

Smart Shopping System

Submitted in partial fulfilment of the requirements

of the degree of

Bachelors of Engineering

By

1. Bhargav Deshpande - 12
2. Jugal Kahlon - 24
3. Arnab Choudhury - 10
4. Yash Dubey - 15

Supervisor(s):

Prof. V.M Joshi



Department of Electronics Engineering

V.E.S. Institute of Technology

2019-20

CERTIFICATE

This is to certify that the project entitled “**Smart Shopping System**” is a bonafide work of “**Bhargav Deshpande (12); Jugal Kahlon (24); Arnab Choudhury (10); Yash Dubey (15)**” submitted to the V.E.S. Institute of Technology in partial fulfilment of the requirement for the award of the Bachelor of Engineering in **Electronics**.

(Name and sign)

Supervisor/Guide

(Name and sign)

Co-Supervisor/Guide

(Name and sign)

Head of Department

(Name and sign)

Principal

Project Report Approval for B. E.

This project report entitled **Smart Shopping System** by **Bhargav Deshpande; Yash Dubey; Jugal Kahlon; Arnab Choudhury** is approved for the degree of Bachelor of Electronics Engineering.

Examiners:

1. _____

2. _____

Declaration

I/We declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I/We have adequately cited and referenced the original sources. I/We also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I/We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Signature:

Name of student and Roll No.

INDEX

Sr No.	Topic	Page No.
1	Introduction	6
2	Literature Review	8
3	Appendix	10
4	Block Diagram	36
5	Results and Future Prospects	40
6	Discussion	50
7	Conclusion	51
8	References	52

ABSTRACT

Today's world consists of a fast growing and demanding population which expects absolute accuracy and complete usage of the time it has. Customers who need to purchase products in supermarkets need a lot of time and patience in coordinating among themselves for a successful shopping experience. We need to address this problem by efficiently using technologies that are already available or developing new ones. In the race of advancing technologies, the world is getting automated at a tremendous rate and in many aspects. In this Paper, we render a reasonable and cost-effective Smart Shopping Cart utilizing IoT (Internet of Things) innovations. Such a framework is pertinent for use in supermarkets, where escaping the long billing queue is almost impossible and in making the shopping experience for the clients peaceful. Rather than compelling the customers to sit tight in a long queue looking at their shopped products, this architecture helps in alleviating the exacting and excruciating billing process. Our proposed system provides the user a hassle free shopping experience while saving a huge amount of his time. Also, we have developed a Barcode Scanner that helps a customer scan his selected products himself without the presence of a Barcode Scanning Module. The Barcode Scanner that we have developed is integrated to our self designed application, using which the user will be able to not only scan his selected products but also pay for them using Paytm. These features save a huge amount of time and make shopping enjoyable. Along with these abilities, we have done our absolute best in making this system full proof. The Smart Kart which we have developed will keep a track of the quantity and the weight of all the items being added to it, thereby providing the shopkeeper a stressless affair. In addition, the Supermarket management will be able to analyze the shopping behaviors of various customers and will result in valuable business insights. These will be very beneficial for these retail stores and in turn promoting the fact that automation results in efficiency. Accordingly, the management team will have the ability to predict the rate of sales of all individual products and make the stock available based on the ongoing customer requirements. Overall, this system will ensure that the customers will have the best shopping experience and save up a huge amount of their valuable time.

Chapter No. 1

Introduction

1.1 General Introduction:

People tend to overshoot their budget when they are shopping at a big shopping center. Moreover they end up in long queues at the end of their shopping waiting for the products to be scanned and billed. The “Smart Shopping Kart” addresses the above problems with ease. It helps the customer in ensuring that the long queues at the billing counter are eliminated as the products are already scanned and the customer just has to pay the bill and bag the items purchased. The system is profitable for the shopping centers as it can help in reducing the number of billing counters and in turn will help in reducing employee costs significantly. We designed and developed a “Smart Kart” which aims to reduce and possibly eliminate the total waiting time of customers, lower the total manpower requirement and increase efficiency overall and to get rid of this, we have proposed “Smart Trolley” using a Barcode scanner.

The traditional shopping carts which are available in a shopping market are nothing but carts with a steel frame moving on wheels. Till now there has been no incorporation of electronics in order to aid the customers and enhance their shopping experience. Though there have been a lot of attempts to modernize the shopping carts all of these attempts are aimed at finding the products in the shopping market in lesser time using web servers and other utilities. The most common problems faced by the customers are wasting time in the queues for billing rather than not being able to find the product of their choice. Hence there is a need to address the most common problems before approaching the more complex ones.

We are designing a smart kart for a smart shopping system where we are using basic cart which is transparent and in which hardware used are ESP8266 Node MCU, UltraSonic Sensors HC-SR04, Load Cells and Camera to scan Barcode (Phone Camera) and the software part used are MIT App Inventor, Google Sheets as the Database. With the increasing reliability and cost effectiveness of Internet of Things (IoT) based connected smart products in the field of consumer applications, it makes better sense to ensure such technologies are put to use in addressing the day-to-day concerns of the common man. In this system, it is possible for the customer to interface the application with the main server database and it will have a facility to generate the bill for all products added into the cart. The proposed system will be helpful for avoiding queues in shopping malls for billing. It is also helpful to reduce total manpower and also customers aware about total bills during shopping. Hence the shopping becomes easy, enjoyable and hassle free.

1.2 Objective:

We aim to outline a framework which alleviates the standardized shopping experience which usually results in a tiring process of selecting the items and waiting in long and tiring queues in order to just pay the bill. Weight sensor and a Load Cell will be utilized to distinguish the nearness of new items added to the shopping cart. The standardized identification scanner separates the standardized tag which is transmitted to the microcontroller through an USB association. The microcontroller peruses data from a SD card embedded into the microcontroller. This SD card has all the data about the item. This information is at that point organized and exhibited to the client for survey and affirmation on a LCD screen. This is the traditional approach of a Smart Shopping System and results in increased costs in both the retailer's and the consumer's end. We have developed a newer approach which will enable a user and a retailer to save a huge amount of time and also a significant monetary benefit on the retailer's side. New items in the shopping cart will be recognized by following the adjustment in the yield of weight sensors. Similar sensors will be utilized to recognize when things are expelled from the shopping cart. A program will be executed to affirm the expulsion from the customer's shopping basket. Another program will be executed to function as an against burglary system to keep the customer from leaving without an effective installment. The carts inbuilt programmed charging framework makes shopping a breeze and has other positive turn offs, for example, liberating staff from tedious checkout filtering, lessening an aggregate number of staffs required and expanding operational effectiveness of the framework.

Benefits

- Improve the shopping experience for all the customers of the store
- Increase efficiency of the exit process
- Eliminates a long waiting queues at the exit counter

Chapter No. 2

Literature Review

The purpose of this study was to review the literature on the current challenges faced by the consumer while shopping in a mall or shop where the queues are the problem for them to stand for the billing process and to resolve this problem there is a technology driven smart shopping system. In a smart shopping system, there is a smart kart used for this process in which it is IoT associated in which scanning is done through the barcode scanner and is saved in the database. The technique of barcode scanning helps in retrieval of prices without actually increasing the total cost of the product. Related work presents the different techniques proposed by the researchers related to IoT based shopping malls. There are also various literature reviews related to the smart trolley which have been taken into consideration.

1. Smart Trolley System for Automated Billing using RFID and IoT: An automated smart shopping system is formed by introducing the concept of IoT to connect all items in the grocery shop. In this system, every product is embedded with an inexpensive RFID. When the product is placed into a smart cart, the product detail is automatically read by the cart equipped with an RFID reader. Hence, billing is carried out from the shopping cart itself preventing customers from waiting in a queue at checkout. Also, expiry date of the product is displayed and the damaged products can be identified with respect to its weight. Secondly, smart shelving is added to this system by introducing RFID readers that can monitor stock, perhaps updating a central server. Thus, making inventory management easier. Finally, the purchases are validated at checkout points.

2. IoT Based Smart Automatic Shopping Cart with Overload Indicator: In such an IOT framework, an Inexpensive RFID tag can be connected to every item which, When set into a savvy shopping basket, can be consequently Read by a truck outfitted with RFID per user. Furthermore, a smarter method of stocking has been implemented into this architecture, included with RFID per user, and can observe and maybe additionally refresh a focal server. Another advantage of this sort of framework is that Inventory management turns out to be significantly less exacting.

3. Smart Shopping Cart for Automated Billing Purpose using Wireless Sensor Networks: With the increasing employment of broad area Wireless Sensor Networks (WSN) in the field of consumer applications, it becomes imperative to address the concerns raised by its application, such as reliability, energy consumption and cost-effectiveness. In this paper, the author describes the implementation of a reliable, fair and cost efficient Smart Shopping Cart using Wireless Sensor Networks. Such a system is suitable for use in places such as supermarkets, where it can help in reducing man power and in creating a better shopping experience for its customers. Instead of making the customers wait in a long queue for

checking-out their shopped items, the system helps in automating the billing process. Along with this ability, the system design also ensures detection of cases of deception invoked by dishonest customers, which makes the smart system fair and attractive to both the buyers and sellers. The system design along with the experimental setup are presented. The results are encouraging and with the use of repeaters at appropriate locations inside the supermarkets, our approach illustrates itself to be conceivable for use outside the laboratory, in real world deployment.

4. Intelligent shopping cart with quick payment based on dynamic target tracking: At present, many supermarkets are still using the traditional shopping cart and bar code scanning. This shopping mode will make clients feel tired to push cart forward and waste much time waiting in line. This paper presents the design and implementation of a new application - an intelligent shopping cart. This cart can accurately follow its master by identifying the color and shape characteristics of the specific image that the master wears. Also, it implements quick payment to provide a more comfortable shopping mode. In this paper, the author firstly designs the architecture of an intelligent shopping cart system. Secondly, he proposes an approach to implement movement control. The author designs an algorithm to realize the dynamic target tracking more accurately and present improved control systems to improve the stability of the shopping cart system. Thirdly, he presents RFID based automatic payment. Finally, implementation of the proposed novel system. The experimental results indicate that the proposed intelligent cart system provides dynamic target tracking with higher accuracy and significantly speeds up the payment process.

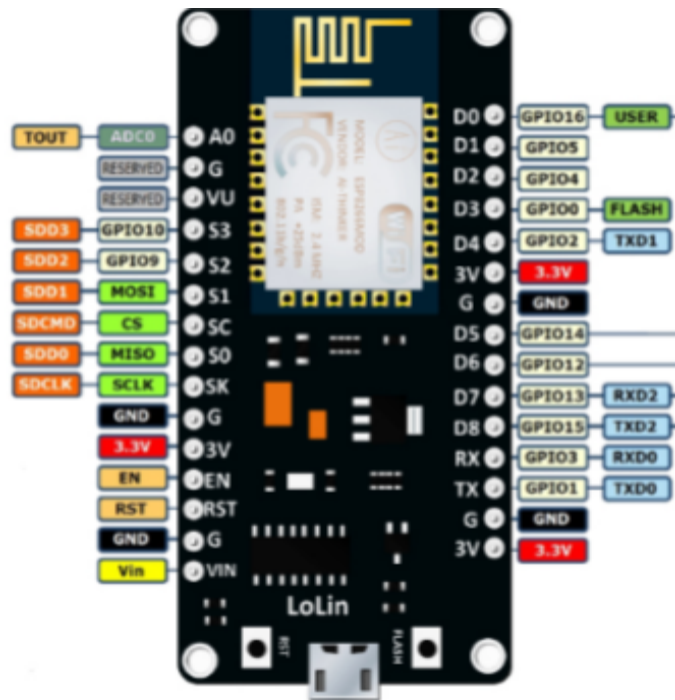
5. Smart Trolley with Advanced Billing System: The paper introduces a smart trolley which is of great use for shopping at the supermarkets remedying the difficulties like waiting in large queues at the payment counter, dragging the trolley throughout the shopping time and calculating the total cost of the items taken, hence providing the quickest shopping experience. Once the trolley is taken along for shopping, it follows the customer throughout the shopping period leaving a certain distance, turns left/right or even ceases along with the customer, signals him/her if any obstacle comes in its way and sums up the total cost price as soon as stuff is dropped into it. At the end of shopping, the customer can swipe the shopping card in the built-in billing system of the trolley for payment which will be accounted at the main counter.

Chapter No. 3

Appendix

3.1 Components Procured:

1. NodeMCU



NodeMCU is an open source IoT platform. It includes firmware which runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is predicated on the ESP-12 module. The term "NodeMCU" by default refers to the firmware instead of the event kits. The firmware uses the Lua scripting language. Due to resource constraints, users got to select the modules relevant for his or her project and build a firmware tailored to their needs. The support for 32-bit ESP32 has also been implemented. The prototyping hardware used can be a circuit card that can function as a dual in-line package (DIP) which integrates a USB controller with a smaller surface-mounted board containing the MCU and antenna. The choice of the DIP format allows for easy prototyping on breadboards. The pin diagram of NodeMCU can be seen above.

A) ESP-12E Module - The ESP-12E module present in the development board which also contains the ESP8266 chip with Tensilica Xtensa® 32-bit LX106 RISC microprocessor which operates at 80 to 160 MHz adjustable clock frequency and supports RTOS. There's also a support of 128 KB RAM and 4MB of non-volatile storage (for program and data storage) only enough to deal with the massive strings that help in making up sites, JSON/XML data, and everything that we throw in these IoT devices nowadays. The ESP8266 chip integrates

802.11b/g/n HT40 Wi-Fi transceiver, therefore it can't only hook up with a WiFi network and interact with the web, but it also can found out its own network, allowing all the opposite devices to attach on to it. This makes the ESP8266 NodeMCU even more versatile.

B) Power Requirement -As the operating voltage range of ESP8266 is 3V to 3.6V, the board comes with a LDO transformer to keep the voltage steady at 3.3V. It can reliably supply up to 600mA, which should be quite enough when ESP8266 pulls the maximum amount as 80mA during RF transmissions. The output of the regulator is additionally broken across one of the edges of the board and labeled as 3V3. This pin supplies power to external components. The ESP8266 NodeMCU is powered via the on-board MicroB USB connector. Alternatively, if you've got a regulated 5V voltage source, then the VIN pin directly supplies the ESP8266 and its peripherals.

C) Peripherals and I/O - The ESP8266 NodeMCU has a total 17 GPIO pins broken out to the pin headers on both sides of the development board.

D) On-board Switches & LED Indicator - The ESP8266 NodeMCU features two buttons. The first one marked as RST that is located on the top left corner is the Reset button which is used to reset the ESP8266 chip. The other FLASH button on the bottom left corner is the download button used while upgrading firmware. The board also has a LED indicator which is user programmable and is connected to the D0 pin of the board.

E) Serial Communication - The board comes with a CP2102 USB-to-UART Bridge Controller from Silicon Labs, which is used for converting USB signal to serial signal and thus allows your computer to program and communicate with the ESP8266 chip.

PINS :

Power Pins - There are four power pins viz. one VIN pin & three 3.3V pins. The VIN pin is used to directly supply the ESP8266 and its peripherals, if you have a regulated 5V voltage source. The 3.3V pins can be used as the output of an on-board voltage regulator. External components can be easily powered using these pins.

GND is the ground pin of ESP8266 NodeMCU development board.

I2C Pins are used to connect all sorts of I2C sensors and peripherals in your project. Support for both I2C Master and I2C Slave is available. I2C interface functionality can be realized programmatically with a maximum clock frequency of 100 kHz. It should be noted that I2C clock frequency should be greater than the slowest clock frequency of the slave device.

GPIO Pins - ESP8266 NodeMCU consists of 17 GPIO pins that can be assigned to different functions like I2C, I2S, UART, PWM, IR Remote Control, LED Light and Button programmatically. Each digitally enabled GPIO can be configured to internal pull-up or pull-down, or also can be set to high impedance. When configured as an input, this GPIO pin can also be set to edge-trigger or level-trigger to generate CPU interrupts.

ADC Channel - A 10-bit precision SAR ADC is embedded on the NodeMCU. The two functions that can be effectively implemented using ADC viz. Testing power supply voltage

of the VDD3P3 pin and testing input voltage of the TOUT pin. However, these functions cannot be implemented at the same time.

UART Pins - ESP8266 NodeMCU has 2 UART interfaces, i.e. UART0 and UART1, which provide asynchronous communication (RS232 and RS485), and can communicate at up to 4.5 Mbps. UART0 (TXD0, RXD0, RST0 & CTS0 pins) can be used for communication. It supports flow control. However, UART1 (TXD1 pin) features only data transmit signals, so it is usually used for printing logs.

SPI Pins - ESP8266 features two SPIs (SPI and HSPI) in slave and master modes.

SDIO Pins - ESP8266 features Secure Digital Input/Output Interface (SDIO) which is used to directly interface SD cards. 4-bit 25 MHz SDIO v1.1 and 4-bit 50 MHz SDIO v2.0 are supported.

PWM Pins - The board has 4 channels of Pulse Width Modulation (PWM). The PWM output can be implemented programmatically and used for driving digital motors and LEDs. PWM frequency range is adjustable from 1000 μ s to 10000 μ s, i.e., between 100 Hz and 1 kHz.

Control Pins are used to control ESP8266. These pins include Chip Enable pin (EN), Reset pin (RST) and WAKE pin.

NodeMCU ESP8266 Specifications & Features

- Microcontroller: Tensilica 32-bit RISC CPU Xtensa LX106
- Operating Voltage: 3.3V
- Input Voltage: 7-12V
- Digital I/O Pins (DIO): 16
- Analog Input Pins (ADC): 1
- UARTs: 1
- SPIs: 1
- I2Cs: 1
- Flash Memory: 4 MB
- SRAM: 64 KB
- Clock Speed: 80 MHz
- USB-TTL based on CP2102 is included onboard, Enabling Plug n Play
- PCB Antenna
- Small Sized module to fit smartly inside your IoT projects

2. Load Cell



A load cell may be a sort of transducer, specifically a force transducer. It converts a force like tension, compression, pressure, or torque into an electrical signal which will be measured and standardized. The electrical signal changes proportionally as the force applied to the load cell increases. The most common sorts of load cell used are hydraulic, pneumatic, and strain gauge .

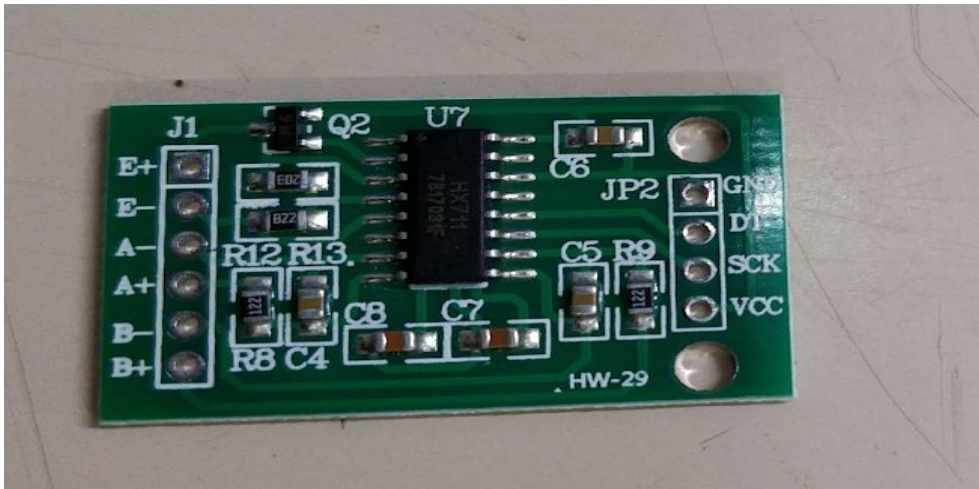
Strain gauge load cells are the type most frequently found in industrial settings. It is ideal because it is very accurate, versatile, and cost-effective. Structurally, a load cell features a metal body to which strain gauges are secured. The body is usually made of aluminium, alloy steel, or stainless steel which makes it very sturdy but also minimally elastic. This elasticity gives rise to the term "spring element", pertaining to the body of the load cell. When this force is exerted on the load cell, its spring element is slightly deformed, and unless overloaded, always returns to its original shape. As the spring element deforms, change in shape is observed in the strain gauge. The resulting alteration to the resistance in the strain gauges can be measured as voltage. The change in voltage is proportional to the amount of force applied to the cell, thus the amount of force can be calculated from the load cell's output.

Strain Gauges

A strain gauge is made of very fine wire, or foil, found out during a grid pattern and attached to a versatile backing. When the form of the strain gauge is altered, a change in its electric resistance occurs. The wire or foil within the strain gauge is arranged such that, when force is applied in one direction, a linear change in resistance results. Tension force stretches a strain gauge , causing it to become thinner and longer, leading to a rise in resistance. Compression force does the opposite. The strain gauge compresses, becomes thicker and shorter, and

resistance decreases. The strain gage is attached to a versatile backing enabling it to be easily attached to a load cell, mirroring the minute changes to be measured. Since the change in resistance measured by one strain gage is extremely small, it's difficult to accurately measure changes. Increasing the amount of strain gauges applied collectively magnifies these small changes into something more measurable. A set of 4 strain gauges set in a specific circuit is called Wheatstone bridge.

3. HX711 Weighing Sensor



The HX711 Weighing Sensor works along with Strain Gauge Load Cell to determine the final weight of all our items. This chip is designed for high-precision electronic scale and design, that consists of two analog input channels, and also programmable gain of 128 integrated amplifier. A bridge voltage electrical bridge (such as pressure, load) sensor model can be provided by the configuration of the input circuit, that is an ideal high-precision, low-cost sampling front-end module.

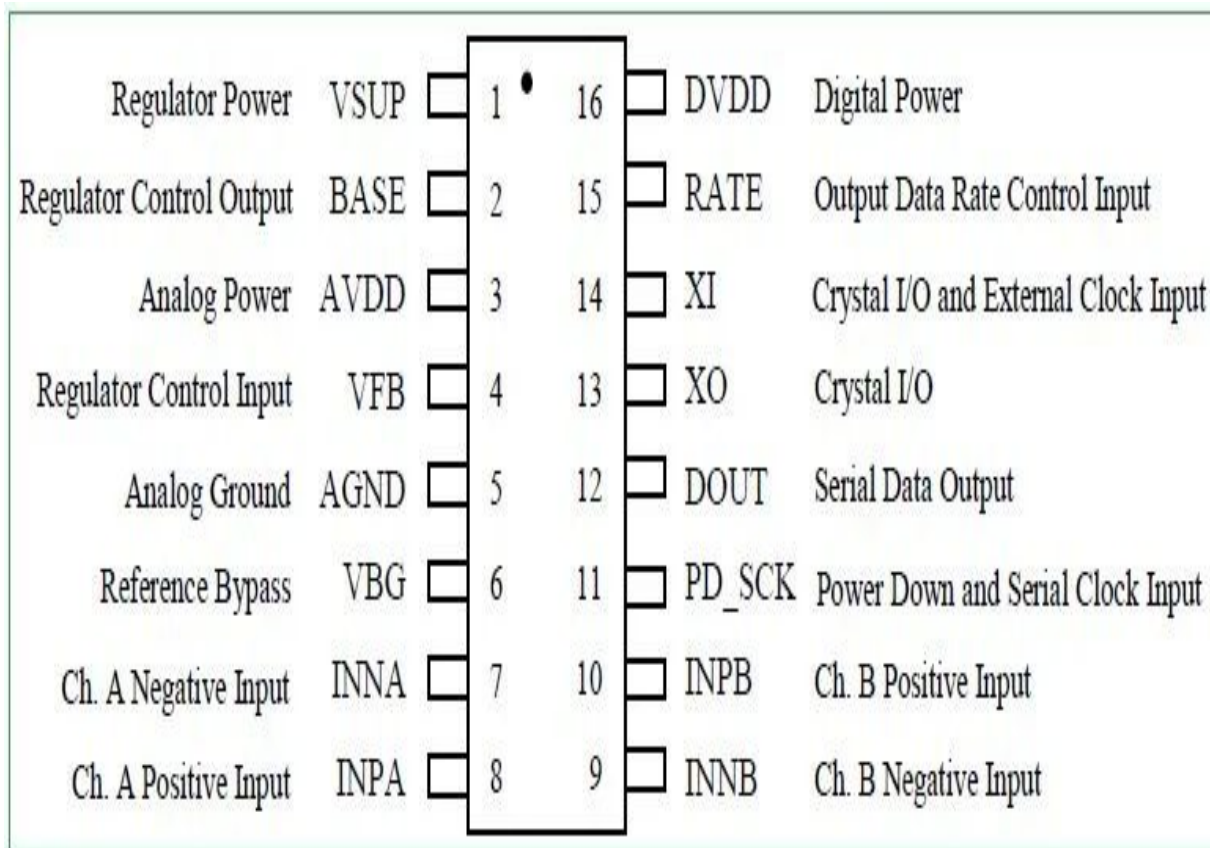
HX711 is a precision 24-bit digitizer (ADC) based on Avia Semiconductors' patented technology, designed for weight scales and industrial control applications to interface directly with a bridge sensor. The input multiplexer controls the selection of either Channel A or B differential input to the low-noise programmable gain amplifier (PGA). Channel A is often programmed with a gain of 128 or 64, like a full-scale differential input voltage of $\pm 20\text{mV}$ or $\pm 40\text{mV}$ respectively, when a 5V supply is connected to an AVDD analog power supply pin. Channel B has a fixed gain of 32. On-chip power supply regulators eliminate the necessity for an external supply regulator to supply analog power for the ADC and therefore the sensor. Clock input is flexible. It is often from an external clock source, a crystal, or the on-chip oscillator that doesn't require any external component. On-chip power on-reset circuitry simplifies digital interface initialization. The internal registers need no programming. All the pins have the controls to the HX711.

Features :

- 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 • Simple digital control and serial interface: pin-driven controls, no programming needed.
- 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°C • 16 pin SOP-16 package.

The pin diagram of the same is given below -



Pin Description -

Pin #	Name	Function	Description
1	VSUP	Power	Regulator supply: 2.7 ~ 5.5V
2	BASE	Analog Output	Regulator control output (NC when not used)
3	AVDD	Power	Analog supply: 2.6 ~ 5.5V
4	VFB	Analog Input	Regulator control input (connect to AGND when not used)
5	AGND	Ground	Analog Ground
6	VBG	Analog Output	Reference bypass output
7	INA-	Analog Input	Channel A negative input
8	INA+	Analog Input	Channel A positive input
9	INB-	Analog Input	Channel B negative input
10	INB+	Analog Input	Channel B positive input
11	PD_SCK	Digital Input	Power down control (high active) and serial clock input
12	DOUT	Digital Output	Serial data output
13	XO	Digital I/O	Crystal I/O (NC when not used)
14	XI	Digital Input	Crystal I/O or external clock input, 0: use on-chip oscillator
15	RATE	Digital Input	Output data rate control, 0: 10Hz; 1: 80Hz
16	DVDD	Power	Digital supply: 2.6 ~ 5.5V

4. Servo Motor



A servo motor is a device which may push or rotate an object with great precision. If you want to rotate an object at some specific angles or distance, then you use a servo motor. It is just made from an easy motor which runs through a servo mechanism. If a motor is DC powered then it is called DC servo motor, and if it is AC powered motor then it is called AC servo motor. We can get a very high torque servo motor in small and lightweight packages. Due to these features they're getting used in many applications like toy cars, RC helicopters and planes, Robotics, Machine etc. Servo motors are rated in kg/cm (kilogram per centimeter) most hobby servo motors are being rated at 3kg/cm or 6kg/cm or 12kg/cm. This kg/cm tells you how much weight your servo motor can lift at a specific distance. For example: A 6kg/cm Servo motor should be ready to lift 6kg if the load is suspended 1cm faraway from the motor's shaft, the greater the space the lesser the weight carrying capacity. The position of a servo motor is set by electrical pulse and its circuitry is placed beside the motor.

Servo Mechanism: It consists of three parts;

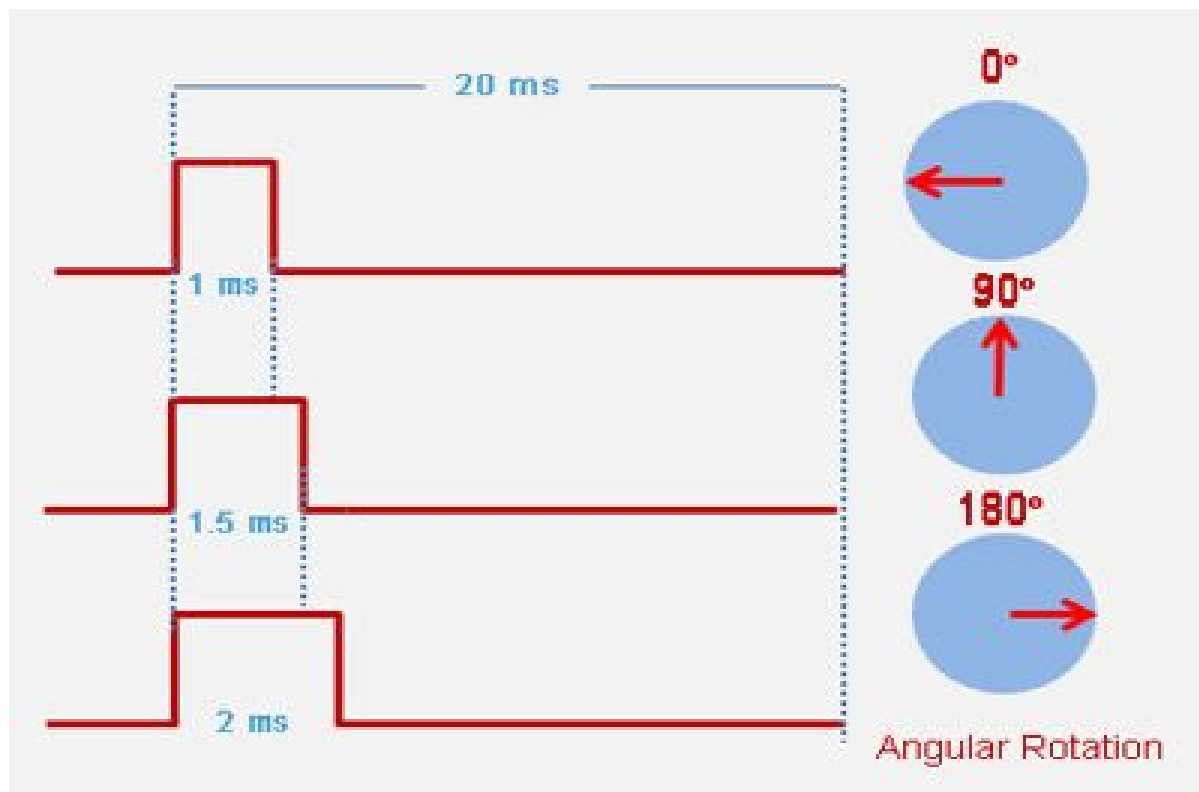
1. Controlled device
2. Output sensor
3. Feedback system

It is a closed loop system where it uses a positive feedback system to control motion and final position of the shaft. Here the device is controlled by a feedback signal generated by comparing output and reference input. Here the reference input signal is compared to the

reference output signal and the third signal is produced by the feedback system. And this third signal acts as input signal to control devices. This signal is present as long as a feedback signal is generated or there is a difference between reference input signal and reference output signal. So the main task of servomechanism is to take care of output of a system at desired value at presence of noises.

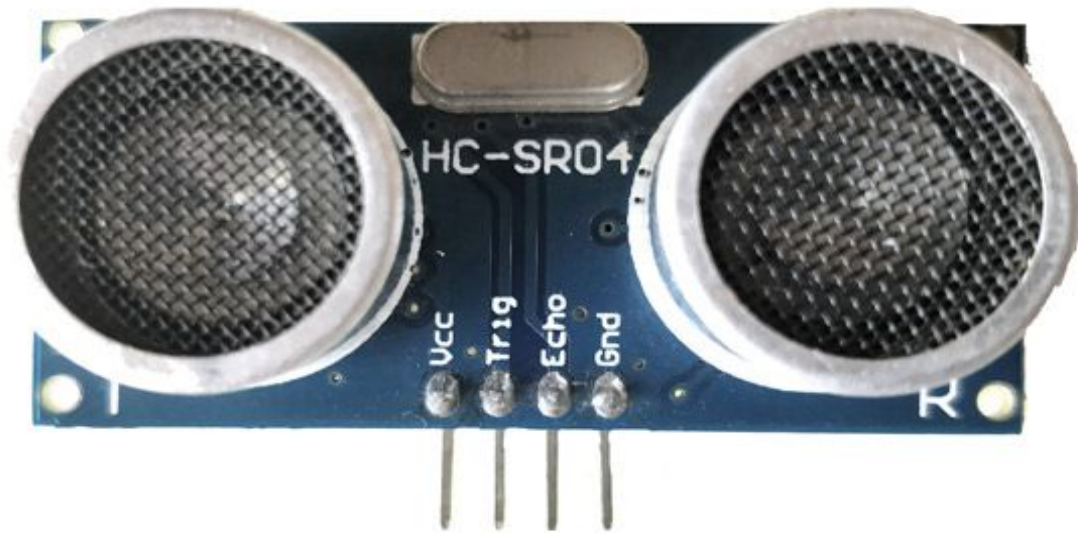
Working principle of Servo Motors: Working principle of Servo Motors: Servo consists of a Motor (DC or AC), a potentiometer, gear assembly and a controlling circuit. First of all we use gear assembly to scale back RPM and to extend torque of the motor. In the initial position, the servo motor shaft, the position of the potentiometer knob is such that there's no electrical signal generated at the output port. Now an electrical signal is given to a different input terminal of the error detector amplifier. Now the difference between these two signals, one comes from a potentiometer and another comes from another source, are going to be processed during a feedback mechanism and output are going to be provided in terms of error signal. This error signal acts because the input for motor and motor starts rotating. Now the motor shaft is connected to the potentiometer and as the motor rotates the potentiometer and it will generate a signal. So because the potentiometer's position changes, its output feedback signal changes. After sometime the position of the potentiometer reaches at a position that the output of the potentiometer is the same as the external signal provided. In this condition, there'll be no output from the amplifier to the motor input as there's no difference between external applied signal and then the signal generated at the potentiometer, and in this situation the motor stops rotating.

Controlling Servo Motor: All motors have three wires coming out of them. Out of which two will be used for Supply (positive and negative) and one will be used for the signal that is to be sent from the MCU. Servo motor is controlled by PWM (Pulse width Modulation) which is provided by the control wires. It is a minimum pulse, a maximum pulse and a repetition rate. Servo motor can turn 90 degrees from either direction from its neutral position. The motor expects to ascertain a pulse every 20 milliseconds (ms) and therefore the length of the heart beat will determine how far the motor turns. For example, a 1.5ms pulse will make the motor turn to the 90° position, such as if pulse is shorter than 1.5ms shaft moves to 0° and if it is longer than 1.5ms than it will turn the servo to 180°. Servo motor works on PWM (Pulse width modulation) principle, means its angle of rotation is controlled by the duration of applied pulse to its Control PIN. Basically a servo motor is made up of a DC motor which is controlled by a variable resistor (potentiometer) and some gears. High speed force of the DC motor is converted into torque by Gears. We know that $WORK = FORCE \times DISTANCE$, in DC motor Force is a smaller amount and distance (speed) is high and in Servo, force is High and distance is a smaller amount. Potentiometer is connected to the output shaft of the Servo, to calculate the angle and stop the DC motor on the required angle.



Servo motors can be rotated from 0 to 180 degree, but it can go up to 210 degree, depending on the manufacturing. This degree of rotation are often controlled by applying the Electrical Pulse of proper width, to its Control pin. Servo checks the pulse in every 20 milliseconds. Pulse of 1 ms (1 millisecond) width can rotate servo to 0 degree, 1.5ms can rotate to 90 degree (neutral position) and a couple of ms pulse can rotate it to 180 degree. All servo motors work directly together with your +5V supply rails but we've to take care on the quantity of current the motor would consume, if you're planning to use quite two servo motors a correct servo shield should be designed.

5. HC - SR04 Ultrasonic Ranging Module



HC-SR04 distance sensor is usually used with both microcontroller and microprocessor platforms like Arduino, ARM, PIC, Raspberry Pie etc. The following guide is universally since it has to be followed irrespective of the type of computational device used. Power the Sensor employing a regulated +5V through the Vcc and Ground pins of the sensor. The current consumed by the sensor is a smaller amount than 15mA and hence are often directly powered by the on board 5V pins (If available). The Trigger and therefore the Echo pins are both I/O pins and hence they will be connected to I/O pins of the microcontroller. To start the measurement, the trigger pin has got to be made high for 10uS then turned off. This action will trigger an ultrasonic wave at frequency of 40Hz from the transmitter and therefore the receiver will await the wave to return. Once the wave is returned after it gets reflected by any object the Echo pin goes high for a particular amount of time which will be equal to the time taken for the wave to return back to the sensor. The amount of your time during which the Echo pin stays high is measured by the MCU/MPU because it gives the information of time taken for the wave to return back to the Sensor. Using this information the distance is measured.

Working - As shown above the HC-SR04 Ultrasonic (US) sensor may be a 4 pin module, whose pin names are Vcc, Trigger, Echo and Ground respectively. This sensor may be a very fashionable sensor utilized in many applications where measuring distance or sensing objects are required. The module has two eyes like projects within the front which forms the

Ultrasonic transmitter and Receiver. The sensor works with the straightforward highschool formula that - $\text{Distance} = \text{Speed} \times \text{Time}$. The transmitter transmits an ultrasonic wave and this wave travels in air and then gets objected by any material, it gets reflected back towards the sensor and this reflected wave is seen by the receiver as shown below.



Now, to calculate the distance using the above formulae, we should know the Speed and time. Since we are using the Ultrasonic wave we all know the universal speed of the US wave at room conditions which is 330m/s. The circuitry inbuilt on the module will calculate the time taken for the US wave to come back and turn on the echo pin high for that same particular amount of time, this way we can also know the time taken. Now simply calculate the space employing a microcontroller or microprocessor.

Applications -

- 1.Used to avoid and detect obstacles with robots like biped robot, obstacle avoider robot, path finding robot etc.
- 2.Used to measure the distance within a wide range of 2cm to 400cm.
- 3.Can be used to map the objects surrounding the sensor by rotating it.
- 4.Depth of certain places like wells, pits etc can be measured since the waves can penetrate through water.

3.2 Software Used:

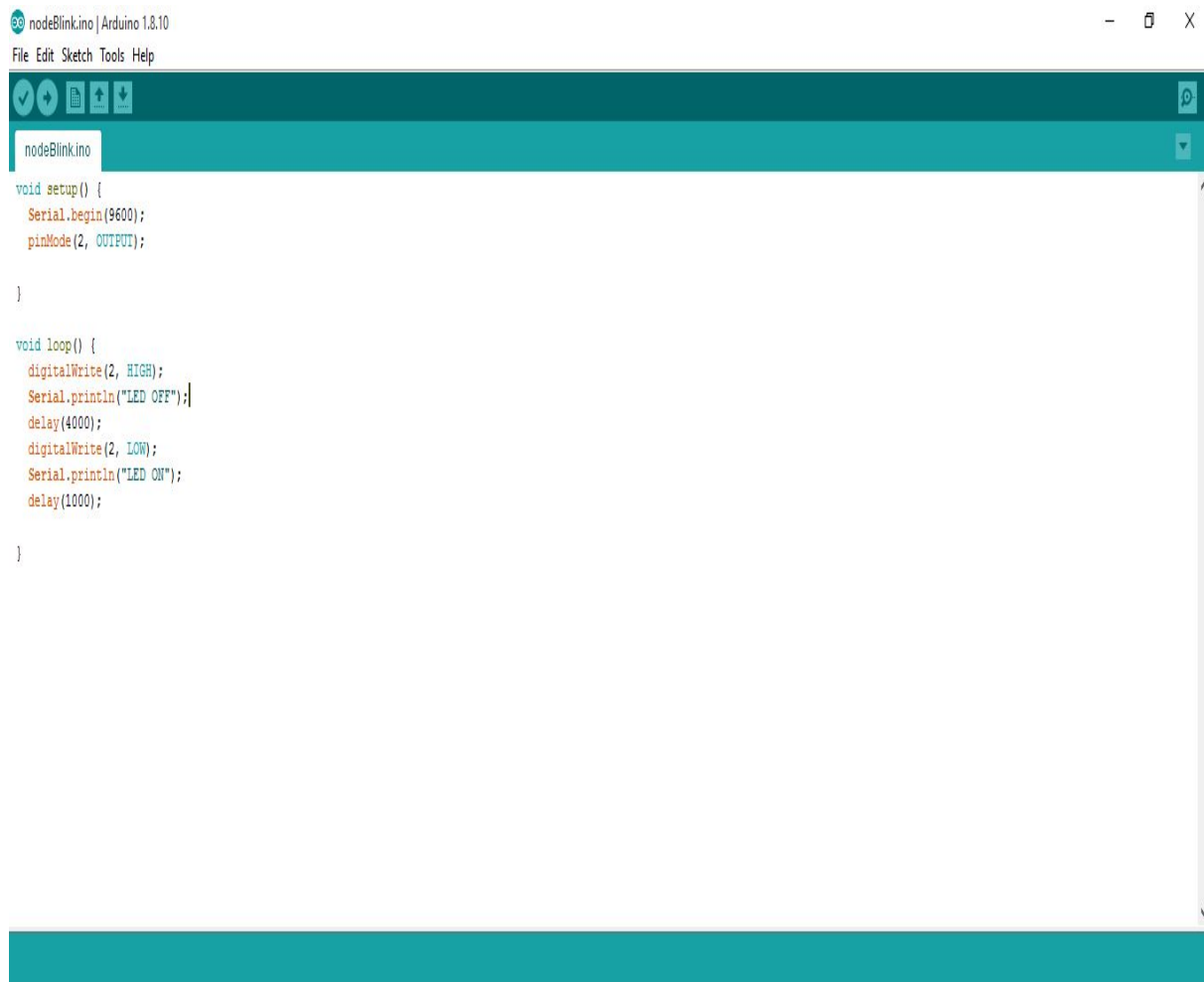
A)Arduino IDE: The Arduino Integrated Development Environment (IDE) is an application supported by multiple platforms (Windows, macOS, Linux) that is written using functions from C and C++ with special rules for code structuring. Using it you can write and upload programs to boards that are compatible with Arduino, but also, with the help of 3rd party cores, other vendor development boards. The source code for the IDE is released under the GNU General Public License, version 2. The Arduino IDE supplies a software library from the Wiring project, which includes many common input and output procedures. The code written by the user only required two basic functions, for starting the sketch and the main program loop, that are linked with a program stub `main()` and compiled into an executable cyclic program with the GNU toolchain, which you will find in the IDE distribution. To convert the code into a text file in hexadecimal encoding, Arduino IDE employs the program `avrdude`, that is loaded with the help of a loader program in the board's firmware. By default, the uploading tool to flash the user code onto official is `avrdude`. The popularity of Arduino is on the rise as a software platform, and now other vendors have started to implement custom open source compilers & tools (cores) that can build and upload sketches to other MCUs that are not supported by Arduino's official line of MCUs.

In order to program the ESP8266 module, we need to connect it to a computer. Since Serial Communication is the only available communication on the ESP8266 ESP-01 module, we need an USB to Serial Adapter like an FTDI, CH340 or FT232RL. If you do not have a dedicated USB to Serial Adapter, do not worry. The Arduino UNO features an on-board USB to Serial Adapter. We are going to use this for programming the ESP8266. We will also be making use of the TX and RX pins of the Arduino to connect to the ESP8266 module and in order to make sure that Arduino isn't using those pins, we can upload a bare minimum sketch to Arduino.

As already seen the required components and the circuit diagram of the project. Now, let us try to understand the design of the circuit. First and foremost, the ESP8266 module works on power supply of 3.3V and anything greater than that, taking 5V as an example, will cause trouble the SoC. So, the VCC Pin and CH_PD Pin of ESP8266 ESP-01 module are connected to a 3.3V Supply. Next important thing to remember is that the ESP8266 WiFi Module has two modes of operation: Programming Mode and Normal Mode. The Programming Mode will enable you to upload the program or firmware to the ESP8266 module and in Normal Mode, the uploaded program or firmware will run normally. When we connect the GPIO0 pin must be connected to GND, the Programming Mode is enabled. As specified below, we have connected the SPDT switch to the GPIO0 pin. Then we have toggled the lever of SPDT which will switch the ESP8266 between Programming mode and normal mode. Also, the RST (Reset) pin plays an important role in enabling Programming Mode. The RST pin is an active LOW pin and hence, it is connected to GND through a Push Button, hence pressing the button

will reset the ESP8266 module. Then we connect the RX and TX pins of ESP8266 Module to the RX and TX Pins on the Arduino board. Since the ESP8266 SoC cannot tolerate 5V, the RX Pin of Arduino is connected through a level converter consisting of a 1K Ω and a 2.2K Ω Resistor. Finally we connect an LED to the GPIO2 pin to test the working of the program. All the necessary connections for enabling the Programming Mode in ESP8266 are specified below.

1. VCC $\rightarrow$ 3.3V
2. GND $\rightarrow$ GND
3. CH_PD $\rightarrow$ 3.3V
4. RST $\rightarrow$ Normally Open; GND to Reset
5. GPIO0 $\rightarrow$ GND
6. TX $\rightarrow$ TX of Arduino
7. RX $\rightarrow$ RX of Arduino (through level converter)



The screenshot shows the Arduino IDE interface with a sketch named 'nodeBlink.ino'. The code is as follows:

```
void setup() {  
  Serial.begin(9600);  
  pinMode(2, OUTPUT);  
}  
  
void loop() {  
  digitalWrite(2, HIGH);  
  Serial.println("LED OFF");  
  delay(4000);  
  digitalWrite(2, LOW);  
  Serial.println("LED ON");  
  delay(1000);  
}
```

B)Google Sheets (Database): Google Sheets is a spreadsheet application which is included as part of a free and web-based software office suite offered by Google within its Google Drive service. The service also includes Google Slides, a presentation software and Google docs, a word processor. Google Sheets is available in the form of a web application, mobile app for Android, iOS, Windows, BlackBerry, and as a desktop application on Google's very own ChromeOS. Google Sheets is compatible with Microsoft Excel file formats. The main feature of Google Sheets is that it also allows users to create and edit files online while collaborating with other users in real-time. Edits are tracked by users with a revision history presenting changes. An editor's position is highlighted with a unique colour, specific to the editor and cursor and a permissions system regulates what users can do. They currently offer features using machine learning, including "Explore", offering answers based on natural language questions in a spreadsheet.

Implementation -

1)Create a Google Spreadsheet

2)Populate your data.

3)Click "share" in the top-left corner. Then click "advanced". Then change "Link Sharing" to "On - Anyone with the link".

4)Paste your spreadsheet's URL & a SQL query into Blockspring's Query Google Spreadsheet API. This returns the results as JSON. You can test it out on Blockspring, or call the API from any language.

At this point you can successfully query your Google Spreadsheet from your code. You can stop here. The next steps are if you want to only expose specific parameters or columns in your API.

5)Open the source code for an existing Google Doc API on Blockspring. Here is a working example Google Doc API you can start with.

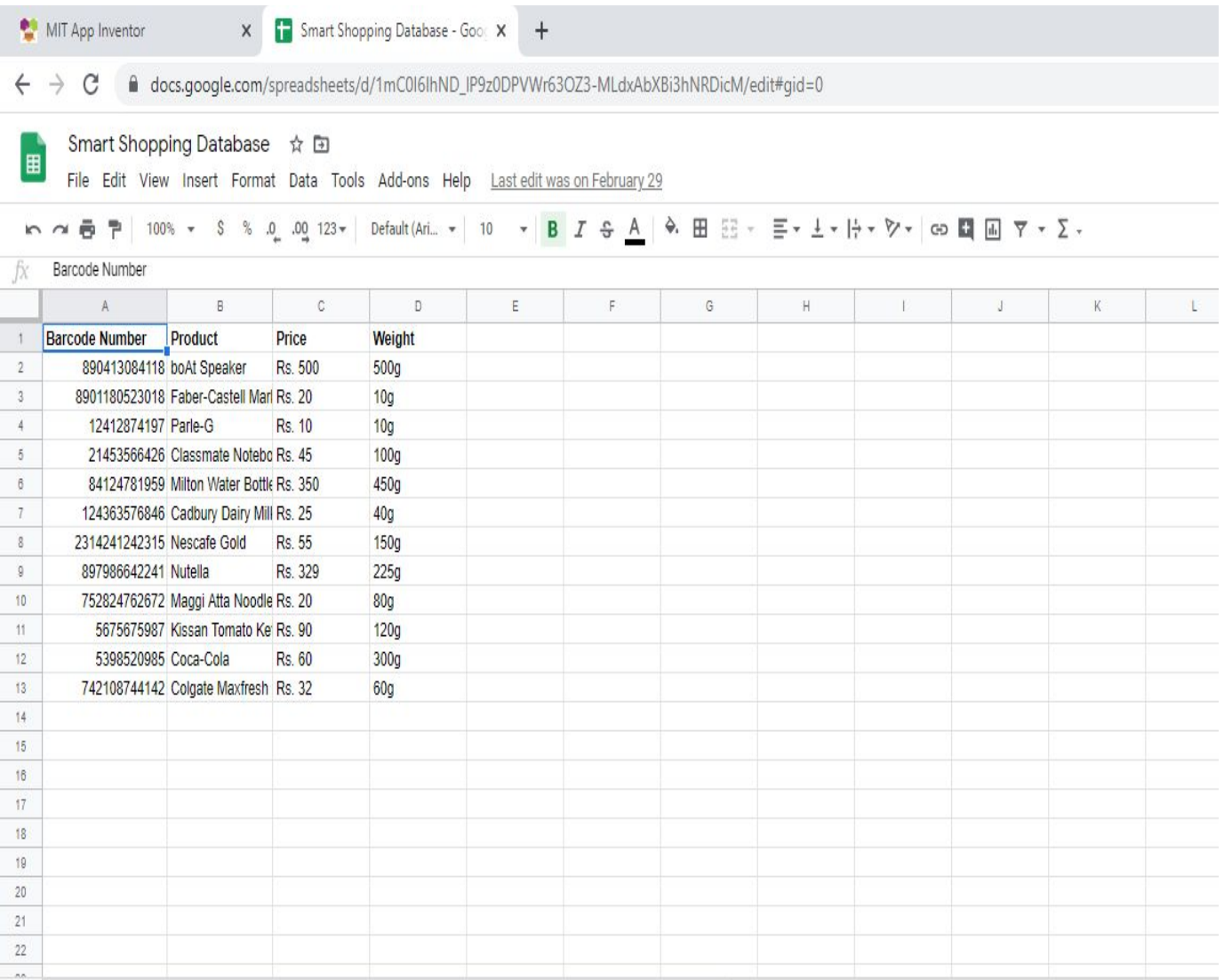
6)On Ln 61, copy and paste your own Google Spreadsheet link.

7)Most of the code is if statements checking if specific URL parameters were supplied, and building up the WHERE clause. Simply replace these if statements with ones that are relevant to your API.

8)Click "Publish"

After Step 4, you will have a fully working API that can be used to get data out of a Google Spreadsheet from Javascript (or any other language). It will take you ~10 minutes, and you will not need to code anything. After Step 8, you will have a fully working API that can also restrict access to data in your Google Spreadsheet and is easily filterable from Javascript. It will take you ~30 minutes, and you will need to code some simple if conditions (in

Javascript). Your API is now live. You can use it in code by clicking the "Run in code" tab on its page.



The screenshot shows a Google Sheet titled "Smart Shopping Database" with the following data:

	A	B	C	D	E	F	G	H	I	J	K	L
1	Barcode Number	Product	Price	Weight								
2	890413084118	boAt Speaker	Rs. 500	500g								
3	8901180523018	Faber-Castell Mar	Rs. 20	10g								
4	12412874197	Parle-G	Rs. 10	10g								
5	21453566426	Classmate Notebo	Rs. 45	100g								
6	84124781959	Milton Water Bottle	Rs. 350	450g								
7	124363576846	Cadbury Dairy Mill	Rs. 25	40g								
8	2314241242315	Nescafe Gold	Rs. 55	150g								
9	897986642241	Nutella	Rs. 329	225g								
10	752824762672	Maggi Atta Noodle	Rs. 20	80g								
11	5675675987	Kissan Tomato Ke	Rs. 90	120g								
12	5398520985	Coca-Cola	Rs. 60	300g								
13	742108744142	Colgate Maxfresh	Rs. 32	60g								
14												
15												
16												
17												
18												
19												
20												
21												
22												
23												

C)MIT App Inventor: MIT App Inventor is a visual programming environment, originally developed by Google and now maintained by the Massachusetts Institute of Technology (MIT). It allows beginners in computer programming to create application software for two operating systems (OS): Android and iOS. It is free and an open-source software, making it widely popular.

It uses a graphical user interface (GUI) very similar to the programming languages Scratch (programming language) and the StarLogo, which allows users to drag and drop visual objects to create an application that can run on mobile devices. The creation of App Inventor, resulted in significant prior research in educational computing, and work done within Google on online development environments.

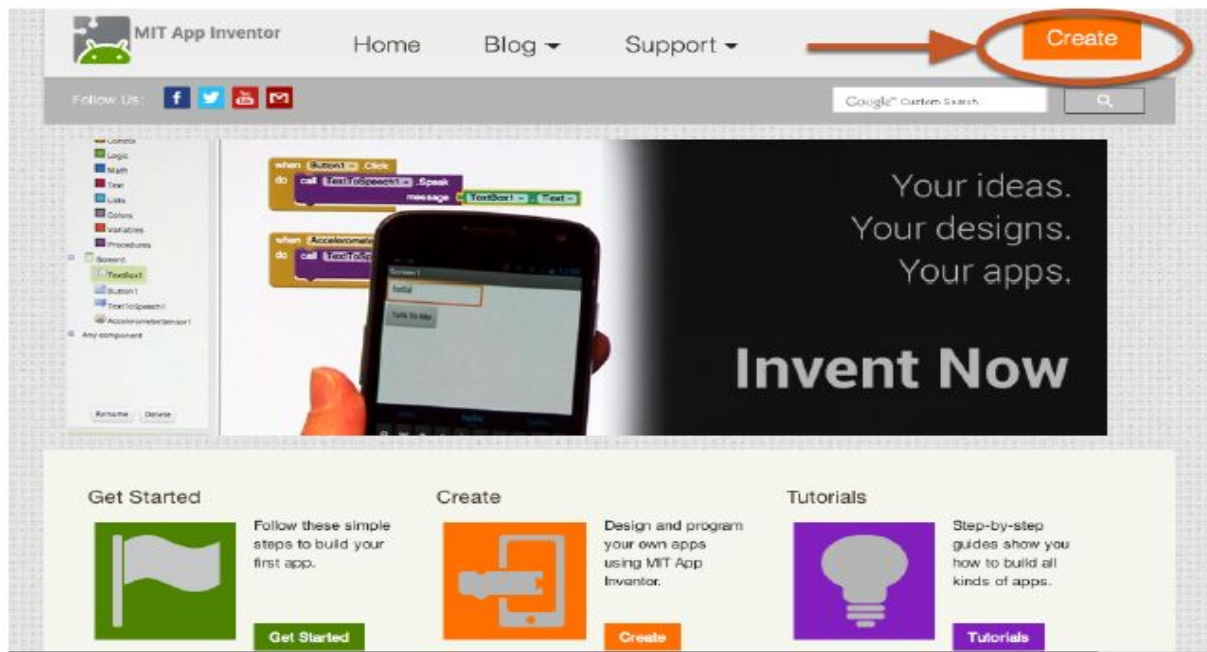
App Inventor is based on and informed by constructionist learning theories, that focus on the fact that programming can act as a vehicle for engaging thought-provoking ideas through active learning. This makes it part of an ongoing movement in computers and education which dates back to the work of Seymour Papert as well as the MIT Logo Group in the 1960s. It has also manifested itself with Mitchel Resnick's work on Lego Mindstorms and StarLogo.App Inventor also supports the use of cloud data via an experimental Firebase, which is a Realtime Database component.

Steps :

1)

To get started, go to App Inventor on the web.

Go directly to ai2.appinventor.mit.edu, or click the orange "Create" button from the App Inventor website.



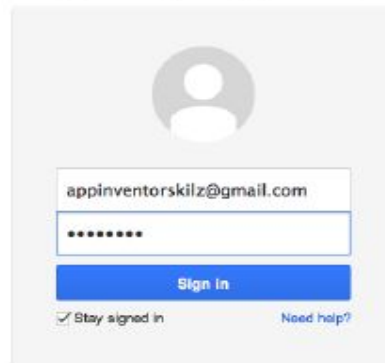
2)

Log in to App Inventor with a gmail (or google) user name and password.

Use an existing gmail account or school-based google account to log in to ai2.appinventor.mit.edu
To set up a brand new gmail account, go to accounts.google.com/SignUp

Google
One account. All of Google.

Sign in with your Google Account

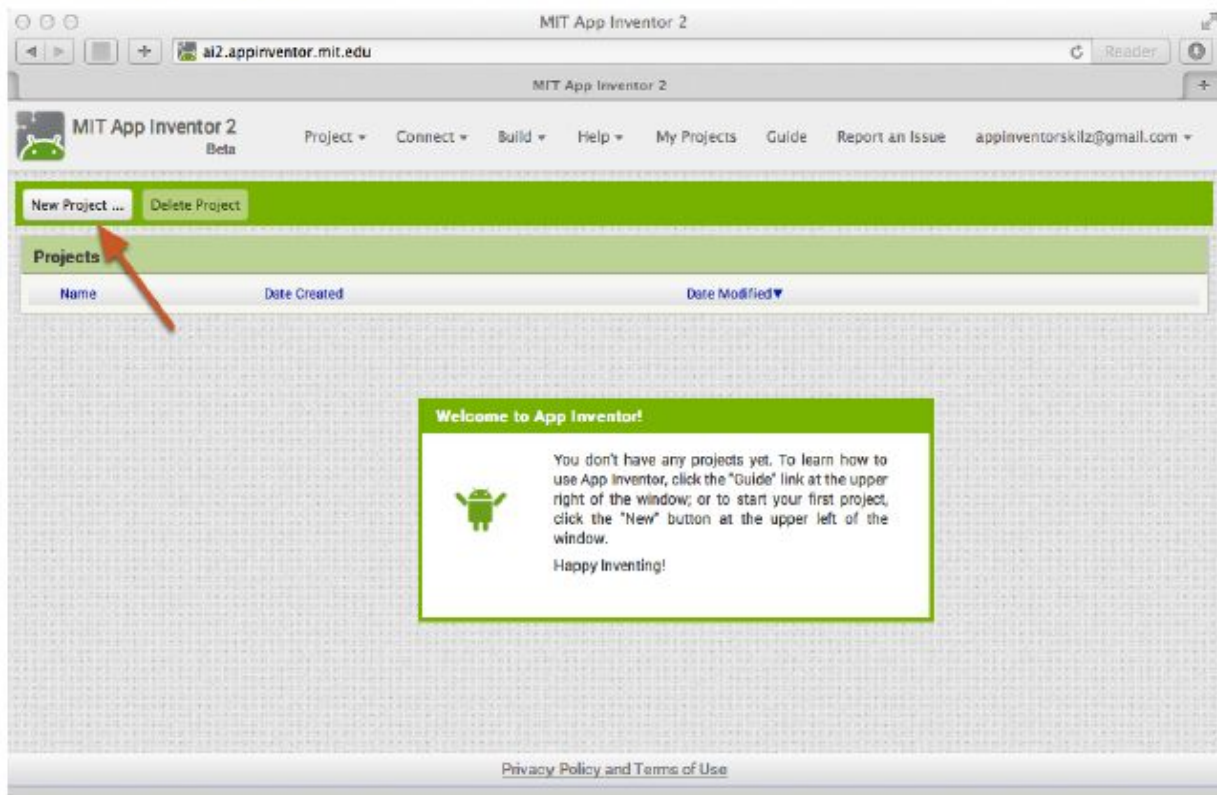


A Google sign-in form with a grey background. At the top is a circular icon with a white person silhouette. Below it is a text input field containing 'appinventorskilz@gmail.com'. Underneath is a password input field with eight dots. A blue 'Sign in' button is below the password field. At the bottom left is a checkbox labeled 'Stay signed in' with a checked mark. At the bottom right is a link 'Need help?'.

[Create an account](#)

3)

Start a new project.



A screenshot of the MIT App Inventor 2 web application. The browser address bar shows 'ai2.appinventor.mit.edu'. The page has a green header with the MIT App Inventor 2 logo and a navigation menu with links: Project, Connect, Build, Help, My Projects, Guide, Report an Issue, and a user dropdown for 'appinventorskilz@gmail.com'. Below the header is a green bar with 'New Project ...' and 'Delete Project' buttons. A red arrow points to the 'New Project ...' button. Below this is a 'Projects' table with columns 'Name', 'Date Created', and 'Date Modified'. The table is currently empty. A large green box with a white border contains a welcome message: 'Welcome to App Inventor!' with an Android robot icon, followed by instructions on how to start a project and a 'Happy inventing!' message. At the bottom of the page is a link for 'Privacy Policy and Terms of Use'.

4)

Name the project "TalkToMe" (no spaces!)

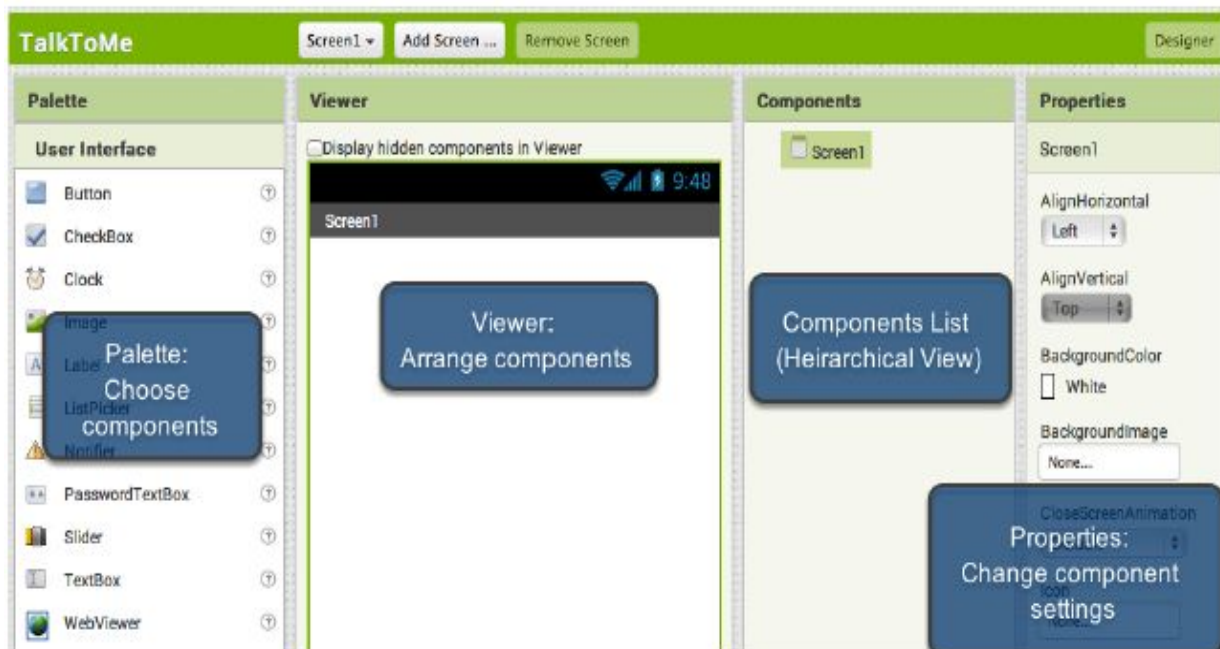
Type in the project name (underscores are allowed, spaces are not) and click OK.



5)

You are now in the Designer, where you lay out the "user interface" of your app.

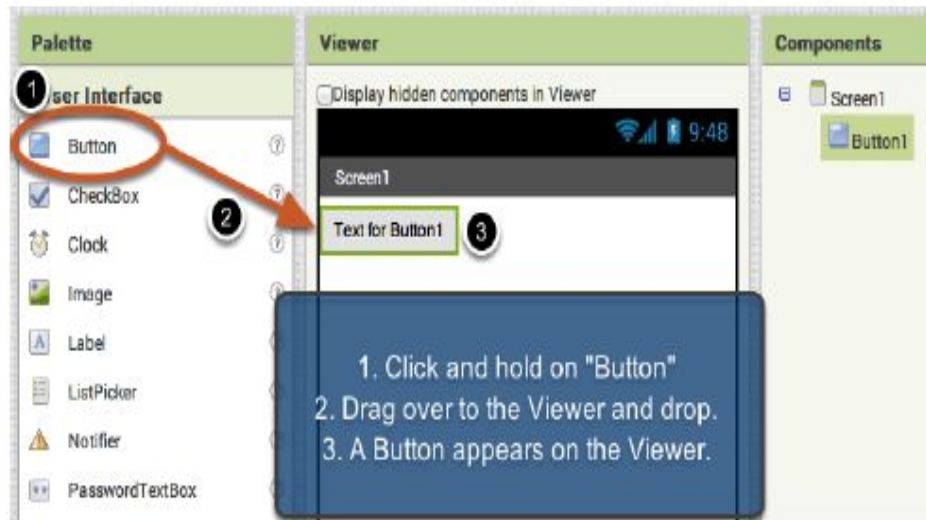
The Design Window, or simply "Designer" is where you lay out the look and feel of your app, and specify what functionalities it should have. You choose things for the user interface things like Buttons, Images, and Text boxes, and functionalities like Text-to-Speech, Sensors, and GPS.



6)

Add a Button

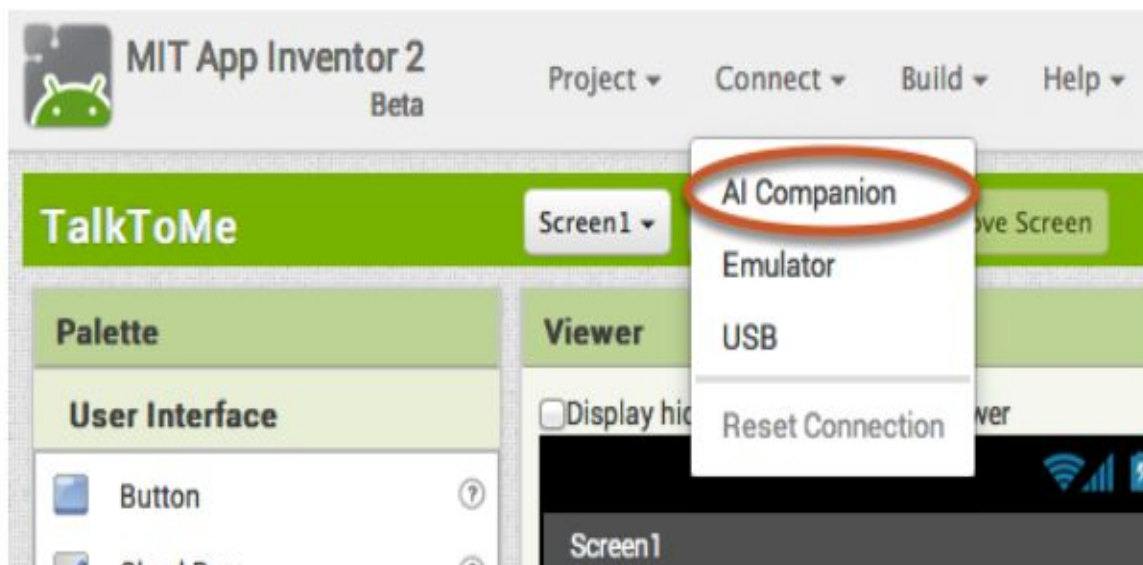
Our project needs a button. **Click and hold** on the word "Button" in the palette. **Drag** your mouse over to the Viewer. **Drop** the button and a new button will appear on the Viewer.



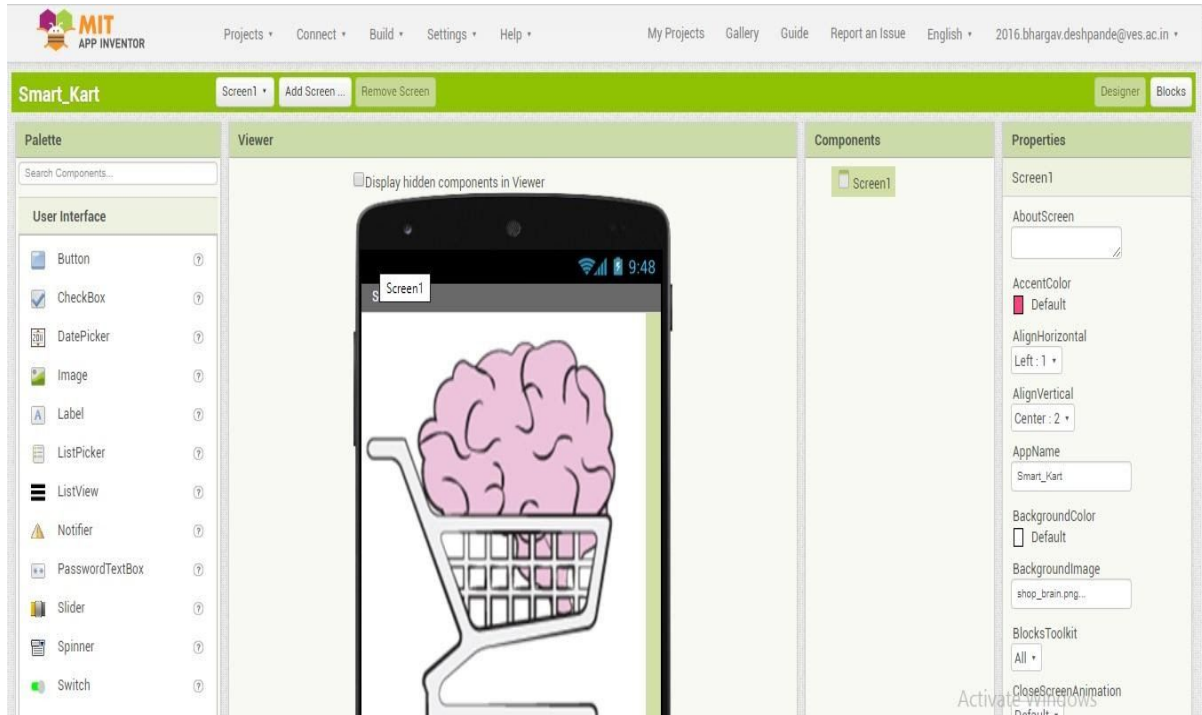
7)

Connect App Inventor to your phone for live testing

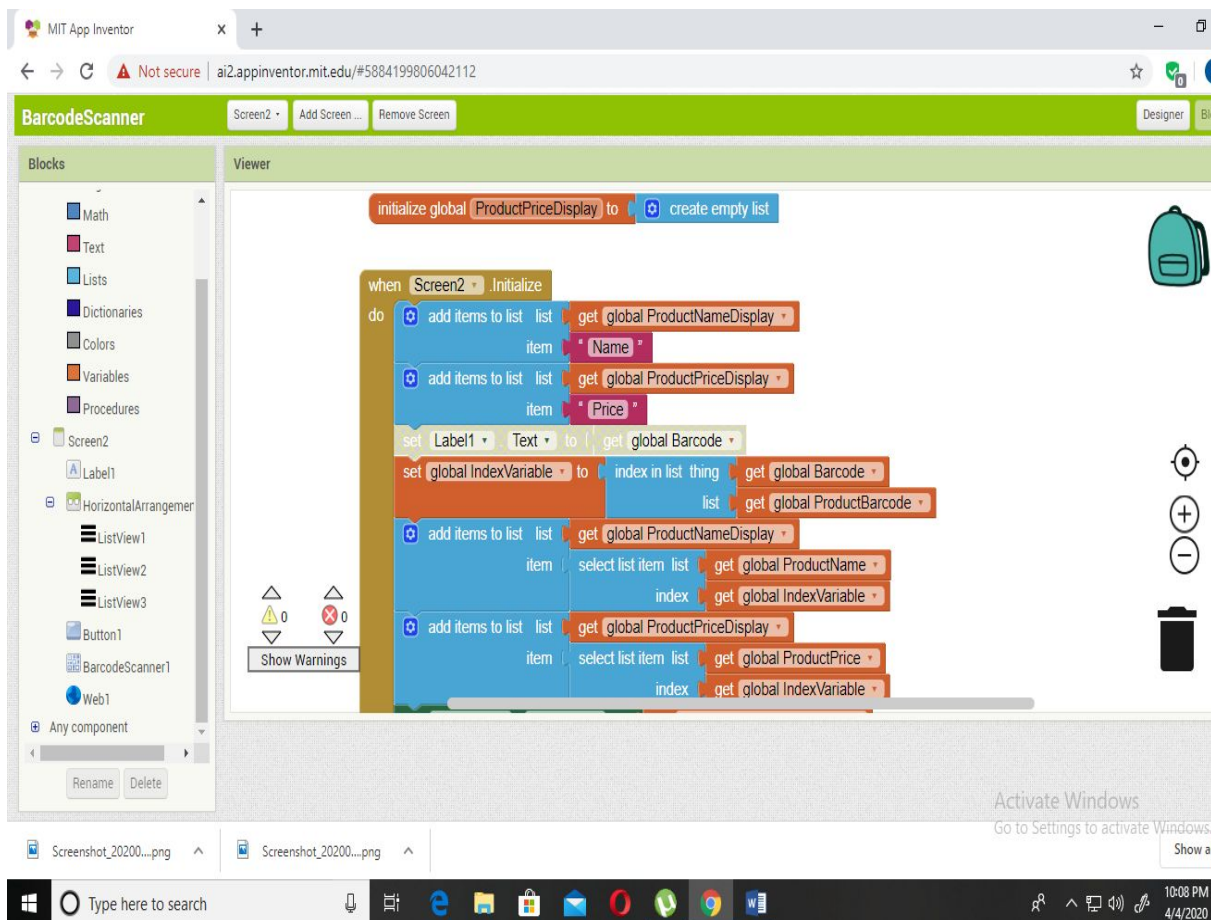
One of the neatest things about App Inventor is that you can see and test your app while you're building it, on a connected device. If you have an **Android phone or tablet**, follow the steps below. *If you do not have a device*, then follow the instructions for [setting up the on-screen emulator](#) (opens a new page) and then come back to this tutorial once you've gotten the emulator connected to App Inventor.



8)



9)



3.3 Implementation:

- HC - SR04 Ultrasonic Ranging Module :

```
Counter
//THIS CODE WORKS FOR + AND - COUNTER!!!!!!!!!!!!!!!!!!!!!!
#define trigPin 13
#define echoPin 12
int counter = 0;
int currentState = 0;
int previousState = 0;
void setup() {
  Serial.begin (9600);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
}

void loop() {
  long duration, distance;
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = (duration/2) / 29.1;
  if (distance <= 10){
    currentState = 1;
  }
  else {
    currentState = 0;
  }
}
```

```
    delay(500);  
    if(currentState != previousState){  
    if(currentState == 1){  
        counter = counter + 1;  
        Serial.println(counter);  
    }  
    }  
    if(distance >= 10 && distance <=50){  
        counter = counter - 1;  
        Serial.println(counter);  
    }  
}
```

- **Servo Motor :**



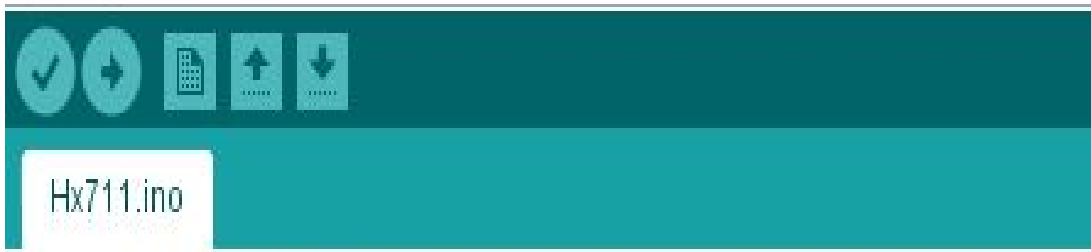
```
#include <Servo.h>

Servo servo;

void setup() {
  servo.attach(2); //D4
  servo.write(0);
  delay(2000);
}

void loop() {
  servo.write(90);
  delay(1000);
  servo.write(0);
  delay(1000);
}
```

- **Weight Sensor (HX711) :**



```
#include <HX711.h>

// Scale Settings
const int SCALE_DOUT_PIN = D2;
const int SCALE_SCK_PIN = D3;

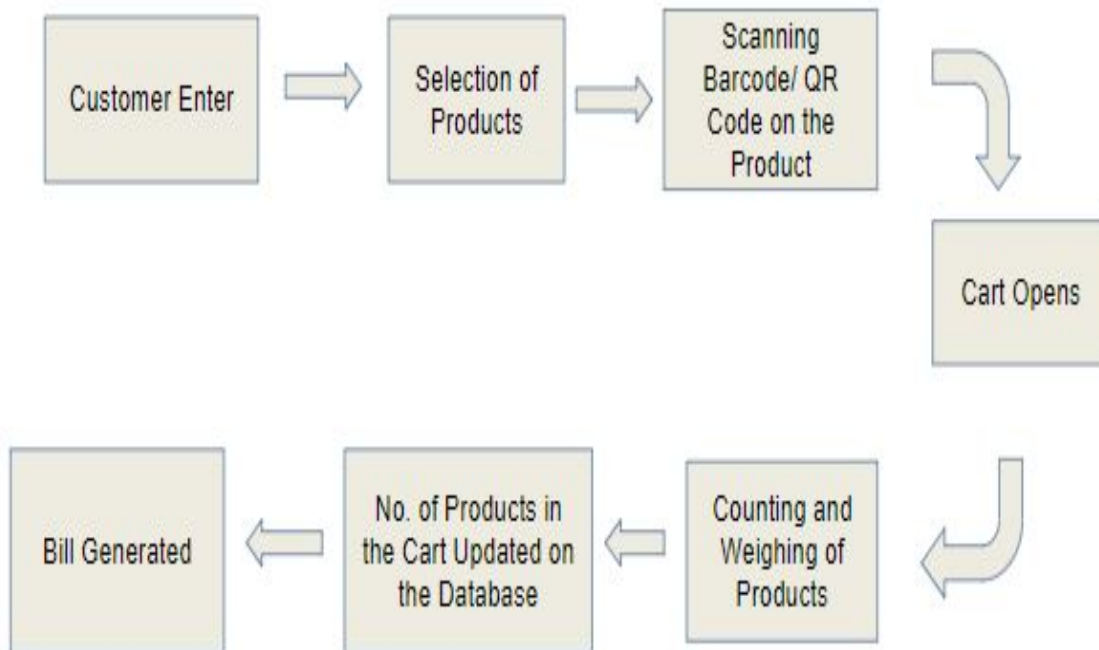
HX711 scale(SCALE_DOUT_PIN, SCALE_SCK_PIN);

void setup() {
  Serial.begin(115200);
  scale.set_scale();// <- set here calibration factor!!!
  scale.tare();
}

void loop() {
  float weight = scale.get_units(1);
  Serial.println(String(weight, 2));
}
```

Chapter No. 4

Block Diagram



Block Diagram Explanation

1. Customer Enter

- The customer enters the supermarket or the store
- The customers need to register themselves to get a smart cart
- They collect the cart and establish a connection between the smart cart and the mobile device
- The connection establishment is done using Bluetooth module of the smart cart and the mobile device

2. Selection of Products

- The customers pickup the products themselves that they want to purchase
- The products should be picked up one by one
- It is easy for the customers to scan the products one by one
- It is mandatory for the products to have a barcode
- The customers should check if the barcodes of the products are not damaged

3. Scanning Barcode or code on the Product

- After the selection of the product the customer should open the application on the mobile phone
- Once the application opens, click on SCAN
- A scanning window appears and starts scanning
- Place the product barcode in front of the scanner
- The cart opens
- The information related to the product is stored in a local database table
- After scanning of one product is complete, click on ADD to add more items
- Once again the scanning window appears and then scan the product barcode
- The cart opens again
- A local table with the information of the products can be viewed in the application on the mobile phone

4. Cart opens

- After scanning the barcode each time a signal is sent to the NodeMCU
- The NodeMCU sends a signal and triggers the Servo Motor on the lid of the smart cart
- The servo motor lifts the lid of the smart cart for a set buffer time
- The product needs to be put inside within this stipulated buffer time
- After each scan the lid opens up and gets closed after the buffer time
- The product has to be scanned and put inside the smart cart one by one
- The lid of the smart cart gets closed with the help of the same servo motor
- The opening of the smart cart lid takes place only after a product is scanned and the closing takes place automatically after the buffer time is over

5. Counting and Weighing of Products

- Once the product enters the smart cart there are two operations taking place
- The first operation is the counting of the object(s) entering the cart at a time
- The ideal number of products entering the cart at a time should be ONE
- The second operation taking place is done by the weighing module that is present in the smart cart
- The weighing module after addition of products each time adds up the weight
- This weight is matched with the total weight on the table of application in the mobile phone
- If at any moment there is a discrepancy between the data on the application and that of the smart cart regarding the count or weight of the products the alarm system of the smart cart starts buzzing immediately

6. Number of products in the cart updated on the database

- All the products are scanned and put inside the cart one by one
- The product gets added on the table of the application on the mobile phone
- After scan of the each product the table is updated again
- After all products are put in the cart the final product table gets updated and also the supermarket or store database of products get updated

7. Bill Generated

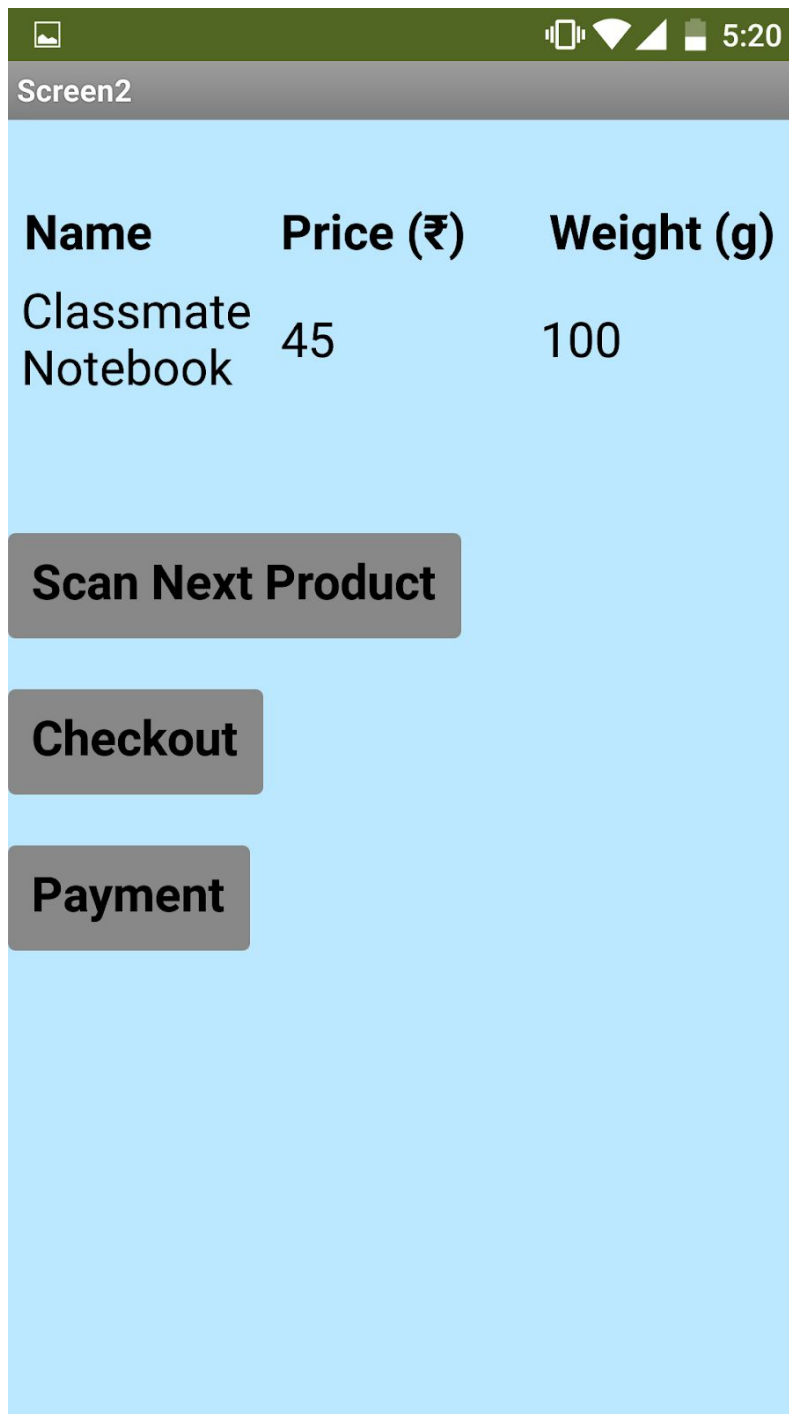
- After scanning all the required products and putting them in the cart there is an option on the application
- This option is GENERATE BILL
- This calculates the total price of all the products from the table on the application in the mobile phone
- Then a detailed bill with Product name, Total count, Total Weight and Total Price is generated
- The bill payment can be done using extended services like Online Payments or Money Wallets by linking them to the application

Chapter No. 5

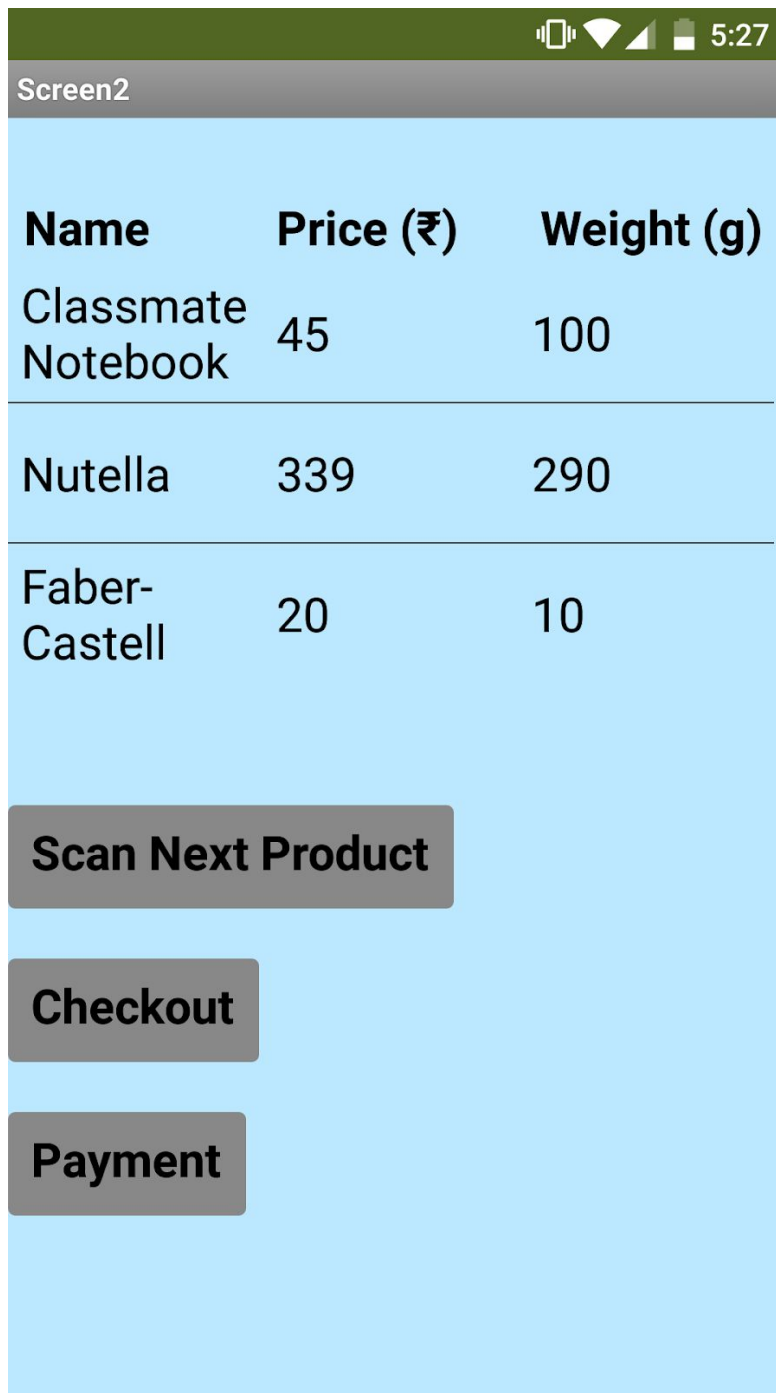
Results and Future Prospects



First Result: Home Screen



Second Result: After Scanning 1st Product



Third Result: After Scanning Multiple Products

Screen2		
Name	Price (₹)	Weight (g)
Classmate Notebook	45	100
Nutella	339	290
Faber-Castell	20	10
Total	404	400
Scan Next Product		
Checkout		
Payment		

Fourth Result: Checkout



Hi Jugal!



Pay



UPI Money
Transfer



Add Money



Recharge & Pay
Bills



Banking &
Finance



Travel & Flights



Nearby Stores



Online Shopping



More

Pay from any
Bank Account. Get Flat
₹50 Cashback

T&C Apply

Online

At Stores



Home



Shop



Scan Any QR



Bank



Inbox

Final Result: Payment using Paytm

Final Test Code -

```
//FINAL PROGRAM DO NOT USE ANYTHING ELSE
#include <SoftwareSerial.h> // TX RX software library for bluetooth
#include <Servo.h>
//#include <ESP8266WiFi.h>
#include <HX711.h>
#define trigPin 13
#define echoPin 12
int counter = 0;
int currentState = 0;
int previousState = 0;
int bluetoothTx = 10; // bluetooth tx to 10 pin
int bluetoothRx = 11; // bluetooth rx to 11 pin
HX711 scale(5, 4);
float calibration_factor = -10000000;
float units;
Servo myservo;
SoftwareSerial bluetooth(bluetoothTx, bluetoothRx);

void setup()
{
  myservo.attach(9); // attach servo signal wire to pin 9
  //WiFi.begin("Bhargav", "123456789");
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  Serial.begin(9600);
  bluetooth.begin(9600);
}
```

```

void loop()
{
    //Read from bluetooth and write to usb serial
    if(bluetooth.available() > 0 ) // receive number from bluetooth
    {
        int servopos = bluetooth.read(); // save the received number to servopos
        Serial.println(servopos); // serial print servopos current number received from bluetooth
        delay(1000);
        myservo.write(servopos); // rotate the servo the angle received from the android app
        delay(15000);
        myservo.write(120);
    }

    long duration, distance;
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);
    duration = pulseIn(echoPin, HIGH);
    distance = (duration/2) / 29.1;
    if (distance <= 10){
        currentState = 1;
    }
    else {
        currentState = 0;
    }
    delay(2000);
    if(currentState != previousState){

```

```

if(currentState != previousState){
  if(currentState == 1){
    counter = counter + 1;
    Serial.print("No. of Objects=");
    Serial.println(counter);
  }
}
if(distance >= 10 && distance <=50){
  counter = counter - 1;
  Serial.print("No. of Objects=");
  Serial.println(counter);
}
scale.set_scale();
scale.tare();
scale.set_scale(calibration_factor); //Adjust to this calibration factor

Serial.print("Reading: ");
units = scale.get_units(), 10;
if (units < 0)
{
  units = 0.00;
}
Serial.print(units);
Serial.print(" grams");
Serial.println();
delay(1000);

if(Serial.available())
{
  char temp = Serial.read();
  if(temp == '+' || temp == 'a')
    calibration_factor += 1;
  else if(temp == '-' || temp == 'z')
    calibration_factor -= 1;
}
}

```

Smart Shopping Kart Model



A model Smart Shopping Kart was designed and integrated as a part of this project. Trivial shopping karts consist of a regular metal chassis capable of only “holding” its contents and that's it. To satisfy our aim of automating the entire shopping experience, we developed a “Smart Shopping Kart” which enables a shopkeeper to keep track of the quantity and the amount of goods sold. With the help of the “Smart Shopping Kart” the shopkeeper can keep a track of everything being bought by a single customer and the accurate amount of bill paid, which ensures that the automated shopping process is completely secure and both the consumer and the shopkeeper can go through a hassle free shopping experience unlike the regular one where a customer has to spend a huge amount of time waiting in queues just to pay the bills.

The “Smart Shopping Kart” consists of a closed lid which opens only when a product is scanned by a customer via the Smart Shopping Application. There are two servo motors placed on the lid of the kart which help in the opening and the closing of the kart respectively. A signal is sent to the main microcontroller placed in the kart as soon as a product is scanned using the Barcode Scanner that we have developed as a part of this project through the

application and the kart is thus opened. To ensure the number of products that are being scanned by the customer and the number of products that are being placed in the kart by the customer are equal, we have used an ultrasonic sensor (HcSR04) which helps us to keep an accurate track of the number of products being placed inside the kart. To ensure complete security of our Shopping Model we have also integrated a Load Cell and a weight sensor (HX711) with the Smart Shopping Kart which will help us in keeping track of the total weight of the products being scanned by the customer and the compare it with the total weight of products being placed inside the kart. This ensures complete security of our kart and minimizes the scope of theft or falsification.

Future Prospects of the Smart Kart

1. We can move our database into relational databases like MySQL for better access and also for storing large amounts of data.
2. We can also build a better application using other platforms to get access to more advanced functions, and maybe make one for iOS as well.
3. We can add new features to our existing application to make it more dynamic with more functions.
4. A version of the application can be developed where the user interface is more friendly and easy to use.
5. By continuing work on it, we'll keep on making our application faster step by step - faster scanning and processing especially.
6. An improved version of the Hardware - Software integration can be developed.
7. Hardware placement could be done in a more efficient way so that the maximum space of the cart could be utilised.
8. The major intent of this project is to maximise automation in the supermarket shopping industry.

Chapter No. 6

Discussions

Our main purpose for choosing this particular topic was to introduce a different approach of a shopping system where a user doesn't have to go through the same hassle of standing in long queues, waiting for a person to complete the billing of all the people in line before him and in turn wasting a huge amount of his time. Automation has become a huge part of our daily lives and it is that thing which will drive our future on a path which is trouble free and more consistent in areas which currently take up a huge amount of our time and efforts. We have kept this in mind and developed such an automated shopping system which will revoke the intervention of an extra human in this cycle where it is not necessarily required. We have successfully developed a Barcode Scanner and integrated it using our application that we designed ourselves. This scanner, unlike the trivial ones, doesn't require a Barcode Scanner hardware module and instead is available on Your smartphone and uses the existing camera present on Your smartphone. This saves up a huge amount of expenditure which would otherwise have been the Shop owner's headache to buy a certain number of Barcode Scanners and let his employees use them. Now all he has to do is ask his customers to download our application and that's it.

We were able to solve another problem with our application and that was the elimination of the RFID cards used in previous versions of the Smart Shopping Systems. The previous versions included the usage of RFID cards on every product and the installation of a RFID scanner at the checkout point. This again increased the total cost of the product as every product then came along with a RFID tag attached to it and the installation costs of the active RFID scanners was an added expense. We were able to fathom this issue thoroughly and resolve it using our Barcode Scanner which repudiates the use of any hardware scanner module and instead uses an inbuilt Barcode Scanner to scan all the products, in turn making this entire process almost cost free.

Chapter No. 7

Conclusion

Thus we have successfully designed and developed an automated Smart Shopping System that ensures a hassle free experience to the consumer wherein he doesn't have to wait in a long queue just to get his products scanned and pay the bill. Instead the consumer will himself be able to scan all the products using the Barcode Scanner that we developed and pay his bill using Paytm using our very own application. This will ensure a cash free transaction as well which is very demanded in such exacting times like the ones we are currently in. We were able to test our system on over 15 different items present in the shopping store and we were able to get absolute accurate results with respect to their weights, quantities and Price. Our system ensures no human to human interaction as the consumer is the only person present in the picture and he will have all the facility to complete the entire process right from the choosing of products, scanning them and even paying for them to himself. Thus ensuring no human to human transaction and also no long waiting queues for bills to be paid.

References

1. <https://www.irjet.net/archives/V5/i4/IRJET-V5I444.pdf>
2. https://www.academia.edu/36673808/IOT_Based_Smart_Automatic_Shopping_Cart_with_Overload_Indicator
3. https://www.thinkmind.org/download.php?articleid=sensorcomm_2013_7_30_10155
4. https://www.ijareeie.com/upload/2017/march/130_MBITS_EEE8.PDF