

ChipWhisperer ver5.x.x 설치 매뉴얼

정보컴퓨터공학과 권혁동

Contents

설치 요약

설치 방법

심플 시리얼 빌드

이거 왜이럼



설치 요약

1. Python 3.x 설치
2. ChipWhisperer 설치 파일 다운 및 설치
3. ChipWhisperer 하드웨어 드라이버 설치
4. AVR GCC or ARM GCC 설치

설치 방법

- 설치하기에 앞서
- 본 설치 방법은 **Windows 64bit**의 설치 방법을 따름
- 사용하는 장비는 ChipWhisperer-Lite 1173 (CW1173)
- 다른 OS도 유사하지만 약간 차이가 존재하니 주의
- **ChipWhisperer 장비는 켜지 않은채로 시작**

설치 방법



- <https://www.python.org/downloads/>
- 파이썬 3.x.x 설치
- 20.03 기준 3.8.2 버전
- **환경변수 등록 필수**

설치 방법

- ChipWhisperer에서 사용하는 패키지 설치 필요
- 시작 → 명령 프롬프트 (또는 윈도우+R -> cmd)
- 다음 명령어들을 사용하여 패키지 설치
 - pip install pyqtgraph
 - pip install configobj
 - pip install pyusb (ChipWhisperer Capture Rev2 사용 시에만)
 - pip install umysql (MySQL 트레이스 형식 출력 필요할 때만)

설치 방법







5.1.3

 colinoflynn released this on 15 Nov 2019 · 47 commits to develop since this release

- Windows installer now available (requires Linux for Windows subsystem + installing make + compilers in Linux on Windows)
- Jupyter notebooks:
 - Use real-time plotting during capture (like old scope view)
 - New LPC1114 Jupyter tutorial based on existing wiki page
 - Improvement to SPA password bypass example shows starting with unknown password
 - Improve default plotting for static plots (matplotlib when small plots instead of bokeh)
- Add PSOC62 HAL, fix NRF52840 HAL and SAM4L HAL
- Add CW-Nano firmware source + schematics

NOTE: The source releases do NOT include the jupyter submodule.

Assets 6

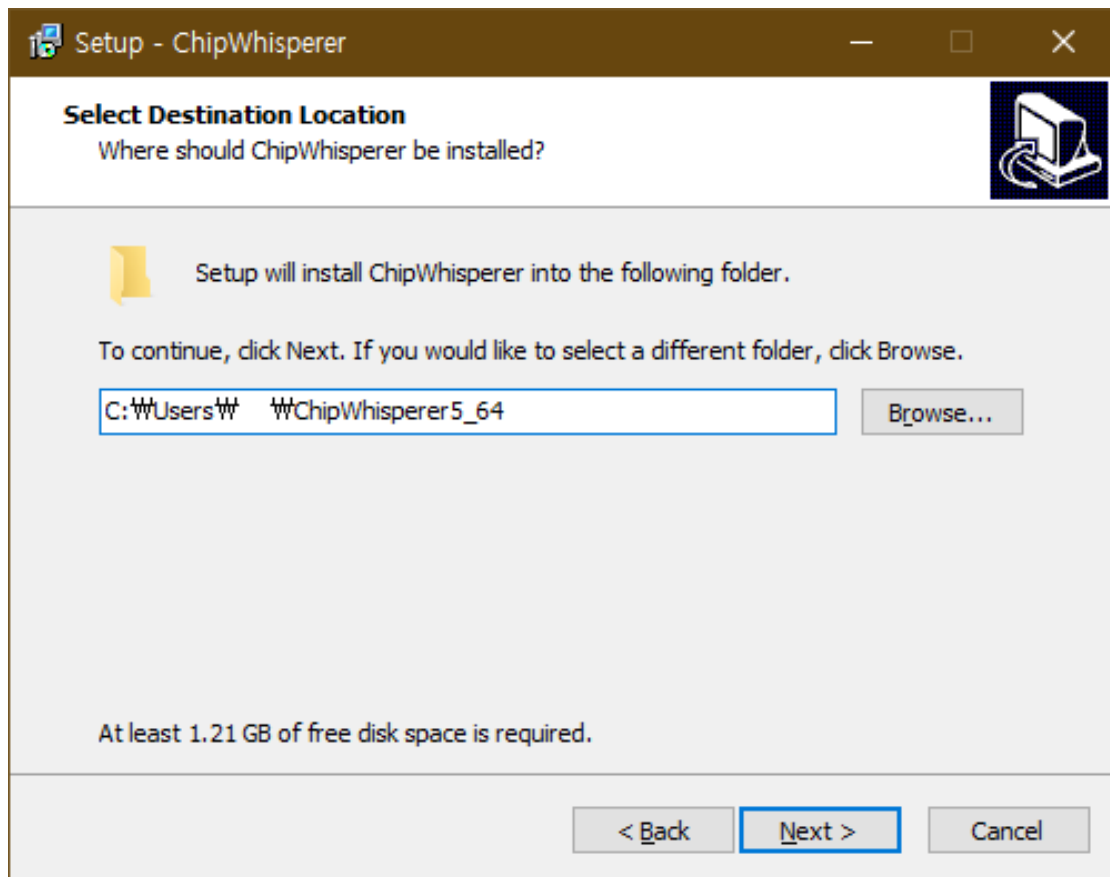
 Chipwhisperer.v5.1.3.Setup.32-bit.exe	428 MB
 Chipwhisperer.v5.1.3.Setup.64-bit.exe	434 MB
 newae_windowsusb_drivers.zip	1.14 MB
 Virtual.Machine.ChipWhisperer.Jupyter-5-1-3.7z	1.62 GB
 Source code (zip)	
 Source code (tar.gz)	

- <https://github.com/newaetech/chipwhisperer/releases/>

- ChipWhisperer 깃허브 릴리즈 페이지에서 설치파일 다운

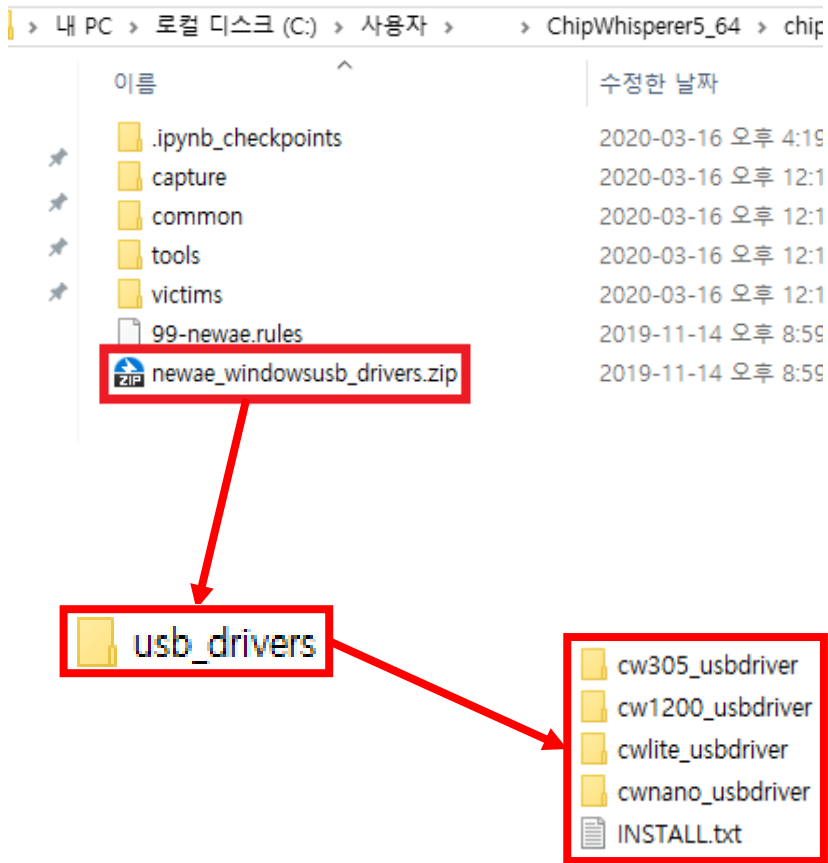
- 20.03 기준 5.1.3 버전

설치 방법



- 설치 파일을 실행하여 설치
- 기본 설치 경로
 - C:\Users\<이름>\ChipWhisperer5_64
- 경로는 바뀌어도 무방
 - 하지만 기본 경로를 권장

설치 방법



- chipwhisperer\hardware 경로

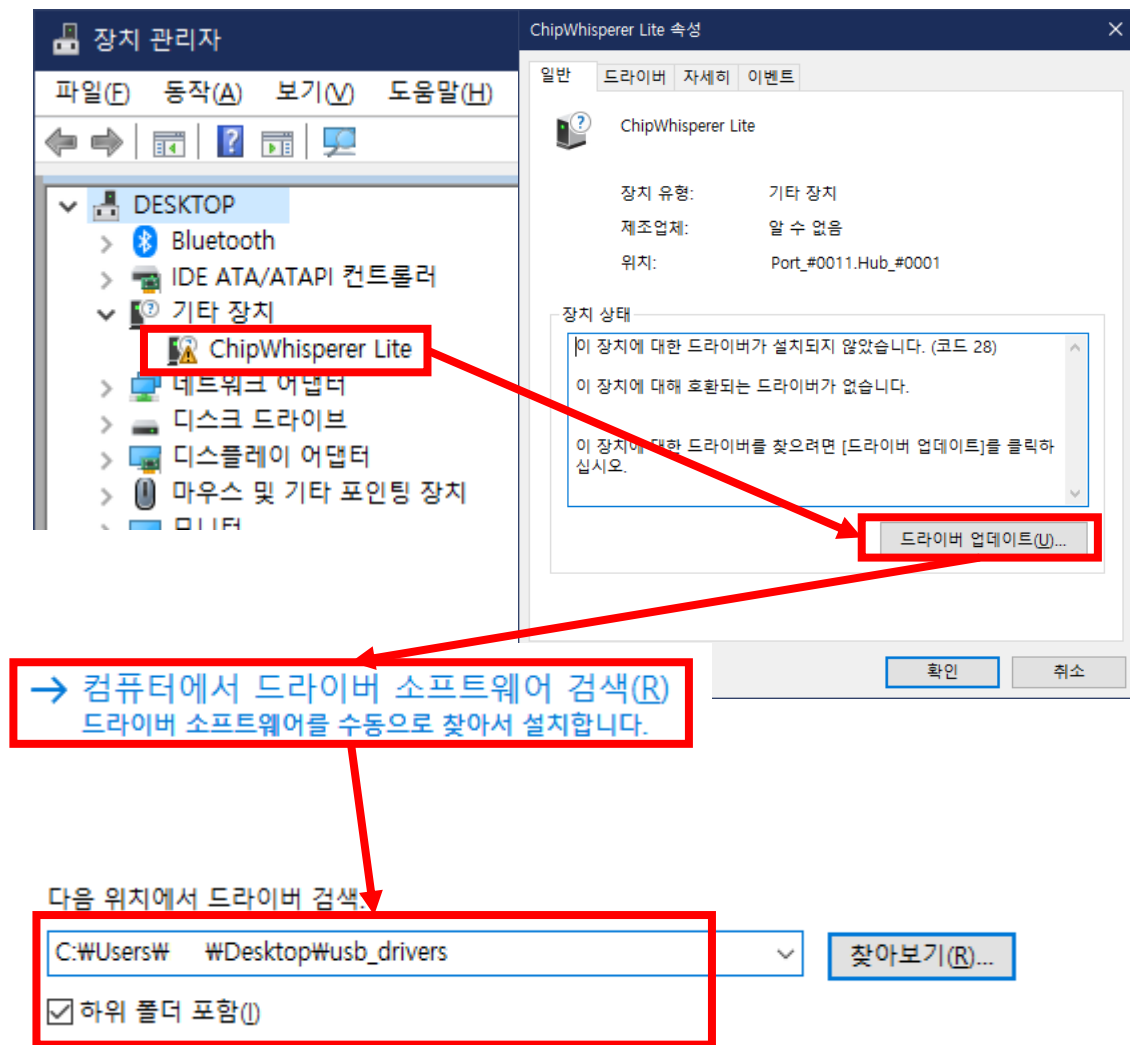
- 드라이버 압축파일 압축 해제

- usb_drivers 폴더 생성 확인

- 편의를 위해 바탕화면으로 이동

* 내용물은 그냥 확인만 하고 usb_drivers 폴더 내에 그대로 둔다

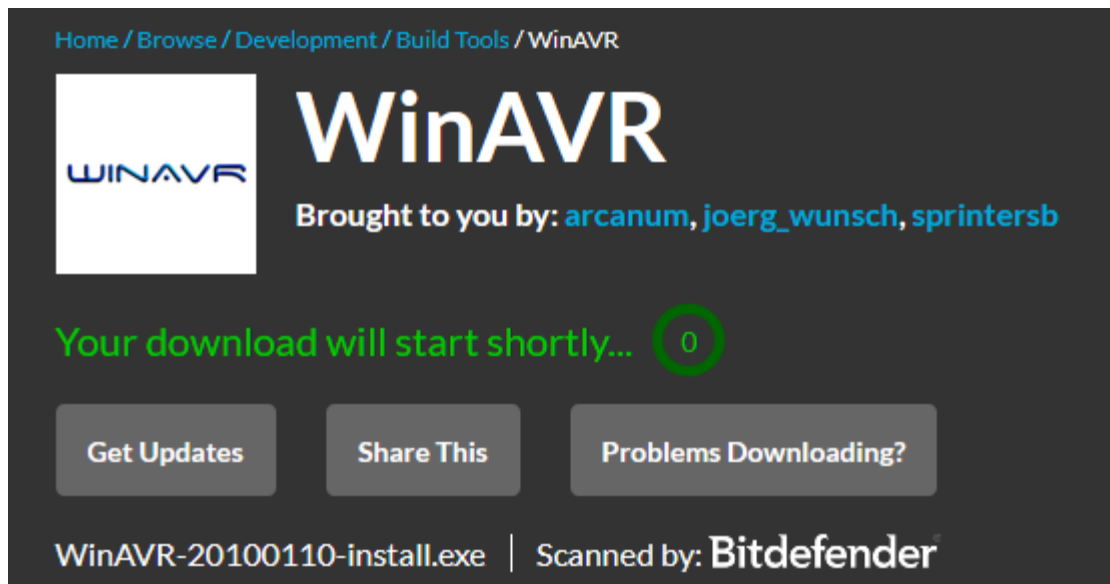
설치 방법



- ChipWhisperer 하드웨어 연결
 - USB 포트 사용
- 장치 관리자에서 장치 확인
- 우클릭 -> 속성 ->
드라이버 업데이트 -> 파일 선택

설치 방법

* 가진 장비의 타겟이 XMEGA 보드일 때 설치



- https://sourceforge.net/projects/winavr/files/latest/download?source=typ_redirect
- WinAVR 설치
 - 20.03기준 버전 20100110
- 설치 시 환경 변수 등록 진행
- 설치 완료 후 재부팅

설치 방법

* 가진 장비의 타킷이 ARM(STM) 일때 설치

In this release

1

[gcc-arm-none-eabi-9-2019-q4-major-win32.exe](#)

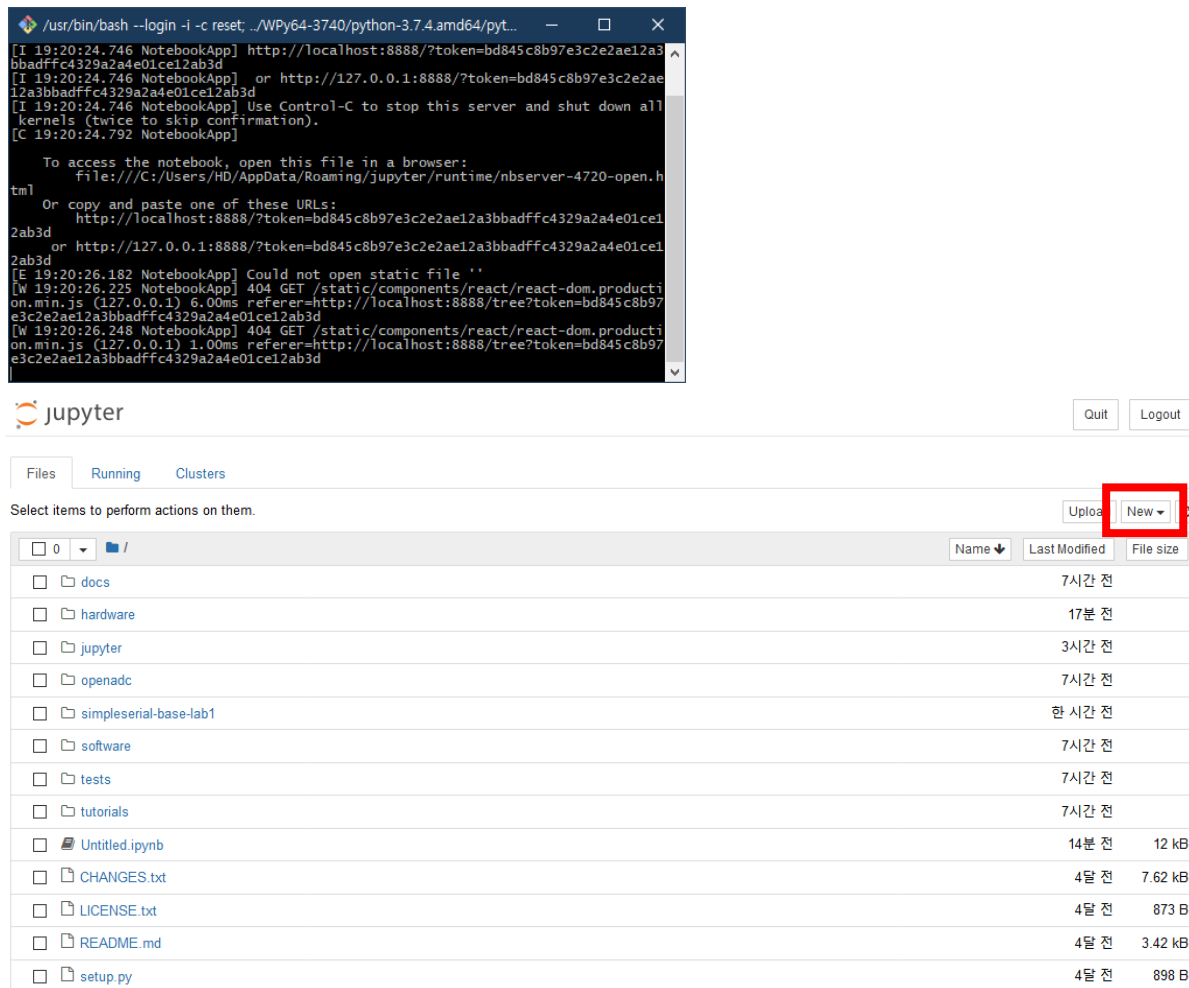
Windows 32-bit Installer (Signed for Windows 10 and later)

(Formerly SHA2 signed binary)

MD5: 033151c92a5cd986e4cbea058f93d91b

- <https://developer.arm.com/tools-and-software/open-source-software/developer-tools/gnu-toolchain/gnu-rm/downloads>
- ARM gcc 설치
 - 20.03기준 버전 9-2019-q4-major
- 설치 시 환경 변수 등록 진행
- 설치 완료 후 재부팅

심플 시리얼 프로젝트 빌드

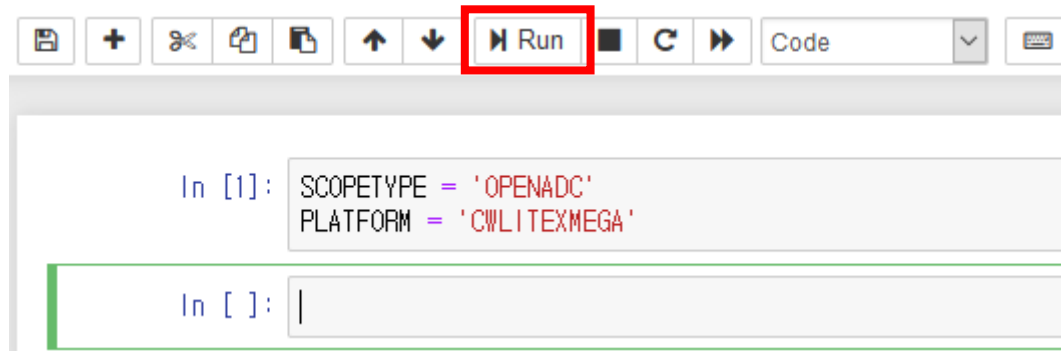


The screenshot displays a Jupyter Notebook interface. The top terminal window shows the execution of a Python script to start a web server. The output indicates that the server is running on port 8888. Below the terminal, the Jupyter file browser is visible, showing a list of files and folders. The 'New' button is highlighted with a red box.

Name	Last Modified	File size
docs	7시간 전	
hardware	17분 전	
jupyter	3시간 전	
openadc	7시간 전	
simpleserial-base-lab1	한 시간 전	
software	7시간 전	
tests	7시간 전	
tutorials	7시간 전	
Untitled.ipynb	14분 전	12 kB
CHANGES.txt	4달 전	7.62 kB
LICENSE.txt	4달 전	873 B
README.md	4달 전	3.42 kB
setup.py	4달 전	898 B

- 본 내용은 <https://chipwhisperer.readthedocs.io/en/latest/tutorials.html>의 Firmware Build Setup에서 발췌
- 코드는 복사하는 것이 편하므로 코드 복사시에는 링크의 텍스트를 참고
- ChipWhisperer 실행 시 Jupyter Notebook이 자동으로 실행
 - 실수로 Notebook을 꺼버렸다면 bash가 실행된 상태에서 localhost:8888 접속
- 우측의 New → Python3를 선택하여 신규 파일 생성

심플 시리얼 프로젝트 빌드



- Notebook은 한 블록에 코드를 입력하고 상단의 Run을 눌러 실행
- 현재 선택한 블록만 부분적으로 실행
- 튜토리얼의 코드를 부분적으로 입력 및 실행하면서 진행
- 화면에 결과를 띄우는 코드가 있다면 그 결과를 화면에 띄워서 보여줌

심플 시리얼 프로젝트 빌드

```
SCOPETYPE = 'OPENADC'  
PLATFORM = 'CWLITEXMEGA'
```

- 1단계: SCOPETYPE과 PLATFORM 정의
- Scope: OPENADC, CWNANO 중에서 선택
- Platform: CWLITEARM, CWLITEXMEGA, CWNANO 중에서 선택
- 현재 사용하는 장비의 내용을 설정

심플 시리얼 프로젝트 빌드

```
%%bash
#check for avr-gcc
avr-gcc --version

#check for ARM gcc
arm-none-eabi-gcc --version
```

실행 결과

```
avr-gcc.exe (WinAVR 20100110) 4.3.3
Copyright (C) 2008 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

arm-none-eabi-gcc.exe (GNU Tools for Arm Embedded Processors 7-2018-q2-update
Copyright (C) 2017 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

- 2단계: 컴파일러 확인
- 컴파일러가 제대로 설치 되었는지 확인
- 제대로 설치 되었다면 우측과 같은 실행 결과 확인 가능
- 컴파일러가 잘 설치 되었다면 본 단계는 생략 가능

심플 시리얼 프로젝트 빌드

```
%%bash
cd ../hardware/victims/firmware/
mkdir -p simpleserial-base-lab1 && cp -r simpleserial-base/* $_
cd simpleserial-base-lab1
```

- 3단계: 프로젝트 파일 복사
- firmware 폴더의 simpleserial-base 폴더 내의 모든 파일을 복사
- 프로젝트를 파일 탐색기를 통해 직접 복사했다면 이 과정은 생략 가능

심플 시리얼 프로젝트 빌드

```
CRYPTO_TARGET = "NONE"
```

- 4단계: TARGET 설정
- 본 프로젝트는 target이 없으므로 NONE으로 설정

심플 시리얼 프로젝트 빌드

```
%%bash -s "$PLATFORM" "$CRYPTO_TARGET"  
cd ../hardware/victims/firmware/simpleserial-base-lab1  
make PLATFORM=$1 CRYPTO_TARGET=$2
```

실행 결과

```
Creating Symbol Table: simpleserial-base-CWLITEXMEGA.sym  
avr-nm -n simpleserial-base-CWLITEXMEGA.elf > simpleserial-base-CWLITEXMEGA.s  
Size after:  


| text | data | bss | dec  | hex | filename                          |
|------|------|-----|------|-----|-----------------------------------|
| 1798 | 16   | 52  | 1866 | 74a | simpleserial-base-CWLITEXMEGA.elf |

  
+-----+  
+ Built for platform CW-Lite XMEGA  
+-----+
```

- 5단계: 프로젝트 빌드
- 방금 복사한 프로젝트를 빌드
- 빌드에 성공하면 우측과 같은 결과를 확인 가능
 - 실제 내용은 매우 긴 내용이며, 슬라이드의 캡처는 하단의 내용 일부만 캡처

심플 시리얼 프로젝트 빌드

```
import chipwhisperer as cw
scope = cw.scope()
target = cw.target(scope, cw.targets.SimpleSerial)

# setup scope parameters
if SCOPETYPE == "OPENADC":
    scope.gain.db = 45
    scope.adc.samples = 3000
    scope.adc.offset = 1250
    scope.adc.basic_mode = "rising_edge"
    scope.clock.clkgen_freq = 7370000
    scope.clock.adc_src = "clkgen_x4"
    scope.trigger.triggers = "tio4"
    scope.io.tio1 = "serial_rx"
    scope.io.tio2 = "serial_tx"
    scope.io.hs2 = "clkgen"
elif SCOPETYPE == "CWNANO":
    scope.io.clkout = 7370000
    scope.adc.clk_freq = 7370000
    scope.io.tio1 = "serial_rx"
    scope.io.tio2 = "serial_tx"

if "STM" in PLATFORM or PLATFORM == "CWLITEARM" or PLATFORM == "CWNANO":
    prog = cw.programmers.STM32FProgrammer
elif PLATFORM == "CW303" or PLATFORM == "CWLITEXMEGA":
    prog = cw.programmers.XMEGAProgrammer
else:
    prog = None

fw_path = '../hardware/victims/firmware/simpleserial-base-lab1/simpleserial-t
cw.program_target(scope, prog, fw_path)
```

실행 결과

XMEGA Programming flash...
XMEGA Reading flash...
Verified flash OK, 1813 bytes

- 6단계: 스크립트 작성
- ChipWhisperer 장비와 통신이 가능하도록 함
- 실행 결과는 현재 가진 장비에 따라 다르게 출력될 수 있음

심플 시리얼 프로젝트 빌드

```
ktp = cw.ktp.Basic() # object to generate fixed/random key and text (default
key, text = ktp.next() # get our fixed key and random text

target.simpleserial_write('k', key)
target.simpleserial_wait_ack()
scope.arm()

target.simpleserial_write('p', text)

ret = scope.capture()
trace = scope.get_last_trace()
output = target.simpleserial_read('r', 16)

from binascii import hexlify
print(hexlify(output))
print(hexlify(text))
```

실행 결과

b'df27bb88a9d3873efbb88cfc688aefa7'
b'df27bb88a9d3873efbb88cfc688aefa7'

- 7단계: 입출력 확인
- ChipWhisperer 장비에 값을 보내고 출력하는 코드

심플 시리얼 프로젝트 빌드

```
scope.dis()  
target.dis()
```

- 8단계: 연결 해제
- ChipWhisperer 장비와 연결 상태를 해제
- 정상적으로 진행이 되었다면 ChipWhisperer 설정에 성공한 것

Troubleshooting

- 2단계에서 컴파일러 확인이 안됨
- 또는, 5단계에서 make 명령어가 동작하지 않음
- import 이후로 동작을 안함

- **해결법**

- 컴파일러 환경변수 설정이 되었는지 확인
- 컴파일러 설치 후 재부팅 했는지 확인
- PLATFORM 변수의 이름에 오타가 있는지 확인
- import 이후로 동작 안하는 건 파이썬 문제, 파이썬 버전이 3.x인지 확인 후 환경변수 설정 확인

Troubleshooting

```
%%bash -s "$PLATFORM" "$CRYPTO_TARGET"  
cd ../hardware/victims/firmware/simpleserial-base-lab1  
make PLATFORM=$1 CRYPTO_TARGET=$2  
  
bash: line 1: cd: ../hardware/victims/firmware/simpleserial-base-lab1: No such file or directory  
make: *** No targets specified and no makefile found. Stop.
```

- 복사 했는데 파일이 없다고 나오는 경우
- **해결법**
- 상대 경로이므로 현재 프로젝트 파일이 위치한 곳을 확인
- 현재 위치한 경로에서 적절히 찾아갈 수 있도록 경로명 수정
 - 만약 프로젝트 최상위 디렉토리라면 앞의 '../'를 지운다
- 또는, 절대경로 사용
 - 절대경로 입력 시 파이썬 문법상의 이유로 '\'는 '\\'로 입력해야 함
 - 절대경로는 ChipWhisperer가 설치된 경로를 찾아가면 획득 가능

Troubleshooting

```
In [19]: cw.program_target(scope, prog, fw_path)

c:\users\chipwhisperer5_64\chipwhisperer\software\chipwhisperer\hardware\naeusb\programmer_xmega.py in enablePDI(self, status)
    307         if status:
    308             # self._xmegaDoWrite(self.XPROG_CMD_LEAVE_PROGMODE)
--> 309             self._xmegaDoWrite(self.XPROG_CMD_ENTER_PROGMODE)
    310             self._pdienabled = True
    311         else:

c:\users\chipwhisperer5_64\chipwhisperer\software\chipwhisperer\hardware\naeusb\programmer_xmega.py in _xmegaDoWrite(self, cmd, data, checkStatus)
    502         status = self._xmegaDoRead(cmd=0x0020, dlen=3)
    503         if status[1] != 0x00:
--> 504             raise IOError("XMEGA Command %x failed: err=%x, timeout=%d" % (status[0], status[1], status[2]))
    505
    506     def _xmegaDoRead(self, cmd, dlen=1):

OSError: XMEGA Command 20 failed: err=1, timeout=1
```

- 뭔가 긴 내용의 오류에 XMEGA Command 20 failed이라고 나오는 경우

- **해결법**

- 가진 장비가 XMEGA 보드가 아니므로 PLATFORM 변수를 수정
 - STM32F303이면 'CWLITEARM' 또는 'CW308_STM32F3'으로 수정
 - CWLITEARM과 CW308_STM32F3은 내부에서 동일한 값으로 취급함

Troubleshooting

```
In [30]: cw.program_target(scope, prog, fw_path)
```

```
Serial baud rate = 115200  
Detected known STM32: STM32F302xB(C)/303xB(C)  
Extended erase (0x44), this can take ten seconds or more  
Attempting to program 4695 bytes at 0x8000000  
STM32F Programming flash...  
STM32F Reading flash...  
Serial baud rate = 38400
```

```
-----  
OSError                                Traceback (most recent call last)  
<ipython-input-30-b876eece5c0> in <module>  
----> 1 cw.program_target(scope, prog, fw_path)
```

```
c:\Users\chipwhisperer5_64\chipwhisperer\sc  
ry(self, addr, fdata, smallblocks)  
553         logging.info("Verify fa  
554         if fails > 3:  
--> 555             raise IOError(""  
556         else:  
557             #Redo this block
```

```
OSError: Verify failed at 0x007f, 0 != b1
```

- 되는 척 하더니 Verify failed라고 뜨는 경우
- **해결법**
- baudrate 문제
- USB 3.0 포트에 USB 3.0 케이블을 사용해서 연결
- 또는, `cw.program_target(scope, prog, fw_path, baud=38400)`으로 baudrate를 명시