



THE IMAGINATION UNIVERSITY PROGRAMME

RVfpga-SoC

설치 가이드

목차

소개	3
Lab 1 설치	4
Lab 2 설치	6
Lab 3 설치	9
Lab 4 설치	10
부록 A: PlatformIO 를 사용하기 위해 Windows 에 드라이버 설치	12
부록 B: Windows 에 Verilator 및 GTKWave 설치	14



1. 소개

이 가이드는 Ubuntu 18.04 운영 체제(OS)에서 RVfpga-SoC에 필요한 도구 및 하드웨어를 설치하는 방법을 보여줍니다. 아래 지침은 Ubuntu 18.04 OS에 대한 것이지만 다른 Linux 운영 체제 및 Windows는 유사한(정확히 동일하지는 않더라도) 단계를 따릅니다. 경우에 따라 Windows OS에 대한 특정 지침이 있는 알림 내용을 (box 형태로) 삽입합니다. Ubuntu를 사용하는 경우 해당 알림 내용을 (box) 무시하십시오.

이 프로세스는 몇 시간(또는 다운로드 속도에 따라 그 이상)이 소요될 수 있지만 대부분의 시간은 프로그램을 다운로드하고 설치하는 동안 대기하는 시간입니다.

표 1에는 RVfpga-SoC에 필요한 소프트웨어와 하드웨어가 나와 있습니다.

표 1. RVfpga-SoC에 필요한 소프트웨어 및 하드웨어

Software		
Name	Website	Cost
Vivado 2019.2 WebPACK	https://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/vivado-design-tools/2019-2.html	무료
VSCode	https://code.visualstudio.com/Download	무료
PlatformIO	https://platformio.org/ VSCode와 같이 설치	무료
Verilator(HDL Simulator) 및 GTKWave	https://github.com/verilator/verilator http://gtkwave.sourceforge.net/	무료
FuseSoC	https://github.com/olofk/fusesoc	무료
RISC-V Toolchain 및 OpenOCD	https://github.com/riscv/riscv-gnu-toolchain https://github.com/riscv/riscv-openocd PlatformIO와 같이 설치	무료
Zephyr Project	https://github.com/zephyrproject-rtos/zephyr	무료
Hardware*		
Name	Website	Cost
Nexys A7 FPGA Board*	https://store.digilentinc.com/nexys-a7-fpga-trainer-board-recommended-for-ece-curriculum/	\$265 (교육할인: \$199)
RISC-V Core 및 System-on-Chip (SoC)**		
Name	Website	Cost
Western Digital's SweRV EH1 Core	https://github.com/chipsalliance/Cores-SweRV	무료
SweRVolf	https://github.com/chipsalliance/Cores-SweRVolf	무료

* 하드웨어는 선택 사항입니다.

** SweRV EH1 코어 및 SweRVolf는 RVfpga-SoC 패키지의 일부로 제공됩니다.

2. Lab 1 설치

이 섹션에서는 RVfpga-SoC 과정의 Lab 1 을 수행하는 데 필요한 소프트웨어를 설치하는 방법을 설명합니다.

1. Vivado 설치:

Vivado 는 Verilog 코드를 보고, 수정하고, 합성하기 위한 Xilinx 도구입니다. 이후 Lab 에서 광범위하게 사용할 것입니다. 설치 지침은 <https://reference.digilentinc.com/vivado/installing-vivado/start> 에서 확인할 수 있으며 아래에 요약되어 있습니다.

Windows: 위에서 참조한 웹 페이지(<https://reference.digilentinc.com/vivado/installing-vivado/start>)에도 Windows 에 Vivado 를 설치하기 위한 자세한 지침이 포함되어 있습니다. Windows 에 대한 특정 지침이 필요한 경우 아래에 알림 내용을 삽입합니다.

1 단계. <https://reference.digilentinc.com/vivado/installing-vivado/start> 로 이동합니다

2 단계. Xilinx 다운로드 페이지로 이동합니다. <https://www.xilinx.com/support/download.html>

Step 3. "자동 압축 풀기 웹 설치 프로그램"을 설치하는 것이 좋습니다. 이 문서를 작성하는 시점에서 다운로드 페이지의 링크인 [Xilinx Unified Installer 2019.2: Linux Self Extracting Web Installer](#) 에 있습니다.

WINDOWS: 이 문서를 작성하는 시점에서 Windows 용 "자동 압축 풀기 웹 설치 프로그램"은 다운로드 페이지의 링크에 있습니다. [Xilinx Unified Installer 2019.2: Windows Self Extracting Web Installer](#)

4 단계. 설치 프로그램을 다운로드하기 전에 Xilinx 계정에 로그인하라는 메시지가 표시됩니다. 아직 계정이 없으면 계정을 만들어야 합니다.

5 단계. 바이너리 파일을 실행합니다. 터미널을 열고 root 로 만듭니다("sudo su" 입력). 그런 다음 바이너리 파일(Xilinx_Unified_2019.2_1106_2127_Lin64.bin)을 터미널로 드래그 합니다. 파일을 실행 가능하게 만들고 실행할 것인지 묻는 메시지가 나타나면 확인을 선택합니다.

문제 해결: 터미널에 권한이 거부되었다고 표시되면 터미널에 다음을 입력합니다(바이너리 파일과 동일한 디렉터리에 있음):

- `sudo chmod +x ./Xilinx_Unified_2019.2_1106_2127_Lin64.bin`
- `sudo ./Xilinx_Unified_2019.2_1106_2127_Lin64.bin`

WINDOWS: Windows 에서는 3 단계와 4 단계에서 다운로드한 .exe 파일을 더블 클릭하여 간단히 실행할 수 있습니다.

6 단계. Vivado 설치 프로그램이 설치 과정을 안내합니다. 중요 참고 사항:

- 설치할 제품으로 **Vivado** (*not* Vitis)를 선택합니다.

- Vivado HL **Webpack** 선택 (*not* Vivado HL System Edition); Webpack 은 무료입니다.
- 그렇지 않으면 기본값을 선택해야 합니다.

Hint: Vivado 의 설치 디렉토리를 변경하는 경우 다음 단계에서 경로를 적절하게 수정해야 합니다.

WINDOWS: Windows 에서는 7 단계와 8 단계가 필요하지 않습니다. 이 두 단계를 무시하고 9 단계로 바로 이동할 수 있습니다.

7 단계. Vivado 가 설치되면 환경을 설정해야 합니다. 터미널을 열고 다음을 입력합니다.

```
> source /tools/Xilinx/Vivado2019.2/settings64.sh
```

터미널을 시작할 때마다 실행되도록 해당 라인 (source /tools/Xilinx/Vivado2019.2/settings64.sh) 을 ~/.bashrc 파일에 추가하십시오.

8 단계. 터미널에 다음을 입력하여 Vivado 를 테스트합니다.

```
> vivado
```

문제 해결:

- 시스템에서 해당 실행 파일을 찾을 수 없으면 경로에 다음을 추가해야 합니다:

```
/tools/Xilinx/DocNav  
/tools/Xilinx/Vivado/2019.2/bin
```

- "application-specific initialization failed..."와 같은 오류가 발생하면 터미널에서 다음을 입력하십시오.

```
> sudo ln -s /lib/x86_64-linux-gnu/libtinfo.so.6  
/lib/x86_64-linux-gnu/libtinfo.so.5
```

2. 케이블 드라이버 설치:

9 단계. Nexys A7 FPGA 보드의 케이블 드라이버를 수동으로 설치해야 합니다.

터미널 창에서 다음을 입력합니다:

```
cd  
/tools/Xilinx/Vivado/2019.2/data/xicom/cable_drivers/lin64/i  
ninstall_script/install_drivers/  
  
sudo ./install_drivers
```

WINDOWS: Windows 에서 Vivado 를 설치하면 PlatformIO 와 호환되지 않는 Nexys A7 보드용 드라이버가 자동으로 설치됩니다. 따라서 **Windows** 를 사용하는 경우 설치 안내서의 부록 A 에 설명된 대로 드라이버를 업데이트해야 합니다.

3. Digilent 보드 파일 설치:

또한 Digilent 보드 파일을 수동으로 설치해야 합니다.

10 단계. Github 저장소에서 vivado-boards [archive](#) 를 다운로드하고 압축을 풉니다.

11 단계. archive 에서 압축을 푼 폴더를 열고 *new/board_files* 디렉토리로 이동합니다. 이 디렉토리 내의 모든 폴더를 선택하고 복사하십시오.

12 단계. Vivado 가 설치된 폴더를 엽니다(*/tools/Xilinx/Vivado* by default). 이 폴더 아래에서 *<version>/data/boards/board_files* 디렉토리로 이동한 다음 보드 파일을 이 디렉토리에 붙여 넣습니다.

13 단계. *new/board_files* 디렉토리로 이동하여 다음을 입력하여 터미널을 사용할 수도 있습니다.

```
> sudo cp -r *  
/tools/Xilinx/Vivado/2019.2/data/boards/board_files
```

WINDOWS: 10 단계에서 설명한대로 다운로드한 폴더를 복사 / 붙여 넣기 합니다. Windows 에서는 *C:\Xilinx\Vivado\2019.2\data\boards\board_files* 에서 Vivado 의 *board_files* 폴더를 찾을 수 있습니다.

3. Lab 2 설치

이 섹션에서는 RVfpga-SoC 과정의 Lab 2 를 수행하는 데 필요한 소프트웨어를 설치하는 방법을 보여줍니다. 여기에는 Visual Studio Code(VSCoDe), PlatformIO, Verilator 및 GTKWave 와 같은 소프트웨어 설치 지침이 포함되어 있습니다.

1. VSCoDe 설치:

VSCoDe 를 설치하려면 다음 단계를 따르세요.

1 단계. 다음 링크에서 .deb 파일을 다운로드 합니다: <https://code.visualstudio.com/Download>

2 단계. 터미널을 열고 터미널에 다음을 입력하여 VSCoDe 를 설치하고 실행합니다:

```
> cd ~/Downloads  
> sudo dpkg -i code*.deb  
> code
```

Windows: VSCoDe 패키지는 <https://code.visualstudio.com/Download> 에서 Windows(.exe 파일)용으로도 사용할 수 있습니다. 이러한 운영 체제에서 응용 프로그램을 설치하고 실행하는 데 사용되는 일반적인 단계를 따르십시오

2. VSCoDe 위에 PlatformIO 를 설치합니다:

PlatformIO 는 Microsoft 의 Visual Studio(VS) Code 를 기반으로 구축된 임베디드 시스템용 통합 개발 환경(IDE)입니다. 이를 통해 C 또는 어셈블리를 사용하여 RISC-V 프로세서(FPGA 에 있음)를 프로그래밍할 수 있습니다. PlatformIO 는 크로스 플랫폼이며 내장된 디버거를 포함합니다.

PlatformIO 를 설치하려면 다음 단계를 따르십시오:

3 단계. 터미널에 다음을 입력하여 python3 유틸리티를 설치합니다.

```
> sudo apt install -y python3-distutils python3-venv
```

Windows: 이 3 단계는 Windows 에서 필요하지 않습니다.

4 단계. 아직 열리지 않은 경우 시작 버튼을 선택하고 검색 메뉴에서 "VSCode"를 입력하여 VSCode 를 시작한 다음, VSCode 를 선택하거나 터미널에 코드를 입력하여 시작합니다.


5 단계. VSCode 에서 VSCode 의 왼쪽 사이드바에 있는 Extensions 아이콘  을 클릭합니다(그림 1 참조)



그림 1. VSCode's Extensions icon

6 단계. 검색 상자에 *PlatformIO* 를 입력하고 옆에 있는 설치 버튼을 클릭하여 PlatformIO IDE 를 설치합니다(그림 2 참조).

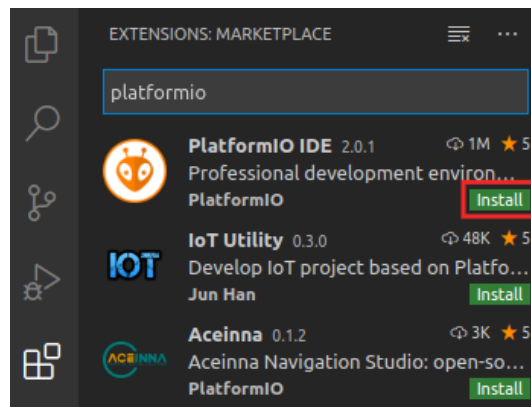


그림 2. PlatformIO IDE Extension

Step 7. 하단의 OUTPUT 창에서 설치 과정을 알려줍니다. 완료되면 오른쪽 하단 창에서 "Reload Now"를 클릭하면 PlatformIO가 VSCode 내부에 설치됩니다(그림 3 참조).

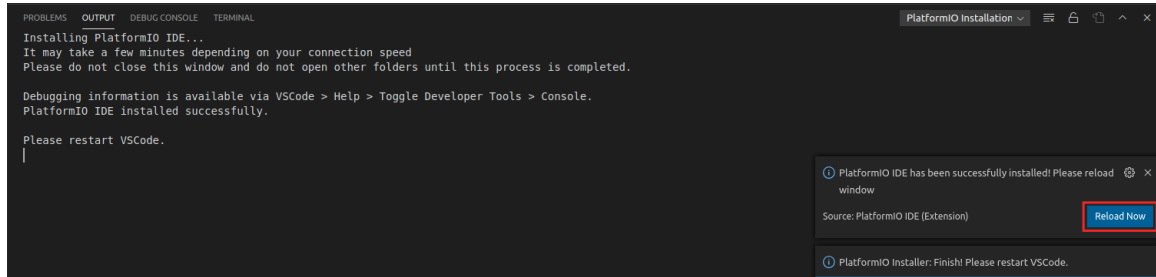


그림 3. PlatformIO 설치 후 Reload Now

3. GTKWave 설치:

Ubuntu 18.04 Linux 시스템에 GTKWave를 설치하려면 다음 단계를 따르십시오.

Ubuntu 터미널을 열고 다음 명령을 입력하십시오.

- `sudo apt-get install git make autoconf g++ flex bison libfl2 libfl-dev`
- `sudo apt-get install -y gtkwave`

Windows: Windows용 GTKWave 설치에 대해서는 부록 B를 참조하십시오.

4. Verilator 설치:

Verilator를 설치하려면 다음 단계를 따르십시오. (지침은

<https://www.veripool.org/projects/verilator/wiki/Installing>에서 볼 수 있지만 아래에 요약되어 있습니다)

- `git clone https://git.veripool.org/git/verilator`
- `cd verilator`
- `git pull`
- `git checkout v4.106`
- `autoconf`
- `./configure`
- `make` (또는 `make -j$(nproc)`를 사용하여 더 빠르게 만들 수 있습니다)
- `sudo make install`
- `export PATH=$PATH:/usr/local/bin` (시스템의 경로 변경)

`/usr/local/bin`을 경로에 영구적으로 추가하려면 `~/.bashrc` 파일에 마지막 줄을 추가하십시오.

Windows: Verilator for Windows 설치에 대해서는 부록 B를 참조하십시오.

4. Lab 3 설치

Lab 3 이후의 설치 지침은 특히 Ubuntu 18.04 운영 체제용입니다. Windows 10 사용자는 [Windows Subsystem for Linux](#)(Linux 용 Windows 하위 시스템)을 사용하여 Lab 의 시뮬레이션 부분을 실행할 수 있습니다. 이 지침은 다른 최신 버전의 Ubuntu 에서도 동일하게 작동합니다. Ubuntu 운영 체제에서 다음 Lab 을 모두 수행하는 것이 좋습니다.

이 섹션에서는 RVfpga-SoC 과정의 Lab 3 을 수행하는 데 필요한 소프트웨어를 설치하는 방법을 보여줍니다.

1. pip 설치:

"FuseSoC" 및 "west" 설치에는 "pip"가 필요합니다. Ubuntu 터미널을 열고 다음 명령을 입력하십시오.

```
> sudo apt install python3-pip
```

2. pyelftools 설치:

west 빌드에는 pyelftools 가 필요합니다. 2 가지 방법을 사용하여 pyelftools 를 설치할 수 있습니다.

```
> pip3 install pyelftools  
또는,
```

```
> sudo apt-get install -y python3-pyelftools python-pyelftools
```

3. FuseSoC 설치:

현재 안정적인 버전의 FuseSoC 를 설치하려면 터미널 창을 열고 다음 명령을 실행합니다. 시스템에 이전 버전의 FuseSoC 가 있는 경우 해당 버전을 안정적인 최신 릴리스로 업그레이드합니다.

```
> sudo pip3 install --upgrade fusesoc
```

4. OpenOCD 설치:

OpenOCD 는 사용자가 임베디드 대상 장치를 프로그래밍하고 디버그할 수 있는 개방형 온칩 디버거입니다. 컴퓨터에 RISC-V OpenOCD 를 설치하려면 다음 단계를 따르십시오.

1 단계. "apt-get"을 사용하여 필요한 종속성을 설치합니다.

```
> sudo apt-get install libusb-1.*  
> sudo apt-get install pkg-config
```

2 단계. riscv-openocd github 저장소 복제:

- `git clone https://github.com/riscv/riscv-openocd.git`
- `cd riscv-openocd`
- `./bootstrap`

참고: 명령을 찾을 수 없다는 오류가 발생하면 다른 종속성을 가지는 다음 명령을 다운로드하여 실행해 보십시오.

```
sudo apt-get install libtool
```

3 단계. 사용자 공간 USB 프로그래밍 라이브러리 개발 파일을 다운로드하여 설치합니다.

- `sudo apt-get install libusb-1.0-0-dev`

4 단계. OpenOCD 가 연결할 수 있는 JTAG 서버 구성

- `./configure --enable-jtag_vpi --enable-ftdi`
- `make`
- `sudo make install`

5. Lab 4 설치

이 섹션에서는 RVfpga-SoC 과정의 Lab 4 를 수행하는 데 필요한 소프트웨어를 설치하는 방법을 보여줍니다.

Ubuntu 터미널을 열고 다음 명령을 입력하십시오.

1. 1. 전제 조건 및 종속성:

1 단계. 다음을 사용하여 저장소를(repositories) 업데이트합니다:

- `sudo apt update`
- `sudo apt upgrade`

2 단계. "apt"를 사용하여 필요한 종속성 설치:

- `sudo apt install --no-install-recommends git cmake
ninja-build gperf \`
- `ccache dfu-util device-tree-compiler wget \`
- `python3-dev python3-pip python3-setuptools python3-tk
python3-wheel xz-utils file \`
- `make gcc gcc-multilib g++-multilib libsdl2-dev`

2. *cmake* 버전을 3.13.1 이상으로 업데이트:

다음 명령어를 이용하여 [kitware third-party apt repository \(저장소\)](#) 추가 지침에 따라

apt 를 사용하여 업데이트된 버전의 cmake 를 얻을 수 있습니다.

3 단계. 다음 명령을 입력하여 cmake 를 다운로드 및 설치/업데이트합니다.

```
> wget -O - https://apt.kitware.com/keys/kitware-
archive-latest.asc 2>/dev/null |
sudo apt-key add -
> sudo apt-add-repository 'deb
https://apt.kitware.com/ubuntu/ bionic main'
> sudo apt update
> sudo apt install cmake
```

3. west 설치:

4 단계. west 를 설치하고 ~/.local/bin 이 PATH 환경 변수에 있는지 확인합니다.

```
> pip3 install --user -U west
> echo 'export PATH=~/.local/bin:"$PATH"' >> ~/.bashrc
> source ~/.bashrc
```

4. Zephyr SDK 설치:

Zephyr SDK(소프트웨어 개발 키트)에는 Zephyr 에서 지원하는 각 아키텍처에 대한 툴체인이 포함되어 있습니다. 또한 사용자 지정 QEMU 바이너리 및 호스트 컴파일러와 같은 추가 호스트 도구가 포함되어 있습니다.

Zephyr SDK 버전 0.12.4 를 설치합니다.

홈(home) 디렉토리에 다음 명령을 입력하십시오.

5 단계 0.12.4 버전 SDK 설치 프로그램을 다운로드합니다([0.12.4 version SDK installer](#)):

```
> wget https://github.com/zephyrproject-rtos/sdk-
ng/releases/download/v0.12.4/zephyr-sdk-0.12.4-x86_64-linux-
setup.run
```

6 단계. 설치 프로그램을 실행하고 ~/zephyr-sdk-0.12.4 에 SDK 를 설치합니다:

```
> chmod +x zephyr-sdk-0.12.4-x86_64-linux-setup.run
> ./zephyr-sdk-0.12.4-x86_64-linux-setup.run -- -d ~/zephyr-
sdk-0.12.4
```

해당 위치 외부에 Zephyr SDK 를 설치하는 경우 Zephyr SDK(소프트웨어 개발 키트) 설치를 읽으십시오: [Install the Zephyr Software Development Kit \(SDK\)](#). SDK 디렉토리를 설치한 후에는 이동할 수 없습니다.

5. PuTTY 설치 :

7 단계. 다음 명령을 입력하여 PuTTY 를 설치합니다.

```
> sudo apt-get install -y putty
```

6. 부록 A: PlatformIO 를 사용하기 위해 Windows 에 드라이버 설치

Zadig 실행 파일을 다운로드하려면 다음 웹 사이트로 이동하십시오(그림 4 참조).

<https://zadig.akeo.ie/>

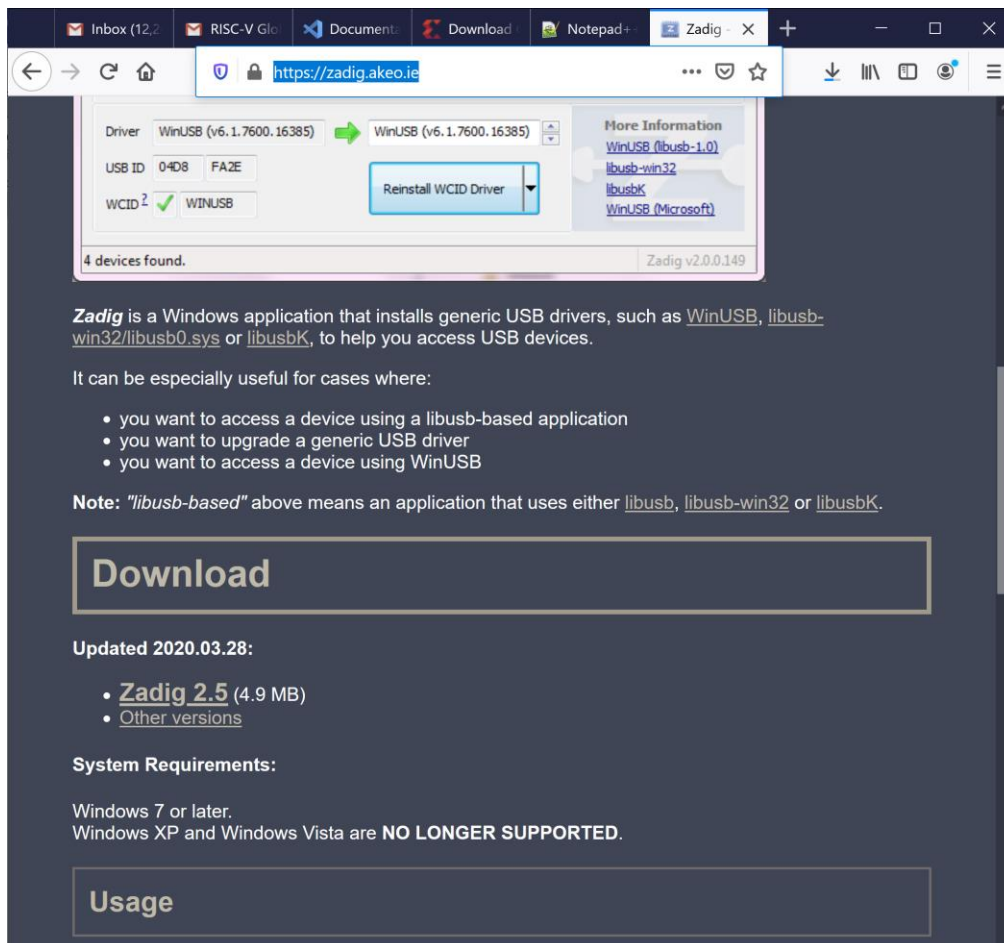


그림 4. PlatformIO 에서 사용하는 Nexys A7 보드 드라이버 설치

Zadig 2.5 를 클릭하고 실행 파일을 저장하십시오. 그런 다음 다운로드한 위치에 있는 zadig-2.5.exe 을 실행합니다. 시작 메뉴에 zadig 를 입력하여 찾을 수도 있습니다. Zadig 가 컴퓨터를 변경하도록 허용할 것 인지 와 업데이트를 확인하도록 허용할 것인지 묻는 메시지가 표시될 것입니다. YES 를 두 번 클릭합니다.

Nexys A7 보드를 컴퓨터에 연결하고 전원을 켭니다. Zadig 에서 옵션 → List All Devices 를 클릭합니다(그림 5 참조).

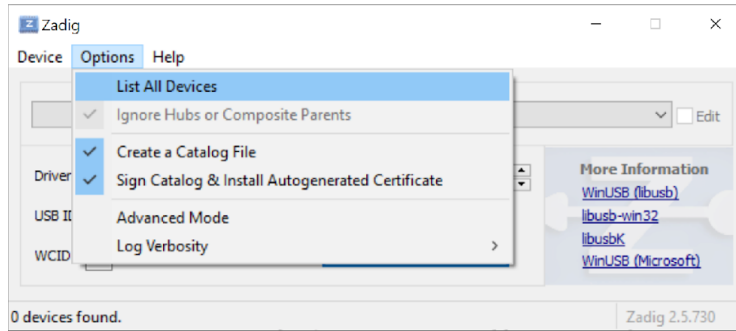


그림 5. Zadig 의 모든 장치 나열

드롭-다운 메뉴를 클릭하면 Digilent USB 장치(인터페이스 0) 및 Digilent USB 장치(인터페이스 1)가 나열됩니다. Digilent USB 장치(인터페이스 0) 전용 새로운 드라이버를 설치합니다(그림 6 참조).

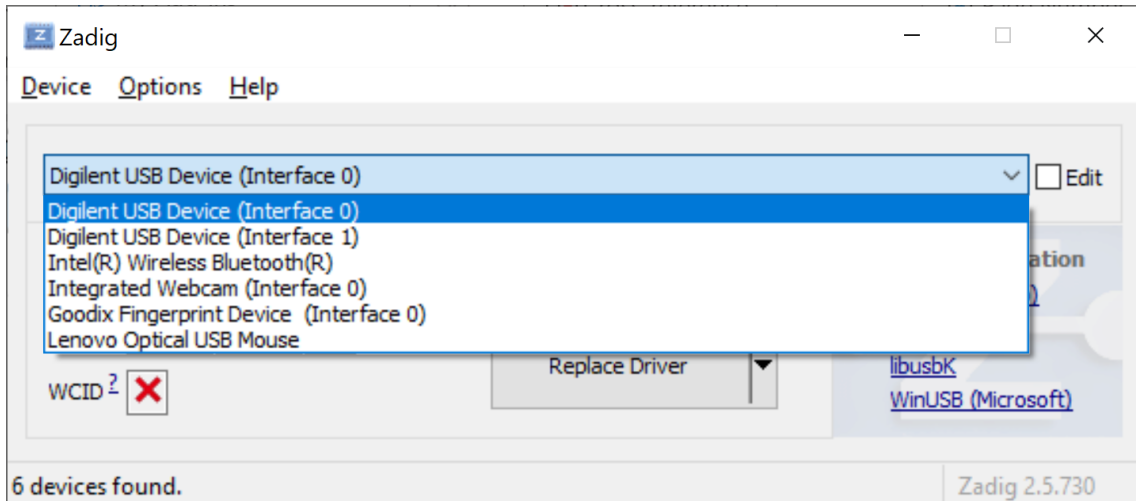


그림 6. Digilent USB 장치용 WinUSB 드라이버 설치(인터페이스 0)

이제 그림 7 과 같이 FTDI 드라이버를 WinUSB 드라이버로 교체합니다. Digilent USB 장치(인터페이스 0)의 드라이버 교체(또는 드라이버 설치)를 클릭합니다. Nexys A7 보드용 드라이버를 설치하거나 이전에 Vivado 를 설치한 경우, Vivado 에서 사용하는 FTDI 드라이버를 PlatformIO 에서 사용하는 WinUSB 드라이버로 교체합니다.

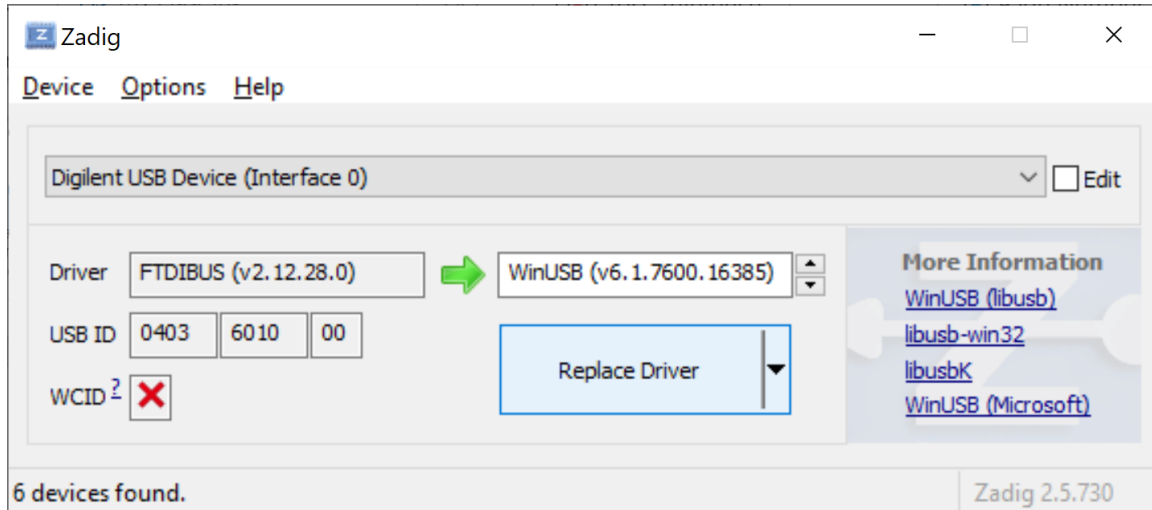


그림 7. Nexys A7 보드용 드라이버 교체

일정 시간(일반적으로 몇 분) 후에 Zadig 는 드라이버가 올바르게 설치되었음을 나타냅니다. 닫기를 클릭한 다음 Zadig 창을 닫습니다.

다음에 PlatformIO 를 사용할 때 드라이버를 다시 설치할 필요가 없습니다. 그러나 이 드라이버는 Windows 의 Vivado 와 호환되지 않습니다.

7. 부록 B: Windows 에 Verilator 및 GTKWave 설치

이 섹션에서는 Windows 10 에서 Verilator 와 GTKWave 를 설치하는 방법을 설명합니다. Windows 에서 Verilator 를 설치하려면 Cygwin 을 사용해야 하므로 먼저 이 프로그래밍/런타임 환경을 설치하는 방법을 설명합니다.

1. Cygwin 설치:

웹 페이지(<https://www.cygwin.com>), 에 설명된 대로 Cygwin 은 Linux 배포판과 유사한 Windows 기능을 제공하는 GNU 및 오픈 소스 도구로 구성됩니다. 다음 단계에 따라 Windows 10 에 Cygwin 을 설치하십시오.

1. 설치 웹페이지(<https://cygwin.com/install.html>)로 이동하여 *setup-x86_64.exe* 라는 설치 파일을 다운로드합니다(그림 8).

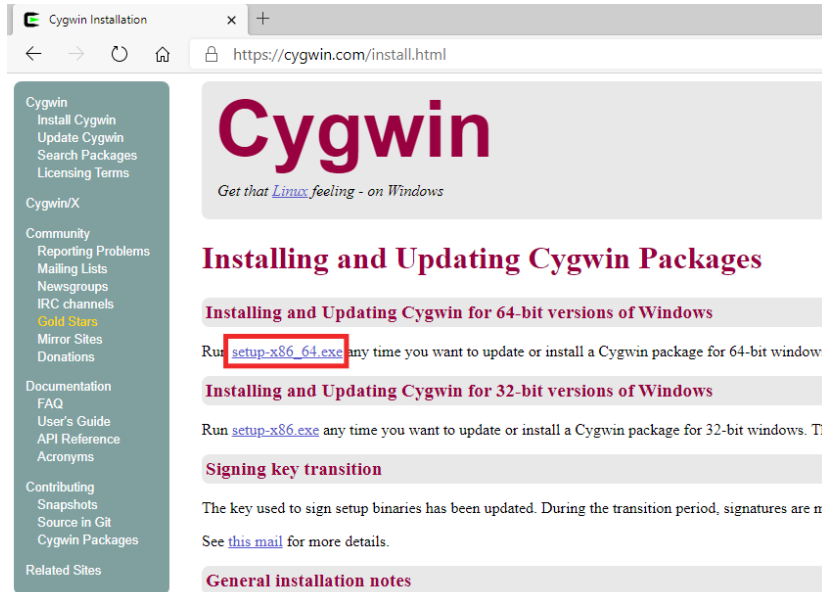


그림 8. Cygwin 설치 웹페이지

2. 컴퓨터에서 설치 파일을 두 번 클릭하여 실행합니다(그림 9). 기본 옵션을 유지하면서 **Next** 을 여러 번 클릭합니다. 설치 프로그램에서 **다운로드 사이트를 선택**하라는 메시지가 표시되며(그림 10) 둘 중 하나를 선택할 수 있습니다.

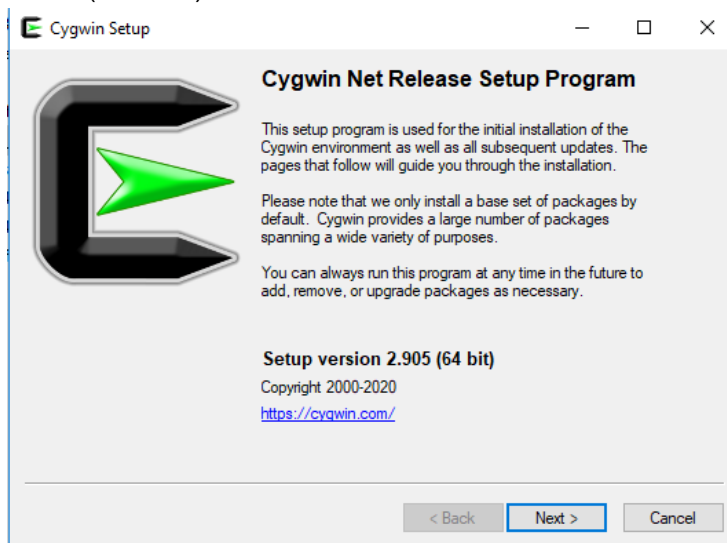


그림 9. Cygwin 설치 창

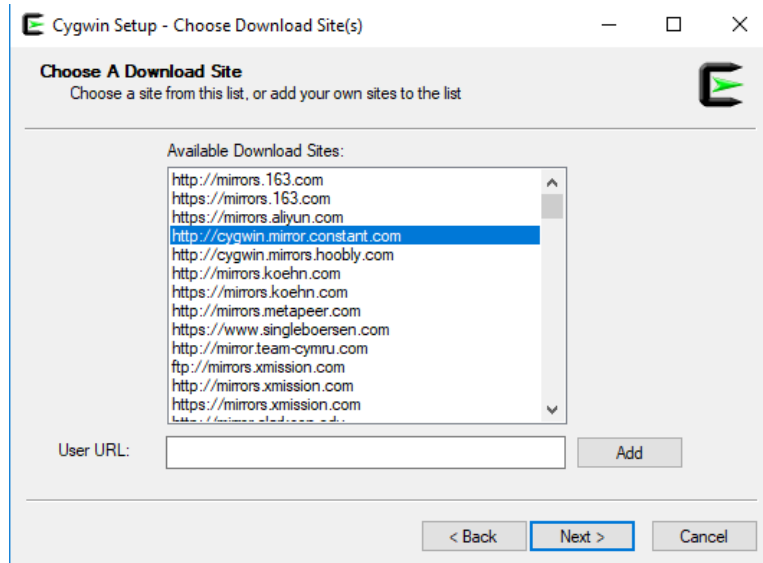


그림 10. 다운로드 사이트 선택

3. 여러 단계를 거치면 **Select Packages** 창이 나타납니다(그림 11). 그림 11 과 같이 **전체(Full)** 보기를 선택합니다.

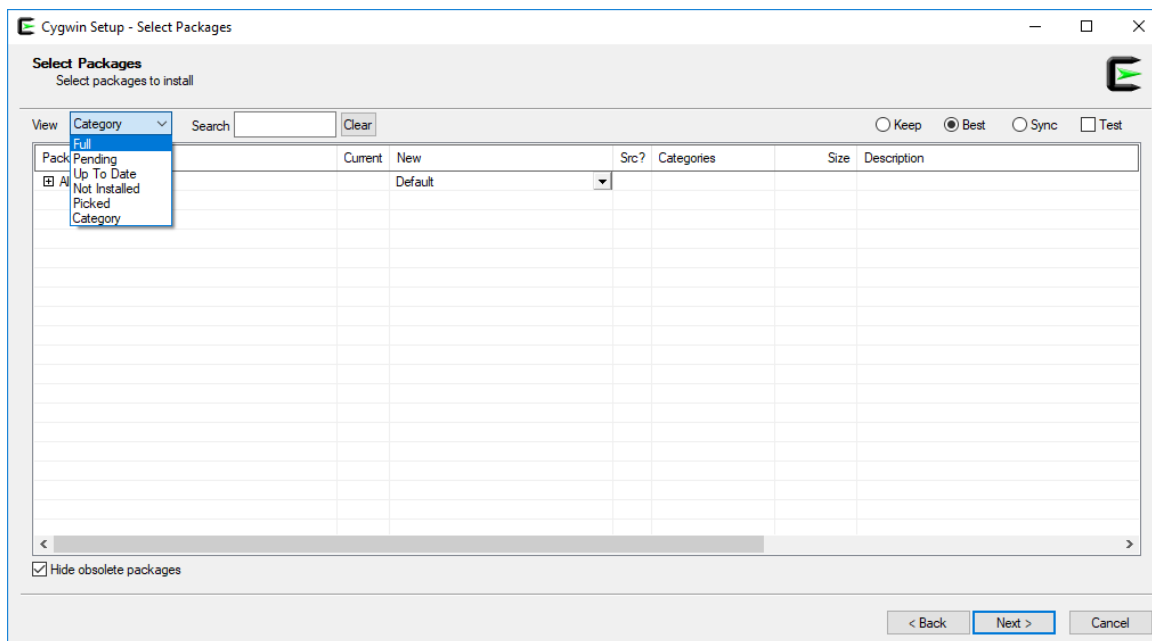


그림 11. 패키지 선택 창

4. 설치할 수 있는 패키지의 전체 목록이 나타납니다(그림 12). **검색** 상자에서 설치할 특정 패키지를 선택합니다.

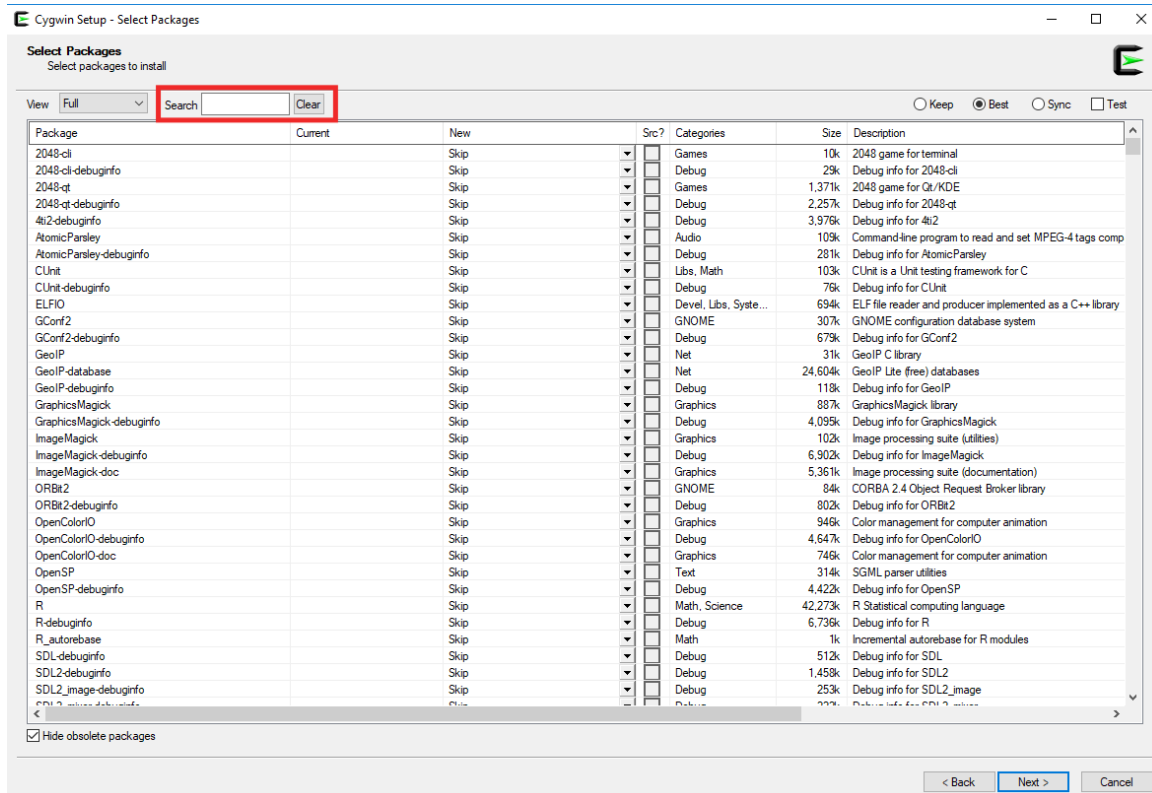


그림 12. 패키지 선택 창 – 전체 보기

Verilator 를 컴파일하고 새 시뮬레이터 바이너리를 생성하려면 다음 패키지를 설치해야 합니다.

- git
- make
- autoconf
- gcc-core
- gcc-g++
- flex
- bison
- perl
- libargp-devel

Cygwin 설치에 최소한 이러한 패키지를 포함하십시오. 아래 단계에 따라 하나씩 선택합니다(목록의 첫 번째 패키지인 `git`; 에 대한 자세한 단계만 표시합니다. 프로세스는 다른 패키지에 대해서도 동일합니다).

- 검색 상자에서 `git` 패키지를 찾습니다(그림 13).

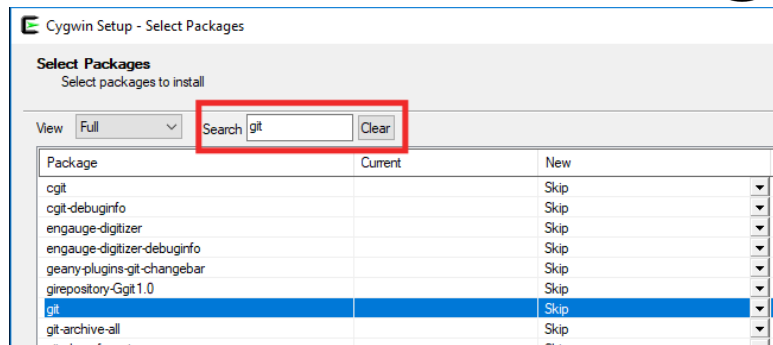


Figure 13. 그림 13. git 패키지 찾기

- 드롭-다운 메뉴에서 최신 버전을 선택하고 확인란을 선택합니다(그림 14).



그림 14. 가장 최신 버전을 선택하고 확인란을 선택합니다.

- 위 목록의 나머지 패키지에 대해서도 동일한 작업을 수행합니다.
5. 9 개의 패키지를 선택했으면 후속 창에서 **다음**을 클릭하여 Cygwin 설치에 이러한 패키지를 포함하고(설치 프로세스, 그림 15 참조, 몇 분 정도 걸릴 수 있음) **Finish** 를 클릭하여 설치를 완료합니다(그림 16).

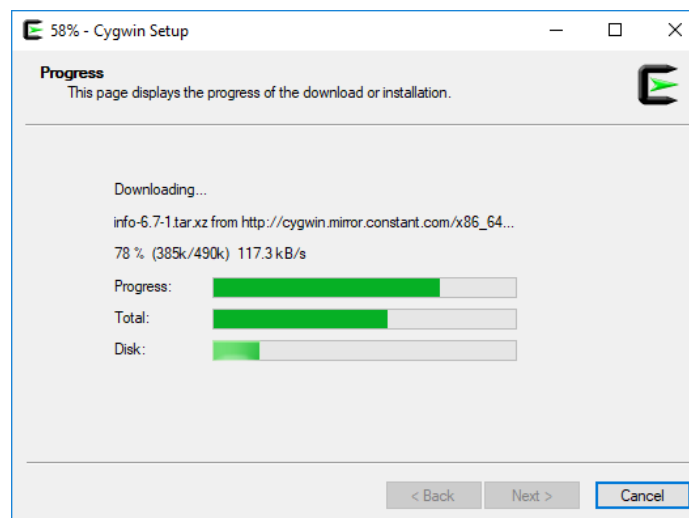


그림 15. Cygwin 설정

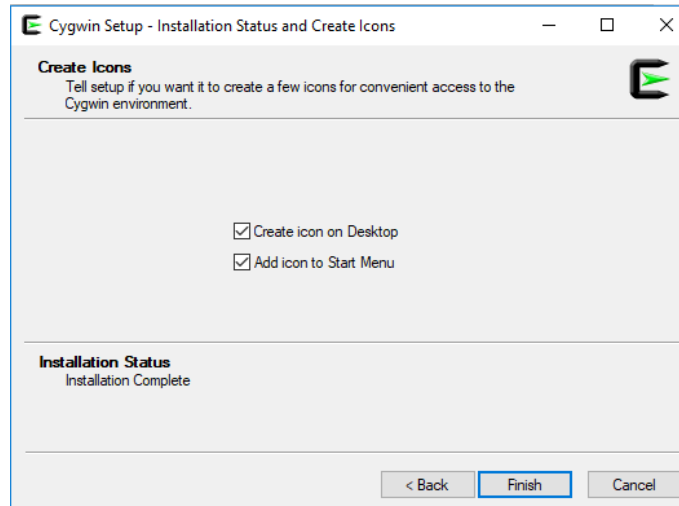


그림 16. 설치 완료

6. Cygwin 설치에 패키지를 추가해야 하는 경우 해당 패키지에 대해 2-5 단계를 반복합니다.

2. Windows 에서 Verilator 설치:

다음 단계에 따라 Windows 10 에 Verilator 를 설치하십시오.

1. Windows 바탕 화면이나 시작 메뉴에서 사용할 수 있는 Cygwin 터미널(그림 17)을 엽니다.



그림 17. Cygwin 터미널

2. 다음 단계에 따라 Verilator 를 빌드하고 설치합니다. 컴퓨터 속도에 따라 시간(몇 시간까지)이 걸릴 수 있습니다.

- `git clone https://git.veripool.org/git/verilator`
- `cd verilator`
- `git pull`
- `git checkout v4.020`

- `autoconf`
- `./configure`
- `make`
- `make install`

3. Windows 에서 GTKWave 설치:

GTKWave 는 <https://sourceforge.net/projects/gtkwave/files/>에서 미리 컴파일된 패키지로 다운로드할 수 있습니다. 가장 최근의 Windows 패키지(이 문서가 작성될 당시 **gtkwave-3.3.100-bin-win64** 라고 함)를 찾아 다운로드하고 압축을 풉니다(압축 해제).

bin 폴더에서 gtkwave 라는 실행 파일을 찾을 수 있으며 Windows 시스템에서 실행하고 사용할 수 있습니다.