

Design & Development of a Microcontroller Based on measuring Real Time Data and Interface with Mobile through Bluetooth



Developed by:

- 1. Shah Md. Manjir Hosain**
ID: 2012-1-55-055
- 2. Kaniz Fatema**
ID: 2011-2-55-023

Project Supervisor:

Dr. Md. Habibur Rahman
Professor
Dept. of Applied Physics Electronics & Electrical Engineering
University Of Dhaka

DECLARATION

We hereby declare that we carried out the work reported in this project in the Department of **Electronics and Communication Engineering, East West University**, under the supervision of **Dr. Md. Habibur Rahman**. We also declare that no part of this work has been submitted elsewhere partially or fully for the award of any other degree or diploma. Any material reproduced in this project has been properly acknowledged. All sources of knowledge used have been duly acknowledged.

Signature:

.....

Kaniz Fatema
ID: 2011-2-55-023

.....

Shah Md. Manjir Hosain
ID: 2012-1-55-055

Approval

The Project titled as '*Design & Developed of a Microcontroller Based on measuring Real Time Data and Interface with Mobile through Bluetooth*' has been submitted to the following respected members of the Board of Examiners of the Faculty of Engineering for partial fulfillment of the requirements for the degree of Bachelor of Science in Electronics & Telecommunications Engineering by the following students and has been accepted as satisfactory.

Submitted By:

Shah Md. Manjir Hosain
ID: 2012-1-55-055

Kaniz Fatema
ID: 2011-2-55-023

Dr. Md. Habibur Rahman

Dr. Gurudas Mandal

Professor, Electrical and Electronics
Engineering
Professor,
University Of Dhaka

Chairperson & Associate
Dept. of Electronics and Communication

ACKNOWLEDGEMENTS

Many people deserve our thanks for their help in completing this project. We would like to thank our department for giving us this chance to do this project. We want to express our thanks and deep appreciation to our advisor **Dr. Md. Habibur Rahman** as he has exhausted all his knowledge and time by following up our daily progress and encouraging advices and even by sharing on the troubles. We would like to extend our thanks to the laboratory staff of ECE Dept. for their fast response and cooperation with us to get some materials we need for our case. We have special acknowledgment for our group members for their understanding each other and hard working from the beginning up to the end. Finally, we would like to thank all persons who were involved with this project for their valuable help and professionalism during this project.

Abstraction

We know smart phone is a big blessing in our daily life. Through the smart phone we can do several things. That simplifies our work. We can also use it as technology purposes. In this project, we have designed and developed an Arduino based measuring system. Here we use real time clock, Arduino uno, sensor, Bluetooth, signal conditioning circuit, and Android mobile. The system can measure radiation, time and temperature. Arduino read data from radiation sensor and save data in EEPROM. When we need those data then we paired our Android mobile with Bluetooth. Consequently we can read data in our mobile phone. Here, an Arduino board has been used as the main controlling unit, HC-05 module as the Bluetooth sensor. A program has been developed and burnt into the microcontroller of the Arduino module. The system is constructed and tested and found to work satisfactorily.

TABLE OF CONTENTS

Short Contents Page			PAGE
TITLE			
1			
DECLARATION			
2			
APPROVAL			
3			
ACKNOWLEDGEMENT			
4			
ABSTRACT			
5			
TABLE	OF		CONTENTS
6			
LIST	OF		FIGURE
8			
LIST	OF		TABLE
8			

CHAPTER 1 INTRODUCTION

- 1.1 Energy
- 9
- 1.2 Solar Energy

9
1.3 Solar radiation
10
1.4 Measurement of solar Radiation
11
1.5 Real Time
11
1.6 Real Time Measurement
13
1.7 Purpose of the project
14
1.8 Outline of project
14

Chapter 2

AN OVERVIEW OF MICROCONTROLLER

2.1 Introduction	
15	
2.2 Microcontroller	15
2.21 Advantages of Microcontroller	
15	
2.3 Arduino-UNO	
16	
2.4 History of the Arduino-UNO	16
2.5 Development	17
2.6 Evaluation	17

Chapter: 3

Theory Behind The Project

3.1 Introduction	19
3.2 Architecture of Arduino UNO	
19	
3.21 power	20
3.22 Pin	20
3.23 Main IC	20
3.24 Voltage Regulator	20
3.25 TX RX LED s	21
3.3 Arduino software	21
3.4 The Integrated Development Environment	

21	
3.5	IDE parts 22
3.6	HC-05 Bluetooth Modules 23
3.7	Pins 23
3.8	Hardware Feature 24
3.9	Software Feature 25

Chapter 4 Hardware Design of The System

4.1	Block Diagram 26
4.2	RTC 27
4.3	Arduino 28
4.4	ADC 29
4.41	ADC Types 30
4.5	Sensor 30
4.6	Flowchart of Program
31	
4.7	Reading Potentiometer
32	
4.8	Connection of Arduino with Bluetooth Device
33	
4.9	Diagram of whole project
34	

Chapter 5 Performance study of the Developed system

5.1	Introduction 35
5.2	Connection between Arduino and Real Time Clock
35	
5.3	Connection between Bluetooth and Arduino
36	
5.4	Developed of whole System
37	
5.5	The output of the project
38	
5.6	Product commercialization
40	

Chapter 6 Conclusion

6.1 Conclusion	41
6.2 Future work Scope	41

Appendix

Programming code	42
References	45

LIST OF FIGURE

1. Figure1.3: The Picture of the solar Radiation	10
2. Figure2.3: Arduino uno	16
3. Figure2.6: schemetic Diagram of Arduino uno	18
4. Figure3.2: Arduino uno R3	19
5. Figure3.7: Bluetooth Device	24
6. Figure3.9: Schematic Diagram of Bluetooth	25
7. Figure4.1: Block Diagram of The Project	26
8. Figure4.6: Flowchart of Whole System	30
9. Figure4.7: Connection Between Arduino and potentiometer	31
10. Figure4.8: Connection between Arduino and Bluetooth	32
11. Figure4.9: Schemetic Diagram of Whole system	33
12. Figure5.2: Connection between Arduino and RCT	34
13. Figure5.3: Arduino connected with Bluetooth	35
14. Figure5.4: The Whole connection of the project	36
15. Figure5.5: Result showed by Android mobile	37
16. Figure5.51: Radiation value at=3.3volt	38
17. Figure5.52: Radiation value at=5vol t	39

LIST OF TABLE

1. Table5.2 Module pinout	35
---------------------------	----

Chapter 1

INTRODUCTION

1.1 Energy:

Thomas young was possibly the first to use the term "Energy". The word energy derives from the ancient Greek *energeia* "activity operation". Normally it can be defined as a system possesses energy it has the ability to do work. It shifts energy from one system to another. In physics energy is a property of objects or converted into different form but cannot be created or destroyed. All forms of energy are either kinetic or potential. If the energy of an object that possesses due to its motion then it is called a kinetic energy. The potential energy is defined as that energy which an object has because of its position. In SI units, energy is measure in joules. The law of conservation of energy states that the total energy of an isolated system remains constant-that is said to be conserved over time. Energy cannot be neither created nor be destroyed, but it transforms one from to another. Fossil Fuels are limited - Although fossil fuel reserves are expected to run dry within the next century, solar power is clean, abundant, and will remain a renewable resource that can meet all of Earth's energy needs for *billions* of years to come.

1.2 Solar Energy:

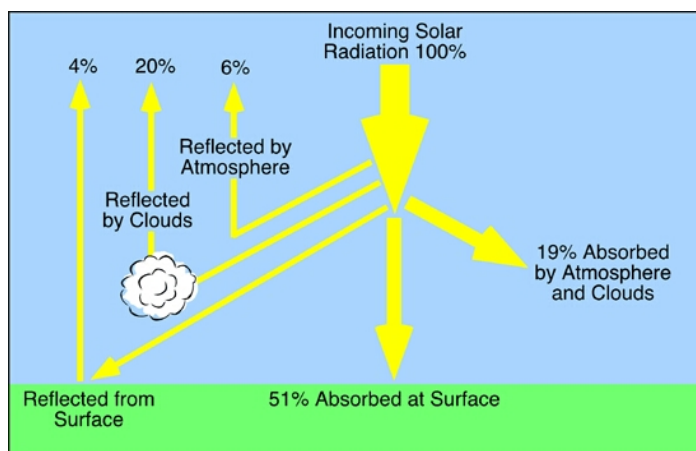
Solar energy is the energy that comes to the Earth from the Sun. It is commonly known as sunlight. This energy arrives in the form of electromagnetic radiation, which is energy that travels in waves. A variety of modern technologies are able to capture and convert solar energy into other useful forms of energy, such as heat and electricity. The Sun is a gigantic nuclear fusion reactor with 330,000 times more mass than Earth. It is by far Earth's most available energy resource. If more solar energy could be harnessed properly, it could provide many times today's global demand for energy. It is an important source of renewable energy and its technologies are broadly characterized as either passive solar or active solar depending on the way they capture and distribute solar energy or convert it into solar power. Active solar techniques include the use of photovoltaic systems, concentrated solar power and solar water heating to harness the energy. Passive solar techniques include orienting a building to the Sun, selecting materials with favorable thermal mass or light dispersing properties, and designing spaces that naturally circulate air.

1.3 Solar radiation:

Solar radiation is the radiation, or energy we get from the sun. It is also known as short-wave radiation. Solar radiation comes in many forms, such as visible light, radio waves, heat (infrared), x-rays, and ultraviolet rays. Measurements for solar radiation are higher on clear, sunny day and usually low on cloudy days. When the sun is down, or there are heavy clouds blocking the sun, solar radiation is measured at zero.

Solar radiation is also define as the light and energy that comes from the sun, and there are many different forms. The **electromagnetic spectrum** explains the different types of light waves that are emitted from the sun. Light waves are similar to waves we see on the ocean - they move up and down and travel from one place to another. The difference is that instead of the water vibrating up and down, **light waves** are vibrations of electromagnetic fields, hence the name the electromagnetic spectrum. Solar radiation depends on two frequincies,one is 'High frequency' and is 'Low frequency'.

Low frequency waves are low-energy waves with a long wavelength. The length of the wave itself is very long for a given period of time. These are things like radar, TV and radio waves. **High frequency waves** are high-energy waves with a short wavelength. This means that the length of the wave itself is very short for a given period of time. These are things like gamma rays, X-rays and ultraviolet rays. The atmosphere affects the amount of solar radiation received. When solar radiation travels through the atmosphere, some of it is absorbed by the atmosphere (16%). Some of it is scattered to space (6%). Some of it is reflected by clouds (28%). About 47% of it reaches the Earth's surface.



1.3 Figure: The picture of solar radiation.

1.4 Measurement of solar radiation:

In PV system design it is essential to know the amount of sunlight available at a particular location at a given time. The two common methods which characterise solar radiation are the solar radiance (or radiation) and solar insolation. The solar radiance is an instantaneous power density in units of kW/m². The solar radiance varies throughout the day from 0 kW/m² at night to a maximum of about 1 kW/m². The solar radiance is strongly dependant on location and local weather. Solar radiance measurements consist of global and/or direct radiation measurements taken periodically throughout the day. The measurements are taken using either a pyranometer (measuring global radiation) and/or a pyrheliometer (measuring direct radiation). In well established locations, this data has been collected for more than twenty years.

A **pyranometer** is a type of actinometer used to measure broadband solar irradiance on a planar surface and is a sensor that is designed to measure the solar radiation flux density (W/m²) from a field of view of 180 degrees. The name pyranometer stems from Greek, "pyr - πρ" meaning "fire" and "ano - vω" meaning "above, sky".

Solar radiation for a particular location can be given in several ways including:

- Typical mean year data for a particular location
- Average daily, monthly or yearly solar insolation for a given location
- Global is flux contours either for a full year, a quarter year or a particular month
- Sunshine hours data
- Solar Isolation Based on Satellite Cloud-Cover Data
- Calculations of Solar Radiation

1.5 Real Time:

Real-time is a term often used to distinguish reporting, depicting, or reacting to events at the same rate and sometimes at the same time as they unfold rather than compressing a depiction or delaying a report or action. The term has uses in many areas, and many articles cover different aspects.

Real-time is a more flexible word. If there is a network work with a speed 1 kbit/s and the designed system can give an output by speed more than 1 kbit/s, it will be a real-time system with respect to that network. So, it depend on the speed of the network and there aren't a limited speed of execution to be a standard for real time that can

according to it, can be decided that the system is a real-time or not.

Real time is the actual time during which a process or event occurs. *Real time* is also a level of computer responsiveness that a user senses as sufficiently immediate or that enables the computer to keep up with some external process. For example, to present visualizations of the weather as it constantly changes. *Real-time* is an adjective pertaining to computers or processes that operate in real time. Real time describes a human rather than a machine sense of time. Real-time responses are often understood to be in the order of milliseconds, and sometimes microseconds. A system not specified as operating in real time cannot usually *guarantee* a response within any timeframe, although *actual* or *expected* response times may be given. A real-time system has been described as one which "controls an environment by receiving data, processing them, and returning the results sufficiently quickly to affect the environment at that time." The term "real-time" is also used in simulation to mean that the simulation's clock runs at the same speed as a real clock, and in process control and enterprise systems to mean "without significant delay". Real-time software may use one or more of the following: synchronous programming languages, real-time operating systems, and real-time networks, each of which provide essential frameworks on which to build a real-time software application. Systems used for many mission critical applications must be real-time, such as for control of fly-by-wire aircraft, or anti-lock brakes on a vehicle, which must produce maximum deceleration but intermittently stop braking to prevent skidding. Real-time processing *fails* if not completed within a specified deadline relative to an event; deadlines must always be met, regardless of system load.

Computing:

- **Real-time computing:** subject to a real-time constraint
- **Near real-time**
- **Real-time clock:** a clock as normally understood, as distinct from a computer clocking oscillator
- **Real-time Control System (RCS):** subject to a real-time constraint

- **Real-time Programming Language (RPL):** a compiled database programming language which expresses work to be done by a particular time

1.6 Applications of real-time computing

- Real-time computer graphics
- Real-time camera system, for controlling cameras in a 3D virtual environment
- Real-time operating system, for running real-time software
- Real-time protection, protection enabled constantly, rather than by, say, a virus scan
- Real-time text, transmitted as it is being typed or produced
- Real time Java, for real-time programs in Java
- Real time disk encryption, encrypting data as it is written to disk
- Real-time web, whereby information is sent to users as it becomes available etc.

1.7 purpose of the project:

In this project we draw a simple circuit to measure a real time data and interface with mobile through Bluetooth. Now a day's mobile is an important part of our life. If it's a android mobile then it is more effective in our daily life. Such as, Arduino based appliance control using smart phone. In this project, we describe how to control and designing a circuit that measure real time data of solar radiation and this radiation save in eeprom and finally send data through Bluetooth to the android mobile appliance control system. Then we can easily measure any data from anywhere but the distance depends on the power range of Bluetooth.

- To design a smart experimental appliance control system.
- To construct a smart instrument controller circuit in the breadboard.
- Test for its functionality.
- Product commercialization.
- To design the control system with low cost components.

1.8 Outline of the report:

Our project is developed by microcontroller to easy our work. we know solar energy is a reliable energy. it is a big issue in our next generation to create something instead of fossil fuel. That's why solar energy is going popular to the user. Solar energy is reliable energy but its implementation cost is so high. Development of reliable energy procedure is high technology. To reduce our experimental cost we use AlterNet way. Hare we use microcontroller device is called Arduino Uno board. It is a open source device. We can use it easily to solve any controlling problem. Here we also need frequent measure of radiation. But it is difficult to collect data. To using this controlling system we can easily collect data from anywhere through Bluetooth. Finally this data show in android mobile.

Chapter 2

An overview of Microcontrollers

2.1 Introductions:

Arduino is a simple microcontroller board and open source development environment that allows us to make computers that drive both functional and creative projects alike. In the project we use Arduino-UNO board. The Arduino platform has power and reset circuitry set up as well as circuitry to program and communicates with the microcontroller over USB. These systems provide sets of digital and analog I/O pins that can be interfaced to various expansion boards and other circuits. On the software side, Arduino provides a number of libraries to make the programming easier. The simplest of these are functions to control and read the I/O pins over the fiddle with the bit masks normally use to interface with mega I/O.

2.2 Microcontroller:

A microcontroller is a computer present in a single integrated circuit which is dedicated to perform one task and execute one specific application. It contains memory, programmable input/output peripherals as well a processor. Microcontrollers are mostly designed for embedded applications and are heavily used in automatically controlled electronic devices such as cell phones, cameras, microwave ovens, washing machines, etc.

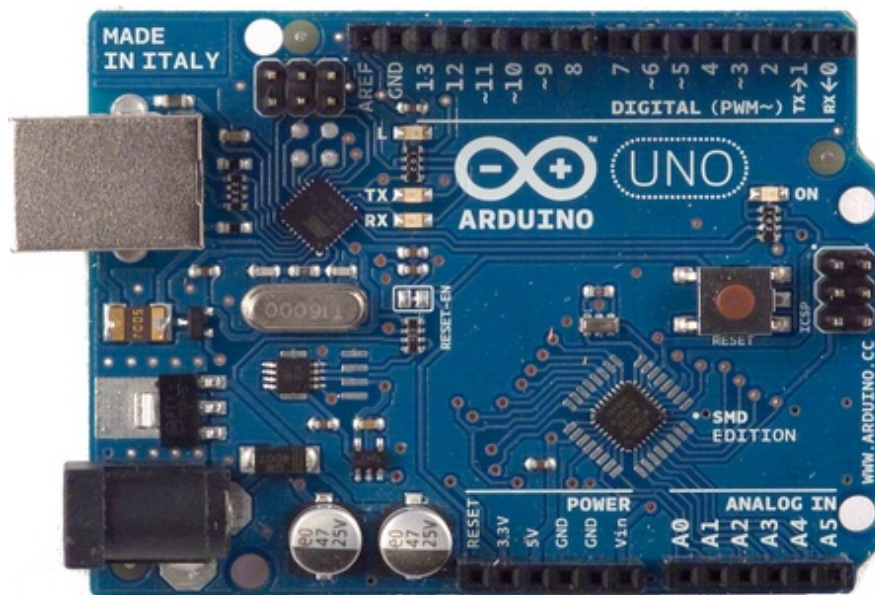
2.21 Advantages of Microcontroller:

1. Flexibility Microcontroller are special types of processor chips that are very small and somewhat flexible due to their programmable nature.
2. Faster speed of execution since microcontroller is fully integrated inside the processor. These devices operate at faster speeds to execute instruction compared to general purpose microcontroller.
3. Inexpensive as microcontroller are fully integrated one chip these device are cheap to manufacture.
4. Many tasks can be performed by microcontrollers respectively and human efforts can be save.
5. Rigid once microcontroller is programme, typically they can not

reprogrammed, if microcontroller are controlled **by RAM and ROM.**

2.3 Arduino UNO:

Arduino is an open-source platform used for building electronics projects. Arduino consists of both a physical programmable circuit board and a piece of software or IDE (Integrated Development Environment) that runs on your computer, used to write and upload computer code to the physical board.



2.3 Figure: Arduino UNO

2.4 History of the arduino uno:

The first Arduino was introduced in 2005, aiming to provide an inexpensive and easy way for novices and professionals to create devices that interact with their environment using sensor and actuator. Common examples of such devices intended for beginner hobbyists include simple robots, thermostats, and motion detectors. From the beginning, the Arduino board was developed to attract artists and designers. The winning microcontroller was created by Hernando Barragan to be used for parsing data to the electronic devices. His aim was that it could be used by non-technical people who only had basic experience with using computers. He first of all wanted it to be used as a prototype tool. Since he needed help to create an easy software tool to program the board, he engaged Casey Reas and Massimo Banzi as his assistants. Reas created the visual programming

language for prototyping tool.

2.5 Development:

The computer, commonly defined as a tool for processing, storing, and displaying information, arose from a long line of analog devices used for effective counting and calculation, ranging from the simple abacus (first invented in Sumeria around 2300 BC), to Napier's Bones (conceived in 1617, and the precursor to the slide rule), to Blaise Pascal's gear-based mechanical calculator (1645). The development of the computer accelerated during the 1940's, spurred on largely by the highly industrialized nature of military production in World War II. The 1960's marked a significant evolutionary leap for computing, due to the development of solid state computers (such as the IBM 1401), which used transistors for processing operations, and magnetic core memory for storage. The invention of integrated circuits in 1959 by Jack Kilby, which enabled transistors and circuits to be fused onto small chips of semiconducting materials (such as silicon), allowed further miniaturization of computer components. Another important development during this decade was the advent of high-level computer programming languages that were written in symbolic language, making computer code somewhat easier to read and learn than previous machine languages. COBOL and FORTRAN were the main languages introduced during this period. The microprocessor was introduced in 1970. The microprocessor essentially miniaturized all hardware components of a computer's central processing unit to fit onto a single, tiny integrated circuit, now more popularly known as a microchip. The microchip also became the main driving component of microcontrollers (such as the Arduino), which generally consist of a microchip, memory storage hardware, and input/output hardware for sensors. The 1970's and 1980's also saw the development of a new generation of more powerful programming languages (such as C, C++, and later Java) for applications in business and science.

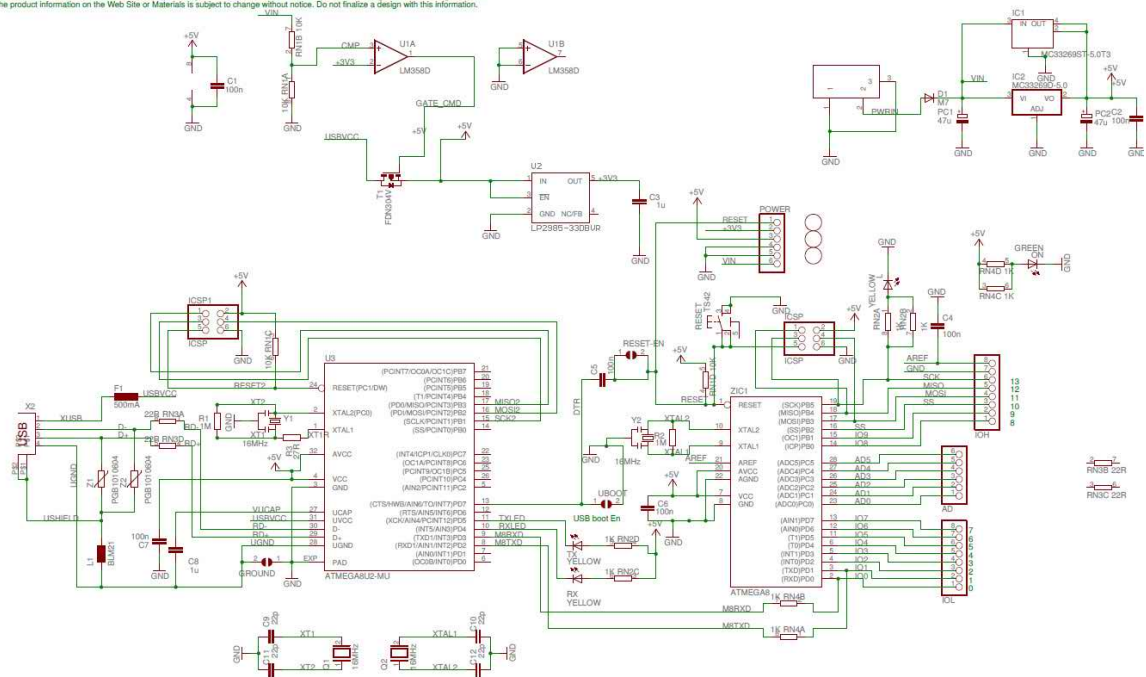
2.6 Evaluation:

The PIC microcontroller board, introduced in 1985 by General Instruments, became one of the most popular tools for electronics enthusiasts (before the Arduino) for several reasons. Other popular boards for hobbyists include the BASIC Stamp (Parallax Inc., 1990), and Wiring both of which share the benefits of simplicity of programming, and a resulting ease of rapid-prototyping. The Arduino project grew largely out of the "DIY"

climate created by the burgeoning popularity of rapid- prototyping boards like PIC. In fact, the immediate precursor to the Arduino was a custom made Wiring microcontroller created by the artist/ designer Hernando Barragan in 2004 for his. In 2005, the Arduino team was formed in Ivrea, Italy, consisting of Barragan, Massim Banzi, David Cuartielles, Dave Mellis, Gianluca Marino, and Nicholas Zambetti. As a result, the Arduino incorporated the following characteristic, a programming environment based on processing language (a programming language conceived by Ben Fry and Casey Reas, also conceived for artists/ designers), the ability to program the board via a standard USB connection, and a low price point. The Arduino achieved rapid success even within its first two years of existence, selling in a quantity of more than 50,000 boards. By 2009, it had spawned over 13 different incarnations, each specialized for different applications. Today, the Arduino microcontroller has become one of the most popular prototyping platforms in the world, and is a prime example of how hardware and software technologies originally created for military, business, and scientific applications can be repurposed to serve the needs of individuals creating projects in the realms of new media art and design.

Arduino™ UNO Reference Design

Reference Designs ARE PROVIDED "AS IS" AND "WITH ALL FAULTS". Arduino DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED. REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Arduino may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "unified". Arduino reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this information.



2.6:Figure : Schematic Diagram of Arduino uno

Chapter 3

Theory Behind the project

3.1 Introduction:

The project include with Arduino UNO board, PC, Power supply, Bluetooth, Real Time clock, potentiometer. We want to connect the Bluetooth device to an Arduino .We developed the connection by using upper equipments. We developed the connection by using upper equipments. At first we developed the connection with real time clock. Then we developed the circuit with conditional circuit and Bluetooth. Finally We have got success in this project

3.2 Architecture of Arduino UNO:

Arduino is an open-source microcontroller platform with development environment that implements Processing/Wiring language which is a subset of C and includes several libraries. The Arduino Uno R3 is a microcontroller board based on the ATmega328 (AVR architecture). It has 20 digital input/output pins where 6 can be used as PWM outputs and 6 can be used as analog inputs. It can be powered from USB port or from external DC power supply. Integrated USB connection can be used for simple FLASH ROM programming or for communication with other devices. Included library allows serial communication on any of the Uno's digital pins.



3.2 Figure: Arduino uno R3

3.21 Power:

The Arduino/Genuino Uno board can be powered via the USB connection or with an external power supply. The power source is selected automatically. 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 USB connection is also how you will load code onto your Arduino board.

3.22 Pins (5V, 3.3V, GND, Analog, Digital, PWM, AREF)

The pins of Arduino are the places where connect wires to construct a circuit. The Arduino has several different kinds of pins, each of which is labeled on the board and used for different functions.

5V: The 5V pin supplies 5 volts of power. Most of the simple components used with the Arduino run happily off of 5 or 3.3 volts.

GND: Full name is Ground. There are several GND pins on the Arduino, any of which can be used to ground circuit.

Analog: The area of pins under the 'Analog In' label (A0 through A5 on the UNO) is

Analog In pins. These pins can read the signal from an analog sensor and convert it into a digital value that we can read.

Digital: Across from the analog pins are the digital pins (0 through 13 on the UNO).

These pins can be used for both digital input and digital output (like powering an LED).

PWM: The digital pins (3, 5, 6, 9, 10, and 11) on the UNO are the PWM(~) pins. These pins act as normal digital pins, but can also be used for something called Pulse-Width Modulation (PWM).

AREF: Stands for Analog Reference. It is sometimes used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

3.23 Main IC

The black thing with all the metal legs is an IC, or Integrated Circuit. The main IC on the Arduino is slightly different from board type to board type, but is usually from the ATmega family of IC's from the ATMEL Company. This can be important, as you may need to know the IC type before loading up a new program from the Arduino software. This information can usually be found in writing on the top side of the IC.

3.24 Voltage Regulator

The voltage regulator is not actually something interacting with on the Arduino. But it is potentially useful to know that it is there and what it's for. It controls the amount of voltage that is let into the Arduino board. It will turn away an extra voltage that might harm the circuit.

3.25 TX RX LEDs

TX is short for transmit, RX is short for receive. In our case, there are two places on the Arduino UNO where TX and RX appear once by digital pins 0 and 1, and a second time next to the TX and RX indicator LEDs. These LEDs will give us some nice visual indications whenever Arduino is receiving or transmitting data.

3.3 Arduino software:

We get Arduino software from the Arduino web site *like* `arduino.cc`. At first we download and install software for Mac, Linux, or Windows from `arduino.cc`. Any Windows user is needed to install a driver. Then connect our board with USB. Run the Arduino application and select the Arduino UNO from the tool to board menu. Then open the sketch file. In sketch file there are we can see many examples like blink. The blink is a primary program of a microcontroller. Click the toolbar button to upload blink in our board.

3.4: The Integrated Development Environment:

An integrated development environment (IDE) is a software application that provides comprehensive facilities to computer programmers for software development. Download and install

Arduino(<http://www.processing.org>). Connect the board to the USB port on our computer. Then Arduino IDE run. Set the correct port our devices using

- Tools->serial port->select port beginning with `/dev/cu.usbserial-`

Set the Arduino board type we are using

- Tools->Board->select correct board type (printed on the chip)

Open :File->Examples->Digital->Blink

Before uploading the code, we can verify it (sketch->verify/compile), however this is not compulsory, as if there are errors while sending program to the board, Arduino will abort the operation and print appropriate warning/exceptions.

Connect our Led to the port 13 and **GND** (ground). **LDE** is directional components. That means if we put it in backwards it will not work. To help we put the **LED** in right, the led factory cut the leg different lengths. the longer leg goes to the hole marked 13 and the shorter one goes in the hole marked **GND**. Make sure our board is connected via USB cable. Then click the upload button or select **File->Upload To I/O Board**. Then we watch for any errors(if wrong board/port selected or bug in the code).if it works-**Let there be light**

An IDE normally consists of a source code editor, build automation tools and a debugger. Most modern IDEs have intelligent code completion. Every microcontroller needs software to be programmed. the Arduino software has own integrated development environment. It is free and everyone can download from its official website using either the Windows, Mac OSX or Linux platform. That allows Arduino board to gain more users and it helps to grow.

3.5 IDE Parts:

- **Compile:** code can be send to the before program. It converts instruction that the board understands. This process is called compiling.
- **Stops:** This stops the compilation process.
- **Create new sketch:** This opens a new window to create new sketch.
- **Open Existing Sketch:** This loads a sketch from a file on our computer.
- **Save Sketch:** This saves the change to the sketch.
- **Upload to Board:** This compiles and transmits over the USB cable to our board.
- **Serial monitor:** When our programs didn't work to reach the point, we just pulled out our hair and tried harder.
- **Tab Button:** This button helps to create new files in our sketch. This is for more advanced programming than we will do in this class.

- **Sketch Editor:** This is where write or edit sketch.
- **Text console:** This show we what the IDE is currently doing and is also where error messages display if make a mistake in typing program.
- **Line Number:** This shows what line number our cursor is on.

3.6 HC-05 Bluetooth Modules

HC-05 Bluetooth Module is one of the most popular Bluetooth module used in embedded projects. It can be easily interfaced with Arduino Board, Raspberry Pi, and Microcontrollers through serial UART interface. HC-05 module is an easy to use Blue-tooth SPP (Serial Port Protocol) module, designed for transparent wireless serial connection setup. Serial port Bluetooth module is fully qualified Bluetooth V2.0+EDR (Enhanced Data Rate) 3Mbps Modulation with complete 2.4GHz radio transceiver and baseband. It uses CSR Blue core 04-External single chip Bluetooth system with CMOS technology and with AFH (Adaptive Frequency Hopping Feature). It has the footprint as small as 12.7mmx27mm. Hope it will simplify your overall design/development cycle.

Specifications

- Frequency : 2.4GHz ISM Band
- Bluetooth Protocol : Bluetooth Specification v2.0 + EDR
- Emission Power : $\leq 4\text{dBm}$, Class 2
- Modulation : GFSK (Gaussian Frequency Shift Keying)
- Operating Voltage : 3.3V
- Input Power Supply : 3.6V ~ 6V
- Security : Authentication & Encryption
- Working Temperature : $-20^{\circ}\text{C} \sim +75^{\circ}\text{C}$
- Dimensions : 35.7mm x 15.2mm x 5.6mm

3.7 Pins

On the bottom side there are labels for the signal direction and voltage levels:

- **KEY:** According to the data sheet, I need to pull-up this pin while

power-on-reset of the module to enforce AT mode. I have not been able to verify this yet. I have been told that some modules have this pin not connected at all?

- **VCC:** Is indicated in the range of 3.6V-6V. The module worked for me both with 3.3V and 5V

- **GND:** Ground

- **TXD:** Serial *output* of the module, to be connected to RX of the microcontroller. Note that this signal is using 3.3V logic level

- **RXD:** Serial *input* of the module, to be connected to the TX of the microcontroller. Note that this signal is using 3.3V logic levels

- **STATE** Connected to LED2 (Pin32) of the module, but no meaning? At least on my module the pin was always low, regardless if paired or not.



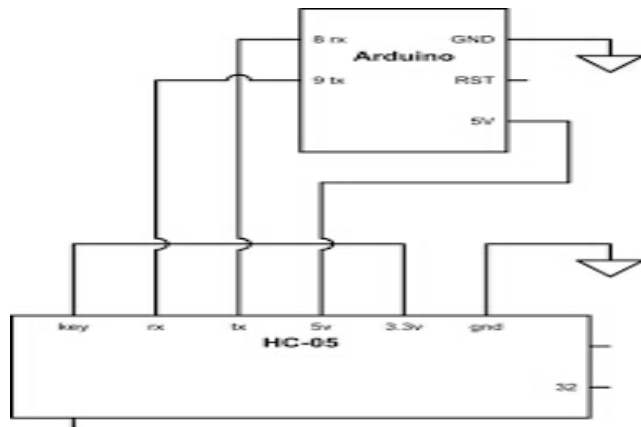
3.7 Figure: Bluetooth device

3.8 Hardware Features

- Typical -80dBm sensitivity
- Up to +4dBm RF transmit power
- Low Power 1.8V Operation ,1.8 to 3.6V I/O
- PIO control
- UART interface with programmable baud rate
- With integrated antenna

3.9 Software Feature:

- Default Baud rate: 38400, Data bits:8, Stop bit:1,Parity:No parity, Data control: has.
- Supported baud rate: 9600,19200,38400,57600,115200,230400,460800.
- Given a rising pulse in PIO0, device will be disconnected.
- Status instruction port PIO1: low-disconnected, high-connected;
- PIO10 and PIO11 can be connected to red and blue led separately. When master and slave are paired, red and blue led blinks 1time/2s in interval, while disconnected only blue led blinks 2times/s.
- Auto-connect to the last device on power as default.
- Permit pairing device to connect as default.
- Auto-pairing PINCODE:"0000" as default
- Auto-reconnect in 30 min when disconnected as a result of beyond the range of connection.



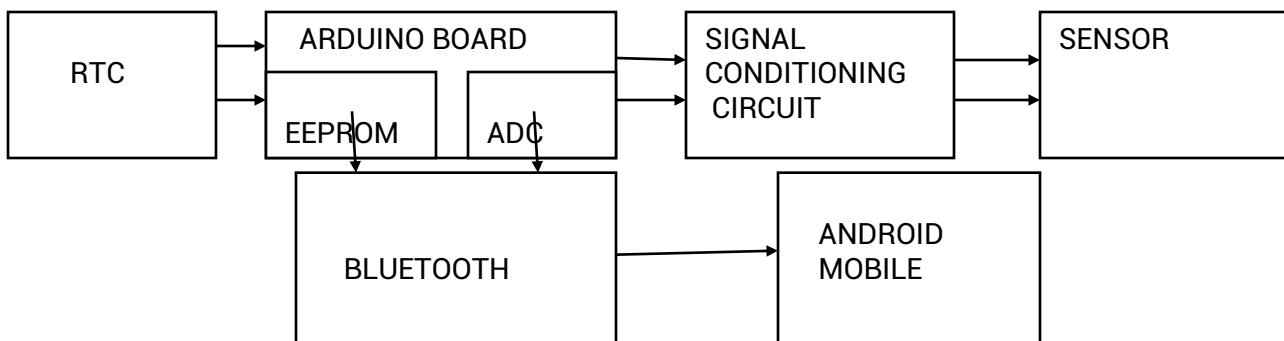
3.9 FIGURE: Schematic Diagram of Bluetooth

Chapter 4

Hardware Design of the system

4.1 Block diagram:

The whole system consists of real time clock, arduino board works as microcontroller, signal conditioning circuit, bluetooth, and android mobile.



4.1 FIGURE: Block Diagram of the Project

4.2 RTC(Real time clock):

A **real-time clock (RTC)** is a computer clock that keeps track of the current time. Although the term often refers to the devices in personal computers, servers and embedded systems, RTCs are present in almost any electronic device which needs to keep accurate time. A real time clock is basically just like a watch - it runs on a battery and keeps time for our even when there is a power outage! Using an RTC, we can keep track of long timelines, even if you reprogram your microcontroller or disconnect it from USB or a power plug. Most microcontrollers, including the Arduino, have a built-in timekeeper called **millis()** and there are also timers built into the chip that can keep track of longer time periods like minutes or days. So why would we want to have a seperate RTC chip? Because of the biggest reason is that **millis()** only keeps track of time *since the Arduino was last powered* - . That means that when the power is turned on, the millisecond timer is set back to 0. The Arduino doesn't know that it's 'Tuesday' or 'March 8th', all it can tell is 'It's been 14,000 milliseconds since I was last turned on'. what if we wanted to set the time on the Arduino? we have to program in the date and time and you could have it count from that point on. But if it lost power, you'd have to reset the time. Much like very cheap alarm clocks: every time they lose power they blink **12:00**

While this sort of basic timekeeping is OK for some projects, some projects such as data-loggers, clocks, etc will need to have **consistent timekeeping that doesn't reset when the Arduino battery dies or is reprogrammed**. Thus, we include a seperate RTC! The RTC chip is a specialized chip that just keeps track of time. It can count leap-years and knows how many days are in a4.2 Figure: The picture of RTC

4.3 Arduino:

Arduino is an open-source prototyping platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. All this is defined by a set of instructions programmed through the

Arduino Software (IDE). There are many other microcontrollers and microcontroller platforms available for physical computing. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

Inexpensive - Arduino boards are relatively inexpensive compared to other microcontroller platforms.

Cross-platform - The Arduino software runs on Windows, Macintosh OSX, and Linux operating systems.

Simple, clear programming environment - The Arduino programming environment is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well.

Open source and extensible software- The Arduino software is published as open source tools, available for extension by experienced programmers.

Open source and extensible hardware - The Arduino is based on Atmel's ATMEGA8 and ATMEGA168 microcontrollers.

4.4 ADC (Analog to Digital Converter):

An **analog-to-digital converter (ADC, A/D, or A to D)** is a device that converts a continuous physical quantity to a digital number that represents the quantity's amplitude. The conversion involves quantization of the input, so it necessarily introduces a small amount of error. Instead of doing a single conversion, an ADC often performs the conversions periodically. The result is a sequence of digital values that have been converted from a continuous-time and continuous-amplitude analog signal to a discrete-time and discrete-amplitude digital signal.

- An ADC is defined by its bandwidth and its signal to noise ratio.
- The actual bandwidth of an ADC is characterized primarily by its sampling rate, and to a lesser extent by how it handles errors such as aliasing.
- The dynamic range of an ADC is influenced by many factors, including the resolution linearity and accuracy and jitter .
- An ADC may also provide an isolated measurement such as an electronic device that converts an input analog voltage or current to a digital number proportional to the magnitude of the voltage or current. However, some non-electronic or only partially electronic devices, such as rotary encoders.

4.41 ADC types

Direct-conversion ADC or **flash ADC**- has a bank of comparators sampling the input signal in parallel, each firing for their decoded voltage range.

Successive-approximation ADC -uses a comparator to successively narrow a range that contains the input voltage.

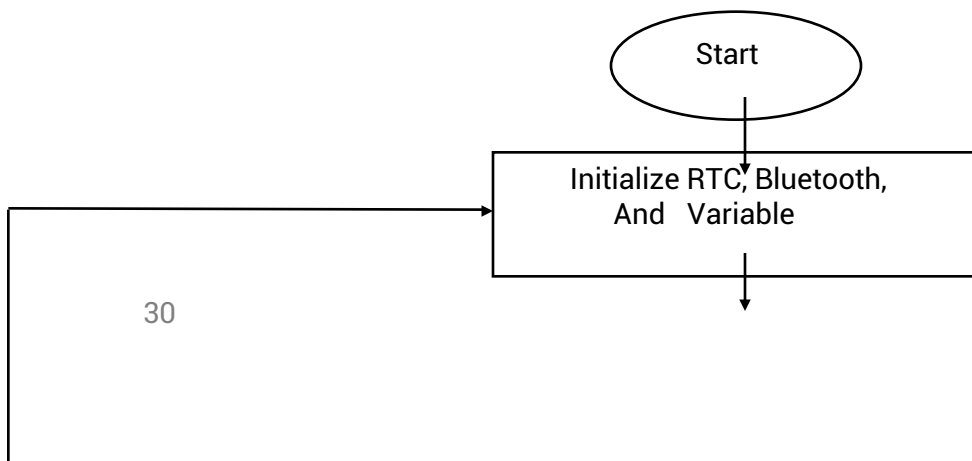
Ramp-compare ADC- produces a saw-tooth signal that ramps up or down then quickly returns to zero.

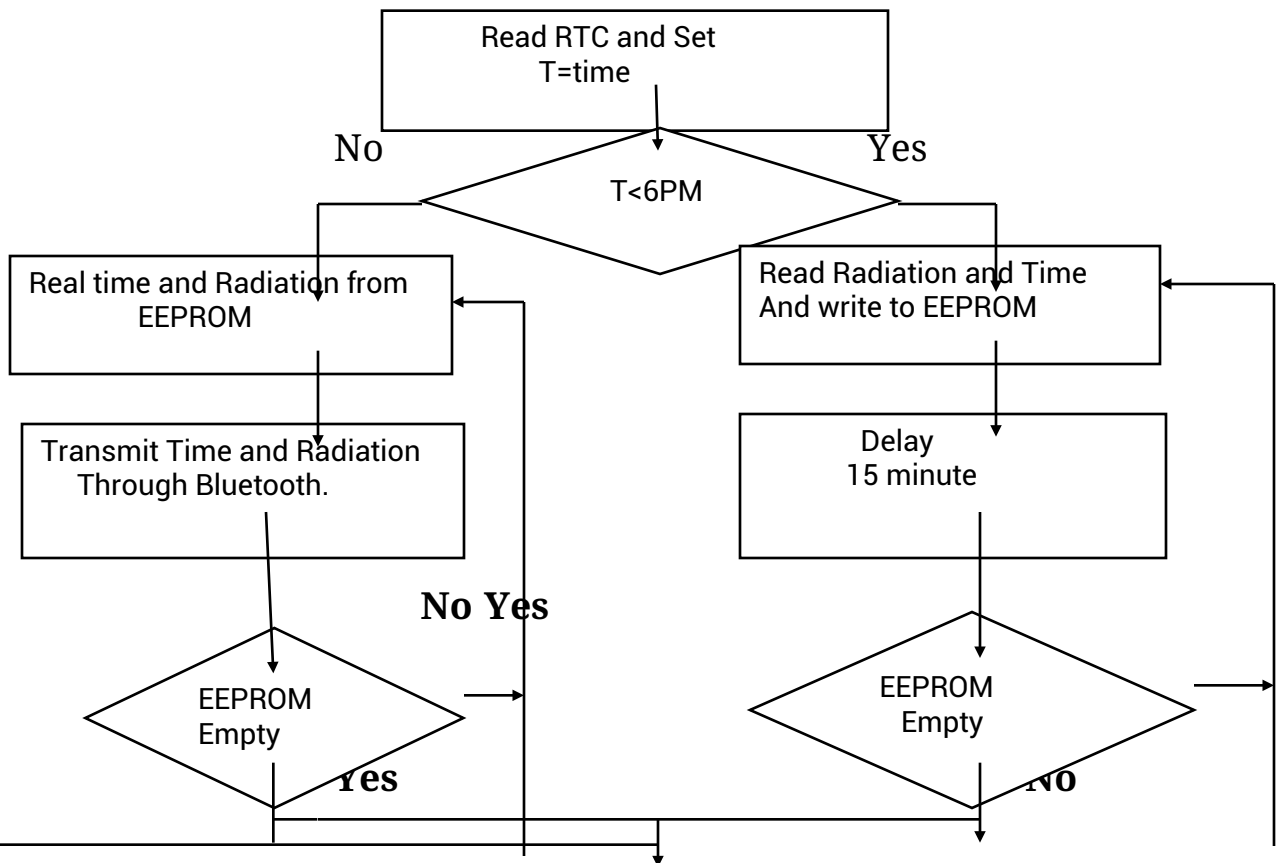
Commercial ADCs often have several inputs that feed the same converter, usually through an analog multiplexer. Different models of ADC may include sample and hold Circuits, instrumentation amplifiers or differential inputs, where the quantity measured is the difference between two voltages.

4.5 Sensor:

A **sensor** is a transducer whose purpose is to sense some characteristic of its environments. It detects events or changes in quantities and provides a corresponding output, generally as an electrical or optical signal; for example, a thermo couple converts temperature to an output voltage. But a mercury-in-glass thermometer is also a sensor; it converts the measured temperature into expansion and contraction of a liquid which can be read on a calibrated glass tu

4.6 Flow Chart of The Program:

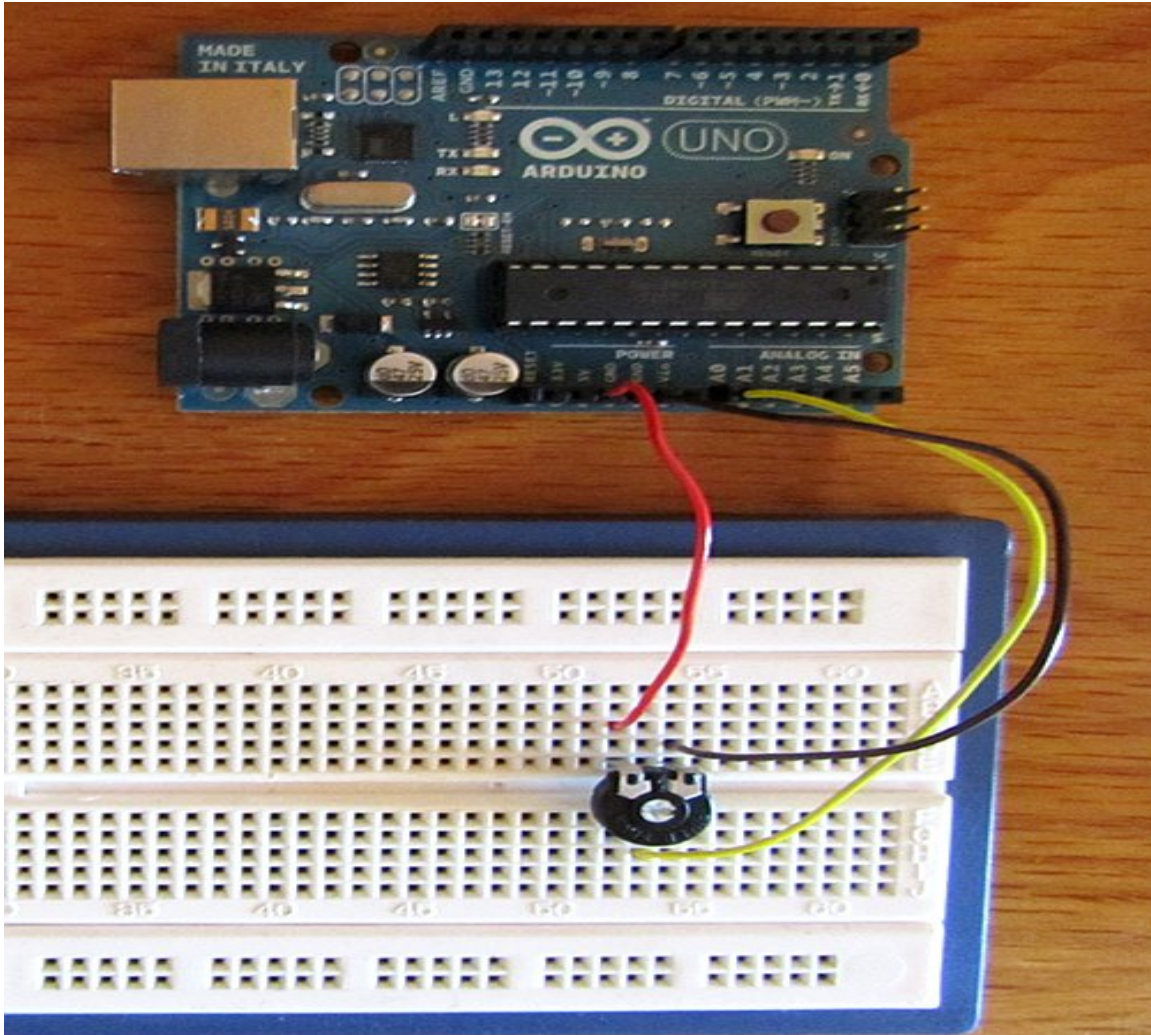




4.6 Fig: Flowchart of whole System

4.7 Reading a Potentiometer:

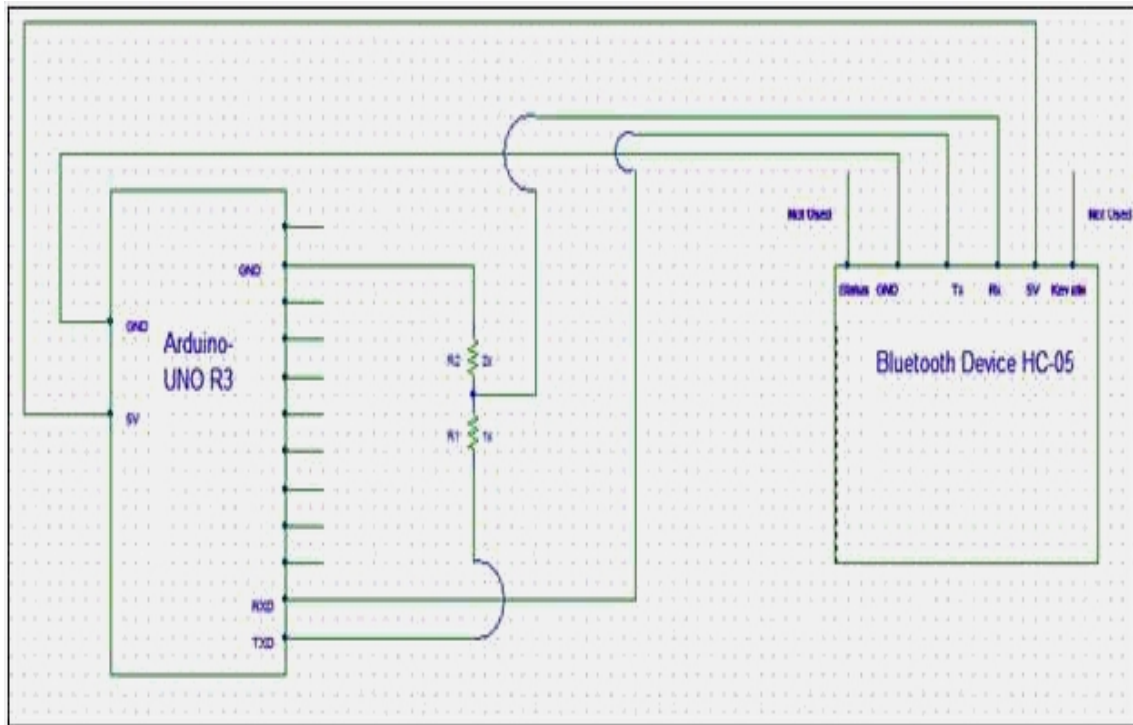
A potentiometer is a simple knob that provides a variable resistance, which we can read into the Arduino board as an analog value. The first goes to ground from one of the outer pins of the potentiometer. The second goes from 5 volts to the other outer pin of the potentiometer. The third goes from analog input 2 to the middle pin of the potentiometer. An analog value is continuously read from one of the analog input pins of the Arduino board. The analog value that is read is continuously sent out of the serial port and displayed in the Arduino IDE serial monitor window. A potentiometer is used to change the analog value.



4.7 Figure: Connection between Aduino and Potentiometer.

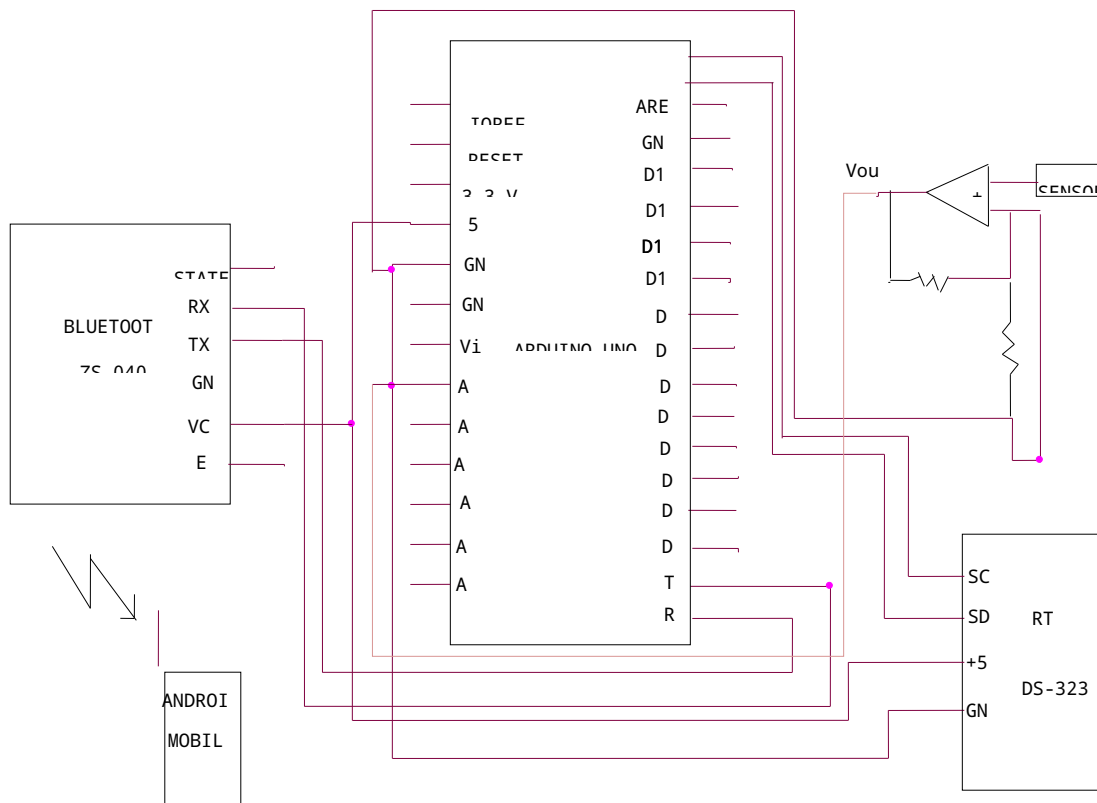
4.8 Connection of Arduino with Bluetooth Device:

Here we have shown connection that we have done in this project. We also developed a connection of Bluetooth device with mobile phone. First, we think about android mobile software which will be appropriate for this project. We install Arduino Bluetooth Controller software on our mobile phone. Then we connect this software with HC-05 Bluetooth module. After connecting software we developed the connection between Arduino and Bluetooth device. We run a program on Arduino board for implementing the project



4.8 Figure: connection between Arduino and Bluetooth.

4.9: whole diagram of project



4.9 Figure: Schemitic Diagram of whole system

Chapter: 5

Performance study of the Developed System

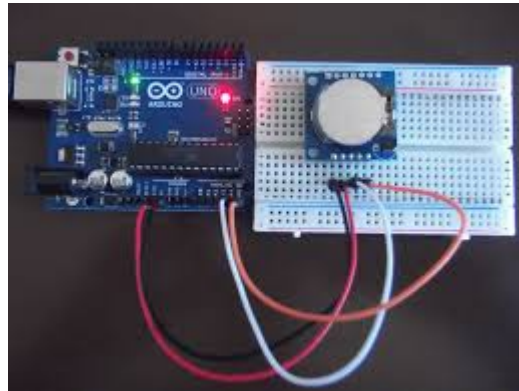
5.1 Implementation

All the parts are connected as circuit design. Then we upload the

programming code in the Arduino and we get positive result. It works properly according to our design.

5.2 Connection between Arduino and real time clock:

In this project connection between Arduino and real time clock was the first work. If you are using a Arduino board, please use D2 (SDA) and D3 (SCL). No other pins on the Arduino are used by the module other than power and GND, so they are all available for connection to other devices in your projects.



5.2 Figure: Connection between Arduino and real Time clock.

Arduinos have not directly the possibility to save time, we expand our Arduino with a small DS1307 Real-Time-Clock (RTC) board. Which can, battery powered, “remember and count” the time and the date.

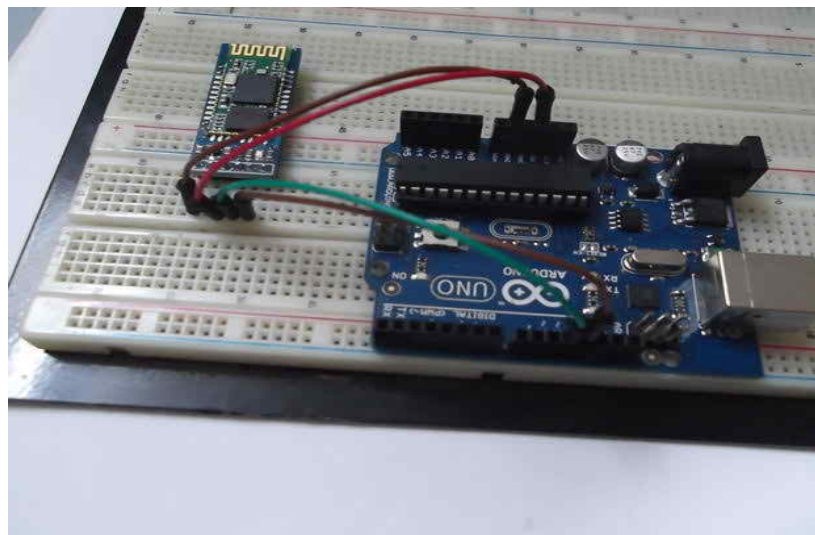
5.2 Table

Module pinout:

Arduino (Nano, Uno)	RTC
5V (VCC)	VCC
GND	GND
A4 (SDA)	SDA

5.3 Connection between Bluetooth and Aduino:

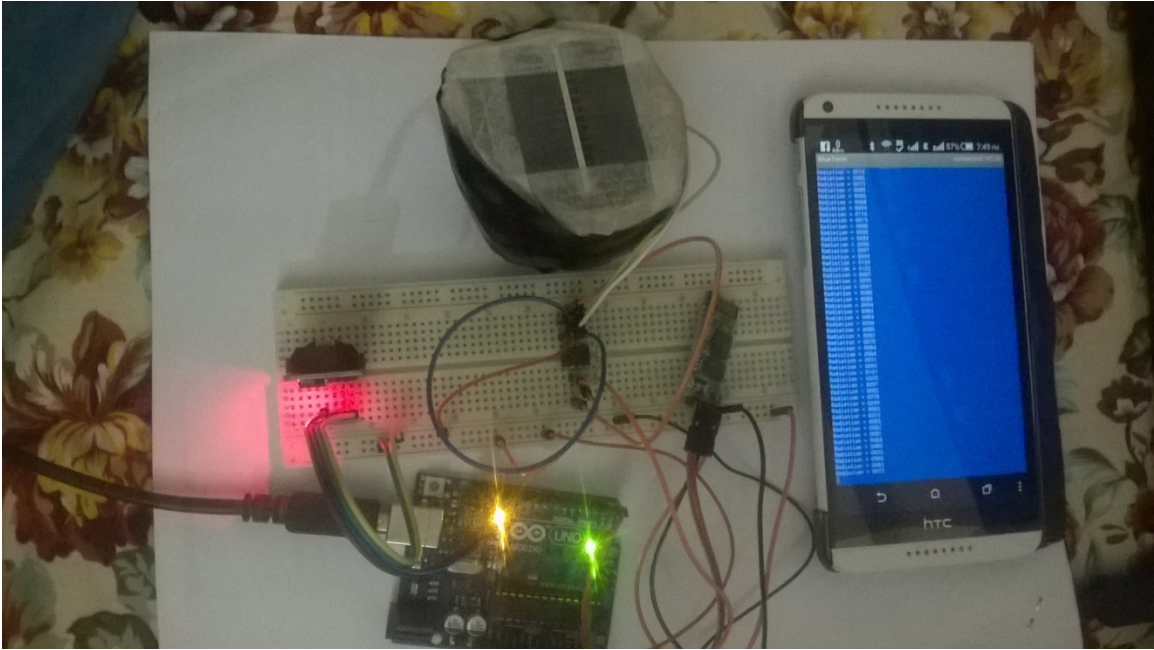
We used software serial library to make pin D10 & D11 As Tx & Rx instead of using the default Rx and tx " D0 &D1 On arduino Board ". For proper connection we read the properties of HC-05 Bluetooth modules from Google. We designed connection between Arduino and HC-05 Bluetooth module as we learn from properties. We developed the connection successfully. Here we represent the photo of this connection.



5.3 Figure: Arduino connected with Bluetooth.

5.4 Development of Whole system:

In this level we connected all the device together. We give connection to Arduino board from every device. Then we upload a program to Arduino for this project. After all successful connection we will get output by controlling every device. Now we are sharing g mobile through Bluetooth device. Then we show radiation data and time in our mobile. Here we collected three types of data. At first we collected data when voltage was zero. Then we collected data when voltage was respectively 3.3and 5volt.From the project we can collect any data from anywhere.



5.4 Figure: The whole Connection of project.

5.5 The output of project:

After applying our desired code we got some result. At first our applied voltage was zero.

```
BlueTerm                                     connected: HC-05
Radiation = 0686
Radiation = 0681
Radiation = 0685
Radiation = 0686
Radiation = 0682
Radiation = 0685
Radiation = 0687
Radiation = 0684
Radiation = 0685
Radiation = 0688
Radiation = 0682
Radiation = 0685
Radiation = 0689
Radiation = 0684
Radiation = 0685
Radiation = 0689
Radiation = 0682
Radiation = 0686
Radiation = 0688
Radiation = 0683
Radiation = 0688
Radiation = 0687
Radiation = 0684
Radiation = 0686
Radiation = 0685
Radiation = 0684
Radiation = 0686
Radiation = 0683
Radiation = 0684
Radiation = 0685
Radiation = 0682
Radiation = 0685
Radiation = 0686
Radiation = 0685
Radiation = 0685
Radiation = 0689
Radiation = 0682
Radiation = 0684
Radiation = 0687
Radiation = 0681
Radiation = 0685
Radiation = 0688
Radiation = 0684
Radiation = 0685
Radiation = 0692
Radiation = 0682
Radiation = 0685
Radiation = 0689
Radiation = 0682
Radiation = 0692
```

5.5 Figure: This result is showed by Android Mobile

When V=3.3volt

```
Radiation = 0007
Radiation = 0014
Radiation = 0006
Radiation = 0008
Radiation = 0016
Radiation = 0001
Radiation = 0004
Radiation = 0014
Radiation = 0007
Radiation = 0014
Radiation = 0014
Radiation = 0007
Radiation = 0013
Radiation = 0009
Radiation = 0007
Radiation = 0004
Radiation = 0000
Radiation = 0006
Radiation = 0005
Radiation = 0000
Radiation = 0004
Radiation = 0007
Radiation = 0001
Radiation = 0006
Radiation = 0008
Radiation = 0001
Radiation = 0005
Radiation = 0017
Radiation = 0006
Radiation = 0006
Radiation = 0015
Radiation = 0002
Radiation = 0006
Radiation = 0016
Radiation = 0001
Radiation = 0006
Radiation = 0015
Radiation = 0001
Radiation = 0015
Radiation = 0014
Radiation = 0007
Radiation = 0005
Radiation = 0015
Radiation = 0001
Radiation = 0013
Radiation = 0014
Radiation = 0008
Radiation = 0013
Radiation = 0009
Radiation = 0007
```

5.51 Figure: Radiation value at v=3.3volt

When voltage=5v

```
Radiation = 1023
Radiation = 1022
Radiation = 1020
Radiation = 1023
Radiation = 1021
Radiation = 1020
Radiation = 1023
Radiation = 1021
Radiation = 1021
Radiation = 1023
Radiation = 1022
Radiation = 1021
Radiation = 1022
Radiation = 1018
Radiation = 1021
Radiation = 1020
Radiation = 1021
Radiation = 1023
Radiation = 1021
Radiation = 1022
Radiation = 1023
Radiation = 1021
Radiation = 1023
Radiation = 1023
Radiation = 1022
Radiation = 1022
Radiation = 1023
Radiation = 1021
Radiation = 1021
Radiation = 1023
Radiation = 1022
Radiation = 1022
Radiation = 1023
Radiation = 1022
Radiation = 1023
Radiation = 1022
Radiation = 1021
Radiation = 1023
Radiation = 1022
Radiation = 1017
Radiation = 1023
Radiation = 1022
Radiation = 1017
Radiation = 1021
Radiation = 1016
Radiation = 1019
Radiation = 1023
Radiation = 1021
Radiation = 1020
```

5.52 Figure: Radiation value at v=5volt

5.6 Product commercialization:

After completing all the circuit and developing the control system, we connected this control system outside of the room where solar radiation is high and saw that it worked properly. Therefore we can say that the real time measurement and interface with mobile through Bluetooth is ready for commercial use.

Chapter 6

conclusion

6.1 conclusion:

At the end of the project we can say the real time measurement and interface with mobile through Bluetooth has been designed and developed for making our life more easy and secured. We used three types voltage respectively 0volt, 3.3volt, and 5volt. We used the HC-05 Bluetooth module for receiving signal from sensor and signal passes to an android mobile phone. Finally, we have designed and developed the whole control system and tested using Android Mobile Phone. We fix all the problems encountered during the design and testing of the system. Finally, we successfully achieved our goals. In this study, the application of microcontroller with improved algorithm of extended specifications has increased the use of android mobile phone and improves the controlling system of any device. The developed real time measurement and interface with mobile through Bluetooth is efficient and the production cost is low. So it is suitable for commercial using.

6.2 Future Work Scope:

This project gives us an opportunity to do a big project in future. The applications stated above are some demo applications that are absolutely possible with its future development. Initially for the limitation of time and required fund we were able to develop just a normal radiation measure for a short distance. Here we can also use RF Receiver and transmitter. Then our receiving range can be wide. Consequently we can receive data from far distance. So, we have a big work scope in this sector. We hope that, we will be able to complete all the features needed for its ultimate applications.

Program code for whole system:

```
#include <EEPROM.h>

const int analogInPin = A0; // Analog input pin that the potentiometer is attached
to

int rad,vl,vh,d4,d3,d2,d1;    // value read from the pot

int i,n,address=0;

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

void loop() {
    int incomingByte;

    address=0;

    for(i=0;i<1;i++)
    {
        rad = analogRead(analogInPin);

        vl=rad%10;

        vh=rad/10;

        //Serial.print(vh*10+vl);

        EEPROM.write(address,vh);

        address++;

        EEPROM.write(address,vl);

        //Serial.print("Radiation = " );

        //Serial.println(rad);

        address++;

        delay(500);
    }
}
```

```
}  
address=0;  
for(i=0;i<1;i++)  
{  
  vh=EEPROM.read(address);  
  address++;  
  vl=EEPROM.read(address);  
  rad=vh*10+vl;  
  d1=rad%10;  
  rad=rad/10;  
  d2=rad%10;  
  rad=rad/10;  
  d3=rad%10;  
  rad=rad/10;  
  d4=rad;  
  Serial.print("Radiation = " );  
  //vl=6;  
  Serial.write(d4+48);  
  Serial.write(d3+48);  
  Serial.write(d2+48);  
  Serial.write(d1+48);  
  Serial.write(10);  
  Serial.write(13);  
  address++;  
  delay(500);  
}
```

```
}
```

```
//Serial.write('A');
```

```
}
```

References:

1. <http://www.mymobilebay.com/stationdata/whatissolarrad.htm>
2. https://www.google.com.bd/search?q=what+is+solar+radiation&ie=utf-8&oe=utf-8&rls=org.mozilla:en-US:official&client=firefox-a&source=hp&channel=np&gws_rd=cr,ssl&ei=tYTBVbC8JY6FuWTh3ongDQ
3. http://study.com/academhttps://en.wikipedia.org/wiki/Solar_energy/lesson/what-is-solar-radiation-definition-effects.html
4. <http://www.agiweb.org/education/energy/solar/>

5. https://en.wikipedia.org/wiki/Solar_energy
6. <http://kids.britannica.com/comptons/art-117453/Lightning-during-a-thunderstorm->
7. <http://www.wisegeek.com/what-is-electrical-energy.htm>
8. <https://en.wikipedia.org/wiki/Arduino>
9. <http://gradestack.com/Microprocessors-and/Introduction-to/Advantages->
10. <http://arduinobasics.blogspot.com/2014/06/433-mhz-rf-module-with-arduino-tutorial.html>
11. [arduino basics. \[blogspot.com/2014/06/433\]\(http://arduinobasics.blogspot.com/2014/06/433\)](http://arduinobasics.blogspot.com/2014/06/433)
12. <https://www.arduino.cc/en/Main/arduino>
13. <https://en.wikipedia.org/wiki/Pyranometer>[https://e](https://en.wikipedia.org/wiki/Pyranometer)
14. <https://en.wikipedia.org/wiki/Real-time>
15. <https://www.google.com.bd/search?q=Schematic+Diagram+of+HC-05>
16. https://en.wikipedia.org/wiki/Real-time_clock
17. <https://en.wikipedia.org/wiki/Sensor>
18. <https://www.google.com.bd/search?q=sensor&client=firefox->