**Padmabhushan Vasantdada Patil Pratishthan’s College of Engineering**



A MINI-PROJECT REPORT

ON

**“ELEVATION COMPUTATION”**

Submitted by:-

**KARTHIK PILLAI VU4F1718038**

**SATYAM YADAV VU4F1718040**

**MOHIT SAINI VU4F1718045**

**Department of Information Technology**

**Padmabhushan Vasantdada Patil Pratishthan’s College of Engineering**

Sion-Chunabhatti Eastern Express Highway Mumbai University, Mumbai 400022

2018-2019

**Padmabhushan Vasantdada Patil Pratishthan’s College of Engineering**

Sion-Chunabhatti Eastern Express Highway

Mumbai 400022

**Department of Information Technology**



CERTIFICATE

Certified that the mini-project work entitled **“ELEVATION COMPUTATION”** is a bonafide workcarried out by:

**KARTHIK PILLAI VU4F1718038**

**SATYAM YADAV VU4F1718040**

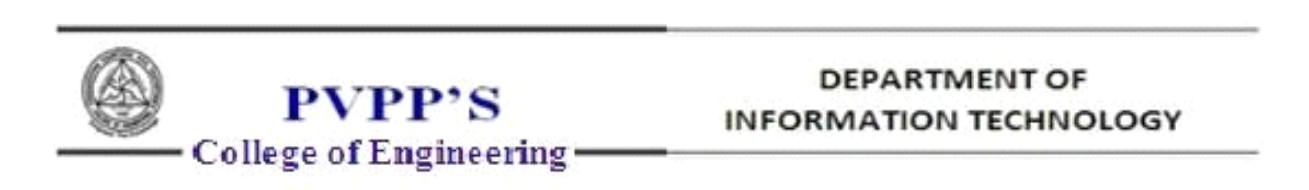
**MOHIT SAINI VU4F1718045**

The report has been approved as it satisfies the academic requirements in respect of mini-project work prescribed for the course.

…………….. ............................ ………………

**Prof. Pravin Patil Dr. Seema Ladhe Dr. Alam Shaikh**

Mini-Project Guide Head of Department Principal



**Institute Vision**

 To provide an environment to explore, encourage and educate students by facilitating innovative research, entrepreneurship, opportunities and employability to achieve professional goals. The following strategic characteristics and aspirations enable the college to realize its vision.

**Institute Mission**

* To provide facilities in the area of research and development.
* To initiate the collaboration with industries and academic institutions in terms of project and internship.
* To build up appropriate moral and ethical skills, to promote holistic development of students through various academic, social and cultural activities.
* To develop leadership and to sharpen the students skill by providing them opportunities for working in an innovative and interactive environment.
* To kindle the zeal among the students and promote their quest for academic excellence.
* To strengthen industry academic interaction to bridge the gap between theory and practice.
* To recruit, retain and enable a diverse community of exceptional faculty and students.
* To mould the students into competent professionals to foster economic development to meet the societal needs globally.

**Department Vision**

 To be a centre of innovation by adopting changes in the field of Information Technology to meet the challenging needs of the society and industry.

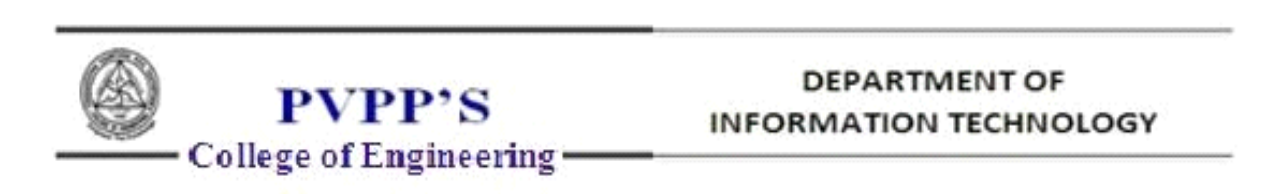
**Department Mission**

 To provide an academic environment to students by including Problem solving, Teamwork and Leadership Skills to achieve their goals in the field of Information Technology.

**Program Educational Objectives (PEOs)**

The IT engineering program has established the following program educational objectives:

* **Excellence:** To motivate students to become successful in the field of Information Technology around the globe.
* **Core skills Set:**  To provide students with a sound foundation in mathematics and engineering fundamentals to solve engineering problems.
* **Innovation:** To train students to identify and formulate the real life problems and obtain solutions using modern tools and technologies.
* **Learning Environment:** To encourage students for lifelong learning to improve their knowledge in depth and enhance career opportunities.
* **Professionalism:** To inculcate ethical values & nurture professional attitude which helps students to become a responsible citizen.



**Program Specific Outcomes (PSOs)**

The graduates of this program will be able to:

* Create model and develop complex information technology solutions by using advanced tools in open source technologies.
* Achieve technical and management skills to meet the current Information Technology demands of the industry and society.

**Program Outcomes (POs)**

* **Basic Engineering Knowledge:** Students should be able to apply the fundamental knowledge in mathematics, science and engineering for solving problems in Information Technology.
* **Problem Analysis:** Students should be able to identify, analyze and interpret data to formulate, design, develop or research various solutions for complex engineering problems in the field of Information Technology.
* **Design / Development of Solution:** Students should be able to design solutions for the benefit of the society.
* **Conduct Investigation of Complex problems:** The students should be able to provide solutions to complex problems that may have multiple solutions, may not be having specific constraints / requirements defined.
* **Modern Tool Usage:** Create, select, and apply appropriate techniques, and modern engineering and IT tools, including prediction and modeling to complex engineering activities, with an understating of the limitations.
* **The Engineer and Society:**  Student should be able to apply the knowledge to assess social issues and responsibilities relevant to engineering practices.
* **Environment and sustainability:**  Students should be able to understand the impact of engineering solutions in social and environmental context and demonstrate the need of sustainable development.



* **Ethics:**  Students should apply ethical principles while handling engineering practice.
* **Individual and Team Work:** Students should be able to function effectively as an individual or as a part of a team in multidisciplinary settings.
* **Communication :** Students should be able to communicate, comprehend and document effectively
* **Project Management and Finance:** Students should be able to demonstrate knowledge and understanding of the engineering and management principles and apply them effectively to manage projects.
* **Lifelong Learning:**  The students should be able to select, design, integrate and administer IT based solutions into the organizational environment by enhancing their knowledge & skill sets.

**Acknowledgement**

We owe our gratitude to the many people who have supported us throughout this journey. We would firstly like to express our heartfelt gratitude towards our respected Principal Dr. Alam Shaikh and our Head of Department Dr. Seema Ladhe for providing us immense facilities, guidance and never ending support.

The completion of any inter-disciplinary project depends upon cooperation and combined efforts of several sources of knowledge. We take this opportunity to express our profound gratitude and deep regards to our guide **Mr. Pravin Patil** for him exemplary guidance, monitoring and constant encouragement throughout the course of this project. The blessing, help and guidance given by him, from time to time shall carry us a long way in the journey of life on which we are about to embark.

Lastly, we thank almighty, our parents, family, friends and well wishers who always looked for the chance to help us in whatever means came forth and for their constant encouragement without which the project would not be a distant reality.

**Abstract**

Now a days in an age where tasks and systems are fusing together with the power of IOT to have a more efficient system of working and to execute jobs quickly! With all the power at our fingertips this is what we have come up with.

The Internet of Things (IoT) shall be able to incorporate transparently and seamlessly a large number of different systems, while providing data for millions of people to use and capitalize. Building a general architecture for the IoT is hence a very complex task, mainly because of the extremely large variety of devices, link layer technologies, and services that may be involved in such a system.

**List of Symbols and Abbreviations:**

* VCC:- Voltage common collector

* GND:- ground

* TXD :- Transmitted
* RXD:- receiver

* L :- Length

* W :- :- Width

* H :- Height

* I/O :- input output

* SRAM:- Static Random Access memory
* EEPROM :- Electrically Erasable Programmable Read-Only Memory

* DC:- Direct Current

* mA:- milliAmpere

* WPA2:- Wi-Fi Protected Access 2

* IPv4:- Internet Protocol HyperText Transfer Protocol

* TCP :- Transmission Control Protocol

* HTTP :- HyperText Transfer Protocol

* FTP :- File Transfer Protocol
* UDP :- User Datagram Protocol

**Chapter 1**

**INTRODUCTION**

**1.1 OVERVIEW**

Traditional methods of measurement involves measuring tapes,height charts and optical devices.Because of the usage of sensors we do not need to use traditional methods like measuring tapes.In order to have highly accurate height measurements in any scientific experiments, research or projects,physical contact should be minimized. Automatic height detection reduces this very possibility and increases efficiency and accuracy.To estimate the object dimensions with precision, measurement of devices and their dimensions is of utmost importance in industry.

The dimensions of nano- particles, microscopic beings, as also the height and dimensions of macroscopic objects, all find applications in day to day life.Accurate dimensions are hence needed in these cases.Dimensions of metal blocks, or bricks in a furnace can be measured also.To make the task of height measurement less tedious: Standing in-front of a measuring scale, manually calculating height is a highly inefficient procedure.Height measuring device removes these inaccuracies introduced due to human intervention and improves the efficiency of systems.

The Ultrasonic sensor calculates the obstacle distance and gyro sensor calculates elevation angle with respect to the ground and then provides an analog output further forwarded to the arduino board.The LCD displays the height that is fed to it and provides the final output.

**1.2 NEED FOR IMPROVEMENT IN ELEVATION COMPUTATION**

The various manual methods have been used till date to measure height.The typical method is to stand in front of a height measuring scale and manually checking the height.However, a number of errors may be introduced due to human errors.The human eye cannot accurately calculate and hence is not a very efficient method.In terms of future scope, the aging of trees cannot be accurately done by this method.Hence the need for Elevation Computation.

Future applications include industries where the level of liquid in a container needs to be measured,in factories where the height of (say) wooden blocks needs to be known in order to cut them precisely,in medical clinics where the height of human beings has to be recorded, in tunnels for proper height detection of huge vehicles, in developing contour maps where height of buildings, mountains etc needs to be known.

**1.3 FEATURES OF ELEVATION COMPUTATION**

* Quicker method to measure elevation of an object from ground level.
* Easily portable device.
* Accurate measurements.
* Easy to use.
* Convenient method as compared to traditional measuring devices.
* Horizontal distance of an object can also be measured accurately.

**1.4 METHODOLOGY OF ELEVATION COMPUTATION**

* We know that a triangle has 3 sides and 3 internal angles.
* Even if we have any 2 of them we can find the other sides and angles.
* Using this method we are creating a device to measure the heights.
* We will use an ultrasonic sensor (HC-SR04) to measure the length of the adjacent side.
* A Gyro sensor (MPU6050) to measure the angle between Adjacent and Hypotenuse.
* Using these measures we can calculate the height of the object.

**1.5 ADVANTAGES OF ELEVATION COMPUTATION**

* It is less time consuming while using the device.
* Small and light packaging.
* This improves the versatility of the device.
* Instant display without added delay further adds to its benefits.
* The automatic measurement removes human intervention altogether.
* Height Measurement device improves efficiency.
* Height of an object can be measured in less than a minute.

**1.6 APPLICATION OF ELEVATION COMPUTATION**

* Height measurement device finds its applications in military services, hospitals, construction sites etc.
* Future applications include industries where the level of liquid in a container needs to be measured,in factories where the height of (say) wooden blocks needs to be known in order to cut them precisely,in medical clinics where the height of human beings has to be recorded, in tunnels for proper height detection of huge vehicles, in developing contour maps where height of buildings, mountains etc needs to be known.

**Chapter 2**

**LITERATURE SURVEY**

Before we started with the design and implementation of the system, a detailed survey on method of height measurement,its existing systems in market has been carried out.

**Height Measurement:**

Height is the measurement of vertical distance but has two meanings in common use. It can either indicate how tall something is or how high up it is. For example the height of the building is 50 meters or the height of the airplane is 10000m. Height measurement device is a device used to accurately measure the distance of objects. Height measurement device finds its applications in military services, hospitals, security systems etc.

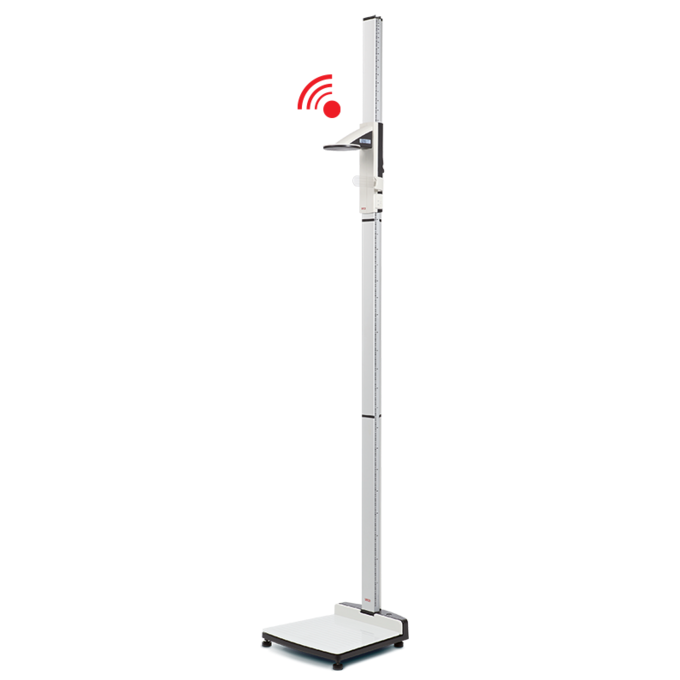
<https://en.wikipedia.org/wiki/Stadiometer>

**Chapter 3**

**SYSTEM STUDY**

3.1 **Existing System:**

Existing systems include: mobile stadiometer as shown in figure below. The measurement range of the said device is from 20 cm to 205 cm. The price is steep which makes it an unfavourable instrument for widespread usage.



Another device is the wall mounted height device as shown in the below figure. Range of this device is 14cm to 200cm. It is available at a price ranging upto 90 dollars which makes it impossible for mass usage.



The accu-hitemeasuring device is the third of the researched pre-existing systems, with a measuring range of 36.2cm to 90cm, and price ranging upwards of 80 dollars.



**3.1 PROPOSED SYSTEM:**

The proposed system ‘IoT based Height Measurement Device (monitor and acknowledgement)’ shall provide the smart way of measuring the height. This device shall be beneficial in measuring heights without doing it physically.

**Features :**

* Quicker method to measure elevation of an object from ground level.
* Easily portable device.
* Accurate measurements.
* Easy to use.
* Convenient method as compared to traditional measuring devices.
* Horizontal distance of an object can also be measured accurately.

**3.2 HARDWARE SPECIFICATION**

**3.2.1 ARDUINO BOARD**

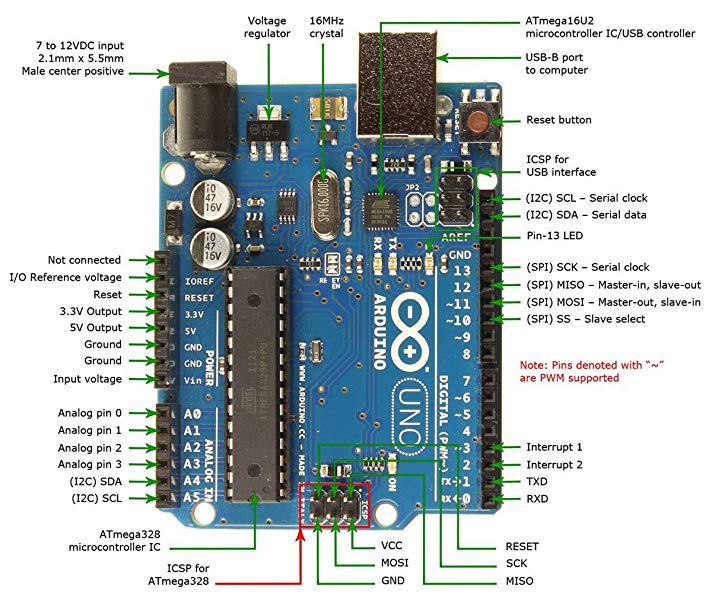
The **Arduino Uno** is an [open-source](https://en.wikipedia.org/wiki/Open-source) [microcontroller board](https://en.wikipedia.org/wiki/Microcontroller_board) based on the [Microchip](https://en.wikipedia.org/wiki/Microchip_Technology) [ATmega328P](https://en.wikipedia.org/wiki/ATmega328P) microcontroller and developed by [Arduino.cc](https://en.wikipedia.org/wiki/Arduino). The board is equipped with sets of digital and analog [input/output](https://en.wikipedia.org/wiki/Input/output) (I/O) pins that may be interfaced to various [expansion boards](https://en.wikipedia.org/wiki/Expansion_board) (shields) and other circuits.The board has 14 Digital pins, 6 Analog pins, and programmable with the [Arduino IDE](https://en.wikipedia.org/wiki/Arduino#Software) (Integrated Development Environment) via a type B [USB cable](https://en.wikipedia.org/wiki/USB_cable). It can be powered by the USB cable or by an external [9-volt battery](https://en.wikipedia.org/wiki/9-volt_battery), though it accepts voltages between 7 and 20 volts. It is also similar to the [Arduino Nano](https://en.wikipedia.org/wiki/Arduino_Nano) and Leonardo. The hardware reference design is distributed under a [Creative Commons](https://en.wikipedia.org/wiki/Creative_Commons) Attribution Share-Alike 2.5 license and is available on the Arduino website. Layout and production files for some versions of the hardware are also available.

The word "[uno](https://en.wiktionary.org/wiki/uno)" means "one" in [Italian](https://en.wikipedia.org/wiki/Italian_language) and was chosen to mark the initial release of the [Arduino Software](https://en.wikipedia.org/wiki/Arduino_Software). The Uno board is the first in a series of USB-based Arduino boards,and it and version 1.0 of the Arduino [IDE](https://en.wikipedia.org/wiki/Integrated_development_environment) were the reference versions of Arduino, now evolved to newer releases. The ATmega328 on the board comes pre programmed with a [bootloader](https://en.wikipedia.org/wiki/Bootloader) that allows uploading new code to it without the use of an external hardware programmer.

While the Uno communicates using the original STK500 protocol, it differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it uses the Atmega16U2 (Atmega8U2 up to version R2) programmed as a [USB-to-serial converter](https://en.wikipedia.org/wiki/Usb_to_serial_adapter).

**Technical specifications**

* [Microcontroller](https://en.wikipedia.org/wiki/Microcontroller): [Microchip](https://en.wikipedia.org/wiki/Microchip_Technology) [ATmega328P](https://en.wikipedia.org/wiki/ATmega328P)
* Operating Voltage: 5 Volts
* Input Voltage: 7 to 20 Volts
* Digital I/O Pins: 14 (of which 6 provide PWM output)
* Analog Input Pins: 6
* DC Current per I/O Pin: 20 mA
* DC Current for 3.3V Pin: 50 mA
* [Flash Memory](https://en.wikipedia.org/wiki/Flash_Memory): 32 KB of which 0.5 KB used by [bootloader](https://en.wikipedia.org/wiki/Booting#BOOT-LOADER)
* [SRAM](https://en.wikipedia.org/wiki/Static_random-access_memory): 2 KB
* [EEPROM](https://en.wikipedia.org/wiki/EEPROM): 1 KB
* Clock Speed: 16 MHz
* Length: 68.6 mm
* Width: 53.4 mm
* Weight: 25 g



**FIG 3.2.4 : ARDUINO BOARD**

**Pins**

**General pin functions**

* **LED**: There is a built-in LED driven by digital pin 13. When the pin is high value, the LED is on, when the pin is low, it's off.
* **VIN**: The input voltage to the Arduino/Genuino 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**: This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 20V), the USB connector (5V), or the VIN pin of the board (7-20V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage the board.
* **3V3**: A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
* **GND**: Ground pins.
* **IOREF**: This pin on the Arduino/Genuino board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs to work with the 5V or 3.3V.
* **Reset**: Typically used to add a reset button to shields which block the one on the board.

**Special pin functions**

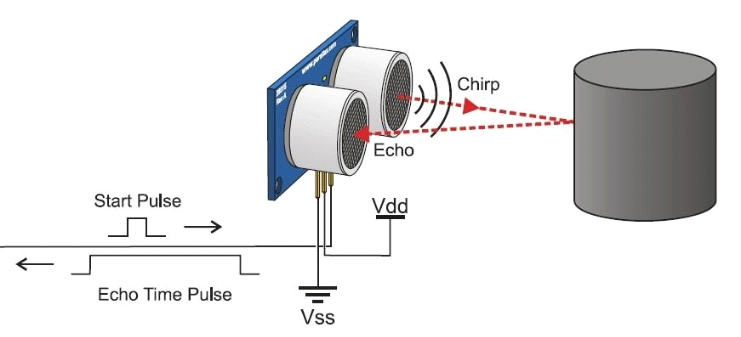
Each of the 14 digital pins and 6 analog pins on the Uno can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive 20 mA as recommended operating condition and has an internal pull-up resistor (disconnected by default) of 20-50k ohm. A maximum of 40mA is the value that must not be exceeded on any I/O pin to avoid permanent damage to the microcontroller. The Uno has 6 analog inputs, labeled A0 through A5, 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 the analogReference() function.

In addition, some pins have specialized functions:

* **Serial** / [UART](https://en.wikipedia.org/wiki/UART): pins 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL serial chip.
* **External interrupts**: pins 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.
* [**PWM**](https://en.wikipedia.org/wiki/Pulse-width_modulation) (pulse-width modulation): 3, 5, 6, 9, 10, and 11. Can provide 8-bit PWM output with the analogWrite() function.
* [**SPI**](https://en.wikipedia.org/wiki/Serial_Peripheral_Interface) (Serial Peripheral Interface): 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication using the SPI library.
* **TWI** (two-wire interface) / [I²C](https://en.wikipedia.org/wiki/I%C2%B2C): A4 or SDA pin and A5 or SCL pin. Support TWI communication using the Wire library.
* **AREF** (analog reference): Reference voltage for the analog inputs.

**3.2.2 ULTRASONIC SENSOR**

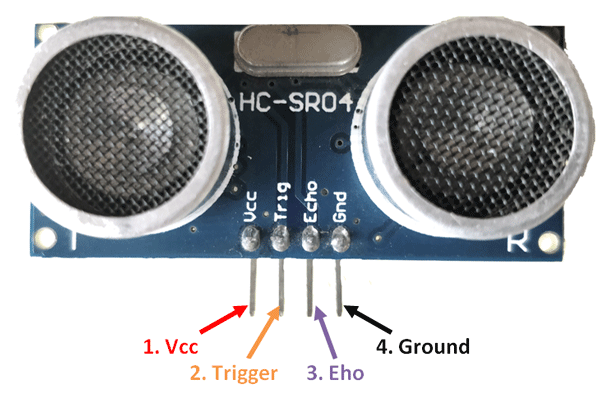
A special sonic transducer is used for ultrasonic proximity sensors, which allows for alternate transmission and reception of sound waves. The sonic waves emitted by the transducer are reflected by an object and received back in the transducer. After having emitted the sound waves, the ultrasonic sensor will switch to receive mode. The time elapsed between emitting and receiving is proportional to the distance of the object from the sensor.



Ultrasonic sensors generate high-frequency sound waves and evaluate the echo which is received back by the sensor, measuring the time interval between sending the signal and receiving the echo to determine the distance to an object.

### **Ultrasonic Sensor Pin Configuration**

|  |  |  |
| --- | --- | --- |
| **Pin Number** | **Pin Name** | **Description** |
| 1 | Vcc | The Vcc pin powers the sensor, typically with +5V |
| 2 | Trigger | Trigger pin is an Input pin. This pin has to be kept high for 10us to initialize measurement by sending US wave. |
| 3 | Echo | Echo pin is an Output pin. This pin goes high for a period of time which will be equal to the time taken for the US wave to return back to the sensor. |
| 4 | Ground | This pin is connected to the Ground of the system. |



### **HC-SR04 Sensor Features**

* Operating voltage: +5V
* Theoretical  Measuring Distance: 2cm to 450cm
* Practical Measuring Distance: 2cm to 80cm
* Accuracy: 3mm
* Measuring angle covered: <15°
* Operating Current: <15mA
* Operating Frequency: 40Hz

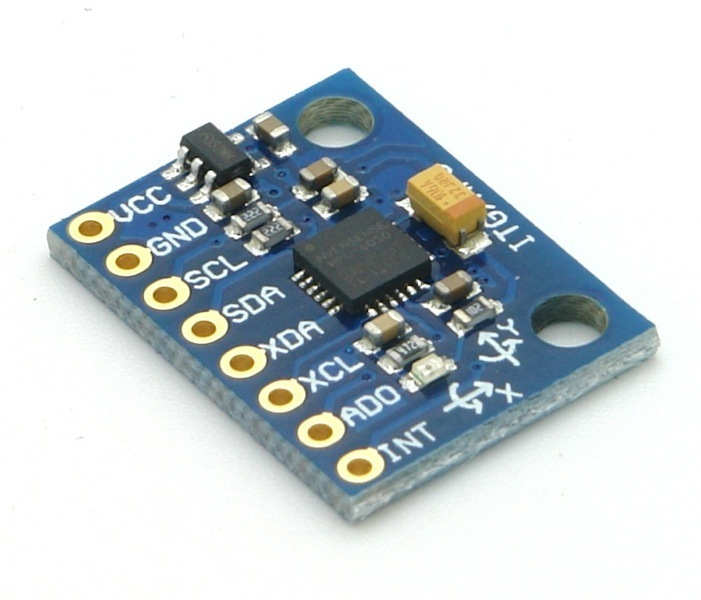
### **Working of HC-SR04 Ultrasonic Sensor**

Power the Sensor using a regulated +5V through the Vcc and Ground pins of the sensor. The current consumed by the sensor is less than 15mA and hence can be directly powered by the on board 5V pins (If available). The Trigger and the Echo pins are both I/O pins and hence they can be connected to I/O pins of the microcontroller. To start the measurement, the trigger pin has to be made high for 10uS and then turned off. This action will trigger an ultrasonic wave at frequency of 40Hz from the transmitter and the receiver will wait for the wave to return. Once the wave is returned after it getting reflected by any object the Echo pin goes high for a particular amount of time which will be equal to the time taken for the wave to return back to the sensor.

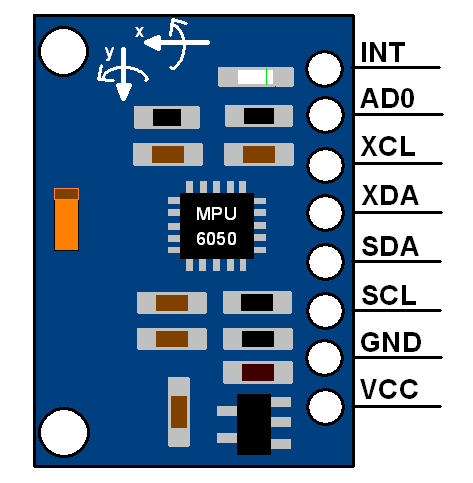
The amount of time during which the Echo pin stays high is measured by the MCU/MPU as it gives the information about the time taken for the wave to return back to the Sensor. Using this information the distance is measured

**3.2.3 GYROSCOPE SENSOR**

The MPU-6050 devices combine a 3-axis gyroscope and a 3-axis accelerometer on the same silicon die together with an on board Digital Motion Processor (DMP), which processes complex 6-axis Motion Fusion algorithms. The device can access external magnetometers or other sensors through an auxiliary master I²C bus, allowing the devices to gather a full set of sensor data without intervention from the system processor. The devices are offered in a 4 mm x 4 mm x 0.9 mm QFN package.



# **MPU-6050 Module**



The MPU-6050 module has 8 pins,

**INT:** Interrupt digital output pin.

**AD0:** I2C Slave Address LSB pin. This is 0th bit in 7-bit slave address of device. If connected to VCC then it is read as logic one and slave address changes.

**XCL:** Auxiliary Serial Clock pin. This pin is used to connect other I2C interface enabled sensors SCL pin to MPU-6050.

**XDA:** Auxiliary Serial Data pin. This pin is used to connect other I2C interface enabled sensors SDA pin to MPU-6050.

**SCL:** Serial Clock pin. Connect this pin to microcontrollers SCL pin.

**SDA:** Serial Data pin. Connect this pin to microcontrollers SDA pin.

**GND:** Ground pin. Connect this pin to ground connection.

**VCC:** Power supply pin. Connect this pin to +5V DC supply.

MPU-6050 module has Slave address (When AD0 = 0, i.e. it is not connected to Vcc) as,

**Slave Write address(SLA+W)**: 0xD0

**Slave Read address(SLA+R)**: 0xD1

**Features**

* I2C Digital-output of 6 or 9-axis MotionFusion data in rotation matrix, quaternion, Euler Angle, or raw data format
* Input Voltage: 2.3 - 3.4V
* Selectable Solder Jumpers on CLK, FSYNC and AD0
* Tri-Axis angular rate sensor (gyro) with a sensitivity up to 131 LSBs/dps and a full-scale range of ±250, ±500, ±1000, and ±2000dps
* Tri-Axis accelerometer with a programmable full scale range of ±2g, ±4g, ±8g and ±16g
* Digital Motion Processing™ (DMP™) engine offloads complex MotionFusion, sensor timing synchronization and gesture detection
* Embedded algorithms for run-time bias and compass calibration. No user intervention required
* Digital-output temperature sensor

**Specifications**

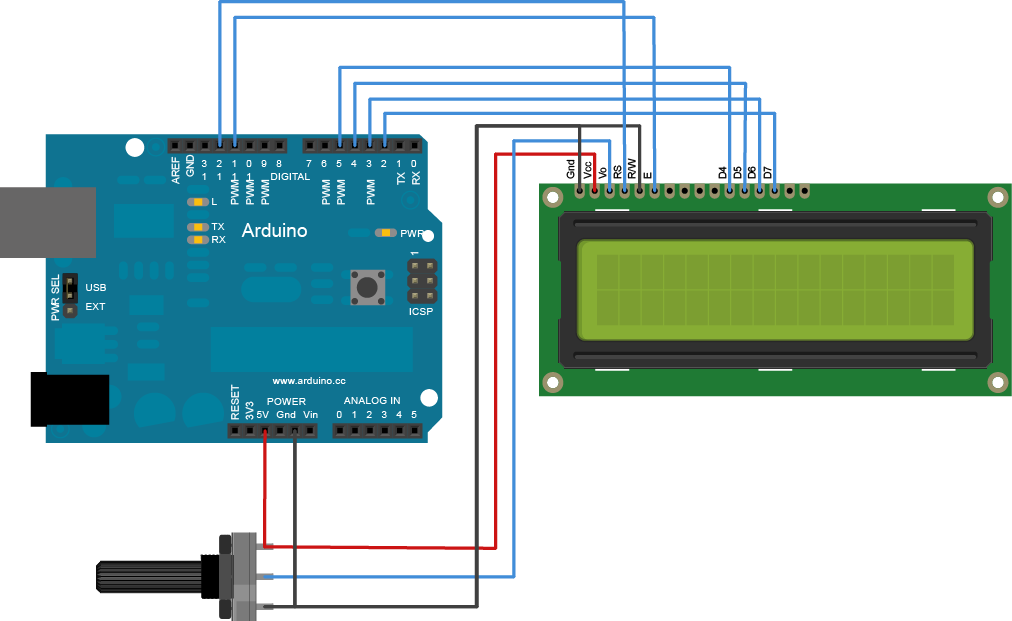
* Chip: MPU-6050
* Power supply: 3~5V Onboard regulator
* Communication mode: standard IIC communication protocol
* Chip built-in 16bit AD converter, 16bit data output
* Gyroscopes range: +/- 250 500 1000 2000 degree/sec
* Acceleration range: +/- 2g, +/- 4g, +/- 8g, +/- 16g
* Pin pitch: 2.54mm
* Great for DIY projects

**3.2.4 LCD** **and POTENTIOMETER**

A liquid-crystal display (LCD) is a [flat-panel display](https://en.wikipedia.org/wiki/Flat_panel_display) or other [electronically modulated optical device](https://en.wikipedia.org/wiki/Electro-optic_modulator) that uses the light-modulating properties of [liquid crystals](https://en.wikipedia.org/wiki/Liquid_crystal). Liquid crystals do not emit light directly, instead using a [backlight](https://en.wikipedia.org/wiki/Backlight) or [reflector](https://en.wikipedia.org/wiki/Reflector_(photography)) to produce images in color or [monochrome](https://en.wikipedia.org/wiki/Monochrome). LCDs are available to display arbitrary images (as in a general-purpose computer display) or fixed images with low information content, which can be displayed or hidden, such as preset words, digits, and [seven-segment displays](https://en.wikipedia.org/wiki/Seven-segment_display), as in a [digital clock](https://en.wikipedia.org/wiki/Digital_clock). They use the same basic technology, except that arbitrary images are made up of many small [pixels](https://en.wikipedia.org/wiki/Pixel), while other displays have larger elements. LCDs can either be normally on (positive) or off (negative), depending on the polarizer arrangement. For example, a character positive LCD with a backlight will have black lettering on a background that is the color of the backlight, and a character negative LCD will have a black background with the letters being of the same color as the backlight. Optical filters are added to white on blue LCDs to give them their characteristic appearance.

A potentiometer is a three-[terminal](https://en.wikipedia.org/wiki/Terminal_(electronics)) [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). A potentiometer is a simple knob that provides a variable resistance, which we can read into the Arduino board as an analog value. In this example, that value controls the rate at which an LED blinks. We connect three wires to the Arduino board. The first goes to ground from one of the outer pins of the potentiometer. The second goes from 5 volts to the other outer pin of the potentiometer. The third goes from analog input 2 to the middle pin of the potentiometer. By turning the shaft of the potentiometer, we change the amount of resistance on either side of the wiper which is connected to the center pin of the potentiometer. This changes the relative "closeness" of that pin to 5 volts and ground, giving us a different analog input.

|  |  |
| --- | --- |
| Terminal 1 | GND |
| Terminal 2 | +5V |
| Terminal 3 | Mid terminal of potentiometer (for brightness control) |
| Terminal 4 | Register Select (RS) |
| Terminal 5 | Read/Write (RW) |
| Terminal 6 | Enable (EN) |
| Terminal 7 | DB0 |
| Terminal 8 | DB1 |
| Terminal 9 | DB2 |
| Terminal 10 | DB3 |
| Terminal 11 | DB4 |
| Terminal 12 | DB5 |
| Terminal 13 | DB6 |
| Terminal 14 | DB7 |
| Terminal 15 | +4.2-5V |
| Terminal 16 | GND |



**3.2.5 BATTERY**



The **nine-volt battery**, or **9-volt battery**, is a common size of battery that was introduced for the early [transistor radios](https://en.wikipedia.org/wiki/Transistor_radio). It has a rectangular prism shape with rounded edges and a polarized snap connector at the top. This type is commonly used in [walkie-talkies](https://en.wikipedia.org/wiki/Walkie-talkie), [clocks](https://en.wikipedia.org/wiki/Clock) and [smoke detectors](https://en.wikipedia.org/wiki/Smoke_detector).

The nine-volt battery format is commonly available in primary carbon-zinc and alkaline chemistry, in primary lithium iron disulfide, and in rechargeable form in nickel-cadmium, nickel-metal hydride and lithium-ion. Mercury-oxide batteries of this format, once common, have not been manufactured in many years due to their mercury content. [Designations](https://en.wikipedia.org/wiki/Battery_nomenclature) for this format include NEDA 1604, IEC 6F22 (for zinc-carbon) or MN1604 6LR61 (for alkaline). The size, regardless of chemistry, is commonly designated **PP3**—a designation originally reserved solely for carbon-zinc, or in some countries, E or E-block.

Most nine-volt alkaline batteries are constructed of six individual 1.5 V LR61 cells enclosed in a wrapper. These cells are slightly smaller than LR8D425 [AAAA cells](https://en.wikipedia.org/wiki/AAAA_battery) and can be used in their place for some devices, even though they are 3.5 mm shorter. Carbon-zinc types are made with six flat cells in a stack, enclosed in a moisture-resistant wrapper to prevent drying. Primary lithium types are made with three cells in series.

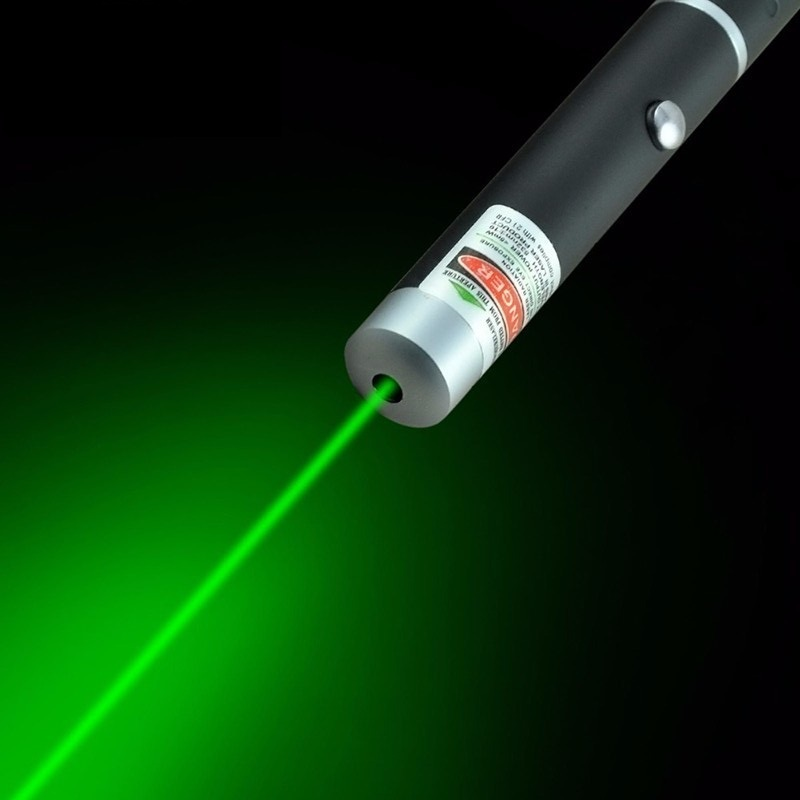
**Connectors**



The battery has both [terminals](https://en.wikipedia.org/wiki/Battery_terminal) in a snap connector on one end. The smaller circular (male) terminal is positive, and the larger hexagonal or octagonal (female) terminal is the negative contact. The connectors on the battery are the same as on the load device; the smaller one connects to the larger one and vice versa. The same snap-style connector is used on other battery types in the [Power Pack](https://en.wikipedia.org/wiki/List_of_battery_sizes#PP_series) (PP) series. Battery polarization is normally obvious, since mechanical connection is usually only possible in one configuration.

A problem with this style of connector is that it is very easy to connect two batteries together in a [short circuit](https://en.wikipedia.org/wiki/Short_circuit), which quickly discharges both batteries, generating heat and possibly a fire. Because of this hazard, nine-volt batteries should be kept in the original packaging until they are going to be used.

**3.2.6 LASER**

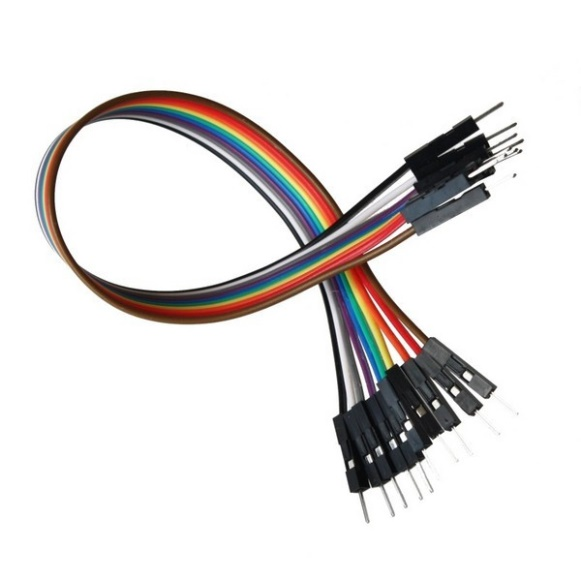


A **laser** is a device that emits [light](https://en.wikipedia.org/wiki/Light) through a process of [optical amplification](https://en.wikipedia.org/wiki/Optical_amplification) based on the [stimulated emission](https://en.wikipedia.org/wiki/Stimulated_emission) of [electromagnetic radiation](https://en.wikipedia.org/wiki/Electromagnetic_radiation). The term "laser" originated as an [acronym](https://en.wikipedia.org/wiki/Acronym) for "**light amplification by stimulated emission of radiation**".

A laser differs from other sources of light in that it emits light [*coherently*](https://en.wikipedia.org/wiki/Coherence_(physics)). [Spatial coherence](https://en.wikipedia.org/wiki/Spatial_coherence) allows a laser to be focused to a tight spot, enabling applications such as [laser cutting](https://en.wikipedia.org/wiki/Laser_cutting) and [lithography](https://en.wikipedia.org/wiki/Photolithography#Light_sources). Spatial coherence also allows a laser beam to stay narrow over great distances ([collimation](https://en.wikipedia.org/wiki/Collimated_light)), enabling applications such as [laser pointers](https://en.wikipedia.org/wiki/Laser_pointer) and [lidar](https://en.wikipedia.org/wiki/Lidar). Lasers can also have high [temporal coherence](https://en.wikipedia.org/wiki/Temporal_coherence), which allows them to emit light with a very narrow [spectrum](https://en.wikipedia.org/wiki/Frequency_spectrum), i.e., they can emit a single color of light. Alternatively, temporal coherence can be used to produce pulses of light with a broad spectrum but durations as short as a [femtosecond](https://en.wikipedia.org/wiki/Femtosecond) ("[ultrashort pulses](https://en.wikipedia.org/wiki/Ultrashort_pulse)").

Lasers are used in [optical disk drives](https://en.wikipedia.org/wiki/Optical_disk_drive), [laser printers](https://en.wikipedia.org/wiki/Laser_printer), [barcode scanners](https://en.wikipedia.org/wiki/Barcode_scanner), [DNA sequencing instruments](https://en.wikipedia.org/wiki/DNA_sequencer), [fiber-optic](https://en.wikipedia.org/wiki/Fiber-optic_communication) and [free-space optical communication](https://en.wikipedia.org/wiki/Free-space_optical_communication), [laser surgery](https://en.wikipedia.org/wiki/Laser_surgery) and skin treatments, cutting and [welding](https://en.wikipedia.org/wiki/Laser_welding) materials, military and [law enforcement](https://en.wikipedia.org/wiki/Law_enforcement) devices for marking targets and [measuring range](https://en.wikipedia.org/wiki/Laser_rangefinder#Military) and speed, and in [laser lighting displays](https://en.wikipedia.org/wiki/Laser_lighting_display) for entertainment. They have been used for car [headlamps](https://en.wikipedia.org/wiki/Headlamp) on luxury cars, by using a blue laser and a phosphor to produce highly directional white light.

**3.2.7 JUMP WIRES**



A jump wire (also known as jumper wire, or jumper) is an [electrical wire](https://en.wikipedia.org/wiki/Electrical_wire), or group of them in a cable with a connector or pin at each end (or sometimes without them – simply "tinned"), which is normally used to interconnect the components of a [breadboard](https://en.wikipedia.org/wiki/Breadboard) or other prototype or test circuit, internally or with other equipment or components, without soldering. Individual jump wires are fitted by inserting their "end connectors" into the slots provided in a breadboard, the [header connector](https://en.wikipedia.org/wiki/Pin_header#Header_connector) of a circuit board, or a piece of test equipment. Though jumper wires come in a variety of colors, the colors don’t actually mean anything. This means that a red jumper wire is technically the same as a black one. But the colors can be used to your advantage in order to differentiate between types of connections, such as ground or power.

Jumper wires typically come in three versions: male-to-male, male-to-female and female-to-female. The difference between each is in the end point of the wire. Male ends have a pin protruding and can plug into things, while female ends do not and are used to plug things into. Male-to-male jumper wires are the most common and what you likely will use most often. When connecting two ports on a breadboard, a male-to-male wire is what you’ll need.

**3.3 SOFTWARE REQUIREMENTS:**

**Arduino software:**

Arduino is an open source programmable circuit board that can be integrated into a wide variety of makerspace projects both simple and complex.  This board contains a [microcontroller](https://en.wikipedia.org/wiki/Microcontroller) which is able to be programmed to sense and control objects in the physical world.   By responding to sensors and inputs, the Arduino is able to interact with a large array of outputs such as LEDs, motors and displays.  Because of its flexibility and low cost, [Arduino](https://www.arduino.cc/) has become a very popular choice for makers and makerspaces looking to create interactive hardware projects.The board is equipped with sets of digital and analog [input/output](https://en.wikipedia.org/wiki/Input/output) (I/O) pins that may be interfaced to various [expansion boards](https://en.wikipedia.org/wiki/Expansion_board) (shields) and other circuits.The board has 14 Digital pins, 6 Analog pins, and programmable with the [Arduino IDE](https://en.wikipedia.org/wiki/Arduino#Software) (Integrated Development Environment) via a type B [USB cable](https://en.wikipedia.org/wiki/USB_cable). It can be powered by the USB cable or by an external [9-volt battery](https://en.wikipedia.org/wiki/9-volt_battery), though it accepts voltages between 7 and 20 volts. It is also similar to the [Arduino Nano](https://en.wikipedia.org/wiki/Arduino_Nano) and Leonardo. The hardware reference design is distributed under a [Creative Commons](https://en.wikipedia.org/wiki/Creative_Commons) Attribution Share-Alike 2.5 license and is available on the Arduino website. Layout and production files for some versions of the hardware are also available.

The word "[uno](https://en.wiktionary.org/wiki/uno)" means "one" in [Italian](https://en.wikipedia.org/wiki/Italian_language) and was chosen to mark the initial release of the [Arduino Software](https://en.wikipedia.org/wiki/Arduino_Software). The Uno board is the first in a series of USB-based Arduino boards,and it and version 1.0 of the Arduino [IDE](https://en.wikipedia.org/wiki/Integrated_development_environment) were the reference versions of Arduino, now evolved to newer releases. The ATmega328 on the board comes pre programmed with a [bootloader](https://en.wikipedia.org/wiki/Bootloader) that allows uploading new code to it without the use of an external hardware programmer.

While the Uno communicates using the original STK500 protocol, it differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it uses the Atmega16U2 (Atmega8U2 up to version R2) programmed as a [USB-to-serial converter](https://en.wikipedia.org/wiki/Usb_to_serial_adapter).

Arduino was introduced back in 2005 in Italy by Massimo Banzi as a way for non-engineers to have access to a low cost, simple tool for creating hardware projects.  Since the board is [open-source](https://en.wikipedia.org/wiki/Open-source_hardware), it is released under a Creative Commons license which allows anyone to produce their own board.  If you search the web, you will find there are hundreds of Arduino compatible clones and variations available but the only official boards have Arduino in its name.

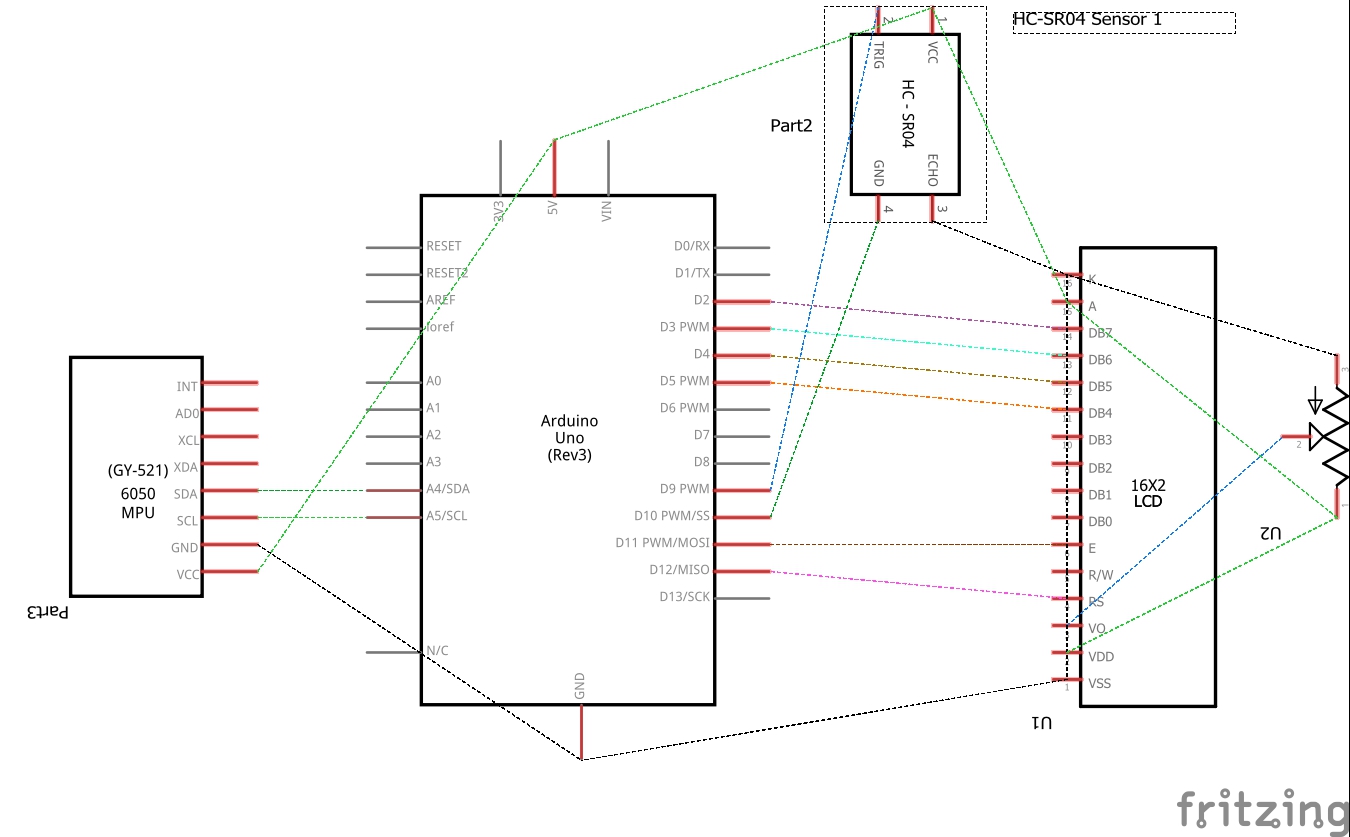
One of the most popular Arduino boards out there is the Arduino Uno.  While it was not actually the first board to be released, it remains to be the most actively used and most widely documented on the market.  Because of its extreme popularity, the Arduino Uno has a ton of project tutorials and forums around the web that can help you get started or out of a jam.  We’re big fans of the Uno because of it’s great features and ease of use.

The Arduino/Genuino Uno has a number of facilities for communicating with a computer, another Arduino/Genuino board, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The 16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, a .inf file is required. The Arduino Software (IDE) includes a serial monitor which allows simple textual data to be sent to and from the board.

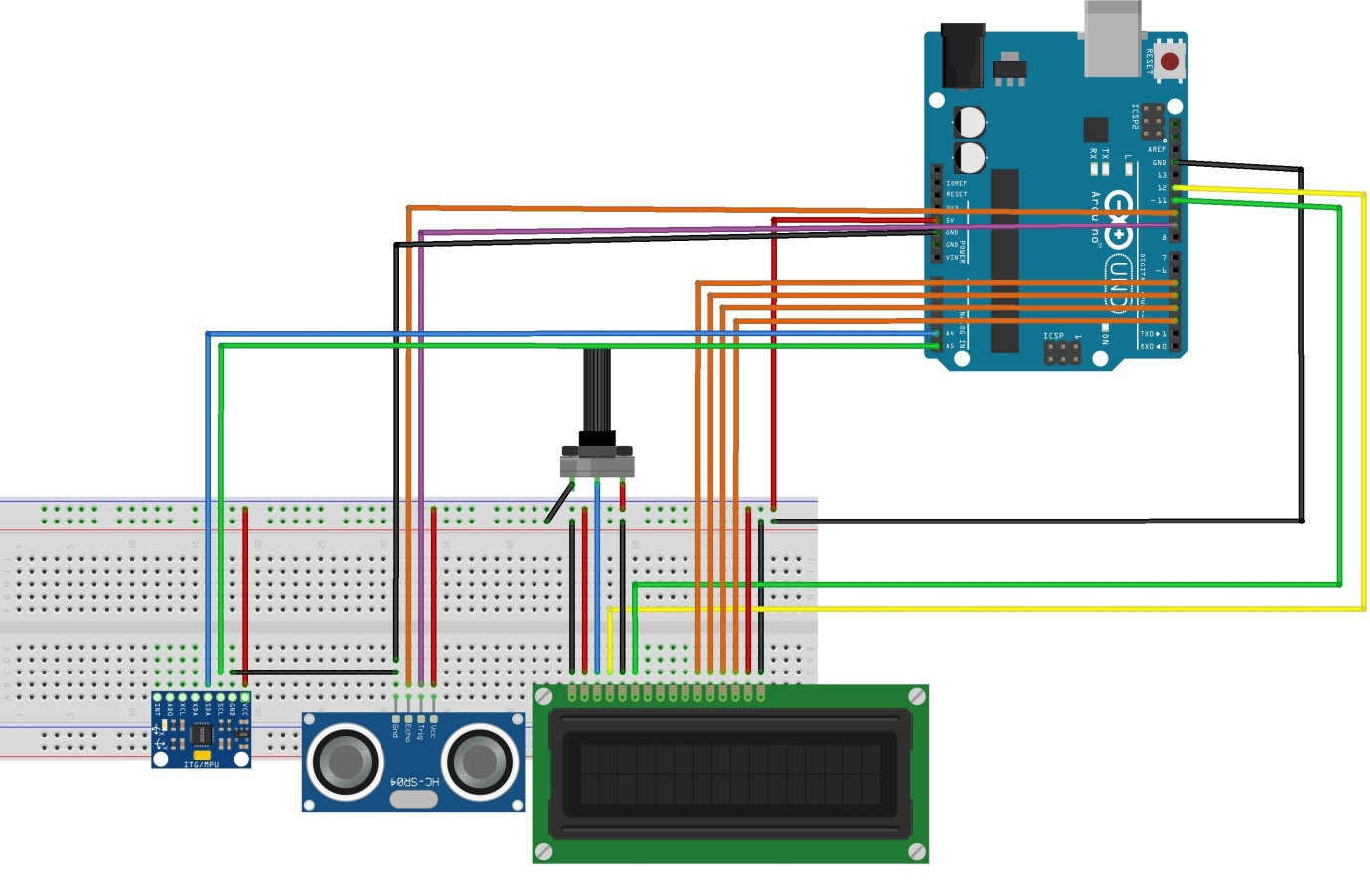
**Chapter 4**

**SYSTEM IMPLEMENTATION**

**4.1. BLOCK DIAGRAM**

****

**4.2 CIRCUIT DIAGRAM**

****

**4.3 CODE SNIPPETS**

**4.3.1 CODE FOR ULTRASONIC SENSOR**

int trigPin = 9;

int echoPin = 10;

float duration, distance;

void setup() {

pinMode(trigPin, OUTPUT);

pinMode(echoPin, INPUT);

Serial.begin(9600);

}

void loop() {

digitalWrite(trigPin, LOW);

delayMicroseconds(2);

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

duration = pulseIn(echoPin, HIGH);

distance = (duration\*0.0343)/2;

Serial.print("Distance: ");

Serial.println(distance);

delay(100);

}

**4.3.2 CODE FOR GYROSCOPIC SENSOR**

#include <MPU6050\_tockn.h>

#include <Wire.h>

MPU6050 mpu6050(Wire);

void setup() {

Serial.begin(9600);

Wire.begin();

mpu6050.begin();

mpu6050.calcGyroOffsets(true);

}

void loop() {

mpu6050.update();

lcd.print(mpu6050.getAngleX());

}

**4.3.3 CODE FOR LCD**

#include <LiquidCrystal.h>

// initialize the library with the numbers of the interface pins

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

void setup() {

// set up the LCD's number of columns and rows:

lcd.begin(16, 2);

// Print a message to the LCD.

lcd.print("HI MUMMY");

}

void loop() {

// set the cursor to column 0, line 1

// (note: line 1 is the second row, since counting begins with 0):

lcd.setCursor(0, 1);

// print the number of seconds since reset:

lcd.print(millis() / 1000);

}

**4.3.3 COMBINED CODE**

//GYRO\*

#include <MPU6050\_tockn.h>

#include <Wire.h>

MPU6050 mpu6050(Wire);

//GYRO

//LCD\*

#include <LiquidCrystal.h>

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

//LCD

//ULTRASONIC\*

int trigPin = 9;

int echoPin = 10;

float duration, distance;

//ULTRASONIC

float height=0;

void setup() {

//LCD\*

lcd.begin(16, 2);

//LCD

//ULTRASONIC\*

pinMode(trigPin, OUTPUT);

pinMode(echoPin, INPUT);

//ULTRASONIC

//GYRO\*

Serial.begin(9600);

Wire.begin();

mpu6050.begin();

mpu6050.calcGyroOffsets(true);

//GYRO

}

void loop() {

//ULTRASONIC\*

digitalWrite(trigPin, LOW);

delayMicroseconds(2);

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

duration = pulseIn(echoPin, HIGH);

distance = ((duration\*0.0343)/2)+30;

// Serial.print("Distance: ");

// Serial.println(distance);

//ULTRASONIC

lcd.setCursor(0,0);

lcd.print("D:");

lcd.setCursor(2,0);

lcd.print(distance);

mpu6050.update();

lcd.setCursor(9,0);

lcd.print("A:");

lcd.setCursor(11,0);

lcd.print(mpu6050.getAngleX());

lcd.setCursor(0,1);

lcd.print("H:");

lcd.setCursor(2,1);

height=(distance\*tan(mpu6050.getAngleX()\*0.01744));

lcd.print(height);

}

**CONCLUSION**

The various manual methods have been used till date to measure height , like manually checking the height. Errors may be introduced due to human eye errors.Hence measuring heights by electronic portable means is most efficient. Hence the need for automatic height measurement.