A

PROJECT REPORT

ON

WEIGH-IN-MOTION

AT

VASU AUTOMATION & ELECTRICALS

In the partial fulfillment of the

Requirements for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

ELECTRONICS & COMMUNICATION

Submitted by

Keniz Pankaj Gandhi
EC-038, 13ECUBS029
B.Tech. SEM. VIII

Under the guidance of

Faculty Supervisor
Prof. Smith S. Thavalapill

External Guide
Mr. Pravin Patel
Project Manager

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

FACULTY OF TECHNOLOGY,

DHARMSINH DESAI UNIVERSITY, NADIAD-387001.

APRIL 2017

NAAC ACCREDITED

CERTIFICATE

This is to certify that Ms. Keniz P Gandhi (13ECUBS029) EC-038 studying in B.Tech. Sem. VIII (Electronics & Communication) of Faculty of Technology, D. D. University has satisfactorily delivered seminars and has satisfactorily completed term work in the subject of Industrial Training / Project.

Date: April 6,2017

Prof. Smith S. Thavalapill
(Faculty Supervisor)

Dr. Nikhil Kothari
(Head of Department)

ISOQAR
QUALITY ASSURED FIRM
ISO 9001 : 2008
Certificate No. 5673 QMS 001

Dharmsinh Desai University

College Road, Nadiad 387 001, India. Ph : 91 0268 2520502 Fax : 91 0268 2520501 Website : www.ddu.ac.in

ACKNOWLEDGEMENT

I take the opportunity to express my gratitude to all the concerned people who have directly or indirectly contributed towards completion of the project.

I extend my sincere gratitude towards **VASU AUTOMATION & ELECTRICALS** for providing the opportunity and resources to work on this project.

I am extremely grateful to **Mr. Pravin Patel**, my mentor in VASU AUTOMATION & ELECTRICALS for his guidance and invaluable advice during the projects. Also to my guide, **Prof. Smith S. Thavalapill** whose insight encouraged me to go beyond the scope of the project and this broadened me learning on this project.

I further extend my sincere gratitude to the Dean of Faculty of Technology, **Prof. D. G. Panchal** and Head of Electronics and Communication Department, **Dr. Nikhil J. Kothari** for providing the opportunity to work at this company.

Thanking You.

KENIZ GANDHI

ABOUT THE COMPANY

Vasu provides Electrical, Electronics and Automation products and turnkey solutions to the clients across the Gujarat, throughout the India. It has been employee owned company since 2007 and trace origins to the early 1992s. They provide services which focus on Customer Centricity rather than Profit Making Motive in such a competitive market.

Their well-equipped quality control system helped them in designing user friendly products that are durable and long lasting. Their spacious warehouse has all the facilities to stock the products in a safe manner, so that they are able to meet all forms of market demands. In addition to all these, they have in house testing and training facility.

Work details at glance:-

- Design, Procurement, Manufacturing, Supply and Start-up of Electrical Systems.
- Design, Manufacturing and Supply of Electronics Projects and Equipment.
- Design, Procurement, Manufacturing, Supply and Start-up of Automation Systems.
- Design, Procurement, Manufacturing, Supply and Start-up of Control (PLC, VFD, HMI, SCADA) Systems.
- Design and Manufacturing of Panels
  (CPRI tested HT & LT panels, DBs, special design panels, C&I- PLC integrated with HMI, Drive/Servo, RIOs and SCADA, instrumentation JBs, etc. as required)
- Management and Start-up Services.
  (Installation, erection, commissioning, O&M and training for O&M staff)
- Repairing Services-VFDs.
- Electrical Liaisons', Retrofitting/Modification.
- Trading Products:- L&T Switchgear, L&T ISP, RR Cables, Neptune Ducati Capacitors, Neptune Bals Industrial Socket, Trinity Meters, Excel Meter Protection Relays.

- Offer CPRI tested Medium and Low Voltage Switchgear as per clients/projects specifications to meet all Industrial Standards with high efficiency, robust construction and compact design.

Segments they serve:-

- Textile
- Solar and Green Energy
- Steel
- Water and Waste Water
- Power
- Paper and Packaging
- Dyeing and Printing
- Diamond
- Machine Systems
- Process and Manufacturing
- Chemical
- Oil and Gas
- Pharmaceutical
- Railway
- Commercial Buildings

SYNOPSIS

Trucks exceeding the legal mass limits increase the risk of traffic accidents and damage to the infrastructure. They also result in unfair competition between transport modes and companies. It is therefore important to ensure truck compliance to weight regulation. New technologies are being developed for more efficient overload screening and enforcement. Weigh-in-Motion (WIM) technologies allow trucks to be weighed in the traffic flow, without any disruption to operations. Weigh-in-Motion systems can be readily built with a low cost and easily available equipments. These equipment are easy to set-up with higher efficiency. Weigh-in-motion data can be collected with its WIM equipment and in LabVIEW via Serial Communication, where the vehicle reports its own weight. Weigh-in-motion devices are designed to capture and record axle weights and gross vehicle weights as vehicles drive over a measurement site. WIM systems are capable of measuring vehicles traveling at a reduced or normal traffic speed and do not require the vehicle to come to a stop. This makes the weighing process more efficient. So, this allows legal drivers to pass without being stopped.

TABLE OF CONTENT

LIST OF TABLE

CHAPTER-1

PROJECT DEFINITION AND DESIGN SPECIFICATIONS

1.1 INTRODUCTION

Weigh-in-Motion (WIM) technology for measuring the weight of moving trucks is used by state highway agencies for monitoring pavement loadings. This technology has many advantages in terms of utility and efficiency over the traditional approach of static weighing. Because of the complexity of WIM systems, the implementation of such systems has been a gradual process in recent years [1].

The State Highway Agencies wants to improve the control of heavy vehicles, and make the control procedures more efficient. One aspect of this work is to improve weight controls. An important part of the project is the collection Weigh-In-Motion (WIM) data, with WIM equipment, and serial communication where the vehicle reports its own weight. The tests are focusing on data quality (WIM vs static weight), user interface and usability, procedure and need for calibration. If a vehicle is overloaded then a warning is shown on the control station. This allows legal drivers to pass without being stopped [2].

In recent years there has been great demand for weight data. Together with traditional traffic data this information is useful for different purposes such as:

- Control of heavy vehicles
- Road planning
- Road maintenance
- Bridge applications
- Traffic Safety
- Competitiveness

Vehicle weight can be measured in different ways. We often define the methods as:

- Static weight: The vehicle has to stop at a weight station.
- Low speed WIM: The vehicle is weighed while driving at 5–15 km/h.
- High speed WIM: The vehicle is weighed while driving at the desired speed [2].

1.2 IMPACT OF WIM SYSTEMS

By reducing the number of overweight vehicles on the roads and by encouraging changes in carrier behavior, the use of WIM systems consequently leads to interesting benefits. The literature review showed an absence of best practice consensus relative to the use of WIM data for evaluating the overall impact of removing overloaded trucks from the roads. The WIM systems were first developed to preserve the road pavement and infrastructure, and to reduce the costs for weight enforcement resources. There is a new tendency to evaluate other effects on the carrier behavior (e.g. reduction of the average amount of overweight and number of overloaded vehicles), the environment (e.g. reduction of harmful emissions, vibrations, noise and fuel consumption), traffic management (e.g. reduction of congestion and road closures), traffic safety (e.g. reduction of traffic risks and accidents) and costs (e.g. reduction of taxes and fees violation, increase of overweight violation capture rates) [2].

1.3 DESIGN SPECIFICATIONS

Table 1.3.1 List of the customer needs driving the product specification [3].

Need	Mapping No.
Weigh-in-motion is inexpensive	1
Weigh-in-motion is reliable	2
Weigh-in-motion is easy to set up	3
Weigh-in-motion is safe	4
Weigh-in-motion can handle small workpieces	5
Weigh-in-motion requires less operating time	6

Table 1.3.2 Product Design Specification [3]

Metric	Mapped No.	Units	Value
Power Source	3	VAC	230
Current Drawn	3	A	1
Set-up time	6	Min	< 45
Ground-fault protection	4	Binary	Yes
Cost	1	₹	< 2000
Can handle weight upto	5	Kg	3
Cycles of failure	2	Integer	< 50

Software used: - EAGLE for PCB SCHEMATIC and BOARD DESIGN.

CS+ for PROGRAMMING THE MCU, ADS1232, LCD and LED

LABVIEW for the MAIN PROJECT PROGRAMMING, SCALING and for OUTPUT DISPLAY

Hardware used: - LOADCELL of 6 kg

RS-485 for COMMUNICATION.

POWER SUPPLY ADAPTER for supplying PCB with 12V SUPPLY.

Components used:-R5F104BC (Microcontroller)

ADS1232 (24-Bit ADC Converter)

MAX-487 (RS-485 Communication)

16x2 LCD (For DISPLAY)

Connectors (For Power Supply, Programming and Communication)

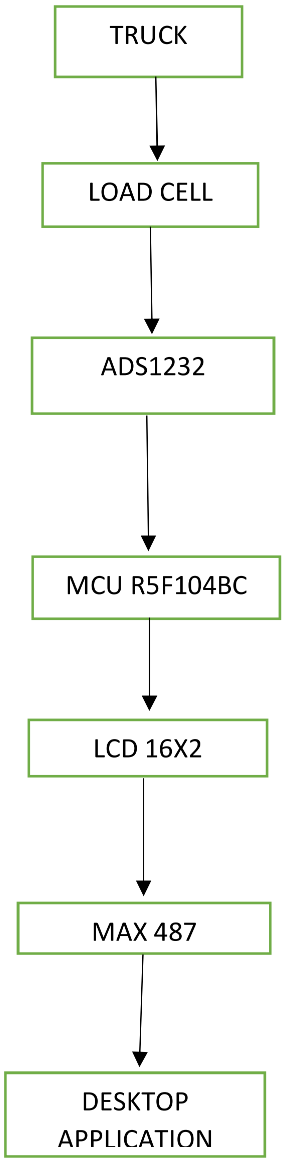
CHAPTER-2

BASIC THEORY

2.1 HOW IT WORKS?

Weigh-in-motion or weighing in motion (WIM) devices are designed to capture and record axle weights and gross vehicle weights as vehicles drive over a measurement site. Unlike static scales, WIM systems are capable of measuring vehicles traveling at a reduced or normal traffic speed and do not require the vehicle to come to a stop. This makes the weighing process more efficient, and, in the case of commercial vehicles, allows for trucks under the weight limit to bypass static scales or inspection [1].

The Block Diagram of the project is given below:-



When the truck is loaded on the sensor pad, the working starts.

Load-cell systems are the most popular technology used. Each cell is comprised of a durable material such as steel or concrete with one or more strain gauges attached to or embedded in it. A strain gauge consists of a wire (or wires) that transmits a mild electric current. As the cell is subjected to weight, the wire in the strain gauge is altered or compressed slightly. The change in the wire results in a difference in the resistance to the current passing through it. The signal from each cell is sent to a junction box, where sensors measure the variance in the current and calculate the amount of weight the scale is supporting [4].

The strain gauges in load cells can be either compression or tension based. A compression strain gauge is based on how much the cell compresses when pressure is applied, while a tension strain gauge is based on the slight change in shape of the cell caused by the weight [4].

A bending-plate system uses metal plates with strain gauges attached to them. As weight is applied to the scale, the plates are subjected to stress. The strain gauge on each plate measures the amount of stress and calculates the load required to cause it. The amounts from each gauge are added together to get the total for that axle [4].

A series of scales are used so that the entire truck can be weighed at once. The scales are typically connected to a single electronic controller that automatically combines the axle weights to get the gross weight [4].

From the Load Cell, the weight, which is in analog form, is converted to Digital form using Analog-to-Digital Converter. The Analog-to-Digital Converter used is ADS1232. The ADS1232 is precision 24-bit analog-to-digital converters (ADCs). With an onboard, Low-Noise PGA precision delta-sigma ADC and internal oscillator, the ADS1232 provide a complete front-end solution for bridge sensor applications including weigh scales, strain gauges and pressure sensors [5].

ADS1232 has two differential inputs/channels. Data rate is 80 SPS and gain selected is 128. Why? As we know the Load cell's electrical sensitivity= 2mV/V (pre-defined manufacturer data).If we consider excitation to be 5V then Full Scale Output Voltage is 10mV.In order to use the most linear portion of the load cell's span, only about two-thirds of this range would be used. The full scale output voltage would be about 6mV.To measure small signal changes within this 6mV full-scale range, so that highest achievable performance is attained [5].

ADS1232 has low-noise PGA (Programmable Gain Amplifier) with higher internal gain in order to magnify small output signal from load cell. Due to its excellent noise specification and front-end gain stage with maximum gain of 128 mV/mV. Due to this, Load Cell is directly interfaced to ADS1232 [5].

The data is then sent to Microcontroller. The IC used is R5F104BC belonging to RL78 by Renesas. It is 32 pin IC. It has Flash ROM memory of 32KB, Address Space of 1MB, Data Flash of 4KB and RAM of 4KB [6]. ADS1232 output pin is connected to MCU pin. Data rate output, Channel selection and Buzzer programming is all done in MCU. MCU output is display on 16x2 LCD.

The data that is sent to Microcontroller is in the form of ADC count. So, Scaling is performed to convert count in the form of weight. Scaling is done using the formula:

Since, load cell of 6 kg is used and full scale output of load cell is 10mV. The full scale input of ADS1232 is 39mV. By setting the 24 bits, the value in decimal is 8388607 which is corresponding to full scale input of ADS1232. By cross-multiplication, the value in decimal 215092.48 which is corresponding to full scale output of load cell. So, Weigh in grams is given by (215,092.48*ads-count)/6000. This is how scaling is done.

Then the output is loaded to desktop application i.e. LabVIEW using MAX487 IC which is connected to PC using RS-485 cable.

CHAPTER-3

DESIGN

3.1 HARDWARE SCHEMATIC

The Hardware Design is done in CadSoftEAGLE. EAGLE stands for Easily Applicable Graphical Layout Editor and is developed by CadSoft Computer GmbH. CadSoft Computer GmbH was acquired by Autodesk Inc. in 2016. EAGLE is a scriptable electronic design automation with schematic capture, printed circuit board layout, auto-router and computer-aided manufacturing features. I have made new library in the software. I have designed the Device, Package and Symbol of ADS1232.

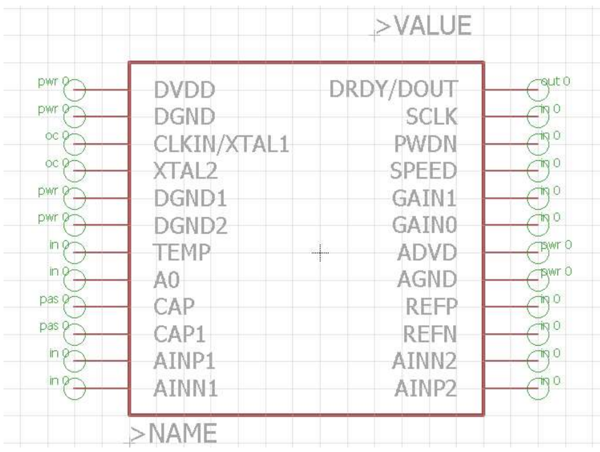


Fig:-3.1.1 Symbol of ADS1232

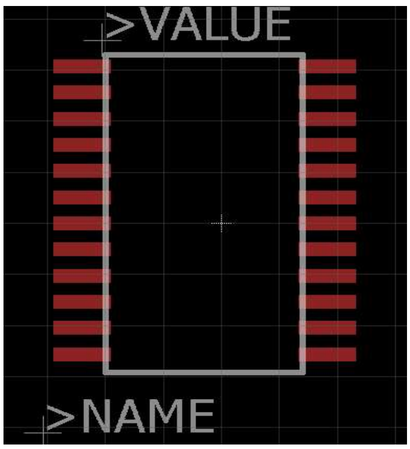


Fig:-3.1.2 Package of ADS1232

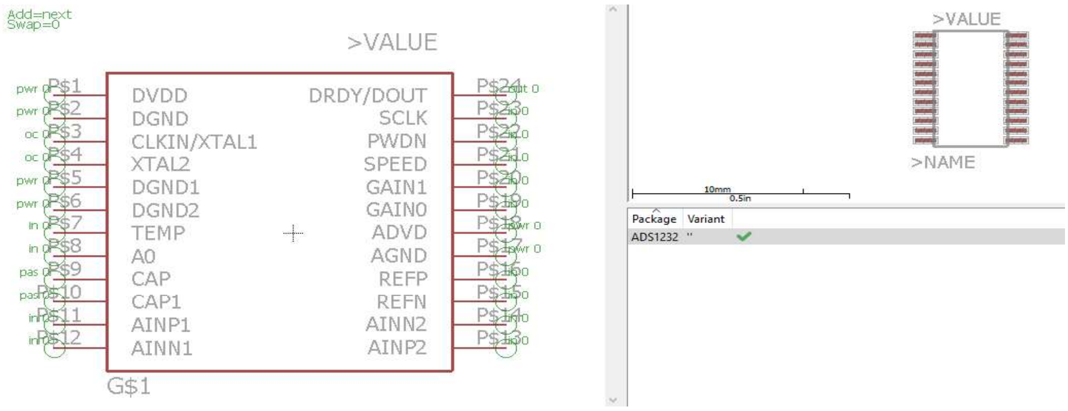


Fig:-3.1.3 Device and Connection Verification of ADS1232.

3.1.1 MICROCONTROLLER CONNECTIONS

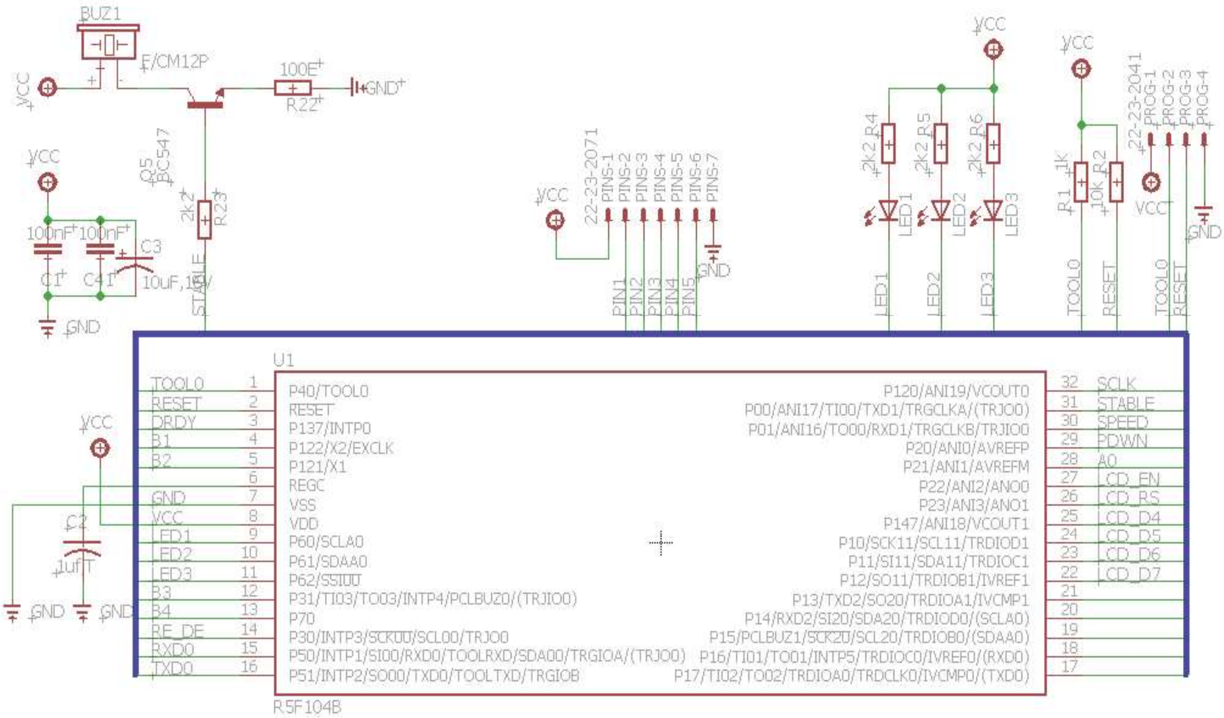


Fig:-3.1.1.1 R5F104BC Connections

The input to LCD is given by MCU. LED are connected to ensure that MCU is in run mode not in Power Down mode. Buzzer is connected to MCU to ensure that No Heavy Load Vehicles would pass. Speed, PWDN, DRDY are the output from ADS1232. RXD0 and TXD0 are used for transmitting and receiving data to/from Communication Port IC MAX487.Connectors are used for Programming, Communication and Power Supply.

3.1.2 ANALOG-TO-DIGITAL CONVERTER CONNECTIONS

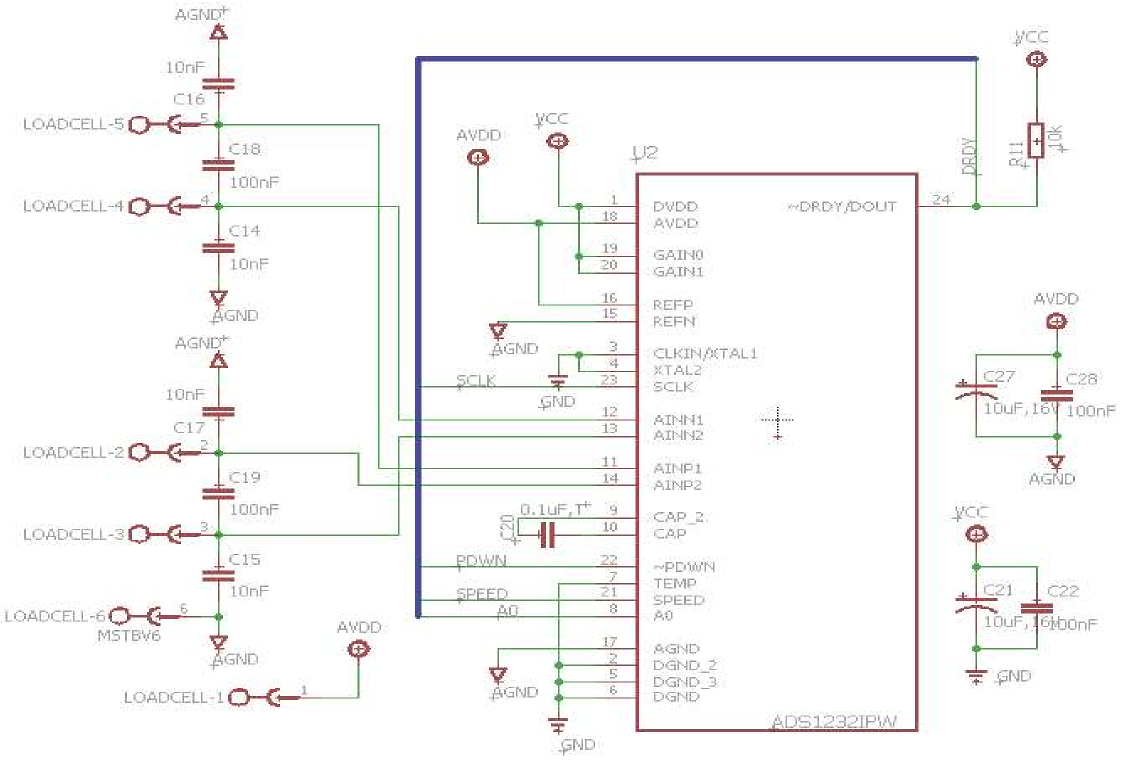


Fig:-3.1.2.1 ADS1232 Connections

Load Cell is interfaced with ADS1232 using Connectors. Gain0 and Gain1are connected because we have selected Gain as 128. SCLK takes 24 bit from ADS1232 and passes it to MCU. SPEED selected is 80 SPS which is connected using bus bar to MCU. A0 is channel select pin so it is also connected to MCU. ~DRDY/DOUT takes the input from ADC when it has falling edge. It take 1 bit in from ADC and outputs the last bit out to MCU.

3.1.3 POWER SUPPLY CONNECTIONS

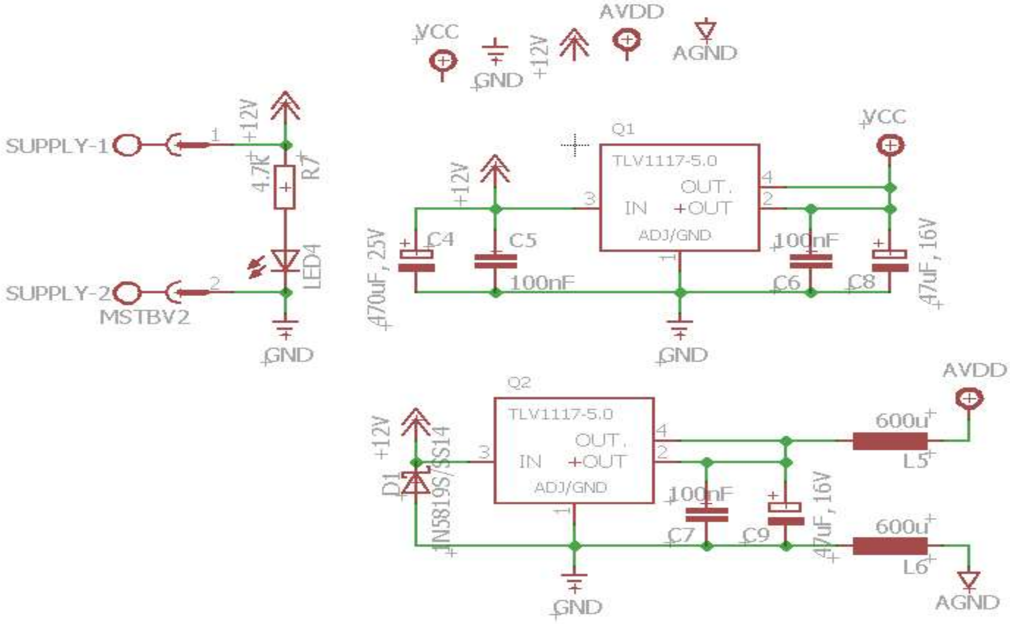


Fig:-3.1.3.1 Voltage Regulator Connections

Voltage regulation is very important point in making a circuit. It regulates the voltage upto 12V DC. It is supplied to PCB using Adaptor of 230V AC and 1Amp Current. The IC used here are Voltage Regulator IC.

3.1.4 LCD CONNECTIONS

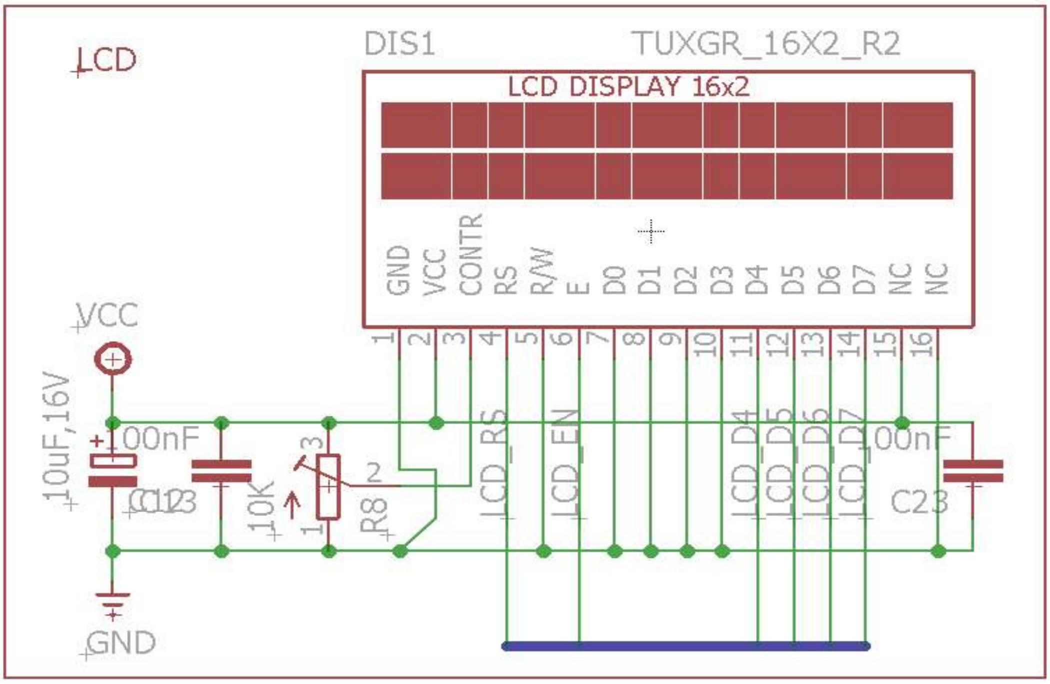


Fig:-3.1.4.1 LCD Connections

The LCD used here is 16*2 LCD in 4-bit Mode. It is connected to MCU using the Bus-Bar.

3.1.5 COMMUNICATION CONNECTIONS

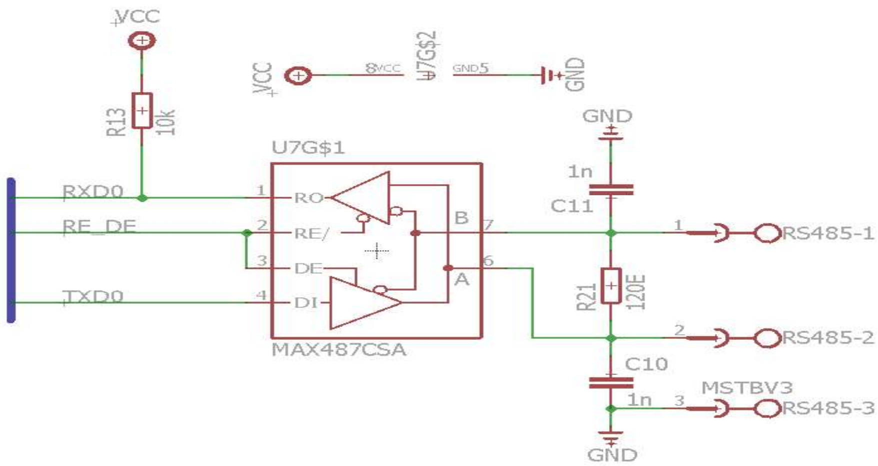


Fig:-3.1.5.1 MAX487 Connections.

MAX487 is used for communication purpose to send data to PC using RS-485 connectors. This is connected to MCU using Bus-bar for transmitting and receiving data.

3.1.6 PUSH BUTTONS

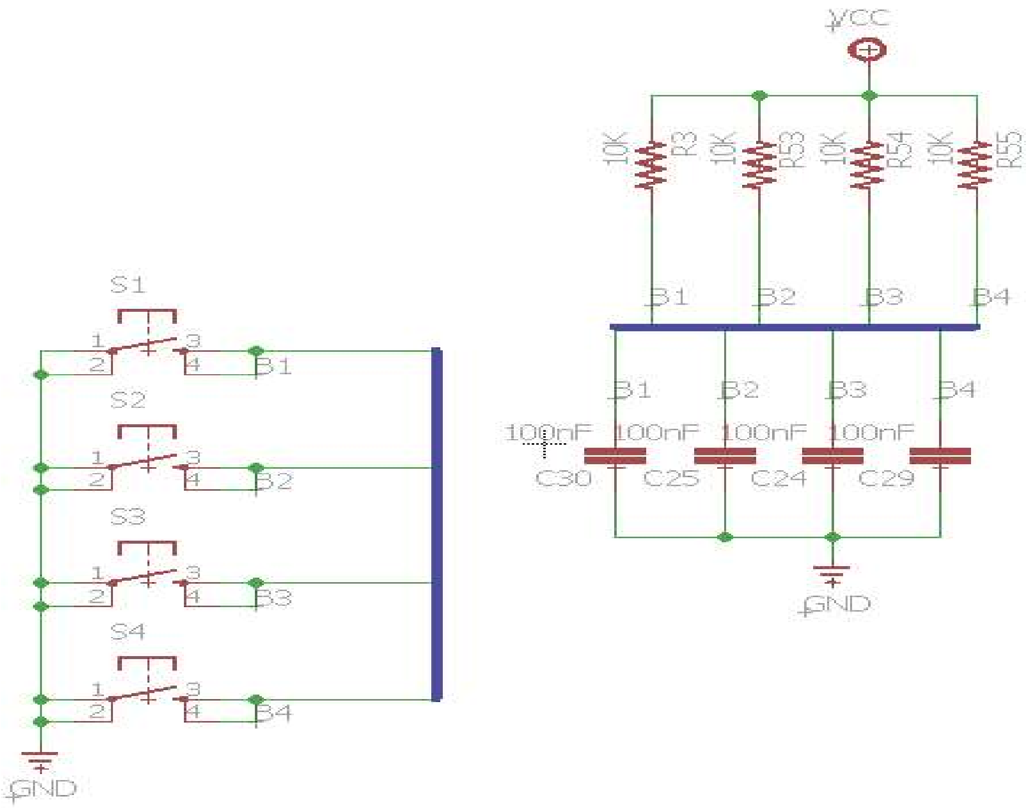


Fig:-3.1.6.1 Push Buttons Connections

These are used to control display on LCD like adding drop-down menu and adjust settings for future use.

3.2 DESIGN ON BOARD

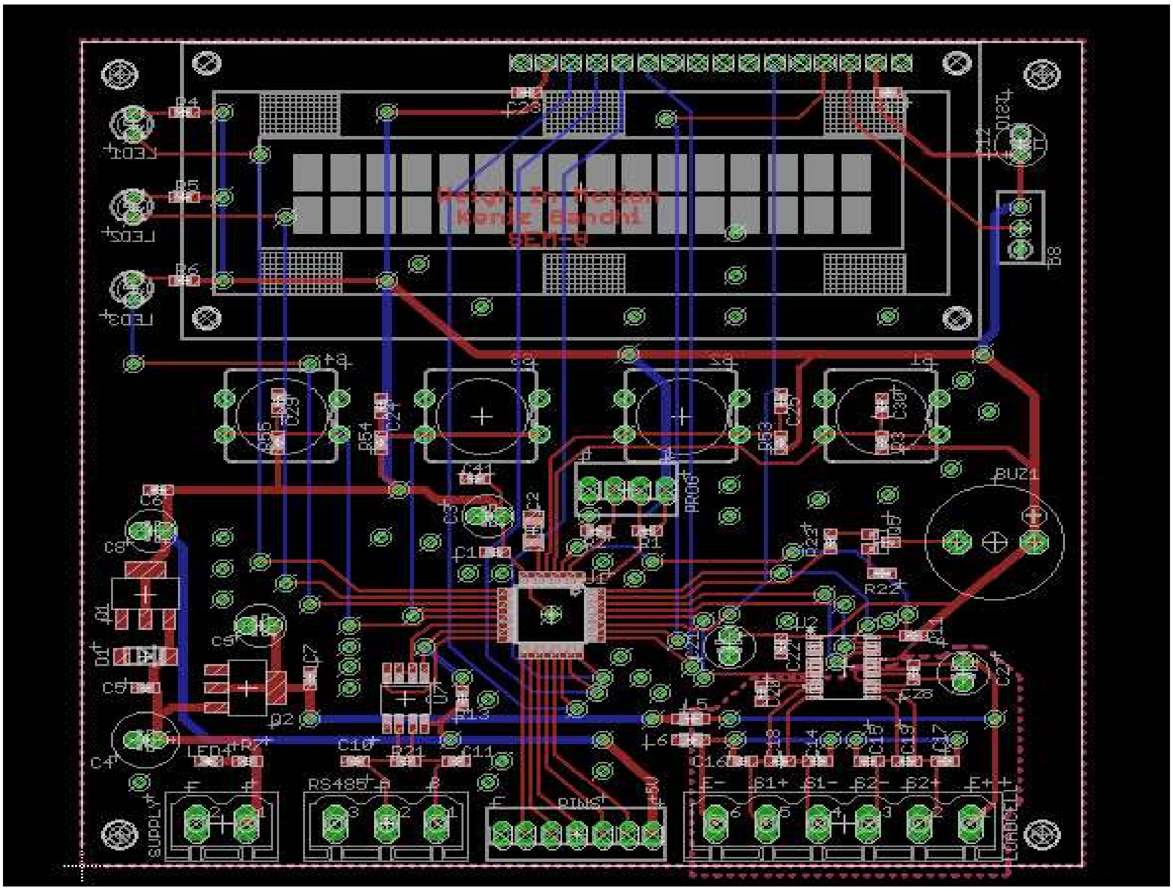


Fig:-3.2.1 PCB Board Design

3.3 SOFTWARE DESIGN

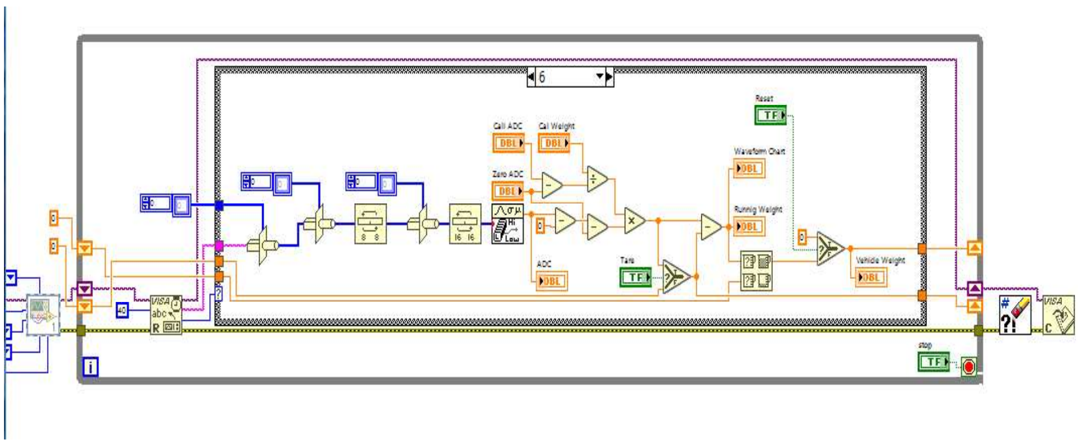


Fig: -3.3.1 Design on LabVIEW

When the program is loaded to LabVIEW, the scaling is done properly as I was unsuccessful in doing scaling in code and then the median value of data is loaded.

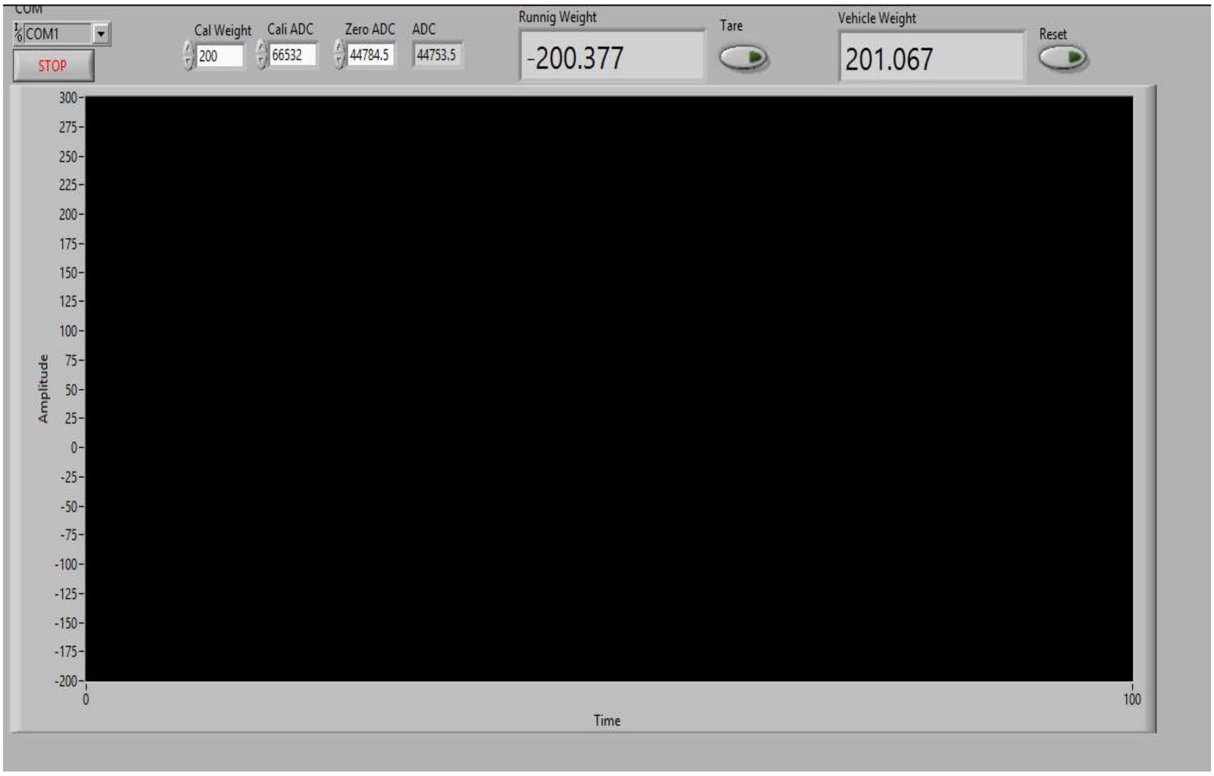


Fig:-3.3.2 Final Output is shown on this screen

CHAPTER-4

ALGORITHM

4.1 SOFTWARE USED

The CS+ integrated development environment provides simplicity, security, and ease of use in developing software through iterative cycles of editing, building, and debugging [7].

You can use the basic software tools for developing software for Renesas MCUs immediately after the initial installation. CS+ is also compatible with Renesas hardware tools including the E1 on-chip debugging emulator (sold separately), which facilitates advanced debugging. Abundant extensions and functions for user support ensure a dependable environment for all users [7].

4.2 INTERFACING MCU (R5F104BC) WITH ADC (ADS1232).

Initially RESET all these pins:

PWDN

A0,

SCLK

GAIN0, GAIN1

↓

INTIALIZE A VARIABLE CONV AS 0 AND

SET PWDN PIN

↓

CHECK FOR CONDITION IF DRDY PIN IS SET OR RESET IF SET THEN SET THE CONV VALUE.

↓

AGAIN CHECK FOR THE CONDITION THAT CONV IS 1 AND DRDY PIN IS RESET, IF IT IS SO THEN RESET DATA AND CONV VALUE

↓

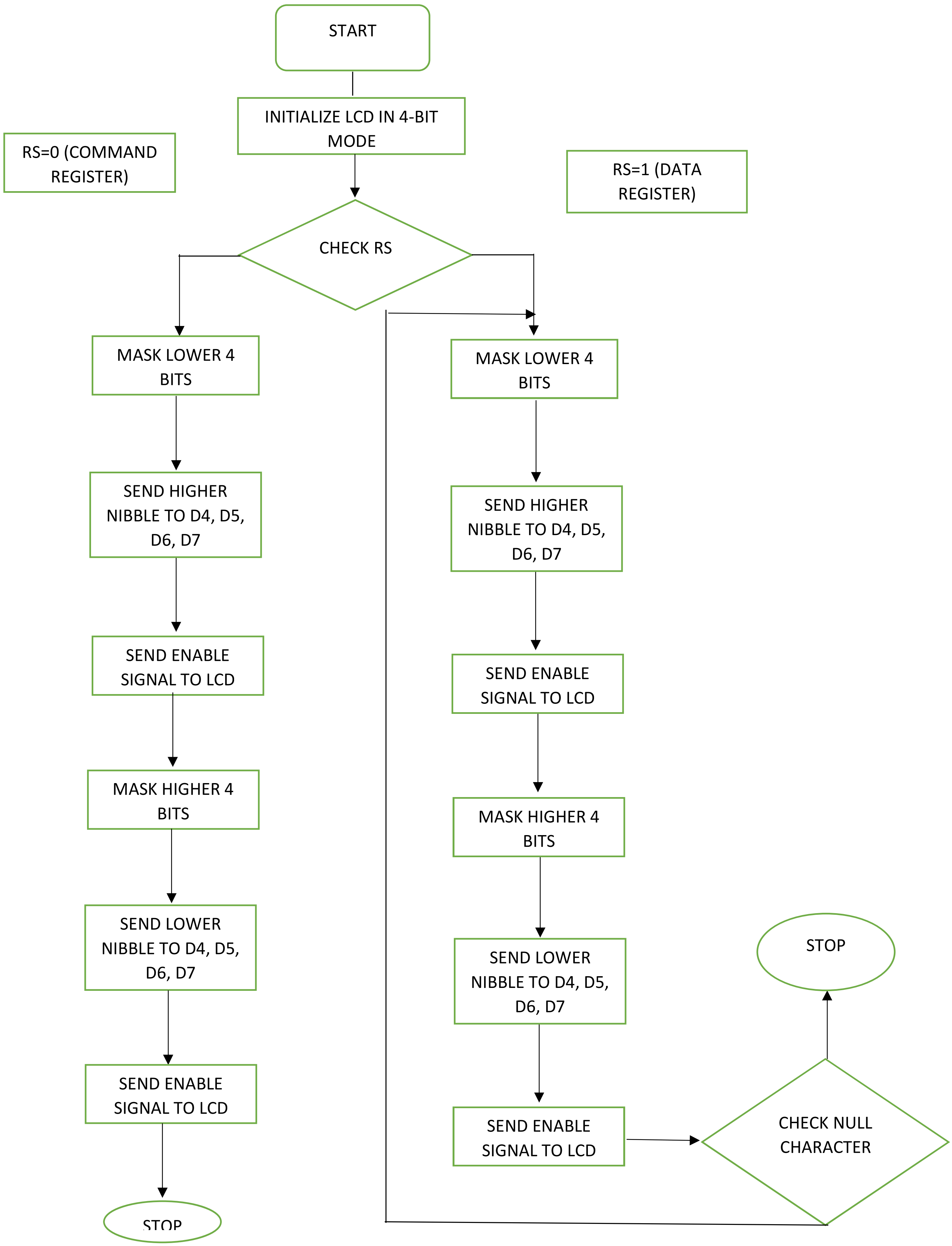
ALLOW THE 24 BITS TO PASS IN TO MCU, BY SHIFTING LEFT THE DATA AND FIRST SETTING THEN RESETTING THE VALUE OF SERIAL CLOCK.

↓

BY SETTING THE SERIAL CLOCK THEN THE BIT TEST ACTION IS PERFORMED.

4.3 INTERFACING LCD WITH MCU (R5F104BC)

Fig:-4.3.1 Successfully initialized the LCD.


START
INITIALIZE LCD IN 4-BIT MODE
RS=0 (COMMAND REGISTER)
RS=1 (DATA REGISTER)
CHECK RS
MASK LOWER 4 BITS
SEND HIGHER NIBBLE TO D4, D5, D6, D7
SEND ENABLE SIGNAL TO LCD
MASK HIGHER 4 BITS
SEND LOWER NIBBLE TO D4, D5, D6, D7
SEND ENABLE SIGNAL TO LCD
STOP
MASK LOWER 4 BITS
SEND HIGHER NIBBLE TO D4, D5, D6, D7
SEND ENABLE SIGNAL TO LCD
MASK HIGHER 4 BITS
SEND LOWER NIBBLE TO D4, D5, D6, D7
SEND ENABLE SIGNAL TO LCD
CHECK NULL CHARACTER
STOP

4.4 SUCCESSFULLY INTERFACED WITH DESKTOP APPLICATION

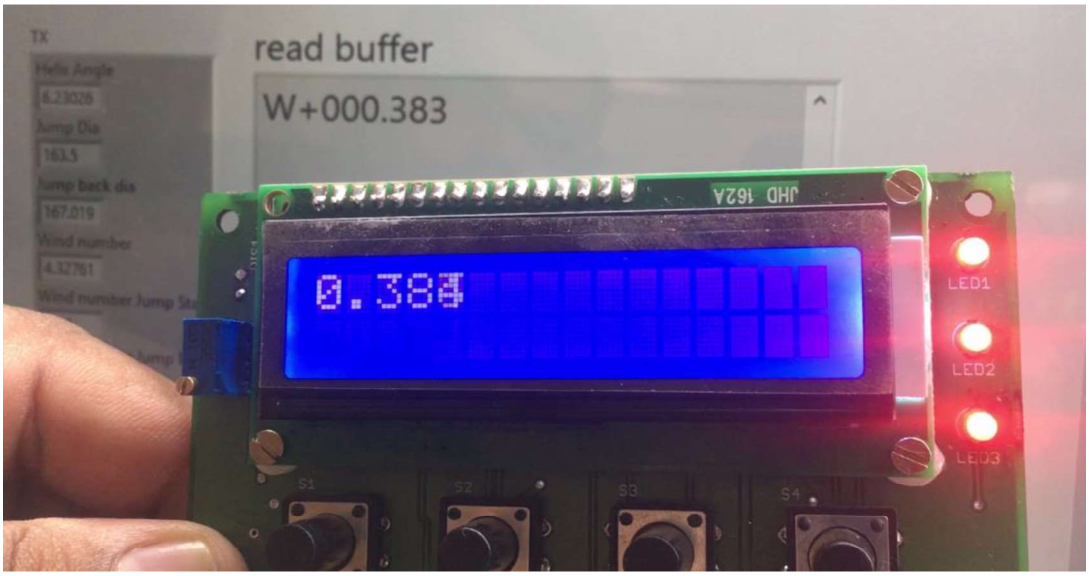


Fig:-4.4.1 Before scaling. Just to test

CHAPTER-5

OUTPUT

5.1 FINAL OUTPUT

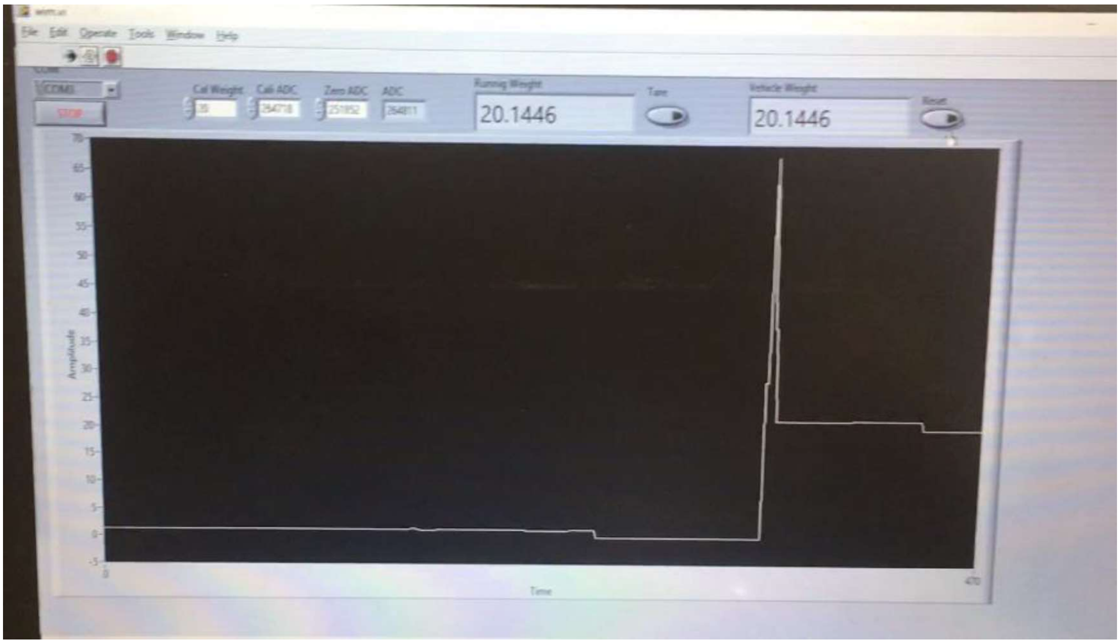


Fig: -5.1.1 Final Output on Desktop Application in form of Waveform also.

The test has to be performed in closed silent room and at normal temperature between 25-35 degree Celsius. Minimize the noise, even don't use fan while performing the experiment. This can cause fluctuations in weighing and can cause noise spikes. The Weighing Scale is to be handled with care else it can cause problems in scaling.

CHAPTER-6

6.1 APPLICATION

There are various practices around the world in using WIM for enforcement. High speed WIM systems are used for direct enforcement, with large tolerances to account for inaccuracies in the system. Tolerances of up to 30% were reported, but this may be acceptable if there are very large and frequent overloads [17].

In some countries, portable high speed WIM systems are used over short time periods to detect overloads, and then to perform static controls with portable scales. However, the accuracy of portable WIM systems is not very good, and thus the efficiency or pre-selection is low [17].

High quality high speed WIM systems are installed in pavements upstream of large weighing stations. These weighing stations – along motorways and highways – are equipped with low speed or static weighing systems as well as parking lots for enforcement. Many of these high speed WIM systems weigh very accurately, due to the effort put into ensuring the weigh pads are absolutely planar with the road surface to minimize vehicle dynamics — or bounce. The WIM system is generally used to identify potentially overloaded trucks and divert them to the weighing area. Depending on the traffic density and the local organization, the suspicious vehicles are chosen one by one; some sequences of vehicles are picked when one of them is detected by the WIM system. If the weigh station is not permanently manned, the WIM system only records statistics outside the enforcement sessions [17].

Coupling high speed WIM for pre-selection and low speed WIM for enforcement is already implemented in several countries. The process could be automated to become more efficient and to require less staff. That is also the challenge in countries where bribery is an issue. The automatic self-calibration of the HS-WIM system using the low speed data could also be improved and more automated [17].

6.2 LIMITATION

Weigh-in-motion suffers from a number of limitations. It requires staff and time to perform weighing. Staff is needed to select and decrease the speed of trucks to 30 km/hr. in the traffic flow, to perform the weighing operation on the control area, and to fine the violators and apply other penalties as needed. It is difficult to safely perform checks on heavily trafficked highways and motorways. With high traffic volume, and the increase on roads of heavy vehicles, weighing becomes ineffective and acts as a limited deterrent.

6.3 CONCLUSION

WIM is a useful tool to contribute towards more compliance with mass regulation. It has been used most successfully for nearly two decades. WIM has helped to reduce the number of overloaded trucks, and contributed to the more efficient and effective use of police officers' time. A reduction in overloaded trucks is also conducive to a reduction in crashes. There are still issues and challenges for WIM technology and application which require more research and development work. It is also essential to better disseminate knowledge and best practices, to exchange experiences, and carry out large scale common tests of WIM sensors and systems.

6.4 FUTURE SCOPE

6.4.1 Bridge (B-) WIM

The concept of bridge (B-) WIM was introduced. It uses instrumented bridge parts (*e.g.* deck, slab or beam) to measure the strains induced by the moving vehicle loads crossing the bridge. It then calculates the axle and vehicle loads, using the calculated or measured transfer function (load to strain) called an influence line (1-D) or influence surface (2-D). In a B-WIM system, the bridge is used as a large scale calibrated to weigh axles and vehicles [17].

Fig:-6.4.1.1 Bridge Weigh-in-Motion [17]

6.4.2 Video-WIM and Automatic Vehicle Identification (AVI)

It involves coupling a WIM system to a video camera with OCR — automatic license plate number recognition. The video-WIM system is installed a few kilometers upstream to a weighing area. If a vehicle passes with an overloaded axle, an overload on the gross weight, or is even over speeding, pictures of the whole vehicle – including number plate – are sent to the compliance officers at the weighing area. The vehicle is then stopped on the weighing area (*e.g.* if located after a toll barrier) or is directed to the weighing area [17].

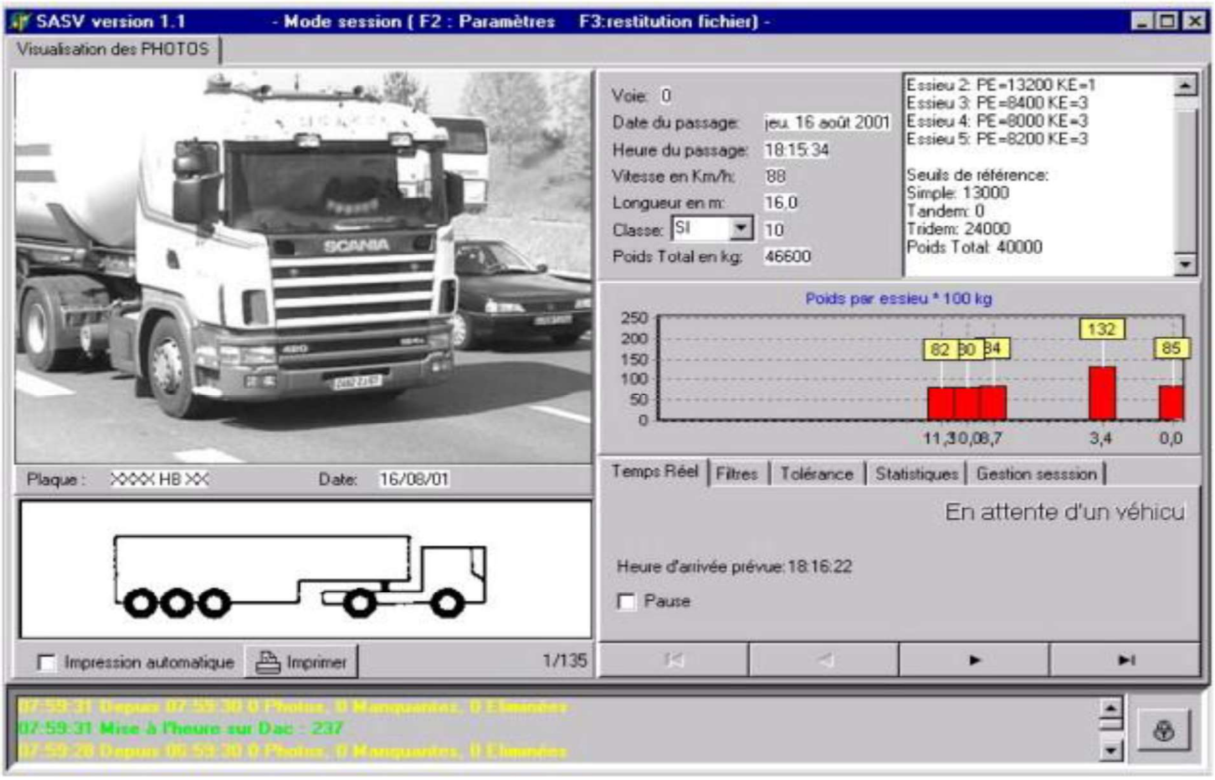


Fig:-6.4.2.1 Video-WIM and AVI [17]

REFERENCES

[1].http://www.ctre.iastate.edu/research/wim_pdf/Wimsummary.htm

[2].http://www.sciencedirect.com/science/article/pii/S2352146516302174

[3].http://www.me.umn.edu/courses/me4054/assignments/pds.html

[4].http://science.howstuffworks.com/engineering/civil/question626.htm

[5].DATASHEET OF ADS1232 BY TEXAS INSTRUMENTS (Page-1)

[6].DATASHEET OF RENESAS RL78 IC (Page 1 and 36).

[7].https://www.renesas.com/en-in/products/software-tools/tools/ide/csplus.html

[8].https://www.maximintegrated.com/en/app-notes/index.mvp/id/4701

[9].http://www.larsentoubro.com/electrical-automation/products-services/products/automation-products-systems/industrial-automation-products/programmable-logic-controller/

[10].ABOUT LXSoft HELP from the LXSoft Software

[11]. Campbell, Sylvester J. (1987). Solid-State AC Motor Controls. New York: Marcel Dekker, Inc. pp. 79–189. ISBN 0-8247-7728-X.

[12].http://www.larsentoubro.com/electrical-automation/products-services/products/automation-products-systems/x2000-series-ac-drives/sx2000/

[13].https://www.techopedia.com/definition/12378/switched-mode-power-supply-smps

[14].PPT (SLIDE NO.2) ON TOPIC-4_3_Sensors_Position (INCREMENTAL SHAFT ENCODER)

[15].https://www.electrical4u.com/induction-motor-types-of-induction-motor/

APPENDIX

APPENDIX-A

ABBREVIATIONS

- WIM - Weigh-in-Motion
- PLC - Programmable Logic Controller
- AVI - Automated Vehicle Identification
- B-WIM – Bridge Weigh-in-Motion
- OCR – Optical Character Reader
- HS-WIM – High Speed Weigh-in-Motion
- MCU – Microcontroller Unit
- ADC – Analog-to-Digital Converter
- LCD – Liquid Crystal Display
- LED – Light Emitting Diode
- PWDN – Power Down
- DRDY – Data Ready
- SCLK – Serial Clock
- RS – Register Select
- R/W – Read/Write
- EN – Enable
- IC – Integrated Circuit
- PCB – Printed Circuit Board
- SPS – Samples per Second
- PGA – Programmable Gain Amplifier
- VFD – Variable Frequency Drive
- HMI – Human Machine Interface
- SMPS – Switch Mode Power Supply
- AC – Alternating Current
- DC – Direct Current
- PPR – Pulse Per Revolution

- PID - Proportional–Integral–Derivative

APPENDIX-B

CODE

ADS1232

C file

```
#include "r_cg_macrodriver.h"
#include "board.h"
#include "common.h"
void ads1232 (void)
{
        int i;
int convert = 0;
int data;
delay_ms(10000);
PWDN_ON;
if( DRDY == 1)
convert = 1;
if((convert == 1) && ( DRDY == 0))
{
data = 0;
convert = 0;
 for(i = 0; i < 24; i++)
    {
    SCLK_ON;
    delay_ms(1);
    shift_left(&data, 1, DRDY);
        SCLK_OFF;
```

```
        delay_ms(1);
        }
     delay_ms(1);
     SCLK_ON;
     delay_ms(1);
     delay_ms(1);
     for(i = 24; i > 0; i--)
        {
         if(bit_test(data, i) == 1)
           printf("1");
         else
           printf("0");
        }
}
```

Header file

```
#ifndef ADS1232_H
#define ADS1232_H
                void ads1232(void);
#endif
```

TIME DELAY FUNCTION

C File

```
#include "r_cg_macrodriver.h"
#include "common.h"
unsigned long int millisecond=0;
  void update_ms( void)
                {
                        millisecond++;
                }
```

```
void delay_ms( unsigned long int ms)
    {
        unsigned long int ms_stamp=millis();
        while( (millisecond-ms_stamp)<ms)
        {
            NOP();
        }
    }
unsigned long int millis(void)
    {
        return millisecond;
    }
```

Header File

```
#ifndef COMMON_H
#define COMMON_H
    void delay_ms( unsigned long int ms);
    void update_ms( void);
    unsigned long int millis(void);
#endif
```

PUSH BUTTON PROGRAM

C File

```
#include "r_cg_macrodriver.h"
#include "board.h"
#include "button.h"
void button(void)
{
    if(B1==0)
```

```
        {
                LED1_ON;
        }
        if(B2==0)
        {
                LED2_ON;
        }
        if(B3==0)
        {
                LED3_ON;
        }
}
```

Header File

```
#ifndef BUTTON_H
#define BUTTON_H
                void button(void);
#endif
```

LCD PROGRAM

C File

```
#include "r_cg_macrodriver.h"
#include "lcd.h"
#include "board.h"
#include "common.h"
void data_write(void)
{
LCD_EN_ON;
delay_ms(2);
LCD_EN_OFF;
```

```
}

void data_read(void)
{
LCD_EN_OFF;
delay_ms(2);
LCD_EN_ON;
}

void LCD_NibbleHSB(unsigned char val)
 {
        LCD_D4=val&0b10000000;
        LCD_D5=val&0b01000000;
        LCD_D6=val&0b00100000;
        LCD_D7=val&0b00010000;
}
 void LCD_NibbleLSB(unsigned char val){
        LCD_D4=val&0b00001000;
        LCD_D5=val&0b00000100;
        LCD_D6=val&0b00000010;
        LCD_D7=val&0b00000001;
}
void send_command(unsigned char cmd)
{

LCD_RS_OFF;
LCD_NibbleHSB(cmd);
data_write(); // give enable trigger
```

```
LCD_NibbleLSB(cmd);
data_write();

}

void send_data(unsigned char data)
{

LCD_RS_ON;
LCD_NibbleHSB(data);
data_write(); // give enable trigger
LCD_NibbleLSB(data);
data_write(); // give enable trigger
}

void send_string(char *s)
{
while(*s)
{
send_data(*s);
s++;
}
}

void lcd_init(void)
{
send_command(0x33);
send_command(0x32);
```

```
send_command(0x28); // 4 bit mode
send_command(0x0E); // clear the screen
send_command(0x01); // display on cursor on
send_command(0x06); // increment cursor
send_command(0x80); // row 1 column 1
}
```

Header File

```
#ifndef LCD_H
#define LCD_H
        void data_write(void);
        void data_read(void);
        void LCD_NibbleHSB(unsigned char val);
        void LCD_NibbleLSB(unsigned char val);
        void send_command(unsigned char cmd);
        void send_data(unsigned char data);
        void send_string(char *s);
        void lcd_init(void);
        #endif
```

Main Program

```
void main(void)
{
lcd_init();
send_string("adc_bit");
send_command(0xC0);
}
```

MAIN HEADER FILE

```
#ifndef FILE_BOARD_H
```

```
#define FILE_BOARD_H
#include "r_cg_macrodriver.h"
#define STABLE P0.0
#define SPEED P0.1
#define LCD_D5 P1.0
#define LCD_D6 P1.1
#define LCD_D7 P1.2
#define PWDN P2.0
#define A0 P2.1
#define LCD_EN P2.2
#define LCD_RS P2.3
#define RE_DE P3.0
#define B3 P3.1
#define RXD_0 P5.0
#define TXD_0 P5.1
#define LED1 P6.0
#define LED2 P6.1
#define LED3 P6.2
#define B4 P7.0
#define SCLK P12.0
#define B2 P12.1
#define B1 P12.2
#define DRDY P13.7
#define LCD_D4 P14.7
#define PWDN_OFF PWDN=0
#define A0_OFF A0=0
#define SCLK_OFF SCLK=0
#define PWDN_ON PWDN=1
```

```
#define SCLK_ON SCLK=1
#define LCD_EN_ON LCD_EN=1
#define LCD_EN_OFF LCD_EN=0
#define LCD_RS_ON LCD_RS=1
#define LCD_RS_OFF LCD_RS=0
#define LED1_ON LED1=1
#define LED1_OFF LED1=0
#define LED2_ON LED2=1
#define LED3_ON LED3=1
#define LED2_OFF LED2=0
#define LED3_OFF LED3=0
#define STABLE_ON STABLE=1
#define STABLE_OFF STABLE=0
#define B2_ON B2=1
#define B2_OFF B2=0
#define B1_ON B1=1
#define B1_OFF B1=0
#define B3_ON B3=1
#define B3_OFF B3=0
#define B4_ON B4=1
#define B4_OFF B4=0
#endif
```