

**Electronics and Embedded**

**Systems Development**

**ELNC-6008**

**Practical Project Report**

|  |
| --- |
| **Smart Water Bottle Filler** |

By

|  |  |
| --- | --- |
| Student Names | Student Numbers |
| Gayathri Muraleedharan Sujatha | 0948198 |
| Swapnil Raj | 0964466 |
| Ameenul Ehsan | 0952936 |
| Nithin Symon | 0964442 |

|  |  |
| --- | --- |
| Faculty Advisor: | Professor Brent Matthews |
| Date Submitted: | April 19, 2020 |

# **Executive Summary**

The product that the team has provided is the filling agent which will provide water as per the capacity of the water bottle in order to prevent the wastage of one of the most precious resource exist on the earth that is “water”.

The product is designed which is customer friendly and can relief their pain of pocket as this product is cost efficient and they can rely on this agent for a really long period. The product is contributing its role of saving energy for the future times and is sustainable in all aspects. Once the user this agent, they will be delightful after using it as this product gives user a special greet on using this product and placing bottle at a specific area, also it will interact with the customers in a unique way by displaying a motivational quote to inspire them to save water.

# **Table of Contents**

[Executive Summary i](#_Toc38213365)

[Table of Contents ii](#_Toc38213366)

[List of Figures iii](#_Toc38213367)

[List of Tables iv](#_Toc38213368)

[Acknowledgements v](#_Toc38213369)

[1.Introduction 1](#_Toc38213370)

[2.Limitations 1](#_Toc38213371)

[3.Design Improvement and Customer attraction 1](#_Toc38213372)

[4.Scope Deliverables 2](#_Toc38213373)

[5.Beneficial Capabilities 2](#_Toc38213374)

[6.Product Design Embodiment 3](#_Toc38213375)

[7.Operational Overview 3](#_Toc38213376)

[8.Technical Issues 4](#_Toc38213377)

[9.Design Justification 5](#_Toc38213378)

[10.Observatory Readings 6](#_Toc38213379)

[11.Analysis of Readings 9](#_Toc38213380)

[12. Conclusion 9](#_Toc38213381)

[13.Recommendations 9](#_Toc38213382)

[14. References 10](#_Toc38213383)

[15.Declaration of Originality 11](#_Toc38213384)

[Appendix I: Description 12](#_Toc38213385)

# **List of Figures**

[Figure 1: Additional feature of LED showing status and manual switch](#_Toc4049551) 2

[Figure 2: Product design of water filler station……………………………………………………………………… 3](#_Toc4049556)

[Figure 3: Product design of water filler station 4](#_Toc4049556)

[Figure 4: Tera term window of bottle 1 6](#_Toc4049556)

[Figure 5: Tera term window of bottle 2 6](#_Toc4049556)

[Figure 6: LCD display showing filling water status 7](#_Toc4049556)

[Figure 7: LCD display showing message after water is filled 7](#_Toc4049556)

[Figure 8: LCD status of the bottle detection and water filling 8](#_Toc4049556)

# **List of Tables**

**Table 1: LED indication……………………………………………………………………………………………………………5**

# **Acknowledgements**

The team has worked hard and continuously on this project and has given considerable amount of time to complete the project within the time. Before moving ahead, the team would like to convey their special regards to the Professor Brent Mathews who gave the team an opportunity to work on this project. Counting down from the starting phase to ending phase, Professor Brent Mathews has stood up for the team and helped the team whenever the team got stuck somewhere in middle of the process and he showered his guidance and helped the team. Professor Brent Mathews has provided the team with the smart water filler chassis to make things clear and easier for the team. Special thanks to Professor Chris Talbot for making things clear for the team whenever the team got stuck on the technical side of this project. The team would like to thank Professor Mona for her guidance and suggestions to improve the project in a unique way.

We are thankful of Fanshawe College for giving the team an opportunity to showcase their hardware and software proficiency by working on the project.

We are thankful to each member of the team who contributed their time, and for all that efforts to complete this project on time.

## 

# **1.Introduction**

Intellect Electronics presents the smart water filling station which can detect water level and smart enough to measure the capacity of water bottle after detecting its presence. This product can save water and contribute to the effort of saving the world’s most important resource. The product will fulfil the customer’s requirements in all the aspects that if from cost efficient model to long term dependability.

The chassis consists of suitable sensors which are used to detect the presence of water bottle being placed and which can calculate the volume of the water bottle after its detection.

The detection signal is used as a green indication to proceed furthermore and calculate the capacity of water and once the capacity is calculated, it will give green signal to open the valve and fill the water bottle up to 90% of its capacity and prevents overfilling of water bottle.

# **2.Limitations**

After designing and completing the project the team has come up with some limitations that are as follows:

* The prototype is not able to prevent overfilling of water bottle if the bottle is already filled partially or having water inside it.
* The automatic filling and volume calculation prototype are compatible with the standard bottle shape.
* User should place the bottle correctly at the specific area provided in the water filling station.

# **3.Design Improvement and Customer attraction**

The team members have worked on the project to make the product more friendly to customers:

* **HMI (Human Machine Interface) feature:** The LCD attached with the chassis will tell the customers about the role they are playing in saving water by using this product. The machine will greet customers whenever the bottle is being detected.
* **LED functionality:**  People are fond of traffic lights functionality and using the same concept, product is having LED’s which will show the proper function of the prototype.

**![A screenshot of a cell phone

Description automatically generated]()**

**Figure 1: Additional feature of LED showing status and manual switch**

* **Auto-Fill switch:** To overcome the limitation at some extent, the product is having auto-fill switch which can be pressed manually by customer in order to fill their water bottle up to their desired percentage. This switch will help the users who are having water bottle but, it is not in standard shape. This will help users who re in hurry and forgets to place water bottle at specific position then they can use autofill to save their time and have their water bottle filled.

# **4.Scope Deliverables**

* **Deliverable 1**: In this functional prototype, the team has used two ultrasonic sensors to detect the height and diameter of the water bottle in order to calculate the volume of the water bottle. Once the volume is calculated, results were converted into time values and these values gives the time period for which valve will open and fill the water bottle.
* **Deliverable 2:** For the non-standard shape bottle, top up manual switch is provided that will be operated by user itself.
* **Deliverable 3:** The team has operated on pic microcontroller and all the required code is saved in the pic microcontroller.

# **5.Beneficial Capabilities**

* User-friendly and this product is easy to access.
* The product cost efficient.
* The product is energy efficient.
* Water saving machine that is filling water and saving water.

# **6.Product Design Embodiment**

A picture containing computer, sitting, table, white

Description automatically generated

**Figure 2: Product design of water filler station**

# **7.Operational Overview**

Flow chart of the water filler working.

A screenshot of a cell phone

Description automatically generated

**Figure 3: Working process of smart water filler**

# **8.Technical Issues**

1. It was a challenge interfacing ultra-sonic sensor HC-SR04 with our PIC18F45K22 microcontroller [1].
2. The position of the ultrasonic sensor, it was placed near to the nozzle to find the exact height of the bottle placed
3. We had trouble in printing the characters in the LCD [2].
4. Due to Covid-19 we were not having enough wires for the connection. We solved it by using the previous semesters alligator clips used for soldering.

# **9.Design Justification**

We used two ultra-sonic sensors for the calculation of the volume. One ultra-sonic sensor is placed at the top near the water nozzle to get the exact height of the machine. Another ultra-sonic is placed at the lower bottom of the machine to find the radius of larger bottle and smaller bottles [3].

We placed one IR sensor for the detection of the bottle, the machine will only calculate and open the nozzle only when the bottle is detected.

Three LED is used for showing the status of water filling:

|  |  |  |
| --- | --- | --- |
| **No.** | **LED** | **Indication** |
| 1 | Green | Bottle detection |
| 2 | Yellow | Water filling |
| 3 | Red | Filling completed |

**Table 1: LED indication**

We added one LCD display for showing the status of water filling and one message at the end when the filling is completed.

# **10.Observatory Readings**

This is the tera term window of our project code

A screenshot of a cell phone

Description automatically generated

**Figure 4: Tera term window of bottle 1**

A screenshot of a computer

Description automatically generated

**Figure 5: Tera term window of bottle 2**

A circuit board

Description automatically generated

**Figure 6: LCD display showing filling water status**

A circuit board

Description automatically generated

**Figure 7: LCD display showing message after water is filled**

A picture containing table, sitting, white

Description automatically generated

**Figure 8: LED status of the bottle detection and water filling**

# **11.Analysis of Readings**

In the this we calculated the radius and height of the bottle using two ultrasonic sensors.

Height of the bottle is calculated by subtracting the ultra-sonic sensors calculated distance from total height from the bottle. And radius is found by subtracting the calculated distance by the ultra-sonic sensor from the center point of the water filler.

Based on the height and radius we calculated the volume of the bottle.

Time is found by dividing the volume by flow rate of the machine.

# **12. Conclusion**

As the years are passing by, humans are altering the natural resources and polluting environment. One of the most important resources is also getting affected due to global warming and various reasons. The most important resource is water and it is very precious for human to survive, in order to save water this machine is programmed to fill water bottle’s as per the capacity of water bottles and prevent the wastage of water. If water is not saved in the present it will not be available for future, so this machine saves water and can be used for long time. This is the purpose of designing sustainable machine.

# **13.Recommendations**

The product is designed to control the energy bills and reduce water wastage. The product will make you inspire and make u feel cheerful every time someone use it. Use this product and contribute your support in order to save water. Use this because it is cost efficient and satisfactory also.

# **14. References**

|  |  |
| --- | --- |
| [1] | microchip, "microchip," PIC, [Online]. Available: https://ww1.microchip.com/downloads/en/DeviceDoc/40001412G.pdf. [Accessed Jan 2020]. |
| [2] | S. E. ELECTRONICS, "beta-estore," [Online]. Available: https://www.beta-estore.com/download/rk/RK-10290\_410.pdf. [Accessed March 2020]. |
| [3] | "components101," September 2020. [Online]. Available: https://components101.com/ultrasonic-sensor-working-pinout-datasheet. [Accessed January 2020]. |

# **15.Declaration of Originality**

The team 03 members undersigned, hereby declares that all the work done in this project report is original. All the data (table, images, figures, output readings) are original piece of work by team 03. All the citations and references have been mentioned suitable.

|  |  |  |
| --- | --- | --- |
| **Student Name** | **Signature** | **Date**  **(DD MMM YYYY)** |
| Nithin Symon | *A close up of a whiteboard  Description automatically generated* | 19 April 2020 |
| Gayathri Muraleedharan Sujatha | WhatsApp Image 2020-01-11 at 12.20.38 PM.jpeg | 19 April 2020 |
| Ameenul Ehsan | A picture containing text  Description automatically generated | 19 April 2020 |
| Swapnil Raj | WhatsApp Image 2020-01-11 at 12.08.01 PM.jpeg | 19 April 2020 |

# **Appendix I: Description**

//---------------------------------- Water Filler Station-----------------------------------------------

/\*-----------------------------------------------------------------------------------------------------------

File Name: waterFillerStation.c

Author: Team 03

Date: 16/02/2020

Modified: Name or None

copyright Fanshawe College, 2019

Description: Filling bottle placed in filling station based on volume.

----------------------------------------------------------------------------------------------\*/

// Preprocessor ------------------------------------------------------------------------------

#include "pragmas.h"

#include <stdlib.h>

#include <stdio.h>

#include <p18f45k22.h>

#include "usart.h"

#include <delays.h>

//-------------------------------- Constants -------------------------------------------------------

#define TRUE 1

#define FALSE 0

#define PI 3.14

#define T0FLAG INTCONbits.TMR0IF

#define LIMIT 4

#define MAXANGLE 180.0

#define SCL TRISCbits.RC3

#define SDA TRISCbits.RC4

#define I2CSTART SSPCON2bits.SEN

#define I2CFLAG PIR1bits.SSPIF

#define I2CSTOP SSPCON2bits.PEN

#define I2CACK SSPCON2bits.ACKSTAT

#define IRS PORTDbits.RD2

#define LED\_RUN PORTDbits.RD1

#define IRSLED PORTAbits.RA3

#define LED\_STOP PORTAbits.RA2

#define INSTWRITE 0x0C

#define INSTWRITE\_EN 0x08

#define DATAWRITE 0x0D

#define DATAWRITE\_EN 0x09

#define HIGHNIBBLE 0xF0

#define LOWNIBBLE 4

#define CHARSIZE 20

#define ON 0 //To turn led on,The led in water filler turn on in low input

#define OFF 1 //to turn led off

//------------------------------Global Variables---------------------------------------------------

int height =0;

int value\_us1;

int time\_low\_us1,time\_high\_us1;

int radius = 0;

int value\_us2;

int time\_low\_us2,time\_high\_us2;

unsigned int volume = 0;

unsigned int time = 0;

float flowRate = 00009463;

char secondTimer=0;

char print1[CHARSIZE]={"FILLING WATER"};

char print2[CHARSIZE]={"YOU SAVED OUR PLANET"};

char index=0;

//------------------------------------------Functions---------------------------------------------

/\*--- set\_osci\_4MHZ ---------------------------------------------------------------------------

Author: Team 03

Date: 16 FEB 2020

Modified: Name or None

Desc: Set oscillator 4MHZ

Input: None

Returns: None

--------------------------------------------------------------------------------------------\*/

void set\_osci\_4MHZ(void)

{

OSCCON = 0x52;

OSCCON2 = 0x04;

OSCTUNE = 0x80;

} // eo set\_osci\_4MHZ

/\*--- set\_serialPort ---------------------------------------------------------------------------

Author: Team 03

Date: 16 FEB 2020

Modified: Name or None

Desc: Serial port configuration

Input: None

Returns: None

--------------------------------------------------------------------------------------------\*/

void set\_serialPort(void)

{

SPBRG1=25;

TXSTA1=0x26;

RCSTA1=0x90;

BAUDCON1=0x40;

}// eo set\_serialPort

/\*--- portConfig ---------------------------------------------------------------------------

Author: Team 03

Date: 16 FEB 2020

Modified: Name or None

Desc: Port configuration

Input: None

Returns: None

--------------------------------------------------------------------------------------------\*/

void portConfig(void)

{

LATA=0x0C;

ANSELA=0x00;

TRISA=0xF3;

LATB=0x00;

ANSELB=0x00;

TRISB=0xCF;

LATD=0x02;

ANSELD=0x00;

TRISD=0xFC;

LATC = 0x00; //Set Outut voltage

ANSELC = 0x00; //Digital

TRISCbits.RC0 = 0;

SCL = 1; //Set as input

SDA = 1; //Set as input

SSPADD = 9; //100 KHz I2C Clock with Fosc = 4Mhz [Fosc/(4\*(SSPADD + 1))]

SSPCON1 = 0x28; //Enable I2C

} // eo portConfig

/\*--- timerConfig ----------------------------------------------------------+-----------------

Author: Team 03

Date: 01 APR 2020

Modified: Name or None

Desc: Timer configuration for 1 second

Input: None

Returns: None

--------------------------------------------------------------------------------------------\*/

void timerConfig(void)

{

T0CON=0x91;

TMR0H=0x0B;

TMR0L=0xDC;

T0FLAG=FALSE;

} // eo timerConfig

/\*--- ResetTimer ---------------------------------------------------------------------------

Author: Team 03

Date: 01/04/2020

Modified: Name or None

Desc: Resetting timer configuration

Input: None

Returns: None

--------------------------------------------------------------------------------------------\*/

void resetTimer(void)

{

TMR0H=0x0B;

TMR0L=0xDC;

T0FLAG=FALSE;

} // eo resetTimer

/\*--- i2c\_init ---------------------------------------------------------------------------

Author: Team 03

Date: 03 APR 2020

Modified: None

Desc: Initalizes the i2c start condition

Input: None

Returns: None

--------------------------------------------------------------------------------------------\*/

void i2c\_init(void)

{

I2CSTART = 1; //initiate Start condition

while(I2CSTART); //Wait

I2CFLAG = 0; //clear SSPIF interrupt flag

}

//eo i2c\_init

/\*--- delay\_i2c ---------------------------------------------------------------------------

Author: Team 03

Date: 03 APR 2020

Modified: None

Desc: Delay 4 us to latch the data in enable pin

Input: addr

Returns: None

--------------------------------------------------------------------------------------------\*/

void delay\_i2c(void)

{

int i=0;

for(i=0;i<LIMIT;i++)

Nop();

}

//eo delay\_i2c

/\*--- i2c\_write ---------------------------------------------------------------------------

Author: Team 03

Date: 03 APR 2020

Modified: None

Desc: write byte via i2c bus

Input: addr

Returns: None

--------------------------------------------------------------------------------------------\*/

void i2c\_write(char addr)

{

SSPBUF = addr; //LCD Slave address with write

while(!I2CFLAG); //Wait upto 9th clock pulse

I2CFLAG = 0; //clear SSPIF interrupt flag

if(I2CACK) //ACK was not received

{

I2CSTOP = 1; //initiate stop condition

while(I2CSTOP); //Wait

return; //Stop the program

}

}

//eo i2c\_write

/\*--- lcd\_cmd\_write ---------------------------------------------------------------------------

Author: Team 03

Date: 03 APR 2020

Modified: None

Desc: write command byte via i2c bus

Input: data

Returns: None

--------------------------------------------------------------------------------------------\*/

void lcd\_write(char data, char write, char write\_en)

{

i2c\_write((data&HIGHNIBBLE) | write);

Delay1KTCYx(2);

i2c\_write(write\_en);

i2c\_write((data<<LOWNIBBLE) | write);

Delay1KTCYx(2);

i2c\_write(write\_en);

}

//eo lcd\_cmd\_write

/\*--- i2c\_stop ---------------------------------------------------------------------------

Author: Team 03

Date: 03 APR 2020

Modified: None

Desc: write command byte via i2c bus

Input: data

Returns: None

--------------------------------------------------------------------------------------------\*/

void i2c\_stop(void)

{

I2CSTOP = 1; //initiate Start condition

while(I2CSTOP); //Wait

}

//eo i2c\_stop

/\*---lcd\_printing--------------------------------------------------------------------------

Author: Team 03

Date: 03 APR 2020

Modified: None

Desc: lcd printing function

Input: None

Returns: None

--------------------------------------------------------------------------------------------\*/

void lcd\_printing(void)

{

i2c\_init(); // Initial LCD

i2c\_write(0x4E); //Slave address(LCD) with write condition

lcd\_write(0x20,INSTWRITE,INSTWRITE\_EN); //DL-4bit,1 line display,5x8 dots

lcd\_write(0x01,INSTWRITE,INSTWRITE\_EN); //clear display

lcd\_write(0x0E,INSTWRITE,INSTWRITE\_EN); //display and cursor ON

lcd\_write(0x1C,INSTWRITE,INSTWRITE\_EN); //shift cursor right

lcd\_write(0x02,INSTWRITE,INSTWRITE\_EN); //home position

}

/\*---message1--------------------------------------------------------------------------

Author: Team 03

Date: 03 APR 2020

Modified: None

Desc: Filling water message

Input: None

Returns: None

--------------------------------------------------------------------------------------------\*/

void message1(void)

{

lcd\_printing();

for(index=0;index<CHARSIZE;index++)

{

lcd\_write(print2[index],DATAWRITE,DATAWRITE\_EN); //Display message 1

}

}

//eo message1

/\*---message2--------------------------------------------------------------------------

Author: Team 03

Date: 03 APR 2020

Modified: None

Desc: Message after filling water

Input: None

Returns: None

--------------------------------------------------------------------------------------------\*/

void message2(void)

{

lcd\_printing();

for(index=0;index<CHARSIZE;index++)

{

lcd\_write(print2[index],DATAWRITE,DATAWRITE\_EN); //Display message 2

}

}

//eo message2

/\*--- initializefunc ---------------------------------------------------------------------------

Author: Team 03

Date: 07 MAR 2020

Modified: None

Desc: Initalizes the overall System.

Input: None

Returns: None

--------------------------------------------------------------------------------------------\*/

void initializefunc(void)

{

set\_osci\_4MHZ();

set\_serialPort();

portConfig();

} //eo initializefunc

/\*----------------------- MAIN FUNCTION --------------------------------------------------------------------------------------------------------------------------------------------------------\*/

void main(void)

{

initializefunc();

while(1)

{

T3CON = 0x06; //initializing timer0

TMR3H= 0;

TMR3L= 0;

T1CON = 0x06; //initializing timer1

TMR1H= 0;

TMR1L= 0;

//------------------------radius calculation---------------------------------------

PORTBbits.RB4=1; //trig high for ultrasonic2

Delay1KTCYx(0.0001); //10micro sec delay

PORTBbits.RB4=0; //trig low for ultrasonic2

while(PORTAbits.RA1 == 0); //wait for echo of ultrasonic2

T1CON = 0x07; //timer starts

while(PORTAbits.RA1 == 1); //waiting stops of ultrasonic2

T1CON = 0x06; //timer stops

time\_low\_us2=TMR1L;

time\_high\_us2=TMR1H;

value\_us2 = time\_low\_us2;

value\_us2|=(time\_high\_us2<<8);

value\_us2/=58.82;

radius= 23; //width of the filling station is 46cm, centre is at 23cm

radius-=value\_us2;

//------------------------height calculation---------------------------------------

PORTBbits.RB5=1; //trig high for ultrasonic1

Delay1KTCYx(0.0001); //10micro sec eeeer SEDGG

PORTBbits.RB5=0; //trig low for ultrasonic1

while(PORTAbits.RA0 == 0); //wait for echo of ultrasonic1

T3CON = 0x07; //timer starts

while(PORTAbits.RA0 == 1); //waiting stops f ultrasonic1

T3CON = 0x06; //timer stops

time\_low\_us1=TMR3L;

time\_high\_us1=TMR3H;

value\_us1 = time\_low\_us1;

value\_us1|=(time\_high\_us1<<8);

value\_us1/=58.82;

height= 30; //Total height of filling station is 30 cm

height-=value\_us1;

//------------------------volume calculation---------------------------------------

volume=height\*radius\*radius\*PI;

printf("\033[1;1HINTELLECT ELECTRONICS");

printf("\033[3;2HVolume=\033[K");

//---------------------Time calculation--------------------------------------------

time = volume;

time/= flowRate;

printf("\033[4;2HTime=\033[K");

//----------------------LEDs for flow indication for time calculated------------------

IRSLED=OFF;;

if(IRS==TRUE) //Bottle detection by IR sensor

{

IRSLED=ON;

if(time)

{

timerConfig();

if(T0FLAG)

{

resetTimer();

if(secondTimer<time)

{

message1();

printf("\033[3;2HVolume= %d\n",volume);

printf("\033[4;2HTime= %d\n",time);

secondTimer++;

LED\_RUN = ON; //water flows

LED\_STOP = OFF;

}

else

{ message2();

secondTimer = 0;

LED\_STOP = ON;

LED\_RUN = OFF;

}

}

}

}//eo if

}//eo while