INTRODUCTION:

Liquefied petroleum gas is being used for the past decades as industrial fuel and for domestic purpose. It has a characteristic of smokeless burning in the air. The main constituents of LPG are propane and butane and depending on the applications their proportions vary. Gas leakage detection in residential houses has become one of the fundamental issues in the recent times. Accidents mainly occur due to the negligence and technical fault. Electronic and press media have reported many accidents which were caused mainly because of gas leakage in residential houses and industries. A better system needs to be developed to reduce the accidents because of gas leakage. The gas is generally stored in metallic cylinders as its boiling point is lower than ambient temperature. Gas is molecularly heavy than other gases present in the air. So whenever the gas is leaked it settles closest to the ground level. And unless you provide a powerful exhaust system it cannot be forcefully disposed into open atmosphere.

Now-a-days LPG leakage detection in homes, restaurants has been a common issue and the detection systems find applications in the market. Presently they are using load cell to measure the weight of the cylinder. When they find it become empty, consumer will order for a new cylinder. There may be a delay in providing the cylinder for few reasons like we may inform the service provider at the last moment when the gas is empty or there may be a delay in informing the gas provider. So in this system we will use a pressure sensor to measure the amount of gas present in the cylinder and also book the gas automatically when it reaches to a certain level. LPG is generally odourless and cannot be detected by human sense of smell because of its odourless nature. A pungent chemical is added to it purposely so that humans can detect the gas. There are few disadvantages anyhow. Firstly, it requires human presence in the vicinity. Secondly, by the time gas leakage is detected, its concentration in the vicinity may exceed the threshold level and may lead to explosion with the spark like light switch. Soin order to monitor its presence, sensing systems are deployed in the premises to detect the leakage and avoid accidents. Many sensors are available in the market which can warn the gas leakage. They make use of transmitters, controllers and other accessories but the cost of these kind of sensing systems is high and has technical complexity and also inaccurate with delays. Therefore, there is a need for the development of lower complexity, low cost and fast response systems.

In the proposed system, we will detect the gas and also alerts the consumer using the SMS. It Contains the door,window unlocking mechanism and switching ON the exhaust fan.

**Arduino MEGA 2560:**

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. The Mega is compatible with most shields designed for the Arduino Duemilanove or Diecimila.

---------------------------------Technical Specifications-------------------------------

Microcontroller- ATmega2560

Operating Voltage- 5V

Input Voltage (recommended)- 7-12V

Input Voltage (limits)- 6-20V

Digital I/O Pins- 54 (of which 14 provide PWM output)

Analog Input Pins- 16

DC Current per I/O Pin- 40 mA

DC Current for 3.3V Pin- 50 mA

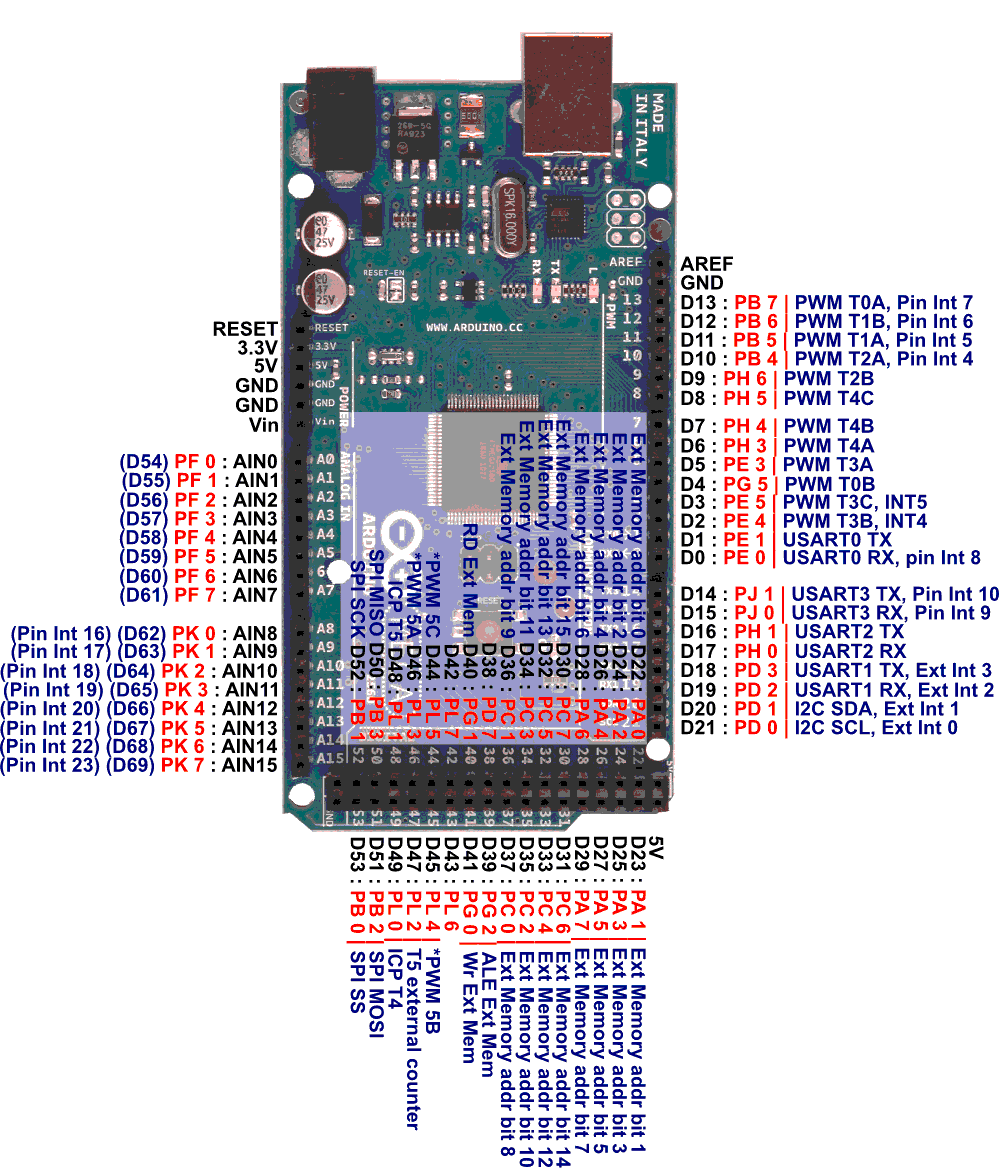
Flash Memory- 256 KB of which 8 KB used by bootloader

SRAM- 8 KB

EEPROM- 4 KB

Clock Speed- 16 MHz

PIN CONFIGURATION:



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 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 Mega 2560 differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter.

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.

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 (which can be read and written with the EEPROM library).

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 (disconnected by default) of 20-50 kOhms. 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. Pins 0 and 1 are also connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip .

• 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. See the attachInterrupt() function for details.

• 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, which, although provided by the underlying hardware, is not currently included in the Arduino language. The SPI pins are also broken out on the ICSP header, which is physically compatible with the Duemilanove and Diecimila.

• 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 (documentation on the Wiring website). Note that these pins are not in the same location as the I2C pins on the Duemilanove.

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.

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 TTL (5V) serial communication. An ATmega8U2 on the board channels one of these over USB and provides a virtual com port to software on the computer (Windows machines will need a .inf file, but OSX and Linux machines will recognize the board as a COM port automatically. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the ATmega8U2 chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A SoftwareSerial library allows for serial communication on any of the Mega's digital pins.

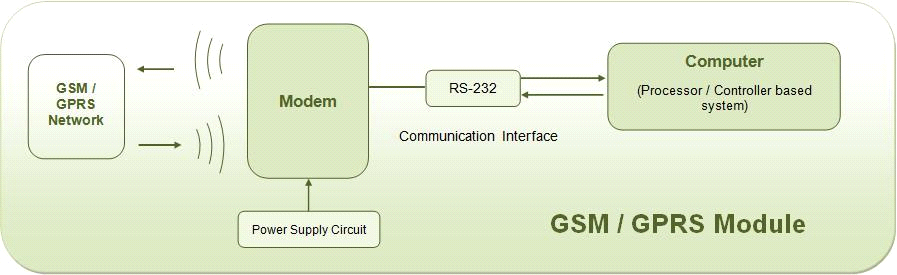
The ATmega2560 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the documentation on the Wiring website for details. To use the SPI communication, please see the ATmega2560 datasheet.

The Arduino Mega2560 can be programmed with the Arduino software (download). For details, see the reference and tutorials.

The Atmega2560 on the Arduino Mega comes preburned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer.

**GSM MODULE**

GSM module is used to establish communication between a computer and a GSM system. Global System for Mobile communication (GSM) is an architecture used for mobile communication in most of the countries. GSM module consists of a GSM modem assembled together with power supply circuit and communication interfaces (like RS-232, USB, etc) for computer. The MODEM is the soul of such modules.



**GSM MODEM:**

GSM MODEM is a class of wireless MODEM devices that are designed for communication of a computer with the GSM network. It requires a SIM (Subscriber Identity Module) card just like mobile phones to activate communication with the network. Also they have IMEI (International Mobile Equipment Identity) number similar to mobile phones for their identification. A GSM MODEM can perform the following operations:

1.      Receive, send or delete SMS messages in a SIM.

2.      Read, add, search phonebook entries of the SIM.

3.      Make, Receive, or reject a voice call.

The MODEM needs AT commands, for interacting with processor or controller, which are communicated through serial communication. These commands are sent by the controller/processor. The MODEM sends back a result after it receives a command. Different AT commands supported by the MODEM can be sent by the processor/controller/computer to interact with the GSM cellular network.



**LOAD CELL:(40kg)**

A **load cell** is a [transducer](/wiki/Transducer) that is used to create an [electrical signal](/wiki/Electrical_signal) whose magnitude is directly proportional to the [force](/wiki/Force) being measured. The various load cell types include hydraulic, pneumatic, and strain gauge.

Strain gauge load cells are the most common in industry. These load cells are particularly stiff, have very good resonance values, and tend to have long life cycles in application. Strain gauge load cells work on the principle that the [strain gauge](/wiki/Strain_gauge) (a planar resistor) deforms when the material of the load cells deforms appropriately. Deformation of the strain gauge changes its electrical resistance, by an amount that is proportional to the strain. The change in resistance of the strain gauge provides an electrical value change that is calibrated to the load placed on the load cell.

A load cell usually consists of four strain gauges in a [Wheatstone bridge](/wiki/Wheatstone_bridge) configuration. Load cells of one strain gauge (quarter bridge) or two strain gauges (half bridge) are also available. The electrical signal output is typically in the order of a few millivolts (mV) and requires amplification by an [instrumentation amplifier](/wiki/Instrumentation_amplifier) before it can be used. The output of the transducer can be scaled to calculate the force applied to the transducer. Sometimes a high resolution [ADC](/wiki/Analog-to-digital_converter), typically 24-bit, can be used directly.

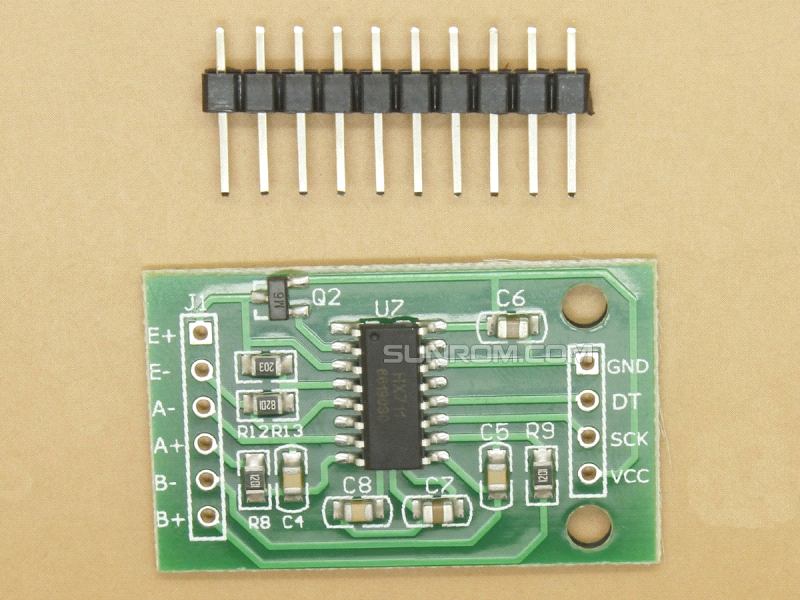
The gauges themselves are bonded onto a beam or structural member that deforms when weight is applied. In most cases, four strain gauges are used to obtain maximum sensitivity and temperature compensation. Two of the gauges are usually in tension can be represented as T1 and T2,and two in compression can be represented as C1 and C2, and are wired with compensation adjustments. The strain gauge load cell is fundamentally a spring optimized for strain measurement. Gauges are mounted in areas that exhibit strain in compression or tension. When weight is applied to the load cell, gauges C1 and C2 compress decreasing their resistances. Simultaneously, gauges T1 and T2 are stretched increasing their resistances. The change in resistances causes more current to flow through C1 and C2 and less current to flow through T1 and T2. Thus a potential difference is felt between the output or signal leads of the load cell. The gauges are mounted in a differential bridge to enhance measurement accuracy. When weight is applied, the strain changes the electrical resistance of the gauges in proportion to the load. Other load cells are fading into obscurity, as strain gauge load cells continue to increase their accuracy and lower their unit costs.

Specifications of YZC-1B LOAD CELL-

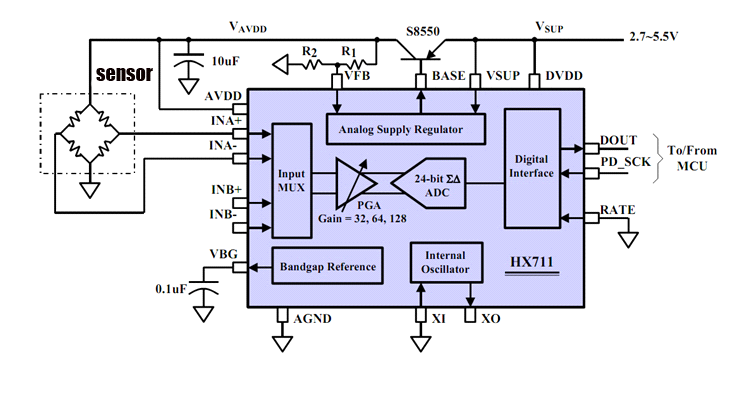
|  |  |
| --- | --- |
| Application | Price computing scales, bench scale etc. |
| Model | YZC-1B |
| Capacity (kg) | 2,3,5,6,8,10,30,35,40,50 |
| Accuracy class | C2 |
| Input resistance Ω | 402±6 |
| Output resistance Ω | 350±3 |
| Total error %F.S | ≤±0.030 |
| Insulation resistance MΩ | 5000 |
| Rated output MV/V | 2.0±0.15 |
| Excitation voltage V | 10~15 |
| Compensated temperature range | -10~+40 |
| Operating temperature range | -35~+80 |
| Temperature effect on zero %F.S/ | 0.003 |
| Creep %F.S/30min | 0.03 |
| Temperature effect on sensitivity %F.S/ | 0.0016 |
| Zero output %F.S | ±1.0 |
| Safe overload %F.S | 150 |
| Load cell material | Aluminum |
| Connecting cable | 4.2\*350mm |
| Method of connecting wire | Red input(+) back input(-) blue output(+) white output(-) |

Size(in mm,1mm=0.03937inches)

HX711 ADC**:**



Based on Avia Semiconductor’s patented technology, HX711 is a precision 24-bit analog- to-digital converter (ADC) designed for weigh scales and industrial control applications to interface directly with a bridge sensor. The input multiplexer selects either Channel A or B differential input to the low-noise programmable gain amplifier (PGA). Channel A can be programmed with a gain of 128 or 64, corresponding to a full-scale differential input voltage of ±20mV or ±40mV respectively, when a 5V supply is connected to AVDD analog power supply pin. Channel B has a fixed gain of 32. On- chip power supply regulator eliminates the need for an external supply regulator to provide analog power for the ADC and the sensor. Clock input is flexible. It can be from an external clock source, a crystal, or the on-chip oscillator that does not require any external component. On-chip power- on-reset circuitry simplifies digital interface initialization. There is no programming needed for the internal registers. All controls to the HX711 are through the pins.

Board Schematic-

MQ6 GAS SENSOR**:**

MQ6 Gas Sensor is used to detect the presence of a dangerous LPG leak in your Home, car or at a service station, storage tank environment. The sensor has excellent sensitivity combined with a good response time. The sensor’s conductivity is more higher along with the gas concentration rising.The sensor can also sense iso-butane, propane, LPG and cigarette smoke. The sensor could be used to detect different combustible gas, especially Methane. Features of MQ6 LPG GAS SENSOR-

* High Sensitivity to LPG, iso-butane, propane
* Small sensitivity to alcohol, smoke
* Good sensitivity to Combustible gas in wide range
* High sensitivity to Propane, Butane, LPG and also response to Natural gas
* Long life and low cost
* Simple drive circuit

Specification of MQ6 LPG GAS Sensor

* Detection Range: 100 - 10,000 ppm iso-butane propane
* Fast Response Time: <10s
* Simple drive circuit
* Heater Voltage: 5.0V
* Dimensions: 18mm Diameter, 17mm High excluding pins, Pins - 6mm High



DC MOTOR**:**

An electric motor is a machine which converts electrical energy into mechanical energy. It is based on the principle that when a current-carrying conductor is placed in a magnetic field, it experiences a mechanical force whose direction is given by [Fleming's Left-hand rule](http://www.ncert.nic.in/html/learning_basket/electricity/electricity/machine/instructions_for_motor.htm) and whose magnitude is given by

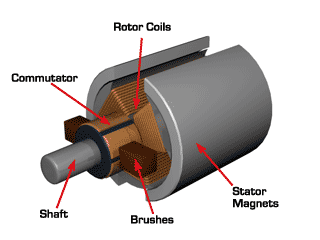
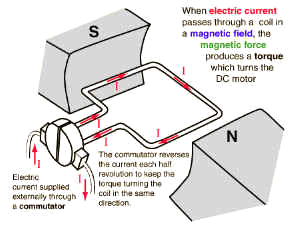
                     Force, F = B I l newton

 Where, B is the magnetic field in weber/m2.

 I is the current in amperes and

  l is the length of the coil in meter.

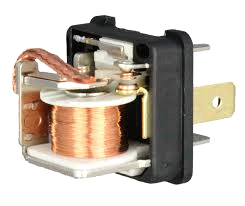
The force, current and the magnetic field are all in different directions.



RELAY

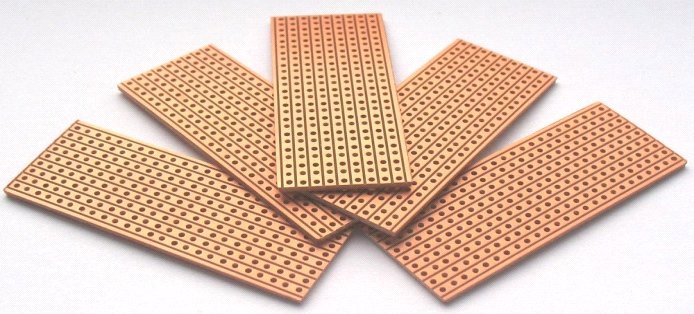
A relay is a simple electromechanical switch made up of an [electromagnet](http://science.howstuffworks.com/electromagnet.htm) and a set of contacts. Relays are found hidden in all sorts of devices. In fact, some of the first computers ever built used relays to implement [Boolean gates](http://computer.howstuffworks.com/boolean.htm).

Relays are switches that open and close circuits electromechanically or electronically. Relays control one electrical circuit by opening and closing contacts in another circuit. When a relay contact is Normally Closed (NC), there is a closed contact when the relay is not energized**.**



**VERO BOARD**

Vero board is a brand of stripboard, a pre-formed circuit board material of copper strips on an insulating bonded paper board.  It was introduced as a general-purpose material for use in constructing [electronic circuits](https://en.wikipedia.org/wiki/Electronic_circuits) - differing from purpose-designed [printed circuit boards](https://en.wikipedia.org/wiki/Printed_circuit_boards) (PCBs) in that a variety of electronics circuits may be constructed using a standard wiring board.

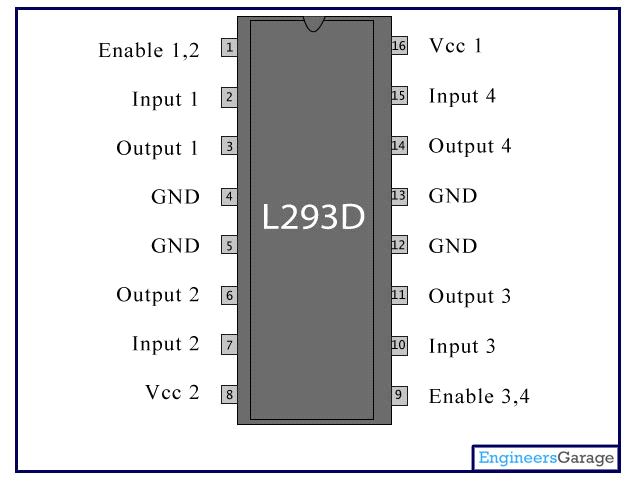


**MOTOR DRIVER:**

L293D is a dual [H-bridge](http://www.engineersgarage.com/electronic-circuits/h-bridge-motor-control) motor driver integrated circuit (IC). Motor drivers act as current amplifiers since they take a low-current control signal and provide a higher-current signal. This higher current signal is used to drive the motors.

L293D contains two inbuilt H-bridge driver circuits. In its common mode of operation, two DC motors can be driven simultaneously, both in forward and reverse direction. The motor operations of two motors can be controlled by input logic at pins 2 & 7 and 10 & 15. Input logic 00 or 11 will stop the corresponding motor. Logic 01 and 10 will rotate it in clockwise and anticlockwise directions, respectively.

Enable pins 1 and 9 (corresponding to the two motors) must be high for motors to start operating. When an enable input is high, the associated driver gets enabled. As a result, the outputs become active and work in phase with their inputs. Similarly, when the enable input is low, that driver is disabled, and their outputs are off and in the high-impedance state.



**POTENTIOMETER:**

A **potentiometer**, informally a **pot**, is a three-[terminal](https://en.wikipedia.org/wiki/Terminal_%28electronics%29) [resistor](https://en.wikipedia.org/wiki/Resistor) with a sliding or rotating contact that forms an adjustable [voltage divider](https://en.wikipedia.org/wiki/Voltage_divider).[[1]](https://en.wikipedia.org/wiki/Potentiometer) If only two terminals are used, one end and the wiper, it acts as a ***variable resistor*** or ***rheostat***.

The measuring instrument called a [potentiometer](https://en.wikipedia.org/wiki/Potentiometer_%28measuring_instrument%29) is essentially a [voltage divider](https://en.wikipedia.org/wiki/Voltage_divider) used for measuring [electric potential](https://en.wikipedia.org/wiki/Electric_potential) (voltage); the component is an implementation of the same principle, hence its name.

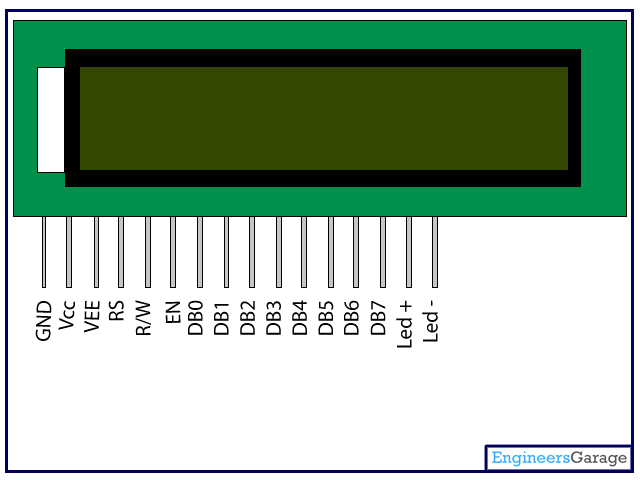
Potentiometers are commonly used to control electrical devices such as volume controls on audio equipment. Potentiometers operated by a mechanism can be used as position [transducers](https://en.wikipedia.org/wiki/Transducer), for example, in a [joystick](https://en.wikipedia.org/wiki/Joystick). Potentiometers are rarely used to directly control significant power (more than a [watt](https://en.wikipedia.org/wiki/Watt)), since the power dissipated in the potentiometer would be comparable to the power in the controlled load.

**LCD (16\*2):**

LCD (Liquid Crystal Display) screen is an electronic display module and find a wide range of applications. A 16x2 LCD display is very basic module and is very commonly used in various devices and circuits. These modules are preferred over [seven segments](http://www.engineersgarage.com/content/seven-segment-display) and other multi segment [LED](http://www.engineersgarage.com/content/led)s. The reasons being: LCDs are economical; easily programmable; have no limitation of displaying special & even [custom characters](http://www.engineersgarage.com/microcontroller/8051projects/create-custom-characters-LCD-AT89C51) (unlike in seven segments), [animations](http://www.engineersgarage.com/microcontroller/8051projects/display-custom-animations-LCD-AT89C51) and so on.

A **16x2 LCD** means it can display 16 characters per line and there are 2 such lines. In this LCD each character is displayed in 5x7 pixel matrix. This LCD has two registers, namely, Command and Data.

The command register stores the command instructions given to the LCD. A command is an instruction given to LCD to do a predefined task like initializing it, clearing its screen, setting the cursor position, controlling display etc. The data register stores the data to be displayed on the LCD. The data is the ASCII value of the character to be displayed on the LCD. Click to learn more about internal structure of a [LCD](http://www.engineersgarage.com/insight/how-lcd-works).



BUZZER

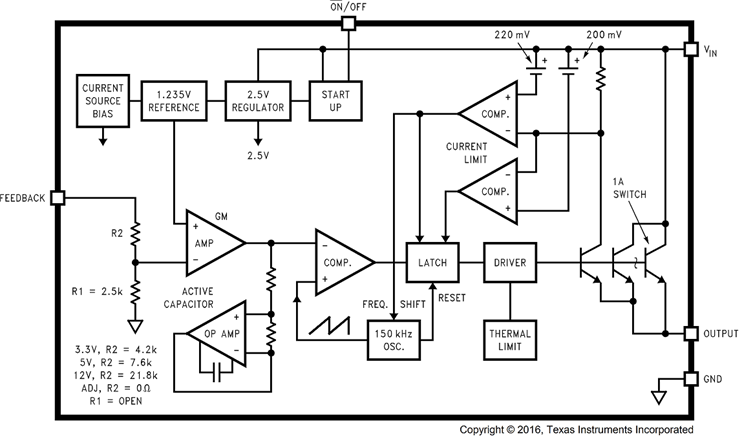
A **buzzer** or **beeper** is an [audio](https://en.wikipedia.org/wiki/Sound) signaling device, Typical uses of buzzers and beepers include [alarm devices](https://en.wikipedia.org/wiki/Alarm_devices), [timers](https://en.wikipedia.org/wiki/Timer), and confirmation of user input such as a mouse click or keystroke.



LM2596 BUCK CONVERTER

The LM2596 regulator is an easy-to-use, nonsynchronous, step-down DC-DC converter with a wide input voltage range up to 40 V. The regulator is capable of delivering up to 3-A DC load current with excellent line and load regulation. These devices are available in fixed output voltages of 3.3-V, 5-V, 12-V and an adjustable output version.

The LM2596 series of regulators are monolithic integrated circuits that provide all the active functions for a step-down (buck) switching regulator, capable of driving a 3-A load with excellent line and load regulation. These devices are available in fixed output voltages of 3.3 V, 5 V, 12 V, and an adjustable output version. Requiring a minimum number of external components, these regulators are simple to use and include internal frequency compensation, and a fixed frequency oscillator. The LM2596 series operates at a switching frequency of 150 kHz. BLOCK DIAGRAM-



**REALISED MODEL**



**PROGRAM:**

**#include "HX711.h"**

**#include <SoftwareSerial.h>**

**#include <LiquidCrystal.h>**

**LiquidCrystal lcd(12,11,5,4,3,2);**

**HX711 cell(A1, A0);**

**long gaslevel=0;**

**float count=0;**

**float grms =0;**

**int lpgsensor=A2;**

**int relaypin1=37;**

**int relaypin2=52;**

**int ledred=50;**

**int ledwhite=48;**

**int gasalertvalue=0;**

**int smscount=0;**

**int smscount1=0;**

**void setup()**

**{**

**digitalWrite(relaypin1,HIGH);**

**digitalWrite(relaypin2,HIGH);**

**digitalWrite(ledred,LOW);**

**digitalWrite(ledwhite,HIGH);**

**pinMode(lpgsensor,INPUT);**

**pinMode(relaypin1,OUTPUT);**

**pinMode(relaypin2,OUTPUT);**

**pinMode(ledred,OUTPUT);**

**pinMode(ledwhite,OUTPUT);**

**Serial.begin(9600);**

**lcd.begin(16,2);**

**//Serial.println("Gasvalue:-");**

**//Serial.println(" \t \t \t WEIGHT");**

**}**

**void loop()**

**{**

**checkgas();**

**checkgaslevel();**

**}**

**void checkgas()**

**{**

**lcd.clear();**

**lcd.setCursor(0,0);**

**lcd.print("Gas Scan - ON");**

**delay(2000);**

**gasalertvalue=analogRead(lpgsensor);**

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

**//Serial.println(gasalertvalue);**

**if(gasalertvalue>120)**

**{**

**digitalWrite(relaypin1,LOW);**

**digitalWrite(relaypin2,LOW);**

**digitalWrite(ledred,HIGH);**

**digitalWrite(ledwhite,LOW);**

**lcd.clear();**

**lcd.setCursor(0,0);**

**lcd.print("Gas leakage!");**

**delay(4000); //callibrate the window opening time and set delay**

**digitalWrite(relaypin2,HIGH); //Check the relay pin to which motor is fed**

**setalert();**

**}**

**else**

**{**

**lcd.clear();**

**lcd.setCursor(0,0);**

**lcd.print("No gas leakage");**

**delay(2000);**

**digitalWrite(relaypin1,HIGH);**

**digitalWrite(relaypin2,HIGH);**

**digitalWrite(ledred,LOW);**

**digitalWrite(ledwhite,HIGH);**

**smscount=0;**

**}**

**}**

**void setalert()**

**{**

**while(smscount<1)**

**{**

**sendtextmessage();**

**}**

**}**

**void sendtextmessage()**

**{**

**lcd.clear();**

**lcd.setCursor(0,0);**

**lcd.print("Sending Message.");**

**delay(2000);**

**Serial.println("AT+CMGF=1");**

**Serial.println("AT+CMGS=\"+919439096990\"\r");**

**Serial.println("Gas Leaking!");**

**Serial.println((char)26);**

**Serial.println("AT+CMGF=1");**

**Serial.println("AT+CMGS=\"+919439667395\"\r");**

**Serial.println("Gas Leaking!");**

**Serial.println((char)26);**

**smscount++;**

**lcd.clear();**

**lcd.setCursor(0,0);**

**lcd.print("Message Sent...");**

**delay(1000);**

**}**

**void checkgaslevel()**

**{**

**count = count+1;**

**gaslevel =cell.read();**

**grms =((gaslevel-36314)/54);**

**//Serial.print("\t \t \t ");**

**//Serial.print(grms);**

**if(grms<=100)**

**{**

**lcd.clear();**

**lcd.setCursor(0,0);**

**lcd.print("Gas level 10%");**

**delay(2000);**

**setalert1();**

**}**

**else**

**{**

**lcd.clear();**

**lcd.setCursor(0,0);**

**lcd.print("Gas level ok");**

**delay(2000);**

**}}**

**void setalert1()**

**{**

**while(smscount1<1)**

**{**

**sendtextmessage1();**

**}**

**}**

**void sendtextmessage1()**

**{**

**lcd.clear();**

**lcd.setCursor(0,0);**

**lcd.print("Sending Message.");**

**delay(2000);**

**Serial.println("AT+CMGF=1");**

**Serial.println("AT+CMGS=\"+919439096990\"\r");**

**Serial.println("Gas Level 10%!");**

**Serial.println((char)26);**

**Serial.println("AT+CMGF=1");**

**Serial.println("AT+CMGS=\"+919777818303\"\r");**

**Serial.println("Deliver gas!");**

**Serial.println((char)26);**

**smscount1++;**

**lcd.clear();**

**lcd.setCursor(0,0);**

**lcd.print("Message Sent...");**

**delay(1000);**

**}**