EEDG/CE 6370
Design and Analysis of Reconfigurable Systems
Homework 8
Hard Processor System (HPS)

1. Laboratory Objectives

- Learn how to install Linux on the DE1-SoC board
- Understand how to write programs for the HPS using the embedded system Design.
- Driving DE1-SoC peripherals from the HPS

2. Summary

In this lab you will learn to program the ARM processor (HPS) in the Cyclone V SoC FPGA being able to access different peripherals the board.

3. Pre-lab

- Review the lecture slides that cover embedded processors.
- Read the Terasic My_First_HPS.pdf

4. Tool and Hardware Requirements

- Intel Quartus Prime
- Intel Embedded Design Software (Standard Edition)
- DE1-SoC board
- microSD memory card for Linux operating system
  - Speed: Class 4 (at least)
  - Minimum 4GB of command-based Linux, 8GB for Desktop versions (**recommended**)

5. Installing Linux on DE1-SoC Board

- Terasic provides different Linux distributions for the DE1-SoC board (Terasic DE1-SoC → Resources)
  https://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=165&No=836&PartNo=4#contents
- The Ubuntu Desktop versions are complete Linux distributions with a GUI, while the others are console based. In this example I will use Linux Console with Framebuffer (command based Linux distribution)
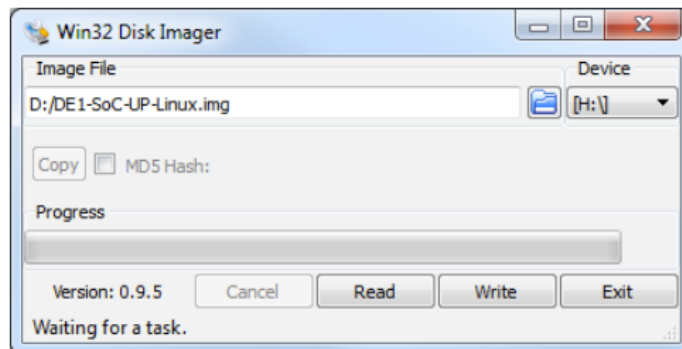
Linux BSP (Board Support Package): MicroSD Card Image

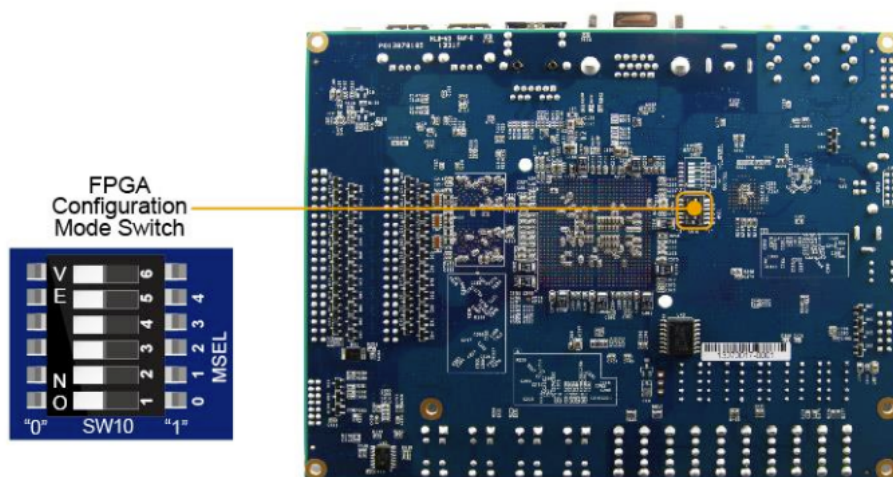| Title	Version	Size	Date	Download
Linux Ubuntu Desktop	1.0		2016-12-28	
Linux LXDE Desktop	1.0		2016-11-14	
Linux Ubuntu Desktop (*)	1.0		2015-06-26	
Linux Console with framebuffer	1.0		2014-03-24	
Linux LXDE Desktop (*)	1.0		2014-03-21	
Linux Console	1.0		2014-01-14	

- Download the Distribution that you want to use.
- Download 'win32 disk imager' software. This software allows to burn the Linux image downloaded onto a memory card.
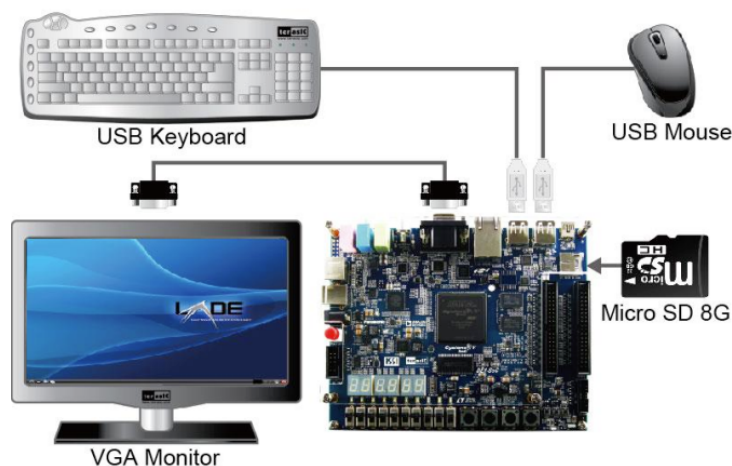
- Burn the selected Linux version onto the memory card as shown below. Specify the file and select the drive where the microSD card is inserted.

- To boot Linux on the DE1-SoC card → Insert microSD card into the FPGA board
- To boot LXDE version make sure HPS is boot from SD card MSEL[4:0]=00000 back of the board. Remember the original settings! (take a picture)

- Connect your USB keyboard to the USB connector on DE1-SoC board (and mouse if Desktop version)
- Connect your VGA monitor to the VGA connector (J9).
- Insert the microSD card with Linux image into the DE1-SoC board.
- Power up the DE1-SoC board.

- You should see two 'friendly' penguins on the VGA monitor when the Linux is booting.
- The default login is: 'root' (no password)
- Enjoy!

6. Installing ARM Programming Flow

To program the HPS (ARM) on the FPGA you need to install a cross compiler that can generate the machine code for the ARM processor on your PC. The following software programs and packages (for Windows) are needed:

1. Intel's Embedded Systems software and ARM's cross compiler
   Download Intel's Embedded Software and Tools for SoCs (Standard Edition)
   https://www.intel.com/content/www/us/en/software/programmable/soc-eds/overview.html#tab-blade-1-1
2. Cygwin (Linux emulator for windows)
3. MinGW
4. Linaro (contains ARM compiler)

The following instructions is a summary of the instructions given in Intel's SoC FPGA embedded development suite (SoC EDS) User guide.
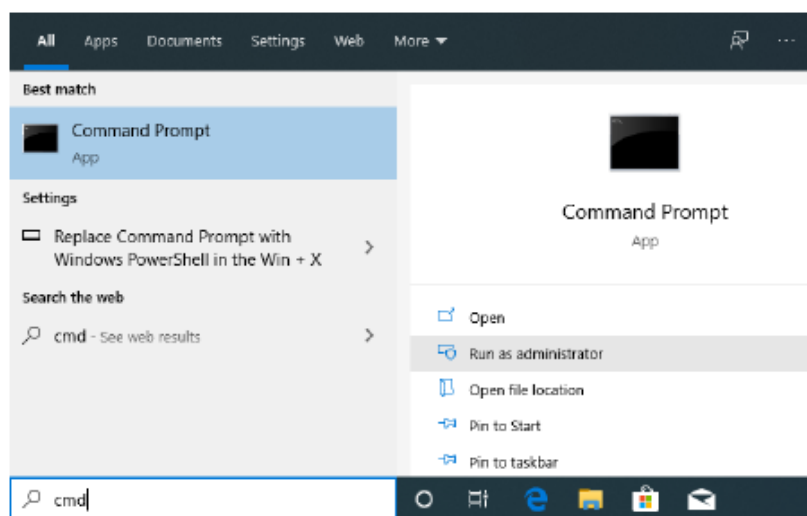
1. Installing Intel SoC EDS
- Download and run the SoC EDS installer from the *Intel SoC FPGA Embedded Development Suite Download Center for FPGAs* webpage.
  The current Intel Quartus Prime software version is 20.1 and the filename for the **standard edition** is SoCEDSSetup-20.1.0.771-windows.exe
- Run the installer and follow the default instructions
- In the **Select Components** dialog box, leave the **Intel Quartus Prime Programmer and Tools** selected (unless you already have full Intel Quartus Prime installed) and click **Next**.
- Install the software accepting all the default options including the drivers.
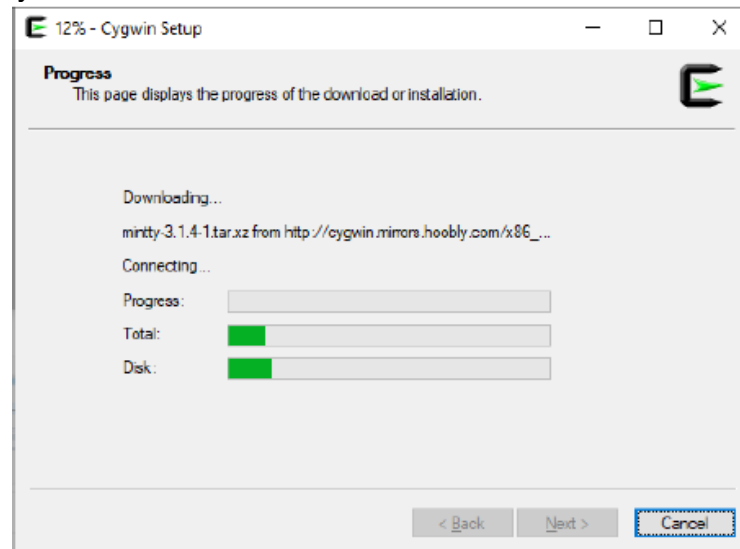
2. Installing Cygwin
Cygwin is a collection of open-source tools that allows Unix or Linux applications to be compiled and run on a Microsoft Windows.
- Download the Cygwin installer from https://cygwin.com/setup-x86_64.exe.
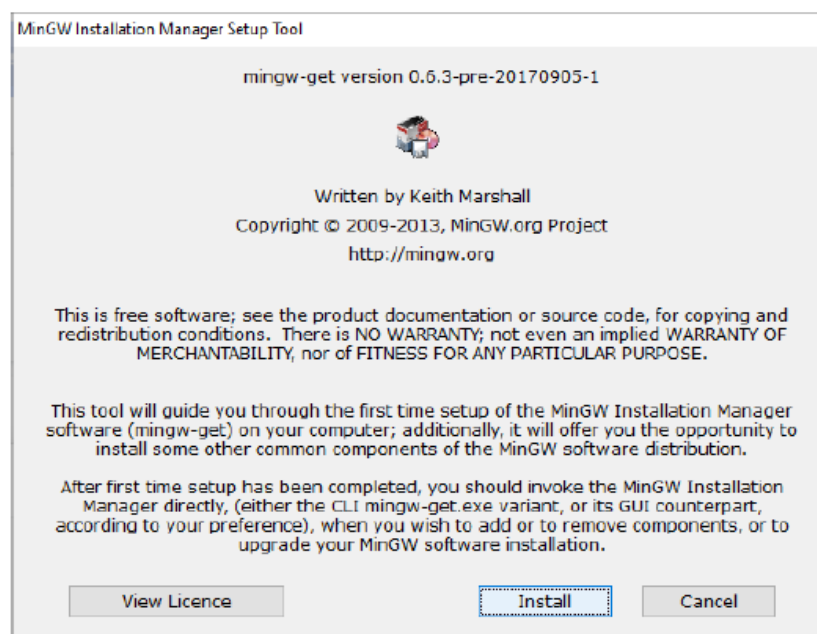- Start a command prompt as administrator

- Change the current directory to cygwin_setup in the SoC EDS software folder installed previously.
  For Intel Quartus Prime Standard Edition software version 20.1:
  cd c:\intelFPGA\20.1\embedded\cygwin_setup\
- Run the soceds-cygwin-setup.bat executable, passing it the full path to where you downloaded the setup-x86_64.exe installer (to make your life easy you can also copy the installer to the same directory as the Cygwin_setup folder
  soceds-cygwin-setup.bat %USERPROFILE%\Downloads\setup-x86_64.exe

- The installer application will start as follows installing all the required Cygwin packages automatically.

3. Installing MinGW
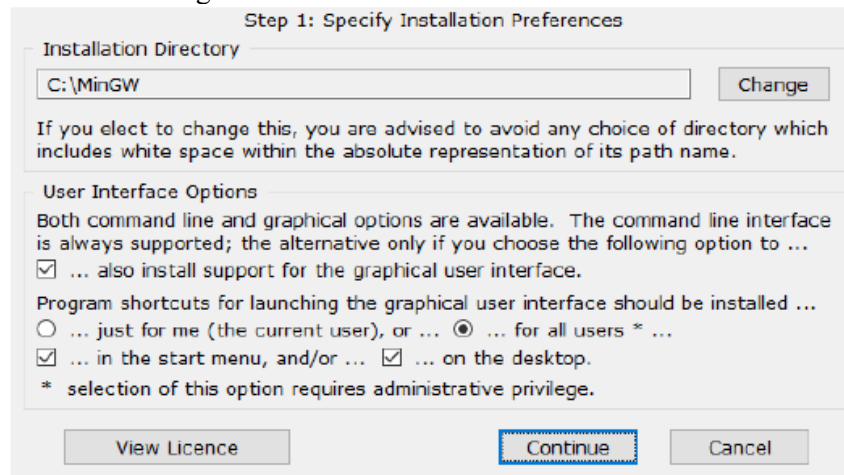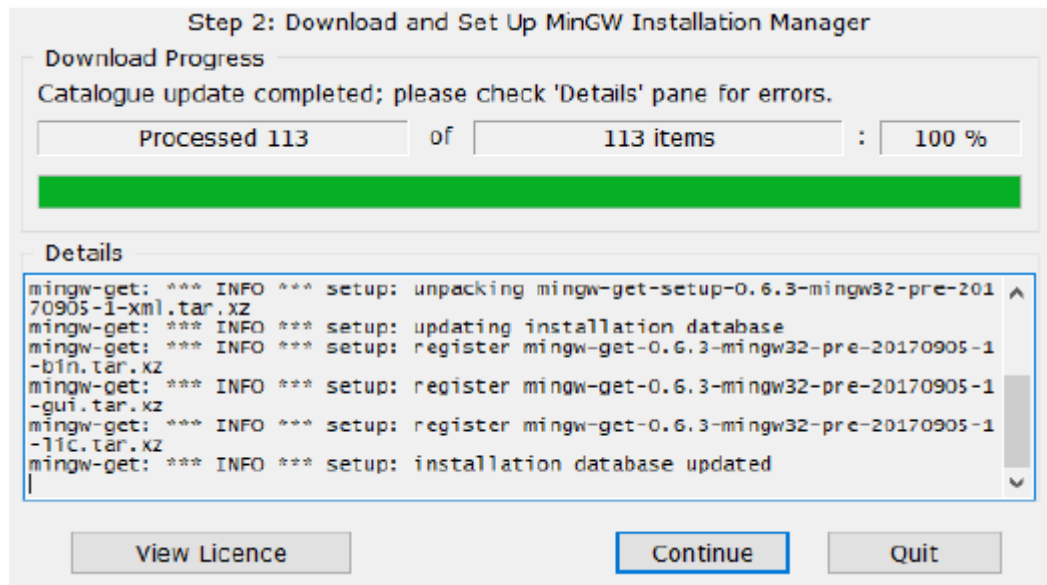MinGW allows to support the GCC compiler on windows systems.
- Go to the SoC EDS and ARM development studio web page on rocketboards.org and navigate to the "install MinGW" section. Download the MinGW installer to your computer and run as an administrator (right click on the installer → run as administrator).
- The MinGW installation Manager Setup tool dialog box will appear. Click Install
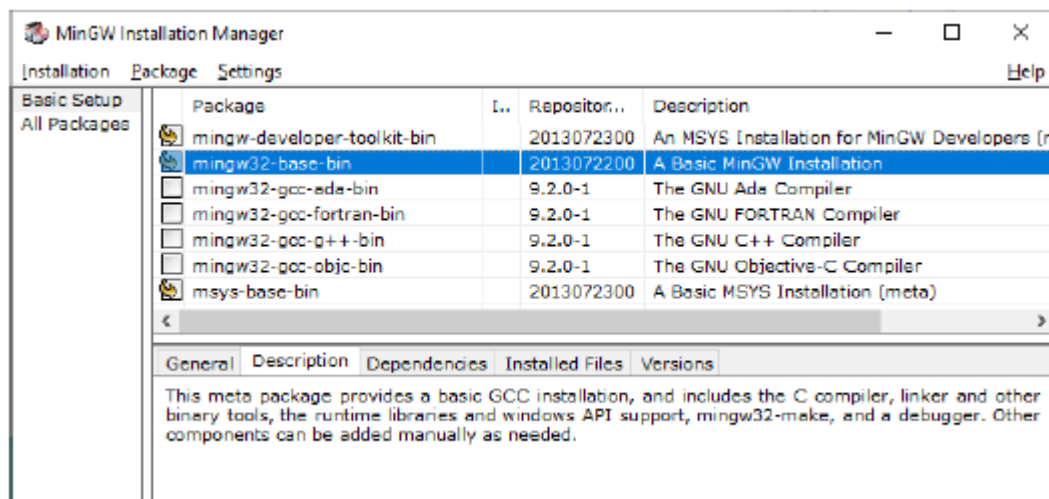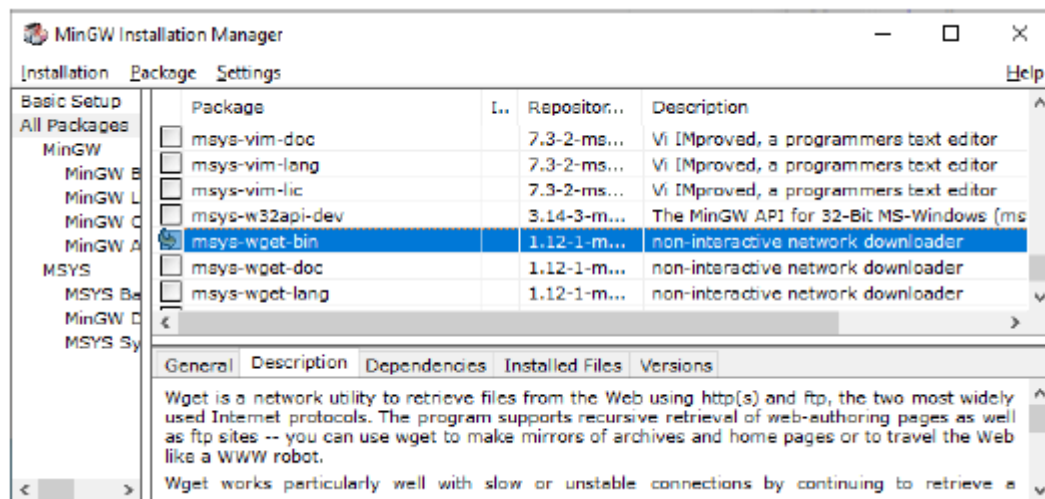
- Leave the default settings.

Step 1: Specify Installation Preferences

Installation Directory

| C:\MinGW | Change |

If you elect to change this, you are advised to avoid any choice of directory which includes white space within the absolute representation of its path name.

User Interface Options

Both command line and graphical options are available. The command line interface is always supported; the alternative only if you choose the following option to ...

☑ ... also install support for the graphical user interface.

Program shortcuts for launching the graphical user interface should be installed ...
○ ... just for me (the current user), or ... ◉ ... for all users * ...
☑ ... in the start menu, and/or ... ☑ ... on the desktop.
* selection of this option requires administrative privilege.

| View Licence | Continue | Cancel |

- Continue and download and setup MinGW installation manager.

Step 2: Download and Set Up MinGW Installation Manager

Download Progress

Catalogue update completed; please check 'Details' pane for errors.

| Processed 113 | of | 113 items | : | 100 % |

Details

```
mingw-get: *** INFO *** setup: unpacking mingw-get-setup-0.6.3-mingw32-pre-201
70905-1-xml.tar.xz
mingw-get: *** INFO *** setup: updating installation database
mingw-get: *** INFO *** setup: register mingw-get-0.6.3-mingw32-pre-20170905-1
-bin.tar.xz
mingw-get: *** INFO *** setup: register mingw-get-0.6.3-mingw32-pre-20170905-1
-gui.tar.xz
mingw-get: *** INFO *** setup: register mingw-get-0.6.3-mingw32-pre-20170905-1
-lic.tar.xz
mingw-get: *** INFO *** setup: installation database updated
|
```

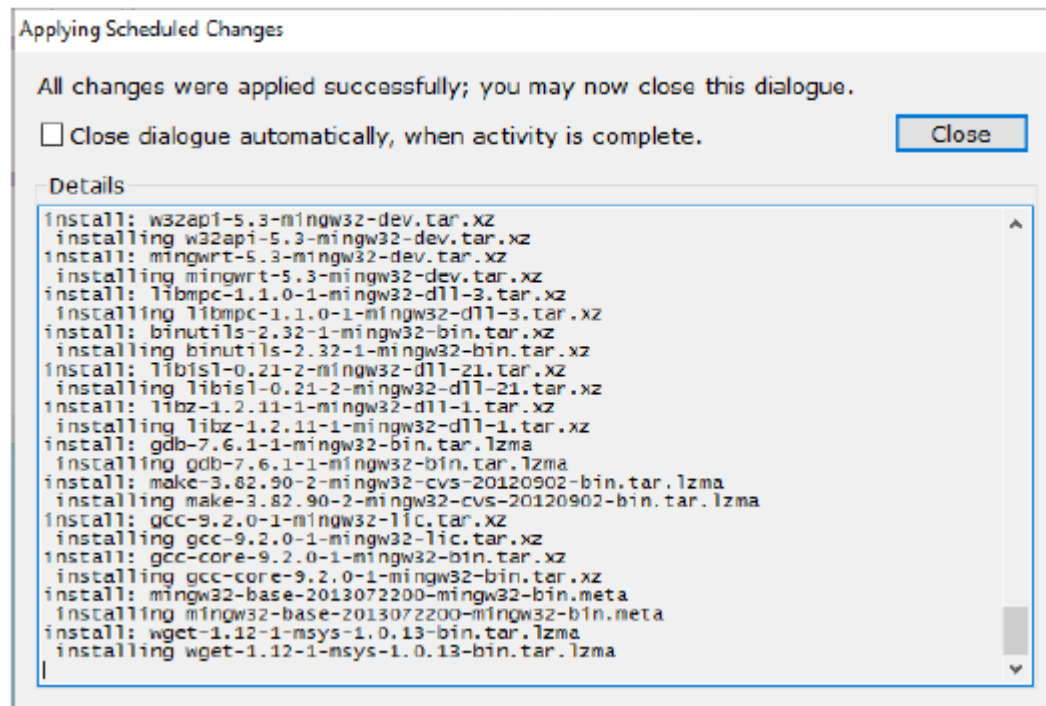| View Licence | Continue | Quit |

- In the Basic Setup view, click on mingw-developer-toolkit-bi\n and mingw32-base-bin and msys-base-bin and select Mark for installation.
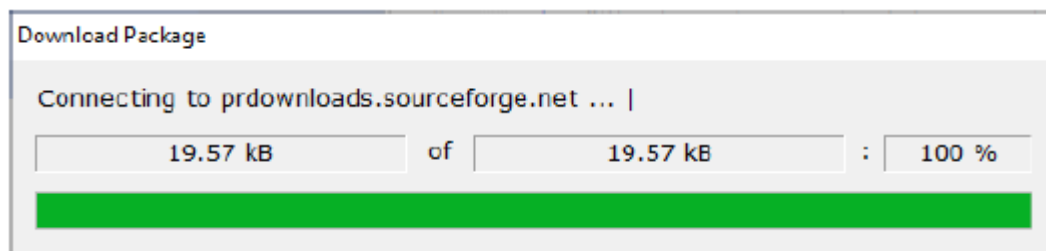
- In the All Packages view, clock on msys-wget-bin and select Mark for Installation

- Select Installation→ Apply Changes from the top menu.
- Click Apply to proceed.

Applying Scheduled Changes

All changes were applied successfully; you may now close this dialogue.

☐ Close dialogue automatically, when activity is complete.   [Close]

Details

```
install: w32api-5.3-mingw32-dev.tar.xz
 installing w32api-5.3-mingw32-dev.tar.xz
install: mingwrt-5.3-mingw32-dev.tar.xz
 installing mingwrt-5.3-mingw32-dev.tar.xz
install: libmpc-1.1.0-1-mingw32-dll-3.tar.xz
 installing libmpc-1.1.0-1-mingw32-dll-3.tar.xz
install: binutils-2.32-1-mingw32-bin.tar.xz
 installing binutils-2.32-1-mingw32-bin.tar.xz
install: libisl-0.21-2-mingw32-dll-21.tar.xz
 installing libisl-0.21-2-mingw32-dll-21.tar.xz
install: libz-1.2.11-1-mingw32-dll-1.tar.xz
 installing libz-1.2.11-1-mingw32-dll-1.tar.xz
install: gdb-7.6.1-1-mingw32-bin.tar.lzma
 installing gdb-7.6.1-1-mingw32-bin.tar.lzma
install: make-3.82.90-2-mingw32-cvs-20120902-bin.tar.lzma
 installing make-3.82.90-2-mingw32-cvs-20120902-bin.tar.lzma
install: gcc-9.2.0-1-mingw32-lic.tar.xz
 installing gcc-9.2.0-1-mingw32-lic.tar.xz
install: gcc-core-9.2.0-1-mingw32-bin.tar.xz
 installing gcc-core-9.2.0-1-mingw32-bin.tar.xz
install: mingw32-base-2013072200-mingw32-bin.meta
 installing mingw32-base-2013072200-mingw32-bin.meta
install: wget-1.12-1-msys-1.0.13-bin.tar.lzma
 installing wget-1.12-1-msys-1.0.13-bin.tar.lzma
```

- The installer will download all the necessary packages.

Download Package

Connecting to prdownloads.sourceforge.net ... |
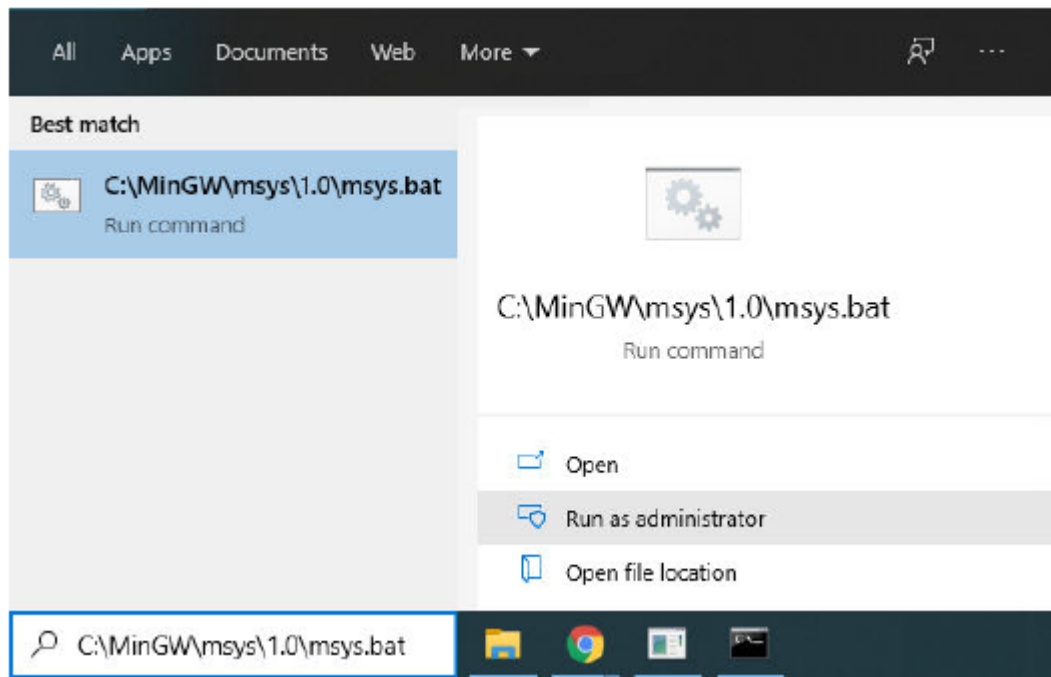
| 19.57 kB | of | 19.57 kB | : | 100 % |

- After the installer applied all the changes → click close
- Select Installation → quit from the top menu.

4. Installing Linaro Bare Metal Toolchain

Bare metal refers to a computer executing instructions directly on logic hardware (peripherals) without an intervening operating system (any of its drivers).

This section shows how to install the Linaro Bare Metal toolchain for Cortex-A9, required for compiling Bare metal programs for the Cyclone V SoC FPGA, **including the ARM cross-compiler.**
- Run C:\MinGW\mysys\1.0\mysys.bat as an administrator.

- In the Msys console go to the linaro folder:For Intel Quartus Prime Standard version 20.1
  ```
  cd c:/intelFPGA/20.1/embedded/host_tools/linaro
  ```

- With any text editor modify the installation script to point to the correct ARM compiler distribution. As shown below the GCCFILE variable is set to:
  GCCFILE=gcc-linaro-7.5.0-2019.12-i686-mingw32_arm-linux-gnueabihf
  and the package that contains this ARM compiler version is obtained through wget:
  wget http://releases.linaro.org/components/toolchain/binaries/7.5-2019.12/arm-linux-gnueabihf/${GCCFILE}.tar.xz -O ${GCCFILE}.tar.xz

```
pushd $INSTALL_DIR

if [ x$OS == "xWindows_NT" ]
then
#    GCCFILE=gcc-linaro-7.5.0-2019.12-i686-mingw32_arm-eabi
     GCCFILE=gcc-linaro-7.5.0-2019.12-i686-mingw32_arm-linux-gnueabihf
else
     GCCFILE=gcc-linaro-7.5.0-2019.12-i686_arm-eabi
fi

#if [ -f ${GCCFILE}.tar.xz ]

#then
#    echo Linaro toolchain tarball found, skipping download
#else
#    wget http://releases.linaro.org/components/toolchain/binaries/7.5-2019.12/arm-eabi/${GCCFILE}.tar.xz -O ${GCCFILE}.tar.xz
     wget http://releases.linaro.org/components/toolchain/binaries/7.5-2019.12/arm-linux-gnueabihf/${GCCFILE}.tar.xz -O ${GCCFILE}.tar.xz
#fi
```

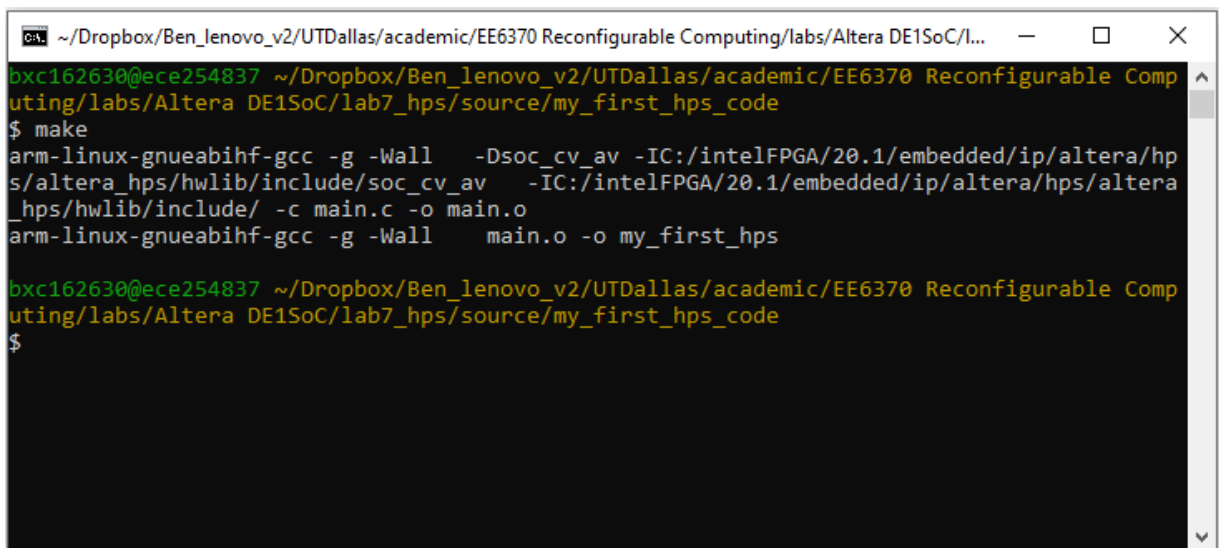- Search any additional arm-eabi and substitute by arm-linux-genuabihf

```
mkdir build
cd build
#../configure --target-arm-eabi --disable-newlib-supplied-syscalls --disable-multilib
../configure --target-arm-linux-gnueabihf --disable-newlib-supplied-syscalls --disable-multilib
make all -j16
make install DESTDIR=$INSTALL_DIR/newlib
popd
```

- Run the installer from the Msys console:
  ./install_linaro.sh
- Upon successful completion, the following are installed in the linaro folder:
  o gcc–GNU Compiler
  o newlib–Newlib library

7. Writing, Compiling, Transfer and Executing Code on DE1-SoC Board

Compiling

- Download the my_first_hps_code.zip file from elearning and unzip.
- This program basically prints a message. The folder contains:
  - main.c (program to be compiled)
  - Makefile (instructions of this program will be compiled including the cross-compiler chosen, which should match the one downloaded from Linaro.

- Open the embedded_command_shell.bat found in:
  C:\intelFPGA\20.1\embedded

- Change directory to where you have unzipped the my_first_hps_code folder
- Compile the code by typing: $make

- This will generate a my_first_hps executable program that can only be executed on an ARM processor.
- To recompile the program after any changes type: $make clean → $make

Transferring to DE1-SoC board

- Copy the executable just generated (my_first_hps_code) to a USB thumb drive.
- Plug the USB thumb drive in one of the USB ports of the DE1-SoC board (not the one where the keyboards it connected as you will need to keyboard to execute the code)
- To access the files in Linux you need to mount the drive.
  - (i) Detect the USB drive:
    $fdisk -l
  - (ii) Create a mount point:
    $mkdir /media/usb-drive
  - (iii) Mount USB drive:
    $mount /dev/sda1 /media/usb-drive (where dev/sda1 depends on where the USB drive is given by fdisk
  - (iv) Accessing data:
    $cd /media/usb-drive or copy the file to your home directory.
  - (v) Unmounting the drive:
    $umount /media/usb-drive (do this once you do not need to access the data on the USB anymore.

- Execute the compiled program: $./my_first_hps
- If nothing happens check the permissions and change it to make it executable:$chmod +x my_first_hps

Congratulations, you should now know how to program and transfer any application on the Cyclone V SoC HPS !

NOTE 1: You can also connect to the DE1-SoC board through its UART. For this, following the instructions in the DE1-SoC Getting Started Guide → Setting up UART Terminal section.

NOTE 2:You can also connect the FPGA board to an ethernet router through an RJ45 cable and assign the board an IP and transfer files between the PC and the DE1-SoC Board through 'scp'

1. Login as a root user
2. Type 'udhcpc' to query an IP from the DHCP server
3. Type 'ifconfig' to check the Ethernet IP for your DE1-SoC board.

UT Dallas Honor Code

UT Dallas values academic integrity. Therefore, all students must understand the meaning and consequences of cheating, plagiarism and other academic offences under the Code of Student Conduct and Disciplinary Procedures.

Group assignments must be completed solely by the members of the group. Cross-group work is not allowed. Moreover, similar assignments have been offered before at UT Dallas and other universities. Any use of information from previous assignments is prohibited. The tutorials are to be taken individually. Failure to respect this rule constitutes dishonesty and is a direct violation of the University Honor Code.

[END]