RFID BASED SMART TROLLEY FOR AUTONATIC BILLING SYSTEM

A Major Project Report Submitted
In partial fulfillment of the requirements for the award of the degree of

Bachelor of Technology
in
Electronics and Communication Engineering

by

K.YESHWANTH REDDY	20N31A04B2
K.RAKESH	20N31A0495
H.SUMEDH	20N31A0485

Under the esteemed guidance of

Mrs.REVATHI
Assistant Professor

Department of Electronics and Communication Engineering

Malla Reddy College of Engineering & Technology

(Autonomous Institution- UGC, Govt. of India)
(Affiliated to JNTUH, Hyderabad, Approved by AICTE, NBA &NAAC with 'A' Grade)
Maisammaguda, Kompally, Dhulapally, Secunderabad – 500100
website: www.mrcet.ac.in

2020-2024

Malla Reddy College of Engineering & Technology

(Autonomous Institution- UGC, Govt. of India)

(Affiliated to JNTUH, Hyderabad, Approved by AICTE, NBA &NAAC with 'A' Grade)
Maisammaguda, Kompally, Dhulapally, Secunderabad – 500100
website: www.mrcet.ac.in

CERTIFICATE

This is to certify that this is the bonafide record of the major project entitled **"RFID BASED SMART TROLLEY FOR AUTOMATIC BILLINF SYSTEM"**, submitted by **K.YESHWANTH REDDY (20N31A04B2), K.RAKESH (20N31A0495)** and **K.SUMEDH (20N31A0485)** of B.Tech in the partial fulfillment of the requirements for the degree of Bachelor of Technology in Electronics and Communication Engineering , Department of Electronics and Communication Engineering during the year 2023-2024. The results embodied in this major project report have not been submitted to any other university or institute for the award of any degree or diploma.

Internal Guide
Mrs.REVATHI
Assistant Profeesor

Head of the Department
Dr.K.Mallikarjuna Lingam
Professor & HOD

External Examiner

DECLARATION

We hereby declare that the major project titled “RFID BASED SMART TROLLEY FOR AUTOMATIC BILLING SYSTEM” submitted to Malla Reddy College of Engineering and Technology (UGC Autonomous), affiliated to Jawaharlal Nehru Technological University Hyderabad (JNTUH) for the award of the degree of Bachelor of Technology in, Electronics and Communication Engineering is a result of original research carried-out in this thesis. It is further declared that the mini project report or any part thereof has not been previously submitted to any University or Institute for the award of degree or diploma.

K.YESHWANTH REDDY - 20N31A04B2

H.SUMEDH 20N31A0485

K.RAKESH - 20N31A0495

ACKNOWLEDGMENTS

We extend our gratitude to **Dr. VSK. Reddy, Director, Malla reddy college of engineering and technology, Hyderabad**, for facilitating us to present the Mini Project.

We wholeheartedly express our deep sense of gratitude to our beloved **Principal**, **Dr S. Srinivasa Rao**, for providing us adequate facilities for our successful completion of project work.

We extend our gratitude to **Prof P. Sanjeeva Reddy, Dean International Studies, Malla reddy college of engineering and technology, Hyderabad**, for facilitating us to present the Mini Project.

We would like to thank **Dr. K.Mallikarjuna Lingam, Head of the ECE Department**, for providing the freedom to use all the facilities available in the department, for successful completion of project.

We are grateful to our **Project Coordinator, Mrs.Vaidehi ,** for extending his support and cooperation throughout our project work.

We are very much thankful to our project guide **Mrs.Revathi,Assistant Professor**,
For her extensive patience and guidance throughout our project work.

We would also like to thank all my **friends** for providing help and moral support at the right time. With great affection and love, we thank our **parents** who were the backbone behind my deeds

By

HAZARI SUMEDH (20N31A0485)
KETHIREDDY YASHWANTH(20N31A04B2)
KOMATI RAKESH (20N31A0495)

.

ABSTRACT

The modern technology has increased the standard of living for the humans. This resulted in large crowds at shopping malls. To handle the large crowd, we must reduce the process of the billing time. This is done using smart shopping system based on RFID. Items that are put in a smart shopping cart are read one by one and the bill is generated and displayed. After the final bill is generated the customer pays the bill by using their Pre charged cards provided by the shopping mall. The aim is to reduce the time consumption needed for the billing system

CHAPTER 1

INTRODUCTION:

Contemporary embedded systems are habitually based on microcontroller's i.e. CPUs in the company of integrated memory as well as peripheral interfaces but ordinary microprocessors by means of external chips for memory and peripheral interface circuits are also still common, especially in more complex systems. Radio frequency identification (RFID) technology may not only be useful for streamlining inventory and supply chains: it could also make shoppers swarm. ZigBee is based on an IEEE 802.15 standard. ZigBee devices often transmit data over longer distances by passing data through intermediate devices to reach more distant ones, creating a mesh network; i.e., a network with no centralized control or high-power transmitter/receiver able to reach all of the networked devices. This paper provides centralized and automated billing system using RFID and ZigBee communication. Each product of canteen will be provided with a RFID tag, to identify its type. Each cart is designed or implemented with a Product Identification Device that contains microcontroller, LCD, an RFID reader, EEPROM, and ZigBee module. Purchasing product information will be read through a RFID reader on cart, mean while product information will be stored into EEPROM attached to it and EEPROM data will be send to Central Billing System through ZigBee module. The central billing system gets the cart information and EEPROM data, it access the product database and calculates the total amount of purchasing for that particular cart. Main aim of this paper was to provide an automatic billing to avoid queue in canteen , malls and super markets.

LITERATURE REVIEW:

Human beings have always technologically advanced to support their needs ever since the beginning of mankind. The main purpose of innovation in technology, irrespective of the domain, has been in simplifying tasks and making everyday jobs easier and faster. During the last decade, the commercial use of RFID has been growing rapidly all over the world. Furthermore, it is projected that the RFID market will reach an estimated US$18.7 billion by the year 2017 (GIAI, 2012). Everywhere retailers are increasingly embedding RFID technology into their supermarket products in order to improve the customers' shopping experience, customer support and develop new services for customers [1]. RFID is a technology that uses radio waves to track, capture, identify and transfer data efficiently and without human intervention. RFID- based system gathers data about a certain object without touching it or seeing it stag and forwards the information to a host computer [2]. The data on the tag include pointer to the central database within an RFID system. RFID-Readers are able to establish a Channel of communication, read the tags and trace the movement of these objects within the coverage area. RFID is a promising technology, which can improve operational efficiency specially a considerable amount of reduction in transaction costs. Tag detection does not require human intervention therefore reduces employment costs and eliminates human errors during data collection. Due to its flexibility and business efficiency, the RFID technology has been widely adopted in a wide range of applications such as supply chain management and inventory, libraries, equipment and parts maintenance, vehicle identification, tracking people, access control, reliable car tracking, manufacturing line control, automated reading and receipt of goods at end sale points, e-passport and much more (Owunwanne and Goel, 2010). Implementation of RFID based applications has become an objective of many Organizations, partially as a result of the decision made by Wal-Mart, the world's biggest retailer, to implement the RFID technology to monitor flow of pallets and packaging in its supply chain and ask their top 100 vendors deploy RFID. The US Department of Defence, Proctor and Gamble and the European retailer Metro Group require their larger suppliers to implement RFID on every box and pallet shipped to them. The grocery industry is a prime candidate for RFID implementation. On-hand shelf inventory system in the supermarket will be linked to the store's information system, thus maintaining real time product information and automatic inventory tracking to keeping the correct inventory levels [3]. Using RFID technology in supermarkets can also provide detailed information on customer purchase behaviour. Currently most of the supermarkets use a barcode-based system whereby an item is assigned a serial number printed on the barcode label attached to an item and the item related information is stored in the database of the back-end system. To perform inventory control, someone has to scan the barcode label of each item and compare them with existing inventory list. This is a lengthy and error prone process; as a result, it's done less frequently and hence often is not up-to-date. The capabilities of the barcode technology are limited in term of functionalities that businesses require (Bendavid et al., 2006). RFID technology offers a solution to the above-mentioned problem.

3. INTRODUCTION ABOUT EMBEDDED SYSTEMS

3.1 INTRODUCTION OF EMBEDDED SYSTEM:

An embedded system is a combination of software and hardware to perform a dedicated task. Some of the main devices used in embedded products are Microprocessors and Microcontrollers.

Microprocessors are commonly referred to as general purpose processors as they simply accept the inputs, process it and give the output. In contrast, a microcontroller not only accepts the data as inputs but also manipulates it, interfaces the data with various devices, controls the data and thus finally gives the result.

An Embedded System is a combination of computer hardware and software, and perhaps additional mechanical or other parts, designed to perform a specific function. A good example is the microwave oven. Almost every household has one, and tens of millions of them are used every day, but very few people realize that a processor and software are involved in the preparation of their lunch or dinner.

This is in direct contrast to the personal computer in the family room. It too is comprised of computer hardware and software and mechanical components (disk drives, for example). However, a personal computer is not designed to perform a specific function rather; it is able to do many different things. Many people use the term general-purpose computer to make this distinction clear. As shipped, a general-purpose computer is a blank slate; the manufacturer does not know what the customer will do wish it. One customer may use it for a network file server another may use it exclusively for playing games, and a third may use it to write the next great American novel.

Frequently, an embedded system is a component within some larger system. For example, modern cars and trucks contain many embedded systems. One embedded system controls the anti-lock brakes, other monitors and controls the vehicle's emissions, and a third displays information on the dashboard. In some cases, these embedded systems are connected by some sort of a communication network, but that is certainly not a requirement.

At the possible risk of confusing you, it is important to point out that a general-purpose computer is itself made up of numerous embedded systems. For example, my computer consists of a keyboard, mouse, video card, modem, hard drive, floppy drive, and sound card-each of which is an embedded system. Each of these devices contains a processor and software and is designed to perform a specific function. For example, the modem is designed to send and receive digital data over analog telephone line. That's it and all of the other devices can be summarized in a single sentence as well.

2. OVERVIEW OF EMBEDDED SYSTEM :

Every embedded system consists of custom-built hardware built around a Central Processing Unit (CPU). This hardware also contains memory chips onto which the software is loaded. The software residing on the memory chip is also called the ‘firmware’.

The same architecture is applicable to any computer including a desktop computer. However, there are significant differences. It is not compulsory to have an operating system in every embedded system. For small appliances such as remote control units, air conditioners, toys etc., there is no need foran operating system and you can write only the software specific to that application.

For applications involving complex processing, it is advisable to have an operating system. In such a case, you need to integrate the application software with the operating system and then transfer the entire software on to the memory chip. Once the software is transferred to the memory chip, the software will continue to run *for* a long time you don’t need to reload new software. Now, let us see the details of the various building blocks of the hardware of an embedded system. As shown in Fig. the building blocks are:

a. Central Processing Unit (CPU)
b. Memory (Read-only Memory and Random Access Memory)
c. Input Devices
d. Output devices
e. Communication interfaces
f. Application-specific circuitry

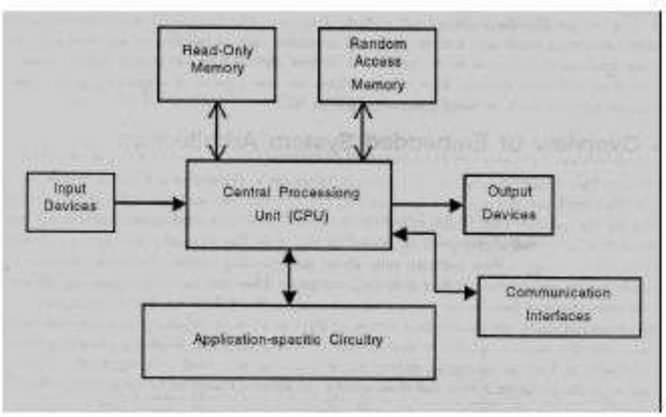


Fig: 3.1 Building blocks of the hardware of an embedded system

CENTRAL PROCESSING UNIT (CPU):

The Central Processing Unit (processor, in short) can be any of the following: microcontroller, microprocessor or Digital Signal Processor (DSP). A micro-controller is a low-cost processor. Its main attraction is that on the chip itself, there will be many other components such as memory, serial communication interface, analog-to digital converter etc.

So, for small applications, a micro-controller is the best choice as the number of external components required will be very less. On the other hand, microprocessors are more powerful, but you need to use many external components with them. D5P is used mainly for applications in which signal processing is involved such as audio and video processing.

➢ **MEMORY:**

The memory is categorized as Random Access 11emory (RAM) and Read Only Memory (ROM). The contents of the RAM will be erased if power is switched off to the chip, whereas ROM retains the contents even if the power is switched off. So, the firmware is stored in the ROM. When power is switched on, the processor reads the ROM; the program is program is executed.

- **INPUT DEVICES**:

  Unlike the desktops, the input devices to an embedded system have very limited capability. There will be no keyboard or a mouse, and hence interacting with the embedded system is no easy task. Many embedded systems will have a small keypad-you press one key to give a specific command. A keypad may be used to input only the digits. Many embedded systems used in process control do not have any input device for user interaction; they take inputs from sensors or transducers 1'fnd produce electrical signals that are in turn fed to other systems.

- **OUTPUT DEVICES**:

  The output devices of the embedded systems also have very limited capability. Some embedded systems will have a few Light Emitting Diodes (LEDs) to indicate the health status of the system modules, or for visual indication of alarms. A small Liquid Crystal Display (LCD) may also be used to display some important parameters.

- **COMMUNICATION INTERFACES**:

  The embedded systems may need to, interact with other embedded systems at they may have to transmit data to a desktop. To facilitate this, the embedded systems are provided with one or a few communication interfaces such as RS232, RS422, RS485, Universal Serial Bus (USB), IEEE 1394, Ethernet etc.

- **APPLICATION-SPECIFIC CIRCUITRY:**

  Sensors, transducers, special processing and control circuitry may be required fat an embedded system, depending on its application. This circuitry interacts with the processor to carry out the necessary work. The entire hardware has to be given power supply either through the 230 volts main supply or through a battery. The hardware has to design in such a way that the power consumption is minimized.

4. DESIGN OF HARDWARE

Arduino Uno

The most common version of Arduino is the Arduino Uno. This board is what most people are talking about when they refer to an Arduino. The Uno is one of the more popular boards in the Arduino family and a great choice for beginners. There are different revisions of Arduino Uno, below detail is the most recent revision (Rev3 or R3).

The Arduino Uno is a microcontroller board based on the ATmega328. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started.

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 (ATmega328) of which 0.5 KB used by
bootloader		
SRAM	:	2 KB (ATmega328)
EEPROM	:	1 KB (ATmega328)
Clock Speed	:	16 MHz
Length	:	68.6 mm
Width	:	53.4 mm

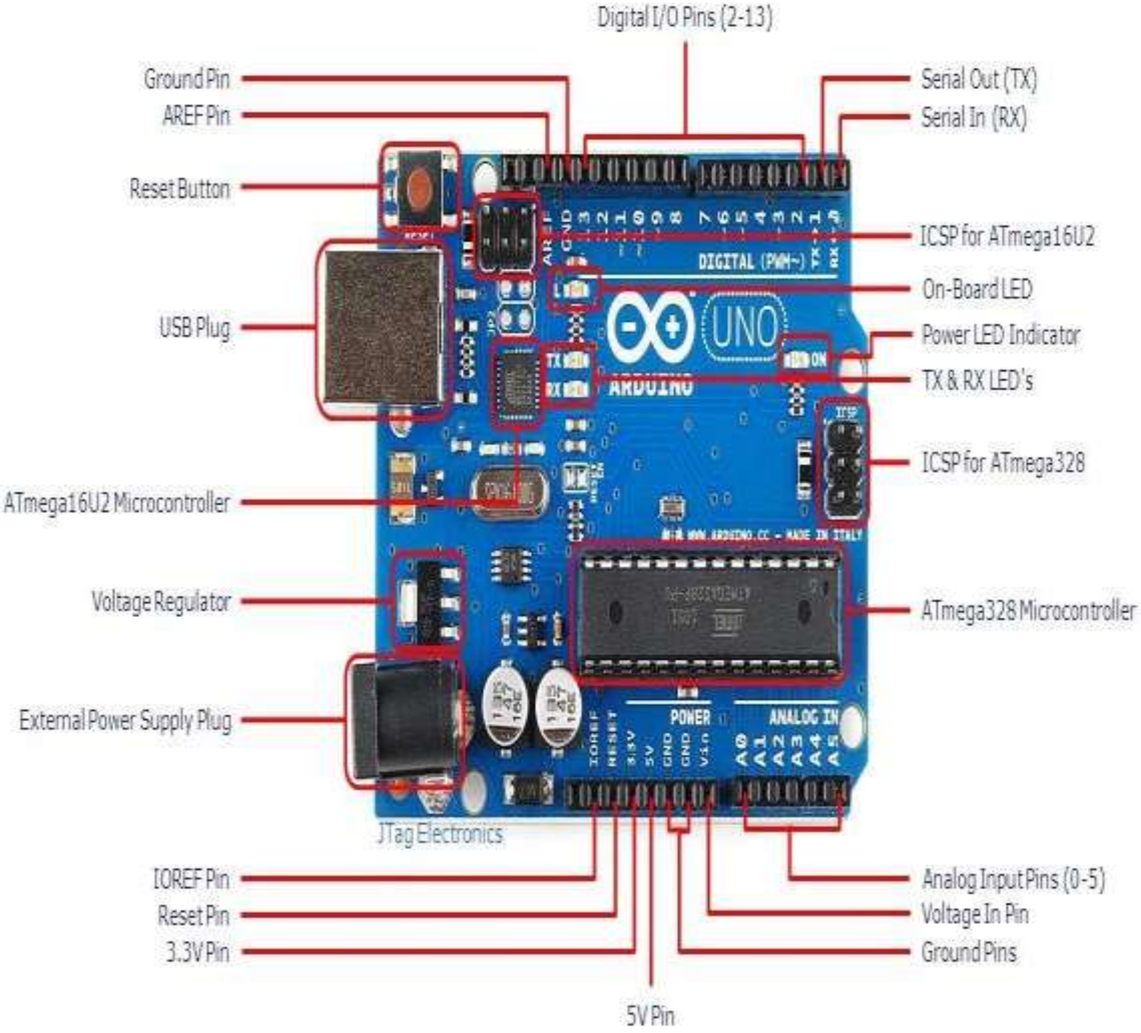


Fig 3.1ArduinoUno R3 Board

- **USB Plug & External Power Supply Plug**

Every Arduino board needs a way to be connected to a power source. The Arduino Uno can be powered from a USB cable coming from your computer or a wall power supply that is terminated in a barrel jack. The power source is selected automatically. The USB connection is also how you will load code onto your Arduino board. Please on my other post on how to program with Arduino can be found in Installing and Programming Arduino.

NOTE: The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable.If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts

.

- **Voltage Regulator:**

The voltage regulator is not actually something you can (or should) interact with on the Arduino. But it is potentially useful to know that it is there and what it's for. The voltage regulator does exactly what it says – it controls the amount of voltage that is let into the Arduino board. Think of it as a kind of gatekeeper; it will turn away an extra voltage that might harm the circuit. Of course,it has its limits, so don't hook up your Arduino to anything greater than 20 volts.

- **Power Pins:**

Voltage In Pin – The input voltage to the Arduino board when it's using an external power source(as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.

5V Pin – This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 – 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. It's not recommended.3.3V Pin – A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.

- **Ground Pins:**

There are several GND pins on the Arduino, any of which can be used to ground your

circuit.

- **IOREF Pin:**

This pin on the Arduino board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs for working with the 5V or 3.3V.

- **Input and Output Pins:**

Each of the 14 digital pins on the Uno can be used as an input or output. They operate at 5 volts. These pins can be used for both digital input (like telling if a button is pushed) and digital output (like powering an LED). Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-5k Ohms. In addition, some pins have specialized functions.

- **Serial Out (TX) & Serial In (RX):**

Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.

- **External Interrupts:**

Pins 2 and 3 can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.

PWM – You may have noticed the tilde (~) next to some of the digital pins (3, 5, 6, 9, 10, and 11). These pins act as normal digital pins, but can also be used for something called Pulse-Width Modulation (PWM). Think of these pins as being able to simulate analog output (like fading an LED in and out).

SPI – Pins 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). SPI stands for Serial Peripheral Interface. These pins support SPI communication using the SPI library.

Analog Input Pins – Labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). These pins can read the signal from an analog sensor (like a temperature sensor) and convert it into a digital value that we can read. By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF Pin (Stands for Analog Reference. Most of the time you can leave this pin alone). Additionally, some pins have specialized functionality:

TWI – Pins A4 or SDA pin and A5 or SCL pin. Support TWI communication using the Wire library.

- **Reset Pin**

Bring this line LOW to reset the microcontroller.Typically used to add a reset button to shields which block the one on the board.

- **LED Indicators**

Power LED Indicator – Just beneath and to the right of the word "UNO" on your circuit board, there's a tiny LED next to the word 'ON'. This LED should light up whenever you plug your Arduino into a power source. If this light doesn't turn on, there's a good chance something is wrong. Time to re-check your circuit!

On-Board LED – There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off. This useful to quickly check if the board has no problem as some boards has a pre-loaded simple blinking LED program in it.

TX & RX LEDs – These LEDs will give us some nice visual indications whenever our Arduino is receiving or transmitting data (like when we're loading a new program ontotheboard).

Reset Button:Pushing the reset button temporarily connect the reset pin to ground and restart

any code that is loaded on the Arduino. This can be very useful if your code doesn't repeat, but you want to test it multiple times.

4.9. POWER SUPPLY:

The power supplies are designed to convert high voltage AC mains electricity to a suitable low voltage supply for electronic circuits and other devices. A power supply can by broken down into a series of blocks, each of which performs a particular function. A d.c power supply which maintains the output voltage constant irrespective of a.c mains fluctuations or load variations is known as "Regulated D.C Power Supply".

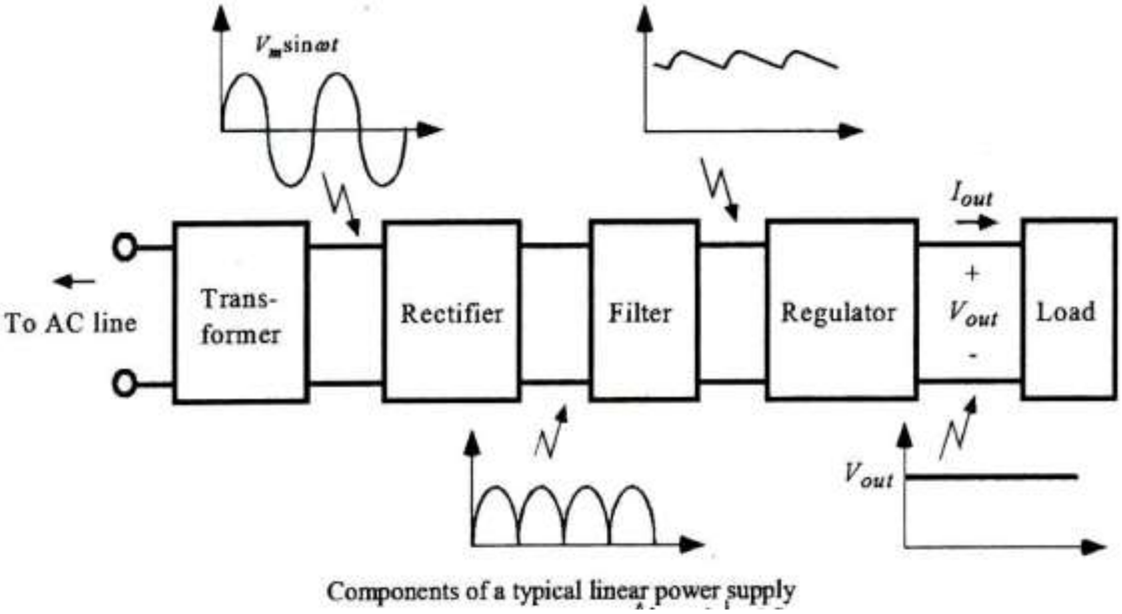


Fig:4.4. Block Diagram of

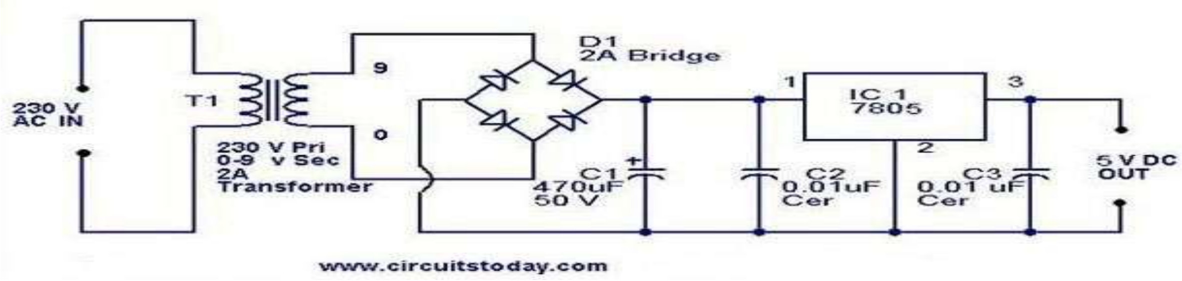


Fig:4.5. Schematic Diagram of Power Supply

1. TRANSFORMER:

A transformer is an electrical device which is used to convert electrical power from one Electrical circuit to another without change in frequency.

When AC is applied to the primary winding of the power transformer it can either be stepped down or up depending on the value of DC needed. In our circuit the transformer of 230v/12-0-12v is used to perform the step down operation where a 230V AC appears as 12V AC across the secondary winding.

2. RECTIFIER:

A circuit which is used to convert a.c to dc is known as RECTIFIER. The process of conversion a.c to d.c is called “rectification.

Bridge Rectifier:

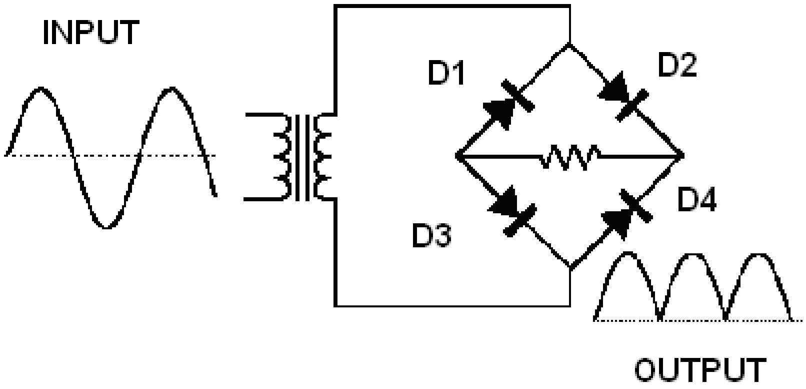


Fig: 4.6 Bridge Rectifier

OPERATION:

During positive half cycle of secondary, the diodes D2 and D3 are in forward biased while D1 and D4 are in reverse biased. During negative half cycle of secondary voltage, the diodes D1 and D4 are in forward biased while D2 and D3 are in reverse biased.

3. FILTER:

A Filter is a device which removes the a.c component of rectifier output but allows the d.c component to reach the load.We have seen that the ripple content in the rectified output of half wave rectifier is **121%** or that of full-wave or bridge rectifier or bridge rectifier is **48%** such high percentages of ripples is not acceptable for most of the applications. Ripples can be removed by one of the following methods of filtering. A capacitor, in parallel to the load, provides an easier by –pass for the ripples voltage though it due to low impedance. At ripple frequency and leave the d.c.to appears the load.

4. VOLTAGE REGULATOR:

As the name itself implies, it regulates the input applied to it. A voltage regulator is an electrical regulator designed to automatically maintain a constant voltage level. In this project, power supply of 5V and 12V are required. In order to obtain these voltage levels,7805 and 7812 voltage regulators are to be used. The first number 78 represents positive supply and the numbers 05,12 represent the required output voltage.

4.8. RS232 CABLE:

To allow compatibility among data communication equipment, an interfacing standard called RS232 is used. Since the standard was set long before the advent of the TTL logic family, its input and output voltage levels are not TTL compatible. For this reason, to connect any RS232 to a microcontroller system, voltage converters such as MAX232 are used to convert the TTL logic levels to the RS232 voltage levels and vice versa.

4.8.1. MAX232 IC:

Max232 IC is a specialized circuit which makes standard voltages as required by RS232 standards. This IC provides best noise rejection and very reliable against discharges and short circuits. MAX232 IC chips are commonly referred to as line drivers.

To ensure data transfer between PC and microcontroller, the baud rate and voltage levels of Microcontroller and PC should be the same. The voltage levels of microcontroller are logic1 and logic 0 i.e., logic 1 is +5V and logic 0 is 0V. But for PC, RS232 voltage levels are considered and they are: logic 1 is taken as -3V to -25V and logic 0 as +3V to +25V. So, in order to equal these voltage levels, MAX232 IC is used. Thus this IC converts RS232 voltage levels to microcontroller voltage levels and vice versa.

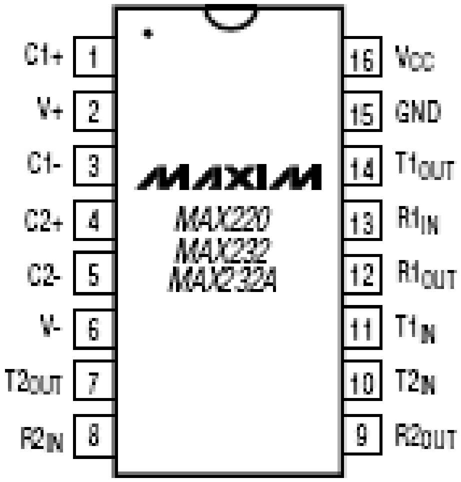


Fig:4.3 Pin diagram of MAX232 IC

4.4. POWER SUPPLY:

The power supplies are designed to convert high voltage AC mains electricity to a suitable low voltage supply for electronic circuits and other devices. A power supply can by broken down into a series of blocks, each of which performs a particular function. A d.c power supply which maintains the output voltage constant irrespective of a.c mains fluctuations or load variations is known as "Regulated D.C Power Supply".

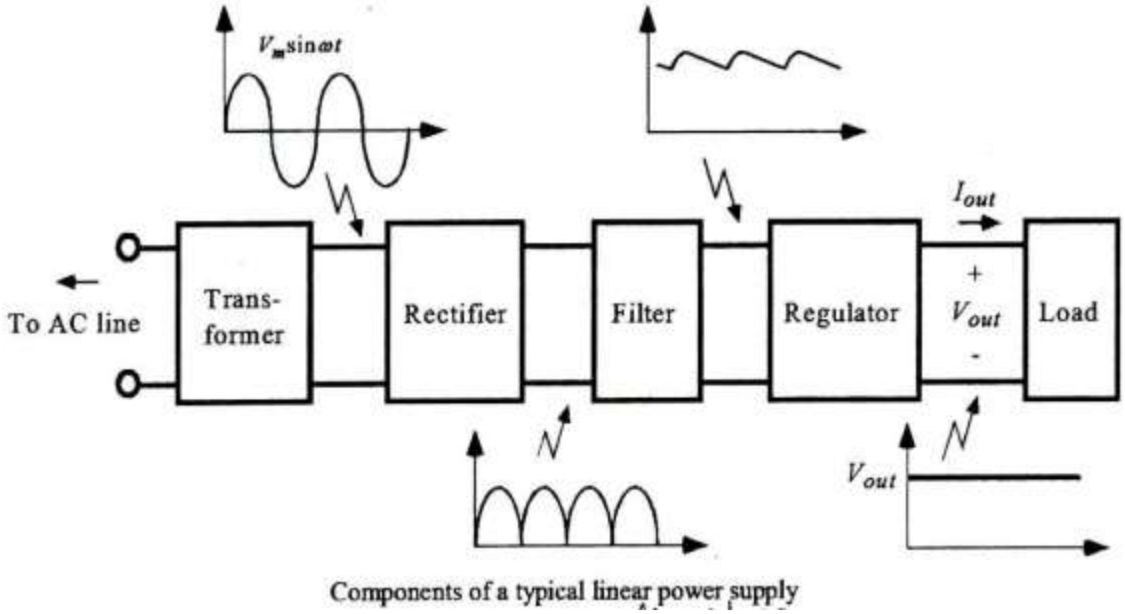
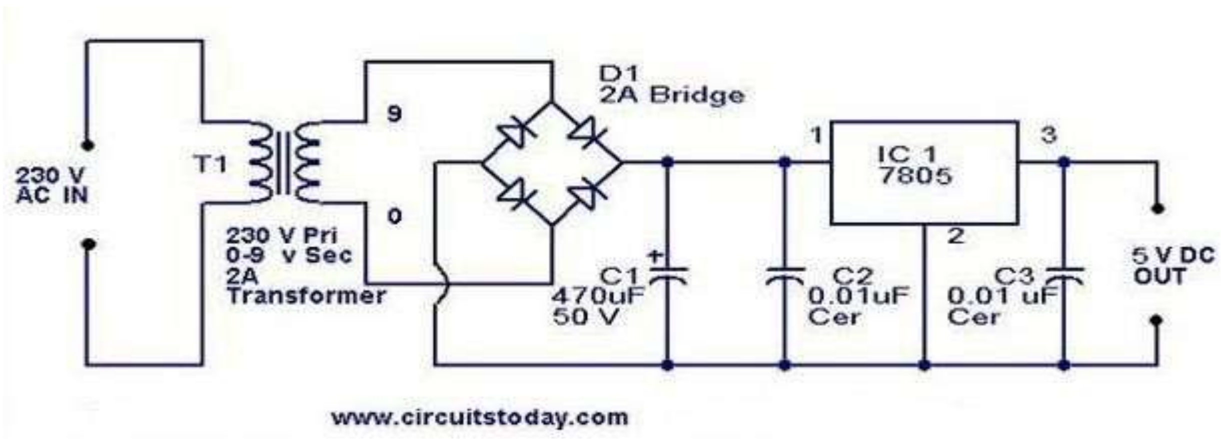


Fig:4.7. Block Diagram of Power Supply



www.circuitstoday.com

Fig:4.5. Schematic Diagram of Power Supply

1. TRANSFORMER:

A transformer is an electrical device which is used to convert electrical power from one Electrical circuit to another without change in frequency.

When AC is applied to the primary winding of the power transformer it can either be stepped down or up depending on the value of DC needed. In our circuit the transformer of 230v/12-0-12v is used to perform the step down operation where a 230V AC appears as 12V AC across the secondary winding.

2. RECTIFIER:

A circuit which is used to convert a.c to dc is known as RECTIFIER. The process of conversion a.c to d.c is called "rectification.

Bridge Rectifier:

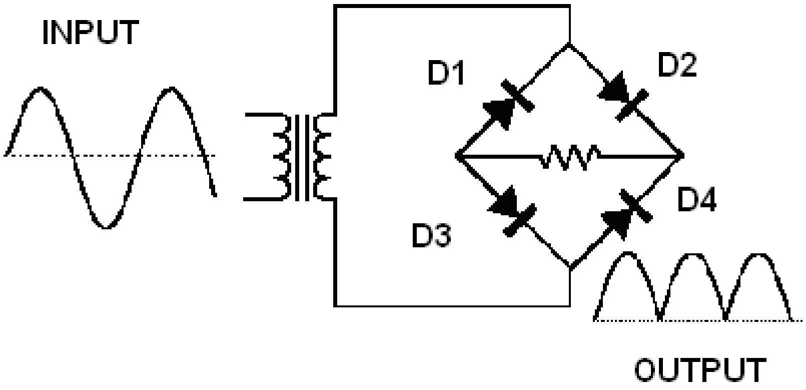


Fig: 4.6 Bridge Rectifier

OPERATION:

During positive half cycle of secondary, the diodes D2 and D3 are in forward biased while D1 and D4 are in reverse biased. During negative half cycle of secondary voltage, the diodes D1 and D4 are in forward biased while D2 and D3 are in reverse biased.

3. FITER

A Filter is a device which removes the a.c component of rectifier output but allows the d.c component to reach the load.We have seen that the ripple content in the rectified output of half wave rectifier is **121%** or that of full-wave or bridge rectifier or bridge rectifier is **48%** such high percentages of ripples is not acceptable for most of the applications. Ripples can be removed by one of the following methods of filtering. A capacitor, in parallel to the load, provides an easier by –pass for the ripples voltage though it due to low impedance. At ripple frequency and leave the d.c.to appears the load.

4. VOLTAGE REGULATOR:

As the name itself implies, it regulates the input applied to it. A voltage regulator is an electrical regulator designed to automatically maintain a constant voltage level. In this project, power supply of 5V and 12V are required. In order to obtain these voltage levels,7805 and 7812 voltage regulators are to be used. The first number 78 represents positive supply and the numbers 05,12 represent the required output voltage.

5. COMMUNICATION NETWORK

In health monitoring system, wireless network is used to forward measurement through a gateway towards cloud. The main network used here is IoT. The meaning of IoT is Internet of Things, simply called as Internet of everything. Different wireless communication technologies can be used for

(i) connecting the IoT device as local networks, and

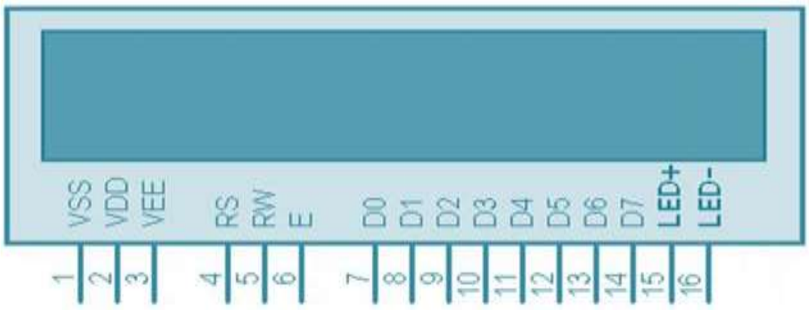
(ii)connecting these local networks (or individual IoT devices) to the Internet.

The connectivity technologies are NFC, Bluetooth, zigbee, cellular network etc. In this project, we use cellular network connectivity because of it has widespread mobile networks like 3G and LTE provide reliable high-speed connectivity to the Internet. However, they have a high power consumption profile and they are not suitable for M2M or local network communication.

6. LCD DISPLAY

Liquid Crystal Display also called as LCD is very helpful in providing user interface as well as for debugging purpose. The most commonly used Character based LCDs are based on Hitachi's HD44780 controller or other which are compatible with HD44580. The most commonly used LCDs found in the market today are 1 Line, 2 Line or 4 Line LCDs which have only 1 controller and support at most of 80 characters, whereas LCDs supporting more than 80 characters make use of 2 HD44780 controllers.

Pin Description:



Pin No.	Name	Description
1	**VSS**	Power supply (GND)
2	**VCC**	Power supply (+5V)
3	**VEE**	Contrast adjust
4	**RS**	0 = Instruction input 1 = Data input
5	**R/W**	0 = Write to LCD module 1 = Read from LCD module
6	**EN**	Enable signal
7	**D0**	Data bus line 0 (LSB)
8	**D1**	Data bus line 1
9	**D2**	Data bus line 2

10	**D3**	Data bus line 3
11	**D4**	Data bus line 4
12	**D5**	Data bus line 5
13	**D6**	Data bus line 6
14	**D7**	Data bus line 7 (MSB)
15	**LED+**	Back Light VCC
16	**LED-**	Back Light GND

DDRAM - Display Data RAM

Display data RAM (DDRAM) stores display data represented in 8-bit character codes. Its extended capacity is 80 X 8 bits, or 80 characters. The area in display data RAM (DDRAM) that is not used for display can be used as general data RAM. So whatever you send on the DDRAM is actually displayed on the LCD. For LCDs like 1x16, only 16 characters are visible, so whatever you write after 16 chars is written in DDRAM but is not visible to the user.

CGROM - Character Generator ROM

Now you might be thinking that when you send an ASCII value to DDRAM, how the character is displayed on LCD? So the answer is CGROM. The character generator ROM generates 5 x 8 dot or 5 x 10 dot character patterns from 8-bit character codes. It can generate 208 5 x 8 dot character patterns and 32 5 x 10 dot character patterns.

CGRAM - Character Generator RAM

As clear from the name, CGRAM area is used to create custom characters in LCD. In the character generator RAM, the user can rewrite character patterns by program. For 5 x 8 dots, eight character patterns can be written, and for 5 x 10 dots, four character patterns can be written.

BF - Busy Flag

Busy Flag is a status indicator flag for LCD. When we send a command or data to the LCD for processing, this flag is set (i.e. BF =1) and as soon as the instruction is executed successfully this flag is cleared (BF = 0). This is helpful in producing and exact amount of delay for the LCD processing.

To read Busy Flag, the condition RS = 0 and R/W = 1 must be met and The MSB of the LCD data bus (D7) act as busy flag. When BF = 1 means LCD is busy and will not accept next command or data and BF = 0 means LCD is ready for the next command or data to process.

Instruction Register (IR) and Data Register (DR)

There are two 8-bit registers in HD44780 controller Instruction and Data register. Instruction register corresponds to the register where you send commands to LCD e.g. LCD shift command, LCD clear, LCD address etc. and Data register is used for storing data which is to be displayed on LCD. When send the enable signal of the LCD is asserted, the data on the pins is latched in to the data register and data is then moved automatically to the DDRAM and hence is displayed on the LCD. Data Register is not only used for sending data to DDRAM but also for CGRAM, the address where you want to send the data, is decided by the instruction you send to LCD.

Commands and Instruction set

Only the instruction register (IR) and the data register (DR) of the LCD can be controlled by the MCU. Before starting the internal operation of the LCD, control information is temporarily stored into these registers to allow interfacing with various MCUs, which operate at different speeds, or various peripheral control devices. The internal operation of the LCD is determined by signals sent from the MCU. These signals, which include register selection signal (RS), read/write signal (R/W), and the data bus (DB0 to DB7), make up the LCD instructions (Table 3). There are four categories of instructions that:

- Designate LCD functions, such as display format, data length, etc.
- Set internal RAM addresses
- Perform data transfer with internal RAM

- **Perform miscellaneous functions**

Although looking at the table you can make your own commands and test them. Below is a brief list of useful commands which are used frequently while working on the LCD.

No.	Instruction	Hex	Decimal
1	Function Set: 8-bit, 1 Line, 5x7 Dots	0x30	48
2	Function Set: 8-bit, 2 Line, 5x7 Dots	0x38	56
3	Function Set: 4-bit, 1 Line, 5x7 Dots	0x20	32
4	Function Set: 4-bit, 2 Line, 5x7 Dots	0x28	40
5	Entry Mode	0x06	6
6	Display off Cursor off (clearing display without clearing DDRAM content)	0x08	8
7	Display on Cursor on	0x0E	14
8	Display on Cursor off	0x0C	12
9	Display on Cursor blinking	0x0F	15
10	Shift entire display left	0x18	24
12	Shift entire display right	0x1C	30
13	Move cursor left by one character	0x10	16
14	Move cursor right by one character	0x14	20
15	Clear Display (also clear DDRAM content)	0x01	1
16	Set DDRAM address or cursor position on Display	0x80+add	128+add
17	Set CGRAM address or set pointer to CGRAM	0x40+add	64+add

Sending Commands to LCD

To send commands we simply need to select the command register. Everything is same as we have done in the initialization routine. But we will summarize the common steps and put them in a single subroutine. Following are the steps:

- move data to LCD port
- select command register
- select write operation
- send enable signal
- wait for LCD to process the command

Sending Data to LCD

To send data we simply need to select the data register. Everything is same as the command routine. Following are the steps:

- move data to LCD port
- select data register
- select write operation
- send enable signal
- wait for LCD to process the data

4.7.SWITCH

A push-button (also spelled pushbutton) or simply button is a simple switch mechanism for controlling some aspect of a machine or a process. Buttons are typically made out of hard material, usually plastic or metal. The surface is usually flat or shaped to accommodate the human finger or hand, so as to be easily depressed or pushed. Buttons are most often biased switches, although many un-biased buttons (due to their physical nature) still require a spring to return to their un-pushed state. Terms for the “pushing” of a button include pressing, depressing, mashing, hitting, and punching.

The "push-button" has been utilized in calculators, push-button telephones, kitchen appliances, and various other mechanical and electronic devices, home and commercial. In industrial and commercial applications, push buttons can be connected together by a mechanical linkage so that the act of pushing one button causes the other button to be released. In this way, a stop button can "force" a start button to be released. This method of linkage is used in simple manual operations in which the machine or process has no electrical circuits for control.

Red pushbuttons can also have large heads (called mushroom heads) for easy operation and to facilitate the stopping of a machine. These pushbuttons are called emergency stop buttons and for increased safety are mandated by the electrical code in many jurisdictions. This large mushroom shape can also be found in buttons for use with operators who need to wear gloves for their work and could not actuate a regular flush-mounted push button.

Fig: Switch

Radio-frequency identification

Radio-frequency identification (**RFID**) uses electromagnetic fields to automatically identify and track tags attached to objects. The tags contain electronically-stored information. Passive tags collect energy from a nearby RFID reader's interrogating radio waves. Active tags have a local power source (such as a battery) and may operate hundreds of meters from the RFID reader. Unlike a barcode, the tag need not be within the line of sight of the reader, so it may be embedded in the tracked object. RFID is one method of automatic identification and data capture (AIDC).[1]

RFID tags are used in many industries. For example, an RFID tag attached to an automobile during production can be used to track its progress through the assembly line; RFID-tagged pharmaceuticals can be tracked through warehouses; and implanting RFID microchips in livestock and pets enables positive identification of animals.

Since RFID tags can be attached to cash, clothing, and possessions, or implanted in animals and people, the possibility of reading personally-linked information without consent has raised serious privacy concerns.[2] These concerns resulted in standard specifications development addressing privacy and security issues. ISO/IEC 18000 and ISO/IEC 29167 use on- chip cryptography methods for untraceability, tag and reader authentication, and over-the-air privacy. ISO/IEC 20248 specifies a digital signature data structure for RFID and barcodes providing data, source and read method authenticity. This work is done within ISO/IEC JTC 1/SC 31 Automatic identification and data capture techniques. Tags can also be used in shops to expedite checkout, and to prevent theft by customers and employees.

In 2014, the world RFID market was worth US$8.89 billion, up from US$7.77 billion in 2013 and US$6.96 billion in 2012. This figure includes tags, readers, and software/services for RFID cards, labels, fobs, and all other form factors. The market value is expected to rise to US$18.68 billion by 2026.

Fig: FasTrack, an RFID tag used for electronic toll collection

In 1945, Léon Theremin invented a listening device for the Soviet Union which retransmitted incident radio waves with the added audio information. Sound waves vibrated a diaphragm which slightly altered the shape of the resonator, which modulated the reflected radio frequency. Even though this device was a covert listening device, rather than an identification tag, it is considered to be a predecessor of RFID because it was passive, being energized and activated by waves from an outside source.

Similar technology, such as the IFF transponder, was routinely used by the allies and Germany in World War II to identify aircraft as friend or foe. Transponders are still used by most powered aircraft. Another early work exploring RFID is the landmark 1948 paper by Harry Stockman, who predicted that "... considerable research and development work has to be done before the remaining basic problems in reflected-power communication are solved, and before the field of useful applications is explored."

Mario Cardullo's device, patented on January 23, 1973, was the first true ancestor[6] of modern RFID, as it was a passive radio transponder with memory. The initial device was passive, powered by the interrogating signal, and was demonstrated in 1971 to the New York Port Authority and other potential users. It consisted of a transponder with 16 bit memory for use as a toll device. The basic Cardullo patent covers the use of RF, sound and light as transmission media. The original business plan presented to investors in 1969 showed uses in transportation (automotive vehicle identification, automatic toll system, electronic license plate, electronic manifest, vehicle routing, vehicle performance monitoring), banking (electronic checkbook, electronic credit card), security (personnel identification, automatic gates, surveillance) and medical (identification, patient history).

An early demonstration of *reflected power* (modulated backscatter) RFID tags, both passive and semi-passive, was performed by Steven Depp, Alfred Koelle, and Robert Frayman at the Los Alamos National Laboratory in 1973 The portable system operated at 915 MHz and used 12- bit tags. This technique is used by the majority of today's UHFID and microwave RFID tags.

The first patent to be associated with the abbreviation RFID was granted to Charles Walton in 1983

Fig: RFID hard tag ‘a3tag’by Retailer Advantage

A radio-frequency identification system uses *tags*, or *labels* attached to the objects to be identified. Two-way radio transmitter-receivers called *interrogators* or *readers* send a signal to the tag and read its response

RFID tags can be either passive, active or battery-assisted passive. An active tag has an on-board battery and periodically transmits its ID signal. A battery-assisted passive (BAP) has a small battery on board and is activated when in the presence of an RFID reader. A passive tag is cheaper and smaller because it has no battery; instead, the tag uses the radio energy transmitted by the reader. However, to operate a passive tag, it must be illuminated with a power level roughly a thousand times stronger than for signal transmission. That makes a difference in interference and in exposure to radiation.

Tags may either be read-only, having a factory-assigned serial number that is used as a key into a database, or may be read/write, where object-specific data can be written into the tag by the system user. Field programmable tags may be write-once, read-multiple; "blank" tags may be written with an electronic product code by the user.

RFID tags contain at least three parts: an integrated circuit that stores and processes information and that modulates and demodulates radio-frequency (RF) signals; a means of collecting DC power from the incident reader signal; and an antenna for receiving and transmitting the signal. The tag information is stored in a non-volatile memory. The RFID tag includes either fixed or programmable logic for processing the transmission and sensor data, respectively.

An RFID reader transmits an encoded radio signal to interrogate the tag. The RFID tag receives the message and then responds with its identification and other information. This may be only a unique tag serial number, or may be product-related information such as a stock number, lot or batch number, production date, or other specific information. Since tags have individual serial numbers, the RFID system design can discriminate among several tags that might be within the range of the RFID reader and read them simultaneously.

RFID systems can be classified by the type of tag and reader. A **Passive Reader Active Tag** (**PRAT**) system has a passive reader which only receives radio signals from active tags (battery operated, transmit only). The reception range of a PRAT system reader can be adjusted from 1–2,000 feet (0–600 m), allowing flexibility in applications such as asset protection and supervision.

An **Active Reader Passive Tag** (**ARPT**) system has an active reader, which transmits interrogator signals and also receives authentication replies from passive tags.

An **Active Reader Active Tag** (**ARAT**) system uses active tags awoken with an interrogator signal from the active reader. A variation of this system could also use a Battery-Assisted Passive (BAP) tag which acts like a passive tag but has a small battery to power the tag's return reporting signal.

Fixed readers are set up to create a specific interrogation zone which can be tightly controlled. This allows a highly defined reading area for when tags go in and out of the interrogation zone. Mobile readers may be hand-held or mounted on carts or vehicles.

Frequencies:

RFID frequency bands[12][13]

Band	**Regulations**	**Range**	**Data speed**	**ISO/IEC 18000section**	**Remarks**	**Approximate tag cost in volume (2006) US $**
120–150 kHz (LF)	Unregulated	10 cm	Low	Part 2	Animal identification, factory data collection	$1
13.56 MHz (HF)	ISM bandworldwide	10 cm –1 m	Low to moderate	Part 3	Smart cards (ISO/IEC 15693, ISO/IEC 14443 A, B). Non fully ISO compatible memory cards (Mifare Classic, iCLASS, Legic, Felica ...). Micro	$0.50 to $5

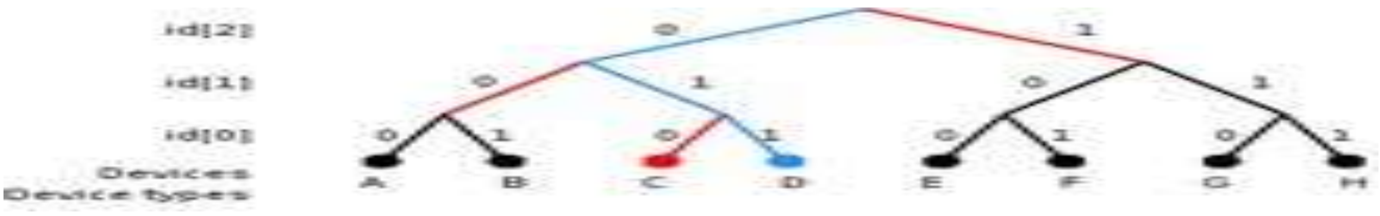
					processor ISO compatible cards (Desfire EV1, Seos)	
433 MHz (UHF)	Short range devices	1–100 m	Moderate	Part 7	Defense applications, with active tags	$5
865–868 MHz (Europe) 902–928 MHz (North America) UHF	ISM band	1–12 m	Moderate to high	Part 6	EAN, various standards; used by railroads[14]	$0.15 (passive tags)
2450–5800 MHz (microwave)	ISM band	1–2 m	High	Part 4	802.11 WLAN, Bluetooth standards	$25 (active tags)
3.1–10 GHz (microwave)	Ultra wide band	Up to 200 m	High	Not defined	Requires semi-active or active tags	$5 projected

Signaling:

Signaling between the reader and the tag is done in several different incompatible ways, depending on the frequency band used by the tag. Tags operating on LF and HF bands are, in terms of radio wavelength, very close to the reader antenna because they are only a small percentage of a wavelength away. In this near field region, the tag is closely coupled electrically with the transmitter in the reader. The tag can modulate the field produced by the reader by changing the electrical loading the tag represents. By switching between lower and higher relative loads, the tag produces a change that the reader can detect. At UHF and higher frequencies, the tag is more than one radio wavelength away from the reader, requiring a different approach. The tag can backscatter a signal. Active tags may contain functionally separated transmitters and receivers, and the tag need not respond on a frequency related to the reader's interrogation signal

An Electronic Product Code (EPC) is one common type of data stored in a tag. When written into the tag by an RFID printer, the tag contains a 96-bit string of data. The first eight bits are a header which identifies the version of the protocol. The next 28 bits identify the organization that manages the data for this tag; Rather like a URL, the total electronic product code number can be used as a key into a global database to uniquely identify a particular product

Often more than one tag will respond to a tag reader, for example, many individual products with tags may be shipped in a common box or on a common pallet. Collision detection is important to allow reading of data. Two different types of protocols are used to "singulate" a particular tag, allowing its data to be read in the midst of many similar tags. In a slotted Aloha system, the reader broadcasts an initialization command and a parameter that the tags individually use to pseudo-randomly delay their responses. When using an "adaptive binary tree" protocol, the reader sends an initialization symbol and then transmits one bit of ID data at a time; only tags with matching bits respond, and eventually only one tag matches the complete ID string



An example of a binary tree method of identifying anRFID rag

Both methods have drawbacks when used with many tags or with multiple overlapping readers.

Bulk reading

"Bulk reading" is a strategy for interrogating multiple tags at the same time, but lacks sufficient precision for inventory control. A group of objects, all of them RFID tagged, are read completely from one single reader position at one time. Bulk reading is a possible use of HF (ISO 18000- 3), UHF (ISO 18000-6) and SHF (ISO 18000-4) RFID tags. However, as tags respond strictly sequentially, the time needed for bulk reading grows linearly with the number of labels to be read. This means it takes at least twice as long to read twice as many labels. Due to collision effects, the time required is greater.
A group of tags has to be illuminated by the interrogating signal just like a single tag. This is not a challenge concerning energy, but with respect to visibility; if any of the tags are shielded by other tags, they might not be sufficiently illuminated to return a sufficient response. The response conditions for inductively coupled HF RFID tags and coil antennas in magnetic fields appear better than for UHF or SHF dipole fields, but then distance limits apply and may prevent success.

Under operational conditions, bulk reading is not reliable. Bulk reading can be a rough guide for logistics decisions, but due to a high proportion of reading failures, it is not (yet) suitable for inventory management. However, when a single RFID tag might be seen as not guaranteeing a proper read, a bunch of RFID tags, where at least one will respond, may be a safer approach for detecting a known grouping of objects. In this respect, bulk reading is a fuzzy method for process support. From the perspective of cost and effect, bulk reading is not reported as an economical approach to secure process control in logistics.

Miniaturization

RFIDs are easy to conceal or incorporate in other items. For example, in 2009 researchers at Bristol University successfully glued RFID micro-transponders to live ants in order to study their behavior.[18] This trend towards increasingly miniaturized RFIDs is likely to continue as technology advances.

Hitachi holds the record for the smallest RFID chip, at 0.05 mm × 0.05 mm. This is 1/64th the size of the previous record holder, the mu-chip.[19] Manufacture is enabled by using the silicon- on-insulator (SOI) process. These dust-sized chips can store 38-digit numbers using 128-bit Read Only Memory (ROM)[] A major challenge is the attachment of antennas, thus limiting read range to only millimeters.

Uses

The RFID tag can be affixed to an object and used to track and manage inventory, assets, people, etc. For example, it can be affixed to cars, computer equipment, books, mobile phones, etc.

RFID offers advantages over manual systems or use of bar codes. The tag can be read if passed near a reader, even if it is covered by the object or not visible. The tag can be read inside a case, carton, box or other container, and unlike barcodes, RFID tags can be read hundreds at a time. Bar codes can only be read one at a time using current devices.

In 2011, the cost of passive tags started at US$0.09 each; special tags, meant to be mounted on metal or withstand gamma sterilization, can go up to US$5. Active tags for tracking containers, medical assets, or monitoring environmental conditions in data centers start at US$50 and can go up over US$100 each. Battery-Assisted Passive (BAP) tags are in the US$3–10 range and also have sensor capability like temperature and humidity

RFID can be used in a variety of applications such as:

1. Place the 'RF Contactless' key card onto the door lock sensor above the handle.
2. When you see the green light above the lock sensor is on, turn the handle to open the door.
3. If the red light is on instead, re-try from Step 1.
4. Should it fail to open the door, please contact the Front Desk for assistance.

Fig: Electronic key for RFID lock system

- Access management
- Tracking of goods
- Tracking of persons and animals
- Toll collection and contactless payment Machine readable travel documents
- Smartdust (for massively distributed sensor networks) Airport baggage
- tracking logistics
- Timing sporting events
- Tracking and billing process

In 2010 three factors drove a significant increase in RFID usage: decreased cost of equipment and tags, increased performance to a reliability of 99.9% and a stable international standard around UHF passive RFID. The adoption of these standards were driven by EPCglobal, a joint venture between GS1 and GS1 US, which were responsible for driving global adoption of the barcode in the 1970s and 1980s. The EPCglobal Network was developed by the Auto-ID Center.

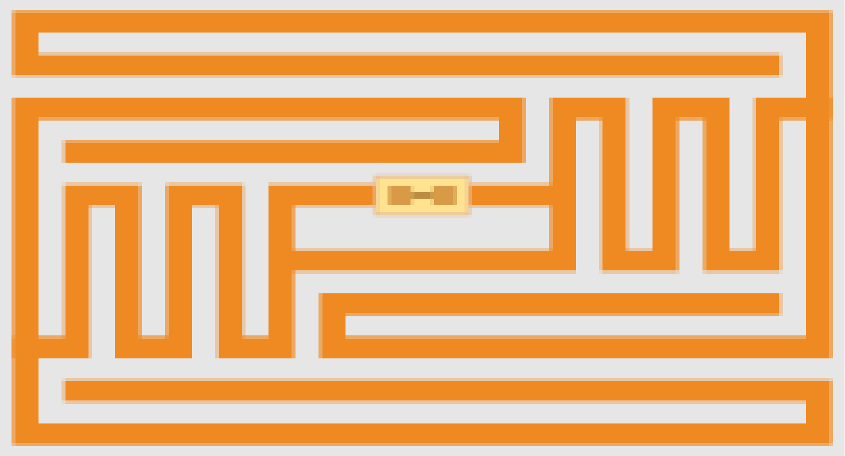


Fig:An EPC RFID tag used by wal-mart

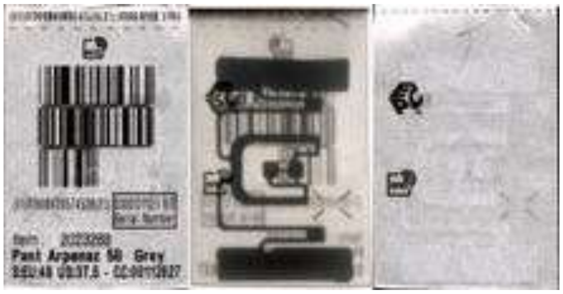


Fig: Sewn-in RFID label in garment manufacture by the French sport supplier Decathlon. Front,back,and transparency scan

RFID provides a way for organizations to identify and manage stock, tools and equipment (asset tracking), etc. without manual data entry. Manufactured products such as automobiles or garments can be tracked through the factory and through shipping to the customer. Automatic identification with RFID can be used for inventory systems. Many organisations require that their vendors place RFID tags on all shipments to improve supply chain management.

Retail

RFID is used for item level tagging in retail stores. In addition to inventory control, this provides both protection against theft by customers (shoplifting) and employees ("shrinkage") by using electronic article surveillance (EAS), and a self checkout process for customers. Tags of different type can be physically removed with a special tool or deactivated electronically once items have been paid for. On leaving the shop customers have to pass near an RFID detector; if they have items with active RFID tags, an alarm sounds, both indicating an unpaid-for item, and identifying what it is.

Casinos can use RFID to authenticate poker chips, and can selectively invalidate any chips known to be stolen.

Fig: Access antenna for vehicle access control

Access control

RFID tags are widely used in identification badges, replacing earlier magnetic stripe cards. These badges need only be held within a certain distance of the reader to authenticate the holder. Tags can also be placed on vehicles, which can be read at a distance, to allow entrance to controlled areas without having to stop the vehicle and present a card or enter an access code.

Advertising

In 2010 Vail Resorts began using UHF Passive RFID tags in ski passes. Facebook is using RFID cards at most of their live events to allow guests to automatically capture and post photos. The automotive brands have adopted RFID for social media product placement more quickly than other industries. Mercedes was an early adopter in 2011 at the PGA Golf Championships, and by the 2013 Geneva Motor Show many of the larger brands were using RFID for social media marketing.

Promotion tracking

To prevent retailers diverting products, manufacturers are exploring the use of RFID tags on promoted merchandise so that they can track exactly which product has sold through the supply chain at fully discounted prices.

Transportation and logistics

Yard management, shipping and freight and distribution centers use RFID tracking. In the railroad industry, RFID tags mounted on locomotives and rolling stock identify the owner, identification number and type of equipment and its characteristics. This can be used with a database to identify the lading, origin, destination, etc. of the commodities being carried.

In commercial aviation, RFID is used to support maintenance on commercial aircraft. RFID tags are used to identify baggage and cargo at several airports and airlines.

Some countries are using RFID for vehicle registration and enforcement. RFID can help detect and retrieve stolen cars.

Fig: E-Z pass reader attached to the pole and mast arm (right) used in traffic monitoring in New York City

RFID is used in intelligent transportation systems. In New York City, RFID readers are deployed at intersections to track E-ZPass tags as a means for monitoring the traffic flow. The data are fed through the broadband wireless infrastructure to the traffic management center to be used in adaptive traffic control of the traffic lights.

Hose stations and conveyance of fluids

The RFID antenna in a permanently installed coupling half (fixed part) unmistakably identifies the RFID transponder placed in the other coupling half (free part) after completed coupling. When connected the transponder of the free part transmits all important information contactless to the fixed part. The coupling's location can be clearly identified by the RFID transponder coding. The control is enabled to automatically start subsequent process steps.

Track & Trace test vehicles and prototype parts

In the automotive industry RFID is used to Track & Trace test vehicles and prototype parts (project Transparent Prototype).

Infrastructure management and protection

At least one company has introduced RFID to identify and locate underground infrastructure assets such as gas pipelines, sewer lines, electrical cables, communication cables, etc.[38]

Passports

The first RFID passports ("E-passport") were issued by Malaysia in 1998. In addition to information also contained on the visual data page of the passport, Malaysian e-passports record the travel history (time, date, and place) of entries and exits from the country.

Other countries that insert RFID in passports include Norway (2005),-Japan (March 1, 2006), most EU countries (around 2006), Australia, Hong Kong, the United States (2007), India (June 2008), Serbia (July 2008), Republic of Korea (August 2008), Taiwan (December 2008), Albania(January 2009), The Philippines (August 2009), Republic of Macedonia (2010), Canada (2013)and Israel (2017).

Standards for RFID passports are determined by the International Civil Aviation Organization (ICAO), and are contained in ICAO Document 9303, Part 1, Volumes 1 and 2 (6th edition, 2006). ICAO refers to the ISO/IEC 14443 RFID chips in e-passports as "contactless integrated circuits". ICAO standards provide for e-passports to be identifiable by a standard e- passport logo on the front cover.

Since 2006, RFID tags included in new United States passports will store the same information that is printed within the passport, and include a digital picture of the owner. The United States Department of State initially stated the chips could only be read from a distance of 10 centimetres (3.9 in), but after widespread criticism and a clear demonstration that special equipment can read the test passports from 10 metres (33 ft) away, the passports were designed to incorporate a thin metal lining to make it more difficult for unauthorized readers to "skim" information when the passport is closed. The department will also implement Basic Access Control (BAC), which functions as a personal identification number (PIN) in the form of characters printed on the passport data page. Before a passport's tag can be read, this PIN must be entered into an RFID reader. The BAC also enables the encryption of any communication between the chip and interrogator. As noted in the section below on security, there are many situations in which these protections have been shown to be insufficient, and passports have been cloned based on scans of them while they were being delivered in the mail.

Transportation payments

In many countries, RFID tags can be used to pay for mass transit fares on bus, trains, or subways, or to collect tolls on highways.

Some bike lockers are operated with RFID cards assigned to individual users. A prepaid card is required to open or enter a facility or locker and is used to track and charge based on how long the bike is parked.

The Zipcar car-sharing service uses RFID cards for locking and unlocking cars and for member identification.

In Singapore, RFID replaces paper Season Parking Ticket (SPT).

Animal identification

RFID tags for animals represent one of the oldest uses of RFID. Originally meant for large ranches and rough terrain, since the outbreak of mad-cow disease, RFID has become crucial in animal identification management. An implantable RFID tag or transponder can also be used for animal identification. The transponders are better known as PIT (Passive Integrated Transponder) tags, passive RFID, or "chips" on animals.[45] The Canadian Cattle Identification Agency began using RFID tags as a replacement for barcode tags. Currently CCIA tags are used in Wisconsin and by United States farmers on a voluntary basis. The USDA is currently developing its own program.

RFID tags are required for all cattle sold in Australia and in some states, sheep and goats as well.

Human implantation

A surgeon implants British scientist Dr Mark Gasson in his left hand with an RFID microchip (March 16, 2009)

Biocompatible microchip implants that utilize RFID technology are being routinely implanted in humans. The first reported experiment with RFID implants was conducted by British professor of cybernetics Kevin Warwick who had an RFID chip implanted in his arm by his general practitioner George Boulos in 1998.[47][48] In 2004 the 'Baja Beach Clubs' operated by Conrad Chase in Barcelona[49] and Rotterdam offered implanted chips to identify their VIP customers,

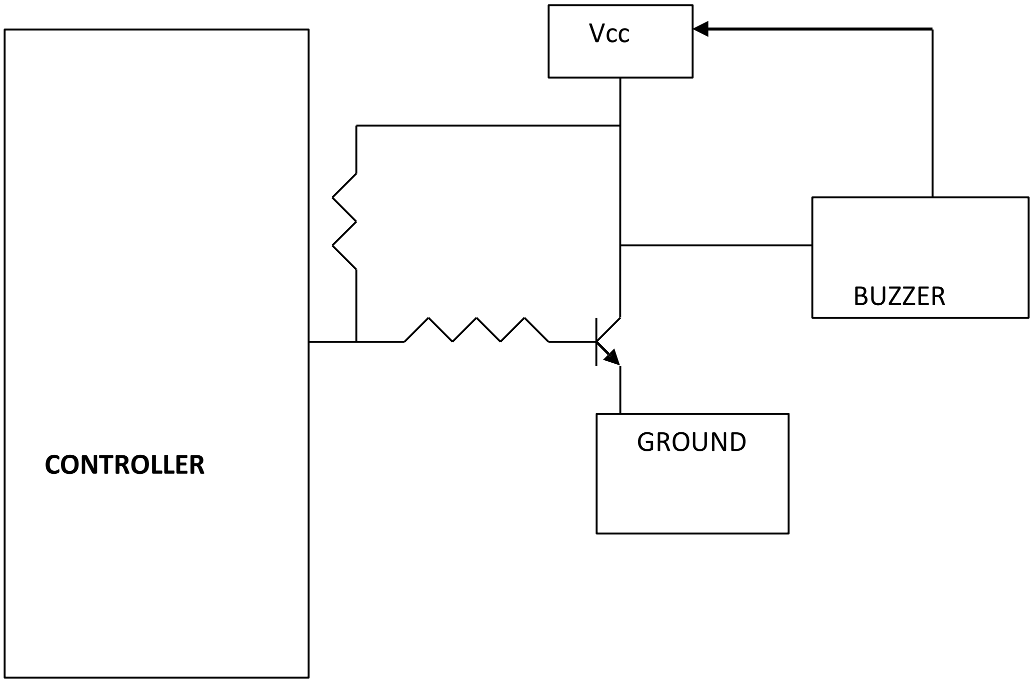
who could in turn use it to pay for service. In 2009 British scientist Mark Gasson had an advanced glass capsule RFID device surgically implanted into his left hand and subsequently demonstrated how a computer virus could wirelessly infect his implant and then be transmitted on to other systems.

The Food and Drug Administration in the United States approved the use of RFID chips in humans in 2004.

There is controversy regarding human applications of implantable RFID technology including concerns that individuals could potentially be tracked by carrying an identifier unique to them. Privacy advocates have protested against implantable RFID chips, warning of potential abuse. Some are concerned this could lead to abuse by an authoritarian government, to removal of freedoms,[52] and to the emergence of an "ultimate panopticon", a society where all citizens behave in a socially accepted manner because others might be watching.

BUZZER DRIVER CIRCUIT:

Digital systems and microcontroller pins lack sufficient current to drive the circuits like relays, buzzer circuits etc. While these circuits require around 10milli amps to be operated, the microcontroller’s pin can provide a maximum of 1-2milli amps current. For this reason, a driver such as a power transistor is placed in between the microcontroller and the buzzer circuit.



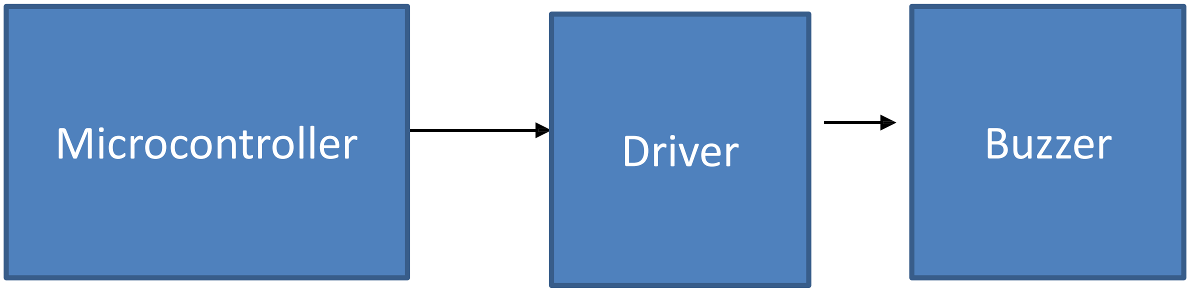
The operation of this circuit is as follows:

The input to the base of the transistor is applied from the microcontroller port pin P1.0.

The transistor will be switched on when the base to emitter voltage is greater than 0.7V (cut-in voltage). Thus when the voltage applied to the pin P1.0 is high i.e., P1.0=1 (>0.7V), the transistor will be switched on and thus the buzzer will be ON.

When the voltage at the pin P1.0 is low i.e., P1.0=0 (<0.7V) the transistor will be in off state and the buzzer will be OFF. Thus the transistor acts like a current driver to operate the buzzer accordingly.

BUZZER INTERFACING WITH THE MICROCONTROLLER :



5. DESIGN OF SOFTWARE

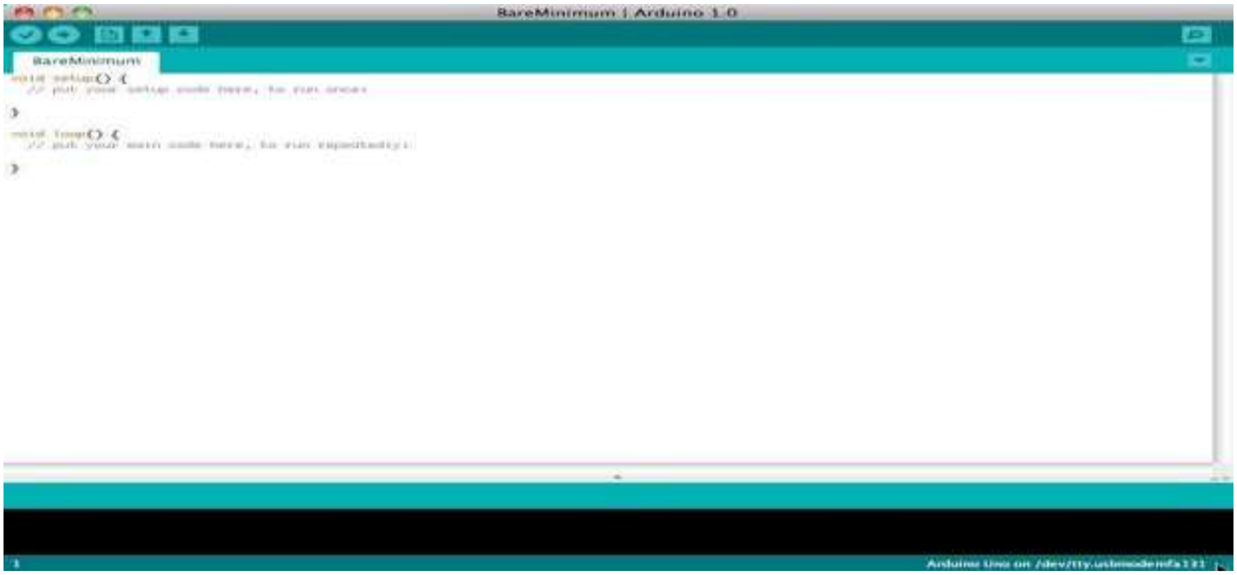
5.1. INTRODUCTION TO ARDUINO IDE SOFTWARE :

This is free software (evaluation version) which solves many of the pain points for an embedded system developer. This software is an Integrated Development Environment(IDE), which integrated text editor to write program, a compiler and it will convert your source code into HEX file. Here is simple guide to start working with Arduino IDE Vision which can be used for:

- Writing programs in Arduino IDE
- Compiling and assembling
- programs Debugging programs

5.2. SOFTWARE STEPS:

Before you can start doing anything with the Arduino, you need to download and install the Arduino IDE (integrated development environment).

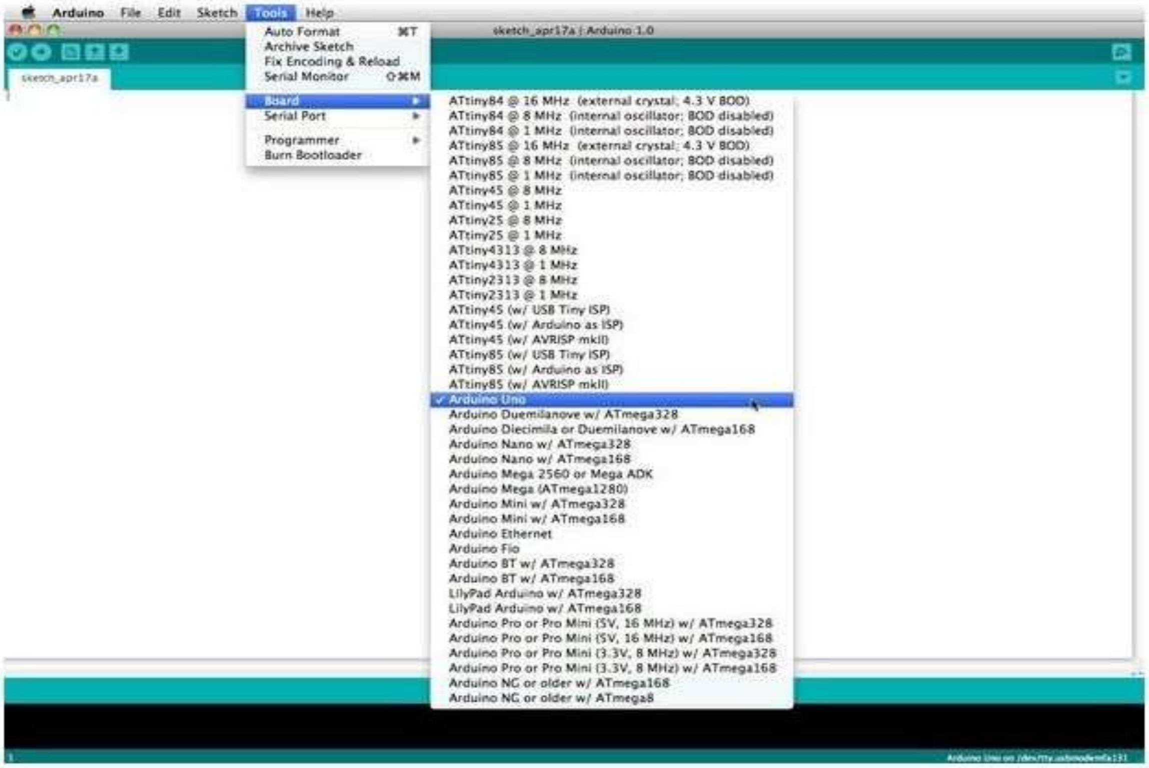


BareMinimum | Arduino 1.0
BareMinimum

```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

1
Arduino Uno on /dev/tty.usbmodemfa131

After the opening IDE the settings are changed in order to connect to the Arduino.


Arduino
File
Edit
Sketch
Tools
Help
sketch_apr17a | Arduino 1.0
sketch_apr17a
Auto Format ⌘T
Archive Sketch
Fix Encoding & Reload
Serial Monitor ⇧⌘M
Board
Serial Port
Programmer
Burn Bootloader
ATtiny84 @ 16 MHz (external crystal; 4.3 V BOD)
ATtiny84 @ 8 MHz (internal oscillator; BOD disabled)
ATtiny84 @ 1 MHz (internal oscillator; BOD disabled)
ATtiny85 @ 16 MHz (external crystal; 4.3 V BOD)
ATtiny85 @ 8 MHz (internal oscillator; BOD disabled)
ATtiny85 @ 1 MHz (internal oscillator; BOD disabled)
ATtiny45 @ 8 MHz
ATtiny45 @ 1 MHz
ATtiny25 @ 8 MHz
ATtiny25 @ 1 MHz
ATtiny4313 @ 8 MHz
ATtiny4313 @ 1 MHz
ATtiny2313 @ 8 MHz
ATtiny2313 @ 1 MHz
ATtiny45 (w/ USB Tiny ISP)
ATtiny45 (w/ Arduino as ISP)
ATtiny45 (w/ AVRISP mkII)
ATtiny85 (w/ USB Tiny ISP)
ATtiny85 (w/ Arduino as ISP)
ATtiny85 (w/ AVRISP mkII)
✓ Arduino Uno
Arduino Duemilanove w/ ATmega328
Arduino Diecimila or Duemilanove w/ ATmega168
Arduino Nano w/ ATmega328
Arduino Nano w/ ATmega168
Arduino Mega 2560 or Mega ADK
Arduino Mega (ATmega1280)
Arduino Mini w/ ATmega328
Arduino Mini w/ ATmega168
Arduino Ethernet
Arduino Fio
Arduino BT w/ ATmega328
Arduino BT w/ ATmega168
LilyPad Arduino w/ ATmega328
LilyPad Arduino w/ ATmega168
Arduino Pro or Pro Mini (5V, 16 MHz) w/ ATmega328
Arduino Pro or Pro Mini (5V, 16 MHz) w/ ATmega168
Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ ATmega328
Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ ATmega168
Arduino NG or older w/ ATmega168
Arduino NG or older w/ ATmega8
Arduino Uno on /dev/tty.usbmodemfa131

Before you can start doing anything in the Arduino programmer, you must set the board-type and serialport.

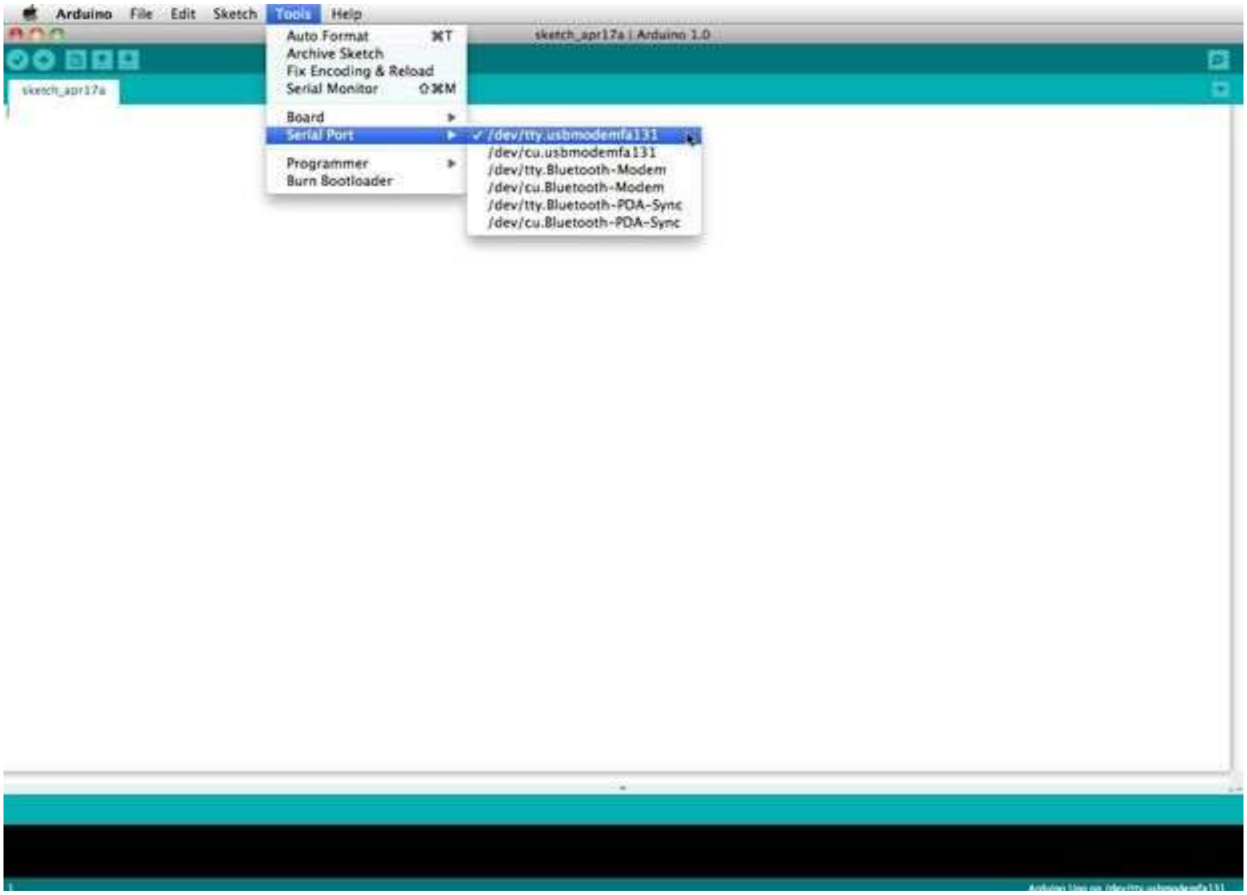
To set the board, go to the following:

Tools --> Boards

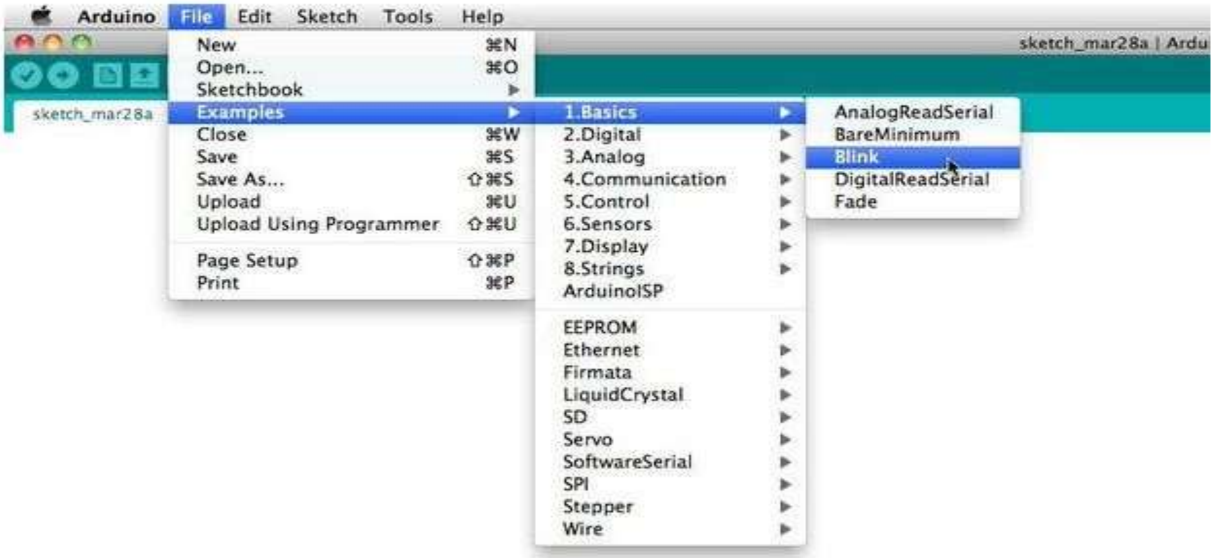
Select the version of board that you are using. Since I have an Arduino Uno plugged in, I obviously selected "Arduino Uno."

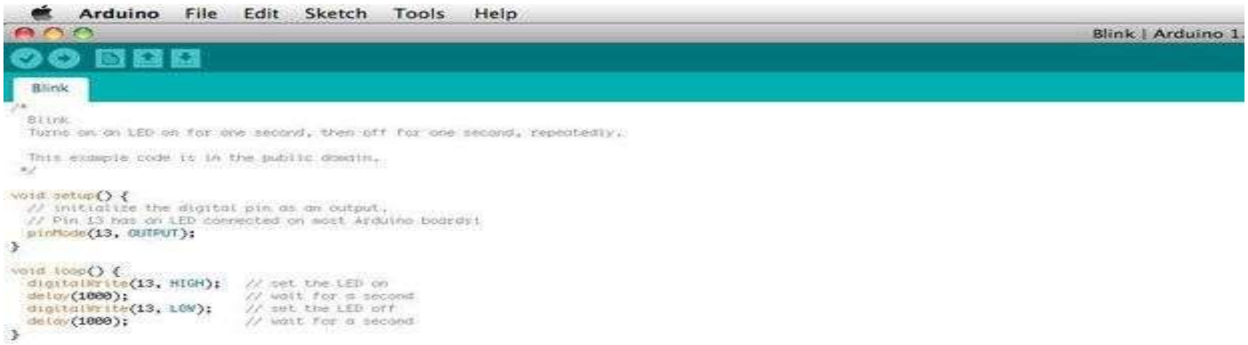
To set the serial port, go to the

following: Tools --> Serial Port



Arduino programs are called sketches. The Arduino programmer comes with a ton of example sketches preloaded. This is great because even if you have never programmed anything in your life, you can load one of these sketches and get the Arduino to do something.





```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  This example code is in the public domain.
 */

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH);   // set the LED on
  delay(1000);              // wait for a second
  digitalWrite(13, LOW);    // set the LED off
  delay(1000);              // wait for a second
}
```

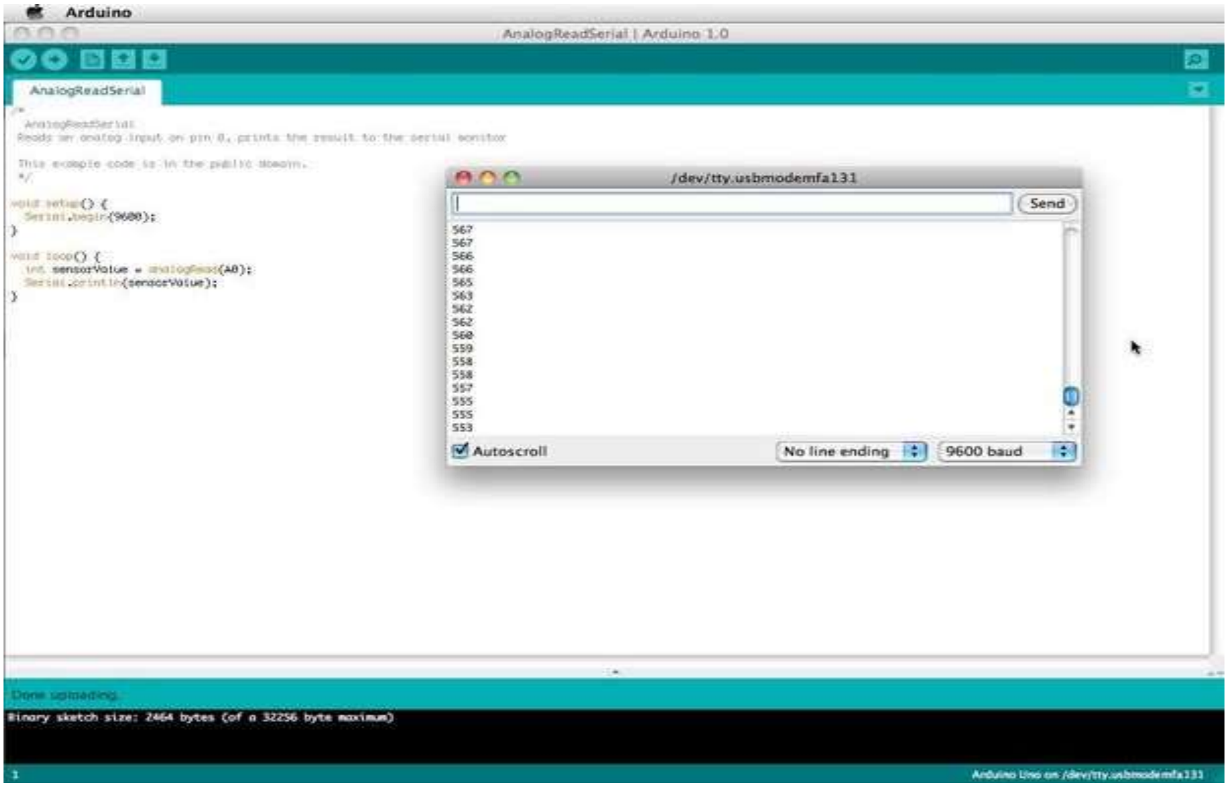
The serial monitor allows your computer to connect serially with the Arduino. This is important because it takes data that your Arduino is receiving from sensors and other devices and displays it in real-time on your computer. Having this ability is invaluable to debug your code and understand what number values the chip is actually receiving.

For instance, connect center sweep (middle pin) of a potentiometer to A0, and the outer pins, respectively, to 5v and ground. Next upload the sketch shown below:

File --> Examples --> 1.Basics --> Analog Read Serial

Click the button to engage the serial monitor which looks like a magnifying glass. You can now see the numbers being read by the analog pin in the serial monitor. When you turn the knob the numbers will increase and decrease.

The numbers will be between the range of 0 and 1023. The reason for this is that the analog pin is converting a voltage between 0 and 5V to a discreet number.



Arduino
AnalogReadSerial | Arduino 1.0
AnalogReadSerial
/dev/tty.usbmodemfa131
Send
Done uploading.
1
Arduino Uno on /dev/tty.usbmodemfa131

```
/*
  AnalogReadSerial
 Reads an analog input on pin 0, prints the result to the serial monitor

 This example code is in the public domain.
 */

void setup() {
  Serial.begin(9600);
}

void loop() {
  int sensorValue = analogRead(A0);
  Serial.println(sensorValue);
}
```

```
567
567
566
566
565
563
562
562
560
559
558
558
557
555
555
553
```

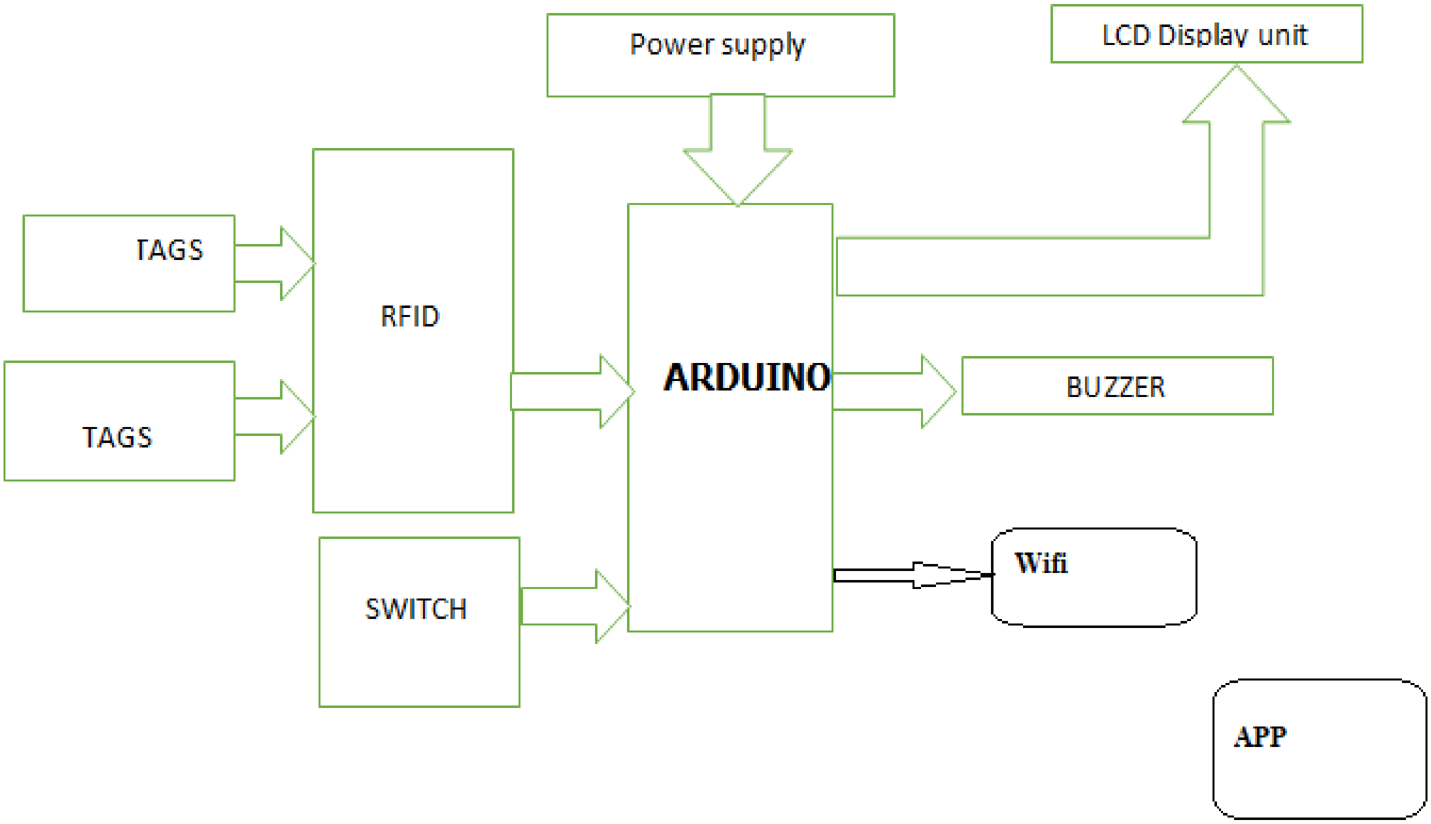
☑ Autoscroll   No line ending   9600 baud

```
Binary sketch size: 2464 bytes (of a 32256 byte maximum)
```

6. PROJECT DESCRIPTION

This chapter deals with working and circuits of “ “. It can be simply understood by its block diagram &circuit diagram.

6.1. BLOCK DIAGRAM:



2. SOFTWARE REQUIREMENTS:

- ➢ KIEL

3. HARDWARE REQUIREMENTS:

- ➢ Power supply
- ➢ ARDUINO
- ➢ SWITCH
- ➢ RFID
- ➢ LCD DISPLAY

REFERENCES:

[1] Hasan Al-Sakran “Agent And Radio FrequencyIdentification Based Architecture For Supermarket Information System”. Journal of Computer Science 9 (6): 699-707, 2013.

[2] Parvathi S J, Nagarathna “RFID Based Supermarket Queue Prevention and Automation System for Physically Challenged People” International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online) Vol. 3, Issue 2, pp: (835-840),
Month: April - June 2015

[3] Edmund Prater and Gregory V. Frazier “Future impacts of RFID on e-supply chains in grocery retailing” Supply Chain Management: An International Journal 10/2 (2005) 134–142 Emerald Group Publishing Limited [ISSN 1359-8546] [DOI 10.1108/13598540510589205]

[4] Zeeshan Ali1, Reena Sonkusare “RFID Based Smart Shopping and Billing” International Journal of Advanced Research in Computer and Communication Engineering Vol. 2, Issue 12,
December 2013

[5] P. Chandrasekar ; T. Sangeetha, “Smart shopping cart with automatic billing system through RFID and ZigBee”, Information Communication and Embedded Systems (ICICES), 2014 International Conference on 27-28 Feb. 2014.

[6] Bhagyashree Bhumkar , Tejasvini Changal , Bhagyashri Dahifale , Prof. Ganesh Deshmukh “International Journal Of Research In Science & Engineering

```
#include <LiquidCrystal.h>

#include <stdio.h>

#include <SoftwareSerial.h>


SoftwareSerial mySerial(8, 9);

LiquidCrystal lcd(6, 7, 5, 4, 3, 2);

int sw       = 10;

int buzzer  = 13;

 int amount=0;


int amount1=500,amount2=500;

int item1=0,item2=0,item3=0,item4=0;

int total=0;


 int sti=0;

 unsigned char rcv,count,gchr='x',gchr1='x',robos='s';

int ii=0,rchkr=0;

String inputString = "";         // a string to hold incoming data

boolean stringComplete = false; // whether the string is complete

void okcheck()

{

  unsigned char rcr;
  do{
      rcr = mySerial.read();
    }while(rcr != 'K');
}

void beep()
{
  digitalWrite(buzzer,LOW);delay(2500);digitalWrite(buzzer,HIGH);
}


void setup()
{
 Serial.begin(9600);serialEvent();
 mySerial.begin(9600);

 pinMode(sw, INPUT);
 pinMode(buzzer, OUTPUT);

 digitalWrite(sw, HIGH);
 digitalWrite(buzzer, HIGH);
```

```
lcd.begin(16, 2);lcd.cursor();

lcd.print("Billing System");


delay(2000);


mySerial.write("AT\r\n"); delay(2500);//                    okcheck();

mySerial.write("ATE0\r\n");                   okcheck();

mySerial.write("AT+CIPSERVER=1,23\r\n");delay(2500);//    okcheck();


lcd.clear();
lcd.print("Waiting For");

lcd.setCursor(0,1);

lcd.print(" Connection ");

do{

   rcv = mySerial.read();

  }while(rcv != 'C');


lcd.clear();lcd.print("Connected");
```

```
  delay(1500);

    lcd.clear();

}

//delay(1000);
//mySerial.write("AT+CIPSEND=0,10\r\n");delay(2000);
//mySerial.write("OTP:1234\r\n");delay(3000);

//5300AA40AE17
//5300AC02837E

void loop()
{
 mn1:lcd.clear();
 lcd.print("Swipe Card");

 if(stringComplete)
   {
    lcd.clear();lcd.print(inputString);
  delay(1500);

if(inputString == "5300ABFE6E68")

  {

   sts1++;

   if(sts1 == 1)

     {

      lcd.clear();        lcd.print("Soap-Add");

      lcd.setCursor(0,1); lcd.print("10Rs");

      item1=1;

      total = (total + 10);

     }

   if(sts1 == 2)

     {sts1=0;

      lcd.clear();        lcd.print("Soap-Delete");

      lcd.setCursor(0,1); lcd.print("10Rs");

      item1=0;

      total = (total - 10);

     }

     delay(2500);

  }

if(inputString == "11002A4C7F08")
```

```
  {
   sts2++;
   if(sts2 == 1)
    {
     lcd.clear();        lcd.print("Shampoo-Add ");
     lcd.setCursor(0,1); lcd.print("20Rs");
     item2=1;
     total = (total + 20);
    }
   if(sts2 == 2)
    {sts2=0;
     lcd.clear();        lcd.print("Shampoo-Delete");
     lcd.setCursor(0,1); lcd.print("20Rs");
     item2=0;
     total = (total - 20);
    }
    delay(2500);
  }
if(inputString == "5300AB7A9C1E")
  {
   sts3++;
   if(sts3 == 1)
    {
     lcd.clear();      lcd.print("Book-Add       ");
     lcd.setCursor(0,1); lcd.print("30Rs");
     item3=1;
     total = (total + 30);
    }
   if(sts3 == 2)
    {sts3=0;
     lcd.clear();      lcd.print("Book-Delete    ");
     lcd.setCursor(0,1); lcd.print("30Rs");
     item3=0;
     total = (total - 30);
    }
    delay(2500);
  }
//5300ABFC1D19
if(inputString == "5300AC01619F")
  {
   sts4++;
   if(sts4 == 1)
```

```
    {
     lcd.clear();        lcd.print("Pen-Add        ");
     lcd.setCursor(0,1); lcd.print("40Rs");

     item4=1;

     total = (total + 40);

    }

   if(sts4 == 2)

    {sts4=0;
     lcd.clear();        lcd.print("Pen-Delete     ");

     lcd.setCursor(0,1); lcd.print("40Rs");

     item4=0;

     total = (total - 40);

    }

    delay(2500);

  }

if(inputString == "28001164603D")

  {
    lcd.clear();lcd.print("Invalid");

    beep();



    mySerial.write("AT+CIPSEND=0,9\r\n");delay(2000);
```

```
      mySerial.write("Invalid\r\n");delay(3000);
    }
     inputString = "";
     stringComplete = false;
 }

if(digitalRead(sw) == LOW)
  {delay(500);
   while(digitalRead(sw) == LOW);
    lcd.clear();lcd.print("Total:");convertl(total);

       delay(1000);
    if(item1 == 1)
      {
        mySerial.write("AT+CIPSEND=0,15\r\n");delay(2000);
        mySerial.write("Soap-10Rs-Add\r\n");delay(3000);
      }

    if(item2 == 1)
      {
        mySerial.write("AT+CIPSEND=0,18\r\n");delay(2000);
      mySerial.write("Shampoo-20Rs-Add\r\n");delay(3000);
    }
  if(item3 == 1)
    {
      mySerial.write("AT+CIPSEND=0,15\r\n");delay(2000);
      mySerial.write("Book-30Rs-Add\r\n");delay(3000);
    }
  if(item4 == 1)
    {
      mySerial.write("AT+CIPSEND=0,14\r\n");delay(2000);
      mySerial.write("Pen-40Rs-Add\r\n");delay(3000);
    }

   mySerial.write("AT+CIPSEND=0,11\r\n");delay(2000);
   mySerial.write("Total:");converts1(total);
   mySerial.write("\r\n");delay(3000);


  total=0;
  item1=item2=item3=item4=0;
 }
delay(500);
```

```
}



void serialEvent()
{
  while (Serial.available())
        {
         char inChar = (char)Serial.read();
           //sti++;
           inputString += inChar;

          if(sti == 12)
            {sti=0;
              stringComplete = true;
            }
        }
}
```

```
int readSerial(char result[])
{
  int i = 0;
  while (1)
  {
    while (mySerial.available() > 0)
    {
      char inChar = mySerial.read();
      if (inChar == '\n')
         {
          result[i] = '\0';
          return 0;
         }
      if (inChar != '\r')
         {
          result[i] = inChar;
          i++;
         }
    }
  }
}


int readSerial1(char result[])
{
  int i = 0;
  while (1)
  {
    while (Serial.available() > 0)
    {
      char inChar = Serial.read();
      if (inChar == '*')
         {
          Serial.flush();
          return 0;
         }
      if (inChar != '*')
         {
          result[i] = inChar;
          i++;
         }
    }
  }
}



void gpsEvent()
{
  gpsString="";
  while(1)
  {
   //while (gps.available()>0)            //Serial incoming data from GPS

    while (mySerial.available() > 0)
   {
    //char inChar = (char)gps.read();
    char inChar = (char)mySerial.read();
```

```
     gpsString+= inChar;                    //store incoming data from GPS to temparary string
str[]

     i++;
    // Serial.print(inChar);
     if (i < 7)

     {
      if(gpsString[i-1] != test[i-1])        //check for right string
      {
        i=0;
        gpsString="";
      }
     }
    if(inChar=='\r')
    {
     if(i>60)
     {
       break;
     }
     else
     {
       i=0;
     }

    }

  }

   if(gps_status)

    break;

  }

}


void get_gps()

{



  lcd.clear();

  lcd.print("Getting GPS Data");

  lcd.setCursor(0,1);

  lcd.print("Please Wait.....");



   gps_status=0;

   int x=0;

   while(gps_status==0)

   {

    gpsEvent();
    int str_lenth=i;

    coordinate2dec();

    i=0;x=0;

    str_lenth=0;

   }

}




void coordinate2dec()

{

  String lat_degree="";

    for(i=19;i<=20;i++)

    lat_degree+=gpsString[i];



  String lat_minut="";

     for(i=20;i<=21;i++)

     lat_minut+=gpsString[i];

     for(i=23;i<=24;i++)

     lat_minut+=gpsString[i];
```

```
String log_degree="";
 for(i=31;i<=33;i++)
  log_degree+=gpsString[i];
String log_minut="";
 for(i=34;i<=35;i++)
 log_minut+=gpsString[i];
 for(i=37;i<=38;i++)
 log_minut+=gpsString[i];



 Speed="";
 for(i=42;i<45;i++)          //extract longitude from string
  Speed+=gpsString[i];


  float minut= lat_minut.toFloat();
  minut=minut/60;
  float degree=lat_degree.toFloat();
  latitude=degree+minut;


  minut= log_minut.toFloat();
```

```
    minut=minut/60;
    degree=log_degree.toFloat();
    logitude=degree+minut;
}


void gps_convert()
{
  if(gps_status)
  {
  Serial.println(gpsString);

  if(gpsString[0] == '$' && gpsString[1] == 'G' && gpsString[2] == 'P' && gpsString[3] == 'R' && gpsString[4] == 'M' && gpsString[5] == 'C')
    {
     //  Serial.println("Don11111111111111111111111111111111111111111111111111111111\r\n");
    // Serial.write(gpsString[18]);Serial.write(gpsString[19]);Serial.write(gpsString[20]);Serial.write(gpsString[21]);Serial.write(gpsString[22]);
     //lcd.setCursor(0,0);
     for(ii=0;ii<9;ii++)
       {
         //lcd.write(gpsString[19+ii]);

         msg1[ii] = gpsString[19+ii];

         //Serial.write(msg1[ii]);

        }

        //Serial.println("\r\n");

      //lcd.setCursor(0,1);

       for(ii=0;ii<10;ii++)

        {

         //lcd.write(gpsString[32+ii]);

         msg2[ii] = gpsString[32+ii];

        // Serial.write(msg2[ii]);

        }


// Serial.write("After ");

  //      converts(lati1);Serial.write("-");

    //   converts(longi1);Serial.write("\r\n");




              convlat(lati); convlong(longi);

         finallat[0] = msg1[0];

         finallat[1] = msg1[1];
         finallat[2] = '.';

         finallat[3] = flat[0]; finallat[4] = flat[1];finallat[5] = flat[2];finallat[6] =
flat[3];finallat[7] = '\0';



         finallong[0] = msg2[0];

         finallong[1] = msg2[1];

         finallong[2] = msg2[2];

         finallong[3] = '.';

         finallong[4] = flong[0];finallong[5] = flong[1];finallong[6] = flong[2];finallong[7] =
flong[3];finallong[8] = '\0';



    }

  }

}


 void convlat(unsigned int value)

       {

             unsigned int a,b,c,d,e,f,g,h;


       a=value/10000;
       b=value%10000;
       c=b/1000;
       d=b%1000;
       e=d/100;
       f=d%100;
       g=f/10;
       h=f%10;



       a=a|0x30;
       c=c|0x30;
    e=e|0x30;
    g=g|0x30;
    h=h|0x30;


  // dlcd(a);
//   dlcd(c);dlcd(e); dlcd(g);dlcd(h);//lcddata('A');//lcddata(' ');lcddata(' ');



            flat[0] = c;
            flat[1] = e;
    flat[2] = g;

  flat[3] = h;




}
```