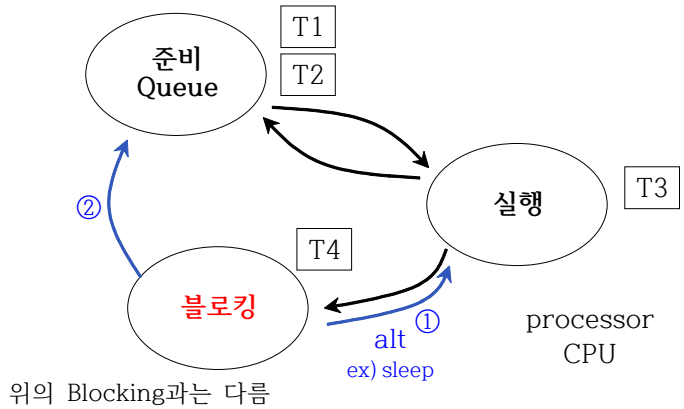


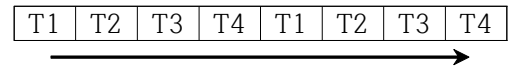
Process

Command	① Blocking - 함수 실행 뒤 다음 함수 실행	
Mode	② NonBlocking // while사용x	- system pooling
	③ Asynk - 콜백 // 쓰레드사용	

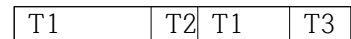
OS 운영체제의 스케줄에 따라 진행



① 라운드 로빈



② 선점형



sleep을 줘야 다른 것도 작동가능.. (sleep없이 CPU점유해봤자 빠른건 아님)
적당히(ms, ns) sleep을 주고 동기화해야함...?

[VS코드 최신업데이트 하기]

VisualStudio사이트 다운로드(Other Download) <https://code.visualstudio.com/#alt-downloads>
zip파일(64비트) 다운받고 압축풀기

C:\SodaIDE에서 data ext만 빼고 모두 삭제
압축푼 것 중 bin을 제외하고 모두 SodaIDE 폴더로 이동

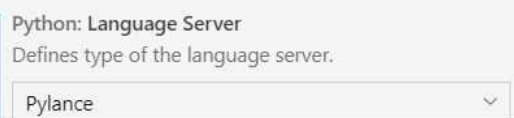
data폴더 - extensions에는 확장프로그램이 들어있음

data 폴더 / suer-data / User만 빼고 모두 삭제
User폴더 안에 settings.json, globalStorage를 남기고 모두 삭제

data/extensions에 pylance가 자동완성기능 가지고있음 (자동업데이트 됨)

vs코드 실행 후 Settings에서

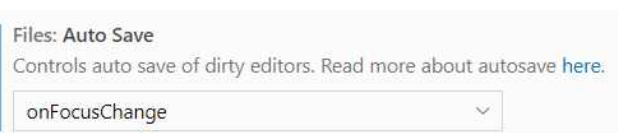
- 검색 : **Language Server** 'Pylance'선택



- 검색 : **Mouse Wheel** ✓(Mouse-zoom기능)



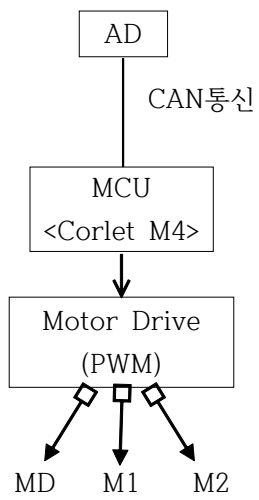
- 검색 : **Auto Save** 'onFocusChange'



=> C:/SodaIDE/data/user-data/User/settings.json 에 설정이 저장됨

VS Code - SerBOT 연결(soda@192.168.55.1)

qt5폴더생성 -> [first.py] 코드작성



[PyQT]

QT5	→	QT6
PySide2, PyQT5		PyQT6
		PySide6

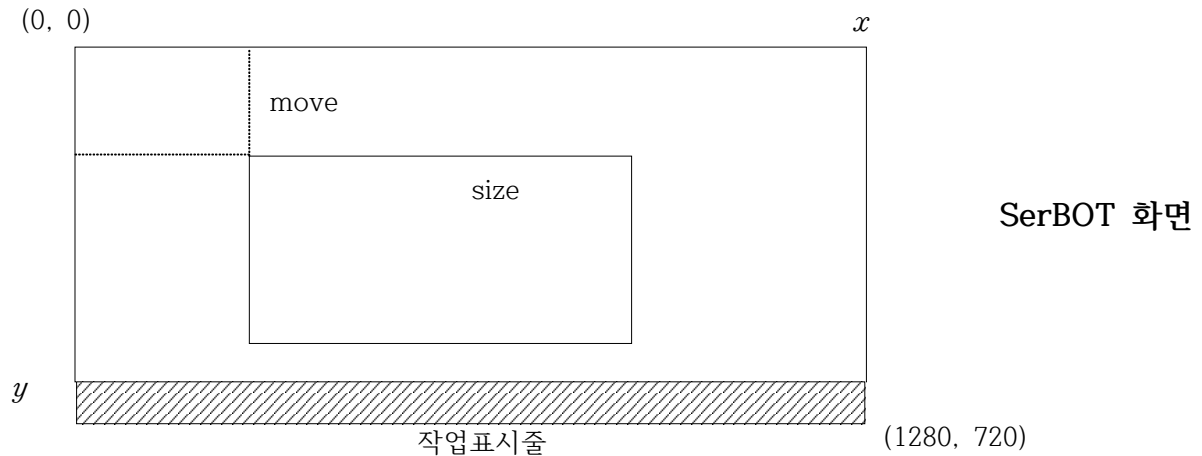
현재까진 matplotlib은 QT5가 적용되어있음..(나중에 6 업글 될지도?)

[PyQT설치]

```
sudo pip3 install pyqt5
```

하드웨어 의존성 없음(linux, win 등)

[second.py] 코드작성



pyqt 사용할 때 main필요

멀티프로세싱을 사용할 때도 main이 필요함 (쓰레드는 아님)

터미널에서

```
export DISPLAY=:0 // 원격지에서 사용할 때(ssh로 접속), 0번 화면에 출력. 터미널 새로만들 때마다 해줘야함
python3 second.py
```

Ctrl+C , Ctrl+\ 강제종료

self.showFullScreen() 전체화면 띄우기

PyQT 튜토리얼 <https://wikidocs.net/book/2165>

```
cb1.stateChanged.connect(self.onCb1Changed)
```

cb1 : 시그널 송신자

stateChanged : 시그널

connect : 송신자-수신자 연결

self : 시그널 수신자

onCb1Changed : 시그널 핸들러

dash-board

☒ Forward

Motor1

Motor2

Motor3

start

stop

Q Design

OmniWheel 라이브러리 들어가서 (import에서 Ctrl+클릭)

생성자에서.. [`def __init__(self):`]

`self._is_read = False` 추가

`super().__init__()` 삭제

소멸자에서.. [`def __del__(self):`]

`super().__del__()` 삭제

...

```
class OmniWheel:
    def __init__(self):
        #super().__init__()
        ...
        self._is_read = False

    def __del__(self):
        #super().__del__()
        ...

    def _readSensor(self):
        #while True:
        while self._is_read:
            ...

    def readStart(self):
        if not hasattr(__main__, ...):
            self._is_read = True
            ...

    def readStop(self):
        if hasattr(self, 'thread'):
            self._is_read = False
            ...
```

Q테이블 위젯

<https://wikidocs.net/128689>

<https://www.riverbankcomputing.com/static/Docs/PyQt5/search.html>

<https://doc.qt.io/qtforpython/PySide6/QtWidgets/index.html>