**HOME AUTOMATION USING STM32**

**REPORT**

*Submitted in Partial Fulfillment of the*

*Requirement of the degree*

**BACHELOR OF TECHNOLOGY**

**IN**

**ELECTRONICS & COMMUNICATION ENGINEERING**

By

**SAURABH JAIN**

14BEC107

Under the Guidance of

Prof. BHUPENDRA FATANIYA



**Department of Electrical Engineering**

**Electronics & Communication Engineering Program**

**Institute of Technology, Nirma University**

# CERTIFICATE

This is to certify that the mini project entitled **“HOME AUTOMATION USING STM32 “**submitted by **SAURABH JAIN (**Roll No. **14BEC107)** as the partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Electronics & Communication Engineering, Institute of Technology, Nirma University. It is the record of work carried out by them under my supervision and guidance. The work submitted in our opinion has reached a level required for being accepted for the examination.

**Date: 04/05/2017**

**Prof. BHUPENDRA FATANIYA**

**MINI project guide**

**Prof. (Dr.) D. K. Kothari Prof. (Dr.) Alka Mahajan**

**Section Head (EC) HOD (EE /EC / IC)**

**ACKNOWLEDGEMENT**

It gives us great pleasure to present our Mini project report on “**HOME AUTOMATION USING STM32**”.

We would like to express our deep gratitude towards **Prof. BHUPENDRA FATANIYA**, who gave us his valuable suggestions, motivation and direction to proceed at each step. He gave us his valuable guidance, indispensable help and inspiration at times in appreciation we offer him our sincere gratitude

## Index

**Chapter Title**

**No.**

**Acknowledgement**

**Abstract**

**Index**

1. **ABOUT STM32 BOARD**
   1. Theory
   2. Key Features
2. **BLUETOOTH MODULE(HC05)**
3. HARDWARE FEATURES
4. SOFTWARE FEATURES
5. PIN DESCRIPTION
6. UART COMMUNICATION
7. ADVANTAGES OF BLUETOOTH MODULE
8. APPLICATION OF BLUETOOTH MODULE
9. INTERFACING BLUETOOTH WITH STM32
10. **PIR SENSORS**
11. ABOUT
12. WORKING
13. SPECFICATION AND PIN DIAGRAM
14. **CIRCUIT DIAGRAM**
15. ABOUT
16. THE OPTO-TRIAC
17. TRIAC OPERATION
18. APPLICATIONS
19. **CONCLUSION AND FUTURE SCOPE,REFERENCES**

### 

### LIST OF FIGURES

**Fig. No. Title Page No .**

Figure1: Quadrant1 Operation of Triac 8

Figure2: Triac Equivalent Circuit 8

Figure3: Quadrant2 Opreation of Traic 9

Figure4: Quadrant 3 Operation of Triac 10

Figure5: Quadrant 4 Operation of Triac 10

Figure6: Characteristics of Triac 11

Figure7: Practical regulator implemented with Triac 12

Figure8: Optotriac 13

Figure9: Diac 13

Figure10: Characteristics of Diac 14

Figure11: Pin Diagram of TSOP1738 16

Figure12: Arduino 18

Figure13: Circuit Diagram 20

Figure14: Zero Crossing Detector 20

Figure15: Waveform of Zero crossing detector 2

**HOME AUTOMATION USING STM32**

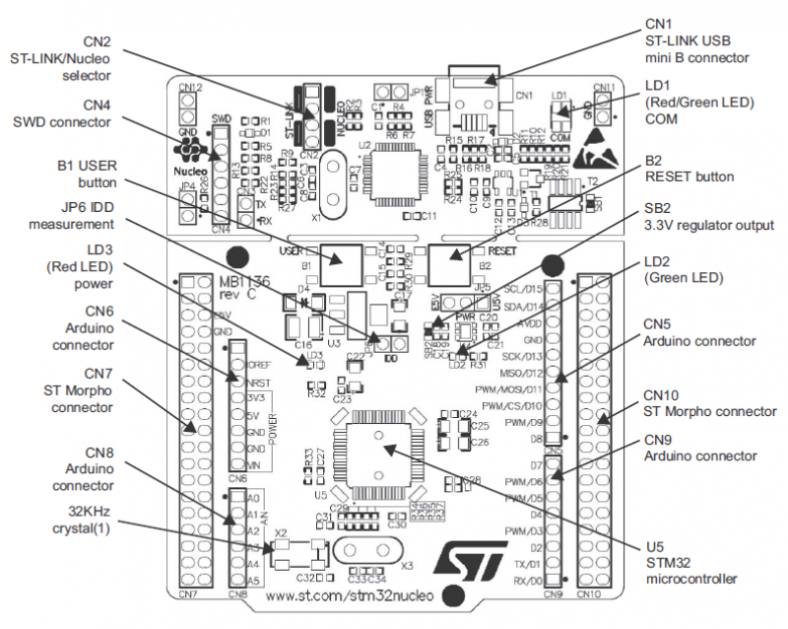
**CHAPTER-1**

**ABOUT STM32 BOARD**

1. **Theory**

The STM32 F4-series is the first group of STM32 microcontrollers based on the ARM Cortex-M4F core. The F4-series is also the first STM32 series to have DSP and floating point instructions. The F4 is pin-to-pin compatible with the STM32 F2-series and adds higher clock speed, 64 KB CCM static RAM, full duplex I²S, improved real-time clock, and faster ADCs. The summary for this series is:

* Core:
* [ARM Cortex-M4F](https://en.wikipedia.org/wiki/ARM_Cortex-M4F) core at a maximum clock rate of 84 / 168 / 180 MHz
* Memory:
  + [Static RAM](https://en.wikipedia.org/wiki/Static_RAM) consists of up to 192 KB general purpose, 64 KB core coupled memory (CCM), 4 KB battery-backed, 80 bytes battery-backed with tamper-detection erase.
  + Flash consists of 512 / 1024 / 2048 [KB](https://en.wikipedia.org/wiki/Kilobyte) general purpose, 30 KB system boot, 512 bytes one-time programmable (OTP), 16 option bytes.
  + Each chip has a factory-programmed 96-bit unique device identifier number.
* Peripherals:
* Common peripherals included in all IC packages are [USB](https://en.wikipedia.org/wiki/USB) 2.0 [OTG](https://en.wikipedia.org/wiki/USB_On-The-Go) HS and FS, two [CAN](https://en.wikipedia.org/wiki/Controller_area_network) 2.0B, one [SPI](https://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus) + two SPI or full-duplex [I²S](https://en.wikipedia.org/wiki/I%C2%B2S), three [I²C](https://en.wikipedia.org/wiki/I%C2%B2C), four [USART](https://en.wikipedia.org/wiki/USART), two [UART](https://en.wikipedia.org/wiki/UART), [SDIO](https://en.wikipedia.org/wiki/Secure_Digital#SDIO) for [SD](https://en.wikipedia.org/wiki/Secure_Digital)/[MMC](https://en.wikipedia.org/wiki/MultiMediaCard) cards, twelve 16-bit [timers](https://en.wikipedia.org/wiki/Timers#Computer_timers), two 32-bit timers, two [watchdog](https://en.wikipedia.org/wiki/Watchdog_timer) timers, [temperature](https://en.wikipedia.org/wiki/Temperature) sensor, 16 or 24 channels into three [ADCs](https://en.wikipedia.org/wiki/Analog-to-digital_converter), two [DACs](https://en.wikipedia.org/wiki/Digital-to-analog_converter), 51 to 140 [GPIOs](https://en.wikipedia.org/wiki/General_Purpose_Input/Output), sixteen [DMA](https://en.wikipedia.org/wiki/Direct_memory_access), improved real-time clock ([RTC](https://en.wikipedia.org/wiki/Real-time_clock)), [cyclic redundancy check](https://en.wikipedia.org/wiki/Cyclic_redundancy_check) (CRC) engine, [random number generator](https://en.wikipedia.org/wiki/Random_number_generation) (RNG) engine. Larger IC packages add 8/16-bit external [memory bus](https://en.wikipedia.org/wiki/Memory_bus) capabilities.
* The STM32F4x7 models add [Ethernet](https://en.wikipedia.org/wiki/Ethernet) [MAC](https://en.wikipedia.org/wiki/Media_Independent_Interface) and [camera interface](https://en.wikipedia.org/wiki/Camera_interface).
* The STM32F41x/43x models add a [cryptographic processor](https://en.wikipedia.org/wiki/Cryptographic_accelerator) for [DES](https://en.wikipedia.org/wiki/Data_Encryption_Standard) / [TDES](https://en.wikipedia.org/wiki/Triple_DES) / [AES](https://en.wikipedia.org/wiki/Advanced_Encryption_Standard), and a hash processor for [SHA-1](https://en.wikipedia.org/wiki/SHA-1) and [MD5](https://en.wikipedia.org/wiki/MD5).
* The STM32F4x9 models add a [LCD-TFT](https://en.wikipedia.org/wiki/TFT_LCD) controller.
* [Oscillators](https://en.wikipedia.org/wiki/Electronic_oscillator) consists of internal (16 MHz, 32 kHz), optional external (4 to 26 MHz, 32.768 to 1000 kHz).
* [IC packages](https://en.wikipedia.org/wiki/Integrated_circuit_packaging): [WLCSP](https://en.wikipedia.org/wiki/WLCSP)64, [LQFP](https://en.wikipedia.org/wiki/LQFP)64, LQFP100, LQFP144, LQFP176, [UFBGA](https://en.wikipedia.org/wiki/UFBGA)176. STM32F429/439 also offers LQFP208 and [UFBGA](https://en.wikipedia.org/wiki/UFBGA)216.
* Operating [voltage](https://en.wikipedia.org/wiki/IC_power_supply_pin) range is 1.8 to 3.6 [volt](https://en.wikipedia.org/wiki/Volt).
  1. **Key Features**
* STM32F407VGT6 microcontroller featuring 32-bit ARM® Cortex® -M4 with FPU core, 1-Mbyte Flash memory, 192-Kbyte RAM in an LQFP100 package
* On-board ST-LINK/V2 on [STM32F4DISCOVERY](http://www.st.com/en/evaluation-tools/stm32f4discovery.html) (old reference) or ST-LINK/V2-A on [STM32F407G-DISC1](http://www.st.com/en/evaluation-tools/stm32f4discovery.html) (new order code)
* USB ST-LINK with re-enumeration capability and three different interfaces:
  + Debug port
  + Virtual Com port (with new order code only)
  + Mass storage (with new order code only)
* Board power supply: through USB bus or from an external 5 V supply voltage
* External application power supply: 3 V and 5 V
* LIS302DL or LIS3DSH ST MEMS 3-axis accelerometer
* MP45DT02 ST-MEMS audio sensor omni-directional digital microphone
* CS43L22 audio DAC with integrated class D speaker driver
* Eight LEDs:
  + LD1 (red/green) for USB communication
  + LD2 (red) for 3.3 V power on
  + Four user LEDs, LD3 (orange), LD4 (green), LD5 (red) and LD6 (blue)
  + 2 USB OTG LEDs LD7 (green) VBUS and LD8 (red) over-current
* Two push-buttons (user and reset)
* USB OTG FS with micro-AB connector
* Extension header for all LQFP100 I/Os for quick connection to prototyping board and easy probing
* Comprehensive free software including a variety of examples, part of STM32CubeF4 package or STSW-STM32068 to use legacy standard libraries.



**CHAPTER-2**

**BLUETOOTH MODULE (HC05)**

HC‐05 module is an easy to use Bluetooth SPP (Serial Port Protocol) module, designed for transparent wireless serial connection setup. 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.Just go through the datasheet for more details

**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.

**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.
* Auto‐pairing PINCODE:”1234” as default.

## Pin Description

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

1. **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.

1. **Vcc:**

Supply Voltage 3.3V to 5V

1. **GND:**

Ground pin

1. **TXD & RXD:**

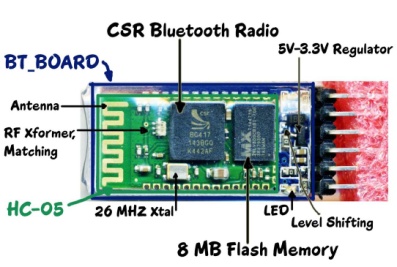
These two pins acts as an UART interface for communication

1. **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 other 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.

1. **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.



**UART COMMUNICATION**

A universalasynchronous receiver/transmitter (UART), is a computerhardware device for asynchronous serial communication in which the data format and transmission speeds are configurable.

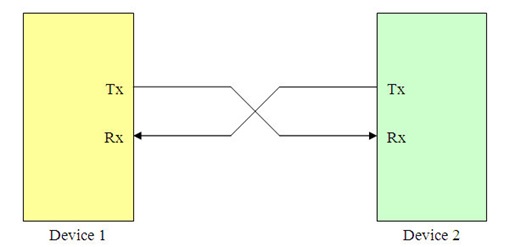
It provides the computer with the RS-232C Data Terminal Equipment ( DTE ) interface so that it can "talk" to and exchange data with modems and other serial devices.

Converts the bytes it receives from the computer along parallel circuits into a single serial bit stream for outbound transmission

On inbound transmission, converts the serial bit stream into the bytes that the computer handles

Adds a parity bit (if it's been selected) on outbound transmissions and checks the parity of incoming bytes (if selected) and discards the parity bit

Adds start and stop delineators on outbound and strips them from inbound transmissions

****

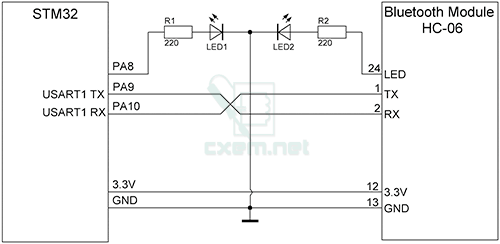
#### Advantages of Bluetooth Module

* When we are traveling with the laptop and with the other wireless devices we don’t need to carry the connection cables.
* Hence it is a wireless technology.
* The cost of the Bluetooth is not an expensive.
* The Bluetooth modules are standardized protocol because they can connect with the same Bluetooth as well as with the [different Bluetooth projects](http://www.efxkits.com/bluetooth-projects) model.
* The Bluetooth module can connect automatically with another Bluetooth device at a range of 30 feet.
* There is a low energy consumption in the Bluetooth devices in generally we can see in the mobile phones.

#### Applications of Bluetooth Module

* Laptops
* Mobile phones
* Wireless technology
* Bluetooth speakers

**INTERFACING BLUETOOTH WITH STM32**

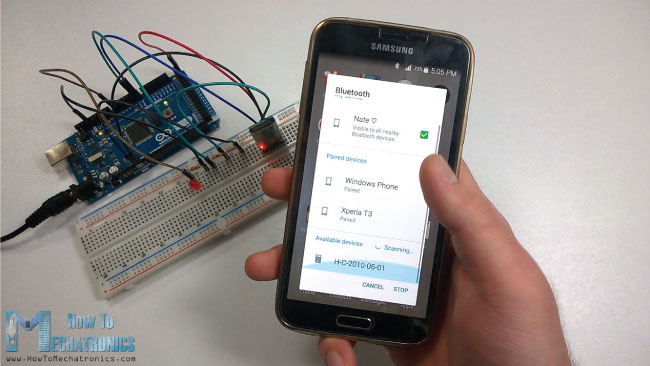
[](http://solderer.tv/wp-content/uploads/2015/07/mc169-1.png)

First we need to define the pin to which our LED will be connected and a variable in which we will store the data coming from the Smartphone. In the setup section we need to define the LED pin as output and set it low right away. As mention previously, we will use the serial communication so we need to begin the serial communication at 38400 baud rate, which is the default baud rate of the Bluetooth module. In the loop section with the Serial.available() function we will check whether there is available data in the serial port to be read. This means that when we will send data to the Bluetooth module this statement will be true so then using the Serial.read() function we will read that data and put it into the “state” variable. So if the Arduino receive the character ‘0’ it will turn the LED off and using the Serial.println() function it will send back to the Smartphone, via the serial port, the String “LED: OFF”. Additionally we will reset the “state” variable to 0 so that the two above lines will be executed only once. Note here that the “state” variable is integer, so when we receive the character ‘0’ that comes from Smartphone, the actual value of the integer “state” variable is 48, which corresponds to character ‘0’, according to the [ASCII](https://en.wikipedia.org/wiki/ASCII) table.. That’s why in the “if” statement we are comparing the “state” variable to a character ‘0’. On the other hand, if the received character is ‘1’, the LED will light up and the String “LED: ON” will be sent back.

Now the code is ready to be uploaded but in order to do that we need to unplug the TX and RX lines because when uploading the Arduino uses the serial communication so the pins RX (digital pin 0) and TX (digital pin1) are busy. We can avoid this step if we use the other TX and RX pins of the Arduino Board, but in that case we will have to use the SoftwareSerial.h library for the serial communication.

## Connecting the Smartphone to the HC-05 Bluetooth Module and the Arduino

Now we are ready to connect the Smartphone to the Bluetooth module and the Arduino. What we need to do here is to activate the Bluetooth and the Smartphone will find the HC-05 Bluetooth module.

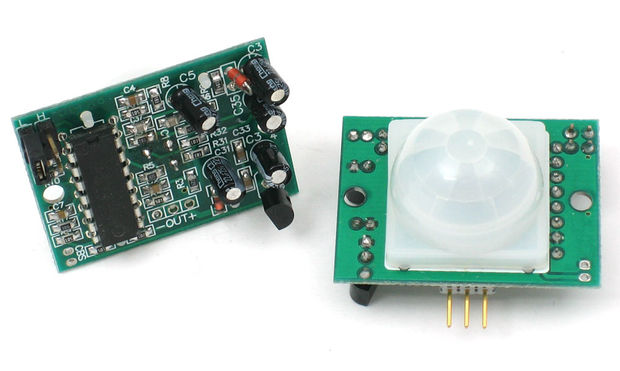


Then we need to pair the devices and the default password of the HC-05 module is 1234. After we have paired the devices we need an application for controlling the Arduino. There are many applications in the Play Store for this purpose which will work with the Arduino code that we wrote. However, I made my own custom application for this tutorial using the MIT App Inventor online application. This is a great and easy to use application for building Android application and in my next tutorial you can find a detailed step by step guide how to build your own custom Android application for your Arduino Project.

Here’s the application that I made. With the connect button we will connect the Smartphone to the Bluetooth module and the status text below the button will tell us whether we have successfully connected. Now using the “Turn ON” and “Turn OFF” buttons we can turn on and off the LED. The text above the buttons is the one that the Arduino is sending back to the Smartphone when a particular button is pressed.

**CHAPTER-3**

**PIR SENSOR**

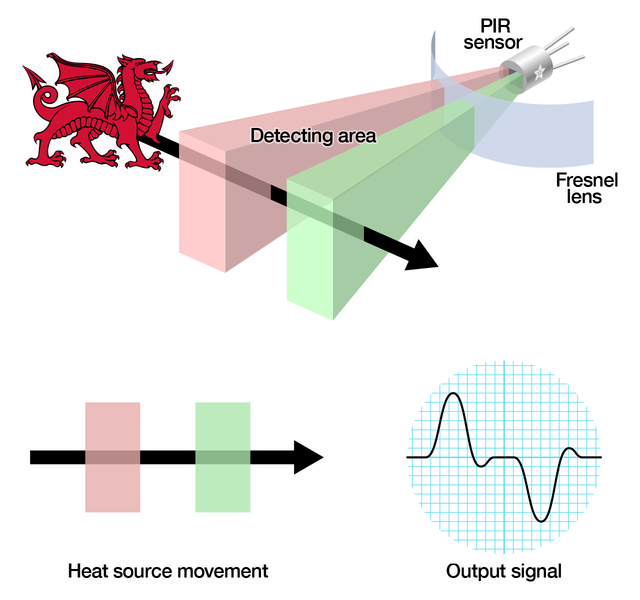
****

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.

All objects with a temperature above total zero Kelvin emit heat energy in the form of radiation. Normally this radiation isn't unmistakable to the human eye since it transmits at infrared wavelengths; however it can be distinguished by electronic gadgets intended for such a reason.

The term passive in this example alludes to the way that PIR gadgets don't produce or emanate any energy for location purposes. They work totally by recognizing the energy radiated by other objects. PIR sensors don't identify or measure "warm"; rather they distinguish the infrared radiation transmitted or reflected from a question.

**WORKING**

An individual PIR sensor recognizes changes in the measure of infrared radiation impinging upon it, which differs relying upon the temperature and surface attributes of the items before the sensor.[2] When a body, for example, a human, goes before the foundation, for example, a divider, the temperature by then in the sensor's field of view will ascend from room temperature to body temperature, and after that back once more. The sensor changes over the subsequent change in the approaching infrared radiation into an adjustment in the yield voltage, and this triggers the discovery. Objects of comparable temperature yet unique surface qualities may likewise have an alternate infrared emanation example, and in this way moving them concerning the foundation may trigger the locator as well.****

**SPECIFICATION AND PIN DIAGRAM**

* **Size:** Rectangular
* **Output:** Digital pulse high (3V) when triggered (motion detected) digital low when idle (no motion detected). Pulse lengths are determined by resistors and capacitors on the PCB and differ from sensor to sensor.
* **Sensitivity range:** up to 20 feet (6 meters) 110 degrees x 70 degrees detection range
* **Power supply:** 3.3V - 5V input voltage

**CHAPTER-3**

**CIRCUIT DIAGRAM**

In this circuit the heap (globule) is associated in arrangement with AC supply and the triac. The door is associated with arduino pin1 for activating. The door is not associated specifically to arduino but rather it is associated through some seclusion circuit (MOC3020). This is on account of there is some reverse of AC current (75 v) through entryway which may devastate the arduino. So to keep this optotriac is being utilized as a detachment. When gating sign is being connected through arduino stick the triac permits to pass the air conditioner current through load until the AC flag turns around its sign.

The activating moment of triac can be dictated by zero intersection location circuit which guarantees that zero intersection of the AC flag. After zero intersection has been recognized the arduino can be customized at which moment it ought to trigger. In the event that the activating heartbeat is connected at stage more prominent than 90 then force of knob is most extreme , generally if the activating heartbeat is connected at under 90 stage then in strained quality of globule will diminish.

### The Opto-Triac

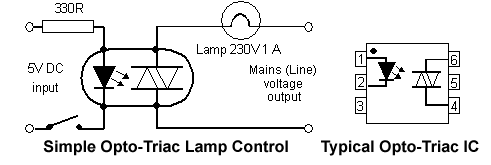


Figure8: Optotriac

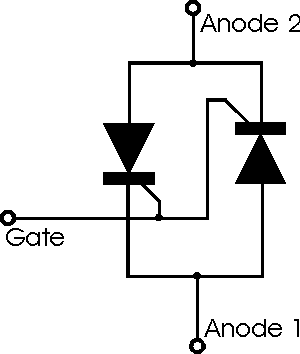
The materials utilized as a part of the fabricate of Triacs and SCRs, similar to any semiconductor gadget, are light touchy. Their conduction is changed by the nearness of light; that is the reason they are ordinarily bundled in little lumps of dark plastic. Be that as it may, if a LED is incorporated inside the bundle, it can turn on the high voltage gadget yield in light of a little information current through the LED. This is the rule utilized as a part of Opto-Triacs and Opto-SCRs, which are promptly accessible in incorporated circuit (IC) frame and don't require exceptionally complex hardware to make them work. Just give a little heartbeat at the perfect time and the power is exchanged on. The fundamental preferred standpoint of these optically initiated gadgets is the incredible protection between the low power and high power circuits, (commonly a few thousand volts). This gives safe disconnection between the low voltage info and high voltage yield.

## TRIAC operation

It can be imagined from the circuit symbol that the TRIAC consists of two thyristors back to back but with a common gate terminal, and the cathode of one thyristor connected to the anode of the other, and vice versa. This configuration is more correctly termed antiparallel

On a basic level, the operation of the TRIAC can be looked on in the format of the antiparallel thyristors, although the actual operation at the semiconductor level is rather complicated.

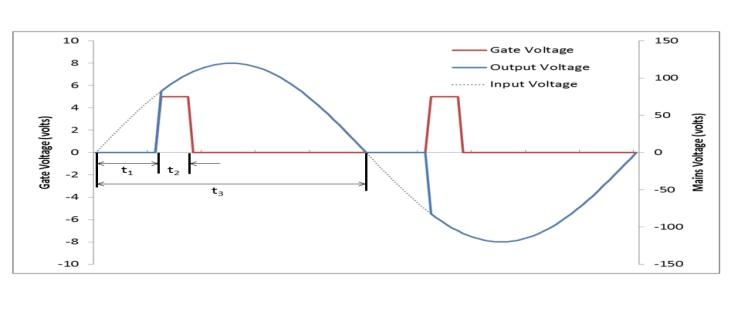
When the voltage on the MT1 is positive with regard to MT2 and a positive gate voltage is applied, one of the SCRs conducts. When the voltage is reversed and a negative voltage is applied to the gate, the other SCR conducts. This is provided that there is sufficient voltage across the device to enable a minimum holding current to flow.

  
**Equivalent circuit of a TRIAC**

In terms of the structure of the device, and its more detailed operation, the main terminals MT1 and MT2 are both connected to p and n regions within the device. The current path depends upon the polarity of the voltage across the main terminals.

As there is considerable scope for confusion, the device polarity is normally described with reference to MT1.

In terms of its operation, the ON characteristics for a TRIAC in any direction are similar to that of a thyristors. However as a result of the physical structure of the TRIAC, the latching current, holding current, and gate trigger current vary according to the different halves of the cycle and which "thyristor" within the TRIAC is being used.



**ADVANTAGES**

* Home appliances like lights and fans can be switched on/off with the help of mobile application directly.
* Voice control facility reduces the work and makes our life easy.
* Very much helpful to the heart patients and old people.
* It is safer than directly switch on/off with the hand.
* Cheaper.

**CONCLUSION AND FUTURE SCOPE**

* ***Conclusion*** – Most reliable and the transmitter and receiver circuit is reduced.
* A single universal system that can be interfaced with all the appliances
* ***Future scope*** – Make everything wireless to make life comfortable.
* Controlling everything with Bluetooth.

**REFERENCES**

* https://www.youtube.com/watch?v=RYLZ3m9ZDyg
* https://www.youtube.com/watch?v=PQ5WzOmqq-k
* http://www.st.com/en/embedded-software/stm32-standard-peripheral-libraries-expansions.html?querycriteria=productId=LN1734
* http://embedded-lab.com/blog/stm32-tutorials/
* https://github.com/istarc/stm32
* http://www.emcu.it/STM32/Atollic/How\_to\_use\_STM32\_examples\_in\_ATOLLIC\_TrueSTUDIO\_Lite/How\_to\_use\_STM32\_examples\_in\_ATOLLIC\_TrueSTUDIO\_Lite.html
* https://stm32f4-discovery.net/2014/09/stm32f4-tutorials-one-place[/](https://stm32f4-discovery.net/2014/09/stm32f4-tutorials-one-place/)