ME-GY 5913 Mechatronics

Course Project Report

Submitted in fulfilment of requirements of Course Project under guidance of Prof. Sanghoon Nathan Lee

By
Guangzhou Li N17078058
Yuanzhi Yao N19218661
Sagar Panchal N18051845

Table of Contents

Introduction

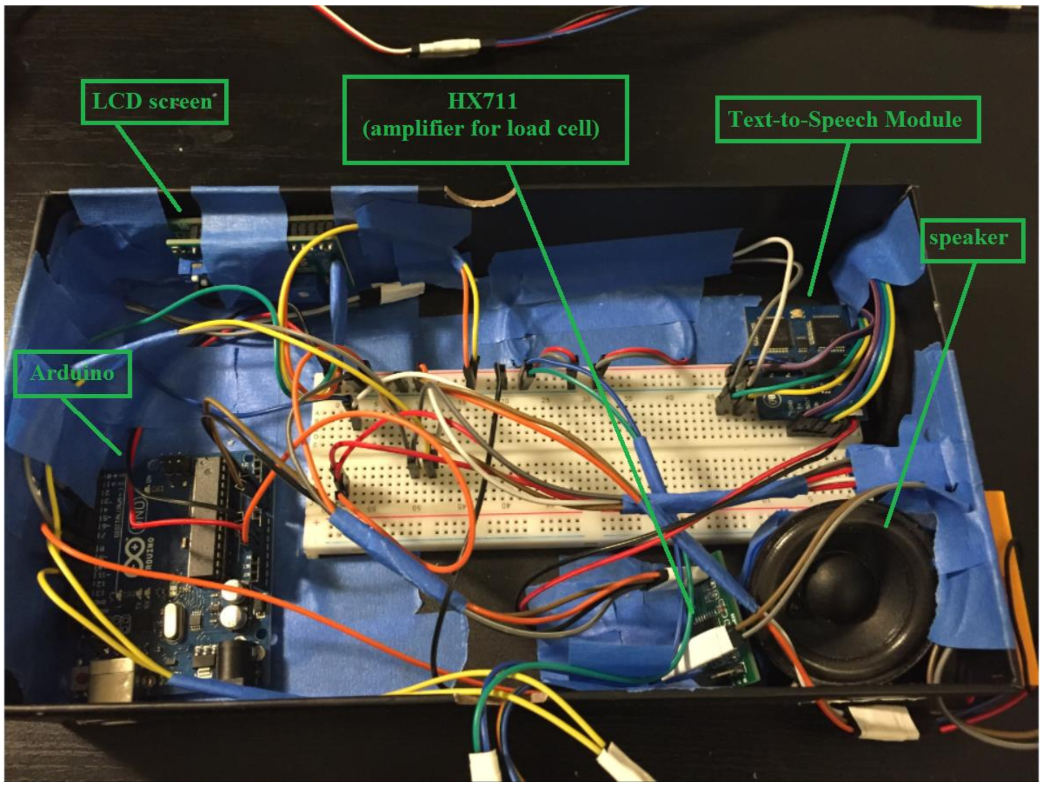
The body’s need of water is persistent. Around 60 percent of an average human body is made up of water. Three quarters of brain and heart, 83 percent of blood, and even the bones, seems to be dry and stiff, consists of 31 percent water. Consuming enough water every day is more important. Water helps to regulate the temperature of body, helps in nourishment of brain and spinal cord, and helps lubricating the bones.

Hydration is something to be taken as a serious topic, though we often didn’t realize how dehydrated we are. Nearly, two-third of the American population doesn’t consume enough water. Many of the Americans tend to guzzle coffee, tea, soft drinks or flavoured drinks. These beverages does not help us to go an extra mile, the way water can. In fact, caffeinated and alcoholic drinks are diuretics and promotes the water loss.

The amount of water needed daily, is the function of amount of energy we use. The water requirements depends on the body weight and daily activity. But considering an average scenario, men are recommended to drink 125 ounces, i.e. equal to 3.7 litres of water daily, and women are recommended to drink 91 ounces of water, i.e. equal to 2.7 litres daily.

Most of the people are not able to consume enough water, just because they are busy, doing their jobs. The solution is brought by the introduction of the Smart Mug. The Smart Mug is the device that keeps the track of total water consumption, alerts the user if he/she has not consumed enough water, for a long time, adds the total water consumption daily, and brings the total to match the total hydration needs of the user.

Components Used

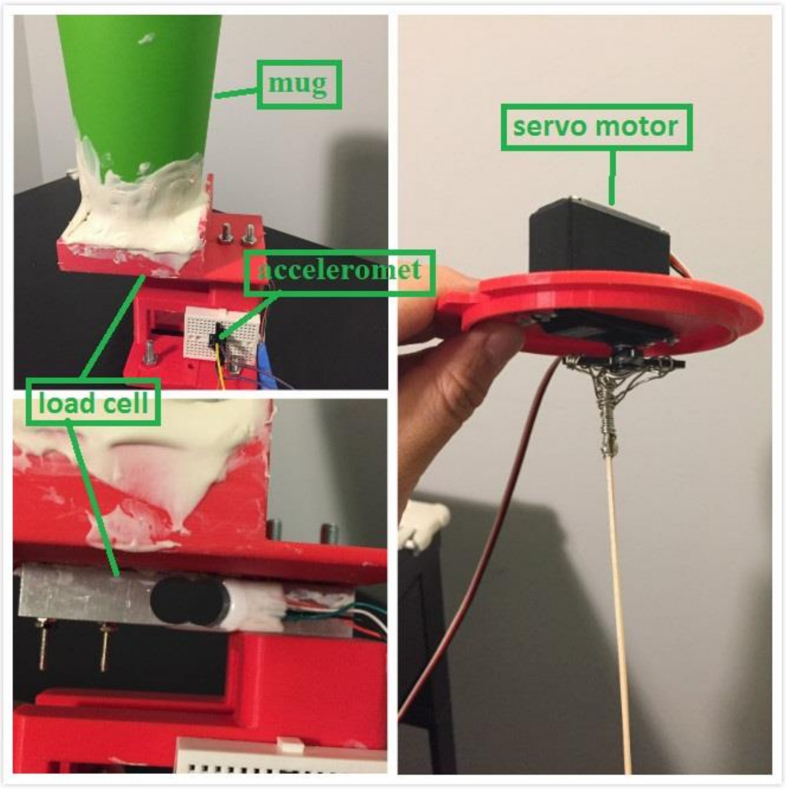


Arduino Microcontroller:

Arduino is an open source electronic prototyping platform that enables the users to create interactive electronic objects. In our project, we have used Arduino/Genuino Uno board, which is based on ATmega328P microcontroller. The board has 14 digital Input/output pins, which can be programmed, to use them as input or output. 6 of these digital input/output pins are allowed to be used for Pulse Width Modulation. The board also has 6 analog input pins. The board is operated on a 16 MHz quartz crystal. The Arduino board is capable of in system programming, after the installation, and has a ICSP header on the board. Arduino is one of the smart devices that can be used to develop standalone interactive devices.

Load Cell:

A Load Cell is one of the easy way to measure the weight of the Mug. The Load Cell is also easy to implement on the Arduino Board. The Load cell gives us the signal which are converted in the form that the Arduino Board is able to understand and then the correct weight is gained from some calculations and conversion into the ounces of water.

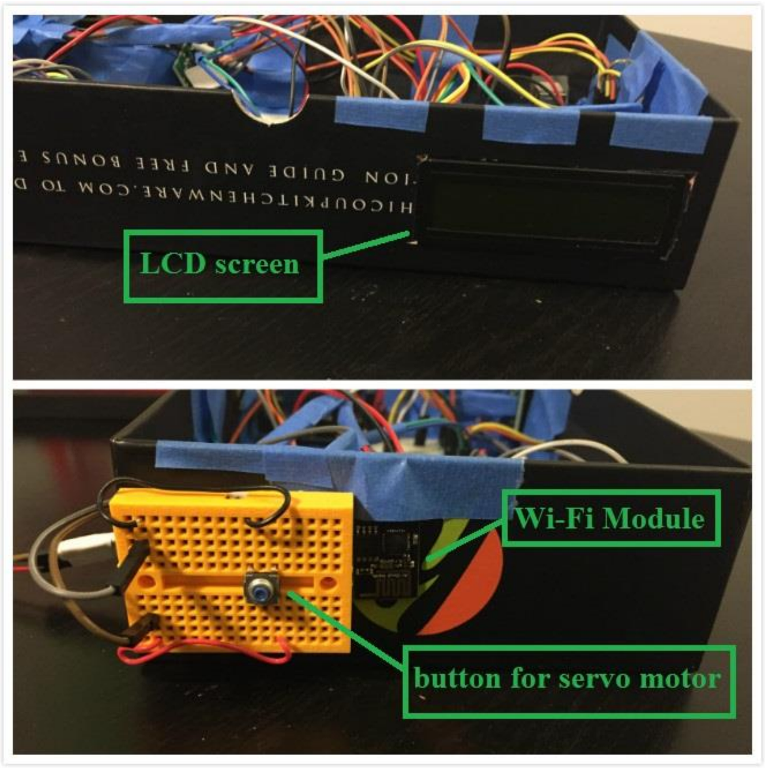


Wi–Fi Module:

We want the system to communicate the user, to alert about the daily consumption of water. We achieved this by using a Wi-Fi module, ESP8266. The ESP8266 module is attached to the system which can connect to any other device over a Wi-Fi network. We can then send the alert messages from the system to the remote devices, so that the user at a remote location can be alerted by the system.

LCD display:

An LCD display is used to show the messages to the user. We have predefined all the messages in the code itself. The messages to be shown to the user may include the information regarding the current hydration level, required hydration, water level in the glass, etc. More information can be added to be shown on the LCD, and would be the matter of further development in the system.



Text-to-Speech Module:

The name of the Text-to-Speech Module is "Emic-2". A special module is connected to the system, so that the user can be alerted in a unique way. Emic-2 is a text to speech module that converts the written text into the audible sound. A speaker is connected to the system so that it can alert the user in the form of audible sound.

Coffee Stirrer:

We have added an additional functionality to the Smart Mug. A Coffee stirrer is operated on the servomotor. A simple mechanism of pushbutton press is used to make the stirrer operable.

Techniques used:

Load Cell:

The Load cell is used to measure the weight of the water inside the mug. The load cell has four terminals, two of which are used to excite the load cell. The load cell is excited at 5 volts of supply. The other two terminals of the load cell gives the analog signal, which is amplified by using the Load Cell Amplifier, HX711. The HX711 gives the number of counts to the Arduino, and depending on the number of counts, and the capacity of the load cell, the calibration of the load cell is done and then based on the calculation, the load cell signal is converted into the correct number of ounces of water.

Remote Connection:

We have implemented the utilization of a WiFi module, to establish the connection between the system and the user. The user can be alerted and reminded about the water consumption via a message on the Cell Phone. Messages on any remote device can be done, satisfying the condition that the device supports the WiFi feature and is kept connected to the system.

Water Consumption:

According to the calculations, an average person need to consume 8 ounces of water every hour while awake. This will keep the body processes to work finely. To keep this track of water consumption, we have implemented a timer that counts the time, starting at zero, every time when the Mug is empty. If the Mug is kept empty for an hour, the alert message is generated to notify the user that he/she requires water as it has been a long time that the user had water.

Correctness in measurement:

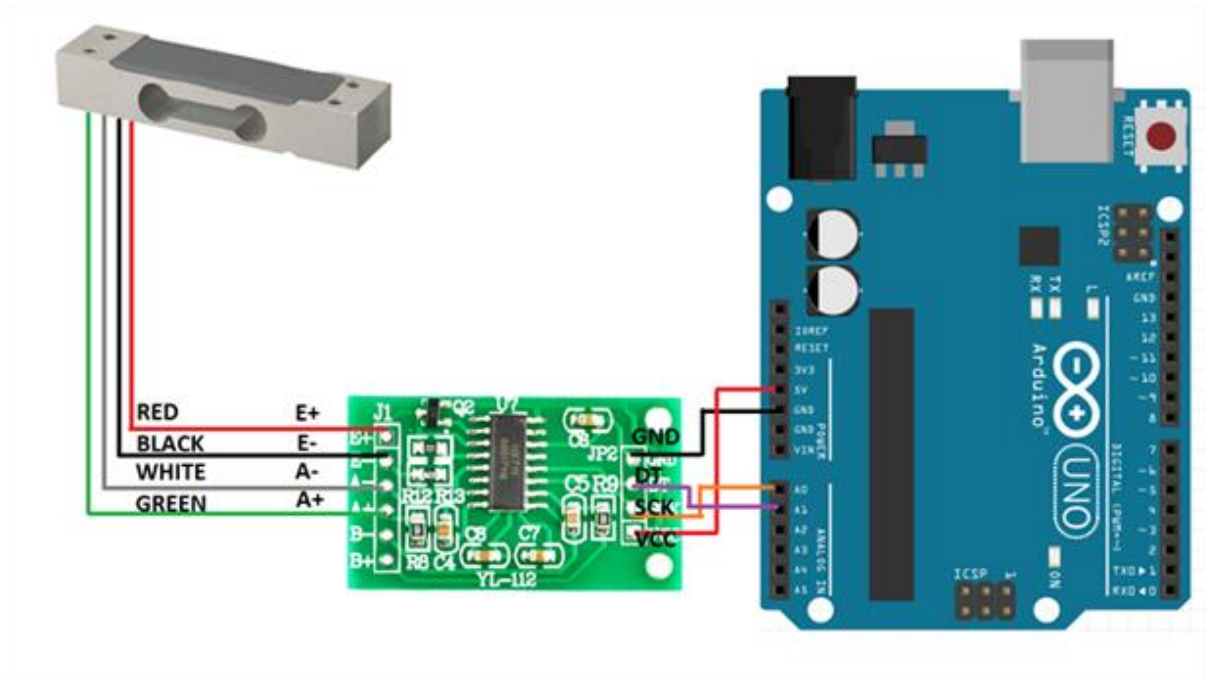
We obviously don’t want to measure an erroneous data regarding the amount of water in the mug. So we have placed a conditional statement before measuring the weight of the mug. We have used an accelerometer, to make sure that the weight is calculated only if the Mug is kept stationary on a stable surface. The accelerometer gives the signals to the Arduino in the form of X and Y axis. If the condition is satisfied, and the values of the x and y variables (defined respectively for X and Y axis) are within the satisfying limits, then and then only the load cell is allowed to measure the weight.

Text-to-Speech Module:

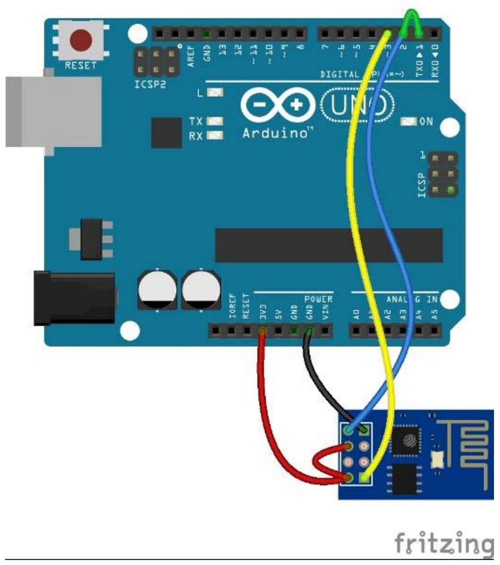
A special module is added to the system that converts the written text to the audible speech. This requires the message to be sounded to the user, to be typed-in the Arduino Code. We can sound the messages that greets the user, reminds the user to drink water, or any other functionality can be added as required.

Hardware Setup:

The Hardware setup section shows the physical setup of all components, individually. Note that the hardware setup is subject to change on the practical design. Firstly, have a look at the image showing the connection of Load Cell to the Arduino. The Load Cell sends the signals in analog form and to make them readable by the Arduino, we use the Load Cell Amplifier, HX711. The connection of the Load Cell circuit to the Arduino is shown below.

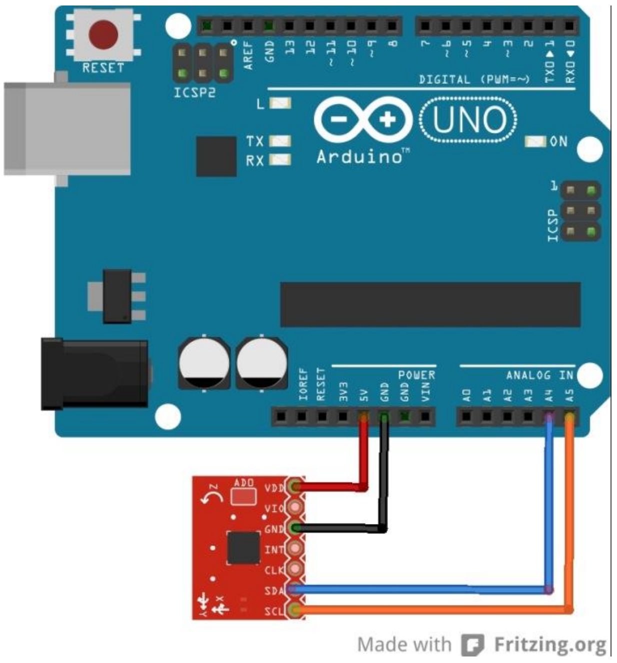


The 5V to the amplifier is provided from the Arduino itself, so there is no need to use an external power supply to excite the amplifier. The WiFi module, which is used to establish the connection between the system and the remote user, is connected as shown below in the picture.

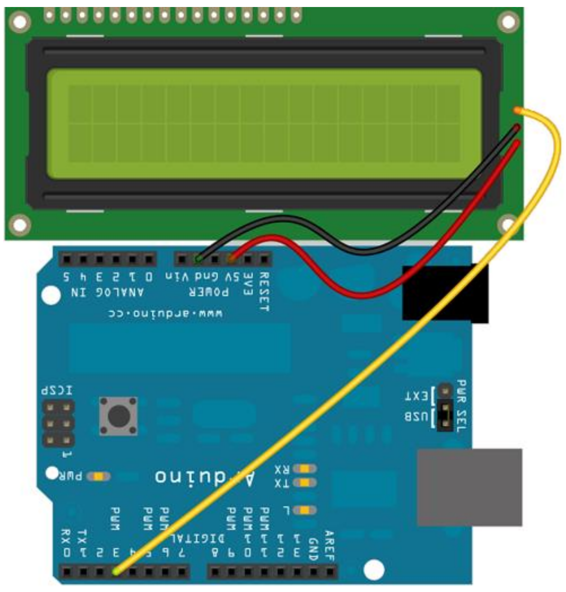


The 3.3V is required for the WiFi module to get powered up. These 3.3V are provided from one of the pin on the Arduino Board itself. The communication is between the WiFi device and the Arduino is carried out by implementing the serial communication.

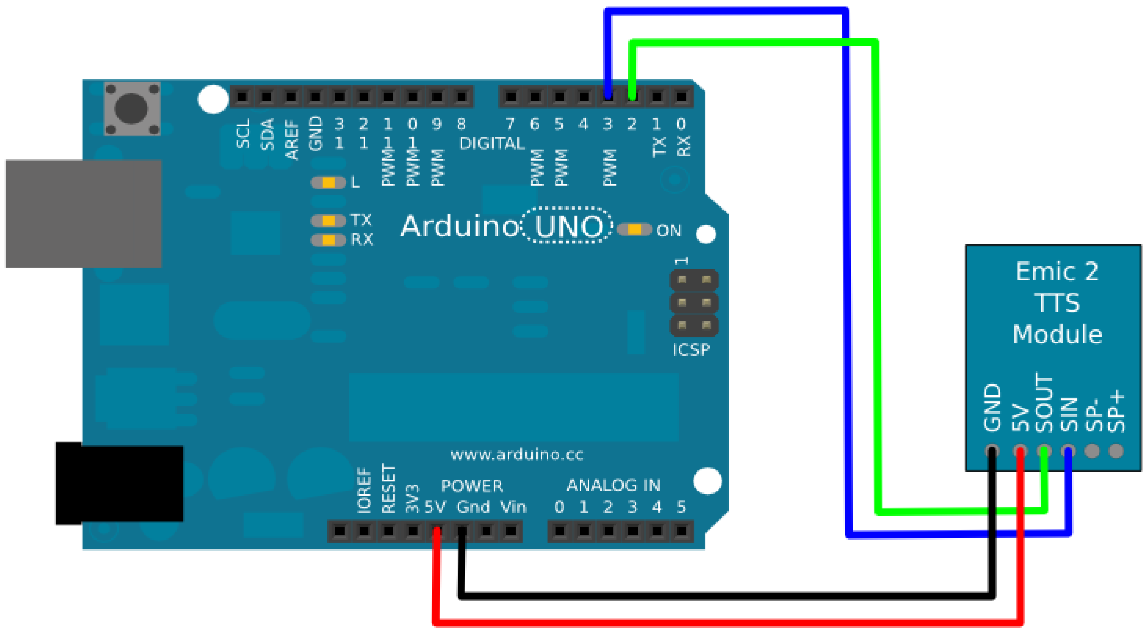
Further, the connection for the Accelerometer is shown in the image below. The 5V required by the accelerometer is provided from the Arduino Board. The Analog inputs are provided from the Accelerometer, and the X-axis and Y-axis values are used to decide whether the Mug is stable or not.



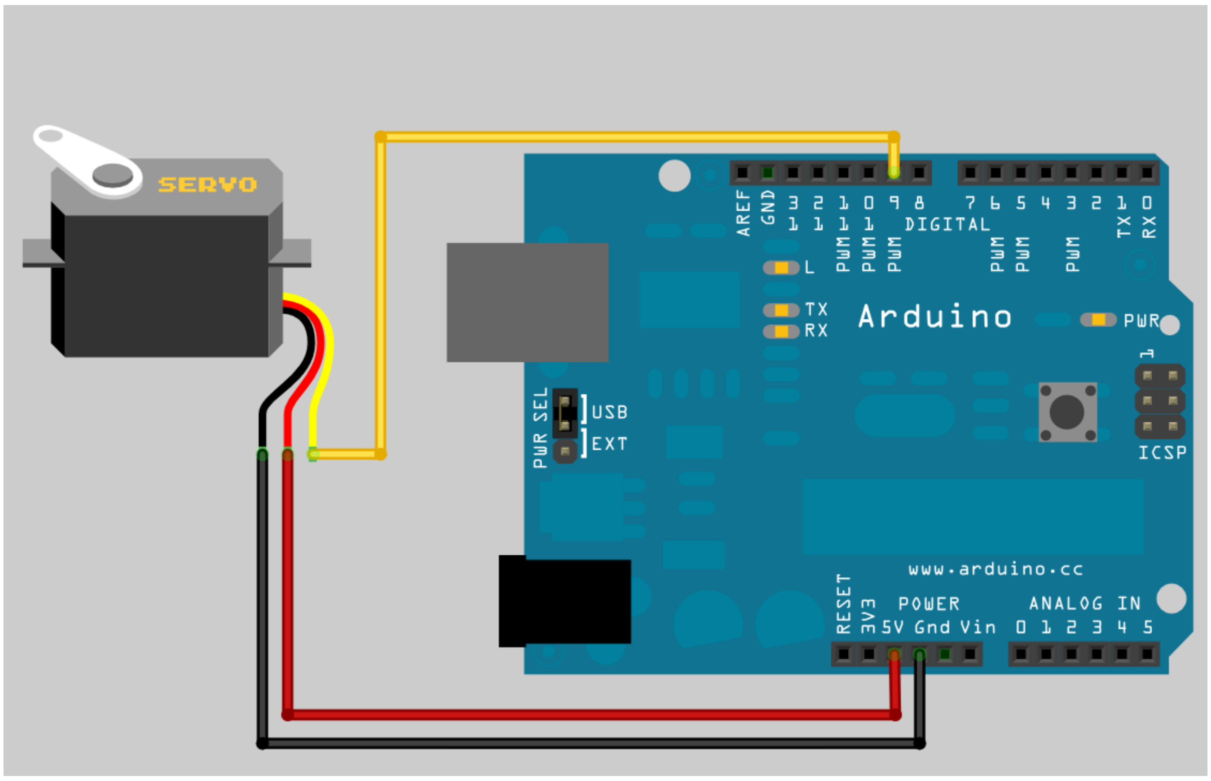
The LCD display is made to communicate with the Arduino Board using serial communication. The simple communication is shown below. The Rx from the LCD should be connected to the Tx from the Arduino Board.



Emic-2 is the Text to Speech module that is used to send the audible messages to the user. The connection of Emic-2 with the Arduino Board is shown below. Also a speaker is made to connect to the Arduino board to sound the messahes loudly.



The Coffee Stirrer is made out of the use of Servo Motor. The Connection are shown below. The speed of the motor is hard coded in the Arduino Code.



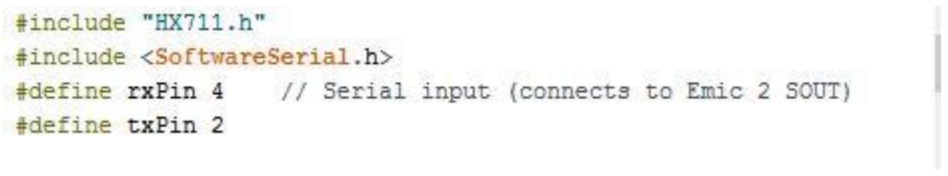
Arduino Software Implementation:

The Arduino language is merely a set of C/C++ functions that can be called from the code. The code is passed directly to C/C++ compiler. A minimal Arduino sketch consists of only two functions:

setup(): This function is called always, once when power is turned on or on the rest of the controller. It is used to initialize the libraries, variables and pin modes in the design.

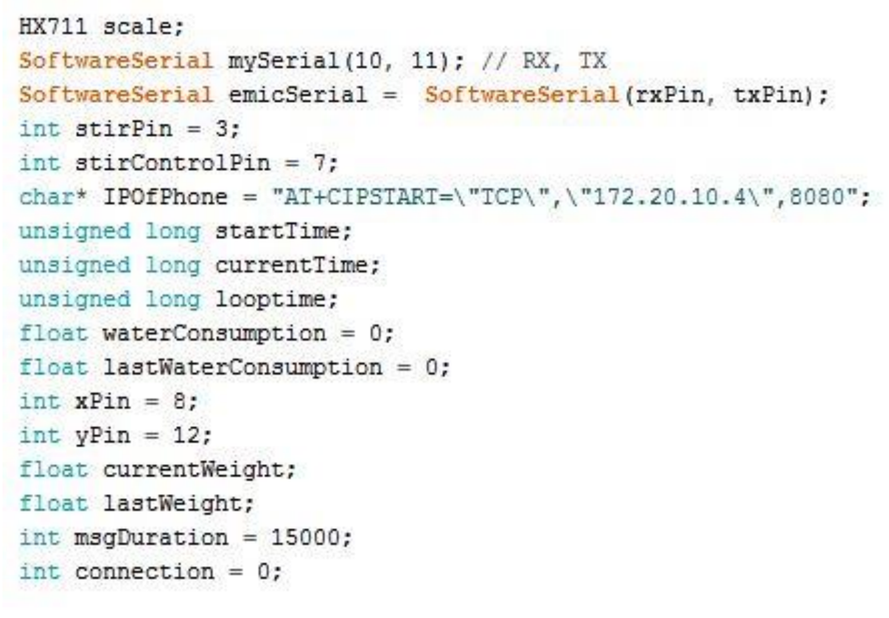
loop(): After setup() is called, the loop() function is repeatedly called. Whatever written in the loop() function, is executed on the repeatedly, until the board is turned off or reset is pressed.

Code:



```
#include "HX711.h"
#include <SoftwareSerial.h>
#define rxPin 4    // Serial input (connects to Emic 2 SOUT)
#define txPin 2
```

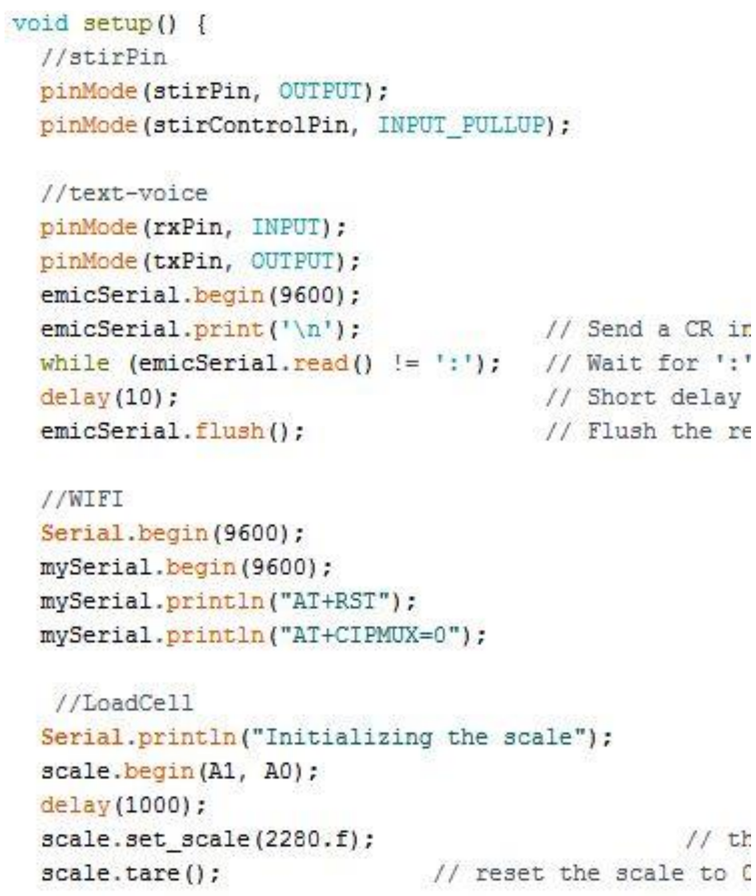
The code begins with defining the libraries. "HX711.h" is an Arduino library to interface ADC, used for load cell. SoftwareSerial.h is the header files, defined so that we can other digital pins for serial communication. Note that the Arduino Uno board already has pins 0 and 1 defined for serial communication. But if we want to use other pins also for serial communication, then we can define them as shown above. We just need to use the header file SoftwareSerial.h.



```
HX711 scale;
SoftwareSerial mySerial(10, 11); // RX, TX
SoftwareSerial emicSerial =  SoftwareSerial(rxPin, txPin);
int stirPin = 3;
int stirControlPin = 7;
char* IPOfPhone = "AT+CIPSTART=\"TCP\",\"172.20.10.4\",8080";
unsigned long startTime;
unsigned long currentTime;
unsigned long looptime;
float waterConsumption = 0;
float lastWaterConsumption = 0;
int xPin = 8;
int yPin = 12;
float currentWeight;
float lastWeight;
int msgDuration = 15000;
int connection = 0;
```

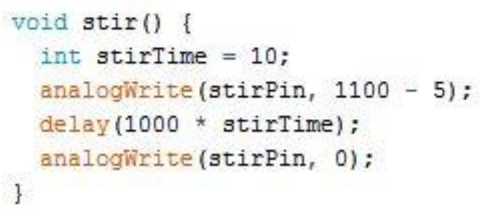
After defining the header files, there comes the turn to define the variables. The variables are of 2 kinds:

Local Variables are those whose scope is limited to a particular function only. While Global variables are those which are defined outside all the functions. Once a variable is defined globally, we can then use them at any part of the code. In the image above, we can see that we have defined several variables globally, so that they can be used inside any function.



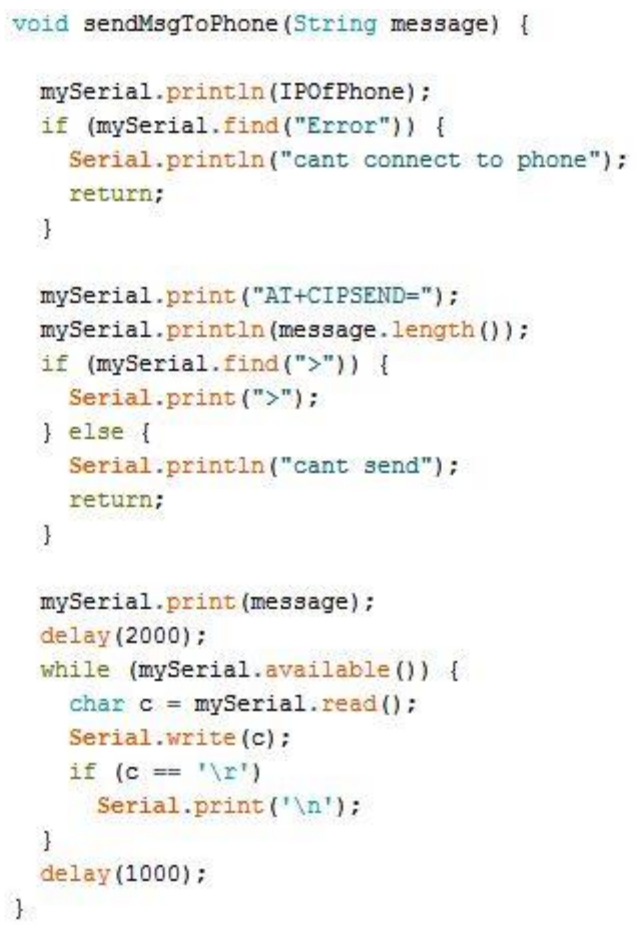
```
void setup() {
  //stirPin
  pinMode(stirPin, OUTPUT);
  pinMode(stirControlPin, INPUT_PULLUP);

  //text-voice
  pinMode(rxPin, INPUT);
  pinMode(txPin, OUTPUT);
  emicSerial.begin(9600);
  emicSerial.print('\n');              // Send a CR ir
  while (emicSerial.read() != ':');    // Wait for ':'
  delay(10);                           // Short delay
  emicSerial.flush();                  // Flush the re

  //WIFI
  Serial.begin(9600);
  mySerial.begin(9600);
  mySerial.println("AT+RST");
  mySerial.println("AT+CIPMUX=0");

   //LoadCell
  Serial.println("Initializing the scale");
  scale.begin(A1, A0);
  delay(1000);
  scale.set_scale(2280.f);                       // th
  scale.tare();               // reset the scale to 0
```

The setup function is always called once when the board is being started up for the first time. The setup function is used to initialize the pin modes, libraries or to establish the serial communications. In our setup function, we have defined the modes of pins for stir function, emic sensor, setting scale of the load cell. We also have used two serial communication, one for emic and for WiFi module, which are initialized in the setup function.



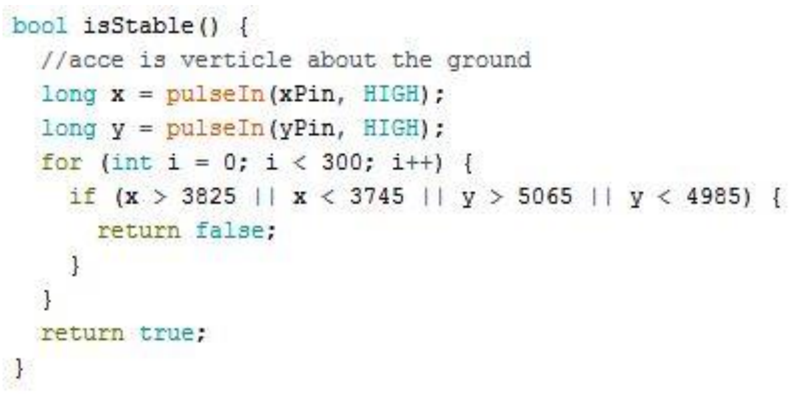
```
void stir() {
  int stirTime = 10;
  analogWrite(stirPin, 1100 - 5);
  delay(1000 * stirTime);
  analogWrite(stirPin, 0);
}
```

We have implemented an additional feature in the Mug known as Coffee Stirrer. A servo motor is used for this purpose. We can see the function stir() for the above purpose. The speed of the motor is controlled by the above code. Note that we have used the function analogWrite() function inside the stir() function to write the desired speed of the motor.



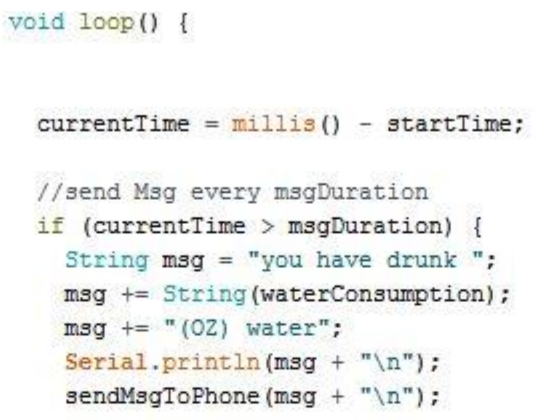
```
void sendMsgToPhone(String message) {

  mySerial.println(IPOfPhone);
  if (mySerial.find("Error")) {
    Serial.println("cant connect to phone");
    return;
  }

  mySerial.print("AT+CIPSEND=");
  mySerial.println(message.length());
  if (mySerial.find(">")) {
    Serial.print(">");
  } else {
    Serial.println("cant send");
    return;
  }

  mySerial.print(message);
  delay(2000);
  while (mySerial.available()) {
    char c = mySerial.read();
    Serial.write(c);
    if (c == '\r')
      Serial.print('\n');
  }
  delay(1000);
}
```

Another additional feature that we have implemented is the use of WiFi. A WiFi module is used to make the communication between the system and the user at remote location. We are able to send the desired messages to the user on the cell phone. The function sendMsgToPhone() has a single argument in the form of a string. Whenever the function is called, the message to be sent to the user is typed in as the argument to the function. We have called the function inside the loop() function which will be shown ahead in the report.



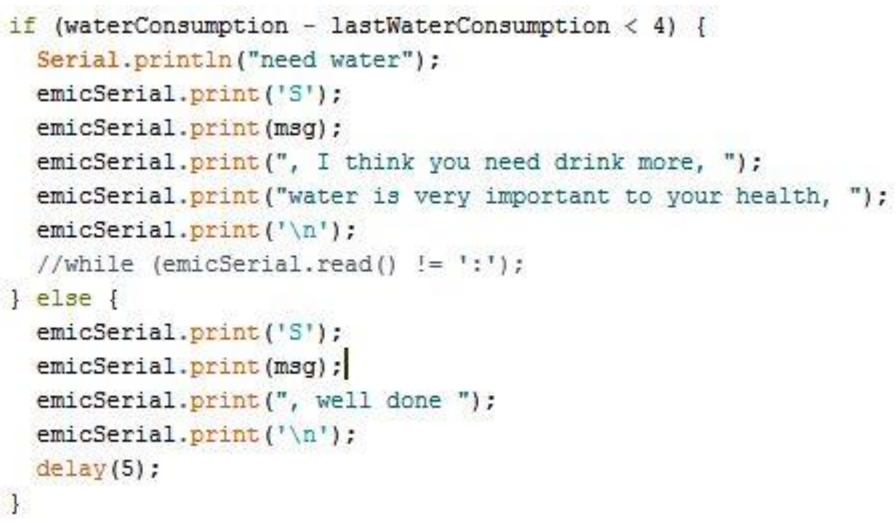
```
bool isStable() {
  //acce is verticle about the ground
  long x = pulseIn(xPin, HIGH);
  long y = pulseIn(yPin, HIGH);
  for (int i = 0; i < 300; i++) {
    if (x > 3825 || x < 3745 || y > 5065 || y < 4985) {
      return false;
    }
  }
  return true;
}
```

Another additional function that we implemented is the use of an accelerometer. It is obvious we don’t want to measure an erroneous number as amount of water inside the mug. So we have placed an accelerometer is placed at the bottom of the mug so that we may know the how tilt the mug is. We have placed the condition using if() condition based on the calculation. If he values of the variables x and y (defined for X and Y axis respectively) satisfies the condition, then and then only, the weight of the Mug is measured.



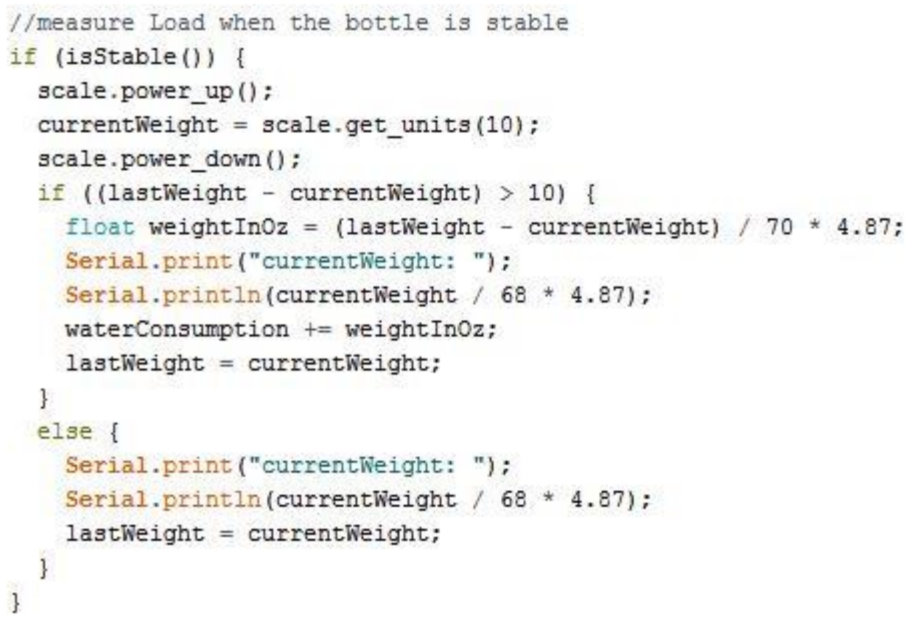
```
void loop() {


  currentTime = millis() - startTime;

  //send Msg every msgDuration
  if (currentTime > msgDuration) {
    String msg = "you have drunk ";
    msg += String(waterConsumption);
    msg += "(OZ) water";
    Serial.println(msg + "\n");
    sendMsgToPhone(msg + "\n");
```

The loop() function is the main function of the Arduino code. Everything written inside the loop() function us called over and over in an infinite loop. All the working principles of the system are written inside the loop() function. Our loop() function starts with the condition of the current time. The variable currentTime is compared to the variable msgDuration, and according to the if condition, a message is displayed on the screen saying how much water did the user drink till now. Also the same message is send to the cell phone of the user.



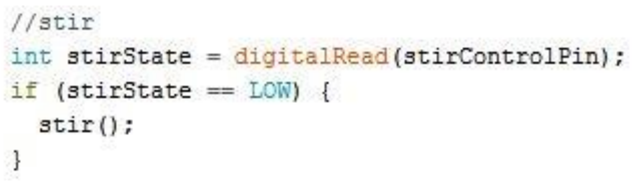
```
if (waterConsumption - lastWaterConsumption < 4) {
  Serial.println("need water");
  emicSerial.print('S');
  emicSerial.print(msg);
  emicSerial.print(", I think you need drink more, ");
  emicSerial.print("water is very important to your health, ");
  emicSerial.print('\n');
  //while (emicSerial.read() != ':');
} else {
  emicSerial.print('S');
  emicSerial.print(msg);
  emicSerial.print(", well done ");
  emicSerial.print('\n');
  delay(5);
}
```

Then, we have compared the water consumption. Using a simple if…else loop, we have placed the condition that if the user doesn't drink enough water, he/she will be reminded automatically by the system. The use of emic text-to-speech converter is implemented here. The user will be alerted by the sound that will call whatever is written as a text in the code.



```
//measure Load when the bottle is stable
if (isStable()) {
  scale.power_up();
  currentWeight = scale.get_units(10);
  scale.power_down();
  if ((lastWeight - currentWeight) > 10) {
    float weightInOz = (lastWeight - currentWeight) / 70 * 4.87;
    Serial.print("currentWeight: ");
    Serial.println(currentWeight / 68 * 4.87);
    waterConsumption += weightInOz;
    lastWeight = currentWeight;
  }
  else {
    Serial.print("currentWeight: ");
    Serial.println(currentWeight / 68 * 4.87);
    lastWeight = currentWeight;
  }
}
```

Further, we always want to measure the amount of water if and only if the mug is kept stable on a stable surface. So, keeping the condition, we have used if…else condition. All the conversions from the load cell signals to the ounces of water are done inside this loop. Based on the calculations, we came up with the number to show the correct weight inside the Mug.



```
//stir
int stirState = digitalRead(stirControlPin);
if (stirState == LOW) {
  stir();
}
```

Finally, we have made a call to the stir function. The Coffee Stirrer is an additional function that is implemented using a servo motor and is controlled using stir() function.

Future Improvements:

Several improvements are possible for the system design of the Smart Mug. Firstly consider the options to know the water level inside the mug. We can use multiple load cells to measure the weight of the water, so that we can measure the water even if the mug is flipped over. We can also have the option to use the capacitive level measurement inside the mug. This would be a miniature and easy to use design. We can directly measure the level of water just by measuring the change in the capacitance. This design can also have some of the problems like each different fluid may have different dielectric. Example, the dielectric of a coffee is different from that of water. Also, the change in temperature of the fluid can affect the dielectric of the fluid. Also, an ultrasonic sensor can be used to measure the water level. This will further creates the need of mapping the variable to convert distance into the water amount. This mapping would be different for different sized and shaped mugs. Also, the placement of the ultrasonic sensor would be a topic of debate. Example, if we keep the sensor on the lid of the mug, then it may create the problem of un-relying signals, if the lid is kept open, or placed somewhere else.

Further, we have hard coded the messages to be shown to the user on the LCD or send to the user via a WiFi network. We can provide a keypad to enter the messages to be shown to the user. We can type-in our own messages, greetings, etc. Also, the inclusion of a keypad can give us more freedom, to choose the how frequently the user want the alert messages, personal hydration needs depending on the requirement of the user, etc.

There are many such points that can be thought for the future enhancements, working on those points, can take the design to an extra-ordinary level.