SOLAR CHARGE CONTROLLER

1. Prof. Prashant P. J. Bandekar, 2. Vishvanath Suhas Mohire, 3. Uthkarsh Hindurao Patil, 4. Rahul Sudhakar Musale, 5. Sneha Milind Bhobekar.
Research scholar, Rajaram Shinde College Of Engineering Pedhambe chiplun , MH Engineering,
Students, Bachelor in Engg. Rajaram Shinde College Of Enginnering Pedhambe, chiplun
Electrical Department pedhambe, chiplun , Ratnagiri, MH

Abstract:-In this project we are using arduinouno which is the main part of project, variable battery supply, solar panel, and adapter and led panel as a load. Here variable battery source is used for demonstration purpose through which we can vary the voltage with the help of pot. When the voltage is above 13.8v, the solar stops charging to avoid over charging. At the time of critical condition i.e. solar is OFF & battery voltage is below 11v then the adaptor is connected automatically. & when it exceeds 11v then adapter gets OFF. When solar comes under the sun rays then the adapter gets OFF because the device gives first priority to solar.

Index Terms: Increasing life of battery by opto coupler

I. INTRODUCTION

Arduino is an open-source platform used for building electronics projects. Arduino consists of both a physical programmable circuit board (often referred to as a microcontroller) 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. The Arduino platform has become quite popular with people just starting out with electronics, and for good reason. Unlike most previous programmable circuit boards, the Arduino does not need a separate piece of hardware (called a programmer) in order to load new code onto the board – you can simply use a USB cable. Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program. Finally, Arduino provides a standard form factor that breaks out the functions of the micro-controller into a more accessible pack.

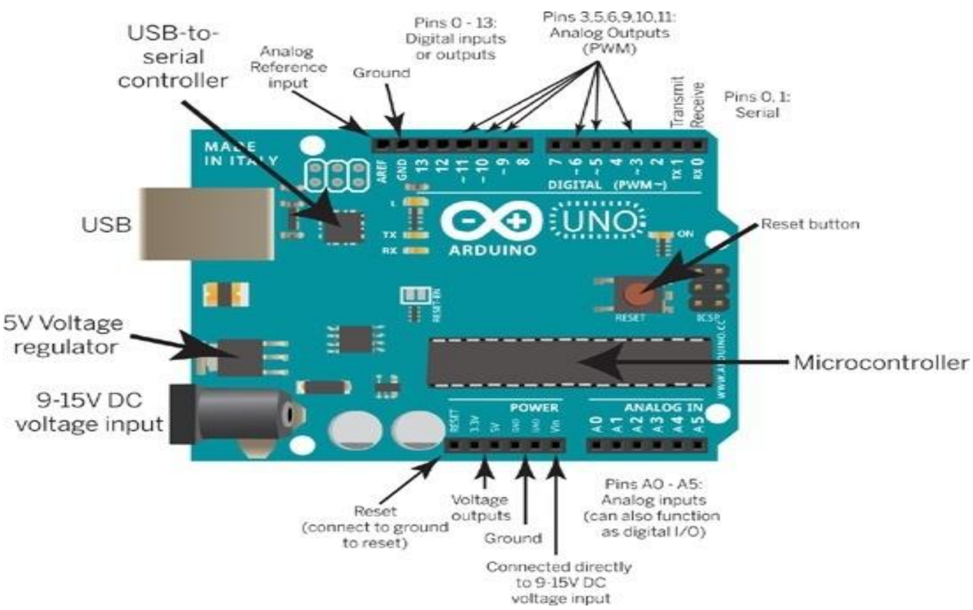

Fig 1. Diagram of Arduino

The Uno is one of the more popular boards in the Arduino family and a great choice for beginners. We'll talk about what's on it and what it can do later in the tutorial. Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals - has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike. Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming. As soon as it reached a wider community, the Arduino board started changing to adapt to new needs and challenges, differentiating its offer from simple 8-bit boards to products for IoT applications, wearable, 3D printing, and embedded environments. All Arduino boards are completely open-source, empowering users to build them independently and eventually adapt them to their particular needs.

The software, too, is open-source, and it is growing through the contributions of users worldwide. Thanks to its simple and accessible user experience, Arduino has been used in thousands of different projects and applications. The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux. Teachers and students use it to build low cost scientific instruments, to prove chemistry and physics principles, or to get started with programming and robotics. Designers and architects build interactive prototypes, musicians and artists use it for installations and to experiment with new musical instruments. Makers, of course, use it to build many of the projects exhibited at the Maker Faire, for example. Arduino is a key tool to learn new things. Anyone - children, hobbyists, artists, programmers - can start tinkering just following the step by step instructions of a kit, or sharing ideas online with other members of the Arduino community. There are many other microcontrollers and microcontroller platforms available for physical computing. Parallax Basic Stamp, Net media's BX-24, Phi gets, MIT's Handyboard, and many others offer similar functionality. All of these tools take the messy details of microcontroller programming and wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems.

II. EASE OF USE

Arduino Uno consists of 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button

Power Jack: Arduino can be power either from the pc through a USB or through external source like adaptor or a battery. It can operate on an external supply of 7 to 12V. Power can be applied externally through the pin Vin or by giving voltage reference through the IO Ref pin.

Digital Inputs: It consists of 14 digital inputs/output pins, each of which provide or take up 40mA current. Some of them have special functions like pins 0 and 1, which act as Rx and Tx respectively , for serial communication, pins 2 and 3-which are external interrupts, pins 3,5,6,9,11 which provides pwm output and pin 13 where LED is connected.

Analog inputs: It has 6 analog input/output pins, each providing a resolution of 10 bits.

A Ref: It provides reference to the analog inputs.

Reset: It resets the microcontroller when low.

Arduino Architecture:

Adriano's processor basically uses the Harvard architecture where the program code and program data have separate memory. It consists of two memories- Program memory and the data memory. The code is stored in the flash program memory, whereas the data is stored in the data memory. The Atmega328 has 32 KB of flash memory for storing code (of which 0.5 KB is used for the boot loader), 2 KB of SRAM and 1 KB of EEPROM and operates with a clock speed of 16MHz.

Arduino Architecture

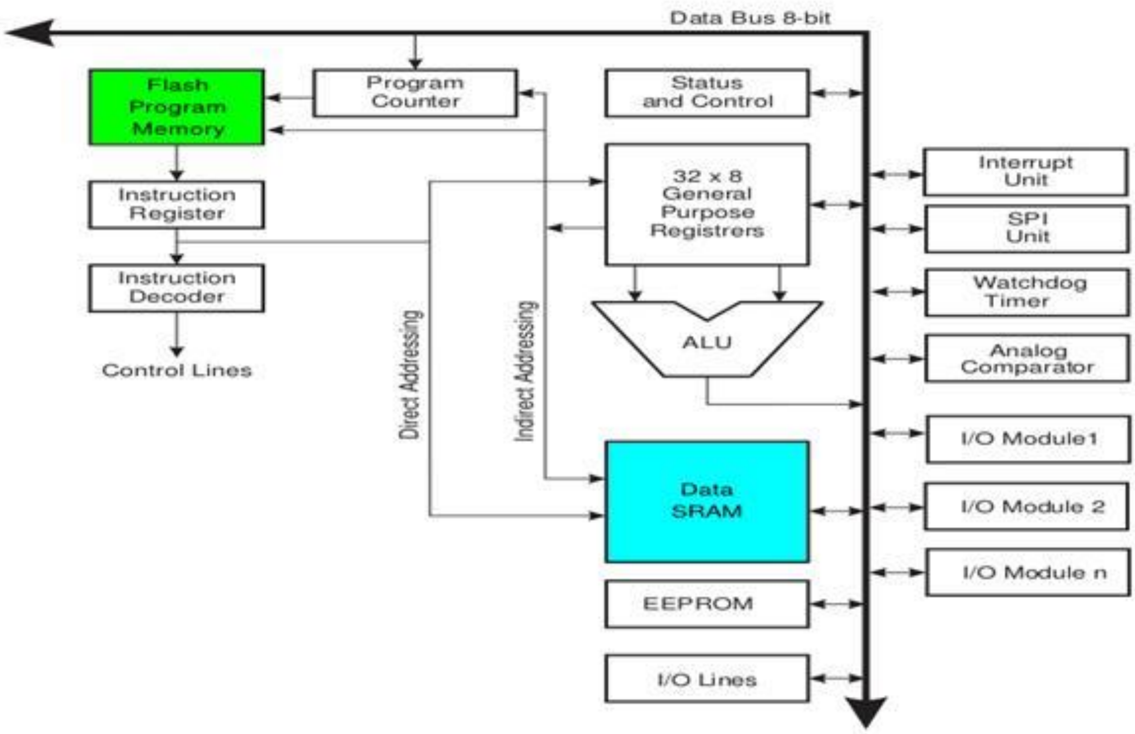

Fig 2. Architecture of Arduino

Steps to program an Arduino

- Programs written in Arduino are known as sketches. A basic sketch consists of 3 parts

1. Declaration of Variables

2. Initialization: It is written in the setup () function.

3. Control code: It is written in the loop () function.

- The sketch is saved with .ino extension. Any operations like verifying, opening a sketch, saving a sketch can be done using the buttons on the toolbar or using the tool menu.
- The sketch should be stored in the sketchbook directory.
- Chose the proper board from the tools menu and the serial port numbers.
- Click on the upload button or chose upload from the tools menu. Thus the code is uploaded by the boot loader onto the microcontroller.

Why Arduino?

Thanks to its simple and accessible user experience, Arduino has been used in thousands of different projects and applications. The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux. Teachers and students use it to build low cost scientific instruments, to prove chemistry and physics principles, or to get started with programming and robotics. Designers and architects build interactive prototypes, musicians and artists use it for installations and to experiment with new musical instruments. Makers, of course, use it to build many of the projects exhibited at the Maker Faire, for example. Arduino is a key tool to learn new things. Anyone - children, hobbyists, artists, programmers - can start tinkering just following the step by step instructions of a kit, or sharing ideas online with other members of the Arduino community. There are many other microcontrollers and microcontroller platforms available for physical computing. Parallax Basic Stamp, Net media's BX-24, Phidgets, MIT's Handyboard, and many others offer similar functionality. All of these tools take the messy details of microcontroller programming and wrap it up in an easy-to-use package. 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. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than $50.
- **Cross-platform -** The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.
- **Simple, clear programming environment -** The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with how the Arduino IDE works.
- **Open source and extensible software -** The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.
- **Open source and extensible hardware -** The plans of the Arduino boards are published under a Creative Commons license, so experienced circuit designers can make their own version of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money.

III. WORKING

There are many varieties of Arduino boards (explained on the next page) that can be used for different purposes. Some boards look a bit different from the one below, but most Arduinos have the majority of these components in common:

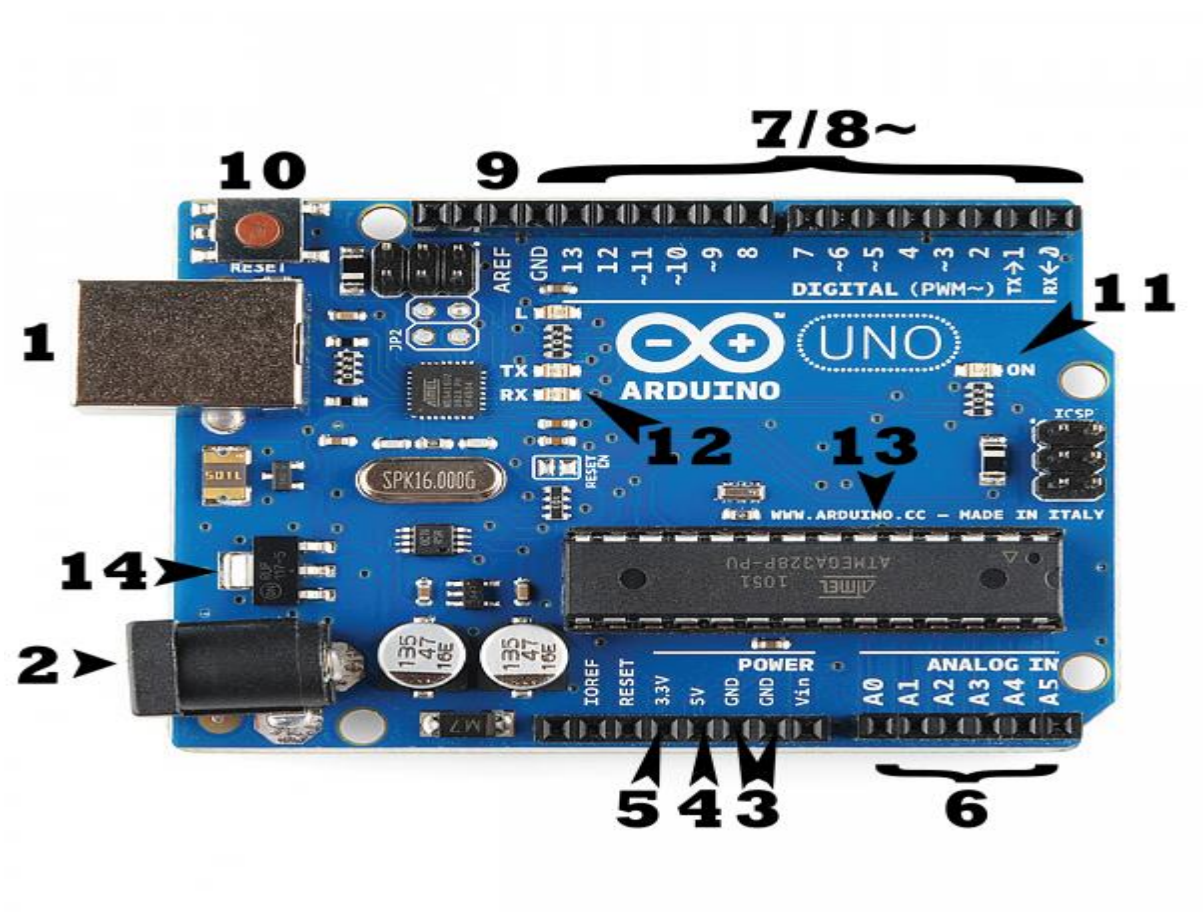

Fig 3. Control Unit of microcontroller

Pin No	Function	Name
1	Ground (0V)	Ground
2	Supply voltage; 5V (4.7V – 5.3V)	Vcc
3	Contrast adjustment; through a variable resistor	V_{EE}
4	Selects command register when low; and data register when high	Register Select
5	Low to write to the register; High to read from the register	Read/write
6	Sends data to data pins when a high to low pulse is given	Enable
7	8-bit data pins	DB0
8		DB1
9		DB2
10		DB3
11		DB4
12		DB5
13		DB6
14		DB7

Table no 1. Specification of component

- Eight(8) Data pins
- VCC (Apply 5v here)
- GND (Ground this pin)
- RS (Register select)
- RW (read - write)
- EN (Enable)
- V0 (Set LCD contrast)

8-Data pins carries 8-bit data or command from an external unit such as microcontroller.

EN (Enable signal)
When you select the register (Command and Data) and set RW (read - write) now it's time to execute the instruction. By instruction i mean the 8-bit data or 8-bit command present on Data lines of LCD. This requires an extra voltage push to execute the instruction and EN (enable) signal is used for this purpose. Usually we make it en=0 and when we want to execute the instruction we make it. High en=1 for some milli seconds. After this we again make it ground en=0.high en=1 for some milli seconds. After this we again make it ground en=0

LCD initialization.

The steps that has to be done for initializing the LCD display is given below and these steps are common for almost all applications.

- Send 38H to the 8 bit data line for initialization
- Send 0FH for making LCD ON, cursor ON and cursor blinking ON.
- Send 06H for incrementing cursor position.
- Send 01H for clearing the display and return the cursor.

Sending data to the LCD.

The steps for sending data to the LCD module is given below. I have already said that the LCD module has pins namely RS, R/W and E. It is the logic state of these pins that make the module to determine whether a given data input is a command or data to be displayed.

- Make R/W low.
- Make RS=0 if data byte is a command and make RS=1 if the data byte is a data to be displayed.
- Place data byte on the data register.
- Pulse E from high to low.
- Repeat above steps for sending another data.

The circuit diagram given above shows how to interface a 16×2 LCD module with AT89S1 microcontroller. Capacitor C3, resistor R3 and push button switch S1 forms the reset circuitry. Ceramic capacitors C1, C2 and crystal X1 is related to the clock circuitry which produces the system clock frequency. P1.0 to P1.7 pins of the microcontroller is connected to the DB0 to DB7 pins of the module respectively and through this route the data goes to the LCD module. P3.3, P3.4 and P3.5 are connected to the E, R/W, RS pins of the microcontroller and through this route the control signals are transferred to the LCD module. Resistor R1 limits the current through the back light LED and so do the back light intensity. POT R2 is used for adjusting the contrast of the display. Program for interfacing LCD to 8051 microcontroller is shown below.

MOSFET:

A **power MOSFET** is a specific type of metal oxide semiconductor field-effect transistor (MOSFET) designed to handle significant power levels.

Compared to the other power semiconductor devices, for example an insulated-gate bipolar transistor (IGBT) or a thyristor, its main advantages are high switching speed and good efficiency at low voltages. It shares with the IGBT an isolated gate that makes it easy to drive. They can be subject to low gain, sometimes to a degree that the gate voltage needs to be higher than the voltage under control.

The design of power MOSFETs was made possible by the evolution of CMOS technology, developed for manufacturing integrated circuits in the late 1970s. The power MOSFET shares its operating principle with its low-power counterpart, the lateral MOSFET.

The power MOSFET is the most widely used low-voltage (that is, less than 200 V) switch. It can be found in most power supplies, DC to DC converters, and low voltage motor controllers.

IV. CONCLUSION

In future we will have to depend on renewable energy. The only source available around us is sunlight. And we can easily convert sunlight energy into electrical energy by using PV cell to meet our requirement. However to extend use of solar power energy to industrial and commercial areas, the price of PV cell need to be brought down through low cost manufacturing techniques. By using solar power energy generation technology the uses of non- conventional resources can be minimized which not only reduce the resources extinction but helps widely in reducing the environmental pollution. By generating power through solar energy the cost of electricity can be minimized and occasionally can be provided with free of cost. Electricity can be supplied even to remote areas without the uses of transmission line.

VI.REFERANCE

1. Khan, B.H. Non- conventional source of energy,5/e, MC. Graw Hill Education
2. Van Valkenburg, M.E: Network Anaysis
3. Milliman, Jacob & Christos Halkias: Integrated Electronics
4. J B Gupta Electronics device & circuit, S. K. Kataria & Son,2009
5. Salivanhanan, S.& S.Arivazhagan: Digital Electronics Vikas Publication
6. KHANCHANDANI 2003 edition, Power electronics
7. VIKAS K SINGH. DPR on “MICROCONTROLLER BASED SOLAR CHARGER”2009
8. Bedford, B D Hoft, R.G.et 1994 PRINCIPLE OF INVERTER CIRCUIT