University of Moratuwa

Faculty of Engineering

Department of Electronic and Telecommunication Engineering

Project report

GasCheck - Gas Cylinder Level Indicator

Group EN-17

Group members:

200439G	**Gunatilake P.T.B**
200445V	**Pasqual A.C**
200449L	**Pathirana K.P.T.R**
200455C	**Perera G.L.S.M**

7th October 2022

Table of contents

The Problem 4

Product Idea Validation 5

Product Description 7

Technical Specifications 8

Product Architecture 9
- Load Cells 9
- Load Cell Amplifier 10
- Microcontroller 10
- Bluetooth Module 11
- Mobile App 11
- Power Supply 12
- Alternative Approaches to the Architecture 12

Technical Feasibility 13
- Accuracy and Measurement Resolution 13
- Power Supply 13
- Power Consumption 13
- Effect of Load Cell Creep 14
- Effect of Shock Loads 14
- Hardware and Software Requirements 15

Enclosure Design 16
- Sketches of Initial Ideas 16
- Final Sketch 16
- Material, Color and Finish 17
- Solidworks Design for the Enclosure 18
- Enclosure of Prototype (CNC machined wood) 20

User Interface Design 21
- Weighing Scale 21
- Mobile App 21
- Prototype of Mobile App 22

Functional Prototype 24
- Implementation using an Arduino board 24
- Implementation using a microcontroller on a breadboard 25
- Results 25

PCB Design **26**
Schematic Diagram 26
PCB Layout Diagram 27
Manufactured PCB 27

Discussion **28**

Project Budget **29**

Marketing, Sales and Beyond **30**
Marketing 30
Sales 30
Product Packaging 31
Maintenance and Repair 31
Reuse of Components 31

Task allocation among the group members **32**

Appendix **33**
References 33
Arduino code 33

The Problem

In Sri Lankan households, LPG (Liquefied Petroleum Gas) is one of the most common fuels used for cooking. But in the current economic crisis, purchasing gas has become a difficulty in many households. Because of that, it has become a necessity to carefully manage the usage of gas and make it last longer. However, in the normal manner of using gas cylinders, conservation can only be done by guesswork as there is no way of knowing how much gas has been used until the cylinder runs out of gas. Overcoming this limitation could be useful for many households in the country.

Another common problem that stems from the same limitation (which is relevant even without a crisis) is unexpectedly running out of gas before a new one could be purchased, causing waste of time and money.

As going by guesswork is not satisfactory, we were motivated to design a device which can detect the remaining gas level in the cylinder to a certain precision. It will benefit all people who use LPG for cooking, especially at this time when conserving gas is highly necessary.

Product Idea Validation

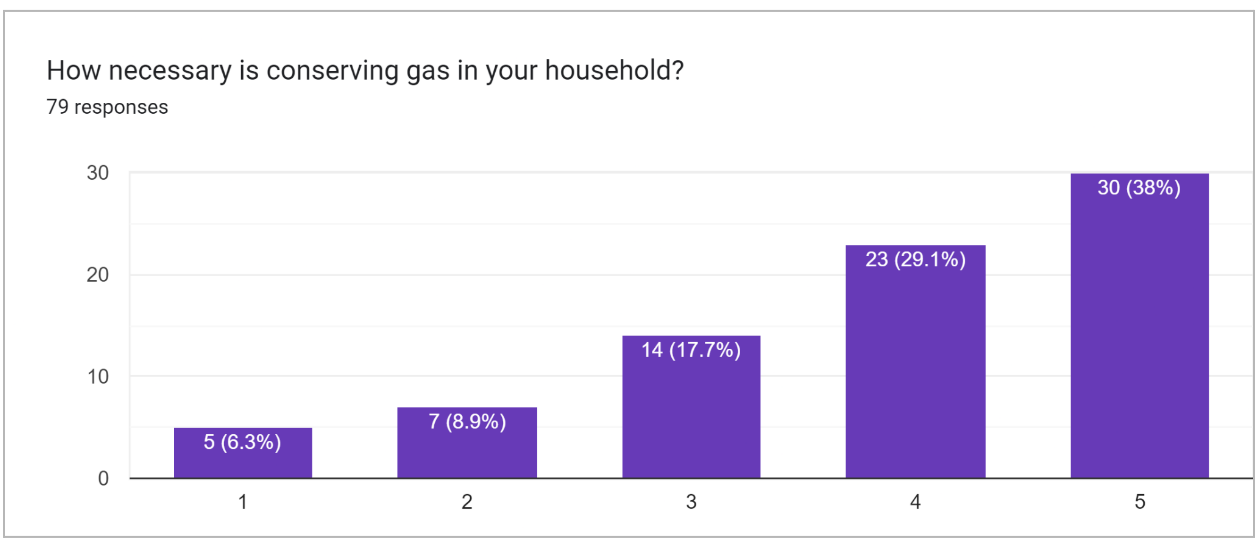
Because of the shortage of LPG supply in the country, people are now more focused on conserving gas as much as possible. Therefore, the lack of a way to know the gas level is a more significant problem than before.

When there is a way to know the amount of gas in the cylinder, it is possible to monitor how much gas was used for a particular task, and this will enable the user to optimally allocate gas amounts for various tasks. Also, knowing that the gas level is low will subconsciously allow the user to efficiently use the remaining amount of gas. Hence, a product that enables this will be beneficial for people to handle the above mentioned problems.

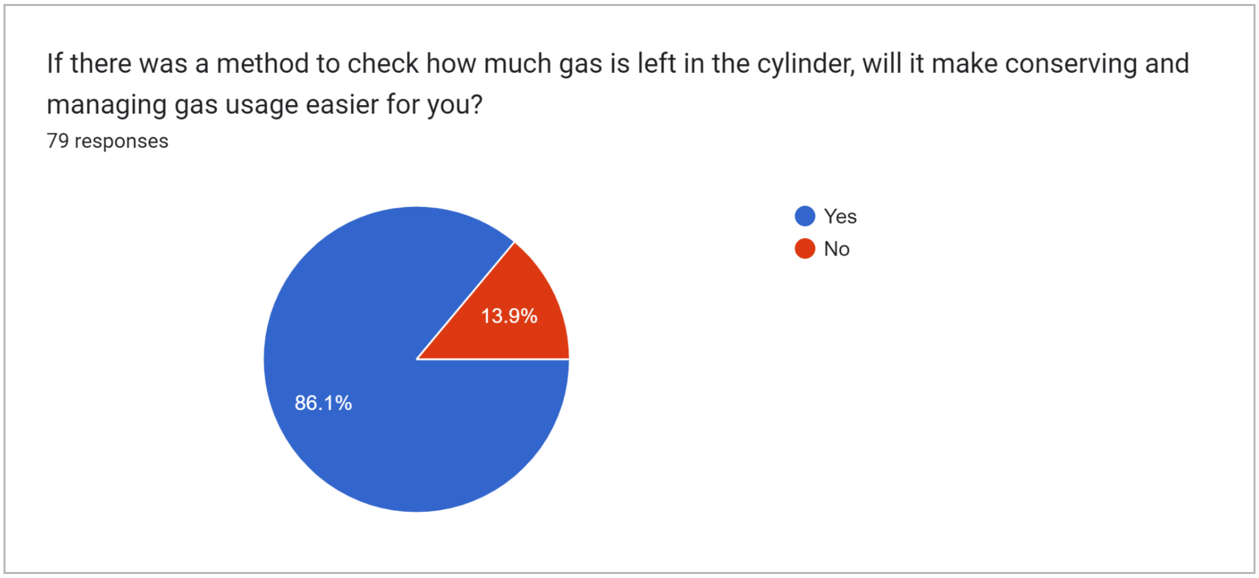
Since the product is to be aimed at the average Sri Lankan household, affordability should also be a priority consideration.

We conducted a short survey through Google Forms to confirm whether the problem we identified is actually worth handling according to the general public, and whether our solution is suitable.

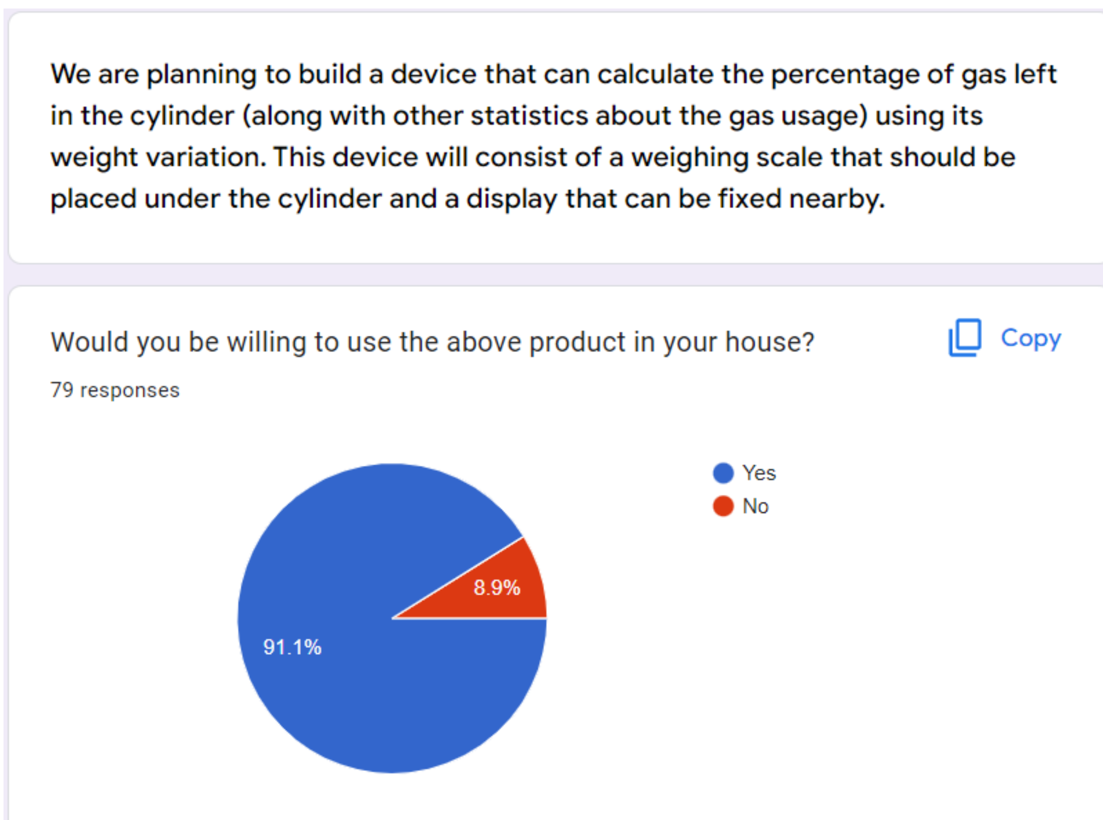
- Out of the 79 responses we received, almost all (76) were from people who use gas as a cooking fuel. Gas conservation at home is extremely necessary for 38% of the respondents, and 84.6% considered it to be at least a moderate necessity.



- 86.1% of the respondents agreed that a method of checking the gas level will make it easier to conserve and manage gas usage in their households.



- More than 90% of them responded that they are willing to use our product, which indicates that our product addresses the problem in an appealing manner to the users, and will be beneficial to a significant portion of LP gas users.



For further details, [reference to the survey]

Product Description

The product we propose to solve the above problem is a device that can calculate the amount of remaining gas in the cylinder by measuring its weight. The product will be a simple weighing scale on which the gas cylinder should be placed. It will track the weight variation of the cylinder over time and calculate the percentage of gas remaining in the cylinder, using the known or precalculated weight of an empty gas cylinder. The device will be connected to the user's mobile phone through Bluetooth, and the information can be viewed through a mobile app.

The app will provide the following information:

- The percentage of gas remaining in the cylinder
- A graph showing the usage of gas over time, enabling the user to identify periods of high gas usage
- Other statistics about gas consumption such as the time the gas cylinder has been in use for and estimated time until gas runs out
- Notification alerts when a low gas level is detected

The product will be compatible with all sizes of gas cylinders that are commonly used in households, and the user has to select the type and the size of the gas cylinder in the app when a new one is placed.

There will also be LED lights on the scale which will change colour depending on the gas level (green - full, yellow - medium, red - low). This will let other people in the household keep track of the gas level as well, without needing access to the connected phone.

When using the product, the scale can be kept under the cylinder without using a significant amount of extra space. Once the cylinder is placed on the scale, the gas level percentage and statistics can be checked at any time without extra effort from the user until the cylinder has to be replaced.

Technical Specifications

- Accuracy of gas percentage
  12.5 kg cylinder: ±1.6%
  5 kg cylinder: ±4%

- Power supply: 9 V non-rechargeable battery

- Power consumption: 182.5 mW

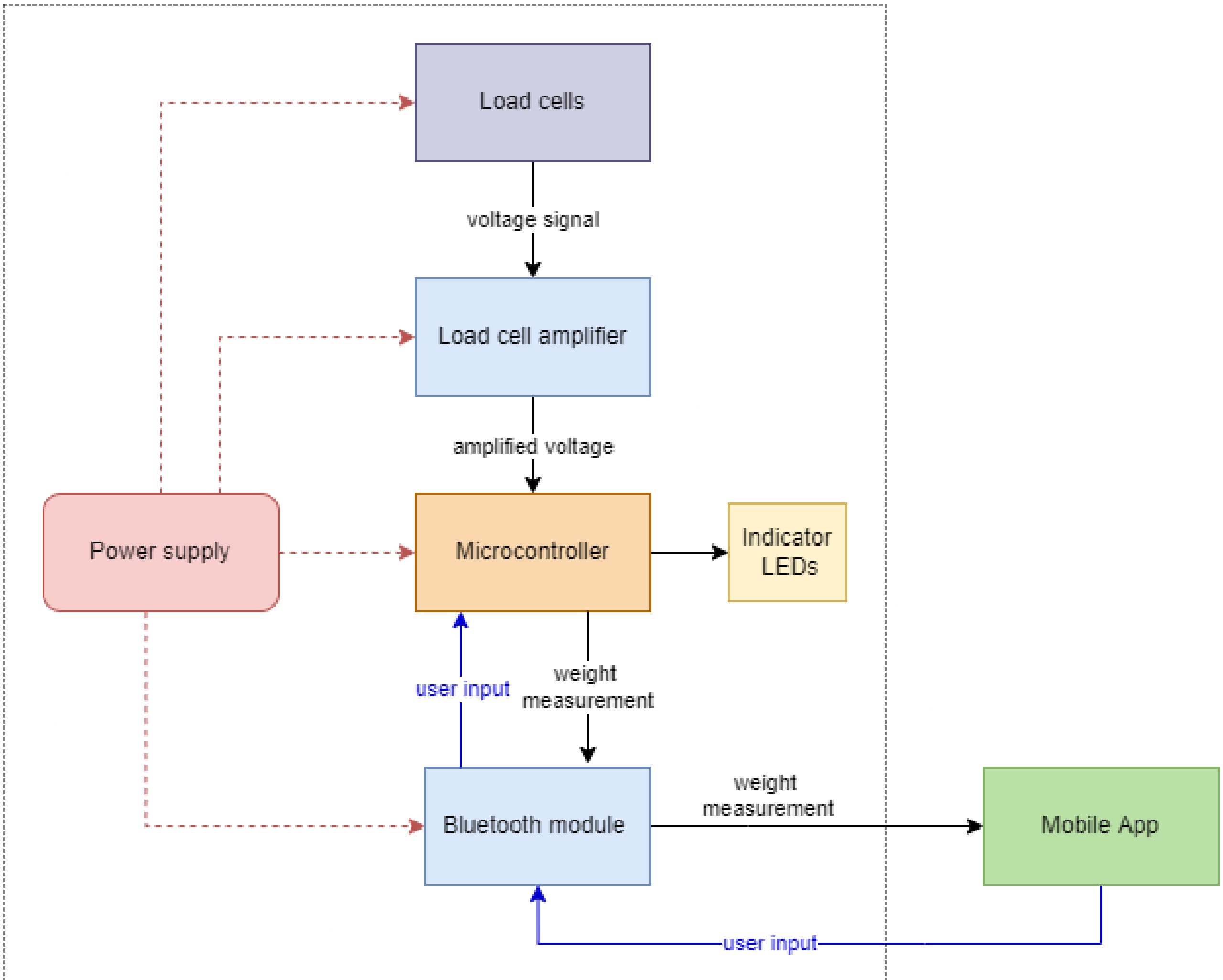
- Dimensions of the scale
  - Height: 3 cm
  - Length and width: 35 x 35 cm

- Expected lifetime: 4 years
  Warranty period for product: 1 year

- The measurements will be updated in the app once per hour whenever the mobile phone is within 10 meters of the scale.

- The mobile app will be able to store data up to one year.

Product Architecture



The functionality of each block are as follows:

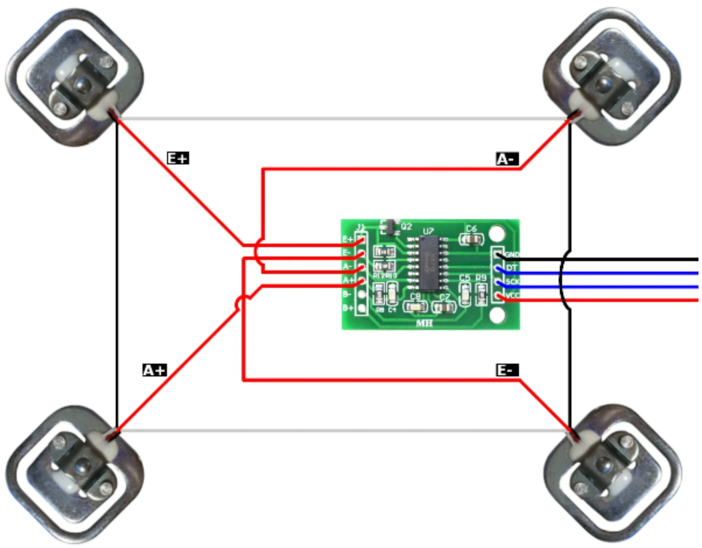
Load Cells

The load cells being used are half bridge strain gauge type load sensors with 50 kg capacity each. The resistance of the strain gauges in the load cells change proportionally to the applied load. This results in a change of output voltage which can be converted into a weight measurement.

For this design, the scale will contain 4 load cells arranged symmetrically and connected in a Wheatstone bridge configuration. This setup allows us to get an output voltage proportional to the load on all 4 load cells without needing to use a voltage summing box to connect them.

Load Cell Amplifier

It is difficult to detect accurate voltage changes directly from the load cells as the change will be very small. Therefore, a load cell amplifier is used to amplify the voltage change before feeding it into the control unit. This task is done by an HX711 load cell amplifier module, which consists of an amplifier with variable gains and a 24-bit analog to digital converter.



Load cell configuration with HX711 [(source)]

Microcontroller

The microcontroller unit used in this design is an ATmega328P-PU microcontroller. It will perform the following tasks:

- Initially calibrates the scale, and takes weight measurements once per hour
- Since a constant load is applied on the scale for a long time, there can be small errors in the measurements due to load cell creep (explained in the technical feasibility section). The microcontroller will try to compensate for these errors and make the reading more accurate.
- It will receive the user input of the size of the cylinder from the mobile app. Using the difference between the weight measurement and the empty gas cylinder weight, the remaining gas percentage can be calculated. The weight of an empty cylinder can be provided in two ways:

1. By measuring the weight when a new cylinder is placed, then subtracting the full weight of gas within it
2. Pre-measuring the empty weight of each type of gas cylinder, since they are made in standard sizes

As method 1 will give inaccurate results if a partially filled cylinder is placed on the scale, method 2 is preferable. If the user does not provide an input, the size of the cylinder will be guessed by the software using weight measurements.

- The results will be sent to the mobile app via the Bluetooth module. When the phone is not connected to Bluetooth, the measurements will be stored in the microcontroller memory until it is possible to send them.

- It will light up LEDs corresponding to the gas level.
    - Low - Red
    - Medium - Yellow
    - Full - Green

- It will operate the zero (tare) button to reset the scale if needed due to errors that can occur over time.

Bluetooth Module

When the phone is connected to Bluetooth, the updated measurements along with the backlog will be sent using the Bluetooth module. It will also communicate user input from the mobile app to the microcontroller. A HC-06 Bluetooth module is used for this purpose.

Mobile App

- Displays the remaining gas percentage
- Sends a notification alert when a low gas level is detected
- Stores previous data in a database and display a graph showing the gas consumption history
- Shows other statistics such as the time the cylinder has been in use for, the busiest times of each day and a rough prediction of when the cylinder will run out of gas
- Prompts the user to input the type and size of the gas cylinder when placing a new one on the scale
- The measurements will be updated every hour if the phone is within 10 m of the scale with Bluetooth turned on.

Power Supply

The device is battery powered, and will require a 9V non-rechargeable battery to operate. Power will be supplied to the microcontroller and other components through a 5V voltage regulator.

Alternative Approaches to the Architecture

Instead of connecting to a mobile app using Bluetooth, a small LCD display with input buttons and a SD card module to store the data could be used.

Between the two methods, there is little to no difference in the cost. However, showing the required statistics in a user-friendly manner can be challenging unless a larger display is used, which significantly increases the cost. The additional parts will make the PCB more complex which is harder to test. Also, it is more convenient for the users if they can simply observe everything from their phone instead of going near the gas cylinder. Therefore, the approach with Bluetooth was chosen for the final design.

Technical Feasibility

Accuracy and Measurement Resolution

According to the specifications of the load cells to be used, the smallest weight change that can be accurately measured is 0.1% of full scale, which is 0.05 kg for one load cell. As the total weight will be distributed among 4 load cells, the smallest accurate measurement for the scale is 0.2 kg. This will provide the percentage of remaining gas at the following resolutions for each size of gas cylinder:

- 12.5 kg: 1.6% (62.5 levels between 0 and 100)
- 5 kg: 4% (25 levels between 0 and 100)
- 2.3 kg: 8.7% (11.5 levels between 0 and 100)

For all cylinder sizes, the above levels of accuracy are sufficient to get a rough measurement of the gas level and to accurately detect a low level of gas. With 5 kg and 12.5 kg cylinders (which are the most commonly used sizes in households), the gas percentage will be more accurately tracked.

As the HX711 module contains a 24-bit ADC, the measurements and the displayed percentages can have a higher resolution than 0.2kg even though the accuracy may not reach the same level.

Power Supply

The operating voltage ranges of the main components in the design are as follows:

- Load sensors: 2.6 - 5.5 V
- Load cell amplifier module: 2.7 - 5 V
- Bluetooth module: 3.3 - 5 V
- Microcontroller: 3.3 - 5 V

Therefore, a power supply with a 9V battery is sufficient for the device to operate.

Power Consumption

The estimated power consumption of each main component are as follows:

- Load sensors: 25mW x 4 = 100mW
- Load cell amplifier: 7.5mW
- Bluetooth module: 40mW
- Microcontroller: 25mW
- LEDs and other components: 20mW

Total power consumption of the product is 182.5 mW.

As the main components only need to stay active once per hour, it is possible to significantly reduce this power consumption by adding sleep cycles.

Effect of Load Cell Creep

Load cell creep is the change that happens in the load cell output while under a load with no other environmental changes. This is an issue that needs to be considered in this application as the load will remain on the scale for a long time.

For the load sensors being used in this device, creep is specified as 0.1% of full scale. This is the amount of creep that can occur when the cells are loaded near full capacity. However, the maximum expected weight on this scale is around 30 kg (total weight of a 12.5 kg cylinder) which is significantly lower than the rated capacity of 200 kg. Therefore, under normal operation the creep should become negligible. Even if a larger creep occurs, theoretically the output should stabilize over time which will eventually provide accurate readings, and there are methods to compensate for creep if it becomes necessary.

For maximum accuracy, the users will be recommended to tare (zero) the scale whenever the cylinder is taken off as this can also reduce the effects of creep.

Effect of Shock Loads

As gas cylinders are usually roughly handled, there is a possibility of it being dropped from a 2-3 inch height when placing it on the scale. This causes a shock overload that may damage the load cells. Therefore, steps must be taken to handle this overload force.

The load sensors being used here have a safe overload capacity 150% of full scale, which is 75 kg. As a weight applied on the scale is equally distributed among the 4 load cells, the system is capable of handling an overload force of up to 300 kg. This can protect the scale from a 30 kg gas cylinder being dropped from a maximum height of 4 cm.

The framing of the weighing platform intended to center the weight (described in the enclosure section) will also act as a natural deterrent against dropping the cylinder from a larger height.

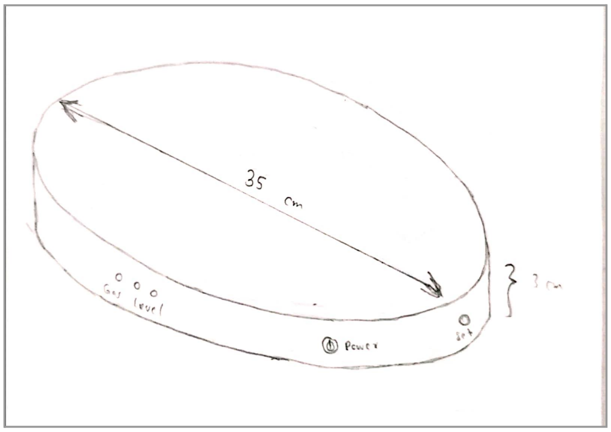
Hardware and Software Requirements

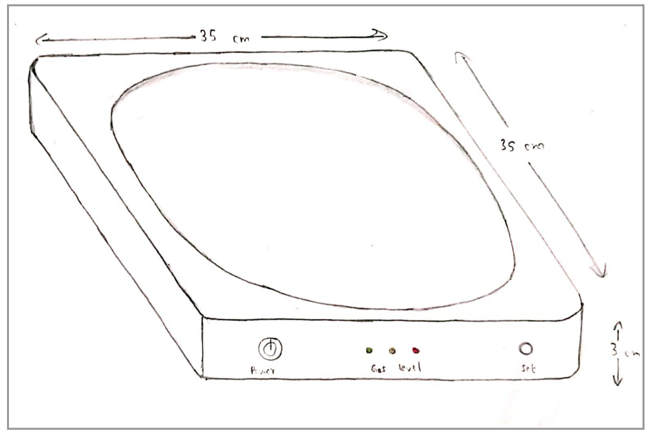
- All the components used in this product are commonly available in the market at an affordable price.
- As the measurements are taken hourly, only a small amount of storage space is necessary for the mobile app (8760 data values for one year). The calculations and statistical analyses do not require special processing power.

Note: The load cell specifications used for feasibility calculations were taken from a seller's website as a standard datasheet was not available.

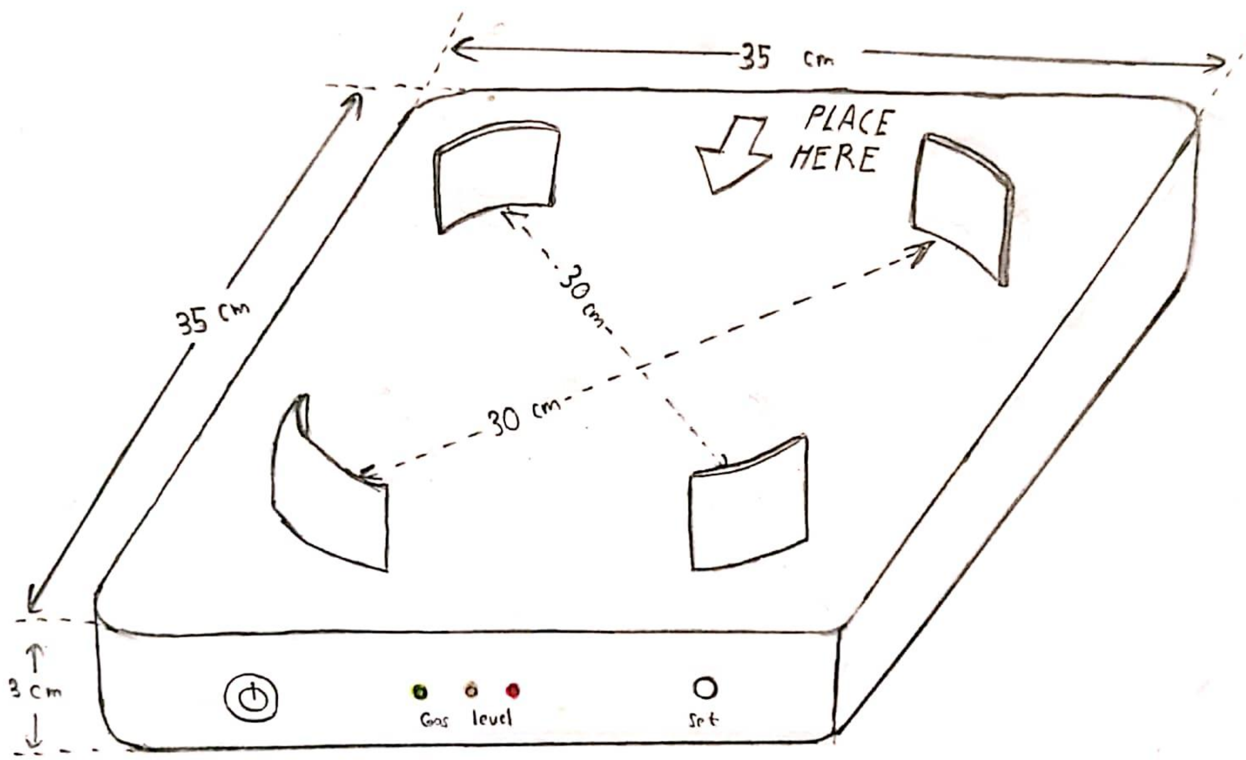
Enclosure Design

Sketches of Initial Ideas





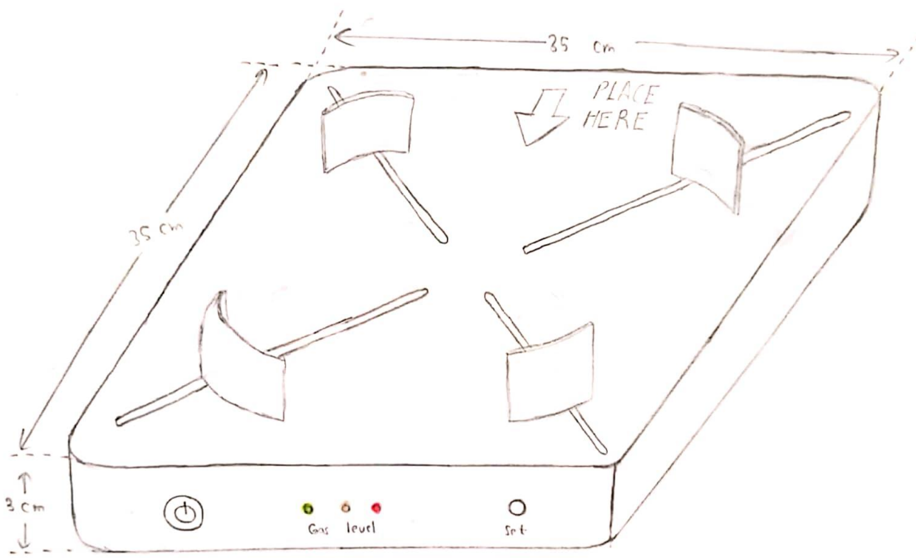
Final Sketch



Note: The button placements were slightly modified in the final design.

- The cylinder has to be placed within the four corner frames.
- The frames will ensure that the weight of the cylinder is evenly distributed between the four load cells, and will prevent it from accidentally shifting out of place. It will also deter the user from dropping the cylinder from a height onto the scale.
- In this design, the frame will only fully serve its purpose for 12.5 kg and 5 kg cylinder sizes. However, the device will still work properly with a small 2.3 kg cylinder when it is placed in the middle of the scale.

- A further improvement to the design would be to make the frames adjustable along railings to fit smaller diameter cylinders as shown below.



Material, Color and Finish

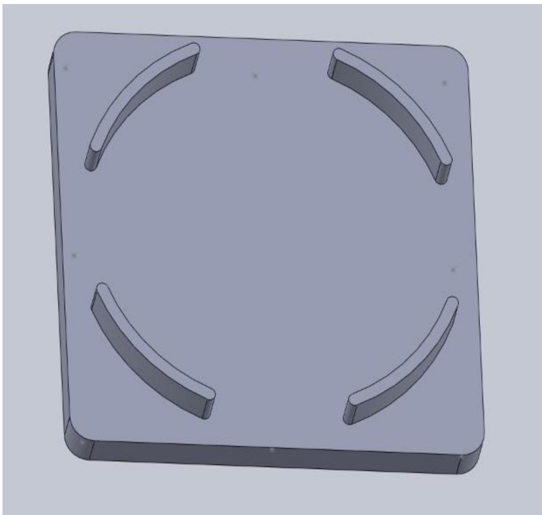
The building material should have the following qualities:
- Low cost, to be able to deliver the product at an affordable price
- Not bendable (bending might affect the accuracy of weight measurements)
- Light, compared to a gas cylinder
- Strong enough to withstand a sudden impact

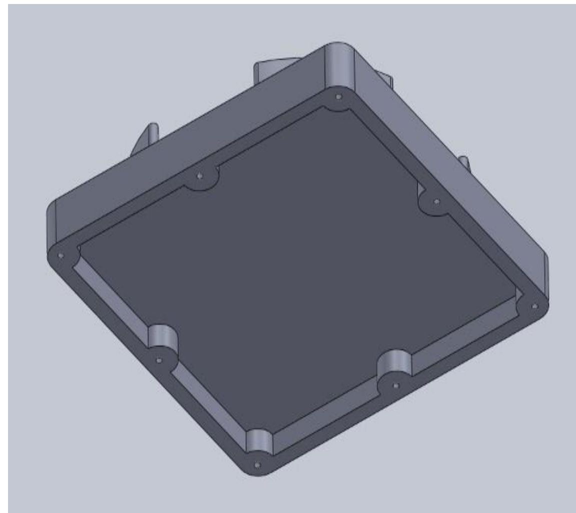
We chose aluminum metal for the enclosure as it fits the above requirements. The color and finish of the scale will also be metallic as it can be perceived as durable and well built.

The prototype was built by CNC machining wood, as it also fits the above qualities.

Solidworks Design for the Enclosure

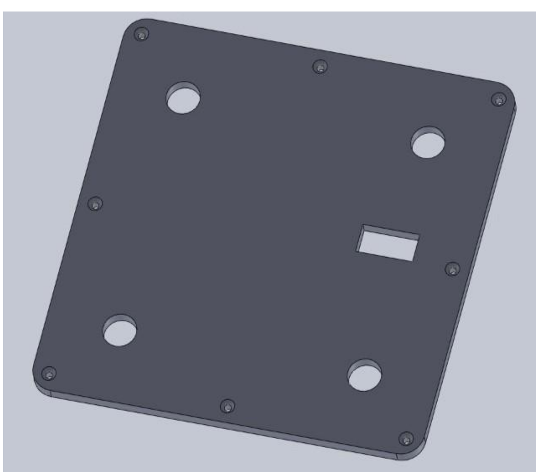
- Top Part





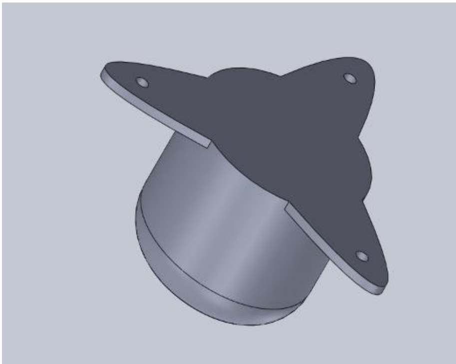
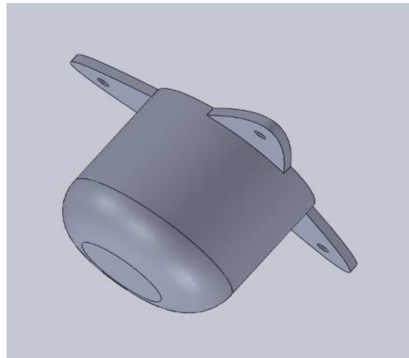
The top part was designed to have a 1.5 cm thickness (when built out of wood) to ensure its strength under rough handling of the gas cylinder. The load cells will be fixed to the inside surface with their load-bearing piece facing down.

- Bottom part



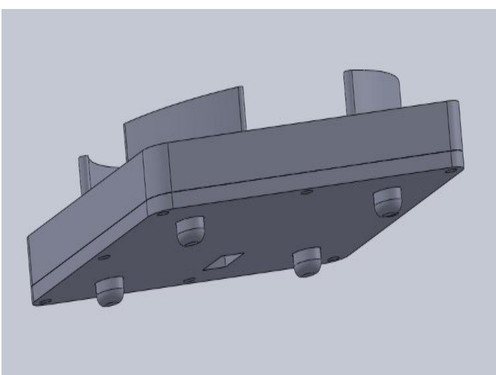
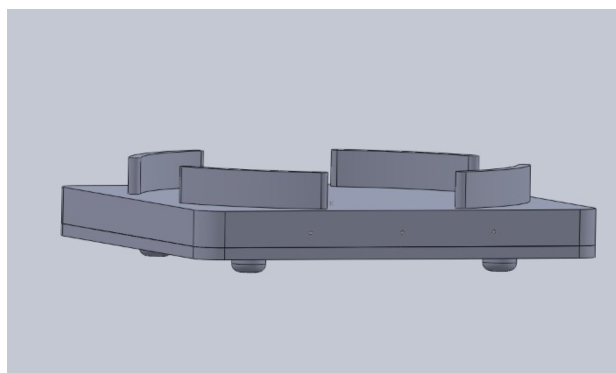
The bottom part contains holes for the feet of the scale and the battery holder. It is designed to be removable (fixed using screws) for easy access to the inside components during testing.

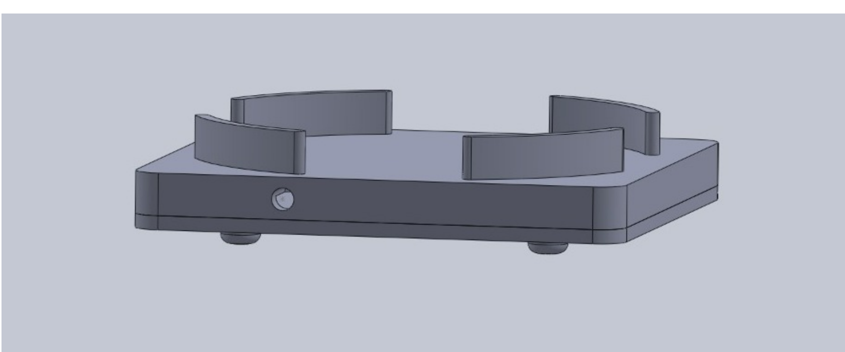
- Feet

Adding feet to the enclosure was required to let the load cells bear the full weight of the scale without leaving exposed load cells or an unstable setup. The feet will be fixed to the inner bottom surface of the scale using nails, in a way that only allows a small vertical motion. When the scale is assembled and placed upright, the feet will completely press against the load cells and transfer the force.

- Full Assembly



Enclosure of Prototype (CNC machined wood)

Top view (before attaching railings)

Bottom view

Rubber pads were attached to the bottom of the feet to prevent slipping.

User Interface Design

Weighing Scale

Since all detailed inputs and outputs are given in the mobile app, the user interface of the scale itself is minimal.

- Front

The current level of gas will be indicated by lighting up one of the above shown LEDs. The color scheme used is intuitive for the user, and can be confirmed by the text if unsure. The red LED will stand out more and indicate urgency when the gas level is low.

- Back

The reset button is used to re-zero the scale, needed due to the reasons explained in the technical feasibility section. The functionality of the button and when to use it will be described in the user manual. The button is placed on the back of the scale so that the user will not accidentally press the button while a cylinder is placed.

Mobile App

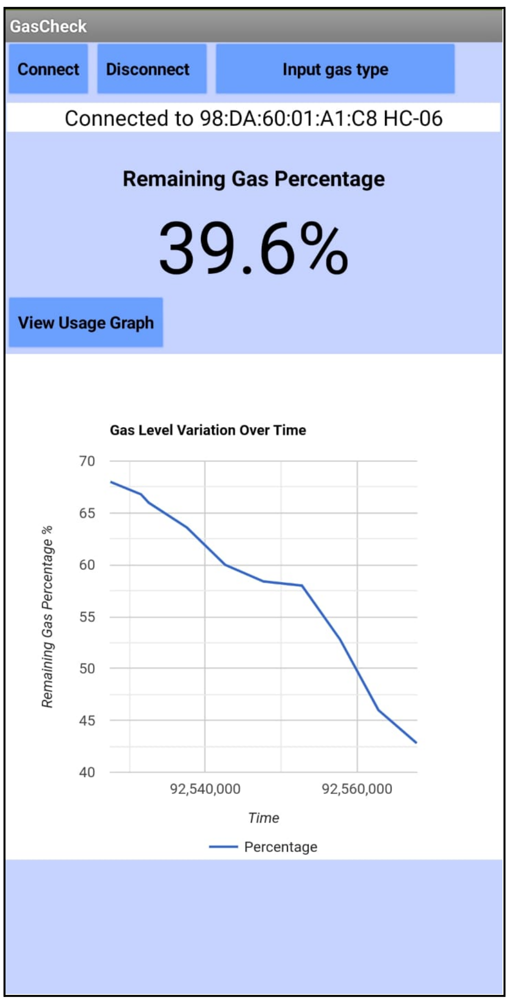
- The gas percentage, graphs and other statistics will be clearly displayed in the mobile app as a data dashboard.

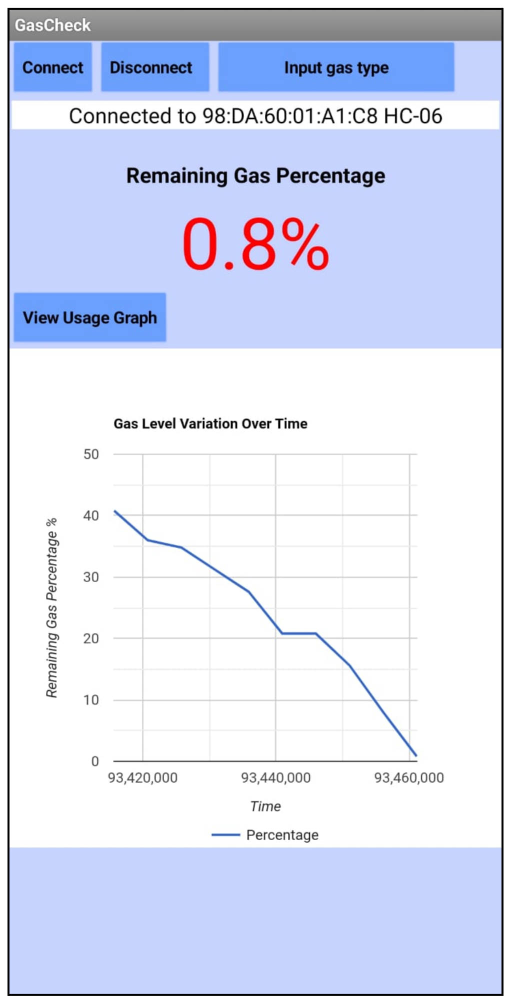
- When it is detected that a new cylinder was placed on the scale, the app will send a notification prompting the user to select the brand and size of the gas cylinder from a dropdown menu.

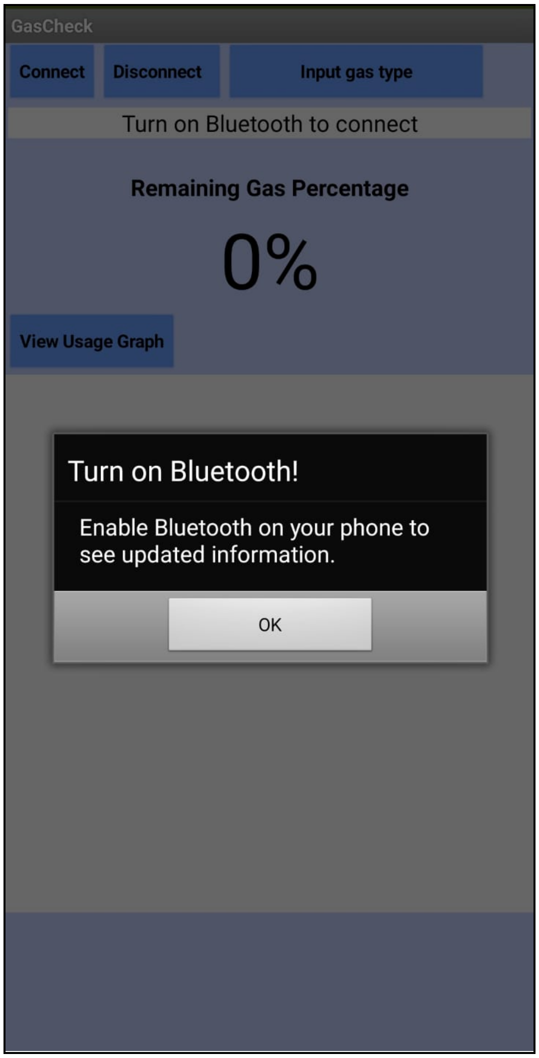
- The app will send a notification when a low gas level is detected, so that the user will be informed even if they do not open the app.
- The measurements will only be updated if Bluetooth is turned on. If the user opens the app without turning it on, a message will appear prompting them to do so.
- After initially pairing with the device, the app will automatically connect with it whenever the device is in range.

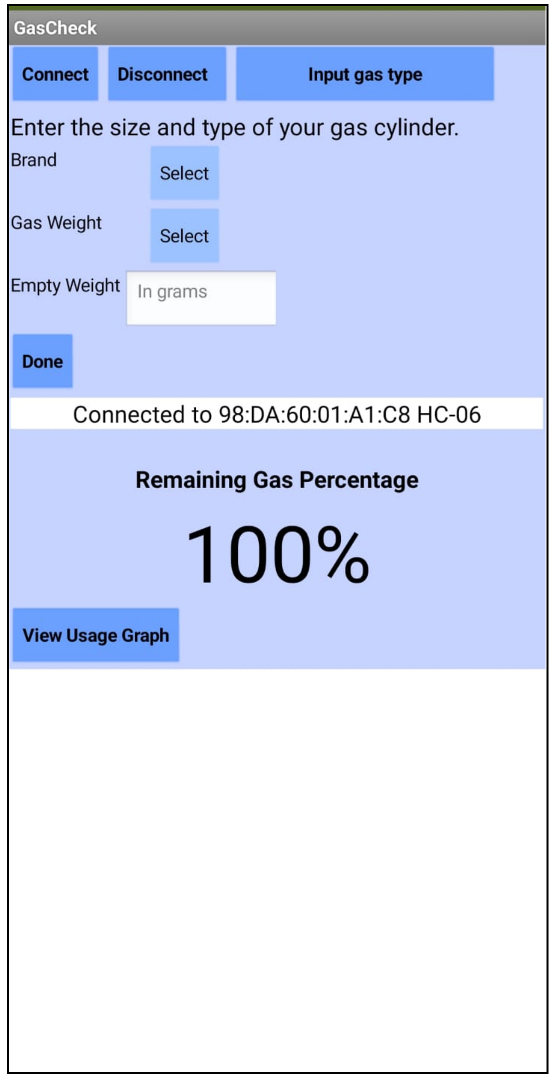
Prototype of Mobile App

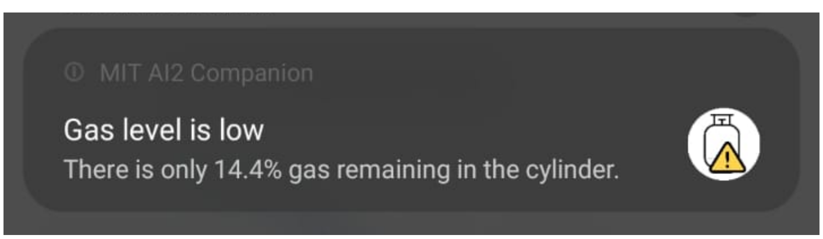
A simple mobile app was designed using MIT App Inventor to complete the functional prototype.

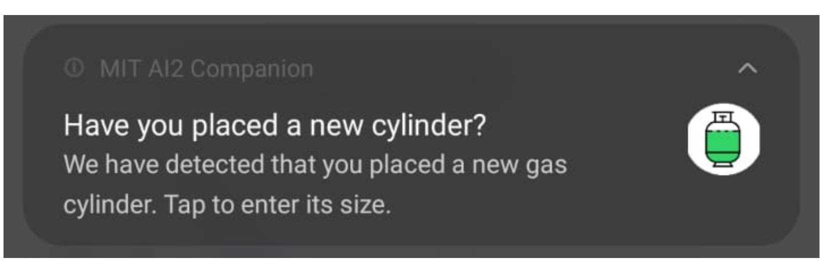





GasCheck
Connect
Disconnect
Input gas type
Turn on Bluetooth to connect
Remaining Gas Percentage
0%
View Usage Graph
Turn on Bluetooth!
Enable Bluetooth on your phone to see updated information.
OK

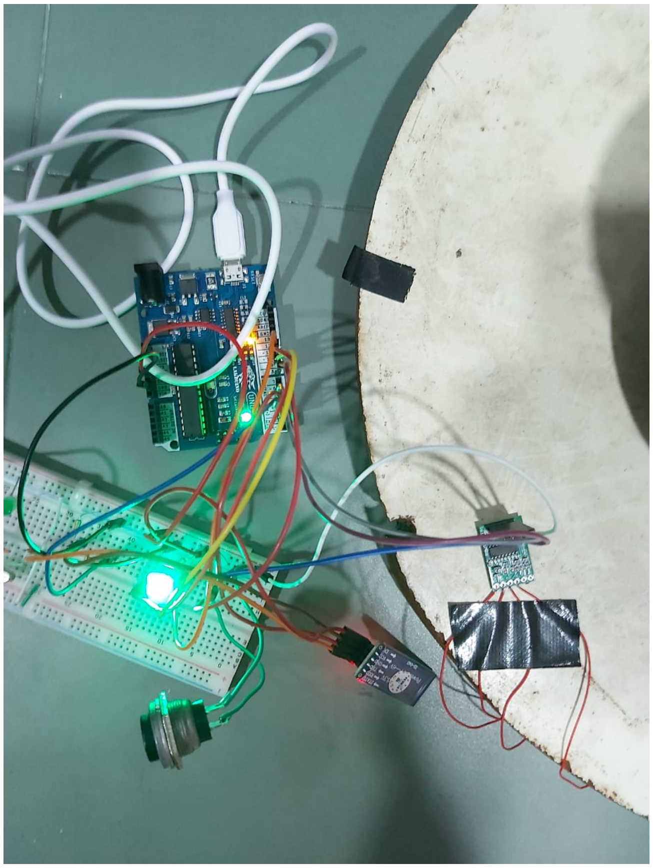

GasCheck
Connect
Disconnect
Input gas type
Enter the size and type of your gas cylinder.
Brand
Select
Gas Weight
Select
Empty Weight
In grams
Done
Connected to 98:DA:60:01:A1:C8 HC-06
Remaining Gas Percentage
100%
View Usage Graph


MIT AI2 Companion
Gas level is low
There is only 14.4% gas remaining in the cylinder.


MIT AI2 Companion
Have you placed a new cylinder?
We have detected that you placed a new gas cylinder. Tap to enter its size.

Functional Prototype

Implementation using an Arduino board



Underside with load cell circuit

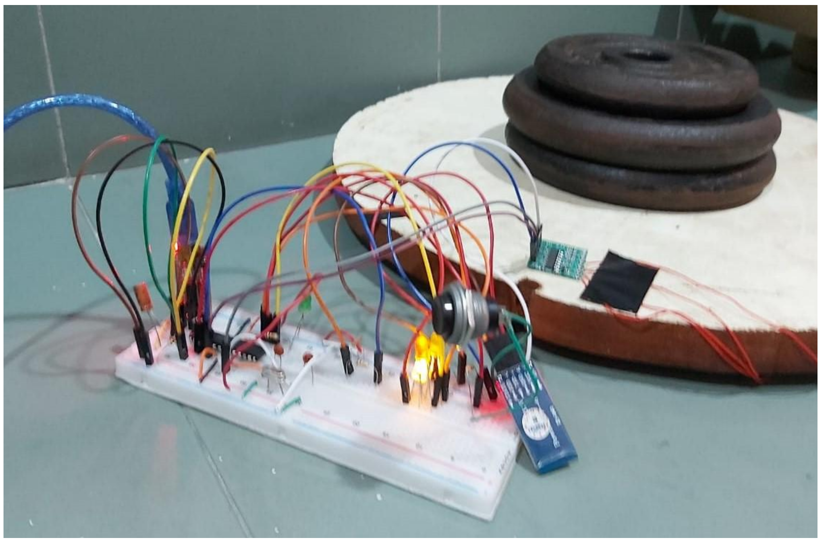
The initial implementation of the product was done using an Arduino board. The HX711_ADC Arduino library was used in the code to convert the output of the HX711 module into weight.

A wooden board with the load cells fixed on the underside was used as the scale. A load cell holder was 3D printed using a standard model to allow the load cell to flex under an applied load.

In the program, the LEDs were set to light up as follows:
- 65-100 % : Green
- 35-65 % : Yellow
- 0-35 % : Red

Values less than 0 and greater than 100 are taken as 0 and 100 respectively, as there can be small errors in the measurement that result in such values.

Implementation using a microcontroller on a breadboard



The next stage of implementation was done using an ATmega328P microcontroller. The same Arduino code from above was used after burning the Arduino bootloader into the microcontroller.

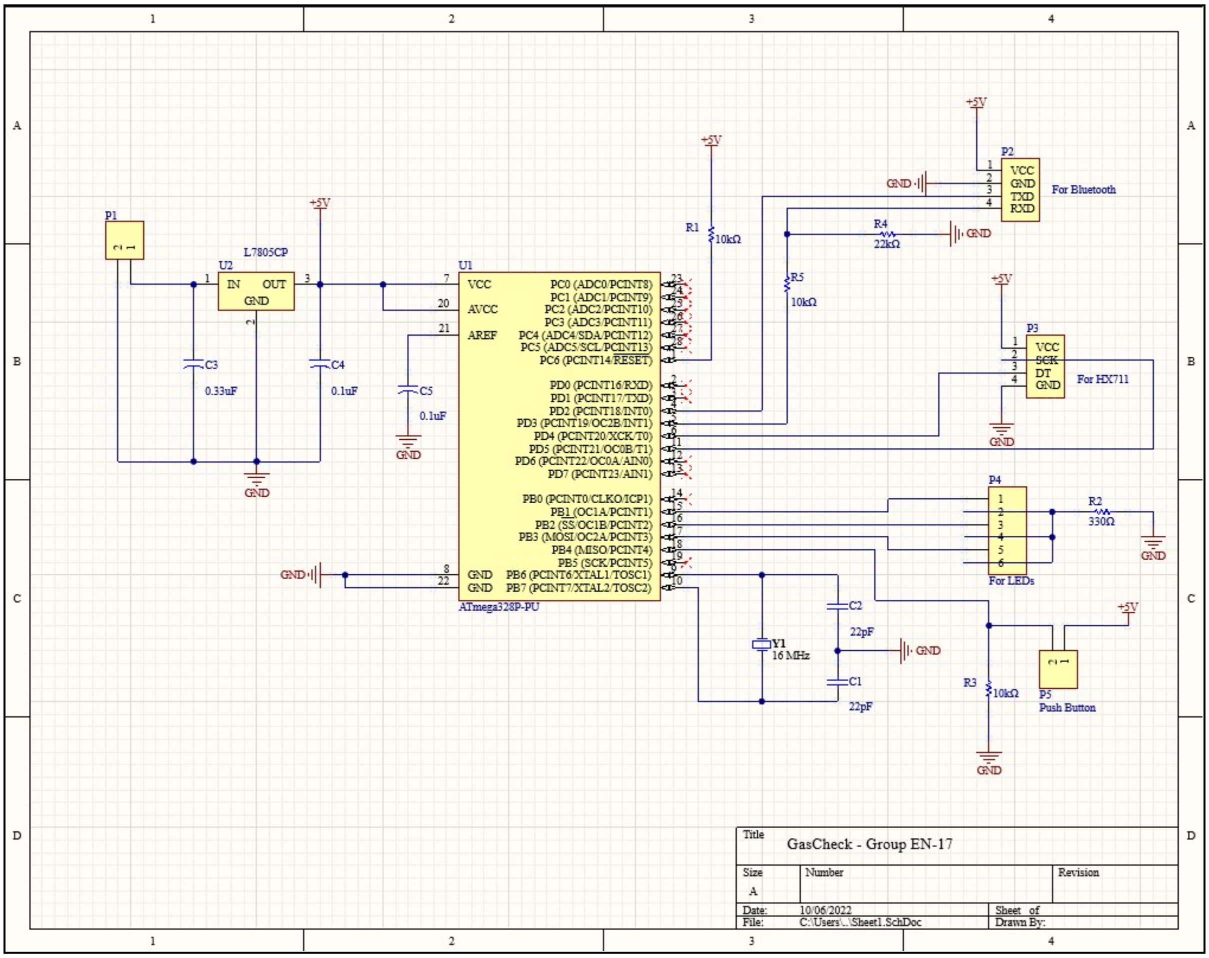
Results

The scale was calibrated using a known 10 kg weight and was tested using a set of 2.5 kg and 1.25 kg weights. The resulting measurements had an accuracy of around ± 0.05 kg for most of the time, with an occasional error of up to 0.1 kg. This matches with the expected level of accuracy calculated in the technical feasibility section.

The device accurately and quickly responded to changes in weight as small as 0.2 kg.

PCB Design

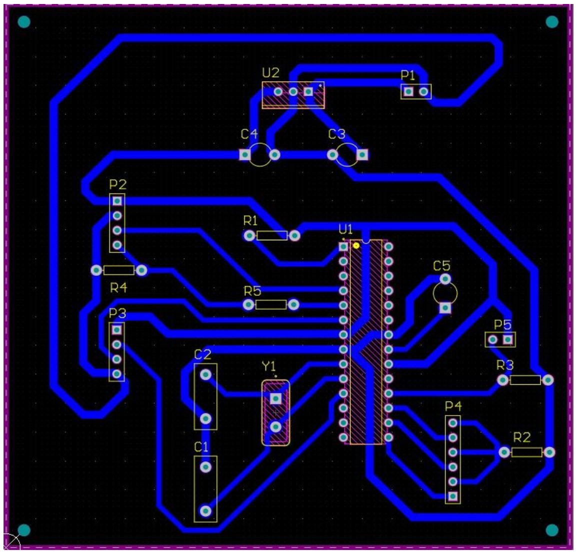
Schematic Diagram



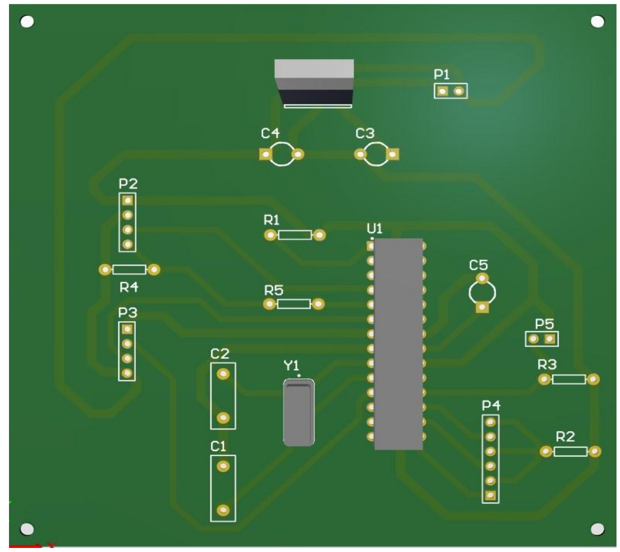
For the Bluetooth module, the TX and RX ports operate at 3.3V. Therefore the 5V output of the microcontroller is connected to the RX pin through a voltage divider to obtain 3.3V. The chosen resistor values are 10 kΩ and 22 kΩ, providing a 3.4V voltage to the module. The TX pin can be directly connected as the microcontroller will interpret 3.3V as the 1 logic level.

The power circuit contains capacitors to further stabilize the regulated voltage.

PCB Layout Diagram



Routing



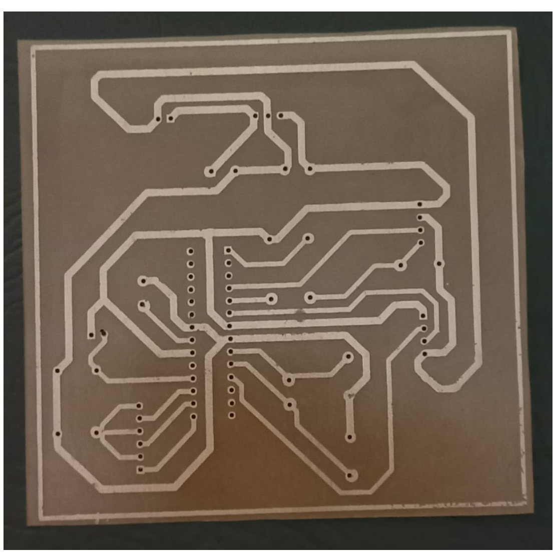
Top 3D View

As we planned to manufacture the PCB using the home-made PCB method, larger trace widths were used to ensure reliability.

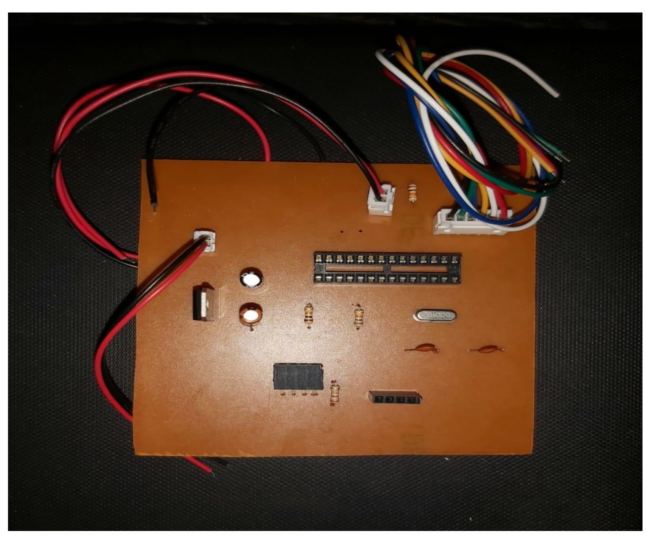
- Signal lines - 1 mm
- Power lines - 1.5 mm

The designed PCB was 9.9cm x 9.6cm. The size of the PCB was not a concern as the enclosure is large enough to accommodate it.

Manufactured PCB



Copper traces



After soldering the components

Female headers are used to directly connect the Bluetooth and HX711 modules to the PCB. JST connectors are used to connect the LEDs, button and battery as they should be placed outside the PCB.

Discussion

- Initially we calibrated the scale using a 1.8 kg known weight. However, as this a lower weight than any gas cylinder, it resulted in large errors (around 5 kg) when measurements were taken for large weights (around 30 kg). This was fixed when we re-calibrated using a 10 kg weight. The reason for choosing 10 kg is because we expect to measure weights roughly between 2.5 kg and 30 kg using this scale, therefore using a value in the middle of the range as the calibration weight will give a more accurate linear approximation of the load vs. voltage function.

- The type of load cells used in this product are extremely sensitive to physical changes. We encountered a problem where the output value started drifting rapidly after one of the load cell wires was accidentally tugged too hard. We were able to isolate that the issue was in a load cell and it was fixed after replacing that load cell. This will not be a problem in the final product as the load cells and wiring is fully contained within the scale and out of reach of the user.

- The voltage signal from the load cell is prone to being noisy. We were able to reduce the noise by making the wires as short as possible and soldering all the connections in place. However, as long as the noise amplitude is small, rapid variations of the voltage will not affect the performance of this application as it will still produce a close enough value for the remaining gas percentage.

Project Budget

	Item	Units	Unit Price (Rs.)	Total Price (Rs.)
Scale	Half bridge strain gauge load cell	4	260.00	1040.00
	HX711 load cell amplifier module	1	340.00	340.00
	HC-06 bluetooth module	1	750.00	750.00
Interface	Push button	2	10.00	20.00
	LEDs	3	4.00	12.00
	Resistors	5	1.00	5.00
Microcontroller	ATMEGA328P-PU	1	1700.00	1700.00
	16 MHz Crystal oscillator	1	40.00	40.00
	22pF Capacitor	2	0.75	1.50
Power circuit	LM7805 voltage regulator	1	60.00	60.00
	0.33µF Capacitor	1	1.50	1.50
	0.1µF Capacitor	1	2.00	2.00
	PCB			500.00
	Enclosure			4130.00
Other costs for manufacturing and packaging				850.00
Total Cost				**9452.00**

- The manufacturing costs will be reduced by buying components in bulk and using cheaper bulk manufacturing methods for the enclosure. This could allow us to reduce the product price to around Rs. 6000.

- Keeping a 10% profit margin, the product can be sold at Rs. 7000.

Marketing, Sales and Beyond

Marketing

The main marketing medium is planned to be social media platforms such as Facebook, Youtube, Instagram, Whatsapp, etc. Posts and videos about the product will be shared on these platforms.

The following points could be highlighted in the posts to showcase the usefulness of the product:

- The problem of gas availability in the country
- The inconvenience of not knowing the remaining gas amount
- That the device requires minimal effort from the user to use
- The use of smart technology to make it convenient for the user to monitor the gas usage at any time

Introduction videos about the device can also be made explaining the following:

- Ease of use and the convenience
- The positive effects on gas conservation seen after using the device in a household
- The accuracy and reliability of the device

Although there are existing products similar to this gas level indicator, they are not readily available to buy in Sri Lanka and are much more expensive. Therefore, comparison with those products is not useful as a marketing strategy.

In addition to social media marketing, the product will have an official website, and advertisements can be posted on online shopping websites such as Aliexpress, Daraz, eBay, etc.

Sales

Initially, we are planning to limit manufacturing to 20 units and observe how the product is perceived by the general public. A 15% discount can be given to encourage people to buy it.

The product will be sold in the following ways in order to reach a larger customer base:

- Through online stores such as Daraz, eBay, Aliexpress
- Through local electronic appliance stores
- Through the official website

Product Packaging

As the enclosure will be built to withstand rough handling, a normal cardboard packaging box with bubble wrap will be sufficient for this product.

The outside of the box will show:

- Product name and a short description (GasCheck - gas cylinder level indicator)
- Images showing the intended use of the product
- That it uses a mobile app and Bluetooth connectivity

The product will come with a user manual that gives instructions on how to pair the device with Bluetooth, how to use the mobile app and the function of buttons and LEDs.

Maintenance and Repair

- No regular maintenance is required for the proper operation of the device.
- The user is recommended to zero the scale before placing a new cylinder to reduce errors caused by load cell creep and other environmental factors.
- The zero button can also be used if the scale starts giving wrong measurements due to some issue.
- Periodic software updates will be provided for the mobile app.
- As only common components are used, it will be possible to repair the device from general electronic repair centers in addition to our own representatives. 1 year warranty will be provided.

Reuse of Components

- The load cells are likely to be the point of failure in a device like this, as they are subjected to continuous strain. In that case, the more expensive electronic components such as the microcontroller and Bluetooth module can be taken out and reused.
- The metal parts of the enclosure can be recycled and used for other purposes.

Task allocation among the group members

Gunatilake P.T.B.	● Initial and final sketches of enclosure ● Enclosure design ● Device assembly (circuits and enclosure)
Pasqual A.C.	● Product architecture and feasibility analysis ● Microcontroller programming for the scale ● Mobile app development ● Testing and troubleshooting the device
Pathirana K.P.T.R.	● Product architecture and feasibility analysis ● PCB design ● Testing and troubleshooting the device
Perera G.L.S.M.	● Marketing, sales and product life cycle analysis ● Device assembly (circuits and enclosure) ● Interfacing with Bluetooth

Appendix

References

- ATmega328P Datasheet
  https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf
- 4 Load cell configuration with HX711
  https://circuitjournal.com/50kg-load-cells-with-HX711
- Load cell specifications https://a.co/d/5yoCstb
- Converting the circuit from Arduino to ATmega328p
  https://www.youtube.com/watch?v=Sww1mek5rHU&feature=youtu.be

Arduino code

```
#include <HX711_ADC.h>
#include <EEPROM.h>
#include <SoftwareSerial.h>
#include "storage.h"  // Header file containing queue for temporary storage

#define ZERO_BUTTON 12
#define RED_LED 9
#define YELLOW_LED 10
#define GREEN_LED 11

#define HX711_DOUT 4
#define HX711_SCK 5
#define BT_RX 2   //Connect this pin to Bluetooth TXD
#define BT_TX 3   //To RXD

SoftwareSerial BTserial(BT_RX, BT_TX);  //RX, TX
HX711_ADC LoadCell(HX711_DOUT, HX711_SCK);

void data_read_all() {  //read all data and empty array
  float data;
  while (!data_is_empty()) {
    data = data_get();
    BTserial.print(data);
    BTserial.print(",");
  }
  BTserial.println();
}

float cal_factor = 24.33;  // These values are updated manually after calibration
```

```
long tare_offset = 8569861;

unsigned long t = 0;  //timestamp
float empty_weight = 2500.0;
float full_gas_weight = 7500.0;

int cur_led = 0;

void calibrate() {   // Calibration is done beforehand
  Serial.println("Remove weights to tare and press t");
  while (1) {
    if (Serial.available() > 0) {
      char command = Serial.read();
      if (command == 't') {
        LoadCell.tare();
        Serial.print("Tare complete. Tare offset: ");
        long offset = LoadCell.getTareOffset();
        Serial.println(offset);
        break;
      }
    }
  }
  float known_mass = 0;
  Serial.println("Place known mass and enter its weight in grams: ");
  while (1) {
    if (Serial.available() > 0) {
      known_mass = Serial.parseFloat();
      if (known_mass != 0) {
        Serial.print("Known mass is: ");
        Serial.println(known_mass);
        break;
      }
    }
  }
  LoadCell.refreshDataSet();
  float newCalibrationValue = LoadCell.getNewCalibration(known_mass);
  Serial.print("New calibration value has been set to: ");
  Serial.println(newCalibrationValue);
  cal_factor = newCalibrationValue;

  Serial.println("Save this and press k");
  while (1) {
    if (Serial.available() > 0) {
      char command = Serial.read();
      if (command == 'k') {
        break;
      }
    }
```

```
  }
}

float calc_percentage(float weight) {
  float gas_weight = weight - empty_weight;
  float perc = gas_weight * 100 / full_gas_weight;
  return perc;
}

void change_led(int cur, int next) {
  if (cur != 0) digitalWrite(cur, LOW);
  if (next != 0) digitalWrite(next, HIGH);
}

void light_led(float perc) {
  int next_led = cur_led;
  if (perc < 0) next_led = 0;
  if (perc >= 0 && perc < 35) next_led = RED_LED;
  if (perc >= 35 && perc < 65) next_led = YELLOW_LED;
  if (perc >= 65) next_led = GREEN_LED;

  if (next_led != cur_led) {
    change_led(cur_led, next_led);
    cur_led = next_led;
  }
}

void process_new_data(float weight) {
  data_put(weight);
  float perc = calc_percentage(weight);
  light_led(perc);
  Serial.print("Remaining gas percentage: ");
  Serial.print(perc);
  Serial.println("%");
}

void setup() {
  pinMode(ZERO_BUTTON, INPUT);
  pinMode(RED_LED, OUTPUT);
  pinMode(YELLOW_LED, OUTPUT);
  pinMode(GREEN_LED, OUTPUT);

  Serial.begin(9600);
  BTserial.begin(9600);
  LoadCell.begin();
  calibrate();  // comment this and uncomment below lines after scale is calibrated
  //LoadCell.setCalFactor(cal_factor);
  //LoadCell.setTareOffset(tare_offset);
```

```
}

void loop() {

  static boolean newDataReady = 0;
  const int measureInterval = 60 * 60 * 1000;

  if (LoadCell.update()) newDataReady = true;

  if (newDataReady) {
    if (millis() > t + measureInterval) {
      float weight = LoadCell.getData();
      Serial.print("Load_cell output val: ");
      Serial.println(weight);
      process_new_data(weight);
      newDataReady = false;
      t = millis();
    }
  }

  // If tare button is pressed, do tare operation
  if (digitalRead(ZERO_BUTTON) == HIGH) {
    LoadCell.tare();
    Serial.print("Tare complete. Tare offset: ");
    long offset = LoadCell.getTareOffset();
    Serial.println(offset);
  }

  if (Serial.available() > 0) {
    char command = Serial.read();
    if (command == 't') {
      LoadCell.tare();
      Serial.print("Tare complete. Tare offset: ");
      long offset = LoadCell.getTareOffset();
      Serial.println(offset);
    }
  }

  if (BTserial.available())
  {
    char comm = BTserial.read();
    if (comm == 'd') {
      data_read_all();
      Serial.println("Data sent");

    } else if (comm == 'i') {
      full_gas_weight = BTserial.parseFloat();
      empty_weight = BTserial.parseFloat();
```

```
      Serial.println("Input parameters");
      Serial.print("Gas weight: ");
      Serial.println(full_gas_weight);
      Serial.print("Empty weight: ");
      Serial.println(empty_weight);
    }
  }
}
```