

サンプルプログラムの概要

1. 概要

本アプリケーションノートでは、ミニマイコン評価カード CT-207 と LCD 評価カード CT-201 を接続して、LCD 評価カードを制御する C 言語ライブラリを使用したプログラムについて説明します。
動作環境は以下のとおりとします。

| | |
|------------|--|
| ターゲットマイコン： | 78K0S/KA1P CX |
| ターゲットボード： | CT-207 + CT-201 |
| 開発環境： | NEC エレクトロニクス社製 統合開発環境 PM plus V5.21 C コンパイラ CC78K0 W3.70、アセンブラ RA78K0S W1.40 |

2. サンプルプログラムの構成

以下にサンプルプログラムのファイル構成を示します。以下に示すファイルは全て、「CT207_CT201Sample」フォルダに収められています。

■ CT207_CT201Sample.prw

統合開発環境 PM plus のワークスペースファイルです。このファイルをダブルクリックすると PM plus が起動し、CT207_CT201Sample のワークスペース、プロジェクトが開いた状態になります

■ CT207_CT201Sample.c

サンプルプログラムの C 言語ソースファイルです。

■ CT-207Lib.c

CT-201 と接続した場合の LCD モジュール、押ボタンスイッチ制御、シリアル通信の制御をまとめた C 言語のライブラリ・ソースです。

■ CT-207.h

CT-207Lib.c のソースファイルの各関数を使用する際に必要なヘッダファイルです。
CT_207Lib.c、CT207_CT201Sample.c からインクルードされています。

■ MakeLib.bat ※

ライブラリ作成用のバッチファイルです。

■ CT207Lib.sub ※

MakeLib.bat でライブラリを作成する際によりびだす、コマンドファイルです。

■ CT-207.lib ※

MakeLib.Bat によって作成されたライブラリです。

※ 本アプリケーションノートでは説明していません。

3. サンプルプログラムの内容

■ プログラムの動作

リセット解除後に、初期設定関数を実行すると、シリアル通信で文字列を送信し、CT-201 の LCD モジュールに文字列を表示します。

```
*****
Sunhayato
78K MiniEvaluation Card
Sample Application
*****
```

ターミナル表示

```
サンハヤト
78K サンプルアプリ
```

LCD モジュール

上記処理のあと、約 1 秒間ウエイトをしてから、シリアル通信で「OK」を送信します。

その後 CT-207、CT-201 の押ボタンスイッチの入力待ち、シリアル通信の受信待ちになります。

LCD モジュールの一段目はカーソル点滅となりキャラクタの受信があった場合に表示します。

LCD モジュールの二段目には左から SW1、SW2、SW3、SW4 の状態（押されている = ON、押されていない = OFF）を示します。SW1、SW2 を押すと、LED1、LED2 が点滅します。

また、各スイッチが押されると対応するキャラクタを送信します。

```
■
OFF OFF OFF OFF
```

LCD モジュール

| 押ボタンスイッチ | 送信記号 | 表示 |
|------------------|---------|---------------------|
| CT-201 の SW1 | LCD SW1 | CT-201 の LCD に状態表示 |
| CT-201 の SW2 | LCD SW2 | CT-201 の LCD に状態表示 |
| CT-201 の SW3 | LCD SW3 | CT-201 の LCD に状態表示 |
| CT-201 の SW4/SW5 | LCD SW4 | CT-201 の LCD に状態表示 |
| CT-207 の SW1 | SW1 | CT-207 の LED1 が速い点滅 |
| CT-207 の SW2 | SW2 | CT-207 の LED2 が速い点滅 |

■ シリアル通信のフォーマット

シリアル通信のフォーマットは以下のとおりです。

| | |
|---------|---------|
| 通信速度 | 9600bps |
| データ長 | 8 ビット |
| ストップビット | 1 ビット |
| パリティ | なし |

■ 使用する周辺機能

サンプルプログラムでは、以下のマイコン周辺機能を使用しています。

| 周辺機能 | I/O | 備考 |
|------------|----------------------------|---|
| ポート P20 | CT-201 の SW1、LCD (DB4) | 押ボタンスイッチを入力する時は入力ポートに設定 LCD に出力する時は出力ポートに設定 |
| P21 | CT-201 の SW2、LCD (DB5) | |
| P22 | CT-201 の SW3、LCD (DB6) | |
| P23 | CT-201 の SW4/SW5、LCD (DB7) | |
| P42,45,123 | CT-201 の LCD (E、RS、R/W) | 出力ポートに設定 |
| P40,41 | CT-207 の SW1、SW2 | 入力ポートに設定 |
| P130,123 | CT-207 の LED1、LED2 | 出力ポートに設定 |
| UART6 送受信 | CT-207 の CN1 | 送受信割り込み |
| タイマ 80 | - | 10msec の割り込み |

4. ライブラリ関数 (CT207Lib.c) の内容

以下に各ライブラリ関数の内容を示します。

■ MCU_Init()

<概要>

| | |
|-----|----------|
| 名前 | MCU_Init |
| 機能 | MCU の初期化 |
| 引数 | なし |
| 戻り値 | なし |

<実行結果>

マイコンが以下のとおり初期化されます。

| | |
|-------------------------------|-----------------------|
| CPU 動作クロック | 分周無し (外部 10MHz) |
| ウォッチドッグ・タイマ | 停止 |
| ポートレジスタ 2,3,12,13 | 0000 0000 |
| ポートレジスタ 4 | 0001 1001 |
| プルアップ抵抗オプション レジスタ PU3,4,12 | 0000 0000 |
| PU2 | 0000 1111 (1= PullUp) |

| | |
|----------------|----------------------------|
| ポートモードレジスタ 2,3 | 1111 1111 (1= IN, 0 = OUT) |
| ポートモードレジスタ 4 | 1101 0010 (1= IN, 0 = OUT) |
| ポートモードレジスタ 12 | 1111 0111 (1= IN, 0 = OUT) |

■ LCD_Init()

<概要>

| | |
|-----|-----------------------------|
| 名前 | LCD_Init |
| 機能 | CT-201 上の LCD(SC1602P) の初期化 |
| 引数 | なし |
| 戻り値 | なし |

<注意事項>

他の LCD 操作関数(LCD_XXX())を呼び出す前にこの関数を実行し、LCD を初期化しておく必要があります。また、LCD が接続されているマイコンのポートは、あらかじめ出力に設定されている必要があります。

<実行結果>

LCD が以下のとおり初期化されます。

| | |
|----------|------------------|
| インターフェイス | 4bit |
| 表示行数 | 2行 |
| フォント | 5 × 8 ドット |
| 表示 | ON、カーソルあり、カーソル点滅 |
| 画面全体シフト | なし |
| カーソル方向 | インクリメント |

■ LCD_Clear()

<概要>

| | |
|-----|-------------------------------|
| 名前 | LCD_Clear |
| 機能 | CT-201 上の LCD(SC1602P) の表示を消去 |
| 引数 | なし |
| 戻り値 | なし |

<実行結果>

LCD の表示が消去されます。

■ LCD_Puts(const char *s)

<概要>

| | |
|-----|-------------------------|
| 名前 | LCD_Puts |
| 機能 | LCD に文字列を表示 |
| 引数 | const char *s : 表示する文字列 |
| 戻り値 | なし |

<実行結果>

LCD に引数 s の文字列が表示されます。カーソルは、s の文字数だけインクリメント（右に移動）されます。

■ LCD_Putc(char c)

<概要>

| | |
|-----|-----------------|
| 名前 | LCD_Putc |
| 機能 | LCD に 1 文字表示 |
| 引数 | char c : 表示する文字 |
| 戻り値 | なし |

<実行結果>

LCD に引数 c の文字を表示します。カーソルは、1 文字分インクリメント（右に移動）されます。

■ LCD_Goto(unsigned char pos)

<概要>

| | |
|-----|---------------------------------|
| 名前 | LCD_Goto |
| 機能 | LCD のカーソル移動 |
| 引数 | unsigned char pos : カーソル移動先アドレス |
| 戻り値 | なし |

<実行結果>

カーソルが引数 pos で指定されたアドレスに移動されます。

<使用例>

カーソルを 2 行目先頭から 3 文字目に移動します。定数 LCD_LINE2START は、2 行目の先頭アドレスを表し、CT207Lib.h 内で定義されています。

```
#include "CT207Lib.h"
中略
LCD_Init();
中略
LCD_Goto(LCD_LINE2START + 2);
```

■ Delay10Us()

<概要>

| | |
|-----|---------------|
| 名前 | Delay10Us |
| 機能 | 10 μ 秒の遅延 |
| 引数 | なし |
| 戻り値 | なし |

<注意事項>

NOP 命令を実行することにより、10 μ 秒の遅延を作り出しています。クロック周波数 8MHz, NOP 命令実行に 2 クロック必要であるとし、NOP 命令を 40 回実行します。

クロック周波数、CPU 変更時には修正が必要です。

<実行結果>

10 μ 秒経過後、関数からリターンします。

■ DelayMs

<概要>

| | |
|-----|------------------------------------|
| 名前 | DelayMs |
| 機能 | m 秒単位の遅延 |
| 引数 | unsigned int ms : 遅延時間は、ms[m 秒]を指定 |
| 戻り値 | なし |

<実行結果>

引数 ms で指定した m 秒経過後、関数からリターンします。

■ UART6_Init()

<概要>

| | |
|-----|-------------------------|
| 名前 | UART6_Init |
| 機能 | シリアル・インタフェース UART6 の初期化 |
| 引数 | なし |
| 戻り値 | なし |

<注意事項>

シリアル通信で使用する TxD6, RxD6 端子は、それぞれ出力、入力に設定され、ポート P43/TxD6 に対応するレジスタは 1 に設定されている必要があります。CPU クロック周波数変更時には、ボーレートレジスタ BRGC6 の設定値を見直す必要があります。

<実行結果>

UART6 が以下の通り設定され、UART6 送受信割り込み許可状態となります。

| | |
|---------|-------------|
| ボーレート | 9600bps 非同期 |
| パリティ | なし |
| キャラクタ | 8 ビット |
| ストップビット | 1 ビット |

■ UART6_Putc(char c)

<概要>

| | |
|-----|--------------------|
| 名前 | UART6_Putc |
| 機能 | UART6 から1キャラクタ送信 |
| 引数 | char c : 送信するキャラクタ |
| 戻り値 | なし |

<実行結果>

UART6 による文字列送信が完了して送信バッファが空になるまで待ち、送信バッファに引数cのキャラクタを設定後、直ちにこの関数からリターンします。引数cのキャラクタはUART6により送信されます。

■ UART6_Puts(const char *s)

<概要>

| | |
|-----|-------------------------|
| 名前 | UART6_Puts |
| 機能 | UART6 から文字列を送信 |
| 引数 | const char *s : 送信する文字列 |
| 戻り値 | なし |

<実行結果>

UART6 による文字列送信が完了して送信バッファが空になるまで待ち、文字列の送信を開始して直ちにこの関数からリターンします。UART6 送信完了割り込みにより、順次文字列に含まれるキャラクタが送信されます。

■ UART6_Getc(char *c)

<概要>

| | |
|-----|--------------------------------|
| 名前 | UART6_Getc |
| 機能 | UART6 による受信キャラクタの取得 |
| 引数 | char *c : 受信キャラクタへのポインタ |
| 戻り値 | char : 受信キャラクタがあるとき1, それ以外のとき0 |

<実行結果>

UART6 の受信バッファに受信キャラクタがあるとき、引数*cに受信キャラクタをコピーし、戻り値1でリターンします。受信バッファが空の時、戻り値0でリターンします。

<使用例>

受信キャラクタがあるとき、そのキャラクタを送信（エコーバック）します。

```
#include "CT207Lib.h"
中略
char c; //SWの状態を保存する変数
UART6_Init();
中略
if(UART6_Getc(&c)){
    // 受信キャラクタがあるときの処理
    UART6_Putc(c); // エコーバック
}
else{
    // 受信キャラクタがないときの処理
}
```

■ SW_GetStat()

<概要>

| | |
|-----|--|
| 名前 | SW_GetStat |
| 機能 | CT-207, CT-201 のスイッチの状態を取得 |
| 引数 | なし |
| 戻り値 | unsigned char : 各スイッチの状態を表す データ使用例の様に、CT207Lib.h で定義された定数を使用して各スイッチの状態を判断します。 |

<注意事項>

あらかじめスイッチの接続されたポートが入力に設定されている必要があります。

<実行結果>

CT-207, CT-201 のスイッチが接続されたポートの状態より、各スイッチの状態を表すデータを戻り値として出力します。

<使用例>

SW_GetStat() 関数により変数 sw に現在のスイッチの状態を格納します。CT207Lib.h で定義されたビットマスクを使用して各スイッチの状態を判定します。

```
#include "CT207Lib.h"
中略
unsigned char SW; //SWの状態を保存する変数
中略
sw = SW_GetStat();
if(sw & SW2_MASK){
    //CT-207のSW2がONのときの処理
}
if(sw & LCDSW1_MASK){
    //CT-201のSW1がONのときの処理
}
```

CT207Lib.h におけるビットマスクの定義

```
#define SWSTAT_LCDSW4_MASK (1 << 5) // CT-201のSW4
#define SWSTAT_LCDSW3_MASK (1 << 4) // CT-201のSW3
#define SWSTAT_LCDSW2_MASK (1 << 3) // CT-201のSW2
#define SWSTAT_LCDSW1_MASK (1 << 2) // CT-201のSW1
#define SWSTAT_SW2_MASK (1 << 1) // CT-207のSW2
#define SWSTAT_SW1_MASK (1 << 0) // CT-207のSW1
```

5. サンプルプログラムの使い方

サンプルプログラムは PM plus プロジェクトで提供しています。書込む場合は、WriteEZ2 で機械語ファイル「CT207_CT201Sample.hex」を書込んでください。

PM plus で編集、ビルドする場合は、ワークスペースファイル「CT207_CT-201Sample.prw」をダブルクリックして PM plus を起動してください。

CT-207 における PM plus の操作方法、プログラムの書込みについては以下のスタートアップガイドを参照してください。

| 名称 | 内容 |
|--------------------------------------|--------------------------------------|
| 78K0S/KA1+ マイコンシリーズ スタートアップガイド C 言語編 | PM plus で C 言語プログラムの開発手順、プログラムの書込み方法 |

本資料について

- 本資料は、電子工作や電子回路、パーソナルコンピュータの操作について一般的な知識をお持ちの方を対象としています。
- 本資料を元に操作するには、NEC エレクトロニクス社製 78K0S/KA1+ シリーズマイコンについての知識や開発環境などが必要です。
- Microsoft®、Windows® は米国 Microsoft 社の米国およびその他の国における登録商標です。
- その他、記載されている製品名は各社の商標または登録商標です。
- 本資料に記載してある内容は、一般的な電子機器（学習教材、事務機器、計測機器、パーソナル機器、コンピュータ機器など）に使用されることを目的としています。高い品質や信頼性が要求され、故障や誤作動が直接人命を脅かしたり人体に危害を及ぼす恐れのある、医療、軍事、航空宇宙、原子力制御、運輸、移動体、各種安全装置などの機器への使用は意図も保証もしていません。
- 本資料の一部、又は全部を当社の承諾なしで、いかなる形でも転載又は複製されることは堅くお断りします。
- 全ての情報は本資料発行時点のものであり、当社は予告なしに本資料に記載した内容を変更することがあります。
- 本資料の内容は慎重に制作しておりますが、万一記述誤りによってお客様に損害が生じても当社はその責任を負いません。
- 本資料に関してのお問合せ、その他お気付きの点がございましたら、当社ホームページのお問い合わせページ (<https://www.sunhayato.co.jp/inquiry/>) よりお問合せください。
- 本資料に関する最新の情報はサンハヤト株式会社ホームページ (<http://www.sunhayato.co.jp/>) に掲載しております。

本資料のご利用にあたって

- 本資料に掲載している内容は、お客様が用途に応じた適切な製品をご購入頂くことを目的としています。その使用により当社及び第三者の知的財産権その他の権利に対する保証、又は実施権の許諾を意味するものではありません。また、権利の侵害に関して当社は責任を負いません。
- 本資料に記載した情報を流用する場合は、お客様のシステム全体で充分評価し適用可能かご判断願います。当社では適用可能判断についての責任を負いません。

