IOT WEIGHING SCALE

A DISSERTATION

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE
OF

MASTER OF TECHNOLOGY
IN
COMPUTER SCIENCE & ENGINEERING

Submitted by:
SUNIL KUMAR SWAMI
2K16\CSE\17

Under Supervision of:
Mr R.K. YADAV
(Assistant Professor)

COMPUTER SCIENCE & ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
(Bawana Road, Delhi-110042)

JUNE 2018

CANDIDATE’S DECLARATION

I, Sunil Kumar Swami, 2k16/CSE/17 student of M.Tech (Computer Science & Engineering), hereby declare that the project dissertation titled “**IOT WEIGHING SCALE**” which is submitted by me to the Department of Computer Science & Engineering, Delhi Technological University, Delhi in partial fulfillment of the requirement for the award of the degree of Master of Technology, is original and not copied from any source without proper citation. This work has not previously formed the basis for the award of any degree, Diploma Associateship, Fellowship or other similar title or recognition.

Place: Delhi (Sunil Kumar Swami)

Date: (2K16/CSE/17)

CERTIFICATE

I hereby certify that the Project Dissertation titled “**IOT WEIGHING SCALE**” which is submitted by **Sunil Kumar Swami**, 2K16/CSE/17 Computer Science & Engineering, Delhi Technological University, Delhi in partial fulfillment of the requirement for the award of degree of Master of Technology, is a record of the project work carried out by the student under my supervision. To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Place: Delhi

Date:

Mr R.K. Yadav
Assistant Professor
Department of Computer Science & Engineering
Delhi Technological University

ACKNOWLEDGEMENT

First and foremost, I would like to express my deep sense of gratitude and indebtedness to my supervisor **Prof. R.K. Yadav** for his invaluable encouragement, suggestions and support from an early stage of this research and providing me extraordinary experiences throughout the work. Above all, his priceless and meticulous supervision at each and every phase of work inspired me in innumerable ways.

I specially acknowledge him for his advice, supervision, and the vital contribution as and when required during this research. His involvement with originality has triggered and nourished my intellectual maturity that will help me for a long time to come. I am proud to record that I had the opportunity to work with an exceptionally experienced Professor like him.

SUNIL KUMAR SWAMI

2K16/CSE/17

ABSTRACT

In current time, the calculations of population of bee for honey strictly increase. So that's why here i am going to discuss that how i can get the weight of bee population for honey. I am taking two kinds of technology like bluetooth and wi-fi. One more is hardware design weighingmachine.so i can get the bee population by using bluetooth and wi-fi module .Bluetooth module and wi-fi module are attached in weighing machine and all are controlled by programming. Bluetooth can measure the weight in small range but wi-fi will work as cloud service in which we can calculate the weight in every 15 second by python programming. Also i got the database server to store all intermediate results and also i can delete all results. There is load cell by using, i can measure up to 5 kg weight. Firstly this weigh goes to ADC (analogue to digital converter) to convert the weight into digital weight, after that is weight will go to microcontroller to control all weight to the led, wi-fi module, bluetooth module. here i use power plug source to get the electricity.

Bluetooth shows only weight is bee population but wi-fi service does not only shows the result but also it stores in database server in every 15 seconds that's why i do not have to go for checking weight to reach at particular place for our goal weight.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLE

LIST OF ABBREVIATIONS

S No	Abbreviated Name	Full Name
1	PCB	Printed circuit Board(PCB)
2	WSN	Wireless Sensor Networks (WSN)
3	IOT	Internet of Things (IOT).
4	SPM	Signal Processing Module (SPM)
5	RAM	Read-Write Memory (RAM)
6	ROM	Read-Only Memory (ROM)
7	GPR	General Purpose Registers (GPR)
8	SFR	Special Function Registers (SFR)
9	MCBEWB	Microcontroller Based Electronic Weighing Balance (MCBEWB)
10	SPI	Serial Programmable Interface (SPI)

CHAPTER -1

DIGITAL WEIGHING MACHINE

1.1 OVERVIEW

Weighing machine is very useful product. It helps us in checking our weight as well as other product's weight. Moreover, Weighing machine are used in many industrial and commercial applications. Without a weight machine it is not possible to know the exact mass of anything. There are two types of weighing measuring system called analog weighing machine and digital weighing machine. The analog weighing system is very erroneous. In digital weighing machine, we don't have to face any problem. Only viewing the display we come to know the exact weight of the product.

In digital weighing machine, we interface with the digital world. In this project, i come to know the basic principle of digital weighing machine and know how to make a digital weight machine easily and cheaply. This project basically helps us to compete with Manufacturer Company of digital weighing machine in future.

Now a day precise measurement and storage of weight is one of the most important activities in industries. The mechanical weighing machines are now replaced by the electronics weighing machine as electronic weighing machines are smart with the advantages like accuracy, reliability, and wide range.

The Electronics weighing bridges are comparatively light weight and easy to operate with direct display. Earlier electronic weighing machine were designed using DPM, Microprocessor and Personal computer. The disadvantage of DPM type weigh machine has no facility to store data internally. Microprocessor and Personal computer based system cost is very high. To remove this drawback microcontroller based weight machine is designed.

This project presents the software and hardware design, results and conclusion. We have seen weight machines at many shops, where machine displays the weight just by placing any item on the weighing platform. So here i am building the Weighing machine to weight goods in crate itself by using arduino and Load cell**s**, having capacity of measuring upto 5kg. This limit can be further increased by using the Load cell of higher capacity.

The Internet of Things (IOT) is the network of physical objects—devices, vehicles, buildings and other items which are embedded with electronics, software, sensors, and network connectivity, which enables these objects to collect and exchange data. The Internet of Things allows objects to be sensed and controlled remotely across existing network infrastructure, creating opportunities for more direct integration of the physical world into computer-based systems, and resulting in improved efficiency, accuracy and economic benefit [1].

A home IOT system essentially integrates sensors into a network so that functionality that is required on a daily basis can be completed without human intervention. The design criteria will mainly have the following parameters:

1. The sensors that we choose for specific functionality, such as mechanical/electrical sensors for signal acquisition.

2. The network uses wi-fi module to get and store data into database server. Also uses local networks such as Bluetooth.

3. The processing capability requirements for the specific functionality. This criterion is to check on how well data processing can be done by choosing a specific processor.

4. The power criteria for the system. The system power consumption should be minimal – hence a power efficient system is to be designed.

5. Additional interfaces for control mechanisms such as manual overrides, calibration or security modes.

An example, IOT system is the Smart weighing system. This system will perform the following functionalities [2].

1. Bees weight is monitored automatically and if the crate is about to full (thresholds can be set based on capacity of crate), then automatically data goes to java program and data will be send on cloud.

2. The system is flexible enough to bring it to the notice of a user using a Mobile App and via this Mobile App the user can check the status of goods.

1.2 COMPONENT USED IN THIS PROJECT

To do the project i had to collect some components. They are given below-

- Load Cell
- Stand of iron to set up Load Cell
- Microcontroller (PIC16F877A)
- Op-Amp (AD620 & LM358)
- Printed circuit Board (PCB)
- Power supply
- Concomitant: Resistors, Capacitors, Connecting Wires.
- Arduino software to do the code

1.3 FEATURES

1. Easy to operate.
2. High speed of response, low settling time and hence rapid weighing.
3. Compact and small in size.
4. Both analog and digital output with printing facility is available.
5. High accuracy even under most adverse environmental conditions such as dust, corrosive, heat and cold.
6. High stability and ruggedness.
7. High resolution.
8. Long equipment life.
9. No technical skill is required to operate.
10. Greater reliability.

1.4 PROBLEM FACED DURING THIS PROJECT

Mainly we faced problem in hardware part.

- By checking the load cell wires again and again we had to find out which of the two are inputs.
- Making of stand of iron was also very difficult part.
- When we use LM358, the output was highest fixed on 3.87 volt. It was little bit difficult to find out the reason behind it. After some analysis we found out that because of the less power supply of 5 volt we were not getting the exact amplified voltage. When we increase the voltage, we got our output from the LM358.
- Moreover we could not make sub tractor circuit for avoiding the leakage voltage only because of time shortness.
- We also burned one of our microcontrollers because of excessive power supply. As because PIC16F877A is very sensitive we have to very careful of its use.

1.5 APPLICATION OF THIS PROPOSED WORK

Beekeeping (Apiculture) can be dated as far back as primitive human culture. Prehistoric wall paintings show humans procuring honey from bee colonies. At the dawn of apiculture most of the hive's raw products i.e. honey and wax, were utilized [1]. However, in modern times honey production is considered a by-product of the western honey bee (*Apis mellifera*), as the pollination of plants is by far their most important activity [2]. Up to 79% of the human food supply today is dependent on pollination, and the honey bee is the most widespread and active pollinator animal worldwide.

Beehive weight is a key indicator of the productivity and condition of the honey bee colony, studies on how to effectively determine beehive weight have been carried out since the birth of apiculture. The USA Department of Agriculture released a document in 1925 outlining the effects of the weather on apiaries. It concluded that factors like ambient temperature, hours of sunshine, variation in temperature, and humidity can affect the weight of the beehive. The most commonly utilized method of weighing is to place a mechanical balance under the beehive, then adjust the scale as the weight changes [3].

A simple rule of thumb developed by beekeepers, is to use a basic tension scale e.g. a luggage weight scale, tilt the beehive on one side and double the value determined. The objective of weighing the colony during the honey production season is to determine when the hive's honey stores have reached maximum capacity, and then harvest them. This is difficult to achieve with the traditional, inaccurate weighing methods. The main difficulty surrounding hive weight measurement is that to gain an accurate picture of the hive status it is necessary to weigh several tens of kilograms but to an accuracy of tens of grams.

Recent advances in the areas of embedded sensing, miniaturization, wireless communications, low power operation, and energy harvesting contribute to form the technology known as ”Wireless Sensor Networks” (WSN). WSN are an integral part of the emerging concept of the Internet of Things (IOT). WSN are fast becoming an integral part of everyday life in many areas including healthcare [4], smart homes [5], and security [6]. They have become a key research area in academia and industry, with many off-the-shelf solutions available.

In this project a solution for remotely determining the weight of a beehive is presented. This solution utilised a load cell platform weighing scale system, enabled with WSN technology and an ultra-accurate analogue to digital converter, for remote collection of precision weight data. A temperature and humidity sensing node from an existing beehive monitoring system was utilised to provide error compensation for the load cells, which are susceptible to drift error as a result of these parameters [7]. These nodes were designed to integrate into a larger sensor network for monitoring the health of a beehive [8]. The final demonstration node can measure weight changes in the range of just tens of grams, making it well suited for determining when the hive has produced its maximum amount of honey as described above.

CHAPTER -2

RELATED WORK

An example of an early scientific study into beehive weight was carried out in 1977 [9]. A high precision mechanical scale was used and modified for use, including protecting the mechanical components from rust and creating a jig to translate the weight of the beehive onto the balance. An elaborate vision system was used to view the changing values. This was one of the earlier implementations of non-invasive weight measurement i.e. the scale was integrated into the system, rather than attached at selected time periods. The scale became part of the beehive. The scale had an accuracy of about 0.5 kg and provided reliable data for the research groups study. Researchers travelled to the beehive and recorded the weight on an hourly basis, over a period of 3 years.

Advances in the development of load cells, pressure sensors and strain gauges have made these technologies more attractive for use in weight measuring applications. Load cells have become a standard in beehive weighing systems. In the application of a single, robust load cell used to weigh an entire beehive [10], the maximum weight the load cell can accommodate is 200 kg, which satisfies the requirements of an average sized beehive. The jig used to translate the weight of the beehive to the load cell allows for uniform weight distribution and can cope should an imbalance of weight occur in the beehive.

Another application of load cells can be seen in the Arnia Scale [11]. The concept of the scale is similar to the previous model, but utilizes a more minimal design. The system uses 4 load cells, one in each corner, to balance the weight load. The design has taken into account greater ventilation of the beehive and allows any hive debris to fall directly to the ground and prevent a build up at the base of the beehive. The overall height adds 35 mm to the overall height of the beehive. The scale has a working capacity of 150 kg, measuring weight in increments of 10 g. The system of strain gauges adds a layer of complexity to the design of the scale. Previously, beehive weight measurement depended greatly on researchers traveling to the site of an apiary and manually recording the value displayed on the mechanical balance [9]. Recordings were often required on an hourly basis, which was cumbersome and required additional attention. Recent development in technology has significantly reduced the need for site visits. The advent of data logger technology has removed the necessity to manually monitor the weight of hives as it can automatically record the changes over time. The accumulated data can then be collected at less frequent intervals.

For completely remote observation, significant strides have been made to allow beekeepers to check on the condition of their beehive. Some systems, such as Arnia, have incorporated wireless components that are able to send data to a singular hub. These data are then accessible from various devices such as PC's, tablets, or smart phones [12]. Such applications provide the necessary information to prevent any sudden loss of bees in the beehive, for example through theft or disease.

The main constraint in achieving higher accuracy and in increasing throughput rate of the passing products is a superimposed noise on a useful signal from the weighing system. The common sources are: (1) constant values (2) the noises: electromagnetic pick-up, power harmonic, thermally unstable circuits and the gain programmable by software (through PWM's output of the microcontroller) (3) strain-gage based sensor (i.e., load cells) particularly sensitive to vibration. The common sources are amplified before transferred into the A/D converter, and therefore the noise is amplified and an error is introduced in the system.

A signal processing module (SPM) acquires the electrical signal from the weighing device and estimates a value of weight for the passing product as its output. The two main aims for improvement are increasing the speed of weighing and achieving good measurement accuracy. Improvement in SPM that provides any one or both aims brings significant benefit to the overall static weighing system [13]. Owing to the physical characteristics, when the load cell detects the external force, it produces a weak current. The weak current is passed through AD7730 that has the PGA (Programmable Gain Amplifier), A/D converter and the two stage digital filter into digital signal [1][2]. The micro-controller (PIC/8031) transfers the detected digital signal to the communication unit and the display unit according to the procedure set on the memory [5].

In system, the amplitude spectra reveal the significant noise components of the weighing system. The first peak is centered at approximately 2.2 Hz, and the second and third peaks appear to be integer multiples of this noise. Hence we can assume that there are no significant harmonic components beyond 40 Hz [14]. Any sampling rate selected must therefore be greater than 80 Hz.

The scale has a working capacity of 150 kg, measuring weight in increments of 10 g. The system of strain gauges adds a layer of complexity to the design of the scale. Previously, beehive weight measurement depended greatly on researchers traveling to the site of an apiary and manually recording the value displayed on the mechanical balance [9]. Recordings were often required on an hourly basis, which was cumbersome and required additional attention. Recent development in technology has significantly reduced the need for site visits.

2.1 PROBLEM STATEMENT

After a comprehensive survey of available weighing devices to measurement of bees population, it has been found that less focus is made on calculating accurate weight, If bees population is very less (<100gm) then there must be very less gap between accurate weight of bees population and measuring weight.

There is also very less focus on cloud service if we are going to calculate the weight of bees population in time to time.

CHAPTER -3

PROPOSED WORK

The main constraint in achieving higher accuracy and increasing throughput rate of the passing products is a superimposed noise on a useful signal from the weighing system. The common sources are: (1) constant values (weight could not vary for same input) (2) the noises: electromagnetic pick-up, power harmonic, thermally unstable circuits and the gain programmable by software (through PWM's output of the microcontroller). The common sources are amplified before transferred into the A/D converter, and therefore the noise is amplified and an error is introduced in the system. (3) strain-gage based sensor (i.e., load cell) particularly sensitive to vibration .(4) cloud service: we get all update in time to time, not go with personally to weighing device specially if measurement is going for bee population .

3.1 PROPOSED SOLUTION

In this work i use strain-gage based sensor (load cell) so i can able to reach with my accurate weight. It means there is very less gap between measuring weight and accurate weight of bees population if weight <100gm.

In this proposed work i use cloud service. Automatic weighing system produces a lot of data that must be recorded and processed, which is why the system, you use to collect is just as important as the scale you choose.

For example: Packaging and shipping companies must weigh containers to ensure that shipments and the associated costs are accurate to control costs and maintain consistency and customer satisfaction.

But in this work i use the cloud service for automatic weight recording of bees population in real time in every 15 seconds so that's why system produces a lot of data that must be recorded.

IOT Weighing Scale

Delete Logs

ID	Time	Weight(gms)
45	13:43:11 09/05/2018	0
46	13:43:13 09/05/2018	3
47	13:43:25 09/05/2018	1
48	13:43:40 09/05/2018	0
49	13:43:55 09/05/2018	1
50	13:44:11 09/05/2018	0
51	13:44:26 09/05/2018	0
52	13:44:41 09/05/2018	241
53	13:44:56 09/05/2018	2
54	13:45:11 09/05/2018	3
55	13:45:27 09/05/2018	0
56	13:45:41 09/05/2018	2
57	13:45:56 09/05/2018	1
58	13:46:12 09/05/2018	2

Figure 1: recorded database server

CHAPTER -4

PROJECT IMPACT /SOCIAL BENEFITS- EXPECTED OUTCOME

A microcontroller is a type of microprocessor furnished in a single integrated circuit and needing minimum support chips. Its principle nature is maximum efficiency at minimum cost. A microcontroller is an intelligent core for a specialized dedicated system. It is a complete computer system with input-output lines, timers, Read-Only Memory (ROM), Read-Write Memory (RAM) and some peripherals such as counters and timers, analog to digital converters, comparators and serial ports [15]. The PIC microcontroller is mostly build on the Harvard architecture and has characteristics which include RISC processor design, single word instructions, machine and data memory configuration and characteristic instruction formats.

The Reduced Instruction Set Computer (RISC) has 35 instructions and each instruction performs more elementary operations. Consequences of this are a smaller silicon area, faster execution and reduced program size with fewer accesses to main memory. In the Harvard architecture the data and instructions use different path and storage areas. This type of machine can read and write instructions to and from memory at the same time and results in a faster machine [16]. Since the device has separate buses for instructions and data it is possible for instructions to be sized differently than data items. Being able to vary the number of bits in each instruction op-code makes possible the optimization of program memory and the use of single-word instructions that can be fetched in one bus cycle [17]. The PIC architecture has a two-stage instruction pipeline, since the fetch and current instruction and execution of the previous one can overlap in time, one complete instruction is fetched and executed at every machine cycle (pipelining).

The PIC clocking system is designed so that an instruction is fetched, decoded, and executed every four clock cycles where the clock is either internal or external. In this manner a PIC equipped with a 4MHz oscillator clock beats at a rate of 0.25 micro seconds. Since each instruction executes at every four clock cycles, each instruction takes 1 microsecond when the pre-scalar is not used.

In the PIC16F690 microchip controller there 18 general purpose I/O pins are available where one pin is only input that is the master clear pin, 3 ports which include port A which is 6 pins wide, port B which is 4 pins wide and port C which is 8 pins wide making the device have 18 pins and 2 pins for powering the device [18]. This microcontroller has three types of memory- ROM, RAM and EEPROM.ROM memory is used to permanently save program being executed that is why it is often referred to as program memory. Since ROM is made of FLASH technology, its contents can be changed by providing special programming voltage [19]. Similar to program memory, the contents of EEPROM are permanently saved even after the power going off. The third memory is the RAM which consists of two parts: the general purpose registers (GPR) and the special function registers (SFR).

Peripheral devices in the pic16f690 microchip controller include the timer which is used as timers or counters. The asynchronous receiver transmitter contains all the clock generators, shift registers and data buffers necessary to perform an input or output serial data transfer independent of device program execution [20].

Accuracy Measurements: Farmer can be assured of getting absolutely accurate measurements of the load due to the precise technology [21].

Cost Efficient: The traditional way of weighing the product involved a number of people loading and unloading, while others read the weight and some would tally the total. This means you had to pay every single person involved in the process, which increased the costs. On the other hand, you just need one to two people to weigh using on-board scales, thus helping you save money [22].

Saves Time: When you have on-site weighing scales, you don’t have to repeatedly weigh the load. You can just weigh it once, either at the warehouse or the farm from where it is collected. This helps you save time by eliminating the need to constantly load and unload the produce to weigh at the site and then again at the collection point [23].

Flexible Solutions: On-site scales are quite versatile, which can be used on different surfaces and a variety of vehicles of different sizes [24].

Data Management: Easily collect and manage the explosion of data from sensors, cloud services , connected equipment and existing systems [25]. Enable farmers to easily visualize data and take action on insights and recommendations.

CHAPTER -5

SYSTEM ARCHITECTURE AND DESCRIPTION

The design of the microcontroller weigh scale was based on the PIC16F690 as the heart of the system. The design was divided into two main parts; hardware and software design.

5.1 SOFTWARE DESIGN

Software design was divided into the following divisions:

a) Arduino program.
b) Microcontroller program.
c) ADC converter program.
d) Bluetooth program.
e) Server program.

This proposed system consists of Weight sensor (load cell), arduino UNO, LCD display, microcontroller, ADC converter, Bluetooth module. The arduino needs the power supply of 5v. The main platform we are using to build the project is arduino which provides us the flexibility to write code effectively in convenient way and also it will provides us features like inexpensive, cross platform, open source and extensible software, easy for beginners. The arduino is a microcontroller based on UNO.

The initialization involved pin configuration. The ADC program and arduino UNO programs can also write in assembly code language.

Server program has divided into some parts like create database, indexing, delete record, update record. Here delete program is used to delete any raw in database.

Below is the flow chat developed to guide in the development of the needed program.

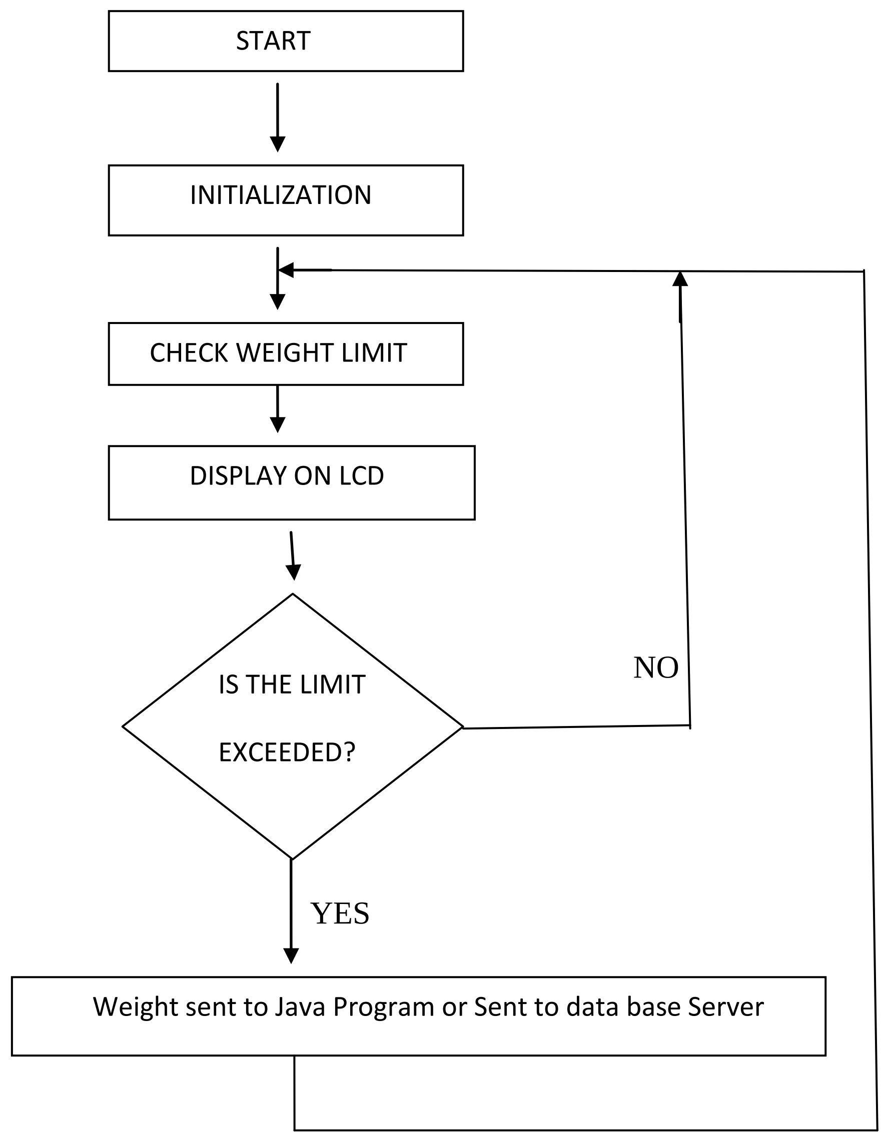

Figure 2: flow chart for the program

Weight machine can able to start when machine is in on mode, if no input value is there then it shows some offset value so we should try more and more to overcome this offset value so we can say it is accurate weighting machine.

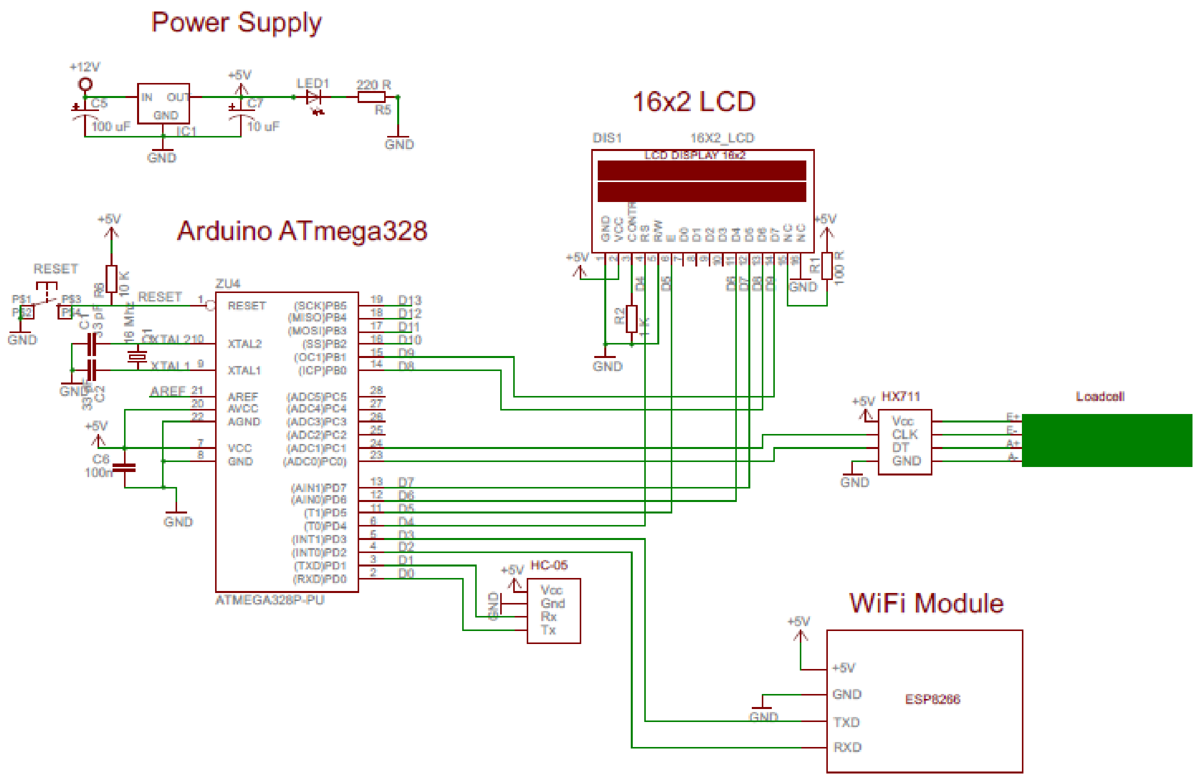

Figure 3: Circuit Diagram for the program.

5.2 HARDWARE DESIGN

LCD to PIC Interface; LCD requires 8 pins for the 8 bit mode of sending data for 8-bit configuration mode of the LCD. PORTC of PIC16F690 is 8 bit wide and was found convenient for the data lines of the LCD. That is, RC0-7 PIC pins where connected to D0-7 data lines of LCD as shown on figure 3. Enable pin (E) of the LCD was connected to RA1 (pin 18) of the microcontroller. Register select (RS) pin of the LCD was connected to RB5 (pin 12) of microcontroller. Since the objective was to write data to the LCD read/write line was grounded; because at no point is data being read from the LCD. This helped free up an extra pin. Enable pin (E) of the LCD was connected to RA1 (pin 18) of the microcontroller. Register select (RS) pin of the LCD was connected to RB5 (pin 12) of microcontroller

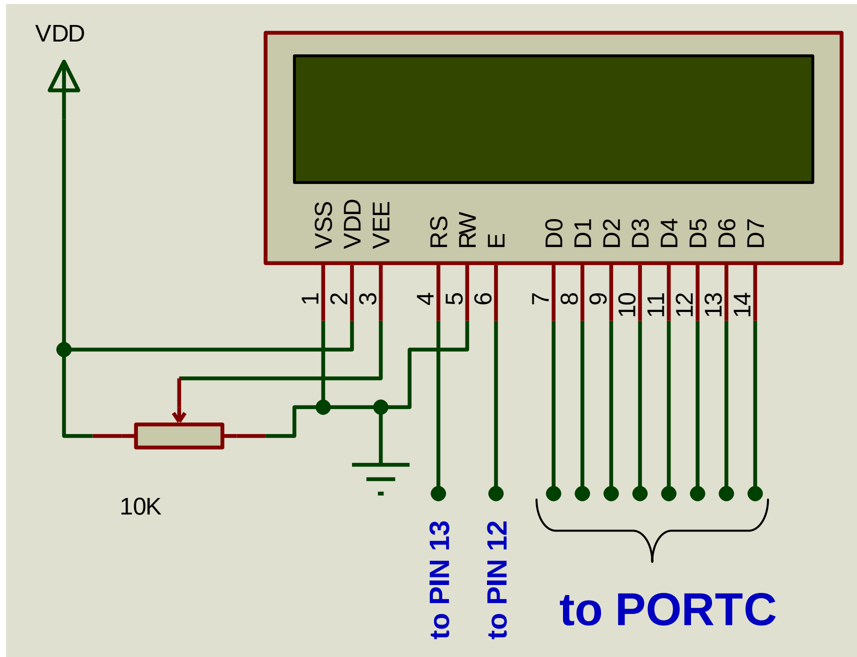

Figure 4: LCD to PIC interface

The 10KΩ resistor connected between V_{SS} and V_{DD} to ground is used to vary the contrast of the LCD brightness. An optimal value is achieved depending on one wish. It varies the voltage from V_{DD} value ground. To achieve a visual sign when operating in the safe limits (0-100gms) a green LED was used. This was connected between VDD and RA5 (pin 3) through a limiting resistor of 330 ohms whose justification is given in equation below.

$$R = \frac{(5.3 - 1.8)V}{25\ mA} = 140\ \Omega$$

To make sure the value of current sourced is far below the maximum allowed current (25 mA) a 330 ohms resistor was chosen.

The System architecture is divided into two parts.

Analog part: This includes the sensor and the amplifier sections.

Digital part: This consists of the Micro-controller along with network connectivity.

This proposed system consists of weight sensor (load cell), arduino UNO, bluetooth module, wi-fi module, LCD display. The arduino needs the power supply of 5v. The main platform we are using to build the project is arduino which provides us the flexibility to write code effectively in convenient way and also it will provides us features like inexpensive, cross platform, open source and extensible software, easy for beginners. The arduino is a microcontroller based on UNO.

Goods on crate are measured on crate itself. If weight is above its tolerate capacity of maximum capacity of crate, it will send the value to java program itself and also send to database server, to indicate crate is about to full. So that responsible person will be attentive. The weight of goods will be stored on cloud.

5.3 LCD DISPLAY

LCD stands for Liquid crystal display. They have become very common with industry by clearly replacing the use of cathode ray tubes (CRT).CRT consumes more power than LCD and also bigger and heavier.

5.4 LOAD CELL

As per dictionary, a load cell described as weight measurement device necessary for electronic scale that displays weight in digits. However, load cell is not restricted to weight measurement in electronic scale. Load cell is passive transducer or sensor which converts applied force into electrical signals. They are also referred to as Load Transducers‖.

Load cells use different operating principles:

- Load cells based on fluid pressure
- Load cells based on elasticity
- Load cell based on magnetostriction or piezoelectric effect

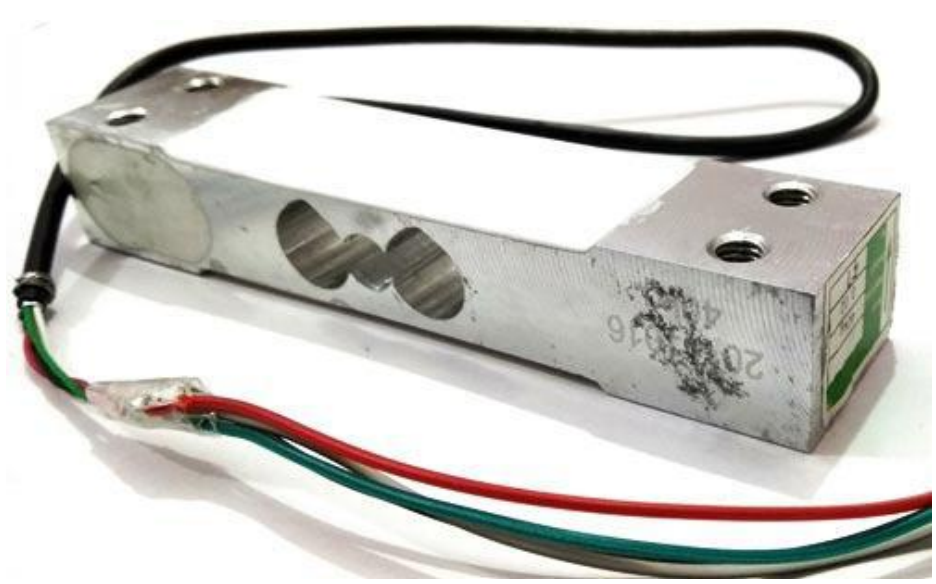

Figure 5: Load cell

Now the electrical signals generated by Load cell is in few millivolts, so they need to be further amplify by some amplifier and hence HX711 Weighing Sensor comes into picture. HX711 Weighing Sensor Module has HX711 chip, which is a 24 high precision A/D converter (Analog to digital converter).HX711 has two analog input channels and we can get gain up to128 by programming these channels. So HX711 module amplifies the low electric output of Load cells and then this amplified & digitally converted signal is fed into the arduino to derive the weight.

A load cell is described as a “weight measurement device necessary for electronic scales that display weights in digits”. However, load cell is not restricted to weight measurement in electronic scales.

Load cell is a passive transducer or sensor which converts applied force into electrical signals. They are also referred to as “Load transducers”.

The load cell based on strain gauges. Hence, the term ‘load cell’ means ‘strain gauge-based load cells’.

The sensing or spring element is the main structural component of the load cell. The element is designed in such a way that it develops a strain, directly proportional to the load applied. Sensing elements are normally made of high strength alloy steels (nickel plated for environmental protection), precipitation-hardened stainless steels, heat-treated aluminum alloys, or beryllium copper alloys.

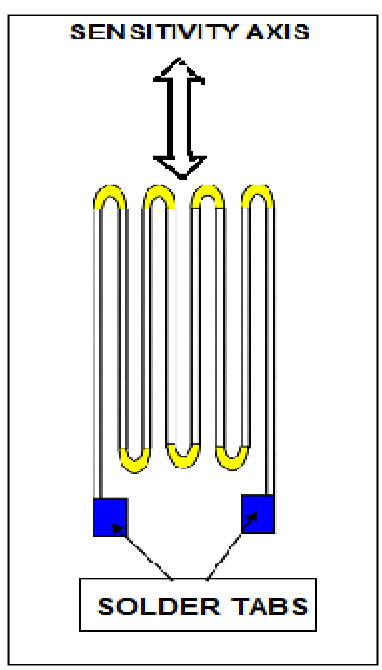

Figure 6: Shows a foil-type strain gauge

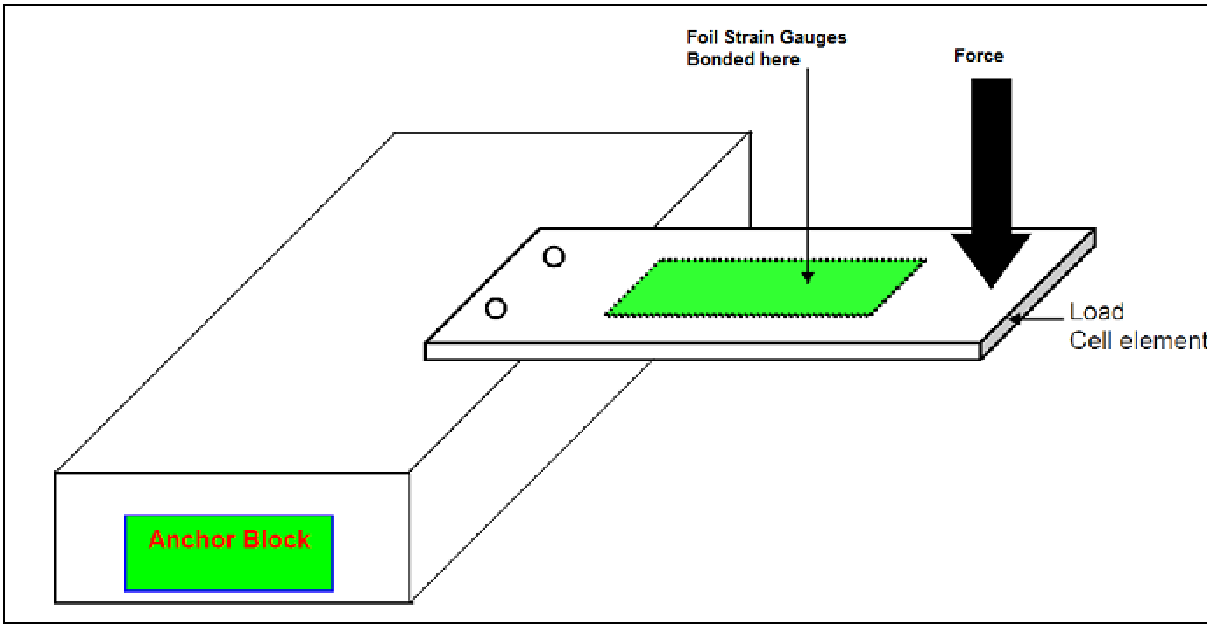

Figure 7: Principle of mounting load cell

5.5 CHARACTERISTICS OF LOAD CELL

1. Highly precise and linear measurements

2. Little influence due to temperature changes.

3. Long operating life due to lack of moving parts or any parts that generate friction. 4. Ease in production due to small number of components.

5. Excellent fatigue characteristics

The Load cell which we are using is for Kitchen scale i.e we can measure weight of vegetables, fruits etc.

Available load cells are 5 Kg., 10 Kg., 20 Kg., 40 Kg., 80 Kg and others. Out of these available load cells we will choose 6 Kg. load cell as it satisfies all the requirements.

We have used the load cell which is available in local market.

Load cell selected – CZL601-6Kg.

5.6 ARDUINO UNO

Arduino Uno is a microcontroller board based on the ATmega328P. (It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.. You can tinker with your UNO without worrying too much about doing something wrong, worst case scenario you can replace the chip for a few dollars and start over again.

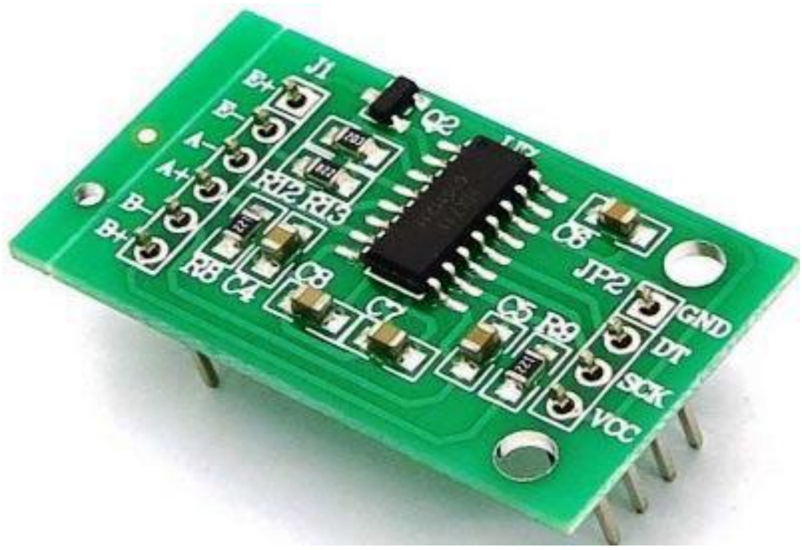

Figure 8: Arduino UNO

5.7 CLOUD SERVICE

Automatic weighing system produces a lot of data that must be recorded and processed, which is why the system, you use to collect is just as important as the scale you choose.

For example: Packaging and shipping companies must weigh containers to ensure that shipments and the associated costs are accurate to control costs and maintain consistency and customer satisfaction [26]. The ideal connection would allow the operator to print complex labels quickly by enabling communication directly between the scale and printer.

One of the latest advancements in data technology, cloud computing allows you to process massive amounts of data through a secure connection that requires no hard drive or dedicated server. Data is simply stored in the cloud and can be accessed from any authorized location throughout the world. This option is ideal for operations that need to compile data from a variety of remote locations.

•	Load cell: Load cell is used as force measuring component. We collected a load cell of 5KG. As a result in our project we only could measure till 5 KG. There are four or two strain gauges in load cell depending on the capacity of load cell. It works like a Wheatstone bridge.

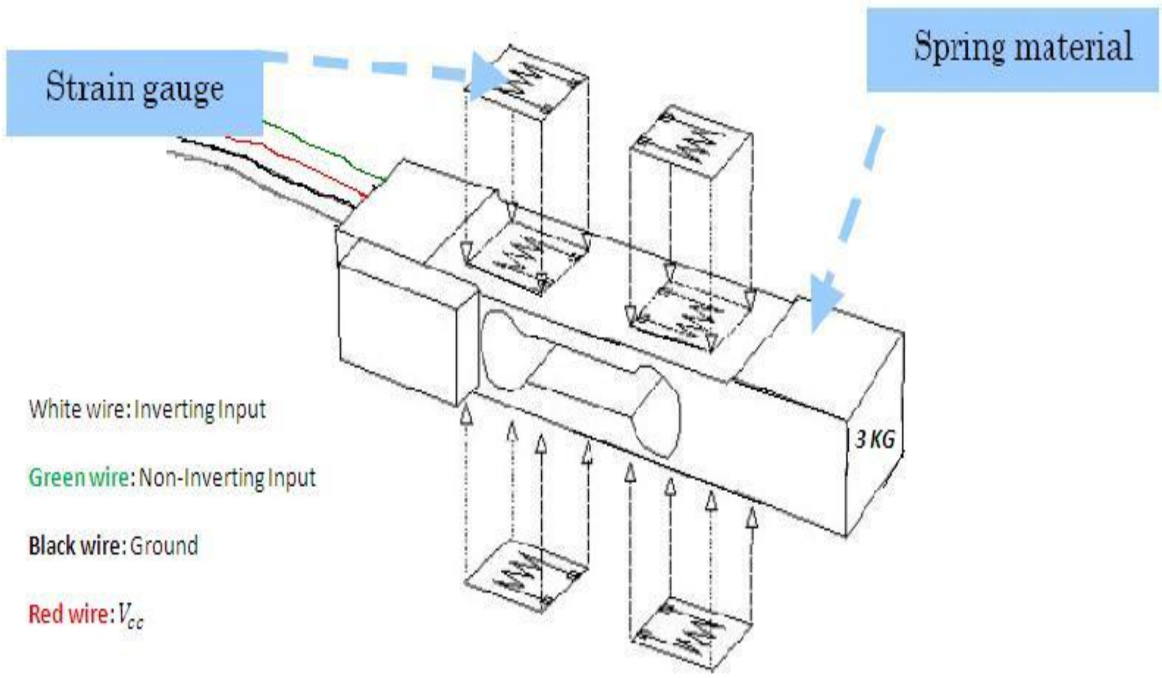

Figure 9.a: Block diagram of load cell

When we don’t give any load to it, it remains balanced; there is no flowing of current as well as corresponding voltage.

Figure 9.b: Load cell

Whenever i hang on a load on to it, imbalance situation is occurred. Stress is caused by external force. Strain gauge converts the deformation to electrical signals. The electrical signal output is typically in the order of a few mill volts.

5.8 OP-AMP AD620 & LM358

It is an instrumentation amplifier which is a type of differential amplifier. They are mainly used to amplify very small differential signals from strain gauges. It has got very low DC offset, low drift, low noise, very high gain. I made our gain 100 using AD620. I also used op-amp LM358 to amplify our AD620 output further more.

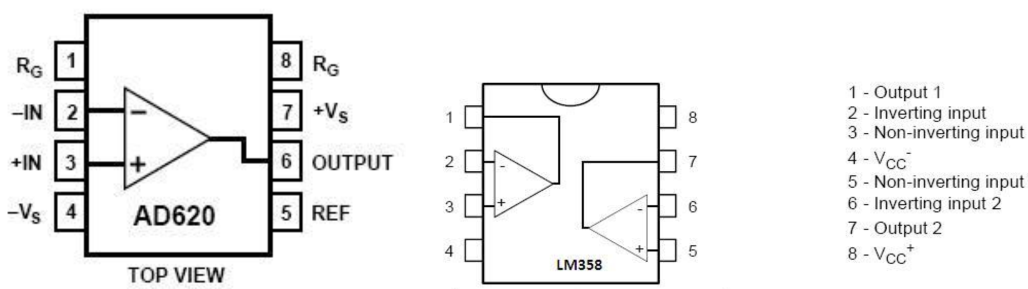

Figure 10: Op-amp AD620

5.9 MICROCONTROLLER (PIC16F877A)

PIC16F877A is a small piece of semiconductor integrated circuits. The package type of these integrated circuits is DIP package. I could use another microcontroller. But PIC16F877A has some advantages. . This package is very easy to be soldered onto the strip board. PIC16F877A is very cheap. Moreover, it is also very easy to be assembled. So i use this chip. This IC can be reprogrammed and erased up to 10,000 times .Therefore it is very good for new product development phase.

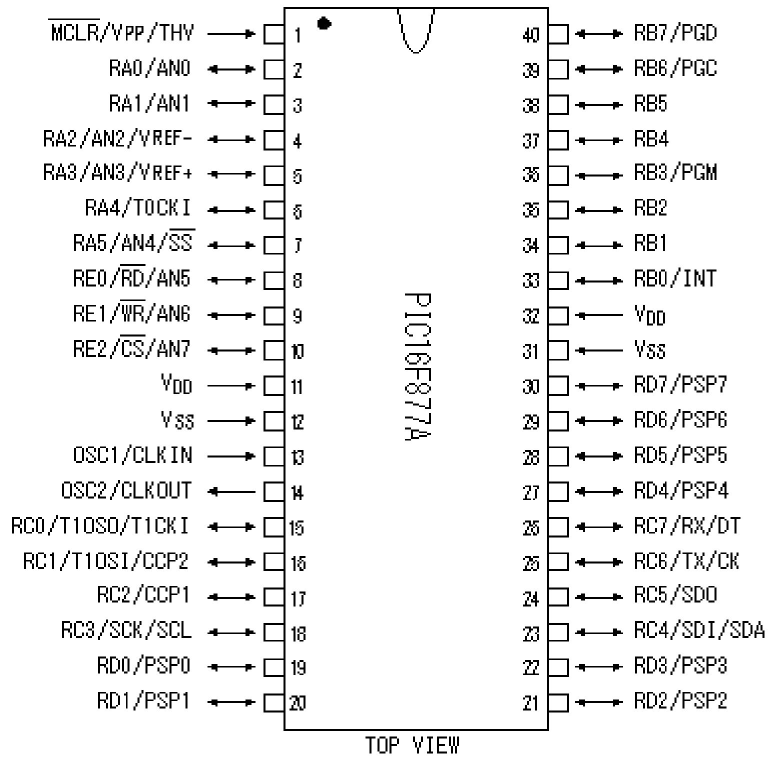

Figure 11: Microcontroller (PIC16F877A)

5.10 PRINTED CIRCUIT BOARD

A printed circuit board or PCB is used to mechanically support and electrically connect electronic components using conductive pathways, tracks or signal traces etched from copper sheets laminated onto a non-conductive substrate. I set up microcontroller in the PCB.

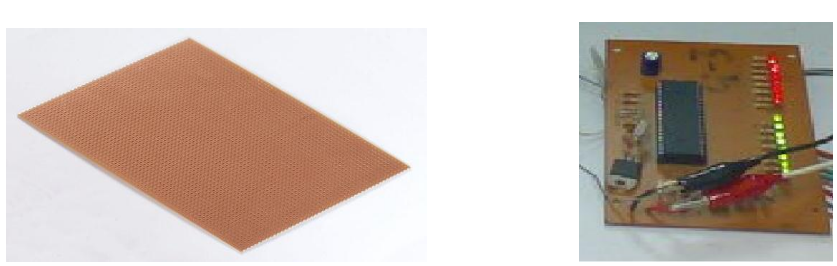

Figure 12: Printed circuit board

Components that i need to make our IC (PIC16F877A) work are just a 5V power supply adapter, a 20MHz crystal oscillator and 2 units of 22pf capacitors. So i set up the components in PCB board. I also set up some LED's for the 7 segment in the PCB.

5.11 HARDWARE DESCRIPTION

Hardware part is done in a breadboard. I constructed the circuit diagram on bread board. Isolate four wires are coming out of the load cell. Two wires are used as input in the amplifier and other two are used as V_{CC} and Ground. I connected it with the AD620; supplied (+ve) 5 volt and (–ve) 5 volt separately using power supply in the AD620 op amp. I measured the change in voltage as the weight flexes via the green and white wires. I measured V_1 across a resistor of 1 k.ohm and connected a capacitor parallel to resistor, so that i can avoid the noise. For one KG it was giving us only 1.8 volts whereas i supposed to get 5 volt for one KG. So to amplify the voltage i passed the output voltage (V_1) to op-amp LM358. Using a variable resistor of 17.2k.ohm i amplify the previous voltage 18 times using following formula:

$$\boldsymbol{V_2 = V_1(1+\frac{R_F}{R_2})}$$

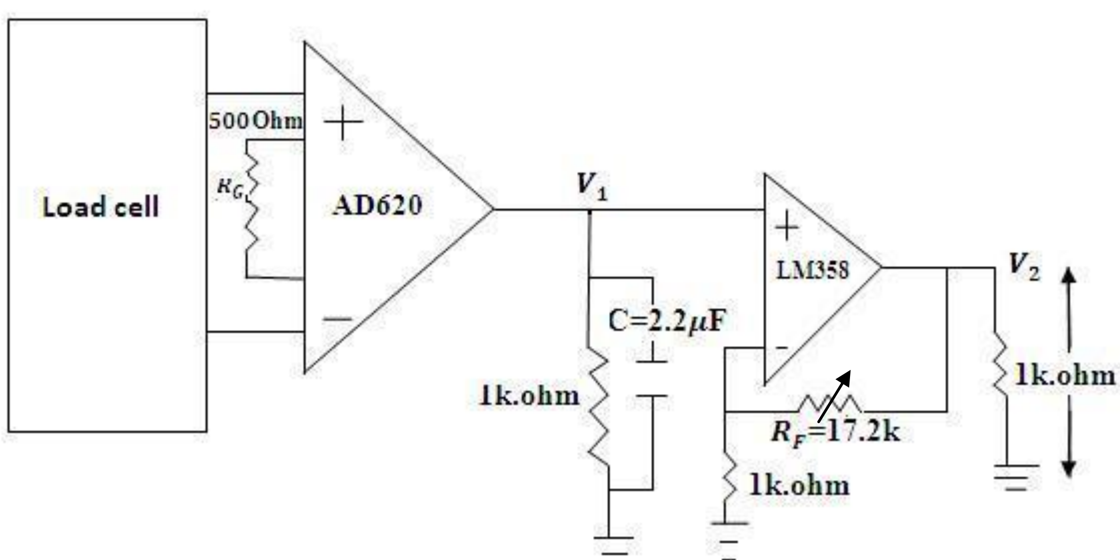

Figure 13: Circuit diagram for hardware part

5.12 SOFTWARE DESCRIPTION

Whenever i got the analog input from the hardware part, i need to pass it through the microcontroller to the 7-segment display.

Software part is mainly the coding or programming. As that input was analog and i wanted to show it on a digital form. The analog to digital conversion was completely done in code. This part was little bit difficult to as i had to check the code many times. Port A is used as out input of the analog input. I used port B of microcontroller for the 7-segment display and used port D for the data selector. I converted the input from binary to 10-bit BCD which displayed in 7-segment in a corresponding decimal value. As i use 10 bit, i could show from 0-1023.

I burn the code into microcontroller using a burner and then set up the chip in a Printed circuit board.

Figure 14: Digital weighing machine

After that i connected the PCB board with the 7-segment display. When i didn't give any load, the display was showing an offset value. For 1/2 kg it was showing us decimal 588 and for one kg it was 1023.

5.13 HC-05 BLUETOOTH MODULE

Make a Bluetooth communication between two separate arduino Boards as master and slave devices.

Figure 15: HC-05 bluetooth module

Before i start with the first example, controlling an arduino using a smartphone, let's take a closer look at the HC-05 Bluetooth module. Comparing it to the HC-06 module, which can only be set as a Slave, the HC-05 can be set as Master as well which enables making a communication between two separate arduino Boards. There are several different versions of this this module but I recommend the one that comes on a breakout board because in that way it's much easier to be connected. The HC-05 module is a Bluetooth SPP (Serial Port Protocol) module, which means it communicates with the arduino via the Serial Communication.

CHAPTER -6

RESULT

In arduino and bluetooth based bees population weighing machine has limited capacity with 5kg load cell and if input cross the tolerate weight then value go to software part and also send the value to database server. LCD display, bluetooth module and wi-fi module used as output devices.

The work proposes a cost effective and user friendly Smart Crate weighing system using bluetooth Module and wi-fi module. The significant advantage of the method is that the weighing system in agriculture is centralized and monitored using the cloud service. It saves the time to measure honey production. This concept avoids delay in analysis of honey production and to make decision about price. The proposed system is robust, reliable and requires less maintenance. The idea proposed in this article can be used by farmers from remote as well as urban area.

Figure 16: Digital display

6.1 EXPERIMENTAL RESULTS AND ANALYSIS

Table 1 below shows simulated results when the design was run on protean. It shows the variation of the displayed value of the weight as the voltage is varied from 0 to 5 volts as well as the resistance drops consequently. The table also shows the status of the LCD as the weight is varied.

Table 1 of simulated results.

VOLTAGE %	RESISTANCE KΩ	VOLTAGE V	DISPLAYED VALUE(grams)	LCD STATUS
0	**10**	**0**	**0.0**	**ON**
1	**9.9**	**0.05**	**9.775**	**ON**
2	**9.8**	**0.1**	**19.55**	**ON**
3	**9.7**	**0.15**	**28.348**	**ON**
4	**9.6**	**0.2**	**37.145**	**ON**
5	**9.5**	**0.25**	**45.943**	**ON**
6	**9.4**	**0.3**	**53.763**	**OFF**
7	**9.3**	**0.35**	**61.583**	**OFF**
8	**9.2**	**0.4**	**69.404**	**OFF**
9	**9.1**	**0.45**	**77.224**	**OFF**
10	**9.0**	**0.5**	**85.044**	**OFF**
11	**8.9**	**0.55**	**91.886**	**OFF**
12	**8.8**	**0.6**	**98.729**	**OFF**
13	**8.7**	**0.65**	**105.571**	**OFF**
14	**8.6**	**0.7**	**112.414**	**OFF**
15	**8.5**	**0.75**	**119.256**	**OFF**

When the voltage to the microcontroller input was at zero, the displayed value was also at zero while the visual alarm was ON to show that the weight was within the specified limit. Behold 100 gm the visual alarm goes off indicating that the weight level is behold the specified minimum limit.

6.2 SYSTEM EVALUATION

- **Number of main ICs.** In terms of the number of components, the developed PIC16F690 Microcontroller Based Electronic Weighing Balance (MCBEWB) uses the least. The microcontroller chip has adequate in-built Flash Memory, EEPROM, RAM and I/O ports. Any microprocessor based electronic weighing balance would require an external RAM, an external EPROM and an I/O device among other support chips. This renders the latter extremely expensive.
- **Portability.** From the portability perspective, the developed system has unlimited range as all that is needed is a single external power supply. It has been designed to make it as compact as possible. The Butchery Electronic Weighing Balance (BEWB) is bulky and therefore has limited portability.
- **Space.** In terms of space, the developed system required least space for both the digital circuitry and the analog circuitry. This is attributed to the fact that minimal number of chips was used and to crown it all, the system was implemented on PCB boards. Owing to the many external chips used, in BEWBs and SRLEWBs, a lot of space is required for circuit implementation.
- **Flexibility.** The developed system is considered the most flexible, as in-system reprogramming is possible. This is courtesy of the built-in Serial Programmable Interface (SPI) of the PIC16F690 microcontroller. Program can be written or erased up to 1000 times. Some BEWBs and SRLEWBs use dedicated chips like those in DMMs. They only perform specific tasks and cannot be changed or modified to do otherwise. For those that are microprocessor based, in-system reprogramming is not possible. Microprocessor based related Control/Data Acquisition Applications use external EPROM for program storage. To change the program, it is erased by exposing it to UV, a tedious process indeed.
- **Range and Resolution.** The developed electronic weighing balance has tried to address both resolution and range. It is the best in as far as striking a balance between range and resolution goes. This is a modest system with a reasonable range while at the same time being sensitive enough. This is quite rare in virtually all-electronic weighing systems.

- **Display Capability.** In as far as display capability is concerned; the developed system enjoys the widest capability. Using the ASCII code, it can display all alphanumeric data/information. It can display special characters. BEWBs and SRLEWBs can only display numeric data/information. Information in letters or symbols is usually engraved alongside the LCD display.

CHAPTER -7
CONCLUSION

7.1 CONCLUSION

The project design objectives of the weight monitoring system were successfully achieved. The system displaying unit LCD was able to monitor the variations in weight. This weigh scale is of great importance for the industries involved in weight taking and also in the retail market.

The weight of a beehive is one of many metrics that give a clear indication to the status of the colony and the health of the honey bees. In recent years, protecting honey bee populations has become an important topic due to increases in colony losses, and there has been an increased interest in developing monitoring systems for apiaries. Precision Apiary projects have become a key area of research in the field of biology and environmental science, but also engineering, with the development of more sophisticated sensors and data collectors. This project outlines the design and development of a platform weighing scale, for deployment as part of a smart beehive. The scale was validated with an ADC evaluation board and PC software initially, providing the characteristics of the platform scale. The system was then separated from the PC and used in a standalone mode. The results of the tests led to the conclusion that this design of platform scale, along with the infrastructure of data collection and distribution was a viable solution to accurately measure weight levels of the beehive. A significant aspect of the experiment was the ability to characterise and read small weight change, in the order of 5 g. The system was integrated into to a WSN node to provide a flow of information between the smart weighing scale and the base station. To provide an environmentally neutral power supply solution, a solar cell, along with a lithium ion polymer battery was used to deliver power to the system. The solution developed mirrored similar projects in the emerging field of Precision Agriculture, i.e. vineyard monitoring, cheese production and other remote processes that may not be feasible to constantly attend. Before the system can be integrated into a busy hive, the scale must be completely weather protected. Future work will include stabilization of the platform against wind, and improvement of the noise performance of the system.

REFERENCES

[1] Saji, Ruhin Mary, Drishya Gopakumar, and Harish Kumar. "A Survey on Smart Garbage Management in Cities using IoT." International Journal Of Engineering And Computer Science5.11 (2016), pp. 1-3.

[2] Mahajan, Kanchan, and J. S. Chitode. "Waste bin monitoring system using integrated technologies." International Journal of Innovative Research in Science, Engineering and Technology3.7 (2014), pp. 2-12.

[3] Parkash, Prabu V. "IOT based Waste management for smart city." International Journal of Innovative Research in Computer and Communication Engineering 4.2 (2016).

[4] Chowdhury, Belal, and Morshed U. Chowdhury. "RFID-based real-time smart waste management system." Telecommunication Networks and Applications Conference, 2007. ATNAC 2007. Australasian. IEEE, 2007, pp. 175-180.

[5] Zygiaris, Sotiris. "Smart city reference model: Assisting planners to conceptualize the building of smart city innovation ecosystems." Journal of the Knowledge Economy 4.2 (2013), pp. 217-231.

[6] Arasteh, H., et al. "Iot-based smart cities: a survey." Environment and Electrical Engineering *(EEEIC), 2016 IEEE 16th International Conference on*. IEEE, 2016, pp. 1-6.

[7] Ciudin, Rodica, et al. "Vacuum waste collection system for an historical city centre." *UPB Scientific Bulletin, Series D: Mechanical Engineering* 76.3 (2014), pp. 215-222.

[8] Rada, E. C., et al. "Web oriented technologies and equipments for MSW collection." Proceedings of the International Conference on Risk Management, Assessment and Mitigation-RIMA. Vol. 10. No. 201. 2010, pp. 150-153.

[9] Mezey, Armand G. "Waste collection system for segregating solid waste into preselected component materials." U.S. Patent No. 5,035,563. 30 Jul. 1991.

[10] Nakou, D., A. Benardos, and D. Kaliampakos. "Assessing the financial and environmental performance of underground automated vacuum waste collection systems." Tunnelling and Underground Space Technology 41 (2014), pp. 263-271.

[11] Popa, Cicerone Laurentiu, and Costel Emil Cotet. "Virtual Model in Monitoring and Optimization of a Selective Waste Collection Integrated System." International Conference

on Advanced Manufacturing Engineering and Technologies . Springer, Cham, 2017, pp. 41-55.

[12] Jalali, Roozbeh, Khalil El-Khatib, and Carolyn McGregor. "Smart city architecture for community level services through the internet of things." Intelligence in Next Generation Networks (ICIN), 2015 18th International Conference on. IEEE, 2015, pp. 108-113.

[13] Cramp, David. "A practical manual of beekeeping." Spring Hill (2008), pp. 7-13.

[14] Bradbear, Nicola. "Bees and their role in forest livelihoods: a guide to the services provided by bees and the sustainable harvesting, processing and marketing of their products." Non-wood Forest Products (2009), pp. 19-25.

[15] Szabo, Tibor I. "Effect of weather factors on honeybee flight activity and colony weight gain." *Journal of Apicultural Research* 19, no. 3 (1980), pp. 164-171.

[16] Kulkarni, Prajakta, and Yusuf Ozturk. "mPHASiS: Mobile patient healthcare and sensor information system." Journal of Network and Computer Applications 34.1 (2011), pp. 402-417.

[17] F. Edwards Murphy, M. Magno, A. Frost, A. Long, N. Corbett, and E. Popovici. "Demo abstract: smartSync; when toys meet wireless sensor networks." Real-World Wireless Sensor Networks. Springer International Publishing, 2014, pp. 91-96,.

[18] Allen, Paul G. "Webcam-based interface for initiating two-way video communication." U.S. Patent No. 7,003,795. 21 Feb. 2006.

[19] Orr, Robert J., and Gregory D. Abowd. "The smart floor: A mechanism for natural user identification and tracking." *C*HI'00 extended abstracts on Human factors in computing systems. ACM, 2000, pp. 275-276.

[20] Murphy, Fiona Edwards, et al. "b+ WSN: Smart beehive for agriculture, environmental, and honey bee health monitoring—Preliminary results and analysis." Sensors Applications Symposium (SAS), 2015 IEEE. IEEE, 2015, pp. 1-6.

[21] McLellan, A. R. "Honeybee colony weight as an index of honey production and nectar flow: a critical evaluation." Journal of Applied Ecology (1977), pp. 401-408.

[22] Fitzgerald, Darren W., et al. "Design and development of a smart weighing scale for beehive monitoring." Signals and Systems Conference (ISSC), 2015 26th Irish. IEEE, 2015, pp.1-6.

[23] Seeley, Thomas D. The wisdom of the hive: the social physiology of honey bee colonies. Harvard University Press, 2009.

[24] Bromenshenk, Jerry J., et al. "Bees as biosensors: chemosensory ability, honey bee monitoring systems, and emergent sensor technologies derived from the pollinator syndrome." Biosensors 5.4 2015, pp. 678-711.

[25] Platt, D., et al. "Novel force transducer for the measurement of tendon force in vivo." Journal of biomechanics 27.12 1994, pp. 1489-1493.

[26] Ancillotti, Emilio, Raffaele Bruno, and Marco Conti. "The role of communication systems in smart grids: Architectures, technical solutions and research challenges." Computer Communications 36.17-18 2013, pp. 1665-1697.

APPENDIX

Arduino codes

```
#include <LiquidCrystal.h>
#include <SoftwareSerial.h>

SoftwareSerial WSerial(2,3);
#include "HX711.h"
LiquidCrystal lcd(4, 5, 6, 7, 8, 9);
int irPin = 11;
// HX711.DOUT  - pin #A0 // HX711.PD_SCK - pin #A1
HX711 scale(A0,A1);    // parameter "gain" is ommited; the default value 128 is used by the
library
long zero_val = 0;
long new_val=0;
long dif_val=0;
float weight;
int wgram;
unsigned long previousMillis = 0;
unsigned long previousMillis2 = 0;
long interval = 250;           // interval at which to blink (milliseconds)

//************************************************

void adjust_loadcell()
{
  delay(1000);
  zero_val = scale.read_average(100);
}
//************************************************
void find_weight()
```

```
{
 new_val = scale.read_average(2);
 //Serial.print("read average: ");
 //Serial.println(new_val);    // print the average of 20 readings from the ADC
 if(new_val> zero_val)
 {
   dif_val = new_val - zero_val;
   //weight = dif_val*0.0002328;
   weight = dif_val * 0.000298;    //adc val to mV
   weight = weight * 1 / 128;  //mV to Kgrams
   //weight = weight * 4;
   //weight = weight + calib_weight;
    wgram = weight*1000;

   /*
   if(weight > (prev_weight+0.050) || weight < (prev_weight-0.050))
   {
   weight = weight + calib_weight;
   }
   else
   {
     weight = prev_weight;
   }
   prev_weight=weight;
   */

 }
 else
 {
   weight = 0;
 }
```

```
}
//*************************************************
void setup()
{
  Serial.begin(9600);
  WSerial.begin(9600);
  Serial.println("Program Started");
  //pinMode(irPin, INPUT);digitalWrite(irPin, HIGH);
  //pinMode(swPin, INPUT);digitalWrite(swPin, HIGH);
  //init_servo();
  lcd.begin(16, 2);
  lcd.setCursor(0, 0);
  lcd.print("Wireless Weigh. ");//Wireless Weighing Scale with Java Application
  lcd.setCursor(0, 1);
  lcd.print("     Scale      ");
  delay(5000);
  lcd.clear();lcd.print("Init Loadcell...");
  adjust_loadcell();
  delay(5000);
  lcd.clear();
}
//*************************************************
void loop()
{
  find_weight();
  unsigned long currentMillis = millis();
  if(currentMillis - previousMillis > interval)
  {
    lcd.clear();
    lcd.print("W(G) : ");lcd.print(wgram);
    WSerial.print("<");WSerial.print(wgram);WSerial.print(">");WSerial.println("");
```

```
Serial.print("XYZ");Serial.print("<");Serial.print(wgram);Serial.print(">");Serial.println("")
;
    previousMillis = currentMillis;
  }
  if(millis() - previousMillis2 > 15000)
  {
    WSerial.println("<U>");
    previousMillis2 = currentMillis;
  }
}
//***********************************************
```

ESP8266 codes

```
#include <ESP8266WiFi.h>
#include <Wire.h>

#define wifi_ssid "AndroidAP"
#define wifi_password "87654321"

WiFiClient espClient;

int weight;
char cmd_arr1[100];
int cmd_count1;

const char* host = "iotweighingscale.xyz";

//***************************************************************
void serial_get_command()
```

```
{
 char inchar=0;
 int temp;
 if(Serial.available() > 0)
 {
   inchar = Serial.read();
   if(inchar == '<')
   {
     cmd_count1=0;
     while(inchar != '>' && cmd_count1<15)
     {
       if(Serial.available() > 0)
       {
         inchar = Serial.read();
         cmd_arr1[cmd_count1++] = inchar;
         cmd_arr1[cmd_count1] = '\0';
       }
     }
     if(inchar == '>')
     {
       cmd_count1--;cmd_arr1[cmd_count1] = '\0';
       if(cmd_arr1[0]=='U')
       {
         upload_data();
       }
       else
       {
         weight = atoi(cmd_arr1);
       }
     }
   }
```

```
  }
}
//******************************************************************
void setup_wifi()
{
  pinMode(5, OUTPUT);
  digitalWrite(5, LOW);
  delay(10);
  // We start by connecting to a WiFi network
  //Serial.println();
  //Serial.print("Connecting to ");
  //Serial.println(wifi_ssid);

  WiFi.begin(wifi_ssid, wifi_password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  digitalWrite(5, HIGH);

  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}
//******************************************************************
void upload_data()
{
  WiFiClient client;
  const int httpPort = 80;
```

```
  if (!client.connect(host, httpPort)) {
    return;
  }
  Serial.println("Client Connected");
  String url = String(weight);
  //url+="\r\n";

  // Request to the server
  client.print(String("GET ") + "/update.php?weight=" + url + " HTTP/1.1\r\n" +
               "Host: " + host + "\r\n" +
               "Connection: close\r\n\r\n");

  for(int i=0;i<5;i++)
  {
    digitalWrite(5, LOW);delay(100);digitalWrite(5, HIGH);delay(100);
  }
}
//****************************************************************
void setup()
{
  Serial.begin(9600);
  //Serial.println("System started");
  setup_wifi();
}
//****************************************************************
void loop()
{
  serial_get_command();
}
//****************************************************************
```

SERVER CODE

INDEX.PHP

```
<!DOCTYPE html>
<html>

<head>
	<style>
	 h1
	 {
		text-align: center;
		color: #017572;
	 }
	 table {
		border-collapse: collapse;
		width: 50%;
		margin-left:25%;
		margin-right:25%;
	}

	th, td {
		text-align: center;
		padding: 8px;
	}

	tr:nth-child(even){background-color: #f2f2f2}

	th {
		background-color: #4CAF50;
		color: white;
	}
```

```
	</style>
	<title>Logs</title>


	<meta http-equiv="refresh" content="10" >

</head>

<body>
<center>
<h1>IOT Weighing Scale</h1><hr>

<input target="_blank" type="button" value="Delete Logs"
onclick="window.open('delete.php','_self')">


<?php

		$servername = "mysql.hostinger.in";
		$username = "u606903796_user";
		$password = "87654321";
		$db = "u606903796_db";



		// Create connection
		$conn = new mysqli($servername, $username, $password, $db);

		// Check connection
		if ($conn->connect_error)
		{
			die("Connection failed: " . $conn->connect_error);
```

```
		}
		//echo "Database - Connected successfully";

		$sql = "SET SESSION time_zone='+5:30'";
		$result = $conn->query($sql);

		echo
"<BR><BR><table><tr><th>ID</th><th>Time</th><th>Weight(gms)</th></tr>";
		$sql = "SELECT * FROM Logs";
		$result = $conn->query($sql);
		if ($result->num_rows > 0)
		{
			// output data of each row
			while($row = $result->fetch_assoc())
			{
					echo "<tr>";
					echo
"<td>".$row["id"]."</td><td>".date_format(date_create($row['time']),"H:i:s
d/m/Y")."</td><td>". $row["weight"]."</td>";
					echo "</tr>";
			}
		}
		else
		{echo "0 results";}
		echo "</table>";
		//Close Connection
		$conn->close();

	?>
</center>
<BR><BR>
```

```
</body>
</html>
```

CREATE.PHP

```
<!DOCTYPEÂ html>
<html>

<head>
	<title>Create Table</title>
</head>

<body>

	<h2><center>Create Table</center></h2>
	<br><hr><hr><br>

	<?php
		$servername = "mysql.hostinger.in";
		$username = "u684796545_user";
		$password = "waheguru";
		$db = "u684796545_db";

		// Create connection
		$conn = new mysqli($servername, $username, $password, $db);

		// Check connection
		if ($conn->connect_error)
		{
			die("Connection failed: " . $conn->connect_error);
```

```
}
echo "Connected successfully";

// sql to create table
$sql = "CREATE TABLE Logs (
id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
time TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
user VARCHAR(3),
weight VARCHAR(5),
amount VARCHAR(5),
balance VARCHAR(5)
)";

if ($conn->query($sql) === TRUE)
{echo "<BR>Table Logs created successfully";}
else
{echo "<BR>Error creating table: " . $conn->error;}

// insert values

$sql = "INSERT INTO Logs (user, weight, amount, balance) VALUES (1,
10.5, 50, 450)";
if ($conn->query($sql) === TRUE)
{echo "<BR>New record created successfully";}
else
{echo "Error: " . $sql . "<br>" . $conn->error;}

echo "<BR><BR>";
//Close Connection
$conn->close();
```

```
	?>

</body>
</html>
```

DELETE.PHP

```
<html>

<head>
	<title>Delete Table</title>
</head>

<body>

	<h2><center>Delete Table</center></h2>
	<br><hr><hr><br>
	<?php

	$servername = "mysql.hostinger.in";
	$username = "u606903796_user";
	$password = "87654321";
	$db = "u606903796_db";

	// Create connection
	$conn = new mysqli($servername, $username, $password, $db);

	// Check connection
	if ($conn->connect_error)
	{
		die("Connection failed: " . $conn->connect_error);
	}
```

```
        echo "Connected successfully";

        $sql = "DELETE FROM Logs";

        if ($conn->query($sql) === TRUE)
        {echo "<BR>Table Deleted successfully";}
        else
        {echo "<BR>Error Deleting table: " . $conn->error;}

        //Close Connection
        $conn->close();

        header('Refresh: 3; URL=index.php');
?>

</body>
</html>
```

UPDATE.PHP

```
<?php
$weight = $_GET['weight'];

$servername = "mysql.hostinger.in";
$username = "u606903796_user";
$password = "87654321";
$db = "u606903796_db";

// Create connection
$conn = new mysqli($servername, $username, $password, $db);

// Check connection
if ($conn->connect_error)
```

```
{
		die("Connection failed: " . $conn->connect_error);
}
echo "Connected successfully";

// insert values
$sql = "INSERT INTO Logs (weight) VALUES ($weight)";
if ($conn->query($sql) === TRUE)
{echo "<BR>New record created successfully";}
else
{echo "Error: " . $sql . "<br>" . $conn->error;}
//Close Connection
$conn->close();


?>
```