HC012 - Radio Serial de baixo consumo

Trabalho de INF 350 - Prof.: Ricardo dos Santos Ferreira

Grupo:

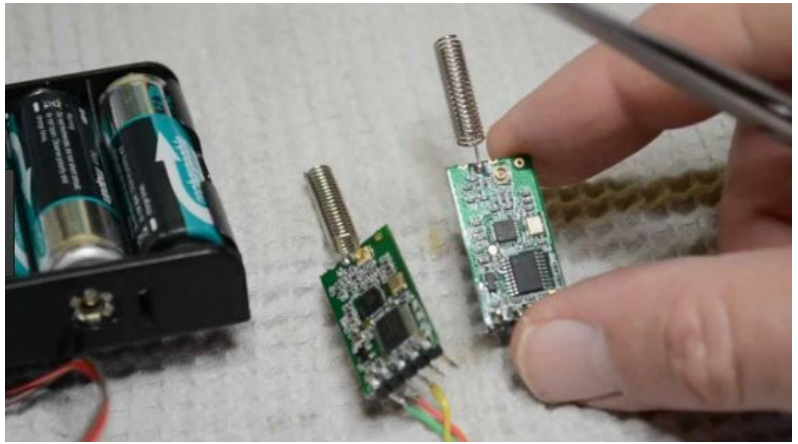
Juliana Moreno - 75763
Michael Canesche - 68064
Vanessa Vasconcelos - 77427

O que é:

- É um módulo que promete comunicação em distâncias de até 1000m.
- Serve para conexão wireless: possui facilidade de uso e se conecta aos mais variados tipos de dispositivos.
- Ideal para aplicações sem fio usando rádio frequência.
- Funciona por interface serial na conexão com o dispositivo (PC, embarcados, etc).
- Usa radiofrequência para comunicação.

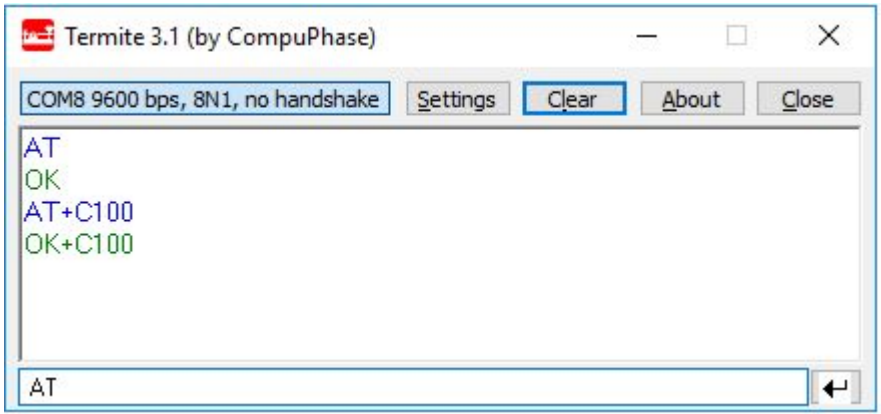
- Existe duas formas de ligar uma antena nesse módulo: pode utilizar a antena espiral que acompanha o módulo, ou utilizar uma antena externa no conector U.FL.

Antena SMA e Mola



Módulo com antena.

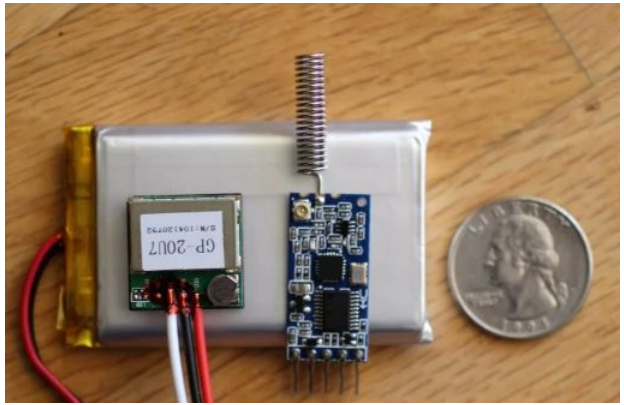
- Há MCU dentro do módulo, o usuário não precisa programar o módulo separadamente.
- Pode ser programado por comandos AT:
  - ➢ altera o baud rate,
  - ➢ parâmetros de inicialização,
  - ➢ velocidade, potência,
  - ➢ frequência de operação,
  - ➢ entre outros.



Alterando a frequência de comunicação do módulo

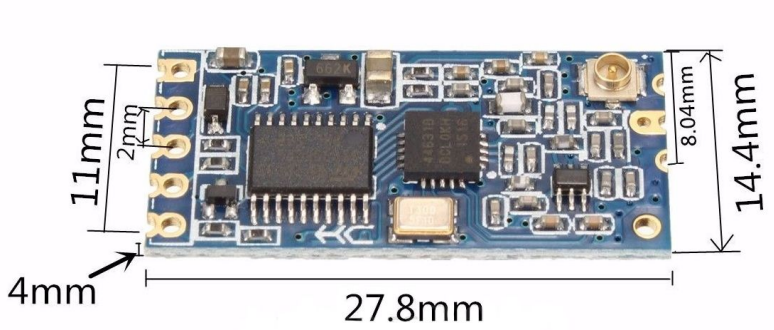
Aplicações

- Sensor sem fio
- Controle de robô sem fio
- Aquisição automática de dados
- Sistema de entrada sem chave do veículo
- Rede sem fios para PC
- Controle remoto industrial e de telemedição
- Automação residencial
- Aquisição sem fio de dados de medidores de gás
- Monitoramento do clima sem fio



Dispositivo de rastreamento muito simples.

Características Físicas



Tensão de alimentação: 3.2V ~ 5.5V

Dimensões (sem antena): 27.8 x 14.4 x 4 mm

Frequência de operação: 433.4 - 473.0MHZ (100 canais de comunicação).

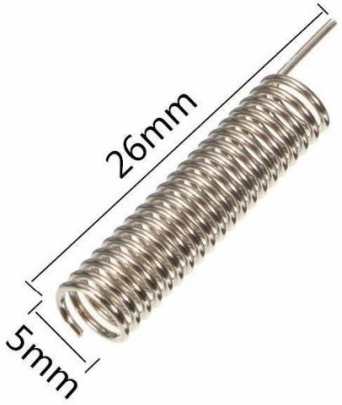
Distância de comunicação padrão: 600m (ambiente aberto) até 1000m (ajustável)

Possui antena (soldado): 5 mm x 26 mm

Anterna externa (conector U.FL);

Potência máxima de transmissão: 100mW

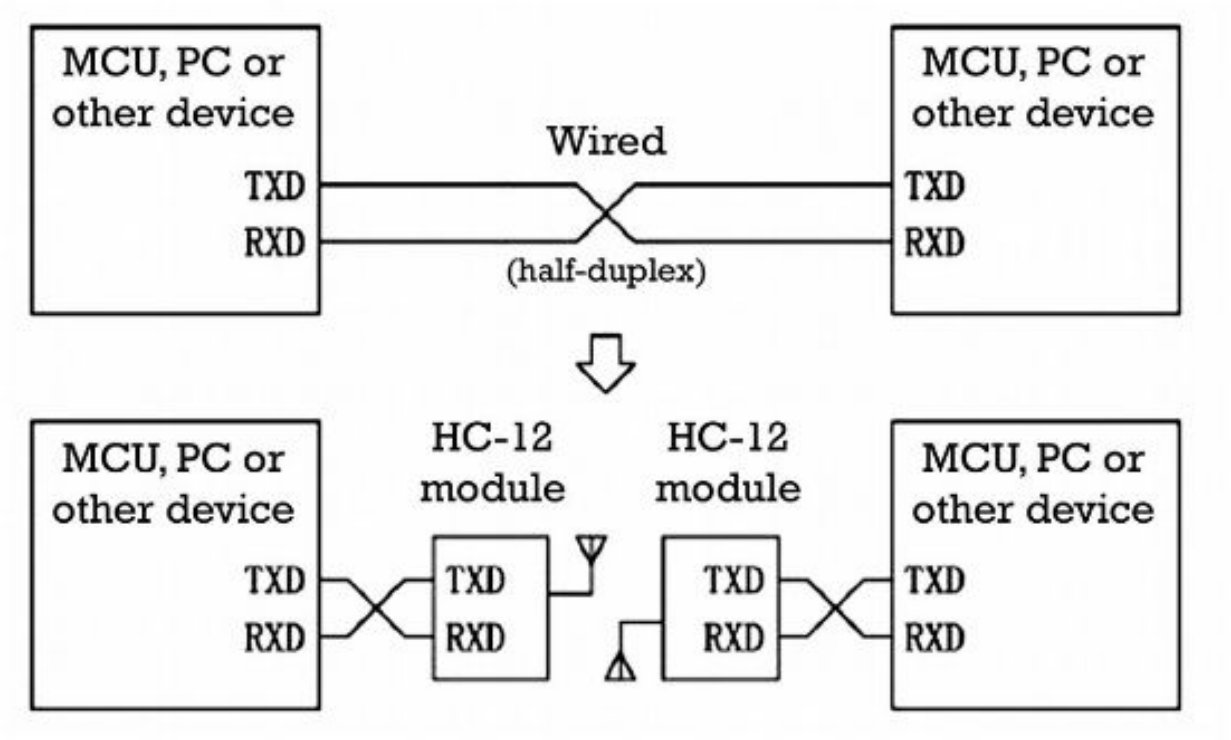
Comunicação Serial



Características Físicas - Módulo

- Pino 1: VCC (3.3V ~ 5V)
- Pino 2: GND
- Pino 3: RX
- Pino 4: TX
- Pino 5: SET (Ativa a configuração quando conectado ao GND)

Princípio Básico do funcionamento



Relação taxa de transmissão da Porta Serial x Ar

Porta Serial (bps)	1200	2400	4800	9600	19,2k	38,4k	57,6k	115,2k
No ar (bps)	5000		15000		58000		236000	

Obs.: O módulo HC-12 automaticamente ajusta a taxa de transmissão (baund rate) no ar de acordo com a taxa de transmissão da Porta Serial.

Sensibilidade do módulo em diferentes *baud rates* (ar)

No ar (bps)	5000	15000	58000	236000
sensibilidade (dBm)	-117	-112	-107	-100

Obs.1: dBm ou dBmW (decibel miliwatt)

Obs.2: Quanto maior o valor (mais próximo de zero), menos vulnerável a interferências externas.

Obs.3: Em geral, a cada vez que a sensibilidade recepção é reduzida em 6dB, a distância de comunicação será reduzida pela metade.

Comparações de cada modo do HC-12

Modo	FU1	FU2	FU3	FU4	Observação
Inativo	3.6mA	80µa	16ma	16ma	valor médio
tempo de delay transmissão	15-25ms	500ms	4-80ms	1000ms	enviar um byte (8 bits)
Teste de Loopback tempo de atraso 1	31ms	-	-	-	Porta serial (9600), 1 byte
Teste de Loopback tempo de atraso 2	31ms	-	-	-	Porta serial (9600), 10 bytes
Alcance da operação a força total (20)	100m	100m	600m (9600bps) 1000m (2400bps)	1800m (1200bps)	condições ideais (sem interferência)

Observações dos modos do HC-12

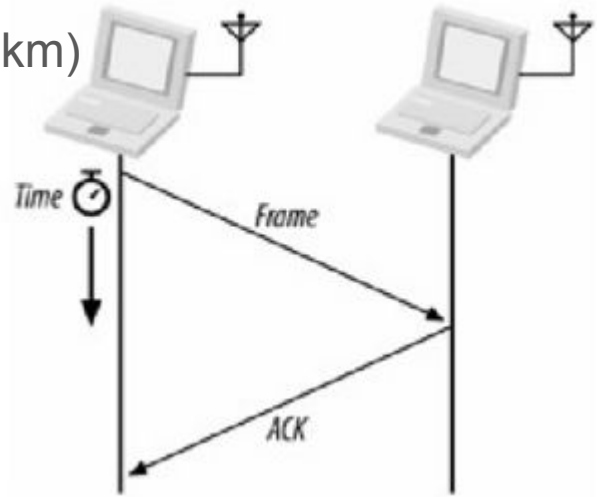
FU1 - Modo moderado de economia de energia.

FU2 - Modo extremo de economia de energia

FU3 - Modo alto de gasto de energia (interessante para evitar interferência externas) (padrão de fábrica)

FU4 - Modo para usos de longos alcances (chegando a 1,8 km)

Teste Loopback: Teste do tempo de ida e volta, desde a transmissão da informação (pino TxD) até o recebimento da confirmação (ack) (pino RxD) no mesmo módulo.



Exemplo - Transmissão Simples[1]

```
#include <SoftwareSerial.h>

const byte HC12RxdPin = 4;                  // Recieve Pin on HC12
const byte HC12TxdPin = 5;                  // Transmit Pin on HC12

SoftwareSerial HC12(HC12TxdPin,HC12RxdPin); // Create Software Serial Port

void setup() {
  Serial.begin(9600);                       // Open serial port to computer
  HC12.begin(9600);                         // Open serial port to HC12
}

void loop() {
  if(HC12.available()){                     // If Arduino's HC12 rx buffer has data
    Serial.write(HC12.read());              // Send the data to the computer
    }
  if(Serial.available()){                   // If Arduino's computer rx buffer has data
    HC12.write(Serial.read());              // Send that data to serial
  }
}
```

[1]https://www.allaboutcircuits.com/projects/understanding-and-implementing-the-hc-12-wireless-transceiver-module/

Exemplo - Envio de Comandos AT[1]

```
#include <SoftwareSerial.h>

const byte HC12RxdPin = 4;                    // "RXD" Pin on HC12
const byte HC12TxdPin = 5;                    // "TXD" Pin on HC12
const byte HC12SetPin = 6;                    // "SET" Pin on HC12

unsigned long timer = millis();               // Delay Timer

char SerialByteIn;                            // Temporary variable
char HC12ByteIn;                              // Temporary variable
String HC12ReadBuffer = "";                   // Read/Write Buffer 1 for HC12
String SerialReadBuffer = "";                 // Read/Write Buffer 2 for Serial
boolean SerialEnd = false;                    // Flag to indicate End of Serial String
boolean HC12End = false;                      // Flag to indiacte End of HC12 String
boolean commandMode = false;                  // Send AT commands

// Software Serial ports Rx and Tx are opposite the HC12 Rx and Tx
// Create Software Serial Port for HC12
SoftwareSerial HC12(HC12TxdPin, HC12RxdPin);

void setup() {

  HC12ReadBuffer.reserve(64);                 // Reserve 64 bytes for Serial message input
  SerialReadBuffer.reserve(64);               // Reserve 64 bytes for HC12 message input

  pinMode(HC12SetPin, OUTPUT);                // Output High for Transparent / Low for Comman
d
  digitalWrite(HC12SetPin, HIGH);             // Enter Transparent mode
  delay(80);                                  // 80 ms delay before operation per datasheet
  Serial.begin(9600);                         // Open serial port to computer
  HC12.begin(9600);                           // Open software serial port to HC12
}

void loop() {

  while (HC12.available()) {                  // While Arduino's HC12 soft serial rx buffer h
as data
    HC12ByteIn = HC12.read();                 // Store each character from rx buffer in byteI
n
    HC12ReadBuffer += char(HC12ByteIn);       // Write each character of byteIn to HC12ReadBu
ffer
    if (HC12ByteIn == '\n') {                 // At the end of the line
      HC12End = true;                         // Set HC12End flag to true
    }
  }
```

```
  while (Serial.available()) {                // If Arduino's computer rx buffer has data
    SerialByteIn = Serial.read();             // Store each character in byteIn
    SerialReadBuffer += char(SerialByteIn);   // Write each character of byteIn to SerialRead
Buffer
    if (SerialByteIn == '\n') {               // Check to see if at the end of the line
      SerialEnd = true;                       // Set SerialEnd flag to indicate end of line
    }
  }

  if (SerialEnd) {                            // Check to see if SerialEnd flag is true

    if (SerialReadBuffer.startsWith("AT")) {  // Has a command been sent from local computer
      HC12.print(SerialReadBuffer);           // Send local command to remote HC12 before cha
nging settings
      delay(100);                             //
      digitalWrite(HC12SetPin, LOW);          // Enter command mode
      delay(100);                             // Allow chip time to enter command mode
      Serial.print(SerialReadBuffer);         // Echo command to serial
      HC12.print(SerialReadBuffer);           // Send command to local HC12
      delay(500);                             // Wait 0.5s for a response
      digitalWrite(HC12SetPin, HIGH);         // Exit command / enter transparent mode
      delay(100);                             // Delay before proceeding
    } else {
      HC12.print(SerialReadBuffer);           // Transmit non-command message
    }
    SerialReadBuffer = "";                    // Clear SerialReadBuffer
    SerialEnd = false;                        // Reset serial end of line flag
  }

  if (HC12End) {                              // If HC12End flag is true
    if (HC12ReadBuffer.startsWith("AT")) {    // Check to see if a command is received from r
emote
      digitalWrite(HC12SetPin, LOW);          // Enter command mode
      delay(100);                             // Delay before sending command
      Serial.print(SerialReadBuffer);         // Echo command to serial.
      HC12.print(HC12ReadBuffer);             // Write command to local HC12
      delay(500);                             // Wait 0.5 s for reply
      digitalWrite(HC12SetPin, HIGH);         // Exit command / enter transparent mode
      delay(100);                             // Delay before proceeding
      HC12.println("Remote Command Executed"); // Acknowledge execution
    } else {
      Serial.print(HC12ReadBuffer);           // Send message to screen
    }
    HC12ReadBuffer = "";                      // Empty buffer
    HC12End = false;                          // Reset flag
  }
}
```

Drop System for Drones!!

https://www.youtube.com/watch?v=vJqRWh3A0cQ

Drop System for Drones - Emissor

```
#include <SoftwareSerial.h>

SoftwareSerial mySerial(2, 3); //RX, TX
const int buttonPin = 8;
int buttonPushCounter = 0;   // counter for the number of button presses
int buttonState = 0;         // current state of the button
int lastButtonState = 0;     // previous state of the button

void setup() {
  pinMode(buttonPin, INPUT);
  pinMode(13, OUTPUT);
  pinMode(12, OUTPUT);
  mySerial.begin(9600);
}

void loop() {

  digitalWrite(13, HIGH);
  int buttonState = digitalRead(buttonPin);//read button state
    // compare the buttonState to its previous state
  if (buttonState != lastButtonState) {
    // if the state has changed, increment the counter
    if (buttonState == HIGH) {
      // if the current state is HIGH then the button
      // wend from off to on:
      buttonPushCounter++;
      Serial.println("on");
      Serial.print("number of button pushes:  ");
      Serial.println(buttonPushCounter);
    } else {
      // if the current state is LOW then the button
      // wend from on to off:
      Serial.println("off");
    }
    // Delay a little bit to avoid bouncing
    delay(50);
  }
  // save the current state as the last state,
  //for next time through the loop
  lastButtonState = buttonState;

  if (buttonPushCounter % 2 == 0) {
    mySerial.println(1111);//send unique code to the receiver to turn on. In this case 1111
    digitalWrite(12, LOW);

} else {
    mySerial.println(0000);
    digitalWrite(12, HIGH);
  }

  delay(20);//delay little for better serial communication
}
```

Drop System for Drones - Receptor

```
#include <SoftwareSerial.h>

#include <Servo.h>

SoftwareSerial mySerial(2, 3); // RX, TX

int servopin = 9;
int pos = 170;
Servo servo1;


void setup() {
  mySerial.begin(9600);
  pinMode(servopin, OUTPUT);
  pinMode(13, OUTPUT);
  servo1.attach(9);
}

void loop() {

  digitalWrite(13, HIGH);
  if(mySerial.available() > 1){
    int input = mySerial.parseInt();//read serial input and convert to integer (-32,768 to 32,767)
    if(input == 1111){//if on code is received
      servo1.write(170);
    }
  if(input == 0000){
      servo1.write(0);
  }
  mySerial.flush();//clear the serial buffer for unwanted inputs

  delay(50);//delay little for better serial communication

}
}
```

http://www.instructables.com/id/Long-Range-Dropping-System-for-Drones-With-Arduino/

Arduino Wing

https://www.youtube.com/watch?v=EjbmKQSqXXE

Outros Exemplos...

Long Range 1.8KM

Acende Leds

GPS Transmission with HC-12

Arduino Remote Beta

HC 12 Uart Transciever

DIY Arduino Car

DCC++ Wireless Throttle

Acessado em 08 de Maio de 2017

Eu quero um...

HC-12 SI4463 microcontrolador de série sem fio, 433 long-range, 1000 M com antena Bluetooth para **HC12**

R$ 11,45 / item

Envio: **R$ 4,35** / item via China Post Registered Air Mail

★★★★★ (41) | Pedidos (65)

ACELEX Jiaqisheng Store

Aliexpress, Acessado em 08 de Maio de 2017

Módulo Transceptor Serial Sem...

R$ 36^{99}

Mercado Livre, Acessado em 08 de Maio de 2017

Obrigado!! Perguntas?