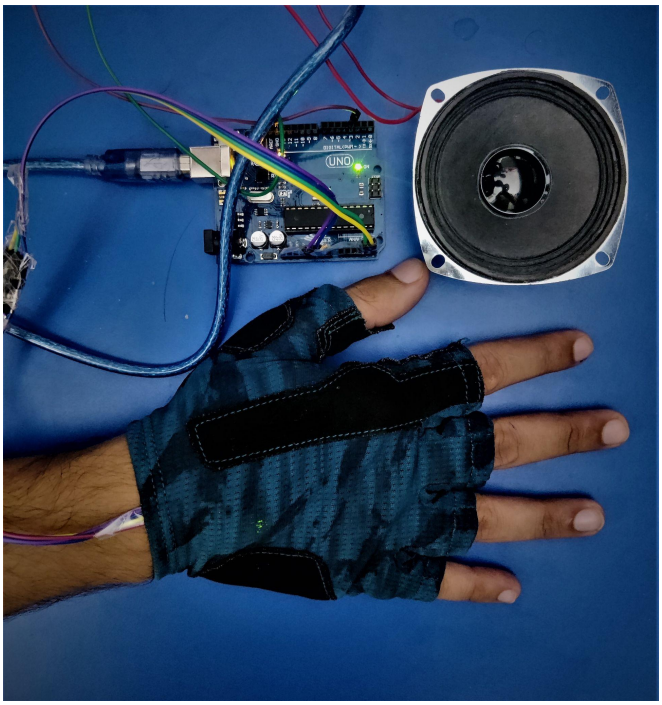
CPR Assist Device

December 2022

1.Overview

The CPR Glove device is a handy CPR coaching device that uses audio feedback, seen in Figure . It coaches the user through the correct steps to perform CPR, while also measuring the compression depth of the chest by taking the acceleration experienced and calculating the depth by assuming the standard time taken by one CPR (0.5 sec).



Based on the compression acceleration, this device can respond to the user by saying either “Push Harder” or “Good compressions” or “Too much pressure”. It also tells the user if the chest compressions are too deep into the chest which could result in rib cage breaking. This device uses an accelerometer, which is a small electromechanical device that will measure acceleration acted upon it. The accelerometer within this device measures the acceleration of each compression made onto the chest. This device uses a metronome system in order to coach the user to compress the chest at the rate of 120 beats per minute (bpm), which is the recommended rate provided by the AHA.

2.About CPR

Cardiopulmonary resuscitation (CPR) is a first aid to restore normal breathing and circulation in an accident, such as a heart attack or drowning, which involves pressurizing the chest from the outside in order to clean the airways connected to the lungs. After the heart stops, if the patient's blood circulation does not work for 4 to 6 minutes, the patient's brain begins to be damaged, and if it lasts more than 6 minutes, the brain and all organs in the patient's brain can stop functioning which can lead the patients to death, thus compressing the patient's chest to maintain a certain amount of blood circulation could be lifesaving. Standard 2010 Common CPR Guidelines recommends 30 chest compressions and 2 ventilations (the act of forcing air in and out of a person who cannot breathe easily) repeated in 1 cycle, and chest compressions should be performed at a depth of at least 5 cm (2 inches) at 100 speeds per minute. In an emergency like that even for highly skilled practitioners it is very difficult to achieve the accuracy of CPR while saving the patient’s life.

Therefore, there is a need for supplementary equipment to accurately perform CPR, here comes our CPR assist glove which will be worn on the hand of rescuer or medical staff to accurately perform the CPR.

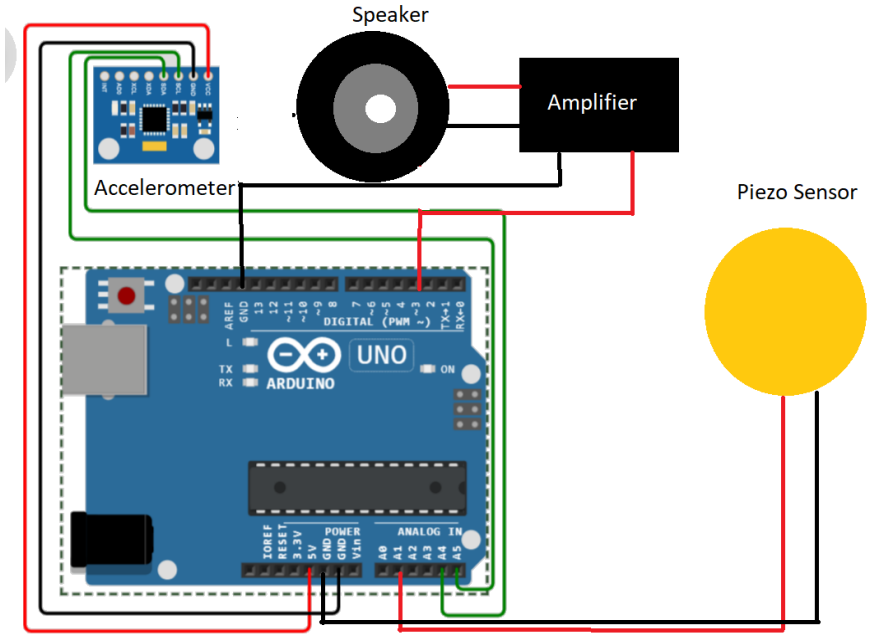
Our device will use the data from multiple sensors and assist the rescuer accordingly through proper audio signal using a speaker.

3. Literature Survey

Device	Measuring Mechanism(s)	Audio Feedback	Visual Feedback	Cost	Pros	Cons
CPRGlove (2020	Accelerometer - measures compression depth	Yes	Yes	$2500	Ergonomic, Reusable, Easy to Use, Measures compression depth	Not portable, Expensive, Not suited for repeatable use
CPR-Plus (2014)	Force Plate - measures compression force	Yes	Yes	Not on Market	Easy to use, No battery needed	Does not measure compression depth, Not ergonomic

Zoll Pocket CPR (2010)	Accelerometer - measures compression depth	Yes	Yes	$220	Portable, Lightweight, Easy to use, Reusable, Measures compression depth	Relies on a battery, Frequent prompts may confuse users
Laerdal CPR Meter (2010)	Accelerometer and Pressure Sensor – measures both compression force and depth	No	Yes	$900	Reusable, Portable, Lightweight, Saves and Records data, Measures compression force and depth	Relies on a battery, requires specific training in order to properly understand its output
CPR-Ezy (2008)	Pressure Sensor – measures compression force	Yes	Yes	$130	Portable, Lightweight, Easy to use, Reusable	Relies on a battery, does not measure compression depth, Not ergonomic

4.Device Schematic



5.Software & Hardware Requirements

1. Arduino IDE to write, debug and upload the source code.
2. Python Wizard to convert and add words of 16 bit 8000 hz audio file to the text-to-speech library.
3. Arduino UNO microcontroller (any ATMEGA328 microcontroller)
4. Piezo-Sensor to measure pressure.
5. Accelerometer (MPU6050) to measure acceleration to calculate depth
6. 10 watt speaker for audio output.
7. Amplifier circuit using LM386 amplifier.

6.Explanation of Schematic

First of all we connect the arduino and accelerometer(MPU6050) which we are doing here by connecting VCC and Ground of Arduino to the accelerometer VCC and Ground. After that, we will connect the SDA of Accelerometer to the A4 of the Arduino and SCL of Accelerometer to the A5 of the Arduino. In this way we configured our Accelerometer. Next we connect the piezo sensor to A3 of the arduino. We make an amplifier circuit using LM386 amplifier and connect its output to a 10 watt speaker, and connect its input to digital pin3 of Arduino and Ground of Arduino.

7.Source Code

```
#include <Wire.h>
#include "Talkie.h"
#include "Vocab_US_Large.h"
#include "Vocab_Special.h"
#include "Vocab_US_Acorn.h"
#include "Vocab_US_TI99.h"
Talkie voice;
const int MPU = 0x68; // MPU6050 I2C address
float AccX, AccY, AccZ;
float AvgAccZ;
int x = 0, breath=0;

void setup() {
 Serial.begin(115200);
  Wire.begin();                // Initialize comunication
  Wire.beginTransmission(MPU);       // Start communication with MPU6050 // MPU=0x68
  Wire.write(0x6B);                  // Talk to the register 6B
  Wire.write(0x00);                  // Make reset - place a 0 into the 6B register
  Wire.endTransmission(true);

  delay(20);
}
float check(){
  float start=0,hard=0,more=0,perfect=0,e=0, final=0 ;
  for(int i=0;i<50;i++)
  {
     Wire.beginTransmission(MPU);
     Wire.write(0x3B); // Start with register 0x3B (ACCEL_XOUT_H)
     Wire.endTransmission(false);
     Wire.requestFrom(MPU, 6, true); // Read 6 registers total, each axis value is stored in 2 registers
     //For a range of +-2g, we need to divide the raw values by 16384, according to the datasheet
     AccX = (Wire.read() << 8 | Wire.read())/16384.0 ; // X-axis value
     AccY = (Wire.read() << 8 | Wire.read())/16384.0 ; // Y-axis value
     AccZ = (Wire.read() << 8 | Wire.read())/16384.0 ; // Z-axis value

```

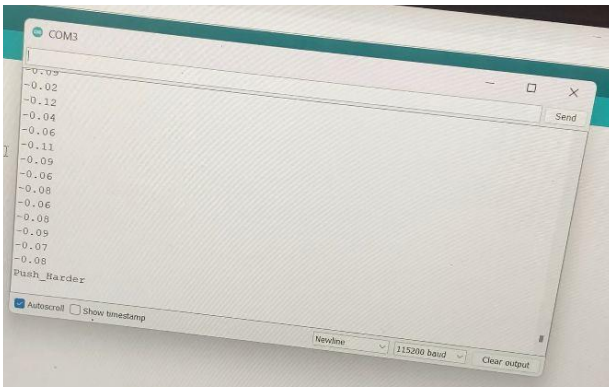
```
    Serial.println(RAccZ);
    delay(50);

}

if(e>=1){
     final = 10; // Flag for very high force
    }
else if(perfect>=3 || (more>=1 && perfect>=2)){ // Flag for perfect force

     final =20;
    }

else if(more>=2 || (hard>=1 && more>=1)){ // Flag for less than recommended force applied

     final = 30;
    }
else if(hard>=1){ // Flag when force is ineffective

     final = 40;
    }
else if(start){ // Flag to start CPR

     final = 50;
    }

return final;  // Return Flag value
}

void loop() {
  float value;

  bool force_exerted=analogRead(3); // Force value sensed by Piezo Sensor
  while(force_exerted){ // If piezo sensor has force >0
    Serial.println("PLEASE sTART cpr");
     value =check(); // Fetch depth value
       if(value==40){
         voice.say(sp2_PUSH);
         voice.say(spa_MORE);
         voice.say(spPAUSE2);
         voice.say(sp2_PUSH);
         voice.say(spa_MORE);
         voice.say(spPAUSE2);
          Serial.println("Push_Harder"); // Audio output to Push More
        }
        else if(value==30){

         voice.say(spa_MORE);
         voice.say(sp2_PRESSURE);
         voice.say(spPAUSE2);
         voice.say(spa_MORE);
         voice.say(sp2_PRESSURE);
         voice.say(spPAUSE2);

          Serial.println("push_bit_more"); // Audio output for a little more pressure
        }
        else if(value==20){
          voice.say(spa_YES);
          voice.say(spa_VERY);
          voice.say(spa_GOOD);
          voice.say(sp2_REPEAT);
          voice.say(spPAUSE2);
```

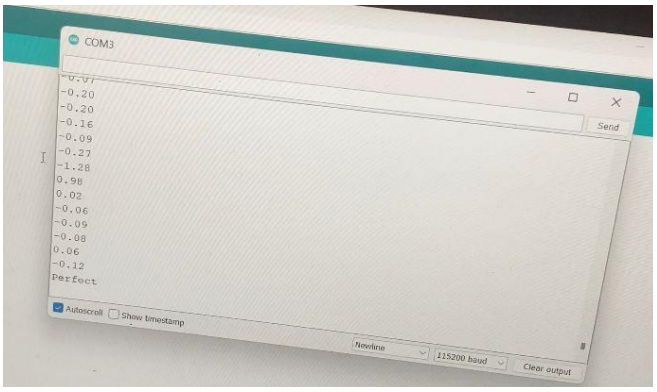
```
117.            voice.say(spa_YES);
118.            voice.say(spa_VERY);
119.            voice.say(spa_GOOD);
120.            voice.say(sp2_REPEAT);
121.            voice.say(spPAUSE2);
122.            Serial.println("Perfect"); // Audio output for perfect CPR
123.          }
124.          else if(value==10){
125.
126.            voice.say(spt_TOO);
127.            voice.say(spt_HIGHER);
128.            voice.say(spPAUSE2);
129.            voice.say(spt_TOO);
130.            voice.say(spt_HIGHER);
131.            voice.say(spPAUSE2);
132.            Serial.println("Too much pressure"); // Audio output for too much force
133.          }
134.          else if(value==50){
135.            voice.say(spt_PLEASE);
136.            voice.say(spa_START);
137.            voice.say(sp2_C);
138.            voice.say(sp2_P);
139.            voice.say(sp2_R);
140.            voice.say(spPAUSE2);
141.            Serial.println("you aren't doing anything"); // Audio output to start CPR
142.          }
143.      }
144. }
```

8.Algorithm and Working of the Code

1. Initially all the required library of code is included in the code using #include <”required_library”> which is upto line 6.
2. From line 6 to line 11 the required variable with its data type is defined.
3. In the void setup which is from line 13 to 22 we are writing code for initialization of printing, communication which is in line 14 and 15 after that we are starting the communication with MPU and later in line 19 we are ending the transmission.
4. In the function defined in the line 23 to 82 which is named as “check()” we are reading the value of relative acceleration(output acceleration - acceleration due to earth in z-axis) of the z-axis in loop for 50 times and initializing start, hard, more, perfect and e as 0.

5. Later each of the values is incremented every time it falls in the one of the 4 defined ranges which we are calculating by fixing the 5cm distance and 250 millisec(push or compression time) value which is ideal value for giving accurate CPR.
6. With the comparison of each occurrence of range we are again deciding what exactly we will assist in the total time of 5 CPR which we are achieving by giving a delay of 50 millisec in 50 time loop which results in a total time of 2500 millisec which is basically total time for 5 CPR.
7. Later in void loop() we are checking that our piezoelectric is sensing some pressure or not which will save us from giving unwanted value while moving the assistance randomly.
8. If our piezoelectric senses some value then only we will call the “check()” function and according to the value we will check again what we need to assist after a total time of 5 assists.
9. Once we will get the value from the function we will give the instruction to the speaker what to assist according to the range in which the value falls and the speaker will give the right instruction to the user of our CPR assistance accordingly.
10. As it’s mentioned above that point after 7th point is in void loop it will happen continuously until the piezoelectric will give no value means until the CPR is stopped.



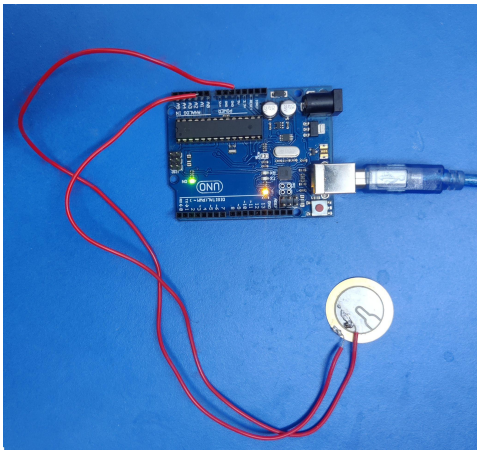


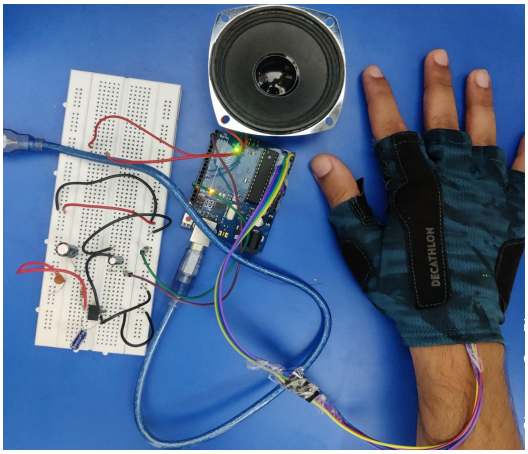
9.Application

These kinds of assistive devices are rare and are not frequently used because of their feasibility, mobility and accuracy. Our design will help us work on these aspects and be able to create a product that has potential to sustain and be extremely useful in the medical field among Hospitals and Teaching Institutions.

These devices will be so handy and feasible that we aim to put this in every first-aid kit available in the market as an essential tool.

Further, this product has potential to be integrated with a fully functional cpr device where this device will be used in conjunction with an expired air resuscitation device (EAR) which will help the patient with oxygen circulation as well as blood circulation during emergency.





Authors : Yash Kumar (21BEC059), Tarun Kumar (21BEC052), Vimal Verma (21BEC057), Ashish Kumar (21BEC006), Abhishek Kumar (21BEC002)

Mentors : Dr. Nataraj K S, Dr. Prakash Pawar

Research Paper Credits : Project Number: MQP JMS 1901 US Application Number: 62/807,637 IRB Number: IRB-19-0397 CPR Assistive Device A Major Qualifying Project Submitted to the Faculty of Worcester Polytechnic Institute in partial fulfillment of the requirements for the degree of a Bachelor of Science By Benjamin Abram Eric Chiem Zoey Foley John Ringuette ME & BME ME & ECE BME BME