



User guide for A7125 reference code generator

RC_A7125_14

Document Title

User guide for A7125 reference code generator

Revision History

<u>Rev. No.</u>	<u>History</u>	<u>Issue Date</u>	<u>Remark</u>
0.0	Preliminary	Sep. 24, 2010	
0.2	Modify UI	Oct. 28, 2010	

AMICCOM CONFIDENTIAL

Important Notice:

AMICCOM reserves the right to make changes to its products or to discontinue any integrated circuit product or service without notice. AMICCOM integrated circuit products are not designed, intended, authorized, or warranted to be suitable for use in life-support applications, devices or systems or other critical applications. Use of AMICCOM products in such applications is understood to be fully at the risk of the customer.

1. 簡介

A7125 範例程式產生器（reference code generator）可以讓使用者在修改一些設定後，產生出對應的 RF 範例程式。

這些範例程式只要經過一些修改後，即可以套用在使用者的系統上，讓使用者能夠快速了解並掌握對 A7125 的基本操作。

範例程式產生器所產生的程式需配合單晶片的 C 語言編譯器來編譯，使用者只需要經過一些修改即可以適用於不同的單晶片環境。

2. 程式外觀

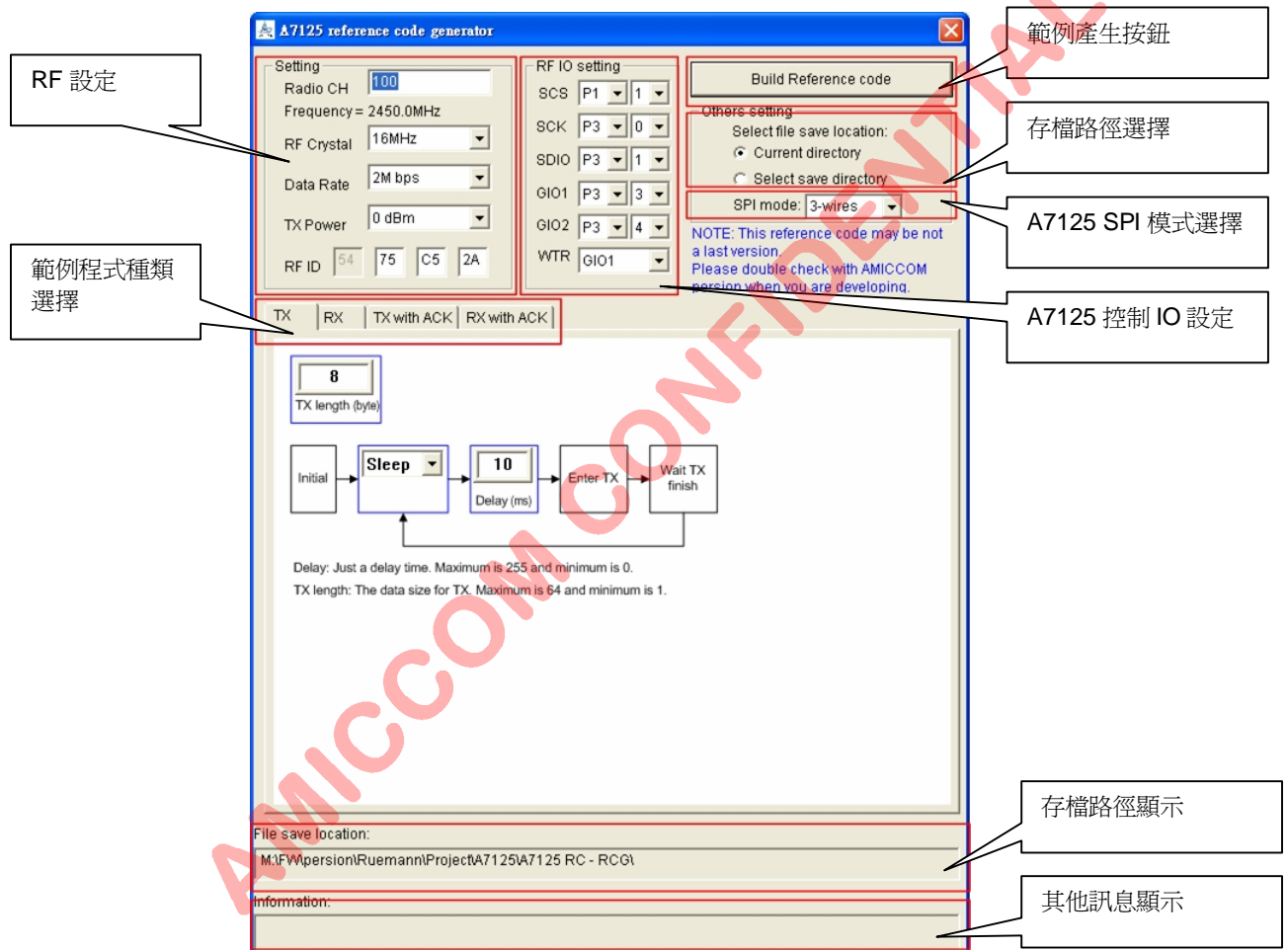


Fig.1 範例程式產生器之執行外觀說明

此範例產生器外觀如 Fig.1 所示，使用者可依需要設定 A7125 的工作頻率、Data rate、Tx power 等設定後，再選擇想要產生的範例格式及要存檔的路徑，按下範例產生鈕後，程式就會產生三個檔案（define.h, A7125reg.h, main.c）以供使用者參考。

3. 選項說明：

3.1 RF 設定：

Radio CH:

- 設定 A7125 的頻道，其範圍為 0~166 的整數，若超過此範圍，程式將會自動修正數值。
- 改變此項設定時，下方會同步顯示此頻道所對應的頻率值。

RF Crystal:

- 設定 A7125 所使用的 Crystal 頻率，目前只有 16MHz 的選項。

Data Rate:

- 設定 FIFO 的 data rate，共有 2M bps, 1M bps 等二個選項可以選擇。

TX Power:

- 設定 A7125 在的發射功率，共有 0dBm, -4dBm, -6 dBm, -8dBm, -12dBm 五個選項可以選擇。

RF ID:

- 設定產生的範例程式中，所使用的 RF ID code。

3.2 RF 控制 IO 設定：

SPI IO setting:

- 根據使用者的環境設定控制 A7125 的 IO，共有 SCS, SCK, SDIO, GIO1, GIO2 這五個 IO 要設定。

WTR setting:

- 選擇 GIO1 或是 GIO2 作為 WTR 輸出。

SPI mode:

- 可設定為 3-wires 或 4-wires 這兩種 SPI 模式，若設為 4-wires 的模式時，依據 WTR setting 選擇另外一個 GIO 當作 SPI 的輸出使用。

3.3 存檔路徑選擇

Current directory:

- 按下此選項時，會將存檔路徑設為範例產生程式所在的路徑。

Select save directory:

- 按下此選項時，會跳出一對話框供使用者選擇想要的存檔路徑。

3.4 範例程式種類選擇

TX:

- 可設定傳輸的封包長度、RF 在閒置時進入 STBY mode 或 SLEEP mode、閒置時間。
- 依據設定將 A7125 設定在 STBY mode 或是 SLEEP mode，等待一個閒置時間後，發射一筆資料，再回到 STBY mode 或 SLEEP mode。

RX:

- 可設定接收的封包長度、RF 在閒置時進入 STBY mode 或 SLEEP mode、閒置時間。
- 依據設定將 A7125 設定在 STBY mode 或是 SLEEP mode，等待一個閒置時間後，進入 RX mode 以接收資料，在接收到資料後，會去檢查 CRC，若 CRC 沒有錯誤，則把資料讀出來，之後再進入 STBY mode 或 SLEEP mode。

TX with ACK:

- 可設定傳輸的封包長度、RF 在閒置時進入 STBY mode 或 SLEEP mode、閒置時間、ACK 封包長度、接收 ACK 時的最大等待時間。
- 依據設定將 A7125 設定在 STBY mode 或是 SLEEP mode，等待一個延遲時間，發射一筆資料後，進入 RX mode 以接收 ACK，若在設定的等待時間內沒有收到 ACK 封包就進入 STBY mode 或 SLEEP mode，若有收到 ACK 封包則檢查 CRC，若 CRC 沒有錯誤，就把 ACK 封包資料讀出，之後再進入 STBY mode 或 SLEEP mode。

RX with ACK:

- 可設定接收的封包長度、RF 在閒置時進入 STBY mode 或 SLEEP mode、閒置時間、ACK 封包長度。
- 依據設定將 A7125 設定在 STBY mode 或是 SLEEP mode，等待一個閒置時間後，進入 RX mode 以接收資料，在接收到資料後，會去檢查 CRC，若 CRC 沒有錯誤，則把資料讀出來並發射一個 ACK 封包，之後再進入 STBY mode 或 SLEEP mode。

4. 範例程式檔案：

使用者按下【範例產生按鈕】後，程式會在所設定的存檔路徑下產生三個檔案：define.h、A7125Reg.h、main.c，其中define.h 是用來定義變數型態，A7125reg.h 是用來定義 A7125 的暫存器位置和 Strobe 命令，main.c 程式主體。下面將說明一些可能需要修改的地方。

4.1 Define.h：

使用者可能需要修改這些常用的變數型態以符合使用者之單晶片編譯環境。

```
#define Sint8   signed   char
#define Uint8   unsigned char
#define Uint16  unsigned int
#define Uint32  unsigned long
```

4.2 公用常式：

範例程式有一個約 1ms 及 100us 的延遲常式如下，使用者需自行修改此延遲常式以符合使用者之系統需求。

```
//!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!//
//                                DELAY 1MS FUNCTION                                //
//                                USER NEED CHANGE THIS FUNCTION TO FIT SYSTEM.      //
//!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!//
void Delay1ms(Uint8 n)
{
    Uint8 i, j;
    while(n--)
    {
        for(j = 0; j < 6; j++)
        {
            for(i = 0; i < 235; i++)
            {
                _nop();
                _nop();
                _nop();
                _nop();
            }
        }
    }
}
```

```
//!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!//
//                                DELAY 100US FUNCTION                                //
//                                USER NEED CHANGE THIS FUNCTION TO FIT SYSTEM.      //
//!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!//
void Delay100us_WTR(Uint8 n)
{
    Uint8 i;
    while(n--)
    {
        for(i = 0; i < 100; i++)
        {
            if ( RF_WTR == 0 )
                return;
            _nop();
            _nop();
            _nop();
        }
    }
}
```