PingPongThread

환경: 윈도우 10 x64, Python 3.6.6 x64, pyserial 3.4

Introduction

로봇 모터 돌리기

from pingpongthread import PingPongThread

PingPongThreadInstance = PingPongThread(number=1)   # 1개 로봇 연결

PingPongThreadInstance.start()                      # 쓰레드 시작

PingPongThreadInstance.wait\_until\_full\_connect()    # 전부 연결될 때까지 기다림

PingPongThreadInstance.run\_motor()  # 모터 회전

PingPongThreadInstance.wait(5)      # 5초 동안 기다림

PingPongThreadInstance.stop\_motor() # 모터 끔

PingPongThreadInstance.end()        # 쓰레드 종료

로봇 모터 속도 설정

from pingpongthread import PingPongThread

PingPongThreadInstance = PingPongThread(number=2)   # 2개 로봇 연결

PingPongThreadInstance.start()                      # 쓰레드 시작

PingPongThreadInstance.wait\_until\_full\_connect()    # 전부 연결될 때까지 기다림

cube\_ID = "all"     # 모든 모터 선택

speed = [30, -20]   # 1번 큐브 시계 방향 30RPM, 2번 큐브 반시계 방향 20RPM

PingPongThreadInstance.run\_motor(cube\_ID, speed)    # 모터 회전

PingPongThreadInstance.wait(5)                      # 5초 동안 기다림

PingPongThreadInstance.stop\_motor() # 모터 끔

PingPongThreadInstance.end()        # 쓰레드 종료

로봇 모터 스케줄 설정

from pingpongthread import PingPongThread

PingPongThreadInstance = PingPongThread(number=2)   # 2개 로봇 연결

PingPongThreadInstance.start()                      # 쓰레드 시작

PingPongThreadInstance.wait\_until\_full\_connect()    # 전부 연결될 때까지 기다림

cube\_ID = "all"                 # 모든 모터 선택

speed = [[30, -20], [30, -20]]  # 속도 스케줄. 1번 큐브 30RPM, -20RPM, 2번 큐브 30RPM, 20RPM

cycle = [[2, 1], [1, 2]]        # 회전 스케줄. 1번 큐브 2바퀴, 1바퀴, 2번 큐브 1바퀴, 2바퀴

PingPongThreadInstance.run\_motor(cube\_ID,

    speed,

    cycle,

    run\_option="schedule",

    wait="schedule")    # 모터 회전, 스케줄이 끝날 때까지 기다림

PingPongThreadInstance.stop\_motor() # 모터 끔

PingPongThreadInstance.end()        # 쓰레드 종료

API Class

**class** PingPongThread

**function** \_\_init\_\_(number)

**parameter** number - 연결 할 로봇 숫자

함수 인스턴스가 만들어지면 자동으로 동글의 시리얼 포트를 찾는다. 시리얼 포트를 찾을 때 까지 다른 동작은 잠긴다. 인스턴스는 한 개 이상 생성되지 않는다.

**function** start()

쓰레드 시작시킨다. 쓰레드에서 시리얼 값을 읽기 시작한다.

**function** end()

쓰레드를 종료한다. 또한 로봇, 동글과의 연결을 끊는다.

**function** disconnect\_master(discovery\_group=None)

**parameter** discovery\_group - 작업을 실행할 로봇의 그룹

마스터 로봇의 연결을 끊는다.

**function** get\_is\_start()

**return** 쓰레드의 시작 여부, bool

쓰레드가 시작되었는지의 여부를 반환한다.

**function** get\_robot\_status(discovery\_group=None)

**parameter** discovery\_group - 작업을 실행할 로봇의 그룹

**return** 로봇의 스테이터스, dict

controller\_status - 컨트롤러에서 가장 최근 보낸 스테이터스

discovery\_group - 스테이터스의 로봇 그룹

connection\_number - 연결할 로봇의 총 개수

stepper\_mode - 스테퍼 모터의 모드 (continue, step, schedule, point)

stepper\_pause - 스테퍼 모터의 일시정지 여부

stepper\_speed - 스테퍼 모터의 속도 (continue, step 모드)

stepper\_step - 스테퍼 모터의 이동 스텝 (step 모드)

stepper\_speed\_schedule - 스테퍼 모터의 속도 스케줄 (schedule 모 드)

stepper\_step\_schedule - 스테퍼 모터의 스텝 스케줄 (schedule 모 드)

stepper\_schedule\_point\_start - 스테퍼 모터의 포인트 시작 스케줄 (point 모드)

stepper\_schedule\_point\_end - 스테퍼 모터의 포인트 끝 스케줄 (point 모드)

stepper\_schedule\_point\_repeat - 스테퍼 모터의 포인트 반복 횟수 스케줄 (point 모드)

processed\_status - 컨트롤러에서 가장 최근 받은 스테이터스

connected\_number - 현재 연결된 로봇 개수

MAC\_address - 연결된 마스터 로봇의 MAC 주소 뒤의 두 자리

stepper\_schedule\_set - 스테퍼 모터의 스케줄이 설정 되었는지의 여부

stepper\_point\_set - 스테퍼 모터의 포인트 스케줄이 설정 되었는지 의 여부

stepper\_played\_pause - 스테퍼 모터의 재생된 스케줄이 pause 상 태인지의 여부

stepper\_played\_schedule\_idx - 스테퍼 모터의 재생된 스케줄의 인 덱스

stepper\_played\_point\_idx - 스테퍼 모터의 재생된 포인트 스케줄의 인덱스

stepper\_played\_repeat\_idx - 스테퍼 모터의 재생된 포인트 스케줄 의 반복 인덱스

로봇 연결의 스테이터스를 반환한다.

**function** wait\_until\_full\_connect(discovery\_group=None)

**parameter** discovery\_group - 작업을 실행할 로봇의 그룹

로봇이 전부 연결될 때까지 기다린다.

**function** play\_once\_full\_connect(discovery\_group=None)

**parameter** discovery\_group - 작업을 실행할 로봇의 그룹

**return** bool

while문에서, 로봇이 전부 연결되면 한 번만 실행한다.

**function** RPM\_to\_SPS(RPM)

**parameter** RPM - 변환 시킬 속도 RPM

**return** 변환된 속도 SPS, int

RPM 단위의 속도를 SPS로 변환시킨다. RPM의 범위는 -30RPM ~ -3RPM, 0RPM, 3RPM ~ 30RPM. SPS의 범위는 -1000SPS ~ -100SPS, 0SPS, 100SPS ~ 1000SPS.

**function** SPS\_to\_RPM(SPS)

**parameter** SPS - 변환 시킬 속도 SPS

**return** 변환된 속도 RPM, float

SPS단위의 속도를 RPM으로 변환시킨다. SPS의 범위는 -1000SPS ~ -100SPS, 0SPS, 100SPS ~ 1000SPS. RPM의 범위는 -30RPM ~ -3RPM, 0RPM, 3RPM ~ 30RPM.

**function** wait(time\_seconds)

**parameter** time\_seconds - 기다릴 시간(초)

time\_seconds초 동안 기다린다.

**function** run\_motor(cube\_ID\_list="all", speed\_list=30, step\_list=None, pause\_list=False, time\_lis t=None, discovery\_group=None, run\_option="continue", speed\_option="RPM", step\_option="CYCLE", sync=False, time\_option=None, wait=0)

**parameter** cube\_ID\_list - 스테퍼 모터를 실행시킬 로봇 번호의 리스트. "all" 이면 모든 로봇을 작동.

**parameter** speed\_list - 스테퍼 모터의 속도, 또는 속도 스케줄의 리스트. "stop" 또는 "sleep"이면 속도는 0. 단위가 RPM이면 범위는 -3 ~ -30, 0, 3 ~ 30. 단위가 SPS이면 범위는 -1000 ~ -100, 0, 100 ~ 1000.

**parameter** step\_list - 스테퍼 모터의 이동 스텝, 또는 이동 스텝 스케줄의 리스트. 단위가 CYCLE이면 범위는 -32.7675 ~ 32.7675. 단위가 STEP이면 범 위는 -65535 ~ 65535.

**parameter** pause\_list - 실행시킬 동작을 일시정지 시킬 것인지 여부에 대한 리스트.

**parameter** time\_list - 스테퍼 모터의 동작 할 시간, 또는 동작 할 시간의 스케줄의 리스트. 단위는 초.

**parameter** discovery\_group - 작업을 실행할 로봇의 그룹.

**parameter** run\_option - 스테퍼 모터의 동작 옵션. "continue"이면 한 가지 속도로 계속 돌아가는 모드. "step"이면 한 가지 속도로 한 가지 스텝을 돌아감. "schedule"이면 정해진 순서의 속도로 정해진 순서의 스텝을 돌아 감.

**parameter** speed\_option - 속도 단위를 정함. "RPM"이면 RPM 단위, "SPS"이 면 SPS 단위.

**parameter** step\_option - 스텝 단위를 정함. "CYCLE"이면 CYCLE 단위, "STEP" 이면 STEP 단위.

**parameter** sync - 싱크로나이즈 모드 설정. True이면 싱크가 맞는지 체크하 고, 맞지 않으면 에러. False이면 체크하지 않음.

**parameter** time\_option - 시간 모드를 정함. None이면 speed\_list와 step\_list 로 작동함. "speed"이면 time\_list와 speed\_list로 작동함. "step"이면 time\_list와 step\_list로 작동함.

**parameter** wait - 기다리는 시간을 정함. 단위는 초. "step" 또는 "schedule" 이면 run\_option에 따라서 자동으로 작동하는 시간이 가장 오래 걸리는 큐브를 기다림. 특정 큐브의 스텝 또는 스케줄이 pause이면 그 스텝 또는 스케줄은 무시.

로봇의 스테퍼 모터를 작동시킨다. step\*speed가 양수이면 시계방향으로 돌아 가고, 음수이면 시계 반대방향으로 돌아간다. speed가 0이면 슬립 모드가 되 며, 슬립 모드일 때 step\_option이 "CYCLE"이면 step초만큼 슬립하고, "STEP"이 면 step밀리초만큼 슬립한다. time\_option이 "step"이면 step이 0이거나 "stop" 또는 "sleep"일 때 슬립 모드가 된다.

**function** stop\_motor(self, cube\_ID\_list="all", discovery\_group=None)

**parameter** cube\_ID\_list - 정지시킬 로봇 번호의 리스트. "all"이면 모든 로봇 을 정지시킴.

**parameter** discovery\_group - 작업을 실행할 로봇의 그룹.

로봇의 스테퍼 모터를 정지시킨다.

**function**  set\_motor\_schedule(cube\_ID\_list, speed\_list, step\_list, pause\_list=True, time\_list= None, discovery\_group=None, speed\_option="RPM", step\_option="CYCLE", sync=False, time\_ option=None, wait=0)

**parameter** cube\_ID\_list - 스테퍼 모터의 스케줄을 정할 로봇 번호의 리스트. "all" 이면 모든 로봇을 작동.

**parameter** speed\_list - 스테퍼 모터의 속도 스케줄의 리스트. "stop" 또는 "sleep"이면 속도는 0. 단위가 RPM이면 범위는 -3 ~ -30, 0, 3 ~ 30. 단위가 SPS이면 범위는 -1000 ~ -100, 0, 100 ~ 1000.

**parameter** step\_list - 스테퍼 모터의 이동 스텝 스케줄의 리스트. 단위가 CYCLE이면 범위는 -32.7675 ~ 32.7675. 단위가 STEP이면 범 위는 -65535 ~ 65535.

**parameter** pause\_list - 실행시킬 동작을 일시정지 시킬 것인지 여부에 대한 리스트.

**parameter** time\_list - 스테퍼 모터의 동작 할 시간, 또는 동작 할 시간의 스케줄의 리스트. 단위는 초.

**parameter** discovery\_group - 작업을 실행할 로봇의 그룹.

**parameter** speed\_option - 속도 단위를 정함. "RPM"이면 RPM 단위, "SPS"이 면 SPS 단위.

**parameter** step\_option - 스텝 단위를 정함. "CYCLE"이면 CYCLE 단위, "STEP" 이면 STEP 단위.

**parameter** sync - 싱크로나이즈 모드 설정. True이면 싱크가 맞는지 체크하 고, 맞지 않으면 에러. False이면 체크하지 않음.

**parameter** time\_option - 시간 모드를 정함. None이면 speed\_list와 step\_list 로 작동함. "speed"이면 time\_list와 speed\_list로 작동함. "step"이면 time\_list와 step\_list로 작동함.

**parameter** wait - 기다리는 시간을 정함. 단위는 초. "step" 또는 "schedule" 이면 run\_option에 따라서 자동으로 작동하는 시간이 가장 오래 걸리는 큐브를 기다림. 특정 큐브의 스텝 또는 스케줄이 pause이면 그 스텝 또는 스케줄은 무시.

로봇의 스테퍼 모터의 스케줄을 설정한다. step\*speed가 양수이면 시계방향으로 돌아 가고, 음수이면 시계 반대방향으로 돌아간다. speed가 0이면 슬립 모드가 되 며, 슬립 모드일 때 step\_option이 "CYCLE"이면 step초만큼 슬립하고, "STEP"이 면 step밀리초만큼 슬립한다. time\_option이 "step"이면 step이 0이거나 "stop" 또는 "sleep"일 때 슬립 모드가 된다.

**function**  play\_motor\_schedule(cube\_ID\_list="all", repeat\_list=[[1]], start\_point\_list=[[None]], sto p\_point\_list=[[None]], start\_and\_stop\_list=[[None]], pause\_list=False, discovery\_group =None, sync=False, wait=0)

**parameter** cube\_ID\_list - 스테퍼 모터의 스케줄을 플레이할 로봇 번호의 리 스트. "all"이면 모든 로봇을 작동.

**parameter** repeat\_list - 반복할 스케줄의 반복 횟수. 입력 값은 양의 정수.

**parameter** start\_point\_list - 실행시킬 스케줄의 시작 인덱스 리스트. 입력 값은 0부터 스케줄 인덱스의 끝까지. 입력이 없으면 [[0]]이 디폴트.

**parameter** stop\_point\_list - 실행시킬 스케줄의 끝 인덱스 리스트. 끝의 인 덱스도 재생함. 입력 값은 0부터 스케줄 인덱스 끝까지. "end"이면 스케줄 끝의 인덱스. 입력이 없으면 [["end"]]가 디폴트.

**parameter** start\_and\_stop\_list - 실행시킬 스케줄의 시작과 끝 인덱스 리스 트. 입력 값은 0부터 스케줄 인덱스 끝까지. 여기에 입력이 있으면 start\_point\_list와 stop\_point\_list는 무시함.

**parameter** pause\_list - 각 큐브의 플레이 리스트를 일시정지 시킬 것인지에 대한 리스트.

**parameter** discovery\_group - 작업을 실행할 로봇의 그룹.

**parameter** sync - 싱크로나이즈 모드 설정. True이면 싱크가 맞는지 체크하 고, 맞지 않으면 에러. False이면 체크하지 않음.

**parameter** wait - 기다리는 시간을 정함. 단위는 초. "step" 또는 "schedule" 이면 자동으로 작동하는 시간이 가장 오래 걸리는 큐브를 기다림. 특정 큐브의 플레이 스케줄이 pause이면 그 플레이 스케줄은 무시.

로봇의 설정된 스케줄을 실행한다. start\_point\_list와 stop\_point\_list에 아무것도 입력하지 않으면 설정된 스케줄의 처음부터 끝까지 실행한다. 설정된 스케줄 이 없으면 에러가 발생한다.

Examples

run\_motor()

1. run\_motor()

전체 로봇을 30RPM으로 작동시킨다.

1. run\_motor([1, 2], [30, -20])

1번, 2번 로봇을 각각 30RPM, -20RPM으로 작동시킨다.