
ID01 UHF RFID Reader 超高频 RFID 读卡器 说明书



概述

ID01 UHF RFID Reader 是基于超高频 RFID 技术设计的一款非接触式超高频 IC 卡读写模块。模块采用 7V 电压，通过 UART、RS485、USB 与用户主板通信，简单易用。

性能描述

- 工作电流：最大电流 200mA
- 输出功率：最大 24dBm
- 读卡距离：>50cm（有效距离与天线、电子标签及工作环境有关）
- 接口方式：UART、USB、RS485
- 支持波特率：9600、19200、38400、57600、115200（单位 kbps）
- 支持协议：ISO18000-6C（EPC G2）
- 识别卡时间：读每 8 字节小于 10ms，写每字节小于 20ms
- 环境温度：-20℃~+65℃
- 模块尺寸：155mm*100mm

引脚定义

对于 ID01 UHF RFID Reader-RS485 模块

红色线：模块电源输入端 7V/2A

黑色线：模块 GND

绿色线：接 RS485 的 A 端

黄绿色线：接 RS485 的 B 端

对于 ID01 UHF RFID Reader-UART 模块

红色线：模块电源输入端 7V/2A

黑色线：模块 GND

绿色线：接 UART 的 TX 端

黄绿色线：接 UART 的 RX 端

对于 ID01 UHF RFID Reader-USB 模块

直接与电脑 USB 端口相连

数据传输格式

1、通信帧格式介绍

1) 命令帧格式定义

数据流通方向：主机----->读写器

命令帧是主机操作读写器得数据帧，格式如下表所示：

| Packet Type | Length | Command Code | Device Number | Command Data | .. | Command Data | Command Data | Checksum |
|-------------|--------|--------------|---------------|--------------|----|--------------|--------------|----------|
| 0xa0 | n+3 | 1 byte | 1byte | Byte 1 | | Byte n-1 | Byte n | cc |

Packet Type 是包类型域，命令帧包类型固定为 0xa0；

Length 是包长域，表示 Length 域后帧中字节数；

Command Code 是命令码域；

Device Number 是设备号域，当设备号 usercode 为 00 时，表示群发；

Command Data 是命令帧中的参数域；

Checksum 是校验和域，规定校验范围是从包类型域到参数域最后一个字节为所有字节的校验和，读写器接受到命令帧后需要计算校验和来检错。

2) 读写器命令完成相应帧格式定义

数据流通方向：读写器----->主机

读写器命令完成响应帧是一种固定长度的数据帧，其格式如下表所示：

| Packet Type | Length | Command Code | Device Number | Status | Checksum |
|-------------|--------|--------------|---------------|--------|----------|
| 0xe4 | 0x04 | 1 byte | 1 byte | 1 byte | cc |

Packet Type 是包类型域，命令帧包类型固定为 0xe4；

Length 是包长域，表示 Length 域后帧中字节数固定为 0x04；

Command Code 是命令码域；

Device Number 是设备号域，当设备号的 usercode 为 00 时，表示群发；

Status 是状态域，状态域表明读写器完成 PC 机命令后读写器状态或执行命令后的结果，其规定如下表所示；

Checksum 是校验和域，规定校验范围是从包类型域到参数域最后一个字节为所有字节的校验和，读写器接受到命令帧后需要计算校验和来检错。

| 序号 | 值 | 名称 | 描述 |
|----|------|---------------|----------|
| 1 | 0x00 | ERR_NONE | 命令成功完成 |
| 2 | 0x02 | CRC_ERROR | CRC 校验错误 |
| 3 | 0x10 | COMMAND_ERROR | 非法命令 |
| 4 | 0x01 | OTHER_ERROR | 其他错误 |

3) 读写器发送的信息帧格式定义

数据流通方向：读写器----->主机

信息帧是返回给主机的数据帧，比如，用于发送标签给主机，其格式定义如下表所示：

| Packet Type | Length | Response Code | Device Number | Response Data | ... | Response Data | Checksum |
|-------------|--------|---------------|---------------|---------------|-----|---------------|----------|
| 0xe0 | n+3 | 1 byte | 1 byte | Byte 1 | | Byte n | cc |

Packet Type 是包类型域，响应帧包类型固定为 0xe0；

Length 是包长域，表示 Length 域后帧中字节数；

Response Code 是信息码域，取值表示信息的类型；

Device Number 是设备号域，当设备号的 usercode 为 00 时，表示群发；

Response Data 是信息帧中的参数域；

Checksum 是校验和域，规定校验范围是从包类型域到参数域最后一个字节为所有字节的校验和，读写器接受到命令帧后需要计算校验和来检错。

2、通信帧详细介绍

1) EPC 标签识别

主机发：

| 应答 | 数据长度 | 命令 | 设备号 | 校验和 |
|-------|-------|-------|----------|----------|
| Data0 | Data1 | Data2 | Data3 | Data5 |
| A0 | 03 | 82 | Usercode | Chechsum |

主机发送命令：A0038200DB

识别失败回：(E40482) 头，(00) usercode ，(05) Status，(91) Checksum

识别成功回：(E40482) 头，(00) usercode ，(01) 标签 ID (123400000000000000000010)，(37)

Checksum

2) EPC 标签数据读取

主机发:

| 应答 | 读取长度 | 命令 | 设备号 | 内存位置 | 地址 | 读取长度 | 校验和 |
|-------|-------|-------|----------|---------|-------|--------|----------|
| Data0 | Data1 | Data2 | Data3 | Data4 | Data5 | Data6 | Data7 |
| A0 | 06 | 80 | Usercode | MemBank | Addr | Length | Checksum |

主机发送命令: A0 06 80 00 01 02 01 D6;从 0x02 地址开始起, 读取 1 个字节的数据。

MemBank:

| | | |
|----|----------|-----|
| 00 | Reserved | 保留区 |
| 01 | EPC | EPC |
| 10 | TID | TID |
| 11 | User | 用户区 |

读取失败从机回: E40480, (00) usercode, (05) Status, (93) Checksum

读取成功从机回: E00880, (00) usercode, 01020112344E

E0 读取成功数据帧头:

08 数据长度;

80 标签读取命令;

usercode 设备号;

01 Membank 类型;

02 地址;

01 读取长度;

1234 所读取的数据;

4E Checksum。

3) EPC 标签单个字节写入

主机发:

| 应答 | 数据长度 | 命令 | 设备号 | 写入方式 | 内存位置 | 地址 | 写入长度 | 写入数据 | 写入数据 | 校验和 |
|-------|-------|-------|----------|-----------|---------|-------|-------|-------|-------|----------|
| Data0 | Data1 | Data2 | Data3 | Data4 | Data5 | Data6 | Data7 | Data8 | Data9 | Datan |
| A0 | 09 | 81 | usercode | WriteMode | MemBank | Addr | 01 | D1 | D2 | Checksum |

主机发送命令: A0 09 81 00 00 01 02 01 12 34 8C;;

写入失败回: E40480, (00) usercode, (05) Status, (96) Checksum;

写入成功回: E40480, (00) usercode, (00) Status, (9B) Checksum;

Status=00: 写入成都;

Status=其他值: 写入失败;

Addr 说明: EPC 区从 0x02-----0x07 有效。

4) EPC 标签数据多个字写入

| 应答 | 数据长度 | 命令 | 设备号 | 写入方式 | 内存位置 | 地址 | 写入长度 | 写入数据 | 写入数据 | 校验和 |
|-------|-------------------|-------|----------|-----------|---------|-------|--------|-------|---------------|----------|
| Data0 | Data1 | Data2 | Data3 | Data4 | Data5 | Data6 | Data7 | Data8 | Data9 | Datan |
| A0 | 07+ (Length*2) | 81 | Usercode | WriteMode | MemBank | Addr | Length | D1 | D (Length) | Checksum |

主机发送命令: A0 0B 81 00 01 01 02 02 55 55 AA AA D0

写入失败回: E0 04 81 (00) usercode (05) Status (96) Checksum

写入成功回: E0 04 81 (00) usercode (00) Status (9B) Checksum

Status=00: 写入成功

Status=其他值: 写入失败

注: 保留区时, addr>=0, addr+Length<=4, 否则参数错误

注: EPC 区时, addr+Length<=8, 且 ADDR>=2, 否则参数错误

注: TID 区不可写, 只可读

注: 数据区根据实际卡片情况, 每次最大写 8 个字节;

5) 标签锁定

| 应答 | 数据长度 | 命令 | 设备号 | 密码 1 | 密码 2 | 密码 3 | 密码 4 | LOCK 类型 | 校验和 |
|-------|-------|-------|----------|-------|-------|-------|-------|-----------|----------|
| Data0 | Data1 | Data2 | Data3 | Data4 | Data5 | Data6 | Data7 | Data8 | Data9 |
| A0 | 08 | A5 | Usercode | MM1 | MM2 | MM3 | MM4 | LOCK Type | Checksum |

LOCK Type 说明:

00: LOCK USER

01: LOCK TID

02: LOCK EPC

03: LOCK ACCESS

04: LOCK KILL

05: LOCK ALL

其他值: 不锁定

6) 标签解锁

| 应答 | 数据长度 | 命令 | 设备号 | 密码 1 | 密码 2 | 密码 3 | 密码 4 | LOCK 类型 | 校验和 |
|-------|-------|-------|----------|-------|-------|-------|-------|-------------|----------|
| Data0 | Data1 | Data2 | Data3 | Data4 | Data5 | Data6 | Data7 | Data8 | Data9 |
| A0 | 08 | A6 | Usercode | MM1 | MM2 | MM3 | MM4 | UNLOCK Type | Checksum |

UNLOCK Type 说明:

00: UNLOCK USER

01: UNLOCK TID

02: UNLOCK EPC

03: UNLOCK ACCESS

04: UNLOCK KILL

05: UNLOCK ALL

其他值: 不解锁

如访问密码为 12345678, 解锁 EPC 区

发送命令: A0 08 A6 00 12 34 56 78 02 9C

从机回: E4 04 A6 (00) usercode (00) Status (72) Checksum

Status=00: 写入成功:

Status=其他值: 写入失败;

(1) EPC 标签 KILL

| 应答 | 数据长度 | 命令 | 设备号 | RFU | 密码 1 | 密码 2 | 密码 3 | 密码 4 | 校验和 |
|-------|-------|-------|----------|-------|-------|-------|-------|-------|----------|
| Data0 | Data1 | Data2 | Data3 | Data4 | Data5 | Data6 | Data7 | Data8 | Data9 |
| A0 | 08 | 86 | Usercode | 00 | MM1 | MM2 | MM3 | MM4 | Checksum |

主机发送命令：A0 08 86 00 00 12 34 56 78 BE

从机回：E4 04 86 (00) usercode (00) Status (92) Checksum

Status = 00: 写入成功

Status = 其它值: 写入失败

(2) 初始化 EPC 标签 CODE

| 应答 | 数据长度 | 命令 | 设备号 | 校验和 |
|-------|-------|-------|----------|----------|
| Data0 | Data1 | Data2 | Data3 | Data4 |
| A0 | 03 | 99 | Usercode | Checksum |

主机发送命令：A0 03 99 00 C4

从机回：E4 04 99 usercode Status Checksum

Status=00: 写入成功;

Status=其它值: 写入失败;

(3) 读取读头软件版本号 CODE

| 应答 | 数据长度 | 命令 | 设备号 | 校验和 |
|-------|-------|-------|----------|----------|
| Data0 | Data1 | Data2 | Data3 | Data4 |
| A0 | 03 | 6A | Usercode | Checksum |

主机发送命令：A0 03 6A 00 F3

从机回：(E0 05 6A) 头, (00) usercode (05 56) 版本号, (56) Checksum

(4) 复位读写器命令帧 CODE

| Packet Type | Length | Command Code | Device Number | Checksum |
|-------------|--------|--------------|---------------|----------|
| A0 | 03 | 65 | 00 | Checksum |

读写器收到此命令帧后, 先返回命令完成帧, 然后读写器复位。

主机发送命令：A0 03 65 00 F8

从机回：E4 04 65 usercode Status Checksum

Status=00: 成功

Status=其它值: 失败;

(5) 停止读取标签 CODE

| 应答 | 数据长度 | 命令 | 设备号 | 校验和 |
|-------|-------|-------|----------|----------|
| Data0 | Data1 | Data2 | Data3 | Data4 |
| A0 | 03 | A8 | Usercode | Checksum |

主机发送命令 A0 03 A8 00 B5

从机回：E0 04 A8 usercode Status Checksum

Status=00: 成功

Status = 其它值: 失败;

注：EPC 标签的操作是以“字”为单位；ISO18000-6B 标签是以“字节”为单位。

(6) 重新识别标签 CODE (多标签模式下有效)

| 应答 | 数据长度 | 命令 | 设备号 | 校验和 |
|-------|-------|-------|----------|----------|
| Data0 | Data1 | Data2 | Data3 | Data4 |
| A0 | 03 | FC | Usercode | Checksum |

主机发送命令：A0 03 FC 00 61

从机回：E0 04 FC usercode Status Checksum

Status=00：成功

Status=其它值：失败；

(7) 从新获取数据 CODE (多标签模式下有效)

| 应答 | 数据长度 | 命令 | 设备号 | 校验和 |
|-------|-------|-------|----------|----------|
| Data0 | Data1 | Data2 | Data3 | Data4 |
| A0 | 03 | FF | Usercode | Checksum |

主机发送命令：A0 03 FF 00 5E

成功回：E0 04 FF 00 02 1B 00 00 12 34 AA AA 00 00 00 00 55 55 AA AA 01 67 FF 00 00 E2 00 05 11 11 18 02 73 00 00 02 9C 01 CB FF

其中：12 34 AA AA 00 00 00 00 55 55 AA AA、E2 00 05 11 11 18 02 73 00 00 02 9C 为 ID 号。

| 应答 | 数据长度 | 命令 | 设备号 | 写入长度 | 写入数据 | 写入数据 | 写入数据 | 写入数据 | 校验和 |
|-------|----------------|-------|----------|------------|-------|-------|-------|-----------------------|-----------------------|
| Data0 | Data1 | Data2 | Data3 | Data4 | Data5 | Data6 | Data7 | Data (4+2*WordLength) | Data (5+2*WordLength) |
| A0 | 4+2*WordLength | 9C | Usercode | WordLength | D1 | D2 | D3 | D (length*2) | Checksum |

如：在标签 EPC 区地址 4,5 写 2 个字(1234 5678)命令：A0 08 9C 00 02 12 34 56 78 A6

从机返回：E0 04 9C usercode Status Checksum

Status=00：成功

Status=其它值：写入失败；

(9) 获取数据 (多标签模式下有效)

| 应答 | 数据长度 | 命令 | 设备号 | 校验和 |
|-------|-------|-------|----------|----------|
| Data0 | Data1 | Data2 | Data3 | Data4 |
| A0 | 03 | A6 | Usercode | Checksum |

主机发送命令主：A0 03 A6 00 B7

从机回：E0 04 A6 (00) usercode (01)TagCount (71)checksum

TagCoun: 标签数据总数,如果没有,标签数据为:0;随后上传标签数据。

(10) 制定 EPC 号读取 TID 区

| 应答 | 数据长度 | 命令 | 设备号 | EIP ID | | | 校验和 |
|-------|-------|-------|----------|--------|-----|--------|--------|
| Data0 | Data1 | Data2 | Data3 | Data | ... | Data15 | Data16 |
| A0 | 0F | AA | Usercode | 00 | ... | 72 | D7 |

D4...D15 分别是 00 02 25 56 52 65 85 74 12 36 65 72,是指定的 EPC ID 号,共 12 个字节。

主机发送命令：A0 0F AA 00 00 02 25 56 52 65 85 74 12 36 65 72 5B

读取成功回：E0 0C AA 00 00 01 3B F4 00 01 26 74 92 0D

E2 00 34 12 01 36 F4 00 为指定的 EPC 号码的 TID 区,共 8 个字节

读取失败从机返回：E4 04 AA usercode Status Checksum (如 E4 04 AA 00 05 69)

(11) 多个字节写标签

a0 XX AB ReaderAddr memtype startaddr wordlength d0 d1 d2 d3 d4 d5 d6 d7 checksum

| 应答 | 数据长度 | 命令 | 设备号 | 内存位置 | 地址 | 写入长度 | 写入数据 | 写入数据 | 校验和 |
|-------|-----------------------|-------|----------|---------|-------|------------|-------|---------------------|----------|
| Data0 | Data1 | Data2 | Data3 | Data4 | Data5 | Data6 | Data7 | Datan-1 | Datan |
| A0 | 06+ (WordLength*2) | AB | Usercode | MemBank | Addr | WordLength | D1 | D (2*WordLength) | Checksum |

主机发送命令: A0 0E AB 00 03 00 04 11 11 22 22 33 33 44 44 4C

写入失败回: E0 04 AB (00) usercode (05) Status (17) Checksum

写入成功回: E0 04 AB (00) usercode (00) Status (1C) Checksum

Status=00: 写入成功

Status=其它值: 写入失败

注: 保留区时, addr >=0, addr+Length<=4, 否则参数错误

注: EPC 区时 addr+Length <=8, 且 ADDR>=2, 否则参数错误

注: TID 区不可写, 只可读

注: 数据区根据实际卡片情况, 每次最大写 8 个字;

注: 1 个字=两个 BYTE;

(12) 控制 BUZZER

| 应答 | 数据长度 | 命令 | 设备号 | 蜂鸣器控制 | 校验和 |
|-------|-------|-------|----------|------------|----------|
| Data0 | Data1 | Data2 | Data3 | Data4 | Data5 |
| A0 | 04 | B0 | Usercode | BuzzerCtrl | Checksum |

BuzzerCtrl=0: 关闭读卡时的 BEEP 声;

BuzzerCtrl=1: 打开读卡时的 BEEP 声;

BuzzerCtrl>=2: 单独响一次 BEEP 声;

主机发送命令: A0 04 B0 00 00 AC;

从机回: E0 04 B0 (00) usercode 00 68

E4 读写器命令完成响应帧帧头

04 数据长度

B0 控制 BUZZER 命令

00 usercode 设备号

00 状态, 00 为控制成功

68 Checksum

(13) 控制继电器

| 应答 | 数据长度 | 命令 | 设备号 | 继电器控制 | 校验和 |
|-------|-------|-------|----------|------------|----------|
| Data0 | Data1 | Data2 | Data3 | Data4 | Data5 |
| A0 | 04 | B1 | Usercode | RelayOnOff | Checksum |

RelayOnOff =0: 关闭继电器;

RelayOnOff =1: 打开继电器;

主机发送命令: A0 04 B1 00 00 AB;

从机回: E0 04 B1 (00) usercode 00 67

E4 读写器命令完成响应帧帧头

04 数据长度

B1 控制继电器命令
 00 sercode 设备号
 00 状态, 00 为控制成功
 67 Checksum

(14) 设备波特率命令

| 应答 | 数据长度 | 命令 | 设备号 | 波特率参数 | 校验和 |
|-------|-------|-------|----------|------------|----------|
| Data0 | Data1 | Data2 | Data3 | Data4 | Data5 |
| A0 | 04 | A9 | Usercode | SelectBaud | Checksum |

设置波特率命令为 A9

SelectBaud 参数为:

00 9600
 01 19200
 02 38400
 03 57600
 04 115200

发送命令 a0 04 a9 00 04 af 设置到 115200

设置正确应答为: E4 04 A9 00 00 6F

发送命令 a0 04 a9 00 00 b3 设置到 9600 命令

设置正确应答为: E4 04 A9 00 00 6F

注意: 发送波特率设置命令正确后, 此次接收时, 还是当前波特率回应, 但下次就以新的波特率来通讯了;

3、Reader 参数设置通讯协议

1) 停止工作设置

| 应答 | 数据长度 | 命令 | 设备号 | 校验和 |
|-------|-------|-------|----------|-----------------|
| Data0 | Data1 | Data2 | Data3 | Data4 |
| 0xa0 | 0x03 | 0x50 | Usercode | Checksum (0x0e) |

主机发送命令: A0 03 50 00 D

从机回: (E4 04 50)头, (00) usercode (00) Status, (C8) Checksum

Status=00: 成功;

Status=其它值: 失败

2) 同时查询读写器得多个设置参数

| 应答 | 数据长度 | 命令 | 设备号 | 查询参数个数 | 查询参数指定的高位地址 | 查询参数指定的低位地址 | 校验和 |
|-------|-------|-------|----------|--------|-------------------|-------------------|----------|
| Data0 | Data1 | Data2 | Data3 | Data4 | Data5 | Data6 | Data7 |
| 0xA0 | 0x06 | 0x63 | Usercode | Length | Parameter address | Parameter address | Checksum |

如: 主机发送命令: A0 06 63 00 05 00 20 D2 (产品标识查询)

从机回: (E0 0B 63) 头, (00) usercode, 05 00 20, (38 32 32 30 FF)参数值, (C2) Checksum

3) 查询读写器得单个设置参数

| 应答 | 数据长度 | 命令 | 设备号 | 查询参数个数 | 查询参数指定的高位地址 | 查询参数指定的低位地址 | 校验和 |
|-------|-------|-------|-------|--------|-------------|-------------|-------|
| Data0 | Data1 | Data2 | Data3 | Data4 | Data5 | Data6 | Data7 |

| | | | | | | | |
|------|------|------|----------|--------|-------------------|-------------------|----------|
| 0xA0 | 0x05 | 0x61 | Usercode | Length | Parameter address | Parameter address | Checksum |
|------|------|------|----------|--------|-------------------|-------------------|----------|

如：主机发：A0 05 61 00 00 65 95（查询功率）

从机回：(E0 06 61)，(00) usercode, 00 65, (96) 参数值 (BE) Checksum

4) 同时设置多个读写器参数

| 应答 | 数据长度 | 命令 | 设备号 | 设置参数个数 | 查询参数指定的高位地址 | 查询参数指定的低位地址 | Command | | | 校验和 |
|-------|-----------------|-------|----------|--------|-------------------|-------------------|---------|-----|--------|----------|
| Data0 | Data1 | Data2 | Data3 | Data4 | Data5 | Data6 | Data7 | ... | Data E | Data F |
| 0xA0 | 0x06+L ength | 0x62 | Usercode | Length | Parameter address | Parameter address | 01 | ... | 01 | Checksum |

Data 7...Data E 分别是 01 04 10 40 00 01 02 01

如主机发送命令：A0 0E 62 00 08 00 92 01 04 10 40 00 01 02 01 FD（频率设置）

从机回：E4 04 62 (00) usercode (00) Status, (B6) Checksum

Status=00: 成功;

Status=其它值: 失败。

5) 设置单个读写器参数

| 应答 | 数据长度 | 命令 | 设备号 | 查询参数指定高位地址 | 查询参数指定低位地址 | 需要设置的参数值 | 校验和 |
|------|------|------|-----|-------------------|-------------------|-----------------|----------|
| 0xA0 | 6 | 0x60 | 00 | Parameter address | Parameter address | Parameter value | Checksum |

Parameter address (MSB)为参数在 EEPROM 中地址的高字节。

Parameter address (LSB)为参数在 EEPROM 中地址的低字节。

Parameter value 为需要设置的参数值。

读写器接收到此命令帧后，将需要设置的参数写入 EEPROM 中，并返回命令完成帧。

如：主机命令：A0 06 60 00 00 65 96 FF（设置功率）

从机回：(E4 04 60) 头, (00) usercode (00) Status, (B8) Checksum

Status=00: 成功;

Status=其它值: 失败

Arduino 示例程序代码

```
unsigned char StopReadCode[5] = {0xA0, 0x03, 0xA8, 0x00, 0xB5}; //Stop reading the label code
unsigned char ResetCode[5] = {0xA0, 0x03, 0x65, 0x00, 0xF8}; //Reset code
unsigned char StopReadCodeCB[6] = {0xE4, 0x04, 0xA8, 0x00, 0x00, 0x74}; //Stop reading code
success and return the value
unsigned char ResetCodeCB[6] = {0xE4, 0x04, 0x65, 0x00, 0x00, 0xB3}; //Reset code success and
return the value
unsigned char data[6] = {};
void setup()
{
  Serial.begin(9600);
}

void loop()
{
  int i;
  int n=1;
  delay(2000);
  while(n)
  {
    Serial.write(StopReadCode, 5);
    if(Serial.available())
    {
      for(i=0; i<6; i++)
      {
        data[i] = Serial.read();
        delay(1);
      }
      for(i=0; i<6; i++)
      {
        if(data[i] == StopReadCodeCB[i])
          n=0;
        else
          n=1;
      }
    }
    delay(50);
  }
  n=1;
  while(n)
  {
    Serial.write(ResetCode, 5);
    if(Serial.available())
```

```
{  
  for(i=0;i<6;i++)  
  {  
    data[i]=Serial.read();  
    delay(1);  
  }  
  for(i=0;i<6;i++)  
  {  
    if(data[i]==ResetCodeCB[i])  
      n=0;  
    else  
      n=1;  
  }  
}  
delay(50);  
}  
While(1);  
}
```