|  |  |  |  |
| --- | --- | --- | --- |
| Software Installation and Configuration Specification | | | |
| PM1 Pachymeter Automated System Level Test Platform | | | |
|  | Prepared for:  **Occuity Ltd**  **3rd March 2022** |  |  |

Table of Contents

[2 INTRODUCTION 5](#_Toc97219428)

[2.1 Purpose 5](#_Toc97219429)

[2.2 Scope 5](#_Toc97219430)

[2.3 Document Overview 5](#_Toc97219431)

[3 Software InstallatioN 6](#_Toc97219432)

[3.1 Overview 6](#_Toc97219433)

[3.2 Third Party Software 6](#_Toc97219434)

[3.2.1 Overview 6](#_Toc97219435)

[3.2.2 Python Programming Language 6](#_Toc97219436)

[3.2.2.1 Download 6](#_Toc97219437)

[3.2.2.2 Installation 7](#_Toc97219438)

[3.2.2.3 Verification 7](#_Toc97219439)

[3.2.3 NI-DAQmx 7](#_Toc97219440)

[3.2.3.1 Download 7](#_Toc97219441)

[3.2.3.2 Installation 7](#_Toc97219442)

[3.2.3.3 Verification 8](#_Toc97219443)

[3.2.3.3.1 NI Package Manager 8](#_Toc97219444)

[3.2.3.3.2 NI MAX 9](#_Toc97219445)

[3.2.4 NI-DAQmx Python API 9](#_Toc97219446)

[3.2.4.1 Installation 10](#_Toc97219447)

[3.2.5 FT232H Driver 10](#_Toc97219448)

[3.2.5.1 Download 10](#_Toc97219449)

[3.2.5.2 Installation 10](#_Toc97219450)

[3.2.6 Python Packages pyftdi and pyusb 11](#_Toc97219451)

[3.2.7 Python Package PyYAML 12](#_Toc97219452)

[3.2.8 Python Package pylint 12](#_Toc97219453)

[3.2.9 Visual Studio Code (IDE) 13](#_Toc97219454)

[3.2.9.1 Download 13](#_Toc97219455)

[3.2.9.2 Installation 13](#_Toc97219456)

[3.2.10 Git Client 14](#_Toc97219457)

[3.2.10.1 Download 14](#_Toc97219458)

[3.2.10.2 Installation 14](#_Toc97219459)

[3.3 PM1 Pachymeter Automated System Level Test Framework 16](#_Toc97219460)

[3.3.1 Overview 16](#_Toc97219461)

[3.3.2 Cloning The Framework Repository 16](#_Toc97219462)

[4 Software CONFIGURATION 17](#_Toc97219463)

[4.1 Visual Studio Code 17](#_Toc97219464)

[4.1.1 Extensions 17](#_Toc97219465)

[4.1.1.1 Python - Microsoft 17](#_Toc97219466)

[4.1.1.2 YAML – Red Hat 17](#_Toc97219467)

[4.1.1.3 GitLens – GitKraken 17](#_Toc97219468)

[5 PM1 Pachymeter Automated System Level Test Framework 18](#_Toc97219469)

[5.1 Framework Development 18](#_Toc97219470)

Disclaimers

The information contained in this document is for information purposes only and is subject to change without prior notice. While LeoTel Software Systems Limited endeavours to provide the updated document to all project parties, the user is advised to check that they are using the latest version of the document.

LeoTel Software Systems Limited disclaims all warranties, express or limited, including, but not limited, to the implied warranties of merchantability and fitness for a particular purpose, except as provided for in a separate software licence agreement.

trademarks

All trademarks recognised.

change history

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Author | Date of Change | Description |
| 0.1 | James Bridson | 02-Mar-2022 | Initial version. |
| 0.2 | James Bridson | 03-Mar-2022 | Content added. |

document references

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Ref. | Document Id | Title | Author | Issue | Date |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Applicable Standards

|  |  |
| --- | --- |
| Term | Description |
| ISO 13485 | Medical devices – quality management systems – Requirements for regulatory purposes |
| IEC 62304 | Medical device software – software lifecycle processes |
| ISO 14971 | Medical devices – Application of risk management to medical devices |
| ISO/IEC 12207 | Systems and software engineering – Software life cycle processes |

GLOSSARY & ACRONYMS

|  |  |
| --- | --- |
| Term | Description |
| AI | Analogue Input |
| AO | Analogue Output |
| APFI | Analogue Programmable Function Interface |
| CC | Constant Current |
| COTS | Commercial Off-The-Shelf |
| DI | Digital Input |
| DIO | Digital Input / Output |
| DNC | Do Not Connect |
| DNF | Do Not Fit |
| DO | Digital Output |
| DUT | Device Under Test |
| FFC | Flexible Flat Cable |
| FPC | Flexible Printed Circuit |
| GPIO | General Purpose Input / Output |
| IEC | International Electrotechnical Commission |
| ISO | International Organization for Standardization |
| LDA PCB | Laser Driver and Amplifier Printed Circuit Board |
| LSC PESS | Linear Scanner Controller Programmable Electronic Sub-System |
| OTS | Off-The-Shelf |
| Pachymeter | A scientific instrument used to measure the thickness of the cornea using laser, light waves or ultrasound. |
| PCB | Printed Circuit Board |
| PCBA | Printed Circuit Board Assembly |
| PDA PESS | Pachymeter Data Acquisition Programmable Electronic Sub-System |
| PE PESS | Pachymeter Environmental Programmable Electronic Sub-System |
| PFI | Programmable Function Interface |
| PI PESS | Pachymeter Interface Programmable Electronic Sub-System |
| TBC | To Be Confirmed |
| UUT | Unit Under Test |
| WP*n* | Work Package *n* |

# INTRODUCTION

This guide documents the software installation and configuration for the Occuity .

## Purpose

This document is primarily intended to be used:

* By software engineers during the project when installing and configuring the software components.

## Scope

This document covers installation and configuration of the software components of the on a Windows 10 (64-bit) platform where the user is able to gain administrative privileges when required.

## Document Overview

The remainder of this document is organised into sections as follows:

Section 3 documents the software installation

Section 4 documents the software configuration

# Software InstallatioN

## Overview

The software components in the PM1 Pachymeter Automated System Level Test Platform are summarised in the following diagram:

[INSERT DIAGRAM HERE]

## Third Party Software

### Overview

The third party software components in the PM1 Pachymeter Automated System Level Test Platform are summarised in the following diagram:

[INSERT DIAGRAM HERE]

Third party software components are those software components provided by third parties to provide drivers for hardware components as well as library and language resources used to implement the PM1 Pachymeter Automated System Level Test Platform Python Framework.

### Python Programming Language

Python is an interpreted, object-orientated, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together.

Python is used to implement the PM1 Pachymeter Automated System Level Test Platform Python Framework gluing together the various hardware entities within the platform, implementing an object-orientated abstraction layer to that hardware and then to implement scripts to support automated testing.

The PM1 Pachymeter Automated System Level Test Platform is based on Python 3.9.5 (64-bit).

[ADD REASON WHY / DEPENDENCIES]

All Python releases are Open Source.

#### Download

The Python 3.9.5 Windows Installer (64-bit) file can be downloaded from this link:

<https://www.python.org/ftp/python/3.9.5/python-3.9.5-amd64.exe>

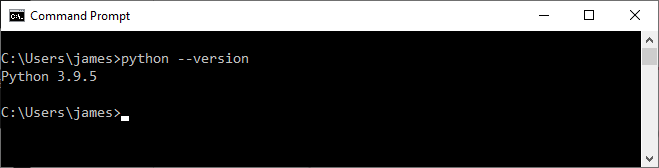
#### Installation

Run the downloaded Python Windows Installer:

* Make sure you select ‘Install Launcher for all users (recommended)’ and ‘Add Python 3.9 to PATH’ checkboxes.
* Now select ‘Customize installation – choose location and features’. Make sure all the ‘Optional Features’ are selected.
* Click ‘Next’ and select ‘Install for all users’, ‘Associate files with Python’, ‘Create shortcuts for installed applications’, ‘Add Python to environment variables’ and ‘Precompile standard library’.
* The installation path should be displayed as ‘C:\Program Files\Python39’ (or similar all user program file path).
* Now click ‘Install’.
* On display of the ‘Setup was successful’ message click on ‘Disable path length limit’ to make sure Windows Python can bypass the 260-character MAX\_PATH limit.
* Now click ‘Close’.

#### Verification

Once the Python Windows Installer has completed successfully check that the Python interpreter is available from a command prompt as shown below:



### NI-DAQmx

NI-DAQmx provides support when using NI data acquisition and signal conditioning devices (i.e. the NI USB-6363 in the case of the PM1 Pachymeter Automated System Level Test Platform).

The PM1 Pachymeter Automated System Level Test Platform is based on NI-DAQmx 21.3 (64-bit).

#### Download

NI-DAQmx 21.3 can be downloaded from this link:

<https://www.ni.com/en-gb/support/downloads/drivers/download.ni-daqmx.html#428058>

#### Installation

Run the downloaded NI Package Manager to install NI-DAQmx 21.3:

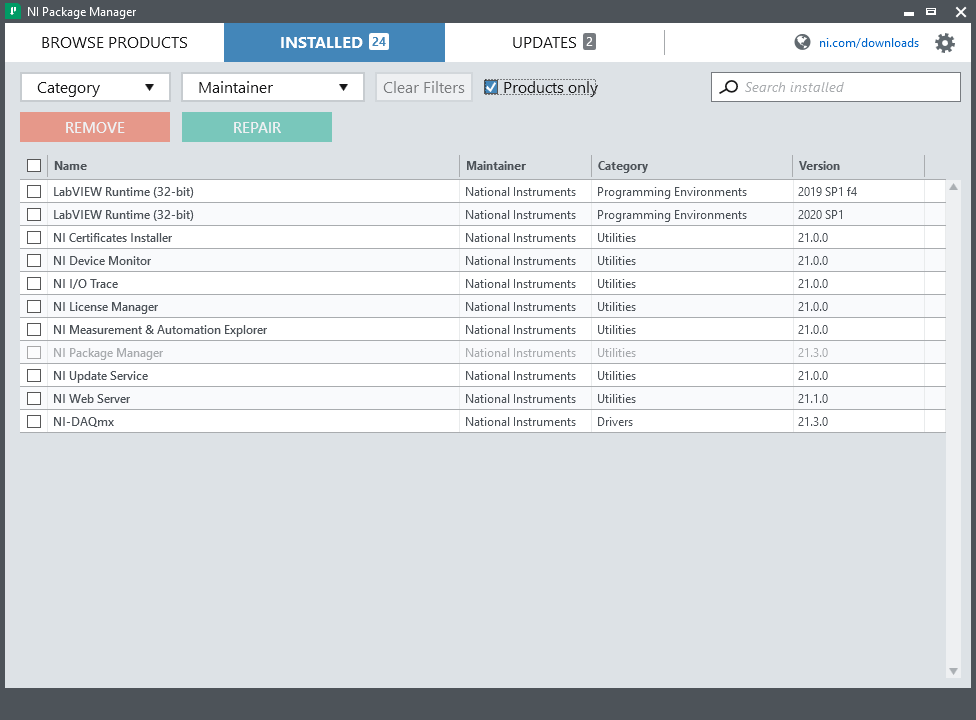
* Accept the license agreement.
* Ensure ‘Disable Windows fast startup’ is selected.
* Click on ‘Next’.
* Click on ‘Next to install the NI Package Manager.
* Once installed NI Package Manager will display ‘Additional items you may wish to install’.
* Select all the items and click on ‘Next’.
* Accept the license agreement.
* Review the summary and click on ‘Next’.
* Once the installation has been completed you will be asked if you wish to check for updates periodically, click on ‘No’.
* When asked ‘Do you want to participate in the NI Customer Experience Improvement Program’ select ‘No’ and click on ‘Ok’.

You must reboot to complete the installation operation and make sure all the NI services are correctly initialised.

#### Verification

##### NI Package Manager

Once the NI Package Manager has completed successfully, and you have rebooted, check that all the components shown below have been installed by launching the NI Package Manager app from the Windows Start Menu:



The NI Package Manager allows you to update your installed options if it should be required.

##### NI MAX

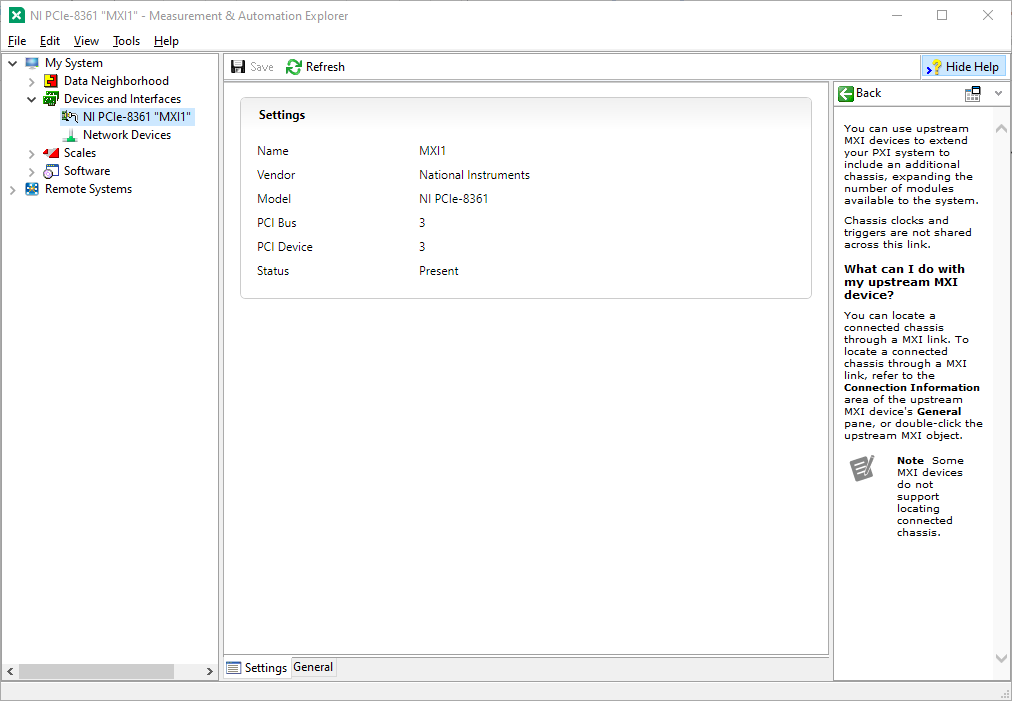
NI Measurement & Automation Explorer (MAX) provides access to the NI hardware (i.e. the NI USB-6363 in the case of the PM1 Pachymeter Automated System Level Test Platform).

It is a free piece of software that cannot be downloaded by itself but is included, and automatically installed, with all NI drivers.

With NI-MAX, you can:

* Configure the NI hardware and software
* Export/Import the system configuration
* Create and edit channels, tasks, interfaces, scales, and virtual instruments
* Create simulated devices
* Execute system diagnostics and run test panels
* View devices/instruments connected to your system and software installed on your system

NI MAX is not normally needed in order to use the PM1 Pachymeter Automated System Level Test Platform but can be convenient for checking the NI hardware connection and executing system diagnostics if required.



### NI-DAQmx Python API

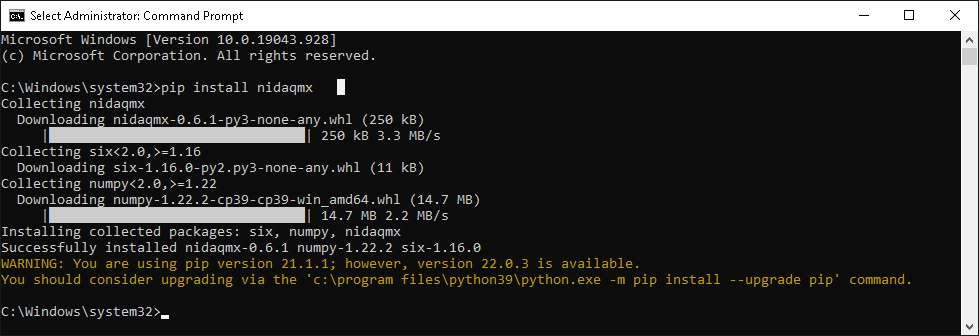
The Python package nidaqmx contains an API (Application Programming Interface) for interacting with the NI-DAQmx driver installed previously. This package was created and is supported by NI. The package is implemented as a complex, highly object-orientated wrapper around the NI-DAQmx C API using the ctypes Python library.

#### Installation

The Python package nidaqmx can be installed with the pip application which will also install any necessary dependencies.

Whilst it is possible to install Python packages without administrator privileges, for a test machine that may be used my many users it is more convenient to install them with administrative privileges.

In order to do this launch a command prompt using ‘Run as Administrator’ and run the following command:



### FT232H Driver

In order to use the FT232H breakout board in SPI mode with the Python package pyftdi/pyusb we need to also install drivers for libusbK support.

The easiest way to do this is to use an application called Zadig.

#### Download

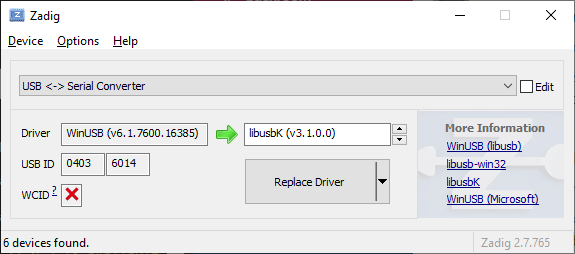
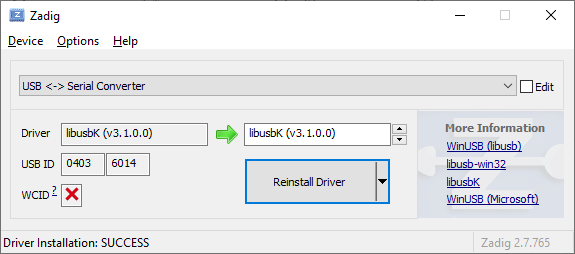
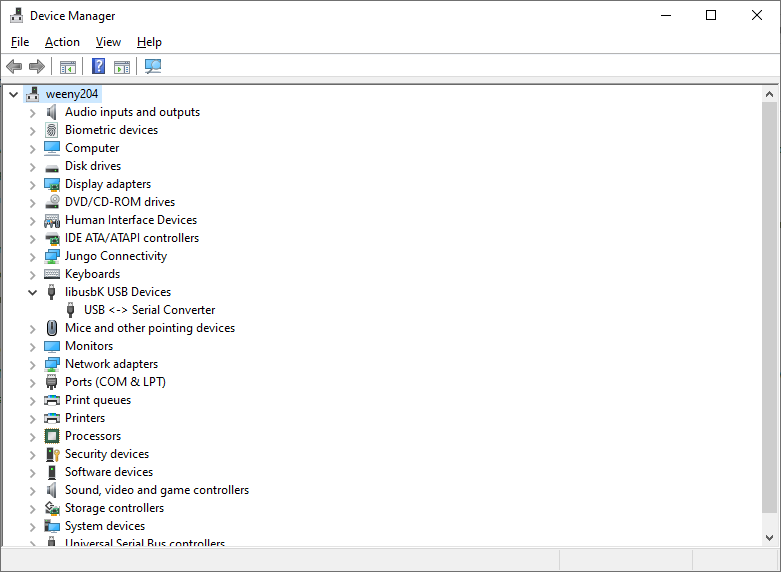
Zadig can be downloaded from the following link:

<https://github.com/pbatard/libwdi/releases/download/v1.4.1/zadig-2.7.exe>

#### Installation

Unplug all FTDI devices from the test computer. After all the FTDI devices are unplugged plug in the PM1 Pachymeter Automated System Level Test Platform FT232H breakout board via a USB-C cable to the test computer so it is the only FTDI device connected to the test computer.

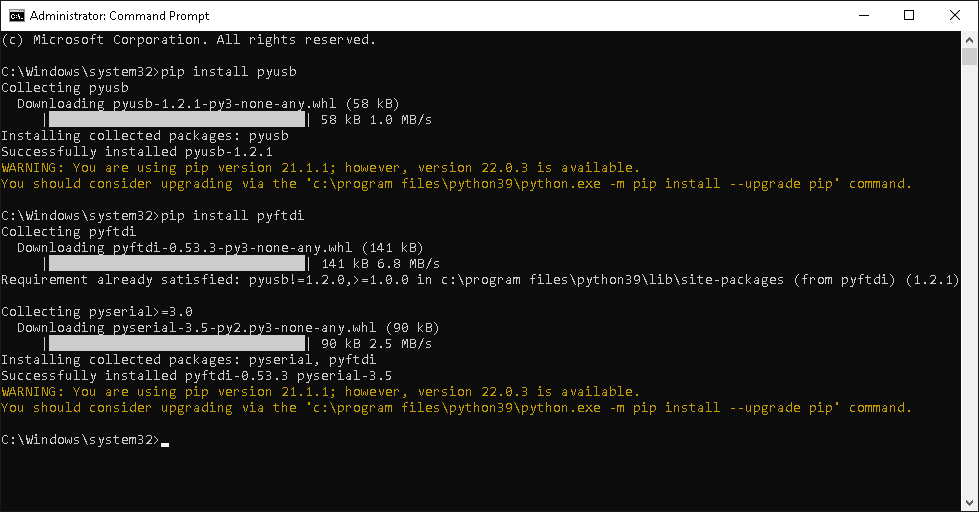
Launch the downloaded Zadig application:

* When asked about the allowing Zadig to automatically update online, click on ‘No’.
* Click on the options menu and select ‘List all Devices’.
* Locate the FT232H serial converter device in the drop down list. Be careful you are picking the correct device with USB ID 0403 6014.
* Click the ‘Up/Down’ arrow buttons on the replacement driver box and select the ‘libusbK’ based driver:  
  
* Click on ‘Replace Driver’.
* Once successfully installed Zadig should update as follows:  
    
  and device manager should show the USB <-> Serial Converter under the libusbK USB Devices node:  
  

### Python Packages pyftdi and pyusb

In order to use the FT232H breakout board for SPI communication the PM1 Pachymeter Automated System Level Test Platform Framework uses the Python packages pyftdi and pyusb.

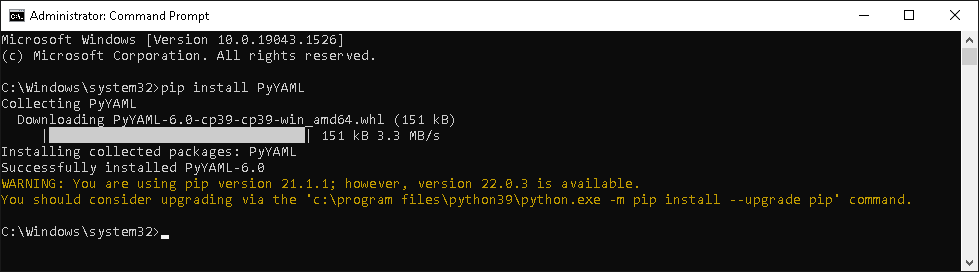
As with the Python package nidaqmx these are installed with the pip application from a command prompt that has been launched via ‘Run as Administrator’:



### Python Package PyYAML

The PM1 Pachymeter Automated System Level Test Platform Framework configuration file uses the Python package PyYAML to parse the YAML format configuration file.

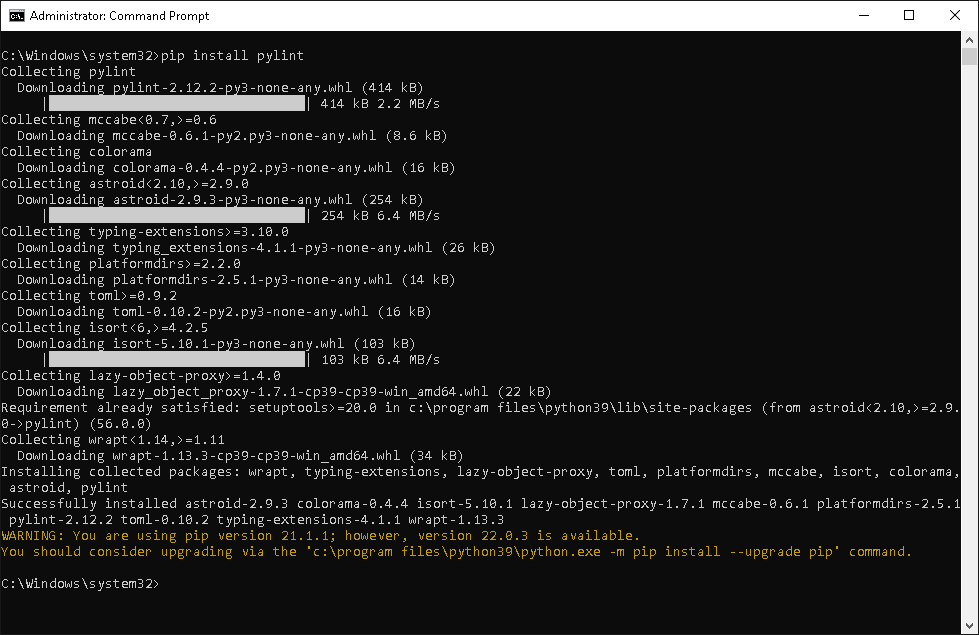
As with the Python package nidaqmx this is installed with the pip application from a command prompt that has been launched via ‘Run as Administrator’:



### Python Package pylint

The PM1 Pachymeter Automated System Level Test Platform Framework uses the Python package pylint to analyse and check the quality of the Python code.

As with the Python package nidaqmx this is installed with the pip application from a command prompt that has been launched via ‘Run as Administrator’:



### Visual Studio Code (IDE)

The PM1 Pachymeter Automated System Level Test Platform Framework development has been carried out using Visual Studio Code (64-bit).

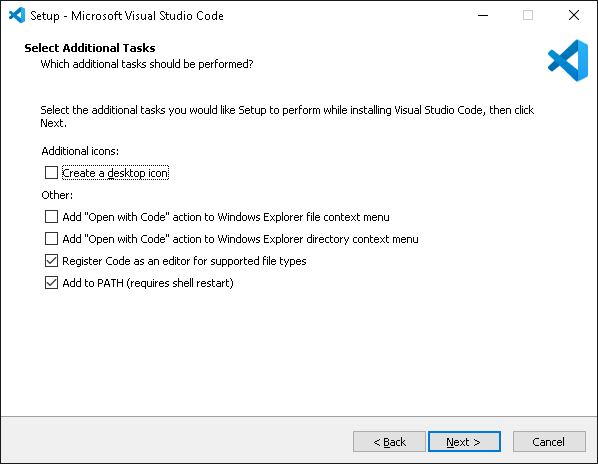
#### Download

Visual Studio Code (64-bit) can be downloaded from the following link:

<https://code.visualstudio.com/sha/download?build=stable&os=win32-x64>

#### Installation

Run the downloaded Visual Studio Code (64-bit) setup application to install Visual Studio Code:

* Accept the license agreement.
* Setup should default to ‘C:\Program Files\Microsoft VS Code’ (or similar all user program file path).
* Click ‘Next’.
* The ‘Select Start Menu Folder’ dialog will be displayed. Click ‘Next’.
* Accept the default components:  
    
  and click ‘Next’.
* Click ‘Install’.
* Once setup has finished untick the ‘Launch Visual Studio Code’ button and click ‘Finish’.

### Git Client

The PM1 Pachymeter Automated System Level Test Platform Framework uses the Git distributed version control system for source control and management tasks.

The PM1 Pachymeter Automated System Level Test Platform Framework development has been carried out using Git for Windows (64-bit).

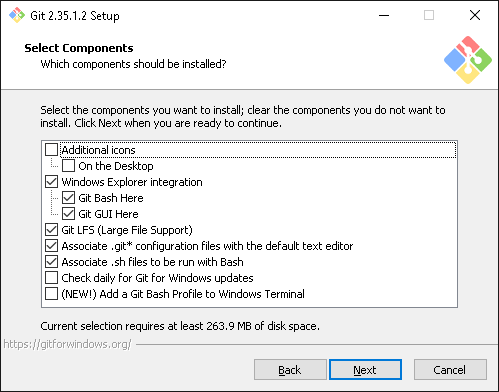
#### Download

Git for Windows 2.35.1(2) (64-bit) can be downloaded from the following link:

<https://github.com/git-for-windows/git/releases/download/v2.35.1.windows.2/Git-2.35.1.2-64-bit.exe>

#### Installation

Run the downloaded Git-2.35.1.2-64-bit.exe application to install Git for Windows:

* Accept the license agreement.
* Setup should default to ‘C:\Program Files\Git’ (or similar all user program file path).
* Click ‘Next’.
* Accept the default components:  
    
  and click ‘Next’.
* The ‘Select Start Menu Folder’ dialog will be displayed. Click ‘Next’.
* In the ‘Choosing the default editor used by Git’ dialog select ‘Use Visual Studio Code as Git’s default editor’ and click ‘Next’.
* Select ‘Let Git decide’ on the ‘Adjusting the name of the initial branch in new repositories’ dialog and click ‘Next’.
* Select ‘Git from the command line and also from 3rd-party software’ on the ‘Adjusting your PATH environment’ dialog and click ‘Next’.
* Select ‘Use bundled OpenSSH’ on the ‘Choosing the SSH executable’ dialog and click ‘Next’.
* Select ‘Use the OpenSSL library’ on the ‘Choosing HTTPS transport backend’ dialog and click ‘Next’.
* Select ‘Checkout Windows-style, commit Unix-style line endings’ on the ‘Configuring the line ending conversions’ dialog and click ‘Next’.
* Select ‘Use MinTTY (the default terminal of MSYS2)’ on the ‘Configuring the terminal emulator to use with Git Bash’ dialog and click ‘Next’.
* Select ‘Default (fast-forward or merge)’ on the ‘Choose the default behaviour of ‘git pull’’ dialog and click ‘Next’.
* Select ‘Git Credential Manager’ on the ‘Choose a credential helper’ dialog and click ‘Next’.
* Select ‘Enable file system caching’ (only) on the ‘Configure extra options’ dialogs and click ‘Next’ and then click ‘Install’.
* Once setup has finished untick the ‘Launch Git Bash’ and ‘View Release Notes’ buttons and click ‘Finish’

## PM1 Pachymeter Automated System Level Test Framework

### Overview

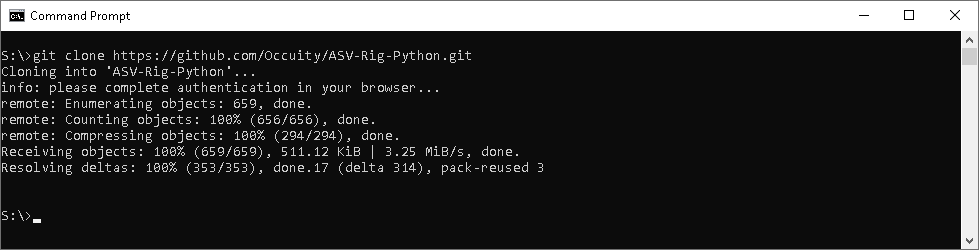
The PM1 Pachymeter Automated System Level Test Platform Framework components are summarised in the following diagram:

[INSERT DIAGRAM HERE]

### Cloning The Framework Repository

The PM1 Pachymeter Automated System Level Test Platform Framework components are managed in a github repository.

You can clone a copy of the repository to your local machine using the git command line:



# Software CONFIGURATION

## Visual Studio Code

### Extensions

The first time you open Visual Studio Code you should configure some extensions to make it easier to develop and use the PM1 Pachymeter Automated System Level Test Platform Framework.

#### Python - Microsoft

Open Visual Studio Code.

Click on the Extensions icon on the left hand side toolbar (or press Ctrl-Shift-X) and enter the search term ‘Python’ into the search bar.

Locate the Microsoft Python extension in the results and click on ‘Install’:

Once installed a ‘Get started with Python development’ panel should be displayed, close the panel.

You now need to configure a Python interpreter location. To do this type ‘Ctrl-Shift-P’ and type the command ‘Python: Python: Select Python Interpreter’ and check that the Python you installed is selected. If not select it now.

You now need to configure a Python linter location. To do this type ‘Ctrl-Shift-P’ and type the command ‘Python: Python: Select Linter’ select pylint as the linter.

Restart Visual Studio Code.

#### YAML – Red Hat

Open Visual Studio Code.

Click on the Extensions icon on the left hand side toolbar (or press Ctrl-Shift-X) and enter the search term ‘YAML into the search bar.

Locate the Red Hat YAML extension in the results and click on ‘Install’:

Once installed, restart Visual Studio Code.

#### GitLens – GitKraken

Open Visual Studio Code.

Click on the Extensions icon on the left hand side toolbar (or press Ctrl-Shift-X) and enter the search term ‘GitLens’ into the search bar.

Locate the GitKraken GitLens extension in the results and click on ‘Install’:

Once installed, restart Visual Studio Code.

# PM1 Pachymeter Automated System Level Test Framework

## Framework Development

TBD

But essentially open VSCode, Navigate to ‘File->Open Workspace from File’, choose the file ‘workspace.code-workspace’ in the cloned repository top level directory.