

處理器設計與實作

實習講義

編撰者

助教：邱健鳴、蔡森至、張瑞元、曾柏翔、許元杰
暨成大電通所計算機架構與系統研究室CASLAB

國立成功大學電機系與電腦與通信工程研究所

Lab6-實驗大綱與目的

1. ARC簡介
2. 實驗環境
 1. License申請
 2. Digilent Driver 安裝
 3. MIDE Lite 安裝
3. 實作一(連接ARC板)
4. 實作二(Build & Debug Project)

ARC簡介

ARC為Synopsys公司出產的指令集，並且有多種系列的

ARC Processors:

+ ARC HS Family	High-speed, 32-bit multicore processors for high-end embedded applications
+ ARC 700 Family	Mid-range to high-end embedded and host applications
+ ARC 600 Family	General-purpose embedded processors with DSP capabilities
+ ARC EM Family	Compact, ultra low-power processors for deeply embedded applications
+ Audio Processors	Single- and dual-core 32-bit audio processors
+ ARC Options	DSP, floating-point, trace and secure pipeline options for ARC cores
+ ARC Audio Solution	ARC audio processors with powerful dual-MAC DSP
+ ARC Tools & Ecosystem	Commercial & open-source tools for development of ARC-based systems

ARC簡介

Synopsys' DesignWare® ARC® Processors are a family of 32-bit CPUs that SoC designers can optimize for a wide range of uses, from deeply embedded to high-performance host applications in a variety of market segments. Designers can differentiate their products by using patented configuration technology to tailor each ARC processor instance to meet specific performance, power and area requirements. The DesignWare ARC processors are also extendable, allowing designers to add their own custom instructions that dramatically increase performance. Synopsys' ARC processors have been used by over 170 customers worldwide who collectively ship more than 1.5 billion ARC-based chips annually.

Ref.

<http://www.synopsys.com/IP/PROCESSORIP/ARCPROCESSORS/Pages/default.aspx>

ARC簡介

今日我們的課程是用到EM系列的ARC processor:

The DesignWare® ARC® EM Family of embedded processors is based on the scalable ARCV2 Instruction Set Architecture (ISA) and is optimized for energy and performance efficiency (DMIPS/mW and DMIPS/mm²). The ARC EM family includes the EM4 (cacheless) and EM6 (instruction and data caches) processor cores, designed for use in power and area-sensitive embedded applications. They offer industry-leading performance efficiency of up to 1.77 DMIPS/MHz, with minimal area and power consumption.

Ref.

<http://www.synopsys.com/IP/ProcessorIP/ARCProcessors/ARCEM/Pages/arc-em-processor-family.aspx>

ARC簡介

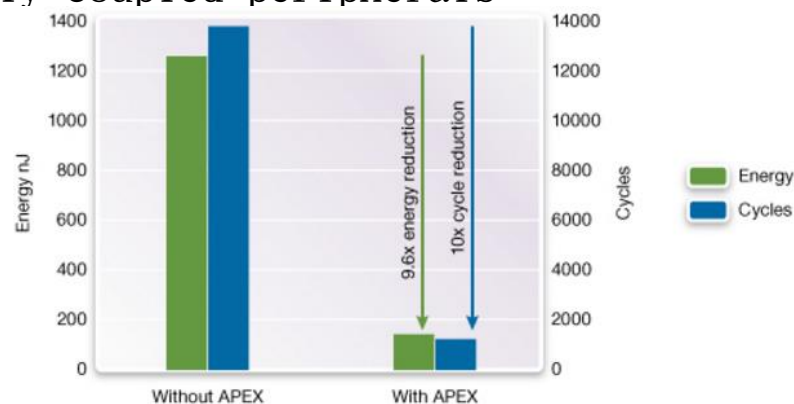
ARC主打的特色之一：APEX

ARC Processor EXtension (APEX) technology enables ARC users to easily add their own custom hardware to the processor, dramatically boosting performance and/or reducing power consumption for their targeted application(s). ARC processors can be extended with:

- User-defined instructions
- User-supplied hardware (e.g., Verilog RTL)
- Core registers
- Auxiliary registers
- Condition & status codes
- Memory mapped blocks and closely coupled peripherals

Ref.

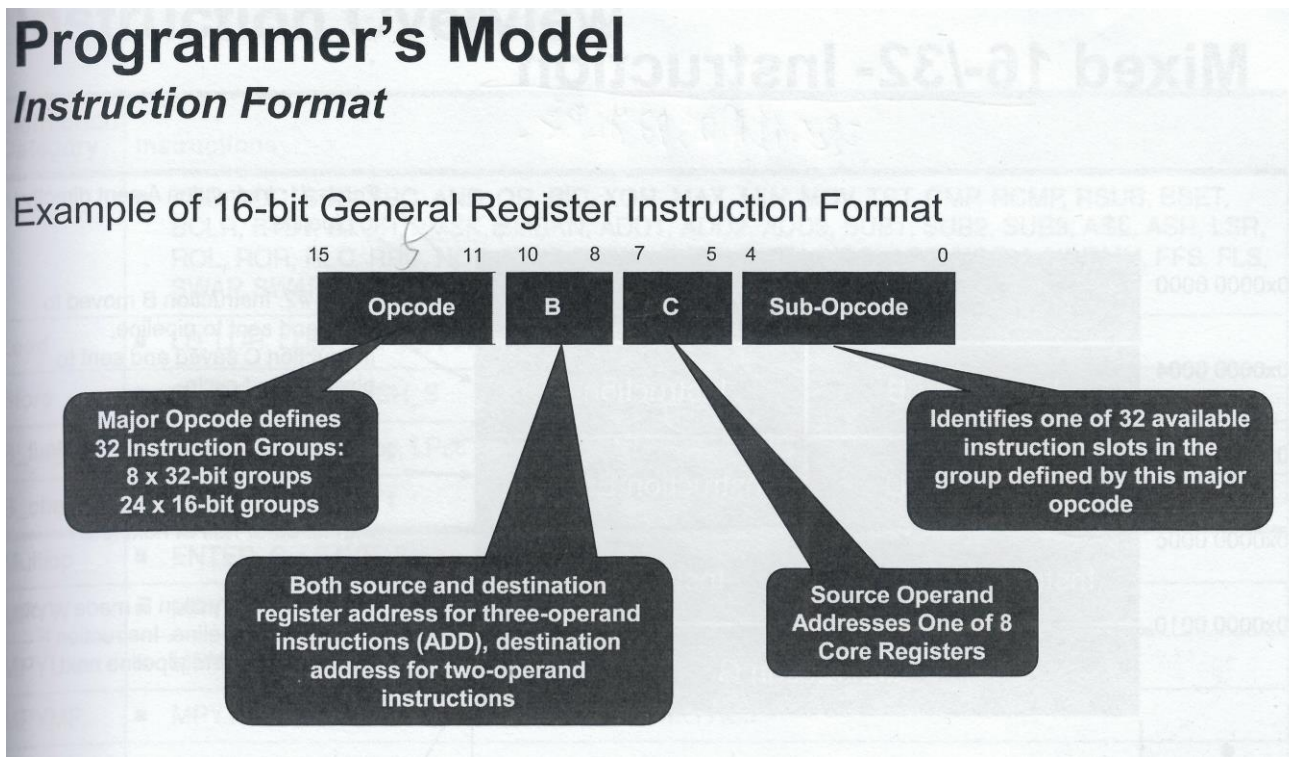
<http://www.synopsys.com/IP/ProcessorIP/ARCProcessors/ARCEM/Pages/arc-em-processor-family.aspx>



Power and cycle count reduction running sensor application software with APEX accelerators

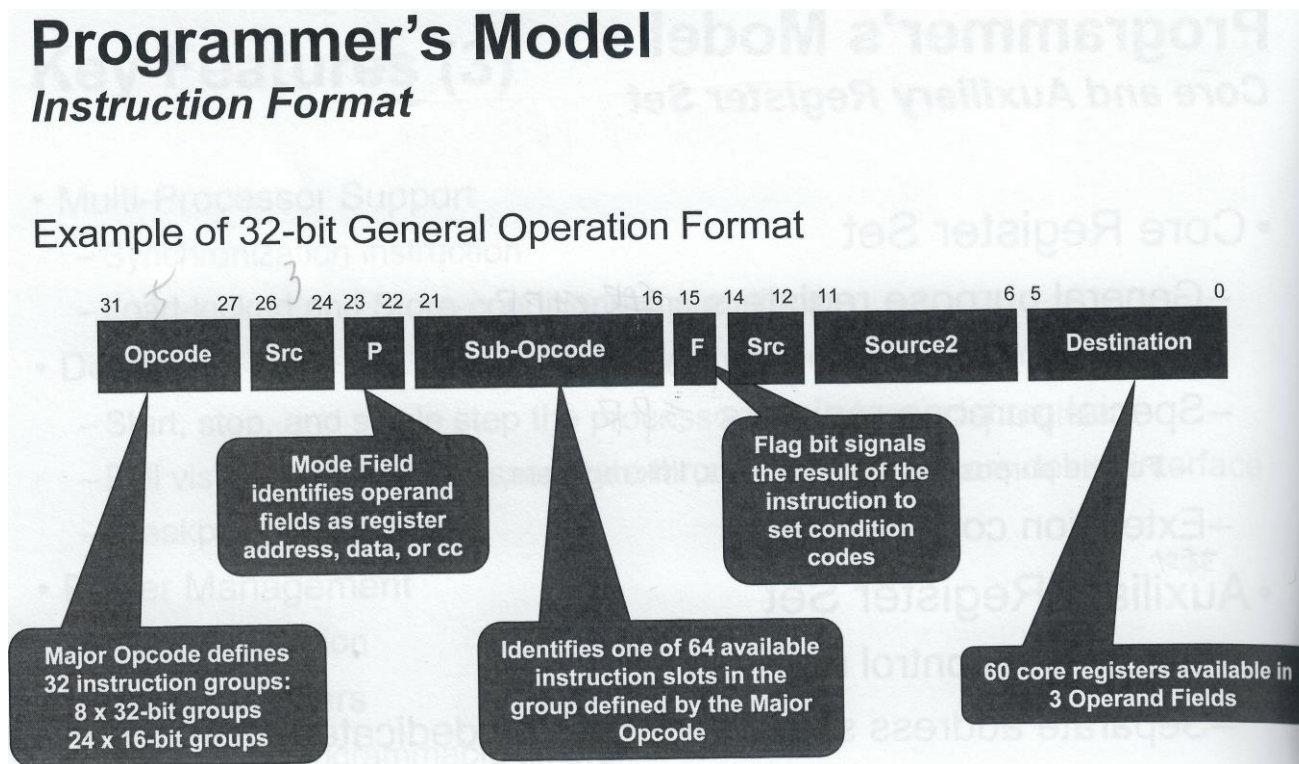
ARC 簡介

ARCV2 的 16bit instruction encoding:



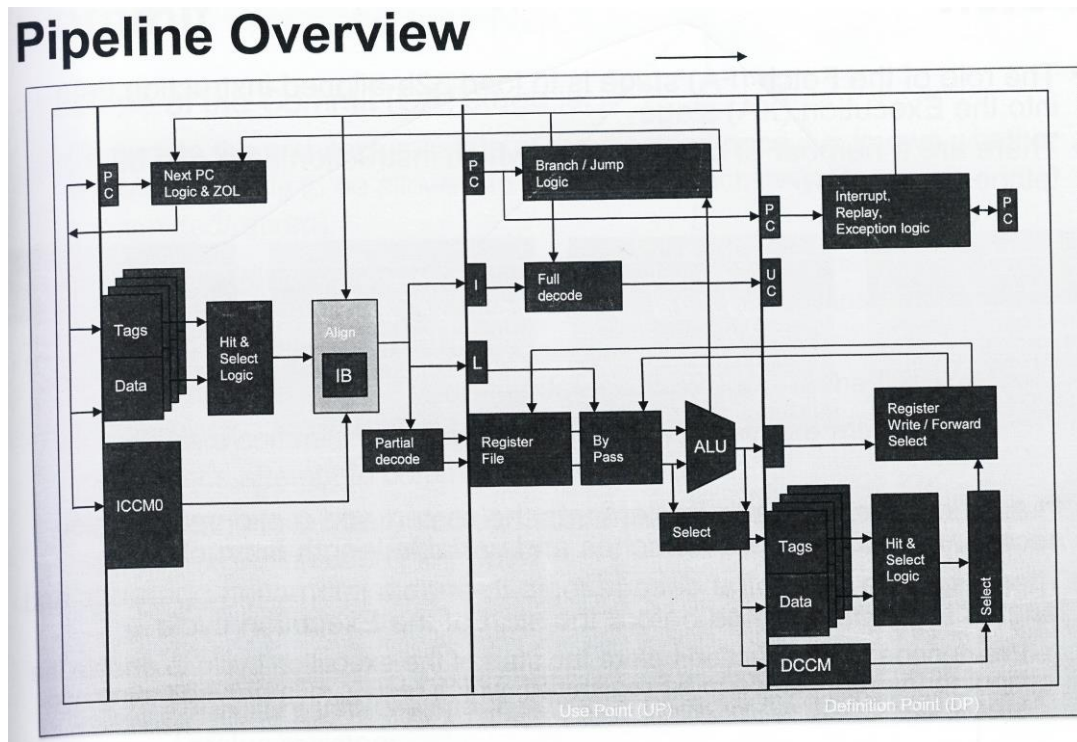
ARC簡介

ARCV2 的32bit instruction encoding:



ARC 簡介

ARC EM 系列的processor 大致上分為3 stage pipeline，而它涵蓋 1. Fetch 2. Execute 3. Commit。以下為其架構圖：



ARC簡介

Board Overview

The ARC EM Starter Kit consists of two boards:

FPGA module with a Xilinx Spartan-6 XC6SLX150-3FGG484C FPGA device.

Base board with extension connectors and peripherals.

The kit has the following on-board peripheral devices:

2x16-bit wide 1 Gbit (128 MB) DDR3 SDRAM

128 Mbit (16 MB) SPI Flash memory

SD card reader

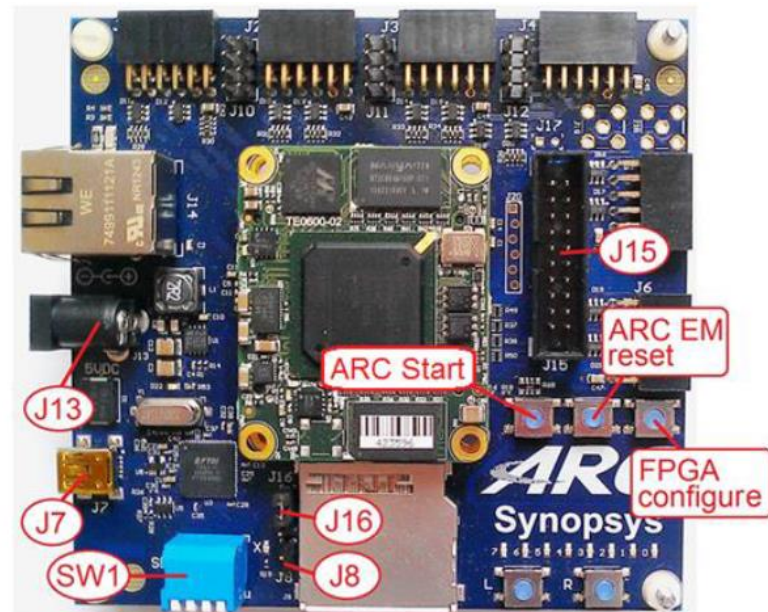
LEDs

Push-buttons

DIP switches

Seven Pmod connectors

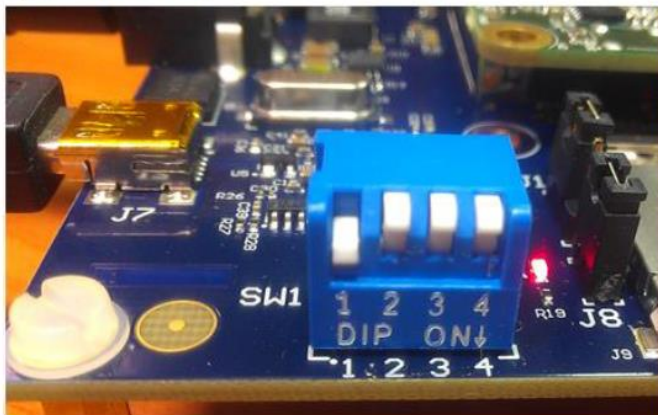
On-board clock oscillator



ARC簡介

本次實驗使用的板子組，ARC EM Starter Kit，可以調整 Base board上的SW1上的決定要燒入到FPGA (SPARTAN-6)的ARC EM processor版本。

Figure 7 SW1 DIP Switch



The definitions of bits 1 and 2 are described in the [Table 3](#).

Table 3 Selecting the Configuration

Bit 1	Bit 2	Configuration
OFF	OFF	ARC_EM5D (and ARC_EM4)
ON	OFF	ARC_EM7D (and ARC_EM6)
OFF	ON	ARC_EM7DFPU (and ARC_EM6PFU) This is ARC_EM7D with Floating Point Unit
ON	ON	Reserved

ARC簡介

不同的ARC EM processor比較:

Standard Features							
Processor	Cache Size (I&D)	Closely Coupled Memories (Up to 2 MB)	DSP included	XY Memory	ECC/Parity	Watchdog Timer	APEX
EM4		✓			✓	✓	✓
EM6	32k	✓			✓	✓	✓
EM5D		✓	✓		✓	✓	✓
EM7D	32k	✓	✓		✓	✓	✓
EM9D		✓	✓	✓	✓	✓	✓
EM11D	32k	✓	✓	✓	✓	✓	✓

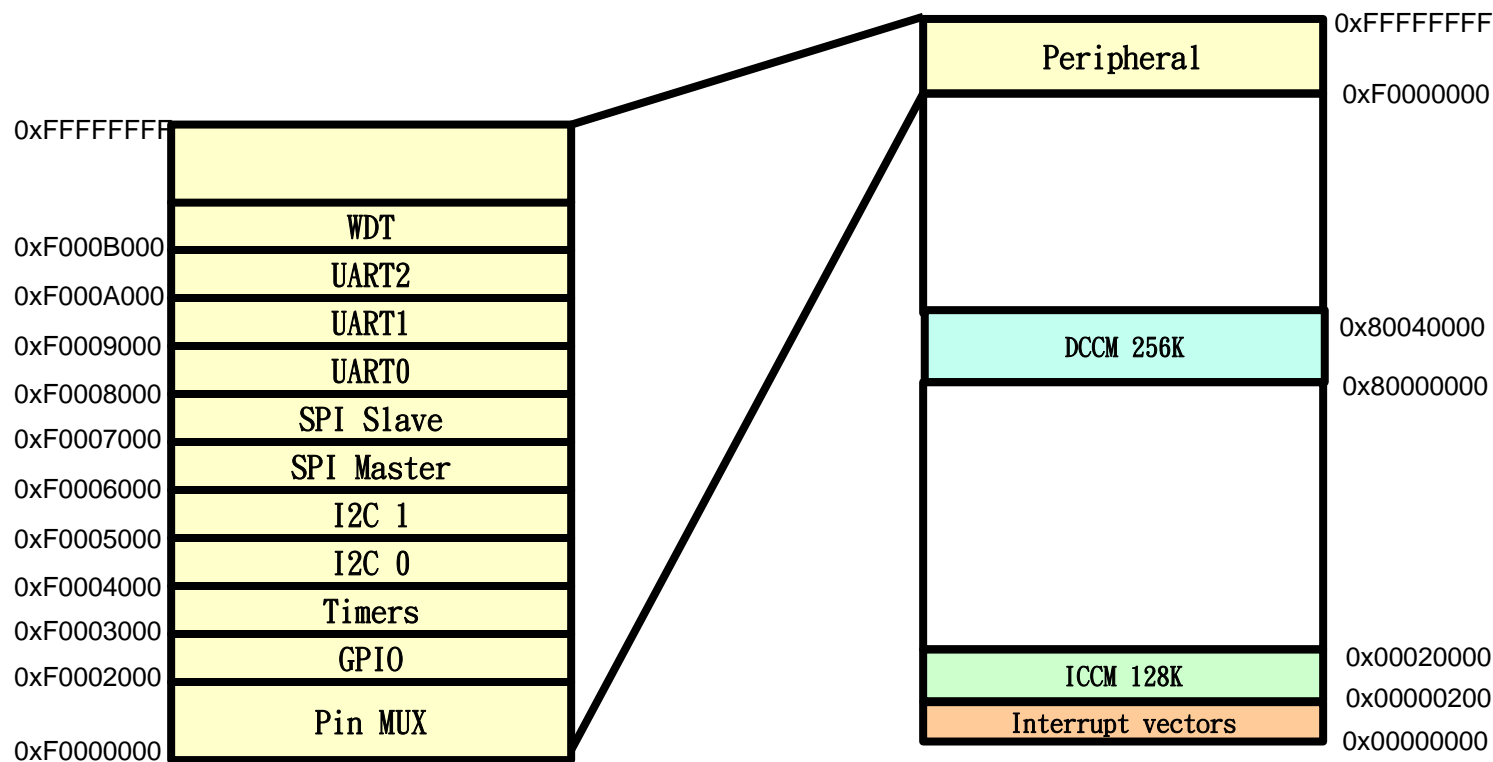
Licensable Options								
Processor	FPU	Trace	MPU	μDMA	Safety Enhancement Package	Enhanced Security Package	CryptoPack	ARConnect
EM4	✓	✓	✓	✓	✓	✓	✓	✓
EM6	✓	✓	✓	✓	✓		✓	✓
EM5D	✓	✓	✓	✓	✓	✓	✓	✓
EM7D	✓	✓	✓	✓	✓		✓	✓
EM9D	✓	✓	✓	✓			✓	✓
EM11D	✓	✓	✓	✓			✓	✓

Ref.

<http://www.synopsys.com/IP/ProcessorIP/ARCProcessors/ARCEM/Pages/arc-em-processor-family.aspx>

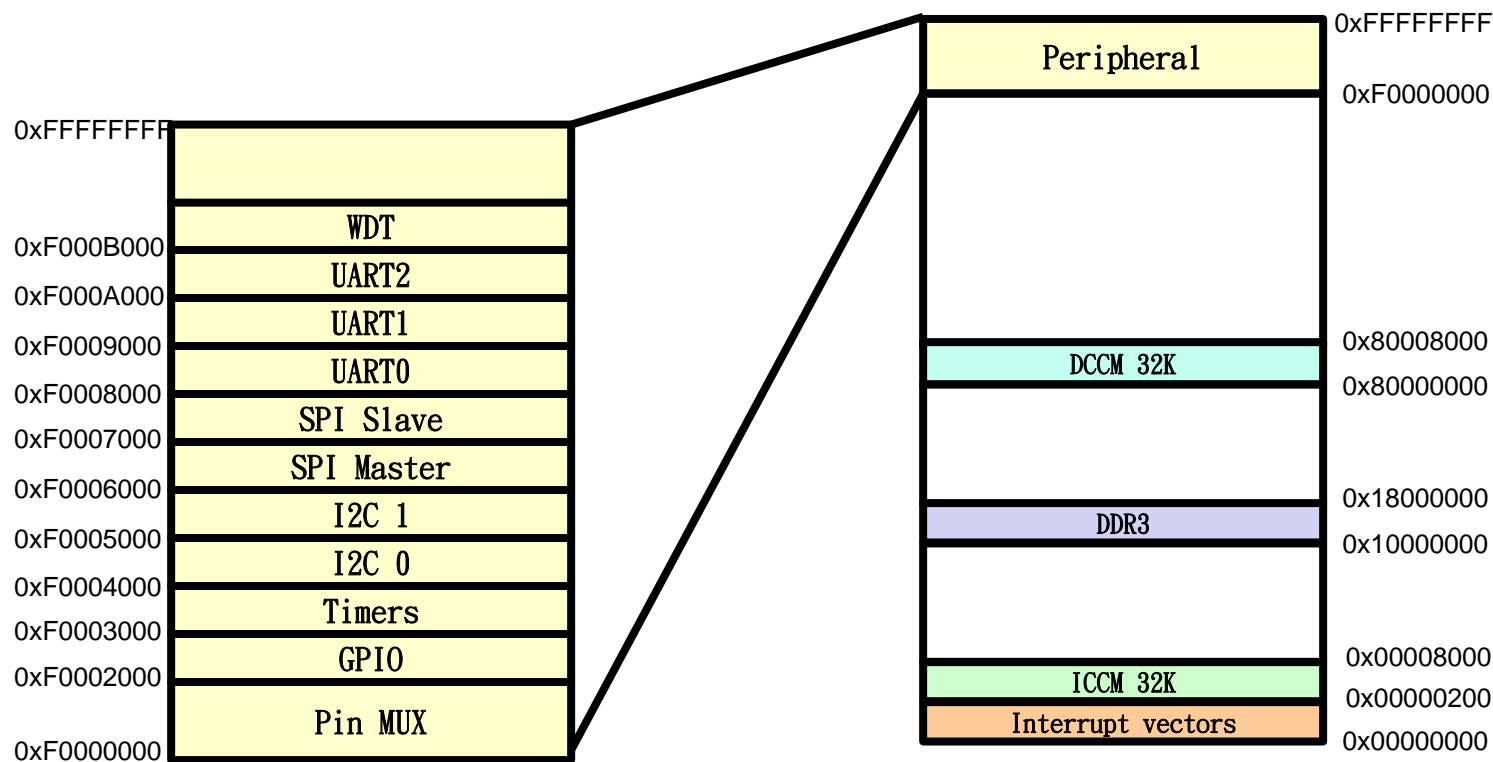
ARC簡介

此為ARC em 5d 的Memory map:



ARC簡介

此為ARC em 7d 的Memory map:



Lab6-實驗大綱與目的

1. ARC簡介
2. 實驗環境
 1. License申請
 2. Digilent Driver 安裝
 3. MIDE Lite 安裝
3. 實作一(連接ARC板)
4. 實作二(Build & Debug Project)

License 申請

之後使用到MetaWare
需要用到License，可
以至ARC網站申請

http://www.synopsys.com/cgi-bin/arcmwtk_lite/reg1.cgi

The screenshot shows the Synopsys website's registration page for the DesignWare ARC MetaWare Toolkit Lite Edition. The page includes a navigation bar with links like TOOLS, IP, and PROCESSOR IP. The main heading is "Download the DesignWare ARC MetaWare Toolkit Lite Edition". Below this, there is a paragraph explaining the availability of the toolkit for academic, personal, and noncompetitive evaluation purposes. A link to the "End-User License Agreement" is provided. The registration form itself contains several fields: Business Email (e24996544@mail.ncku.edu.tw), First Name (SenChih), Last Name (Tsai), Phone (0935508722), Job Role (Other), Job Title (Other), Country (Taiwan), Address (No.1, Dasyue Rd., East District, Tainan City 701, Taiwan), City (Tainan), State/Province, and Postal/Zip Code (701). There are also dropdown menus for "Which of the following best describes you or your business?" (Student), "What is the name of your current company, university, or affiliation?" (National Cheng Kung University), "For what purpose will you use the ARC MetaWare Toolkit Lite Edition?" (Academic Instruction or Co), and "How did you learn about the ARC MetaWare Toolkit Lite Edition?" (From my professor). Two checkboxes are present: "I agree to the terms and conditions of the ARC MetaWare Toolkit Lite Edition End-User Software License Agreement" and "Remember me (requires browser cookies to be enabled)". A "continue >>" button is at the bottom.

實驗環境 (1/10)

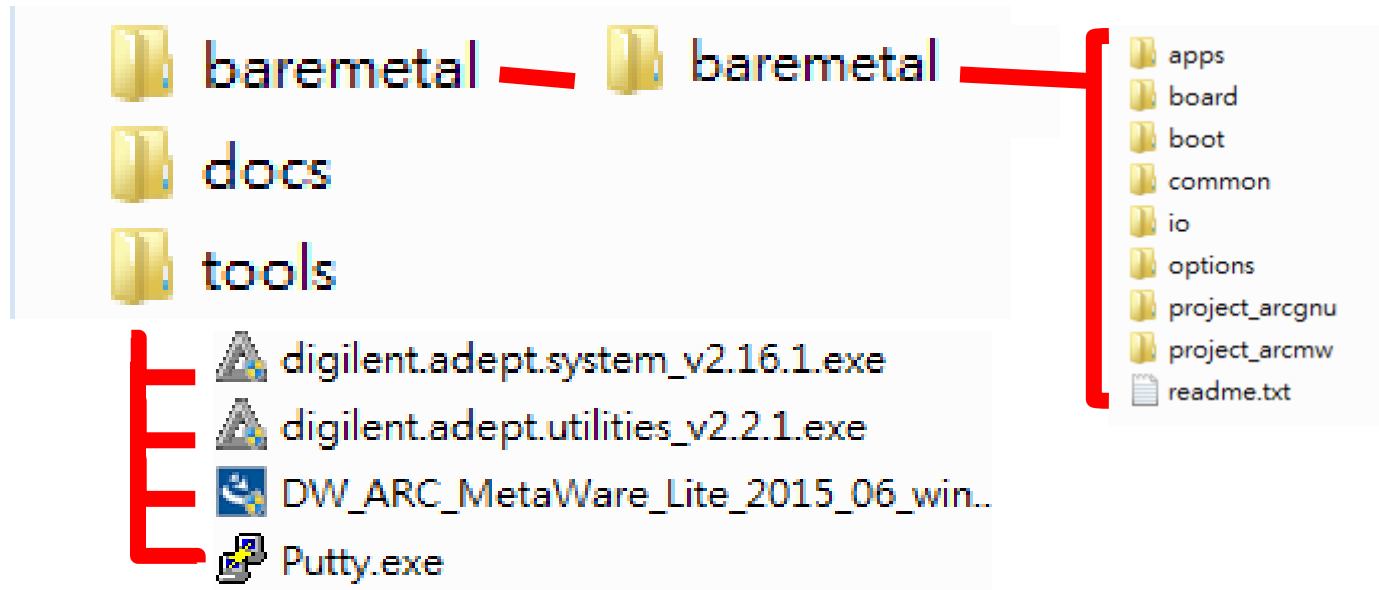
Step1. 準備筆電、電腦。建議作業系統: Windows 7 x64
並且準備好ARC板子及mini USB - USB 線。



實驗環境 (2/10)

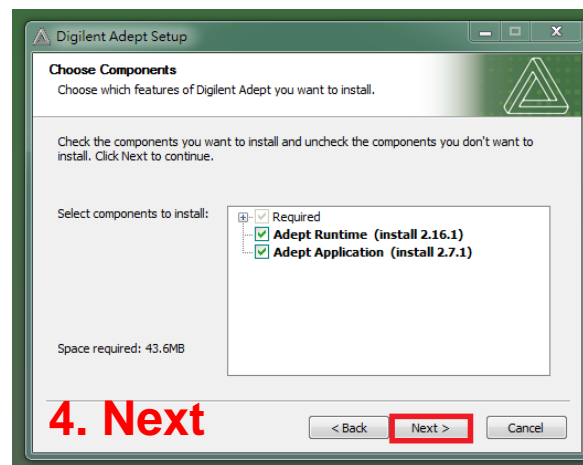
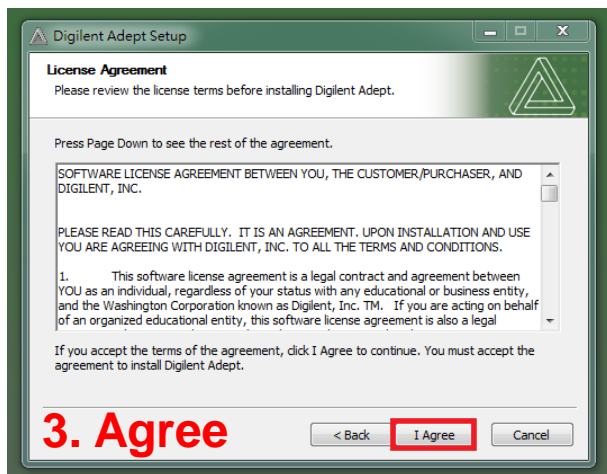
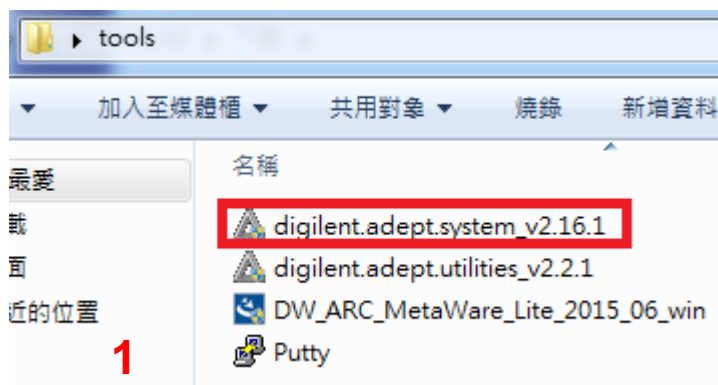
Step2. 在筆電、電腦上準備實驗需要的檔案

1. 專案檔 (baremetal)
2. 軟體、驅動安裝檔 (tools)



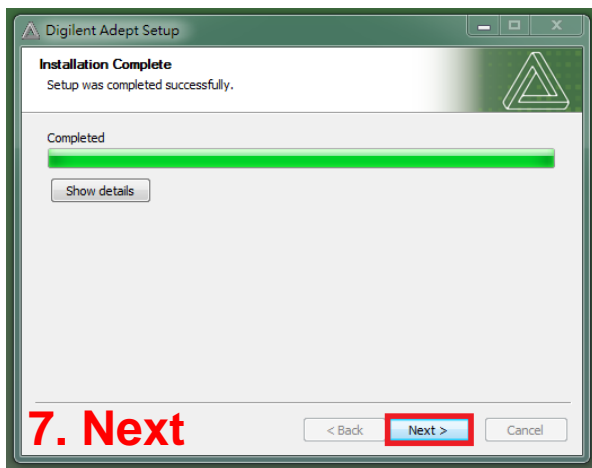
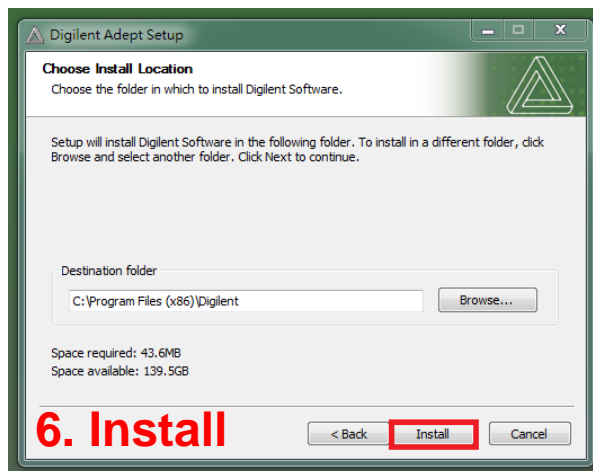
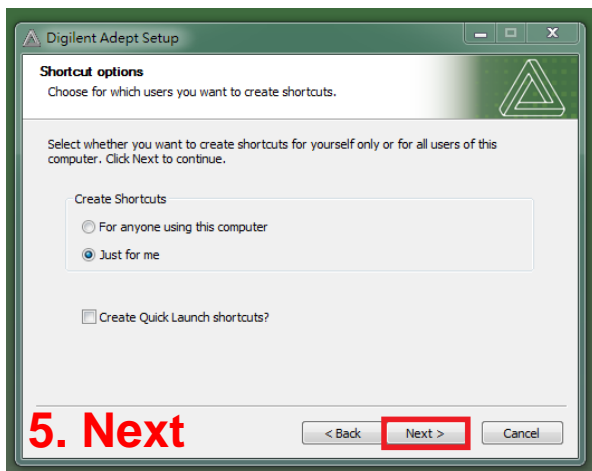
實驗環境 (3/10)

Step3. 在筆電上安裝 digilent system



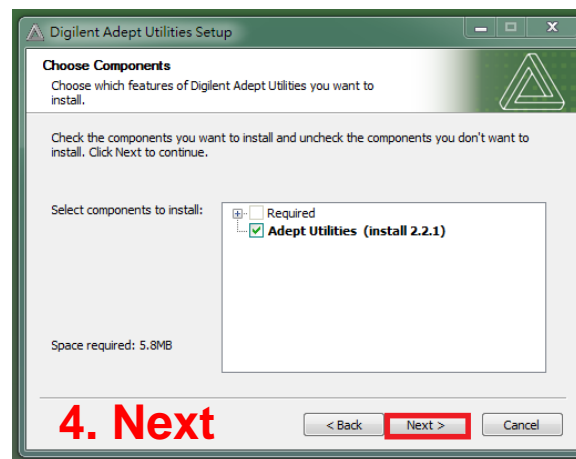
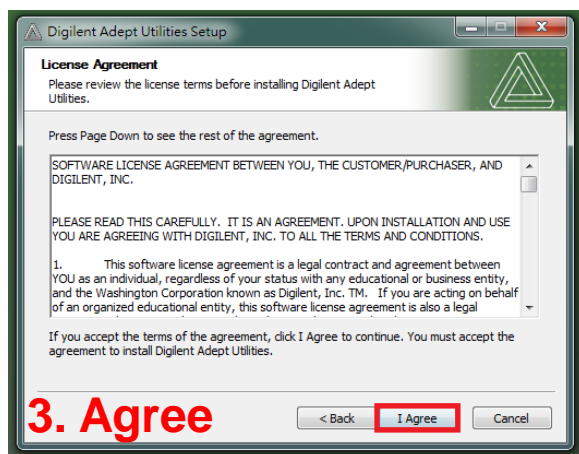
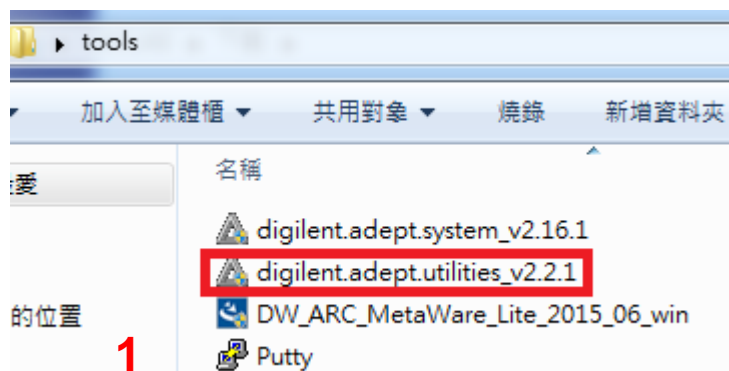
實驗環境 (4/10)

Step3. 在筆電上安裝 digilent system



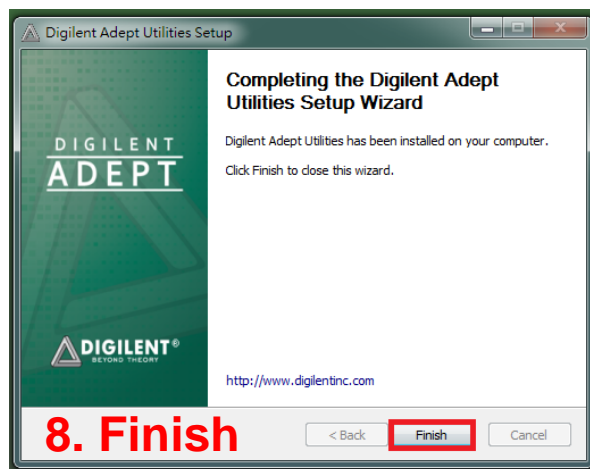
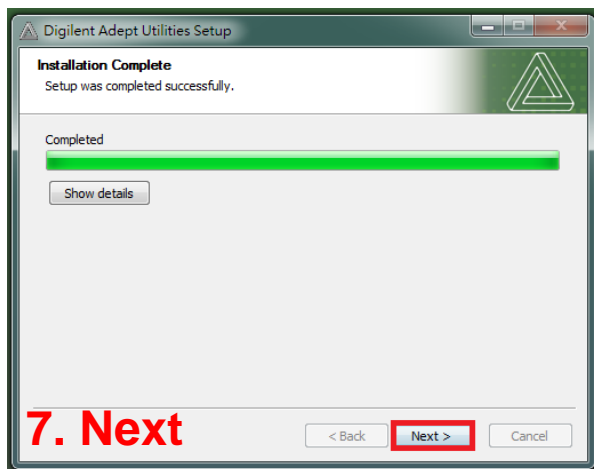
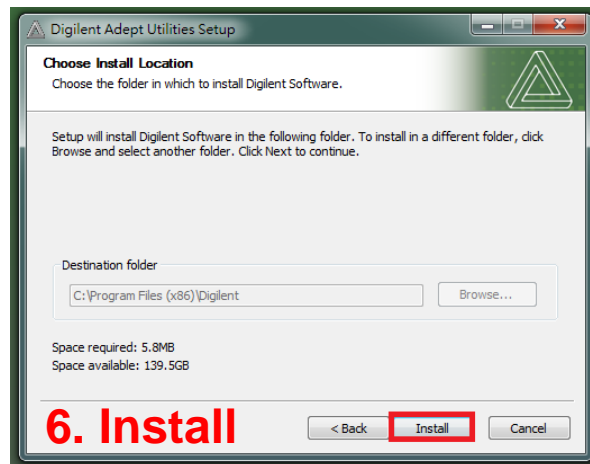
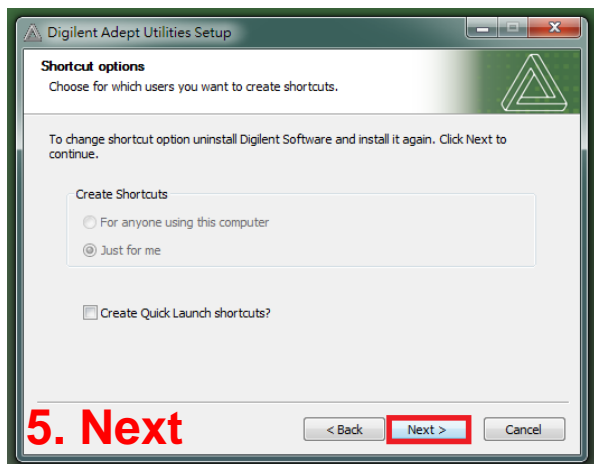
實驗環境 (5/10)

Step4. 在筆電上安裝 digilent utilities



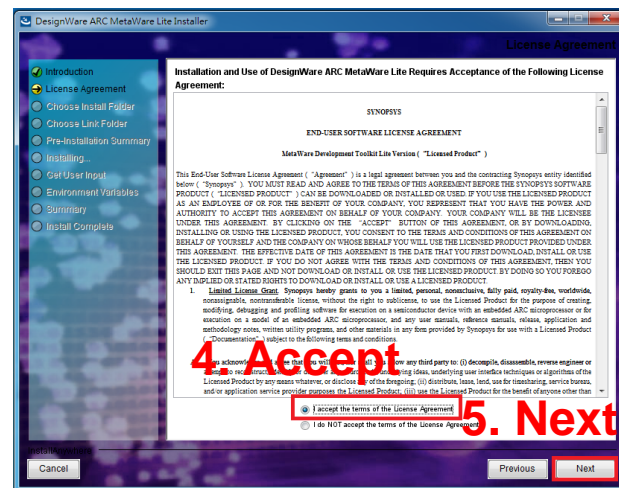
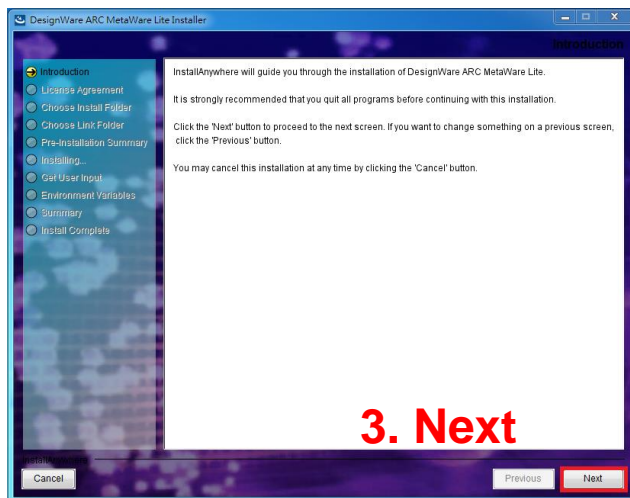
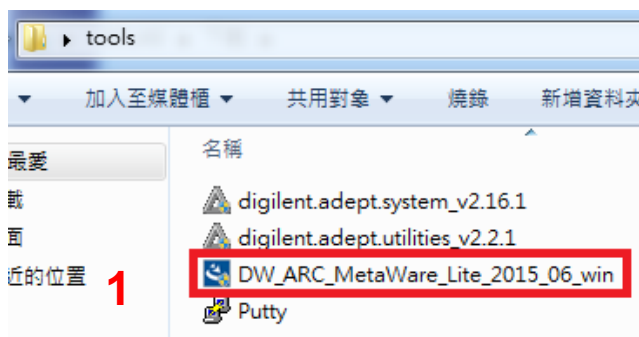
實驗環境 (6/10)

Step4. 在筆電上安裝 digilent utilities



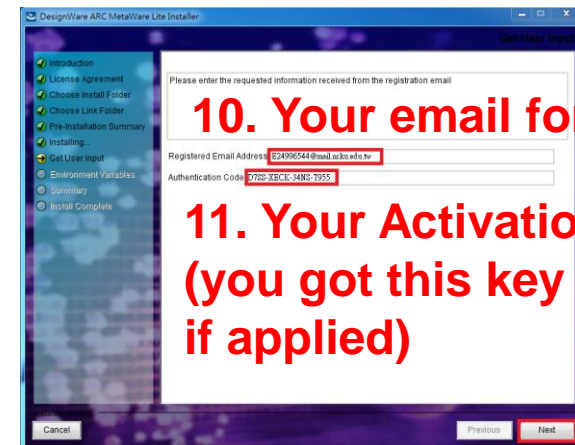
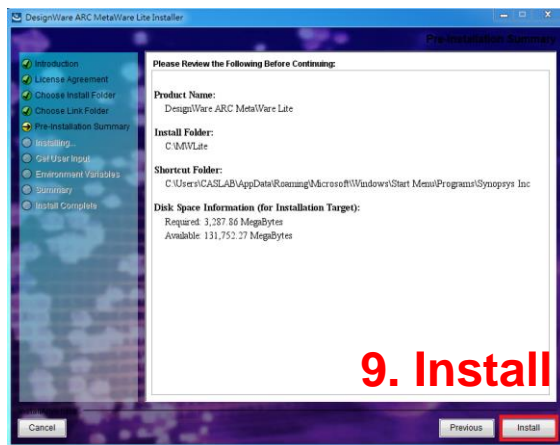
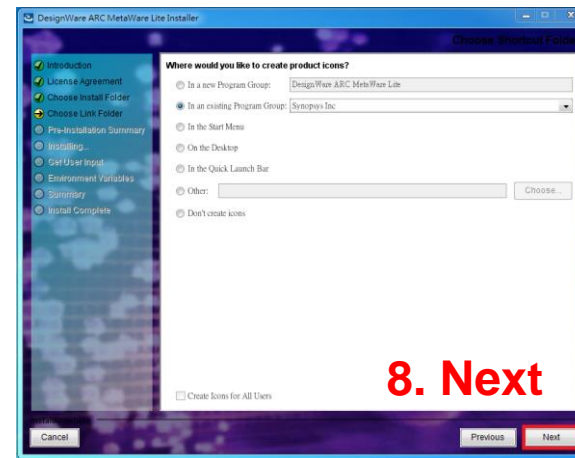
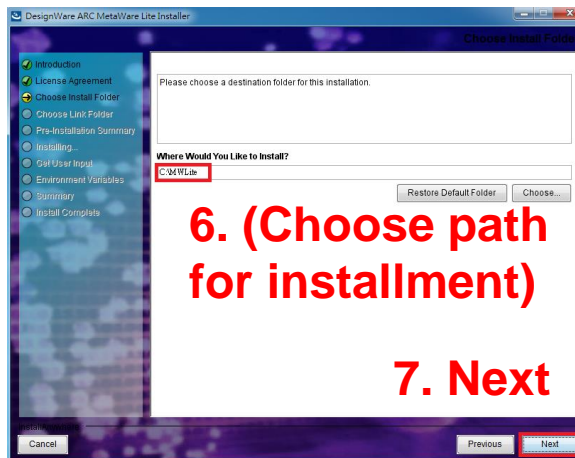
實驗環境 (7/10)

Step5. 在筆電上安裝 MetaWare_Lite



實驗環境 (8/10)

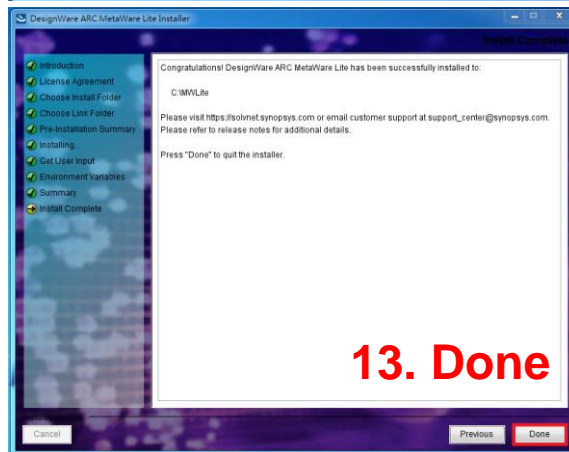
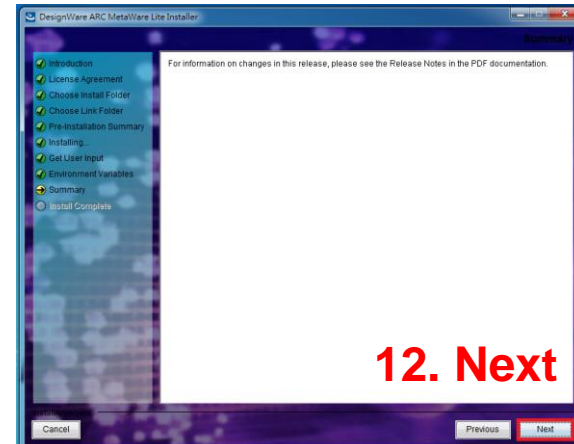
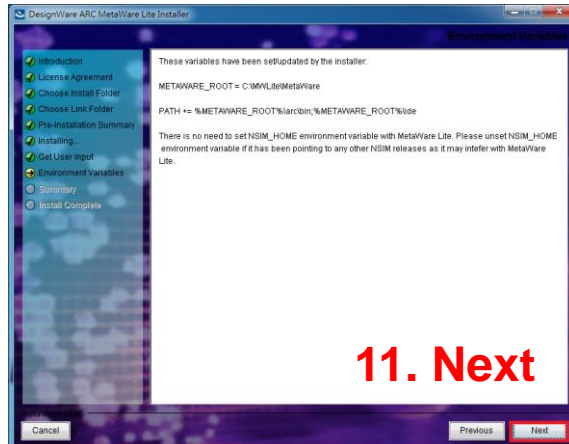
Step5. 在筆電上安裝 MetaWare_Lite



12. Next 24

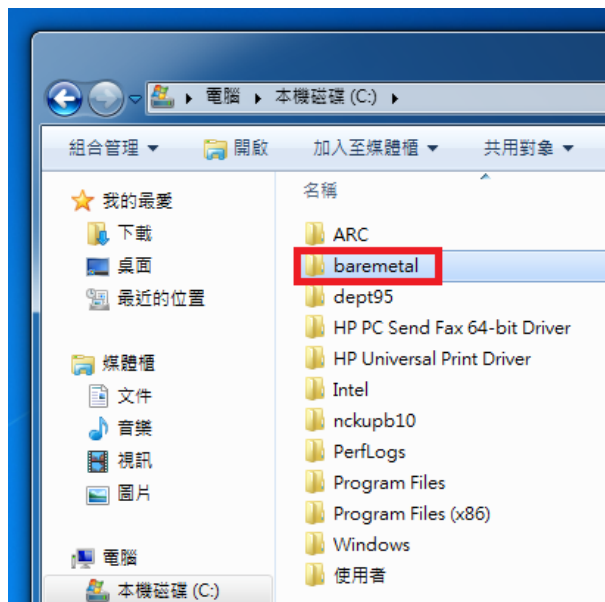
實驗環境 (9/10)

Step5. 在筆電上安裝 MetaWare_Lite



實驗環境 (10/10)

Step6. 將筆電上的專案檔(baremetal 資料夾)移到沒有中文名稱的路徑下。(這裡建議在C槽下)。

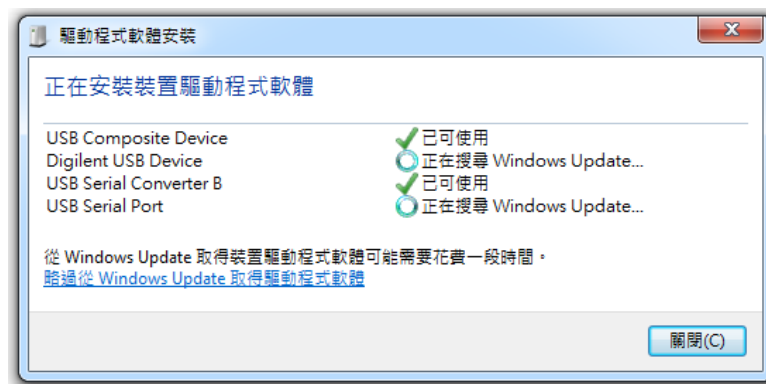


Lab6-實驗大綱與目的

1. ARC簡介
2. 實驗環境
 1. License申請
 2. Digilent Driver 安裝
 3. MIDE Lite 安裝
3. 實作一 (連接ARC板)
4. 實作二 (Build & Debug Project)

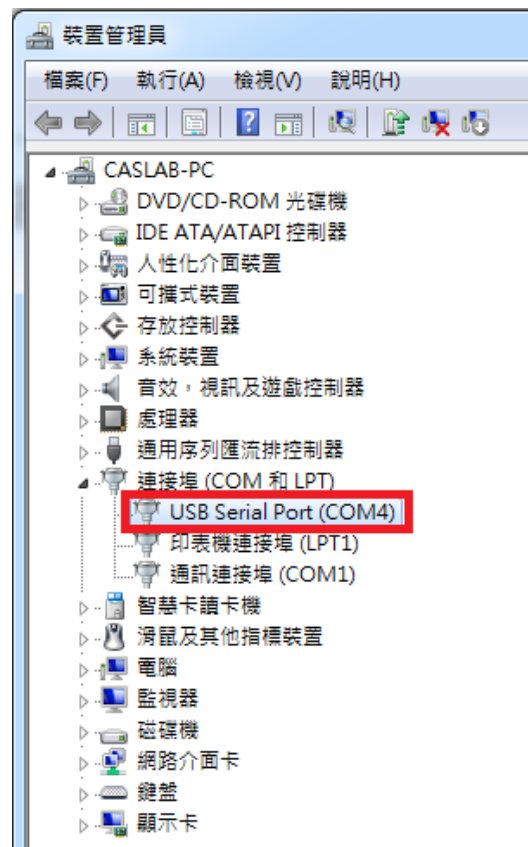
實作 一（連接ARC板）

Step1. 將ARC的J7（mini USB）接到筆電的USB插槽。此時，筆電會進行驅動程式軟體安裝更新。



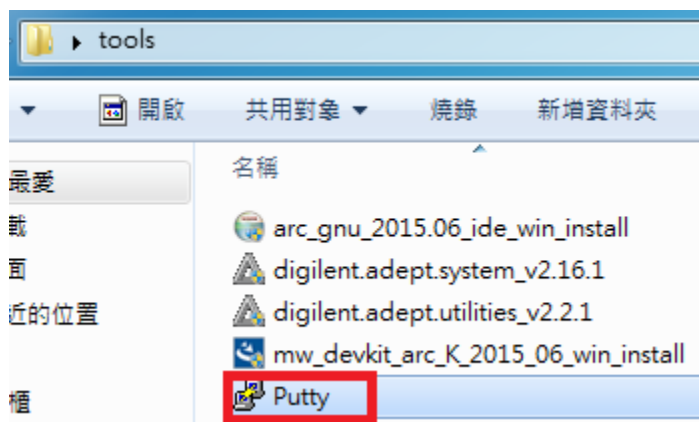
實作 一（連接ARC板）

Step2. 打開裝置管理員(控制台->硬體和音效->裝置管理員)，點開連接埠(COM和LPT)，查看USB Serial Port是COM幾。

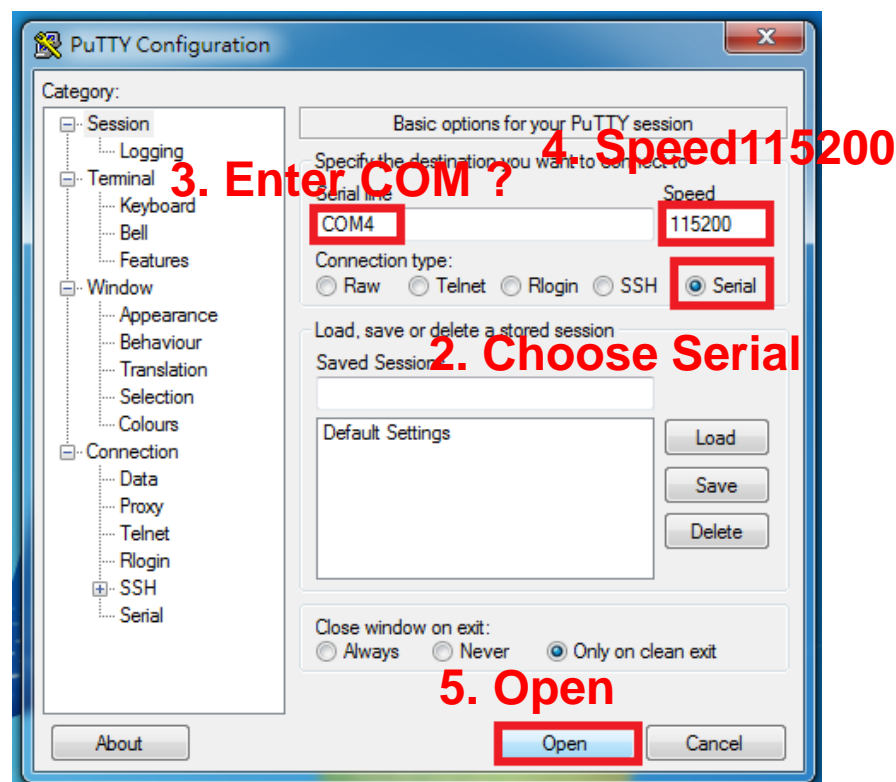


實作 一（連接ARC板）

Step3. 打開Putty，並設定Configuration。



1. Click Putty



3. Enter COM ? 4. Speed 115200

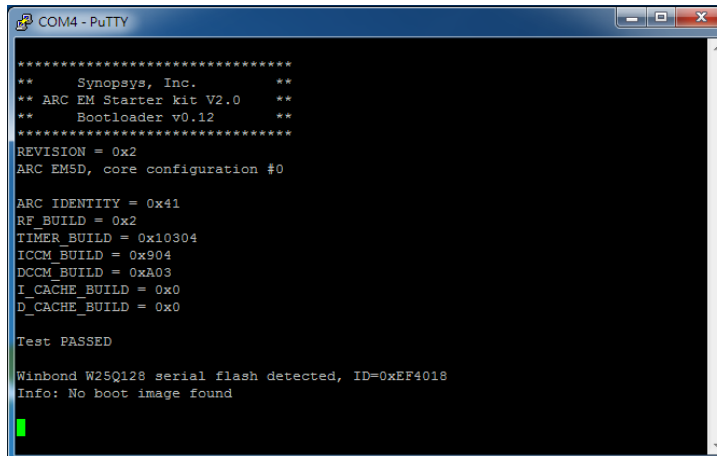
2. Choose Serial

5. Open

實作 一（連接ARC板）

Step4. 按下ARC板測試鈕，
此會將板子資訊輸出至Putty。

**2. Check the information in
Putty. :D**



```
COM4 - PuTTY
*****
**   Synopsys, Inc.   **
** ARC EM Starter kit V2.0 **
** Bootloader v0.12  **
*****
REVISION = 0x2
ARC EM5D, core configuration #0

ARC_IDENTITY = 0x41
RF_BUILD = 0x2
TIMER_BUILD = 0x10304
ICCM_BUILD = 0x904
DCCM_BUILD = 0xA03
I_CACHE_BUILD = 0x0
D_CACHE_BUILD = 0x0

Test PASSED

Winbond W25Q128 serial flash detected, ID=0xEF4018
Info: No boot image found
```

1. Click The 'R' or 'C' button

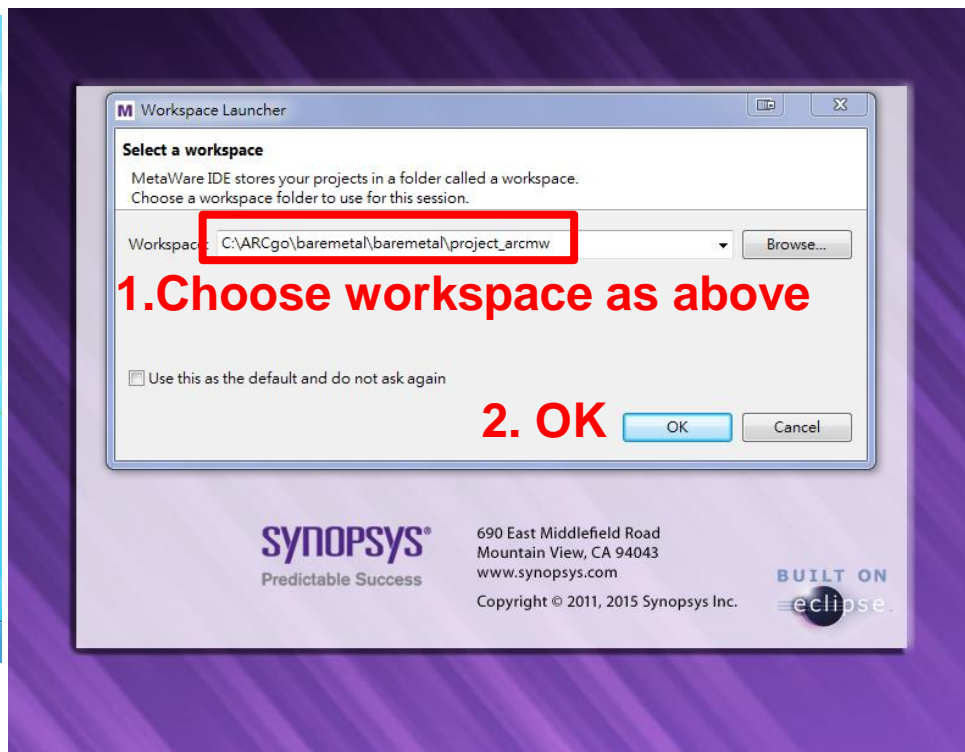


Lab6-實驗大綱與目的

1. ARC簡介
2. 實驗環境
 1. License申請
 2. Digilent Driver 安裝
 3. MIDE Lite 安裝
3. 實作一(連接ARC板)
4. 實作二(Build & Debug Project)

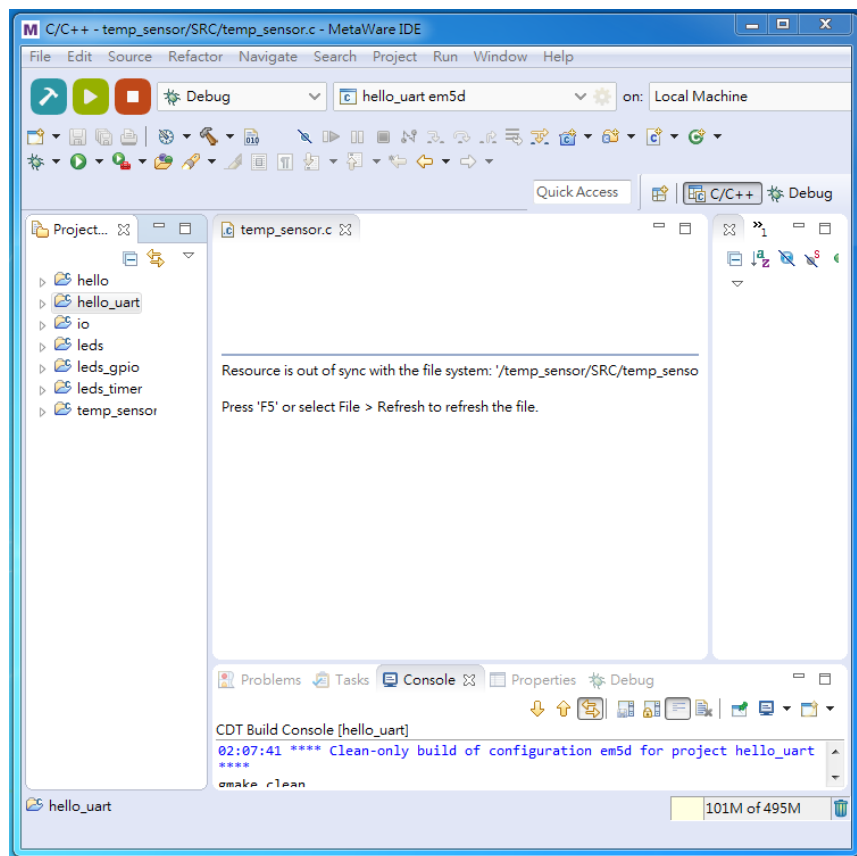
實作 二 (Build & Debug Project)

Step1. 開啟專案，選擇workspace在專案資料夾
(專案路徑/baremetal/baremetal/project_arcmw/)



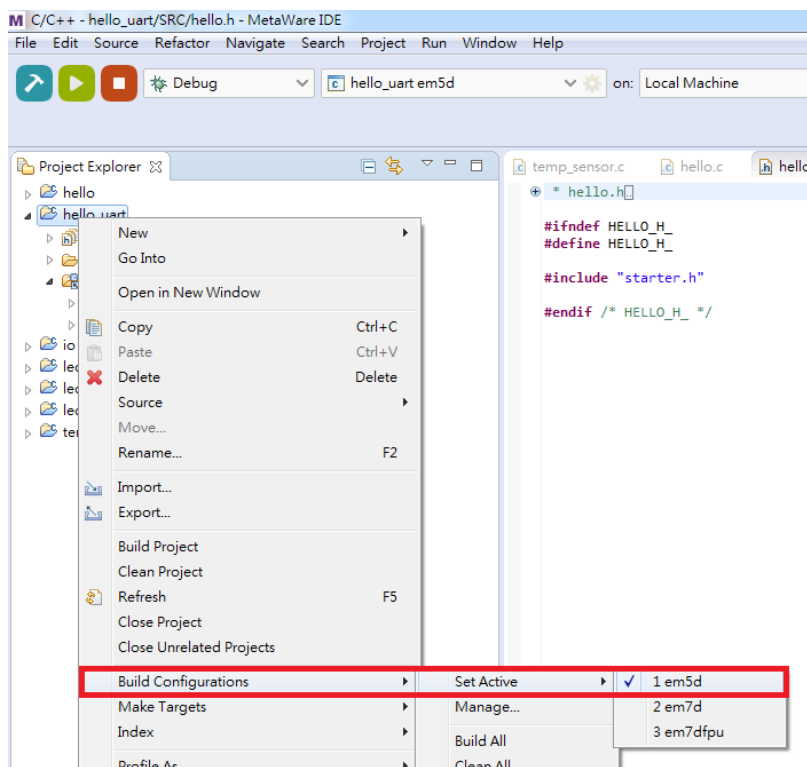
實作 二 (Build & Debug Project)

Step2. 以下是Metaware IDE 的介面



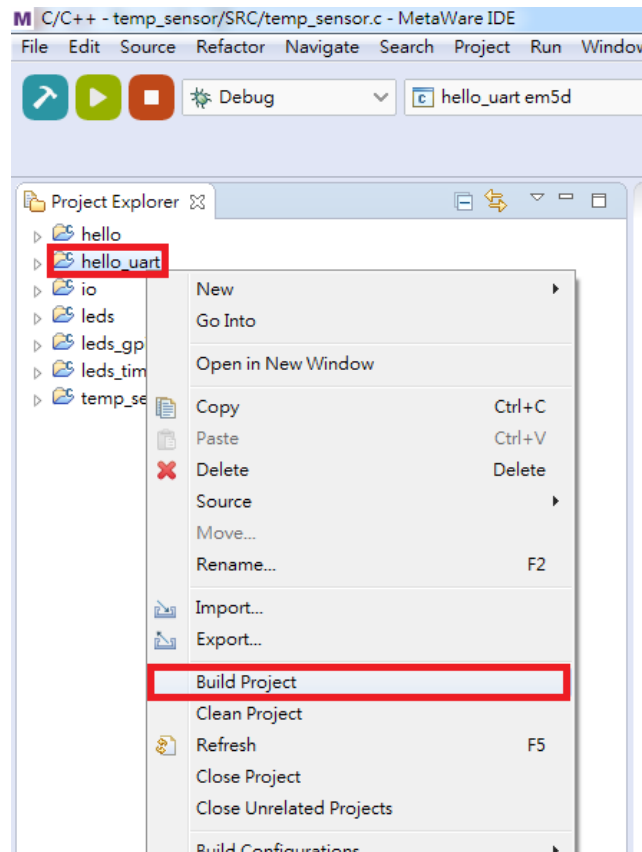
實作 二 (Build & Debug Project)

Step3. 查看Build Configuration 是否為現在連接的CPU型號(若ARC板上的SW1皆為OFF，則為em5d)



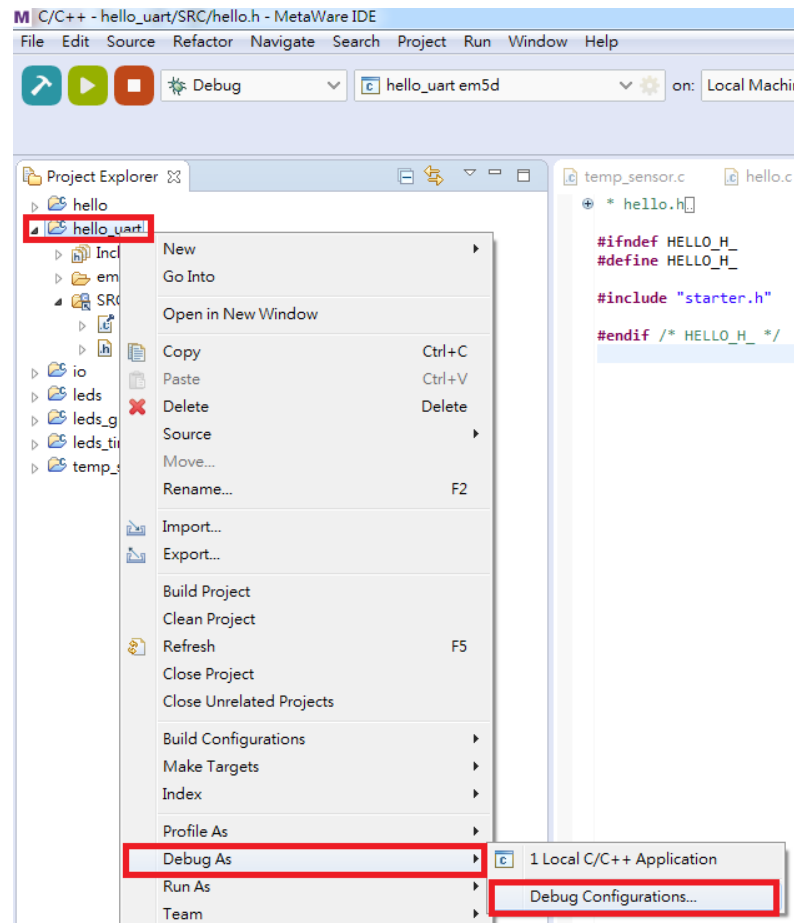
實作 二 (Build & Debug Project)

Step4. Build project (hello uart)



實作 二 (Build & Debug Project)

Step5. Debug (hello uart)



實作 二 (Build & Debug Project)

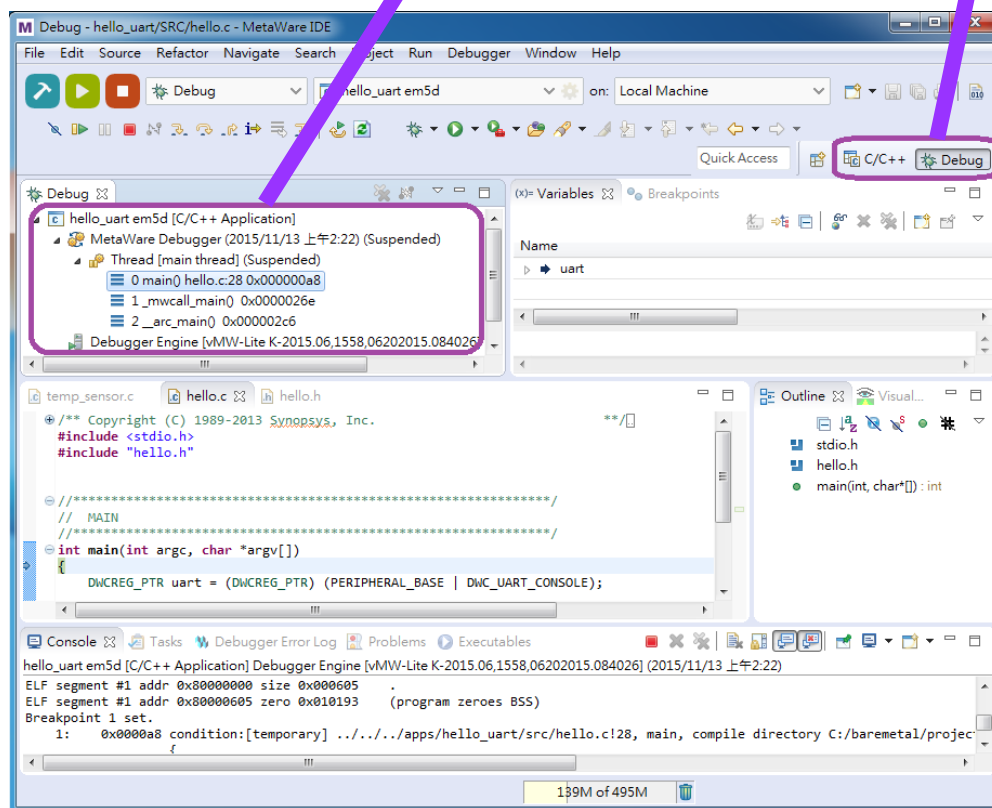
Step5. Debug (hello uart)

The screenshot shows the 'Debug Configurations' window in the MetaWare IDE. The left pane lists project configurations, with 'hello_uart em5d' selected. The right pane shows the configuration for this project, with the 'Debugger' tab active. Red boxes and numbers highlight the following steps:

- 1. Choose built project**: Points to the 'hello_uart em5d' project in the left pane.
- 2. Debugger**: Points to the 'Debugger' tab in the right pane.
- 3. Target Selection**: Points to the 'Target Selection' sub-tab in the 'Debugger Options' section.
- 4. Hardware**: Points to the 'Hardware' dropdown menu in the 'Select the target' section.
- 5. Digilent JTAG cable**: Points to the 'Digilent JTAG cable' dropdown menu in the 'Choose hardware connection type' section.
- 6. 4-wire (standard)**: Points to the '4-wire (standard)' dropdown menu in the 'Choose whether to use 4 or 2-wire communications' section.
- 7. Debug**: Points to the 'Debug' button at the bottom right of the window.

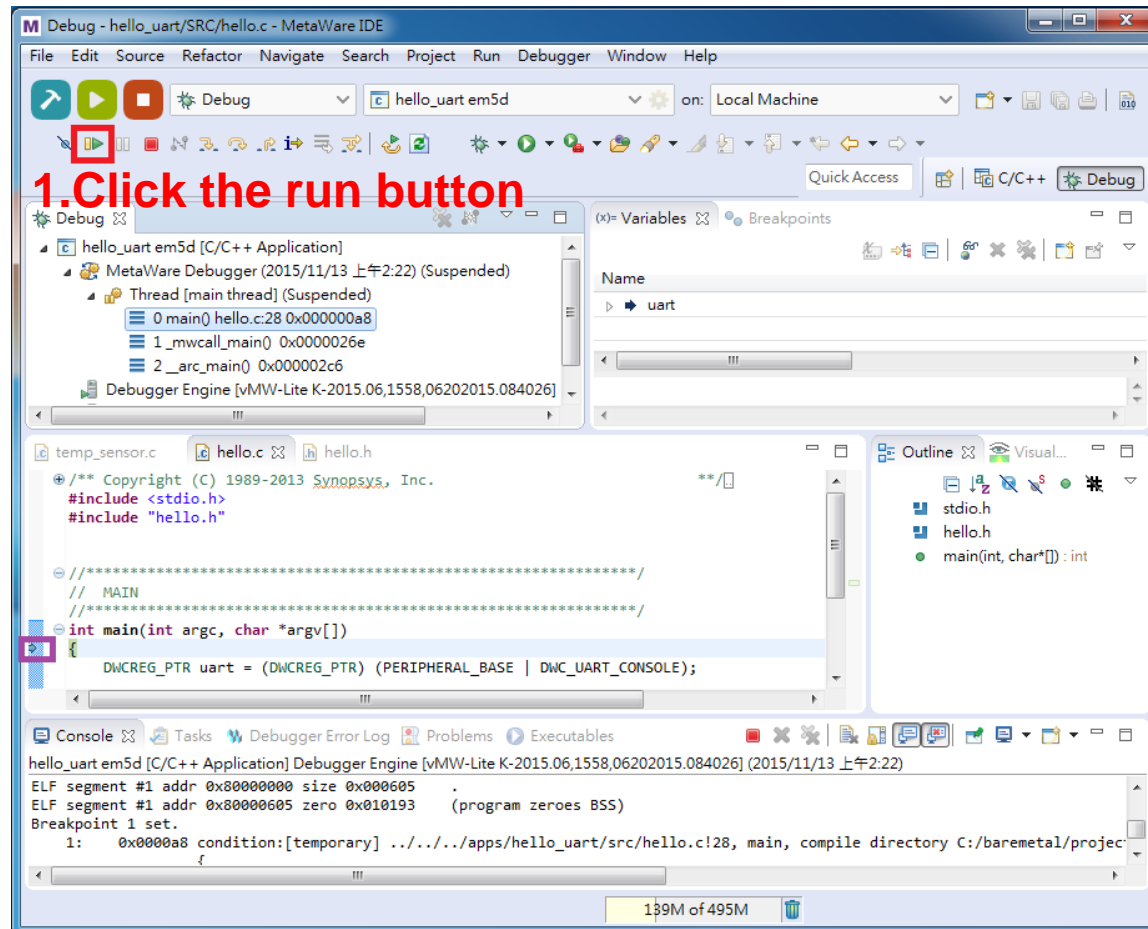
實作 二 (Build & Debug Project)

如果連接Target(ARC板)成功後，會自動進入Debug Mode，且在Debug視窗可以看到一個啟動的Project。



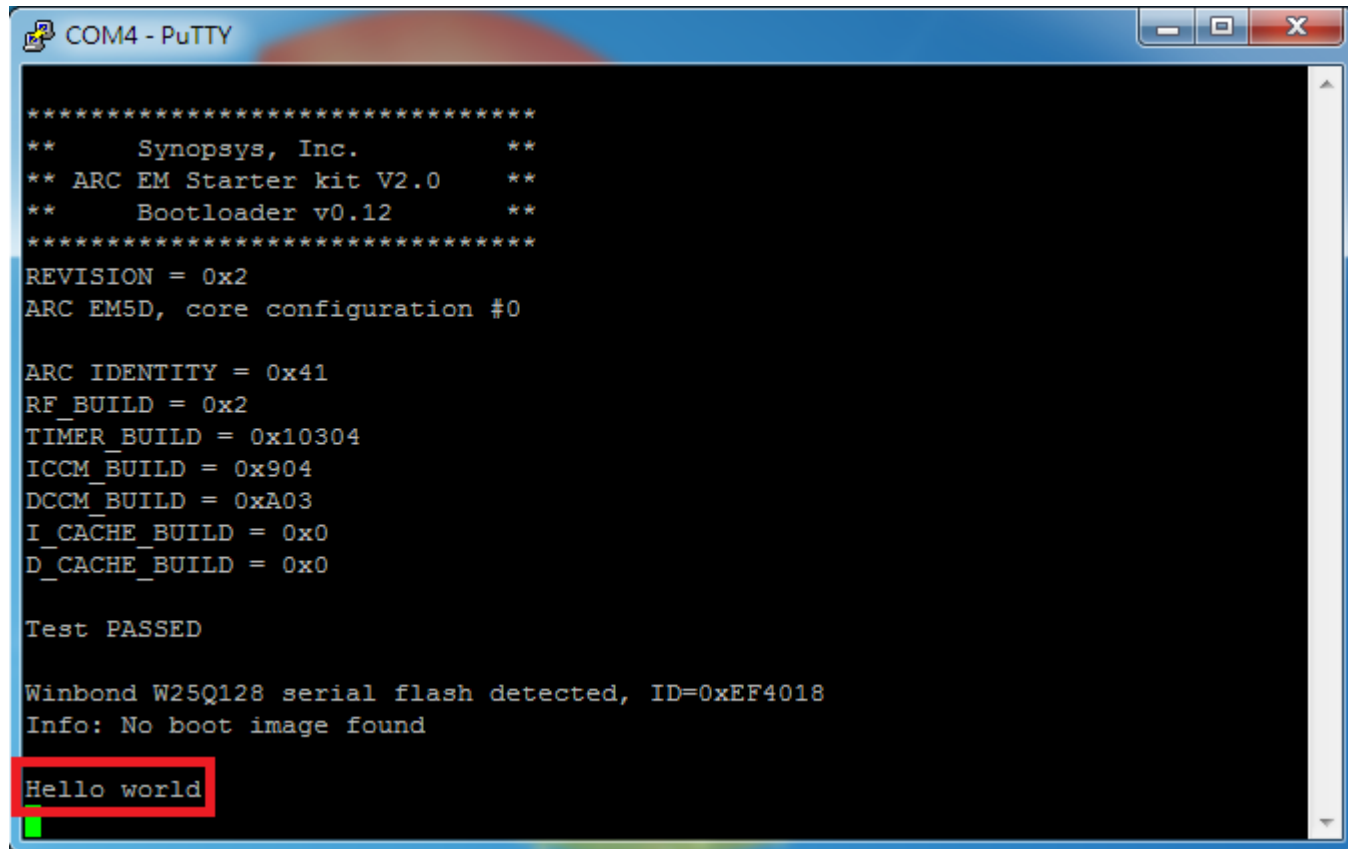
實作 二 (Build & Debug Project)

Step6. Run the program (hello uart)



實作 二 (Build & Debug Project)

Step7. Verify your Project



```
COM4 - PuTTY

*****
**      Synopsys, Inc.      **
** ARC EM Starter kit V2.0  **
**      Bootloader v0.12   **
*****
REVISION = 0x2
ARC EM5D, core configuration #0

ARC IDENTITY = 0x41
RF_BUILD = 0x2
TIMER_BUILD = 0x10304
ICCM_BUILD = 0x904
DCCM_BUILD = 0xA03
I_CACHE_BUILD = 0x0
D_CACHE_BUILD = 0x0

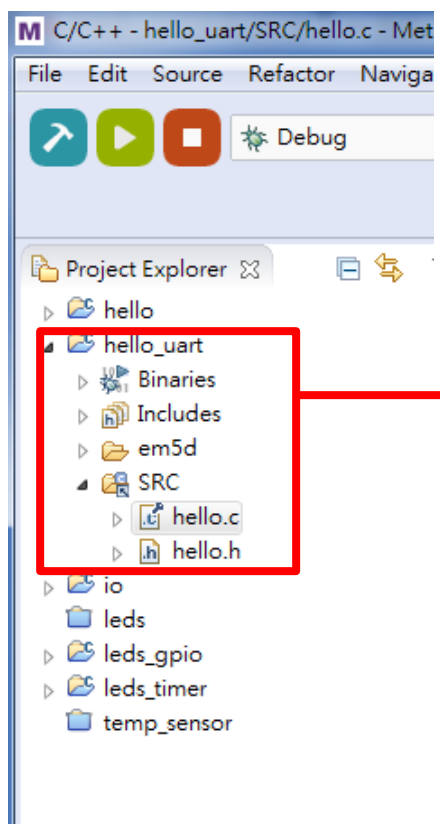
Test PASSED

Winbond W25Q128 serial flash detected, ID=0xEF4018
Info: No boot image found

Hello world
```

實作 二 (Build & Debug Project)

當然，不能只看到板子能成功通過uart和電腦溝通就結束了，我們需要再深入一點點了解專案的內容。



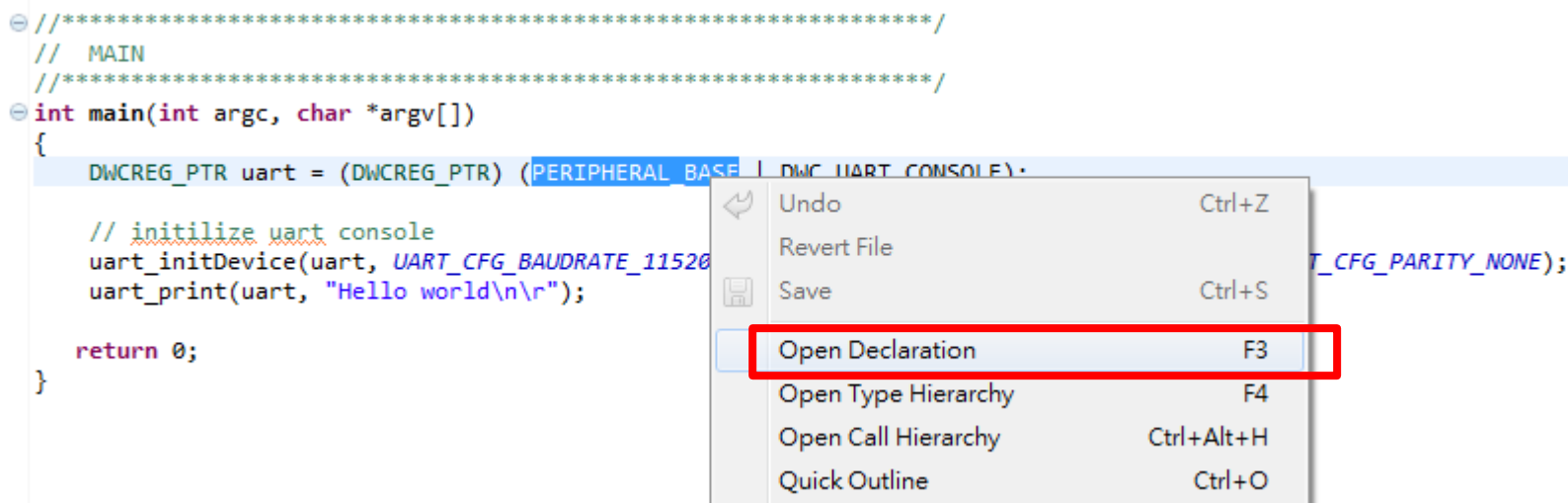
點開專案(hello_uart)
然後點開source code(SRC)
點開hello.c，看看主程式內容。

實作 二 (Build & Debug Project)

在看code之前，先提示大家一個trace code的小技巧:

使用IDE的好處其中之一，可以選取變數、方法等等，按下F3可以得到程式碼宣告或者實現的地方。非常快就可以trace出不懂的部分。

Ex:



實作 二 (Build & Debug Project)

這個statement是建立uart在memory map中的位址。請想想，這個位址的數值是什麼呢？

```
hello.c
+ /** Copyright (C) 1989-2013 Synopsys, Inc.
#include <stdio.h>
#include "hello.h"

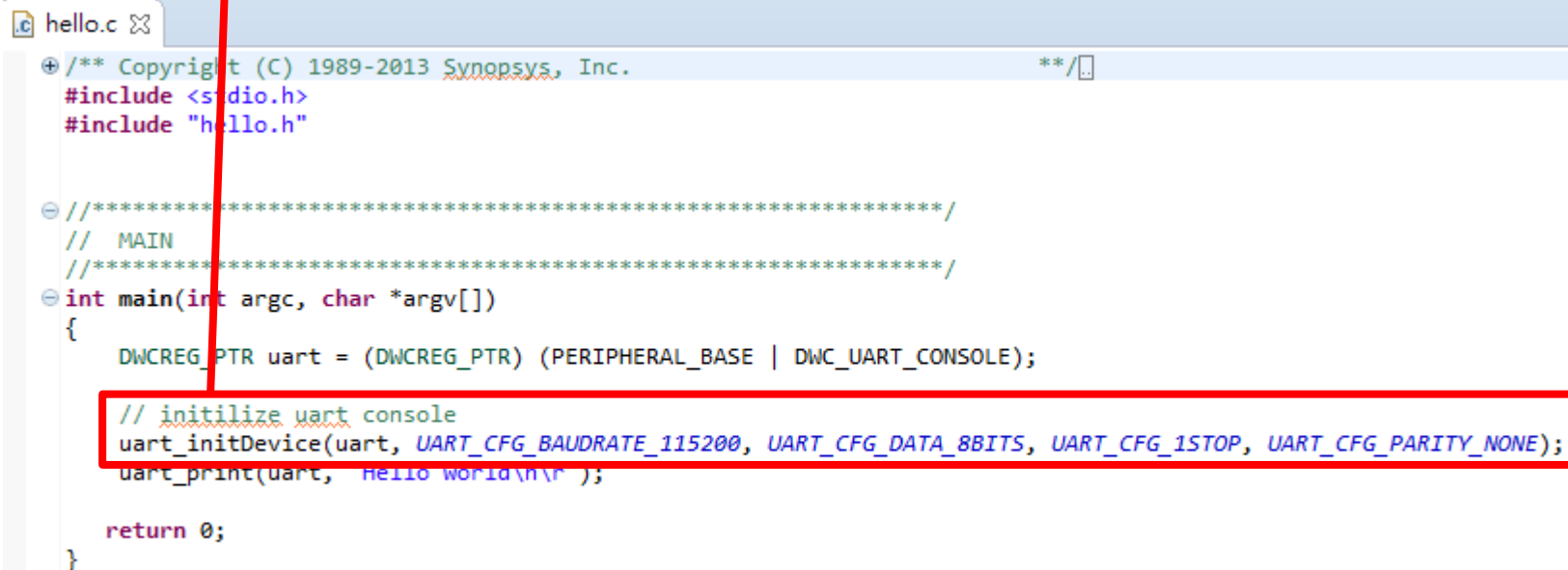
- /*******
// MAIN
- /*******
int main(int argc, char *argv[])
{
    DWCREG_PTR uart = (DWCREG_PTR) (PERIPHERAL_BASE | DWC_UART_CONSOLE);

    // initialize uart console
    uart_initDevice(uart, UART_CFG_BAUDRATE_115200, UART_CFG_DATA_8BITS, UART_CFG_1STOP, UART_CFG_PARITY_NONE);
    uart_print(uart, "Hello world\n\r");

    return 0;
}
```

實作 二 (Build & Debug Project)

這個部分是設定uart的傳輸方式 (Baud: 115200, 8N1)，如果有興趣可以去trace code，並自行學習uart，這裡提供一個網站連結。
(<http://www.lammertbies.nl/comm/info/serial-uart.html#THR>)



```
hello.c
+ /** Copyright (C) 1989-2013 Synopsys, Inc. */
#include <stdio.h>
#include "hello.h"

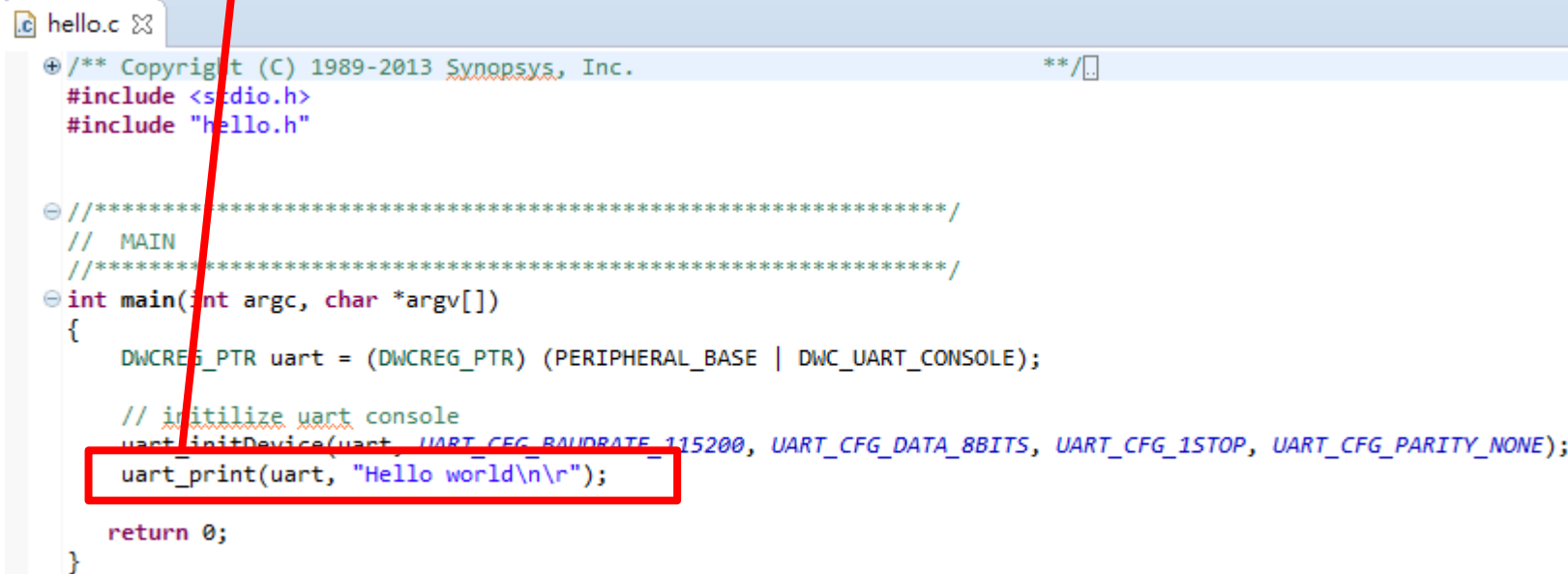
//*****
// MAIN
//*****
int main(int argc, char *argv[])
{
    DWCREG_PTR uart = (DWCREG_PTR) (PERIPHERAL_BASE | DWC_UART_CONSOLE);

    // initialize uart console
    uart_initDevice(uart, UART_CFG_BAUDRATE_115200, UART_CFG_DATA_8BITS, UART_CFG_1STOP, UART_CFG_PARITY_NONE);
    uart_print(uart, "Hello world\n");

    return 0;
}
```

實作 二 (Build & Debug Project)

透過uart是如何print出字元，可以trace這部分的code，和自行學習。



```
hello.c
+ /** Copyright (C) 1989-2013 Synopsys, Inc. **/
#include <stdio.h>
#include "hello.h"

//*****
// MAIN
//*****
int main(int argc, char *argv[])
{
    DWCREG_PTR uart = (DWCREG_PTR) (PERIPHERAL_BASE | DWC_UART_CONSOLE);

    // initialize uart console
    uart_initDevice(uart, UART_CFG_BAUDRATE_115200, UART_CFG_DATA_8BITS, UART_CFG_1STOP, UART_CFG_PARITY_NONE);
    uart_print(uart, "Hello world\\n\\r");

    return 0;
}
```

實驗結報

⊕ 結報格式

- **這次不需要繳交結報**，但是請各位可以開始想想期末Project的題目和方向。P. S. 助教可提供的模組包括溫度感測、LCD模組。

⊕ TA Contact Information:

- 助教信箱：a2215689@gmail.com
- Rm 92617
- Office hour：(Monday)14:00pm~16:00pm