جامعة الأمير محمد بن فهد بن عبد العزيز
PRINCE MOHAMMAD BIN FAHD UNIVERSITY

College of Engineering
Department of Electrical Engineering
Fall 2012-2013

Learning Outcome Assessment III/EE
EEEN 4311/Section 106

Course Instructor
Dr. Samir Elnakla

# Final Product Report
# Parking Monitoring Control System

Group:


| | |
|---|---|
| Ali Al Misbah | #200600302 |
| Muneer Al-Hashim | #200600086 |
| Salman Al-Belaihi | #200801405 |
| Mustafa Al-Qurain | #200800565 |

# Abstract

Not finding a parking space for you sometimes is indeed a critical issue. An important appointment might be missed. Time will be wasted and more money will be spent. Criticism and inconvenience are more likely to happen. So, parking monitoring is an important solution. It has many practical advantages. It can arrange the passage of vehicles inside the parking area in addition to avoiding contention. It can reduce the suffering of sick and disabled people looking for a parking space. It is a good denote to such part of the society. It is like providing them with the smile and hope that helps them to overcome their disability. This project can provide information about the future need to make more parking spaces, also has many uses such as security issues and data base.The project consists of four mainsubsystems, the first subsystem is designing the power supply, in addition, the sensor subsystem circuit which includes RFID and IR sensor.The third subsystem is the LED matrix to display the free parking lots, authorized, and unauthorized personnel. Moreover, the received signal by RFID will be processed using Arduino controller subsystem. Furthermore, the output of the controller will be processed by PC to handle data base using C# software.

# 1 Table of Contents

# Chapter 1: Introduction & Product Development

## 1.1Introduction

### 1.1.1Project Definition

Design a project of parking monitoring and control system to count the number of automobiles entering and leaving a parking, open the gate for authorized personnel, provide information about free parking spaces, create data base to provide statistics about people entering and leaving the parking space in addition to using this system for security issues such as preventing violating people to enter the parking space. This system uses efficient sensors and display circuits to withstand the surrounding environment.The system will be powered form power utility available to supply the system components with required supply for sensors, display, Arduino, door motors, and data base computer.

### 1.1.2Project Objectives

- Design an efficient parking system to provide statistical information and enhance the security of utility.
- Saving people's time in finding parking lots.
- Learn about energy impact on environment and our role as engineers to offer solutions;saving energy of automobiles which in turns reduces $CO_2$ emissions and other harmful gases; hence better green environment.

### 1.1.3Project Specifications

Main marketing features or specifications are:

- Low power consumption.
- High quality sensor and long life LED.
- Very clear display and efficient controller.
- Easy to install and maintain.
- Easily designed system to minimize the cost.
- The operating temperature of the whole system shall not exceed 70 degrees Celsius.

### 1.1.4Product Flowchart and Connection

At the entry (see figure 1.1)

If the person is authorized to enter the parking space, the RFID reader will read the tag and send its data to the Arduino. Arduino will check if this ID is authorized or not. If no, The LED display will show not Authorized. If yes the gate will open and close after the car enters the parking. At the same time, the counter will decrement and the LED display will show the number of free parking spaces. Also, Arduino will send the information to C# data base. In case of visitors, there will be pushbutton to open the gate, make decrement for the display, and close the gate after the car gets in.

At the exit (see figure 1.1)

The RFID reader will read the tag and send its data to the Arduino. The gate will open and close after the car leaves the parking. At the same time, the counter will increment and the LED display will show the number of free parking spaces. Also, Arduino will send the information to C# data base. In case of visitors, there will be pushbutton to open the gate, make increment for the display, and close the gate after the car gets out.

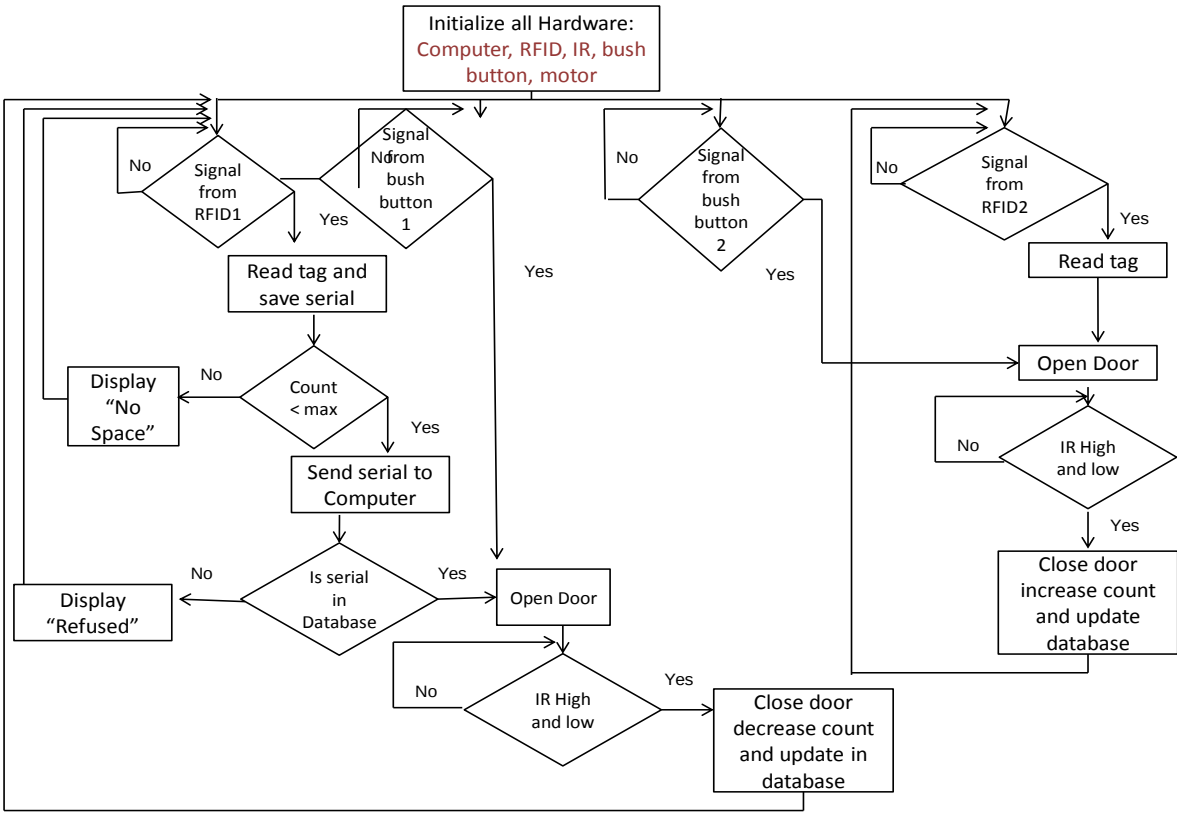



**Figure-1.1: Flowchart**

**At the entry:** (see figure 1.2)

Pin 1, 7 are connected to the ground at the Arduino

Pin 2, 11 are connected to the +5V DC at the Arduino

Pin 9 is the serial communication connected to the configured RX3 at the Arduino

**At the exit:** (see figure 1.2**)**

Pin 1, 7 are connected to the ground at the Arduino

Pin 2, 11 are connected to the +5V DC at the Arduino

Pin 9 is the serial communication connected to the configured RX2 at the Arduino

## Connection of LED matrix with Arduino (see figure 1.2)

VCC (red) to the +5Vat Arduino Board

Ground (black) to ground at the Arduino

DATA(orange) to the digital pin 2 (As configured in the program)

WRITE (yellow) to the digital pin 3 (As configured in the program)

CS0 (white) to the digital pin 4 (As configured in the program

External power supplycircuit toexternal power supply socket

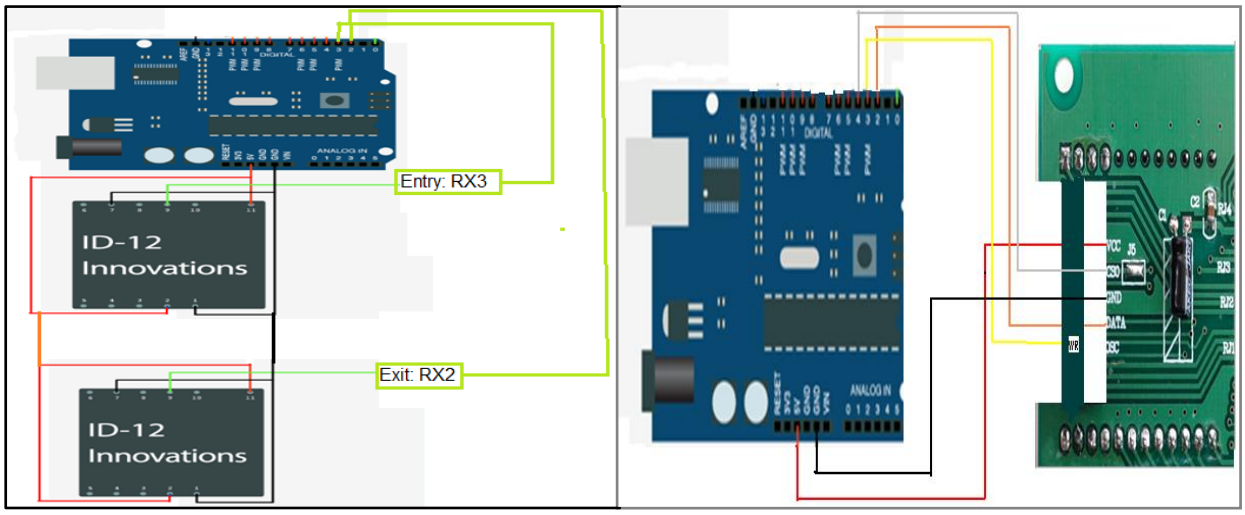



**Figure-1.2: circuit Connection**

### 1.1.5Product Prototype


**Figure-1.3: Product Prototype**

### 1.1.6Background or Problem

Making a parking monitoring system especially at Saudi Arabia will be significantly useful due to the increasing usage of parking lot as population and buildings increase more and more which indicates that there is really lack of parking and a large amount of traffic that look for free parking spaces. However, this would be time consuming for people to keep searching and

also there is an environmental issue since the automobiles is emitting CO2 which is a major factor in affecting environment. In order to deal with the problems of the metropolitan areas, there is a need to develop advanced solutions for efficient parking space management especially at airports, malls, universities, and any large parking lots that are used heavily. There are a number of companies that provide parking management solutions such as Siemens, Nortech and Omnitech.a brief introduction about each is described below but Omnitech was focused on due to the available information they provided regarding used technology.

Siemens provides several parking management services including: intelligent parking guidance systems, convenient pay-and-display machines, and modern car park technology. It uses Sitraffic Guide: parking guidance system which uses dynamic message signs to provide information to the driver on the current occupancy levels of individual parking areas and guide people along the shortest route to final parking destination which will dramatically reduce traffic volumes and environmental pollution. (See figure 1.4)

**Figure-1.4: Dynamic message sign**

Nortech uses group of products that provides
methods of counting vehicles entering or leaving a car park or depot and keeping track of available spaces. it uses what they call it NorParc Level Counting and Guidance System which offers real-time on-screen information feedback, alarms and report generation. It can manage both small, single level car parks and large multi-storey car parks with signs showing available space on each level.

Omnitech uses modern parking management systems which consists of entrance and exit gate barrier units, parking guidance, car parking control and garage payment system, and parking counting systems in all various types which are all connected from manual to automatic pay stations and a car park management center. The main services that are relevant and related to this context is parking guidance system and parking counting system.One type of sensor is the OmnyPark OP-511 which is an ultrasonic parking sensor that is used for garage parking guidance system designed with an ultrasonic detector use as one of the main part of the parking guidance and management system solution. Parking sensor makes use of ultrasonic detecting the vehicle parking space to get message and transfer to Zone Control Unit or other equipment through the RS485 connection. One parking space needs one Ultrasonic Detector which is installed in the middle of the car park space.

**Specification of OmnyPark OP-511 Ultrasonic Parking Sensor(** See figure 1.5)

Operating Voltage:DC 12～24V

Operating Current: 10mA ） no indicator （, 28mA ） with indicator

Operating Temperature:-20＋～80℃

Detecting Distance: 0.1～3.5m

Detecting Area 3.5m: 0.4 sq.m

Size: 100X23mm

Weight: 113g

Communication: RS-485

Communication distance: ≤1000m

Serial port setting: 4800bps，N, 8, 1

**Figure-1.5: OmnyPark OP-511 Ultrasonic Parking Sensor**

Also, there are several techniques and technologies used for counting and displaying functions as shown in the figure 1.6.

Specification;

1. Electrical supply Voltage: 9 to 24V DC
2. Current Requirement: 250mA @ 12 volts
3. Physical Display: 20 x 7 LED matrix giving 4 character display (choice of yellow or red)
4. Viewing angle: 20o each side of centre
5. Housing: Black powder coated Zintec
6. Dimensions (mm): 380 x 205 x 120 (W x H x D)
7. Display adjustments: 4 brightness levels selectable via NCT200 module
8. Mounting: Options for mounting directly on a flat surface or by using brackets (included) to enable tilt adjustment for wall mounting and both tilt and pan adjustment for pole mounting (3” diameter)
9. Cable Termination: Plug-in screw terminal blocks
10. Operation Counting capacity: -999 to 9999
11. Inputs Count Bus: RS485 bus from an NCT200 counting module delivering up to 8 count values.

**Figure-1.6: Variable Messgae Sign VMS**

The count value that is displayed is controlled by the node setting on DIP switches.

However, our product would not follow the technique used as shown above which is installing one detector at each parking lot since this would require lots of wiring and conduits and these types are for parking lots with roofs. Our product will have same the flexibility to be working effectively in shielded parking, non-shielded regardless of the size of the parking since the idea is to install sensors in the entry and exit of each parking lot zone and count up and count down the display according to the automobiles entering and exiting the parking lot zone.

## 1.2: Product Development

### 1.2.1 Productplanning

This product, like any product shall be made, need a feasibility study to ensure that this product can be made in terms of the availability of components, cost, implementation, professional technical skills required, safety requirements and most importantly its need by the customers.

Our product's components were checked in the local market and found them available relatively not costly. However, there are some challenges that would play roles on the progressing of the product designing.One of them is that theteam was not sure about whether the team can go to the level of programming a microcontroller due to the lack of knowledge and experience of dealing with it in terms of theoretical or practical experience. Also, the controller circuit requires good skills and experience in programming and interfacing the microcontroller with the sensors, display, and data base computer which might impede the product development process and success**.**Another challenging factor is time management since all of team members are full of tasks other than project which would impede as well the progress of the product development.

### 1.2.2 Customer Needs and Product Specifications

The need for such a product is in fact interests several customers. So, contacting the target customers was really an essential step to study the needs and specifications required by customers. Therefore, this was done by collecting data from field through conducting surveys on the targeted parking to have a clear idea of what the customers need. Also, meeting or interviewing the targeted managers of the targeted location to emphasize the advantages of the product and how that product will increase their income by gaining more satisfaction from their customers.

A security officer of Prince Mohamed University was asked for the need of such monitoring system and how it would be beneficial to them. He said that it would be very beneficial since in the morning there are 3 security officers work in the parking area trying just to guide students and close the entrances of those filled parking spaces. In fact, they highly appreciate such a system to be implemented in Prince Mohamed University according to PMU Security officer. Also, ordinary people were asked about parking lots in malls and airports and similar responses we received. Another step is to analyze such data to know how it can be implemented cost effectively and efficiently in line with the features preferred by customers. Then, the gathered information were analyzed to define the product specifications that match needs and requirements taking into consideration any safety issues or standards.

According to customer needs, all what they have in common is to find free parking spaces easily without detailed and specific technology to be used. However, some prefer a display to show the number of free parking spaces in addition to an indication of a particular zone is full or not.

### 1.2.3 Concept Generation

Many ideas can be generated by considering the different ways and technologies to implement each subsystem. The below ideas were generated from all group members.

**Energy:** (1) normal grid, (2) solar (3) battery banks (4) multiple DC output supply to accommodate the different rating for devices used in the control system.
**Counting elements:** (1) ultrasonic sensor,(2) motion sensors, and (3) IR sensors (4) Optical sensors. (5) Magnetic sensor
**Processing:** : (1) controller, (2) data base computer.
**Communication and connection:** all units interconnected using wires and cable tray systems.
**Display and Action**: (1) indoor/outdoor LED display, (2) Mech. Closing bar (3) Servo motor (4) DC motor

All of these alternatives will result in different ideas for the design implementation so a careful selection is highly required to select the best idea for each subsystem or component in order to fit the main objective of the product design.

### 1.2.4 Concept Selection

Selecting the best alternative among several options is mainly dependent on the availability of components, cost, reliability, implementation, professional technical skills required, safety requirements and most importantly its need by the customers.

Therefore, studying the alternatives, in terms of all essential design steps in order to select the best one, requires right decision making. Then testing the selected option against several factors such as customer preferences and needs, national standards if it shall be recommended by government law, availability of material, estimated material costs, financial budget, feasibility as well as efficiency of which all will determine the final selection of the design.

### 1.2.5Features of the product:

1- One parking zone for employee and others (students or visitors). They will be provided with RFID tag to allow them passing from the gate.

2- All the customers' mobile numbers will be attached with the serial number of their tags to be used for send and receiving SMS from the system and send the violation to their mobile number with the number of days that will not be authorized to use the parking lot.

3- ArduinoUno and Mega 2560 instead of microcontroller because it is more stable and easier to deal with and it will be used as the brain to control our system by sending and receiving instructions from inputs and output devices as follows:

   a. Input devices:

      i. RFID reader ID-12 with built in antenna one in the entrance and one in the exit which will detect the unique serial number of the tag and send to Arduino to take the appropriate action (allow or denied).

      ii. IR distance sensor in the entrance and exit to make sure that the car has passed the gate and it will send a signal to allow closing the gate.

      iii. Push button to open and close the gates incase of RFID reader failure.

   b. Output devices:

      i. Open and close gate by using stepper motor one in the entrance and exit.

      ii. 16x24 Red LED Matrix Panel to display the number of available parks and it will display the word "Full" when no parking available.

c. Input and output device:

i. Laptop will be used to add new customer or delete unwanted one and it will be used to display the customer's information.

ii. Cellular Shield with SM5100B will be used to send the number of available park by SMS and it will receive an SMS to reserve park.

4- The system will have the ability to reject specific cars that has violation or security issue.

5- The system will use C # as logging Data for security in a database and showing:

a. Name, ID and Plate number.

b. Time in and time out.

6- Monthly statistics showing how many visitors and violations per month.

### 1.2.6Product Functional Diagram:

The block diagram shown in figure 1.7 identifies how the Arduino connected with others devices.

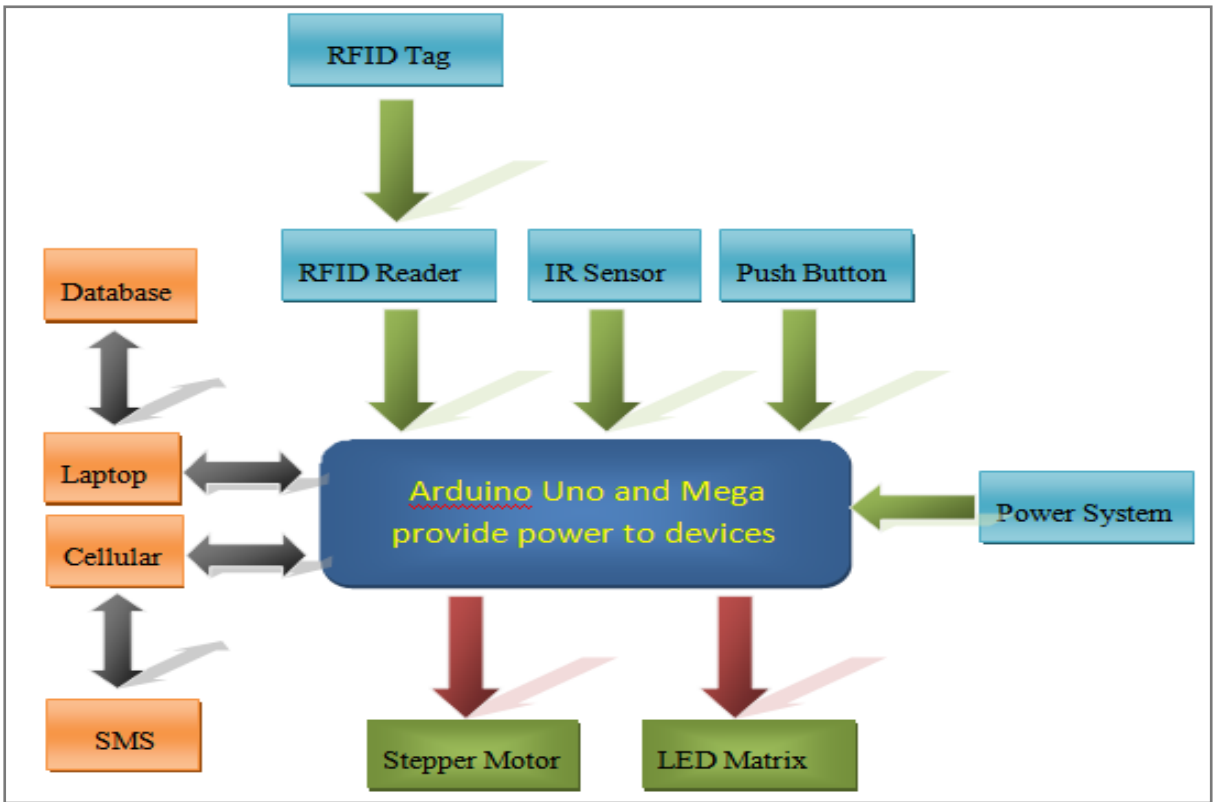



**Figure-1.7: Functional Diagram**

The following components were selected depending on cost and efficiency:-

**Power Supply:** 110/220 VAC to 9VDC to power Arduino and 5 VDC with supply current of 2A to power cellar shield.

**Sensors:**RFID sensor and IR sensor.

RFID reader: RFID reader type ID-12 with built in antenna.

RFID tag: Passive RFID tags type sparkfun.

IR Sensor: Sharp IR sensor typeGP2Y0A21YK0F

**Microcontroller:***Microchip Arduino mega 2560* that receives input from sensor subsystem and sends output to display subsystem and data base.

**Display:** 16*24 LED matrix type PID555

**Motors:**Servo motor type HS-5485HB

# Chapter 2:ArduinoController, RFID Reader, RFID Tag, and IR Sensor

## 2.1 Arduino Microcontroller

It's an open-source physical computing platform based on a simple microcontroller board, and a development environment for writing software for the board. Arduino can be used to develop interactive objects, taking inputs from a variety of switches or sensors, and controlling a variety of lights, motors, and other physical outputs.

### 2.1.1 Microcontroller Definition, Option, and Selection

A microcontroller (sometimes abbreviated µC, uC or MCU) is defined as a small computer on a single integrated circuit containing a processor core, memory, and programmable I/O peripherals. Program memory in the form of NOR flash or OTP ROM is also often included on chip, as well as a typically small amount of RAM Microcontrollers are designed for embedded applications.

### 2.1.2 Introduction about Microcontrollers

The controller is the core system of the project or any electrical circuit since it acts like the brain that takes decisions and receives and sends output accordingly. The controlling function in this project shall use the microcontroller rather than digital IC's. In fact, it is really a good chance to learn about microcontroller and how to program and design microcontroller based circuits since almost all controlling functions in electronic devices are made usingmicrocontrollers.Also, it gave flexibility in changing programs or modify or enhance the functionality easily since all it requires changing the program and downloading it to chip

whereas in digital IC's we have to change the physical hardware configuration to meet the required result which will be time consuming in addition to complexity.

However, there are several types of microcontrollers in the market. The selection of the microcontroller was based in how common it is in market and availability. Also, it depended on the size of the program memory, RAM and the clock frequency (speed), how many I/O it can accommodate as well as interrupts.

Basically, a microcontroller is a device which integrates a number of the components of a microprocessor system onto a single microchip and optimized to interact with the outside world through on-board interfaces. So, it is in fact a little gadget that houses a microprocessor, ROM (Read Only Memory), RAM (Random Access Memory), I/O (Input Output functions), and various other specialized circuits all in one package.

On the other hand, a microprocessor is normally optimized to co-ordinate the flow of information between separate memory and peripheral devices which are located outside it. Connections to a microprocessor include address, control and data busses that allow it to select one of its peripherals and send to or retrieve data from it. Because a microcontroller's processor and peripherals are built on the same silicon, the devices are self-contained and rarely have any bus structures extending outside their packages.

## Microcontrollers Options, and Selections

The common used microcontrollers which are suitable for our project are:

1. Arduino: An Arduino is a tiny computer that you can program to process inputs and outputs going to and from the chip. It is connected to PC using the USB cable. It is an open source electronics prototyping platform based on flexible, easy to use hardware and software. It's intended for artists, designers, hobbyist and engineers. Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motor and other actuators. The microcontroller on the board is programmed using the Arduino programming language.

2. PIC16F84A: It has a Watchdog Timer which can be shut off only through configuration bits. It runs off its own RC oscillator for added reliability. There are two timers that offer necessary delays on power-up. One is the Oscillator Start-up Timer (OST), intended to keep the chip in reset until the crystal oscillator is stable. The other

is the Power-up Timer (PWRT), which provides a fixed delay of 72 ms (nominal) on power-up only. This design keeps the device in reset while the power supply stabilizes. With these two timers on-chip, most applications need no external reset circuitry. SLEEP mode offers a very low current power-down mode. The user can wake-up from SLEEP through external reset, Watchdog Timer time-out or through an interrupt. Several oscillator options are provided to allow the part to fit the application. The RC oscillator option saves system cost while the LP (Low Frequency Power)crystal option saves power. A set of configuration bits are used to select the various options.

The type selected is Arduino Mega 2560 which belongs to the most well-known company that produces friendly microcontrollers.New to the product line is the new Arduino Mega with increased memory and number of I/O pins.This was chosen because of its availability of resources such as tutorials and books.

## 2.1.3ArduinoProperties, Schematic Diagram, and Power

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560. It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button.

- **Specifications and Features of Arduino Mega 2560**

Some of the detailed technical specification is mentioned below:-

- Operating voltage of 5 V.
- 16 MHz clock speed.
- The number of digital I/O's pins is 54.
- The number of analog input pins is 16.
- 4 hardware serial ports.
- Flash Memory: 32 KB
- SRAM: 2 KB
- EPROM: 1 KB
- Clock Speed: 16 MHZ

What sets a microcontroller apart from other processors are special circuits to deal with the needs of real time applications. There is an important feature of Arduino which is Software Serial library that allows for serial communication on any of the Mega's digital pins. The Arduino 2560 has a host of such features intended to maximize system reliability, minimize cost through elimination of external components, provide power saving operating modes and offer code protection. These features as illustrated in figure 2.1 next pageare:

- Reset button
- Digital I/O
- Power pins
- Analog inputs
- Voltage regulator
- In-circuit serial programming
- FTDI USB chip and USB jack
- Power jack
- ICSP header

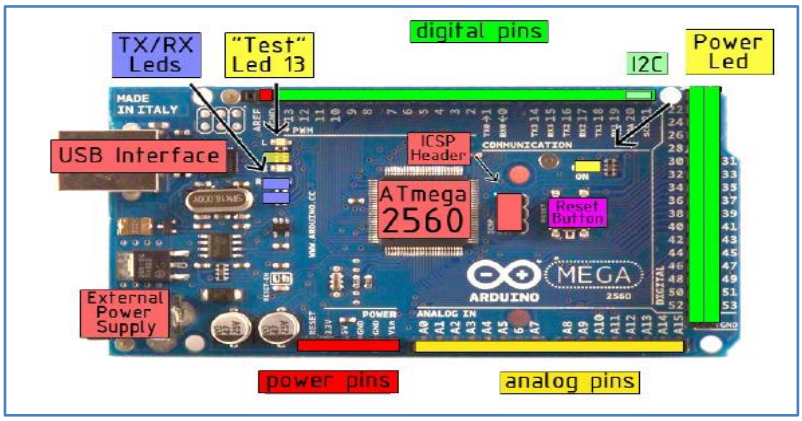



**Figure-2.1: Arduino Mega 2560 (components)**

The Arduino can be used to develop stand-alone interactive objects or it can be connected to a computer to retrieve or send data to the Arduino and then act on that data. The Arduino is an amazing device. Users can us to make many things from interactive works of art to robots. With a little enthusiasm to learn how to program the Arduino and make it interact with other components a well as a bit of imagination.Users can build many things.TheArduino can be connected to LED Matrix displays, RFID readers, buttons, switches,motors, temperature sensors, pressure sensors, distance sensors, webcams, printers, GPS receivers, and Ethernet modules. The Arduino board is made of an Atmel AVR Microprocessor, a crystal or

oscillator (basically a crude clock that sends time pulses to the microcontroller to enable it to operate at the correct speed) and a 5-volt linear regulator. USB connector is used to connect to a PC or Mac to upload or retrieve data. The board exposes the microcontroller I/O (Input/Output) pins to enable you to connect those pins to other circuits or to sensors, etc.

## ArduinoSchematic Diagram (see figure 2.2)

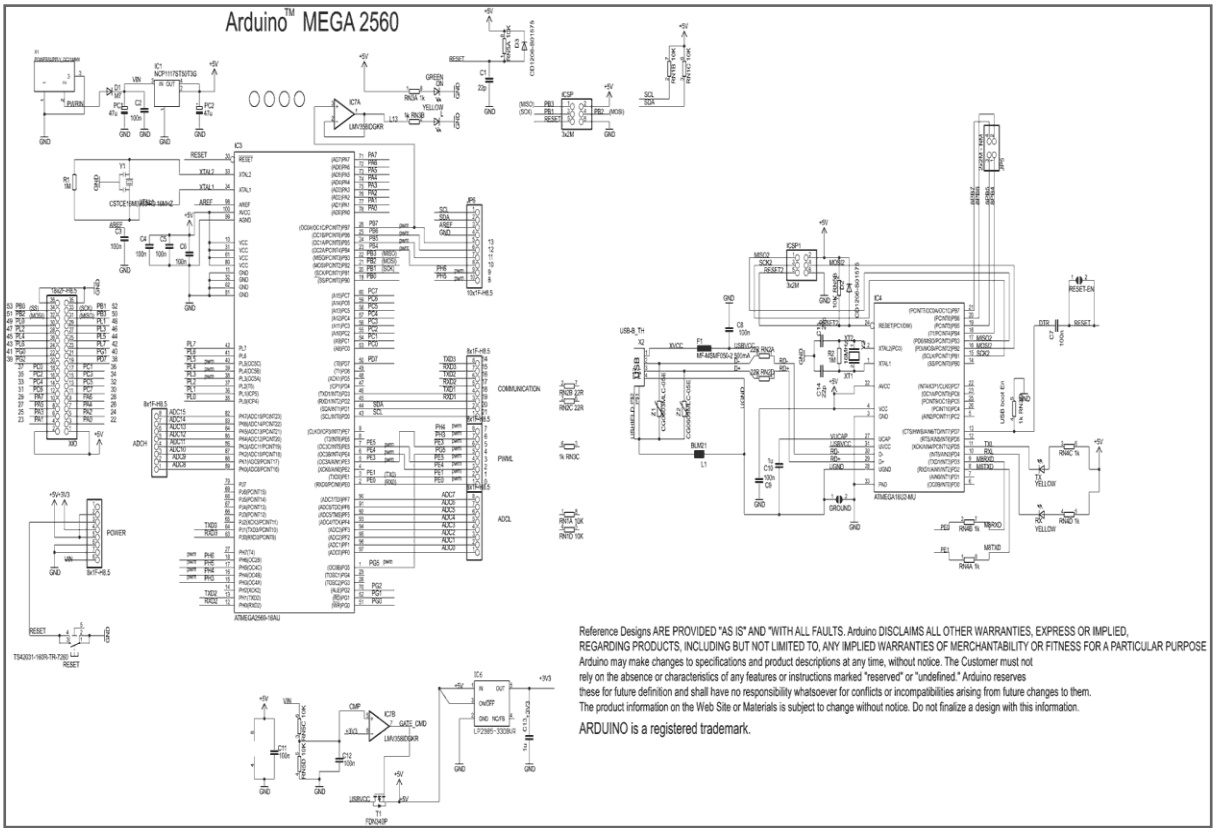



**Figure-2.2: Arduino Mega 2560 Schematic**

## Powering the Arduino Mega 2560

The most common way to power the Arduinois to plug one end of the USB cable into the Arduino and the other end into a computer. The computer will then power the Arduino. (See figures 2.3 next page)

**Figure-2.3: Arduino USB Cable and Power Adapter**

Another way to power up the Arduino is to plug in a battery or wall adapter into the DC jack. The rating should be 9V DC 100-500mA power adapter, with a 2.1mm barrel plug and positive tip. Right next to the USB jack, there is a jumper with 3 pins. If the jumper is on the two pins nearest USB jack that means you are planning to power the Arduino via the USB cable. If it's on the two pins nearer the DC Jack then it means you are planning to power the Arduino using a 9V battery or wall adapter.

### 2.1.4 Arduino Software, and Controller Chip

### IDE Software to program the Arduino

The Arduino has his own application software thatenables the programmer to download and upload programs and other functionalities such as debugging. Download the Arduino IDE (The software name) from the Arduino download page. This software uses C language as the programming language for theArduino as shown in figure 2.4.

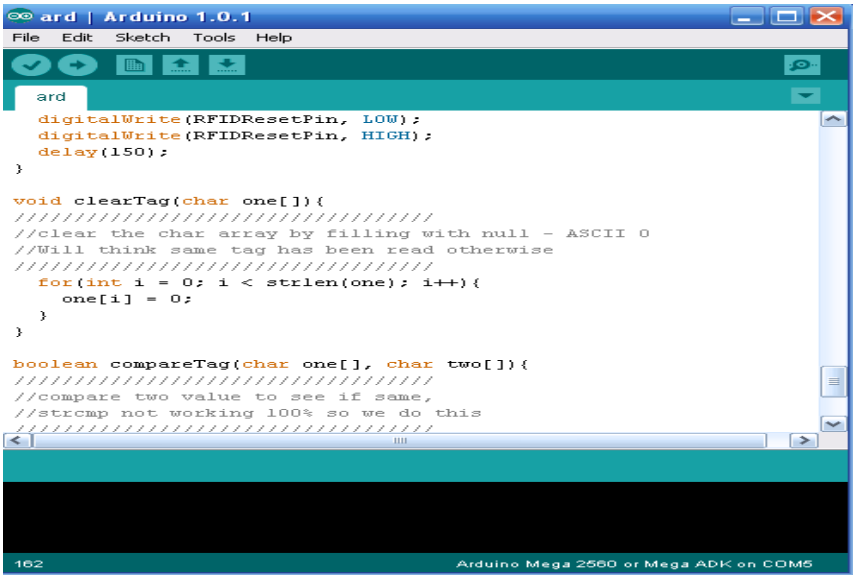



**Figure-2.4: Arduino IDE Software**

The communication between Arduino and PC is accomplished using USB cable to transfer data serially as shown figure 2.5

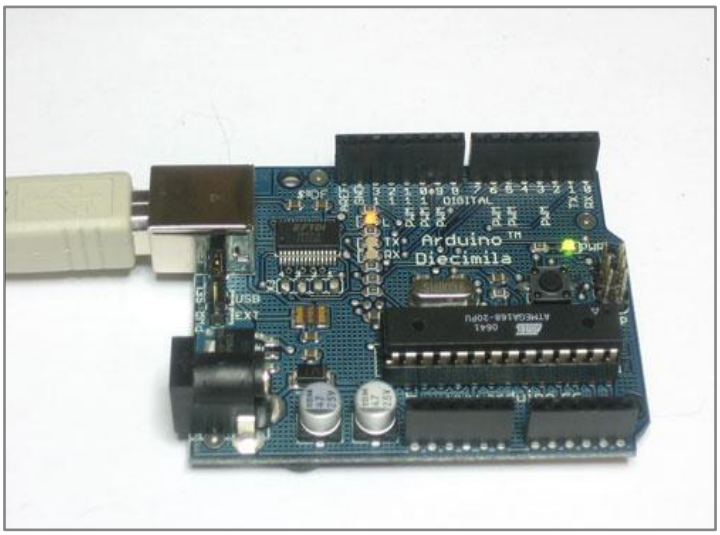



**Figure-2.5: Communication between Arduino and PC**

Seeing serial data requires a monitor, which like your display monitor will show us what data is being transferred as shown in figure 2.6


**Figure-2.6: Arduino Serial Monitor**

We can also send serial data back to the Arduino using the Serial Monitor.The Baud Rate: default setting is 9600 baud for serial monitor as shown in figure 2.7.

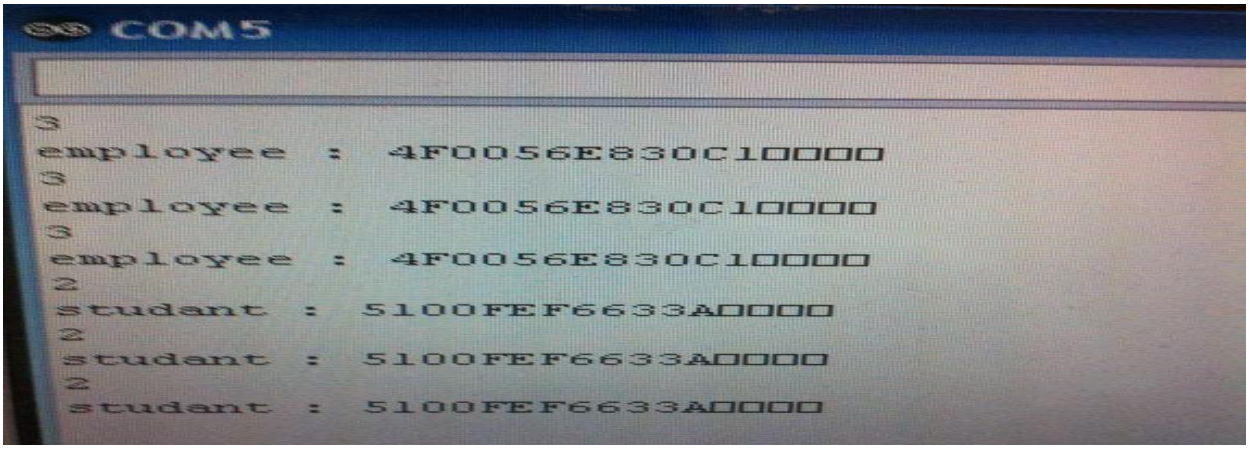



**Figure-2.7: Detecting and displaying the RFID tag by Serial Monitor**

## 2.1.5 Configuring the Arduino Software

After downloading and installing the Arduino software, the first step is to configure the Arduino software for the correct chip. Look for the chip on the Arduino that looks like as shown **in** figure 2.8.


**Figure-2.8: Selecting Arduino Chip**

Next, is to configure the Serial Port: (See figure 2.9).

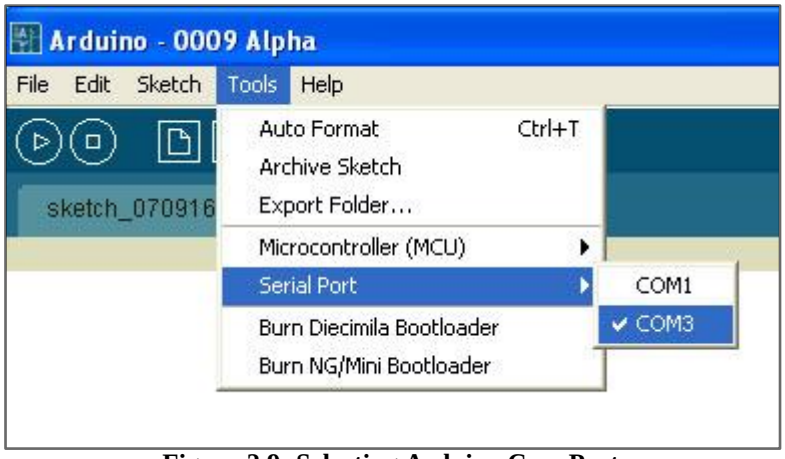



**Figure-2.9: Selecting Arduino Com Port**

Set the PC communication baud rate to 9600 pits per second as shown figure 2.10next page

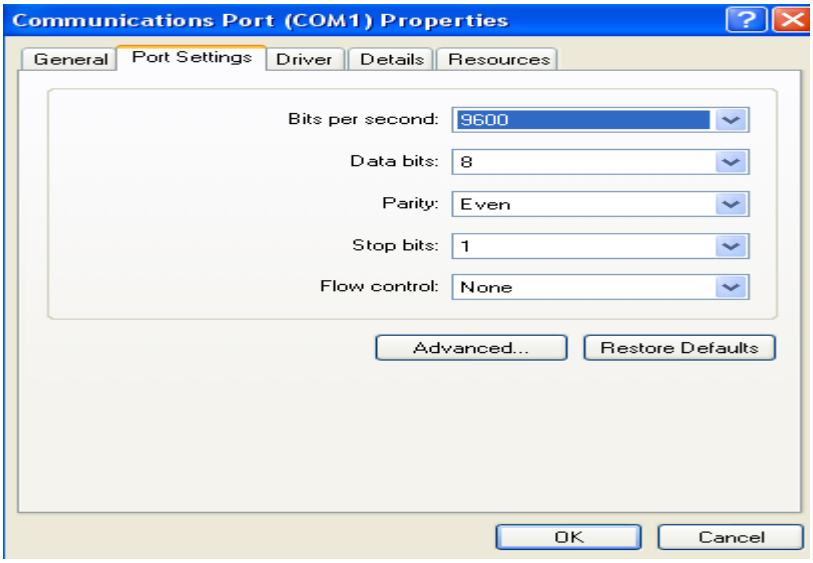



**Figure-2.10: Setup of Com Port Baud rate**

As shown in the figure 2.11 below, the code can be written in this window:

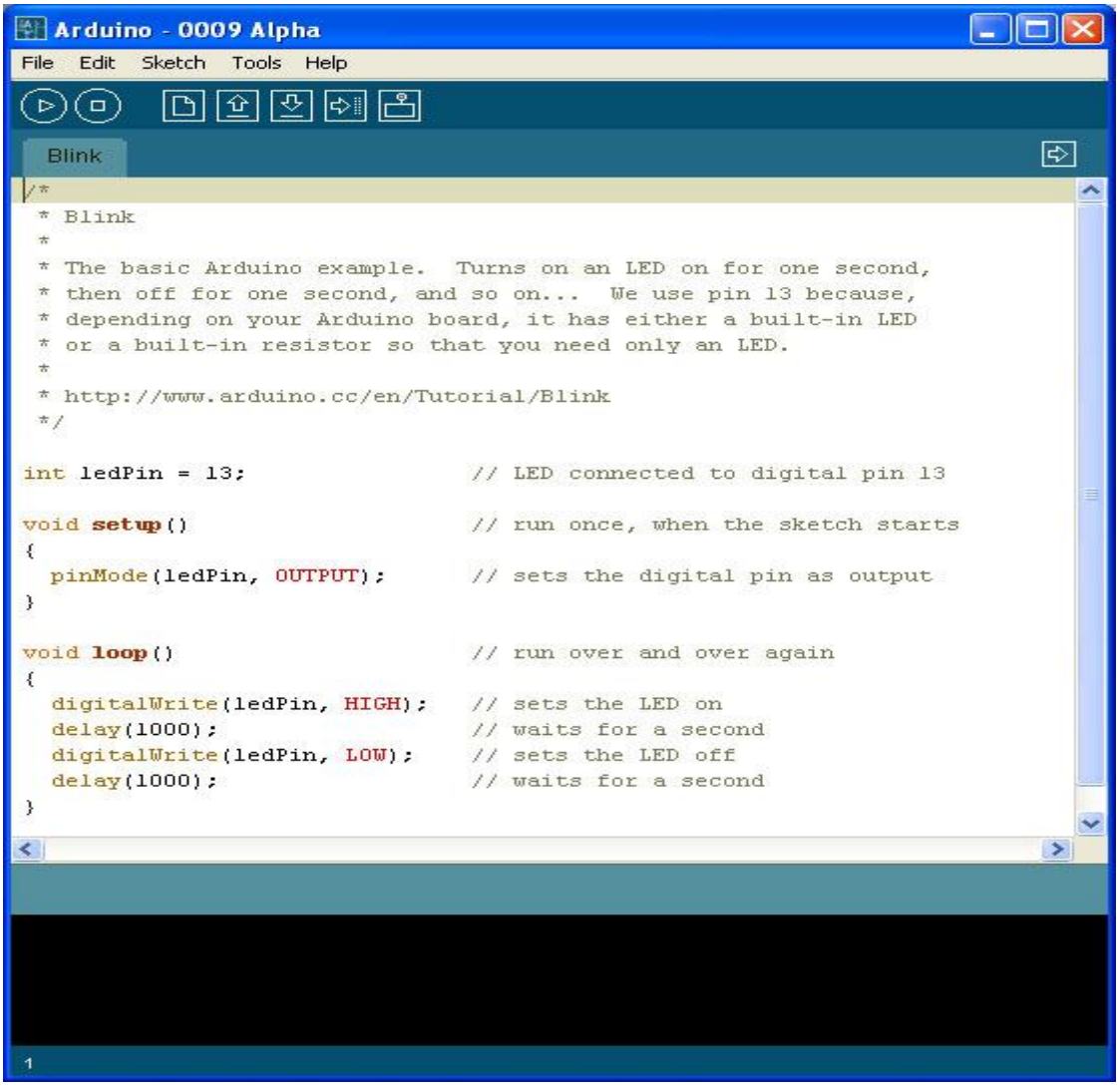



**Figure-2.11: Arduino Software**

After verifying the code, it can be sent to Arduino by pressing upload button

### 2.1.6 Introduction about Coding of IDE Arduino Software

There are two main sections in this software. Below are some of main functions that are used in programming the Arduino and essential to learn for even simple programs:

**The setup ()** function is called when a sketch starts. It is used to initialize variables, pin modes, start using libraries, etc. The setup function will only run once, after each power up or reset of the Arduino board.

After creating a setup () function, which initializes and sets the initial values, the **loop ()** function does precisely what its name suggests, and loops consecutively, allowing your program to change and respond. It is used it to actively control the Arduino board.

#define is a useful C component that allows the programmer to give a name to a constant value before the program is compiled. Defined constants in Arduino do not take up any program memory space on the chip. The compiler will replace references to these constants with the defined value at compile time.

This can have some unwanted side effects though, if for example, a constant name that had been #defined is included in some other constant or variable name. In that case the text would be replaced by the #defined number (or text). See figure 2.12

```
int led =13;  // defines the variable "led" with value of 13

voidsetup(){

  pinMode(led,OUTPUT);// initialize the digital pin as output.
}

voidloop(){
digitalWrite(led,HIGH);   // turn the LED on
  delay(1000);              // wait for a second
  digitalWrite(led,LOW);    // turn the LED off
  delay(1000);              // wait for a second
}
```

**Figure-2.12: Defined Number**

### 2.1.7 Atmega328 Controller

The Arduino uses atmega328 microcontroller which uses 2 KB of SRAM. Memory is part of the microcontroller whose function is to store data. Two important concepts are brought to us: addressing and memory location. Memory consists of all memory locations, and addressing is nothing but selecting one of them. This means that we need to select the desired memory location on one hand, and on the other hand we need to wait for the contents of that location. Also, the main operation that occurs is the read/wirte operation. This is done by supplying an additional line called control line. This line is designated as R/W (read/write). This control line is used to control which operation CPU shall execute whether it is read or write operation. So, if R/W is logic 1 then it is a read operation and if it is 0 then it is a write operation.

## 2.2: (RFID Reader, RFID Tag, and IR Sensor)

### 2.2.1 Sensor Definition

A sensor (also called detectors) is a device that measures a measurable attribute and converts it into a signal which can be read by an observer or by an instrument.

### 2.2.2 Sensor Options, and Selections

The common used sensors which are suitable for our project are:

3 RFID Sensor: RFID is a form of automatic identification that uses electric or magnetic fields at radio frequencies to transmit information. Every RFID system includes a radio frequency (RF) subsystem, which is composed of tags and readers. This technology has been described as a major enabling technology for automated, contactless, and wireless data collection.

4 Passive infrared sensors: Looks for body heat. No energy is emitted from the sensor. The passive infrared (PIR) sensor has the ability to detect movements at distances up to 6-45 m. Power supply input: 5-20 VDC. Cost range is $6.9.

5 Active Infrared Sensor: Active infrared sensors illuminate detection zones withlow power infrared energy supplied by laser diodes operating in the near infrared region of the electromagnetic spectrum. Power supply input: 3-20 VDC. Cost range is $1.9.

6 Ultrasonic (active): Sends out pulses of ultrasonic waves and measures the reflection off a moving object. The range of frequency: above 40-50 KHZ. Firstly, it is inaudible to humans and therefore undetectable by the user. Secondly, ultrasound waves can be produced with high directivity. Thirdly, they are a compression vibration of matter (usually air). Finally, they have a lower propagation speed than light or radio waves.Power supply input: 5-20 VDC. Cost range is $2.9.

7 Sensors that combine passive infrared presence detection with ultrasound or Doppler radar. The passive infrared-ultrasonic combination provides enhanced accuracy for presence and queue detection, vehicle counting, and height and distance discrimination.

8 Microwave (active): Sensor sends out microwave pulses and measures the reflection off a moving object. Microwave sensors utilize electromagnetic fields and devices

internally operating at frequencies starting from ~300 MHz up to the terahertz range. Since it is using electromagnetic waves that can affect health of human beings, more research can be conducted to make sure about its effects.

9 Passive Acoustic Array Sensors: Acoustic sensors measure vehicle passage, presence, and speed by detecting acoustic energy or audible sounds produced by vehicular traffic from a variety of sources within each vehicle and from the interaction of a vehicle s tires with the road.

In conclusion, the most common used sensors are RFID sensor, infrared sensors, and ultrasonic sensors. By comparing these types of sensors according to the accuracy, simplicity, security, reliability, health, and cost, it is recommended touse both RFID sensor and infrared sensors.

### 2.2.3 RFID Sensor Operation:

RFID readers are devices that wirelessly communicate with tags to identify the item connected to each tag and possibly associate the tagged item with related data. In other words, readers send electromagnetic signal to the tag and read its response. They generally transmit their observations to a controller and a computer system running RFID software.
The antenna of the RFID reader emits radio signals to activate the tag. When an RFID tag passes through the electromagnetic zone, it detects the reader's activation signal. The reader decodes the data encoded in the tag's integrated circuit (silicon chip) and the data is passed to the host computer for processing. See figure 2.13

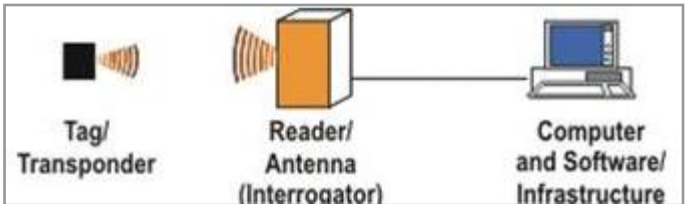



**Figure-2.13: RFID Sensor Operation**

### 2.2.4 RFID Readerand RFID Tag Properties:

- The ID-12 comes with internal antennas.
- RFID-12 has read range of 12 cm.
- RFID reader needs 5 VDC power.

- RFID-12 sends 125 KHZ signal to power tags.
- Each tag sends back a unique 12 digit hex serial number.


**Figure-2.14: RFID Reader**

- Passive RFID tag is cheaper, smaller, and it does not need battery.
- Passive RFID tags lasts for longer time than active tags.
- RFID technology does not require a line of sight.
- RFID tags can stand in a harsh environment.


**Figure-2.15: RFID Tag**

### 2.2.5 Sensor Placement and Alignment

The RFID readers at the entry and exit of the parking lot should be placed at minimum distance of 10 cm from each other. In order to get accurate results, we needed to place and align the sensors properly. That was achieved by following the instructions to have a minimum distance of 10 Cm between them as well as installing them at the same level and angle.

### 2.2.6 RFID Reader Circuit

Our RFID reader works on 5 VDC power supply to produce a resonant frequency of 125 KHz to power RFID tags. Our design circuit is shown in figure 2.16 below.

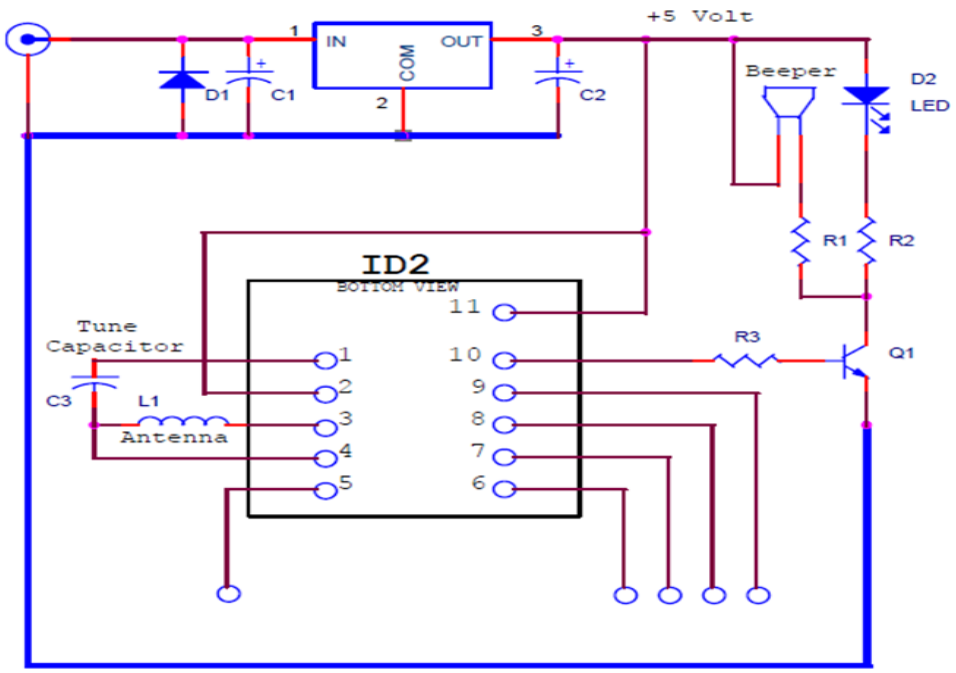



**Figure-2.16: Transmitter Sensor Circuit**

## 2.2.7 Designing the RFID reader to Operate at 125 KHZ

Radio frequency identification technology is based on resonant frequency principle where the inductive reactance is equal to capacitive reactance. That means:

$\omega L = 1/\omega C$

$$\omega = \frac{1}{\sqrt{LC}}$$

$$f_{res} = \frac{1}{2\pi\sqrt{C_{res} \bullet L_{antenna}}}$$

As per the data sheet of RFID reader type ID-12, the inductance of the reader L= 1.08 mH, and the capacitance C= 1.5nF.

By substituting these values in this formula:

$$f_{res} = \frac{1}{2\pi\sqrt{C_{res} \bullet L_{antenna}}}$$

We achieve the 125 KHz resonant frequency.

Figure 2.17 shows start and end sequence of magnetic timing in the RFID reader

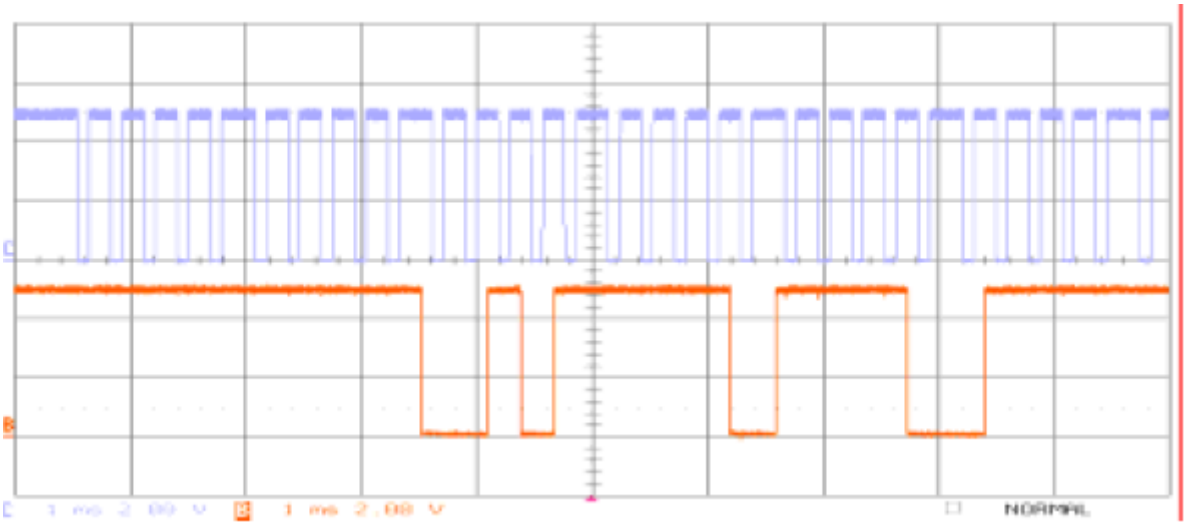


**Figure-2.17: Start and End Sequence of Magnetic Timing**

### 2.2.8 RFID Tag Principle of Operation

RFID tag consists of microchip which stores the ID number and antenna. When the RFID tag is powered by the 125 KHz frequency signal, the data which are stored in the microchip will be sent through the magnetic signal to the reader. See figure 2.18

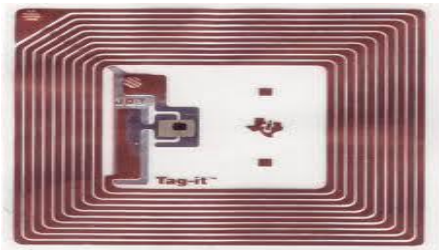



**Figure-2.18: RFID Tag Circuit**

### 2.2.9 RFID Tag Properties

Sparkfun RFID tag is a passive tag and not reprogrammable (read only) type.That is because the Data is stored in ROM memory. So, we cannot change, delete, or add to the information stored in the microchip of the RFID tag.

### 2.2.10 RFID Reader Setup and Arduino Interface

#### RFID Reader connection Setup

For suitable use, the breakout board was soldered with the RFID-12 reader using soldering iron in order to be connected to the Arduino in a safe and professional way as shown in the figures below:

1. RFID Breakout

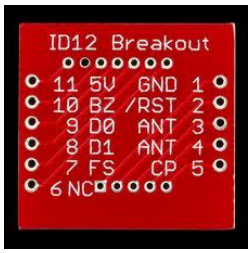
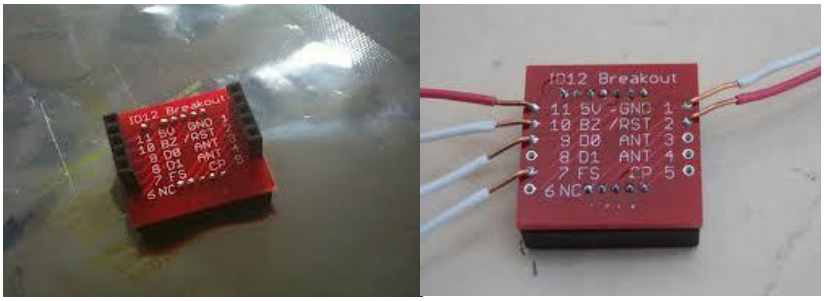



**Figure-2.19 : RFID ID-12 Breakout**

2. RFID-12 Reader

**Figure-2.20 : RFID ID-12 Reader**

3. Final Appearance



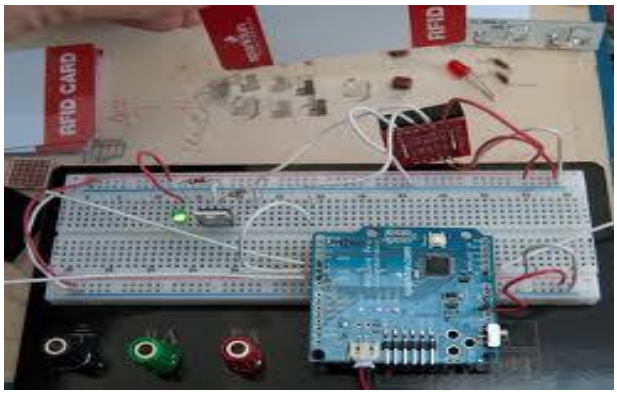




**Figure-2.21: RFID Final Setup**

## 2.2.11RFID Reader and Arduino Interface

The ID-12 RFID reader requires +5V (pin 2, 11), ground (pin 1, 7), and serial communication (pin 9) to send The ID to the Arduino. Refer to figure 2.22

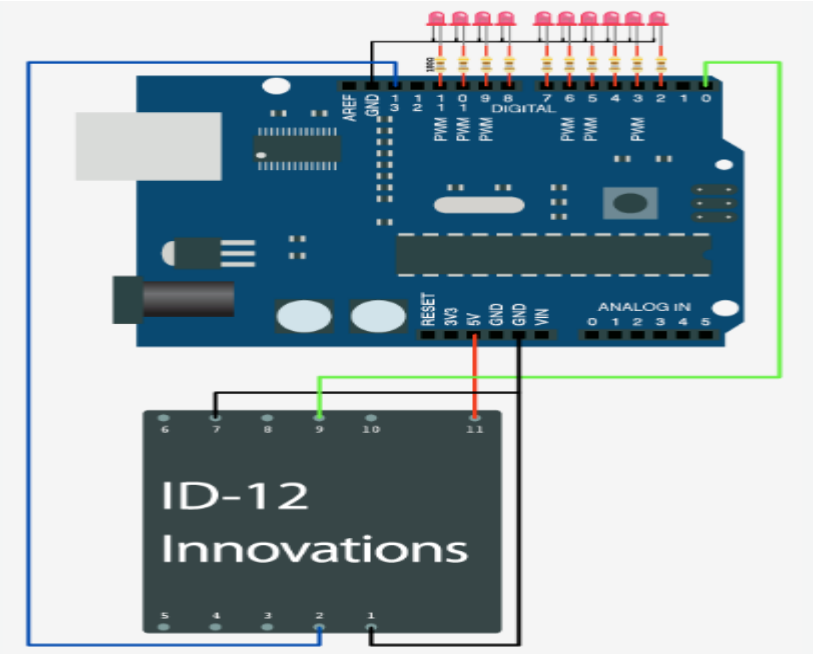



**Figure-2.22 : RFID ID-12 Reader and Arduino**

# 2.3 :( IR Sensor)

## 2.3.1 Introduction about IR Sensors

Infrared sensors are a type of light sensors. They function in the infrared part of the frequency spectrum. They can be used for the same purposes:

1. Detecting the presence of an object
2. Measuring the distance to a nearby object (clever electronics required)

## 2.3.2 Sharp IR Sensors Type (GP2Y0A21YK0F)

It is an analog output type sensor because it detects infrared light between 10-80 cm.
Inside the tin can, there is anIR detector, transmitter, and a demodulator. Refer to figure 2.23 for Sharp IR Sensors Type (GP2Y0A21YK0F) internal circuit.

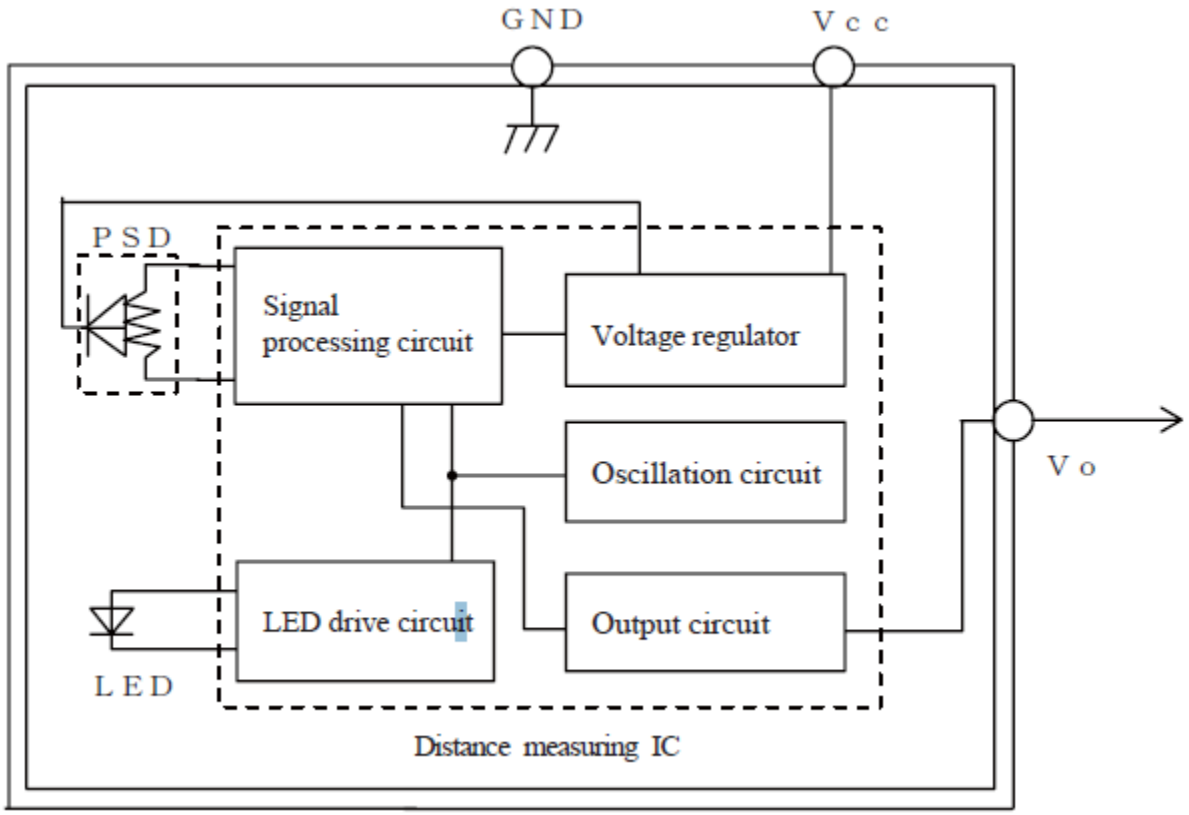



**Figure-2.23 : Internal Circuit of Sharp IR Sensor**

### 2.3.3 Features of Sharp IR Sensor

- Measures distance in range from 10-80cm.
- Designed to interface to small micro-controllers.
- Less influence on the colors of the reflected objects and their reflectivity.
- Typical current consumption 30 mA.
- 3.1V at 10cm to 0.4V at 80cm.
- Operating temperature -40-70°C.

### 2.3.4 Elimination of the effect of the stray IR light

To eliminate the effect of the stray IR light, the IR emitters are modulated and the output of the IR Detectors is demodulated to look for specific range of detection.

### 2.3.5 Calibration of Sharp IR

The sensor was adjusted to make detection between 20cm and 30 cm. The 30 cm is the free space between the position of IR sensor and beyond the entranceas shown figure 2.24.

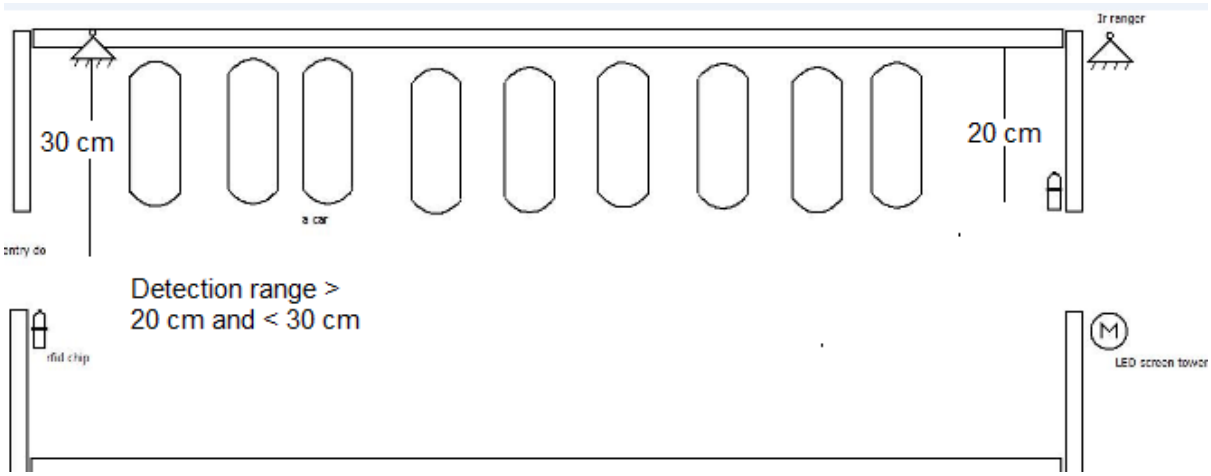



**Figure-2.24 : Configuring IR Detection range**

That was achieved by setting the free distance (20-30 cm) in the IR code, measuring a distance of 30 cm from the IR and comparing the actual measurement with the measurement in the serial monitor of Arduino. So, if there is a car isin a range greater than 20 cm and less than 30 cm in addition to a valid RFID tag, the gate will open. If the car is not within this distance, that means it gets in and the gate will close.

### 2.3.6 Sharp IR Connection

This type of sensors has three wires:

- Red is VCC the power supply.
- Black is the ground.
- Yellow is the IR analog pulse.

### 2.3.7 IR and Arduino Interface

Connecting of IR sensor with arduino is indicated in the figure 2.25

- Red wire can be connected to +5V in Arduino
- Black wire is connected to ground in Arduino
- Yellow wire is the analog pulse which can be connected to analog inputs in Arduino.

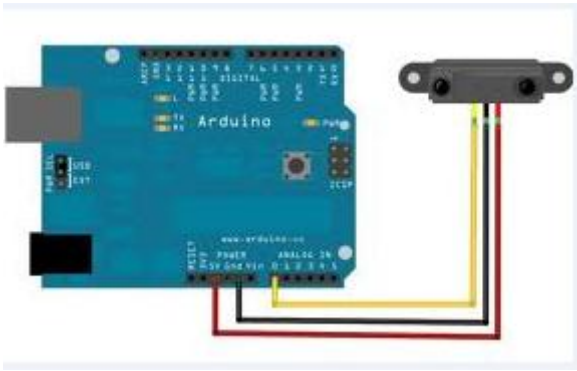



**Figure-2.25 : IR and Arduino Connection**

# Chapter 3: Display Circuit, Servo Motor and Power Supply

## 3.1:Display Circuit

### 3.1.1 Display Definition

A display device is an output device for presentation of information in visual. Some displays can show only digits or alphanumeric characters. They are called segment displays, because they are composed of several segments that switch on and off to give appearance of desired output. The segments are usually single LEDs or liquid crystals. They are mostly used in digital watches and pocket calculators.

### 3.1.2 Display Options, and Selections

There are three outdoor displays that can be used in parking system:

1. LED or light-emitting diode (LED) is a semiconductor light sourceLEDs are used as indicator lamps in many devices and are increasingly used for other lightingLight-emitting diodes are used in applications as diverse as aviation lighting, automotive lighting, advertising, general lighting, and traffic signals.

2. LCD or liquid crystal display is a flat panel display, electronic visual display, or video display that uses the light modulating properties of liquid crystals (LCs). LCsdo not emit light directly.LCDs are used in a wide range of applications, including computer monitors, television, instrument panels, aircraft cockpit displays, signage, etc. They are common in consumer devices such as video players, gaming devices, clocks, watches, calculators, and telephones.

3. Seven- Segments: seven-segment display (SSD), or seven-segment indicator, is a form of electronic display device for displaying decimal numerals that is an alternative to the more complex dot-matrix displays. Seven-segment displays are widely used in digital clocks, electronic meters, and other electronic devices for displaying numerical information.

Our selection was on16 X 24 LED matrix display because it is clear, has low power consumption, cheap, and easy to configure it with Arduinomicrocontroller. See figure 3.1

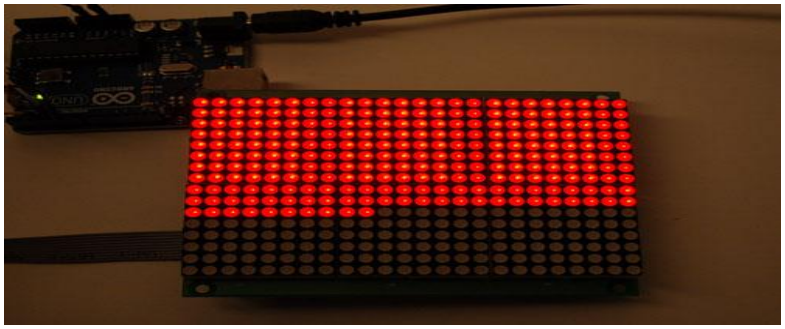


**Figure-3.1 : 16X24 LED Matrix**

## 6.2 16X24 LED matrix Operation, Properties, Placement, and Alignment

### 3.1.3 LED Matrix Properties

- Low Power Consumption (60mW).
- Clear display.
- Easy to program.
- It is designed to be 'chainable‘ up to 8 panels together to make an extra-long display as shown in the figure 3.2 below.

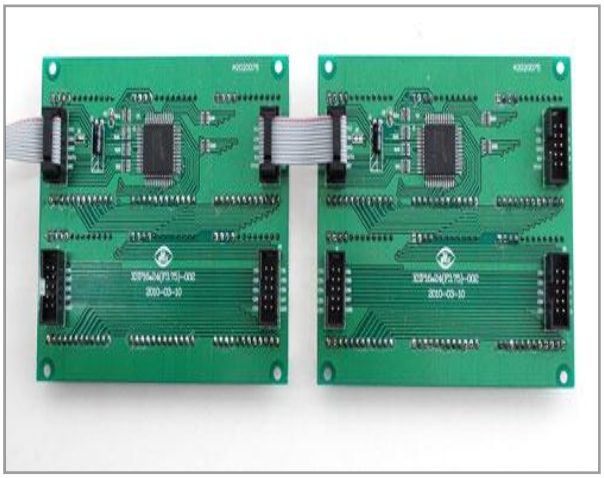


**Figure-3.2 : Connecting Two LED Matrix**

### 3.1.4 Schematics of LED Matrix:

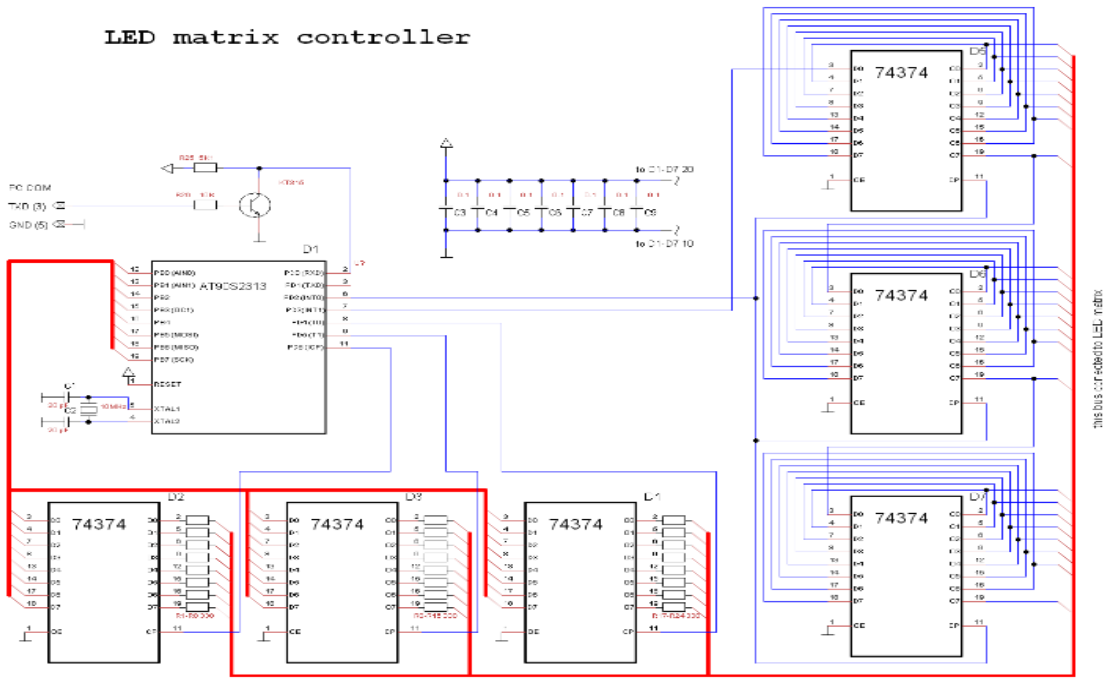



**Figure-3.3 : Schematic of LED Matrix**

### 3.1.5 Introduction about 16X24 LED Matrixes:

These LED panels take care of all the work of making a big matrix display. Each panel has six 8x8 red matrix modules, for a 16x24 matrix. The panel has a HT1632C chip on the back which does all the multiplexing work and has a 3-pin SPI (Serial Peripheral Interface) -like serial interface to talk to it and set LEDs on or off. There are a few extras as well, such as being able to change the brightness of the entire display, or blink the entire display at 1 Hz.

The Arduinohas a fulllibrary for this type of LED matrix that not only takes care of controlling the display but also intelligently handles chained displays, so that they appear to be one long matrix. The library has functions for drawing pixels, lines, rectangles, circles and text.

### 3.1.6 Wiring of LED Matrix:

Wiring is thankfully fairly simple, much easier than trying to actually wire up 6 x 8x8 matrices. The HT1632C driver chip requires only 3 data pins - data, write, and chip select (cs).

Begin by plugging in a 2X5 pin IDC cable into the top left socket. See figure 3.4

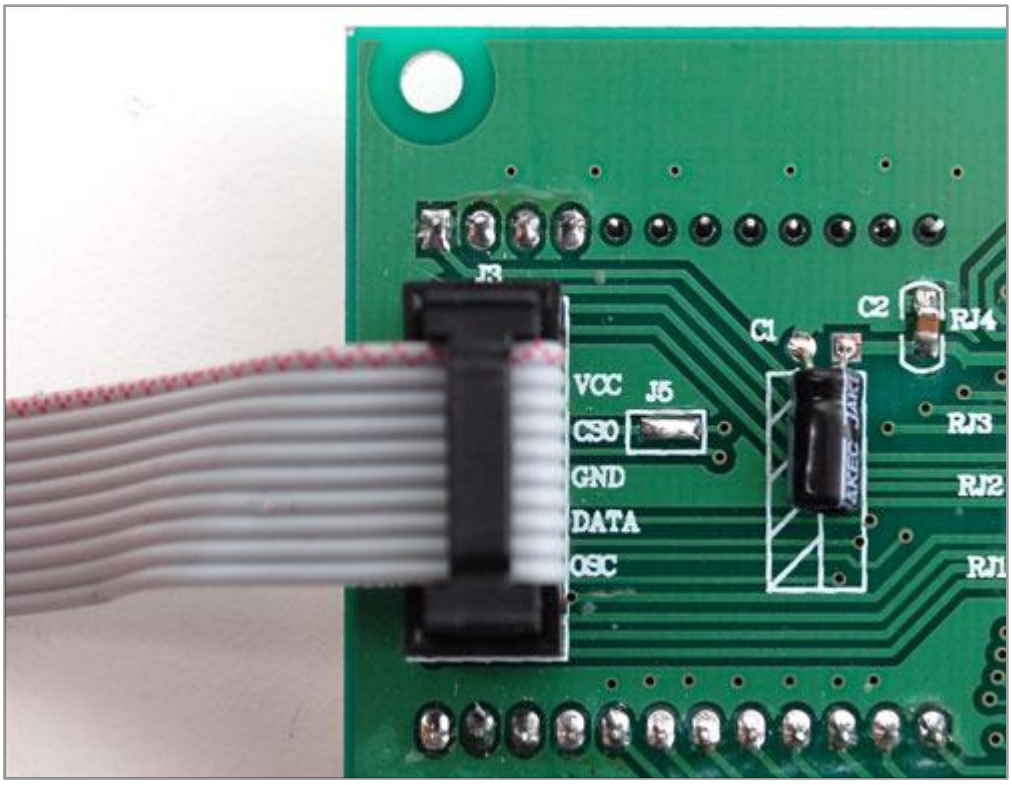



**Figure-3.4 : Connecting LED Matrix to Arduino**

Next, you will need a few wires, the following convention can be used: a Red wire for VCC (+5V), a Black wire for Ground, a White wire for CS0 (chip select #0), an orange wire for Data and a yellow wire for Write.

So, the connection between LED matrix and Arduino is as shown in the table below and figure 3.5 in next page.

| From Pin # at LED Matrix | To pin # at Arduino Board |
|---|---|
| VCC (red) | +5V |
| Ground (black) | Ground |
| DATA(orange) | digital pin 2 (As configured in the program) |
| WRITE (yellow | digital pin 3 (As configured in the program) |
| and CS0 (white) | digital pin 4 (As configured in the program) |

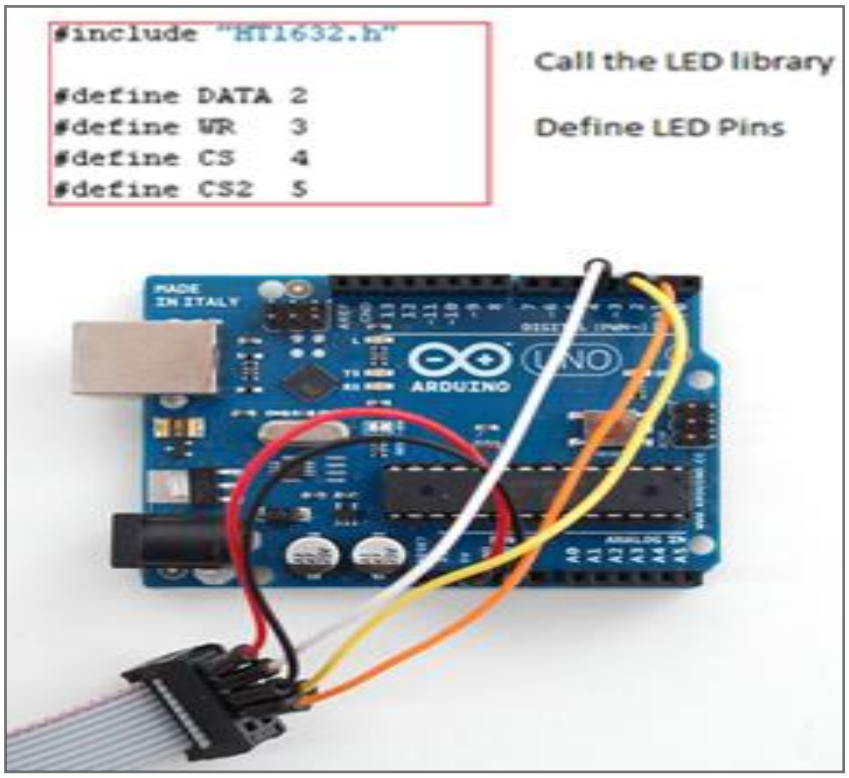



**Figure-3.5 : Connection between LED Matrix and Arduino**

# 3.2 (Servo Motor)

## 3.2.1 Motor Options, and Selections

The common actuators which are suitable for the project are:

1. DC Motor: A DC motor is a mechanically commutated electric motor powered from direct current (DC). A DC motor has a two wire connection. All drive power is supplied over these two wires—think of a light bulb. When you turn on a DC motor, it just starts spinning round and round. Most DC motors are pretty fast, about 5000 rpm (revolutions per minute).With the DC motor, its speed (or more accurately, its power level) is controlled using a technique named *pulse width modulation*, or simply PWM.
2. Servo Motor: A servomotor is a rotary actuator that allows for precise control of angular position.A servo motor is an entirely different story. The servo motor is actually an assembly of four things: a normal DC motor, a gear reduction unit, a position-sensing device (usually a potentiometer—a volume control knob), and a control circuit. The servo has a 3 wire connection: power, ground, and control. The

power source must be constantly applied; the servo has its own drive electronics that draw current from the power lead to drive the motor. Refer to figure 3.6

**Figure-3.6 : Servo Motor type (HS-5485HB)**

In conclusion, we selected servo motor because it is easy to deal with, more reliable, and easy to configure with Arduino. Unlike DC motor, it does not require external circuits such as switches or H-bridge components.

### 3.2.2Introduction about Servo Motor

A servomotor is a rotary actuator that allows for precise control of angular position. It consists of a motor coupled to a sensor for position feedback, through a reduction gearbox. Italso requires a relatively sophisticated controller, often a dedicated module designed specifically for use with servomotors.

Servomotors are used in applications such as robotics, CNC machinery or automated manufacturing.

### 3.2.3 Servo Motor TypeHitecModel (HS-5485HB)

The HS-5485HB is Hitec's second digital standard servo world. This servo can be programmed using our hand-held servo programmer such as Arduino software. It can be configured to travel from 0° up to 180°.

Servo speed is stated in seconds. Just as a car has a 0-60 mph time, a servo has a 0-60 degree time. The lower the 60 degree time, the faster the servo operates. Its speed can be adjusted to be slow or fast using software such as Arduino software.

### 3.2.4 Features of Servo Motor

- Designed to interface to small micro-controllers.
- Required Pulse: 3-5 Volt Peak to Peak Square Wave
- Operating Voltage: 4.8-6.0 Volts
- Operating Temperature Range: -20 to +60 Degree C
- Current Drain (4.8V): 8.8mA/idle and 400mA no load operating
- Current Drain (6.0V): 9.1mA/idle and 500mA no load operating

### 3.2.5 Controlling Servo Motor

Servos are controlled by sending them a pulse of variable width. The control wire is used to send this pulse. The parameters for this pulse are that it has a minimum pulse, a maximum pulse, and a repetition rate. Given the rotation constraints of the servo, neutral is defined to be the position where the servo has exactly the same amount of potential rotation in the clockwise direction as it does in the counter clockwise direction and that position is always around 1.5 milliseconds (ms). See figure 3.7.

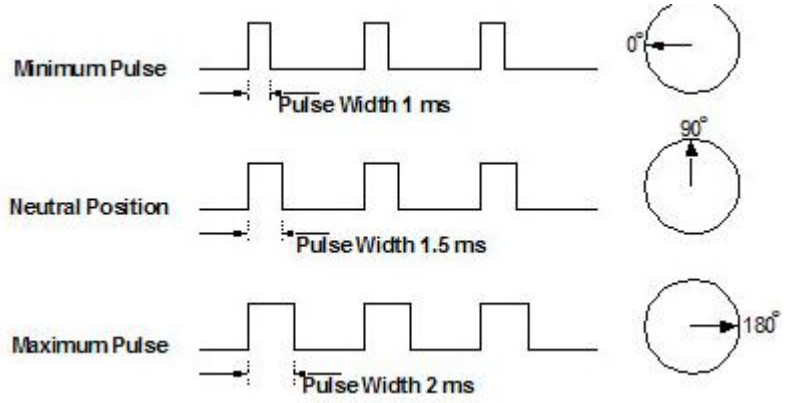



**Figure-3.7 : Controlling Servo Motor**

The angle is determined by the duration of a pulse that is applied to the control wire. This is called Pulse width Modulation. The servo expects to see a pulse every 20 ms. The length of the pulse will determine how far the motor turns. For example, a 1.5 ms pulse will make the motor turn to the 90 degree position (neutral position).

### 3.2.6 Servo Motor Connection

All of our Hitec servos come with the “S” or universal connector. This connector works with any brand of receiver, servo controller or servo extension.

This type of Motors has three wires:

- Red is VCC the power supply.
- Black is the ground.
- Yellow is for the digital pulse that controls the rotation of the motor.

### 3.2.7 Servo Motor and Arduino Interface

Red wire can be connected to +5V in Arduino

Black wire is connected to ground in Arduino

Yellow wire is the digital pulse which can be connected to digital output in Arduino.

See figure 3.8 for connection between between servo and Arduino

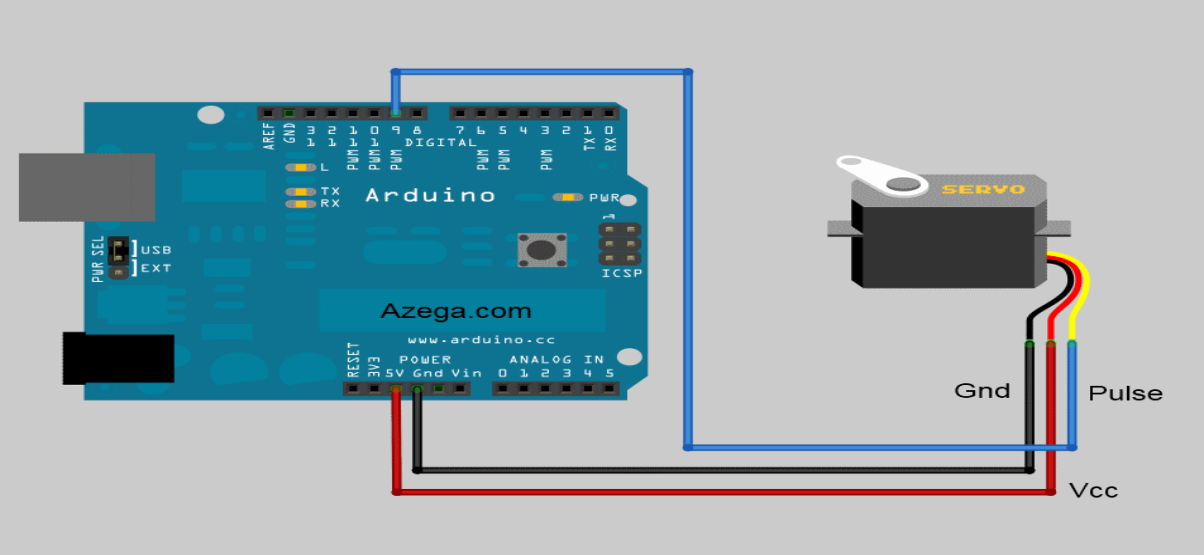



**Figure-3.8 : Servo Motor and Arduino connection**

## 3.3Electrical Power Supply

A power supply is a device that supplies electric power to one or more electric loads. A regulated power supply is one that controls the output voltage or current to a specific value; the controlled value is held nearly constant despite variations in either load current or the voltage supplied by the power supply's energy source.

### 3.3.1DC power supply block diagram

Regulated DC power supply is done by many steps: AC current is rectified and then filtered by capacitor to obtain rippled current and after that regulated by voltage regulator to obtain DC regulated power supply. See figure3.9.

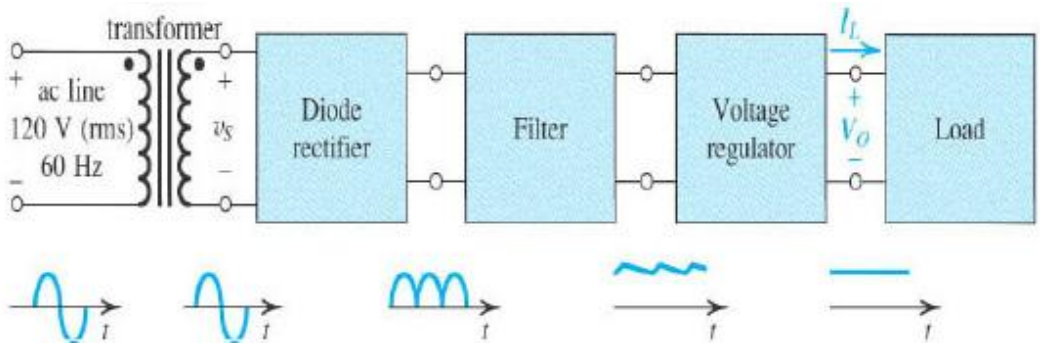



**Figure-3.9: Power Supply Block Diagram**

### 3.3.2 Component Definitions

**Rectifier**

All DC power supplies have a rectifier, consisting of a set of diodes that changes the reversing flow of AC to one-way DC. While the rectifier prevents the current from reversing direction, the DC it produces carries a lot of electrical noise. You can charge batteries with it, but any electronic device, such as a stereo, would pick up a loud, low-frequency hum. To eliminate the noise, the power supply has filter circuits that smooth the DC output. So the rectifier will remove positive or negative cycle as it is designed to produce DC.

**Filter**

Other than the issues of low power and limited life, a battery is an ideal DC source. It delivers steady current virtually free of electrical noise. Rectified DC, on the other hand, needs filter circuits to remove noise or minimize ripple voltage. Usually, these filters are simple, consisting of large electrolytic capacitors. Though the DC coming from the filter has less noise, the voltage can wander unpredictably, causing reliability issues with sensitive electronics. To solve that problem, the power supply uses a voltage regulator.

**Regulator**

A voltage regulator "locks" the power supply's voltage to a fixed value. A circuit's power demands can vary over time; unless the power supply has a regulator, a greater current demand can reduce the supply's voltage. With the regulator, the voltage remains steady. The regulator also removes some of the power supply's remaining noise.

Circuit of DC power supply is simple as indicated in figure 3.10

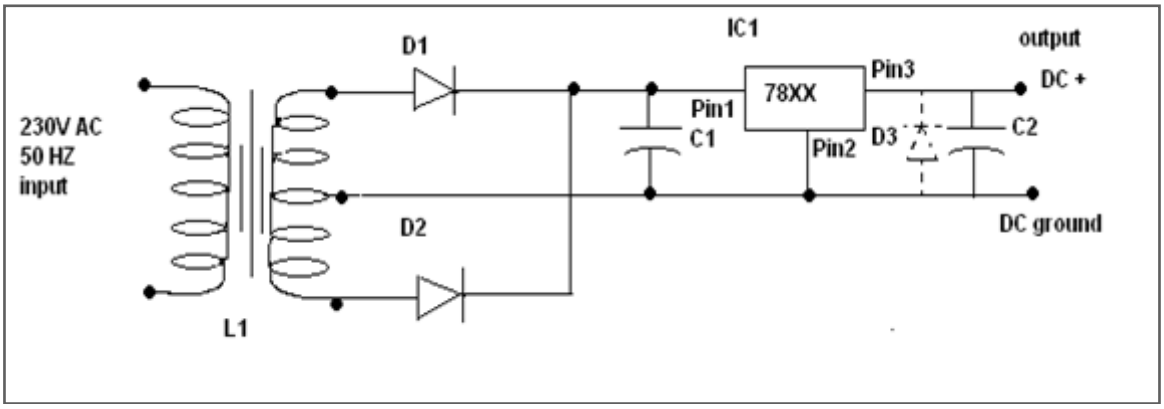



**Figure-3.10: Power Supply Circuit Diagram**

### 3.3.3Components of Power Supply

1. D1, D2 = SCHOTTKY RECTIFIER 5Amp
2. L1 = center tap Step down transformer with input of 120 V AC 60 Hz and output of (13.8V AC 60Hz)
3. C1 = 3300uf capacitor
4. C2 = 10 ufcapacitor
5. IC1 : =L78S09 for +9V DC with 2A output (for microcontroller circuit ) and LM323T5V with 3A output (cellular shield ).

### 3.3.4 Components Selection:

1. Diode Schoriky rectifier, PIV (peak inverse Voltage<BV

VS Max=VO Max+VD= 13V +0.7V=13.7

PIV=2Vs (Max)-VD=2*13.8-.7=20.6V <80 (140 RMS reverse V)

2. Capacitor V min;

V P secondary-VD-Vr (ripple) =13.8V-0.7-.5=12.6 DCV

3. Capacitance1 ; C= Vm/(2fRVr)=3300uf

### 3.3.5 Operation of Power Supply Circuit

120V 60Hz AC power is stepped down to 13.8V 60Hz. After that, positive power cycle from transformer secondary will go through diode rectifier D1 and continue to positive side of capacitor filter C1.also the negative power cycle from transformer secondary through rectifier diode D2 to positive side of capacitor filter C1. The ripple in the voltages after rectifier

diodes is reduced by the filter to the desired one. Then the rippled voltage goes though the IC L78S09 to get regulated 9 DC V which is the supply of the 9 DC V microcontroller circuits. The LM323T is then connected in parallel with L78S09 to get 5V that supplies the cellular shield.

# Chapter 4: Database System

C# is a general purpose programming object oriented language invented around 1999 or 2000 by Anders Hejlsberg at Microsoft. It is very similar to Java in its syntax with a major difference being that all variable types are descended from a common ancestor class.

C# is intended to be a simple, modern, general-purpose, object-oriented programming language:

1. Unsafe operations such as direct memory manipulation are not allowed.
2. C# has been based according to the current trend and is very powerful and simple for building, scalable, applications.
3. Object Oriented Programming (OOP) is a method of developing software that models the real world using objeject. For example a driving simulation might have road objects, vehicle objects and possibly human objects.

So, the purpose of C# is to precisely define a series of operations that a computer can perform to accomplish a task. Most of these operations involve manipulating numbers and text, but anything that the computer can physically do can be programmed in C#.

**What is C# used for?**

It can be used for all sorts of applications such as computer games, and operation system. There are also web based applications running on the asp.net platform.

## 4.1 C# features:

1. C# is built in C++, so it behaves much like language. In fact, it is just that C# is so much friendlier than C++.it is much easier to build data base using C# than using C++ even though that C# built in C++.
2. You can also interoperate with other languages
3. It can be used to save a database of customers.

## 4.2 C# program general structure

C# programs can consist of one or more files. Each file can contain zero or more namespaces. A **namespace** can contain types such as **classes**, **methods**, and **eventhandlers**. (Refer to figure 4.1.

```
namespace SampleNamespace
{
    class SampleClass
    {
        public void SampleMethod()
        {
            System.Console.WriteLine(
              "SampleMethod inside SampleNamespace");
        }
    }
}
```

**Figure-4.1: C#, General Structure of the Program**

**Namespace**: is keyword used to declare a scope. This namespace scope lets you organize code and gives you a way to create globally unique types. They are used especially to provide the C# compiler a context for all the named information in your program, such as variable names

**Class**: All code and data in C# must be enclosed in a class. A class is a construct that enables you to create your own custom types by grouping together variables of other types, methods and events. A class is like a blueprint. It defines the data and behavior of a type.

**Methods**:A method is a code block that contains a series of statements. A program causes the statements to be executed by calling the method and specifying any required method arguments. In C#, every executed instruction is performed in the context of a method. The Main method is the entry point for every C# application and it is called by the common language runtime (CLR) when the program is started.

Methods are declared in a class by specifying the access level such as public or private, optional modifiers such as abstract or sealed, the return value, the name of the method, and any method parameters. These parts together are the signature of the method.

**Event handler:** is a method that contains the code that gets executed in response to a specific event that occurs in an application

Event handlers are used in graphical user interface (GUI) applications to handle events such as button clicks and menu selections, raised by controls in the user interface. A single event handler can be used to process events raised by multiple controls

## 4.3 C# selection in Parking Monitoring and Control System:

1. We can create a Windows Forms Application (screens)
2. Connect to multiple data sources, including Access and SQL Server
3. Add, edit, and update data in a database we choose and design
4. Enable users to choose their data source at runtime
5. Communicate with microcontroller.
6. It will help in having the control automated.
7. Develop database in C# is much better than in **Excel** or **Access**. Although **Excel** and **Access** can interface with inputs from devices but in fact the interface is only from side: they can read only and to have an action based in certain information in the database another program is needed.
8. C# advantage over **visual basic**:C# is a more efficient language than **VB** (meaning you can do the same work in less code), which translates into faster development

time. Both business and technical reasons are welcome. Also, In C# you can document the code you write using XML. C# is the only programming language in Visual Studio .NET with this feature.

9. C# is better than **Java** in that:

   A- It has dynamic variables, if you want them.

   B- It allows you to define new value (or non-reference) types

   C- It's deeply integration with Windows, in case it is needed

The C# program has been developed for the ***Parking Monitoring and Control System***done by *Windows Form Applicatio*n and all user interfaces (main form/screen) work and design has been done by the tool box and manipulation the properties that allows the programmer to add buttons, tab control, serial port in order to interface with devices such as the controller, list view, labels, text and combo boxes. However, working inside the user interface or the screen has been done by certain commands and codes.

The parking monitoring program in the C# consists of four screens (forms):

1- Tags screen where it can be access from tab tag control button. Adding new tags No, customer name, customer mobile name and expiration date of a tag and photo of the customer as well. See figure 4.2

   Tag information can be edited and an assigned tag for a customer can be deleted from database from the tag screen-database.

The list view –white screen in the middle will show data base we have

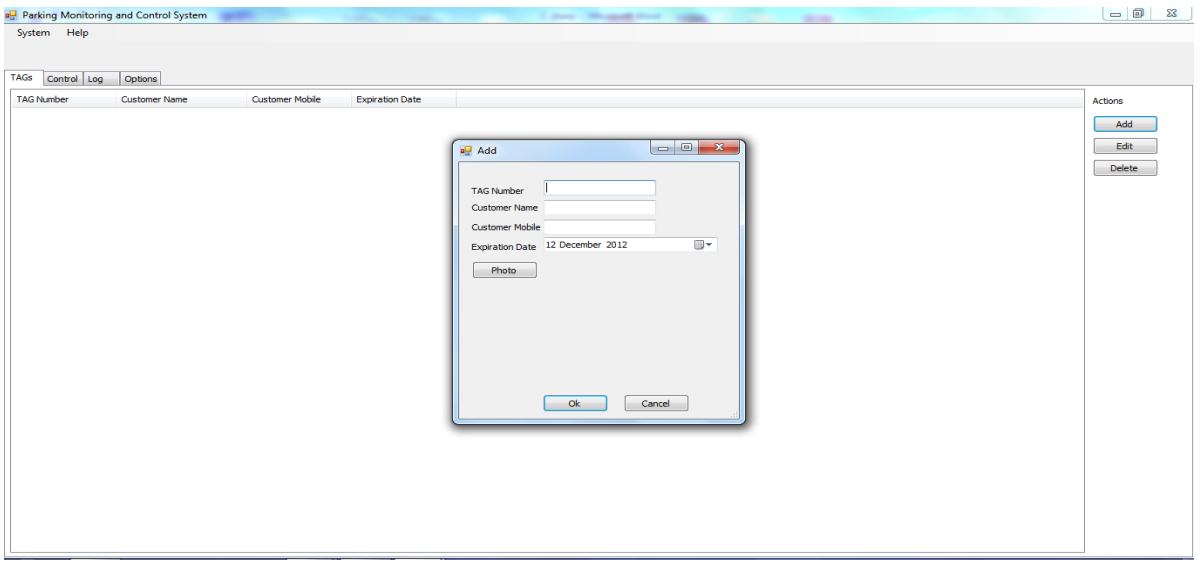



**Figure-4.2: Tag Form (Screen)**

2- Control screen that can be access from tab control button. Most of functionally of communication with the controller will happen here: Gates can be opened and closed in case a customer has a valid tag miss or forgot his tag. Gate can also be selected for this action. See figure 4.3

The textbox inside (dark one) will display to the program user action taken such as sending opening or closing command and whether communication port is open or close to communicate.

The total parking spaces can be set here to allow counting for the free parking spaces.

Displaying text will allow the user to display on the led matrix through the controller.

In the control screen we can write and read from the microcontroller

Start listening function will allow opening the serial port to read to whatever command sent by the controller and stop listening will stop the C# from reading the controller.

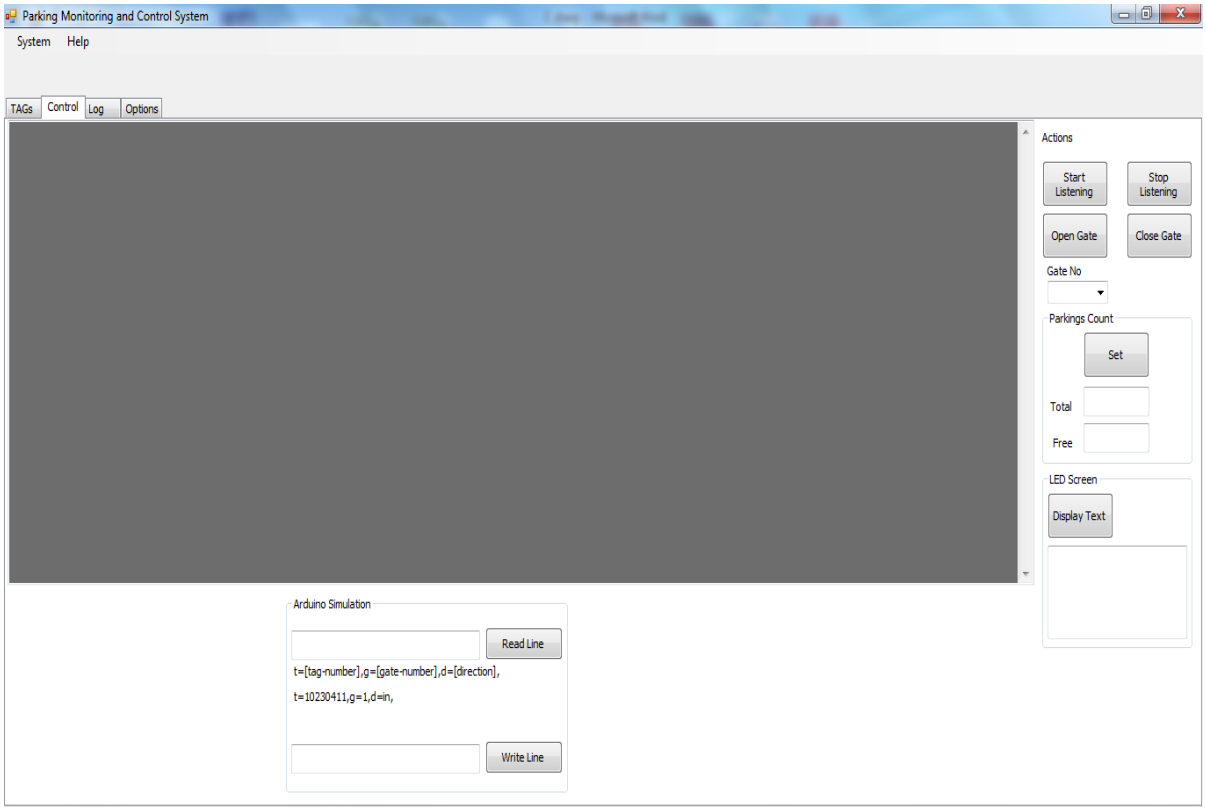



**Figure-4.3: Control Form (Screen)**

3-The log screen will make historical data (time in/ out) for the customers have defined tags and indicate the gates as well. See figure 4.4 below

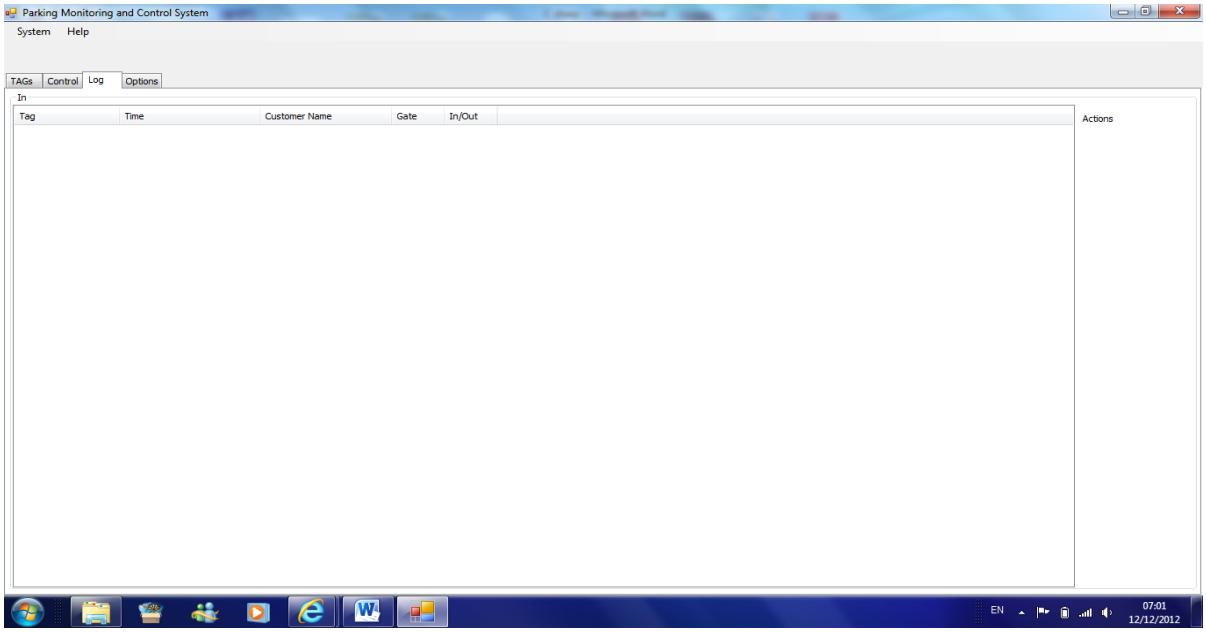



**Figure-4.4: LOG Form (Screen)**

4-Option screen; will allow user to select communication port so that controller can interface with C# program. See figure 4.5 below.

Option screen will also allow the user of the program to select Buad Rate- data transmission rate (bits/second) for modems.

It will also allow user send notification of limited days valid in his account to used tags.The option screen will user set validation of the tag: tag is used only by only one car at a time.

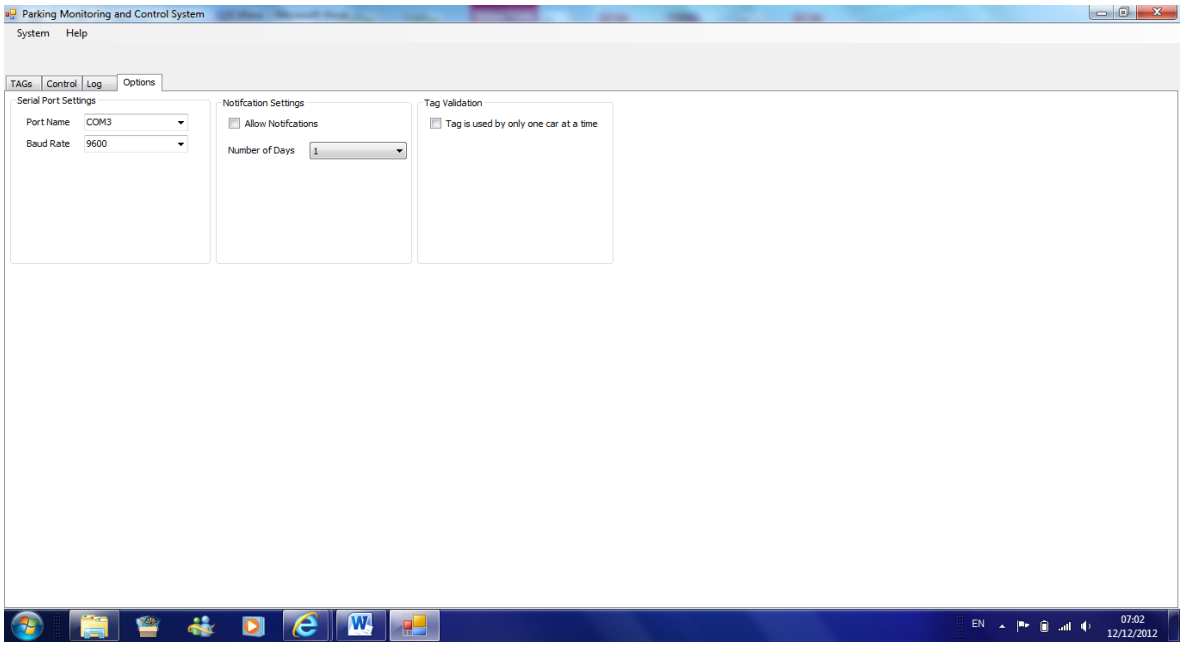



**Figure-4.5: Option Form (Screen)**

# CH 5: Cellular Shield with SM5100B

**5.1 Description:** The Cellular Shield for Arduino includes all the parts needed to interface theArduino with **an SM5100B cellular module**. This allows addingSMS, GSM functionalities to theArduino-based project. In order for the cellular shield to send SMS, SIM card and an antenna are neededas well as sending texts.

**5.2 About to GSM/GPRS Module - SM5100B**, The SM5100B is a miniature, quad-band GSM 850/EGSM 900/DCS 1800/PCS 1900 module, which can be integrated into a great number of wireless projects. It can be used to accomplish almost anything a normal cell phone can – such as SMS text messages.

This module features two UARTS, an SPI interface (**Serial Peripheral Interface Bus)**, and two 10-bit ADCs. It also supports Li-ion battery charging, a 4x6 keypad, and an LCD interface. Inputs/outputs are available for a speaker and microphone. An antenna does come attached to the module.

Power supplied to the module should be regulated between 3.3-4.2VDC (3.6V nominal).

**RF Functionalities**

The RF part of **SM5100B** module converts RF signals to baseband for receiver chain and translates base bandsignals into RF frequency spectrum.

**The operating frequencies are:**

Rx (EGSM 850): 869 to 894MHz

Tx (EGSM 850): 824 to 849MHz

Rx (EGSM 900): 925 to 960MHz

Tx (EGSM 900): 880 to 915MHz

Rx (DCS 1800): 1805 to 1880MHz

Tx (DCS 1800): 1710 to 1785MHz

Rx (PCS 1900): 1930 to 1990MHz

Tx (PCS 1900): 1850 to 1910MHz

### 5.3 Main Components of Cellular Shield:

- 60-pin SM5100B connector,
- a SIM card socket, and
- SPX29302 voltage regulator configured to regulate the Arduino's raw voltage to 3.8V.
- The board's red LED indicates power. The Arduino's reset button is also brought out on the shield.

### 5.4 Schematic Diagram of Cellular Shield

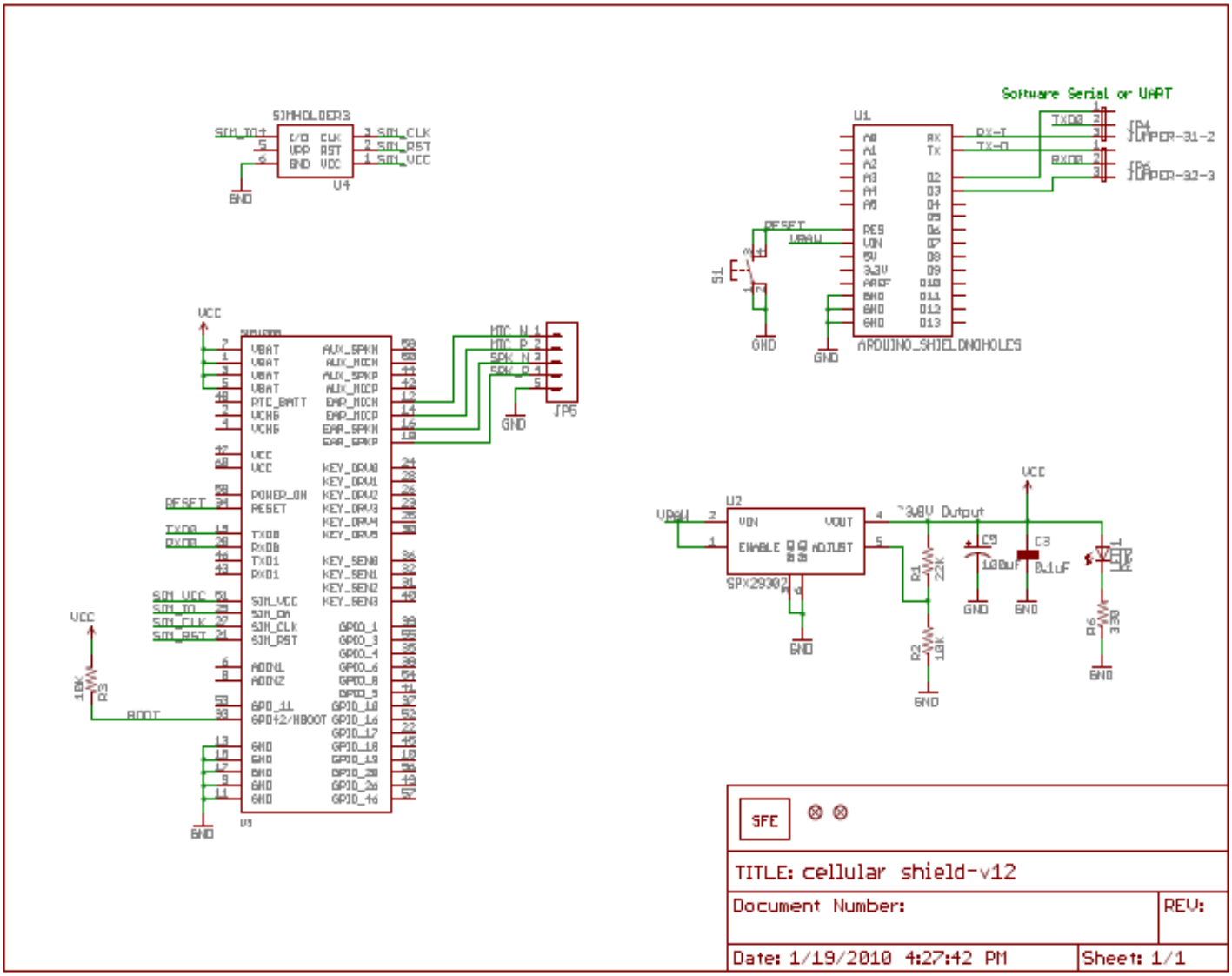



**Figure-5.1: Schematic Diagram of Cellular Shield**

### 5.5SM5100B-D GSM/GPRS Module Specification (Preliminary)

| | |
|---|---|
| Connection | 60 pins |
| Power supply | VBAT: 3.3V to 4.2V range, 3.6V typical. |
| Power consumption | Off mode: <100uA<br>Sleep mode: <2.0mA<br>Idle mode: <7.0mA (average)<br>Communication mode: 350 mA (average,GSM)<br>Communication mode: 2000mA (Typical peak during TX slot,GSM) |
| Li-ion Battery charging management and interface (OPTION) | Li-ion Battery charging management is included. The charger Interface is provided on 60-pin connector<br>. (only for 3.7V Li-ion Battery) |
| Frequency bands | EGSM900 +GSM850+ DCS1800+PCS1900 |
| Transmit power | Class 4 (2W) for EGSM900/GSM850<br>Class 1 (1W) for DCS1800/PCS1900 |
| Supported SIM card | 3V/1.8V SIM card. (auto recognise) |
| Keyboard interface | 4x6 keyboard interface is provided |
| UART0 interface with flow control | Up to 460 kbps<br>Full hardware flow control signals (+3.0V) are provided on 60 pins. |
| UART1 interface without flow control | 2-Wire UART interface<br>Up to 460 kbps |
| LCD interface | Support standard SPI interface, |
| Temperature range | Normal range: -10°C to +55°C (full compliant)<br>Storage: -40°C to +85°C |
| Weight | < 9g |
| Physical dimensions | 35.0X39.0X2.9 mm (typical) |

### 5.7 Power Consumption

| Mode | Condition | I(Typ) | I(Max) |
|---|---|---|---|
| Off mode | off | 1.6 A | 2 A |
| GSM850/900mode | During TX bursts @PCL5* | 100 mA | 130 mA |
| GSM850/900 mode | During RX bursts | 350 mA | 400 mA |
| GSM850/900 mode | Average @PCL5* | 250 mA | 300 mA |
| GSM850/900 mode | Average @PCL8* | 250 mA | 20 mA |
| GSM850/900 mode | Average idle mode | 13 mA | 20 mA |

### 5.8 Connection with Arduino Mega 2560

Jumper wire from GSM shield pin D2 to Mega pin 19,

Jumper from GSM shield pin D3 to Mega pin 18, as shown figure 5.1 next page

Figure 5.1 in next page indicates how the connection is done.

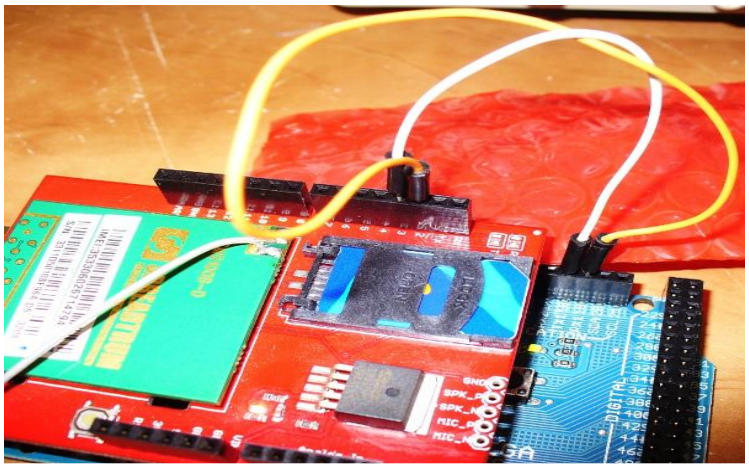


**Figure-5.1: Cellular ShieldConnection with Arduino Mega 2560**

# Chapter 6: Set up,Testing and Results

## 6.1 Setup and Connection

This test is to check the integration of RFID reader, LED matrix, servo motor, and IR sensor which are essential for a reliable operation of our parking monitoring system.

For this test, the code in appendix A was used by sending it to the Arduino using Arduinoprogram. Thefollowing connections shown in the figures below were made:

### 6.1.1RFID readers

Both RFID readers have been tested after the set up shown in figure 6.1 next page

#### At the entry:

Pin 1, 7 are connected to the ground at the Arduino

Pin 2, 11 are connected to the +5V DC at the Arduino

Pin 9 is the serial communication connected to the configured RX3 at the Arduino

#### At the exit:

Pin 1, 7 are connected to the ground at the Arduino

Pin 2, 11 are connected to the +5V DC at the Arduino

Pin 9 is the serial communication connected to the configured RX2 at the Arduino

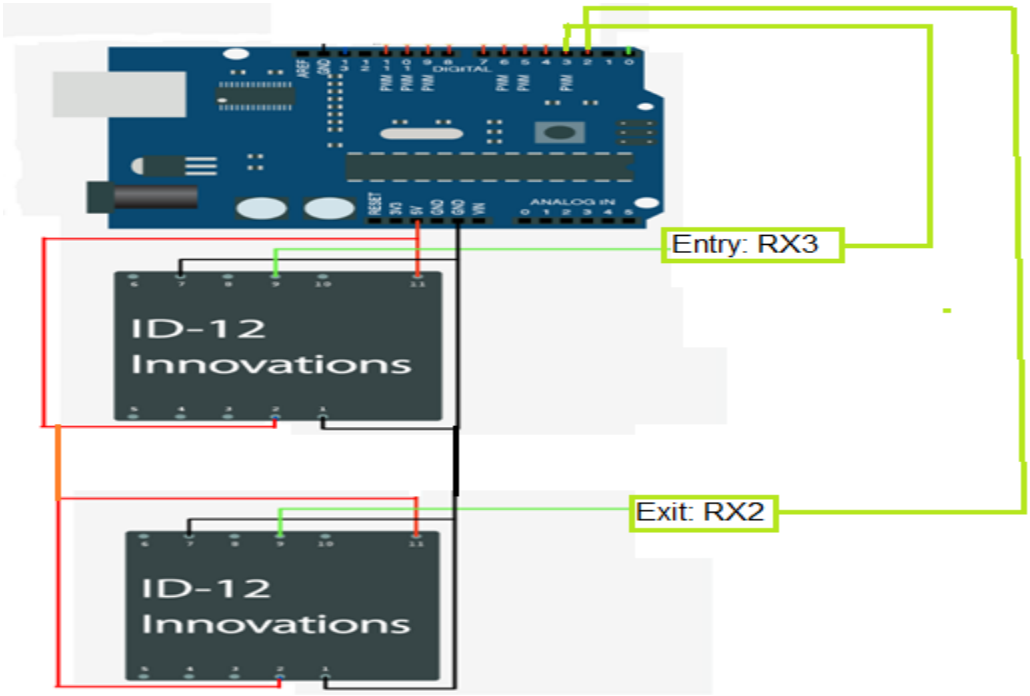



**Figure-6.1 : Entry, Exit RFID ID-12 Readers and Arduino connection**

## 6.1.2LED Display

The interface test of LED display with Arduino UNO was done after the following connection has been made as shown in figure 6.2 next page

VCC (red) to the +5Vat Arduino Board

Ground (black) to ground at the Arduino

DATA(orange) tothe digital pin 2 (As configured in the program)

WRITE (yellow) to the digital pin 3 (As configured in the program)

CS0 (white) to the digital pin 4 (As configured in the program)

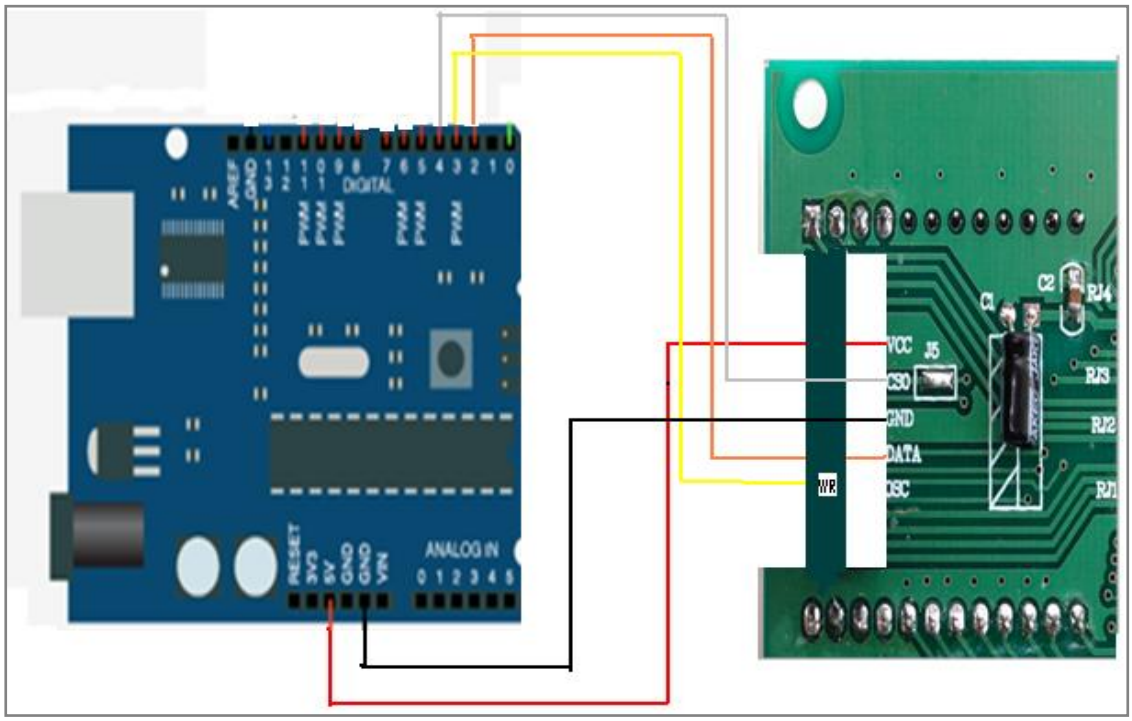



**Figure-6.2 : Connection between LED Matrix and Arduino**

### 6.1.3Servo Motor

At the entry:

Black wire is connected to the ground at the Arduino

Red wire connected to the +5V DC at the Arduino

White wire in the motor is the pulse and it is connected to the digital output in Arduino configured at pin 10

At the exit:

Black wire is connected to the ground at the Arduino

Red wire connected to the +5V DC at the Arduino

White wire in the motor is the pulse and it is connected to the digital output in Arduino configured at pin 9

See figure 6.3 for connecting two servo motor with Arduino

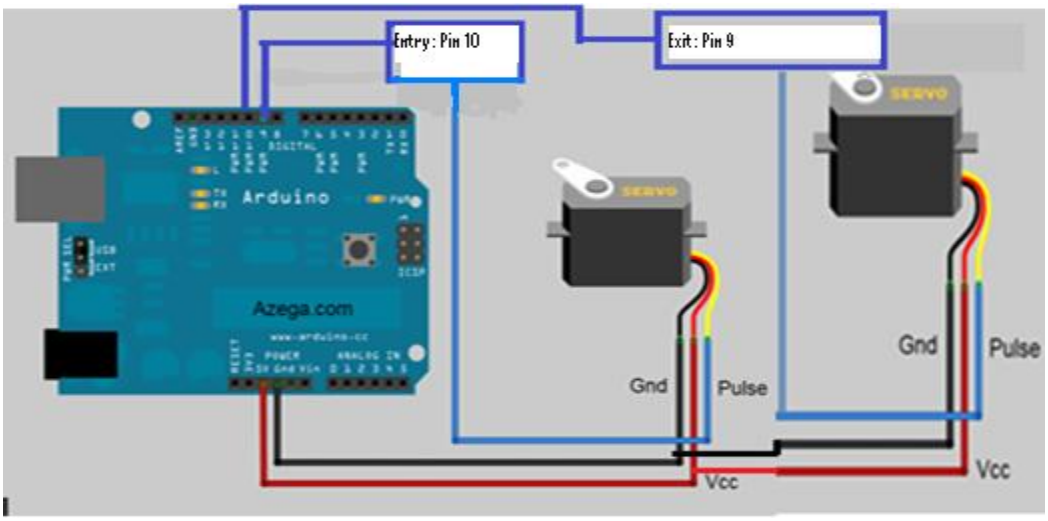



**Figure-6.3 : Connection between Entry, Exit Servo Motor and Arduino**

## 6.1.4Sharp IR Sensor

At the entry:

Black wire is connected to the ground at the Arduino

Red wire connected to the +5V DC at the Arduino

Yellow wire in the IR is the analog reading and it is connected to analog input in Arduino configured at pin A2

At the exit:

Black wire is connected to the ground at the Arduino

Red wire connected to the +5V DC at the Arduino

Yellow wire in the IR is the analog reading and it is connected to analog input in Arduino configured at pin A1

Figure 6.4 below indicates the connection of two IR sensors with Arduino.

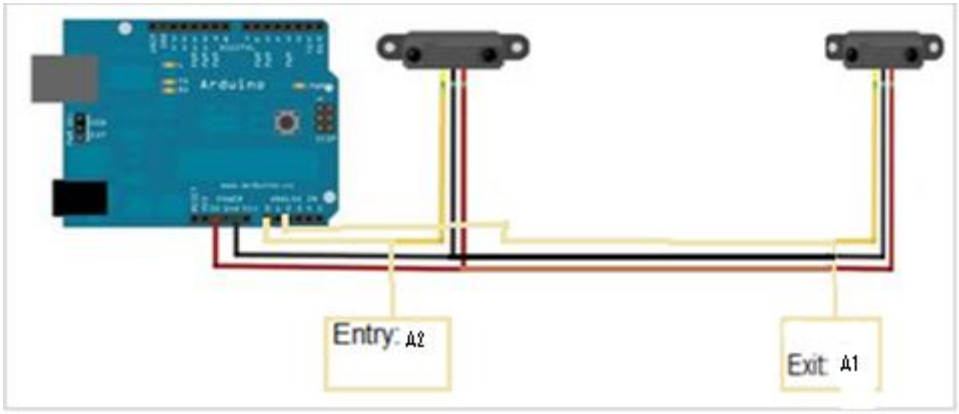



**Figure-6.4 : Connection between Entry, Exit IR Sensor and Arduino**

## 6.2Results of Testing of RFID Readers Circuit

After making the above connection and setup, the parking monitoring system was tested with successful results as follows:

1. At the moment of sending the code to the Arduino, The display shows 10 free parkings. See fiigre 6.5

**Figure-6.5 : Free 10 Display**

4. If an employee tag is scanned , the counter will be decremented and the gate will open as indicated in the figure 6.6

**Figure-6.6 : Employee Display and gate opening**

2. The gate does not close until the car gets in.

3. If student tag was scanned , the gate will open, and the counter will be decremented as indicated in figure 6.7.

**Figure-6.7: Student Display and gate opening**

4. The gate does not close until the car gets in.
5. If the car left the parking, the gate will open and the counter will be incremented.
6. The gate does not close until the car gets out.

7. If the parking is full, then display will show “Full” as shown in figurer 6.8.

**Figure-6.8 : Full Display**

8. If the person is not authorized, the display will be show”Not Auth ” as in figure 6.9.

**Figure-6.9: Not Authorized Display**

9. If there is a visitor, then the gate can be opened using push button and that will be shown in the display and recorded in the database.

## 6.3C#,Testingof Interfacing with Arduino Microcontroller

This test is to check the interfacing between the C# program with the microcontroller connected with RFID readers, servo motors and LED matrix.Refer to figure 6.10,figure 6.11, figure 6.12 and figure 6.13.

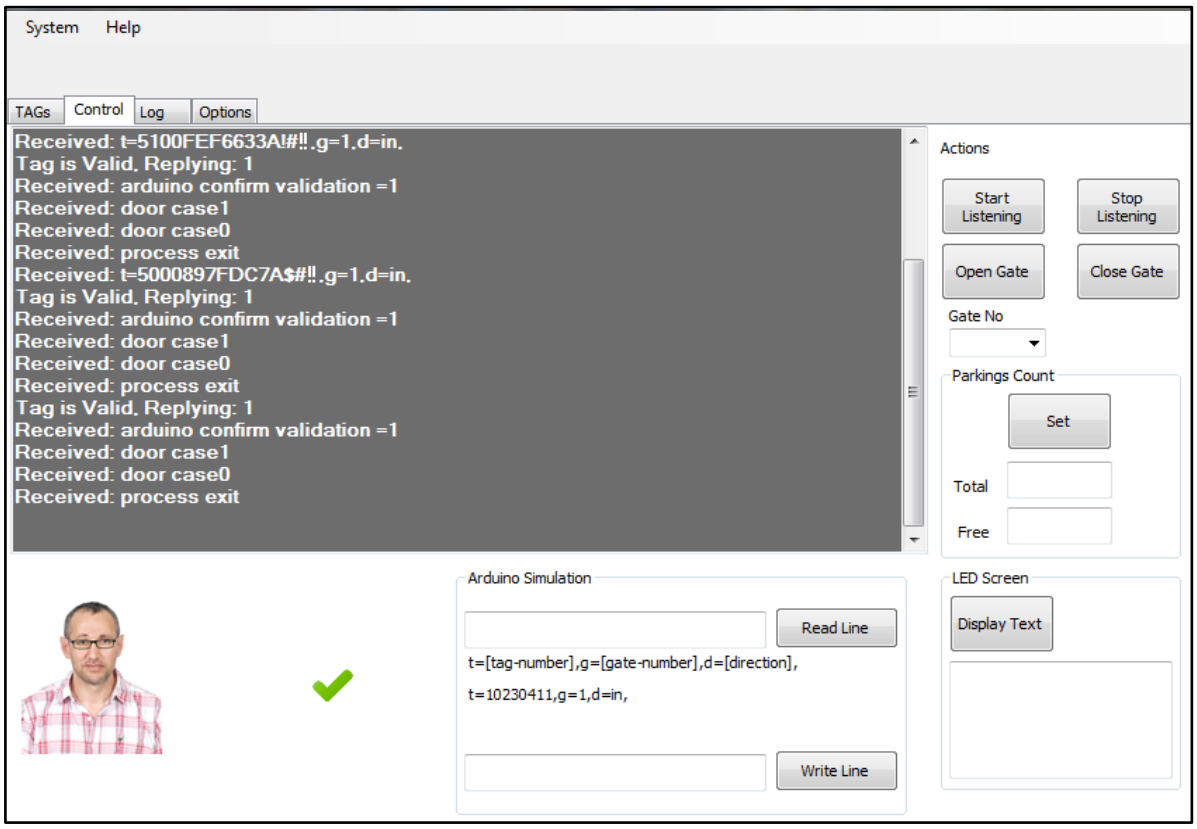



**Figure-6.10 : Control Screen Test With A Valid REID Tag User.**

The above shows our Dr.Samir Elnakla and showed that he is authorized and his membership is vlaid to enter the parking.

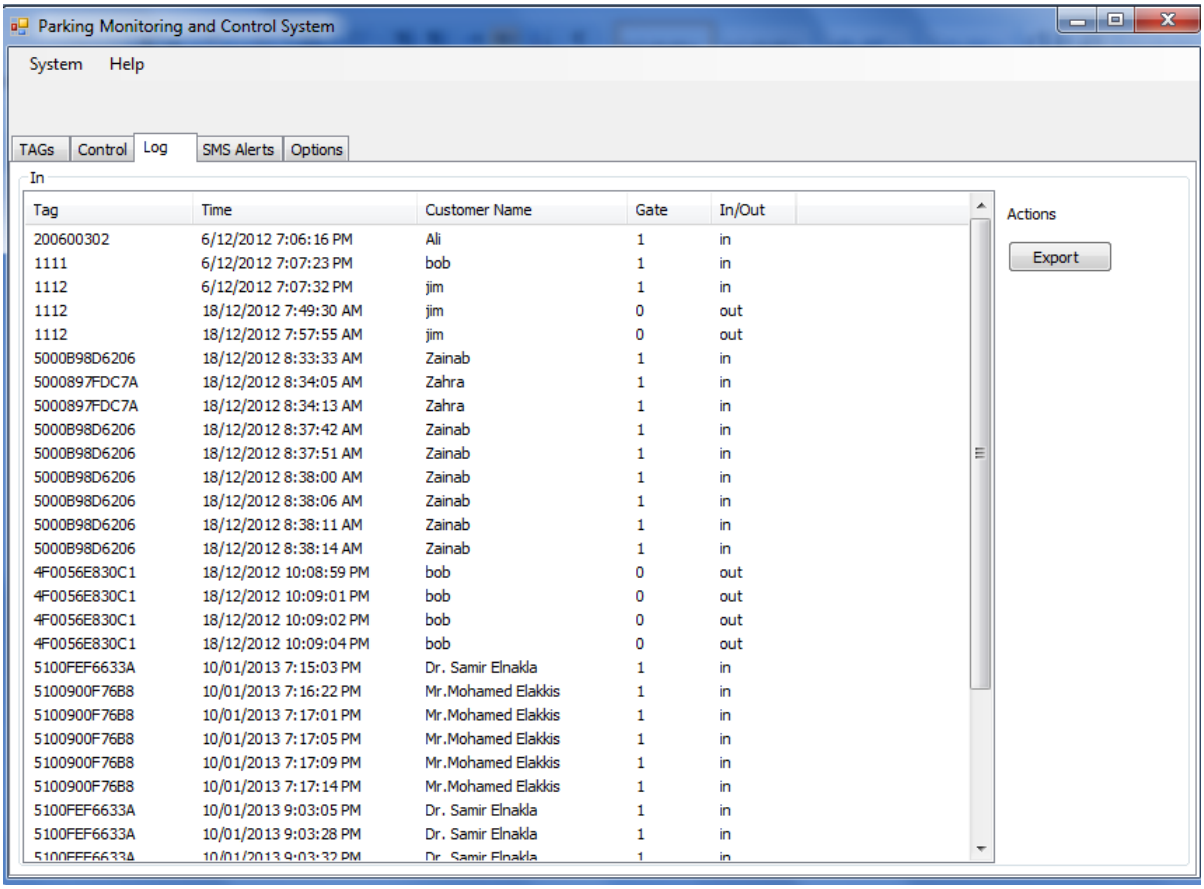



**Figure-6.11 : Historical log and Export option**

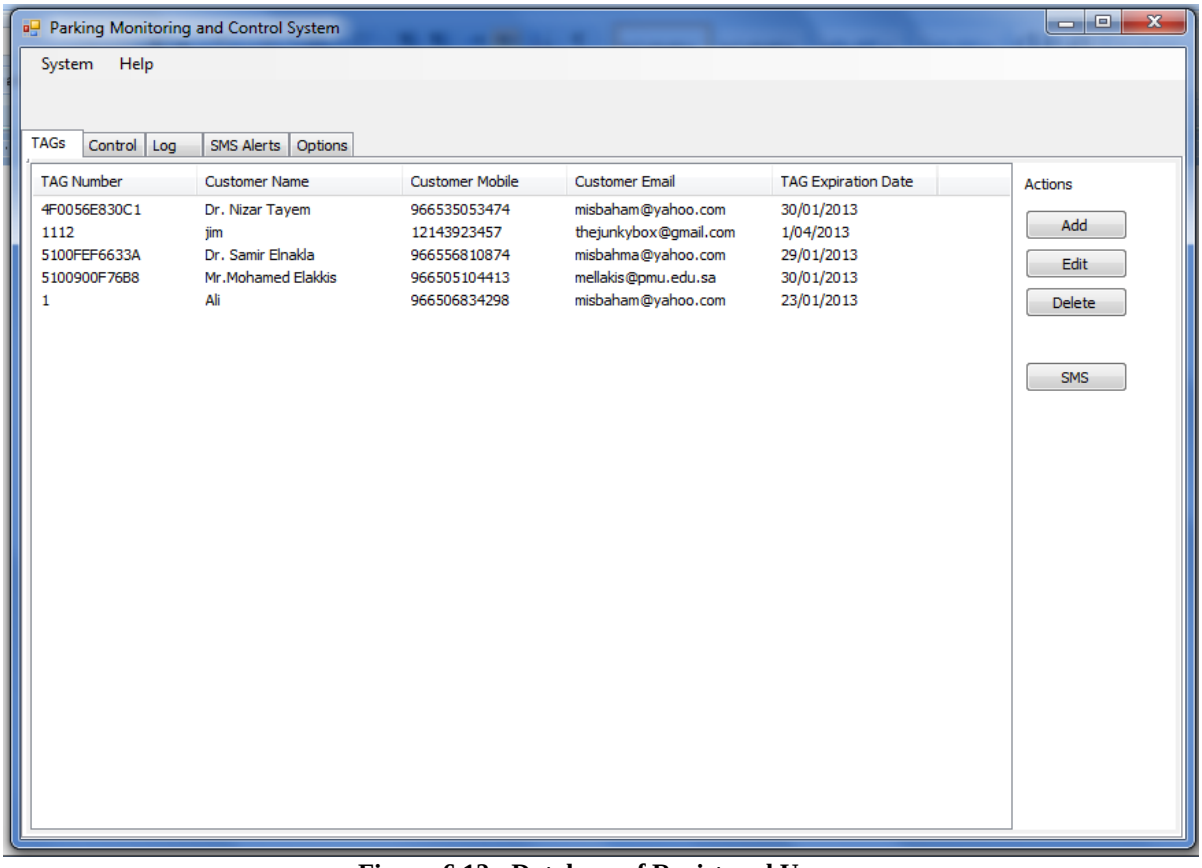



**Figure-6.12 : Database of Registered Users**

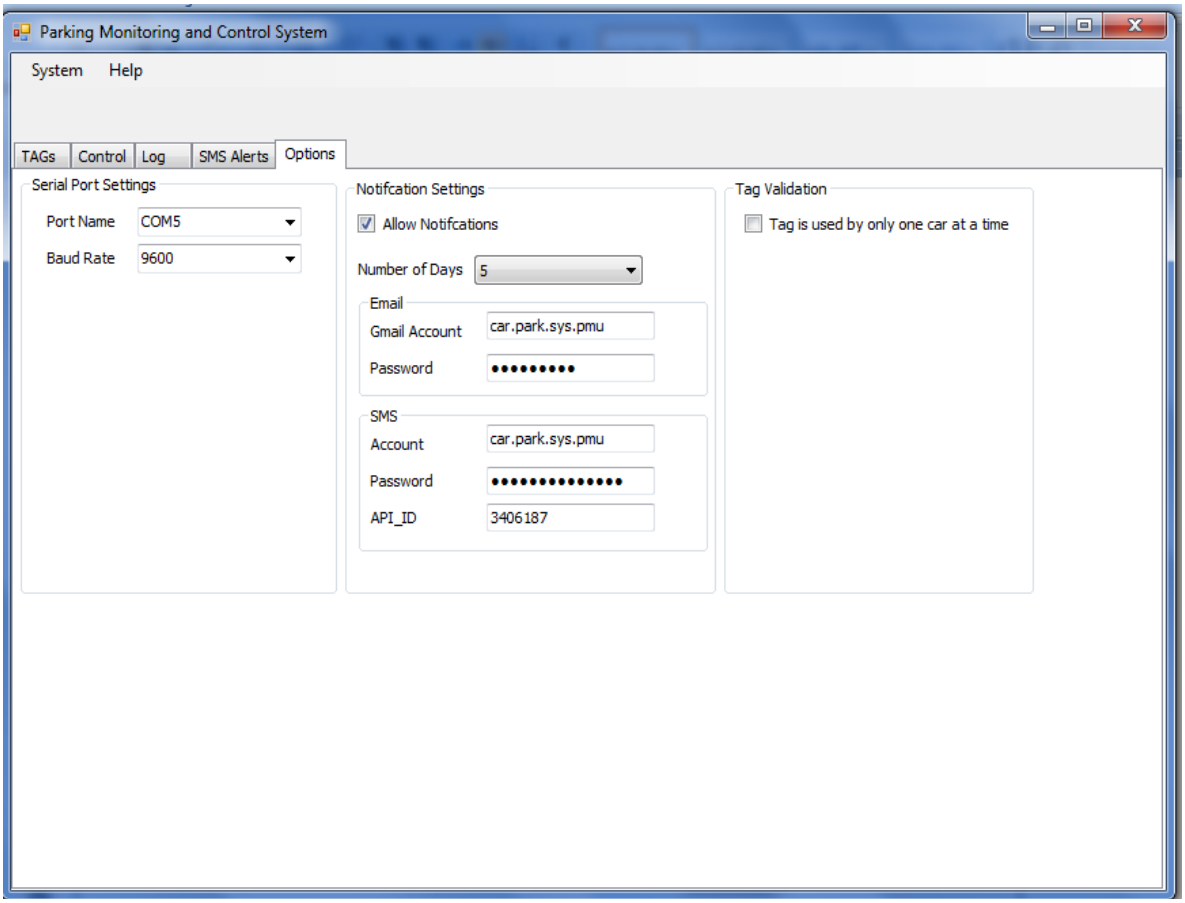



**Figure-6.12 : Option Tab**

Regarding E-mail notification, the registered user on the database and automatic e-mail notification will be sent to the user when his membership is soon to expire. The below figure shows an example of the email notification sent through customer's email.

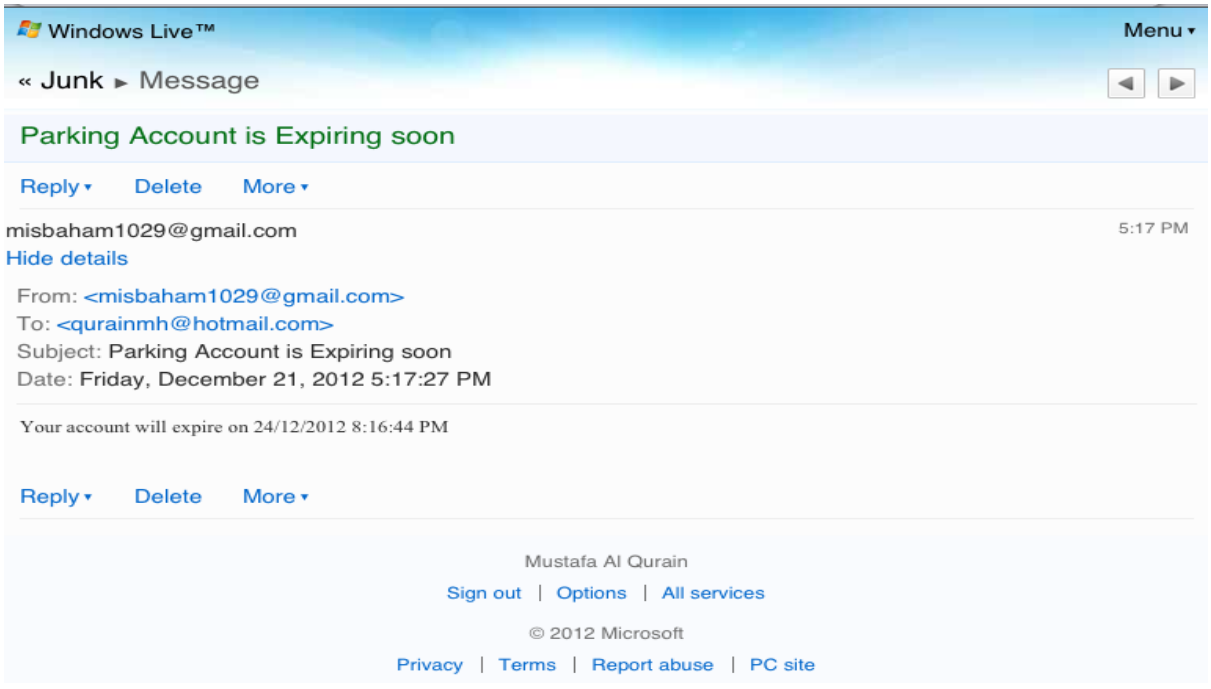



**Figure-6.13 : Option Screen With Sending Notifcation Email (B)**

However, the latest test conducted on C# are shown in the figures below. But now these include a mojor feature of the system which is sending SMS notifications upon the entry of the car park user.According to the type of SMS subscriptions the user is subscirbed to, the user recieves SMS notification about events, car park regulations, renewing membership or any message intended to be sent.

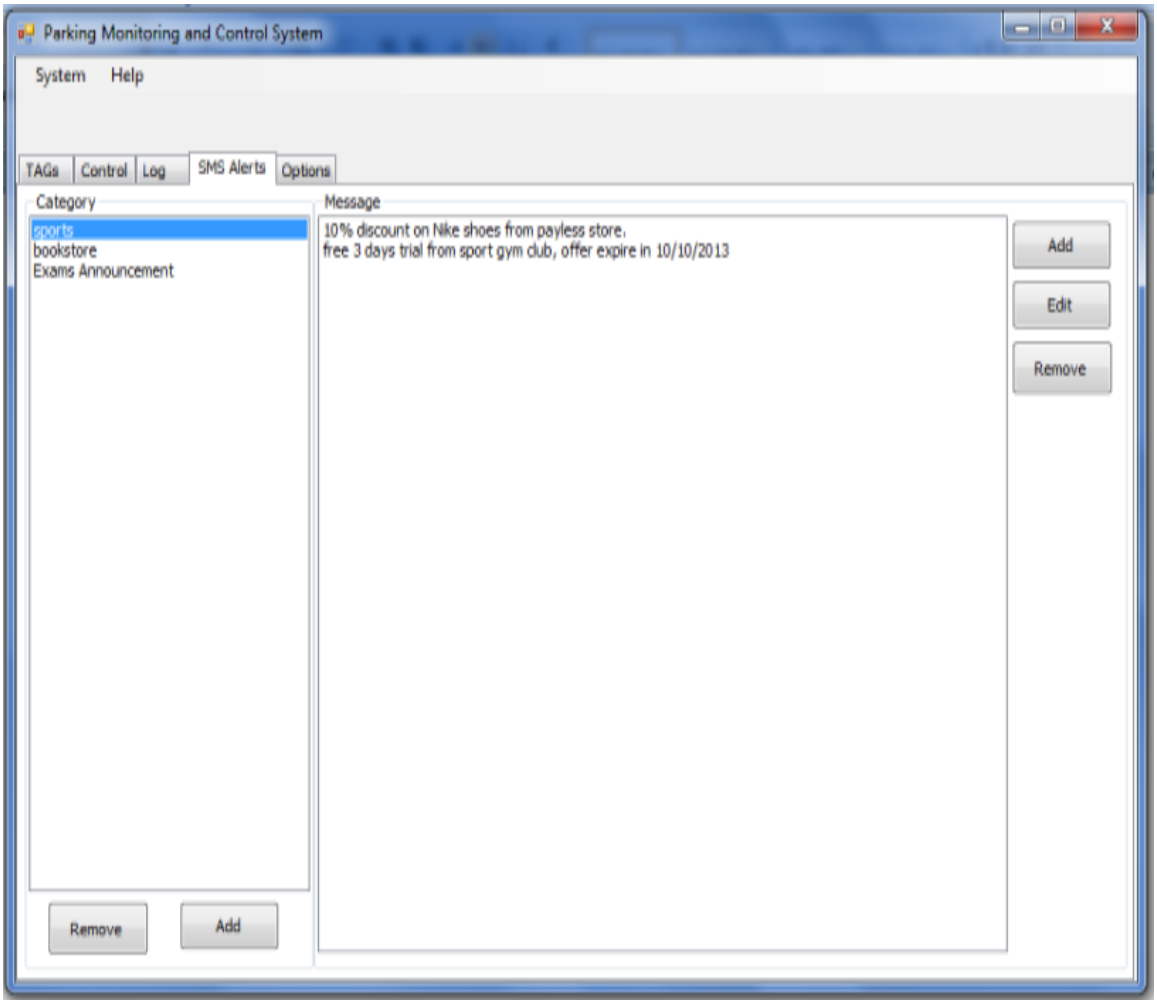



Figure-6.14 : SMS Alert Subscription

The blow figures 6.15 shows the entries for adding or editing a user which includes RFID tag, customer's name, mobile, email, expiration date, photo and the SMS subscriptiions.

The advantage of using this technique of SMS subscription is the flexibility to send any message for any purpose to the users of the parking system.

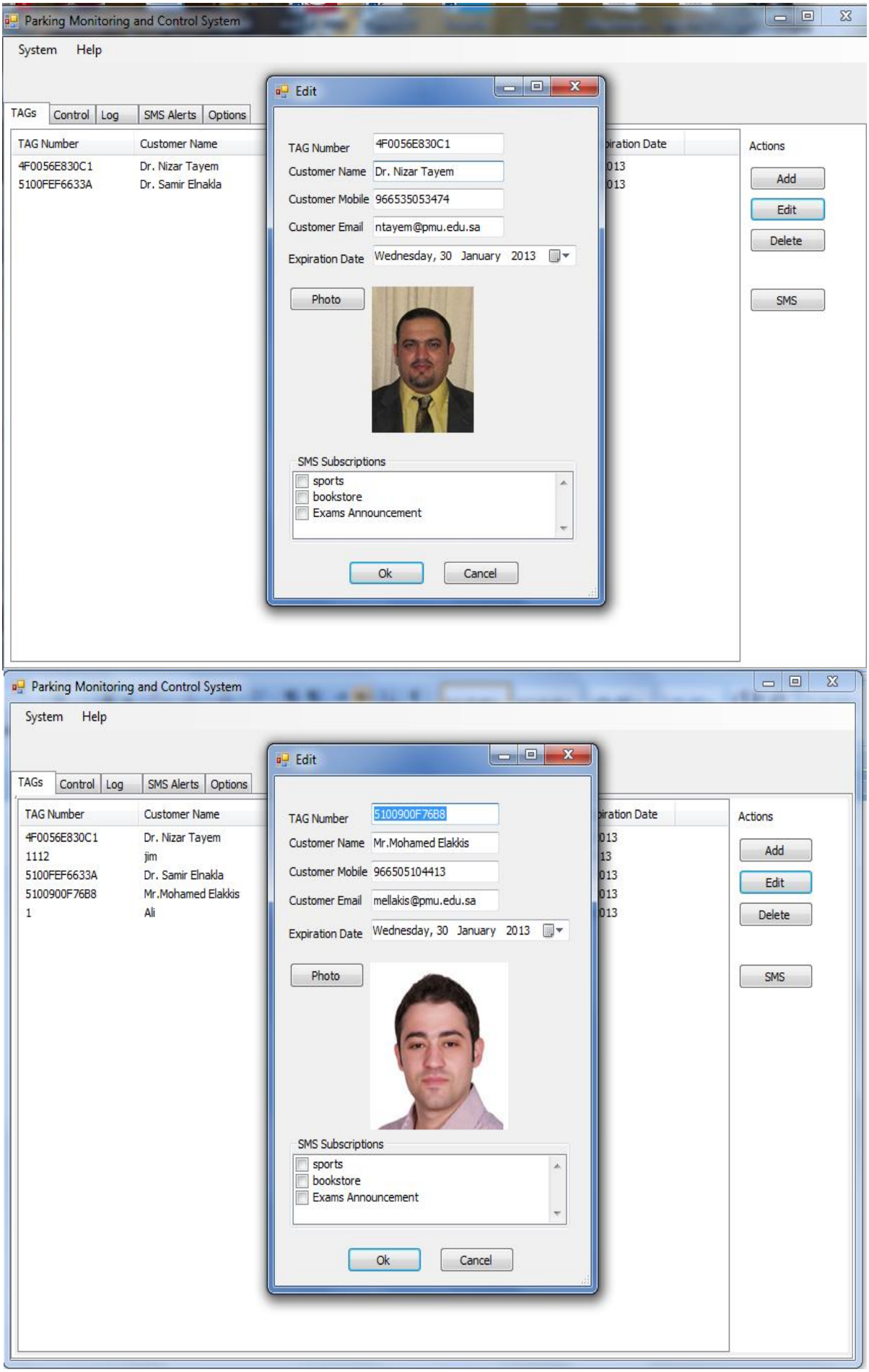



**Figure 6.15: Two samples of customers' info registered in the database**

# Chapter 7: Difficulties, Refinement, and Conclusion

## 7.1 Difficulties and Obstacles

Many challenges were face in the project such as:

- Learning Arduino programming languages which is based special programming language in addition to C.
- Learning database programming language (C#) which mainly based on Java and C or C++. This is merely for software or IT professionals but taking the challenge was overwhelming and interesting but yet difficult with slow progress.
- Solving the conflicts between RFID and LED matrix when connecting them with Arduino.
- Controlling the motor forward and reverse rotation using DC motor; however, finally servomotors were used since programing codes only needed without complicated hardware design.

## 7.2 Product Refinement

In circuit design, itwas decided to change the DC motor with the servo motor type. That is due to its simplicity and reliabilit. Unlike DC motor, it does not require using H-bridge or transistor for smooth operation achievement.

## 7.3Conclusions

In the conclusion we say:

1. The course was interesting and challenging to us. It was interesting to use the knowledge you earned to build a real circuit which is based on theories and engineering problems. One of the most important challenges that faced is the little experience in dealing with C language and C# language in addition to electronic circuits. Another challenge was selecting the appropriate components for project such as motors and its availability in the market.
2. Performing functional test of the electronic components is important prior to building the circuit and the testing. Also, having testing equipment in the weekend could have helped a lot in avoiding delays.

3. During undergraduate studies, there is no adequate skills and knowledge about microcontroller languages, operation, and design. This is why most of the project time was focusing on programming the Arduino and creating the data base which required extra efforts and more time.

4. The group tasks list were divided as follows:

- Power Supply: Ali and Mustafa
- RFID, IR, servo, and Display: Muneer and Salman
- Arduino hardware: All group
- Arduino software (creating the parking system code): Muneer and Salman
- Data base (C#): Ali and Mustafa

However, teamwork requires that everybody to contribute in every subsystem. In fact, every member spent effort, has a role in each subsystem, and gained the knowledge about the whole project subsystems. So, the actual work list can sometimes change according to the needs and other factors as well.

# Chapter 8: Real Implementation (Future Work)

In the real implementation of the parking system, we will be seeking cost effective and reliable product which will guide drivers into the nearest parking space automatically;

To achieve this, we would use:

- A camera for more security precautions. The camera costs $75 (amazon)**Wireless G Network Camera DCS 920**
- Ultrasonic sensor since it has longer range of detection and more accurate under sun light. Ultrasonic detector type KEYTOP


**Figure 8.1: KEYTOP Ultrasonic Detecting Sensor**

| | |
|---|---|
| **Operating Voltage** | **DC 24V** |
| **Operating Current** | **10mA** |
| **Power** | **0.24W** |
| **Communication Method** | **RS485** |
| **Detecting Distance** | **0.1~3.5m** |
| **Communication Distance** | **≤1000m** |
| **Communication Rate** | **4800bps** |
| **Operating Temperature** | **-20°C~80°C** |

**Table 8.1: KEYTOP Ultrasonic Detecting Sensor Specification**

- The LED display as shown in the figure below will be installed at the main entrances of car park and the intersections of the main channels. In other places, it can be installed according to the needs. Display the quantity of the current free parking spaces in the connected region in the form of number and text. It can be 24-hour used. The internal procedures can be modified at any time according to the customers' demands to display the other needed information of the users.


**Figure 8.2: LED Display**

| | |
|---|---|
| Operating Voltage | DC5V, 5A |
| Panel size | 36cm*12cm |
| Dot matrix | 48*16 |
| Luminance | 300cd/m2 |

**Table 8.2: LED Display Specification**

- Central controller unit as shown in the figure below can handle huge amounts of information wirelessly. Look at the table below for more specifications about CCU.


**Figure 8.3: GCCU (General Central Control Unit)**

| | |
|---|---|
| **Operating Voltage** | **DC 5V** |
| **Operating current** | **375 mA** |
| **Power** | **1.875W** |
| **Communication way** | **RS485,LAN,wireless** |
| **Communication Distance** | **≤1000m** |
| **Maximum Number of Connections** | **64 node controllers** |
| **Operating Temperature** | **-20°C~+80°C** |

**Table 8.3: GCCU Specifications**

1- Parking guidance system type SP2-114. The SP2-114 signal parking guidance sensor is use for signalling car parks vacant spaces designed for very busy car parks, regardless of size. It is especially advantageous for multi-storey car parks where motorists often find themselves driving round in vain to find an empty space. The parking system ensures they will find one using guidance sensor directly to it in real time, by floor, zone and lane.

2- 110 series AC servo motor type Hitec which can withstand high temperature environment of -20 to 60 degree centigrade.

3- 120/220V relay connected with Arduino to control AC motors that are proposed to be used for gates

# 7 Bibliography


1. Neamen, D. A. (3rd Edition). Microelectronics, Circuit Analysis and Design.
2. Ultrasonic VS infrared sensors, 2009,http://bildr.org/2011/02/rfid-arduino/
3. https://www.sparkfun.com/products/8419?
4. http://www.sparkfun.com/datasheets/Sensors/ID-12-Datasheet.pdf
5. http://www.mobiusconsulting.com/papers/HFE0805_RFIDTutorial.pdf
6. http://ladyada.net/products/16x24LEDmatrix/
7. http://www.zebra.com/id/zebra/na/en/documentlibrary/manuals/en/rfid_programming_guide_P1026148-001.File.tmp/P1026148-001.pdf?dvar1=ProductLiterature&dvar2=RFID%20Programming%20Guide%20(P1026148-001)%20(en)&dvar3=10662
8. http://csrc.nist.gov/publications/nistpubs/800-98/SP800-98_RFID-2007.pdf
9. (http://arduino.cc/en/Tutorial/HomePage
10. (http://www.jeremyblum.com/category/arduino-tutorials/)
11. (http://www.adafruit.com/tutorials)
12. https://www.sparkfun.com/products/10782
13. http://download.cnet.com/Personal-C-Sharp/3000-10247_4-10509460.html
14. http://www.codeproject.com/Articles/473828/Arduino-Csharp-and-serial-interface
15. http://arduino.cc/forum/index.php/topic%2c131445.0.html

# 14 Appendix A: Code of Arduino Software

```
#include <Servo.h>
#include "HT1632.h"

#define DATA 2
#define WR   3
#define CS   4
#define CS2  5

//int max =10;
//max count of cars
int count = 10;

//mssage id to write on the screen
int scrwrt =0;
int newtag =0;

int entrybutton =8;
int exitbutton =9;
int button_status =0;
int oldbtn_status=0;
int exitbtn_status=0;
int oldexitbtn_status=0;

int ledpin=7;

// vars to controle when counter goes less then zero or up more than ten
int fl=0;
int fr=0;
// use this line for single matrix
HT1632LEDMatrix matrix = HT1632LEDMatrix(DATA, WR, CS);

int RFIDResetPin = 13;

//Register your RFID tags here
// there is no need for those tags when we connect to computer because we will find tags saved to computer later
char tag1[16] = "5100FEF6633A";
char tag2[16] = "4F0056E830C1";

//non used yet vars
int maxcars = 12;
int updown=0;
int oldexittag=0;
int oldentrytag=0;


//doors servo variable area
int open_door=1;
int close_door=0;
```

```
int entry_pin=10;
Servo entry_door;
int exit_pin=11;
Servo exit_door;

//this is ir distance
int ir_entry_pin = 1;
int ir_exit_pin = 2;
float ir_free_dsitance=40;//20 cm this is brown area Moneer


//doorcase
int entry_door_case=0; // 0 closed, 1= open
int exit_door_Case=0;


void setup(){
//init debugging and computer connection serial
Serial.begin(9600);
//init serials for two rfids please salman
Serial2.begin(9600);
Serial3.begin(9600);
//send count to computer
Serial.println(count);
scrwrt=10;

//init servo motors
 init_door(entry_door, entry_pin);
 init_door(exit_door, exit_pin);

// call init the ledscr procedure from its library
matrix.begin(HT1632_COMMON_16NMOS);
// fill all dots for 500 ms
matrix.fillScreen();
delay(500);

//clear led screen
matrix.clearScreen();

pinMode(entrybutton, INPUT);
pinMode(exitbutton, INPUT);

pinMode(ledpin, OUTPUT);
//initrfid
pinMode(RFIDResetPin, OUTPUT);
digitalWrite(RFIDResetPin, HIGH);

//write to led sceen
matrix.print("on..");
// update the led screen
matrix.writeScreen();
```

```
button_status=digitalRead(entrybutton);
exitbtn_status=digitalRead(exitbutton);

}

int release1=0;
int release2=0;

void loop(){
char tagString[13];
int index = 0;
boolean reading = false;


 button_status=digitalRead(entrybutton);
 exitbtn_status =digitalRead(exitbutton);


 if ((button_status ==1) && (oldbtn_status==0))
     {
       oldbtn_status = button_status;
       digitalWrite(ledpin, HIGH);
     }

 else if ((button_status ==0) && (oldbtn_status==1))
     {
       oldbtn_status = button_status;
       digitalWrite(ledpin, 0);
       release1 = 1;
       //process_entry();
       scrwrt =8;
       updown=-1;
       count +=updown;
       if ((updown<0)&&(count==0))
       {
       scrwrt=13;
       }
       else if((updown>0)&&(count==10))
       {
       scrwrt=10;
       }
     }

 if ((exitbtn_status ==1) && (oldexitbtn_status==0))
     {
       oldexitbtn_status = exitbtn_status;
       digitalWrite(ledpin, HIGH);
     }

 else if ((exitbtn_status ==0) && (oldexitbtn_status==1))
     {
```

```
        oldexitbtn_status = exitbtn_status;
        digitalWrite(ledpin, 0);
        release2 = 1;
        scrwrt =8;
        updown = 1;
        count +=updown;
        if ((updown<0)&&(count==0))
        {
        scrwrt=13;
        }
        else if((updown>0)&&(count==10))
        {
        scrwrt=10;
        }
      }

 // test if there is a message we want to output
if (scrwrt> 0)
  {
  // scrwrt==10 raise when park is free and contain 0 cars
if (scrwrt==10)
  {
if (fr==0) //
    {
rstscr();  // see the comment near the procedure it self
matrix.print("Free");
matrix.writeScreen();
matrix.setCursor(7, 8);   // next line, 8 pixels down
matrix.writeScreen();
matrix.print(count);
matrix.writeScreen();
fr++; // fr prevent counter goes less than 0
 if (updown>0) {
 process_exit();
 release2=0;
Serial.println("process entry");

}

  }
  }
if (scrwrt==13)
  {
if(fl==0)
    {
rstscr();
matrix.setCursor(0, 4);   // next line, 8 pixels down
matrix.writeScreen();
matrix.print("Full");
matrix.writeScreen();
fl++;//fl prevent counter goes more than 10
```

```
count=0;

if (updown<0) {

  process_entry();
  Serial.println("process exit");
  release1=0;
}
//else
//{
//
//  process_exit();
// release2=0;
//Serial.println("process entry");
//}
    }

  }

if (scrwrt==1)// non vaild tag
  {
fl=0;
fr=0;
rstscr();
matrix.print("Not");
matrix.writeScreen();
matrix.setCursor(0, 8);   // next line, 8 pixels down
matrix.writeScreen();
matrix.print("Auth");
matrix.writeScreen();
delay(1000);
rstscr();  // see the comment near the procedure it self
matrix.print("Free");
matrix.writeScreen();
matrix.setCursor(7, 8);   // next line, 8 pixels down
matrix.writeScreen();
matrix.print(count);
matrix.writeScreen();

  }
if (scrwrt==2&&count<10)         // valid tag received
  {
fl=0;
fr=0;
rstscr();

matrix.print("stu:");
matrix.writeScreen();
matrix.setCursor(9, 8);   // next line, 8 pixels down
matrix.writeScreen();
matrix.print(count);
matrix.writeScreen();
```

```
//open or close the door
if (updown<0) {

  process_entry();
  Serial.println("process exit");
  release1=0;
}
else if (updown>0)
{

  process_exit();
 release2=0;
Serial.println("process entry");
}
else
{
 scrwrt=10;
}
  }
if (scrwrt==3&&count<10) // vaild tag emp recieved
  {
fl=0;
fr=0;
rstscr();

matrix.print("emp:");
matrix.writeScreen();
matrix.setCursor(9, 8);   // next line, 8 pixels down
matrix.writeScreen();
matrix.print(count);
matrix.writeScreen();

//open or close the door
if (updown<0) {
  process_entry();
  Serial.println("process exit");
  release1=0;
}
else if (updown>0)
{
  process_exit();

 release2=0;
Serial.println("process entry");
}
  else {
 scrwrt=10;
  }


  }
```

```
if (scrwrt==4) // valid tag received
  {
fl=0;
fr=0;
rstscr();

matrix.print("Vis:");
matrix.writeScreen();
matrix.setCursor(9, 8);   // next line, 8 pixels down
matrix.writeScreen();
matrix.print(count);
matrix.writeScreen();

//open or close the door
if (updown<0) {

  process_entry();
  Serial.println("process exit");
  release1=0;
}
else
{

  process_exit();
  release2=0;
Serial.println("process entry");
}
  }
  if (scrwrt==8) // valid tag received
  {
fl=0;
fr=0;
rstscr();

matrix.print("Vis:");
matrix.writeScreen();
matrix.setCursor(9, 8);   // next line, 8 pixels down
matrix.writeScreen();
matrix.print(count);
matrix.writeScreen();

//open or close the door
if (updown<0) {

  process_entry();
  Serial.println("process exit");
  release1=0;
}
else
{

  process_exit();
```

```
  release2=0;
Serial.println("process entry");
}
  }
scrwrt=0;
  }

clearTag(tagString); //Clear the char of all value
// if there is tag near serial2 reader catch it
while(Serial2.available()){

int readByte = Serial2.read(); //read next available byte Entry
if(readByte == 2) reading = true; //begining of tag
if(readByte == 3) reading = false; //end of tag

if(reading &&readByte != 2 &&readByte != 10 &&readByte != 13){
     //store the tag
tagString[index] = readByte;
index ++;
   }
//scrwrt=1; //put message
updown=-1; // the tag is for entry so updown is positave
  }


// if there is tag near serial3 rfid catch it Exit
while(Serial3.available()){

int readByte = Serial3.read(); //read next available byte

if(readByte == 2) reading = true; //begining of tag
if(readByte == 3) reading = false; //end of tag

if(reading &&readByte != 2 &&readByte != 10 &&readByte != 13){
     //store the tag
tagString[index] = readByte;
index ++;
   }
//scrwrt=1;

updown=1;// the tag is for exit  so updown is positave

}
checkTag(tagString); //Check if it is a match

resetReader(); //reset the RFID reader

}
// if new tag came it must processed
void checkTag(char tag[]){
if((strlen(tag) == 0)&& (release1==0 ))
return; //empty, no need to continue
```

```
if ((updown<0)&&(count==1))
{
scrwrt=13;// FULL
}
else if((updown>0)&&(count==10))
{
scrwrt=10; // Free 10
}
else if(compareTag(tag, tag1))
{ // if matched tag1, do this
count +=updown;

// need to open the door or close it
//Serial.println(count);
if (count >0)
{scrwrt =2;// student
}
}

else if(compareTag(tag, tag2))
{ //if matched tag2, do this
count +=updown;
Serial3.println(count);
scrwrt=3;// employee

}

else {
scrwrt=1;
  //Serial.print("unknown");
//Serial.println(tag);
  }
}

void lightLED(int pin){
/////////////////////////////////////
//Turn on LED on pin "pin" for 250ms
/////////////////////////////////////
Serial.println(pin);

digitalWrite(pin, HIGH);
delay(250);
digitalWrite(pin, LOW);
}

void resetReader(){
/////////////////////////////////////
//Reset the RFID reader to read again.
/////////////////////////////////////
digitalWrite(RFIDResetPin, LOW);
digitalWrite(RFIDResetPin, HIGH);
```

```
delay(150);
}

void clearTag(char one[]){
////////////////////////////////////
//clear the char array by filling with null - ASCII 0
//Will think same tag has been read otherwise
////////////////////////////////////
for(int i = 0; i <strlen(one); i++){
one[i] = 0;
  }
}

boolean compareTag(char one[], char two[]){
////////////////////////////////////
//compare two value to see if same,
//strcmp not working 100% so we do this
////////////////////////////////////

if(strlen(one) == 0) return false; //empty


for(int i = 0; i < 12; i++){
if(one[i] != two[i]) return false;
  }

return true; //no mismatches
}


void rstscr() // clear screen and reset font , cursor position, finally it update the screen
{
matrix.clearScreen();
matrix.setTextSize(1);    // size 1 == 8 pixels high
matrix.setTextColor(1);   // 'lit' LEDs
matrix.setCursor(0, 0);
matrix.writeScreen();

}

void init_door(Servo door, int pin)
{
 door.attach(pin);
  Serial.println();
  Serial.print("door pin ");
  Serial.print(pin);
  Serial.println(" attached");

}

void change_door(Servo door, int door_case)
{
```

```
  Serial.println();
  Serial.print("door case");
  Serial.print(door_case);
  Serial.println();

  if(door_case==0)
 {
   door.write(90);

  }
  else
  {
    door.write(180);
  }

}

int test_ir(int ir_pin)
{
  float volts = analogRead(ir_pin)*0.0048828125;     // value from sensor * (5/1024) - if running 3.3.volts then change 5 to 3.3
  float distance = 65*pow(volts, -1.10);              // worked out from graph 65 = theretical distance
  if (distance<ir_free_dsitance) return 0;
  else return 1;
}

void process_exit()
{
 change_door(exit_door, open_door);
 while (test_ir(ir_exit_pin)>0) ;
 //delay(500);
 change_door(exit_door, close_door);
}

void process_entry()
{
 change_door(entry_door, open_door);
 //delay(500);
 while (test_ir(ir_entry_pin)>0) ;
 change_door(entry_door, close_door);
}
```

# 15 Appendix B: Code of C # (Sharp) Data Base

```
namespace ParkingManager
{
    partial class FormMain
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should be disposed;
otherwise, false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code

        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
            this.components = new System.ComponentModel.Container();
            System.ComponentModel.ComponentResourceManager resources = new
System.ComponentModel.ComponentResourceManager(typeof(FormMain));
            this.panel1 = new System.Windows.Forms.Panel();
            this.menuStrip1 = new System.Windows.Forms.MenuStrip();
            this.systemToolStripMenuItem = new
System.Windows.Forms.ToolStripMenuItem();
            this.openDatabaseToolStripMenuItem = new
System.Windows.Forms.ToolStripMenuItem();
            this.saveDatabaseToolStripMenuItem = new
System.Windows.Forms.ToolStripMenuItem();
            this.exitToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
            this.helpToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
            this.aboutToolStripMenuItem = new
System.Windows.Forms.ToolStripMenuItem();
            this.panel2 = new System.Windows.Forms.Panel();
            this.tabControl1 = new System.Windows.Forms.TabControl();
            this.tabPage1 = new System.Windows.Forms.TabPage();
            this.listViewCustomers = new System.Windows.Forms.ListView();
            this.columnHeader1 = ((System.Windows.Forms.ColumnHeader)(new
System.Windows.Forms.ColumnHeader()));
            this.columnHeader2 = ((System.Windows.Forms.ColumnHeader)(new
System.Windows.Forms.ColumnHeader()));
            this.columnHeader3 = ((System.Windows.Forms.ColumnHeader)(new
System.Windows.Forms.ColumnHeader()));
            this.columnHeader8 = ((System.Windows.Forms.ColumnHeader)(new
System.Windows.Forms.ColumnHeader()));
            this.columnHeader4 = ((System.Windows.Forms.ColumnHeader)(new
System.Windows.Forms.ColumnHeader()));
            this.panel3 = new System.Windows.Forms.Panel();
            this.buttonSMS = new System.Windows.Forms.Button();
```

```
            this.buttonDeleteTag = new System.Windows.Forms.Button();
            this.buttonEditTag = new System.Windows.Forms.Button();
            this.buttonAddTag = new System.Windows.Forms.Button();
            this.label1 = new System.Windows.Forms.Label();
            this.tabPage3 = new System.Windows.Forms.TabPage();
            this.txtCommandsLog = new System.Windows.Forms.TextBox();
            this.panel4 = new System.Windows.Forms.Panel();
            this.groupBox7 = new System.Windows.Forms.GroupBox();
            this.buttonWriteArduino = new System.Windows.Forms.Button();
            this.textBoxWriteArduino = new System.Windows.Forms.TextBox();
            this.label14 = new System.Windows.Forms.Label();
            this.label13 = new System.Windows.Forms.Label();
            this.buttonArduinoWrite = new System.Windows.Forms.Button();
            this.textBoxReadArduino = new System.Windows.Forms.TextBox();
            this.pictureBox2 = new System.Windows.Forms.PictureBox();
            this.pictureBox1 = new System.Windows.Forms.PictureBox();
            this.panel5 = new System.Windows.Forms.Panel();
            this.comboBoxGate = new System.Windows.Forms.ComboBox();
            this.label10 = new System.Windows.Forms.Label();
            this.groupBox3 = new System.Windows.Forms.GroupBox();
            this.label9 = new System.Windows.Forms.Label();
            this.label2 = new System.Windows.Forms.Label();
            this.txtFreeParking = new System.Windows.Forms.TextBox();
            this.txtTotalParkingCount = new System.Windows.Forms.TextBox();
            this.buttonSetParking = new System.Windows.Forms.Button();
            this.groupBox2 = new System.Windows.Forms.GroupBox();
            this.txtLED = new System.Windows.Forms.TextBox();
            this.buttonDisplayLED = new System.Windows.Forms.Button();
            this.buttonCloseGate = new System.Windows.Forms.Button();
            this.buttonOpenGate = new System.Windows.Forms.Button();
            this.buttonStopListening = new System.Windows.Forms.Button();
            this.buttonStartListening = new System.Windows.Forms.Button();
            this.label3 = new System.Windows.Forms.Label();
            this.tabPage2 = new System.Windows.Forms.TabPage();
            this.groupBox5 = new System.Windows.Forms.GroupBox();
            this.listViewLog = new System.Windows.Forms.ListView();
            this.columnHeader5 = ((System.Windows.Forms.ColumnHeader)(new
System.Windows.Forms.ColumnHeader()));
            this.columnHeader10 = ((System.Windows.Forms.ColumnHeader)(new
System.Windows.Forms.ColumnHeader()));
            this.columnHeader6 = ((System.Windows.Forms.ColumnHeader)(new
System.Windows.Forms.ColumnHeader()));
            this.columnHeader7 = ((System.Windows.Forms.ColumnHeader)(new
System.Windows.Forms.ColumnHeader()));
            this.columnHeader9 = ((System.Windows.Forms.ColumnHeader)(new
System.Windows.Forms.ColumnHeader()));
            this.imageList1 = new System.Windows.Forms.ImageList(this.components);
            this.panel6 = new System.Windows.Forms.Panel();
            this.buttonExport = new System.Windows.Forms.Button();
            this.label12 = new System.Windows.Forms.Label();
            this.tabPage4 = new System.Windows.Forms.TabPage();
            this.groupBox6 = new System.Windows.Forms.GroupBox();
            this.checkBoxTag1 = new System.Windows.Forms.CheckBox();
            this.groupBox4 = new System.Windows.Forms.GroupBox();
            this.groupBox9 = new System.Windows.Forms.GroupBox();
            this.textBoxSmsApi = new System.Windows.Forms.TextBox();
            this.label19 = new System.Windows.Forms.Label();
            this.label17 = new System.Windows.Forms.Label();
            this.textBoxSmsPassword = new System.Windows.Forms.TextBox();
            this.label18 = new System.Windows.Forms.Label();
            this.textBoxSmsAccount = new System.Windows.Forms.TextBox();
            this.groupBox8 = new System.Windows.Forms.GroupBox();
            this.label15 = new System.Windows.Forms.Label();
            this.textBoxGmailPassword = new System.Windows.Forms.TextBox();
            this.label16 = new System.Windows.Forms.Label();
```

```
            this.textBoxGmailAccount = new System.Windows.Forms.TextBox();
            this.label11 = new System.Windows.Forms.Label();
            this.comboBoxNotifcations = new System.Windows.Forms.ComboBox();
            this.checkBoxNotifications = new System.Windows.Forms.CheckBox();
            this.groupBox1 = new System.Windows.Forms.GroupBox();
            this.comboBoxStopBits = new System.Windows.Forms.ComboBox();
            this.comboBoxParity = new System.Windows.Forms.ComboBox();
            this.comboBoxDataBits = new System.Windows.Forms.ComboBox();
            this.comboBoxRate = new System.Windows.Forms.ComboBox();
            this.comboBoxPortName = new System.Windows.Forms.ComboBox();
            this.label8 = new System.Windows.Forms.Label();
            this.label7 = new System.Windows.Forms.Label();
            this.label6 = new System.Windows.Forms.Label();
            this.label5 = new System.Windows.Forms.Label();
            this.label4 = new System.Windows.Forms.Label();
            this.tabPage5 = new System.Windows.Forms.TabPage();
            this.groupBox11 = new System.Windows.Forms.GroupBox();
            this.listBoxSMS = new System.Windows.Forms.ListBox();
            this.panel8 = new System.Windows.Forms.Panel();
            this.buttonEditMsg = new System.Windows.Forms.Button();
            this.buttonRemoveMsg = new System.Windows.Forms.Button();
            this.buttonAddMsg = new System.Windows.Forms.Button();
            this.groupBox10 = new System.Windows.Forms.GroupBox();
            this.listBoxSMSCategory = new System.Windows.Forms.ListBox();
            this.panel7 = new System.Windows.Forms.Panel();
            this.buttonRemoveCategory = new System.Windows.Forms.Button();
            this.buttonAddCategory = new System.Windows.Forms.Button();
            this.timer1 = new System.Windows.Forms.Timer(this.components);
            this.imageList2 = new System.Windows.Forms.ImageList(this.components);
            this.panel1.SuspendLayout();
            this.menuStrip1.SuspendLayout();
            this.panel2.SuspendLayout();
            this.tabControl1.SuspendLayout();
            this.tabPage1.SuspendLayout();
            this.panel3.SuspendLayout();
            this.tabPage3.SuspendLayout();
            this.panel4.SuspendLayout();
            this.groupBox7.SuspendLayout();

        ((System.ComponentModel.ISupportInitialize)(this.pictureBox2)).BeginInit();

        ((System.ComponentModel.ISupportInitialize)(this.pictureBox1)).BeginInit();
            this.panel5.SuspendLayout();
            this.groupBox3.SuspendLayout();
            this.groupBox2.SuspendLayout();
            this.tabPage2.SuspendLayout();
            this.groupBox5.SuspendLayout();
            this.panel6.SuspendLayout();
            this.tabPage4.SuspendLayout();
            this.groupBox6.SuspendLayout();
            this.groupBox4.SuspendLayout();
            this.groupBox9.SuspendLayout();
            this.groupBox8.SuspendLayout();
            this.groupBox1.SuspendLayout();
            this.tabPage5.SuspendLayout();
            this.groupBox11.SuspendLayout();
            this.panel8.SuspendLayout();
            this.groupBox10.SuspendLayout();
            this.panel7.SuspendLayout();
            this.SuspendLayout();
            //
            // panel1
            //
            this.panel1.Controls.Add(this.menuStrip1);
            this.panel1.Dock = System.Windows.Forms.DockStyle.Top;
```

```
            this.panel1.Location = new System.Drawing.Point(0, 0);
            this.panel1.Name = "panel1";
            this.panel1.Size = new System.Drawing.Size(850, 61);
            this.panel1.TabIndex = 0;
            //
            // menuStrip1
            //
            this.menuStrip1.Items.AddRange(new System.Windows.Forms.ToolStripItem[] {
            this.systemToolStripMenuItem,
            this.helpToolStripMenuItem});
            this.menuStrip1.Location = new System.Drawing.Point(0, 0);
            this.menuStrip1.Name = "menuStrip1";
            this.menuStrip1.Size = new System.Drawing.Size(850, 24);
            this.menuStrip1.TabIndex = 0;
            this.menuStrip1.Text = "menuStrip1";
            //
            // systemToolStripMenuItem
            //
            this.systemToolStripMenuItem.DropDownItems.AddRange(new
System.Windows.Forms.ToolStripItem[] {
            this.openDatabaseToolStripMenuItem,
            this.saveDatabaseToolStripMenuItem,
            this.exitToolStripMenuItem});
            this.systemToolStripMenuItem.Name = "systemToolStripMenuItem";
            this.systemToolStripMenuItem.Size = new System.Drawing.Size(57, 20);
            this.systemToolStripMenuItem.Text = "System";
            //
            // openDatabaseToolStripMenuItem
            //
            this.openDatabaseToolStripMenuItem.Name = "openDatabaseToolStripMenuItem";
            this.openDatabaseToolStripMenuItem.Size = new System.Drawing.Size(154,
22);
            this.openDatabaseToolStripMenuItem.Text = "Open Database";
            this.openDatabaseToolStripMenuItem.Click += new
System.EventHandler(this.openDatabaseToolStripMenuItem_Click);
            //
            // saveDatabaseToolStripMenuItem
            //
            this.saveDatabaseToolStripMenuItem.Name = "saveDatabaseToolStripMenuItem";
            this.saveDatabaseToolStripMenuItem.Size = new System.Drawing.Size(154,
22);
            this.saveDatabaseToolStripMenuItem.Text = "Save Database";
            this.saveDatabaseToolStripMenuItem.Click += new
System.EventHandler(this.saveDatabaseToolStripMenuItem_Click);
            //
            // exitToolStripMenuItem
            //
            this.exitToolStripMenuItem.Name = "exitToolStripMenuItem";
            this.exitToolStripMenuItem.Size = new System.Drawing.Size(154, 22);
            this.exitToolStripMenuItem.Text = "Exit";
            this.exitToolStripMenuItem.Click += new
System.EventHandler(this.exitToolStripMenuItem_Click);
            //
            // helpToolStripMenuItem
            //
            this.helpToolStripMenuItem.DropDownItems.AddRange(new
System.Windows.Forms.ToolStripItem[] {
            this.aboutToolStripMenuItem});
            this.helpToolStripMenuItem.Name = "helpToolStripMenuItem";
            this.helpToolStripMenuItem.Size = new System.Drawing.Size(44, 20);
            this.helpToolStripMenuItem.Text = "Help";
            //
            // aboutToolStripMenuItem
            //
            this.aboutToolStripMenuItem.Name = "aboutToolStripMenuItem";
```

```
            this.aboutToolStripMenuItem.Size = new System.Drawing.Size(107, 22);
            this.aboutToolStripMenuItem.Text = "About";
            this.aboutToolStripMenuItem.Click += new
System.EventHandler(this.aboutToolStripMenuItem_Click);
            //
            // panel2
            //
            this.panel2.Controls.Add(this.tabControl1);
            this.panel2.Dock = System.Windows.Forms.DockStyle.Fill;
            this.panel2.Location = new System.Drawing.Point(0, 61);
            this.panel2.Name = "panel2";
            this.panel2.Size = new System.Drawing.Size(850, 532);
            this.panel2.TabIndex = 1;
            //
            // tabControl1
            //
            this.tabControl1.Controls.Add(this.tabPage1);
            this.tabControl1.Controls.Add(this.tabPage3);
            this.tabControl1.Controls.Add(this.tabPage2);
            this.tabControl1.Controls.Add(this.tabPage5);
            this.tabControl1.Controls.Add(this.tabPage4);
            this.tabControl1.Dock = System.Windows.Forms.DockStyle.Fill;
            this.tabControl1.Location = new System.Drawing.Point(0, 0);
            this.tabControl1.Name = "tabControl1";
            this.tabControl1.SelectedIndex = 0;
            this.tabControl1.Size = new System.Drawing.Size(850, 532);
            this.tabControl1.TabIndex = 0;
            //
            // tabPage1
            //
            this.tabPage1.Controls.Add(this.listViewCustomers);
            this.tabPage1.Controls.Add(this.panel3);
            this.tabPage1.Location = new System.Drawing.Point(4, 22);
            this.tabPage1.Name = "tabPage1";
            this.tabPage1.Padding = new System.Windows.Forms.Padding(3);
            this.tabPage1.Size = new System.Drawing.Size(842, 506);
            this.tabPage1.TabIndex = 0;
            this.tabPage1.Text = "TAGs";
            this.tabPage1.UseVisualStyleBackColor = true;
            //
            // listViewCustomers
            //
            this.listViewCustomers.Columns.AddRange(new
System.Windows.Forms.ColumnHeader[] {
            this.columnHeader1,
            this.columnHeader2,
            this.columnHeader3,
            this.columnHeader8,
            this.columnHeader4});
            this.listViewCustomers.Dock = System.Windows.Forms.DockStyle.Fill;
            this.listViewCustomers.FullRowSelect = true;
            this.listViewCustomers.Location = new System.Drawing.Point(3, 3);
            this.listViewCustomers.Name = "listViewCustomers";
            this.listViewCustomers.Size = new System.Drawing.Size(717, 500);
            this.listViewCustomers.TabIndex = 1;
            this.listViewCustomers.UseCompatibleStateImageBehavior = false;
            this.listViewCustomers.View = System.Windows.Forms.View.Details;
            this.listViewCustomers.SelectedIndexChanged += new
System.EventHandler(this.listViewCustomers_SelectedIndexChanged);
            //
            // columnHeader1
            //
            this.columnHeader1.Text = "TAG Number";
            this.columnHeader1.Width = 120;
            //
```

```
            // columnHeader2
            //
            this.columnHeader2.Text = "Customer Name";
            this.columnHeader2.Width = 150;
            //
            // columnHeader3
            //
            this.columnHeader3.Text = "Customer Mobile";
            this.columnHeader3.Width = 120;
            //
            // columnHeader8
            //
            this.columnHeader8.Text = "Customer Email";
            this.columnHeader8.Width = 150;
            //
            // columnHeader4
            //
            this.columnHeader4.Text = "TAG Expiration Date";
            this.columnHeader4.Width = 120;
            //
            // panel3
            //
            this.panel3.Controls.Add(this.buttonSMS);
            this.panel3.Controls.Add(this.buttonDeleteTag);
            this.panel3.Controls.Add(this.buttonEditTag);
            this.panel3.Controls.Add(this.buttonAddTag);
            this.panel3.Controls.Add(this.label1);
            this.panel3.Dock = System.Windows.Forms.DockStyle.Right;
            this.panel3.Location = new System.Drawing.Point(720, 3);
            this.panel3.Name = "panel3";
            this.panel3.Size = new System.Drawing.Size(119, 500);
            this.panel3.TabIndex = 0;
            //
            // buttonSMS
            //
            this.buttonSMS.Location = new System.Drawing.Point(9, 149);
            this.buttonSMS.Name = "buttonSMS";
            this.buttonSMS.Size = new System.Drawing.Size(75, 23);
            this.buttonSMS.TabIndex = 4;
            this.buttonSMS.Text = "SMS";
            this.buttonSMS.UseVisualStyleBackColor = true;
            this.buttonSMS.Click += new System.EventHandler(this.buttonSMS_Click);
            //
            // buttonDeleteTag
            //
            this.buttonDeleteTag.Location = new System.Drawing.Point(9, 93);
            this.buttonDeleteTag.Name = "buttonDeleteTag";
            this.buttonDeleteTag.Size = new System.Drawing.Size(75, 23);
            this.buttonDeleteTag.TabIndex = 3;
            this.buttonDeleteTag.Text = "Delete";
            this.buttonDeleteTag.UseVisualStyleBackColor = true;
            this.buttonDeleteTag.Click += new
System.EventHandler(this.buttonDeleteTag_Click);
            //
            // buttonEditTag
            //
            this.buttonEditTag.Location = new System.Drawing.Point(9, 64);
            this.buttonEditTag.Name = "buttonEditTag";
            this.buttonEditTag.Size = new System.Drawing.Size(75, 23);
            this.buttonEditTag.TabIndex = 2;
            this.buttonEditTag.Text = "Edit";
            this.buttonEditTag.UseVisualStyleBackColor = true;
            this.buttonEditTag.Click += new
System.EventHandler(this.buttonEditTag_Click);
            //
```

```
            // buttonAddTag
            //
            this.buttonAddTag.Location = new System.Drawing.Point(9, 35);
            this.buttonAddTag.Name = "buttonAddTag";
            this.buttonAddTag.Size = new System.Drawing.Size(75, 23);
            this.buttonAddTag.TabIndex = 1;
            this.buttonAddTag.Text = "Add";
            this.buttonAddTag.UseVisualStyleBackColor = true;
            this.buttonAddTag.Click += new
System.EventHandler(this.buttonAddTag_Click);
            //
            // label1
            //
            this.label1.AutoSize = true;
            this.label1.Location = new System.Drawing.Point(6, 9);
            this.label1.Name = "label1";
            this.label1.Size = new System.Drawing.Size(42, 13);
            this.label1.TabIndex = 0;
            this.label1.Text = "Actions";
            //
            // tabPage3
            //
            this.tabPage3.Controls.Add(this.txtCommandsLog);
            this.tabPage3.Controls.Add(this.panel4);
            this.tabPage3.Controls.Add(this.panel5);
            this.tabPage3.Location = new System.Drawing.Point(4, 22);
            this.tabPage3.Name = "tabPage3";
            this.tabPage3.Size = new System.Drawing.Size(842, 506);
            this.tabPage3.TabIndex = 2;
            this.tabPage3.Text = "Control";
            this.tabPage3.UseVisualStyleBackColor = true;
            //
            // txtCommandsLog
            //
            this.txtCommandsLog.BackColor = System.Drawing.SystemColors.GrayText;
            this.txtCommandsLog.Dock = System.Windows.Forms.DockStyle.Fill;
            this.txtCommandsLog.Font = new System.Drawing.Font("Microsoft Sans Serif",
9.75F, System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.txtCommandsLog.ForeColor = System.Drawing.Color.White;
            this.txtCommandsLog.Location = new System.Drawing.Point(0, 0);
            this.txtCommandsLog.Multiline = true;
            this.txtCommandsLog.Name = "txtCommandsLog";
            this.txtCommandsLog.ReadOnly = true;
            this.txtCommandsLog.ScrollBars = System.Windows.Forms.ScrollBars.Vertical;
            this.txtCommandsLog.Size = new System.Drawing.Size(653, 309);
            this.txtCommandsLog.TabIndex = 7;
            this.txtCommandsLog.TextChanged += new
System.EventHandler(this.txtCommandsLog_TextChanged);
            //
            // panel4
            //
            this.panel4.Controls.Add(this.groupBox7);
            this.panel4.Controls.Add(this.pictureBox2);
            this.panel4.Controls.Add(this.pictureBox1);
            this.panel4.Dock = System.Windows.Forms.DockStyle.Bottom;
            this.panel4.Location = new System.Drawing.Point(0, 309);
            this.panel4.Name = "panel4";
            this.panel4.Size = new System.Drawing.Size(653, 197);
            this.panel4.TabIndex = 8;
            //
            // groupBox7
            //
            this.groupBox7.Controls.Add(this.buttonWriteArduino);
            this.groupBox7.Controls.Add(this.textBoxWriteArduino);
            this.groupBox7.Controls.Add(this.label14);
```

```
            this.groupBox7.Controls.Add(this.label13);
            this.groupBox7.Controls.Add(this.buttonArduinoWrite);
            this.groupBox7.Controls.Add(this.textBoxReadArduino);
            this.groupBox7.Location = new System.Drawing.Point(317, 10);
            this.groupBox7.Name = "groupBox7";
            this.groupBox7.Size = new System.Drawing.Size(321, 179);
            this.groupBox7.TabIndex = 2;
            this.groupBox7.TabStop = false;
            this.groupBox7.Text = "Arduino Simulation";
            //
            // buttonWriteArduino
            //
            this.buttonWriteArduino.Location = new System.Drawing.Point(227, 131);
            this.buttonWriteArduino.Name = "buttonWriteArduino";
            this.buttonWriteArduino.Size = new System.Drawing.Size(88, 30);
            this.buttonWriteArduino.TabIndex = 12;
            this.buttonWriteArduino.Text = "Write Line";
            this.buttonWriteArduino.UseVisualStyleBackColor = true;
            this.buttonWriteArduino.Click += new
System.EventHandler(this.buttonWriteArduino_Click);
            //
            // textBoxWriteArduino
            //
            this.textBoxWriteArduino.Location = new System.Drawing.Point(6, 134);
            this.textBoxWriteArduino.Multiline = true;
            this.textBoxWriteArduino.Name = "textBoxWriteArduino";
            this.textBoxWriteArduino.Size = new System.Drawing.Size(215, 27);
            this.textBoxWriteArduino.TabIndex = 11;
            //
            // label14
            //
            this.label14.AutoSize = true;
            this.label14.Location = new System.Drawing.Point(6, 84);
            this.label14.Name = "label14";
            this.label14.Size = new System.Drawing.Size(111, 13);
            this.label14.TabIndex = 10;
            this.label14.Text = "t=10230411,g=1,d=in,";
            //
            // label13
            //
            this.label13.AutoSize = true;
            this.label13.Location = new System.Drawing.Point(6, 61);
            this.label13.Name = "label13";
            this.label13.Size = new System.Drawing.Size(219, 13);
            this.label13.TabIndex = 9;
            this.label13.Text = "t=[tag-number],g=[gate-number],d=[direction],";
            //
            // buttonArduinoWrite
            //
            this.buttonArduinoWrite.Location = new System.Drawing.Point(227, 28);
            this.buttonArduinoWrite.Name = "buttonArduinoWrite";
            this.buttonArduinoWrite.Size = new System.Drawing.Size(88, 30);
            this.buttonArduinoWrite.TabIndex = 8;
            this.buttonArduinoWrite.Text = "Read Line";
            this.buttonArduinoWrite.UseVisualStyleBackColor = true;
            this.buttonArduinoWrite.Click += new
System.EventHandler(this.buttonArduinoWrite_Click);
            //
            // textBoxReadArduino
            //
            this.textBoxReadArduino.Location = new System.Drawing.Point(6, 31);
            this.textBoxReadArduino.Multiline = true;
            this.textBoxReadArduino.Name = "textBoxReadArduino";
            this.textBoxReadArduino.Size = new System.Drawing.Size(215, 27);
            this.textBoxReadArduino.TabIndex = 7;
```

```
            //
            // pictureBox2
            //
            this.pictureBox2.Location = new System.Drawing.Point(214, 77);
            this.pictureBox2.Name = "pictureBox2";
            this.pictureBox2.Size = new System.Drawing.Size(56, 50);
            this.pictureBox2.TabIndex = 1;
            this.pictureBox2.TabStop = false;
            //
            // pictureBox1
            //
            this.pictureBox1.Location = new System.Drawing.Point(8, 10);
            this.pictureBox1.Name = "pictureBox1";
            this.pictureBox1.Size = new System.Drawing.Size(191, 179);
            this.pictureBox1.TabIndex = 0;
            this.pictureBox1.TabStop = false;
            //
            // panel5
            //
            this.panel5.Controls.Add(this.comboBoxGate);
            this.panel5.Controls.Add(this.label10);
            this.panel5.Controls.Add(this.groupBox3);
            this.panel5.Controls.Add(this.groupBox2);
            this.panel5.Controls.Add(this.buttonCloseGate);
            this.panel5.Controls.Add(this.buttonOpenGate);
            this.panel5.Controls.Add(this.buttonStopListening);
            this.panel5.Controls.Add(this.buttonStartListening);
            this.panel5.Controls.Add(this.label3);
            this.panel5.Dock = System.Windows.Forms.DockStyle.Right;
            this.panel5.Location = new System.Drawing.Point(653, 0);
            this.panel5.Name = "panel5";
            this.panel5.Size = new System.Drawing.Size(189, 506);
            this.panel5.TabIndex = 1;
            //
            // comboBoxGate
            //
            this.comboBoxGate.FormattingEnabled = true;
            this.comboBoxGate.Items.AddRange(new object[] {
            "1",
            "2",
            "3",
            "4"});
            this.comboBoxGate.Location = new System.Drawing.Point(15, 146);
            this.comboBoxGate.Name = "comboBoxGate";
            this.comboBoxGate.Size = new System.Drawing.Size(69, 21);
            this.comboBoxGate.TabIndex = 9;
            //
            // label10
            //
            this.label10.AutoSize = true;
            this.label10.Location = new System.Drawing.Point(12, 130);
            this.label10.Name = "label10";
            this.label10.Size = new System.Drawing.Size(47, 13);
            this.label10.TabIndex = 8;
            this.label10.Text = "Gate No";
            //
            // groupBox3
            //
            this.groupBox3.Controls.Add(this.label9);
            this.groupBox3.Controls.Add(this.label2);
            this.groupBox3.Controls.Add(this.txtFreeParking);
            this.groupBox3.Controls.Add(this.txtTotalParkingCount);
            this.groupBox3.Controls.Add(this.buttonSetParking);
            this.groupBox3.Location = new System.Drawing.Point(9, 173);
            this.groupBox3.Name = "groupBox3";
```

```
            this.groupBox3.Size = new System.Drawing.Size(171, 140);
            this.groupBox3.TabIndex = 7;
            this.groupBox3.TabStop = false;
            this.groupBox3.Text = "Parkings Count";
            //
            // label9
            //
            this.label9.AutoSize = true;
            this.label9.Location = new System.Drawing.Point(10, 113);
            this.label9.Name = "label9";
            this.label9.Size = new System.Drawing.Size(28, 13);
            this.label9.TabIndex = 9;
            this.label9.Text = "Free";
            //
            // label2
            //
            this.label2.AutoSize = true;
            this.label2.Location = new System.Drawing.Point(7, 80);
            this.label2.Name = "label2";
            this.label2.Size = new System.Drawing.Size(34, 13);
            this.label2.TabIndex = 8;
            this.label2.Text = "Total ";
            //
            // txtFreeParking
            //
            this.txtFreeParking.Location = new System.Drawing.Point(47, 102);
            this.txtFreeParking.Multiline = true;
            this.txtFreeParking.Name = "txtFreeParking";
            this.txtFreeParking.Size = new System.Drawing.Size(75, 27);
            this.txtFreeParking.TabIndex = 7;
            //
            // txtTotalParkingCount
            //
            this.txtTotalParkingCount.Location = new System.Drawing.Point(47, 69);
            this.txtTotalParkingCount.Multiline = true;
            this.txtTotalParkingCount.Name = "txtTotalParkingCount";
            this.txtTotalParkingCount.Size = new System.Drawing.Size(75, 27);
            this.txtTotalParkingCount.TabIndex = 6;
            //
            // buttonSetParking
            //
            this.buttonSetParking.Location = new System.Drawing.Point(47, 19);
            this.buttonSetParking.Name = "buttonSetParking";
            this.buttonSetParking.Size = new System.Drawing.Size(75, 42);
            this.buttonSetParking.TabIndex = 5;
            this.buttonSetParking.Text = "Set";
            this.buttonSetParking.UseVisualStyleBackColor = true;
            this.buttonSetParking.Click += new
System.EventHandler(this.buttonSetParking_Click);
            //
            // groupBox2
            //
            this.groupBox2.Controls.Add(this.txtLED);
            this.groupBox2.Controls.Add(this.buttonDisplayLED);
            this.groupBox2.Location = new System.Drawing.Point(9, 319);
            this.groupBox2.Name = "groupBox2";
            this.groupBox2.Size = new System.Drawing.Size(171, 161);
            this.groupBox2.TabIndex = 6;
            this.groupBox2.TabStop = false;
            this.groupBox2.Text = "LED Screen";
            //
            // txtLED
            //
            this.txtLED.Location = new System.Drawing.Point(6, 67);
            this.txtLED.Multiline = true;
```

```
            this.txtLED.Name = "txtLED";
            this.txtLED.Size = new System.Drawing.Size(159, 85);
            this.txtLED.TabIndex = 6;
            //
            // buttonDisplayLED
            //
            this.buttonDisplayLED.Location = new System.Drawing.Point(6, 19);
            this.buttonDisplayLED.Name = "buttonDisplayLED";
            this.buttonDisplayLED.Size = new System.Drawing.Size(75, 42);
            this.buttonDisplayLED.TabIndex = 5;
            this.buttonDisplayLED.Text = "Display Text";
            this.buttonDisplayLED.UseVisualStyleBackColor = true;
            this.buttonDisplayLED.Click += new
System.EventHandler(this.buttonDisplayLED_Click);
            //
            // buttonCloseGate
            //
            this.buttonCloseGate.Location = new System.Drawing.Point(105, 84);
            this.buttonCloseGate.Name = "buttonCloseGate";
            this.buttonCloseGate.Size = new System.Drawing.Size(75, 42);
            this.buttonCloseGate.TabIndex = 4;
            this.buttonCloseGate.Text = "Close Gate";
            this.buttonCloseGate.UseVisualStyleBackColor = true;
            this.buttonCloseGate.Click += new
System.EventHandler(this.buttonCloseGate_Click);
            //
            // buttonOpenGate
            //
            this.buttonOpenGate.Location = new System.Drawing.Point(9, 84);
            this.buttonOpenGate.Name = "buttonOpenGate";
            this.buttonOpenGate.Size = new System.Drawing.Size(75, 42);
            this.buttonOpenGate.TabIndex = 3;
            this.buttonOpenGate.Text = "Open Gate";
            this.buttonOpenGate.UseVisualStyleBackColor = true;
            this.buttonOpenGate.Click += new
System.EventHandler(this.buttonOpenGate_Click);
            //
            // buttonStopListening
            //
            this.buttonStopListening.Location = new System.Drawing.Point(105, 37);
            this.buttonStopListening.Name = "buttonStopListening";
            this.buttonStopListening.Size = new System.Drawing.Size(75, 42);
            this.buttonStopListening.TabIndex = 2;
            this.buttonStopListening.Text = "Stop Listening";
            this.buttonStopListening.UseVisualStyleBackColor = true;
            this.buttonStopListening.Click += new
System.EventHandler(this.buttonStopListening_Click);
            //
            // buttonStartListening
            //
            this.buttonStartListening.Location = new System.Drawing.Point(9, 37);
            this.buttonStartListening.Name = "buttonStartListening";
            this.buttonStartListening.Size = new System.Drawing.Size(75, 42);
            this.buttonStartListening.TabIndex = 1;
            this.buttonStartListening.Text = "Start Listening";
            this.buttonStartListening.UseVisualStyleBackColor = true;
            this.buttonStartListening.Click += new
System.EventHandler(this.buttonStartListening_Click);
            //
            // label3
            //
            this.label3.AutoSize = true;
            this.label3.Location = new System.Drawing.Point(6, 9);
            this.label3.Name = "label3";
            this.label3.Size = new System.Drawing.Size(42, 13);
```

```
            this.label3.TabIndex = 0;
            this.label3.Text = "Actions";
            //
            // tabPage2
            //
            this.tabPage2.Controls.Add(this.groupBox5);
            this.tabPage2.Location = new System.Drawing.Point(4, 22);
            this.tabPage2.Name = "tabPage2";
            this.tabPage2.Padding = new System.Windows.Forms.Padding(3);
            this.tabPage2.Size = new System.Drawing.Size(842, 506);
            this.tabPage2.TabIndex = 4;
            this.tabPage2.Text = "Log";
            this.tabPage2.UseVisualStyleBackColor = true;
            //
            // groupBox5
            //
            this.groupBox5.Controls.Add(this.listViewLog);
            this.groupBox5.Controls.Add(this.panel6);
            this.groupBox5.Dock = System.Windows.Forms.DockStyle.Fill;
            this.groupBox5.Location = new System.Drawing.Point(3, 3);
            this.groupBox5.Name = "groupBox5";
            this.groupBox5.Size = new System.Drawing.Size(836, 500);
            this.groupBox5.TabIndex = 0;
            this.groupBox5.TabStop = false;
            this.groupBox5.Text = "In";
            //
            // listViewLog
            //
            this.listViewLog.Columns.AddRange(new System.Windows.Forms.ColumnHeader[]
{
            this.columnHeader5,
            this.columnHeader10,
            this.columnHeader6,
            this.columnHeader7,
            this.columnHeader9});
            this.listViewLog.Dock = System.Windows.Forms.DockStyle.Fill;
            this.listViewLog.LargeImageList = this.imageList1;
            this.listViewLog.Location = new System.Drawing.Point(3, 16);
            this.listViewLog.Name = "listViewLog";
            this.listViewLog.Size = new System.Drawing.Size(694, 481);
            this.listViewLog.TabIndex = 0;
            this.listViewLog.UseCompatibleStateImageBehavior = false;
            this.listViewLog.View = System.Windows.Forms.View.Details;
            //
            // columnHeader5
            //
            this.columnHeader5.Text = "Tag";
            this.columnHeader5.Width = 120;
            //
            // columnHeader10
            //
            this.columnHeader10.Text = "Time";
            this.columnHeader10.Width = 160;
            //
            // columnHeader6
            //
            this.columnHeader6.Text = "Customer Name";
            this.columnHeader6.Width = 150;
            //
            // columnHeader7
            //
            this.columnHeader7.Text = "Gate";
            //
            // columnHeader9
            //
```

```
            this.columnHeader9.Text = "In/Out";
            //
            // imageList1
            //
            this.imageList1.ColorDepth = System.Windows.Forms.ColorDepth.Depth8Bit;
            this.imageList1.ImageSize = new System.Drawing.Size(64, 64);
            this.imageList1.TransparentColor = System.Drawing.Color.Transparent;
            //
            // panel6
            //
            this.panel6.Controls.Add(this.buttonExport);
            this.panel6.Controls.Add(this.label12);
            this.panel6.Dock = System.Windows.Forms.DockStyle.Right;
            this.panel6.Location = new System.Drawing.Point(697, 16);
            this.panel6.Name = "panel6";
            this.panel6.Size = new System.Drawing.Size(136, 481);
            this.panel6.TabIndex = 1;
            //
            // buttonExport
            //
            this.buttonExport.Location = new System.Drawing.Point(9, 36);
            this.buttonExport.Name = "buttonExport";
            this.buttonExport.Size = new System.Drawing.Size(75, 23);
            this.buttonExport.TabIndex = 2;
            this.buttonExport.Text = "Export ";
            this.buttonExport.UseVisualStyleBackColor = true;
            this.buttonExport.Click += new
System.EventHandler(this.buttonExport_Click);
            //
            // label12
            //
            this.label12.AutoSize = true;
            this.label12.Location = new System.Drawing.Point(6, 10);
            this.label12.Name = "label12";
            this.label12.Size = new System.Drawing.Size(42, 13);
            this.label12.TabIndex = 1;
            this.label12.Text = "Actions";
            //
            // tabPage4
            //
            this.tabPage4.Controls.Add(this.groupBox6);
            this.tabPage4.Controls.Add(this.groupBox4);
            this.tabPage4.Controls.Add(this.groupBox1);
            this.tabPage4.Location = new System.Drawing.Point(4, 22);
            this.tabPage4.Name = "tabPage4";
            this.tabPage4.Padding = new System.Windows.Forms.Padding(3);
            this.tabPage4.Size = new System.Drawing.Size(842, 506);
            this.tabPage4.TabIndex = 3;
            this.tabPage4.Text = "Options";
            this.tabPage4.UseVisualStyleBackColor = true;
            //
            // groupBox6
            //
            this.groupBox6.Controls.Add(this.checkBoxTag1);
            this.groupBox6.Location = new System.Drawing.Point(521, 6);
            this.groupBox6.Name = "groupBox6";
            this.groupBox6.Size = new System.Drawing.Size(228, 295);
            this.groupBox6.TabIndex = 6;
            this.groupBox6.TabStop = false;
            this.groupBox6.Text = "Tag Validation";
            //
            // checkBoxTag1
            //
            this.checkBoxTag1.AutoSize = true;
            this.checkBoxTag1.Location = new System.Drawing.Point(15, 24);
```

```
this.checkBoxTag1.Name = "checkBoxTag1";
this.checkBoxTag1.Size = new System.Drawing.Size(199, 17);
this.checkBoxTag1.TabIndex = 0;
this.checkBoxTag1.Text = "Tag is used by only one car at a time";
this.checkBoxTag1.UseVisualStyleBackColor = true;
//
// groupBox4
//
this.groupBox4.Controls.Add(this.groupBox9);
this.groupBox4.Controls.Add(this.groupBox8);
this.groupBox4.Controls.Add(this.label11);
this.groupBox4.Controls.Add(this.comboBoxNotifcations);
this.groupBox4.Controls.Add(this.checkBoxNotifications);
this.groupBox4.Location = new System.Drawing.Point(242, 6);
this.groupBox4.Name = "groupBox4";
this.groupBox4.Size = new System.Drawing.Size(273, 295);
this.groupBox4.TabIndex = 5;
this.groupBox4.TabStop = false;
this.groupBox4.Text = "Notifcation Settings";
//
// groupBox9
//
this.groupBox9.Controls.Add(this.textBoxSmsApi);
this.groupBox9.Controls.Add(this.label19);
this.groupBox9.Controls.Add(this.label17);
this.groupBox9.Controls.Add(this.textBoxSmsPassword);
this.groupBox9.Controls.Add(this.label18);
this.groupBox9.Controls.Add(this.textBoxSmsAccount);
this.groupBox9.Location = new System.Drawing.Point(10, 161);
this.groupBox9.Name = "groupBox9";
this.groupBox9.Size = new System.Drawing.Size(257, 104);
this.groupBox9.TabIndex = 8;
this.groupBox9.TabStop = false;
this.groupBox9.Text = "SMS";
//
// textBoxSmsApi
//
this.textBoxSmsApi.Location = new System.Drawing.Point(94, 69);
this.textBoxSmsApi.Name = "textBoxSmsApi";
this.textBoxSmsApi.Size = new System.Drawing.Size(124, 20);
this.textBoxSmsApi.TabIndex = 8;
//
// label19
//
this.label19.AutoSize = true;
this.label19.Location = new System.Drawing.Point(6, 72);
this.label19.Name = "label19";
this.label19.Size = new System.Drawing.Size(41, 13);
this.label19.TabIndex = 7;
this.label19.Text = "API_ID";
//
// label17
//
this.label17.AutoSize = true;
this.label17.Location = new System.Drawing.Point(6, 20);
this.label17.Name = "label17";
this.label17.Size = new System.Drawing.Size(47, 13);
this.label17.TabIndex = 3;
this.label17.Text = "Account";
//
// textBoxSmsPassword
//
this.textBoxSmsPassword.Location = new System.Drawing.Point(94, 43);
this.textBoxSmsPassword.Name = "textBoxSmsPassword";
this.textBoxSmsPassword.Size = new System.Drawing.Size(124, 20);
```

```
this.textBoxSmsPassword.TabIndex = 6;
this.textBoxSmsPassword.UseSystemPasswordChar = true;
//
// label18
//
this.label18.AutoSize = true;
this.label18.Location = new System.Drawing.Point(6, 46);
this.label18.Name = "label18";
this.label18.Size = new System.Drawing.Size(53, 13);
this.label18.TabIndex = 4;
this.label18.Text = "Password";
//
// textBoxSmsAccount
//
this.textBoxSmsAccount.Location = new System.Drawing.Point(94, 13);
this.textBoxSmsAccount.Name = "textBoxSmsAccount";
this.textBoxSmsAccount.Size = new System.Drawing.Size(124, 20);
this.textBoxSmsAccount.TabIndex = 5;
//
// groupBox8
//
this.groupBox8.Controls.Add(this.label15);
this.groupBox8.Controls.Add(this.textBoxGmailPassword);
this.groupBox8.Controls.Add(this.label16);
this.groupBox8.Controls.Add(this.textBoxGmailAccount);
this.groupBox8.Location = new System.Drawing.Point(10, 81);
this.groupBox8.Name = "groupBox8";
this.groupBox8.Size = new System.Drawing.Size(257, 74);
this.groupBox8.TabIndex = 7;
this.groupBox8.TabStop = false;
this.groupBox8.Text = "Email";
//
// label15
//
this.label15.AutoSize = true;
this.label15.Location = new System.Drawing.Point(6, 20);
this.label15.Name = "label15";
this.label15.Size = new System.Drawing.Size(76, 13);
this.label15.TabIndex = 3;
this.label15.Text = "Gmail Account";
//
// textBoxGmailPassword
//
this.textBoxGmailPassword.Location = new System.Drawing.Point(94, 43);
this.textBoxGmailPassword.Name = "textBoxGmailPassword";
this.textBoxGmailPassword.Size = new System.Drawing.Size(124, 20);
this.textBoxGmailPassword.TabIndex = 6;
this.textBoxGmailPassword.UseSystemPasswordChar = true;
//
// label16
//
this.label16.AutoSize = true;
this.label16.Location = new System.Drawing.Point(6, 46);
this.label16.Name = "label16";
this.label16.Size = new System.Drawing.Size(53, 13);
this.label16.TabIndex = 4;
this.label16.Text = "Password";
//
// textBoxGmailAccount
//
this.textBoxGmailAccount.Location = new System.Drawing.Point(94, 13);
this.textBoxGmailAccount.Name = "textBoxGmailAccount";
this.textBoxGmailAccount.Size = new System.Drawing.Size(124, 20);
this.textBoxGmailAccount.TabIndex = 5;
//
```

```
            // label11
            //
            this.label11.AutoSize = true;
            this.label11.Location = new System.Drawing.Point(7, 57);
            this.label11.Name = "label11";
            this.label11.Size = new System.Drawing.Size(83, 13);
            this.label11.TabIndex = 2;
            this.label11.Text = "Number of Days";
            //
            // comboBoxNotifcations
            //
            this.comboBoxNotifcations.DropDownStyle =
System.Windows.Forms.ComboBoxStyle.DropDownList;
            this.comboBoxNotifcations.FormattingEnabled = true;
            this.comboBoxNotifcations.Items.AddRange(new object[] {
            "1",
            "2",
            "3",
            "4",
            "5",
            "6",
            "7"});
            this.comboBoxNotifcations.Location = new System.Drawing.Point(95, 54);
            this.comboBoxNotifcations.Name = "comboBoxNotifcations";
            this.comboBoxNotifcations.Size = new System.Drawing.Size(124, 21);
            this.comboBoxNotifcations.TabIndex = 1;
            //
            // checkBoxNotifications
            //
            this.checkBoxNotifications.AutoSize = true;
            this.checkBoxNotifications.Location = new System.Drawing.Point(9, 24);
            this.checkBoxNotifications.Name = "checkBoxNotifications";
            this.checkBoxNotifications.Size = new System.Drawing.Size(110, 17);
            this.checkBoxNotifications.TabIndex = 0;
            this.checkBoxNotifications.Text = "Allow Notifcations";
            this.checkBoxNotifications.UseVisualStyleBackColor = true;
            this.checkBoxNotifications.CheckedChanged += new
System.EventHandler(this.checkBoxNotifications_CheckedChanged);
            //
            // groupBox1
            //
            this.groupBox1.Controls.Add(this.comboBoxStopBits);
            this.groupBox1.Controls.Add(this.comboBoxParity);
            this.groupBox1.Controls.Add(this.comboBoxDataBits);
            this.groupBox1.Controls.Add(this.comboBoxRate);
            this.groupBox1.Controls.Add(this.comboBoxPortName);
            this.groupBox1.Controls.Add(this.label8);
            this.groupBox1.Controls.Add(this.label7);
            this.groupBox1.Controls.Add(this.label6);
            this.groupBox1.Controls.Add(this.label5);
            this.groupBox1.Controls.Add(this.label4);
            this.groupBox1.Location = new System.Drawing.Point(3, 3);
            this.groupBox1.Name = "groupBox1";
            this.groupBox1.Size = new System.Drawing.Size(233, 298);
            this.groupBox1.TabIndex = 4;
            this.groupBox1.TabStop = false;
            this.groupBox1.Text = "Serial Port Settings";
            //
            // comboBoxStopBits
            //
            this.comboBoxStopBits.FormattingEnabled = true;
            this.comboBoxStopBits.Location = new System.Drawing.Point(86, 127);
            this.comboBoxStopBits.Name = "comboBoxStopBits";
            this.comboBoxStopBits.Size = new System.Drawing.Size(121, 21);
            this.comboBoxStopBits.TabIndex = 9;
```

```
this.comboBoxStopBits.Visible = false;
//
// comboBoxParity
//
this.comboBoxParity.FormattingEnabled = true;
this.comboBoxParity.Location = new System.Drawing.Point(86, 101);
this.comboBoxParity.Name = "comboBoxParity";
this.comboBoxParity.Size = new System.Drawing.Size(121, 21);
this.comboBoxParity.TabIndex = 8;
this.comboBoxParity.Visible = false;
//
// comboBoxDataBits
//
this.comboBoxDataBits.FormattingEnabled = true;
this.comboBoxDataBits.Location = new System.Drawing.Point(86, 75);
this.comboBoxDataBits.Name = "comboBoxDataBits";
this.comboBoxDataBits.Size = new System.Drawing.Size(121, 21);
this.comboBoxDataBits.TabIndex = 7;
this.comboBoxDataBits.Visible = false;
//
// comboBoxRate
//
this.comboBoxRate.FormattingEnabled = true;
this.comboBoxRate.Items.AddRange(new object[] {
"4800",
"7200",
"9600",
"14400",
"19200 "});
this.comboBoxRate.Location = new System.Drawing.Point(86, 49);
this.comboBoxRate.Name = "comboBoxRate";
this.comboBoxRate.Size = new System.Drawing.Size(121, 21);
this.comboBoxRate.TabIndex = 6;
//
// comboBoxPortName
//
this.comboBoxPortName.FormattingEnabled = true;
this.comboBoxPortName.Items.AddRange(new object[] {
"COM1",
"COM2",
"COM3",
"COM4",
"COM5"});
this.comboBoxPortName.Location = new System.Drawing.Point(86, 23);
this.comboBoxPortName.Name = "comboBoxPortName";
this.comboBoxPortName.Size = new System.Drawing.Size(121, 21);
this.comboBoxPortName.TabIndex = 5;
//
// label8
//
this.label8.AutoSize = true;
this.label8.Location = new System.Drawing.Point(17, 130);
this.label8.Name = "label8";
this.label8.Size = new System.Drawing.Size(49, 13);
this.label8.TabIndex = 4;
this.label8.Text = "Stop Bits";
this.label8.Visible = false;
//
// label7
//
this.label7.AutoSize = true;
this.label7.Location = new System.Drawing.Point(17, 104);
this.label7.Name = "label7";
this.label7.Size = new System.Drawing.Size(33, 13);
this.label7.TabIndex = 3;
```

```
this.label7.Text = "Parity";
this.label7.Visible = false;
//
// label6
//
this.label6.AutoSize = true;
this.label6.Location = new System.Drawing.Point(17, 78);
this.label6.Name = "label6";
this.label6.Size = new System.Drawing.Size(50, 13);
this.label6.TabIndex = 2;
this.label6.Text = "Data Bits";
this.label6.Visible = false;
//
// label5
//
this.label5.AutoSize = true;
this.label5.Location = new System.Drawing.Point(17, 52);
this.label5.Name = "label5";
this.label5.Size = new System.Drawing.Size(58, 13);
this.label5.TabIndex = 1;
this.label5.Text = "Baud Rate";
//
// label4
//
this.label4.AutoSize = true;
this.label4.Location = new System.Drawing.Point(17, 26);
this.label4.Name = "label4";
this.label4.Size = new System.Drawing.Size(57, 13);
this.label4.TabIndex = 0;
this.label4.Text = "Port Name";
//
// tabPage5
//
this.tabPage5.Controls.Add(this.groupBox11);
this.tabPage5.Controls.Add(this.groupBox10);
this.tabPage5.Location = new System.Drawing.Point(4, 22);
this.tabPage5.Name = "tabPage5";
this.tabPage5.Padding = new System.Windows.Forms.Padding(3);
this.tabPage5.Size = new System.Drawing.Size(842, 506);
this.tabPage5.TabIndex = 5;
this.tabPage5.Text = "SMS Alerts";
this.tabPage5.UseVisualStyleBackColor = true;
//
// groupBox11
//
this.groupBox11.Controls.Add(this.listBoxSMS);
this.groupBox11.Controls.Add(this.panel8);
this.groupBox11.Dock = System.Windows.Forms.DockStyle.Fill;
this.groupBox11.Location = new System.Drawing.Point(222, 3);
this.groupBox11.Name = "groupBox11";
this.groupBox11.Size = new System.Drawing.Size(617, 500);
this.groupBox11.TabIndex = 1;
this.groupBox11.TabStop = false;
this.groupBox11.Text = "Message";
//
// listBoxSMS
//
this.listBoxSMS.Dock = System.Windows.Forms.DockStyle.Fill;
this.listBoxSMS.FormattingEnabled = true;
this.listBoxSMS.Location = new System.Drawing.Point(3, 16);
this.listBoxSMS.Name = "listBoxSMS";
this.listBoxSMS.Size = new System.Drawing.Size(523, 481);
this.listBoxSMS.TabIndex = 2;
//
// panel8
```

```
            //
            this.panel8.Controls.Add(this.buttonEditMsg);
            this.panel8.Controls.Add(this.buttonRemoveMsg);
            this.panel8.Controls.Add(this.buttonAddMsg);
            this.panel8.Dock = System.Windows.Forms.DockStyle.Right;
            this.panel8.Location = new System.Drawing.Point(526, 16);
            this.panel8.Name = "panel8";
            this.panel8.Size = new System.Drawing.Size(88, 481);
            this.panel8.TabIndex = 3;
            //
            // buttonEditMsg
            //
            this.buttonEditMsg.Location = new System.Drawing.Point(3, 41);
            this.buttonEditMsg.Name = "buttonEditMsg";
            this.buttonEditMsg.Size = new System.Drawing.Size(76, 32);
            this.buttonEditMsg.TabIndex = 2;
            this.buttonEditMsg.Text = "Edit";
            this.buttonEditMsg.UseVisualStyleBackColor = true;
            this.buttonEditMsg.Click += new
System.EventHandler(this.buttonEditMsg_Click);
            //
            // buttonRemoveMsg
            //
            this.buttonRemoveMsg.Location = new System.Drawing.Point(3, 80);
            this.buttonRemoveMsg.Name = "buttonRemoveMsg";
            this.buttonRemoveMsg.Size = new System.Drawing.Size(77, 35);
            this.buttonRemoveMsg.TabIndex = 1;
            this.buttonRemoveMsg.Text = "Remove";
            this.buttonRemoveMsg.UseVisualStyleBackColor = true;
            this.buttonRemoveMsg.Click += new
System.EventHandler(this.buttonRemoveMsg_Click);
            //
            // buttonAddMsg
            //
            this.buttonAddMsg.Location = new System.Drawing.Point(3, 3);
            this.buttonAddMsg.Name = "buttonAddMsg";
            this.buttonAddMsg.Size = new System.Drawing.Size(76, 32);
            this.buttonAddMsg.TabIndex = 0;
            this.buttonAddMsg.Text = "Add";
            this.buttonAddMsg.UseVisualStyleBackColor = true;
            this.buttonAddMsg.Click += new
System.EventHandler(this.buttonAddMsg_Click);
            //
            // groupBox10
            //
            this.groupBox10.Controls.Add(this.listBoxSMSCategory);
            this.groupBox10.Controls.Add(this.panel7);
            this.groupBox10.Dock = System.Windows.Forms.DockStyle.Left;
            this.groupBox10.Location = new System.Drawing.Point(3, 3);
            this.groupBox10.Name = "groupBox10";
            this.groupBox10.Size = new System.Drawing.Size(219, 500);
            this.groupBox10.TabIndex = 0;
            this.groupBox10.TabStop = false;
            this.groupBox10.Text = "Category";
            //
            // listBoxSMSCategory
            //
            this.listBoxSMSCategory.Dock = System.Windows.Forms.DockStyle.Fill;
            this.listBoxSMSCategory.FormattingEnabled = true;
            this.listBoxSMSCategory.Location = new System.Drawing.Point(3, 16);
            this.listBoxSMSCategory.Name = "listBoxSMSCategory";
            this.listBoxSMSCategory.Size = new System.Drawing.Size(213, 437);
            this.listBoxSMSCategory.TabIndex = 0;
            this.listBoxSMSCategory.SelectedIndexChanged += new
System.EventHandler(this.listBoxSMSCategory_SelectedIndexChanged);
```

```
            //
            // panel7
            //
            this.panel7.Controls.Add(this.buttonRemoveCategory);
            this.panel7.Controls.Add(this.buttonAddCategory);
            this.panel7.Dock = System.Windows.Forms.DockStyle.Bottom;
            this.panel7.Location = new System.Drawing.Point(3, 453);
            this.panel7.Name = "panel7";
            this.panel7.Size = new System.Drawing.Size(213, 44);
            this.panel7.TabIndex = 1;
            //
            // buttonRemoveCategory
            //
            this.buttonRemoveCategory.Location = new System.Drawing.Point(14, 2);
            this.buttonRemoveCategory.Name = "buttonRemoveCategory";
            this.buttonRemoveCategory.Size = new System.Drawing.Size(77, 35);
            this.buttonRemoveCategory.TabIndex = 1;
            this.buttonRemoveCategory.Text = "Remove";
            this.buttonRemoveCategory.UseVisualStyleBackColor = true;
            this.buttonRemoveCategory.Click += new
System.EventHandler(this.buttonRemoveCategory_Click);
            //
            // buttonAddCategory
            //
            this.buttonAddCategory.Location = new System.Drawing.Point(114, 2);
            this.buttonAddCategory.Name = "buttonAddCategory";
            this.buttonAddCategory.Size = new System.Drawing.Size(76, 32);
            this.buttonAddCategory.TabIndex = 0;
            this.buttonAddCategory.Text = "Add";
            this.buttonAddCategory.UseVisualStyleBackColor = true;
            this.buttonAddCategory.Click += new
System.EventHandler(this.buttonAddCategory_Click);
            //
            // timer1
            //
            this.timer1.Interval = 5000;
            this.timer1.Tick += new System.EventHandler(this.timer1_Tick);
            //
            // imageList2
            //
            this.imageList2.ImageStream =
((System.Windows.Forms.ImageListStreamer)(resources.GetObject("imageList2.ImageStream"
)));
            this.imageList2.TransparentColor = System.Drawing.Color.Transparent;
            this.imageList2.Images.SetKeyName(0, "check-64.png");
            this.imageList2.Images.SetKeyName(1, "delete_2.png");
            //
            // FormMain
            //
            this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
            this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
            this.ClientSize = new System.Drawing.Size(850, 593);
            this.Controls.Add(this.panel2);
            this.Controls.Add(this.panel1);
            this.MainMenuStrip = this.menuStrip1;
            this.Name = "FormMain";
            this.StartPosition = System.Windows.Forms.FormStartPosition.CenterScreen;
            this.Text = "Parking Monitoring and Control System";
            this.Load += new System.EventHandler(this.Form1_Load);
            this.panel1.ResumeLayout(false);
            this.panel1.PerformLayout();
            this.menuStrip1.ResumeLayout(false);
            this.menuStrip1.PerformLayout();
            this.panel2.ResumeLayout(false);
            this.tabControl1.ResumeLayout(false);
```

```
            this.tabPage1.ResumeLayout(false);
            this.panel3.ResumeLayout(false);
            this.panel3.PerformLayout();
            this.tabPage3.ResumeLayout(false);
            this.tabPage3.PerformLayout();
            this.panel4.ResumeLayout(false);
            this.groupBox7.ResumeLayout(false);
            this.groupBox7.PerformLayout();
            ((System.ComponentModel.ISupportInitialize)(this.pictureBox2)).EndInit();
            ((System.ComponentModel.ISupportInitialize)(this.pictureBox1)).EndInit();
            this.panel5.ResumeLayout(false);
            this.panel5.PerformLayout();
            this.groupBox3.ResumeLayout(false);
            this.groupBox3.PerformLayout();
            this.groupBox2.ResumeLayout(false);
            this.groupBox2.PerformLayout();
            this.tabPage2.ResumeLayout(false);
            this.groupBox5.ResumeLayout(false);
            this.panel6.ResumeLayout(false);
            this.panel6.PerformLayout();
            this.tabPage4.ResumeLayout(false);
            this.groupBox6.ResumeLayout(false);
            this.groupBox6.PerformLayout();
            this.groupBox4.ResumeLayout(false);
            this.groupBox4.PerformLayout();
            this.groupBox9.ResumeLayout(false);
            this.groupBox9.PerformLayout();
            this.groupBox8.ResumeLayout(false);
            this.groupBox8.PerformLayout();
            this.groupBox1.ResumeLayout(false);
            this.groupBox1.PerformLayout();
            this.tabPage5.ResumeLayout(false);
            this.groupBox11.ResumeLayout(false);
            this.panel8.ResumeLayout(false);
            this.groupBox10.ResumeLayout(false);
            this.panel7.ResumeLayout(false);
            this.ResumeLayout(false);

        }

        #endregion

        private System.Windows.Forms.Panel panel1;
        private System.Windows.Forms.Panel panel2;
        private System.Windows.Forms.TabControl tabControl1;
        private System.Windows.Forms.TabPage tabPage1;
        private System.Windows.Forms.ListView listViewCustomers;
        private System.Windows.Forms.Panel panel3;
        private System.Windows.Forms.Label label1;
        private System.Windows.Forms.TabPage tabPage3;
        private System.Windows.Forms.Panel panel5;
        private System.Windows.Forms.Label label3;
        private System.Windows.Forms.ColumnHeader columnHeader1;
        private System.Windows.Forms.ColumnHeader columnHeader2;
        private System.Windows.Forms.Button buttonDeleteTag;
        private System.Windows.Forms.Button buttonEditTag;
        private System.Windows.Forms.Button buttonAddTag;
        private System.Windows.Forms.MenuStrip menuStrip1;
        private System.Windows.Forms.ToolStripMenuItem systemToolStripMenuItem;
        private System.Windows.Forms.ToolStripMenuItem openDatabaseToolStripMenuItem;
        private System.Windows.Forms.ToolStripMenuItem saveDatabaseToolStripMenuItem;
        private System.Windows.Forms.ToolStripMenuItem exitToolStripMenuItem;
        private System.Windows.Forms.ToolStripMenuItem helpToolStripMenuItem;
        private System.Windows.Forms.ToolStripMenuItem aboutToolStripMenuItem;
        private System.Windows.Forms.TabPage tabPage4;
```

```
        private System.Windows.Forms.GroupBox groupBox1;
        private System.Windows.Forms.ComboBox comboBoxStopBits;
        private System.Windows.Forms.ComboBox comboBoxParity;
        private System.Windows.Forms.ComboBox comboBoxDataBits;
        private System.Windows.Forms.ComboBox comboBoxRate;
        private System.Windows.Forms.ComboBox comboBoxPortName;
        private System.Windows.Forms.Label label8;
        private System.Windows.Forms.Label label7;
        private System.Windows.Forms.Label label6;
        private System.Windows.Forms.Label label5;
        private System.Windows.Forms.Label label4;
        private System.Windows.Forms.Button buttonStopListening;
        private System.Windows.Forms.Button buttonStartListening;
        private System.Windows.Forms.GroupBox groupBox2;
        private System.Windows.Forms.TextBox txtLED;
        private System.Windows.Forms.Button buttonDisplayLED;
        private System.Windows.Forms.Button buttonCloseGate;
        private System.Windows.Forms.Button buttonOpenGate;
        private System.Windows.Forms.TextBox txtCommandsLog;
        private System.Windows.Forms.GroupBox groupBox3;
        private System.Windows.Forms.Label label9;
        private System.Windows.Forms.Label label2;
        private System.Windows.Forms.TextBox txtFreeParking;
        private System.Windows.Forms.TextBox txtTotalParkingCount;
        private System.Windows.Forms.Button buttonSetParking;
        private System.Windows.Forms.ColumnHeader columnHeader3;
        private System.Windows.Forms.ColumnHeader columnHeader4;
        private System.Windows.Forms.ComboBox comboBoxGate;
        private System.Windows.Forms.Label label10;
        private System.Windows.Forms.GroupBox groupBox4;
        private System.Windows.Forms.Label label11;
        private System.Windows.Forms.ComboBox comboBoxNotifcations;
        private System.Windows.Forms.CheckBox checkBoxNotifications;
        private System.Windows.Forms.Timer timer1;
        private System.Windows.Forms.TabPage tabPage2;
        private System.Windows.Forms.GroupBox groupBox5;
        private System.Windows.Forms.ListView listViewLog;
        private System.Windows.Forms.ColumnHeader columnHeader5;
        private System.Windows.Forms.ColumnHeader columnHeader6;
        private System.Windows.Forms.ColumnHeader columnHeader7;
        private System.Windows.Forms.ColumnHeader columnHeader9;
        private System.Windows.Forms.ImageList imageList1;
        private System.Windows.Forms.ColumnHeader columnHeader10;
        private System.Windows.Forms.Panel panel4;
        private System.Windows.Forms.PictureBox pictureBox1;
        private System.Windows.Forms.PictureBox pictureBox2;
        private System.Windows.Forms.ImageList imageList2;
        private System.Windows.Forms.Panel panel6;
        private System.Windows.Forms.Label label12;
        private System.Windows.Forms.GroupBox groupBox6;
        private System.Windows.Forms.CheckBox checkBoxTag1;
        private System.Windows.Forms.GroupBox groupBox7;
        private System.Windows.Forms.Button buttonArduinoWrite;
        private System.Windows.Forms.TextBox textBoxReadArduino;
        private System.Windows.Forms.Label label14;
        private System.Windows.Forms.Label label13;
        private System.Windows.Forms.Button buttonWriteArduino;
        private System.Windows.Forms.TextBox textBoxWriteArduino;
        private System.Windows.Forms.Button buttonExport;
        private System.Windows.Forms.TextBox textBoxGmailPassword;
        private System.Windows.Forms.TextBox textBoxGmailAccount;
        private System.Windows.Forms.Label label16;
        private System.Windows.Forms.Label label15;
        private System.Windows.Forms.ColumnHeader columnHeader8;
        private System.Windows.Forms.Button buttonSMS;
```

```
        private System.Windows.Forms.GroupBox groupBox9;
        private System.Windows.Forms.TextBox textBoxSmsApi;
        private System.Windows.Forms.Label label19;
        private System.Windows.Forms.Label label17;
        private System.Windows.Forms.TextBox textBoxSmsPassword;
        private System.Windows.Forms.Label label18;
        private System.Windows.Forms.TextBox textBoxSmsAccount;
        private System.Windows.Forms.GroupBox groupBox8;
        private System.Windows.Forms.TabPage tabPage5;
        private System.Windows.Forms.GroupBox groupBox11;
        private System.Windows.Forms.ListBox listBoxSMS;
        private System.Windows.Forms.Panel panel8;
        private System.Windows.Forms.Button buttonRemoveMsg;
        private System.Windows.Forms.Button buttonAddMsg;
        private System.Windows.Forms.GroupBox groupBox10;
        private System.Windows.Forms.Panel panel7;
        private System.Windows.Forms.Button buttonRemoveCategory;
        private System.Windows.Forms.Button buttonAddCategory;
        private System.Windows.Forms.ListBox listBoxSMSCategory;
        private System.Windows.Forms.Button buttonEditMsg;
    }
}


namespace ParkingManager
{
    partial class FormAbout
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should be disposed;
otherwise, false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code

        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
            this.label1 = new System.Windows.Forms.Label();
            this.textBox1 = new System.Windows.Forms.TextBox();
            this.buttonOk = new System.Windows.Forms.Button();
            this.SuspendLayout();
            //
            // label1
            //
            this.label1.AutoSize = true;
            this.label1.Location = new System.Drawing.Point(12, 18);
            this.label1.Name = "label1";
```

```
            this.label1.Size = new System.Drawing.Size(189, 13);
            this.label1.TabIndex = 2;
            this.label1.Text = "Parking Monitoring and Control System";
            //
            // textBox1
            //
            this.textBox1.Location = new System.Drawing.Point(15, 44);
            this.textBox1.Multiline = true;
            this.textBox1.Name = "textBox1";
            this.textBox1.ReadOnly = true;
            this.textBox1.Size = new System.Drawing.Size(290, 139);
            this.textBox1.TabIndex = 1;
            //
            // buttonOk
            //
            this.buttonOk.Location = new System.Drawing.Point(126, 191);
            this.buttonOk.Name = "buttonOk";
            this.buttonOk.Size = new System.Drawing.Size(75, 23);
            this.buttonOk.TabIndex = 0;
            this.buttonOk.Text = "Ok";
            this.buttonOk.UseVisualStyleBackColor = true;
            this.buttonOk.Click += new System.EventHandler(this.buttonOk_Click);
            //
            // FormAbout
            //
            this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
            this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
            this.ClientSize = new System.Drawing.Size(317, 226);
            this.Controls.Add(this.buttonOk);
            this.Controls.Add(this.textBox1);
            this.Controls.Add(this.label1);
            this.Name = "FormAbout";
            this.Text = "About";
            this.Load += new System.EventHandler(this.FormAbout_Load);
            this.ResumeLayout(false);
            this.PerformLayout();

        }

        #endregion

        private System.Windows.Forms.Label label1;
        private System.Windows.Forms.TextBox textBox1;
        private System.Windows.Forms.Button buttonOk;
    }
}

namespace ParkingManager
{
    partial class FormSMS
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should be disposed;
otherwise, false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
```

```
            components.Dispose();
        }
        base.Dispose(disposing);
    }

    #region Windows Form Designer generated code

    /// <summary>
    /// Required method for Designer support - do not modify
    /// the contents of this method with the code editor.
    /// </summary>
    private void InitializeComponent()
    {
        this.label1 = new System.Windows.Forms.Label();
        this.textBoxTo = new System.Windows.Forms.TextBox();
        this.textBoxSubject = new System.Windows.Forms.TextBox();
        this.label2 = new System.Windows.Forms.Label();
        this.textBoxMessage = new System.Windows.Forms.TextBox();
        this.label3 = new System.Windows.Forms.Label();
        this.buttonSend = new System.Windows.Forms.Button();
        this.buttonCancel = new System.Windows.Forms.Button();
        this.SuspendLayout();
        //
        // label1
        //
        this.label1.AutoSize = true;
        this.label1.Location = new System.Drawing.Point(12, 26);
        this.label1.Name = "label1";
        this.label1.Size = new System.Drawing.Size(20, 13);
        this.label1.TabIndex = 0;
        this.label1.Text = "To";
        //
        // textBoxTo
        //
        this.textBoxTo.Location = new System.Drawing.Point(66, 23);
        this.textBoxTo.Name = "textBoxTo";
        this.textBoxTo.Size = new System.Drawing.Size(155, 20);
        this.textBoxTo.TabIndex = 1;
        //
        // textBoxSubject
        //
        this.textBoxSubject.Location = new System.Drawing.Point(66, 49);
        this.textBoxSubject.Name = "textBoxSubject";
        this.textBoxSubject.Size = new System.Drawing.Size(155, 20);
        this.textBoxSubject.TabIndex = 3;
        //
        // label2
        //
        this.label2.AutoSize = true;
        this.label2.Location = new System.Drawing.Point(12, 52);
        this.label2.Name = "label2";
        this.label2.Size = new System.Drawing.Size(43, 13);
        this.label2.TabIndex = 2;
        this.label2.Text = "Subject";
        //
        // textBoxMessage
        //
        this.textBoxMessage.Location = new System.Drawing.Point(66, 75);
        this.textBoxMessage.Multiline = true;
        this.textBoxMessage.Name = "textBoxMessage";
        this.textBoxMessage.Size = new System.Drawing.Size(155, 80);
        this.textBoxMessage.TabIndex = 5;
        //
        // label3
        //
```

```
            this.label3.AutoSize = true;
            this.label3.Location = new System.Drawing.Point(12, 78);
            this.label3.Name = "label3";
            this.label3.Size = new System.Drawing.Size(50, 13);
            this.label3.TabIndex = 4;
            this.label3.Text = "Message";
            //
            // buttonSend
            //
            this.buttonSend.Location = new System.Drawing.Point(129, 178);
            this.buttonSend.Name = "buttonSend";
            this.buttonSend.Size = new System.Drawing.Size(75, 23);
            this.buttonSend.TabIndex = 6;
            this.buttonSend.Text = "Send";
            this.buttonSend.UseVisualStyleBackColor = true;
            this.buttonSend.Click += new System.EventHandler(this.buttonSend_Click);
            //
            // buttonCancel
            //
            this.buttonCancel.DialogResult = System.Windows.Forms.DialogResult.Cancel;
            this.buttonCancel.Location = new System.Drawing.Point(34, 178);
            this.buttonCancel.Name = "buttonCancel";
            this.buttonCancel.Size = new System.Drawing.Size(75, 23);
            this.buttonCancel.TabIndex = 7;
            this.buttonCancel.Text = "Cancel";
            this.buttonCancel.UseVisualStyleBackColor = true;
            this.buttonCancel.Click += new
System.EventHandler(this.buttonCancel_Click);
            //
            // FormSMS
            //
            this.AcceptButton = this.buttonSend;
            this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
            this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
            this.CancelButton = this.buttonCancel;
            this.ClientSize = new System.Drawing.Size(238, 213);
            this.ControlBox = false;
            this.Controls.Add(this.buttonCancel);
            this.Controls.Add(this.buttonSend);
            this.Controls.Add(this.textBoxMessage);
            this.Controls.Add(this.label3);
            this.Controls.Add(this.textBoxSubject);
            this.Controls.Add(this.label2);
            this.Controls.Add(this.textBoxTo);
            this.Controls.Add(this.label1);
            this.Name = "FormSMS";
            this.StartPosition = System.Windows.Forms.FormStartPosition.CenterParent;
            this.Text = "Send Message";
            this.Load += new System.EventHandler(this.FormSMS_Load);
            this.ResumeLayout(false);
            this.PerformLayout();

        }

        #endregion

        private System.Windows.Forms.Label label1;
        private System.Windows.Forms.TextBox textBoxTo;
        private System.Windows.Forms.TextBox textBoxSubject;
        private System.Windows.Forms.Label label2;
        private System.Windows.Forms.TextBox textBoxMessage;
        private System.Windows.Forms.Label label3;
        private System.Windows.Forms.Button buttonSend;
        private System.Windows.Forms.Button buttonCancel;
    }
```

```
}
namespace ParkingManager
{
    partial class FormSmsCategory
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should be disposed;
otherwise, false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code

        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
            this.buttonOk = new System.Windows.Forms.Button();
            this.buttonCancel = new System.Windows.Forms.Button();
            this.textBoxCategory = new System.Windows.Forms.TextBox();
            this.SuspendLayout();
            //
            // buttonOk
            //
            this.buttonOk.Location = new System.Drawing.Point(117, 65);
            this.buttonOk.Name = "buttonOk";
            this.buttonOk.Size = new System.Drawing.Size(75, 23);
            this.buttonOk.TabIndex = 5;
            this.buttonOk.Text = "&Ok";
            this.buttonOk.UseVisualStyleBackColor = true;
            this.buttonOk.Click += new System.EventHandler(this.buttonOk_Click);
            //
            // buttonCancel
            //
            this.buttonCancel.DialogResult = System.Windows.Forms.DialogResult.Cancel;
            this.buttonCancel.Location = new System.Drawing.Point(26, 65);
            this.buttonCancel.Name = "buttonCancel";
            this.buttonCancel.Size = new System.Drawing.Size(75, 23);
            this.buttonCancel.TabIndex = 4;
            this.buttonCancel.Text = "&Cancel";
            this.buttonCancel.UseVisualStyleBackColor = true;
            //
            // textBoxCategory
            //
            this.textBoxCategory.Location = new System.Drawing.Point(23, 28);
            this.textBoxCategory.Name = "textBoxCategory";
            this.textBoxCategory.Size = new System.Drawing.Size(172, 20);
            this.textBoxCategory.TabIndex = 6;
            //
            // FormSmsCategory
```

```
            //
            this.AcceptButton = this.buttonOk;
            this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
            this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
            this.CancelButton = this.buttonCancel;
            this.ClientSize = new System.Drawing.Size(225, 100);
            this.Controls.Add(this.textBoxCategory);
            this.Controls.Add(this.buttonOk);
            this.Controls.Add(this.buttonCancel);
            this.Name = "FormSmsCategory";
            this.StartPosition = System.Windows.Forms.FormStartPosition.CenterParent;
            this.Text = "Category";
            this.Load += new System.EventHandler(this.FormSmsCategory_Load);
            this.ResumeLayout(false);
            this.PerformLayout();

        }

        #endregion

        private System.Windows.Forms.Button buttonOk;
        private System.Windows.Forms.Button buttonCancel;
        private System.Windows.Forms.TextBox textBoxCategory;
    }
}
namespace ParkingManager
{
    partial class FormSMSNotification
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should be disposed;
otherwise, false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code

        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
            this.label1 = new System.Windows.Forms.Label();
            this.textBoxSMS = new System.Windows.Forms.TextBox();
            this.buttonCancel = new System.Windows.Forms.Button();
            this.buttonOk = new System.Windows.Forms.Button();
            this.SuspendLayout();
            //
            // label1
            //
            this.label1.AutoSize = true;
```

```
            this.label1.Location = new System.Drawing.Point(12, 9);
            this.label1.Name = "label1";
            this.label1.Size = new System.Drawing.Size(50, 13);
            this.label1.TabIndex = 0;
            this.label1.Text = "Message";
            //
            // textBoxSMS
            //
            this.textBoxSMS.Location = new System.Drawing.Point(15, 25);
            this.textBoxSMS.Multiline = true;
            this.textBoxSMS.Name = "textBoxSMS";
            this.textBoxSMS.Size = new System.Drawing.Size(352, 53);
            this.textBoxSMS.TabIndex = 1;
            //
            // buttonCancel
            //
            this.buttonCancel.DialogResult = System.Windows.Forms.DialogResult.Cancel;
            this.buttonCancel.Location = new System.Drawing.Point(95, 85);
            this.buttonCancel.Name = "buttonCancel";
            this.buttonCancel.Size = new System.Drawing.Size(75, 23);
            this.buttonCancel.TabIndex = 2;
            this.buttonCancel.Text = "&Cancel";
            this.buttonCancel.UseVisualStyleBackColor = true;
            //
            // buttonOk
            //
            this.buttonOk.Location = new System.Drawing.Point(186, 85);
            this.buttonOk.Name = "buttonOk";
            this.buttonOk.Size = new System.Drawing.Size(75, 23);
            this.buttonOk.TabIndex = 3;
            this.buttonOk.Text = "&Ok";
            this.buttonOk.UseVisualStyleBackColor = true;
            this.buttonOk.Click += new System.EventHandler(this.buttonOk_Click);
            //
            // FormSMSNotification
            //
            this.AcceptButton = this.buttonOk;
            this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
            this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
            this.CancelButton = this.buttonCancel;
            this.ClientSize = new System.Drawing.Size(389, 120);
            this.Controls.Add(this.buttonOk);
            this.Controls.Add(this.buttonCancel);
            this.Controls.Add(this.textBoxSMS);
            this.Controls.Add(this.label1);
            this.Name = "FormSMSNotification";
            this.StartPosition = System.Windows.Forms.FormStartPosition.CenterParent;
            this.Text = "Message";
            this.Load += new System.EventHandler(this.FormSMSNotification_Load);
            this.ResumeLayout(false);
            this.PerformLayout();

        }

        #endregion

        private System.Windows.Forms.Label label1;
        private System.Windows.Forms.TextBox textBoxSMS;
        private System.Windows.Forms.Button buttonCancel;
        private System.Windows.Forms.Button buttonOk;
    }
}

namespace ParkingManager
{
```

```
    partial class FormTag
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should be disposed;
otherwise, false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code

        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
            this.txtTagNumber = new System.Windows.Forms.TextBox();
            this.label1 = new System.Windows.Forms.Label();
            this.label2 = new System.Windows.Forms.Label();
            this.txtCustomerName = new System.Windows.Forms.TextBox();
            this.label3 = new System.Windows.Forms.Label();
            this.txtCustomerMobile = new System.Windows.Forms.TextBox();
            this.label4 = new System.Windows.Forms.Label();
            this.dateTimeTagExpiration = new System.Windows.Forms.DateTimePicker();
            this.buttonOk = new System.Windows.Forms.Button();
            this.buttonCancel = new System.Windows.Forms.Button();
            this.buttonPhoto = new System.Windows.Forms.Button();
            this.pictureBox1 = new System.Windows.Forms.PictureBox();
            this.label5 = new System.Windows.Forms.Label();
            this.textBoxEmail = new System.Windows.Forms.TextBox();
            this.groupBox1 = new System.Windows.Forms.GroupBox();
            this.checkedListBoxSmsSubscriptions = new
System.Windows.Forms.CheckedListBox();

((System.ComponentModel.ISupportInitialize)(this.pictureBox1)).BeginInit();
            this.groupBox1.SuspendLayout();
            this.SuspendLayout();
            //
            // txtTagNumber
            //
            this.txtTagNumber.Location = new System.Drawing.Point(97, 24);
            this.txtTagNumber.Name = "txtTagNumber";
            this.txtTagNumber.Size = new System.Drawing.Size(129, 20);
            this.txtTagNumber.TabIndex = 0;
            //
            // label1
            //
            this.label1.AutoSize = true;
            this.label1.Location = new System.Drawing.Point(12, 31);
            this.label1.Name = "label1";
            this.label1.Size = new System.Drawing.Size(69, 13);
            this.label1.TabIndex = 1;
```

```
this.label1.Text = "TAG Number";
//
// label2
//
this.label2.AutoSize = true;
this.label2.Location = new System.Drawing.Point(12, 53);
this.label2.Name = "label2";
this.label2.Size = new System.Drawing.Size(82, 13);
this.label2.TabIndex = 3;
this.label2.Text = "Customer Name";
//
// txtCustomerName
//
this.txtCustomerName.Location = new System.Drawing.Point(97, 50);
this.txtCustomerName.Name = "txtCustomerName";
this.txtCustomerName.Size = new System.Drawing.Size(129, 20);
this.txtCustomerName.TabIndex = 2;
//
// label3
//
this.label3.AutoSize = true;
this.label3.Location = new System.Drawing.Point(12, 79);
this.label3.Name = "label3";
this.label3.Size = new System.Drawing.Size(85, 13);
this.label3.TabIndex = 5;
this.label3.Text = "Customer Mobile";
//
// txtCustomerMobile
//
this.txtCustomerMobile.Location = new System.Drawing.Point(97, 76);
this.txtCustomerMobile.Name = "txtCustomerMobile";
this.txtCustomerMobile.Size = new System.Drawing.Size(129, 20);
this.txtCustomerMobile.TabIndex = 4;
//
// label4
//
this.label4.AutoSize = true;
this.label4.Location = new System.Drawing.Point(12, 135);
this.label4.Name = "label4";
this.label4.Size = new System.Drawing.Size(79, 13);
this.label4.TabIndex = 7;
this.label4.Text = "Expiration Date";
//
// dateTimeTagExpiration
//
this.dateTimeTagExpiration.Location = new System.Drawing.Point(97, 129);
this.dateTimeTagExpiration.Name = "dateTimeTagExpiration";
this.dateTimeTagExpiration.Size = new System.Drawing.Size(200, 20);
this.dateTimeTagExpiration.TabIndex = 8;
//
// buttonOk
//
this.buttonOk.Location = new System.Drawing.Point(73, 425);
this.buttonOk.Name = "buttonOk";
this.buttonOk.Size = new System.Drawing.Size(75, 23);
this.buttonOk.TabIndex = 9;
this.buttonOk.Text = "Ok";
this.buttonOk.UseVisualStyleBackColor = true;
this.buttonOk.Click += new System.EventHandler(this.buttonOk_Click);
//
// buttonCancel
//
this.buttonCancel.DialogResult = System.Windows.Forms.DialogResult.Cancel;
this.buttonCancel.Location = new System.Drawing.Point(166, 425);
this.buttonCancel.Name = "buttonCancel";
```

```
            this.buttonCancel.Size = new System.Drawing.Size(75, 23);
            this.buttonCancel.TabIndex = 10;
            this.buttonCancel.Text = "Cancel";
            this.buttonCancel.UseVisualStyleBackColor = true;
            this.buttonCancel.Click += new
System.EventHandler(this.buttonCancel_Click);
            //
            // buttonPhoto
            //
            this.buttonPhoto.Location = new System.Drawing.Point(15, 168);
            this.buttonPhoto.Name = "buttonPhoto";
            this.buttonPhoto.Size = new System.Drawing.Size(75, 23);
            this.buttonPhoto.TabIndex = 12;
            this.buttonPhoto.Text = "Photo";
            this.buttonPhoto.UseVisualStyleBackColor = true;
            this.buttonPhoto.Click += new System.EventHandler(this.buttonPhoto_Click);
            //
            // pictureBox1
            //
            this.pictureBox1.Location = new System.Drawing.Point(96, 168);
            this.pictureBox1.Name = "pictureBox1";
            this.pictureBox1.Size = new System.Drawing.Size(201, 151);
            this.pictureBox1.TabIndex = 13;
            this.pictureBox1.TabStop = false;
            //
            // label5
            //
            this.label5.AutoSize = true;
            this.label5.Location = new System.Drawing.Point(12, 105);
            this.label5.Name = "label5";
            this.label5.Size = new System.Drawing.Size(79, 13);
            this.label5.TabIndex = 15;
            this.label5.Text = "Customer Email";
            //
            // textBoxEmail
            //
            this.textBoxEmail.Location = new System.Drawing.Point(97, 102);
            this.textBoxEmail.Name = "textBoxEmail";
            this.textBoxEmail.Size = new System.Drawing.Size(129, 20);
            this.textBoxEmail.TabIndex = 14;
            //
            // groupBox1
            //
            this.groupBox1.Controls.Add(this.checkedListBoxSmsSubscriptions);
            this.groupBox1.Location = new System.Drawing.Point(15, 325);
            this.groupBox1.Name = "groupBox1";
            this.groupBox1.Size = new System.Drawing.Size(282, 84);
            this.groupBox1.TabIndex = 16;
            this.groupBox1.TabStop = false;
            this.groupBox1.Text = "SMS Subscriptions";
            //
            // checkedListBoxSmsSubscriptions
            //
            this.checkedListBoxSmsSubscriptions.Dock =
System.Windows.Forms.DockStyle.Fill;
            this.checkedListBoxSmsSubscriptions.FormattingEnabled = true;
            this.checkedListBoxSmsSubscriptions.Location = new System.Drawing.Point(3,
16);
            this.checkedListBoxSmsSubscriptions.Name =
"checkedListBoxSmsSubscriptions";
            this.checkedListBoxSmsSubscriptions.ScrollAlwaysVisible = true;
            this.checkedListBoxSmsSubscriptions.Size = new System.Drawing.Size(276,
65);
            this.checkedListBoxSmsSubscriptions.TabIndex = 0;
```

```
            this.checkedListBoxSmsSubscriptions.ItemCheck += new
System.Windows.Forms.ItemCheckEventHandler(this.checkedListBoxSmsSubscriptions_ItemChe
ck);
            //
            // FormTag
            //
            this.AcceptButton = this.buttonOk;
            this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
            this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
            this.CancelButton = this.buttonCancel;
            this.ClientSize = new System.Drawing.Size(318, 460);
            this.Controls.Add(this.groupBox1);
            this.Controls.Add(this.label5);
            this.Controls.Add(this.textBoxEmail);
            this.Controls.Add(this.pictureBox1);
            this.Controls.Add(this.buttonPhoto);
            this.Controls.Add(this.buttonCancel);
            this.Controls.Add(this.buttonOk);
            this.Controls.Add(this.dateTimeTagExpiration);
            this.Controls.Add(this.label4);
            this.Controls.Add(this.label3);
            this.Controls.Add(this.txtCustomerMobile);
            this.Controls.Add(this.label2);
            this.Controls.Add(this.txtCustomerName);
            this.Controls.Add(this.label1);
            this.Controls.Add(this.txtTagNumber);
            this.Name = "FormTag";
            this.StartPosition = System.Windows.Forms.FormStartPosition.CenterParent;
            this.Text = "FormTag";
            this.Load += new System.EventHandler(this.FormTag_Load);
            ((System.ComponentModel.ISupportInitialize)(this.pictureBox1)).EndInit();
            this.groupBox1.ResumeLayout(false);
            this.ResumeLayout(false);
            this.PerformLayout();

        }

        #endregion

        private System.Windows.Forms.TextBox txtTagNumber;
        private System.Windows.Forms.Label label1;
        private System.Windows.Forms.Label label2;
        private System.Windows.Forms.TextBox txtCustomerName;
        private System.Windows.Forms.Label label3;
        private System.Windows.Forms.TextBox txtCustomerMobile;
        private System.Windows.Forms.Label label4;
        private System.Windows.Forms.DateTimePicker dateTimeTagExpiration;
        private System.Windows.Forms.Button buttonOk;
        private System.Windows.Forms.Button buttonCancel;
        private System.Windows.Forms.Button buttonPhoto;
        private System.Windows.Forms.PictureBox pictureBox1;
        private System.Windows.Forms.Label label5;
        private System.Windows.Forms.TextBox textBoxEmail;
        private System.Windows.Forms.GroupBox groupBox1;
        private System.Windows.Forms.CheckedListBox checkedListBoxSmsSubscriptions;
    }
}
using System;
using System.Collections.Generic;
using System.Diagnostics;
using System.IO;
using System.Net;
using System.Text;
using System.Windows.Forms;
```

```
namespace ParkingManager
{
   public class Logic
    {
       public static bool SendSMS(string account, string password, string api, string
phone, string subject, string body, bool showProviderMsg)
       {
           bool success = true;

          string value =  SendSmsClickatell(account, password, api,
phone,subject,body);
          if (showProviderMsg)
          {
              MessageBox.Show(value, "SMS Provider");
          }

           return success;
       }

       public static string SendSmsClickatell(string account, string password, string
api, string phone, string subject, string body)
       {
          // bool success = true;

           WebClient client = new WebClient();

           client.Headers.Add("user-agent", "Mozilla/4.0 compatible; MSIE 6.0; Windows
NT 5.2; .NET CLR 1.0.3705;)");
           client.QueryString.Add("user", account);
           client.QueryString.Add("password", password);
           client.QueryString.Add("api_id", api);
           client.QueryString.Add("to", phone);
           client.QueryString.Add("text", body);
           string baseurl = "http://api.clickatell.com/http/sendmsg";
           Stream data = client.OpenRead(baseurl);
           StreamReader reader = new StreamReader(data);
           string s = reader.ReadToEnd();
           Debug.WriteLine(s);
           data.Close();
           reader.Close();
           return s;

          // return success;
       }
    }
}

using System;
using System.Collections.Generic;
using System.Text;

namespace ParkingManager
{
    [Serializable]
    public class LogRecord
    {
        public TagCard Tag;
        public DateTime Time;
        public int GateNumber;
        public bool GoingIn;



    }
```

```
}

using System;
using System.Collections.Generic;
using System.Windows.Forms;

namespace ParkingManager
{
    static class Program
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new FormMain());
        }
    }
}

using System;
using System.Collections.Generic;
using System.Drawing;
using System.Text;

namespace ParkingManager
{
    [Serializable]
    public class TagCard
    {
        public string TagID;
        public string CustomerName;
        public string CustomerNID;
        public string CustomerMobile;
        public string CustomerEmail;
        public bool NotificationSent = false;
        public DateTime Expiration;
        public bool CustomerHasSecurityClearance;
        public Image Photo;
        public List<string> SmsSubscriptions = new List<string>();
    }
}
```

```
using System;
using System.Collections.Generic;
using System.Drawing;
using System.IO;
using System.Runtime.Serialization.Formatters.Binary;
using System.Text;
using System.Windows.Forms;

namespace ParkingManager
{
    [Serializable]
    public class Database
    {
        public string PortName;
        public string PortSpeed;
        public bool TagValidationRule1;
        public bool EnableNotification;
        public int NotficationDays;

        public string GmailAccount;
        public string GmailPassword;
        public string SmsAccount;
        public string SmsPassword;
        public string SmsApi;


        public Dictionary<string, TagCard> Tags = new Dictionary<string, TagCard>();
        public List<LogRecord> Records = new List<LogRecord>();
        public Dictionary<string, Image> Photos = new Dictionary<string, Image>();

        public Dictionary<string, List<string>> SmsSubscriptions = new
Dictionary<string, List<string>>();



        public void SaveDatabase(string file)
        {
            BinaryFormatter f = new BinaryFormatter();
            FileStream fs = new FileStream(file, FileMode.Create, FileAccess.Write,
FileShare.Read);
            f.Serialize(fs, this);
            fs.Close();

        }
```

```
        public static Database ReadDatabase(string file)
        {
            FileStream fs = new FileStream(file, FileMode.Open, FileAccess.Read,
FileShare.ReadWrite);
            BinaryFormatter f = new BinaryFormatter();
            Database db = (Database)f.Deserialize(fs);
            fs.Close();

            return db;

        }

    }
}


namespace ParkingManager
{
    partial class FormAbout
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should be disposed;
otherwise, false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code

        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
            this.label1 = new System.Windows.Forms.Label();
            this.textBox1 = new System.Windows.Forms.TextBox();
            this.buttonOk = new System.Windows.Forms.Button();
            this.SuspendLayout();
            //
            // label1
            //
            this.label1.AutoSize = true;
            this.label1.Location = new System.Drawing.Point(12, 18);
            this.label1.Name = "label1";
            this.label1.Size = new System.Drawing.Size(189, 13);
            this.label1.TabIndex = 2;
            this.label1.Text = "Parking Monitoring and Control System";
            //
            // textBox1
            //
            this.textBox1.Location = new System.Drawing.Point(15, 44);
```

```
            this.textBox1.Multiline = true;
            this.textBox1.Name = "textBox1";
            this.textBox1.ReadOnly = true;
            this.textBox1.Size = new System.Drawing.Size(290, 139);
            this.textBox1.TabIndex = 1;
            //
            // buttonOk
            //
            this.buttonOk.Location = new System.Drawing.Point(126, 191);
            this.buttonOk.Name = "buttonOk";
            this.buttonOk.Size = new System.Drawing.Size(75, 23);
            this.buttonOk.TabIndex = 0;
            this.buttonOk.Text = "Ok";
            this.buttonOk.UseVisualStyleBackColor = true;
            this.buttonOk.Click += new System.EventHandler(this.buttonOk_Click);
            //
            // FormAbout
            //
            this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
            this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
            this.ClientSize = new System.Drawing.Size(317, 226);
            this.Controls.Add(this.buttonOk);
            this.Controls.Add(this.textBox1);
            this.Controls.Add(this.label1);
            this.Name = "FormAbout";
            this.Text = "About";
            this.Load += new System.EventHandler(this.FormAbout_Load);
            this.ResumeLayout(false);
            this.PerformLayout();

        }

        #endregion

        private System.Windows.Forms.Label label1;
        private System.Windows.Forms.TextBox textBox1;
        private System.Windows.Forms.Button buttonOk;
    }
}
```

# 17Appendix D: Bill of Materials

Bills of all components bought &used, quantities and cost

# SparkFun Invoice

## #654618

6175 Longbow Drive • Boulder, CO 80301
1-303-284-0979

Invoice Created: 2012-12-15 12:05:36
Ordered: 2012-12-08 09:25:36

**Billing**
Mustafa Al Qurain
Saudi arabia
Safwa, 31921
Saudi Arabia

**Delivery**
Mustafa Al Qurain
Saudi arabia
Safwa, 31921
Saudi Arabia
00966505877917
Qurainmh@gmail.com

**Shipping Method: FedEx International Priority**
Tracking Number: 794255318049
**Payment Method:** Credit Card
**Order Status:** Shipped 2012-12-09

| SKU | Product Name | Qty | Price | Total |
|---|---|---|---|---|
| BOB-08688 | TEMT6000 Breakout Board | 1 | $4.95 | $4.95 |
| COM-08227 | Humidity and Temperature Sensor - SHT15 | 1 | $28.95 | $28.95 |
| COM-08348 | Ambient Light Sensor - TEMT6000 | 1 | $1.50 | $1.50 |
| COM-11139 | Rocker Switch - DPDT | 1 | $0.95 | $0.95 |
| PRT-10088 | Arduino Project Enclosure | 1 | $11.95 | $11.95 |
| PRT-11351 | Enclosure - Aluminum (120x95x35mm) | 1 | $5.95 | $5.95 |
| SEN-08257 | Humidity and Temperature Sensor - SHT15 Breakout | 1 | $41.95 | $41.95 |
| SEN-08630 | PIR Motion Sensor | 1 | $9.95 | $9.95 |
| SEN-09404 | Methane CNG Gas Sensor - MQ-4 | 1 | $4.95 | $4.95 |
| SEN-11610 | LinkSprite JPEG Color Camera TTL Interface - Infrared | 1 | $49.95 | $49.95 |

**SUPPLIER'S CERTIFICATION OF COMPLIANCE**
It is hereby certified that the product(s) provided in this shipment conform to the requirements and the manufacturer's part number identified in the customer's purchase order and these parts have been received, stored and shipped by SparkFun Electronics.

Signature, SparkFun Customer Service Manager

| | |
|---|---|
| Subtotal | $161.05 |
| Shipping/Handling | $33.05 |
| Grand Total | $194.10 |
| **Transactions:** | |
| Dec-08 Credit Card 9161 | $194.10 |
| **Order Balance:** | **$0.00** |

Please direct any questions or concerns to customerservice@sparkfun.com or 303.284.0979.

Page 1 of 1

# SparkFun Invoice

# #644146

6175 Longbow Drive • Boulder, CO 80301
1-303-284-0979

Invoice Created: 2012-12-15 12:10:27
Ordered: 2012-11-24 21:50:14

**Billing**
Mustafa Al Qurain
Saudi arabia
Safwa, 31921
Saudi Arabia

**Delivery**
Mustafa Al Qurain
Saudi arabia
Safwa, 31921
Saudi Arabia
00966505877917
Qurainmh@gmail.com

## Shipping Method: FedEx International Priority

Tracking Number: 794143045129
**Payment Method:** Credit Card
**Order Status:** Shipped 2012-11-26

| SKU | Product Name | Qty | Price | Total |
|---|---|---|---|---|
| PRT-10007 | Arduino Stackable Header Kit | 1 | $1.50 | $1.50 |
| SEN-08419 | RFID Reader ID-12 (125 kHz) | 1 | $29.95 | $29.95 |
| SEN-08423 | RFID Reader Breakout | 1 | $0.95 | $0.95 |

**SUPPLIER'S CERTIFICATION OF COMPLIANCE**
It is hereby certified that the product(s) provided in this shipment conform to the requirements and the manufacturer's part number identified in the customer's purchase order and these parts have been received, stored and shipped by SparkFun Electronics.

Signature, SparkFun Customer Service Manager

| | |
|---|---|
| Subtotal | $32.40 |
| Shipping/Handling | $33.79 |
| Grand Total | $66.19 |
| **Transactions:** | |
| Nov-24 Credit Card 9161 | $66.19 |
| **Order Balance:** | **$0.00** |

Please direct any questions or concerns to customerservice@sparkfun.com or 303.284.0979.

# SparkFun Invoice

# #624232

6175 Longbow Drive • Boulder, CO 80301
1-303-284-0979

Invoice Created: 2012-12-15 12:14:30
Ordered: 2012-10-18 08:59:44

**Billing**
Mustafa Al Qurain
Saudi arabia
Safwa, 31921
Saudi Arabia

**Delivery**
Mustafa Al Qurain
Saudi Aramco
Albaironi st.
Safwa city , 31921
Saudi Arabia
00966505877917
Qurainmh@gmail.com

## Shipping Method: FedEx International Priority

Tracking Number: 799225047756
**Payment Method:** Credit Card
**Order Status:** Shipped 2012-10-18

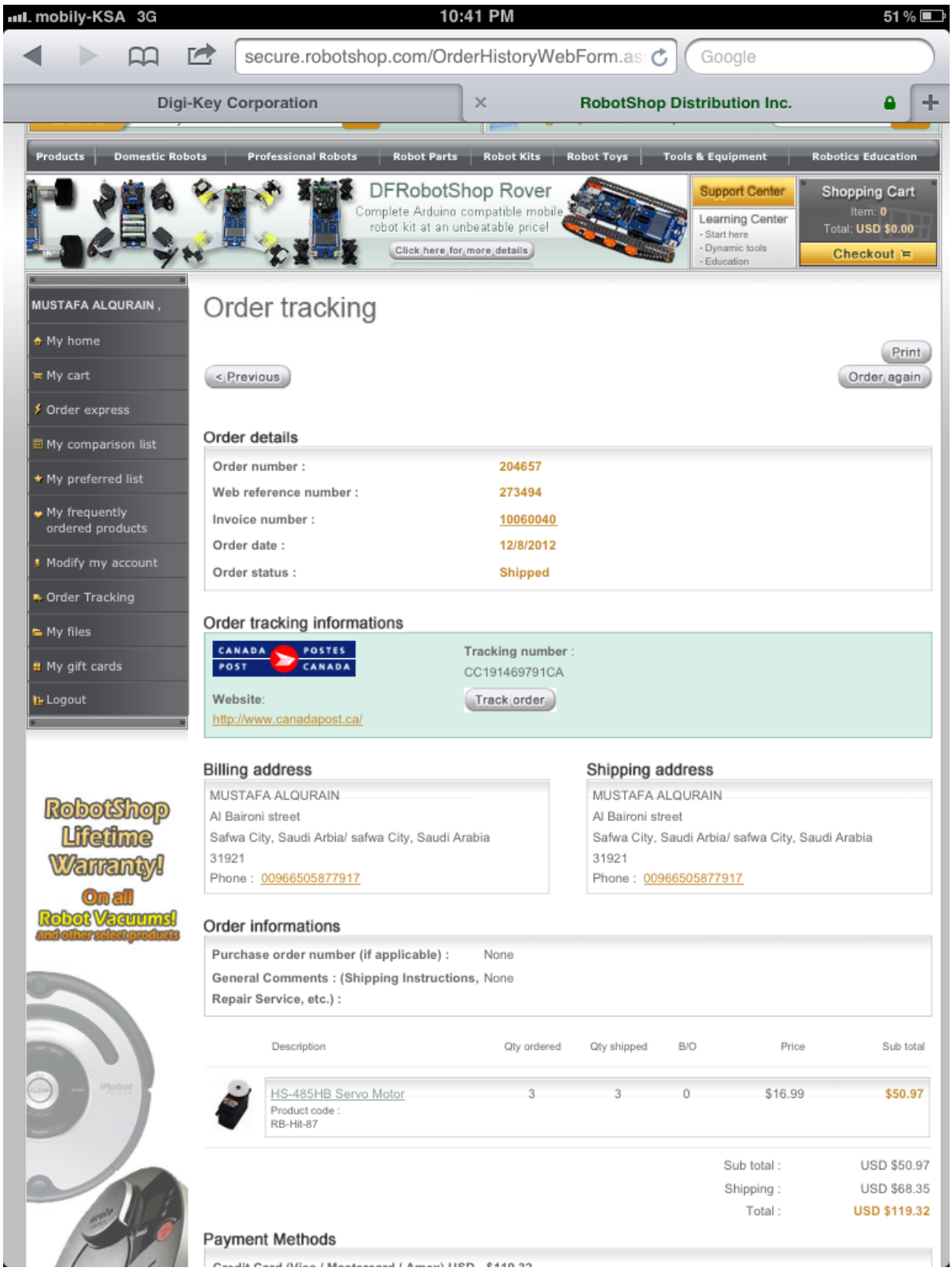
| SKU | Product Name | Qty | Price | Total |
|---|---|---|---|---|
| CEL-00675 | Quad-band Cellular Duck Antenna SMA | 1 | $7.95 | $7.95 |
| PRT-10007 | Arduino Stackable Header Kit | 1 | $1.50 | $1.50 |

**SUPPLIER'S CERTIFICATION OF COMPLIANCE**
It is hereby certified that the product(s) provided in this shipment conform to the requirements and the manufacturer's part number identified in the customer's purchase order and these parts have been received, stored and shipped by SparkFun Electronics.

Signature, SparkFun Customer Service Manager

| | |
|---|---|
| Subtotal | $9.45 |
| Shipping/Handling | $33.65 |
| Grand Total | $43.10 |
| **Transactions:** | |
| Oct-18 Credit Card 9161 | $43.10 |
| **Order Balance:** | **$0.00** |

Please direct any questions or concerns to customerservice@sparkfun.com or 303.284.0979.


mobily-KSA 3G
10:41 PM
51 %
secure.robotshop.com/OrderHistoryWebForm.as
Google
Digi-Key Corporation
RobotShop Distribution Inc.
Products
Domestic Robots
Professional Robots
Robot Parts
Robot Kits
Robot Toys
Tools & Equipment
Robotics Education
DFRobotShop Rover
Complete Arduino compatible mobile robot kit at an unbeatable price!
Click here for more details
Support Center
Learning Center
- Start here
- Dynamic tools
- Education
Shopping Cart
Item: 0
Total: USD $0.00
Checkout
MUSTAFA ALQURAIN ,
My home
My cart
Order express
My comparison list
My preferred list
My frequently ordered products
Modify my account
Order Tracking
My files
My gift cards
Logout
Order tracking
Print
< Previous
Order again
Order details
Order number : 204657
Web reference number : 273494
Invoice number : 10060040
Order date : 12/8/2012
Order status : Shipped
Order tracking informations
CANADA POSTES POST CANADA
Tracking number :
CC191469791CA
Website:
http://www.canadapost.ca/
Track order
Billing address
MUSTAFA ALQURAIN
Al Baironi street
Safwa City, Saudi Arbia/ safwa City, Saudi Arabia
31921
Phone : 00966505877917
Shipping address
MUSTAFA ALQURAIN
Al Baironi street
Safwa City, Saudi Arbia/ safwa City, Saudi Arabia
31921
Phone : 00966505877917
Order informations
Purchase order number (if applicable) : None
General Comments : (Shipping Instructions, Repair Service, etc.) : None
Description
Qty ordered
Qty shipped
B/O
Price
Sub total
HS-485HB Servo Motor
Product code : RB-Hit-87
3
3
0
$16.99
$50.97
Sub total : USD $50.97
Shipping : USD $68.35
Total : USD $119.32
Payment Methods
RobotShop Lifetime Warranty!
On all Robot Vacuums!
and other select products

# SparkFun Invoice

# #569718

6175 Longbow Drive • Boulder, CO 80301
1-303-284-0979

Invoice Created: 2012-07-15 03:05:22
Ordered: 2012-07-15 15:04:24

**Billing**
Salman Albelaihi
Mob 0506234617
Aramco Udhailiyah
Udhailiyah, Alhasa 31311
Saudi Arabia

**Delivery**
Salman Albelaihi
Mob 0506234617
Aramco Udhailiyah
Udhailiyah, Alhasa 31311
Saudi Arabia
+966506234617
albelasa@hotmail.com

## Shipping Method: FedEx International Priority

**Payment Method:** PayPal Express Checkout
**Order Status:** New Order

| SKU | Product Name | UPC | Qty | Price | Total |
|---|---|---|---|---|---|
| CEL-09607 | Cellular Shield with SM5100B | | 1 | $99.95 | $99.95 |
| COM-09151 | Speaker - 0.5W 8Ohm | | 1 | $1.95 | $1.95 |
| COM-09578 | SpeakJet | | 1 | $24.95 | $24.95 |
| COM-09811 | Text to Speech chip for SpeakJet - TTS256 | | 1 | $21.95 | $21.95 |
| COM-10169 | RFID Tag - 125kHz | 845156000272 | 6 | $1.95 | $11.70 |
| DEV-11061 | Arduino Mega 2560 R3 | | 1 | $58.95 | $58.95 |
| PRT-10112 | 2mm 10pin XBee Header | | 2 | $0.95 | $1.90 |
| RTL-10782 | VoiceBox Shield Retail | 845156001088 | 1 | $41.95 | $41.95 |
| SEN-08419 | RFID Reader ID-12 (125 kHz) | | 2 | $29.95 | $59.90 |
| SEN-08423 | RFID Reader Breakout | | 2 | $0.95 | $1.90 |

**SUPPLIER'S CERTIFICATION OF COMPLIANCE**
It is hereby certified that the product(s) provided in this shipment conform to the requirements and the manufacturer's part number identified in the customer's purchase order and these parts have been received, stored and shipped by SparkFun Electronics.

Signature, SparkFun Customer Service Manager

| | |
|---|---|
| Subtotal | $325.10 |
| Shipping/Handling | $35.87 |
| Grand Total | $360.97 |
| **Transactions:** | |
| Jul-15 Paypal | $360.97 |
| albelasa@gmail.com | |
| **Order Balance:** | **$0.00** |

Please direct any questions or concerns to customerservice@sparkfun.com or 303.284.0979.

Page 1 of 1

**ADAFRUIT INDUSTRIES**
**80 NASSAU ST #4C**
**NEW YORK NY 10038**
**FAX: (917) 210-3397**
**TEL: (646) 248-7822 / (OHM) BIT-STAB**

**SOLD TO:**
Salman Albelaihi
Aramco Udhailiyah 993
0506234617
Udhailiyah, Hasa 31311
Saudi Arabia

7185538740
albelasa@gmail.com

**SHIP TO:**
Salman Albelaihi
Aramco Udhailiyah 993
0506234617
Udhailiyah, 31311
Hasa, Saudi Arabia

**Invoice No. 152237**
**Date Ordered:** Sunday 10 June, 2012
**Payment Method:** Credit Card

| | Item Name | Model | Tax | Price (ex) | Price (inc) | Total (ex) | Total (inc) |
|---|---|---|---|---|---|---|---|
| ( )  | 1 x DSO Nano v2 - Pocket-size color digital oscilloscope | PID:468 v2.0 | 0% | $99.95 | $99.95 | $99.95 | $99.95 |
| ( )  | 1 x Budget Pack for Arduino (Arduino Uno R3) | PID:193 Uno w/328 | 0% | $49.50 | $49.50 | $49.50 | $49.50 |
| ( )  | 1 x 9 VDC 1000mA regulated switching power adapter - UL listed | PID:63 | 0% | $6.95 | $6.95 | $6.95 | $6.95 |
| ( )  | 1 x 9V battery holder with switch & 5.5mm/2.1mm plug | PID:67 | 0% | $3.95 | $3.95 | $3.95 | $3.95 |
| ( )  | 2 x IR distance sensor w/cable (10cm-80cm) | PID:164 GP2Y0A21YK0F | 0% | $13.00 | $13.00 | $26.00 | $26.00 |
| ( )  | 1 x Pocket Autoranging Digital Multimeter | PID:850 | 0% | $24.95 | $24.95 | $24.95 | $24.95 |
| ( )  | 1 x 16x24 Red LED Matrix Panel - Chainable HT1632C Driver | PID:555 | 0% | $24.95 | $24.95 | $24.95 | $24.95 |
| | 1 x Premium Female/Female Jumper Wires - 40 x 3" (75mm) | PID:794 | 0% | $5.95 | $5.95 | $5.95 | $5.95 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| ( )  | 1 x Premium Male/Male Jumper Wires - 40 x 3" (75mm) | PID:759 | 0% | $6.95 | $6.95 | $6.95 | $6.95 |
| ( )  | 1 x Premium Female/Male 'Extension' Jumper Wires - 40 x 3" (75mm) | PID:825 | 0% | $6.95 | $6.95 | $6.95 | $6.95 |
| ( )  | 1 x Open source hardware - Sticker! | PID:693 | 0% | $1.00 | $1.00 | $1.00 | $0.00 (1 FREE) |

**Sub-Total:** $256.10
**Sales Tax:** $0.00
**United Parcel Service (1 pkg x 2.44 lbs total) (UPS Worldwide Expedited *(recommended)*):** $58.75
**Total:** $314.85

**Notes:**
My Mobile Number is 0506234617

1Z71EY056790728845

---

**ADAFRUIT INDUSTRIES**
**80 NASSAU ST #4C**
**NEW YORK NY 10038**
**FAX: (917) 210-3397**
**TEL: (646) 248-7822 / (OHM) BIT-STAB**

## COMMERCIAL INVOICE

**SOLD BY:**
Adafruit Industries
80 Nassau st #4C
New York NY 10038
Fax: (917) 210-3397
Tel: (646) 248-7822 / (ohm) bit-stab

**CUSTOMER:**
Salman Albelaihi
182-21 150th Avenue
DHA 4446
Springfield Gardens, New York 11413
United States

**SOLD TO:**
Salman Albelaihi
Aramco Udhailiyah 993
0506234617
Udhailiyah, Hasa 31311
Saudi Arabia

7185538740
albelasa@gmail.com

**SHIP TO:**
Salman Albelaihi
Aramco Udhailiyah 993
0506234617
Udhailiyah, 31311
Hasa, Saudi Arabia

**Invoice No. 152237**
**Date Ordered:** Sunday 10 June, 2012
**Payment Method:** Credit Card

| **Quantity** (pieces) | **Item Name** | **Model** | **Tax** | **Price (ex)** | **Total (ex)** |
|---|---|---|---|---|---|

| | | | | | |
|---|---|---|---|---|---|
| 1<br>9030.83.0000<br>China<br>Digital storage oscilloscope | DSO Nano v2 - Pocket-size color digital oscilloscope | v2.0 | 0% | **$99.95** | **$99.95** |
| 1<br>8542.90.0000<br>United States<br>Educational electronic parts kit | Budget Pack for Arduino (Arduino Uno R3) | Uno w/328 | 0% | **$49.50** | **$49.50** |
| 1<br>8504.40.9510<br>China<br>Electrical transformer, rectifying power supply, <50W | 9 VDC 1000mA regulated switching power adapter - UL listed | | 0% | **$6.95** | **$6.95** |
| 1<br>8536.9010<br>China<br>Battery holder/connector | 9V battery holder with switch & 5.5mm/2.1mm plug | | 0% | **$3.95** | **$3.95** |
| 2<br>8541.40.95.00<br>China<br>Photosensitive semiconductor devices, light-emitting diodes: Other | IR distance sensor w/cable (10cm-80cm) | GP2Y0A21YK0F | 0% | **$13.00** | **$26.00** |
| 1<br>9030.31.0000<br>China<br>Instruments for measuring voltage multimeters w/o recording | Pocket Autoranging Digital Multimeter | | 0% | **$24.95** | **$24.95** |
| 1<br>8541.40.2000<br>China<br>LEDs | 16x24 Red LED Matrix Panel - Chainable HT1632C Driver | | 0% | **$24.95** | **$24.95** |
| 1<br>8544.42.9000<br>China<br>Insulated cable, fitted with connectors | Premium Female/Female Jumper Wires - 40 x 3" (75mm) | | 0% | **$5.95** | **$5.95** |
| 1<br>8544.42.9000<br>China<br>Insulated cable, fitted with connectors | Premium Male/Male Jumper Wires - 40 x 3" (75mm) | | 0% | **$6.95** | **$6.95** |
| 1<br>8544.42.9000<br>China<br>Insulated cable, fitted with connectors | Premium Female/Male 'Extension' Jumper Wires - 40 x 3" (75mm) | | 0% | **$6.95** | **$6.95** |
| 1<br>3919.90.5060<br>United States<br>Vinyl sticker | Open source hardware - Sticker! | | 0% | **$1.00** | **$1.00** |



Adafruit Industries, Unique & fun DIY electronics and kits

| | |
|---|---|
| **Sub-Total:** | $256.10 |
| **Sales Tax:** | $0.00 |
| **United Parcel Service (1 pkg x 2.44 lbs total) (UPS Worldwide Expedited *(recommended)*):** | $58.75 |
| **Total:** | **$314.85** |

**ADAFRUIT INDUSTRIES**
**80 NASSAU ST #4C**
**NEW YORK NY 10038**
**FAX: (917) 210-3397**
**TEL: (646) 248-7822 / (OHM) BIT-STAB**

**CUSTOMER:**
Salman Albelaihi
182-21 150th Avenue
DHA 4446
Springfield Gardens, New York 11413
United States

**SOLD TO:**
Salman Albelaihi
Aramco Udhailiyah 993
0506234617
Udhailiyah, Hasa 31311
Saudi Arabia

7185538740
albelasa@gmail.com

**SHIP TO:**
Salman Albelaihi
Aramco Udhailiyah
993
Udhailiyah, 31311
Hasa, Saudi Arabia

**Invoice No. 160198**
**Date Ordered:** Sunday 15 July, 2012
**Payment Method:** PayPal

| | Item Name | Model | Tax | Price (ex) | Price (inc) | Total (ex) | Total (inc) |
|---|---|---|---|---|---|---|---|
| ( )  | 1 x 16x24 Red LED Matrix Panel - Chainable HT1632C Driver | PID:555 | 0% | **$24.95** | **$24.95** | **$24.95** | **$24.95** |

**Sub-Total:** $24.95
**Sales Tax:** $0.00
**United States Postal Service (0.55lbs) (First-Class Mail International Parcel** incl. $1.60 insurance**):** $12.32
**Total:** **$37.27**

**ADAFRUIT INDUSTRIES**
**80 NASSAU ST #4C**
**NEW YORK NY 10038**
**FAX: (917) 210-3397**
**TEL: (646) 248-7822 / (OHM) BIT-STAB**

## COMMERCIAL INVOICE

**SOLD BY:**
Adafruit Industries
80 Nassau st #4C
New York NY 10038
Fax: (917) 210-3397
Tel: (646) 248-7822 / (ohm) bit-stab

**CUSTOMER:**
Salman Albelaihi
182-21 150th Avenue
DHA 4446
Springfield Gardens, New York 11413
United States

**SOLD TO:**
Salman Albelaihi
Aramco Udhailiyah 993
0506234617
Udhailiyah, Hasa 31311
Saudi Arabia

7185538740

**SHIP TO:**
Salman Albelaihi
Aramco Udhailiyah
993
Udhailiyah, 31311
Hasa, Saudi Arabia

Adafruit Industries, Unique & fun DIY electronics and kits

albelasa@gmail.com

**Invoice No. 160198**

**Date Ordered:** Sunday 15 July, 2012
**Payment Method:** PayPal

| **Quantity** (pieces) | **Item Name** | **Model** | **Tax** | **Price (ex)** | **Total (ex)** |
|---|---|---|---|---|---|
| 1 | 16x24 Red LED Matrix Panel - Chainable HT1632C Driver | | 0% | **$24.95** | **$24.95** |
| 8541.40.2000 | | | | | |
| China | | | | | |
| LEDs | | | | | |

**Sub-Total:** $24.95
**Sales Tax:** $0.00
**United States Postal Service (0.55lbs) (First-Class Mail International Parcel** incl. $1.60 insurance**):** $12.32
**Total:** **$37.27**

**Digi-Key CORPORATION**

701 Brooks Ave. South, P.O. Box 677, Thief River Falls, MN 56701-0677 USA

www.digikey.com
Orders 1-218-681-7979
Fax 1-218-681-3380

**Invoice #** 40319927
**U.S. $**

**Tracking # 490895293596**

40319927

Sold To:
CUSTOMER 6881150
MUSTAF ALQURAIN
SAUDI ARBIA
SAFWA CITY 31921
SAUDI ARABIA

Bill To:
MUSTAFA H ALQURAIN
SAUDI ARABIA
ALBAIRONI STREET
SAFWA CITY 31921
SAUDI ARABIA

| Terms | Invoice Date | Page |
|---|---|---|
| Mastercard | 10-DEC-2012 | 1 |
| Customer Purchase Order | | Sales Order 34799170 |
| Back Orders: Accepts to 9-JAN-2013 | | Account 2038959 |
| Entered By / Date: A0FX/10-DEC-2012 | Shipped Via: XF1D | Ship Date: 10-DEC-2012 |

*Easy to Remember:*
*1-800-DIGI-KEY*

| For Office Use Only | Received | VAT/Tax ID | Billing | Pack List No. | Printing Date | Currency Type | MSC # |
|---|---|---|---|---|---|---|---|
| | INTERNET | | BILL SHIP | 1 | 23-DEC-2012 | U.S. $ | 0 |

| Idx | Box | Ordered | Cancelled | Shipped | Item Number/Description | Back Order | Unit Price US $ | Amount US $ |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 2 | 0 | 2 | A102430-ND<br>RES 0.68 OHM 25W 5% WW LUG<br>HTSUS, 8533.29.0000 ECCN, EAR99<br>LEAD, LEAD FREE ROHS, ROHS COMP REACH, REACH UNAFFECTED<br>COUNTRY/ORIGIN, USA<br>CAGE, 0PJN9 | | 2.59000 | 5.18<br>AUG-2011 |
| 2 | 1 | 2 | 0 | 2 | 497-6038-5-ND<br>IC REG LDO 9V 2A TO-220<br>HTSUS, 8542.39.0000 ECCN, EAR99<br>LEAD, LEAD FREE ROHS, ROHS COMP REACH, REACH UNAFFECTED<br>COUNTRY/ORIGIN, MALAYSIA<br>CAGE, 50088 | | .97000 | 1.94<br>JUL-2011 |
| 3 | 1 | 12 | 0 | 12 | TIP2955GOS-ND<br>TRANS PNP 15A 60V TO218<br>HTSUS, 8541.29.0095 ECCN, EAR99<br>LEAD, LEAD FREE ROHS, ROHS COMP REACH, REACH UNAFFECTED<br>COUNTRY/ORIGIN, MALAYSIA<br>CAGE, 5V1P1 | | 1.61000 | 19.32<br>JUN-2012 |
| 4 | 1 | 3 | 0 | 3 | 497-8359-5-ND<br>IC REG LDO ADJ 3A TO220-3<br>HTSUS, 8542.39.0000 ECCN, EAR99<br>LEAD, LEAD FREE ROHS, ROHS COMP REACH, REACH UNAFFECTED<br>COUNTRY/ORIGIN, MALAYSIA<br>CAGE, 50088 | | 1.60000 | 4.80<br>JUL-2011 |
| 5 | 1 | 2 | 0 | 2 | 497-6038-5-ND<br>IC REG LDO 9V 2A TO-220<br>HTSUS, 8542.39.0000 ECCN, EAR99<br>LEAD, LEAD FREE ROHS, ROHS COMP REACH, REACH UNAFFECTED<br>COUNTRY/ORIGIN, MALAYSIA<br>CAGE, 50088 | | .97000 | 1.94<br>JUL-2011 |
| 6 | 1 | 2 | 0 | 2 | 497-6706-ND<br>IC REG LDO ADJ 5A TO-3<br>HTSUS, 8542.39.0000 ECCN, EAR99<br>LEAD, LEAD FREE ROHS, ROHS COMP REACH, REACH UNAFFECTED<br>COUNTRY/ORIGIN, MALAYSIA<br>CAGE, 50088 | | 13.04000 | 26.08<br>JUL-2011 |
| 7 | 1 | 6 | 0 | 6 | 50SQ080-ND<br>DIODE SCHOTTKY 80V 5A DO-204AR<br>HTSUS, 8541.10.0080 ECCN, EAR99<br>LEAD, LEAD FREE ROHS, ROHS COMP REACH, REACH UNAFFECTED<br>COUNTRY/ORIGIN, CHINA | | 1.95000 | 11.70<br>DEC-2011 |

Claims for pricing errors, shortages, and defective product must be reported within 30 days of invoice date.

*Contact Customer Service at 1-218-681-7979*

DUNS No: 05 760 2120 FEI No: 41-1234968 Any applicable sales tax not collected on this invoice is the responsibility of the customer.

**Digi-Key CORPORATION**

701 Brooks Ave. South, P.O. Box 677, Thief River Falls, MN 56701-0677 USA

www.digikey.com
Orders 1-218-681-7979
Fax 1-218-681-3380

**Invoice #** 40319927
**U.S. $**

**Tracking # 490895293596**

Sold To:
CUSTOMER 6881150
MUSTAF ALQURAIN
SAUDI ARBIA
SAFWA CITY 31921
SAUDI ARABIA

| Terms | Invoice Date | Page |
|---|---|---|
| Mastercard | 10-DEC-2012 | 2 |
| Customer Purchase Order | Shipped Via | |
| | XF1D | |

*Easy to Remember:*
*1-800-DIGI-KEY*

40319927

| Idx | Box | Ordered | Cancelled | Shipped | Item Number/Description | Back Order | Unit Price US $ | Amount US $ |
|---|---|---|---|---|---|---|---|---|
| 8 | 1 | 3 | 0 | 3 | A102133-ND | | 2.59000 | 7.77 |
| | | | | | RES 1.0 OHM 25W 5% WW LUG | | | |
| | | | | | HTSUS: 8533.29.0000 ECCN: EAR99 | | | |
| | | | | | LEAD: LEAD FREE ROHS: ROHS COMP REACH: REACH UNAFFECTED | | | AUG-2011 |
| | | | | | COUNTRY/ORIGIN: SWITZERLAND | | | |
| | | | | | CAGE: 0PJN9 | | | |
| 9 | 1 | 3 | 0 | 3 | A102412-ND | | 2.17000 | 6.51 |
| | | | | | RES 3.3 OHM 1GW 5% WW LUG | | | |
| | | | | | HTSUS: 8533.21.0080 ECCN: EAR99 | | | |
| | | | | | LEAD: LEAD FREE ROHS: ROHS COMP REACH: REACH UNAFFECTED | | | AUG-2011 |
| | | | | | COUNTRY/ORIGIN: SWITZERLAND | | | |
| | | | | | CAGE: 0PJN9 | | | |
| | | | | | BOX 1 SHIPPED KIPR WEIGHT 0 LBS 14 OZS | | | |
| | | | | | BOX ID 490895293596 | | | |
| | | | | | NO EEI 30.37(a) | | | |
| | | | | | NO LICENSE REQUIRED | | | |
| | | | | | TOTAL INVOICED | | | 85.24 |
| | | | | | SHIPPING CHARGES APPLIED | | | 120.00 |
| | | | | | ** CHARGES SUBTOTAL ** | | | 205.24 |
| | | | | | TOTAL CHARGED TO CREDIT CARD | | | 205.24 |
| | | | | | | | | U.S. $$ |
| | | | | | INCOTERM 2010: CPT SAFWA CITY, SAUDI ARABIA | | | |
| | | | | | YOUR CREDIT CARD HAS BEEN CHARGED THE ABOVE INDICATED AMOUNT | | | |
| | | | | | THE ORDER IS COMPLETE | | | |
| | | | | Ship To: | MUSTAFA ALQURAIN | | | |
| | | | | | SAUDI ARBIA | | | |
| | | | | | SAFWA CITY 31921 | | | |
| | | | | | SAUDI ARABIA | | | |
| | | | | Ship From: | DIGI-KEY CORPORATION | | | |
| | | | | | 701 BROOKS AVE. SOUTH | | | |
| | | | | | P.O. BOX 677 | | | |
| | | | | | THIEF RIVER FALLS MN 56701-0677 | | | |
| | | | | General - | WEB ORDER ID: 43634312 | | | |

These commodities, technology or software were exported from the United States in accordance with the Export Administration regulations. Diversion contrary to U.S. law prohibited.

**CERTIFICATE OF COMPLIANCE:** The Digi-Key components included in the above shipment are genuine components and were provided by the applicable manufacturer to Digi-Key. Test reports (chemical, physical, electrical, etc., together with results of any tests performed by the manufacturer) are on file (either here or in the plant of the manufacturer) and will be made available upon request. These components have been handled in accordance with the requirements of applicable quality standards. This certification is valid only to the original customer and is not transferable. Contact Customer Service at 800-858-3018 if you have any questions.

Kim Gilbert, Customer Service Manager

Scott Frisbie, Director Corporate Quality

Claims for pricing errors, shortages, and defective product must be reported within 30 days of invoice date.

*Contact Customer Service at 1-218-681-7979*

DUNS No. 06-360-9130 FEI No. 41-1294068 Any applicable sales tax not collected on this invoice is the responsibility of the customer

# 18Appendix E: Datasheets

1-Voltage regulator data sheet

## UTC LM78XX LINEAR INTEGRATED CIRCUIT

### ABSOLUTE MAXIMUM RATINGS

( Operating temperature range applies unless otherwise specified )

| PARAMETER | SYMBOL | RATING | | UNIT |
|---|---|---|---|---|
| Input voltage(for Vo=5~18V)<br>(for Vo=24V) | VI | | 35<br>40 | V<br>V |
| Output Current | I o | | 1 | A |
| Power Dissipation | PD | Internally | Limited | W |
| Operating Junction Temperature Range | TOPR | -20 | +150 | ℃ |
| Storage Temperature Range | TSTG | -55 | +150 | ℃ |

### UTC LM7805 ELECTRICAL CHARACTERISTICS

( VI=10V, Io=0.5A, Tj= 0°C - 125°C, C1=0.33uF, Co=0.1uF, unless otherwise specified )(Note 1)

| PARAMETER | SYMBOL | TEST CONDITIONS | MIN | TYP | MAX | UNIT |
|---|---|---|---|---|---|---|
| Output Voltage | Vo | Tj=25°C, Io=5mA - 1.0A | 4.80 | 5.0 | 5.20 | V |
| | | VI =7.5V to 20V,<br>Io=5mA - 1.0A,PD<15W | 4.75 | | 5.25 | V |
| Load Regulation | ΔVo | Tj=25℃,Io=5mA - 1.5A | | | 50 | mV |
| | | Tj=25℃,Io=0.25A - 0.75A | | | 25 | mV |
| Line regulation | ΔVo | VI =7V to 25V,Tj=25°C | | | 50 | mV |
| | | VI =7.5V to 20V,Tj=25℃,Io=1A | | | 50 | mV |
| Quiescent Current | Iq | Tj=25℃, Io=<1A | | | 8.0 | mA |
| Quiescent Current Change | ΔIq | VI =7.5V to 20V | | | 1.0 | mA |
| | ΔIq | Io=5mA - 1.0A | | | 0.5 | mA |
| Output Noise Voltage | VN | 10Hz<=f<=100kHz | | 40 | | μV |
| Temperature coefficient of Vo | ΔVo/ΔT | Io=5mA | | -0.6 | | mV/°C |
| Ripple Rejection | RR | VI =8V - 18V,f=120Hz,Tj=25°C | 62 | 80 | | dB |
| Peak Output Current | IPK | Tj=25°C | | 1.8 | | A |
| Short-Circuit Current | ISC | VI=35V, Tj=25°C | | 250 | | mA |
| Dropout Voltage | Vd | Tj=25°C | | 2.0 | | V |

### UTC LM7806 ELECTRICAL CHARACTERISTICS

( VI=11V, Io=0.5A, Tj= 0°C - 125°C, C1=0.33uF, Co=0.1uF, unless otherwise specified )(Note 1)

| PARAMETER | SYMBOL | TEST CONDITIONS | MIN | TYP | MAX | UNIT |
|---|---|---|---|---|---|---|
| Output Voltage | Vo | Tj=25°C, Io=5mA - 1.0A | 5.76 | 6.0 | 6.24 | V |
| | | VI =8.5V to 21V,<br>Io=5mA - 1.0A, PD<15W | 5.70 | | 6.30 | V |
| Load Regulation | ΔVo | Tj=25℃,Io=5mA - 1.5A | | | 60 | mV |
| | | Tj=25℃,Io=0.25A - 0.75A | | | 30 | mV |
| Line regulation | ΔVo | VI =8V to 25V,Tj=25°C | | | 60 | mV |
| | | VI =8.5V to 21V,Tj=25℃,Io=1A | | | 60 | mV |
| Quiescent Current | Iq | Tj=25℃, Io=<1A | | | 8.0 | mA |
| Quiescent Current Change | ΔIq | VI =8.5V to 21V | | | 1.0 | mA |
| | ΔIq | Io=5mA - 1.0A | | | 0.5 | mA |
| Output Noise Voltage | VN | 10Hz<=f<=100kHz | | 45 | | μV |
| Temperature coefficient of Vo | ΔVo/ΔT | Io=5mA | | -0.7 | | mV/°C |
| Ripple Rejection | RR | VI =9V - 19V,f=120Hz,Tj=25°C | 59 | 75 | | dB |

# L78S00 SERIES

## 2A POSITIVE VOLTAGE REGULATORS

- OUTPUT CURRENT TO 2A
- OUTPUT VOLTAGES OF 5; 7.5; 9; 10; 12; 15; 18; 24V
- THERMAL OVERLOAD PROTECTION
- SHORT CIRCUIT PROTECTION
- OUTPUT TRANSITION SOA PROTECTION

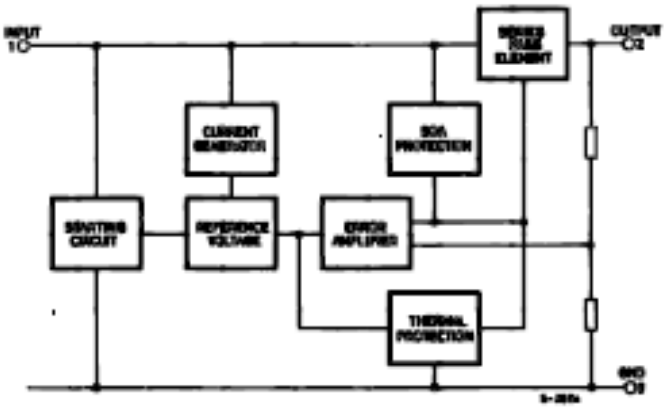
TO-220

TO-3

### DESCRIPTION

The L78S00 series of three-terminal positive regulators is available in TO-220 and TO-3 packages and with several fixed output voltages, making it useful in a wide range of applications. These regulators can provide local on-card regulation, eliminating the distribution problems associated with single point regulation. Each type employs internal current limiting, thermal shut-down and safe area protection, making it essentially indestructible. If adequate heat sinking is provided, they can deliver over 2A output current. Although designed primarily as fixed voltage regulators, these devices can be used with external components to obtain adjustable voltages and currents.

### SCHEMATIC DIAGRAM



## ABSOLUTE MAXIMUM RATINGS

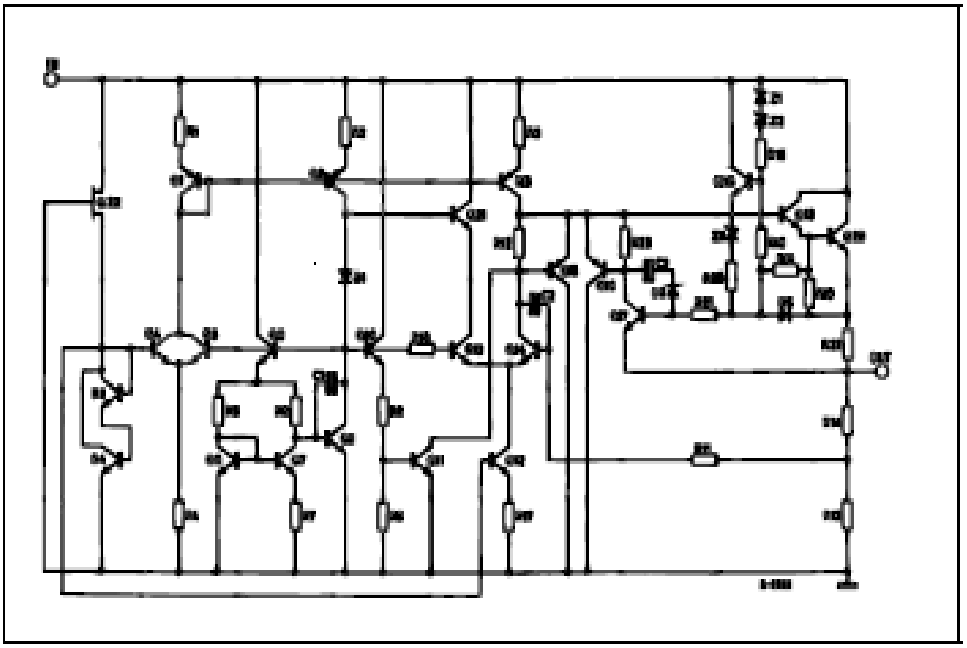
| Symbol | Parameter[2] | | Value | Unit |
|---|---|---|---|---|
| $V_I$ | DC Input Voltage | for $V_O$ = 5 to 18 V | 35 | V |
| | | for $V_O$ = 24 V | 40 | |
| $I_O$ | Output Current | | Internally Limited | |
| $P_{tot}$ | Power Dissipation | | Internally Limited | |
| $T_{stg}$ | Storage Temperature Range | | -65 to 150 | °C |
| $T_{op}$ | Operating Junction Temperature Range | for L78S00 | -55 to 150 | °C |
| | | for L78S00C | 0 to 150 | |

Absolute Maximum Ratings are those values beyond which damage to the device may occur. Functional operation under these condition is not implied.

## THERMAL DATA

| Symbol | Parameter | | TO-220 | TO-3 | Unit |
|---|---|---|---|---|---|
| $R_{thj\text{-}case}$ | Thermal Resistance Junction-case | Max | 5 | 4 | °C/W |
| $R_{thj\text{-}amb}$ | Thermal Resistance Junction-ambient | Max | 50 | 35 | °C/W |

## SHEMATIC DIAGRAM



Bulletin PD-2.060 rev. G 06/05

International
IOR Rectifier

# 50SQ... SERIES

## SCHOTTKY RECTIFIER 5 Amp

### Major Ratings and Characteristics

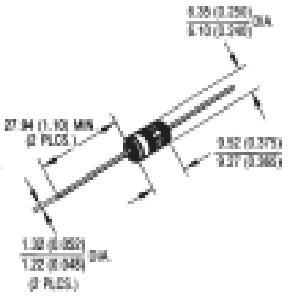
| Characteristics | | 50SQ... | Units |
|---|---|---|---|
| $I_{F(AV)}$ | Rectangular waveform | 5 | A |
| $V_{RRM}$ | range | 60 / 100 | V |
| $I_{FSM}$ | @tp = 5 µs sine | 1900 | A |
| $V_F$ | @5Apk, $T_J$ = 125°C | 0.52 | V |
| $T_J$ | range | -55 to 175 | °C |

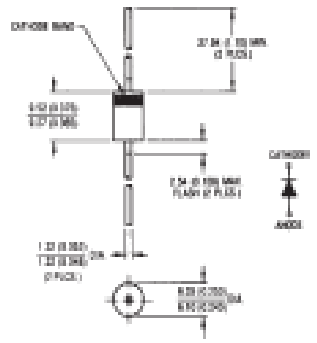
### Description/ Features

The 50SQ... axial leaded Schottky rectifier series has been optimized for low reverse leakage at high temperature. The proprietary barrier technology allows for reliable operation up to 175°C junction temperature. Typical applications are in switching power supplies, converters, free-wheeling diodes, and reverse battery protection.

- 175° C $T_J$ operation
- High purity, high temperature epoxy encapsulation for enhanced mechanical strength and moisture resistance
- Low forward voltage drop
- High frequency operation
- Guard ring for enhanced ruggedness and long term reliability
- Lead-Free plating

### CASE STYLE AND DIMENSIONS





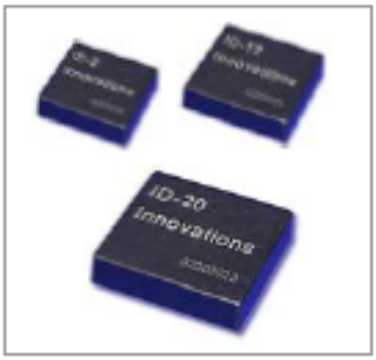
Conforms to JEDEC Outline DO - 204AR

Dimensions in millimeters and inches

# ID SERIES DATASHEET MAR 01, 2005

## ID-2/ID-12 Brief Data

The ID2. ID12 and ID20 are similar to the obsolete ID0, ID10 and ID15 MK(ii) series devices, but they have extra pins that allow Magnetic Emulation output to be included in the functionality. The ID-12 and ID-20 come with internal antennas, and have read ranges of 12+ cm and 16+ cm, respectively. With an external antenna, the ID-2 can deliver read ranges of up to 25 cm. All three readers support ASCII, Wiegand26 and Magnetic ABA Track2 data formats.




ID2 / ID12 / ID20 PIN-OUT


1. GND
2. RES (Reset Bar)
3. ANT (Antenna)
4. ANT (Antenna)
5. CP
6. Future
7. +/- (Format Selector)
8. D1 (Data Pin 1)
9. D0 (Data Pin 0)
10. LED (LED / Beeper)
11. +5V

### *Operational and Physical Characteristics*

| Parameters | ID-2 | ID-12 | ID-20 |
|---|---|---|---|
| Read Range | N/A (no internal antenna) | 12+ cm | 16+ cm |
| Dimensions | 21 mm x 19 mm x 6 mm | 26 mm x 25 mm x 7 mm | 40 mm x 40 mm x 9 mm |
| Frequency | 125 kHz | 125 kHz | 125 kHz |
| Card Format | EM 4001 or compatible | EM 4001 or compatible | EM 4001 or compatible |
| Encoding | Manchester 64-bit, modulus 64 | Manchester 64-bit, modulus 64 | Manchester 64-bit, modulus 64 |
| Power Requirement | 5 VDC @ 13mA nominal | 5 VDC @ 30mA nominal | 5 VDC @ 65mA nominal |
| I/O Output Current | +/-200mA PK | - | - |
| Voltage Supply Range | +4.6V through +5.4V | +4.6V through +5.4V | +4.6V through +5.4V |

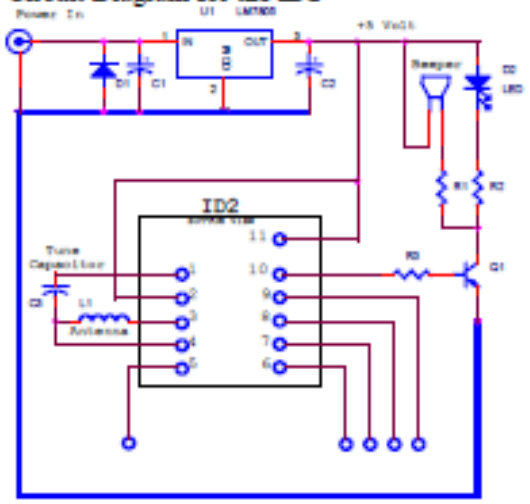
### *Pin Description & Output Data Formats*

| Pin No. | Description | ASCII | Magnet Emulation | Wiegand26 |
|---|---|---|---|---|
| Pin 1 | Zero Volts and Tuning Capacitor Ground | GND 0V | GND 0V | GND 0V |
| Pin 2 | Strap to +5V | Reset Bar | Reset Bar | Reset Bar |
| Pin 3 | To External Antenna and Tuning Capacitor | Antenna | Antenna | Antenna |
| Pin 4 | To External Antenna | Antenna | Antenna | Antenna |
| Pin 5 | Card Present | No function | Card Present * | No function |

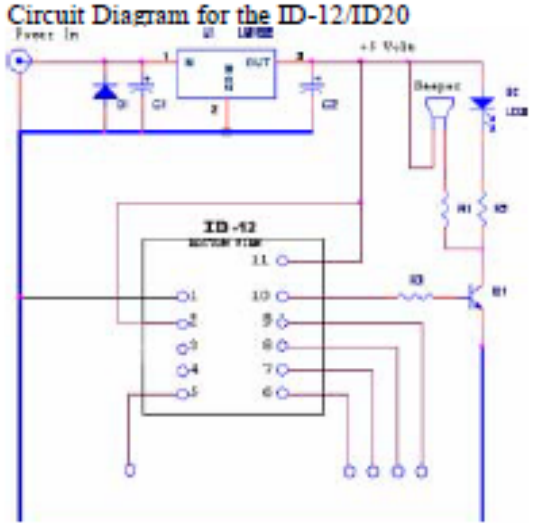
| | | | | |
|---|---|---|---|---|
| Pin 6 | Future | Future | Future | Future |
| Pin 7 | Format Selector (+/-) | Strap to GND | Strap to Pin 10 | Strap to +5V |
| Pin 8 | Data 1 | CMOS | Clock * | One Output * |
| Pin 9 | Data 0 | TTL Data (inverted) | Data * | Zero Output * |
| Pin 10 | 3.1 kHz Logic | Beeper / LED | Beeper / LED | Beeper / LED |
| Pin 11 | DC Voltage Supply | +5V | +5V | +5V |

* Requires 4K7 Pull-up resistor to +5V

## Circuit Diagram for the ID2




## Circuit Diagram for the ID-12/ID20



SHARP GP2Y0A21YK0F

# GP2Y0A21YK0F

## Distance Measuring Sensor Unit
## Measuring distance: 10 to 80 cm
## Analog output type

### ■Description

GP2Y0A21YK0F is a distance measuring sensor unit, composed of an integrated combination of PSD (position sensitive detector) , IRED (infrared emitting diode) and signal processing circuit.
The variety of the reflectivity of the object, the environmental temperature and the operating duration are not influenced easily to the distance detection because of adopting the triangulation method.
This device outputs the voltage corresponding to the detection distance. So this sensor can also be used as a proximity sensor.

### ■Features

1. Distance measuring range : 10 to 80 cm
2. Analog output type
3. Package size : 29.5×13×13.5 mm
4. Consumption current : Typ. 30 mA
5. Supply voltage : 4.5 to 5.5 V

### ■Agency approvals/Compliance

1. Compliant with RoHS directive (2002/95/EC)

### ■Applications

1. Touch-less switch
   (Sanitary equipment, Control of illumination, etc. )
2. Robot cleaner
3. Sensor for energy saving
   (ATM, Copier, Vending machine)
4. Amusement equipment
   (Robot, Arcade game machine)





Sheet No.: E6-A00201EN
Date Dec.01.2006

5-IR sensor



## ■Absolute Maximum Ratings (Ta=25℃,Vcc=5V)

| Parameter | Symbol | Rating | Unit |
|---|---|---|---|
| Supply voltage | $V_{CC}$ | -0.3 to +7 | V |
| Output terminal voltage | $V_O$ | -0.3 to $V_{CC}$+0.3 | V |
| Operating temperature | $T_{opr}$ | -10 to +60 | ℃ |
| Storage temperature | $T_{stg}$ | -40 to +70 | ℃ |

## ■Electro-optical Characteristics (Ta=25℃,Vcc=5V)

| Parameter | Symbol | Conditions | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|
| Average supply current | $I_{CC}$ | L=80cm (Note 1) | — | 30 | 40 | mA |
| Distance measuring | ΔL | (Note 1) | 10 | — | 80 | cm |
| Output voltage | $V_O$ | L=80cm (Note 1) | 0.25 | 0.4 | 0.55 | V |
| Output voltage differential | $\Delta V_O$ | Output voltage differece between L=10cm and L=80cm (Note 1) | 1.65 | 1.9 | 2.15 | V |

* L : Distance to reflective object

Note 1 : Using reflective object : White paper (Made by Kodak Co., Ltd. gray cards R-27・white face, reflectance; 90%)

## ■Recommended operating conditions

| Parameter | Symbol | Rating | Unit |
|---|---|---|---|
| Supply voltage | $V_{CC}$ | 4.5 to 5.5 | V |

# 1 OVERVIEW

## 1.1 OBJECT OF THE DOCUMENT

This document gives an overview of the SM5210 module: a miniature, single-side board, quad-band GSM 850/EGSM 900/DCS 1800/PCS 1900 module, ready for integration in various kinds of Fix wireless phones and other wireless devices.

## 1.2 SYSTEM BLOCK DIAGRAM

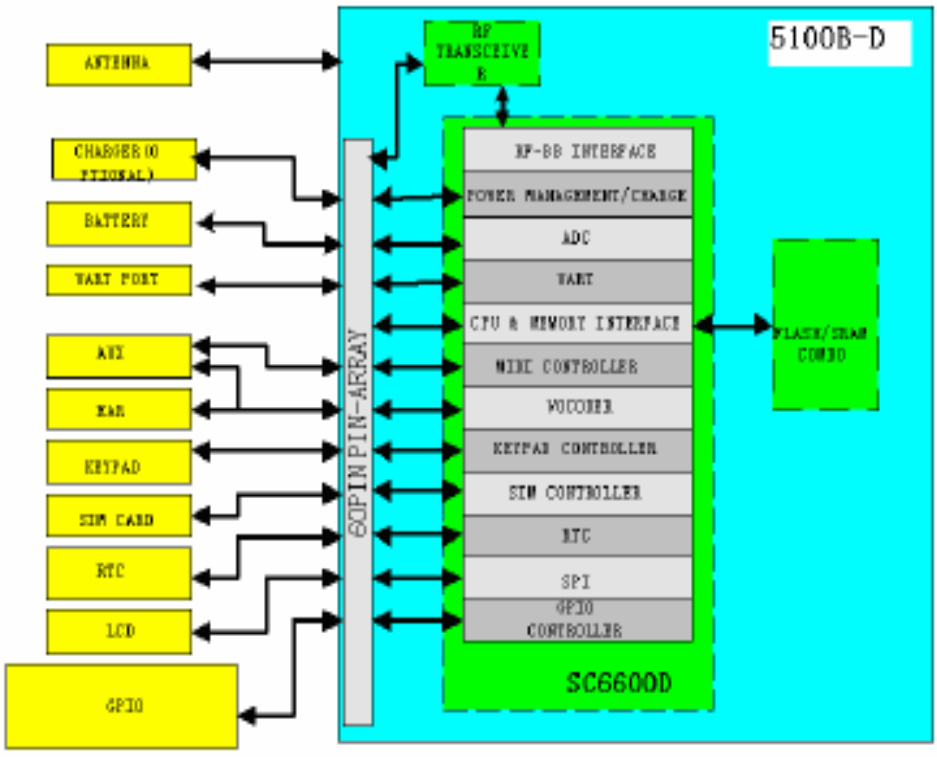



Figure 1: System block diagram

## 1.3 PRODUCT OVERVIEW

| | |
|---|---|
| Temperature range | Normal range: -10°C to +55° C (full compliant)<br>Storage: -40°C to +85°C |
| Weight | < 9g |
| Physical dimensions | 35.0X39.0X2.9 mm (typical) |

| | |
|---|---|
| Connection | 60 pins |
| Power supply | VBAT: 3.3V to 4.2V range, 3.6V typical. |
| Power consumption | Off mode: <100uA<br>Sleep mode: <2.0mA<br>Idle mode: <7.0mA (average)<br>Communication mode: 350 mA (average,GSM)<br>Communication mode: 2000mA (Typical peak during TX slot,GSM) |
| LHon Battery charging management and interface (OPTION) | Li-ion Battery charging management is included. The charger interface is provided on 60-pin connector. (only for 3.7V Li-ion Battery) |
| Frequency bands | EGSM900 +GSM850+ DCS1800+PCS1900 |
| Transmit power | Class 4 (2W) for EGSM900/GSM850<br>Class 1 (1W) for DCS1800/PCS1900 |
| Supported SIM card | 3V/1.8V SIM card. (auto recognise) |
| Keyboard interface | 4x6 keyboard interface is provided |
| UART0 interface with flow control | Up to 460 kbps<br>Full hardware flow control signals (+3.0V) are provided on 60 pins. |
| UART1 interface without flow control | 2-Wire UART interface<br>Up to 460 kbps |
| LCD interface | Support standard SPI interface, |

## 1.4 FUNCTIONAL DESCRIPTION

### 1.4.1 RF Functionalities

The RF part of this module converts RF signals to baseband for receiver chain and translates base band signals into RF frequency spectrum.
The operating frequencies are:

Rx (EGSM 850): 869 to 894MHz
Tx (EGSM 850): 824 to 849MHz

Rx (EGSM 900): 925 to 960MHz
Tx (EGSM 900): 880 to 915MHz

Rx (DCS 1800): 1805 to 1880MHz
Tx (DCS 1800): 1710 to 1785MHz

Rx (PCS 1900): 1930 to 1990MHz
Tx (PCS 1900): 1850 to 1910MHz

### 1.4.2 Baseband Functionalities

The baseband part of SM5210 is composed of a SPREADTRUM's SC6600D chip. This chipset is using 0.18μm mixed signal CMOS technology which allows massive integration as well as low power consumption. SC6600D provides single-chip solution to wireless Quad-band telephone handsets and data modems confirming to the EGSM 900, GSM 850, DCS 1800 and PCS 1900.

8-Servo motor specification

## Specifications

| | |
|---|---|
| Motor Type: | 3 Pole |
| Bearing Type: | Top Ball Bearing |
| Speed (4.8V/6.0V): | 0.20 / 0.17 sec @ 60 deg. |
| Torque oz./in. (4.8V/6.0V): | 72 / 89 |
| Torque kg./cm. (4.8V/6.0V): | 5.2 / 6.4 |
| Size in Inches: | 1.57 x 0.78 x 1.49 |
| Size in Millimeters: | 39.88 x 19.81 x 37.85 |
| Weight ounces: | 1.59 |
| Weight grams: | 45.08 |

8-Arduino MEGA 2560 Technical Specification

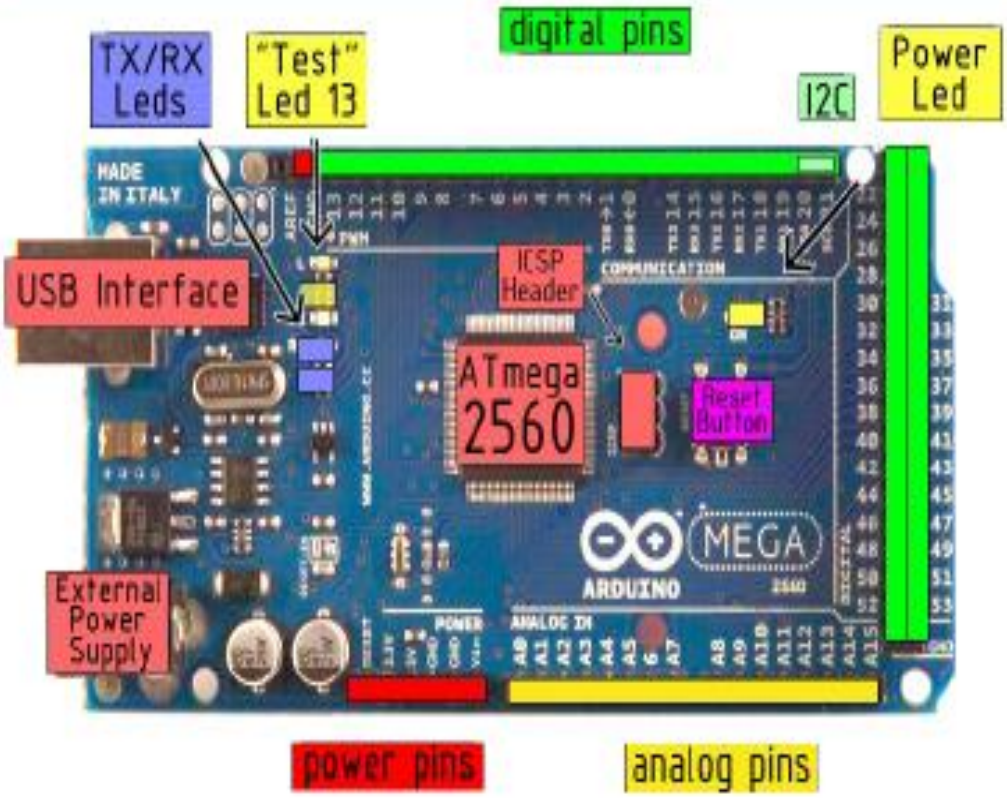
# Technical Specification

EAGLE files: arduino-mega2560-reference-design.zip Schematic: arduino-mega2560-schematic.pdf

## Summary

| | |
|---|---|
| Microcontroller | ATmega2560 |
| Operating Voltage | 5V |
| Input Voltage (recommended) | 7-12V |
| Input Voltage (limits) | 6-20V |
| Digital I/O Pins | 54 (of which 14 provide PWM output) |
| Analog Input Pins | 16 |
| DC Current per I/O Pin | 40 mA |
| DC Current for 3.3V Pin | 50 mA |
| Flash Memory | 256 KB of which 8 KB used by bootloader |
| SRAM | 8 KB |
| EEPROM | 4 KB |
| Clock Speed | 16 MHz |

## the board



# Technical Specification

EAGLE files: arduino-duemilanove-uno-design.zip Schematic: arduino-uno-schematic.pdf

## Summary

| | |
|---|---|
| Microcontroller | ATmega328 |
| Operating Voltage | 5V |
| Input Voltage (recommended) | 7-12V |
| Input Voltage (limits) | 6-20V |
| Digital I/O Pins | 14 (of which 6 provide PWM output) |
| Analog Input Pins | 6 |
| DC Current per I/O Pin | 40 mA |
| DC Current for 3.3V Pin | 50 mA |
| Flash Memory | 32 KB of which 0.5 KB used by bootloader |
| SRAM | 2 KB |
| EEPROM | 1 KB |
| Clock Speed | 16 MHz |

## the board