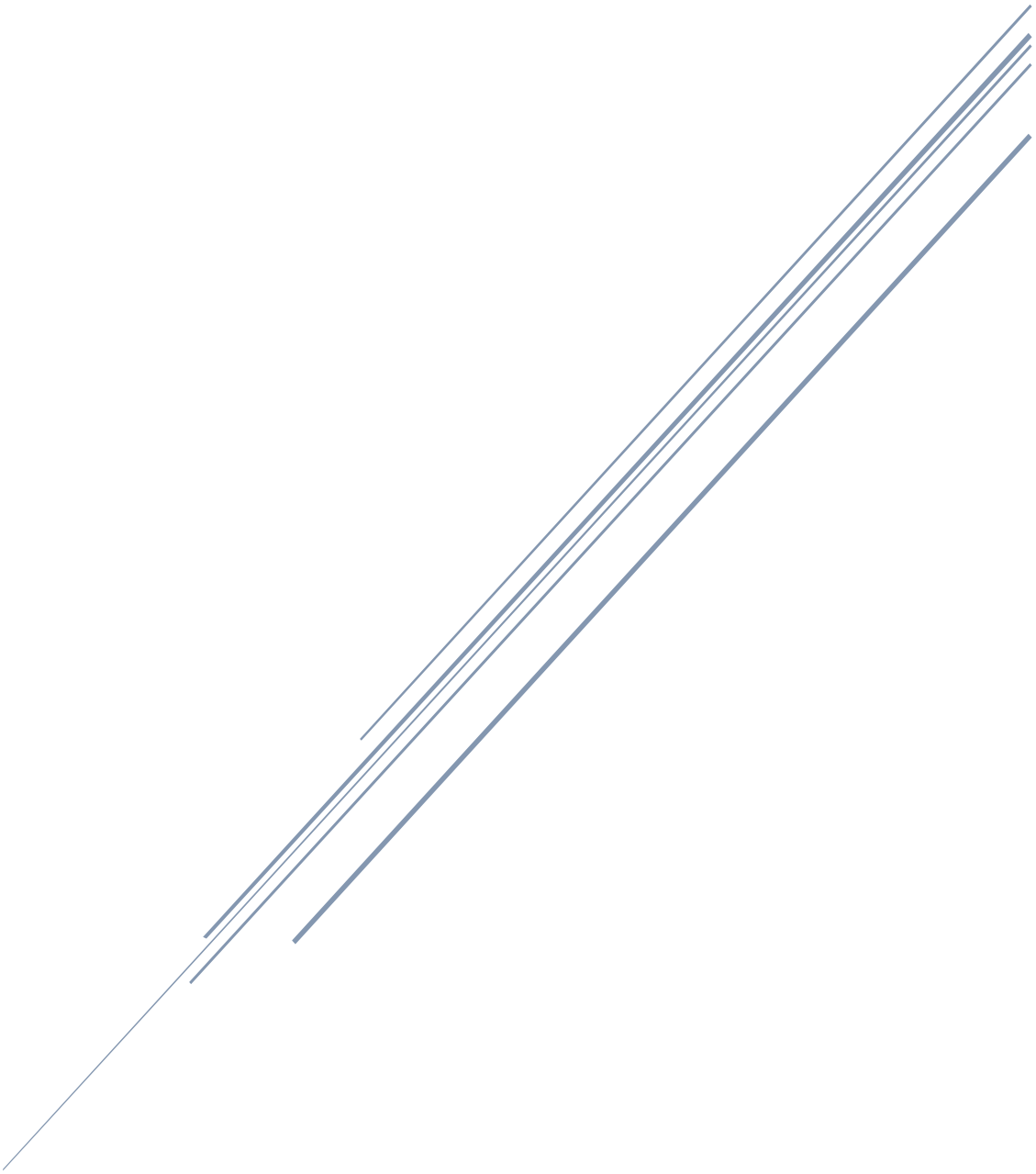
SMART SHELF PROJECT REPORT

Divya Kalkotwar

Sagar Sharma

Shivaram Singh Rajput



KTH Royal Institute of Technology

IS-2500 RFID SYSTEMS

Contents

List of Figures

Abstract

The report outlines the design, implementation of ‘Smart Shelf’ and its applications in supermarkets and fulfilment centres. Supermarkets and fulfilment centres require manual intervention to count the number of items placed on the shelf to provide continuous flow of stock to its customers. The project ‘Smart Shelf’ takes advantage of radio frequency identification technology to determine the stock available on a shelf.

Introduction

The smart shelves enable the retailer supermarkets such as Walmart, Lidl, Aldi etc. and fulfilment centres such as Amazon FCs to determine the inventory available on the shelves in real time. With the help of radio frequency identification technology, the supermarkets and fulfilment centres can control and keep products in continuous stock by automatically keeping track of the items available on the shelf. Instead of an employee manually going to each shelf and counting the number of items available, using a smart shelf helps determine the number of items available on the shelf automatically when scanned by the RFID reader, thereby, reducing time and increasing efficiency in keeping inventory. Another advantage of the smart shelf is that, they are significantly cheaper to design and adopt.

Hardware Architecture

Block Diagram

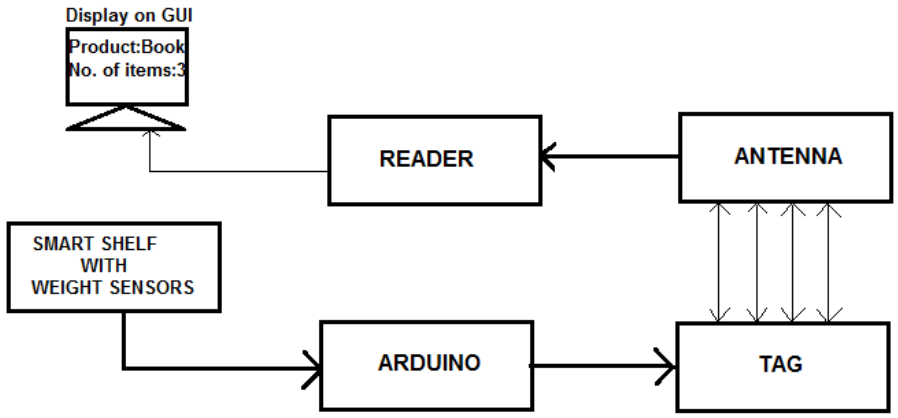


Figure 1 Block diagram of the Smart shelf system.

The Smart shelf comprising of weight sensors (Force sensitive resistors) is connected to Arduino. Force sensitive resistors are connected as shown in fig 6, one end of this series connection is

connected to the 5V pin of Arduino and the other end is grounded. The Voltage across 1MΩ Resistor is measured through the analog input pins of Arduino using the concept of voltage divider circuits. Based on the Voltage difference measured at the analog pins of Arduino, the number of Items left on the shelf is calculated. The number of items on the shelf and synchronization bits are sent from Arduino to the reader by modulating the RFID ISO tag using OOK modulation.

The data sent by OOK modulation is received by the reader through a loop antenna which resonates at 13.56MHz frequency. Once the information is received at the reader by demodulating the tag OOK sequence, the product and the number of items are displayed on the GUI.

Arduino Uno

Arduino Uno is a microcontroller board based on ATmega328p with 14 digital input/output pins, 6 analog input pins, 16 MHz quartz crystal, USB connection, a power jack and a reset button. Uno can be programmed using Arduino IDE. The ATmega328 on the Uno comes pre-programmed with a boot loader that allows you to upload new code with ease. It communicates using the original STK500 protocol. The board supports communication over USB and appears as a virtual COM port to software on the computer, without any external drivers. The Arduino IDE includes a serial monitor which allows the data to be sent to and from the board. In our project, Arduino Uno is used to calculate the number of items present on the smart shelf and to modulate the RFID tag with OOK modulation.

ISO TagIt Tags

ISO TagIt is an ISO standard, operating at 13.56 MHz frequency and offer maximum distance of 1.5 Meters. Communication from the reader requires amplitude shift keying. The tag is modulated by the Arduino such that the RFID reader receives the number of items present on the smart shelf.

Figure 2 Transponder tag

TI S6350 RFID Reader

The reader operates in high frequency band 13.56 MHz and antennas utilize the magnetic field to transfer power to the tag. When antenna is connected to the reader, the reader requires 50Ω impedance, quality factor less than 20 and SWR closer to 1. It supports Tag-it HF, Tag-it HF-I (ISO15693 compliant) and all other ISO15693 compliant transponders from various suppliers.

Figure 3Reading from Antenna Analyser connected to our Antenna

Antenna

The antenna used for the project is a loop antenna of 32 X 32 cm dimension made of solid copper wire making it rigid & self-supporting. The antenna resonates at 13.56MHz.

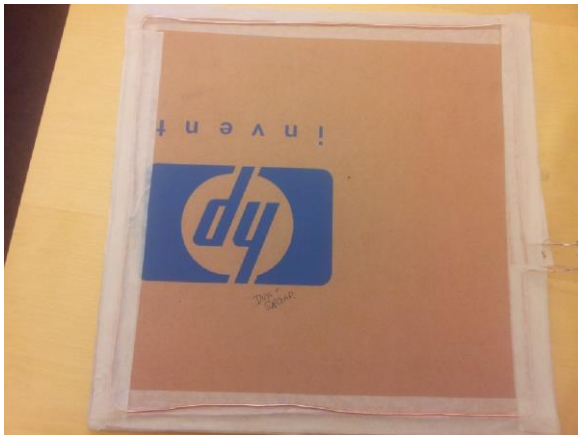


Figure 4 Antenna Layout (32X32 cm)

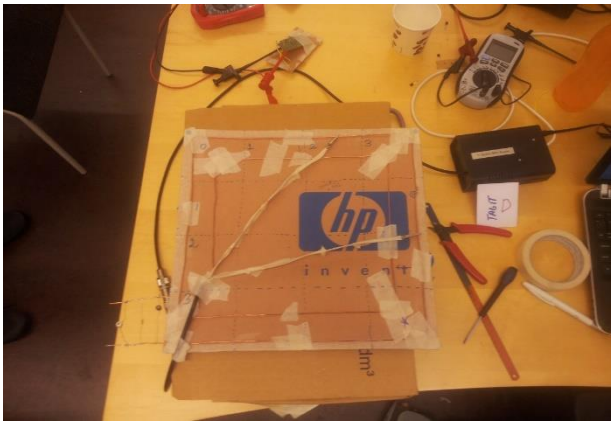


Figure 5 Antenna with T Match

Weight Sensors and functionality

The weight sensors used for the project are made by the team and are built using single sided copper plates, conductive foam and wires. Each copper plate of 5 X 5 cm dimension is soldered to a wire. Two copper plates are placed on each other with conductive foam in between. On applying pressure/force the resistance across the plates changes. Nine such sensors are used in the project. When all the nine sensors were connected in series, the total resistance was highly unstable as it would vary from 20M to above 200MΩ. Thus, we decided to use the sensors in the arrangement shown in figure 6. There are 9 weight sensors in the smart shelf. Two weight sensors are connected in series with a 1MΩ resistor to form a group. Four such groups are then connected in parallel. This is next connected in parallel to the last weight sensor and 1MΩ resistor (Both in series). As shown below, one end of the circuit is connected to 5V and the other end to ground. Using the concept of a voltage divider circuit, we measure the change in average voltage across the 1MΩ resistor on all the 5 branches by reading the voltages through the analog pins of the Arduino (A0 to A4).

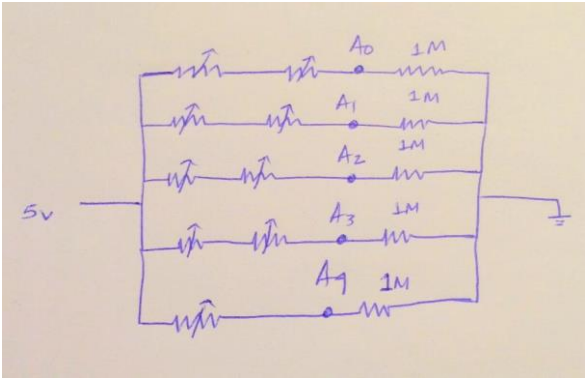


Figure 6 Arrangement of weight sensors

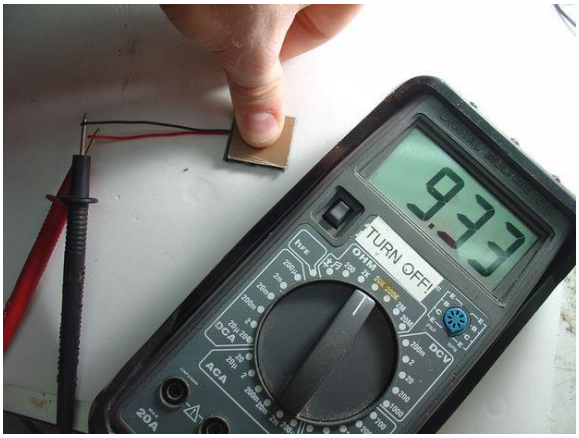
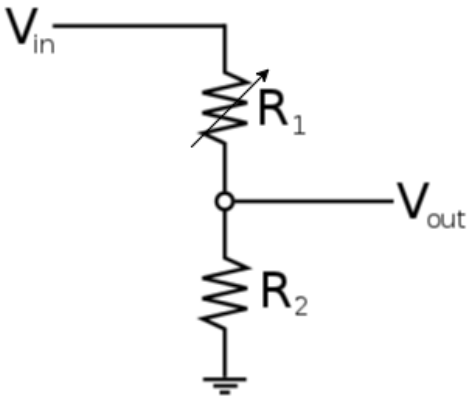


Figure 7 (Left)Change in resistance on application of force & (Right)Arrangement of weight sensor .



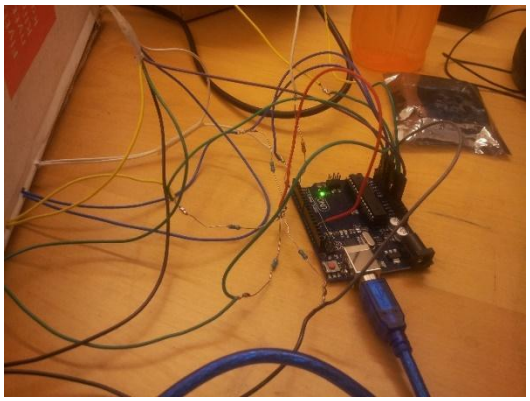


Figure 8 Voltage divider circuit to measure the resistance from weight sensor, connected to Arduino

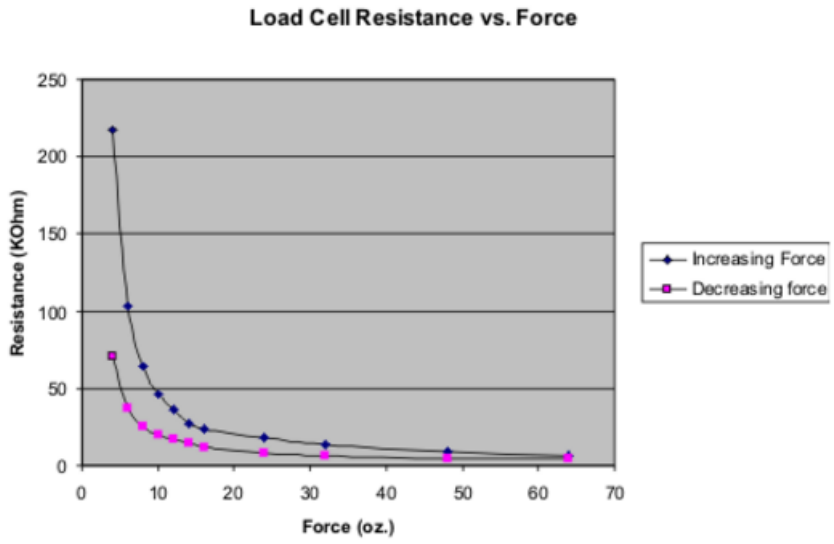


Figure 9 Graph representing change in resistance with applied pressure on the sensors

OOK Modulation

On-Off keying denotes amplitude shift keying that represents a series of 1's and 0's by simply switching the carrier on and off. We chose to use OOK as a modulation scheme, because it was simple and sufficient for the project. We used the concepts of RFID where a tag backscatters the RF signals received by using a modulation scheme. In our project we send the information (number of items on smart shelf) from the Arduino to the reader. We do this by switching the tag on and off and the reader receiving this information demodulates the information by interpreting the ON as a "1" and OFF as a "0". To achieve this, we converted the number of items on the shelf in the Arduino to bits of 1's and 0's and simply switched the tag on and off using a NPN BC547 transistor as a switch.

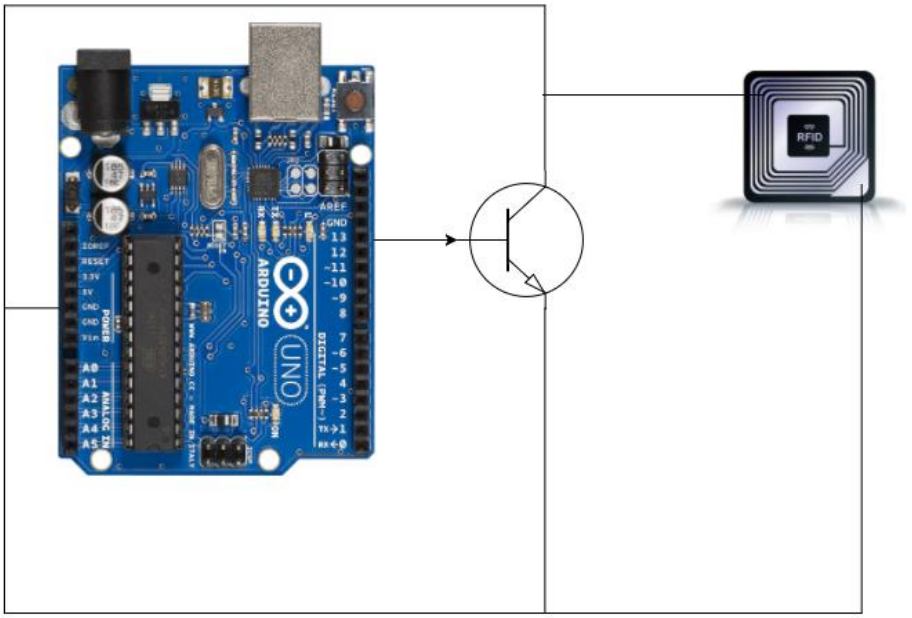


Figure 10 OOK modulation for sending data to reader using transistor as switch and tag as Antenna.

As shown above, the transistor is used as switch to modulate the tag. The base of the transistor is connected to a digital output pin of the Arduino and the collector and emitter are connected to two ends of the tag antenna which are connected to a common ground of the board. If the tag has to be read, the base of the transistor receives a high input from the board which allows the flow of current through the antenna to the IC in the tag. If the tag should not be read, the base of the transistor receives a low input from the board which connects the tag antenna to ground that breaks the circuits and does not allow the flow of current through the antenna to the IC in the tag.

Outline of functionality

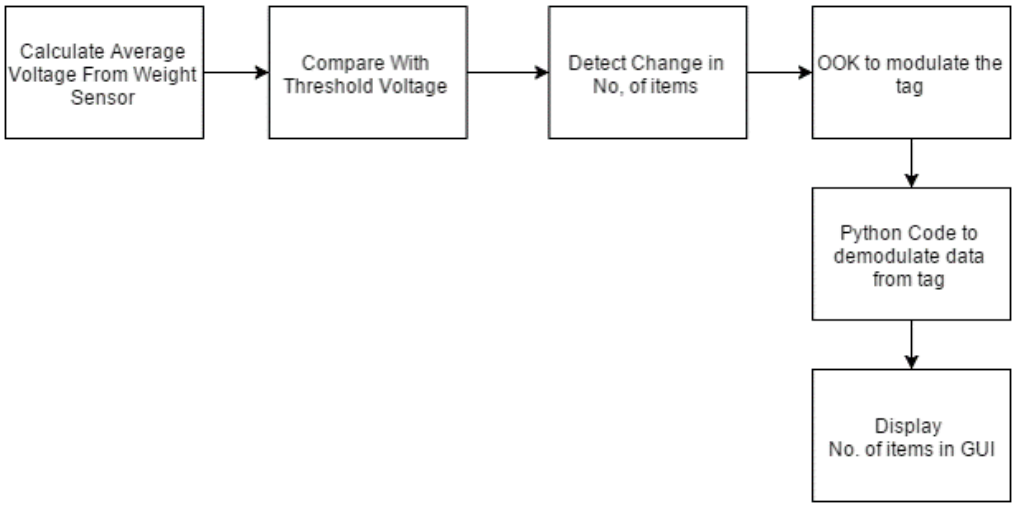


Figure 11 Representation of the outline of functionality of the system

The shelf consists of 9 weight sensors, as shown in fig 6 and the voltage across the 1MΩ resistor is measured at one of the five analog pins of the Arduino. When an item is placed on the shelf, an average of all five voltages measured at the analog pins is calculated.

The average voltage is calculated every 500ms and the change in this average voltage from previous voltage is detected by placing or removal of item..

The resistance of weight sensors decreases exponentially, when pressure/force is applied to the sensors as shown in fig 10. As a result, we set threshold values to detect the change in number of items present on the shelf. The threshold values used for our setup are [0.45, 0.34, 0.22, 0.17, 0.09, 0.09, 0.04, 0.03, 0.03, and 0.03]. For example, when there are no items on the shelf and the average voltage measured is V1 and the voltage after placing an item on the shelf is V2. If the difference of V2 and V1 is greater than 0.45, then we increment the number of items by 1. To detect the second item on the shelf, the difference in the voltages should be greater than 0.34 and so on. Same idea is used to detect the removal of an item from the shelf, instead of looking for increment in the voltage we look for decrement in the voltage.

Once the number of items on the shelf is detected, the information is used in the OOK modulation of the tag. Before sending the 'number of items', 5 synchronization bits are sent i.e. "01111".Therefore in total 10 bits (Synchronization bits (5bits) + Number of items (5 bits)) of data is sent through the OOK modulation. The modulation takes place at the rate of 50ms/bit.

The modulated information is received using the TI S6350 RFID Reader. At the receiver, we initially look for the synchronization bits and once there is a match, the following 5 bits representing the number of items is received and displayed on the GUI.

Software Flow

Sender: Arduino

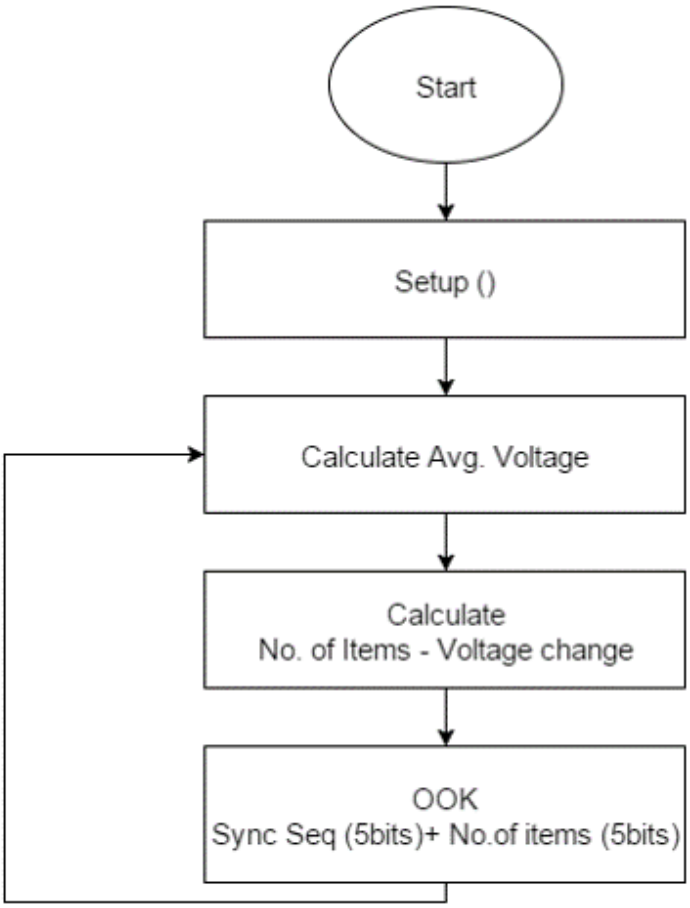


Figure 12 Flowchart of Arduino code.

Initially the pins on the Arduino board are configured, then the following 3 steps occurs continuously every 500ms. The figure is self-explanatory and all the steps are discussed in detail in the previous sections.

Receiver: Reader

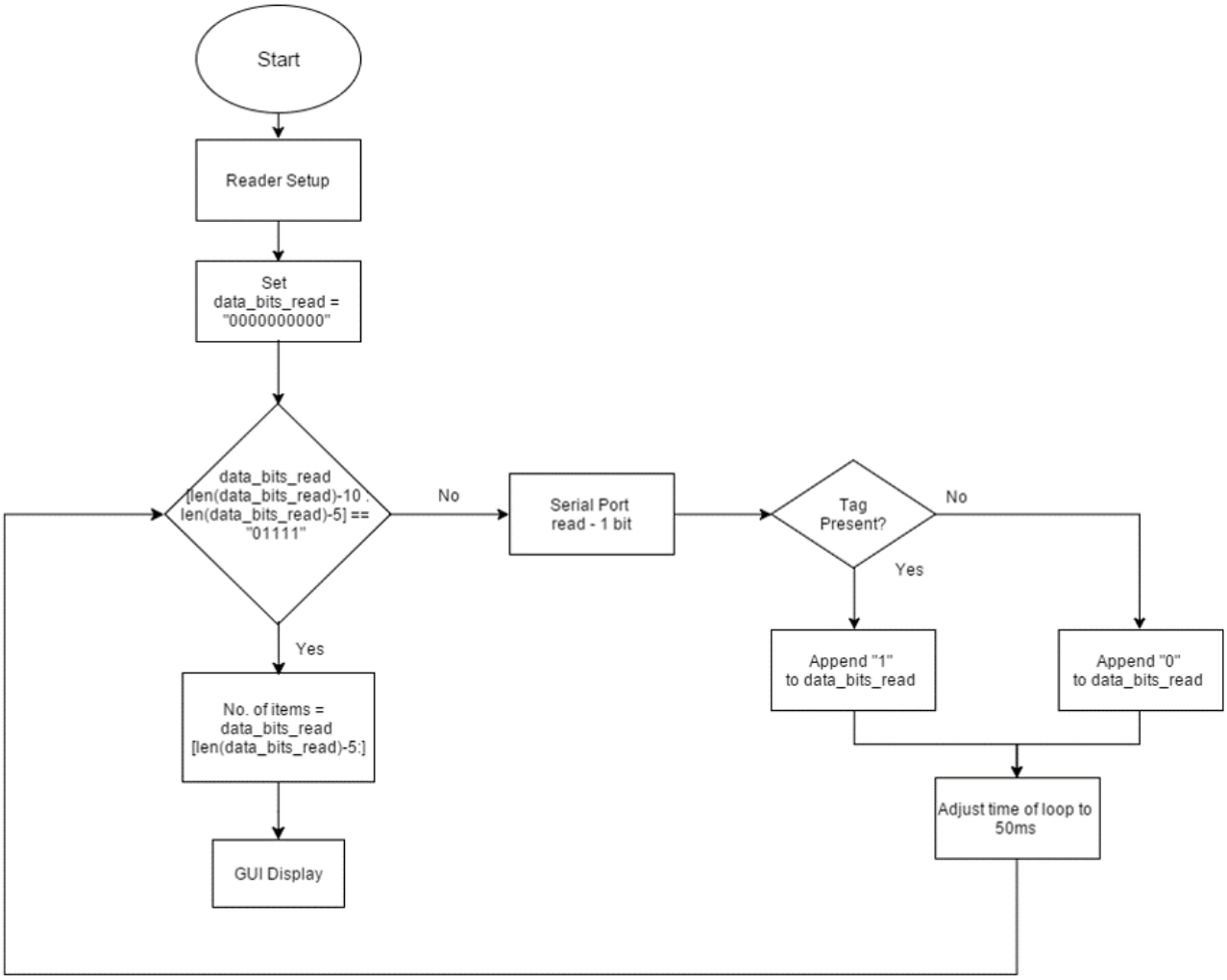


Figure 13 Flowchart of Python Code

The TI S6350 RFID reader which is connected to the PC, can be controlled using Python serial port extensions (pyserial). The initial setup such as the COM port through which the communication is established and the baud rate at which it should operate etc. are set to communicate with the reader.

Once the initial setup is done, the tag will be read with the help of reader every 50ms. The tag is being modulated using OOK modulation, when the tag sends a 0, the reader cannot detect the tag and when the tag sends 1, and reader detects the tag. The series of one's and zero's received by the reader is stored as a growing string, since the tag is always sending the same 10 bits repeatedly, the 10^{th} bit to 6^{th} bit from the end of the string is checked and if it is a match with the synchronization bits"01111", the rest 5 bits, i.e. 5^{th} bit from the end of the string to end of the string, contains the information of number of Items on the shelf and it is displayed on the graphical interface.

Calculation of the threshold voltages

There are nine weight sensors which are used to detect the weight on the shelf as shown in fig 6. The weight sensors behaviour varies exponentially with the weight being placed on it as shown in fig 10. When the first item is placed on the sensor the increase in voltage across the 1MΩ resistor is large, but when more weight is placed on the sensor the increase in the voltage reduces. This method shows that the minimum weight the shelf is able to detect is 500gms or more, because the variation in the voltage after placing the item on the shelf is large enough to differentiate among scenarios like additions, reductions and no change. To be able to clearly detect the above scenarios threshold voltages have to be set. We ran five sets of trials to find threshold voltages for each increasing item placed on the shelf.

Trial 1: Keeping the threshold constant

Threshold = 0.3V		
0	0.6	0
1	1.2	0.6
2	1.77	0.57
3	2.13	0.36
4	2.4	0.27
5	2.99	0.59
6	3.32	0.33
missed	3.44	0.12
missed	3.66	0.22
missed	3.8	0.14
missed	3.84	0.04

Trial 2: Decreasing the constant threshold

Threshold = 0.2V		
0	0.44	0
1	1.18	0.74
missed	1.36	0.18
missed	1.46	0.1
missed	1.58	0.12
2	1.95	0.37
missed	1.97	0.02
3	2.55	0.58
4	2.76	0.21
missed	2.92	0.16
missed	2.97	0.05
missed	3.05	0.08
missed	3.16	0.11

Trial 3: Assessing the threshold voltages for each increment

Items on shelf	Initial voltage	Final voltage	Diff in voltage
1	0.44	1.18	0.74
2	1.46	1.58	0.12
3	1.58	1.95	0.37
4	1.97	2.55	0.58
5	2.76	2.92	0.16
6	2.97	3.05	0.08
7	3.16	3.23	0.07

8	3.24	3.28	0.04
9	3.29	3.35	0.06

Trial 4: Assessing the threshold voltages for each increment

Items on shelf	Initial voltage	Final voltage	Diff in voltage
1	1.01	1.97	0.96
2	2.1	2.59	0.49
3	2.64	2.96	0.32
4	3.197	3.42	0.223
5	3.55	3.88	0.33
6	3.91	4.06	0.15
7	4.14	4.18	0.04
8	4.21	4.24	0.03
9	4.25	4.29	0.04

Trial 5: Assessing the threshold voltages for each decrement

Items on shelf	Initial voltage	Final voltage	Diff in voltage
9	4.35	4.32	0.03
8	4.35	4.31	0.04
7	4.33	4.27	0.06
6	4.27	4.23	0.04
5	4.24	4.02	0.22
4	4.07	3.86	0.21
3	3.6	3.29	0.31
2	3.31	2.17	1.14
1	2	1.208	0.792

The above trial helped us find the initial values for the threshold voltages. Several more testing was done to try and get the threshold voltages right such that an increment and decrement can be detected by the shelf. The final threshold values used were:

threshold[10] = {0.45,0.34,0.22,0.17,0.09,0.09,0.04,0.03,0.03,0.03};

After testing this several times we were able to get 85% accuracy of detecting the addition, reduction and no change of items. We found that since the weight sensors were unstable, the threshold values were unstable too. We tested the following scenarios:

- Placement of book in 1 stack - This test passed up till 4 books were placed.
- Placement of book in 2 stacks side by side – This test passed up till 5 books were placed.
- Placement of books randomly (book surface area not in contact with the shelf) - This test case failed because if the book's entire surface area is not in contact with the shelf, the change in voltage is not significant enough to detect an addition or reduction.
- Increment or decrement of an item - This test case passed because there is a significant change in voltage to define an addition or reduction of a book
- Increment or decrement of more than one item - This test case failed because although we add more than one book on the shelf, the voltage change is not significant enough to detect more than one book
- More than 5 books – This test case failed because after adding 5 books, the 6th book's addition made a voltage change of 0.02-0.03V which was too small and insignificant. This is a property of the weight sensors used as shown in fig 10.

Characterizing the antenna

We used the antenna built in the Lab 2 as the antenna for the reader. The size of the antenna is 32x32cms and the maximum distance vertically where a tag can be read by the reader is 4cms. The tag can be read by the antenna if placed anywhere within this space.

OOK Scheme solution

In our project the reader demodulates the tag data received to find the number of items placed on the smart shelf. We are aware that there are two parts to the design. One is the Arduino and the other is the reader. Both these components are performing their respective duties without synchronization between them. The Arduino first calculates the number of items and then sends the 5 bits of data to the reader by modulating the tag using OOK. Meanwhile the reader is trying to read the bits from the Arduino by demodulating the data of the tag. Since there is no synchronization

between the two there is a necessity of using a synchronization sequence before sending the data bits so that the reader is aware when the data is valid. We chose 5 bits for the data (maximum 31 items) and thus had to choose a minimum of 5 bits for the sync sequence. The Arduino sends 10 bits of data (5 sync bits and 5 data bits). The start sync sequence is chosen to be “01111” because continuous 0’s or 1’s are not favourable. Now that the Arduino sends 10 bits, the reader must read 10 bits.

In the initial python code of the reader, it would read 10 bits in a loop of 10. At the end of every loop, we issued a sleep command for the ‘symbol time’ predefined by the team. Initially we chose a symbol time of 1sec per bit, which meant that the Arduino modulated the tag for a symbol time of 1sec for each bit being sent. Similarly, in the python loop, after the instructions, we issued a sleep command for 1 sec. Apart from this the algorithm in the python code would always read only 10 bits irrespective of when it is issued. Using this mechanism we got an accuracy of 80%.

An accuracy of 80% is quite low & to improve the accuracy of the synchronization further we had to:

- Find the appropriate symbol width - We found that each loop to read a bit in the python code for the reader does not exceed 36ms. We added a margin and decided a width of 50ms. Now every loop in the python code would run for exactly 50ms. Thus, in the Arduino code, every bit sent is in a gap of 50ms. This ensured that although the two systems may be asynchronous, the widths of the symbols are exactly matched. This helped us find the minimum symbol width to be used.
- Find the right algorithm in the python code of the reader - We decided that the algorithm had to be more robust to scan for the sync sequence. Once the algorithm finds the right synchronization sequence, it can be sure it is synced to the Arduino and it reads the next 5 bits as the data bits. So now the algorithm, when active, starts reading the bits from the tag every 50ms and keeps waiting for the right sync sequence. This algorithm helped us get an accuracy of 95%.

Conclusion – Divya Kalkotwar

I believe that the smart shelf design is very useful and is a simple solution for inventory at supermarkets, fulfilment centres etc. These are the conclusions of the design from my perspective:

1. RFID enabled shelves which work using weight sensors work very well as a tool for inventory in the current scenario for supermarkets & fulfilment centres across the globe.
2. The DIY weight sensors are not stable but work well for a prototype of the smart shelf
3. Calibration of the weight sensors takes lot of time and effort. We had to take several readings to find the right threshold voltages which accurately detected increment and decrement of a book.
4. Working with the weight sensor took a lot of time. We took the help of the professor to cut the pieces of the copper plated board using the PCB cutter. Next the weight sensors were built using conductive foam. The sensors were connected in series to find that they were highly unstable. After that we worked on ways to make it stable, find the right threshold voltage and finally testing the smart shelf.
5. The change in the resistance of the weight sensors is detected by the varying voltage across the 1M Ohm resistor connected in parallel to the weight sensor. By using a voltage divider circuit, the voltage across the 1M Ohm resistor varies with the varying resistance of the weight sensor. Since the weight sensor design is not very precise, we found that the accuracy of the detection of increment and decrement at lower numbers is very high(upto 90%) but if over 5 items are present on the smart shelf and one tries to add or remove an item the accuracy of detecting this event goes down to 50%. This is the reason we limited the number of items to five.
6. The On-Off keying works well and is optimally chosen. We ran several tests to find the right OOK scheme with sync bits. This helped in improving the accuracy to 95%.
7. We worked on several tests to help improve the design. We tested the system using similar books for around 500gms each.

I am happy with the final project output. In my opinion the design is unique because it combines weight sensors and RFID to give a simple and low-cost solution to the problem of inventory in supermarkets and fulfilment centres.

The RFID part of the project involves the interpretation of the OOK of the tag modulated by the Arduino. We use an NPN BC547 transistor connected to the Arduino to act as a switch to modulate the tag. The python code of the reader interprets the OOK and finds how many items are placed on the smart shelf. The antenna connected to the reader is 32x32cms in dimension and a tag can be

read all over the antenna and up to 4 cms of height from the antenna. The antenna is a 13.56MHz antenna and the system is able to read only ISO tags as of now. The estimated cost of the sensor, microcontroller, tag and a plastic platform on which the items will be placed would be 10$ per shelf.

Tag	10 cents
Microcontroller - Atmel	1 Euro
Sensor	1 Euro
Plastic Platform	5 Euro
Extra components	2 Euro
Total	10 Euro

The size of the shelf considered is 60x40cms. This could be a way of not replacing the existing shelves but to add a plastic layer with the sensor and RFID system on every shelf such that the items can be easily placed on the shelf.

The design is not yet perfect. I would like to add some improvements to the existing design. These are improvements on the current design and do not require a complete redesign of the system.

1. Using more accurate sensors to make the product is more reliable.
2. Detect multiple additions and reductions to the smart shelf.
3. Have a simple easy to use interface to calibrate the items placed on the smart shelf.
4. Send the data of multiple shelves using one microcontroller and one tag.
5. The smart shelf should be able to work well for a large range of 50gms to 5kg items.
6. A way to indicate to the customer that the shelf is about to run out of the products on it, probably activating decorative lights on the shelf that might attract more customers.

When I look at the existing design critically I realise that it has some flaws but as a prototype for a concept it is quite convincing. The smart shelf helps to solve existing problems of inventory in supermarkets and fulfilment centres in a non-invasive way by using one the most common attributes of shelves: A shelf generally has items stacked together that weigh the same. The cost of the shelf extension which will perform the weight calculation is a one-time cost for 5 years minimum. This helps the supermarkets and fulfilment centres do inventory in a very simple and reliable manner. The solution proposed is apt for the current scenario where no changes have to be made to the existing shelves at supermarkets and fulfilment centres.

Conclusion – Sagar Sharma

This reports provides explains about implementation and application of smart shelf. We successfully designed force sensitive resistors at very low cost. The shelf was successfully able to read the number of items (books) placed and display it with the help of GUI and is reliable for most of the attempts. As our data indicates we were able to get 85% accuracy for detecting the addition, reduction and no change of items. While the system could read addition/reduction of items with an increment of 1, the design did not work quite well when we placed more than one item on the shelf at the same time and we had incorrect details of the items placed on the shelf. The maximum distance to which the tag could be read is 4 cm. The maximum number of items that can be detected placed on the shelf is 5-6. The minimum weight detected for the item to be detected is 500 gm. While there are number of ways to implement smart shelves, our design is significantly cheap compared to other technologies and easy to adapt in the supermarkets and inventories. In terms of cost,

Sl. No	Item	Cost in Euros
1	TagIt tags	0.3
2	Arduino Uno	20
3	Antenna for reader	<1
4	Sensors	1
5	Reader	~180
6	Resistors, transistors, etc.	1
	Total	204

As you can see, the entire design is significantly cheap. If the cost of the reader is ignored, since a single reader can be used multiple times to read the tag, the overall cost of the system reduces to less than twenty five euros. I am confident that supermarkets such as Wallmart, Aldi, Lidl etc. and fulfilment centres such as Amazon FC's will be interested in the design because of the low cost of design and easy to adapt feature. I think people will be ready to pay about 40 euros for each finished design with interface, because supermarkets such are in need of the product to automate the process of counting the items in shelf and improve utilize the employee's time on more important matters, therefore increasing the efficiency to serve their customers. If we were to redesign the shelf, we would use more stable force sensitive resistors as the sensors were slightly unstable even after we redesigned the alignment and connection. As a result we will be able to detect more than one items placed on the shelf at the same time. Also, if the design was made available for

commercial purposes, we will try to integrate more than 5-6 items. Also we set the thresholds based on a single item. Provided more time we will also integrate different items.

Conclusion – Shivaram Singh Rajput

The weight sensors designed was a force sensitive resistors (FSR) and its resistance decreases exponentially with the force applied. We used 9 weight sensors for our smart shelf and initially all these were connected in series and hence, measuring the voltage across the 1MΩ resistors which was connected in series with these 9 FSRs, became very difficult due to the high fluctuation in the voltages. Then we changed the setup to connect only 2 FSRS in series and which is connected in series with 1MΩ, now we had 4 such connections and 1 single FSR connected in series with 1MΩ resistor. Now the Voltage fluctuation across each of these 1MΩ resistor was very stable and all these very connected to the analog pins of Arduino and the Average voltage was taken. This gave us a very good reading to identify the number of items on the shelf.

The data shown in the tables in section 3 were obtained by using similar books as product, each of those weighed around 500grams. The threshold voltages where set based on continuous testing on the shelf. After hardcoding the values the placing and removal of the books was detected as the increment and decrement of number of items with a good accuracy up to 5 books. Then the system becomes a bit unstable due to the fluctuations in resistance caused by conductive foam and the exponential decrease of resistance.

The demodulation technique used in the python code significantly improved the accuracy of receiving the data. When we only received 10bits every to find the number of items the accuracy was 80% the Accuracy increased to 95% when we read the information about number of items after receiving the Sync bits.The tag was placed right on the antenna in order to get good accuracy. The antenna was able to detect the tag up to a distance of 3 to 4cms. Tag modulation and demodulation gives accuracy of 95% with a data rate of 50ms/bit.

Once the overall setup is done the system is very reliable considering its limitation of detecting limited number of items, it detects 1 item at a time and the setup can be used only with a particular item, for example Book in our case.

Smart Shelf is very useful to maintain the inventory of the grocery shops and other retailers. The smart shelf that was built by us is inexpensive considering that it’s a prototype.

Component	Cost
Antenna	1 Euro
Tag	1 Euro
Smart Shelf	5 Euros
Arduino	15 Euros
Total	22 Euros

The estimate shown above excludes the cost of reader, because the same reader can be used to get the information from all the shelves with different product. Our prototype costs around 20 euros where the Arduino takes up 75% of the cost, but when mass produced the overall cost of such a setup should not be more than 8 to 10 Euros. The reader can cost up to 150 Euros. If a big retailer needs to setup such a Smart shelf system, it may cost around 1000 Euros for 100 shelves.

Improvement and Redesign

The accuracy of our setup mainly depends on the smart shelf's ability to identify the number of items on it .In order to improve the accuracy we need to use a weight sensor which is stable and more accurate. This can be achieved by replacing the Force sensitive resistors with Strain gauge load cell or piezoelectric load cell.

The OOK modulation gave an accuracy of 95%, this can be improved by using modulation techniques like FM modulation which can give a better accuracy

The redesigned shelf will have the following capabilities:

1. Better sensitivity for the items placed on it to i.e. detect items which weigh 50 grams.
2. Detect placing or removal of multiple items at a time.
3. Capability of calibrating it to any item just by entering how much one item weighs.

References

- Making Force resistive sensors: http://forums.parallax.com/discussion/108049/make-your-own-load-cell-force-sensor-from-stuff-you-already-have
- Python Code : …\python\ti_s6350_rfid\iso\s6350_iso_transponder_details.py