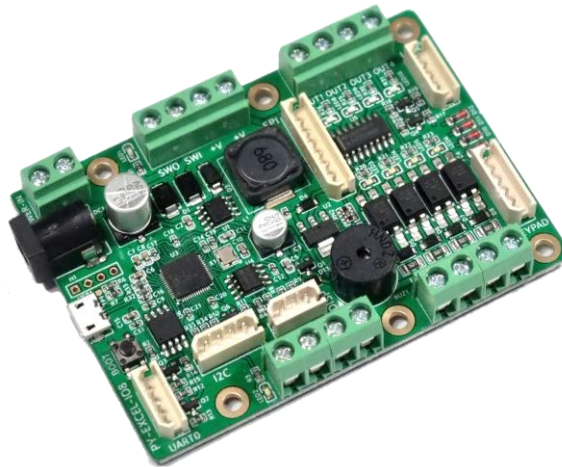


PY-EXCEL-IO8

실무용 excel기반 제어보드



Board Ver : REV1

목 차

1. 안전상의 주의 사항	5
2. 주요 특징	6
3. Excel 플랫폼	8
3-1 시작하기	8
3-2 Excel Micro Python 설정	9
3-2-1 설정1 (옵션-고급)	9
3-2-2 설정2 (옵션-언어 교정)	10
3-3 Excel Micro Python 프로그래밍	11
3-3-1 Excel에서 Micro Python 들여쓰기(Indent)	11
3-3-2 Excel 열(Column) 변환	12
3-3-3 Excel 셀 값 가져오기 (기본)	13
3-3-4 Excel 셀 값 가져오기 (응용)	13
3-3-5 Excel 주석 사용하기	14
3-3-6 Excel 수식 (기본)	15
3-3-7 Excel 수식 (응용)	16
3-3-8 Excel Sheet (main.py)	17
3-4 Excel VBA	18
3-4-1 Range(micropython)	18
3-5 Excel EEPROM 모드	20
3-6 다른 Excel 파일에 Sheet 복사	22
3-7 샘플 소스 코드 리스트	23
3-8 Windows OS, Excel 버전	23
3-9 Excel 단축키 리스트	24
4. Thonny IDE 개발 툴	26
4-1 시작하기	26
4-2 PY-EXCEL-IO8 보드 연결하기	27
4-2-1 Micro USB 케이블 연결	27
4-2-2 Thonny IDE에서 연결 설정	28

- 4-3 lib 폴더 사용하기 29
 - 4-3-1 lib 폴더 열기 29
 - 4-3-2 lib 폴더 내용 확인 30
 - 4-3-3 lib 폴더 파일 import (klcd.py) 31
- 5. px-downloader-v100.exe 32**
 - 5-1 시작하기 32
 - 5-2 PY-EXCEL-IO8 보드 연결 33
 - 5-2-1 Board Connection 33
 - 5-3 Excel 파일 실행 34
 - 5-3-1 Excel Open 34
 - 5-3-2 Stop PGM, Run PGM, Download 35
 - 5-3-3 px-downloader-v100.exe 디버깅 테스트 36
 - 5-3-4 Make Flowchart 37
 - 5-3-5 Flowchart 변환 구역 38
 - 5-3-6 Flowchart 도형 사용자 설정 39
 - 5-3-7 Flowchart 도형 사용자 설정에서 주석 추가하기 40
 - 5-3-8 Flowchart 도형 리스트1 41
 - 5-3-9 Flowchart 도형 리스트2 42
 - 5-3-10 Flowchart 도형 리스트3 43
 - 5-3-11 Micro Python To Excel 파일 변환 (*.py to *.xlsm) 44
 - 5-3-12 Micro Python To Excel 파일 변환 (samplevba.xlsm) 45
- 6. Arduino IDE 개발 툴 46**
 - 6-1 시작하기 46
 - 6-2 PY-EXCEL-IO8 보드 연결하기 47
 - 6-2-1 Arduino IDE 보드 추가 47
 - 6-2-2 blink.ino 코드 작성, 업로드 49
- 7. Visual Studio Code 개발 툴 50**
 - 7-1 시작하기 50


- 7-2 Visual Studio Code에서 PlatformIO IDE(Extension 설치) 51
 - 7-2-1 PlatformIO IDE 설치 51
 - 7-2-2 PlatformIO IDE blink 프로젝트 생성 52
 - 7-2-3 blink 프로젝트 소스 코드, 설정 파일 53
- 7-3 Build 완료 파일 저장 54
- 8. PY-EXCEL-IO8 제어보드 55**
 - 8-1 보드 사양 55
 - 8-1-1 BOOT SW (Micro Python 펌웨어) 56
 - 8-2 PY-EXCEL-IO8 하드웨어 57
 - 8-2-1 전체 PERIPHERAL CONNECTION 배치도 57
 - 8-2-2 POWER SUPPLY 58
 - 8-2-3 UART0,1 58
 - 8-2-4 I2C 59
 - 8-2-5 SPI 59
 - 8-2-6 KEYPAD 60
 - 8-2-7 DISP 60
 - 8-3 PY-EXCEL-IO8 회로도 61
 - 8-3-1 MCU 회로도 61
 - 8-3-2 INPUT 회로도 62
 - 8-3-3 OUTPUT 회로도 63
 - 8-4 PY-EXCEL-IO8 DIMENSIONS 64
- 9. 개정 이력 65**

※ **사용하시기 전에** ※


- PY-EXCEL-IO8을 구입해 주셔서 대단히 감사합니다.
- PY-EXCEL-IO8은 Excel을 기반으로 프로그래밍할 수 있는 새로운 형태의 제어보드입니다.
- 산업용 하드웨어 제어가 가능하기에 안전상의 주의사항, 취급 방법을 읽으신 후 사용하여 주십시오.

1. 안전상의 주의 사항

◆ **설치**

 주의	<p>제품을 취급할 장소에서는 청결을 유지하여 주십시오. 금속조각, 케이블 찌꺼기 등은 고장의 원인이 됩니다.</p> <p>다른 부품과 밀폐된 공간에 설치할 경우, 냉각팬 등을 설치하시어 50°C 이하가 되도록 하십시오.</p>
--	---

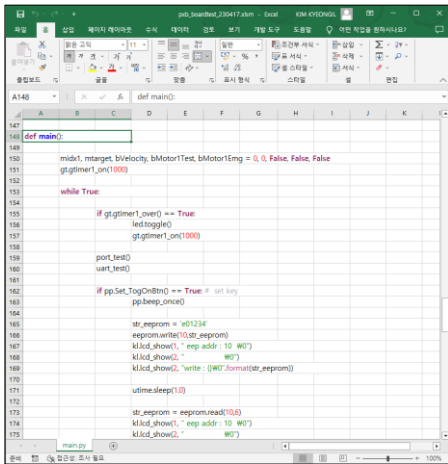
◆ **배선**

 주의	<p>제품의 전원 입력 전압은 정격 범위를 반드시 지켜 주십시오. 오동작 및 고장의 원인이 됩니다.</p> <p>접속은 배선도에 따라 극성에 유의하여 주십시오. 잘못된 배선은 오동작 및 고장의 원인이 됩니다.</p> <p>입력 전원이 OFF 되어 있는 것을 확인한 후 작업해 주십시오. 화재 및 부상을 초래할 수 있습니다.</p>
---	--

2. 주요 특징

1 Excel 기반 프로그래밍

PY-EXCEL-IO8는 Excel을 기반으로 프로그래밍하고 배포할 수 있는 새로운 제어보드입니다. Excel은 전 세계 약 7억 5천만 명 이상이 사용하는 가장 보편화된 소프트웨어입니다. Excel에서 수정, 배포할 수 있음에 따라 프로그래머(개발자)와 오퍼레이터(사용자, End User) 사이의 소통에서 발생하는 비효율을 개선하고 특히 현장에 따라 달라질 수 있는 조건을 쉽게 반영할 수 있습니다.

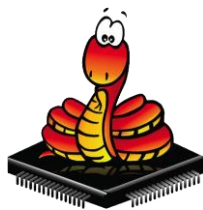


Micro Python for Excel 사용 예

2 Micro Python 언어

Micro Python 프로그래밍 언어는 Python 3를 기반으로 마이크로프로세서용으로 개발된 언어입니다.

Micro Python은 간결한 문법으로 가독성을 높이고 작은 양의 코드로 더 많은 작업을 수행할 수 있으며 복잡한 구문으로 인한 오류 발생을 줄여 빠른 개발이 가능합니다.



Micro Python

3 EEPROM 모드 지원

EEPROM 모드는 Excel 파일에서 어드레스와 데이터만 사용하여 PY-EXCEL-IO8에 내장된 EEPROM 값을 변경하는 기능입니다. 소스 코드를 확인하지 않고 내부 값을 변경하기 때문에 프로그래머가 아닌 사용자 혹은 오퍼레이터에게 유용하게 사용될 수 있습니다.

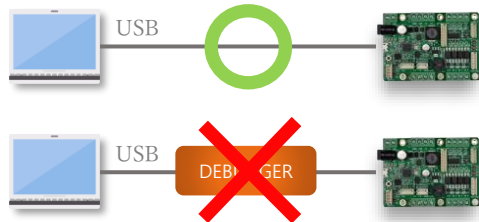
px-downloader-v1.00.exe를 통하여 PY-EXCEL-IO8 제어보드에 다운로드합니다.

	A	B	C	D
1				
2	eeprom write command			
3				
4	어드레스 10번 123456 저장			
5	ADDR	10	123456	

EEPROM 모드 사용 예

4 소프트웨어 디버깅

PY-EXCEL-IO8는 소프트웨어 디버깅을 지원합니다. Micro Python 용 디바이스는 공통으로 REPL(Read Evaluate Print Loop)방식으로 프로그램을 실행하고 테스트할 수 있습니다. 이 기능은 현존하는 가장 쉬운 프로그래밍 실행 방식입니다.



5 Excel 데이터 동시 저장

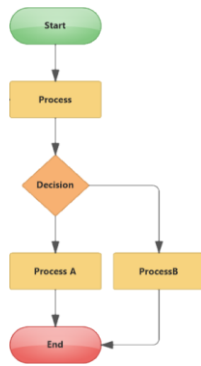
Excel의 여러 Sheet에 소스 코드를 포함한 프로젝트에 사용되는 리스트, 수식 등 각종 데이터를 동시에 저장할 수 있습니다. 생산성을 한 단계 업그레이드하세요.



Excel Sheet 활용 예

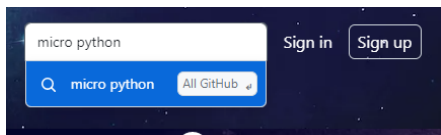
6 Flow Chart 자동 생성

Micro Python을 기반으로 작성된 프로그램에서 파워포인트용 Flow Chart를 생성할 수 있습니다. 원하는 도형과 색상도 사용자가 지정할 수 있습니다.



7 방대한 Python 검색 자료

Github에서 Micro Python을 검색 시 약 7,300개 이상의 결과를 확인할 수 있습니다. Github, 블로그 등 다양한 채널을 통하여 샘플 코드를 확보할 수 있습니다.



Github에서 micro python 검색

8 EASY CONTROL 확장보드

저희 라온컴퍼니는 제어보드와 쉽게 연동 작동이 가능한 다양한 기능의 확장보드를 보유하고 있고 지속하여 개발합니다.

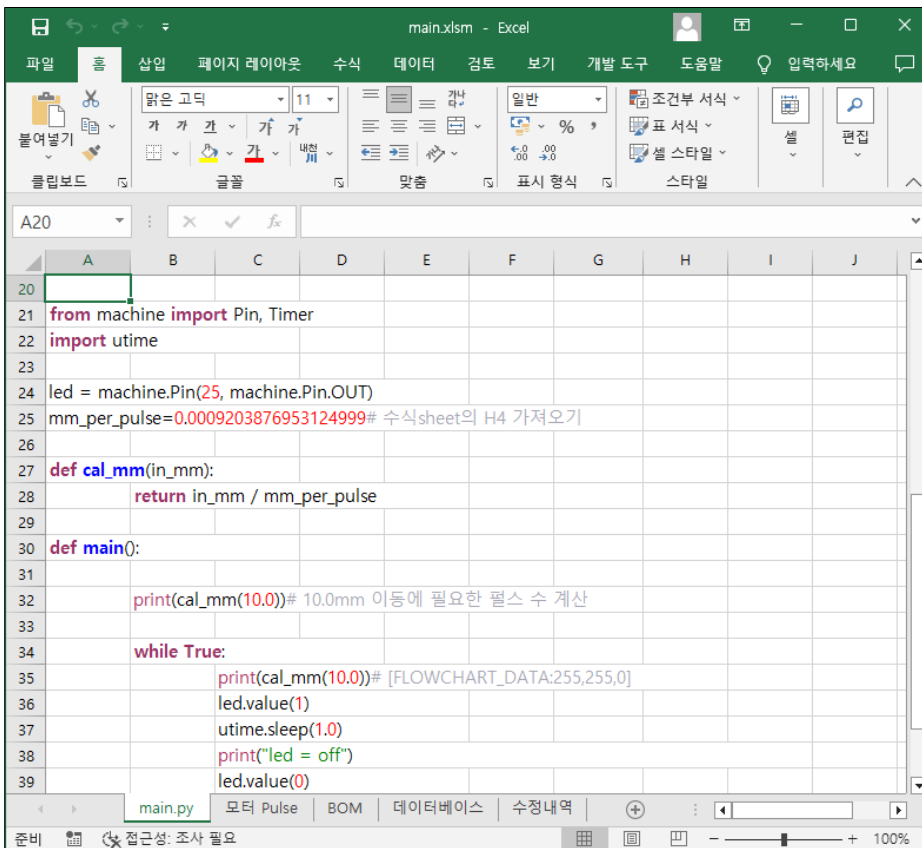
제품 사진	모델명
	LC-KLCD-R2
	LC-FND4-056-RV1
	XY-EXTIN-NPN8-RV1
	XY-EXTOUT-TR8-RV2
	XY-EXTOUT-RLY248-RV1
	XY-STEP-TMC5160A (SPI)
	XY-CONV-DRIVER-RV1

3. Excel 플랫폼

3-1. 시작하기

Excel은 단일 소프트웨어로는 전 세계에서 가장 많이 사용되는 생산성 TOOL입니다. 최근 AI를 활용한 생산성 향상 소프트웨어 (MS-copilot, Google-Bard AI)도 등장하고 있습니다. 이러한 AI를 결합한 생산성 도구는 실무의 모습을 완전히 바꾸게 될 것입니다. PY-EXCEL-IO8 제어보드는 Excel에서 프로그래밍하고 수정, 배포가 가능한 최초의 Circuit Board로 혁신적인 실무 프로그래밍 방식으로 높은 생산성 향상을 실현하였습니다. 이 장에서는 Micro Python 프로그래밍을 위한 Excel 설정 방법과 수식 사용, Excel VBA, EEPROM 모드에 대해서 알아보겠습니다.

[Figure 3-1] Excel Micro Python 프로그래밍



기존 Excel 업무와 프로그래밍의 결합은 사용자에게 새로운 경험을 제공합니다. 복잡한 계산 수식을 작성하거나 평균값을 소스 코드와 연결하고 BOM 리스트나 매뉴얼을 Excel 한 파일에 관리 등 혁신적인 작업이 가능합니다. 또한, 많은 사용자들 기반으로 국내, 해외 사용자까지 한 번에 지원할 수 있습니다.

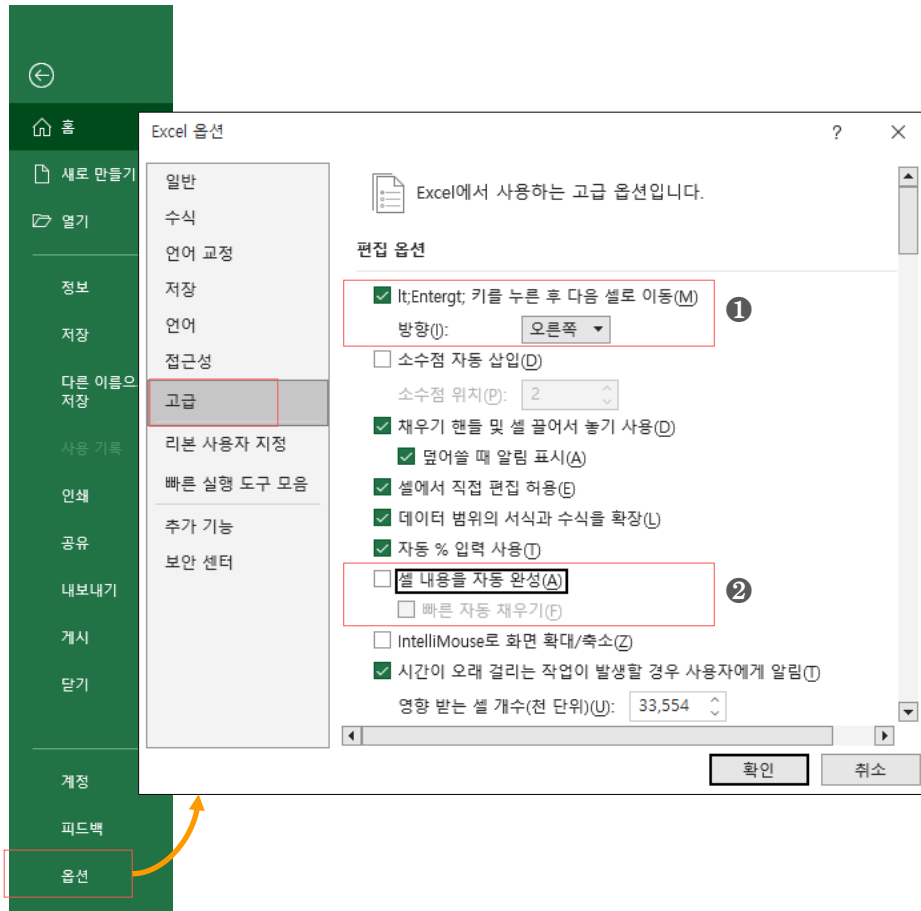
3-2. Excel Micro Python 설정

Excel에서 Micro Python을 프로그래밍하기 전 몇 가지 설정을 변경해야 합니다. Excel에서 설정 변경에 대해 알아보겠습니다.

3-2-1. 설정1 (옵션-고급)

Excel의 좌측 파일 메뉴에서 옵션을 선택하여 “Excel 옵션” 창을 엽니다. Excel 옵션의 고급 항목을 선택하고 ① ‘<Enter>’ 키를 누른 후 다음 셀로 이동(M)’에서 방향을 “오른쪽”을 선택, ② ‘셀 내용을 자동 완성(A)’ 기능을 해제하여 코딩 중 Enter 키를 누르면 오른쪽 셀로 이동하게 하고 자동으로 문자가 변하지 않게 합니다.

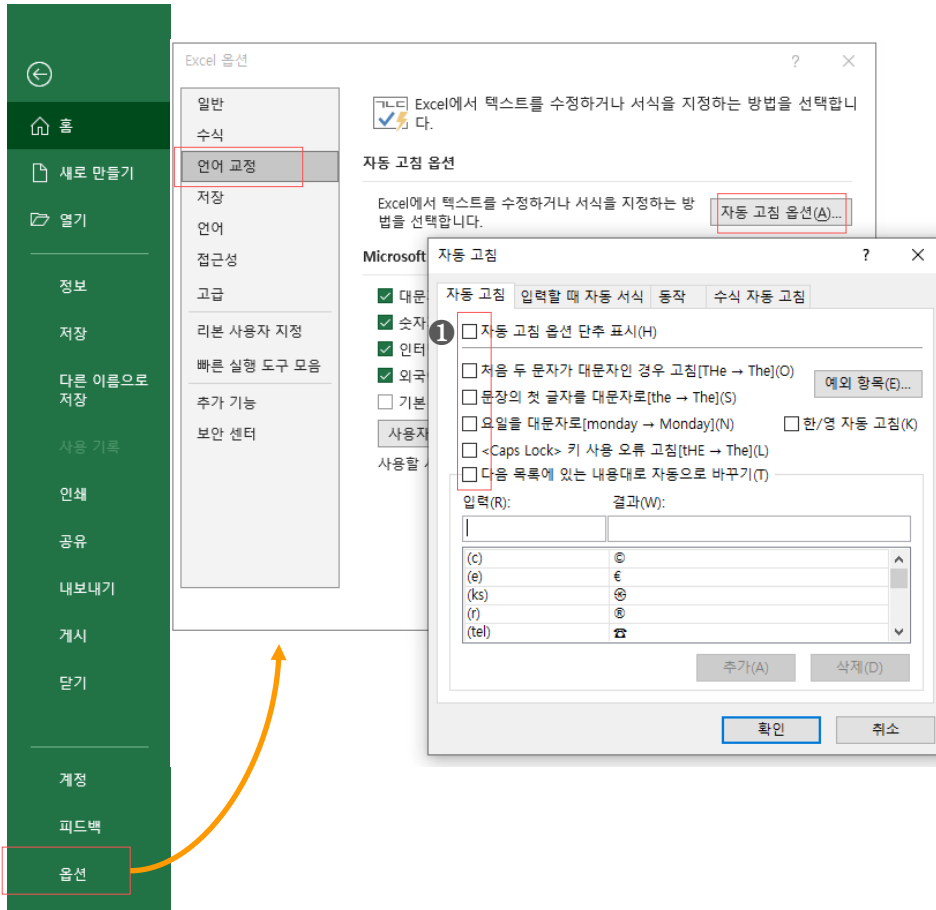
[Figure 3-2] Excel Micro Python 프로그래밍 사용하기 고급 설정



3-2-2. 설정2 (옵션-언어 교정)

Excel 옵션의 언어 교정 항목을 선택하고 자동 고침 옵션(A)을 선택합니다. ❶ 자동 고침 탭에서 '자동 고침 옵션 단추 표시(H)'를 포함한 아래 체크 박스를 모두 해제하여 코딩 중 문자가 변하지 않게 합니다. 이제 기본적인 설정이 완료되었습니다. 다음 페이지부터 Micro Python이 Excel에서 어떻게 표시되는지 알아보겠습니다.

[Figure 3-3] Excel Micro Python 프로그래밍 사용하기 언어교정 설정



3-3. Excel Micro Python 프로그래밍

Excel 플랫폼에서 Micro Python 코드가 Thonny IDE, Excel에서 표현되는 방식과 주요 특징을 알아봅니다. Excel에 표시되는 Micro Python코드는 px-downloader-V1.00.exe에서 *.py를 *.xism파일로 변환하였습니다.

※ 이 Datasheet는 Excel에서 사용법을 설명하고 있습니다. Python 문법은 별도 도서, 자료를 이용해 주시기를 바랍니다.

3-3-1. Excel에서 Micro Python 들여쓰기(Indent)

[Figure 3-4] Thonny IDE 들여쓰기 (blink.py)

```

6  def main():
7
8      while True:
9          print("led = on")
10         led.value(1)
    
```

[Figure 3-5] Excel 들여쓰기 (blink.xism)

6	def main():								
7									
8	while True:								
9		print("led = on")							
10	들여쓰기 구간	led.value(1)							

들여쓰기(Indent)는 Python 언어의 주요 특징입니다. Micro Python에서도 동일하며 들여쓰기는 공백 4칸을 의미합니다. Thonny IDE에서는 Tab 버튼으로 공백 4칸 삽입할 수 있고 Excel에서 열(Column) 1칸은 들여쓰기 1칸을 의미합니다. [Figure 3-4]는 Thonny IDE, [Figure 3-5]은 Excel에서 blink.py의 들여쓰기를 보여주고 있습니다.

참고 [Figure 3-6] Thonny IDE 들여쓰기 에러 (invalid syntax)

```

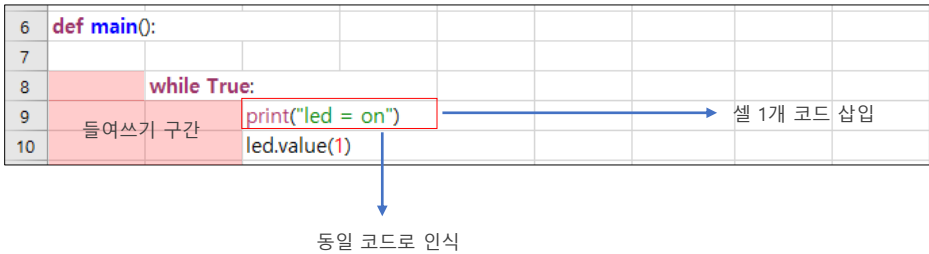
6  def main():
7
8      while True:
9          print("led = on")
10         led.value(1)
    
```

[Figure 3-6]는 Tab(0x09)으로 들여쓰기가 삽입된 경우입니다. 이 때, Thonny IDE에서 **"SyntaxError: invalid syntax"** 에러가 발생합니다. 다른 프로그램에서 코드를 가져올 경우 들여쓰기를 확인하지 않으면 에러가 발생할 수 있습니다. Thonny IDE에서는 Tab 키를 누르면 공백 4칸(space 0x20 * 4ea)이 삽입됩니다.

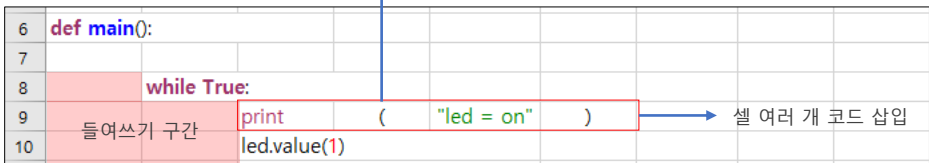
3-3-2. Excel 열(Column) 변환

Excel에서 Micro Python을 코딩할 때 들여쓰기 1칸은 열(Column) 1칸을 의미합니다. [Figure 3-7]에서 보는 바와 같이 A열에서 시작하여 문자가 존재하는 열 이전까지 들여쓰기의 구간입니다. 문자가 있는 셀부터 코드의 시작이고 우측 셀은 모두 코드로 변환됩니다. 즉, 여러 개의 셀을 사용하여 최종적으로 1개 라인을 작성할 수 있습니다. 이 점은 Excel Micro Python의 주요 특징 중 하나로 셀 값 가져오기를 이용하여 계산된 값, 또는 다른 시트의 값을 가져오는 것이 가능합니다.

[Figure 3-7] Excel 열 1개 사용 (blink.xlsm)



[Figure 3-8] Excel 열 여러 개 사용



[Figure 3-7]은 셀 1개에 코드가 삽입되어 있고, [Figure 3-8]은 셀 여러 개에 코드가 삽입되어 있습니다. Excel에서 열(column)이 사용되는 방법을 보여주고 있고 모두 동일하게 동작하는 코드입니다.

3-3-3. Excel 셀 값 가져오기 (기본)

Excel에서 Micro Python을 코딩할 때 셀 값 가져오기는 주요 특징 중 하나입니다. 이 기능은 매우 유용하게 사용될 수 있습니다. 계산된 수식 값, 리스트 평균값을 가져오거나 다른 시트의 내용을 참조하는 것이 가능하여 프로젝트에 사용되는 데이터를 유기적으로 결합할 수 있습니다. [Figure 3-9]는 [Figure 3-8]에서 셀 여러 개로 코드 삽입한 내용에 셀 값 가져오기를 사용해 보았습니다.

[Figure 3-9] Excel Cell 값 가져오기 사용


6	<code>def main():</code>						
7	<code>#</code>	<code>led on string</code>	<code>:</code>	<code>"led = on"</code>			
8		<code>while True:</code>					
9		<code>print</code>	<code>(</code>	<code>"led = on"</code>	<code>)</code>		
10		<code>led.value(1)</code>					

Cell 값 가져오기

3-3-4. Excel 셀 값 가져오기 (응용)

다음은 셀 값 가져오기를 응용하여 실제 프로젝트에 사용되는 방법을 설명합니다. [Figure 3-10]은 BOM 시트에 파트 리스트, 부품 이름을 함수 이름으로 지정합니다. [Figure 3-11]는 BOM 시트에서 지정한 함수인 "PRESS_CYL_ON()"을 main.py 시트로 가져와서 실제 코드에 적용하는 방법을 보여줍니다.

[Figure 3-10] BOM시트의 함수이름 지정, 셀 값 가져오기

No (번호)	Name (이름)	Funtion (함수)	Purpose (목적)	Vendor (제작사)	Part Name	Picture (사진)	Qty (수량)
5	1	PRESS_CYL_ON()	커버 압착 실린더	SMC	CDQ2A32		1
6		PRESS_CYL_OFF()					

[Figure 3-11] main.py시트 함수 이름 가져오기

```

31 def PRESS_CYL_ON() :
32     pp.Output1_On()
33
34 def PRESS_CYL_OFF() :
35     pp.Output1_Off()
36
37 pp = pyports.Ports()
38
39 def main():
40
41     PRESS_CYL_ON()
    
```

[Figure 3-11]에서 main.py시트의 31번 라인에서 함수정의 "def PRESS_CYL_ON();"는 여러 개의 셀로 나누어 작성되어 있음을 유의하시기 바랍니다. def와 PRESS_CYL_ON(): 사이에 " "(space)가 포함 되어 있습니다.

3-3-5. Excel 주석 사용하기

Excel Micro Python에서 주석 처리를 알아보겠습니다. 주석은 크게 2가지로 지정할 수 있습니다. 라인 전체 주석 처리, 라인 뒷부분 주석 처리입니다. [Figure 3-12]는 라인 전체 주석으로 A셀에 #을 삽입하여 뒤에 오는 모든 셀을 주석 처리하는 것입니다. [Figure 3-13]은 라인 뒷부분 주석이고 수행할 코드가 있는 경우에 사용합니다. 두 가지 모두 #을 삽입 후 오른쪽으로 따라오는 셀은 모두 주석 처리됩니다.

[Figure 3-12] 전체 라인 주석

6	<code>def main():</code>								
7	<code># led on string</code>	<code>:</code>	<code>"led = on"</code>						라인 전체 주석
8	<code>while True:</code>								
9	<code>print</code>	<code>(</code>	<code>"led = on"</code>	<code>)</code>					
10	<code>led.value(1)</code>								

[Figure 3-13] 라인의 뒷부분 주석 처리

6	<code>def main():</code>								
7									
8	<code>while True:</code>								
9	<code>print("led = on")</code>					<code># led on</code>			
10	<code>led.value(1)</code>								

뒤에 셀 모두 주석

주의 [Figure 3-14] 블록 주석 처리 (Excel 지원 안 됨)

```

6 def main():
7     ...
8     while True:
9         print("led = on")
10        led.value(1)
11        utime.sleep(1.0)
12        print("led = off")
13        led.value(0)
14        utime.sleep(1.0)
15    ...
16 main()
    
```

[Figure 3-14]은 Thonny IDE에서 블록 주석이 사용되는 방법을 보여줍니다. '''(작은 따옴표 3개)로 블록 주석이 시작되고 '''(작은 따옴표 3개)로 블록 주석이 끝납니다. Excel에서는 '''(작은 따옴표 3개)를 삽입할 수 없기 때문에 사용할 수 없습니다.

3-3-6. Excel 수식 (기본)

Excel Micro Python에서 또 다른 주요 특징은 Excel의 수식을 프로그래밍에 적용할 수 있다는 것입니다. 이 기능은 용량이 PC 대비 상대적으로 작은 마이크로프로세서에서 더 유용합니다. 메모리를 많이 차지하는 연산 데이터를 수식으로 처리 후 결과값을 사용할 수 있습니다. 단, 프로그램 중에 생성되는 데이터의 연산은 Excel 수식을 대신해 Micro Python의 코드로 연산해야 합니다. [Figure 3-15], [Figure 3-16]에서 수식이 사용되는 기본 방식을 설명합니다.

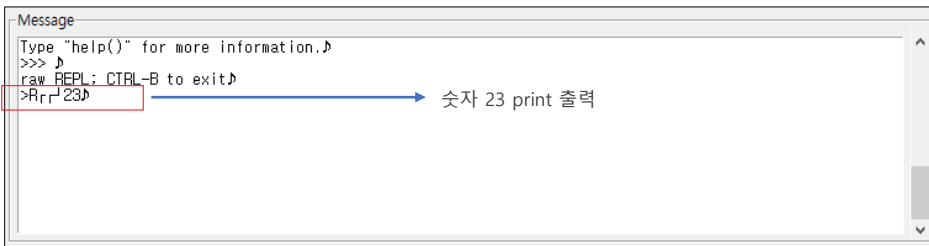
[Figure 3-15] Excel 수식 기본 사용하기

10	def main():						
11							
12	print	=G12+H12	# 수식	11	12		
13							
14	while True:					셀 G12, H12	

[Figure 3-16] Excel 수식 기본 사용하기 결과

10	def main():						
11							
12	print	(23)	# 수식	11	12
13							
14	while True:						

[Figure 3-17] px-downloader-v1.00.exe에서 print 확인



[Figure 3-17]은 PY-EXCEL-IO8에서 코드가 실행되고 px-downloader-v1.00.exe의 Message창에 보드에서 출력되는 데이터(23)를 보여줍니다.

3-3-7. Excel 수식 (응용)

Excel 수식을 응용하여 좀 더 복잡한 기능을 구현해 보겠습니다. 스텝 모터와 타이밍 풀리를 구동한다고 가정하고 Excel의 수식을 사용해 보겠습니다. 타이밍 풀리는 지름이 15 ϕ 이고 1회전 시 47.12385mm를 이동합니다. 스텝 모터는 가장 일반적인 스텝 각 1.8 $^{\circ}$ 로 지정합니다. 이 스텝 모터는 200 스텝 시 1회전(360 $^{\circ}$)하게 됩니다. 여기서 모터 드라이버 (XY-STEP-TMC5160A-SPI)의 256분해능을 사용 시 1회전에 사용되는 펄스의 수는 51200(200*256)이 되고 펄스 당 0.00092039mm를 이동할 것입니다. [Figure 3-18]은 수식 시트를 생성 후 펄스 당 이동 거리를 계산하기 위한 수식을 작성한 것입니다. [Figure 3-19]는 작성한 수식을 main.py 시트로 가져온 후 10.0mm 이동 시 필요한 펄스의 수를 출력하는 프로그램입니다.



[Figure 3-18] 수식 시트에 펄스 당 이동 거리 계산

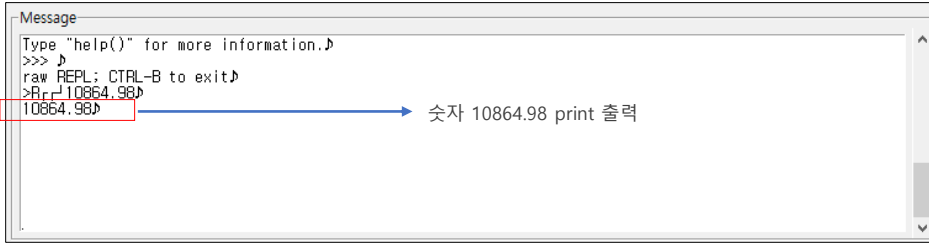
	A	B	C	D	E	F	G	H	I
1									
2		풀리			스텝 모터				
3	지름	파이	1회전 길이	스텝각	스텝1회전	드라이버 분해능	펄스1회전	펄스 (mm)	
4	15	3.14159	47.12385	1.8	200	256	51200	0.00092039	
5									

[Figure 3-19] main.py 시트에 펄스 당 이동(mm) 값 가져오기

```

5 mm_per_pulse = 0.0009204 # 수식sheet의 H4 가져오기
6
7 def cal_mm(in_mm):
8     return in_mm / mm_per_pulse
9
10 def main():
11
12     print(cal_mm(10.0)) # 10.0mm 이동에 필요한 펄스 수 계산
13
14 while True:
    
```


[Figure 3-20] 스텝 모터 10.0mm 이동 시 펄스 수 출력

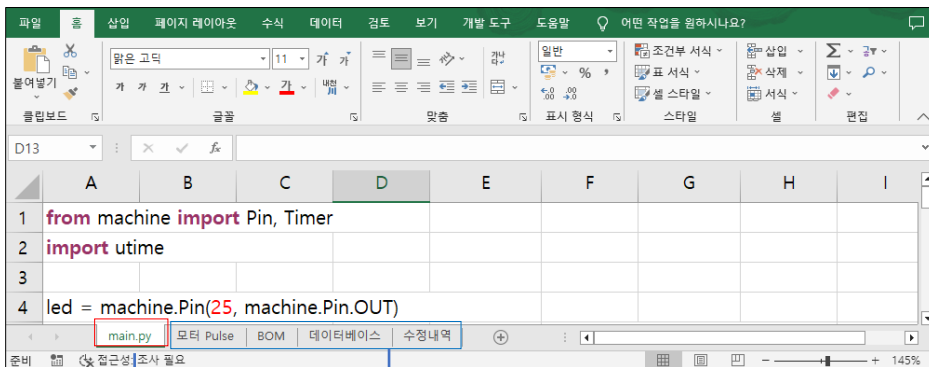


[Figure 3-20]은 스텝 모터를 10.0mm 이동 시 필요한 펄스 수를 print한 px-downloader-v1.00.exe의 Message창입니다. 이와 같이, Excel 수식을 응용할 수 있는 방법은 매우 다양할 수 있습니다. 수식 외에도 Excel은 수많은 기능을 내장하고 있습니다. Excel을 이용한 프로그래밍이 더욱 중요한 이유입니다.

3-3-8. Excel Sheet (main.py)

Excel Micro Python 프로그래밍 시 첫 번째 시트 이름은 "main.py"이고 이 시트에 코드가 작성되어야 합니다. px-downloader-V1.00.exe에서 Excel을 읽을 때 1번 시트의 이름이 "main.py"를 확인하고 Micro Python 파일(main.py)로 변환하게 됩니다. Excel의 1번 시트가 "main.py"가 아니면 정상적으로 읽을 수 없습니다. 2번 시트부터는 사용자 공간이므로 원하는 내용을 삽입할 수 있습니다.

[Figure 3-21] main.py 시트



1번 시트는 프로그램 영역으로 반드시 "main.py"여야 합니다

2번 시트부터 사용자 지정 영역입니다.

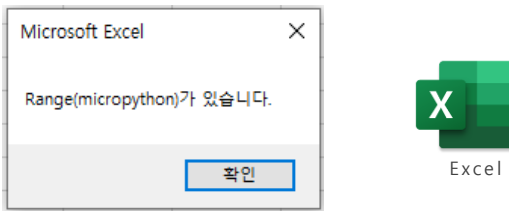
3-4. Excel VBA

VBA는 Visual Basic for Applications의 약자로 Excel, Power Point등 MS Office프로그램에 내장된 프로그래밍 언어입니다. Excel Micro Python은 VBA 코드가 포함된 Excel 파일을 사용합니다. VBA 코드는 키워드, 주석 등 프로그램의 가독성을 높이는 기능(문자의 색상, 폰트 변경)과 Excel 파일 Open 시 Range를 자동 선택하는 등의 기능을 수행합니다. 기준이 되는 VBA 코드는 "samplevba.xlsm"(기본 제공)에서 사용됩니다. px-downloader-V1.00.exe에서 *.py를 *.xlsm로 파일 변환 시 생성되는 Excel 파일에는 VBA 코드가 자동 저장되어 있습니다.

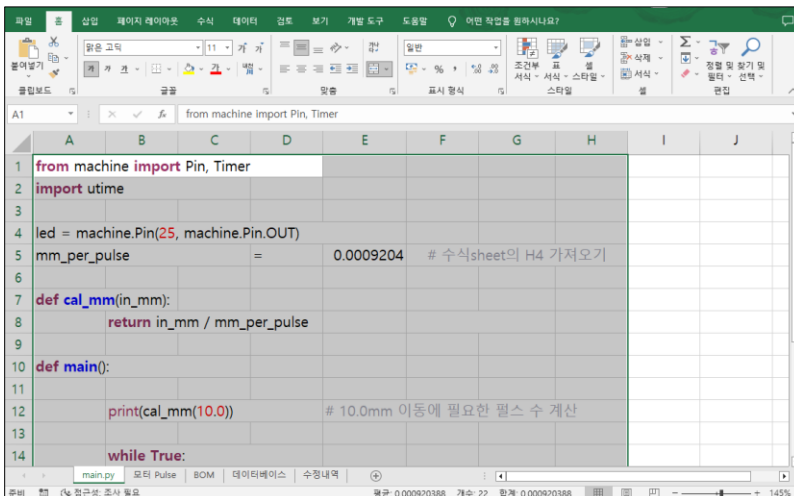
3-4-1. Range(micropython)

Excel 파일 Open시 "micropython" Range가 자동 선택됩니다. Range 선택 시 프로그램 내용에서 키워드, 함수, 주석 등 문자의 색상과 폰트를 변경합니다. 이 기능은 VBA 코드에 의해서 실행되고 프로그램의 가독성을 높이기 위한 것으로 이 기능이 없어도 프로그램은 정상 동작하지만, 가독성은 매우 떨어집니다. [Figure 3-22], [Figure 3-23]은 Excel 파일 Open시 "micropython" Range가 자동 선택되는 모습입니다.

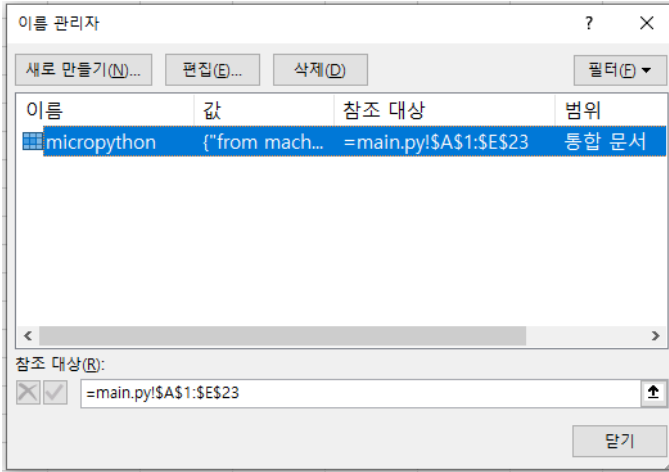
[Figure 3-22] Excel 파일 Open시 Range선택 알림 창



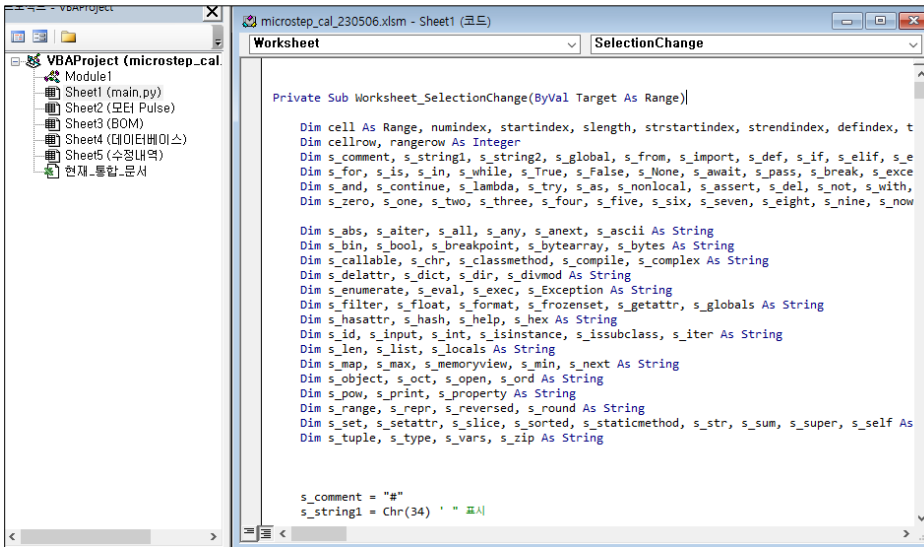
[Figure 3-23] Excel 파일 Open시 Range(micropython) 자동 선택



[Figure 3-24] Range(micropython) 이름 관리자



[Figure 3-25] VBA 코드 적용 화면

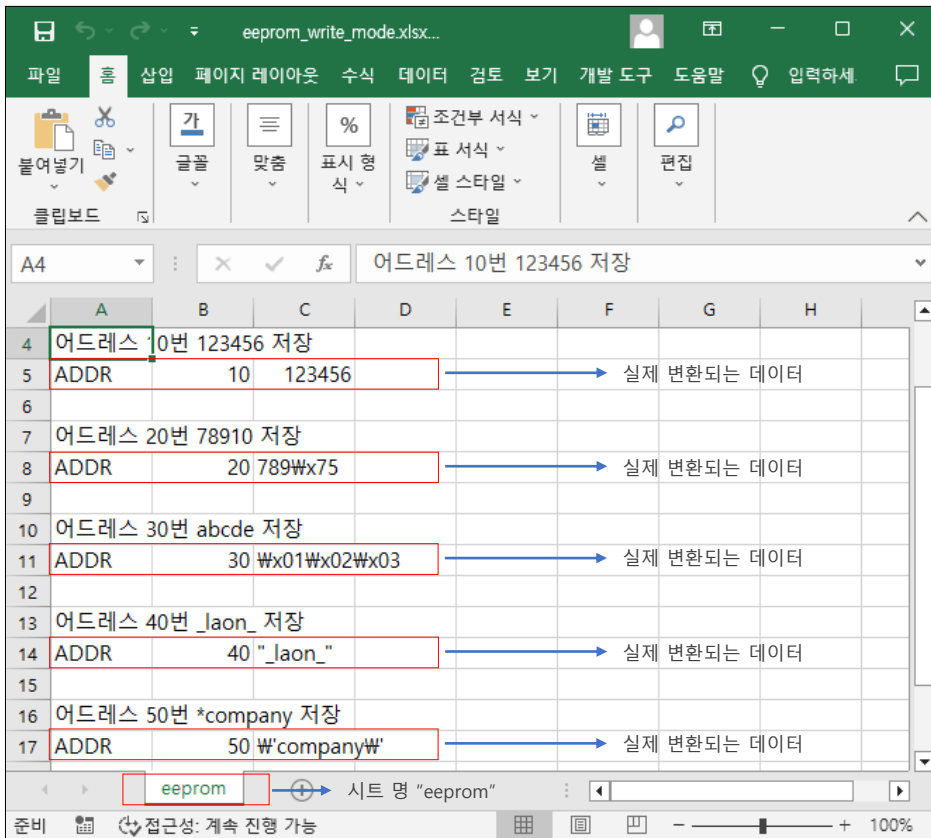


[Figure 3-24]는 "micropython" Range가 지정되어 있는 이름 관리자입니다. 사용자는 이름 관리자를 통하여 Range 범위를 변경할 수 있습니다. [Figure-25]는 VBA 코드 편집기 화면입니다. 사용자의 요구에 따라 키워드를 추가하거나 다른 기능도 추가할 수 있습니다. samplevba.xlsm파일을 수정하면 px-downloader-v1.00.exe를 이용해 변경하는 모든 파일에 수정한 VBA 코드가 적용됩니다.

3-5. Excel EEPROM 모드

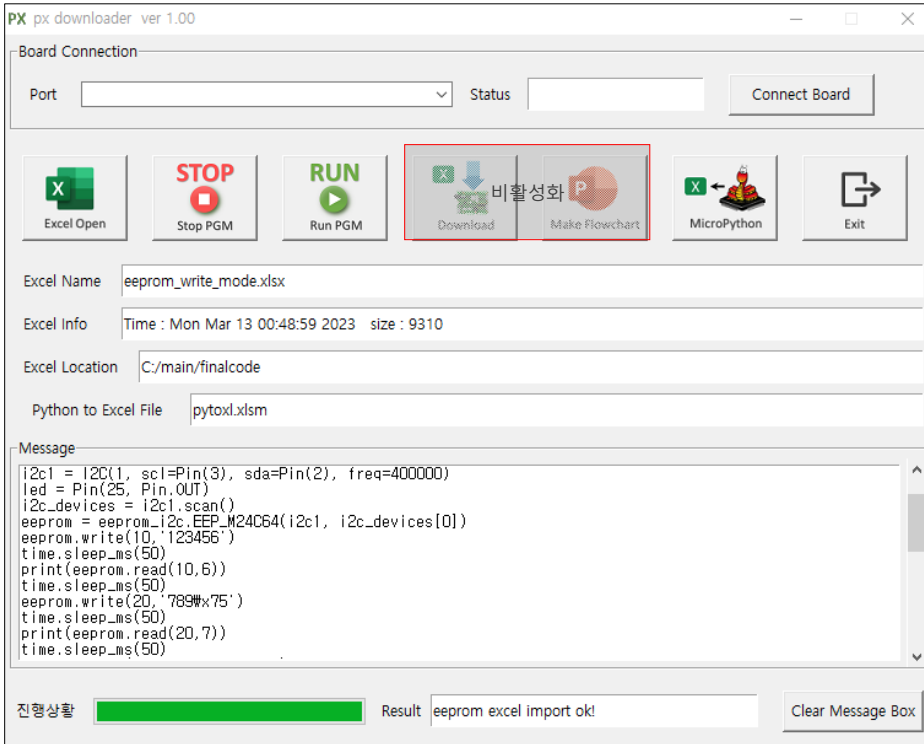
Excel EEPROM 모드는 프로그램의 내용(코드)을 작성하지 않고 PY-EXCEL-IO8의 데이터를 변경할 수 있는 기능입니다. 실제 실무에서는 프로그래머 외에 사용자가 코드의 내용을 이해하기까지는 시간이 필요합니다. 혹은, 코드의 내용은 필요하지 않은 사용자도 많습니다. 이 경우에는 복잡한 코드를 설명하기보다 데이터 변경이 필요한 부분만 Excel 파일로 전달하는 것이 더 효율적일 수 있습니다. Excel EEPROM 모드의 구성과 사용 방법을 알아보겠습니다.

[Figure 3-26] Excel EEPROM 모드 파일



[Figure 3-26]은 Excel EEPROM 모드 파일 화면입니다. eeprom_write_mode.xlsx파일이 EEPROM 모드가 적용된 샘플 코드이고 기본 제공됩니다. EEPROM 모드는 "ADDR"로 시작하는 Row(행)가 저장 데이터로 변환됩니다. 5번 행을 예로 들면 어드레스 10번에 "123456"을 문자열 형태로 저장합니다. 데이터 저장 시 문자열로 저장되므로 소스 코드에서 integer 등으로 변환이 필요할 수 있습니다. "ADDR" 키워드가 포함된 행을 제외한 나머지 내용은 변환하지 않습니다. 시트 명은 "eeprom"으로 지정되어야 하고 VBA 코드는 적용되지 않은 xlsx 파일을 사용할 수 있습니다.

[Figure 3-27] px-downloader-v1.00.exe에서 Excel EEPROM 모드 File Open

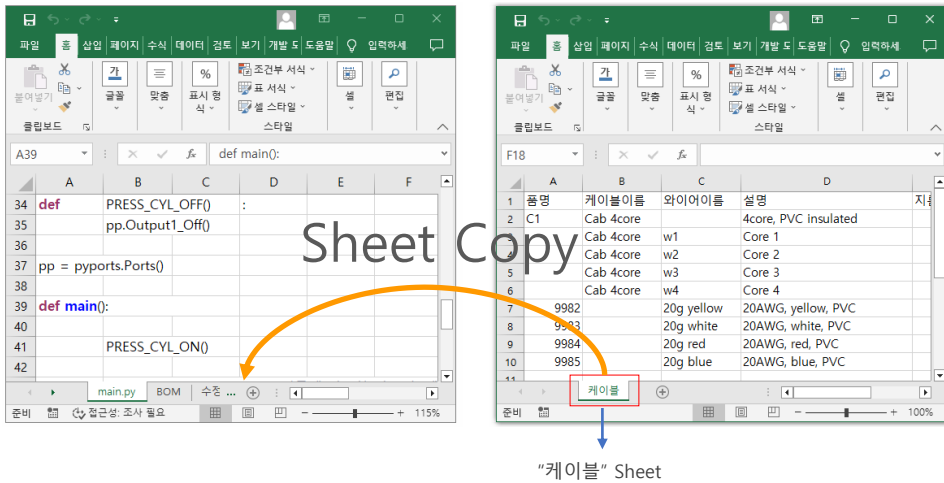


[Figure 3-27]은 px-downloader-v1.00.exe에서 Excel EEPROM 모드 파일 Open 시 화면입니다. Excel EEPROM 모드 파일은 실행 가능한 main.py로 변환되어 있습니다. main.py는 Run 버튼만 누르면 PY-EXCEL-IO8의 eeprom 데이터를 변경합니다. Excel EEPROM 모드로 파일을 열면 Download 버튼과 Make Flowchart 버튼이 비활성화 되어 있는 것을 확인할 수 있습니다.

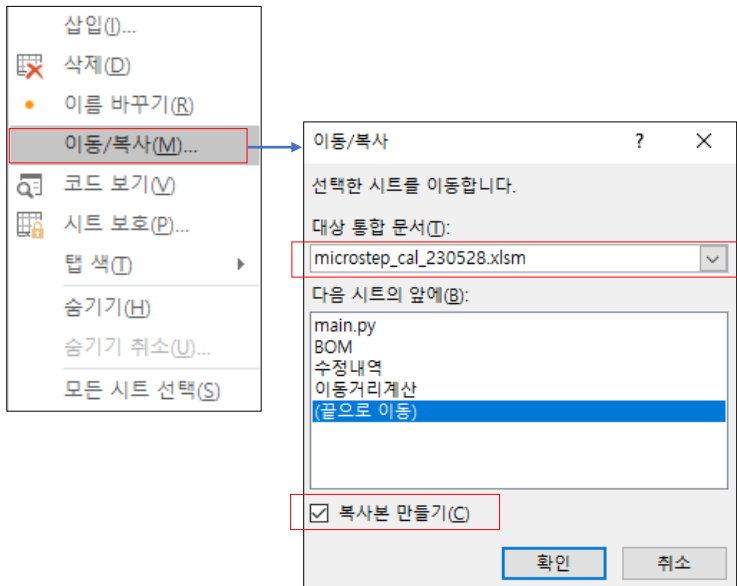
3-6. 다른 Excel 파일에 Sheet 복사

Excel 파일에 소스 코드를 포함한 각종 데이터를 함께 저장하는 것은 PY-EXCEL-IO8의 주요 기능 중 하나입니다. 다른 Excel 파일에 시트를 복사하는 방법을 알아보겠습니다. 우선, [Figure 3-28]과 같이 Excel 파일 2개(소스 코드, 케이블리스트)를 열어서 동시에 화면에 나오게 합니다. 다음, 케이블리스트 Excel 파일의 "케이블" Sheet에서 마우스 우측 버튼을 누릅니다. 나타나는 메뉴에서 '이동/복사(M)'를 선택하고 이동/복사 화면에서 '대상 통합 문서(T):'에서 저장될 Excel 파일을 선택합니다. 시트의 위치를 선택하고 '복사본 만들기(C)'를 체크한 후 확인을 누르면 복사가 완료됩니다.

[Figure 3-28] Excel 파일 Sheet 복사하기




















[Figure 3-29] "케이블" Sheet에서 마우스 우측 클릭



3-7. 샘플 소스 코드 리스트

PY-EXCEL-IO8은 제품 구입 시 샘플 코드 파일을 제공합니다. 파일의 종류는 프로그램 코드, Excel EEPROM 모드, Flowchart 도형 생성, 보드 테스트 샘플 코드입니다. [Figure 3-30]은 샘플 코드 리스트입니다.

[Figure 3-30] 샘플 코드 리스트

 eeprom_write_mode.xlsx	→ EEPROM WRITE 모드 Excel 파일
 pxb_boardtest.py	→ PY-EXCEL-IO8 제어보드 테스트 Micro Python 파일
 pxb_boardtest.xlsm	→ PY-EXCEL-IO8 제어보드 테스트 Excel 파일
 pxb_eeprom.py	→ 내장 EEPROM 테스트 Micro Python 파일
 pxb_eeprom.xlsm	→ 내장 EEPROM 테스트 Excel 파일
 pxb_extbd.py	→ IN,OUT 확장 보드 테스트 Micro Python 파일
 pxb_extbd.xlsm	→ IN,OUT 확장 보드 테스트 Excel 파일
 pxb_fnd4.py	→ LC-FND4 확장 보드 테스트 Micro Python 파일
 pxb_fnd4.xlsm	→ LC-FND4 확장 보드 테스트 Excel 파일
 pxb_i2c_l3g4200d.py	→ I2C 포트 테스트 Micro Python 파일
 pxb_i2c_l3g4200d.xlsm	→ I2C 포트 테스트 Excel 파일
 pxb_klcd.py	→ LC-KLCD 확장 보드 테스트 Micro Python 파일
 pxb_klcd.xlsm	→ LC-KLCD 확장 보드 테스트 Excel 파일
 pxb_spi_tmc5160.py	→ XY-STEP-TMC5160A SPI 포트 테스트 Micro Python 파일
 pxb_spi_tmc5160.xlsm	→ XY-STEP-TMC5160A SPI 포트 테스트 Excel 파일
 pxb_uart.py	→ UART0,1 포트 테스트 Micro Python 파일
 pxb_uart.xlsm	→ UART0,1 포트 테스트 Excel 파일
 sample_flowchart_shape.pptx	→ Flowchart 도형 생성 Power Point 파일
 sample_flowchart_shape.xlsm	→ Flowchart 도형 생성 테스트 Excel 파일

3-8. Windows OS, Excel 버전

PY-EXCEL-IO8과 Excel 파일을 테스트한 OS와 Excel 버전입니다.

항목	테스트 버전
OS	<ul style="list-style-type: none"> ▪ Windows 10 ▪ Windows 11
Excel	<ul style="list-style-type: none"> ▪ Excel 2010 이상

3-9. Excel 단축키 리스트

Excel은 매우 다양한 기능을 포함하고 있습니다. 이러한 Excel의 장점을 PY-EXCEL-IO8 코딩에 활용할 수 있습니다. 실제 단축키는 더욱 다양하고 필요한 기능은 VBA를 통해 구현도 가능합니다. [Figure 3-31], [Figure 3-32]는 단축키 리스트입니다.

[Figure 3-31] Excel 단축키 리스트1

● Ctrl 키		● F1 ~ F12 키	
Ctrl + A	모두 선택	F1	도움말
Ctrl + C	복사	F2	현재 셀 편집
Ctrl + D	윗 셀 복사 붙여넣기	F3	셀 이름 넣기
Ctrl + F	찾기	F4	마지막 작업 반복
Ctrl + I	글자 기울이기	F5	이동
Ctrl + N	새 통합문서	F6	다음 틀로 이동
Ctrl + O	통합문서 열기	F7	맞춤법 검사
Ctrl + P	인쇄	F8	영역 선택
Ctrl + R	왼쪽셀 복사 붙여넣기	F9	모두 다시 계산
Ctrl + S	저장	F10	메뉴 활성화
Ctrl + V	붙여넣기	F11	현재 범위 차트 만들기
Ctrl + X	잘라내기	F12	다른 이름으로 저장
Ctrl + Y	재실행	● Alt 키 , 기타	
Ctrl + Z	실행 취소	Home	첫 행으로 이동
Ctrl + 1	셀 서식창 표시	End	모드 끄/켄
Ctrl + 2	글씨 굵게	Tab	다음 셀로 이동
Ctrl + 4	글씨 밑줄	Enter	입력, 링크 실행
Ctrl + 5	삭제선	Delete	삭제
Ctrl + 9	행 숨기기	Alt	단축키 확인
Ctrl + F2	인쇄 미리보기	Alt + F1	차트 삽입
Ctrl + F4	창 닫기	Alt + F2	다른 이름으로 저장
Ctrl + F11	새 매크로 시트	Alt + F4	종료
Ctrl + F12	열기	Alt + F8	매크로 대화 상자
Ctrl + Home	워크시트 처음으로	Alt + F	파일 메뉴
Ctrl + End	워크시트 끝으로	Alt + ↓	선택한 드롭다운 목록 열기
Ctrl + Tab	다음 창으로	Ctrl + ;	현재 날짜 입력
Ctrl + +	셀/행/열 삽입	Ctrl + Page up	이전 워크시트로
Ctrl + -	셀/행/열 삭제	Ctrl + Page down	다음 워크시트로

[Figure 3-32] Excel 단축키 리스트2

● Shift 키	
Shift + F2	메모 삽입
Shift + F3	함수 마법사
Shift + F5	찾기
Shift + F9	현재 시트 재계산
Shift + F10	팝업 메뉴
Shift + F11	새 시트 삽입
Shift + ↑ ↓ ← →	선택한 셀과 그 방향의 셀 선택
Shift + Enter	윗 행으로 이동
Shift + Space Bar	행 전체 선택
Shift + Tab	왼쪽 열로 이동
Shift + Page up	선택한 셀과 1페이지 아래의 셀까지 모두 선택
Shift + Page down	선택한 셀과 1페이지 위의 셀까지 모두 선택
● Ctrl 키 + Shift 키	
Ctrl + Shift + #	년, 월, 일로 날짜 서식 적용
Ctrl + Shift + %	소수 자릿수 없이 백분율 서식 적용
Ctrl + Shift + &	선택한 셀에 윤곽선 테두리 적용
Ctrl + Shift + :	현재 시간 입력
Ctrl + Shift + @	시간, 분, AM/PM 시간 서식 적용
Ctrl + Shift + ^	소수 두 자리 지수 숫자 서식 적용
Ctrl + Shift + P	글꼴 크기 변경
Ctrl + Shift + F6	이전 통합문서로
Ctrl + Shift + _	선택한 셀 윤곽선 테두리 제거
Ctrl + Shift + ~	일반 숫자 서식 적용
Ctrl + Shift + F	글꼴 변경
Ctrl + Shift + ↑	선택한 셀과 위쪽 데이터 셀 모두 선택
Ctrl + Shift + ↓	선택한 셀과 아래쪽 데이터 셀 모두 선택
Ctrl + Shift + ←	선택한 셀과 좌측 데이터 셀 모두 선택
Ctrl + Shift + →	선택한 셀과 우측 데이터 셀 모두 선택

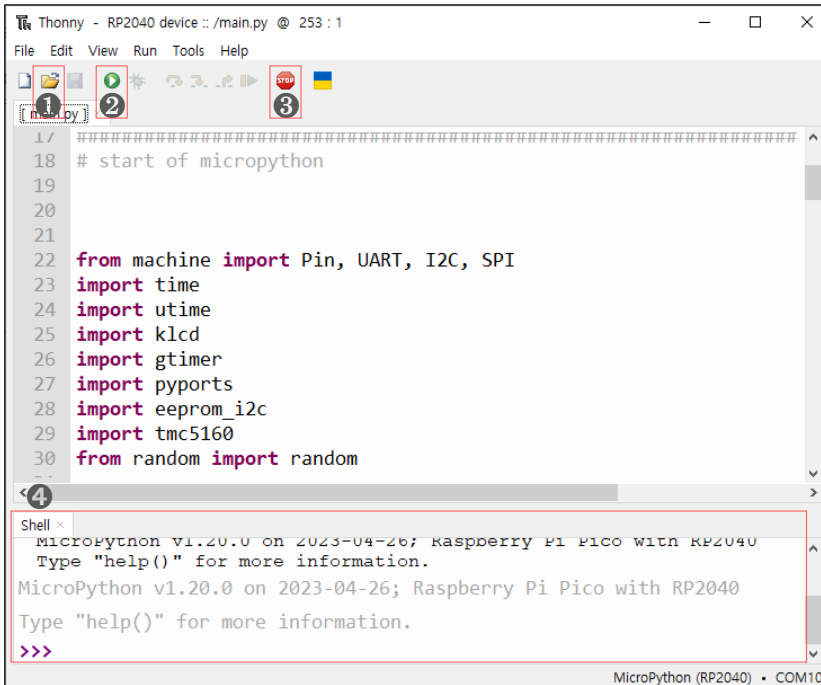
4. Thonny IDE 개발 툴

4-1. 시작하기

PY-EXCEL-IO8 제어보드는 Thonny IDE를 이용하여 프로그래밍이 가능합니다. Thonny IDE는 main.py, lib 폴더에 파일 저장, REPL(Read Evaluate Print Loop)프롬프트를 사용할 수 있습니다. Excel에서는 lib 폴더 접근이 되지 않으므로 고급 개발을 진행 시 Thonny IDE를 필수로 사용해야 합니다. 이 장에서는 Thonny IDE에서 PY-EXCEL-IO8을 연결하고 lib 폴더의 *.py를 확인하고 import 하는 방법을 알아보겠습니다. (※Thonny IDE는 Online에서 무료로 다운로드 가능합니다.)



[Figure 4-1] Thonny IDE에서 PY-EXCEL-IO8 연결된 상태



[Figure 4-1]는 Thonny IDE의 실행 화면을 보여줍니다. PY-EXCEL-IO8과 같은 RP2040 Device를 내장한 보드에 유용한 유용한 소프트웨어입니다.

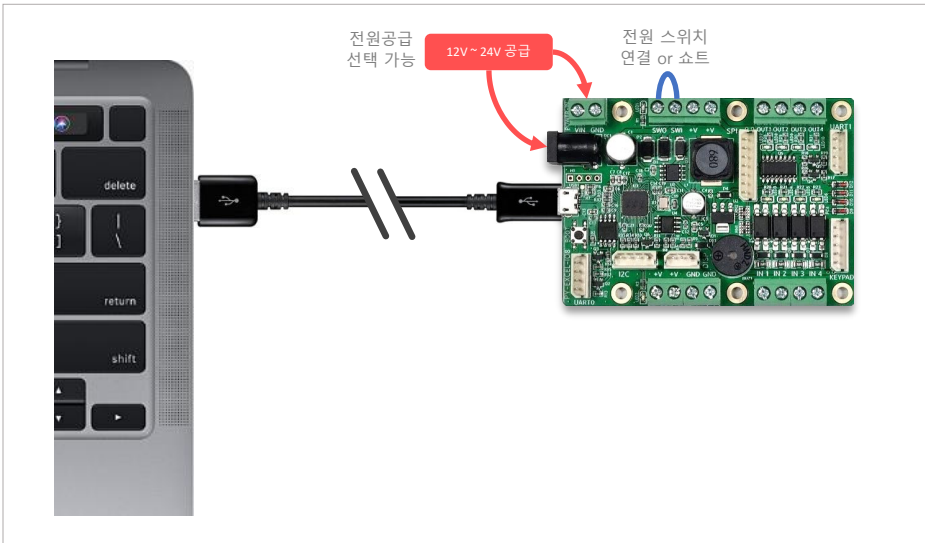
- ① 파이썬 파일 Open (PC/RP2040 선택 가능)
- ② 프로그램 실행 (화면에 표시된 *.py 실행)
- ③ 보드 Stop / Reset
- ④ REPL 프롬프트 화면 (코드를 문자열 단위로 실행)

4-2. PY-EXCEL-IO8 보드 연결하기

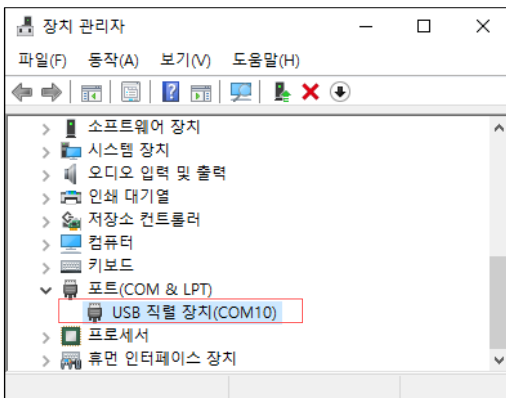
PY-EXCEL-IO8 제어보드는 USB로 파이썬 프로그램 다운로드, 디버깅을 수행합니다. Thonny IDE에서 보드를 연결하는 방법을 알아보겠습니다.

4-2-1. Micro USB 케이블 연결

[Figure 4-2] Micro USB 케이블 연결



[Figure 4-3] PC에서 PY-EXCEL-IO8보드를 Comport로 인식

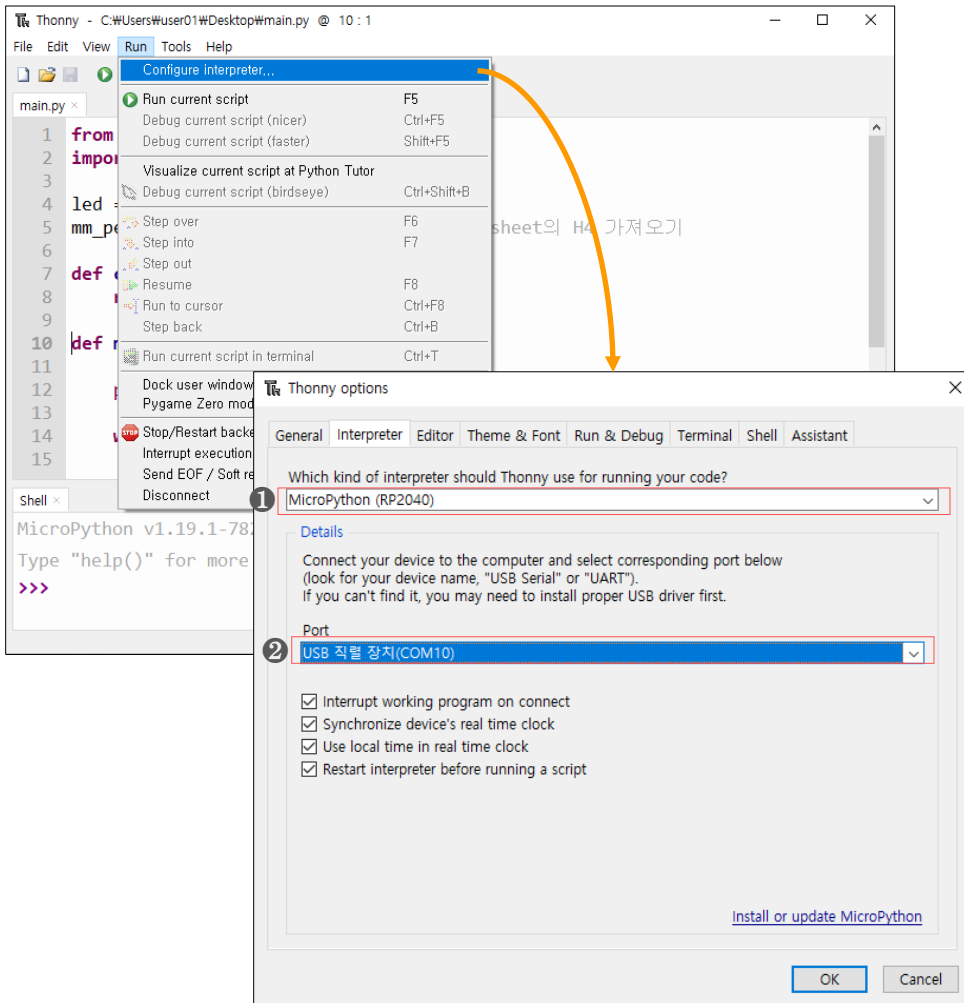


제품에는 펌웨어가 저장되어 있기 때문에 처음 PC에서 연결하여도 Comport로 인식됩니다. [Figure 4-3]에서는 "COM10"로 연결되었으나 PC마다 다른 Comport Number가 생성됩니다. 보드는 외부전원을 사용합니다. DC12V ~ 24V를 별도 공급하고 SWO, SWI 핀에 전원 스위치를 연결하거나 쇼트상태로 유지하여 주십시오.

4-2-2. Thonny IDE에서 연결 설정

Thonny IDE를 실행합니다. Run/Configure Interpreter항목을 선택합니다. Thonny Options창이 나타나고 Interpreter항목이 선택되어 있습니다. 여기서 ① Micro Python (RP2040), ② 보드가 연결된 Comport를 선택하고 OK를 클릭합니다. 이제 보드가 Thonny IDE에 연결되었으며 *.py파일을 실행할 수 있습니다.

[Figure 4-4] Thonny IDE에서 PY-EXCEL-IO8을 처음 연결

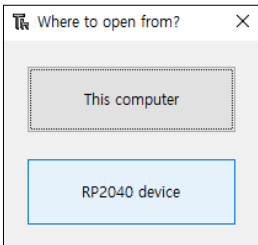


4-3. lib 폴더 사용하기

ePY-EXCEL-IO8은 보드는 main.py를 메인 프로그램으로 동작하고 lib 폴더에 저장된 프로그램 파일을 import 할 수 있습니다. 이번에는 lib 폴더에 파일을 Open하는 방법과 Import 되는 방식을 설명합니다. 본 설명서는 파이썬 문법은 설명하지 않음을 참고하여 주시기 바랍니다.

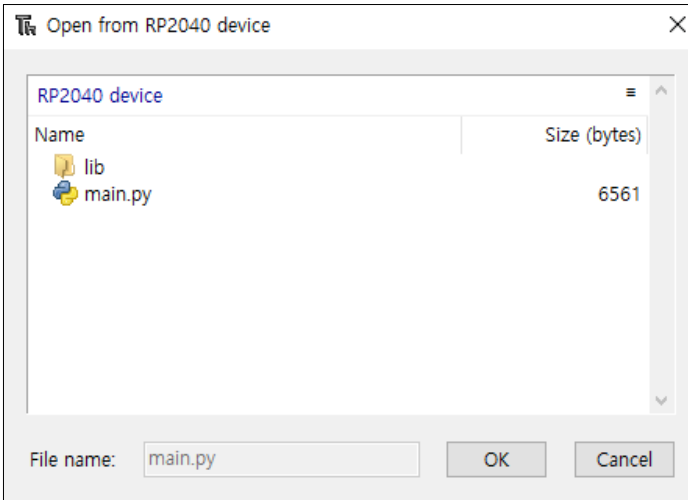
4-3-1. lib 폴더 열기

[Figure 4-5] Thonny IDE에서 PC or RP2040보드 Open 대상 선택



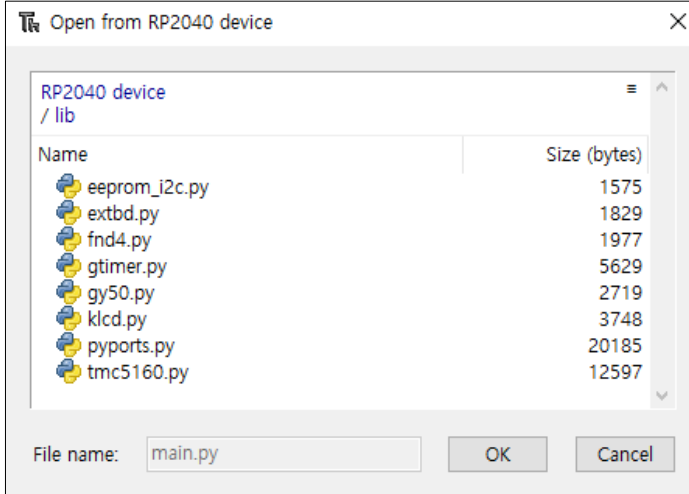
[Figure 4-5]는 Thonny IDE에서 Open아이콘을 선택할 때 나타나는 창입니다. 보드가 연결되어 있으면 PC/RP2040중 선택하여 파이썬 파일을 열수 있습니다. 이번에는 RP2040을 선택하여 PY-EXCEL-IO8의 Flash에 저장된 파일을 Open합니다. [Figure 4-6]은 RP2040을 선택 후 보드에 저장된 파일을 확인합니다. main.py는 메인 프로그램으로 Power On 시 구동되는 파일이며 lib 폴더의 파일을 import 하여 사용할 수 있습니다. lib 폴더를 선택하여 내용을 확인합니다.

[Figure 4-6] RP2040보드의 main.py와 lib폴더



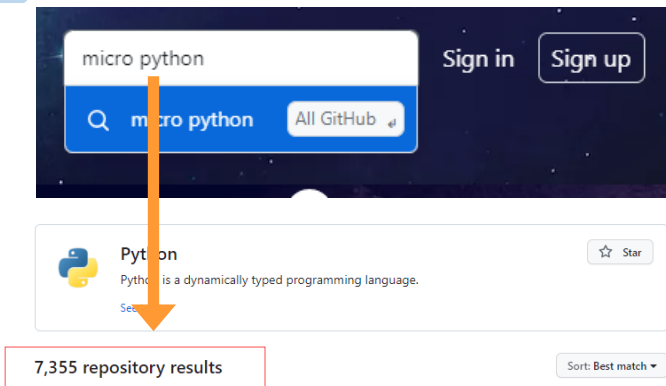
4-3-2. lib 폴더 내용 확인

[Figure 4-7] PY-EXCEL-IO8의 lib 폴더에 저장된 파일



[Figure 4-7]은 lib 폴더의 내용입니다. PY-EXCEL-IO8의 모든 기능을 사용할 수 있는 파일이고 사용자가 수정, 추가, 또는 삭제할 수 있습니다. 새로운 파일을 추가 시 Github의 자료를 참고하여 더욱 쉽게 기능확장이 가능합니다.

참고 [Figure 4-8] Github에서 micro python 검색



[Figure 4-8]은 Github에서 micro python을 검색한 결과입니다. 7,300개 이상의 Python 관련 자료의 결과가 나오며 원하는 기능확장을 추가할 수 있습니다.

4-3-3. lib 폴더 파일 import (klcd.py)

[Figure 4-9] main.py에서 lib 폴더의 klcd.py를 import

```

1 23 import klcd
24 import pyports
25 import gtimer
26 from gy50 import GY50

2 35 lcd1602 = klcd.Klcd()
36
37 # klcd display version
38 def klcd_version():
39     lcd1602.lcd_show(1, sFILE_NAME)
40     lcd1602.lcd_show(2, sFILE_TEST)

```

[Figure 4-9]는 main.py에서 lib 폴더의 klcd.py를 참조하는 내용입니다. main.py에서 ❶“klcd”(py는 제거)를 import하고 klcd.py의 ❷“Klcd”클래스로 객체(object)를 생성합니다. 생성된 인스턴스 객체(lcd1602)를 사용하여 klcd.py에 설정된 메서드를 사용할 수 있습니다.

[Figure 4-10] klcd.py의 class Klcd

```

108 class Klcd:
109
110
111     def lcd_blink_on(self, line, time):
112         global iTmrBlink
113         global iTmrValue
114         global iBlinkLine
115         global bLineBlinkOn
116         global bLineBlinkTog
117         iTmrBlink = 0
118         iTmrValue = time
119         iBlinkLine = line
120         bLineBlinkOn = True
121         bLineBlinkTog = False

```

[Figure 4-10]은 klcd.py의 내용의 일부입니다. Klcd클래스의 시작부분으로 def로 시작하는 메서드를 코딩할 수 있고 제작된 메서드는 main.py에서 Klcd로 객체 생성 후 사용 가능할 수 있습니다.

5. px-downloader-v100.exe

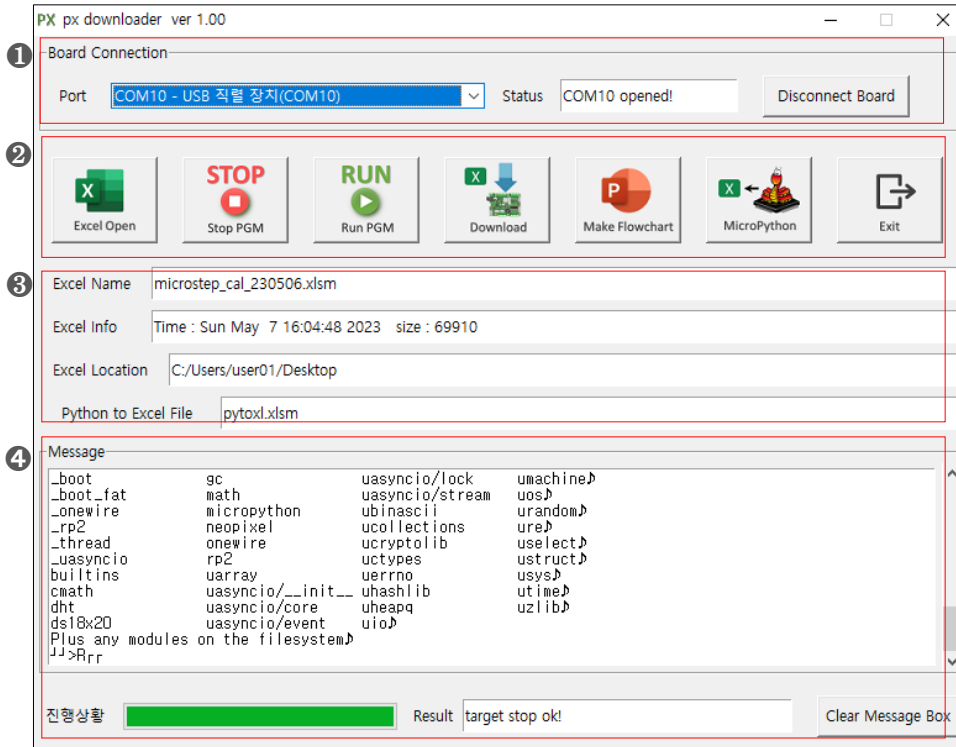
5-1. 시작하기

px-downloader-v1.00.exe는 Excel 파일을 PY-EXCEL-IO8 제어보드에서 실행, 저장할 수 있는 소프트웨어입니다. Thonny IDE와 동일한 방법으로 PY-EXCEL-IO8을 연결하고 디버깅합니다. 부가적인 기능은 Power Point용 Flow Chart를 생성하고 Micro Python File(*.py)을 Excel File(*.xlsm)로 변환할 수 있습니다. 이 장에서는 px-downloader-v100.exe에서 PY-EXCEL-IO8을 제어하는 방법과 부가 기능의 사용 방법을 알아보겠습니다.



px-downloader-v100.exe

[Figure 5-1] px-downloader-v100.exe 화면



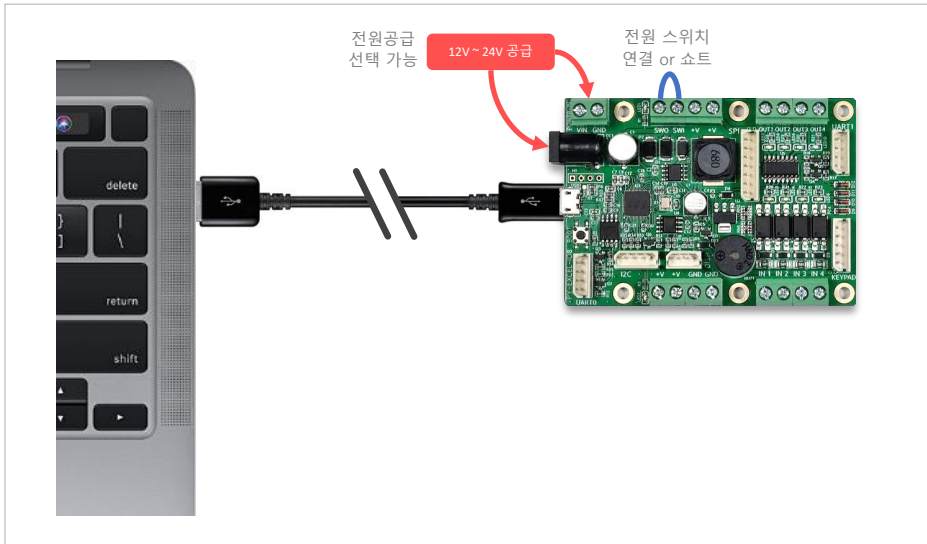
- ① PY-EXCEL-IO8보드 연결 Comport 설정
- ② 제어 동작 버튼
- ③ Excel 파일 정보, Micro Python File(*py) to Excel File(*xlsm) 변환 이름
- ④ Run, Debug Message 창, 진행상황 표시

5-2. PY-EXCEL-IO8 보드 연결

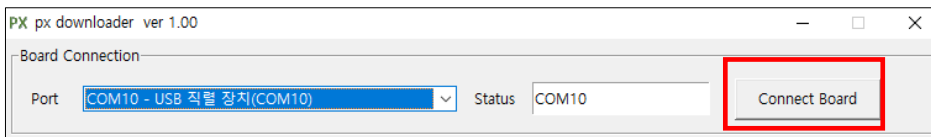
px-downloader-v1.00.exe에서 PY-EXCEL-IO8 제어보드를 연결하는 방법은 Thonny IDE와 동일합니다. Micro USB 케이블로 PC와 보드를 연결하고 px-downloader-v100.exe에서 인식된 Comport를 선택한 후 "Connect Board"버튼을 누릅니다.

5-2-1. Board Connection

[Figure 5-2] Micro USB 케이블 연결



[Figure 5-3] px-downloader-v100.exe 보드 연결



[Figure 5-2], [Figure 5-3]은 PY-EXCEL-IO8 제어보드를 PC에 연결하고 px-downloader-v100.exe에서 보드를 확인하는 모습입니다. Board Connection항목에서 연결되어 있는 보드의 Comport를 선택하고 "Connect Board" 버튼을 누르면 Comport연결을 시도하고 완료 시 버튼의 글자는 "Disconnect Board"로 바뀝니다. 이제 Excel 파일을 읽고 다운로드할 수 있습니다.

5-3. Excel 파일 실행

px-downloader-v1.00.exe의 주요 기능인 Excel 파일을 열고 PY-EXCEL-IO8 제어보드를 제어하고 Flowchart 생성, Micro Python 파일에서 Excel 파일로 변환하는 방식을 설명합니다.

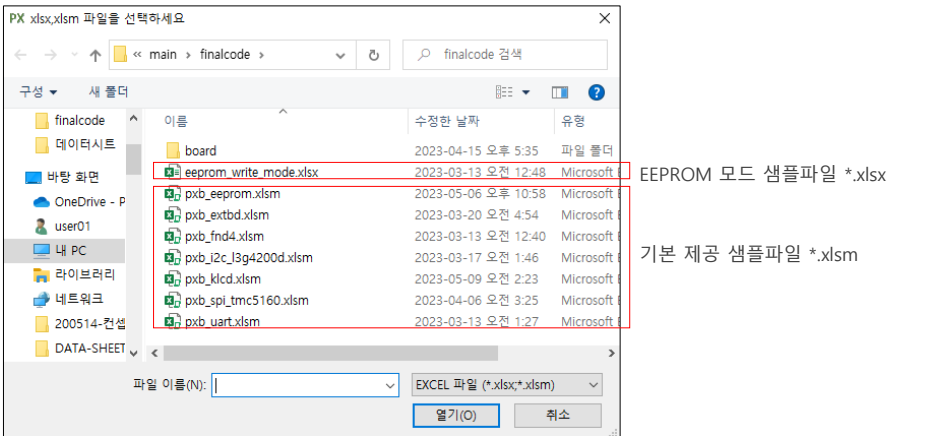
5-3-1. Excel Open

Micro Python 코드가 포함된 Excel 파일을 엽니다. VBA 코드가 포함된 *.xlsm 확장자를 가지며 첫 번째 Sheet 명은 "main.py"로 되어 있어야 합니다. VBA 코드는 실제 동작과는 무관하며 Micro Python 코드를 보기 쉽게 하기 위한 키워드 글자 색상, 스타일, 주석 부분 글자 색상 변경 등이 포함된 VBA 코드입니다. VBA 코드에 대한 자세한 내용은 '3-3. Excel VBA (Micro Python Range)' 항목을 참고해 주시기를 바랍니다.

EEPROM 모드용 Excel 파일은 *.xlsx 확장자도 가능하며 첫 번째 Sheet 명을 "eprom"으로 지정해야 합니다. 모든 내용은 샘플 코드에서 확인할 수 있습니다. 정상적으로 Open 되면 Micro Python 파일 "main.py"로 변환되어 있습니다.



[Figure 5-4] px-downloader-v100.exe Excel 파일 열기



[Figure 5-5] Excel 파일 정보 표시

Excel Name	pxb_klcd.xlsm
Excel Info	Time : Tue May 9 02:23:48 2023 size : 72661
Excel Location	C:/Users/user01/Desktop/KPRO/K_WORK/PRO23/PY_EXCEL_IO8_REV1/KPGM23/main/finalcode

5-3-2. Stop PGM, Run PGM, Download

px-downloader-v1.00.exe에서 PY-EXCEL-IO8를 제어하는 버튼으로는 Stop(Reset) PGM, Run PGM, Download가 있습니다. 각 버튼의 기능을 설명합니다.

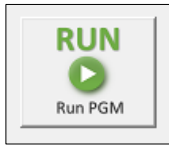


[Figure 5-6] Stop PGM 버튼



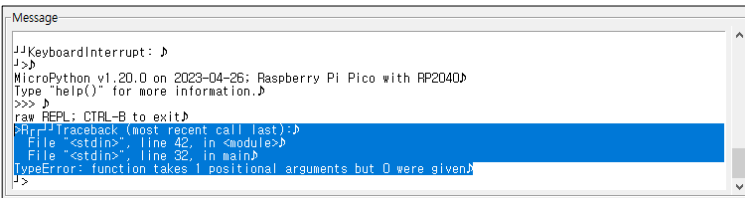
PY-EXCEL-IO8 제어보드를 Stop(Reset)하는 버튼입니다. 보드가 Comport로 연결된 상태에서 동작 중인 프로그램을 중단합니다. 보드가 Reset 된 후 Run PGM 버튼, Download 버튼을 실행할 수 있습니다. 처음 보드를 연결 시 Stop(Reset)을 실행해야 합니다.

[Figure 5-7] Run PGM 버튼

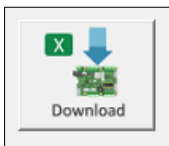


Run PGM버튼은 Excel Open 버튼을 눌러 import 된 *.xslm프로그램을 실행합니다. *.xslm의 첫 번째 Sheet "main.py"를 실행하고 다른 Sheet의 내용은 상관이 없습니다. 프로그램을 실행 중 발생하는 오류는 Message 창에 나타나고 오류의 특성에 따라 프로그램이 중단되기도 합니다. [Figure 5-8]은 Message 창 오류 발생 시 표시 내용입니다. 자세한 내용은 다음 페이지를 참조해 주세요.

[Figure 5-8] Run PGM 오류 발생



[Figure 5-9] Download 버튼



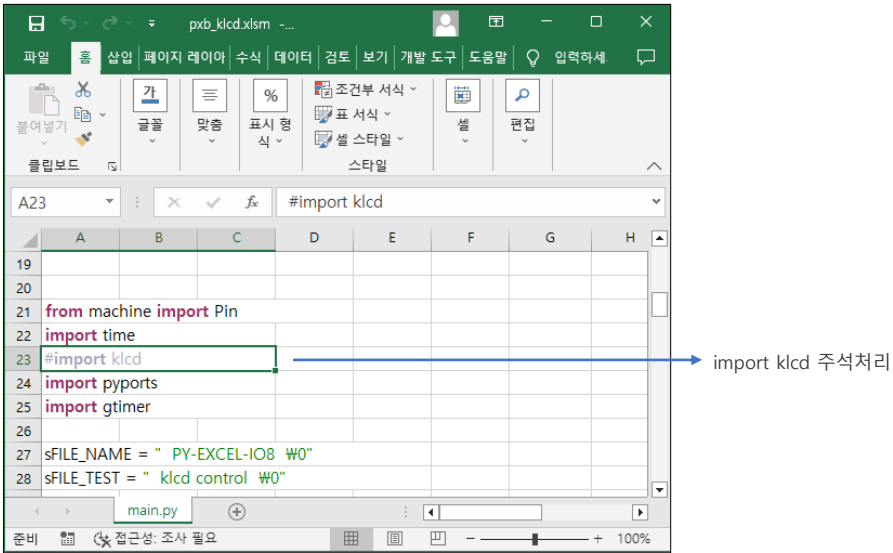
Download 버튼은 import 된 *.xslm프로그램을 PY-EXCEL-IO8 제어보드로 저장합니다. 저장 시 "main.py"로 저장되며 Thonny IDE에서도 읽어볼 수 있습니다. 저장이 완료되면 보드를 Power Off → Power On 하여 main.py가 정상적으로 실행되는지 확인하시기 바랍니다.

※ Run PGM버튼은 프로그램을 저장하지 않고 실행합니다. main.py로 저장되지 않음을 유의하시기 바랍니다.

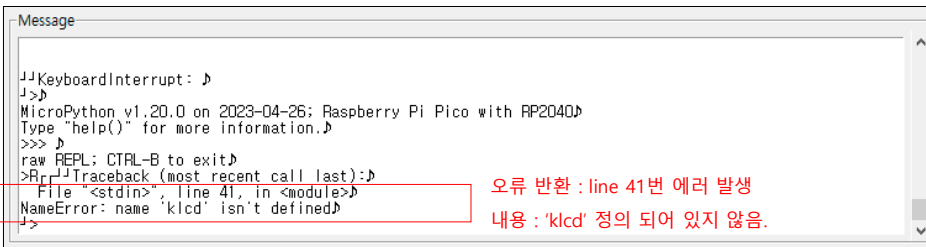
5-3-3. px-downloader-v100.exe 디버깅 테스트

px-downloader-v1.00.exe에서는 Run PGM 버튼으로 프로그램을 실행 후 Message 창을 통하여 디버깅합니다. 프로그램에 오류가 있으면 Message 창으로 내용이 반환됩니다. pxb_klcd.xlsm파일에서 오류를 발생시키고 디버깅하는 방법을 알아보겠습니다.

[Figure 5-10] pxb_klcd.xlsm 오류 발생 테스트



[Figure 5-11] Message 창 오류 반환



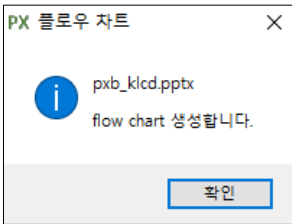
[Figure 5-10], [Figure 5-11]은 샘플 코드 pxb_klcd.xlsm파일로 오류 발생 테스트 화면입니다. import klcd를 주석처리 후 파일저장, 프로그램 실행 시 PY-EXCEL-IO8 제어보드에서 에러 메시지가 반환되었습니다. 내용은 line 41번에서 klcd가 지정되지 않았다는 내용으로 import klcd없이 klcd내용이 사용되어 에러가 발생한 것입니다. Thonny IDE에서 동일하게 확인할 수 있습니다. 디버그 방식은 px-downloader.v100과 Thonny IDE 모두 동일합니다. 오류를 수정 후 재시도하면 정상 동작합니다.

5-3-4. Make Flowchart

px-downloader-v100.exe에는 Flowchart를 생성하는 기능이 포함되어 있습니다. Power Point용 Flowchart는 메뉴얼이나 Instruction을 제작 시 유용하게 사용될 수 있습니다. 사용자는 코딩 시 Flowchart 용 도형, 색상을 지정할 수 있습니다. 자세한 내용을 알아보겠습니다.

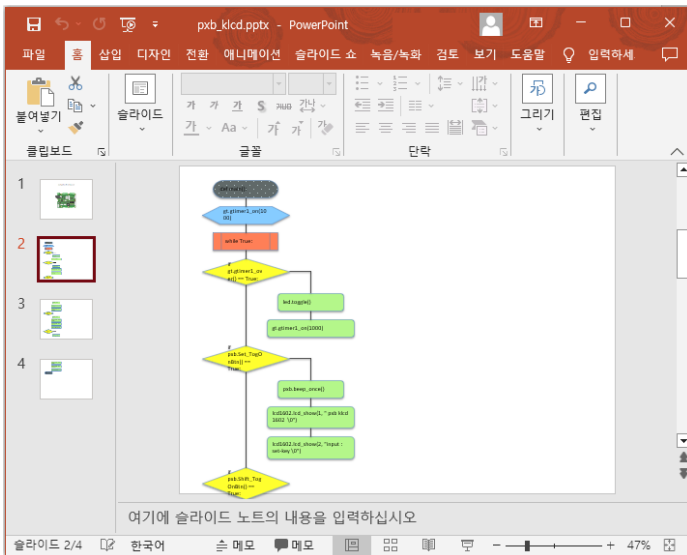


[Figure 5-12] Flowchart 생성 알림 창



*.xslm파일을 Open 후 Make Flowchart 버튼을 누르면 변환을 시작합니다. 프로그램의 main함수(def main(): ~ main()까지)를 변환합니다. 즉, Flowchart를 변환할 부분은 main함수에 해당하며 코드의 모든 부분을 변환 시 양이 너무 많기 때문에 사용자가 메소드를 이용하여 변수를 초기화하거나 기능을 하나의 메소드로 만들어 변환할 부분을 조절할 수 있습니다. 일반적으로 생성된 Flowchart는 양이 많기 때문에 필요한 부분만을 복사하여 새로운 Power Point 파일을 제작하는 것도 좋은 방법입니다.

[Figure 5-13] 생성된 Flowchart 파일



5-3-5. Flowchart 변환 구역

px-downloader-v100.exe의 Flowchart를 생성 기능은 프로그램의 main함수(def main(): ~ main()까지)를 변환합니다. 코드의 모든 부분을 변환 시 양이 너무 많기 때문에 main함수의 내용만 변환됩니다. 사용자는 메소드를 이용하여 변수를 초기화하거나 기능을 통합하여 변환할 부분을 조절할 수 있습니다. [Figure 5-14]는 Excel 파일, [Figure 5-15]는 변환된 Flowchart로 전체적인 흐름을 보여줍니다.

[Figure 5-14] Flowchart 변환 구역 표시(main함수)

```

7 def cal_mm(in_mm):
8     return in_mm / mm_per_pulse
9
10 def main():
11     print(cal_mm(10.0)) # 10.0mm 이동에 필요한 펄스 수 계산
12
13     while True:
14         print(cal_mm(10.0))
15         led.value(1)
16         utime.sleep(1.0)
17         print("led = off")
18         led.value(0)
19         utime.sleep(1.0)
20
21
22 main()
    
```

- ➔ ① main 시작 변환 시작
- ➔ ② loop 시작 전 처리 구역
- ➔ ③ loop 시작
- ➔ ④ loop 반복 동작 구역
- ➔ ⑤ main 끝 변환 종료

[Figure 5-15] 생성된 Flowchart 파일

- ➔ ① main 시작 변환 시작
- ➔ ② loop 시작 전 처리 + 주석 변환
- ➔ ③ loop 시작
- ➔ ④ loop 반복 동작 구역
- ➔ ⑤ main 끝 변환 종료

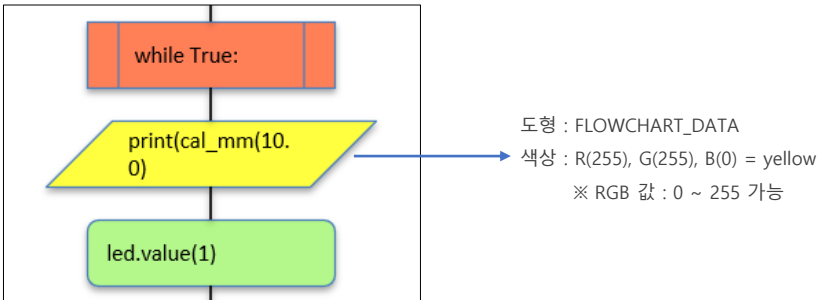
5-3-6. Flowchart 도형 사용자 설정

px-downloader-v100.exe의 Flowchart를 생성 기능에서 사용자는 도형의 모양, 색상을 코드에서 지정할 수 있습니다. 우선, 사용자 도형을 추가할 라인에 주석을 추가하고 대괄호 "[]"안에 도형을 선택, Color값(0~255)을 R,G,B 순서로 작성합니다. [Figure 5-16]은 Excel에서 Flowchart 사용자 도형 사용 코드를 보여줍니다. [Figure 5-17]은 변환된 Flowchart의 도형을 보여줍니다.

[Figure 5-16] Excel에서 Flowchart 사용자 지정 도형 사용 코드

34		while True:				
35			<code>print(cal_mm(10.0))# [FLOWCHART_DATA:255,255,0]</code>			
36			<code>led.value(1)</code>			

[Figure 5-17] Flowchart 사용자 지정 도형 변환



5-3-7. Flowchart 도형 사용자 설정에서 주석 추가하기

사용자는 도형의 모양, 색상을 코드에서 지정하고 Flowchart에 주석 박스를 동시에 생성할 경우, 사용자는 주석 내용과 도형 지정 내용(대괄호"[]")을 동시에 삽입하여 생성할 수 있습니다. 우선, 주석을 시작하고 ('#' 삽입) 주석 도형 안에 들어갈 내용을 작성합니다. 다음, 도형 지정 내용(대괄호"[]")을 작성하여 해당 동작을 수행할 수 있습니다. 띄어쓰기를 '#, 주석 박스 내용, 도형 지정 내용' 사이에 추가해야 합니다. 샘플 코드의 "sample_flowchart_shape.xlsm"을 참고할 수 있고 변환된 Power Point 파일은 "sample_flowchart_shape.pptx"로 샘플 코드와 함께 제공됩니다. [Figure 5-18]은 Excel에서 Flowchart 주석 박스, 사용자 도형 동시 사용 코드를 보여줍니다. [Figure 5-19]는 변환된 Flowchart의 사용자 지정 도형과 주석 박스를 보여줍니다.

[Figure 5-18] Excel에서 Flowchart 사용자 지정 도형, 주석 박스 동시 사용 코드

26		while True:							
27			print(1) # 주석처리 부분	[FLOWCHART_ALTERNATE_PROCESS:255,0,0]					
28			print(2) #	[FLOWCHART_CARD:255,0,0]					

주석 박스 생성

[Figure 5-19] Flowchart 사용자 지정 도형과 주석 박스


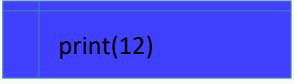

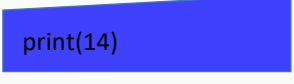
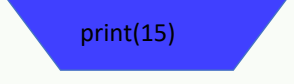

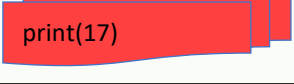

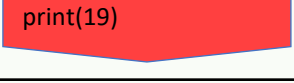
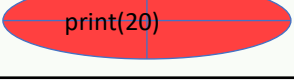


5-3-8. Flowchart 도형 리스트1

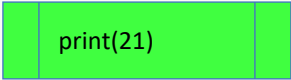
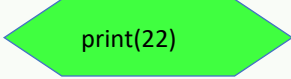


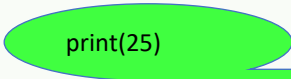


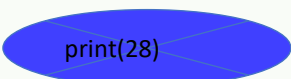
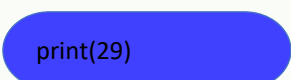
px-downloader-v100.exe의 Flowchart를 생성 기능에서 사용할 수 있는 도형 리스트 입니다. 도형 색상은 5개씩 Red, Green, Blue로 표현되어 있습니다. 샘플 코드의 "sample_flowchart_shape.pptx"에 관련 도형이 저장되어 있습니다. 사용자 지정 도형 사용 시 참고할 수 있습니다.

SHAPE 이름	SHAPE 모양
FLOWCHART_ALTERNATE_PROCESS RGB(255,0,0)	
FLOWCHART_CARD RGB(255,0,0)	
FLOWCHART_COLLATE RGB(255,0,0)	
FLOWCHART_CONNECTOR RGB(255,0,0)	
FLOWCHART_DATA RGB(255,0,0)	
FLOWCHART_DECISION RGB(0,255,0)	
FLOWCHART_DELAY RGB(0,255,0)	
FLOWCHART_DIRECT_ACCESS_STORAGE RGB(0,255,0)	
FLOWCHART_DISPLAY RGB(0,255,0)	
FLOWCHART_DOCUMENT RGB(0,255,0)	

5-3-9. Flowchart 도형 리스트2

SHAPE 이름	SHAPE 모양
FLOWCHART_EXTRACT RGB(0,0,255)	 print(11)
FLOWCHART_INTERNAL_STORAGE RGB(0,0,255)	 print(12)
FLOWCHART_MAGNETIC_DISK RGB(0,0,255)	 print(13)
FLOWCHART_MANUAL_INPUT RGB(0,0,255)	 print(14)
FLOWCHART_MANUAL_OPERATION RGB(0,0,255)	 print(15)
FLOWCHART_MERGE RGB(255,0,0)	 print(16)
FLOWCHART_MULTIDOCUMENT RGB(255,0,0)	 print(17)
FLOWCHART_OFFLINE_STORAGE RGB(255,0,0)	 print(18)
FLOWCHART_OFFPAGE_CONNECTOR RGB(255,0,0)	 print(19)
FLOWCHART_OR RGB(255,0,0)	 print(20)

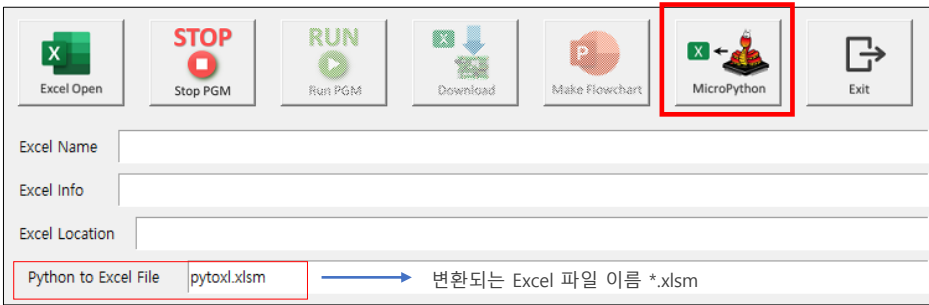
5-3-10. Flowchart 도형 리스트3

SHAPE 이름	SHAPE 모양
FLOWCHART_PREDEFINED_PROCESS RGB(0,255,0)	
FLOWCHART_PREPARATION RGB(0,255,0)	
FLOWCHART_PROCESS RGB(0,255,0)	
FLOWCHART_PUNCHED_TAPE RGB(0,255,0)	
FLOWCHART_SEQUENTIAL_ACCESS_STORAGE RGB(0,255,0)	
FLOWCHART_SORT RGB(0,0,255)	
FLOWCHART_STORED_DATA RGB(0,0,255)	
FLOWCHART_SUMMING_JUNCTION RGB(0,0,255)	
FLOWCHART_TERMINATOR RGB(0,0,255)	

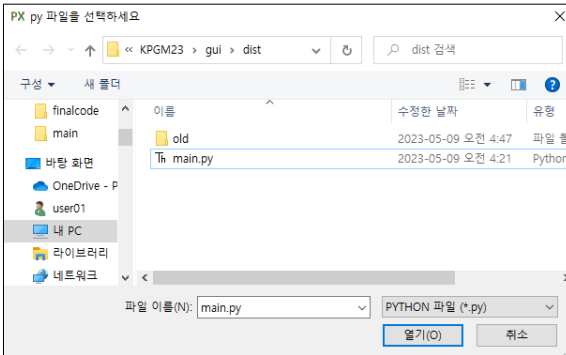
5-3-11. Micro Python To Excel 파일 변환 (*.py to *.xlsm)

px-downloader-v100.exe는 Micro Python 파일을 Excel 파일로 변환할 수 있습니다. MicroPython 버튼을 누르면 *.py를 선택하는 창이 나타나고 Python 파일을 선택 시 지정된 Excel 파일(pytoxl.xlsm)을 생성합니다. 변환 Excel 파일 이름을 "Python to Excel File" 항목에 작성하여 Excel 파일 이름을 변경할 수 있습니다. 변환할 Excel 파일은 *.xlsm확장자를 삽입해 주세요. Excel 파일 변환은 기본 제공되는 "samplevba.xlsm"기준으로 VBA 코드와 sheet 이름(main.py), 프로그램 시작의 설명 부분이 공유됩니다. [Figure 5-20], [Figure 5-20], [Figure 5-21]은 Micro Python 파일을 선택하고 Excel 파일로 변환하는 방법을 보여줍니다.

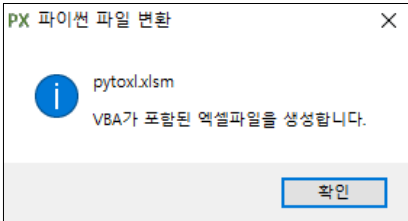
[Figure 5-20] Micro Python to Excel 파일 변환



[Figure 5-21] Python 파일 선택 화면



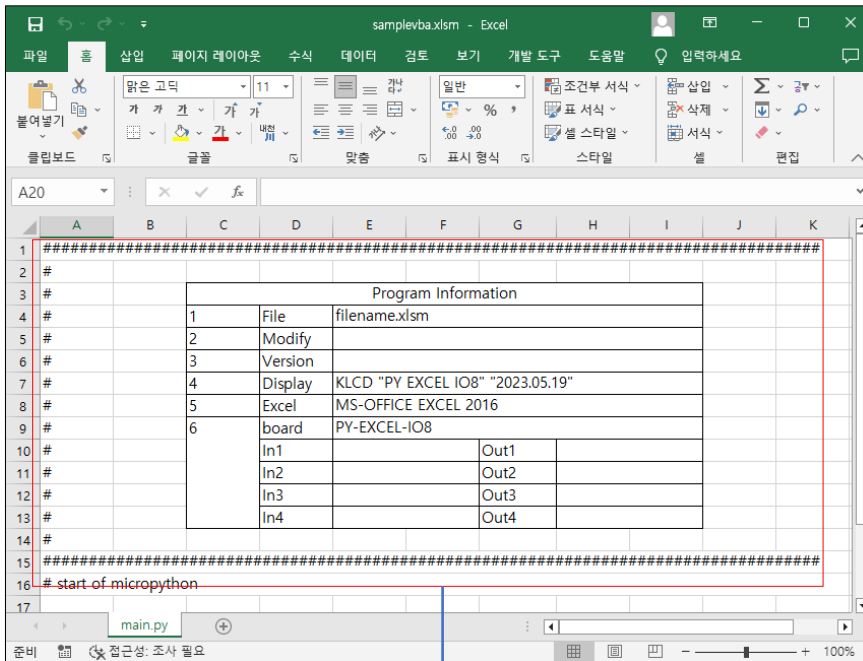
[Figure 5-21] pytoxl.xlsm 파일 변환 시작



5-3-12. Micro Python To Excel 파일 변환 (samplevba.xlsm)

[Figure 5-22]는 기본 제공되는 "samplevba.xlsm"로 파일 변환 시 VBA 코드와 sheet 이름(main.py), 프로그램 시작의 설명 부분이 공유됩니다. 프로그램 시작의 설명 부분은 아래에 작성될 프로그램을 설명하는 부분이며 프로그램 날짜, 주요 특징, 포트 이름, 파일 버전 등 일반적인 내용을 추가하는 데 사용합니다. 사용자가 필요시 sample.xlsm을 수정할 수 있습니다. Sheet 이름 "main.py"는 변경할 수 없고 첫 번째 Sheet가 아니면 정상적인 변환이 되지 않습니다. 또한, VBA 코드를 수정 시 신중히 진행 하시기 바랍니다.

[Figure 5-22] samplevba.xlsm 파일



프로그램 시작 설명 부분
 날짜, 주요 특징, 포트 이름, 파일 버전 등 정보
 사용자 변경 가능하고 '#'삽입하여
 주석 처리해야 합니다.

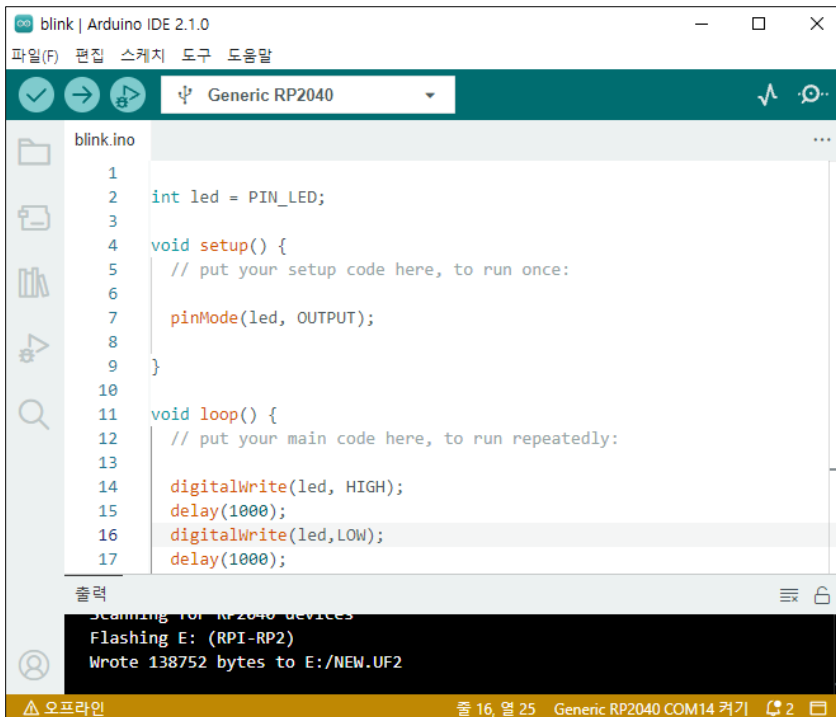
6. Arduino IDE 개발 툴

6-1. 시작하기

Arduino IDE(아두이노 통합개발환경)는 편집기, 컴파일러, 업로더 등이 합쳐진 소프트웨어 환경입니다. Arduino IDE는 아두이노 보드를 포함한 많은 보드를 개발할 수 있는 개발환경으로 PY-EXCEL-IO8에 사용된 RP2040 프로세서도 사용할 수 있습니다. C언어 개발 도구임에도 디버거를 사용하지 않고 USB로 프로그래밍이 가능하여 쉽게 시작할 수 있습니다. 이 장에서는 Arduino IDE 환경에서 PY-EXCEL-IO8 제어보드를 연결하고 프로그래밍하는 방법을 설명합니다. 단, Arduino IDE의 C언어를 사용 시 Excel을 사용할 수 없습니다.



[Figure 6-1] Arduino IDE에서 PY-EXCEL-IO8 연결



6-2. PY-EXCEL-IO8 보드 연결하기

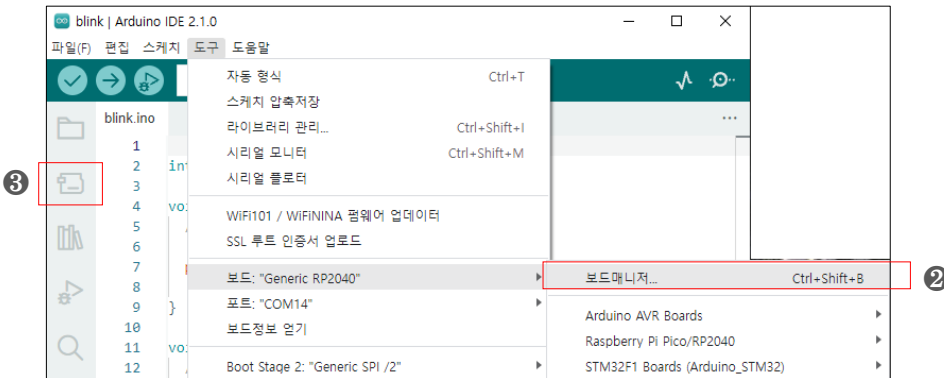
Arduino IDE에서 PY-EXCEL-IO8 제어보드를 연결하는 방법을 알아봅니다. PY-EXCEL-IO8와 PC를 연결하는 방법은 Thonny IDE와 동일하나 처음 연결을 시도한다면 Micro USB 커넥터 좌측 "BOOT" 스위치를 누르고 Power On 하십시오.

6-2-1. Arduino IDE 보드 추가

[Figure 6-2] Arduino IDE 기본 설정/추가 보드 관리자 등록

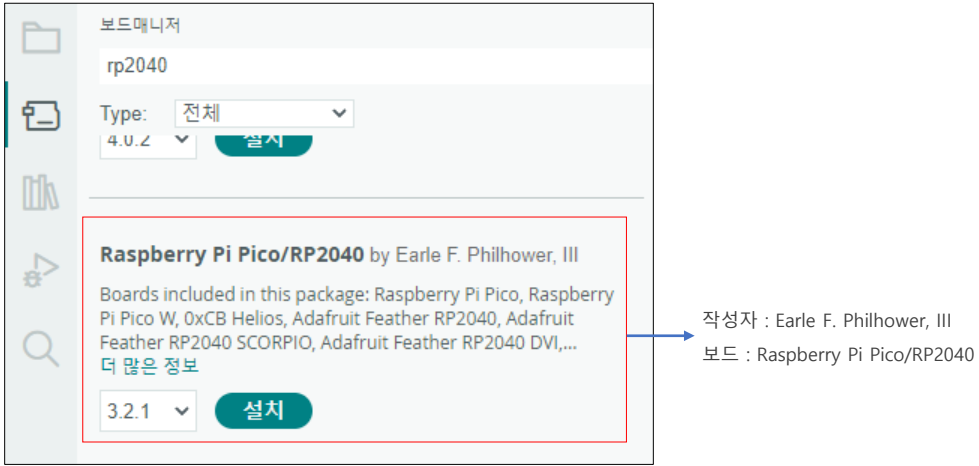


[Figure 6-3] Arduino IDE 보드매니저

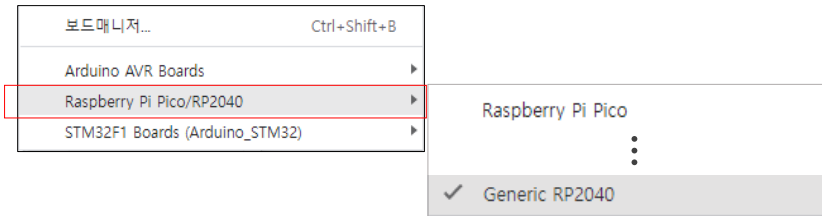


[Figure 6-2]는 Arduino IDE에서 PY-EXCEL-IO8 제어보드를 처음 연결 시 보드 추가하는 방법을 보여줍니다. 프로그램 상단의 '파일/기본 설정'을 선택 시 [Figure 6-2]와 같은 화면이 나옵니다. ① 추가 보드 관리자 URL을 선택하고 https://github.com/earlephilhower/arduino-pico/releases/download/global/package_rp2040_index.json을 삽입합니다. 다음, [Figure 6-3]과 같이 ②도구/보드/보드매니저'를 선택하거나 좌측 ③ 보드매니저 아이콘이 있으면 선택하세요.

[Figure 6-4] 보드매니저 RP2040 검색



[Figure 6-5] 보드 RP2040 검색

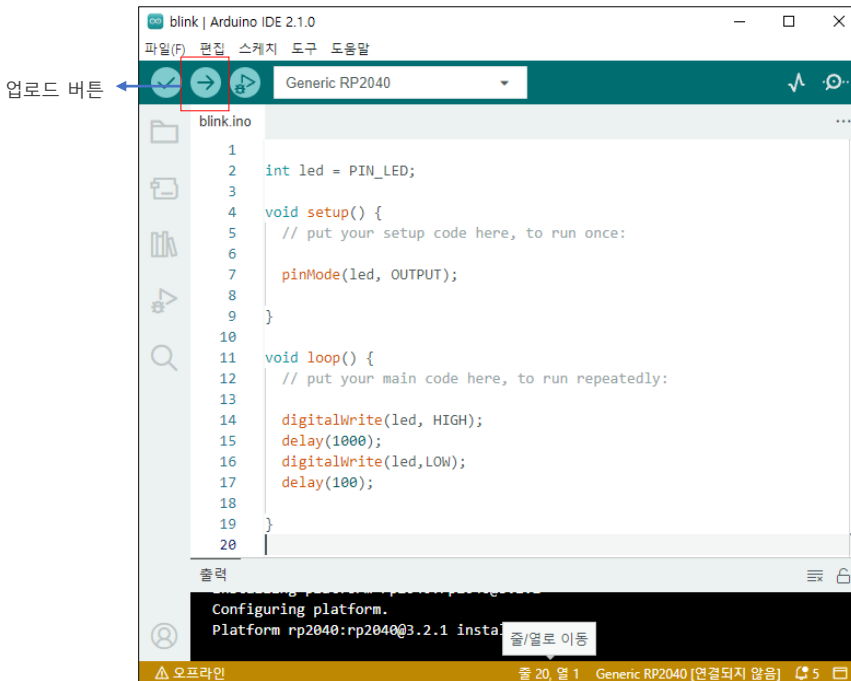


[Figure 6-4]는 보드매니저에서 "RP2040"을 검색하여 'Raspberry Pi Pico/RP2040 by Earle F. Philhower, III'를 설치합니다. 설치 완료 후 '도구/보드/보드'를 선택하면 [Figure 6-5]와 같이 'Raspberry Pi Pico/RP2040'항목이 추가 되었습니다. 항목 아래에서 'Generic RP2040'을 선택하거나 'Raspberry Pi Pico'를 선택합니다. 이제, 보드 추가는 완료 되었습니다.

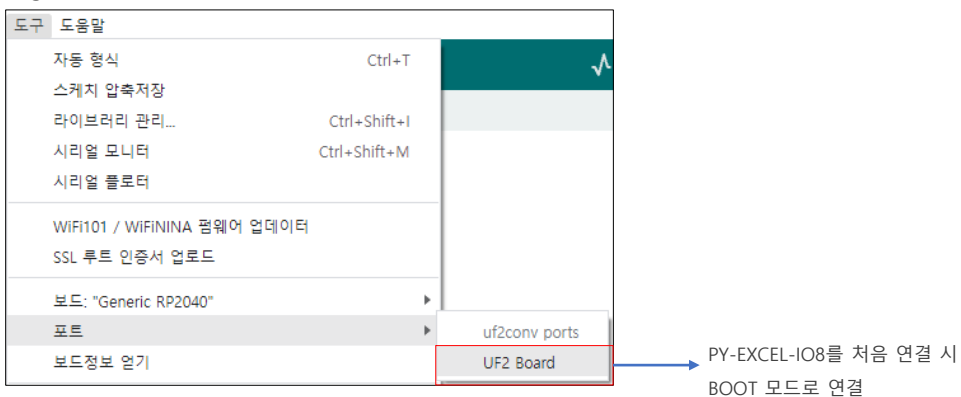
6-2-2. blink.ino 코드 작성, 업로드

Arduino IDE에서 PY-EXCEL-IO8 제어보드를 추가하고 연결 준비를 마쳤습니다. Arduino IDE에서 blink.ino파일을 신규 생성하고 [Figure 6-6]과 같이 코드를 작성하십시오. [Figure 6-7]의 내용은 PY-EXCEL-IO8이 BOOT 모드(Micro USB 좌측 "BOOT"스위치 누르고 Power On)에서 '도구/포트'에서 'UF2 board'를 선택합니다. Arduino IDE 상단 '업로드' 버튼을 누르면 파일을 보드로 다운로드하고 LED가 점멸합니다. 업로드 진행 후 '도구/포트'에서 'UF2 board'는 Serial 포트 'COMx'로 자동 변경되고 코드를 수정할 수 있습니다.

[Figure 6-6] Arduino IDE 코드 작성 blink.ino



[Figure 6-7] PY-EXCEL-IO8 제어보드 BOOT 모드 포트 선택



7. Visual Studio Code 개발 툴

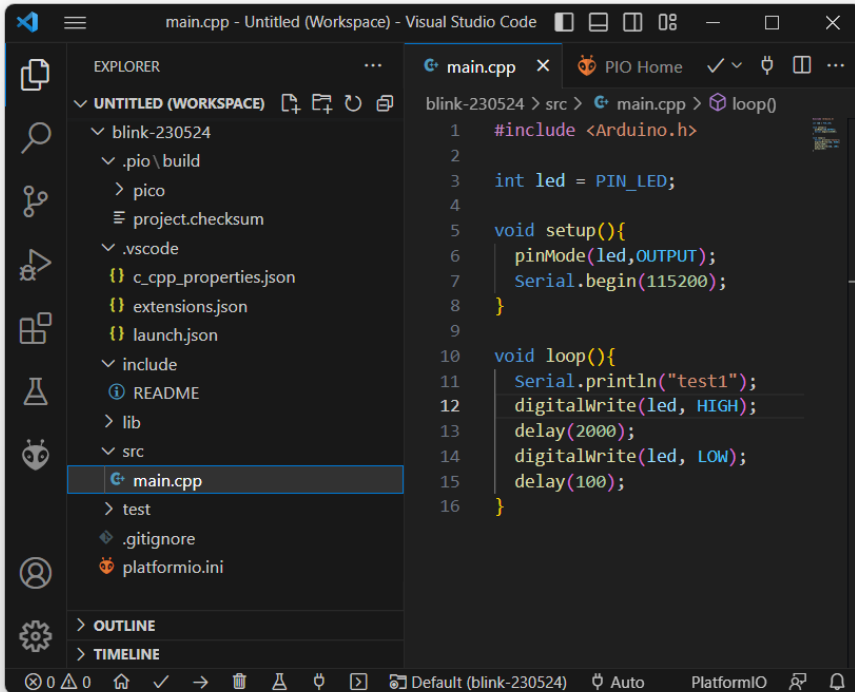
7-1. 시작하기

Visual Studio Code(이하 vscode)는 Micro Soft의 텍스트 편집기로 IDE(통합개발환경)은 아니지만 IDE수준까지 확장할 수 있는 Extension(확장 기능)이라는 가장 큰 장점을 가지고 있는 소프트웨어입니다. 범용으로 사용되는 대부분의 프로그래밍 언어를 사용할 수 있으며 우리가 해볼 내용은 6장에서 코딩한 Arduino기반의 코드를 vscode에서 구현해 보도록 하겠습니다. Arduino IDE에서와 마찬가지로 C언어로 프로그래밍하여 Excel을 사용할 수 없습니다.



Visual Studio Code

[Figure 7-1] Visual Studio Code에서 PY-EXCEL-IO8 프로그래밍

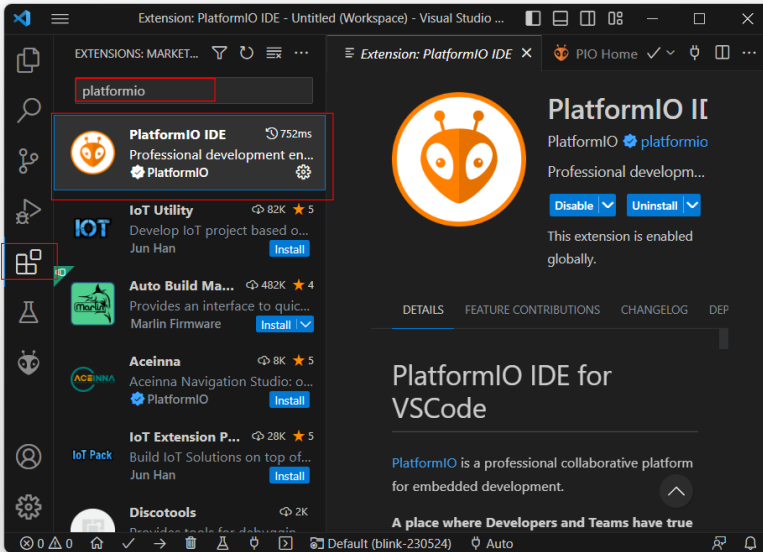


7-2. Visual Studio Code에서 PlatformIO IDE(Extension 설치)

Vscode는 Extension이라는 가장 강력한 기능을 가지고 있습니다. PlatformIO IDE라는 Extension을 설치하고 별도의 프로그램 추가 설치 없이 코딩을 시작할 수 있습니다.

7-2-1. PlatformIO IDE 설치

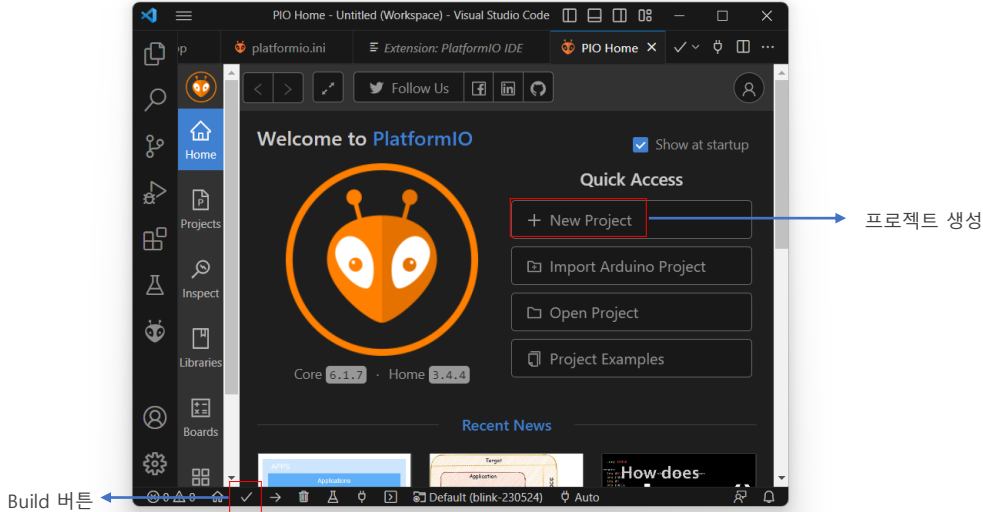
[Figure 7-2] Vscode에서 PlatformIO IDE검색



[Figure 7-2]는 Vscode에서 PlatformIO IDE를 검색하고 설치하는 화면입니다. 좌측 기능바에서 Extension(확장)을 선택 후 'platformio'를 검색하여 PlatformIO IDE를 설치합니다.

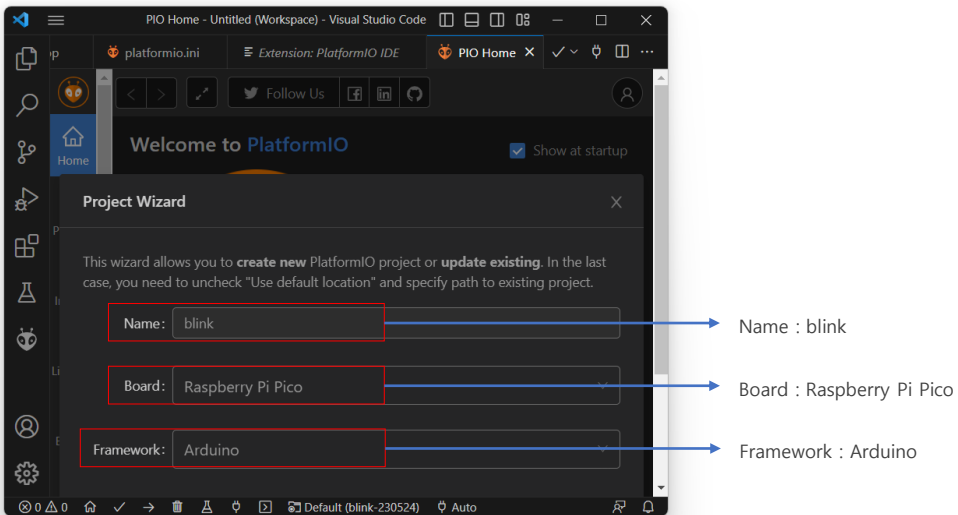
7-2-2. PlatformIO IDE blink 프로젝트 생성

[Figure 7-3] PlatformIO IDE Home화면



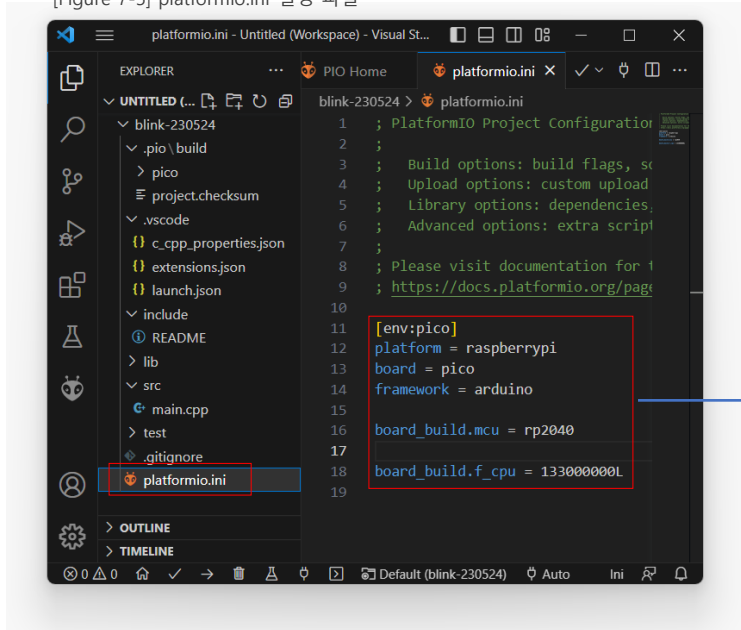
[Figure 7-3], [Figure 7-4]는 PlatformIO IDE에서 신규 프로젝트를 생성합니다. Home 탭에서 'New Project'를 선택하고 Project Wizard가 나타나면 Name : blink, Board : Raspberry Pi Pico, framework : Arduino를 삽입합니다. 이제 blink 프로젝트가 생성되었습니다.

[Figure 7-4] blink 프로젝트 생성



7-2-3. blink 프로젝트 소스 코드, 설정 파일

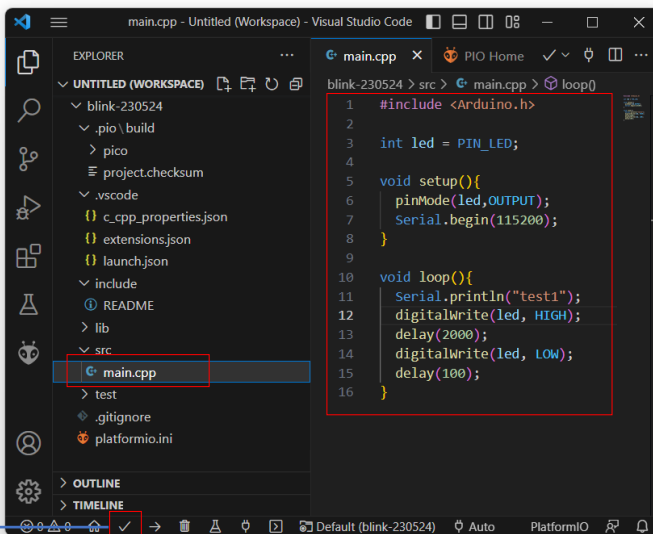
[Figure 7-5] platformio.ini 설정 파일



platform = raspberrypi
board = pico
framework = arduino
board_build.mcu = rp2040
board_build.f_cpu = 133000000L

[Figure 7-5]는 platform.ini 설정 파일로 보드의 설정 내용을 삽입할 수 있습니다. 그림과 같이 삽입, 수정 해주세요.

[Figure 7-6] main.cpp 소스코드



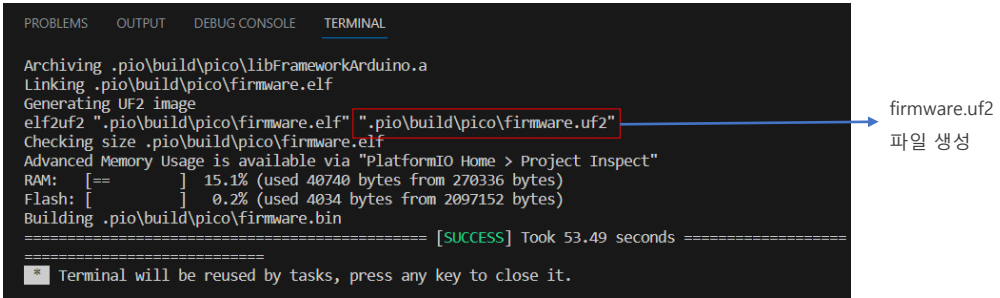
[Figure 7-6]는 main.cpp 소스 코드입니다. Arduino IDE에서 사용한 blink와 동일한 코드를 추가하고 하단 좌측 'v'(build)를 눌러 컴파일을 시작합니다.

Build 버튼

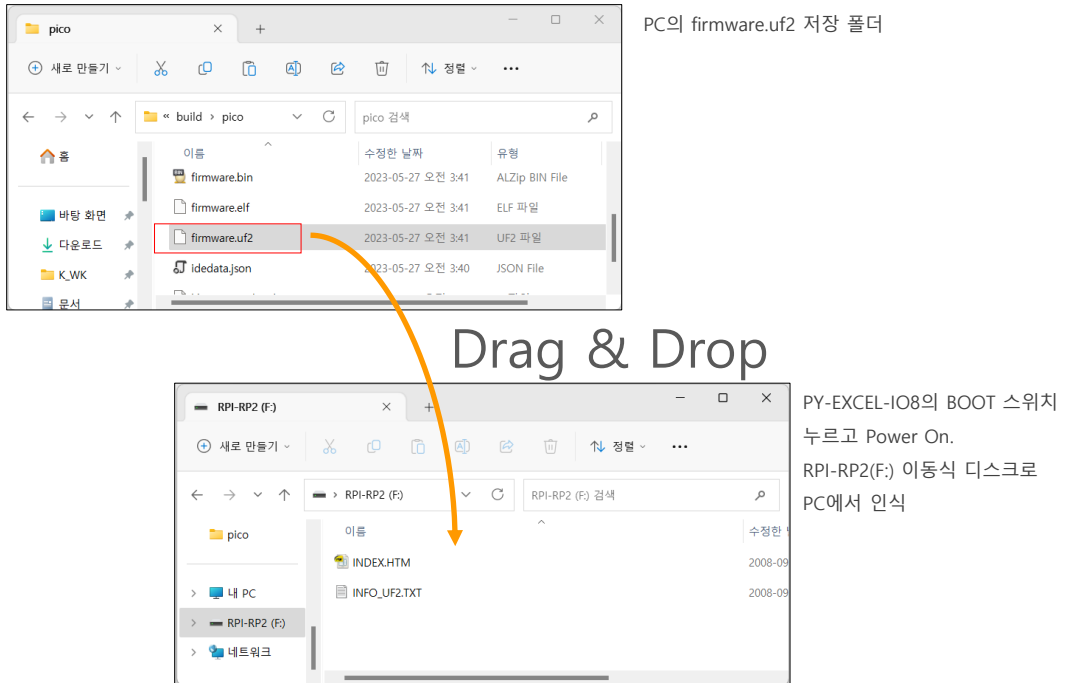
7-3. Build 완료 파일 저장

[Figure 7-7]은 Vscode에서 PlatformIO IDE가 Build 완료 시 터미널 화면의 내용입니다. "SUCCESS"가 출력되어 있고 "firmware.uf2"파일이 생성되어 있습니다. 이 파일을 PY-EXCEL-IO8 보드에 저장할 수 있습니다. [Figure 7-8]은 Build 된 파일을 보드에 저장하는 방법입니다. 먼저, PY-EXCEL-IO8의 Micro USB 케이블을 PC에 연결 후 'BOOT' 스위치를 누르고 Power On 합니다. PC에서 RPI-RP2(F)라는 이동식 디스크로 인식됩니다. 다음, firmware.uf2파일을 이동식 디스크로 Copy하거나 Drag합니다. PY-EXCEL-IO8은 자동으로 LED가 점멸합니다. (프로그램 실행)

[Figure 7-7] Vscode에서 Build Success



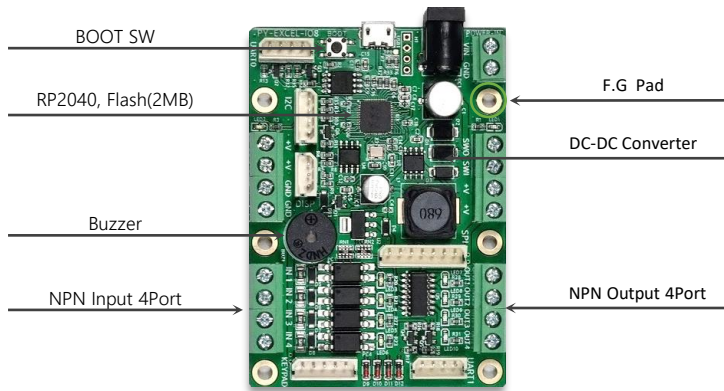
[Figure 7-8] firmware.uf2파일 저장



8. PY-EXCEL-IO8 제어보드

8-1. 보드 사양

PY-EXCEL-IO8 제어보드는 RP2040 프로세서를 사용합니다. RP2040은 Raspberry Pi에서 개발한 첫 번째 MCU로 동급 프로세서보다 큰 저장용량과 빠른 Clock Speed, 기타 유용한 기능을 가지고 있습니다. RP2040을 사용한 여러 형태의 제어보드가 있지만 실전, 산업용 제어보드는 PY-EXCEL-IO8이 유일합니다. 이 밖에, 일반 산업용 전압(DC12V~24V) 사용과 NPN 센서 입력, NPN 출력이 지원되고 UART, I2C, SPI 통신 커넥터로 기능 확장이 용이합니다.



[Table 7-1] PY-EXCEL-IO8 제어보드 사양

항목	사양
MCU	ARM 프로세서 32bit (Cortex M0+) <ul style="list-style-type: none"> ▪ RP2040 ▪ External FLASH : 2MB SRAM : 264KB
Power Supply	12VDC ~ 24VDC
DC Power Output	<ul style="list-style-type: none"> ▪ DC 5V - 1.5A max, Under 1.0A recommended ▪ DC 3.3V – 500mA max, Under 300mA recommended
Output	Darlington Transistor(500mA max) * 4 Port , Buzzer
Input	Photo Coupler Input * 4 Port , Key * 4 Port
Display	<ul style="list-style-type: none"> ▪ LC-KLCD-R2 (Character LCD) ▪ LC-FND4-056 (FND 4 Digit) ▪ DWIN Display, Nextion Touch Screen (4.3 Inch, 7.0 Inch)
Peripheral	UART * 2 Ports, SPI * 1Port, I2C * 1Port
Memory	External EEPROM 24C64 (64Kbit)

8-1-1. BOOT SW (Micro Python 펌웨어)

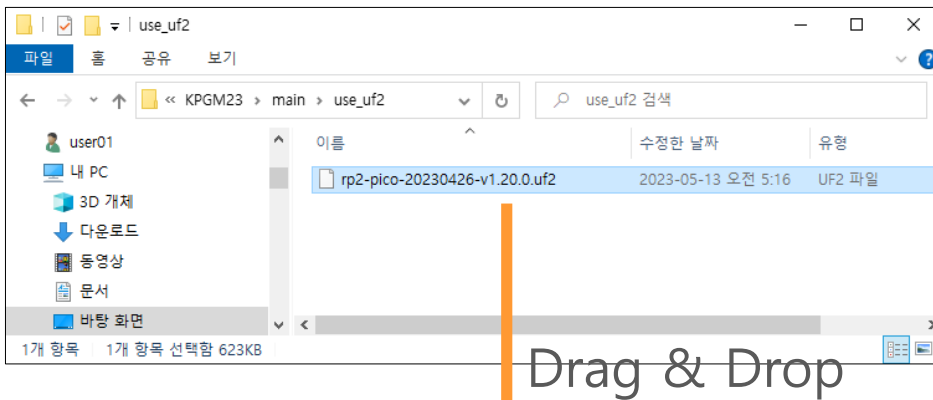
BOOT SW는 Micro Python 펌웨어를 다운로드하는 데 사용됩니다. USB 케이블을 PC에 연결 후 BOOT SW를 누르고 Power On 시 PC에서 USB 메모리로 인식하게 됩니다. 이 상태에서 Micro Python 펌웨어를 USB 메모리로 Drag & Drop 하면 보드는 자동으로 Micro Python을 실행하고 PC에서 보드를 Com Port로 인식됩니다. 이제 PY-EXCEL-IO8 제어보드에 Micro Python 프로그래밍을 수행할 수 있습니다. PY-EXCEL-IO8은 출하 시 펌웨어가 다운로드 되어있고 기본 프로그램(main.py)과 lib 폴더에 확장 파일이 저장된 상태로 출고됩니다.

참고

DOWNLOAD FILE : [rp2-pico-20230426-v1.20.0.uf2](#)

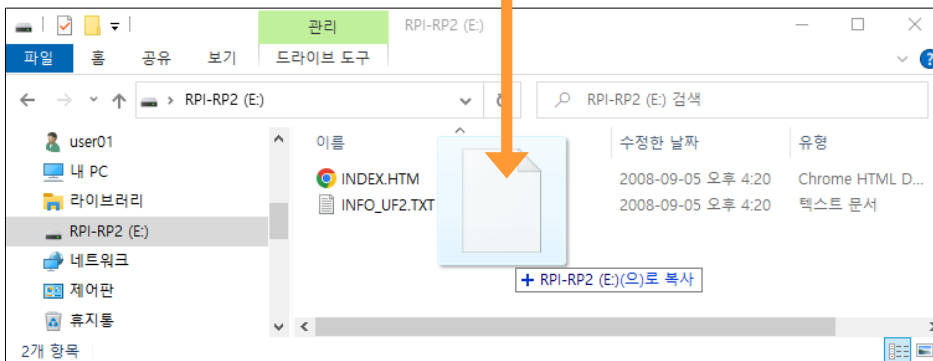
새로운 버전의 rp2-pico(RP2040용) *.uf2 파일이 사용될 수 있습니다. 펌웨어 파일을 업데이트할 경우 동작 테스트 해보시기 바랍니다. (업데이트 후에도 main.py, lib 폴더는 삭제되지 않습니다.)

[Figure 7-1] PC에 저장된 Micro Python 펌웨어 파일



Drag & Drop

[Figure 7-2] PC에서 USB 메모리로 인식된 PY-EXCEL-IO8 제어보드 폴더

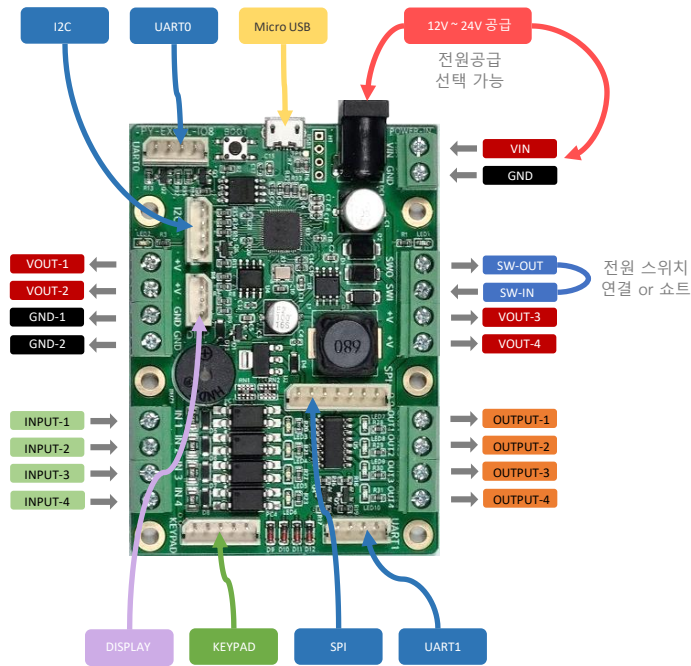


8-2. PY-EXCEL-IO8 하드웨어

PY-EXCEL-IO8 제어보드는 실무, 산업현장에서 발생하는 요구를 빠르게 대처할 수 있도록 산업용 입출력 기능과 다양한 통신 포트를 내장하고 있습니다.

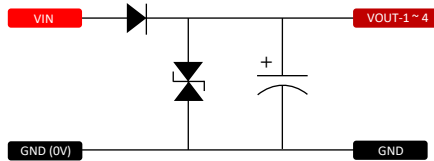
8-2-1. 전체 PERIPHERAL CONNECTION 배치도

PY-EXCEL-IO8 입출력, 통신 커넥터 배치도입니다.



8-2-2. POWER SUPPLY

VIN,GND(0V)는 외부 전원 입력포트 입니다. 허용 범위는 DC 12V ~ DC 24V이고 DC-JACK과 연결되어 있습니다. VOUT1~4 전원 출력과 제어보드 내부 사용 전력의 합이 3.0A를 넘지 않아야 합니다. 보호 회로가 포함되어 제어보드의 손상을 방지합니다.

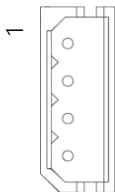


항목	NAME	SPEC
Power Supply	VIN	12VDC ~ 24VDC
Power Output	VOUT-1,2,3,4	MAX 3.0A

8-2-3. UART0,1

UART0, UART1 통신 커넥터입니다. UART Serial 통신은 가장 보편적인 통신의 하나로 사용하기 쉽고 안정적입니다. DC5.0V-1.0A 전원 출력이 가능하며 소비전력이 큰 터치스크린(DWIN Display, Nextion Display)에 Direct 연결하고 TX,RX로 제어할 수 있습니다. XY-EXTIN-NPN8(입력확장), XY-EXTOUT-TR8(출력확장)을 제어하는 소스 코드가 제공됩니다.

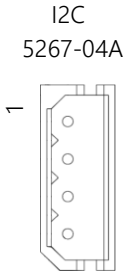
UART0,1
5267-04A



NO	NAME	DESCRIPTION	DIR
1	GND	External Power Supply(-), GND	OUT
2	RX	UARTx Receive Signal, 5V TTL Level	IN
3	TX	UARTx Transmit Signal, 5V TTL Level	OUT
4	5VDC	External Power Supply(+), DC 5.0V	OUT

8-2-4. I2C

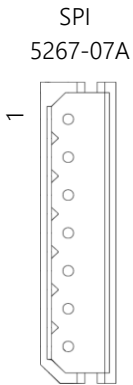
I2C 통신 커넥터로 I2C는 2개 라인으로 n 대의 대상을 제어할 수 있어 확장이 용이합니다. L3G-4200D 자이로센서 보드(GY50)를 제어하는 소스 코드가 제공됩니다.



NO	NAME	DESCRIPTION	DIR
1	GND	External Power Supply(-), GND	OUT
2	SDA0	I2C0 Data, 3.3V Level	IN/OUT
3	SCL0	I2C0 Clock, 3.3V Level	OUT
4	3.3VDC	External Power Supply(+), DC 3.3V	OUT

8-2-5. SPI

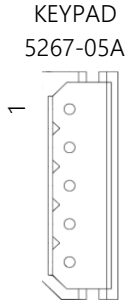
SPI 통신 커넥터로 기본 CS0(Chip Select) 외 추가 SCS(Spare Chip Select) 포트가 포함되어 2개의 SPI 보드에 연결 할 수 있습니다. SPI 통신은 I2C보다 빠른 송수신이 가능하여 보다 큰 데이터 처리에 적합합니다. XY-STEP-TMC5160A 모션보드를 제어하는 소스 코드가 제공됩니다.



NO	NAME	DESCRIPTION	DIR
1	GND	External Power Supply(-), GND	OUT
2	SCS	Spare Chip Select	OUT
3	CS0	Chip Select	OUT
4	MOSI0	SPI0 Master Out Slave In	IN/OUT
5	MISO0	SPI0 Master In Slave Out	IN/OUT
6	SCK0	SPI0 Clock	OUT
7	3.3VDC	External Power Supply(+), DC 3.3V	OUT

8-2-6. KEYPAD

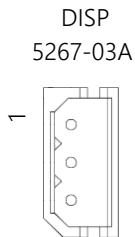
KEYPAD 입력 포트에 Switch 4개를 입력 받아 Display, 기능 제어 시 유용합니다. LC-M4-R11 (switch 4port) 보드를 연결할 수 있습니다.



NO	NAME	DESCRIPTION	DIR
1	GND	Switch Common, GND	OUT
2	SW4	Switch 4 (Key Down)	IN
3	SW3	Switch 3 (Key Up)	IN
4	SW2	Switch 2 (Key →)	IN
5	SW1	Switch 1 (Key Set)	IN

8-2-7. DISP

DISP 포트에 간편하게 Display를 추가할 수 있습니다. LC-KLCD-R2 (캐릭터 LCD 1602, 2004), LC-FND4-056-RV1 (7 Segment FND-4ea, 최대 16자리 표시)를 기본 사용할 수 있고 소스 코드가 제공됩니다.



NO	NAME	DESCRIPTION	DIR
1	GND	Display Power Supply(-), GND	OUT
2	DISP	Display Signal, UART TX	OUT
3	5VDC	Display Power Supply(+), DC 5.0V	OUT

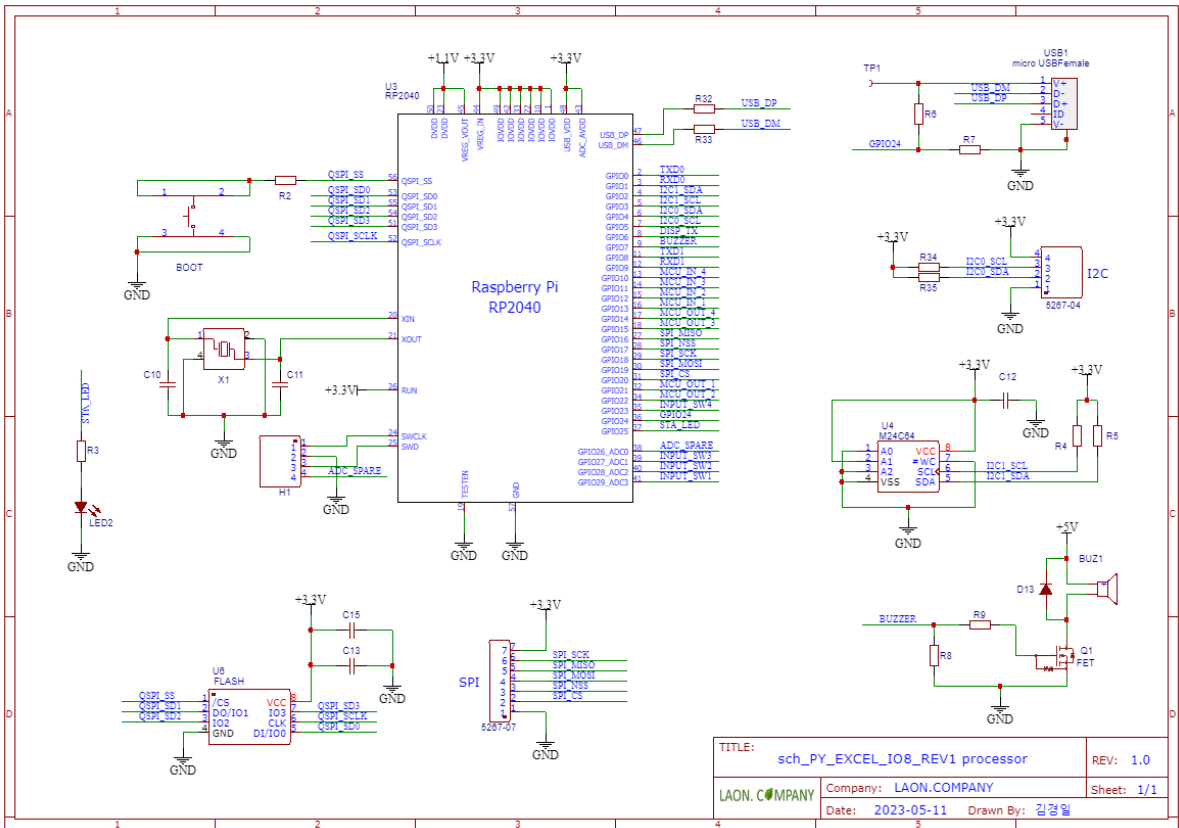
8-3. PY-EXCEL-IO8 회로도

PY-EXCEL-IO8 제어보드의 MCU, INPUT, OUTPUT 회로도를 제공합니다.

8-3-1. MCU 회로도

PY-EXCEL-IO8 제어보드 MCU 회로도는 입출력, 통신 커넥터 포트의 PIN 넘버를 확인할 수 있습니다.

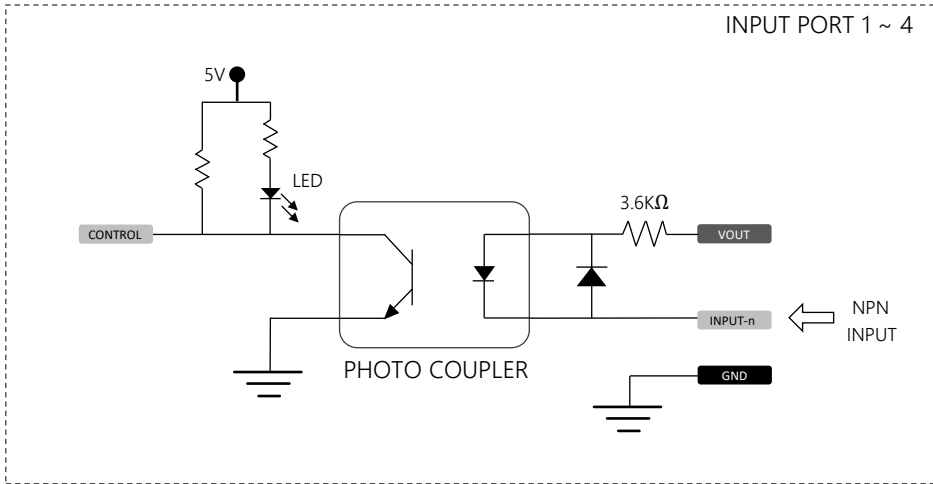
[Figure 7-3] PY-EXCEL-IO8 processor 회로도



8-3-2. INPUT 회로도

PY-EXCEL-IO8 제어보드 INPUT포트 회로도 입니다. Photo Coupler Isolation은 제어부와 센서부를 절연하여 제어의 안정성을 향상시킵니다. 동일한 4개의 포트에 Switch 연결, Relay 접점 입력, NPN-TYPE의 센서를 연결할 수 있습니다.

[Figure 7-4] PY-EXCEL-IO8 INPUT 회로도



항목	SPEC
NPN INPUT	Photo Coupler Isolation 4PORT

※ 각종 수동, 자동입력 부품 연결 가능



스위치



말굽 센서



에어리어 센서



포토 센서



광 화이버 센서

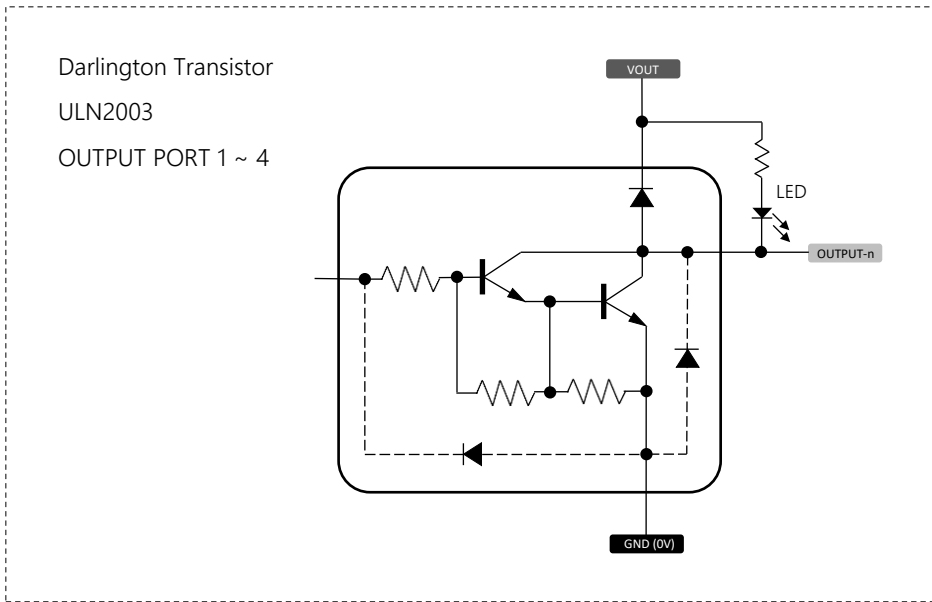


도어 센서

8-3-3. OUTPUT 회로도

PY-EXCEL-IO8 제어보드 OUTPUT포트 회로도 입니다. ULN2003 chip에서 4개의 포트가 출력됩니다. 릴레이, Sol-Valve, SSR, 타워램프, 타이머 등 출력 제어가 가능합니다.

[Figure 7-5] PY-EXCEL-IO8 OUTPUT 회로도



항목	SPEC
NPN OUTPUT	~ 500mA (Max) 4PORT

※ 각종 출력 제어 부품 연결 가능



솔레노이드 에어 밸브



SSR



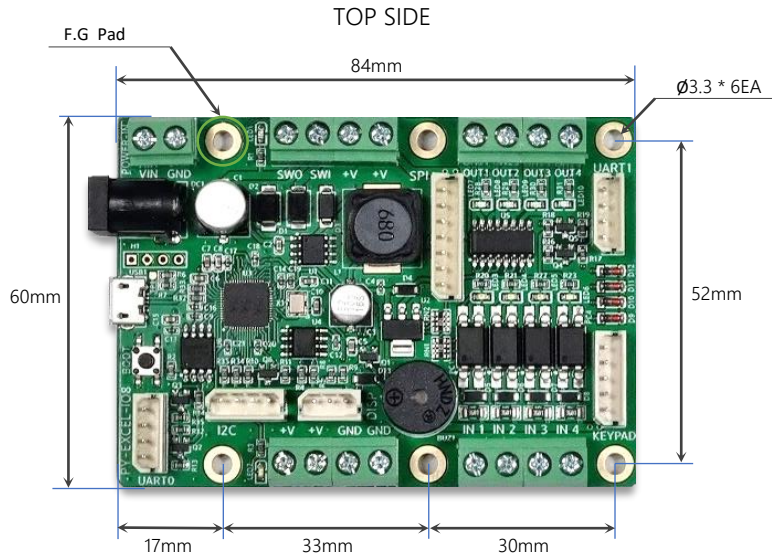
릴레이



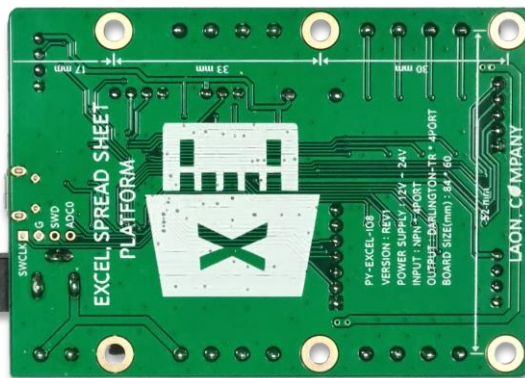
타워램프

8-4. PY-EXCEL-IO8 DIMENSIONS

PY-EXCEL-IO8 제어보드 사이즈입니다. F.G(Frame Ground, 접지)는 VIN,GND 전원 투입 포트 측 고정 홀을 사용하고 Common Ground와 연결되어 있습니다. 보드 고정홀은 $\varnothing 3.3 * 6EA$ 이며 M3 나사를 사용할 수 있습니다.



BOTTOM SIDE



9. 개정 이력

No	항목	내용	Revision	개정일
1	데이터시트	PY-EXCEL-IO8 보드 버전 : REV1 최초작성	VERSION 1.0	2023.06.10
2				