RF 433MHz transfer for Temperture

RF 433MHz transfer for Temperture application, there are some sensors type for read

1. DS18B20

2. NTC 103AT

3. DHT22

RF433MHz傳輸sensor程序主要邏輯方塊是A/B方式取其一

Ａ. 有定義者SENSOR\_TYPE\_DS18b20

1. 讀取數位式傳感器DS18B20 (單線式資料傳輸) 溫度資訊

2. 所得資訊透過RF433MHz傳輸資料給HOST

3. 傳輸格式：

#define SENSOR\_ID2 0x33340000 //0xFFFE0000 address(15bits) + 0x00010000 sign bit(1bit)+0x0000FFFF data(16bits) //DS18b20溫度數據專用

B. 無定義DS18b20者

1. 讀取數位式傳感器DHT22 (單線式資料傳輸) 溫度/濕度資訊

2. DHT22若無法正確讀取資訊則轉為讀取ＡＤ上的類比資訊１０３ＡＴ

3. 所得資訊透過RF433MHz傳輸資料給HOST

4. 傳輸格式：

#define SENSOR\_ID0 0x33300000 //0xFFFE0000 address(15bits) + 0x0000FFFF data(16bits) for Ain6,Ain7 //ADC數據專用

#define SENSOR\_ID1 0x33320000 //0xFFFE0000 address(15bits) + 0x00010000 sign bit(1bit)+0x0000FFFF data(16bits) //DHT22溫濕度數據專用

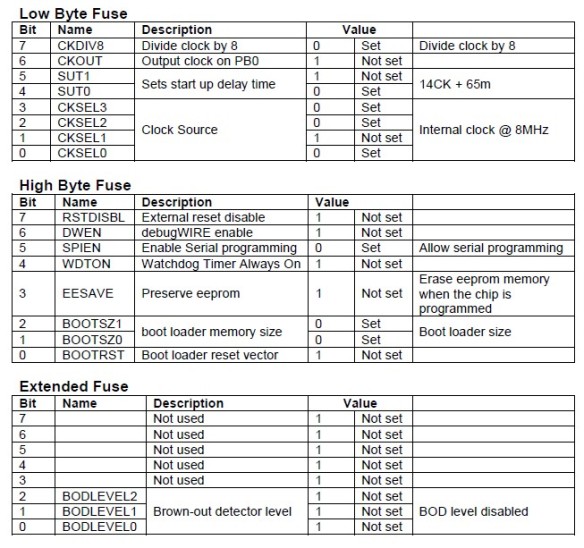
**Arduino ATMEGA328P Default Fuse description**

<http://www.martyncurrey.com/arduino-atmega-328p-fuse-settings/>

### New ATmega 328P Chip.

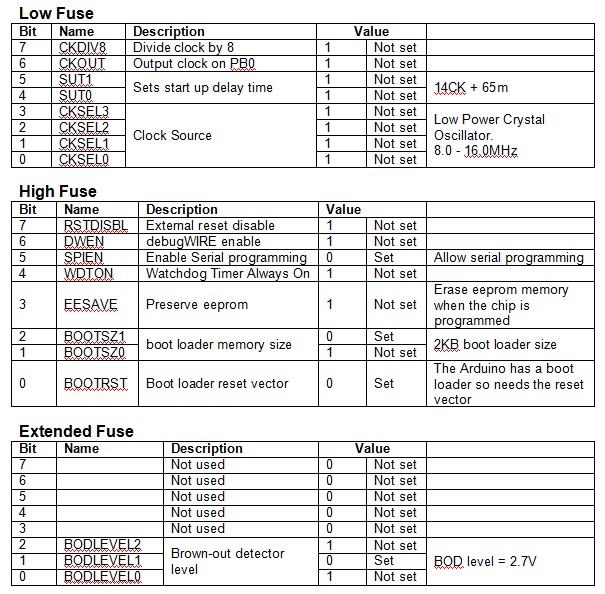
New 328/328P chips generally have the following fuse settings:

Low fuse = 0x62 (B01100010)  
High fuse = 0xD9 (B11011001)  
Extended fuse = 0xFF (B11111111)



### Arduino Duemilanove or Nano w/ ATmega328 Default Fuse Settings

Low fuse = 0xFF (B11111111)  
High fuse = 0xDA (B11011110)  
Extended fuse = 0x05 (B00000101)



除了LB, HB, EB, 第四個byt，還有一個參數是Lock bit常見的應用如下：

​

-U lock:w:0x3F:m //設定 lock byte 為 0x3F (沒保護)

-U lock:w:0x0F:m //設定 lock byte 為 0x0F (bootloader保護)

-U lock:w:0x07:m //設定 lock byte 為 0x07 (bootloader保護，程式可更新，但無法讀取)

詳細的說明如下：

| **Lock Bit** | **Bit No** |
| --- | --- |
| Reserve | 7 |
| Reserve | 6 |
| BLB12 | 5 |
| BLB11 | 4 |
| BLB02 | 3 |
| BLB01 | 2 |
| LB2 | 1 |
| LB1 | 0 |

| **BLB12** | **BLB11** | **SPM寫到bootloader區** | **Applcation區讀取bootloader區資料** |
| --- | --- | --- | --- |
| 1 | 1 | 沒有限制 | 沒有限制 |
| 1 | 0 | 限制 | 沒有限制 |
| 0 | 0 | 限制 | 限制 |
| 0 | 1 | 沒有限制 | 限制 |

| **BLB02** | **BLB01** | **SPM寫到Applcation區** | **bootloader區讀取Applcation區資料** |
| --- | --- | --- | --- |
| 1 | 1 | 沒有限制 | 沒有限制 |
| 1 | 0 | 限制 | 沒有限制 |
| 0 | 0 | 限制 | 限制 |
| 0 | 1 | 沒有限制 | 限制 |

| **LB2** | **LB1** | **Fuse bit,Flash與EEPROM寫入** | **Boot lock,Flash與EEPROM驗證** |
| --- | --- | --- | --- |
| 1 | 1 | 沒有鎖定限制 | 沒有鎖定限制 |
| 1 | 0 | 鎖定限制 | 沒有鎖定限制 |
| 0 | 0 | 鎖定限制 | 鎖定限制 |
| 0 | 1 | 沒有鎖定限制 | 鎖定限制 |

可以依照自己的需求對IC作設定。

使用終端模式是關鍵！

在終端模式下，設置了鎖定位後，我能夠讀取所有存儲單元中的0xFF！我認為這對我來說已經足夠了！

C：\ Users \“ YourName”> avrdude -c arduino -p m328p -P COM3 -b 19200 -t avrdude

：AVR設備已初始化並準備接受指令

。################################################ | 100％0.01s

平均值：設備簽名= 0x1e950f平均值

>轉儲閃存

>>>轉儲閃存

0000 0c 94 5c 00 0c 94 6e 00 0c 94 6e 00 0c 94 6e 00 | 。\。.n。.n。.n。|

0010 0c 94 6e 00 0c 94 6e 00 0c 94 6e 00 0c 94 6e 00 | .n。.n。.n。.n。|

0020 0c 94 6e 00 0c 94 6e 00 0c 94 6e 00 0c 94 6e 00 | .n。.n。.n。.n。|

0030 0c 94 6e 00 0c 94 6e 00 0c 94 6e 00 0c 94 6e 00 | .n。.n。.n。.n。|

avrdude>退出

>>>退出

C：\ Users \“ YourName”>

AVRDUDE：AVR器件初始化並準備接受指示

閱讀| ################################################ | 100％0.01s

avrdude：設備簽名= 0x1e950f avrdude

：讀取輸入文件“ 0x00”

avrdude：寫入鎖定（1個字節）：

寫入| ################################################ | 100％0.03S

AVRDUDE：1字節鎖的書面

AVRDUDE：驗證鎖存儲器針對0×00：

AVRDUDE：負載數據從輸入文件0×00鎖定數據：

AVRDUDE：輸入文件0×00包含1個字節

AVRDUDE：讀片上的鎖數據：

讀| ################################################ | 100％0.01s

avrdude：正在驗證...

avrdude：已驗證1個字節的鎖

avrdude：安全模式：保險絲OK

avrdude完成。謝謝。

C：\ Users \“ YourName”> avrdude -c arduino -p m328p -P COM3 -b 19200 -t avrdude

：AVR設備已初始化並準備接受指令

。################################################ | 100％0.02秒

avrdude：設備簽名= 0x1e950f avrdude

>轉儲Flash

>>>轉儲Flash

0000 ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff | ............. ... |

0010 ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff | ................ |

0020 ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff | ................ |

0030 ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff | ................ |

謝謝大家的幫助！