

Tutorial 2 – WE-I Project Environment Setup & Development Flow

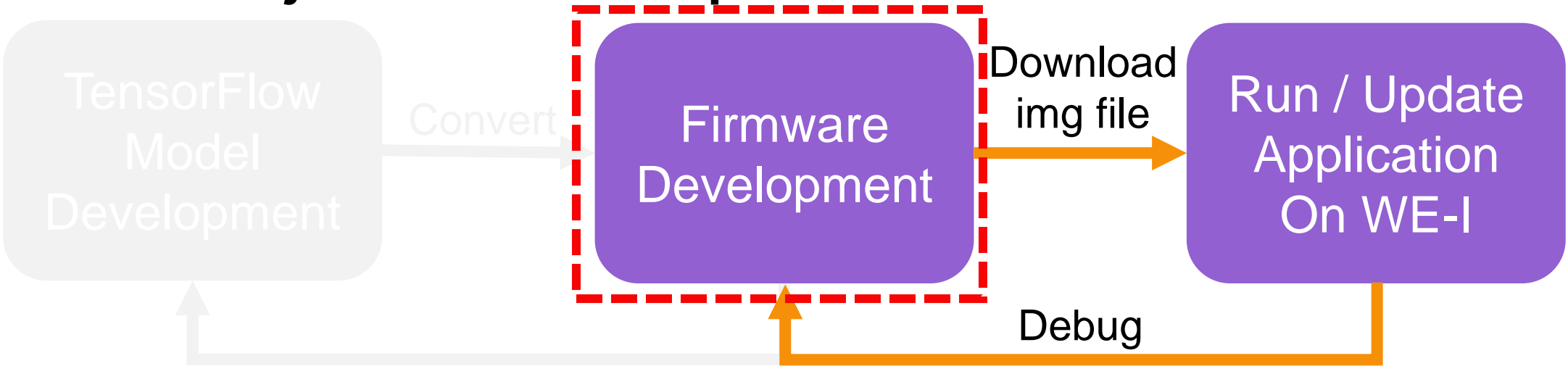


WE-I Project Development Flow



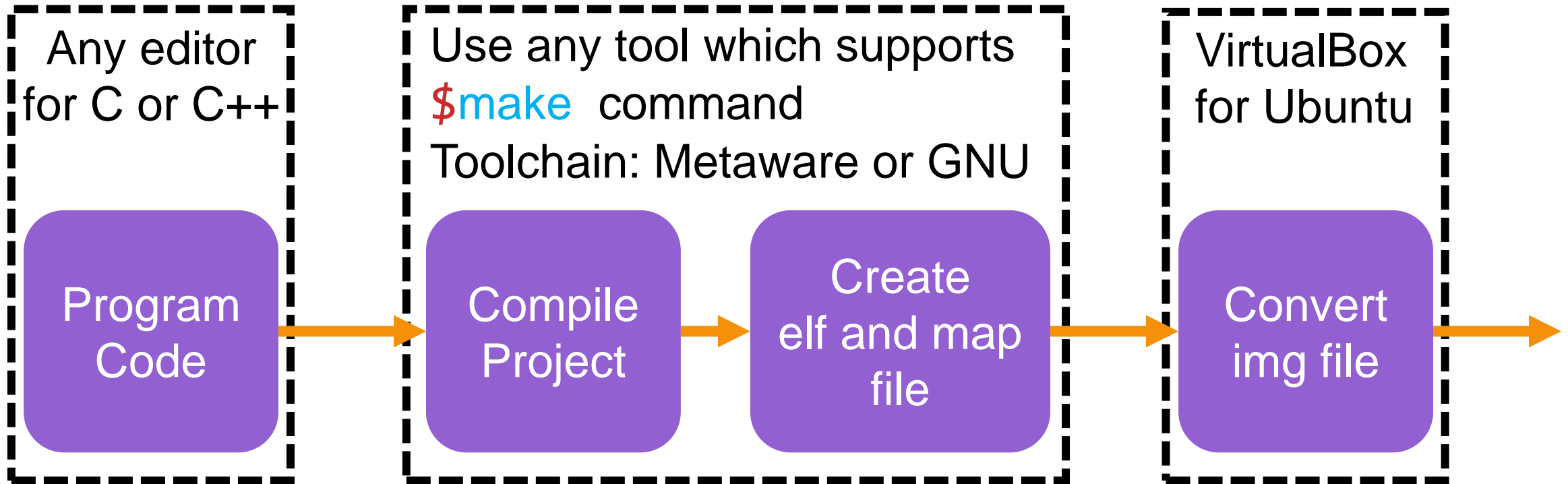
Stage	TensorFlow Model Development	Firmware Development	Run / Update Application On WE-I
Tool	Anaconda Cygwin	Cygwin Metaware or ARC GNU VirtualBox (Ubuntu 20.04)	Tera Term USB Micro
Language	Python 3	C language C++ language	

WE-I Project Development Flow



Stage	TensorFlow Model Development	Firmware Development	Run / Update Application On WE-I
Tool	Anaconda Cygwin	Cygwin Metaware or ARC GNU VirtualBox (Ubuntu 20.04)	Tera Term USB Micro
Language	Python 3	C language C++ language	

Firmware Development



Download and Setup SDK



Download and Setup SDK

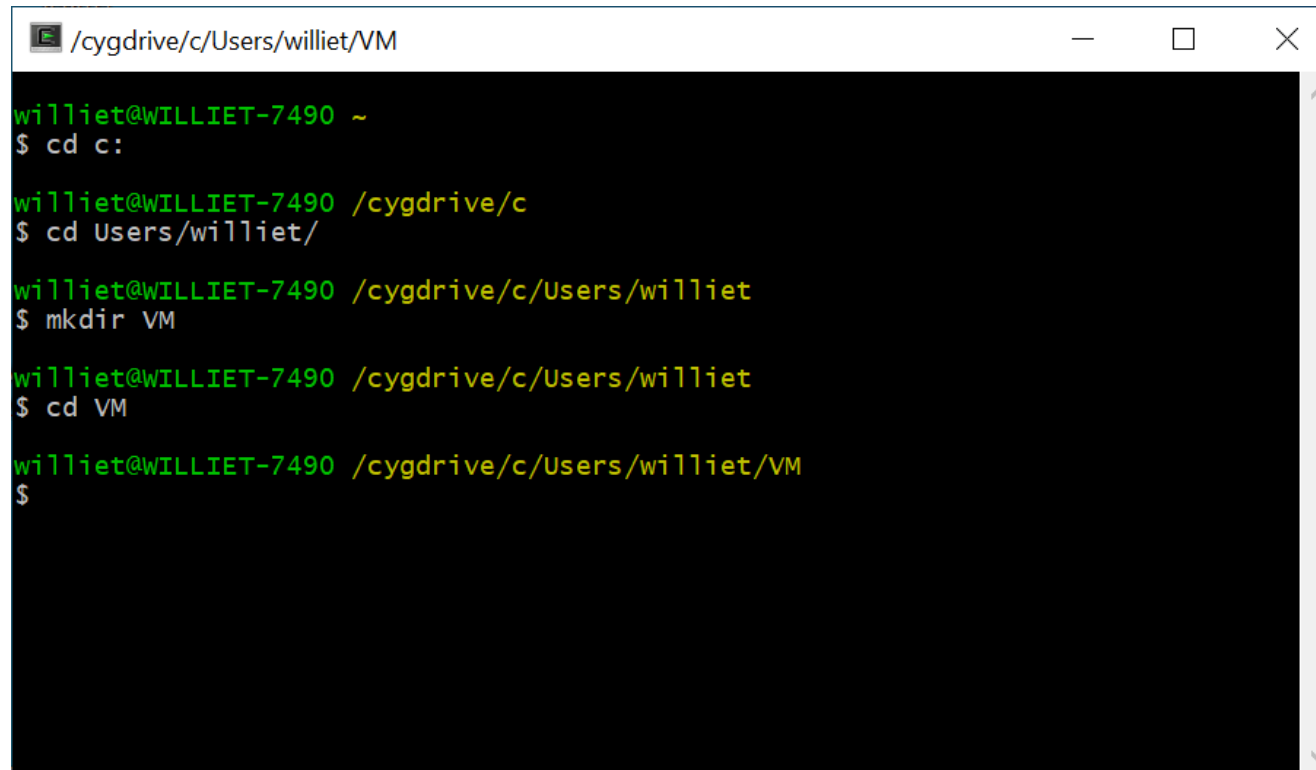
1. Open Cygwin64 Terminal

\$ cd c:

\$ cd Users/{username}/ (to your working file path)

\$ mkdir VM (Suggest create a new folder named “VM”)

\$ cd VM



```
/cygdrive/c/Users/williet/VM

williet@WILLIET-7490 ~
$ cd c:

williet@WILLIET-7490 /cygdrive/c
$ cd Users/williet/

williet@WILLIET-7490 /cygdrive/c/Users/williet
$ mkdir VM

williet@WILLIET-7490 /cygdrive/c/Users/williet
$ cd VM

williet@WILLIET-7490 /cygdrive/c/Users/williet/VM
$
```

Download and Setup SDK

Commands in cygwin64 terminal

2. Download SDK from Synopsys Github

```
$ git clone https://github.com/foss-for-synopsys-dwc-arc-processors/arc\_contest.git
```

```
$ cd arc_contest
```

```
$ git submodule init
```

```
$ git submodule update
```

```
$ cd himax_tflm
```

```
$ make download
```

Download and Setup SDK

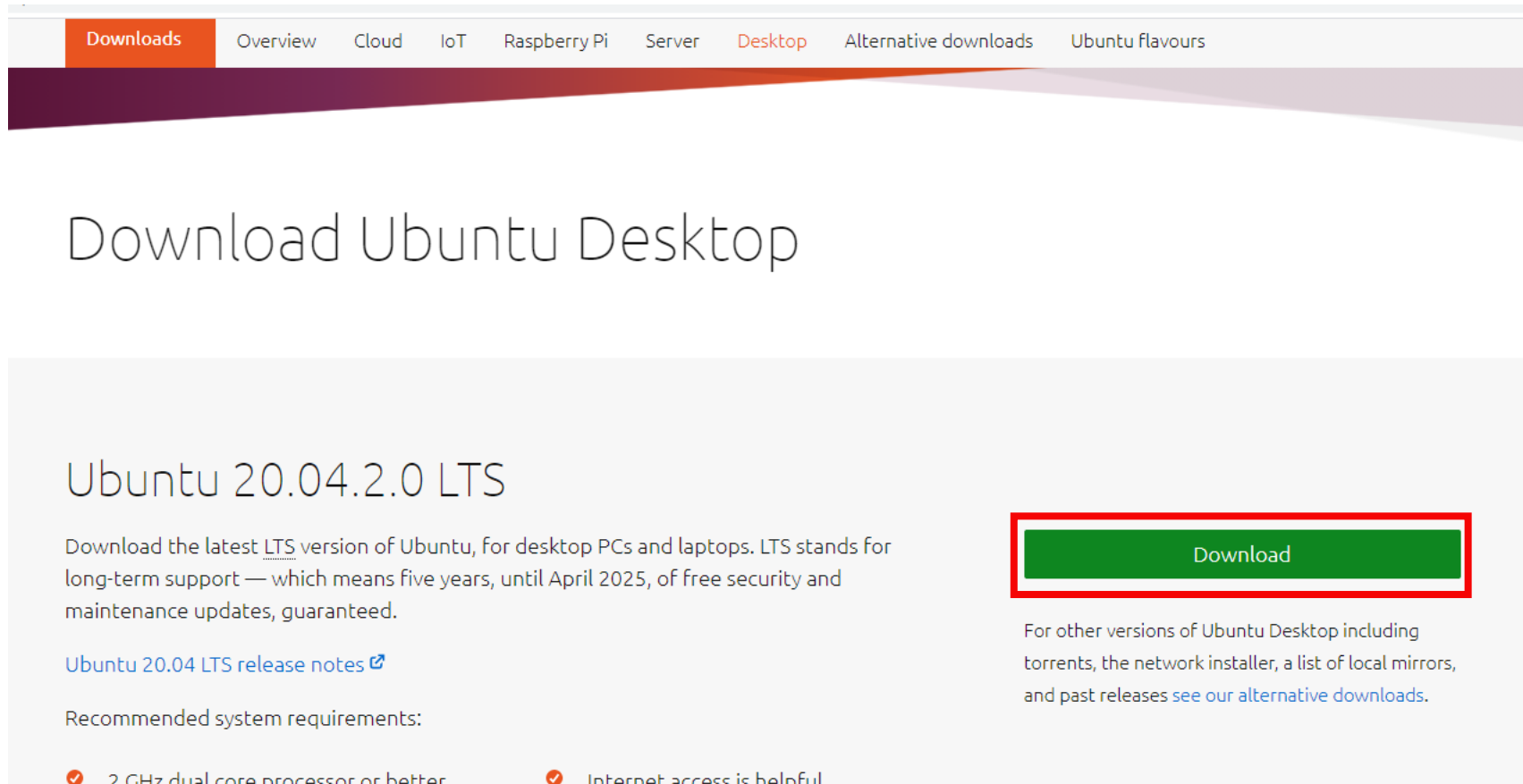
After these steps, your file structure will be like:

```
arc_contest
|
---- bsp_tflu
---- doc_tutorial
---- himax_tflm
    |
    ---- himax_we1_sdk
    ---- image_gen_linux
    ---- tensorflow
    ---- third_party
---- Synopsys_SDK
    |
    ---- Example_Project
    ---- User_Project
```


Download and Setup SDK

3. 下載Ubuntu映像檔

<https://ubuntu.com/download/desktop>



The screenshot shows the Ubuntu website's 'Downloads' section, specifically the 'Desktop' tab. The page title is 'Download Ubuntu Desktop'. It features a large green 'Download' button, which is highlighted with a red rectangular border. Below the button, there is a link to 'Ubuntu 20.04 LTS release notes' and a section for 'Recommended system requirements' with two items: '2 GHz dual core processor or better' and 'Internet access is helpful', both marked with red checkmarks.

Downloads Overview Cloud IoT Raspberry Pi Server Desktop Alternative downloads Ubuntu flavours

Download Ubuntu Desktop

Ubuntu 20.04.2.0 LTS

Download the latest LTS version of Ubuntu, for desktop PCs and laptops. LTS stands for long-term support — which means five years, until April 2025, of free security and maintenance updates, guaranteed.

[Ubuntu 20.04 LTS release notes](#)

Recommended system requirements:

- ✓ 2 GHz dual core processor or better
- ✓ Internet access is helpful

[Download](#)

For other versions of Ubuntu Desktop including torrents, the network installer, a list of local mirrors, and past releases [see our alternative downloads](#).

Download and Setup SDK

4. Open VM Oracle VirtualBox

5. 新增(N) > 選擇類型Linux、版本Ubuntu (64-bit)

? ×



← 建立虛擬機器

名稱和作業系統

請為新的虛擬機器選擇描述性名稱和目的地資料夾，並選取要在其上安裝的作業系統類型。您選擇的名稱將在整個 VirtualBox 中使用，以標識這部電腦。

名稱: VM_Ubuntu

機器資料夾: C:\Users\williet\VirtualBox VMs

類型(T): Linux

版本(V): Ubuntu (64-bit)

專家模式(E)

下一個(N)

取消

Download and Setup SDK

6. 記憶體大小請依照個人開發需求設定

← 建立虛擬機器

記憶體大小

選取配置到虛擬機器的記憶體量 (RAM)，單位 MB。

建議的記憶體大小為 **1024MB**。

4 MB 16384 MB

1024 MB

下一個(N) 取消

Download and Setup SDK

7. 硬碟設定請依照個人開發需求設定，建議至少開20GB。

?

×

← 建立虛擬機器

← 建立虛擬硬碟

硬碟

如果您希望能加入虛擬硬碟到新的機器。可以建立新的硬碟檔或從清單選取一個或使用資料夾圖示選取另一個位置。

如果需要更多複雜存放裝置設定，可以略過此步驟，並在機器建立時進行變更機器設定。

建議硬碟的大小為 **10.00 GB**。

☐ 不加入虛擬硬碟(D)

☒ 立即建立虛擬硬碟(C)

☐ 使用現有虛擬硬碟檔案(U)

UB.vdi (標準, 10.00 GB)

建立

取消

硬碟檔類型

請選擇新的虛擬硬碟希望使用的檔案類型。 如果不需要用在其它虛擬化軟體，您可以保留這個設定不變更。

☒ VDI (VirtualBox 磁碟映像)

☐ VHD (虛擬硬碟)

☐ VMDK (虛擬機器磁碟)

專家模式(E)

下一個(N)

取消

Download and Setup SDK

7. 硬碟設定請依照個人開發需求設定，建議至少開20GB。

← 建立虛擬硬碟

存放裝置在實體硬碟

請選擇新的虛擬硬碟檔是否應根據使用來成長（動態分配），或建立為其最大大小（固定大小）。

動態分配的硬碟檔將只在填滿時使用實體硬碟的空間（直到最大的**固定大小**），儘管它的空間釋放時不會再次自動縮小。

固定大小硬碟檔在某些系統需要比較長的時間建立，但通常用起來比較快。

☒ 動態分配(D)

☐ 固定大小(E)

下一個(N)

取消

← 建立虛擬硬碟

檔案位置和大小

請在以下的方塊中輸入新虛擬硬碟檔的名稱，或按一下資料夾圖示以選擇建立檔案的其它資料夾。

C:\Users\williet\VirtualBox VMs\VM_Ubuntu\VM_Ubuntu.vdi



選擇虛擬硬碟的大小（以 MB 位元組為單位）。這個大小是對虛擬機器將能夠存儲在硬碟上的檔案資料量的限制。



建立

取消

Download and Setup SDK

8. 啟動新增的虛擬機，並且設定映像檔(步驟3所抓的檔案)
(第一次開機需要設定使用者等資訊，請依照各自需求設定，如步驟9所示)



← 選取啟動磁碟

請選取虛擬光碟檔或包含磁碟的實體光碟機，以便從其啟動新的虛擬機器。

該磁碟應適合從電腦啟動，並且包含您希望在虛擬機器上安裝的作業系統 (如果您現在要如此做)。下次關閉虛擬機器時，磁碟將自動從虛擬磁碟機中退出，但如果需要，也可以使用「裝置」功能表自行執行此操作。

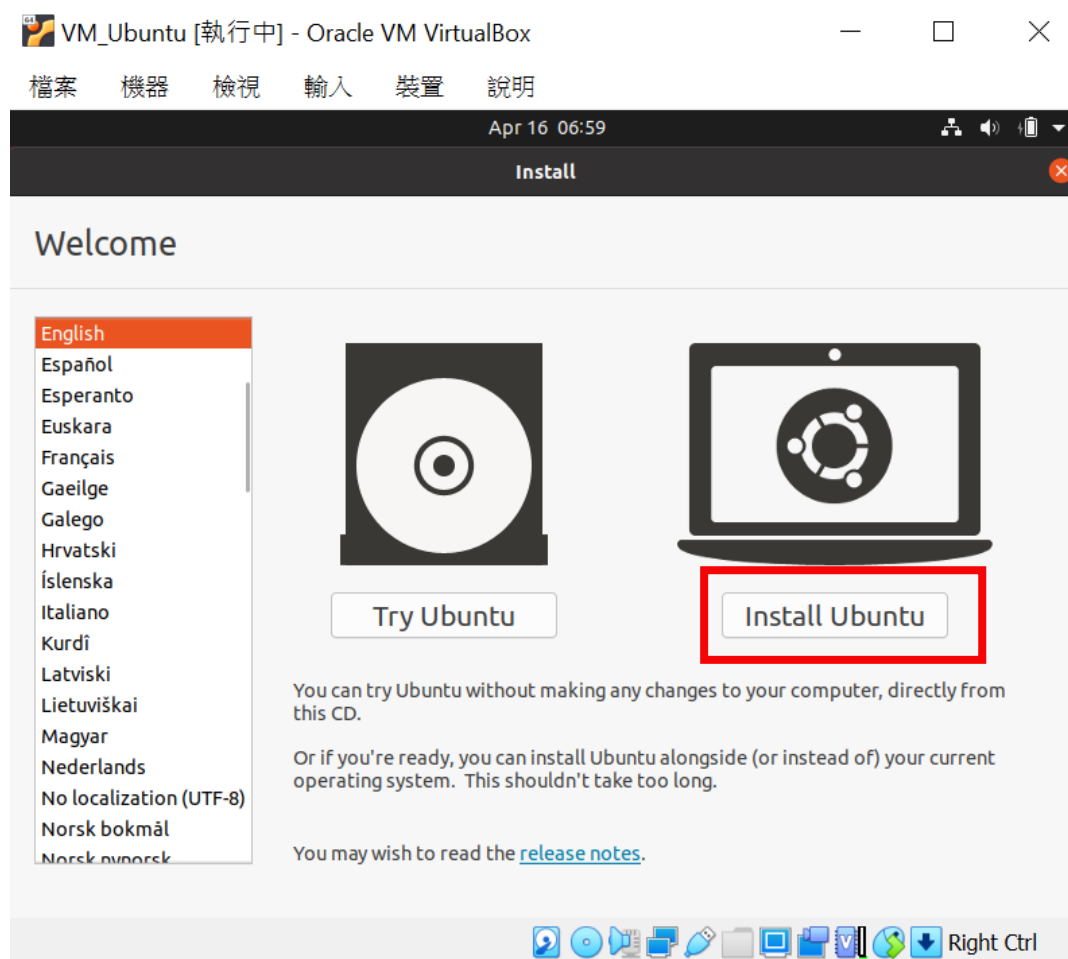
ubuntu-20.04.2.0-desktop-amd64.iso (2.68 GB)

開始

取消

Download and Setup SDK

9. 選擇安裝，並依照使用者需求自行更改設定(如時區或者鍵盤輸入法)，其餘預設即可，接著等待安裝完畢



Download and Setup SDK

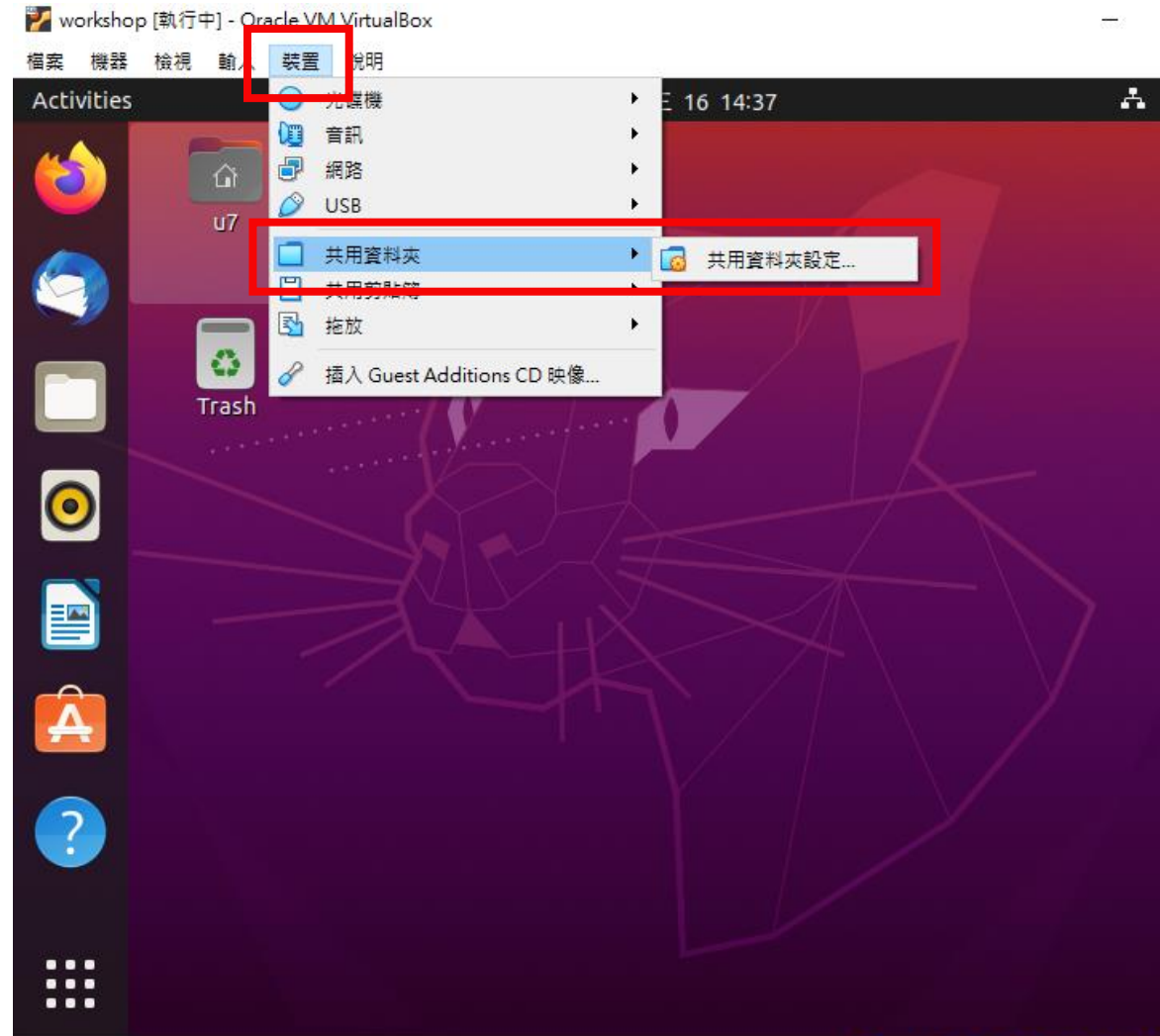
10. 開啟Ubuntu，開始設定共用資料夾。

在開發的過程中，因為編譯是在Windows介面
轉換燒錄用的Image是在Ubuntu

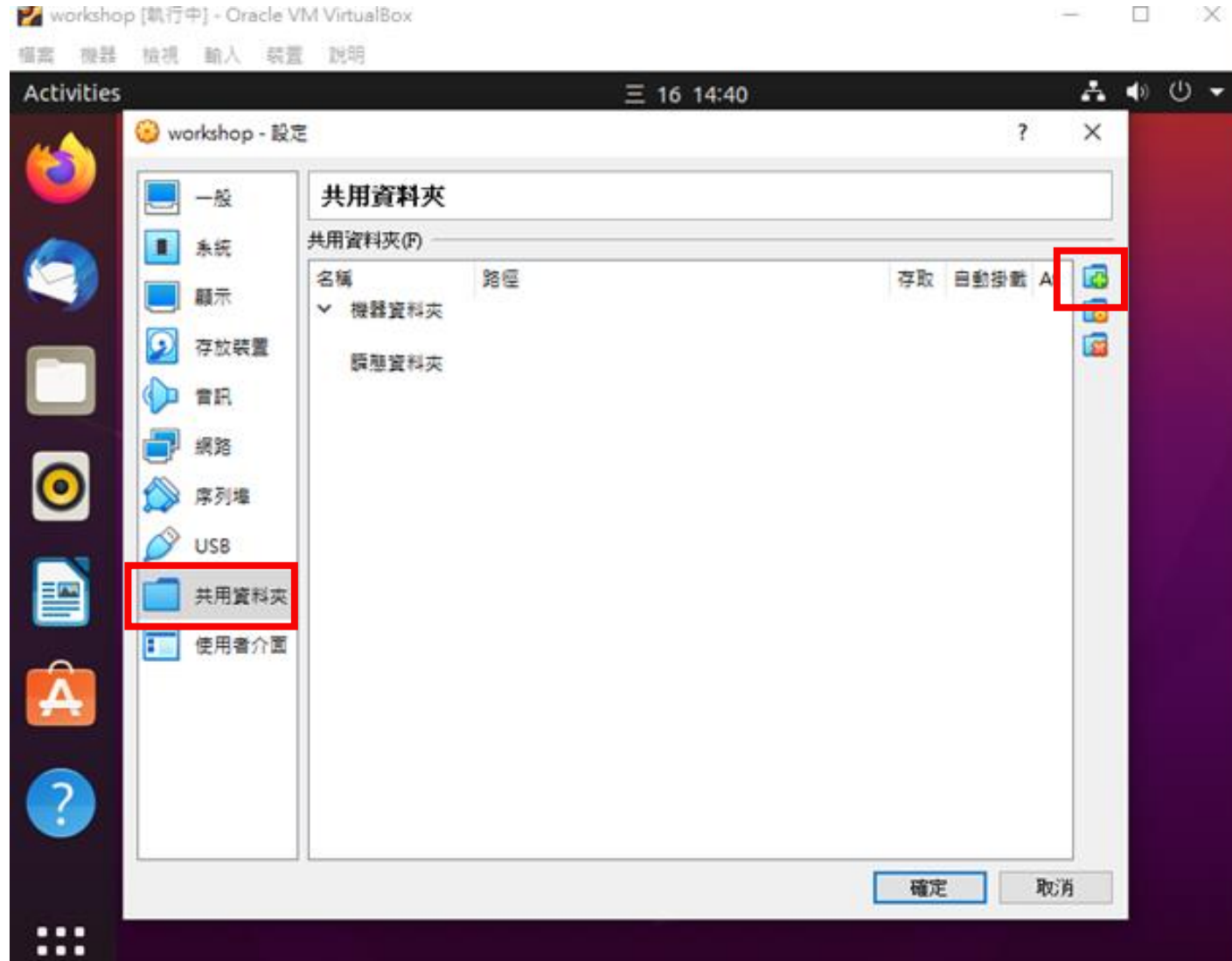
因此要設定共用資料夾，使兩個開發環境可以看到共同的檔案

Download and Setup SDK

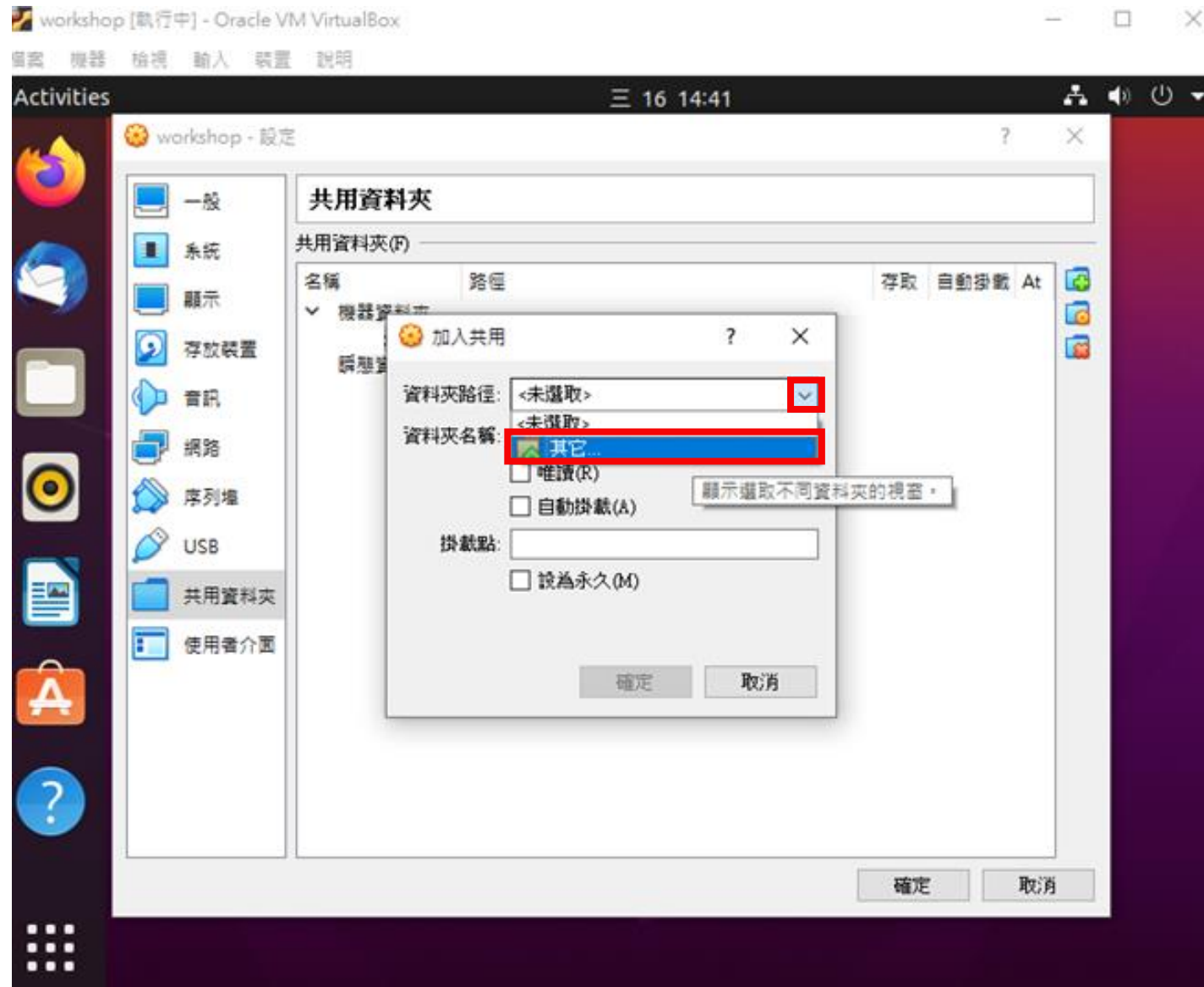
11. Set the folder to be shared



Download and Setup SDK

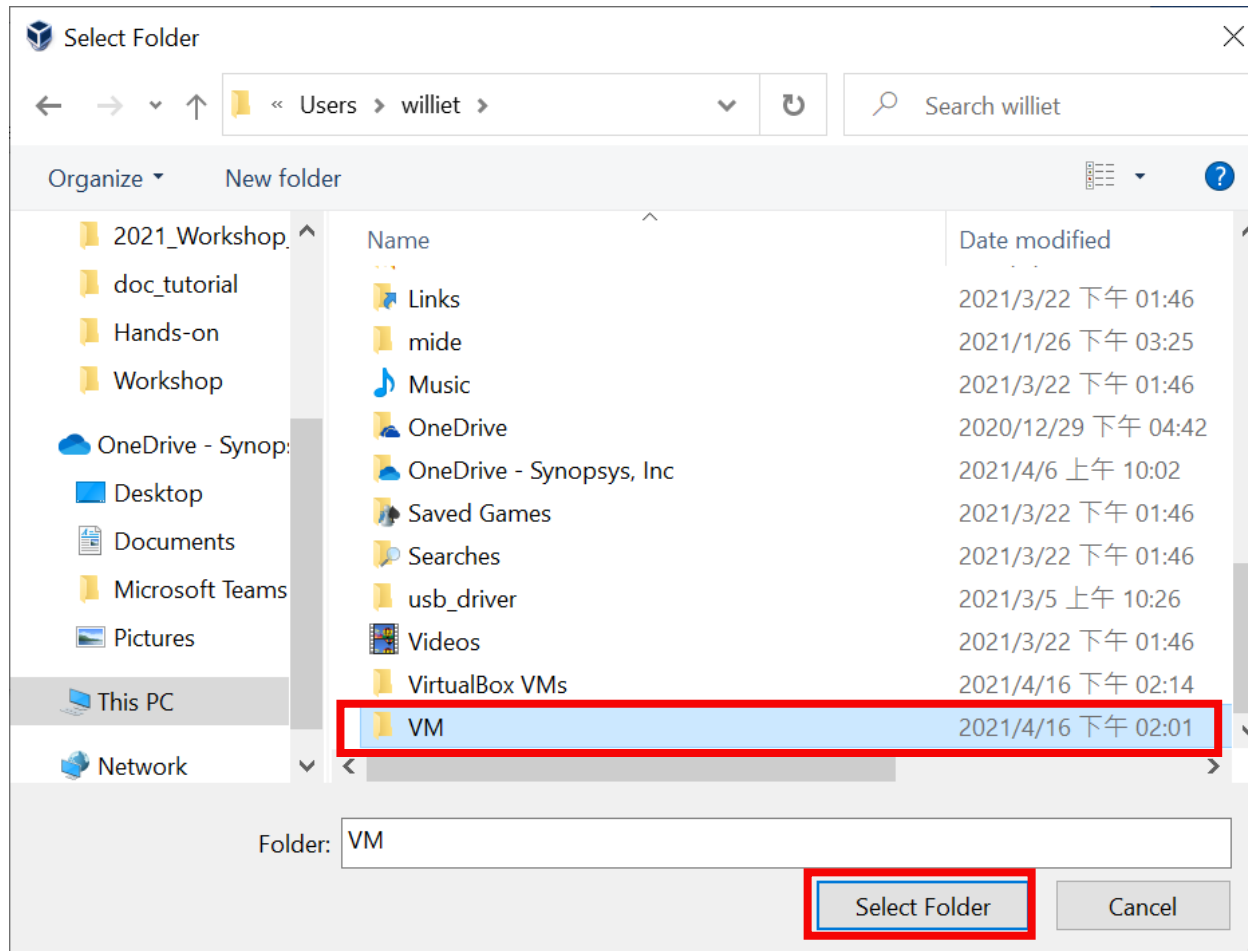


Download and Setup SDK

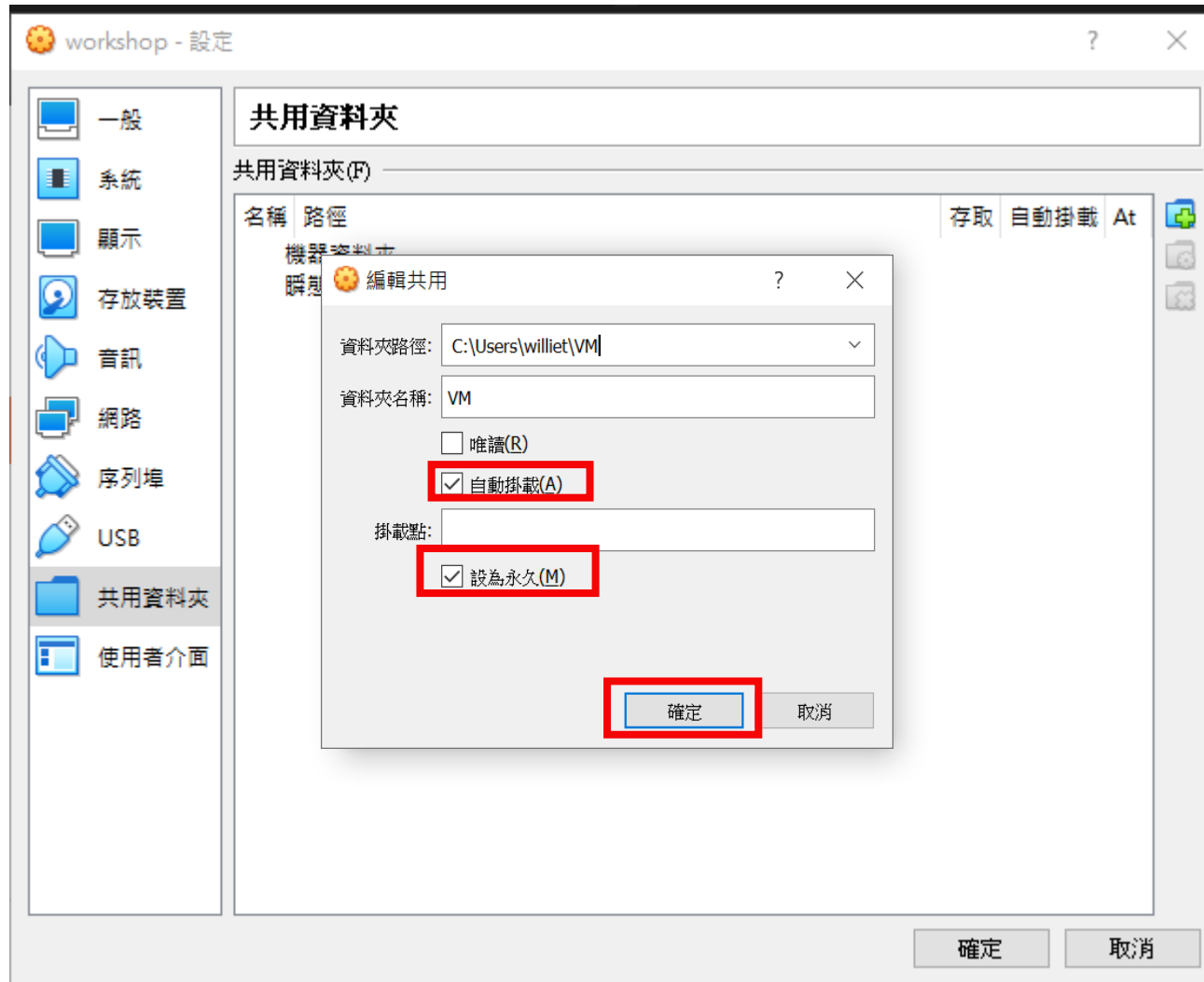


Download and Setup SDK

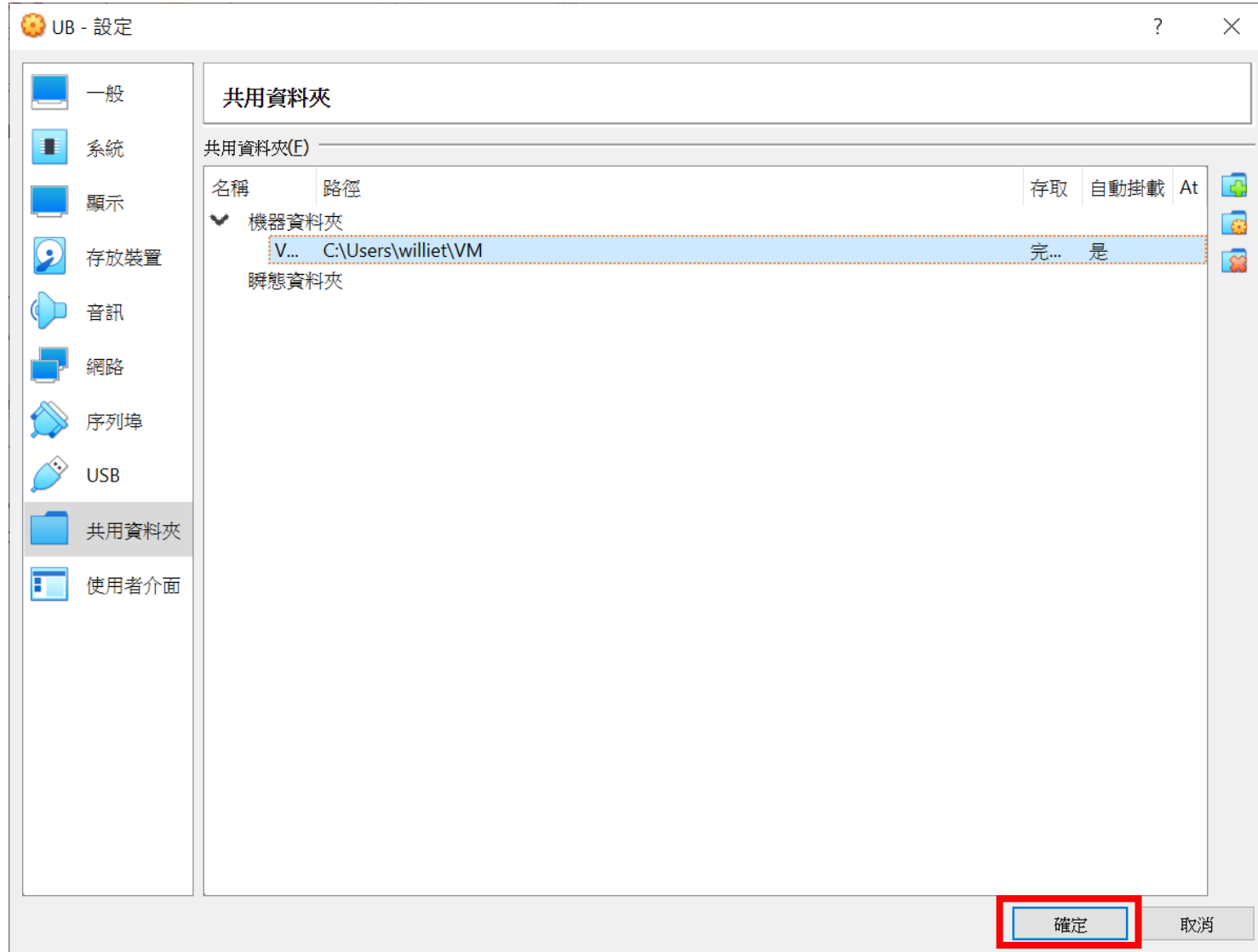
Path “C:\Users\{username}\VM”



Download and Setup SDK

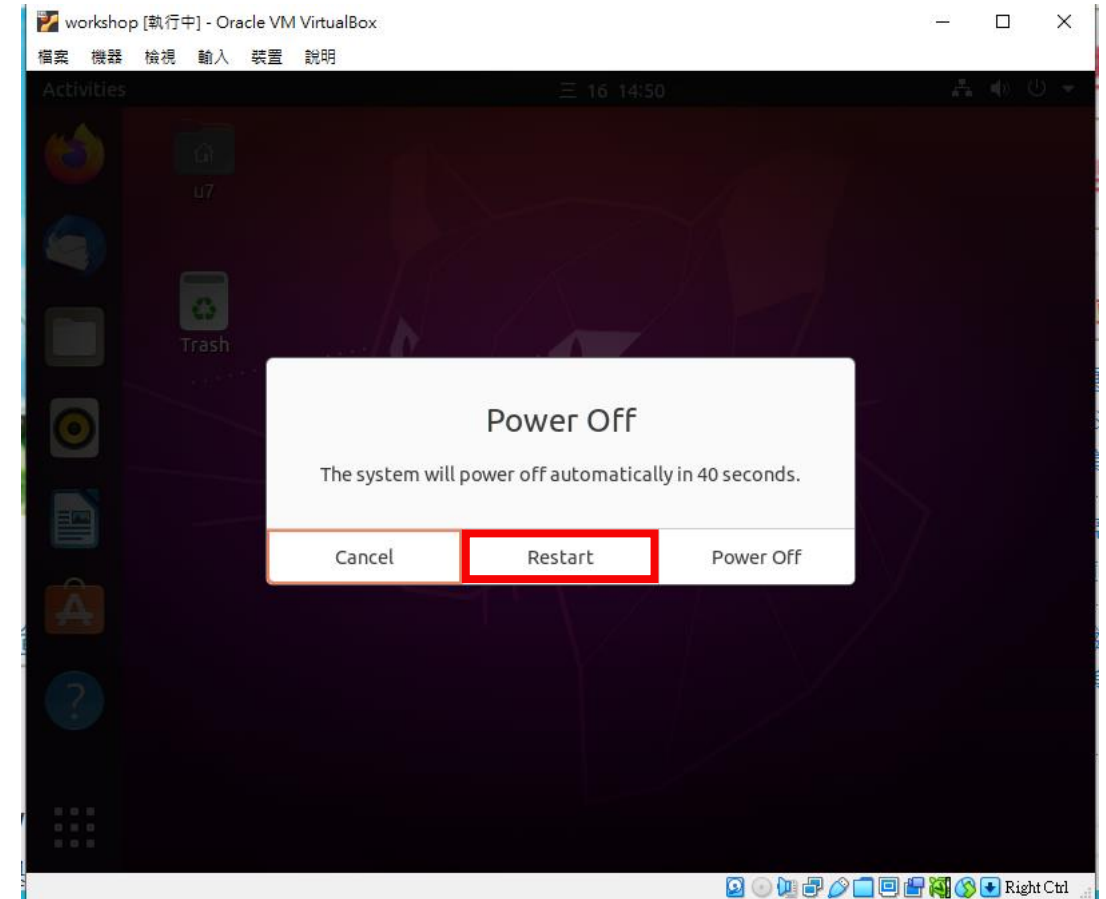
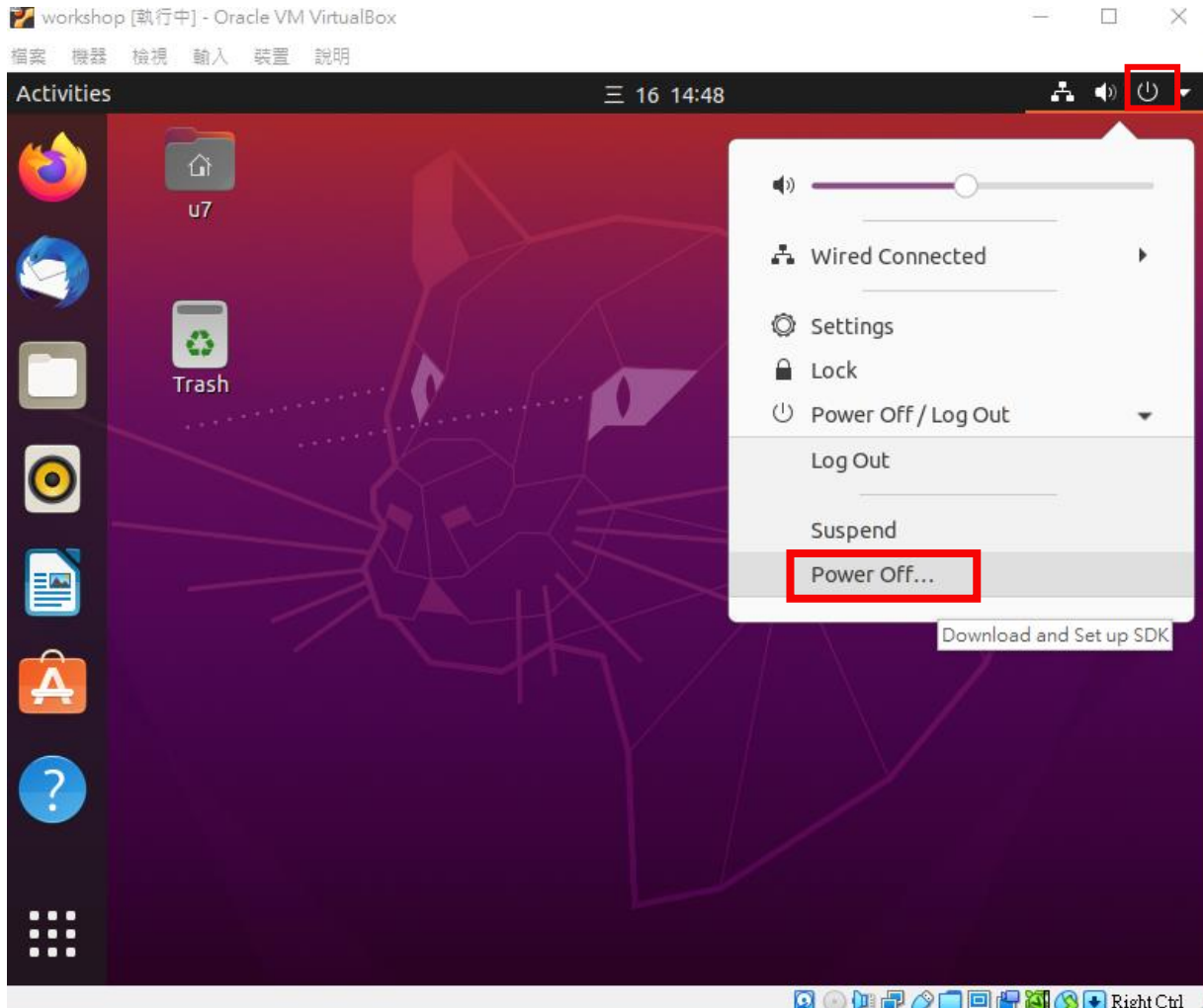


Download and Setup SDK



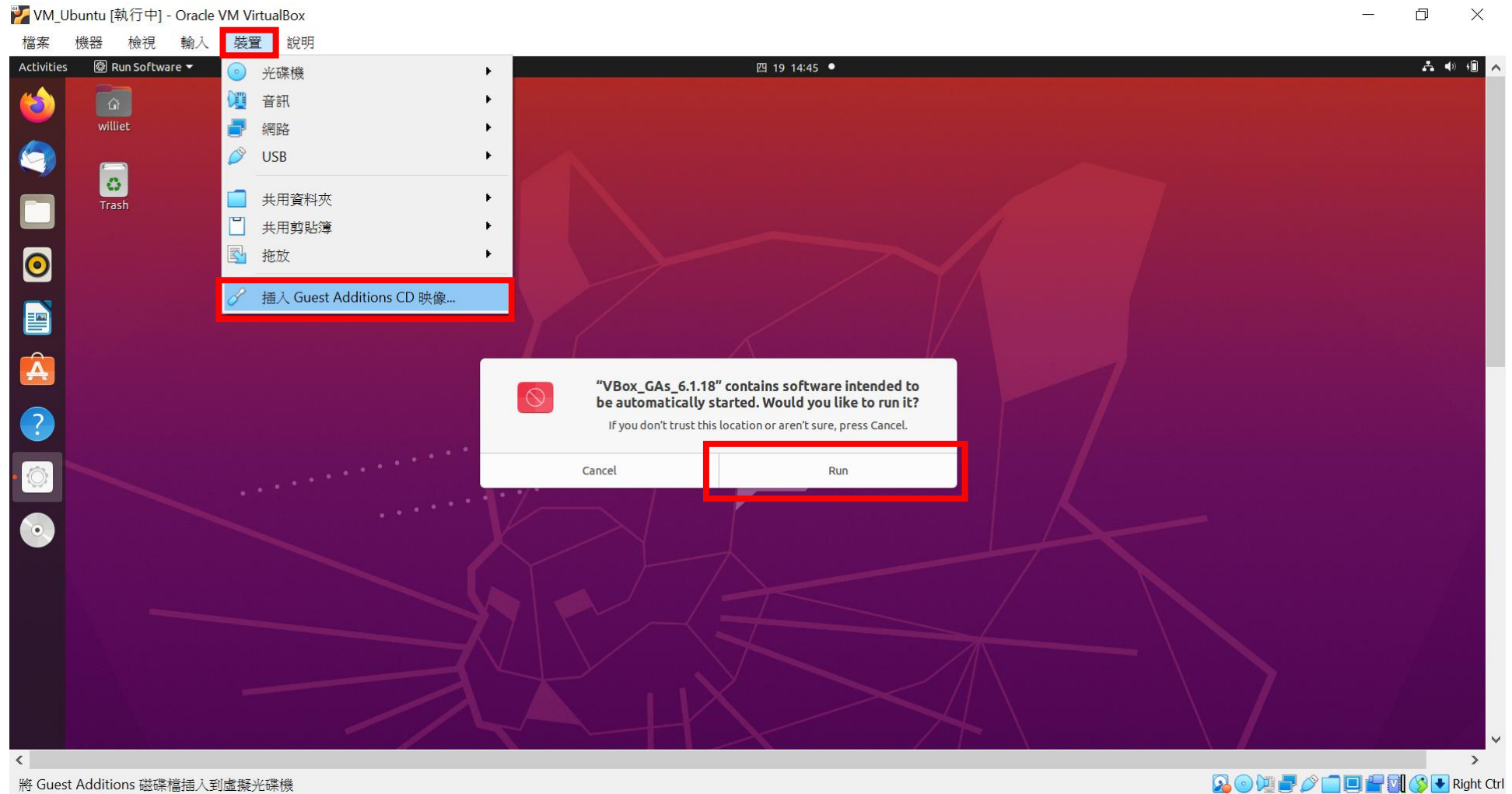
Download and Setup SDK

Restart 



Download and Setup SDK

Run Guest Addition CD

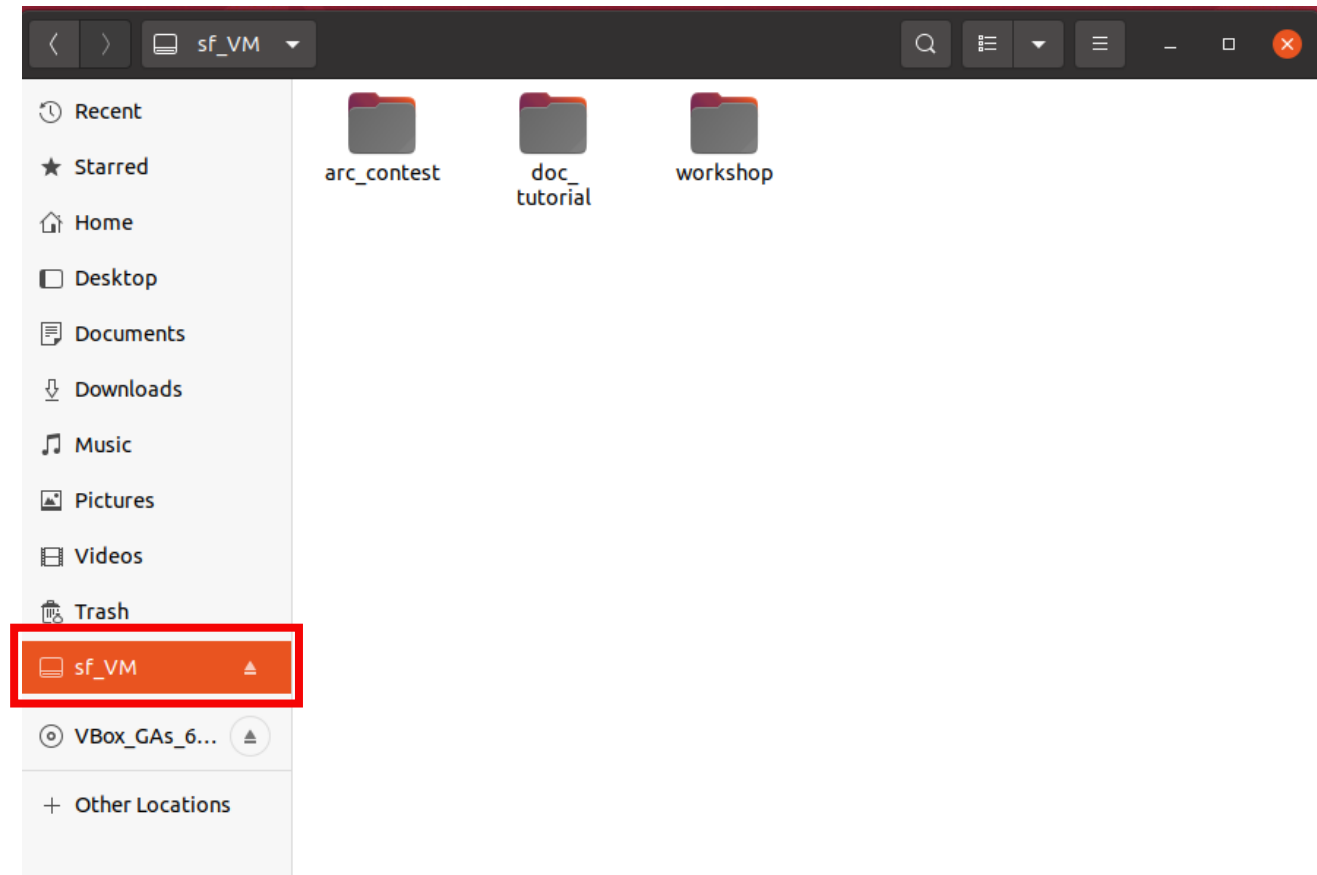


Download and Setup SDK

After you mount, you can see files in Ubuntu “sf_VM” which are also in Windows “C:\Users\{username}\VM”

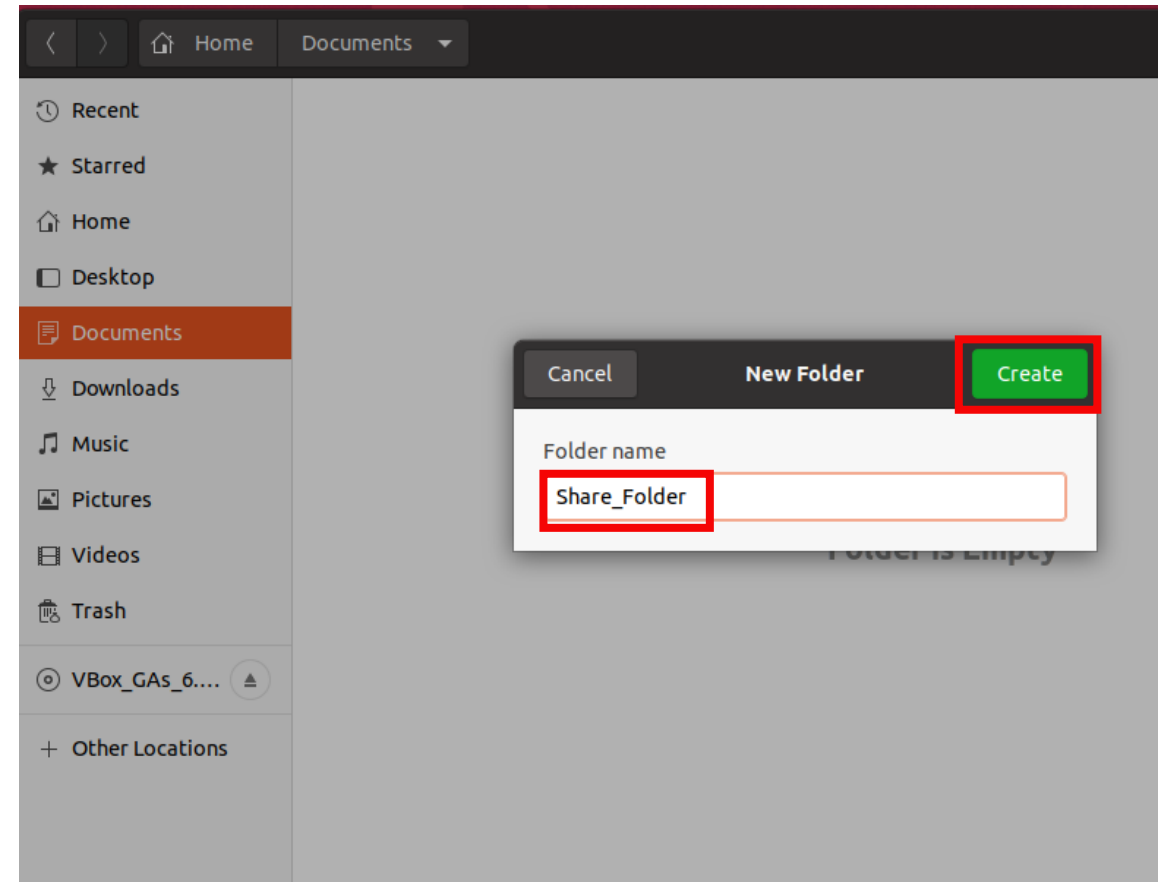
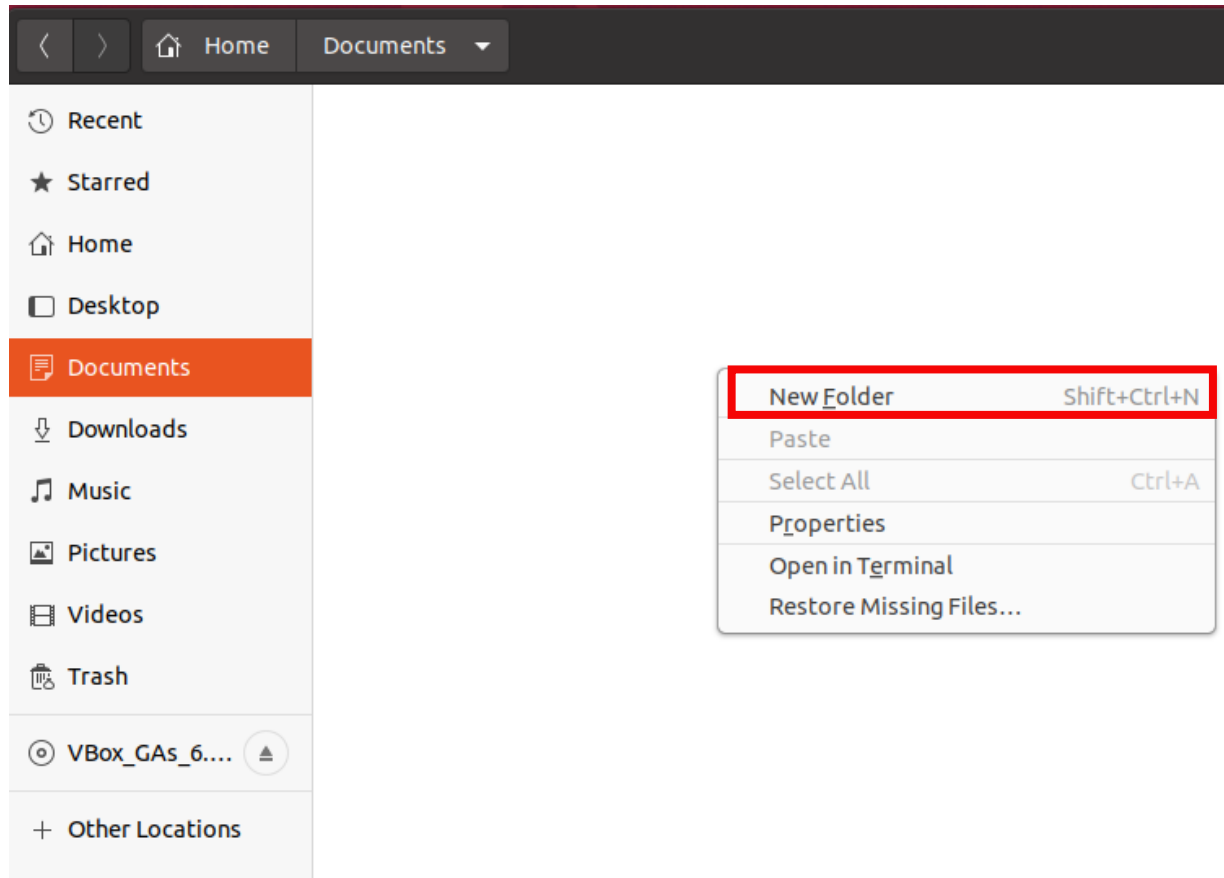
If you mount well, go to step 13.

If shared folder doesn't mount, please run step 12.



Download and Setup SDK

12. Create a new folder at “Home/Documents/Share_Folder”

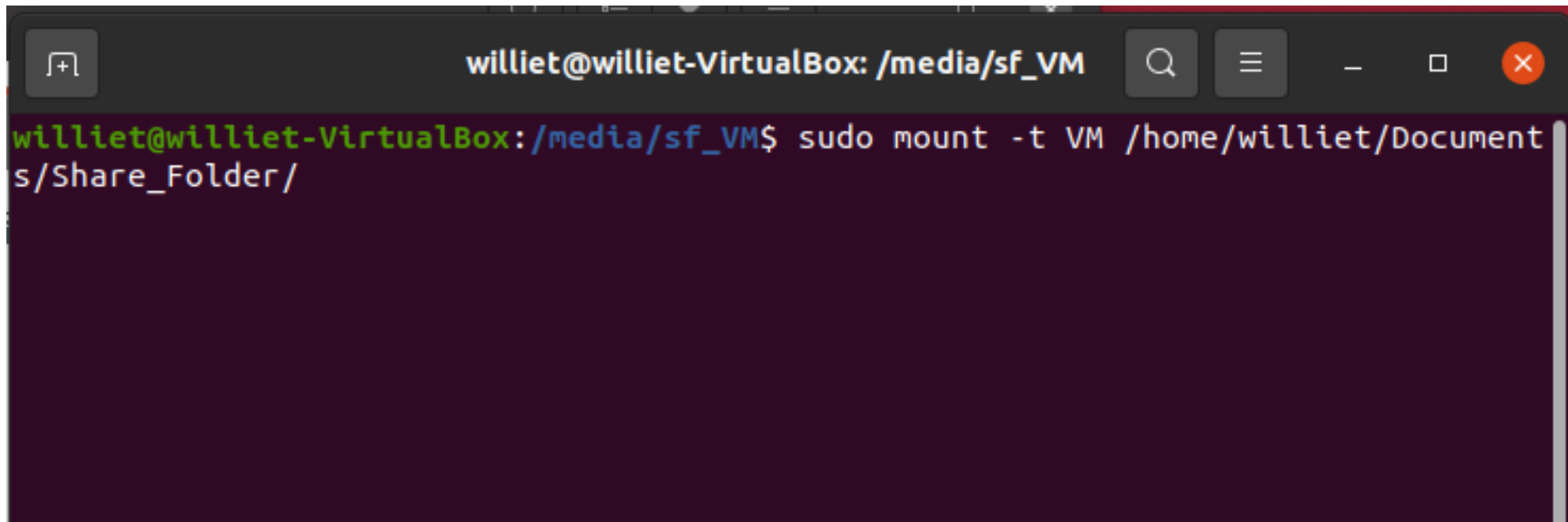


Download and Setup SDK

Open Ubuntu terminal: Ctrl+Alt+t (or right click > Open in Terminal)

\$ `sudo mount -t vboxsf VM /home/{username}/Documents/Share_Folder`

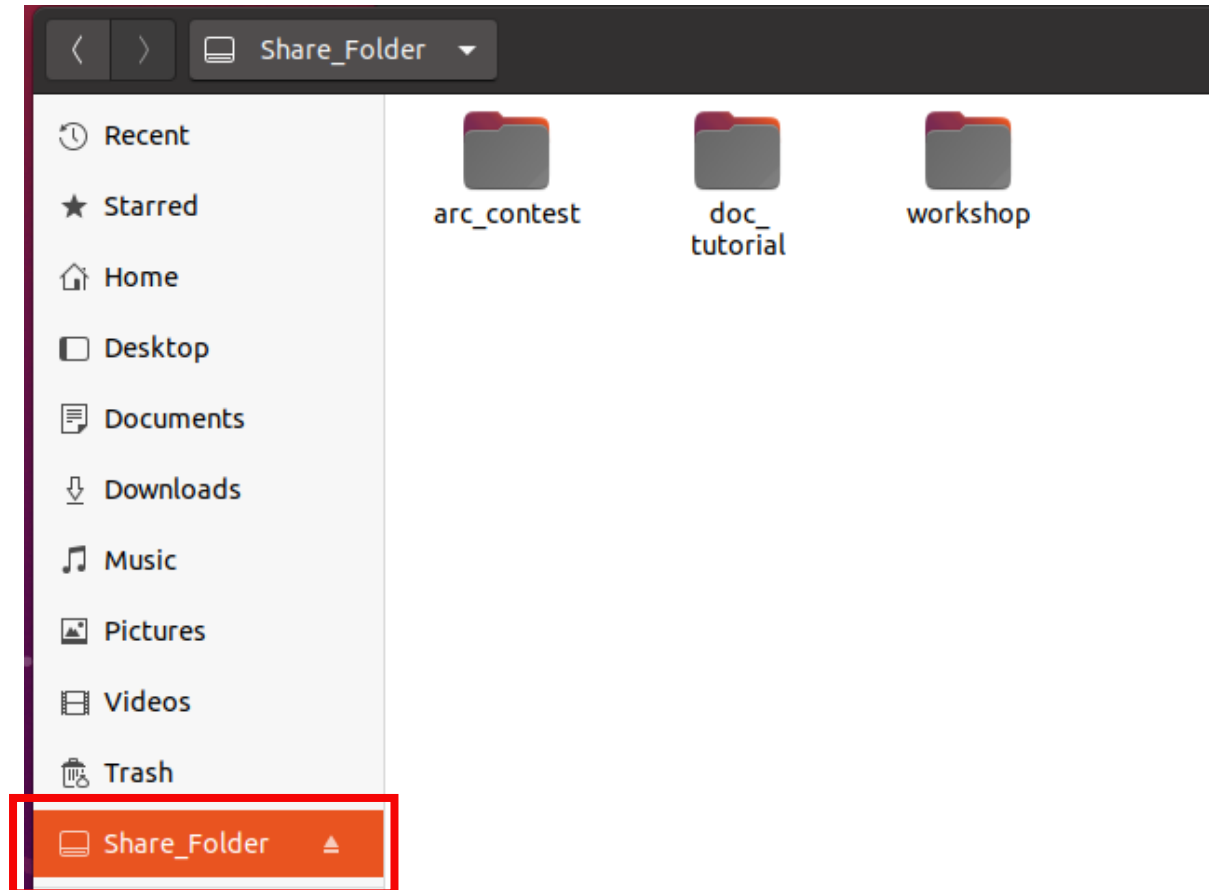
\$ Key-in your user password

A screenshot of a terminal window titled 'williet@williet-VirtualBox: /media/sf_VM'. The terminal shows the command 'sudo mount -t VM /home/williet/Document s/Share_Folder/' being entered. The prompt is 'williet@williet-VirtualBox:/media/sf_VM\$'. The terminal has a dark background with light-colored text. The window title bar includes standard Ubuntu window controls (minimize, maximize, close) and a search icon.

```
williet@williet-VirtualBox: /media/sf_VM
williet@williet-VirtualBox:/media/sf_VM$ sudo mount -t VM /home/williet/Document
s/Share_Folder/
```

Download and Setup SDK

After you mount, you can see files in Ubuntu
“home/{username}/Documents/Share_Folder” which are also in
Windows “C:\Users\{username}\VM”

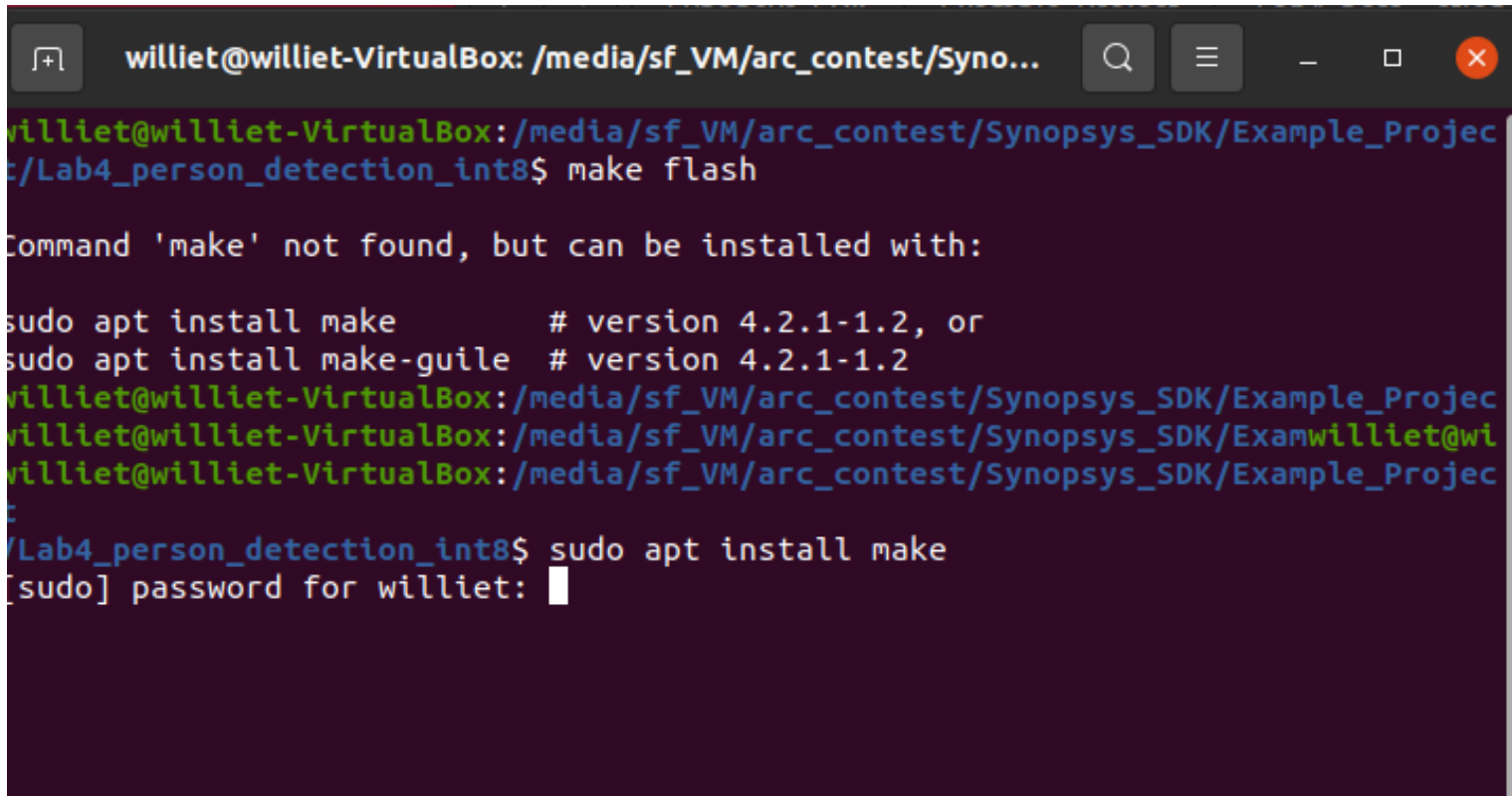


Download and Setup SDK

13. Install “make” and “git” command

\$ `sudo apt install make`

\$ Key-in your user password

A terminal window titled 'williet@williet-VirtualBox: /media/sf_VM/arc_contest/Synopsys_SDK/Example_Projec' shows a user attempting to run 'make flash'. The terminal output indicates that 'make' is not found and provides instructions to install it using 'sudo apt install make'. The user enters the password, and the installation process begins.

```
williet@williet-VirtualBox: /media/sf_VM/arc_contest/Synopsys_SDK/Example_Projec
t/Lab4_person_detection_int8$ make flash

Command 'make' not found, but can be installed with:

sudo apt install make          # version 4.2.1-1.2, or
sudo apt install make-guile    # version 4.2.1-1.2

williet@williet-VirtualBox: /media/sf_VM/arc_contest/Synopsys_SDK/Example_Projec
t/Lab4_person_detection_int8$ sudo apt install make
[sudo] password for williet: 
```

Download and Setup SDK

14. Download and setup arc-elf bin file

<https://github.com/foss-for-synopsys-dwc-arc-processors/toolchain/releases>

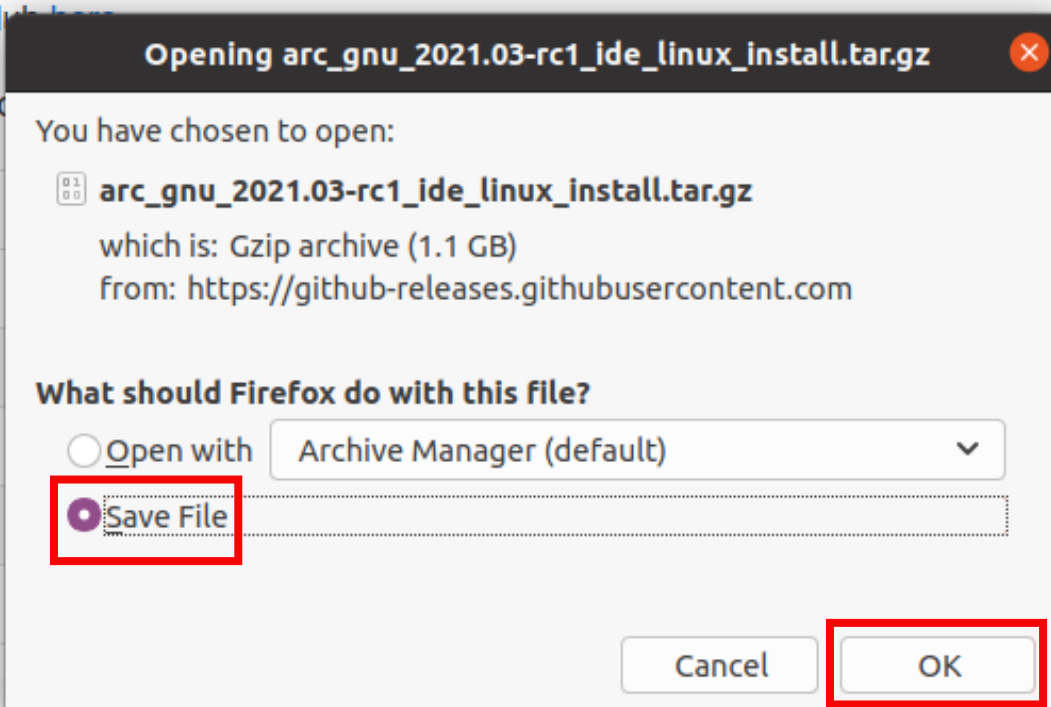
6. Native toolchain for ARC HS is not usable due to wrong build configuration.

Please report any problems by filing an Issue in GitHub [here](#).

Note: The toolchain is only supported for 64-bit version.


	Linux x86_64
Baremetal	Little endian \ Big endian
Linux/uClibc ARC700	Little endian \ Big endian
Linux/uClibc ARC HS	Little endian \ Big endian
Linux/glibc ARC HS	Little endian \ Big endian
IDE	Download

3ab6d863c12948484c7ae5e37ae1ffe8c32b3b1eacb5866ef090ade9025f6415 *arc_gnu_2021.03-rc1_prebuilt_elf32_le_linux.
d4fb3a6a9c67bcd088a624b266adc3bad47da42cah45aabf6112532ea1854f22 *arc_gnu_2021.03-rc1_sources.tar.gz



Opening arc_gnu_2021.03-rc1_ide_linux_install.tar.gz

You have chosen to open:

 arc_gnu_2021.03-rc1_ide_linux_install.tar.gz
which is: Gzip archive (1.1 GB)
from: <https://github-releases.githubusercontent.com>

What should Firefox do with this file?

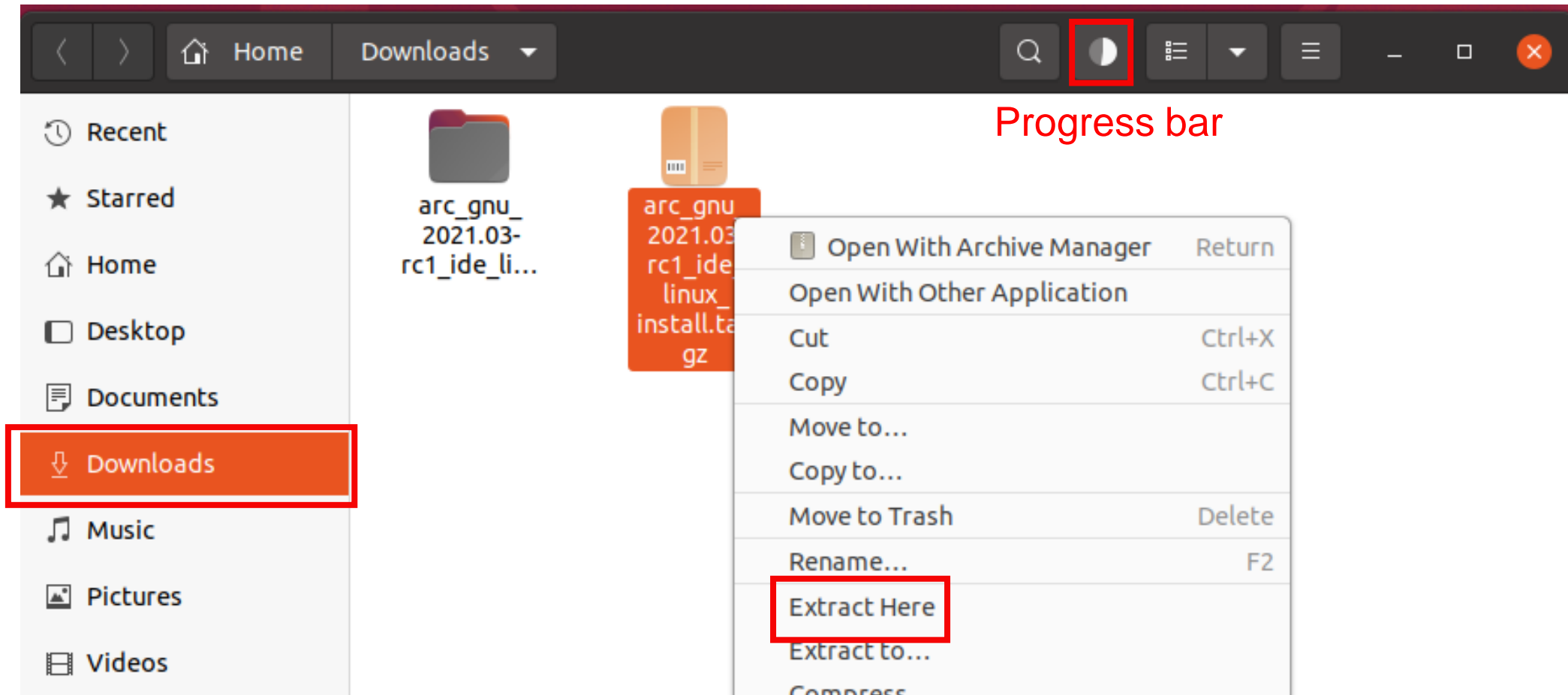
☐ Open with Archive Manager (default) ▼

☒ Save File

Cancel OK

Download and Setup SDK

15. Go to “Downloads” and unzip install package (Right Click > Extract Here)



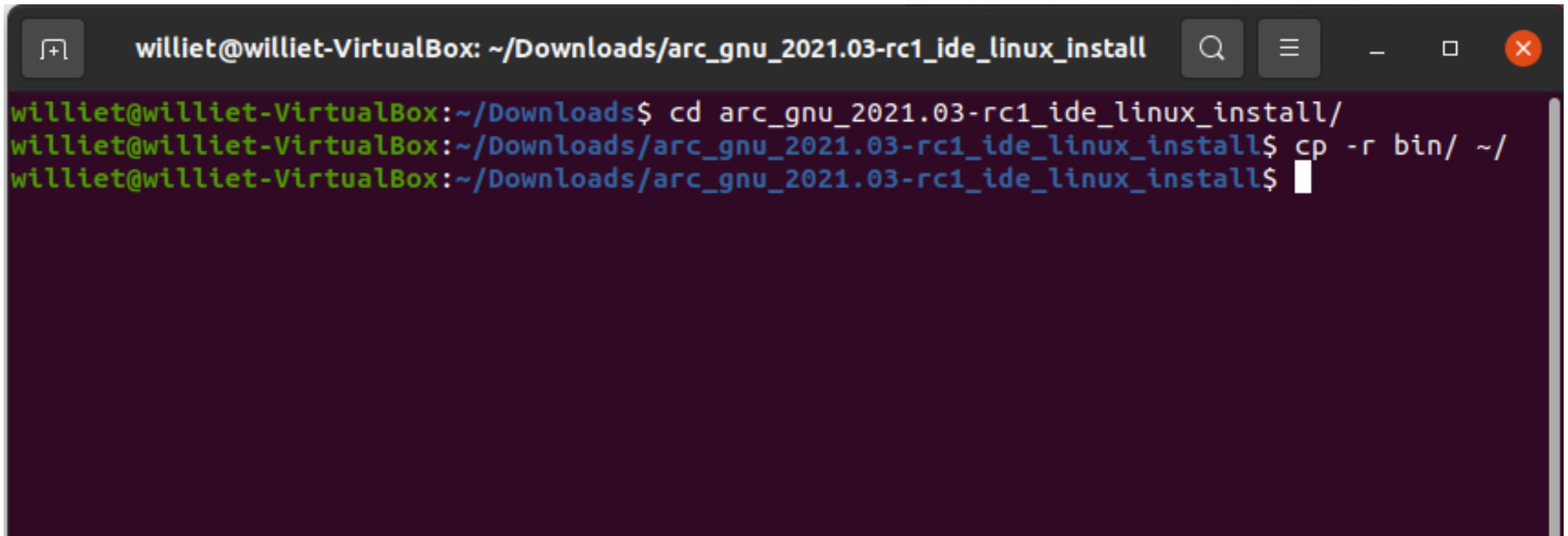
Download and Setup SDK

16. Open Terminal, go to folder and copy “bin” folder to “Documents”

```
$ cd arc_gnu_2021.03-rc1_ide_linux_install/
```

(Each version is different folder name)

```
$ cp -r /bin/ ~/
```

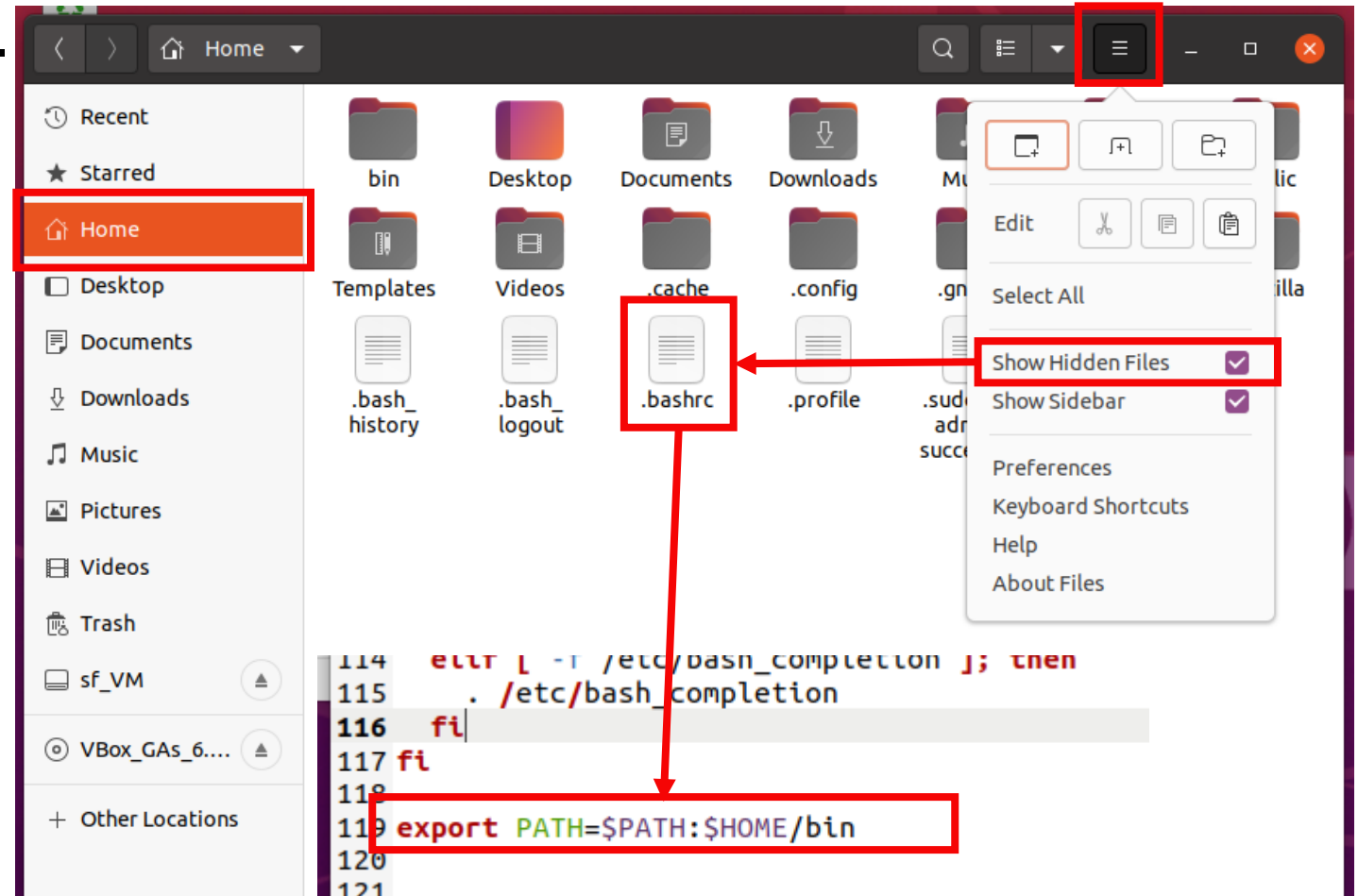


```
williet@williet-VirtualBox: ~/Downloads/arc_gnu_2021.03-rc1_ide_linux_install
williet@williet-VirtualBox:~/Downloads$ cd arc_gnu_2021.03-rc1_ide_linux_install/
williet@williet-VirtualBox:~/Downloads/arc_gnu_2021.03-rc1_ide_linux_install$ cp -r bin/ ~/
williet@williet-VirtualBox:~/Downloads/arc_gnu_2021.03-rc1_ide_linux_install$
```


Download and Setup SDK

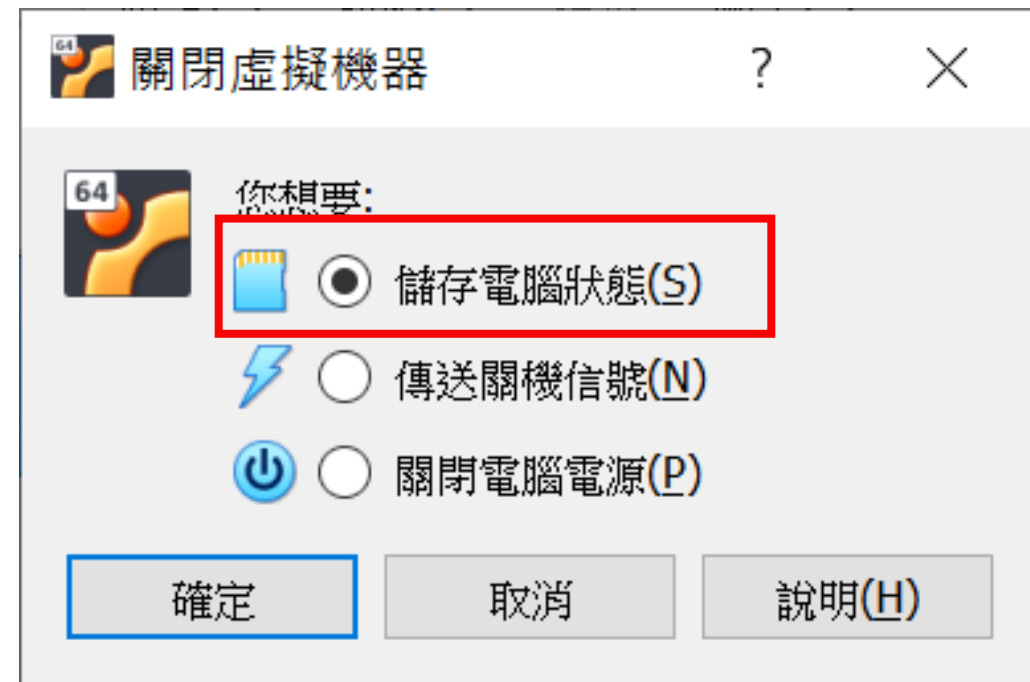
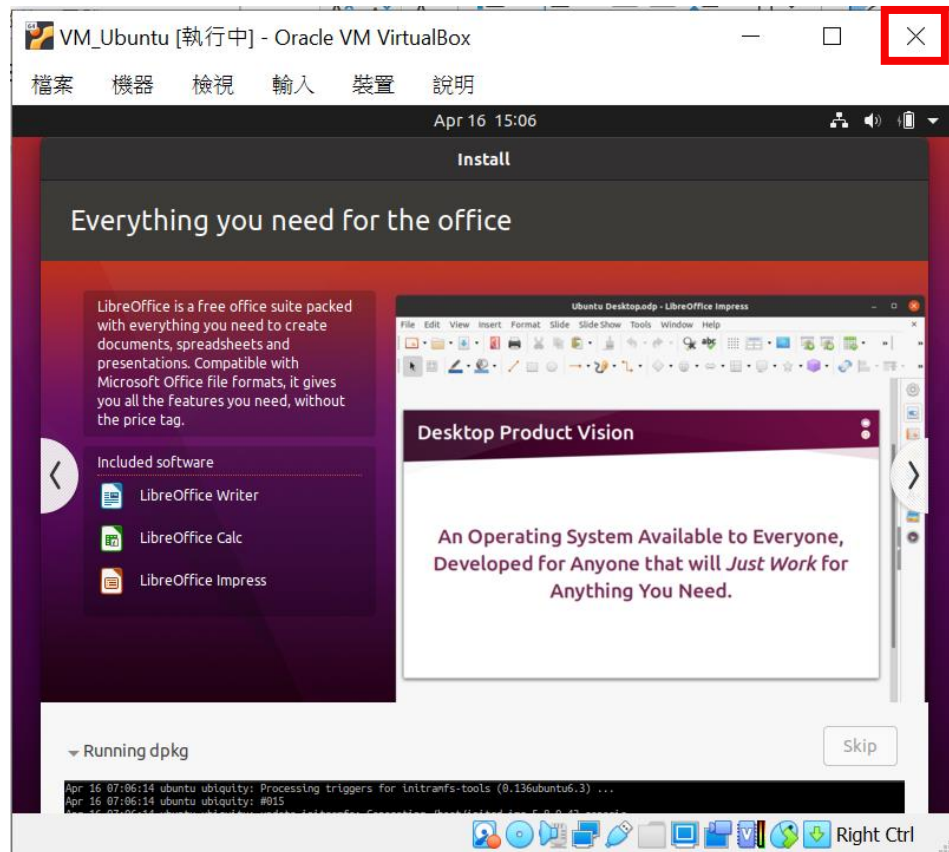
17. Edit /home/.bashrc for setting environment variable,
add “`export PATH=$PATH:$HOME/bin`” at the last line.

18. After edit it, restart Ubuntu.



Download and Setup SDK

19. Suggest! Close your VirtualBox with “Store Computer State”
If not, you may need to remount shard folder every time.



Program Code & Make Project & Make Flash File

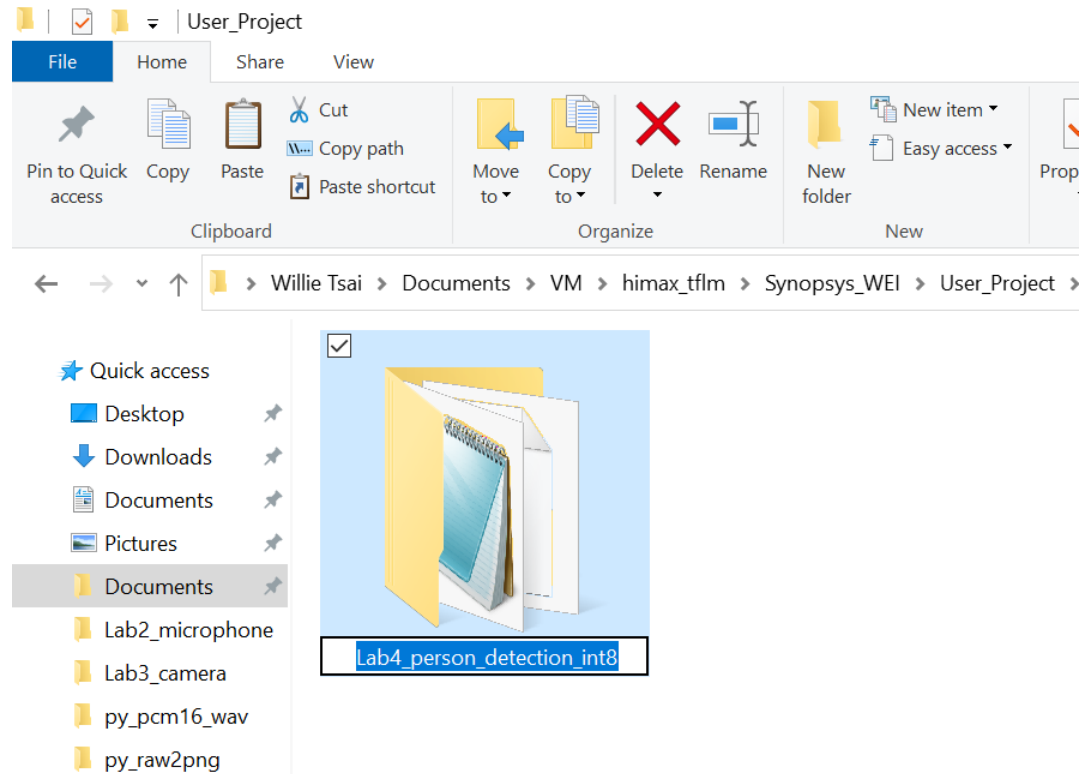


Program Code

1. Copy folder

“arc_contest/Synopsys_SDK/Example_Project/Lab4_person_detection_int8”
to folder “arc_contest/Synopsys_SDK/User Project/”

2. Rename the folder “Lab4_person_detection_int8” to “Flow_Test”



Program Code

3. Go into folder “*Flow_Test*” you will see folder “src” and “inc”
“src” folder: always keep your .c and .cc file in here.
“inc” folder: always keep your .h file in here.
(c file: c language)
(cc file: c++ language)

Make Project (By cygwin64)

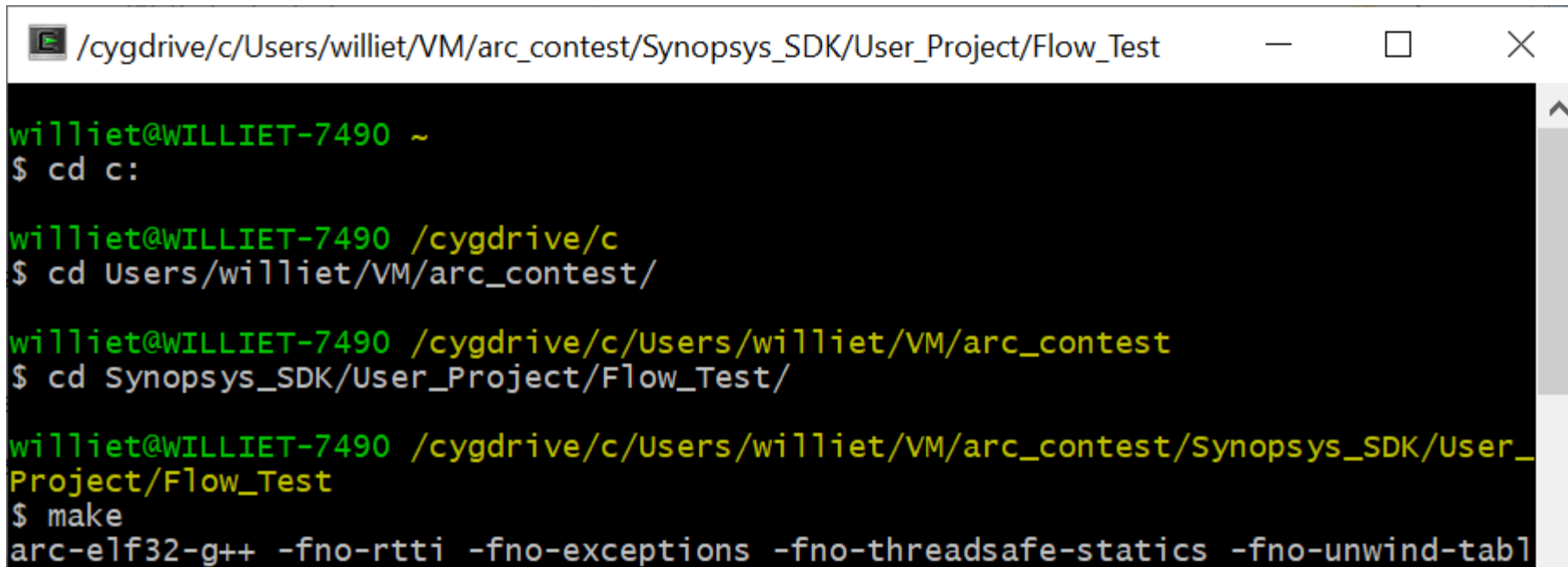
4. Go to your project path in cygwin64 terminal

```
$ cd c:
```

```
$ cd Users\{username}\VM\arc_contest
```

```
$ cd Synopsys_SDK/User_Project/Flow_Test
```

```
$ make
```



```
/cygdrive/c/Users/williet/VM/arc_contest/Synopsys_SDK/User_Project/Flow_Test
```

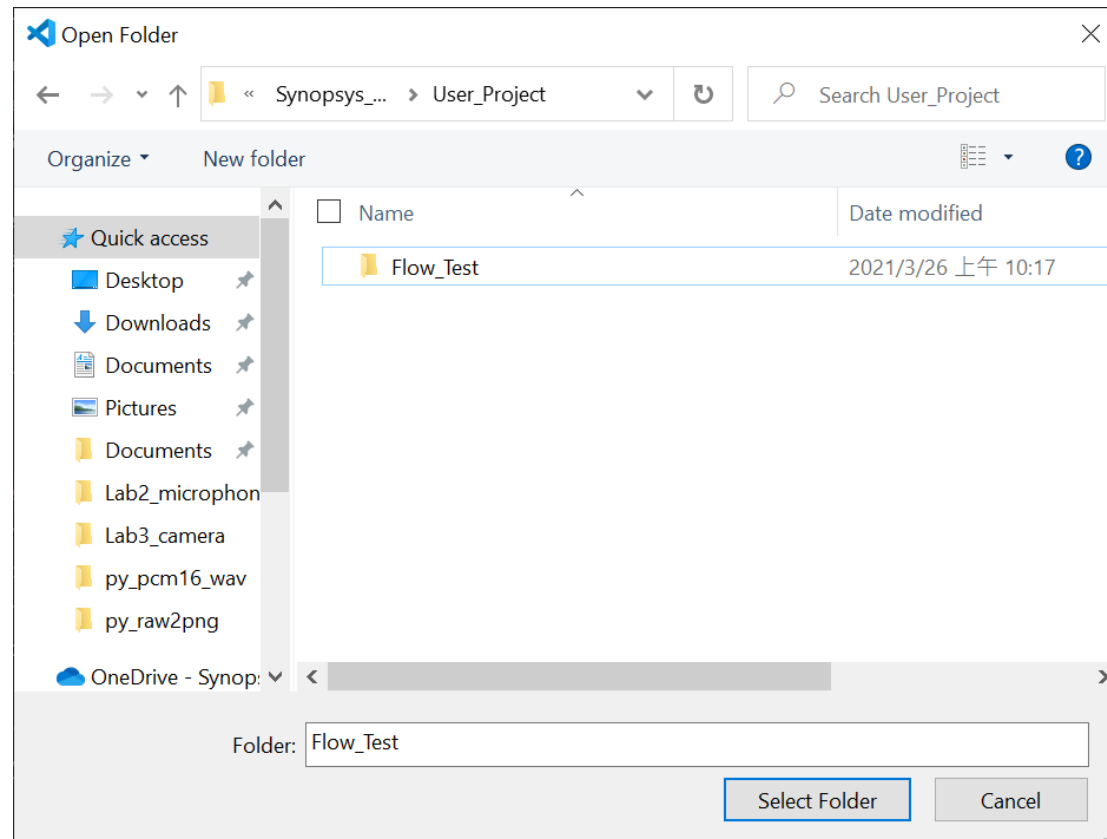
```
williet@WILLIET-7490 ~  
$ cd c:  
  
williet@WILLIET-7490 /cygdrive/c  
$ cd Users/williet/VM/arc_contest/  
  
williet@WILLIET-7490 /cygdrive/c/Users/williet/VM/arc_contest  
$ cd Synopsys_SDK/User_Project/Flow_Test/  
  
williet@WILLIET-7490 /cygdrive/c/Users/williet/VM/arc_contest/Synopsys_SDK/User_  
Project/Flow_Test  
$ make  
arc-elf32-g++ -fno-rtti -fno-exceptions -fno-threadsafe-statics -fno-unwind-tabl
```

Make Project (By Visual Studio Code)

4. You can also use Visual Studio Code to make project.

Open Folder and Select Folder:

“...../arc_contest/Synopsys_SDK/User_Project/Flow_Test”

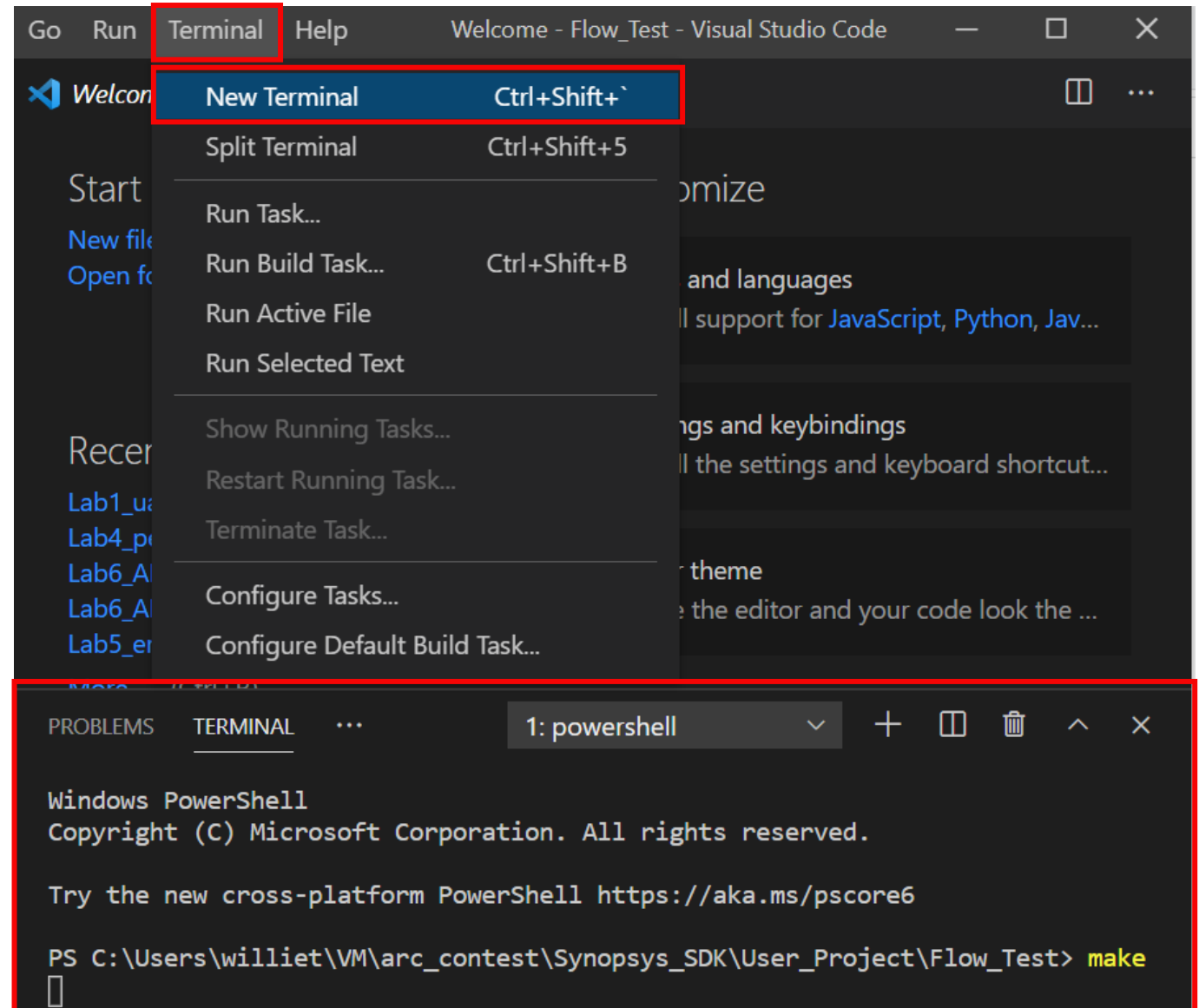


Make Project (Visual Studio Code)

Terminal > New Terminal
You will see terminal block.

Type command:

\$ make

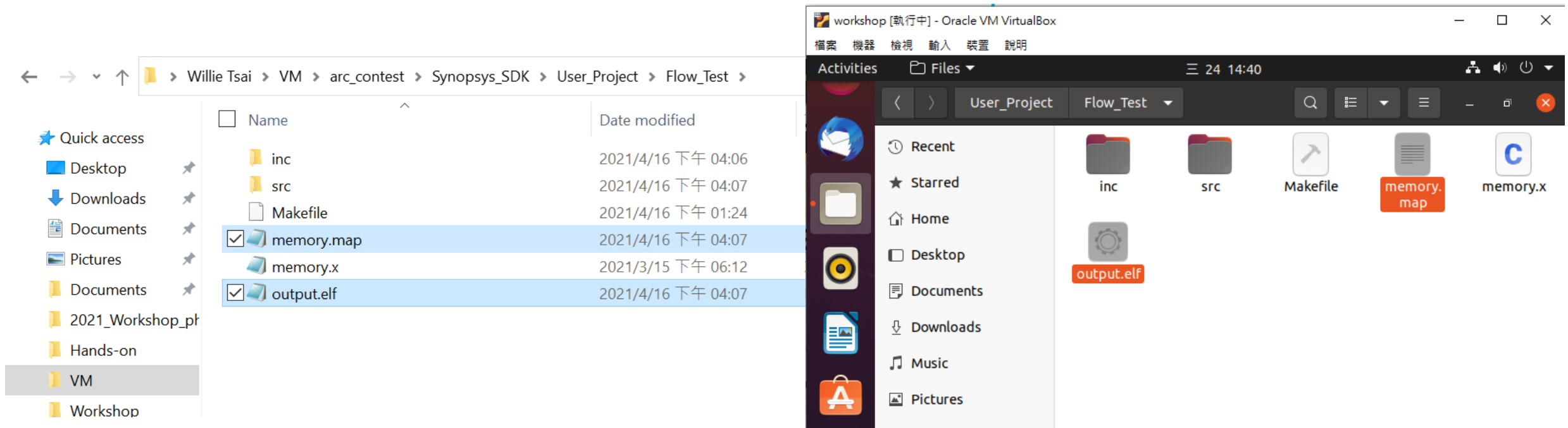


Make Project

5. Check your folder whether contains .elf and .map files.

GNU: output.elf & memory.map

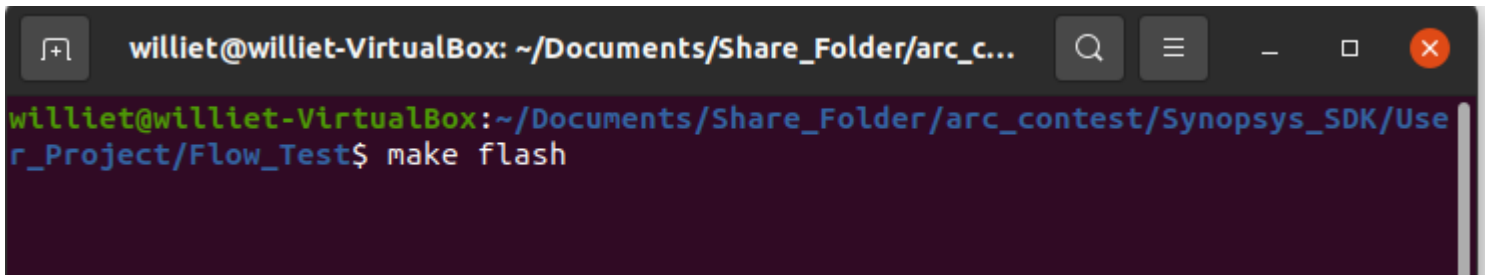
MetaWare: output.elf & output.map



Make Flash File

6. Open terminal in VirtualBox Ubuntu, and go to the same path:
“...../arc_contest/Synopsys_SDK/User_Project/Flow_Test”

\$ make flash

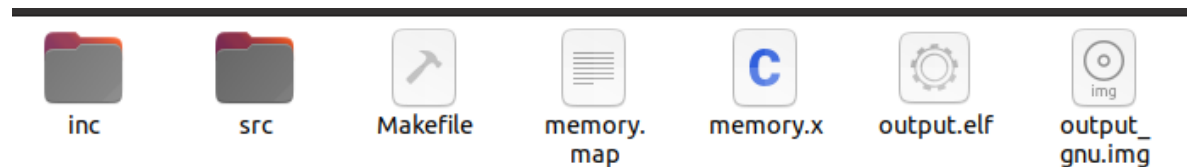


```
williet@williet-VirtualBox: ~/Documents/Share_Folder/arc_c...  
williet@williet-VirtualBox:~/Documents/Share_Folder/arc_contest/Synopsys_SDK/User_Project/Flow_Test$ make flash
```

7. Check your img file in directory, it will be downloaded to WE-I.

GNU: output_gnu.img

MetaWare: output_mwdt.img



Make Project and Flash File

There are some commands can be used,

\$ **make** : compile and link your project, then create .elf and .map file

\$ **make flash** : combine .elf and .map file to .img file

\$ **make clean** : remove all .o file of this project

\$ **make clean_all** : remove all .o file of this project and third party

You can add a command for changing toolchain

(default toolchain is gnu, define in makefile)

“ARC_TOOLCHAIN=mwdt”: compile with MetaWare

“ARC_TOOLCHAIN=gnu”: compile with ARC GNU Toolchain

Please use \$ **make clean_all** before you change toolchain.

Update and Run Application On WE-I

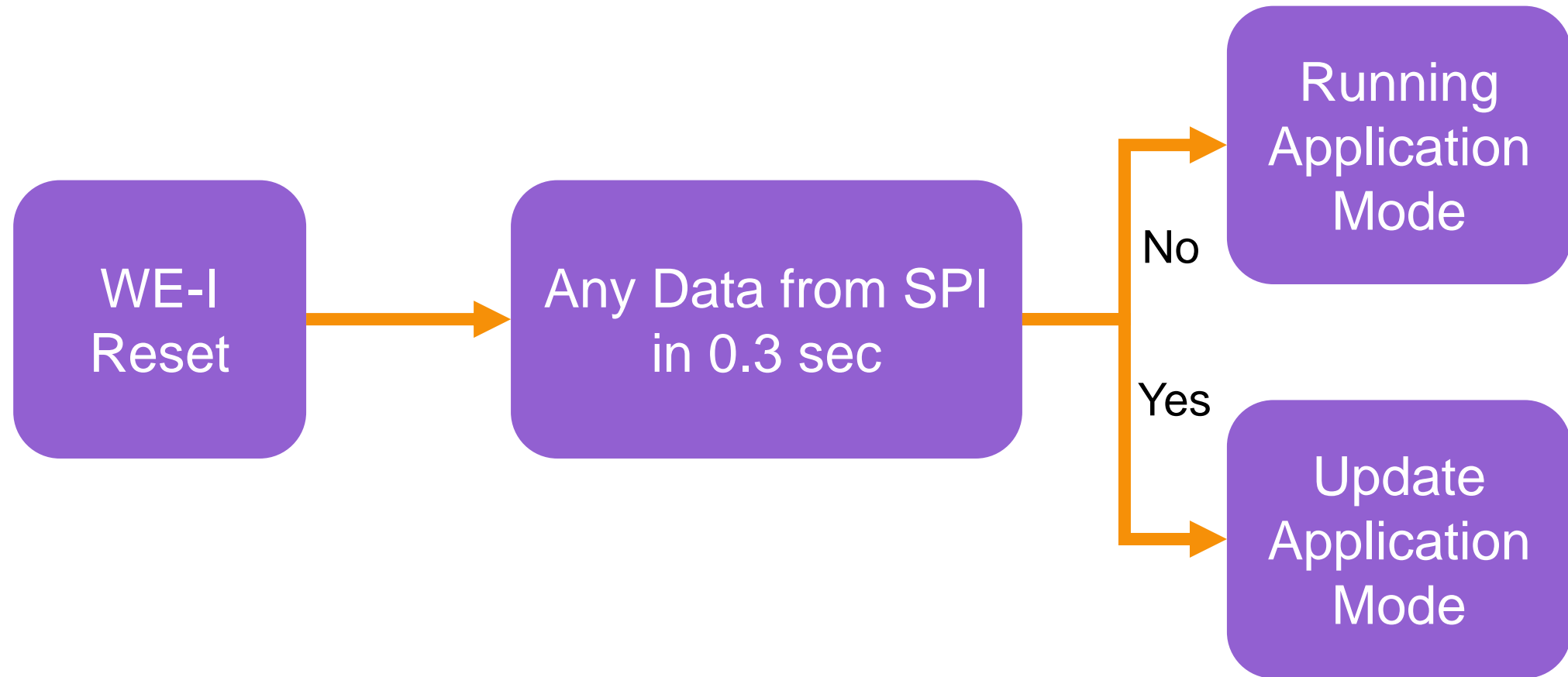


Project Development Flow

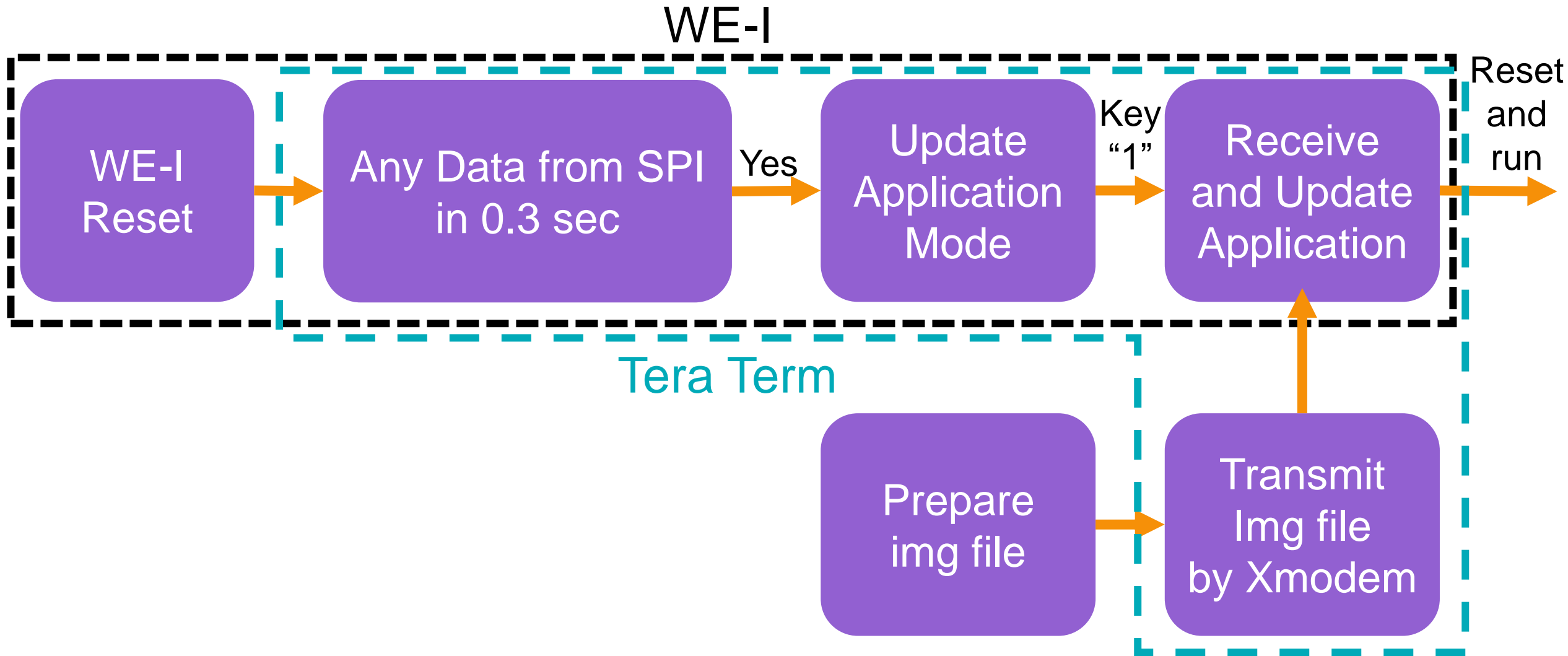


Stage	TensorFlow Model Development	Firmware Development	Run / Update Application On WE-I
Tool	Anaconda Cygwin	Cygwin Metaware or ARC GNU VirtualBox (Ubuntu 20.04)	Tera Term USB Micro
Language	Python 3	C language C++ language	

Run / Update Application On WE-I



Update Application On WE-I



Update Application On WE-I

Connect WE-I with Tera Term

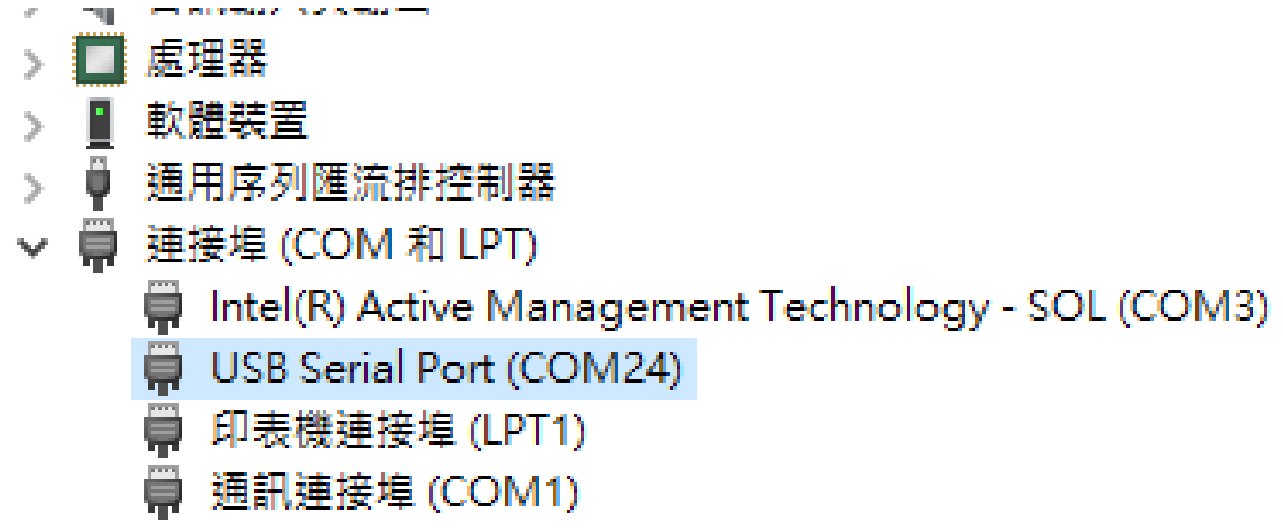
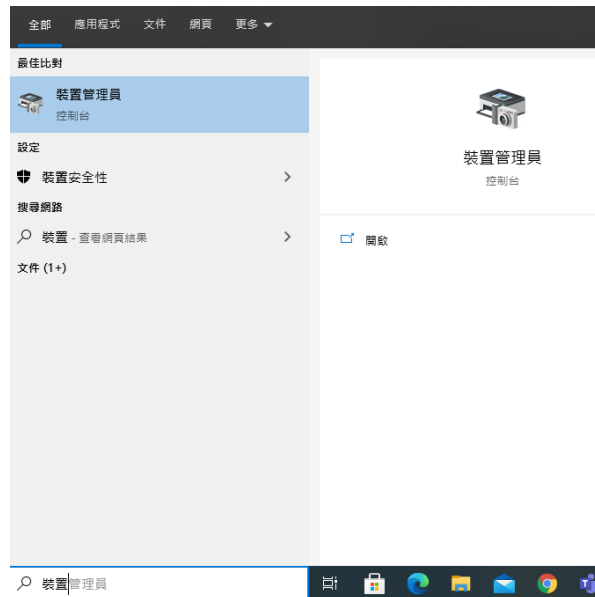
1. Connect WE-I and PC by USB Cable

2. Check your WE-I usb port number

裝置管理員 > 連接埠(COM & LPT) > USB Serial Port (COMx)

x: This is your WE-I usb port number

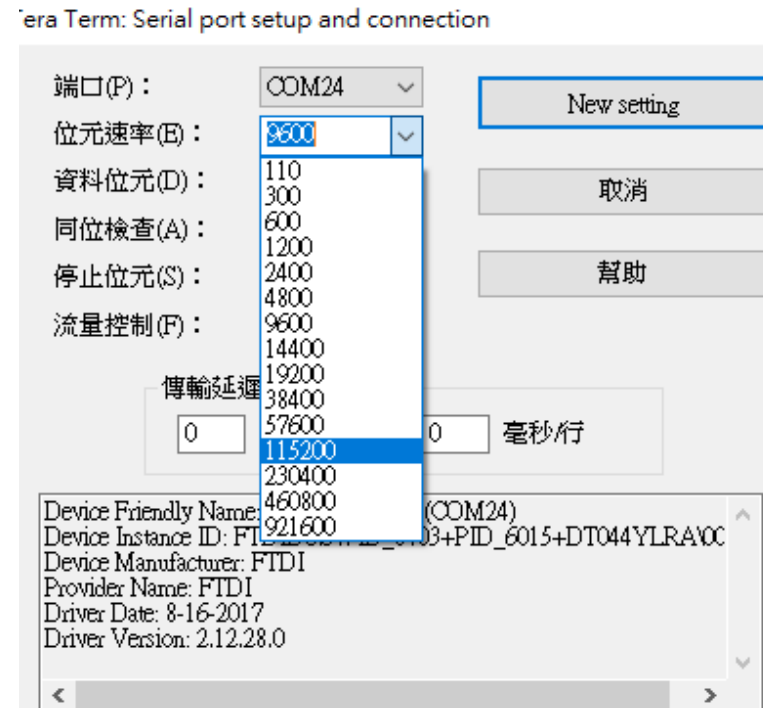
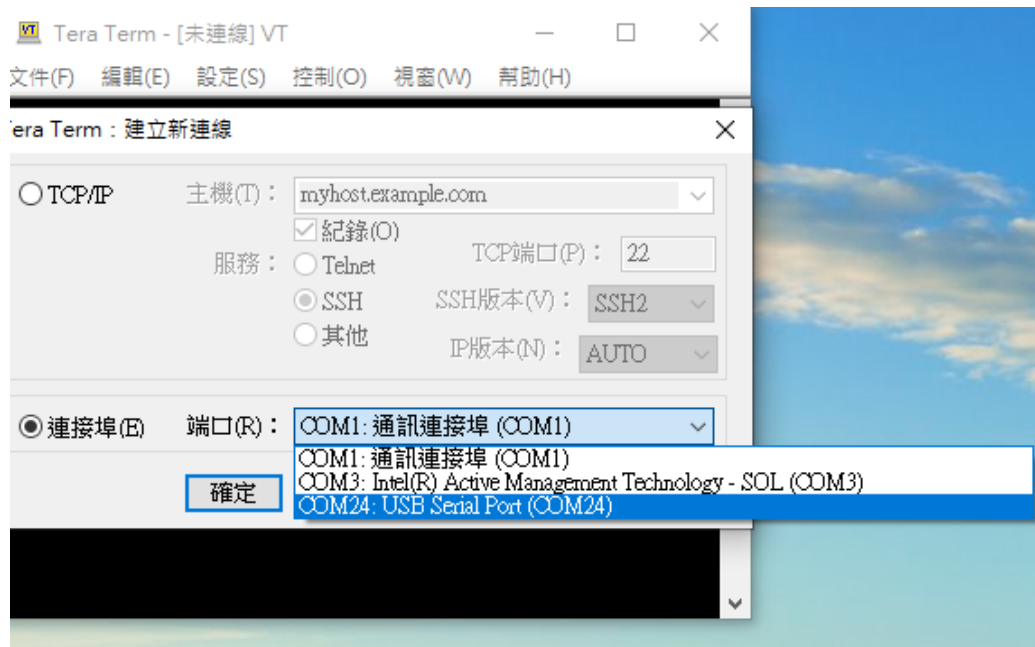
(If USB Serial Port is not shown here, please refer to Appendix-2)



Update Application On WE-I

Connect WE-I with Tera Term

3. Open tera term and select “COMx: USB Serial Port (COMx)”
4. Tera term Setup > Serial Port > Change Baud to 115200, and keep other setting.



Update Application On WE-I

Connect WE-I with Tera Term

5. Reset WE-I by pushing SW2, you will see startup information on tera term.

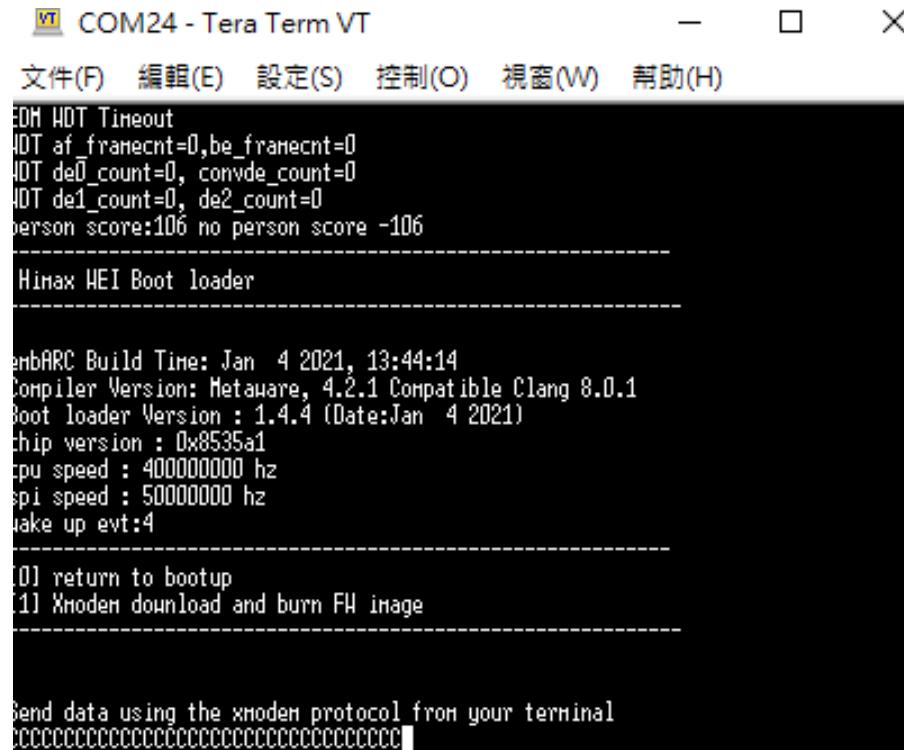


```
-----  
Himax WEI Boot loader  
-----  
  
enbARC Build Time: Jan  4 2021, 13:44:14  
Compiler Version: MetaWare, 4.2.1 Compatible Clang 8.0.1  
Boot loader Version : 1.4.4 (Date:Jan  4 2021)  
chip version : 0x8535a1  
cpu speed : 400000000 hz  
spi speed : 50000000 hz  
wake up evt:4  
...secure lib version = 352380df9a347b1187d2361bfcd4455178a1ebcb  
1st APPLICATION addr[31]=21000 (main-1966)  
Bootloader Done !!!!!  
jump to app FH : 0x10000004  
12 bytes lost due to alignment. To avoid this loss, please make sure the t  
HM0360 RevB,C,D Config  
person score:-2 no person score 2  
person score:-6 no person score 6
```

Update Application On WE-I

Enable Update Application Mode with Tera Term

1. Finish to connect WE-I with Tera Term
2. Click on any display area
3. Keep to press key “1” on the keyboard, and press reset button
4. WE-I will start to receive img file by Xmodem



The screenshot shows a Tera Term VT window titled "COM24 - Tera Term VT". The menu bar includes "文件(F)", "編輯(E)", "設定(S)", "控制(O)", "視窗(W)", and "幫助(H)". The terminal output displays the following text:

```
EDM WDT Timeout
WDT af_framecnt=0,be_framecnt=0
WDT de0_count=0, convde_count=0
WDT de1_count=0, de2_count=0
person score:106 no person score -106
-----
Himax WEI Boot loader
-----

emBARC Build Time: Jan  4 2021, 13:44:14
Compiler Version: MetaWare, 4.2.1 Compatible Clang 8.0.1
Boot loader Version : 1.4.4 (Date:Jan  4 2021)
chip version : 0x8535a1
cpu speed : 400000000 hz
spi speed : 50000000 hz
wake up evt:4
-----

[0] return to bootup
[1] Xmodem download and burn FH image
-----

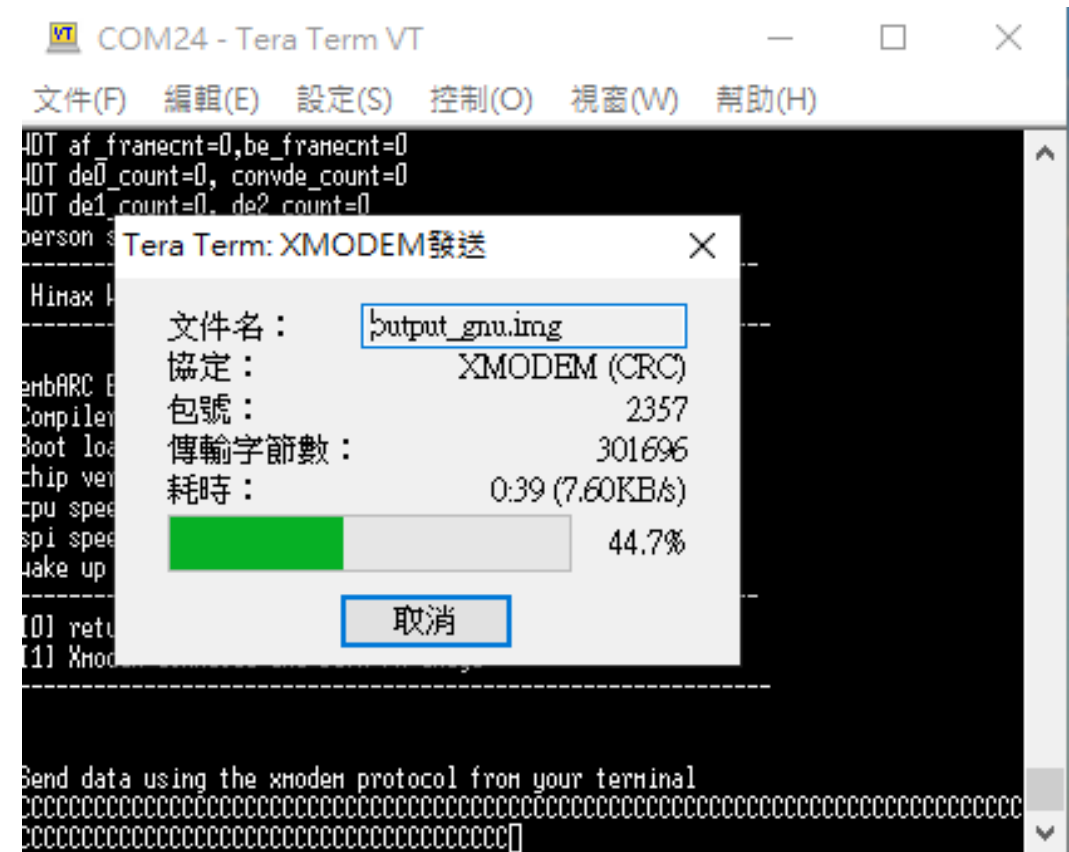
Send data using the xmodem protocol from your terminal
cccccccccccccccccccccccccccccccccccccccc
```

Update Application On WE-I

Enable Update Application Mode with Tera Term

5. Tera term File > Transfer > XMODEM > Send > select img file

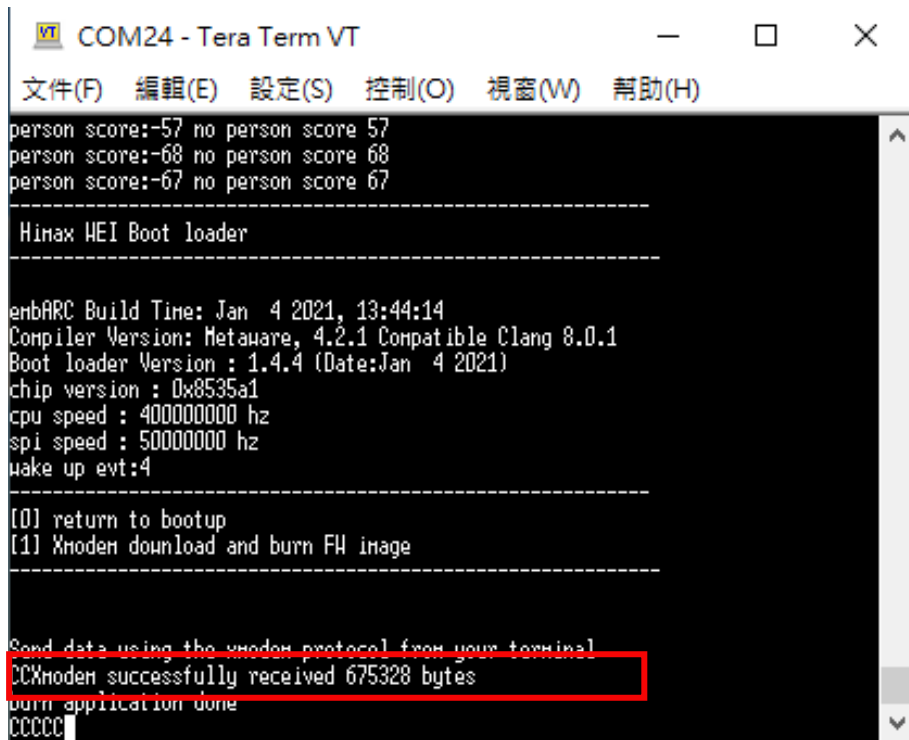
6. Wait for Transmit



Update Application On WE-I

Enable Update Application Mode with Tera Term

7. Terminal will show “Xmodem successfully received xxx bytes” after transmission
8. Press reset button to run your application



```
COM24 - Tera Term VT
文件(F) 編輯(E) 設定(S) 控制(O) 視窗(W) 幫助(H)
person score:-57 no person score 57
person score:-68 no person score 68
person score:-67 no person score 67
-----
Hinax WEI Boot loader
-----
enbARC Build Time: Jan  4 2021, 13:44:14
Compiler Version: MetaWare, 4.2.1 Compatible Clang 8.0.1
Boot loader Version : 1.4.4 (Date:Jan  4 2021)
chip version : 0x8535a1
cpu speed : 400000000 hz
spi speed : 50000000 hz
wake up evt:4
-----
[0] return to bootup
[1] Xmodem download and burn FH image
-----
Send data using the xmodem protocol from your terminal
CCXmodem successfully received 675328 bytes
burn application done
CCCCC
```

Run Application On WE-I

1. Connect USB cable to power up WE-I
you can also power up by 2.54 pitch connector
2. For debug easily, suggest to use Tera Term and print date or result
3. Start to develop your project, and debug your code

Appendix-2: Troubleshooting - Update VCP Driver



Troubleshooting – Update VCP driver

If the USB serial port is not shown in Ports (COM & LPT):

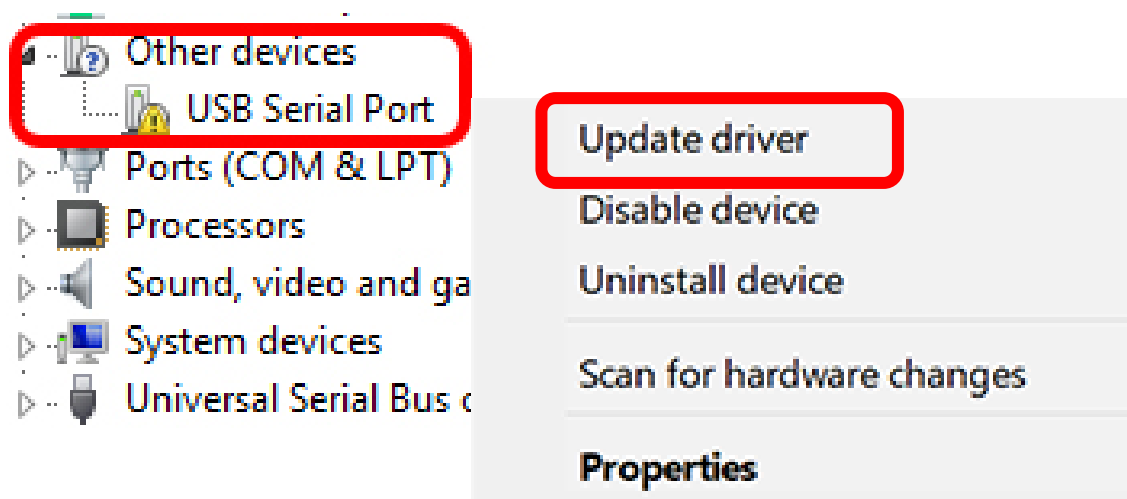
1. Download VCP driver: <https://ftdichip.com/drivers/vcp-drivers/>
Select Windows/X64 version
2. Unzip the downloaded file (CDM v2.12.28 WHQL Certified)

Operating System	Release Date	Processor Architecture				
		X86 (32-Bit)	X64 (64-Bit)	PPC	ARM	MIPSII
Windows*	2017-08-30	2.12.28	2.12.28	-	-	-
Linux	-	-	-	-	-	-

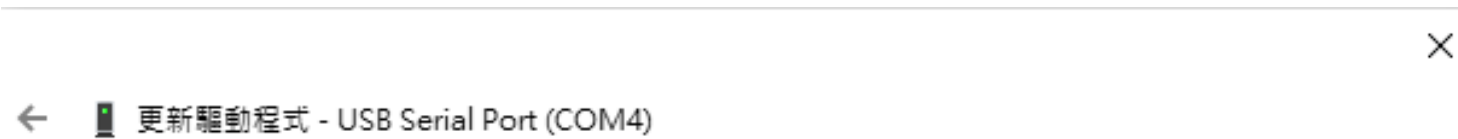
Troubleshooting – Update VCP driver

3. Click Other devices > USB Serial Port > Update driver

4. Choose “瀏覽電腦上的驅動程式”



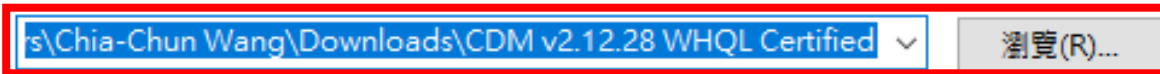
Troubleshooting – Update VCP driver



在您的電腦上瀏覽驅動程式

5. Choose the downloaded folder
(CDM v2.12.28 WHQL Certified)

在此位置搜尋驅動程式:

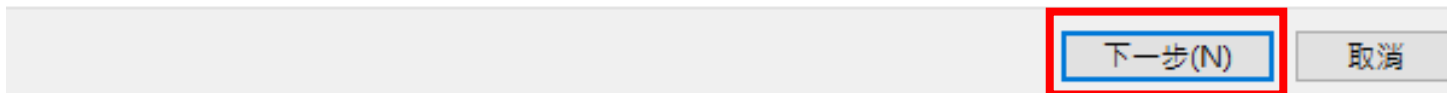


☒ 包含子資料夾(I)

6. Select

→ 讓我從電腦上的可用驅動程式清單中挑選(L)
此清單將會顯示與裝置相容的可用驅動程式，以及與裝置屬於同類別的所有驅動程式。

7.



Troubleshooting – Update VCP driver

8. Finish

