

**A MAJOR PROJECT ON**

**“IOT based Home Automation Systems”**

**PROJECT WORK SUBMITTED TO THE UTTARAKHAND BOARD OF TECHNICAL EDUCATION ROORKEE**

**SUBMITTED BY - PARTH SRIVASTAVA**

**ROLL NO – 16111050011**

**UNDER THE GUIDANCE OF**

**DR. RAJ KUMAR**

**(LECTURER OF BFIT)**

**DEHRADUN**

**DEPARTMENT OF COMPUTER SCIENCE ENGINEERING**

**BABA FARID INSTITUTION OF TECHNOLOGY DEHRADUN**

**CERTIFICATE**

This is to that the project report titled **“IOT based Home Automation Systems’’** submitted for the partial fulfilment of Diploma in Computer Science engineering of this collage under **‘’ UTTARAKHAND BOARD OF TECHNICAL EDUCATION ROORKEE ‘’** is a bonafied work carried out by the student under the general guidance and supervision of Dr. Raj Kumar during the academic session 2018-2019.

**Place: - Bfit Dehradun Dr. Raj Kumar**

**Date: - (LECTURER OF BFIT)**

**DEHRADUN**

**DECLARATION**

I am **PARTH SRIVASTAVA** here by declare that the project work entitled **“IOT based Home Automation Systems”** is carried out in the Baba Farid Group of Institute & Technology Dehradun, under the guidance and supervision of Dr. Raj Kumar during the academic session 2018-2019 (VI Sem May 2019) and has not been submitted before for the award of any diploma or degree of this or my other university.

**Place: - Dehradun**

**Date: -**

**PREFACE**

The project is essential feature pre-required for the award of diploma in mechanical engineering this year. We are asked to prepare a project on **IOT based Home Automation Systems** the project is given by:

***‘’UTTARAKHAND BOARD OF TECHNICAL EDUCATION’’***

This is brief project report wraps all the necessary information regarding the project detail specification detailed drawing of necessary part of various components.

I tried my best to make the project successful and practicable, but even there may be some mistake I beg my pardon for that.

I am heartily thankful to my respected teacher and my friend for their guidance and valuable help in completing this project.

**PARTH SRIVASTAVA**

**Final year**

**Computer Science engineering**

**Baba Farid group of institute & technology Dehradun**

**ACKNOWLEDGEMENT**

We are first of all thank to god who has instilled the determination and capability in our self to perform our tasks.

We wish to express our sincere regard and appreciation our guide Dr. Raj Kumar for his guidance, professional views and continuing interest on project.

We are also thankful to Miss. PRIYANKA RAWAT, head of Engineering Department.

Of course, completing our studies would not have been possible without any support of family, friend and other important day our life.

We most sincere to thank go to parents for their parental support.

**PARTH SRIVASTAVA**

**Final year**

**Computer Science Engineering**

**Baba Farid Group of Institute & Technology Dehradun**

**CERTIFICATE**

This is to that the project report titled “IOT based Home Automation Systems’’ submitted for the partial fulfilment of Diploma in Computer Science engineering is carried out in BABA FARID GROUP OF INSTITUTE & TECHNOLOGY, Dehradun during the academic session 2018-19 (VI Sem, May 2019).

**Place: Dehradun Miss. Priyanka Rawat**

**Date: Head of department**

**Contents**

|  |
| --- |
| **IOT** |
| * History of IOT |
| * Applications of IOT |
| * Enabling technologies for IoT |
| **Components Used** |
| * Arduino Uno |
| * + Technical Specifications |
| * + Pin Specification |
| * + Microchip ATmega328P |
| * + - Key parameters |
| * 2-Step 5v Relay |
| * + Features |
| * + Schematic |
| * + Principle |
| * Passive Infrared Sensor |
| * + Operation |
| * HC-05 Bluetooth Receiver |
| * 16x2 LCD Module |
| * Breadboard |
| **Configuration Diagram** |
| **The Code** |
| * Arduino IDE |
| **Project Summary** |

# **IOT**

The Internet of things (IoT) is the extension of Internet connectivity into physical devices and everyday objects. Embedded with electronics, Internet connectivity, and other forms of hardware (such as sensors), these devices can communicate and interact with others over the Internet, and they can be remotely monitored and controlled

The definition of the Internet of things has evolved due to convergence of multiple technologies, real-time analytics, machine learning, commodity sensors, and embedded systems. Traditional fields of embedded systems, wireless sensor networks, control systems, automation (including home and building automation), and others all contribute to enabling the Internet of things. In the consumer market, IoT technology is most synonymous with products pertaining to the concept of the "smart home", covering devices and appliances (such as lighting fixtures, thermostats, home security systems and cameras, and other home appliances) that support one or more common ecosystems, and can be controlled via devices associated with that ecosystem, such as smartphones and smart speakers.

The IoT concept has faced prominent criticism, especially in regards to privacy and security concerns related to these devices and their intention of pervasive presence.

# **History of IOT**

The concept of a network of smart devices was discussed as early as 1982, with a modified Coke vending machine at Carnegie Mellon University becoming the first Internet-connected appliance, able to report its inventory and whether newly loaded drinks were cold or not. Mark Weiser's 1991 paper on ubiquitous computing, "The Computer of the 21st Century", as well as academic venues such as UbiComp and PerCom produced the contemporary vision of the IoT. In 1994, Reza Raji described the concept in IEEE Spectrum as "[moving] small packets of data to a large set of nodes, so as to integrate and automate everything from home appliances to entire factories". Between 1993 and 1997, several companies proposed solutions like Microsoft's at Work or Novell's NEST. The field gained momentum when Bill Joy envisioned device-to-device communication as a part of his "Six Webs" framework, presented at the World Economic Forum at Davos in 1999.

The term "Internet of things" was likely coined by Kevin Ashton of Procter & Gamble, later MIT's Auto-ID Center, in 1999, though he prefers the phrase "Internet for things". At that point, he viewed Radio-frequency identification (RFID) as essential to the Internet of things, which would allow computers to manage all individual things.

A research article mentioning the Internet of Things was submitted to the conference for Nordic Researchers in Norway, in June 2002, which was preceded by an article published in Finnish in January 2002. The implementation described there was developed by Kary Främling and his team at Helsinki University of Technology and more closely matches the modern one, i.e. an information system infrastructure for implementing smart, connected objects.

Defining the Internet of things as "simply the point in time when more 'things or objects' were connected to the Internet than people", Cisco Systems estimated that the IoT was "born" between 2008 and 2009, with the things/people ratio growing from 0.08 in 2003 to 1.84 in 2010.

# **Applications of IOT**

The extensive set of applications for IoT devices is often divided into consumer, commercial, industrial, and infrastructure spaces.

**Consumer applications**

A growing portion of IoT devices are created for consumer use, including connected vehicles, home automation, wearable technology (as part of Internet of Wearable Things (IoWT)), connected health, and appliances with remote monitoring capabilities.

* **Smart home**

IoT devices are a part of the larger concept of home automation, which can include lighting, heating and air conditioning, media and security systems. Long-term benefits could include energy savings by automatically ensuring lights and electronics are turned off.

A smart home or automated home could be based on a platform or hubs that control smart devices and appliances. For instance, using Apple's Home Kit, manufacturers can have their home products and accessories controlled by an application in iOS devices such as the iPhone and the Apple Watch. This could be a dedicated app or iOS native applications such as Siri. This can be demonstrated in the case of Lenovo's Smart Home Essentials, which is a line of smart home devices that are controlled through Apple's Home app or Siri without the need for a Wi-Fi bridge. There are also dedicated smart home hubs that are offered as standalone platforms to connect different smart home products and these include the Amazon Echo, Google Home, Apple's Home Pod, and Samsung's SmartThings Hub.

* **Elder care**

One key application of smart home is to provide assistance for those with disabilities and elderly individuals. These home systems use assistive technology to accommodate an owner's specific disabilities. Voice control can assist users with sight and mobility limitations while alert systems can be connected directly to cochlear implants worn by hearing impaired users. They can also be equipped with additional safety features. These features can include sensors that monitor for medical emergencies such as falls or seizures. Smart home technology applied in this way can provide users with more freedom and a higher quality of life.

The term "Enterprise IoT" refers to devices used in business and corporate settings. By 2019, it is estimated that the EIoT will account for 9.1 billion devices.

**Commercial application**

* **Medical and healthcare**

The Internet of Medical Things (also called the internet of health things) is an application of the IoT for medical and health related purposes, data collection and analysis for research, and monitoring. This 'Smart Healthcare', as it is also called, led to the creation of a digitized healthcare system, connecting available medical resources and healthcare services.

IoT devices can be used to enable remote health monitoring and emergency notification systems. These health monitoring devices can range from blood pressure and heart rate monitors to advanced devices capable of monitoring specialized implants, such as pacemakers, Fitbit electronic wristbands, or advanced hearing aids. Some hospitals have begun implementing "smart beds" that can detect when they are occupied and when a patient is attempting to get up. It can also adjust itself to ensure appropriate pressure and support is applied to the patient without the manual interaction of nurses. A 2015 Goldman Sachs report indicated that healthcare IoT devices "can save the United States more than $300 billion in annual healthcare expenditures by increasing revenue and decreasing cost." Moreover, the use of mobile devices to support medical follow-up led to the creation of 'm-health', used "to analyze, capture, transmit and store health statistics from multiple resources, including sensors and other biomedical acquisition systems".

Specialized sensors can also be equipped within living spaces to monitor the health and general well-being of senior citizens, while also ensuring that proper treatment is being administered and assisting people regain lost mobility via therapy as well. These sensors create a network of intelligent sensors that are able to collect, process, transfer, and analyses valuable information in different environments, such as connecting in-home monitoring devices to hospital-based systems. Other consumer devices to encourage healthy living, such as connected scales or wearable heart monitors, are also a possibility with the IoT. End-to-end health monitoring IoT platforms are also available for antenatal and chronic patients, helping one manage health vitals and recurring medication requirements.

* **Transportation**

Digital variable speed-limit sign.

The IoT can assist in the integration of communications, control, and information processing across various transportation systems. Application of the IoT extends to all aspects of transportation systems (i.e. the vehicle, the infrastructure, and the driver or user). Dynamic interaction between these components of a transport system enables inter- and intra-vehicular communication, smart traffic control, smart parking, electronic toll collection systems, logistic and fleet management, vehicle control, safety and road assistance. In Logistics and Fleet Management for example, an IoT platform can continuously monitor the location and conditions of cargo and assets via wireless sensors and send specific alerts when management exceptions occur (delays, damages, thefts, etc.). This can only be possible with the IoT and its seamless connectivity among devices. Sensors such as GPS, Humidity, and Temperature send data to the IoT platform and then the data is analyzed and then sent to the users. This way, users can track the real-time status of vehicles and can make appropriate decisions. If combined with Machine Learning, then it also helps in reducing traffic accidents by introducing drowsiness alerts to drivers and providing self-driven cars too.

V2X communications

IoT enables vehicle-to-everything communication (V2X), which consists of three main components of connected environment: vehicle to vehicle communication (V2V), vehicle to infrastructure communication (V2I) and vehicle to pedestrian communications (V2P). V2V empowers vehicles to exchange data, V2I allows them to network with the transport infrastructure (traffic signs and lights etc.) and V2P senses signals from the user’s smartphones to prevent collisions, involving pedestrians. By constantly analyzing real-time data, V2X designs a transport ecosystem where vehicles, infrastructure and people are interconnected with each other to keep the environment safe from any type of accidents. V2X is the first step to autonomous driving and connected road infrastructure that provides connected cars with needed safety.

* **Building and home automation**

IoT devices can be used to monitor and control the mechanical, electrical and electronic systems used in various types of buildings (e.g., public and private, industrial, institutions, or residential) in home automation and building automation systems. In this context, three main areas are being covered in literature:

The integration of the Internet with building energy management systems in order to create energy efficient and IOT-driven "smart buildings".

The possible means of real-time monitoring for reducing energy consumption and monitoring occupant behaviors.

The integration of smart devices in the built environment and how they might to know how to be used in future applications.

# **Enabling technologies for IoT**

There are many technologies that enable the IoT. Crucial to the field is the network used to communicate between devices of an IoT installation, a role that several wireless or wired technologies may fulfill:

* **Addressability**

The original idea of the Auto-ID Center is based on RFID-tags and distinct identification through the Electronic Product Code. This has evolved into objects having an IP address or URI. An alternative view, from the world of the Semantic Web focuses instead on making all things (not just those electronic, smart, or RFID-enabled) addressable by the existing naming protocols, such as URI. The objects themselves do not converse, but they may now be referred to by other agents, such as powerful centralized servers acting for their human owners. Integration with the Internet implies that devices will use an IP address as a distinct identifier. Due to the limited address space of IPv4 (which allows for 4.3 billion different addresses), objects in the IoT will have to use the next generation of the Internet protocol (IPv6) to scale to the extremely large address space required. Internet-of-things devices additionally will benefit from the stateless address auto-configuration present in IPv6, as it reduces the configuration overhead on the hosts, and the IETF 6LoWPAN header compression. To a large extent, the future of the Internet of things will not be possible without the support of IPv6; and consequently, the global adoption of IPv6 in the coming years will be critical for the successful development of the IoT in the future.

* **Short-range wireless**

Bluetooth mesh networking – Specification providing a mesh networking variant to Bluetooth low energy (BLE) with increased number of nodes and standardized application layer (Models).

Light-Fidelity (Li-Fi) – Wireless communication technology similar to the Wi-Fi standard, but using visible light communication for increased bandwidth.

Near-field communication (NFC) – Communication protocols enabling two electronic devices to communicate within a 4 cm range.

Radio-frequency identification (RFID) – Technology using electromagnetic fields to read data stored in tags embedded in other items.

Wi-Fi – technology for local area networking based on the IEEE 802.11 standard, where devices may communicate through a shared access point or directly between individual devices.

ZigBee – Communication protocols for personal area networking based on the IEEE 802.15.4 standard, providing low power consumption, low data rate, low cost, and high throughput.

Z-Wave - Wireless communications protocol used primarily for home automation and security applications

* **Medium-range wireless**

LTE-Advanced – High-speed communication specification for mobile networks. Provides enhancements to the LTE standard with extended coverage, higher throughput, and lower latency.

* **Long-range wireless**

**Low-power wide-area networking (LPWAN**) – Wireless networks designed to allow long-range communication at a low data rate, reducing power and cost for transmission. Available LPWAN technologies and protocols: LoRaWan, Sigfox, NB-IoT, Weightless.

**Very small aperture terminal (VSAT)** – Satellite communication technology using small dish antennas for narrowband and broadband data.

* **Wired**

**Ethernet** – General purpose networking standard using twisted pair and fiber optic links in conjunction with hubs or switches.

**Power-line communication (PLC)** – Communication technology using electrical wiring to carry power and data. Specifications such as HomePlug or G.hn utilize PLC for networking IoT devices.

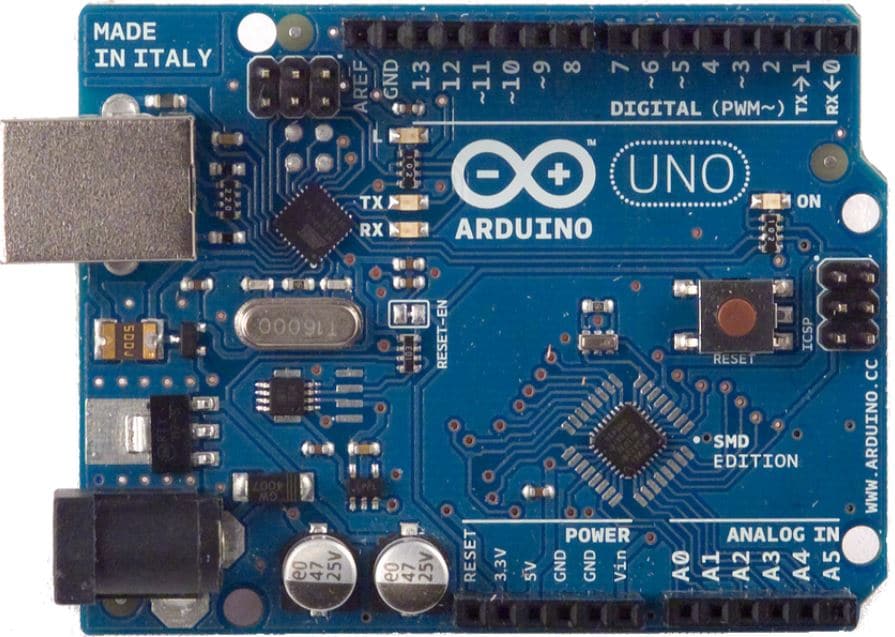
# **Components Used**

1. **Arduino Uno**
2. **2-Step 5v Relay**
3. **PIR Sensor**
4. **HC-05 Bluetooth Receiver**
5. **16x2 Led Display**
6. **5v LED**
7. **5v Internal Speaker**
8. **Breadboard**
9. **Jumper Cables**

# **Arduino Uno**

The Arduino UNO is an open-source microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino.cc. The board is equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The board has 14 Digital pins, 6 Analog pins, and programmable with the Arduino IDE (Integrated Development Environment) via a type B USB cable. It can be powered by a USB cable or by an external 9-volt battery, though it accepts voltages between 7 and 20 volts. It is also similar to the Arduino Nano and Leonardo. The hardware reference design is distributed under a 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.

"Uno" means one in Italian and was chosen to mark the release of Arduino Software (IDE) 1.0. The Uno board and version 1.0 of Arduino Software (IDE) were the reference versions of Arduino, now evolved to newer releases. The Uno board is the first in a series of USB Arduino boards, and the reference model for the Arduino platform. The ATmega328 on the Arduino Uno comes preprogrammed with a bootloader that allows uploading new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol. The Uno also 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://www.arrow.com/en/products/a000073/arduino-corporation)**Technical Specifications**

* Microcontroller - Microchip 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 - 32 KB of which 0.5 KB used by bootloader
* SRAM - 2 KB
* EEPROM - 1 KB
* Clock Speed - 16 MHz
* Length - 68.6 mm
* Width - 53.4 mm
* Weight - 25 g

**Pin Specification**

**Digital Pins**

In addition to the specific functions listed below, the digital pins on an Arduino board can be used for general purpose input and output via the pinMode(), digitalRead(), and digitalWrite() commands. Each pin has an internal pull-up resistor which can be turned on and off using digitalWrite() (w/ a value of HIGH or LOW, respectively) when the pin is configured as an input. The maximum current per pin is 40 mA.

**Serial**: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. On the Arduino Diecimila, these pins are connected to the corresponding pins of the FTDI USB-to-TTL Serial chip. On the Arduino BT, they are connected to the corresponding pins of the WT11 Bluetooth module. On the Arduino Mini and LilyPad Arduino, they are intended for use with an external TTL serial module (e.g. the Mini-USB Adapter).

**External Interrupts**: 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. See the attachInterrupt() function for details.

**PWM**: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analogWrite() function. On boards with an ATmega8, PWM output is available only on pins 9, 10, and 11.

**BT Reset**: 7. (Arduino BT-only) Connected to the reset line of the bluetooth module.

SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language.

LED: 13. On the Diecimila and LilyPad, 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.

**Analog Pins**

In addition to the specific functions listed below, the analog input pins support 10-bit analog-to-digital conversion (ADC) using the analogRead() function. Most of the analog inputs can also be used as digital pins: analog input 0 as digital pin 14 through analog input 5 as digital pin 19. Analog inputs 6 and 7 (present on the Mini and BT) cannot be used as digital pins.

**I2C**: 4 (SDA) and 5 (SCL). Support I2C (TWI) communication using the Wire library (documentation on the Wiring website).

**Power Pins**

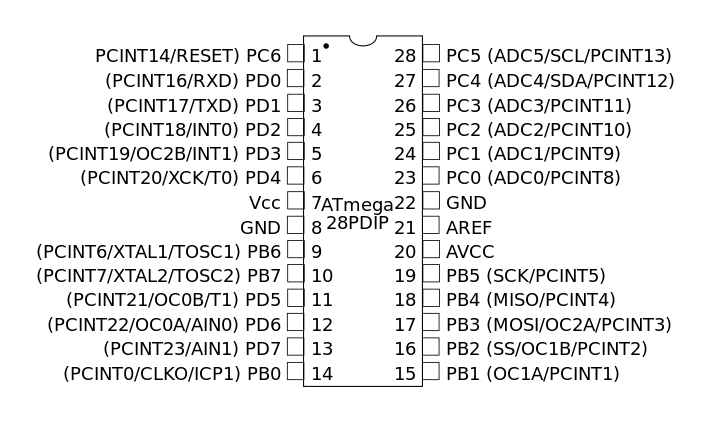
**VIN** (sometimes labelled "9V"). The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin. Note that different boards accept different input voltages ranges, please see the documentation for your board. Also note that the LilyPad has no VIN pin and accepts only a regulated input.

**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.

**GND**. Ground pins.

# **Microchip ATmega328P**

The ATmega328 is a single-chip microcontroller created by Atmel in the megaAVR family (later Microchip Technology acquired Atmel in 2016). It has a modified Harvard architecture 8-bit RISC processor core.

 The Atmel 8-bit AVR RISC-based microcontroller combines 32 kB ISP flash memory with read-while-write capabilities, 1 kB EEPROM, 2 kB SRAM, 23 general purpose I/O lines, 32 general purpose working registers, three flexible timer/counters with compare modes, internal and external interrupts, serial programmable USART, a byte-oriented 2-wire serial interface, SPI serial port, 6-channel 10-bit A/D converter (8-channels in TQFP and QFN/MLF packages), programmable watchdog timer with internal oscillator, and five software selectable power saving modes. The device operates between 1.8-5.5 volts. The device achieves throughput approaching 1 MIPS per MHz.

ATmega328P Pin Diag

# **Key parameters**

|  |  |
| --- | --- |
| **Parameter** | **Value** |
| CPU type | 8-bit AVR |
| Performance | 20 MIPS at 20 MHz |
| Flash memory | 32 kB |
| SRAM | 2 kB |
| EEPROM | 1 kB |
| Pin count | 28 or 32 pins: PDIP-28, MLF-28, TQFP-32, MLF-32 |
| Maximum operating frequency | 20 MHz |
| Number of touch channels | 16 |
| Hardware QTouch Acquisition | No |
| Maximum I/O pins | 23 |
| External interrupts | 2 |
| USB Interface | No |
| USB Speed | – |

# [2cha1.jpg](http://wiki.sunfounder.cc/index.php?title=File:2cha1.jpg)**2-Step 5v Relay**

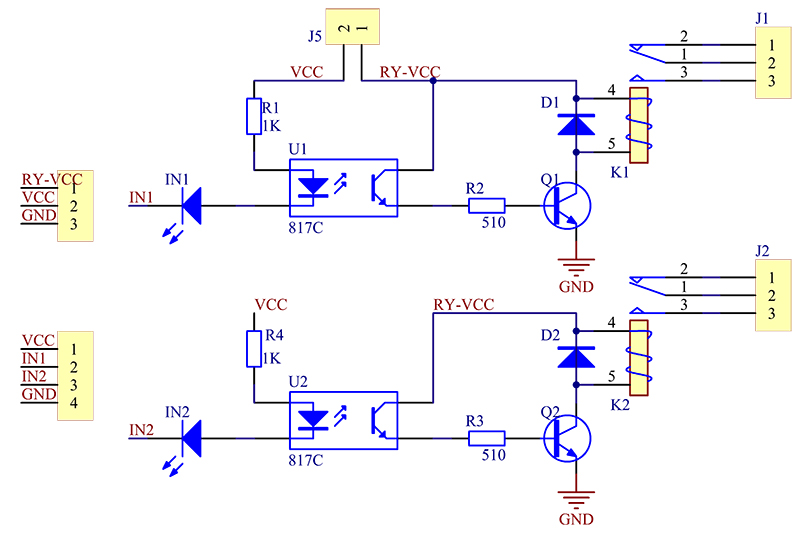
This is a LOW Level 5V 2-channel relay interface board, and each channel needs a 15-20mA driver current. It can be used to control various appliances and equipment with large current. It is equiped with high-current relays that work under AC250V 10A or DC30V 10A. It has a standard interface that can be controlled directly by microcontroller.

**Features**

* Relay Maximum output: DC 30V/10A, AC 250V/10A
* 2 Channel Relay Module with Optocoupler LOW Level Triger expansion board, which is compatible with arduino
* Standard interface that can be controlled directly by microcontroller ( 8051, AVR, \*PIC, DSP, ARM, ARM, MSP430, TTL logic)
* Relay of high quality loose music relays SPDT. A common terminal, a normally open, one normally closed terminal
* optocoupler isolation, good anti-jamming

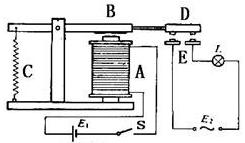
**Schematic**

VCC and RY-VCC are also the power supply of the relay module. When you need to drive a large power load, you can take the jumper cap off and connect an extra power to RY-VCC to supply the relay; connect VCC to 5V of the MCU board to supply input signals.

[](http://wiki.sunfounder.cc/index.php?title=File:2cha2.jpg)

**Principle**

See the picture below: A is an electromagnet, B armature, C spring, D moving contact, and E fixed contacts. There are two fixed contacts, a normally closed one and a normally open one. When the coil is not energized, the normally open contact is the one that is off, while the normally closed one is the other that is on.

[](http://wiki.sunfounder.cc/index.php?title=File:2cha3.jpg)

Add a certain voltage to the coil and some currents will pass through the coil thus generating the electromagnetic effect. So the armature overcomes the tension of the spring and is attracted to the core, thus closing the moving contact of the armature and the normally open contact (or you may say releasing the former and the normally closed contact). After the coil is de-energized, the electromagnetic force disappears and the armature moves back to the original position, releasing the moving contact and normally closed contact. The closing and releasing of the contacts results in power on and off of the circuit.

**Input:**

VCC : Connected to positive supply voltage (supply power according to relay voltage)  
GND : Connected to negative supply voltage  
IN1: Signal triggering terminal 1 of relay module  
IN2: Signal triggering terminal 2 of relay module

**Output:**

Each submodular of the relay has one NC(nomalclose), one NO(nomalopen) and one COM(Common). So there are 2 NC, 2 NO and 2 COM of the channel relay in total. NC stands for the normal close port contact and the state without power; No stands for the normal open port contact and the state with power. COM means the common port. You can choose NC port or NO port according to whether power or not.

# **Passive Infrared Sensor**

A passive infrared sensor (PIR sensor) is an electronic sensor that measures infrared (IR) light radiating from objects in its field of view. They are most often used in PIR-based motion detectors. PIR sensors are commonly used in security alarms and automatic lighting applications. PIR sensors detect general movement, but do not give information on who or what moved. For that purpose, an active IR sensor is required.

PIR sensors are commonly called simply "PIR", or sometimes "PID", for "passive infrared detector". The term passive refers to the fact that PIR devices do not radiate energy for detection purposes. They work entirely by detecting infrared radiation (radiant heat) emitted by or reflected from objects.

All objects with a temperature above absolute zero emit heat energy in the form of radiation. Usually this radiation isn't visible to the human eye because it radiates at infrared wavelengths, but it can be detected by electronic devices designed for such a purpose.

[](https://www.rhydolabz.com/sensors-ir-pir-sensors-c-137_150/pir-motion-detection-sensor-hcsr501-pr-1512.html)Infrared radiation enters through the front of the sensor, known as the 'sensor face'. At the core of a PIR sensor is a solid state sensor or set of sensors, made from pyroelectric materials—materials which generate energy when exposed to heat. Typically, the sensors are approximately 1/4 inch square (40 mm2), and take the form of a thin film. Materials commonly used in PIR sensors include gallium nitride (GaN), caesium nitrate (CsNO3), polyvinyl fluorides, derivatives of phenylpyridine, and cobalt phthalocyanine. The sensor is often manufactured as part of an integrated circuit.

# **Operation**

A PIR sensor can detect changes in the amount of infrared radiation impinging upon it, which varies depending on the temperature and surface characteristics of the objects in front of the sensor.[2] When an object, such as a person, passes in front of the background, such as a wall, the temperature at that point in the sensor's field of view will rise from room temperature to body temperature, and then back again. The sensor converts the resulting change in the incoming infrared radiation into a change in the output voltage, and this triggers the detection. Objects of similar temperature but different surface characteristics may also have a different infrared emission pattern, and thus moving them with respect to the background may trigger the detector as well.[3]

PIRs come in many configurations for a wide variety of applications. The most common models have numerous Fresnel lenses or mirror segments, an effective range of about 10 meters (30 feet), and a field of view less than 180°. Models with wider fields of view, including 360°, are available, typically designed to mount on a ceiling. Some larger PIRs are made with single segment mirrors and can sense changes in infrared energy over 30 meters (100 feet) from the PIR. There are also PIRs designed with reversible orientation mirrors which allow either broad coverage (110° wide) or very narrow "curtain" coverage, or with individually selectable segments to "shape" the coverage.

# **HC-05 Bluetooth Receiver**

**HC‐05 module** is an easy to use **Bluetooth SPP (Serial Port Protocol) module**, designed for transparent wireless serial connection setup. The HC-05 Bluetooth Module can be used in a Master or Slave configuration, making it a great solution for wireless communication. This serial port Bluetooth module is fully qualified **Bluetooth V2.0+EDR (Enhanced Data Rate)** 3Mbps Modulation with complete 2.4GHz radio transceiver and baseband. It uses **CSR Bluecore 04**‐External single chip Bluetooth system with CMOS technology and with AFH (Adaptive Frequency Hopping Feature).

**Bluetooth Module HC-05**

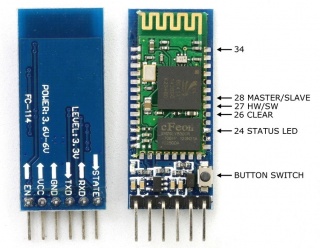
The Bluetooth module HC-05 is a MASTER/SLAVE module. By default, the factory setting is SLAVE. The Role of the module (Master or Slave) can be configured only by AT COMMANDS. The slave modules cannot initiate a connection to another Bluetooth device, but can accept connections. Master module can initiate a connection to other devices. The user can use it simply for a serial port replacement to establish connection between MCU and GPS, PC to your embedded project, etc.

**Hardware Features**

* Typical ‐80dBm sensitivity.
* Up to +4dBm RF transmit power.
* 3.3 to 5 V I/O.
* PIO (Programmable Input/Output) control.
* UART interface with programmable baud rate.
* With integrated antenna.
* With edge connector.

**Software Features**

* Slave default Baud rate: 9600, Data bits:8, Stop bit:1, Parity: No parity.
* Auto‐connect to the last device on power as default.
* Permit pairing device to connect as default.
* Auto‐pairing PINCODE:”1234” as default.

[](https://wiki.eprolabs.com/index.php?title=File:FC-114.jpg)

**Pin Description**

The HC-05 Bluetooth Module has 6pins. They are as follows:

**ENABLE:**

When enable is pulled **LOW**, the module is disabled which means the module will **not turn on** and it **fails to communicate**. When enable is **left open or connected to 3.3V**, the module is enabled i.e. the module **remains on** and **communication also takes place**.

**Vcc:**

Supply Voltage 3.3V to 5V

**GND:**

Ground pin

**TXD & RXD:**

These two pins act as an UART interface for communication

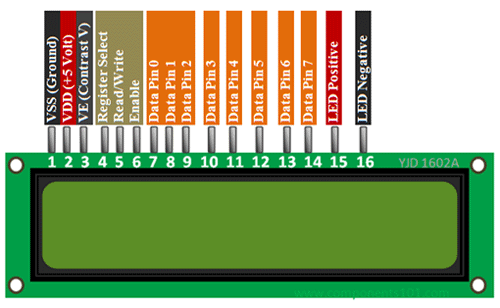
**STATE:**

It acts as a status indicator. When the module is **not connected to / paired** with any other Bluetooth device, signal goes **Low**. At this **low state**, the **led flashes continuously** which denotes that the module is **not paired** with another device. When this module is **connected to/paired** with any other Bluetooth device, the signal goes **High**. At this **high state**, the **led blinks with a constant delay** say for example 2s delay which indicates that the module is **paired**.

**BUTTON SWITCH:**

This is used to switch the module into AT command mode. To enable AT command mode, press the button switch for a second. With the help of AT commands, the user can change the parameters of this module but only when the module is not paired with any other BT device. If the module is connected to any other Bluetooth device, it starts to communicate with that device and fails to work in AT command mode.

# **16x2 LCD Module**

[](https://components101.com/sites/default/files/component_pin/16x2-LCD-Pinout.png)

**16x2 LCD Module Pinout**

**Pin Configuration**

|  |  |  |
| --- | --- | --- |
| **Pin No:** | **Pin Name:** | **Description** |
| 1 | Vss (Ground) | Ground pin connected to system ground |
| 2 | Vdd (+5 Volt) | Powers the LCD with +5V (4.7V – 5.3V) |
| 3 | VE (Contrast V) | Decides the contrast level of display. Grounded to get maximum contrast. |
| 4 | Register Select | Connected to Microcontroller to shit between command/data register |
| 5 | Read/Write | Used to read or write data. Normally grounded to write data to LCD |
| 6 | Enable | Connected to Microcontroller Pin and toggled between 1 and 0 for data acknowledgement |
| 7 | Data Pin 0 | Data pins 0 to 7 forms a 8-bit data line. They can be connected to Microcontroller to send 8-bit data.  These LCD’s can also operate on 4-bit mode in such case Data pin 4,5,6 and 7 will be left free. |
| 8 | Data Pin 1 |
| 9 | Data Pin 2 |
| 10 | Data Pin 3 |
| 11 | Data Pin 4 |
| 12 | Data Pin 5 |
| 13 | Data Pin 6 |
| 14 | Data Pin 7 |
| 15 | LED Positive | Backlight LED pin positive terminal |
| 16 | LED Negative | Backlight LED pin negative terminal |

**Features of 16×2 LCD module**

* Operating Voltage is 4.7V to 5.3V
* Current consumption is 1mA without backlight
* Alphanumeric LCD display module, meaning can display alphabets and numbers
* Consists of two rows and each row can print 16 characters.
* Each character is build by a 5×8 pixel box
* Can work on both 8-bit and 4-bit mode
* It can also display any custom generated characters
* Available in Green and Blue Backlight

**16x2 Display Equivalents**

Dot Matrix LED Display, 7-Segment LED Display, OLED Display, TFT LCD Screen Display

**Brief Description on LCD modules**

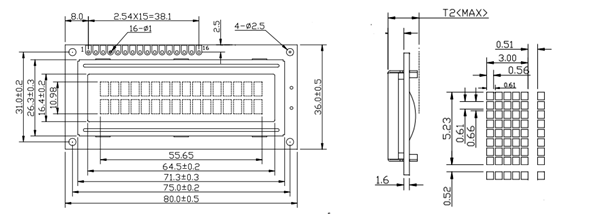
LCD modules are vey commonly used in most embedded projects, the reason being its cheap price, availability and programmer friendly. Most of us would have come across these displays in our day to day life, either at PCO’s or calculators. The appearance and the pinouts have already been visualized above now let us get a bit technical.

**16×2 LCD** is named so because; it has 16 Columns and 2 Rows. There are a lot of combinations available like, 8×1, 8×2, 10×2, 16×1, etc. but the most used one is the 16×2 LCD. So, it will have (16×2=32) 32 characters in total and each character will be made of 5×8 Pixel Dots.  A Single character with all its Pixels is shown in the below picture.

16x2 LCD Pixel

Now, we know that each character has (5×8=40) 40 Pixels and for 32 Characters we will have (32×40) 1280 Pixels. Further, the LCD should also be instructed about the Position of the Pixels. Hence it will be a hectic task to handle everything with the help of MCU, hence an **Interface IC like HD44780**is used, which is mounted on the backside of the LCD Module itself. The function of this IC is to get the **Commands and Data** from the MCU and process them to display meaningful information onto our LCD Screen. You can learn how to interface an LCD using the above-mentioned links. If you are an advanced programmer and would like to create your own library for interfacing your Microcontroller with this LCD module then you have to understand the HD44780 IC is working and commands which can be found its datasheet.

**2D model of 16×2 LCD module**



# [https://upload.wikimedia.org/wikipedia/commons/thumb/1/19/Electronics-White-Breadboard.jpg/220px-Electronics-White-Breadboard.jpg](https://en.wikipedia.org/wiki/File:Electronics-White-Breadboard.jpg)**Breadboard**

A breadboard is a construction base for prototyping of electronics. Originally the word referred to a literal bread board, a polished piece of wood used for slicing bread. In the 1970s the solderless breadboard (a.k.a. plugboard, a terminal array board) became available and nowadays the term "breadboard" is commonly used to refer to these.

Because the solderless breadboard does not require soldering, it is reusable. This makes it easy to use for creating temporary prototypes and experimenting with circuit design. For this reason, solderless breadboards are also popular with students and in technological education. Older breadboard types did not have this property. A stripboard (Veroboard) and similar prototyping printed circuit boards, which are used to build semi-permanent soldered prototypes or one-offs, cannot easily be reused. A variety of electronic systems may be prototyped by using breadboards, from small analog and digital circuits to complete central processing units (CPUs).

# **Configuration Diagram**

**The Code**

/\* This is a proof of concept for "IOT based Home Automation Systems"

\* The code is written int the Aurdrino programming language

\* The code has only been tested on a Aurduino Uno

\* The corrosponding Android appilication for the control of the items is available online

\* For any further doubts or inquiries contact at "parthsvas@gmail.com" & "abhinathshrotriya29@gmail.com"

\*/

// Loading the library needed to run the LCD Display

#include <LiquidCrystal.h>

// Creating variables for pin numbers to increse understanding of the code

int r1 = 6;

int r2 = 5;

int spk = 4;

int led = 3;

int sensor = 2;

char data = 0;

// Assigning different pins of the LCD Display to different pins for output

int rs = 7, en = 8, d4 = 9, d5 = 10, d6 = 11, d7 = 12;

LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

void setup()

{

// Seting the baud rate for serial data transmission

Serial.begin(9600);

// Setting the pins according to their I/O requirement

pinMode(2, INPUT);

pinMode(3, OUTPUT);

pinMode(4, OUTPUT);

pinMode(5, OUTPUT);

pinMode(6, OUTPUT);

// Printing a message on the LCD Display

lcd.begin(16, 2);

lcd.setCursor(0, 0);

lcd.print("CSE Diploma 6th");

lcd.setCursor(0, 1);

lcd.print("Major Project");

}

void loop()

{

if (Serial.available() > 0)

{

data = Serial.read();

Serial.print(data);

Serial.print("\n");

lcd.clear();

lcd.setCursor(2, 0);

lcd.print("BT Connected");

}

// Setting up a Switch-Case statement for the various inputs to their actions and printing a notification on the LCD Display

switch (data)

{

case 'a':

lcd.setCursor(2, 1);

lcd.print("Relay 1 Off");

digitalWrite(r1, LOW);

break;

case 'b':

lcd.setCursor(2, 1);

lcd.print("Relay 1 On");

digitalWrite(r1, HIGH);

break;

case 'c':

lcd.setCursor(2, 1);

lcd.print("Relay 2 Off");

digitalWrite(r2, LOW);

break;

case 'd':

lcd.setCursor(2, 1);

lcd.print("Relay 2 On");

digitalWrite(r2, HIGH);

break;

case 'e':

lcd.setCursor(2, 1);

lcd.print("Speaker On");

digitalWrite(spk, HIGH);

break;

case 'f':

lcd.setCursor(2, 1);

lcd.print("Speaker Off");

digitalWrite(spk, LOW);

break;

}

// Setting up the PIR Sensor to respond to any motion

int sensorval = digitalRead(sensor);

Serial.println(sensorval);

if (sensorval == HIGH) {

digitalWrite(led, LOW);

}

else {

digitalWrite(led, HIGH);

}

}

# **Arduino IDE**

Arduino programs are written in the Arduino Integrated Development Environment (IDE). Arduino IDE is a special software running on your system that allows you to write sketches (synonym for program in Arduino language) for different Arduino boards. The Arduino programming language is based on a very simple hardware programming language called processing, which is similar to the C language. After the sketch is written in the Arduino IDE, it should be uploaded on the Arduino board for execution.

The first step in programming the Arduino board is downloading and installing the Arduino IDE. The open source Arduino IDE runs on Windows, Mac OS X, and Linux. Download the Arduino software (depending on your OS) from the official website and follow the instructions to install.

Now let’s discuss the basics of Arduino programming.

The structure of Arduino program is pretty simple. Arduino programs have a minimum of 2 blocks,

Preparation & Execution

Each block has a set of statements enclosed in curly braces:

void setup( )

{

statements-1;

.

.

statement-n;

}

void loop ( )

{

statement-1;

.

.

.

statement-n;

}

Here, setup ( ) is the preparation block and loop ( ) is an execution block.

The setup function is the first to execute when the program is executed, and this function is called only once. The setup function is used to initialize the pin modes and start serial communication. This function has to be included even if there are no statements to execute.

void setup ( )

{

pinMode (pin-number, OUTPUT); // set the ‘pin-number’ as output

pinMode (pin-number, INPUT); // set the ‘pin-number’ as output

}

After the setup ( ) function is executed, the execution block runs next. The execution block hosts statements like reading inputs, triggering outputs, checking conditions etc..

In the above example loop ( ) function is a part of execution block. As the name suggests, the loop( ) function executes the set of statements (enclosed in curly braces) repeatedly.

Void loop ( )

{

digitalWrite (pin-number,HIGH); // turns ON the component connected to ‘pin-number’

delay (1000); // wait for 1 sec

digitalWrite (pin-number, LOW); // turns OFF the component connected to ‘pin-number’

delay (1000); //wait for 1sec

}

Note: Arduino always measures the time duration in millisecond. Therefore, whenever you mention the delay, keep it in milli seconds.

# **Project Summary**

This is a proof of concept for “IOT Based Home Automation Systems” is a project developed and researched by Parth Srivastava and Abhinath Shrotriya, under the guidance of our teachers and faculty. A special thanks to Mr Arpit Sharma and Dr Raj Kumar for introducing us the world of IOT and its multitude of applications.

The project uses the configuration described in the diagram and the Arduino is programmed using the code referenced.

The Arduino is the main hub and the processing and the memory unit. Its takes input from various sources like the HC-05 Bluetooth Module and the PIR Sensor. The input then is processed by the microcontroller and the output is generated according to the instructions stored in the memory of the Arduino.

The PIR sensor introduces the main concept of complete wireless autonomy. The idea can further be utilised to include more complicated tasks as if the sensor detects movement after a certain time it can send a notification to the user or activate lights or trigger another event. The most practical use of the PIR sensor is useful for security purposes.

The Relays used in the project act like a simple switch and can thereby used to switch on or off any electronic devices. In the project the input is provided by the user through Bluetooth but the relays can be triggered by any event e.g. from a PIR sensor or a magnetic latch that can detect if a window or a door is opened. It can be used to activate any electronic device no matter its capacity or load by a simple event trigger.

Many other sensors or communication devices can be used in tandem with Arduino like microcontrollers e.g. A Wi-Fi module can connected to the Arduino that lets the Arduino connect to the internet and receive its input from there this in theory increases the range of the project to unlimited heights.

The applications of this concept us endless in home automation from security to increase in quality of life. This proof of concept cost around ₹1000+, so with a litte streamlining and creation of less complicated or a single PCB/Microcontroller can reduce the cost of the product up to by 25%.

**Acknowledgements and References**

* <https://www.wikipedia.org/>
* <https://www.arduino.cc/>
* <https://www.arduino.cc/en/Main/Education>
* <https://www.geeksforgeeks.org/>
* <https://stackoverflow.com/>
* <https://www.google.com/>

**Created by Parth Srivastava and Abhinath Shrotriya of Diploma CSE 6th Sem**