

21022008_NguyenHuuTho

Câu 1:

14:45

4G 57

Cancel

Done s...

Try Again

Unknown device

Connect

Unknown device

Connect

Unknown device

Connect

Unknown device

Connect

[TV] Samsung AU7700...

Connect

Unknown device

Connect

21022008_HuuThoCSE

Connect

Unknown device

Connect

Unknown device

Connect

Unknown device

Connect

Unknown device

Connect

Unknown device

Connect

```

#include <SoftwareSerial.h>
SoftwareSerial HM10(2, 3); // RX, TX
char appData;
String inData = "";

void setup() {
    HM10.begin(9600);
    Serial.begin(9600);
    Serial.println("HM10 serial started at 9600");
    HM10.begin(9600);
    pinMode(13, OUTPUT);
    digitalWrite(13, LOW);
    sendCommand("AT");
    sendCommand("AT+ROLE0");
    sendCommand("AT+UUID0xFFE0");
    sendCommand("AT+CHAR0xFFE1");
    sendCommand("AT+NAME21022008_HuuThoCSE");
}

void sendCommand(const char * command){
    Serial.print("Command send :");
    Serial.println(command);
    HM10.println(command);
    delay(100);

    char reply[100];
    int i = 0;
    while (HM10.available()) {
        reply[i] = HM10.read();
        i += 1;
    }

    reply[i] = '\0';
    Serial.print(reply);
    Serial.println("Reply end");
}

void loop() {
    HM10.listen();
    while (HM10.available() > 0) {
        appData = HM10.read();
        inData = String(appData);
        Serial.write(appData);
    }
}

```

```

if (Serial.available()) {
    delay(10);
    HM10.write(Serial.read());
}

if ( inData == "F") {
    Serial.println("LED OFF");
    digitalWrite(13, LOW);
    delay(500);
}

if ( inData == "O") {
    Serial.println("LED ON");
    digitalWrite(13, HIGH);
    delay(5000);
    digitalWrite(13, LOW);
}
}

```

Câu 2:

```

#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <SoftwareSerial.h>

// Đặt địa chỉ của LCD là 0x27 cho màn hình hiển thị 16 ký tự và 2 dòng
LiquidCrystal_I2C lcd(0x27, 16, 2);
SoftwareSerial HM10(2, 3); // RX, TX

String inData = "";

void setup() {
    lcd.init();                // Khởi tạo LCD
    lcd.backlight();           // Bật đèn nền
    HM10.begin(9600);          // Khởi tạo Bluetooth module với baud rate
9600
    Serial.begin(9600);         // Khởi tạo Serial Monitor với baud rate 9600
    Serial.println("HM10 serial started at 9600");
    pinMode(13, OUTPUT);       // Cài đặt chân 13 làm đầu ra
    digitalWrite(13, LOW);     // Đặt chân 13 ở mức thấp
}

```

```

    sendCommand("AT");                // Gửi lệnh kiểm tra đến module Bluetooth
    sendCommand("AT+ROLE0");          // Đặt module Bluetooth ở chế độ Peripheral
    sendCommand("AT+UUID0xFFE0");    // Đặt UUID dịch vụ
    sendCommand("AT+CHAR0xFFE1");    // Đặt UUID đặc điểm
    sendCommand("AT+NAME21022008_HuuThoCSE"); // Đặt tên Bluetooth module

    lcd.print("BT Setup Done");      // Hiển thị thông báo trên LCD
}

void sendCommand(const char* command) {
    Serial.print("Command send: ");
    Serial.println(command);
    HM10.println(command); // Gửi lệnh đến module HM10
    delay(100);

    char reply[100];
    int i = 0;
    // Đọc phản hồi từ module Bluetooth
    while (HM10.available()) {
        reply[i] = HM10.read();
        i += 1;
    }

    reply[i] = '\0'; // Kết thúc chuỗi
    Serial.print(reply);
    Serial.println(" Reply end");

    lcd.clear();
    lcd.print("Cmd: ");                // Hiển thị lệnh gửi trên LCD
    lcd.print(command);
    lcd.setCursor(0, 1);              // Di chuyển con trỏ xuống dòng thứ hai
    lcd.print("Rply: ");
    lcd.print(reply);                 // Hiển thị phản hồi trên LCD
}

void loop() {
    HM10.listen(); // Lắng nghe dữ liệu từ module Bluetooth
    if (HM10.available() > 0) {
        inData = ""; // Làm rỗng chuỗi inData
        // Đọc tất cả các ký tự khả dụng từ module Bluetooth
        while (HM10.available() > 0) {
            char receivedChar = HM10.read();
            if (isPrintable(receivedChar)) { // Kiểm tra nếu ký tự có thể in
                inData += receivedChar; // Ghép từng ký tự vào chuỗi
            }
        }
    }
}

```

```

    delay(5); // Thêm độ trễ nhẹ để bộ đệm có thời gian điền đầy
}

if (inData.length() > 0) {
    Serial.println(inData); // In chuỗi đầy đủ ra Serial Monitor
    lcd.clear(); // Xóa màn hình LCD
    lcd.print(inData); // Hiển thị chuỗi nhận được trên LCD
}
}

// Kiểm tra nếu có dữ liệu từ Serial Monitor để gửi đến module Bluetooth
if (Serial.available()) {
    while (Serial.available()) { // Khi có dữ liệu từ Serial Monitor
        char c = Serial.read(); // Đọc ký tự
        HM10.write(c); // Gửi ký tự đó qua Bluetooth
    }
}
}

```

Câu 3:

```

#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);

#include <SoftwareSerial.h>
#include <Servo.h>

Servo myServo;
int servoPin = 9;

SoftwareSerial HM10(2, 3); // RX = 2, TX = 3

char appData;
String inData = "";
String command = "";
int angle = 0;
void setup()

{
    lcd.init();
    lcd.backlight();

    myServo.attach(servoPin);
}

```

```

myServo.write(angle);

Serial.begin(9600);
Serial.println("HM10 serial started at 9600");
HM10.begin(9600);
pinMode(13, OUTPUT);
digitalWrite(13, LOW);
}
void loop()
{
  HM10.listen();
  while (HM10.available() > 0) {
    appData = HM10.read();
    if ((int)appData == 32 || (int)appData > 47) {
      inData = inData + String(appData);
    } else {
      Serial.println((int)appData);
      command = command + String((int)appData);
    }
  }
  if (Serial.available()) {
    delay(10);
    HM10.write(Serial.read());
  }
  if (inData != "") {
    Serial.println();
    lcd.clear();
    lcd.print(inData);
    inData = "";
    command = "";
  }
  if (command != "") {
    Serial.println(command);
    if (command == "-11112080") {
      Serial.println("phai");
      if (angle < 180) {
        angle += 5;
        myServo.write(angle);
      }
    }
    if (command == "-11112040") {
      Serial.println("trai");
      if (angle > 0) {
        angle -= 5;
        myServo.write(angle);
      }
    }
  }
}

```

```
    }  
  }  
  if (command.indexOf("-11112000") != -1 || command.indexOf("-103000") != -1) {  
    command = "";  
  }  
}  
delay(10);  
}
```