

Published 下午3:43 by Loser's Lab with 0 comment

## [筆記] C語言中使用單一位元的做法：Bit Field 位元欄位

在嵌入式的設計中，經常有判斷旗標要被使用。  
但使用 `char` 或是 `_Bool` 這些宣告則太過浪費記憶體，  
而C語言中有個方法可以只需要一個位元就可以做。

這個方法稱為 `Bit Field`，現在較新的 C 語言書都不太提到，  
確定有談到這個方法的書目前只看過 [K & R](#) 與 [Pointers on C](#)。

做 `Bit Field` 時必須使用 `struct`

例如：

先寫好

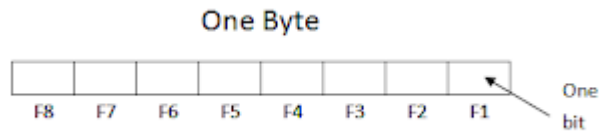
```
struct Flag{  
    unsigned F1 : 1;  
    unsigned F2 : 1;  
    unsigned F3 : 1;  
    unsigned F4 : 1;  
    unsigned F5 : 1;  
    unsigned F6 : 1;  
    unsigned F7 : 1;  
    unsigned F8 : 1;  
};
```

最尾數的1代表使用一個bit。

然後要使用時則宣告

```
struct Flag flagexample;
```

那麼這個叫做 `flagexample` 的變數在記憶體內的實況則會是



如此設定和宣告過後，就可以這樣用：

```
flagexample.F2 = 1;
flagexample.F6 = 0;
```

並且可以應用在判斷是，例如：

```
if ( flagexample.F6 == 1 ) {.....}
```

如果與 `union` 聯合 混合使用。例如：

```
union{
    unsigned char AllMsg;
    struct {
        unsigned bit0:1 ;
        unsigned bit1:1 ;
        unsigned bit2:1 ;
        unsigned bit3:1 ;
        unsigned bit4:1 ;
        unsigned bit5:1 ;
        unsigned bit6:1 ;
        unsigned bit7:1 ;
    };
} Message;
```

這段程式代表直接宣告一個變數叫做 `Message`，程式操作時

```
Message.AllMsg = 0x80;
```

可等同於

```
Message.bit7 = 1; Message.bit6 = 0; Message.bit5 = 0; Message.bit4 = 0;
Message.bit3 = 0; Message.bit2 = 0; Message.bit1 = 0; Message.bit0 = 0;
```

-----

在 `MSP430` 操作的情況下，使用 `Bit Field` 直接操作硬體位元，為了不要再多定義記憶體來搬運所以操作略有點不同。例如：

```
struct SFRbitDef{  
    unsigned B0 : 1;  
    unsigned B1 : 1;  
    unsigned B2 : 1;  
    unsigned B3 : 1;  
    unsigned B4 : 1;  
    unsigned B5 : 1;  
    unsigned B6 : 1;  
    unsigned B7 : 1;  
};
```

先建好結構後，將結構指向實際配合硬體的位址：

```
#define P2input ((struct SFRbitDef *) 0x28);  
#define Ploutput ((struct SFRbitDef *) 0x21);
```

後面的 0x28 和 0x21 分別是搭配硬體用的記憶體位址。

於是存取的時候，以這樣寫成：

```
if ( (P2input->B1) == 0 ) { Ploutput->B4 = 1; }  
// 此段程式是讀取 P2 的 B1接腳，若該接腳電位為零則將 P1 的 B4 接腳輸出為 1。
```

同樣的也可以這樣應用：

```
Ploutput->B4 = ~(Ploutput->B4);  
// 直接簡單的將該位元反向
```

---

 筆記

[← 較新的文章](#)

[首頁](#)

[較舊的文章 →](#)

0 意見:

## 張貼留言

留言身分：

Google 帳戶

發布

預覽

訂閱：[張貼留言 \(Atom\)](#)

---

### SEARCH

Go

---

### POPULAR POSTS

#### One Byte



[筆記] C語言中使用單一位元的做法：Bit Field 位元欄位

在嵌入式的設計中，經常有判斷旗標要被使用。但使用 `char` 或是 `_Bool` 這些宣告則太過浪費記憶體，而C語言中有個方法可以只需要一個位元就可以做。這個方法稱為 `Bit Field`，現在較新的 C 語言書都不太提到，確定有談到這個方法的書目前只看過 `K &...`



[基礎] OpenOCD 與 STM32

在單晶片開發時，大部分都採用燒錄後執行程式，並且透過UART吐出數據的方式來 debug。不過隨著系統的複雜度上升，這樣的開發方式逐漸會遇到瓶頸。也因此，比較大的晶片例如32位元的，通常會內建 JTAG 裝置，也就是一種後門，方便進行 debug。

在此我們採...



[基礎] STM32與開源軟體

這一次的主角是STM32。照片中是一塊STM32開發板，搭配的 debugger 是 ST-Link-V2。STM32具有優異的性能，各種強大的設計使得開發產品非常容易，而且這晶片也不

貴。這次作為實驗對象的是 STM32F103vet6。在前面的章節不斷...



### [實驗] STM8與低通濾波器

進行一些控制操作的時候，需要透過ADC讀取一些外部sensor資訊。有時候會遇到sensor狀態不穩，也因此讀取的數值就會莫名飄動。在這種時候會需要一些簡單的濾波器。關於濾波器的技術，一些前輩都寫過了。例如以8051為基礎的軟體濾波器操作 h...



### [基礎] STM8 的基本系統

為了要能夠利用單晶片做各種事情，必須準備一個基本的系統來調控各種功能。絕大部分的控制，都跟時間有關，所以接下來要講的是一個根據時間運作來操作的系統。並且要把UART的連線做起來，使得PC或者另一端能夠藉此下指令或是讀取晶片內部資料之類的。基本的硬體如...



### [實驗] RS-485 初步探討

本次實驗的主角是RS-485。這是個MAX485晶片所組成的RS-485模組，方便我們使用。RS485是一種優秀的通訊機制，可以進行長距離傳送、穩定可靠，據說波音飛機上就使用這種機制當作控制用的通訊。根據 MAX485 的 datasheet，可以看...



### [基礎] 8051與開源軟體

8051是一款老式的控制晶片，因為優異的設計使得從1981年推出至今仍沒被淘汰，也連帶的產生了許多衍生系列，而Intel目前已經沒再生產。雖說主角是8051，不過現在純正的8051已經不那麼好用，而且純正的大概也很難買到，大多使用相容型的衍伸系列。衍伸系列通常具...



### [基礎] Windows程式與開源軟體

本次的實驗是寫個windows的GUI程式。傳統上認為windows視窗程式只能由Visual C來寫，但其實自由軟體也是能做的。一樣使用CodeBlocks，但必須增加 compiler 稱作 mingw 的，這是個支援windows程式開發的compiler。...

Part	Manufacturer	Part Number	Package	Price (USD)
MCU	TI	MSP430F2202	SOIC8	0.10
Flash	TI	M24C02	SOIC8	0.10
Resistor	Yageo	10K	0603	0.01
Capacitor	Yageo	100nF	0603	0.01
LED	Kingbright	5mm Red	5mm	0.05
Push Button	Omron	B3F	12mm	0.50

### [基礎] MSP430 與開源軟體

這次的實驗晶片對象是msp430。mps430 是 TI 所出的MCU，性能優異而且長年霸佔省電應用的市場。近幾年來價格也大幅度下殺。直接從TI網站擷取的價格(2016年)，大抵上低於一美元 本次實驗的硬體則是 LaunchPad，除了主晶片是...

Part	Manufacturer	Part Number	Package	Price (USD)
MCU	TI	MSP430F2202	SOIC8	0.10
Flash	TI	M24C02	SOIC8	0.10
Resistor	Yageo	10K	0603	0.01
Capacitor	Yageo	100nF	0603	0.01
LED	Kingbright	5mm Red	5mm	0.05
Push Button	Omron	B3F	12mm	0.50

### [實驗] Raspberry Pi 3 之 LED 閃亮

為了要能夠讓樹莓派做點簡單的小事，我們先嘗試驅動GPIO來點亮LED。根據 樹莓派驅動GPIO的網頁 說明，有幾種模式：1. C語言 2. C#語言 3. Ruby 4. Perl 5. Python 6. Scratch 7. Java 8. Sh...

基礎

---

筆記

---

實驗

---

## BLOG ARCHIVE

### ▼ 2016 (12)

► 九月 (3)

► 八月 (2)

▼ 六月 (7)

[筆記] C語言中使用單一位元的做法：Bit Field 位元欄位

[基礎] Mspdebug 與 Msp430

[基礎] MSP430 與開源軟體

[基礎] Windows程式與開源軟體

[基礎] STM8 的基本系統

[基礎] STM8 與開源軟體

[基礎] 8051與開源軟體

---

## 關於我自己

Loser's Lab

因為買不起昂貴的商業開發軟體，所以本私人實驗室都使用開源。

檢視我的完整簡介

Copyright © 2021 魯蛇的嵌入式實驗室

