MODULE 4

- **Open Source Embedded Development Boards**

Introduction. ATmega2560 microcontroller – Block diagram and pin description. Arduino Mega 256 board – introduction and pin description. Simple applications – Solar Tracker, 4-digit 7-Segment LED Display, Tilt Sensor, Home Security Alarm System, Digital Thermometer, IoT applications.

Course Outcome

- After completion of the module the student will be able to
  - Realize external communication interface to the microcontroller

Introduction to Open Source Boards

Open source is one of the most interesting technological topics these days. As its name implies, "open source" refers to software and/or hardware that is freely available. To put another way, people can modify and share open source software and hardware.

There are many points to be considered while choosing open source, some of which are as follows:

- **Open Source Nature:** The advantage of open source is a free license coupled with the fact that the hardware and software are accessible, which means users can change the code, modify the hardware, and add new features to existing versions. Moreover, open source allows multiple contributors to identify and fix any bugs that are present in an existing system.
- **Protection:** The security with open source has both advantages and disadvantages. One benefit is that we can add more value to a product by spotting bugs and fixing them quickly. Contra wise, since all users can view and change the source code, this may lead to unknown bugs and affect the quality of the open source platform.
- **Availability:** Developers can take existing software routines and enhance them as per their applications' requirements. In addition to open source applications, there are also various open source operating systems, such as Android for mobiles and Ubuntu, Fedora, Linux Mint, and Chrome for desktop and embedded applications.

Open Source Development boards

There are dozens of development boards available in the market and some of the popular boards are **Raspberry Pi, Qualcomm Snapdragon, BeagleBone Black, Arduino Mega 2560 etc.**

ATmega2560

ATmega2560 is an 8-bit microcontroller from Microchip. Originally this microcontroller belongs to AVR (Advanced Virtual RISC) family and was introduced in the year 1996 by two students in Norvegian Institute of Technology. This microcontroller is introduced into the market by ATMEL in 1996. Later, in 2016 this company was acquired by Microchip, the famous microcontroller manufacturer behind PIC family of microcontrollers. ATmega2560 is a high performance and low power microcontroller. It's working on a crystal oscillator running at 16MHz. About the memory, it has a Program memory and two types of Data memory. One of the speciality of AVR family of microcontrollers is that, it uses Flash Memory as the Program memory. The advantage of using Flash memory over EEPROM is that the contents of the memory can be erased in a flash as compared to erasing the program written in an EEPROM. In an EEPROM, you can erase the memory in units of blocks; but in a Flash memory, the entire contents can be erased simultaneously resulting faster erasing of memory contents as opposed to EEPROM. ATmega2560 has 256Kb of flash memory. As the data memory, it offers two variants: an 8KB SRAM memory and a 4KB EEPROM. So it offers both volatile and non-volatile form of data memory. If you want to store any critical data in the microcontroller, that you can store in the 4KB EEPROM area. So, having separate code and data memory, ATmega2560 follows Harvard Architecture. It also offers to interface 64KB of external memory, if required. ATmega2560 has 32 general purpose registers of 8-bits.

Atmega2560 also offers In-System Programming (ISP) capability. This means you can download a new program into the Flash memory without removing the controller from the circuit board. Or if the existing program needs to updated, all you needed is to use this feature to burn the new update into the microcontroller.

ATmega2560 has 6 timers/ counters (of which two are 8-bit and four of them are 16bit). It comes with an RTC to keep track of current time and date. 14 PWM channels and sixteen 10-bit ADC are also part of this controller. For communicating with other devices serially, this controller supports 4 UARTS, one I2C and one SPI (Serial Peripheral Interface). These are different serial communication protocols using which a device can send data serially to another device. Each UARTs contain a TX and RX pins for one to one communication with other devices. I2C needs two lines SDA(Serial Data) and SCK(Serial Clock) lines to synchronously send data between two devices in a master-slave architecture. SPI is the most

advanced one among these which requires four lines – Serial Clock(SCK), Master-in Slave-out(MISO) line (for transmitting data from a slave to master(controller)), Master-out Slave-in(MOSI) line (for receiving data from a slave to master(controller)) and a Slave Select(SS) line. Atmega2560 comes with a Watchdog timer to make sure that the controller doesn't get into an indefinite loop.

ATmega2560 controller comes in a 100 pin TQFP package (Thin Quad Flat Package). [Quad means pins are present at the four sides of the IC]. Of which, 86 are I/O lines(grouped into 11 ports – Port A – 8 bits, Port B – 8 bits, Port C – 8 bits, Port D – 8 bits, Port E – 8 bits, Port F – 8 bits, Port G – 6 bits, Port H – 8 bits, Port J – 8 bits, Port K – 8 bits, Port L – 8 bits). Some of these pins are also having some dual functions in addition being used as I/O lines. To connect the Crystal oscillator, two pins XTAL1 and XTAL2 are provided. To reset the controller, a RESET pin there which is active LOW. Four VCC pins and five GND pins are provided in the controller. In connection with ADC, two pins AVCC and AREF are also there making the total tally of pins to 100.

Features

• High Performance, Low Power AVR 8-Bit Microcontroller

• Advanced RISC Architecture

– 135 Powerful Instructions – Most Single Clock Cycle Execution

– 32 × 8 General Purpose Working Registers

– Up to 16 MIPS Throughput at 16MHz

• High Endurance(life cycle) Non-volatile Memory Segments

– 64K/128K/256KBytes of In-System Self-Programmable Flash

– 4Kbytes EEPROM

– 8Kbytes Internal SRAM

– Write/Erase Cycles: 10,000 Flash/100,000 EEPROM

– Data retention: 20 years at 85C/ 100 years at 25C

• In-System Programming by On-chip Boot Program

• Endurance: Up to 64Kbytes Optional External Memory Space

• JTAG (IEEE® std. 1149.1 compliant) Interface

– Boundary-scan Capabilities According to the JTAG Standard

– Extensive On-chip Debug Support

– Programming of Flash, EEPROM, Fuses, and Lock Bits through the JTAG Interface

• Peripheral Features

– Two 8-bit Timer/Counters

– Four 16-bit Timer/Counter

– Real Time Counter with Separate Oscillator

– Four 8-bit PWM Channels

– Twelve PWM Channels with Programmable Resolution from 2 to 16 Bits

– 16-channel, 10-bit ADC

– Four Programmable Serial USART

– Master/Slave SPI Serial Interface

– Programmable Watchdog Timer with Separate On-chip Oscillator

• Special Microcontroller Features

– Power-on Reset and Programmable Brown-out Detection

– Internal Calibrated Oscillator

– External and Internal Interrupt Sources

– Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby

• I/O and Packages

– 86 Programmable I/O Lines

–100-lead TQFP, 100-ball CBGA

• Temperature Range:

– 40C to 85C Industrial

• Ultra-Low Power Consumption

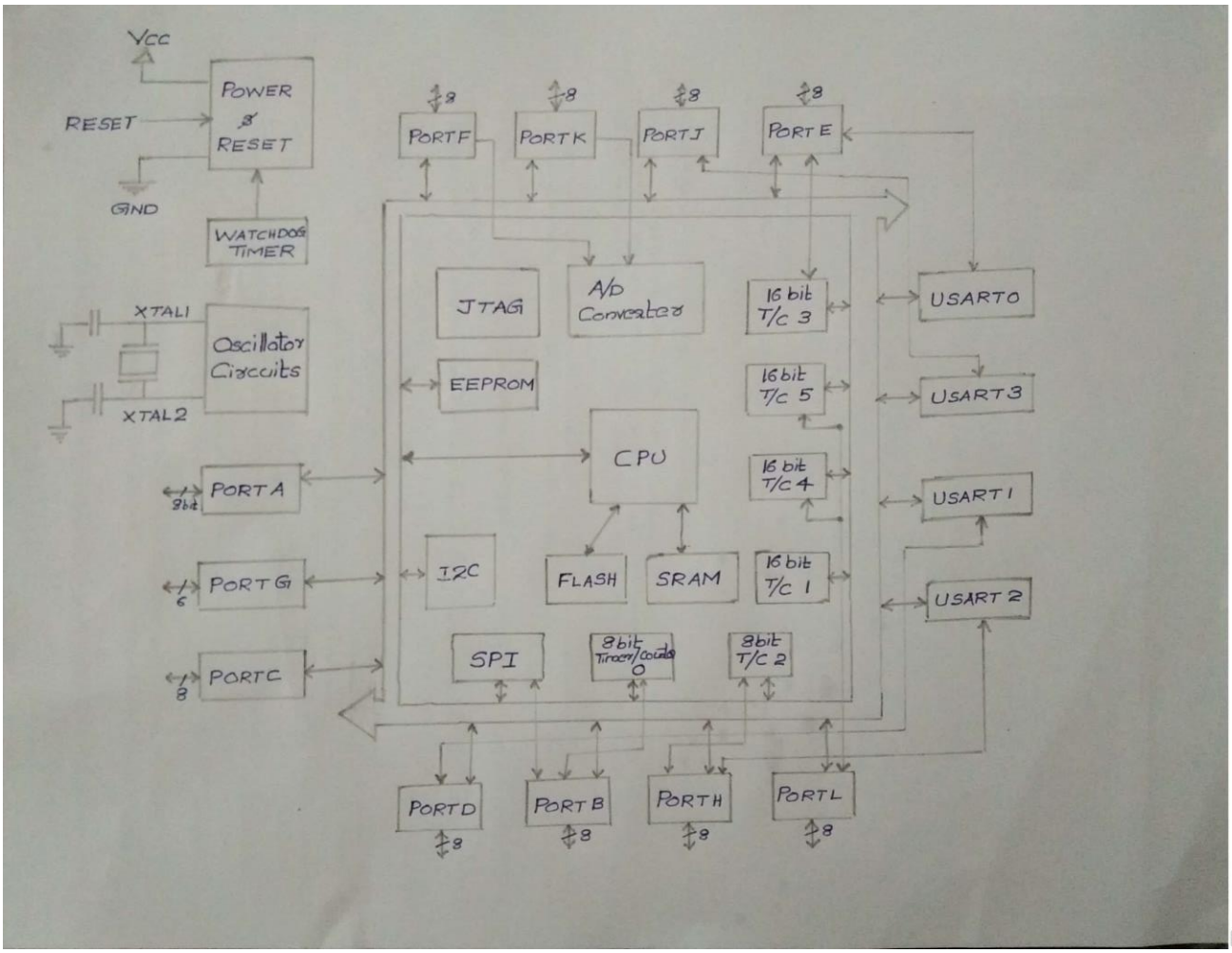
– Active Mode: 1MHz, 1.8V: 500µA

– Power-down Mode: 0.1µA at 1.8V

• Speed Grade:

– ATmega2560:• 0 - 16MHz @ 4.5V - 5.5V

- **ATmega2560 Block Diagram**



The ATmega2560 is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega2560 achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

The AVR core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

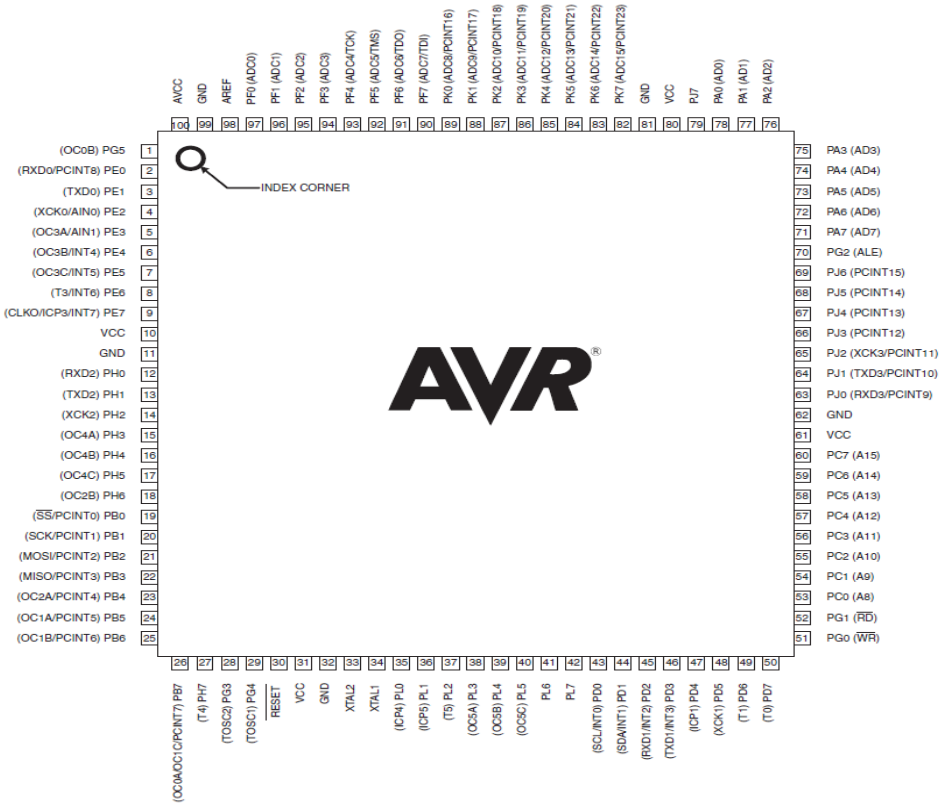
The ATmega2560 provides the following features: 256Kbytes of In-System Programmable Flash with Read-While-Write capabilities, 4K bytes EEPROM, 8K bytes SRAM, 86 general purpose I/O lines, 32 general purpose working registers, Real Time Counter (RTC), six flexible Timer/Counters, 4 USARTs, a 16-channel, 10- bit ADC, Watchdog Timer with Internal Oscillator, an SPI serial port, JTAG test interface-used used for accessing the On-chip Debug system and programming and six software selectable power saving modes. The Idle mode stops the CPU while allowing the SRAM, Timer/Counters, SPI port, and interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the Oscillator, disabling all other chip functions until the next interrupt or Hardware Reset. In Power-save mode, the asynchronous timer continues to run, allowing the user to maintain a timer base while the rest of the device is sleeping. The ADC Noise Reduction mode stops the CPU and all I/O modules except Asynchronous Timer and ADC, to minimize switching noise during ADC conversions. In Standby mode, the Crystal/Resonator Oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low power consumption. In Extended Standby mode, both the main Oscillator and the Asynchronous Timer continue to run.

The device is manufactured using Atmel’s high-density non-volatile memory technology. The On-chip ISP Flash allows the program memory to be reprogrammed in-system through an SPI serial interface, by a conventional non-volatile memory programmer, or by an On-chip Boot program running on the AVR core. The boot program can use any interface to download the application program in the application Flash memory. Software in the Boot Flash section will continue to run while the Application Flash section is updated, providing true Read-While-Write operation. By combining an 8-bit RISC CPU with In-System Self-Programmable Flash on a

monolithic chip, the Atmel ATmega2560 is a powerful microcontroller that provides a highly flexible and cost effective solution to many embedded control applications.

The ATmega2560 AVR is supported with a full suite of program and system development tools including: C compilers, macro assemblers, program debugger/simulators, in-circuit emulators, and evaluation kits.

➢ Pin Descriptions



VCC Digital supply voltage.

GND Ground.

Port A (PA7..PA0)

Port A is an 8-bit bi-directional I/O port with internal pull-up resistors. The Port A has an alternate function as the address low byte and data lines for the External Memory.

Port B (PB7..PB0)

Port B is an 8-bit bi-directional I/O port with internal pull-up resistors. Port B also serves the functions of various special features – SPI serial communication & timer/Counter operation.

Port C (PC7..PC0)

Port C is an 8-bit bi-directional I/O port with internal pull-up resistors. The Port C has an alternate function as the address high byte for the External Memory.

Port D (PD7..PD0)

Port D is an 8-bit bi-directional I/O port with internal pull-up resistors. Port D also serves the functions of various special features – Timer/ Counter operation and as Interrupt pins.

Port E (PE7..PE0)

Port E is an 8-bit bi-directional I/O port with internal pull-up resistors. The Port E also serves the functions of various special features –as Interrupt pins & for USART 0.

Port F (PF7..PF0)

Port F is an 8-bit bi-directional I/O port. Port F serves as analog inputs to the A/D Converter.

Port G (PG5..PG0)

Port G is a 6-bit I/O port with internal pull-up resistors. Port G also serves the functions of various special features (ALE, RD and WR signals & Timer/Counter related pins).

Port H (PH7..PH0)

Port H is a 8-bit bi-directional I/O port with internal pull-up resistors. Port H also serves the functions of various special features (USART and Timer / Counter operation).

Port J (PJ7..PJ0)

Port J is an 8-bit bi-directional I/O port. Port J also serves the functions of various special features (USART and Interrupt operation related pins).

Port K (PK7..PK0)

Port K is an 8-bit bi-directional I/O port. Port K serves as analog inputs to the A/D Converter.

Port L (PL7..PL0)

Port L is an 8-bit bi-directional I/O port. Port L also serves the functions of various special features (Timer/ Counter operation).

RESET

Reset input. A low level on this pin for longer than the minimum pulse length will generate a reset, even if the clock

is not running.

XTAL1 & XTAL2

Pins to which the crystal oscillator is connected.

AVCC

AVCC is the supply voltage pin for Port F and the A/D Converter.

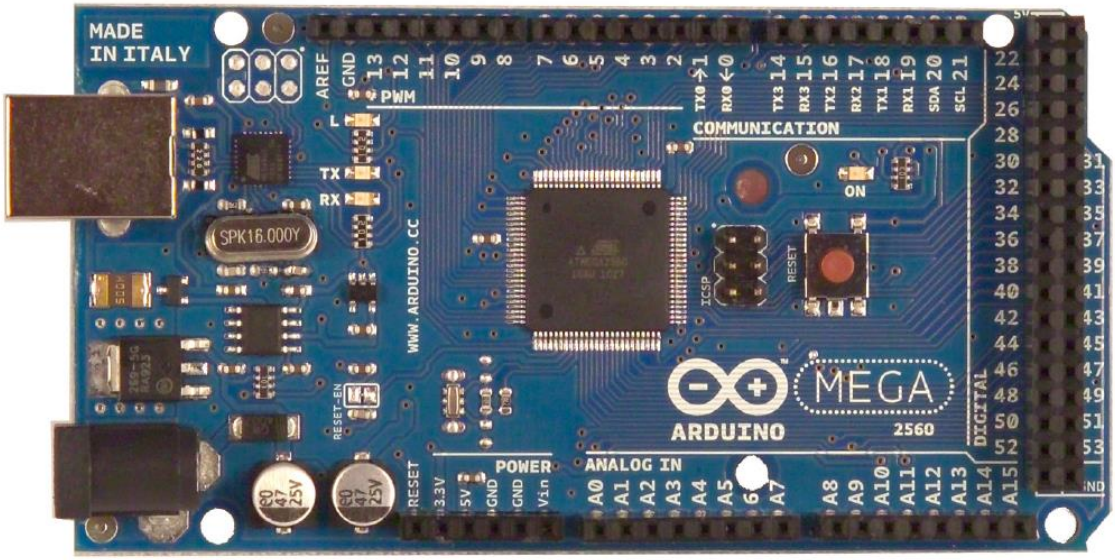
AREF

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

➢ Arduino Mega 2560 Board

Overview

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560. It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC to-DC adapter or battery to get started.



Power

The Arduino Mega can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

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.

The power pins are as follows:

- **VIN.** 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.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

Memory

The ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM.

Input and Output

Each of the 54 digital pins on the Mega can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor.

In addition, some pins have specialized functions:

● **Serial: 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX).** Used to receive (RX) and transmit (TX) TTL serial data.

● **External Interrupts: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2).** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.

● **PWM: 0 to 13.** Provide 8-bit PWM output with the analogWrite() function.

● **SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS).** These pins support SPI communication using the SPI library.

● **LED: 13.** 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.

● **I2C: 20 (SDA) and 21 (SCL).** Support I2C (TWI) communication using the Wire library.

Analog Pins

The Mega2560 has 16 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). 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 and analogReference() function.

There are a couple of other pins on the board:

● **AREF.** Reference voltage for the analog inputs. Used with analogReference().

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

Communication

The Arduino Mega2560 has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega2560 provides four hardware UARTs for serial communication.

A SoftwareSerial library allows for serial communication on any of the Mega2560's digital pins. The ATmega2560 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus. For SPI communication, use the SPI library.

➢ Solar Tracker

Solar Tracker is a Device which follows the movement of the sun as it rotates from the east to the west every day. The Solar Panel Tracker is designed to follow the sun movement so that maximum light intensity hits on the solar panel, thus increasing the power efficiency. This system can also be successfully implemented in other solar energy-based projects like water heaters and steam turbines.

Types of solar tracking system

As per the mode of motions, the solar tracking system is classified into two types,

- **Single-axis solar tracking system** - adjustment of the surface happens around only one axis. Single axis trackers are adjusted every month or so account for seasonal changes in the suns position, the single axis is then used to track the daily movement of the sun across the sky.
- **Dual-axis solar tracking system**- rotation of the surface happens by rotating around two axes simultaneously. Dual axis trackers eliminate the need for monthly adjustment by using one axis to track the suns daily movement and another axis to track the seasonal movement.

A single axis solar tracker improves solar output by around 25% and a dual axis tracker by around 40% .

We have designed a single-axis solar tracking system. In this system, the whole solar panel moves from east to west in a day to point in the direction of the sun. This solar tracker control system is designed to take light measurements from the east and west (left and right) side of the solar panel and determine which way to move the panel to point it directly at the source of the light.

A servo is used to actuate the panel tracker. Although this tracker is single axis, the two sensors and servo can simply be duplicated to provide dual axis control.

How it will work?

The solar panel uses photovoltaic cells (PV cells). The PV cells detect the light intensity and according to that, the tracker adjusts the direction of solar panel to the position of the Sun in the sky.

Every time, the tracker adjusts the panel perpendicular to the Sun, so more sunlight strikes the solar panel and less light is reflected. Hence, it absorbs more energy which can be converted into power.

In this Prototype, we are using the LDR sensor to detect the light(sun) intensity and servo motors for automatic rotation of the panel using the Arduino microcontroller. Arduino Mega 256 board is used to control the motor as per the output of the LDR sensor.

Sun Tracking Solar Panel using Arduino, in which we will use two LDRs (Light-dependent resistor) to sense the light and a servo motor to automatically rotate the solar panel in the direction of the sunlight. The advantage of this is that the Solar panels will always follow the sunlight and will always face the sun to get charge all the time and can provide maximum power.

The LDRs (light dependent resistors) or PRs (photo-resistors) change resistance with changing light, therefore they need to be connected in such a way that the changing resistance is converted into a changing voltage signal which the Arduino understands. The servo is controlled through one of the Arduino's PWM outputs.

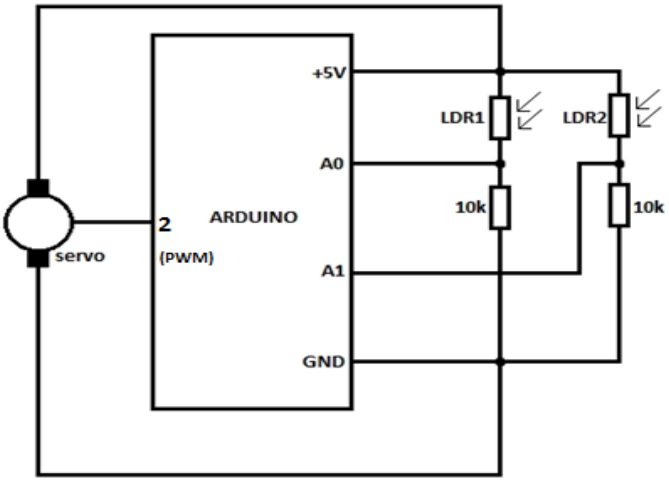
The following are the component requires to build a solar tracking system using Arduino

- Servo Motor (sg90)
- Solar panel
- Arduino Mega256
- LDR's X 2 (Light Dependent Resistor)
- 10K resistors X 2
- Battery (6 to 12V)

Circuit Diagram

The circuit diagram for the Arduino Based Solar Tracker Using LDR & Servo Motor is given below. This solar tracker system uses the Arduino Mega board, a servomotor, 2 LDRs and 2 resistors to rotate the solar panel towards the Sun or a source of light.

Servo motor can rotate approximately 90° in each direction and is controlled using the included Arduino's Servo Library.



Principle:

Two LDR's (Light Dependent Resistor) LDR1 & LDR2 are connected to Analog pins of the Arduino. A solar plate is attached in parallel to the axis of the servo motor and both the sensors are kept on the solar plate as shown in the figure .

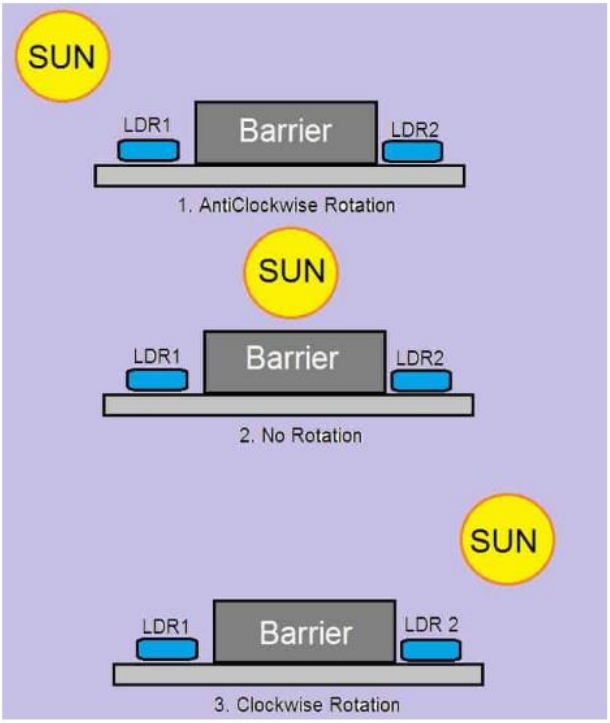
The design & the arrangement is done in such a manner that the movement of the sun is from LDR1 to LDR2, as shown in the image below.

There are three cases that are to be followed:-

Case 1: Sun is in the left side
Light on LDR1 is high because the shadow of barrier falls on LDR2 so solar plate moves anti-clockwise.

Case 2: Sun is in right Side
Light on LDR2 is high because the shadow of barrier falls on LDR1 so solar plate movie clockwise.

Case 3: Sun is in the Center
Light on both LDR's is equal so, plate will not rotate in any direction.



Working:

LDR's are working as light detectors. **LDR (Light Dependent Resistor)** also known as photo resistor is the light sensitive device. Its resistance decrease when the light falls on it. The two LDR's are placed at the two sides of the solar panel and the **Servo Motor** is used to rotate the solar panel. The servo will move the solar panel towards the LDR whose resistance will be low, that way it will keep following the light. And if there is some amount of light falling on both the LDR, then the servo will not rotate. The servo will try to move the solar panel in the position where both LDR's will have the same resistance means where the same amount of light will fall on both the resistors and if the resistance of one of the LDR will change then it rotates towards lower resistance LDR.

Arduino converts the analogs values of LDR sensors (pins A0, A1) into digitals. Then it controls a servomotor(left-right) using Pulse-Width Modulation (PWM) signal(pins 9) to track the sun.

4- Digit 7- Segment LED Display

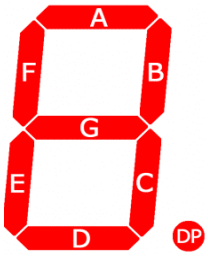
A **Seven-segment display** (**SSD**), or **seven-segment indicator**, is a form of electronic display device for displaying decimal numerals that is an alternative to the more complex dot matrix displays. A 7-segment display has the lowest price amongst all types of displays. It is widely used in devices that shows numerical information.

Seven-segment displays are widely used in digital clocks, electronic meters, basic calculators, counter machines, fancy shop banners and other electronic devices that display numerical information.

To display alphabets and symbols, opting-in for LCD would be the best choice.

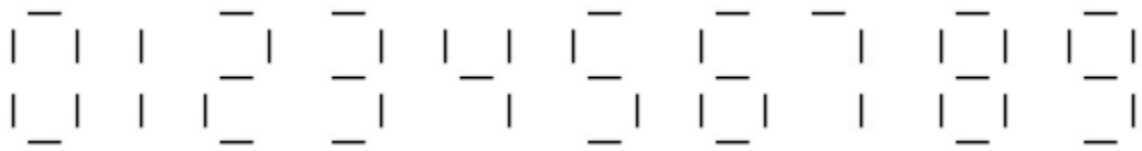
What is a 7-Segment Display?

A 7-segment display is nothing but a pack of seven LEDs connected together where each LED is known as a segment. All of them can be controlled individually.



Seven segment displays consist of 7 LEDs, called segments, arranged in the shape of an "8". Most 7-segment displays actually have 8 segments, with a dot on the right side of the digit that serves as a decimal point. Each segment is named with a letter A to G, and DP for the decimal. The circular LED indicates decimal point in numeral.

Each segment on the display can be controlled individually, just like a regular LED. Each segment can be powered separately to display digits from 0 to 9. The following figure shows the pattern of digits displayed by a 7 – segment LED display.



7-Segment displays are available in various colors (Red, Blue, and Green) and sizes (0.56 to 6.5 inches). Sometimes two to four 7-segment displays are packed together to form a big display (refer to following image).

1-Digit

2-Digit

3-Digit

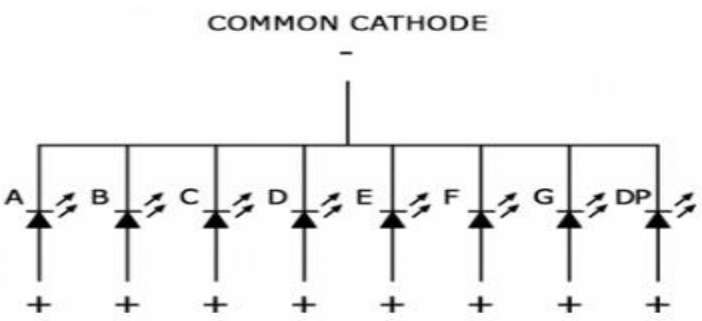
4-Digit

7-Segment Display Types

Depending upon the LED anode and cathode connections 7-segment displays are divided into two types.

1. Common Cathode (CC)

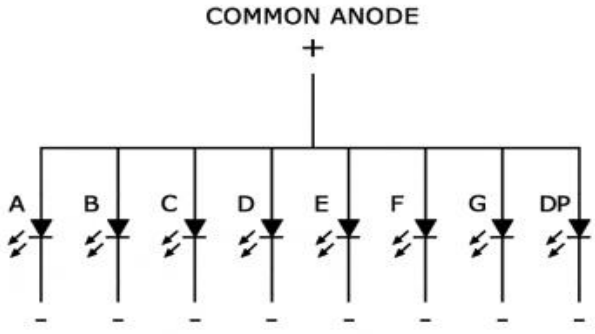
In this configuration, all cathode terminals are connected together, and anode terminals are left open.To turn on the LED display, connect cathode to GND terminal and anode to 5V supply.



Individual segments are turned on and off by switching power to the anodes:

2. Common Anode (CA)

In this configuration, all anode terminals are connected, and cathode terminals are kept open.To turn on the LED display, connect anode to 5V supply and cathode to GND.



Individual segments are turned on and off by switching power to the cathodes:

Arduino 4-Digit 7-Segment LED Display

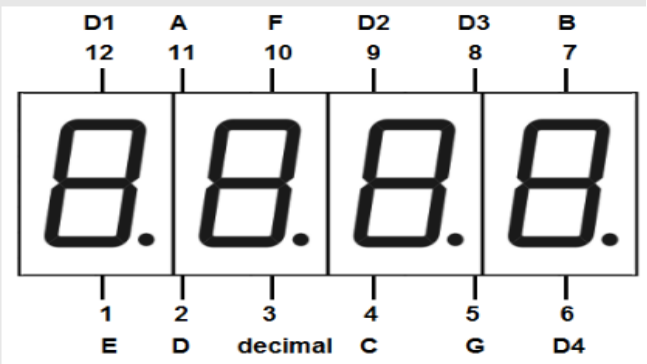
A 4 – digit 7 segment LED display is used to display numbers using Arduino.Either a compact module containing four 7- segment LED displays can be used or four individual 7 – segment displays can be used by multiplexing them.

FEATURES

- Perfect display uniformity for LED segment display
- Low power dissipation
- Wide viewing angle
- Luminous intensity compatible

PIN OUT – 4 DIGIT 7 SEGMENT DISPLAY

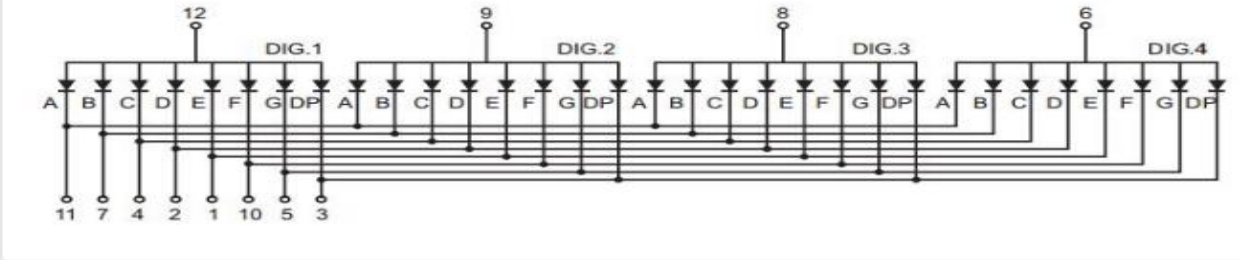
A 4-digit 7-segment LED display has 12 pins. 8 of the pins are for the 8 LEDs on each of the 7 segment displays, which includes A-G and DP (decimal point). The other 4 pins represent each of the 4 digits from D1-D4.



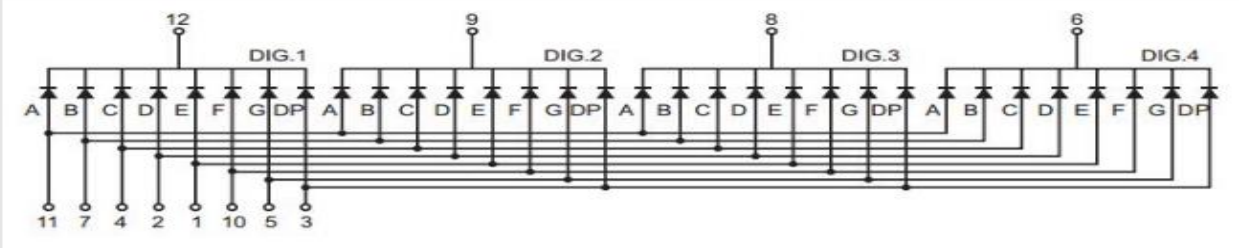
COMMON ANODE OR COMMON CATHODE

Each segment in the display module is multiplexed, meaning it shares the same connection points. And each of the four digits in the module have their own common cathode / common anode connection point. This allows each digit to be turned on or off independently. Also, this multiplexing technique turns the massive amount of microcontroller pins necessary to control a display into just eleven or twelve.

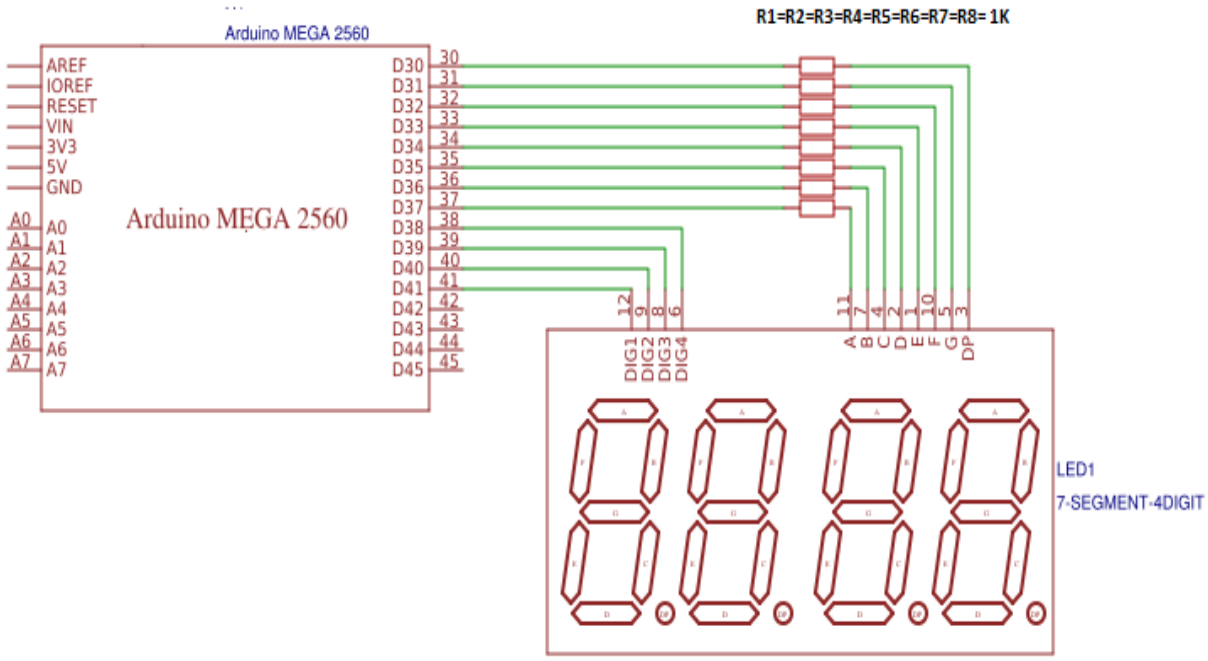
Common Anode



Common Cathode



Interfacing 7-segment Display with Arduino



We assume that the selected 7 – segment module is of common cathode (CC) type.

Arduino digital pins 37,36,35,34,33,32,31 and 30 should go to Display pins 11, 7,4,2,1,10,5 and 3 in correct order. They are connected through 1K resistors.

We assume that the selected 7 – segment module is of common cathode type.

The four common pins (common cathode pins) from four displays 12,9,8 and 6 are connected to Arduino digital pins 41,40,39 and 38 respectively. As the cathode is the common terminal here each display will be selected by giving a low signal to the CC pins and a high on the remaining pins (anode pins) will turn on corresponding display segments as shown in truth table below.

Segments (✓ = ON)							Display	Segments (✓ = ON)							Display
a	b	c	d	e	f	g		a	b	c	d	e	f	g	
✓	✓	✓	✓	✓	✓		0	✓	✓	✓	✓	✓	✓	✓	8
	✓	✓					1	✓	✓	✓			✓	✓	9
✓	✓		✓	✓		✓	2	✓	✓	✓		✓	✓	✓	A
✓	✓	✓	✓			✓	3			✓	✓	✓	✓	✓	b
	✓	✓			✓	✓	4	✓			✓	✓	✓		C
✓		✓	✓		✓	✓	5		✓	✓	✓	✓		✓	d
✓		✓	✓	✓	✓	✓	6	✓			✓	✓	✓	✓	E
✓	✓	✓					7	✓				✓	✓	✓	F

So how we are going to display a number like 1234 on this 4 digit display? For this we are going to use a method called multiplexing. What multiplexing does is simple – show one digit at a time on a display unit and switch between display units very fast. Due to persistence of vision, human eye can not differentiate between which display is ON/OFF. The human eye just visualizes all the 4 display units to be ON all the time. Let's say we need to show 1234. First we turn on the segments relevant to "1" and turn on the 1st display unit. Then we send signals to show "2", turn off 1st display unit and turn on 2nd display unit. We repeat this process for next two numbers and switching between display units should be done very fast (about within one second delay). As our eyes can't pick a change occurring repeatedly to any object within 1 second, what we see is 1234 appearing on the display at the same time

TILT SENSOR

A Tilt Sensor or a Tilt Switch is a component that detects orientation of an object. It is an electronic device used to detect the orientation or tilting of an object and provides digital output based on the orientation.

One of the best example for the application of a tilt sensor is its use in aircrafts. The horizontal and vertical orientation or inclination of the airplane will be provided by the tilt sensor to on board computers. This information is provided to the pilot for safe travelling.

They are sometimes referred as "mercury switches", "rolling ball sensors", "tilt switches", "inclinometer", "accelerometer". Tilt sensors are used in various fields like robotics, aviation, vehicle secular systems, portable computers.

Principle:

A tilt sensor is a device that produces an electrical signal which is proportional to the angular movement. Orientation or Inclination can be detected using these sensors. It behaves like a normal ON – OFF switch that is controlled by tilting. Basically, it has a ball inside it which moves and makes the circuit. So tilt sensor can turn on or off the circuit based on its orientation.

There are different types of tilt sensors based on the axes it can measure.

- For a simple one axes orientation, a tilt switch with accurate angle of orientation can be used.
- An accelerometer based 3- axis tilt sensor is used to detect full motion in three axes

A simple tilt sensor is basically a switch that will turn ON or OFF based on angle or orientation of the sensor. Such sensor is useful for single axis tilt detection.
In this explanation, we use a single axes tilt sensor.

There are two technologies that are used in the implementation of Tilt Sensors: Mercury based and Roller Ball based. Older tilt sensors are made of mercury.

1) **Mercury Tilt Sensor**

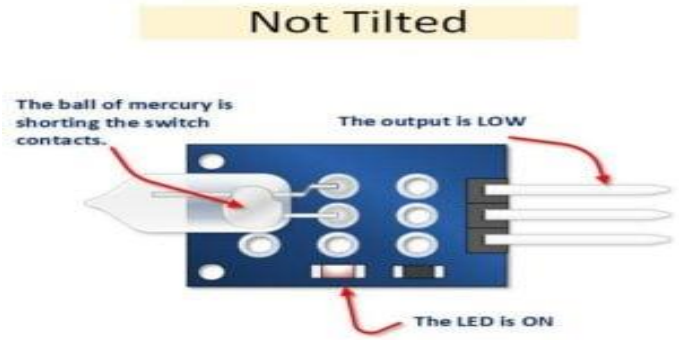
Tilt sensor consists of a mercury ball and two contacts in a sealed enclosure, in which the mercury ball will short the leads based on the orientation.

A blob of mercury is placed in a small glass tube with two metal contacts coming out. When the sensor is held upright, the mercury will make contact with both the terminals and the switch is closed.

When the sensor is tilted in either direction, the mercury goes off contact with the terminals and the switch is open.

Working

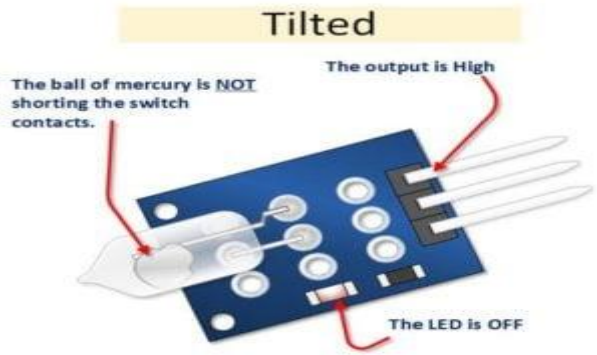
Not Tilted



When the sensor is in "Not Tilted" position, the mercury ball will be at the bottom and shorting the contacts as shown in the image below. This will turn ON the LED and the output will be LOW.

Tilted

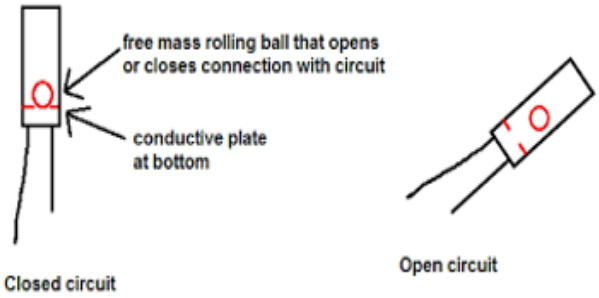
When the sensor is in "Tilted" position, the mercury ball will move away from the contacts as shown in the image below. This will turn OFF the LED and the output will be HIGH.



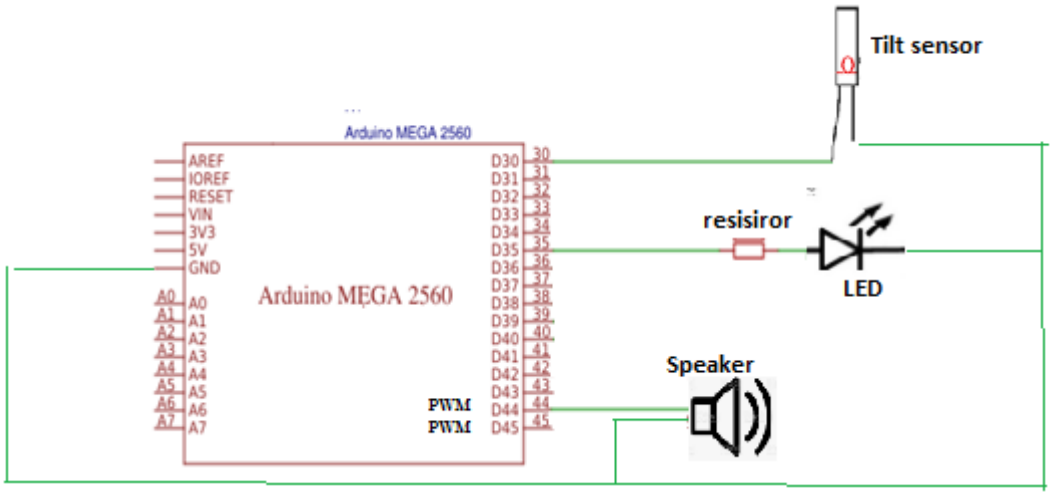
2) Roller Ball Sensor

In roller ball based tilt sensors, one or two metal balls are used to close or open the switch. When the sensor is positioned upright, the metal ball makes contact with both the terminals and closes the switch. When the orientation of the sensor is changed i.e. tilted at an angle, the metal ball loses contact with the terminals and the switch is open.

The working of the Tilt sensor is illustrated in the figure.



Circuit Design of Arduino Tilt Sensor



A Tilt sensor is similar to a normal switch except that the current flows through it only when it is tilted at a certain angle. Hence, a tilt sensor is used to detect the tilt or orientation of an object.

To **connect a Tilt sensor with the Arduino**, it requires 5v dc input to operate. That 5v is supplied using Arduino and the output of Tilt sensor is given at PIN 30 of the Arduino. LED is connected with the PIN 35 of the Arduino with 220-ohm resistor to limit the current to a safe value. And, the buzzer is directly connected to the PIN 44 of the Arduino.

- As mentioned earlier, a tilt sensor is basically a switch. One end or terminal of the tilt sensor is connected to any of the digital I/O pins of Arduino.
- In this project, it is connected to pin 30 of Arduino. The other terminal of the sensor is connected to ground.

Explanation

- Pin 30 is set as input to get input data from the Tilt sensor.
    - When the tilt sensor is inclined beyond a particular angle, it's output goes HIGH. Which is read continuously in the infinite loop.
    - When the pin 30 is read HIGH, it is programmed to turn ON and OFF both LED and Buzzer alternatively with some delay.
- Pin 35 and pin 44 are defined as output pins for LED and Buzzer respectively.

A buzzer and an LED are used to indicate the detection of the tilt by Arduino.

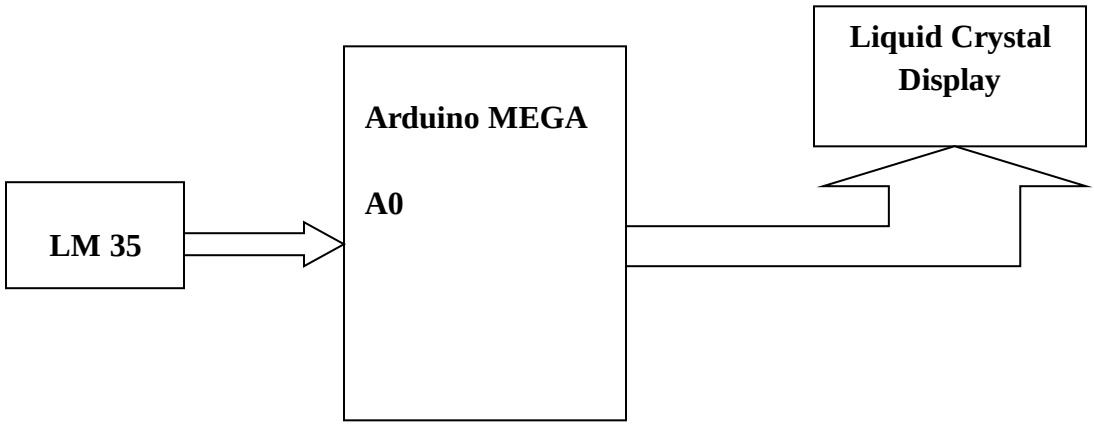
- The buzzer is controlled by the PWM output of the Arduino to generate different tones. Hence, positive terminal of the buzzer is connected to any of the PWM pins of the Arduino. In this demonstration, it is connected to pin 44. The other terminal of the buzzer is connected to ground.
- LED is also used to indicate the tilt action. As the output current from Arduino is only 20mA, we are connecting the LED directly to Arduino without any current limiting resistor.It is advised to use a current limiting resistor just to be safe. Anode of the LED is connected to pin 35 of the Arduino while the cathode is connected to ground.

Application

- Tilt sensors can be used in security systems, where the orientation of the object is used as the security measure.
- Tilt sensors are often used in gaming consoles and mobile phones.
- They are used in navigation systems of boats, airplanes etc., to determine the pitch and roll.

Digital Thermometer

Thermometers are useful apparatus for temperature measurement. Figure below shows the block diagram of a digital thermometer using Arduino Mega.

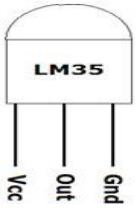


We can divide this **Arduino based thermometer** into three sections - The first section senses the temperature by using temperature sensor LM35, second section converts the temperature value into a suitable numbers in Celsius scale which is done by Arduino, and last part of system displays temperature on 16x2 LCD.

LM35 is an analog, linear temperature sensor whose output voltage varies linearly with change in temperature. LM35 is three terminal linear temperature sensor from National semiconductors. It can measure temperature from-***55 degree celsius to +150 degree celsius.*** The voltage output of the LM35 increases 10mV per degree Celsius rise in temperature. LM35 can be operated from a 5V supply.

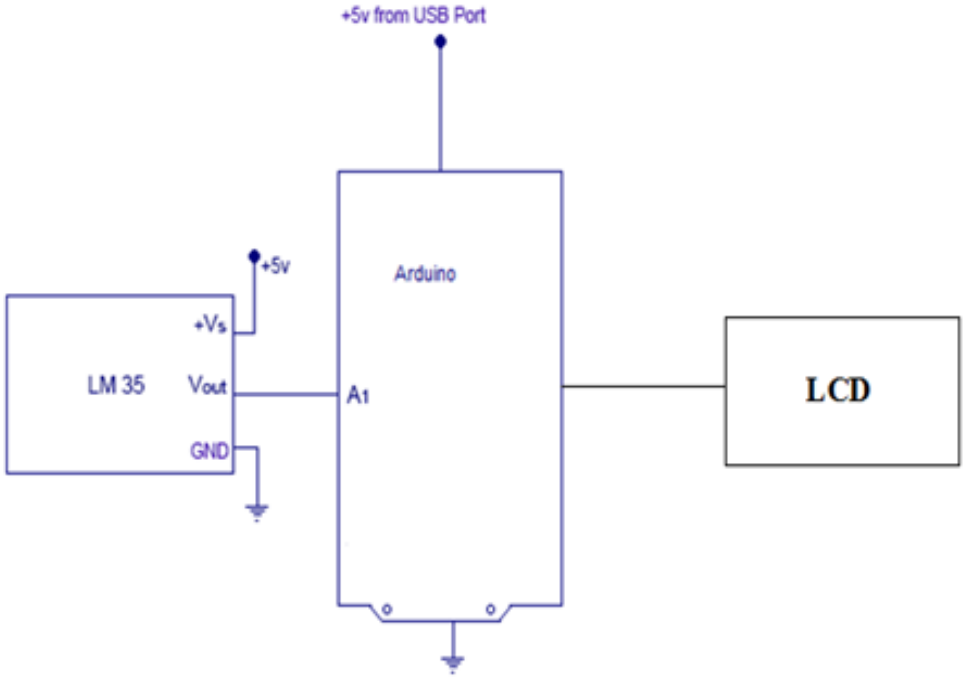
LM35 Temperature Sensor

LM35 is a 3 pin temperature sensor which gives 1 degree Celsius on every 10mVolt change. This sensor can sense up to 150 degree Celsius temperature. 1 number pin of lm35 sensor is Vcc, second is output and third one is Ground.



Pin No	Function	Name
1	Supply voltage; 5V (+35V to -2V)	Vcc
2	Output voltage (+6V to -1V)	Output
3	Ground (0V)	Ground

In this **digital temperature sensor with Arduino**, Arduino MEGA is used to control the whole process. An LM35 temperature sensor is used for sensing environment temperature. Arduino reads output voltage of temperature sensor by using Analog pin A0 and performs the calculation to convert this Analog value to a digital value of current temperature. After calculations arduino sends the temperature value to 16x2 LCD unit by using appropriate commands of LCD.



IoT applications :

Arduino Based Home Automation through Internet of Things(IoT)

Human beings are very lazy creatures, we want to do all of our work by just sitting at a one place and not bothering to go out and socialize. Emerging technology has on the other hand is trying its best to fulfil our demand by everyday introducing something new. One of the emerging and interesting field is Internet of Things. Under Internet of Things we have a small domain called Home automation in which focus is on automating almost every device of our homes. We can make our own security system just by using sensors and doing some code. We can control the devices wirelessly by using Bluetooth or Wi-Fi module. It is all about connecting out devices with the network so that we can control almost anything from anywhere. Unlimited amount of sensors are available in the market and our task is only to deeply understand their functioning and implement them in various areas of concern. One of the examples is of water level sensor, we can use it in our water tanks to measure the level of water and automatically the motor will be turned off when the level exceeds some threshold value.

Internet of Things:

IoT is the network of devices (that are mainly various types of sensors attached to many devices) which permits these sensors to collect and communicate with user and transmit over the network. Due to this we can remotely access or control any of our devices and in interaction between the computer world and the physical world is merging as the new innovation which is helping in increasing efficiency economic benefits and accuracy. We can see how it is efficient, economically beneficial by taking this simple example of turning off the lights automatically or automated taps from which water comes only when hands or any other object is beneath the tap, this way our electricity and water which are very crucial resources are not wasted at all. Smart homes are made by using Internet of Things. We use various sensors at different places and make them connected to the network so they can communicate with each other and there is no need of asking for the inputs and other details from the user.

IoT can be used in the following fields:

- Tracking the surrounding or the environment
- Managing the infrastructure
- Home automation
- Business automation
- Hospitals and other medical stores
- Security systems

• For preserving and conserving resources

Hardware Description :

This hardware implementation contains 4 different parts. (i)A 16X2 LCD display for displaying status of the system and IP address of the local Wi-Fi network (ii) Relay for switching the load automatically, (iii) Arduino as decision maker and (iv) esp8266 version 1 for connecting to local Wi-Fi.

- **16X2 LCD:**

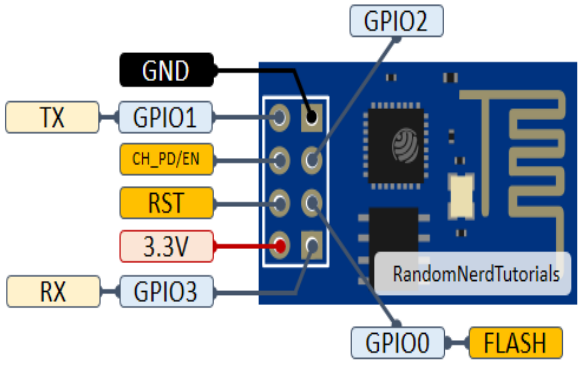
16×2 LCD is used to display 16 characters in two lines. It is easy to interface with Arduino due to its available library. In this project this LCD is used to display the status of the appliances whether it is on or off.

- **4-Channel Relay** :

4-Channel relay is connected to the Arduino Uno and its output is connected to the home appliances in a sequence as (i) fan (ii) light (iii) room-heater and (iv) TV set. Relay takes low current and voltage and triggers the switch which is connected to a high voltage. 4 input pins of relay are connected to Arduino which takes5V supply from it and can trigger up to 10A, 250V supply

- **Wifi Module: (ESP8266-01)**

The ESP8266-01 is a highly compact board, used as a peripheral for any board through serial (RX/TX) and also as a standalone board. The board requires 3.3 V and can be programmed with any FTDI operating at 3.3 V. The pins include power (+3.3 V and GROUND), RX / TX, CH_PD to enable the chip and 2 General Purpose Input Output (GPIO) [14].



ESP8266 is a wifi chip that provides Transfer Control Protocol (TCP) and Internet Protocol (IP). There are different ESP8266 modules available in the market. It has 6 pins and operates on 3.3v.

ESP8266 wifi module to Arduino:

- Module Vcc to 3.3v.

- Module CH_PD to 3.3v.
- Module Ground to Arduino ground.
- Module Tx to Arduino Rx.
- Module Rx to Arduino Tx.

WIFI Wi-Fi(wireless fidelity) is a wireless communication technology which is used here to provide a hotspot through which ESP8266-01 module can connect. The router will assign a unique IP address to the module for establishing a connection between smart phone and ESP8266-01.

System Design

Figure.1 and 2.Gives an idea about the operation of home automation system.

Arduino is perfect for this project as it provides much pins to interface relay module,**16×2 LCD** and ESP8266 wifi module. The four different appliances such as fan, light, room heater and TV are operated remotely using Wi-Fi and through an application installed on android or iPhone. These appliances are connected through Arduino with its digital input/output pins. These devices are connected with local Wi-Fi using a communicating module called esp8266.

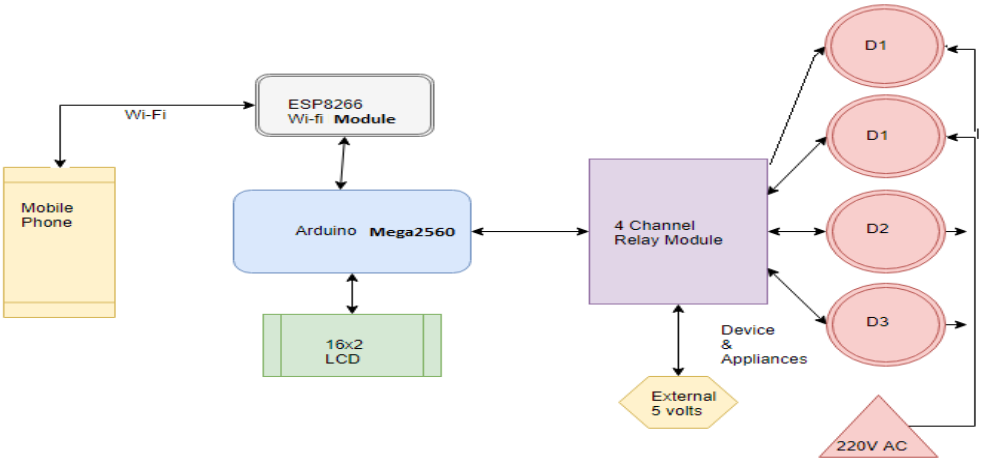
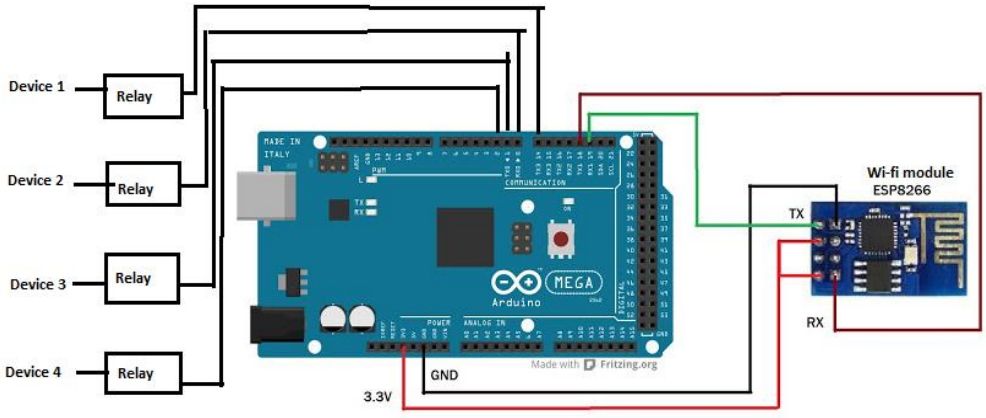


Figure 1

IOT application- Smart Home

Figure 2



Conexion Arduino Mega 2560 + ESP8266