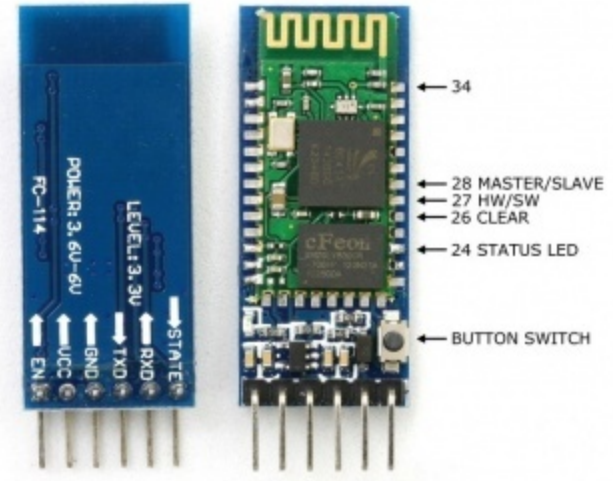


Wireless Serial Communication Using HC-05 Bluetooth Module



Bluetooth Module HC-05

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.



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 other device wirelessly.

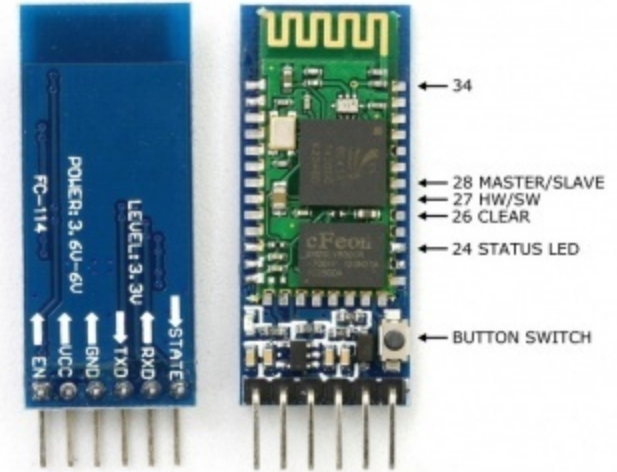


Hardware Features

- 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.
- Permit pairing device to connect as default.
- Auto-pairing PINCODE:"1234" as default.



Pin Description

The HC-05 Bluetooth Module has 6 pins. 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 acts as an UART interface for communication



Pin Description

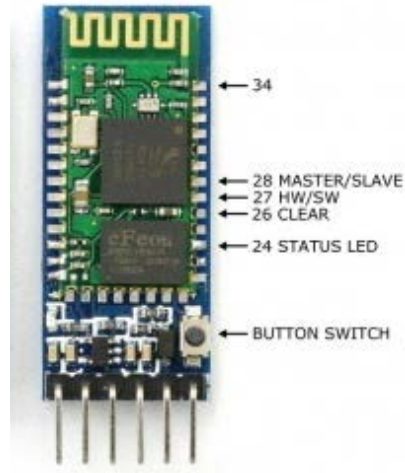
STATE: It acts as a status indicator.

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.

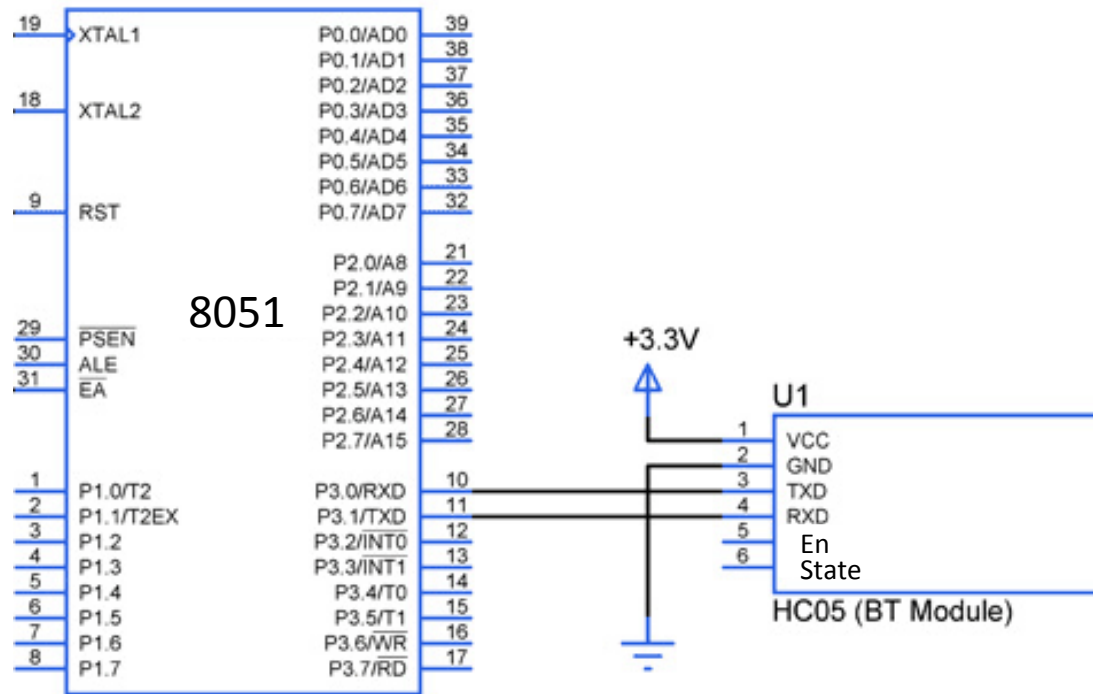
When this module is connected to/paired with any other Bluetooth device, the signal goes High. At this high state, the led blinks with a constant delay say for example 2s delay which indicates that the module is paired.



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.



Hardware Connections

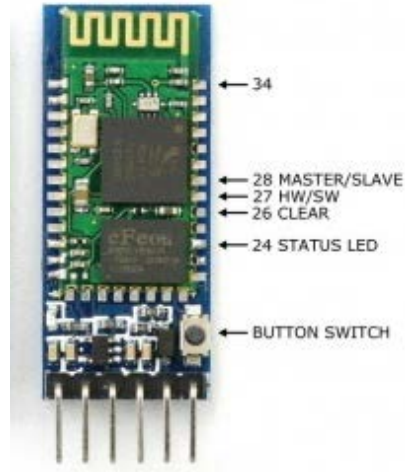


APPLICATIONS:

WIRELESS MOBILE CONTROLLED HOME AUTOMATION

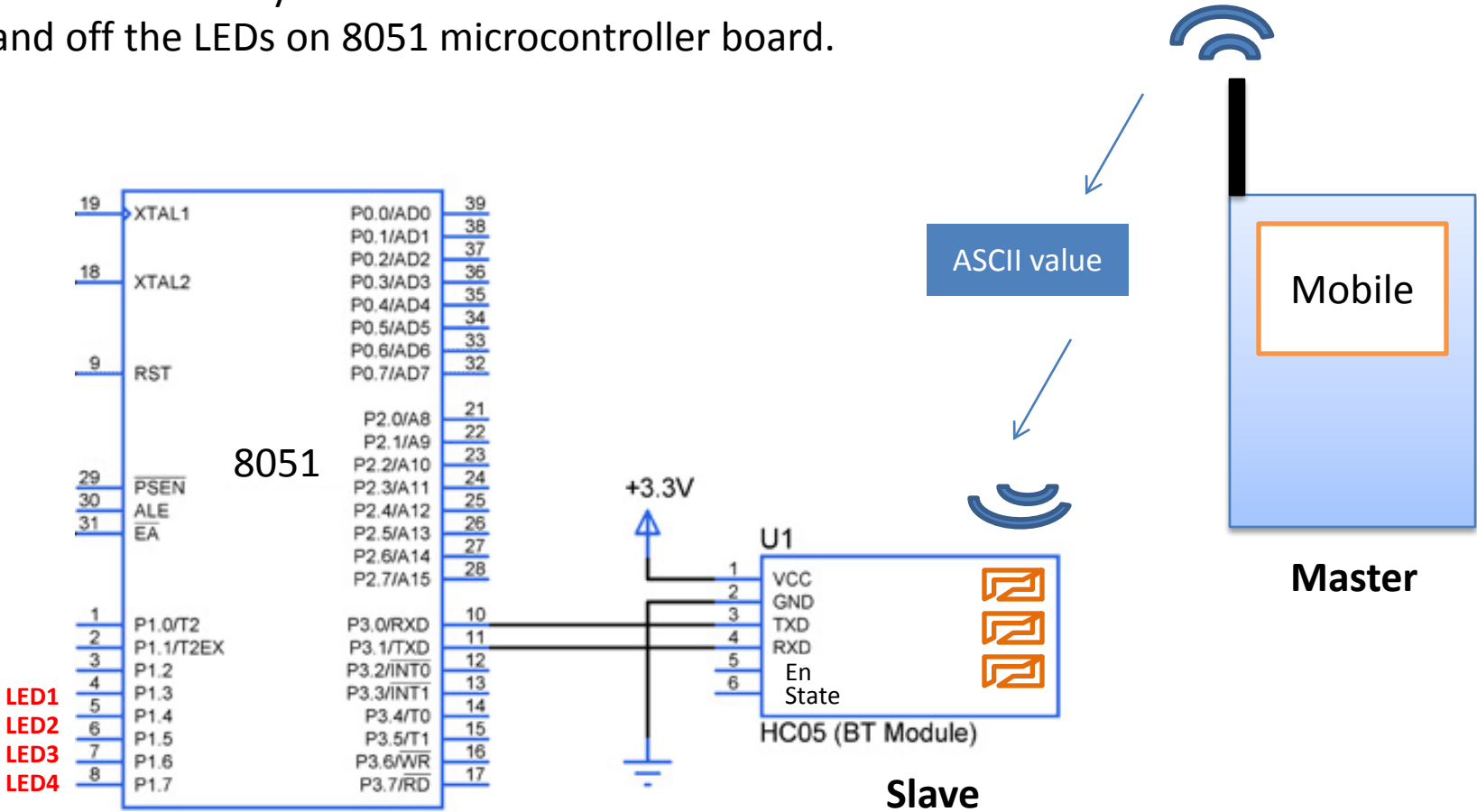
WIRELESS MOBILE CONTROLLED ROBOT

WIRELESS NODES TO RECEIVE DATA TO MOBILE



Application:

Send data wirelessly from mobile to microcontroller to turn on and off the LEDs on 8051 microcontroller board.



```
#include <reg52.h>
```

```
void main()  
{
```

```
    unsigned char temp;
```

```
    TMOD = 0x20;    // timer1 mode2  
    SCON=0x50;    // mode1 r-en  
    TH1=0x0FD;    //9600 baud rate  
    TR1=1;    //on timer1
```

UART initialization

```
    P1=0x00; // port 1 off
```

```
    while(1)  
    {
```

```
        //rx  
        while(!RI); // wait till RI become 1  
        temp=SBUF; // collect received data into variable temp  
        RI=0;    // Clear RI flag
```

Wait till serial data is received

```
        if (temp=='1')  
        {  
            P1=(0X01<<4); // LED1 glow  
        }  
        else if (temp=='2')  
        {  
            P1=(0X01<<5); // LED1 glow  
        }  
        else if (temp=='3')  
        {  
            P1=(0X01<<6); // LED1 glow  
        }  
        else if (temp=='4')  
        {  
            P1=(0X01<<7); // LED1 glow  
        }  
        else if (SBUF=='5')  
        {  
            P1=(0X00); // LED1 off  
        }
```

Logic to generate control action based on received data

```
    }  
}
```