SOFTWARE INSTALLATION AND CONFIGURATION SPECIFICATION

PM1 Pachymeter Automated System Level Test Platform

Prepared for:

Occuity Ltd

3rd March 2022

Registered in England and Wales: Company Number 04493471
Registered Office: Lines House, 78 High Street, Stevenage, SG1 3DW, United Kingdom

LSSL/20220302/01 0.2

1 TABLE OF CONTENTS

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*

2 INTRODUCTION

This guide documents the software installation and configuration for the Occuity PM1 Pachymeter Automated System Level Test Platform.

2.1 PURPOSE

This document is primarily intended to be used:

- By software engineers during the PM1 Pachymeter Automated System Level Test Platform project when installing and configuring the software components.

2.2 SCOPE

This document covers installation and configuration of the software components of the PM1 Pachymeter Automated System Level Test Platform on a Windows 10 (64-bit) platform where the user is able to gain administrative privileges when required.

2.3 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

3 SOFTWARE INSTALLATION

3.1 OVERVIEW

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

[INSERT DIAGRAM HERE]

3.2 THIRD PARTY SOFTWARE

3.2.1 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.

3.2.2 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.

3.2.2.1 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

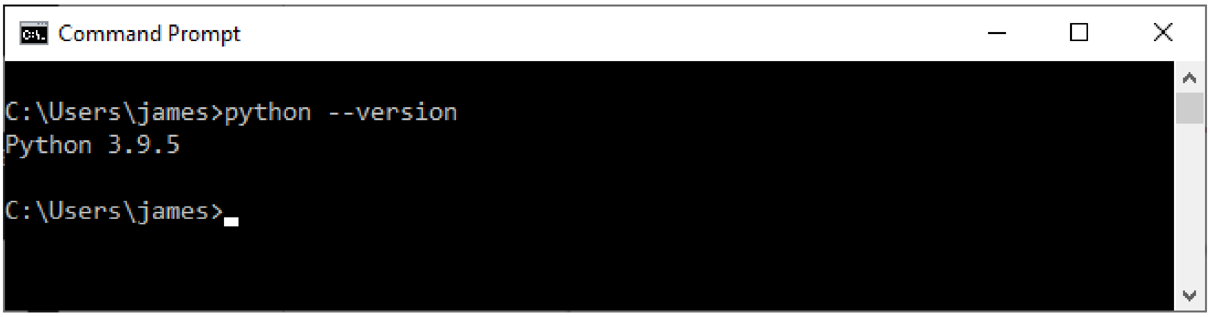
3.2.2.2 Installation

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'.

3.2.2.3 Verification

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



```
C:\Users\james>python --version
Python 3.9.5

C:\Users\james>_
```

3.2.3 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).

3.2.3.1 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

3.2.3.2 Installation

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

- Accept the license agreement.
- Ensure 'Disable Windows fast startup' is selected.

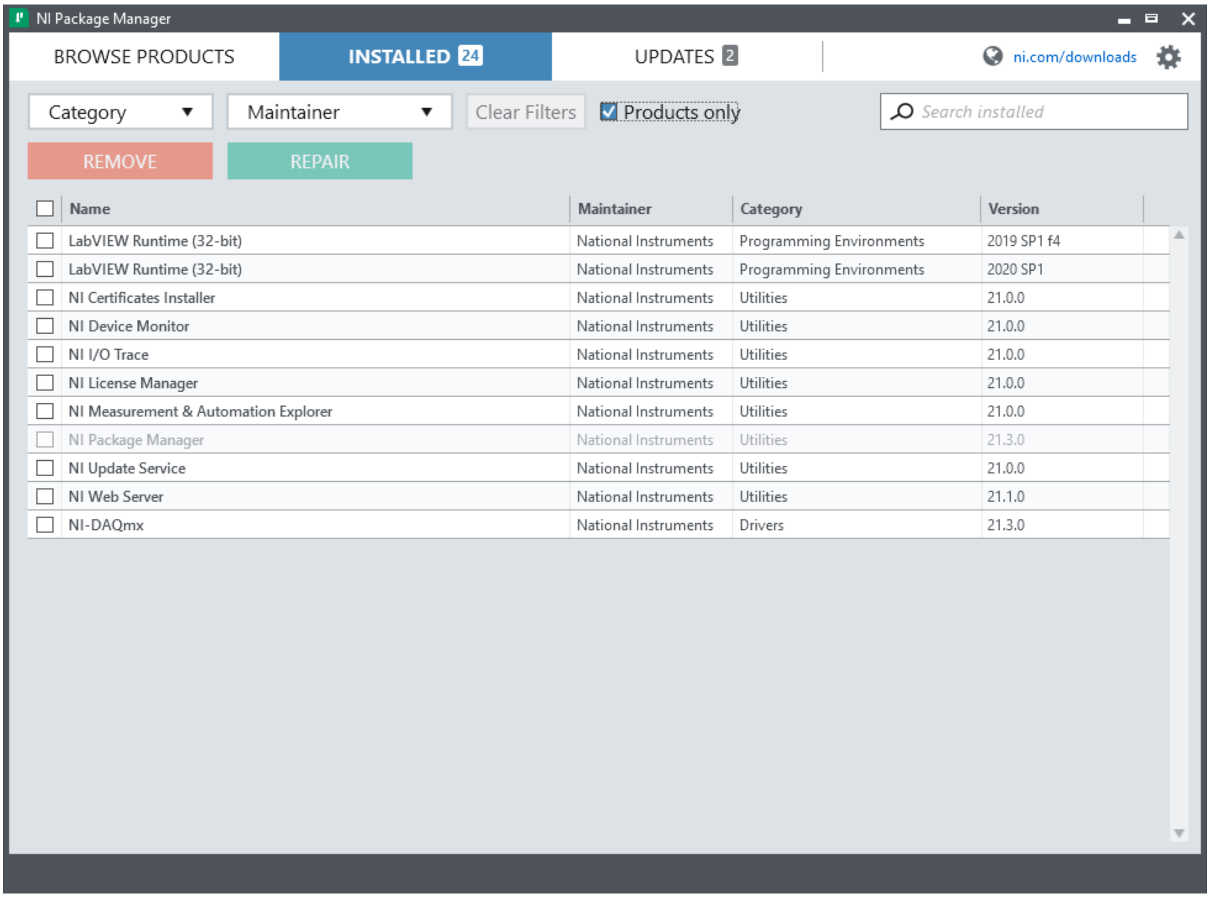
- Click on ‘Next’.
- Click on ‘Next to install the NI Package Manager.
- Once installed NI Package Manager will display ‘Additional items you may wish to install’.
- Select all the items and click on ‘Next’.
- Accept the license agreement.
- Review the summary and click on ‘Next’.
- Once the installation 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.

3.2.3.3 VERIFICATION

3.2.3.3.1 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:



☐	Name	Maintainer	Category	Version
☐	LabVIEW Runtime (32-bit)	National Instruments	Programming Environments	2019 SP1 f4
☐	LabVIEW Runtime (32-bit)	National Instruments	Programming Environments	2020 SP1
☐	NI Certificates Installer	National Instruments	Utilities	21.0.0
☐	NI Device Monitor	National Instruments	Utilities	21.0.0
☐	NI I/O Trace	National Instruments	Utilities	21.0.0
☐	NI License Manager	National Instruments	Utilities	21.0.0
☐	NI Measurement & Automation Explorer	National Instruments	Utilities	21.0.0
☐	NI Package Manager	National Instruments	Utilities	21.3.0
☐	NI Update Service	National Instruments	Utilities	21.0.0
☐	NI Web Server	National Instruments	Utilities	21.1.0
☐	NI-DAQmx	National Instruments	Drivers	21.3.0

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

3.2.3.3.2 NI MAX

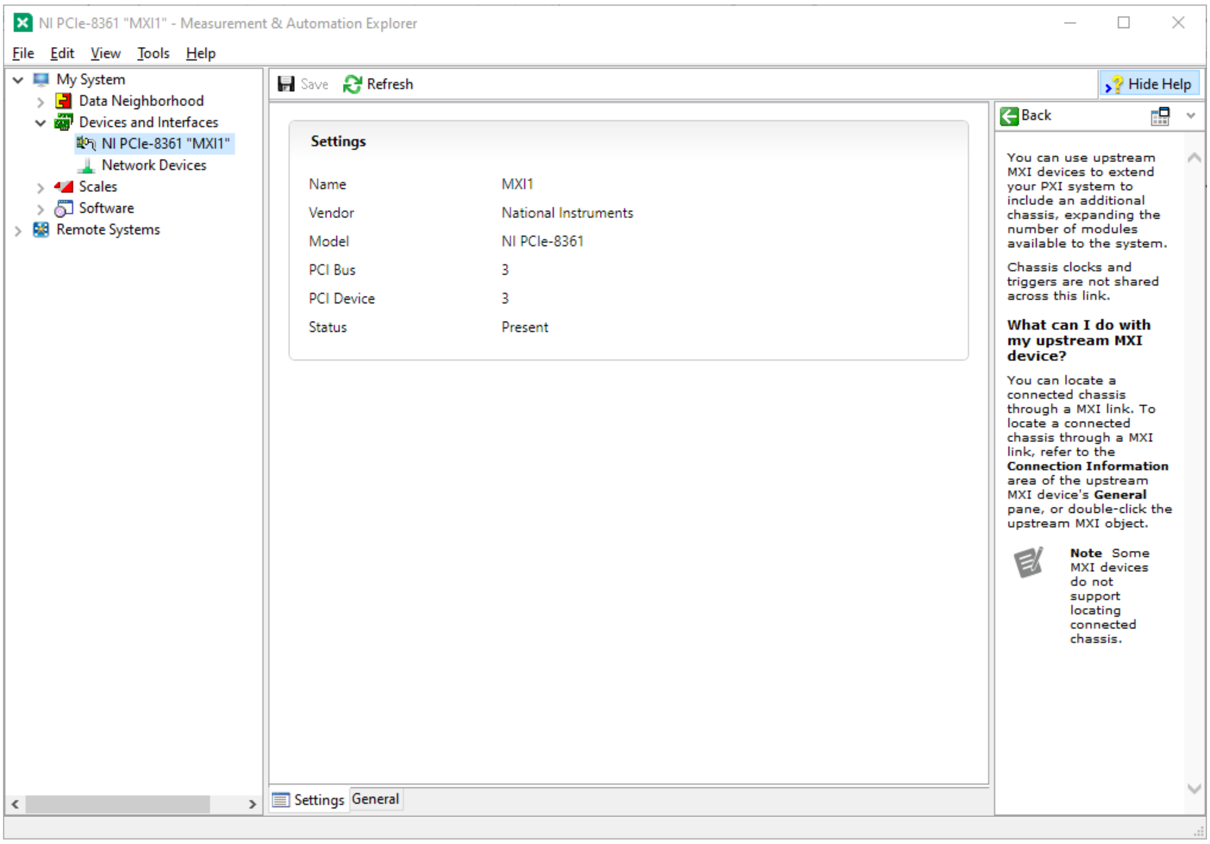
NI Measurement & Automation Explorer (MAX) provides access to the NI hardware (i.e. the NI USB-6363 in the case of the PM1 Pachymeter Automated System Level Test Platform).

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.



Settings

Name	MXI1
Vendor	National Instruments
Model	NI PCIe-8361
PCI Bus	3
PCI Device	3
Status	Present

You can use upstream MXI devices to extend your PXI system to include an additional chassis, expanding the number of modules available to the system.

Chassis clocks and triggers are not shared across this link.

What can I do with my upstream MXI device?

You can locate a connected chassis through a MXI link. To locate a connected chassis through a MXI link, refer to the **Connection Information** area of the upstream MXI device's **General** pane, or double-click the upstream MXI object.

Note Some MXI devices do not support locating connected chassis.

3.2.4 NI-DAQmx Python API

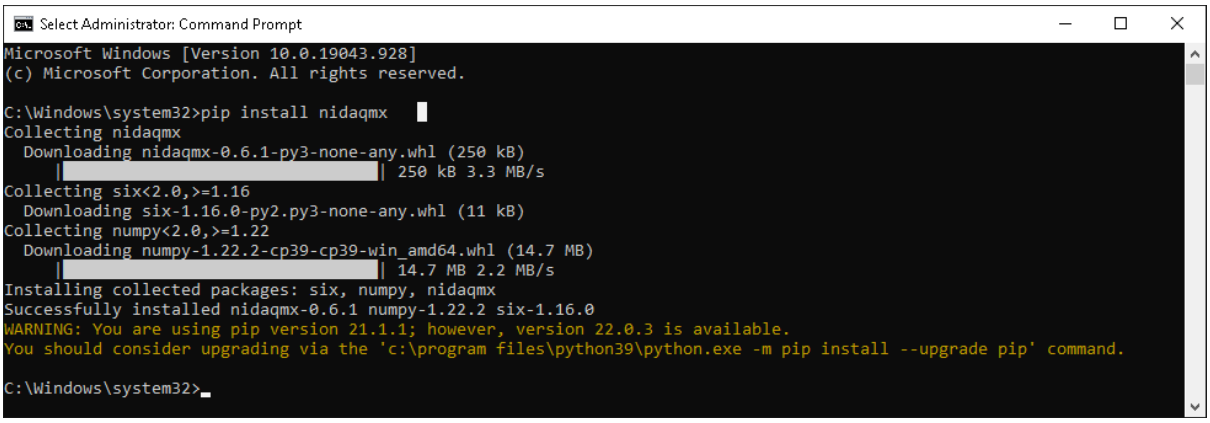
The Python package nidaqmx contains an API (Application Programming Interface) for interacting with the NI-DAQmx driver installed previously. This package was created and is supported by NI. The package is implemented as a complex, highly object-orientated wrapper around the NI-DAQmx C API using the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

3.2.4.1 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:



```
Microsoft Windows [Version 10.0.19043.928]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>pip install nidaqmx
Collecting nidaqmx
  Downloading nidaqmx-0.6.1-py3-none-any.whl (250 kB)
     |████████████████████████████████| 250 kB 3.3 MB/s
Collecting six<2.0,>=1.16
  Downloading six-1.16.0-py2.py3-none-any.whl (11 kB)
Collecting numpy<2.0,>=1.22
  Downloading numpy-1.22.2-cp39-cp39-win_amd64.whl (14.7 MB)
     |████████████████████████████████| 14.7 MB 2.2 MB/s
Installing collected packages: six, numpy, nidaqmx
Successfully installed nidaqmx-0.6.1 numpy-1.22.2 six-1.16.0
WARNING: You are using pip version 21.1.1; however, version 22.0.3 is available.
You should consider upgrading via the 'c:\program files\python39\python.exe -m pip install --upgrade pip' command.

C:\Windows\system32>
```

3.2.5 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.

3.2.5.1 DOWNLOAD

Zadig can be downloaded from the following link:

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

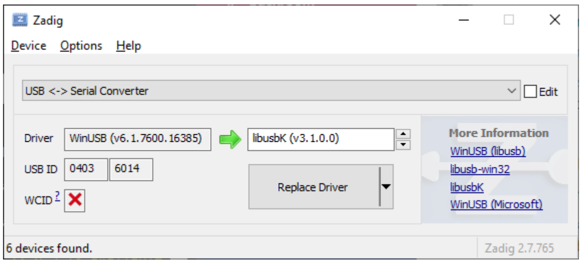
3.2.5.2 INSTALLATION

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:



USB <-> Serial Converter ☐ Edit

Driver WinUSB (v6.1.7600.16385) → libusbK (v3.1.0.0)

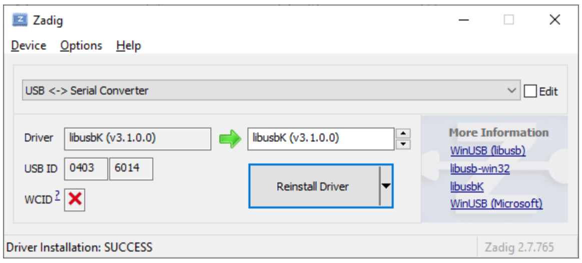
USB ID 0403 6014

WCID ☒

6 devices found.

Zadig 2.7.765

- Click on ‘Replace Driver’.
- Once successfully installed Zadig should update as follows:



USB <-> Serial Converter ☐ Edit

Driver libusbK (v3.1.0.0) → libusbK (v3.1.0.0)

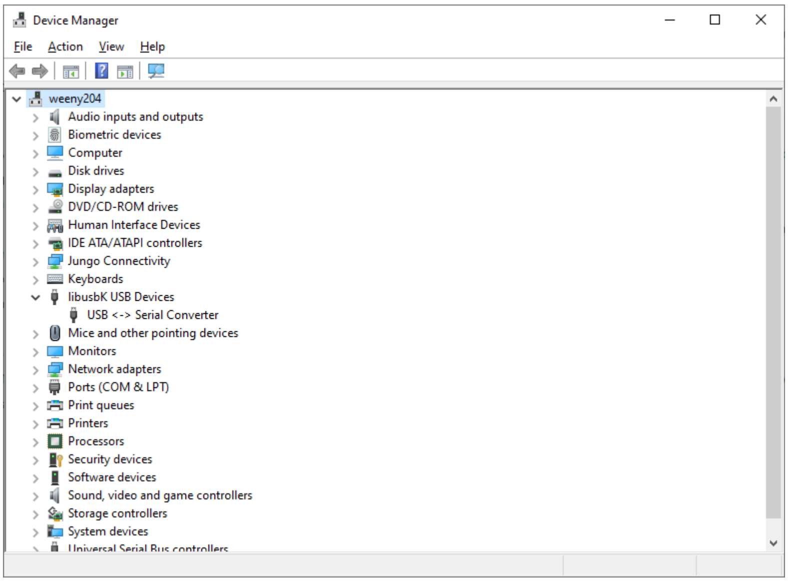
USB ID 0403 6014

WCID ☒

Driver Installation: SUCCESS

Zadig 2.7.765

and device manager should show the USB <-> Serial Converter under the libusbK USB Devices node:

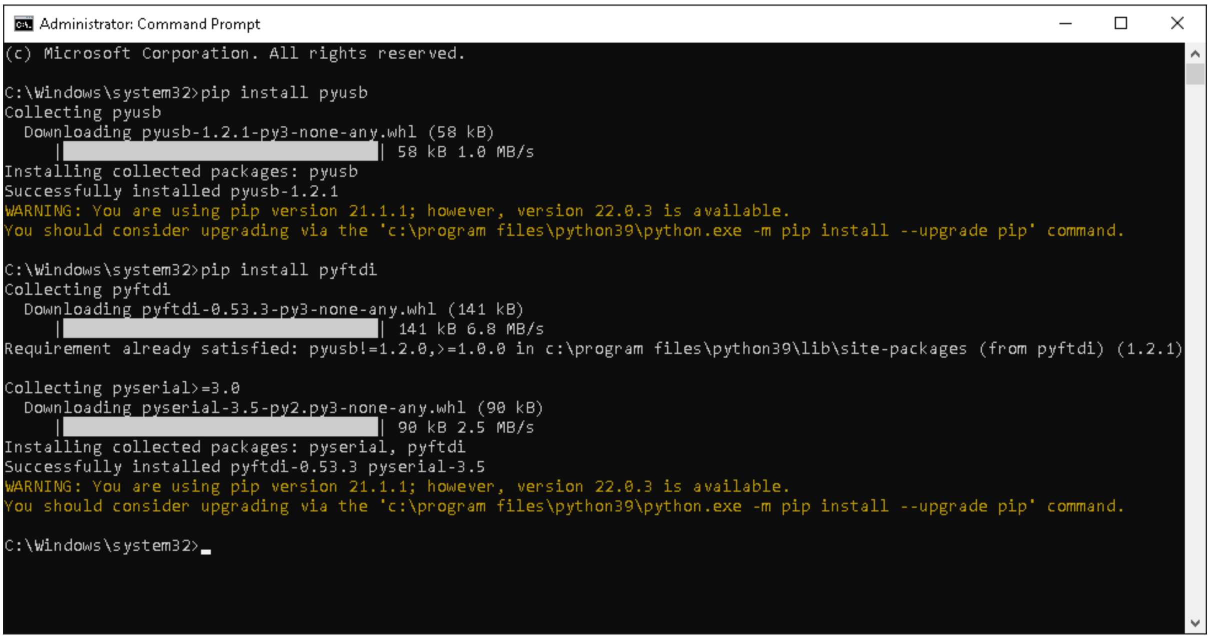


weeny204
- Audio inputs and outputs
- Biometric devices
- Computer
- Disk drives
- Display adapters
- DVD/CD-ROM drives
- Human Interface Devices
- IDE ATA/ATAPI controllers
- Jungo Connectivity
- Keyboards
- libusbK USB Devices
  - USB <-> Serial Converter
- Mice and other pointing devices
- Monitors
- Network adapters
- Ports (COM & LPT)
- Print queues
- Printers
- Processors
- Security devices
- Software devices
- Sound, video and game controllers
- Storage controllers
- System devices
- Universal Serial Bus controllers

3.2.6 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’:

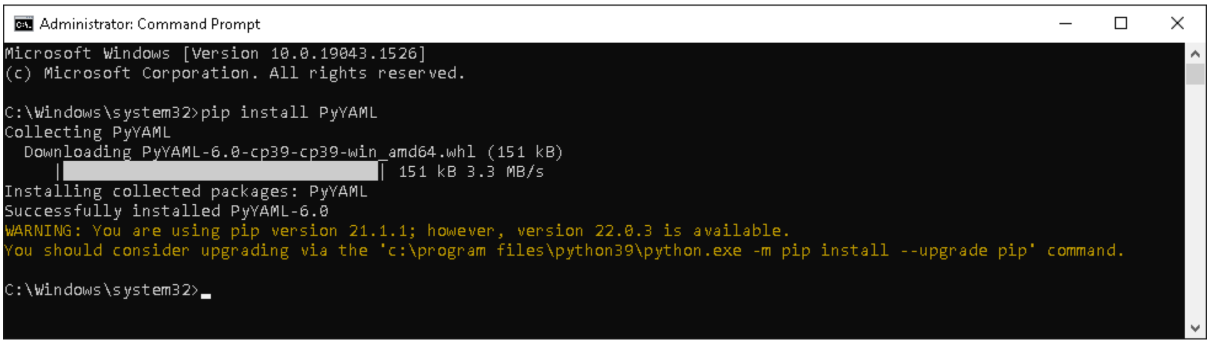


```
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>pip install pyusb
Collecting pyusb
  Downloading pyusb-1.2.1-py3-none-any.whl (58 kB)
     |████████████████████████████████| 58 kB 1.0 MB/s
Installing collected packages: pyusb
Successfully installed pyusb-1.2.1
WARNING: You are using pip version 21.1.1; however, version 22.0.3 is available.
You should consider upgrading via the 'c:\program files\python39\python.exe -m pip install --upgrade pip' command.

C:\Windows\system32>pip install pyftdi
Collecting pyftdi
  Downloading pyftdi-0.53.3-py3-none-any.whl (141 kB)
     |████████████████████████████████| 141 kB 6.8 MB/s
Requirement already satisfied: pyusb!=1.2.0,>=1.0.0 in c:\program files\python39\lib\site-packages (from pyftdi) (1.2.1)

Collecting pyserial>=3.0
  Downloading pyserial-3.5-py2.py3-none-any.whl (90 kB)
     |████████████████████████████████| 90 kB 2.5 MB/s
Installing collected packages: pyserial, pyftdi
Successfully installed pyftdi-0.53.3 pyserial-3.5
WARNING: You are using pip version 21.1.1; however, version 22.0.3 is available.
You should consider upgrading via the 'c:\program files\python39\python.exe -m pip install --upgrade pip' command.

C:\Windows\system32>_
```

3.2.7 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’:

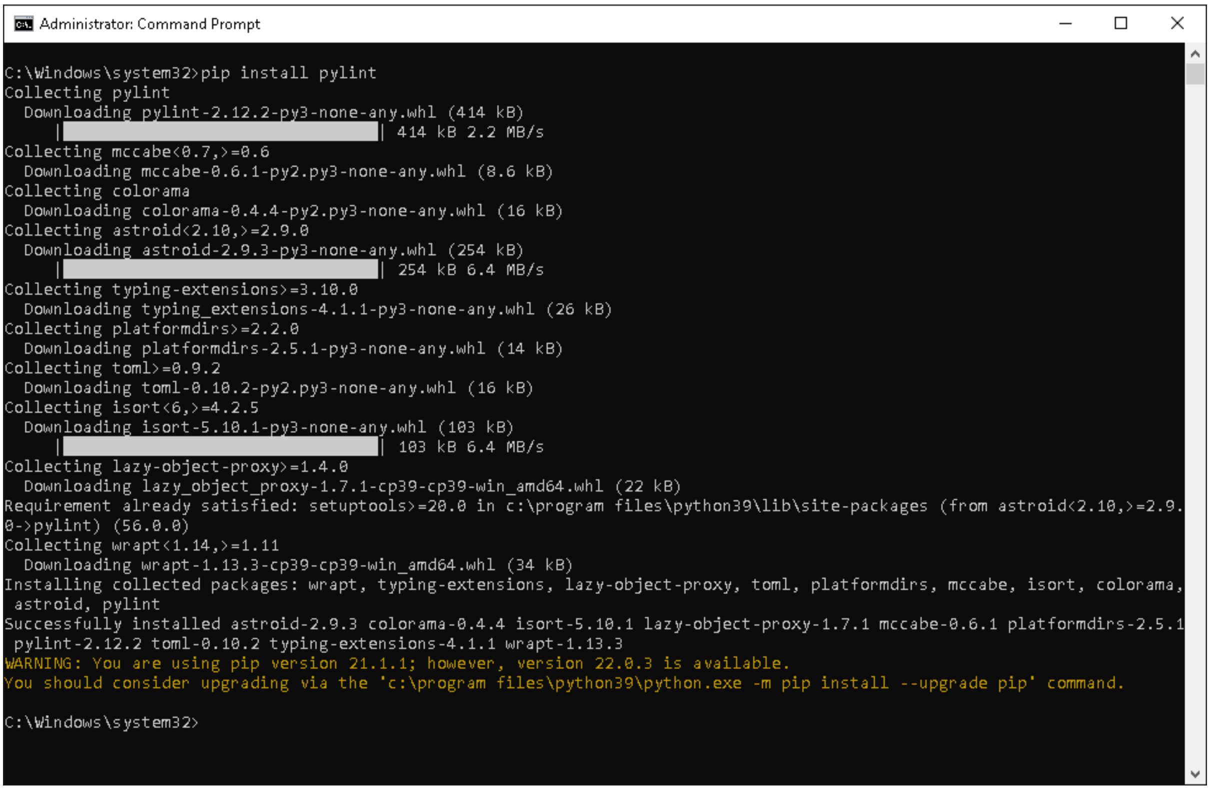


```
Microsoft Windows [Version 10.0.19043.1526]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>pip install PyYAML
Collecting PyYAML
  Downloading PyYAML-6.0-cp39-cp39-win_amd64.whl (151 kB)
     |████████████████████████████████| 151 kB 3.3 MB/s
Installing collected packages: PyYAML
Successfully installed PyYAML-6.0
WARNING: You are using pip version 21.1.1; however, version 22.0.3 is available.
You should consider upgrading via the 'c:\program files\python39\python.exe -m pip install --upgrade pip' command.

C:\Windows\system32>_
```

3.2.8 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’:



```
C:\Windows\system32>pip install pylint
Collecting pylint
  Downloading pylint-2.12.2-py3-none-any.whl (414 kB)
     |████████████████████████████████| 414 kB 2.2 MB/s
Collecting mccabe<0.7,>=0.6
  Downloading mccabe-0.6.1-py2.py3-none-any.whl (8.6 kB)
Collecting colorama
  Downloading colorama-0.4.4-py2.py3-none-any.whl (16 kB)
Collecting astroid<2.10,>=2.9.0
  Downloading astroid-2.9.3-py3-none-any.whl (254 kB)
     |████████████████████████████████| 254 kB 6.4 MB/s
Collecting typing-extensions>=3.10.0
  Downloading typing_extensions-4.1.1-py3-none-any.whl (26 kB)
Collecting platformdirs>=2.2.0
  Downloading platformdirs-2.5.1-py3-none-any.whl (14 kB)
Collecting toml>=0.9.2
  Downloading toml-0.10.2-py2.py3-none-any.whl (16 kB)
Collecting isort<6,>=4.2.5
  Downloading isort-5.10.1-py3-none-any.whl (103 kB)
     |████████████████████████████████| 103 kB 6.4 MB/s
Collecting lazy-object-proxy>=1.4.0
  Downloading lazy_object_proxy-1.7.1-cp39-cp39-win_amd64.whl (22 kB)
Requirement already satisfied: setuptools>=20.0 in c:\program files\python39\lib\site-packages (from astroid<2.10,>=2.9.
0->pylint) (56.0.0)
Collecting wrapt<1.14,>=1.11
  Downloading wrapt-1.13.3-cp39-cp39-win_amd64.whl (34 kB)
Installing collected packages: wrapt, typing-extensions, lazy-object-proxy, toml, platformdirs, mccabe, isort, colorama,
 astroid, pylint
Successfully installed astroid-2.9.3 colorama-0.4.4 isort-5.10.1 lazy-object-proxy-1.7.1 mccabe-0.6.1 platformdirs-2.5.1
 pylint-2.12.2 toml-0.10.2 typing-extensions-4.1.1 wrapt-1.13.3
WARNING: You are using pip version 21.1.1; however, version 22.0.3 is available.
You should consider upgrading via the 'c:\program files\python39\python.exe -m pip install --upgrade pip' command.

C:\Windows\system32>
```

3.2.9 Visual Studio Code (IDE)

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

3.2.9.1 Download

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

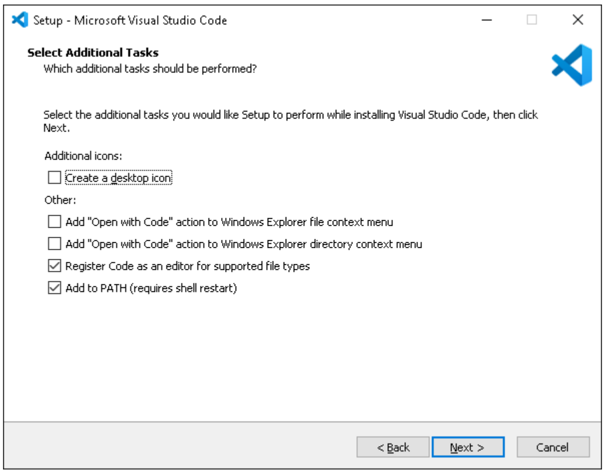
https://code.visualstudio.com/sha/download?build=stable&os=win32-x64

3.2.9.2 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:



Select Additional Tasks
Which additional tasks should be performed?

Select the additional tasks you would like Setup to perform while installing Visual Studio Code, then click Next.

Additional icons:

☐ Create a desktop icon

Other:

☐ Add "Open with Code" action to Windows Explorer file context menu
☐ Add "Open with Code" action to Windows Explorer directory context menu
☑ Register Code as an editor for supported file types
☑ Add to PATH (requires shell restart)

and click ‘Next’.

- Click ‘Install’.
- Once setup has finished untick the ‘Launch Visual Studio Code’ button and click ‘Finish’.

3.2.10 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).

3.2.10.1 Download

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

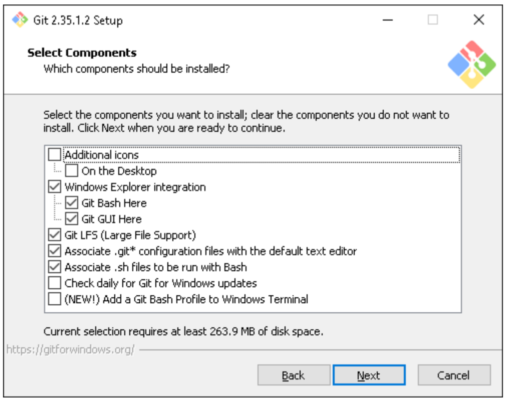
https://github.com/git-for-windows/git/releases/download/v2.35.1.windows.2/Git-2.35.1.2-64-bit.exe

3.2.10.2 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:



Select Components
Which components should be installed?

Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.

- ☐ Additional icons
  - ☐ On the Desktop
- ☑ Windows Explorer integration
  - ☑ Git Bash Here
  - ☑ Git GUI Here
- ☑ Git LFS (Large File Support)
- ☑ Associate .git* configuration files with the default text editor
- ☑ Associate .sh files to be run with Bash
- ☐ Check daily for Git for Windows updates
- ☐ (NEW!) Add a Git Bash Profile to Windows Terminal

Current selection requires at least 263.9 MB of disk space.

https://gitforwindows.org/

  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’

3.3 PM1 Pachymeter Automated System Level Test Framework

3.3.1 Overview

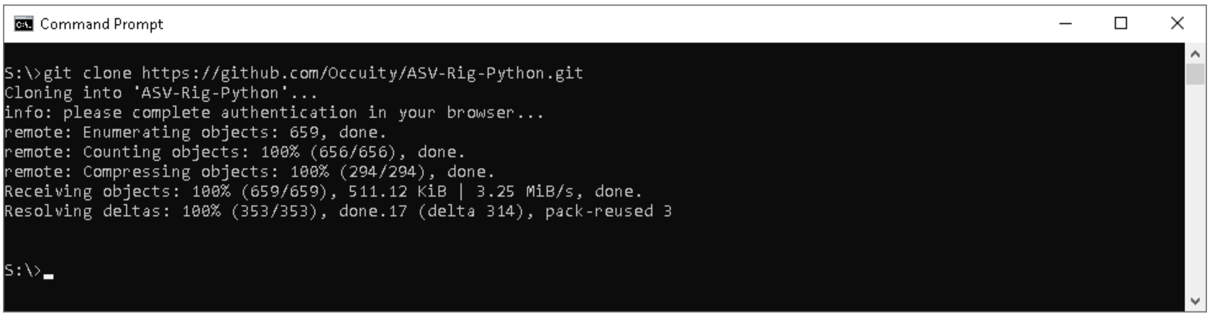
The PM1 Pachymeter Automated System Level Test Platform Framework components are summarised in the following diagram:

[INSERT DIAGRAM HERE]

3.3.2 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:



```
S:\>git clone https://github.com/Occuity/ASV-Rig-Python.git
Cloning into 'ASV-Rig-Python'...
info: please complete authentication in your browser...
remote: Enumerating objects: 659, done.
remote: Counting objects: 100% (656/656), done.
remote: Compressing objects: 100% (294/294), done.
Receiving objects: 100% (659/659), 511.12 KiB | 3.25 MiB/s, done.
Resolving deltas: 100% (353/353), done.17 (delta 314), pack-reused 3


S:\>
```

4 SOFTWARE CONFIGURATION

4.1 VISUAL STUDIO CODE

4.1.1 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.

4.1.1.1 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.

4.1.1.2 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.

4.1.1.3 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.

5 PM1 PACHYMETER AUTOMATED SYSTEM LEVEL TEST FRAMEWORK

5.1 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.