

# PSoC Hardware Experiment

## How to fix your problem in your project

### B2 HWEX 2020, Meiji University

B2\_PRO.pptx 11 Slides July 21<sup>st</sup>, 2020

---

Version 1.32

Renji Mikami

# 課題製作の問題対処例 1

Wikiサイトの複数のソースコードをコピーして一気にコンパイルをかけてたくさんのエラーが出て、直しきれなくなる。

- 原因1 : コードの量が多くなるとバグの発生数がそれ以上に多くなる。バグがひとつだと楽に直せるが、複数になるとエラーの場所の特定がどんどん難しくなる。
- 対策1 : 少ないステップ数のコードでコンパイルを進めて、段階的に拡張しコンパイル&ゴーを進める。エラー発生時に、拡張したソース部分をコメントアウトしてコンパイルしてみる。
- 対策2 : 一気にビルドせずに、Generate Config > Compile > Build と順にエラー確認しながら進める。

# 課題製作の問題対処例 2

Wikiサイトの複数のソースコードをコピーしてまとめて、一気にビルドした。エラー表示が出なくても基板では動かない。

- 原因1 : Global Resource のクロック設定などは、各Projectで異なった値に設定されている。設定の異なるProjectをそのまま合体させてもうまくいかない。
- 対策1 : 合体させるProject間でGlobal Resource設定が共通化できるかを考える。
- 原因2 : 同じユーザーモジュールを使用しても設定が異なっている。
- 対策2 : ユーザーモジュールの設定を調べてみる。
- 原因3 : 外部スイッチやLEDを使用する場合のGPIO(ピン)のpinout設定が異なっている。
- 対策3 : View > Pinout でPinモジュールの設定を調べてみる。

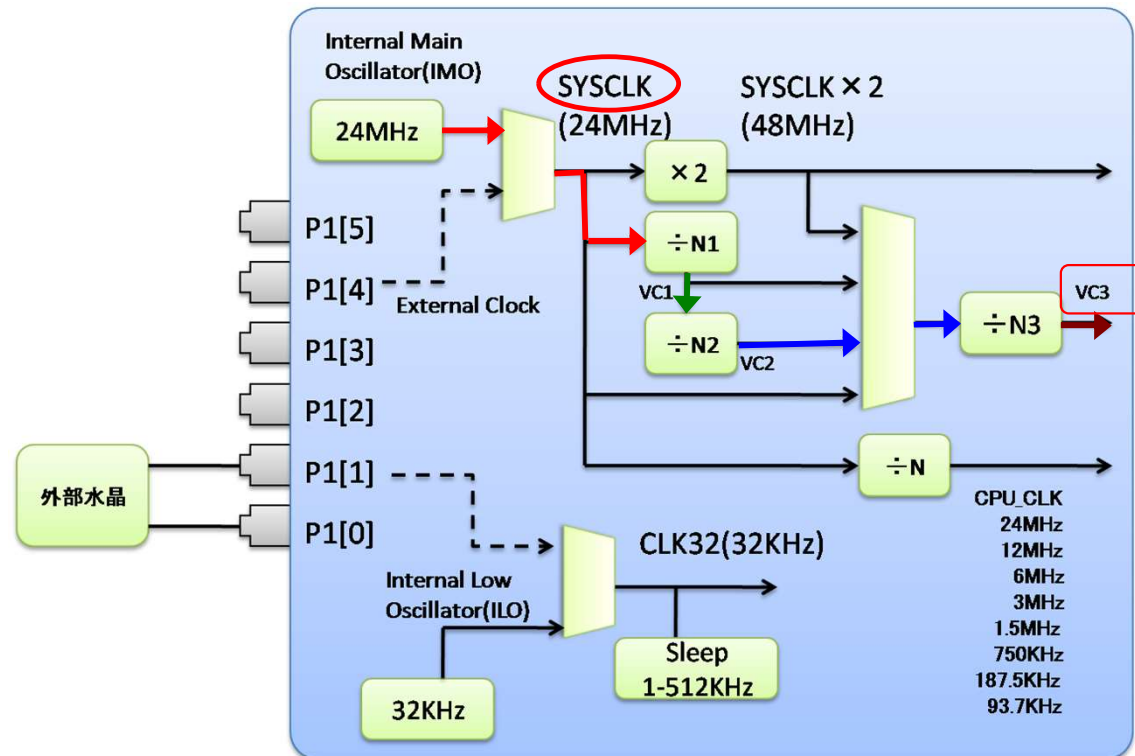
# Global Resources のクロック系共用設計

Counter8, PWM16, Timer16 で共通で使用できるように設定する

Global Resources - timer\_pwm2

CPU_Clock	3_MHz (SysClk/8)
32K_Select	Internal
PLL_Mode	Disable
Sleep_Timer	512_Hz
VC1= SysClk/N	4
VC2= VC1/N	4
VC3 Source	VC2
VC3 Divider	25
SysClk Source	Internal 24_MHz
SysClk*2 Disable	No
Analog Power	SC On/Ref Low
Ref Mux	(Vdd/2)+/-BandGap
AGndBypass	Disable
Op-Amp Bias	Low
A_Buff_Power	Low
SwitchModePump	OFF
Trip Voltage [LVD]	4.81V (5.00V)
LVDThrottleBack	Disable
Supply Voltage	5.0V
Watchdog Enable	Disable

**CPU\_Clock**  
Selects the CPU clock speed, from 93.75 KHz to 24 MHz. Derived from the SysClk. This setting affects the Power on Reset level in order to prevent the CPU from runnin...



timer\_pwm2 と PWM\_UART\_2 はクロック設定が同じなので統合できる

The diagram illustrates the clock distribution and configuration for two projects: 'timer\_pwm2' and 'PWM\_UART\_2'. Both projects share the same clock source, 'VC3', which is derived from the 'SYSCLK (24MHz)' signal. The 'VC3' signal is distributed to the 'timer\_pwm2' project's 'Timer 16.1' module and the 'PWM\_UART\_2' project's 'Counter8' module. The 'Counter8' module is configured with a clock of 'SysClk\*2', which is equivalent to 'VC3' in this context. The 'PWM16\_1' module in the 'timer\_pwm2' project is also configured with a clock of 'VC3'. The workspace explorer shows the project structures for both, and the parameter windows confirm the shared clock source and configuration.

**Workspace Explorer - timer\_pwm2**

- Workspace 'timer\_pwm2' (1 project)
  - timer\_pwm2 [CY8C27443-24PXI]
    - timer\_pwm2 [Chip]
      - timer\_pwm2 - 2 User Modules
        - PWM16\_1
        - Timer 16.1

**Workspace Explorer - PWM\_UART\_2**

- Workspace 'PWM\_UART\_2' (1 project)
  - PWM\_UART\_2 [CY8C27443-24PXI]
    - PWM\_UART\_2 [Chip]
      - pwm\_uart\_2 - 3 User Modules
        - Counter8
        - PWM16
        - UART

**Parameters - Counter8**

Name	Counter8
User Module	Counter8
Version	2.5
Clock	SysClk*2
ClockSync	Unsynchronized
Enable	High
CompareOut	None
TerminalCountOut	Row_0_Output_0
Period	155
CompareValue	78
CompareType	Less Than Or Equal
InterruptType	Terminal Count
InvertEnable	Normal

**Parameters - PWM16\_1 (timer\_pwm2)**

Name	PWM16_1
User Module	PWM16
Version	2.5
Clock	VC3
Enable	High
CompareOut	None

**Parameters - Timer16\_1 (timer\_pwm2)**

Name	Timer16_1
User Module	Timer16
Version	2.5
Clock	VC3
Capture	Low
TerminalCountOut	None
CompareOut	None
Period	9999
CompareValue	0
CompareType	Less Than Or Equal
InterruptType	Terminal Count
ClockSync	Sync to SysClk
TC_PulseWidth	Full Clock
InvertCapture	Normal

# クロック系設定が異なるプロジェクトの例 BPF / timer\_pwm\_2 / BPF

プロジェクトを統合する場合は双方で共有できるように Global Resource / クロック系を再設計します。

## BPFプロジェクト

Global Resources - bpf	
CPU_Clock	3_MHz (SysClk/8)
32K_Select	Internal
PLL_Mode	Disable
Sleep_Timer	512_Hz
VC1= SysClk/N	2
VC2= VC1/N	1
VC3 Source	VC1
VC3 Divider	1
SysClk Source	Internal 24 MHz
SysClk*2 Disab	No
Analog Power	SC On/Ref Low
Ref Mux	(Vdd/2)+/-BandGap
AGndBypass	Disable
Op-Amp Bias	Low
A_Buff_Power	Low
SwitchModePun	OFF
Trip Voltage [L	4.81V (5.00V)
LVDThrottleBac	Disable
Supply Voltage	5.0V
Watchdog Enab	Disable

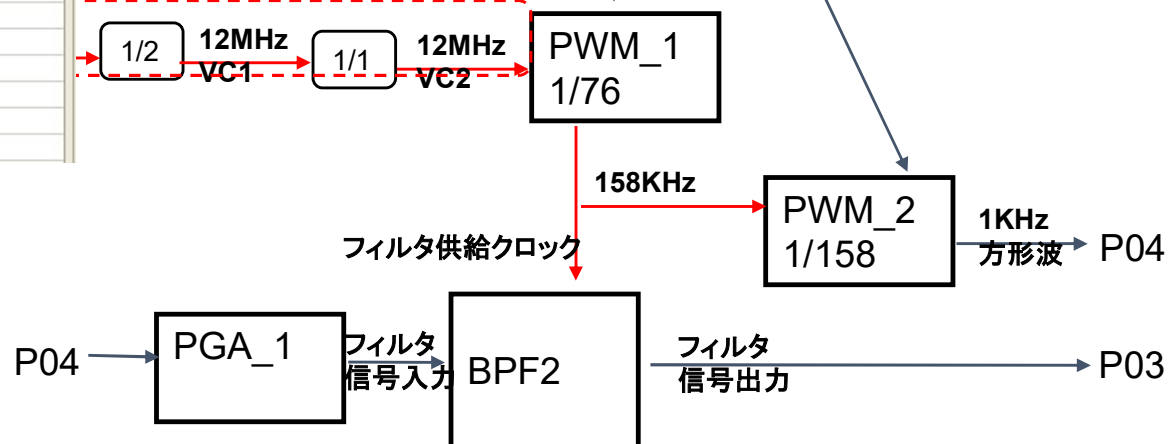
Properties - PWM8_1	
Name	PWM8_1
User Module	PWMS
Version	2.5
Clock	VC2
Enable	High
CompareOut	None
TerminalCountC	None
Period	75
PulseWidth	37
CompareType	Less Than Or Equal
InterruptType	Terminal Count
ClockSync	Sync to SysClk
InvertEnable	Normal

Properties - PWM8_2	
Name	PWM8_2
User Module	PWMS
Version	2.5
Clock	DBB00
Enable	High
CompareOut	Row_0_Output_0
TerminalCountC	None
Period	157
PulseWidth	78
CompareType	Less Than Or Equal
InterruptType	Terminal Count
ClockSync	Sync to SysClk
InvertEnable	Normal

## timer\_pwm2プロジェクト

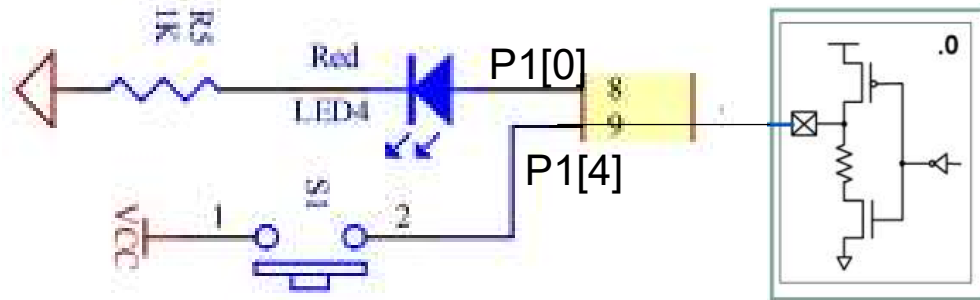
Global Resources - timer_pwm2	
CPU_Clock	3_MHz (SysClk/8)
32K_Select	Internal
PLL_Mode	Disable
Sleep_Timer	512_Hz
VC1= SysClk/N	4
VC2= VC1/N	4
VC3 Source	VC2
VC3 Divider	25
SysClk Source	Internal 24 MHz
SysClk*2 Disab	No
Analog Power	SC On/Ref Low
Ref Mux	(Vdd/2)+/-BandGap
AGndBypass	Disable
Op-Amp Bias	Low
A_Buff_Power	Low
SwitchModePun	OFF
Trip Voltage [LVD	4.81V (5.00V)
LVDThrottleBack	Disable
Supply Voltage	5.0V
Watchdog Enable	Disable

異なっています



## 外部スイッチやLEDを使用する場合のGPIO(ピン)のpinout設定

- 例 P1[0] : このピンが外部(基板上)のLEDに接続してある場合は、ピンをロジックStrongDrive出力にする設定する。(LEDの片側はGNDに接続済)
- Port\_1\_0, StdCPU, Strong, DisableInt, 0 に設定。



- 例 P1[4] : このピンが外部(基板上)のSWに接続してある場合は、ピンをロジック出力かつプルダウン設定にしてスイッチON時に電流が流れて入力がHIGHになるようにする。
- Port\_1\_4, StdCPU, Pull Down, DisableInt, 1 に設定。

詳細は、サイトの[gpio\\_poll.pdf](#)を参照してください。

The screenshot shows the 'Pinout - gpio\_poll' configuration window. The table below lists the configuration for various pins. Pin P1[0] is highlighted in blue, and its configuration is detailed in the table. Pin P1[4] is also highlighted, and a blue arrow points to the 'Drive' field, which is set to 'Pull Up'. The word 'Down' is written next to the arrow.

Pin	Port	StdCPU	High Z Analog	DisableInt	Initial
P1[0]	Port_1_0	StdCPU	Strong	DisableInt	0
P1[1]	Port_1_1	StdCPU	High Z Analog	DisableInt	
P1[2]	Port_1_2	StdCPU	High Z Analog	DisableInt	
P1[3]	Port_1_3	StdCPU	High Z Analog	DisableInt	
P1[4]	Port_1_4	StdCPU	Pull Up	DisableInt	1
P1[5]	Port_1_5	StdCPU	High Z Analog	DisableInt	
P1[6]	Port_1_6	StdCPU	High Z Analog	DisableInt	
P1[7]	Port_1_7	StdCPU	High Z Analog	DisableInt	

# 課題製作の問題対処例 3

どう考えても原因のわからないエラーに遭遇する

- 原因1 : 全角文字や空白、文字数、ディレクトリの深さ
- 対策1 : 英字を使用、C:¥直下にプロジェクトを置く、文字数を減らす
- 原因2 : OSや開発ソフトのバグの可能性
- 対策2 : 開発ソフト(やWindows)を再起動してみる
- 原因3 : ハードウェア問題の可能性
- 対策3 : デバイスの逆差しや足の曲がり、接続部品の接触不良を調べる、デバイスの不良の場合は交換する。
- 原因4 : 長時間作業では集中力が落ちてバグを見逃しやすくなる
- 対策4 : 休憩、リフレッシュ、日を変えて再チャレンジ



# うまい設計の進め方

## 距離センサーによって音程を変化させる課題例

- 失敗例：LAB3ADC(電圧計)のソースとtimer\_pwm2(ドレミ)のプロジェクトのどちらかのクローンをベースにもう一つのCソースをコピーして、必要なユーザーモジュールを追加して一気にビルド。エラーメッセージをもとにデバッグを進める。エラーが多くてデバッグに手間取る。
- 成功例：課題機能を複数のブロックに分割し、それぞれを動作検証してからつなぎ合わせる
- 機能1：センサーからの電圧を測り“変数”に代入するブロック
- 機能2：“変数”または“定数”から音を出すブロック
- 作業1：センサーの電圧を“変数”に格納してこの値をLCDに表示させ、距離と変数値の関係を調べてみる。
- 作業2：音を出すブロックを追加する前にGlobal Resource の設定値と追加するユーザーモジュールがうまく整合するかをチェックする。
- 作業3：音を出すブロックを追加して、“定数”を使ってブロック単独で音が出るかを確認する。
- 作業4：追加ブロックの定数を機能1ブロックの“変数”に置き換えて、2つのブロックを統合する。

# 設計の再開などについて

次の週に課題制作を継続する場合は、そのままの状態でも保管OKです。

制作中のプロジェクトは、念のため(事故防止のために)ディレクトリごとUSBメモリにコピーして保管することをおすすめします。

- 後方の棚に、チーム名を記載して、ジャンパー、センサーやアクチュエータをつないだまま保管できます。光センサー(cds-カドミウムセル)などは個体差がありますので、同じ個体で調整してそのまま制作をすすめてください。
- 開発ツールは、HPのURLから無償でダウンロードできます。自分用のノートPCを持ち込み開発してもかまいません。
- もっとやりたいチームには、基板などを貸し出しします。教室が空いている場合などは、PCやツールを使用できます。TAさんに聞いてみてください。

# Memo

- フォローアップURL
- <http://mikami.a.la9.jp/meiji/MEIJI.HTM>

- 担当講師
- 三上廉司(みかみれんじ)
- Renji\_Mikami(at\_mark)nifty.com (Default - Recommended)
- mikami(at\_mark)meiji.ac.jp (Alternative)
- [http://mikami.a.la9.jp/\\_edu.htm](http://mikami.a.la9.jp/_edu.htm)

