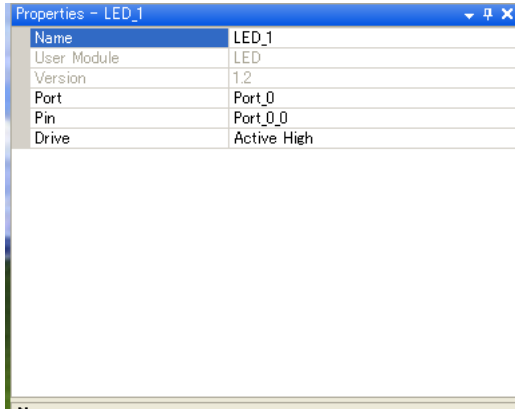


# 1. LED

LEDを光らせる。

- ・ LEDモジュールを置き、設定する。
- ・ 画面右下のUser Modulesツリーから、LEDモジュールを探して、ダブルクリック。
- ・ 画面右上のWorkspace ExplorerツリーのLED\_1をクリック。
- ・ 画面左中に設定項目が出る。



- ・ プログラムを書く。
- ・ 画面右上のWorkspace Explorerのmain.cをダブルクリック。

```
void main(void){
    LED_1_Start();
    LED_1_On();
    while(1){
    }
}
```

# 2. LED+Switch

LEDをスイッチで光らせたり、消したりする。

- ・ ピンの設定をする。画面右上のWorkspace Explorerから、projectname [Pinout]をダブルクリックする。
  - ・ LEDのピン (Drive : Strong)
  - ・ スイッチのピン (Drive : Pull Down)
- ・ プログラムを書く。

```
void main(void){
    while(1){
        PRT0DR &= 0b00000010;
        if((PRT0DR & 0b00000000) == 0b00000010){
            PRT0DR |= 0b00000001;
        } else {
            PRT0DR &= 0b11111110;
        }
    }
}
```

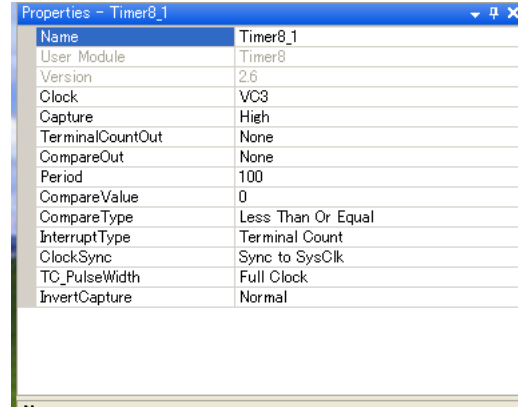
[補足]

- ・ PRT0DR (ポート0のデータレジスタ) を使うと、ポート0のピンの入力を見たり、出力を変えたりすることができる。
- ・ ただし、ピンの設定を忘れないように。
- ・ データレジスタを読み込む (参照する) と、何が入力されているか (0or1) がわかる。
- ・ データレジスタに書き込む (代入する) と、出力 (0or1) を変えることができる。

# 3. LED+Timer

LEDを一定間隔で点滅させる。

- ・ LEDモジュールを置き、設定する。
- ・ (もしくは、ピンの設定をする。)
- ・ Timer8モジュールを置き、設定する。



- ・ プログラムを書く。

BYTE flag;

```
void main(void){
    flag=0;
    while(1){
        M8C_EnableGInt;
        LED_1_Start();
        Timer8_1_EnableInt();
        Timer8_1_Start();
        while(1){
        }
    }
}

#pragma interrupt_handler Timer8_1_ISR
void Timer8_1_ISR(void){
    if(flag==0){
        LED_1_On();
        flag=1;
    }else if(flag==1){
        LED_1_Off();
        flag=0;
    }
}
```

## 4. LED+PWM

LEDの明るさを変える。

- ・PWM8モジュールを置き、設定する。

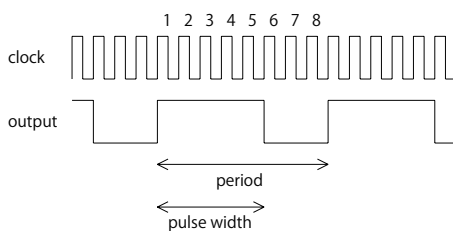
Properties - PWM8_1	
Name	PWM8_1
User Module	PWM8
Version	2.5
Clock	VC1
Enable	High
CompareOut	Row_0_Output_0
TerminalCountOut	None
Period	0
PulseWidth	0
Compare Type	Less Than Or Equal
InterruptType	Terminal Count
ClockSync	Sync to SysClk
InvertEnable	Normal

- ・PWM8モジュールから、LEDのピンへ内部配線する。
- ・LEDのピンの設定をする (Drive : Strong) 。
- ・プログラムを書く。

```
void main(void){
    int i;
    BYTE pulsewidth=0;
    PWM8_1_Start();
    PWM8_1_WritePeriod(100);
    while(1){
        PWM8_1_WritePulseWidth(pulsewidth);
        pulsewidth=(pulsewidth+1)%100;
        for(i=0; i<10000; i++);
    }
}
```

[補足]

- ・PWMとは、Pulse Width Modulation (パルス幅変調) の略で、パルス波を発生させることができる。
- ・period=周期、pulsewidth=Highの区間。
- ・デューティ比=pulsewidth/period
- ・pulsewidthとperiodを、WritePulseWidth()とWritePeriod()で書き変えることができる。
- ・下図は、period=8、pulsewidth=5のときの例。



## 5. LED+Sensor+ADC

光センサを用いて、LEDの明るさを変える。

- ・PWM8モジュールを置き、設定する。
- ・PWM8モジュールから、LEDのピンへ内部配線する。
- ・LEDのピンの設定をする (Drive : Strong) 。
- ・PGAモジュールを置き、設定する。

Properties - PGA_1	
Name	PGA_1
User Module	PGA
Version	3.2
Gain	1.000
Input	AnalogColumn_InputSelect_1
Reference	AGND
Analog Bus	Disable

- ・ADCINCモジュールを置き、設定する。

Properties - ADCINC_1	
Name	ADCINC_1
User Module	ADCINC
Version	1.1
DataFormat	Unsigned
Resolution	8 Bit
Data Clock	VC1
PosInput	ACB01
NegInput	ACB00
NegInputGain	Disconnected
Clock Phase	Normal
PulseWidth	1
PWM Output	None

- ・光センサのピンから、PGAモジュールを介し、ADCINCモジュールへ内部配線する。
- ・プログラムを書く。

BYTE data;

```
void main(void){
    M8C_EnableGInt;
    PWM8_1_WritePeriod(100);
    PWM8_1_Start();
    PGA_1_Start(PGA_1_HIGHPOWER);
    ADCINC_1_Start(ADCINC_1_HIGHPOWER);
    ADCINC_1_GetSamples(0);
    while(1){
        while(ADCINC_1_fIsDataAvailable()==0);
        data=ADCINC_1_bClearFlagGetData();
        data=(int)((double)data/256.0*100);
        PWM8_1_WritePulseWidth(data);
    }
}
```