RFID BASED SMART TROLLEY IN SUPERMARKETS

The Mini Project Report

Submitted by

Ch.Udaya Raghava Sai	**-B140817EC**
M. Pavan Kumar	**-B140493EC**
R.M.Alagappan	**- B140490EC**
Venkatesh.U	**- B140709EC**

In partial fulfilment for the award of the Degree of the

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Under the Guidance of

Ameer P.M,
Assistant Professor,
ECED,NITC

DEPARTMENT OF
ELECTRONICS AND COMMUNICATION ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY CALICUT
KERALA, INDIA 673 601
APRIL 2017

CERTIFICATE

This is to certify that this project report entitled “RFID BASED SMART TROLLEY IN SUPERMARKETS” is a bona fide record of the mini project work done by Ch.Udaya Raghava Sai-B140817EC, M.Pavan Kumar-B140493EC, R.M.Alagappan-B140490EC, Venkatesh.U-B140709EC in partial fulfilment of the requirements for the award of the Degree of Bachelor of Technology in Electronics and Communication Engineering by the National Institute of Technology Calicut, India.

Ameer P.M,
Assistant Professor,
Department of ECE, NIT Calicut.

Sameer S.M,
Head of the Department of ECE,
NIT Calicut

Date: **April 20,2017**

OffIce Seal

ACKNOWLEDGEMENTS

First of all we would like to thank Abhilash sir for offering us this opportunity ,our guide Ameer P.M sir for motivating us throughout the project. we would like to express our gratitude towards Lyla B.Das madam and Jumshir sir for guiding us in synchronizing embedded systems networks and database management system which is necessary for our project.

ABSTRACT

In this project, we used Radio Frequency Identification(RFID) to implement Shopping trolley. Generally a lot of customer’s time is wasted while waiting in queue at the billing counter.Also the customer will not know the total cost after each item is added.

The long queues at the billing counter are because it takes time to scan each and every item by barcode reader.

Our project uses RFID based smart trolley to overcome these issues. The RFID reader scans all the products that were put in trolley and displays the cost of item and total cost on LCD display.

The customer can directly collect the bill at the billing counter without having to wait for much time

CONTENTS

1. Chapter 1: Introduction……………………………………………………………………….1
2. Chapter 2: Project Setup & Details…………………………………………………………2
3. Chapter 3 : Block and Circuit Diagram………………………………………………7
4. Chapter 4 : Implementation……………………………………………………………….9
5. Chapter 5 : Working Principle…………………………………………………………12
6. Chapter 6 : C codes………………………………………………………………………13
7. Chapter (last) : Conclusion……………………………………………………………26
8. References………………………………………………………………………………27

CHAPTER 1

Project Vision:

With the outburst of population living in cities and the advent of super markets, providing wide range of products to the customers, long queues are commonly seen at the billing counters and are expected to increase rapidly in future. This is motivating many prospective customers towards online shopping. To compete with e-shopping, it is necessary to modernise the current shopping system by reducing the queues at counters and improve customer satisfaction. Also through this project, we can reduce the number of billing counters, thus saving man power and equipment.

Introduction:

In this project we have designed a simple, cost-effective network infrastructure consisting of nodes(i.e each trolley acts as a node), communicating with the central database present at the billing counter, through central routers. The customer can know the cost of item added and total cost of items in the cart from the LCD display attached to the cart. He can also remove items from the cart.

We used RFID technology replacing the conventional barcode system for scanning the products. Each item is associated with a unique RFID tag which consists the tag number and the details of which are stored in the central database.

CHAPTER 2

PROJECT SETUP:

Hardware used:

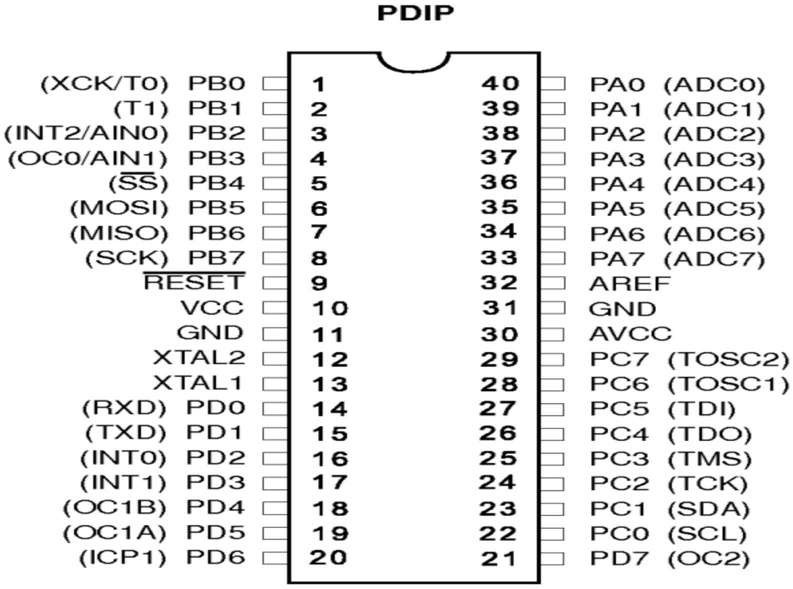
- ATmega 16
- ESP8266 Wifi Module
- EM 18 RFID Reader
- RFID tags
- 16x2 LCD display
- WiFi Router
- Crystal Oscillator
- USB Asp programmer
- Resistors,Capacitors

Software Used:

- WinAVR Programmers Notepad
- Arduino IDE
- XAMPP
    - ➢Apache Server
    - ➢MySQL
    - ➢phpMyAdmin
- Notepad++

Details:

1.Atmega 16



Pin Descriptions

The following ports of Atmega 16A have been used in the project.

VCC Digital supply voltage.

GND Ground.

Port A (PA7..PA0) Port A serves as the analog inputs to the A/D Converter.Port A also serves as an 8-bit bi-directional I/O port, if the A/D Converter is not used. Port pins can provide internal pull-up resistors (selected for each bit). The Port A output buffers have symmetrical drive characteristics with both high sink and source capability. When pins PA0 to PA7 are used as inputs and are externally pulled low, they will source current if the internal pull-up resistors are activated. The Port A pins are tri-stated when a reset condition becomes active,even if the clock is not running.

Port B (PB7..PB0) Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.Port B also serves the functions of various special features of the ATmega16.

Port C (PC7..PC0) Port C is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port C output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active,even if the clock is not running. If the JTAG interface is enabled, the pull-up resistors on pins PC5(TDI), PC3(TMS) and PC2(TCK) will be activated even if a reset occurs.Port C also serves the functions of the JTAG interface and other special features of the ATmega 16.

RESET Reset Input. A low level on this pin for longer than the minimum pulse length will generate a reset, even if the clock is not running. Shorter pulses are not guaranteed to generate a reset.

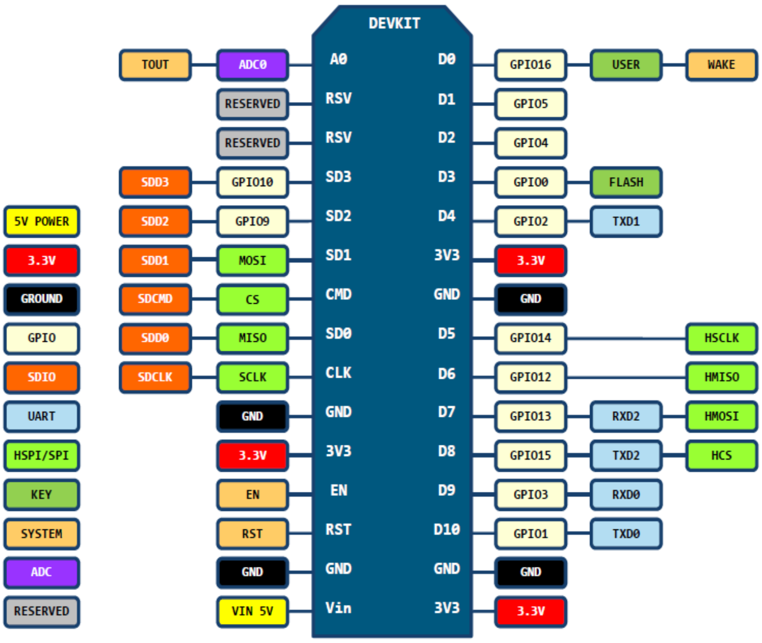
XTAL1 Input to the inverting Oscillator amplifier and input to the internal clock operating circuit.

XTAL2 Output from the inverting Oscillator amplifier.

AVCC AVCC is the supply voltage pin for Port A and the A/D Converter. It should be externally connected to VCC, even if the ADC is not used. If the ADC is used, it should be connected to VCC through a low-pass filter.

AREF AREF is the analog reference pin for the A/D Converter.

2. ESP8266 Wifi Module



It is based on widely explored esp8266 System on Chip from Expressif. It has features of WIFI accesspoint and station.

ESP8266 offers-

--Event-driven API for network applicaitons

--10 GPIOs D0-D10, PWM functionality, IIC and SPI communicaiton, 1-Wire and ADC A0 etc. all in one board

--Wifi networking (can be uses as access point and/or station, host a webserver), connect to internet to fetch or upload data.

This project uses the pins D1, D8, D5, GND of NodeMCU.

3. Radio Frequency Identification(RFID)

Radio frequency identification is a method of identifying,tracking or verifying objects/persons with the help of electronic tags/labels attached to them. These electronic tags are capable of receiving and transmitting radio frequencies and are called RFID tags or RFID labels. The ID information from these tags is obtained using RFID readers.

Each RFID tag has a transmitter and receiver called a transponder because it ‘transmits and responds’.The RFID reader transmits a signal to interrogate the tag.The tag receives the signal and responds by transmitting its identity information.The system then displays this information on LCD display.

‣ RFID Tag/Label

RFID tags belong to a class of radio devices known as ‘transponders’.A transponder is a combination of transmitter and receiver which is designed to receive a specific radio signal and automatically transmit a reply.There are two types of tags:active and passive. Active tags need battery for power,while passive tags don’t.They make use of the power of the signal transmitted by the RFID reader.This prevents unnecessary power wastage during idle hours.A passive RFID tag consists of
(a) Encoding/Decoding circuitry
(b) Memory
(c) Antenna
(d) Communication control

RFID tags have risen as a major complement to the conventionally used barcode system.RFID provides many advantages over the barcode such as:

- RFID can be used to read multiple tags at a time where as barcodes can read only one item at a time.
- RFID tags can be read even if there is no line of sight.Barcode requires the code to be scanned in the line of sight.

‣ RFID Reader

RFID reader is the interrogator. The reader is also a transreceiver,that is,transmitter plus receiver.Handheld devices have reader and antenna together as one unit, while larger systems usually separate antennas from the reader.

A reader usually contains a system interface such as RS-232 serial port or Ethernet jack, Crytographic encoding or decoding circuitry, a power supply or a battery or communications control unit.

The circuit uses EM18 RFID reader.

4. LCD Display

LCD (Liquid Crystal Display) screen is an electronic display module. LCDs are economical, easily programmable. A 16x2 LCD means it can display 16 characters per line and there are 2 such lines. The command register stores the command instructions given to the LCD. A command is an instruction given to LCD to do a predefined task like initializing it, clearing its screen, setting the cursor position, controlling display etc. The data register stores the data to be displayed on the LCD. The data is the ASCII value of the character to be displayed on the LCD.

5.Softwares

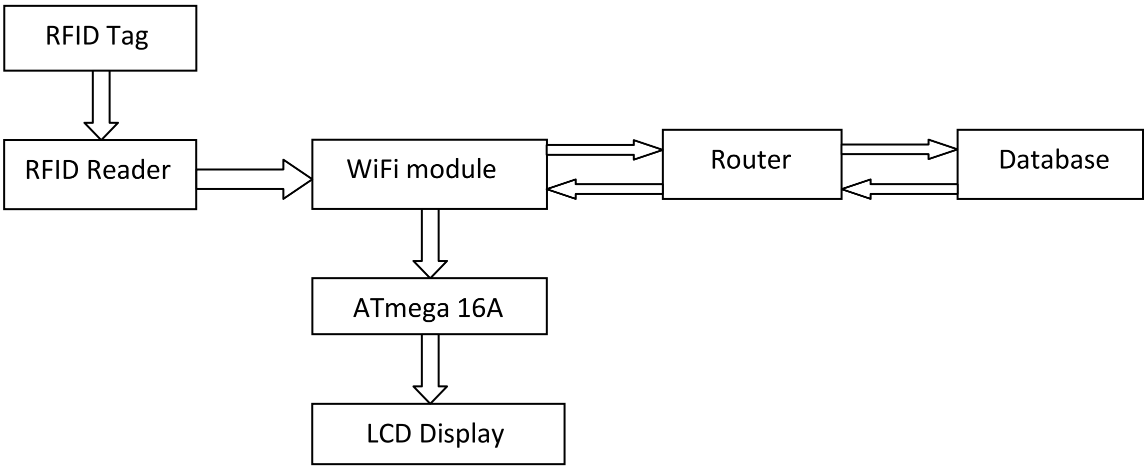
Programmers Notepad (PN) is a editing tool with syntax highlighting. As well it can be configured so that much command line typing need not be done, which makes it easier to work with.It is used to program the ATmega16A microcontroller.

The Arduino Integrated Development Environment or Arduino Software (IDE) contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus.It is used to program ESP8266 nodemcu.

XAMPP is a free and open source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages. XAMPP stands for Cross-Platform (X), Apache (A), MariaDB (M), PHP (P) and Perl (P). It is a simple, lightweight Apache distribution that makes it extremely easy for developers to create a local web server for testing and deployment purposes. Everything needed to set up a web server – server application (Apache), database (MariaDB), and scripting language (PHP) – is included in an extractable file.

CHAPTER 3

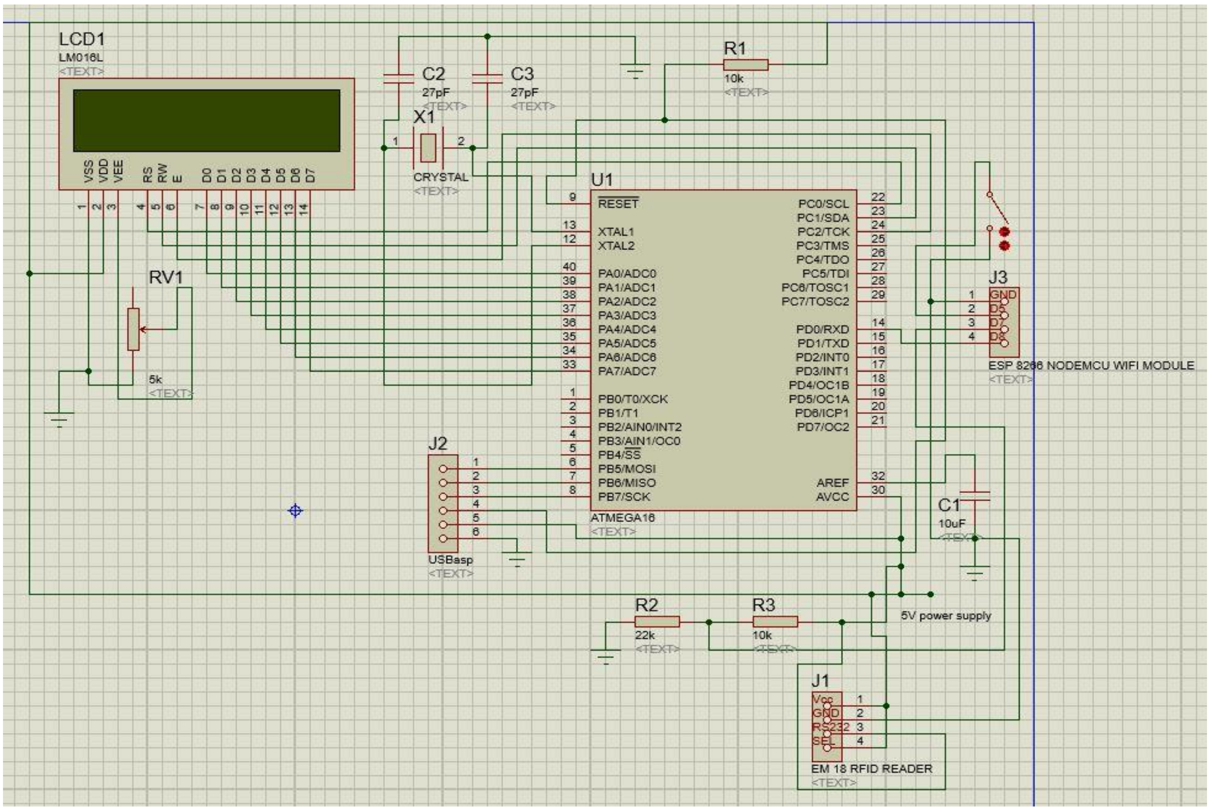
Block Diagram



Flow of operation

The RFID reader scans the tag and reads the tag number.This tag number is then sent to the wifi module . The wifi module which is connected to router sends this tag number to the Database through wifi.From the database the required details such as item cost,cumulative cost corresponding to the tag number are received by the wifi module.It sends this data to ATmega16 microcontroller.LCD displays this data, by receiving it from ATmega 16.

Circuit Diagram



CHAPTER 4

Implementation:

WiFi Module:

The D7,D8 pins of WiFi module are configured as receiver and transmitter respectively and are used for serial communication. The D5 pin is used as a switch,which when connected to Vcc,the scanned item will be added to the cart and which when conncted to ground, the scanned item will be removed from the cart.

The WiFi module is connected to the WiFi Router.

ATmega 16A:

This microcontroller is used for sending the data received from wifi module to the LCD display.

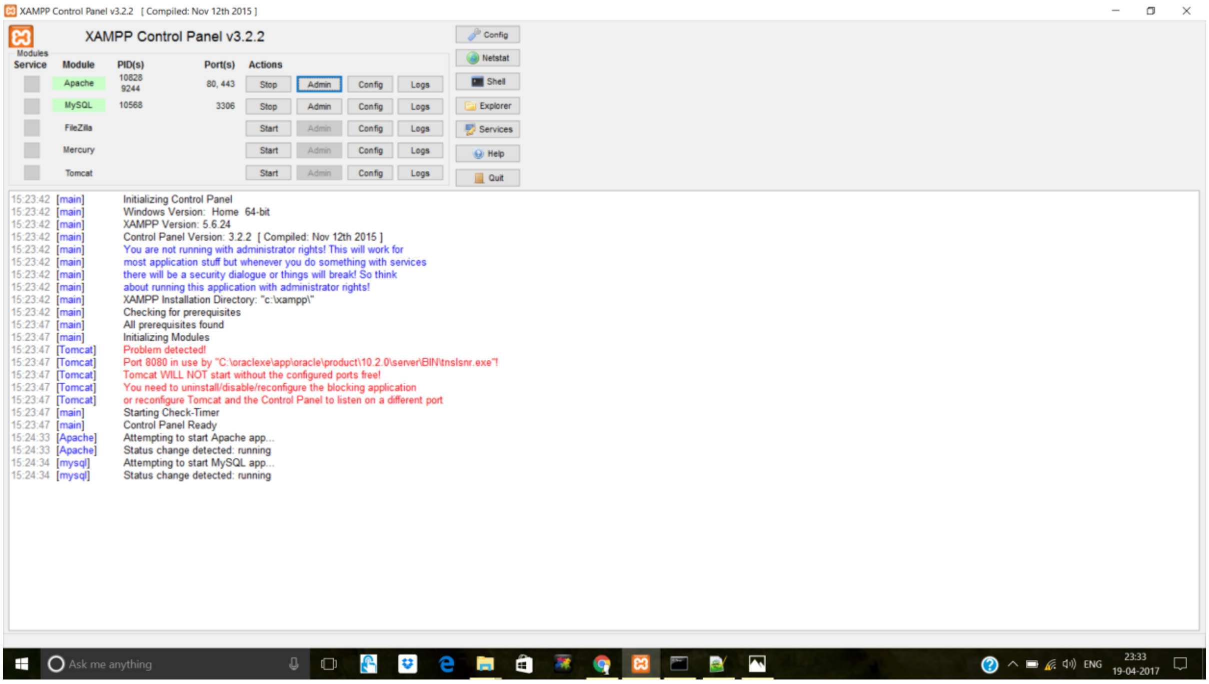
All the Port A pins and PC0,PC1,PC2 pins of Port C are used to interface ATmega 16A with LCD display. A 4MHz crystal oscillator is connected between 12 and 13 pins of ATmega 16. The USB Asp programmer is used to program the microcontroller.

LCD Display:

It is used to display the name and cost of scanned item and also the cumulative cost of items present in the cart.

The pins RS,RW,E along with D0 to D7 pins are used to interface it with ATmega 16

Database:

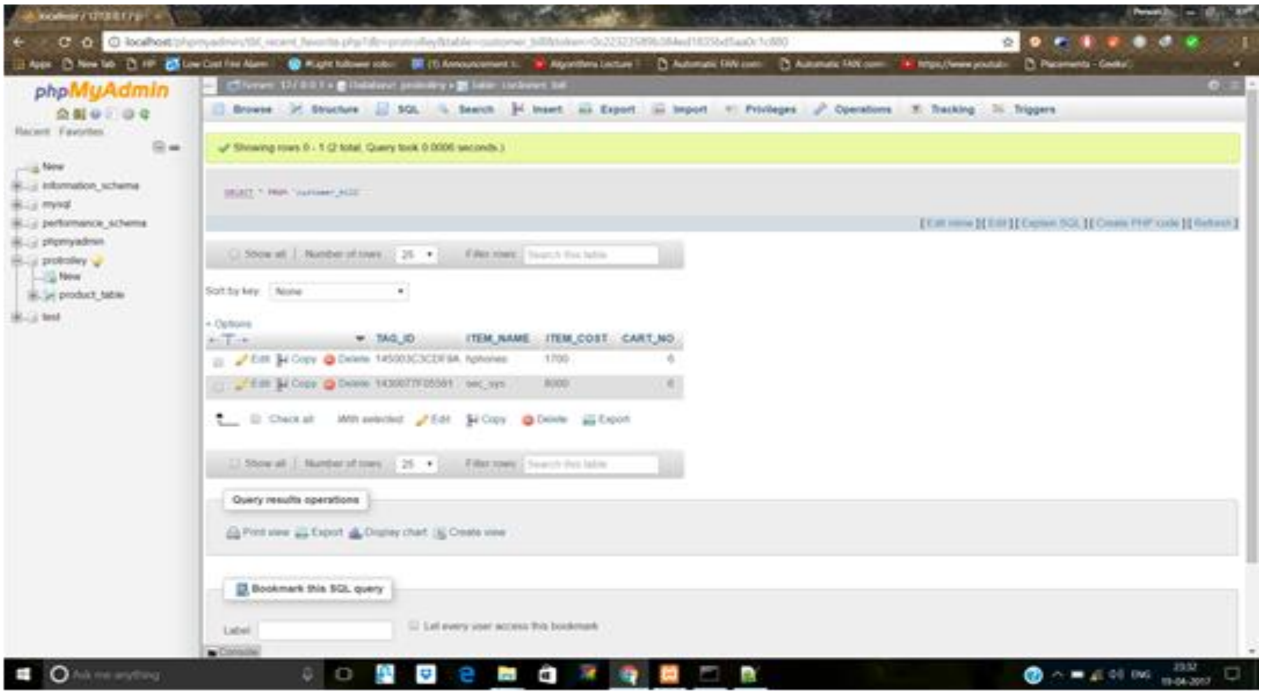

XAMPP Control Panel v3.2.2 [Compiled: Nov 12th 2015]

XAMPP Control Panel v3.2.2

Service	Module	PID(s)	Port(s)	Actions			
□	Apache	10828 9244	80, 443	Stop	Admin	Config	Logs
□	MySQL	10568	3306	Stop	Admin	Config	Logs
□	FileZilla			Start	Admin	Config	Logs
□	Mercury			Start	Admin	Config	Logs
□	Tomcat			Start	Admin	Config	Logs

```
15:23:42  [main]     Initializing Control Panel
15:23:42  [main]     Windows Version:  Home  64-bit
15:23:42  [main]     XAMPP Version: 5.6.24
15:23:42  [main]     Control Panel Version: 3.2.2  [ Compiled: Nov 12th 2015 ]
15:23:42  [main]     You are not running with administrator rights! This will work for
15:23:42  [main]     most application stuff but whenever you do something with services
15:23:42  [main]     there will be a security dialogue or things will break! So think
15:23:42  [main]     about running this application with administrator rights!
15:23:42  [main]     XAMPP Installation Directory: "c:\xampp\"
15:23:42  [main]     Checking for prerequisites
15:23:47  [main]     All prerequisites found
15:23:47  [main]     Initializing Modules
15:23:47  [Tomcat]   Problem detected!
15:23:47  [Tomcat]   Port 8080 in use by "C:\oraclexe\app\oracle\product\10.2.0\server\BIN\tnslsnr.exe"!
15:23:47  [Tomcat]   Tomcat WILL NOT start without the configured ports free!
15:23:47  [Tomcat]   You need to uninstall/disable/reconfigure the blocking application
15:23:47  [Tomcat]   or reconfigure Tomcat and the Control Panel to listen on a different port
15:23:47  [main]     Starting Check-Timer
15:23:47  [main]     Control Panel Ready
15:24:33  [Apache]   Attempting to start Apache app...
15:24:33  [Apache]   Status change detected: running
15:24:34  [mysql]    Attempting to start MySQL app...
15:24:34  [mysql]    Status change detected: running
```

The database is created using Xampp software .It consists of phpmyadmin which is a database management system.phpmyadmin is compatible with php language and MySQL language. Two tables ‘customer bill’ and ‘product table’ are created in the database named ‘protrolley’. Each item in the shopping market will have an RFID tag. The ‘product table’ consists of details of all the items present in the shopping market. It consists of details like item name, item cost corresponding to each and every tag(i.e for each an every item).

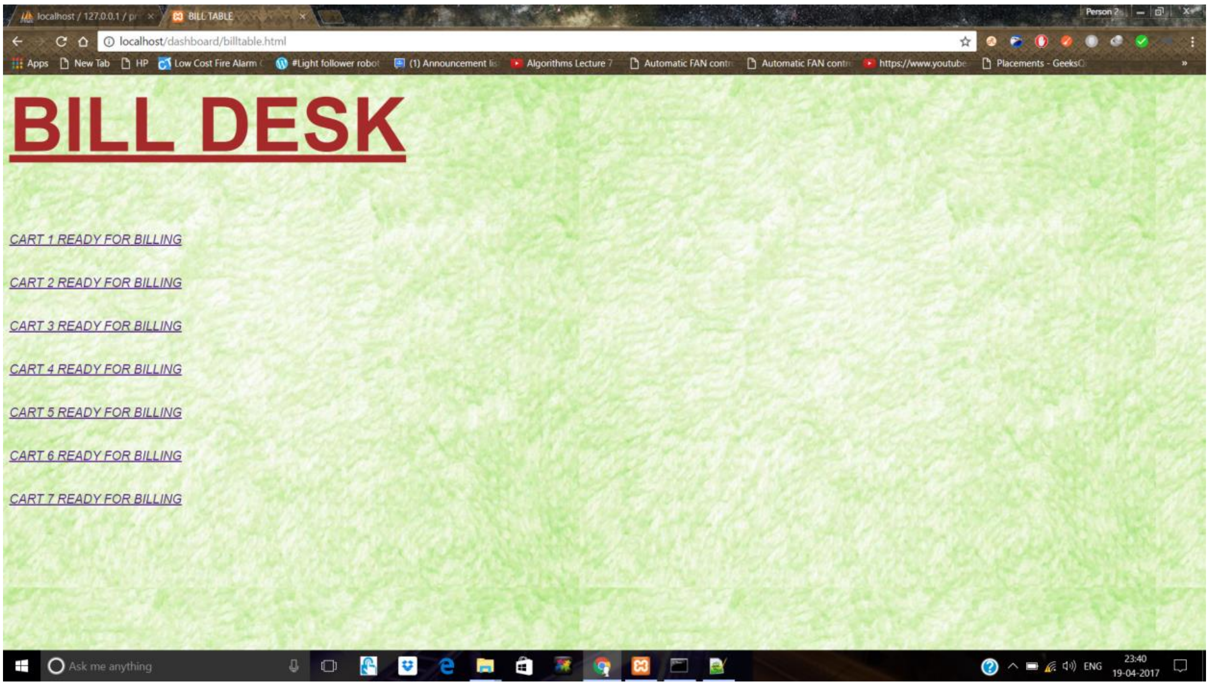

✓ Showing rows 0 - 1 (2 total, Query took 0.0006 seconds.)

```
SELECT * FROM `customer_bill`
```

TAG_ID	ITEM_NAME	ITEM_COST	CART_NO
14500C3C3CDF8A	hphones	1700	6
1430077F05581	sec_sys	8000	6

The ’customer bill’ table corresponds to items added in various carts. If an item is added to a particular cart ,then the item name, item cost and the cart number in which it is placed will be stored in ‘customer table’.The tables have operations to Edit,Copy and Delete the contents present in the table

php codes are written for accessing and updating data in the tables remotely through urls.The php codes written are bridge.php and bill.php.



BILL DESK

CART 1 READY FOR BILLING

CART 2 READY FOR BILLING

CART 3 READY FOR BILLING

CART 4 READY FOR BILLING

CART 5 READY FOR BILLING

CART 6 READY FOR BILLING

CART 7 READY FOR BILLING

A html webpage billtable.html is created with the help of a php code which helps the shopkeeper to generate bill corresponding to every cart present in the shopping market.

WiFi Router:

It is used to setup a local network for transferring data between Database and WiFi module.
The Router need not be connected to internet for this data transfer.

CHAPTER 5

Working principle:

After the trolley is set up with the circuit, it will be ready for use by the customers who can add and remove products from the trolley. When a product is added to the trolley,the EM18 RFID reader present in the circuit scans the RFID tag attached to the product and reads the the tag number of the product.The tag number is received at the RS232 serial port of the RFID reader. The tag number is then sent to database,with the help of this code snippet.

```
String url = "/dashboard/bridge.php?cartno=6&option=append&tagid=";
String frl=url+phrase;
```

option=append is for adding an item to cart(no.6 here) and similarly option=omit is for removing an item from the cart. Now the tag id, cart number and option will be sent as an argument to bridge.php, which obtains the details about the tag id i.e the item name,item cost from the product table and updates the customer bill table with these values along with the cart number in which the item is placed. Then the item name, its cost and the cumulative cost of the items corresponding to the cart are received by the WiFi module, which is sent to LCD display through ATmega 16A to display these details.

The same procedure takes place for each item that is added to the cart. Similarly an item can be removed from the cart (by connecting D5 pin of wifi module to ground). When an item is removed from the cart, it will also be removed from the customer bill table.The customer bill table contains all the items that are currently present in all the carts of the shopping market.

When the customer finishes placing all the required items in the cart, he goes to billing counter. At the billing counter, the shopkeeper will be ready with billtable.html webpage which consists of hyperlinks for each cart. When clicked, the hyperlink invokes bill.php, which filters the items of a particular cart from the customer bill table and generates the bill with the items of that cart, which can be printed.

Data can be cleared after billing is done so that the trolley can be reused again.

CHAPTER 6

C codes

Arduino code for WiFi module:

```
#include <ESP8266WiFi.h>
#include <SoftwareSerial.h>
SoftwareSerial mySerial(13,15);
const char* ssid  = "room 208";
const char* password = "123456789";
const char* host = "192.168.0.2";
const char* passcode = "uday";
String phrase;
char s;

#define SW1 14

int switch1;

WiFiClient client;

void setup() {
mySerial.begin(9600); // Setting the baud rate of Software Serial Library
Serial.begin(9600); //Setting the baud rate of Serial Monitor
delay(10);
Serial.println();
Serial.println();
Serial.print("Connecting to ");
Serial.println(ssid);
WiFi.begin(ssid, password); // Connecting to Wifi
while (WiFi.status() != WL_CONNECTED) {
delay(500);
Serial.print(".");

  }

 Serial.println("");
Serial.println("WiFi connected");
Serial.println("IP address: ");
Serial.println(WiFi.localIP());
delay(5000);

  // Connect to host
  Serial.print("Connecting to ");
Serial.println(host);

  // Use WiFiClient class to create TCP connections

 const int httpPort = 80;
if (!client.connect(host, httpPort)) {
Serial.println("Connection failed!");

}

}

void loop(){
//for scanning the rfid tag
if(mySerial.available()>0)

 { phrase = '1';

    for(int i=0;i<20000;i++)

    {

    if(mySerial.available()>0)

    {

     s = mySerial.read();

     phrase = String(phrase + s);

    }

    }

  pinMode(SW1, INPUT);

  // Read switch value

  switch1 = digitalRead(SW1);

  // Create a URL for the request. Modify YOUR_HOST_DIRECTORY so that you're pointing to the PHP
file.

  if(switch1 == HIGH )

  {

  String url = "/dashboard/bridge.php?cartno=6&option=append&tagid=";

  String frl=url+phrase;
  // This will send the request to the server
  Serial.print("Requesting URL: ");
  Serial.println(frl);
  client.print(String("GET ") + frl +  " HTTP/1.1\r\n" +
               "Host: " + host + "\r\n" +
               "Connection: close\r\n\r\n");
 unsigned long timeout = millis();
  while (client.available() == 0) {
     if (millis() - timeout > 50) {
     client.stop();
     client.connect(host,80);
     return;
    }
   }
    Serial.println("Server echo:");
     // This will receive data from server
    while(client.available())
    {
    char c = client.read();
    mySerial.write(c);
    Serial.write(c);
    }
  }
  else
  {
    String url = "/dashboard/bridge.php?cartno=6&option=omit&tagid=";
    String frl=url+phrase;
  // This will send the request to the server
  Serial.print("Requesting URL: ");
  Serial.println(frl);
  client.print(String("GET ") + frl + " HTTP/1.1\r\n" +
               "Host: " + host + "\r\n" +
               "Connection: close\r\n\r\n");
  unsigned long timeout = millis();
  while (client.available() == 0) {
    if (millis() - timeout > 50) {
      client.stop();
      client.connect(host,80);
      return;
  }
  }
   Serial.println("Server echo:");
   // This will receive data from server
  while(client.available())
  {
    char c = client.read();
    mySerial.write(c);
    Serial.write(c);
  }
  }
  Serial.println();
  Serial.println("Closing connection");
  }
}
```

AVR code for ATmega 16A:

```
#include<avr/io.h>

/*Includes io.h header file where all the Input/Output Registers and its Bits are defined for all AVR
microcontrollers*/


#define F_CPU   4000000

/*Defines a macro for the delay.h header file. F_CPU is the microcontroller frequency value for the
delay.h header file. Default value of F_CPU in delay.h header file is 1000000(1MHz)*/


#include<util/delay.h>

/*Includes delay.h header file which defines two functions, _delay_ms (millisecond delay) and
_delay_us (microsecond delay)*/


#define         LCD_DATA_PORT           PORTA

/*Defines a macro for the lcd.h header File. LCD_DATA_PORT is the microcontroller PORT Register to
which the data pins of the LCD are connected. Default PORT Register for data pins in lcd.h header file
is PORTA*/


#define         LCD_CONT_PORT           PORTC

/*Defines a macro for the lcd.h header File. LCD_CONT_PORT is the microcontroller PORT Register to
which the control pins of the LCD are connected. Default PORT Register for control pins in lcd.h
header file is PORTB*/


#define         LCD_RS          PC0

/*Defines a macro for the lcd.h header file. LCD_RS is the microcontroller Port pin to which the RS
pin of the LCD is connected. Default Port pin for RS pin in lcd.h header file is PB0*/


#define         LCD_RW          PC1
```

```
/*Defines a macro for the lcd.h header file. LCD_RW is the microcontroller Port pin to which the RW
pin of the LCD is connected. Default Port pin for RW pin in lcd.h header file is PB1*/


#define         LCD_EN                  PC2
/*Defines a macro for the lcd.h header file. LCD_EN is the microcontroller Port pin to which the EN
pin of the LCD is connected. Default Port pin for EN pin in lcd.h header file is PB2*/


#include<avr/lcd.h>
/*Includes lcd.h header file which defines different functions for all Alphanumeric LCD(8-Bit
Interfacing Method). LCD header file version is 1.1*/


#include<avr/usart.h>
/*Includes usart.h header file which defines different functions for USART. USART header file version
is 1.1*/


int main(void)
{
        DDRA=0xff;
        /*All 8 pins of PortA are declared output (data pins of LCD are connected)*/
        DDRC=0x07;
        /*PC0, PC1 and PC2 pins of PortC are declared output (control pins of LCD are connected)*/
         char rfid_data[250];
        /*Variable declaration*/
        int a;
        usart_init();
        /*USART initialization*/
        lcd_init();
        /*LCD initialization*/
```

```
lcd_string_write("Nitc");

/*String display in 1st row of LCD*/

 lcd_command_write(0xc0);

/*Cursor moves to 2nd row 1st column of LCD*/

 lcd_string_write("project");

/*String display in 2nd row of LCD*/


_delay_ms(1000);

_delay_ms(1000);

_delay_ms(1000);

_delay_ms(1000);

/*Display stays for 4 seconds*/


/*Start of infinite loop*/

while(1)

{

        lcd_command_write(0x01);

        /*Clear screen*/

        lcd_string_write(" Show RFID ");

        /*String display in 1st row of LCD*/


        for(unsigned char i=0;i<250;i++)

        {

                rfid_data[i]=usart_data_receive();

                /*Receives data from wifi module through USART and stores it in the array*/

                if(rfid_data[i] == '*')

                {
```

```
                              a = i;
          } }
                     lcd_command_write(0x01);
                     /*Clear screen*/
                     for(unsigned char i=218;i<a;i++)
                     {
                        lcd_data_write(rfid_data[i]);
                     }
                      lcd_command_write(0xC0);
                     /*Cursor moves to 2nd row 1st column*/
                    for(unsigned char i=a+1;i<250;i++)
                     {
                               lcd_data_write(rfid_data[i]);
                               /*Total cost is displayed in 2nd row of LCD*/
                     }


                     _delay_ms(1000);
                     _delay_ms(1000);
                     _delay_ms(1000);
                     _delay_ms(1000);
                     _delay_ms(1000);
                     _delay_ms(1000);


                     /*Display stays for 6 seconds*/
           }
     }
     /*End of program*/
```

Bridge.php

```
<?php
       $dbhost='localhost';
                 $username='root';
                 $password='uday';
                 $db='protrolley';
                 $cumu_cost=0;
                $conn = mysqli_connect("$dbhost","$username","$password","$db");//connecting to
database
               if(!$conn)
                              echo "error";


                $tag=$_GET['tagid'];
                 $cart=$_GET['cartno'];
                $opt=$_GET['option'];
                $req="select * from product_table where TAG_ID='$tag'";//extracting details of tag from
product table
                $res=mysqli_query($conn,$req);
                if(!$res)
                echo "not extracted successfully";
     while($exe=$res->fetch_assoc())
                 {
   $name= $exe['ITEM_NAME'];
   $cost= $exe['ITEM_COST'];
                 }
                  echo '#'."$name".' ';
                   echo "$cost".'*';
```

```
                        if($opt=='append')

                        {$rev="insert into customer_bill(TAG_ID,ITEM_NAME,ITEM_COST,CART_NO) values
('$tag','$name','$cost','$cart')";//inserting item into customer_bill table

                         $ref=mysqli_query($conn,$rev);

                         if(!$ref)

              echo "NIS";}

  if($opt=='omit')

                        {$rev="delete from customer_bill where TAG_ID='$tag'";//deleting  item  from
customer_bill table

                         $ref=mysqli_query($conn,$rev);

                         if(!$ref)

                        echo "not deleted successfully";}

                         $rej="select * from customer_bill where CART_NO='$cart'";

              $reh=mysqli_query($conn,$rej);

              if(!$rej)

               echo " not extracted successfully";

    while($exf=$reh->fetch_assoc())

               {

       $cumu_cost=$cumu_cost+$exf['ITEM_COST'];

               }

               echo "T.cost=$cumu_cost".'<br>'.'@';

               ?>
```

Bill.php

```
<html>

                         <body>

<?php

      $dbhost='localhost';
```

```
                $username='root';

                $password='uday';

                $db='protrolley';

               $i=0;

               $total_cost=0;

                $conn = mysqli_connect("$dbhost","$username","$password","$db");//connecting  to
      database

              if(!$conn)

                     echo "error";

                $cart=$_GET['cartno'];

               echo "CART NUMBER =$cart".'<br>';

               $req="select * from customer_bill where CART_NO='$cart'";

               $res=mysqli_query($conn,$req);

               if(!$res)

                echo " not extracted successfully";

  while($exe=$res->fetch_assoc())

                {  $i=$i+1;

    $total_cost=$total_cost+$exe['ITEM_COST'];//billing by  adding all  the  costs

  echo $i.')'.' ';

 echo $exe['ITEM_NAME'].'-'.'?'.' ';

 echo $exe['ITEM_COST'].'<br>';

                }

                echo "total cost = ?$total_cost".'<br>';

                 $rev="delete from customer_bill where CART_NO='$cart'";//Emptying cart  after  billing

                    $ref=mysqli_query($conn,$rev);

                    if(!$ref)

                         echo "cart not  emptied after billing";
```

```
                        ?>
                        <form>
<input type="button" onclick="window.print()" value="print bill"></form>//printing the bill
</body>
            </html>
```

Bildesk.html

```
<html>
<head>
<title> BILL TABLE</title>
  <font color="brown" size="7" face="arial">
<u><b><h1>BILL DESK</h1></b></u>
</font>
<br>
</head>
<body background="pic.jpg">
<font face="arial" color="red">
<i>
<a href="http://localhost/dashboard/bill.php?cartno=1">CART 1 READY FOR BILLING</a><br><br><br>
<a href="http://localhost/dashboard/bill.php?cartno=2">CART 2 READY FOR BILLING</a><br><br><br>
<a href="http://localhost/dashboard/bill.php?cartno=3">CART 3 READY FOR BILLING</a><br><br><br>
<a href="http://localhost/dashboard/bill.php?cartno=4">CART 4 READY FOR BILLING</a><br><br><br>
<a href="http://localhost/dashboard/bill.php?cartno=5">CART 5 READY FOR BILLING</a><br><br><br>
```

```
<a href="http://localhost/dashboard/bill.php?cartno=6">CART 6 READY FOR
BILLING</a><br><br><br>

<a href="http://localhost/dashboard/bill.php?cartno=7">CART 7 READY FOR
BILLING</a><br><br><br>

</i></font>

</body>

</h
```

CHAPTER (last)

CONCLUSION

The objectives required have been achieved in this project. The product obtained is easy to use and cost effective.No special training is required to use the product. There are few aspects where the project needs to be improved for it to be implemented in a supermarket.The project needs to be fabricated on PCB for better connectivity.Also the security of the communication must be increased .

REFERENCES:

[1] Lyla B. Das, Embedded Systems : An Integrated Approach, 1st Edition, *Pearson,*2013.

[2] Aboli Hanwante, Poonam Thakare, Smart Trolley using RFID*, International Journal of Research in Science and Engineering Volume 1 Special Issue 1*, 2012.

[3] Zeeshan Ali, Reena Sonkusare, RFID based Smart Shopping and Billing*, International Journal of Research in Science and Engineering , Volume 2, Issue 12*, 2013.

[4] www.rfidjournal.com