240105 Yuki Nozawa

descSPIM-Python Installation Instructions

※This guide is intended for use with Windows 11.
※Computer specs require at least 64 GB.

1. Environment creation

1-1. Application Installation

Install the required environment from the following URLs.

1-1-1. Installation of Pyhton

https://www.python.org/

Download the installer and follow the instructions to install Python. Select "Add Python to PATH" during installation and add Python to the PATH environment variable. Currently, this PC is using 3.10.9 | packaged by Anaconda, Inc. | (main, Mar 1 2023, 18:18:15) [MSC v.1916 64 bit (AMD64)]. Different versions and bit numbers may lead to errors.

1-1-2. Installation of Anaconda

https://www.anaconda.com/download

Run the downloaded installer. Follow the instructions provided by the installer. Usually, the default settings are fine." If you select the option "Add Anaconda to my PATH environment variable", you can use Anaconda from the command line (Windows).
We recommend downloading the latest version of Anaconda Distribution. Currently on this PC, Anaconda uses conda 23.3.1.
 Search and open Anaconda prompt from home.

```
jupyter notebook
```

Execute the above command and jupyter notebook should be up and running.

1-1-3. Installation of Micro-Manger

https://download.micro-manager.org/nightly/2.0/Windows/

Install the latest version from the above URL. At this time, if the version date is included in the installation, such as "C:\Program Files\Micro-Manager-2.0.231214", change the name to "C:\Program Files\Micro-Manager-2.0".
 Next, from the Start menu, select "Control Panel" -> "System" -> "Advanced System Settings" -> "Environment Variables" -> "Path" from "System Environment Variables" -> "Edit" -> "New" -> add the path of the directory where Micro-Manager is installed (e.g. C:\Program Files\Micro-Manager-2.0)
When editing is complete, click "OK" on all windows to close them. Then restart the PC.

1-1-4. Installation of ThorCam (application for Thorlabs cameras)

https://www.thorlabs.co.jp/software_pages/ViewSoftwarePage.cfm?Code=ThorCam

Install ThorCam Software for 64-Bit Windows from the above URL. Follow the installation guide. Use version 3.7.0.6.

1-1-5. Installation of Kinesis® software (Thorlabs motor application)

https://www.thorlabs.co.jp/newgrouppage9.cfm?objectgroup_id=10285

Install Kinesis® software from the above URL. Follow the installation guide. Use version 1.14.44.

1-1-6. Installation of Arduino IDE

https://www.arduino.cc/

Install Arduino IDE from the above URL. Follow the installation guide. Use version 2.2.1.

1-2. Environment Setting

1-2-1. Creating a new environment

Launch Anaconda prompt and execute the following commands. (myenv) and install the necessary commands in it.

```
python -m venv myenv
myenv\Scripts\activate
```

Execute the above command,

```
(myenv) (base) C:\Users\username>
```

Confirm that the above command is displayed as shown below.

1-2-2. Install the command

```
pip install jupyter
```

```
pip install pyserial
# Refer to the URL https://github.com/pyserial/pyserial
```

```
pip install pythonnet
# Refer to the URL https://github.com/pythonnet/pythonnet
```

```
pip install pymmcore-plus
pip install pymmcore_widget#
pip install "pymmcore-plus[cli]" # may not be needed
pip install PyQt5
pip install PyQt6 # may not be needed
# Refer to the URL https://pymmcore-plus.github.io/pymmcore-widgets/
```

```
pip install napari-micromanager
mmcore install
# Refer to the URL
https://github.com/pymmcore-plus/napari-micromanager?tab=readme-ov-file
```

```
pip install pycromanager
# See URL https://github.com/micro-manager/pycro-manager
```

```
pip install keyboard
# reference URL https://github.com/boppreh/keyboard
```

Execute the above command while checking the reference URL.

1-3. Initial setup procedure in Micro-Manager

1-3-1. Moving TSICam device adapters

https://micro-manager.org/TSI

Follow the steps above to move the dll file into the micro manager file." C:\Users\username\Desktop\Scientific_Camera_Interfaces\Scientific Camera Interfaces\SDK\Native Toolkit\dlls\Native_64_lib" and the dll files in the "C:\ Program Files\Micro-Manager-2.0".

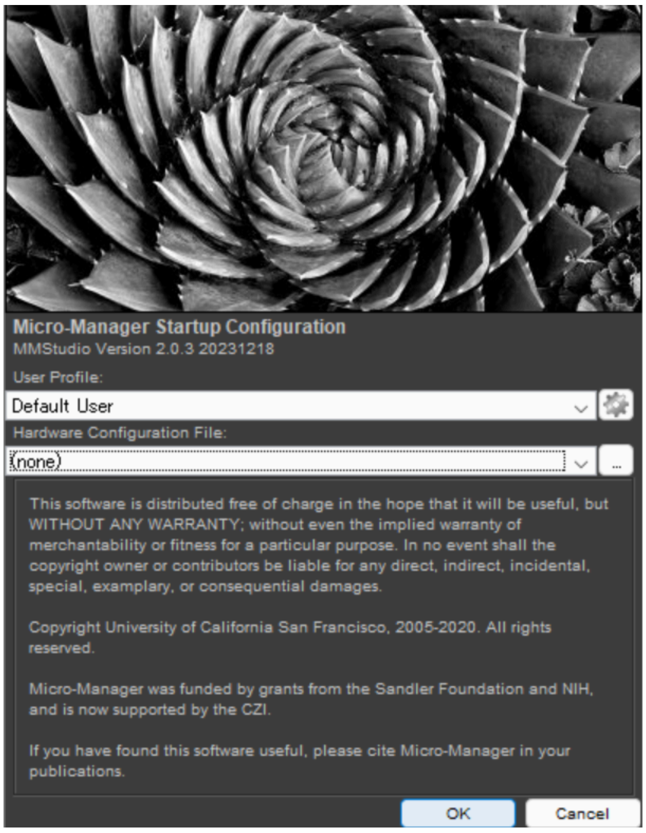
1-3-2. Install ThorlabKinesis device adapter

https://github.com/micro-manager/mmdev-ThorlabsKinesis/releases

Download mmdev-ThorlabsKinesis-20230207.71-x64.zip of v20230207.71 from the above URL. Unzip the zip file and copy "C:\Users\username\Downloads\mmdev-ThorlabsKinesis-20230207.71-x64\mmdev-ThorlabsKinesis-20230207.71-x64\mmgr_dal_ThorlabsKinesis.dll" to "C:\Program Files\Micro-Manager Move "C:\Program Files\Micro-Manager-Micro-Manager-2.0".

1-3-3. Create Configuration File

Launch Micro-Manager from the Start menu.
When Micro-Manager starts up, select "none" in Hardware Configuration File without changing User Profile and click OK to start Micro-Manager.



Micro-Manager Startup Configuration
MMStudio Version 2.0.3 20231218
User Profile: Default User
Hardware Configuration File: (none)

This software is distributed free of charge in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of merchantability or fitness for a particular purpose. In no event shall the copyright owner or contributors be liable for any direct, indirect, incidental, special, examplary, or consequential damages.

Copyright University of California San Francisco, 2005-2020. All rights reserved.

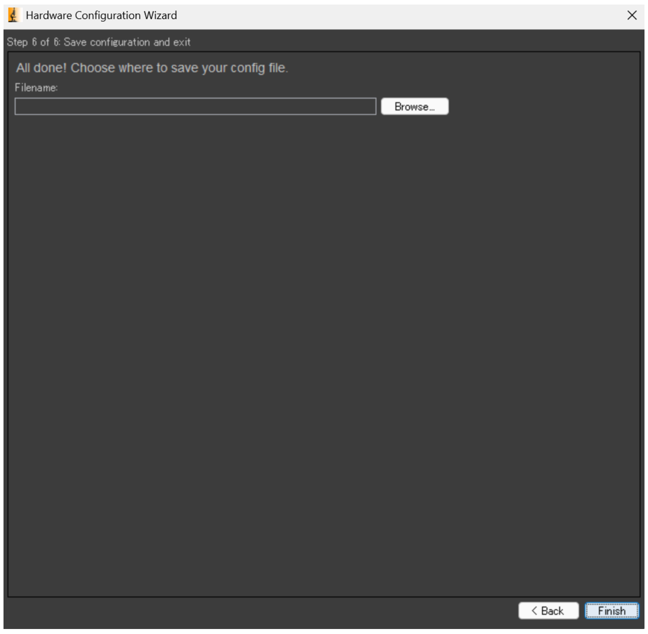
Micro-Manager was funded by grants from the Sandler Foundation and NIH, and is now supported by the CZI.

If you have found this software useful, please cite Micro-Manager in your publications.

After Micro-Manager is launched, select device → Hardware Configurartion Wizard from the menu bar.
At this time, make sure that TSICam and KDC101 motors (2 units) are connected to the PC. Poor connection may cause errors.

(a) Creating TSICam configuration file

Select "Create new configuration" and go to Next, select TSI → TSICam: Thorlabs Scientific Imaging camera from Available Devices:, and click "add". The subsequent windows will remain the same and proceed to the following window.


Hardware Configuration Wizard

Step 6 of 6: Save configuration and exit

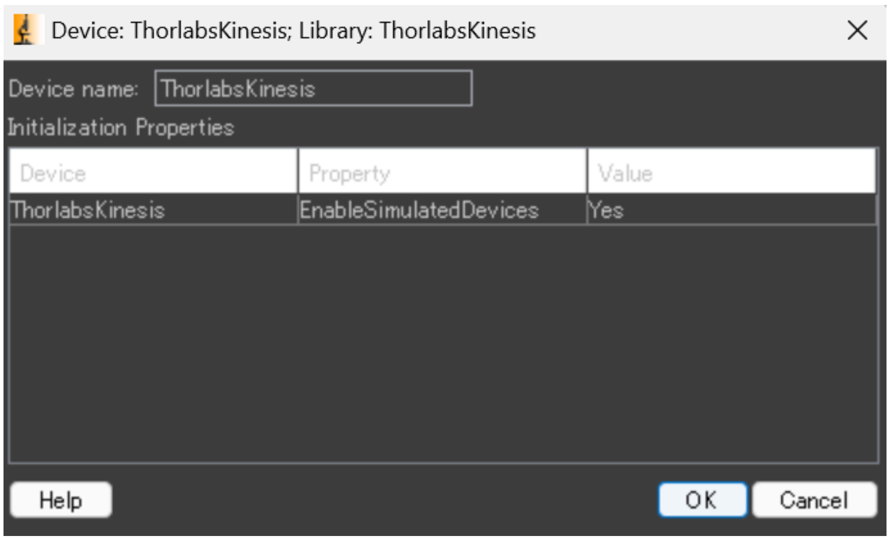
All done! Choose where to save your config file.

Filename:

In this window, specify the file name and file path. For clarity, the file name should include camera. Also, the file path should be on the desktop. (because there are fewer errors related to file calls).
Click Finish to confirm that the configuration file exists in the specified path, then close Micro-Manager.

(b) Creating the ThorlabKinesis configuration file

Launch Micro-Manager again from the Start menu. After launching Micro-Manager, do not change the User Profile, select none in Hardware Configuration File, and click OK to launch Micro-Manager.
Select devices → Hardware Configrartion Wizard from the menu bar, and from Available Devices:, select ThorlabKinesis → ThorlabsKinesis: Thorlabs Kinesis Hub, Click add.
In the next window, the Device Name should remain ThorlabsKinesis.

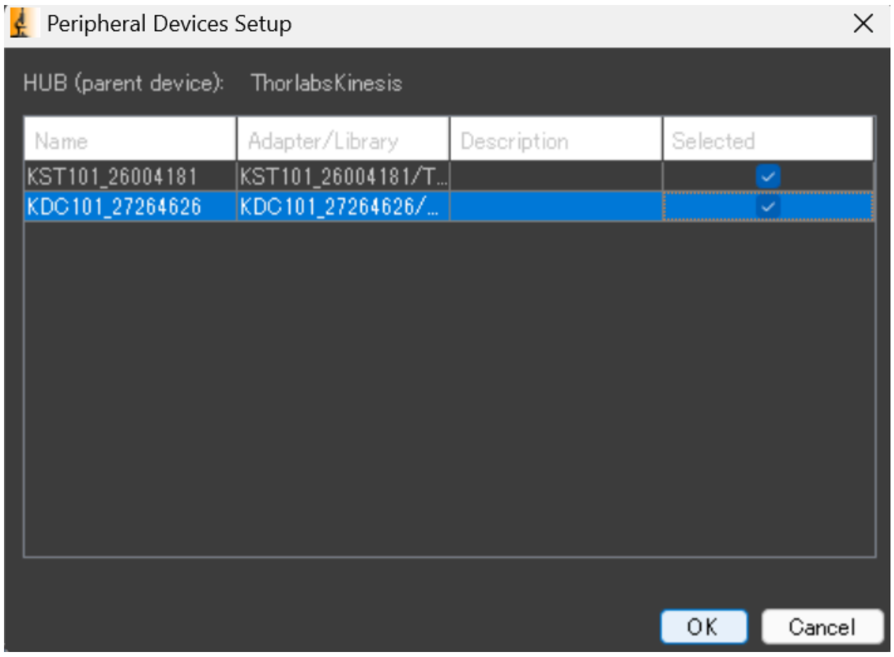

Device: ThorlabsKinesis; Library: ThorlabsKinesis

Device name: ThorlabsKinesis

Initialization Properties

Device	Property	Value
ThorlabsKinesis	EnableSimulatedDevices	Yes

Select OK and proceed.

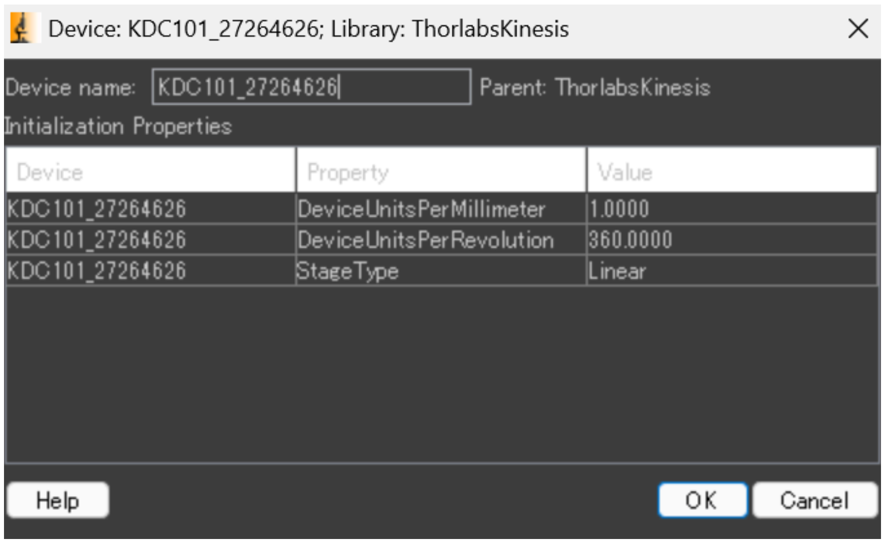

Peripheral Devices Setup

HUB (parent device): ThorlabsKinesis

Name	Adapter/Library	Description	Selected
KST101_26004181	KST101_26004181/T...		✓
KDC101_27264626	KDC101_27264626/...		✓

Confirm that two motors are recognized. After confirming, select all the checkboxes under Selected and click OK.

Check the Device name, which shows the serial number after KDC_101_. If it is the number of the stage on the camera side, set the device name to cameraStage. Similarly, if it is the sample side, set it to sampleStage.

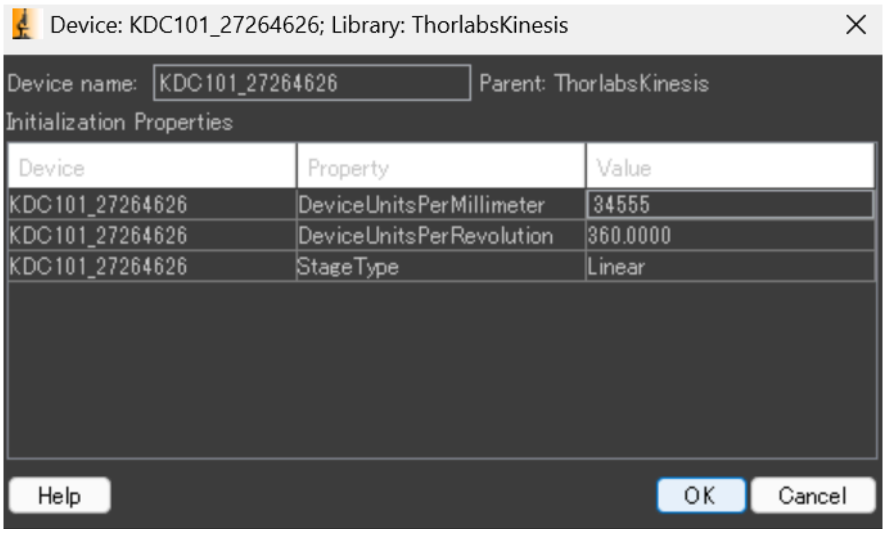

Device: KDC101_27264626; Library: ThorlabsKinesis

Device name: KDC101_27264626 Parent: ThorlabsKinesis

Initialization Properties

Device	Property	Value
KDC101_27264626	DeviceUnitsPerMillimeter	1.0000
KDC101_27264626	DeviceUnitsPerRevolution	360.0000
KDC101_27264626	StageType	Linear

After changing the Device name, change the value of Value in the first line from 1.0000 to 34555.


Device: KDC101_27264626; Library: ThorlabsKinesis

Device name: KDC101_27264626 Parent: ThorlabsKinesis

Initialization Properties

Device	Property	Value
KDC101_27264626	DeviceUnitsPerMillimeter	34555
KDC101_27264626	DeviceUnitsPerRevolution	360.0000
KDC101_27264626	StageType	Linear

※The value given to "Value" is specific to the device unit of each motor model, and this value is used in this descSPIM because a Z825B motor is used. See below for the calculation method.
https://github.com/micro-manager/mmdev-ThorlabsKinesis

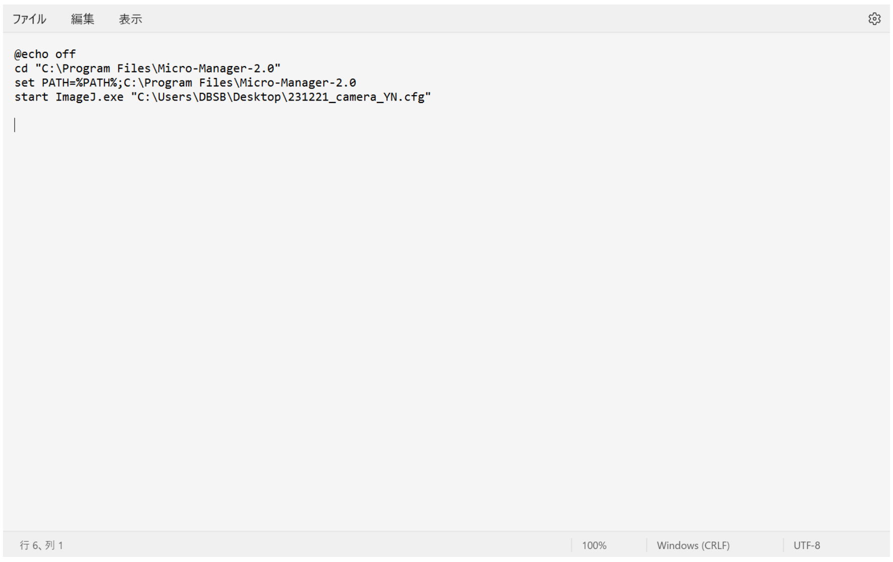
After that, proceed to the same window as before and specify the file name and file path. Click "Finish" to confirm that the configuration file exists in the specified path, and then close Micro-Manager.

1-3-4. Make the Configuration File into a batch file.

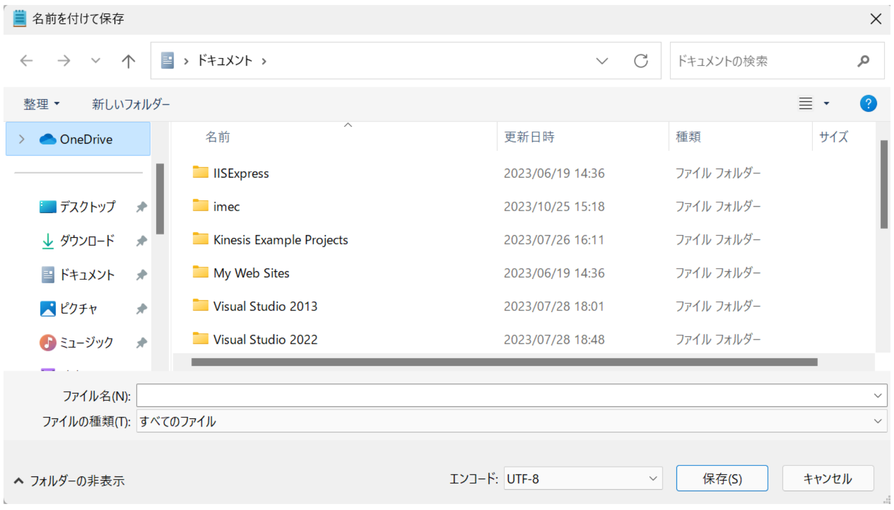
Launch Notepad from the Start menu.

```
@echo off
cd "C:\Program Files\Micro-Manager-2.0"
set PATH=%PATH%;C:\Program Files\Micro-Manager-2.0
start ImageJ.exe "C:\Users\username\Desktop\configration file name of
camera.cfg"
```

Fill in the It will look like the following.



```
@echo off
cd "C:\Program Files\Micro-Manager-2.0"
set PATH=%PATH%;C:\Program Files\Micro-Manager-2.0
start ImageJ.exe "C:\Users\DBSB\Desktop\231221_camera_YN.cfg"
```



名前	更新日時	種類	サイズ
IISExpress	2023/06/19 14:36	ファイル フォルダー	
imec	2023/10/25 15:18	ファイル フォルダー	
Kinesis Example Projects	2023/07/26 16:11	ファイル フォルダー	
My Web Sites	2023/06/19 14:36	ファイル フォルダー	
Visual Studio 2013	2023/07/28 18:01	ファイル フォルダー	
Visual Studio 2022	2023/07/28 18:48	ファイル フォルダー	

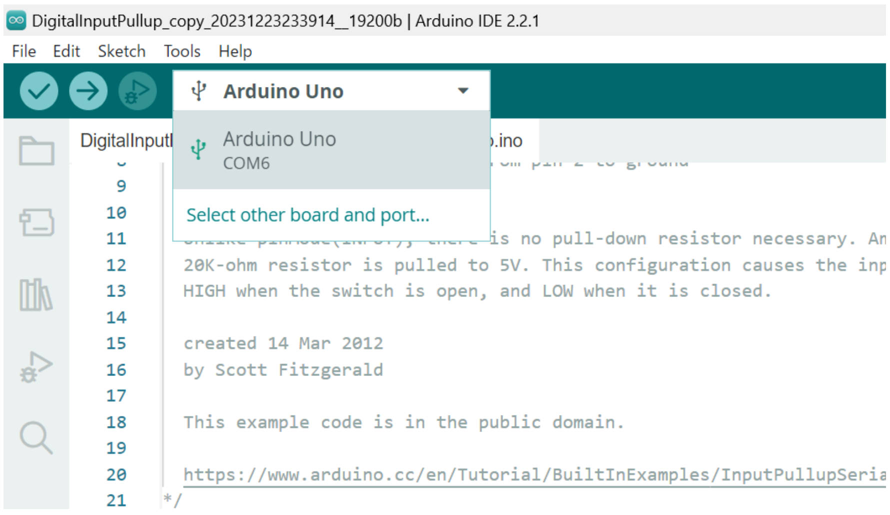
ファイル名(N):
ファイルの種類(T): すべてのファイル
エンコード: UTF-8

Select File→Save As and change the file type to All Files.
Fill in the file name as MMcfg.bat. The file name can be arbitrary, as long as you can distinguish it yourself. The preferred destination is desktop.

※ Depending on your environment settings, batch file may not be necessary.

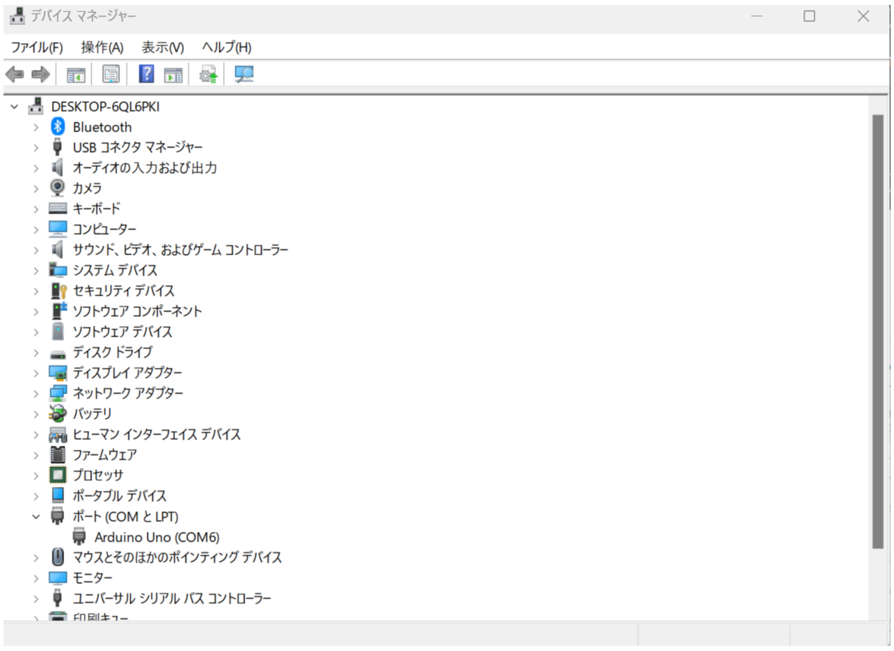
1-4. initial setup of the Arduino device

Once the Arduino device has been built, connect the device's USB to the PC. Launch Arduino IDE 2.2.1 from Home.


DigitalInputPullup_copy_20231223233914__19200b | Arduino IDE 2.2.1

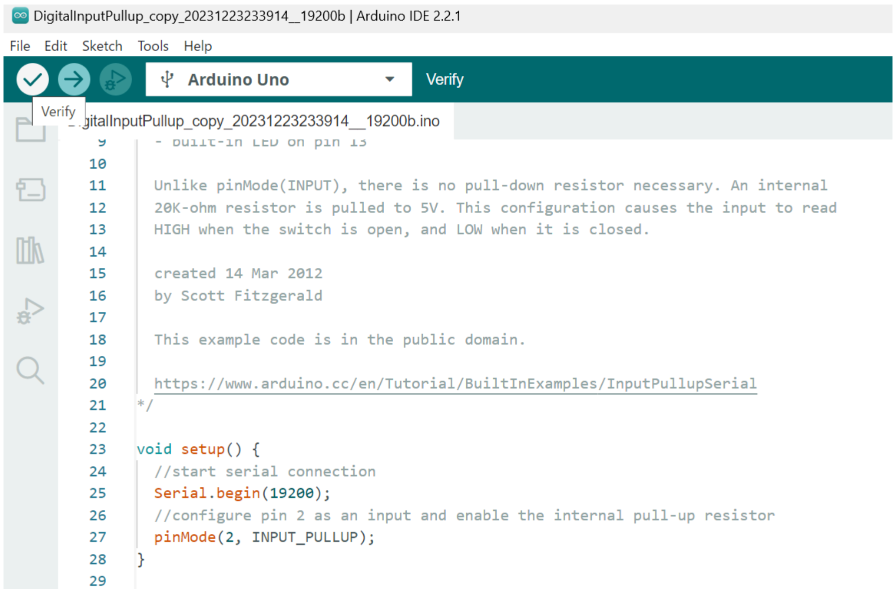
```
is no pull-down resistor necessary. An
20K-ohm resistor is pulled to 5V. This configuration causes the inp
HIGH when the switch is open, and LOW when it is closed.

created 14 Mar 2012
by Scott Fitzgerald

This example code is in the public domain.

https://www.arduino.cc/en/Tutorial/BuiltInExamples/InputPullupSeria
*/
```

After launching, click on Arduino Uno in the upper left corner of the window and check the COM number. You will see that it is connected to COM 6 on this PC.
Next, launch Device Manager from Home.



DESKTOP-6QL6PKI
- Bluetooth
- USB コネクタ マネージャー
- オーディオの入力および出力
- カメラ
- キーボード
- コンピューター
- サウンド、ビデオ、およびゲーム コントローラー
- システム デバイス
- セキュリティ デバイス
- ソフトウェア コンポーネント
- ソフトウェア デバイス
- ディスク ドライブ
- ディスプレイ アダプター
- ネットワーク アダプター
- バッテリ
- ヒューマン インターフェイス デバイス
- ファームウェア
- プロセッサ
- ポータブル デバイス
- ポート (COM と LPT)
  - Arduino Uno (COM6)
- マウスとそのほかのポインティング デバイス
- モニター
- ユニバーサル シリアル バス コントローラー
- 印刷キュー

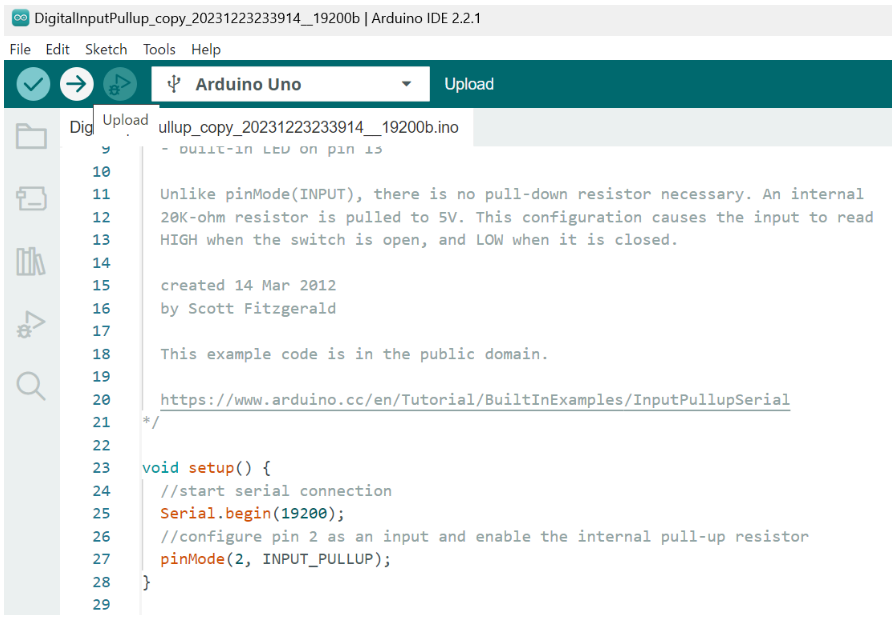
Ports (COM and LPT) → Verify that the Arduino Uno (COM 6) is connected. This method of checking the connection in the device manager is also useful when checking for program errors that originate from a bad USB connection.
Open "C:\Users\DBSB\Downloads\DigitalInputPullup_copy_20231223233914__19200b.ino" (*231226 file name must be changed) from File→Open.


DigitalInputPullup_copy_20231223233914__19200b | Arduino IDE 2.2.1

File Edit Sketch Tools Help
Arduino Uno
Verify
Verify
DigitalInputPullup_copy_20231223233914__19200b.ino

```
  - built-in LED on pin 13

  Unlike pinMode(INPUT), there is no pull-down resistor necessary. An internal
  20K-ohm resistor is pulled to 5V. This configuration causes the input to read
  HIGH when the switch is open, and LOW when it is closed.

  created 14 Mar 2012
  by Scott Fitzgerald

  This example code is in the public domain.

  https://www.arduino.cc/en/Tutorial/BuiltInExamples/InputPullupSerial
*/

void setup() {
  //start serial connection
  Serial.begin(19200);
  //configure pin 2 as an input and enable the internal pull-up resistor
  pinMode(2, INPUT_PULLUP);
}

```

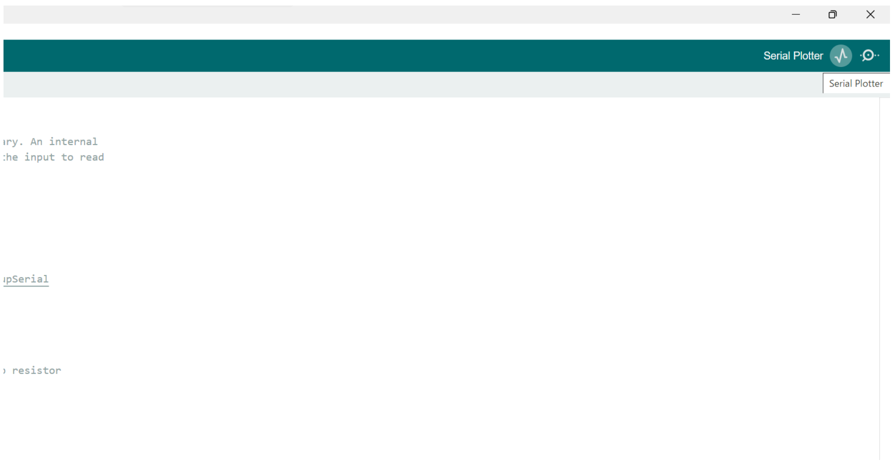
Click ☑ (Verify) to debug the code. Verify that no errors occur during execution.


DigitalInputPullup_copy_20231223233914__19200b | Arduino IDE 2.2.1

File Edit Sketch Tools Help
Arduino Uno
Upload
Upload
DigitalInputPullup_copy_20231223233914__19200b.ino

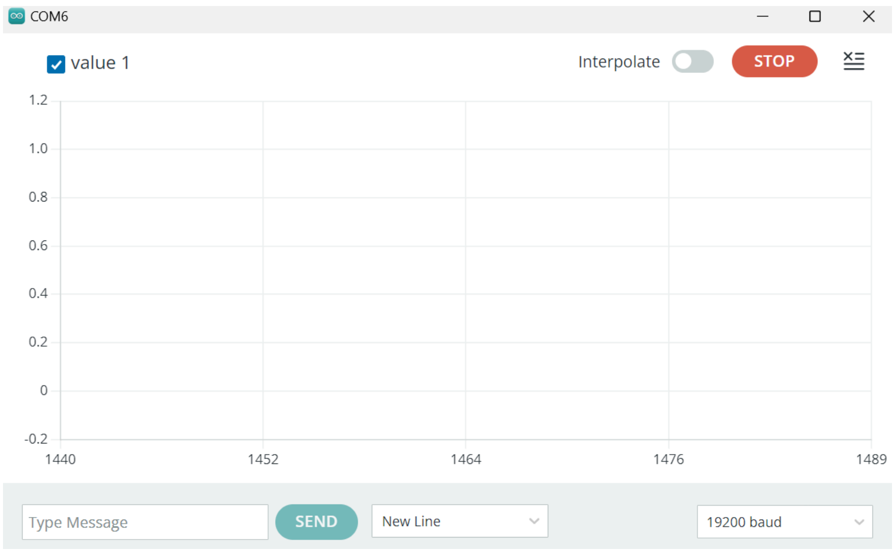
```
  - built-in LED on pin 13

  Unlike pinMode(INPUT), there is no pull-down resistor necessary. An internal
  20K-ohm resistor is pulled to 5V. This configuration causes the input to read
  HIGH when the switch is open, and LOW when it is closed.

  created 14 Mar 2012
  by Scott Fitzgerald

  This example code is in the public domain.

  https://www.arduino.cc/en/Tutorial/BuiltInExamples/InputPullupSerial
*/

void setup() {
  //start serial connection
  Serial.begin(19200);
  //configure pin 2 as an input and enable the internal pull-up resistor
  pinMode(2, INPUT_PULLUP);
}

```

Click → (Upload) to compile and upload DigitalInputPullup_copy_20231223233914__19200b.ino to the Arduino device. This operation makes the above file uploaded to the device semi-permanently.



```
ıry. An internal
:he input to read

ıpSerial

› resistor
```

Next, click on Serial Plotter in the upper right corner of the window to launch the following window.

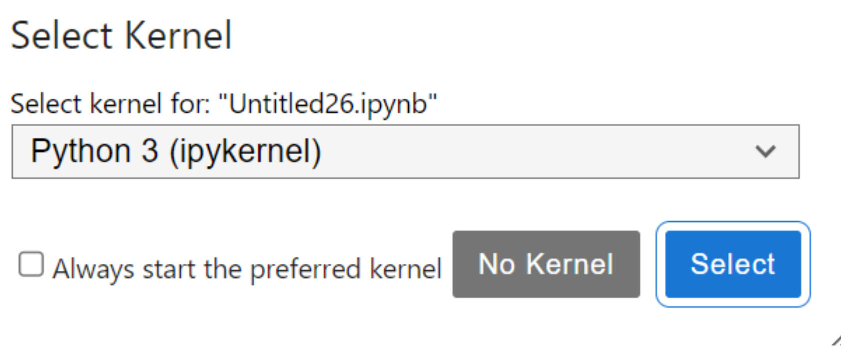


Change the value of baud in the lower right corner of the window to 19200. Confirm that the horizontal axis of the displayed graph (Time, seconds) is advancing with time. Also, confirm that the signal is displayed on the graph by connecting the two cables of the device.

Once confirmed, verify that the Arduino device is also recognized by jupyter notebook, but with the Micro-Manger disconnected.

```
myenv\Scripts\activate
jupyter notebook
```

Launch the above on Anaconda prompt and open a new notebook with File → New → Notebook in the upper right corner of the window.


Select Kernel

Select kernel for: "Untitled26.ipynb"

Python 3 (ipykernel)

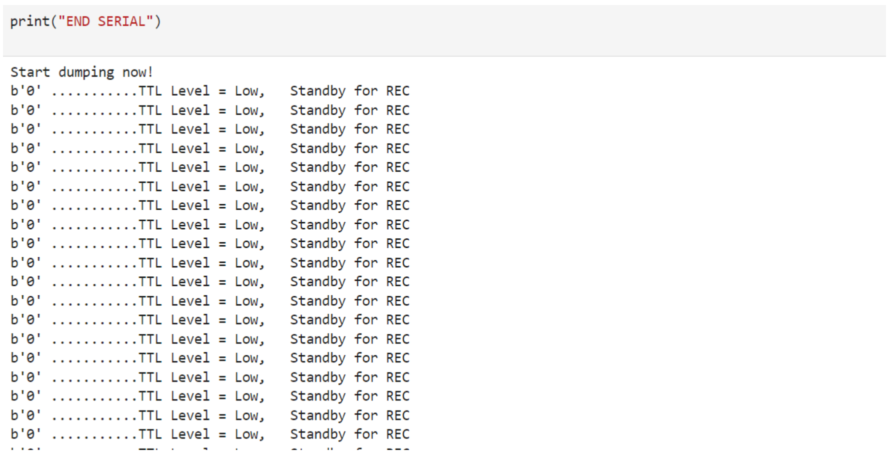
When the window "Select Kernel" appears, select Python 3 (ipykernel), paste the following code and execute it.

You can also upload the descSPIM_Python program file, select + from the menu bar at the beginning of the program, and open a new cell to perform the same check.

```
print("Start dumping now!")

######### initialization for USB/Serial ########

import serial
import time
import threading

COM = "COM6"           # Using DEVICE Manager, Check the COM number &
revise if necessary!
bitRate = 19200        # Serial data transmission speed over USB
(Default 9600 was too slow)

ser = serial.Serial(COM, bitRate, timeout=1)

######### Check the status of Digital Input on Pin 2, Arduino Board
##############

n = 0

def read_serial():
    while True:
        data = ser.read(512)   # Received 512 bytes (One Packet Data?)
        pickup = data[0:1]     # Seems "Bit2" status embedded in this
position...
        if pickup.decode("utf-8") == "0":
            print(pickup, "..........TTL Level = Low,   Standby for
REC")
 if pickup.decode("utf-8") == "1":
```

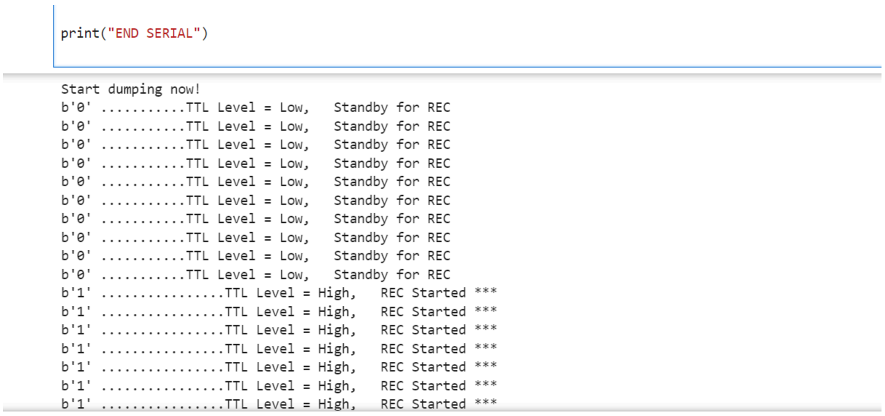
```
            print(pickup, "................TTL Level = High,   REC
Started ***")
        time.sleep(0.1)        # DO NOT REMOVE THIS DELAY !! Somehow
it's important!

serial_thread = threading.Thread(target=read_serial)
serial_thread.daemon = True
serial_thread.start()

input("Press Enter to stop...")

ser.close()

print("END SERIAL")
```

The above code is designed to continuously acquire the TTL signal of the Arduino device, get the camera's standby and recording, and display and confirm the signal.



```
print("END SERIAL")

Start dumping now!
b'0' ..........TTL Level = Low,   Standby for REC
b'0' ..........TTL Level = Low,   Standby for REC
b'0' ..........TTL Level = Low,   Standby for REC
b'0' ..........TTL Level = Low,   Standby for REC
b'0' ..........TTL Level = Low,   Standby for REC
b'0' ..........TTL Level = Low,   Standby for REC
b'0' ..........TTL Level = Low,   Standby for REC
b'0' ..........TTL Level = Low,   Standby for REC
b'0' ..........TTL Level = Low,   Standby for REC
b'0' ..........TTL Level = Low,   Standby for REC
b'0' ..........TTL Level = Low,   Standby for REC
b'0' ..........TTL Level = Low,   Standby for REC
b'0' ..........TTL Level = Low,   Standby for REC
b'0' ..........TTL Level = Low,   Standby for REC
b'0' ..........TTL Level = Low,   Standby for REC
b'0' ..........TTL Level = Low,   Standby for REC
b'0' ..........TTL Level = Low,   Standby for REC
b'0' ..........TTL Level = Low,   Standby for REC
b'0' ..........TTL Level = Low,   Standby for REC
```

A series of messages will appear as shown above.
Leave the jupyter notebook tab as it is and start ThorCam. Specify an appropriate file and confirm that the message changes from Standby for REC to REC Started as shown below when you start recording.



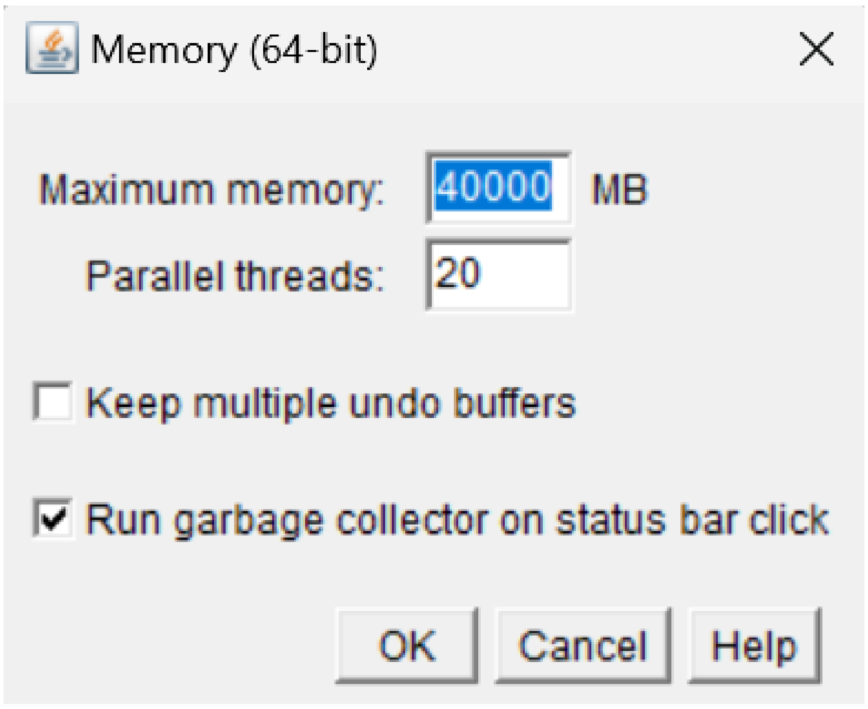
```
print("END SERIAL")
```

```
Start dumping now!
b'0' ..........TTL Level = Low,   Standby for REC
b'0' ..........TTL Level = Low,   Standby for REC
b'0' ..........TTL Level = Low,   Standby for REC
b'0' ..........TTL Level = Low,   Standby for REC
b'0' ..........TTL Level = Low,   Standby for REC
b'0' ..........TTL Level = Low,   Standby for REC
b'0' ..........TTL Level = Low,   Standby for REC
b'0' ..........TTL Level = Low,   Standby for REC
b'0' ..........TTL Level = Low,   Standby for REC
b'0' ..........TTL Level = Low,   Standby for REC
b'1' ...............TTL Level = High,   REC Started ***
b'1' ...............TTL Level = High,   REC Started ***
b'1' ...............TTL Level = High,   REC Started ***
b'1' ...............TTL Level = High,   REC Started ***
b'1' ...............TTL Level = High,   REC Started ***
b'1' ...............TTL Level = High,   REC Started ***
b'1' ...............TTL Level = High,   REC Started ***
```

1-5. Changing RAM (C:) settings

Since Micro-Manager uses the C drive as a temporary storage location for captured images, it is essential to change the setting when performing high stack imaging. The procedure is described below.

Launch Micro-Mnager, then launch Edit → Options → Memory & Threads from the imageJ menu bar.



Memory (64-bit)

Maximum memory: 40000 MB

Parallel threads: 20

☐ Keep multiple undo buffers

☑ Run garbage collector on status bar click

In the above window, change the number in the Maximum memory box. The number to be added here depends on the number of stacks required for the shoot. Specifically, 1000 stacks require about 25 GB of C drive memory. Therefore, for this installation, it is recommended that Maximum memory be set to 40000 MB to 50000 MB or more.

When shooting more than 500 to 1000 stacks for the first time, it is recommended to open the Task Manager from the Home menu, go to Performance -> Memory, and observe the memory usage while shooting to make sure that RAM (C:) does not exceed its capacity.

See: https://micro-manager.org/Micro-Manager_Configuration_Guide

2. Executing Python Code

2-1. Regarding pre-confirmation.ipynb

Launch the myenv environment again from the Anaconda prompt.

```
jupyter notebook
```

Execute the above and launch jupyter notebook.
Upload and open the file "pre-confirmation.ipynb". This file is used to verify that the environment you set up in step 1 is functioning properly on jupyter notebook.
Check the cells from the top to the bottom and confirm that there are no differences from the information shown below.

```
import sys

print("Python Version:")
print(sys.version)
```

```
Python Version:
3.10.9 | packaged by Anaconda, Inc. | (main, Mar  1 2023, 18:18:15) [MSC v.1916 64 bit (AMD64)]
```

```
import platform

print(platform.architecture())
```

```
('64bit', 'WindowsPE')
```

```
import subprocess

result = subprocess.run(['conda', '--version'], stdout=subprocess.PIPE)

print(result.stdout.decode('utf-8'))
```

```
conda 23.3.1
```

```
import os
```

```
print("Current Directory:", os.getcwd())
```

```
Current Directory: C:\Users\username
```

```
import sys
print(sys.path)
```

```
['C:\\Users\\username', 'C:\\ProgramData\\anaconda3\\python310.zip',
'C:\\ProgramData\\anaconda3\\DLLs', 'C:\\ProgramData\\anaconda3\\lib',
'C:\\ProgramData\\anaconda3', 'C:\\Users\\username\\myenv', '',
'C:\\Users\\username\\myenv\\lib\\site-packages',
'C:\\Users\\username\\myenv\\lib\\site-packages\\win32',
'C:\\Users\\username\\myenv\\lib\\site-packages\\win32\\lib',
'C:\\Users\\username\\myenv\\lib\\site-packages\\Pythonwin']
```

```
import sys
sys.version
```

```
'3.10.9 | packaged by Anaconda, Inc. | (main, Mar  1 2023, 18:18:15) [MSC v.1916 64 bit (AMD64)]'
```

```
import pkg_resources

pymmcore_plus_version =
pkg_resources.get_distribution("pymmcore_plus").version
print("pymmcore_plus version:", pymmcore_plus_version)
```

```
pymmcore_plus version: 0.8.4
```

2-2. Regarding 240302_descSPIM_Python_versionA.ipynb

This file contains the minimum program required for imaging, without detailed functions for imaging.

2-3. Regarding 240302_descSPIM_Python_versionB.ipynb

This file is useful when you want to perform repetitive imaging of spatially identical structures, such as in the case of multicolor imaging. By keeping the initial and final position values in advance, it is possible to perform the same distance and velocity imaging at the same position any number of times.
2-4. descSPIM_24.

2-4. Regarding 240302_descSPIM_Python_completion_file.ipynb

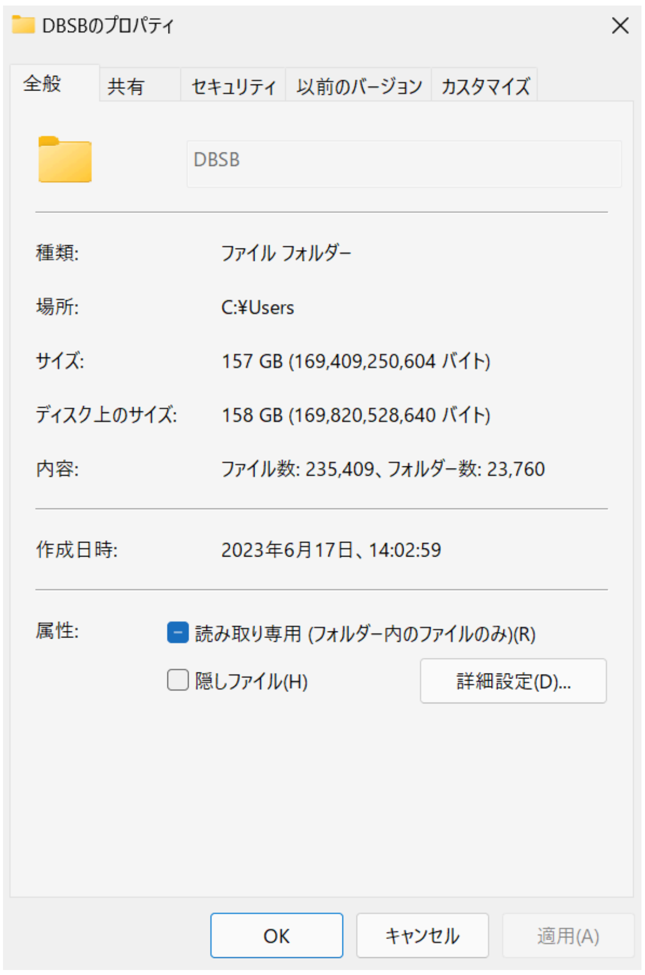
This file is a program that combines all the functions described in 2-3 and 2-4. Initially, we recommend that you actually work on imaging using this file and check what functions each cell has.
After that, you will be able to perform imaging efficiently by using the programs 2-3 and 2-4, which are labor-saving programs according to your purpose.

3. Troubleshooting

The following is a Q&A on possible errors and how to handle them.

3-1. Errors related to Permission

Permission may differ depending on the initial settings of the Windows PC used. This may be one of the reasons why unexpected errors occur even if the environment is configured correctly.
In this case, start up a file with a user name that has windows permission. In the case of this PC, the file is located in "C:\Users\DBSB Right-click on the file and select "Properties" -> "General" to open the following window.



DBSBのプロパティ

DBSB

種類: ファイル フォルダー

場所: C:¥Users

サイズ: 157 GB (169,409,250,604 バイト)

ディスク上のサイズ: 158 GB (169,820,528,640 バイト)

内容: ファイル数: 235,409、フォルダー数: 23,760

作成日時: 2023年6月17日、14:02:59

属性: [✓ 読み取り専用 (フォルダー内のファイルのみ)(R)] 隠しファイル(H)

In the window above, uncheck the Read Only (only files in folders)(R) checkbox and click OK. 3-2.

3-2. Error due to USB connection failure

If you get an error message like "device is not found," the program is not recognizing the device. Make sure that all connected USB adapters are deeply connected. Reconnecting may improve the situation. 3-3.

3-3. failure to load configuration file

If an error message such as "failed to load TSICam" is displayed when Micro-Manager is launched, the configuration file may not be loaded properly. This error indicates that the configuration file created for TSICam (see (3) Initial Setup Procedure in Micro-Manager) failed to load. This may be improved by editing environment variables. Follow the steps below to edit it.

Open "Control Panel" from the Start menu.
→Select "System and Security".
→Select "System".
→Select "Advanced System Settings" from the menu on the left.
→Click the "Environment Variables" button.
→Select "Path" from "System Environment Variables" and click "Edit".
→Click "New" and add the path to the directory where Micro-Manager is installed. Example: C:/Program Files/Micro-Manager-2.0
→When editing is complete, click "OK" to close.
→Click "OK" on all windows to close them in order to save the changes.

3-4. backslash error

When a path is copied and pasted, it is displayed as a backslash like "C:\Program Files\Micro-Manager-2.0". In many cases, this is not a problem, but correcting it to "C:/Program Files/Micro-Manager-2.0" may solve the error. 3-5.

3-5. insufficient memory in Jupyter notebook

Kernel may cause an error due to insufficient memory. Replace the user name with your Windows user name.

```
cd C:\Users\<ユーザー名>\.jupyter
```

Open the configuration file jupyter_notebook_config.py in a text editor.
Start Notepad and open the configuration file using the following command.

```
notepad jupyter_notebook_config.py
```

This allows Notepad to open the configuration file.
Once the configuration file is generated, edit the configuration file using a text editor.
Execute the following command.

```
notepad C:\Users\<ユーザー名>\.jupyter\jupyter_notebook_config.py
```

The location of the configuration file should be adjusted according to the actual installation location.
Edit the configuration file and set the memory usage limit.
For example, add c.NotebookApp.max_mem_rate and set the value as follows.

```
c.NotebookApp.max_mem_rate = 0.9
```

This setting allows the notebook to use up to 90% of the available memory.
Restart Jupyter Notebook by executing the following command.

```
jupyter notebook
```

Free memory by stopping other processes.

To free up memory by stopping other processes, it is necessary to use a tool such as Task Manager or System Monitor to check the running processes and terminate any unnecessary processes.
Click on the icon at the far right of the taskbar (usually the "^" up arrow) and select "Task Manager" from the task tray. (Or press Ctrl + Shift + Esc at the same time.) The Task Manager window will appear, and select the "Processes" tab. In the Process tab, look for processes that are using a lot of memory (or CPU). Those processes are likely to be consuming system memory.
Select the unneeded process and click the "End Process" button at the bottom of the screen to terminate the process. However, do not change them so that they do not affect the system or important applications.
Insufficient memory resources may also be a problem.
Insufficient memory can lead to a kernel crash. Memory usage should be checked in detail and additional memory should be added if necessary.
Check the Jupyter Notebook log for error messages and details of the problem. Logs are usually stored in the ~/.jupyter directory. Checking the error messages will help identify the problem.
Notebook cells are expected to be executed in order from the front, so rewinding and executing may cause errors.

How to check the Jupyter Notebook log

By checking the logs, it is easier to identify error factors.
Identify the location of the log file
Jupyter Notebook log files are usually stored in the ~/.jupyter directory. However, the default log file name is jupyter.log. Since the location and name of the log file can be changed, the following command is used to identify a reliable location.

```
jupyter --paths
```

This command will show the location of Jupyter Notebook configuration files, log files, etc. Identify the location of the log files.

Viewing the log files

Open Anaconda prompt and navigate to the directory where the log files reside.

```
cd ~/.jupyter
```

View log files using a text editor
View log files using a text editor.

```
notepad jupyter.log
```

Checking Error Messages in Log Files

Log files contain execution logs and error messages from Jupyter Notebook. By examining error messages and problem details, you may be able to identify the root cause of the issue.

3-6. Error Due to Insufficient RAM

If you are using a PC with less than 64 GB of capacity, or if you already have other applications running on a PC with more than 64 GB, you may encounter issues related to insufficient RAM (C:). When using a computer with insufficient RAM, the kernel may crash, preventing the execution of cells. Follow the steps below to check and optimize your RAM:

However, it is recommended to use a system with more than 64 GB of capacity, especially when dealing with large stacks.

Checking RAM:

Open Task Manager.
Once Task Manager is open, click on the "Performance" tab at the top.
Click on the "Memory" section to display RAM information. Here, you can check the total installed RAM capacity and the amount of RAM currently in use.

Optimizing RAM:

Optimizing RAM is achieved by terminating unnecessary processes and applications. Below are some possible optimization methods:

Terminating unnecessary applications:

Open Task Manager and select unnecessary applications from the task list.
Click the "End Task" button to terminate them.

Managing applications that automatically start at boot:

Click on the "Startup" tab in Task Manager.
Disable applications that start automatically at boot to reduce unnecessary processes during system startup.

Use the Disk Cleanup tool to delete unnecessary files and temporary files, freeing up disk space.