HC-05 is a Bluetooth device used for wireless communication. It works on serial communication (USART).

It is a 6 pin module.

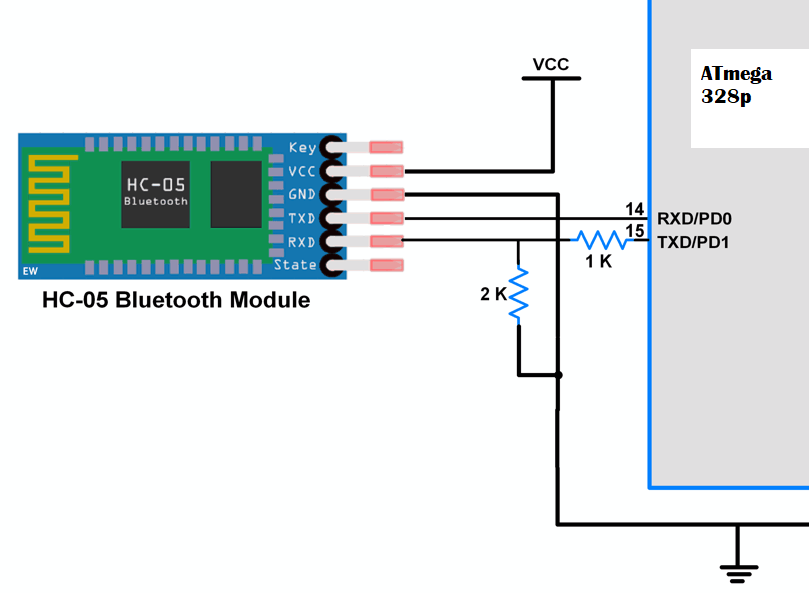
The device can be used in 2 modes; data mode and command mode.

The data mode is used for data transfer between devices whereas command mode is used for changing the settings of the Bluetooth module.

AT commands are required in command mode.

The module works on 5V or 3.3V. It has an onboard 5V to 3.3V regulator.

As the HC-05 Bluetooth module has a 3.3 V level for RX/TX and the microcontroller can detect 3.3 V level, so, no need to shift the transmit level of the HC-05 module. But we need to shift the transmit voltage level from the microcontroller to RX of the HC-05 module.



Test the circuit

LED at PB0

#include <avr/io.h>

#include <util/delay.h>

#define Fosc 16000000

#define USART\_BAUDRATE 9600

#define BAUD\_PRESCALE (((Fosc / (USART\_BAUDRATE \* 16UL))) - 1) //type constant

void usart\_init(){

UCSR0B = (1 << RXEN0) | (1 << TXEN0); /\* Turn on the transmission and reception circuit\*/

UCSR0C = (0 << UPM00) |(0 << UPM01) |(0 << USBS0) |(1 << UCSZ00) | (1 << UCSZ01); // Use 8-bit character sizes

//no parity, 1 stop bit

UBRR0H = ( BAUD\_PRESCALE>>8); /\* Load upper 8-bits of the baud rate value into the high byte of the UBRR register\*/

UBRR0L = BAUD\_PRESCALE; /\* Load lower 8-bits of the baud rate value into the low byte of the UBRR register\*/

}

void usart\_send(char ch)

{

while ((UCSR0A & (1 << UDRE0)) == 0); /\*Do nothing until UDR is ready for more data to be written to it\*/

UDR0 = ch; /\*Echo back the received byte back to the computer\*/

}

char usart\_receive()

{

while ((UCSR0A & (1 << RXC0)) == 0) ; /\* Do nothing until data have been received and is ready to be read from UDR\*/

return (UDR0); /\*Fetch the received byte value into the variable "ByteReceived"\*/

}

void usart\_sendStr(char \* ch){

int i = 0;

for (i = 0; ch[i]!='\0'; i++){

while (( UCSR0A & (1<<UDRE0)) == 0); /\* Do nothing until UDR is ready for more data to be written to it\*/

UDR0 = ch[i];

}

}

int main(void){

DDRB = 0xFF;

char Data\_in;

usart\_init();

while(1)

{

Data\_in = usart\_receive(); /\* receive data from Bluetooth device\*/

if(Data\_in =='1')

{

LED |= (1<<PB0); /\* Turn ON LED \*/

usart\_sendStr("LED\_ON");/\* send status of LED i.e. LED ON \*/

}

else if(Data\_in =='2')

{

LED &= ~(1<<PB0); /\* Turn OFF LED \*/

usart\_sendStr("LED\_OFF"); /\* send status of LED i.e. LED OFF \*/

}

else

usart\_sendStr(“Select proper option"); /\* send message for selecting proper option \*/

}

return 0;

}

With USART library

#include <avr/io.h>

#include "USART\_RS232\_H\_file.h" /\* include USART library \*/

#define LED PORTB /\* connected LED on PORT pin \*/

int main(void)

{

char Data\_in;

DDRB = 0xff; /\* make PORT as output port \*/

USART\_Init(9600); /\* initialize USART with 9600 baud rate \*/

LED = 0;

while(1)

{

Data\_in = USART\_RxChar(); /\* receive data from Bluetooth device\*/

if(Data\_in =='1')

{

LED |= (1<<PB0); /\* Turn ON LED \*/

USART\_SendString("LED\_ON");/\* send status of LED i.e. LED ON \*/

}

else if(Data\_in =='2')

{

LED &= ~(1<<PB0); /\* Turn OFF LED \*/

USART\_SendString("LED\_OFF"); /\* send status of LED i.e. LED OFF \*/

}

else

USART\_SendString("Select proper option"); /\* send message for selecting proper option \*/

}

}