**CHAPTER 3**

**SYSTEM COMPONENTS**

The following block diagram describes system architecture and functionality. Arduino MEGA and motor driver are given by power supply. When a user sends an input to Arduino MEGA, Arduino gives the command to the stepper motor driver. The stepper motor driver operates the stepper motor in forward and reverse direction. Based on this operation, elevator moves up and down. Seven-segment display is used for showing the current position of elevator.



Figure 3.1. Block Diagram of Elevator

**3.1. Hardware Components**

The basic hardware components used in elevator system are as follow:

* Arduino MEGA 2560
* Ultrasonic Sensor
* DC Motor
* 4×3 Keypad
* Stepper motor NEMA 17
* Motor driver L298N
* Seven-Segment Display
* Push-Buttons
* Buzzer
* Inductive Sensor
* Power supply
* Micro Limit Switch

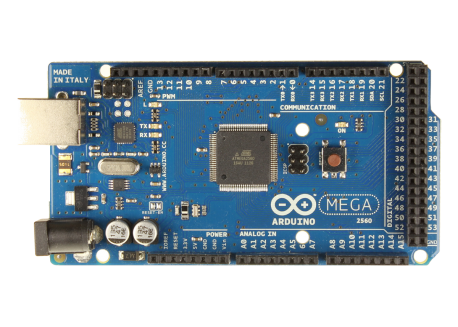
**3.2. Arduino MEGA**

Figure 3.2. Arduino Mega

The Arduino Mega2560 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 suppliedwith 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 Mega2560 differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip.

Figure 2.11. Arduino Mega

Figure 2.11. Arduino Mega

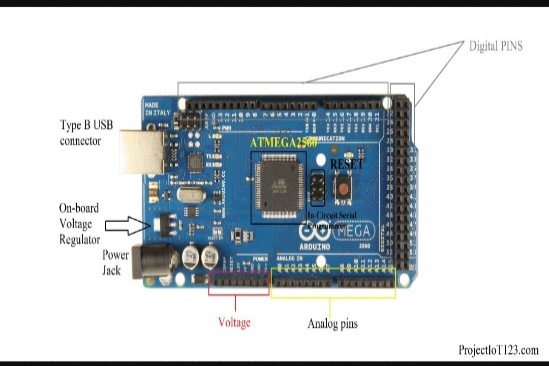


Figure 3.3. Arduino MEGA Pin Diagram

* 5V & 3.3V: This pin is used to provide output regulated voltage around 5V. This regulated power supply powers up the controller and other components on the board. It can be obtained from Vin of the board or USB cable or another regulated 5V voltage supply. While another voltage regulation is provided by 3.3V pin. Maximum power it can draw is 50mA.
* GND: There are 5 ground pins available on the board which makes it useful when more than one ground pins are required for the project.
* Reset: This pin is used to reset the board. Setting this pin to LOW will reset the board.
* Vin: It is the input voltage supplied to the board which ranges from 7V to 20V. The voltage provided by the power jack can be accessed through this pin. However, the output voltage through this pin to the board will be automatically set up to 5V.
* Serial Communication: RXD and TXD are the serial pins used to transmit and receive serial data i.e. Rx represents the transmission of data while TX used to receive data. There are four combinations of these serial pins are used where Serial 0 contains RX (0) and TX (1), Serial 1 contains TX (18) and RX (19), Serial 2 contains TX (16) and RX (17), and Serial 3 contains TX (14) and RX (15).
* External Interrupts: Six pins are used for creating external interrupts: interrupt 0(0), interrupt 1(3), interrupt 2(21), interrupt 3(20), interrupt 4(19), interrupt 5(18). These pins produce interrupts by a number of ways i.e. providing LOW value, rising or falling edge or changing value to the interrupt pins.
* LED: This board comes with built-in LED connected to digital pin 13. HIGH value at this pin will turn the LED on and LOW value will turn it off.
* AREF.: AREF stands for Analog Reference Voltage which is a reference voltage for analog inputs.
* Analog Pins: There are 16 analog pins incorporated on the board labeled as A0 to A15. It is important to note that all these analog pins can be used as digital I/O pins. Each analog pin comes with 10-bit resolution. These pins can measure from ground to 5V. However, the upper value can be changed using AREF and analog Reference () function.
* I2C: Two pins 20 and 21 support I2C communication where 20 represents SDA (Serial Data Line mainly used for holding the data) and 21 represents SCL (Serial Clock Line mainly used for providing data synchronization between the devices)
* SPI Communication: SPI stands for Serial Peripheral Interface used for the transmission of data between the controller and other peripherals components. Four pins i.e. 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS) are used for SPI communication.

Arduino MEGA is comparatively larger than other boards available in the market. It comes 4-inch length and 2.1-inch width. However, USB port and power jack are slightly extended from the given dimensions.

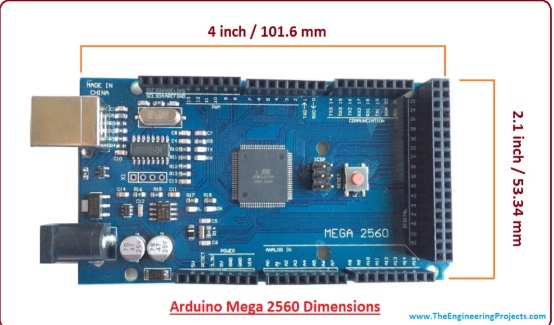


Figure 3.4. Arduino MEGA Dimension

Table 3.1. Arduino MEGA Specification

|  |  |
| --- | --- |
| Microcontroller | ATmega2560 |
| Operating voltage | 5 V |
| Input voltage (recommended) | 7-12 V |
| Input voltage (Limits) | 6-20 V |
| Digital I/O pins | 54 (of which 15 provide PWM output) |
| Analog input pins | 16 |
| DC current per I/O pin | 40 Ma |
| DC current for 3.3V pin | 50 Ma |
| Flash memory | 256 KB of which 8 KB is used by bootloader |
| SRAM | 8 KB |
| EEPROM | 4 KB |
| Clock speed | 16 MHz |

**3.3. Ultrasonic Sensor**

Ultrasonic sensors are self-contained solid-state devices designed for non-contact sensing of solid and liquid objects. For many applications, such as monitoring the level of water in a tank, ultrasonic technology lets a single device to do a job that would otherwise require multiple sensors.

The Ultrasonic Sensor sends out a high-frequency sound pulse and then tomes how long it takes for the echo of the sound to reflect back. The sensor has 2 openings on its front. One opening transmits ultrasonic waves, the other receives them.

The Arduino Ultrasonic Range Detection Sensor is used with Arduino in order to calculate distances from objects. It is an IC that works by sending an ultrasound pulse to echo back, calculating the time taken in microseconds (1microsend=1.0 x 10-6 seconds). It can trigger a pulse as fast as 20 times a second and it can determine objects up to 3 meters away and as near as 3cm. it needs a 5V power supply to run. And then it waits and listens for the pulse to echo back, by calculating the time taken in microseconds. Adding the Arduino Ultrasonic Range Detection Sensor to the Arduino is very easy only 4 pins to worry about. Power, Ground, Trigger and Echo. Since it needs 5V and Arduino provides 5V.

Specification of ultrasonic sensor:

* Voltage: DC 5V
* Current: 15mA
* Frequency: 40Hz
* Max range: 4m
* Min range: 2cm
* Measuring angle: 15 degree
* Trigger input signal: 10µs
* Echo output signal: Input lever signal and the range in proportion
* Dimension: 40×20×15mm

Figure 3.5. Ultrasonic Sensor

3.3.1. Connections of Ultrasonic Sensor and Arduino MEGA

The HC-SR04 Ultrasonic module has 4 pin, ground, VCC, Trig and Echo. The ground and VCC pins of the module needs to be connected to the ground and the 5Volts pins on the Arduino Board respectively and the trig and echo pins to any digital I/O pin on the Arduino Board.

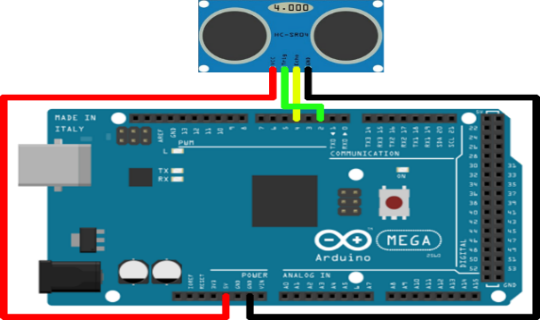


Figure 3.6. Connections of Ultrasonic Sensor and Arduino

**3.4. DC Motor**

A DC motor is any of a class of rotary electrical machines that converts direct current electrical energy into mechanical energy. The most common types relay on the forces produced by magnetic fields. Nearly all types of DC motors have some internal mechanism, either electromechanical or electronic, to periodically change the direction of current flow in part of the motor.

DC motors were the first type widely used, since they could be powered from existing direct-current lighting power distribution systems. A DC motor’s speed can be controlled over a wide range, using either a variable supply voltage or by changing the strength of current in its field windings.

Small DC motors are used in tools, toys and appliances. The universal motor can operate on direct current but is a light weight brushed motor used for portable power tools and appliances. Larger DC motors are used in propulsion of electric vehicles, elevator and hoists or in drives for steel rolling mills. The advent of power electronic has made replacement of DC motors with AC motors possible in many applications.

Table 3.2. Specifications of DC Motor

|  |  |
| --- | --- |
| Characteristic | Specification |
| Nominal Voltage | 4.5V |
| RPM | 15000 |
| Current | 1.58A |
| Torque | 21.8g/cm |

**3.5. 4×3 Keypad**



Figure 3.7. 4×3 Keypad Matrix

This 4x3 matrix keypad has 12 built-in pushbutton contacts connected to row and column lines. A microcontroller can scan these lines for a button pressed state. In the keypad library, the Propeller sets all the column lines to input, and all the row lines to input. Then, it picks a row and sets it high. After that, it checks the column lines one at a time. If the column connection stays low, the button on the row has not been pressed. If it goes high, the microcontroller knows which row and which column.

The keypad library supports pretty much any number of rows and columns. So, the program has to tell it, keypad has four rows and four columns, which i/o pins the lines are connected to and what value each button represents. The rows array will be used to tell the keypad library that the top row is connected to P7, the second row to P6 and so on. Likewise, the columns array list the leftmost column as connected to P3, the next over connected to P2 and so on. The values array stores the value that the program to give us for each button press.

3.5.1. Matrix Keypad Interface Logic

Initially, all switches are assumed to be released. So there is no connection between the rows and columns. When any one of the switches are pressed, the corresponding row and column are connected. This will drive that column pin low. Using this logic, the button press can be detected. The colors red and black are for logic high and low respectively. There are the steps involved in determining the key that was pressed.

* Step 1: The first step involved in interfacing the matrix keypad is to write all logic 0’s to the matrix keypad is to write all logic 1’s to the columns. In the image, black line symbolizes logic 0 and red line symbolizes logic 1. The circled key is pressed and see how the key press can be detected by a software routine.
* Step 2: Now the software has to scan the pins connected to columns of the keypad. If it detects a logic 0 in any one of the columns, then a key press was made in that columns, then a key press was made in that column. This is because the event of the switch press shorts the C2 line with R2. Hence C2 is driven low.
* Step 3: Once the column corresponding to the key pressed is located, the next thing that the software has to do is to start writing logic 1’s to the rows sequentially and check if C2 becomes high. The logic is that if a button in that row was pressed, then the value written to that row will be reflected in the corresponding column (C2) as they are short circuited.
* Step 4: The procedure is followed till C2 goes high when logic high is written to a row. In this case, the logic high to the second row will be reflected in the second column.

Specification of keypad:

* Interface: 7 pins access to 4×3 matrix
* Operating temperature: o to 60℃
* Cable length: 3.3 inch
* Keypad Width: 2.7 inch
* Keypad length: 3 inch
* Number of Keys: 12

3.5.2. Connections and Operations

There is a switch connecting each row and column. The entire switch is open (not connected), when the pressed either one buttons, the switch is now closed (connected). As that see, now there is a connection between the row and column. So it is connecting the first 4 pins to the column as input. The other 4 pins are connected to the row as output. The input means that is the input to the microcontroller while output is from the microcontroller. Node that the input to the microcontroller has to connect to pull high resistor or that can use the internal pull up from Arduino itself.

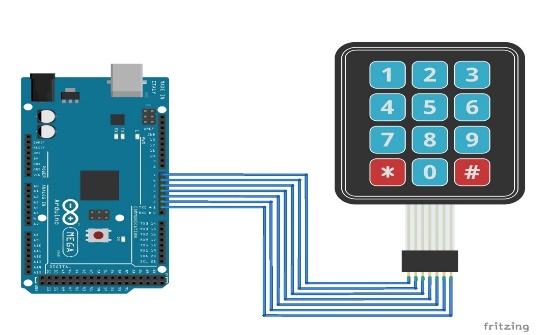


Figure 3.8. 4x3 matrix keypad withArduino

The microcontroller sends LOW to each row one at a time and check whether there is a low signal detected on the column. If there is no button pressed, the microcontroller will scan for the next row and read for LOW signal. Since the column is pull HIGH internally, so no button pressed with return 1to the microcontroller. When press the button, now the row and column are connected. The 0 from row would make the column return a 0 to the microcontroller. Finally, the software part is reached.

**3.6. Stepper Motor**

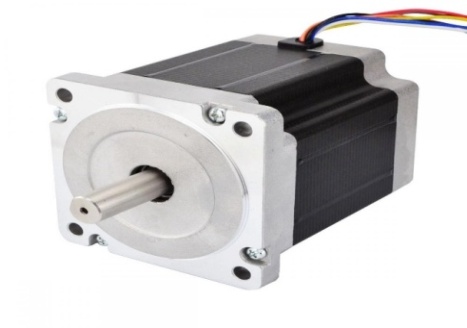


Figure 3.9. Stepper motor

A stepper motor is a brushless DC electric motor that divides a full rotation into a number of equal steps. This is very useful because it can be precisely positioned without any feedback sensor, which represents an open-loop controller. The stepper motor consists of a rotor that is generally a permanent magnet and it is surrounded by the windings of the stator.

Stepper motors are DC motors that move in discrete steps. They have multiple coils that are organized in groups called ‘phases’. By energizing each phase in sequence, the motor will rotate one step at a time. With a computer controlled stepping the user can achieve very precise positioning and/or speed control. The speed of the motor is controlled by the frequency of the pulses. The stepper motor is used for precise positioning with a motor, such as hard disk drives, robotics, antennas, telescopes, and some toys. Stepper motors cannot run at high speeds, but have a high holding torque.

Specifications:

* Motor type: NEMA 17
* Size: 42.3 mm square ×48 mm, not including the shaft
* Weight: 350g
* Shaft diameter: 5 mm
* Steps per revolution: 200
* Current rating: 1.2 A per coil
* Voltage rating: 4 V
* Resistance: 3.3Ω per coil
* Inductance: 2.8 mH per coil
* Lead length: 30 cm

**3.7. L298N Motor Driver**

The L298N is a dual H-Bridge motor driver which allows speed and direction control of two DC motors at the same time. The module can drive DC motors that have voltages between 5 and 35V, with a peak current up to 2A. Let’s take a closer look at the pinout of L298N module and explain how it works. The module has two screw terminal blocks for the motor A and B, and another screw terminal block for the Ground pin, the VCC for motor and a 5V pin which can either be an input or output. This depends on the voltage used at the motors VCC. The module has an onboard 5V regulator which is either enabled or disabled using a jumper. If the motor supply voltage is up to 12V we can enable the 5V regulator and the 5V pin can be used as output, for example for powering our Arduino board. But if the motor voltage is greater than 12V the user must disconnect the jumper because those voltages will cause damage to the onboard 5V regulator. In this case the 5V pin will be used as input as the user needs to connect it to a 5V power supply in order the IC to work properly.

The user can note here that this IC makes a voltage drop of about 2V. So for example, if the user uses a 12V power supply, the voltage at motors terminals will be about 10V, which means that we won’t be able to get the maximum speed out of our 12V DC motor.

Next are the logic control inputs. The Enable A and Enable B pins are used for enabling and controlling the speed of the motor. If a jumper is present on this pin, the motor will be enabled and work at maximum speed, and if the user removes the jumper the user can connect a PWM input to this pin and in that way control the speed of the motor. If we connect this pin to a Ground the motor will be disabled.

Next, the Input 1 and Input 2 pins are used for controlling the rotation direction of the motor A, and the inputs 3 and 4 for the motor B. Using these pins, the user actually controls the switches of the H-Bridge inside the L298N IC. If input 1 is LOW and input 2 is HIGH the motor will move forward, and vice versa, if input 1 is HIGH and input 2 is LOW the motor will move backward. In case both inputs are same, either LOW or HIGH the motor will stop. The same applies for the inputs 3 and 4 and the motor B.

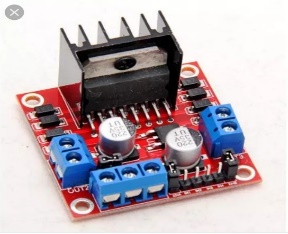


Figure 3.10. L298N Motor Driver

**3.8. Seven-Segment Display**

A Seven Segment Display (SSD) is one of the most common, cheap and simple to use display. A Seven-Segment Display or Seven-Segment Indicator is a form of electronic display device for displaying decimal numerical that is an alternative to the more complex dot matrix displays. Seven-Segment Displays are widely used in digital clocks, electronic meters, basic calculators and other electronic devices that display numerical information. Seven-Segment Display used to show the location of car in the elevator control system.

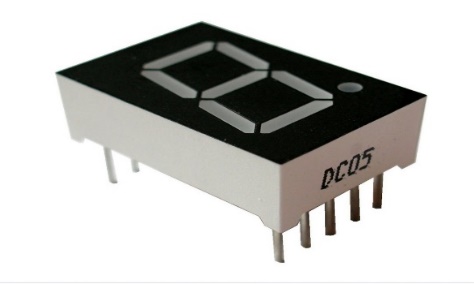


Figure 3.11. Seven-Segment Display

3.8.1. Types of Seven-Segment Displays

There are two types of Seven-Segment Display. They are:

* Common cathode display
* Common anode display

3.8.1.1. Common cathode

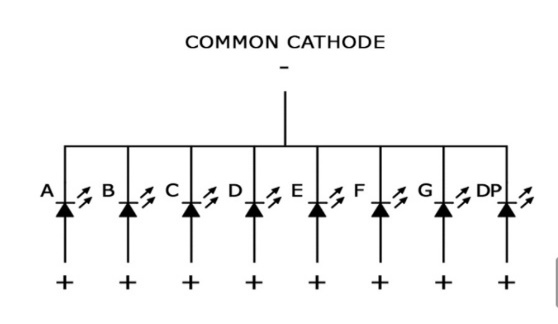


Figure 3.13. Common cathode

A common cathode Seven-Segment display is simpler to use in electronic field. For the simple use of a BJT as a switch, the emitter-collector junctions get shorted when there is an input signal at the base terminal, else it remains cut off. The input can be given through a suitable resistor.

Common cathode, the negative terminal of all the LEDs is commonly connected to the ‘COM’ pin. A segment can be lighted up when ‘1’ is given to the respective LED segment and ground is connected to the common. The internals are given as below:

3.8.1.2. Common Anode

Common anode, the positive terminal of all the LEDs is commonly connected to the ‘COM’ port of the SSD. A segment can be lighted up when the ‘COM’ port is connected to the positive battery supply, and ground is given to the respective segment.

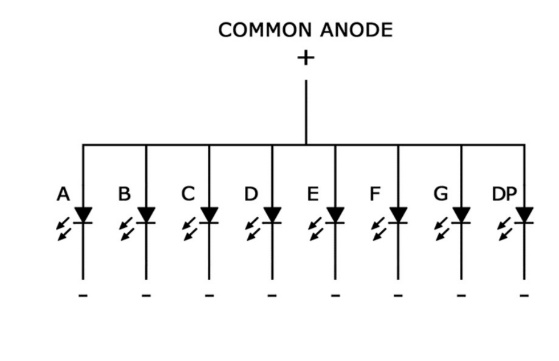
****

Figure 3.14. Common Anode

**3.9. Push-button**

Push-buttons are used as call buttons. A push-button is a device that provides control of equipment by pressing a button. The contacts of a pushbutton are generally of the double break types. Two sets of contacts, via, Normally Opened (NO) another Normally Closed (NC) are provided in a pushbutton. When the pushbutton is pressed by applying force, the NO contact closes and NC contact opens.

A pushbutton or simply button is a simple switch mechanism for controlling some aspect of a machine or process. Buttons are typically made out of hard material, usually plastic or metal. The surface is usually flat or shaped to accommodate the human finger or hand, so as to be easily depressed or pushed. Buttons are most often biased switches, though even many unbiased buttons (due to their physical nature) require a spring to return to their unpushed state. Different people use different terms for the pushing of the button such as press, depress, mash and punch.



Figure 3.15. Push-button

**3.10. Buzzer**

A buzzer is a small yet efficient component to add sound features to our project/system. It is very small and compact two pins structure can be easily used on breadboard and even on PCBs which makes this a widely used component in most electronic applications.

3.10.1. Buzzer features and specifications

* Rated Voltage: 6V DC
* Operating Voltage: 4-8V DC
* Rated Current: <30 mA
* Sound Type: Continuous Beep
* Resonant Frequency: ~2300Hz
* Operating Temperature: -25℃ to +80℃
* Storage Temperature: -30℃ to +85℃
* Weight: 2g
* Diameter: 11.78mm



Figure 3.16. Buzzer

**3.11. Inductive Sensor**

An inductive sensor is a device that uses the principle of electromagnetic induction to detect or measure objects. An inductor develops a magnetic field when a current flow through it; alternatively, a current will flow through a circuit containing an inductor when the magnetic field through it changes. This effect can be used to detect metallic objects that interact with a magnetic field. Non-metallic substances such as liquids or some kinds of dirt do not interact with magnetic field, so an inductive sensor can operate in wet or dirty conditions.

3.11.1. Features and specification

* Power: 6V to 36V DC
* Output Current: 200mA
* Detection Range: 4mm
* Detected Material: Iron/ Steel Alloys
* Output Pin: NPN 3-wires Normally Open
* Wire Length: 1.2 meter
* Dimensions: 2.2×6cm
* Net Weight: 85g



Figure 3.17. Inductive Sensor

**3.12. Micro Limit Switch**

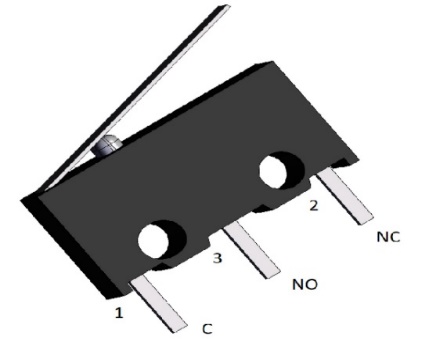


Figure 3.18. Micro Limit Switch

Micro limit switch is often used in different types of control circuits. Micro limit switches are much smaller in size than an ordinary limit switch. This permits them to be used in small spaces that would not be accessible to the larger device. The limit switch is like 1 NO + 1 NC push button. The push button is actuated by hand whereas the limit switch is operated mechanically. The main function of the limit switch is to open or close an electric circuit when the physical limit of the operation of the control device has reached.

**3.13. Power Supply**

The power supply is the first requirement for the project. Figure shows the power supply used for elevator control system.

There is a switch, choose the proper voltage before use. Features of the power supply are the following:

* AC input 110 and 220 V by switch
* Protection: Short circuit, overload, over voltage and over temperature
* Low cost, high reliability
* Two years warranty

****

Figure 3.19. Power Supply