STMicroelectronics SensorTile Tutorial: Displacement Estimation by Inertial Sensing

Table of Contents

1. INTRODUCTION TO THIS TUTORIAL **3**
List of Required Equipment and Materials 3
Prerequisite Tutorials 3
2. DISPLACEMENT ESTIMATION SYSTEM SETUP **4**
2.1 SensorTile Displacement Estimation System Installation 4
2.2 Python Animation System Installation 5
Mac OS *6*
Windows *11*
3. DISPLACEMENT ESTIMATION SYSTEM INTERPRETATION **22**
3.1 Displacement Estimation: Trapezoidal integral 22
3.2 System Sampling Frequency and Sensor Handler Setup 22
3.3 Filters 24
3.4 Velocity and displacement by Integration 25
3.5 Data Streaming 25

1. Introduction to This Tutorial

We have learned inertial motion sensing, finite state machine based posture recognition, and digital signal processing in the previous tutorials. In this tutorial, we will further update DataLog application and estimate the SensorTile x-axis displacement using SensorTile inertial accelerometer. In addition, we will have a Python animation system to visualize the acceleration and displacement in real time.

The Tutorial steps provide:

1. An introduction to installation of Displacement Estimation project and Python animation system.
   - Displacement estimation project installation
   - Python animation system installation
2. An introduction to Displacement Estimation by Inertial Sensor system.
   - Acceleration processing
   - Displacement processing

For more information regarding the SensorTile board, please open the following link.
www.st.com/sensortile

List of Required Equipment and Materials

1) 1x STMicroelectronics SensorTile kit.
2) 1x STMicroelectronics Nucleo Board.
3) 1x Personal Computer with two USB type-A inputs OR you must have a powered USB hub.
4) 1x USB 2.0 A-Male to Micro-B Cable (micro USB cable).
5) 1x USB 2.0 A-Male to Mini-B Cable (mini USB cable).
6) Network access to the Internet.
7) 1x Small piece of foam pad

Prerequisite Tutorials

It is recommended that users have completed and are familiar with the contents of the following tutorials before proceeding.

1) Tutorials up to tutorial 4.

Your instructor will ensure you have the required background regarding motion sensing, digital signal processing, and basic physics of acceleration, velocity, and displacement.

2. Displacement Estimation System Setup

In the previous tutorials, we explored the DataLog application and AudioLoop application in the SensorTile starter firmware package. In this project, we further update DagtaLop application to estimate displacement using acceleration in DataLog application. You can visualize the x-axis acceleration and displacement by Python animation system through USB data streaming.

2.1 SensorTile Displacement Estimation System Installation

1. Download the project from google drive (https://drive.google.com/open?id=1msaomIBQ2QJ7QXi3mmlH6nnCPaHCGwKs) and decompress it to your working directory.

2. Open the IDE as instructed in the Tutorial 1***.*** Select the same workspace as in Tutorial 1.

3. Once the IDE is open, first remove your current project in System WorkBench and import the DataLog project from corresponding directories as instructed in the document labeled ***STMicroelectronics SensorTile Tutorial: Introduction to STMicroelectronics Development Environment and DataLog Project Example***. Uncheck AudioLoop and BLE projects (first two projects) to properly import the DataLog project. See Figure 1.

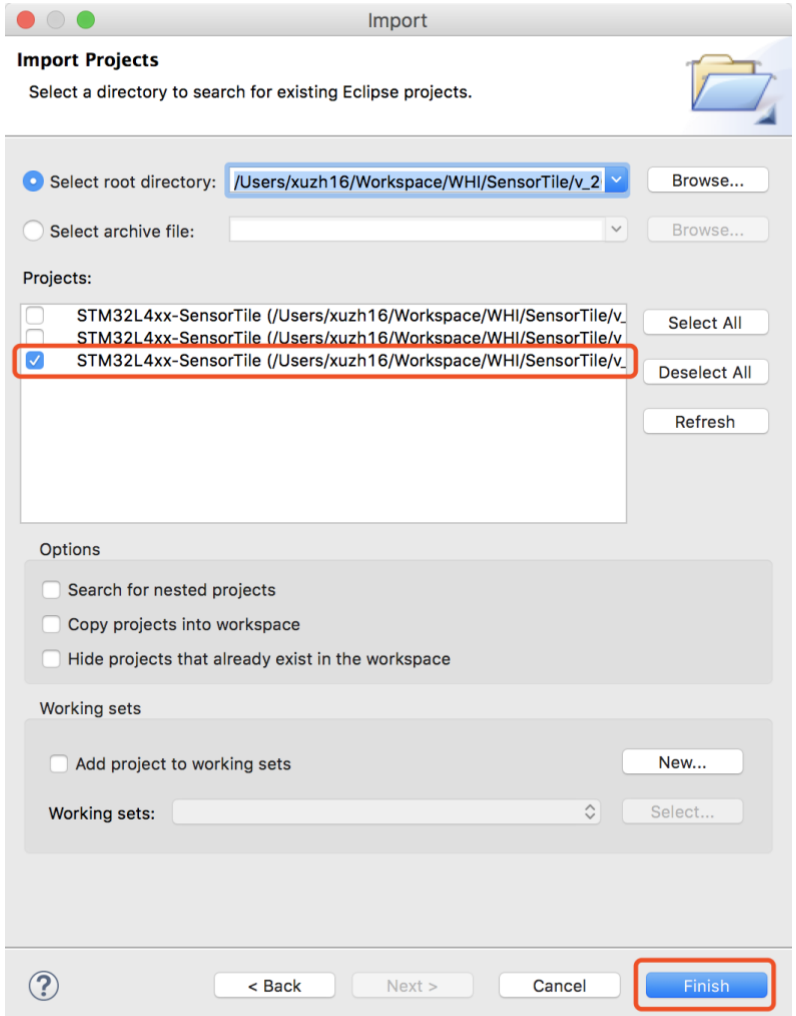

Figure 1: Import DataLog project from modified start firmware package.

4. Once you successfully import the DataLog project, you can build the project. Connect the SensorTile and Neuclo Board with your computer. You can then either run the project in ***Debug*** mode or utilize *st-flash tool* on Mac or *ST-LINK utilities tool* on Windows introduced in Tutorial 1 to flash the latest firmware to SensorTile.

2.2 Python Animation System Installation

Python is a programming language that lets you work quickly and integrate systems more effectively. Python is also the most popular programming platform for machine learning. For more information, please refer to Python official website ([https://www.python.org/)](https://www.python.org/).

In this section, we will introduce Python installation, Python package installation, and Python animation system setup for both Mac OS and Windows.

Download the Python animation system from google drive (https://drive.google.com/file/d/1KiWOmCrowU-k6nYQ1Sw_lZko6B0xpJF4/view?usp=sharing) and unzip it to your working directory. You should find that the animation system is composed of ***SensorTile_Animation_args.py*** and ***SensorTile_Serial.py.***

You can follow the instructions for Mac OS or Windows according to your computer operating system. If you have any concerns about the installation, please ask your course mentors or your teaching assistant, or instructor for assistance.

Mac OS Installation and Operation

2.2.1 Python Installation

You should have Python installed by default in your Mac. Issue command "$ *python*" in a Terminal session to check if Python is installed. This SensorTile Animation system is compatible with both Python 2 and Python 3. You only need one version of Python to properly use the system.

If you have not installed Python in your Mac, use the link (https://www.python.org/downloads/) to download Python 3.7.2 and install accordingly.

2.2.2 Run Python

Open terminal and issue command "$ ***python***". If you have multiple Python version installed, make sure you run the correct command to use the corresponding Python. For example, you could issue "$ ***python***" to start Python 2 and issue "$ ***python3***" to start Python 3. See Figure 2.

```
Xus-MacBook-Pro:SensorTile_Inertial_Sensing_Tutorial xuzh16$ python
Python 2.7.14 |Anaconda custom (64-bit)| (default, Oct  5 2017, 02:28:52)
[GCC 4.2.1 Compatible Clang 4.0.1 (tags/RELEASE_401/final)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>>
Xus-MacBook-Pro:SensorTile_Inertial_Sensing_Tutorial xuzh16$ python3
Python 3.7.2 (default, Jan 13 2019, 12:50:15)
[Clang 10.0.0 (clang-1000.11.45.5)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Figure 2: Python 2 and Python 3. Either version is fine.

You can use command “***exit()***” to exit Python.

2.2.3 Install Required Python Packages

You can find and install software developed and shared by the Python community through PyPl (pip), where PyPl is embedded with Python. You need to install Python packages to operate the SensorTile Python Animation system.

Upgrade pip

In terminal, first issue command “$ ***python -m pip install --upgrade pip***”. See Figure 3.

```
Xus-MacBook-Pro:SensorTile_Inertial_Sensing_Tutorial xuzh16$ python -m pip insta
ll --upgrade pip
DEPRECATION: Python 2.7 will reach the end of its life on January 1st, 2020. Ple
ase upgrade your Python as Python 2.7 won't be maintained after that date. A fut
ure version of pip will drop support for Python 2.7.
Requirement already up-to-date: pip in /anaconda2/lib/python2.7/site-packages (1
9.0.3)
```

Figure 3: Upgrade pip

Install PySerial

In terminal, issue command “$ **sudo *pip install pyserial==3.4***” to install PySerial. See Figure 4.

```
Xus-MacBook-Pro:SensorTile_Inertial_Sensing_Tutorial xuzh16$ pip install pyseria
l==3.4
```

Figure 4: Install PySerial

Install matplotlib

In terminal, issue command “$ **sudo *pip install matplotlib==3.0.2***” to install matplotlib. See Figure 5.

```
Xus-MacBook-Pro:SensorTile_Inertial_Sensing_Tutorial xuzh16$ pip install matplot
lib==3.0.2
```

Figure 5: Install matplotlib

Verify packages are installed

In Python, issue command according to Figure 6.

```
[Xus-MacBook-Pro:SensorTile_Inertial_Sensing_Tutorial xuzh16$ python3
 Python 3.7.2 (default, Jan 13 2019, 12:50:15)
 [Clang 10.0.0 (clang-1000.11.45.5)] on darwin
 Type "help", "copyright", "credits" or "license" for more information.
[>>> import serial
[>>> serial.__version__
 '3.4'
[>>> import matplotlib
[>>> matplotlib.__version__
 '3.0.2'
 >>> 
```

Figure 6: Verify packages installed

2.2.4 Execute the SensorTile Python Animation System

Find the serial port address of the SensorTile

Referring to Tutorial 1, you can now find the serial port address of SensorTile that you have used with Screen to visualize the USB data streaming.

Operate the SensorTile Python Animation System

Navigate into the Python animation system directory and issue the command
$ python SensorTile_Animation_args.py {SerialAddress}

to start the system. SerialAddress should be serial address found in previous step. You shoud notice that the device ending in "1" is the SensorTile and device ending in "2" is the Nucleo board while checking the serial address.

On a typical Mac, the serial port device file address is "/dev/cu.usbmodem1461421". Therefore, an example of command to initialize the Python animation system on this machine is

$ python SensorTile_Animation_args.py /dev/cu.usbmodem1461421

as shown in Figure 7.

```
Xus-MacBook-Pro:SensorTile_Inertial_Sensing_Tutorial xuzh16$ python SensorTile_A
nimation_args.py /dev/cu.usbmodem1461421
```

Figure 7: Start Python animation system

2.2.5 Python Animation System

The Python Animation System is designed to plot x-axis displacement and acceleration data in real time in Python.

The animation system window will will pop up after issuing the command. The animation system is shown in Figure 8 below. There will be an expected delay of data plotting during about the first 5 – 10 seconds of the animation caused by buffering of data at the serial port of the Mac. Later on, the plotting should perform well in real time. However, a constant delay may be observed for some machines. However, this does not limit operation or usability of the SensorTile Animation System.

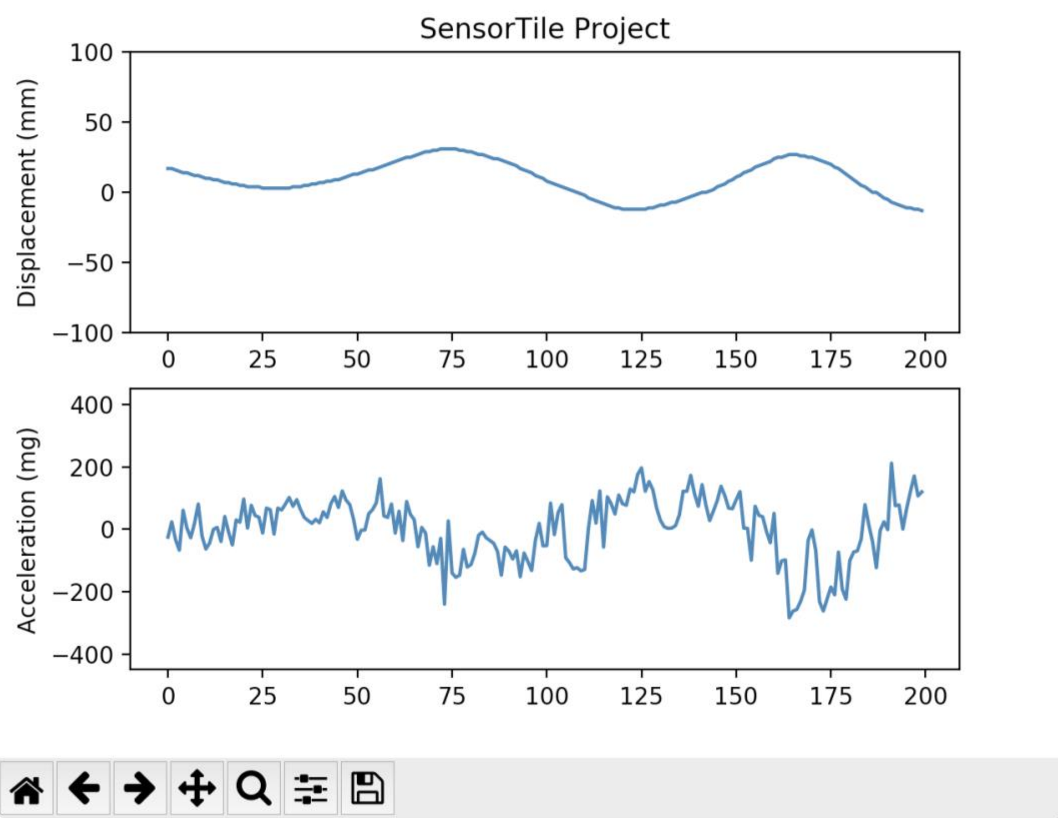

Figure 8: Python Animation System of x-axis displacement and acceleration

You may move SensorTile along its x-axis to visualize the acceleration and displacement.

2.2.6 Shutdown the Python Animation System

It is very important to close the animation system properly, otherwise the system will fail to close the serial connection, which will eventually lead to serial buffer overflow and system failure.

In order to properly shutdown the animation system, you need to click the red "x" in the left-top corner shown in Figure 9.

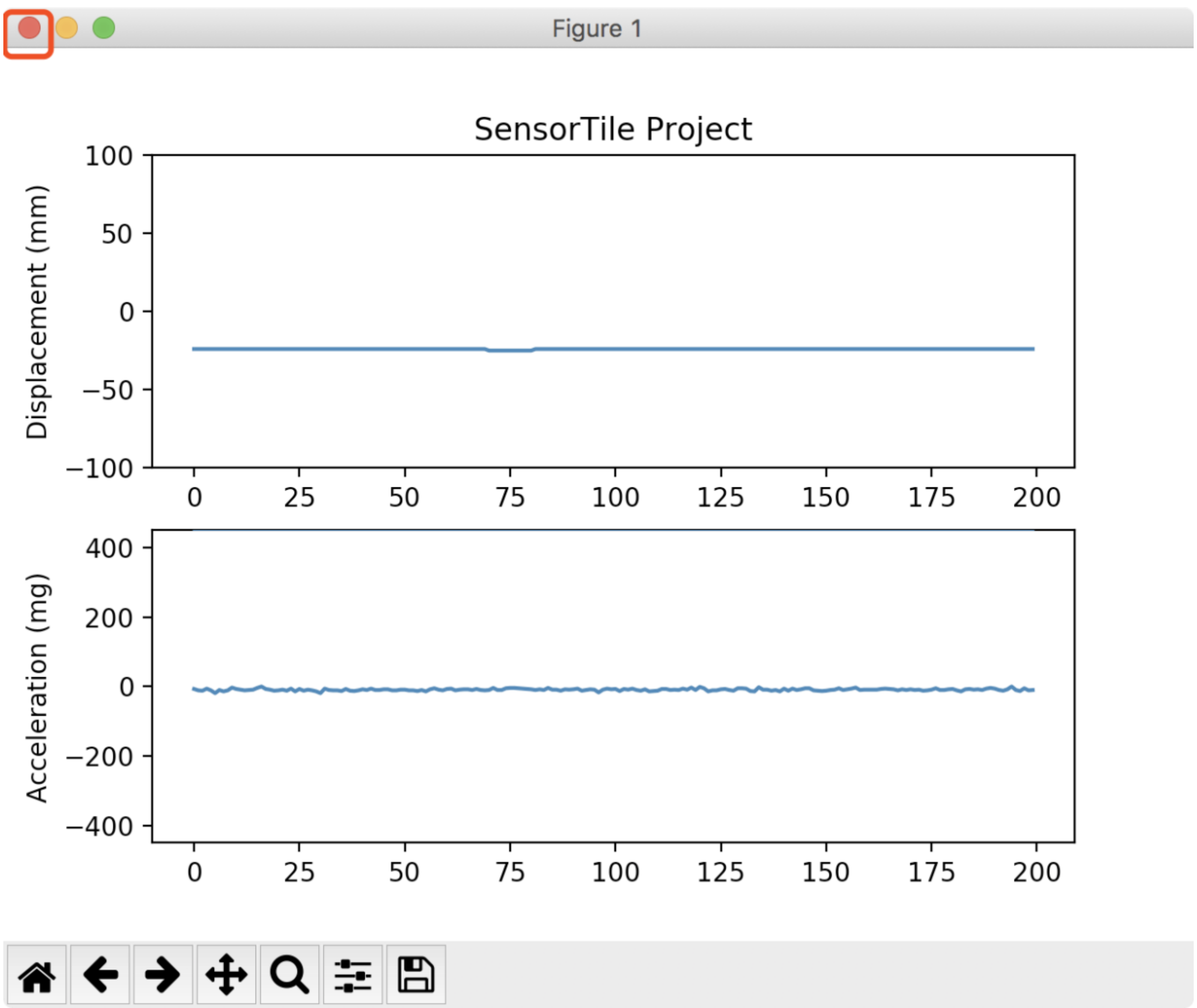

Figure 9: Shutdown the Python Animation System

If the system is closed properly, you are able to see "Close serial connection" message in terminal referring to Figure 10.

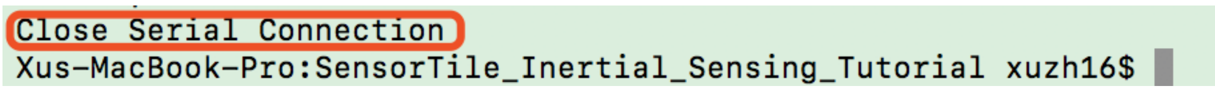

Figure 10: Properly shutdown the Python animation system

Windows System Installation and Operation

2.2.1 Python Installation

This step is required for students who have not installed Python before. If you have any versions of Python (the default Python distribution from python.org, Anaconda Python, PyCharm IDE, or others) installed in your system, make sure that you install all the packages required in Python Package Installation section. The system is compatible for both Python 2 and Python 3. You only need one version of Python to properly use the system.

If you have not installed Python before in Windows, please refer to the link (https://www.python.org/downloads/) to download latest Python 3.7.2. See Figure 11.

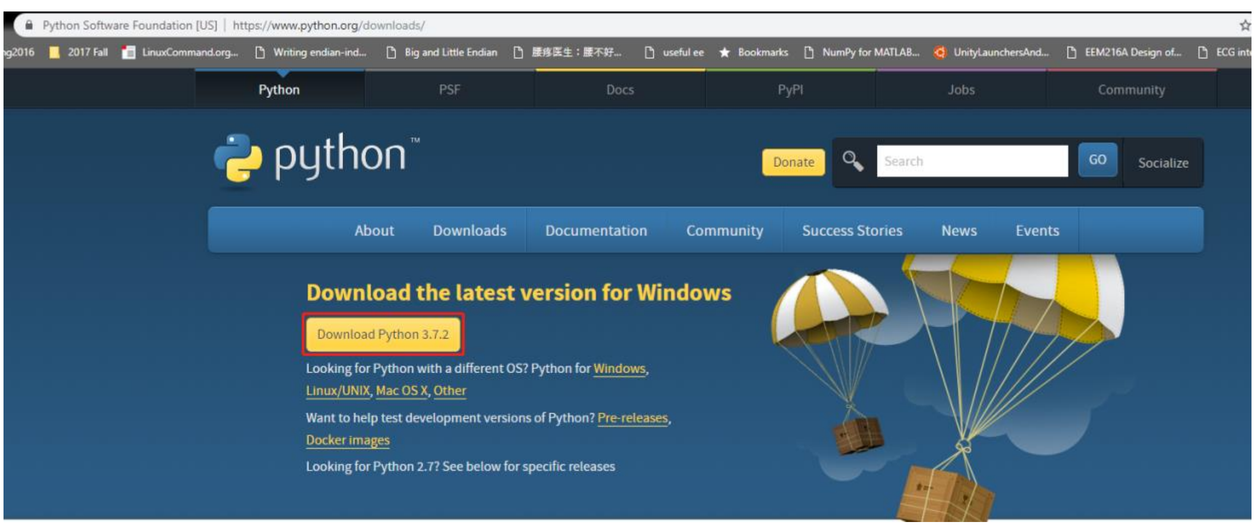

Figure 11: Download Python 3.7.2 for Windows.

Find the downloaded "***python-3.7.2.exe***" link for your machine and double click it to start the installation. Check "***Add Python 3.7 to PATH***". **This step is very important to ensure that users may directly run Python at the Windows command prompt.** See Figure 12 to install Python accordingly.

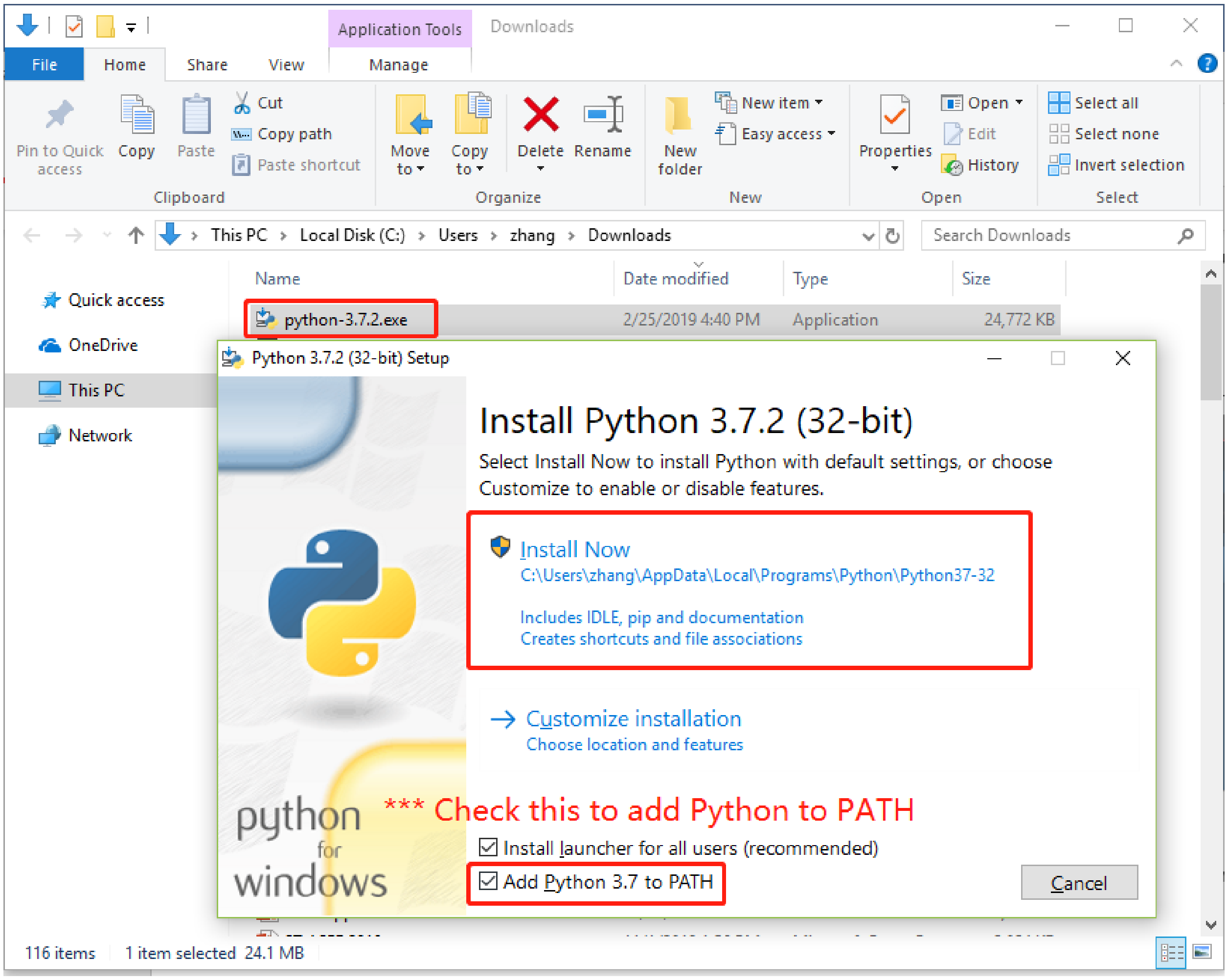

Figure 12: Python Installation

2.2.2 Issue Command Prompt to Check Python Installation

Enter “command” in Windows searching bar and open ***Command Prompt*** as shown in Figure 13.

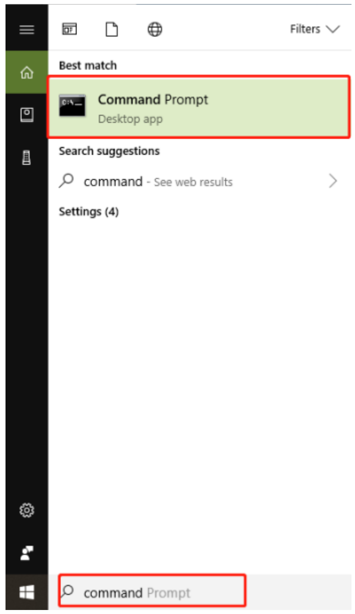

Figure 13: Open Command Prompt in Windows.

Issue “***python***” in the Command Prompt and check the Python version. See Figure 14.

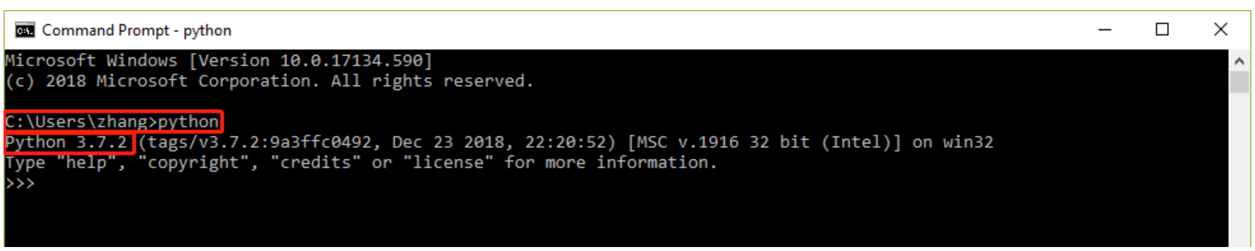

Figure 14: Python in Command Prompt.

Now, you have successfully installed Python 3.7.2. You can use command “***exit()***” to exit Python.

2.2.3 Install Required Python Packages

You can find and install software developed and shared by the Python community through PyPl (pip), where PyPl is embedded with Python. You need to install Python packages to operate the Python animation system.

Upgrade pip

Issue command “***python -m pip install --upgrade pip***” in Windows Command Prompt. See Figure 15.

```
Command Prompt
Microsoft Windows [Version 10.0.17134.590]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\zhang>python
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 22:20:52) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> exit()

C:\Users\zhang>python -m pip install --upgrade pip
Collecting pip
  Using cached https://files.pythonhosted.org/packages/d8/f3/413bab4ff08e1fc4828dfc59996d721917df8e8583ea85385d51125dcef
f/pip-19.0.3-py2.py3-none-any.whl
Installing collected packages: pip
  Found existing installation: pip 18.1
    Uninstalling pip-18.1:
      Successfully uninstalled pip-18.1
Successfully installed pip-19.0.3

C:\Users\zhang>
```

Figure 15: Upgrade pip.

Install PySerial

Issue command “***pip install pyserial==3.4***” in Windows Command Prompt to install pyserial package. See Figure 16.

```
C:\Users\zhang>pip install pyserial==3.4
Collecting pyserial==3.4
  Using cached https://files.pythonhosted.org/packages/0d/e4/2a744dd9e3be04a0c0907414e2a01a7c88bb3915cbe3c8cc06e209f59c3
0/pyserial-3.4-py2.py3-none-any.whl
Installing collected packages: pyserial
Successfully installed pyserial-3.4

C:\Users\zhang>
```

Figure 16: Install PySerial

Install matplotlib

Issue command “***pip install matplotlib==3.0.2***” in Windows Command Prompt to install matplotlib package. See Figure 17.

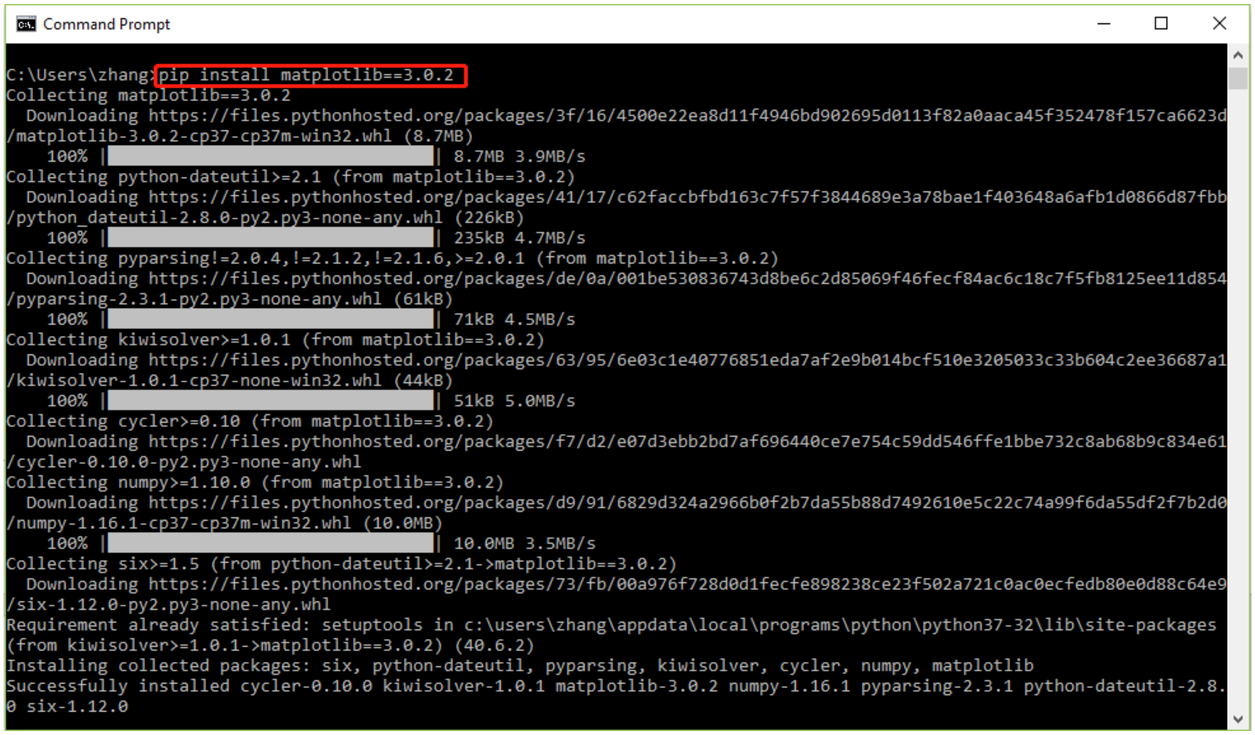

Figure 17: Install matplotlib

Verify package installed

Follow instruction as shown in Figure 18 to verify if Python packages are successfully installed.

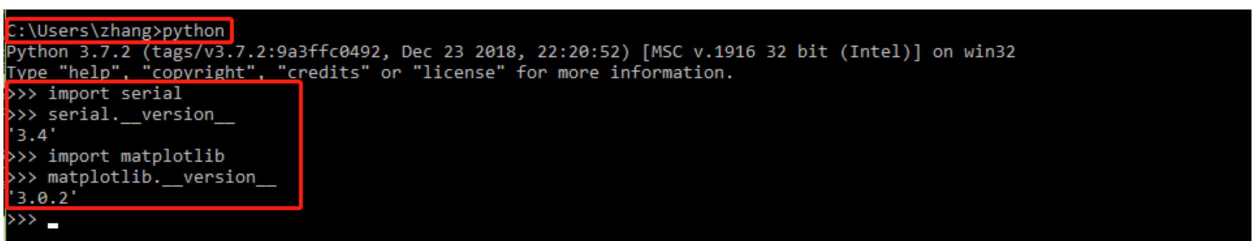

Figure 18: Verify packages installed

2.2.4 Run the SensorTile Python Animation System

Find the directory of Python animation system

Right click on the Python file and check the properties. Find the Location of the Python animation system as shown in Figure 19.

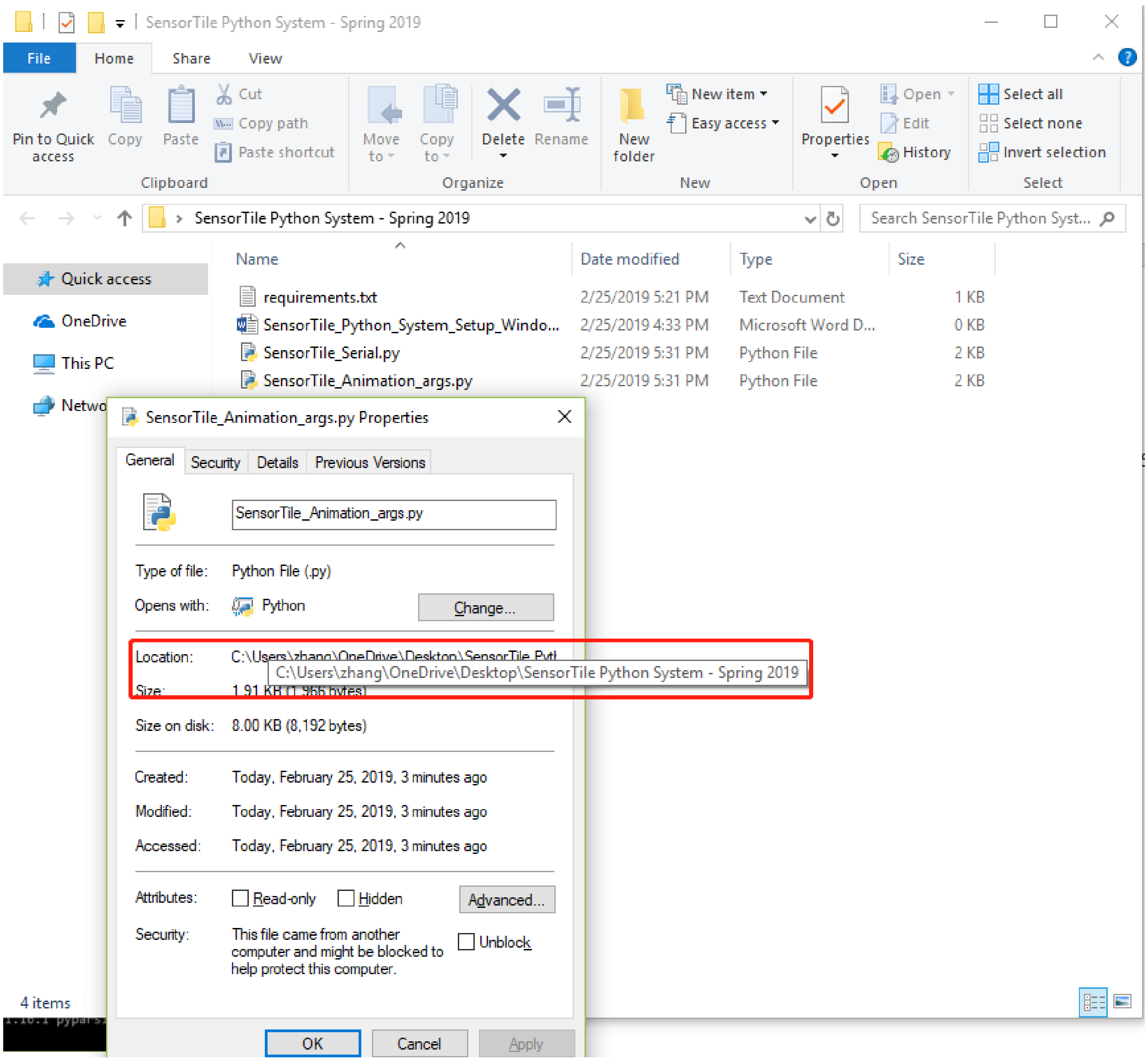

Figure 19: Directory Location of Python System

Navigate into the system directory and operate the system

Navigate into the animation system directory in Windows Command Prompts as shown in Figure 20.

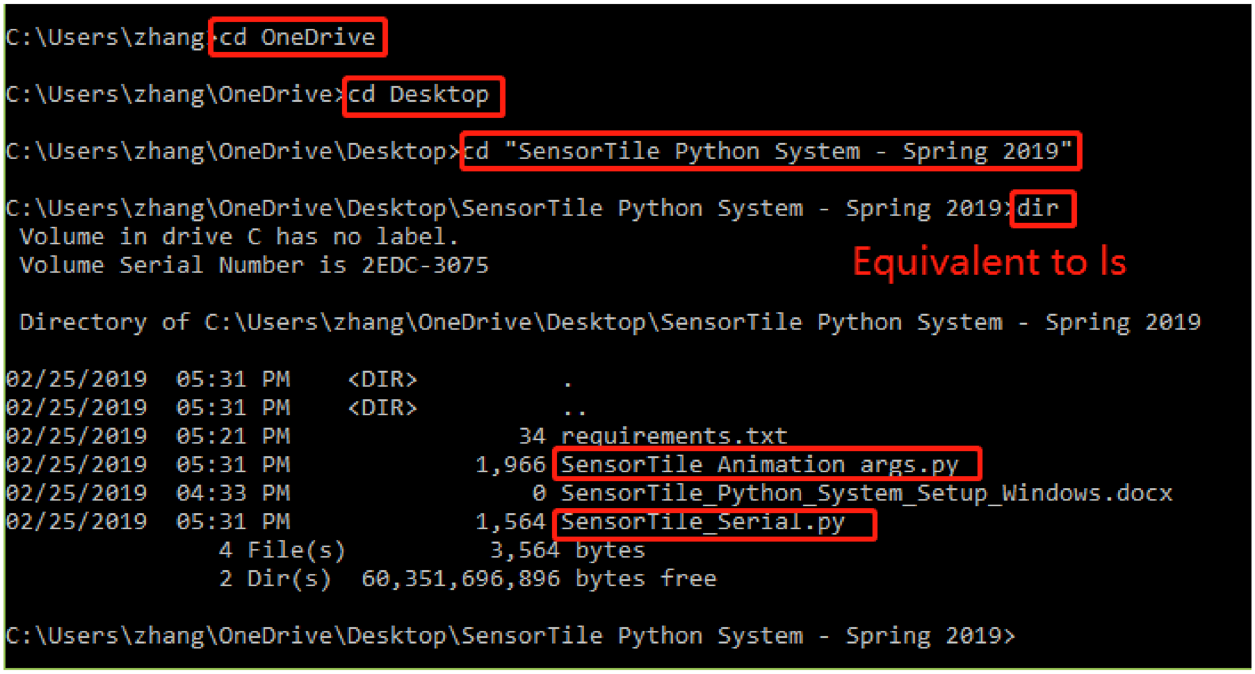

Figure 20: Navigate into the animation system directory in Windows Command Prompts

Find the serial port address of the SensorTile

You need to find the serial port in Device Manager in Windows system according to Tutorial 1. For example, on a typical machine, the SensorTile serial port is “COM3” as shown in Figure 21.

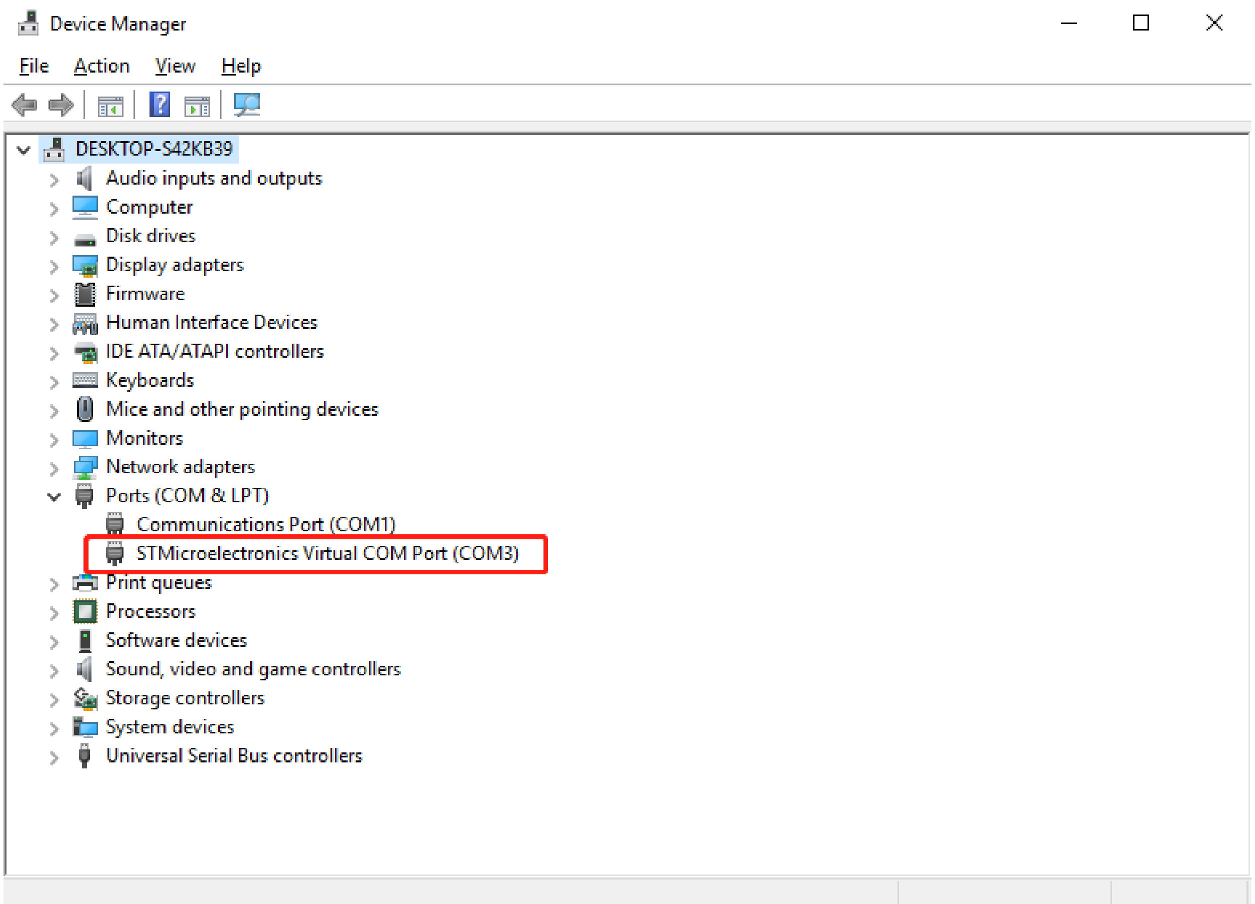

Figure 21: COM port of SensorTile in Device Manager

Operate the Python Animation System

Make sure you have navigated into the Python animation system directory. Then you may issue the command "***python SensorTile_Animation_args.py serialAddress***". For example, the serial port is "COM3" and therefore I should issue "***python SensorTile_Animation_args.py COM3***"to start the Python animation system shown in Figure 22.

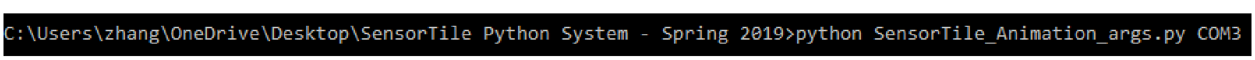

Figure 22: Command to start Python animation system

2.2.5 Python Animation System

The Python Animation System is designed to plot x-axis displacement and acceleration data in real time in Python.

The animation system will pop up after issuing the command and you should observe message "Start Serial Connection" in the command prompt. The animation system is shown in Figure 23 below. There will be an expected delay of data plotting during about the first 5 – 10 seconds of the animation caused by serial buffer issues. Later on, the plotting will perform in near real time. However, a constant delay is also normal for some machines. However, this does not limit the usability of the SensorTile Animation system.

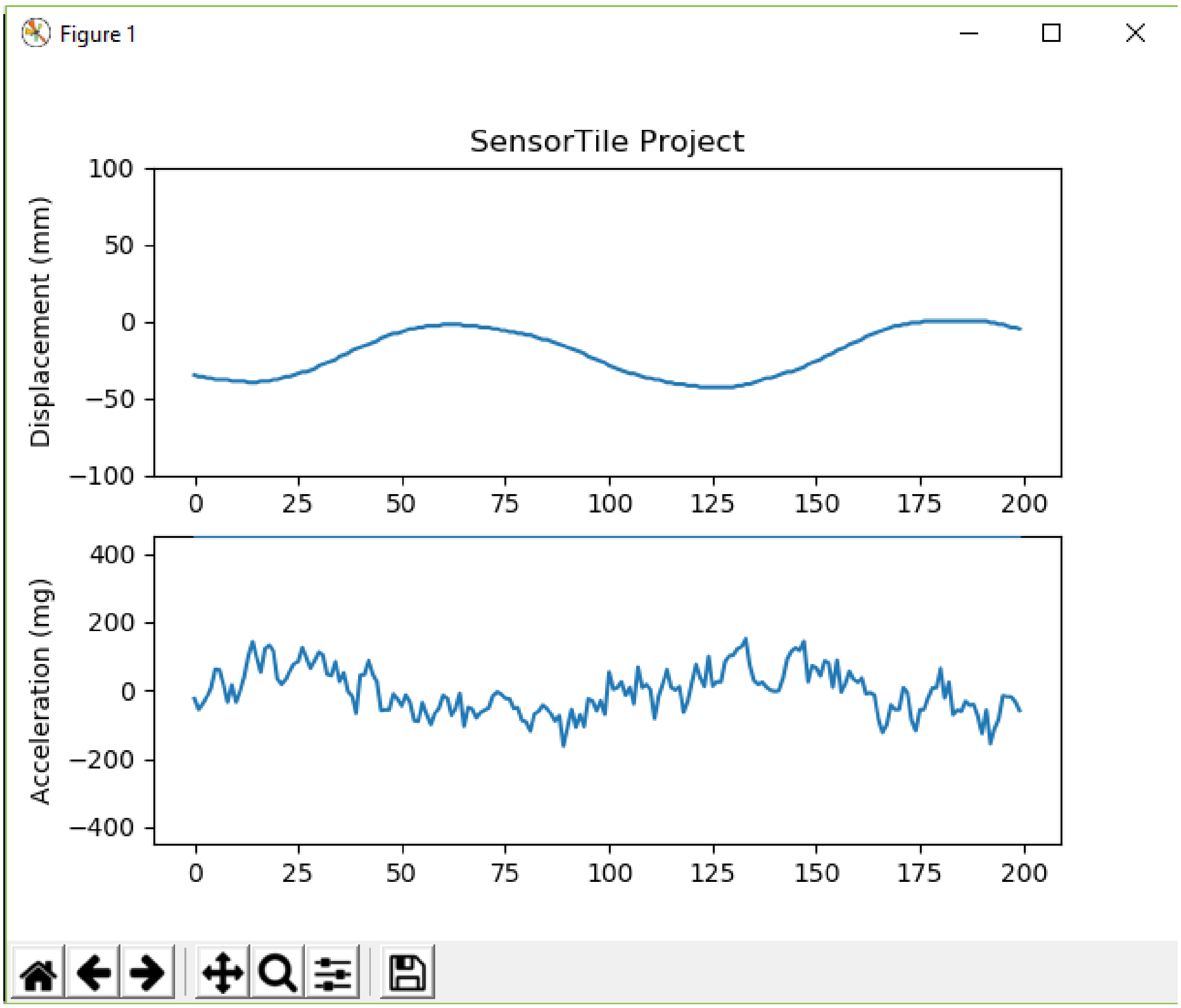

Figure 23: Python animation system of x-axis displacement and acceleration

You may move SensorTile along x-axis to visualize the acceleration and displacement.

2.2.6 Shutdown the Python Animation System

It is very important to close the animation system properly, otherwise the system will fail to close the serial connection, which will eventually lead to serial buffer overflow and system failure.

In order to properly shutdown the animation system, you need to click the red “x” in the right-top corner shown in Figure 24.

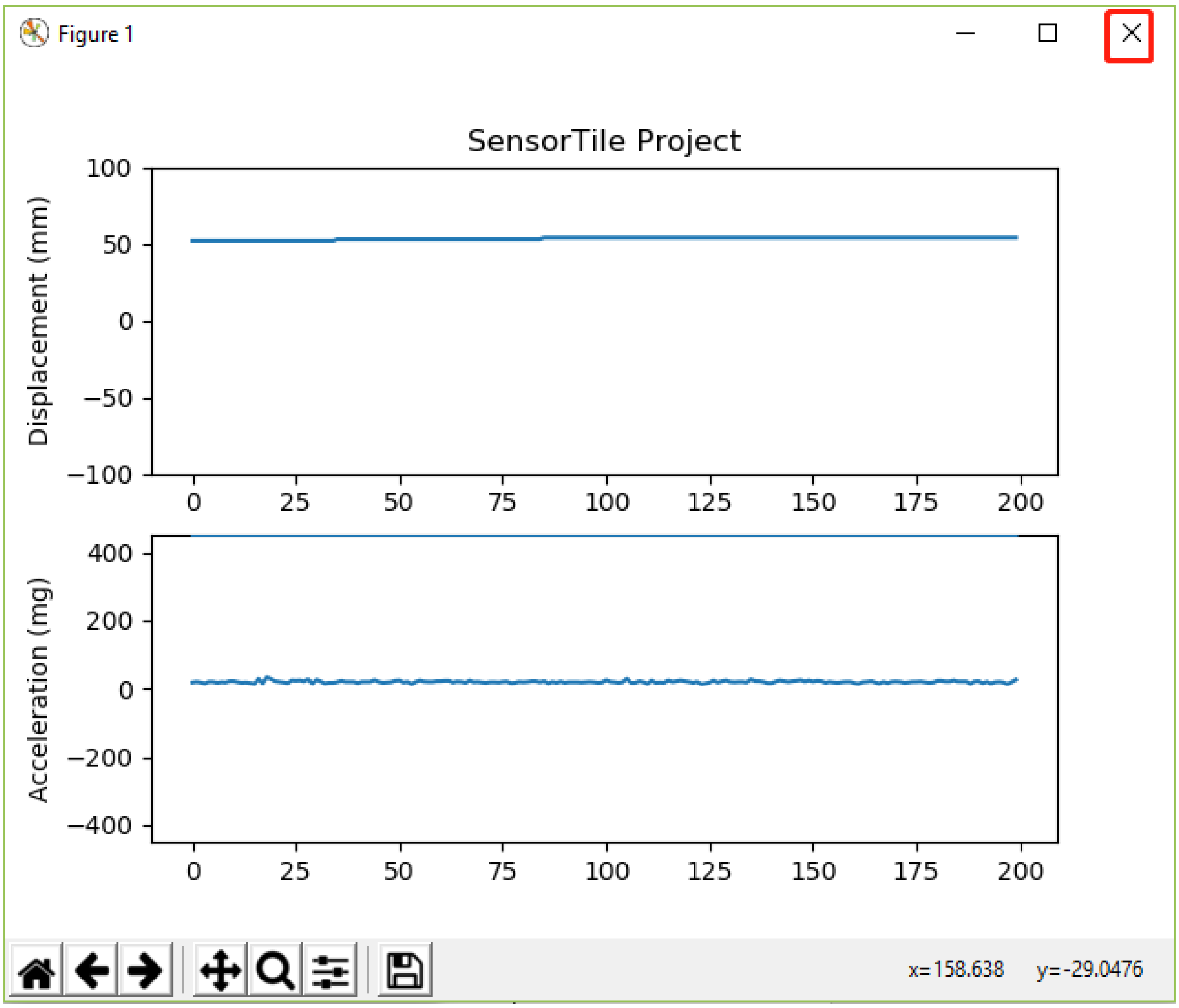

Figure 24: Shutdown the Python Animation System

If the system is closed properly, you are able to see “Close serial connection” message in command prompt referring to Figure 25.

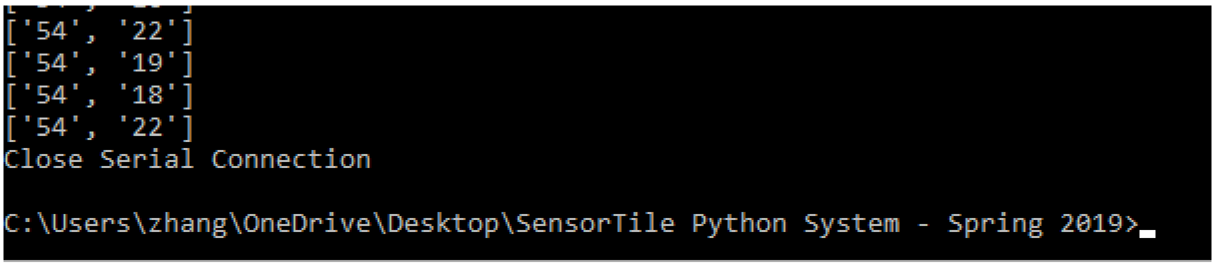

Figure 25: Properly shutdown the Python animation system

3. The Displacement Estimation System

3.1 Displacement Estimation: Trapezoidal integral

Fundamental physics describes the familiar relationship between acceleration, velocity, and displacement. Velocity is determined by performing an integral over time of acceleration. Displacement is determined by performing an integral over time of velocity.

$$v(t) = \int a(t)\,dt + v_0$$
$$d(t) = \int v(t)\,dt + d_0$$

Since the SensorTile acceleration data source is a discrete time, sampled signal, where acceleration is sampled at 100 Hz, a discrete trapezoidal integration is computed on velocity and displacement using the relationships below. Δt in this case equals the sampling period of 1/(100 Hz) = 10milli-seconds.

$$v(1) = v(0) + \left[a(0) + \frac{a(1) - a(0)}{2}\right] * \Delta t = v(0) + \frac{a(1) + a(0)}{2} * \Delta t$$
$$d(1) = d(0) + \left[v(0) + \frac{v(1) - v(0)}{2}\right] * \Delta t = d(0) + \frac{v(1) + v(0)}{2} * \Delta t$$

where a(1) is the current acceleration sample, a(0) is the previous acceleration sample, v(1) is current velocity computation, v(0) is the previous velocity computation, d(1) is the current displacement computation, d(0) is the previous displacement computation, and Δt is 0.01s. In addition, a(x) is the acceleration signal based on acceleration sampled by the SensorTIle system.

In the section, we will introduce 1) Sensor sampling rate, 2) Acceleration sensor signal processing, and 3) Computation and signal processing of displacement data.

3.2 System Sampling Frequency and Sensor Handler Setup

The Datalog application operates with a sensor signal sampling period of 100 milli-seconds, or 10 Hz. In order to provide adequate resolution for displacement computation, In the displacement estimation system, the data acquisition period is reduced to 10 ms in ***main.c*** as shown in Figure 26. Therefore, the system operates with sampling frequency 100 Hz.

```c
/* Private typedef -----------------------------------------------------------*/

/* Private define ------------------------------------------------------------*/

/* Data acquisition period [ms] */
#define DATA_PERIOD_MS (10)
```

Sampling period: 10ms

Figure 26: Displacement estimation system data acquisition period

In addition, this application enables only the ***Accelero_Sensor_Handler()*** in ***main.c*** as shown in Figure 27. You will be able to enable rate gyroscope signals next.

```c
  while (1)
  {
    /* Get sysTick value and check if it's time to execute the task */
    msTick = HAL_GetTick();
    if(msTick % DATA_PERIOD_MS == 0 && msTickPrev != msTick)
    {
      msTickPrev = msTick;
      if(SendOverUSB)
      {
        BSP_LED_On(LED1);
      }
#ifdef NOT_DEBUGGING
      else if (SD_Log_Enabled)
      {
        BSP_LED_On(LEDSWD);
      }
#endif
      RTC_Handler( &RtcHandle );

      Accelero_Sensor_Handler( LSM6DSM_X_0_handle );
      /*
      Gyro_Sensor_Handler( LSM6DSM_G_0_handle );

      Magneto_Sensor_Handler( LSM303AGR_M_0_handle );

      Pressure_Sensor_Handler( LPS22HB_P_0_handle );

      if(!no_T_HTS221)
      {
        Temperature_Sensor_Handler( HTS221_T_0_handle );
      }
      if(!no_H_HTS221)
      {
        Humidity_Sensor_Handler( HTS221_H_0_handle );
      }
```

Disable other sensor handlers that are not needed

Figure 27: Displacement estimation system sensor handler management

3.3 Digital Signal Processing Discrete Time Filters

Two different filters in the system: 1) An anti-aliasing filter to avoid aliasing during signal acquisition, and 2) High pass filtering to eliminate drift in displacement sensing. The filter coefficients are calculated as shown in Figure 28.

datalog_application.c

```c

/*
 * Low Pass Filter coefficients for Anti-Aliasing filter
 */

float fo_AA = 5;
float Wo_AA, IWon_AA, iir_0_AA, iir_1_AA, iir_2_AA;

Wo_AA = 2 * 3.141592654 * fo_AA;
IWon_AA = 2 / (Wo_AA * Tsample);
iir_0_AA = 1 / (1 + IWon_AA);
iir_1_AA = iir_0_AA;
iir_2_AA = iir_0_AA * (1 - IWon_AA);

/*
 * High Pass Filter coefficients for filter operating on velocity and displacement
 */

float fo_h = 0.3;
float Wo_h, IWon_h, iirh_0, iirh_1, iirh_2;

Wo_h = 2 * 3.141592654 * fo_h;
IWon_h = 2 / (Wo_h * Tsample);
iirh_0 = 1 - 1/(1 + IWon_h);
iirh_1 = -iirh_0;
iirh_2 = (1/(1+IWon_h))*(1-IWon_h);

```

Figure 28: Filter coefficients calculation

Anti-aliasing filter is applied to the sampled acceleration (red box) and the high pass filter is applied to calculated velocity and displacement (green box) as shown in Figure 29.

```c
if(SendOverUSB) /* Write data on the USB */
{

    /*
    * Anti-aliasing filter applied to acceleration
    */

      accel_x_direct = (float)acceleration.AXIS_X;
      accel_x_direct_filter = iir_0_AA*accel_x_direct + iir_1_AA*accel_x_direct_prev - iir_2_AA*accel_x_direct_filter_prev;
      accel_x_direct_filter_prev = accel_x_direct_filter;

    /*
    * Integration of acceleration
    */
       velocity = velocity + (accel_x_direct + accel_x_direct_prev)*9.81*Tsample/2; // 1 mg = 9.81 mm/s^2
       accel_x_direct_prev = accel_x_direct;

       /*
        * High pass filter applied to velocity
        */

       velocity_filter = velocity*iirh_0 + velocity_prev*iirh_1 - velocity_filter_prev*iirh_2;
       velocity_filter_prev = velocity_filter;

       /*
        * Integration of velocity
        */

       displacement = displacement + (velocity_filter + velocity_filter_prev)*Tsample/2;
       velocity_prev = velocity;

       /*
        * High pass filter applied to displacement
        */

       displacement_filter = displacement*iirh_0 + displacement_prev*iirh_1 - displacement_filter_prev*iirh_2;
       displacement_prev = displacement;
       displacement_filter_prev = displacement_filter;

```

Figure 29: Filtering and displacement estimation by integration.

3.4 Velocity and displacement by Integration

According to the trapezoidal integral formula introduced above, we can estimate the velocity and displacement (blue box) accordingly as shown in Figure 29 above.

3.5 Data Streaming

Then, the integer part of the x-axis displacement and acceleration is transmitted through the USB serial port as shown in Figure 30. It is important to note that the 9600 Baud rate serial transmission limits the data payload that may be transmitted during each 10 milli-second period. Thus, only the integer parts of the signals are transmitted to reduce data rate below the maximum available.

datalog_application.c

```c
            /*
             * Assignment of integer values for output (only integer part of value is supplied due to
             * communication rate limits)
             *
             */

            floatToInt((accel_x_direct), &d1_ax, &d2_ax, 4);
            floatToInt(accel_x_direct_filter, &d1_x, &d2_x, 4);
            floatToInt(velocity_filter, &d1_v, &d2_v, 4);
            floatToInt(displacement_filter, &d1_df, &d2_df, 4);

            /*
             * Data transmission
             * Note that during each 10ms period, less than 10 characters may be transmitted at the rate of 9600
             * baud and at 10 bits per character.  Thus, communication payload must be reduced to at most
             * two integers.
             *
             */

            sprintf( dataOut, "%i\t%i\r\n",
                    (int)d1_df, (int)d1_ax);
            CDC_Fill_Buffer(( uint8_t * )dataOut, strlen( dataOut ));
```

Figure 30: Displacement and acceleration data streaming.