

사용 설명서

Nano Stage
(ST01, PDV01)
25.5.14

Sejong Scientific Instruments Inc.



Title of equipment: Nano Stage

Model: ST01, PDV01

제품 개요:

본 제품은 step motor 로 구동하는 motorized stage 와 10nm 분해능의 encoder 를 결합하여 backlash 를 보상한 제품입니다. 드라이버는 microcontroller 와 1/256 microstep 기반 motor driver 를 결합하고 고도의 제어 알고리즘을 적용하여 10 nm 제어를 가능하게 한 제품입니다.

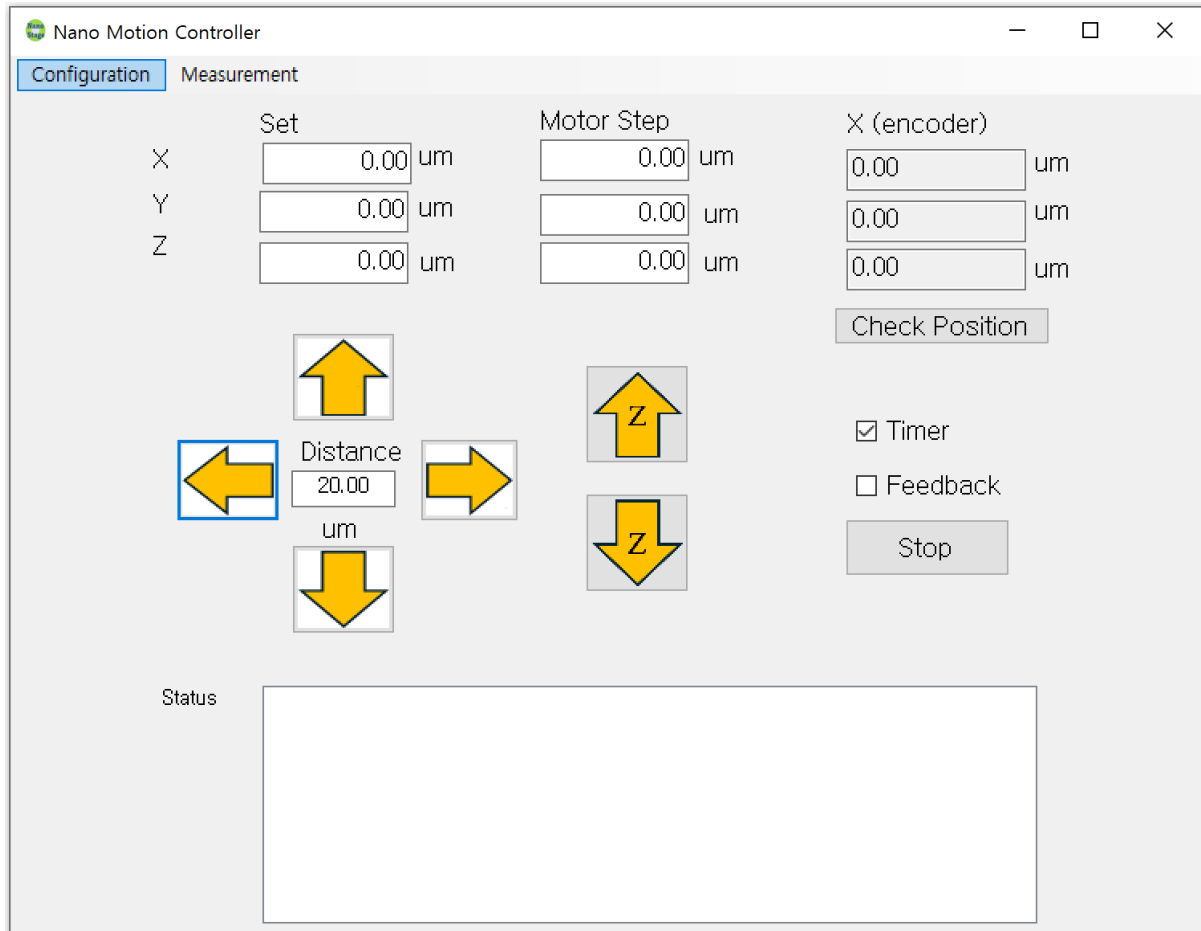
구성품: X, XY, XYZ motorized stage, motor driver, 노트북컴퓨터

하드웨어 개요:

(설치 및 연결) PC 와 motor driver 는 USB 케이블로 연결하고 motor driver 와 stage 는 주어진 dsub 케이블로 연결합니다. 각 축마다 2 개의 dsub 케이블이 있는데, 모터구동과 encoder 를 위해 둘 다 연결되어야 합니다.

소프트웨어 개요:

본 응용 프로그램은 Windows 10, 11 에서 동작합니다.



메인화면

“Set”은 사용자가 원하는 위치 값을 입력하는 창입니다.

“Motor Step”은 모터 드라이버가 결정한 회전 값으로 사용자에게 참고용으로 보여줍니다.

“X(encoder)”는 인코더가 실제 측정된 스테이지의 위치를 나타냅니다.

“Check Position” 버튼을 사용자가 클릭하면 인코더가 현재 측정한 위치 값을 “X(encoder)”에 보여줍니다.

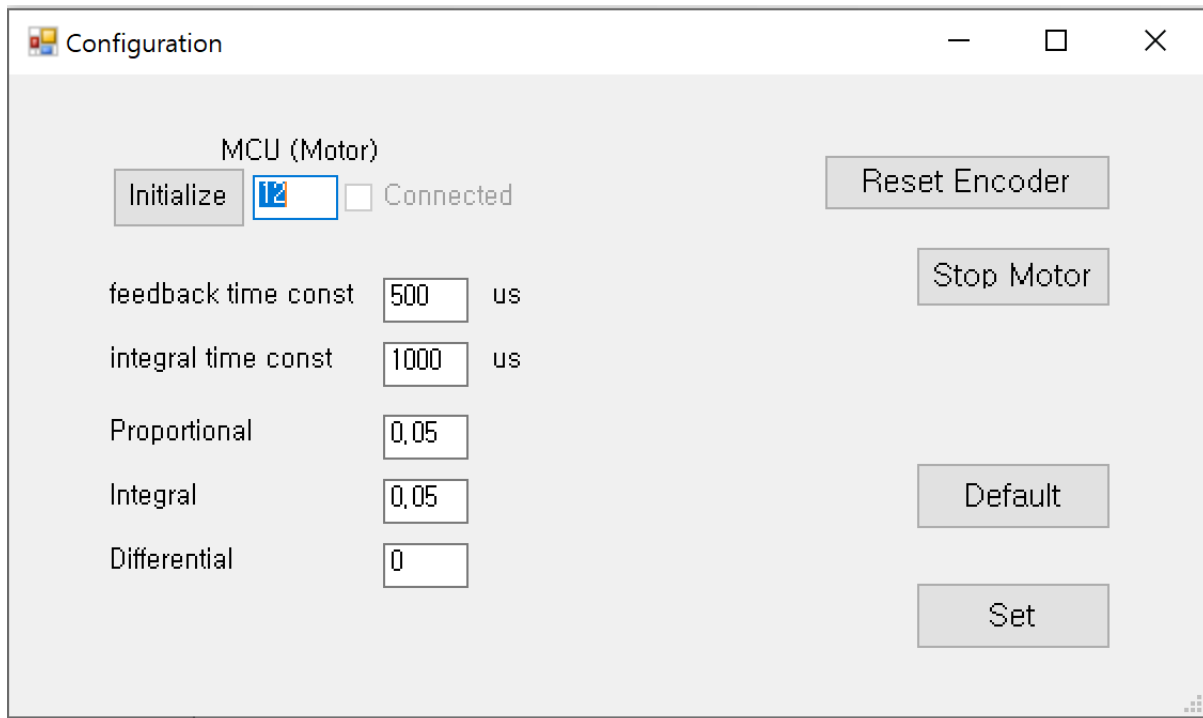
화살표로 표시된 버튼은 사용자가 마우스로 클릭해서 위치를 바꿀 수 있도록 하는 버튼입니다.

“Distance”에 표시된 값만큼 이동합니다. “Distance” 값은 사용자가 원하는 데로 바꿀 수 있습니다.

“Timer” check box 가 체크되어 있을 때는 계속적으로 인코더 값의 변화를 실시간으로 보여주며,

“Feedback” check box 가 체크되어 있을 때는 인코더로 측정된 값이 “Set” 값과 동일해지도록 지속적으로 feedback 제어를 하며, 체크되어 있지 않으면 feedback 제어를 멈춥니다.

“Status” 창에서는 드라이버의 제어 상황을 보여줍니다.



설정 창 화면

설정 메뉴에서는 사용자들이 작성 조건 및 연결된 장치에 대한 전반적인 설정을 수정할 수 있습니다.

MCU 포트 번호:

- 마이크로프로세서가 제대로 연결되면 "연결됨" 체크박스에 표시가 됩니다.
- 연결에 실패한 경우, 마이크로프로세서의 올바른 COM 포트 번호를 확인해야 합니다:

- Windows의 **장치 관리자(Device Manager)**를 엽니다.
- **포트(COM & LPT)** 항목에서 MCU 장치에 할당된 포트 번호를 확인합니다.
- 확인한 포트 번호를 입력란에 입력한 후 **"Initialize"** 버튼을 클릭합니다.
- 연결에 성공하면 "연결됨" 체크박스가 표시됩니다.

- "feedback time const"는 피드백제어 시간을 보여줍니다.
- "integration time const"는 적분 시간을 보여줍니다.

- "Proportional"은 P 제어를 위한 gain 값을 보여줍니다.
- "Integral"은 I 제어를 위한 gain 값을 보여줍니다.
- "Differential"은 D 제어를 위한 gain 값을 보여줍니다.
- "Reset Encoder"는 인코더 값을 다시 설정하는 버튼입니다.
- "Stop Motor"는 모터의 움직임을 강제로 멈추게 하는 버튼입니다.
- "Default"는 PID 제어 파라미터들을 제품 출시에 설정된 원래 값으로 바꾸는 버튼입니다.
- "Set"는 현재 변경한 파라미터 값들로 변경하는 적용하는 버튼입니다.

사용자 응용 프로그램 개발:

사용자 프로그래밍을 위해서 UART 시리얼 통신 표준을 따르고 있으며, 일반적인 프로그램 언어에서 제공되는 표준 통신 함수를 사용하여 text 기반의 아래 명령어들을 송수신 합니다.

윈도우용 pc에서는 우선 장치관리자에서 포트 번호를 확인하여 해당 포트 번호로 포트를 열어야 합니다.

예를 들어 Windows API (Win32) 사용하는 C++에서는 아래와 같은 함수를 사용하면 됩니다.

```
#include <windows.h>
    ○ CreateFile() - 포트 열기
    ○ ReadFile(), WriteFile() - 송수신
    ○ SetCommState(), GetCommState() - 설정
```

Python에서는 아래와 같은 함수를 사용합니다.

```
import serial
ser = serial.Serial('COM4', 9600) # 포트, 속도
ser.write(b'Hello')               # 전송
data = ser.readline()             # 수신
ser.close()

//주요 함수:
serial.Serial()
read(), readline(), write(), in_waiting
```

사용자 프로그래밍을 위한 시리얼 통신 명령어:

- 명령어 "m"은 모터를 해당 위치만큼 구동하는 명령어로 encoder 의 값을 무시하고 이동함
- 명령어 "v"은 encoder 의 값을 읽고 이를 반영하여 stage 를 해당 위치만큼 이동하는 명령어임.
- "m" 또는 "v" 다음에 오는 숫자(0, 1, 또는 2), 공백, 다음 또 다른 숫자가 나와야 합니다. 첫번째 숫자는 축을 나타내며, 두번째 숫자는 목표지점의 절대값입니다.
- "p" : position request, 현재 위치 값을 보내주도록 요청함
- "s" : 이동하는 속도를 변경하는 명령어. "s" 다음에 정수형의 숫자가 오며 숫자가 속도를 나타냄, 64000 가 최대 속도이며 이보다 높은 값을 설정하면 모터가 정상적으로 움직이지 못할 수 있음.
- "cl" : check the limit switch, 현재 stage 가 허용된 범위 내에 있는 지를 확인하는 명령어임. limit switch 가
- "fb 1" : 정지된 상태에서 feedback 제어를 켜는 명령어
- "fb 0" : 정지된 상태에서 feedback 제어를 끄는 명령어

Example:

"m 0 5000" : x 축에 5000 step 무조건 이동

"v 1 5000" : y 축에 5000 step 인코더 값에 맞춰서 이동

"s 64000" : 이동하는 속도를 64000 으로 할 것

유지관리 기술자를 위한 명령어

PID 제어를 위한 파라미터 조정 명령어로 일반 사용자는 바꾸지 않는 것을 권장합니다.

"ft 200": feedback time constant as 200 us

"fg 2000": integral time constant as 2000 us

"fp 10": feedback proportionality setting as 0.1

"fi 50": feedback integration setting as 0.5

"fd 3": feedback differential setting as 0.03

