



M-FPD3-QUAD-CAM-UVC 사용설명서 KOR

REVISION 1.2.0

MGSG CO.,LTD

Revision history

Revision	Date	Description	Update by
1.2.0	2019.06.01	Initial creation	jhyoo
	2019.06.13	Update	jhyoo
	2019.10.03	Update	jhyoo
	2019.10.11	Update	jhyoo
	2019.10.25	Darknet yolo v3 example, image stitching(panorama image generate) example	Jhyoo

Table of Contents

- 1. Package**
- 2. Overview**
- 3. Examples**
- 4. Support**

1. Package

1.1 내용물



- M-FPD3-QUAD-CAM-UVC 보드 1EA
- USB3.0 케이블 AM-BM 1.2meter 1EA
- 전원 어댑터 DC 12V, 1A이상

Figure 1.1.1 M-FPD3-QUAD-CAM-UVC 내용물

2. Overview

M-FPD3-QUAD-CAM-UVC는 카메라 보드 4개를 연결할 수 있는 USB 3.0 출력의 개발보드입니다.

RJ45커넥터를 사용하는 FPD LINK 3 인터페이스 카메라 보드와 연결하여 사용 할 수 있습니다.

4채널 카메라를 연결하여 UVC로 출력하는, 다운로드 가능한 binary 파일과

회로도 및 USB3.0 영상 전송 예제(FPGA generated pattern image) 소스코드가 제공됩니다.

2.1 보드 구성

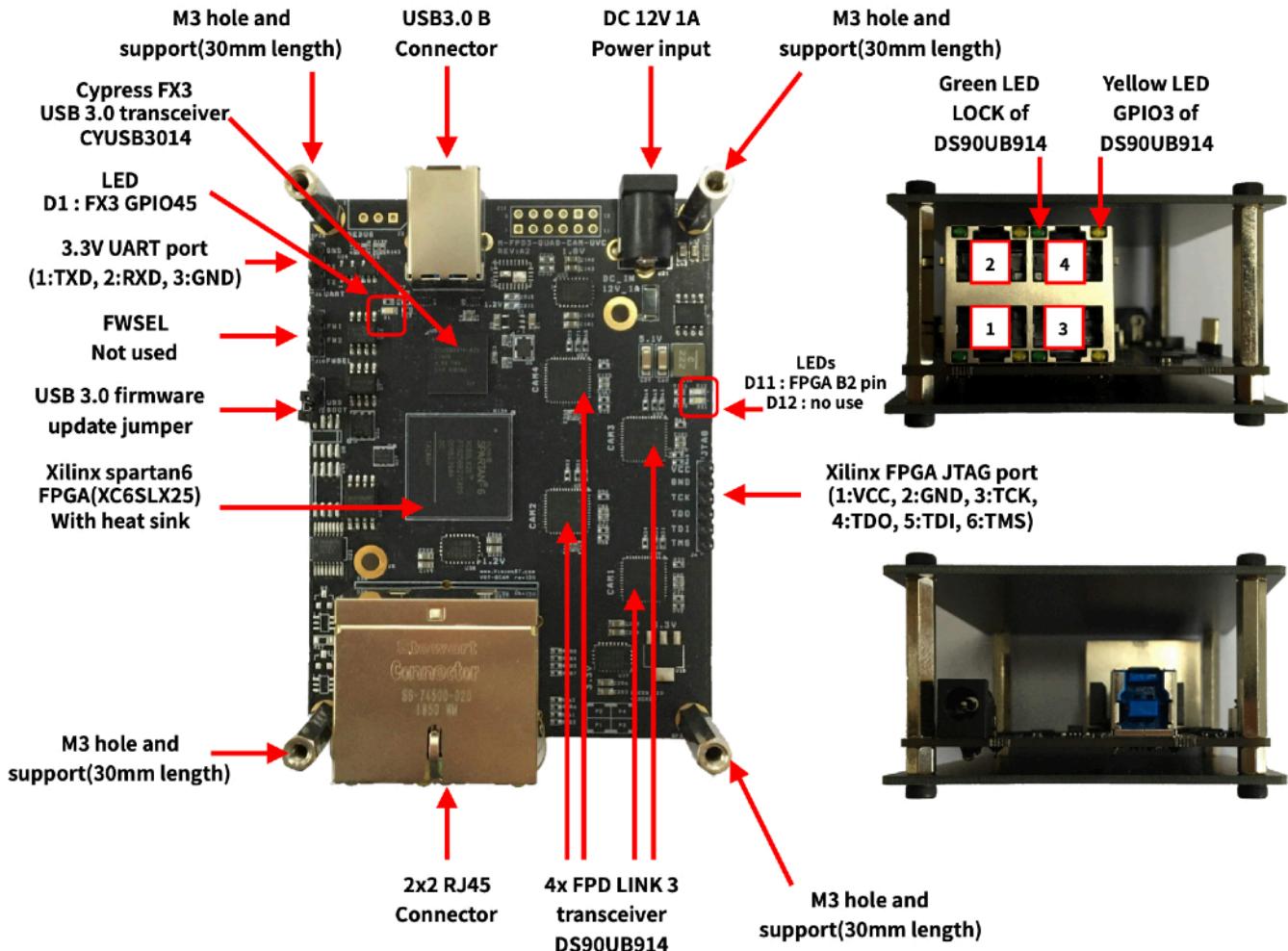


Figure 2.1.1 M-FPD3-QUAD-CAM-UVC 보드 구성

2.2 보드 크기

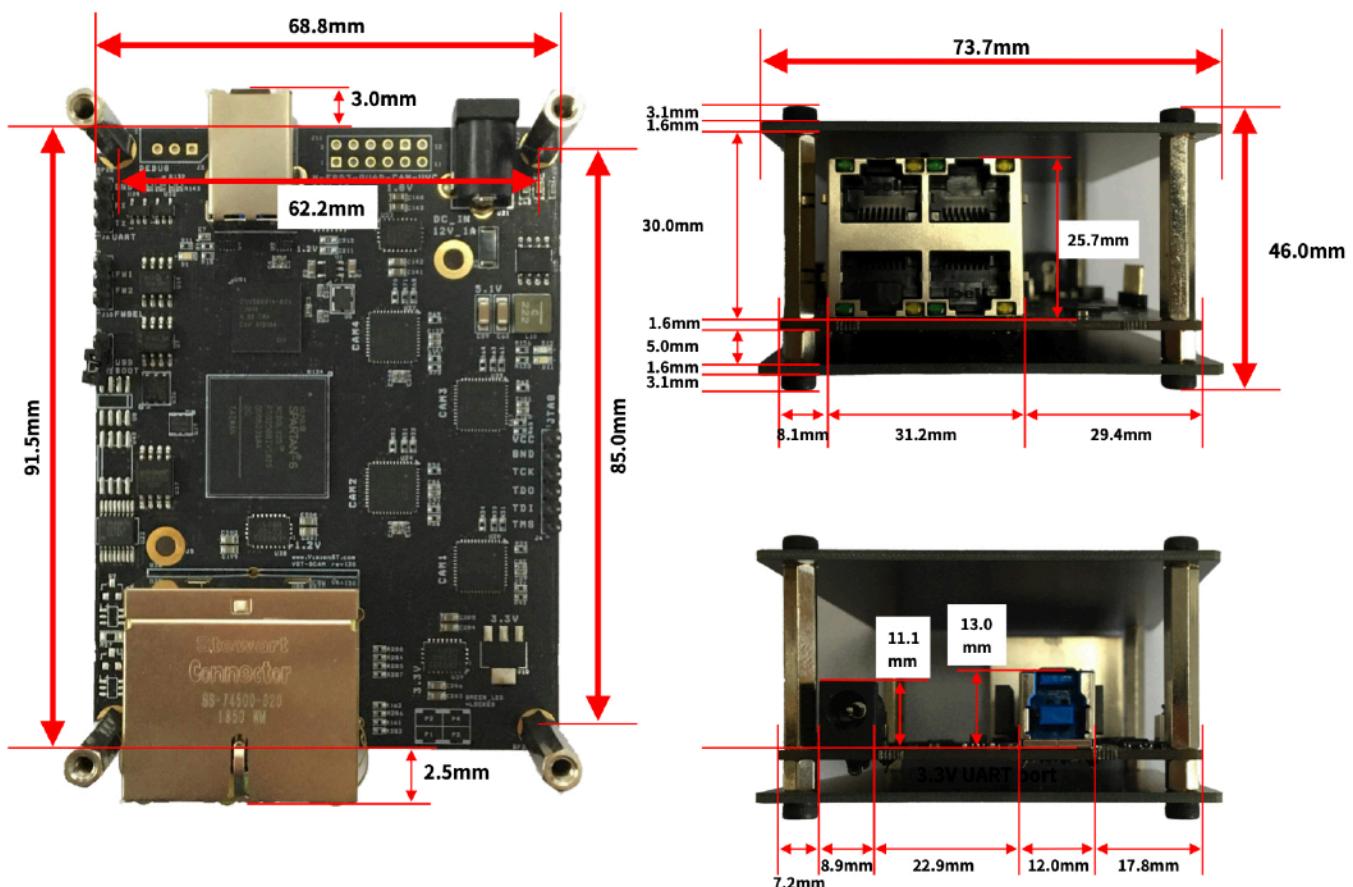


Figure 2.2.1 M-FPD3-QUAD-CAM-UVC 보드 차수

2.3 사양

- 기본 동작(구매 시 다운로드 되어있는 펌웨어)
 - UVC camera, 7680x1080@15fps, uncompressed color YUY2 format(binary 파일 제공)
 - 지원 가능 해상도 펌웨어, uncompressed color YUY2 format(binary 파일 제공)
 - UVC camera, 7680x1080@15fps(4 camera 1920x1080 해상도)
 - AR0230 sensor size : 5.76x3.24mm(sensor pixel size 3um)
 - UVC camera, 3840x540@30fps(4 camera 960x540 해상도)
 - AR0230 sensor size : 5.76x3.24mm(sensor pixel size 3um)
 - UVC camera, 5120x720@30fps(4 camera 1280x720 해상도)
 - AR0230 sensor size : 3.84x2.16mm(sensor pixel size 3um)
 - 기타 해상도 : 가로방향 1920 이하, 세로방향 1080 이하
- USB 3.0 super speed
 - USB 3.0 B connector
- 전원 입력
 - DC 12V 1A max
- FPGA
 - XC6SLX25-2FTG256C , Xilinx Spartan6
- 네개의 FPD LINK 3 입력 인터페이스
 - Quad RJ45 connector
 - DS90UB914QSQ
- Xilinx JTAG 연결 단자
 - 2.54mm 간격 6핀 male 커넥터(1:VCC, 2:GND, 3:TCK, 4:TDO, 5:TDI, 6:TMS)
- UART 단자
 - 2.54mm 간격 3핀 male 커넥터
 - 1 : TXD, 2: RXD, 3: GND
- LEDs
 - D1 : Blue LED, FX3 GPIO 45에 연결됨
 - D11 : Blue LED, FPGA B2 pin에 연결됨
 - D12 : no use

3. Examples

3.1 시스템 요구 및 기본사항

- Github 자료 링크 : <https://github.com/mgsgo/M-FPD3-QUAD-CAM-UVC>
 - 예제, 회로도 등
- Xilinx Spartan6 개발툴
 - Xilinx ISE 14.7
 - https://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/design-tools/v2012_4---14_7.html
 - https://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/design-tools/14_7-windows.html
- Cypress FX3 SDK 1.3.4 개발툴
 - <https://www.cypress.com/documentation/software-and-drivers/ez-usb-fx3-software-development-kit>

3.2 영상 전송 기본 펌웨어 7680x1080@15fps 4camera 1920x1080 동작

- 기본 다운로드되어있는 7680x1080 해상도 카메라 영상 전송 예제(binary 파일 제공)
 - 4개의 카메라가 모두 연결되어있어야 동작함.
 - 4개의 카메라 초기화 후, video 신호 확인이 완료되면 LED D11을 켠다. 그리고 USB 3.0 통신을 시작한다.
 - 필요한 프로그램 : UVC 지원 프로그램
 - Python capture code 예제 Github 링크 : https://github.com/mgsgo/M-FPD3-QUAD-CAM-UVC/tree/master/rev110/python_capture_code
 - Python/opencv 설치 동영상 링크 : <https://youtu.be/4O2ecZpBHJI>
 - Windows10 예제 실행 동영상 링크 : <https://www.youtube.com/watch?v=vIV-92WtKc4>
 - Jetson nano 예제 실행 동영상 링크 : https://www.youtube.com/watch?v=MRlrfN_J2s

- 기본 동작 케이블 연결
 - USB 3.0 펌웨어 점퍼 OFF
 - UAB 3.0 AM BM cable 연결
 - 4x CAT7 SSTP 케이블 연결
 - DC 12V 1A 전원 아답터 연결

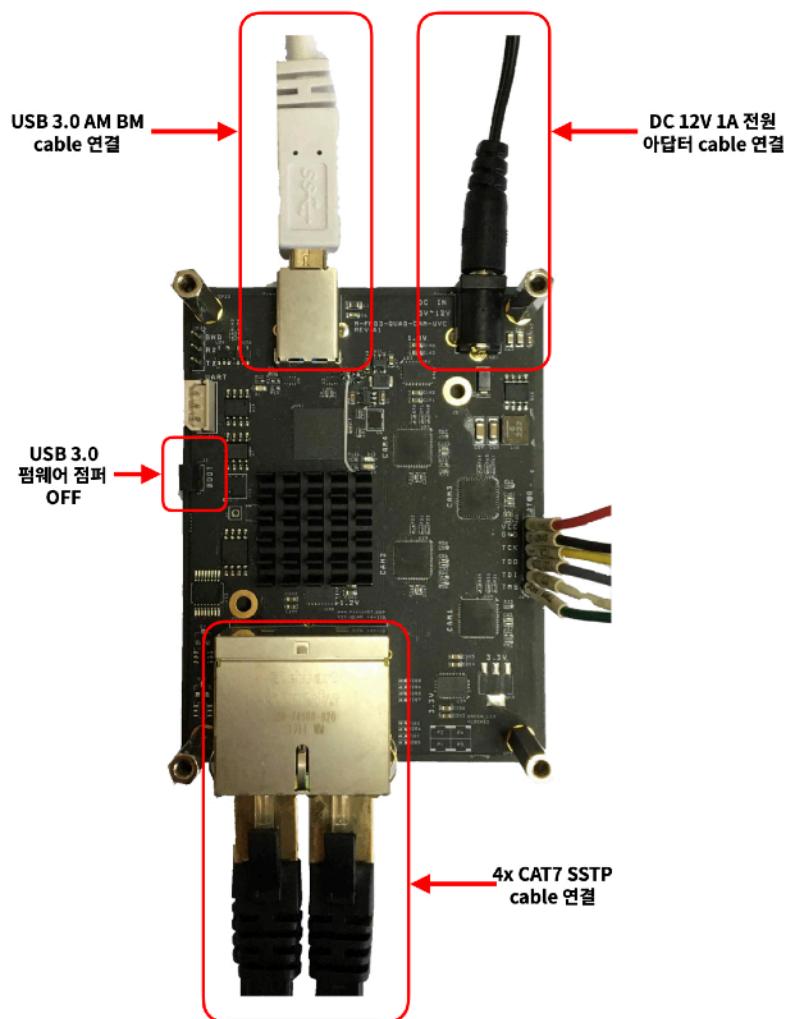


Figure 3.2.1 M-FPD3-QUAD-CAM-UVC 보드 케이블 연결

- cap7680x1080_display1920x1080x4.py

- 7680x1080 해상도 영상 capture/1920x1080 해상도 영상 display x4 python code

```
import cv2
capture = cv2.VideoCapture(0)
capture.set(cv2.CAP_PROP_FRAME_WIDTH, 7680)
capture.set(cv2.CAP_PROP_FRAME_HEIGHT, 1080)
while True:
    ret, frame = capture.read()
    image = frame.copy()

    image_1 = image[0:1080, 0:1920]
    image_2 = image[0:1080, 1920:3840]
    image_3 = image[0:1080, 3840:5760]
    image_4 = image[0:1080, 5760:7680]

    cv2.imshow("video1", image_1)
    cv2.imshow("video2", image_2)
    cv2.imshow("video3", image_3)
    cv2.imshow("video4", image_4)

    if cv2.waitKey(1) > 0: break
capture.release()
cv2.destroyAllWindows()
```

- cap7680x1080_display960x540x4.py

- 7680x1080 해상도 영상 capture/960x540 해상도 영상 display x4 python code

```
import cv2
capture = cv2.VideoCapture(0)
capture.set(cv2.CAP_PROP_FRAME_WIDTH, 7680)
capture.set(cv2.CAP_PROP_FRAME_HEIGHT, 1080)
while True:
    ret, frame = capture.read()
    image = frame.copy()

    image_1 = image[0:1080, 0:1920]
    image_2 = image[0:1080, 1920:3840]
    image_3 = image[0:1080, 3840:5760]
    image_4 = image[0:1080, 5760:7680]

    image_1 = cv2.resize(image_1, (960, 540))
    image_2 = cv2.resize(image_2, (960, 540))
    image_3 = cv2.resize(image_3, (960, 540))
    image_4 = cv2.resize(image_4, (960, 540))

    cv2.imshow("video1", image_1)
    cv2.imshow("video2", image_2)
    cv2.imshow("video3", image_3)
    cv2.imshow("video4", image_4)

    if cv2.waitKey(1) > 0: break
capture.release()
cv2.destroyAllWindows()
```

- cap7680x1080_display960x540x4_save7680x1080x1x15fps.py
 - 7680x1080 해상도 영상 capture/960x540 해상도 영상 display/7680x1080 해상도 영상x1 저장

python code

```
import cv2
capture = cv2.VideoCapture(0)
fourcc = cv2.VideoWriter_fourcc(*'mp4v')
capture.set(cv2.CAP_PROP_FRAME_WIDTH, 7680)
capture.set(cv2.CAP_PROP_FRAME_HEIGHT, 1080)

writer_1 = cv2.VideoWriter('7680x1080_cam1234.mp4', fourcc, 15.0, (7680,1080))
while True:
    ret, frame = capture.read()
    image = frame.copy()

    image_1 = image[0:1080,  0:1920]
    image_2 = image[0:1080, 1920:3840]
    image_3 = image[0:1080, 3840:5760]
    image_4 = image[0:1080, 5760:7680]

    image_1 = cv2.resize(image_1, (960, 540))
    image_2 = cv2.resize(image_2, (960, 540))
    image_3 = cv2.resize(image_3, (960, 540))
    image_4 = cv2.resize(image_4, (960, 540))

    cv2.imshow("video1", image_1)
    cv2.imshow("video2", image_2)
    cv2.imshow("video3", image_3)
    cv2.imshow("video4", image_4)
    writer_1.write(image)
    if cv2.waitKey(1) > 0: break

capture.release()
cv2.destroyAllWindows()
```

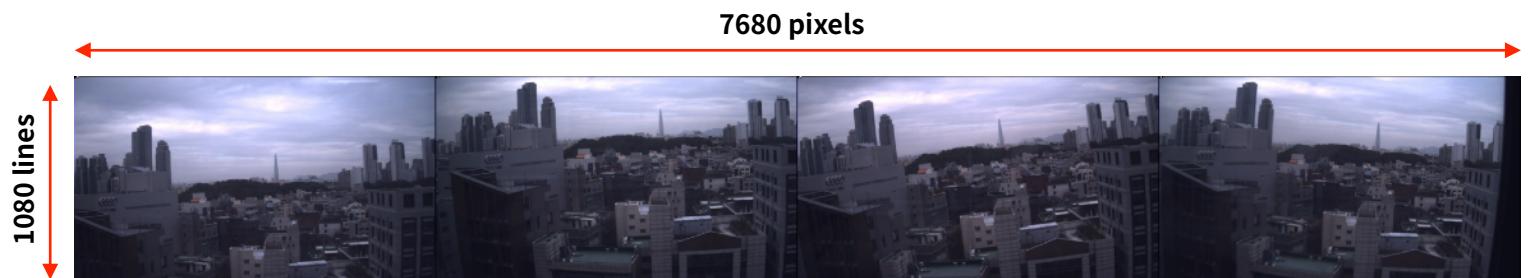


Figure 3.2.2 M-FPD3-QUAD-CAM-UVC 보드 영상 전송 기본 펌웨어(7680x1080) capture 결과

- cap7680x1080_display960x540x4_save1920x1080x4x15fps.py
 - 7680x1080 해상도 영상 capture/960x540 해상도 영상 display/1920x1080 해상도 영상x4 저장

python code

```
import cv2
capture = cv2.VideoCapture(0)
fourcc = cv2.VideoWriter_fourcc(*'mp4v')
capture.set(cv2.CAP_PROP_FRAME_WIDTH, 7680)
capture.set(cv2.CAP_PROP_FRAME_HEIGHT, 1080)
writer_1 = cv2.VideoWriter('1920x1080_cam1.mp4', fourcc, 15.0, (1920,1080))
writer_2 = cv2.VideoWriter('1920x1080_cam2.mp4', fourcc, 15.0, (1920,1080))
writer_3 = cv2.VideoWriter('1920x1080_cam3.mp4', fourcc, 15.0, (1920,1080))
writer_4 = cv2.VideoWriter('1920x1080_cam4.mp4', fourcc, 15.0, (1920,1080))
while True:
    ret, frame = capture.read()
    image = frame.copy()
    image_1 = image[0:1080,  0:1920]
    image_2 = image[0:1080, 1920:3840]
    image_3 = image[0:1080, 3840:5760]
    image_4 = image[0:1080, 5760:7680]
    writer_1.write(image_1)
    writer_2.write(image_2)
    writer_3.write(image_3)
    writer_4.write(image_4)
    image_1 = cv2.resize(image_1, (960, 540))
    image_2 = cv2.resize(image_2, (960, 540))
    image_3 = cv2.resize(image_3, (960, 540))
    image_4 = cv2.resize(image_4, (960, 540))
    cv2.imshow("video1", image_1)
    cv2.imshow("video2", image_2)
    cv2.imshow("video3", image_3)
    cv2.imshow("video4", image_4)
    if cv2.waitKey(1) > 0: break
capture.release()
cv2.destroyAllWindows()
```

- 윈도우 프로그램에서의 UVC control
 - **선명도(auto exposure enable) 0일 때**
 - 밝기 : 이미지센서 exposure
 - 개인 : 이미지센서 개인

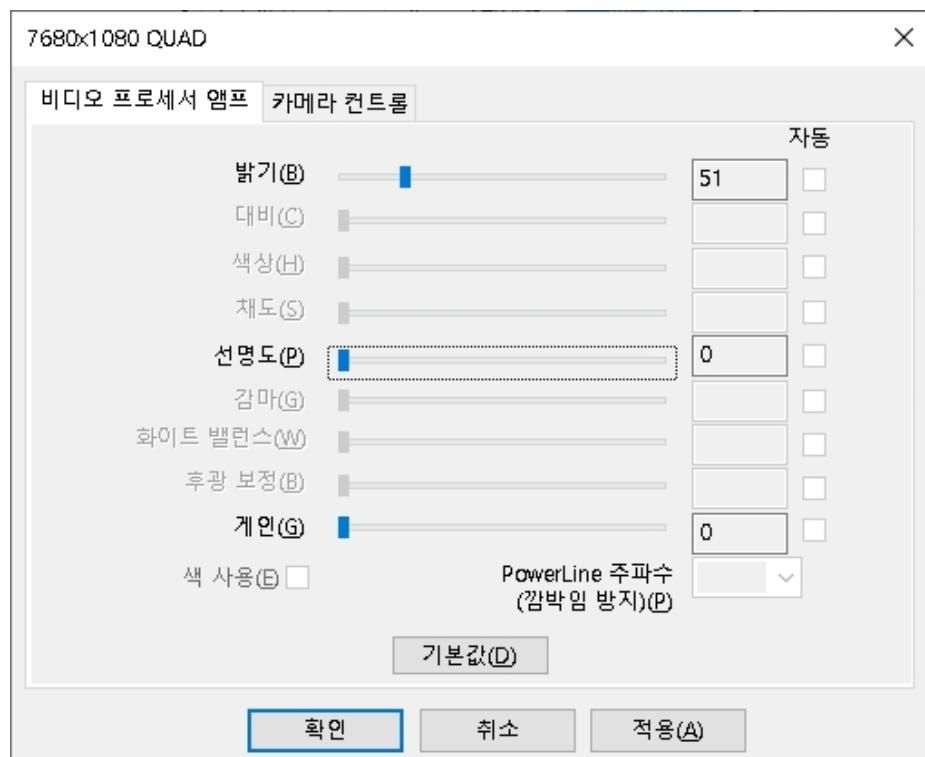


Figure 3.2.3 M-FPD3-QUAD-CAM-UVC 밝기/개인/선명도(auto exposure) 제어

- **선명도(auto exposure enable) 1일 때**

- 밝기 : 원하는 평균 밝기값
- 개인 : 상관없음

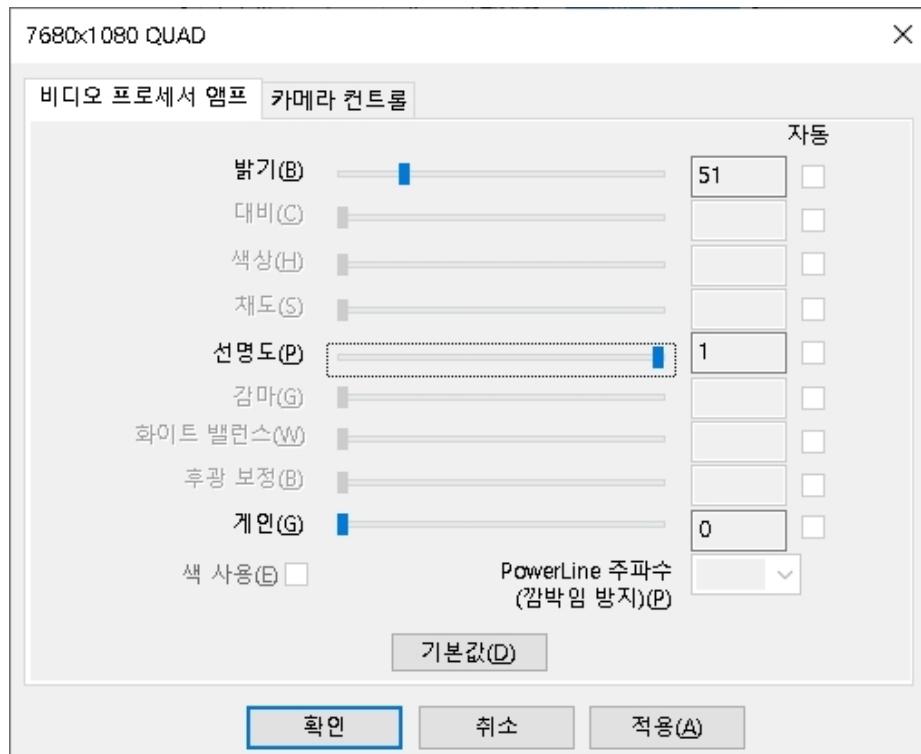


Figure 3.2.4 M-FPD3-QUAD-CAM-UVC 밝기/개인/선명도(auto exposure) 제어

3.3 펌웨어 업데이트

- USB 3.0 펌웨어 업데이트
 - 동영상 링크 : not yet prepared
- USB 3.0 펌웨어 업데이트시의 케이블 연결
 - USB 3.0 펌웨어 점퍼 ON
 - UAB 3.0 AM BM cable 연결
 - DC 12V 1A 전원 아답터 연결

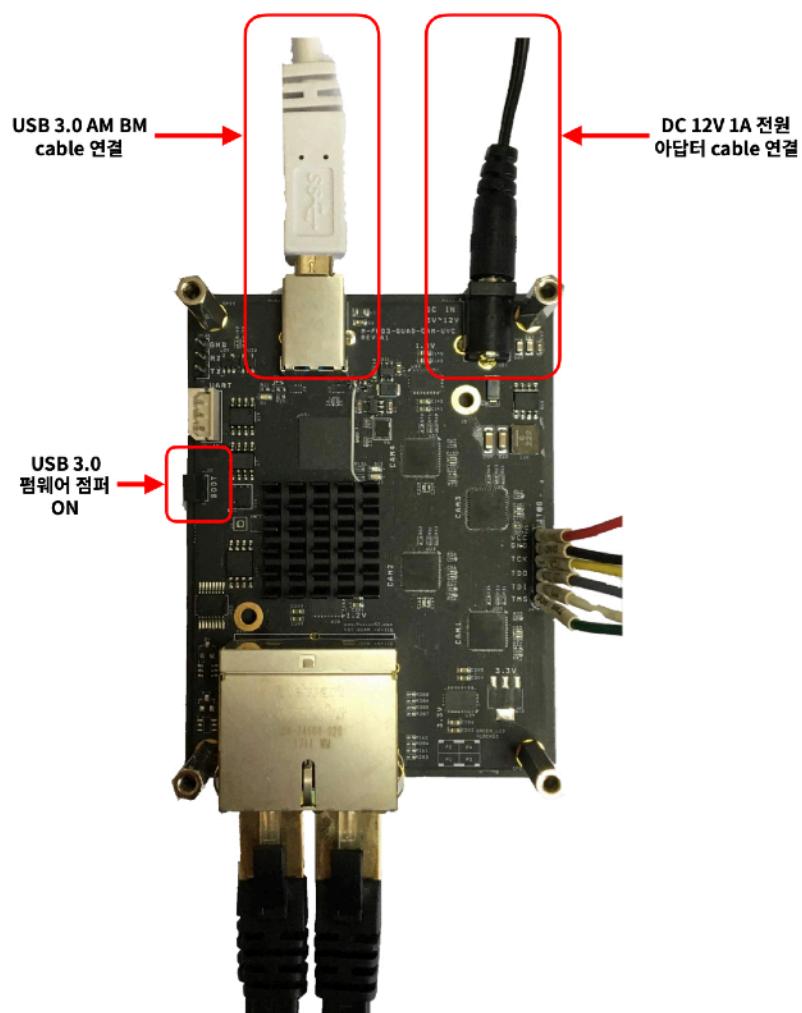


Figure 3.3.1 M-FPD3-QUAD-CAM-UVC 보드 USB 3.0 케이블 연결, USB 3.0 펌웨어 업데이트 점퍼 설정

- USB 3.0 펌웨어 업데이트 프로그램(USB control center)
 - USB control center 실행, device 선택
 - Program/FX3/I2C EEPROM 선택
 - USB control center 실행, device 선택, file 선택
 - Programming I2C EEPROM Succeeded

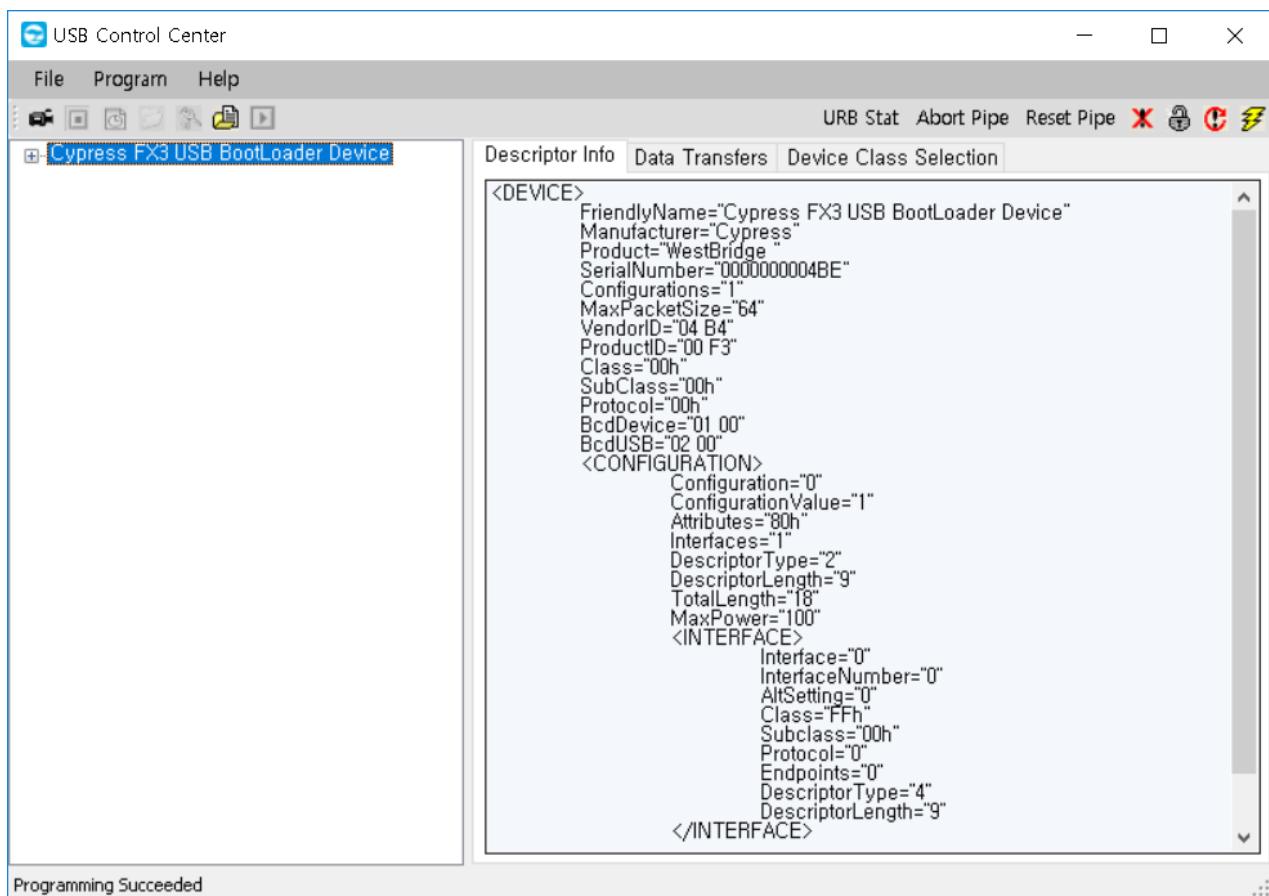


Figure 3.3.2 M-FPD3-QUAD-CAM-UVC 보드 USB 3.0 펌웨어 업데이트 device 선택

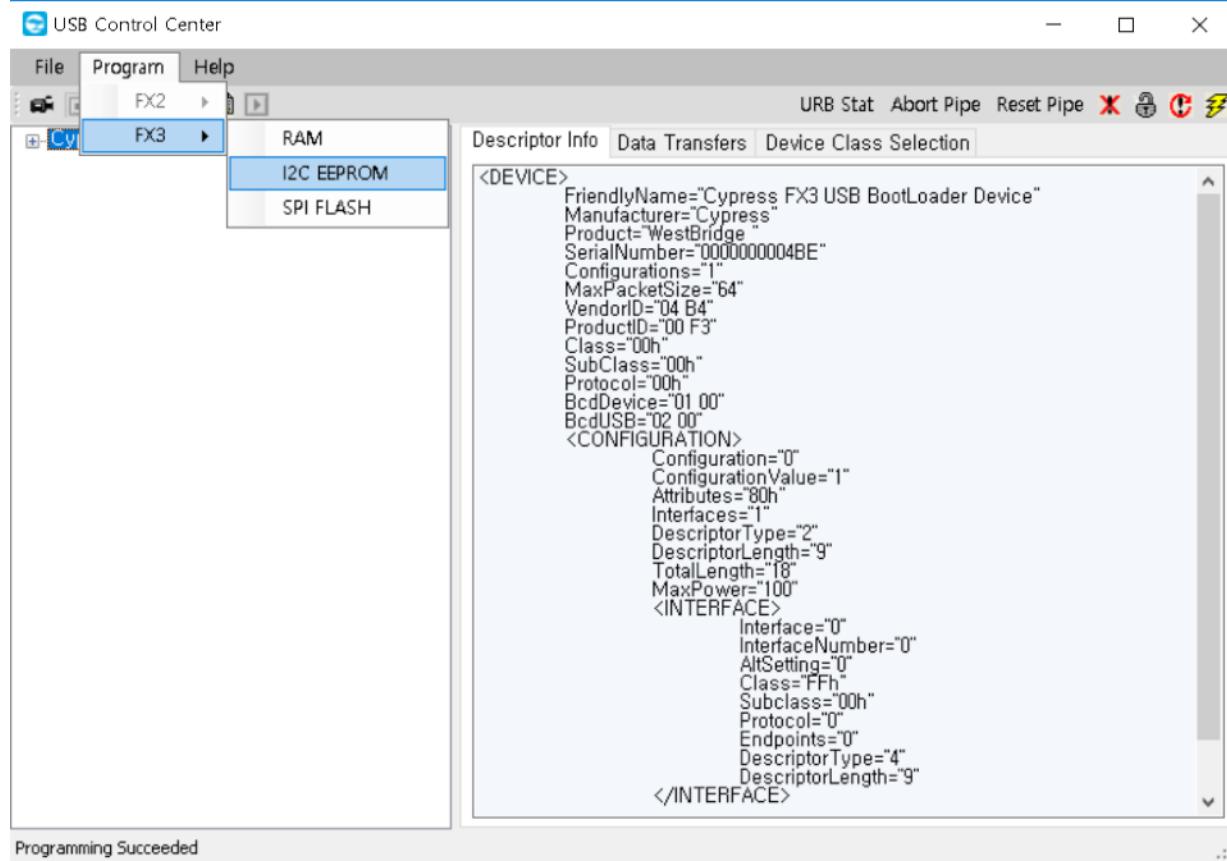


Figure 3.3.3 M-FPD3-QUAD-CAM-UVC USB 3.0 펌웨어 업데이트 EEPROM 선택

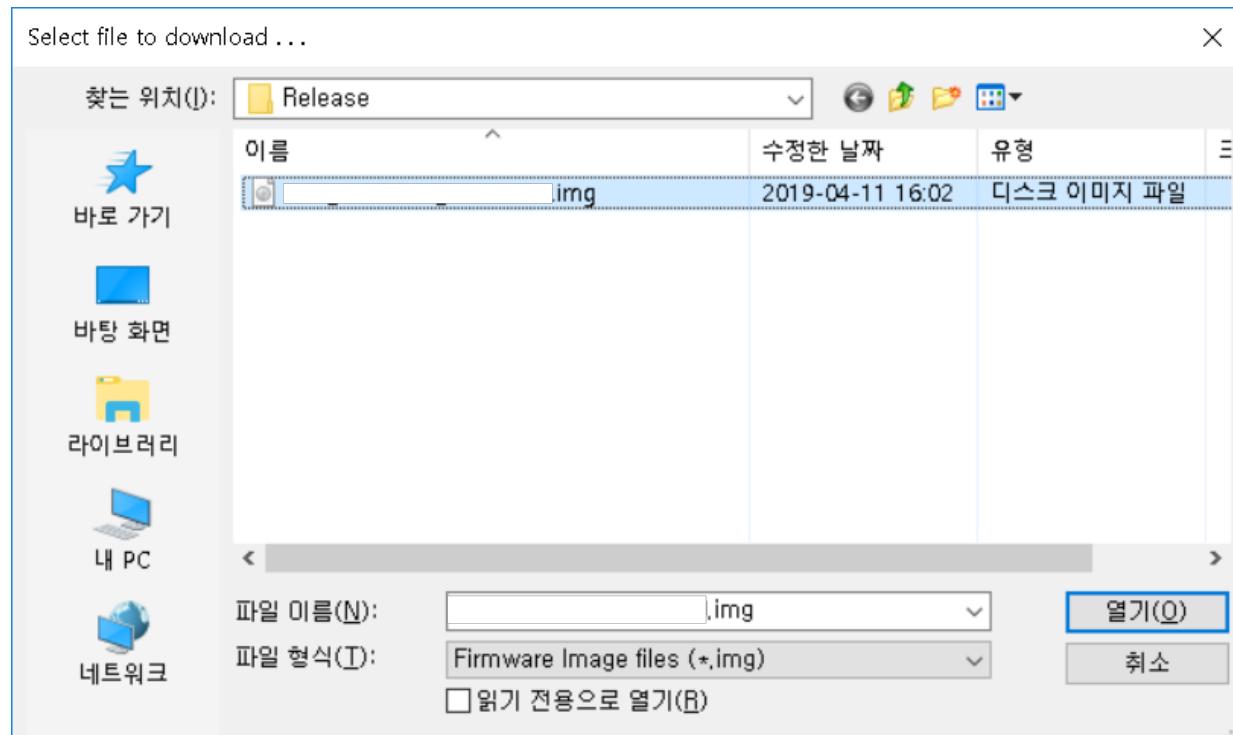


Figure 3.3.4 M-FPD3-QUAD-CAM-UVC 보드 USB 3.0 펌웨어 업데이트 device 선택, file 선택

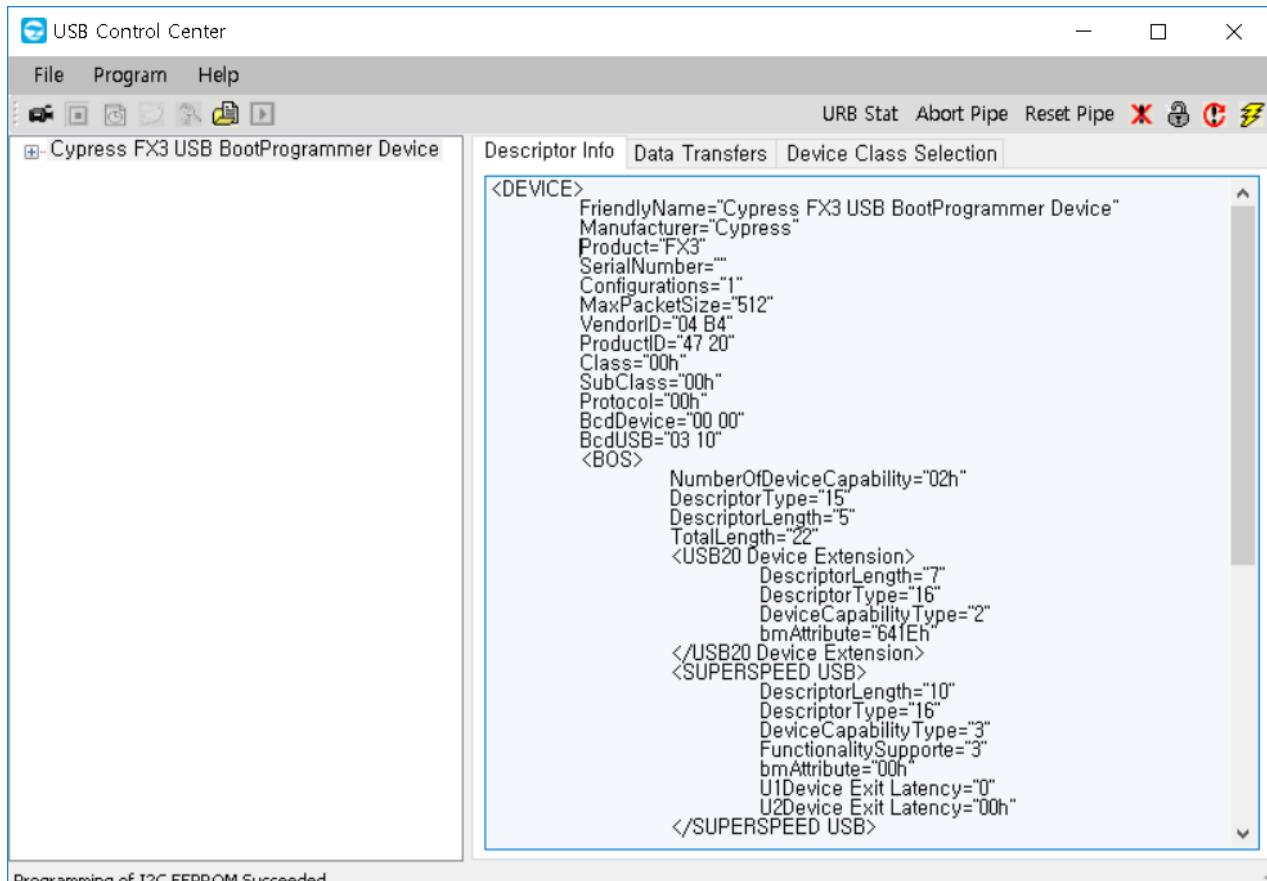


Figure 3.3.5 M-FPD3-QUAD-CAM-UVC 보드 USB 3.0 펌웨어 업데이트 Programming Succeeded

- FPGA 펌웨어 업데이트
 - 동영상 링크 : not yet prepared
- FPGA 펌웨어 업데이트시의 케이블 연결
 - Xilinx JTAG cable 연결(1:VCC, 2:GND, 3:TCK, 4:TDO, 5:TDI, 6:TMS)
 - DC 12V 1A 전원 아답터 연결

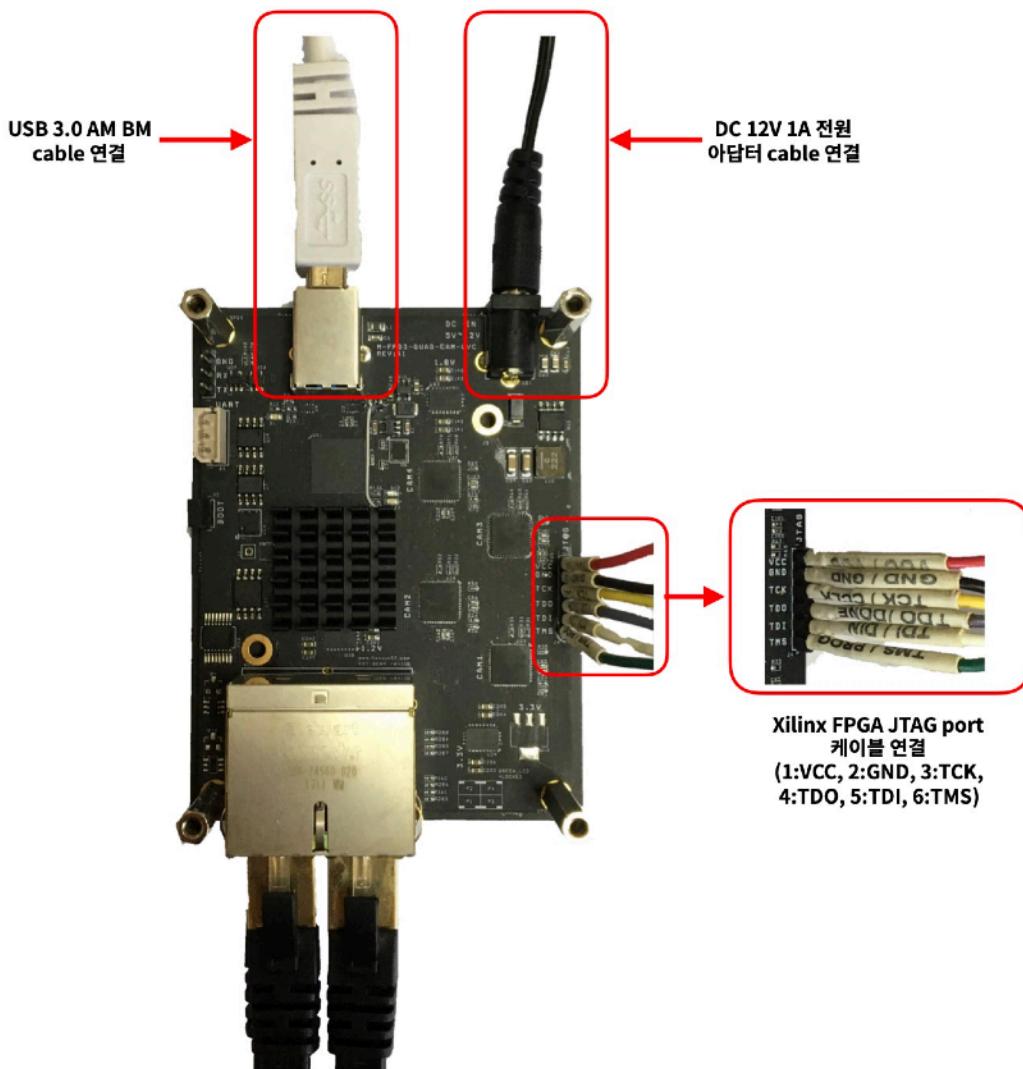


Figure 3.3.6 M-FPD3-QUAD-CAM-UVC 보드 FPGA 펌웨어 업데이트 케이블 연결

- FPGA 펌웨어 업데이트 프로그램 실행 (Xilinx ISE iMPACT)
 - Xilinx iMPACT 실행, Initialize chain
 - Add SPI-BPI Flash
 - MCS file select
 - MCS file select(SPI PROM--W25Q64FV select)
 - MCS file select(SPI PROM--W25Q64FV select)
 - Program
 - Program(OK)
 - Program(Program Succeed)

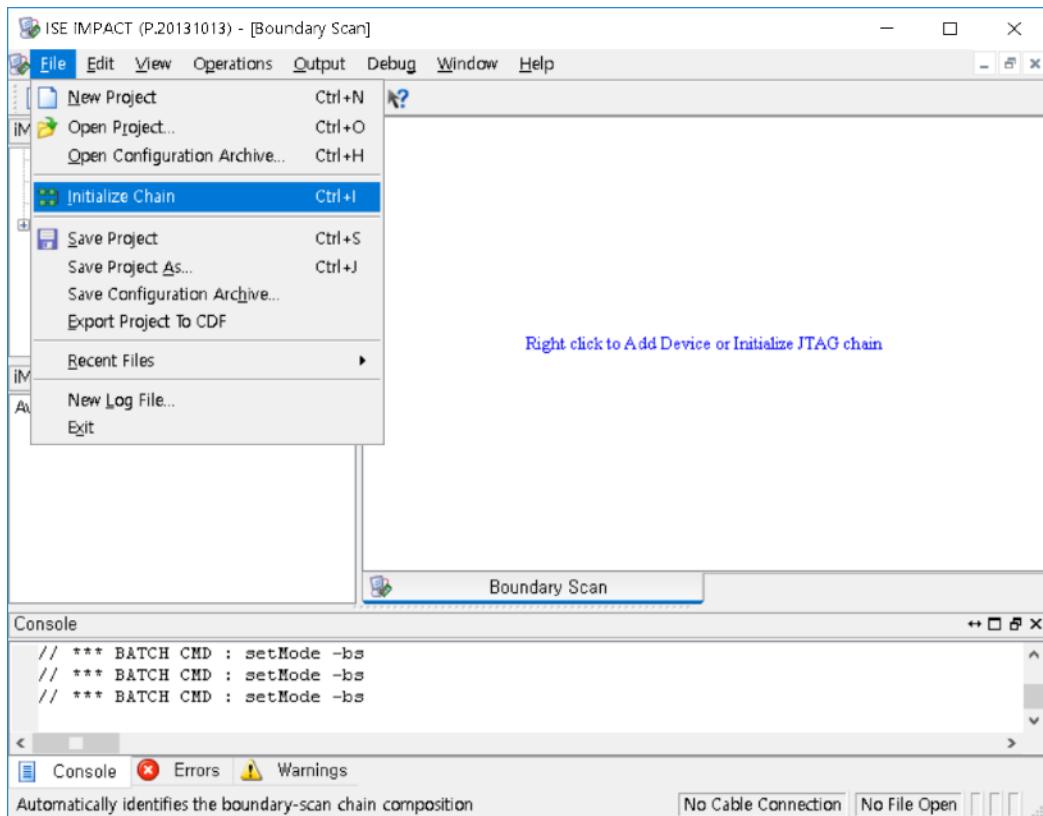


Figure 3.3.7 M-FPD3-QUAD-CAM-UVC 보드 FPGA 펌웨어 업데이트 Initialize chain

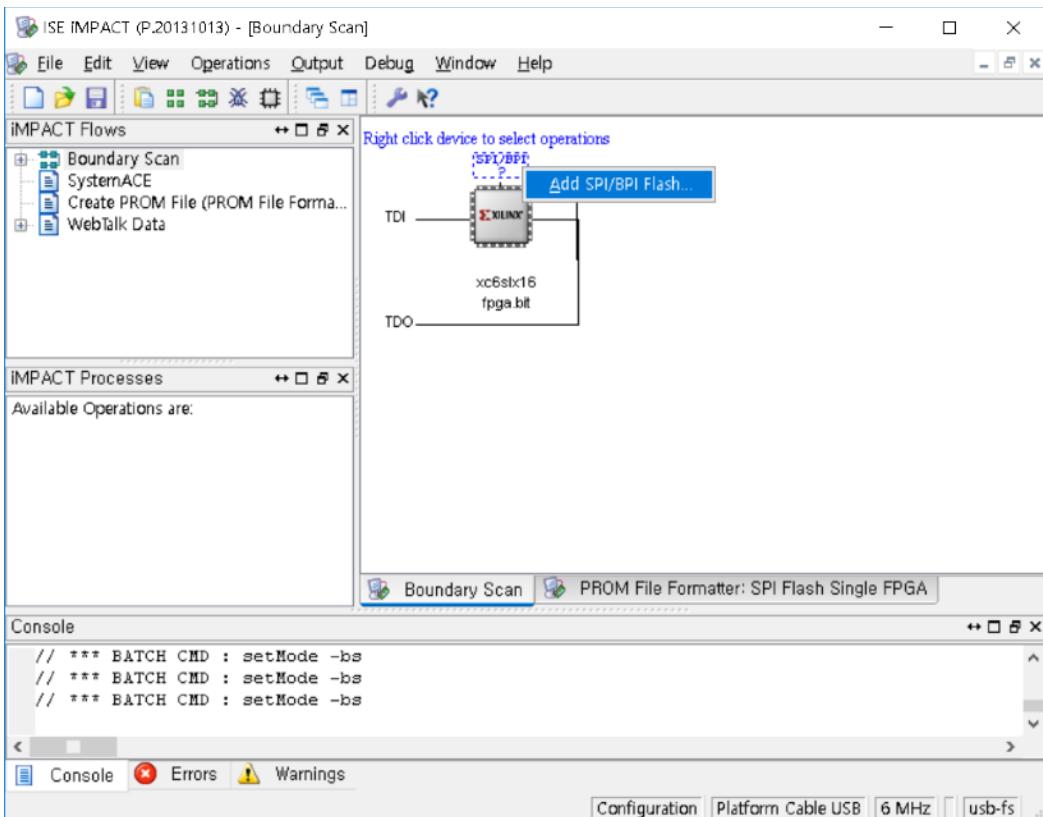


Figure 3.3.8 M-FPD3-QUAD-CAM-UVC 보드 FPGA 펌웨어 업데이트 Add SPI-BPI Flash

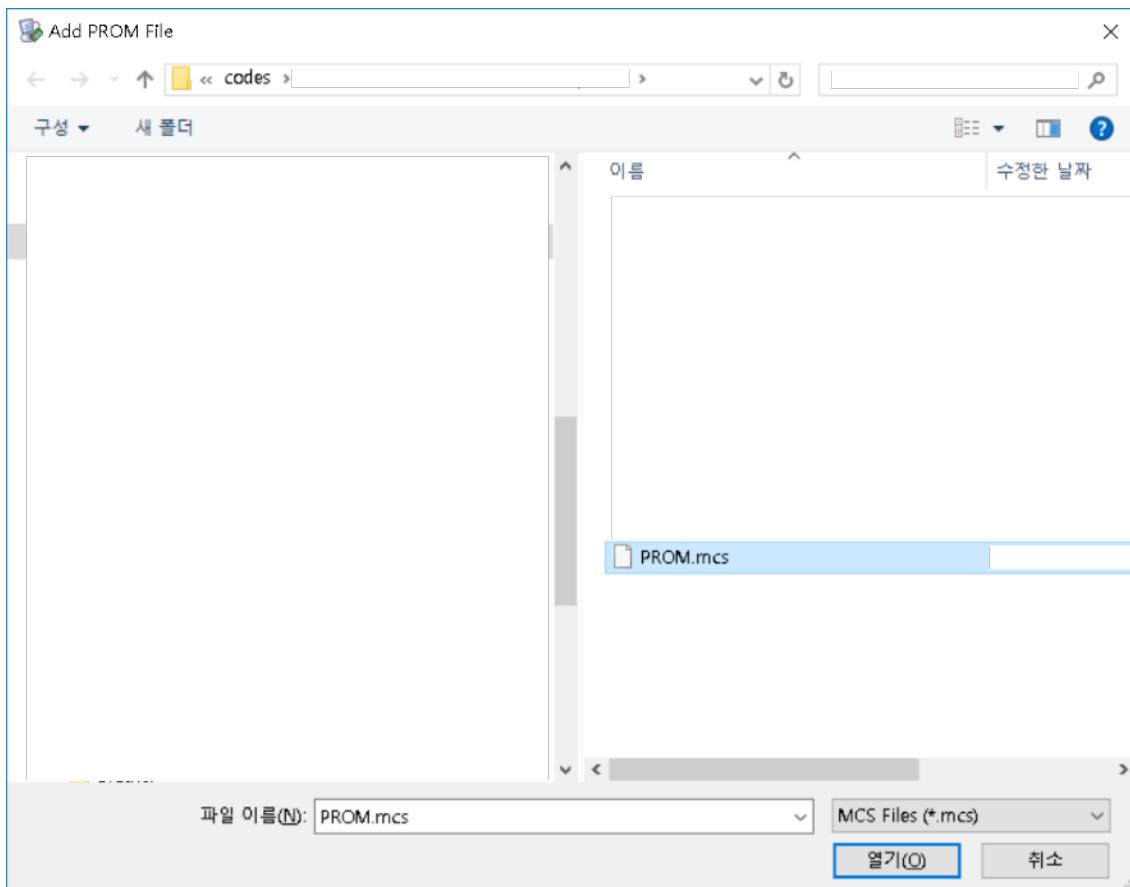


Figure 3.3.9 M-FPD3-QUAD-CAM-UVC 보드 FPGA 펌웨어 업데이트 MCS file select

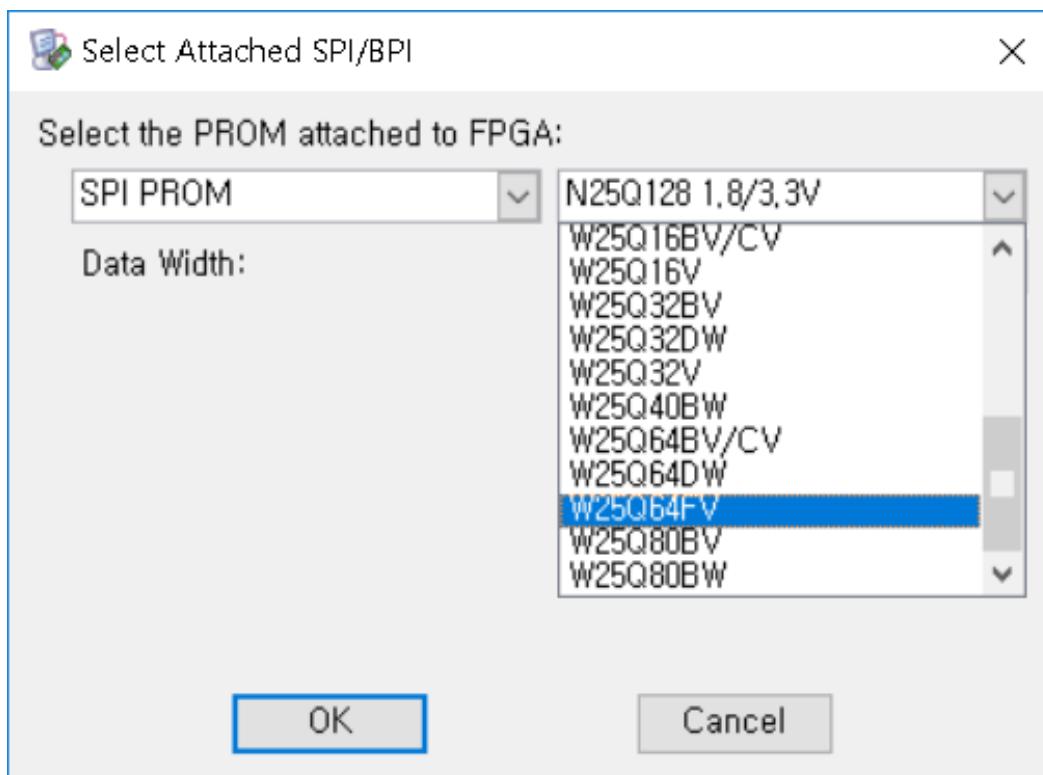


Figure 3.3.10 M-FPD3-QUAD-CAM-UVC 보드 FPGA 펌웨어 업데이트 SPI PROM--W25Q64FV select

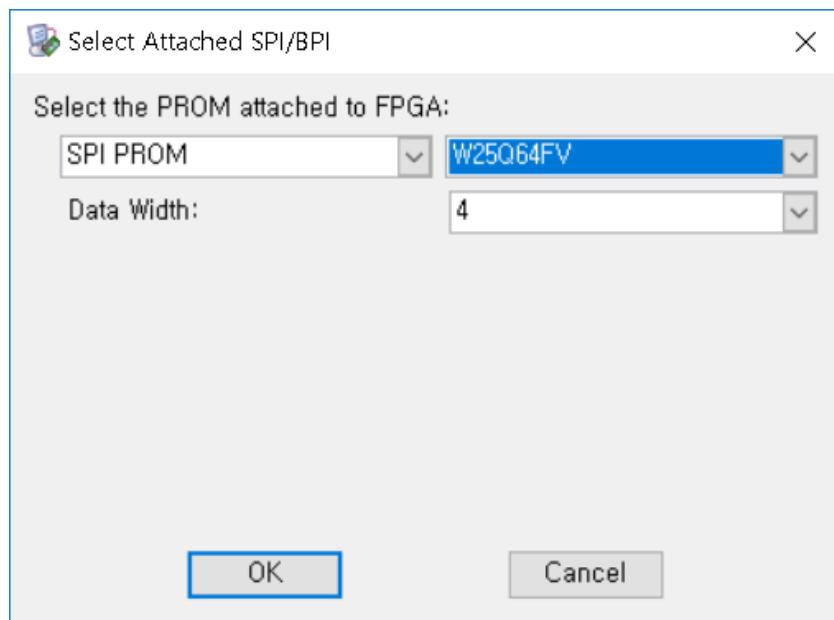


Figure 3.3.11 M-FPD3-QUAD-CAM-UVC 보드 FPGA 펌웨어 업데이트 SPI PROM--W25Q64FV select

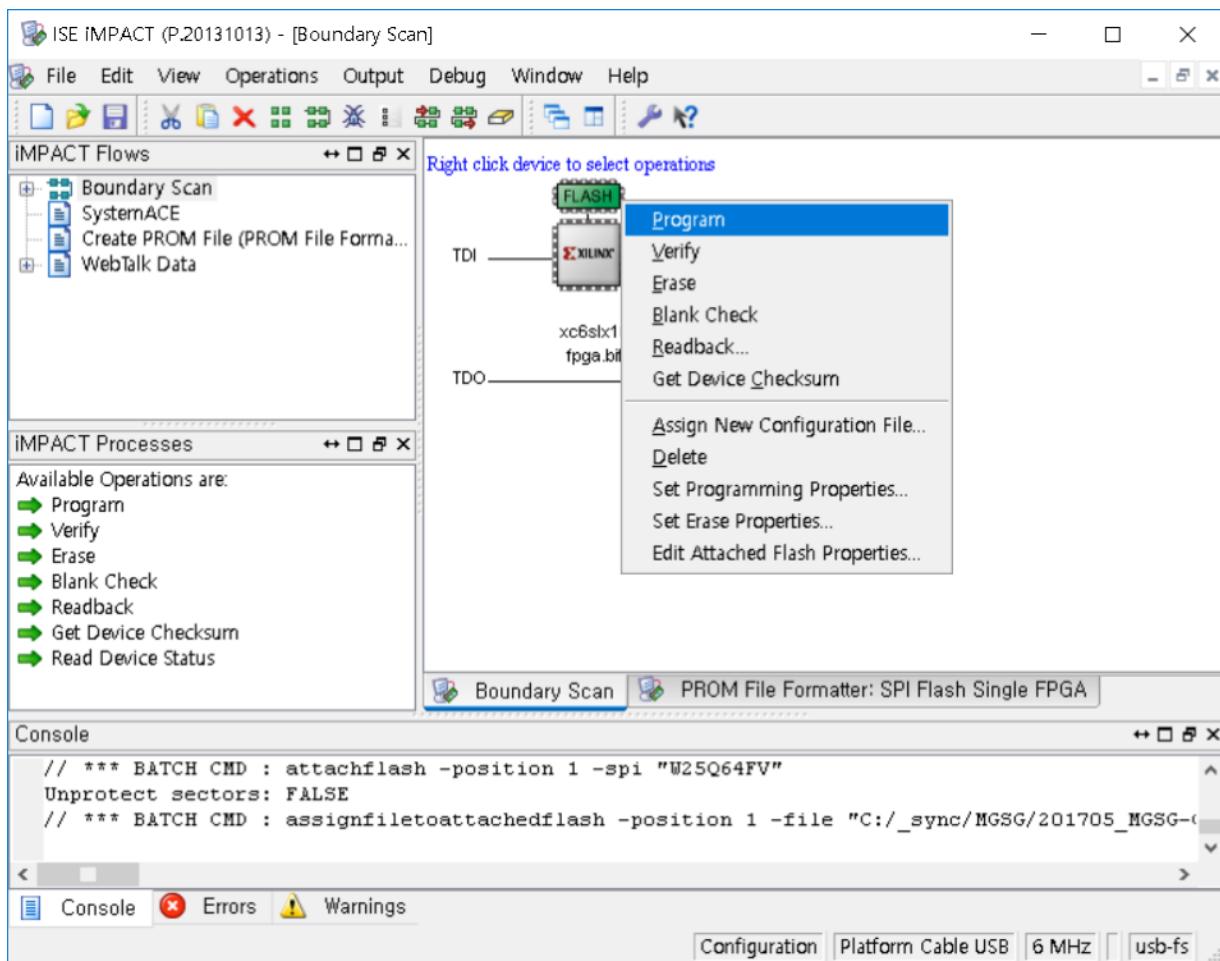


Figure 3.3.12 M-FPD3-QUAD-CAM-UVC 보드 FPGA 펌웨어 업데이트 Program

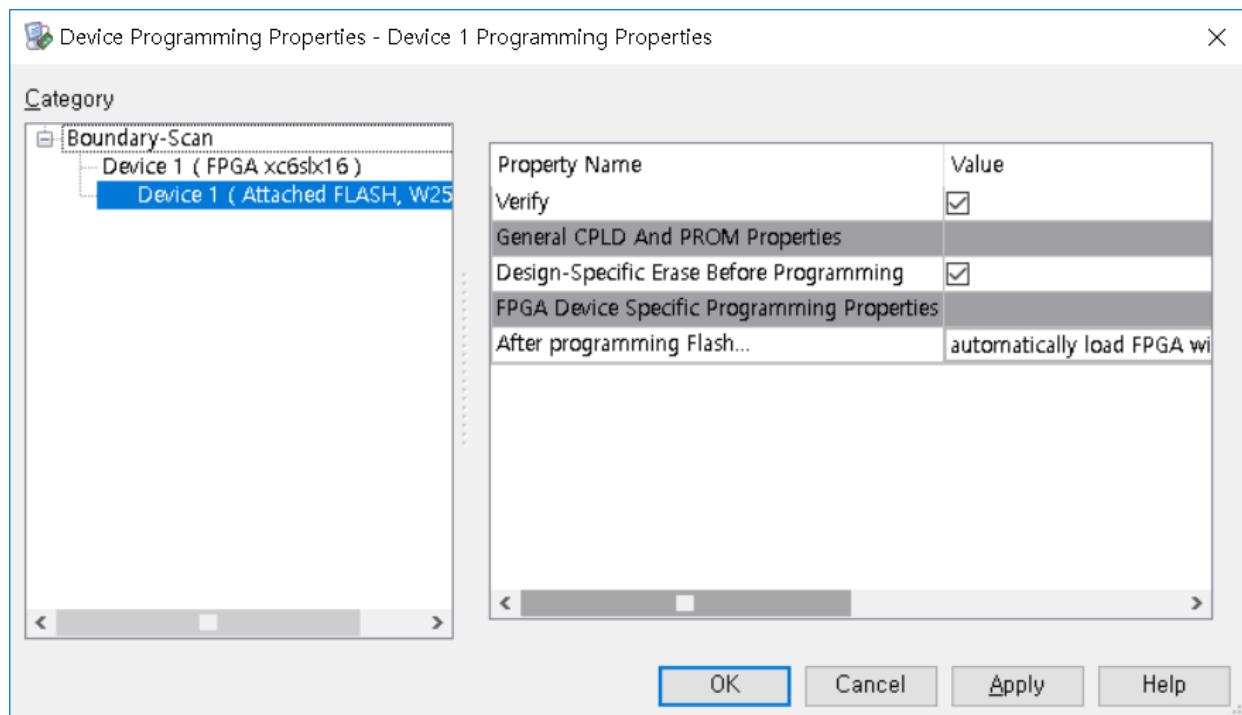


Figure 3.3.13 M-FPD3-QUAD-CAM-UVC 보드 FPGA 펌웨어 업데이트 Program(OK)

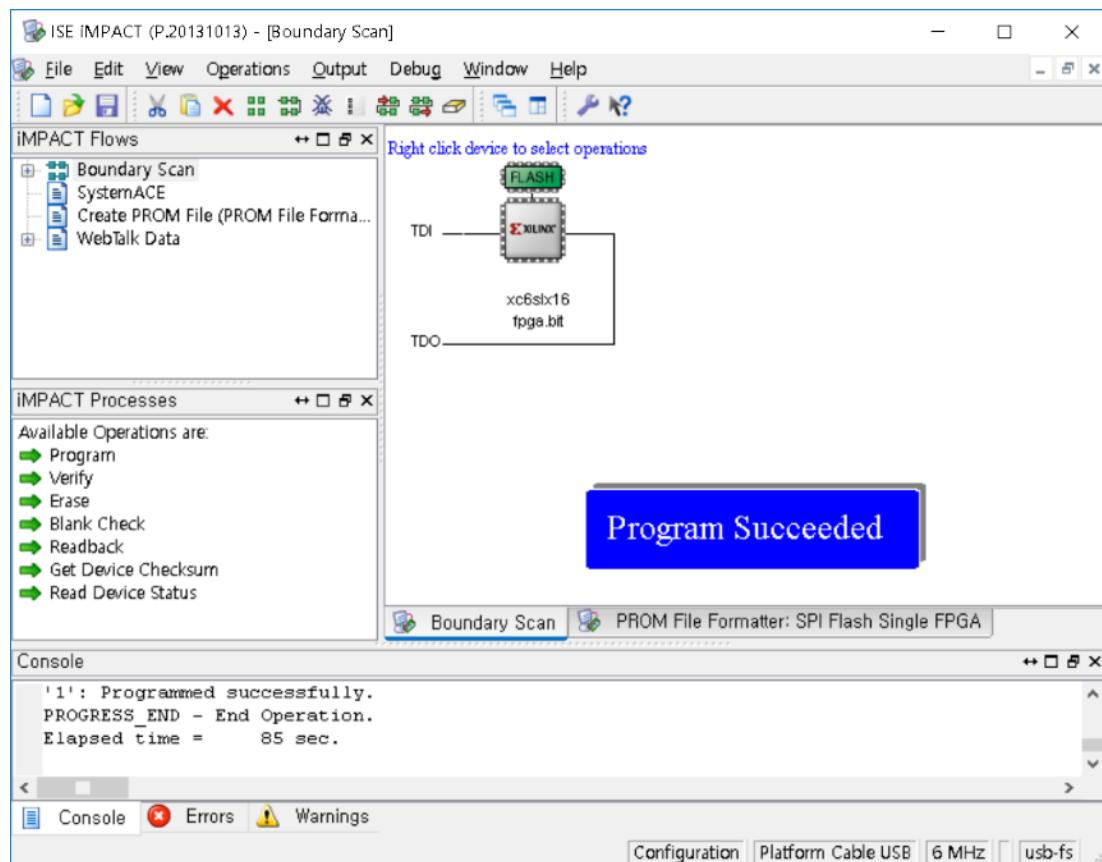


Figure 3.3.14 M-FPD3-QUAD-CAM-UVC 보드 FPGA 펌웨어 업데이트 Program Succeed

3.4 FPGA_USB3.0_UVC 영상 전송 예제

- USB 3.0 펌웨어 업데이트
 - Github 링크 : https://github.com/mgsgo/M-FPD3-QUAD-CAM-UVC/tree/master/rev110/example_FPGA_USB3.0_UVC
 - 동영상 링크 : not yet prepared
- FPGA_USB3.0_UVC 예제 케이블 연결
 - USB 3.0 펌웨어 점퍼 ON
 - UAB 3.0 AM BM cable 연결
 - Xilinx JTAG 연결
 - DC 12V 1A 전원 아답터 연결

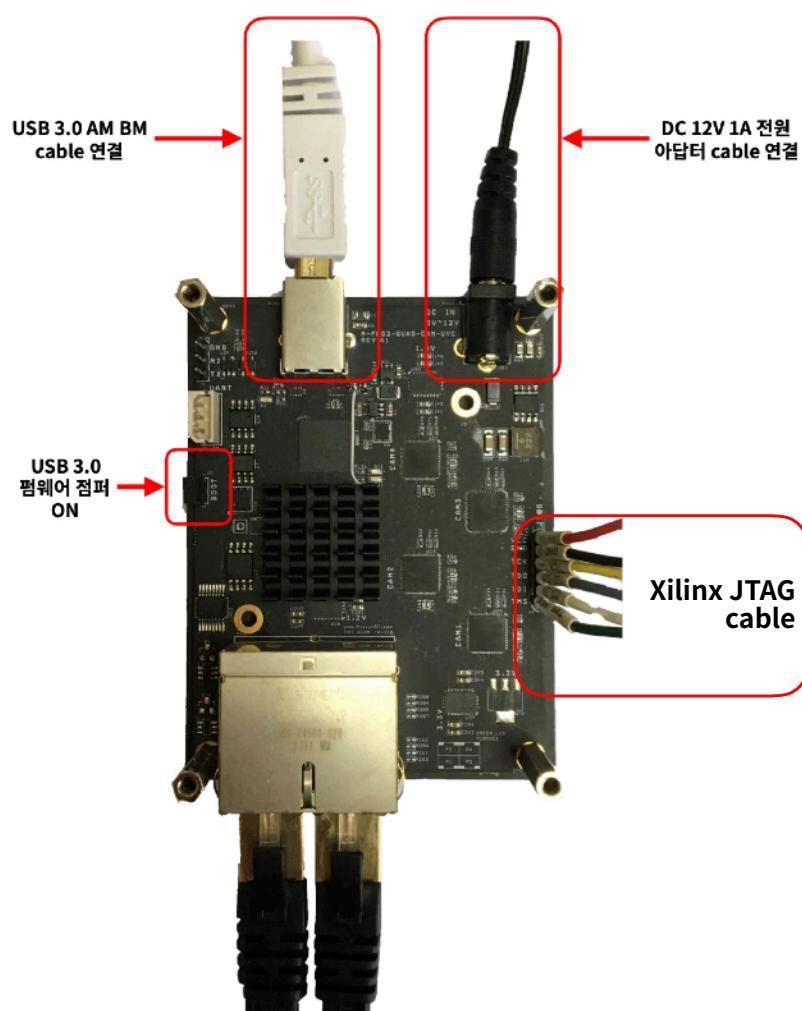


Figure 3.4.1 M-FPD3-QUAD-CAM-UVC 보드 FPGA_USB3.0_UVC 예제 케이블 연결/점퍼 설정

- FPGA 펌웨어 다운로드 (3.3 펌웨어 업데이트 참조)
- USB 3.0 펌웨어 다운로드 (3.3 펌웨어 업데이트 참조)
- Python capture code 실행(1920x1080x30fps)

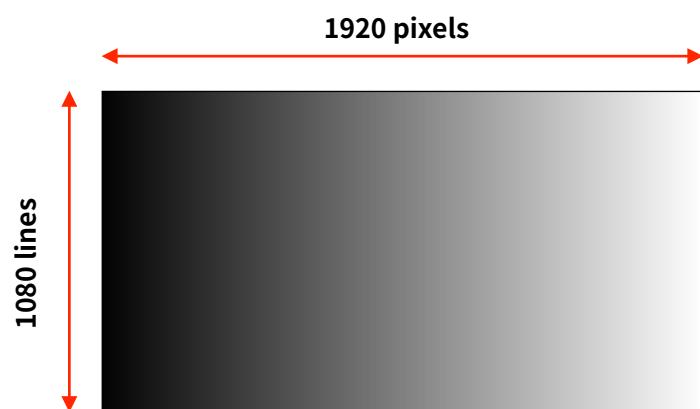


Figure 3.4.2 M-FPD3-QUAD-CAM-UVC 보드 FPGA_USB3.0_UVC 예제 영상 capture 결과

3.5 Deep learning 구동 예제(darknet yolo v3)

- Darknet yolo
 - Link : <https://pjreddie.com/darknet/yolo/>
 - Github 실행방법 링크 : https://github.com/mgsgo/M-FPD3-QUAD-CAM-UVC/tree/master/_rev110/darknet_yolo
 - 예제 실행 동영상 : <https://youtu.be/eqhc5cb-uYA>

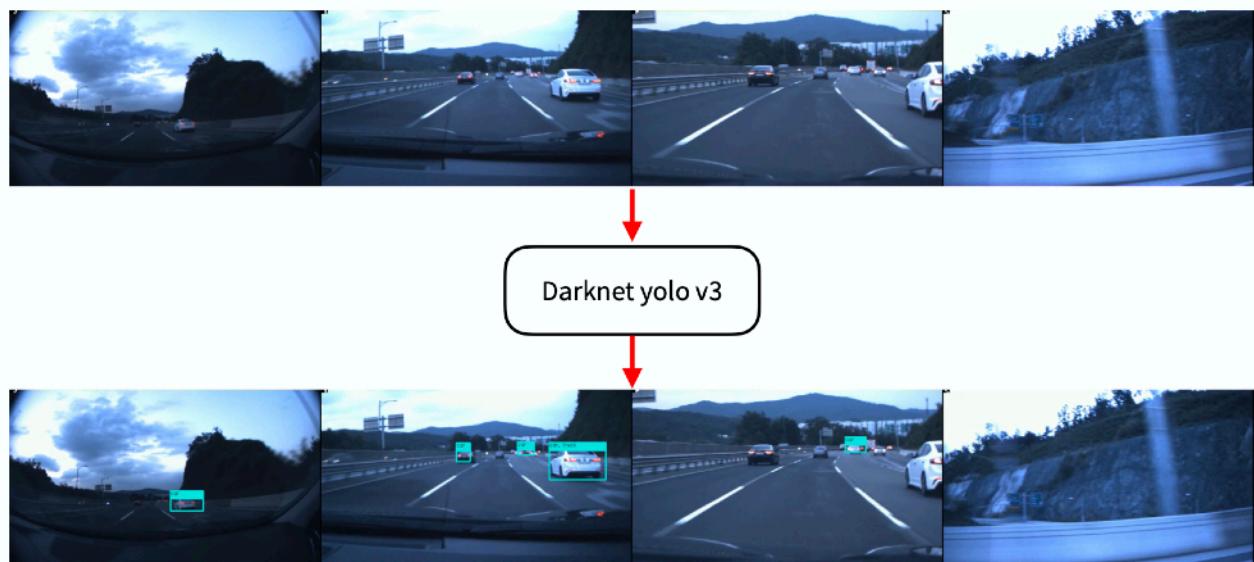


Figure 3.5.1 M-FPD3-QUAD-CAM-UVC 보드 darknet yolo v3 deep learning 코드 수행 화면

3.6 Image stitching 예제(panorama image generate)

- Image stitching
 - Link : <https://www.pyimagesearch.com/2018/12/17/image-stitching-with-opencv-and-python/>
 - Github python 소스코드 링크 : [https://github.com/mgsgo/M-FPD3-QUAD-CAM-UVC/tree/master/rev110/python_panorama\(image_stitching\)](https://github.com/mgsgo/M-FPD3-QUAD-CAM-UVC/tree/master/rev110/python_panorama(image_stitching))

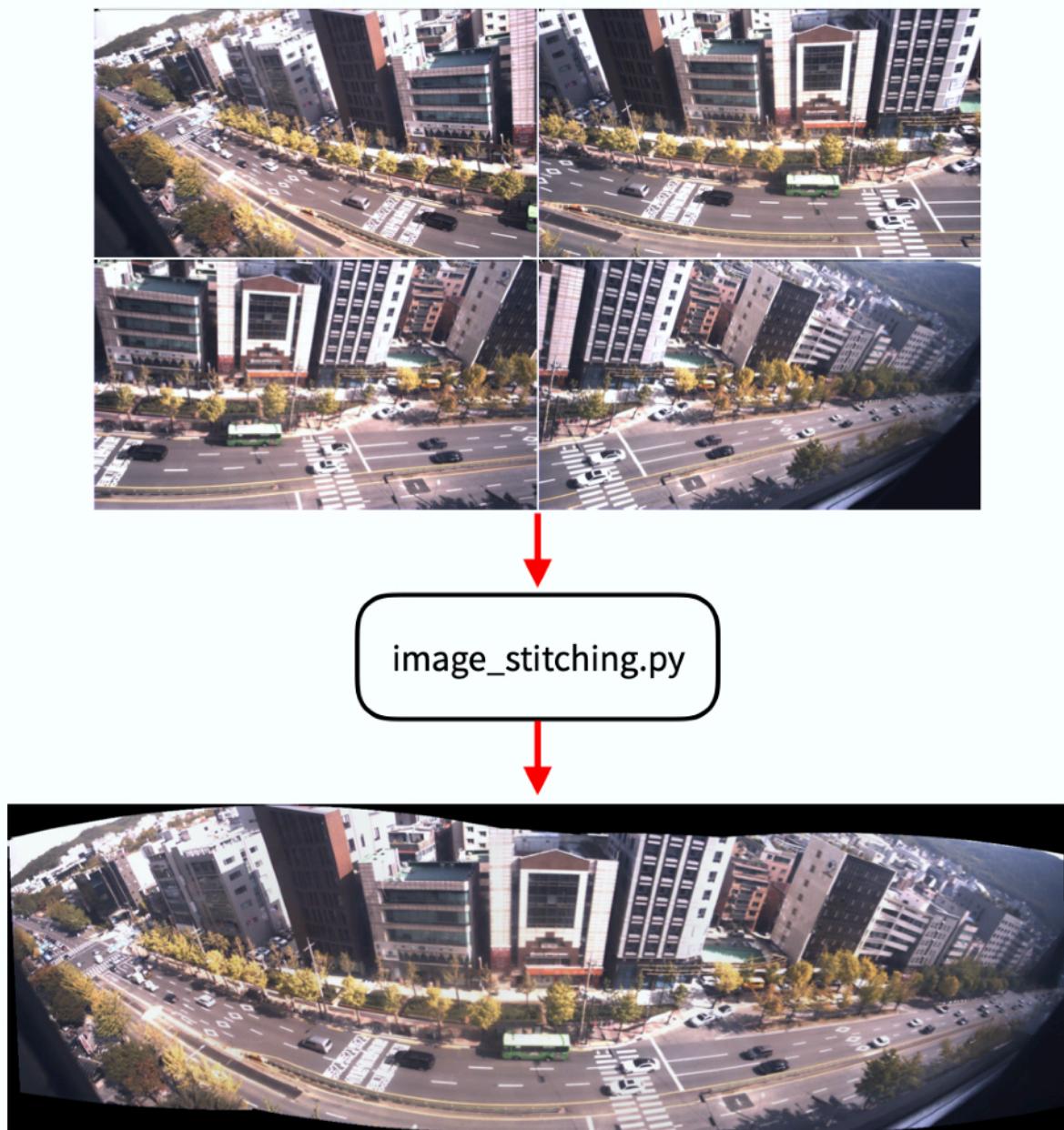


Figure 3.6.1 M-FPD3-QUAD-CAM-UVC 보드 image stitching(panorama image generate) 코드 수행 화면

4. Support

4.1 구매 지원

각 유통업체를 통해서 구매 가능

4.2 기술지원

Contact to : mgsg_opensource@gmail.com

Github page : github.com/mgsgo/M-FPD3-QUAD-CAM-UVC

Homepage : www.mgsgo.com