

MCU 일정 및 진행 상황

1. 일정

2. 진행상황

- 모터 선정을 위한

계산

- 모터 선정

팀장
신호준
팀원
장유미

강유빈

일정

- 총 기간
- 10주 (프로젝트 기간) (5/8 ~ 7/16)
- + 1주(포트폴리오 준비 기간) (7/17 ~ 7/23)
- 여유 기간 2주
- 혹시 모를 추가 사항에 대비한 기간

공기압 모터 ← 포션기

1. 포션 돌리고 : 엔진 1개

포션 모터
(위치)

(FreeRTOS)

GPIO
범용 입출력

MCU

(PID)

BLDC 모터 (속도 제어)

각종 센서 (I²C)

1. 압력 센서 (포)

CAN 통신
DMA 통신

★ 중앙 제어 (Linux)

DSP

영상 처리
(OpenCV)

시스템 프로그래밍
데바이스 드라이버

나머지 양쪽에 엔진 하나씩

→ 엔진 2개 제어

1. 속도, 방향을 각각 제어

2. 가속 단계

1. 포션 식별

→ 거리는? (거리 측정 X, 포션 방향으로만 인식)

2. 거리 식별

도로를 구분 (DMA)

탱크 팀 프로젝트 일정표

	1주	2주	3주	4주	5주	6주	7주	8주	9주	10주
모터 제어 및 조향										
CAN통신 DMA통신										
센서 통신 및 제어(포탑)										
포탑 구현 및 주행 제어										
전원 회로 및 회로 설계										

5월

SUN	MON	TUE	WED	THU	FRI	SAT
	1	2	3	4	5	6
7 Project 시작 및 Motor 계산	8	9	10	11	12	13
14 Motor and ESC 구매	15	16	17	18	19	20
21 Motor 속도 및 조향 제어	22	23	24	25	26	27
28 도 및 조향 제어	29	30	31			

탱크 모터 계산식(이상적인 값)

1초에 100cm를
가는 것을 목표!!

$$W = 15 \text{ kg} \quad D = 5 \text{ cm} \quad f = 2 \text{ 회전/sec}$$

실제 무게: 6.5 kg 자중 (2바퀴감)

$$J = 15 \times 2.5 \times 2.5 / 8 = 11.72$$

$$T_a = 11.72 / 980 \times 2 \times 3.14 \times 6.4 = 0.49$$

$$T_m = 0.1 \times 15 \times 2.5 / 4 = 0.94$$

가속 구간시 필요한 토크 = $0.49 + 0.94 = 1.43$

등속 구간시 필요한 토크 = 0.94

$$1.43 \times 1.5 = \underline{2.15 \text{ kgf}\cdot\text{cm}}$$

안전계수

← mNm 단위 변환 → 210.7 mNm

이상의 토크를 가지는
모터를 선정하자.

$$2\pi r \times 2.5$$

$$= 5\pi = 5 \times 3.14 = 15.7$$

↳ 1바퀴를 굴리면 15.7 cm를 간다.

$$100 \div 15.7 = 6.4$$

↳ 1초에 6.4바퀴 굴려야 함

기어비를 고려한 토크 값 계산.

$$210.7 / (20 \times 0.6) = \underline{4.39 \text{ mNm}}$$

↓ ÷ 9.8 단위 변환

$$0.45 \text{ kgf}\cdot\text{cm}$$

↓ 1.5배 마찰 고려

$$\underline{0.68 \text{ kgf}\cdot\text{cm}}$$

목표 수정! (고려해야할 사항들)

- RPM=50000 이상
- 전력=2kW이상
- 가격=10만원 대. 그 이상은 안됨.
- 제품은 해외제품으로 (단, 중국제품 제외)

탈락한 모터 1



Goolsky

Goolsky 4076 2000KV 4P Sensorless Brushless Motor for 1/8 RC Monster Truck Car

[Be the first to review this item](#)

Price: **\$43.44** & **FREE Shipping**

In Stock.

Get it as soon as **June 7 - 28** when you choose **Standard Shipping** at checkout.

Ships from and sold by [stooctam-us](#).

- 4 poles Hi-torque motor design.
- CNC machined 6061 T6 billet aluminum heatsink can.
- High purity copper windings maximizes efficiency.
- High RPM ABEC5 oversized bearings.
- Universal fit all 1/8 brushless cars-multi-mounting system for M3 or M4 screws.

New (1) from **\$43.44** & **FREE shipping**.



SUMMER

Max Watts: 2.8kW

RPM : 37000

Torque : 7.3T

탈락 이유 : sensorless

탈락한 모터 2



Roll over image to zoom in

Neewer

Neewer® Sensored Inrunner Brushless Motor 2150KV 4-Pole for 1/8 RC Off-road Car Buggy

[Be the first to review this item](#)

Price: **\$60.81** & **FREE Shipping**. [Details](#)

Only 1 left in stock - order soon.

Want it Thursday, May 18? Order within **3 hrs 10 mins** and choose **Two-Day Shipping** at checkout. [Details](#)

Sold by [Global_omo](#) and [Fulfilled by Amazon](#). Gift-wrap available.

- 2150kv 4 pole 12 slots
- 3 Chase 4 Poles
- Sensored Brushless Motor
- For 1/8 RC Car
- High torque and efficiency

New (2) from **\$60.81** & **FREE shipping**. [Details](#)



Toys delivered
monthly [Learn more](#)

Max Watts: 2kW

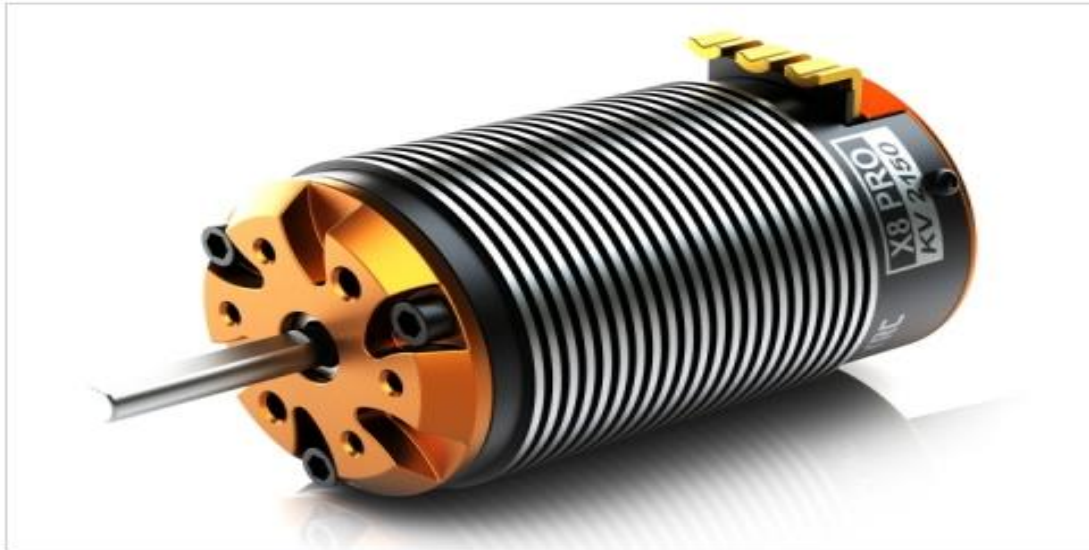
RPM : 47730

Torque : 4T

탈락 이유 · 토크가 4T로 기준보다 낮음

후보 모터 1

TORO X8 PRO FOR 1/8 BUGGY



SKYRC

Model: SK-400009-11-12

Max Watts: 2.4kW

RPM : 45000

Torque : 5T

RPM을 낮추면 토크가 올라감

Introduction

Specifications

Manual

Item	KV (RPM/ VOLT)	Watts	Max Voltage	Max Current	Internal Resistance (ohm)	Max RPM	Turns	Net Weight
SK-400009-11	2350	2400	19V	126A	0.0062	45000	1Y	350g
SK-400009-12	2150	2400	21V	114A	0.0071	45000	2D	350g

중국제품이라는 점에서 신뢰성이 낮음

후보 모터 2



JIMI RC

GoolRC High Performance 4274 2250KV Sensored Brushless Motor for 1/8 RC Car Monster Truck Bigfoot

[Be the first to review this item](#)

Price: **\$57.85** & **FREE Shipping**

In Stock.

Get it as soon as **June 7 - 28** when you choose **Standard Shipping** at checkout.

Ships from and sold by [stooctam-us](#).

- Great quality, Stable performance
- Perfect after-sales service
- Manufacturer directly supply products.

New (2) from **\$57.85** & **FREE shipping**.

Max Watts: 2.8kW

RPM : 49500

Torque : 5.5T

RPM을 낮추면 토크가 올라감

후보 모터 3

Hobbywing XERUN 120A SCT PRO ESC w/ TenShock 2450KV 3.5Y 6 Pole Sensored Brushless Motor Black C



Hobbywing XERUN 120A SCT PRO ESC
w/ TenShock 2450KV 3.5Y 6 Pole
Sensored Brushless Motor Black Combo
#CB0977

[CB0977]

Price: **USD\$163.30** ~~USD\$238.90~~

You Save: **\$75.60 (32%)**

Stock: 20+

Buy this item and earn 163 points valued at USD\$0.65.

Max Watts: 3.5kW

RPM : 51450

Torque : 6.6T

- RPM을 낮추면 토크가 올라감
- ESC combo로 따로 ESC를 찾지 않아도 된다는 장점이 있음

끝

DSP 일정 및 계획 발표

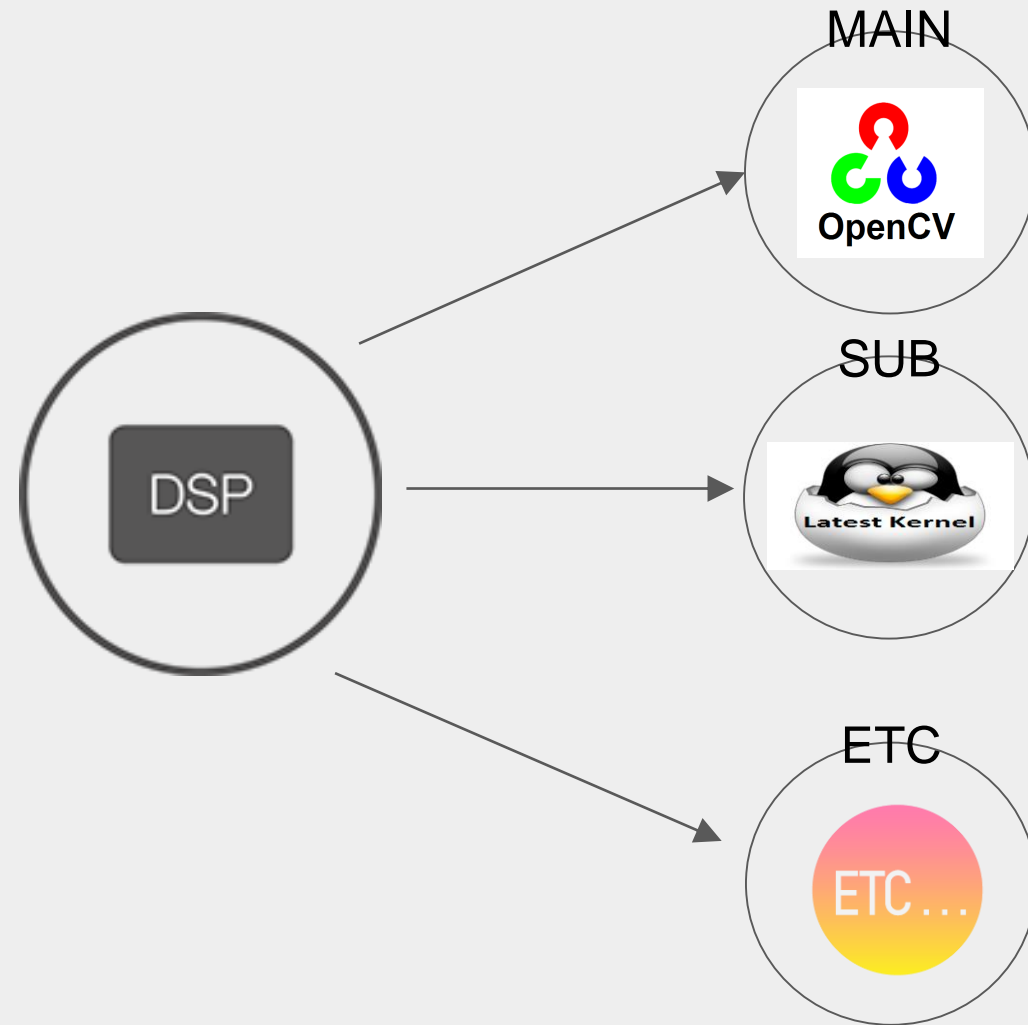
발표자 : 전병용

목차

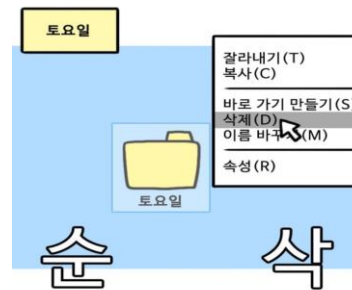
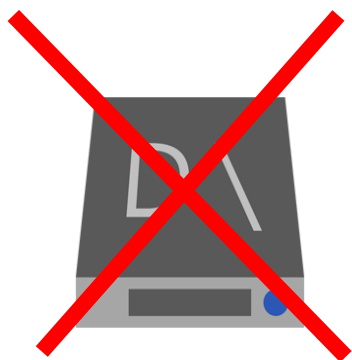
1. 소개
2. 현재 진행중
3. 추후 계획

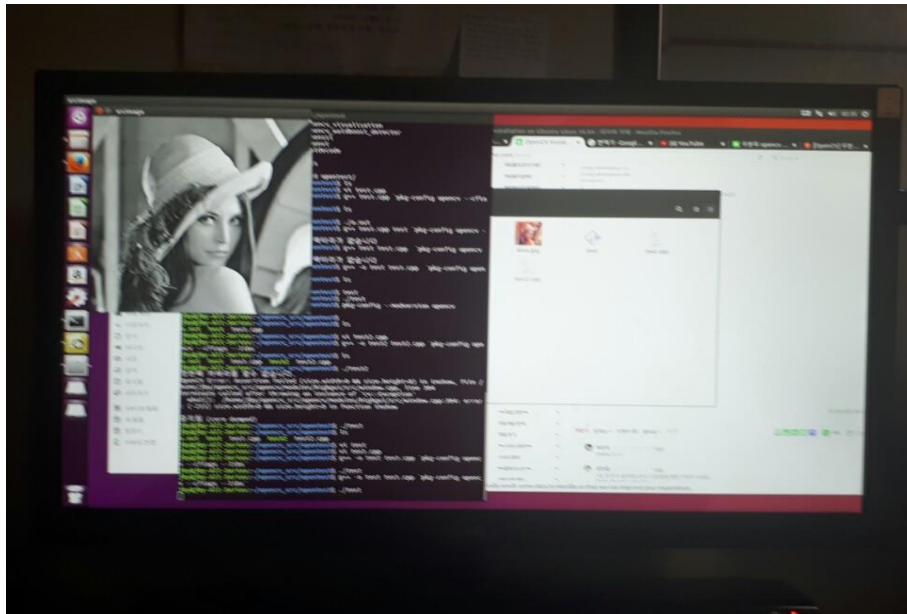
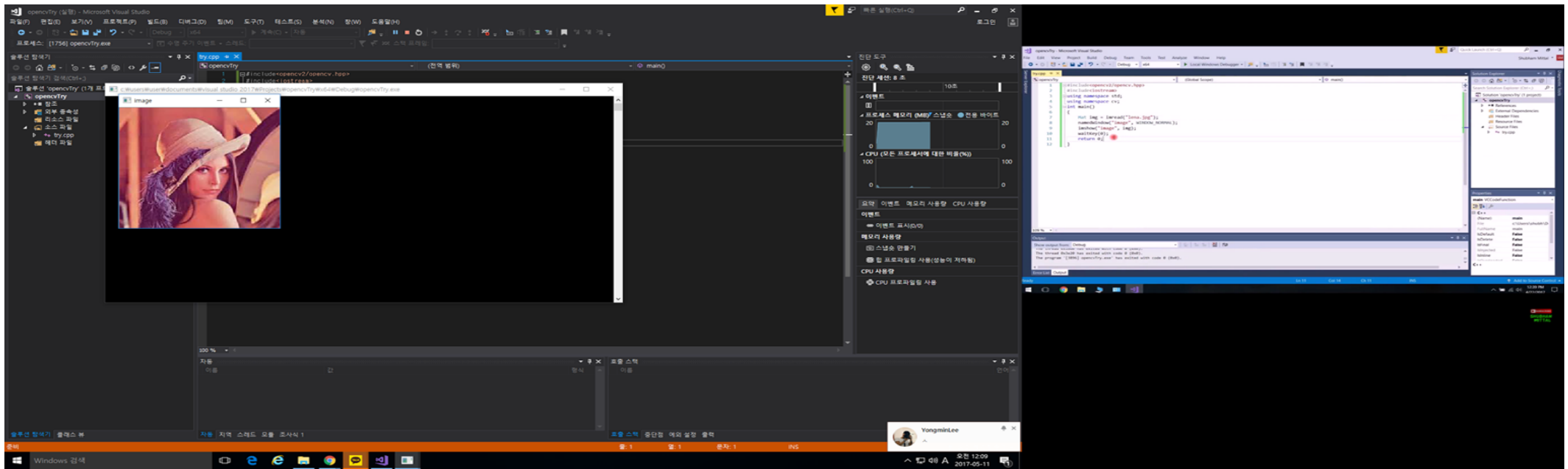
소개

TANK

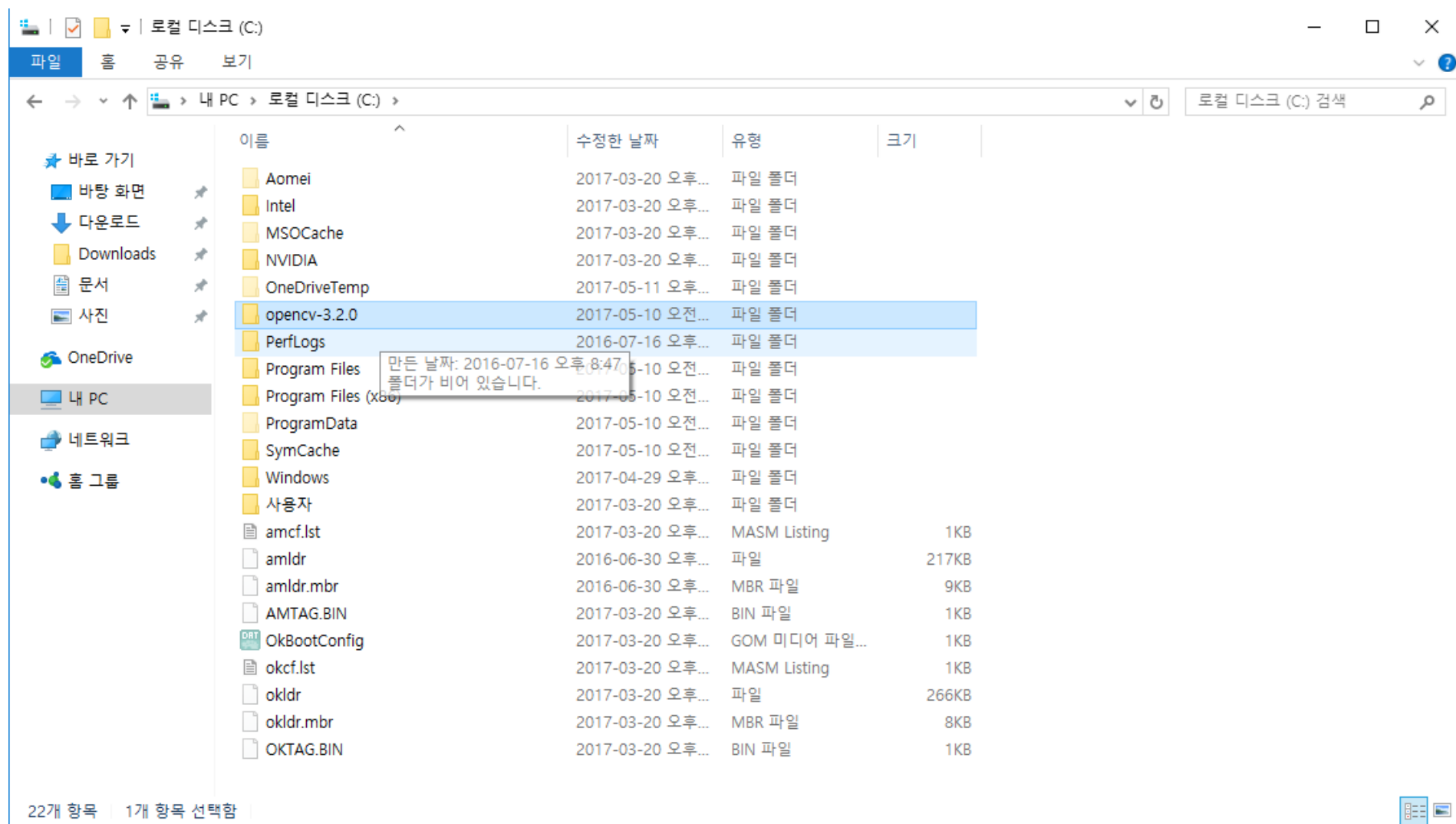


현재 진행중

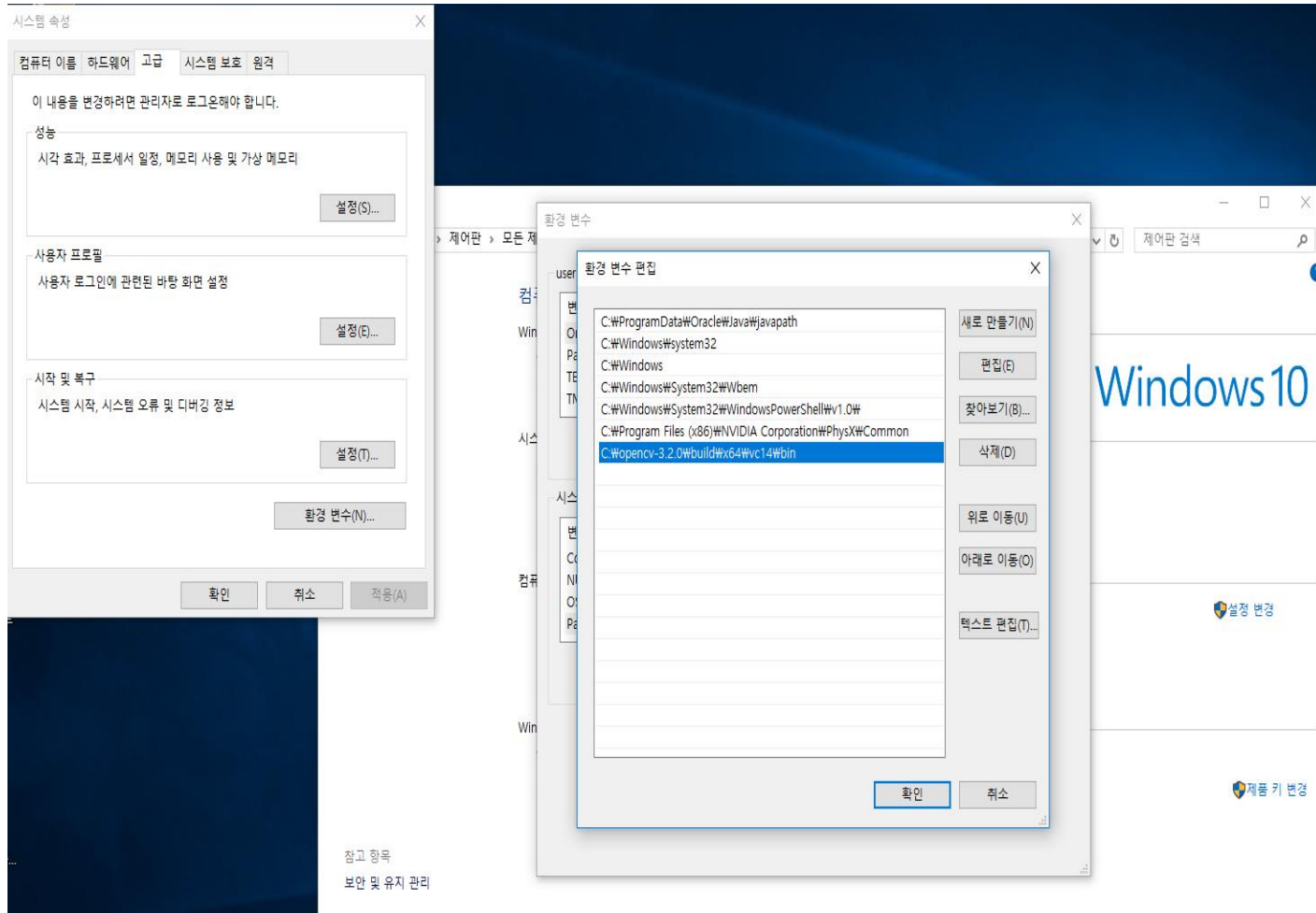




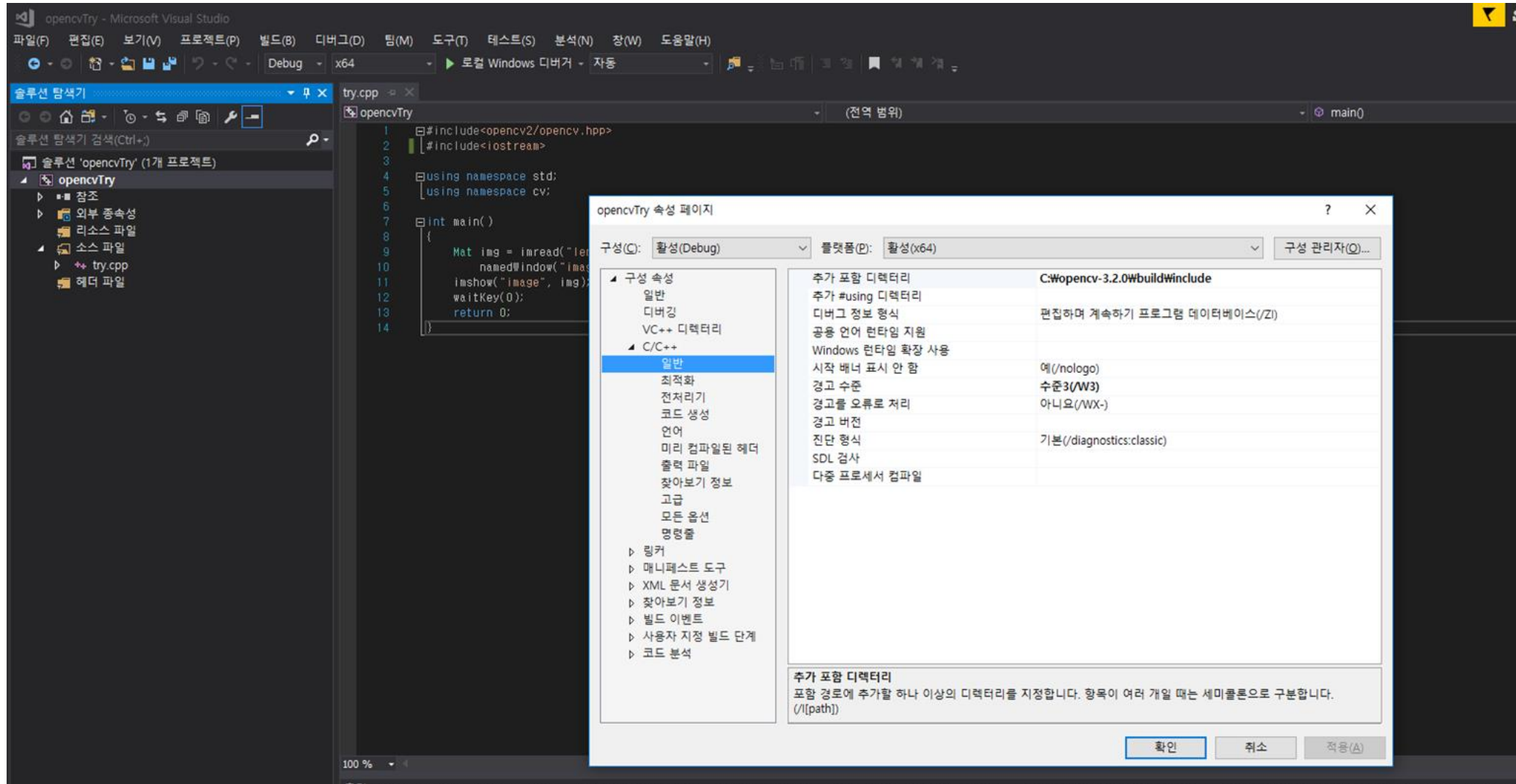
설치 과정(윈도우)



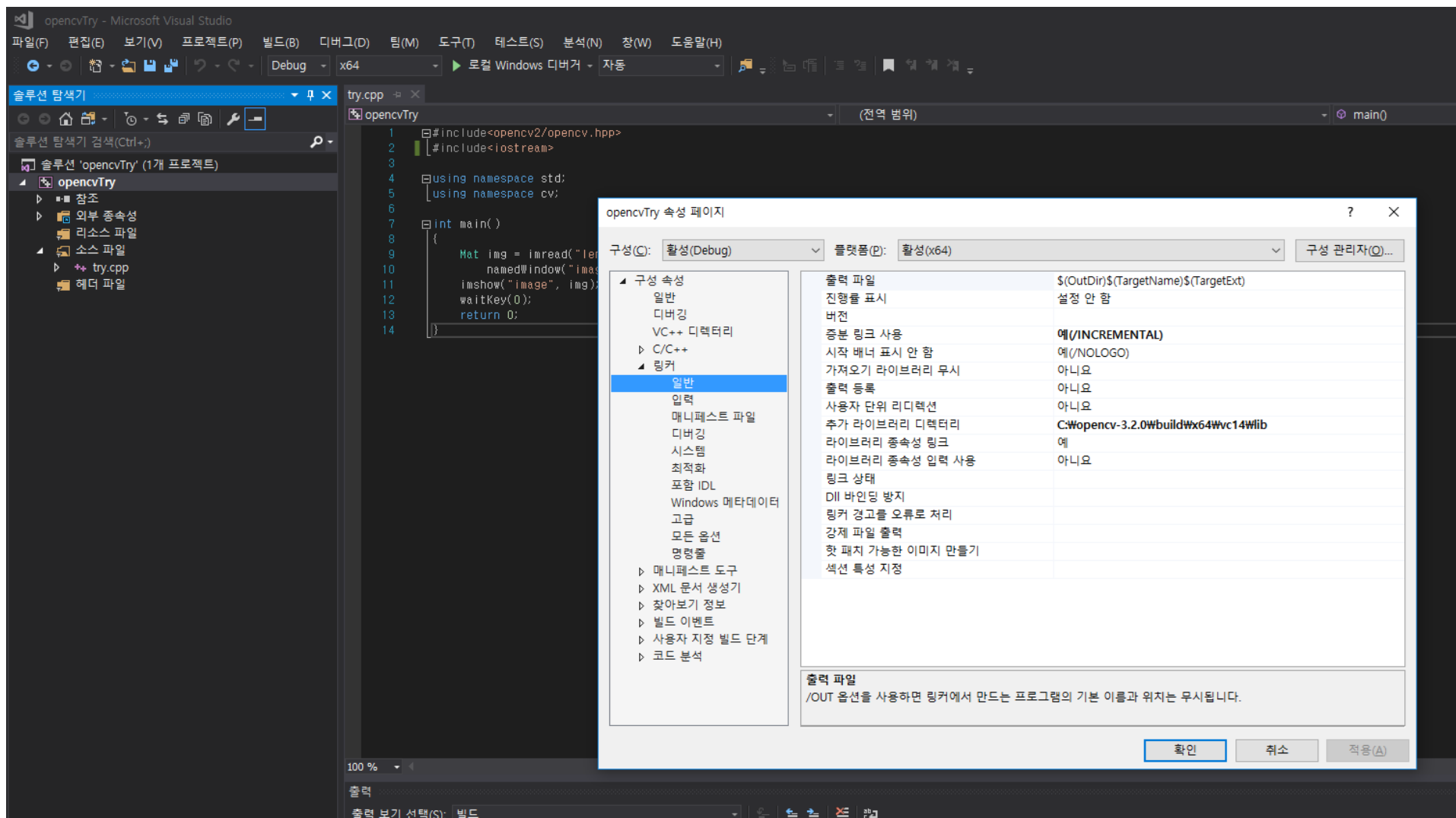
설치 과정



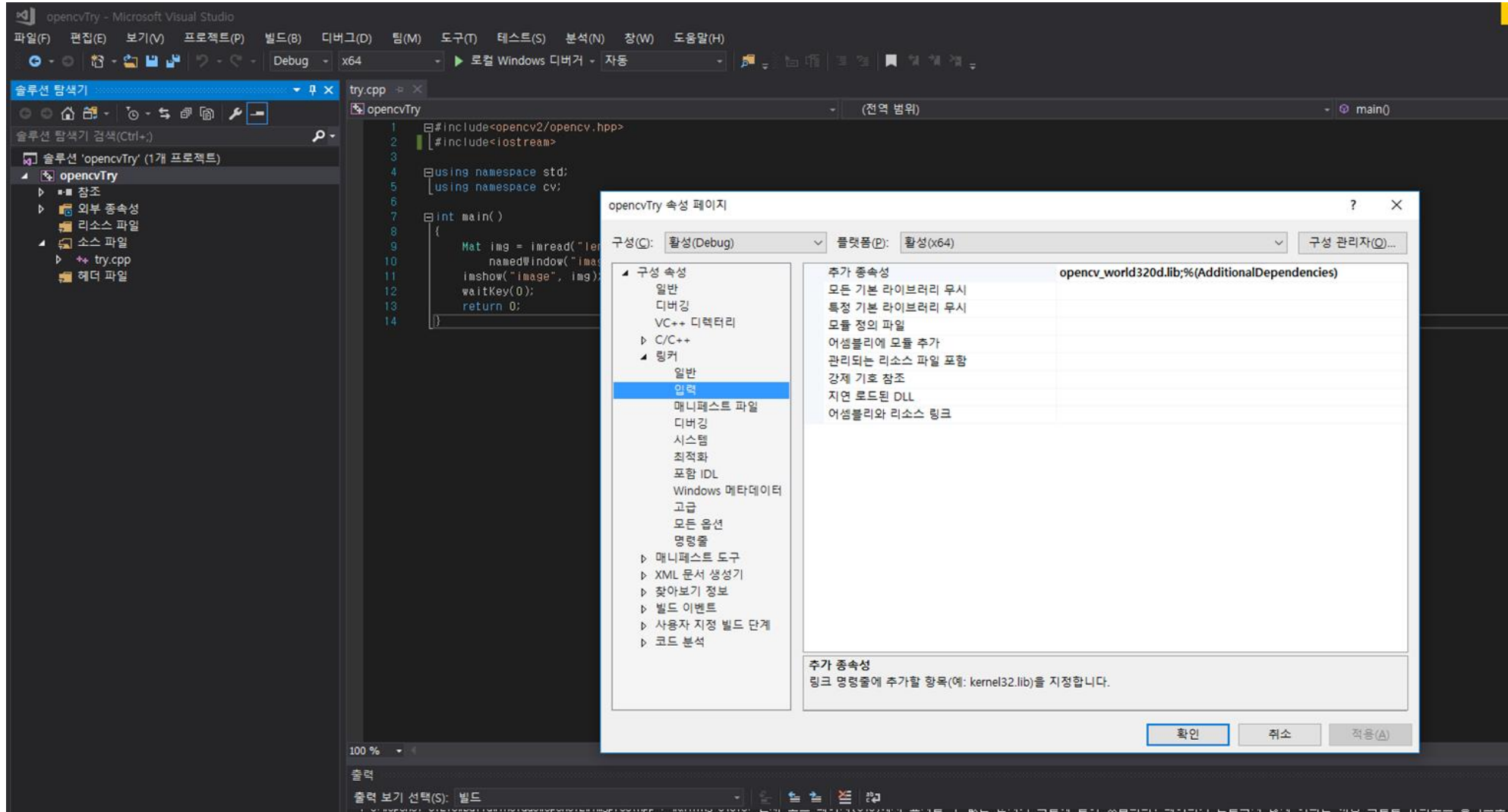
설치 과정



설치 과정



설치 과정



설치 과정(우분투)

1. `sudo apt-get update`
2. `sudo apt-get install build-essential checkinstall cmake git pkg-config yasm libtiff5-dev libjpeg-dev libjasper-dev libavcodec-dev libavformat-dev libswscale-dev libdc1394-22-dev libxine2-dev libgstreamer0.10-dev libgstreamer-plugins-base0.10-dev libv4l-dev python-dev python-numpy libtbb-dev libgtk2.0-dev libfaac-dev libmp3lame-dev libopencore-amrnb-dev libopencore-amrwb-dev libtheora-dev libvorbis-dev libxvidcore-dev x264 v4l-utils libopenexr-dev python-tk libeigen3-dev libx264-dev`
3. `sudo add-apt-repository ppa:mc3man/gstffmpeg-keep`
4. `sudo apt-get update`
5. `sudo apt-get install ffmpeg gstreamer0.10-ffmpeg`
6. `mkdir opencv_src`
7. `cd opencv_src`
8. `git clone https://github.com/opencv/opencv.git`
9. `git clone https://github.com/opencv/opencv_contrib.git`
10. `cd opencv`
11. `mkdir build`
12. `cd build`
13. `cmake -D CMAKE_BUILD_TYPE=RELEASE -D CMAKE_INSTALL_PREFIX=/usr -D BUILD_NEW_PYTHON_SUPPORT=ON -D WITH_V4L=ON -D WITH_TBB=ON -D WITH_IPP=OFF -D OPENCV_EXTRA_MODULES_PATH=../../opencv_contrib/modules ..`
14. `make -j $(nproc)`
15. `sudo make install`
16. `sudo ldconfig`
17. `pkg-config --modversion opencv`
18. 아래 코드를 입력한다.
19. `g++ ~~~.cpp `pkg-config opencv --cflags --libs``
20. 결과를 확인한다.

18. 아래 코드를 입력한다.

```
#include <opencv2/opencv.hpp>
using namespace cv;
using namespace std;
int main()
{
    Mat srcImage =
    imread("adf4159.jpg", IMREAD_GRAYSCALE);
    if( srcImage.empty() )
        return -1;

    imshow("srcImage", srcImage);
    waitKey();
    return 0;
}
```

19. `g++ ~~~.cpp `pkg-config opencv --cflags --libs``

20. 결과를 확인한다.


```
try.cpp -> X
opencvtry - (전역 범위) - 0 main()

1 #include<opencv2/opencv.hpp>
2 #include<iostream>
3 using namespace std;
4 using namespace cv;
5 int main()
6 {
7     Point2f pt1(0.1f, 0.2f), pt2(0.3f, 0.4f);
8
9     Point pt3 = (pt1 + pt2).toInt();
10    Point2f pt4 = (pt1 - pt2).toInt();
11    Point pt5 = Point(10, 10);
12    Point2f pt6 = Point2f(10.0f, 10.0f);
13
14    cout << "pt1:" << pt1 << endl;
15    cout << "pt2:" << pt2 << endl;
16    cout << "pt3:" << pt3 << endl;
17    cout << "pt4:" << pt4 << endl;
18    cout << "pt5:" << pt5 << endl;
19    cout << "pt6:" << pt6 << endl;
20
21
22    if (pt1 == pt2)
23        cout << "pt1 is equal to pt2" << endl;
24    else
25        cout << "pt1 is not equal to pt2" << endl;
26
27    float fValue = pt1.dot(pt2);
28    cout << "fValue = " << fValue << endl;
29
30    double normValue = norm(pt1);
31    cout << "normValue = " << normValue << endl;
32
33    Point pt(150, 150);
34    Rect rect(100, 100, 200, 200);
35    if (pt.inside(rect))
36        cout << "pt is an inside point in rect" << endl;
37    else
38        cout << "pt is not an inside point in rect " << endl;
39    return 0;
40 }
```

C:\Windows\System32\cmd.exe

```
***** VIDEOINPUT LIBRARY - 0.1995 - TPU07 *****

pt1:[0.1, 0.2]
pt2:[0.3, 0.4]
pt3:[4, 6]
pt4:[-2, -2]
pt5:[10, 10]
pt6:[10, 10]
pt1 is not equal to pt2
fValue = 0.11
normValue = 0.223607
pt is an inside point in rect
계속하려면 아무 키나 누르십시오 . . .
```

```
opencvtry - Microsoft Visual Studio
파일(F)  편집(E)  프로젝트(P)  빌드(B)  디버그(D)  도구(T)  텍스트(T)  분석(A)  창(W)  도움말(H)
Debug - x64 -> 로딩 Windows 디버거 - 자동
솔루션 탐색기
솔루션 탐색기 검색(CH+):
솔루션 opencvtry (1개 프로젝트)
  opencvtry
    참조
    외부 종속성
    리소스 파일
    소스 파일
    try.cpp
    헤더 파일

try.cpp -> X
opencvtry - (전역 범위) - 0 main()

1 #include<opencv2/opencv.hpp>
2 #include<iostream>
3
4 using namespace std;
5 using namespace cv;
6
7 int main()
8 {
9     Mat A1(1, 2, DataType_uchar::type);
10    //Mat A1(1, 2, CV_8U);
11
12    A1.at<uchar>(0, 0) = 1;
13    A1.at<uchar>(0, 1) = 2;
14    cout << "A1" << endl;
15    cout << "depth" << A1.depth() << ", "
16        << "channels" << A1.channels() << endl;
17
18    Mat A2(1, 2, DataType_Vec3_uchar::type);
19    //Mat A2(1, 2, CV_30C3);
20    A2.at<Vec3_uchar>(0, 0) = Vec3d(10, 20, 30);
21    A2.at<Vec3_uchar>(0, 1) = Vec3d(40, 50, 60);
22    cout << "A2" << A2 << endl;
23    cout << "depth" << A2.depth() << ", "
24        << "channels" << A2.channels() << endl;
25
26    Mat B(1, 2, DataType_float::type);
27    //Mat B(1, 2, CV_32F);
28    B.at<float>(0, 0) = 10.0f;
29    B.at<float>(0, 1) = 20.0f;
30    cout << "B" << B << endl;
31    cout << "depth" << B.depth() << ", "
32        << "channels" << B.channels() << endl;
33
34    Mat C(1, 2, DataType_Point::type);
35    //Mat C(1, 2, CV_32C2);
36    C.at<Point>(0, 0) = Point(100, 200);
37    C.at<Point>(0, 1) = Point(200, 200);
38    cout << "C" << C << endl;
39    cout << "depth" << C.depth() << ", "
40        << "channels" << C.channels() << endl;
41
42    Mat D(1, 2, DataType_complex_double::type);
43    //Mat D(1, 2, CV_64FC2);
44    D.at<complex<double>>(0, 0) = complex<double>(10.0, 20.0);
45    D.at<complex<double>>(0, 1) = complex<double>(10.0, 20.0);
46    cout << "D" << D << endl;
47    cout << "depth" << D.depth() << ", "
48        << "channels" << D.channels() << endl;
49
50    return 0;
51
52 }
```

추후 계획

