

# Cで開発する PICモジュール プログラム・サンプル集

見本

Beginner's guide to PIC programming with MikroC PRO

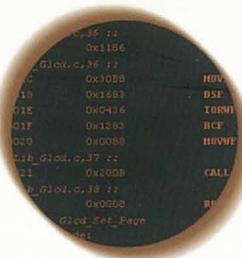
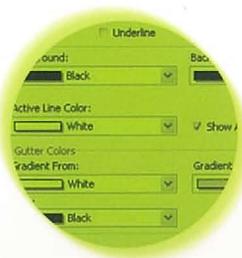
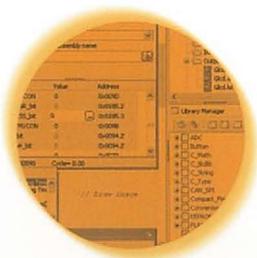


稲崎弘次 著

## mikroC PROで らくらくプログラミング



CD-ROM付き  
本書で解説した  
サンプル・プログラム  
を収録



CQ出版社

## ◎ 本書の狙い

PICを使った装置(電圧計, 電流計, 温度計, 照度計, 周波数カウンタ, 容量計, データ・ロガー, モータ制御, 時計など)を開発する際には, PICが内蔵するモジュールや外付けのモジュールを駆使して目的を実現することになります。

これらのモジュールを使用するためには, ある程度の経験や専門的な知識が必要です。

そこで, 本書では, よく使用されるモジュール(内蔵や外付け)をサンプル回路とサンプル・プログラムで制御方法を示し, 各モジュールの使い方を解説しました。

初心者が容易に理解を深めることができ, かつ, さまざまな自作装置の設計開発に応用できるよう, 事例となるサンプルの回路とプログラムは, 意図的に動作可能な最小構成としました。

また, プログラムの構造をシンプルにするために, プログラムをライブラリ化(関数化)してプログラミングを簡単にしています。

各関数内には, 処理の説明と関連する関数名や変数名などを明記しています。

処理の中で, mikroCが提供する標準ライブラリ(関数)を使用している箇所では, ヒントとして関連した関数についても明記しています。

掲載しているサンプル・プログラムは, 付属のCD-ROMに収録しています。

### 〈主要な内蔵モジュール〉

- I/Oポート
- TIMER
- A-D変換
- CCP
- USART
- コンパレータ
- EEPROM
- WDT

### 〈主要な外付けモジュール〉

- LCD
- 7セグメント
- A-D変換(12ビット) ※SPI通信
- D-A変換(12ビット) ※SPI通信
- EEPROM ※I<sup>2</sup>C通信
- RTC ※I<sup>2</sup>C通信
- キーパッド
- GLCD
- SDメモリ・カード

## ◎ 付属CD-ROMについて

本書で紹介するサンプル・プログラムは、すべて付属のCD-ROMに収められています。

mikroC PROのプロジェクト形式になっているので、そのままコピーしてコンパイルすることができます。

### 付属CD-ROMの内容

テーマ	フォルダ名	使用PIC名	キー・ポイント
2-1 I/OポートでLEDの交互点滅	sample_0001	PIC16F88	ポートの向き、アクティブ・ハイとロー、時間ディレイ
2-2 I/OポートのスイッチでLEDを制御	sample_0002	PIC16F88	チャタリング除去、ポートの反転
3-1 TIMER0でLEDの遅延点滅	sample_0003	PIC16F88	TIMER0の使用、クロック周波数、プリスケアラ、オーバフロー
3-2 TIMER1でLEDの遅延点滅	sample_0004	PIC16F88	TIMER1の使用、タイマの開始と停止
3-3 TIMER2でLEDの遅延点滅	sample_0005	PIC16F88	TIMER2の使用、ポストスケアラ、周期設定
3-4 ウォッチ・ドッグ・タイマでリセット	sample_0014	PIC16F88	異常時処理対策
4-1 LCDにASCII文字を表示させる	sample_0015	PIC16F88	キャラクタ表示
4-2 LCDに独自キャラクタを表示させる	sample_0016	PIC16F88	CGRAM表示
4-3 グラフィックLCDで表示	sample_0023	PIC16F88	グラフィック
4-4 7セグメントLEDで表示	sample_0017	PIC16F88	ダイナミック点灯
5-1 キー・パッドから直接入力	sample_0022	PIC16F88	キー・サーチ
5-2 内蔵EEPROMの読み書き	sample_0013	PIC16F88	データの記録と利用
5-3 SDメモリ・カードの読み書き	sample_0024	PIC18F2620	SDメモリ・カード(SDC)の読み書き
6-1 USARTモジュールでシリアル通信	sample_0009	PIC16F88	通信速度、データの受信と送信、パソコン通信
6-2 シリアル通信中の並列処理	sample_0010	PIC16F88	割り込みによる並列処理
7-1 A-D変換モジュール	sample_0006	PIC16F88	アナログ・チャネルの選択、上位と下位8ビット、ビット操作
7-2 CCPモジュールのPWMモード	sample_0007	PIC16F88	PWMモード、周期とデューティ、PWMの開始と停止
7-3 CCPモジュールのコンペア・モード	sample_0008	PIC16F88	コンペア・モード、割り込み処理
7-4 CCPモジュールのキャプチャ・モード	sample_0029	PIC16F88	キャプチャ・モード、パルス幅の測定
7-5 コンパレータの相対比較	sample_0011	PIC16F88	コンパレータのモード、電圧の比較
7-6 コンパレータの絶対比較	sample_0012	PIC16F88	コンパレータの参照電圧
8-1 A-D変換モジュール(12ビット)	sample_0018	PIC16F88	SPI通信、高精度A-D変換
8-2 D-A変換モジュール(12ビット)	sample_0019	PIC16F88	SPI通信、高精度D-A変換
8-3 外付けの大容量EEPROM	sample_0020	PIC16F88	I <sup>2</sup> C通信、大容量記憶
8-4 リアルタイム・クロック	sample_0021	PIC16F88	I <sup>2</sup> C通信、時刻制御
9-1 温度センサ	sample_0025	PIC16F88	A-D変換+LCD
9-2 周波数カウンタ	sample_0026	PIC16F88	TIMER+LCD
9-3 電卓	sample_0027	PIC16F88	キーパッド+LCD
9-4 データ・ロガー	sample_0028	PIC18F2620	A-D変換+SDC+LCD
9-5 ストップウォッチ	sample_0030	PIC16F88	TIMER2+7セグメントLED
9-6 電圧電流計	sample_0031	PIC16F88	D-A変換モジュール(12ビット)+LCD

# CONTENTS

まえがき .....	3
◎ 本書の狙い .....	4
◎ 付属CD-ROMについて .....	5
<b>[第1章] 開発の準備をしよう!</b> .....	<b>8</b>
1-1 使用するPICと開発環境 .....	8
サンプル・プログラムで使用するPICについて .....	8
開発環境について .....	11
開発の流れ .....	13
回路図の見方 .....	14
コンフィギュレーション・ビットの設定 .....	15
ライブラリの設定 .....	16
<b>[第2章] ポートをON/OFFさせてみよう!</b> .....	<b>18</b>
2-1 I/OポートでLEDの交互点滅 .....	18
2-2 I/OポートのスイッチでLEDを制御 .....	22
<b>Column</b> ...1 LED用の電流制限抵抗 .....	21
<b>Column</b> ...2 スwitchのチャタリング除去 .....	25
<b>[第3章] タイマを使ってみよう!</b> .....	<b>26</b>
3-1 TIMER0でLEDの遅延点滅 .....	26
3-2 TIMER1でLEDの遅延点滅 .....	30
3-3 TIMER2でLEDの遅延点滅 .....	34
3-4 ウォッチドッグ・タイマでリセット .....	38
<b>Column</b> ...3 TIMERの周波数カウンタへの応用 .....	33
<b>[第4章] データを表示させてみよう!</b> .....	<b>42</b>
4-1 LCDにASCII文字を表示させる .....	42
4-2 LCDに独自キャラクタを表示させる .....	46
4-3 グラフィックLCDで表示 .....	52
4-4 7セグメントLEDで表示 .....	60
<b>Column</b> ...4 mikroC PROが提供するツール .....	59
<b>Column</b> ...5 スタティック点灯とダイナミック点灯 .....	65
<b>[第5章] キー入力やデータの保存をしてみよう!</b> .....	<b>66</b>
5-1 キーボードから直接入力 .....	66

5-2	内蔵EEPROMの読み書き	70
5-3	SDメモリ・カードの読み書き	74
<b>Column</b> …6	Keypad(キー・マトリクス)の原理	69
<b>Column</b> …7	EEPROMへのデータの書き込みと読み出し	73
<b>Column</b> …8	SDメモリ・カードの応用	79
<b>[第6章] パソコンとデータ通信してみよう!</b>		<b>80</b>
6-1	USARTモジュールでシリアル通信	80
6-2	シリアル通信中の並列処理	84
<b>Column</b> …9	USARTの接続形態	83
<b>Column</b> …10	割り込み処理と並列処理	87
<b>[第7章] 測定してみよう!</b>		<b>88</b>
7-1	A-D変換モジュール	88
7-2	CCPモジュールのPWMモード	92
7-3	CCPモジュールのコンペア・モード	96
7-4	CCPモジュールのキャプチャ・モード	100
7-5	コンパレータの相対比較	106
7-6	コンパレータの絶対比較	110
<b>Column</b> …11	PWMによるモータ制御	95
<b>Column</b> …12	割り込み処理	99
<b>Column</b> …13	パルス幅の測定	105
<b>Column</b> …14	コンパレータとヒステリシス	109
<b>[第8章] 通信でモジュールを制御してみよう!</b>		<b>114</b>
8-1	A-D変換モジュール(12ビット)	114
8-2	D-A変換モジュール(12ビット)	120
8-3	外付けの大容量EEPROM	124
8-4	リアルタイム・クロック(RTC)	130
<b>Column</b> …15	A-D変換結果の換算方法	119
<b>[第9章] モジュールを組み合わせてみよう!</b>		<b>136</b>
9-1	温度センサ→A-D変換+LCD表示	136
9-2	周波数カウンタ→TIMER+LCD	140
9-3	電卓→キーパッド+LCD	146
9-4	データ・ロガー→A-D変換+SDC	152
9-5	ストップウォッチ→TIMER2+7セグメントLED	158
9-6	電圧電流計→D-A変換モジュール+LCD	166
<b>Column</b> …16	よく使われる温度センサの例	139
<b>Column</b> …17	カソード・コモンとアノード・コモン	165
<b>Column</b> …18	電圧と電流の測定	171
あとがき		172
索引		173

2-1

# I/OポートでLEDの交互点滅

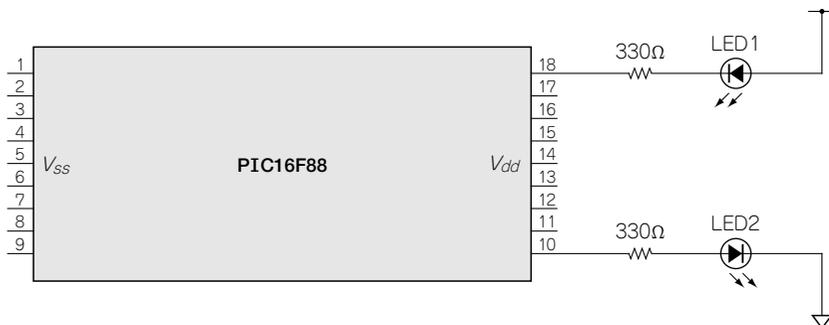
## 内蔵モジュールを使ってみよう!

### ● 機能概要

I/O(入出力)ポートは、最もよく使用するモジュールです。PICは、I/Oポートを複数持っており、プログラムで各ポートのピンを入力用または出力用に設定することができます。ここでは、LEDを接続するポートのピンを出力用に設定します。そのピンに対して、プログラムでhigh (1)を出力するとピンには約5Vの電圧が出力されるため、LEDが点灯し、low (0)を出力するとピンには約0Vの電圧が出力されるため、LEDが消灯します。

### ● サンプル回路

- LED1とLED2を使用します。
- LED1は、出力ポートがロー (low) になると点灯するので、アクティブ・ロー (low) になります。
- LED2は、出力ポートがハイ (high) になると点灯するので、アクティブ・ハイ (high) になります。



※電源とパソコンの接続は省略しています。詳細はp.14を参考にしてください。

図2-1 I/OポートでLEDの交互点滅

## <ポイント>

- I/Oポートに接続したLED1とLED2を交互に点滅させます。
- LEDは、I/Oポートへの接続形態によって、アクティブ・ロー (low) とアクティブ・ハイ (high) があるので、双方に対する制御を行います。
- デレイ (遅延) 関数を使用します。

## ● サンプル・プログラム

リスト2-1 関数宣言、マクロ定義、メイン関数、入出力ポート初期化関数

```
//*****  
//■■■■関数宣言■■■■  
extern void    main();  
extern void    init_port();  
//*****  
//■■■■マクロ定義■■■■  
//LED1  
sbit          LED1          at    PORTA.B1;  
sbit          LED1_Direction at    TRISA.B1;  
#define       LED1_ON      LED1 = 0  
#define       LED1_OFF     LED1 = 1  
//LED2  
sbit          LED2          at    PORTB.B4;  
sbit          LED2_Direction at    TRISB.B4;  
#define       LED2_ON      LED2 = 1  
#define       LED2_OFF     LED2 = 0  
//other  
#define       INPUT_MODE   1  
#define       OUTPUT_MODE  0  
//*****  
//■■■■メイン関数■■■■  
void    main()  
{  
    OSCCON = 0b01110000;    //クロックを8MHzに設定します。  
    ANSEL  = 0b00000000;    //A-D変換モジュールは使用しません。  
    //  
    init_port();  
    //  
    while (1) {  
        LED1_ON;  
        LED2_OFF;  
        Delay_ms(50);  
        //  
        LED1_OFF;  
        LED2_ON;  
        Delay_ms(50);  
    }  
}  
//*****  
//■■■■入出力ポート初期化関数■■■■  
void    init_port()  
{  
    LED1_Direction = OUTPUT_MODE;  
    LED2_Direction = OUTPUT_MODE;  
}  
//*****
```

//文字より右側は、コメントとみなされます。

このプログラムの中で作成する関数を宣言します。ここでは、メイン関数と入出力ポート初期化関数を宣言しています。

LED1を接続するポートおよびそのポートの方向(入力方向または出力方向)を設定するためのレジスタを割り当てます。

LED1をONまたはOFFするためのコードをマクロ化しておきます。マクロ化することにより、可読性を上げたり、変更個所の局所化を図ることが可能となります。

LED2についても、LED1と同様の定義を行います。

ポートの方向(入力方向または出力方向)を設定するための値をマクロ化しておきます。

PICのクロックを、内蔵の8MHzに設定します。A-D変換モジュールは、デフォルトで使用するために設定されているので、使用しないに設定し直します。

入出力ポート初期化関数を呼び出します。

メイン関数です。

While(1)にすることにより、処理を永続ループさせることができます。

LEDのON(点灯)OFF(消灯)を行います。その後、50msecのディレイ(遅延)を行います。ここでは、LED1とLED2を交互に点滅させています。

入出力ポート初期化関数です。

LED1とLED2を接続するポートの方向(入力方向または出力方向)を、出力方向に設定します。

9-4

# データ・ロガー→A-D変換+SDC

## モジュールを組み合わせてみよう!

### ● 機能概要

四つのアナログ・データを周期的(約0.1秒周期)に取り込み、電圧値に換算した値をSDC(SDメモリ・カード)に記録し、LCDに表示します。記録の開始と停止は、スイッチ(SW)により行います。サンプルは、記録と停止を繰り返した場合でも同じ名前のファイルに上書きしていますが、ON/OFFを繰り返すたびにファイル名を変更することで、より使いやすくなります。また、RTC(リアルタイム・クロック)を組み合わせるとより実用的になります。

### ● サンプル回路

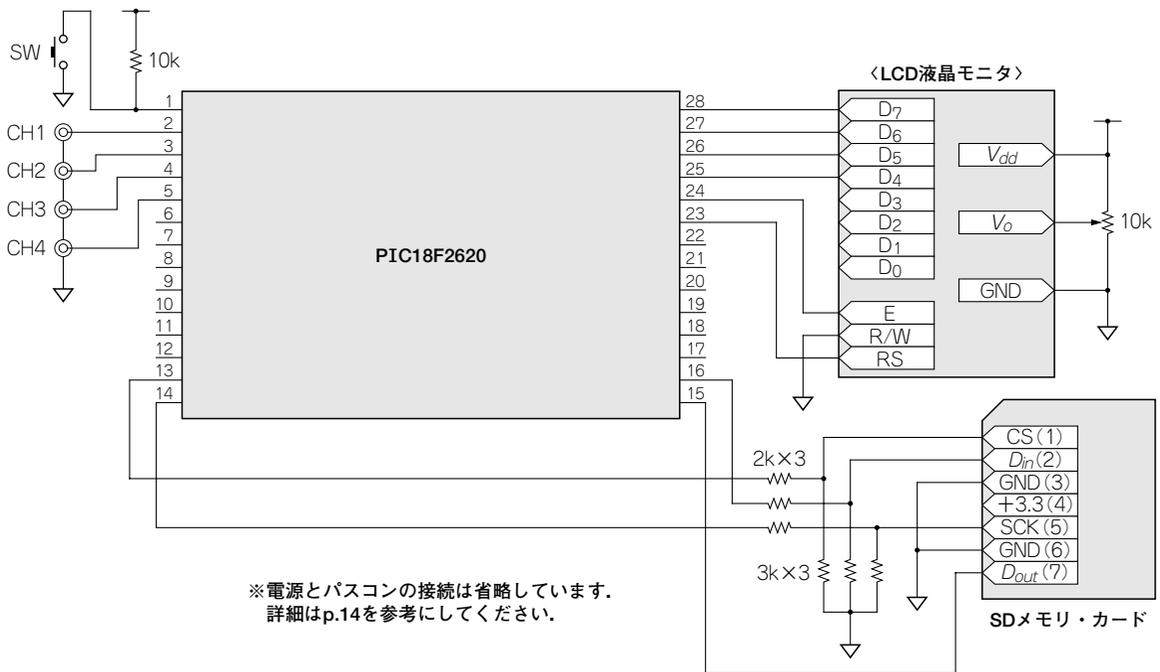


図9-6 データ・ロガー→A-D変換+SDC+LCD

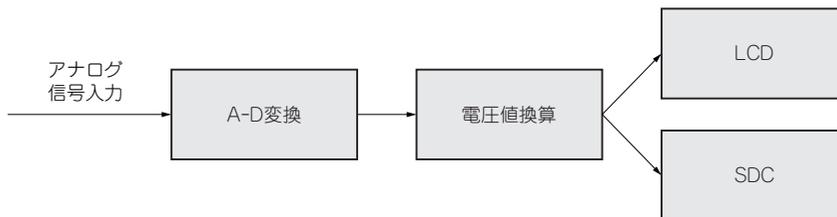


図9-7 データ・ロガーのブロック・ダイアグラム

```

1669 1694 1337 1372
1679 1630 1372 1396
1625 1606 1303 1518
1762 1684 1293 1577
1635 1464 1440 1572
1591 1533 1474 1533
1503 1611 1513 1406
1533 1650 1484 1342
  
```

図9-8 記録されるファイル  
“log.txt”のフォーマット

#### 〈使用するモジュール〉

- A-D変換モジュール
- SDC
- LCD

#### 〈ポイント〉

- SDCに記録するときのファイル名は“log.txt”です。

### ● プログラムの概要

#### 〈関数宣言〉(リスト9-11)

- メイン関数 (main) とADC初期化関数 (init\_adc) を宣言します。
- LCD初期化関数 (init\_lcd) を宣言します。
- SDC初期化関数 (init\_sdc) とSDCファイル・オープン関数 (sdc\_open) を宣言します。
- ロガー関数 (logger) を宣言します。

#### 〈マクロ定義〉(リスト9-11)

- SDCに関する定義 (チップ・セレクト信号) を記述します。
- LCDに関する定義 (制御信号とデータ信号) を記述します。
- ADCに関する定義 (入力チャンネルの向き) を記述します。
- スイッチに関する定義を記述します。

見本

ISBN978-4-7898-4216-7

C3055 ¥2800E

**CQ出版社**

定価：本体2,800円（税別）



9784789842167



1923055028005

Cで開発する  
**PICモジュール・**  
**プログラム・サンプル集**

Beginner's guide to PIC programming with MikroC PRO

このPDFは、CQ出版社発売の「Cで開発するPICモジュール・プログラム・サンプル集」の一部見本です。

内容・購入方法などにつきましては以下のホームページをご覧ください。

内容 <http://shop.cqpub.co.jp/hanbai//books/42/42161.htm>

購入方法 <http://www.cqpub.co.jp/hanbai/order/order.htm>