CODESYS 와 컴파일파이 시작가이드

컴파일파이(라즈베리파이)에서 쓸만한 개발환경을 찾고 계시다면 CODESYS 를 추천합니다. 세계 주요 메이저 PLC 메이커에서 사용하고 있는 CODESYS 는 IEC61131-3 표준을 성실히 구현하였을 뿐만 아니라, 지난 25 년간의 꾸준한 업그레이드로 안정적이면서도 파워풀한 성능을 가지고 있는 SOFTPLC 개발환경 소프트웨어입니다.

그동안 PC 급에서 실행되던 CODESYS 는 런타임 라이선스 가격이 비싸, 일반 사용자들은 쉽게 접근할 수 없었지만, 수년전 Raspberry Pi 버전을 출시하면서 진입장벽이 낮아졌습니다. 저희 회사의 컴파일 파이 제품은 라즈베리파이를 산업용으로 재해석하여 만든 터치패널 PC 제품이므로, 여기에서 라즈베리용 CODESYS 가 매우 잘 실행되고 있습니다. 본 책은 컴파일 공식 블로그인 http://blog.naver.com/cubloc 에 있는 CODESYS 강좌를 읽기 쉽게 정리한 PDF 문서입니다. 모쪼록 고객 여러분의 실무에 도움이 되었으면 합니다.

컴파일 테크놀로지 주식회사 대표이사

목차

제 1 장. CODESYS 맛보기

CODESYS란?

CODESYS 가 뭐냐면 자동화 개발 환경 소프트웨어입니다. Visual Studio 와 비슷하다고 보시면 됩니다. 단 IEC61131-3 표준에서 제시한 ST 언어, LD 언어를 지원합니다. ST 언어는 PASCAL 하고 비슷한데, 제가 보기엔 C 언어하고도 매우 비슷합니다. LD 언어는 PLC 에서 쓰던 레더로직입니다.

둘 다 알아야 되는 건 아닙니다. 여러개 언어를 지원하는데 그중 편한걸 골라서 한개만 써도 됩니다. C 언어를 아시는 분은 ST 언어 쓰시고, 레더로직만 아시는 분은 LD 또는 펑션블록만 쓰면 됩니다.

제가 가장 맘에 드는 점은 Beckhoff 같은 큰 회사가 사용하고 있는 만큼, 안정성과 신뢰성이 확보되었다는 점입니다. 보통은 C# 을 사용해서 프로그램을 짜는데, 사실 C#만 할줄 아는 사람에 부탁해서 어찌어찌 코딩을 했다고 하더라도, 현장운영시 삐긋거리는 일들이 다반사죠. C#으로 코딩하려면 가장 기본적인 부분부터 다 코딩을 해야해서, 필드버스 통신 프로토콜까지도 손대 주어야 하기 때문에 이 부분에서 문제가 많이 발생합니다.

그런데, CODESYS 는 풍부한 필드버스 프로토콜을 지원하고 있고, 안정화되어 있다는 얘기입니다. 일단 접고 들어가는거죠.

CODESYS 가 가지고 있는 기능중에서 Visualization, 우리말로 하면 시각화라는 기능인데, 일종의 HMI 기능을 가지고 있는 것입니다. 대부분의 PLC 메이커들이 HMI 제품라인업을 가지고 있긴하지만, PLC 소프트웨어와 HMI 소프트웨어가 별개로 되어 있다보니, 서로 어드레스 맞추고 연결하는데 고충이 있죠.

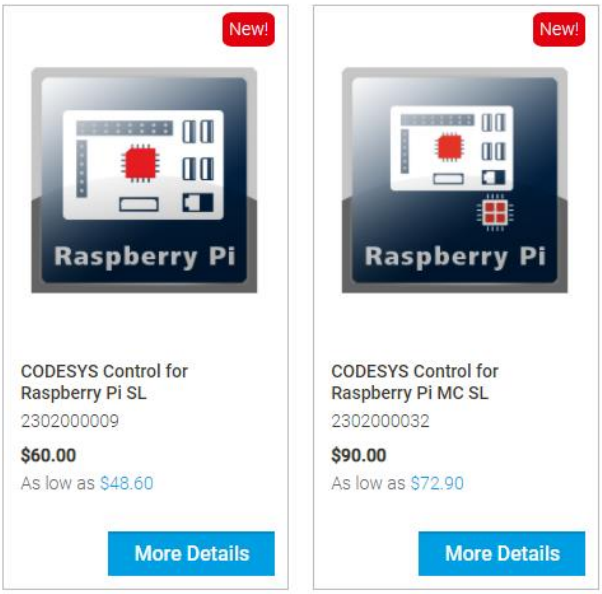
CODESYS 는 이 두개가 통합되어 있어, 매우 편리합니다.

WebVisu 라는 기능도 가히 획기적입니다. Visualization 으로 만든 HMI 화면을 웹을 통해서 원격으로 접속해서 제어할 수 있는 기능인데, 요즘 4 차산업혁명 얘기할때 자주 등장하는 ioT/스마트팜/스마트팩토리가 결국 원격지에서 스마트폰으로 공장/농장 제어한다는 건데, 이게 WebVisu 를 활용하면 큰 노력없이 바로 구현가능해지더군요.

CODESYS 에서 라즈베리파이를 지원하고 있기 때문에, 저희 컴파일파이에서도 CODESYS 를 쓸 수 있습니다.

최근 들어 미국과 일본에서 저희 컴파일파이를 사용하고 있는 고객분들이 CODESYS 관련사항을 가장 많이 물어봐 주시고 해서, 대응을 해드리다보니 저희도 자연스럽게 접하게 되었습니다. 우리나라에서는 많이 안 쓰시는거 같은데, 유럽과 북미/일본 등지에서는 꽤 많이 쓰이고 있다고 합니다.

CODESYS 는 사실 PC 용 소프트웨어라서 조금 비싼편인데, 이게 라즈베리용으로 나오면서 런타임 1 카피당 US$ 60 이라는 합리적인 가격으로 출시되어 있습니다. (2020 년 기준)



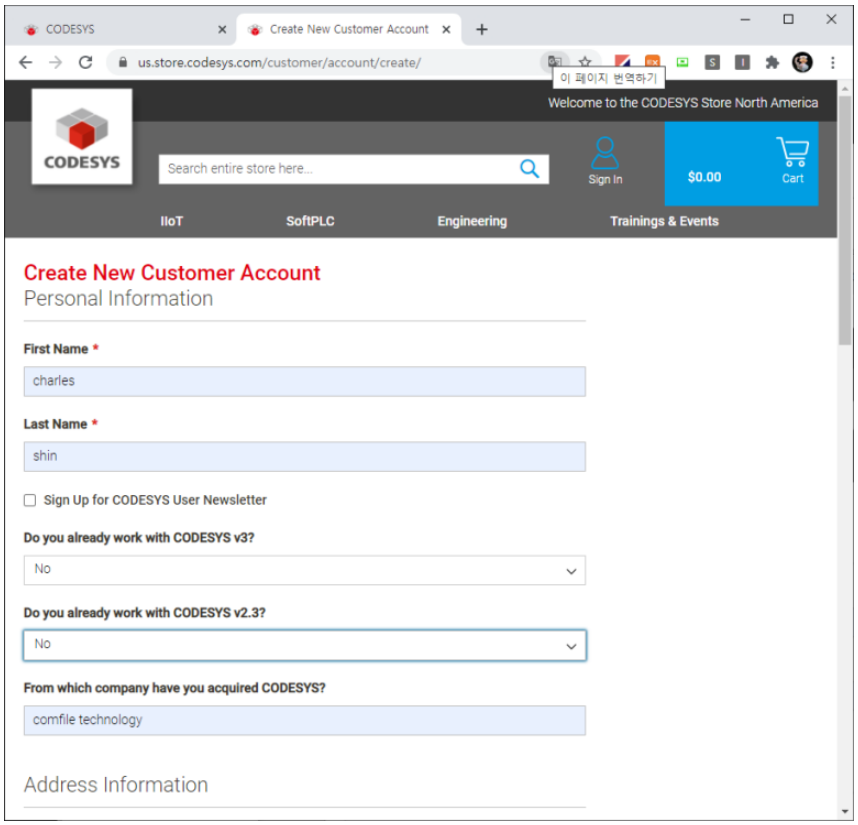
단돈 60 불로 신뢰성 있는 프로그램을 만들 수 있는 건데, C#엔지니어 한명에게 들어가는 인건비와 개발에 소요되는 시간, 그리고 시행착오에 소비되는 노력등을 고려했을때 결코 비싼 가격은 아니라고 생각합니다.

테스트하기 위해서 런타임 라이선스를 사실 필요는 없습니다. 테스트하면 어차피 계속 빌드/실행을 반복하기 때문입니다. 계속 돌렸을 경우 2 시간이후 동작이 멈춥니다. 나중에 양산하실 때 런타임 라이선스를 구입하시면 됩니다. 국내에서는 알트소프트에서 http://altsoft.kr/products/codesys/ CODESYS 판매와 기술지원을 하고 있습니다.

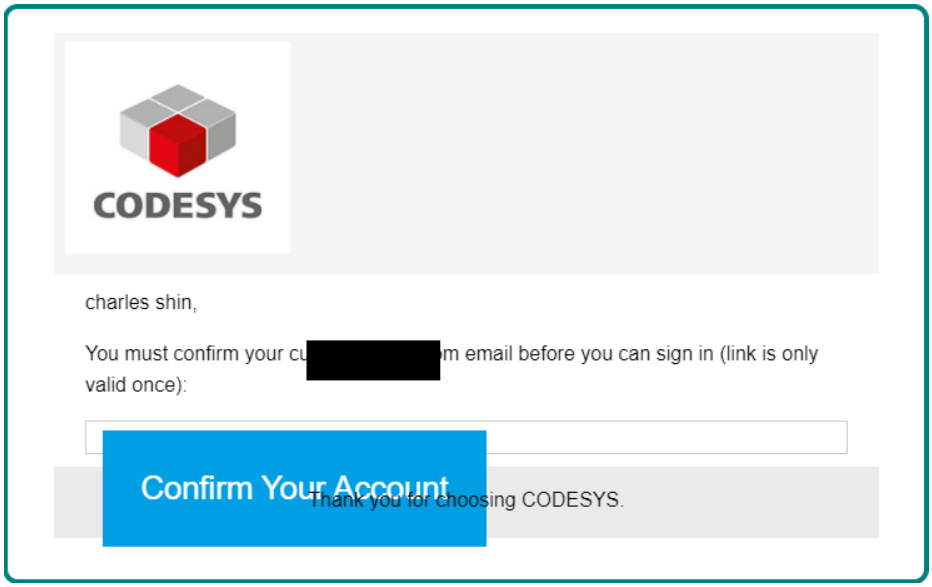
CODESYS 설치

https://us.store.codesys.com/

접속해서 CODESYS 를 다운로드 하세요. 먼저 회원가입을 해야합니다.



빈칸을 다 채우고 나면 등록했던 이메일로 컨펌메일이 하나 날라옵니다. 확인해주면 가입완료.

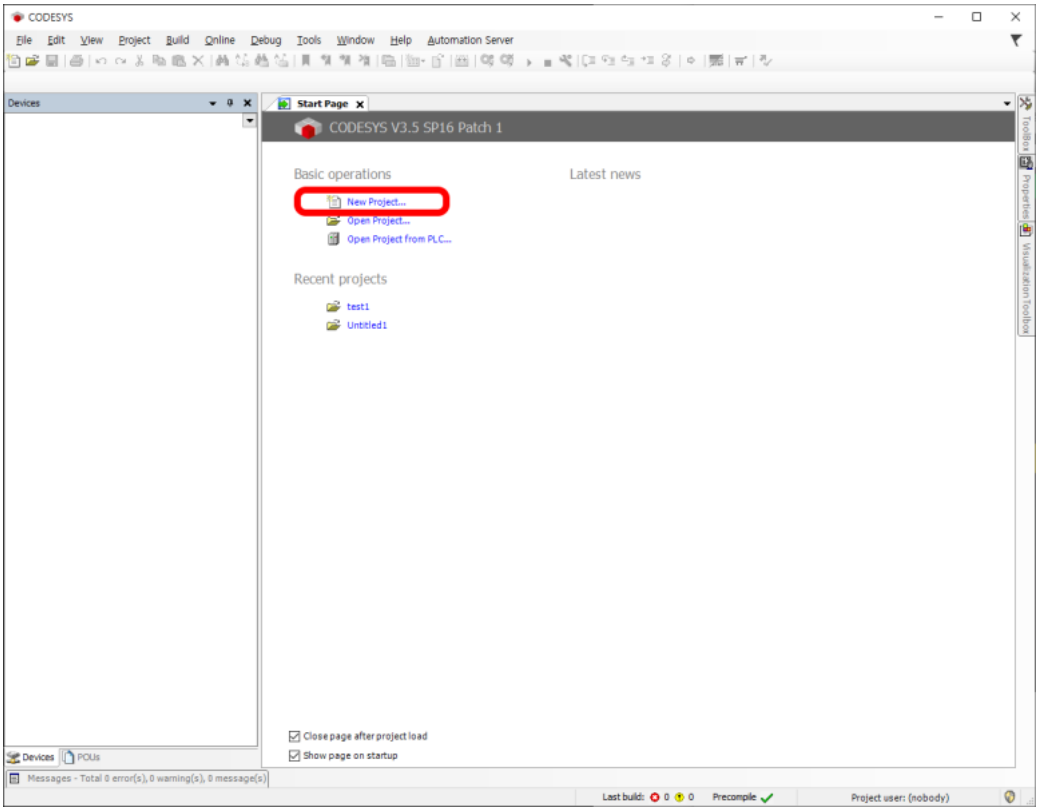


로그인하신뒤 이제 다운로드하면됩니다. 용량은 1 기가를 넘어가네요. 좀 기다려야합니다. 다운로드가 끝나면 설치하시면 됩니다. 설치시간이 꽤 걸립니다. 인내심 필요!!

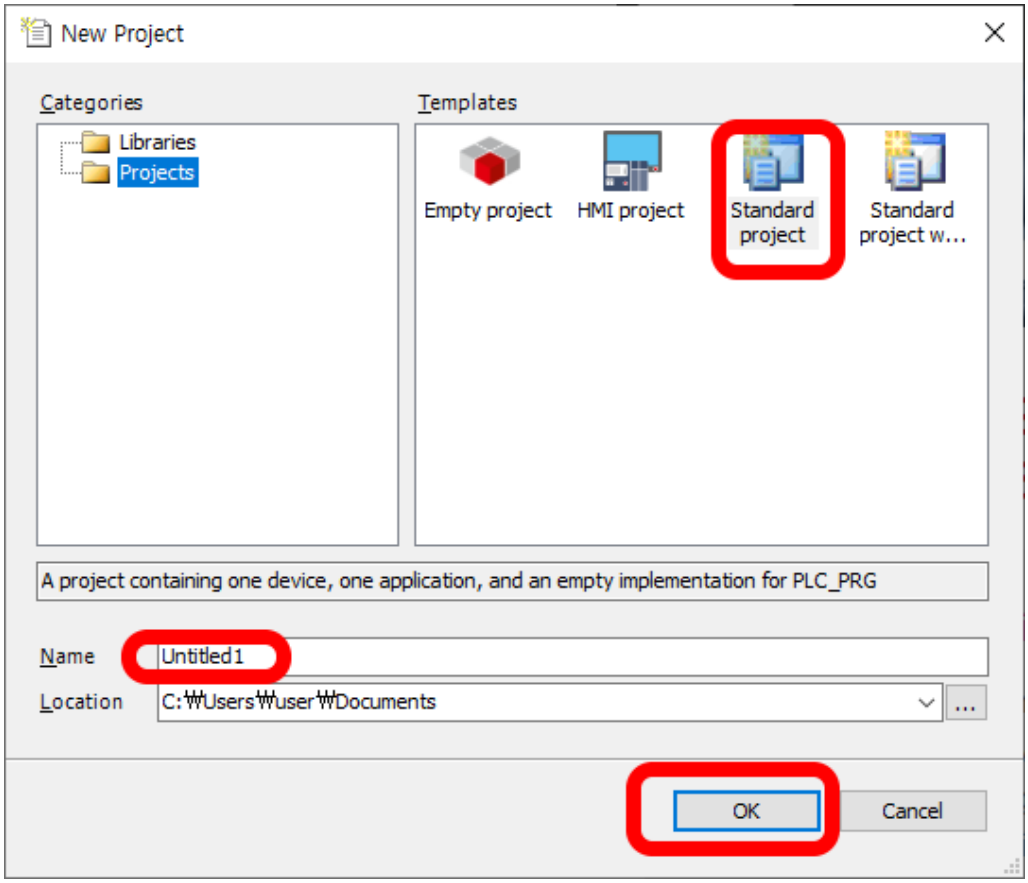
맛보기 프로젝트

컴파일파이와 연결없이 그냥 PC 에서 CODESYS 를 실행시켜서 동작까지 가보겠습니다. 아래 표시한 CODESYS V3.5 SPXX PATCHX 를 실행시키세요.

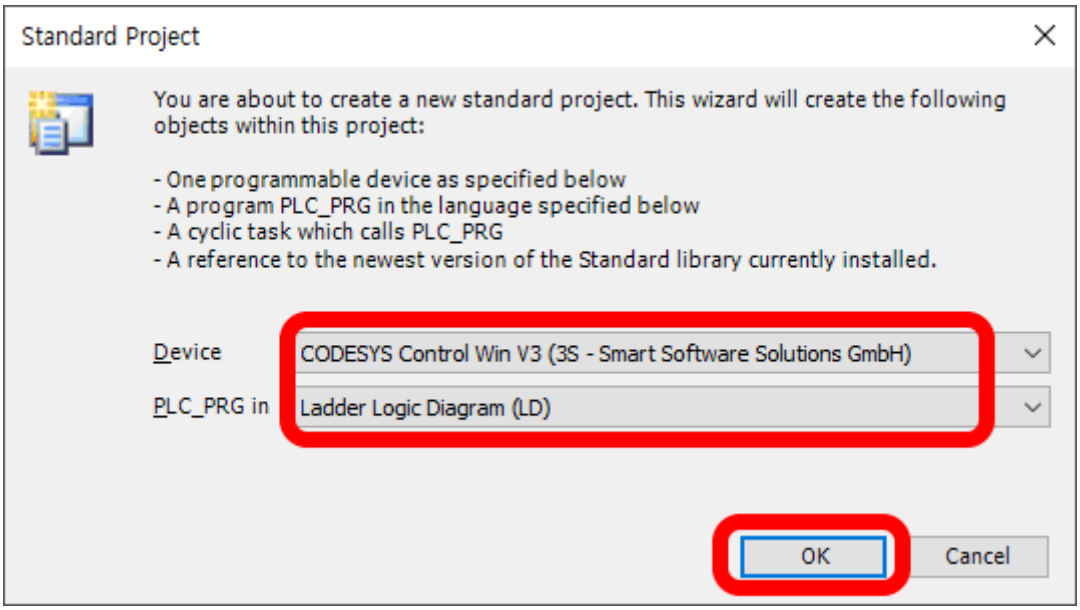
그러면 이런 화면이 뜨는데 여기서 New Project 를 누르세요.



그런 다음 아래화면이 뜨면 Standard Project 로 선택하고 적당한 이름으로 프로젝트명을 입력하고 ok 를 누르세요.

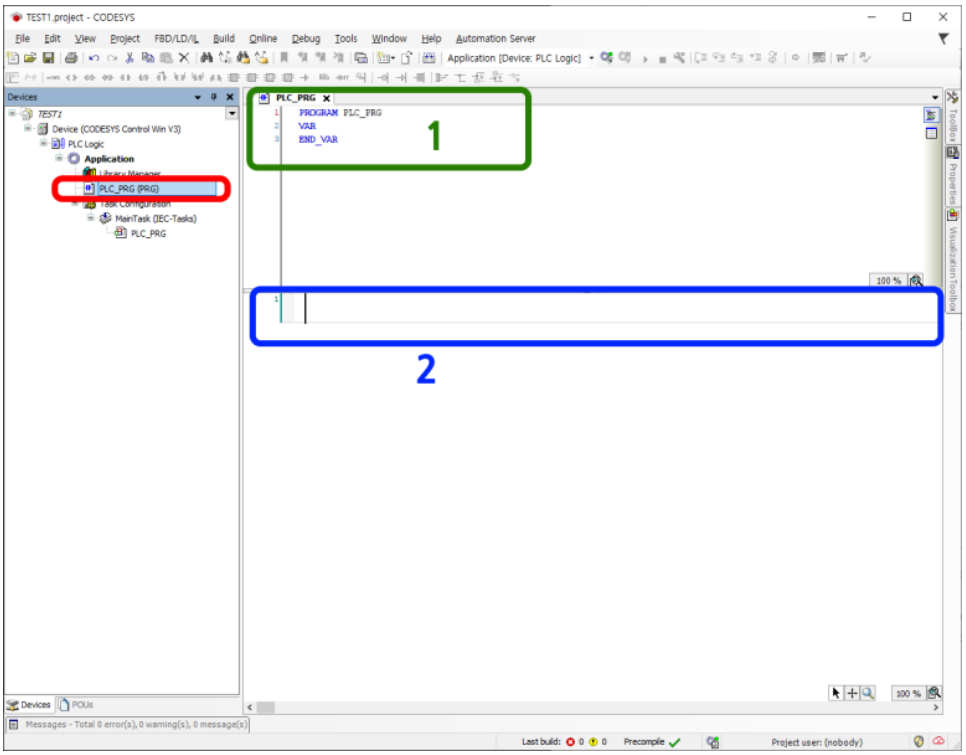


다음 선택할 내용은 이겁니다.

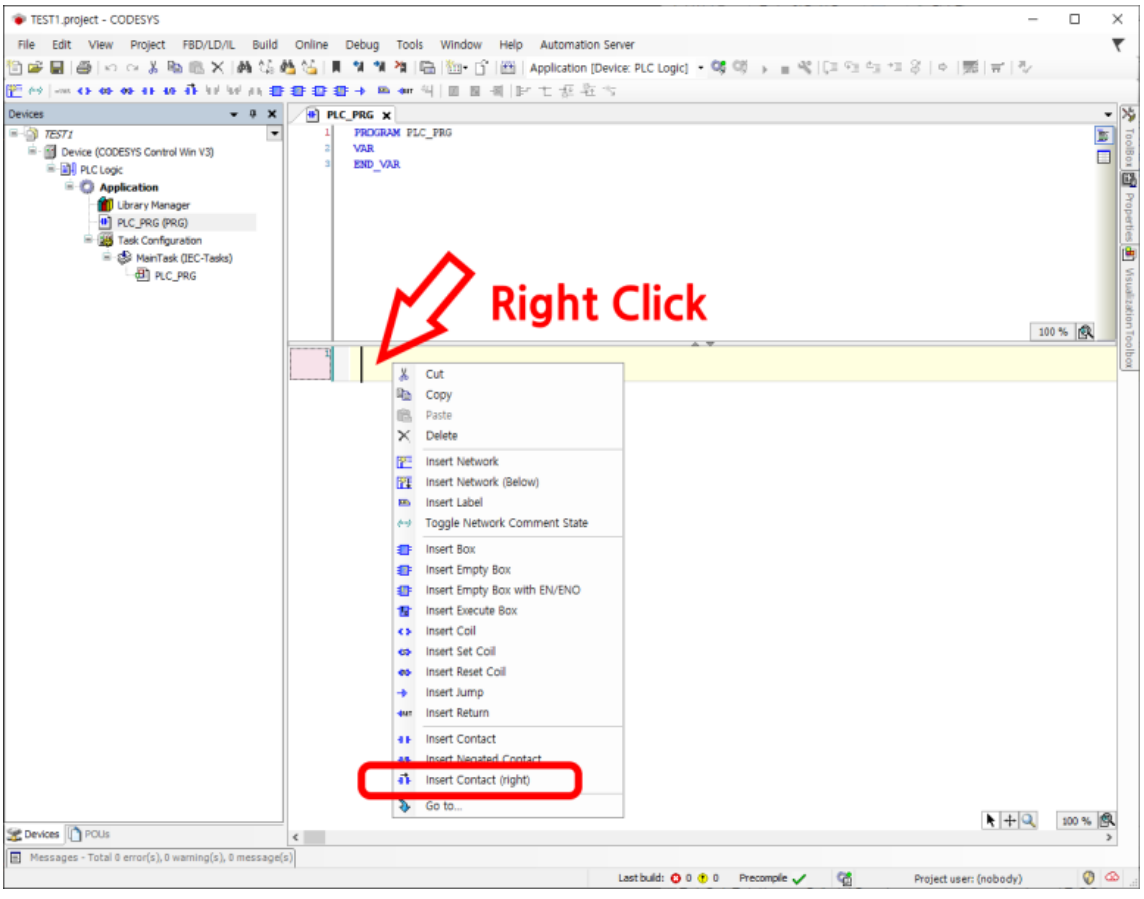


PC 용 CODESYS 이름이 Control Win 입니다. 이걸 고르시고, 밑에는 레더로직 다이어그램으로 하세요.

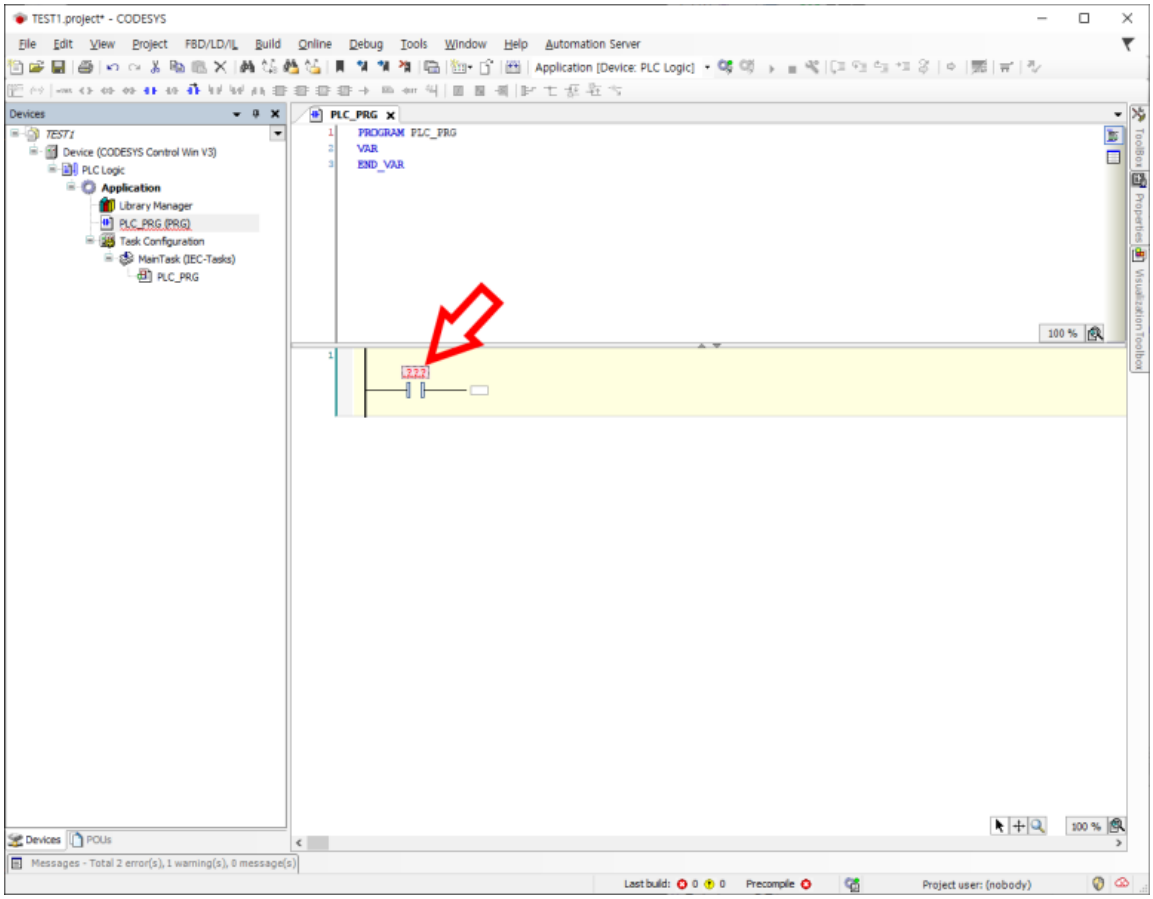
다음 화면은 이건데, 빨간 박스친 부분을 더블클릭하면 1 번과 2 번 부분이 보입니다. 1 번은 변수선언이고, 2 번은 레더로직입니다.



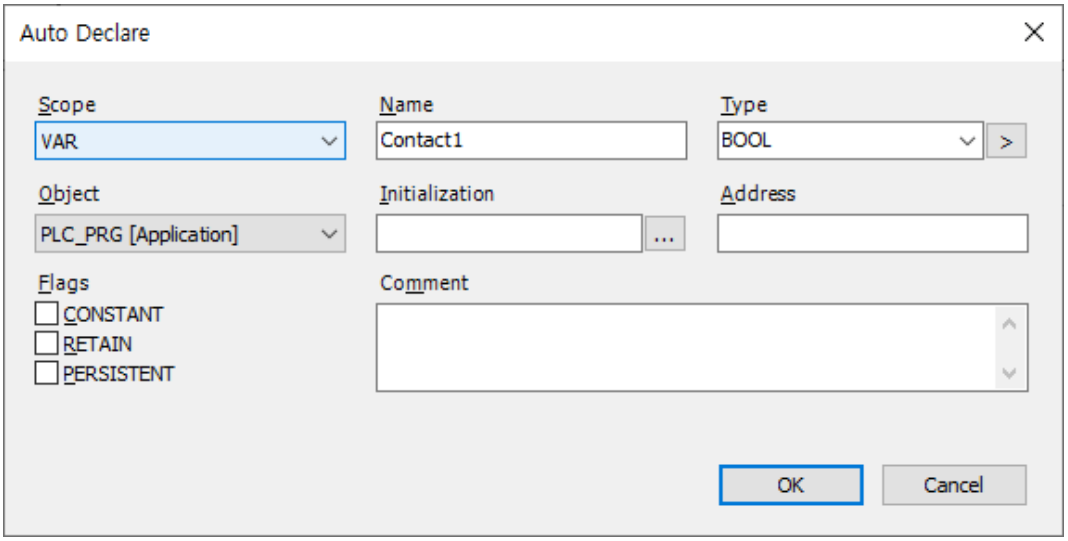
간단한 레더로직을 하나 그려보겠습니다. 화살표 위치에서 마우스 우클릭을 하고 Insert Contact 를 선택하세요.



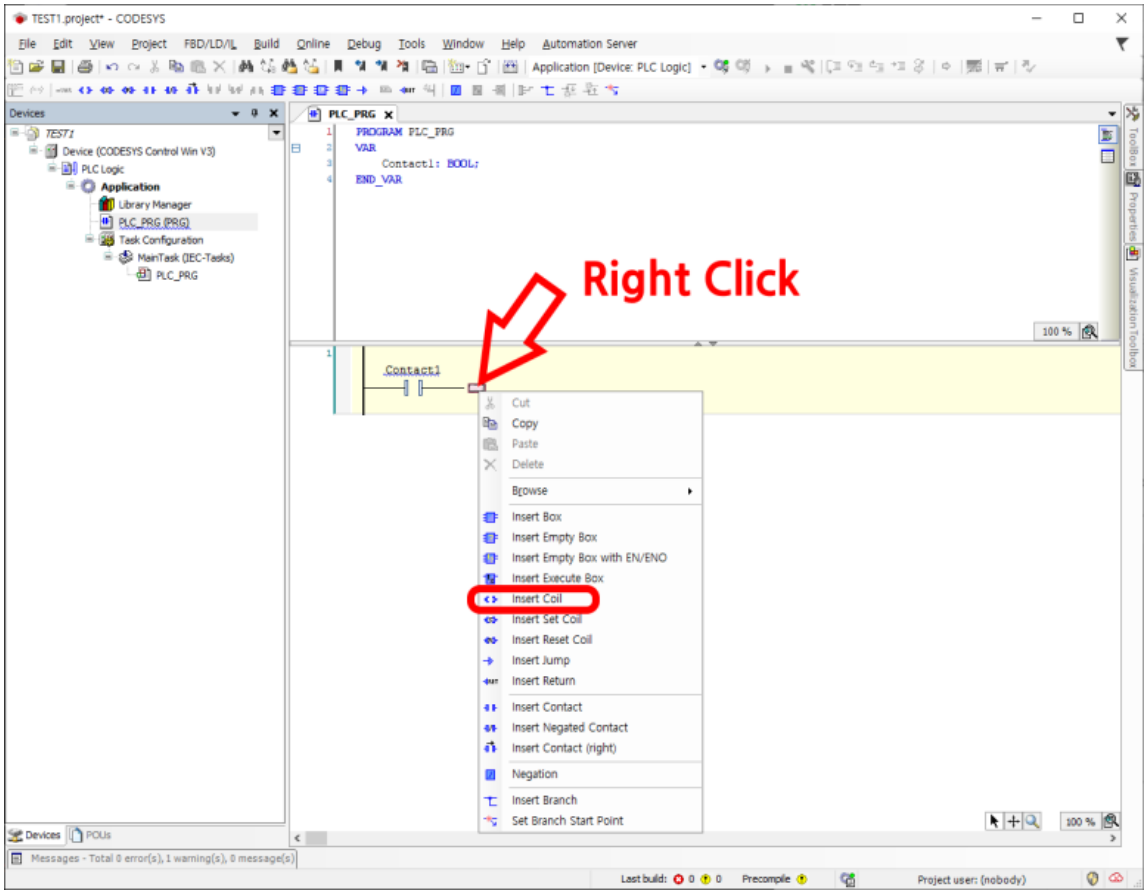
심볼이 하나 표시되는데, 그 위에 물음표 ??? 이 부분을 바꿀예정입니다. 여기에 Contact1 이라고 입력하세요.



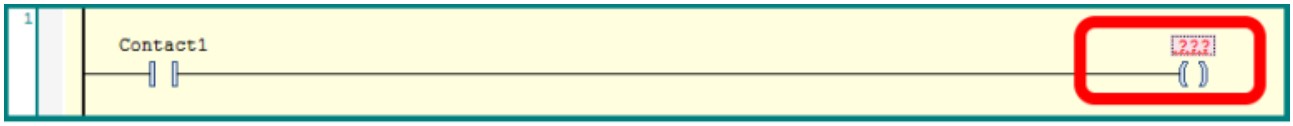
이렇게 입력하면 아래와 같은 박스가 뜹니다. 그냥 OK 누르세요.



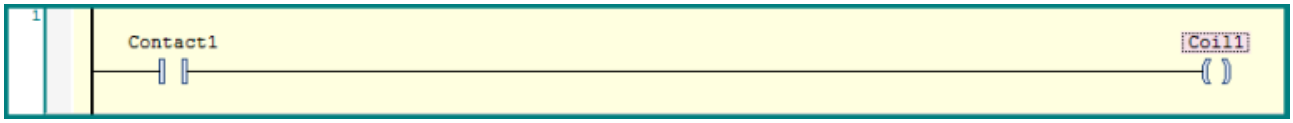
이번엔 정확히 아래 화살표 지점에 조그만 박스를 우클릭하고 insert Coil 을 선택하세요.



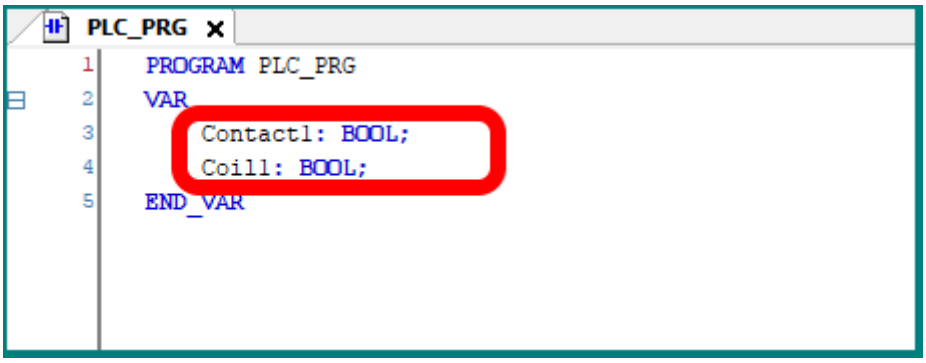
출력 코일이 하나 생겼습니다.



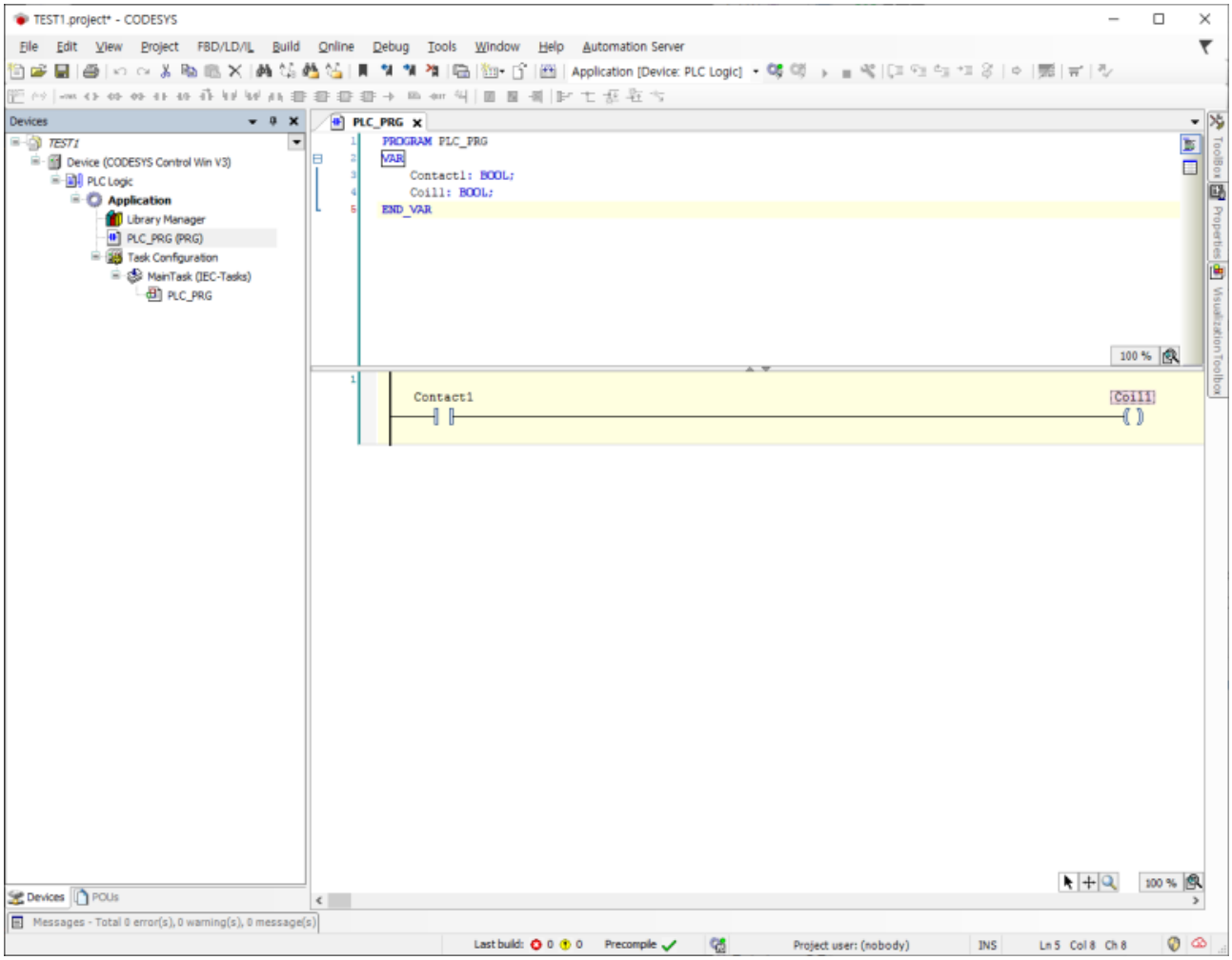
여기 이름을 Coil1 로 바꿔주세요.



이때 위를 잠깐 보시면 이렇게 바뀌어 있을 겁니다. 알아서 변수 선언이 된 것입니다.



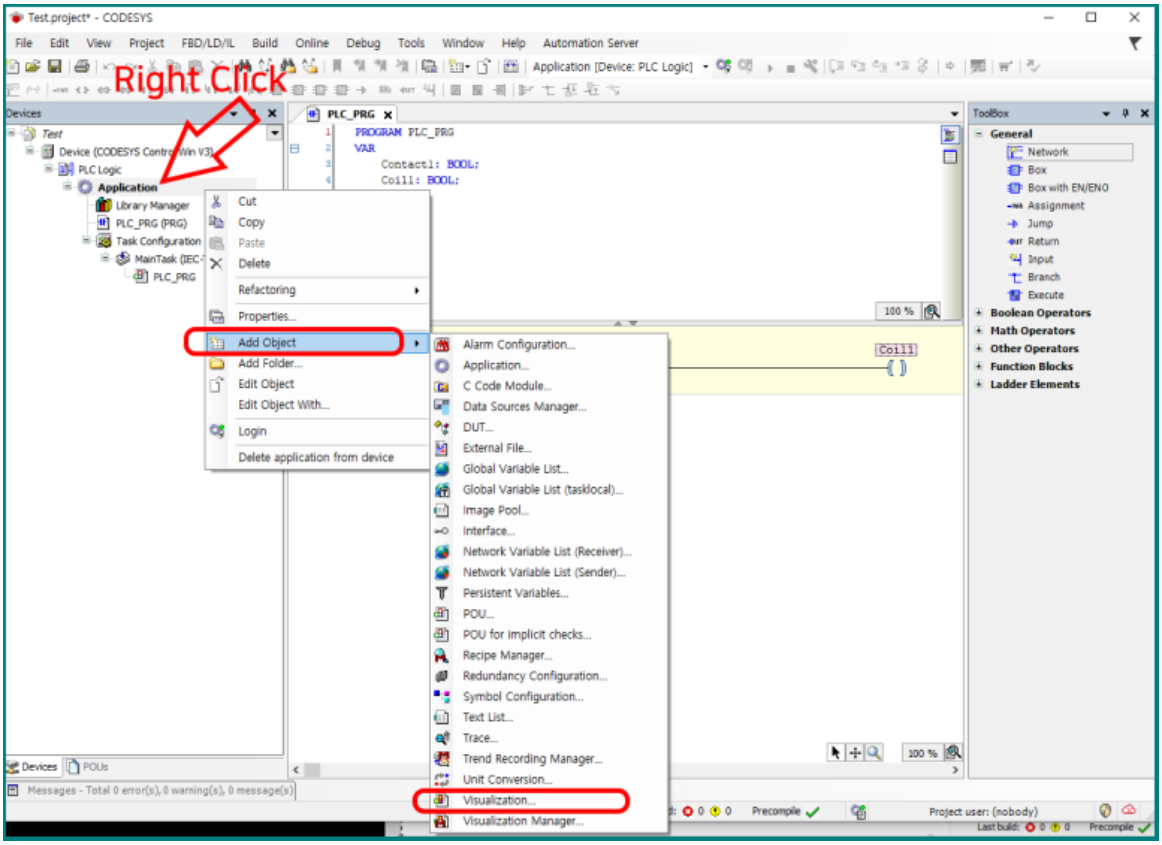
쟈쟌, 첫번째 레더로직 다이어그램이 완성되었습니다.



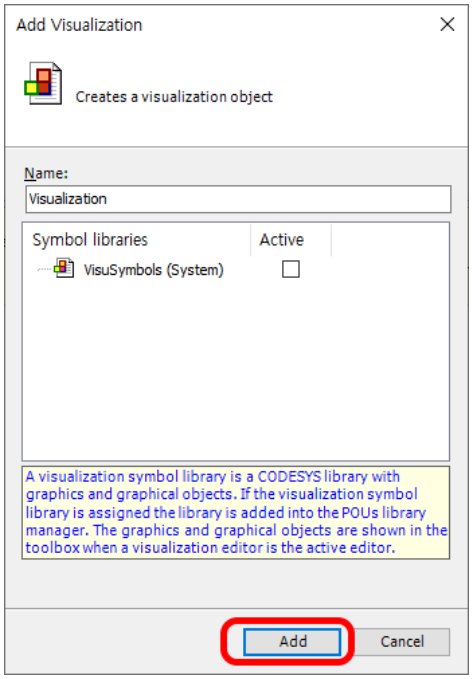
Contact1 을 누르면 Coil1 이 켜지는 단순한 레더입니다.

시각화 (Visualization)

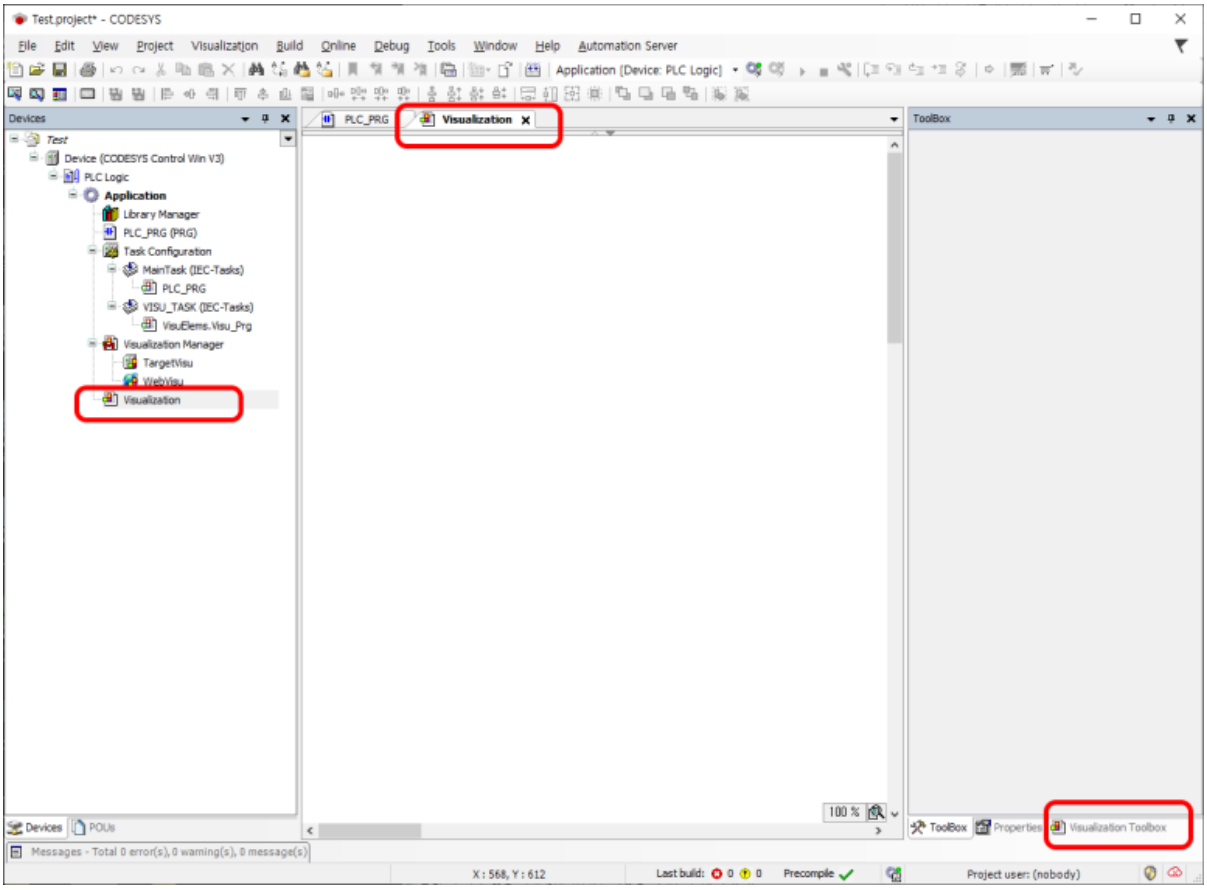
CODESYS 에는 HMI 에 해당하는 Visualization(시각화 : 비주얼라이제이션) 이라는 기능이 포함되어 있습니다. 앞에서 만든 레더로직을 시각화 해보도록 하겠습니다.



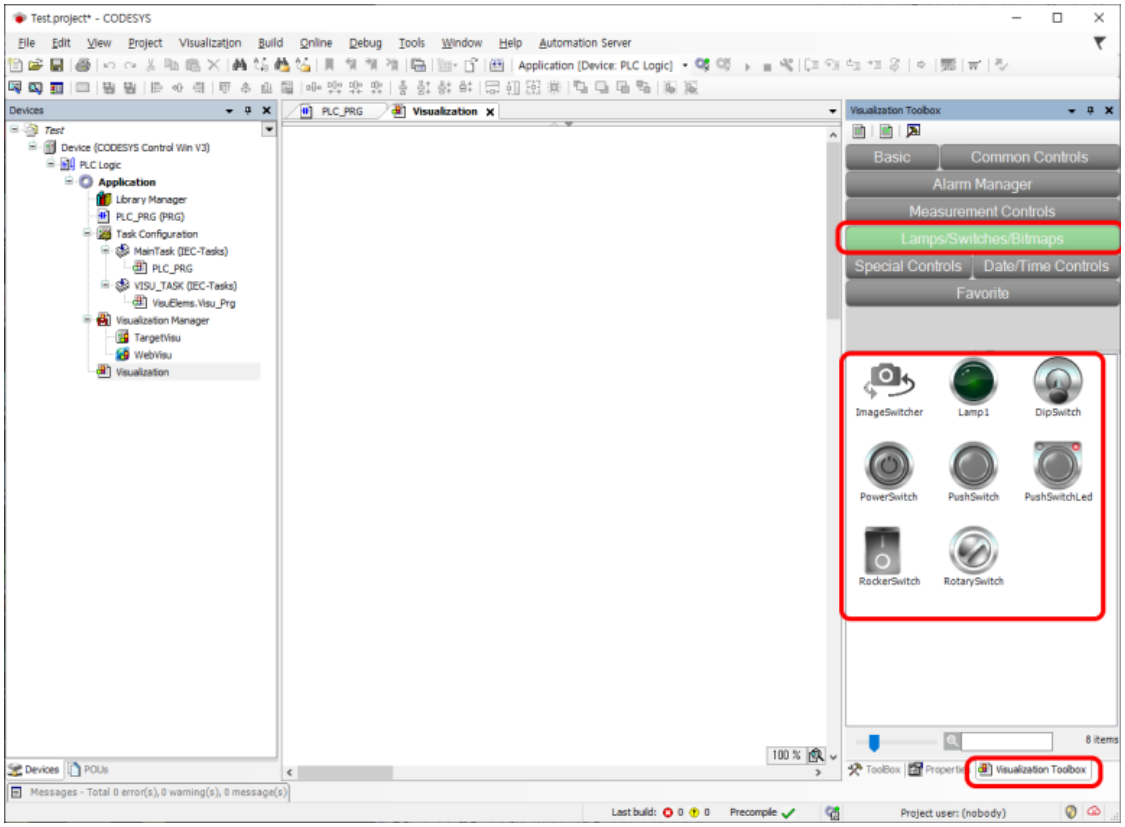
왼쪽 창에서 Application 을 우클릭한 뒤 Add Object 에서 Visualization 을 선택하세요.



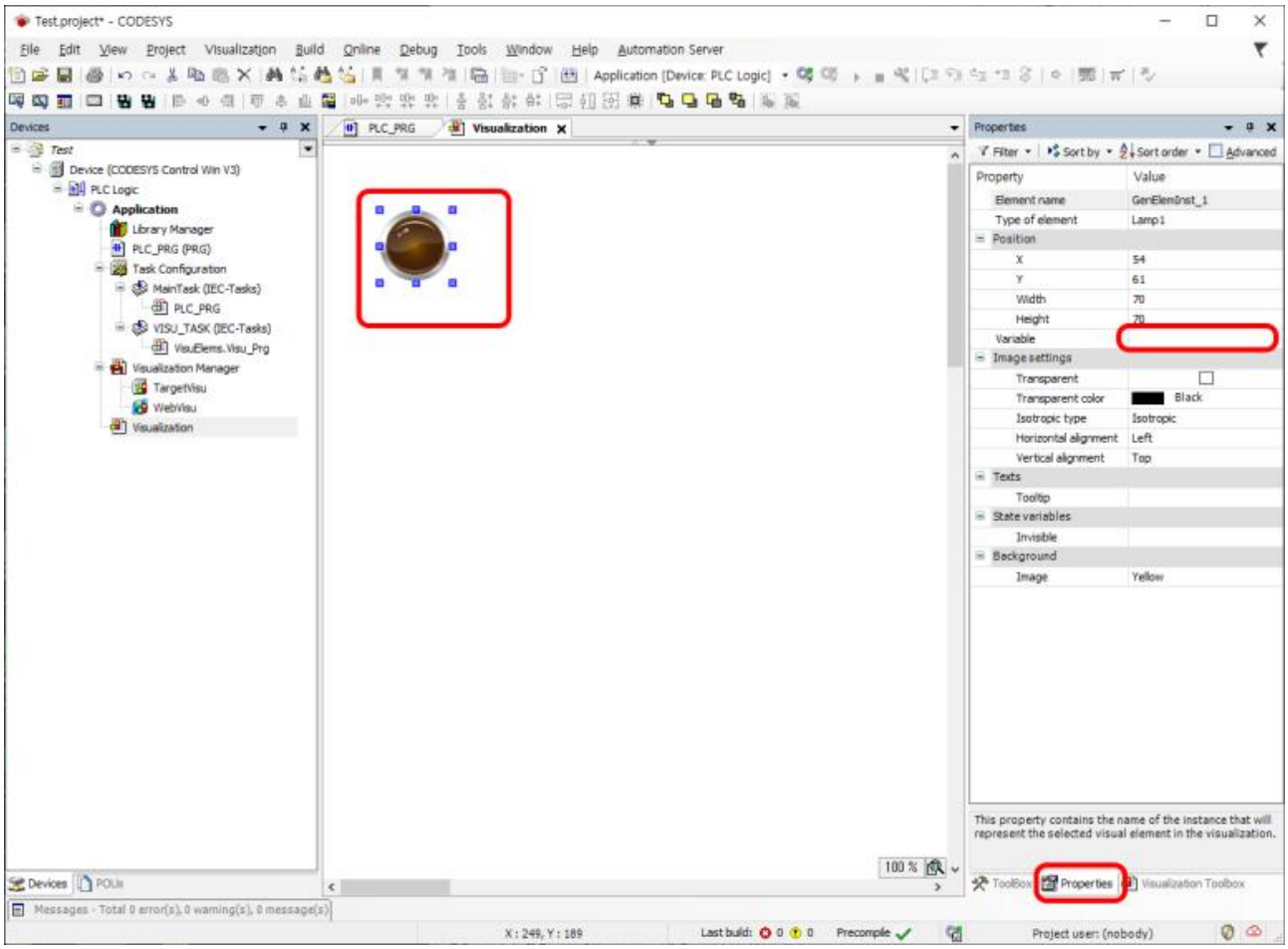
이 창이 나오면 Add 누르세요. 그러면 (한참 뒤) 아래와 같은 창이 나옵니다.



오른쪽 아래 있는 Visualization Toolbox 를 눌러보세요.

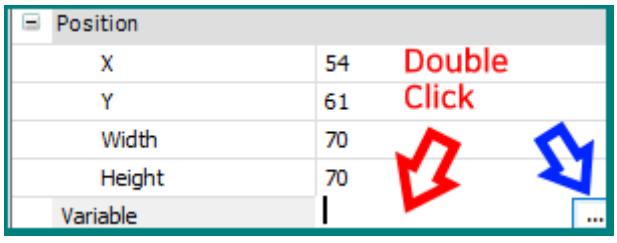


그리고 Lamp/Switches/Bitmaps 탭을 클릭하면, 그 아래 각종 램프와 스위치가 표시됩니다. 여기서 램프(Lamp)하나를 끄집어내서 화면 중앙으로 가지고 오세요.

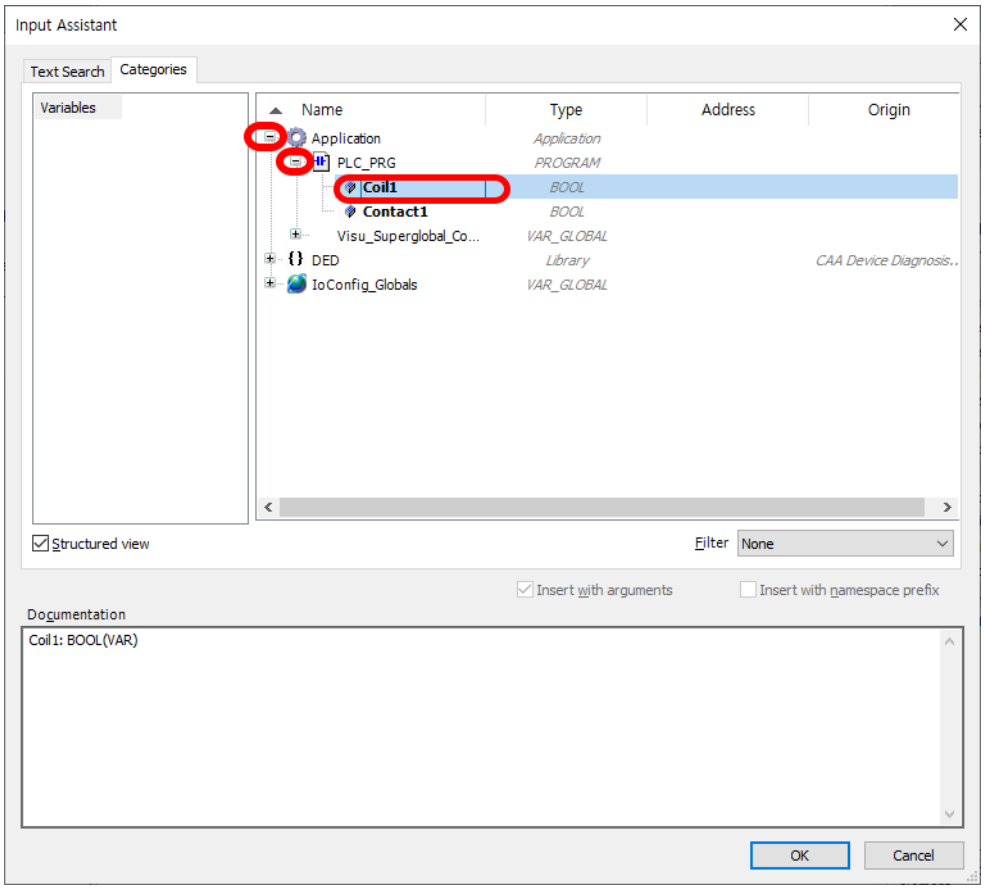


그러면, 화면 오른쪽이 Properties 창으로 바뀝니다. 여기에 보면 Variable 이라는 항목이 있는데, 여기에다가 Coil1을 입력하세요. 전 강의에서 레더로직에 사용되었던 이름(Name)입니다.

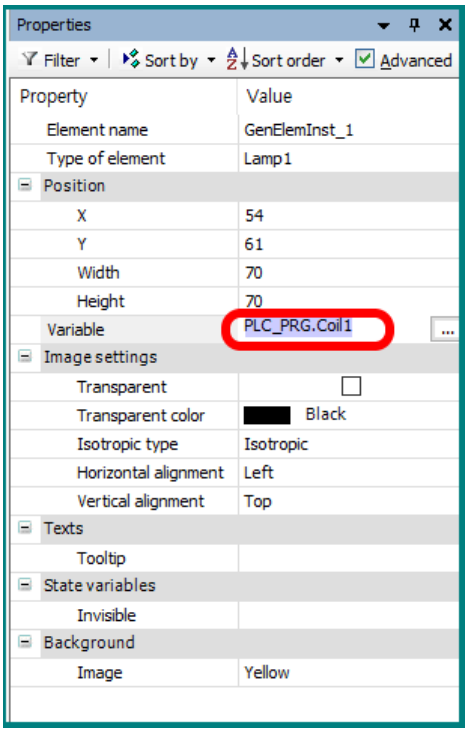
그냥 입력하지 말고, 이렇게 하세요. 빈칸을 더블클릭하면 오른쪽에 쩜쩜쩜[...]이 표시됩니다. 이걸 클릭하세요.



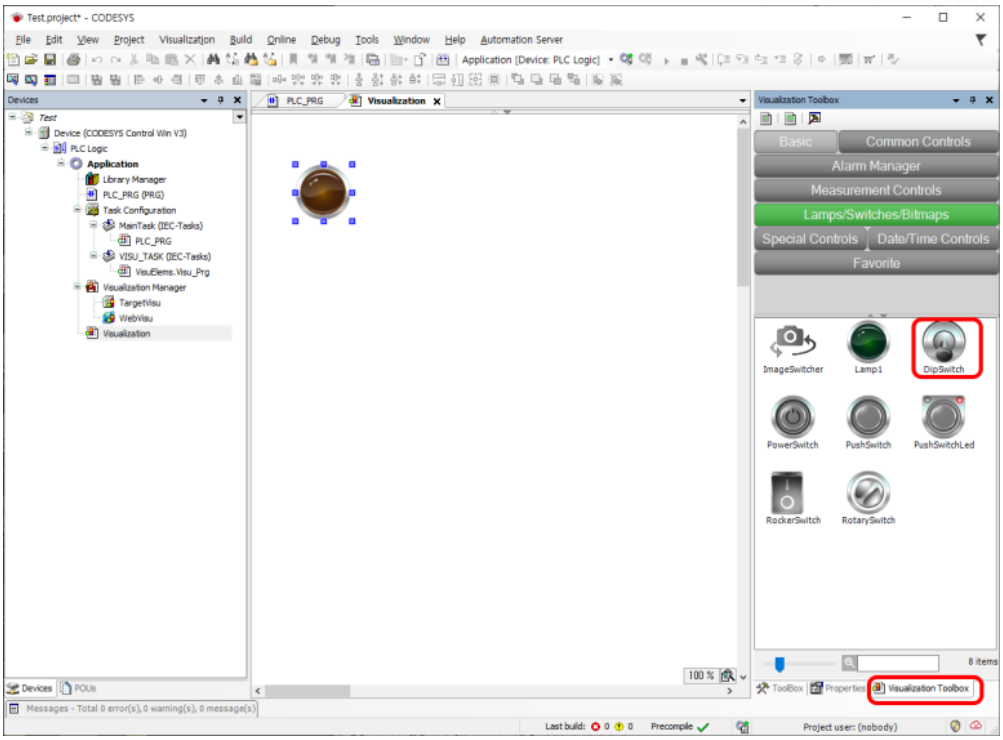
그럼 이런게 표시됩니다. 당황하지 마시고, Application 옆에 + 표시를 클릭하면 펼쳐집니다. 계속 펼쳐서 아래같은 모양을 만들고 여기서 Coil1 을 선택하세요.



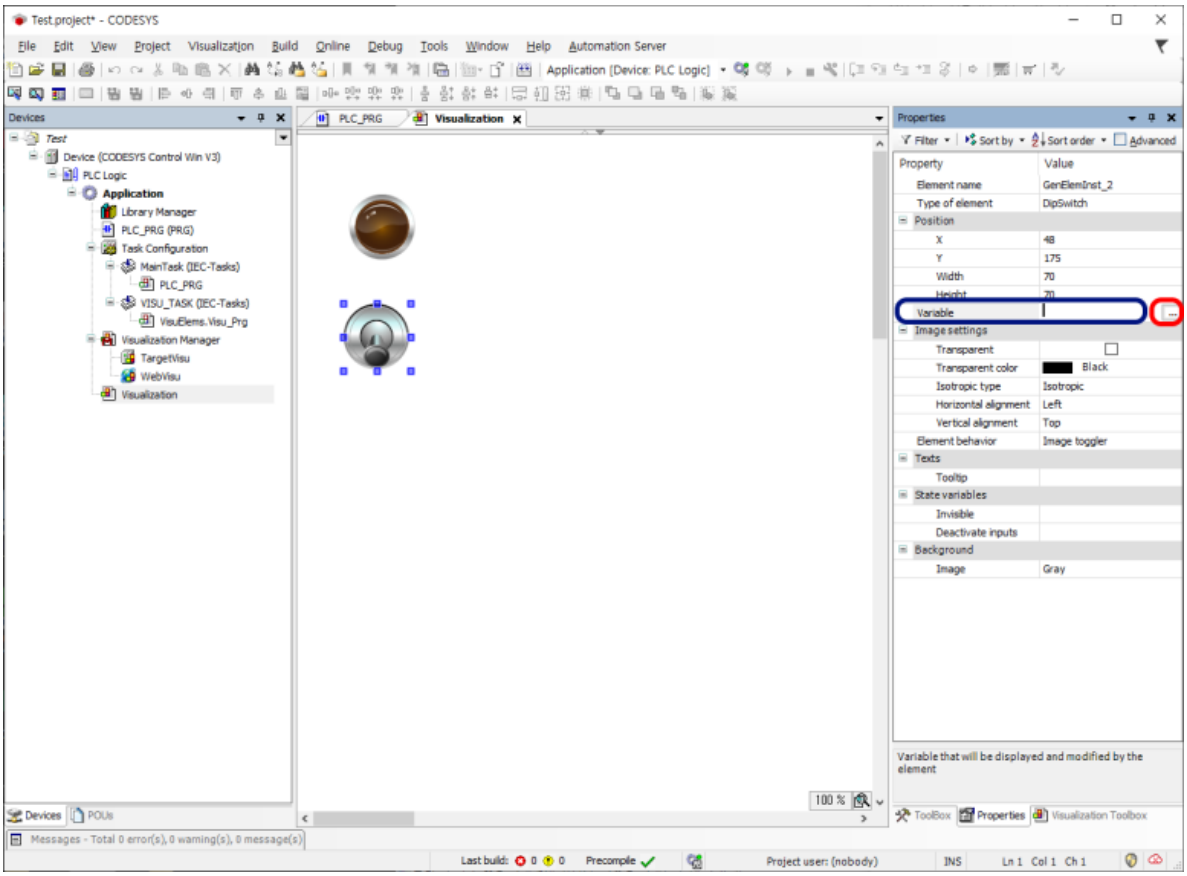
그러면 이렇게 입력이 자동으로 됩니다.



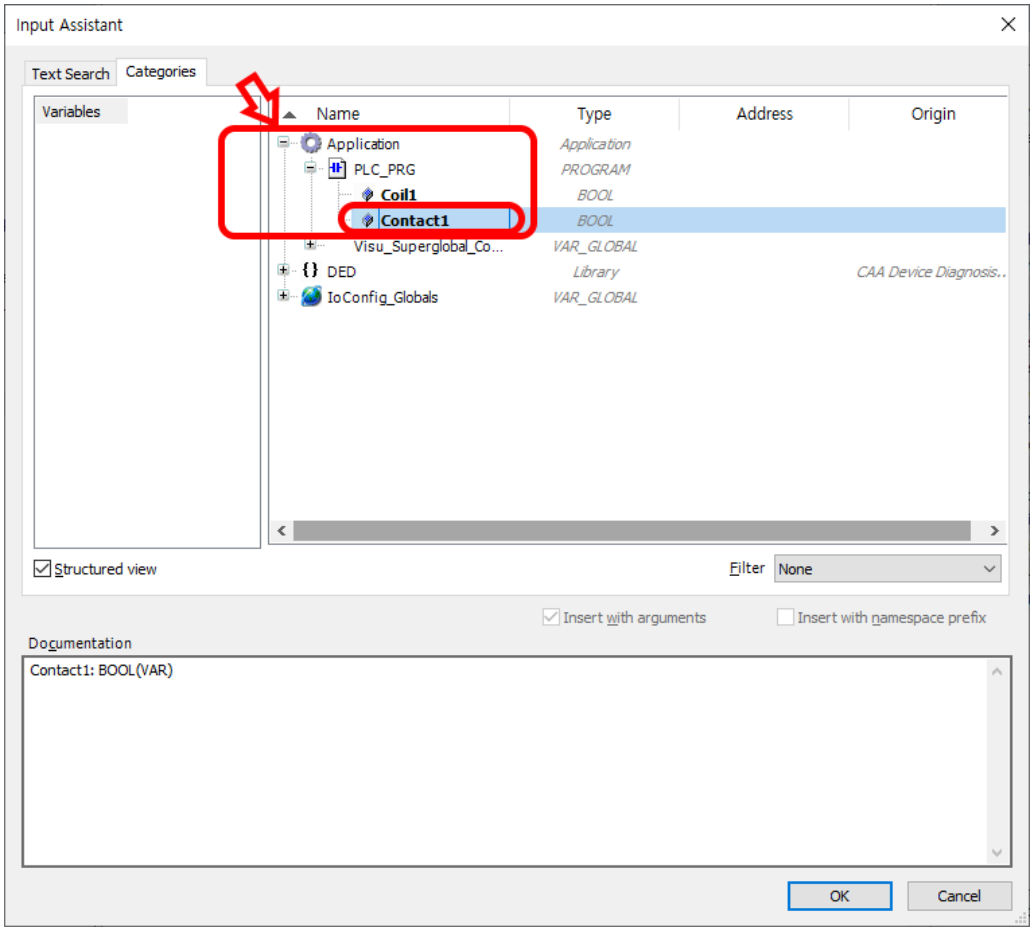
다시 화면 오른쪽 하단에 Visualization Toolbox 탭을 누른뒤, 이번엔 DipSwitch 를 화면 중앙으로 끄집어 내어 보세요.



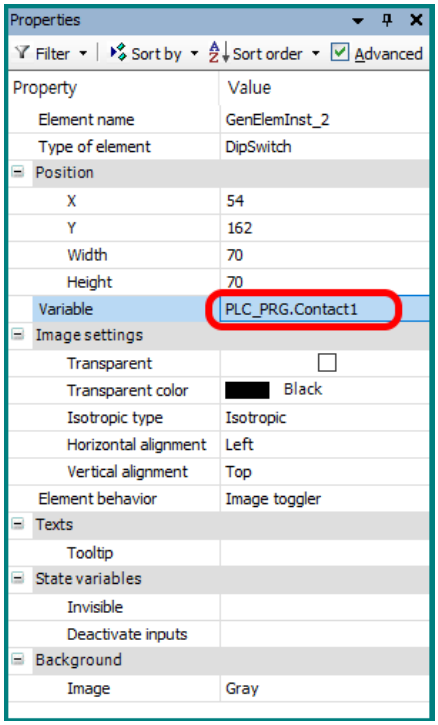
마찬가지로 Variable 이름을 정해주어야 합니다. 아까처럼 빈칸을 더블클릭해서 쩜쩜쩜을 나타나게 한뒤 클릭하세요.



이번엔 Contact1 을 선택해주면 됩니다.

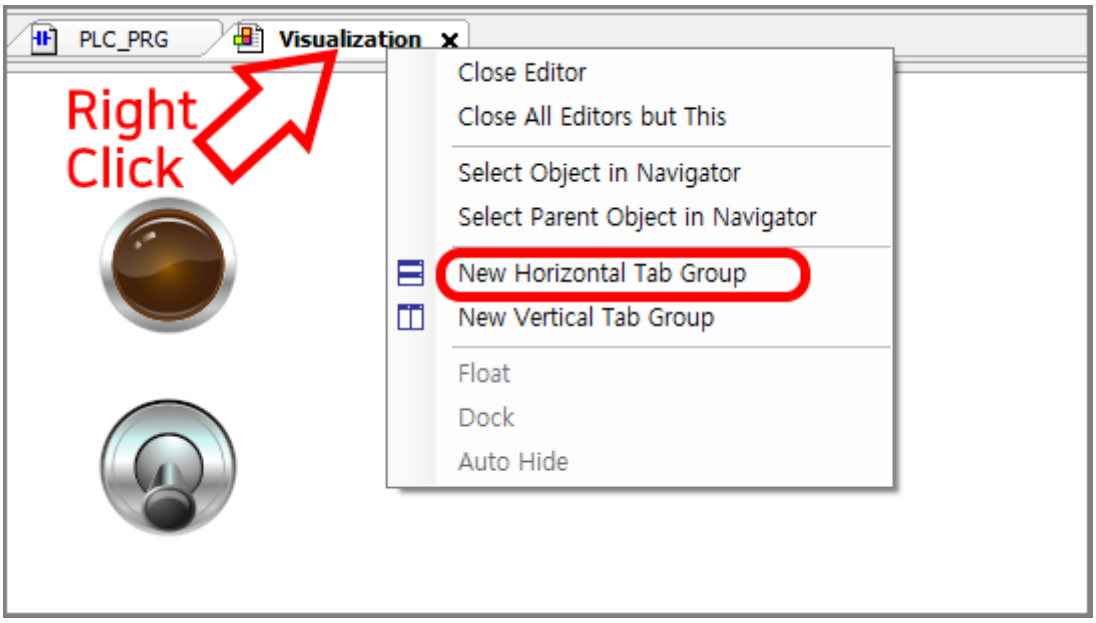


잘 되었군요.

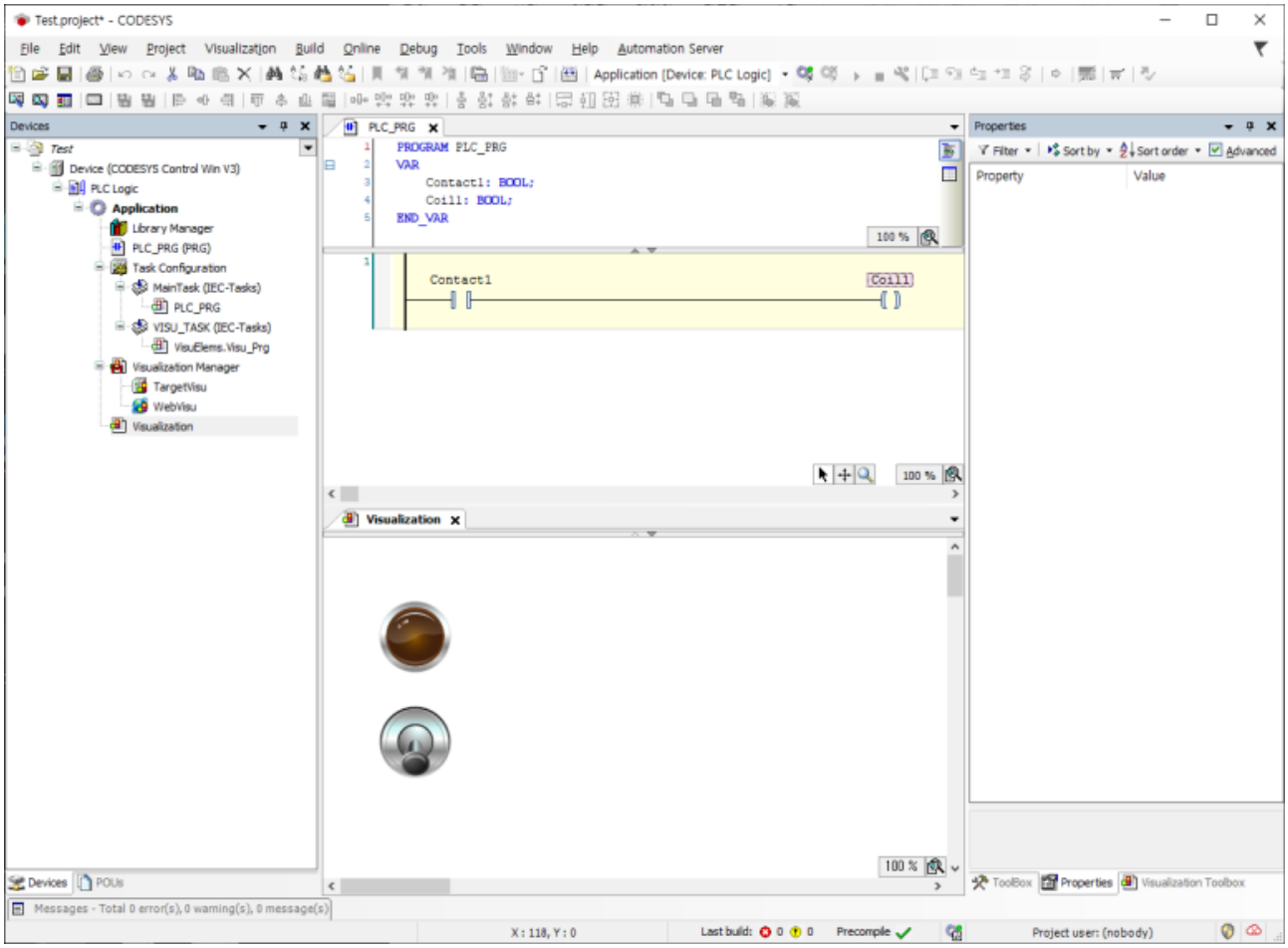


작성은 끝났습니다. 실행시켜보는 일만 남았습니다.

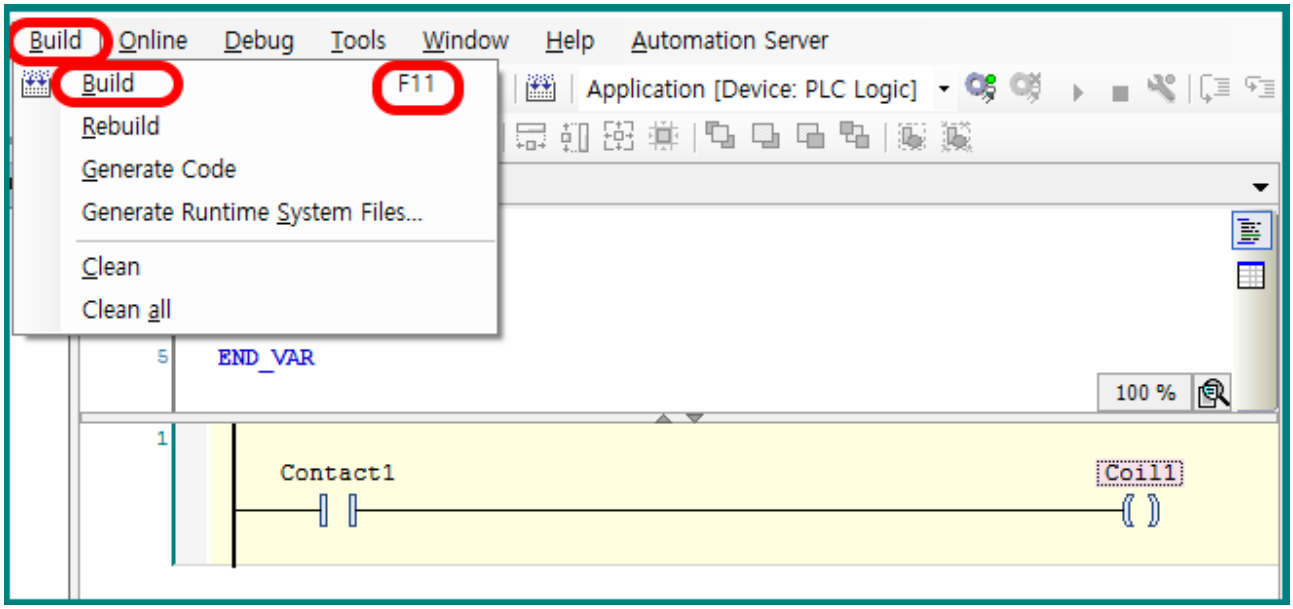
그전에 화면을 좀 분할해서 보기 편하도록 바꿔보겠습니다. Visualization 탭을 우클릭한 뒤 New Horizontal Tab 을 선택하세요.



그러면 아래처럼 바뀝니다. 레더로직과 시각화(Visualization)이 둘다 보이는군요.

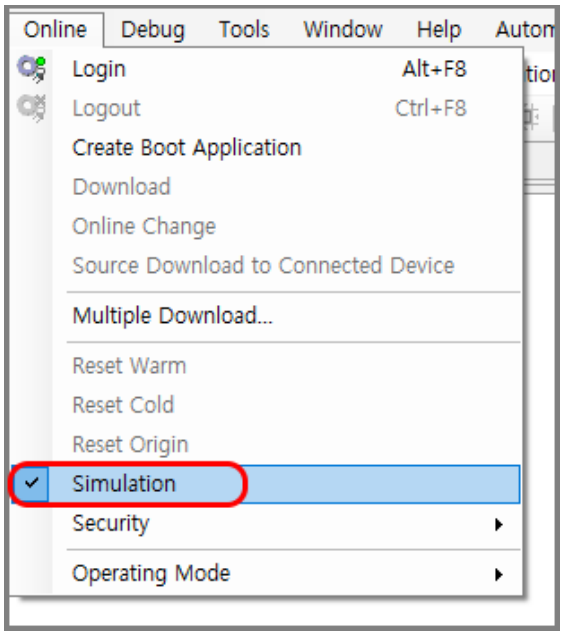


제일 먼저 Build 를 해주어야 합니다. 상단 Build 메뉴에서 Build 를 선택해도 되고, 그냥 F11 을 눌러도 됩니다.

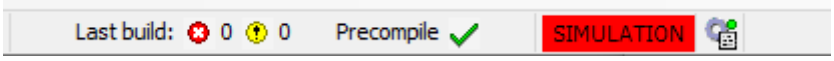


화면 하단을 보면, 빌드 하는 동안 어떤 메시지가 표시됩니다. 문제가 있다면 이곳에 내용을 표시합니다. 저는 문제없이 통과되었습니다.

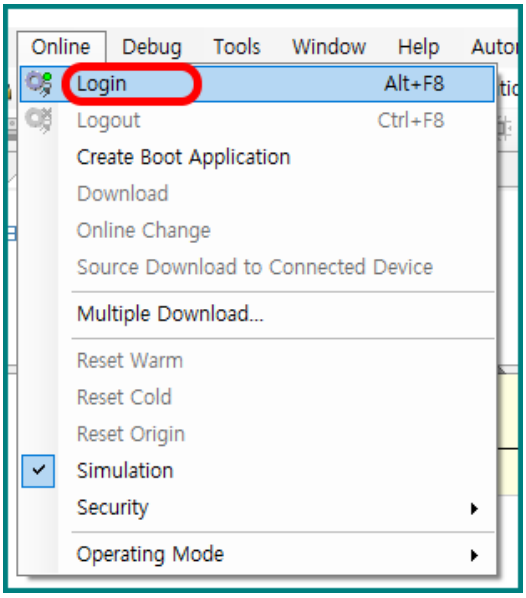
타겟 PC 없이 지금 개발중인 PC 에서 시뮬레이션으로 동작결과를 보겠습니다. 상단 Online 메뉴에서 Simulation 을 클릭해서 활성화시켜 놓으세요.



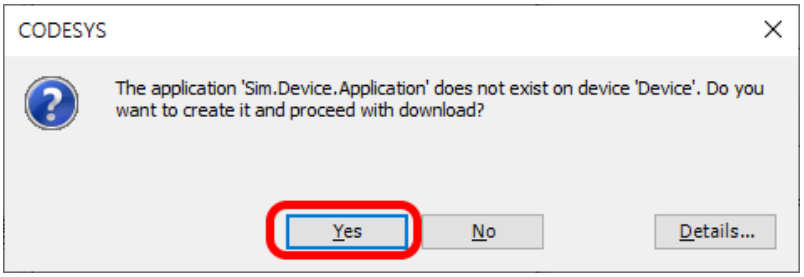
화면 하단에 보면 빨간색으로 SIMULATION 중이라고 표시됩니다.



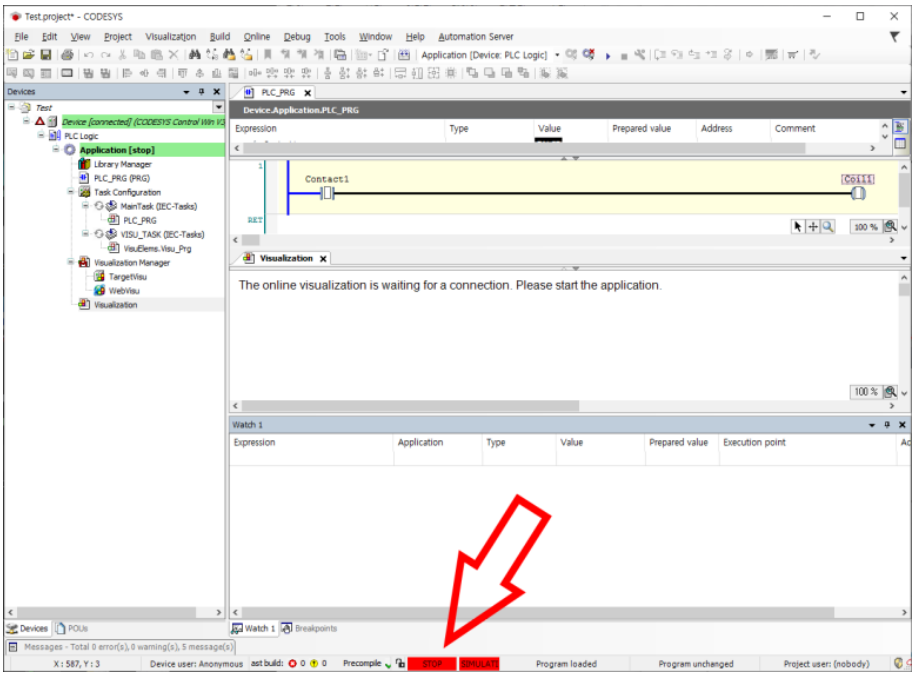
그 다음 다시 Online 메뉴에서 Login 을 선택하세요.



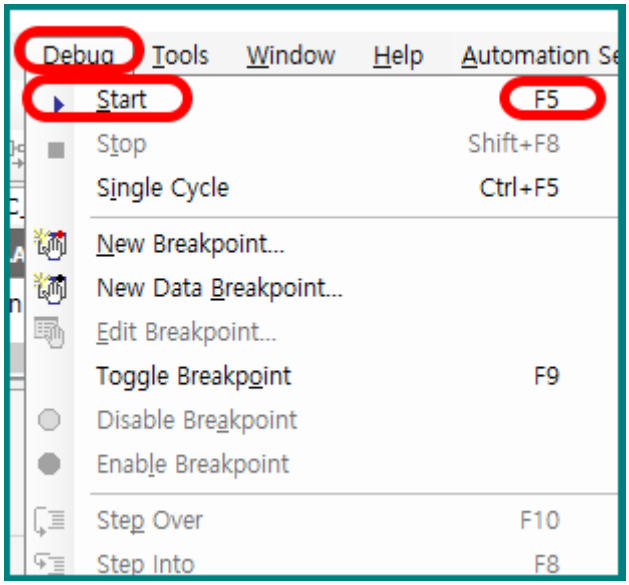
그러면 아래박스가 나오는데, 디바이스를 생성하겠는지 물어보는 것입니다. 당연히 Yes



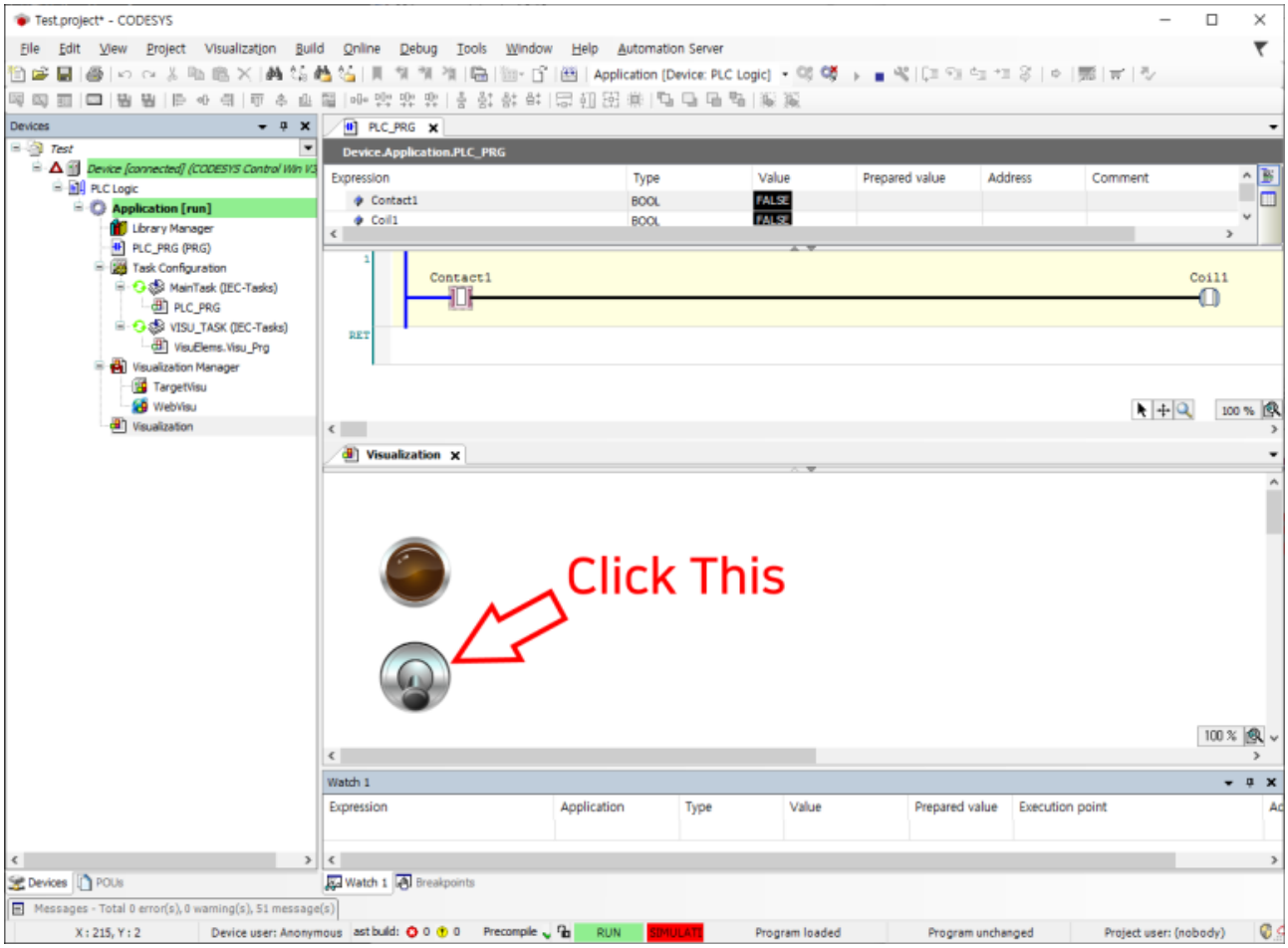
잠시 후, 화면은 아래와 같은 상태가 되는데, 하단 쪽을 자세히 보면 현재 STOP 상태임을 알 수 있습니다.



상단 Debug 메뉴에서 Start 를 선택하거나, 그냥 F5 를 누르면 시작됩니다.



아래 딥스위치를 클릭해보세요. 레더로직과 연동해서 동작이 되는 것을 확인하실 수 있습니다.



제 2 장. 컴파일파이 준비

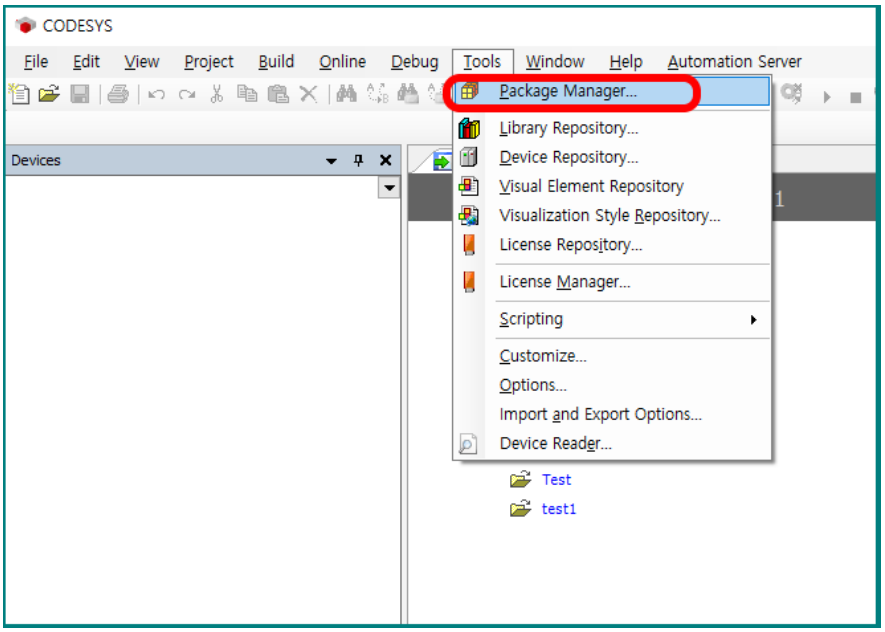
CODESYS는 원래 PC용 개발툴이라서 PC에서는 바로 실행이 가능한데, 이것을 라즈베리 파이 (컴파일 파이)에서 실행하도록 하기 위한 조정작업이 좀 있습니다. 나름 복잡한데, 이 과정만 잘 극복(?)하시면 아주 편안하고 쾌적한 개발환경이 그 보상으로 주어집니다. 조금만 참고 이 책에서 이끄는대로 따라와 보세요.

라즈베리 패키지 설치

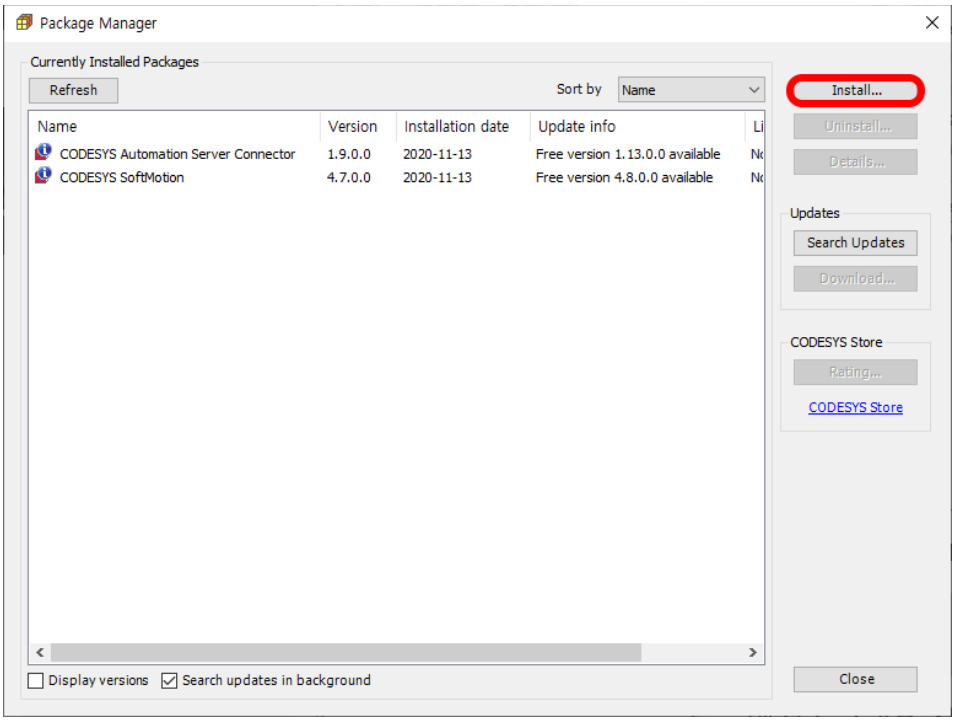
우선 아래 링크에서 "CODESYS 라즈베리파이 패키지" 파일을 다운로드하세요. (아래 링크가 안되면 store.Codesys.com 에서 raspberry 를 검색한 뒤 다운로드할 수 있습니다.)

https://store.codesys.com/ftp_download/3S/RaspberryPI_SL/603001/3.5.16.20/CODESYS%20Control%20for%20Raspberry%20PI%203.5.16.20.package

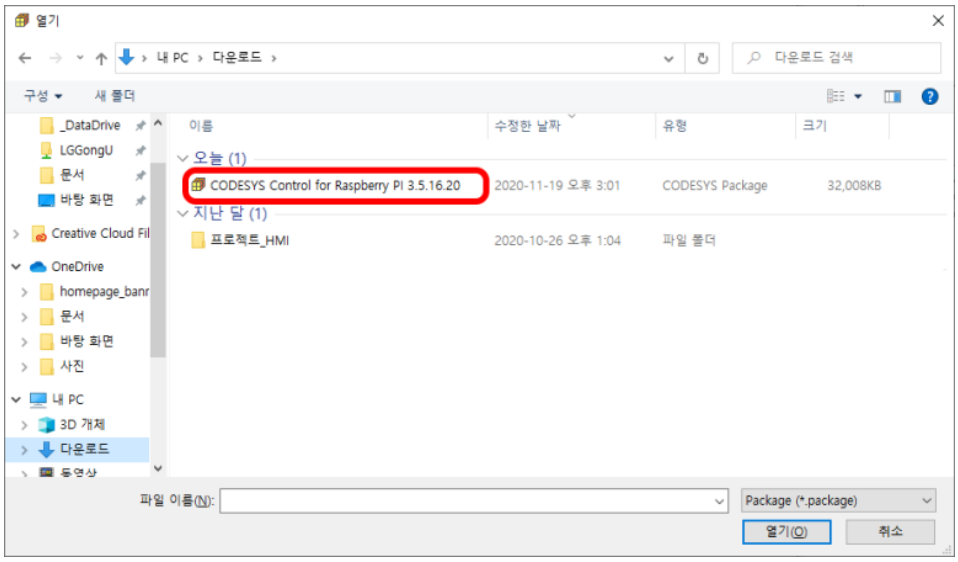
CODESYS 를 실행하고 (다른거 오픈하기 전에) TOOLS 메뉴에 있는 Package Manager 를 선택해주세요.

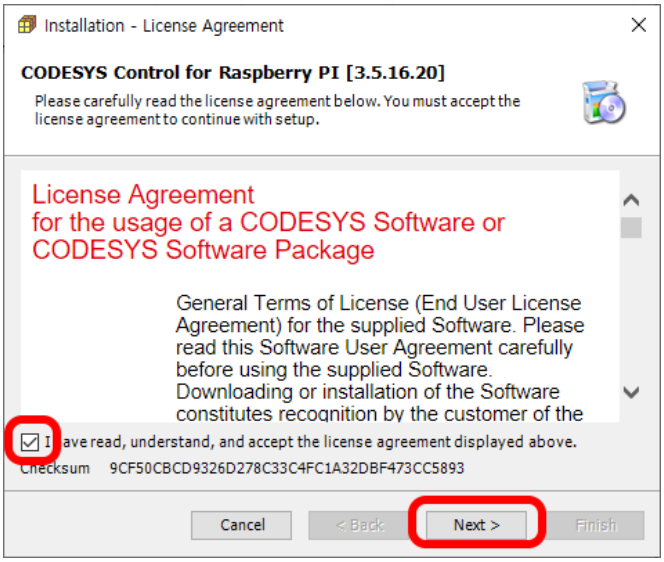


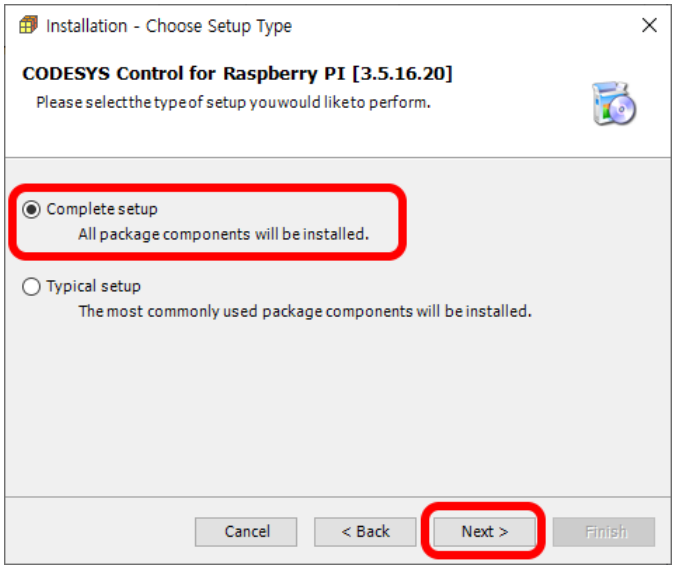
이런 박스가 나오는데 여기서 Install 을 선택하세요.

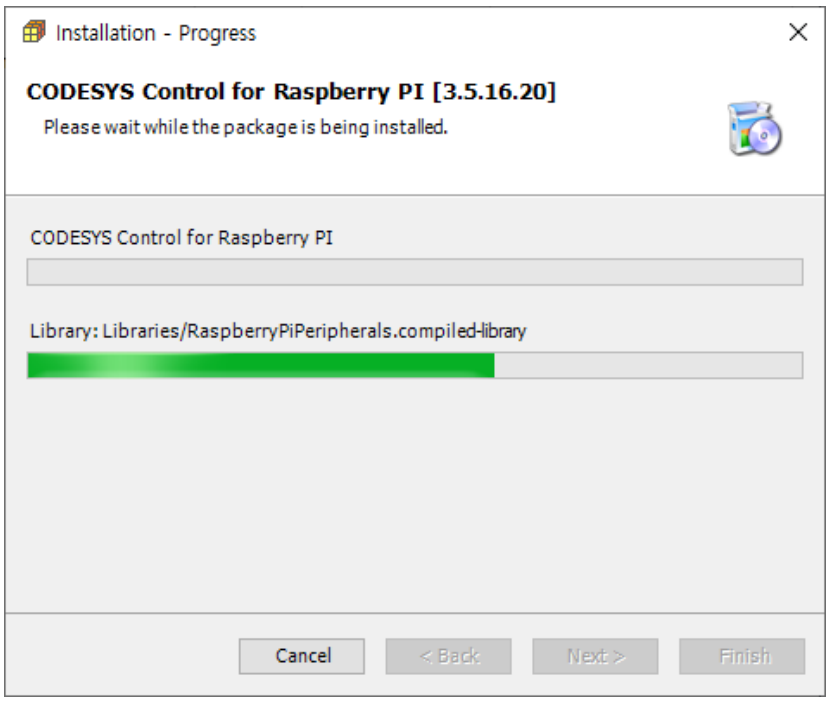


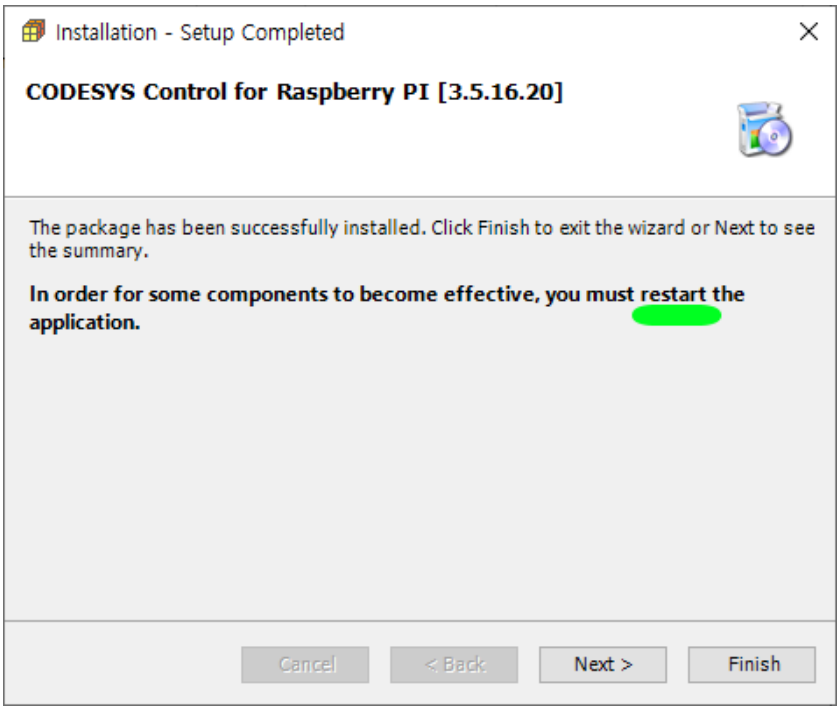
그리고 좀전에 다운로드 받은 CODESYS Control for Raspberry Pi 파일을 선택해주세요.











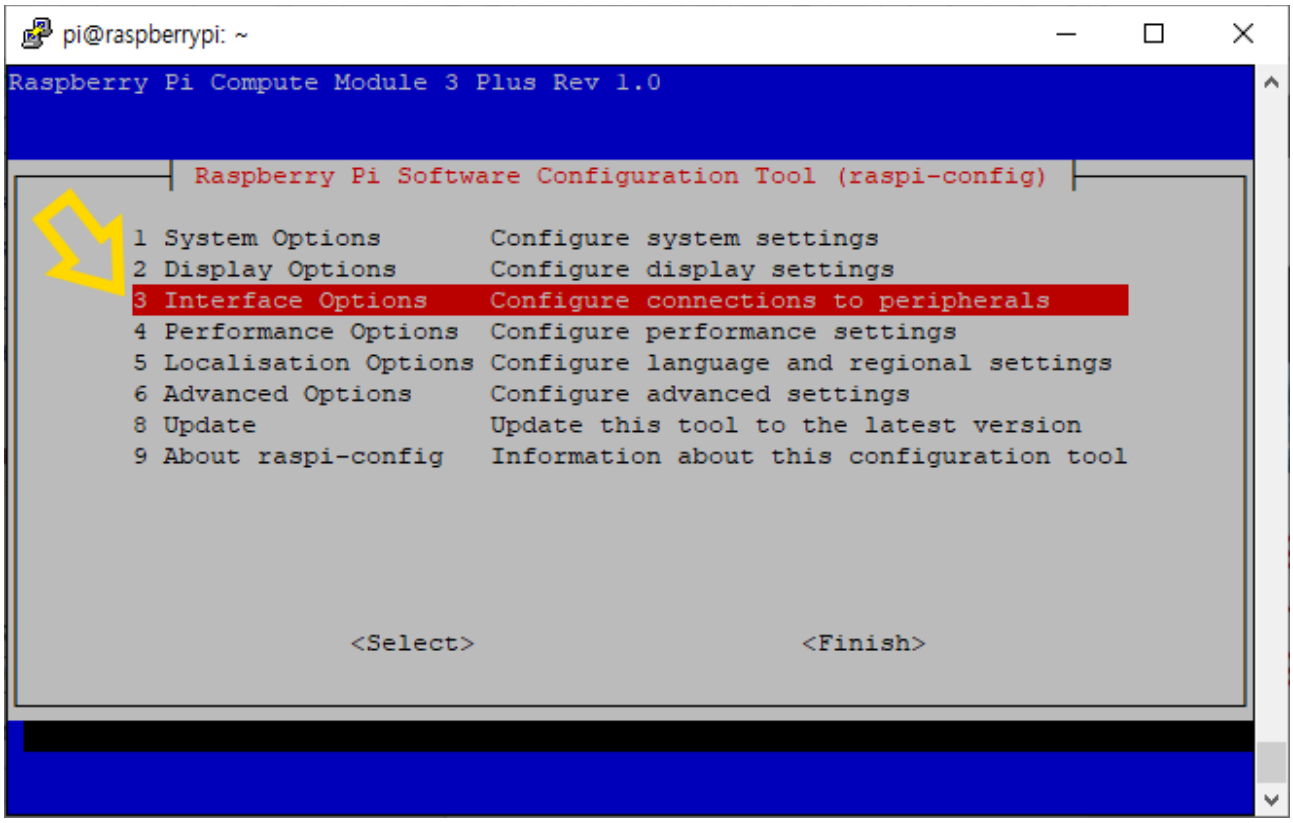
끝났습니다. 다시 ReStart 하라고 써있네요..

SSH 활성화

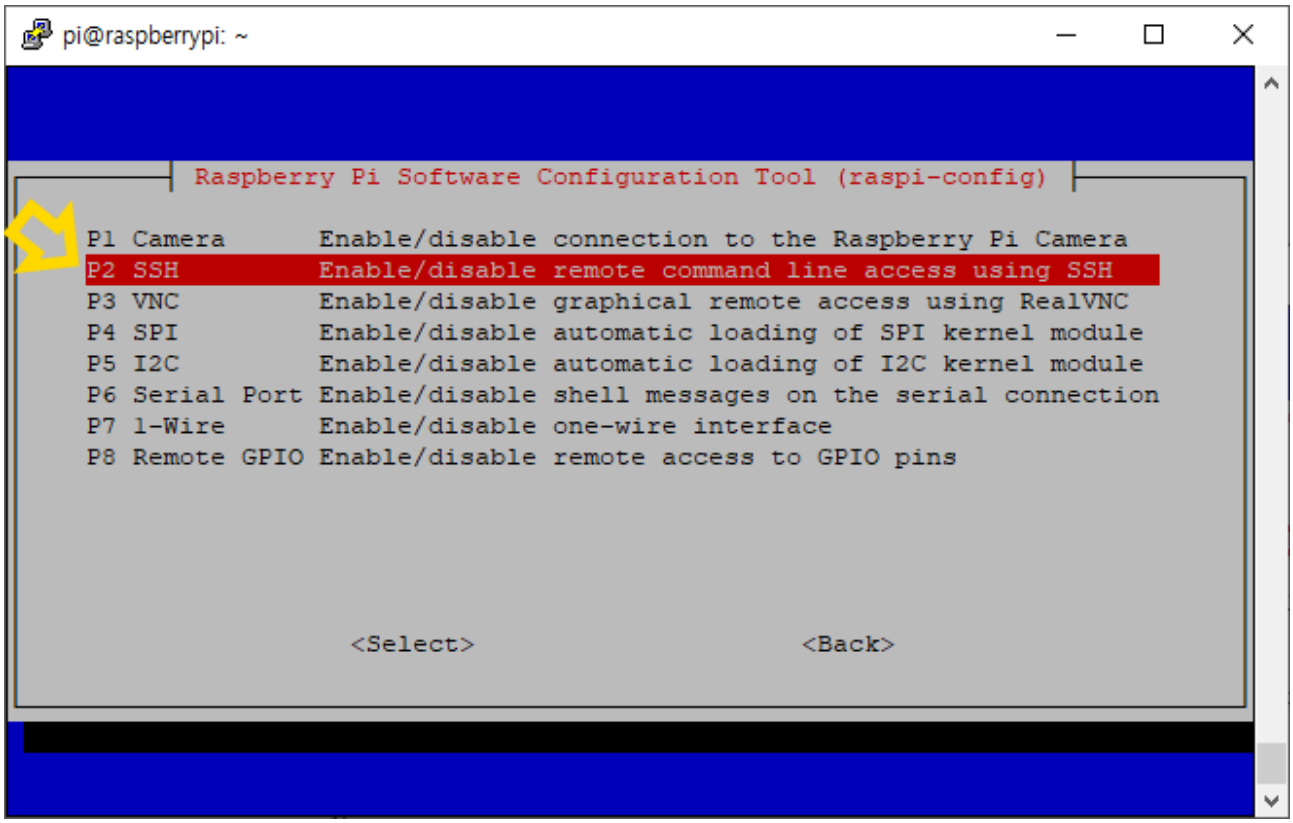
이제 ComfilePi 가 필요한 시점이 되었습니다. 컴파일 파이는 저희 회사(컴파일 테크놀로지 주식회사)에서 만든 산업용 라즈베리 기반 터치패널 PC 입니다. (자세한 내용은 www.comfile.co.kr) 어쨌던 내부에 라즈베리파이가 들어 있습니다. 네트워크 케이블과 키보드, 마우스까지 연결하세요. 이때 네트워크는 작업중인 PC 와 동일 네트워크여야 합니다.

ComfilePi 제품에서 터미널 창을 켜시고 ifconfig 라는 명령어를 입력하면 현재 접속된 IP ADDRESS 를 확인할 수 있습니다. 이거까지 확인해두세요.

그리고 한가지 더 컴파일파이에서 SSH 를 활성화시켜주어야 합니다. 구입시 최초상태에서는 비활성화되어 있습니다. CODESYS 가 SSH 를 통해서 접근합니다. 위 터미널 창에서 sudo raspi-config 를 입력하시면 아래와 같은 창이 뜹니다. 여기에서 interface Options 을 선택하세요.

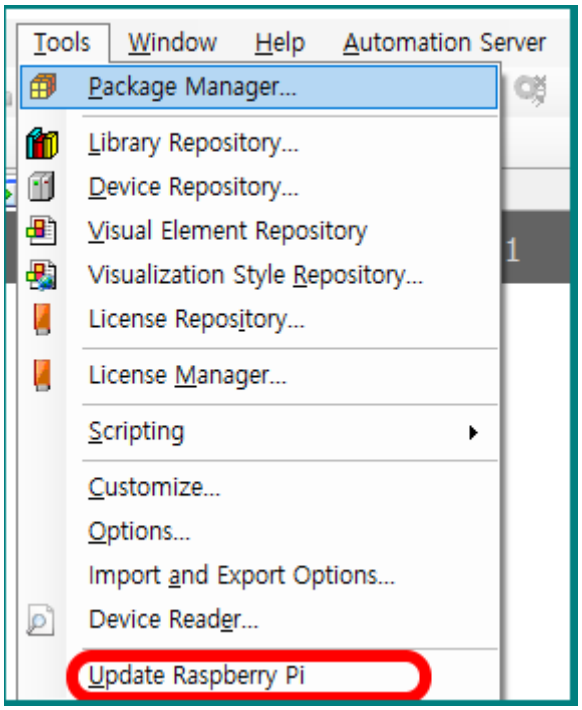


그리고 SSH 를 선택하세요.

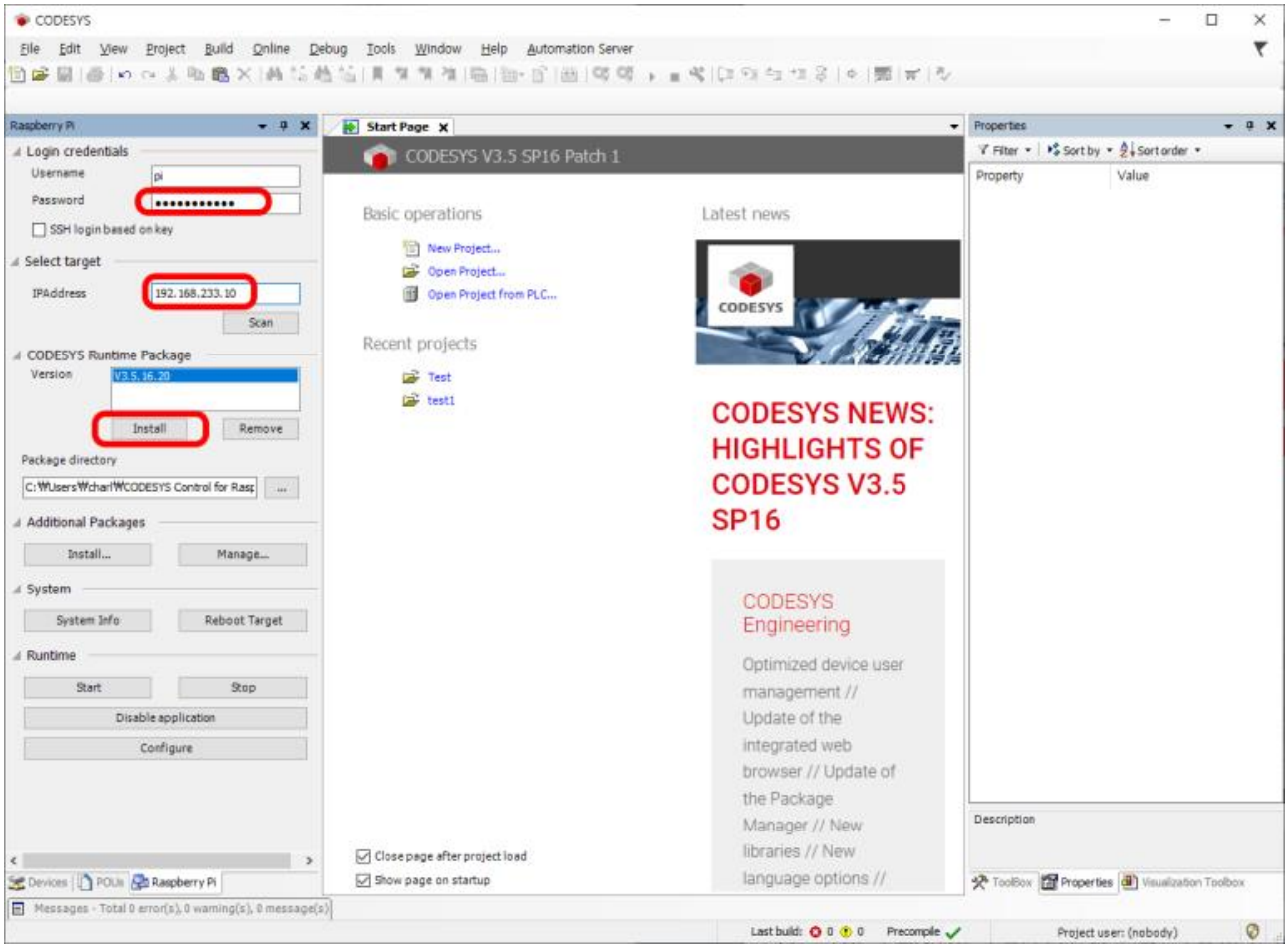


컴파일파이(라즈베리파이)를 한번 리부팅 시켜주세요. 좀 전에 바뀐 설정들이 적용되게 하기 위함입니다.

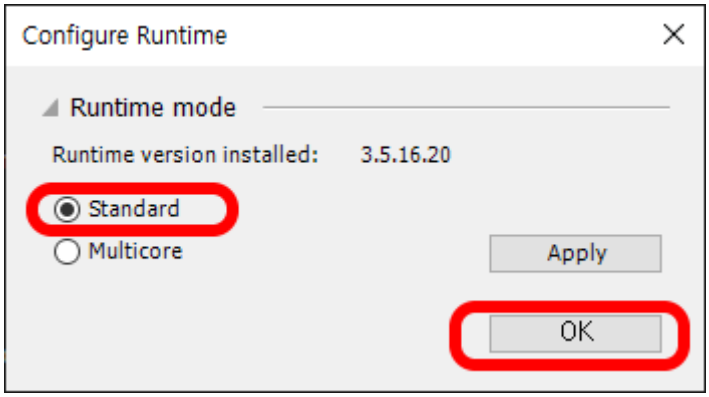
이제 CODESYS 로 와서 Tools 메뉴의 Update Raspberry Pi 를 선택해주세요. (이 작업은 컴파일파이에 CODESYS 관련파일을 설치하는 작업입니다. 최초에 한번만 해주면 다시 해줄 필요가 없습니다.)



비번은 소문자로 raspberry 입력하시고, IP 주소는 좀전에 컴파일 파이 창에서 확인한 번호를 입력하세요. 그리고 Install 을 눌러주세요.

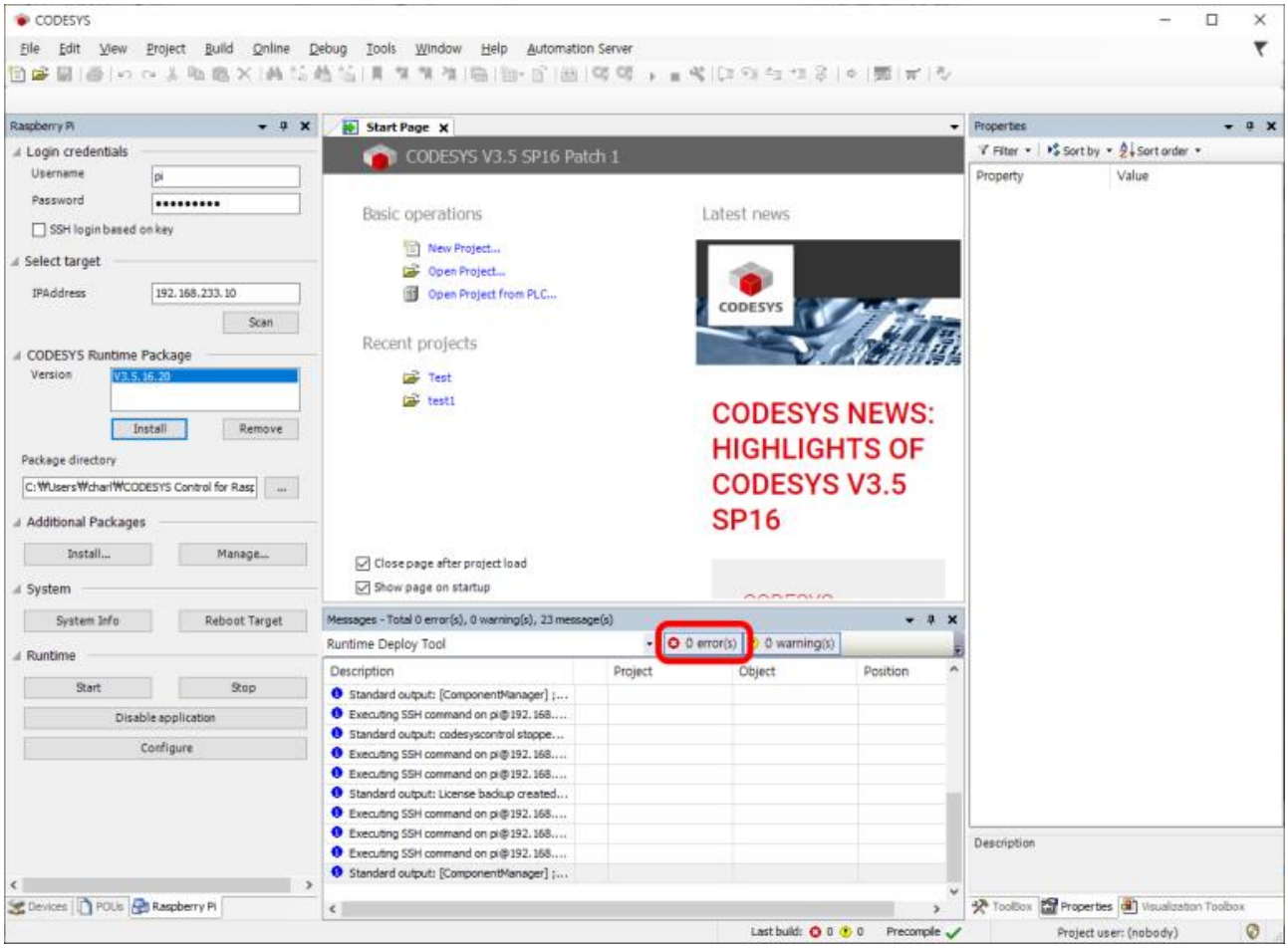


연결과정에서 Standard 와 멀티코어중 하나를 선택하라는 화면이 나오는데, 이건 어떤 라이센스를 사용할 지 물어보는 것입니다.



Multicore 는 멀티코어를 지원하는 라이선스를 구입했을 때, 설치해서 사용하도록 하시고, 여기에선 싱글코어인 Standard 로 진행하겠습니다.

잠시뒤 설치가 끝나면 화면이 이렇게 바뀝니다.

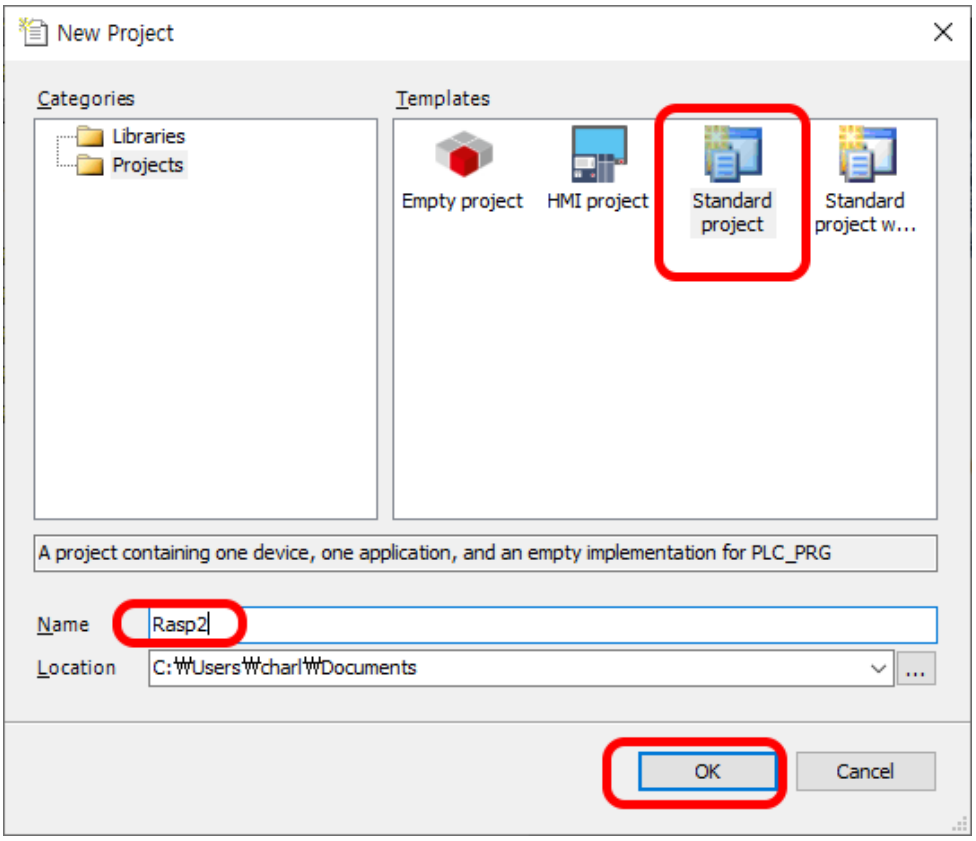


0 Error 를 확인하세요.

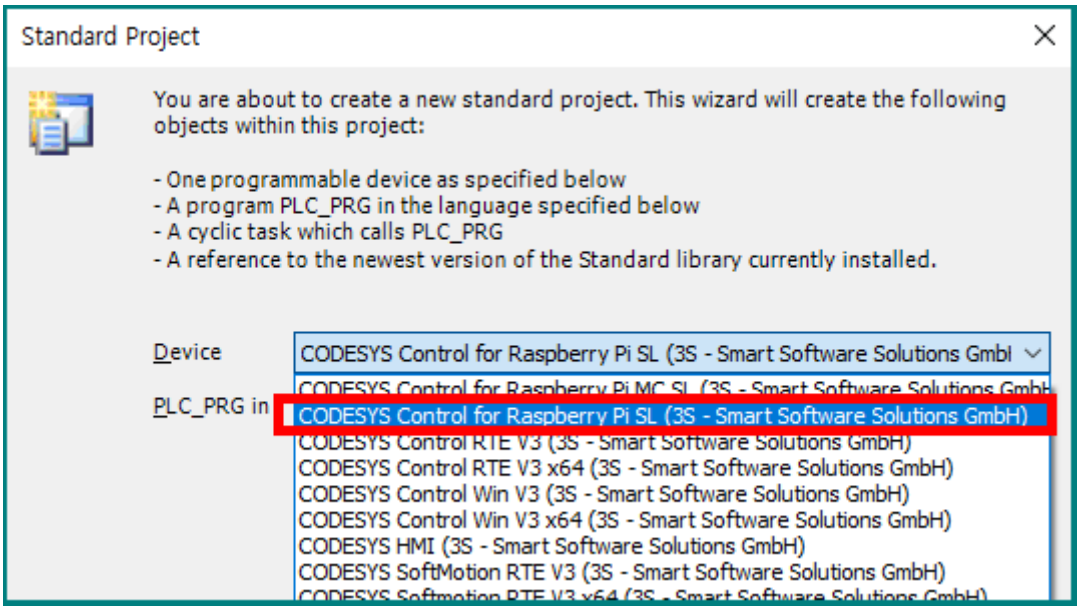
제 3 장 컴파일파이 프로젝트

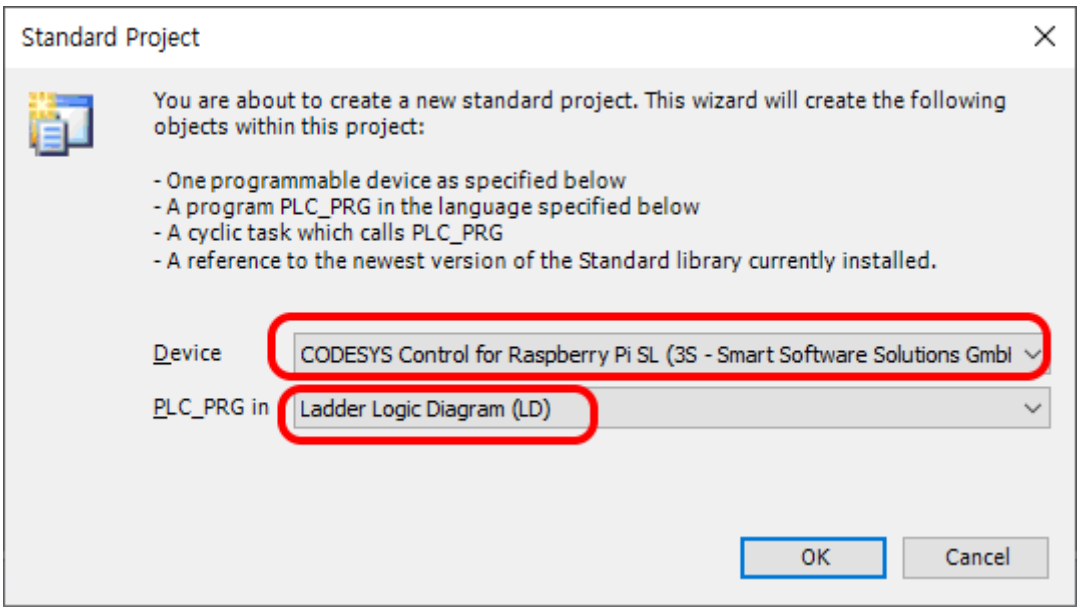
컴파일파이 프로젝트

CODESYS 를 시작한 뒤 New Project 를 선택한 뒤 Standard Project 를 선택하세요.

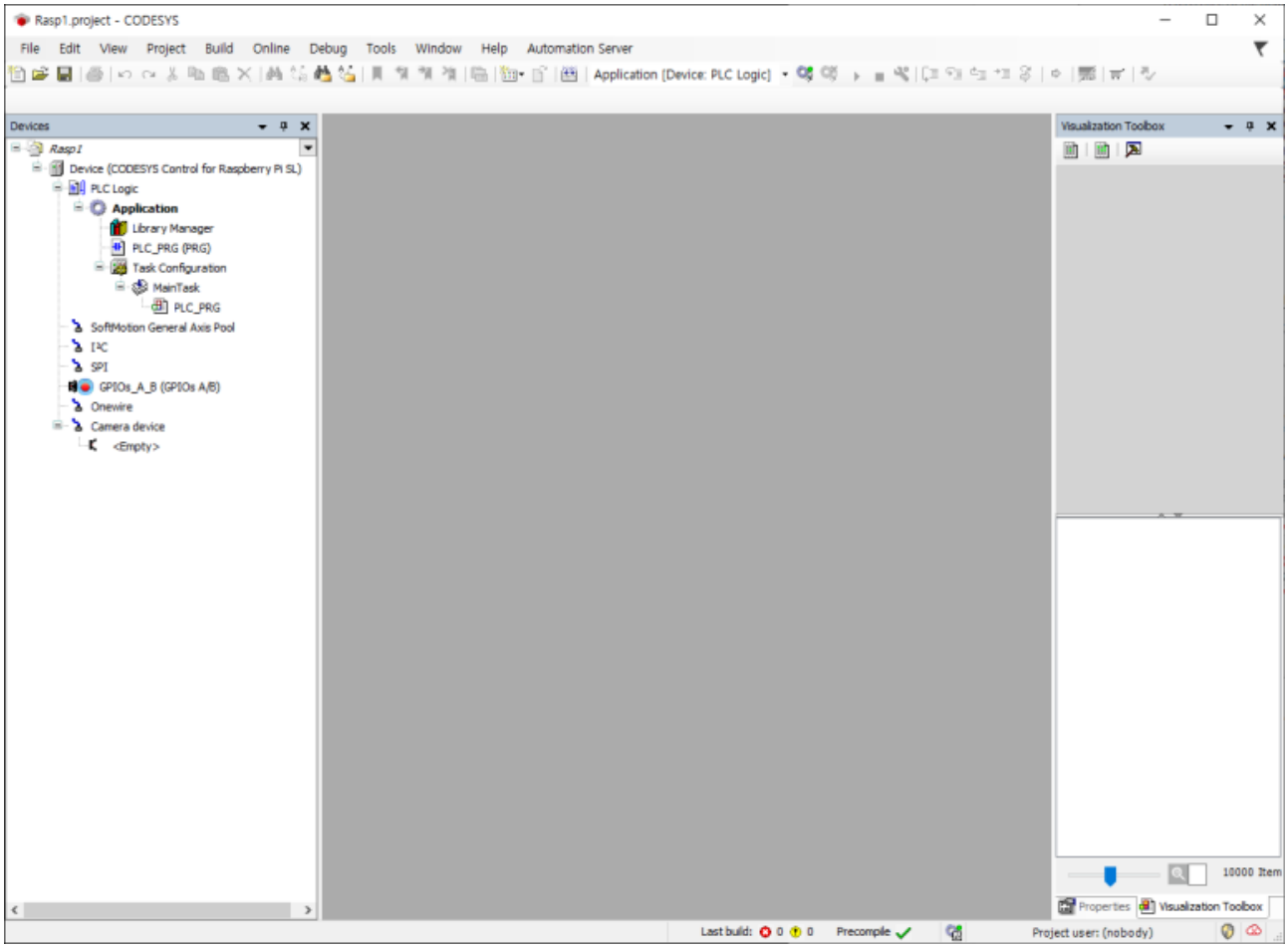


Device 에 처음에는 없었던 CODESYS Control for Raspberry Pi 가 포함되어 있는 것을 보실 수 있습니다. 여기에서 SL 을 선택하겠습니다. (SL 은 싱글 라이선스라는 뜻)

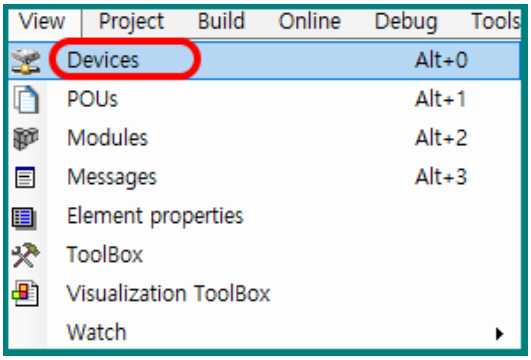




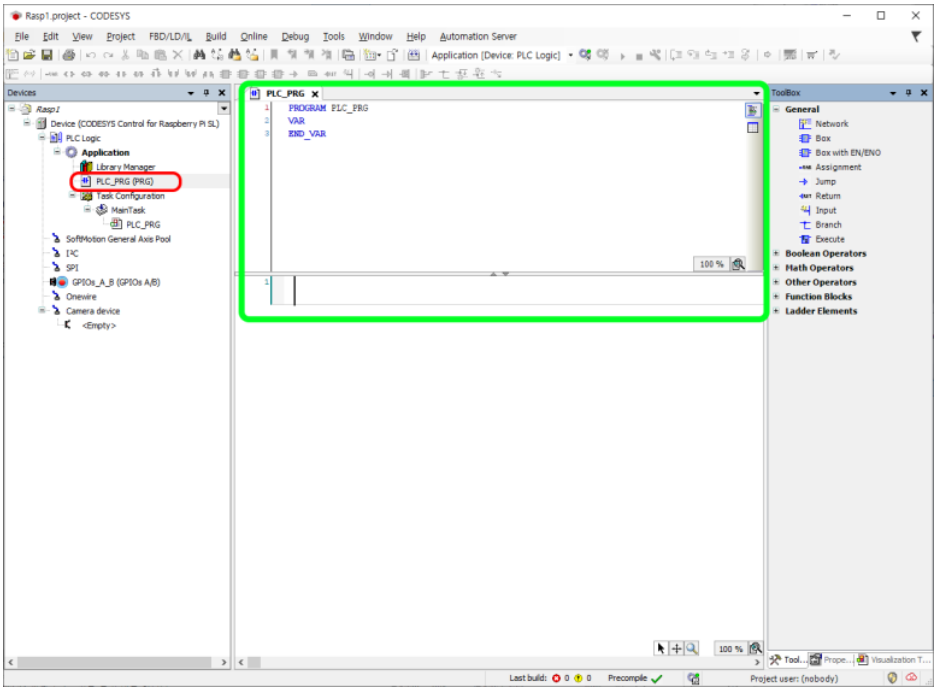
이렇게 선택하고 OK 를 누르세요. 이런 화면이 됩니다.



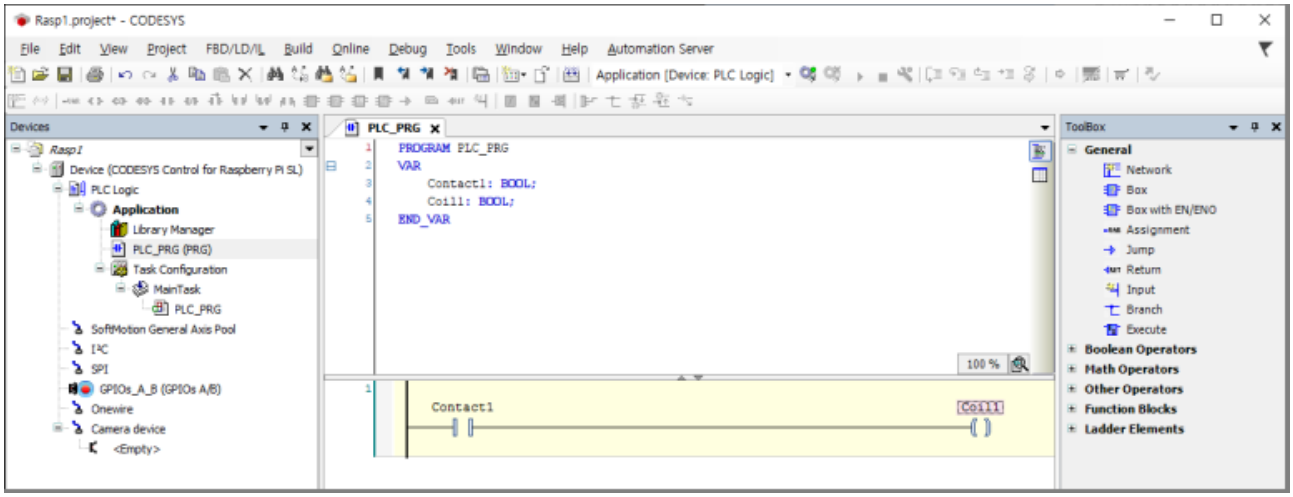
만약 왼쪽에 Device 창이 안보이시면, 당황하지 마시고 View 메뉴에서 Device 를 선택하시면 됩니다.



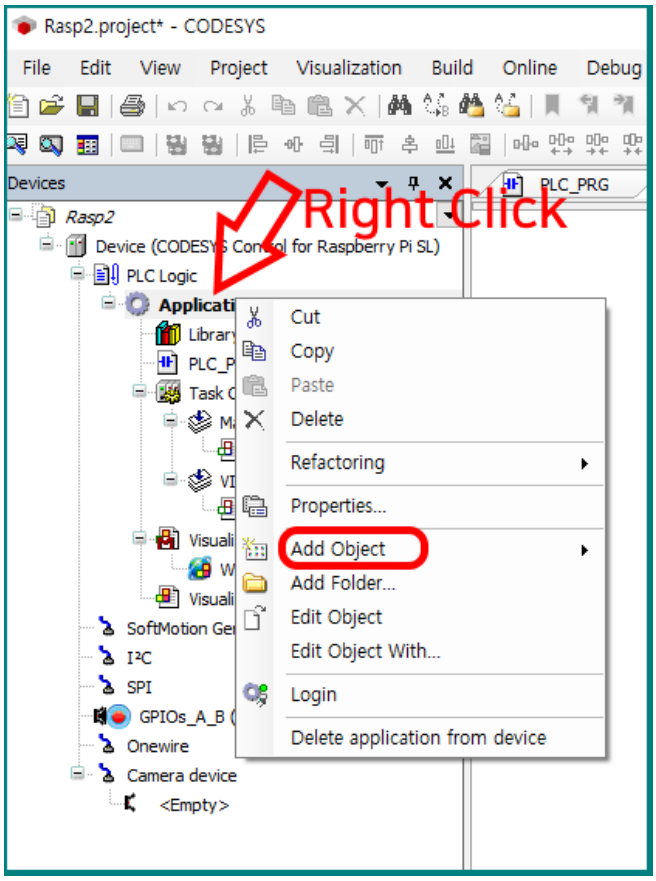
그리고 PLC_PROG 를 더블 클릭하면 가운데 부분이 이렇게 바뀝니다.



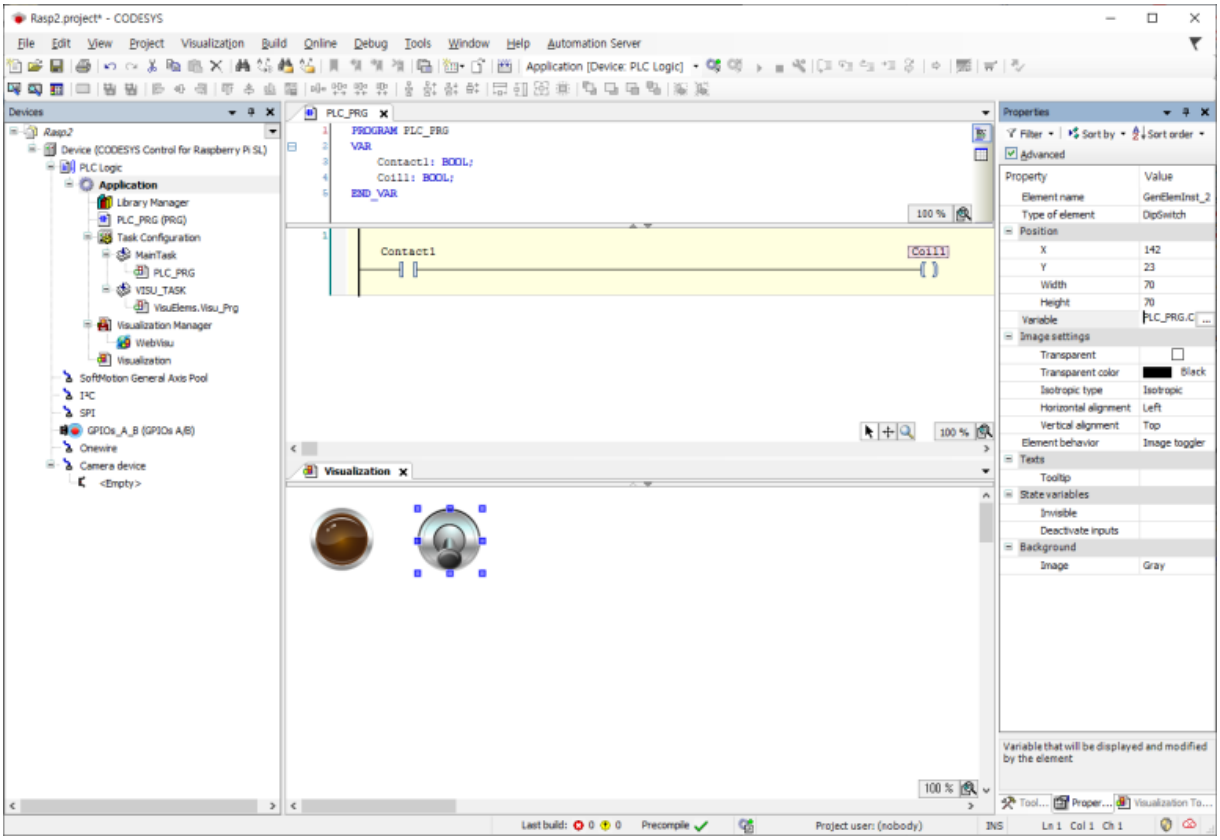
여기에 앞에서 설명한 것과 동일하게 기본적인 레더로직을 입력해보세요.



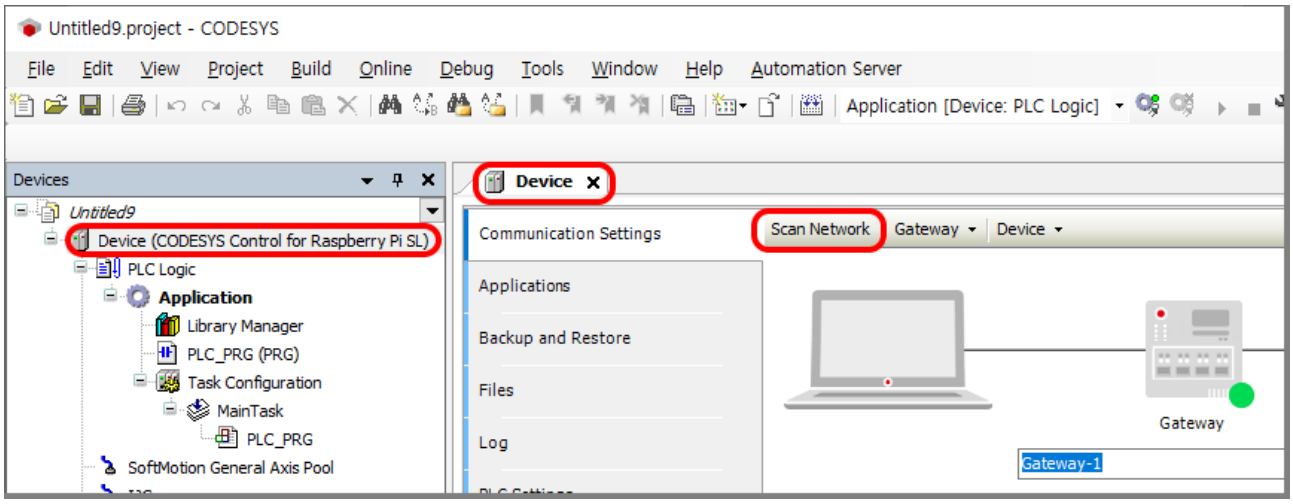
그리고 Visualization 도 만들어보세요. Application 에다 커서를 대고 우 클릭한 뒤 Add Object--> Visualization 을 선택해야합니다.



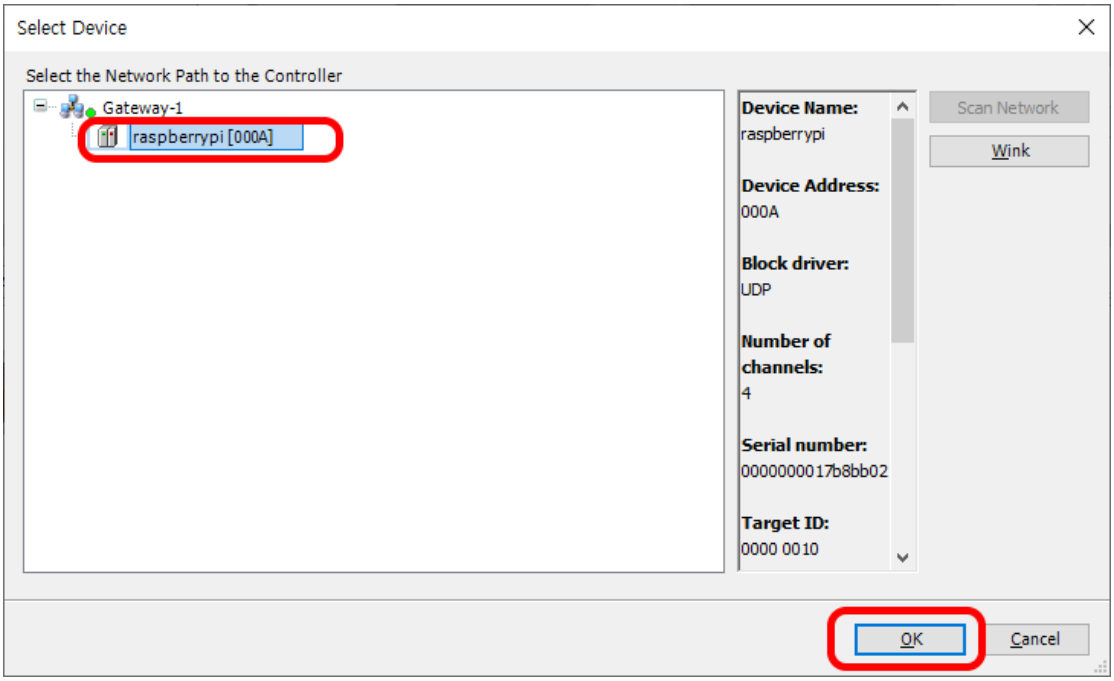
앞에서 설명한대로 똑같이 만들어보세요.



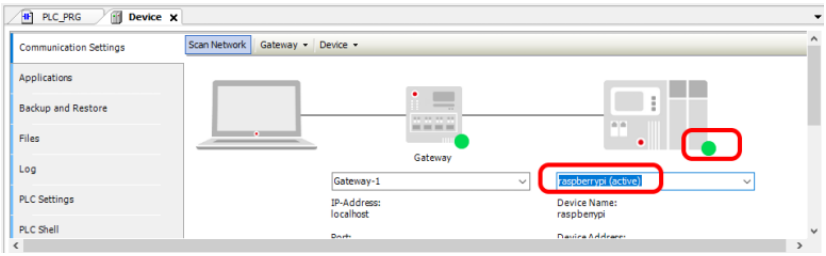
그 다음 컴파일 파이와 연결해야 합니다. Device 를 더블 클릭하면, Device 창이 나옵니다. 여기에서 Scan Network 를 클릭해주세요.



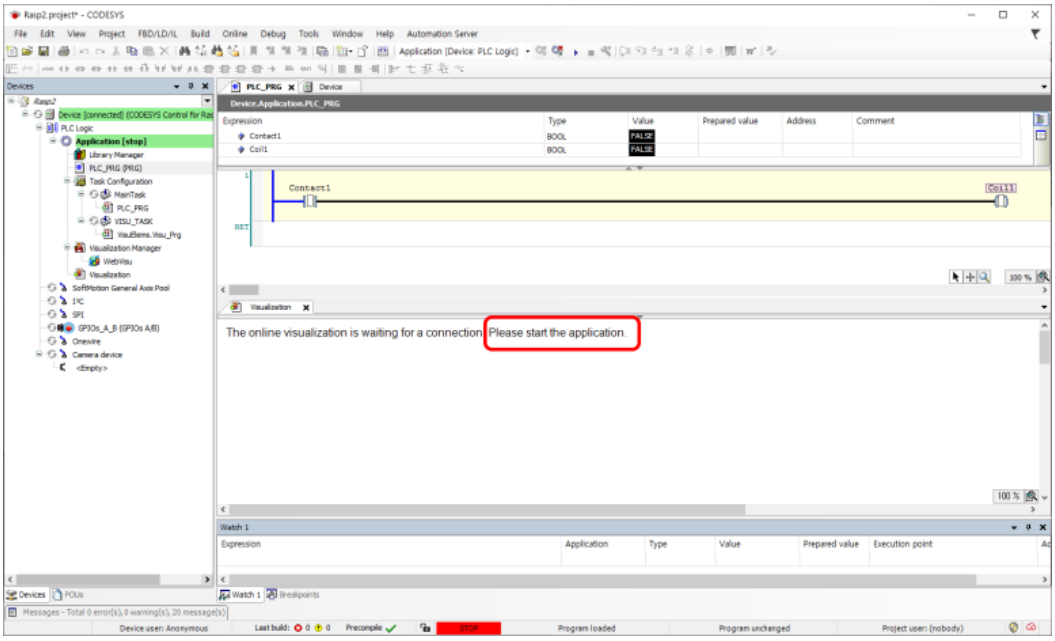
잠시 뒤 네트워크를 스캔한 결과가 나옵니다. raspberrypi 를 선택하고 OK 를 눌러주세요. 만약 스캔에서 못 찾는다고 나오면 동일네트워크가 아닌 경우입니다.(동일 네트워크란? 같은 라우터/인터넷 공유기에 연결된 것을 의미합니다. 같은 네트워크인데도 못 찾으면 컴파일파이를 한번 껐다가 켜보세요.)



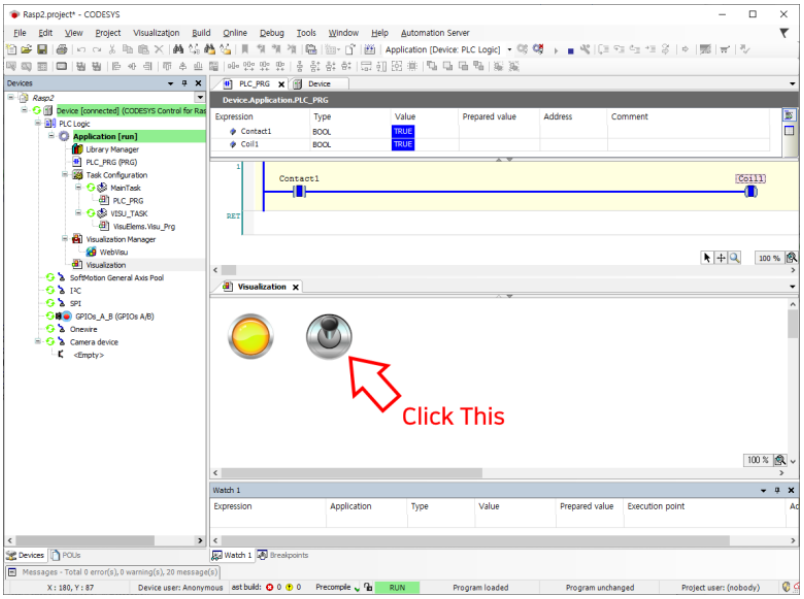
화면이 이렇게 바뀌었다면 9 부능선을 넘은 것입니다. CODESYS 와 컴파일파이(라즈베리파이)가 연결된 것입니다. 이렇게 네트워크를 스캔하고 컴파일파이(라즈베리파이)를 연결하는 일은 CODESYS 를 실행시킨 뒤 최초에 한번은 꼭 해주어야 합니다.



F11 을 눌러서 Build 한번 해주시구요. Online 메뉴의 Login 을 누르면 잠시뒤 화면이 이렇게 바뀝니다. (이 과정에서 실행파일이 컴파일파이쪽으로 배포됩니다.)



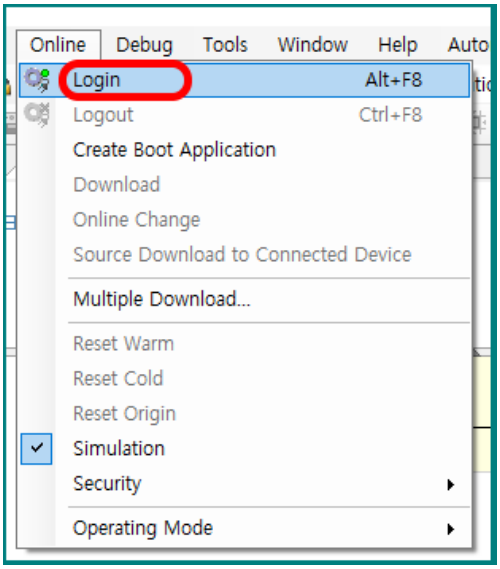
최초에는 STOP 상태가 되어 있습니다. 화면에도 Start 를 하라고 되어 있습니다. F5 키를 누르면 시작됩니다.시작한 뒤 화면의 딥 스위치를 눌러보세요. 램프가 On/OFF 되면 성공입니다.



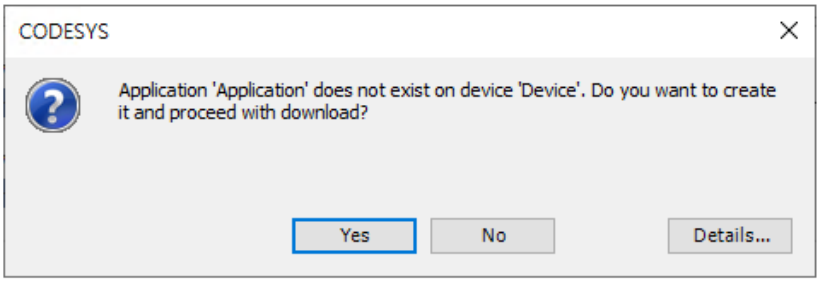
여기까지 해서, 개발용 PC 에서 작성한 CODESYS 프로젝트를 컴파일파이쪽에 배포하고 실행시키는 것까지 해보았습니다.

배포(로그인)과 자동시작

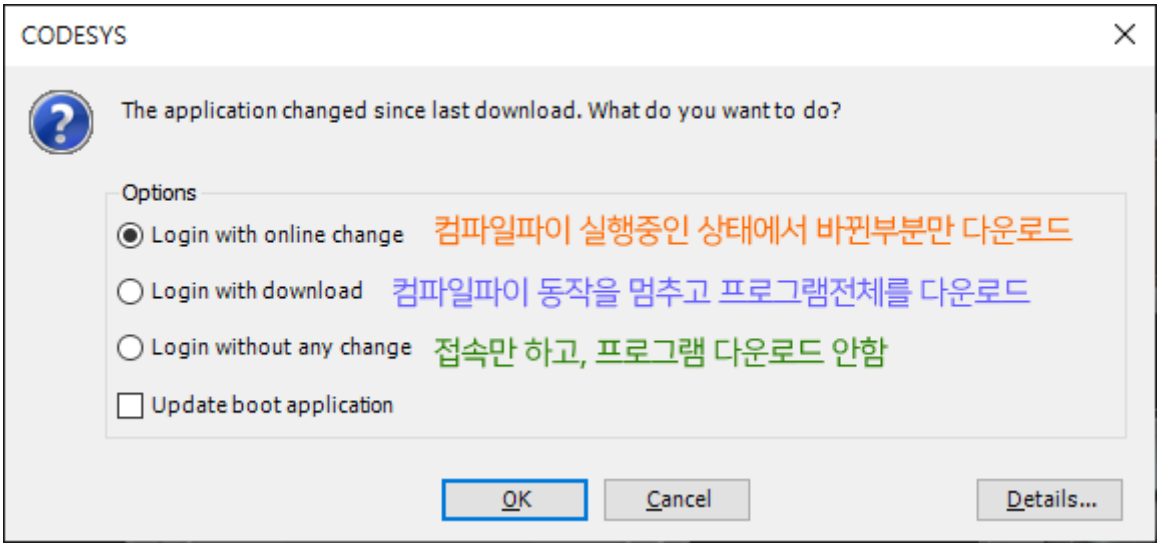
온라인메뉴에서 로그인(또는 Alt+F8)하면 실행파일이 컴파일파이 (라즈베리파이)쪽으로 배포(다운로드)된다고 앞에서 설명드렸습니다.



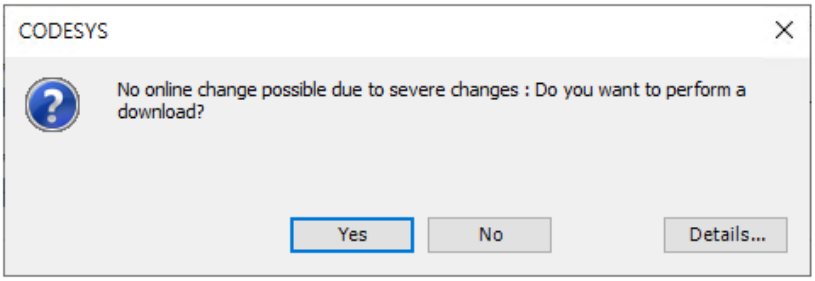
만약, 비어있는 컴파일파이(라즈베리파이)에 최초로 배포할때에는 다음과 같은 멧세지가 표시됩니다. Yes 를 누르면 됩니다.



뭔가 실행중이라면 이런 선택사항이 표시되는데, 각각의 옵션이 의미하는 바는 아래와 같습니다.

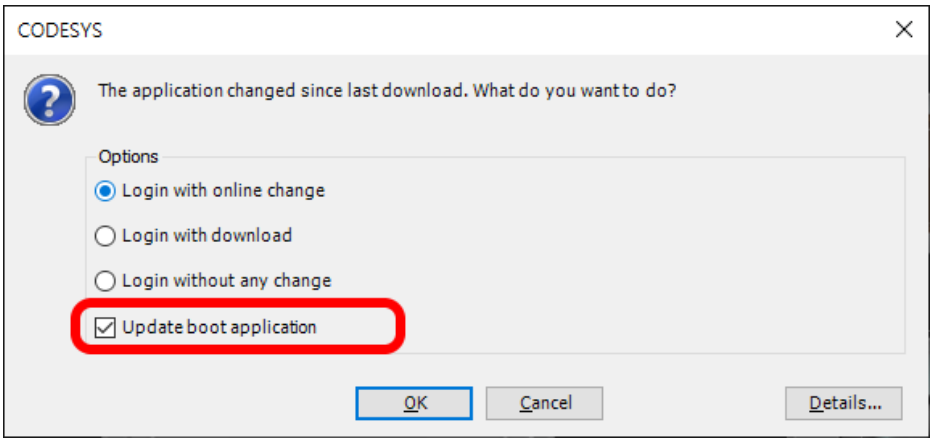


맨위에 있는 Login with online change 를 선택했는데, 큰 규모의 변경이 있어서 도저히 실행중인 상태에서 바꾸는게 불가능할 경우 아래 박스가 표시됩니다.

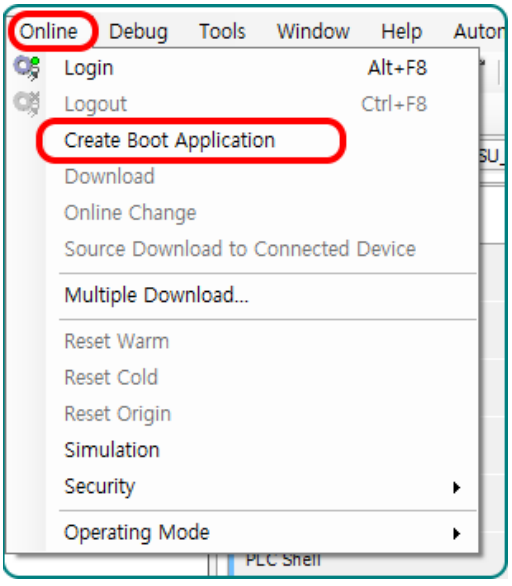


여기서 Yes 를 누르면, 두번째 옵션인 "Login with download" 즉, 실행중인 프로그램을 멈추고, 전체 프로그램을 다운로드합니다.

맨 아래 Update boot application 에 체크를 하면, 추후 전원이 켜질때마다 자동시작됩니다.



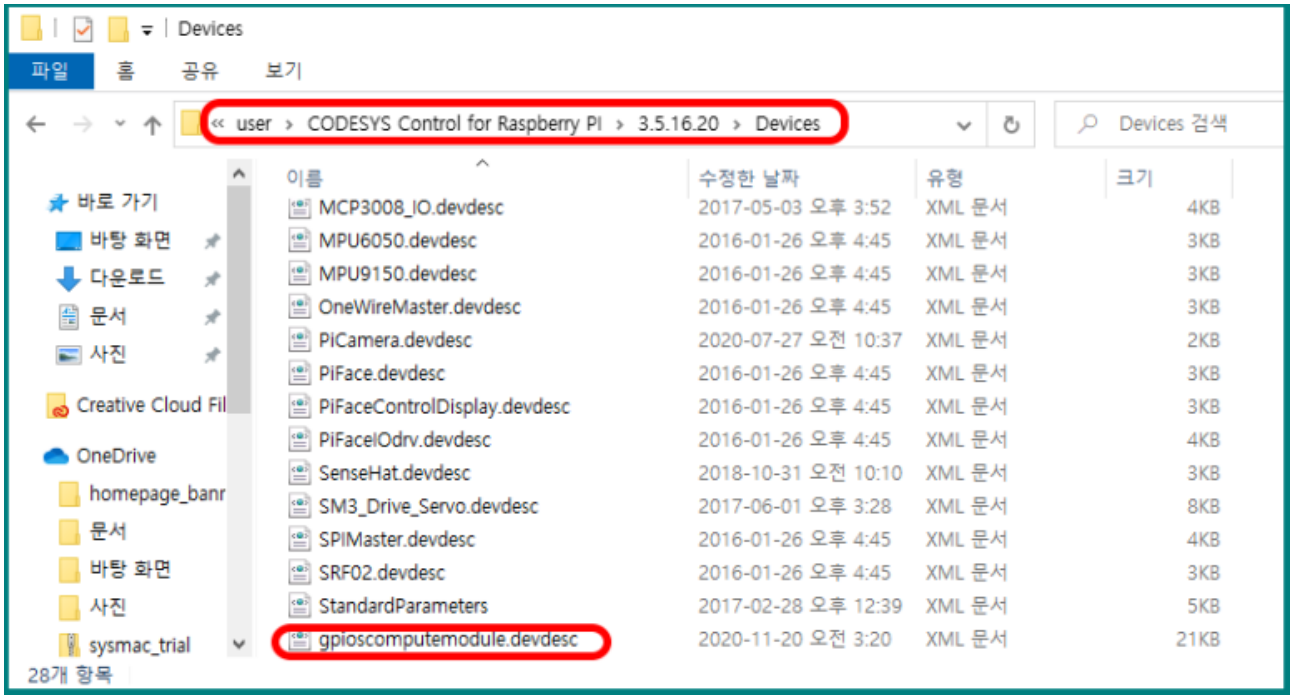
만약 위 팝업박스에서 자동시작 선택을 깜박했다면, 직접 Online 메뉴에서 Create Boot Application 을 한번 실행시켜주세요. (Online 상태에서) 그러면 자동시작되도록 설정됩니다.



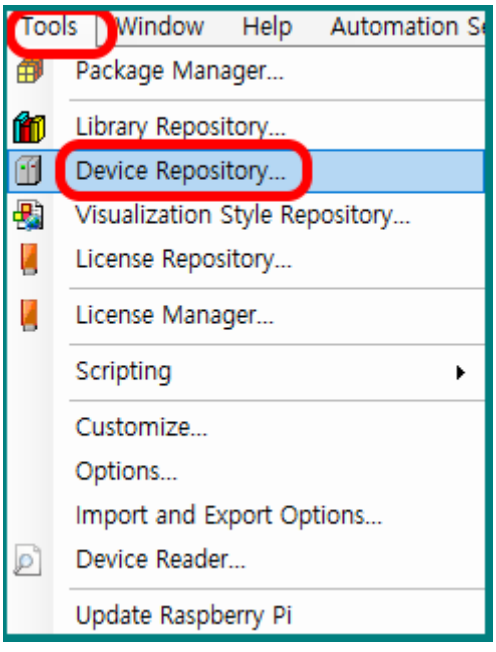
GPIO 셋업 변경

컴파일 파이 뒷면에 있는 GPIO 를 CODESYS 에서 쓸 수 있도록 조치(?)를 한번 해보겠습니다. 컴파일 파이는 오리지널 라즈베리파이가 아닌 Compute 모듈을 사용하고 있기 때문에, GPIO 셋팅을 Compute 모듈로 바꾸어 주어야 합니다. https://www.comfile.co.kr/download/pi/gpioscomputemodule.devdesc.zip

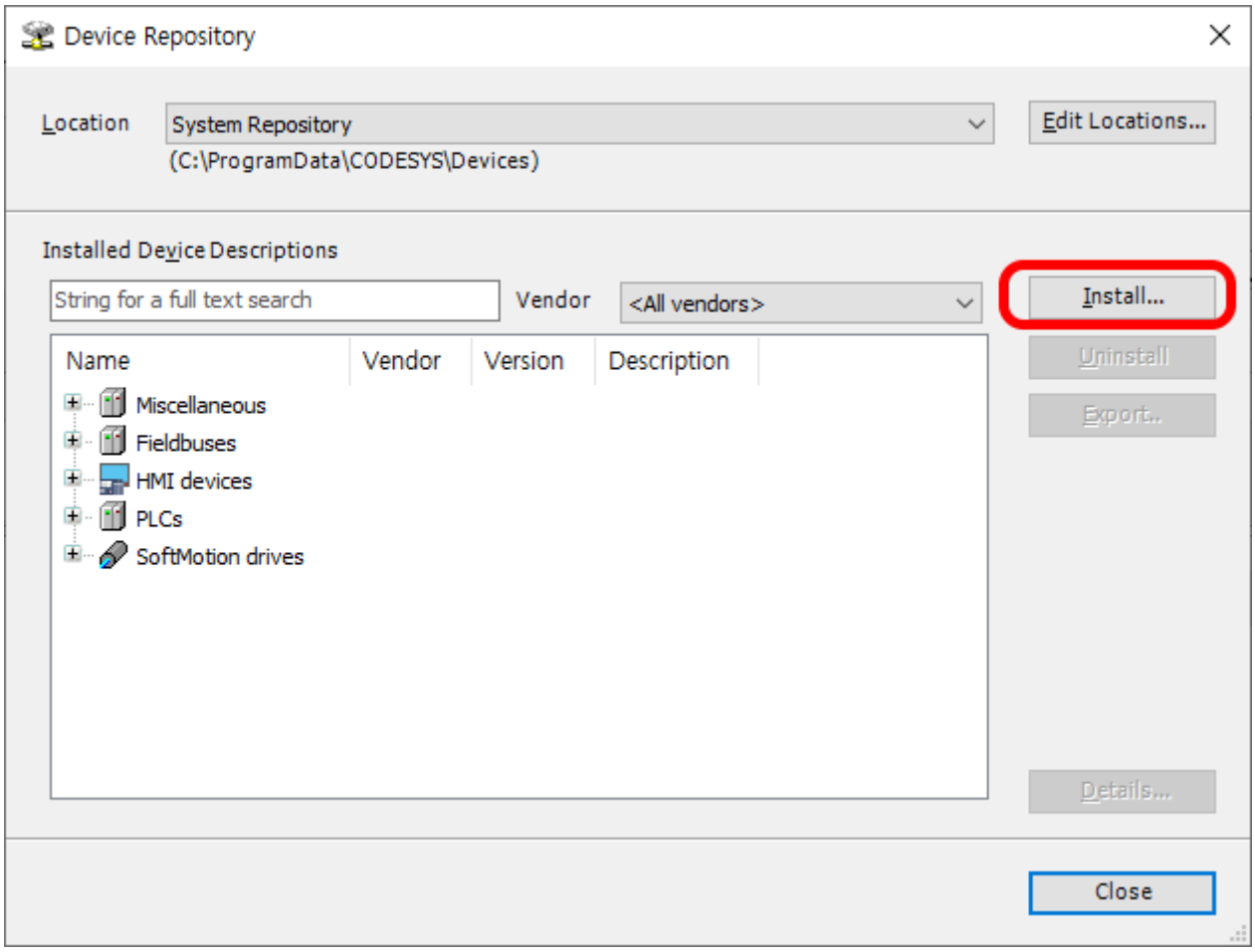
이 파일을 다운로드한뒤 압축을 풀어주세요. xml 파일이 하나 나옵니다. 이 파일을 c:₩사용자₩CODESYS Control for Raspberry Pi₩버전번호₩Devices 아래 복사해주세요.



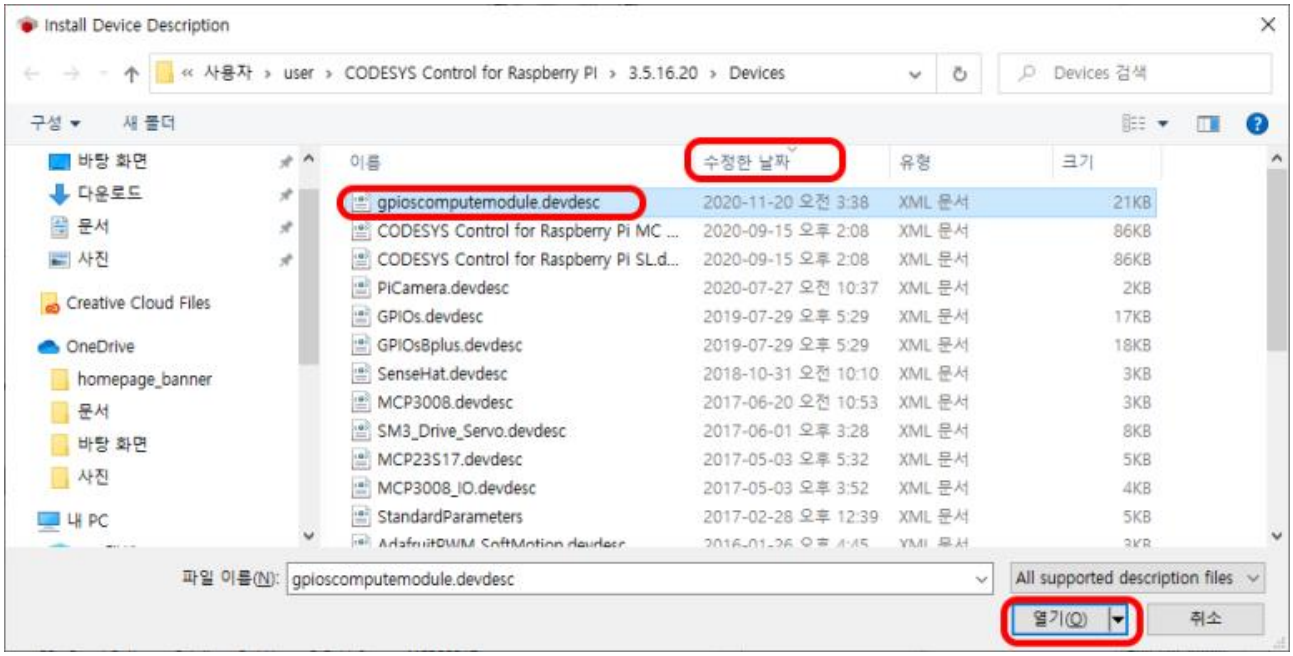
그리고 Tools 메뉴에서 Device Repository(리포지토리)를 선택하세요.



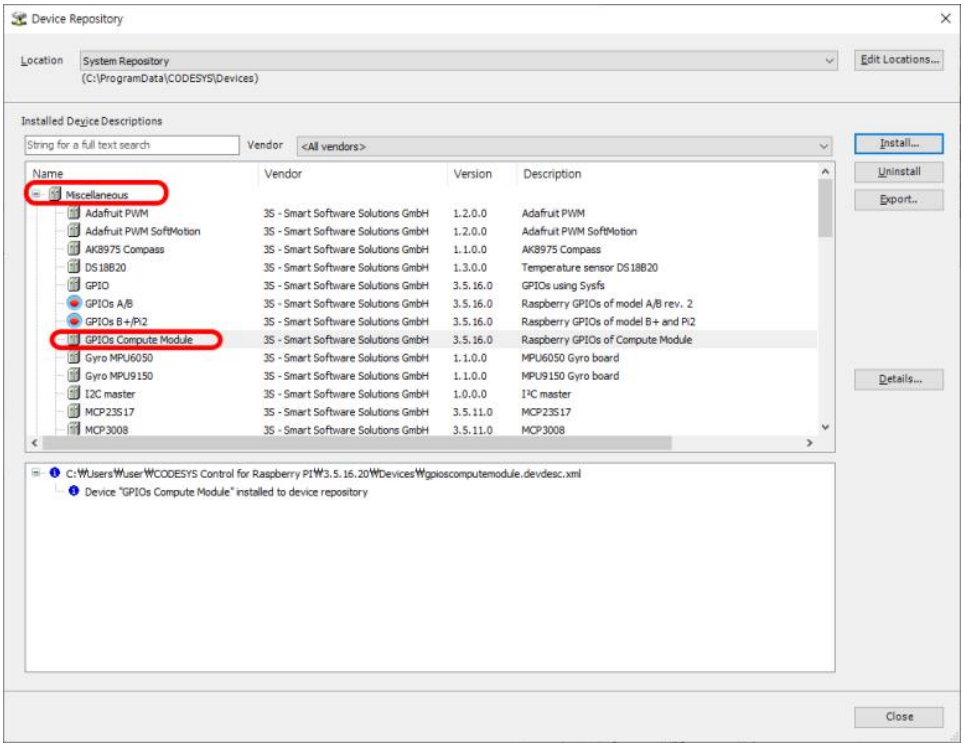
아래 박스에서 Install 누르시고,



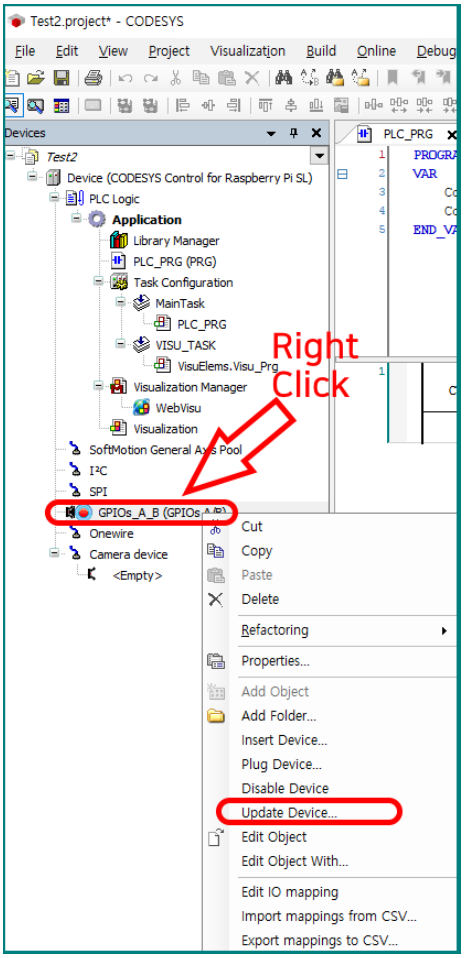
파일을 찾는 창이 뜨면, 방금전 그 폴더에 가서 옮긴파일을 선택하세요. (날짜순으로 보면 가장 최근위치에 있음). 그리고 열기를 누르세요.



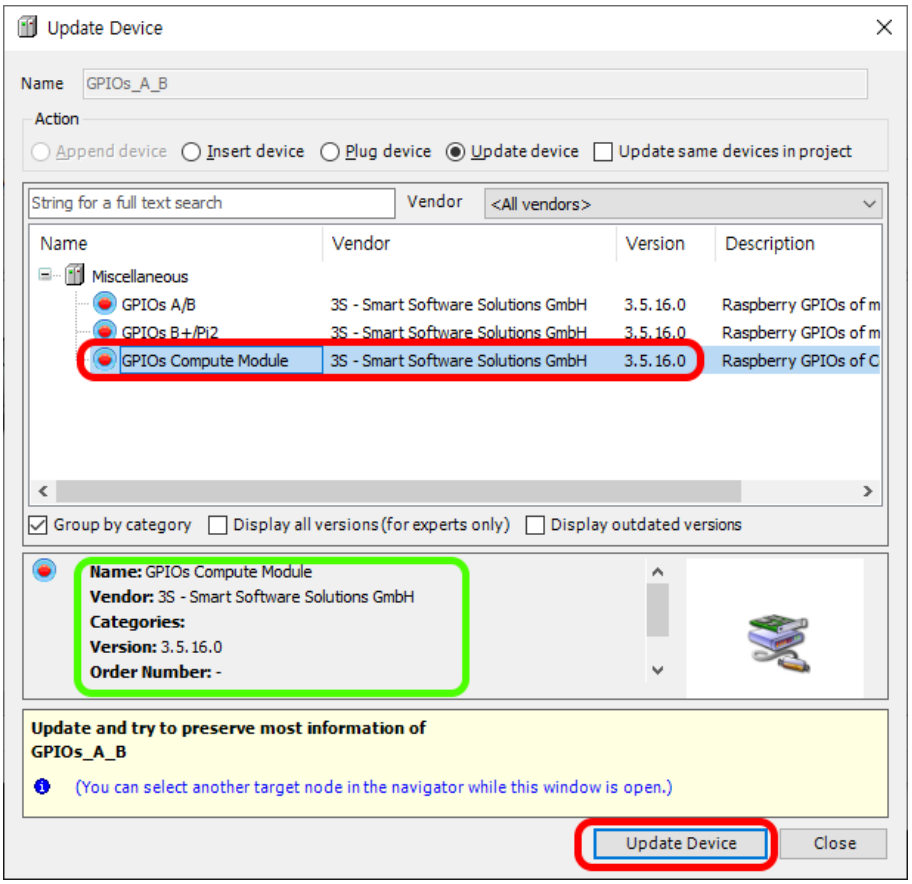
Miscellaneous 를 펼쳐보면 방금 추가한 내용을 확인해 보실 수 있습니다. (원래는 없었어요!)



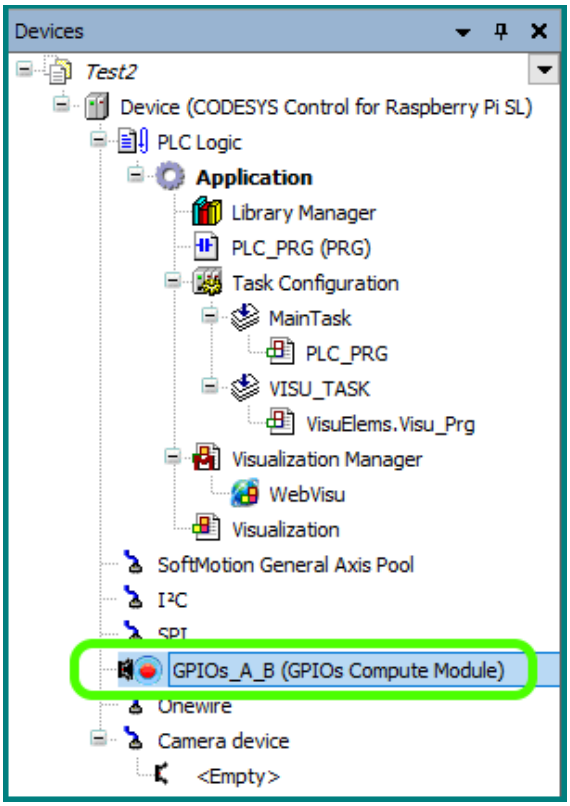
그 다음 왼쪽에 있는 Device 창에서 GPIO_A_B 를 우클릭한뒤 Update Device 를 누르세요.



이런 창이 뜹니다. 여기에서 방금 설치했던 GPIOs Compute Module 을 선택하고, 하단의 Update Device 를 누르세요.

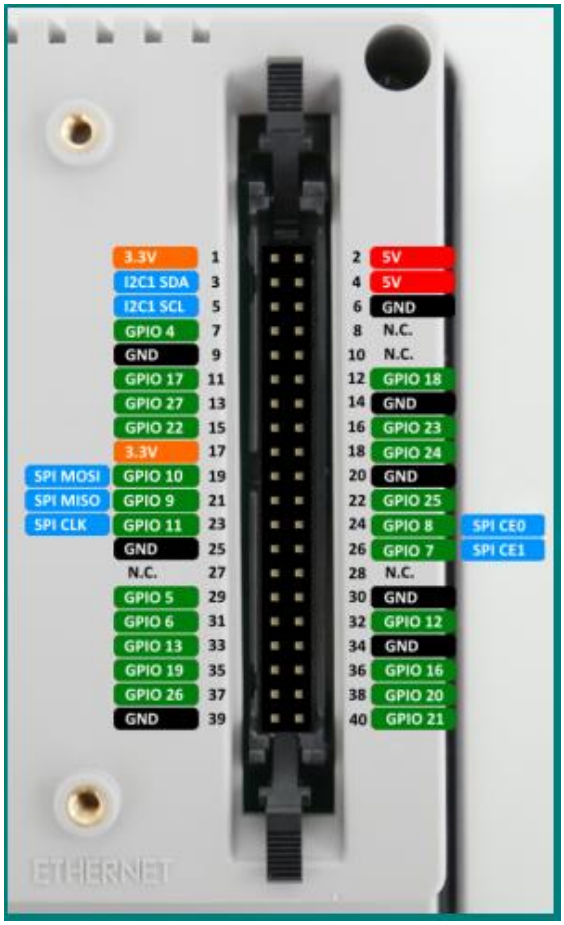


이렇게 바뀌었으면 성공한 겁니다.



GPIO와 CODESYS연결

드디어 CODESYS 내부에서 벌어지는 일들이 외부 I/O 와 연결되는 순서에 도달했습니다. 컴파일 파이 뒷면에 보면 40 핀의 헤더소켓이 존재하고 여기에 GPIO 가 배열되어 있습니다.



대충보면 GPIO 4 번부터 시작해서 일부번호는 빠져있고 (내부적으로 SERIAL 등에 사용됨) 맨 끝번호는 GPIO 27 번입니다. (오리지널 라즈베리파이와 핀배치가 동일합니다.) 그리고 저희 회사 제품중에 이곳에 부착할 수 있는 I/O 보드가 3 종이 있습니다. 이중 CP-IO22 와 CP-IO19R 을 CODESYS 와 함께 사용할 수 있습니다.

CP-IO22
(A02001)
44,000원 (VAT 포함)

ComfilePi용 I/O보드
- DC입력 11점
- NPN TR (싱크)출력 11점
- 옵토커플러를 사용한 전원분리형

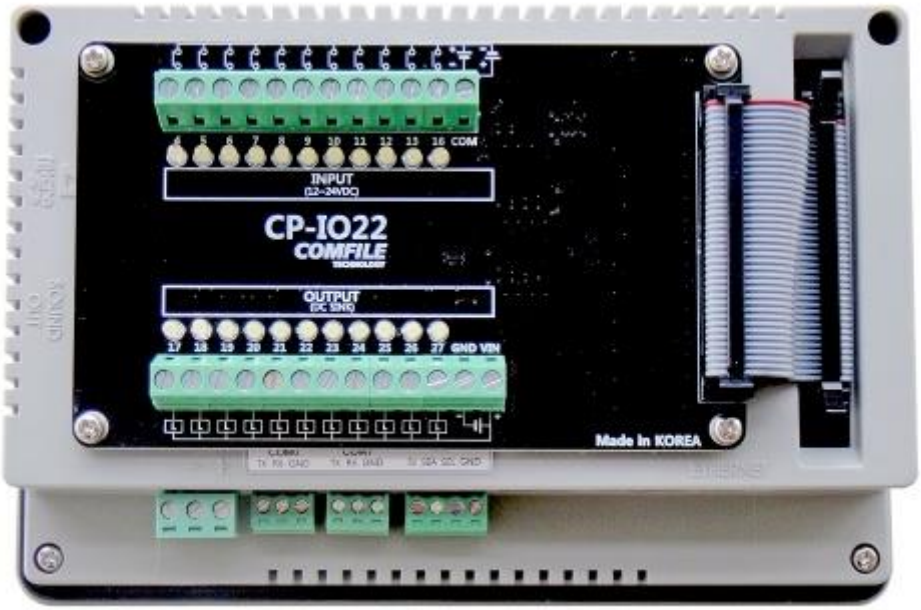
CP-IO19R
(A02003)
55,000원 (VAT 포함)

ComfilePi용 I/O보드
- DC입력 11점
- Relay출력 8점
- 옵토커플러를 사용한 전원분리형

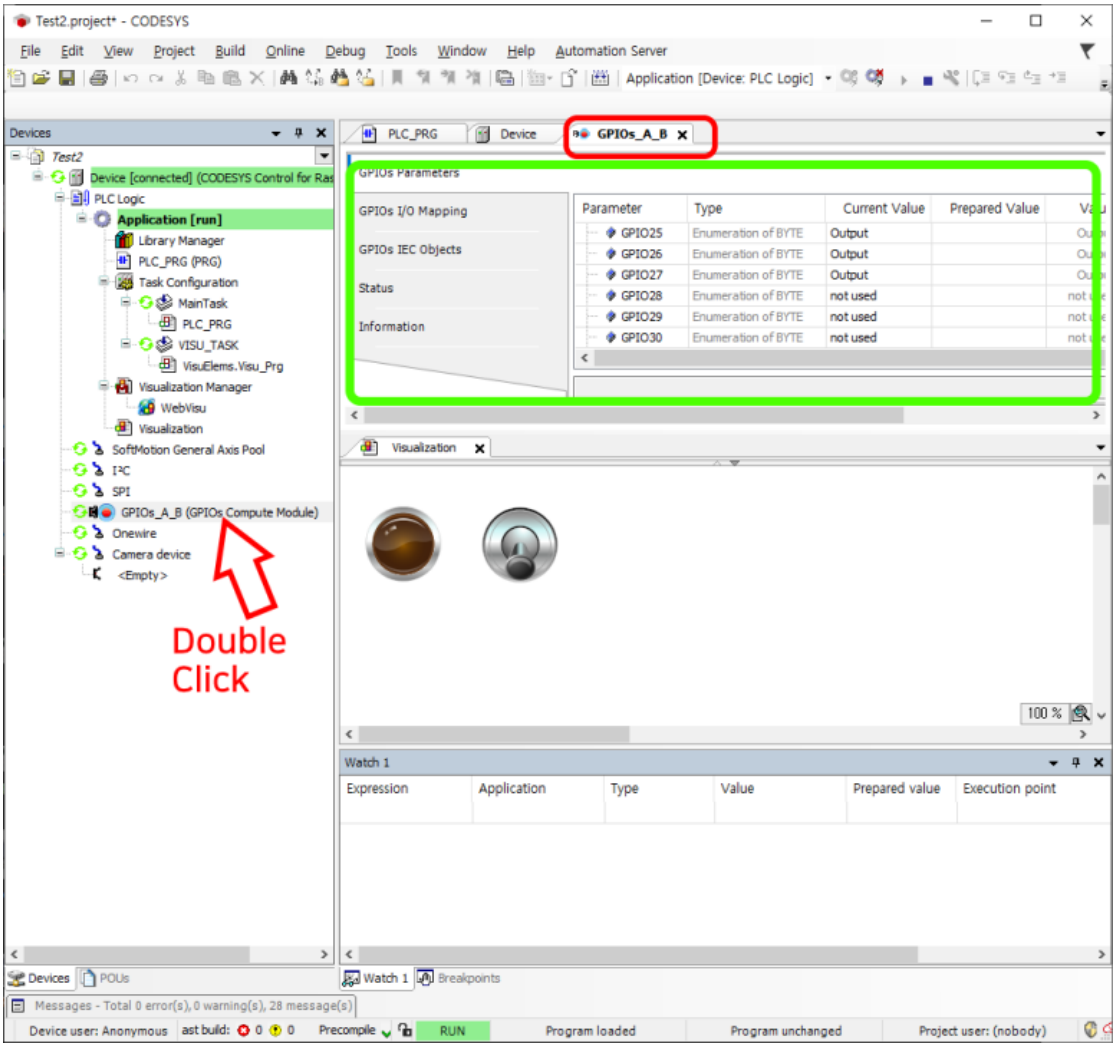
CP-IO22-A4-2
(A02004)
88,000원 (VAT 포함)

ComfilePi용 I/O보드
- DC입력 11점
- DC NPN출력 11점
- ADC 4 채널 (16비트)
- DAC 2 채널 (12비트)
- 디지털쪽은 옵토커플러사용

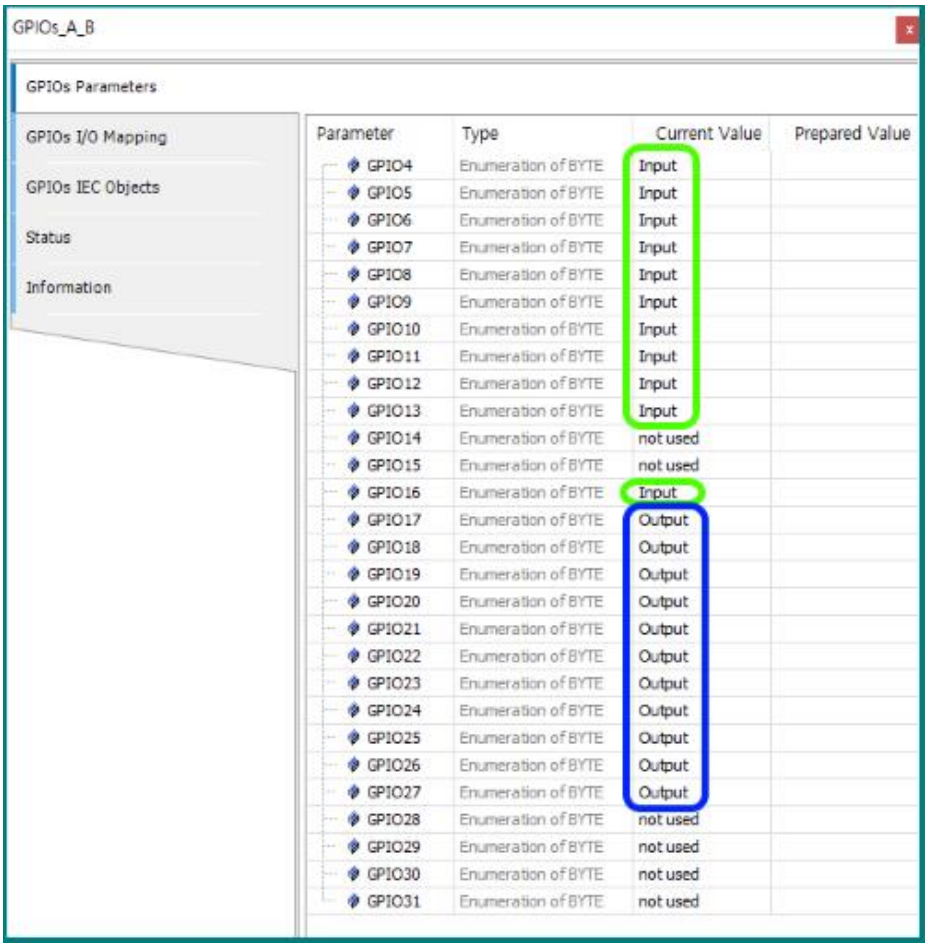
뒤에 붙이면 이런 모양이 됩니다.



자 이제 CODESYS 로 와서 왼쪽에 있는 GPIOs_A_B 를 더블클릭하면 연두색 박스로 표시해놓은 부분이 보입니다.

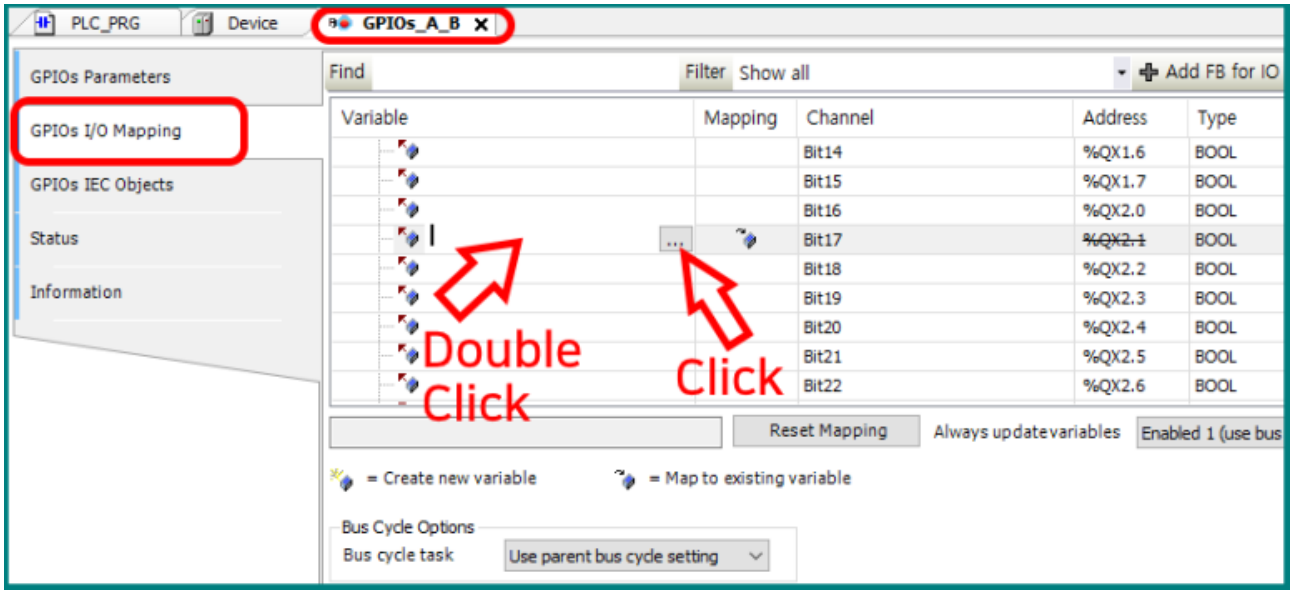


최초 상태에서는 Current Value 가 not used 로 되어 있는데, 이것을 가지고 계신 CPIO 보드 상황에 맞게 Input 또는 Output 으로 변경해주어야 합니다. 제가 지금 사용중인 CP-IO22 에 맞게 아래처럼 바꾸어 주었습니다.

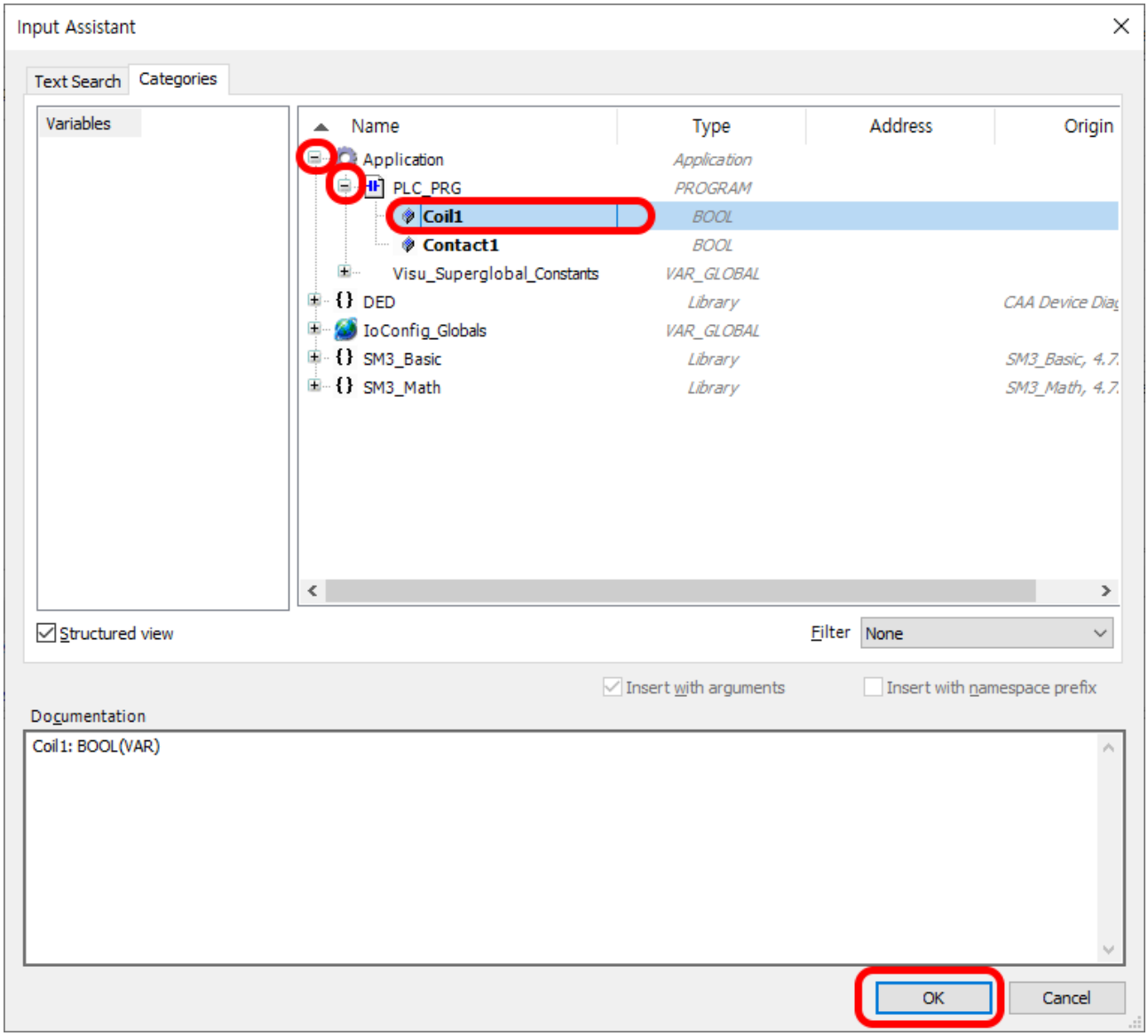


그리고 앞서 강좌에서 작성해 놓은 레더로직의 출력코일 (Coil1)을 GPIO17 번 포트와 연결해 보겠습니다.

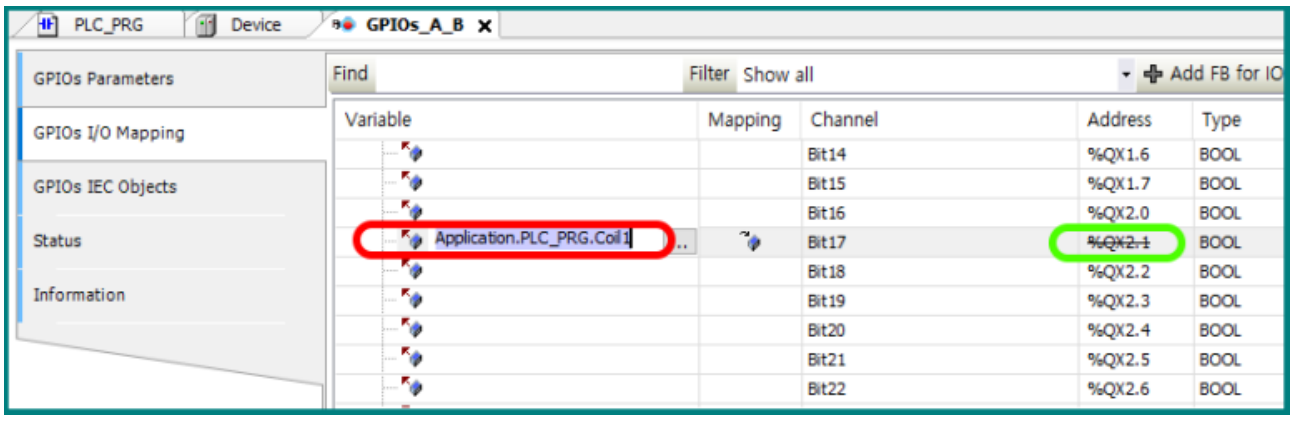
GPIOs_A_B 탭에서 GPIOs I/O Mapping 을 누르시고 ..



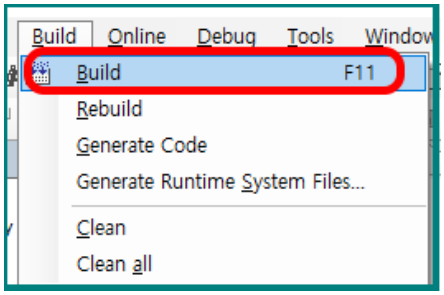
아래 창이 나오면 Application 쪽을 펼쳐서 Coil1 변수를 선택하세요.



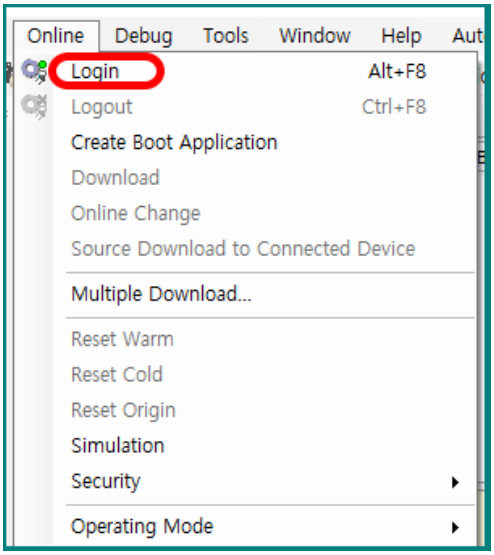
그러면 이렇게 됩니다. 연두색 박스를 보면 기존 연결되었던 변수는 해제되었군요. 원래 I/O 포트는 %QX2.1 과 같은 형태의 변수로 억세스가 가능합니다.



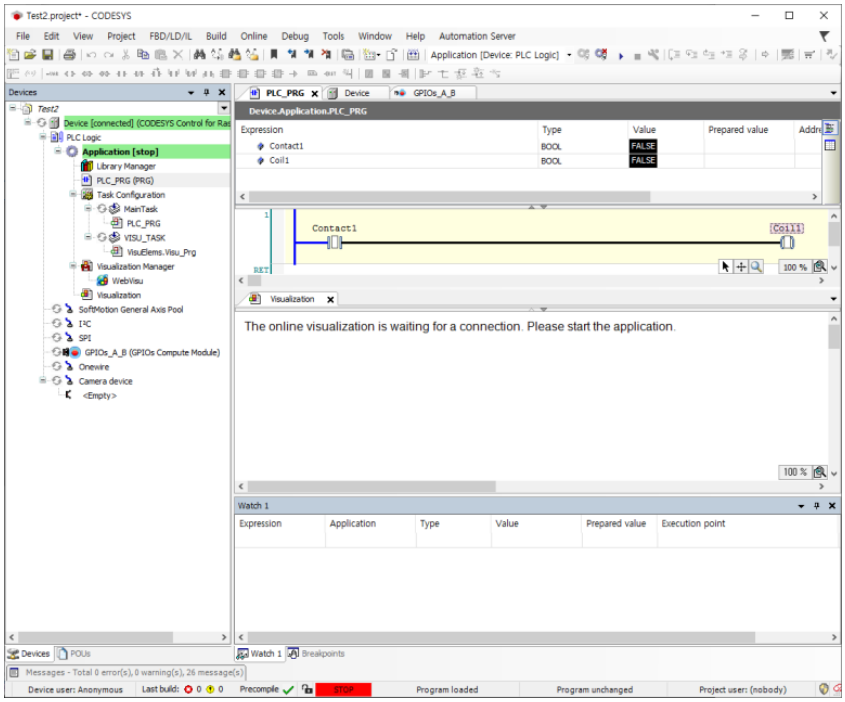
이제 F11 을 눌러서 빌드(컴파일)을 해봅시다.



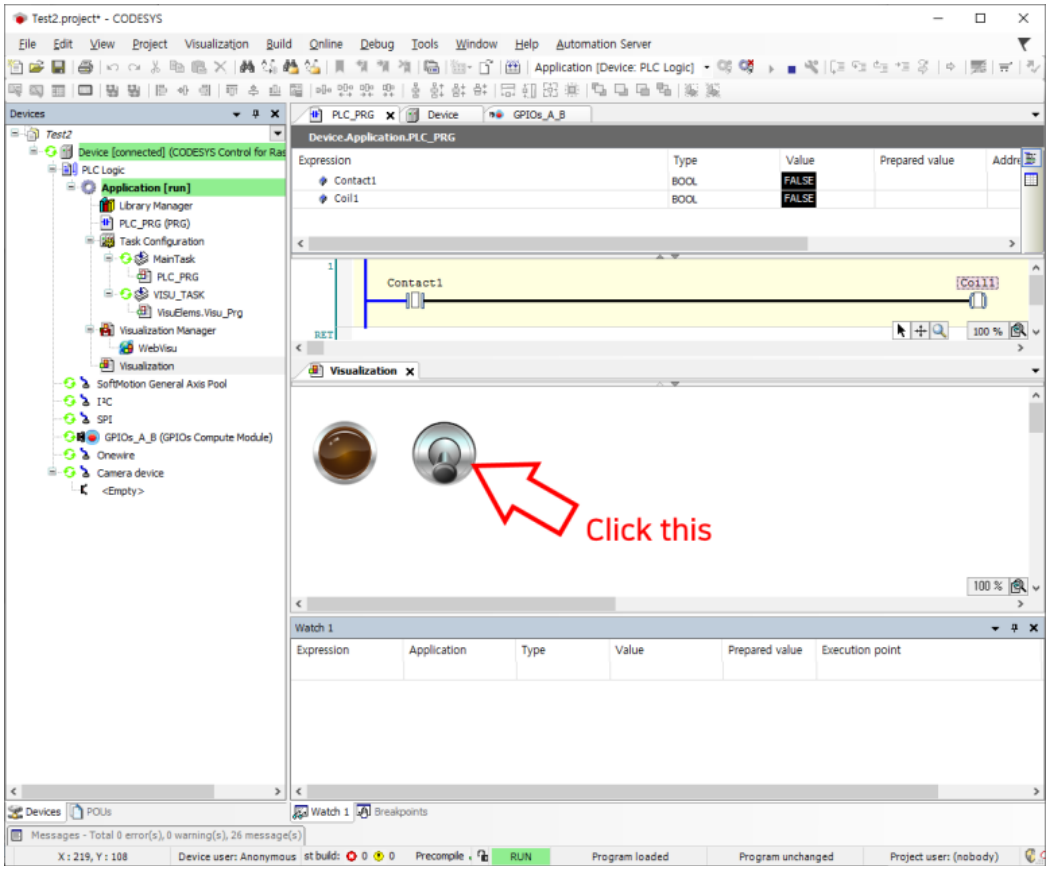
에러가 0 이면 통과된겁니다. 이제 ALT+F8 을 눌러서 로그인하세요. 이때 빌드된 결과가 컴파일 파이 쪽으로 배포(전송)됩니다.



아래 상태가 되면 준비된 것입니다. F5 를 눌러서 실행시켜보세요.



스위치를 클릭했을때, CPIO 보드의 포트 17 에서 불이 들어오면 성공한 것입니다.

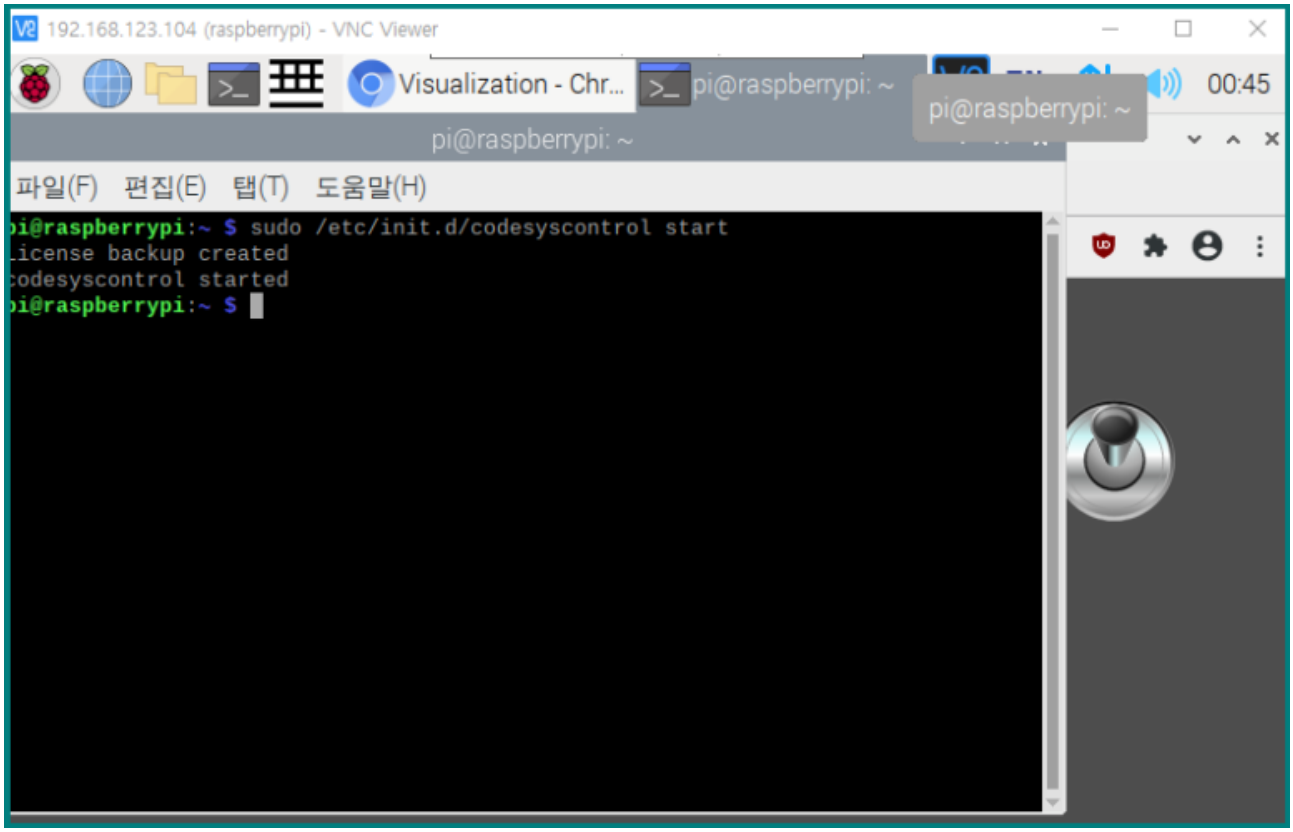


재시작 명령

런타임 라이센스를 구매하지 않아도, 실험이나 공부하는데 지장은 없습니다. 어차피 컴파일/배포를 계속 반복하니까요. 배포한뒤 2 시간이 지나면 동작을 멈춥니다. 이때 아래 명령으로 재시작 시킬 수 있습니다.

sudo /etc/init.d/codesyscontrol start

코멘드 창에 위와 같이 명령어를 입력하면 재시작합니다.

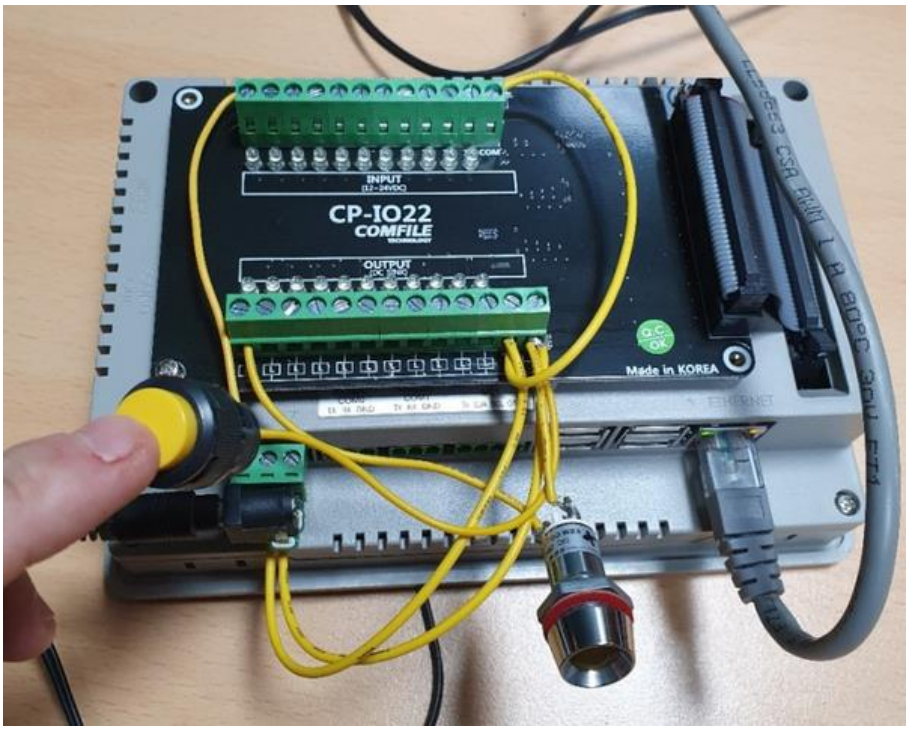


sudo /etc/init.d/codesyscontrol stop

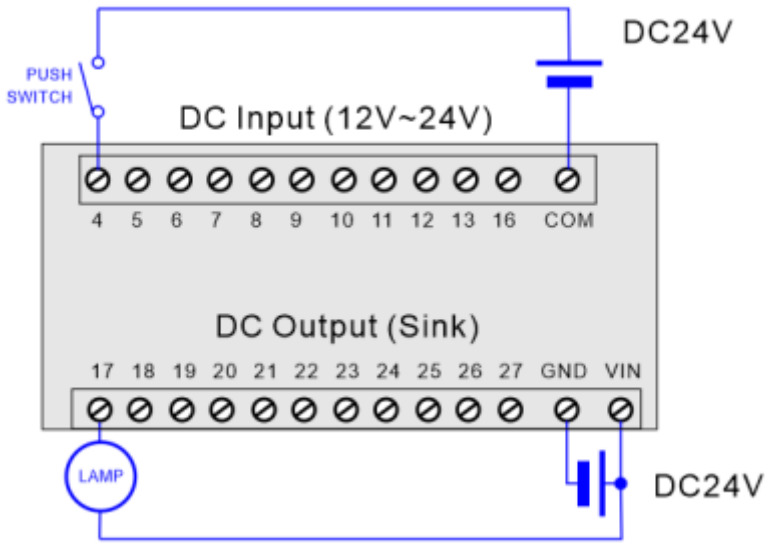
이 명령은 CODESYS 실행을 강제로 멈추게 합니다.

입력과 출력

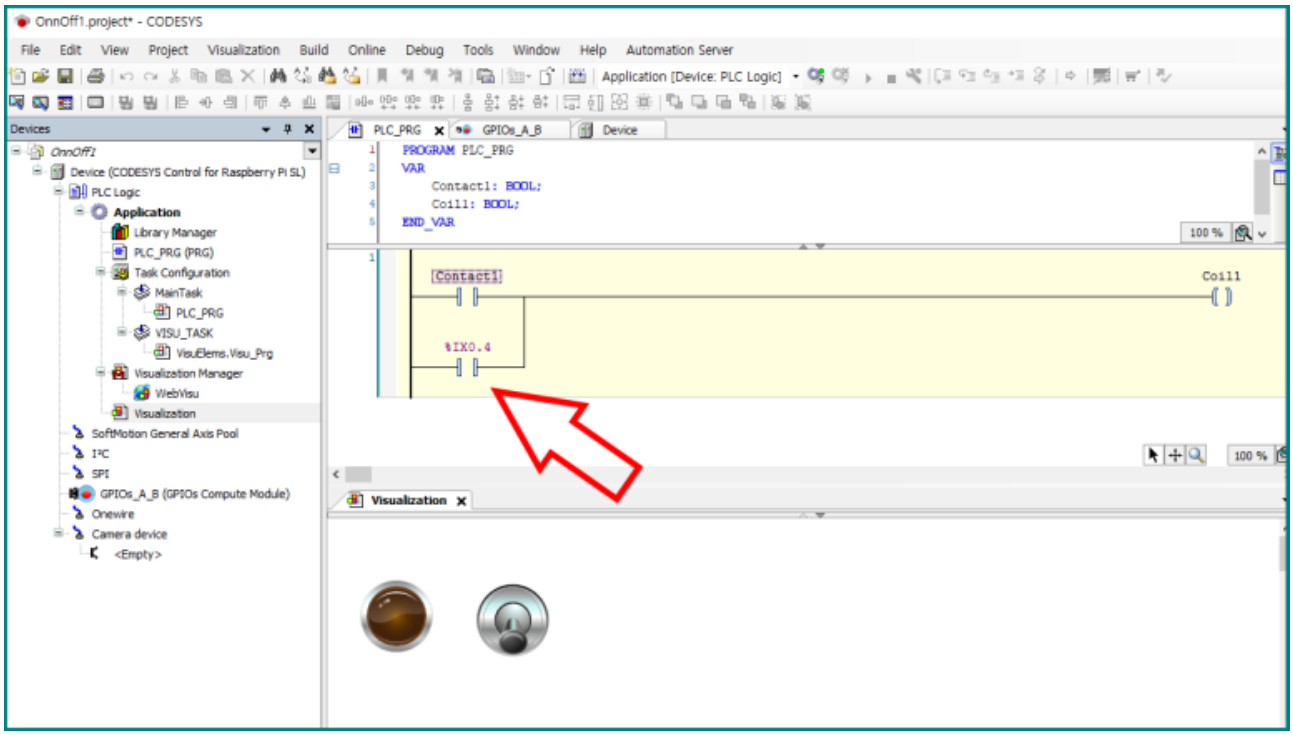
앞서 강좌에서 컴파일파이 출력을 테스트해봤는데, 이번엔 입력과 출력을 모두 테스트해 보겠습니다. 이 실험을 위해서 CP-IO 보드에 스위치와 램프를 연결했습니다.



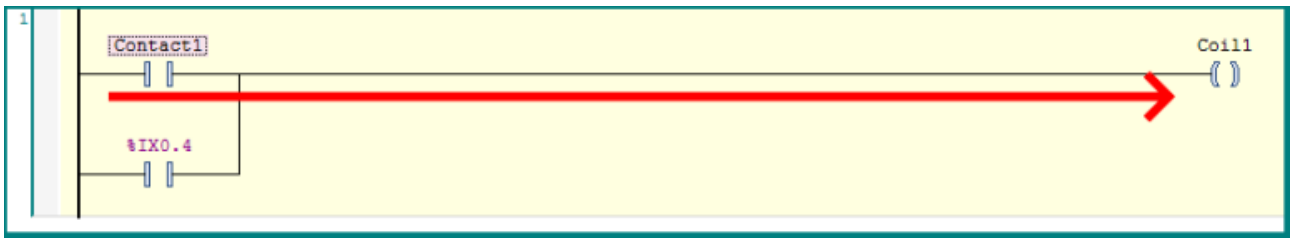
결선은 이렇게 했습니다.



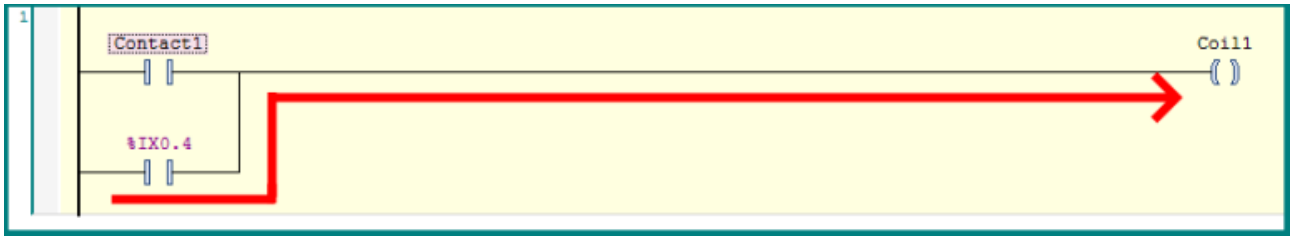
CODESYS 프로그램은 앞서 만든 것을 좀더 발전시켰습니다.



스위치를 4 번포트에 달아놔서, 그에 해당하는 어드레스인 %IX0.4 접점을 병렬로 하나 더 연결한 것입니다. 비주얼라이제이션 에서 토글스위치를 눌러도 켜지고

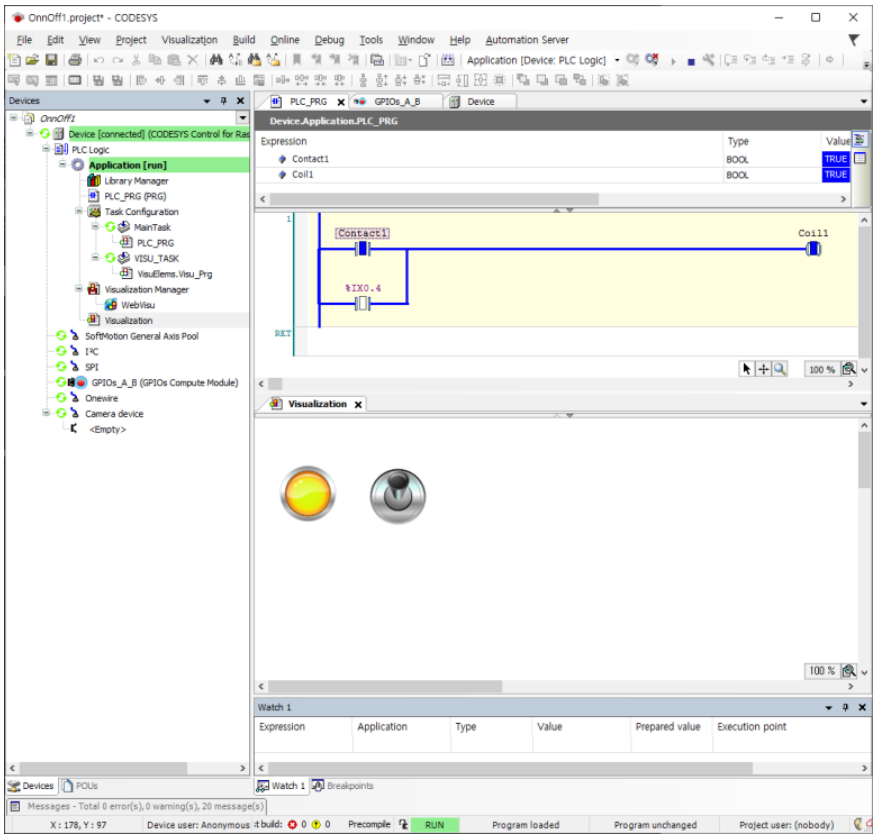


직접 손으로 스위치를 눌러도 켜지게 해놓은 것입니다.

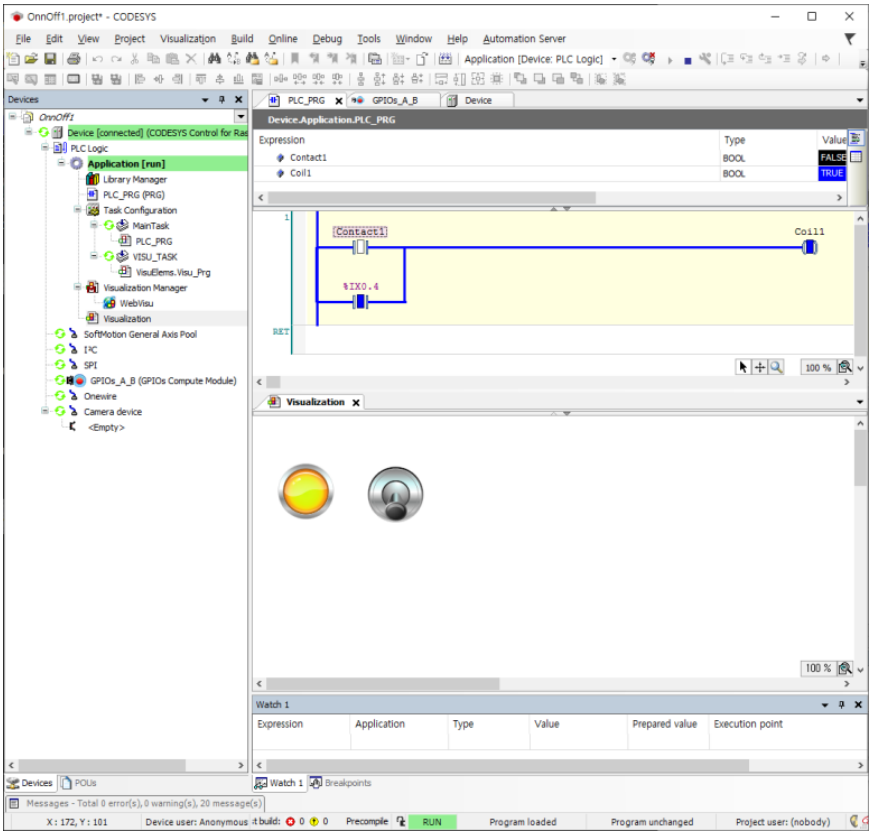


빌드한 뒤 실행시켜 보았습니다.

비주얼라이제이션에서 토글스위치 ON !!!



잘되는거 확인했고, 이번엔 직접 스위치를 눌러보니까...

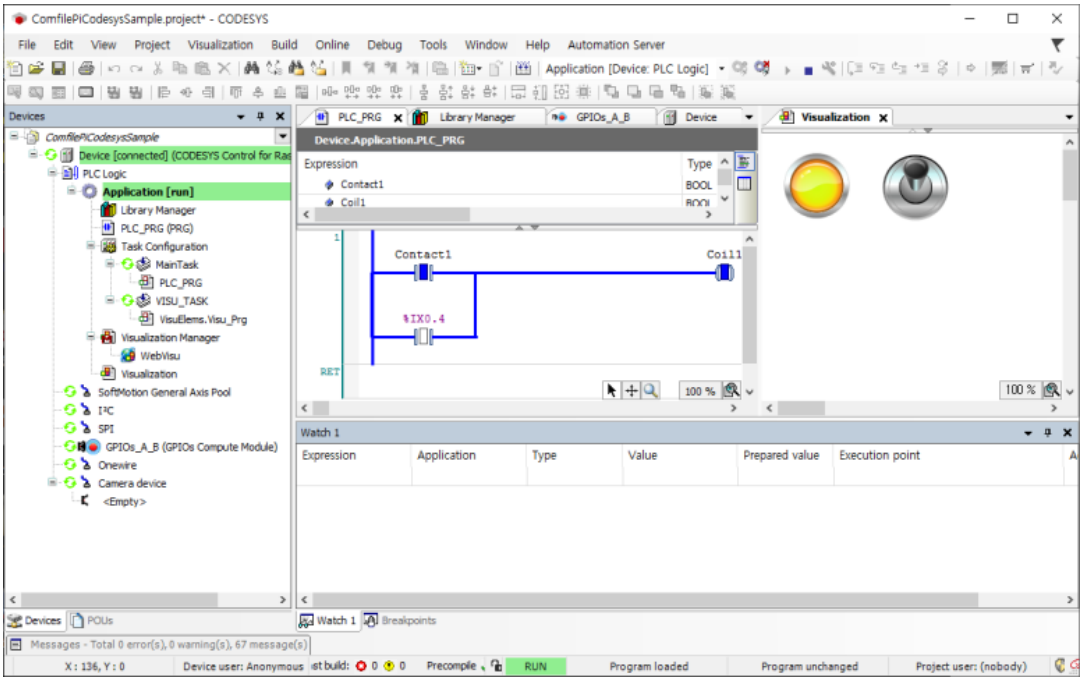


역시 잘됩니다. 동영상으로도 감상해보세요. https://youtu.be/yrU5yqCgQVQ

WebVISU

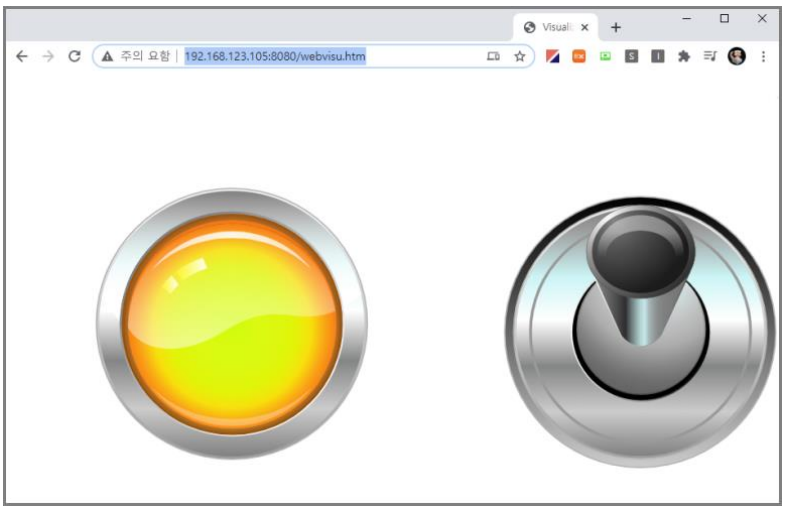
ComfilePi (라즈베리 파이)용 CODESYS 는 시각화(Visualization)기능을 웹에서 접근할 수 있도록 해주는 WebVisu 기능도 포함되어 있습니다. @.@!! 스마트폰에서도 제어가능합니다. (PC 용 CODESYS 는 별도로 되어있어서 비용을 추가로 내야하는데 말이죠.!!)

어떤건지 한번 보시죠. 일단 실행중인 화면입니다.



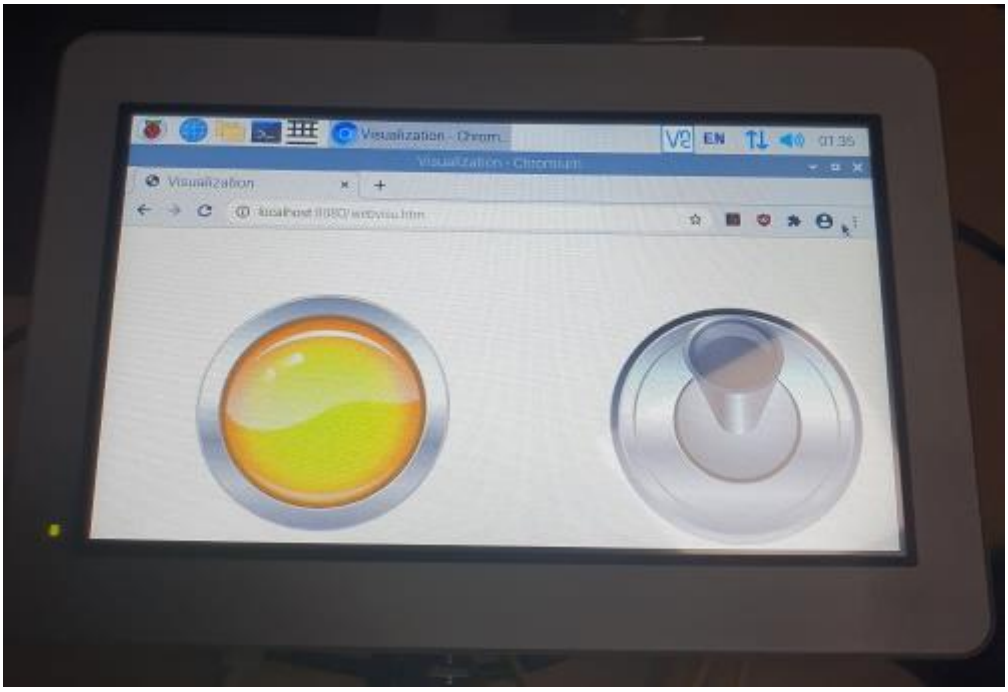
개발하고 있는 PC 에서 웹브라우저 (크롬 추천)을 실행시킨뒤, 주소창에 컴파일파이의 IP 어드레스를 입력해주세요. 포트번호와 접근파일까지 입력해주면 아래처럼 짷 하고 뜹니다. http://192.168.123.105:8080

위에 IP 주소 그대로 치시는 분은 없겠죠? 여러분의 네트워크에 있는 컴파일파이(라즈베리파이)주소를 찾아서 입력해야 합니다. 위 주소는 제 네트워크의 상황입니다.

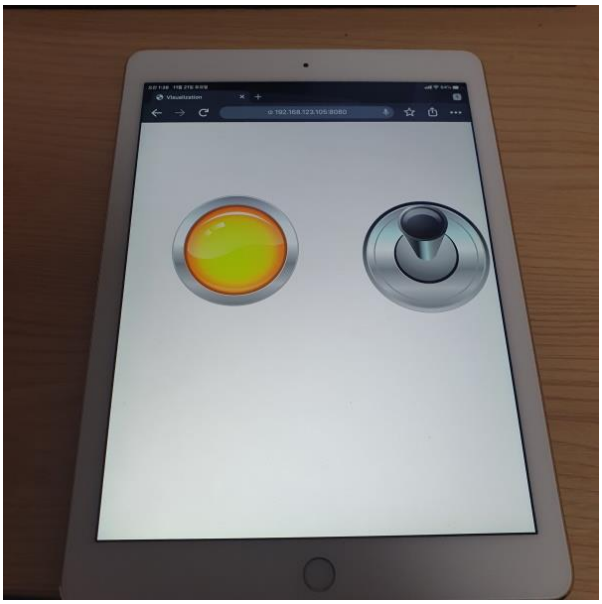


컴파일파이에 있는 웹브라우저에서도 잘 됩니다. 주소는 이렇게 입력하세요.

localhost:8080



태블릿(스마트폰)에서도 잘됩니다. 주소는 PC 와 똑같이 입력했습니다.
http://192.168.123.105:8080



https://youtu.be/OtcE5q0OpEA

WebVisu 기능이 참 쓸만합니다. 4 차산업혁명 설명에 주로 등장하는 IOT, 스마트팩토리 / 스마트팜이 결국은 멀리서 공장이나 농장을 스마트폰으로 제어하거나 감시할 수 있는 시스템인데, 이 WebVisu 기능을 활용하면, 특별한 코딩없이도 이런 것들을 구현할 수 있습니다.

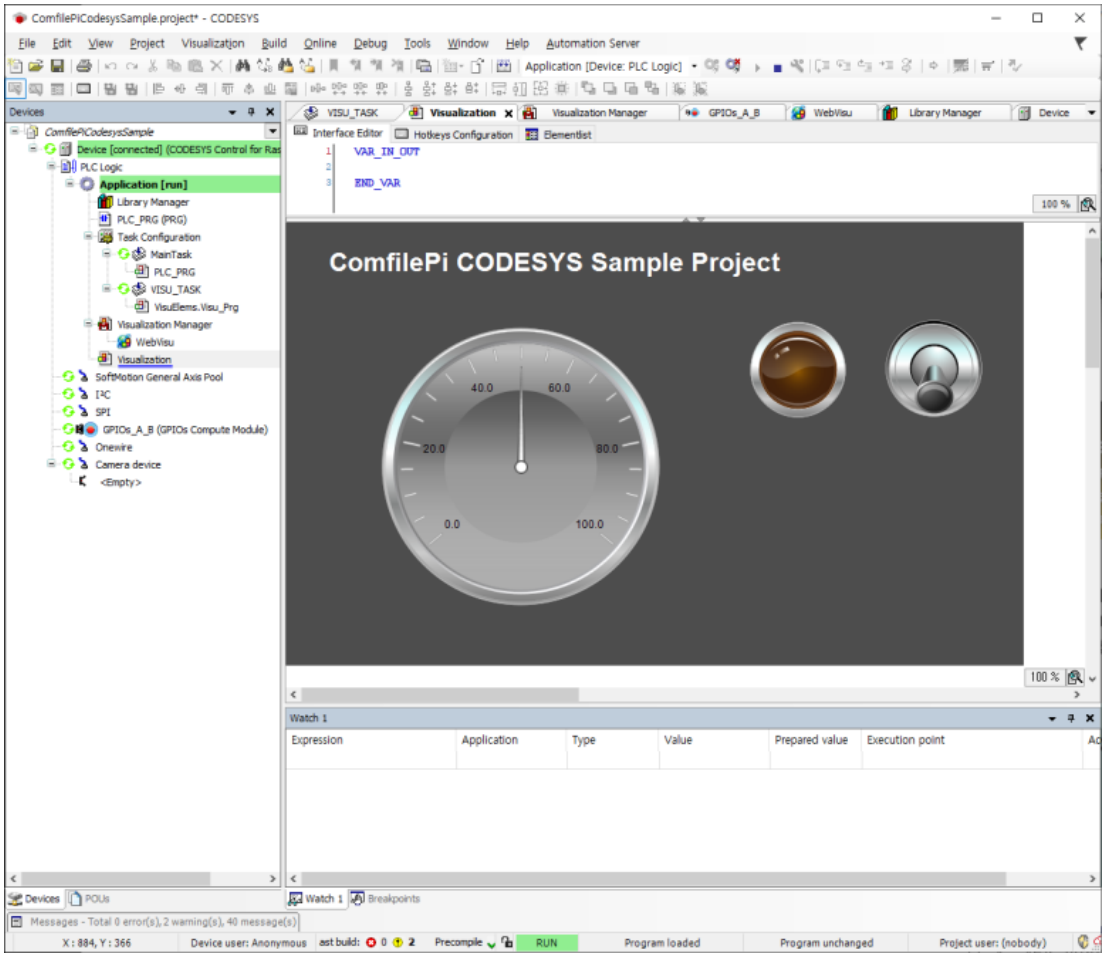
HMI로 활용

자 앞서 강좌에서 컴파일파이에 실행중인 최종파일은 컴파일파이를 껐다가 켜도 자동실행되도록 하는 방법을 설명드렸구요. 이 상태에서 컴파일파이의 웹브라우저를 이용해서 스스로 WebVisu 화면을 볼 수 있다는 것도 설명드렸습니다.

아 그러면 잠깐만 !

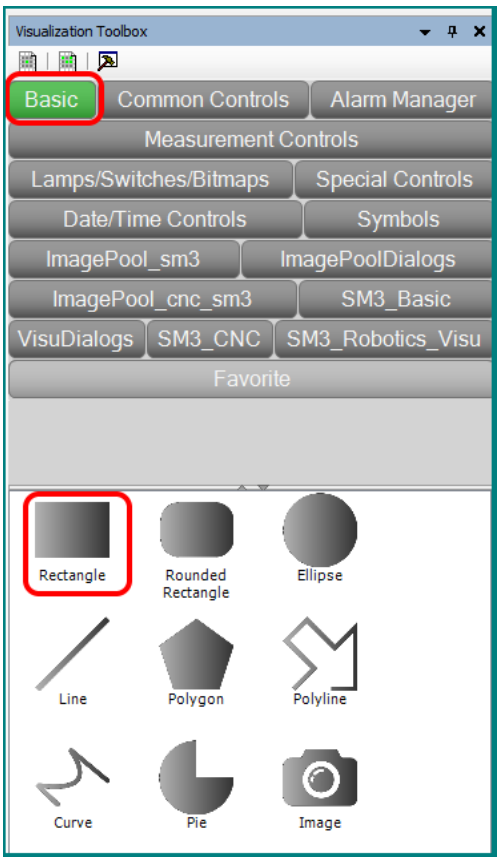
그럼 컴파일파이가 켜질때, 자동으로 웹브라우저를 띄우고 WebVisu 를 볼 수 있도록 하면, 어떻게 되는건가 @.@ HMI 처럼 쓸 수 있게 되는 거 되는거 아닌가?

그래서 한번 해봤습니다. 화면을 이렇게 구성해보았습니다.

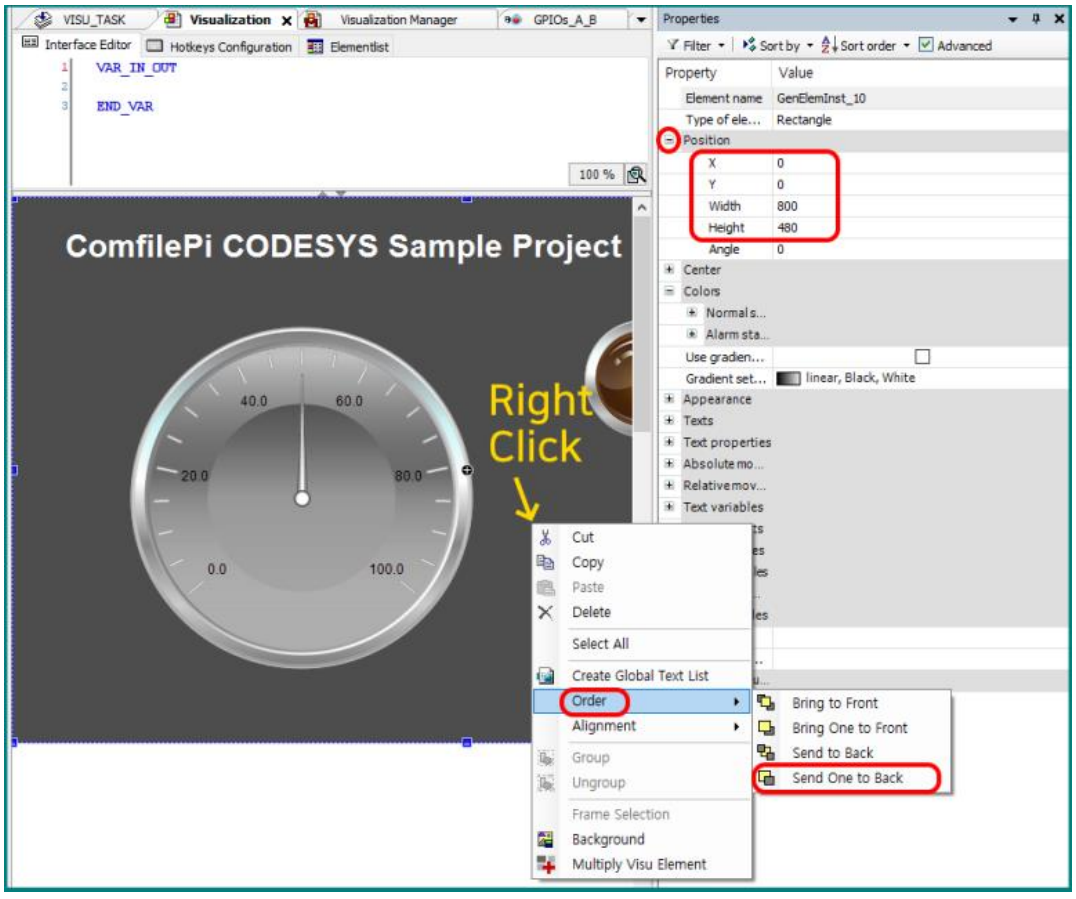


바탕에 짙은회색으로 칠해진 박스를 깔았습니다. 웹브라우저가 이미지만 딱 맞춰서 보여주기때문에, 바탕에 컴파일파이 7 인치 제품화면과 동일한 해상도인 800 x 480 배경을 깔아둔 것입니다.

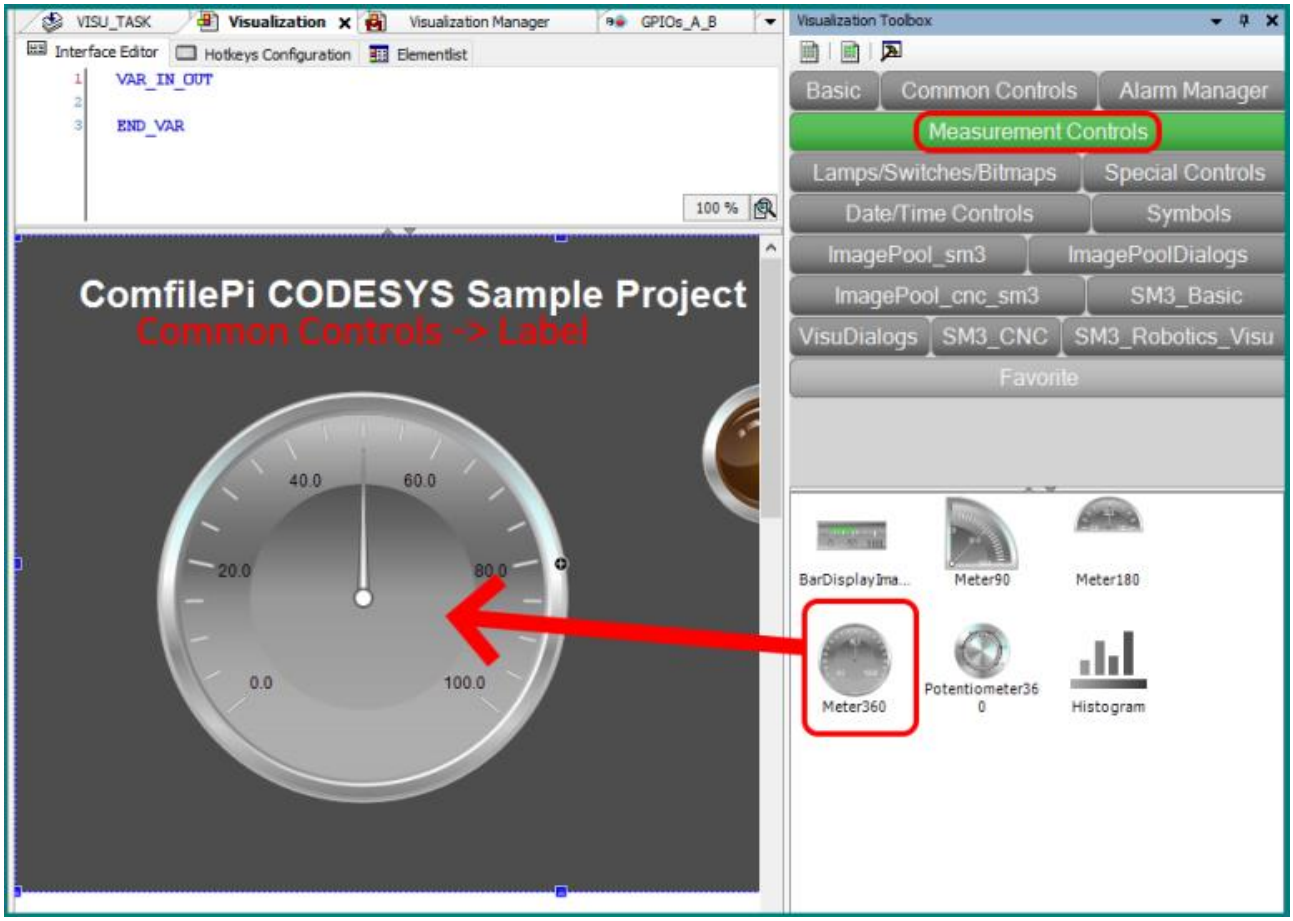
Visualization Toolbox 에서 Basic 탭을 선택한뒤 밑에서 Rectangle(사각형)을 골랐습니다.



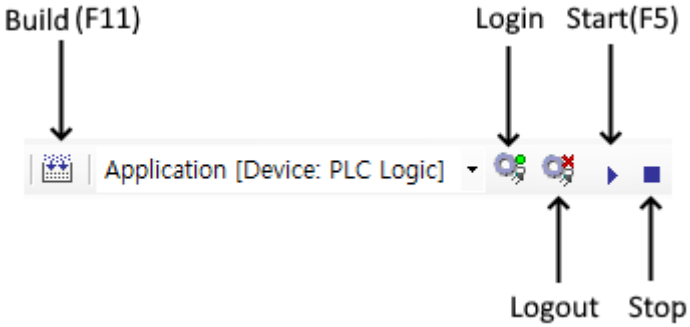
그리고 Position 을 펼쳐서 X,Y 를 0 으로, 넓이와 높이는 800 과 480 으로 조정했습니다. 사각형의 표시순서는 가장아래 (Send One to Back)으로 바꾸었습니다.



그리고 좀 모양을 내기 위해서 텍스트와 게이지 하나를 큼지막하게 표시했습니다. 아무 동작도 안하는 말그대로 멋좀 낸거에 불과합니다.

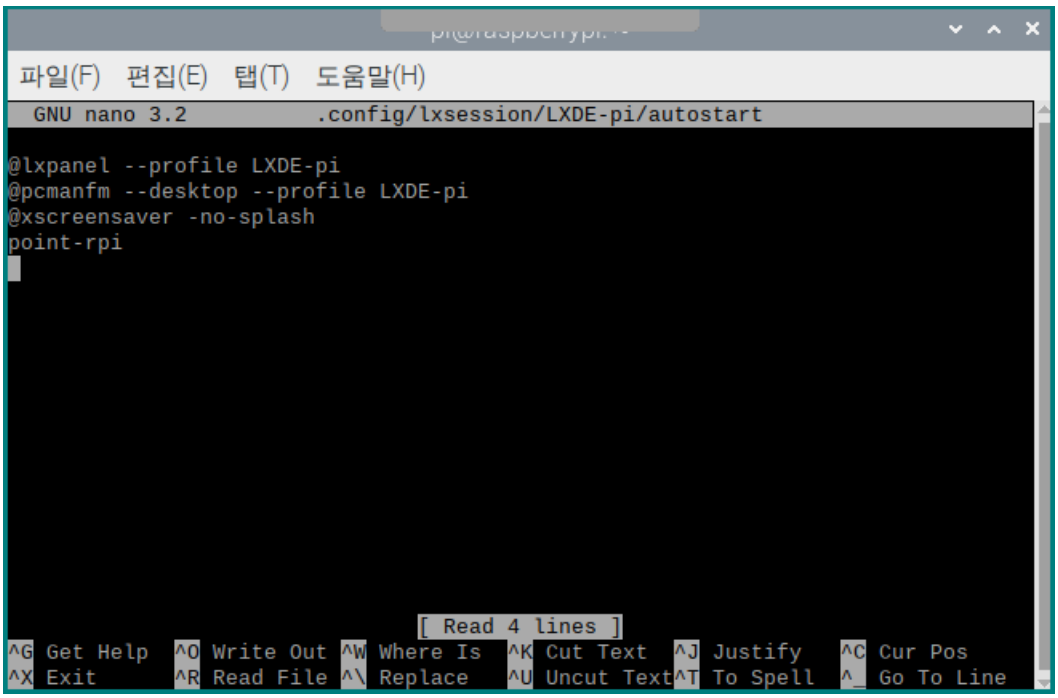


수정이 끝났으면 컴파일파이에 다운로드 해보겠습니다. 앞서 설명드렸듯이, Build --> Login --> Start 순서대로 실행해주어야 합니다. 상단의 아이콘을 차례대로 클릭하면 편합니다.

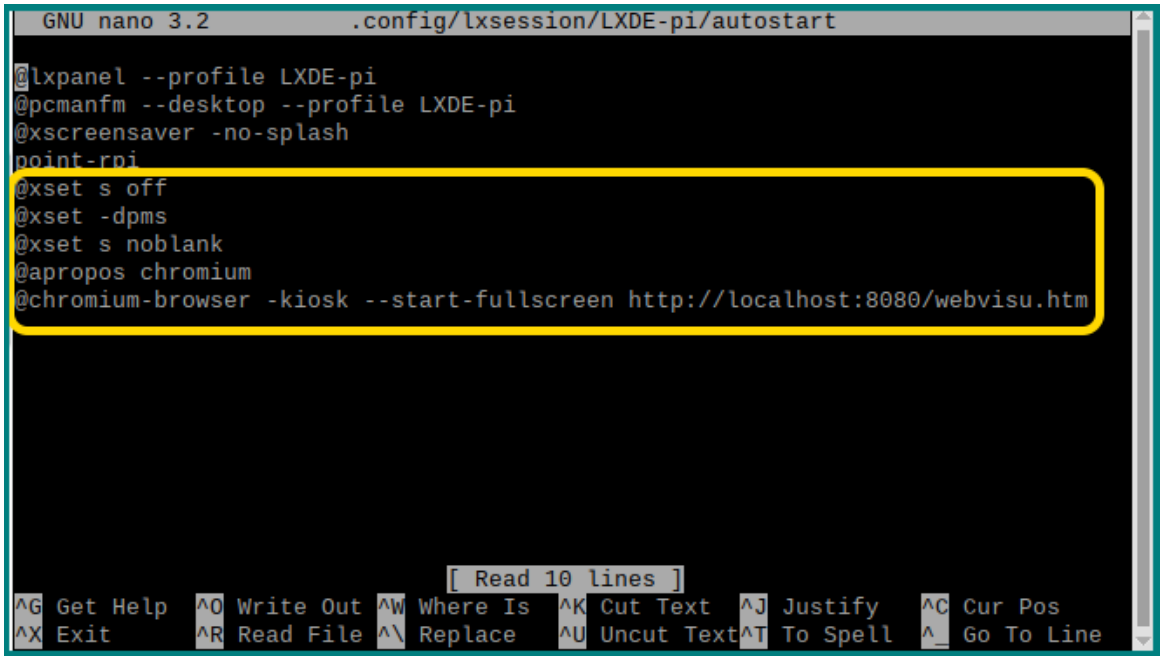


이제 컴파일파이(라즈베리파이)에서 자동실행되는 부분을 조정해주어야 합니다.

코멘드 창에서 sudo nano .config/lxsession/LXDE-pi/autostart 라고 입력해주면, 에디터와 함께 해당파일이 뜹니다.



이 파일이 자동시작을 관리해주는 파일인데, 이 밑에 아래처럼 명령 몇 줄을 더 입력합니다. 브라우저를 최대화면으로 실행시키고 WebVisu 를 보여주라는 뜻입니다. 입력이 끝나고 저장할때는 ctrl+X 누르고 Y 누르면 됩니다.



@xset s off
@xset -dpms
@xset s noblank
@apropos chromium
@chromium-browser -kiosk --start-fullscreen http://localhost:8080/webvisu.htm

어떻게 되는지는 아래 동영상으로 확인해 보시죠. 전원이 켜지면 WebVisu 화면으로 넘어갑니다. 컴파일파이를 HMI 로도 쓸 수 있게 되었습니다. https://youtu.be/eGe-jU_aY2A

커서감추기

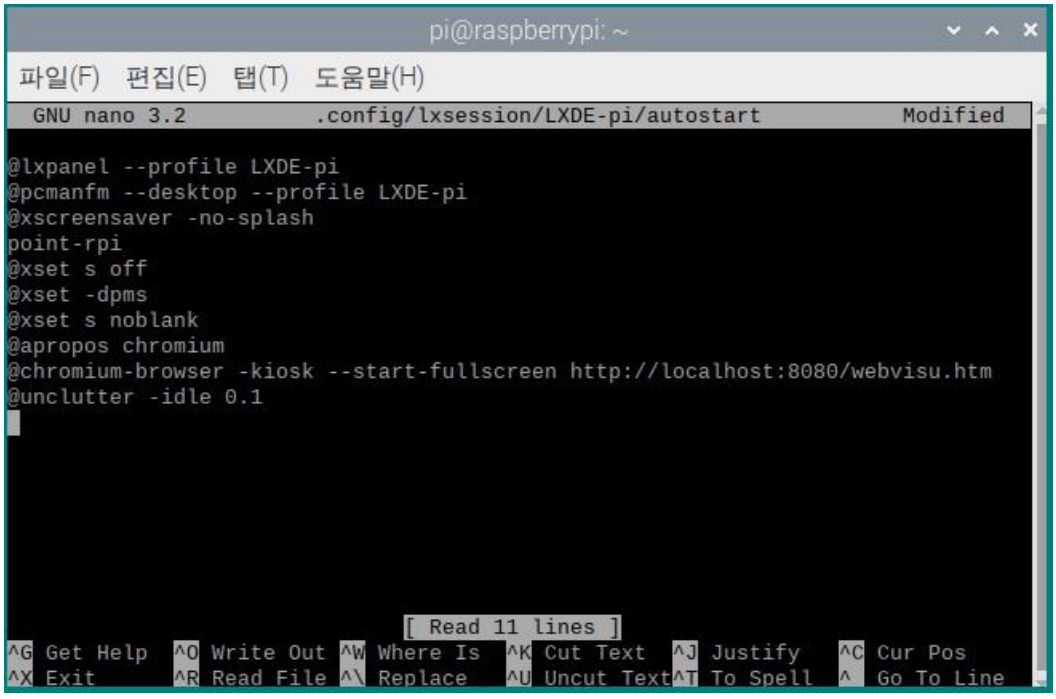
컴파일파이(라즈베리파이)시작시 자동으로 웹브라우저를 풀스크린으로 시작하게 하는 것은 성공했는데, 커서가 눈에 좀 거슬리더군요.

일단 커멘드 창에서 아래 명령어를 쳐서 unclutter 라는 프로그램을 설치해주세요.

sudo apt-get install unclutter

그리고 바로앞 강좌에서 설명드렸던 자동시작설정 파일에 다음 문장을 삽입해줍니다.

@unclutter -idel 0.1



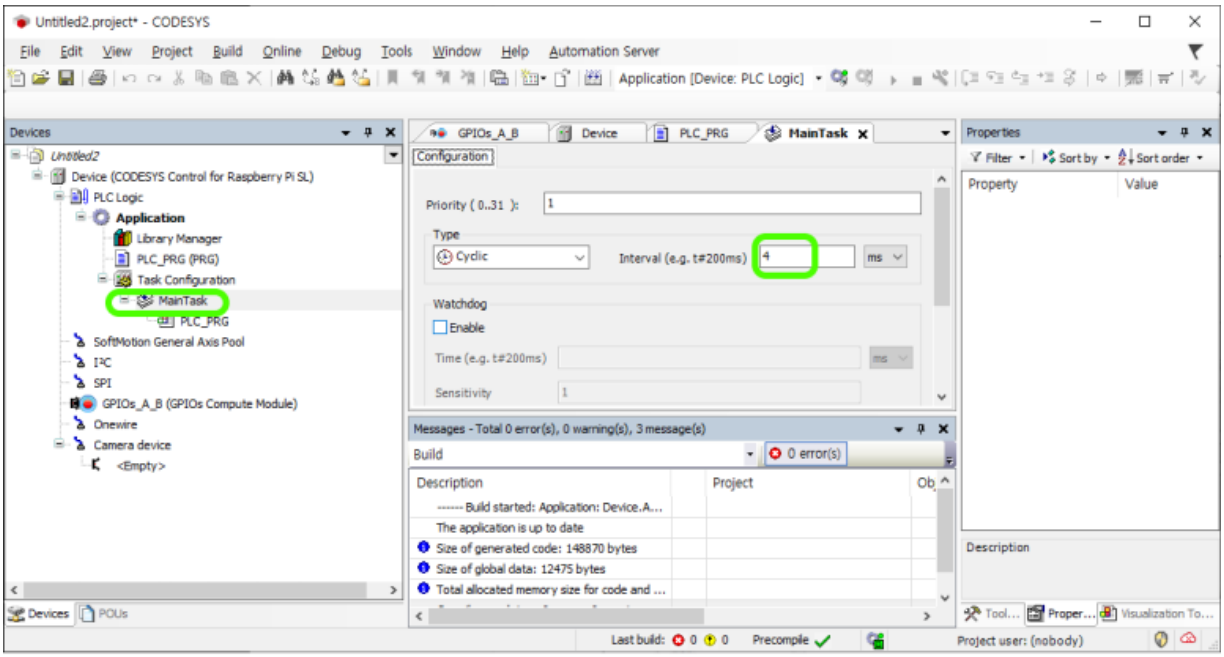
이렇게 하면 평상시 커서는 표시되지 않습니다. 화면을 터치하거나, 마우스를 움직일때에는 표시되기 때문에, 컴파일파이(라즈베리파이)조작에는 문제가 없습니다. https://youtu.be/54TKCJ47Fm8

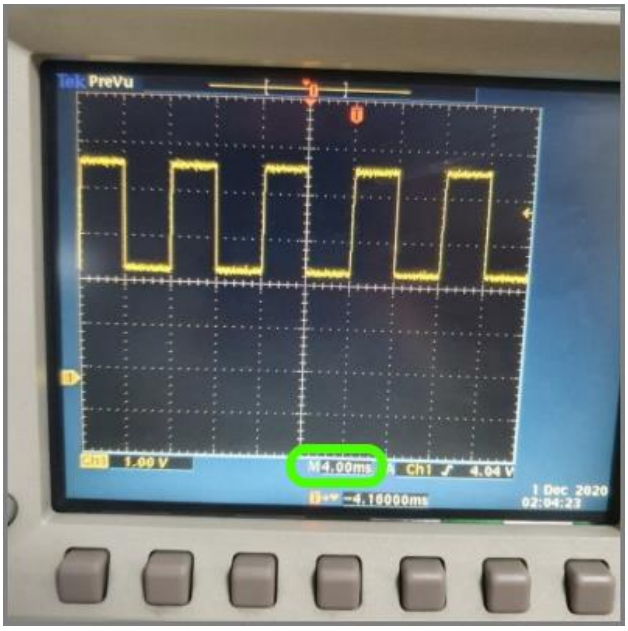
퍼포먼스 확인

CODESYS + 컴파일파이 조합으로 퍼포먼스가 어떤지 한번 실험해 보았습니다. 컴파일파이 (CPi-A 또는 B070WR)의 메인칩 사양은 이렇습니다.

- 1.2GHz 64 비트 쿼드코어 ARM Cortex-A53

CODESYS 왼쪽창에서 Main Task 를 더블클릭하면 스캔타임 주기를 설정할 수 있는 창이 나옵니다. 디폴트 값으로는 4mS 가 들어 있습니다.





정확히 4mS 간격으로 스캔이 되고 있습니다.

측정은 이 프로그램으로 했습니다. 출력포트중 하나를 스캔마다 토글시키는 ST 프로그램입니다.

```
PROGRAM PLC_PRG
VAR
    Avalue : DWORD ;
END_VAR
```

```
Avalue := Avalue + 16#20000;
%QD0 := Avalue;
```

이번에는 컴파일파이에 동영상을 틀어 놓고 측정해봤습니다. 앞에서는 동영상이 플레이되고, 뒤에서는 CODESYS 런타임이 돌고 있는 상황입니다. 어떤지 동영상으로 함 보시죠.

https://youtu.be/NH7L1iRRu7k

보시는 것처럼 전반적으로 무리없이 스캔을 잘 돌고 있었습니다. 생각보다 라즈베리파이 (컴파일파이)에서 CODESYS 가 잘 수행되는 것을 확인했습니다.

프로그램이 복잡해지면 두가지 선택지가 있습니다.

1. 런타임을 한단계 높은 MC (멀티코어)로 선택한다. 그럼 좀 복잡한 테스크만 다른 코어에서 실행시킬 수 있음.
2. 스캔타임 간격을 더 벌린다.

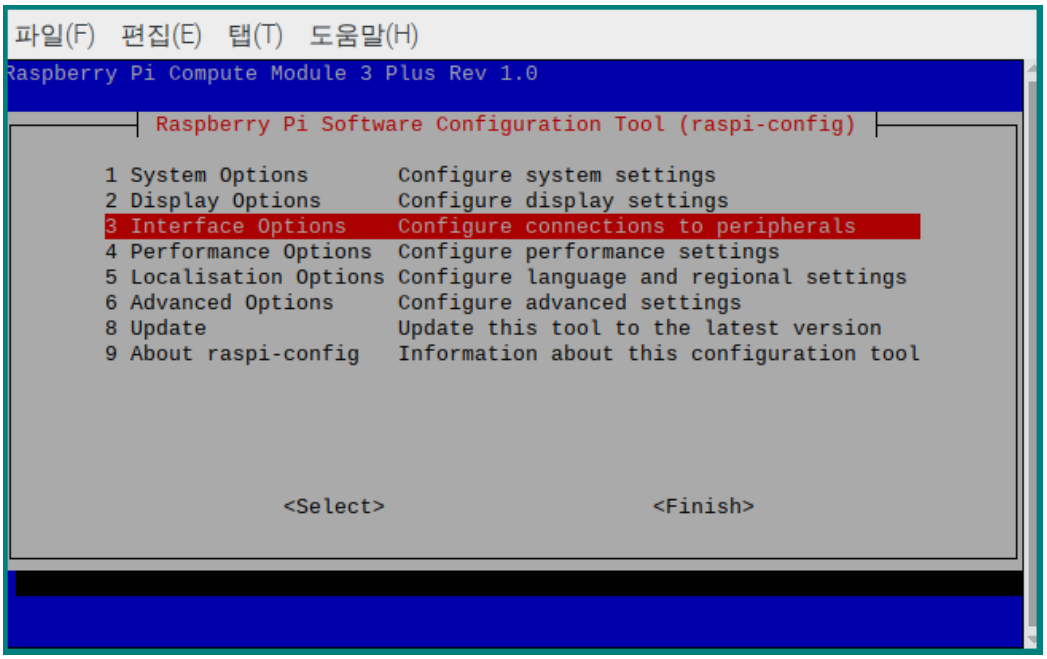
싱글코어 런타임을 쓰더라도 스캔타임을 20mS 정도로 하면, 왠만한 프로그램은 무리없이 돌아갈 것 같습니다. 실제로 PC 용 CODESYS 의 MainTask 디폴트 주기는 20mS 로 되어 있습니다.

제 4 장. 모드버스RTU

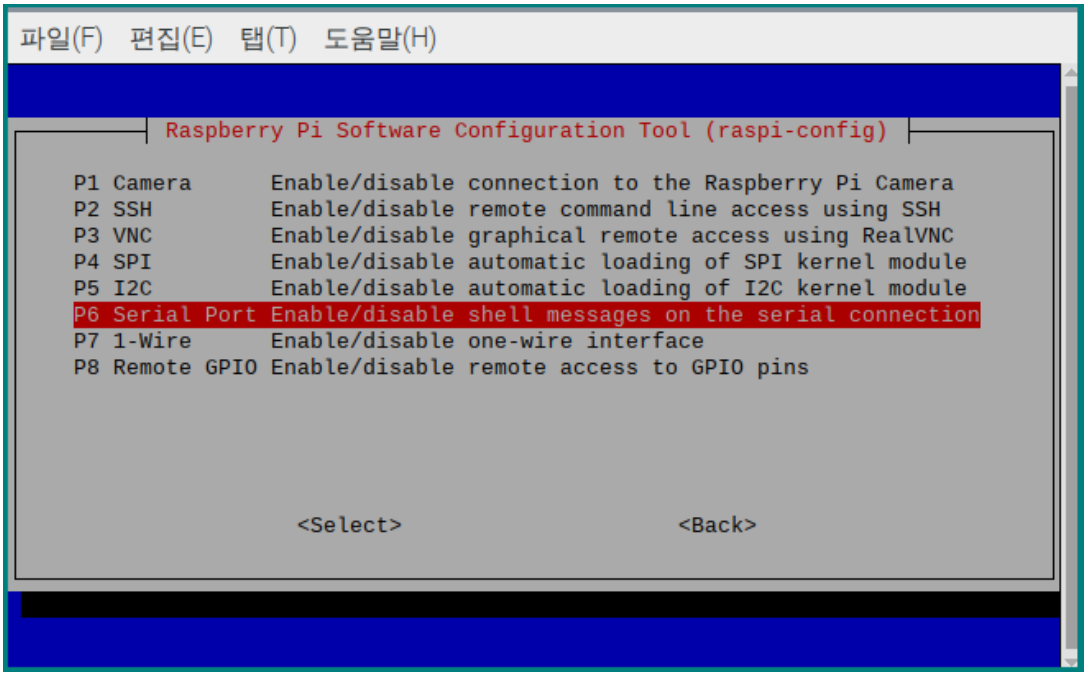
시리얼포트 사전준비작업

CODESYS 에는 기본적으로 시리얼 기능을 지원하며 MODBUS RTU 통신이 가능합니다. 컴파일파이(라즈베리파이)에서는 약간의 사전작업을 해주어야 합니다.

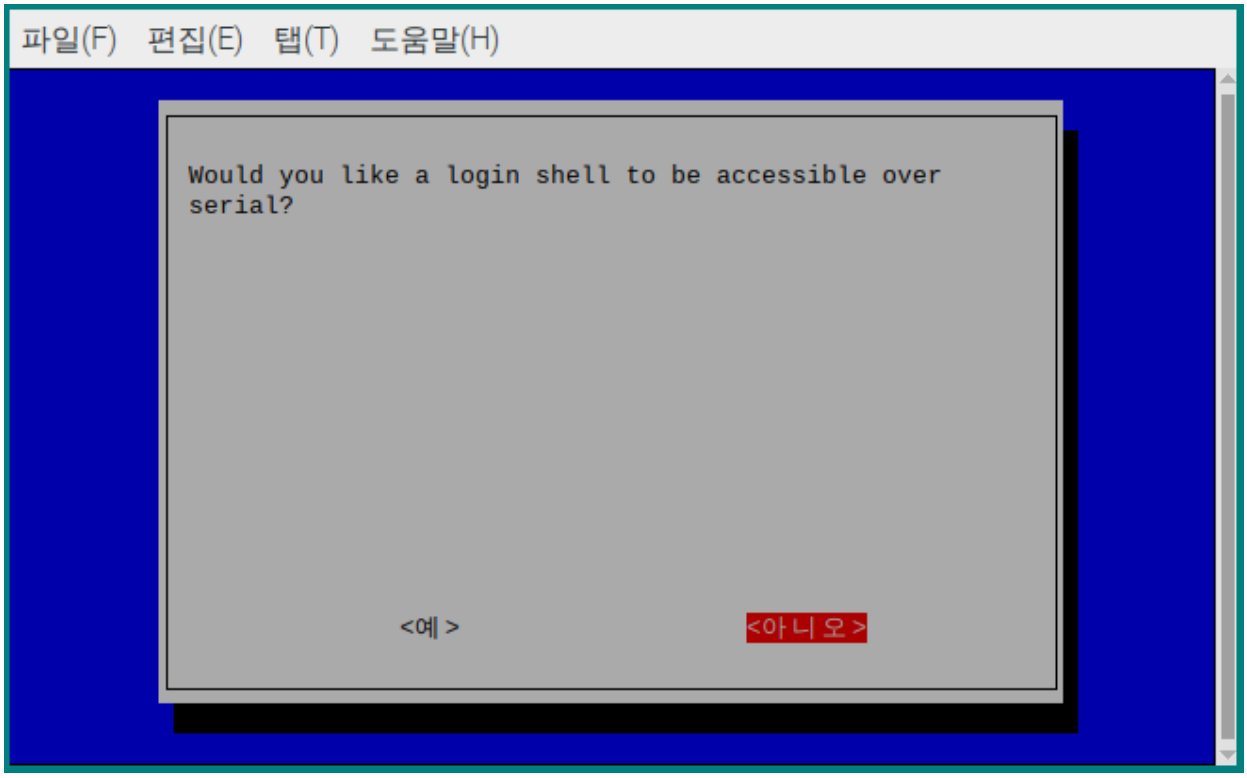
1. 우선 컴파일파이(라즈베리파이)의 Serial 기능을 켜주어야 합니다. 터미널창에서 sudo raspi-config 입력 후 3 번 Interface Options 을 선택해주세요.



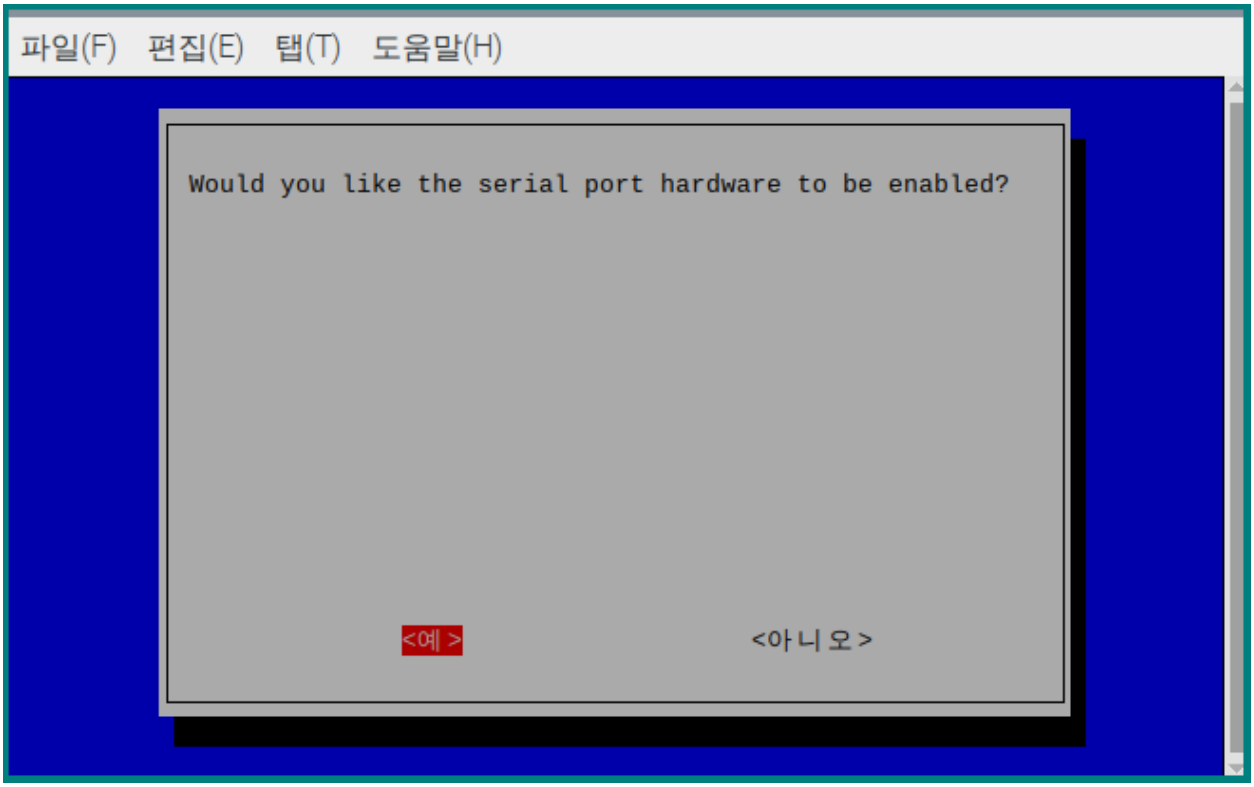
2. Serial Port 를 켜주는 옵션을 선택해주세요.



3. 이건 꺼주세요. 쉘 로그인을 시리얼 포트에서 지원하는 건데, 오히려 방해됩니다.

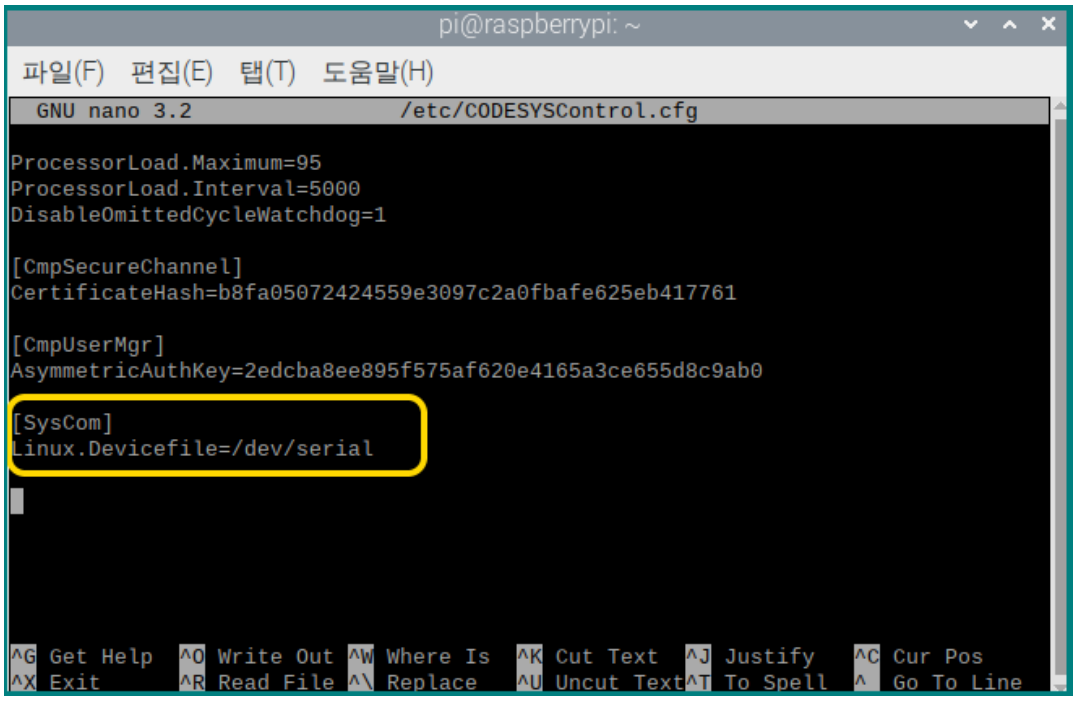


4. (하드웨어 시리얼포트) 이것만 켜주면 됩니다.



코멘드 창에서 sudo nano /etc/CODESYSControl.cfg 를 입력해주세요.

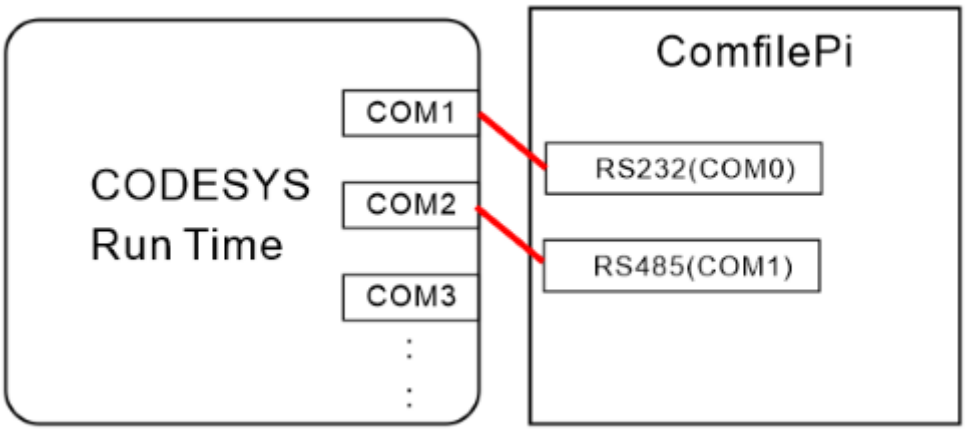
이런 파일이 뜨는데, 맨 뒤에 아래 내용을 추가해주세요.



[SysCom]
Linux.Devicefile=/dev/serial

추가하시고, 저장은 ctrl+X 그리고 Y 치면 됩니다.

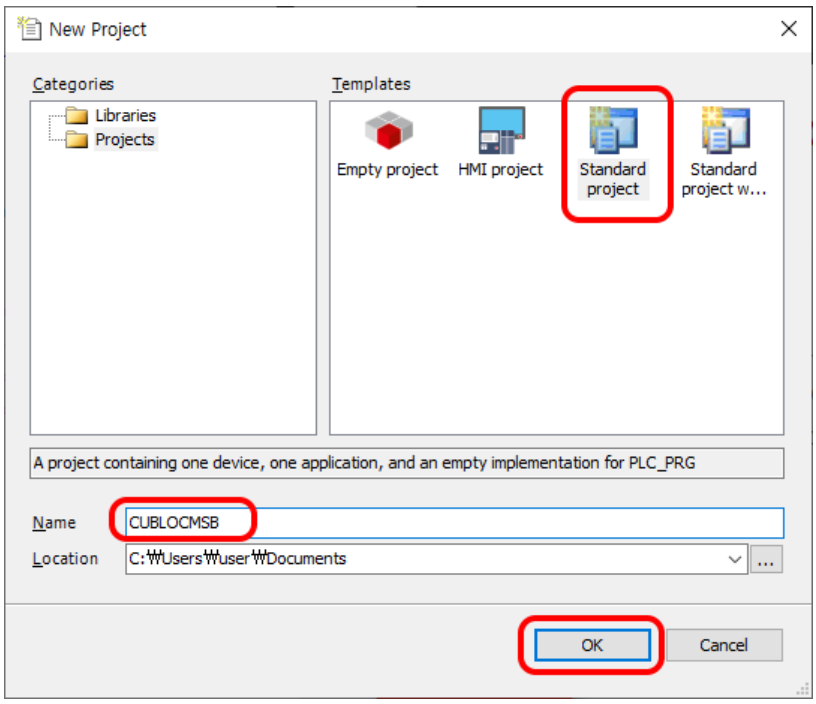
그러면 컴파일파이의 실제 포트와 CODESYS 런타임이 이렇게 연결됩니다.



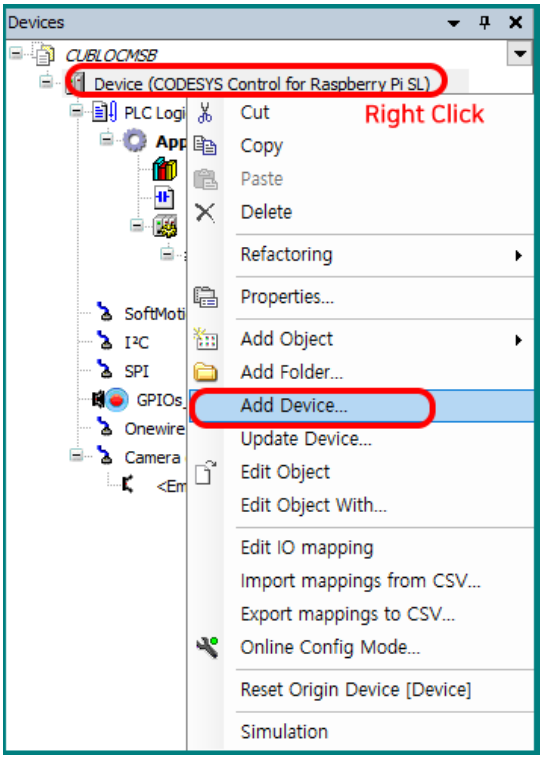
컴파일파이 뒷면에 적힌 COM 번호와 CODESYS 의 COM 번호가 다릅니다. CODESYS 에서는 COM1 부터 시작합니다. (주의 요망)

MODBUS RTU

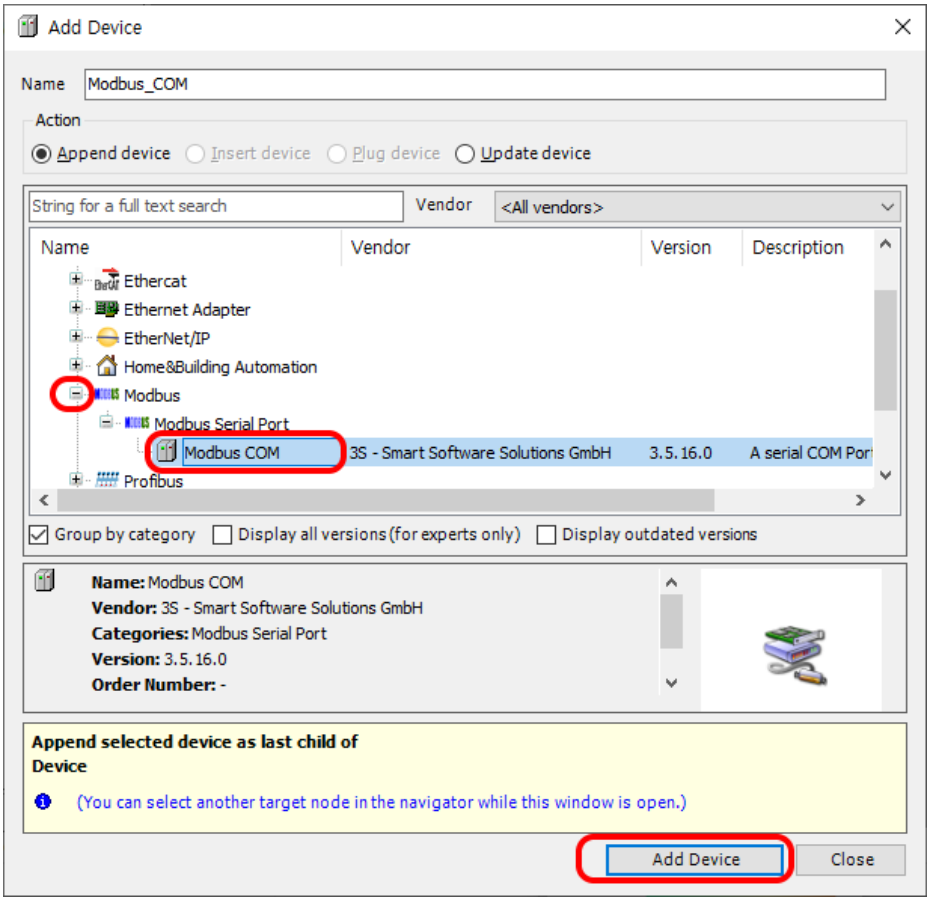
컴파일 파이 와 CODESYS 를 사용해서 CUBLOC MSB 와 연결해서 MODBUS RTU 통신을 해보겠습니다. 우선 CODESYS 를 열고 새 프로젝트를 시작합니다.



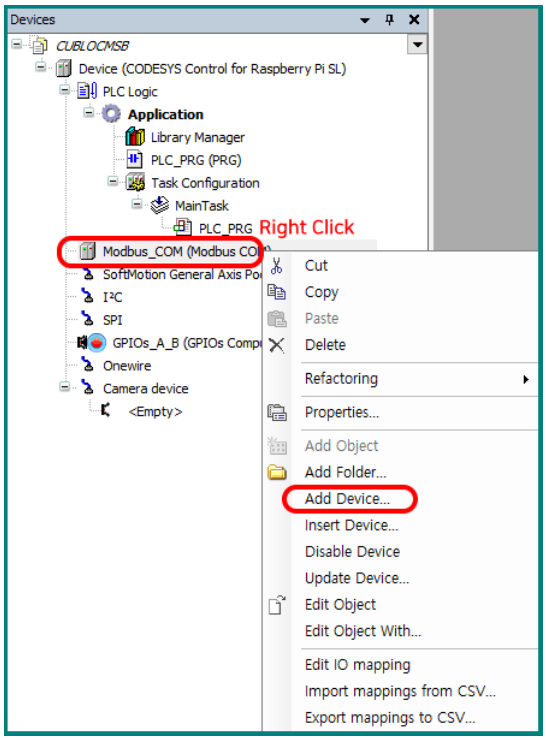
Device 에서 우클릭하고 Add Device 선택



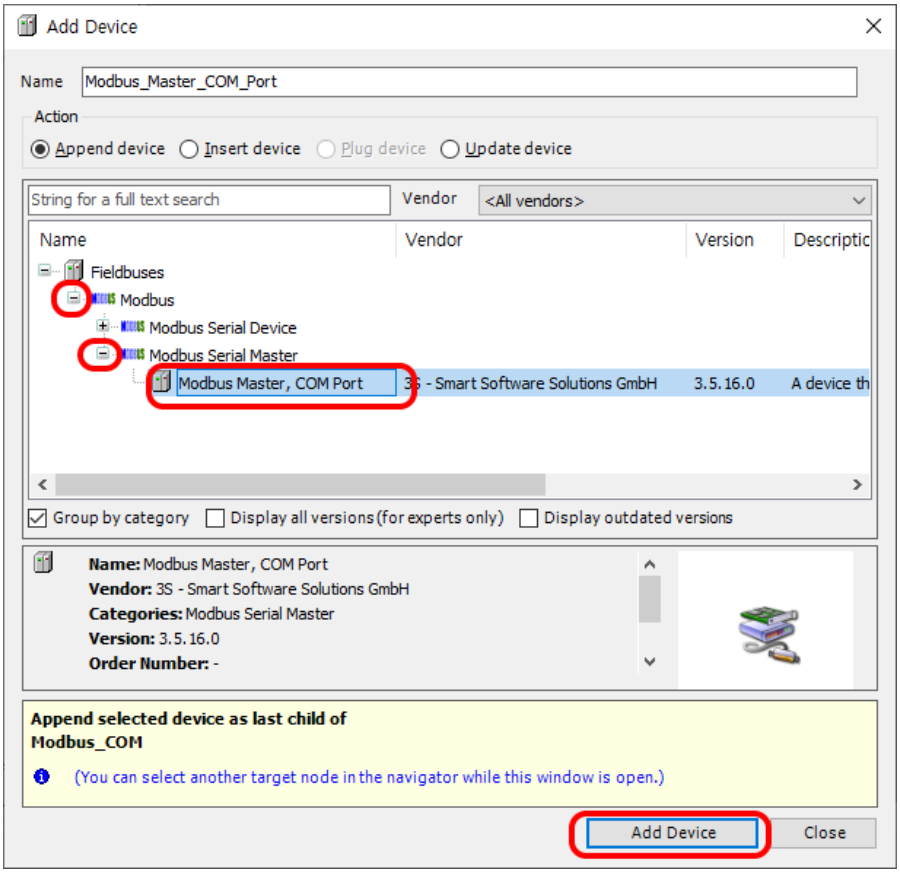
여기서 Modbus COM 선택, 그리고 밑에 Add Device 선택.



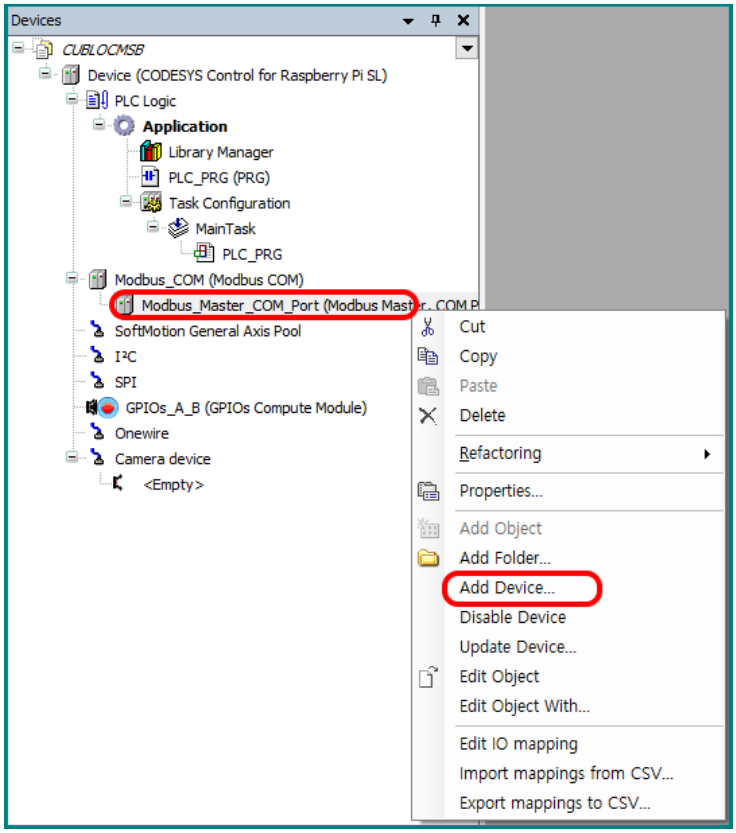
이번엔 방금 추가된 Modbus_COM 우클릭후 Add Device 선택.



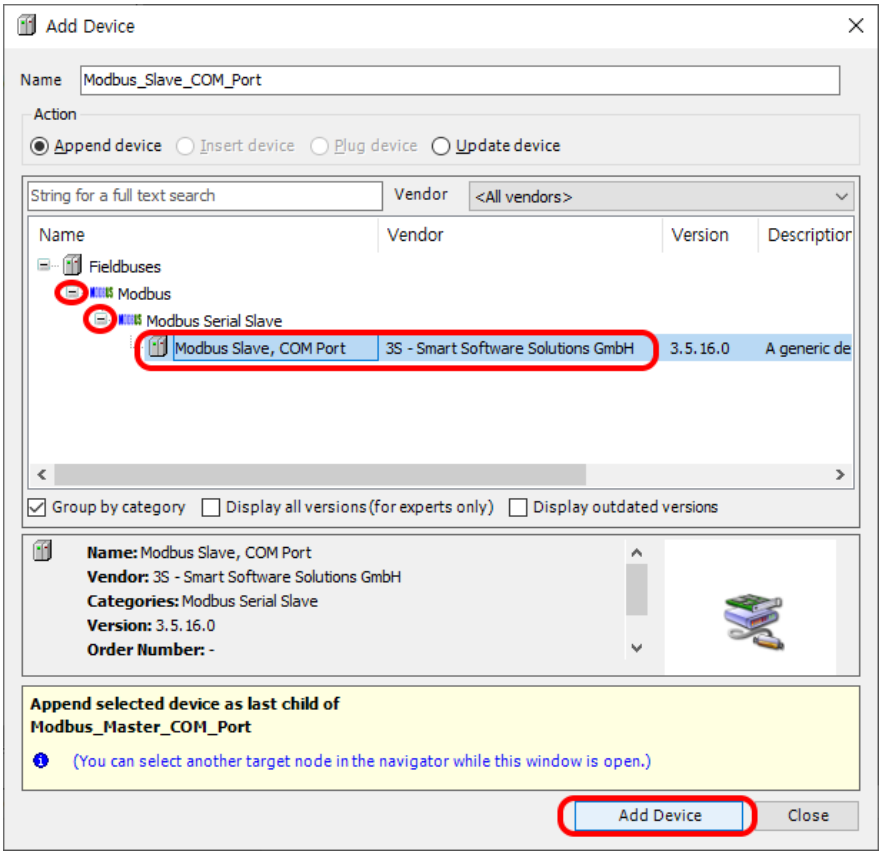
여기서 Modbus Master 선택 후, 하단의 Add Device 선택.



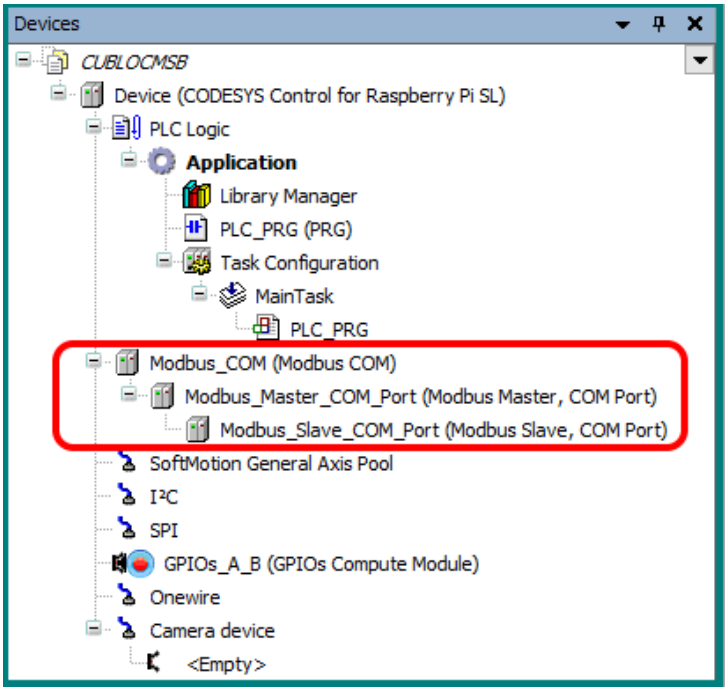
방금 추가된 Modbus_Master 우클릭하고, Add Device 선택.



이번엔 Modbus_Slave 선택후 하단의 Add Device 선택.

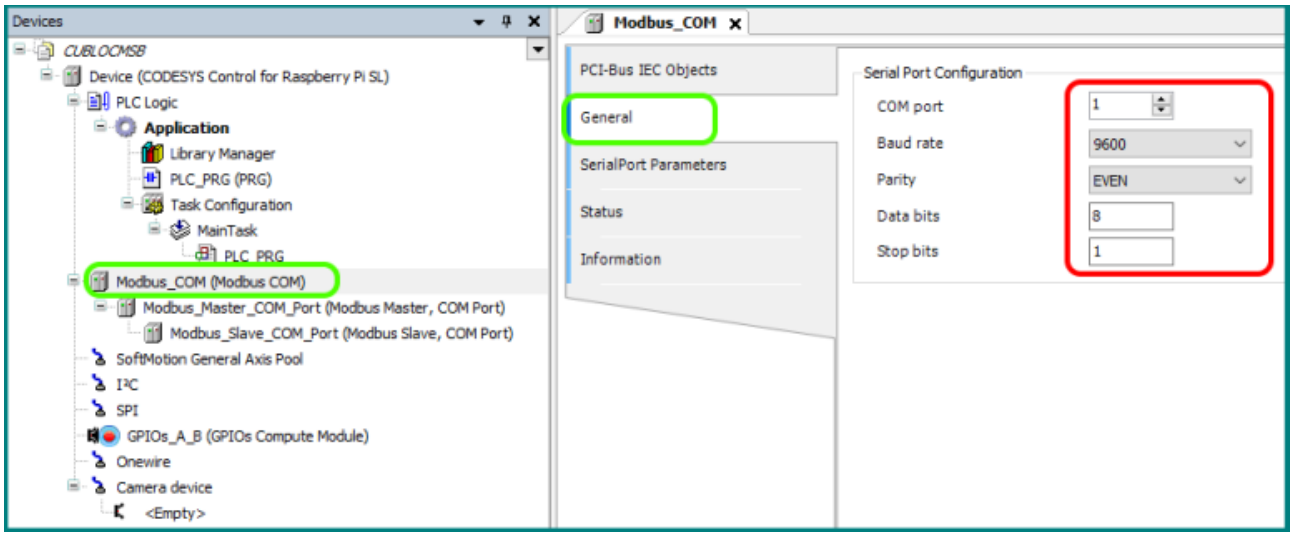


여기까지 하면 왼쪽 탭이 아래모양이 됩니다.

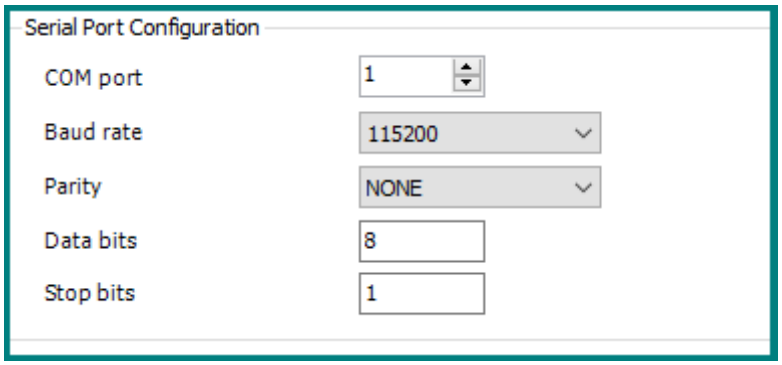


여기까지 일단 설명드리면, CODESYS 가 Master 역할을 수행하고, 그 밑에 필요한 Slave 를 거느리는건데, 지금은 1 개만 넣어놓은 상황입니다.

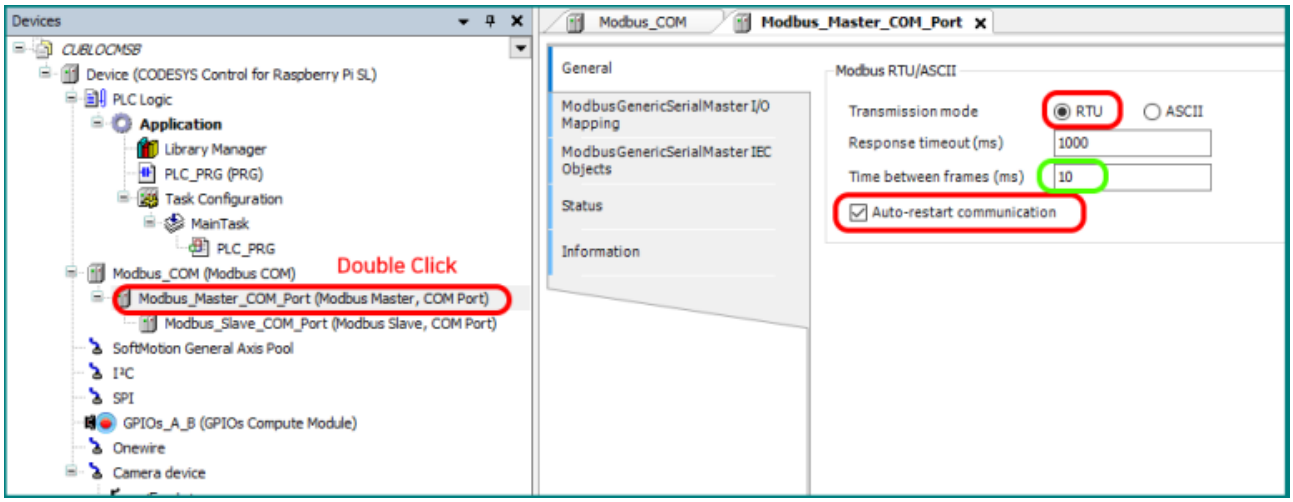
자 이제 속성을 정해보겠습니다. Modbus_COM 을 더블클릭해보세요. 아래 창이 보입니다. 여기에서 필요한 부분을 수정하면 됩니다.



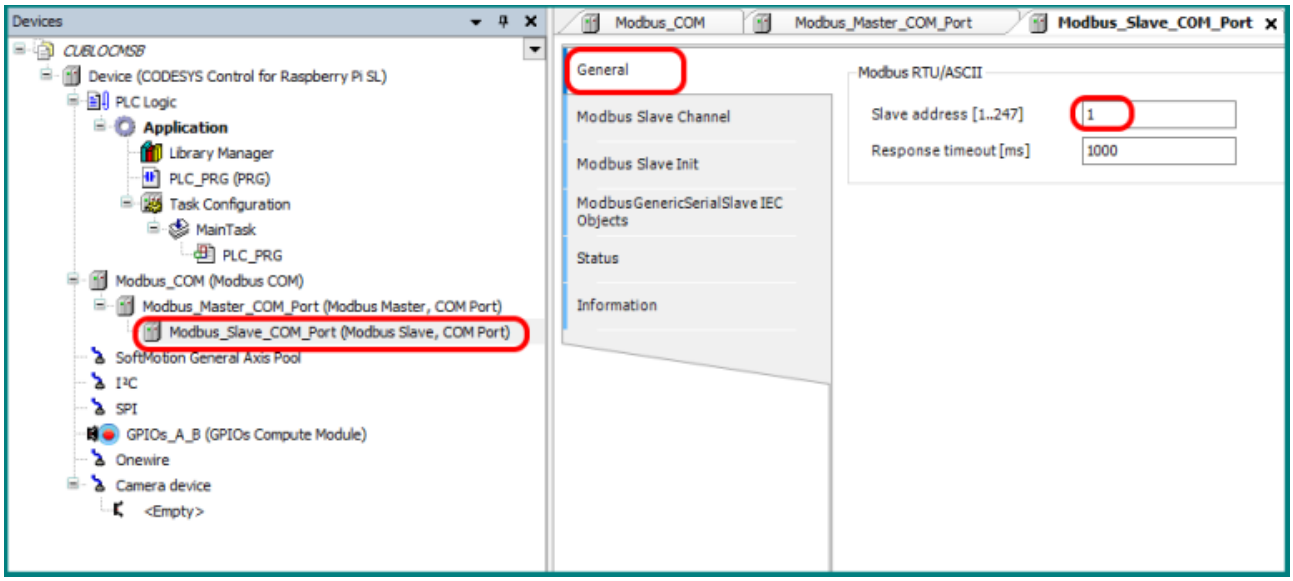
CUBLOC MSB 와 연결해야 하기에 아래처럼 수정했습니다. 115200, none 패리티, 8 비트, 1 스톱비트.



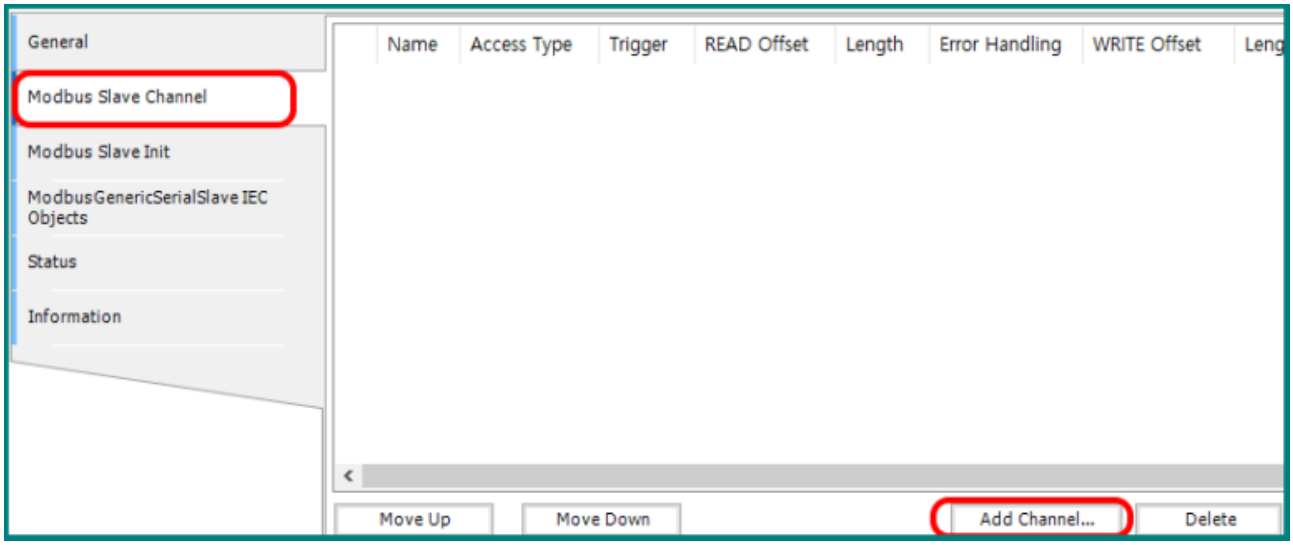
다음은 Modbus_Master 를 더블 클릭하세요. 여기에서 RTU 를 선택하고 Auto-restart 를 반드시 체크하세요. 연두색 박스의 프레임간격은 10mS 정도가 적당합니다.



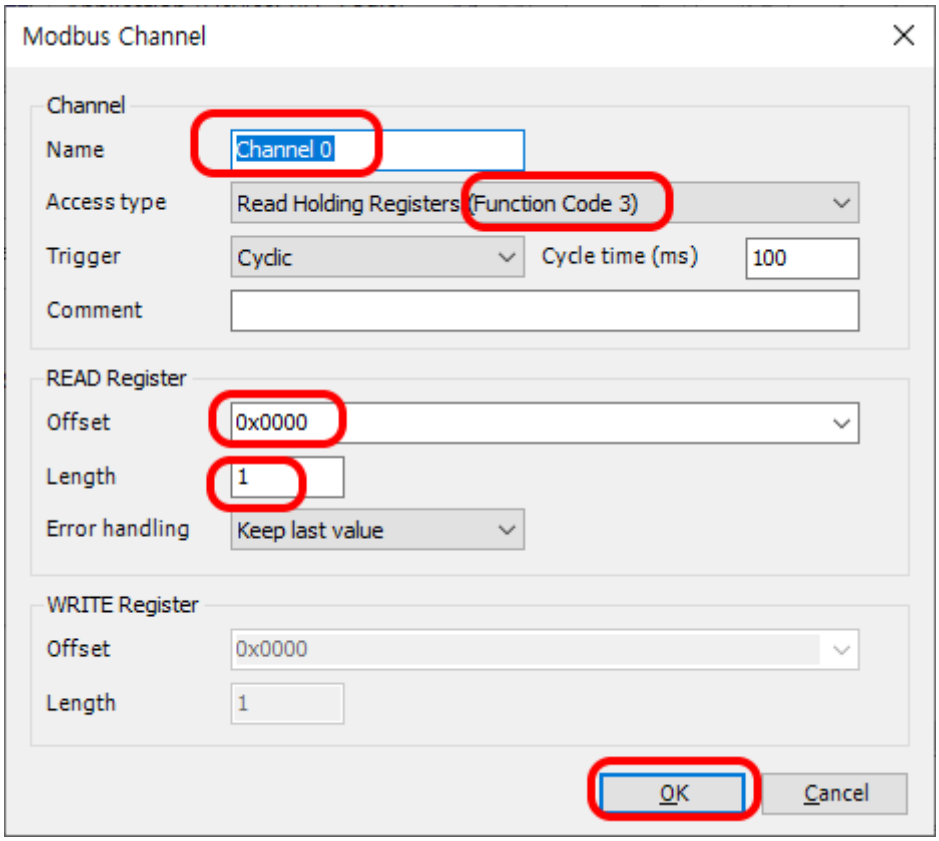
다음은 Modbus_Slave 를 더블클릭하고, General 탭에서 슬레이브 어드레스를 1 로 맞추세요. 큐블록에서도 1 로 할 예정입니다.



이제부터 실제 통신할 내용을 적는 부분이 나옵니다. (지금까지는 설정이였고, 이제부터가 통신내용입니다.) CODESYS 에서는 채널이라는 이름으로 통신할 내용을 지칭하고 있습니다. 밑에 Add Channel 을 누르세요.



동그라미 친 부분이 중요한 부분인데, 이름(Name)은 읽어올 정보가 어떤건지 적어주면 됩니다. 저는 여기를 D0 라고 바꾸겠습니다.



큐블록의 MODBUS 어드레스 표입니다.

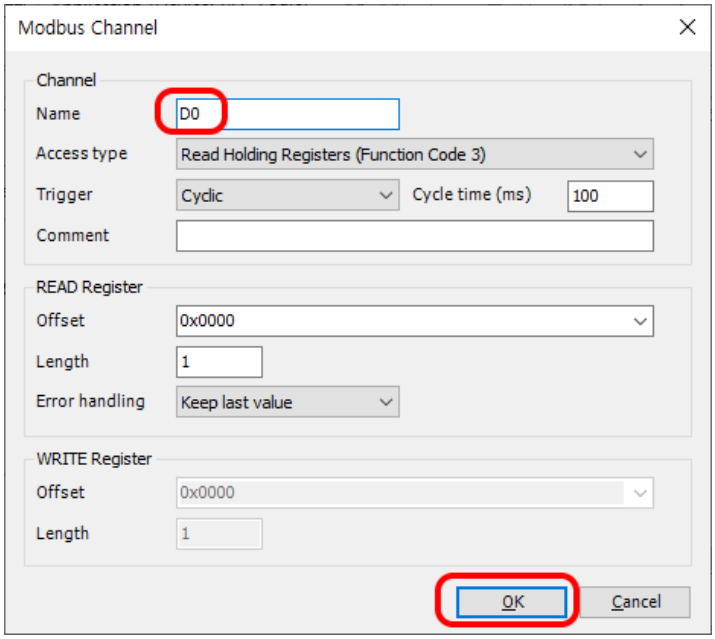
비트 (Coil, Input Status)

영역과 범위	주소	펑션
P area (P0 ~ P127)	0 ~127	1,2,4,15
M area (M0 ~ M2047)	4096 ~ 6145	1,2,4,15

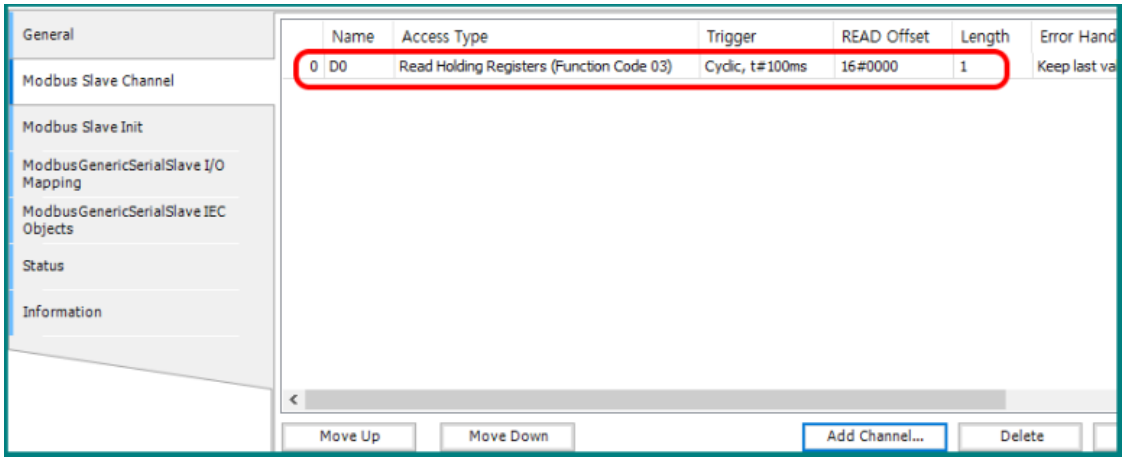
워드 (Holding/Input Registers)

영역 범위	주소	펑션
D area (D0 ~ D511)	0 ~ 511	3, 4, 6, 16
T area (T0 ~ T255)	1000 ~ 1255	3, 4, 6, 16
C area (C 0~ C255)	2000 ~ 2255	3, 4, 6, 16
WM area (WM0 ~ WM255)	3000 ~ 3255	3, 4, 6, 16

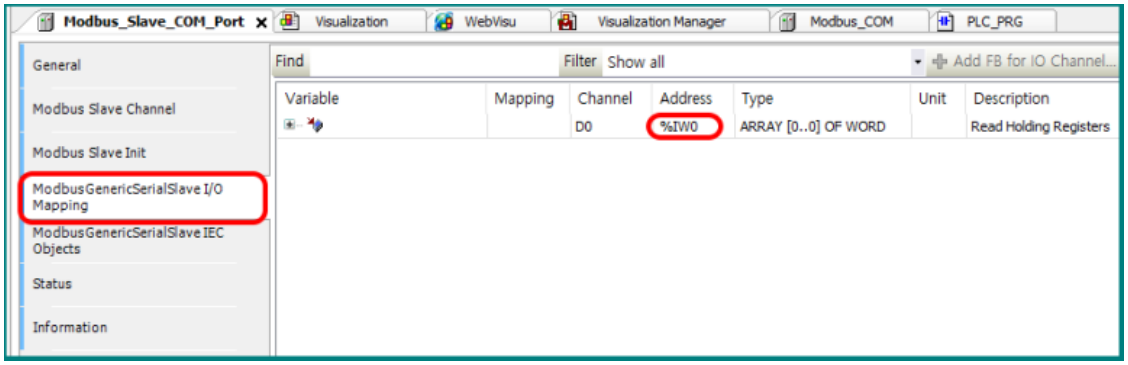
지금은 D0 를 읽을꺼니까 펑션코드 3 번에 Offset 을 0 으로 하세요.



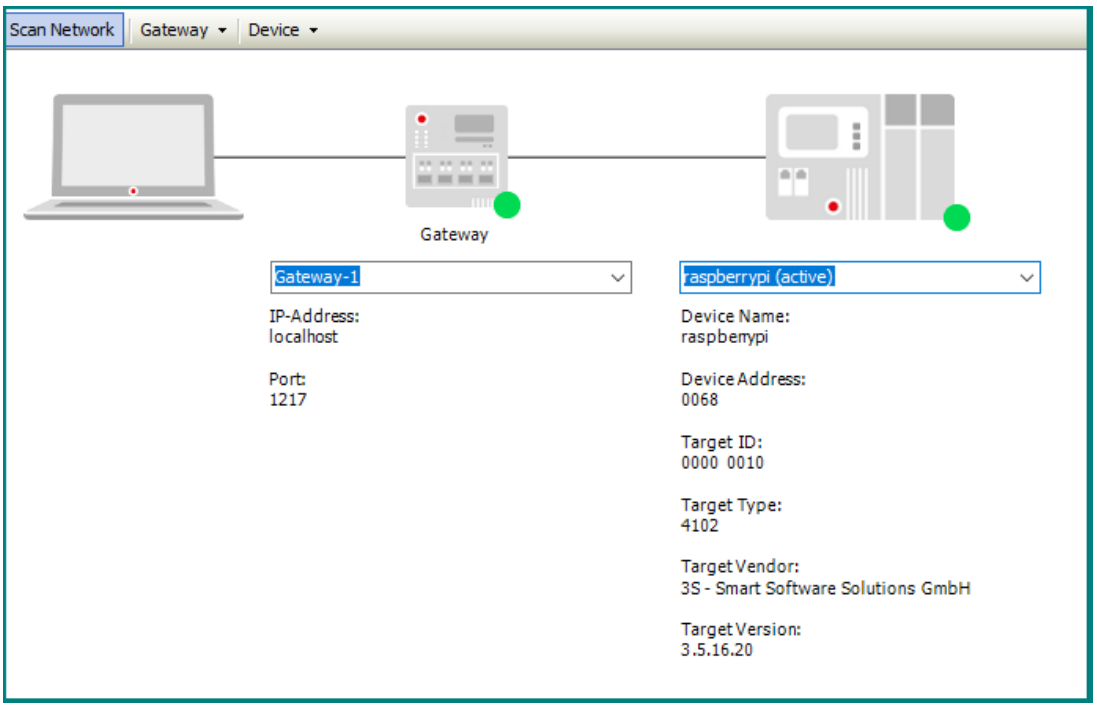
이런 모습이 되었군요. 여기에다 더 읽어올 값이나, 쓸 값이 있다면 계속 Add Channel 을 해주면 됩니다. 지금 당장은 복잡하니까 그냥 D0 하나만 읽어오도록 하겠습니다.



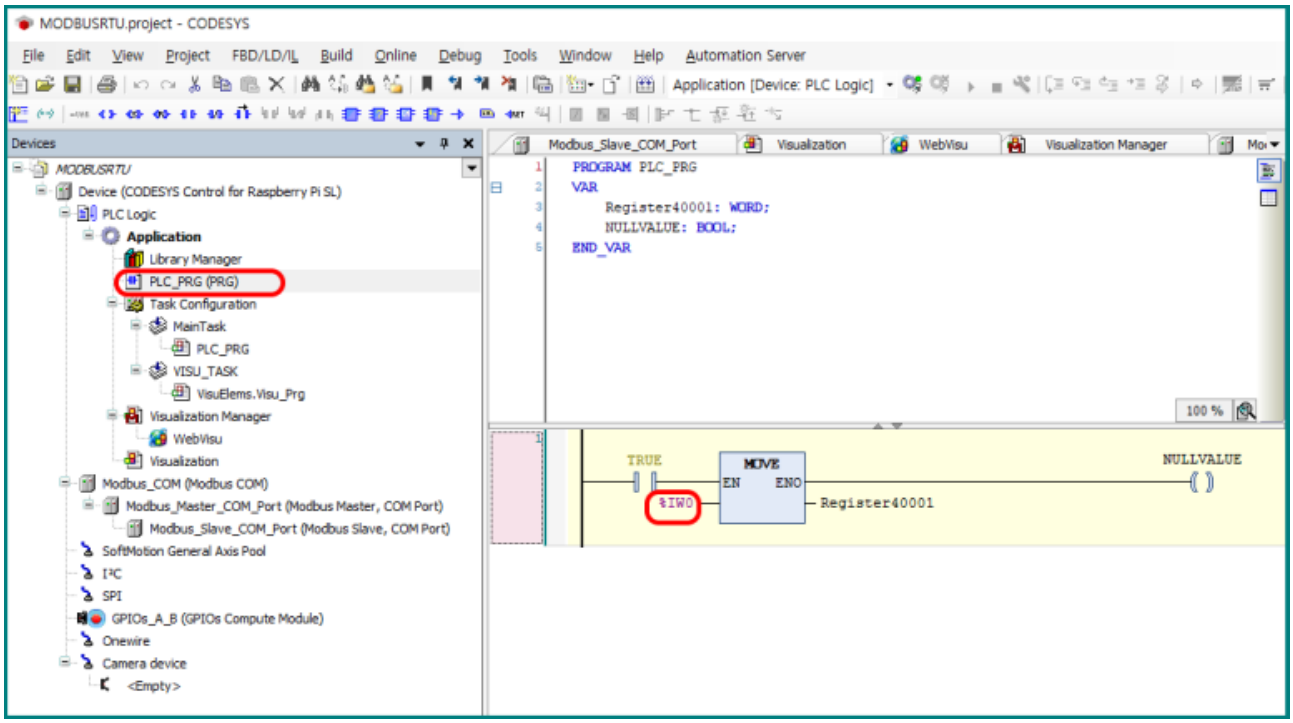
I/O Mapping 을 보면, 방금 추가한 D0 채널이 %IW0 직접번지에 할당되어 있음을 볼 수 있습니다. %IW0 을 잘 기억해두세요.



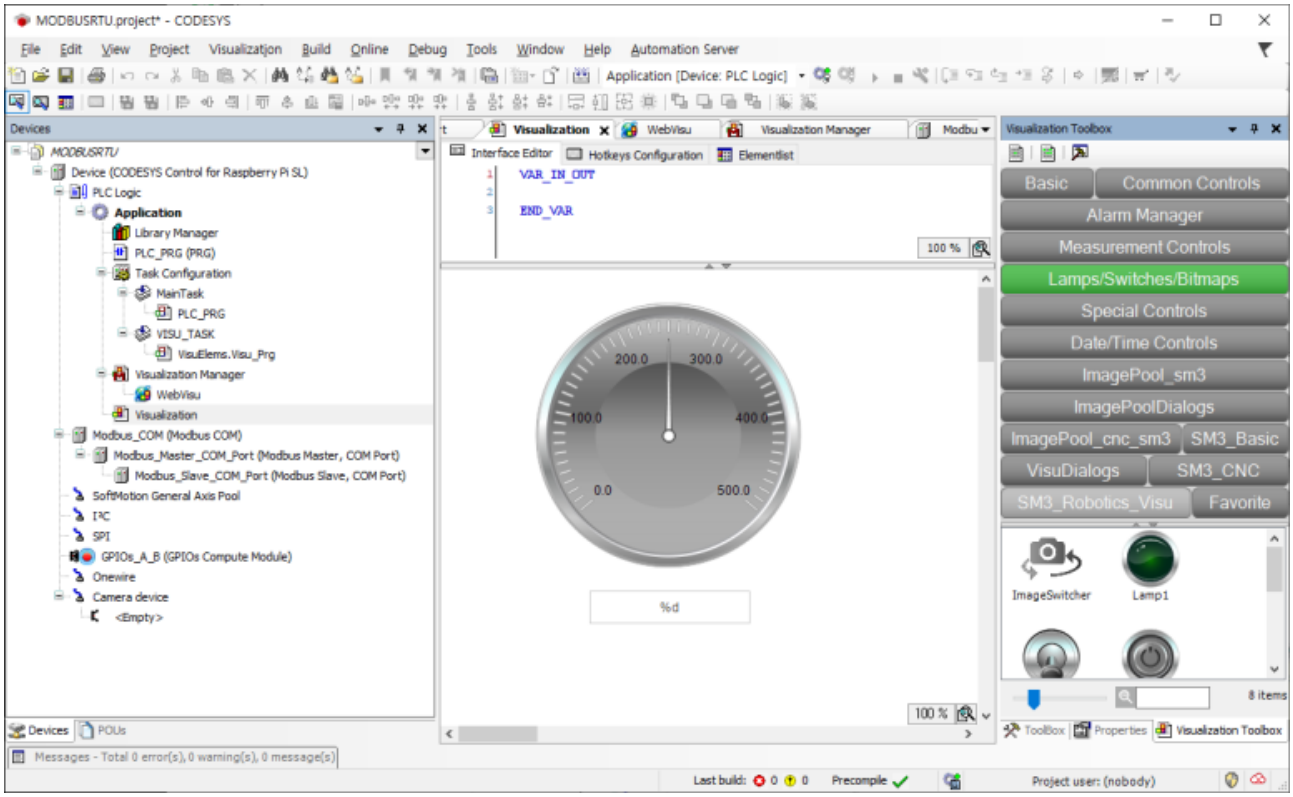
통신관련해서 할일은 다 끝났습니다. 뭘 한거냐면 코데시스의 COM1 포트에서 10 mS 마다 슬레이브 어드레스 1 번에서 40001 번지를 단 한워드만 읽어서 %IW0 에 저장하라고 설정한 것입니다.
언제나 늘 해왔듯, 디바이스 한번 찾아서 링크 시켜주구요.



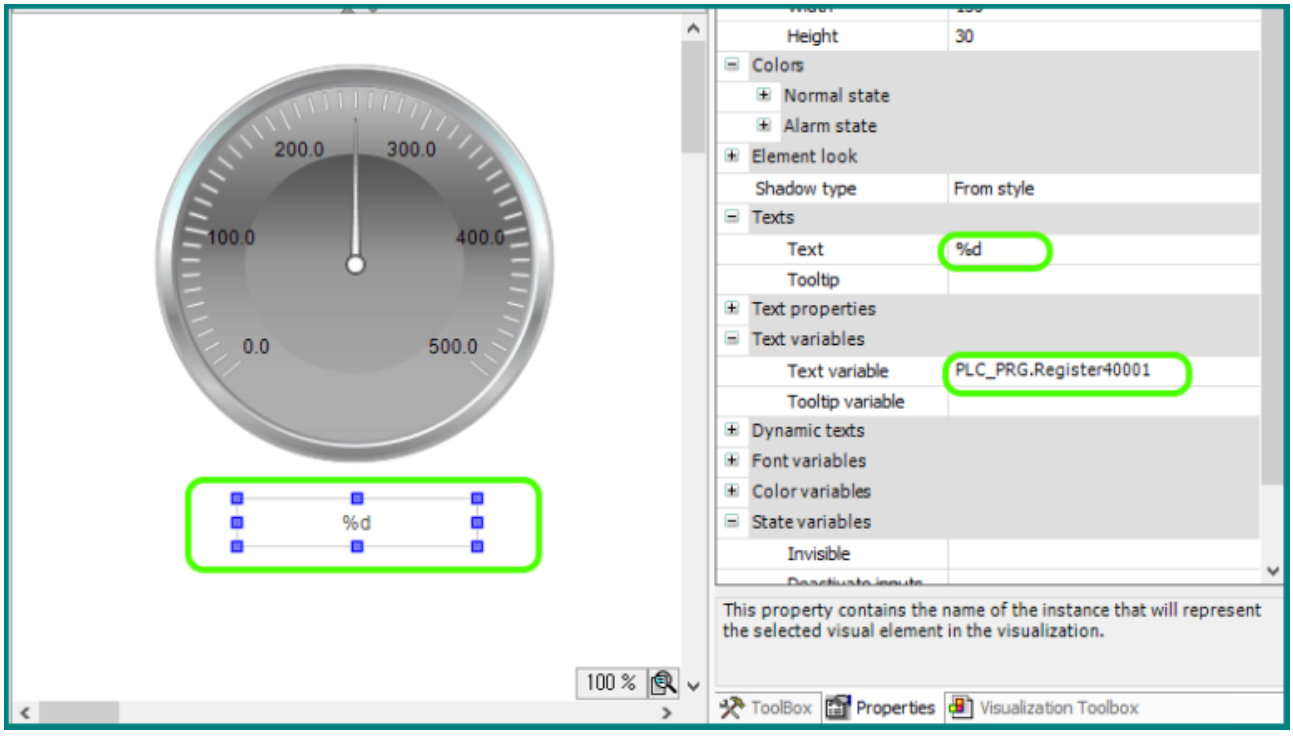
프로그램을 작성해줄 차례입니다. MODBUS RTU 에서 받아온 값을 Register40001 이라는 변수에 저장하는 레더로직입니다.



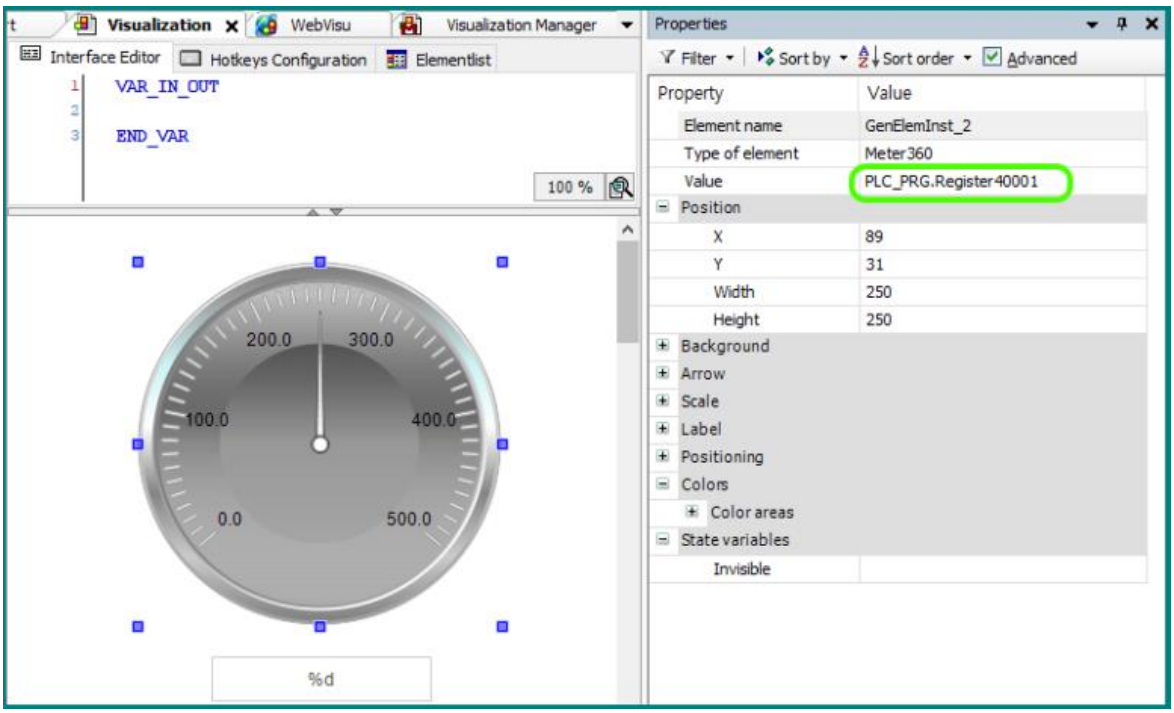
비주얼라이제이션(시각화)에서도 몇개 넣었습니다.



이건 Common Controls 에 있는 Text Field 라는 콘트롤인데 여기에서 2 개를 수정해 주었습니다.

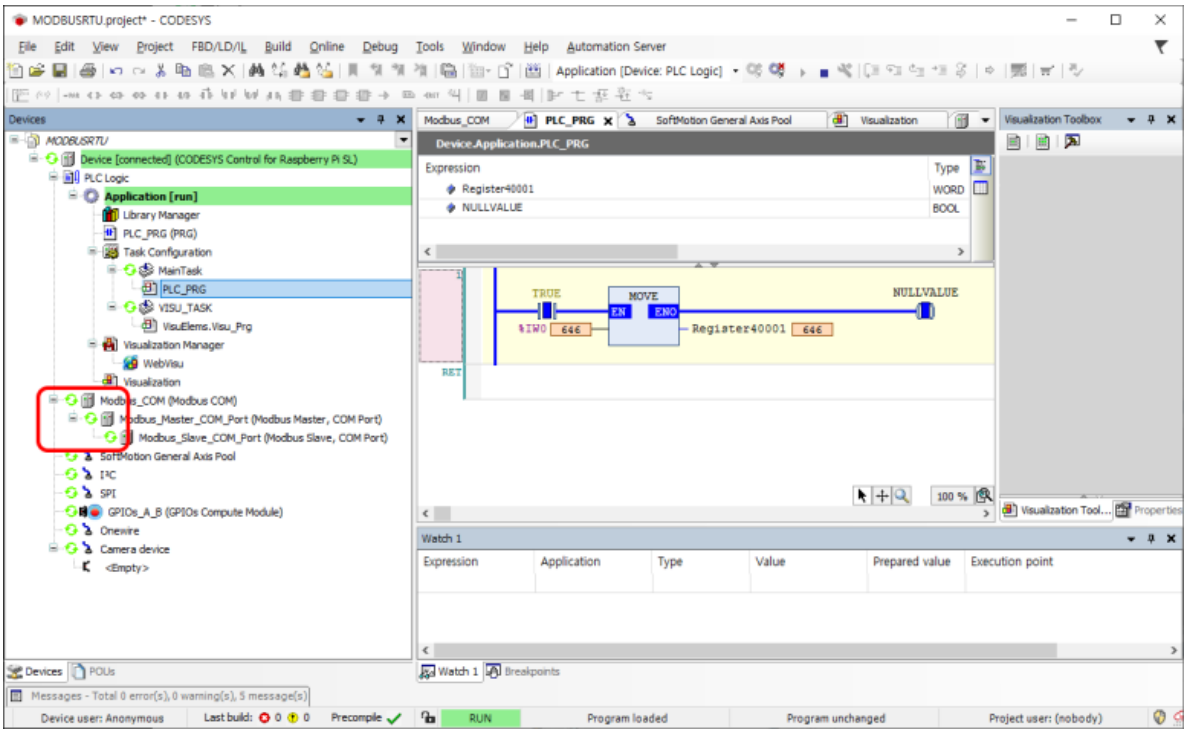


게이지는 Value 부분만 넣어줬습니다.

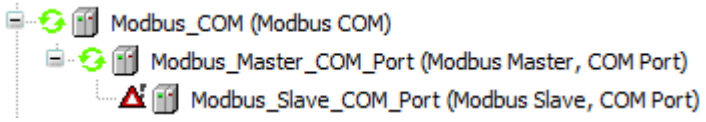


자 이제 빌드하고 온라인하고, 실행까지 해보세요.

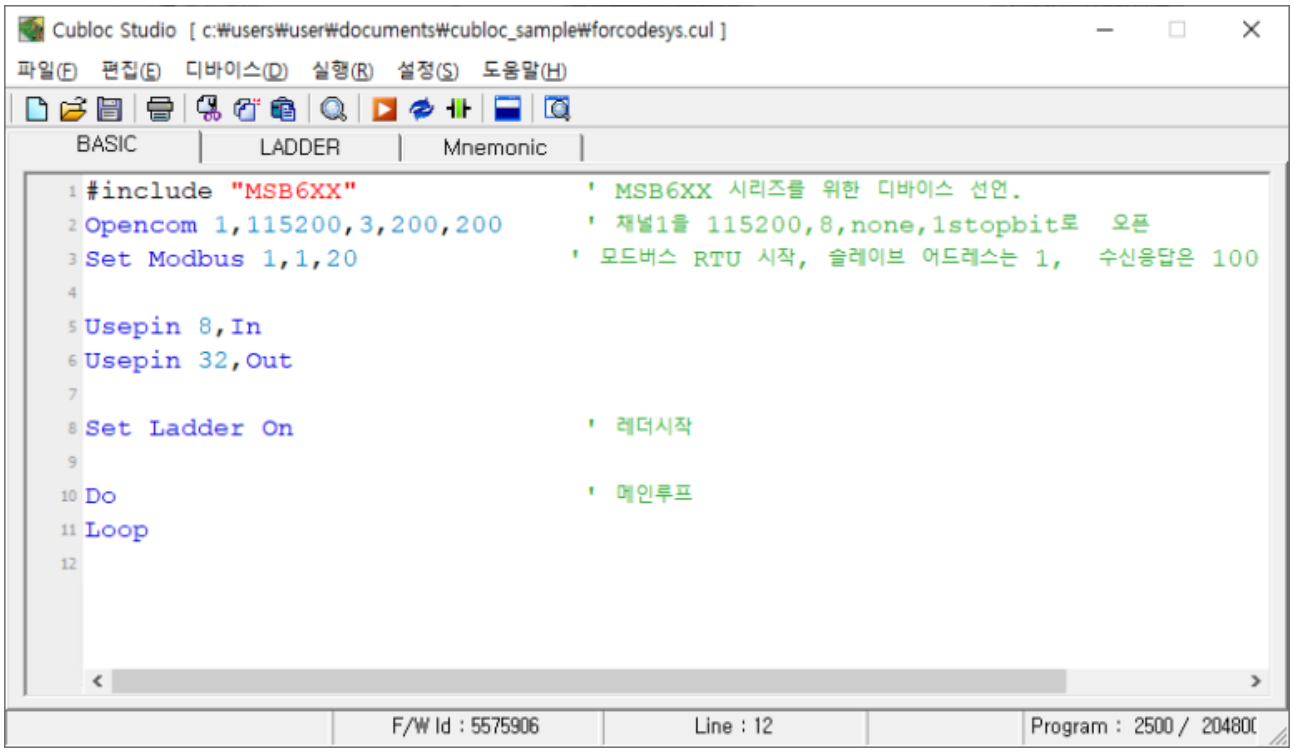
통신이 잘되고 있을 때 만 빨간박스 부분이 연두색 동그라미로 표시됩니다.

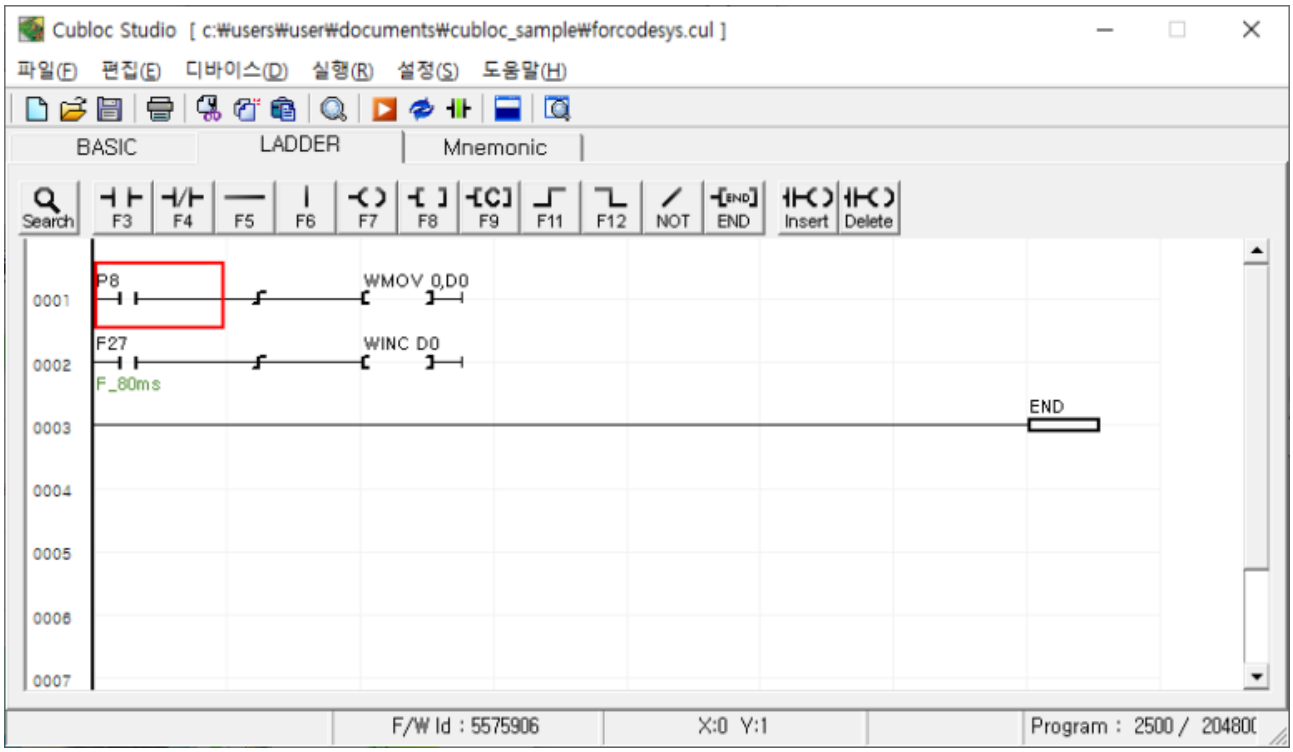


통신이 잘 안되면 삼각형 박스가 나옵니다. 느낌표는 과거에 통신에러가 있었다는 표식입니다. (삼각형과 느낌표가 동시에 나오면 선로이상이거나 상대측이 반응을 안하는 것입니다.)



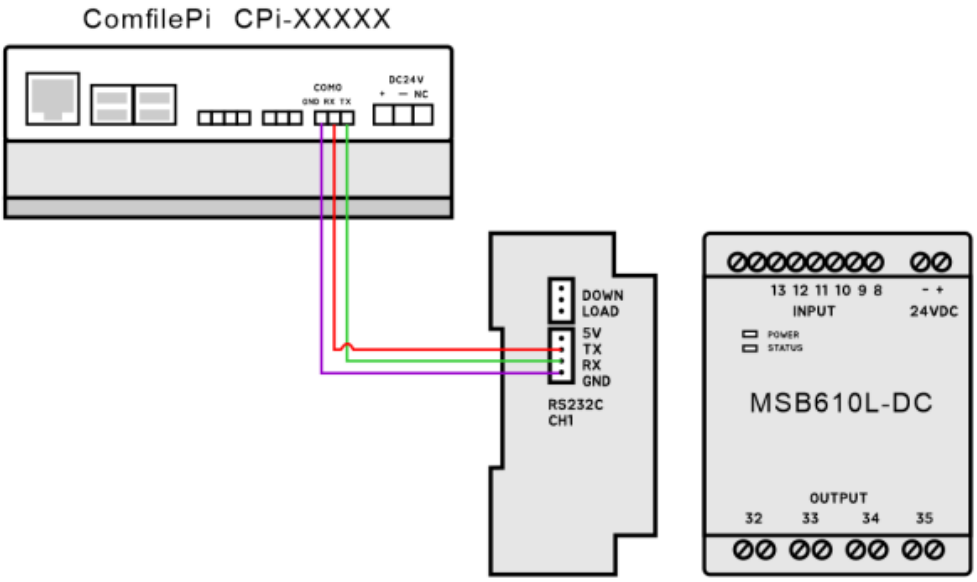
큐블록은 MSB610L-DC 모델을 썼구요. 큐블록 쪽 소스는 이렇습니다.

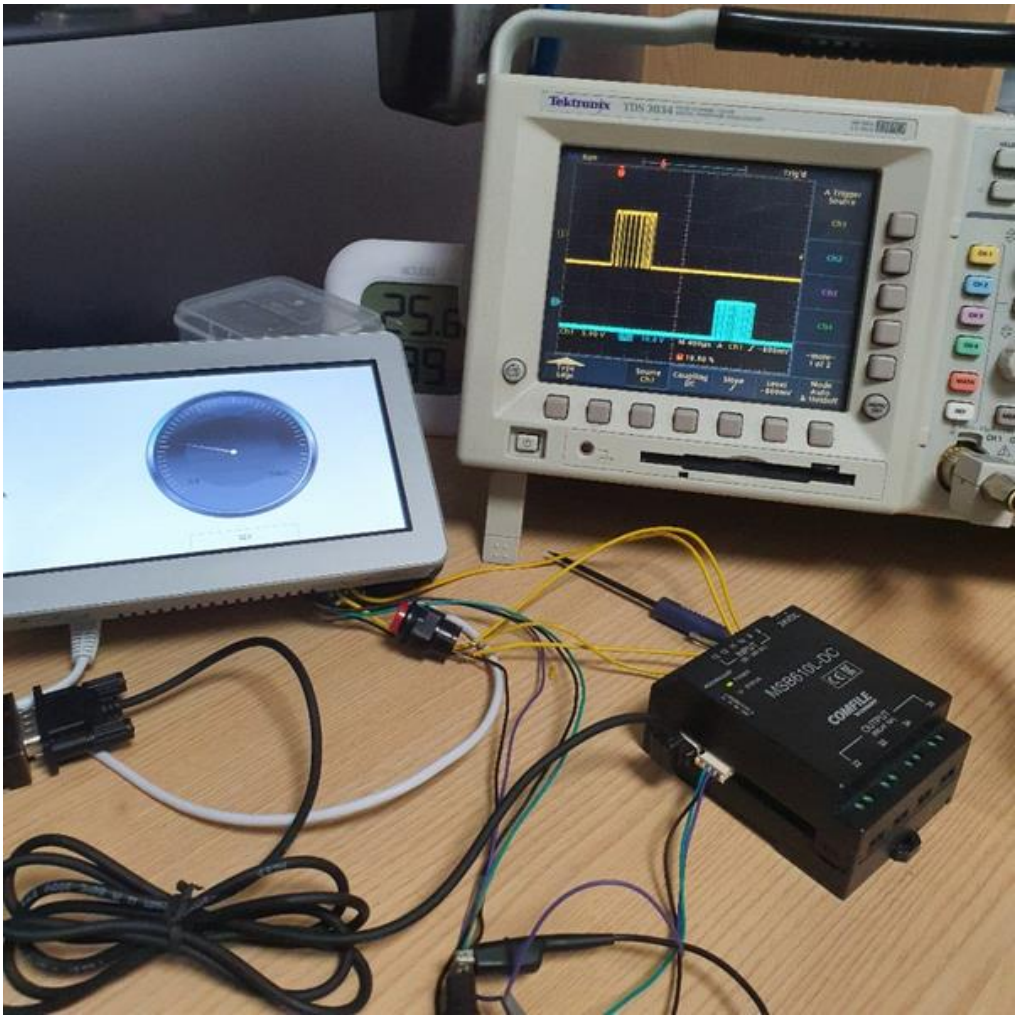




그냥 D0 를 80mS 마다 1 씩 증가하는 프로그램입니다. 이 값을 읽어서 WebVisu 로 볼 예정입니다.

결선은 이렇게 했습니다.



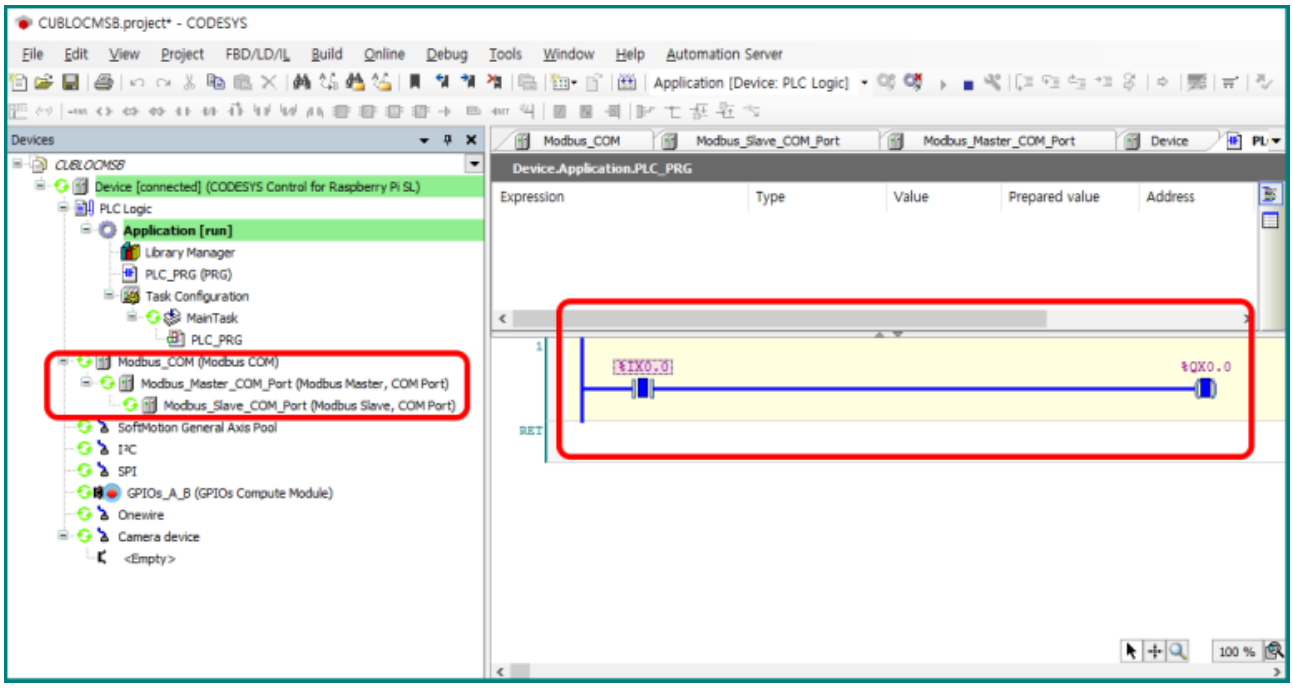


잘됩니다. 화면에는 MSB 로부터 읽어온 증가되는 숫자가 표시됩니다. https://youtu.be/1jw20I734_0

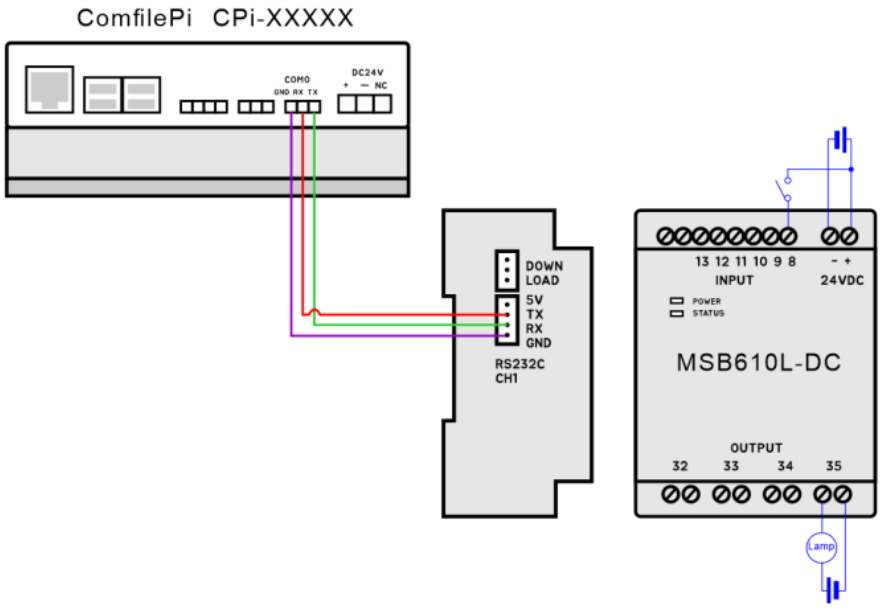
MSB를 이용한 I/O확장예

컴파일 파이에서 CODESYS 를 사용하다가 I/O 가 부족할 때 쓸 수 있는 방법을 소개합니다. 바로 MODBUS-RTU 통신을 이용해서 MSB 를 확장 I/O 로 사용하는 방법입니다.

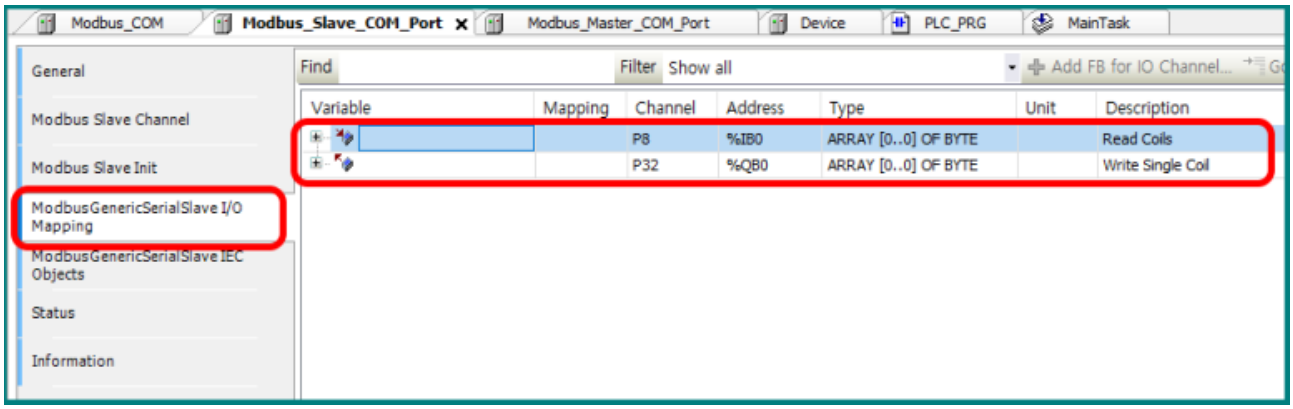
실행 화면부터 보여드리겠습니다.



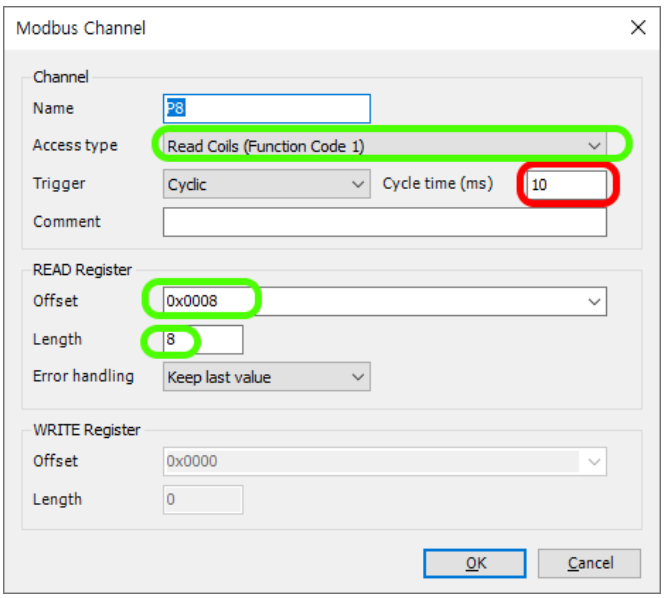
왼쪽 MODBUS 쪽으로 보면 모두 녹색불이 들어와 있습니다. 통신이 잘 되고 있는 상황입니다. 중간에 있는 레더를 보면 하나의 입력을 받아서 출력으로 내보내고 있는 건데, 이게 모두 CUBLOC MSB 에 있는 입력포트와 출력포트입니다.



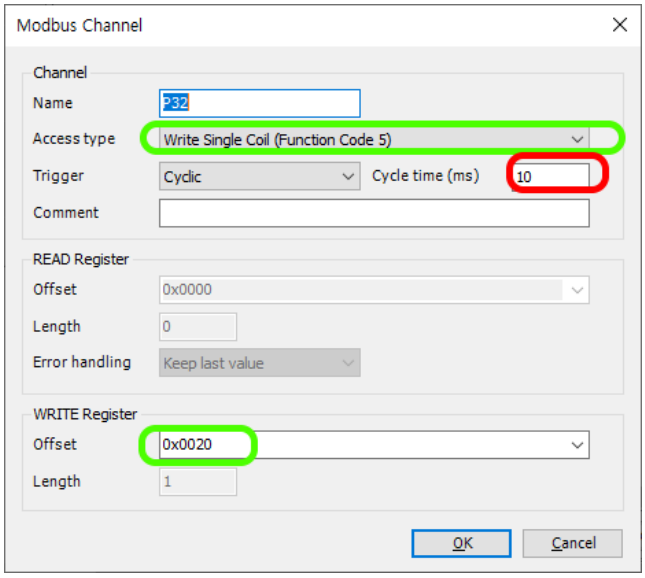
Modbus_Slave 쪽 설정에서 I/O Mapping 을 보시면, 아래처럼 2 개의 채널을 등록해 놨습니다.



위에 Read Coils 은 MSB 의 포트를 읽어오는 채널이고,



아래의 Write Single Coil 은 MSB 에 쓰는(Write)채널입니다. 즉 읽어서 쓰는 구조입니다.



빨간색 박스로 표시한 곳의 숫자를 줄이면 응답속도가 빨라집니다. 참고하세요. (10 이하로는 줄이지 마세요.)

큐블록의 MODBUS 주소는 아래 표를 참조하세요.

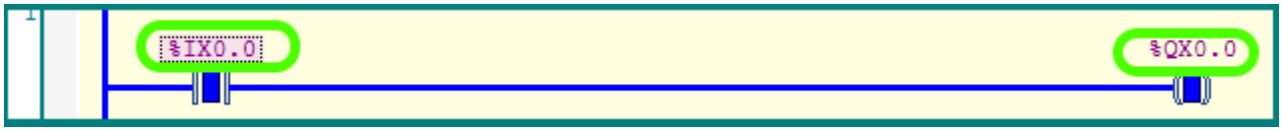
비트 (Coil, Input Status)

영역과 범위	주소	평션
P area (P0 ~ P127)	0 ~127	1,2,4,15
M area (M0 ~ M2047)	4096 ~ 6145	1,2,4,15

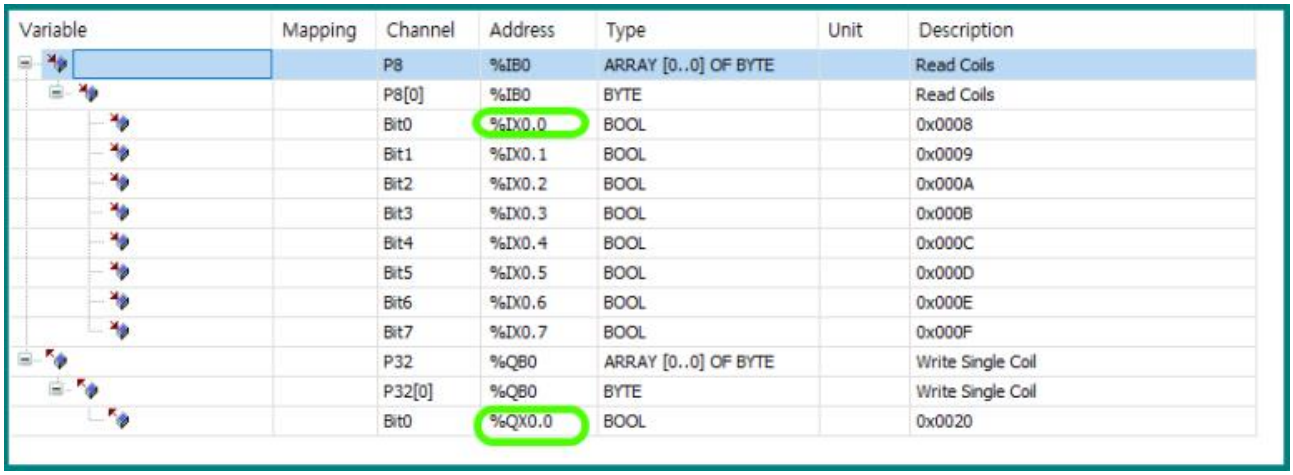
워드 (Holding/Input Registers)

영역 범위	주소	평션
D area (D0 ~ D511)	0 ~ 511	3, 4, 6, 16
T area (T0 ~ T255)	1000 ~ 1255	3, 4, 6, 16
C area (C 0~ C255)	2000 ~ 2255	3, 4, 6, 16
WM area (WM0 ~ WM255)	3000 ~ 3255	3, 4, 6, 16

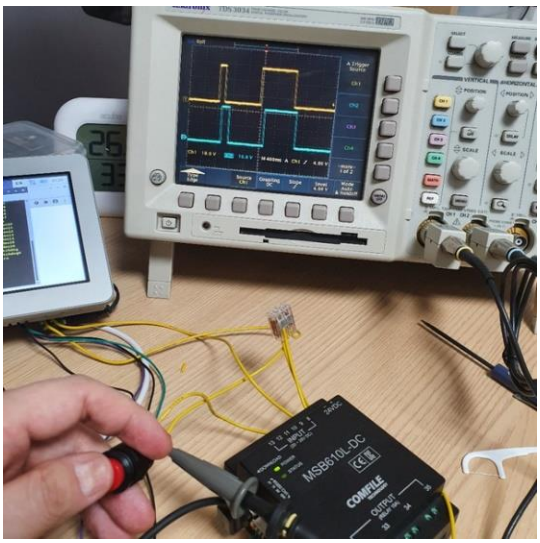
레더로직에 있는 번지가 MODBUS-RTU 에서 MSB 쪽에 링크된 번지입니다.



Modbus_Slave 의 I/O Mappling 을 좀 펼쳐보면 이 어드레스들이 보입니다.

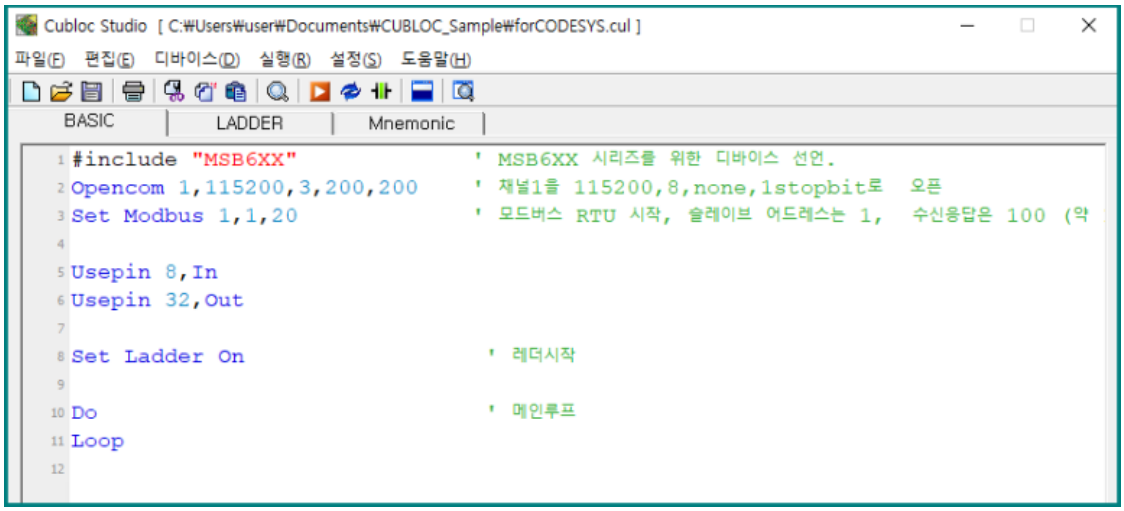


Variable	Mapping	Channel	Address	Type	Unit	Description
		P8	%IB0	ARRAY [0..0] OF BYTE		Read Coils
		P8[0]	%IB0	BYTE		Read Coils
		Bit0	%IX0.0	BOOL		0x0008
		Bit1	%IX0.1	BOOL		0x0009
		Bit2	%IX0.2	BOOL		0x000A
		Bit3	%IX0.3	BOOL		0x000B
		Bit4	%IX0.4	BOOL		0x000C
		Bit5	%IX0.5	BOOL		0x000D
		Bit6	%IX0.6	BOOL		0x000E
		Bit7	%IX0.7	BOOL		0x000F
		P32	%QB0	ARRAY [0..0] OF BYTE		Write Single Coil
		P32[0]	%QB0	BYTE		Write Single Coil
		Bit0	%QX0.0	BOOL		0x0020



https://youtu.be/vYZWQrzhreg 스코프 위에 파형이 입력신호, 아래파형이 출력신호입니다.

큐블록 MSB 쪽 소스입니다.



#include "MSB6XX"

Opencom 1,115200,3,200,200

Set Modbus 1,1,20

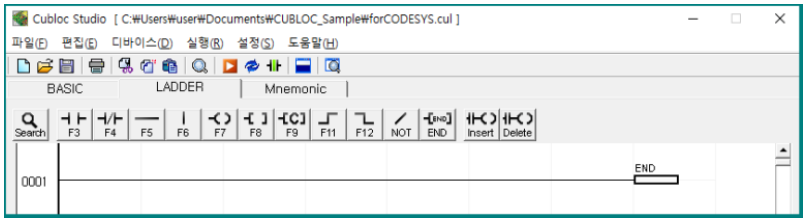
Usepin 8,In

Usepin 32,Out

Set Ladder On ' 레더시작

Do ' 메인루프

Loop



레더쪽은 아무것도 없이 END 명령만 있습니다.

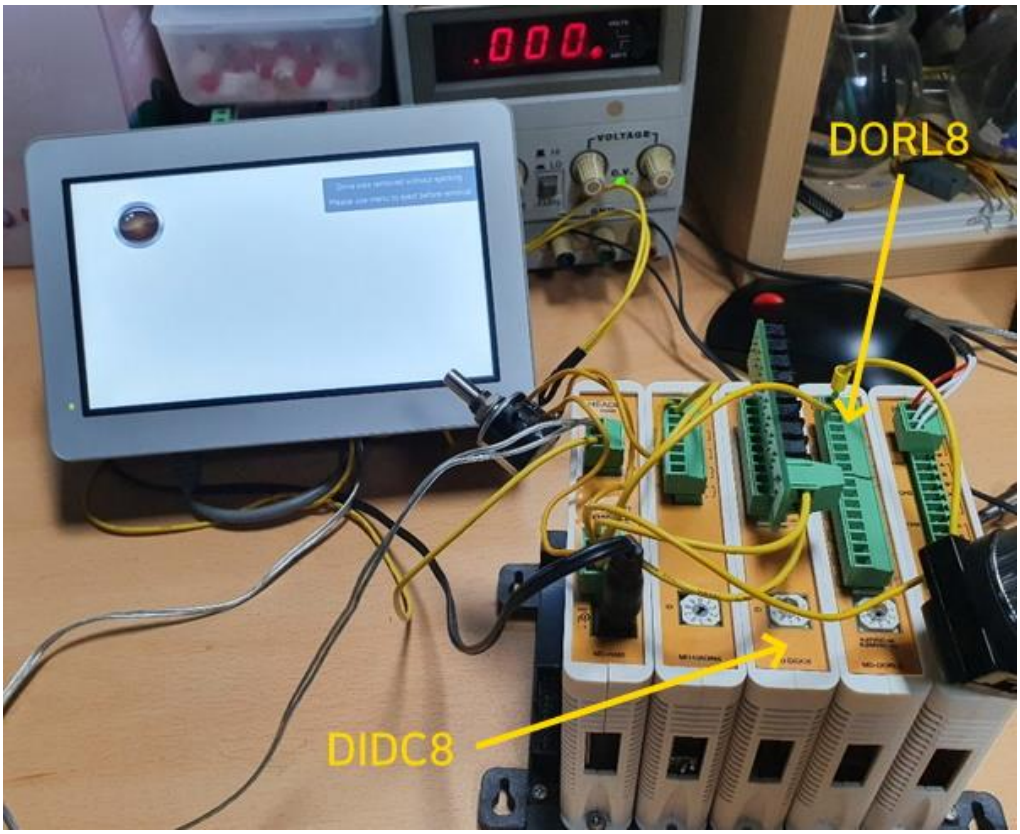
MODPORT 연결

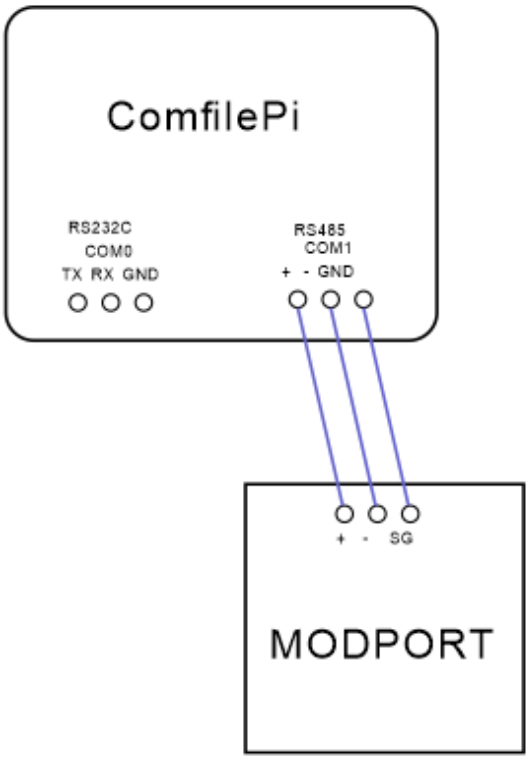
저희 회사에 MODPORT (RS485 MODBUS-RTU) 필드 IO 제품이 있습니다. 컴파일파이+CODESYS 에 딱 어울리는 제품입니다.

모델명	제품코드	설명	가격
MD-H485	M00101	485헤더모듈	16,500원
MD-DIDC8	M00203	DC입력 8점	38,500원
MD-DOSO8	M00206	DC 소스출력 8점	38,500원
MD-DOSI8	M00205	DC 싱크출력 8점	38,500원
MD-DORL8	M00204	릴레이출력 8점	49,500원
MD-ADIN4	M00201	AD입력 4채널 (13.3비트) 0~10V, 1~5V, 4~20mA	55,000원
MD-HADIN4	M00207	AD입력 4채널 (16.6비트) 0~10V, 1~5V, 4~20mA	77,000원
MD-THRT4	M00208	PT100옴 (RTD) 입력 4채널	55,000원
MD-DAOUT2	M00202	DA전압출력 2채널 (16비트) 0~10V, 0~5V	66,000원
MD-DAOUT2B	M00209	DA전류출력 2채널 (16비트) 4~20mA, 0~20mA	66,000원

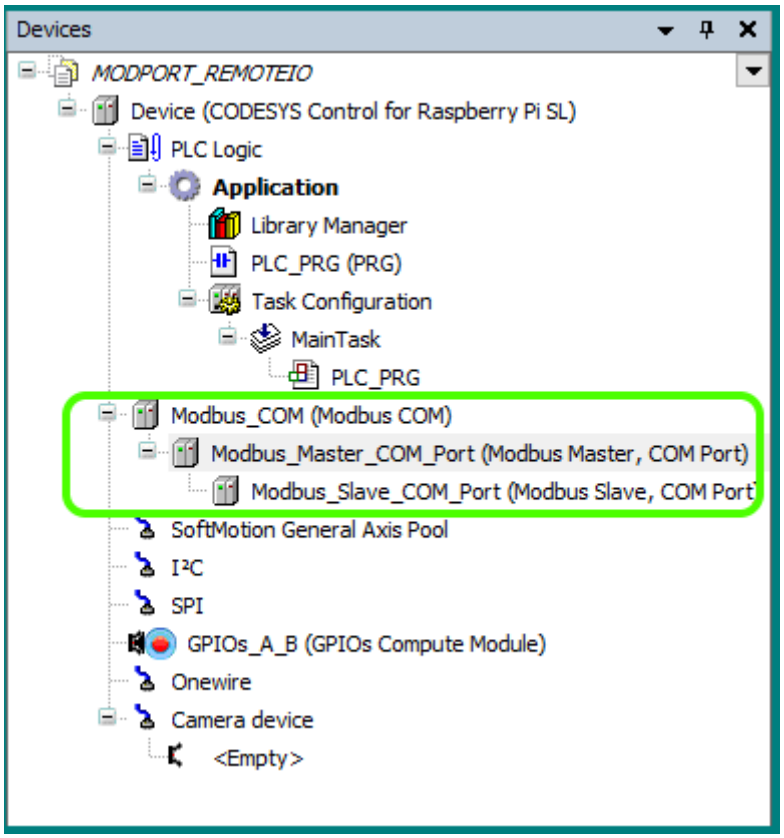
가격이 그렇게 비싸지는 않습니다.
https://www.comfile.co.kr/shop/goods/goods_view.php?goodsno=162&category=009006

그럼 연결하는 방법을 설명드리겠습니다. ComfilePi RS485 포트와 MODPORT 를 아래 사진처럼 연결했습니다.

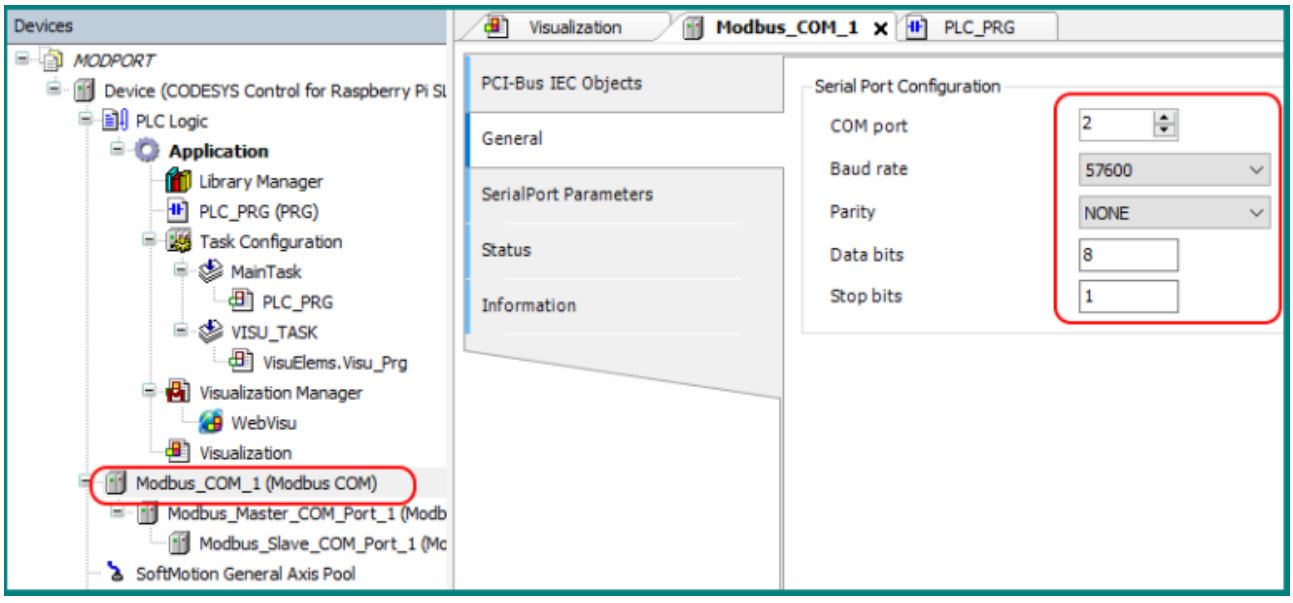




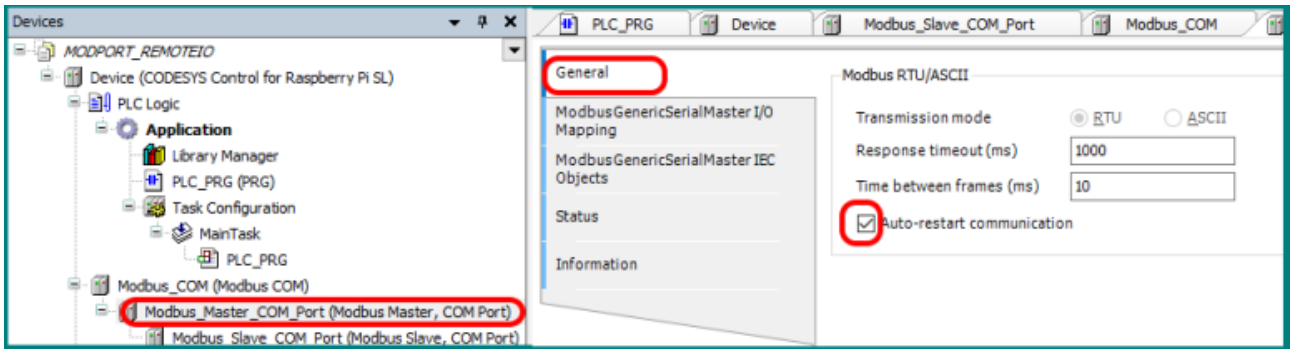
설명 드린바와 같이 MODBUS 설정을 해주세요. 이렇게 3 개가 되도록 추가해주세요.



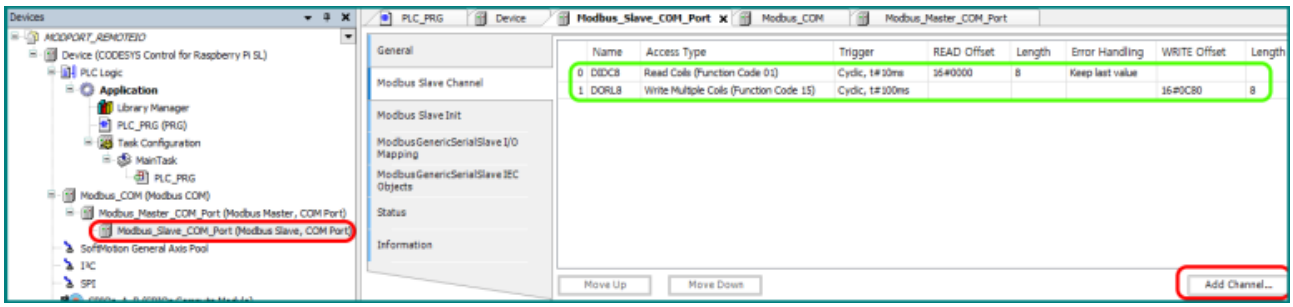
첫번째. Modbus_COM 더블클릭후 General 옵션에서 아래 보이는 것처럼 바꿔주세요. COM Port 는 2 입니다.



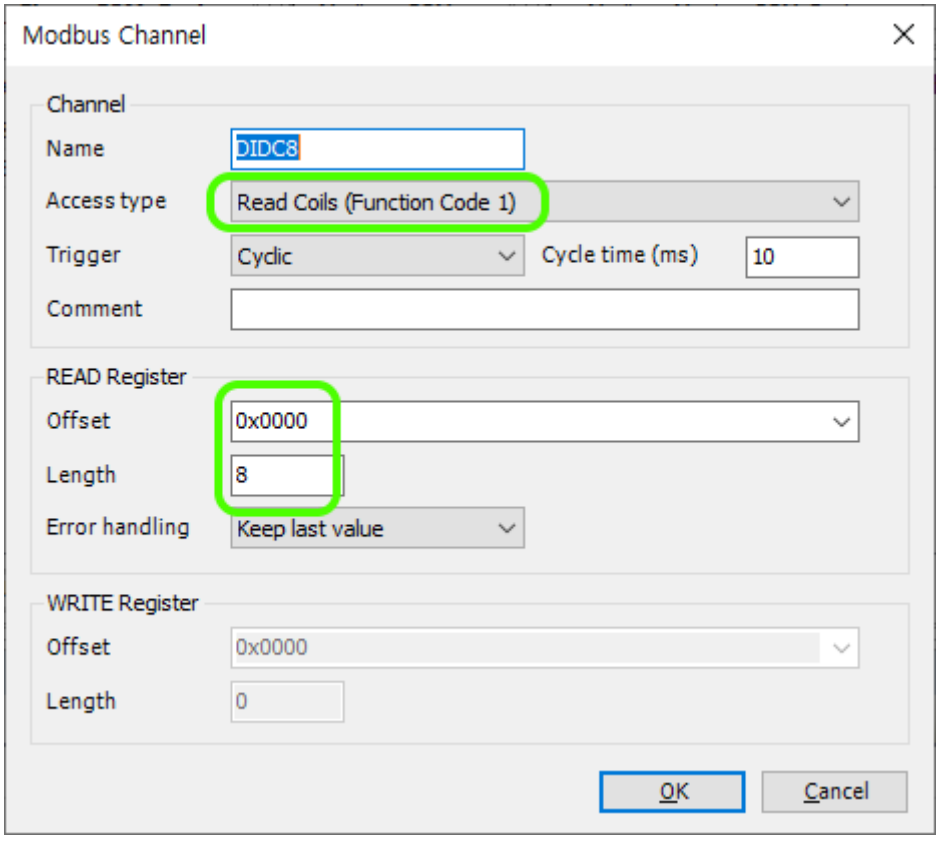
두번째. Modbus_Master_COM_Port 를 더블클릭후, General 옵션에서 Auto-restart 를 체크해주세요.



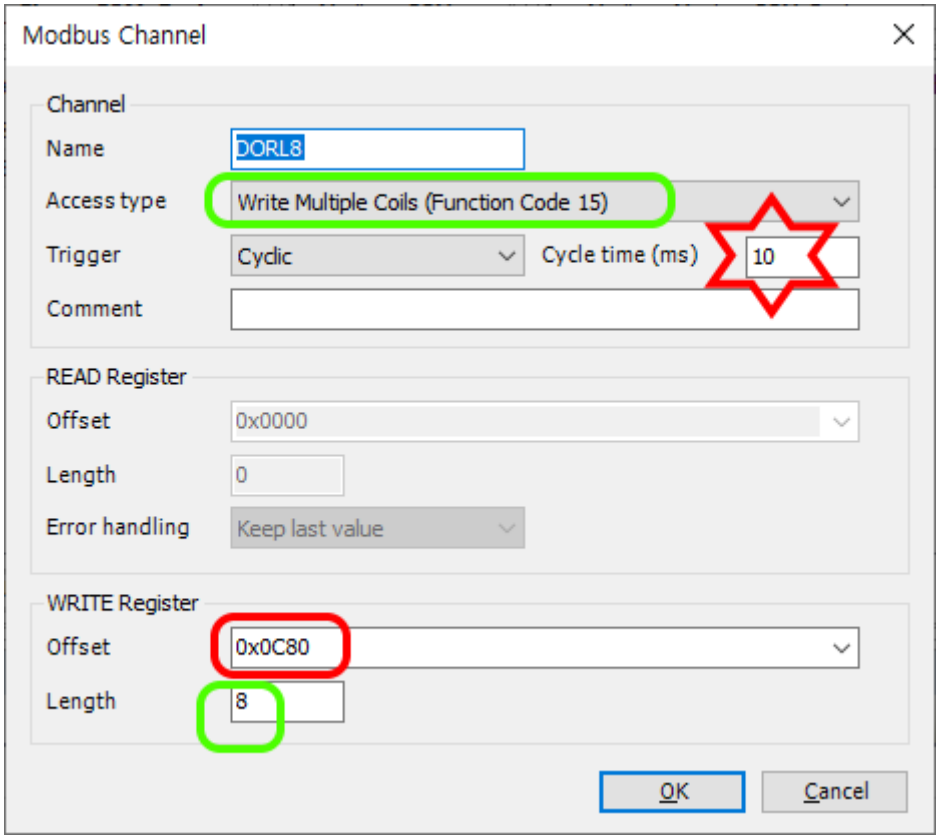
세번째. Modbus_Slave_COM_Port 를 더블클릭후 Add channel 을 눌러서 2 개 항목을 추가해 주세요.



첫번째 항목은 DIDC8 모듈에서 8 개의 입력상태를 읽어오는 채널값입니다. MD-DIDC8 은 DI 8 점 입력모듈입니다.



두번째 항목은 DORL8 모듈에 8 개의 값을 써넣는 채널값입니다. MD-DIORL8 은 릴레이 출력모듈입니다. Cycle time 은 10mS 정도가 적당합니다.



여기에서 Offset 값을 눈여겨 봐주세요. MODPORT 메뉴얼을 보면 MD-DORL8 의 시작어드레스는 3200 으로 나와있습니다. 그래서 Offset 에 3200 을 써줬습니다. (자동으로 16 진수값이 표시됩니다.)

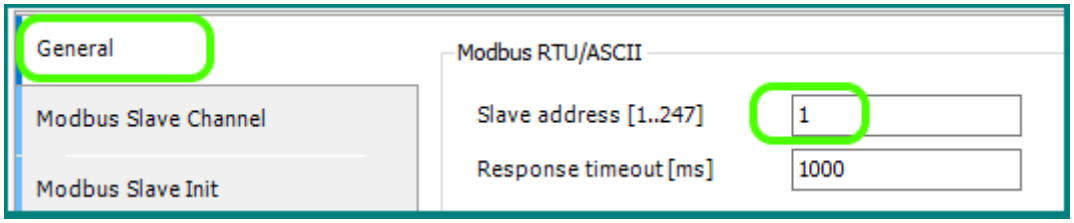
입 력

시작어드레스	제품명	지원 펑션코드
0-7	MD-DIDC8 디지털입력	1
100-103	MD-ADIN4 아날로그입력	3
200-207	MD-HADIN4 고해상도 아날로그 입력	3
300-303	MD-THRT4 RTD 온도입력	3

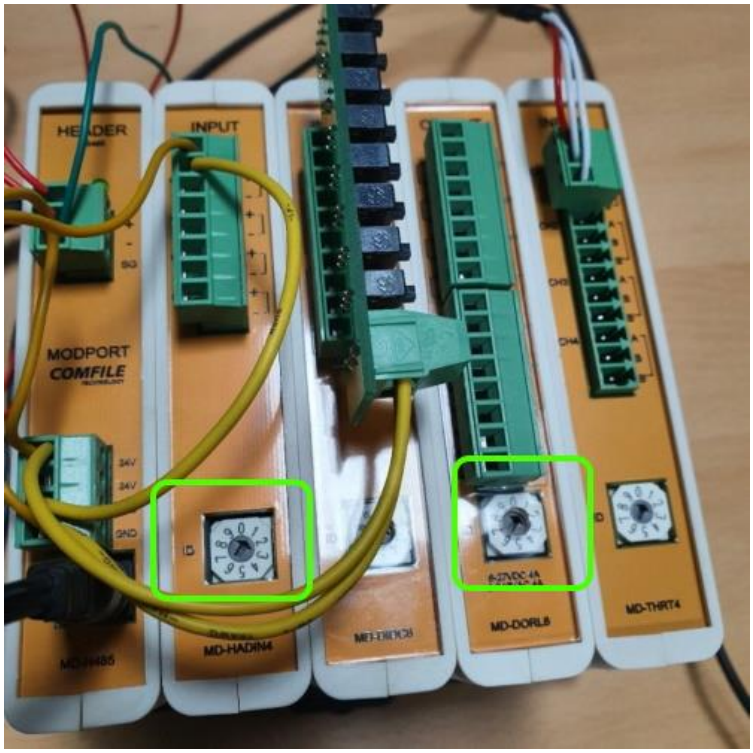
출 력

시작어드레스	제품명	지원 펑션코드
3000-3007	MD-DOSO8 디지털 소스 출력	1, 5, 15
3100-3107	MD-DOSI8 디지털 싱크 출력	1, 5, 15
3200-3207	MD-DORL8 릴레이 출력	1, 5, 15
3300-3301	MD-DAOUT2 아날로그 출력 (전압)	6, 16
3400-3401	MD-DAOUT2B 아날로그 출력 (전류)	6, 16

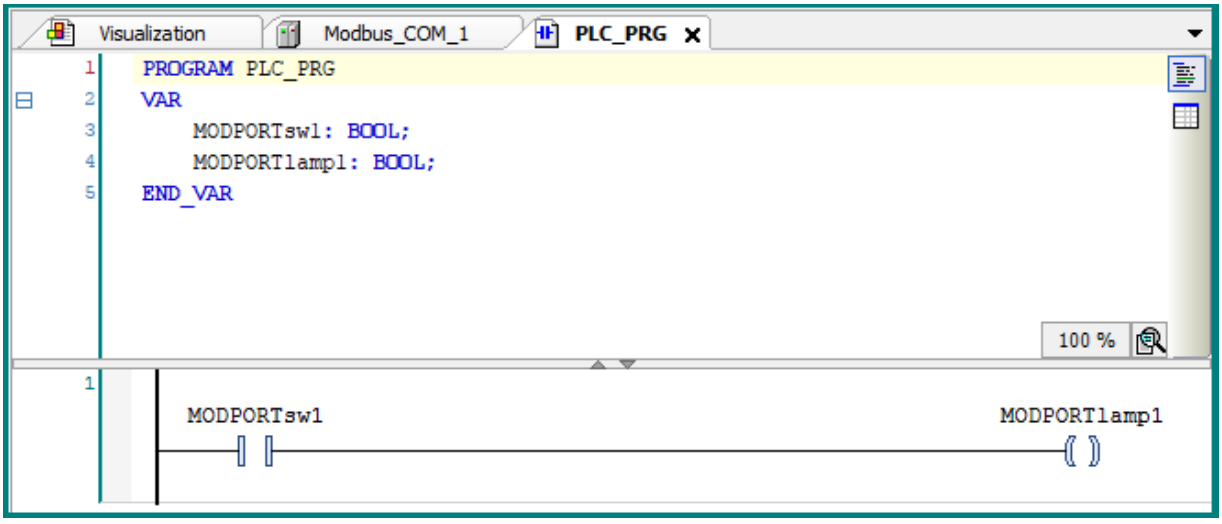
Modbus_Slave_COM_Port 에서 하나 더 신경써줘야 하는 부분이, General 탭에서 Slave Address 입니다. 아래처럼 1 로 해놨을 경우,



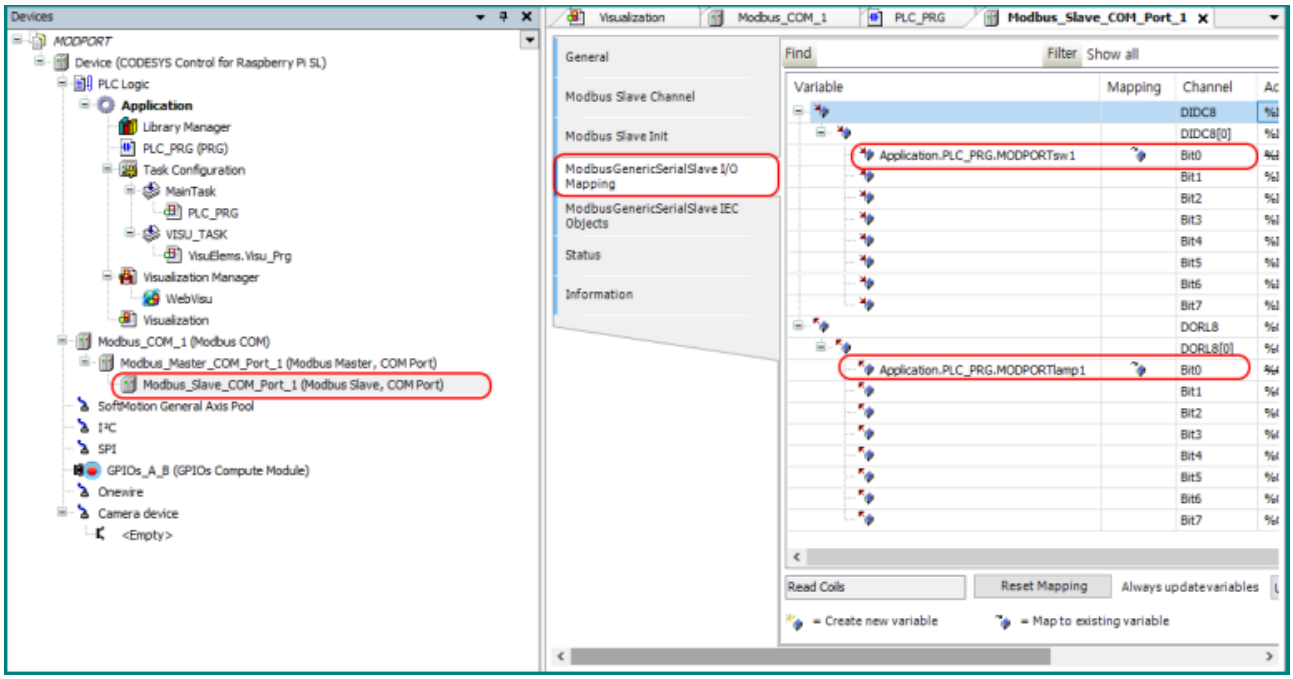
MODPORT 제품에도 ID 라는 로터리스위치를 1 로 맞추어야 합니다.



자 이제 레더로직을 써줄 차례입니다. 보시는 것처럼 하나를 입력받아서 전달해주도록 해놨습니다.



I/O 매핑을 보면 이렇게 되어 있습니다.



동작은 물론 잘 됩니다.
https://youtu.be/xKSBUECEAIY

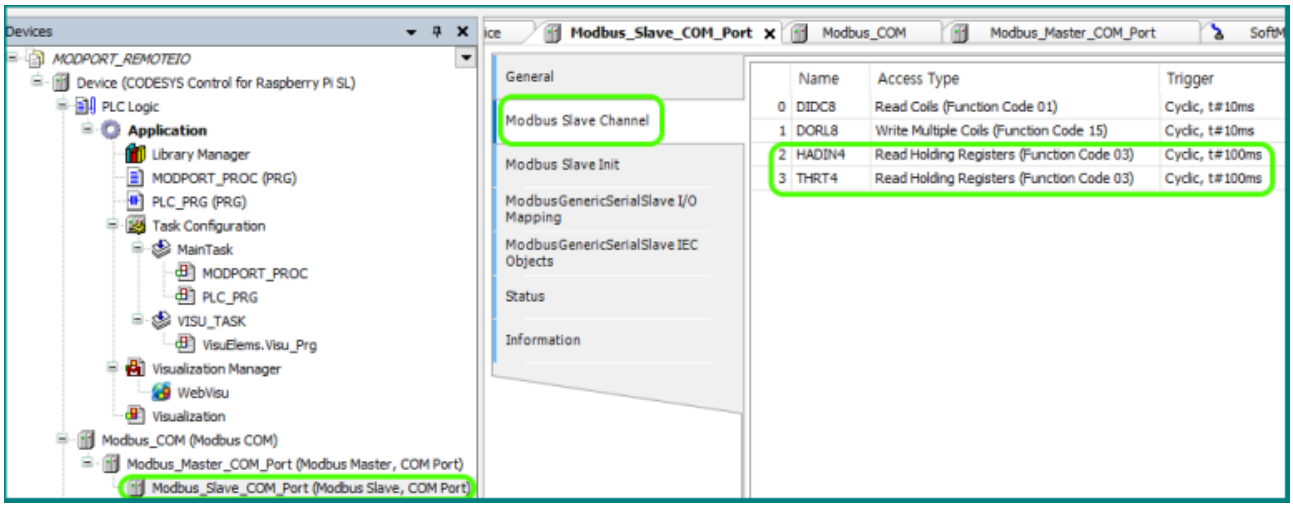
온도입력과 아날로그 입력

MODPORT를 이용해서 CODESYS에 A/D 입력값과 온도값을 표시해 보겠습니다. MODPORT에는 A/D 입력 모듈과 온도 입력 모듈이 있습니다.

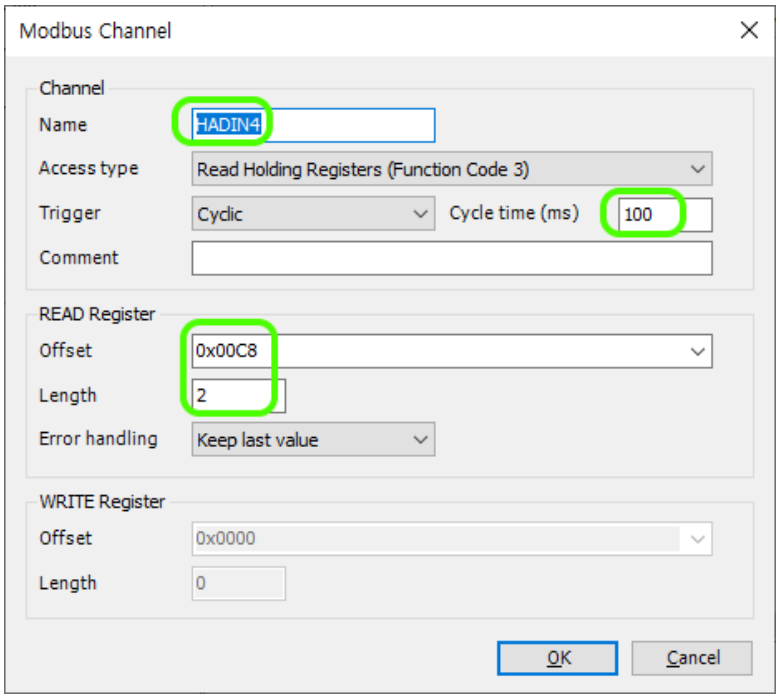
모델명	제품코드	설명	가격
MD-H485	M00101	485헤더모듈	16,500원
MD-DIDC8	M00203	DC입력 8점	38,500원
MD-DOSO8	M00206	DC 소스출력 8점	38,500원
MD-DOSI8	M00205	DC 싱크출력 8점	38,500원
MD-DORL8	M00204	릴레이출력 8점	49,500원
MD-ADIN4	M00201	AD입력 4채널 (13.3비트) 0~10V, 1~5V, 4~20mA	55,000원
MD-HADIN4	M00207	AD입력 4채널 (16.6비트) 0~10V, 1~5V, 4~20mA	77,000원
MD-THRT4	M00208	PT100옴 (RTD) 입력 4채널	55,000원
MD-DAOUT2	M00202	DA전압출력 2채널 (16비트) 0~10V, 0~5V	66,000원
MD-DAOUT2B	M00209	DA전류출력 2채널 (16비트) 4~20mA, 0~20mA	66,000원

테스트할 수 있게, 볼륨과 PT100 센서를 연결해 두었습니다.

CODESYS 쪽 MODBUS 설정쪽을 보겠습니다.



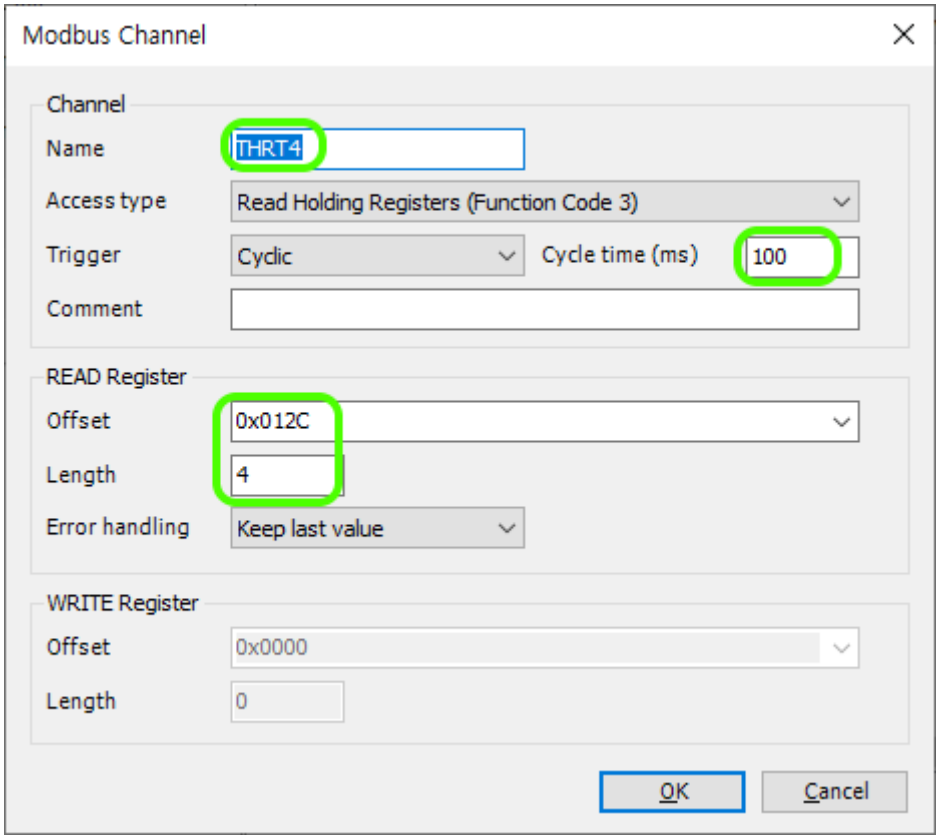
Slave 쪽 채널에 2 개를 더 추가했습니다. 먼저 HADIN4 내용을 보겠습니다. 이름은 HADIN4 로 하고 사이클간격은 100mS 로 했습니다. (아날로그 양은 갑자기 변하지 않으므로 빠르게 설정할 필요가 없습니다.)



Offset 값은 MODPORT 메뉴얼을 보고 200 을 입력했습니다. (200 을 입력하면 자동으로 16 진수인 0xC8 로 바뀜)

시작어드레스	제품명	지원 펑션코드
0-7	MD-DIDC8 디지털입력	1
100-103	MD-ADIN4 아날로그입력	3
200-207	MD-HADIN4 고해상도 아날로그 입력	3
300-303	MD-THRT4 RTD 온도입력	3

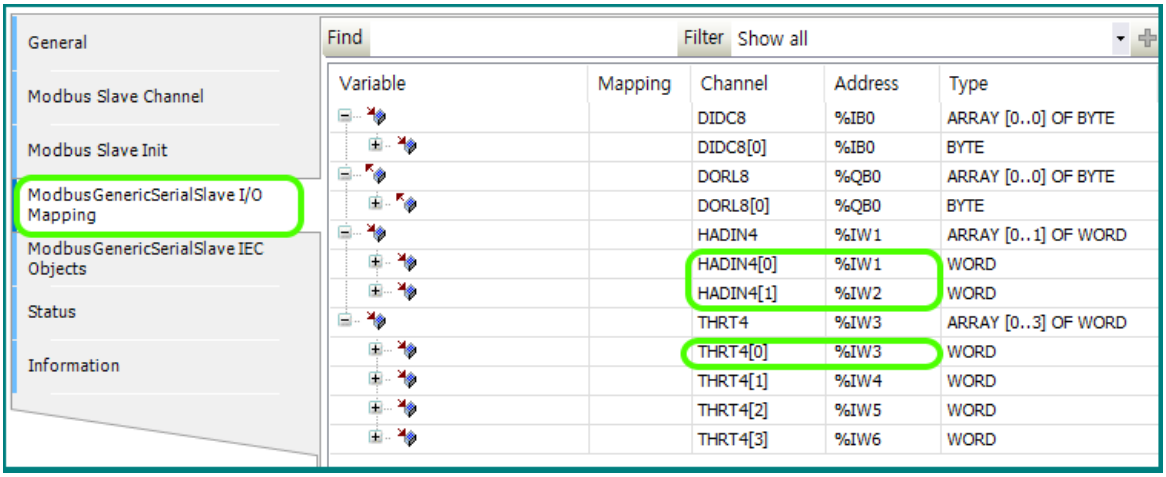
다음은 THRT4 쪽 채널설정입니다. 역시 사이클타임은 100mS 로 했습니다. 온도가 급작스럽게 변하지는 않으므로 이정도면 충분합니다.



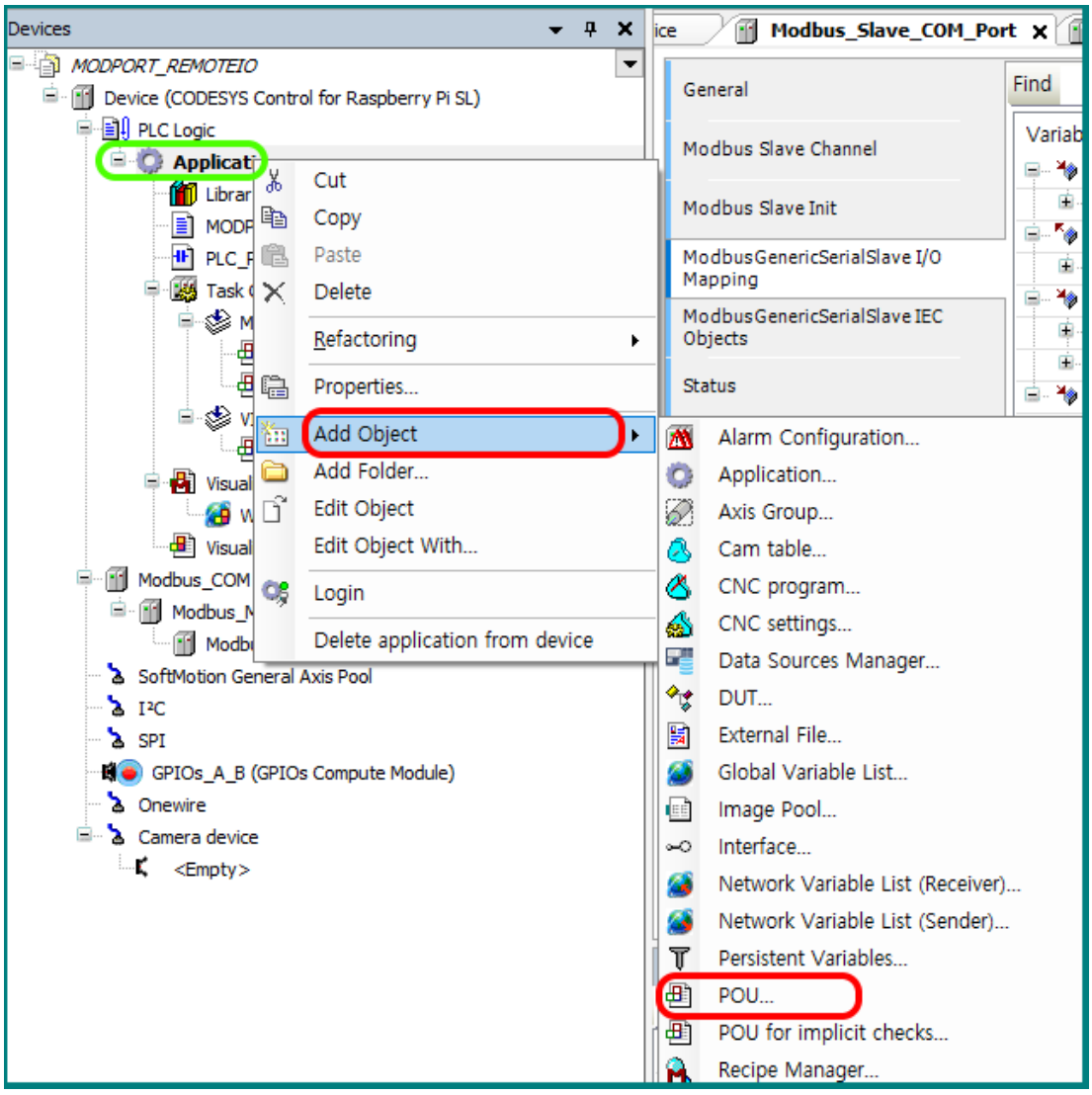
Offset 도 메뉴얼에 있는데로 300 으로 했습니다. (300 = 0x12C)

시작어드레스	제품명	지원 펑션코드
0-7	MD-DIDC8 디지털입력	1
100-103	MD-ADIN4 아날로그입력	3
200-207	MD-HADIN4 고해상도 아날로그 입력	3
300-303	MD-THRT4 RTD 온도입력	3

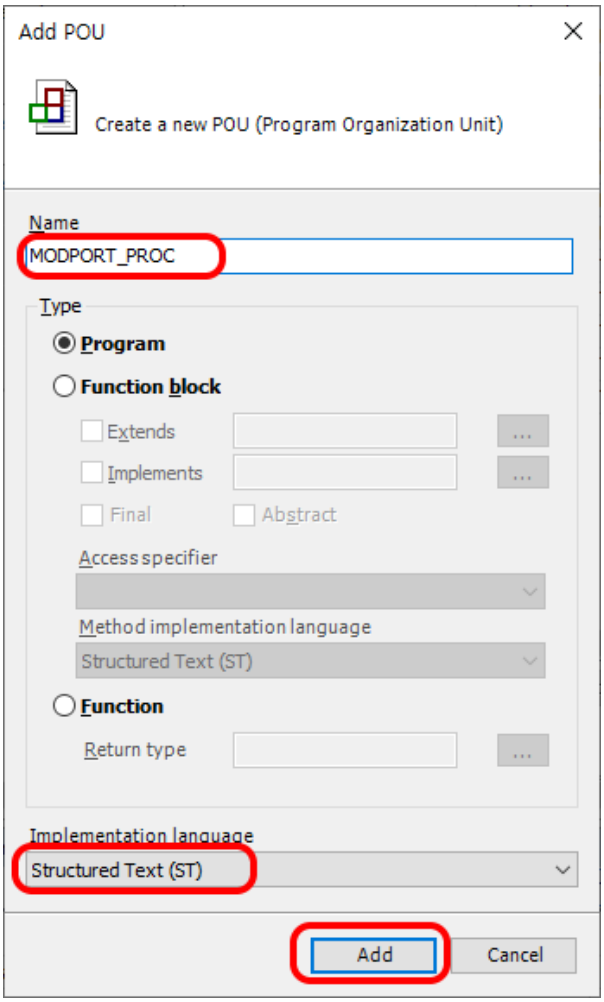
이번엔 프로그램 쪽을 보겠습니다. 관련 직접주소가 %IW1 부터 시작하는데, 이것을 어떤 변수에 넣어주어야 Visualization 에서 표시할 수 있습니다.



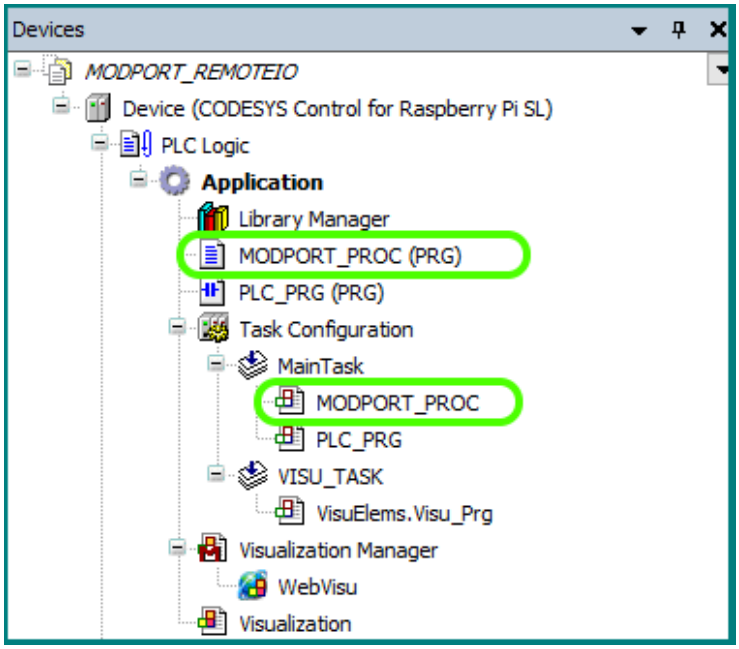
그래서 POU 를 하나 추가했습니다.



POU 에서 언어설정을 다르게 할 수도 있습니다. 여기에선 ST 언어로 설정했습니다.



그리고 이걸 MainTask 쪽에 넣었습니다. 이제 Main 실행시 방금 추가한 POU 도 같이 실행됩니다.



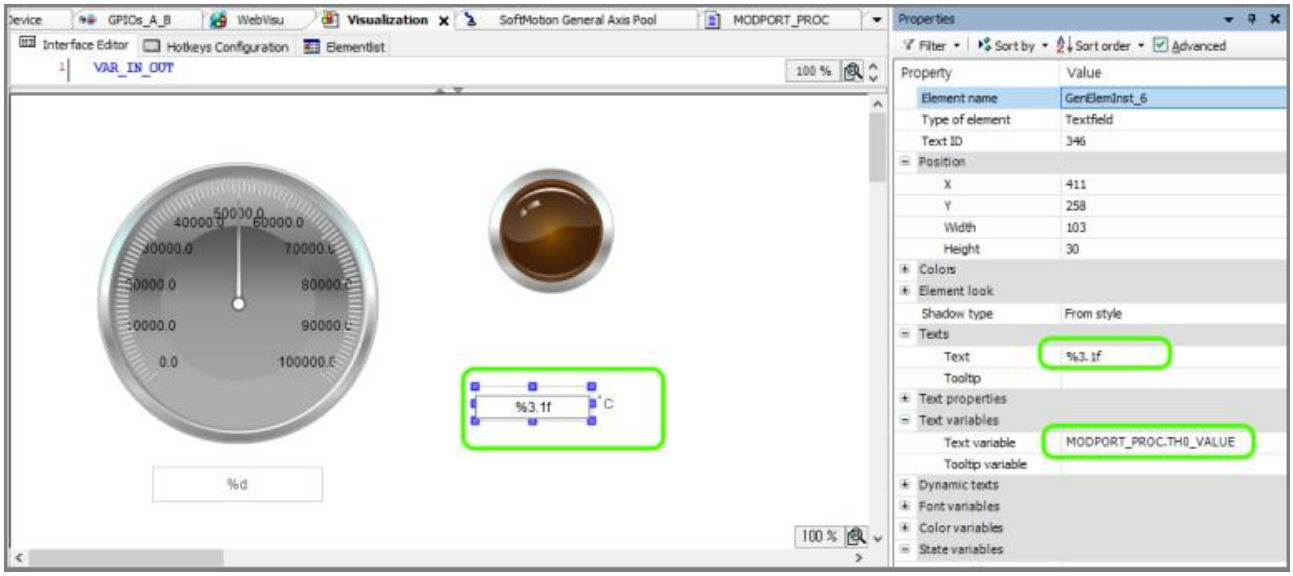
그리고 이 POU 안에 이렇게 코딩을 해서 넣었습니다. 별건 아니고 MODPORT 에서 읽어온 값은 직접주소로 저장되는데, 이걸 좀 가공(?)해서 변수에 할당한 것입니다.

```
PROGRAM MODPORT_PROC
VAR
    HAD0_VALUE : DWORD;
    TH0_VALUE : REAL;
    TH_ORG : WORD;
END_VAR
```

```
HAD0_VALUE := %IW1 * 65536 + %IW2;

TH_ORG := %IW3;
IF (TH_ORG AND 16#8000) = 16#8000 THEN
    TH_ORG := TH_ORG - 16#8000;
    TH0_VALUE := TH_ORG/10;
    TH0_VALUE := -TH0_VALUE;
ELSE
    TH0_VALUE := TH_ORG/10;
END_IF
```

그렇게 해서 HAD0_VALUE 와 TH0_VALUE 가 만들어 졌습니다. 이제 이걸 시각화(Visualization)에서 표시하도록 해보겠습니다. 화면을 이렇게 꾸며봤습니다.



실행시켜보겠습니다. 결과는 동영상으로 보세요.

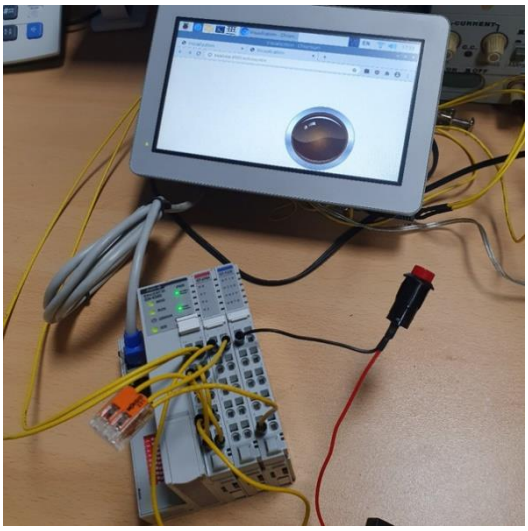
https://youtu.be/6p2eYwHHDk4

제 5 장. 이더켓

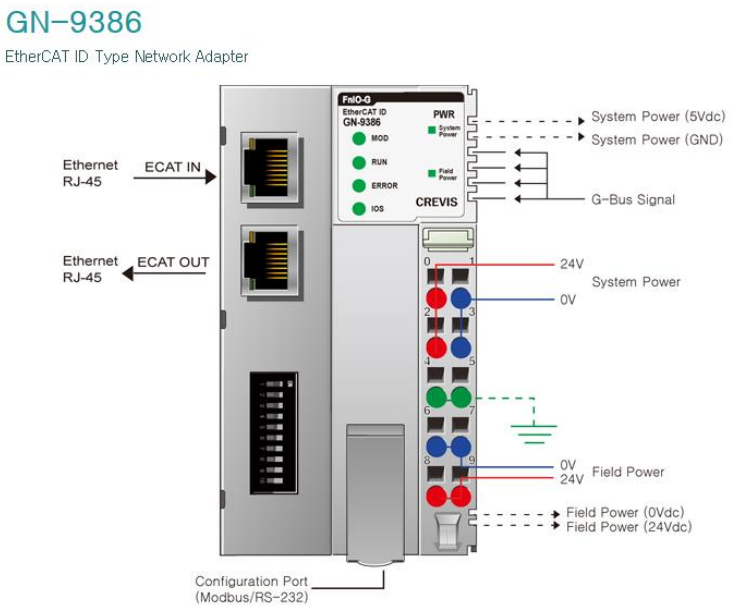
Ethercat 은 필드버스 통신 중 하나입니다.

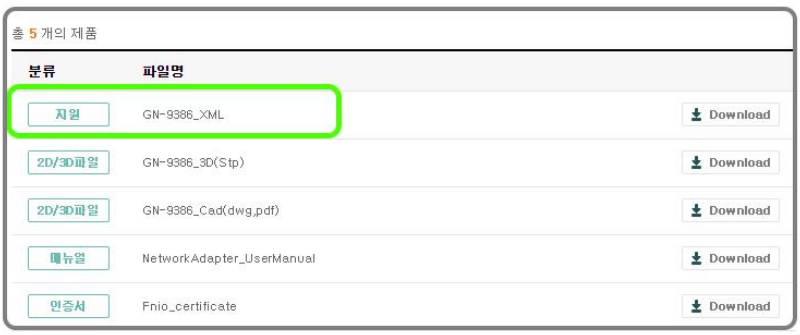
EHTERCAT 연결

ETHERCAT 연결을 해보겠습니다. 컴파일파이에 이더넷 포트가 하나밖에 없어서 WIFI 동글을 끼워서 인터넷선과 연결하고, 이더넷 포트는 ETHERCAT 모듈에 연결했습니다.

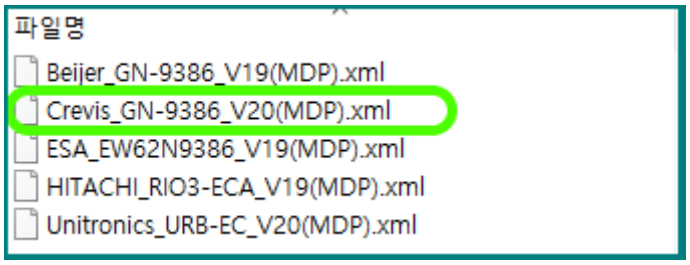


ETHERCAT 제품은 CREVIS 사의 GN-9386 을 선택했습니다. 해당 회사의 사이트에가서 I/O 설정용 파일인 XML 을 다운받았습니다.

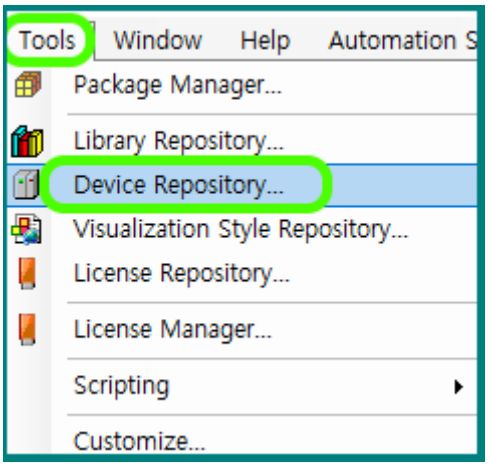




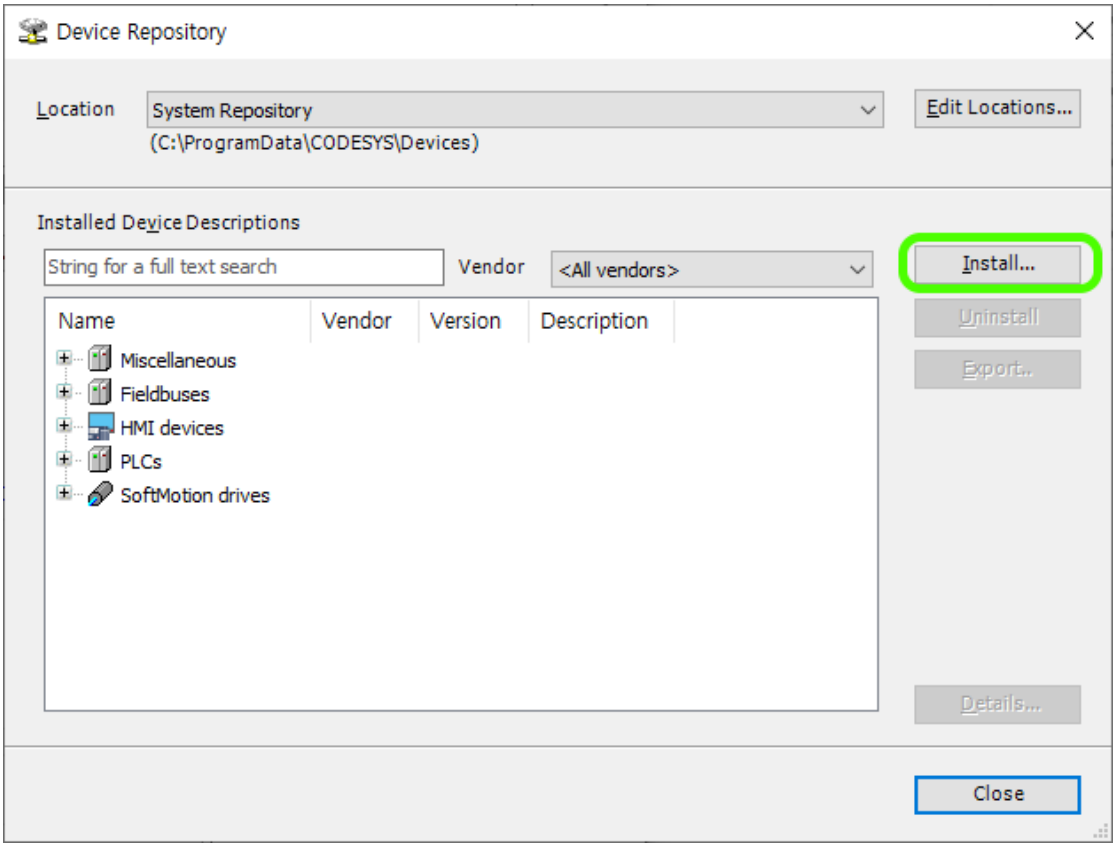
압축을 푸니 여러개가 있던데, 저는 동그라미 친거 하나만 꺼냈습니다.



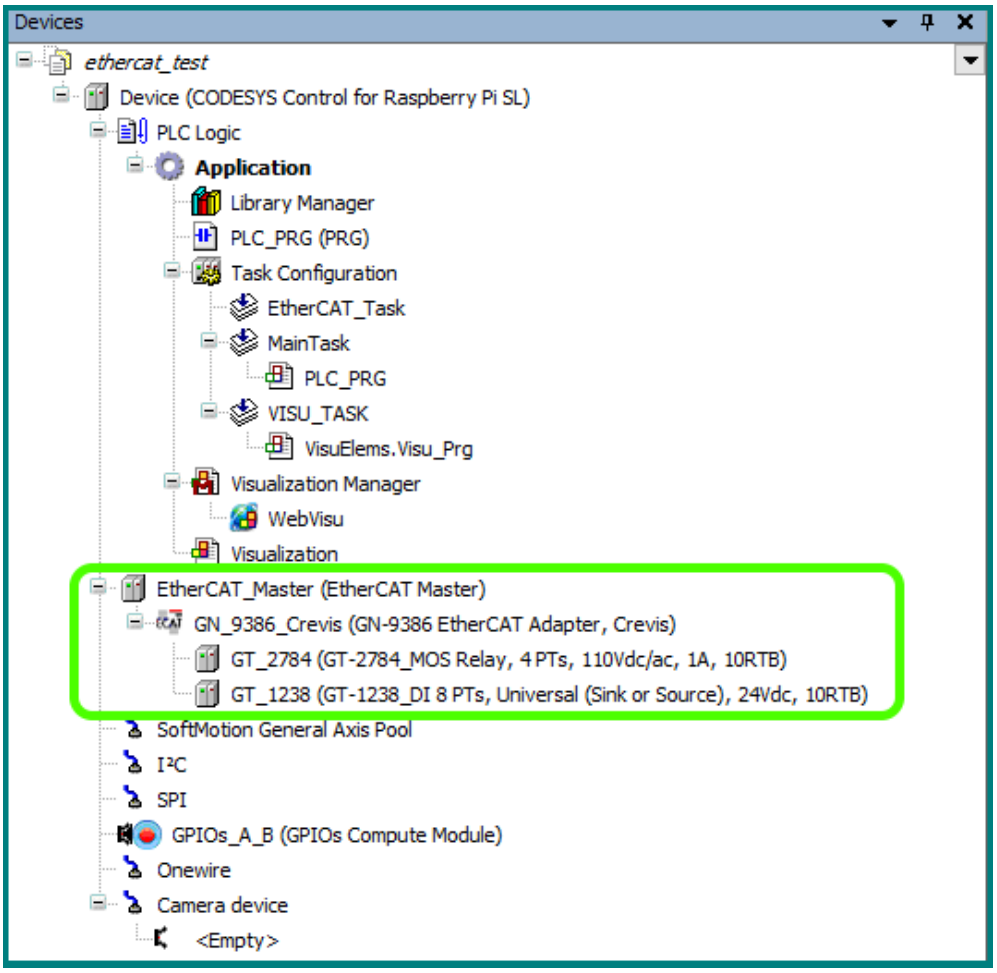
Toosl 메뉴에서 Device Repository 를 선택하고..



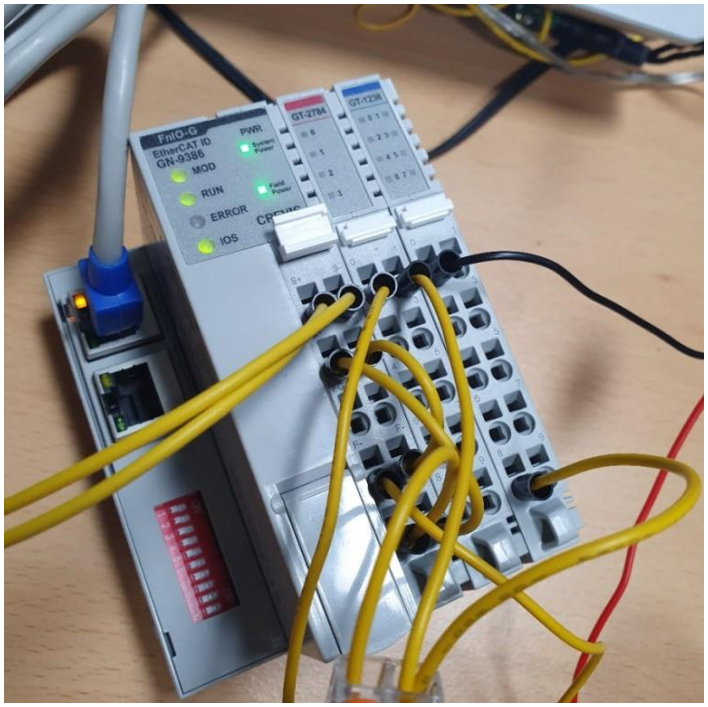
Install 을 누른뒤, 방금 다운받은 파일을 설치했습니다.



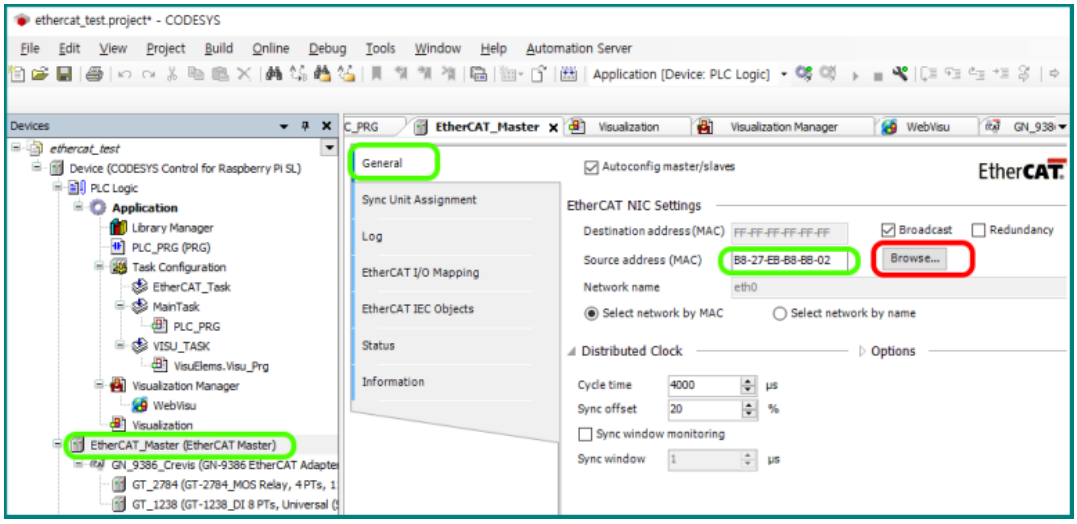
그리고 디바이스를 추가 해줘야겠죠. 아래 그림처럼 차례대로 추가했습니다.



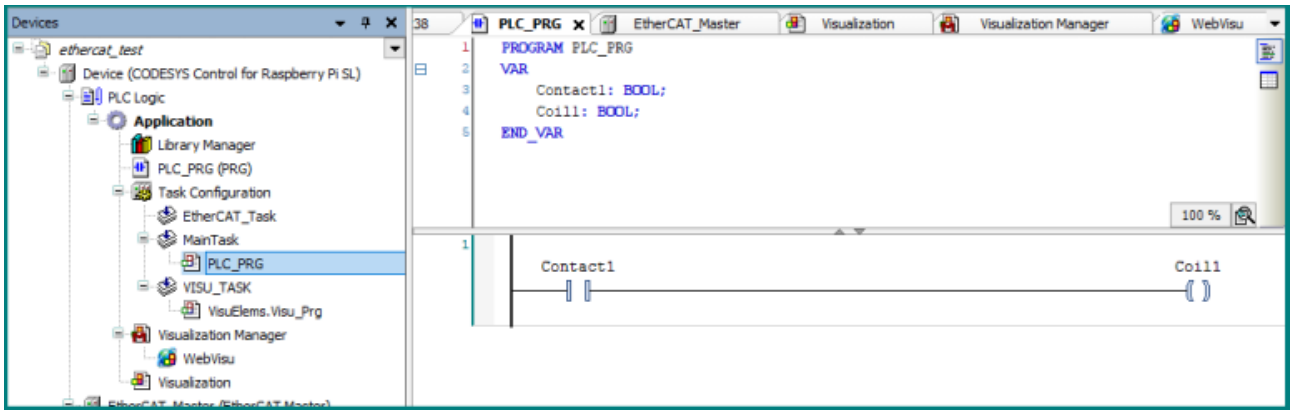
사진을 보면 9386 옆에 2784 와 1238 이 꼽혀있는데, 이 순서대로 Device 를 추가해 준것입니다.



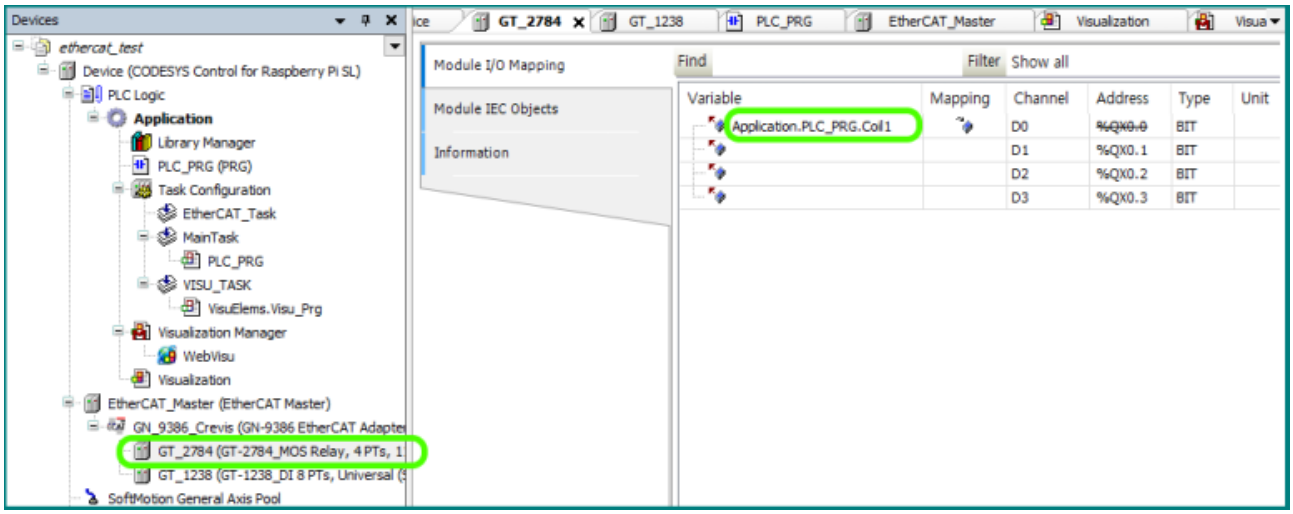
그 다음은 MAC 어드레스를 잡아줘야하는데, EtherCAT Master 에서 General 탭을 누른뒤, Browse 를 눌러주세요. 그럼 연결된거 한개가 달랑 뜹니다. 그걸 선택해주면 끝.



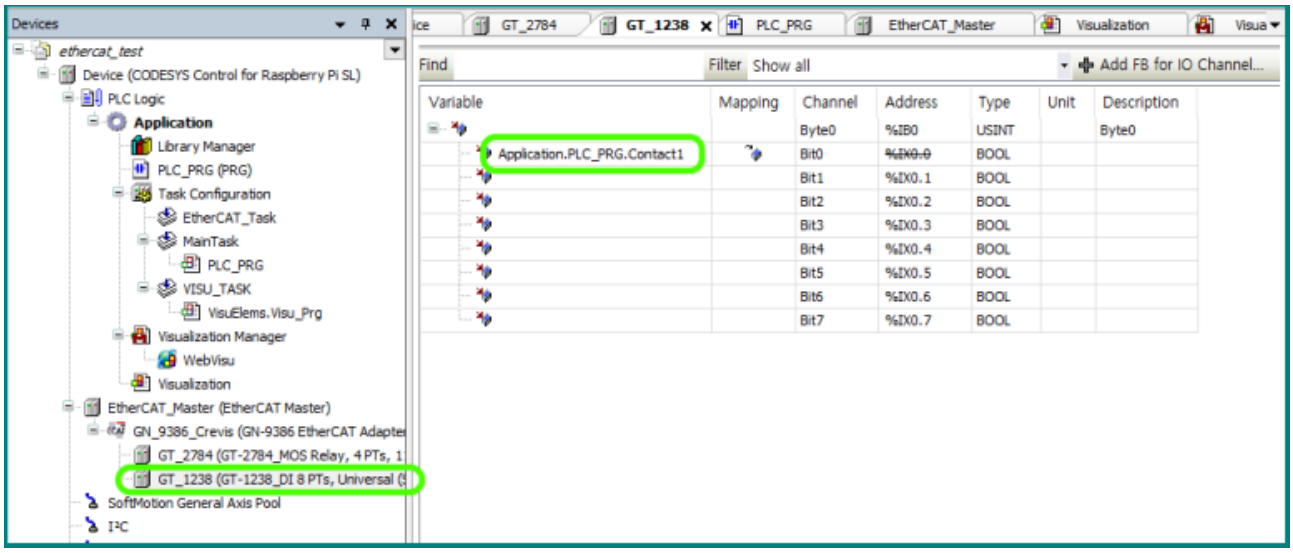
이더켓 모듈 설정은 다 된겁니다. 이제 프로그램을 짜 줘야 겠죠. 간단하게 이렇게 짰습니다. 한개 입력받아서 출력하는 레더입니다.



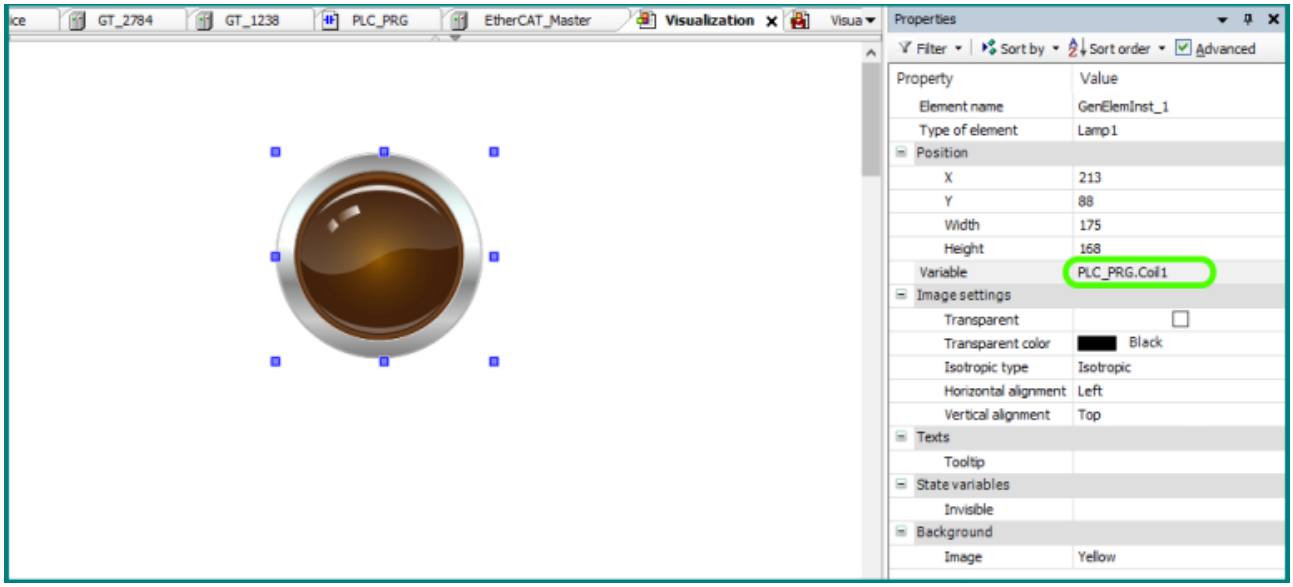
I/O 맵핑을 해줄 차례입니다. 보시는 것처럼 출력의 0 번 포트를 Coil1 과 연결 했습니다.



입력도 0 번포트를 Contact1 과 연결 했습니다.



비주얼라이제이션도 심심하니까 램프 하나 해줘야겠죠.



결과는 동영상에서 보시는 것처럼 잘 됩니다. 이더켓이라 반응이 좀 빠른편입니다. (응답속도 15~20mS 수준입니다. MODBUS-RTU 는 80mS~100mS 정도입니다.)

https://youtu.be/qhJOPQ9G_tc

마치며

저 역시도 CODESYS 를 공부하면서 느낀점이 있다면, "쉽다"입니다. 똑 같은 프로그램을 C#이나 Visual BASIC.NET 으로 구현하려면, 시간도 오래걸리고 디버깅에 쏟는 노력이 상당한데, CODESYS 는 그 반의 반도 안되는 노력과 시간으로 그만큼의 퀄리티를 갖는 프로그램을 만들어 낼 수 있는 툴입니다.

유일한 단점이라면 런타임 라이선스를 구입해야 한다는 점인데, 원래 CODESYS 는 PC 용 소프트웨어라서 이것 저것 다 합치면 한카피당 거의 100 만원 정도의 라이선스비용이 드는 소프트웨어였습니다.

그런데 몇 년전 라즈베리파이용 CODESYS 가 출시되면서 그 가격이 60 불로 책정되었습니다. 게다가 240 불 상당의 WebVisu 기능까지 포함하고 있습니다. 저희 회사의 컴파일 파이는 라즈베리기반의 산업용 PC 이기 때문에, 라즈베리용 CODESYS 가 아주 잘 돌아갑니다.

CODESYS 는 특히 이런분들에게 유용할 것 같습니다.

1. 그동안 C#이나 JAVA 등과 같은 PC 용 언어를 익히지 못해, PC 기반 소프트웨어를 작성할 수 없었던분.
2. 레더정도를 다룰 수 있는 기술력은 있지만, PC 언어는 어렵다고 느끼셨던 분
3. 인터넷, IOT, 클라우드와 같은 신기술을 도입하고 싶었지만 엄두가 안났던 분

모쪼록 이런 분들에게 이 책과 저희 제품인 ComfilePi 가 도움이 되었으면 하는 바램입니다.

자주 묻는 질문과 대답

Q. 타회사 SCADA 툴처럼 태그 수 제한이 있습니까? ..... 없습니다.
Q. 런타임 라이선스가 꼭 있어야 쓸 수 있습니까? .....아닙니다. 없어도 쓸수는 있습니다.
Q. 런타임 라이선스는 생산수량만큼 있어야 합니까? .....네 한대당 하나입니다.
Q 공부하기 위해서 컴파일파이를 꼭 사야합니까? ..... 아닙니다. PC 에서도 시뮬레이션으로 돌릴 수 있습니다.
Q. 런타임 라이선스 대량구매시 할인혜택은? .....있습니다. Store.codeys.com 에서 확인해보세요.
Q. 산업용 사용을 위해서 꼭 MC (멀티코어)를 써야 합니까? .....아닙니다. 싱글코어용을 써도 됩니다.