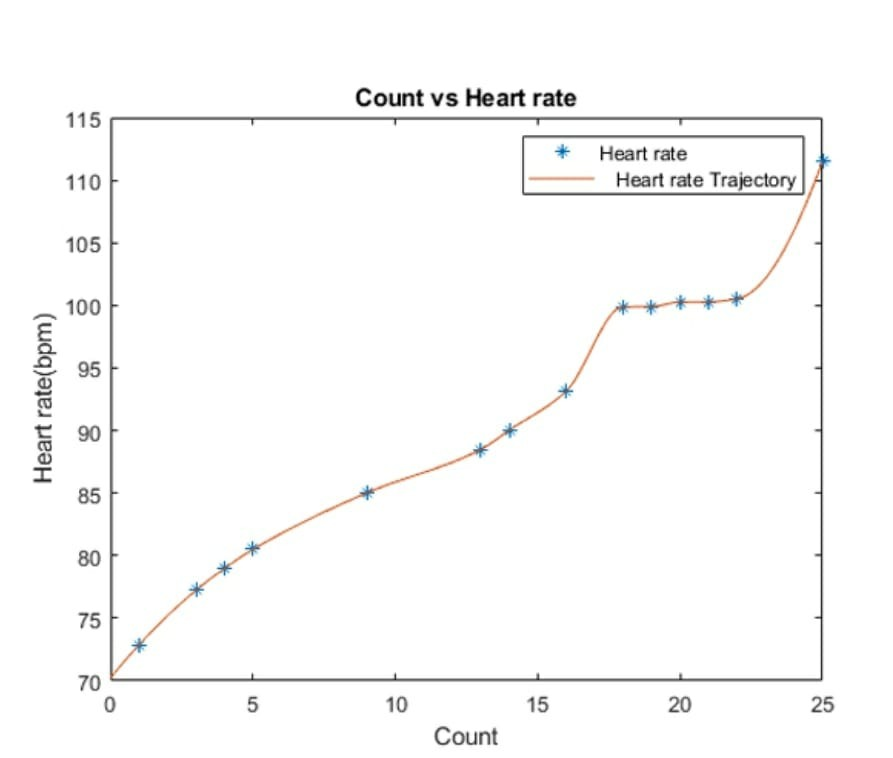
The number of times a person’s heart beats per minute is called the heart rate. The unit of heart rate is bpm i.e. beats per minute. A normal person’s heart beats in the range of 60 to 100 bpm. But according to the AHA guidelines the heart beat should range from 100-120 bpm. The maximum heart rate of individuals depends upon their age.

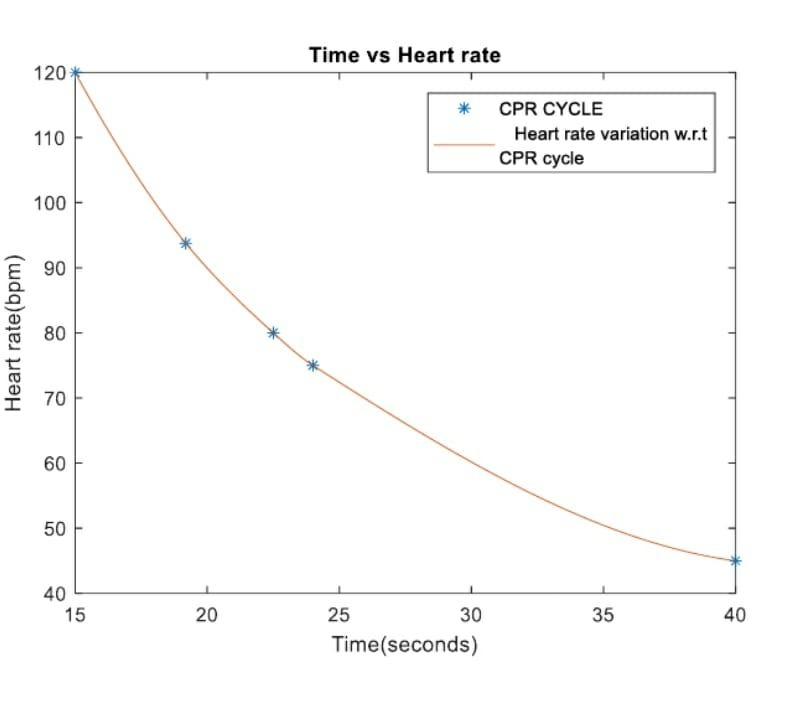
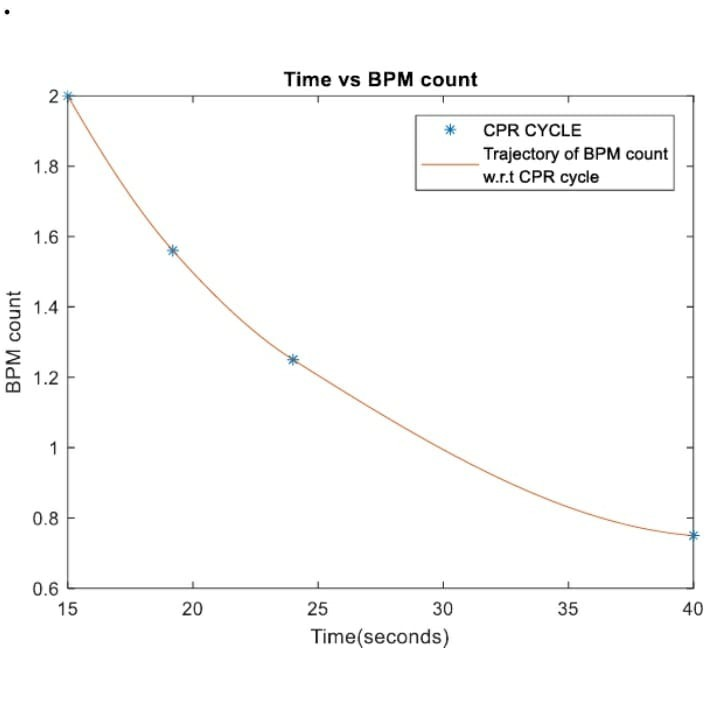
The equation (1) shows how heart rate is calculated. 𝐻𝑒𝑎𝑟𝑡𝑟𝑎𝑡𝑒 = 𝐶𝑃𝑅𝑐𝑜𝑢𝑛𝑡 𝑇𝑖𝑚𝑒𝑟𝑎𝑡𝑒 × 60 (1) The time rate is measured as the difference between the first and the last chest compression performed. The time rate in AtMega328P is measured in milliseconds.

Equation (2) gives the time rate of the CPR in seconds. 𝑇𝑖𝑚𝑒𝑟𝑎𝑡𝑒 = 𝑇𝑖𝑚𝑒𝑜𝑓𝑙𝑎𝑠𝑡𝑐𝑜𝑚𝑝𝑟𝑒𝑠𝑠𝑖𝑜𝑛 𝑇𝑖𝑚𝑒𝑜𝑓𝑓𝑖𝑟𝑠𝑡𝑐𝑜𝑚𝑝𝑟𝑒𝑠𝑠𝑖𝑜𝑛 × 1000 (𝑖𝑛 𝑠𝑒𝑐𝑜𝑛𝑑𝑠) BPM count is the ratio of chest compressions to time rate. It is another parameter found out after a cycle of chest compressions conducted. The unit of BPM count is second-1 . The total number of chest compressions per cycle is taken as 30.

Equation (3) is used to calculate the BPM count of the patient. 𝐵𝑃𝑀𝑐𝑜𝑢𝑛𝑡 = 30 𝑇𝑖𝑚𝑒𝑅𝑎𝑡𝑒 Using BPM count, the heart rate of the patient can be found out. Equation is used to find the heart rate from BPM count. 𝐻𝑒𝑎𝑟𝑡𝑟𝑎𝑡𝑒 = 𝐵𝑃𝑀𝑐𝑜𝑢𝑛𝑡 × 60

**Graphs**





**Conclusion**

This work proposes the development of a miniaturized CPR feedback device for treating persons suffering from cardiac arrest. The proposed method tracks pressure applied on FSR and number of compressions per minute. Furthermore the equipment is so handy that anyone can use it. Various graphical plots are found out and are analyzed. The results are found out to be reliable with the standard guidelines. This prototype can be further extended by using a mannequin and FSR with higher specification.