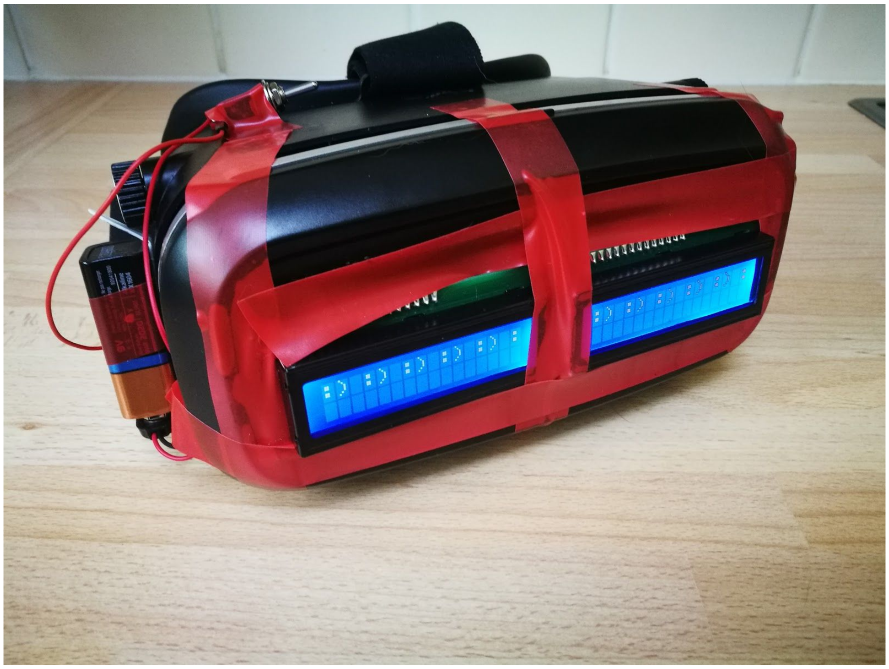
LCD Emotion goggles

By Rui Fan and Michael Hindley

About:

The project we worked on, the LCD emotion goggles, is a fun idea to have a wearable piece of technology that would read the wearer's voice input using a microphone module and display different emotions (Happy, sad, or excited) through LCD screens on the front of the goggles.



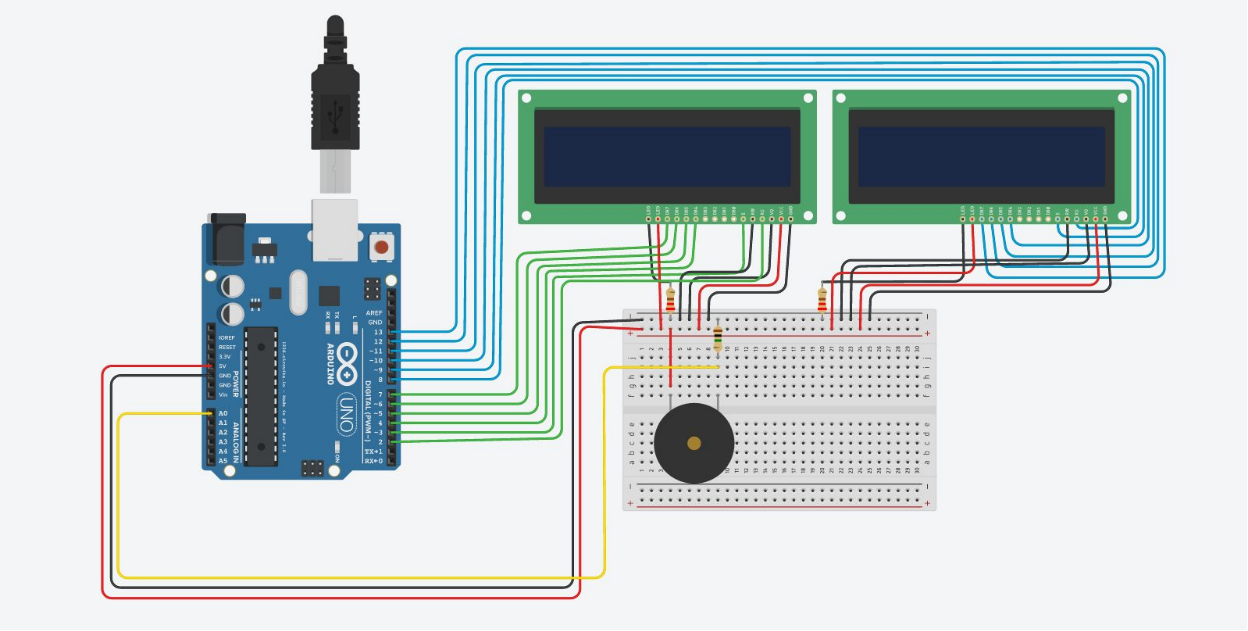
Idea and Design:

The inspiration for the project came was drawn from Wrench, one of the protagonists from the video game Watch Dogs 2. The character’s emotions ingame are entirely conveyed through an LCD mask he wears. He shows his excitement with ^ ^ when the codes run just right, and expresses his displeasure towards extracting an uncooperative toaster chip with x x. Since it had lots of potential as wearable tech, we decided to try and create a similar device using what we already have in our Arduino kits. Our final design has two inputs: a microphone on the bottom of the goggles to pick up the user's voice, and an on/off toggle switch to stop the device from using all of its battery charge when not in use.

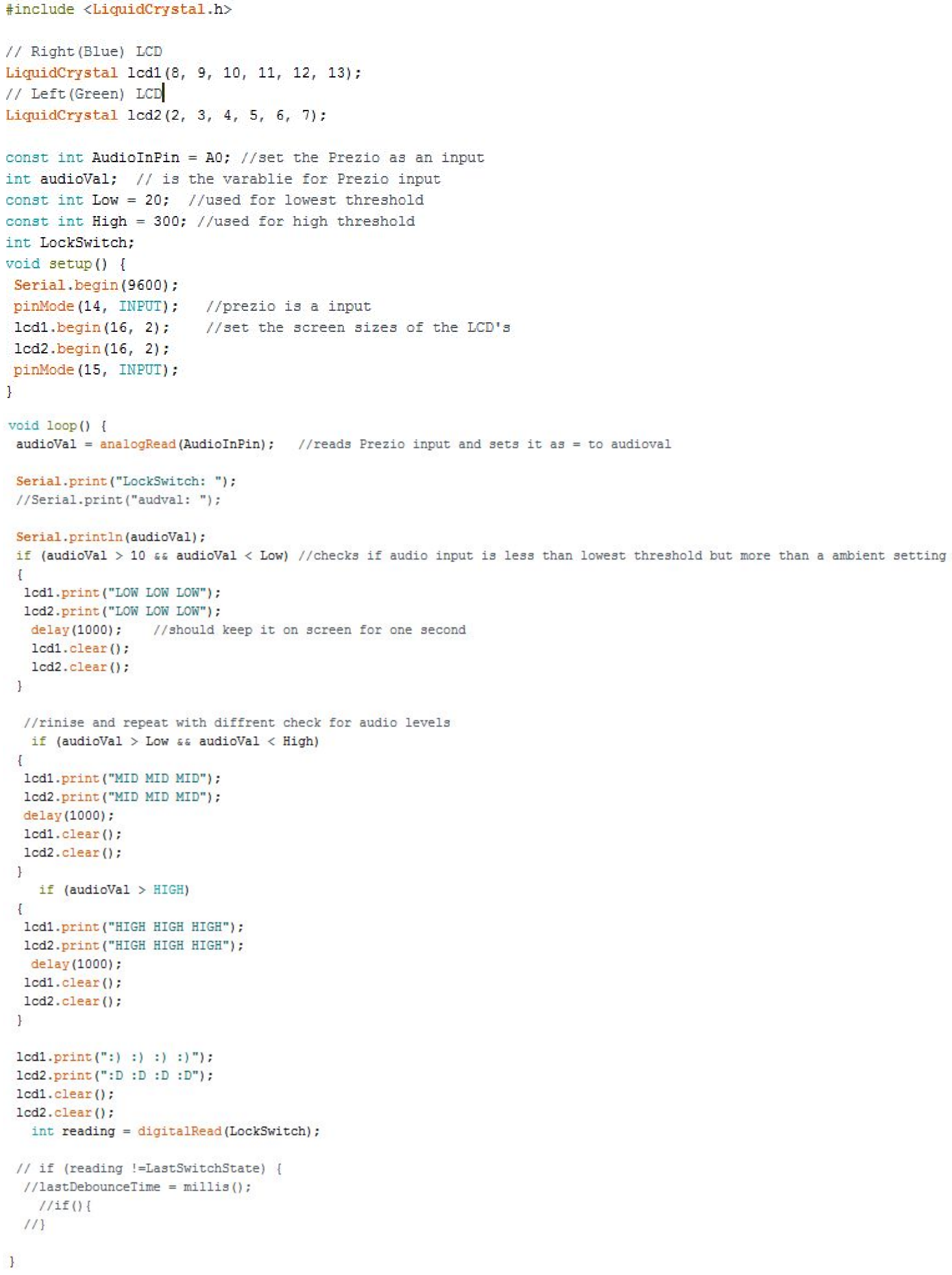
An early concern we had was not having enough input pins on the Arduino board and we might need two boards for the project to work, however this was quickly proven false once we started working on the project.

Plan and required tasks and materials:

The initial plan was concepted using Circuits.io, a free website that not only allowed us to plan the wiring but also test the code for the project. The components we used were the Arduino Uno, wires, resistors and two LCDs. for the audio input we initially used a piezo as a stand in for a microphone in our rough prototyping phase. We added a more powerful and sensitive microphone for the final version.



Initial plan in circuits.io



```
#include <LiquidCrystal.h>

// Right(Blue) LCD
LiquidCrystal lcd1(8, 9, 10, 11, 12, 13);
// Left(Green) LCD
LiquidCrystal lcd2(2, 3, 4, 5, 6, 7);

const int AudioInPin = A0; //set the Prezio as an input
int audioVal;  // is the varablie for Prezio input
const int Low = 20;  //used for lowest threshold
const int High = 300; //used for high threshold
int LockSwitch;
void setup() {
 Serial.begin(9600);
 pinMode(14, INPUT);   //prezio is a input
 lcd1.begin(16, 2);    //set the screen sizes of the LCD's
 lcd2.begin(16, 2);
 pinMode(15, INPUT);
}

void loop() {
 audioVal = analogRead(AudioInPin);   //reads Prezio input and sets it as = to audioval

 Serial.print("LockSwitch: ");
 //Serial.print("audval: ");

 Serial.println(audioVal);
 if (audioVal > 10 && audioVal < Low) //checks if audio input is less than lowest threshold but more than a ambient setting
 {
  lcd1.print("LOW LOW LOW");
  lcd2.print("LOW LOW LOW");
   delay(1000);    //should keep it on screen for one second
   lcd1.clear();
   lcd2.clear();
 }

  //rinise and repeat with diffrent check for audio levels
   if (audioVal > Low && audioVal < High)
 {
  lcd1.print("MID MID MID");
  lcd2.print("MID MID MID");
  delay(1000);
  lcd1.clear();
  lcd2.clear();
 }
    if (audioVal > HIGH)
 {
  lcd1.print("HIGH HIGH HIGH");
  lcd2.print("HIGH HIGH HIGH");
   delay(1000);
  lcd1.clear();
  lcd2.clear();
 }

 lcd1.print(":) :) :) :)");
 lcd2.print(":D :D :D :D");
 lcd1.clear();
 lcd2.clear();
   int reading = digitalRead(LockSwitch);

 // if (reading !=LastSwitchState) {
  //lastDebounceTime = millis();
    //if(){
  //}

}
```

Initial test code written to prototype the circuit

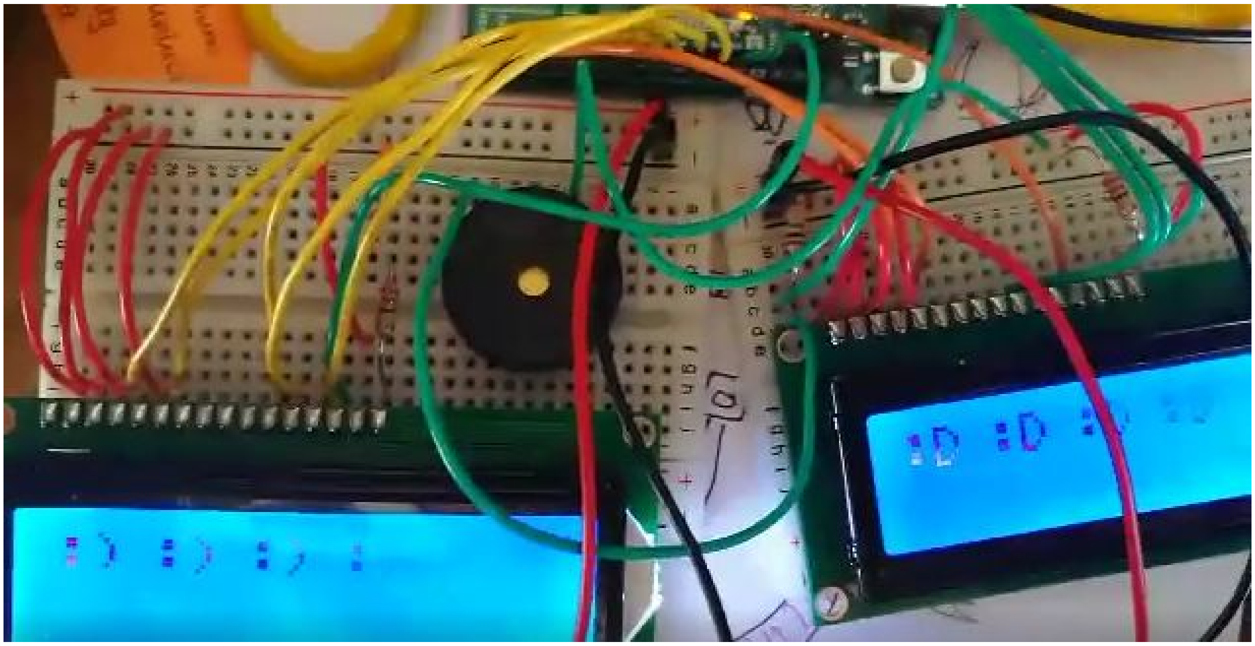
Materials:

1X breadboard power strip
5X 220 ohm Resistors
1X Arduino uno
1X 2 prong toggle switch
2X LCD screens 16x2 characters
1X 9V battery
1X HALJIA High Sensitivity Microphone
1X muit LED lamp
41X wires
1X roll of electrical tape (apply liberally)
1X laserbreak 360 phone VR headset

VR headset, Arduino Uno and 16X2 LCD screen

Prototyping:

With the plan set we proceeded to create the physical version of the circuit. The first problem we encountered was having to spread the LCD monitors across two breadboards. This was because the pin inputs were soldered on differently and have to be attached and rerouted into power source and pins into the Arduino Uno. The power was extended to both breadboards with wires from the kit, while the piezo had to be rewired completely to the second breadboard due to space restrictions. We found some wires which were not working and a lot of time was spent testing each wire. Below shows the initial setup, the right being the original breadboard, while the left is the extension of the first.



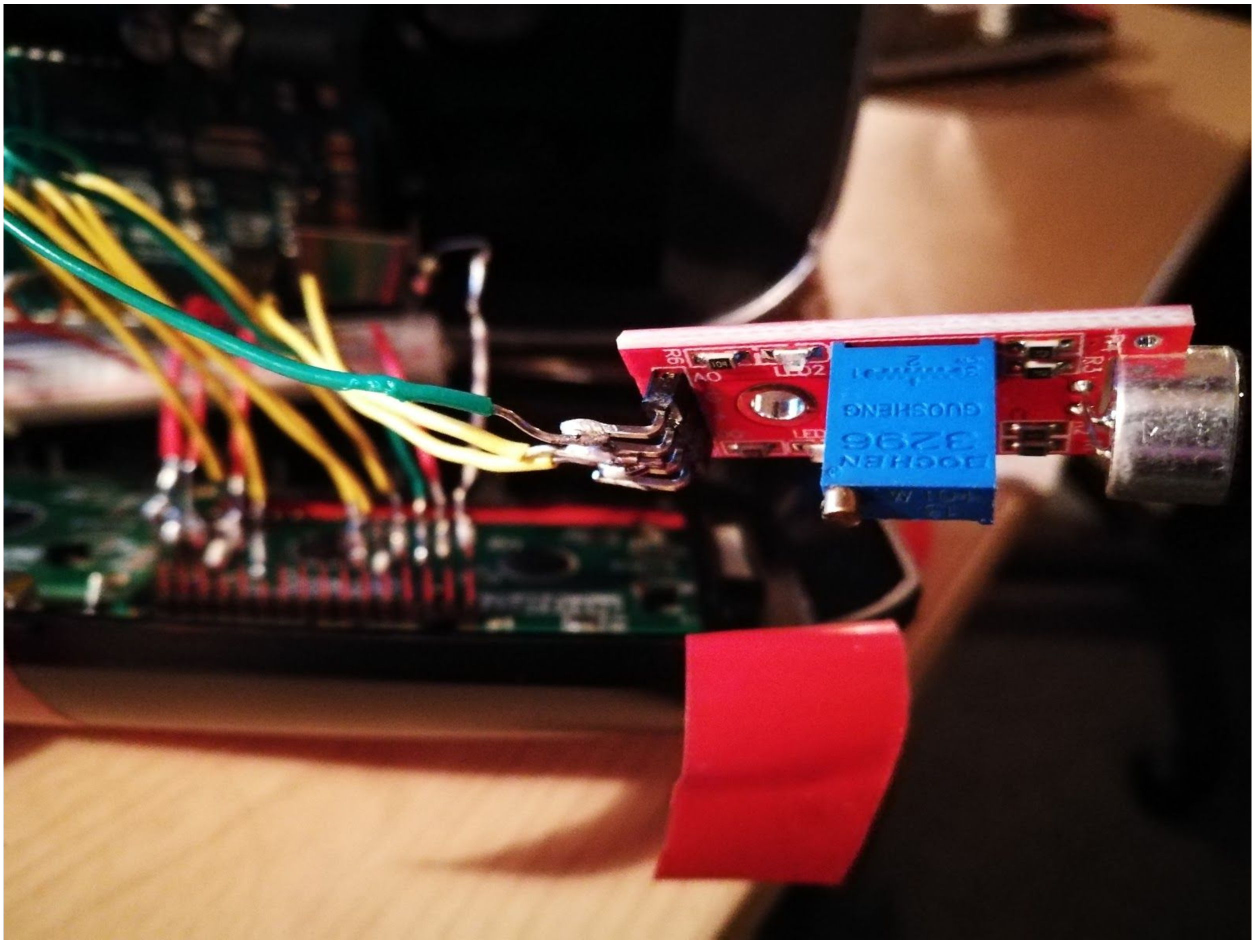
:D

At this point the headset was acquired and solved our enclosure problem, and we were eager to try things out with things put together with the electronics now completed and codes running.

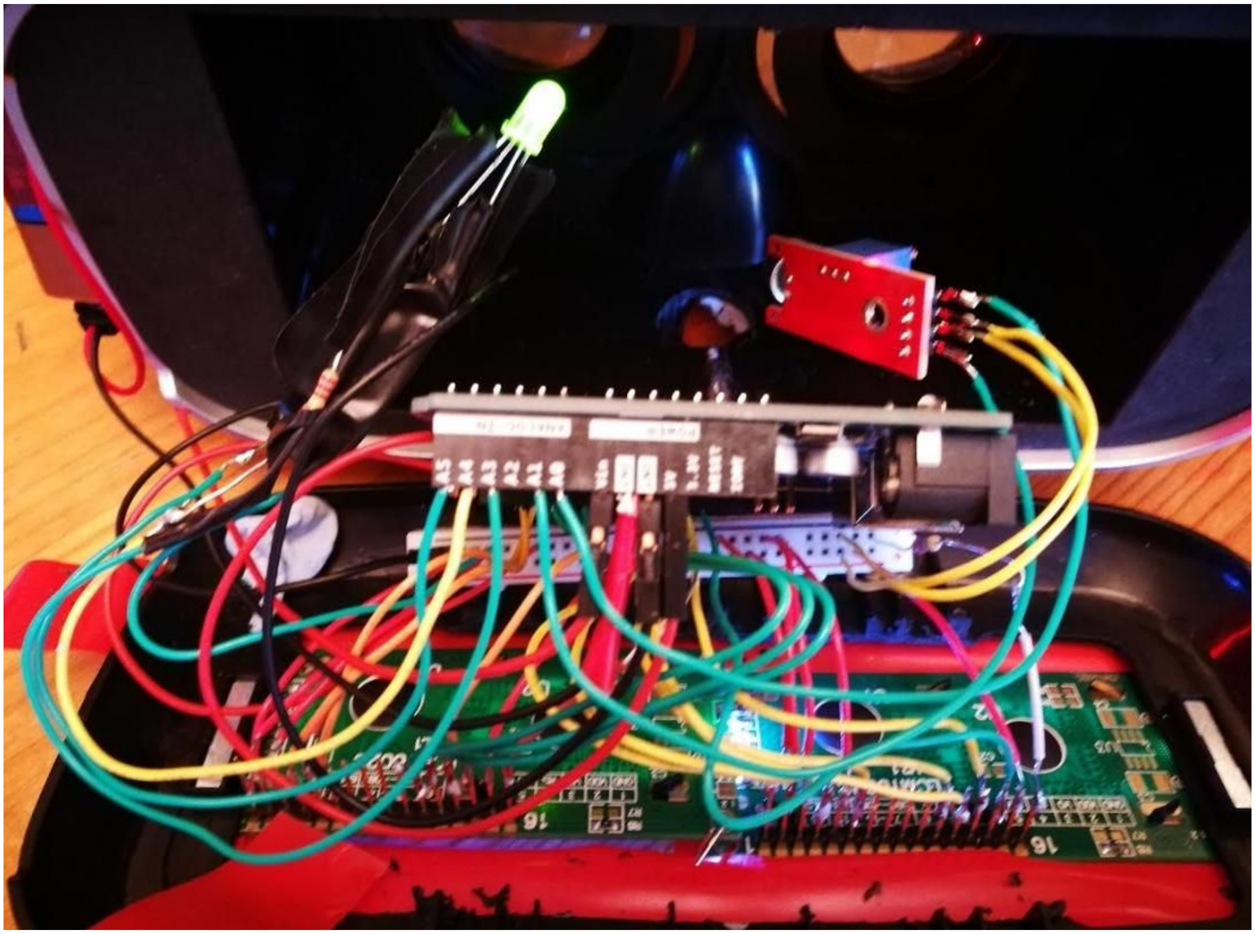
There’s a saying: if anything worked on the first attempt, it would be considered an unexpected gift. In our case this sense of caution was proven justified.

Stuffing the components into the headset was a challenge and we quickly realised that even though the circuit runs, the dimensions compared to the headset simply did not measure up. The solution involved soldering connection wires directly onto the LCD display pins; the longer ones reach the arduino, whilst the shorter ones connect to a power source via the breadboards. The number of breadboards was delightfully reduced to one after soldering, which further improved the living space for the components once they are more comfortably accommodated inside the headset. At this point we also created a much more dynamic solution to our power supply by using a 9V battery with its positive output going through a two pronged toggle switch. This gave us a much more realistic not to mention lighter solution for our power supply and also gave us our second input.

We also soldered in a multi coloured LED lamp that would change colour depending on what emotion was currently displayed on the screen. This helped to give the user wearing the goggles some feedback as to what was being displayed on the LCD screens. The code also required some small modifications for this to work but overall it was a very small change.



Soldered LCD display (faced down in the distance) connected to Arduino Uno, with additional new microphone chip in the foreground



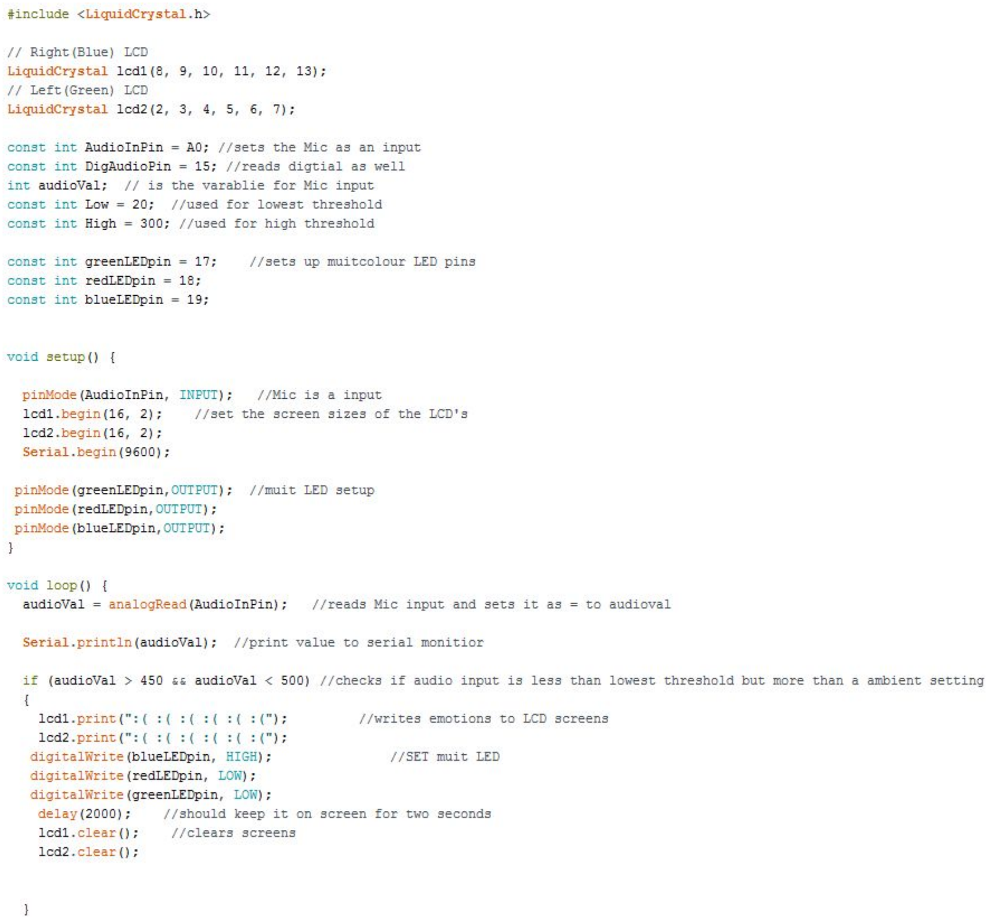
The finished circuit with multi LED Right, Microphone left the LCD screen and power strip bottom and our Arduino center

With soldering done and microphone swapped out to replace the piezo, the dimensions now are suitable for the headset. Now we had the problem of securing the screens and presenting them safely without worrying about components popping out. We realised the battery has to stay out with the switch to make changing power source easier, as well as save space for the wires inside. Our main problem was preventing the LCDs from escaping the headset enclosure. This was fixed with a healthy dose of practicality and red duct tape to secure the appropriate parts.

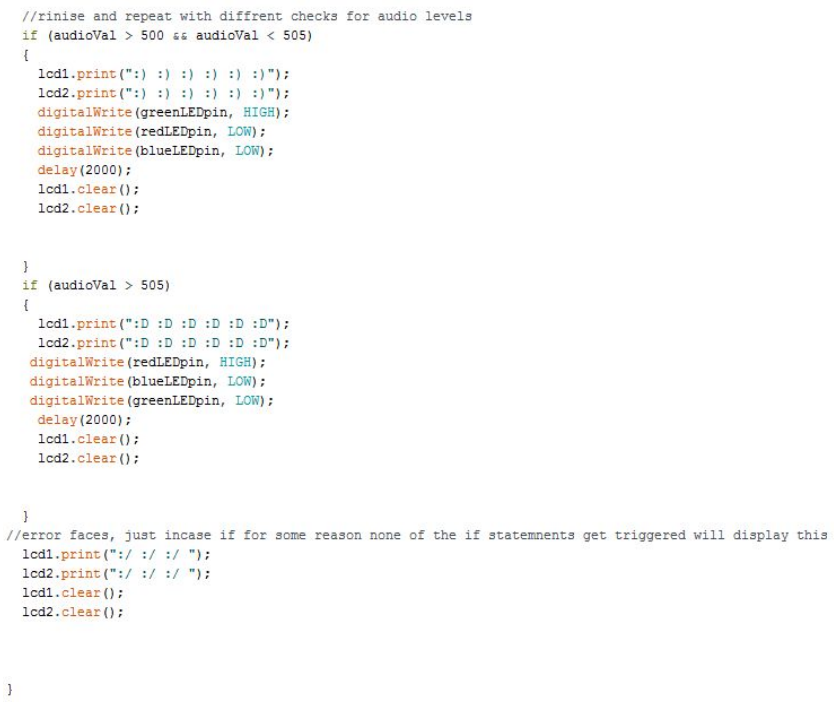
Code:

The final code to be loaded onto the Arduino did not change dramatically from the initial prototyping stage. The primary challenge was finding a sufficient sound range for the microphone that would work for our headset.Initially the microphone we used was very sensitive to loud noises but struggled to pick up quieter ones. As a result in the final code the range for the lowest value is much larger than the others this is so the lowest emotion states has as much of a chance as appearing as the other emotion states. Beyond that the core of the code are three if statements checking to see if the what value the microphone is and writing different emotions to the LCD’s for each state. The multi colour LED is also set at this time to tell the user what emotion is being displayed. That state is then held for 2 seconds(2000 milliseconds), this is so each change has enough time to be noticed and the screens do not just cycle rapidly without the viewers being able to react to a change. The screens are then cleared and the loop segment starts again.

In theory this procedure is very simple and not much can go wrong, there exist one know bug, this being that sometimes the microphone value won't be caught by the if statements (possibly because its to low). This display the error faces on the screens (:/ :/ :/, error faces) as a whole this does not break the goggles but is frustrating.

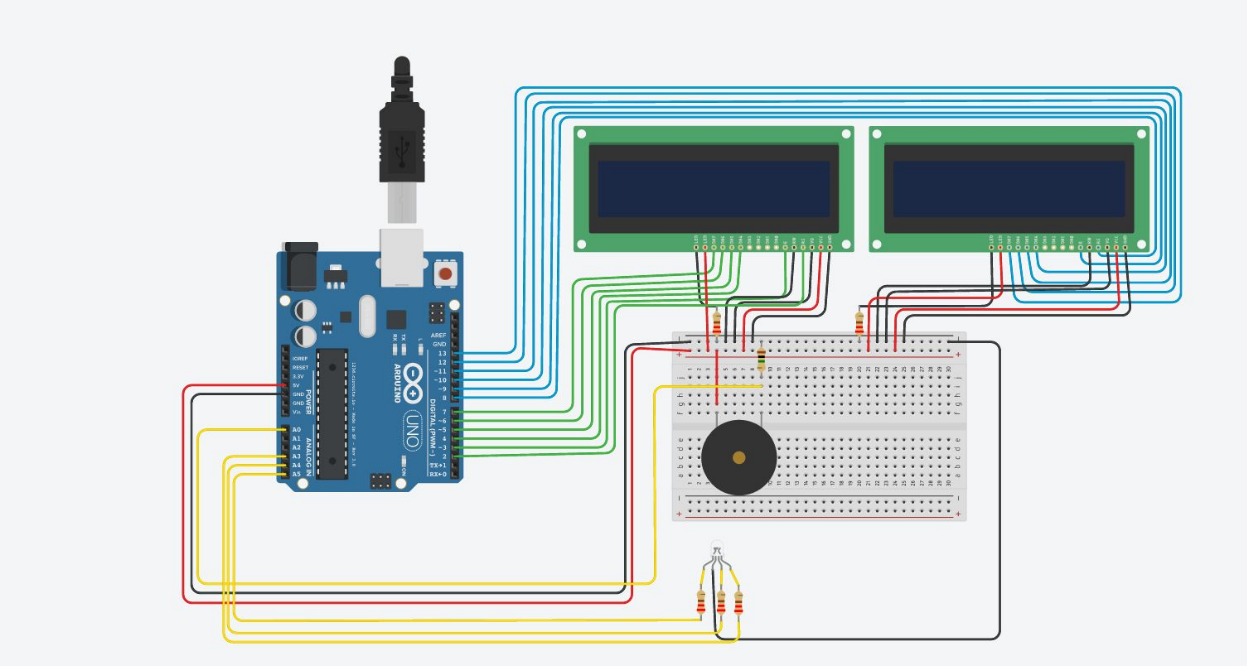


```
#include <LiquidCrystal.h>

// Right(Blue) LCD
LiquidCrystal lcd1(8, 9, 10, 11, 12, 13);
// Left(Green) LCD
LiquidCrystal lcd2(2, 3, 4, 5, 6, 7);

const int AudioInPin = A0; //sets the Mic as an input
const int DigAudioPin = 15; //reads digtial as well
int audioVal;  // is the varablie for Mic input
const int Low = 20;  //used for lowest threshold
const int High = 300; //used for high threshold

const int greenLEDpin = 17;    //sets up muitcolour LED pins
const int redLEDpin = 18;
const int blueLEDpin = 19;


void setup() {

  pinMode(AudioInPin, INPUT);   //Mic is a input
  lcd1.begin(16, 2);    //set the screen sizes of the LCD's
  lcd2.begin(16, 2);
  Serial.begin(9600);

 pinMode(greenLEDpin,OUTPUT);  //muit LED setup
 pinMode(redLEDpin,OUTPUT);
 pinMode(blueLEDpin,OUTPUT);
}

void loop() {
  audioVal = analogRead(AudioInPin);   //reads Mic input and sets it as = to audioval

  Serial.println(audioVal);  //print value to serial monitior

  if (audioVal > 450 && audioVal < 500) //checks if audio input is less than lowest threshold but more than a ambient setting
  {
    lcd1.print(":( :( :( :( :( :(");         //writes emotions to LCD screens
    lcd2.print(":( :( :( :( :( :(");
   digitalWrite(blueLEDpin, HIGH);               //SET muit LED
   digitalWrite(redLEDpin, LOW);
   digitalWrite(greenLEDpin, LOW);
    delay(2000);    //should keep it on screen for two seconds
    lcd1.clear();    //clears screens
    lcd2.clear();


  }
```

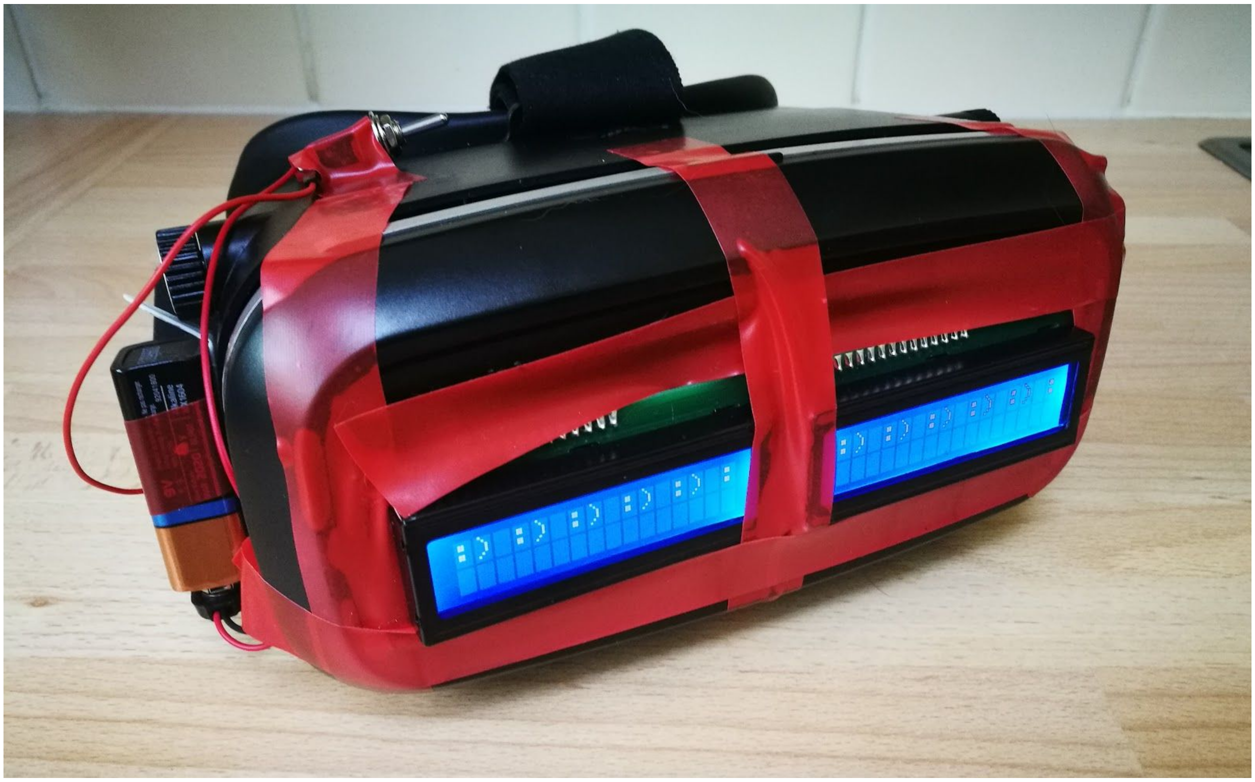


```
  //rinise and repeat with diffrent checks for audio levels
  if (audioVal > 500 && audioVal < 505)
  {
    lcd1.print(":) :) :) :) :) :)");
    lcd2.print(":) :) :) :) :) :)");
    digitalWrite(greenLEDpin, HIGH);
    digitalWrite(redLEDpin, LOW);
    digitalWrite(blueLEDpin, LOW);
    delay(2000);
    lcd1.clear();
    lcd2.clear();


  }
  if (audioVal > 505)
  {
    lcd1.print(":D :D :D :D :D :D");
    lcd2.print(":D :D :D :D :D :D");
   digitalWrite(redLEDpin, HIGH);
   digitalWrite(blueLEDpin, LOW);
   digitalWrite(greenLEDpin, LOW);
    delay(2000);
    lcd1.clear();
    lcd2.clear();


  }
//error faces, just incase if for some reason none of the if statemnents get triggered will display this
  lcd1.print(":/ :/ :/ ");
  lcd2.print(":/ :/ :/ ");
  lcd1.clear();
  lcd2.clear();



}
```

The final code for the project, not that different from the early code.



The final plan on circuits.io



Happy all round

Conclusions:

Michael:

I was pleased with how this project turned out, if I could do it again I would sim to rely less on tape to as a building component and would like to use some more advanced tools to fabricate the enclosure for the circuit and power supply. I would also like to use some more advanced LCD screens possibly to change the colour of displayed images and possibly add animation to the project. I do however feel a lot more confident in my understanding of electronics and associated code. I also feel that if i had to do the project again I would be able to be far more ambitious.

Rui:

The project turned out great and we produced a nice wearable tech with very low budget. With the right components upgraded we can definitely reduce the space and weight of our gear. The lesson to take from my part is that, if space is the problem for the project, the first thing to do should be minimising the amount of circuit boards/components required; else it will come back and bite us in the backside. I feel much more experienced in the hands-on parts of the project, particularly soldering and wiring up using the breadboard. If we had more time and resources we could possibly expand on the idea more and create a better enclosure and have more appropriately-placed symbols for where the eyes would be for the wearer to create a more complete experience.

Full source code available from github:
https://github.com/michaelhmt/LED_Glasses

Circuits.io project:
https://circuits.io/circuits/4249775-glasses