dsPIC33CK 入門

A Leading Provider of Smart, Connected and Secure Embedded Control Solutions

Linda Lai

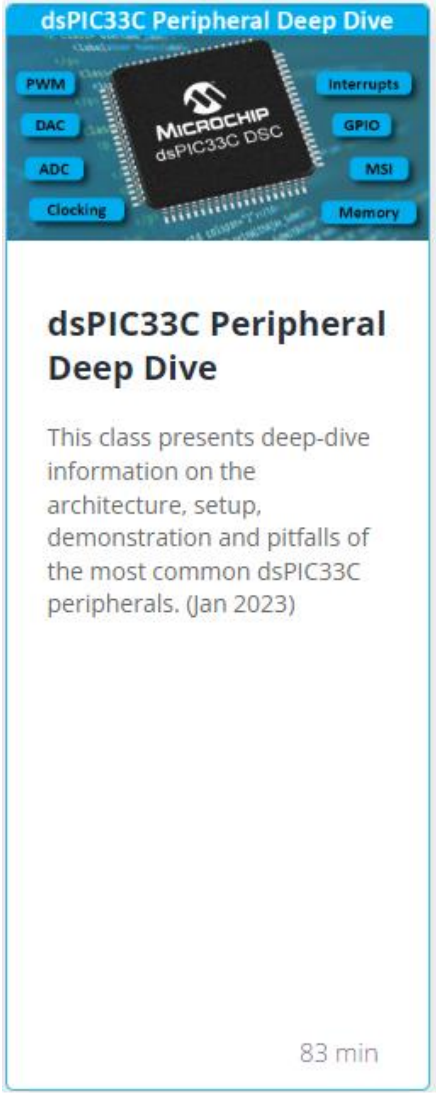
2024/3/26

SMART | CONNECTED | SECURE

大綱

- **dsPIC33CK DSC**介紹
- 開發工具介紹
- 入門程式(開發環境及硬體測試)
  - 實驗1：基礎I/O控制
  - 實驗2：透過UART顯示信息
- 內建硬體程式開發
  - 實驗3：Timer
  - 實驗4：Timer (SCCP)
  - 實驗5：Clock
  - 實驗6：Config(Special Features)

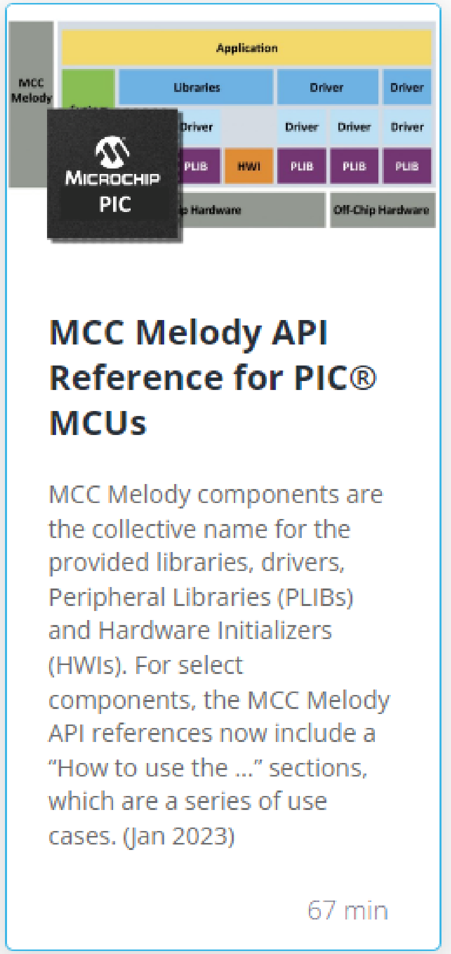
參考資料 – MICROCHIP UNIVERSITY



dsPIC33C Peripheral Deep Dive

This class presents deep-dive information on the architecture, setup, demonstration and pitfalls of the most common dsPIC33C peripherals. (Jan 2023)

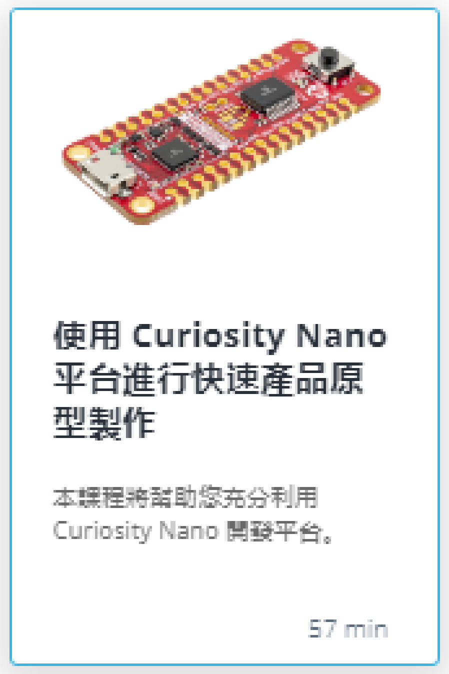
83 min



MCC Melody API Reference for PIC® MCUs

MCC Melody components are the collective name for the provided libraries, drivers, Peripheral Libraries (PLIBs) and Hardware Initializers (HWIs). For select components, the MCC Melody API references now include a “How to use the …” sections, which are a series of use cases. (Jan 2023)

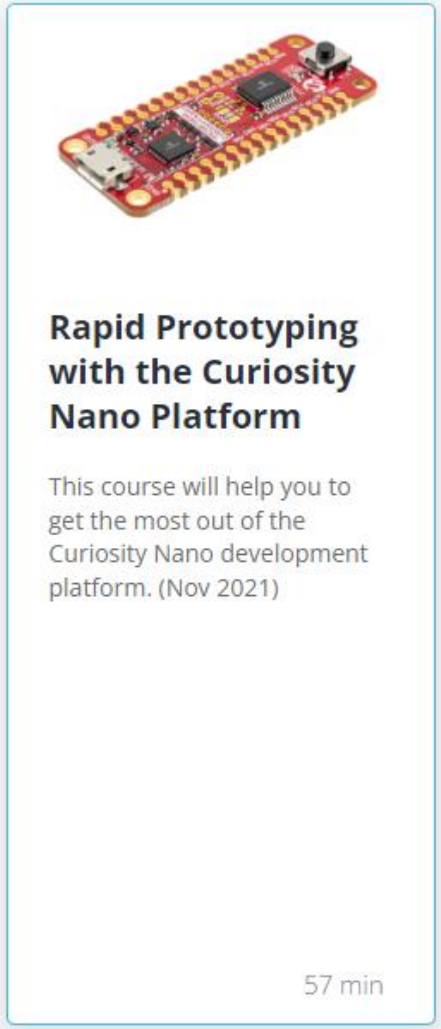
67 min



使用 Curiosity Nano 平台進行快速產品原型製作

本課程將幫助您充分利用 Curiosity Nano 開發平台。

57 min



Rapid Prototyping with the Curiosity Nano Platform

This course will help you to get the most out of the Curiosity Nano development platform. (Nov 2021)

57 min

https://mu.microchip.com

參考資料 – CAE專家教室

PIC1001課程

為基於學習 **MCC(MPLAB Code Configurator)程式庫產生器** 而開發, 課程中採用 **APP041**實驗板, 並引入 **PIM模組子板** 概念，使用者可以透過購買不同的PIM模組子板, 來更換所需的MCU 8/16/32-bit 微控制器.

APP041實驗板出廠已配置了 **PIC24FJ128GB106 16-bit MCU**, 而本課程會再搭配 **dsPIC33CK256MP506 (APP041-3 PIM模組子板)** 來上課, 為學習如何使用 **MCC程式庫產生器**,提供了一個基礎且紮實的入門課程.

本課程共計 **14章節** 與 **17個動手操作實驗**.

PIC1001-Melody課程 NEW

為基於學習 **MCC(MPLAB Code Configurator) Melody** 程式庫產生器 的 基礎驅動程式(Driver) 與 周邊函式庫 **Peripheral Library(PLIB)** 而開發, 課程中採用 **APP041** 實驗板, 主晶片為 **dsPIC33CK 16-bit MCU**, 並引入 **PIM模組子板** 概念，使用者可以透過購買不同的PIM模組子板, 來更換所需的MCU 8/16/32-bit 微控制器.

APP041實驗板出廠已配置了 **PIC24FJ128GB106 16-bit MCU**, 而本課程會再搭配 **dsPIC33CK256MP506 (APP041-3 PIM模組子板)** 來上課, 為學習如何使用 **MCC Melody** 程式庫產生器,提供了一個基礎且紮實的入門課程.

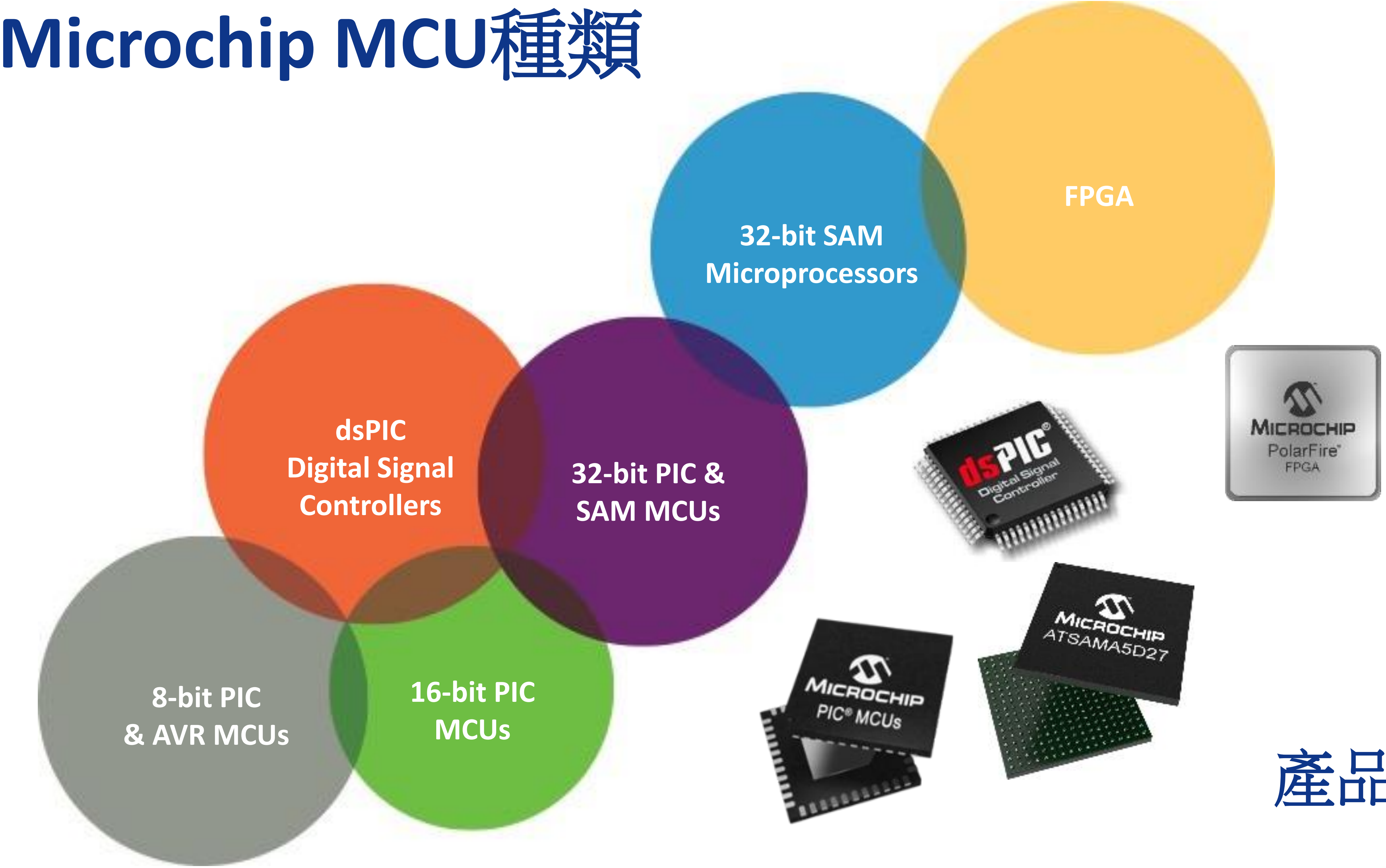
本課程共計 **14章節** 與 **17個動手操作實驗**.

https://www.microchip.com.tw/modules/tadnews/page.php?ncsn=2&nsn=222#PageTab8

dsPIC33CK - DSC

Digital Signal Controllers

Microchip MCU種類



FPGA

32-bit SAM Microprocessors

dsPIC Digital Signal Controllers

32-bit PIC & SAM MCUs

8-bit PIC & AVR MCUs

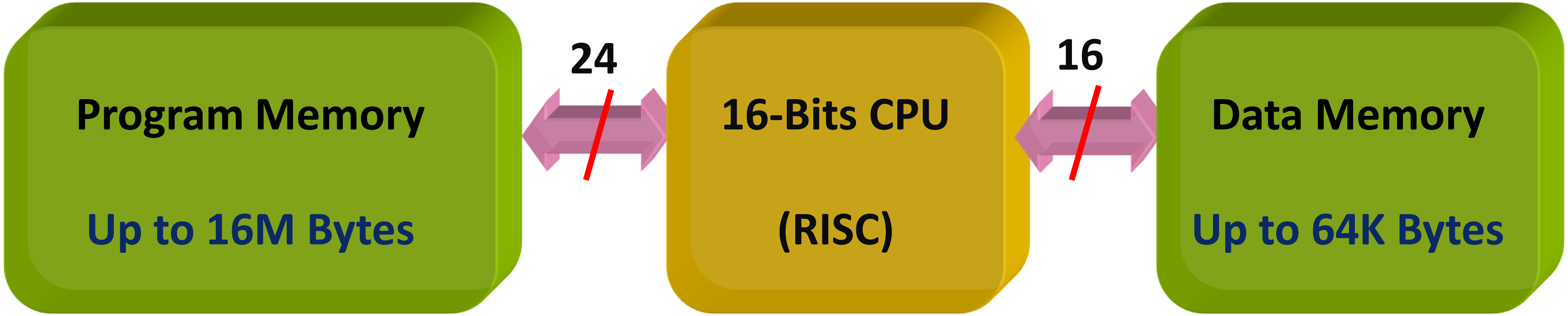
16-bit PIC MCUs

產品多樣

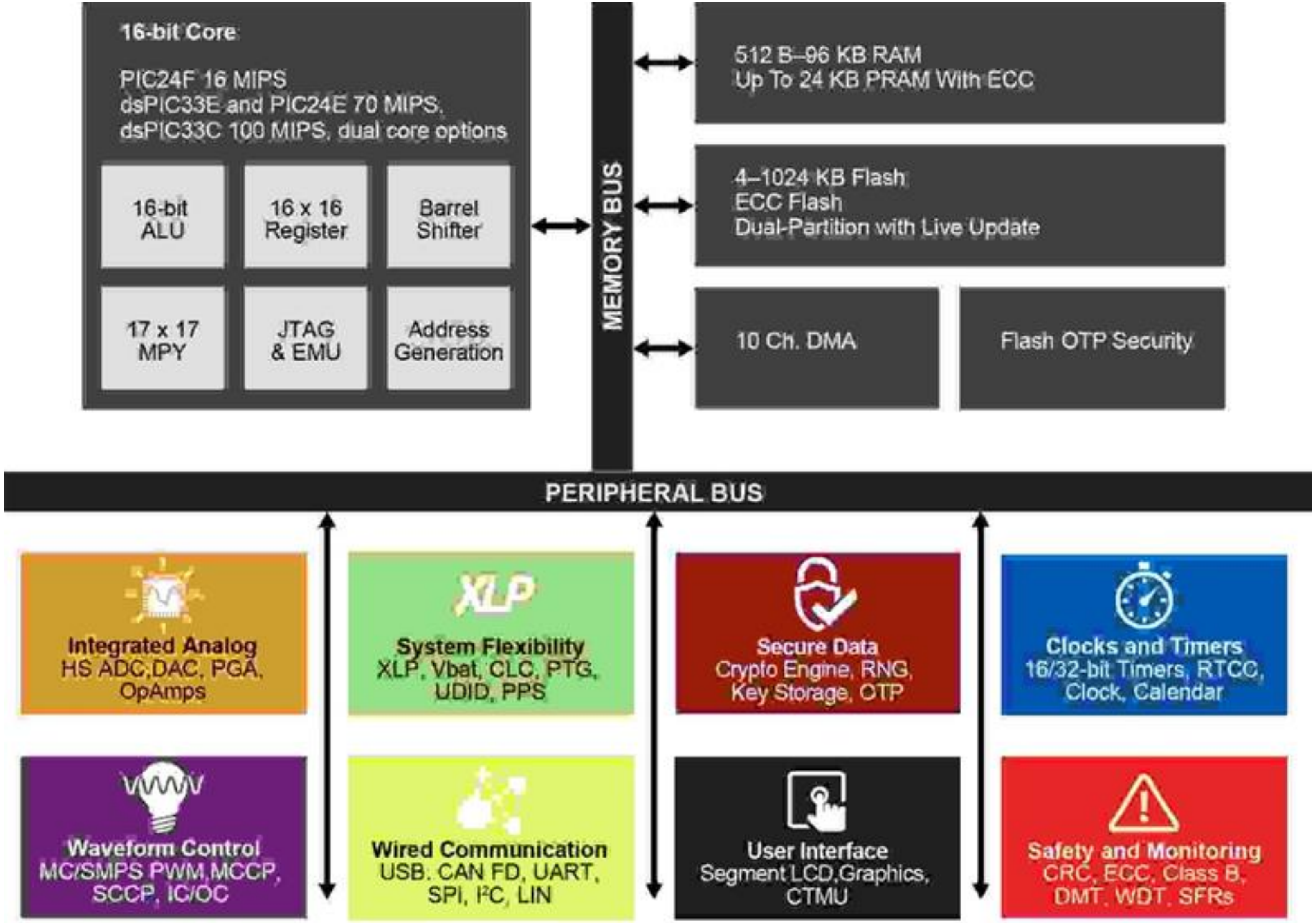
16-bits 核心架構

- Microchip 16-bits PIC/dsPIC® MCU 是哈佛架構 (Harvard)
  - 意思是程式記憶體跟資料記憶體各自擁有獨立的匯流排.

- 程式記憶體匯流排為24-bits, 最大到16M Bytes
- 資料記憶體匯流排為16-bits, 最大到64K Bytes



16-bits MCU方塊圖



16-bit Core

PIC24F 16 MIPS
dsPIC33E and PIC24E 70 MIPS,
dsPIC33C 100 MIPS, dual core options

16-bit ALU

16 x 16 Register

Barrel Shifter

17 x 17 MPY

JTAG & EMU

Address Generation

MEMORY BUS

512 B–96 KB RAM
Up To 24 KB PRAM With ECC

4–1024 KB Flash
ECC Flash
Dual-Partition with Live Update

10 Ch. DMA

Flash OTP Security

PERIPHERAL BUS

Integrated Analog
HS ADC,DAC, PGA,
OpAmps

XLP

System Flexibility
XLP, Vbat, CLC, PTG,
UDID, PPS

Secure Data
Crypto Engine, RNG,
Key Storage, OTP

Clocks and Timers
16/32-bit Timers, RTCC,
Clock, Calendar

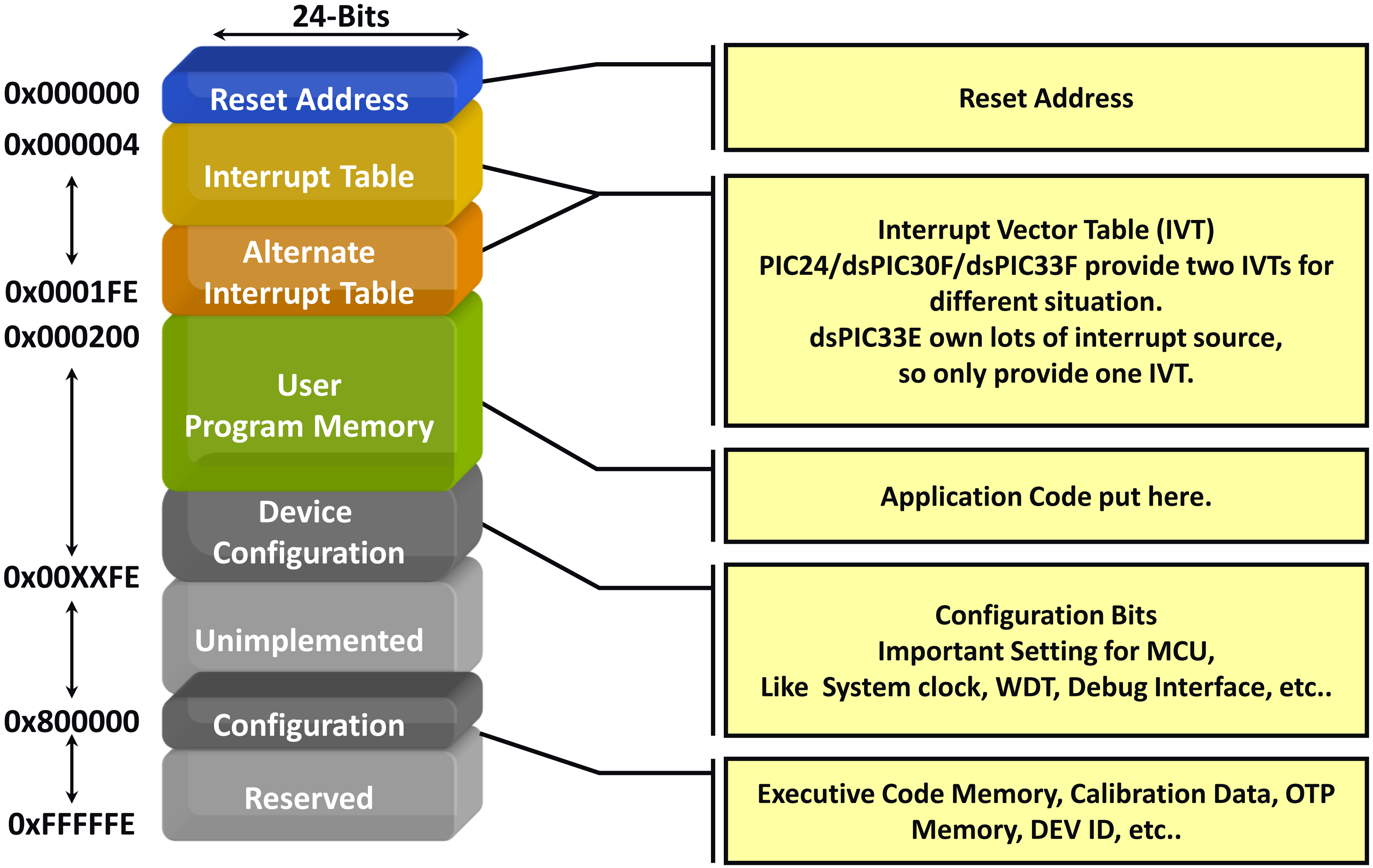
Waveform Control
MC/SMPS PWM,MCCP,
SCCP, IC/OC

Wired Communication
USB, CAN FD, UART,
SPI, I²C, LIN

User Interface
Segment LCD,Graphics,
CTMU

Safety and Monitoring
CRC, ECC, Class B,
DMT, WDT, SFRs

程式記憶體規劃



Reset Address

Interrupt Vector Table (IVT)
PIC24/dsPIC30F/dsPIC33F provide two IVTs for different situation.
dsPIC33E own lots of interrupt source, so only provide one IVT.

Application Code put here.

Configuration Bits
Important Setting for MCU,
Like System clock, WDT, Debug Interface, etc..

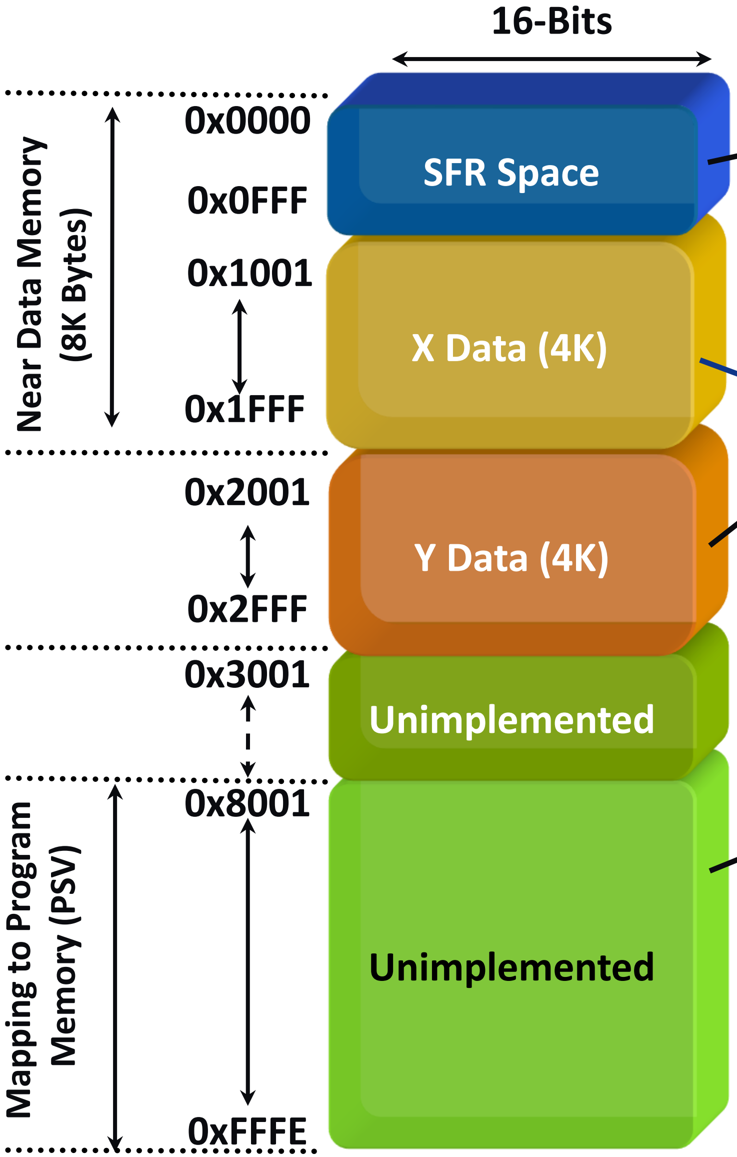
Executive Code Memory, Calibration Data, OTP Memory, DEV ID, etc..

Program Memory Mapping @dsPIC33CK64MC105

資料記憶體規劃

The **8K** byte area is referred as the **near** data space.

The **64K** byte area is referred as the **far** data space.



The first 4K bytes are primarily occupied with Special Function Registers (SFRs).

X Data, Y Data area only for DSP Engine For Dual Access Operation.

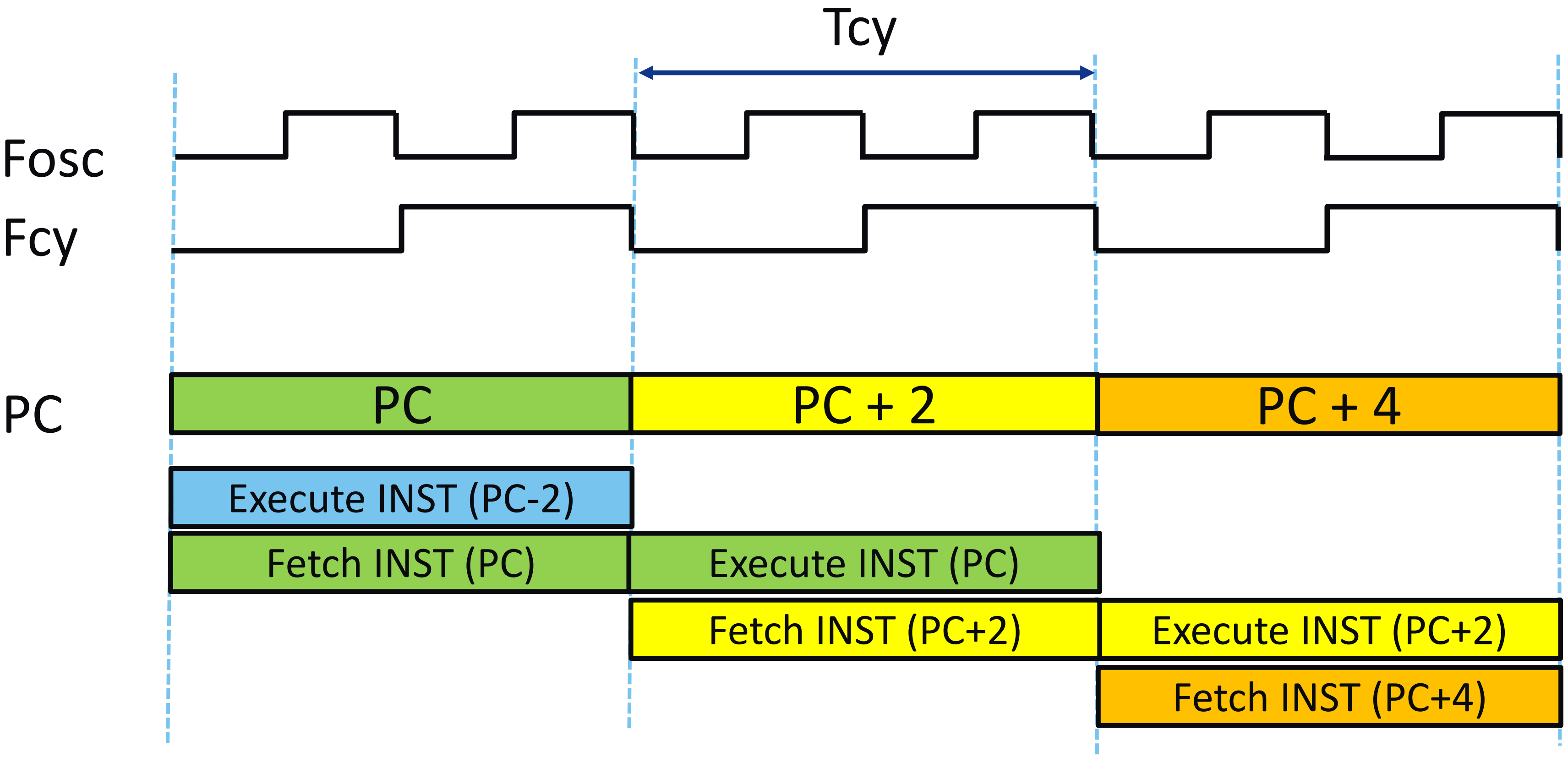
All Data is X Data area for PIC24 series.

0x8000 ~ 0xFFFF is mapping area for PSV use

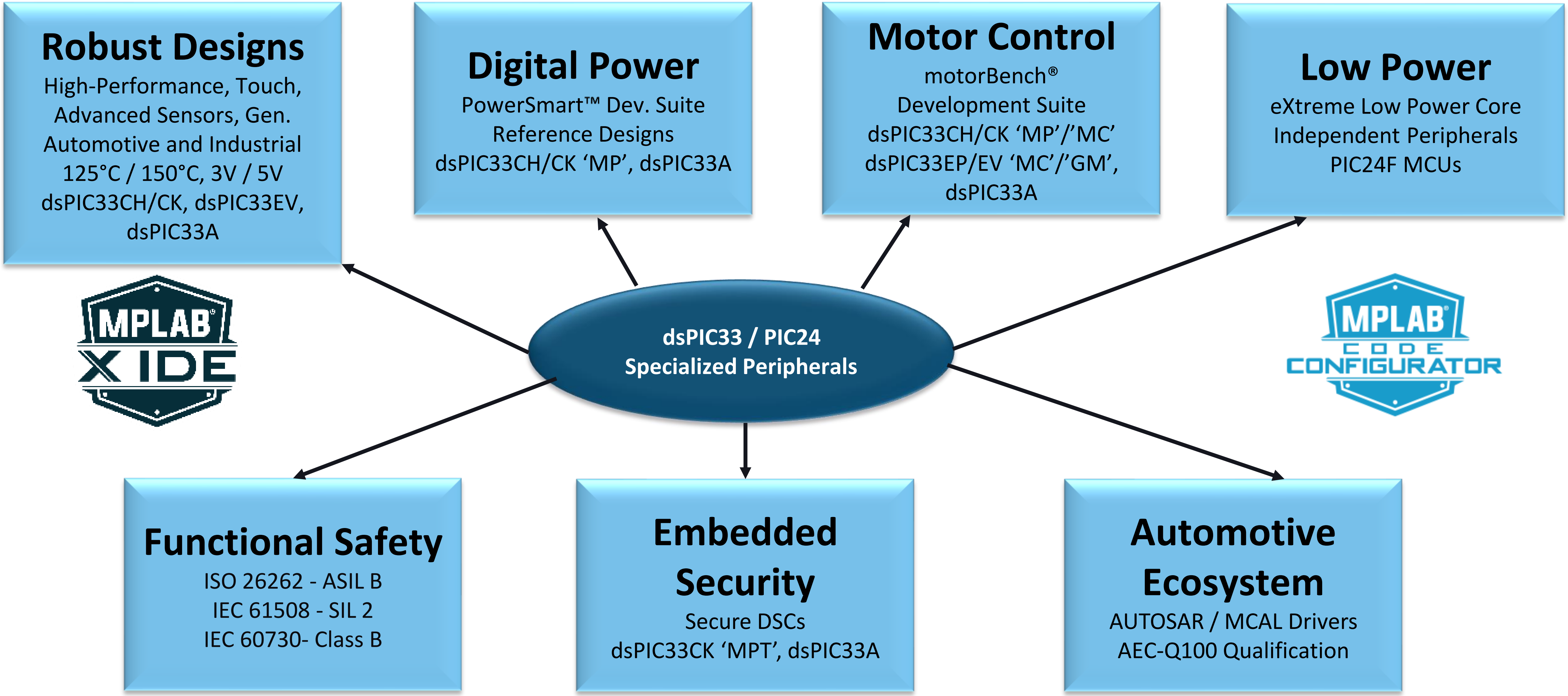
Data Memory Mapping @dsPIC33CK64MC105

dsPIC33C 指令執行流程

兩個震盪器的Clock 完成一個指令週期



dsPIC33 DSCs and PIC24 MCUs – 強項



dsPIC33 / PIC24
Specialized Peripherals

Robust Designs
High-Performance, Touch, Advanced Sensors, Gen. Automotive and Industrial
125°C / 150°C, 3V / 5V
dsPIC33CH/CK, dsPIC33EV, dsPIC33A

Digital Power
PowerSmart™ Dev. Suite
Reference Designs
dsPIC33CH/CK 'MP', dsPIC33A

Motor Control
motorBench®
Development Suite
dsPIC33CH/CK 'MP'/'MC'
dsPIC33EP/EV 'MC'/'GM', dsPIC33A

Low Power
eXtreme Low Power Core
Independent Peripherals
PIC24F MCUs

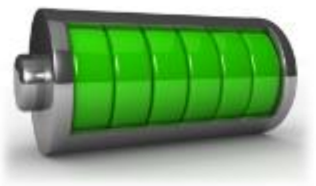
Functional Safety
ISO 26262 - ASIL B
IEC 61508 - SIL 2
IEC 60730- Class B

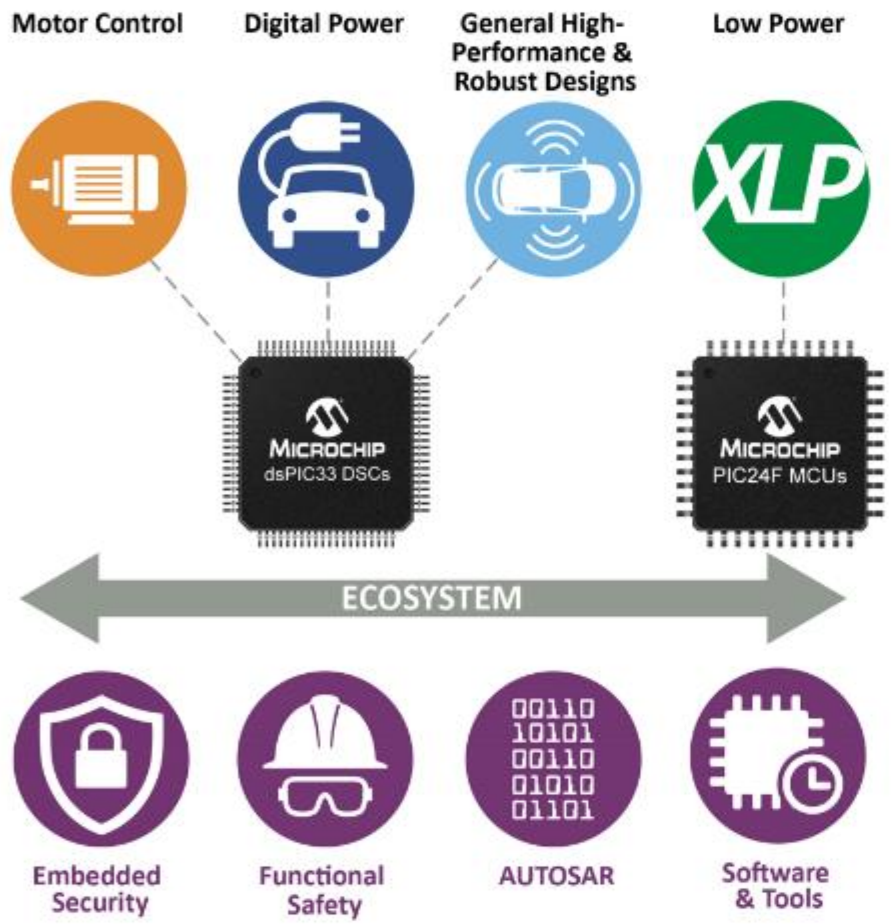
Embedded Security
Secure DSCs
dsPIC33CK 'MPT', dsPIC33A

Automotive Ecosystem
AUTOSAR / MCAL Drivers
AEC-Q100 Qualification

dsPIC33 DSCs / PIC24 MCUs - 強項

- **dsPIC33 = High-Performance, Real-Time Control**
  - **Robust and General High-Performance Embedded Designs**
    - Reliable operations in extreme conditions (150°C), AEC Q100 Grade 0
    - Advanced sensor interfacing, real-time control, touch and general high-performance designs in automotive and industrial
  - **Motor Control**
    - Market leader in high efficiency and performance motor control
    - Simplifying motor control with motorBench® Development Suite
  - **Digital Power**
    - Market leader in >300W power supplies
    - Automotive expansion in DC/DC, OBC, wireless power, USB PD
    - Large momentum for wireless power: automotive, appliance, tools
    - Simplifying digital power with PowerSmart™ Development Suite
- **PIC24F = Low Cost, Low Power, High Memory**
  - Low-cost, low-power families with more memory and faster peripherals
  - Stay in the PIC® ecosystem! - Easy migration from 8-bit PIC MCUs
- **Functional Safety with dsPIC33 DSCs and PIC24 MCUs**
  - Certified Functional Safety Solution for ISO 26262, IEC 61508, IEC 60730
- **Embedded Security with dsPIC33 DSCs and PIC24 MCUs**
  - dsPIC33C MPT Secure DSCs and dsPIC33A DSCs with integrated subsystem
  - Robust security together with ATECC608/TA100
- **Automotive Ecosystem for dsPIC33C DSCs**
  - AUTOSAR, MCAL Drivers and Tools





MPLAB® Code Configurator

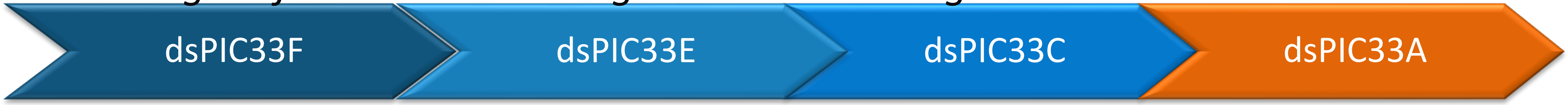
MPLAB® X IDE

MPLAB® XC Compiler

MPLAB® ICD 4

dsPIC33 DSC 的發展

Increasing Performance and Integration with each generation



dsPIC33F

- 40 MHz
- Single core
- Flexible PWMs
- Multiple S/H 10-bit ADC
- 16 – 256 KB Flash

dsPIC33E

- 70 MHz
- Single core
- Improved PWMs
- 12-bit ADC
- Integrated Op-Amps
- 32 – 512 KB Flash
- 5x5 mm package

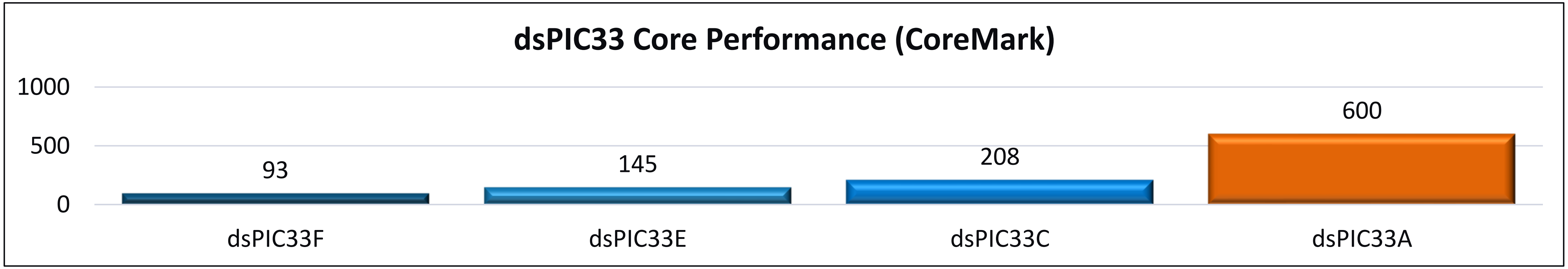
dsPIC33C

- 100 MHz
- Single/dual cores
- Improved PWMs
- 3.5 Msps ADCs
- Integrated PGAs
- Functional safety
- 4x4 mm package

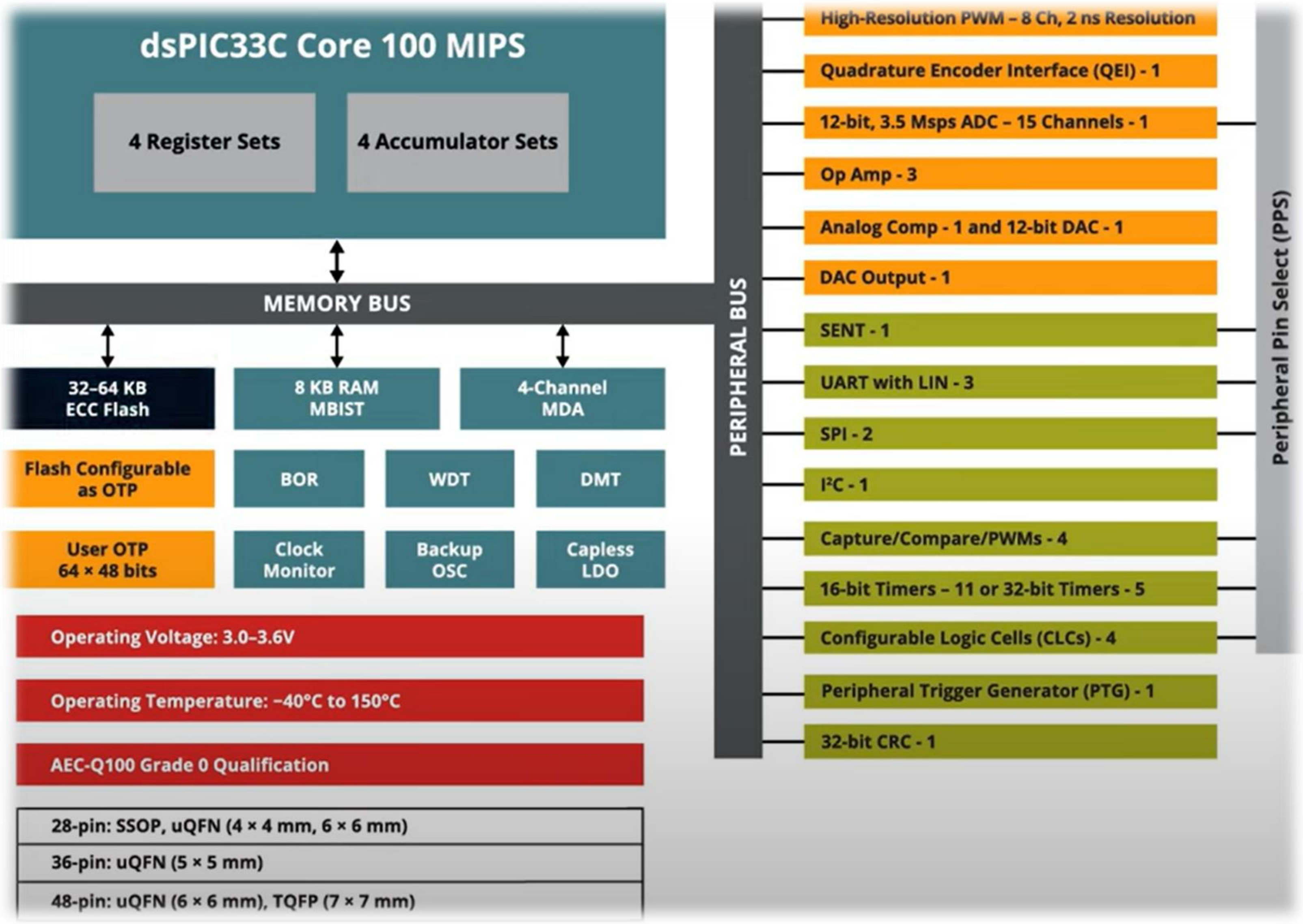
dsPIC33A

- 200+ MHz
- Single/dual cores
- 32-bit math operations, DP-FPU
- Faster ADCs & Op-Amp
- Functional Safety and Security
- Step up from dsPIC33C/E
- Backwards compatibility

As performance requirements increase, dsPIC33A DSCs provide a next step up in performance while preserving legacy code and ecosystem. The dsPIC33A DSCs offer common peripherals and look and feel to the older dsPIC33 generations and provide an easy migration path forward and still maintain low latency real-time control



本課程用的MCU - dsPIC33CK64MC105



dsPIC33C Core 100 MIPS

4 Register Sets

4 Accumulator Sets

MEMORY BUS

32–64 KB ECC Flash

8 KB RAM MBIST

4-Channel MDA

Flash Configurable as OTP

BOR

WDT

DMT

User OTP 64 × 48 bits

Clock Monitor

Backup OSC

Capless LDO

Operating Voltage: 3.0–3.6V

Operating Temperature: –40°C to 150°C

AEC-Q100 Grade 0 Qualification

28-pin: SSOP, uQFN (4 × 4 mm, 6 × 6 mm)

36-pin: uQFN (5 × 5 mm)

48-pin: uQFN (6 × 6 mm), TQFP (7 × 7 mm)

PERIPHERAL BUS

High-Resolution PWM – 8 Ch, 2 ns Resolution

Quadrature Encoder Interface (QEI) - 1

12-bit, 3.5 Msps ADC – 15 Channels - 1

Op Amp - 3

Analog Comp - 1 and 12-bit DAC - 1

DAC Output - 1

SENT - 1

UART with LIN - 3

SPI - 2

I²C - 1

Capture/Compare/PWMs - 4

16-bit Timers – 11 or 32-bit Timers - 5

Configurable Logic Cells (CLCs) - 4

Peripheral Trigger Generator (PTG) - 1

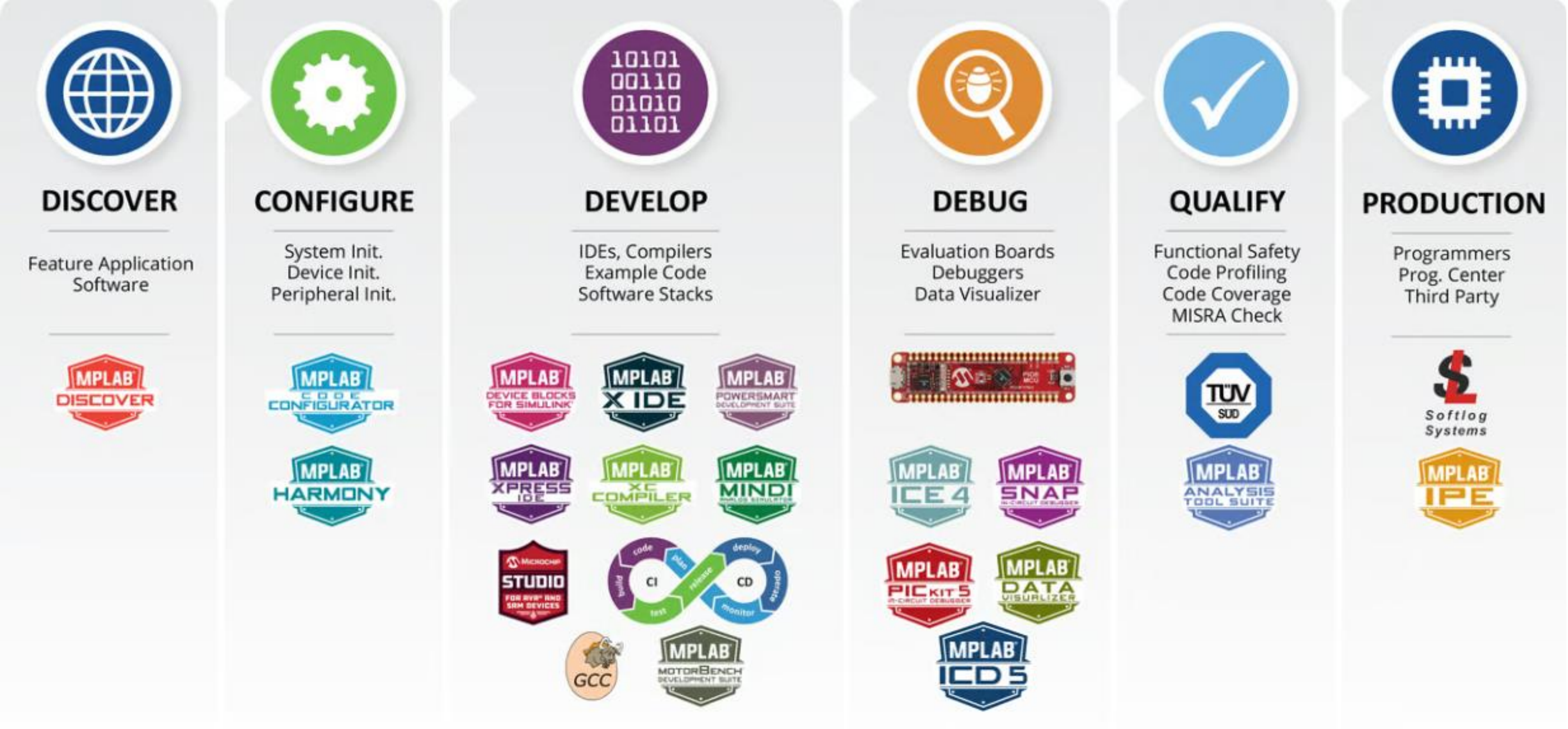
32-bit CRC - 1

Peripheral Pin Select (PPS)

開發工具介紹

X IDE,XC,MCC及相關硬體

Microchip 開發工具



DISCOVER

Feature Application Software

CONFIGURE

System Init.
Device Init.
Peripheral Init.

DEVELOP

IDEs, Compilers
Example Code
Software Stacks

DEBUG

Evaluation Boards
Debuggers
Data Visualizer

QUALIFY

Functional Safety
Code Profiling
Code Coverage
MISRA Check

PRODUCTION

Programmers
Prog. Center
Third Party

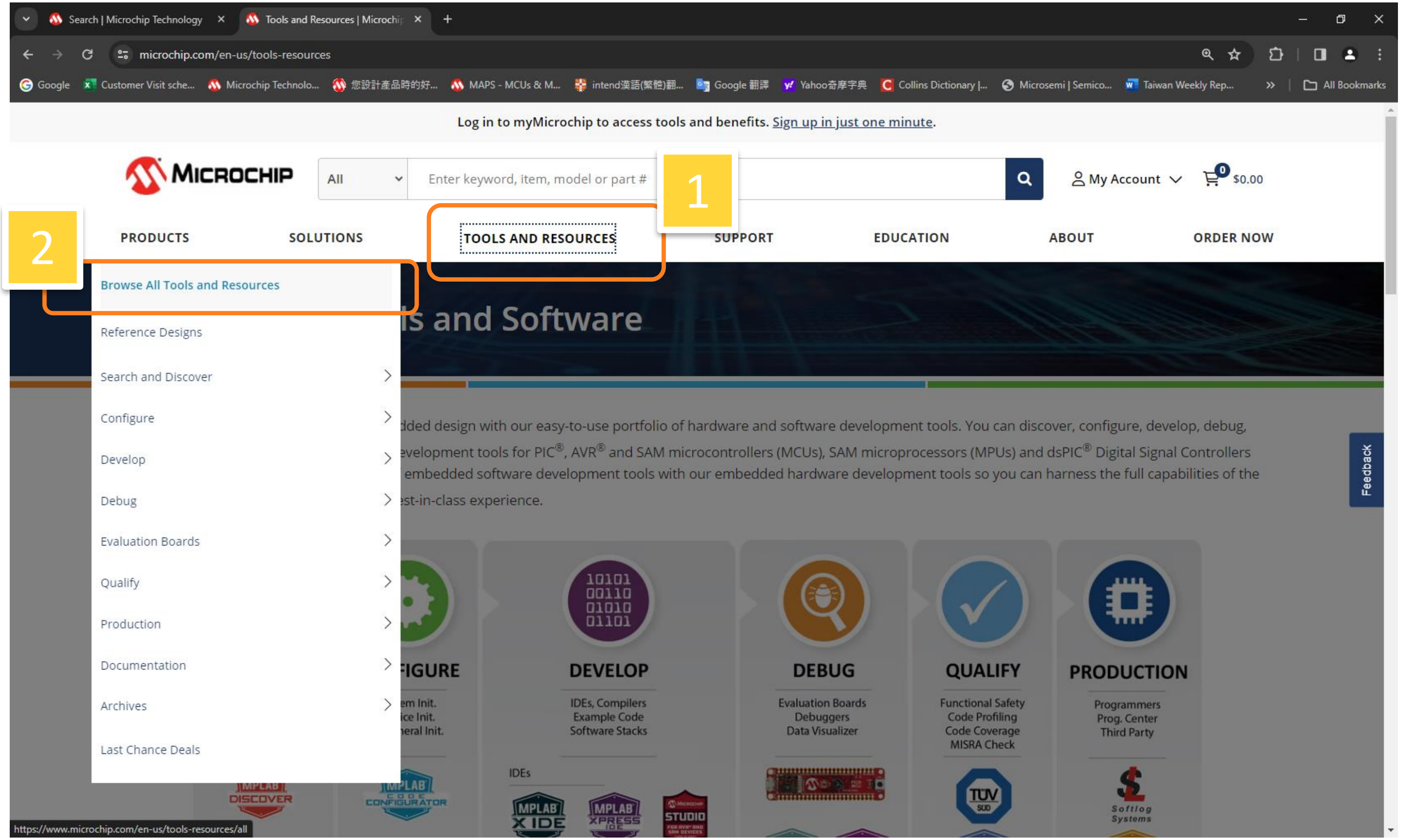
MPLAB DISCOVER
MPLAB CODE CONFIGURATOR
MPLAB HARMONY
MPLAB DEVICE BLOCKS FOR SIMULINK
MPLAB X IDE
MPLAB POWERSMART DEVELOPMENT SUITE
MPLAB XPRESS IDE
MPLAB XC COMPILER
MPLAB MINDI ANALOG SIMULATOR
MICROCHIP STUDIO FOR AVR AND SAM DEVICES
code
plan
deploy
build
CI
release
CD
operate
test
monitor
GCC
MPLAB MOTORBENCH DEVELOPMENT SUITE
MPLAB ICE 4
MPLAB SNAP IN-CIRCUIT DEBUGGER
MPLAB PICKIT 5 IN-CIRCUIT DEBUGGER
MPLAB DATA VISUALIZER
MPLAB ICD 5
TÜV SÜD
MPLAB ANALYSIS TOOL SUITE
Softlog Systems
MPLAB IPE

https://www.microchip.com/en-us/tools-resources

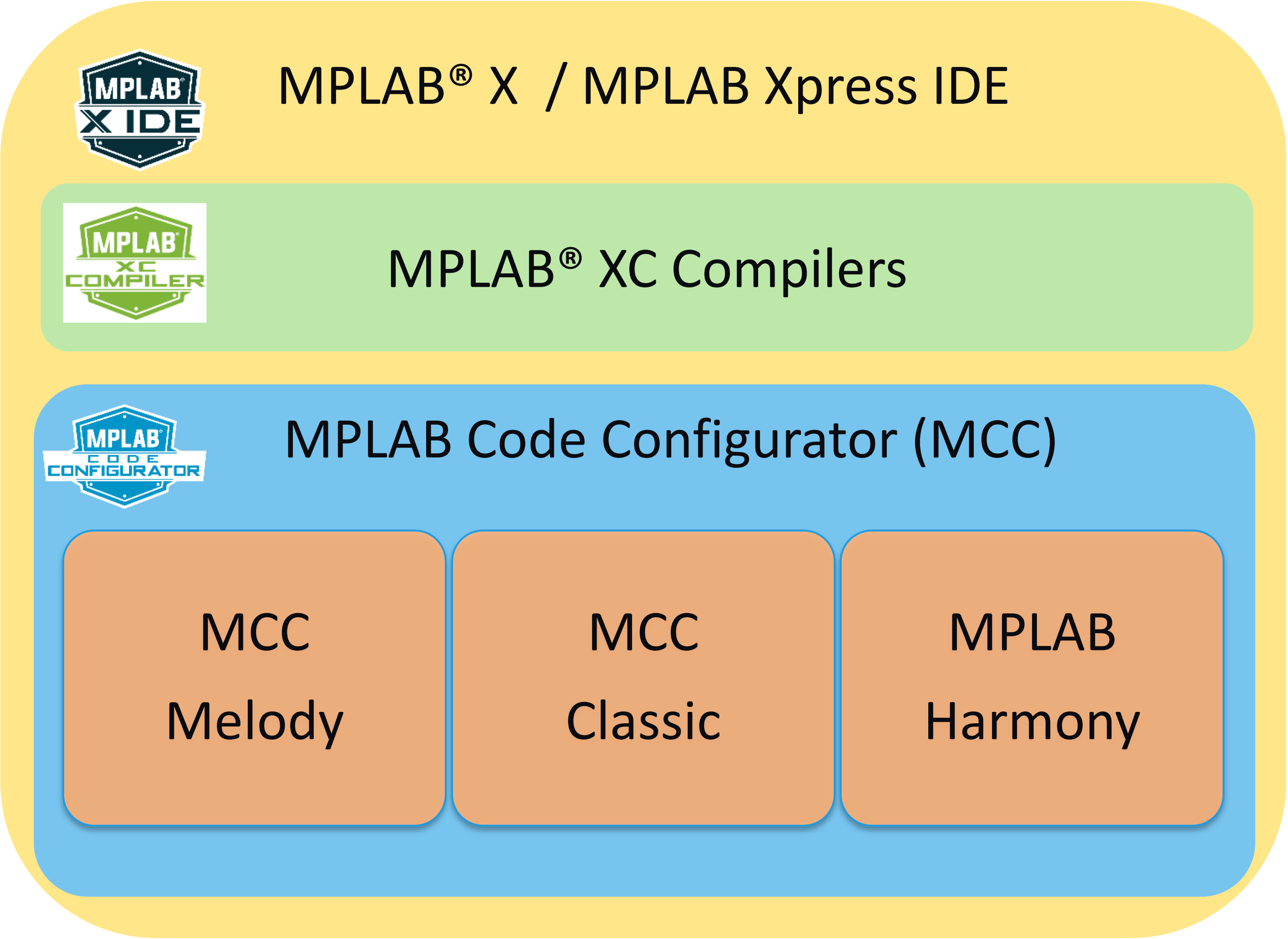
MICROCHIP

Microchip 開發工具

從首頁進入方式


Search | Microchip Technology
Tools and Resources | Microchip
microchip.com/en-us/tools-resources
Log in to myMicrochip to access tools and benefits. Sign up in just one minute.
MICROCHIP
All
Enter keyword, item, model or part #
My Account
$0.00
1
2
PRODUCTS
SOLUTIONS
TOOLS AND RESOURCES
SUPPORT
EDUCATION
ABOUT
ORDER NOW
Browse All Tools and Resources
Reference Designs
Search and Discover
Configure
Develop
Debug
Evaluation Boards
Qualify
Production
Documentation
Archives
Last Chance Deals
s and Software
dded design with our easy-to-use portfolio of hardware and software development tools. You can discover, configure, develop, debug,
evelopment tools for PIC®, AVR® and SAM microcontrollers (MCUs), SAM microprocessors (MPUs) and dsPIC® Digital Signal Controllers
embedded software development tools with our embedded hardware development tools so you can harness the full capabilities of the
est-in-class experience.
FIGURE
DEVELOP
IDEs, Compilers
Example Code
Software Stacks
DEBUG
Evaluation Boards
Debuggers
Data Visualizer
QUALIFY
Functional Safety
Code Profiling
Code Coverage
MISRA Check
PRODUCTION
Programmers
Prog. Center
Third Party
IDEs
Feedback
https://www.microchip.com/en-us/tools-resources/all

應用軟體架構



MPLAB® X / MPLAB Xpress IDE

MPLAB® XC Compilers

MPLAB Code Configurator (MCC)

MCC Melody

MCC Classic

MPLAB Harmony

IDE 用於產生專案及除錯

MCC 透過圖形化界面規劃後，產生周邊硬體驅動程式及程式碼到IDE中

XC 根據所載入的程式碼編譯成HEX

各項目錄

支援軟硬體規劃設定，並根據用戶規劃產生程式碼

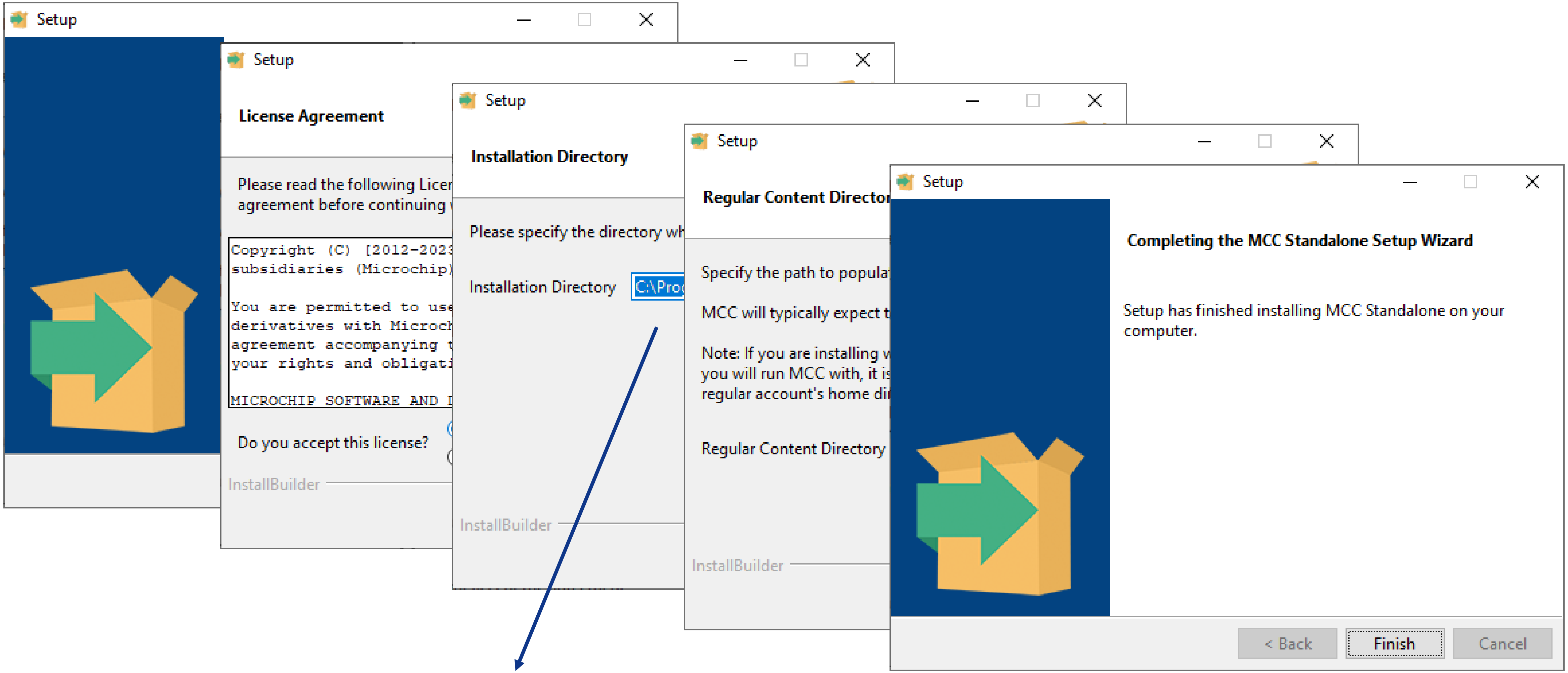
課程使用軟體

- **MPLAB X IDE v6.20** https://www.microchip.com/mplab/mplab-x-ide
- **MPLAB XC16 v2.10** https://www.microchip.com/mplab/compilers
- 在IDE介面中下載
  - **MPLAB® MCC v5.5.0**

    https://ww1.microchip.com/downloads/aemDocuments/documents/DEV/ProductDocuments/SupportingCollateral/mcc-installer-5.5.0-windows.exe
  - **MPLAB® MCC Melody Libraries**
  - **dsPIC33CK-MP DFP v1.11.346. (Device Family Pack)**

舊版軟體下載連結 https://www.microchip.com/en-us/tools-resources/archives/mplab-ecosystem

MCC 離線版安裝



Setup

Setup

License Agreement

Please read the following Licen
agreement before continuing w

Copyright (C) [2012-2023
subsidiaries (Microchip)

You are permitted to use
derivatives with Microch
agreement accompanying t
your rights and obligati

MICROCHIP SOFTWARE AND I

Do you accept this license?

InstallBuilder

Setup

Installation Directory

Please specify the directory wh

Installation Directory C:\Prog

InstallBuilder

Setup

Regular Content Director

Specify the path to popula

MCC will typically expect t

Note: If you are installing w
you will run MCC with, it is
regular account's home dir

Regular Content Directory

InstallBuilder

Setup

Completing the MCC Standalone Setup Wizard

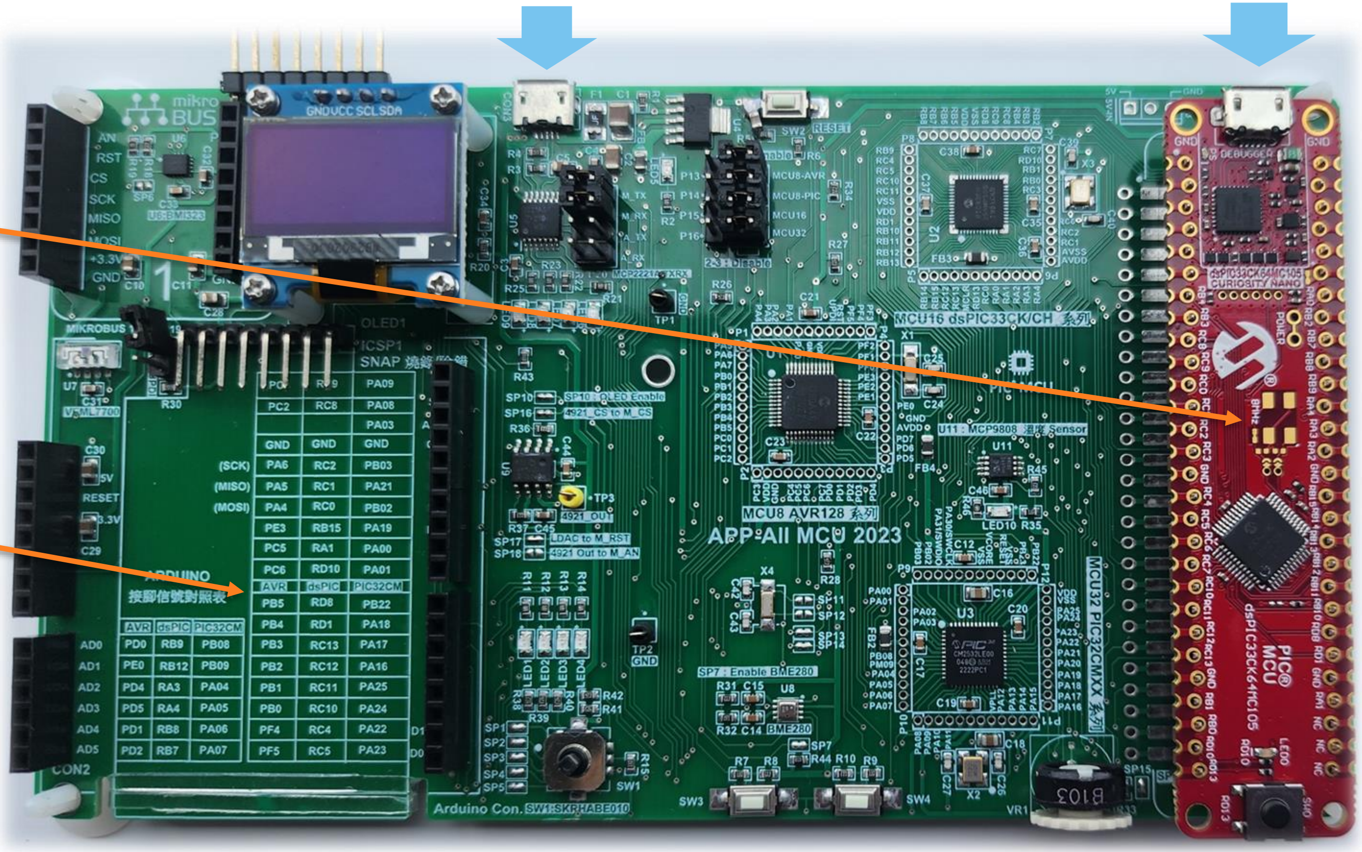
Setup has finished installing MCC Standalone on your computer.

< Back   Finish   Cancel

- C:\Program Files\Microchip\mcc\MccStandalone-5.5.0

課程使用實驗板

- dsPIC33CK Curiosity Nano
- APP-All MCU 2023



USB to I2C/UART

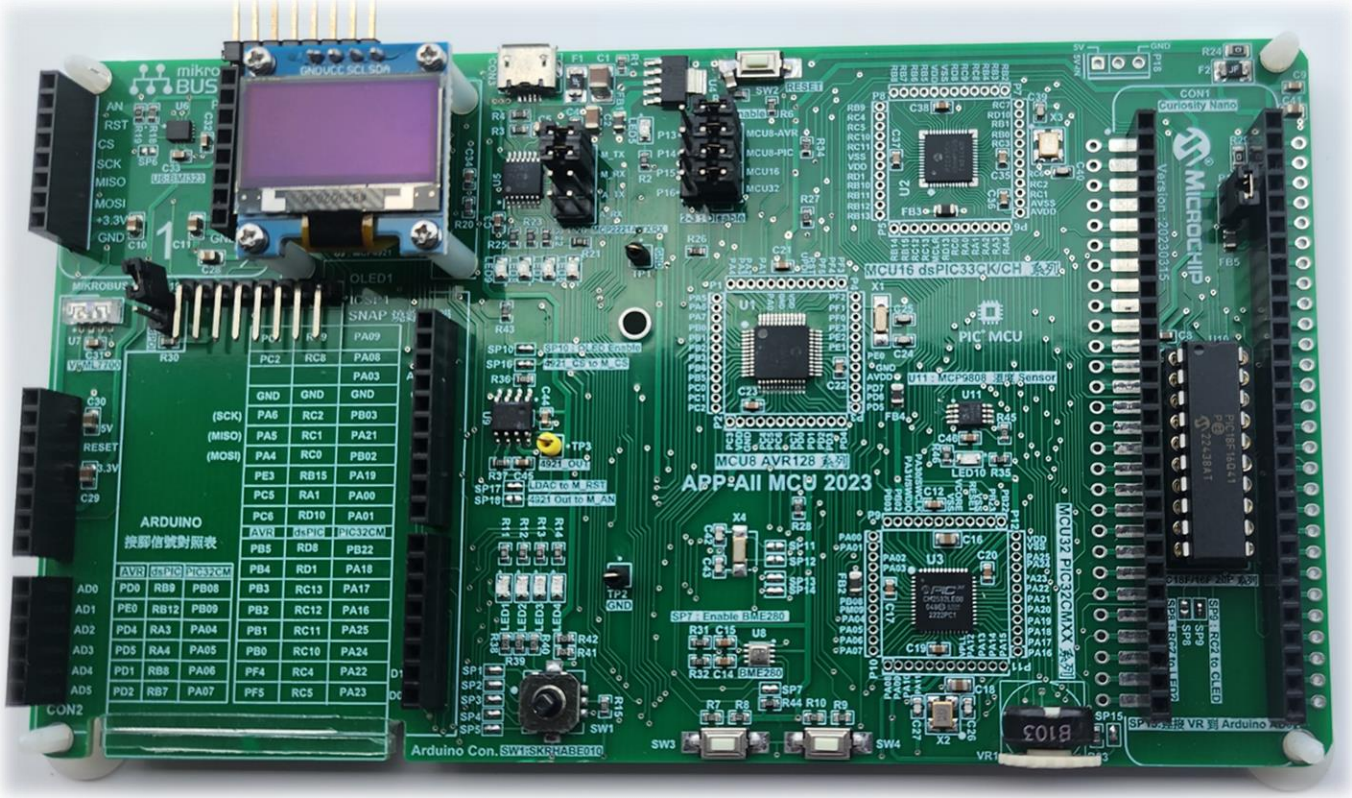
Program/Debug/UART

dsPIC33CK Curiosity Nano特色

- 內建Debugger
  - 板子資訊已內建到MPLAB X IDE中
  - 支援燒錄及除錯
  - 電源/狀態指示燈(綠色)
  - 一個邏輯分析腳 (DGI GPIO)
  - 虛擬COM port (CDC)
- USB 供電
  - 電源可調
  - MIC5353 LDO由Debugger控制
  - 1.8-3.6v 輸出電壓
  - 最大輸出電流500 mA (受環境溫度和輸出電壓限制)
- 使用dsPIC33CK64MC105
  - The High Performance 100 MHz
  - 64KB of ECC Flash and 8KB of RAM
- 一個LED (黃色)
- 一個按鍵

APP-All MCU 2023 內建的周邊

- I2C 介面的
  - 六軸 IMU - BOSCH BMI323
  - Lighting Sensor – Vishay 的VEML7700-TT
  - Humidity sensor - BOSCH BME280
  - 溫度 Sensor – Microchip MCP9808
  - OLED Display - 單色128 * 64
- SPI 介面的
  - DAC – Microchip MCP4921
- 2個 WS2812B One-Wire Color LED
- 1個 MCP2221A 作實驗板上的UART 以及 I2C 介面轉換至 USB 的介面 IC
- 1個 ALPS 的SKRHABE010 五向開關



實驗一

基礎I/O控制

入門實驗

實驗一：基礎I/O控制

- 本實驗目的：
  - 確認開發環境設定完成
    - 驗證電腦安裝的軟體X IDE,XC,MCC正常運作
    - 紅板上內建的燒錄及除錯可正常運行
    - MCU可以受所撰寫的程式控制IO

- 結果呈現：
  - 黃色LED 一秒閃一次

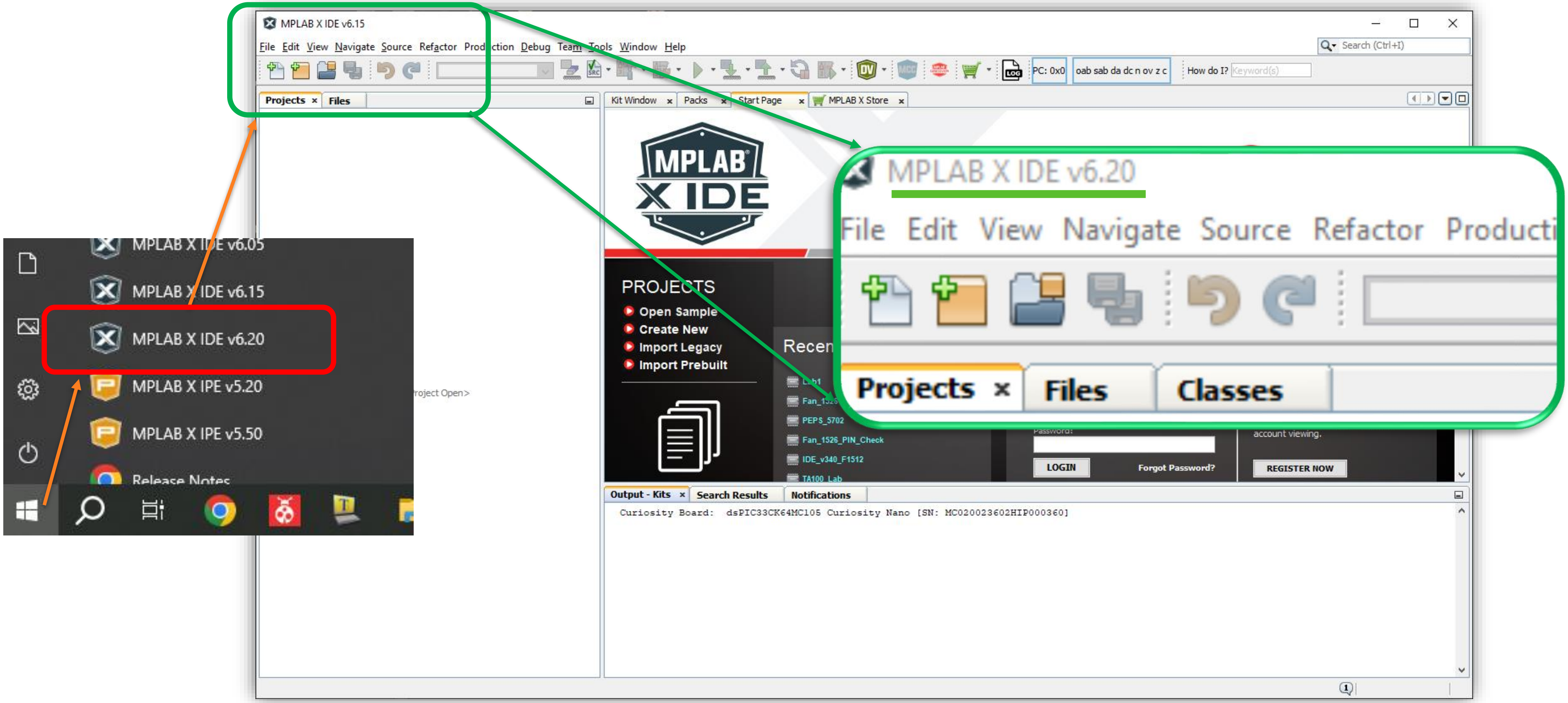
前置作業

確認軟體IDE、XC、MCC

確認硬體EVB、USB

1. 開啟MPLAB X IDE

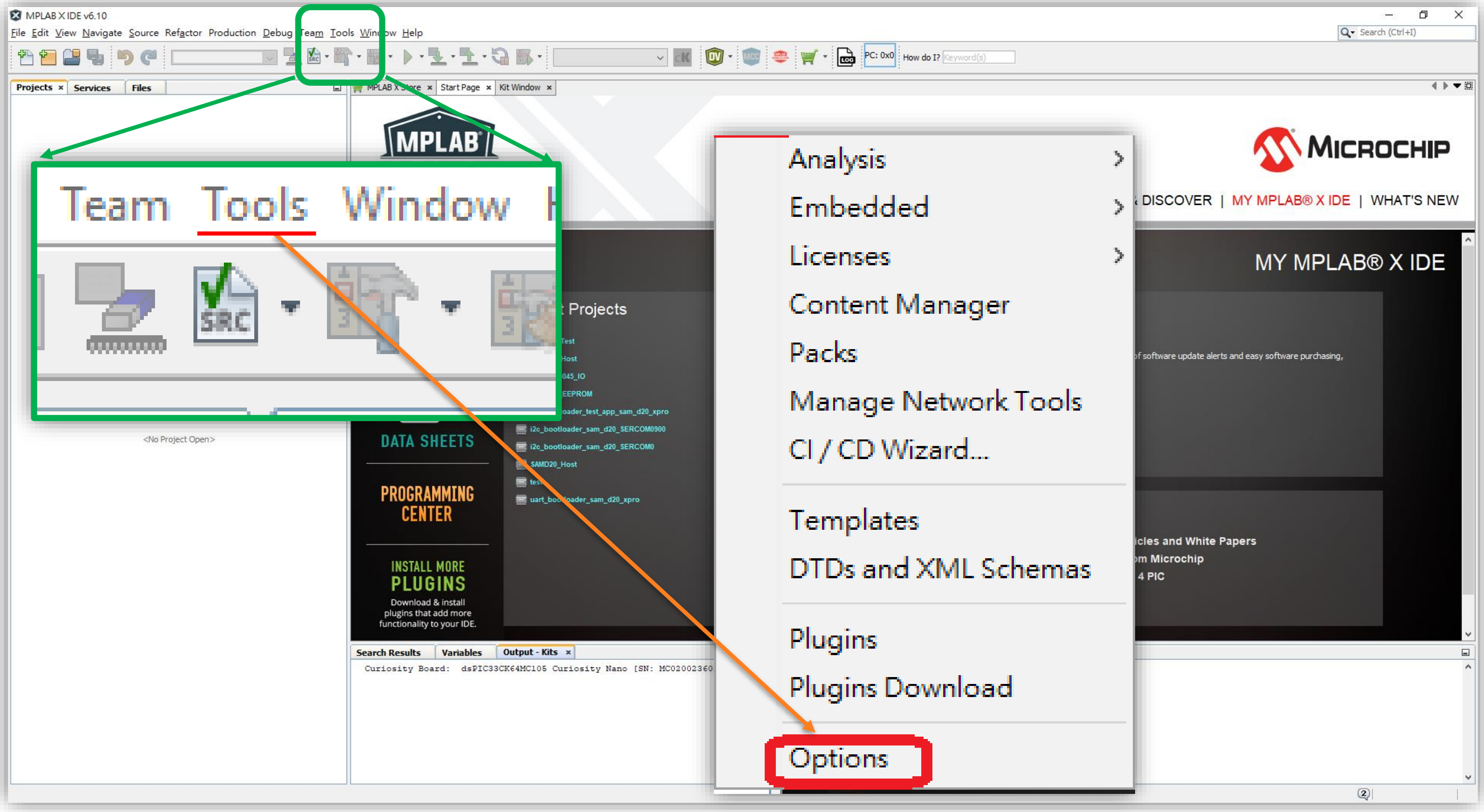
點選開始⊞ --> 找到MPLAB X IDE v6.20點選



```
Curiosity Board:  dsPIC33CK64MC105 Curiosity Nano [SN: MC020023602HIP000360]
```

2. 確認XC16已安裝

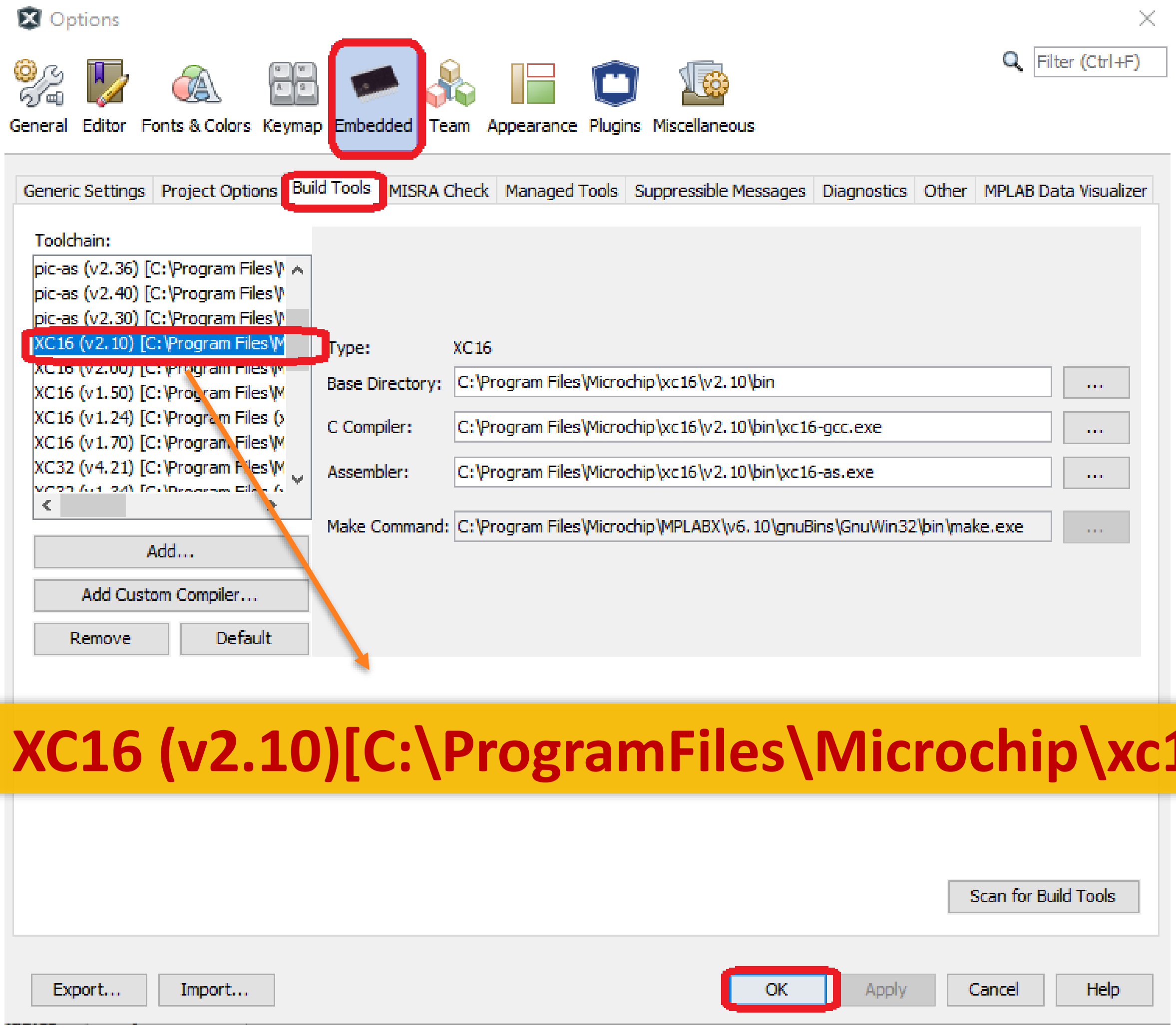
選Tools --> Options



```
Curiosity Board:  dsPIC33CK64MC105 Curiosity Nano [SN: MC02002360
```

2.1確認XC16已安裝

確認XC16 v2.1已安裝



Toolchain:

pic-as (v2.36) [C:\Program Files\
pic-as (v2.40) [C:\Program Files\
pic-as (v2.30) [C:\Program Files\
XC16 (v2.10) [C:\Program Files\M
XC16 (v1.50) [C:\Program Files\M
XC16 (v1.24) [C:\Program Files (x
XC16 (v1.70) [C:\Program Files\M
XC32 (v4.21) [C:\Program Files\M

Type: XC16

Base Directory: C:\Program Files\Microchip\xc16\v2.10\bin

C Compiler: C:\Program Files\Microchip\xc16\v2.10\bin\xc16-gcc.exe

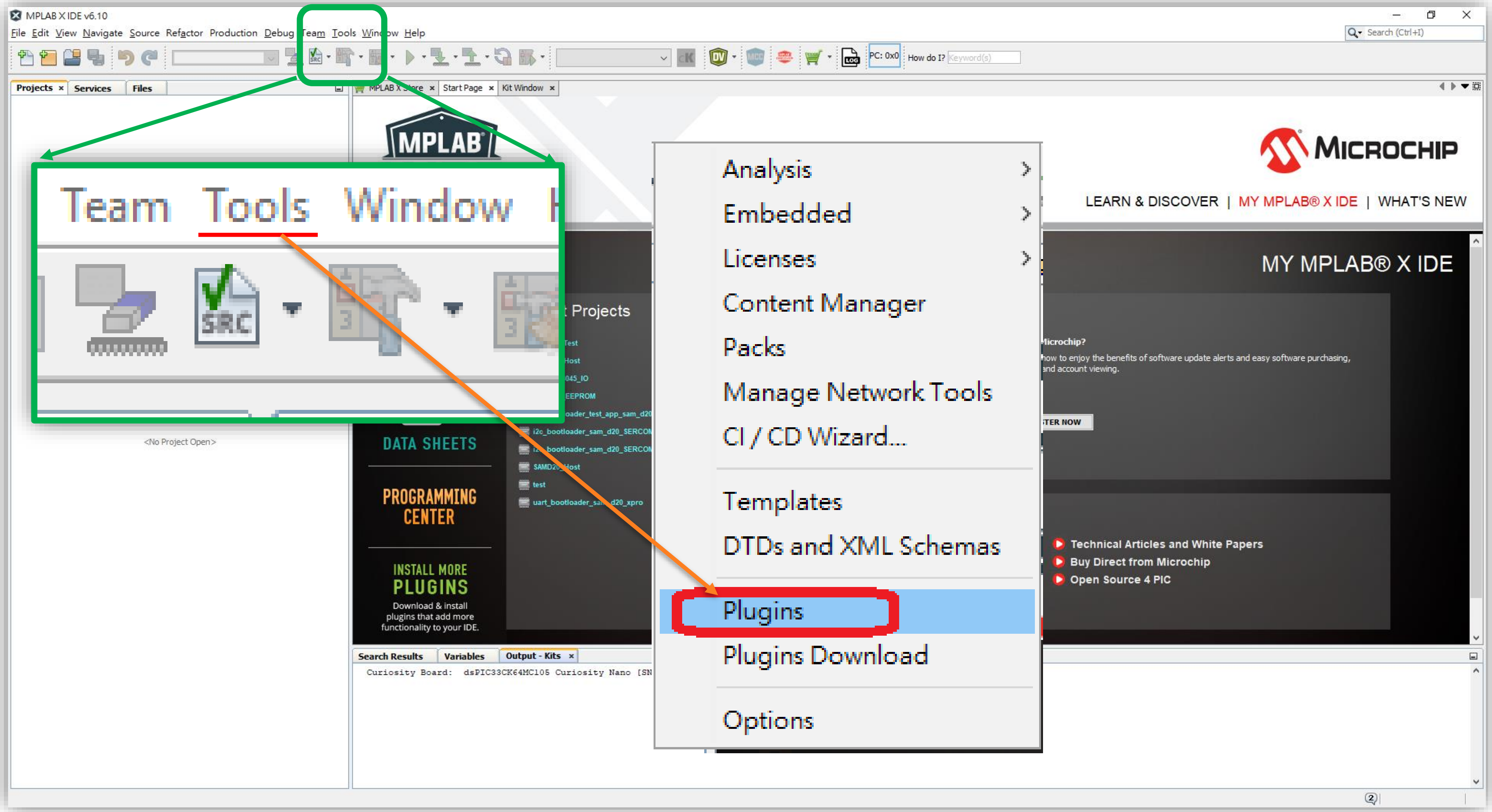
Assembler: C:\Program Files\Microchip\xc16\v2.10\bin\xc16-as.exe

Make Command: C:\Program Files\Microchip\MPLABX\v6.10\gnuBins\GnuWin32\bin\make.exe

確認有 XC16 (v2.10)[C:\ProgramFiles\Microchip\xc16\v2.10\bin

3. 確認MCC已安裝

選Tools --> Plugins

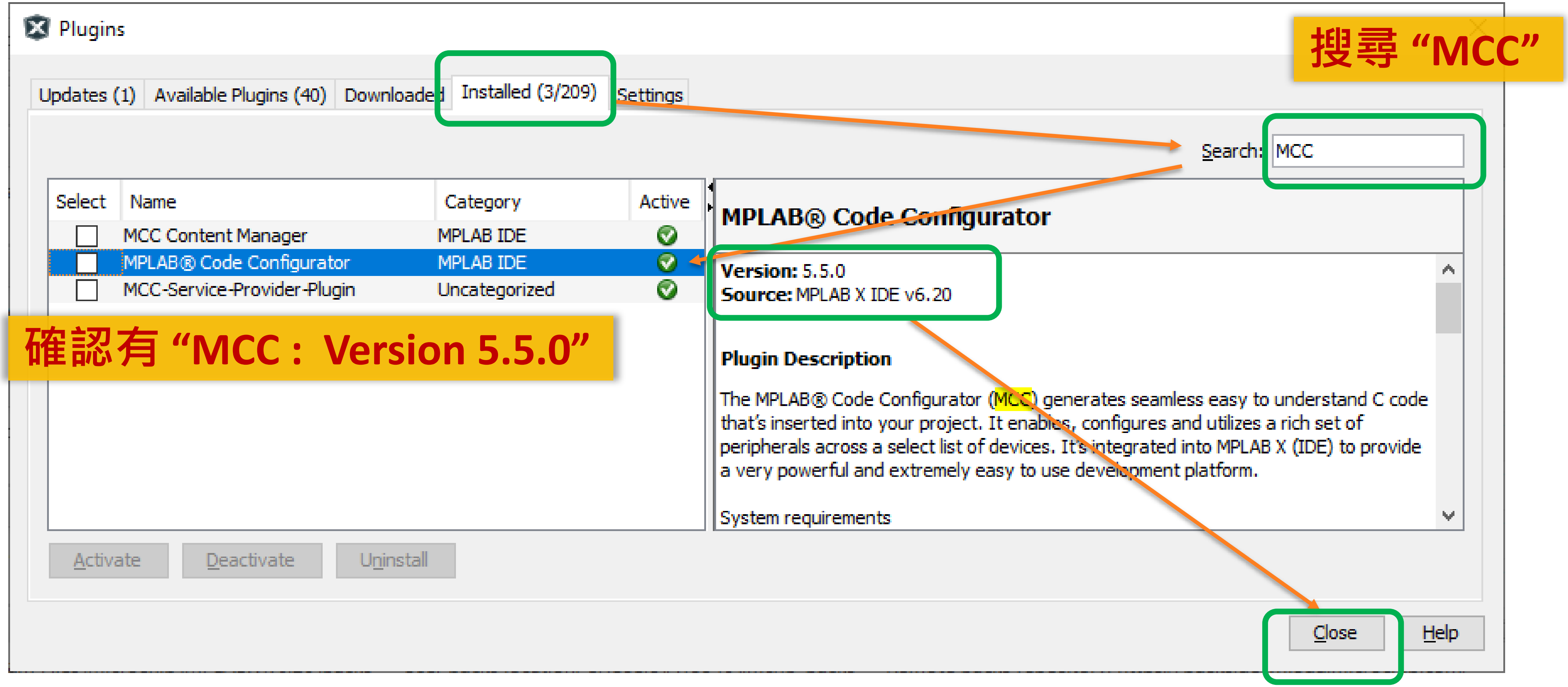


MPLAB X IDE v6.10
File Edit View Navigate Source Refactor Production Debug Team Tools Window Help
Team Tools Window
Projects Services Files
MPLAB X Store Start Page Kit Window
<No Project Open>
Analysis
Embedded
Licenses
Content Manager
Packs
Manage Network Tools
CI / CD Wizard...
Templates
DTDs and XML Schemas
Plugins
Plugins Download
Options
LEARN & DISCOVER | MY MPLAB® X IDE | WHAT'S NEW
MY MPLAB® X IDE
DATA SHEETS
PROGRAMMING CENTER
INSTALL MORE PLUGINS
Download & install plugins that add more functionality to your IDE.
Technical Articles and White Papers
Buy Direct from Microchip
Open Source 4 PIC
Search Results Variables Output - Kits

```
Curiosity Board:  dsPIC33CK64MC105 Curiosity Nano [SN
```

3.1確認MCC已安裝

確認MCC v5.3.7已啟動(Active)



Plugins

搜尋 “MCC”

Search: MCC

Select	Name	Category	Active
□	MCC Content Manager	MPLAB IDE	✓
□	MPLAB® Code Configurator	MPLAB IDE	✓
□	MCC-Service-Provider-Plugin	Uncategorized	✓

確認有 “MCC： Version 5.5.0”

MPLAB® Code Configurator

Version: 5.5.0
Source: MPLAB X IDE v6.20

Plugin Description

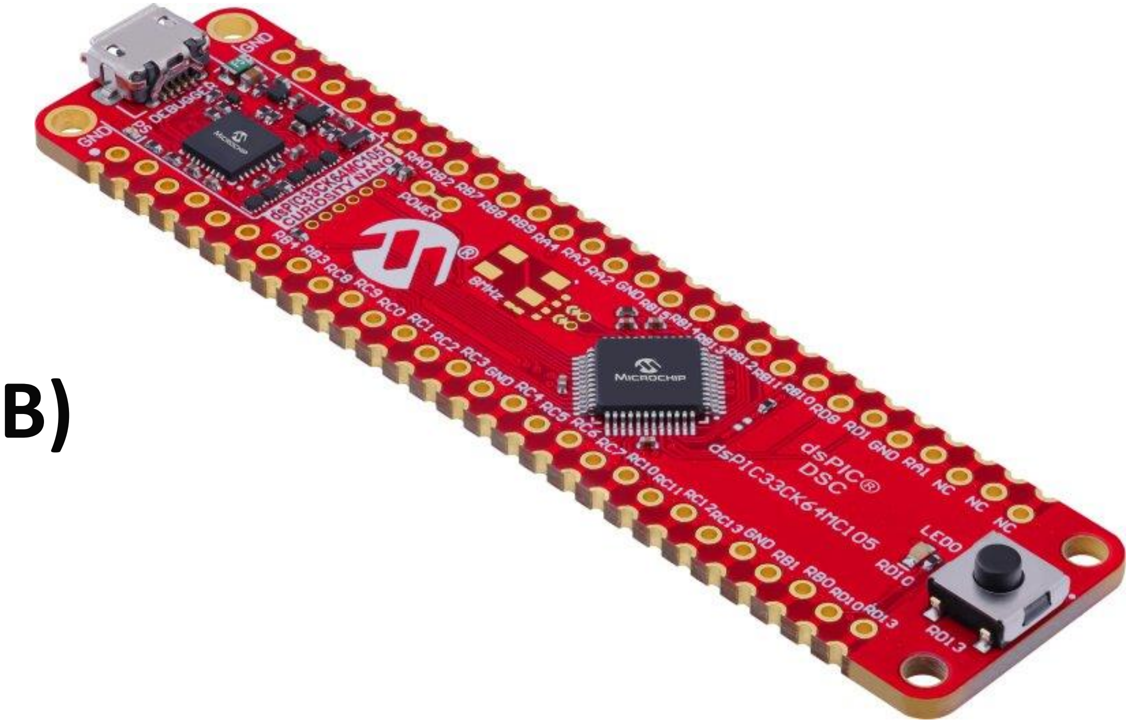
The MPLAB® Code Configurator (MCC) generates seamless easy to understand C code that's inserted into your project. It enables, configures and utilizes a rich set of peripherals across a select list of devices. It's integrated into MPLAB X (IDE) to provide a very powerful and extremely easy to use development platform.

System requirements

4.確認硬體

入門實驗，一塊板子、一條USB線

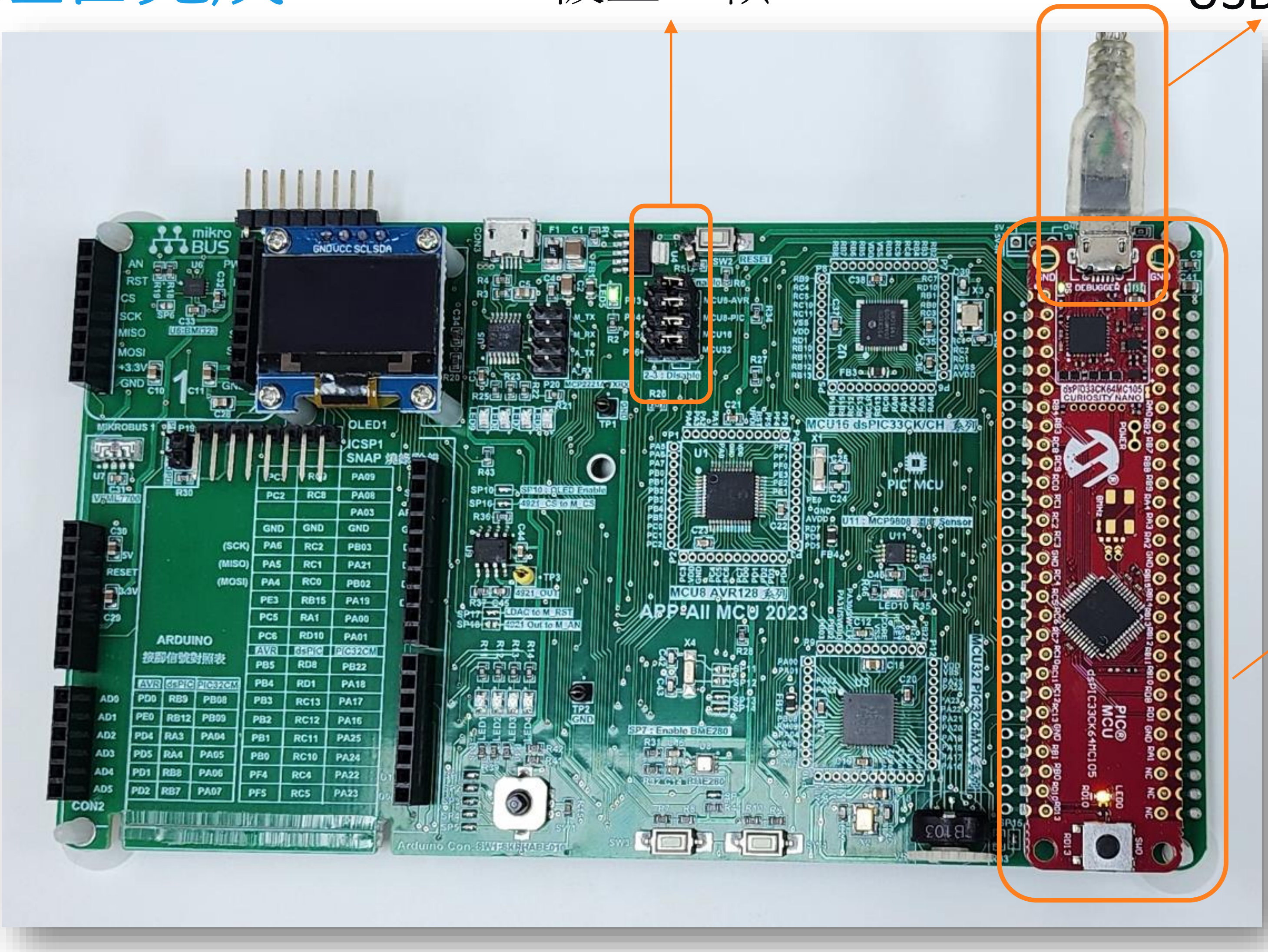
- **dsPIC33CK Curiosity Nano**



- **1條Micro USB (新版是Type C USB)**

4.1 確認硬體

周邊實驗，組合完成



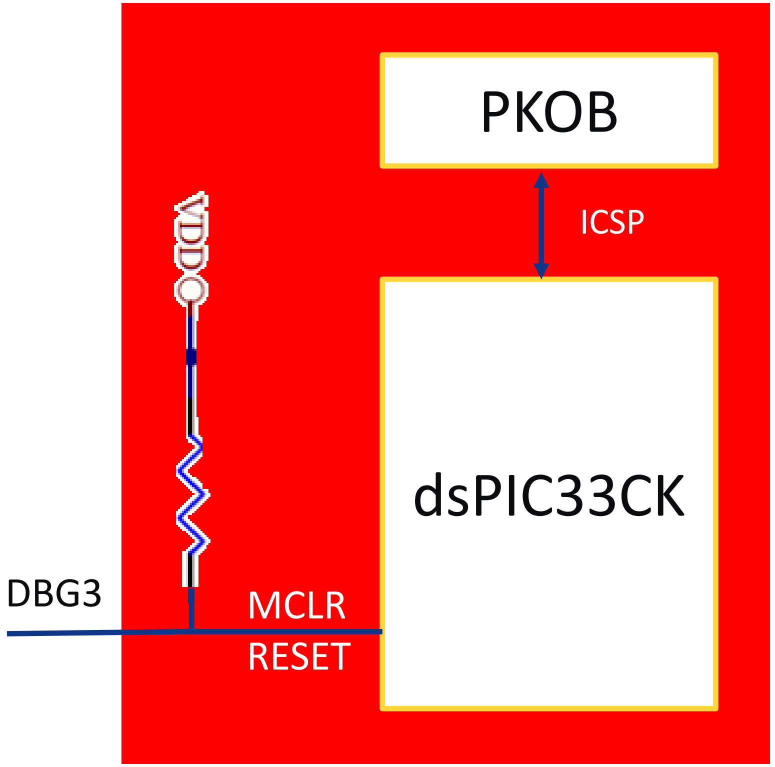
RESET板上四顆MCU

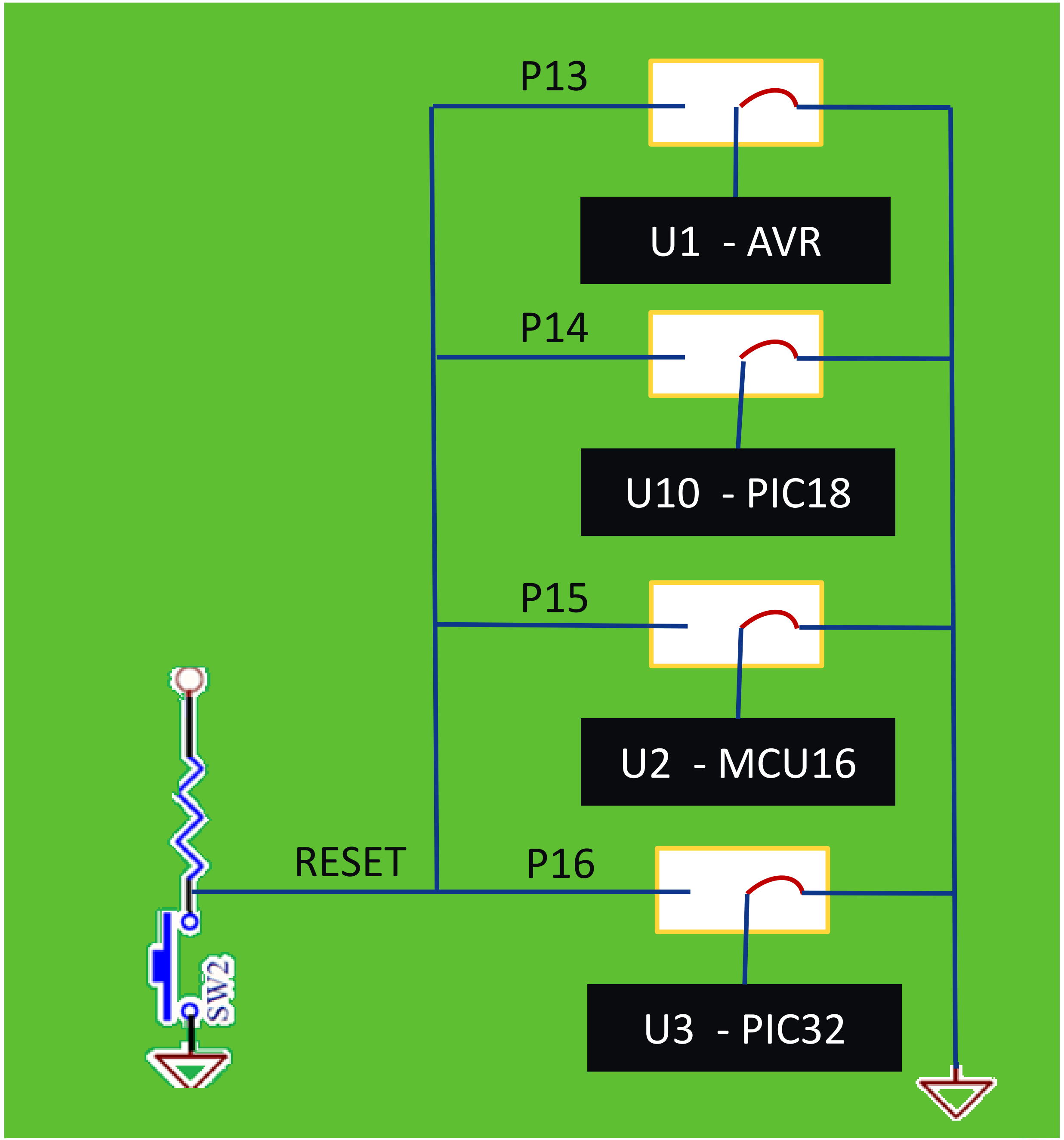
USB接到電腦

接到載板上→
連接板上的
LED等...

4.2 RESET架構圖

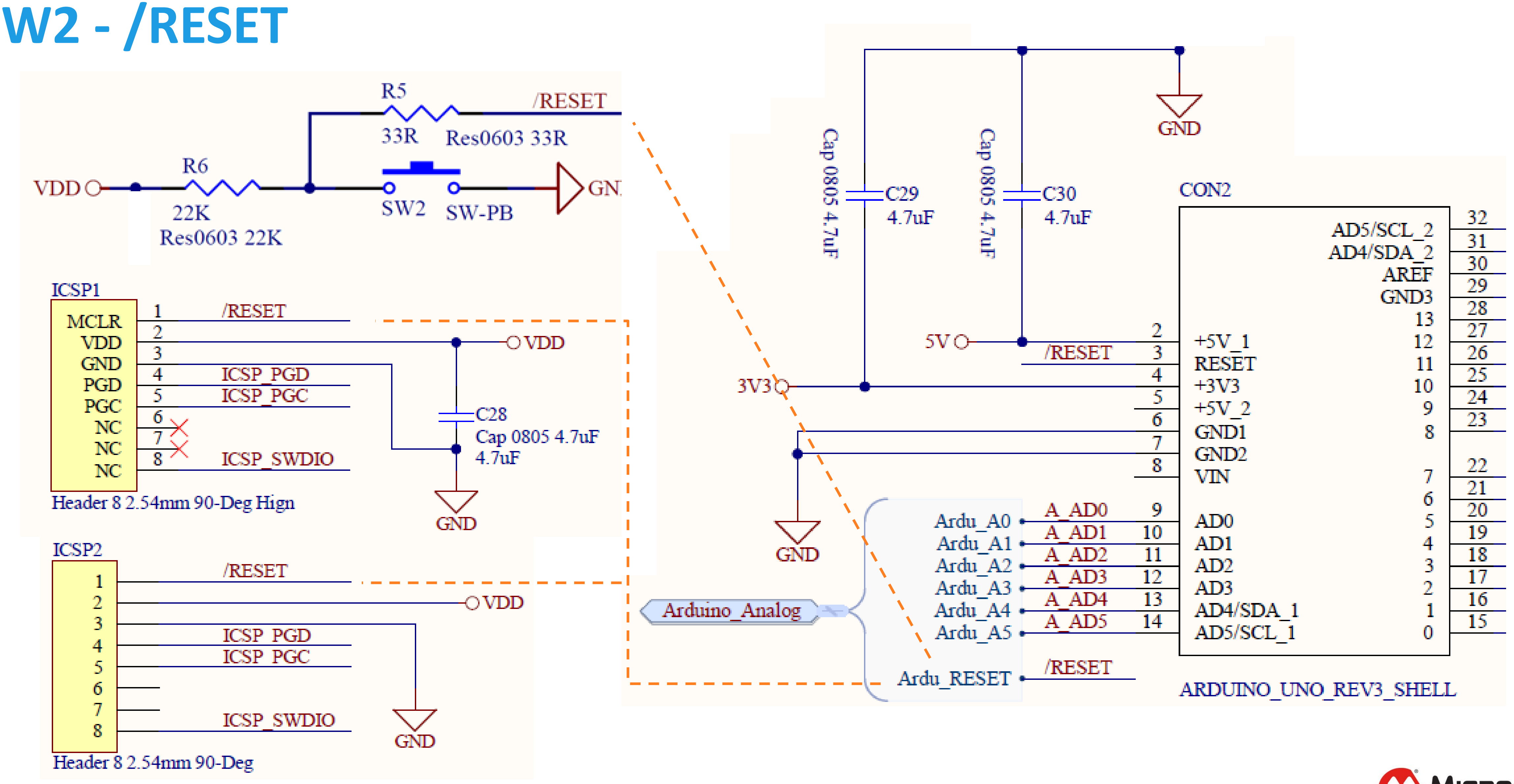
RESET 相關連線





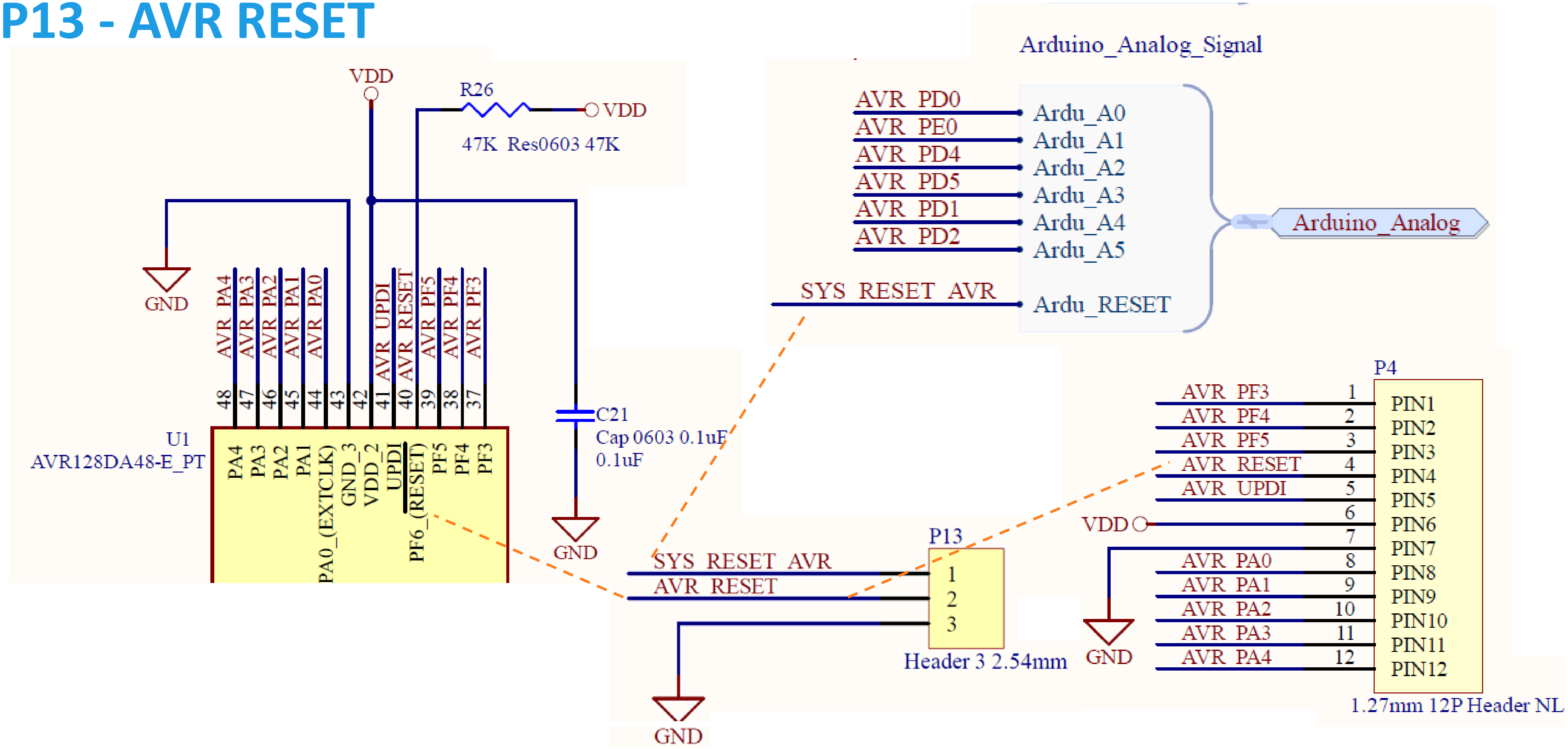
4.3 RESET相關電路

SW2 - /RESET



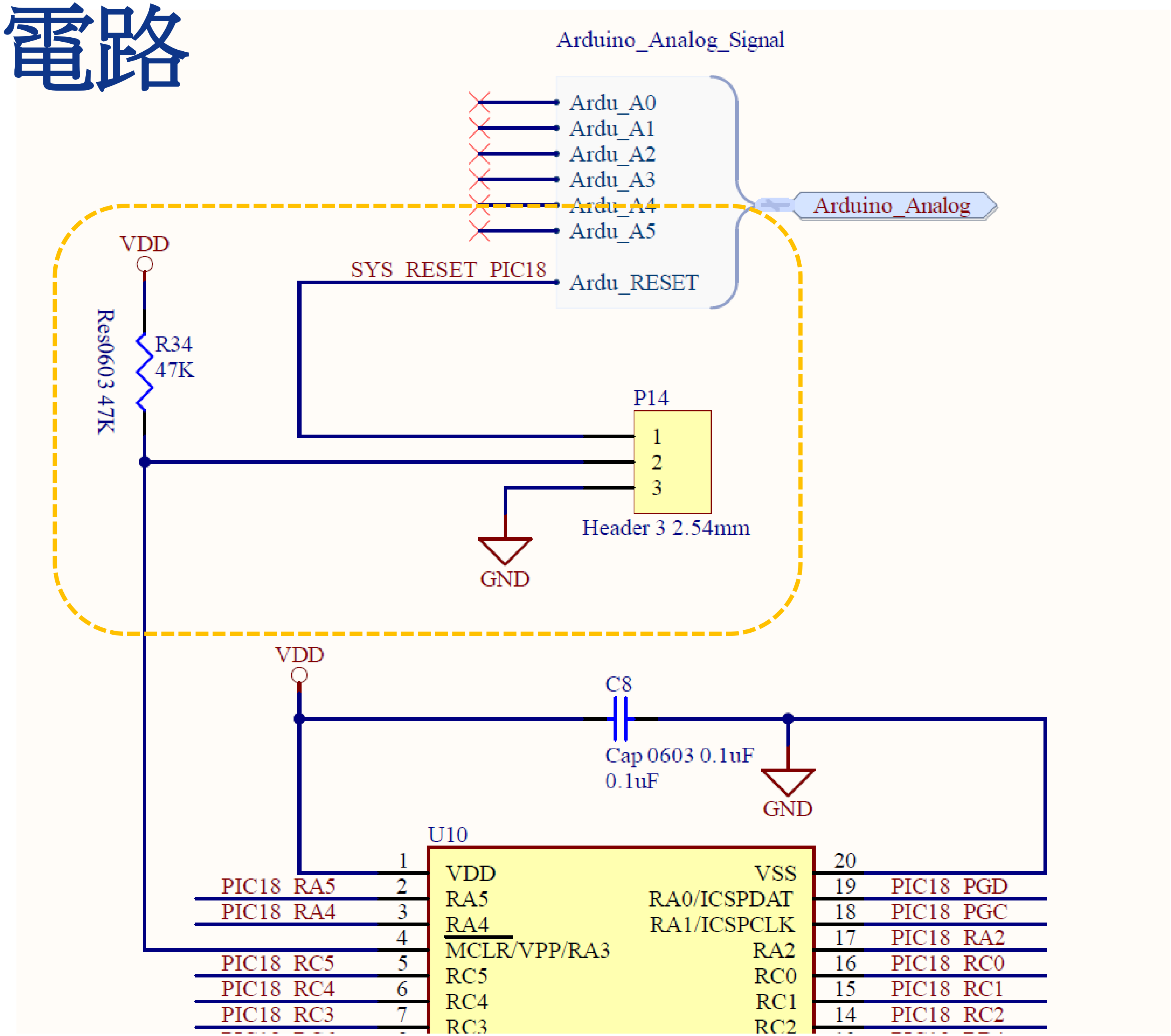
4.4 RESET相關電路

P13 - AVR RESET



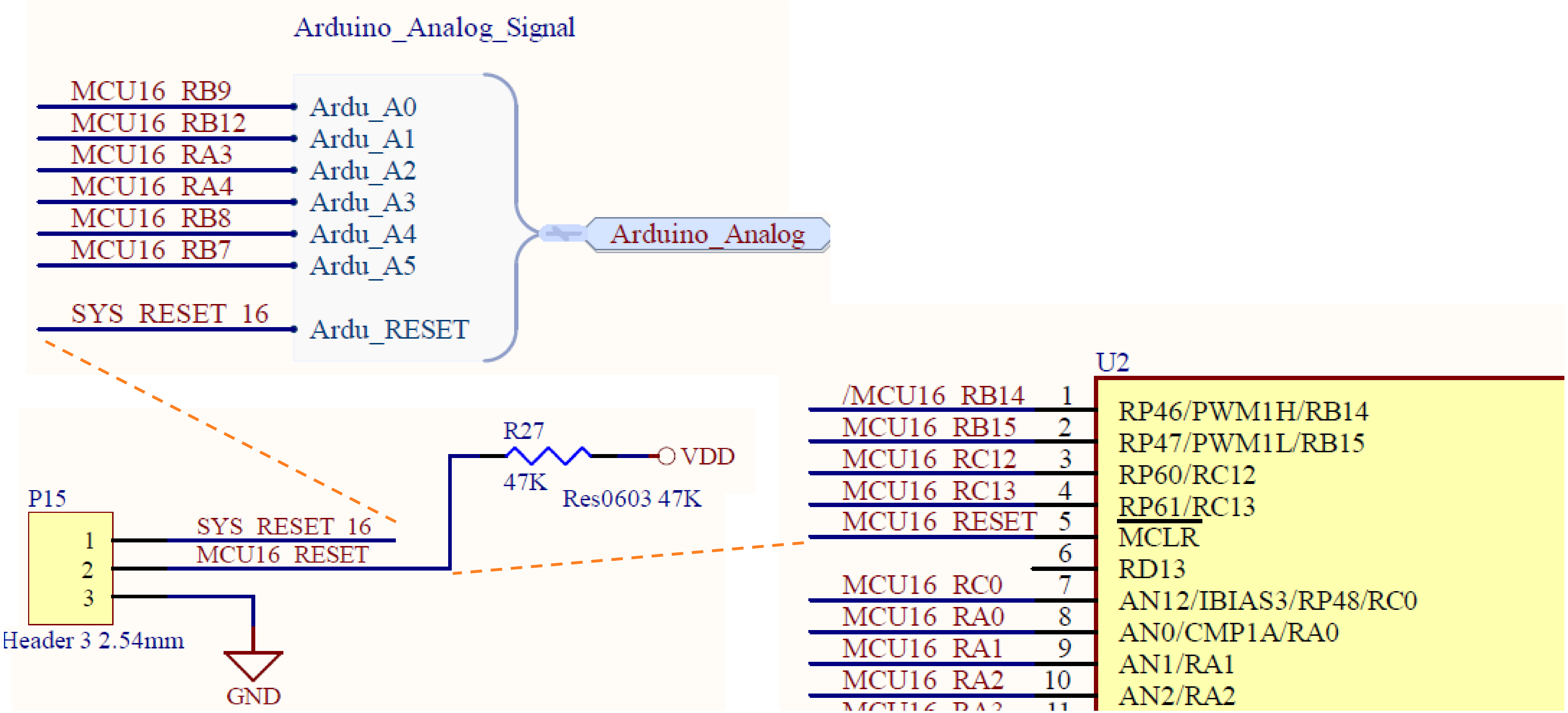
4.5 RESET相關電路

P14 - PIC18 RESET



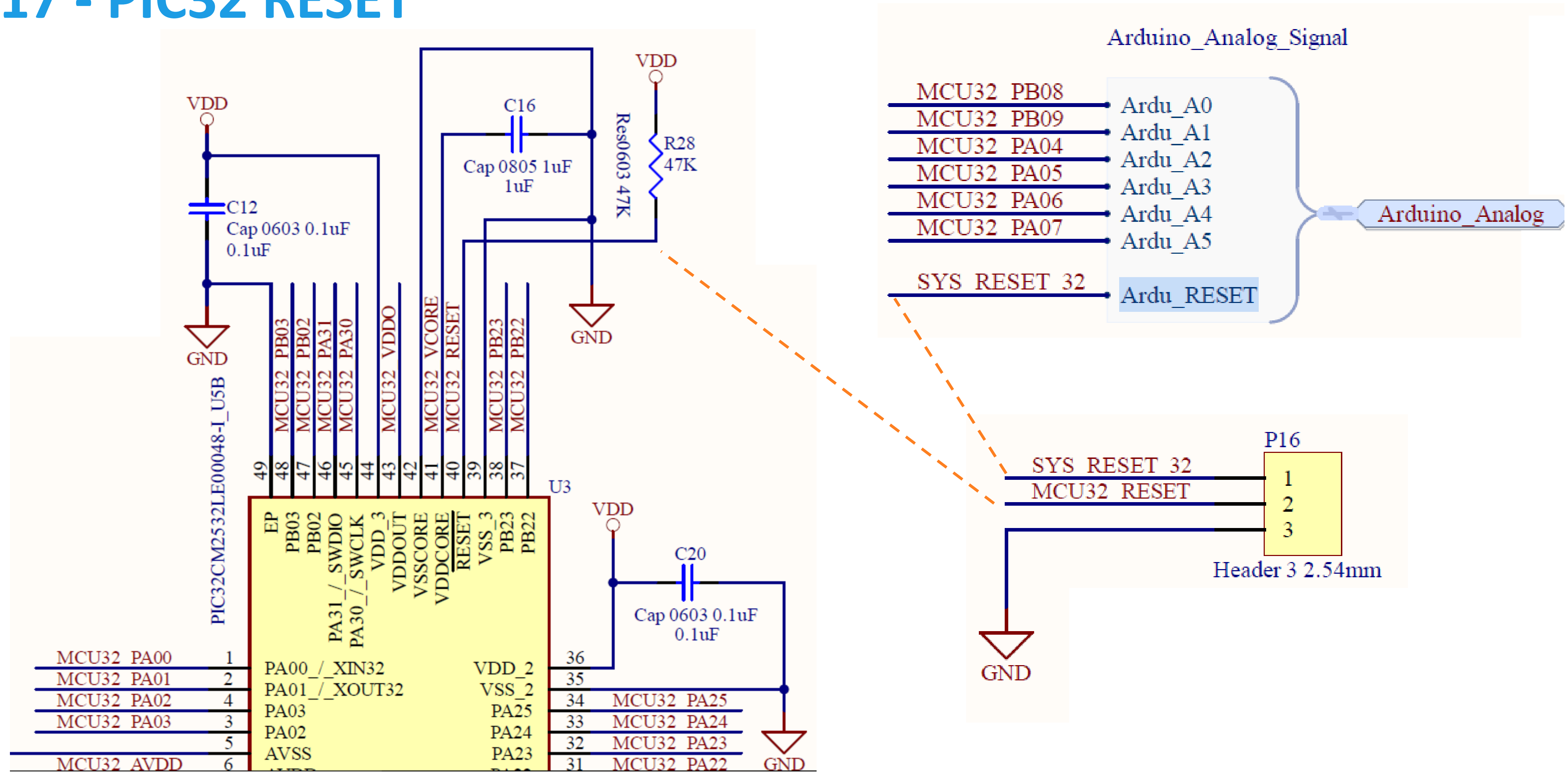
4.6 RESET相關電路

P15 – MCU16 RESET



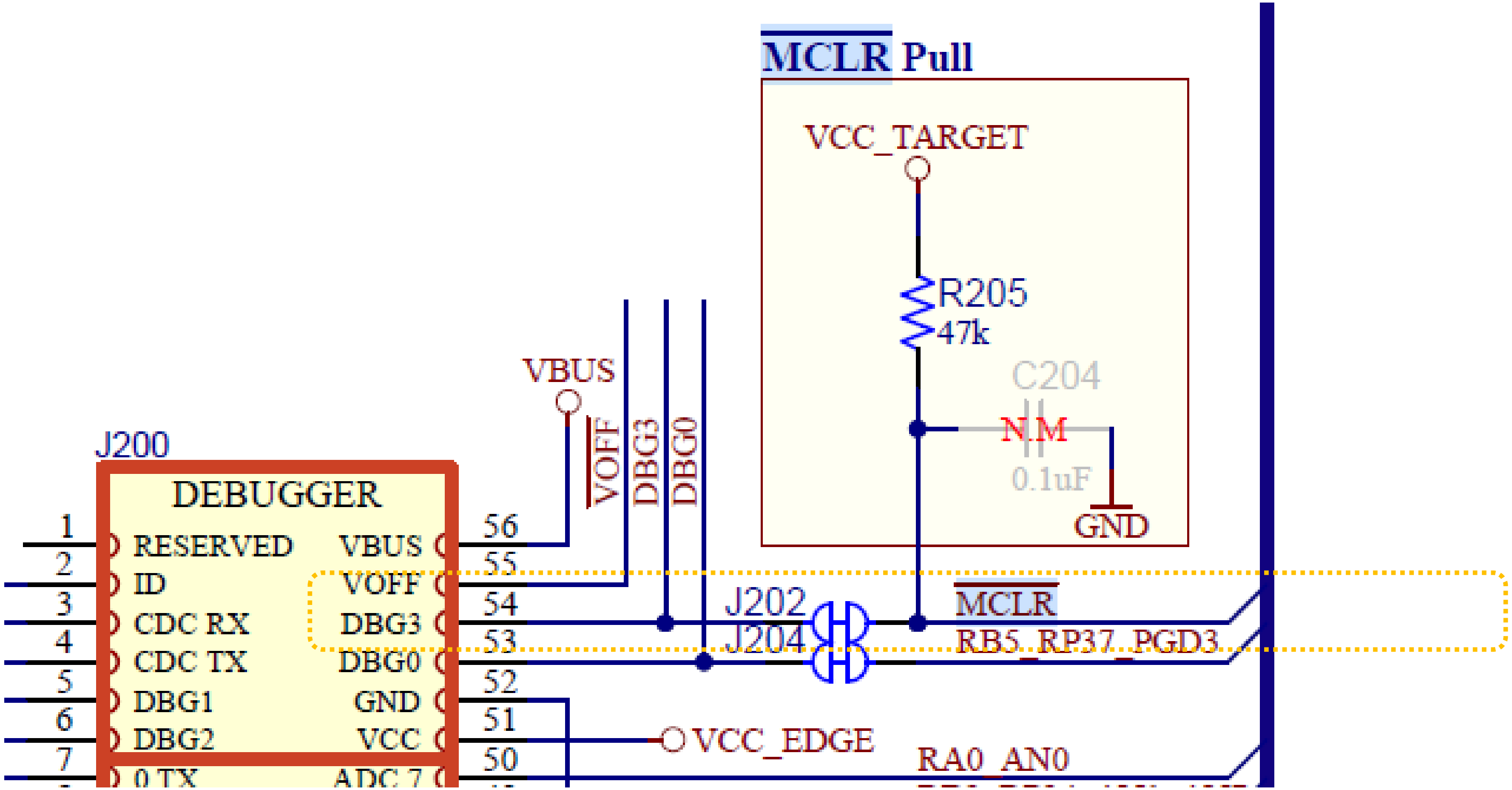
4.7 RESET相關電路

P17 - PIC32 RESET



4.8 RESET相關電路

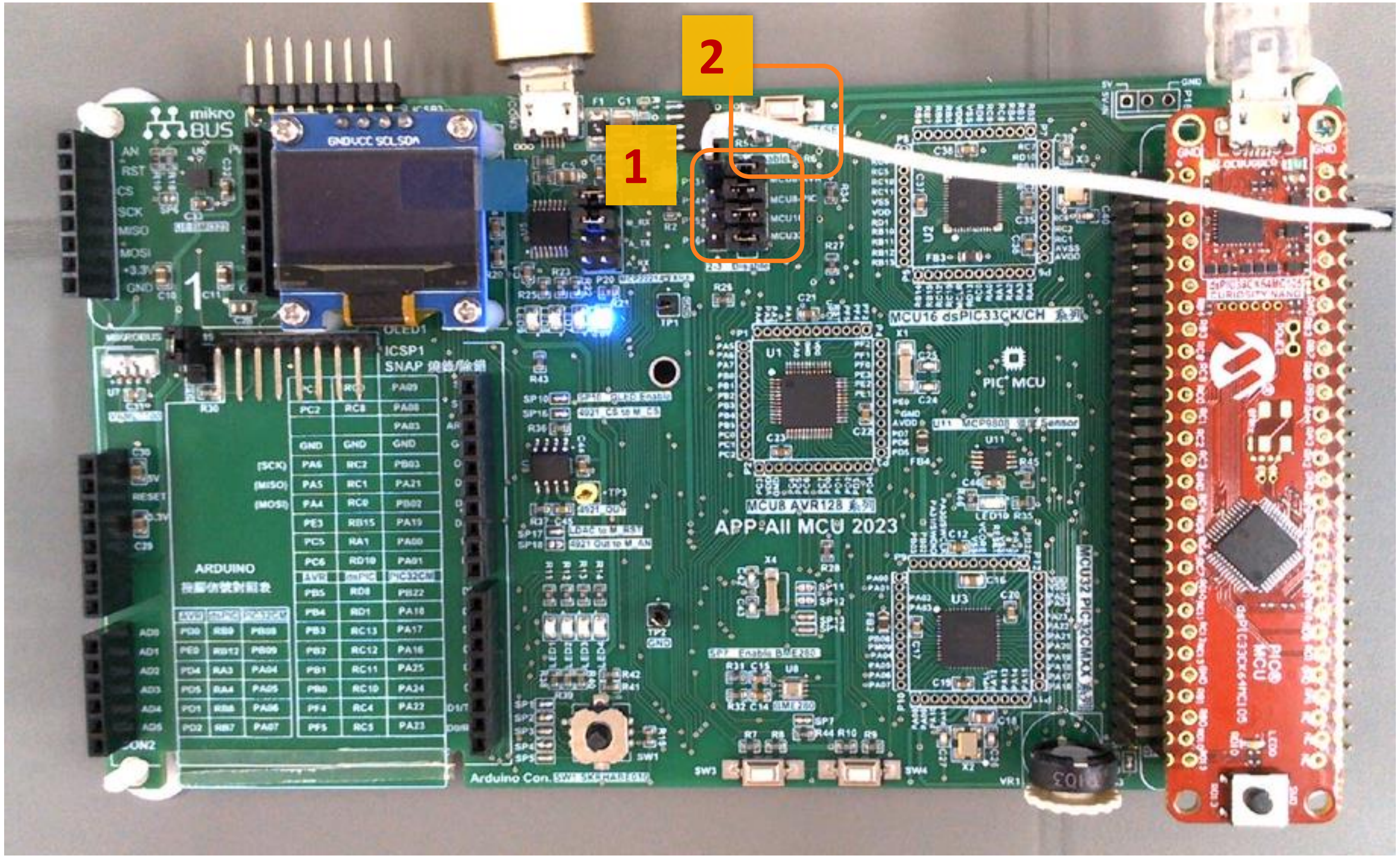
小板(dsPIC33CK) RESET



4.9 RESET相關線路

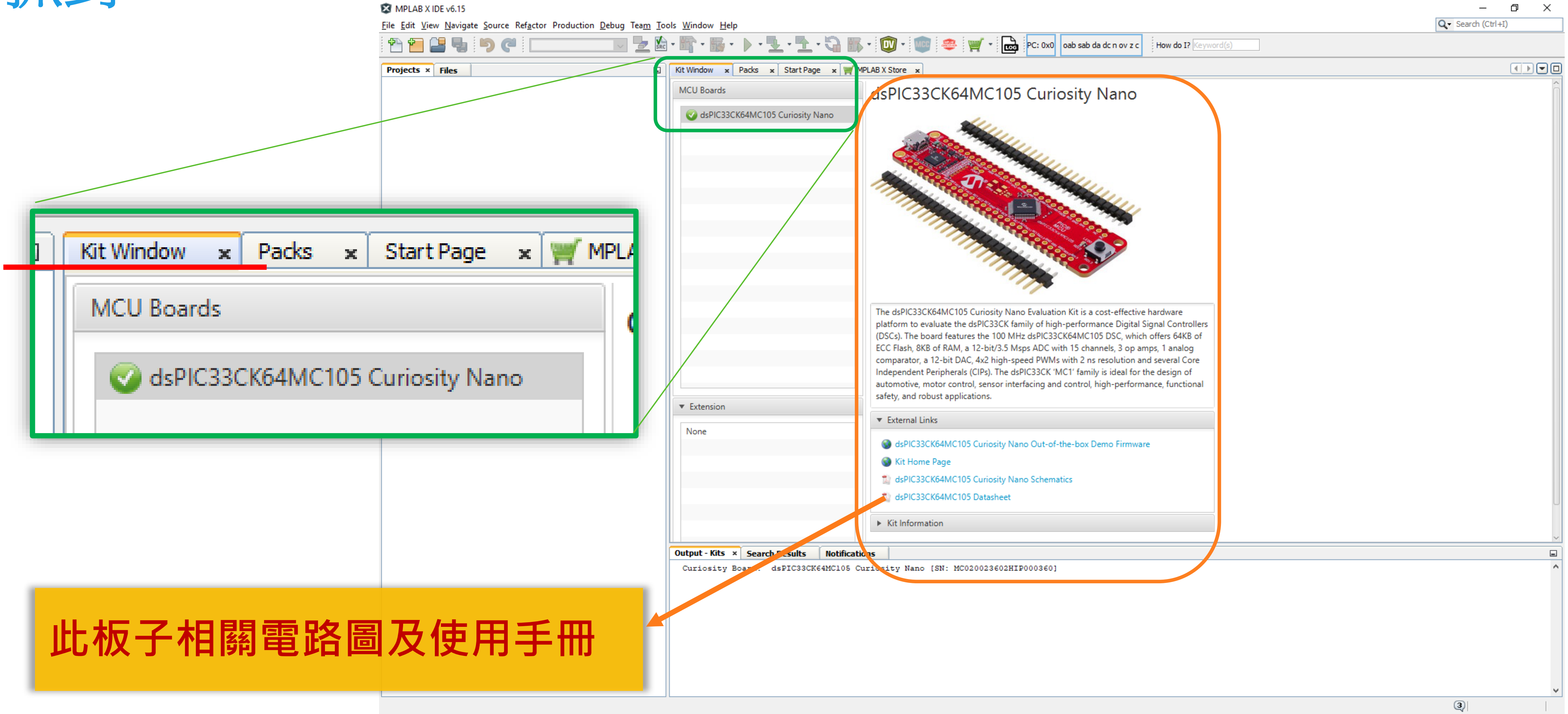
1. RESET板上四顆MCU P13~15 pin2-3短路

2. 接上白線讓SW2控制dsPIC33CK的 MCLR (可不接)



5.開啟MPLAB X IDE

抓到dsPIC33CKMC105 demo board



dsPIC33CK64MC105 Curiosity Nano

The dsPIC33CK64MC105 Curiosity Nano Evaluation Kit is a cost-effective hardware platform to evaluate the dsPIC33CK family of high-performance Digital Signal Controllers (DSCs). The board features the 100 MHz dsPIC33CK64MC105 DSC, which offers 64KB of ECC Flash, 8KB of RAM, a 12-bit/3.5 Msps ADC with 15 channels, 3 op amps, 1 analog comparator, a 12-bit DAC, 4x2 high-speed PWMs with 2 ns resolution and several Core Independent Peripherals (CIPs). The dsPIC33CK 'MC1' family is ideal for the design of automotive, motor control, sensor interfacing and control, high-performance, functional safety, and robust applications.

External Links

- dsPIC33CK64MC105 Curiosity Nano Out-of-the-box Demo Firmware
- Kit Home Page
- dsPIC33CK64MC105 Curiosity Nano Schematics
- dsPIC33CK64MC105 Datasheet

Kit Information

```
Curiosity Board: dsPIC33CK64MC105 Curiosity Nano [SN: MC020023602HIP000360]
```

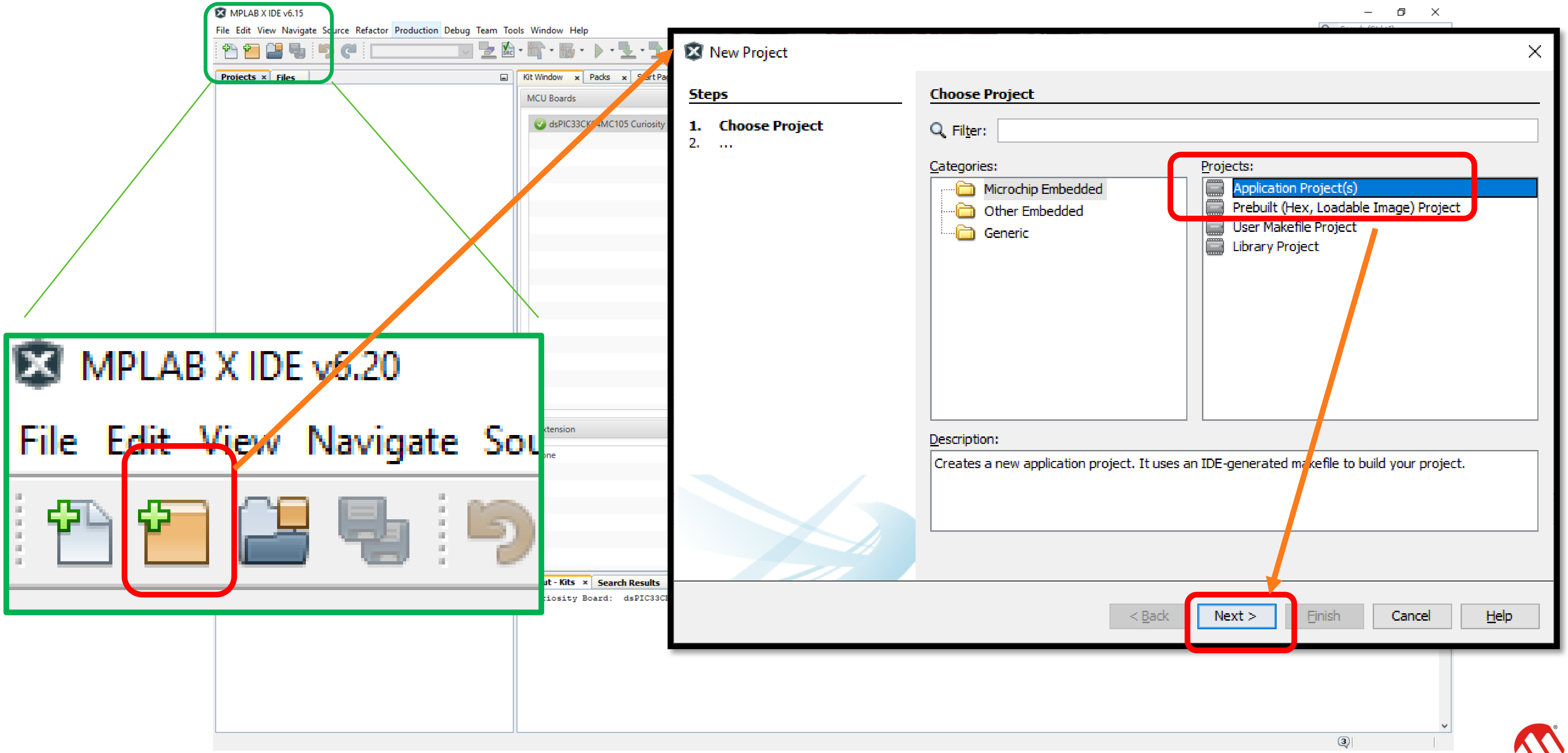
此板子相關電路圖及使用手冊

實驗步驟

實驗一

1.開一個新的專案

點選 啟動Project精靈，選Application Project(s)



New Project

Steps

1. Choose Project
2. ...

Choose Project

Filter:

Categories:
- Microchip Embedded
- Other Embedded
- Generic

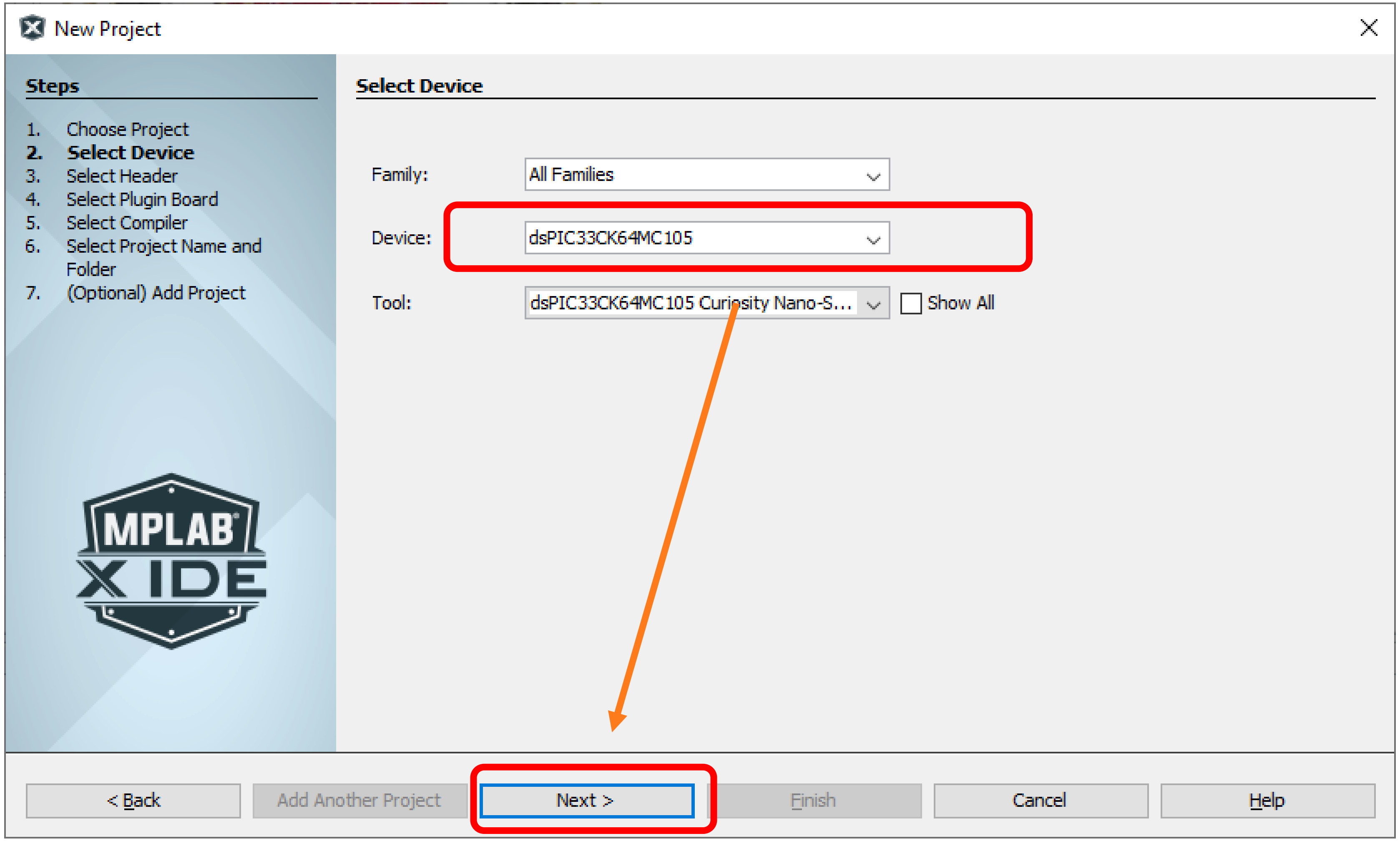
Projects:
- Application Project(s)
- Prebuilt (Hex, Loadable Image) Project
- User Makefile Project
- Library Project

Description:

Creates a new application project. It uses an IDE-generated makefile to build your project.

2. 選IC型號

dsPIC33CK64MC105



New Project

Steps

1. Choose Project
2. **Select Device**
3. Select Header
4. Select Plugin Board
5. Select Compiler
6. Select Project Name and Folder
7. (Optional) Add Project

Select Device

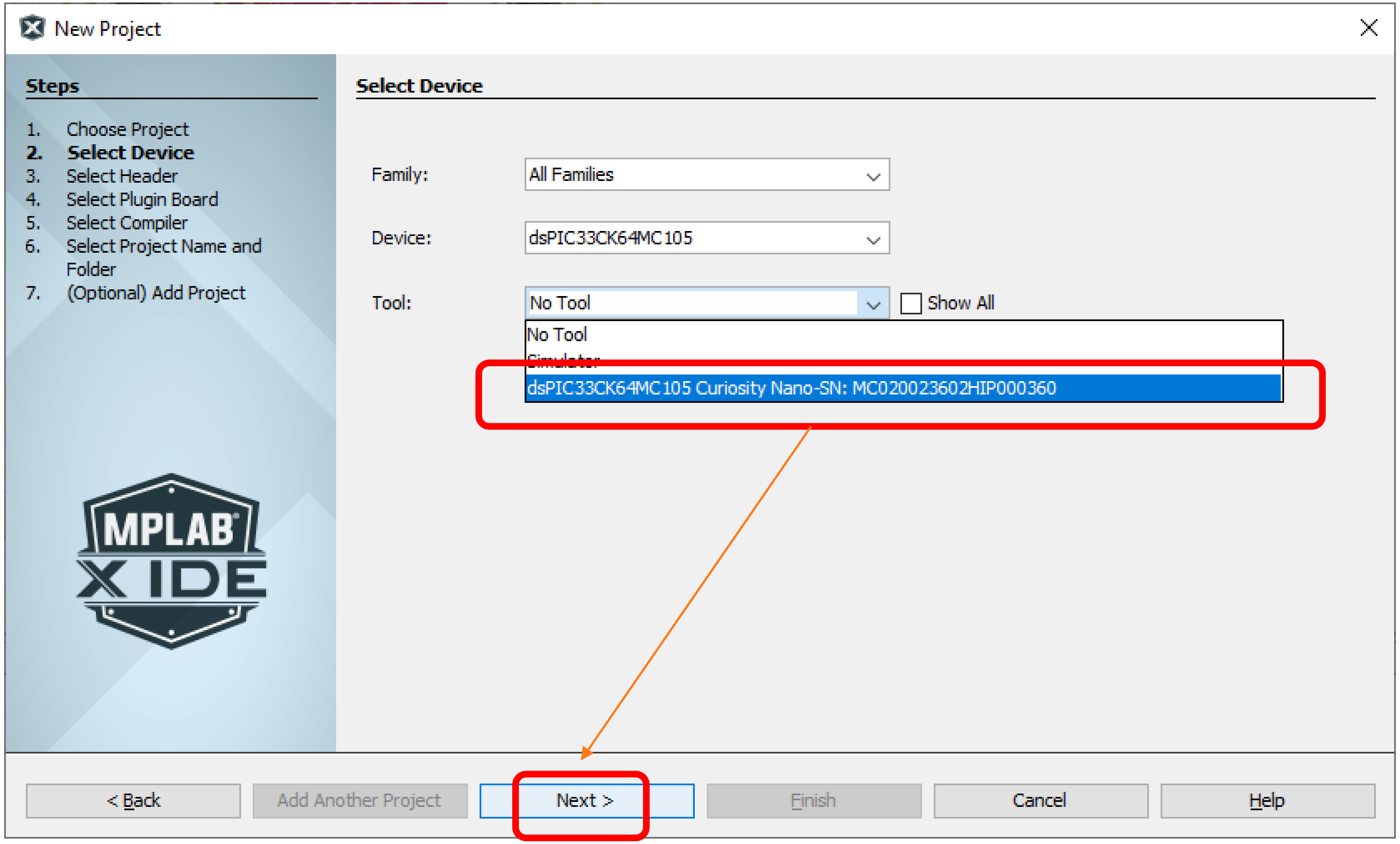
Family: All Families

Device: dsPIC33CK64MC105

Tool: dsPIC33CK64MC105 Curiosity Nano-S... □ Show All

3. 選燒錄器(Tool)

dsPIC33CK64MC105 Curiosity Nano-SN:xxxx



New Project

Steps

1. Choose Project
2. **Select Device**
3. Select Header
4. Select Plugin Board
5. Select Compiler
6. Select Project Name and Folder
7. (Optional) Add Project

Select Device

Family: All Families

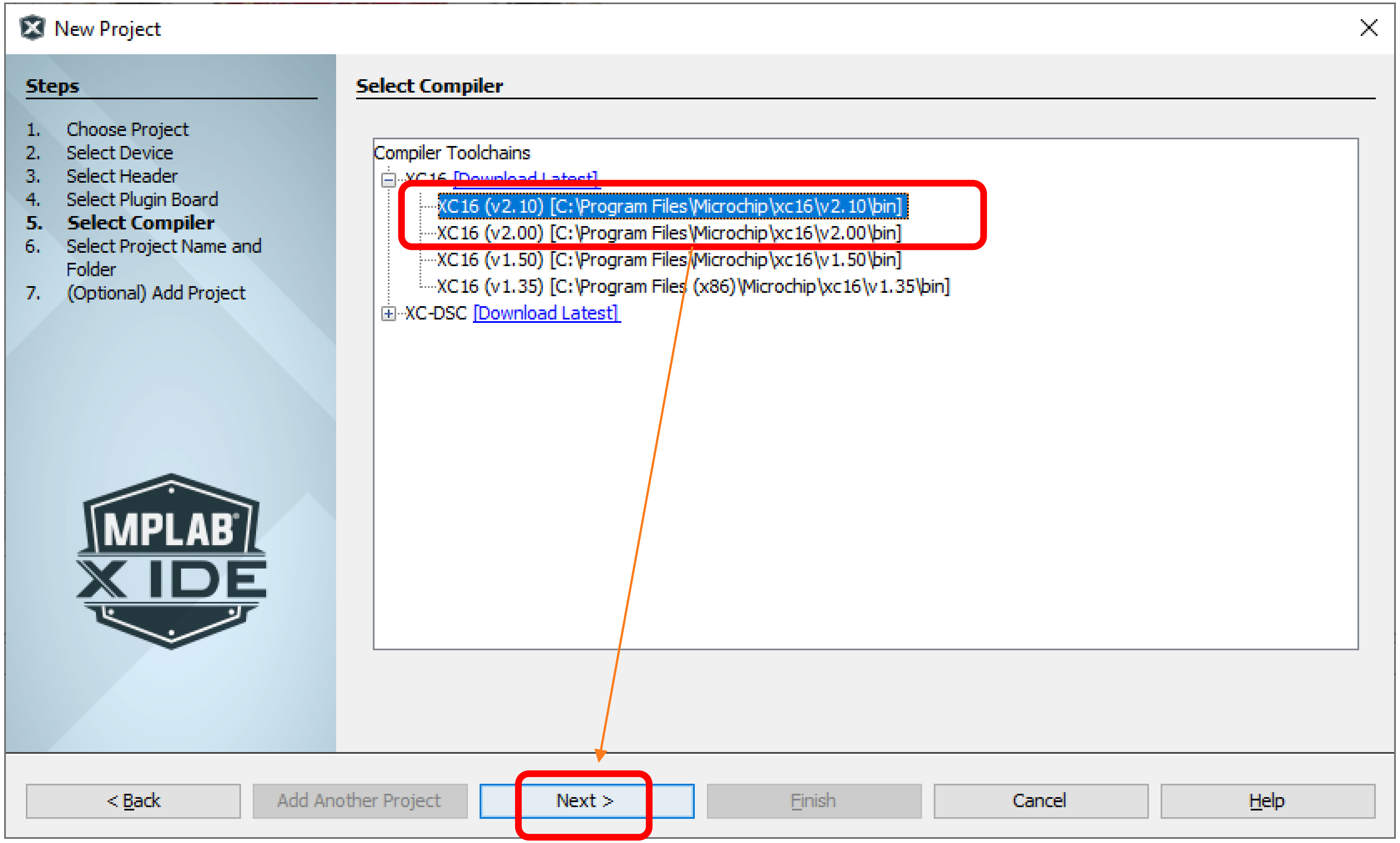
Device: dsPIC33CK64MC105

Tool: No Tool □ Show All

No Tool
Simulator
dsPIC33CK64MC105 Curiosity Nano-SN: MC020023602HIP000360

4. 選編輯軟體

XC16 (v2.10)



Steps

1. Choose Project
2. Select Device
3. Select Header
4. Select Plugin Board
5. **Select Compiler**
6. Select Project Name and Folder
7. (Optional) Add Project

Select Compiler

Compiler Toolchains

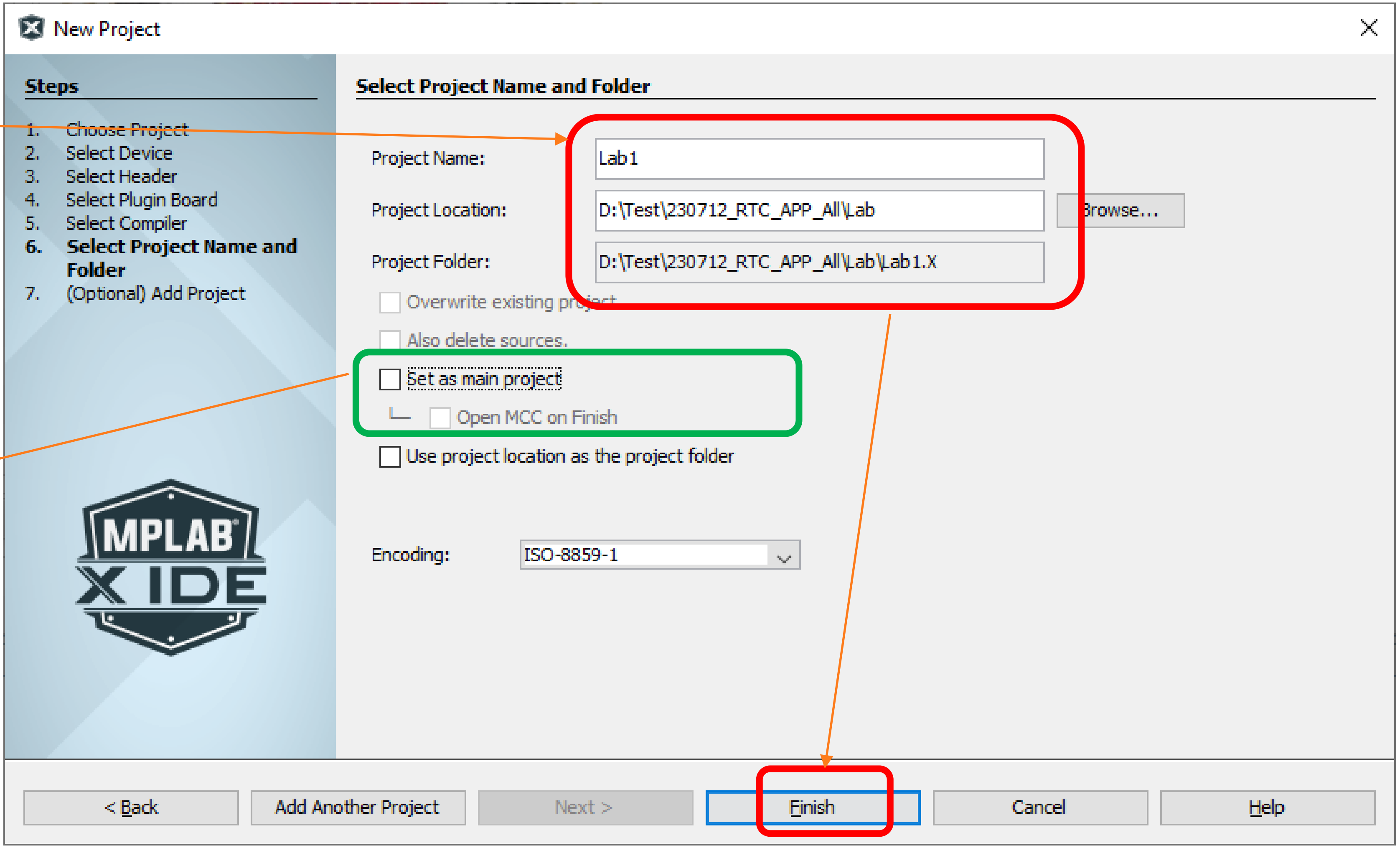
```
⊟ XC16 [Download Latest]
    XC16 (v2.10) [C:\Program Files\Microchip\xc16\v2.10\bin]
    XC16 (v2.00) [C:\Program Files\Microchip\xc16\v2.00\bin]
    XC16 (v1.50) [C:\Program Files\Microchip\xc16\v1.50\bin]
    XC16 (v1.35) [C:\Program Files (x86)\Microchip\xc16\v1.35\bin]
⊞ XC-DSC [Download Latest]
```

5.設定專案名稱

Lab1

- **Lab1**
  - 此專案名字可自訂
  - 但不能太長
  - 不要中文或特殊符號

- **Set as main project**
  - Open MCC on Finish
  - 取消打勾，就不會直接開MCC



New Project

Steps

1. Choose Project
2. Select Device
3. Select Header
4. Select Plugin Board
5. Select Compiler
6. **Select Project Name and Folder**
7. (Optional) Add Project

Select Project Name and Folder

Project Name: Lab1

Project Location: D:\Test\230712_RTC_APP_All\Lab

Project Folder: D:\Test\230712_RTC_APP_All\Lab\Lab1.X

□ Overwrite existing project

□ Also delete sources.

□ Set as main project

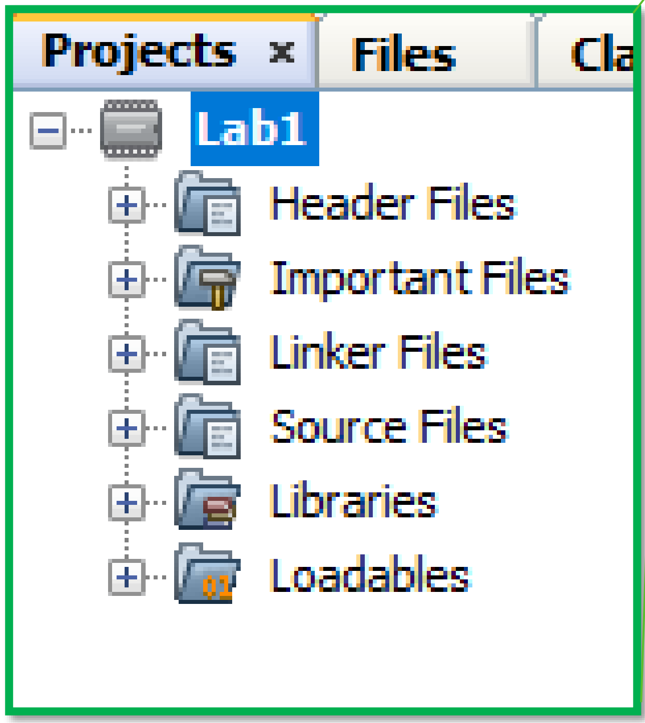
□ Open MCC on Finish

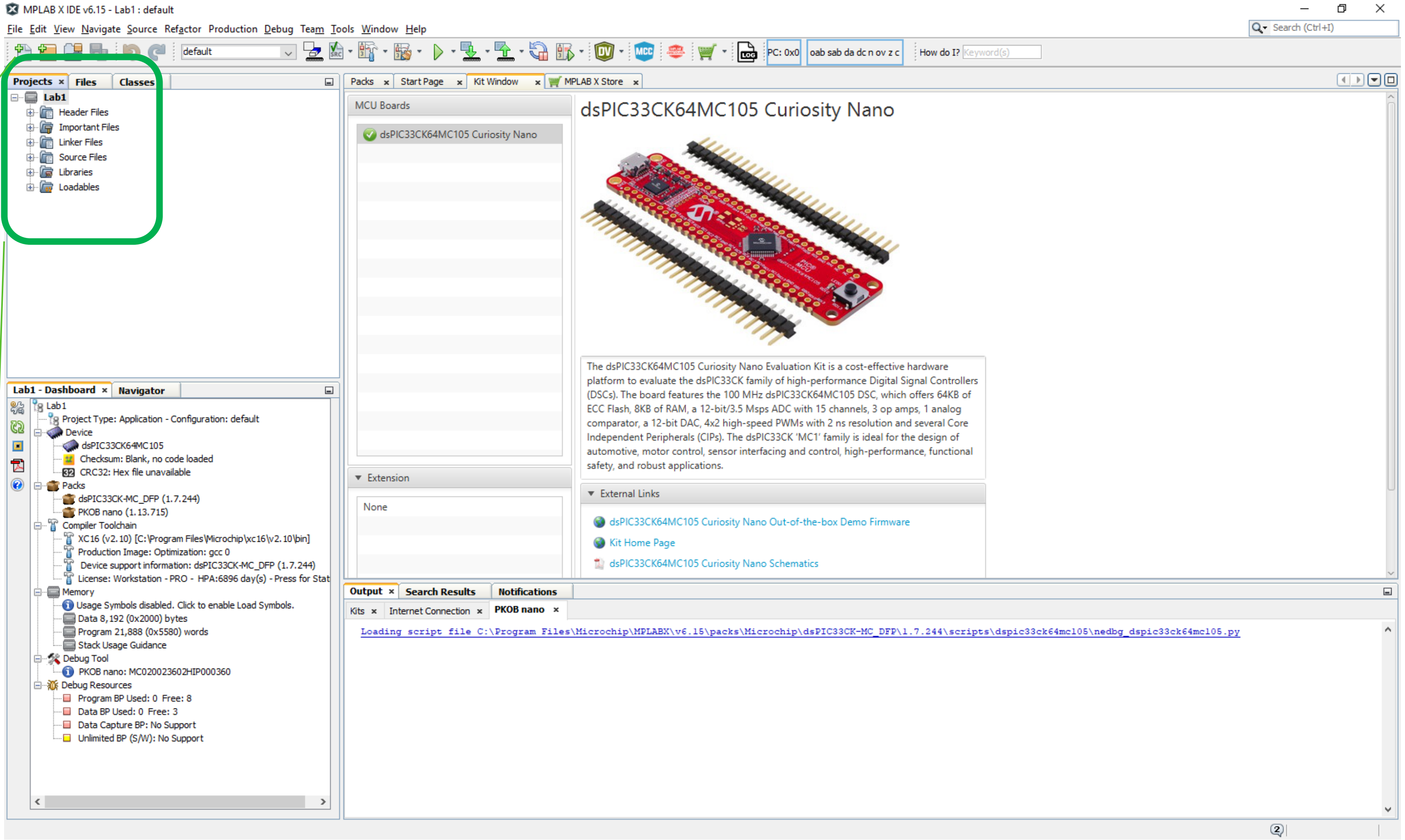
□ Use project location as the project folder

Encoding: ISO-8859-1

6.專案框架完成

此時資料夾是空的




MPLAB X IDE v6.15 - Lab1 : default
File Edit View Navigate Source Refactor Production Debug Team Tools Window Help
Search (Ctrl+I)
default
PC: 0x0
oab sab da dc n ov z c
How do I? Keyword(s)
Projects × Files Classes
Lab1
Header Files
Important Files
Linker Files
Source Files
Libraries
Loadables
Packs × Start Page × Kit Window × MPLAB X Store ×
MCU Boards
dsPIC33CK64MC105 Curiosity Nano
Extension
None
Lab1 - Dashboard × Navigator
Output × Search Results Notifications
Kits × Internet Connection × PKOB nano ×

Lab1
- Project Type: Application - Configuration: default
- Device
  - dsPIC33CK64MC105
  - Checksum: Blank, no code loaded
  - CRC32: Hex file unavailable
- Packs
  - dsPIC33CK-MC_DFP (1.7.244)
  - PKOB nano (1.13.715)
- Compiler Toolchain
  - XC16 (v2.10) [C:\Program Files\Microchip\xc16\v2.10\bin]
  - Production Image: Optimization: gcc 0
  - Device support information: dsPIC33CK-MC_DFP (1.7.244)
  - License: Workstation - PRO - HPA:6896 day(s) - Press for Stat
- Memory
  - Usage Symbols disabled. Click to enable Load Symbols.
  - Data 8,192 (0x2000) bytes
  - Program 21,888 (0x5580) words
  - Stack Usage Guidance
- Debug Tool
  - PKOB nano: MC020023602HIP000360
- Debug Resources
  - Program BP Used: 0 Free: 8
  - Data BP Used: 0 Free: 3
  - Data Capture BP: No Support
  - Unlimited BP (S/W): No Support

dsPIC33CK64MC105 Curiosity Nano

The dsPIC33CK64MC105 Curiosity Nano Evaluation Kit is a cost-effective hardware platform to evaluate the dsPIC33CK family of high-performance Digital Signal Controllers (DSCs). The board features the 100 MHz dsPIC33CK64MC105 DSC, which offers 64KB of ECC Flash, 8KB of RAM, a 12-bit/3.5 Msps ADC with 15 channels, 3 op amps, 1 analog comparator, a 12-bit DAC, 4x2 high-speed PWMs with 2 ns resolution and several Core Independent Peripherals (CIPs). The dsPIC33CK 'MC1' family is ideal for the design of automotive, motor control, sensor interfacing and control, high-performance, functional safety, and robust applications.

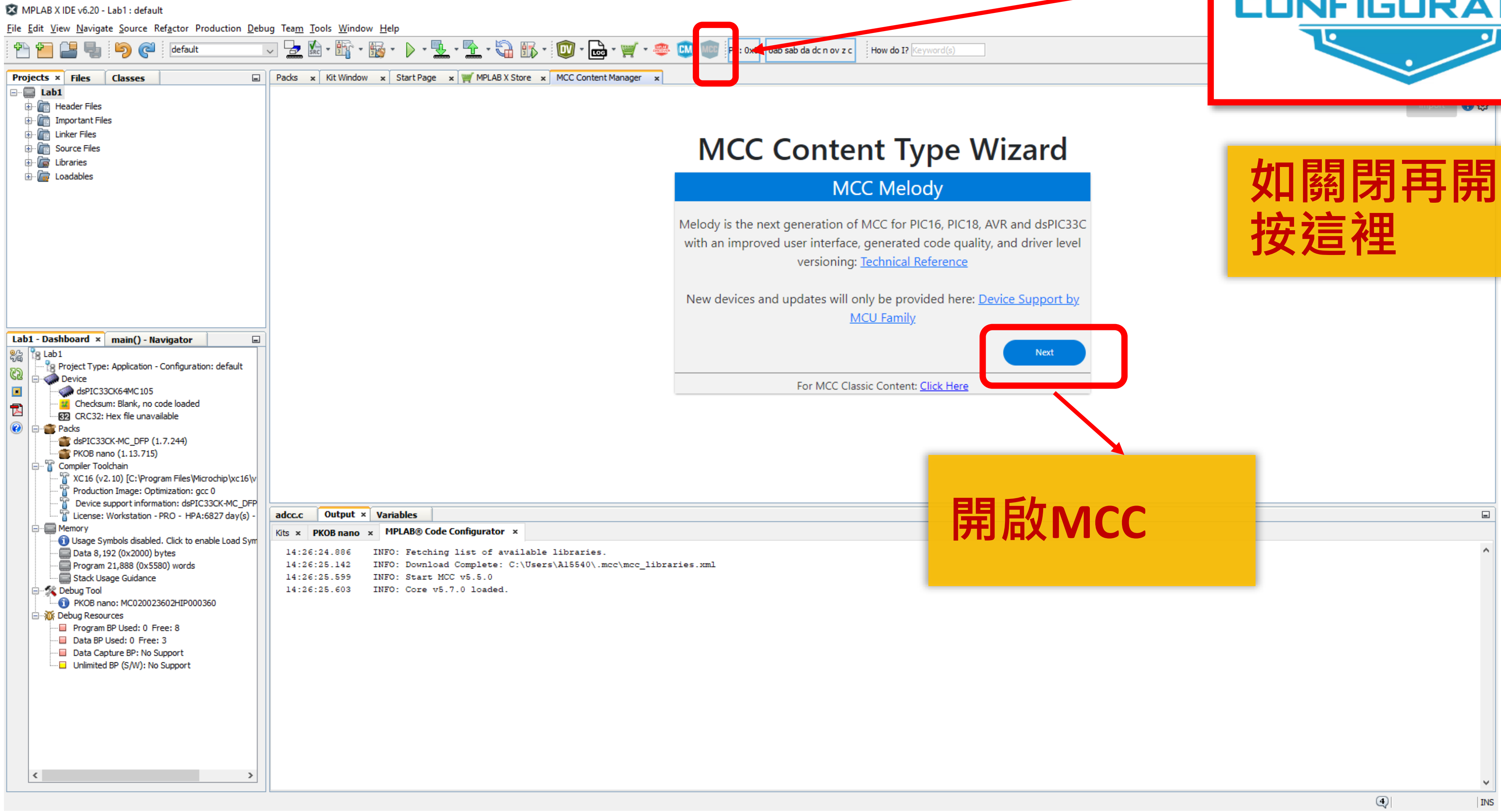
External Links

- dsPIC33CK64MC105 Curiosity Nano Out-of-the-box Demo Firmware
- Kit Home Page
- dsPIC33CK64MC105 Curiosity Nano Schematics

```
Loading script file C:\Program Files\Microchip\MPLABX\v6.15\packs\Microchip\dsPIC33CK-MC_DFP\1.7.244\scripts\dspic33ck64mc105\nedbg_dspic33ck64mc105.py
```

7.開啟MCC Melody

點選 NEXT



Lab1
Project Type: Application - Configuration: default
Device
dsPIC33CK64MC105
Checksum: Blank, no code loaded
CRC32: Hex file unavailable
Packs
dsPIC33CK-MC_DFP (1.7.244)
PKOB nano (1.13.715)
Compiler Toolchain
XC16 (v2.10) [C:\Program Files\Microchip\xc16\v
Production Image: Optimization: gcc 0
Device support information: dsPIC33CK-MC_DFP
License: Workstation - PRO - HPA:6827 day(s) -
Memory
Usage Symbols disabled. Click to enable Load Sym
Data 8,192 (0x2000) bytes
Program 21,888 (0x5580) words
Stack Usage Guidance
Debug Tool
PKOB nano: MC020023602HIP000360
Debug Resources
Program BP Used: 0 Free: 8
Data BP Used: 0 Free: 3
Data Capture BP: No Support
Unlimited BP (S/W): No Support

MCC Content Type Wizard

MCC Melody

Melody is the next generation of MCC for PIC16, PIC18, AVR and dsPIC33C with an improved user interface, generated code quality, and driver level versioning: Technical Reference

New devices and updates will only be provided here: Device Support by MCU Family

Next

For MCC Classic Content: Click Here

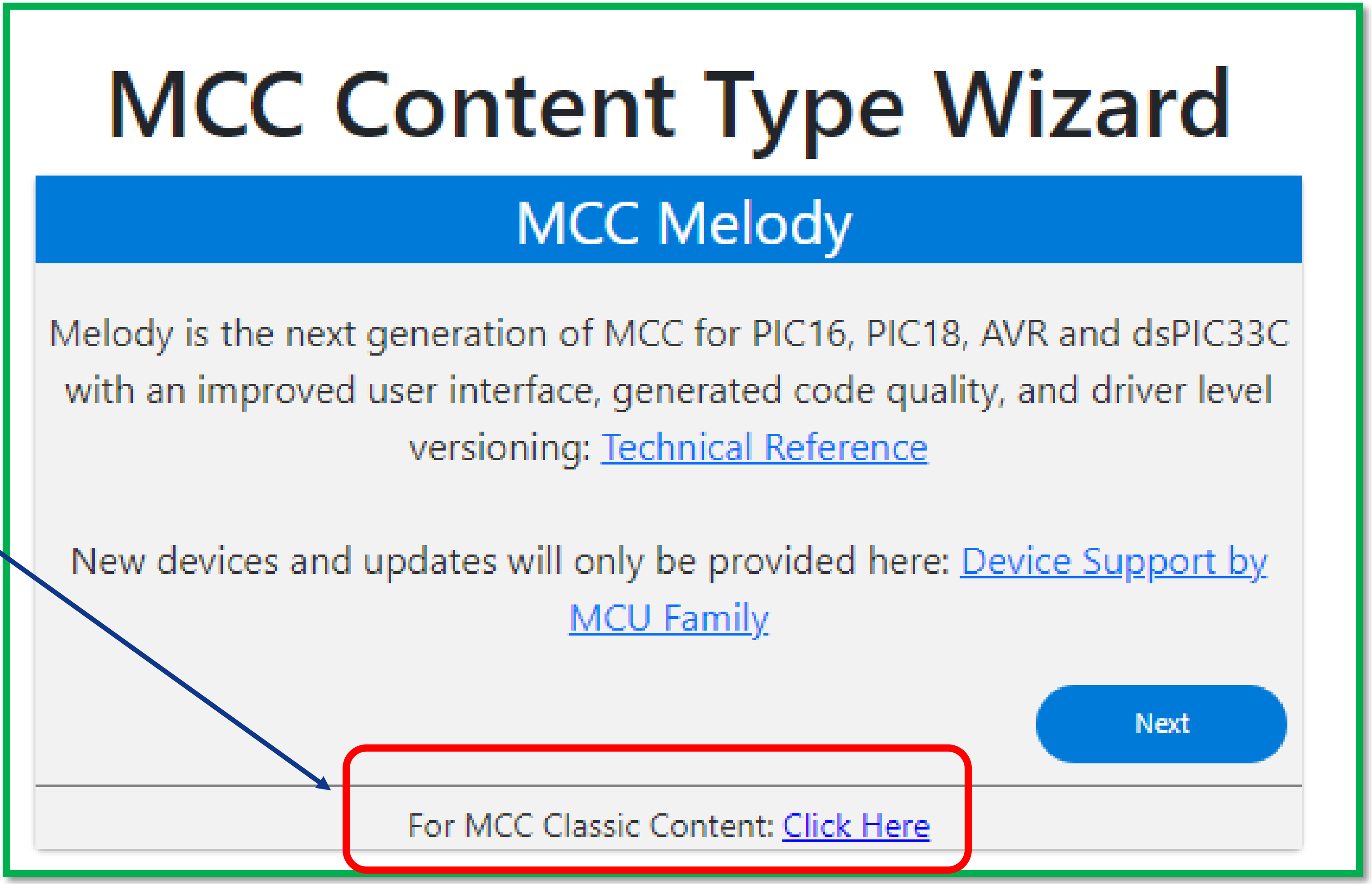
```
14:26:24.886    INFO: Fetching list of available libraries.
14:26:25.142    INFO: Download Complete: C:\Users\A15540\.mcc\mcc_libraries.xml
14:26:25.599    INFO: Start MCC v5.5.0
14:26:25.603    INFO: Core v5.7.0 loaded.
```

如關閉再開
按這裡

開啟MCC

8.選用MCC

MPLAB IDE v6.20默認開啟Melody



MCC Content Type Wizard

MCC Melody

Melody is the next generation of MCC for PIC16, PIC18, AVR and dsPIC33C with an improved user interface, generated code quality, and driver level versioning: Technical Reference

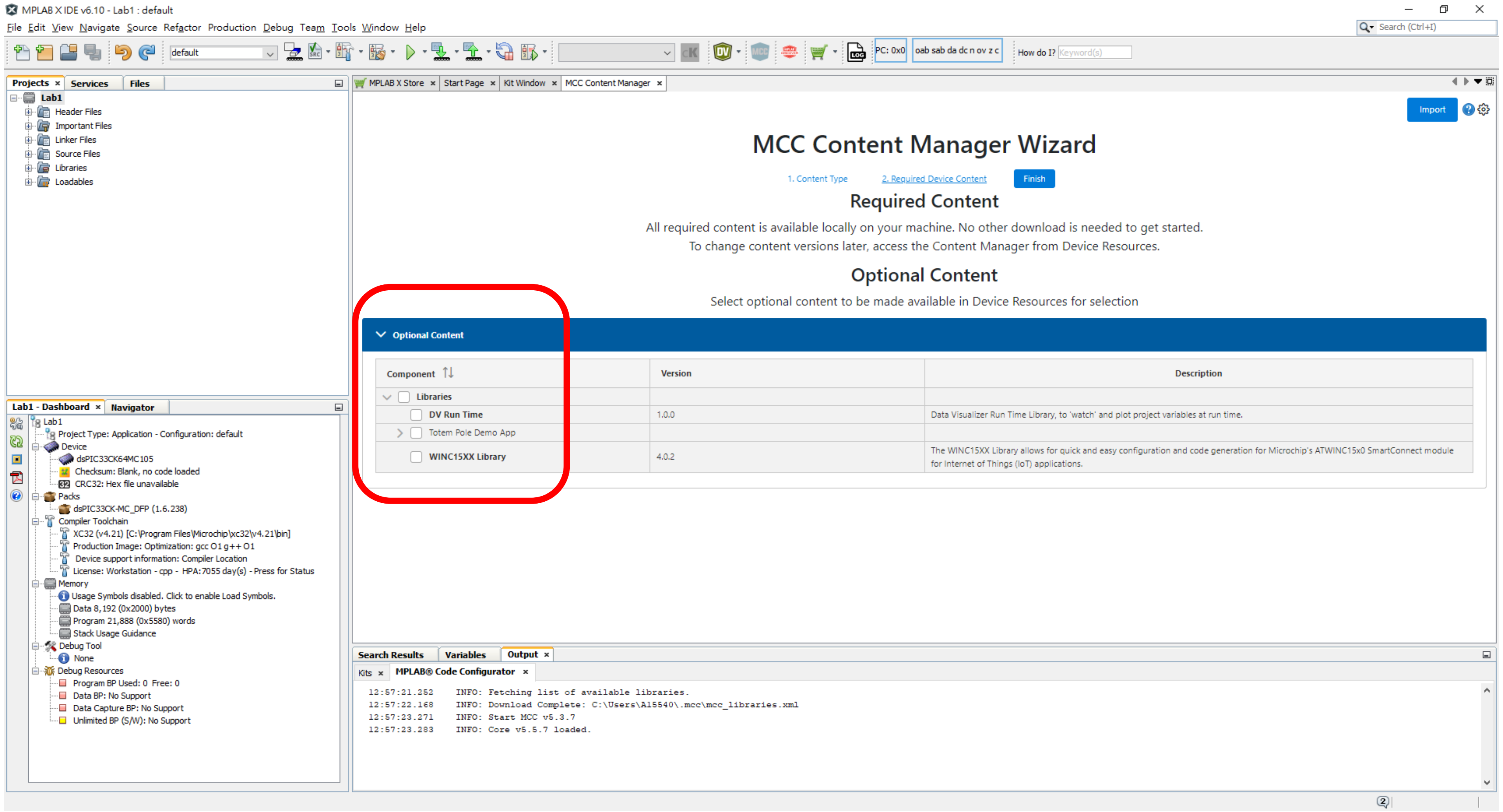
New devices and updates will only be provided here: Device Support by MCU Family

For MCC Classic Content: Click Here

要用Classic要選

9.選所需程式庫

根據專案需求選程式庫，可以不選



Lab1
Project Type: Application - Configuration: default
Device
dsPIC33CK64MC105
Checksum: Blank, no code loaded
CRC32: Hex file unavailable
Packs
dsPIC33CK-MC_DFP (1.6.238)
Compiler Toolchain
XC32 (v4.21) [C:\Program Files\Microchip\xc32\v4.21\bin]
Production Image: Optimization: gcc O1 g++ O1
Device support information: Compiler Location
License: Workstation - cpp - HPA:7055 day(s) - Press for Status
Memory
Usage Symbols disabled. Click to enable Load Symbols.
Data 8,192 (0x2000) bytes
Program 21,888 (0x5580) words
Stack Usage Guidance
Debug Tool
None
Debug Resources
Program BP Used: 0 Free: 0
Data BP: No Support
Data Capture BP: No Support
Unlimited BP (S/W): No Support

MCC Content Manager Wizard

Required Content

All required content is available locally on your machine. No other download is needed to get started.
To change content versions later, access the Content Manager from Device Resources.

Optional Content

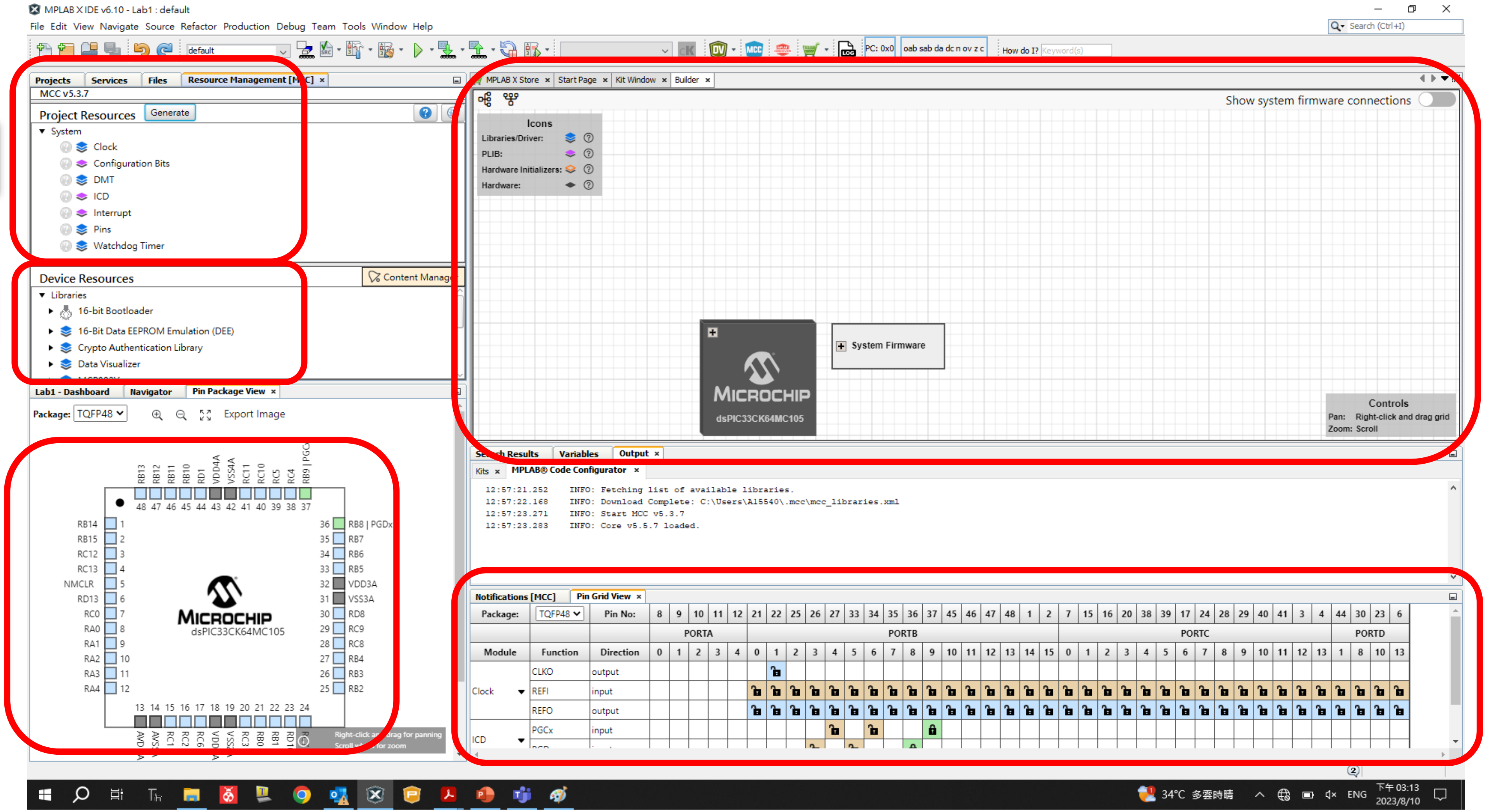
Select optional content to be made available in Device Resources for selection

Optional Content

Component	Version	Description
□ Libraries		
□ DV Run Time	1.0.0	Data Visualizer Run Time Library, to 'watch' and plot project variables at run time.
□ Totem Pole Demo App		
□ WINC15XX Library	4.0.2	The WINC15XX Library allows for quick and easy configuration and code generation for Microchip's ATWINC15x0 SmartConnect module for Internet of Things (IoT) applications.

```
12:57:21.252    INFO: Fetching list of available libraries.
12:57:22.168    INFO: Download Complete: C:\Users\A15540\.mcc\mcc_libraries.xml
12:57:23.271    INFO: Start MCC v5.3.7
12:57:23.283    INFO: Core v5.5.7 loaded.
```

10. MCC GUI
開啟



Project Resources
- System
  - Clock
  - Configuration Bits
  - DMT
  - ICD
  - Interrupt
  - Pins
  - Watchdog Timer

Device Resources
- Libraries
  - 16-bit Bootloader
  - 16-Bit Data EEPROM Emulation (DEE)
  - Crypto Authentication Library
  - Data Visualizer

```
12:57:21.252    INFO: Fetching list of available libraries.
12:57:22.168    INFO: Download Complete: C:\Users\A15540\.mcc\mcc_libraries.xml
12:57:23.271    INFO: Start MCC v5.3.7
12:57:23.283    INFO: Core v5.5.7 loaded.
```

Package:	TQFP48	Pin No:	8	9	10	11	12	21	22	25	26	27	33	34	35	36	37	45	46	47	48	1	2	7	15	16	20	38	39	17	24	28	29	40	41	3	4	44	30	23	6
			PORTA					PORTB																PORTC														PORTD			
Module	Function	Direction	0	1	2	3	4	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	0	1	2	3	4	5	6	7	8	9	10	11	12	13	1	8	10	13
Clock ▾	CLKO	output							🔓																																
	REFI	input						🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓
	REFO	output						🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓
ICD ▾	PGCx	input										🔓		🔓			🔒																								

IC使用資源

IC可用資源

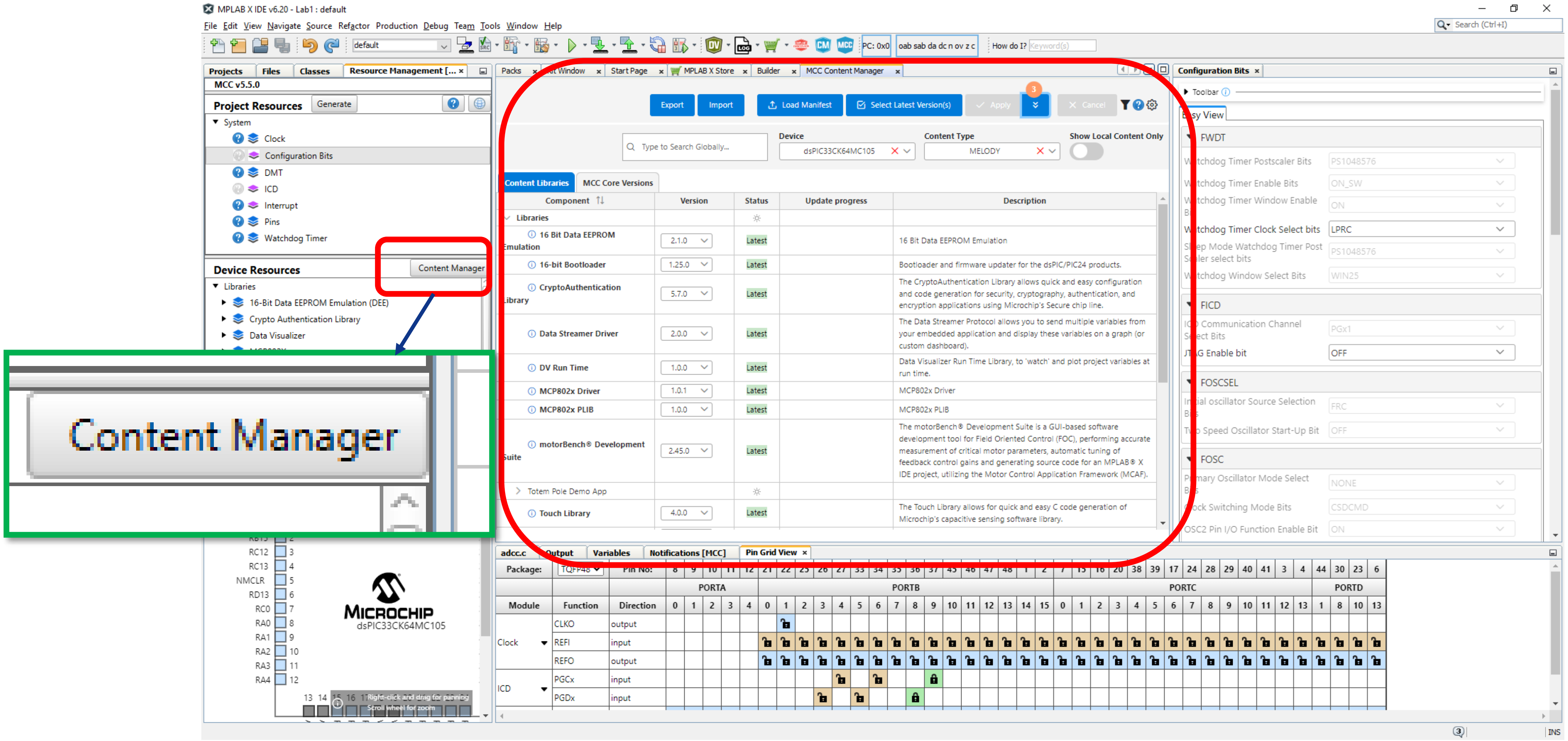
腳位規劃

MCC
程式庫
設計
架構圖

腳位功能
規劃

11.下載MCC資料

Content Manager –選擇其它程式庫



Device: dsPIC33CK64MC105

Content Type: MELODY

Show Local Content Only

Component	Version	Status	Update progress	Description
Libraries				
16 Bit Data EEPROM Emulation	2.1.0	Latest		16 Bit Data EEPROM Emulation
16-bit Bootloader	1.25.0	Latest		Bootloader and firmware updater for the dsPIC/PIC24 products.
CryptoAuthentication Library	5.7.0	Latest		The CryptoAuthentication Library allows quick and easy configuration and code generation for security, cryptography, authentication, and encryption applications using Microchip's Secure chip line.
Data Streamer Driver	2.0.0	Latest		The Data Streamer Protocol allows you to send multiple variables from your embedded application and display these variables on a graph (or custom dashboard).
DV Run Time	1.0.0	Latest		Data Visualizer Run Time Library, to 'watch' and plot project variables at run time.
MCP802x Driver	1.0.1	Latest		MCP802x Driver
MCP802x PLIB	1.0.0	Latest		MCP802x PLIB
motorBench® Development Suite	2.45.0	Latest		The motorBench® Development Suite is a GUI-based software development tool for Field Oriented Control (FOC), performing accurate measurement of critical motor parameters, automatic tuning of feedback control gains and generating source code for an MPLAB® X IDE project, utilizing the Motor Control Application Framework (MCAF).
Totem Pole Demo App				
Touch Library	4.0.0	Latest		The Touch Library allows for quick and easy C code generation of Microchip's capacitive sensing software library.

FWDT
- Watchdog Timer Postscaler Bits: PS1048576
- Watchdog Timer Enable Bits: ON_SW
- Watchdog Timer Window Enable Bit: ON
- Watchdog Timer Clock Select bits: LPRC
- Sleep Mode Watchdog Timer Post Scaler select bits: PS1048576
- Watchdog Window Select Bits: WIN25

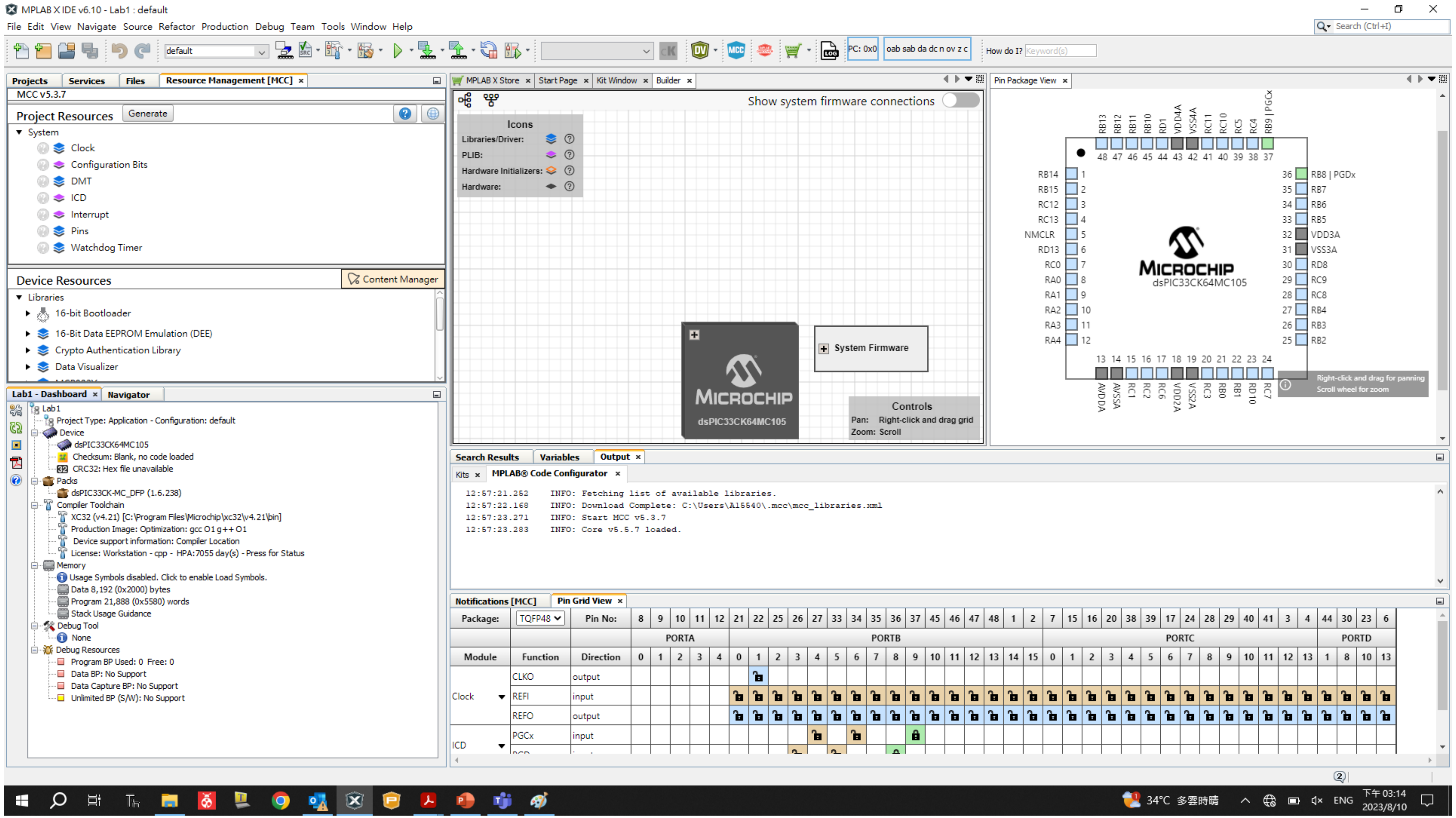
FICD
- ICD Communication Channel Select Bits: PGx1
- JTAG Enable bit: OFF

FOSCSEL
- Initial oscillator Source Selection Bits: FRC
- Two Speed Oscillator Start-Up Bit: OFF

FOSC
- Primary Oscillator Mode Select Bits: NONE
- Clock Switching Mode Bits: CSDCMD
- OSC2 Pin I/O Function Enable Bit: ON

12.重新排版

可以用拖曳方式，調成你喜歡的配置



MCC v5.3.7

Project Resources — Generate

- System
  - Clock
  - Configuration Bits
  - DMT
  - ICD
  - Interrupt
  - Pins
  - Watchdog Timer

Device Resources — Content Manager

- Libraries
  - 16-bit Bootloader
  - 16-Bit Data EEPROM Emulation (DEE)
  - Crypto Authentication Library
  - Data Visualizer

Lab1 - Dashboard | Navigator

- Lab1
  - Project Type: Application - Configuration: default
  - Device
    - dsPIC33CK64MC105
    - Checksum: Blank, no code loaded
    - CRC32: Hex file unavailable
  - Packs
    - dsPIC33CK-MC_DFP (1.6.238)
  - Compiler Toolchain
    - XC32 (v4.21) [C:\Program Files\Microchip\xc32\v4.21\bin]
    - Production Image: Optimization: gcc O1 g++ O1
    - Device support information: Compiler Location
    - License: Workstation - cpp - HPA:7055 day(s) - Press for Status
  - Memory
    - Usage Symbols disabled. Click to enable Load Symbols.
    - Data 8,192 (0x2000) bytes
    - Program 21,888 (0x5580) words
    - Stack Usage Guidance
  - Debug Tool
    - None
  - Debug Resources
    - Program BP Used: 0 Free: 0
    - Data BP: No Support
    - Data Capture BP: No Support
    - Unlimited BP (S/W): No Support

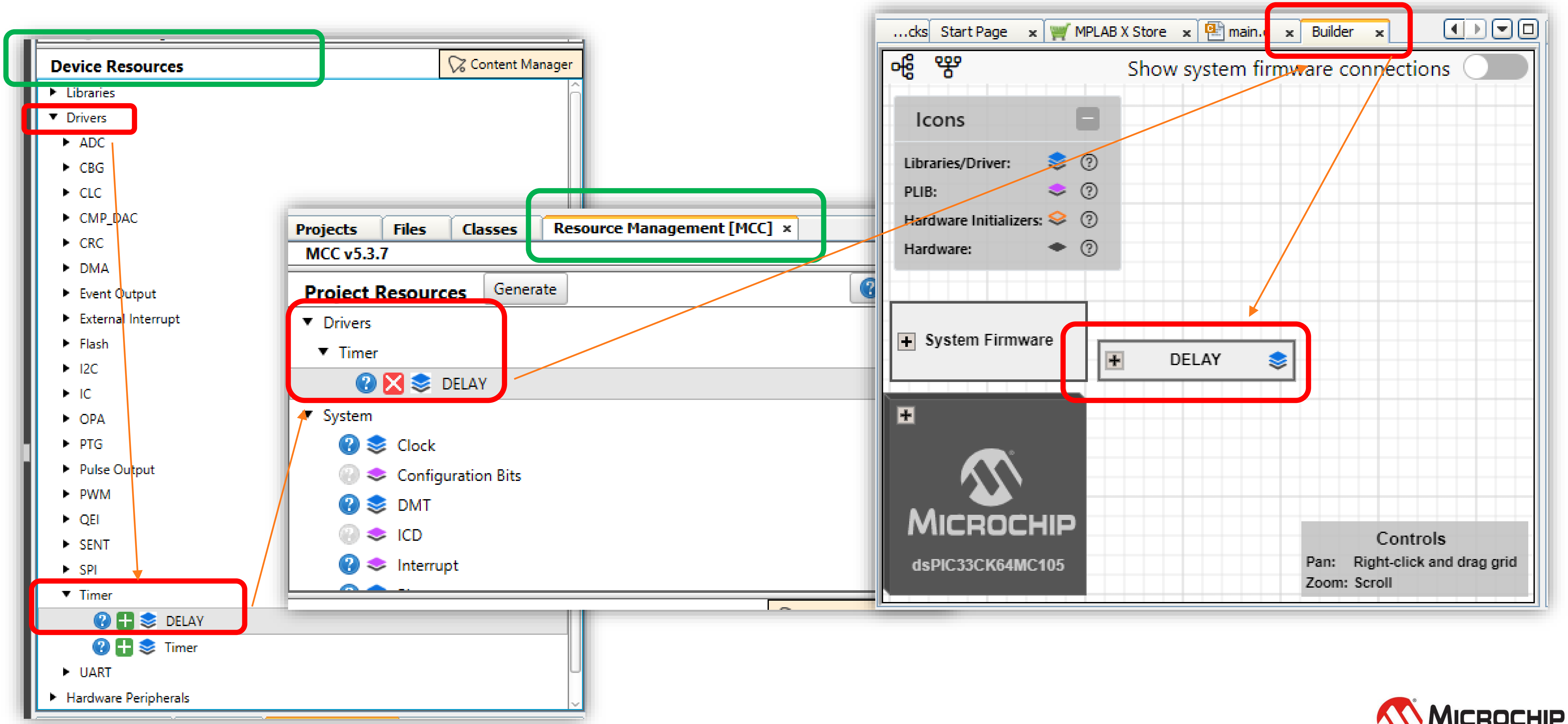
MPLAB® Code Configurator

```
12:57:21.252    INFO: Fetching list of available libraries.
12:57:22.168    INFO: Download Complete: C:\Users\A15540\.mcc\mcc_libraries.xml
12:57:23.271    INFO: Start MCC v5.3.7
12:57:23.283    INFO: Core v5.5.7 loaded.
```

Package:	TQFP48	Pin No:	8	9	10	11	12	21	22	25	26	27	33	34	35	36	37	45	46	47	48	1	2	7	15	16	20	38	39	17	24	28	29	40	41	3	4	44	30	23	6
			PORTA					PORTB																PORTC														PORTD			
Module	Function	Direction	0	1	2	3	4	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	0	1	2	3	4	5	6	7	8	9	10	11	12	13	1	8	10	13
Clock	CLKO	output							🔓																																
	REFI	input						🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓
	REFO	output						🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓
ICD	PGCx	input										🔓		🔓			🔒																								

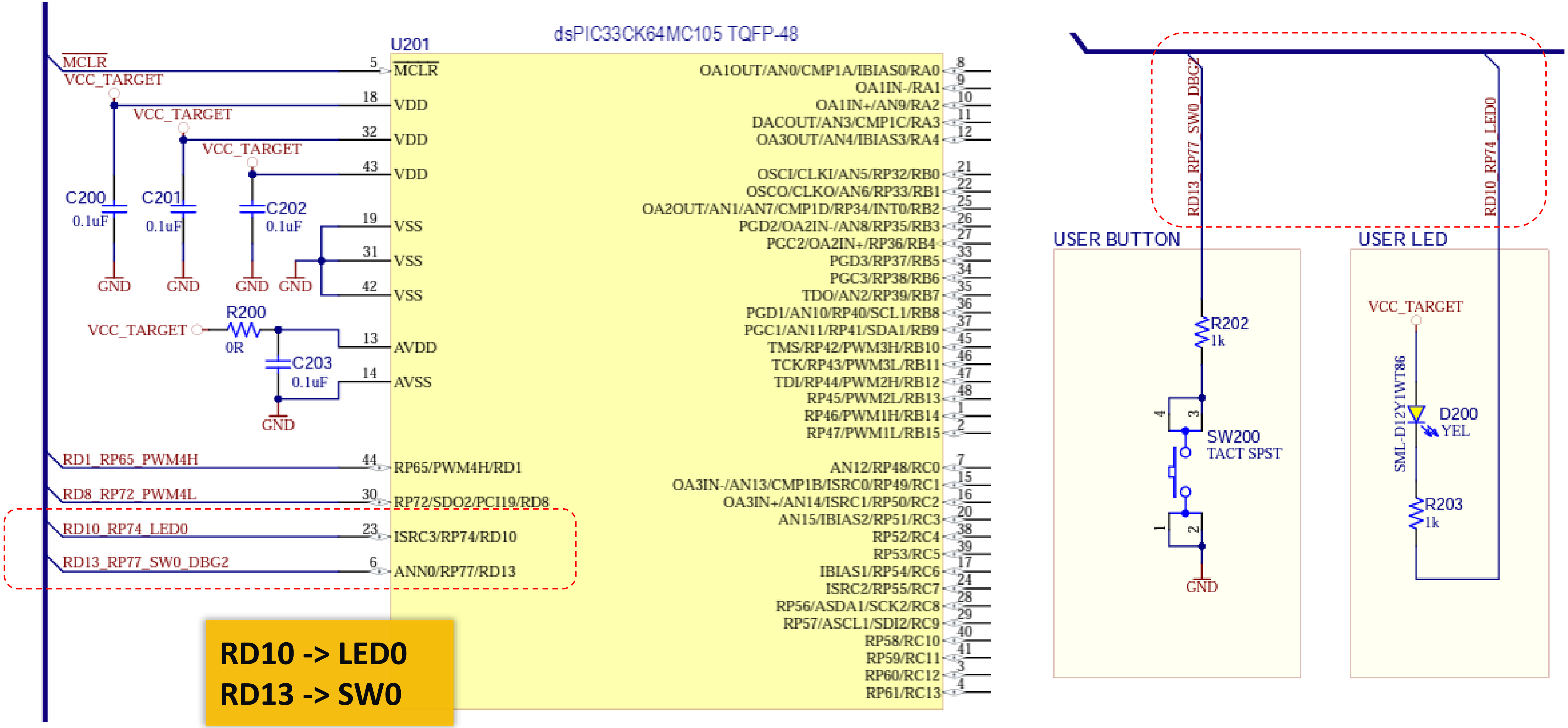
13.加入Delay函式

到Drivers->Timer->點選DELAY



14.確認接腳

dsPIC33CK64MC105_Curiosity_Nano_Schematics.pdf



RD10 -> LED0
RD13 -> SW0

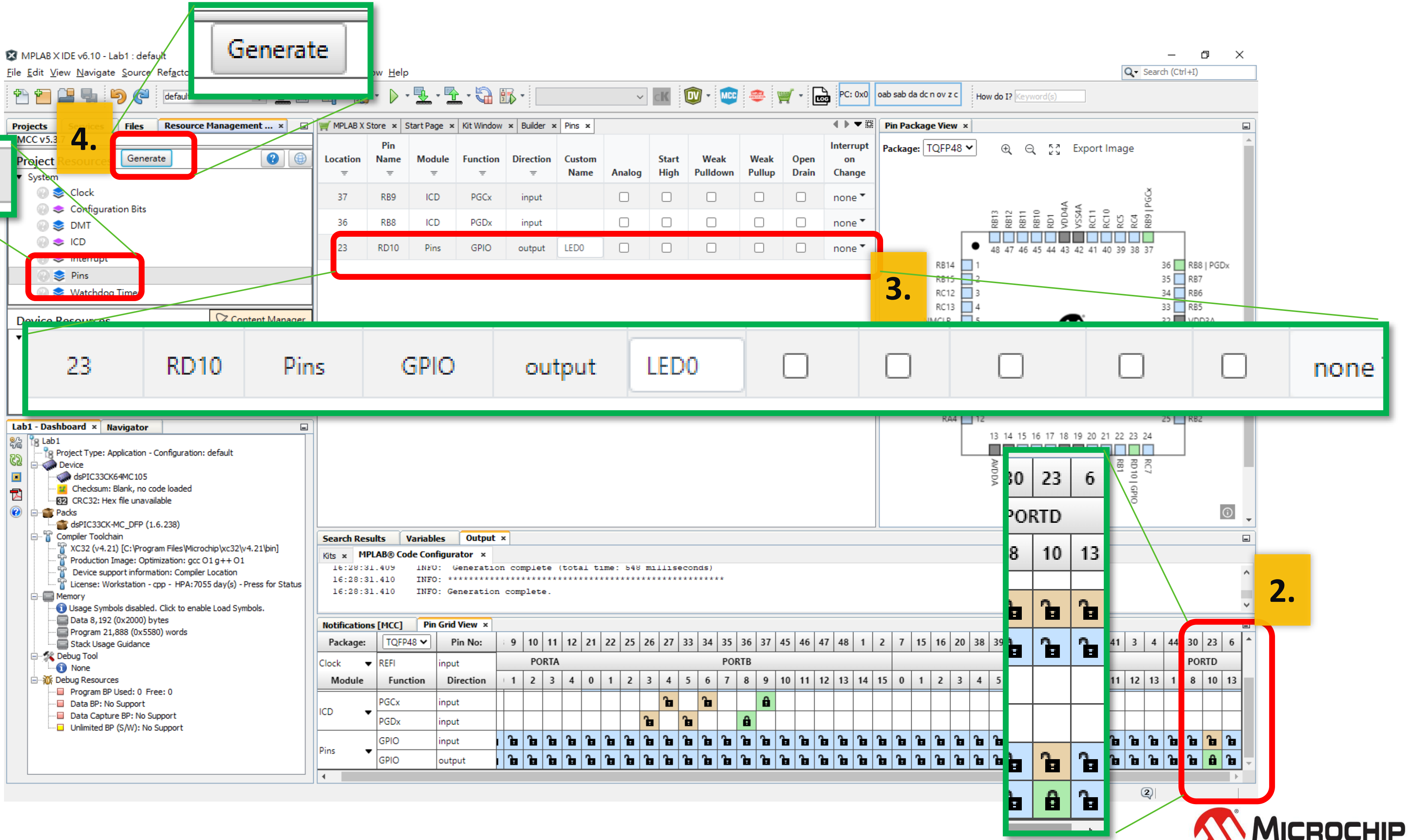
15.設定LED0

1. 點Pins

2. 點RD10
   設輸出

3. 改名
   LED0

4. 產生
   CODE



Location	Pin Name	Module	Function	Direction	Custom Name	Analog	Start High	Weak Pulldown	Weak Pullup	Open Drain	Interrupt on Change
37	RB9	ICD	PGCx	input		□	□	□	□	□	none
36	RB8	ICD	PGDx	input		□	□	□	□	□	none
23	RD10	Pins	GPIO	output	LED0	□	□	□	□	□	none

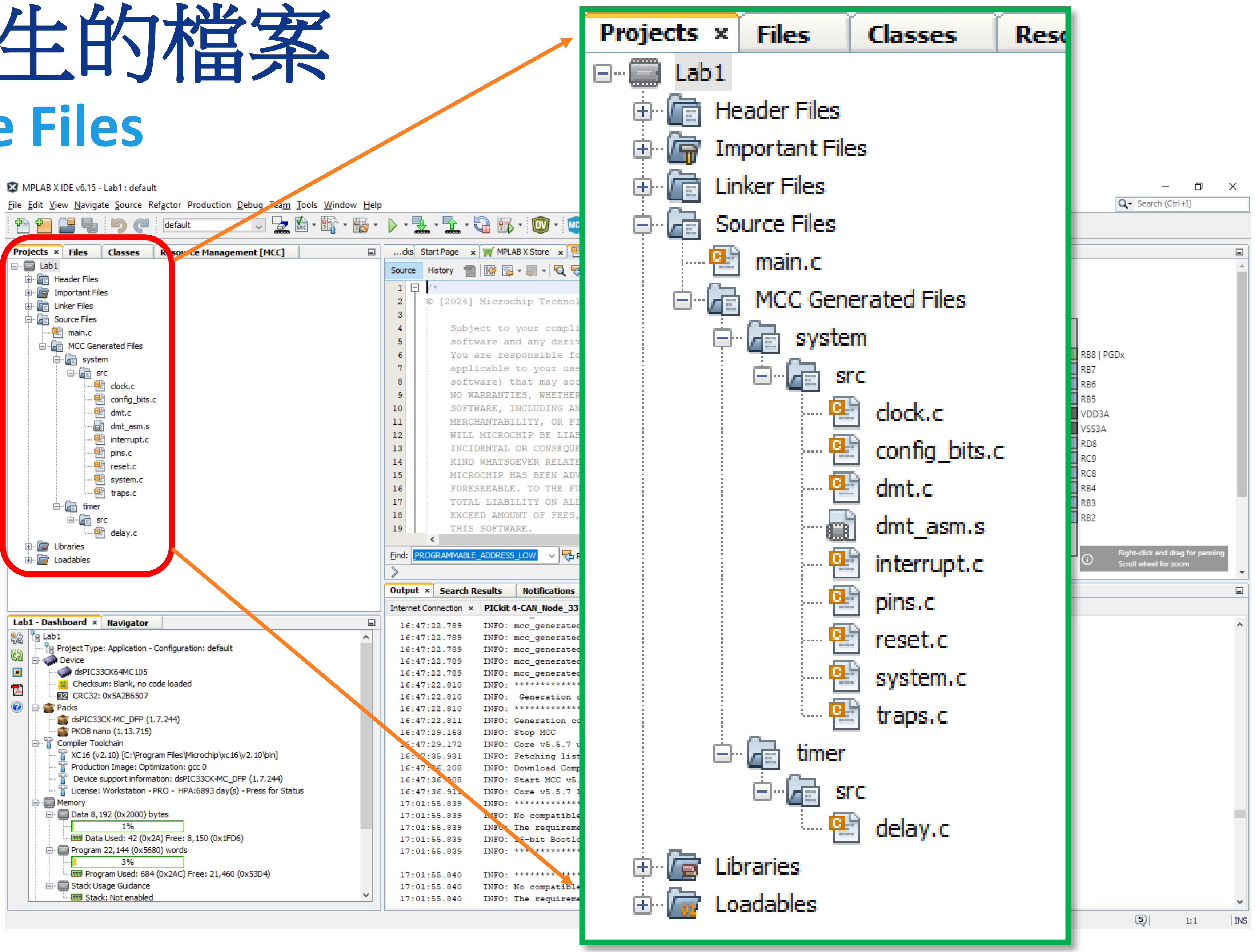
```
16:28:31.409    INFO:  Generation complete (total time: 548 milliseconds)
16:28:31.410    INFO: ****************************************************
16:28:31.410    INFO: Generation complete.
```

Module	Function	Direction
Clock	REFI	input
ICD	PGCx	input
	PGDx	input
Pins	GPIO	input
	GPIO	output

Lab1 - Dashboard
- Lab1
  - Project Type: Application - Configuration: default
  - Device
    - dsPIC33CK64MC105
    - Checksum: Blank, no code loaded
    - CRC32: Hex file unavailable
  - Packs
    - dsPIC33CK-MC_DFP (1.6.238)
  - Compiler Toolchain
    - XC32 (v4.21) [C:\Program Files\Microchip\xc32\v4.21\bin]
    - Production Image: Optimization: gcc O1 g++ O1
    - Device support information: Compiler Location
    - License: Workstation - cpp - HPA:7055 day(s) - Press for Status
  - Memory
    - Usage Symbols disabled. Click to enable Load Symbols.
    - Data 8,192 (0x2000) bytes
    - Program 21,888 (0x5580) words
    - Stack Usage Guidance
  - Debug Tool
    - None
  - Debug Resources
    - Program BP Used: 0 Free: 0
    - Data BP: No Support
    - Data Capture BP: No Support
    - Unlimited BP (S/W): No Support

16. 產生的檔案

在Source Files



```
Lab1
  Header Files
  Important Files
  Linker Files
  Source Files
    main.c
    MCC Generated Files
      system
        src
          clock.c
          config_bits.c
          dmt.c
          dmt_asm.s
          interrupt.c
          pins.c
          reset.c
          system.c
          traps.c
      timer
        src
          delay.c
  Libraries
  Loadables
```

```
/*
© [2024] Microchip Technol

    Subject to your compli
    software and any deriv
    You are responsible fo
    applicable to your use
    software) that may acc
    NO WARRANTIES, WHETHER
    SOFTWARE, INCLUDING AN
    MERCHANTABILITY, OR FI
    WILL MICROCHIP BE LIAB
    INCIDENTAL OR CONSEQUE
    KIND WHATSOEVER RELATE
    MICROCHIP HAS BEEN ADV
    FORESEEABLE. TO THE FU
    TOTAL LIABILITY ON ALL
    EXCEED AMOUNT OF FEES,
    THIS SOFTWARE.
```

Find: PROGRAMMABLE_ADDRESS_LOW

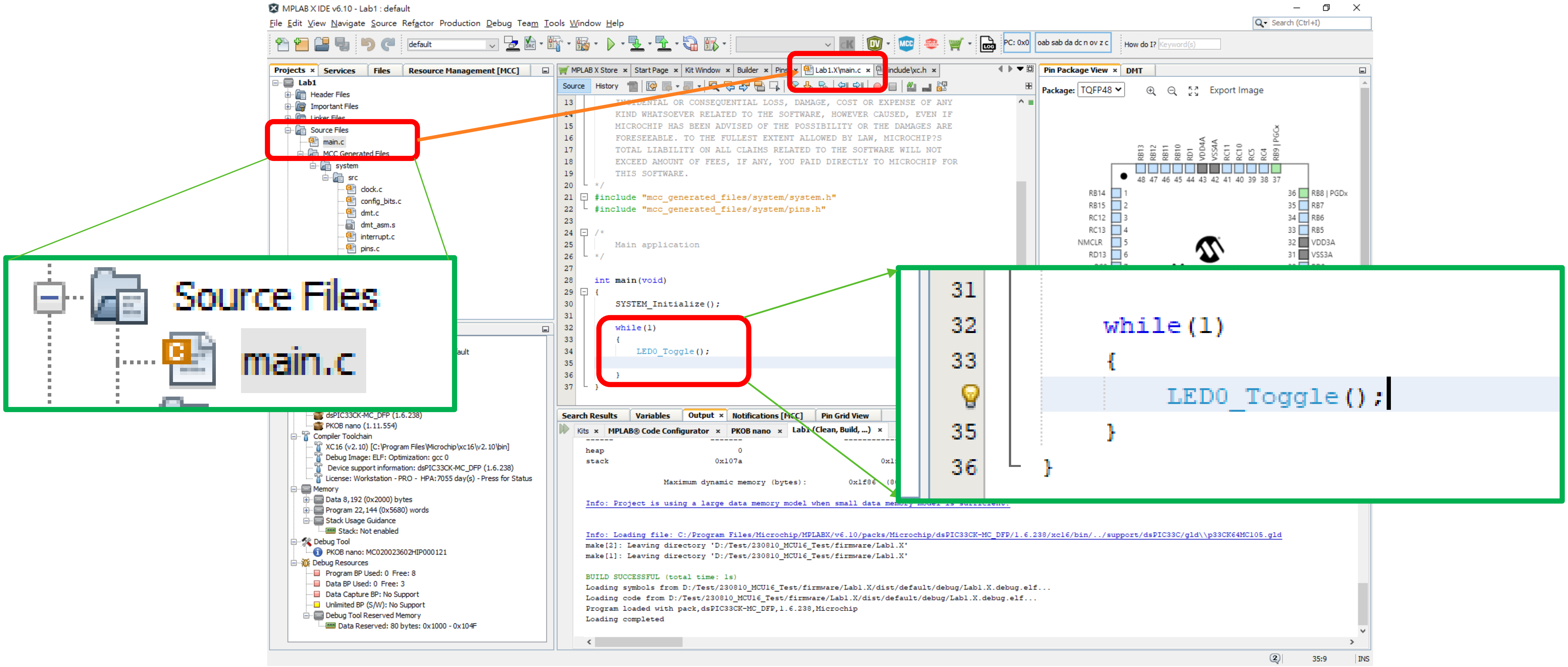
Lab1 - Dashboard

```
Lab1
  Project Type: Application - Configuration: default
  Device
    dsPIC33CK64MC105
    Checksum: Blank, no code loaded
    CRC32: 0x5A2B6507
  Packs
    dsPIC33CK-MC_DFP (1.7.244)
    PKOB nano (1.13.715)
  Compiler Toolchain
    XC16 (v2.10) [C:\Program Files\Microchip\xc16\v2.10\bin]
    Production Image: Optimization: gcc 0
    Device support information: dsPIC33CK-MC_DFP (1.7.244)
    License: Workstation - PRO - HPA:6893 day(s) - Press for Status
  Memory
    Data 8,192 (0x2000) bytes
      1%
      Data Used: 42 (0x2A) Free: 8,150 (0x1FD6)
    Program 22,144 (0x5680) words
      3%
      Program Used: 684 (0x2AC) Free: 21,460 (0x53D4)
    Stack Usage Guidance
      Stack: Not enabled
```

```
16:47:22.789   INFO: mcc_generated
16:47:22.789   INFO: mcc_generated
16:47:22.789   INFO: mcc_generated
16:47:22.789   INFO: mcc_generated
16:47:22.789   INFO: mcc_generated
16:47:22.810   INFO: ************
16:47:22.810   INFO:  Generation c
16:47:22.810   INFO: ************
16:47:22.811   INFO: Generation co
16:47:29.153   INFO: Stop MCC
16:47:29.172   INFO: Core v5.5.7 u
16:47:35.931   INFO: Fetching list
16:47:36.208   INFO: Download Comp
16:47:36.908   INFO: Start MCC v5.
16:47:36.911   INFO: Core v5.5.7 I
17:01:55.839   INFO: ************
17:01:55.839   INFO: No compatible
17:01:55.839   INFO: The requireme
17:01:55.839   INFO: 16-bit Bootlo
17:01:55.839   INFO: ************

17:01:55.840   INFO: ************
17:01:55.840   INFO: No compatible
17:01:55.840   INFO: The requireme
```

17. 寫程式 - LED

開啟main.c → 在while (1) →加入 LED0_Toggle(); →達到LED反向

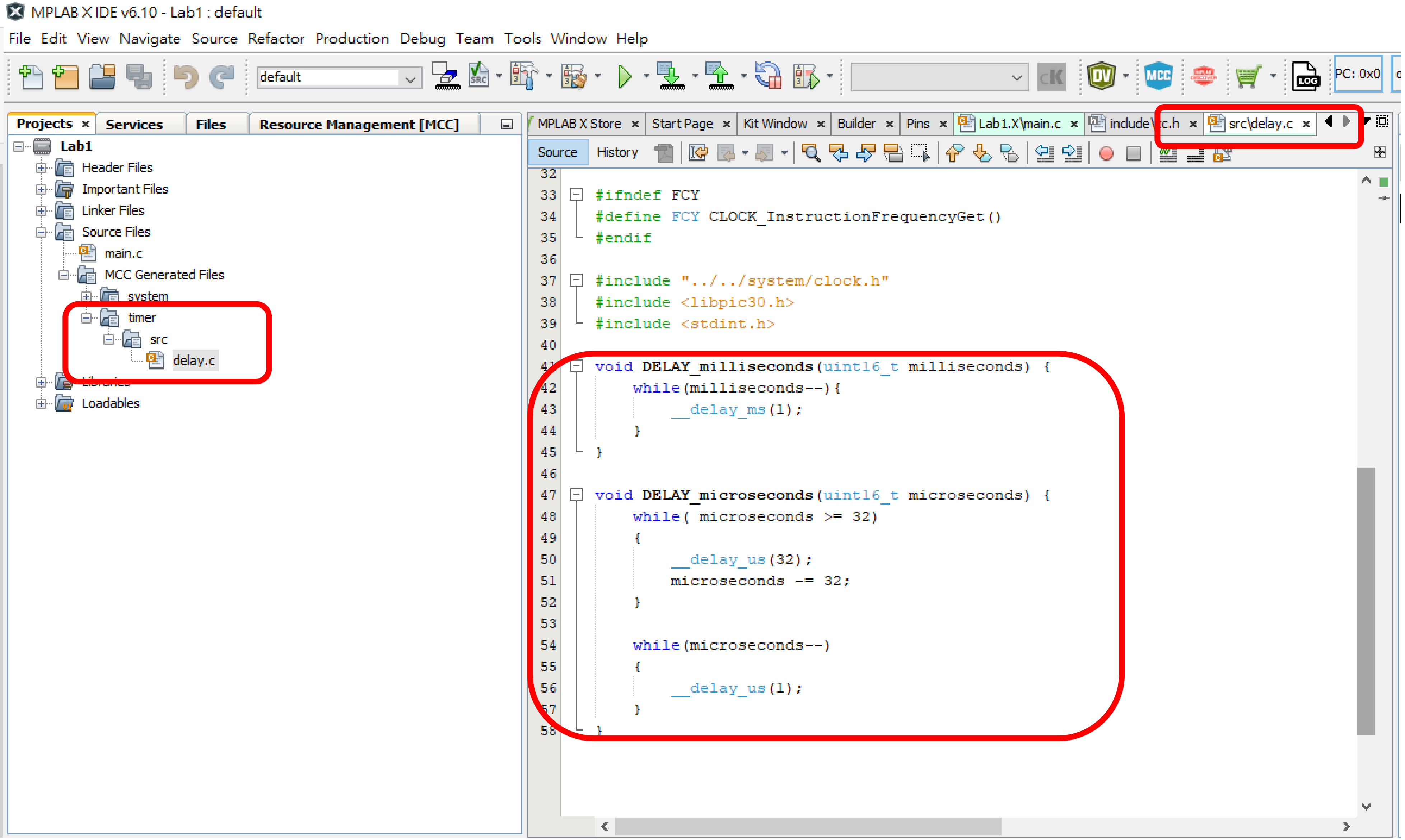


```c
    INCIDENTAL OR CONSEQUENTIAL LOSS, DAMAGE, COST OR EXPENSE OF ANY
    KIND WHATSOEVER RELATED TO THE SOFTWARE, HOWEVER CAUSED, EVEN IF
    MICROCHIP HAS BEEN ADVISED OF THE POSSIBILITY OR THE DAMAGES ARE
    FORESEEABLE. TO THE FULLEST EXTENT ALLOWED BY LAW, MICROCHIP?S
    TOTAL LIABILITY ON ALL CLAIMS RELATED TO THE SOFTWARE WILL NOT
    EXCEED AMOUNT OF FEES, IF ANY, YOU PAID DIRECTLY TO MICROCHIP FOR
    THIS SOFTWARE.
*/
#include "mcc_generated_files/system/system.h"
#include "mcc_generated_files/system/pins.h"

/*
    Main application
*/

int main(void)
{
    SYSTEM_Initialize();

    while(1)
    {
        LED0_Toggle();

    }
}
```

```
dsPIC33CK-MC_DFP (1.6.238)
PKOB nano (1.11.554)
Compiler Toolchain
    XC16 (v2.10) [C:\Program Files\Microchip\xc16\v2.10\bin]
    Debug Image: ELF: Optimization: gcc 0
    Device support information: dsPIC33CK-MC_DFP (1.6.238)
    License: Workstation - PRO - HPA:7055 day(s) - Press for Status
Memory
    Data 8,192 (0x2000) bytes
    Program 22,144 (0x5680) words
    Stack Usage Guidance
        Stack: Not enabled
Debug Tool
    PKOB nano: MC020023602HIP000121
Debug Resources
    Program BP Used: 0  Free: 8
    Data BP Used: 0  Free: 3
    Data Capture BP: No Support
    Unlimited BP (S/W): No Support
    Debug Tool Reserved Memory
        Data Reserved: 80 bytes: 0x1000 - 0x104F
```

```
------                    --------                    ------------
heap                             0
stack                      0x107a                     0x1

                Maximum dynamic memory (bytes):        0x1f86  (8

Info: Project is using a large data memory model when small data memory model is sufficient.


Info: Loading file: C:/Program Files/Microchip/MPLABX/v6.10/packs/Microchip/dsPIC33CK-MC_DFP/1.6.238/xc16/bin/../support/dsPIC33C/gld\\p33CK64MC105.gld
make[2]: Leaving directory 'D:/Test/230810_MCU16_Test/firmware/Lab1.X'
make[1]: Leaving directory 'D:/Test/230810_MCU16_Test/firmware/Lab1.X'

BUILD SUCCESSFUL (total time: 1s)
Loading symbols from D:/Test/230810_MCU16_Test/firmware/Lab1.X/dist/default/debug/Lab1.X.debug.elf...
Loading code from D:/Test/230810_MCU16_Test/firmware/Lab1.X/dist/default/debug/Lab1.X.debug.elf...
Program loaded with pack,dsPIC33CK-MC_DFP,1.6.238,Microchip
Loading completed
```

18. Delay函式

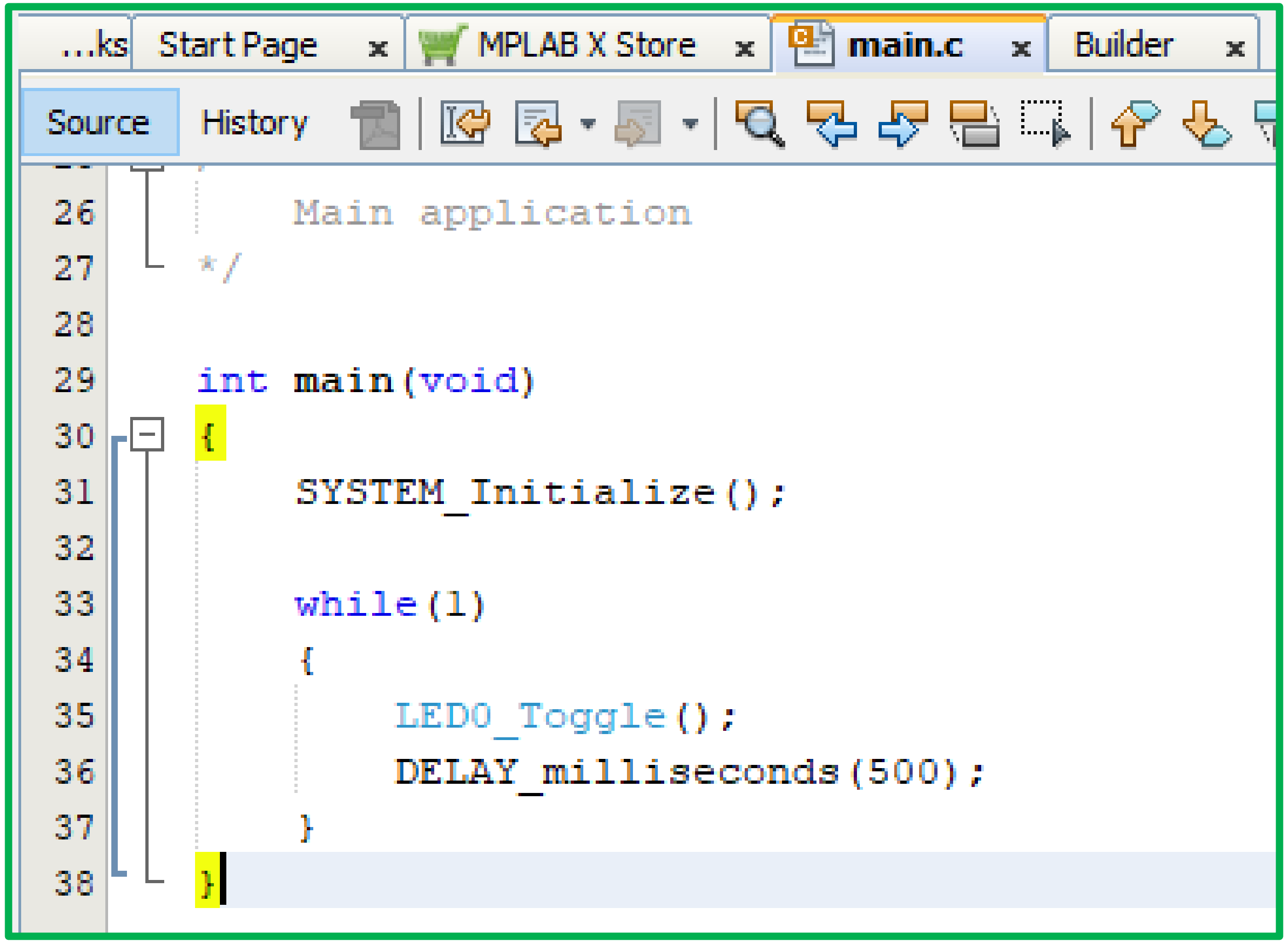


```c
#ifndef FCY
#define FCY CLOCK_InstructionFrequencyGet()
#endif

#include "../../system/clock.h"
#include <libpic30.h>
#include <stdint.h>

void DELAY_milliseconds(uint16_t milliseconds) {
    while(milliseconds--){
        __delay_ms(1);
    }
}

void DELAY_microseconds(uint16_t microseconds) {
    while( microseconds >= 32)
    {
        __delay_us(32);
        microseconds -= 32;
    }

    while(microseconds--)
    {
        __delay_us(1);
    }
}
```

19. 寫程式 - Delay

加入延時 500ms

- **MCC產生的delay.c**
  - 含有delay函式
- **加一行delay減緩 LED 反向速度**
- **DELAY_milliseconds(500);**

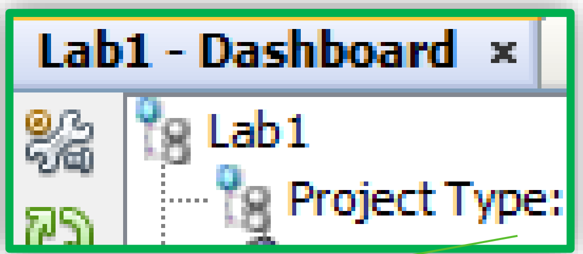


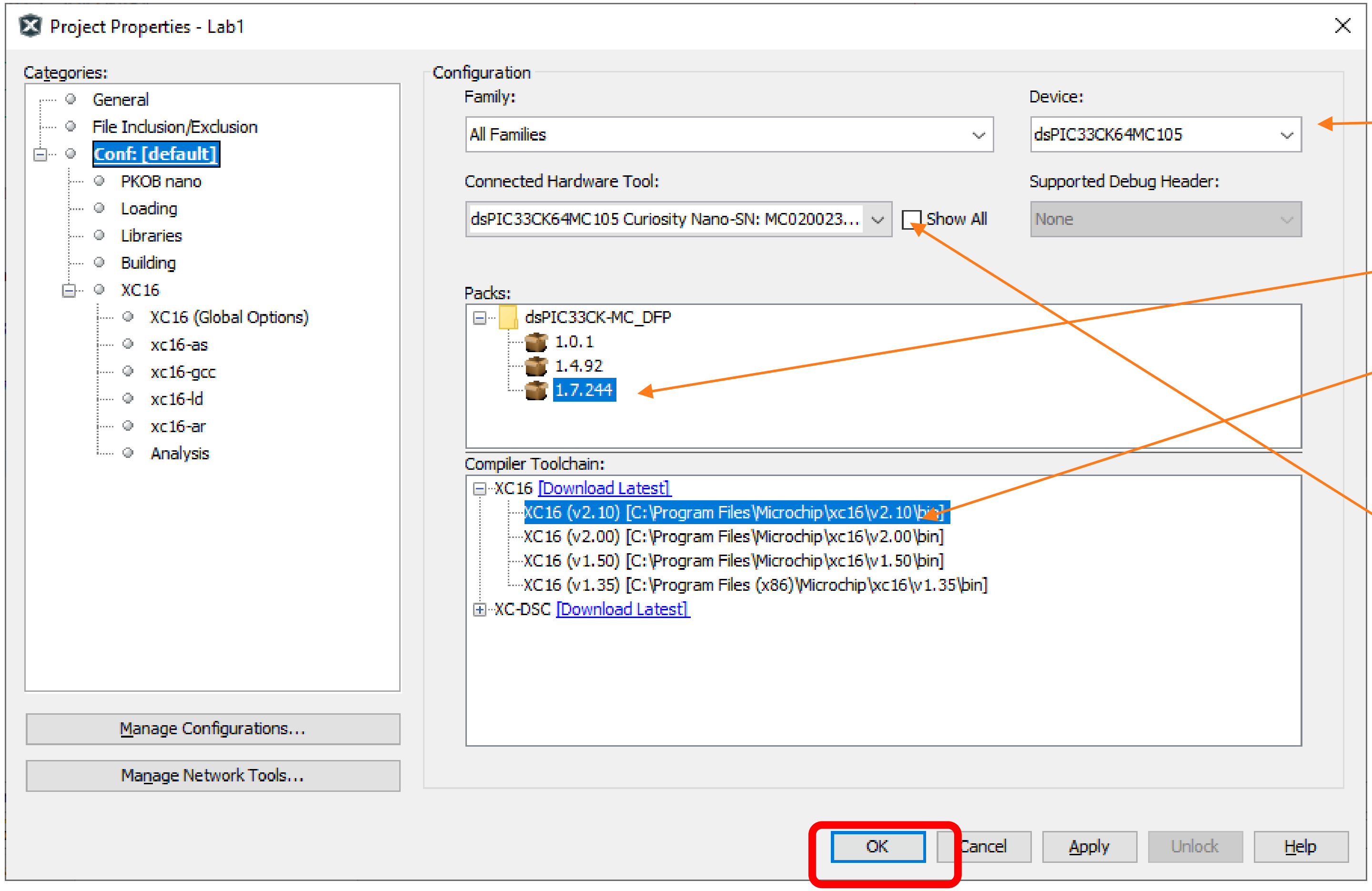
```
        Main application
*/

int main(void)
{
    SYSTEM_Initialize();

    while(1)
    {
        LED0_Toggle();
        DELAY_milliseconds(500);
    }
}
```

20.確認編譯環境(Dashboard)

Device\DFP\XC16\Debug Tool

1. 點選 ⚙ 後可修改





Project Properties - Lab1

Categories:

- General
- File Inclusion/Exclusion
- Conf: [default]
  - PKOB nano
  - Loading
  - Libraries
  - Building
  - XC16
    - XC16 (Global Options)
    - xc16-as
    - xc16-gcc
    - xc16-ld
    - xc16-ar
    - Analysis

Manage Configurations...

Manage Network Tools...

Configuration

Family: All Families

Device: dsPIC33CK64MC105

Connected Hardware Tool: dsPIC33CK64MC105 Curiosity Nano-SN: MC020023... ☐ Show All

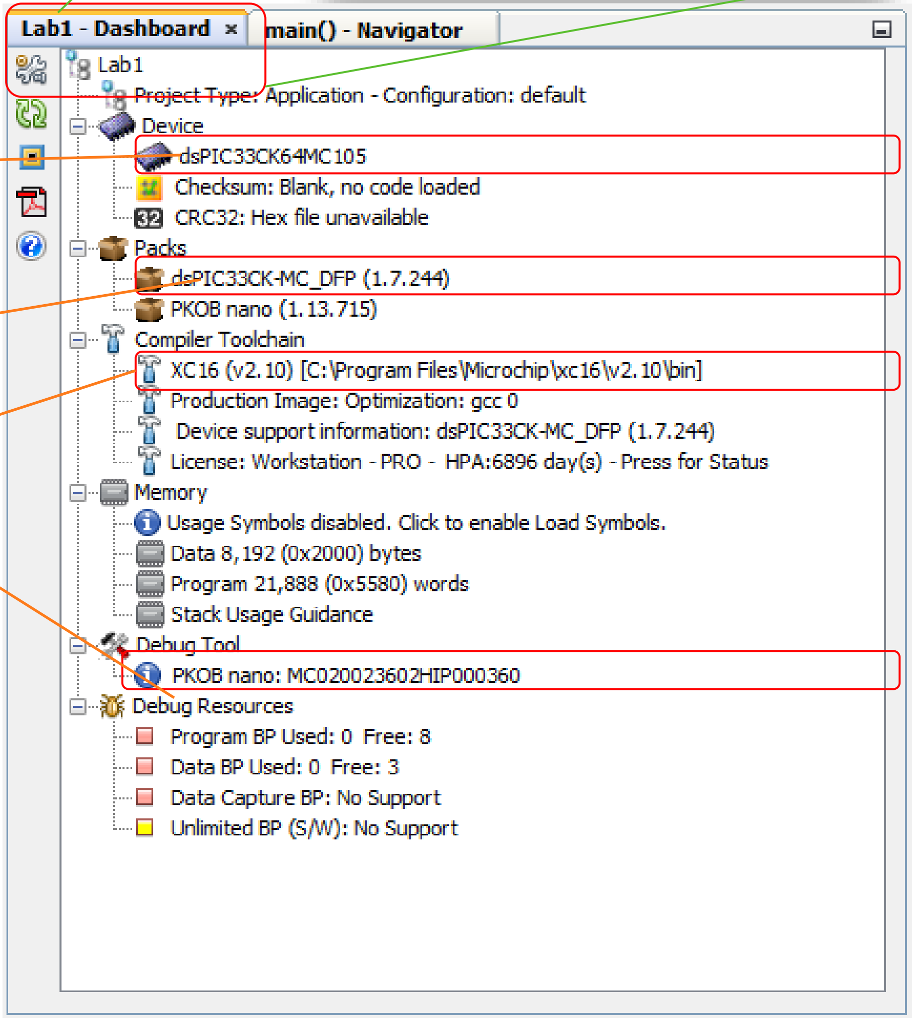
Supported Debug Header: None

Packs:

- dsPIC33CK-MC_DFP
  - 1.0.1
  - 1.4.92
  - 1.7.244

Compiler Toolchain:

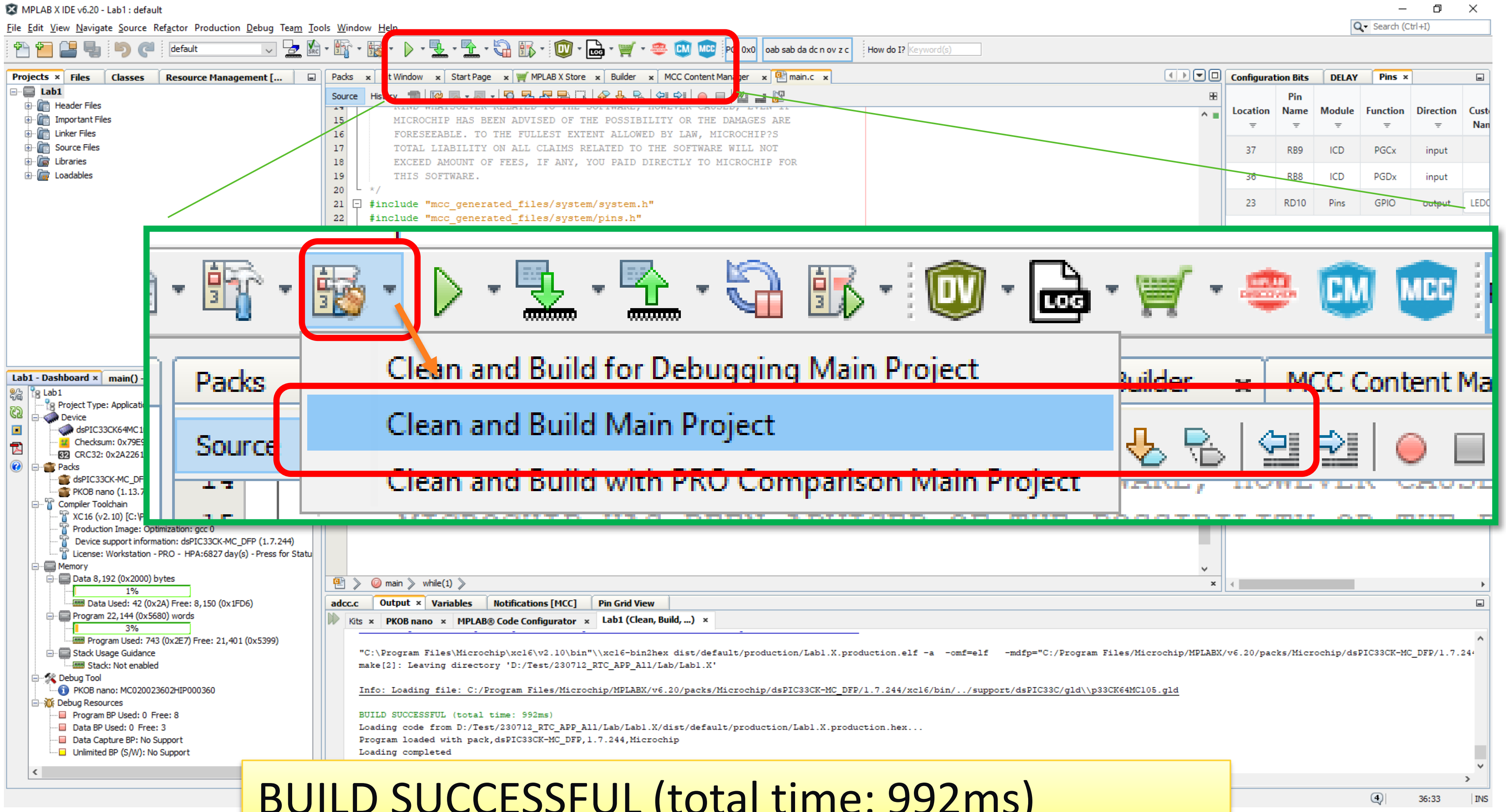
- XC16 [Download Latest]
  - XC16 (v2.10) [C:\Program Files\Microchip\xc16\v2.10\bin]
  - XC16 (v2.00) [C:\Program Files\Microchip\xc16\v2.00\bin]
  - XC16 (v1.50) [C:\Program Files\Microchip\xc16\v1.50\bin]
  - XC16 (v1.35) [C:\Program Files (x86)\Microchip\xc16\v1.35\bin]
- XC-DSC [Download Latest]



- Lab1
  - Project Type: Application - Configuration: default
  - Device
    - dsPIC33CK64MC105
    - Checksum: Blank, no code loaded
    - CRC32: Hex file unavailable
  - Packs
    - dsPIC33CK-MC_DFP (1.7.244)
    - PKOB nano (1.13.715)
  - Compiler Toolchain
    - XC16 (v2.10) [C:\Program Files\Microchip\xc16\v2.10\bin]
    - Production Image: Optimization: gcc 0
    - Device support information: dsPIC33CK-MC_DFP (1.7.244)
    - License: Workstation - PRO - HPA:6896 day(s) - Press for Status
  - Memory
    - Usage Symbols disabled. Click to enable Load Symbols.
    - Data 8,192 (0x2000) bytes
    - Program 21,888 (0x5580) words
    - Stack Usage Guidance
  - Debug Tool
    - PKOB nano: MC020023602HIP000360
  - Debug Resources
    - Program BP Used: 0 Free: 8
    - Data BP Used: 0 Free: 3
    - Data Capture BP: No Support
    - Unlimited BP (S/W): No Support

21.啟動編輯

Clean and Build Project(Lab1)



```
    MICROCHIP HAS BEEN ADVISED OF THE POSSIBILITY OR THE DAMAGES ARE
    FORESEEABLE. TO THE FULLEST EXTENT ALLOWED BY LAW, MICROCHIP?S
    TOTAL LIABILITY ON ALL CLAIMS RELATED TO THE SOFTWARE WILL NOT
    EXCEED AMOUNT OF FEES, IF ANY, YOU PAID DIRECTLY TO MICROCHIP FOR
    THIS SOFTWARE.
*/
#include "mcc_generated_files/system/system.h"
#include "mcc_generated_files/system/pins.h"
```

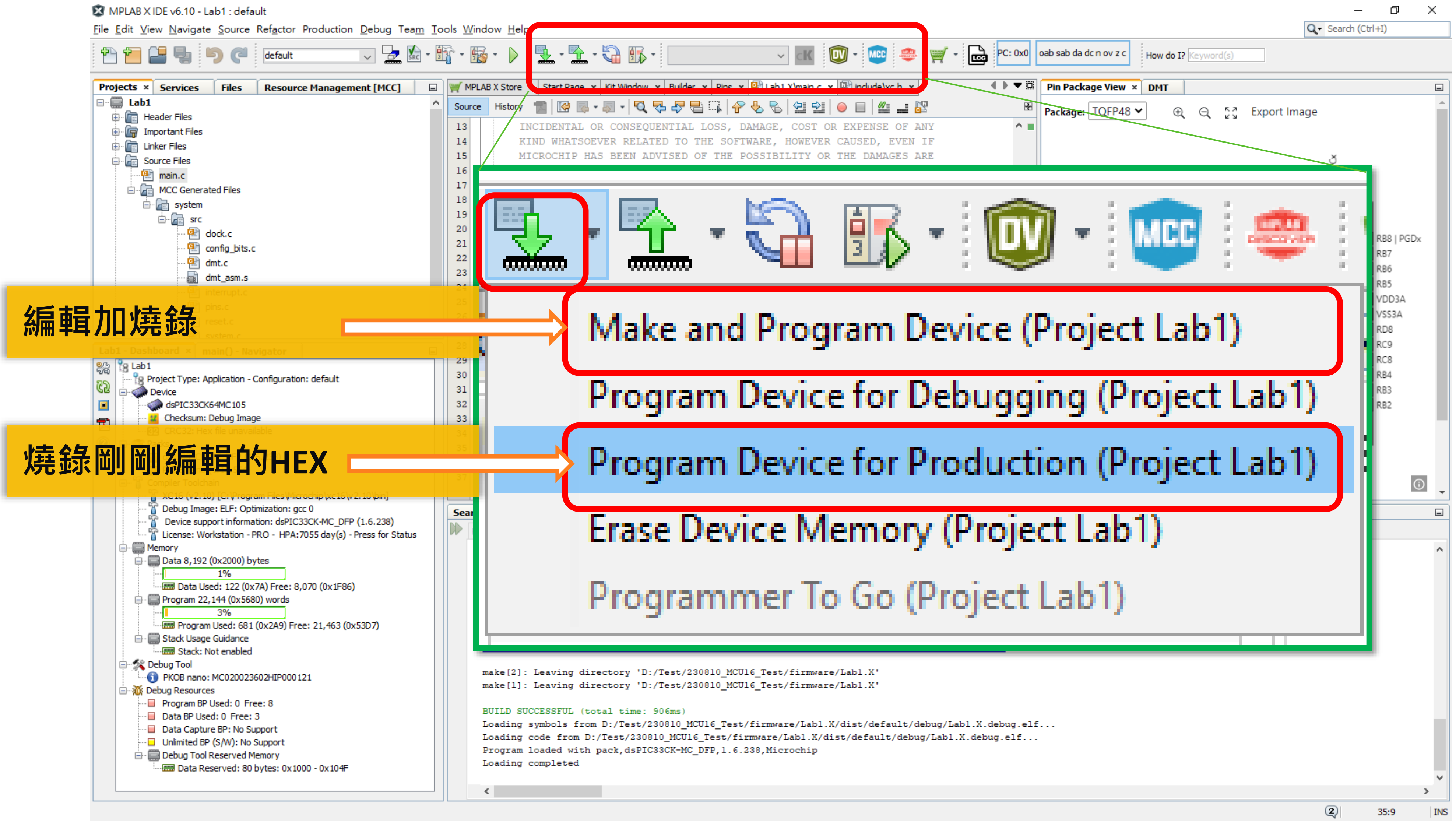
Location	Pin Name	Module	Function	Direction	Custom Name
37	RB9	ICD	PGCx	input	
36	RB8	ICD	PGDx	input	
23	RD10	Pins	GPIO	output	LEDC

Lab1 - Dashboard
- Lab1
  - Project Type: Applicati
  - Device
    - dsPIC33CK64MC1
    - Checksum: 0x79E
    - CRC32: 0x2A2261
  - Packs
    - dsPIC33CK-MC_DF
    - PKOB nano (1.13.7
  - Compiler Toolchain
    - XC16 (v2.10) [C:\P
    - Production Image: Optimization: gcc 0
    - Device support information: dsPIC33CK-MC_DFP (1.7.244)
    - License: Workstation - PRO - HPA:6827 day(s) - Press for Statu
  - Memory
    - Data 8,192 (0x2000) bytes
      - 1%
      - Data Used: 42 (0x2A) Free: 8,150 (0x1FD6)
    - Program 22,144 (0x5680) words
      - 3%
      - Program Used: 743 (0x2E7) Free: 21,401 (0x5399)
    - Stack Usage Guidance
      - Stack: Not enabled
  - Debug Tool
    - PKOB nano: MC020023602HIP000360
  - Debug Resources
    - Program BP Used: 0 Free: 8
    - Data BP Used: 0 Free: 3
    - Data Capture BP: No Support
    - Unlimited BP (S/W): No Support

```
"C:\Program Files\Microchip\xc16\v2.10\bin"\\xc16-bin2hex dist/default/production/Lab1.X.production.elf -a  -omf=elf   -mdfp="C:/Program Files/Microchip/MPLABX/v6.20/packs/Microchip/dsPIC33CK-MC_DFP/1.7.24
make[2]: Leaving directory 'D:/Test/230712_RTC_APP_All/Lab/Lab1.X'

Info: Loading file: C:/Program Files/Microchip/MPLABX/v6.20/packs/Microchip/dsPIC33CK-MC_DFP/1.7.244/xc16/bin/../support/dsPIC33C/gld\\p33CK64MC105.gld

BUILD SUCCESSFUL (total time: 992ms)
Loading code from D:/Test/230712_RTC_APP_All/Lab/Lab1.X/dist/default/production/Lab1.X.production.hex...
Program loaded with pack,dsPIC33CK-MC_DFP,1.7.244,Microchip
Loading completed
```

BUILD SUCCESSFUL (total time: 992ms)

22.啟動燒錄



編輯加燒錄

燒錄剛剛編輯的HEX

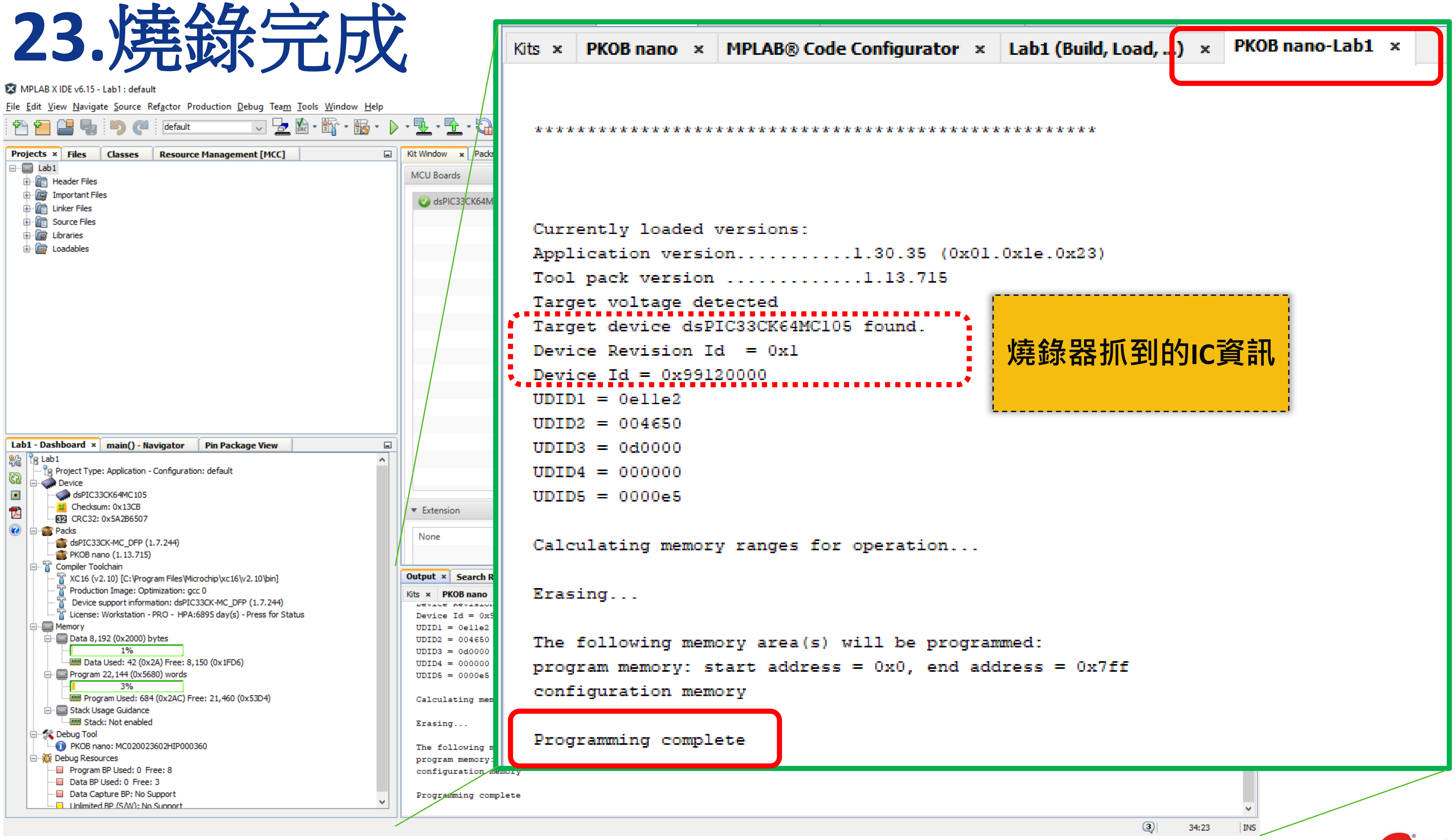
- Make and Program Device (Project Lab1)
- Program Device for Debugging (Project Lab1)
- Program Device for Production (Project Lab1)
- Erase Device Memory (Project Lab1)
- Programmer To Go (Project Lab1)

Lab1
- Project Type: Application - Configuration: default
- Device
  - dsPIC33CK64MC105
  - Checksum: Debug Image
  - Debug Image: ELF: Optimization: gcc 0
  - Device support information: dsPIC33CK-MC_DFP (1.6.238)
  - License: Workstation - PRO - HPA:7055 day(s) - Press for Status
- Memory
  - Data 8,192 (0x2000) bytes
    - 1%
    - Data Used: 122 (0x7A) Free: 8,070 (0x1F86)
  - Program 22,144 (0x5680) words
    - 3%
    - Program Used: 681 (0x2A9) Free: 21,463 (0x53D7)
  - Stack Usage Guidance
    - Stack: Not enabled
- Debug Tool
  - PKOB nano: MC020023602HIP000121
- Debug Resources
  - Program BP Used: 0 Free: 8
  - Data BP Used: 0 Free: 3
  - Data Capture BP: No Support
  - Unlimited BP (S/W): No Support
  - Debug Tool Reserved Memory
    - Data Reserved: 80 bytes: 0x1000 - 0x104F

```
INCIDENTAL OR CONSEQUENTIAL LOSS, DAMAGE, COST OR EXPENSE OF ANY
KIND WHATSOEVER RELATED TO THE SOFTWARE, HOWEVER CAUSED, EVEN IF
MICROCHIP HAS BEEN ADVISED OF THE POSSIBILITY OR THE DAMAGES ARE
```

```
make[2]: Leaving directory 'D:/Test/230810_MCU16_Test/firmware/Lab1.X'
make[1]: Leaving directory 'D:/Test/230810_MCU16_Test/firmware/Lab1.X'

BUILD SUCCESSFUL (total time: 906ms)
Loading symbols from D:/Test/230810_MCU16_Test/firmware/Lab1.X/dist/default/debug/Lab1.X.debug.elf...
Loading code from D:/Test/230810_MCU16_Test/firmware/Lab1.X/dist/default/debug/Lab1.X.debug.elf...
Program loaded with pack,dsPIC33CK-MC_DFP,1.6.238,Microchip
Loading completed
```

23.燒錄完成



MPLAB X IDE v6.15 - Lab1 : default
File Edit View Navigate Source Refactor Production Debug Team Tools Window Help
default
Projects × Files Classes Resource Management [MCC]
Lab1
Header Files
Important Files
Linker Files
Source Files
Libraries
Loadables
Kit Window × Packs
MCU Boards
dsPIC33CK64M
Extension
None
Lab1 - Dashboard × main() - Navigator Pin Package View
Output × Search R
Kits × PKOB nano
Kits × PKOB nano × MPLAB® Code Configurator × Lab1 (Build, Load, ...) × PKOB nano-Lab1 ×
34:23 INS

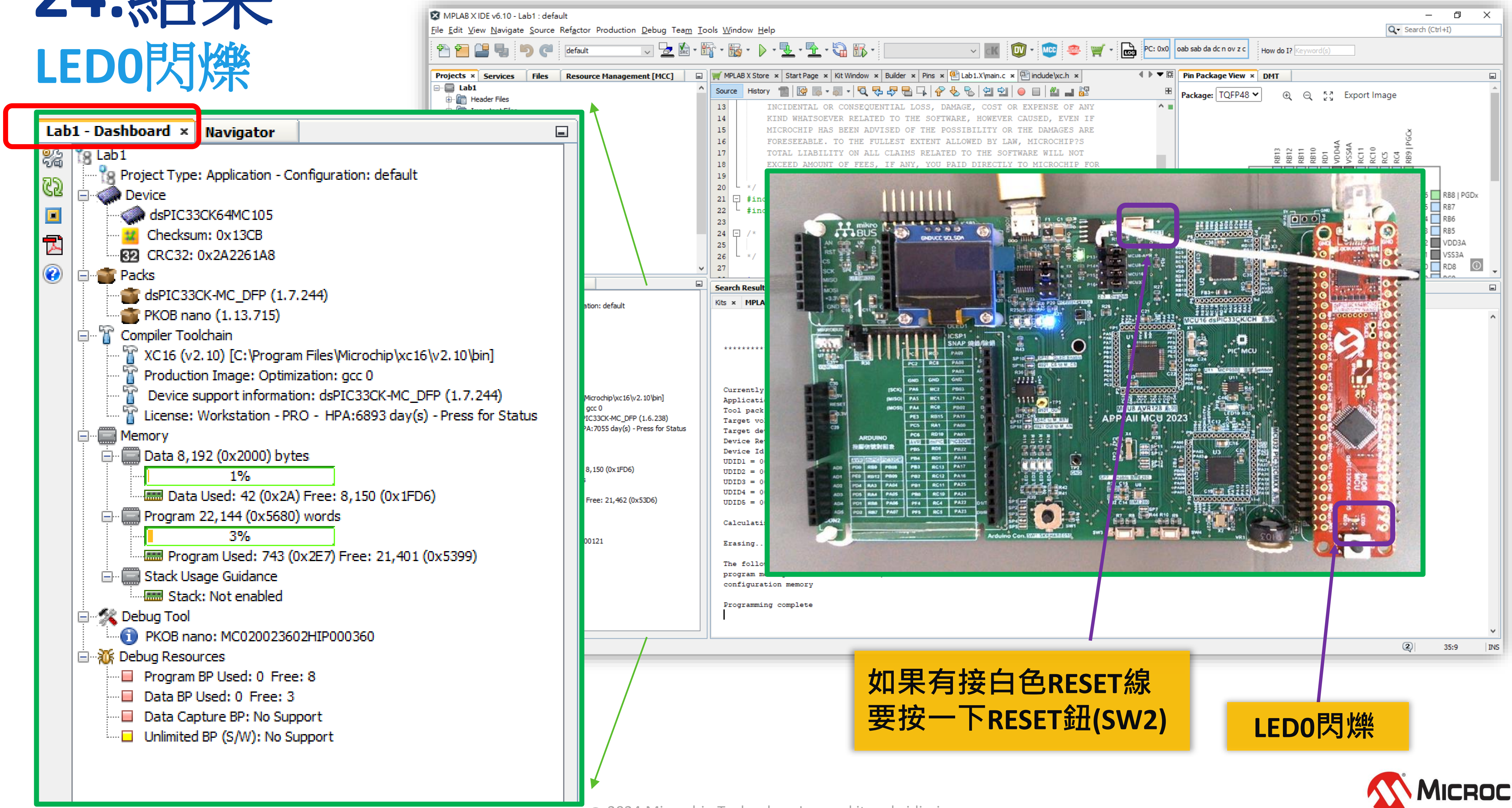
Lab1
Project Type: Application - Configuration: default
Device
dsPIC33CK64MC105
Checksum: 0x13CB
CRC32: 0x5A2B6507
Packs
dsPIC33CK-MC_DFP (1.7.244)
PKOB nano (1.13.715)
Compiler Toolchain
XC16 (v2.10) [C:\Program Files\Microchip\xc16\v2.10\bin]
Production Image: Optimization: gcc 0
Device support information: dsPIC33CK-MC_DFP (1.7.244)
License: Workstation - PRO - HPA:6895 day(s) - Press for Status
Memory
Data 8,192 (0x2000) bytes
1%
Data Used: 42 (0x2A) Free: 8,150 (0x1FD6)
Program 22,144 (0x5680) words
3%
Program Used: 684 (0x2AC) Free: 21,460 (0x53D4)
Stack Usage Guidance
Stack: Not enabled
Debug Tool
PKOB nano: MC020023602HIP000360
Debug Resources
Program BP Used: 0 Free: 8
Data BP Used: 0 Free: 3
Data Capture BP: No Support
Unlimited BP (S/W): No Support

```
*****************************************************

Currently loaded versions:
Application version...........1.30.35 (0x01.0x1e.0x23)
Tool pack version .............1.13.715
Target voltage detected
Target device dsPIC33CK64MC105 found.
Device Revision Id  = 0x1
Device Id = 0x99120000
UDID1 = 0e1le2
UDID2 = 004650
UDID3 = 0d0000
UDID4 = 000000
UDID5 = 0000e5

Calculating memory ranges for operation...

Erasing...

The following memory area(s) will be programmed:
program memory: start address = 0x0, end address = 0x7ff
configuration memory

Programming complete
```

燒錄器抓到的IC資訊

24.結果

LED0閃爍



MPLAB X IDE v6.10 - Lab1 : default
File Edit View Navigate Source Refactor Production Debug Team Tools Window Help
Lab1 - Dashboard
Navigator
Pin Package View
Package: TQFP48
Export Image
Programming complete

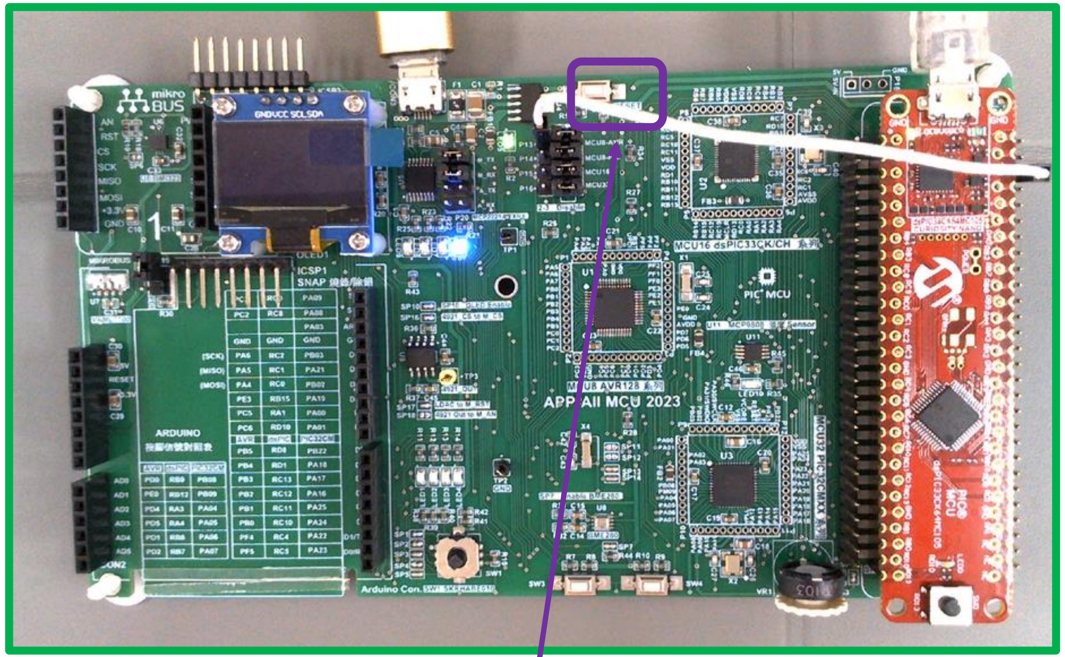
Lab1
- Project Type: Application - Configuration: default
- Device
  - dsPIC33CK64MC105
  - Checksum: 0x13CB
  - CRC32: 0x2A2261A8
- Packs
  - dsPIC33CK-MC_DFP (1.7.244)
  - PKOB nano (1.13.715)
- Compiler Toolchain
  - XC16 (v2.10) [C:\Program Files\Microchip\xc16\v2.10\bin]
  - Production Image: Optimization: gcc 0
  - Device support information: dsPIC33CK-MC_DFP (1.7.244)
  - License: Workstation - PRO - HPA:6893 day(s) - Press for Status
- Memory
  - Data 8,192 (0x2000) bytes
    - 1%
    - Data Used: 42 (0x2A) Free: 8,150 (0x1FD6)
  - Program 22,144 (0x5680) words
    - 3%
    - Program Used: 743 (0x2E7) Free: 21,401 (0x5399)
  - Stack Usage Guidance
    - Stack: Not enabled
- Debug Tool
  - PKOB nano: MC020023602HIP000360
- Debug Resources
  - Program BP Used: 0 Free: 8
  - Data BP Used: 0 Free: 3
  - Data Capture BP: No Support
  - Unlimited BP (S/W): No Support

如果有接白色RESET線
要按一下RESET鈕(SW2)

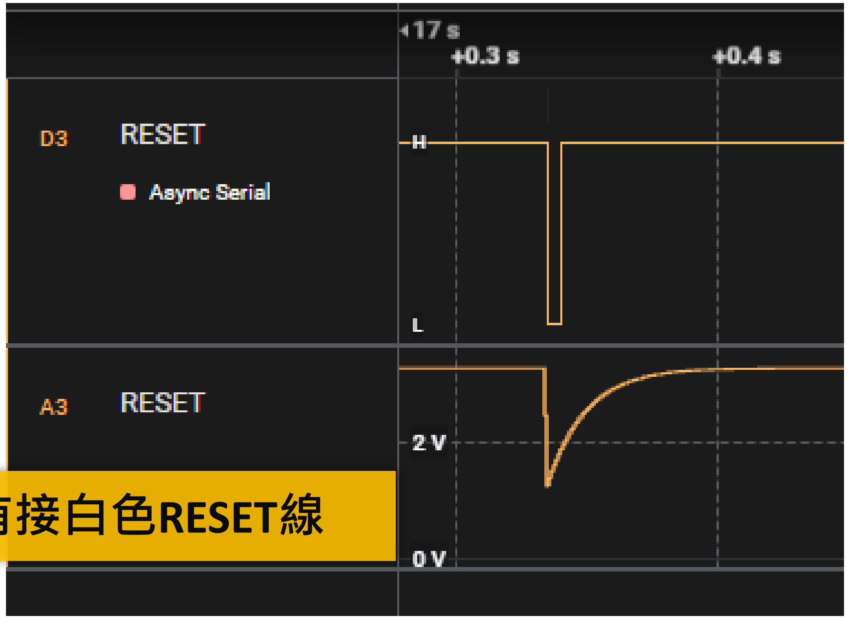
LED0閃爍

25. RESET

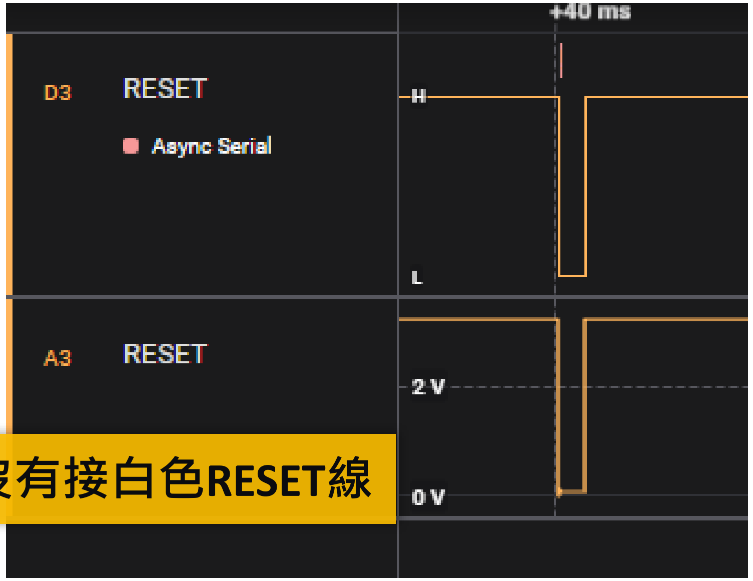
要再按一次的原因



因紅板內建燒錄器
RESET Pin驅動能力不足驅動綠板
燒完後，需要按一下RESET鈕(SW2)
讓MCU Reset 完全



有接白色RESET線



沒有接白色RESET線

26.成果

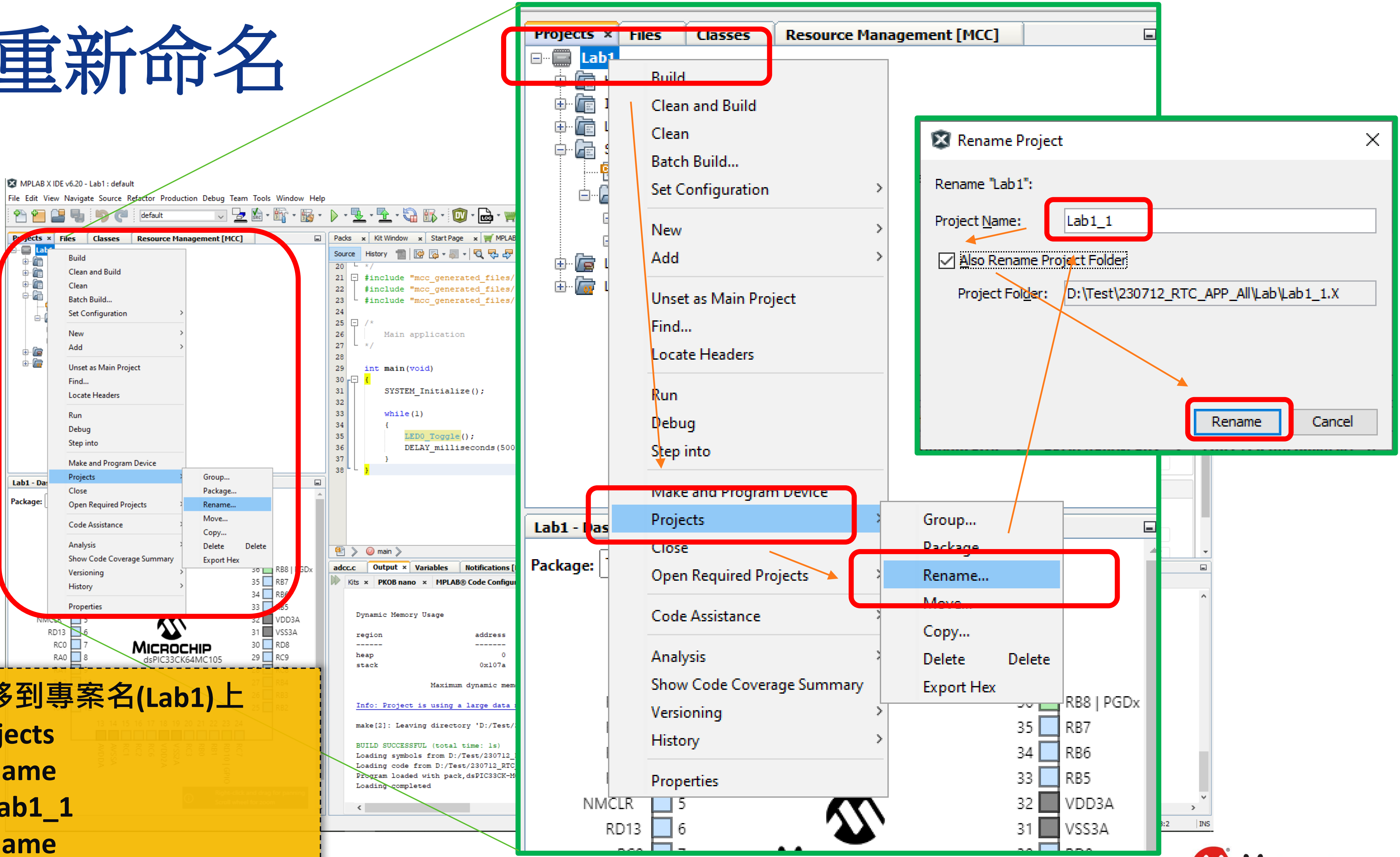
- 以上步驟完成，看到**LED**閃爍代表
  - MPLAB X IDE → 安裝OK
  - MCC → 安裝OK
  - XC16 → 安裝OK

  - EVB(紅板) → 燒錄功能 OK
  - dsPIC33 → 型號正確 OK

  - 可以開始進一步練習

專案重新命名

Lab1_1



```
#include "mcc_generated_files/
#include "mcc_generated_files/
#include "mcc_generated_files/

/*
    Main application
*/

int main(void)
{
    SYSTEM_Initialize();

    while(1)
    {
        LED0_Toggle();
        DELAY_milliseconds(500
    }
}
```

```
Dynamic Memory Usage

region                         address
------                         -------
heap                                 0
stack                           0x107a

        Maximum dynamic mem

Info: Project is using a large data

make[2]: Leaving directory 'D:/Test/

BUILD SUCCESSFUL (total time: 1s)
Loading symbols from D:/Test/230712_
Loading code from D:/Test/230712_RTC
Program loaded with pack,dsPIC33CK-M
Loading completed
```

Rename "Lab1":

Project Name: Lab1_1

☑ Also Rename Project Folder

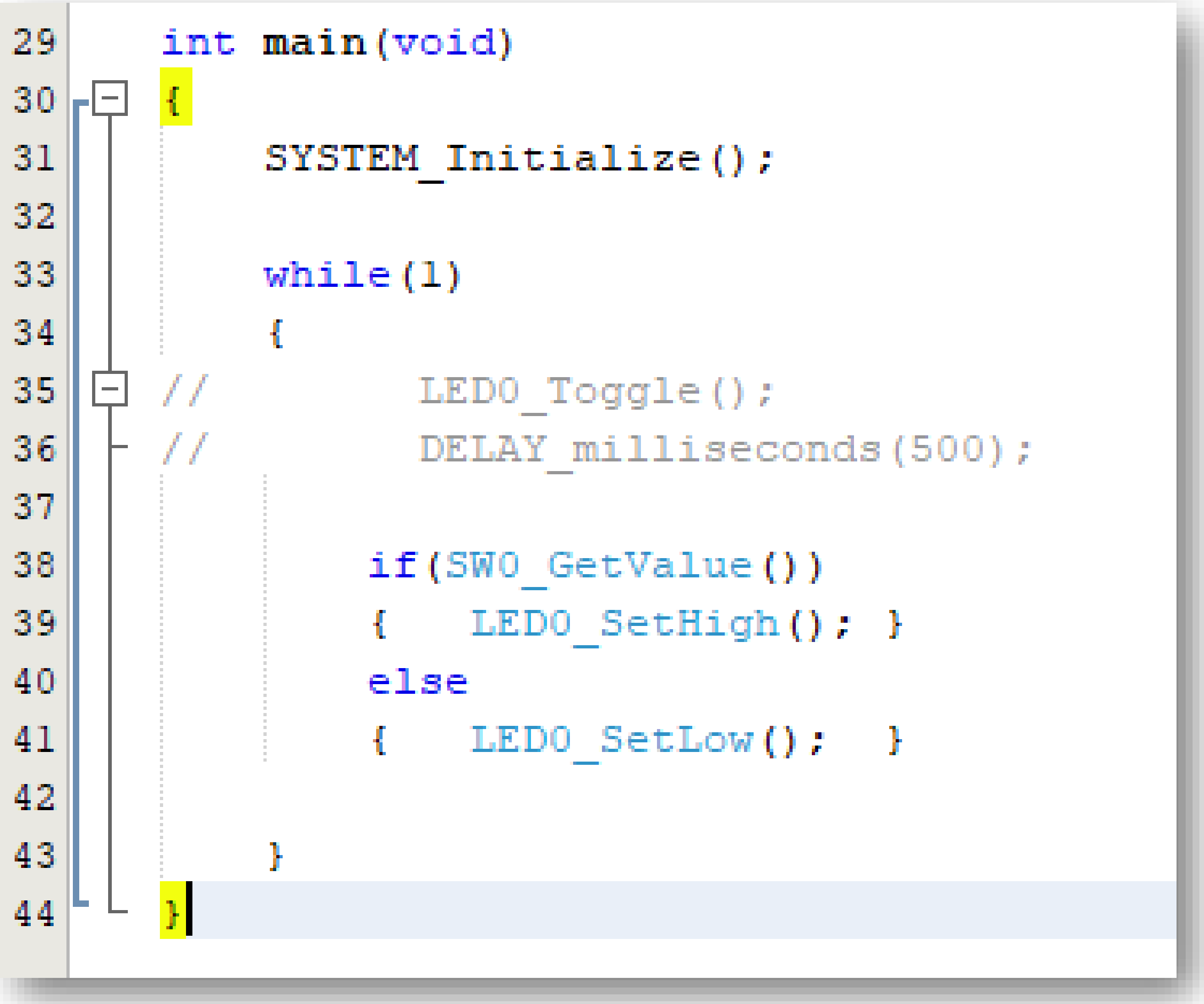
Project Folder: D:\Test\230712_RTC_APP_All\Lab\Lab1_1.X

1. 滑鼠移到專案名(Lab1)上
2. 選Projects
3. 選Rename
4. 改為Lab1_1
5. 按Rename

測試input

Lab1_1 透過SW0控制LED0

1) 延續Lab1
2) 開啟MCC
3) 設定輸入(參考步驟14,15)
   a) Project Resource→點Pins
   b) 在Pins /input處→點RD13
   c) IO_RD13改名成SW0
   d) 並在同一行，勾選Weak Pullup
   e) 在產生一次Code
4) 改Code (如右圖) (參考步驟17)
   到main.c 的while(1){…}
5) 燒入(參考步驟22)
6) 結果: 按SW0 -> LED0 亮



```
int main(void)
{
    SYSTEM_Initialize();

    while(1)
    {
//        LED0_Toggle();
//        DELAY_milliseconds(500);

        if(SW0_GetValue())
        {   LED0_SetHigh(); }
        else
        {   LED0_SetLow();  }

    }
}
```

實驗二

透過UART(非同步串列通訊)顯示信息

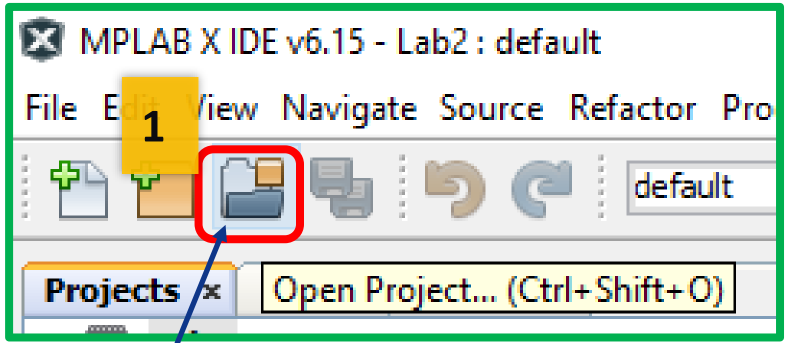
入門實驗

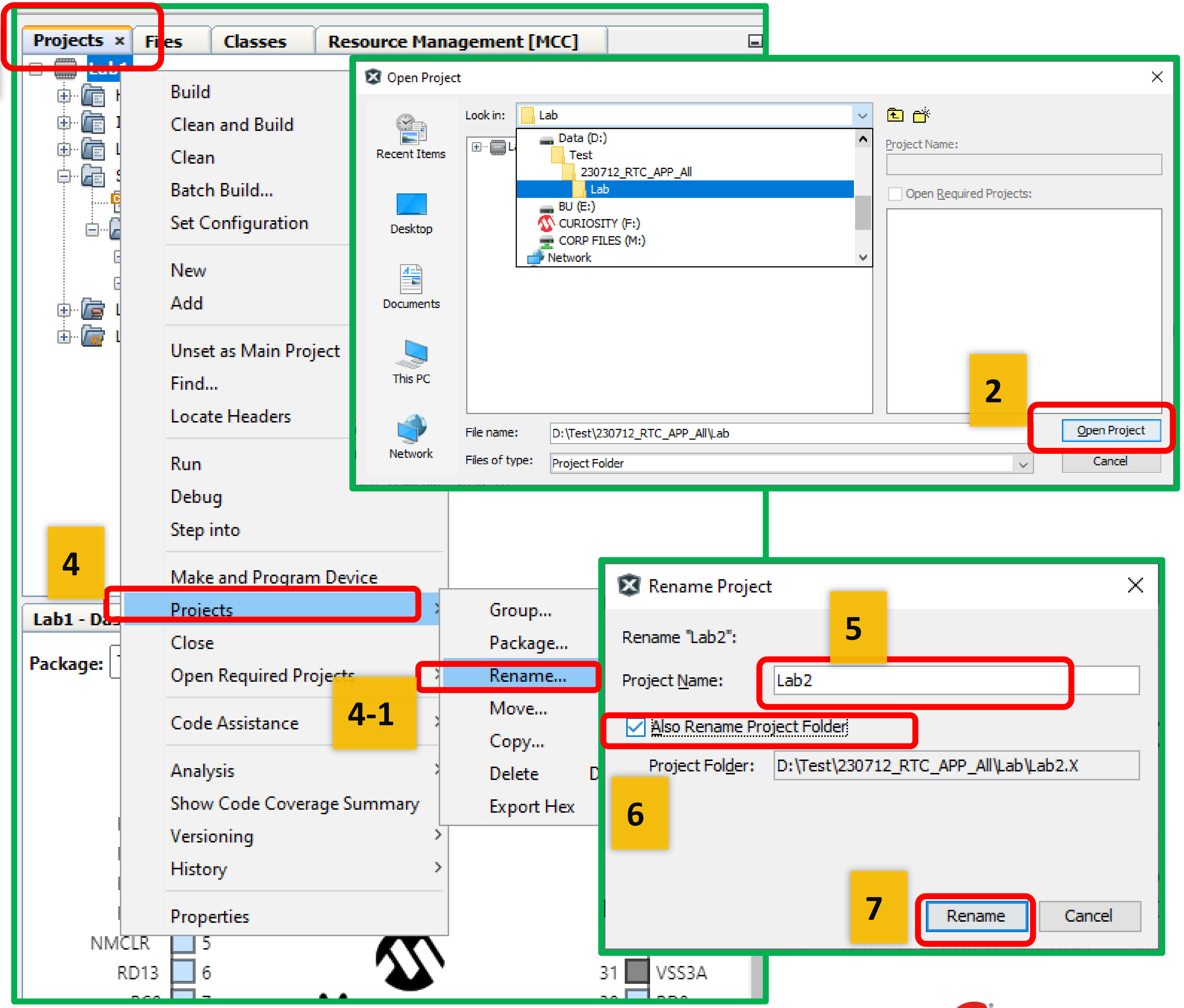
實驗二：透過UART(非同步串列通訊)顯示信息

- 本實驗目的：
  - 確認底板正常
    - 訊號可以通過 115200-8-N-1格式傳到底板
    - MCP2221動作正常
    - 方便後續輸出除錯訊息
- 結果呈現：
  - 在電腦終端機上顯示字串
    - dsPIC33: Hi

1.建立Lab2

將Lab 1改名成Lab2


MPLAB X IDE v6.15 - Lab2 : default
File
Edit
View
Navigate
Source
Refactor
Pro
default
Projects
Open Project... (Ctrl+Shift+O)
1



Look in: Lab

File name: D:\Test\230712_RTC_APP_All\Lab

Files of type: Project Folder

Rename "Lab2":

Project Name: Lab2

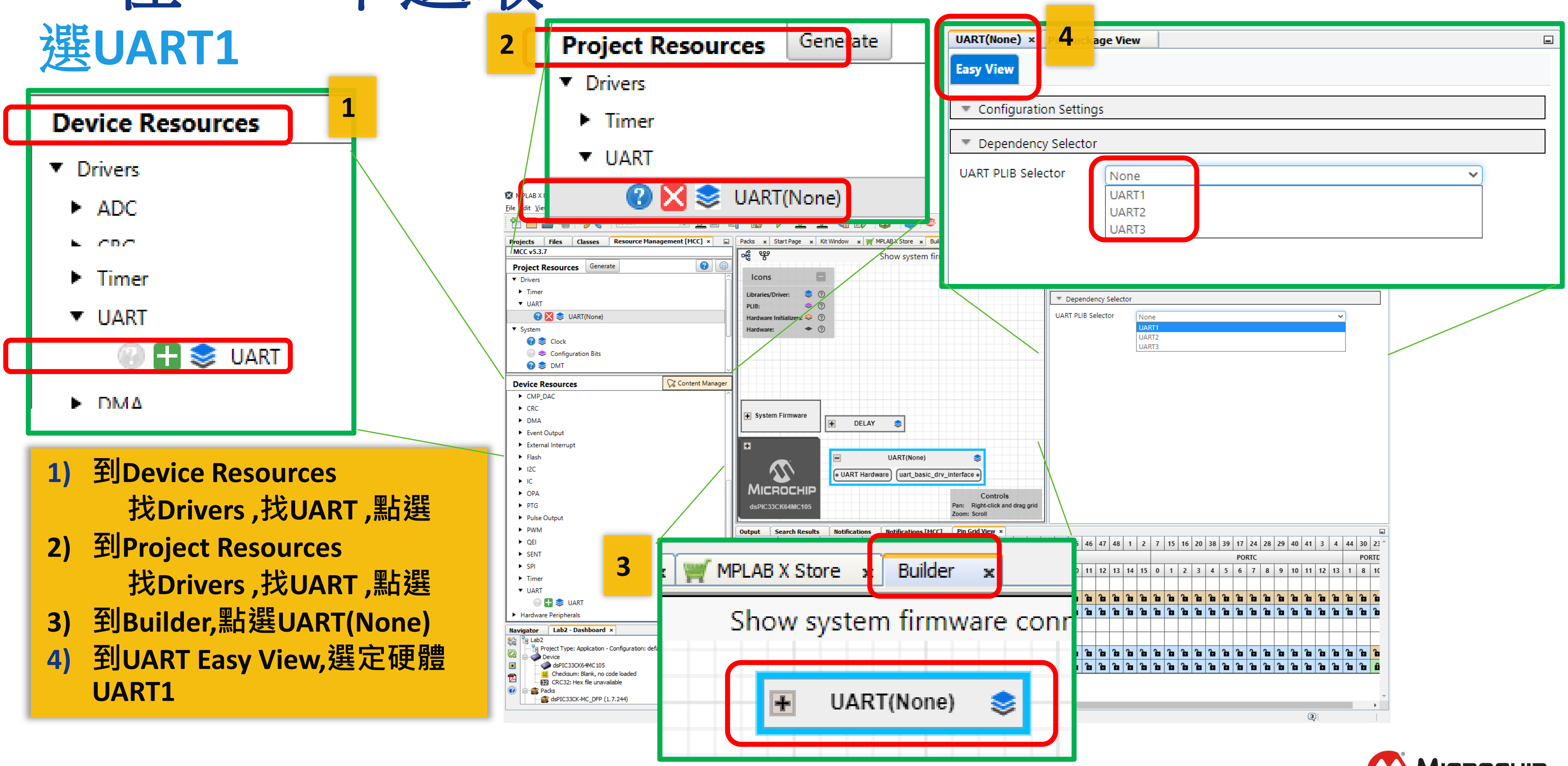
☑ Also Rename Project Folder

Project Folder: D:\Test\230712_RTC_APP_All\Lab\Lab2.X

1) 點選 →開啟專案
2) 到對映路徑中選取→Lab1
3) 到Projects 在Lab1上
   滑鼠點右鍵
4) 找到Projects → Rename
5) 修改成Lab2
6) 勾選順便改資料夾名
7) 按 Rename 完成名稱修改

2. 在MCC中選取UART

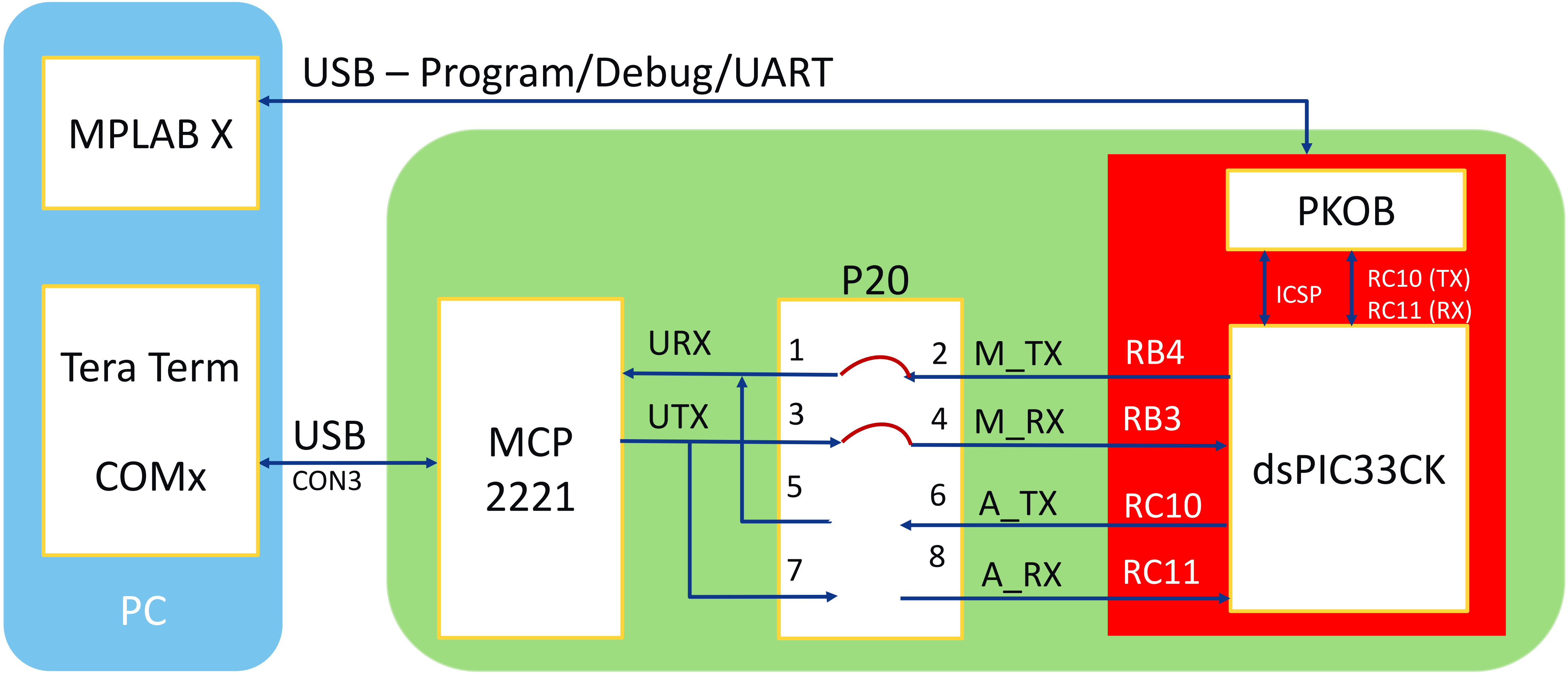
選UART1



1) 到Device Resources
  找Drivers ,找UART ,點選
2) 到Project Resources
  找Drivers ,找UART ,點選
3) 到Builder,點選UART(None)
4) 到UART Easy View,選定硬體 UART1

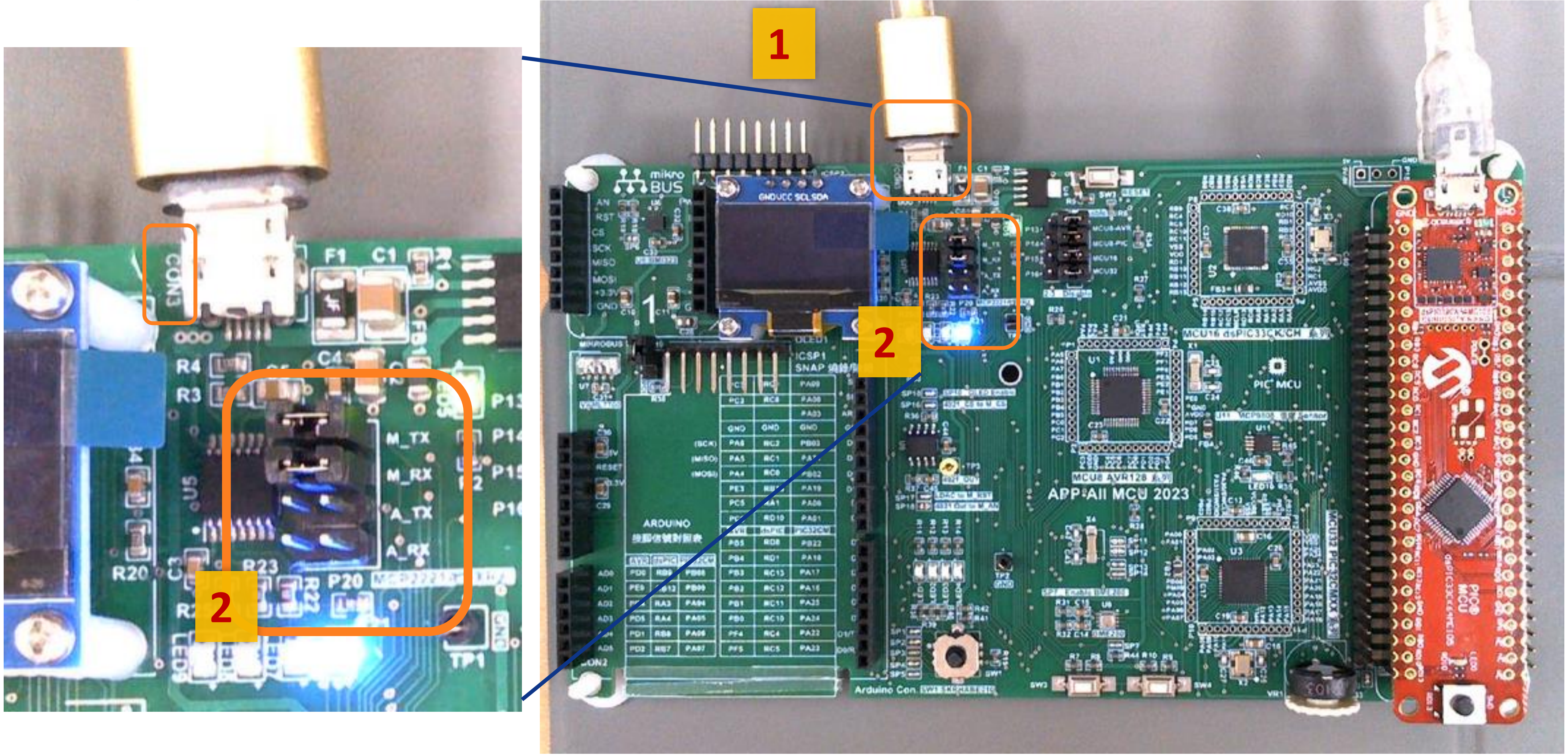
3.架構圖

UART 相關連線



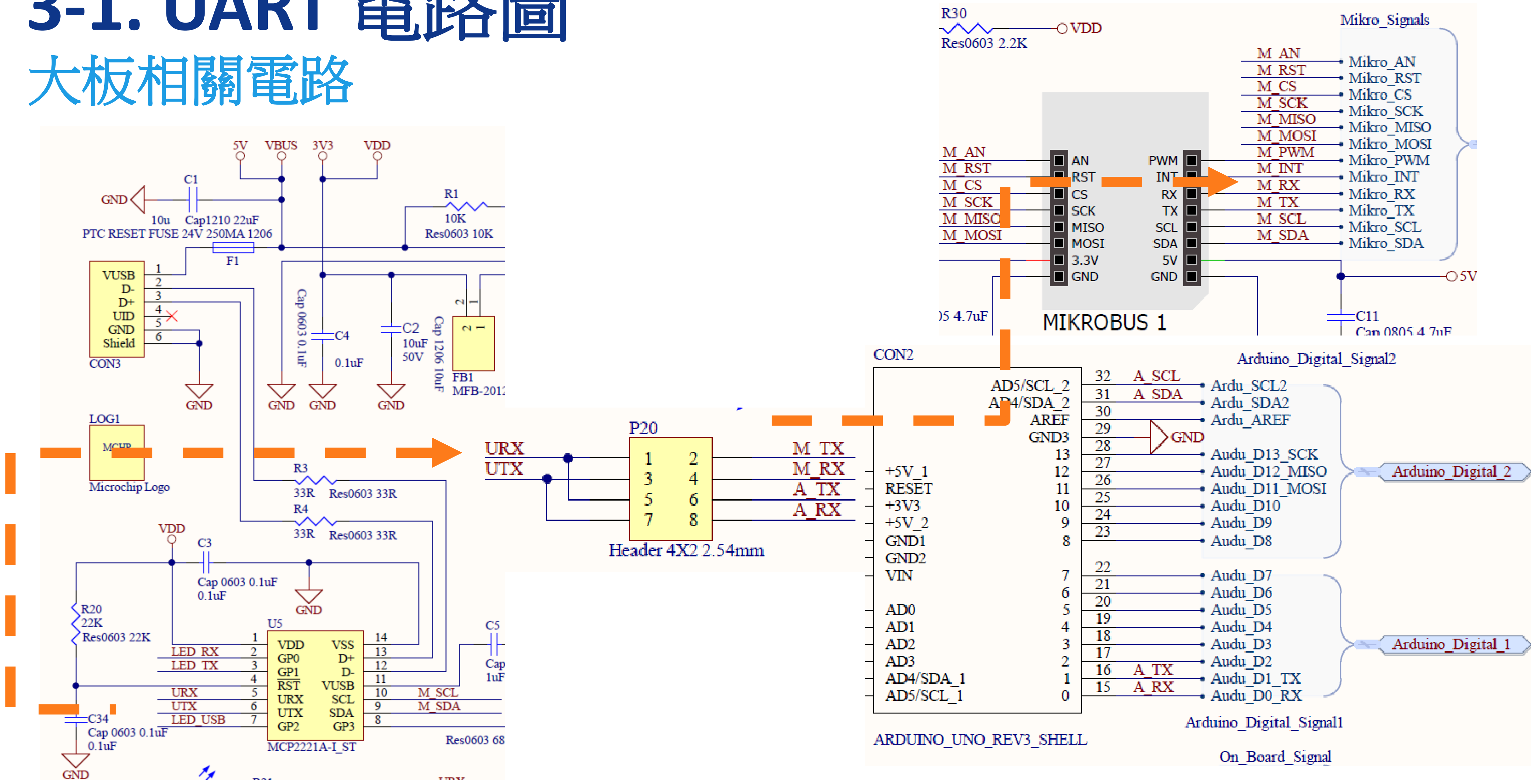
3-2. UART 相關接線

P20-Jump 位置



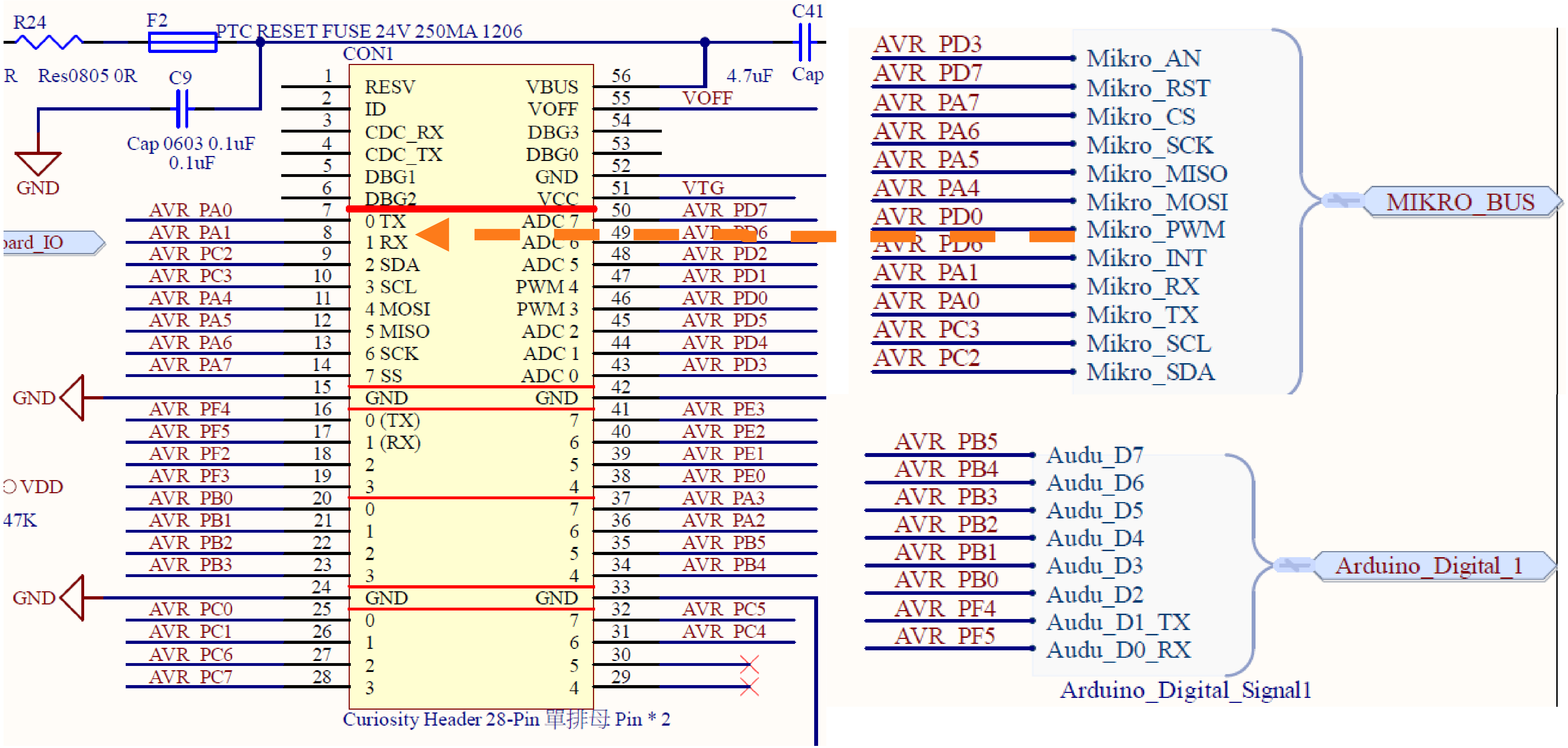
3-1. UART 電路圖

大板相關電路



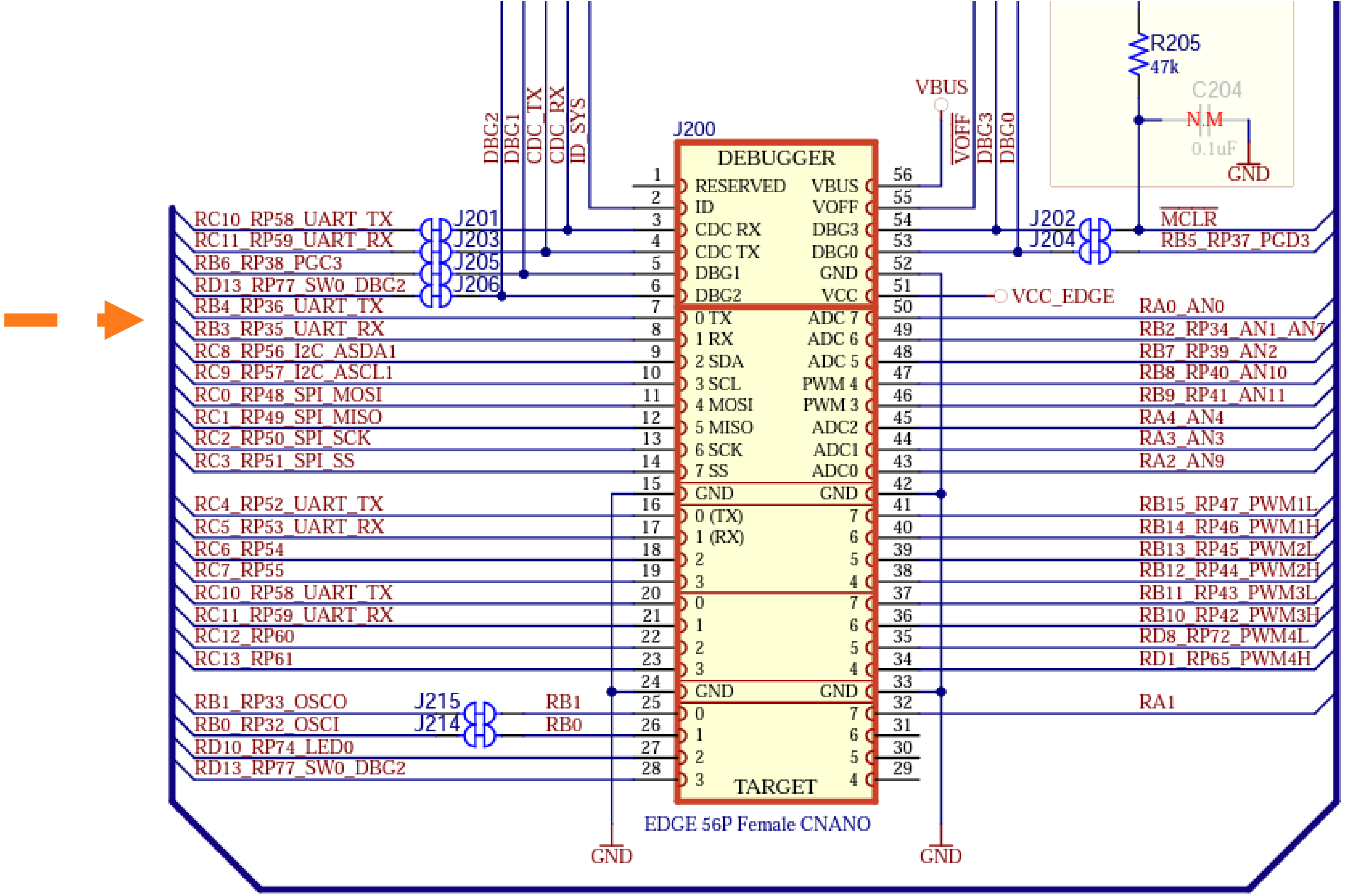
3-2. UART電路圖

大板相關電路



3-3. UART電路圖

小板相關電路

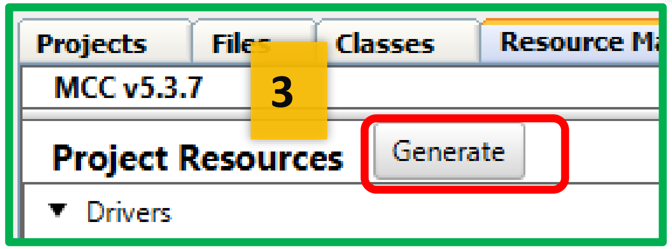


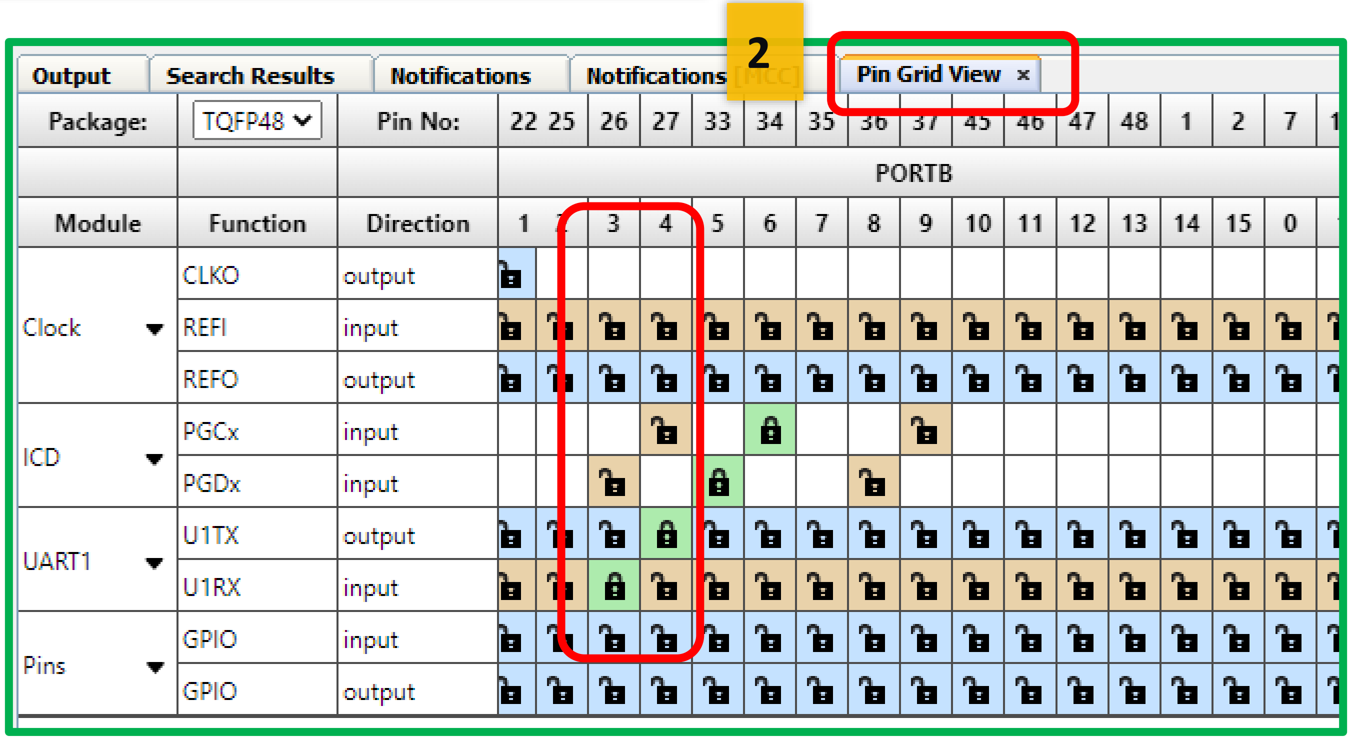
MICROCHIP

4.確認UART腳位

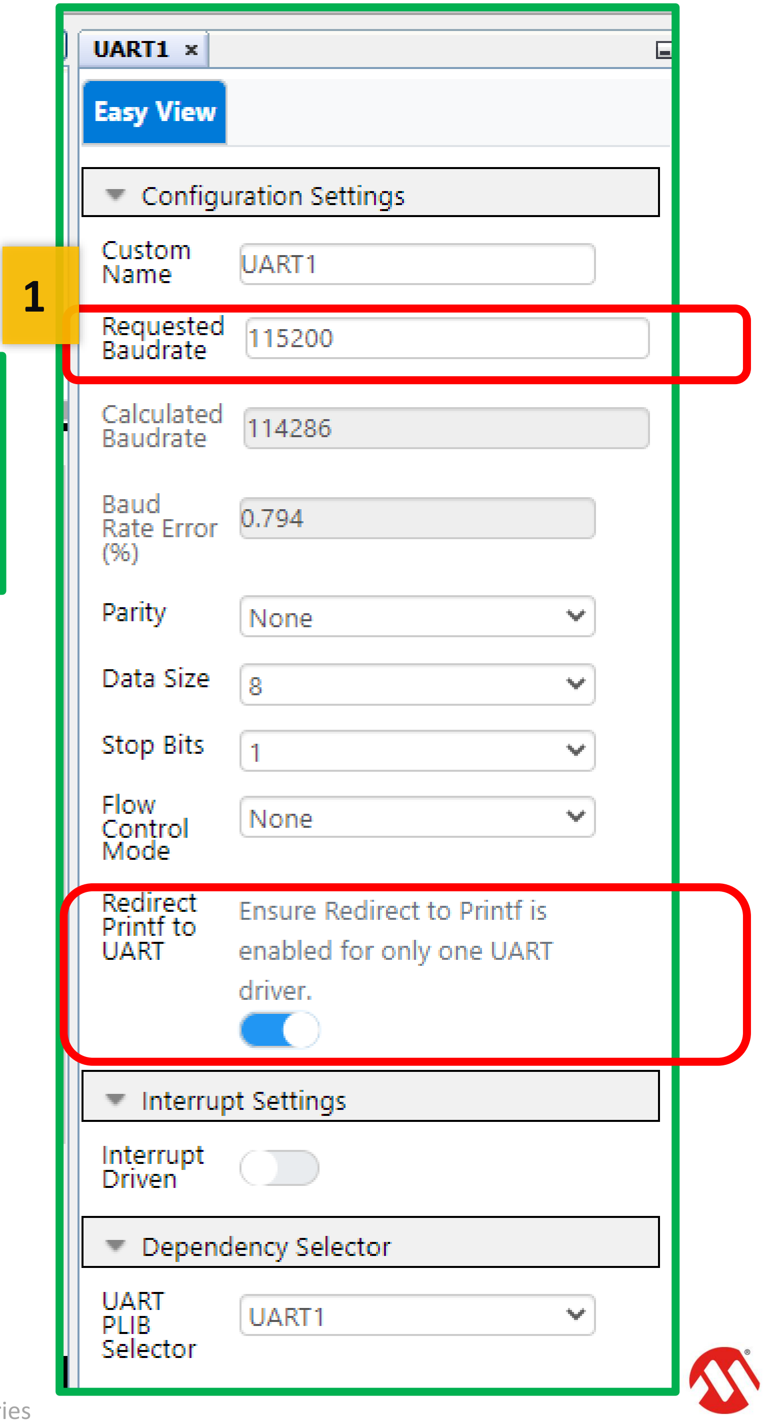
設腳位RB4 TX, RB3 RX

1) 到UART1, Easy View
   改115200-8-N-1
   Redirect printf to UART
2) 到Pin Grid View
   設RB4 TX, RB3 RX
3) 產生Code





Package:	TQFP48	Pin No:	22	25	26	27	33	34	35	36	37	45	46	47	48	1	2	7
			PORTB															
Module	Function	Direction	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	0
Clock ▼	CLKO	output	🔓															
	REFI	input	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓
	REFO	output	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓
ICD ▼	PGCx	input				🔓		🔒			🔓							
	PGDx	input			🔓		🔒			🔓								
UART1 ▼	U1TX	output	🔓	🔓	🔓	🔒	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓
	U1RX	input	🔓	🔓	🔒	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓
Pins ▼	GPIO	input	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓
	GPIO	output	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓	🔓



▼ Configuration Settings

Custom Name	UART1
Requested Baudrate	115200
Calculated Baudrate	114286
Baud Rate Error (%)	0.794
Parity	None
Data Size	8
Stop Bits	1
Flow Control Mode	None
Redirect Printf to UART	Ensure Redirect to Printf is enabled for only one UART driver. ●

▼ Interrupt Settings

Interrupt Driven	○

▼ Dependency Selector

UART PLIB Selector	UART1

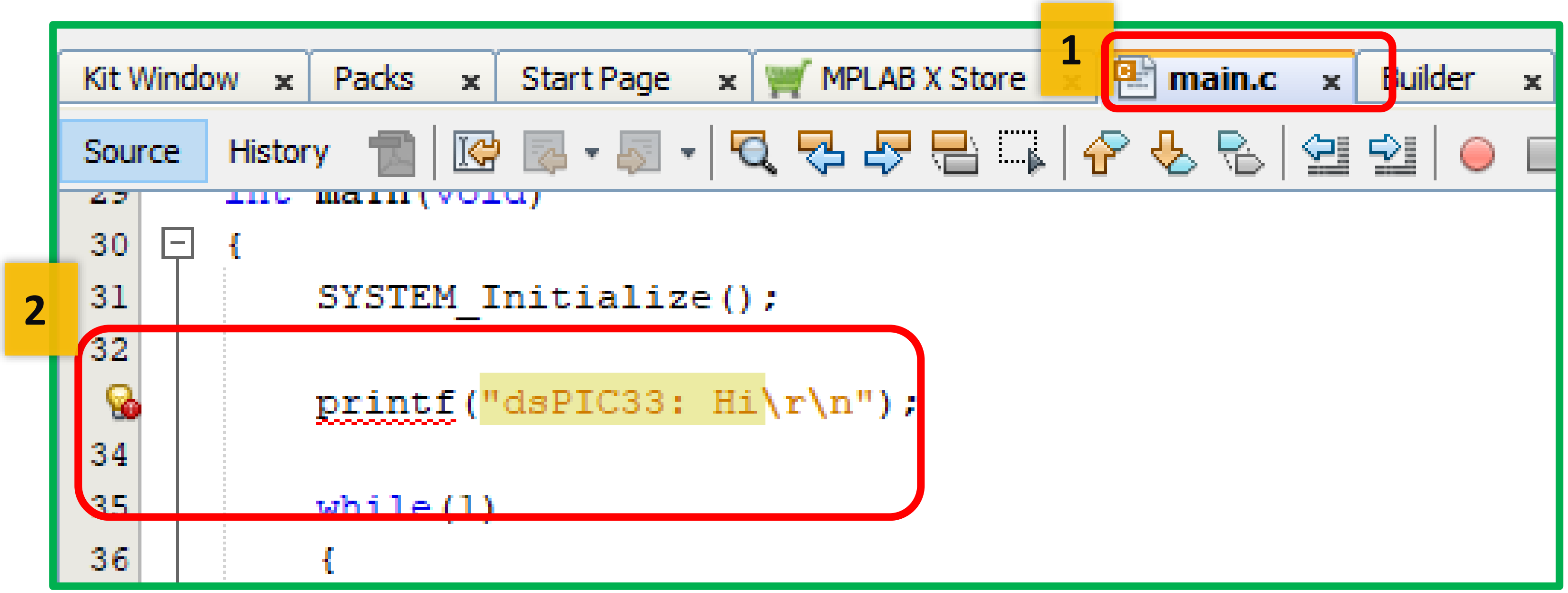
5.寫程式

printf(“Hi”);

1) 到main.c

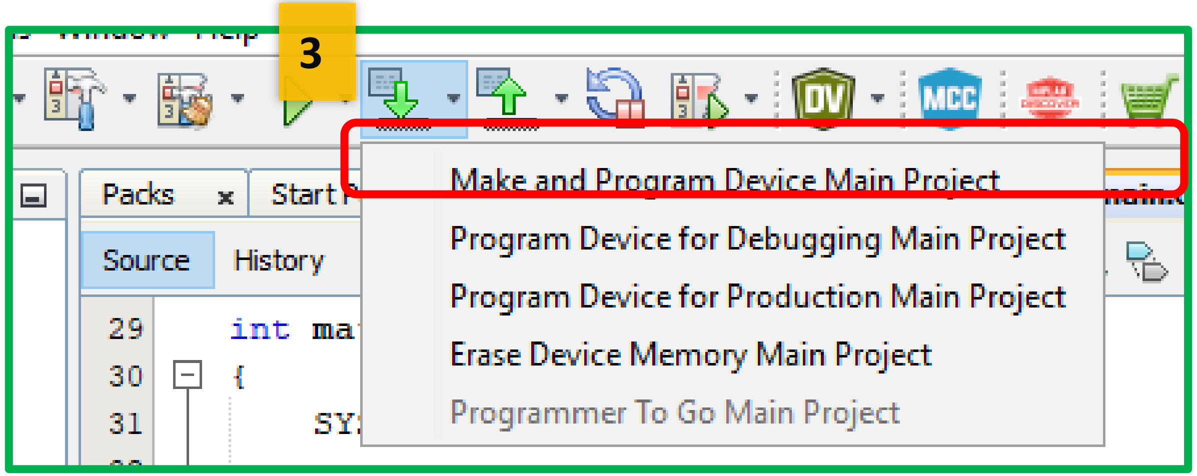
2) 到while(1); 上一行
寫 printf("dsPIC33: Hi\r\n");

3) 燒錄


1

2

```
30  {
31      SYSTEM_Initialize();
32
        printf("dsPIC33: Hi\r\n");
34
35      while(1)
36      {
```


3

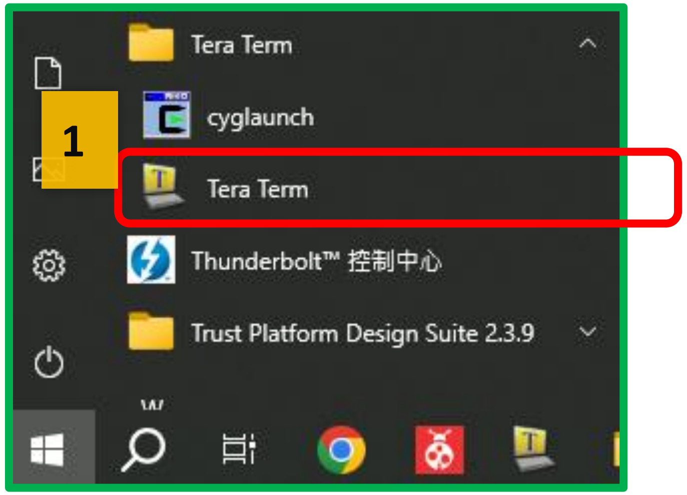
Make and Program Device Main Project
Program Device for Debugging Main Project
Program Device for Production Main Project
Erase Device Memory Main Project
Programmer To Go Main Project

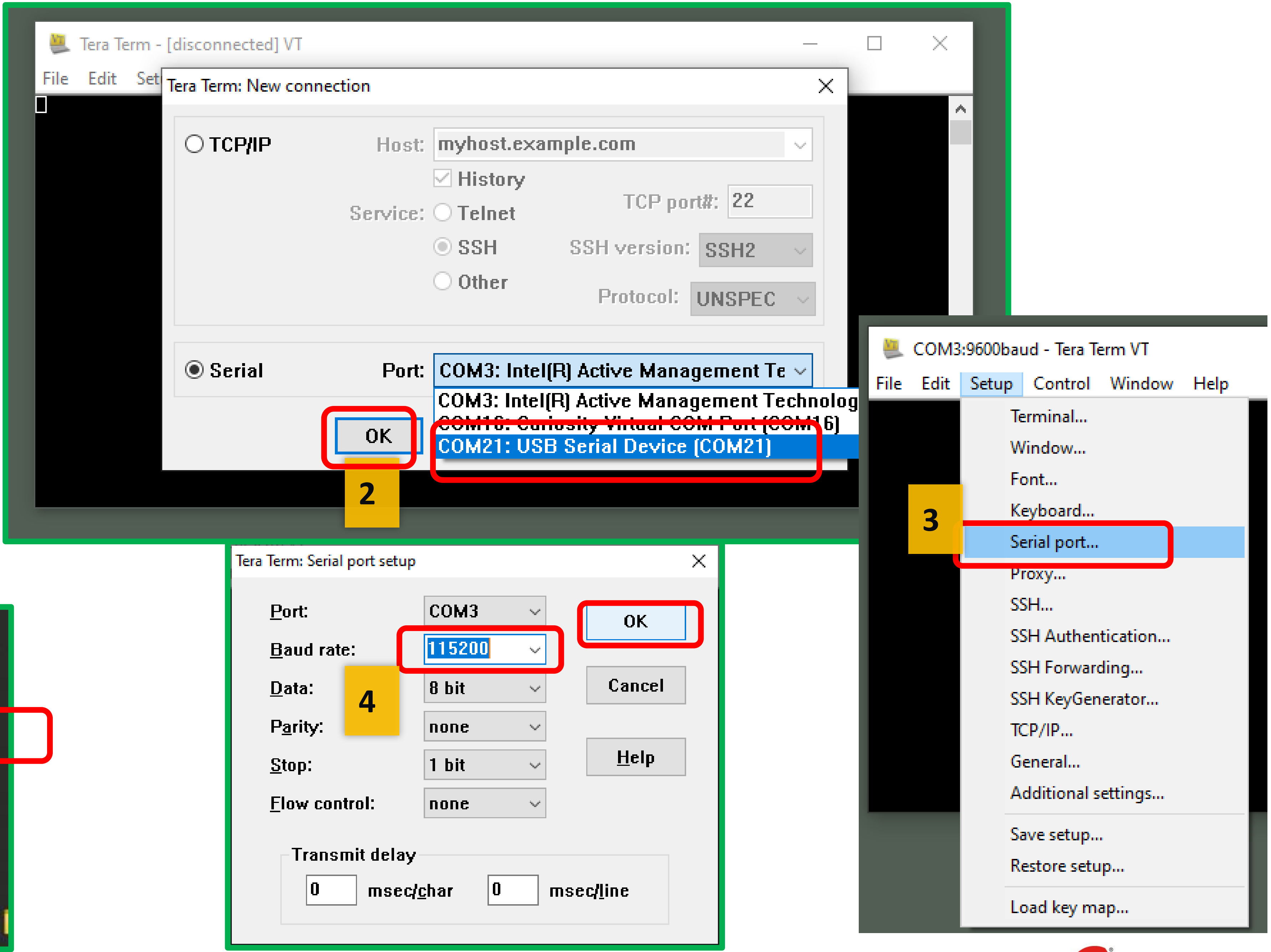
MICROCHIP

6.結果

顯示dsPIC33:Hi

1) 開啟Tera Term
2) 找到COM port
   COM21
3) Setup -> Serial port
4) Baud rate → 115200





Tera Term: New connection

○ TCP/IP Host: myhost.example.com
☑ History
Service: ○ Telnet TCP port#: 22
● SSH SSH version: SSH2
○ Other Protocol: UNSPEC

● Serial Port: COM3: Intel(R) Active Management Te
COM3: Intel(R) Active Management Technolog
COM16: Curiosity Virtual COM Port (COM16)
COM21: USB Serial Device (COM21)

OK

COM3:9600baud - Tera Term VT

File Edit Setup Control Window Help

Terminal...
Window...
Font...
Keyboard...
Serial port...
Proxy...
SSH...
SSH Authentication...
SSH Forwarding...
SSH KeyGenerator...
TCP/IP...
General...
Additional settings...
Save setup...
Restore setup...
Load key map...

Tera Term: Serial port setup

Port: COM3
Baud rate: 115200
Data: 8 bit
Parity: none
Stop: 1 bit
Flow control: none

OK
Cancel
Help

Transmit delay
0 msec/char 0 msec/line

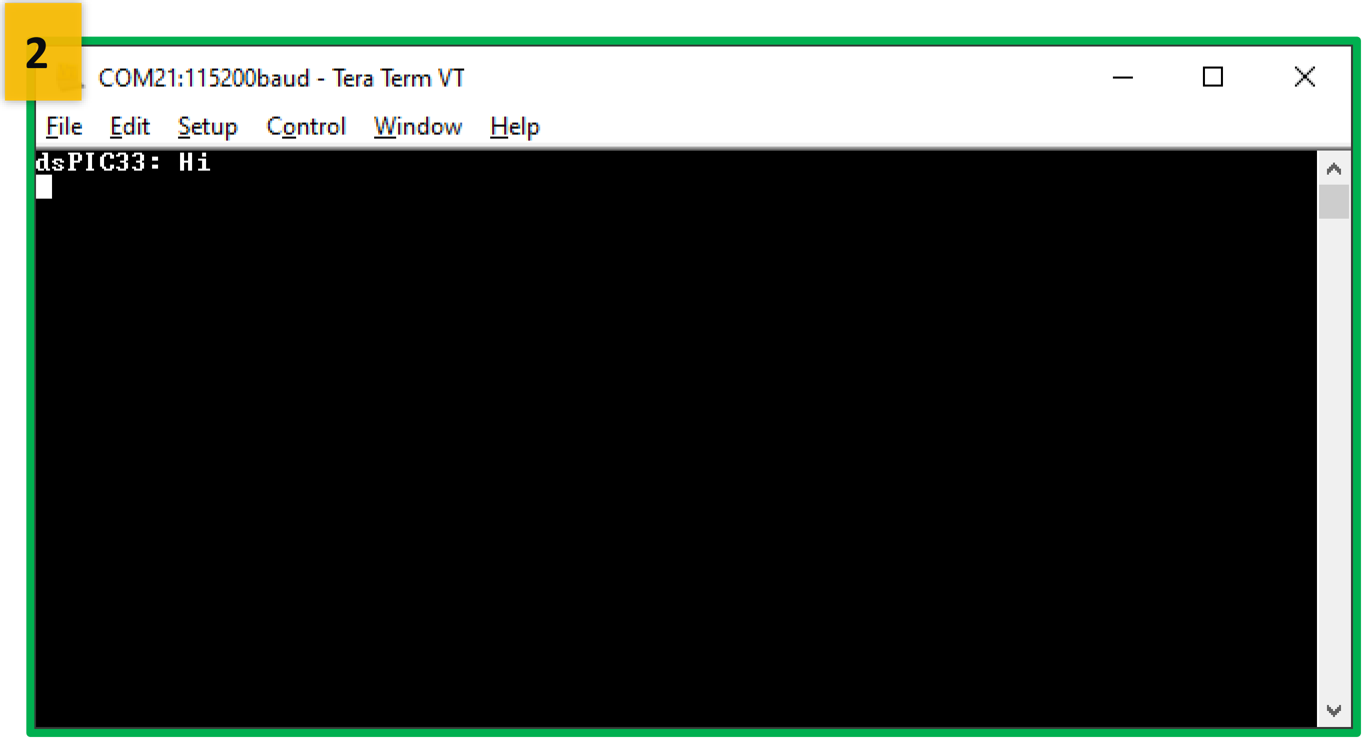
6-1.結果

PC上顯示 dsPIC33: Hi

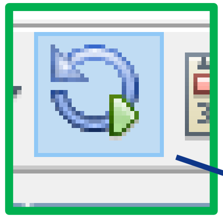
1) 按MPLAB 上的 Reset icon
2) Tera Term 上顯示Hi

進階測試：
根據步驟3架構圖修改腳位由紅板上內建的UART輸出到另一個COM PORT

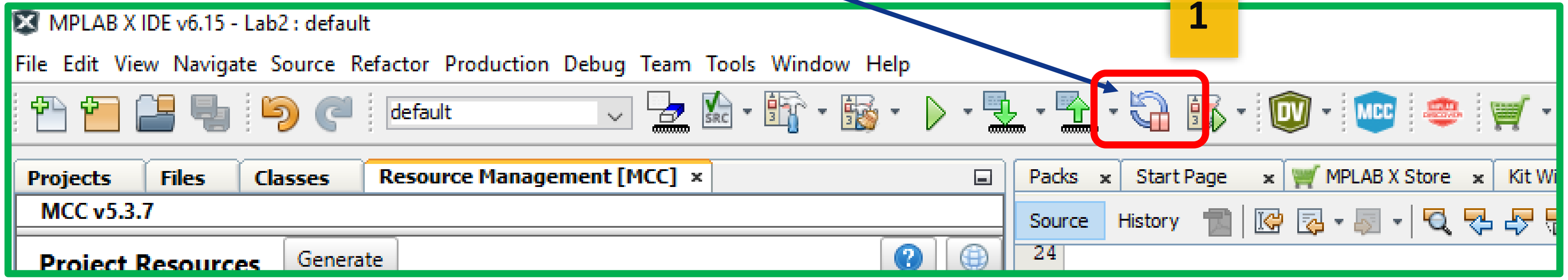
2



```
dsPIC33: Hi
```



1



MICROCHIP

內建硬體程式開發

Timer/PWM/Config

實驗三

Timer

內建硬體程式開發

實驗三：Timer

- 本實驗目的：
  - 架設計時器
    - 最常用的硬體”時間或計數”
    - 用輪詢(Polling)
    - 用中斷(Interrupt)

- 結果
  - 一個LED 每秒只亮100ms (Polling 或 Interrupt)

背景知識

dsPIC33CK datasheet

確認IC内建Timer數量

Datasheet：只有一個16-bit Timer???

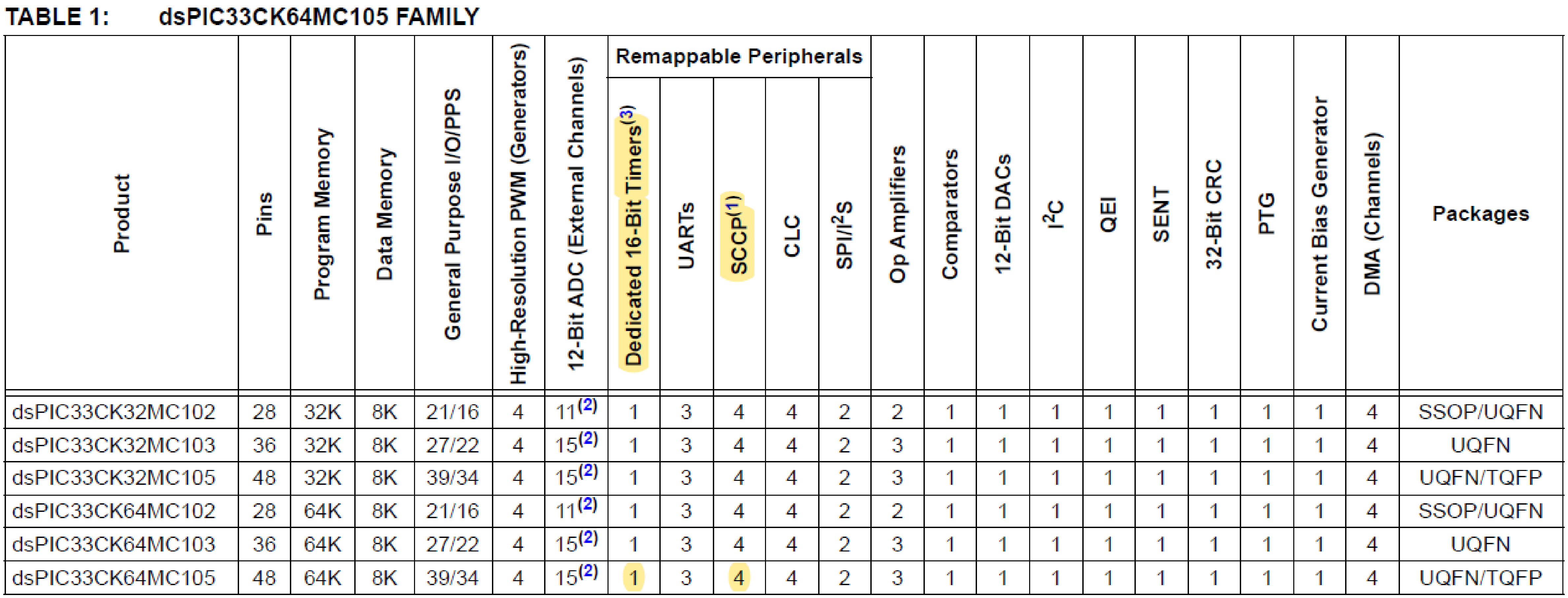


TABLE 1: dsPIC33CK64MC105 FAMILY

Product	Pins	Program Memory	Data Memory	General Purpose I/O/PPS	High-Resolution PWM (Generators)	12-Bit ADC (External Channels)	Remappable Peripherals					Op Amplifiers	Comparators	12-Bit DACs	I^2C	QEI	SENT	32-Bit CRC	PTG	Current Bias Generator	DMA (Channels)	Packages
							Dedicated 16-Bit Timers[3]	UARTs	SCCP[1]	CLC	SPI/I^2S											
dsPIC33CK32MC102	28	32K	8K	21/16	4	11[2]	1	3	4	4	2	2	1	1	1	1	1	1	1	1	4	SSOP/UQFN
dsPIC33CK32MC103	36	32K	8K	27/22	4	15[2]	1	3	4	4	2	3	1	1	1	1	1	1	1	1	4	UQFN
dsPIC33CK32MC105	48	32K	8K	39/34	4	15[2]	1	3	4	4	2	3	1	1	1	1	1	1	1	1	4	UQFN/TQFP
dsPIC33CK64MC102	28	64K	8K	21/16	4	11[2]	1	3	4	4	2	2	1	1	1	1	1	1	1	1	4	SSOP/UQFN
dsPIC33CK64MC103	36	64K	8K	27/22	4	15[2]	1	3	4	4	2	3	1	1	1	1	1	1	1	1	4	UQFN
dsPIC33CK64MC105	48	64K	8K	39/34	4	15[2]	1	3	4	4	2	3	1	1	1	1	1	1	1	1	4	UQFN/TQFP

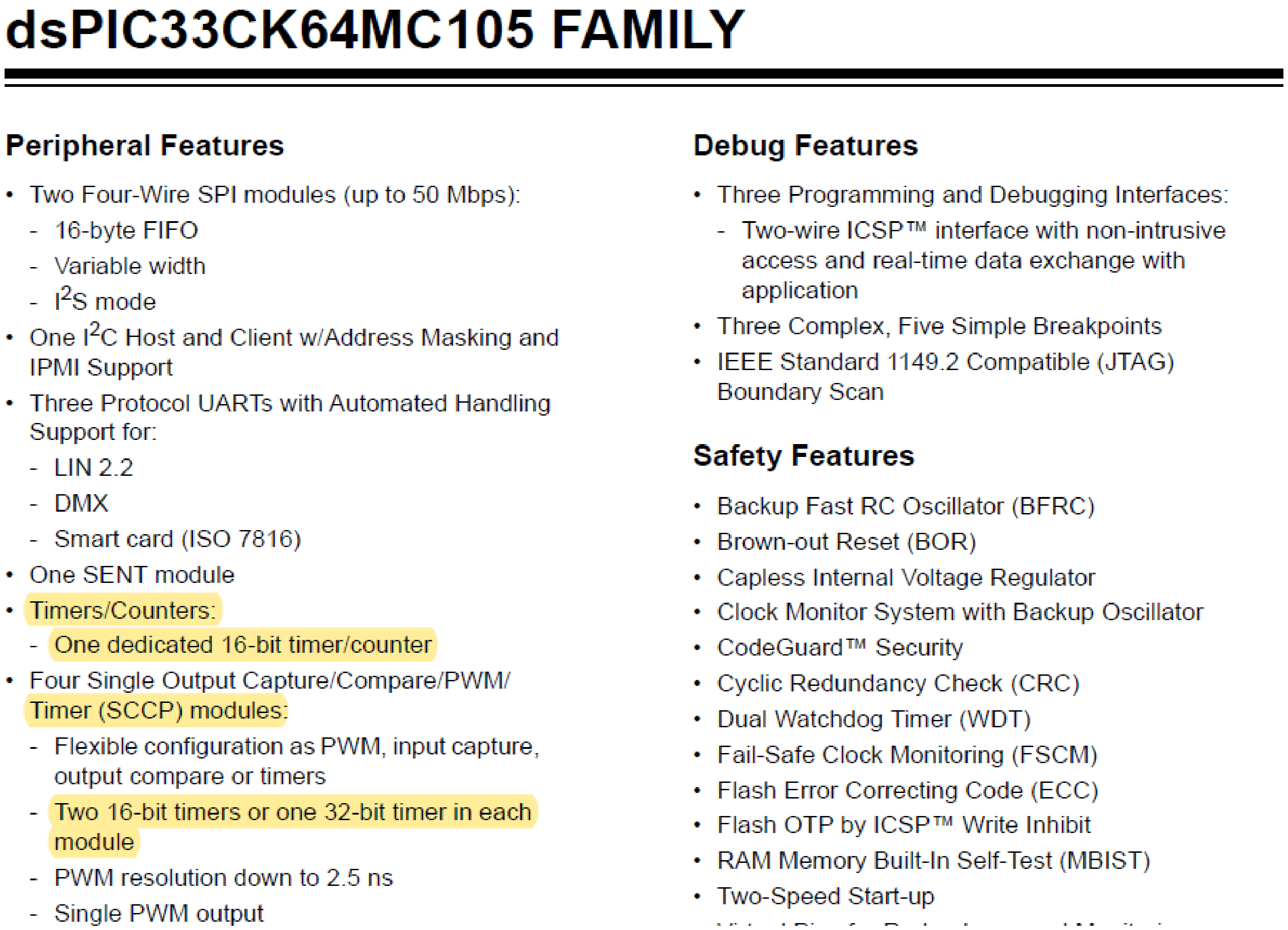
計時器：Timer

16-bit x 1 + (16-bit x 8 或 32-bit x4)

最多可以有
9個16-bit計時器：

1個16-bit
的專用計時器

8個16-bit
由4個SCCP設成
的計時器



dsPIC33CK64MC105 FAMILY

Peripheral Features

- Two Four-Wire SPI modules (up to 50 Mbps):
  - 16-byte FIFO
  - Variable width
  - I^2S mode
- One I^2C Host and Client w/Address Masking and IPMI Support
- Three Protocol UARTs with Automated Handling Support for:
  - LIN 2.2
  - DMX
  - Smart card (ISO 7816)
- One SENT module
- Timers/Counters:
  - One dedicated 16-bit timer/counter
- Four Single Output Capture/Compare/PWM/Timer (SCCP) modules:
  - Flexible configuration as PWM, input capture, output compare or timers
  - Two 16-bit timers or one 32-bit timer in each module
  - PWM resolution down to 2.5 ns
  - Single PWM output

Debug Features

- Three Programming and Debugging Interfaces:
  - Two-wire ICSP™ interface with non-intrusive access and real-time data exchange with application
- Three Complex, Five Simple Breakpoints
- IEEE Standard 1149.2 Compatible (JTAG) Boundary Scan

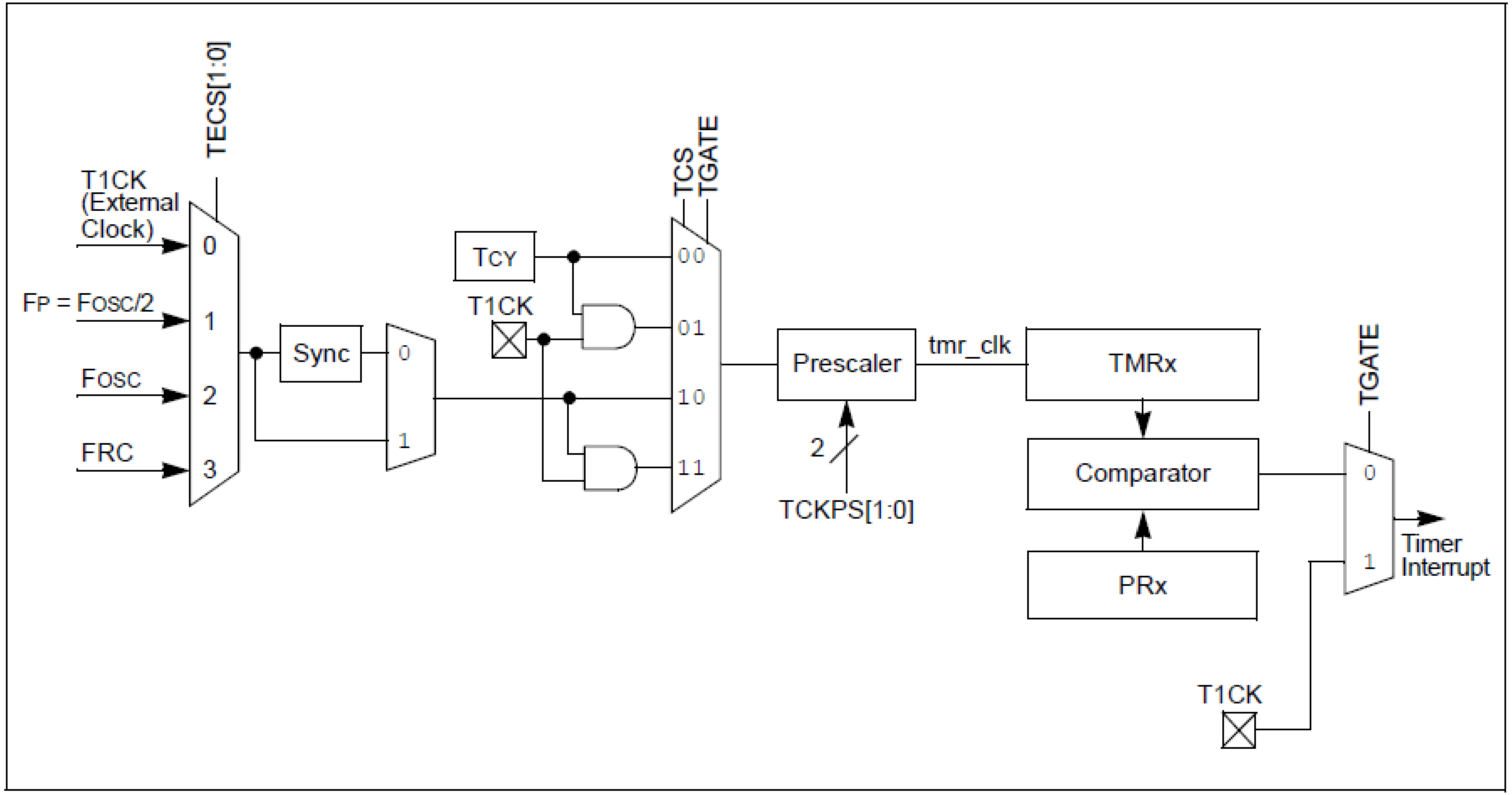
Safety Features

- Backup Fast RC Oscillator (BFRC)
- Brown-out Reset (BOR)
- Capless Internal Voltage Regulator
- Clock Monitor System with Backup Oscillator
- CodeGuard™ Security
- Cyclic Redundancy Check (CRC)
- Dual Watchdog Timer (WDT)
- Fail-Safe Clock Monitoring (FSCM)
- Flash Error Correcting Code (ECC)
- Flash OTP by ICSP™ Write Inhibit
- RAM Memory Built-In Self-Test (MBIST)
- Two-Speed Start-up

Timer架構

專用16-bit 計時/計數器

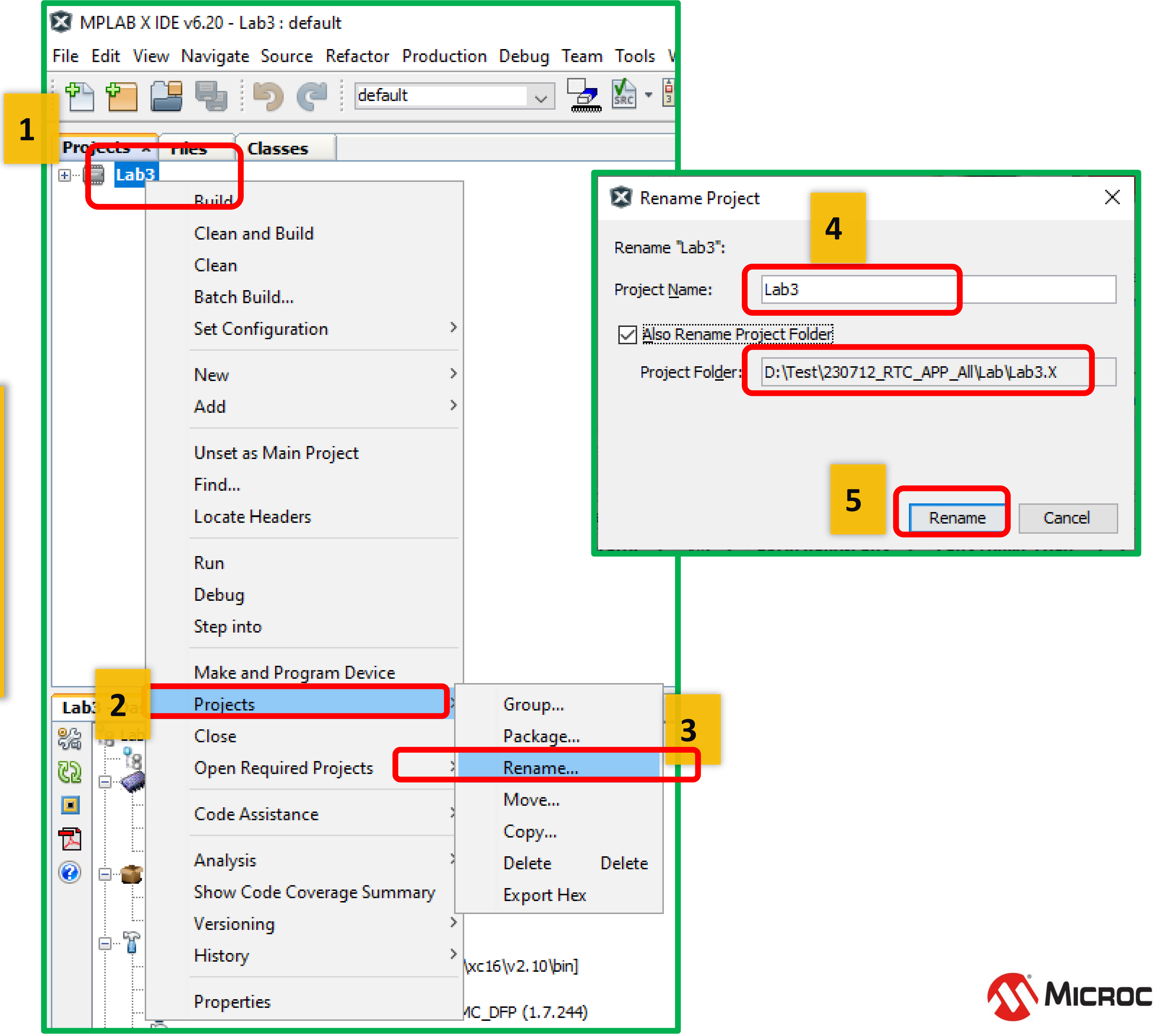
FIGURE 19-1: 16-BIT TIMER1 MODULE BLOCK DIAGRAM



1.建立Lab3

將Lab 2改名成Lab3

1) 到Projects 在Lab2上 滑鼠點右鍵
2) 找到Projects -> Rename
3) 修改成Lab3
4) 勾選順便改資料夾名
5) 按 Rename 完成名稱修改



Rename "Lab3":

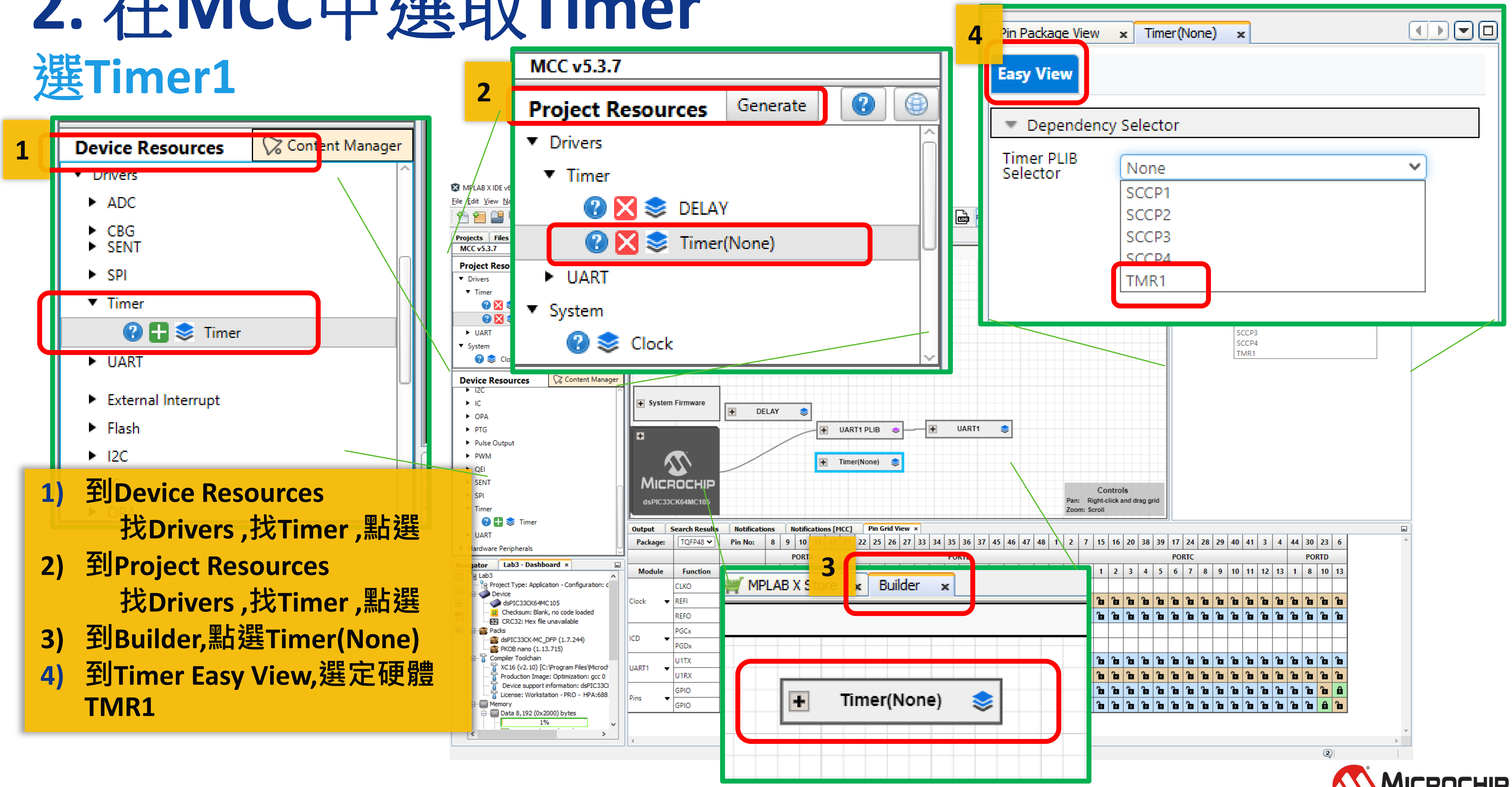
Project Name: Lab3

☑ Also Rename Project Folder

Project Folder: D:\Test\230712_RTC_APP_All\Lab\Lab3.X

2. 在MCC中選取Timer

選Timer1



1) 到Device Resources
   找Drivers ,找Timer ,點選

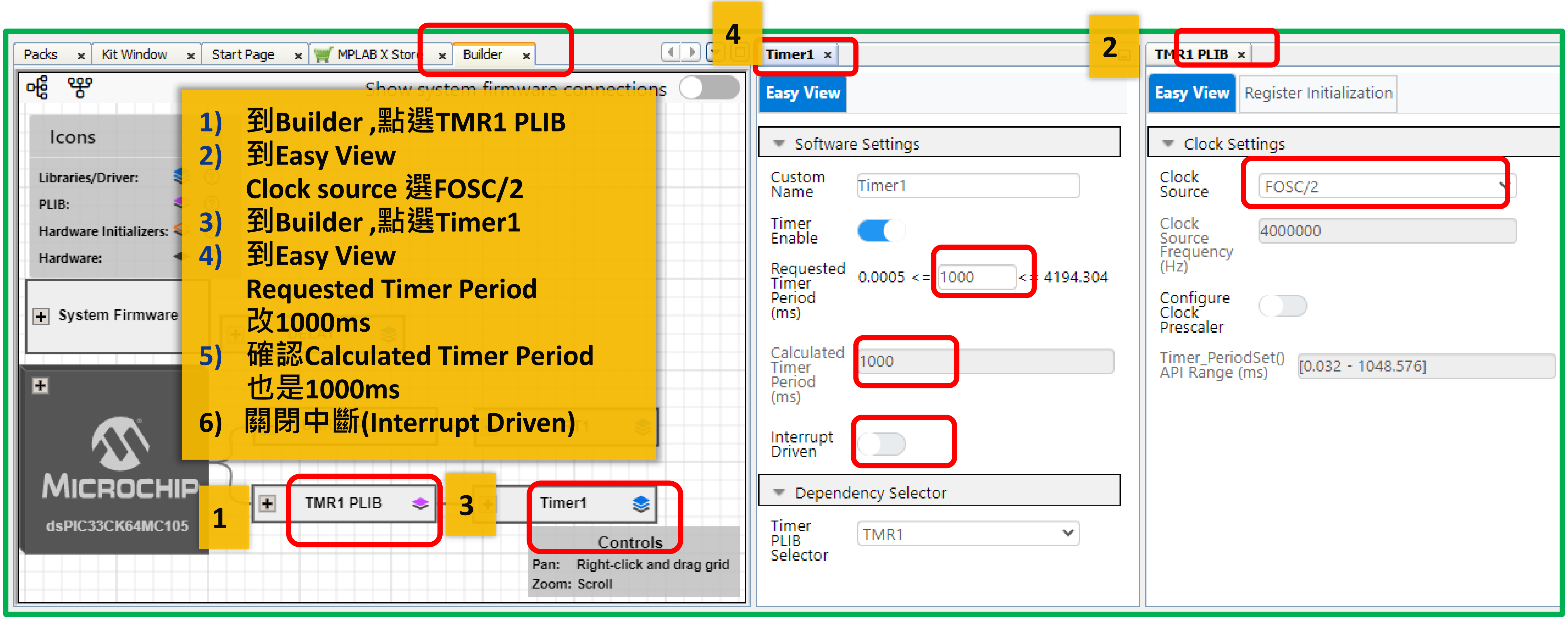
2) 到Project Resources
   找Drivers ,找Timer ,點選

3) 到Builder,點選Timer(None)

4) 到Timer Easy View,選定硬體 TMR1

3. 設定Timer

設定1 sec => 1000ms



1) 到Builder ,點選TMR1 PLIB
2) 到Easy View
   Clock source 選FOSC/2
3) 到Builder ,點選Timer1
4) 到Easy View
   Requested Timer Period 改1000ms
5) 確認Calculated Timer Period 也是1000ms
6) 關閉中斷(Interrupt Driven)

Timer1 – Easy View – Software Settings:

- Custom Name: Timer1
- Timer Enable: on
- Requested Timer Period (ms): 0.0005 <= 1000 <= 4194.304
- Calculated Timer Period (ms): 1000
- Interrupt Driven: off

Dependency Selector:

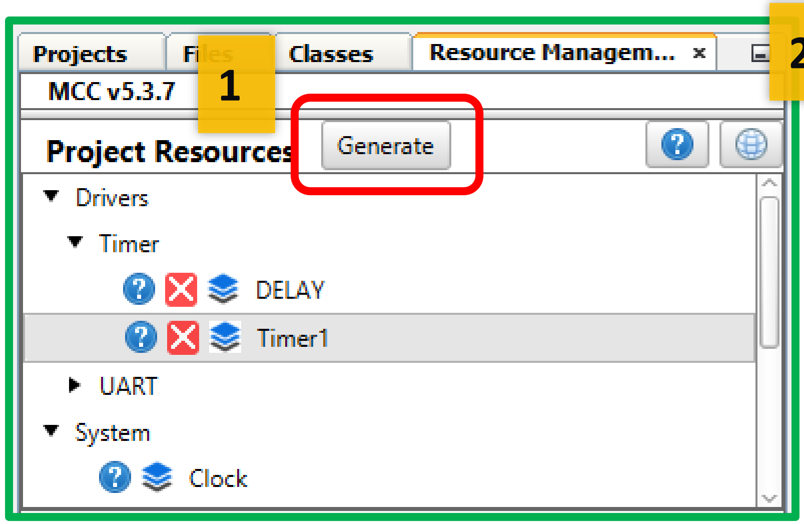
- Timer PLIB Selector: TMR1

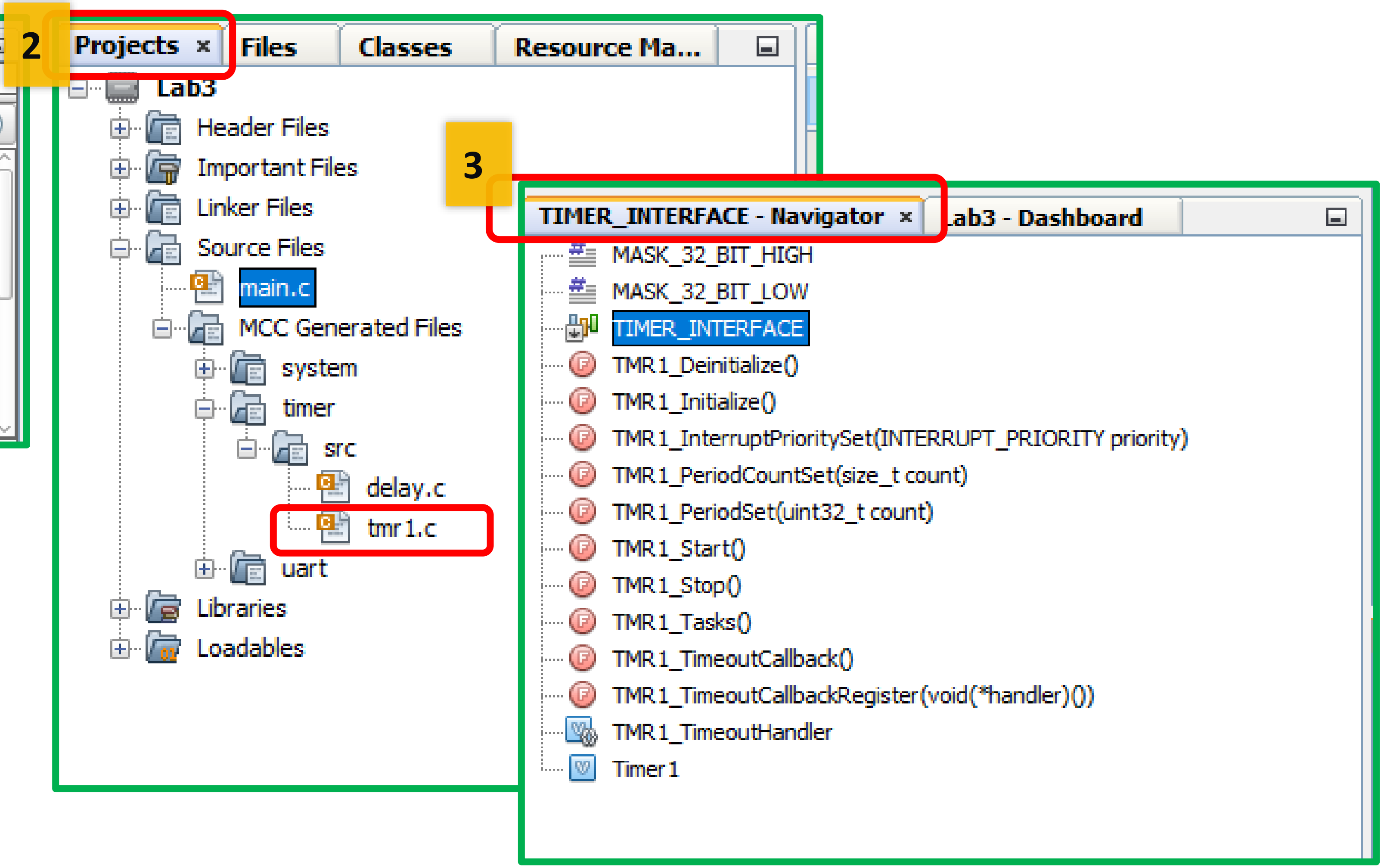
TMR1 PLIB – Easy View – Clock Settings:

- Clock Source: FOSC/2
- Clock Source Frequency (Hz): 4000000
- Configure Clock Prescaler: off
- Timer_PeriodSet() API Range (ms): [0.032 - 1048.576]

Controls
Pan: Right-click and drag grid
Zoom: Scroll

4. 產生CODE





```
MASK_32_BIT_HIGH
MASK_32_BIT_LOW
TIMER_INTERFACE
TMR1_Deinitialize()
TMR1_Initialize()
TMR1_InterruptPrioritySet(INTERRUPT_PRIORITY priority)
TMR1_PeriodCountSet(size_t count)
TMR1_PeriodSet(uint32_t count)
TMR1_Start()
TMR1_Stop()
TMR1_Tasks()
TMR1_TimeoutCallback()
TMR1_TimeoutCallbackRegister(void(*handler)())
TMR1_TimeoutHandler
Timer1
```

1) 按 Generate

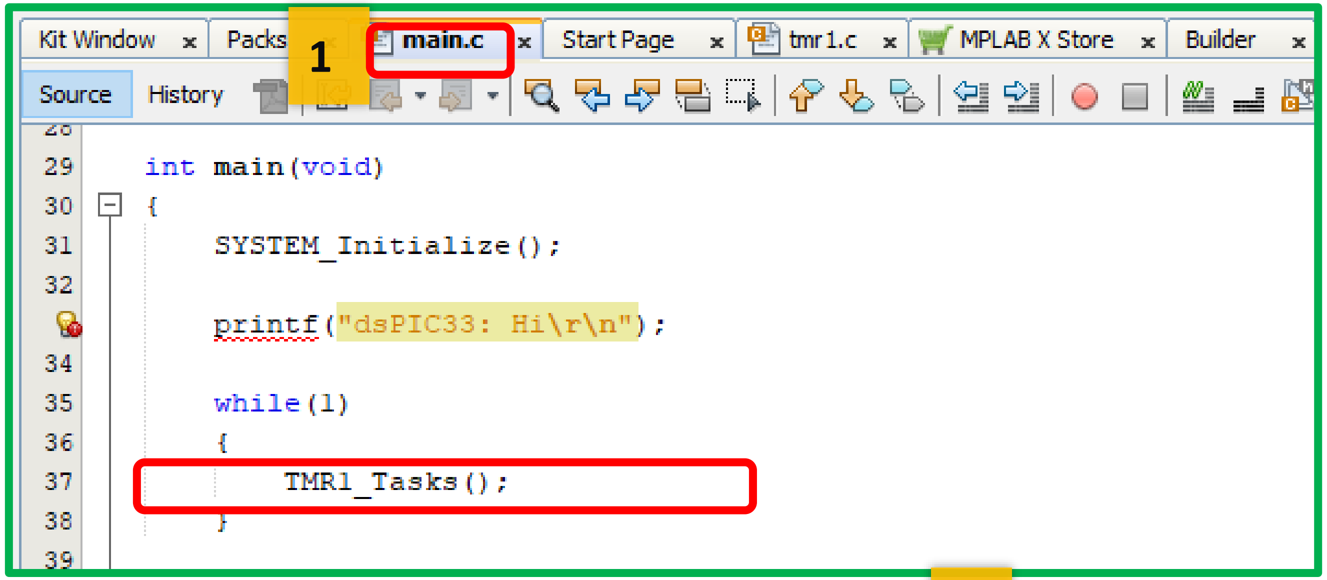
2) 到Projects
多出了一個tmr1.c

3) 開啟tmr1.c
到Navigator
可以看到有哪些API可用

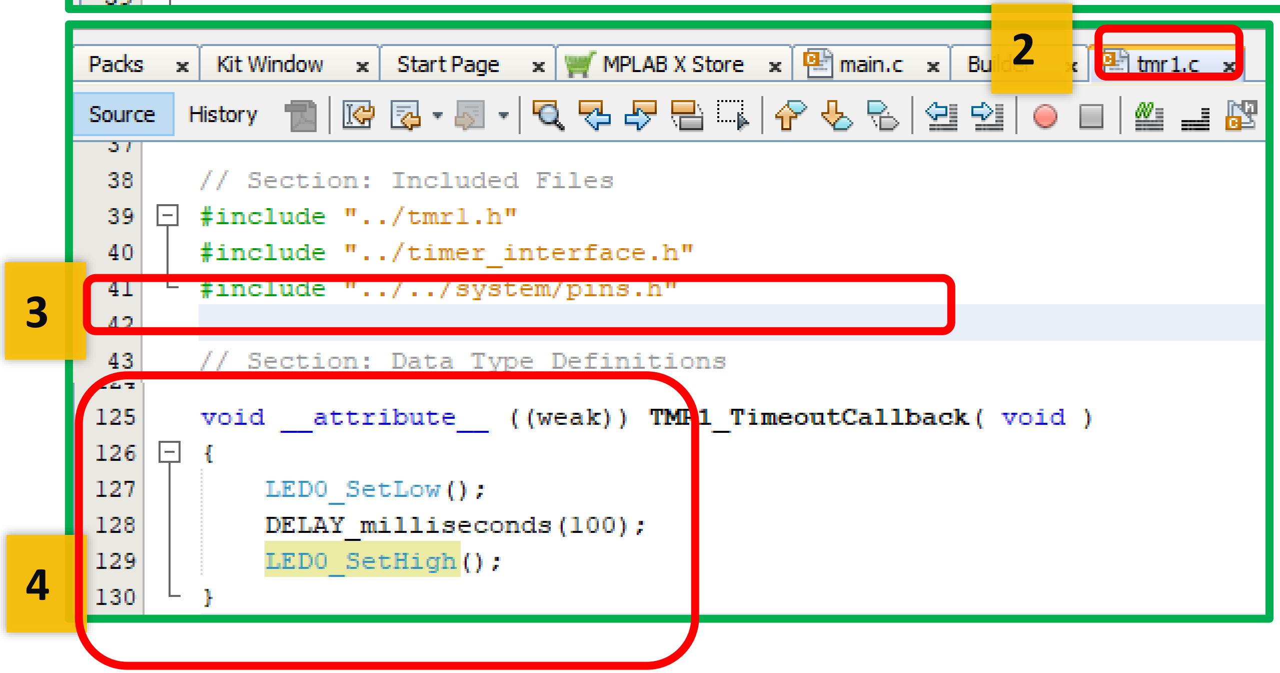
5. 寫CODE

LED0 - 1 Hz

1) 到main.c
   的While(1) {}內
   寫TMR1_Task();
2) 到tmr1.c
3) 開頭處
   #include "../../system/pins.h“
4) TMR1_TimeoutCallback(void)
   {}內寫
   LED0_SetLow();
   DELAY_milliseconds(100);
   LED0_SetHigh();



```
int main(void)
{
    SYSTEM_Initialize();

    printf("dsPIC33: Hi\r\n");

    while(1)
    {
        TMR1_Tasks();
    }
```



```
// Section: Included Files
#include "../tmr1.h"
#include "../timer_interface.h"
#include "../../system/pins.h"

// Section: Data Type Definitions

void __attribute__ ((weak)) TMR1_TimeoutCallback( void )
{
    LED0_SetLow();
    DELAY_milliseconds(100);
    LED0_SetHigh();
}
```

6.結果

Lab3_0

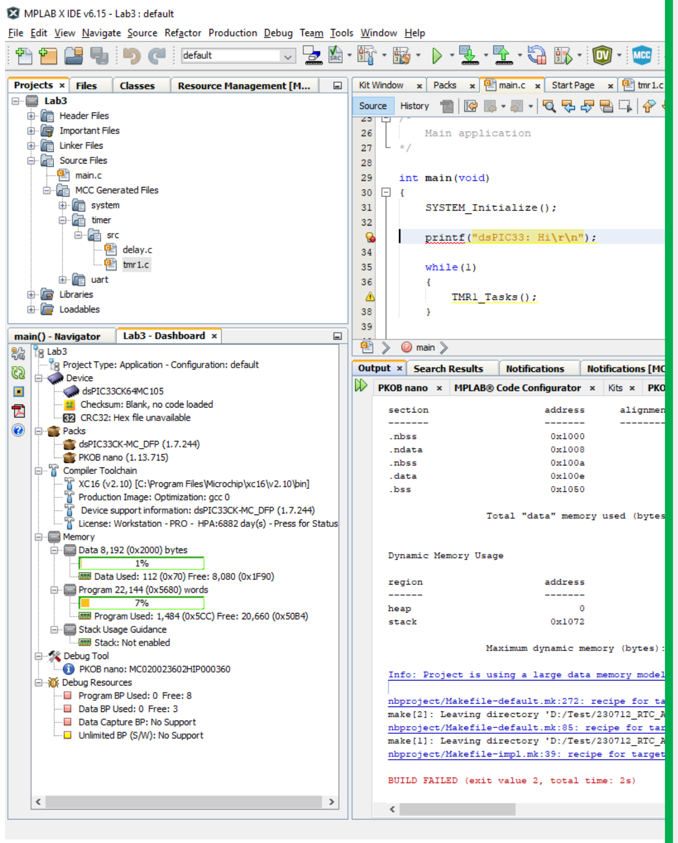
1)燒錄測試

2)結果

紅板上LED0 , 每秒亮100ms

7. 補充：出現此錯誤

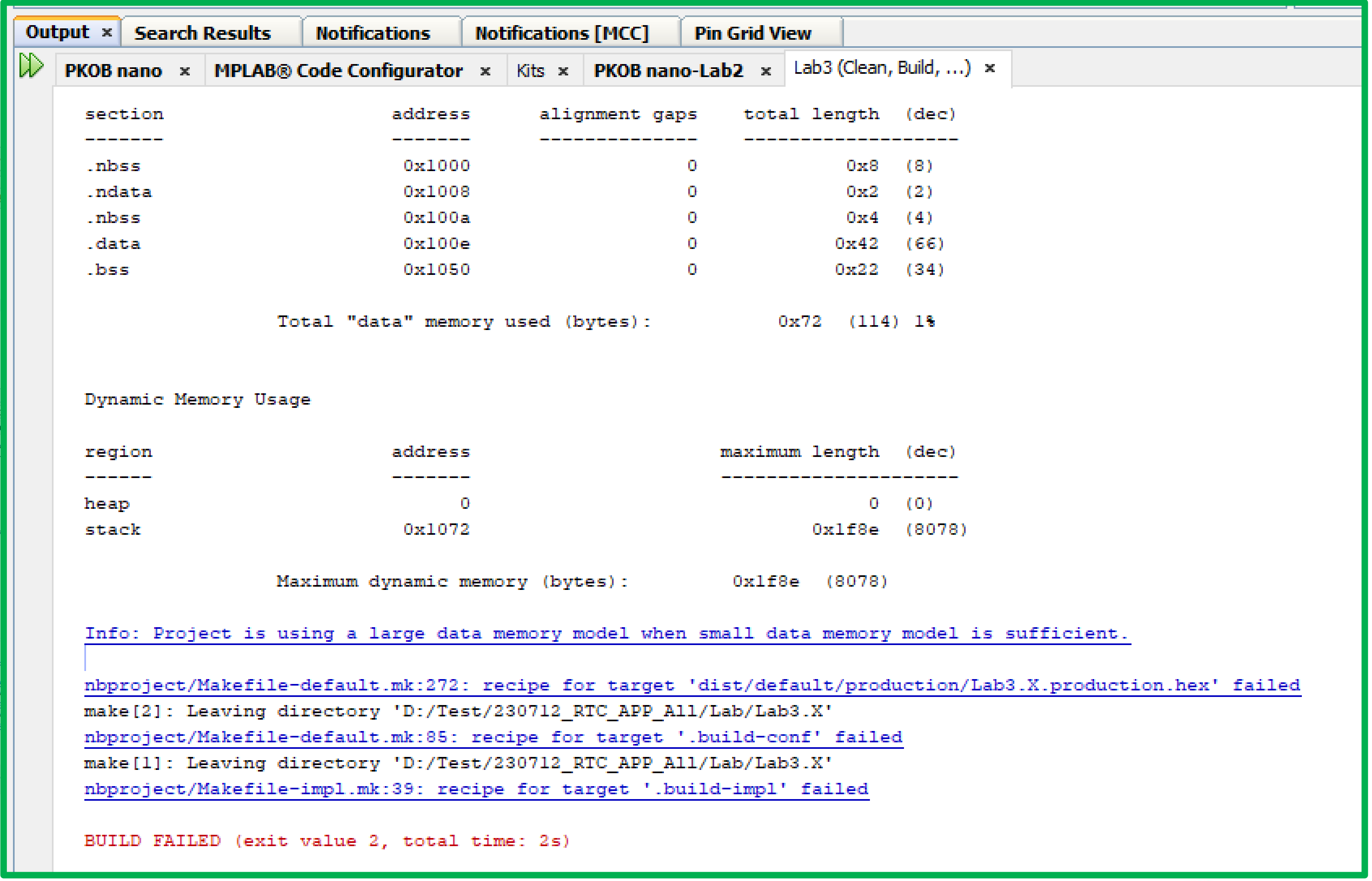
代表沒寫到這行 #include "./../../system/pins.h“



MPLAB X IDE v6.15 - Lab3 : default

```
/*
    Main application
 */

int main(void)
{
    SYSTEM_Initialize();

    printf("dsPIC33: Hi\r\n");

    while(1)
    {
        TMR1_Tasks();
    }
```

Lab3
- Project Type: Application - Configuration: default
- Device
  - dsPIC33CK64MC105
  - Checksum: Blank, no code loaded
  - CRC32: Hex file unavailable
- Packs
  - dsPIC33CK-MC_DFP (1.7.244)
  - PKOB nano (1.13.715)
- Compiler Toolchain
  - XC16 (v2.10) [C:\Program Files\Microchip\xc16\v2.10\bin]
  - Production Image: Optimization: gcc 0
  - Device support information: dsPIC33CK-MC_DFP (1.7.244)
  - License: Workstation - PRO - HPA:6882 day(s) - Press for Status
- Memory
  - Data 8,192 (0x2000) bytes
    - 1%
    - Data Used: 112 (0x70) Free: 8,080 (0x1F90)
  - Program 22,144 (0x5680) words
    - 7%
    - Program Used: 1,484 (0x5CC) Free: 20,660 (0x50B4)
  - Stack Usage Guidance
    - Stack: Not enabled
- Debug Tool
  - PKOB nano: MC020023602HIP000360
- Debug Resources
  - Program BP Used: 0 Free: 8
  - Data BP Used: 0 Free: 3
  - Data Capture BP: No Support
  - Unlimited BP (S/W): No Support

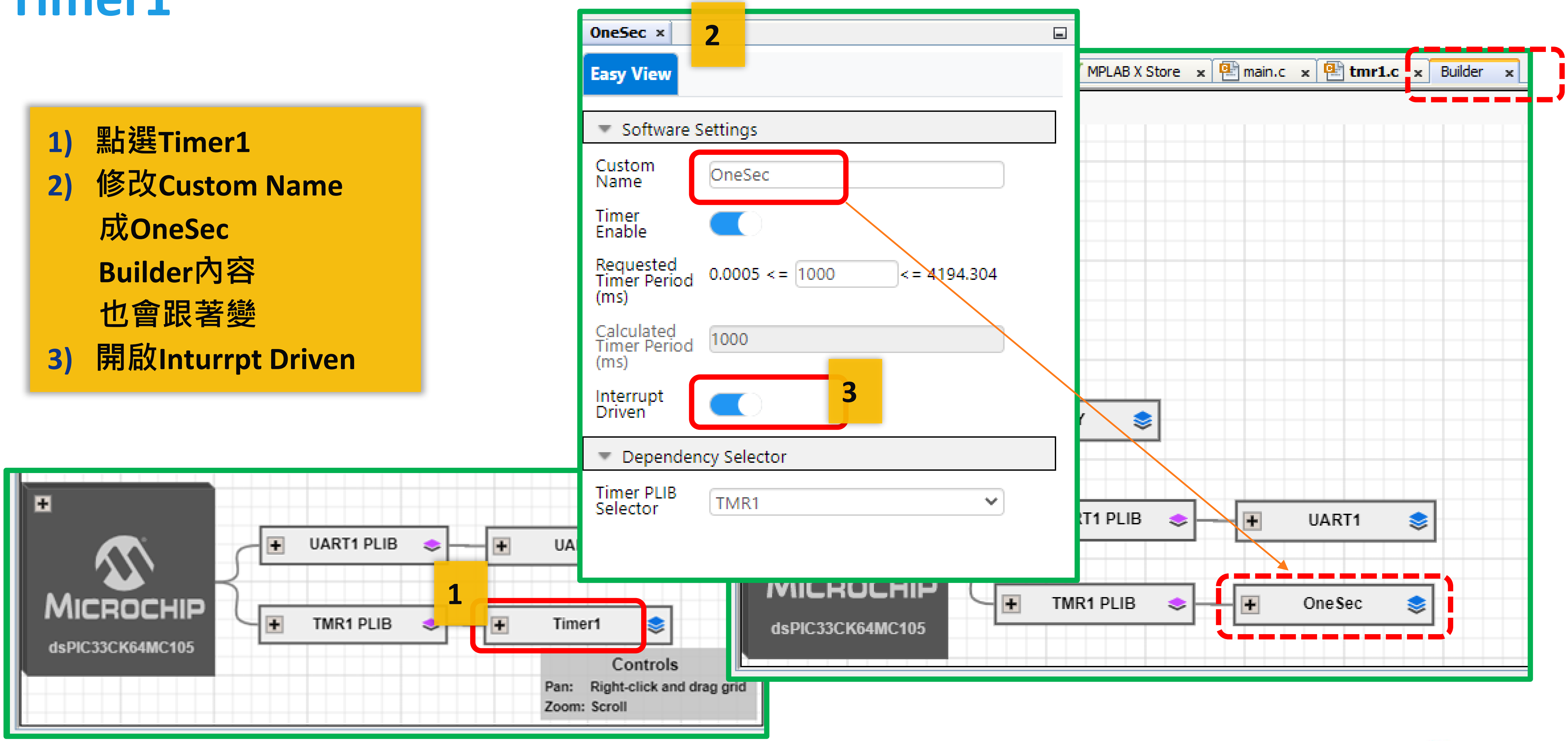


```
section              address     alignment gaps    total length  (dec)
-------              -------     --------------    -------------------
.nbss                 0x1000                  0           0x8  (8)
.ndata                0x1008                  0           0x2  (2)
.nbss                 0x100a                  0           0x4  (4)
.data                 0x100e                  0          0x42  (66)
.bss                  0x1050                  0          0x22  (34)

                 Total "data" memory used (bytes):          0x72  (114) 1%


Dynamic Memory Usage

region               address                      maximum length  (dec)
------               -------                      ---------------------
heap                       0                                   0  (0)
stack                 0x1072                              0x1f8e  (8078)

                 Maximum dynamic memory (bytes):         0x1f8e  (8078)

Info: Project is using a large data memory model when small data memory model is sufficient.

nbproject/Makefile-default.mk:272: recipe for target 'dist/default/production/Lab3.X.production.hex' failed
make[2]: Leaving directory 'D:/Test/230712_RTC_APP_All/Lab/Lab3.X'
nbproject/Makefile-default.mk:85: recipe for target '.build-conf' failed
make[1]: Leaving directory 'D:/Test/230712_RTC_APP_All/Lab/Lab3.X'
nbproject/Makefile-impl.mk:39: recipe for target '.build-impl' failed

BUILD FAILED (exit value 2, total time: 2s)
```

8.改用中斷方式

Timer1

1) 點選Timer1
2) 修改Custom Name 成OneSec Builder內容 也會跟著變
3) 開啟Inturrpt Driven

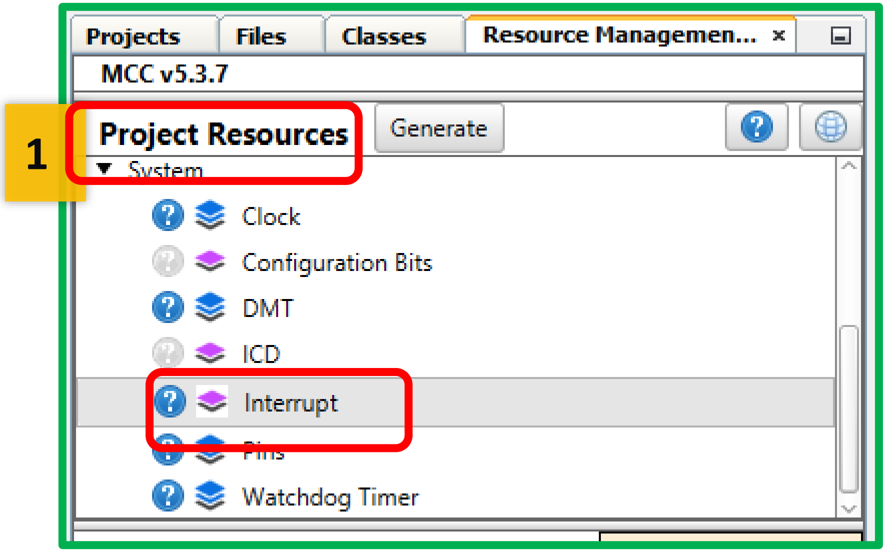


9.設定中斷

設定中斷優先權

1) 到Project Resource, 點Interrupt
2) 到Easy View , 找到OneSec
   修改優先權等級 IPL 為 2

軟體可調優先權最高7
硬體內定優先權最高15

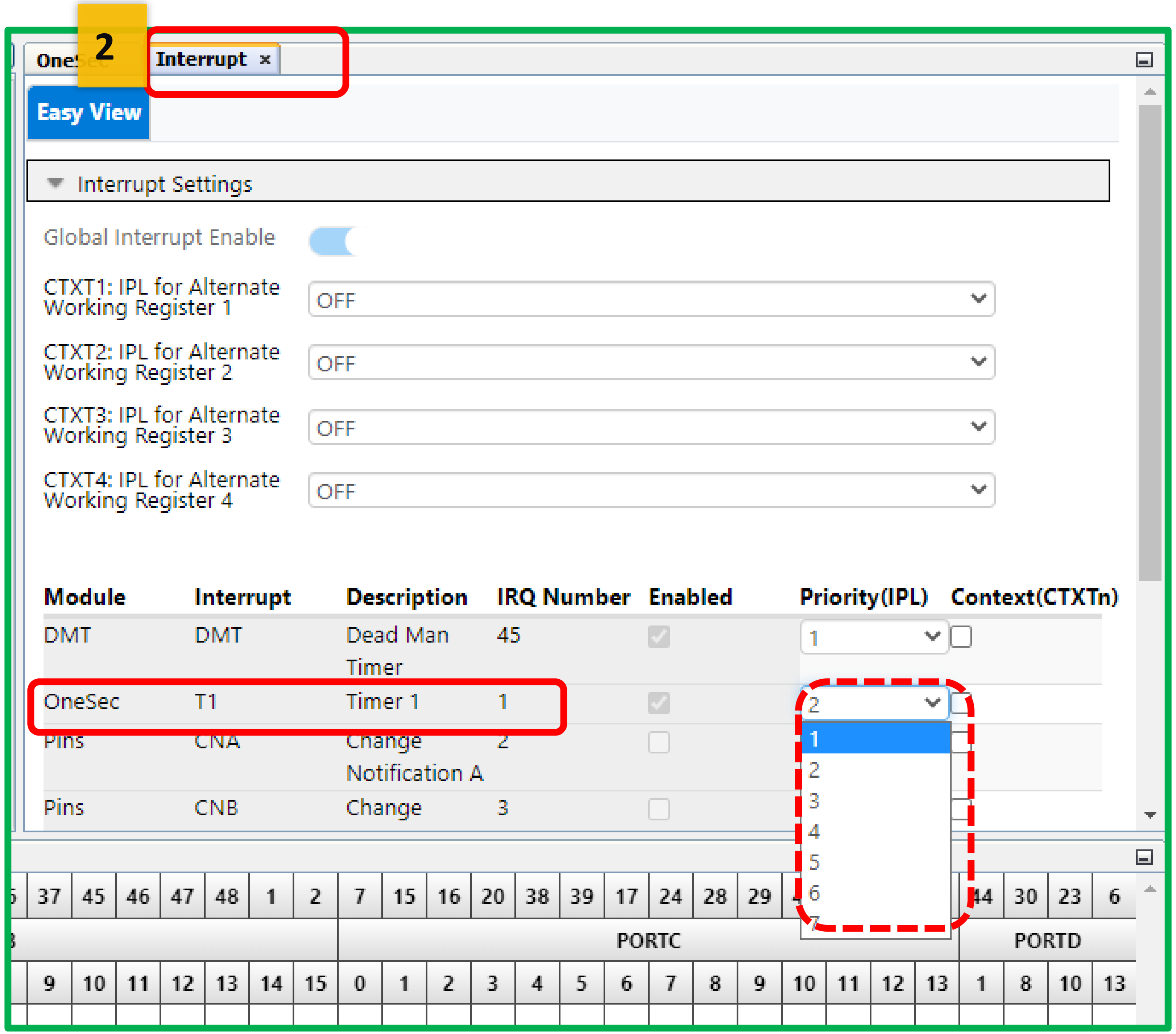

1

Projects | Files | Classes | Resource Managemen... ×

MCC v5.3.7

Project Resources | Generate

- System
  - Clock
  - Configuration Bits
  - DMT
  - ICD
  - Interrupt
  - Pins
  - Watchdog Timer


2

OneSec | Interrupt ×

Easy View

Interrupt Settings

Global Interrupt Enable

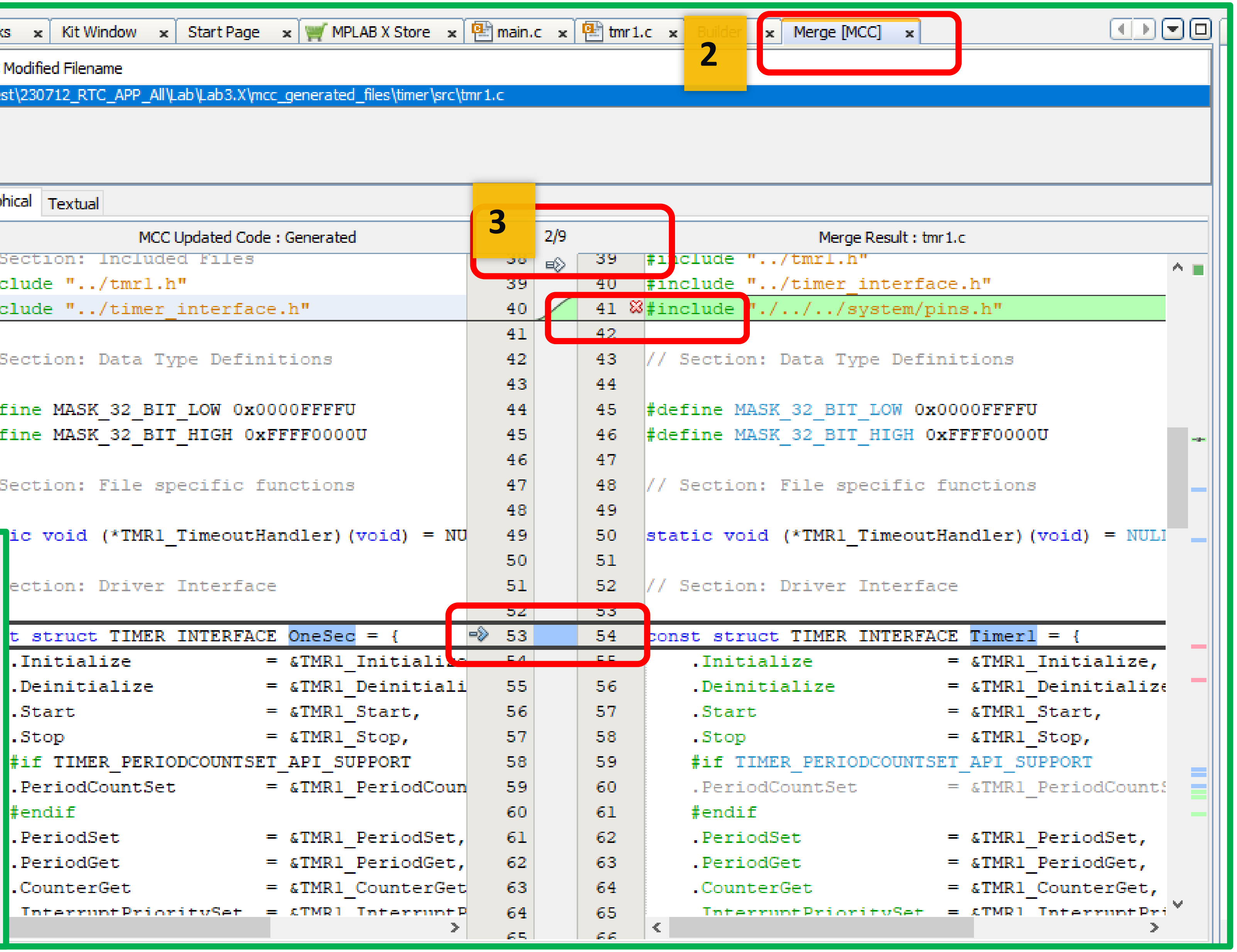
CTXT1: IPL for Alternate Working Register 1	OFF
CTXT2: IPL for Alternate Working Register 2	OFF
CTXT3: IPL for Alternate Working Register 3	OFF
CTXT4: IPL for Alternate Working Register 4	OFF

Module	Interrupt	Description	IRQ Number	Enabled	Priority(IPL)	Context(CTXTn)
DMT	DMT	Dead Man Timer	45	☑	1	□
OneSec	T1	Timer 1	1	☑	2	□
Pins	CNA	Change Notification A	2	□		□
Pins	CNB	Change	3	□		□

```
5 37 45 46 47 48 1 2 7 15 16 20 38 39 17 24 28 29 4 ... 44 30 23 6
                        PORTC                          PORTD
9 10 11 12 13 14 15 0 1 2 3 4 5 6 7 8 9 10 11 12 13 1 8 10 13
```

10.產生CODE

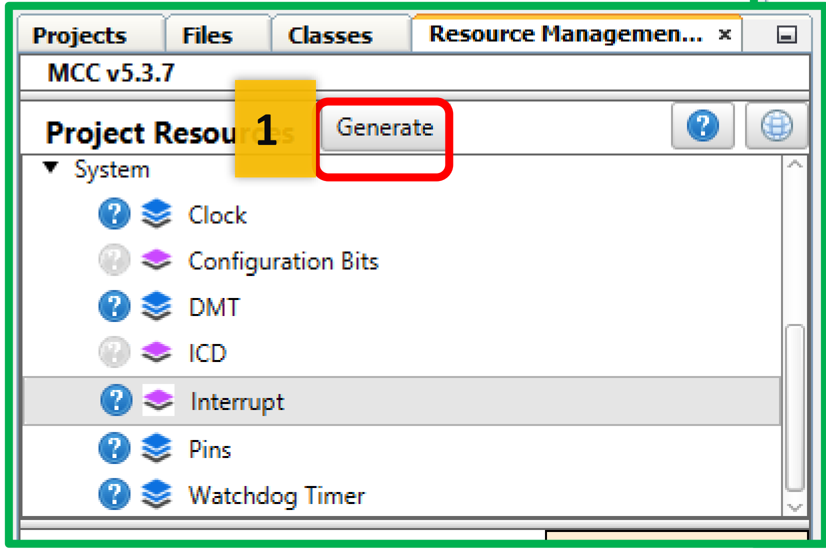
1) 按 Generate
2) 確認差異
   X 或 ➔ 達到刪除或取代
3)按=> 全部取代



D:\Test\230712_RTC_APP_All\Lab\Lab3.X\mcc_generated_files\timer\src\tmr1.c

```
// Section: Included Files
#include "../tmr1.h"
#include "../timer_interface.h"

// Section: Data Type Definitions

#define MASK_32_BIT_LOW 0x0000FFFFU
#define MASK_32_BIT_HIGH 0xFFFF0000U

// Section: File specific functions

ic void (*TMR1_TimeoutHandler)(void) = NU

ection: Driver Interface

t struct TIMER_INTERFACE OneSec = {
.Initialize            = &TMR1_Initiali
.Deinitialize          = &TMR1_Deinitiali
.Start                 = &TMR1_Start,
.Stop                  = &TMR1_Stop,
#if TIMER_PERIODCOUNTSET_API_SUPPORT
.PeriodCountSet        = &TMR1_PeriodCoun
#endif
.PeriodSet             = &TMR1_PeriodSet,
.PeriodGet             = &TMR1_PeriodGet,
.CounterGet            = &TMR1_CounterGet
.InterruptPrioritySet  = &TMR1_InterruptP
```

```
#include "../tmr1.h"
#include "../timer_interface.h"
#include "../../../system/pins.h"

// Section: Data Type Definitions

#define MASK_32_BIT_LOW 0x0000FFFFU
#define MASK_32_BIT_HIGH 0xFFFF0000U

// Section: File specific functions

static void (*TMR1_TimeoutHandler)(void) = NULl

// Section: Driver Interface

const struct TIMER_INTERFACE Timer1 = {
    .Initialize            = &TMR1_Initialize,
    .Deinitialize          = &TMR1_Deinitialize
    .Start                 = &TMR1_Start,
    .Stop                  = &TMR1_Stop,
    #if TIMER_PERIODCOUNTSET_API_SUPPORT
    .PeriodCountSet        = &TMR1_PeriodCount
    #endif
    .PeriodSet             = &TMR1_PeriodSet,
    .PeriodGet             = &TMR1_PeriodGet,
    .CounterGet            = &TMR1_CounterGet,
    .InterruptPrioritySet  = &TMR1_InterruptPri
```

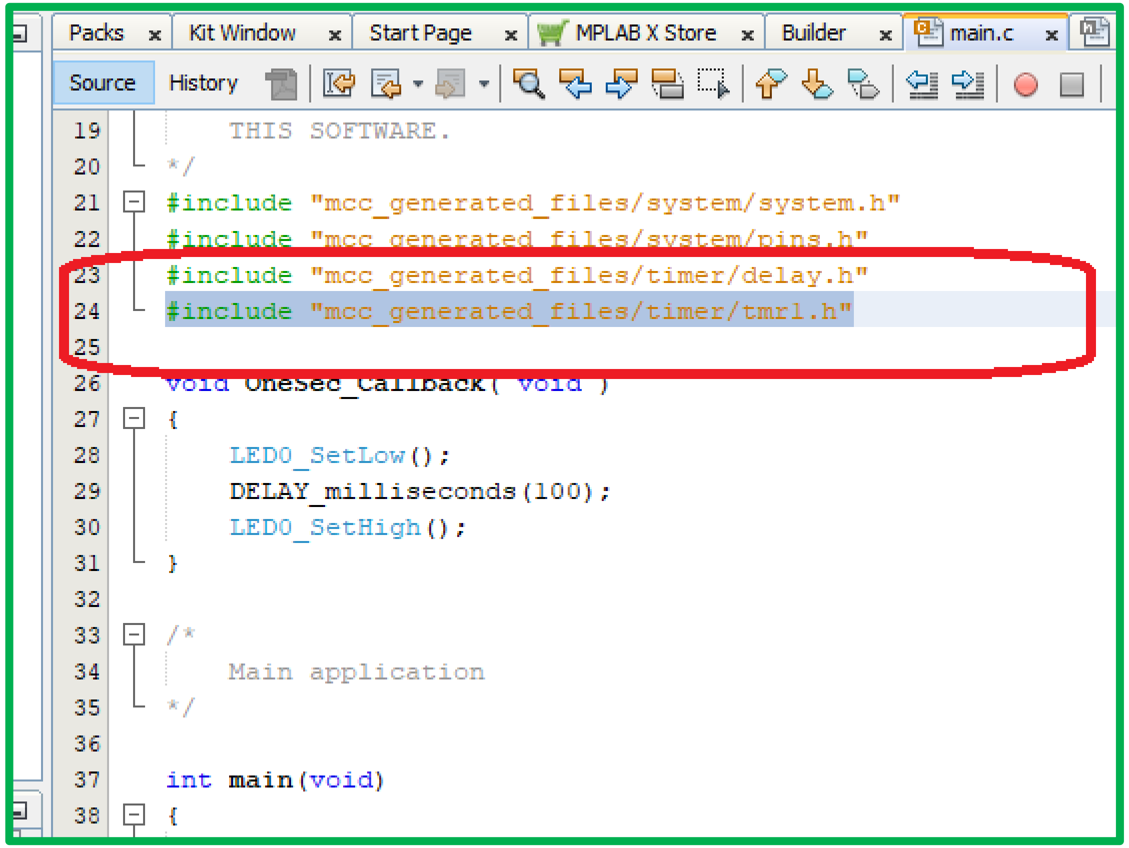


11.寫CODE

tmr1.h

1) 在main.c

2) 開頭加入
#include "mcc_generated_files/timer/tmr1.h“



```
    THIS SOFTWARE.
*/
#include "mcc_generated_files/system/system.h"
#include "mcc_generated_files/system/pins.h"
#include "mcc_generated_files/timer/delay.h"
#include "mcc_generated_files/timer/tmr1.h"

void OneSec_Callback( void )
{
    LED0_SetLow();
    DELAY_milliseconds(100);
    LED0_SetHigh();
}

/*
    Main application
*/

int main(void)
{
```

12.寫CODE

用CallBack寫法

1) 繼續修改main.c

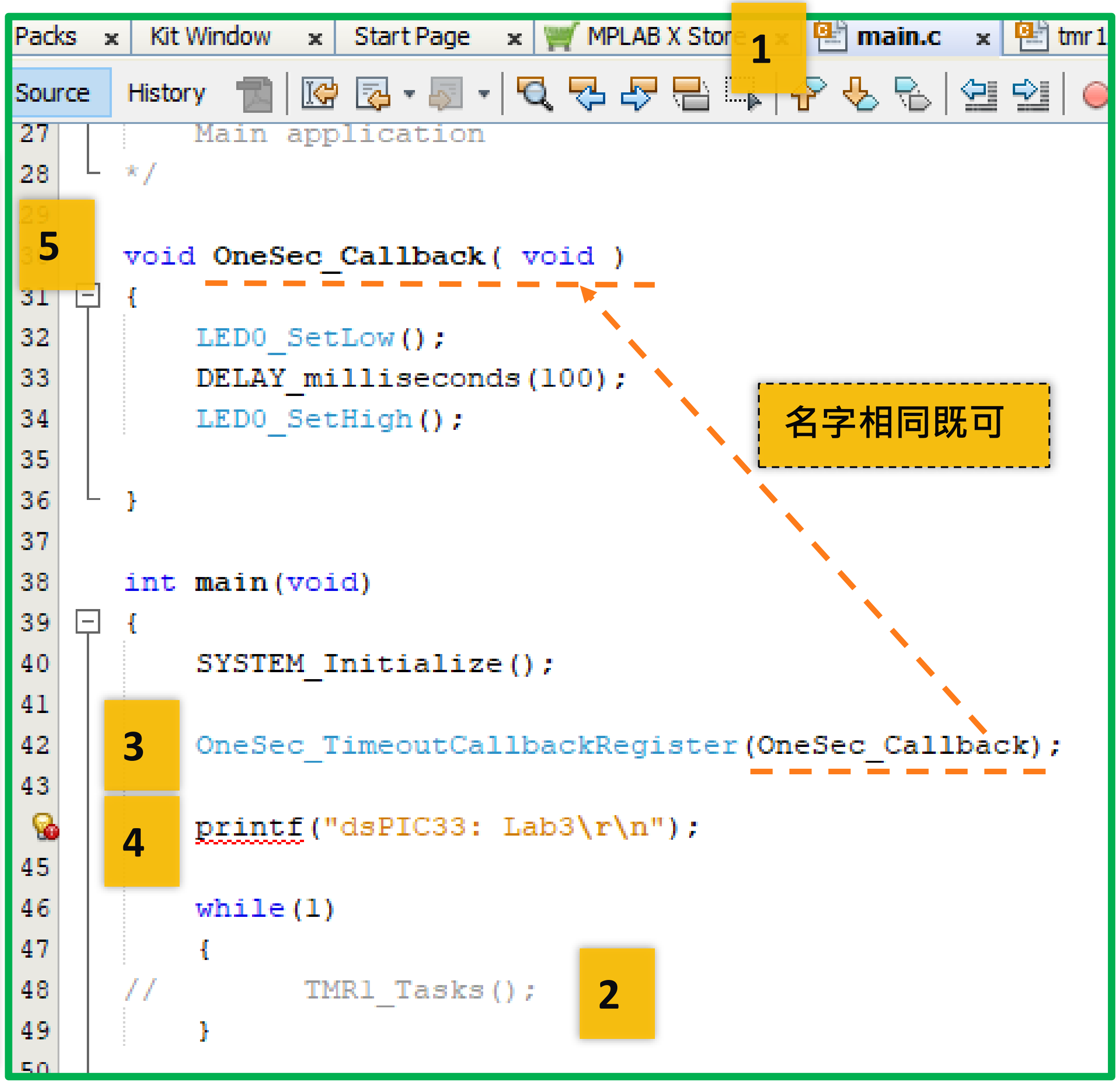
2) 刪除while(1)中之前加的程式

3) 在SYSTEM_Initialize後告知中斷時要呼叫的函式
OneSec_TimeoutCallbackRegister(OneSec_Callback);

4) 修改printf(“dsPIC33: Lab3\r\n”);
方便辨識是否是執行新程式

5) 告知中斷後要做的事

```
void OneSec_Callback( void )
{
   LED0_SetLow();
   DELAY_milliseconds(100);
   LED0_SetHigh();
}
```



```
      Main application
   */

   void OneSec_Callback( void )
   {
       LED0_SetLow();
       DELAY_milliseconds(100);
       LED0_SetHigh();

   }

   int main(void)
   {
       SYSTEM_Initialize();

       OneSec_TimeoutCallbackRegister(OneSec_Callback);

       printf("dsPIC33: Lab3\r\n");

       while(1)
       {
   //          TMR1_Tasks();
       }
```

名字相同既可

© 2024 Microchip Technology Inc. and its subsidiaries

12.結果

Lab3

1)燒錄測試

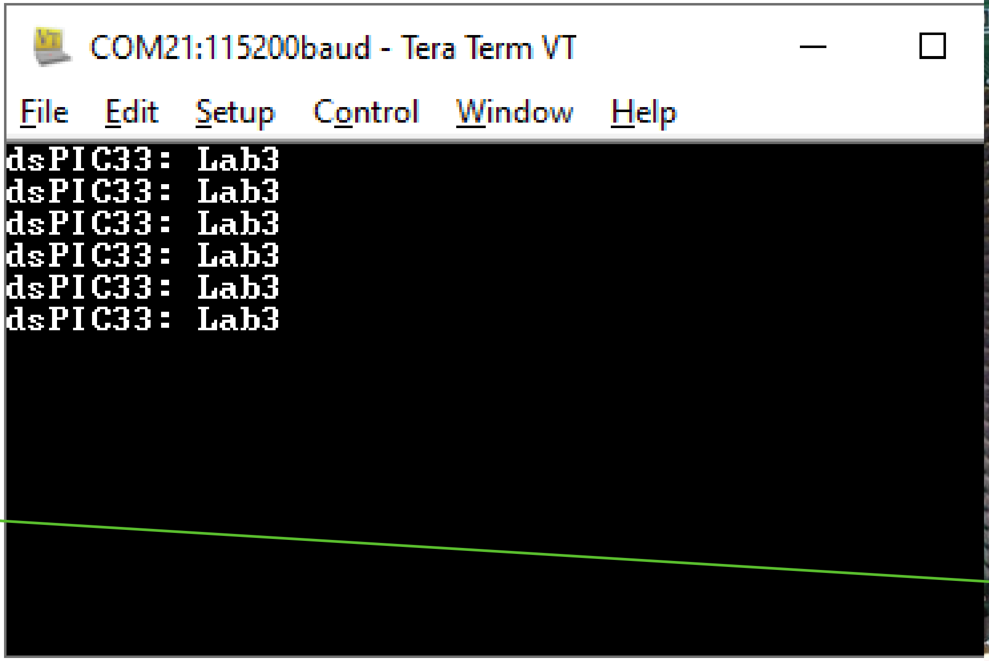
2)成果

紅板上LED0

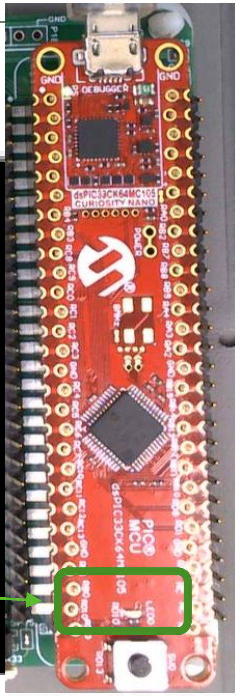
每秒亮100ms

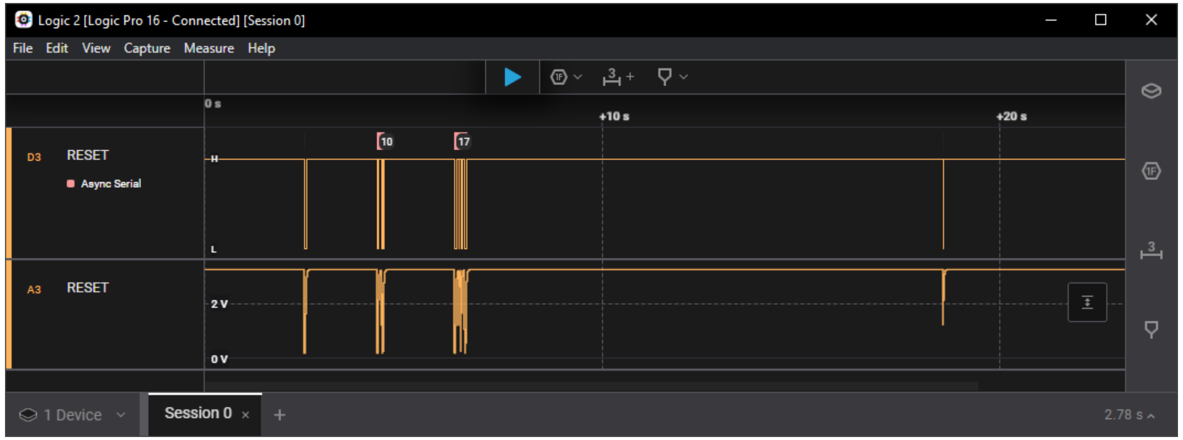
PC顯示 dsPIC33: Lab3

- 會顯示很多行dsPIC33: Lab3?
- 是因為燒錄時會拉放RESET多次


COM21:115200baud - Tera Term VT
File Edit Setup Control Window Help

```
dsPIC33: Lab3
dsPIC33: Lab3
dsPIC33: Lab3
dsPIC33: Lab3
dsPIC33: Lab3
dsPIC33: Lab3
```




Logic 2 [Logic Pro 16 - Connected] [Session 0]
File Edit View Capture Measure Help
0 s
+10 s
+20 s
10
17
D3 RESET
Async Serial
H
L
A3 RESET
2 V
0 V
1 Device
Session 0
2.78 s

實驗四

Timer(SCCP)

內建硬體程式開發

實驗四：Timer(SCCP)

- 本實驗目的：
  - 架設計時器
    - 用一個SCCP完成兩個16-bit Timer
    - 分別控制底板的LED1 ,LED2 /LED3, LED4

- 結果
  - LED1,2 亮1 Hz
  - LED3,4 亮2 Hz

背景知識

dsPIC33CK datasheet

SCCP架構

SCCP(Single Output Capture/Compare/PWM/Timer modules)

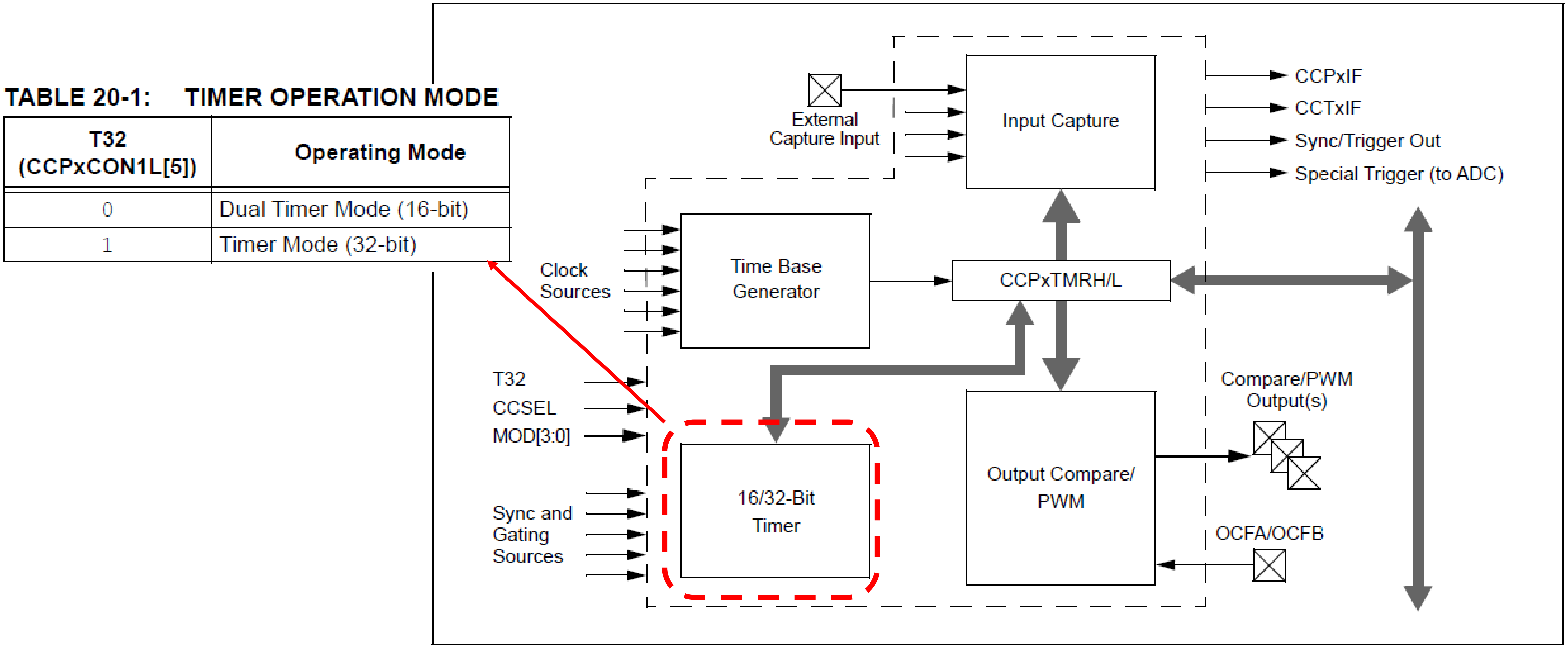


TABLE 20-1: TIMER OPERATION MODE

T32 (CCPxCON1L[5])	Operating Mode
0	Dual Timer Mode (16-bit)
1	Timer Mode (32-bit)

FIGURE 20-1: SCCPx CONCEPTUAL BLOCK DIAGRAM

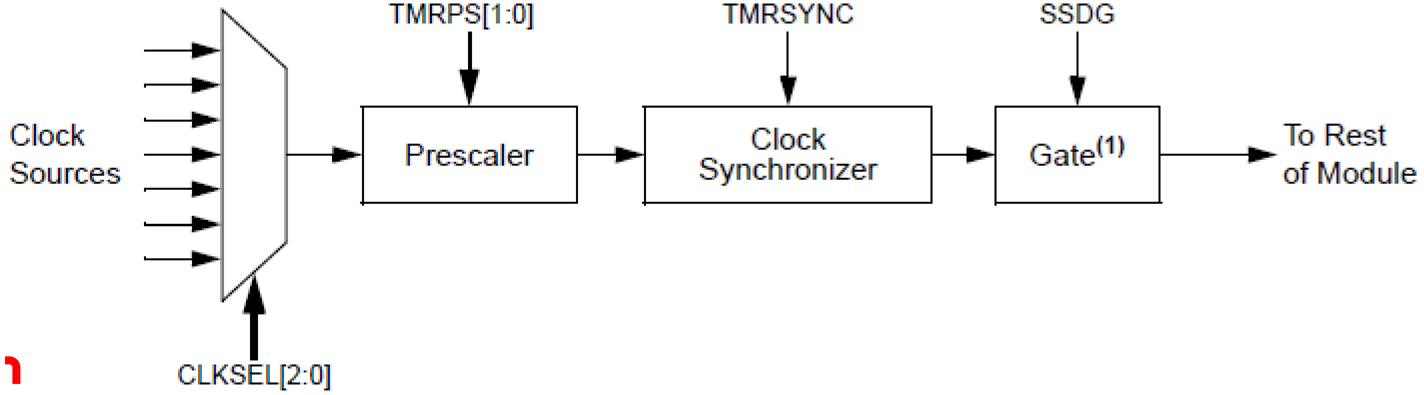
SCCP架構

SCCP(Single Output Capture/Compare/PWM/Timer modules)

CLKSEL[2:0]: CCPx Time Base Clock Select bits

111 = PPS TxCK input
110 = CLC4
101 = CLC3
100 = CLC2
011 = CLC1
010 = Reserved
001 = Reference Clock (REFCLKO)
000 = Peripheral Clock (F_P = $F_{OSC}/2$)

FIGURE 20-2: TIMER CLOCK GENERATOR



Note 1: Gating is available in Timer modes only.

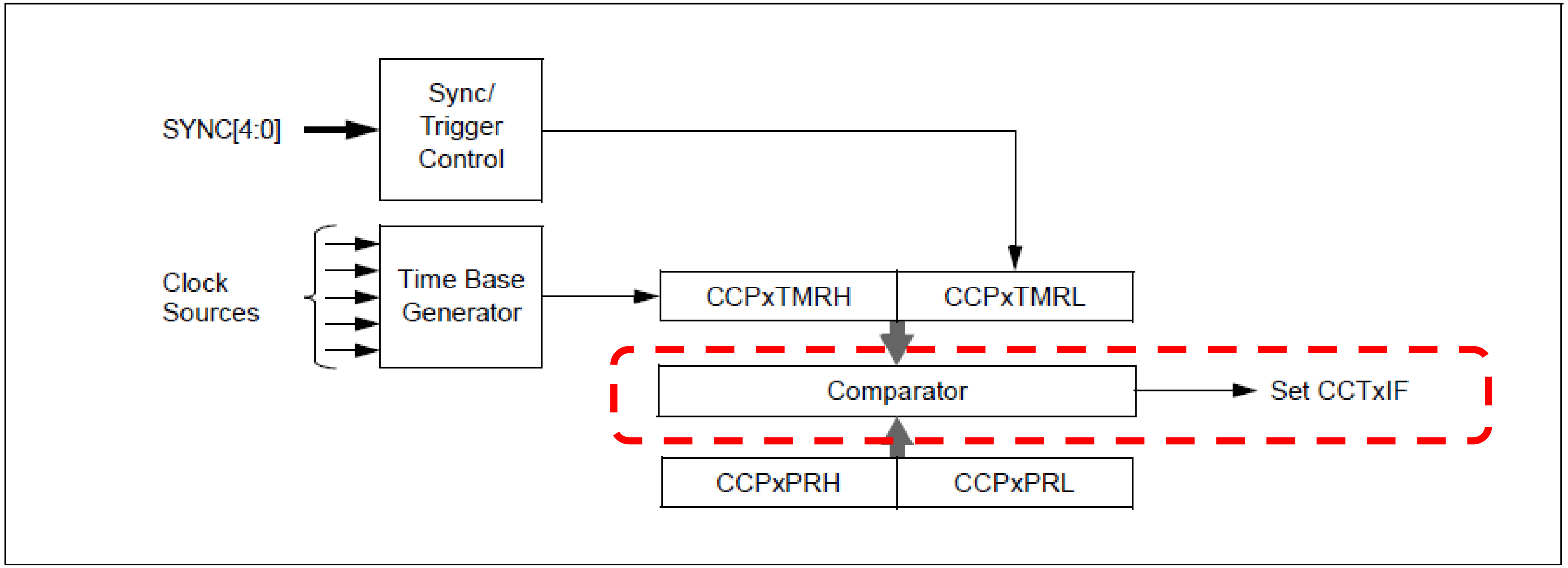
SCCP架構

設定成：一個32-bit 計時器

TABLE 20-5: SYNCHRONIZATION SOURCES

SYNC[4:0]	Synchronization Source
00000	None; Timer with Rollover on CCPxPR Match or FFFFh
10100	UART3 RX Edge Detect
10101	UART3 TX Edge Detect
10111	Comparator 1 Output
11000-11110	Reserved
11111	None; Timer with Auto-Rollover (FFFFh → 0000h)

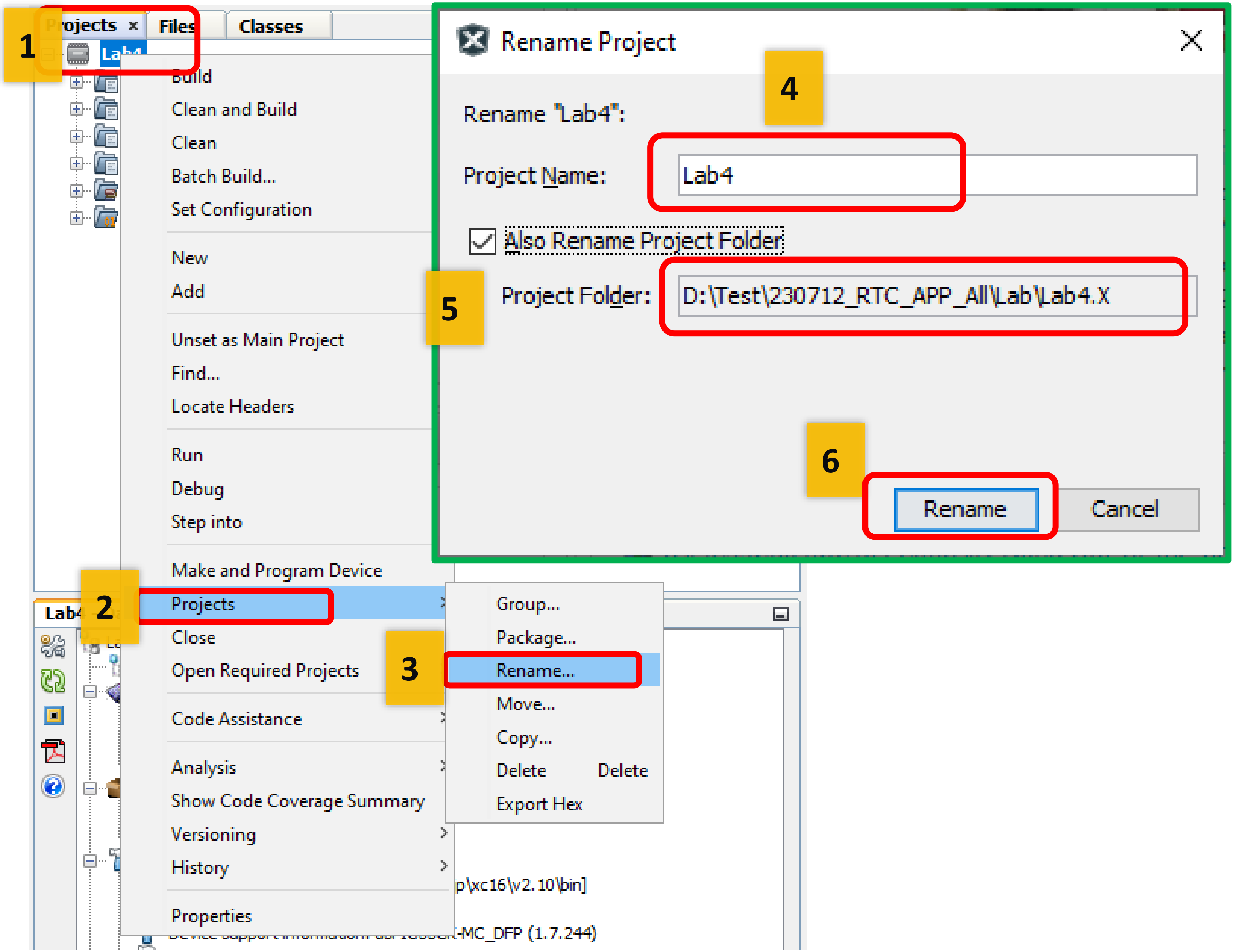
FIGURE 20-4: 32-BIT TIMER MODE



1.建立Lab4

將Lab 3改名成Lab4

1) 到Projects 在Lab3上
   滑鼠點右鍵
2) 找到Projects
3) 找到Rename
4) 修改成Lab4
5) 勾選順便改資料夾名
6) 按 Rename - 完成名稱修改



Rename Project

Rename "Lab4":

Project Name: Lab4

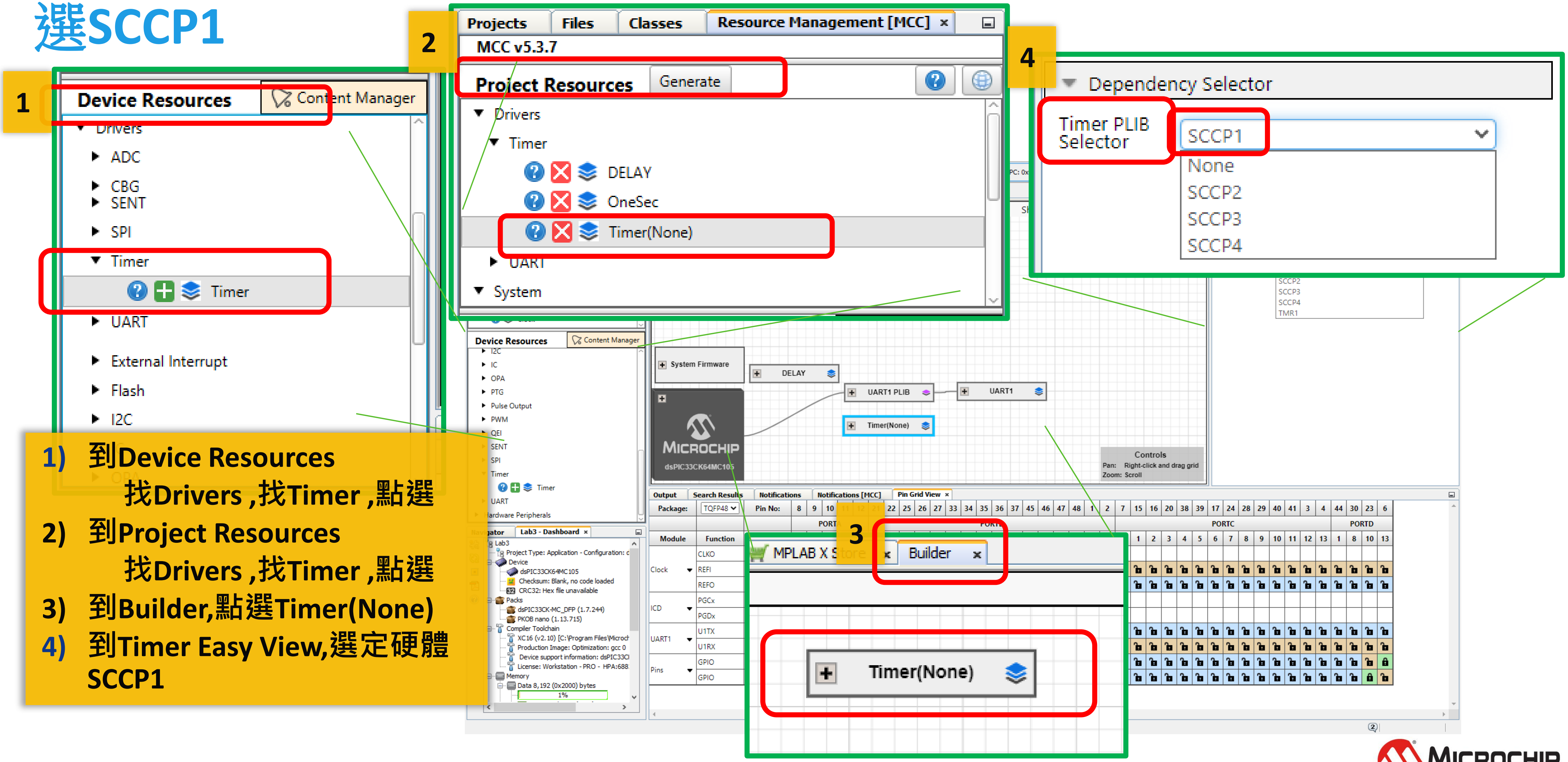
☑ Also Rename Project Folder

Project Folder: D:\Test\230712_RTC_APP_All\Lab\Lab4.X

Rename Cancel

2. 再加一組Timer

選SCCP1



1) 到Device Resources
找Drivers ,找Timer ,點選

2) 到Project Resources
找Drivers ,找Timer ,點選

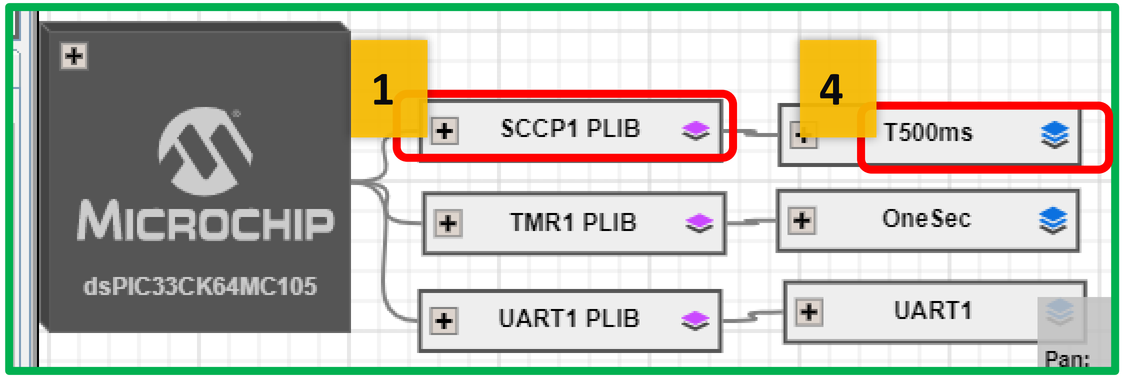
3) 到Builder,點選Timer(None)

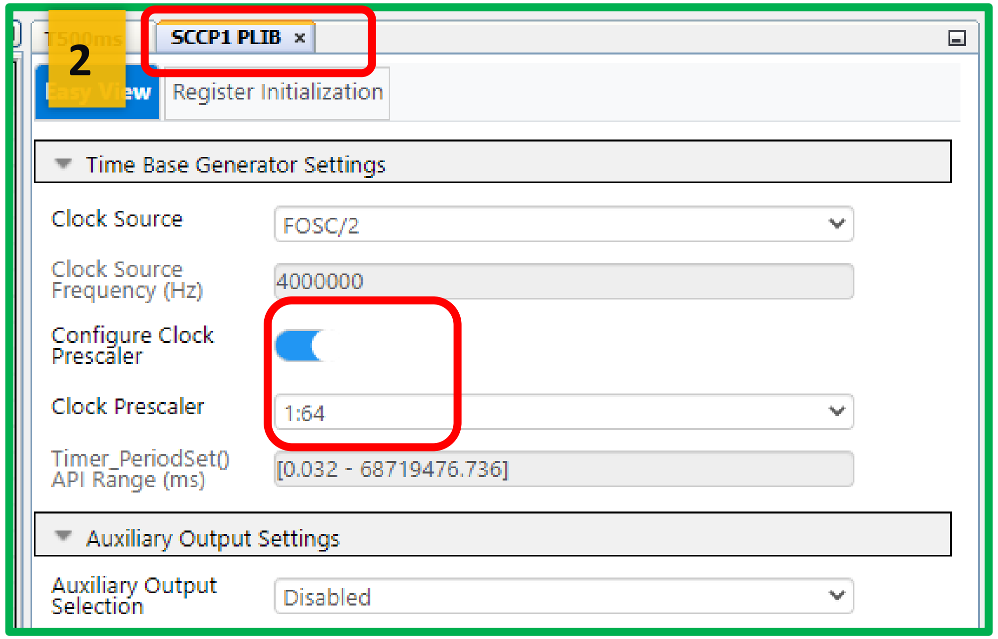
4) 到Timer Easy View,選定硬體 SCCP1

2. 設定SCCP

選SCCP1

1) 選SCCP1 PLIB
2) 設Clock Prescaler 1:64
3) 選Timer1 設，Custom Name 改名 T500ms 500ms , 開Interrupt
4) 名字連動成T500ms





Time Base Generator Settings

Clock Source: FOSC/2

Clock Source Frequency (Hz): 4000000

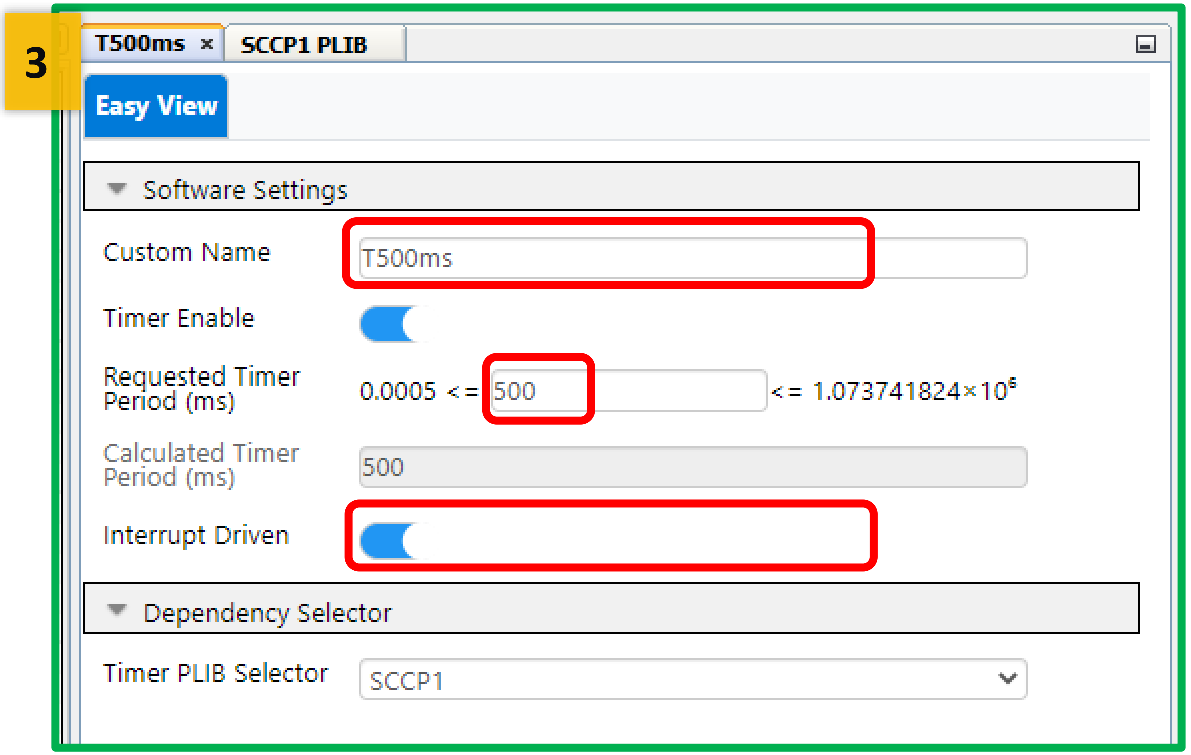
Configure Clock Prescaler: ●

Clock Prescaler: 1:64

Timer_PeriodSet() API Range (ms): [0.032 - 68719476.736]

Auxiliary Output Settings

Auxiliary Output Selection: Disabled



Software Settings

Custom Name: T500ms

Timer Enable: ●

Requested Timer Period (ms): 0.0005 <= 500 <= 1.073741824×10⁶

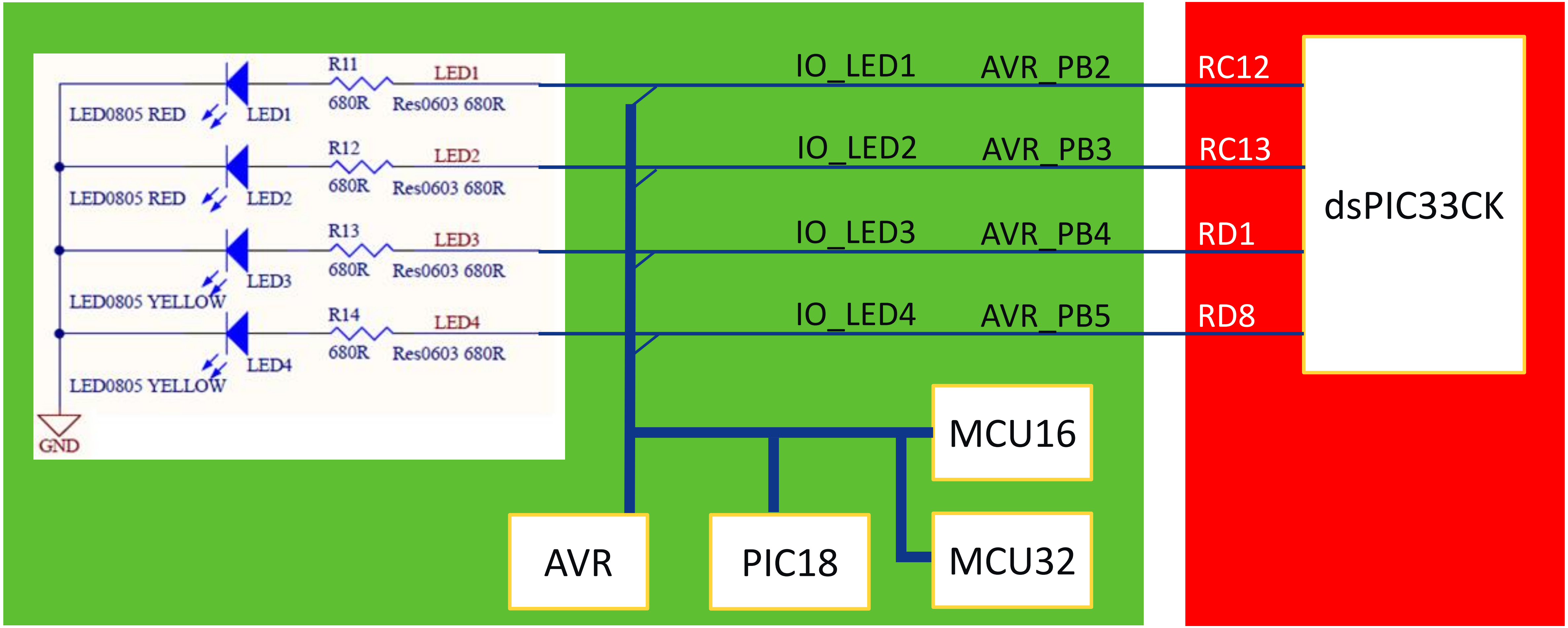
Calculated Timer Period (ms): 500

Interrupt Driven: ●

Dependency Selector

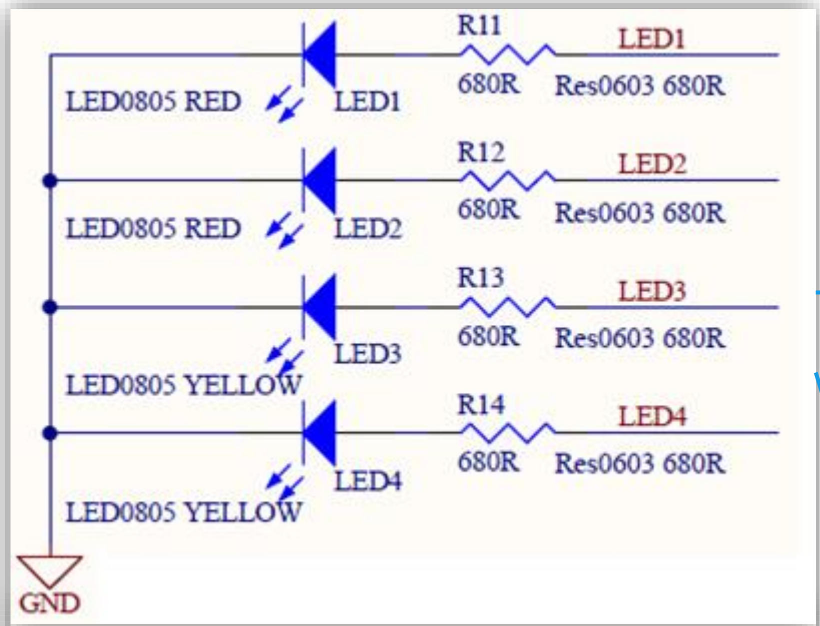
Timer PLIB Selector: SCCP1

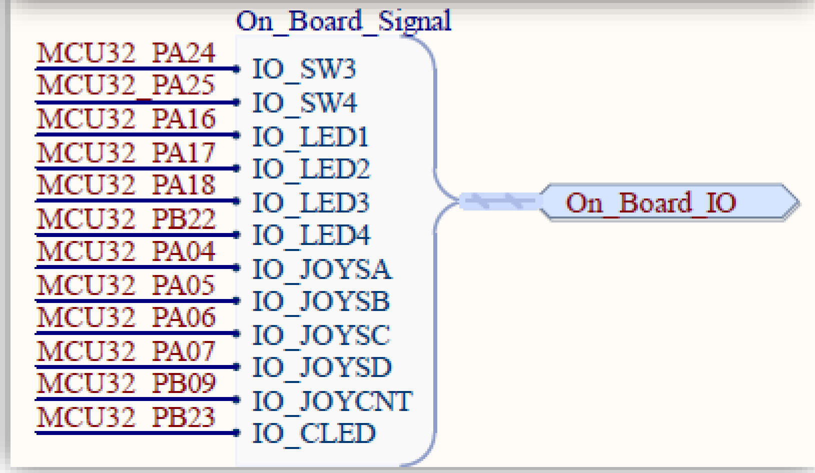
3. LED架構圖

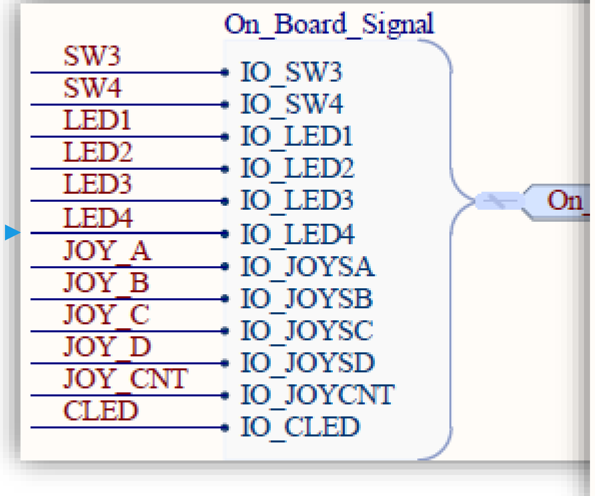

R11
LED1
680R
Res0603 680R
LED0805 RED
LED1
R12
LED2
680R
Res0603 680R
LED0805 RED
LED2
R13
LED3
680R
Res0603 680R
LED3
LED0805 YELLOW
R14
LED4
680R
Res0603 680R
LED4
LED0805 YELLOW
GND
IO_LED1
AVR_PB2
RC12
IO_LED2
AVR_PB3
RC13
IO_LED3
AVR_PB4
RD1
IO_LED4
AVR_PB5
RD8
dsPIC33CK
MCU16
MCU32
AVR
PIC18

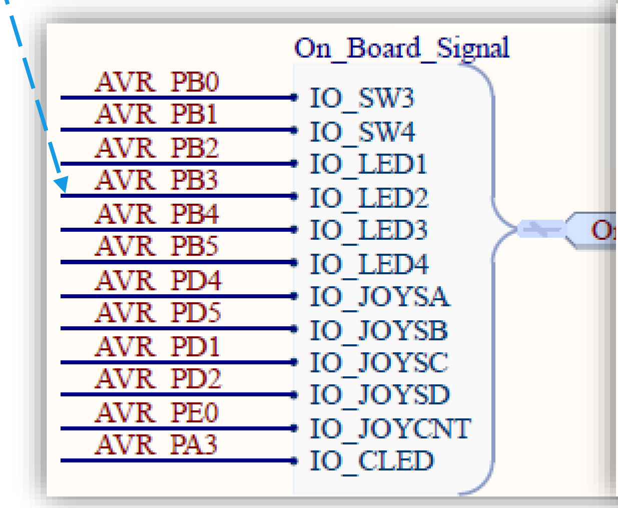
3-1. LED架構圖

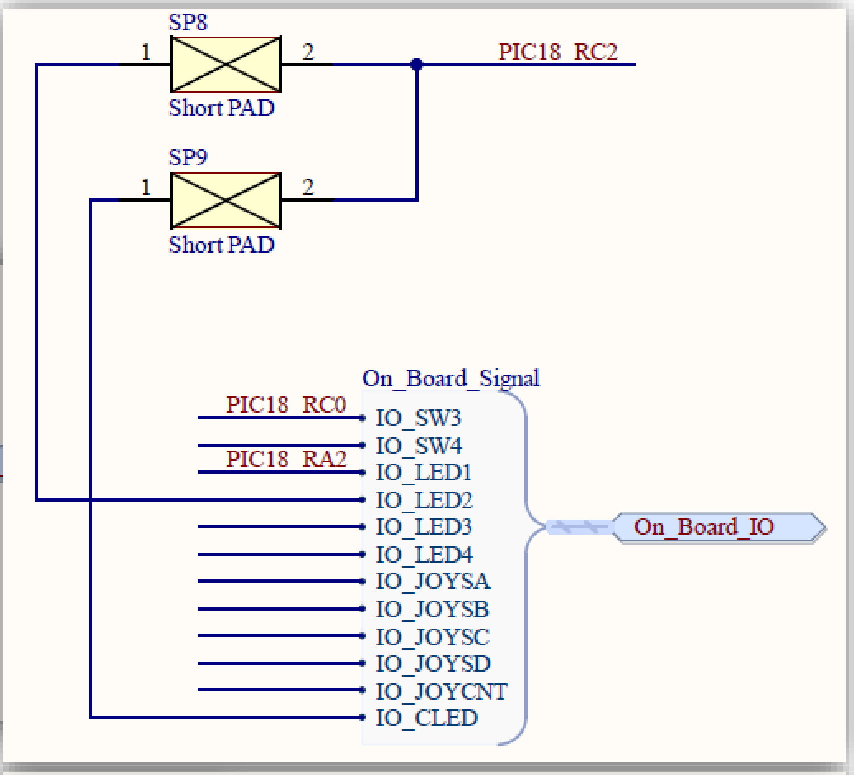
LED電路 (APP_CuriosityNano2Arduino_V20230315.pdf)

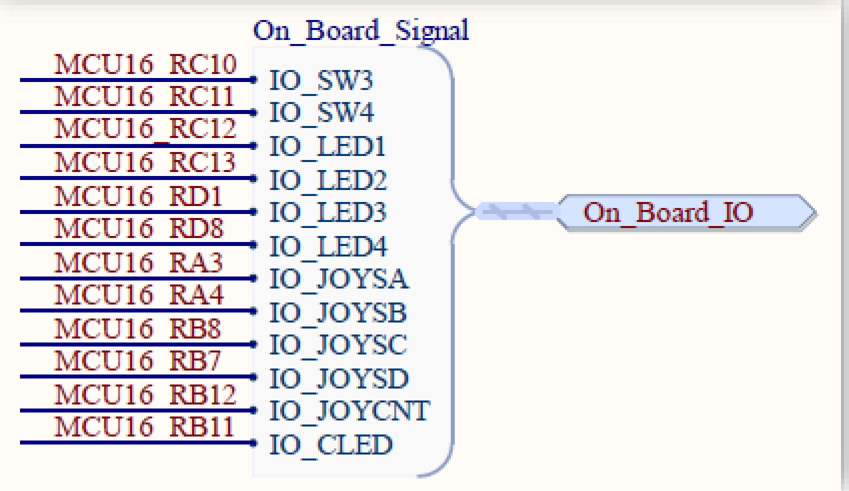






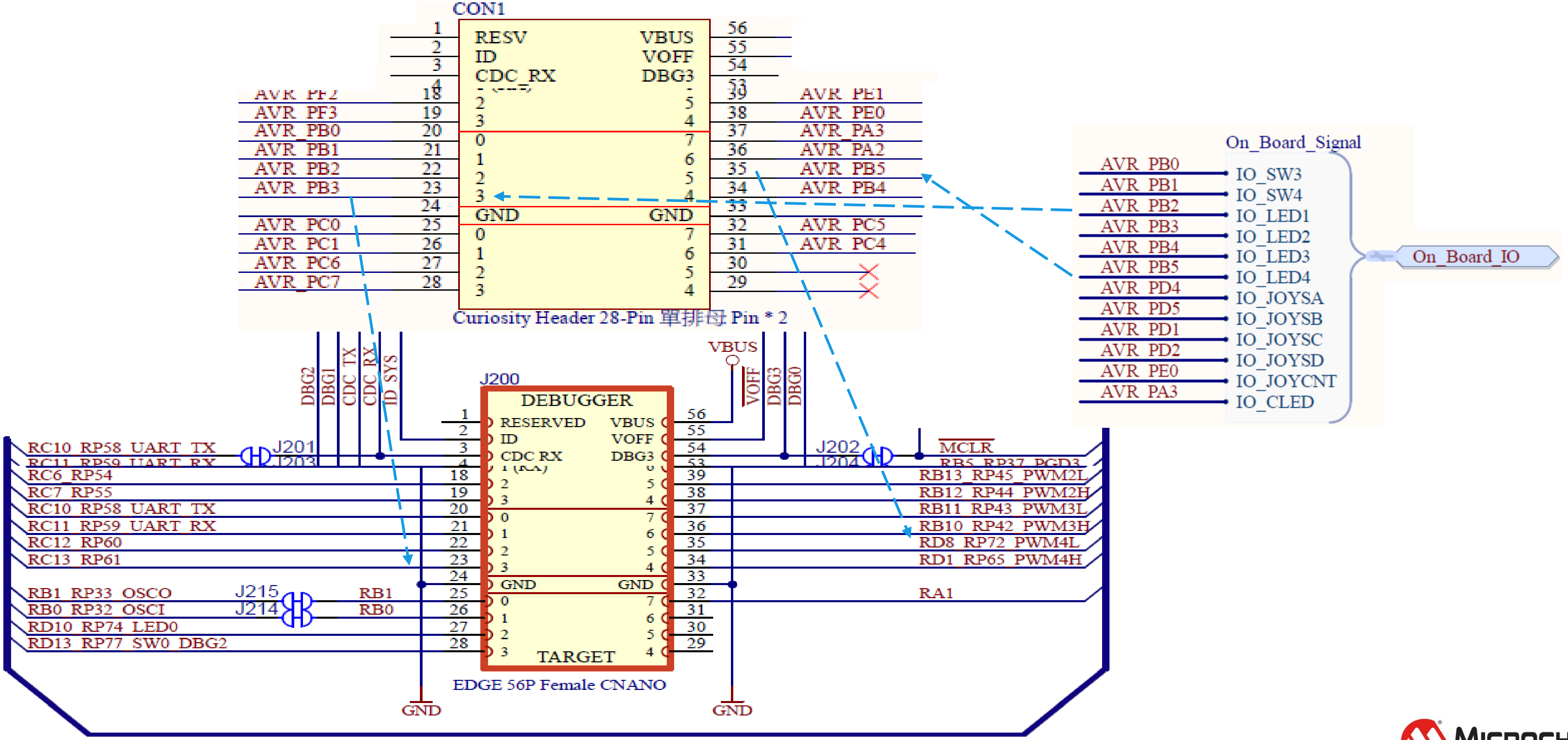




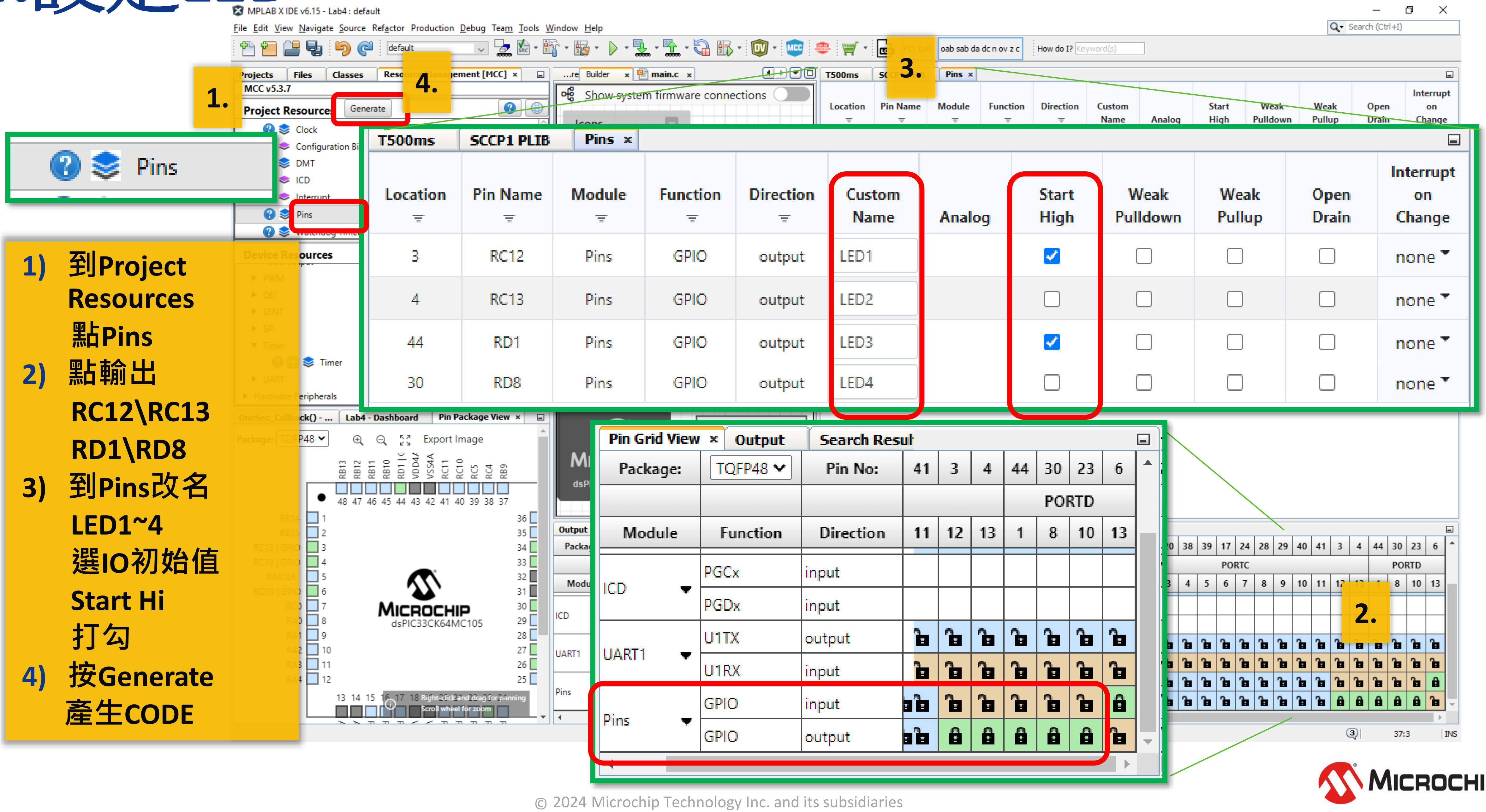


3-2. LED架構圖

LED電路 (APP_CuriosityNano2Arduino_V20230315.pdf \ dsPIC33CK64MC105_Curiosity_Nano_Schematics)



4.設定LED



Pins

T500ms | SCCP1 PLIB | Pins

Location	Pin Name	Module	Function	Direction	Custom Name	Analog	Start High	Weak Pulldown	Weak Pullup	Open Drain	Interrupt on Change
3	RC12	Pins	GPIO	output	LED1		☑	☐	☐	☐	none
4	RC13	Pins	GPIO	output	LED2		☐	☐	☐	☐	none
44	RD1	Pins	GPIO	output	LED3		☑	☐	☐	☐	none
30	RD8	Pins	GPIO	output	LED4		☐	☐	☐	☐	none

Pin Grid View | Output | Search Resul

Package:	TQFP48	Pin No:	41	3	4	44	30	23	6
						PORTD			
Module	Function	Direction	11	12	13	1	8	10	13
ICD	PGCx	input							
	PGDx	input							
UART1	U1TX	output	🔓	🔓	🔓	🔓	🔓	🔓	🔓
	U1RX	input	🔓	🔓	🔓	🔓	🔓	🔓	🔓
Pins	GPIO	input	🔓	🔓	🔓	🔓	🔓	🔓	🔒
	GPIO	output	🔓	🔒	🔒	🔒	🔒	🔒	🔓

1) 到Project Resources 點Pins
2) 點輸出 RC12\RC13 RD1\RD8
3) 到Pins改名 LED1~4 選IO初始值 Start Hi 打勾
4) 按Generate 產生CODE

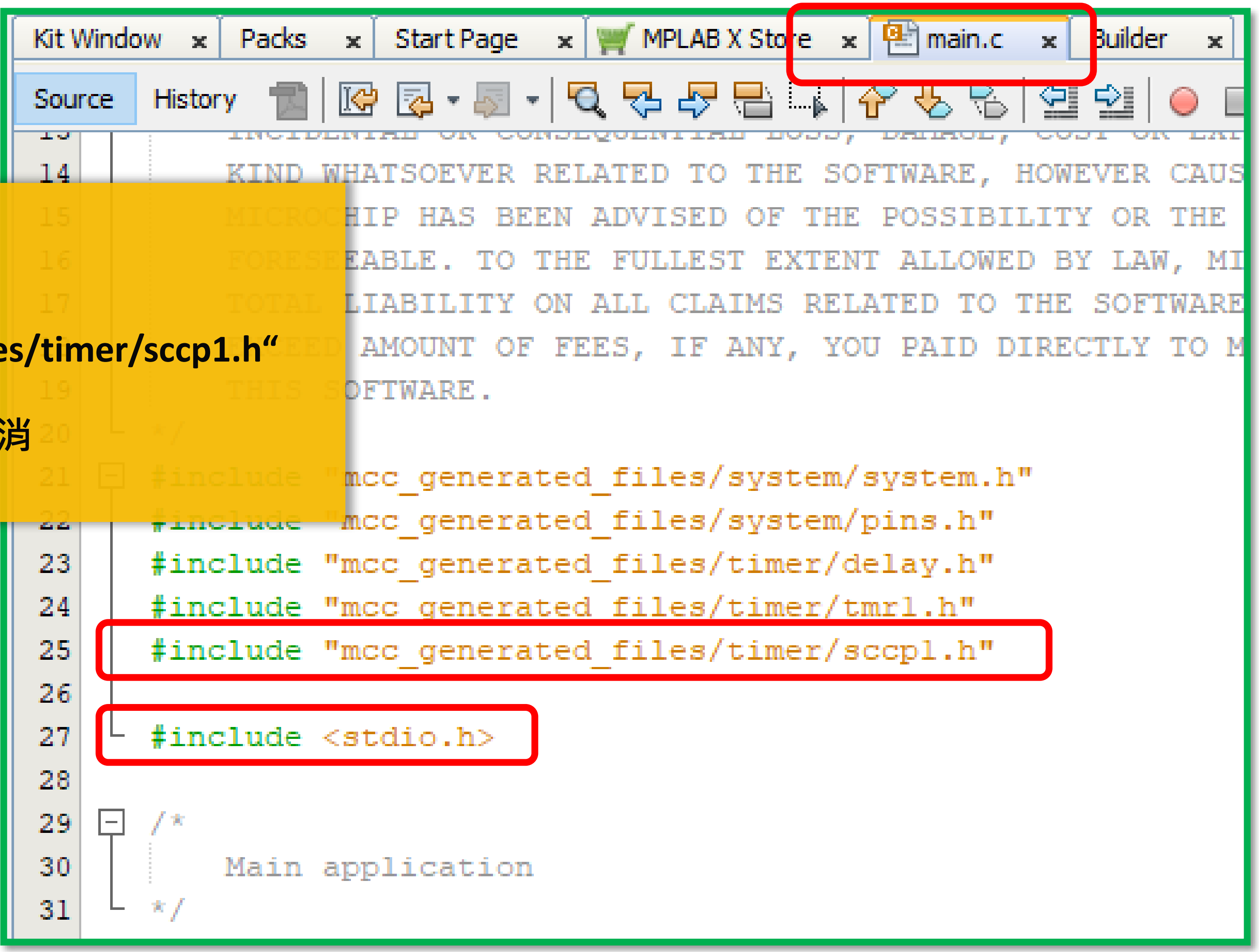
5. 寫CODE

sccp1.h

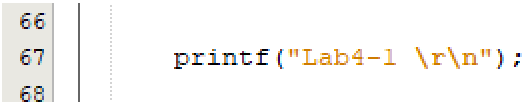
1) 在main.c

2) 開頭加入
#include "mcc_generated_files/timer/sccp1.h“

PS, 如果要讓printf (); 警告取消
加入#include <stdio.h>



```
    INCIDENTIAL OR CONSEQUENTIAL LOSS, DAMAGE, COST OR EXP
    KIND WHATSOEVER RELATED TO THE SOFTWARE, HOWEVER CAUS
    MICROCHIP HAS BEEN ADVISED OF THE POSSIBILITY OR THE
    FORESEEABLE. TO THE FULLEST EXTENT ALLOWED BY LAW, MI
    TOTAL LIABILITY ON ALL CLAIMS RELATED TO THE SOFTWARE
    EXCEED AMOUNT OF FEES, IF ANY, YOU PAID DIRECTLY TO M
    THIS SOFTWARE.
*/
#include "mcc_generated_files/system/system.h"
#include "mcc_generated_files/system/pins.h"
#include "mcc_generated_files/timer/delay.h"
#include "mcc_generated_files/timer/tmr1.h"
#include "mcc_generated_files/timer/sccp1.h"

#include <stdio.h>

/*
    Main application
*/
```



```
    printf("Lab4-1 \r\n");
```

5-1. 寫Code

1) 繼續在 main.c

2) 加入

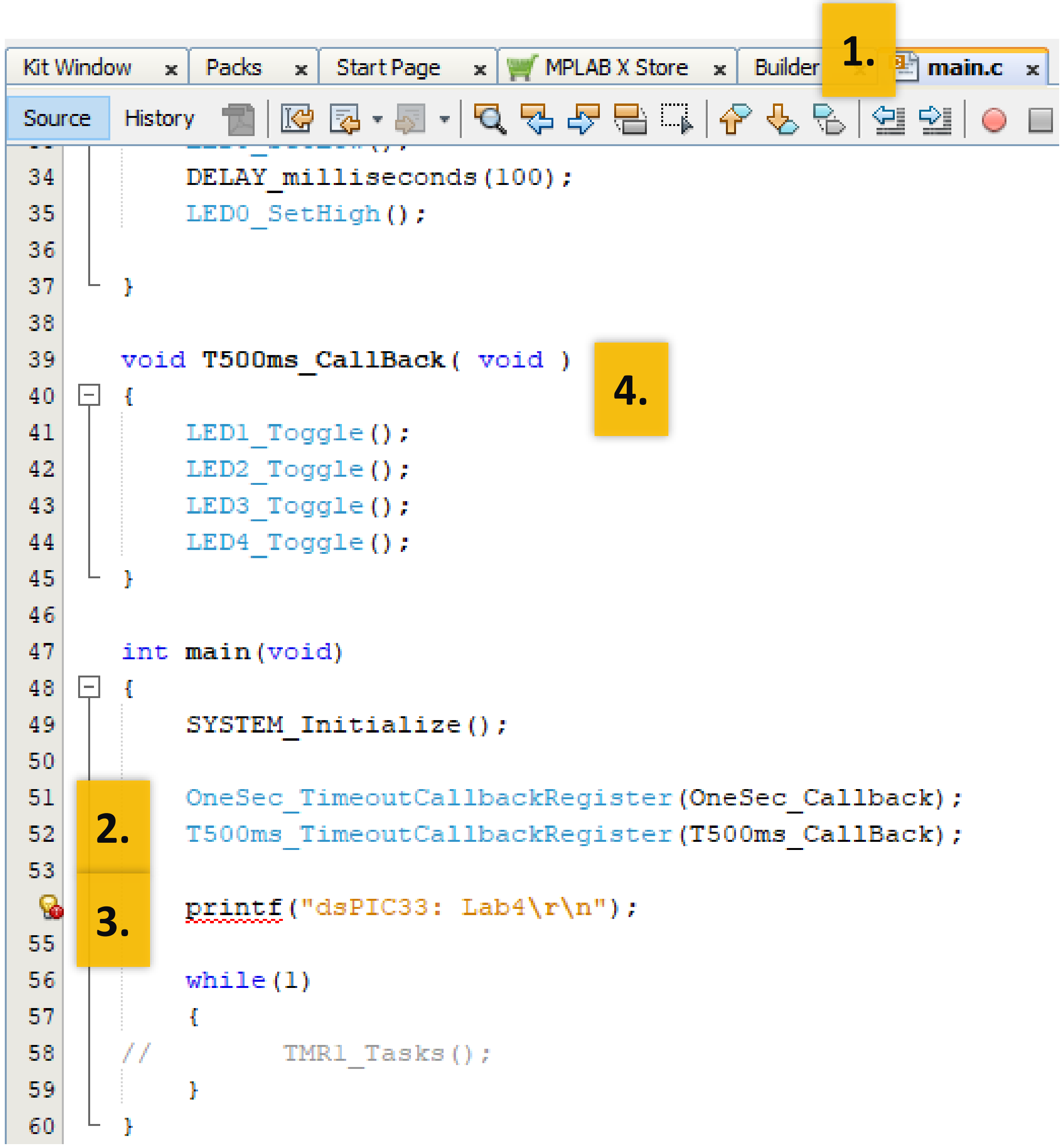
T500ms_TimeoutCallbackRegister(T500ms_CallBack);

3) 改 printf(“dsPIC33: Lab4\r\n");

4) 加入

```
void T500ms_CallBack( void )
{
   LED1_Toggle();
   LED2_Toggle();
   LED3_Toggle();
   LED4_Toggle();
}
```

5) 燒錄



```
        DELAY_milliseconds(100);
        LED0_SetHigh();

    }

    void T500ms_CallBack( void )
    {
        LED1_Toggle();
        LED2_Toggle();
        LED3_Toggle();
        LED4_Toggle();
    }

    int main(void)
    {
        SYSTEM_Initialize();

        OneSec_TimeoutCallbackRegister(OneSec_Callback);
        T500ms_TimeoutCallbackRegister(T500ms_CallBack);

        printf("dsPIC33: Lab4\r\n");

        while(1)
        {
    //          TMR1_Tasks();
        }
    }
```

6.結果

Lab4

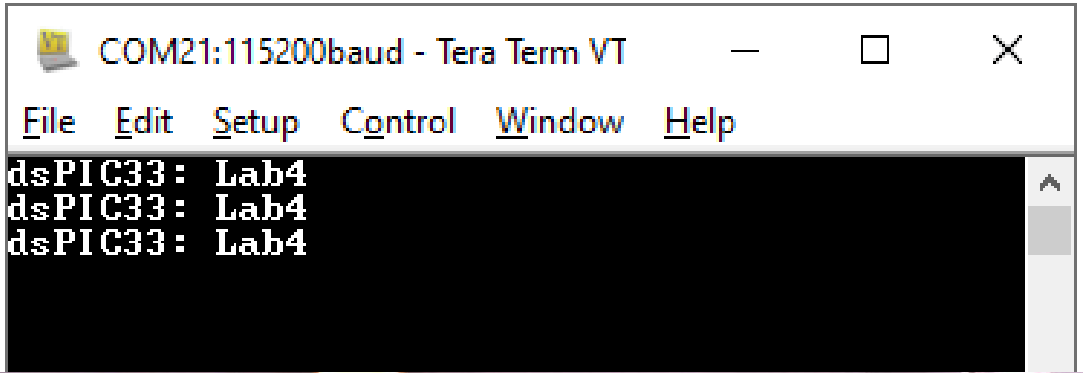
1) 燒錄測試

2) 成果

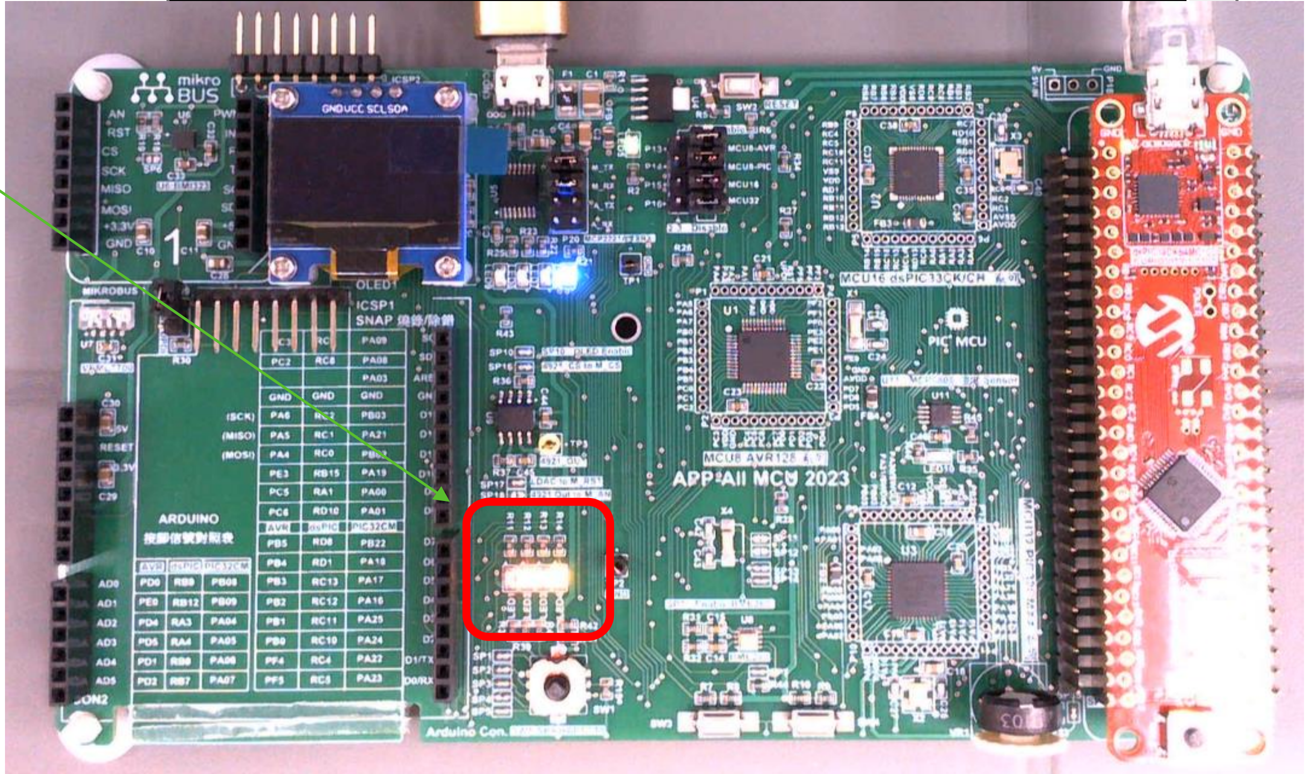
綠板上LED1~4交互閃爍

PC顯示 dsPIC33: Lab4

LED0 持續一秒閃一下



```
dsPIC33: Lab4
dsPIC33: Lab4
dsPIC33: Lab4
```



7. 改成Timer x 2

SCCP(Single Output Capture/Compare/PWM/Timer modules)

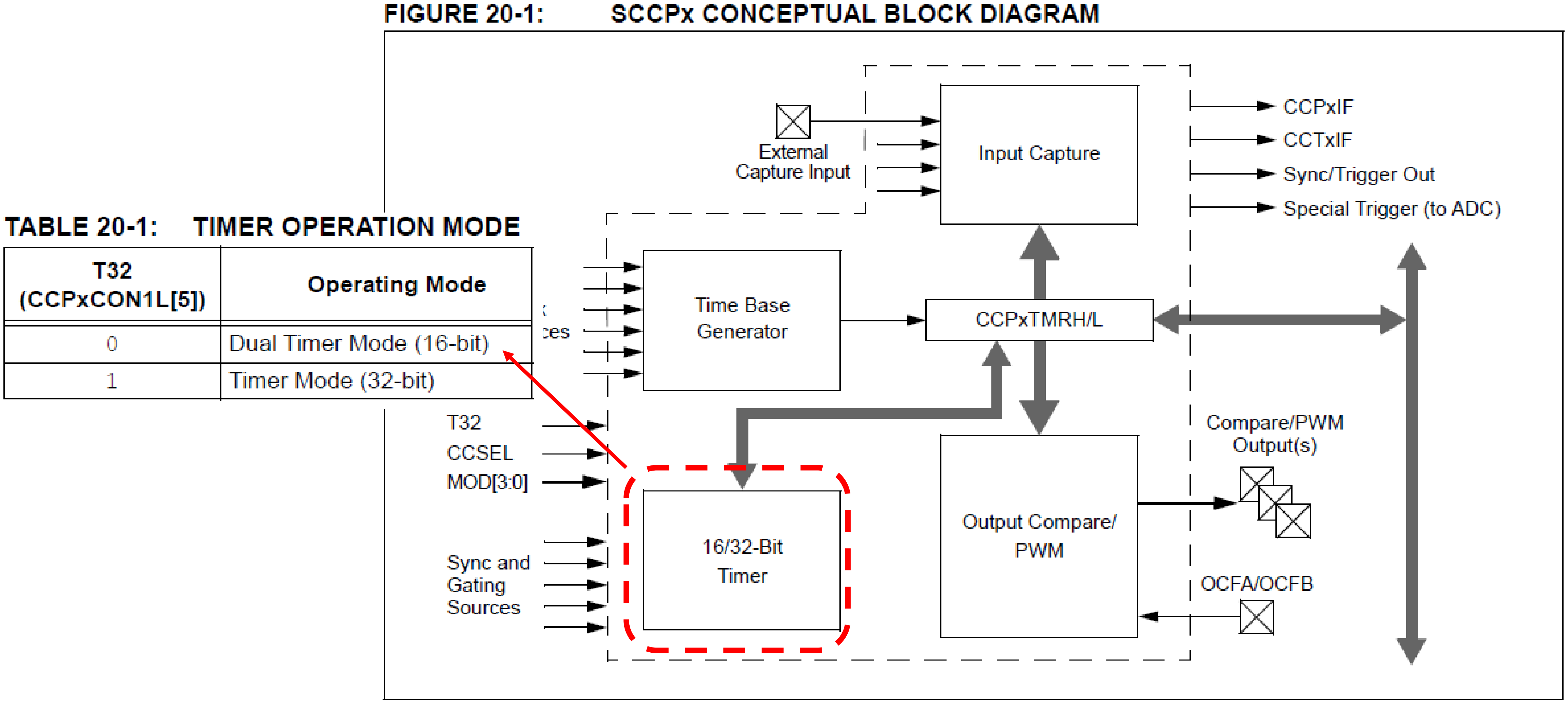


FIGURE 20-1: SCCPx CONCEPTUAL BLOCK DIAGRAM

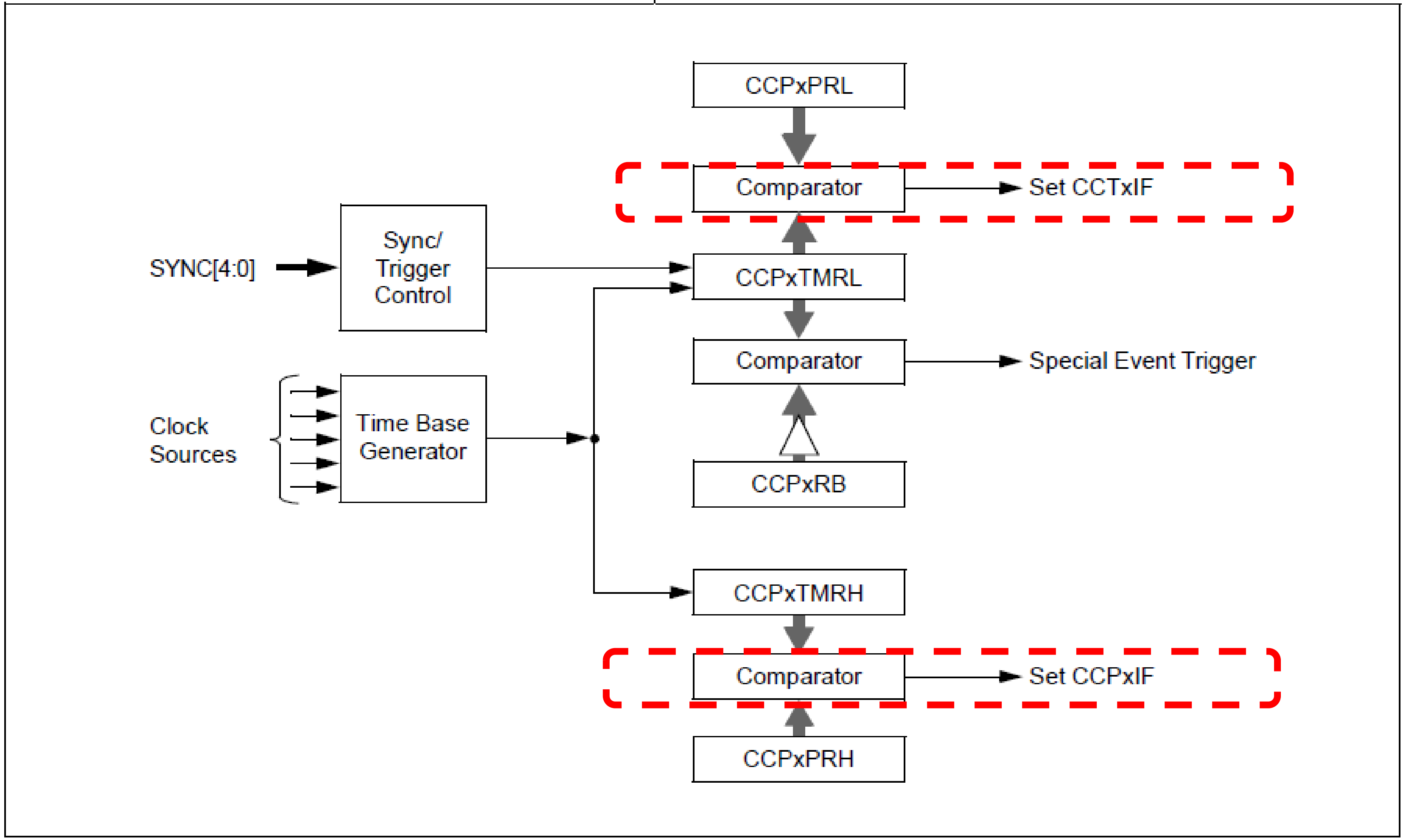
TABLE 20-1: TIMER OPERATION MODE

T32 (CCPxCON1L[5])	Operating Mode
0	Dual Timer Mode (16-bit)
1	Timer Mode (32-bit)

7-1. 改成 Timer x 2

SCCP設定成：兩個16-bit 計時器

FIGURE 20-3: DUAL 16-BIT TIMER MODE



8.設定第二組Timer – 125ms

改暫存器

1) 到main.c

2) 修改void T500ms_CallBack(void) 讓出兩個LED剩

   LED1_Toggle();

   LED2_Toggle();

3) 到SYSTEM_Initialize();後根據啟動第二組Timer

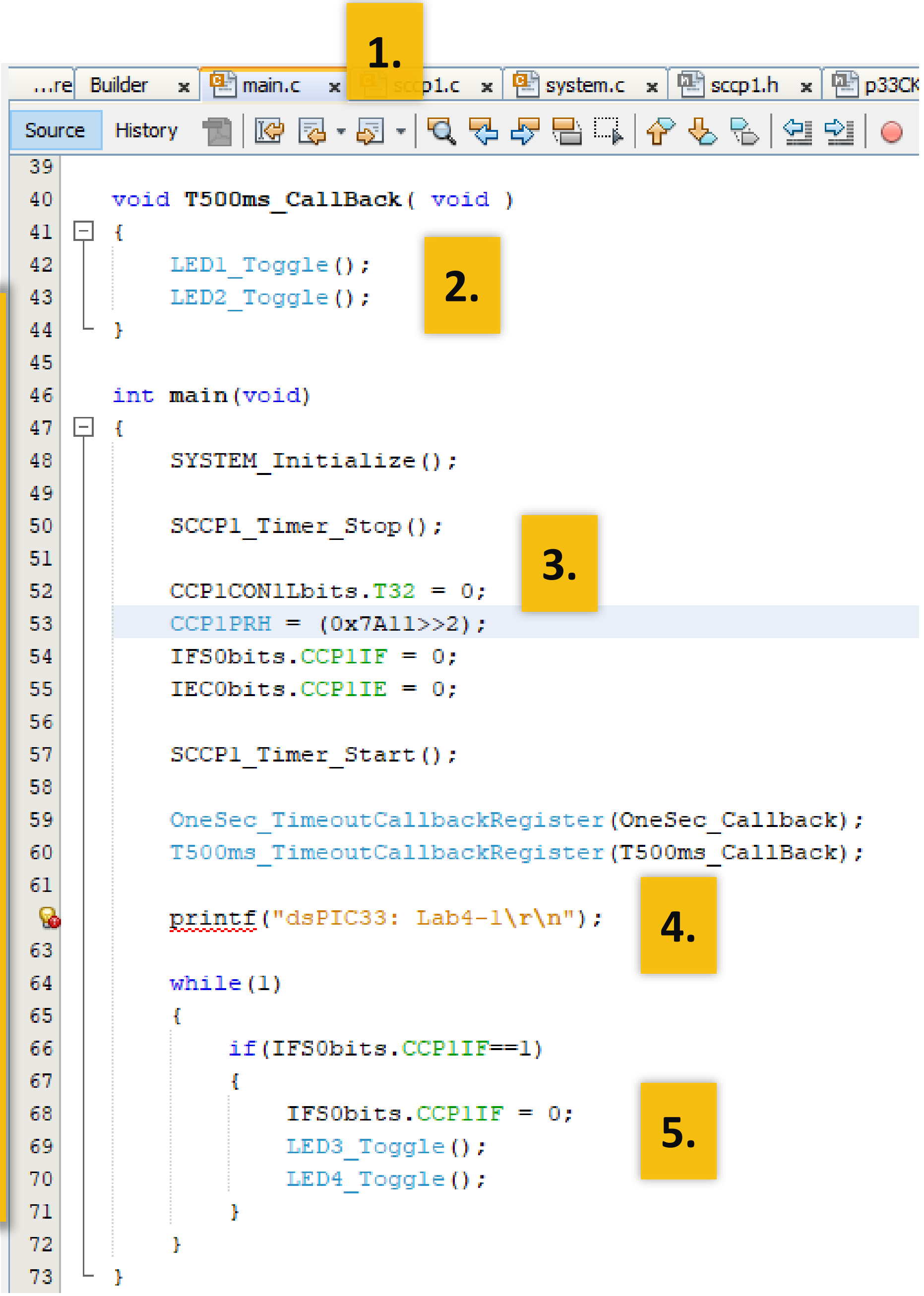
```
SCCP1_Timer_Stop();
CCP1CON1Lbits.T32 = 0;
CCP1PRH = (0x7A11>>2);
IFS0bits.CCP1IF = 0;
IEC0bits.CCP1IE = 0;
SCCP1_Timer_Start();
```

4) 改printf(“dsPIC33: Lab4-1\r\n");

5) 到while(1){　}中

```
if(IFS0bits.CCP1IF==1)
    {
        IFS0bits.CCP1IF = 0;
        LED3_Toggle();
        LED4_Toggle();
    }
```

6) 燒錄



```
void T500ms_CallBack( void )
{
    LED1_Toggle();
    LED2_Toggle();
}

int main(void)
{
    SYSTEM_Initialize();

    SCCP1_Timer_Stop();

    CCP1CON1Lbits.T32 = 0;
    CCP1PRH = (0x7A11>>2);
    IFS0bits.CCP1IF = 0;
    IEC0bits.CCP1IE = 0;

    SCCP1_Timer_Start();

    OneSec_TimeoutCallbackRegister(OneSec_Callback);
    T500ms_TimeoutCallbackRegister(T500ms_CallBack);

    printf("dsPIC33: Lab4-1\r\n");

    while(1)
    {
        if(IFS0bits.CCP1IF==1)
        {
            IFS0bits.CCP1IF = 0;
            LED3_Toggle();
            LED4_Toggle();
        }
    }
}
```

9.結果

Lab4-1

1) 燒錄測試

2) 結果

小板上

LED0閃爍 1Hz

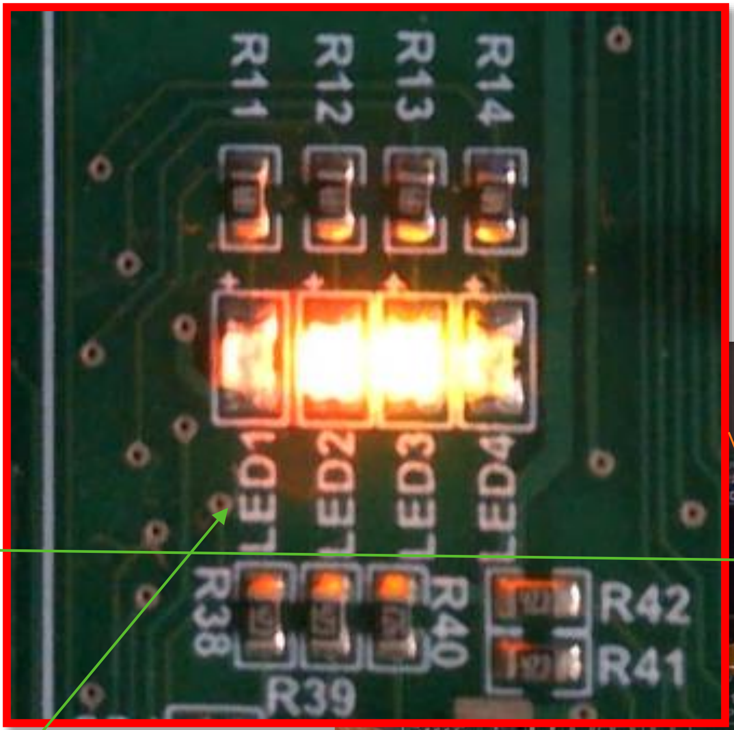
大板上

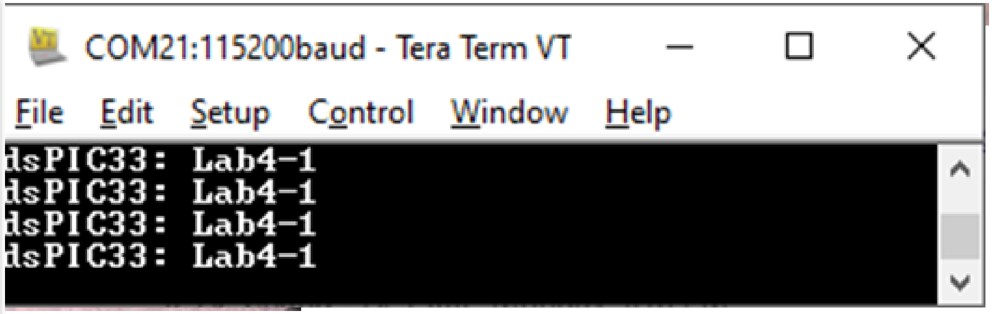
LED1~2交互慢閃 1Hz

LED3~4交互快閃 2Hz

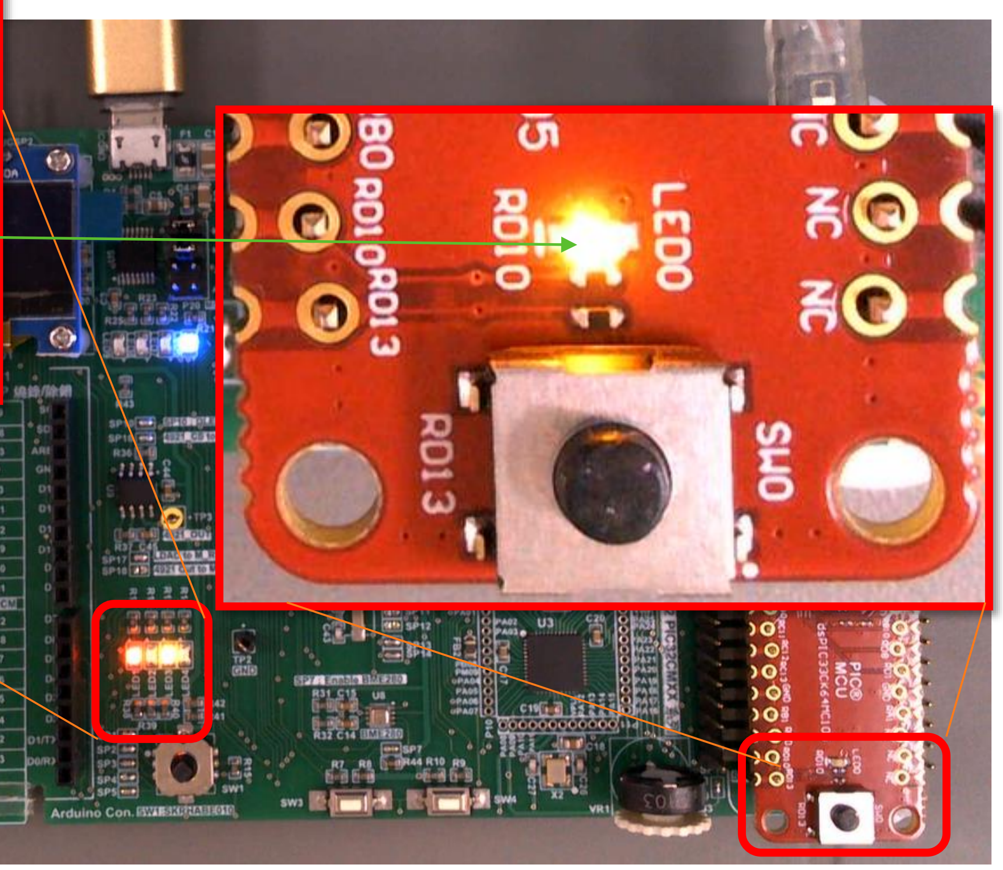
PC顯示 dsPIC33: Lab4-1

LED1 會頓一下?
是因為LED0 delay 100ms造成的





```
dsPIC33: Lab4-1
dsPIC33: Lab4-1
dsPIC33: Lab4-1
dsPIC33: Lab4-1
```



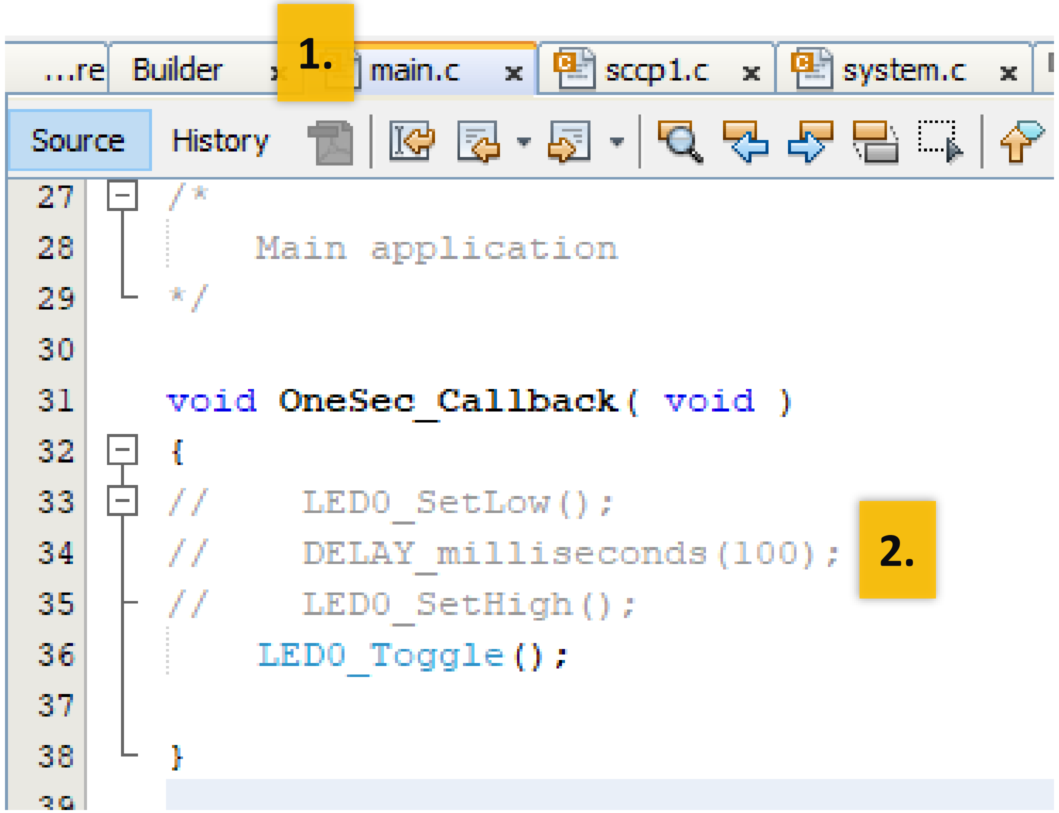
10.刪100ms Delay

再看一次結果

1) 到main.c

2) 修改

```
void OneSec_Callback( void )
{
//   LED0_SetLow();
//   DELAY_milliseconds(100);
//   LED0_SetHigh();
   LED0_Toggle();
}
```



```
/*
    Main application
*/

void OneSec_Callback( void )
{
//     LED0_SetLow();
//     DELAY_milliseconds(100);
//     LED0_SetHigh();
    LED0_Toggle();

}
```

實驗五

Clock

內建硬體程式開發

實驗五：Clock

- 本實驗目的：
  - 了解Clock架構
    - 如何改變System Clock 設定
    - 了解Clock 對 Timer及UART影響
    - 在不同的Clock設定下，達到所需功能
- 結果呈現：
  - UART 115200-8-N-1
  - LED0　亮1 Hz
  - LED1,2 亮1 Hz
  - LED3,4 亮2 Hz

背景知識

dsPIC33CK datasheet

Clocks架構

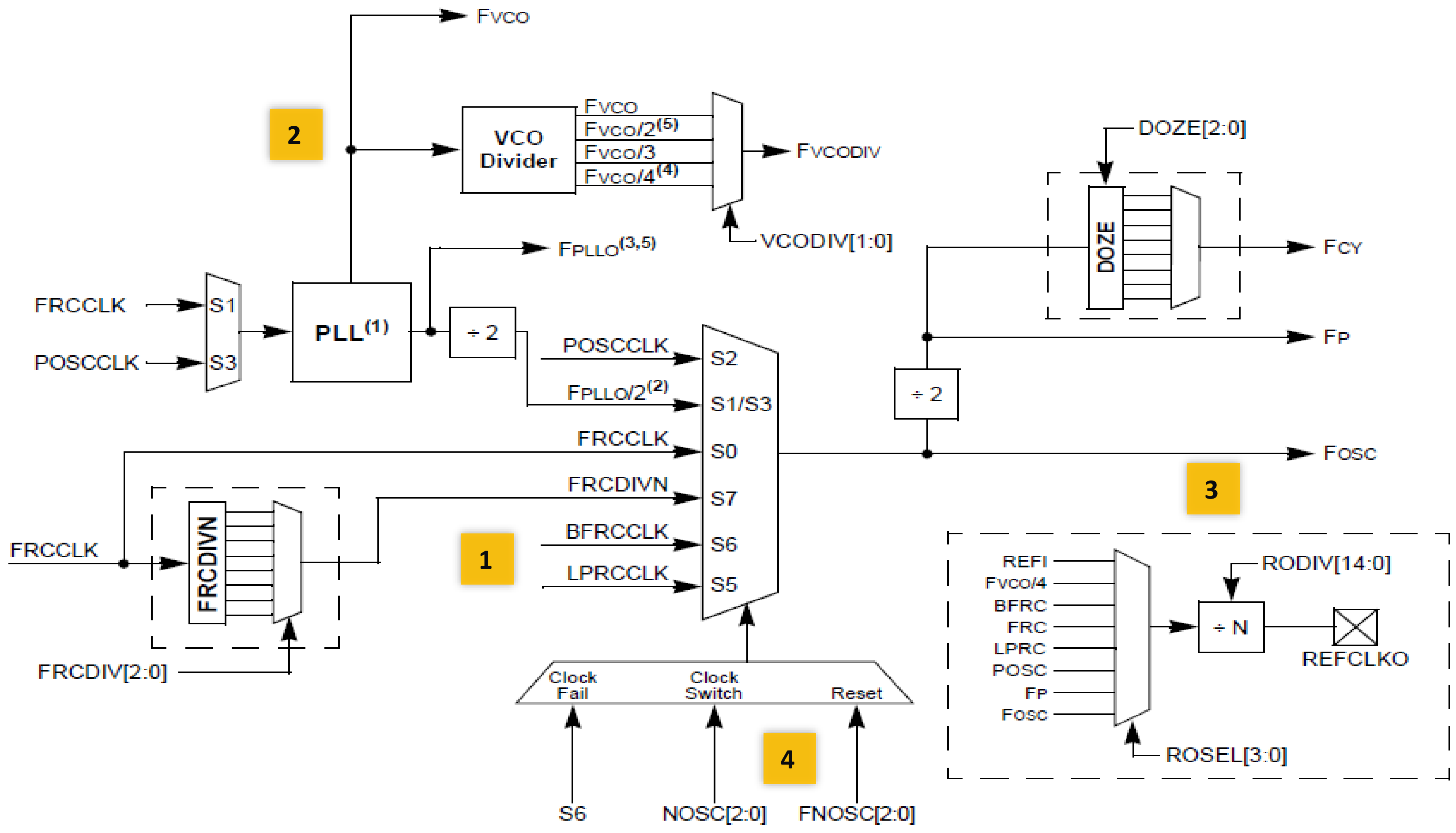
1. 具有多種不同輸入

2. 內含PLL電路
   用來除頻及倍頻

3. 硬體模組
   (Timer,PWM…)
   可選不同時脈來源

4. 軟體運行中
   可以切換Clock

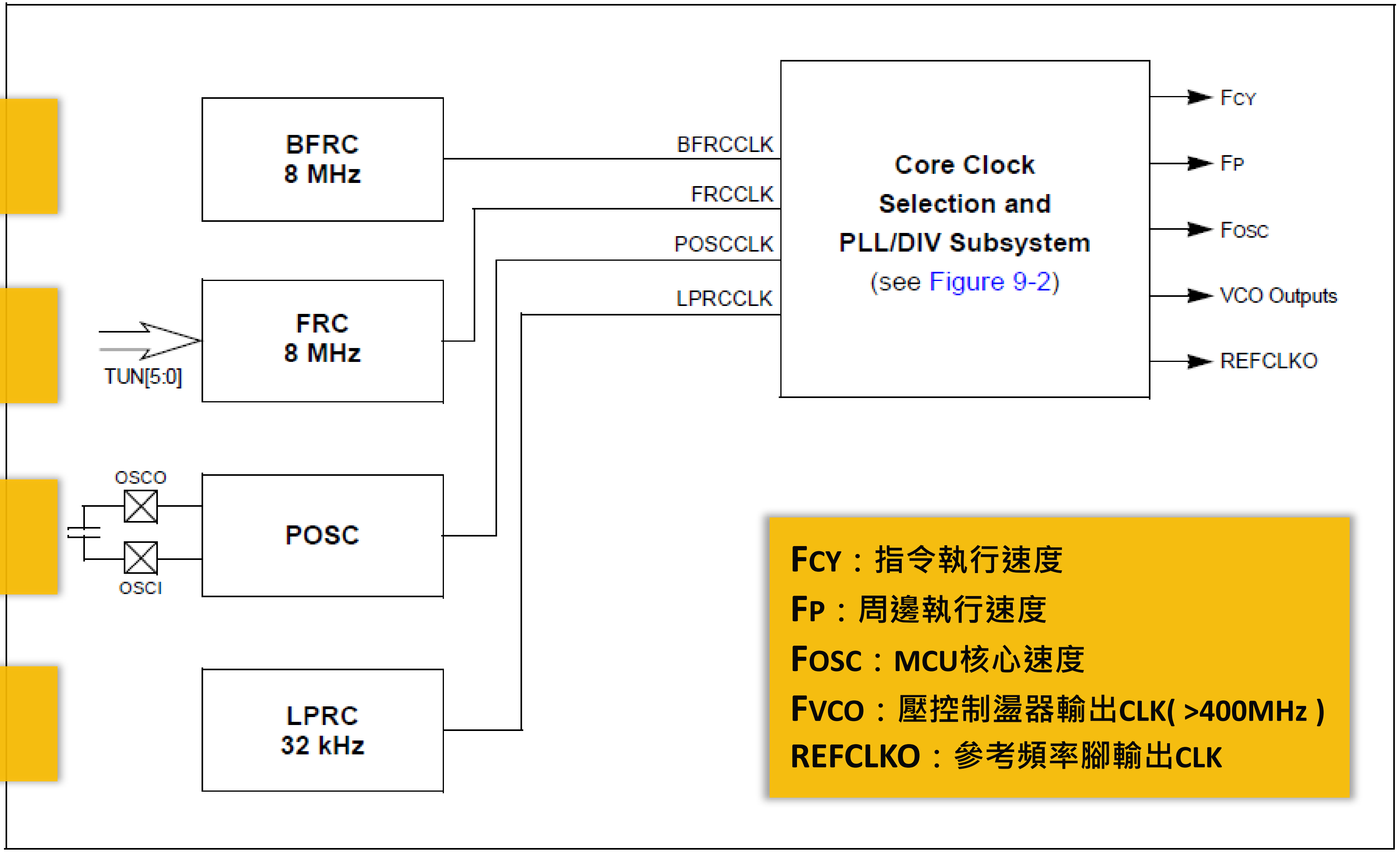
FIGURE 9-2: dsPIC33CK64MC105 CORE OSCILLATOR SUBSYSTEM



Note 1: See Figure 9-3 for details of the PLL module.
2: XTPLL, HSPLL, ECPLL, FRCPLL (FPLLO).
3: Clock option for PWM.
4: Clock option for ADC.
5: Clock option for DAC.

Clock 來源

FIGURE 9-1: dsPIC33CK64MC105 CORE CLOCK SOURCES BLOCK DIAGRAM



備用振盪器
Backup Internal Fast RC

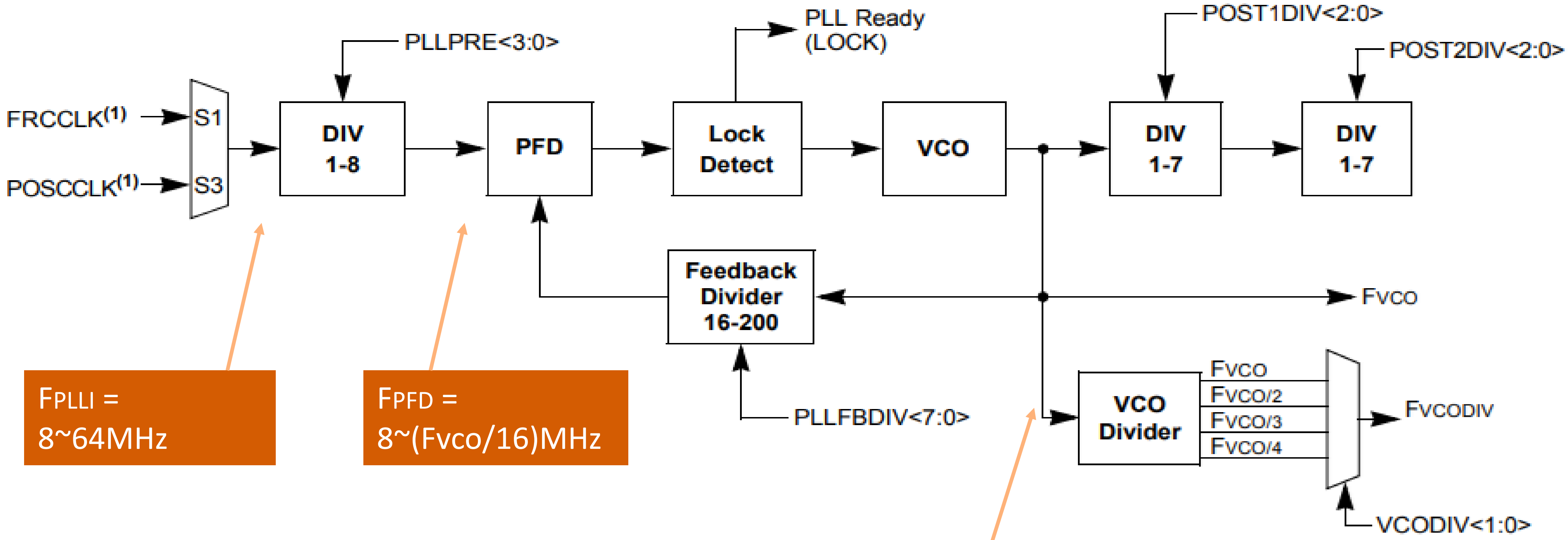
內建高速振盪器
Internal Fast RC

主振盪器
Primary Oscillator

內建低速振盪器
Internal Low-Power RC

FCY：指令執行速度
FP：周邊執行速度
FOSC：MCU核心速度
FVCO：壓控制盪器輸出CLK(>400MHz)
REFCLKO：參考頻率腳輸出CLK

鎖相迴路(PLL)方塊圖



FPLLI =
8~64MHz

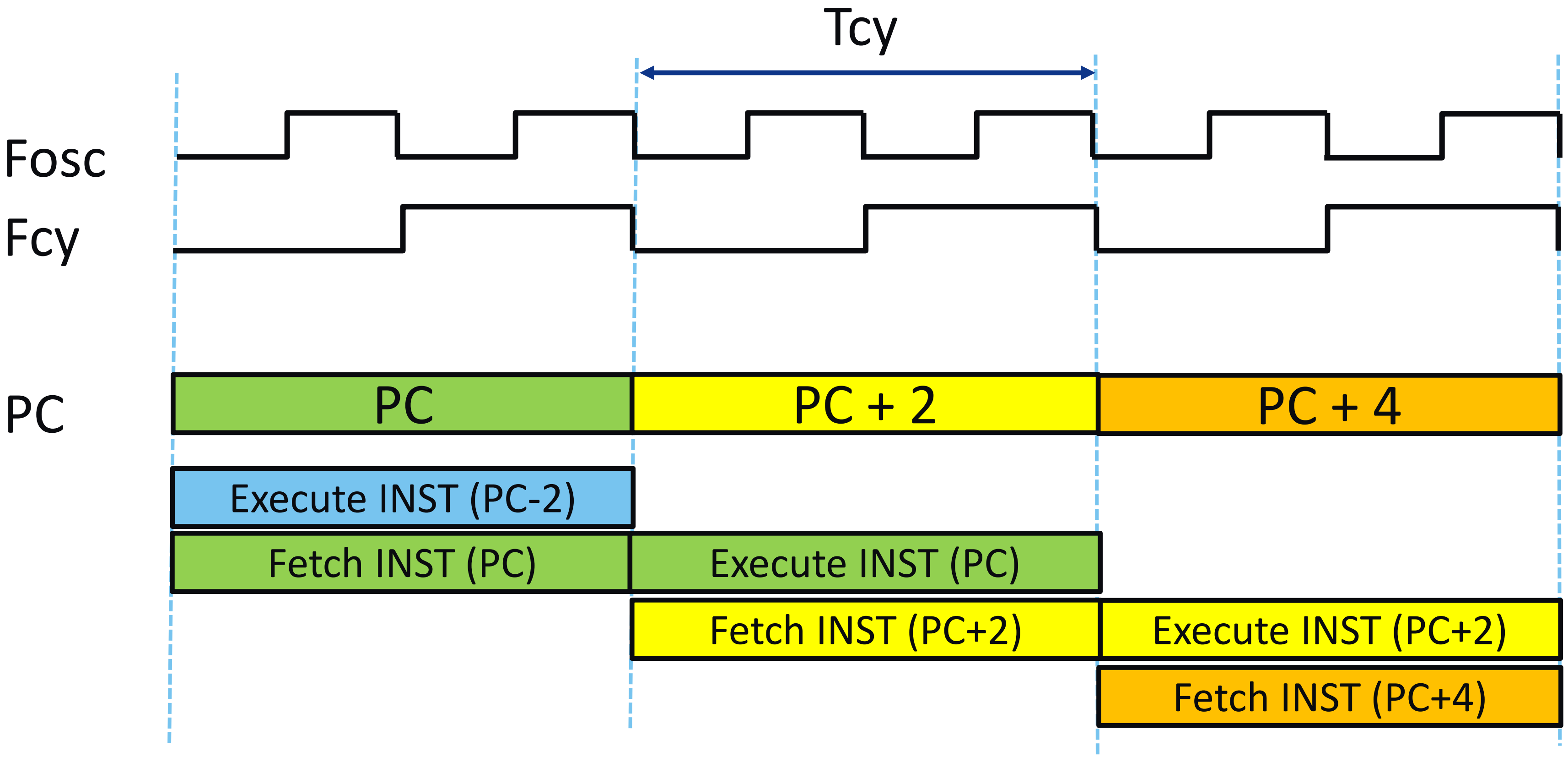
FPFD =
8~(Fvco/16)MHz

Fvco =
400~1600MHz

- VCO - 壓控振盪器 - Voltage Control Oscillator
- PFD - 相位頻率偵測器 - Phase Frequency Detector
- 每一級都有一些限制，就算瞬間超過也可能導致運行異常

dsPIC33C 指令執行流程

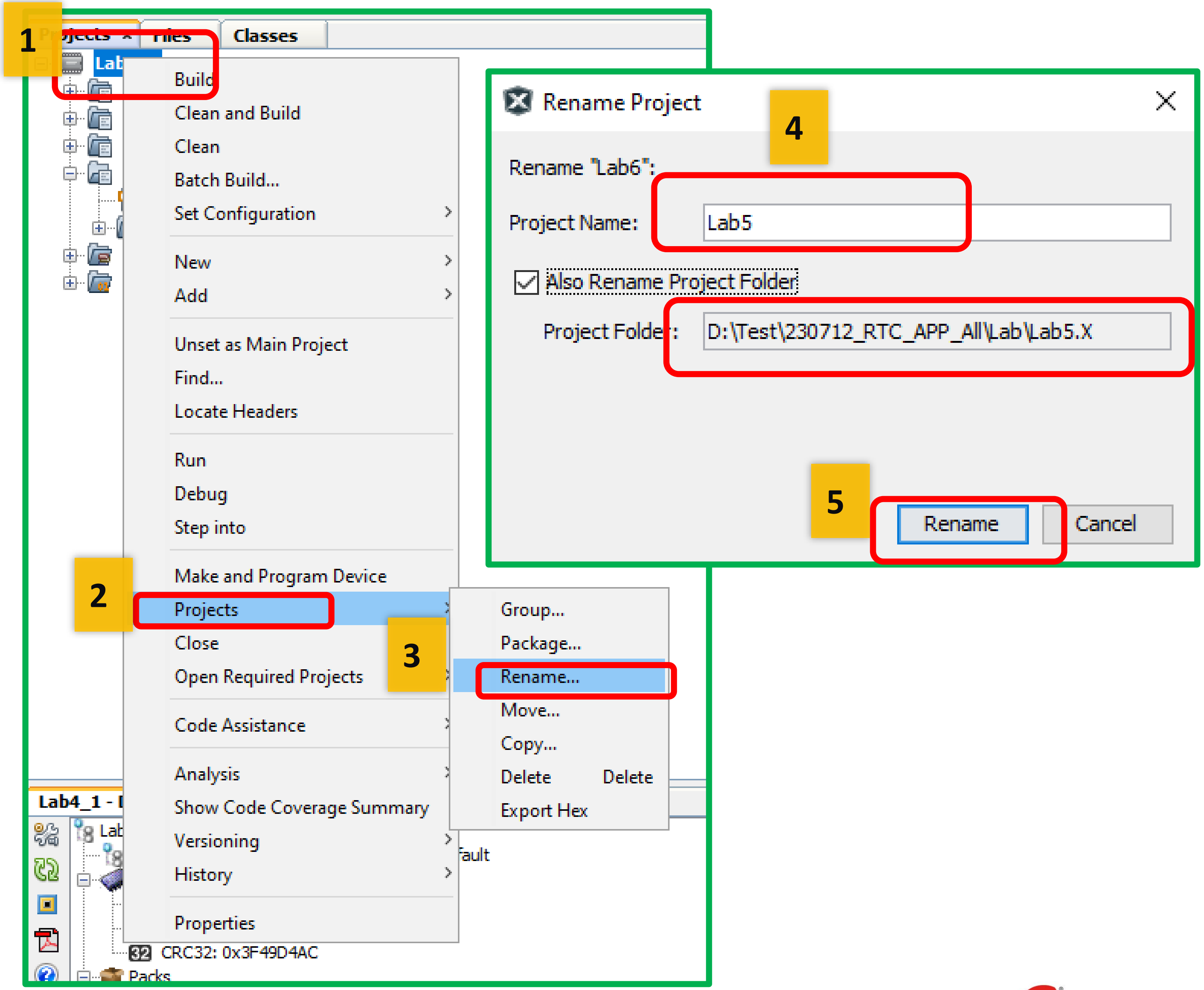
兩個震盪器的時脈 完成一個指令週期



1.建立Lab4

將Lab 4改名成Lab5

1) 到Projects 在Lab4上
   滑鼠點右鍵
2) 找到Projects
3) 找到Rename
3) 修改成Lab5
4) 勾選順便改資料夾名
5) 按 Rename - 完成名稱修改



Rename Project

Rename "Lab6":

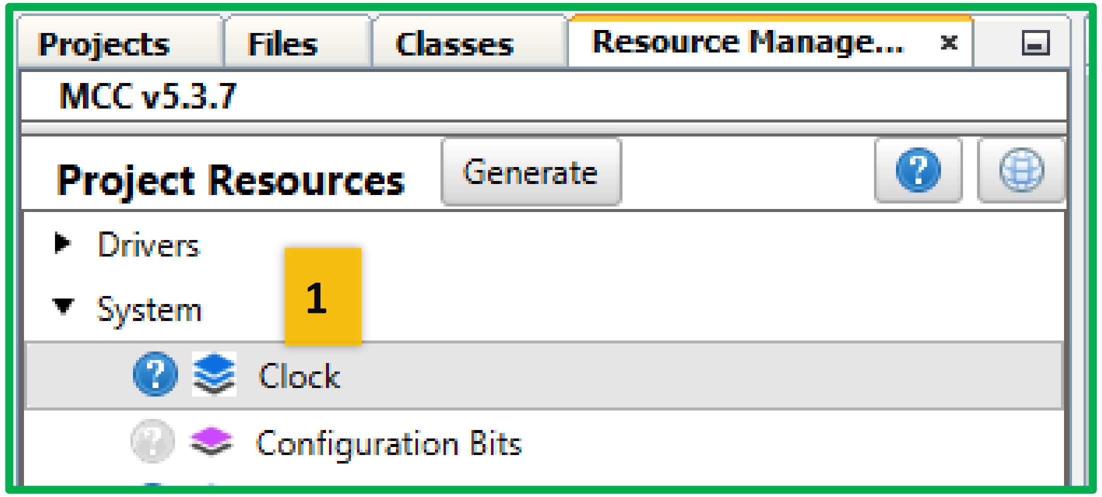
Project Name: Lab5

☑ Also Rename Project Folder

Project Folder: D:\Test\230712_RTC_APP_All\Lab\Lab5.X

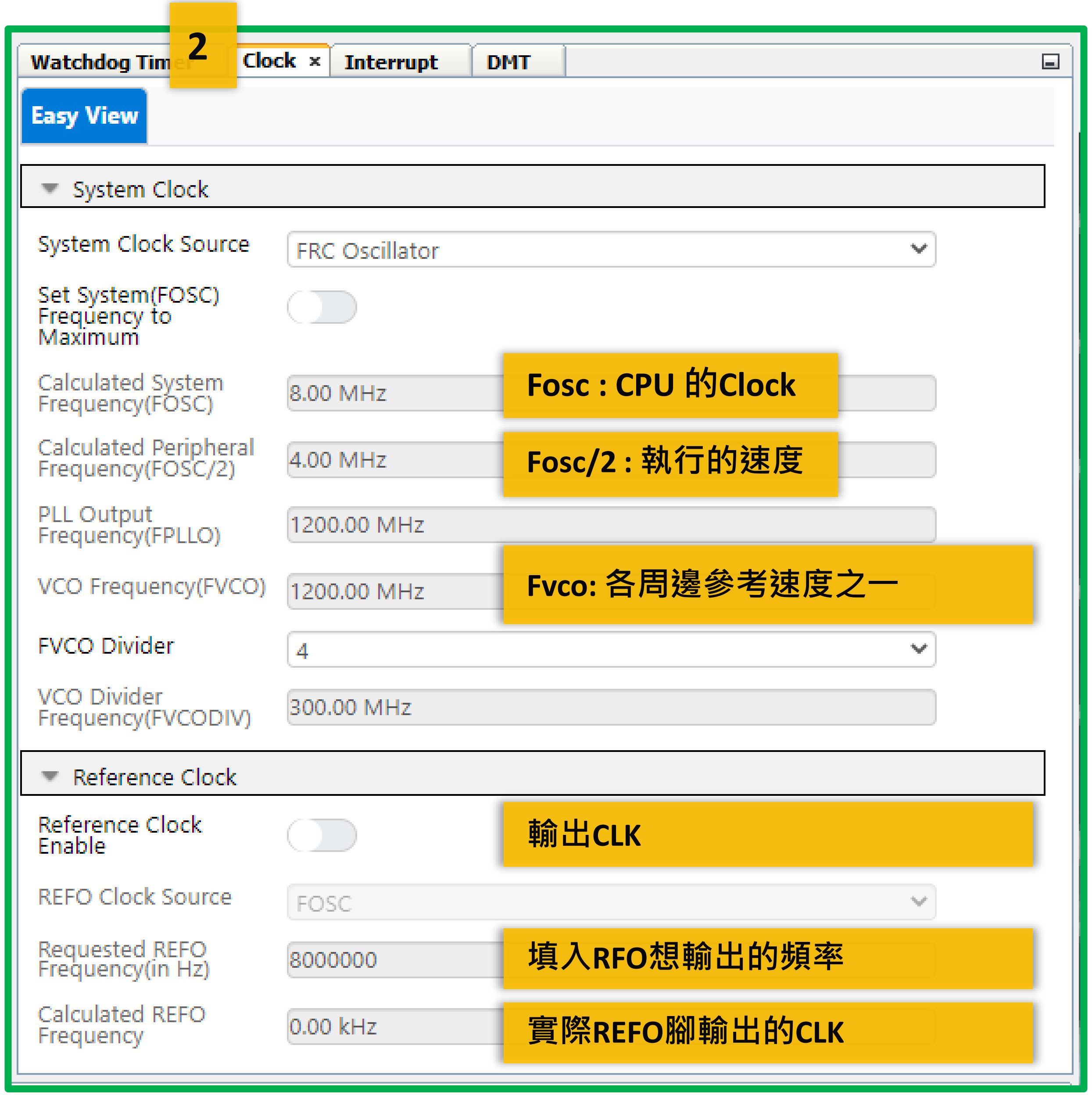
2. 開啟MCC

觀察預設值= FRC , 8MHz



1) 到Project Resources
 找System ,找Clock ,點選

2) 到Clcok的 Easy View



System Clock

System Clock Source	FRC Oscillator
Set System(FOSC) Frequency to Maximum	
Calculated System Frequency(FOSC)	8.00 MHz
Calculated Peripheral Frequency(FOSC/2)	4.00 MHz
PLL Output Frequency(FPLLO)	1200.00 MHz
VCO Frequency(FVCO)	1200.00 MHz
FVCO Divider	4
VCO Divider Frequency(FVCODIV)	300.00 MHz

Reference Clock

Reference Clock Enable	
REFO Clock Source	FOSC
Requested REFO Frequency(in Hz)	8000000
Calculated REFO Frequency	0.00 kHz

Fosc : CPU 的Clock

Fosc/2 : 執行的速度

Fvco: 各周邊參考速度之一

輸出CLK

填入RFO想輸出的頻率

實際REFO腳輸出的CLK

3. 修改Fosc到最快

FRC , 200MHz

1) System Clock Source
   選內部震盪 FRC +升頻 PLL
2) 請MCC根據此Clock來源
   設到Fosc跑最快
3) 設REFO輸出頻率，填500MHz
   根據實際輸出顯示只有100MHz
4) 產生CODE

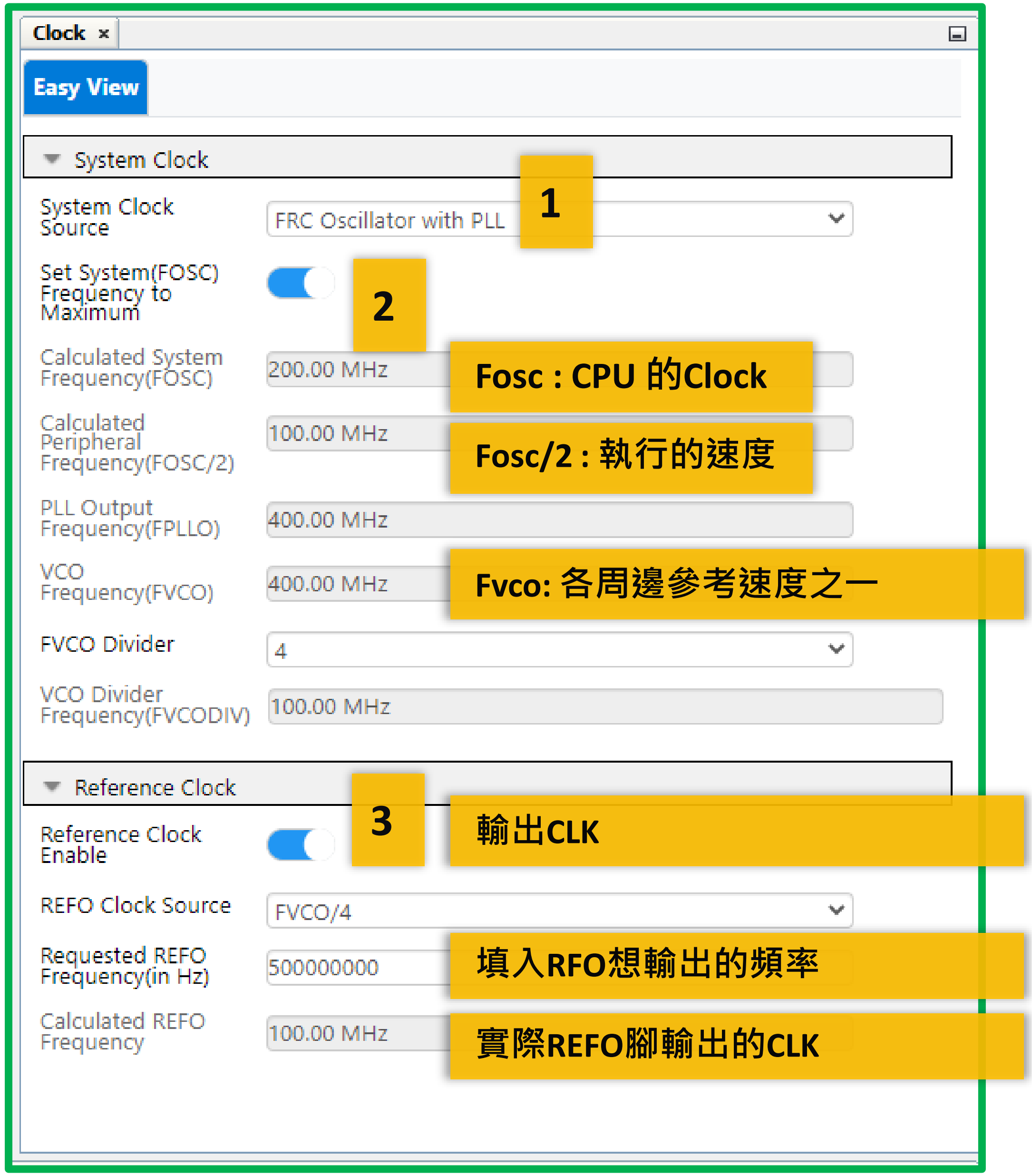
分析

上一步跟新設定後MCC內容

Fosc 8 MHz -> 200 MHz

Fvco 1200 MHz -> 400 MHz

REFO 0 MHz -> 100 MHz



Clock ×

Easy View

System Clock

System Clock Source: FRC Oscillator with PLL

Set System(FOSC) Frequency to Maximum: on

Calculated System Frequency(FOSC): 200.00 MHz

Calculated Peripheral Frequency(FOSC/2): 100.00 MHz

PLL Output Frequency(FPLLO): 400.00 MHz

VCO Frequency(FVCO): 400.00 MHz

FVCO Divider: 4

VCO Divider Frequency(FVCODIV): 100.00 MHz

Reference Clock

Reference Clock Enable: on

REFO Clock Source: FVCO/4

Requested REFO Frequency(in Hz): 500000000

Calculated REFO Frequency: 100.00 MHz

4.結果

Lab5

1) 燒錄測試

2) 結果

小板上

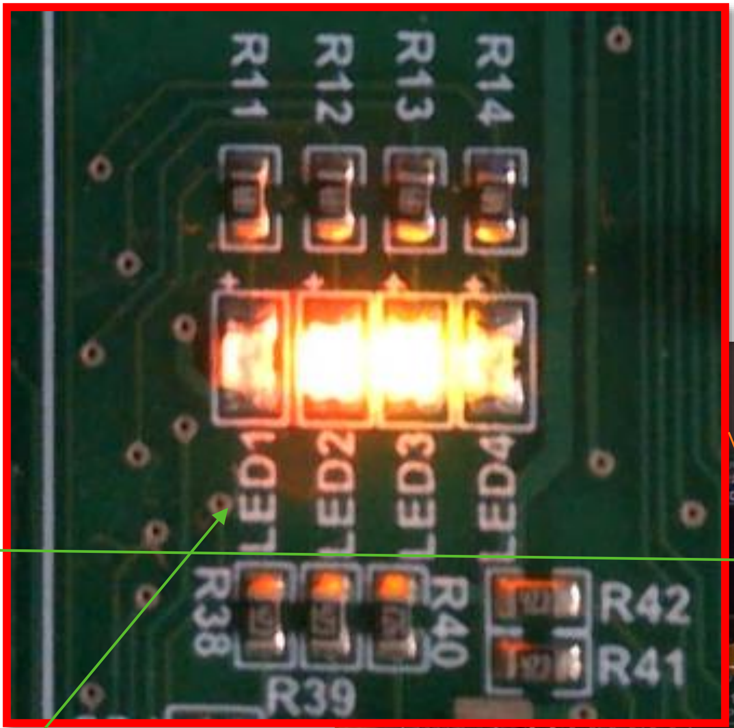
LED0看起來恆亮

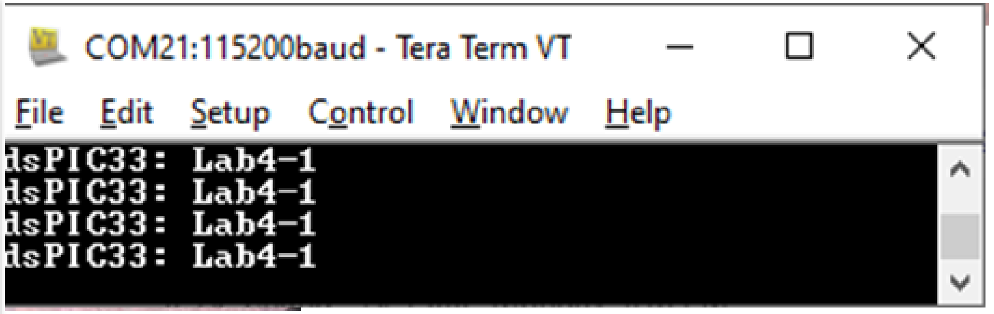
大板上

LED1~2交互快閃

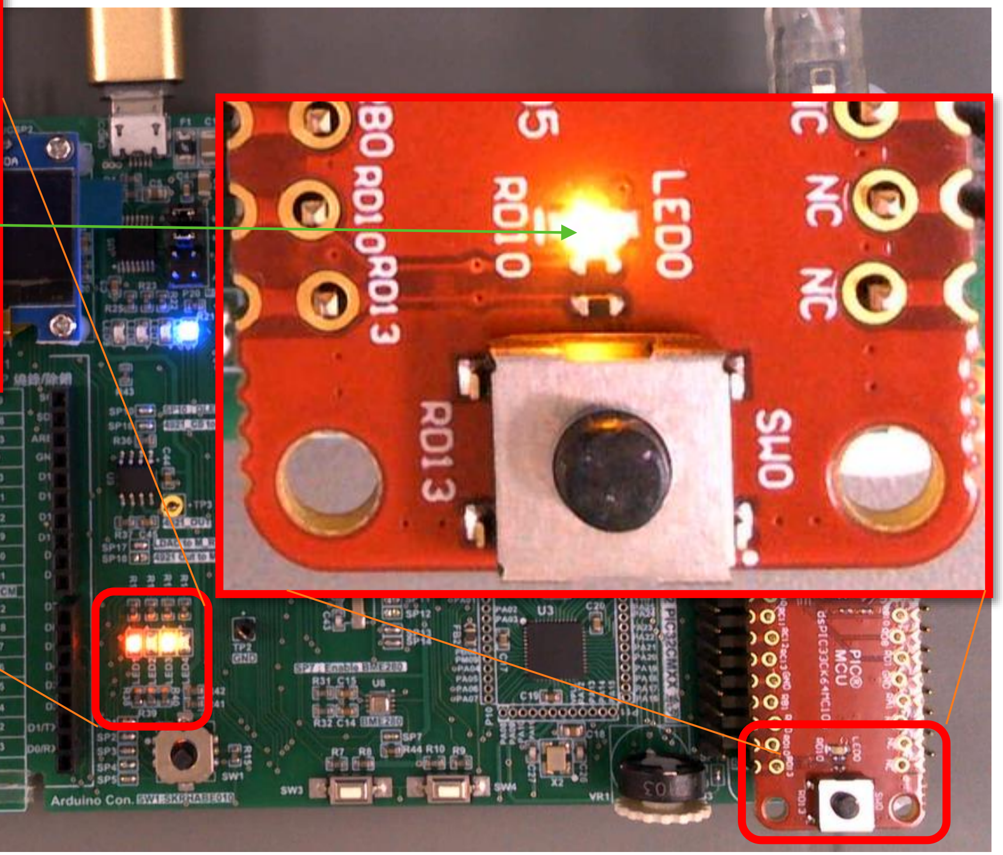
LED3~4交互看起來恆亮

PC顯示 dsPIC33: Lab4-1





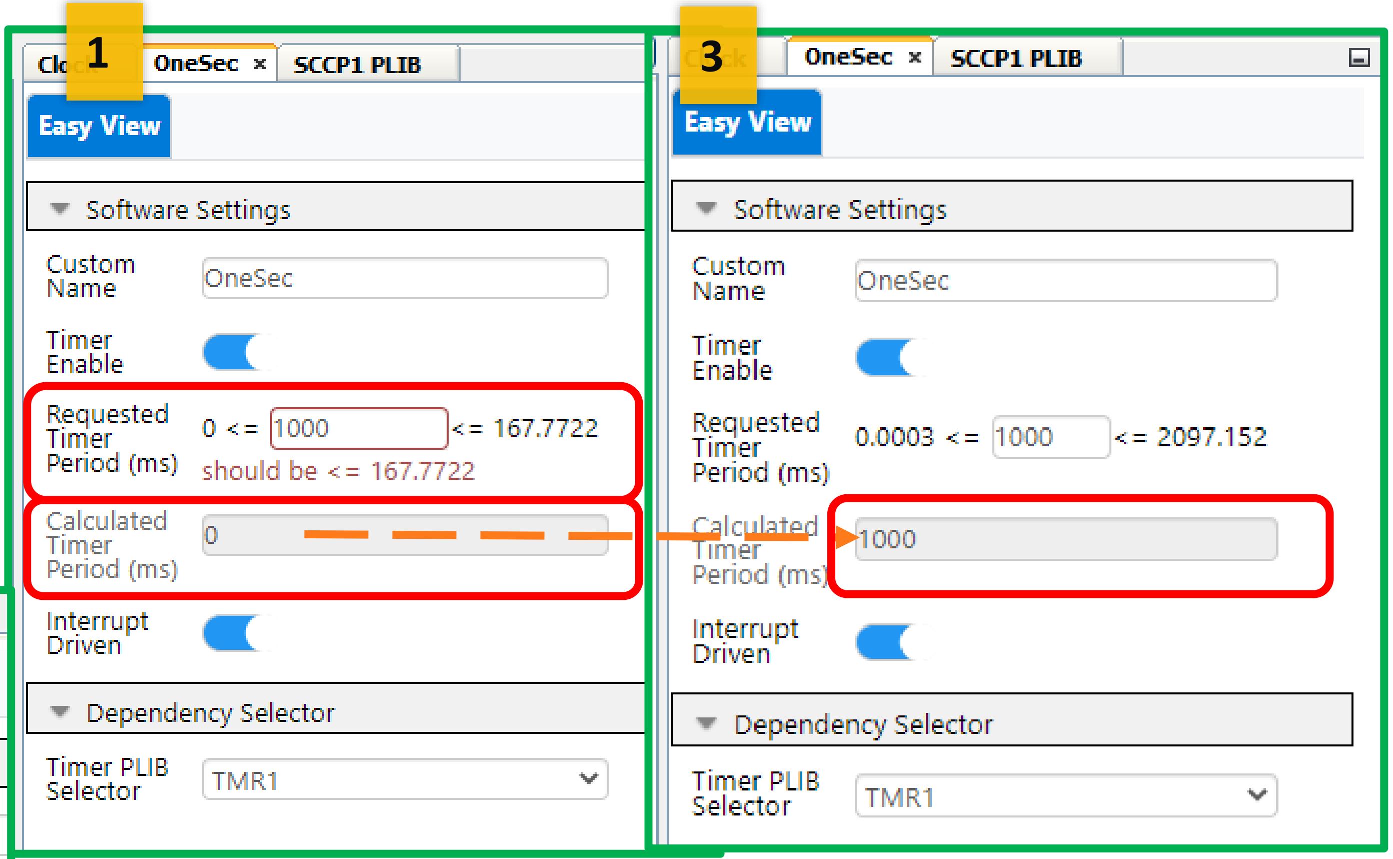
```
dsPIC33: Lab4-1
dsPIC33: Lab4-1
dsPIC33: Lab4-1
dsPIC33: Lab4-1
```



5. 修正LED0

因MCU時脈變快造成

1) 查看OneSec設定
   發現出現警告並告知實際 0ms
2) 開啟TMR1 PLIB
   改變時脈來源
   FOSC/2 = 100 MHMz→ FRC 8 MHMz
3) 查看OneSec變化



1

Clock | OneSec × | SCCP1 PLIB

Easy View

Software Settings

Custom Name: OneSec

Timer Enable: ON

Requested Timer Period (ms): 0 <= 1000 <= 167.7722
should be <= 167.7722

Calculated Timer Period (ms): 0

Interrupt Driven: ON

Dependency Selector

Timer PLIB Selector: TMR1

3

Clock | OneSec × | SCCP1 PLIB

Easy View

Software Settings

Custom Name: OneSec

Timer Enable: ON

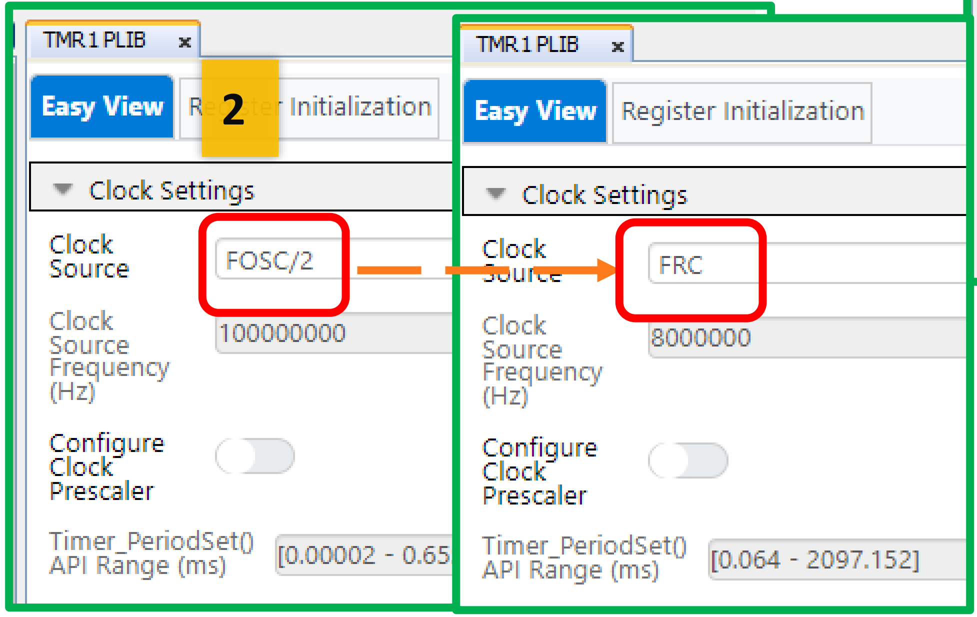
Requested Timer Period (ms): 0.0003 <= 1000 <= 2097.152

Calculated Timer Period (ms): 1000

Interrupt Driven: ON

Dependency Selector

Timer PLIB Selector: TMR1



2

TMR1 PLIB ×

Easy View | Register Initialization

Clock Settings

Clock Source: FOSC/2

Clock Source Frequency (Hz): 100000000

Configure Clock Prescaler: OFF

Timer_PeriodSet() API Range (ms): [0.00002 - 0.65

TMR1 PLIB ×

Easy View | Register Initialization

Clock Settings

Clock Source: FRC

Clock Source Frequency (Hz): 8000000

Configure Clock Prescaler: OFF

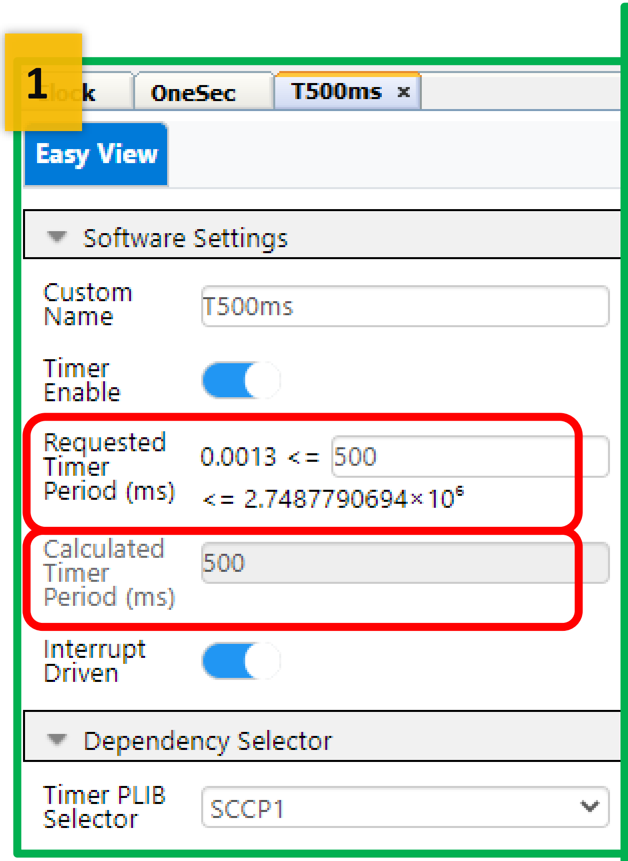
Timer_PeriodSet() API Range (ms): [0.064 - 2097.152]

6. 修正LED1~4

T500ms

1) 查看T500ms設定
   發現沒有警告

2) 開啟sccp1.c程式
   CCP1PRH = 0xB; 不是零
   所以不能用16-bit Timer模式



Software Settings

Custom Name: T500ms

Timer Enable: ●

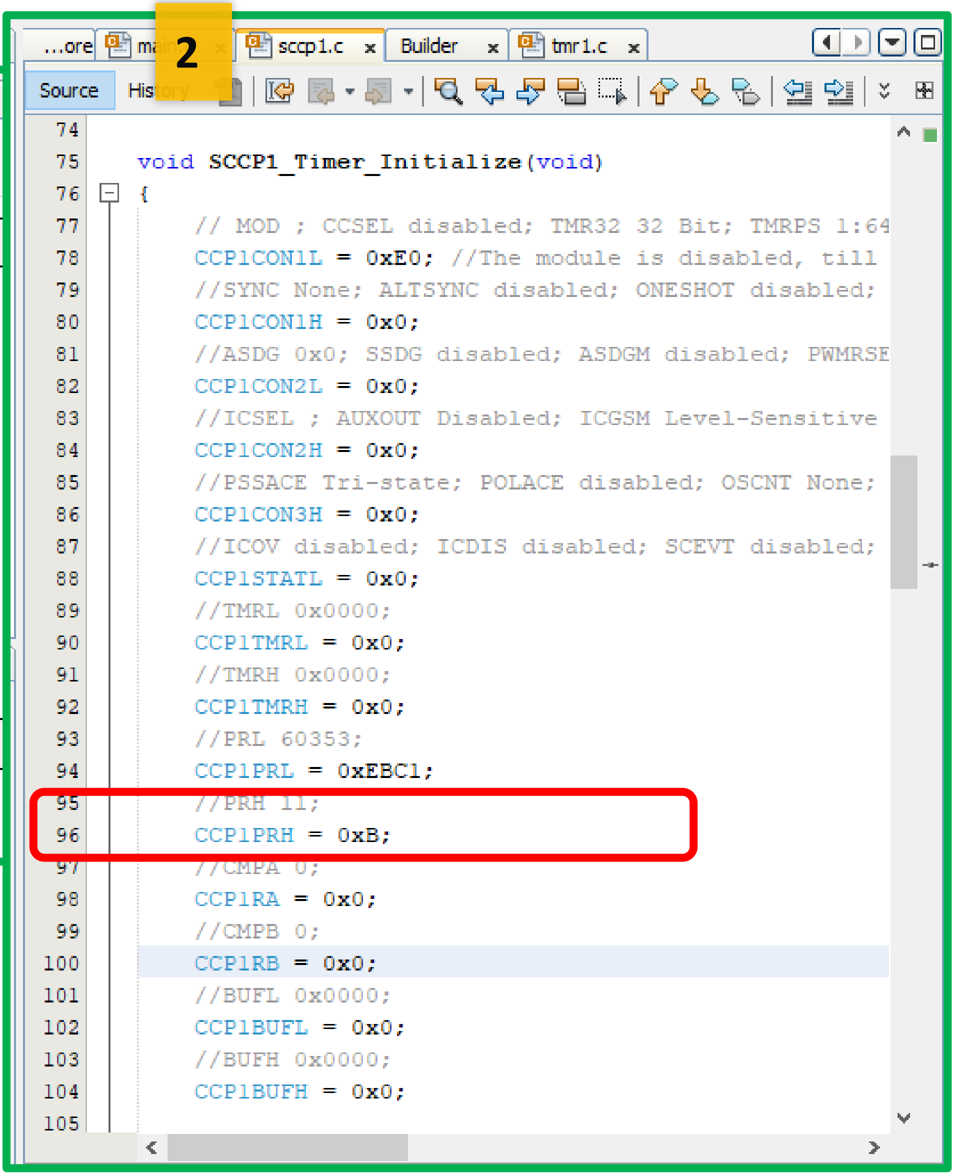
Requested Timer Period (ms): 0.0013 <= 500 <= 2.7487790694×10⁶

Calculated Timer Period (ms): 500

Interrupt Driven: ●

Dependency Selector

Timer PLIB Selector: SCCP1



```c

void SCCP1_Timer_Initialize(void)
{
    // MOD ; CCSEL disabled; TMR32 32 Bit; TMRPS 1:64
    CCP1CON1L = 0xE0; //The module is disabled, till
    //SYNC None; ALTSYNC disabled; ONESHOT disabled;
    CCP1CON1H = 0x0;
    //ASDG 0x0; SSDG disabled; ASDGM disabled; PWMRSE
    CCP1CON2L = 0x0;
    //ICSEL ; AUXOUT Disabled; ICGSM Level-Sensitive
    CCP1CON2H = 0x0;
    //PSSACE Tri-state; POLACE disabled; OSCNT None;
    CCP1CON3H = 0x0;
    //ICOV disabled; ICDIS disabled; SCEVT disabled;
    CCP1STATL = 0x0;
    //TMRL 0x0000;
    CCP1TMRL = 0x0;
    //TMRH 0x0000;
    CCP1TMRH = 0x0;
    //PRL 60353;
    CCP1PRL = 0xEBC1;
    //PRH 11;
    CCP1PRH = 0xB;
    //CMPA 0;
    CCP1RA = 0x0;
    //CMPB 0;
    CCP1RB = 0x0;
    //BUFL 0x0000;
    CCP1BUFL = 0x0;
    //BUFH 0x0000;
    CCP1BUFH = 0x0;

```

6-1. 修正LED1~4

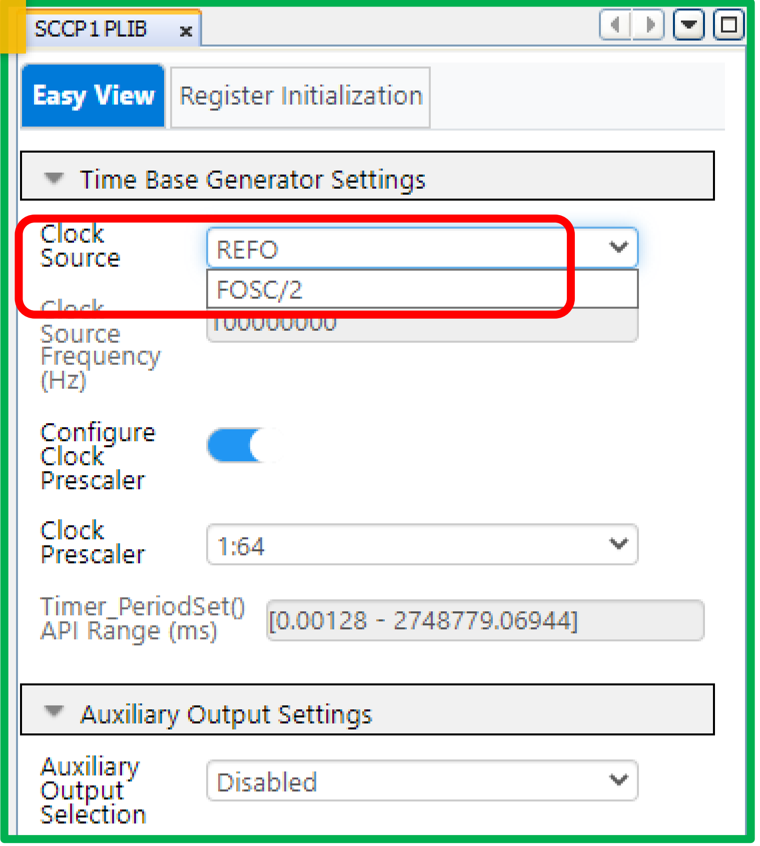
T500ms

1) 開啟SCCP1 PLIB 修改頻率來源成REFO
2) 到Clock改REFO為4MHz
3) 再產生一次CODE
4) 觀看sccp1.c

   CCP1PRL = 0x7A11;

   CCP1PRH = 0x0;

1



Time Base Generator Settings

Clock Source: REFO (FOSC/2)

Clock Source Frequency (Hz): 100000000

Configure Clock Prescaler: on

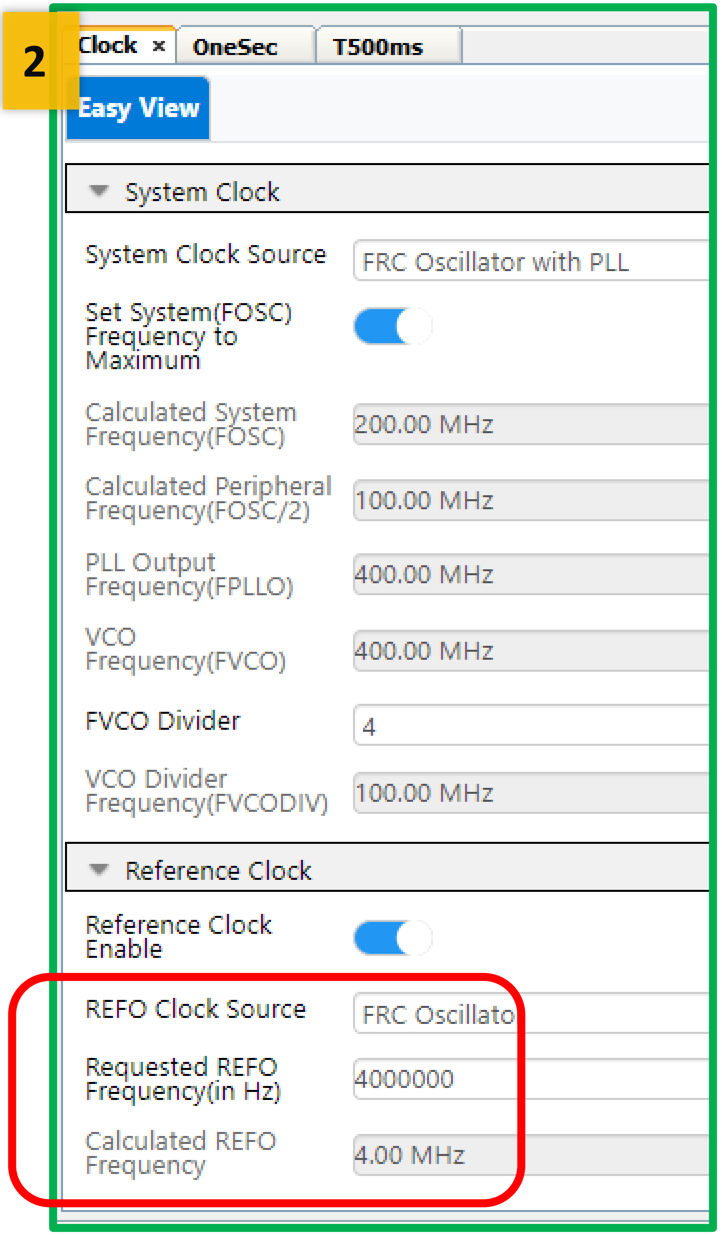
Clock Prescaler: 1:64

Timer_PeriodSet() API Range (ms): [0.00128 - 2748779.06944]

Auxiliary Output Settings

Auxiliary Output Selection: Disabled

2



System Clock

System Clock Source: FRC Oscillator with PLL

Set System(FOSC) Frequency to Maximum: on

Calculated System Frequency(FOSC): 200.00 MHz

Calculated Peripheral Frequency(FOSC/2): 100.00 MHz

PLL Output Frequency(FPLLO): 400.00 MHz

VCO Frequency(FVCO): 400.00 MHz

FVCO Divider: 4

VCO Divider Frequency(FVCODIV): 100.00 MHz

Reference Clock

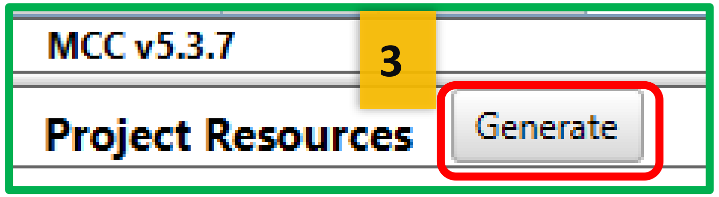
Reference Clock Enable: on

REFO Clock Source: FRC Oscillato

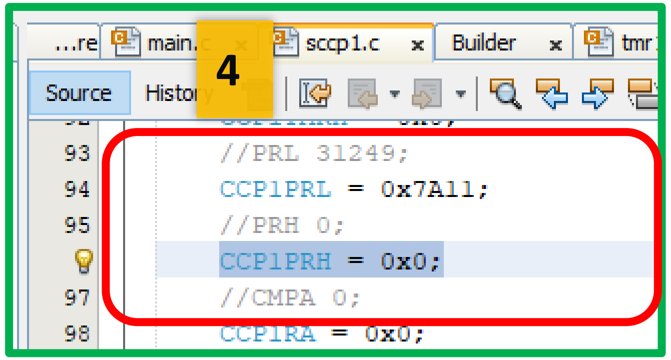
Requested REFO Frequency(in Hz): 4000000

Calculated REFO Frequency: 4.00 MHz

3



4



```
//PRL 31249;
CCP1PRL = 0x7A11;
//PRH 0;
CCP1PRH = 0x0;
//CMPA 0;
CCP1RA = 0x0;
```

7.結果

回復原來樣貌

1) 燒錄測試

2) 結果

小板上

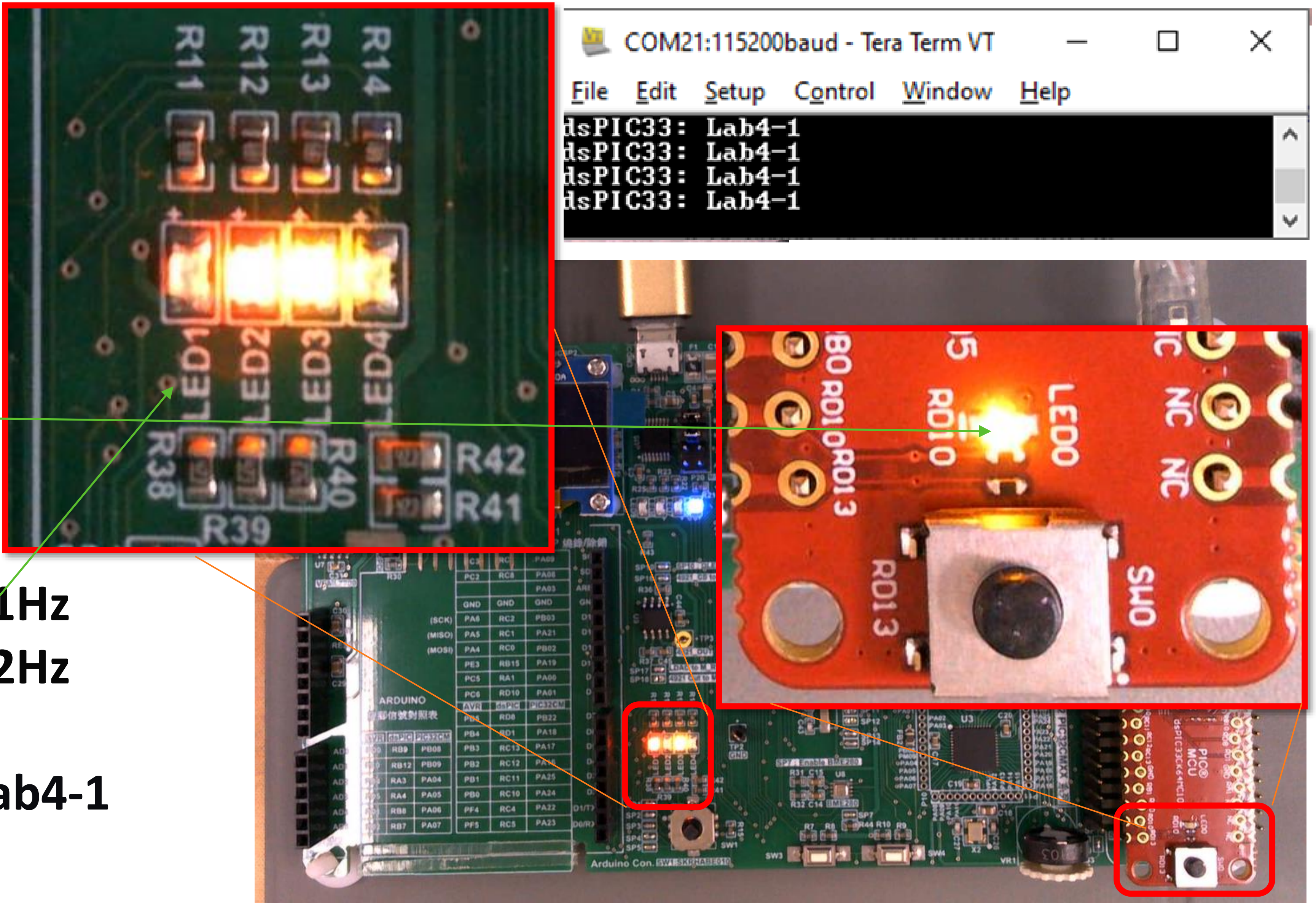
LED0閃爍 1Hz

大板上

LED1~2交互慢閃 1Hz

LED3~4交互快閃 2Hz

PC顯示 dsPIC33: Lab4-1



```
dsPIC33: Lab4-1
dsPIC33: Lab4-1
dsPIC33: Lab4-1
dsPIC33: Lab4-1
```

實驗六

Config(Special Features)

內建硬體程式開發

實驗六：Config(Special Features)

- 本實驗目的：
  - 了解Special Features有哪些設定
  - 會影響那些動作
  - 詳細內容哪裡找
  - 測試看門狗WDT

- 結果呈現：
  - LED0 亮1 Hz
  - LED1,2 亮1 Hz
  - LED3,4 亮2 Hz

背景知識

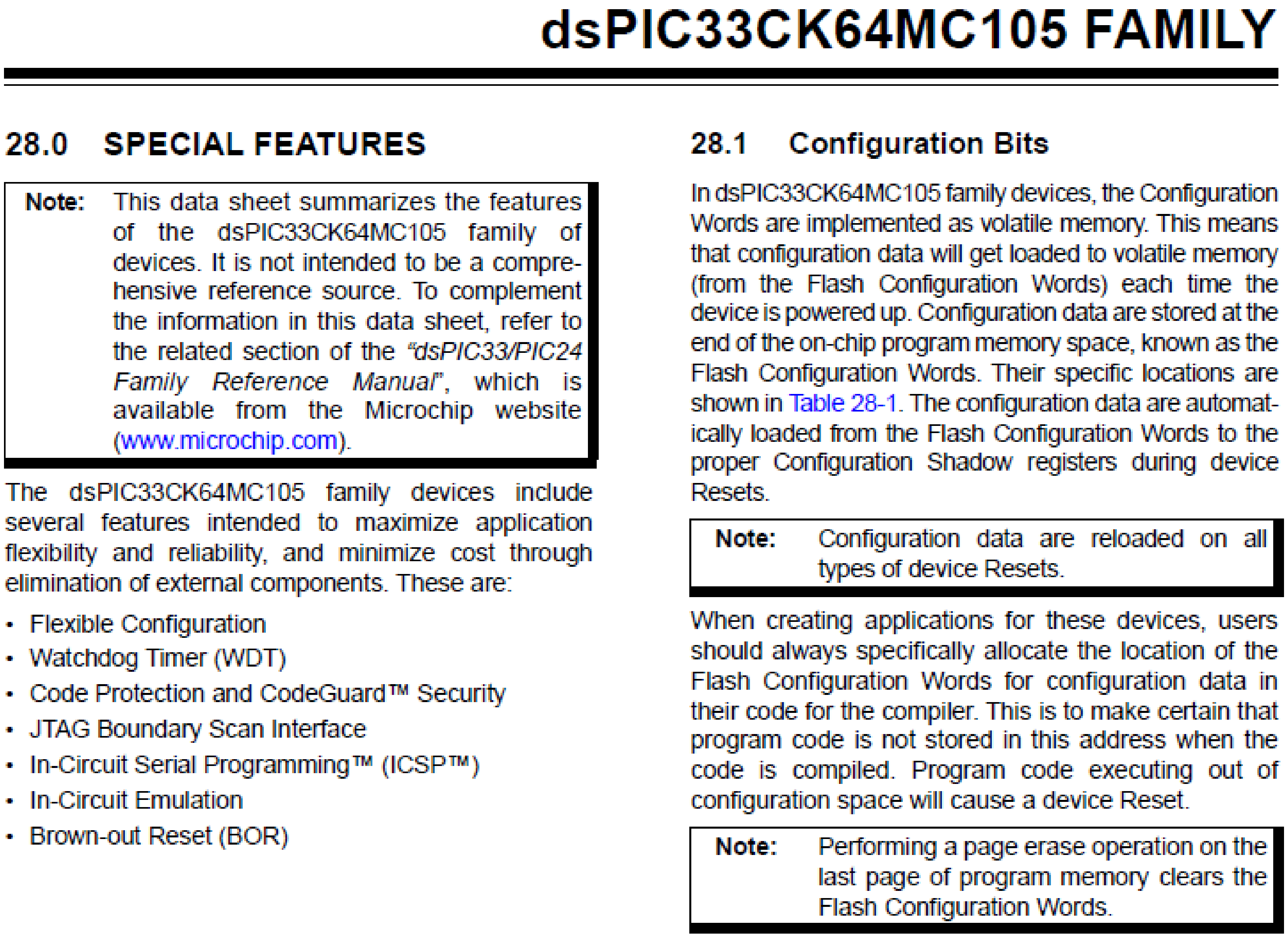
dsPIC33CK datasheet

Config
在設什麼

- **Config：IC的特殊功能(Special Features)**
- **軟體執行前，對IC的設定**

  例如：

  時脈來源選定、看門狗(WDT)是否開啟、Flash內資料是否可透過燒錄器被讀取(Code Protection Enable)等



dsPIC33CK64MC105 FAMILY

28.0 SPECIAL FEATURES

> **Note:** This data sheet summarizes the features of the dsPIC33CK64MC105 family of devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to the related section of the *"dsPIC33/PIC24 Family Reference Manual"*, which is available from the Microchip website (www.microchip.com).

The dsPIC33CK64MC105 family devices include several features intended to maximize application flexibility and reliability, and minimize cost through elimination of external components. These are:

- Flexible Configuration
- Watchdog Timer (WDT)
- Code Protection and CodeGuard™ Security
- JTAG Boundary Scan Interface
- In-Circuit Serial Programming™ (ICSP™)
- In-Circuit Emulation
- Brown-out Reset (BOR)

28.1 Configuration Bits

In dsPIC33CK64MC105 family devices, the Configuration Words are implemented as volatile memory. This means that configuration data will get loaded to volatile memory (from the Flash Configuration Words) each time the device is powered up. Configuration data are stored at the end of the on-chip program memory space, known as the Flash Configuration Words. Their specific locations are shown in Table 28-1. The configuration data are automatically loaded from the Flash Configuration Words to the proper Configuration Shadow registers during device Resets.

> **Note:** Configuration data are reloaded on all types of device Resets.

When creating applications for these devices, users should always specifically allocate the location of the Flash Configuration Words for configuration data in their code for the compiler. This is to make certain that program code is not stored in this address when the code is compiled. Program code executing out of configuration space will cause a device Reset.

> **Note:** Performing a page erase operation on the last page of program memory clears the Flash Configuration Words.

資料存放相關位置

記憶體安排

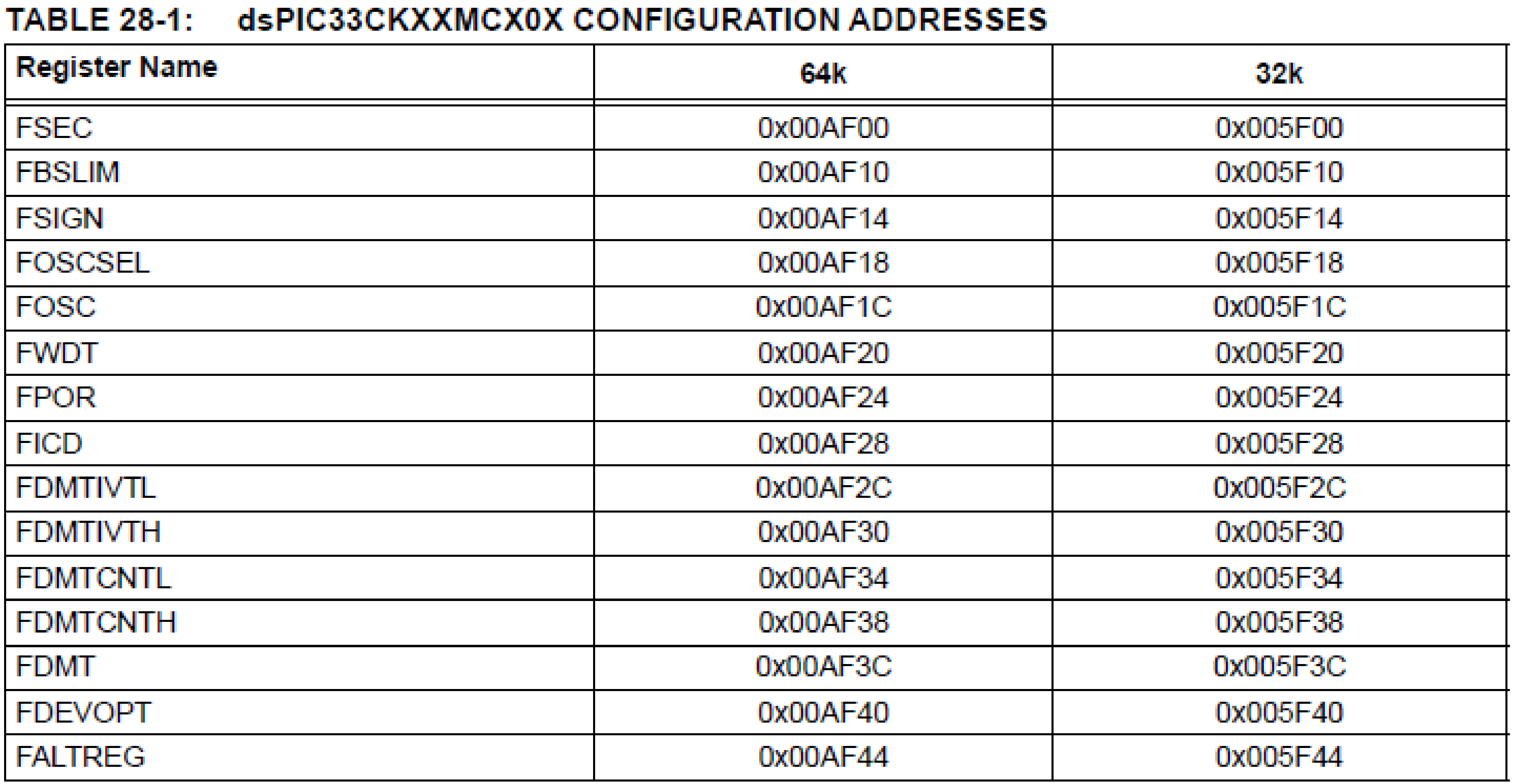


TABLE 28-1: dsPIC33CKXXMCX0X CONFIGURATION ADDRESSES

Register Name	64k	32k
FSEC	0x00AF00	0x005F00
FBSLIM	0x00AF10	0x005F10
FSIGN	0x00AF14	0x005F14
FOSCSEL	0x00AF18	0x005F18
FOSC	0x00AF1C	0x005F1C
FWDT	0x00AF20	0x005F20
FPOR	0x00AF24	0x005F24
FICD	0x00AF28	0x005F28
FDMTIVTL	0x00AF2C	0x005F2C
FDMTIVTH	0x00AF30	0x005F30
FDMTCNTL	0x00AF34	0x005F34
FDMTCNTH	0x00AF38	0x005F38
FDMT	0x00AF3C	0x005F3C
FDEVOPT	0x00AF40	0x005F40
FALTREG	0x00AF44	0x005F44

詳細內容

Datasheet中有各個bit說明

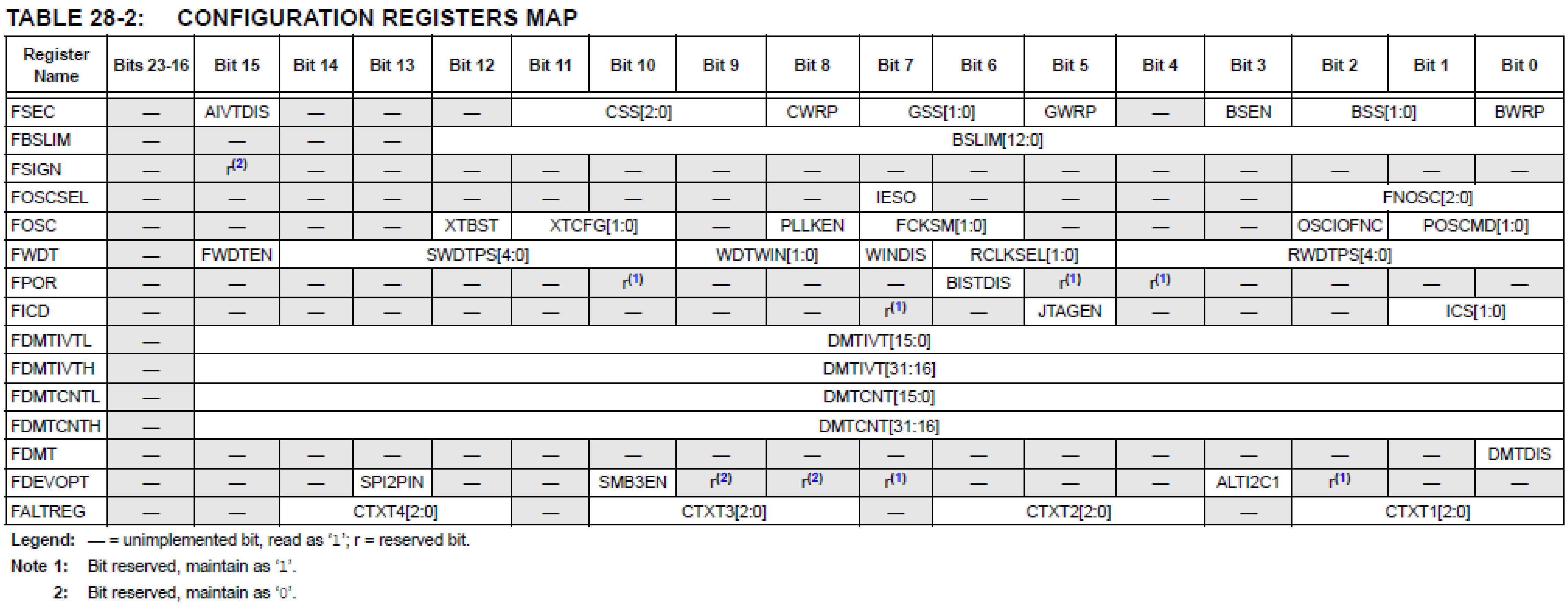


TABLE 28-2: CONFIGURATION REGISTERS MAP

Register Name	Bits 23-16	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
FSEC	—	AIVTDIS	—	—	—	CSS[2:0]			CWRP	GSS[1:0]		GWRP	—	BSEN	BSS[1:0]		BWRP
FBSLIM	—	—	—	—	BSLIM[12:0]												
FSIGN	—	r[2]	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
FOSCSEL	—	—	—	—	—	—	—	—	—	IESO	—	—	—	—	FNOSC[2:0]		
FOSC	—	—	—	—	XTBST	XTCFG[1:0]		—	PLLKEN	FCKSM[1:0]		—	—	—	OSCIOFNC	POSCMD[1:0]	
FWDT	—	FWDTEN	SWDTPS[4:0]					WDTWIN[1:0]		WINDIS	RCLKSEL[1:0]		RWDTPS[4:0]				
FPOR	—	—	—	—	—	—	r[1]	—	—	—	BISTDIS	r[1]	r[1]	—	—	—	—
FICD	—	—	—	—	—	—	—	—	—	r[1]	—	JTAGEN	—	—	—	ICS[1:0]	
FDMTIVTL	—	DMTIVT[15:0]															
FDMTIVTH	—	DMTIVT[31:16]															
FDMTCNTL	—	DMTCNT[15:0]															
FDMTCNTH	—	DMTCNT[31:16]															
FDMT	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DMTDIS
FDEVOPT	—	—	—	SPI2PIN	—	—	SMB3EN	r[2]	r[2]	r[1]	—	—	—	ALTI2C1	r[1]	—	—
FALTREG	—	—	CTXT4[2:0]			—	CTXT3[2:0]			—	CTXT2[2:0]			—	CTXT1[2:0]		

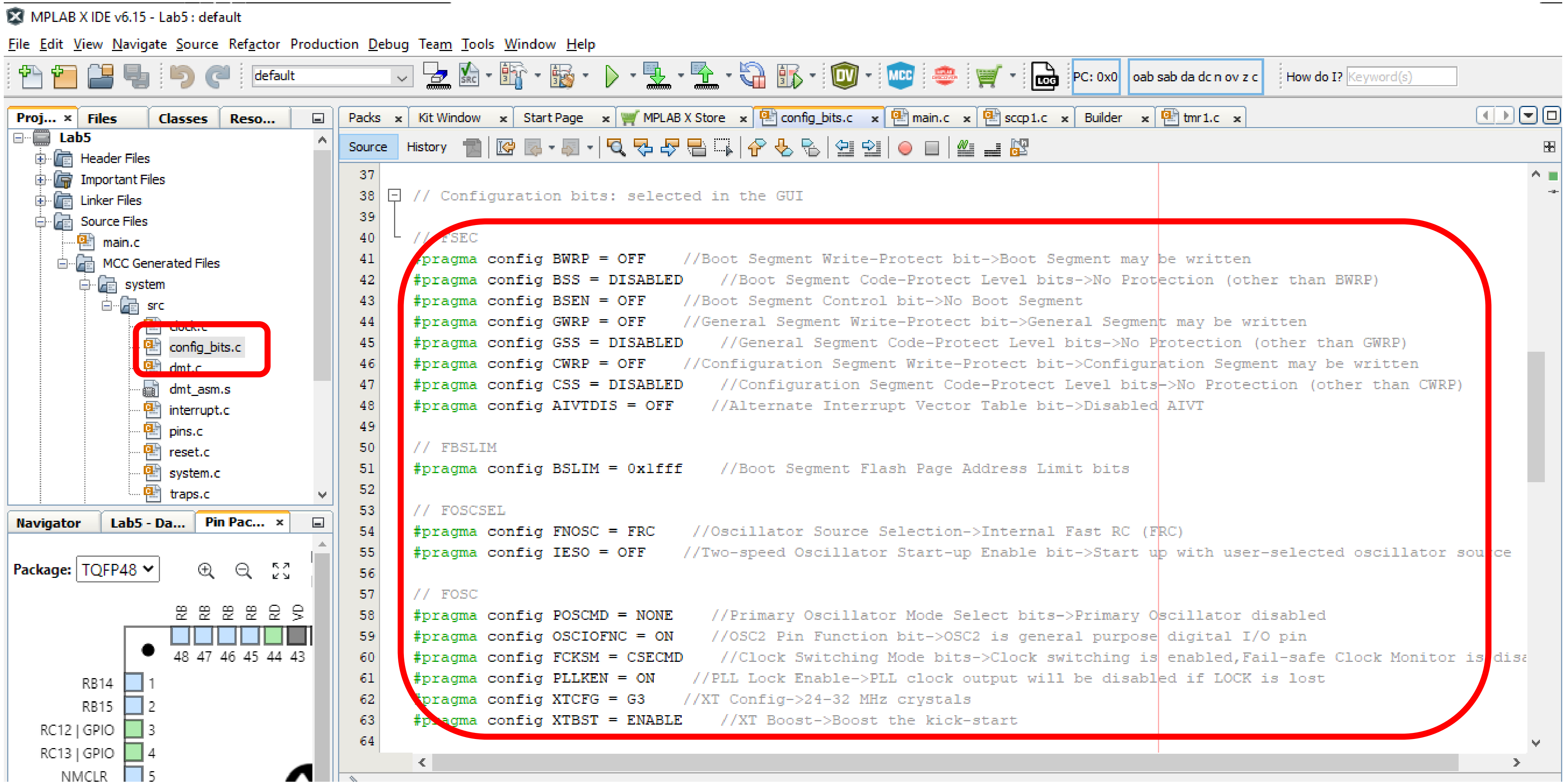
Legend: — = unimplemented bit, read as '1'; r = reserved bit.

Note 1: Bit reserved, maintain as '1'.

2: Bit reserved, maintain as '0'.

程式存放位置

xxx\Lab\Lab5.X\mcc_generated_files\system\src\config_bits.c



```
// Configuration bits: selected in the GUI

// FSEC
#pragma config BWRP = OFF    //Boot Segment Write-Protect bit->Boot Segment may be written
#pragma config BSS = DISABLED    //Boot Segment Code-Protect Level bits->No Protection (other than BWRP)
#pragma config BSEN = OFF    //Boot Segment Control bit->No Boot Segment
#pragma config GWRP = OFF    //General Segment Write-Protect bit->General Segment may be written
#pragma config GSS = DISABLED    //General Segment Code-Protect Level bits->No Protection (other than GWRP)
#pragma config CWRP = OFF    //Configuration Segment Write-Protect bit->Configuration Segment may be written
#pragma config CSS = DISABLED    //Configuration Segment Code-Protect Level bits->No Protection (other than CWRP)
#pragma config AIVTDIS = OFF    //Alternate Interrupt Vector Table bit->Disabled AIVT

// FBSLIM
#pragma config BSLIM = 0x1fff    //Boot Segment Flash Page Address Limit bits

// FOSCSEL
#pragma config FNOSC = FRC    //Oscillator Source Selection->Internal Fast RC (FRC)
#pragma config IESO = OFF    //Two-speed Oscillator Start-up Enable bit->Start up with user-selected oscillator source

// FOSC
#pragma config POSCMD = NONE    //Primary Oscillator Mode Select bits->Primary Oscillator disabled
#pragma config OSCIOFNC = ON    //OSC2 Pin Function bit->OSC2 is general purpose digital I/O pin
#pragma config FCKSM = CSECMD    //Clock Switching Mode bits->Clock switching is enabled,Fail-safe Clock Monitor is disa
#pragma config PLLKEN = ON    //PLL Lock Enable->PLL clock output will be disabled if LOCK is lost
#pragma config XTCFG = G3    //XT Config->24-32 MHz crystals
#pragma config XTBST = ENABLE    //XT Boost->Boost the kick-start
```

Config設定

常見問題

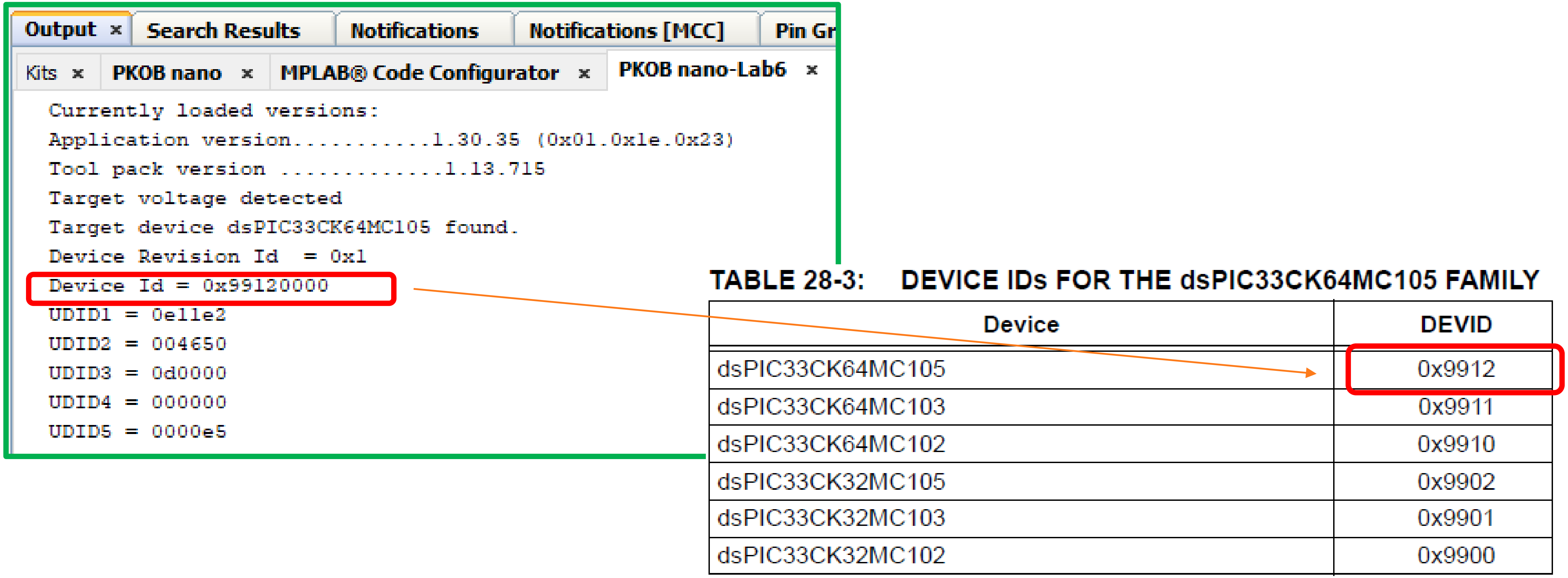
- **時脈來源(FNOSC)**
  - 須搭陪外部電路，例如選外部Oscillator卻沒接，將無法運行

- **Crystal驅動(XTCFG)**
  - 不同石英需搭配不同驅動器、耗電也不同，選錯振不起來
  - 例如4~8 MHz Gain值選0

- **啟動看門狗(FWDTEN、WDTCON)**
  - **須搭陪軟體，啟動後軟體沒定時復歸，MCU會一值RESET**

IC型號

辨別同一家族IC



```
Currently loaded versions:
Application version...........1.30.35 (0x01.0x1e.0x23)
Tool pack version ..............1.13.715
Target voltage detected
Target device dsPIC33CK64MC105 found.
Device Revision Id  = 0x1
Device Id = 0x99120000
UDID1 = 0e11e2
UDID2 = 004650
UDID3 = 0d0000
UDID4 = 000000
UDID5 = 0000e5
```

TABLE 28-3: DEVICE IDs FOR THE dsPIC33CK64MC105 FAMILY

Device	DEVID
dsPIC33CK64MC105	0x9912
dsPIC33CK64MC103	0x9911
dsPIC33CK64MC102	0x9910
dsPIC33CK32MC105	0x9902
dsPIC33CK32MC103	0x9901
dsPIC33CK32MC102	0x9900

其他內建功能

OTP、VREG 、BOR

- **User OTP Memory**
  - 內建64 bit OTP 記憶體
  - 此區只可寫一次，可用於存工廠資訊，防止出貨後被變動

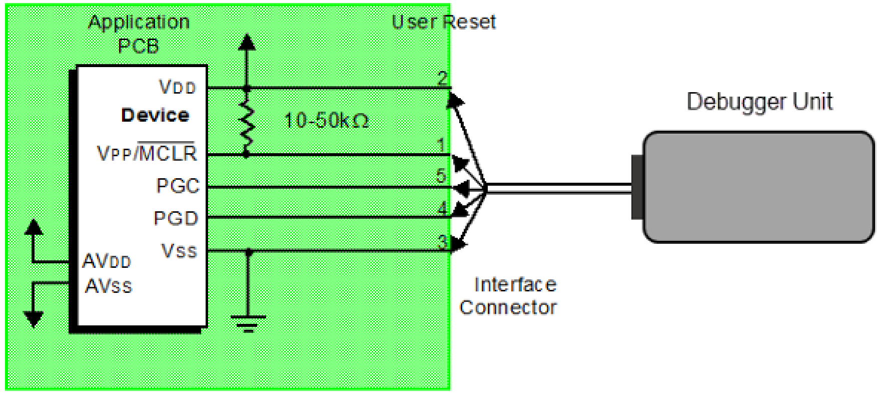
- **On-Chip Voltage Regulator(VREGCON)**
  - 可讓PLL電路跑在1.2V，達到省電，但此時系統也不能跑太快<8MHz

- **Brown-out Reset (BOR)**
  - 避免系統電壓不穩誤動作，穩定前**MCU**處於**RESET**狀態

其他內建功能

JTAG、Code Protection

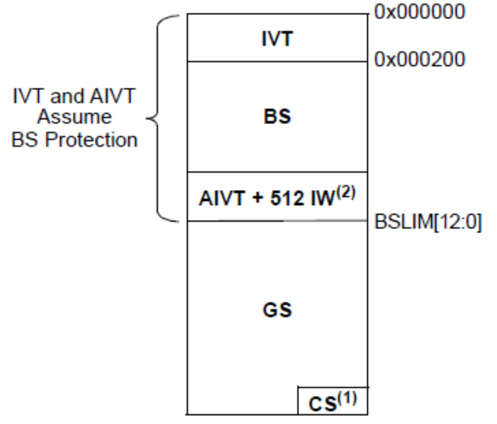
- **In-Circuit Serial Programming™(ICSP™)**
- **In-Circuit Debugger**
  - Microchip MCU的燒錄及除錯介面
  - 此IC有多組PGC及PGD，選一組連接到燒錄器(ICD、PICKit 等…)
  - 需搭陪設定，選錯debug時無法近入debug模式



- **Code Protection 及 CodeGuard™ Security**
  - 不同區，可做不同設定(No, Standard, Enhanced, High Security)
  - 開機區：Boot Segment (BS)
  - 應用程式區：General Segment(GS)
  - 硬體設定區： Configuration Segment (CS)

- IVT：中斷向量表 (Interrupt Vector Table)
- AIVT：備用中斷向量表 (Alternate Interrupt Vector Table)

FIGURE 28-3: SECURITY SEGMENTS EXAMPLE



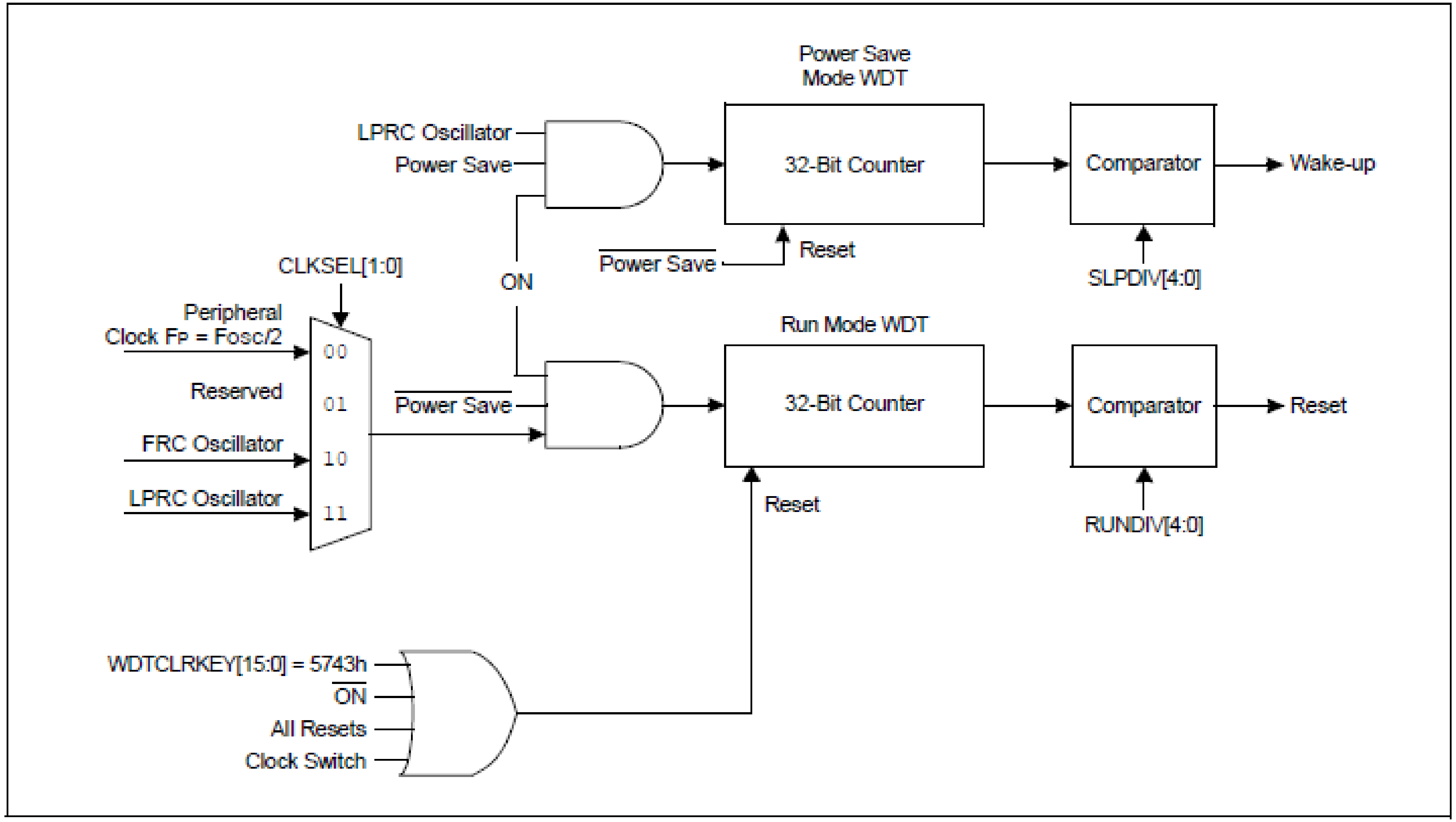
Note 1: If CS is write-protected, the last page (GS + CS) of program memory will be protected from an erase condition.

2: The last half (256 IW) of the last page of BS is unusable program memory.

Watchdog架構圖

- 可以用Config或軟體啟動
- WDT也有多個CLK來源
- 一般用跟主系統不同
- 常見32KHz
  所以Timer精細度比較差

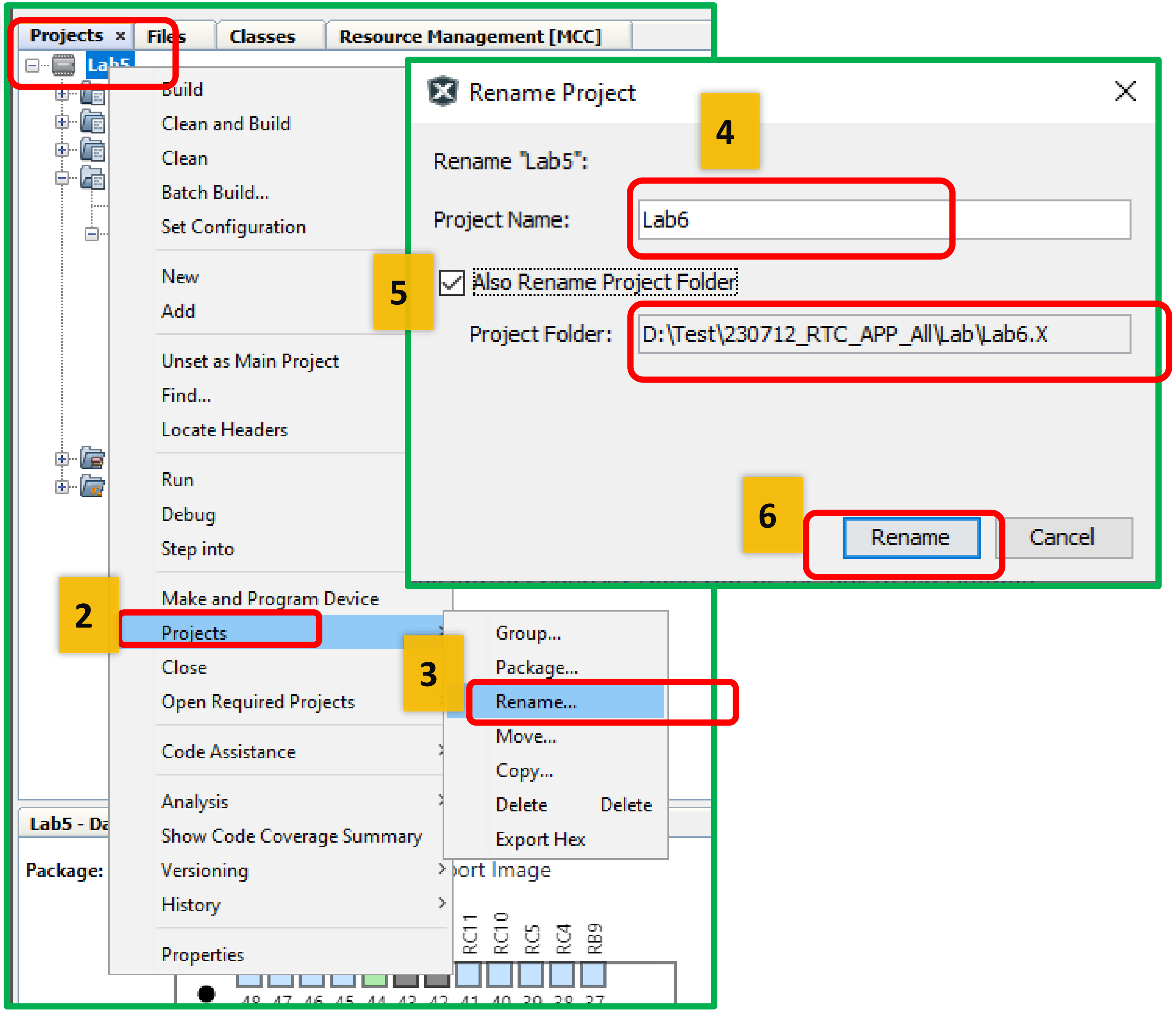
FIGURE 28-2: WATCHDOG TIMER BLOCK DIAGRAM



1.建立Lab5

將Lab 5改名成Lab6

1) 到Projects 在Lab4上
   滑鼠點右鍵
2) 找到Projects
3) 找到Rename
4) 修改成Lab6
5) 勾選順便改資料夾名
6) 按 Rename - 完成名稱修改



Rename Project

Rename "Lab5":

Project Name: Lab6

☑ Also Rename Project Folder

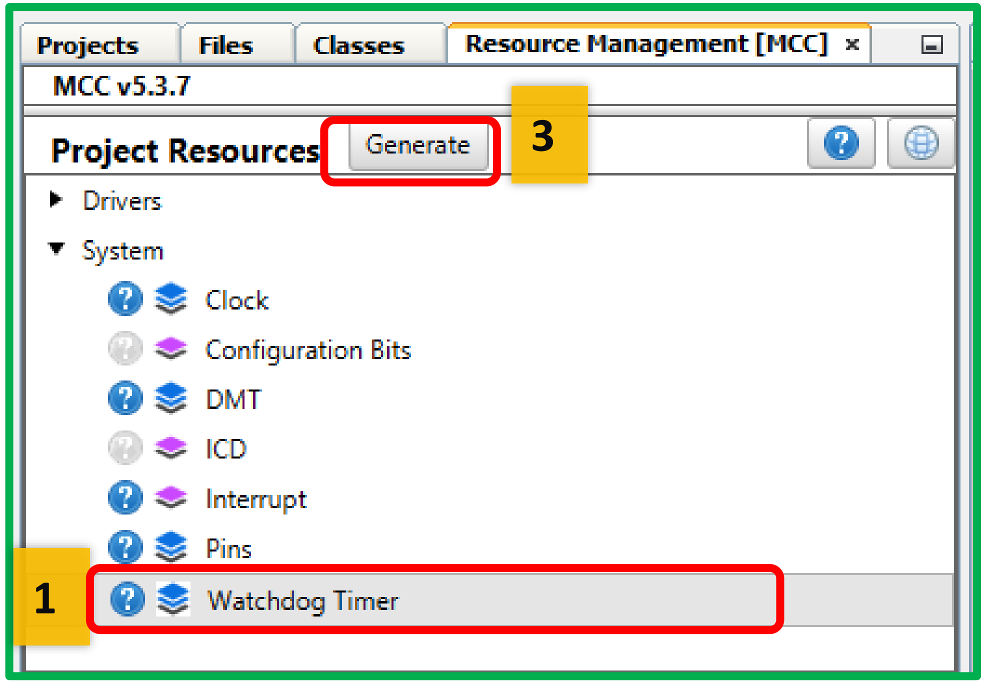
Project Folder: D:\Test\230712_RTC_APP_All\Lab\Lab6.X

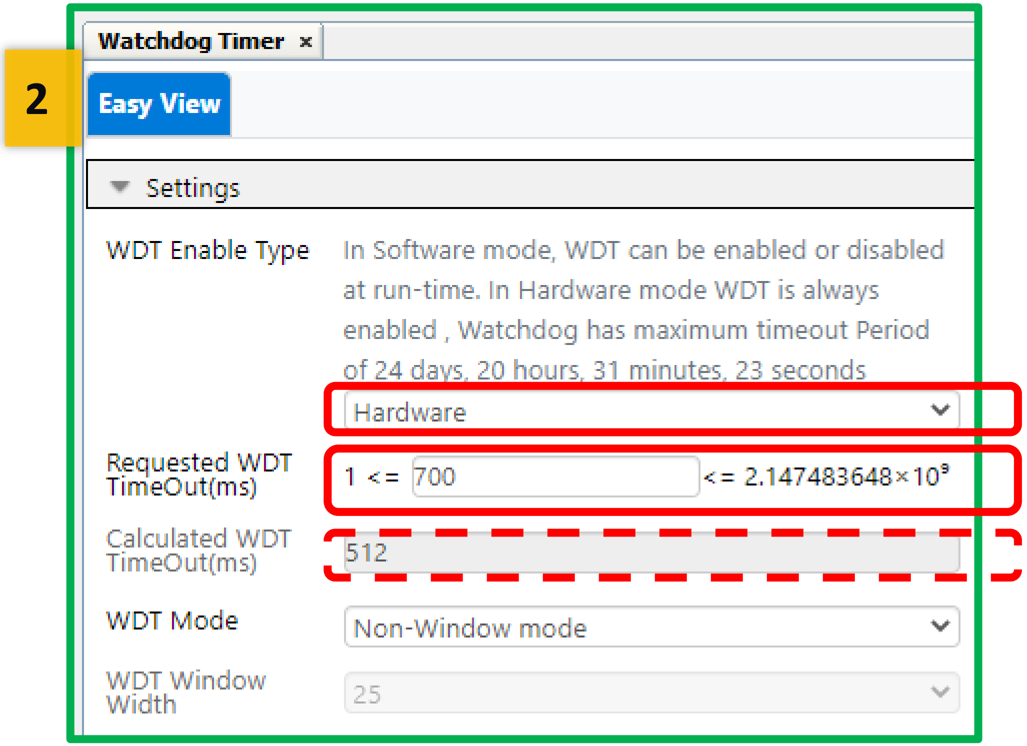
Rename　Cancel

2. 開啟MCC

設定Watchdog

1) 到Project Resources
   找System ,找Watchdog Timer,點選
2) 到Watchdog Timer的 Easy View
   選Hardware啟動
   改700ms 實際 512ms
3) 產生Code





WDT Enable Type: In Software mode, WDT can be enabled or disabled at run-time. In Hardware mode WDT is always enabled , Watchdog has maximum timeout Period of 24 days, 20 hours, 31 minutes, 23 seconds

Hardware

Requested WDT TimeOut(ms): 1 <= 700 <= 2.147483648×10^9

Calculated WDT TimeOut(ms): 512

WDT Mode: Non-Window mode

WDT Window Width: 25

3.結果

Lab6

1) 燒錄測試

2) 結果

小板上

LED0 看起來恆亮

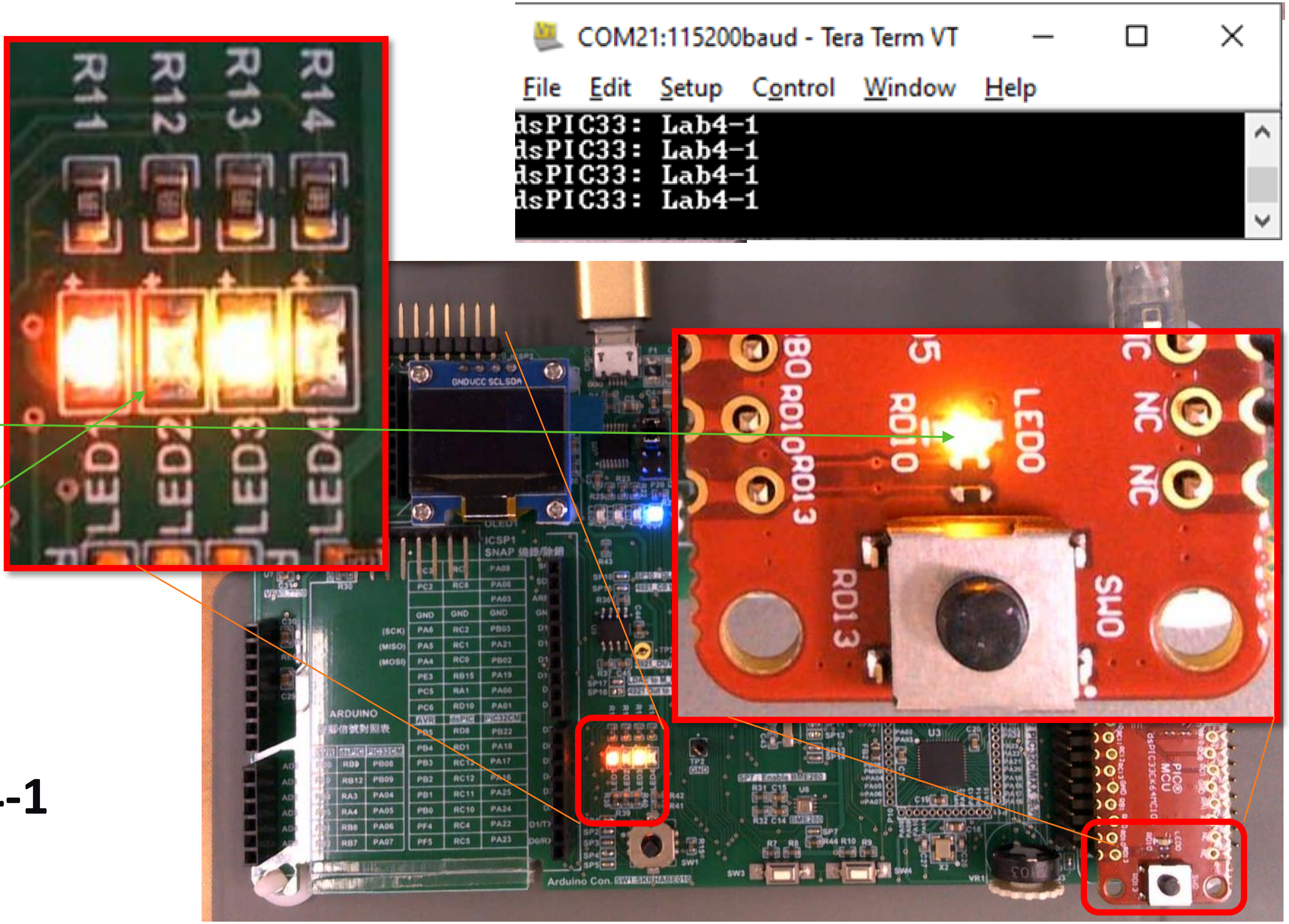
大板上

LED1 看起來恆亮

LED2 小閃一下

LED3~4交互快閃

PC顯示 dsPIC33: Lab4-1

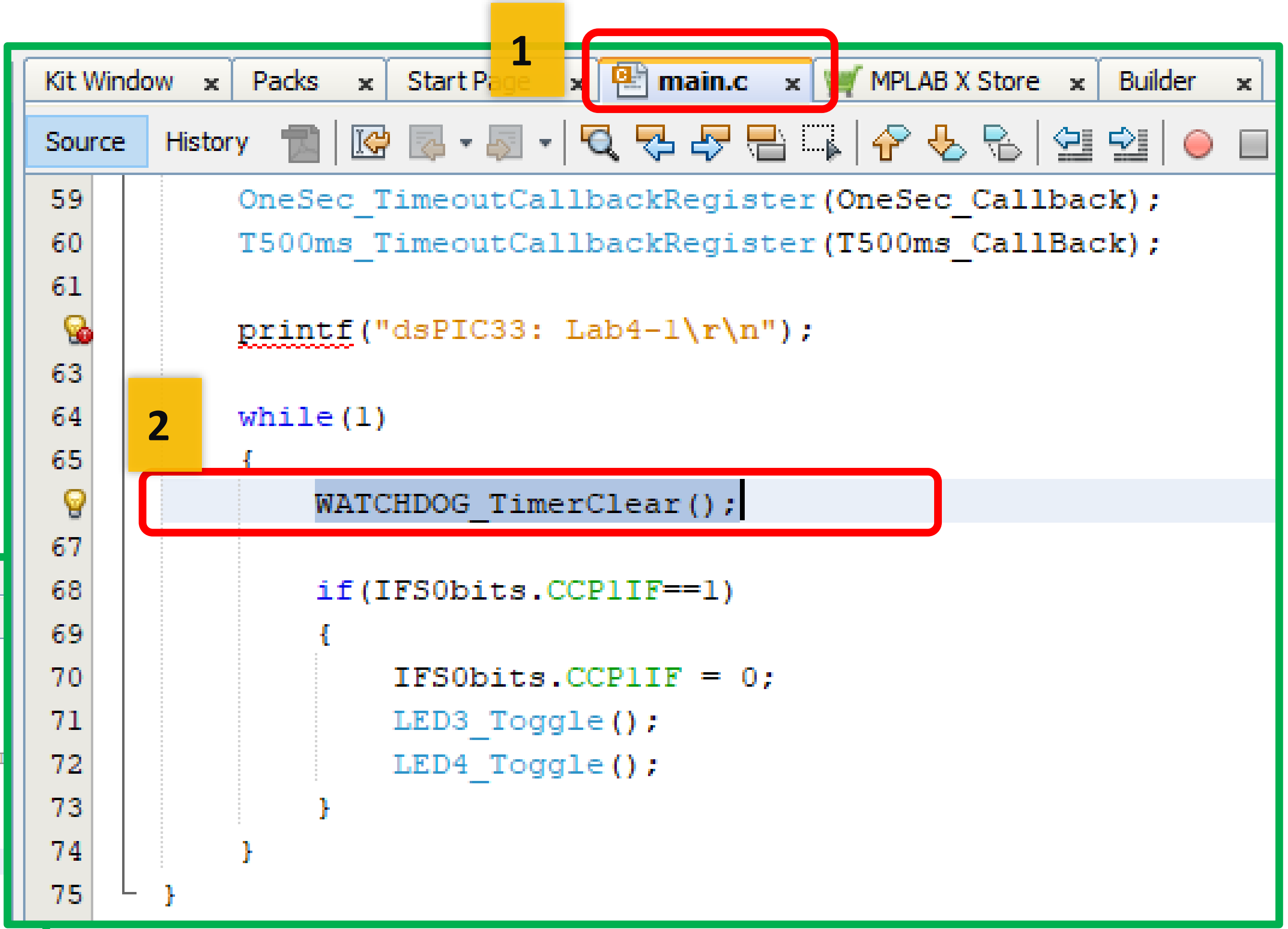

COM21:115200baud - Tera Term VT
File Edit Setup Control Window Help
R11
R12
R13
R14
LED0
RD10
SW0
RD13
ARDUINO
ICSP1
OLED1
Arduino Con.

```
dsPIC33: Lab4-1
dsPIC33: Lab4-1
dsPIC33: Lab4-1
dsPIC33: Lab4-1
```

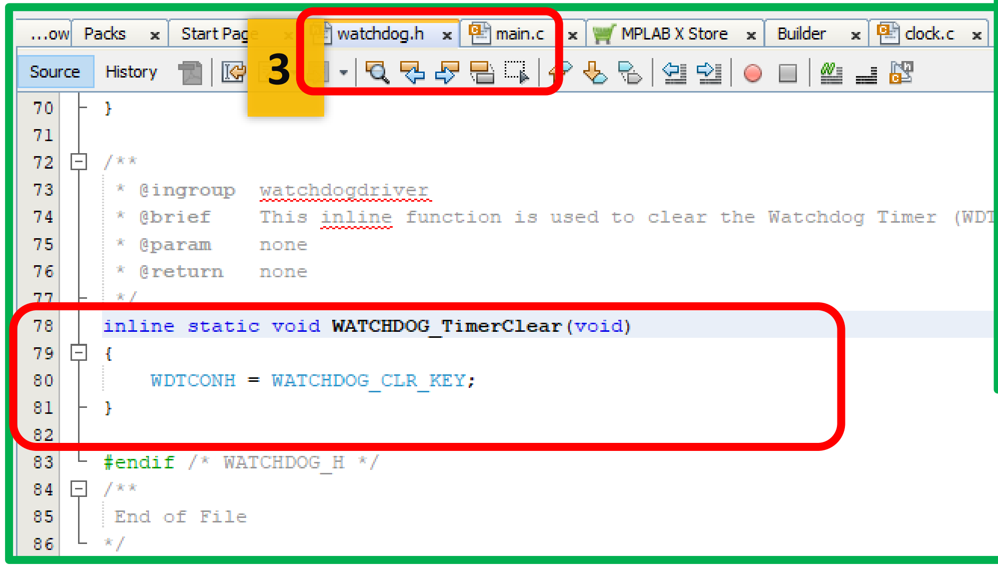
4. 修正

加入Clear WDT

1) 開啟main.c
2) 到while(1){}
   加入WATCHDOG_TimerClear();
3) 函式來源 watchdog.h
4) 燒錄



```
    OneSec_TimeoutCallbackRegister(OneSec_Callback);
    T500ms_TimeoutCallbackRegister(T500ms_CallBack);

    printf("dsPIC33: Lab4-1\r\n");

    while(1)
    {
        WATCHDOG_TimerClear();

        if(IFS0bits.CCP1IF==1)
        {
            IFS0bits.CCP1IF = 0;
            LED3_Toggle();
            LED4_Toggle();
        }
    }
}
```



```
}

/**
 * @ingroup   watchdogdriver
 * @brief     This inline function is used to clear the Watchdog Timer (WDT
 * @param     none
 * @return    none
 */
inline static void WATCHDOG_TimerClear(void)
{
    WDTCONH = WATCHDOG_CLR_KEY;
}

#endif /* WATCHDOG_H */
/**
 End of File
*/
```

5.結果

回復原來樣貌

1) 燒錄測試

2) 結果

小板上

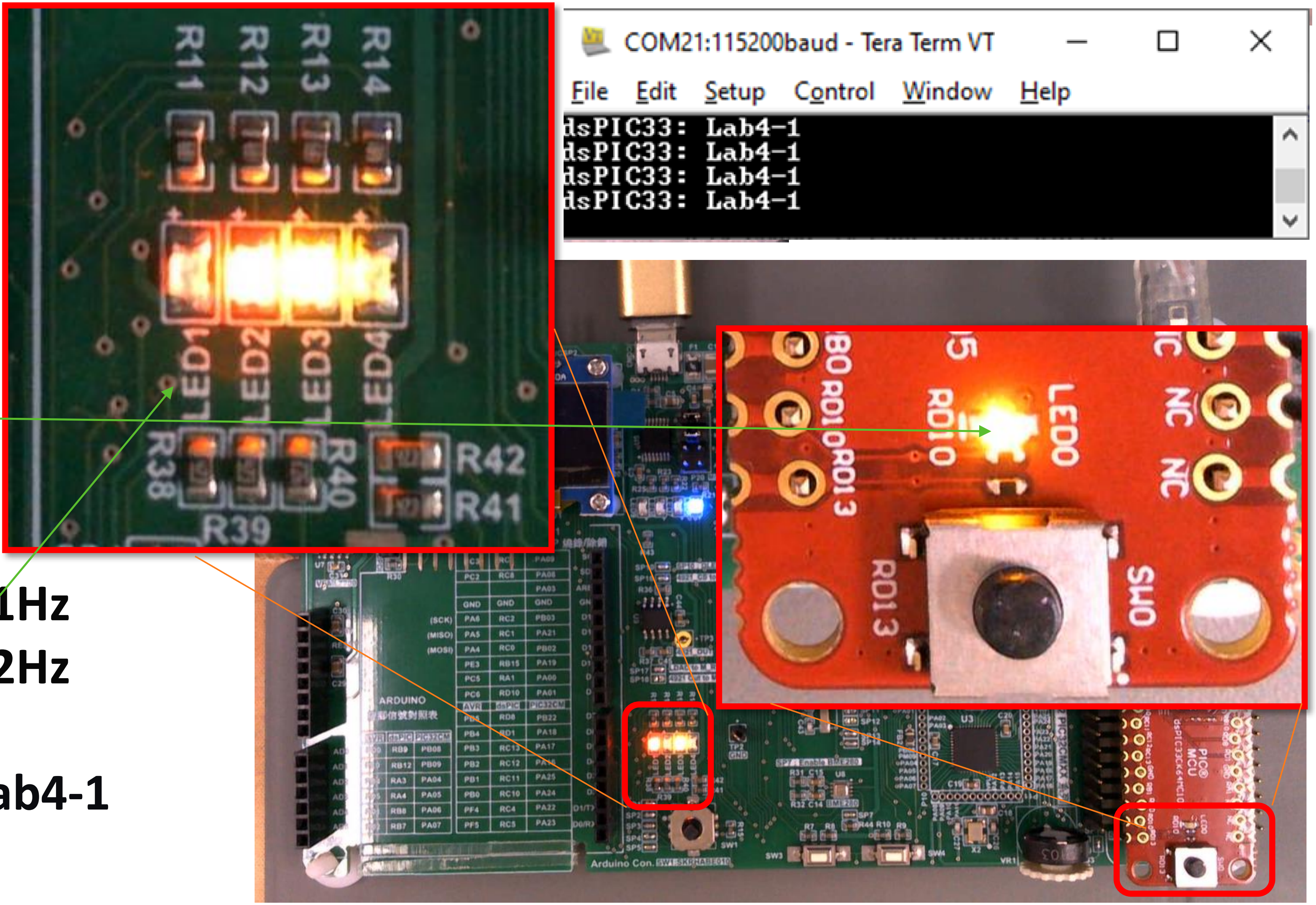
LED0閃爍 1Hz

大板上

LED1~2交互慢閃 1Hz

LED3~4交互快閃 2Hz

PC顯示 dsPIC33: Lab4-1



```
dsPIC33: Lab4-1
dsPIC33: Lab4-1
dsPIC33: Lab4-1
dsPIC33: Lab4-1
```

感謝各位的參與

參考文件

- **Oscillator Module with High-Speed PLL**
  - https://ww1.microchip.com/downloads/en/DeviceDoc/dsPIC33-PIC24-FRM-Oscillator-Module-with-High-Speed-PLL-70005255b.pdf
- **Dual Watchdog Timer**
  - https://ww1.microchip.com/downloads/aemDocuments/documents/MCU16/ProductDocuments/ReferenceManuals/dsPIC33-PIC24-FRM-Dual-Watchdog-Timer-DS70005250.pdf
- **CodeGuard™ Intermediate Security**
  - https://ww1.microchip.com/downloads/aemDocuments/documents/OTH/ProductDocuments/ReferenceManuals/70005182a.pdf