INDEX

TITLE **Page. No.**

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Now a day's many people posses at least a single pet. But as due to fast pace of life & long working hours pets are left out alone in the house for almost a whole day long. In this situation it becomes a great concern for the pet owners to take care of their food timings. Also in some cases when the family is out of town & the pets can't be taken along then the family have to depend on someone for their daily feed. Thus we thought of developing a Pet Feeder using IoT.

By using IoT we can look after the food timings of the pet from any location. In our system we are using a mobile application which will make it easy for the user to remotely operate the system. All the controlling of the system can be carried out using this application. Besides mobile application system also contains various hardware components like motor, load cell, analog to digital converter etc. In mentioned system we also look forward to carry out data analysis making use of cloud. Hence using this system we can not only feed the pets but also get an analysis on the application itself about the quantity of food that is being consumed by the pet.

ORGANISATION OF REPORT

In this report of project titled “PET FEEDER USING INTERNET OF THINGS” we have broadly explained about the project chapter wise. Hence this report gives a brief idea about the project & work that we have completed in this project. The report is presented in a chaptered format so as to cover all different aspects that are covered under this subject.

Chapter1: Introduction

This chapter throws light on need & scope of project. It also makes us understand the previously used system & short information on it.

Chapter 2: Literature Survey

In this chapter we aim to get a detail idea about the project topic & current trends related to the topic on which people are working. This can be done referring various sources such as different websites & research papers.

Chapter3: System Design

This chapter comprises of the block diagram & its detailed explanation also the specifications of the system.

Chapter4: Hardware Requirements

It gives detailed information about the hardware components of the system & also about the software that is being used.

Chapter5: Software Implementation

Chapter6: Result & Observations

Chapter7: Conclusion & Future Scope

References

CHAPTER 1

INTRODUCTION

In today's world of technological advancement the main motto of human beings is to bring ease to life using automation in every possible field. The spectrum of automation has a wide spread beginning from our own houses to the universe. Automation is preferred due to the advantages such as negligible manual work & précised accuracy. One such trending technology is Internet of Things.

Since old times pet feeding was carried out manually. Very initially people were not much aware about the type of diet a pets needs to be fed. They use to feed them with whatever human beings eat or the leftover food of the house. As the literacy of the mass rose gradually people came to knew about pet food & the diet that they need. Also different kinds of foods were launched in the market. But the method of pet feeding remained the same that is the food was left into the bowl & was not looked after then.

But as life is progressing, pets are now treated more like a family member or a friend. Thus the pet owners also are much more concerned about the health & dietary habits of their pets. Thus people are feeling a need for a system to feed their pets whenever they are not available for them.

This is how the concept of developing a pet feeder originated & is getting popular day by day throughout the world. Currently many students as well as researchers are working on this concept using different domains such as GSM, various microcontrollers, IoT etc.

In our project we too are making an effort to the ongoing work. Thus we opted to work with IoT domain combined with some flavors of data analysis as well. This project describes briefly about how pets can be fed without any manual process, automatically from any location using the techniques of IoT. The concept of IoT is emerging & popularizing with a large pace and has become vital factor of our lives. Using IoT in collaborations with different electronic devices and developing a friendly gadget is in trend. IoT has a greatest advantage in which we can operate our system from any place throughout world. To make an ease for user the control of the system is automated using an application & also manually as a backup in case of emergency.

CHAPTER 2

LITERATURE SURVEY

2.1) Background –

2.2) Study of Reference Papers –

Studying paper [1] helped us to understand the working of automatic pet feeder that was developed by the author using ATMEGA8 controller with DC motor & valves. This system is a very basic version of pet feeder system. It is only helpful in serving the preliminary purpose of feeding a pet that too restricted for domestic use. In paper [2] the author used PIC18 microcontroller to design a pet feeder. It was designed almost same as the one in the first paper except that with DC motor it also uses Steeper motor for turn table. The other components such as buzzer, LCD display, Knobs, power supply etc are also used in the systems in paper [1] & [2].

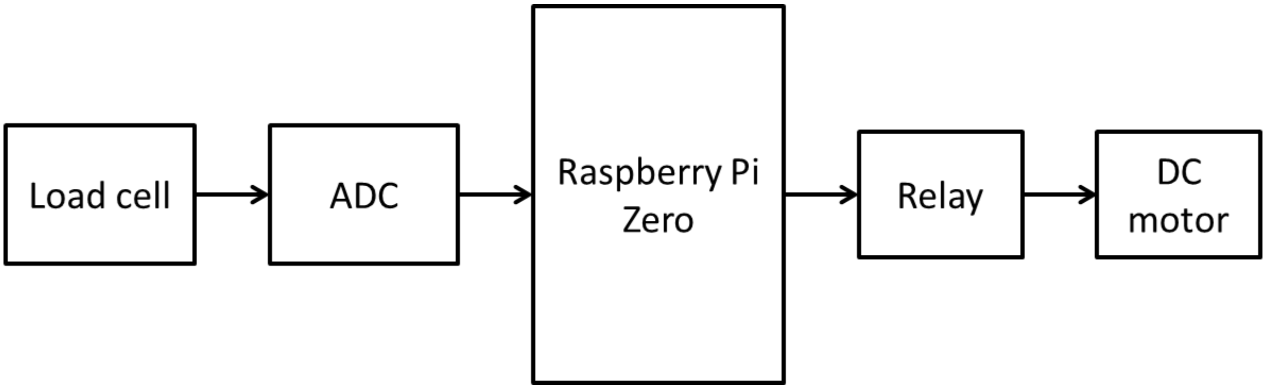
The author of paper [3] just proposed to develop a system that they were going to develop later on. This proposed design includes Arduino UNO board & a wifi module thus using these elements we can control the system when we are not at home. Rest of the components that are used are servo motor, RTC, LCD display. The concept of their system is that whenever the system senses the presence of the pet near system it disperses food. This system is having many drawbacks.

Paper [4] presents a complete pet care system developed in the domain of internet of things (IoT). The system is further divided into many sub systems that are pet door, pet food feeder & pet collar. This system makes use of Arduino controller with various other components like wifi module, motor, gps tracker. Paper [5] represents a study of proposed product that gives a brief idea about pet monitoring system which is a pet door & also a pet feeding system. The study reveals that amongst the current technologies IoT is the on which can be used to design the desired system. Paper [6] also puts a light on a proposed system that can be developed using IoT. This proposed system states development of a pet care system having sub modules like pet feeder, pet pooping pad & camera to monitor pet’s movement. Paper [7] is a remote controlled & GSM based pet feeder system. This is also a proposed system. As the name suggest the system is based on GSM module, IR remote control, 89c52 microcontroller & many other components.

CHAPTER 3

SYSTEM SCHEMATIC AND SPECIFICATION

3.1) BLOCK DIAGRAM-



3.2) Block Diagram Description-

3.2.1) Load Cell – A load cell is a transducer that is used to create an electrical signal whose magnitude is directly proportional to the force being measured. The various load cell types include hydraulic, pneumatic, and strain gauge. Except for certain laboratories where precision mechanical balances are still used, strain gage load cells dominate the weighing industry. Pneumatic load cells are sometimes used where intrinsic safety and hygiene are desired, and hydraulic load cells are considered in remote locations, as they do not require a power supply. Strain gage load cells offer accuracies from within 0.03% to 0.25% full scale and are suitable for almost all industrial applications. Load cell designs can be distinguished according to the type of output signal generated (pneumatic, hydraulic, electric) or according to the way they detect weight (bending, shear, compression, tension, etc.)

3.2.1.1) Load Cell History -

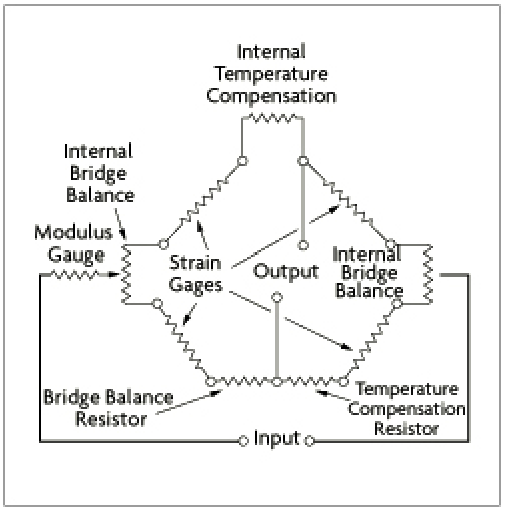
Before strain gage-based load cells became the method of choice for industrial weighing applications, mechanical lever scales were widely used. Mechanical scales can weigh everything from pills to railroad cars and can do so accurately and reliably if they are properly calibrated and maintained. The method of operation can involve either the use of a weight balancing mechanism or the detection of the force developed by mechanical levers. The earliest, pre-strain gage force sensors included hydraulic and pneumatic designs. In 1843, English physicist Sir Charles Wheatstone devised a bridge circuit that could measure electrical resistances. The Wheatstone bridge circuit is ideal for measuring the resistance changes that occur in strain gages. Although the first bonded resistance wire strain gage was developed in the 1940s, it was not until modern electronics caught up that the new technology became technically and economically feasible. Since that time, however, strain gages have proliferated both as mechanical scale components and in stand-alone load cells.

3.2.1.2) Types of Load Cell –

Hydraulic load cells are force -balance devices, measuring weight as a change in pressure of the internal filling fluid. In a rolling diaphragm type hydraulic load cell, a load or force acting on a loading head is transferred to a piston that in turn compresses a filling fluid confined within an elastomeric diaphragm chamber. As force increases, the pressure of the hydraulic fluid rises. This pressure can be locally indicated or transmitted for remote indication or control. Output is linear and relatively unaffected by the amount of the filling fluid or by its temperature. If the load cells have been properly installed and calibrated, accuracy can be within 0.25% full scale or better, acceptable for most process weighing applications. Because this sensor has no electric components, it is ideal for use in hazardous areas. Typical hydraulic load cell applications include tank, bin, and hopper weighing. For maximum accuracy, the weight of the tank should be obtained by locating one load cell at each point of support and summing their outputs.

Pneumatic load cells also operate on the force-balance principle. These devices use multiple dampener chambers to provide higher accuracy than can a hydraulic device. In some designs, the first dampener chamber is used as a tare weight chamber. Pneumatic load cells are often used to measure relatively small weights in industries where cleanliness and safety are of prime concern. The advantages of this type of load cell include their being inherently explosion proof and insensitive to temperature variations. Additionally, they contain no fluids that might contaminate the process if the diaphragm ruptures. Disadvantages include relatively slow speed of response and the need for clean, dry, regulated air or nitrogen.

Strain-gage load cells convert the load acting on them into electrical signals. The gauges themselves are bonded onto a beam or structural member that deforms when weight is applied. In most cases, four strain gages are used to obtain maximum sensitivity and temperature compensation. Two of the gauges are usually in tension, and two in compression, and are wired with compensation adjustments when weight is applied, the strain changes the electrical resistance of the gauges in proportion to the load. Other load cells are fading into obscurity, as strain gage load cells continue to increase their accuracy and lower their unit costs. Strain gauge load cells usually feature four strain gauges in a Wheatstone bridge configuration, which is an electrical circuit that balances two legs of a bridge circuit. The force being measured deforms the strain gauge in this type of load cell, and the deformation is measured as change in electrical signal. There are several common strain gauge load cell configurations, including shear beam, s-type, and compression. In this project we are using strain gauge load cell which is single point or platform strain gauge.



3.2.1.3) Calibration -

Calibration can be defined as a set of operations that compares the accuracy of a measuring instrument of any type (such as a load cell) against a recognised standard. The process of calibration may also include adjusting the measuring instrument to bring it in alignment with the standard. By calibrating measuring devices against recognised standards, they can then be traced back to show they comply with International Standards.

Kilograms–Unit Conversion

Step 1

Obtain the individual load cell's characteristics. These will usually be shown as basic resistance in ohms, recommended excitation voltage and sensitivity expressed as mV/V excitation, per unit weight in kilograms, or mill volts-per-volt, and maximum weight capacity in kilograms.

Step 2

Multiply the load cell's reading in mill volts by the full scale weight of the load cell. If the load cell's reading is 14 mill volts, (0.014-volts) and the load cell's maximum sensing weight is 500 kilograms, this multiplication will yield 7 volt-kilograms.

Step 3

Divide the product from Step 2, 7 volt-kilograms, by sensitivity in mill volts/volt times the excitation voltage. If the load cell's sensitivity is 2mV/V and the excitation voltage provided by the converter is 10 volts, then the divider will be 20 mill volts or 0.02 volts. The division operation will be 7 volt-kilograms divided by 0.02 volts, yielding a weight of 350 kilograms on the scale using the load cell. A digital weigh-scale meter would perform this calculation automatically and read out 350.0 kilograms on its indicator face.

3.2.2) ADC (HX711) – In this system we need an ADC for conversion of analog signal to digital as we need only digital signal for Raspberry Pi as an input. Here we have used standard ADC that is HX711. It converts received weight from load cell to its digital form & feeds as an input to the processor.

This module uses 24 high precision A/D converter chip HX711. It is a specially designed for the high precision electronic scale design, with two analog input channels, the internal integration of 128 times the programmable gain amplifier. The input circuit can be configured to provide a bridge type pressure bridge (such as pressure, weighing sensor mode), is of high precision, and low cost is an ideal sampling front-end module.

This Load Cell Amplifier is a small breakout board for the HX711 IC that allows you to easily read load cells to measure weight. By connecting the amplifier to your microcontroller you will be able to read the changes in the resistance of the load cell and with some calibration you'll be able to get very accurate weight measurements. This can be handy for creating your own industrial scale, process control, or simple presence detection.

The HX711 uses a two wire interface (Clock and Data) for communication. Any microcontroller's GPIO pins should work and numerous libraries have been written making it easy to read data from the HX711. Load cells use a four wire Wheatstone to connect to the HX711. Load cell is connected with HX711 Load cell Amplifier using four wires. These four wires are Red, Black, White and Green/Blue. There may be slight variation in colors of wires from module to module.

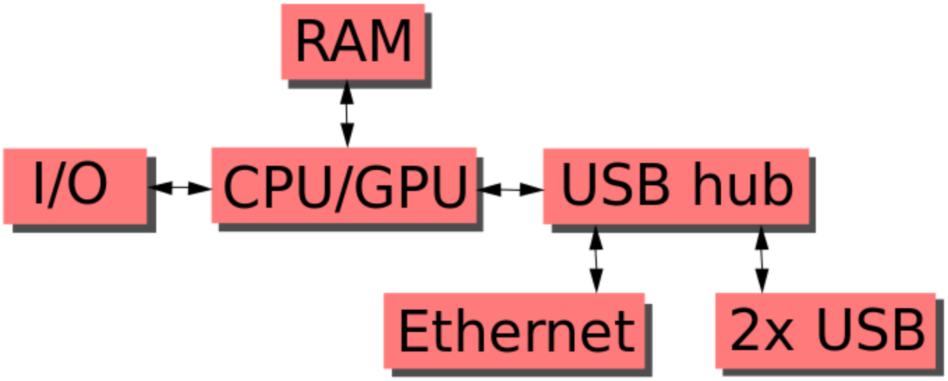
These colors correspond to the conventional color coding of load cells, where red, black, green and white wires come from the strain gauge on the load cell and yellow is an optional ground wire that is not hooked up to the strain gauge but is there to ground any small outside EMI (electromagnetic interference). Sometimes instead of a yellow wire there is a larger black wire, foil, or loose wires to shield the signal wires to lessen EMI.

Wheatstone Bridge Node	**"Typical" Wire Color**
Excitation+ (E+) or VCC	RED
Excitation- (E-) or GND	BLACK or YELLOW
Output+ (O+), Signal+ (S+), or Amplifier+ (A+)	WHITE
O-, S-, or A-	GREEN or BLUE

3.2.3) Raspberry Pi 0 –

Several generations of Raspberry Pis have been released. All models feature a Broadcom system on a chip (SoC) with an integrated ARM-compatible central processing unit (CPU) and on-chip graphics processing unit (GPU).

A Raspberry Pi Zero with smaller size and reduced input/output (I/O) and general-purpose input/output (GPIO) capabilities was released in November 2015 for US$5. By 2017, it became the newest mainline Raspberry Pi. On 28 February 2017, the Raspberry Pi Zero W was launched, a version of the Zero with Wi-Fi and Bluetooth capabilities.This block diagram describes Model B and B+; Model A, A+, and the Pi Zero are similar, but lack the Ethernet and USB hub components. The Ethernet adapter is internally connected to an additional USB port. In Model A, A+, and the Pi Zero, the USB port is connected directly to the system on a chip (SoC). On the Pi 1 Model B+ and later models the USB/Ethernet chip contains a five-port USB hub, of which four ports are available, while the Pi 1 Model B only provides two. On the Pi Zero, the USB port is also connected directly to the SoC, but it uses a micro USB (OTG) port.



The Broadcom BCM2835 SoC used in the first generation Raspberry Pi includes a 700 MHz ARM11 76JZF-S processor, Video Core IV graphics processing unit (GPU), and RAM. It has a level 1 (L1) cache of 16 KB and a level 2 (L2) caches of 128 KB. The level 2 cache is used primarily by the GPU. The SoC is stacked underneath the RAM chip, so only its edge is visible. The 1176JZ(F)-S is the same CPU used in the original iPhone, although at a higher clock rate, and mated with a much faster GPU. The Raspberry Pi Zero and Zero W use the same Broadcom BCM2835 SoC as the first generation Raspberry Pi, although now running at 1GHz CPU clock speed.

Relay - Relays are the primary protection as well as switching devices in most of the control processes or equipments. All the relays respond to one or more electrical quantities like voltage or current such that they open or close the contacts or circuits. A relay is a switching device as it works to isolate or change the state of an electric circuit from one state to another.

Classification or the types of relays depend on the function for which they are used. Some of the categories include protective, reclosing, regulating, auxiliary and monitoring relays.

Chapter 4

Hardware Implementation

4.1)Raspberry pi Zero-

Raspberry pi Zero is a super computer which is a processor in tiny way. It helps to work same as like CPU and performs the same functions and in less cost. At 65mm x 30mm, the Raspberry pi Zero is smaller than a credit card, and it’s only 5mm thick to boot. It is powerful than many other smart devices. The raspberry pi Zero runs on the exact same single-core Broadcom BCM2835.We has to provide our own storage in the form of a micro SD card, which can be inserted in the supplied slot. The Raspberry Pi Zero runs on Raspbian operating system, Raspbian is free to download, Raspbian also grants access to the Pi Store, which is essentially an app store filled with thousands of ‘packages’. To raspberry we can also connect various portable devices and also camera known as Pi camera. Raspberry Pi can be connected to monitor through HDMI or through wireless connection also. And all the specific Features are covered in Specifications

Specifications :

- 1GHz, single-core CPU.
- 512MB RAM.
- Mini HDMI and USB On-The-Go ports.
- Micro USB power.
- HAT-compatible 40-pin header.
- Composite video and reset headers.
- CSI camera connector.
- 802.11n wireless LAN.
- Bluetooth 4.0.

Fig 4.1

4.2) Load Cell:

A load cell is a transducer that is used to create an electrical signal whose magnitude is directly proportional to the force being measured. The various load cell types include hydraulic, pneumatic, and strain gauge.

Strain gauge load cells are the most common in industry. These load cells are particularly stiff, have very good resonance values, and tend to have long life cycles in application. Strain gauge load cells work on the principle that the strain gauge (a planar resistor) deforms when the material of the load cells deforms appropriately. Deformation of the strain gauge changes its electrical resistance, by an amount that is proportional to the strain. The change in resistance of the strain gauge provides an electrical value change that is calibrated to the load placed on the Load cell. The electrical signal output is typically in the order of a few milli volts (mV) and requires amplify

4.3) Analog to Digital Converter:

ADC is a device which can take an analog signal, such as an electrical current, and digitalize it into a binary format that the computer can understand. It is therefore required to define the rate at which new digital values are sampled from the analog signal. The values can be reproduced at the output using interpolation formula. Mainly there are two steps for the analog to digital conversion: S/H: Sampling and holding & Q/E: Quantizing and Encoding.Based on Avia Semiconductor’s patented technology, HX711 is a precision 24-bit analog to-digital converter (ADC) designed for weigh scales and industrial control applications to interface directly with a bridge sensor. The input multiplexer selects either Channel A or B differential input to the low-noise programmable gain amplifier (PGA). Channel A can be programmed with a gain of 128 or 64, corresponding to a full-scale differential input voltage of ±20mV or ±40mV respectively, when a 5V supply is connected to AVDD analog power supply pin. Channel B has a fixed gain of 32. On chip power supply regulator eliminates the need for an external supply regulator to provide analog power for the ADC and the sensor. Clock input is flexible. It can be from an external clock source, a crystal, or the on-chip oscillator that does not require any external component. On-chip power on-reset circuitry simplifies digital interface initialization. There is no programming needed for the internal registers. All controls to the HX711 are through the pins.

Specifications:

- Two selectable differential input channels.
- On-chip active low noise PGA with selectable gain of 32, 64 and 128.
- On-chip power supply regulator for load-cell and ADC analog power supply.
- On-chip oscillator requiring no external component with optional external crystal.
- On-chip power-on-reset Selectable 10SPS or 80SPS output data rate.
- Simultaneous 50 and 60Hz supply rejection.
- Current consumption including on-chip analog power supply regulator: normal operation < 1.5mA, power down < 1uA.
- Operation supply voltage range: 2.6 ~ 5.5V • Operation temperature range: -40 ~ +85℃.

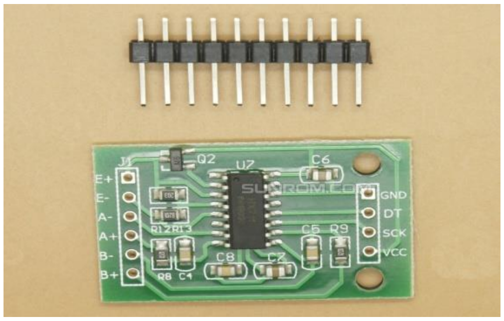


Fig 4.3

4.4) DC Motor:

A DC motor is any of a class of rotary electrical machines that converts direct current electrical energy into mechanical energy. The most common types rely on the forces produced by magnetic fields. Nearly all types of DC motors have some internal mechanism, either electromechanical or electronic; to periodically change the direction of current flow in part of the motor. A DC motor's speed can be controlled over a wide range, using either a variable supply voltage or by changing the strength of current in its field windings. Small DC motors are used in tools, toys, and appliances. The universal motor can operate on direct current but is a lightweight brushed motor used for portable power tools and appliances. Larger DC motors are used in propulsion of electric vehicles, elevator and hoists, or in drives for steel rolling mills.

The advent of power electronics has made replacement of DC motors with AC motors possible in many applications.

Specifications:

- DC supply: 4 to 12V.
- RPM: 200 at 12V.
- Total length: 46mm.
- Motor diameter: 36mm.
- Motor length: 25mm.
- Brush type: Precious metal
- Gear head diameter: 37mm
- Gear head length: 21mm
- Output shaft: Centered
- Shaft diameter: 6mm
- Gear assembly: Spur
- Shaft length: 22mm

Fig 4.4

4.5) Relay-

A relay is an electrically operated switch. Many relays use an electromagnet to mechanically operate a switch, but other operating principles are also used, such as solid-state relays. Relays are used where it is necessary to control a circuit by a separate low-power signal, or where several circuits must be controlled by one signal. Relays were used extensively in telephone exchanges and early computers to perform logical operations. Solid-state relays control power circuits with no moving parts, instead using a semiconductor device to perform switching. Relays with calibrated operating characteristics and sometimes multiple operating coils are used to protect electrical circuits from overload or faults; in modern electric power systems these functions are performed by digital instruments still called "protective relays". Magnetic latching relays require one pulse of coil power to move their contacts in one direction, and another, redirected pulse to move them back. Repeated pulses from the same input have no effect. Magnetic latching relays are useful in applications where interrupted power should not be able to transition the contacts. Since relays are switches, the terminology applied to switches is also applied to relays; a relay switches one or more poles, each of whose contacts can be thrown by energizing the coil. Normally open (NO) contacts connect the circuit when the relay is activated; the circuit is disconnected when the relay is inactive. Normally closed (NC) contacts disconnect the circuit when the relay is activated; the circuit is connected when the relay is inactive. All of the contact forms involve combinations of NO and NC connections.

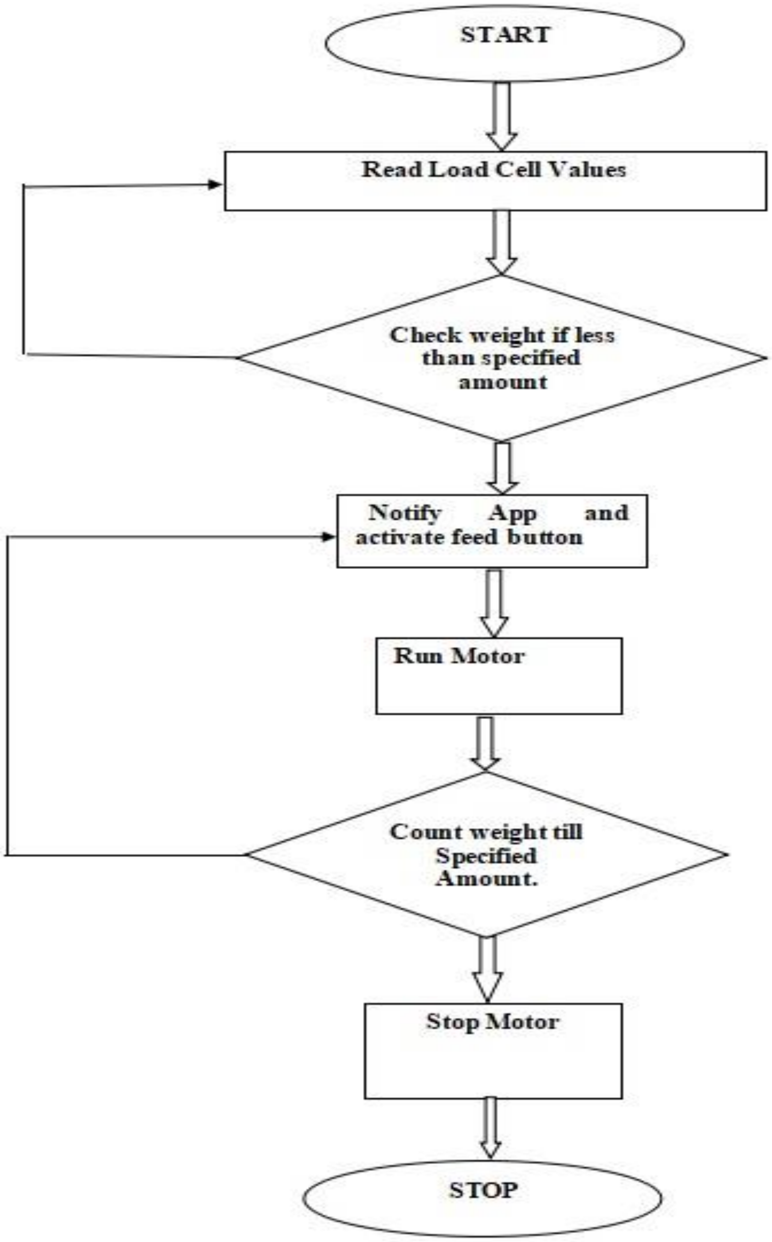
SONGLE
CQC
10A 250VAC 10A 125VAC
10A 30VDC 10A 28VDC
SRD-12VDC-SL-C

Fig 4.5

Chapter 5

Software Implementation

5.1)Flow Chart-


START
Read Load Cell Values
Check weight if less than specified amount
Notify App and activate feed button
Run Motor
Count weight till Specified Amount.
Stop Motor
STOP

5.2) Algorithm-

1. Initialize the system.
2. Interpret Load Cell Values
3. Verify Weight
4. Notify on App if weight is appropriate.
5. When weight is accurate run motor.
6. As soon as weight is accurate.
7. Stop Motor.

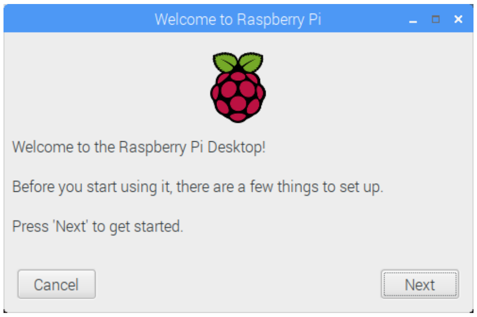
1.1 Raspbian :

Raspbian is a Debian-based computer operating system for Raspberry Pi. There are several versions of Raspbian including Raspbian Stretch and Raspbian Jessie. Since 2015 it has been officially provided by the Raspberry Pi Foundation as the primary operating system for the family of Raspberry Pi single-board computers. Raspbian was created by Mike Thompson and Peter Green as an independent project. The initial build was completed in June 2012. The operating system is still under active development. Raspbian is highly optimized for the Raspberry Pi line's low-performance ARM CPUs.

Raspbian uses PIXEL, Pi Improved X-Window Environment, Lightweight as its main desktop environment as of the latest update. It is composed of a modified LXDE desktop environment and the Open box stacking window manager with a new theme and few other changes. The distribution is shipped with a copy of computer algebra program Mathematica and a version of Minecraft called Minecraft Pi as well as a lightweight version of Chromium as of the latest version.

Setup Wizard

To install Raspbian on to your Raspberry Pi’s SD card is bit of difficult. Use of the standard installation, or rely on NOOBS offers an easy setup. There’s the Pi Bakery, for configuring your Raspbian installation before the image is written to SD card, but other than that, there’s a bit of tweaking required.


Welcome to Raspberry Pi

Welcome to the Raspberry Pi Desktop!

Before you start using it, there are a few things to set up.

Press 'Next' to get started.

With the June 2018 release of Raspbian comes the Setup Wizard, which lets set your country and language, enter a new password, connect to a Wi-Fi network, and check for updates. Not only does this give you the best possible user experience, it also gets you online quickly, and overcomes that famous Raspberry Pi security issue: the pi/raspberry username and password combination.

Pi Compatibility

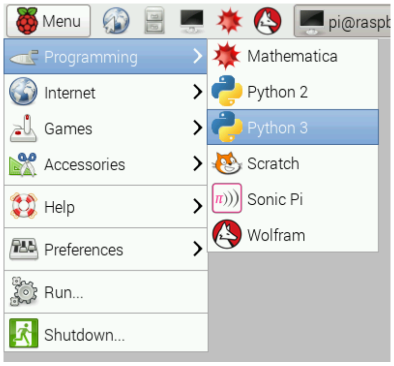
Several different models of Raspberry Pi and the operating system runs well on all of them. Due to hardware differences, however, some applications don't run on lower-spec models. For instance, the Raspberry Pi Zero and original Raspberry Pi cannot run the Chromium browser (although several alternatives are available).This means that instructions for performing tasks, building projects, whatever you're doing, are the same regardless of Pi model.

5.2) Python-

Python is a wonderful and powerful programming language that's easy to use (easy to read and write) and with Raspberry Pi lets you connect your project to the real world.
Python syntax is very clean, with an emphasis on readability and uses standard English keywords.
Start by opening IDLE from the desktop.

IDLE

The easiest introduction to Python is through IDLE, a Python development environment. Open IDLE from the Desktop or applications menu:



IDLE gives you a REPL (Read-Evaluate-Print-Loop), which is a prompt you can enter Python commands into. Because it's a REPL, you even get the output of commands printed to the screen without using print.

Note: two versions of Python are available — Python 2 and Python 3. Python 3 has was first released in 2008 and Python 2 development ended with 2.7, which was released in 2010. Python 3 is recommended, but Python 2 is available for legacy applications which do not support Python 3 yet.

You can use variables if you need to but you can even use it like a calculator. For example:

```
>>> 1 + 2
3
>>> name = "Sarah"
>>> "Hello " + name
'Hello Sarah'
```

IDLE also has syntax highlighting built in and some support for autocompletion. You can look back on the history of the commands you've entered in the REPL with Alt + P (previous) and Alt + N (next).

Python files in IDLE

To create a Python file in IDLE, click File > New File and you'll be given a blank window. This is an empty file, not a Python prompt. You write a Python file in this window, save it, then run it and you'll see the output in the other window.

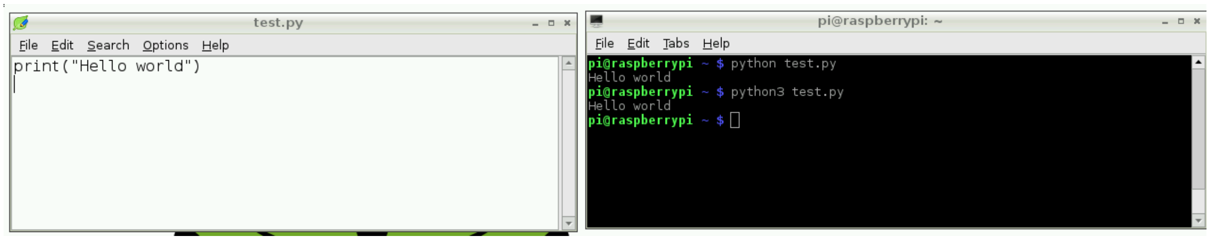
For example, in the new window, type:

```
n = 0

for i in range(1, 101):
    n += i

print("The sum of the numbers 1 to 100 is:")
print(n)
```

Then save this file (File > Save or Ctrl + S) and run (Run > Run Moduleor hit F5) and you'll see the output in your original Python window.

Executing Python files from the command line

You can write a Python file in a standard editor like Vim, Nano, or LeafPad, and run it as a Python script from the command line. Just navigate to the directory the file is saved in (use cd and ls for guidance) and run with python3, e.g. python3 hello.py.



```
print("Hello world")
```

```
pi@raspberrypi ~ $ python test.py
Hello world
pi@raspberrypi ~ $ python3 test.py
Hello world
pi@raspberrypi ~ $ 
```

Other ways of using Python

Command Line

The standard built-in Python shell is accessed by typing python3 in the terminal.

This shell is a prompt ready for Python commands to be entered. You can use this in the same way as IDLE, but it does not have syntax highlighting or autocompletion. You can look back on the history of the commands you've entered in the REPL by using the Up/Down keys. Use Ctrl + D to exit.

IPython

IPython is an interactive Python shell with syntax highlighting, autocompletion, pretty printing, built-in documentation, and more. IPython is not installed by default. Install with:

```
sudo pip3 install ipython
```

Then run with ipython from the command line. It works like the standard python3, but has more features. Try typing len? and hitting Enter. You're shown information including the docstring for the len function:

```
Type:       builtin_function_or_method
String Form:<built-in function len>
Namespace:  Python builtin
Docstring:
len(object) -> integer

Return the number of items of a sequence or mapping.
```

Try the following dictionary comprehension:

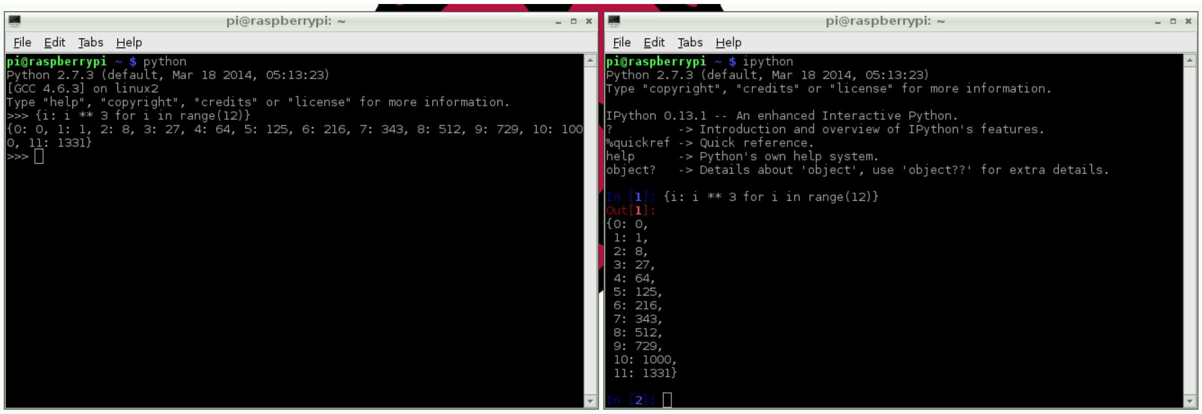
```
{i: i ** 3 for i in range(12)}
```

This will pretty print the following:

```
{0: 0,
 1: 1,
 2: 8,
 3: 27,
 4: 64,
 5: 125,
 6: 216,
 7: 343,
 8: 512,
 9: 729,
 10: 1000,
 11: 1331}
```

In the standard Python shell, this would have printed on one line:

```
{0: 0, 1: 1, 2: 8, 3: 27, 4: 64, 5: 125, 6: 216, 7: 343, 8: 512, 9: 729, 10: 1000, 11: 1331}
```



```
pi@raspberrypi ~ $ python
Python 2.7.3 (default, Mar 18 2014, 05:13:23)
[GCC 4.6.3] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> {i: i ** 3 for i in range(12)}
{0: 0, 1: 1, 2: 8, 3: 27, 4: 64, 5: 125, 6: 216, 7: 343, 8: 512, 9: 729, 10: 100
0, 11: 1331}
>>> 
```

```
pi@raspberrypi ~ $ ipython
Python 2.7.3 (default, Mar 18 2014, 05:13:23)
Type "copyright", "credits" or "license" for more information.

IPython 0.13.1 -- An enhanced Interactive Python.
?         -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help      -> Python's own help system.
object?   -> Details about 'object', use 'object??' for extra details.

In [1]: {i: i ** 3 for i in range(12)}
Out[1]:
{0: 0,
 1: 1,
 2: 8,
 3: 27,
 4: 64,
 5: 125,
 6: 216,
 7: 343,
 8: 512,
 9: 729,
 10: 1000,
 11: 1331}

In [2]: 
```

You can look back on the history of the commands you've entered in the REPL by using the Up/Down keys like in python. The history also persists to the next session, so you can exit ipython and return (or switch between v2/3) and the history remains. Use Ctrl + D to exit.

Installing Python libraries

apt

Some Python packages can be found in the Raspbian archives, and can be installed using apt, for example:

sudo apt update

sudo apt install python-picamera

This is a preferable method of installing, as it means that the modules you install can be kept up to date easily with the usual sudo apt update and sudo apt upgrade commands.

Pip

Not all Python packages are available in the Raspbian archives, and those that are can sometimes be out of date. If you can't find a suitable version in the Raspbian archives, you can install packages from the Python Package Index (known as PyPI).

To do so, install pip:

sudo apt install python3-pip

Then install Python packages (e.g. simplejson) with pip3:

```
sudo pip3 install simplejson
```

The official Python Package Index (PyPI) hosts files uploaded by package maintainers. Some packages require compilation (compiling C/C++ or similar code) in order to install them, which can be a time-consuming task, particlarly on the single-core Raspberry Pi 1 or Pi Zero.

piwheels is a service providing pre-compiled packages (called Python wheels) ready for use on the Raspberry Pi. Raspbian Stretch is pre-configired to use piwheels for pip. Read more about the piwheels project at www.piwheels.org.

GPIO in Python

Using the GPIO Zero library makes it easy to get started with controlling GPIO devices with Python.

LED

Run this in an IDE like IDLE or Thonny, and the LED will blink on and off repeatedly.

LED methods include on(), off(), toggle(), and blink().

Button

To control an LED connected to GPIO17, you can use this code:

```
from gpiozero import LED
from time import sleep

led = LED(17)

while True:
    led.on()
    sleep(1)
    led.off()
    sleep(1).
```

To read the state of a button connected to GPIO2, you can use this code:

```
from gpiozero import Button
from time import sleep

button = Button(2)

while True:
    if button.is_pressed:
        print("Pressed")
    else:
        print("Released")
    sleep(1)
```

Button functionality includes the properties is_pressed and is_held; callbacks when_pressed, when_released, and when_held; and methods wait_for_press() and wait_for_release.

Button + LED

To connect the LED and button together, you can use this code:

```
from gpiozero import LED, Button

led = LED(17)
button = Button(2)

while True:
    if button.is_pressed:
        led.on()
    else:
        led.off()
```

Alternatively:

```
from gpiozero import LED, Button
```

```
led = LED(17)
button = Button(2)

while True:
    button.wait_for_press()
    led.on()
    button.wait_for_release()
    led.off()
```

or:

```
from gpiozero import LED, Button

led = LED(17)
button = Button(2)

button.when_pressed = led.on
button.when_released = led.off
```

MIT App Inventer

App Inventor for Android is an open-source web application originally provided by Google, and now maintained by the Massachusetts Institute of Technology (MIT), which allows to create software applications for the Android operating system (OS).It uses a graphical interface very similar to Scratch and the StarLogo TNG user interface, which allows us to drag-and-drop visual objects to create an application that can run on Android devices. In creating App Inventor, Google drew upon significant prior research in educational computing, as well as work done within Google on online development environments.

App Inventor and the projects on which it is based are informed by constructionist learning theories, which emphasizes that programming can be a vehicle for engaging powerful ideas through active learning.App Inventor also supports the use of cloud data via an experimental FirebaseDB component.

Getting Started with MIT App Inventor 2

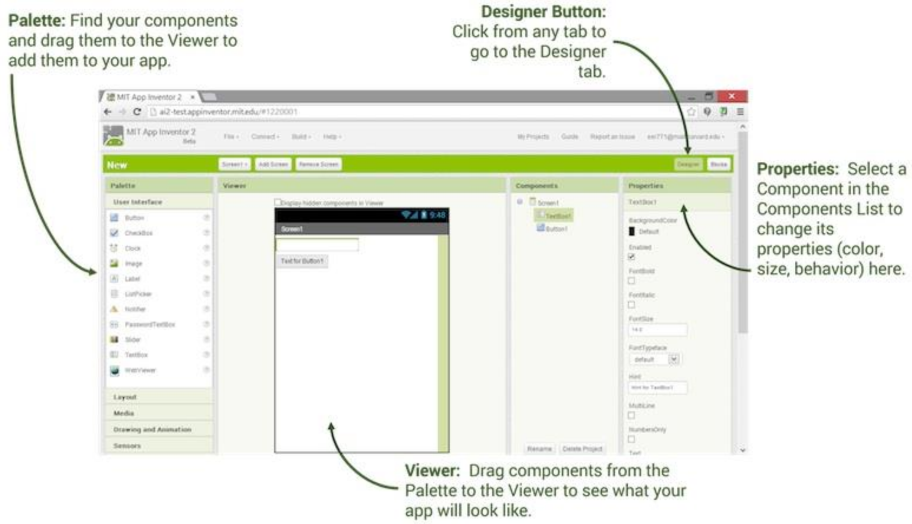
App Inventor is a cloud-based tool, which means you can build apps right in your web browser. This website offers all the support you'll need to learn how to build your own apps. Visit it at ai2.appinventor.mit.edu. You can get there by clicking the orange "Create Apps!" button from any page on this website.

Setup Instructions:Set up your phone or tablet for live testing (or,start the emulator if you don't have a mobile device)

Designer and Blocks Editor Overview: Take a tour of the App Inventor environment

Designer and Blocks Editor

App Inventor consists of the **Designer** and the **Blocks Editor**. Designer editor works for the UI of the application whisch us also known as front end development of the application . The Blocks in the App Inventor are the set of commands and Expressions. These are described in detail below.



Palette: Find your components and drag them to the Viewer to add them to your app.

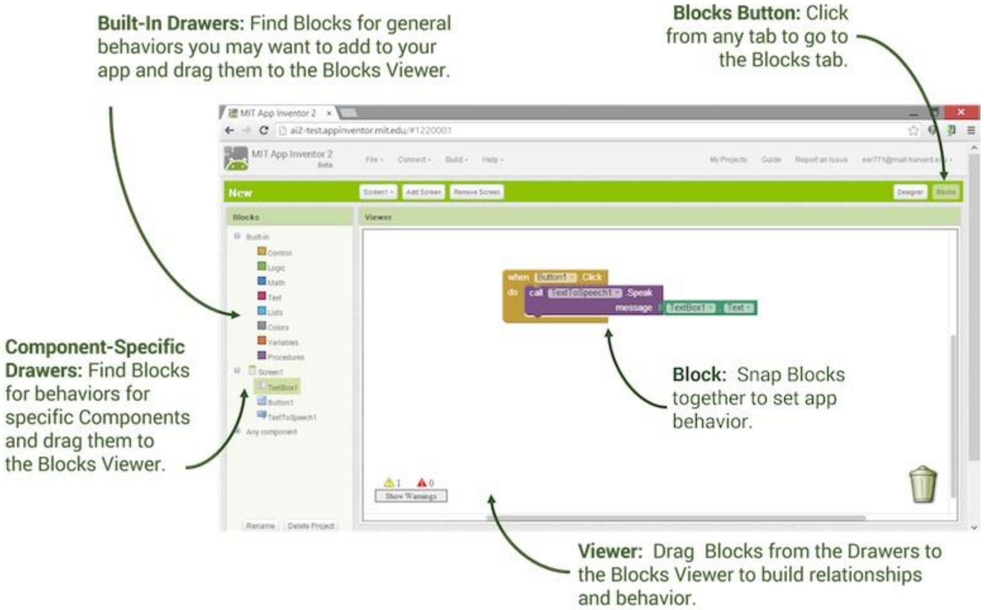
Designer Button: Click from any tab to go to the Designer tab.

Properties: Select a Component in the Components List to change its properties (color, size, behavior) here.

Viewer: Drag components from the Palette to the Viewer to see what your app will look like.

App Inventor Blocks Editor

Program the app's behavior by putting blocks together.



Built-In Drawers: Find Blocks for general behaviors you may want to add to your app and drag them to the Blocks Viewer.

Blocks Button: Click from any tab to go to the Blocks tab.

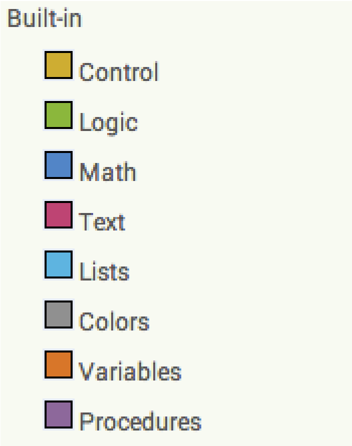
Component-Specific Drawers: Find Blocks for behaviors for specific Components and drag them to the Blocks Viewer.

Block: Snap Blocks together to set app behavior.

Viewer: Drag Blocks from the Drawers to the Blocks Viewer to build relationships and behavior.

App Inventor Built-in Blocks

Built-in blocks are available regardless of which components are in your project. In addition to these *language blocks*, each component in your project has its own set of blocks specific to its own events, methods, and properties. This is an overview of all of the Built-In Blocks available in the Blocks Editor.



Built-in

- 🟨 Control
- 🟩 Logic
- 🟦 Math
- 🟥 Text
- 🟦 Lists
- ⬛ Colors
- 🟧 Variables
- 🟪 Procedures

Fire Base

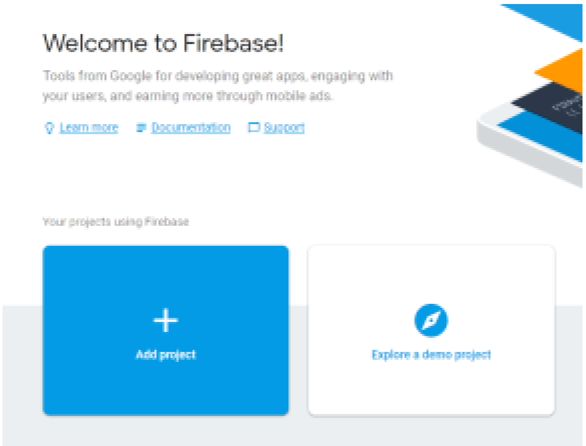
Firebase is a mobile and web application development platform developed by Firebase, In 2011. Envolve provided developers an API that enables the integration of online chat functionality into their websites. After releasing the chat service, they found that it was being used to pass application data that weren't chat messages. Developers were using Envolve to sync application data such as game state in real time across their users. Separate chat system and the real-time architecture that powered it. There first product is Firebase Real-time Database, an API that synchronizes application data across iOS, Android, and Web devices, and stores it on Firebase's cloud. The product assists software developers in building real-time, collaborative applications.

How to use firebase with MIT app inventor 2.

Integrating firebase with app inventor is straightforward and it can be done in 3 simple steps.

1. Create real-time database

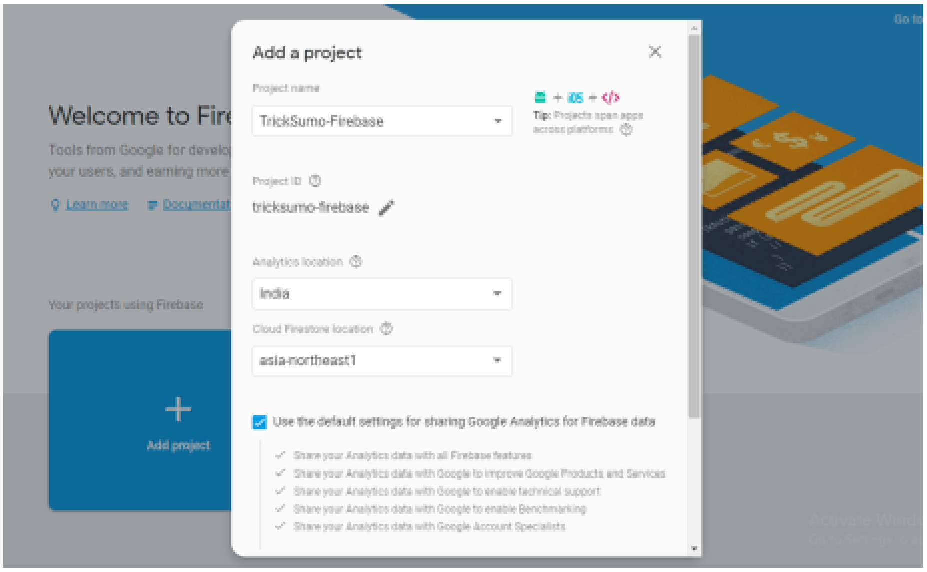
Navigate to Google Firebase and login using your google account. It will redirect you to Google firebase console.



Welcome to Firebase!

Tools from Google for developing great apps, engaging with your users, and earning more through mobile ads.

Click on new project, enter a project name and cloud firestore location, then hit create project button.



Add a project

Project name: TrickSumo-Firebase

Tip: Projects span apps across platforms

Project ID: tricksumo-firebase

Analytics location: India

Cloud Firestore location: asia-northeast1

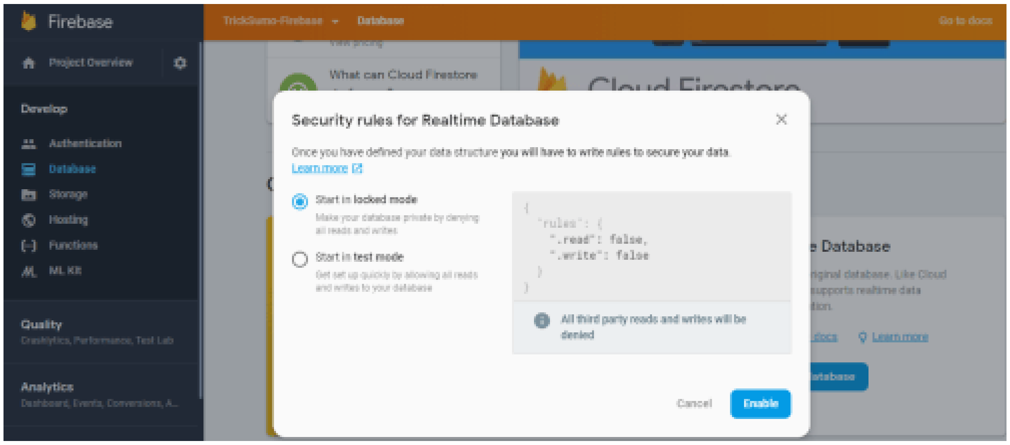
☑ Use the default settings for sharing Google Analytics for Firebase data

- ✓ Share your Analytics data with all Firebase features
- ✓ Share your Analytics data with Google to improve Google Products and Services
- ✓ Share your Analytics data with Google to enable technical support
- ✓ Share your Analytics data with Google to enable Benchmarking
- ✓ Share your Analytics data with Google Account Specialists

Welcome to Fire
Tools from Google for develo
your users, and earning more
Learn more
Documentat
Your projects using Firebase
Add project
Go to

earch for **Develop** option in the left panel, then expand Develop and select **Database**. It will open the firebase database catalog. Scroll down to select real-time database and click on **create database** button.

Keep Security rules for Real-time Database in locked mode and enable it.



Security rules for Realtime Database

Once you have defined your data structure you will have to write rules to secure your data.
Learn more

● Start in locked mode
Make your database private by denying all reads and writes

○ Start in test mode
Get set up quickly by allowing all reads and writes to your database

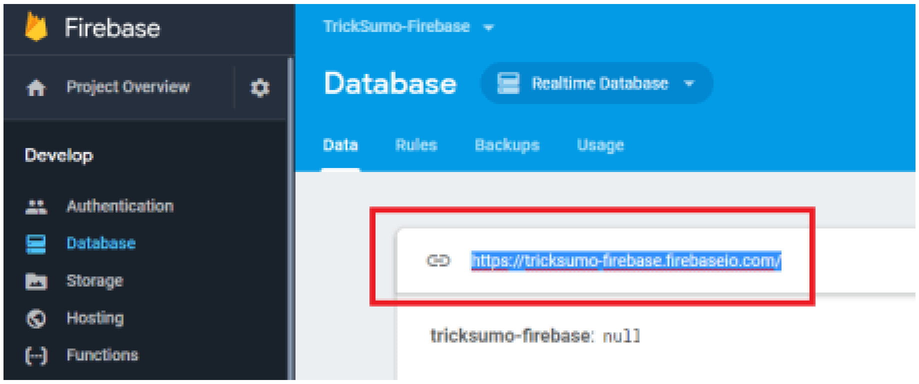
```
{
  "rules": {
    ".read": false,
    ".write": false
  }
}
```

All third party reads and writes will be denied

Now your database is enabled and online. Its time to find out secret database details (URL + KEY) so that you can access it remotely.

2. Find real-time database authorization details

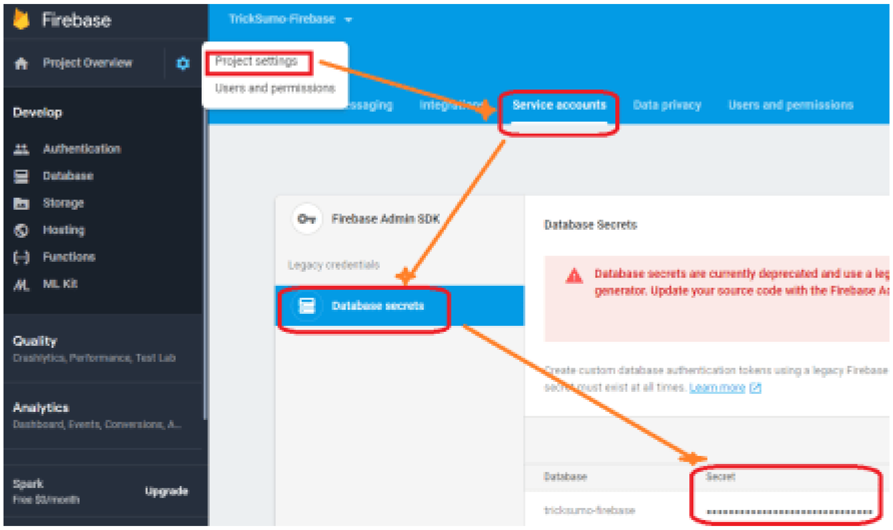
Open your database and copy Real-time database URL. Paste the link to firebase database on the notepad.



https://tricksumo-firebase.firebaseio.com/

tricksumo-firebase: null

Click on Project Overview option on the left panel and select project settings. Navigate to **Service accounts** >> **Database secrets** >> **Secret**. Copy the database authentication key and save in a safe location.



Database secrets are currently deprecated and use a leg
generator. Update your source code with the Firebase A

Create custom database authentication tokens using a legacy Firebase secret must exist at all times. Learn more

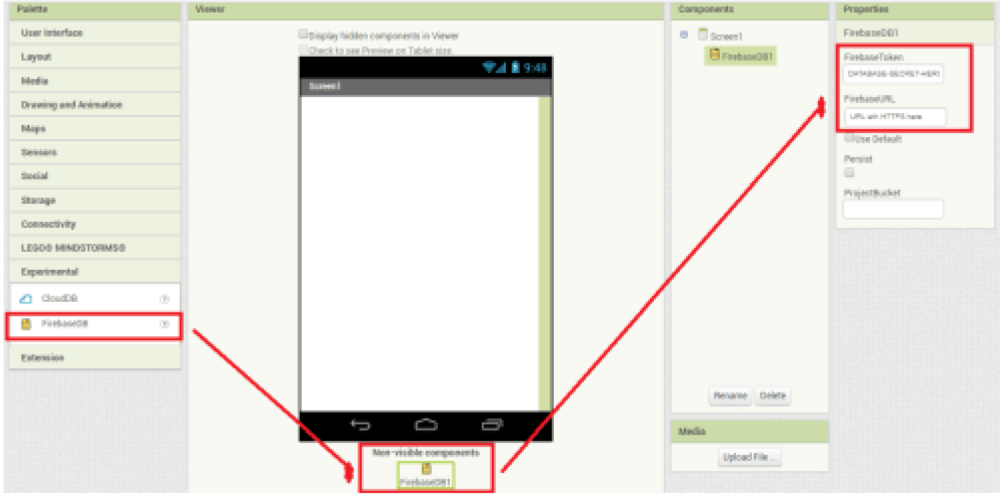
Database	Secret
tricksumo-firebase	••••••••••••••••••••••••••••••••

3. Integrate firebase with MIT app inventor

First of all, go to MIT app inventor home page, click on create apps and login using valid credentials.

Find the “Experimental” option (located in the left palette window) and expand it. Drag and drop “FirebaseDB” in the workspace (viewer window).

After that, create a new project and give it any desired name.

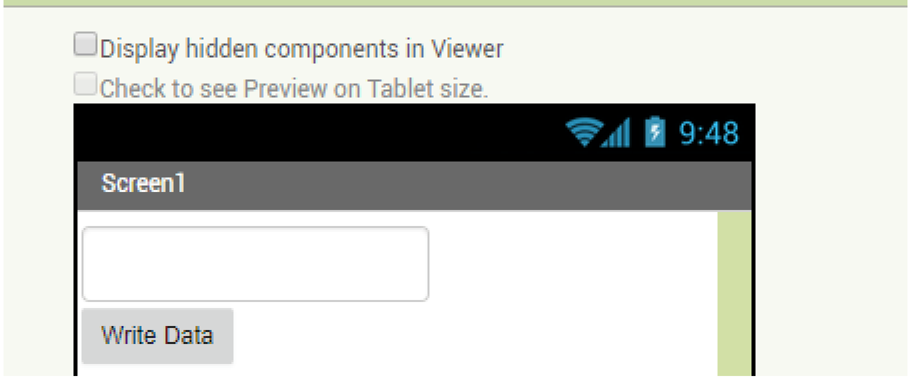


That' it, now you are ready to use google firebase with MIT app inventor. Now, let's dig a little bit deeper in how to read/write data from/to firebase using app inventor.

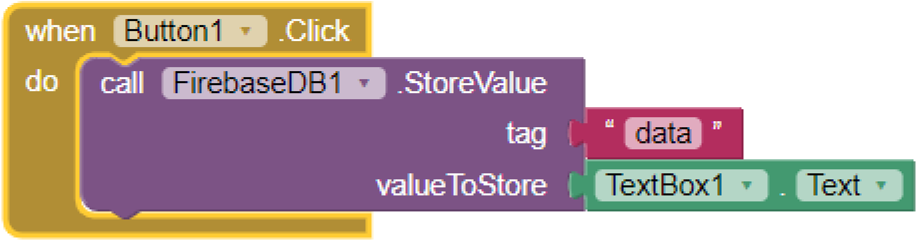
To Write Data To Firebase Using MIT App Inventor?

As we say, the best way to learn something is to do it (Do it Lean it), let's create a simple app which can write data to firebase. At this point, I am assuming that you had already integrated firebase with app inventor. Now we need to add some other components.

Drag and drop a TextBox and a Button to the workspace. Click on the button and give it a name (let it be "Write Data").

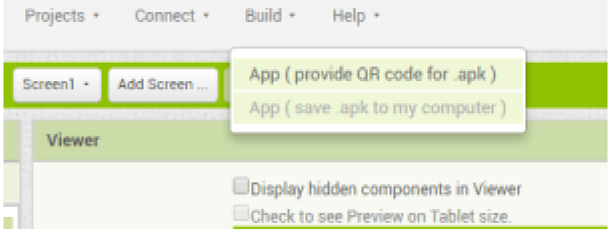

□ Display hidden components in Viewer
□ Check to see Preview on Tablet size.
9:48
Screen1
Write Data

After that, click on BLOCKS option located at the top right corner of Windows. This is workspace where we will write our logic. Create logic as shown below.



```
when Button1 .Click
do  call FirebaseDB1 .StoreValue
                         tag  " data "
                valueToStore  TextBox1 . Text
```

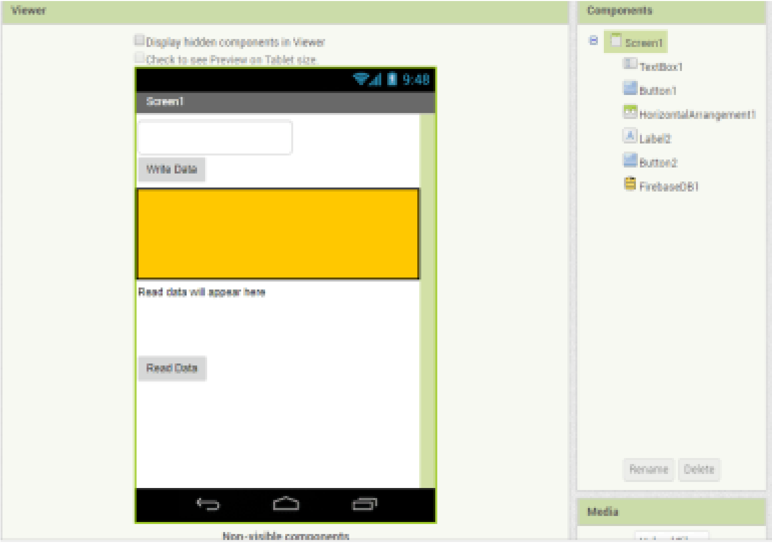
Then click on build and save apk file to your PC, transfer file to smartphone and install on it.


Projects ▾
Connect ▾
Build ▾
Help ▾
Screen1 ▾
Add Screen ...
App (provide QR code for .apk)
App (save .apk to my computer)
Viewer
□Display hidden components in Viewer
□Check to see Preview on Tablet size.

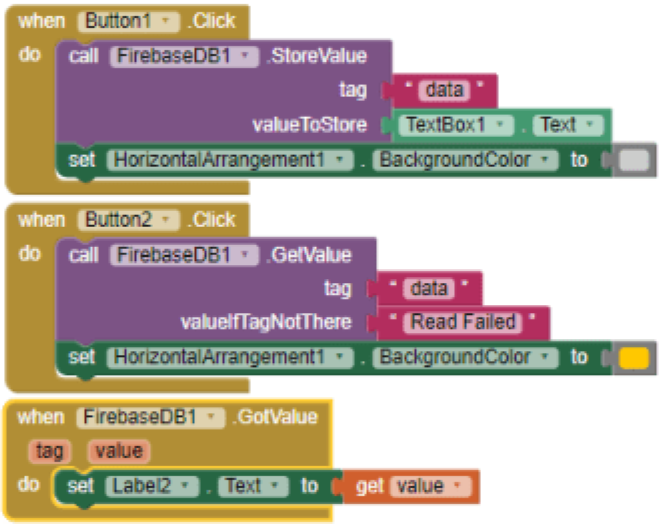
Run the app, enter something in the text field and click the button. Your data would be saved to firebase. You can view saved data by visiting Google firebase console.

How To Read Data From Firebase Using MIT App Inventor?

Drag a label and second button (let it be Read Data).



Add code blocks as shown below. Save it and run on any android device or android emulator.



```
when Button1 .Click
do  call FirebaseDB1 .StoreValue
                        tag  " data "
               valueToStore  TextBox1 . Text
    set HorizontalArrangement1 . BackgroundColor to
when Button2 .Click
do  call FirebaseDB1 .GetValue
                        tag  " data "
           valueIfTagNotThere  " Read Failed "
    set HorizontalArrangement1 . BackgroundColor to
when FirebaseDB1 .GotValue
 tag  value
do  set Label2 . Text to  get value
```

After that, go to console and edit value manually or write any value using app. Then press **“read data”** button and it will fetch and show new data in label.

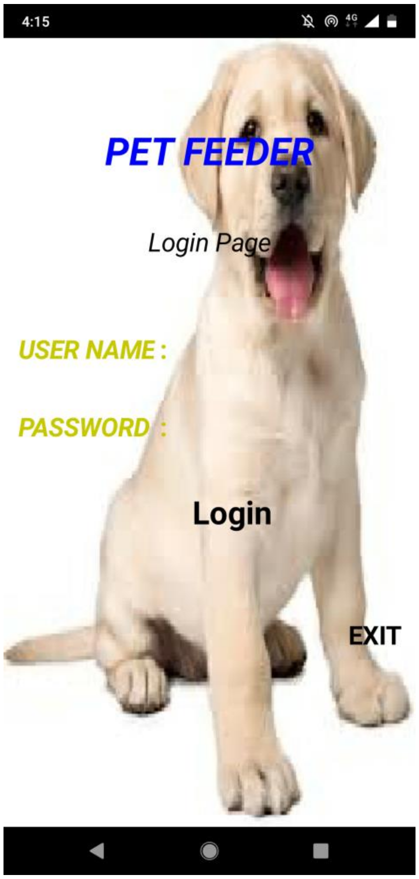
In this way you can update or fetch data from firebase. Thus we used it for developing app and IOT based pet feeder project.

Chapter 6

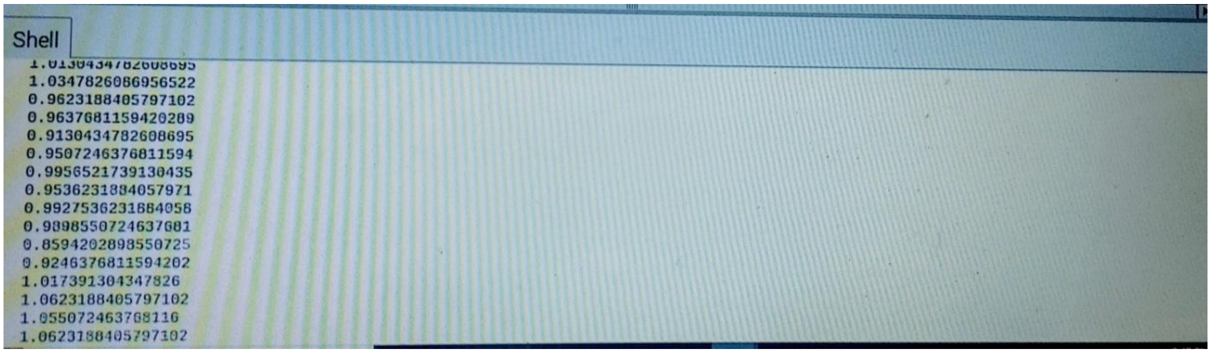
Result & Observation

6.1) Results –

6.1.1) Application -

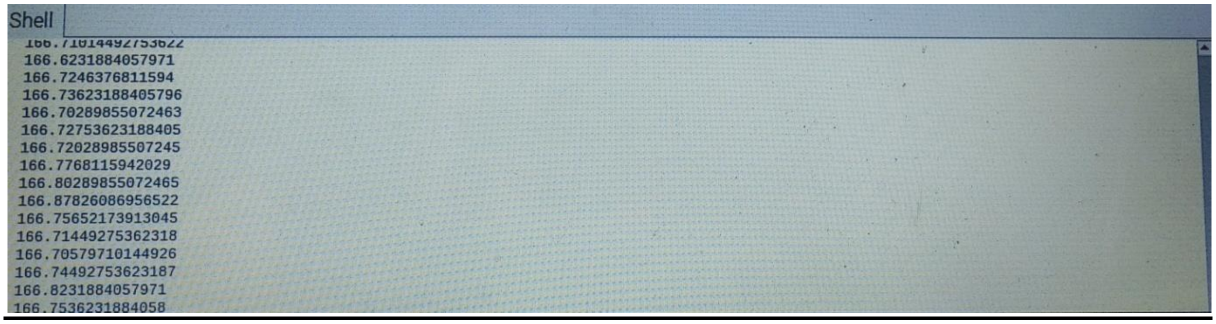


6.1.2. Load Cell –



```
 1.0130434782608695
 1.0347826086956522
 0.9623188405797102
 0.9637681159420289
 0.9130434782608695
 0.9507246376811594
 0.9956521739130435
 0.9536231884057971
 0.9927536231884058
 0.9898550724637681
 0.8594202898550725
 0.9246376811594202
1.017391304347826
1.0623188405797102
1.055072463768116
1.0623188405797102
```

Initial value



```
 166.71014492753622
 166.6231884057971
 166.7246376811594
 166.73623188405796
 166.70289855072463
 166.72753623188405
 166.72028985507245
 166.7768115942029
 166.80289855072465
 166.87826086956522
 166.75652173913045
 166.71449275362318
 166.70579710144926
 166.74492753623187
 166.8231884057971
 166.7536231884058
```

Final Value

<u>6.2) Observation-</u> In this project we observed that on the basis of IoT domain we have implemented pet feeder which is able feed the pet though we are at remote places. During this project t we learnt how to interface raspberry pi with relay, dc motor & load cell with calibration. Also we learnt to build Android Application using MIT APP INVENTOR.

Chapter 7

Future Scope & Conclusion

7.1 Future Scope- The product can be used for the surveillance and the control of pets by additional features of the CCTV camera and audio speaker by which the instructions to the pets can be conveyed remotely .Also it is useful in various poultries , aquariums, pet houses etc.

Conclusion-

The project describes detailed explanation of android based smart pet feeder. The use of motors proves clear utilization of simple devices which can perform complex operations. All these simple devices and user friendly android application make the product very reliable and cost effective.

REFERENCES

[1] AUTOMATIC PET FEEDER, MANOJ M, International Journal of Advances in Science Engineering and Technology, Volume- 3, Issue-3, Special Issue Sept.-2015.

[2] Programmable Pet Feeder, Tessema Gelila Berhan1 , Worku Toyiba Ahemed2 , Tessema Zelalem Birhan, International Journal of Scientific Engineering and Research (IJSER) Volume 3 Issue 11, November 2015.

[3] Automatic Pet Feeder Using Arduino, Mritunjay Subhashchandra Tiwari, Sahil Manoj Hawal , Nikhil Navanath Mhatre , Akshay Ramesh Bhosale,Mainak Bhaumik, International Journal of Innovative Research in Science, Engineering and Technology, Vol. 7, Issue 3, March 2018.

[4] Automatic Pet Monitoring and Feeding System Using IoT S.Subaashri, M.Sowndarya , D.K.S. Sowmiyalaxmi , S.V.Sivassan, C. Rajasekaran, International Journal of ChemTech Research Vol.10 No.14,2017.

[5] The Study and Application of the IoT in Pet Systems Chung-Ming Own , Haw-Yun Shin , Chen-Ya Teng Advances in Internet of Things, 2013.

[6] Smart Pet Care System using Internet of Things, Seungcheon Kim, International Journal of Smart Home Vol. 10, No. 3, (2016).

[7] REMOTE CONTROLLED AND GSM BASED AUTOMATED PET FEEDER, Prashant Singh, Amit Kumar Sharma, Payal Sood, Paramdeep Singh, IJEEE, Vol. 2, Issue 2 (April, 2015).

[8] Smart Dog Feeder Design Using Wireless Communication, MQTT and Android Client, Vania, Kanisius Karyono, Hargyo Tri Nugroho I. , International Conference on Computer, Control, Informatics and its Applications 2016.