

# GOING WIRELESS WITH THE 8051 MICRO CONTROLLER

SREEJAN DAS

DEPT. OF ECE, CEMK

In case if the article is being read by any person with prior knowledge about micro controllers, he/she must be familiar with the concept of Microcontroller 8051, the first and basic microcontroller(mcu) built by Intel in 1981. Though we are mostly familiar about some of the operations that can be performed by the mcu, most of us never thought of interfacing wirelessly through 8051. We prefer to use Atmega or ESP 32 now a days for the convenient and faster operations.

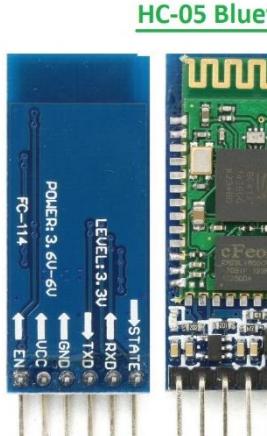
Even if it's limited to some extent, still we can push the powers of an 8051 mcu to interface a **HC-05 Bluetooth module** and go wireless. It is known that the module HC-05 is itself a basic master slave Bluetooth module of version BT 2.0, still it comes handy when used in small projects as it's cost effective.

In the following article we will figure it out how we can interface a HC05 with our humble 8051 microcontroller.

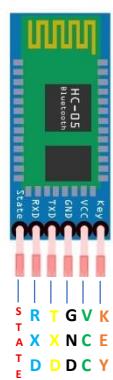
Let's just have a brief overview of HC-05 module,

The **HC-05** is a low-cost Bluetooth module that allows devices to **communicate wirelessly over serial (UART)**\*. It's compatible with microcontrollers like 8051, Arduino, and STM32.

\*UART (Universal Asynchronous Receiver/Transmitter)



HC-05 Bluetooth Module

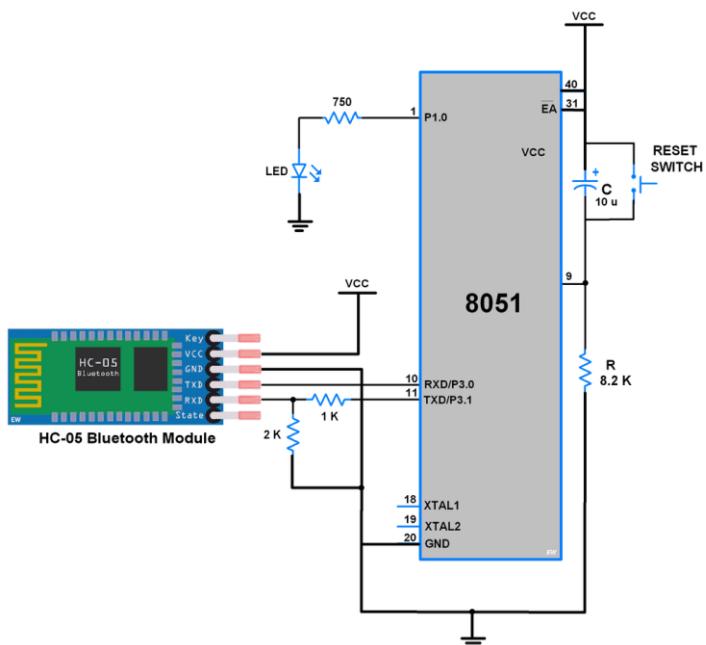


## ◆ Key Specs of HC 05:

- Bluetooth 2.0 (SPP Profile)
- Baud Rate: Default 9600 bps
- Operating Voltage: 3.3V logic (but 5V tolerant VCC)
- Interfaces: TXD (transmitter), RXD (receiver), GND, VCC

8051 microcontrollers have built in UART port (TX = P3.1, RX = P3.0) to send and receive serial data. To communicate with the HC-05, we configure the UART peripheral to **9600 bps, 8-bit data, 1 stop bit, no parity** — matching the HC-05's default settings.

Following is the circuit diagram to interface the module with the 8051 microcontroller, its simple as a cake walk. To verify our interfacing, we will draw a simple example like blinking a led with serial commands 0 or 1 to turn off or on the led as directed by the code we used for interfacing.



Above is the circuit diagram of connecting the HC-05 with the 8051 micro controller.

Now we can verify the job by a simple example as stated earlier:

```
#include <reg51.h>

sbit LED = P2^0; // LED connected to P2.0

void UART_Init() {
    TMOD |= 0x20; // Timer1 Mode2 (8-bit auto-reload)
    TH1 = 0xFD; // 9600 baud rate for 11.0592MHz
    SCON = 0x50; // Mode 1, 8-bit data, 1 stop bit, REN enabled
    TR1 = 1; // Start Timer1
}

char UART_Read() {
    while (RI == 0); // Wait until a byte is received
    RI = 0; // Clear Receive Interrupt flag
    return SBUF; // Return received byte
}

void main() {
    char cmd;
    UART_Init(); // Initialize UART
    LED = 0; // LED initially OFF
    while (1) {
        cmd = UART_Read(); // Read serial input
        if (cmd == '1') {
            LED = 1; // Turn ON LED
        }
        else if (cmd == '0') {
            LED = 0; // Turn OFF LED
        }
    }
}
```

android and connect to the module via Bluetooth. In the serial input you enter 1, the LED blinks and 0 to turn it off, initially the LED remains off.

### ◆ Real-World Applications

- Bluetooth-Controlled Robot
- Smart Light Control via Android
- Secure Door Lock with Password via Bluetooth
- Voice-Controlled Devices (via phone + HC-05)

### ◆ Why This Matters

Understanding HC-05 and UART interfacing lays the foundation for:

- **Wireless projects** without needing Wi-Fi
- Working with GSM, GPS, and other UART-based modules
- Transitioning to IoT-ready MCUs like **ESP32**

### ◆ Conclusion

The interfacing of HC-05 with 8051 is a practical example of embedded systems in action. It combines **hardware-level understanding, UART configuration, and real-world wireless communication**.

As students and makers, mastering this gives you the power to **create smart, interactive systems** using nothing more than a microcontroller and a few lines of code.

Run this code using **Keil uVision** and burn using **Progisp**. The code should verify successfully. Then install an application **Serial Bluetooth Terminal** in