

Programmer's Guide

Version 24.16.0 June, 2022



MIL-STD-1553

Python Based Application Programming Interface

Programmer's Guide

Version 24.16.0 June, 2022

AIM NO. 60-11945-37-24.16.0



AIM – Gesellschaft für angewandte Informatik und Mikroelektronik mbH

AIM GmbH

Sasbacher Str. 2 D-79111 Freiburg / Germany Phone +49 (0)761 4 52 29-0 Fax +49 (0)761 4 52 29-33 sales@aim-online.com

AIM UK Office

Cressex Enterprise Centre, Lincoln Rd. High Wycombe, Bucks. HP12 3RB / UK Phone +44 (0)1494-446844 Fax +44 (0)1494-449324 salesuk@aim-online.com

AIM GmbH - Munich Sales Office

Terofalstr. 23a D-80689 München / Germany Phone +49 (0)89 70 92 92-92 Fax +49 (0)89 70 92 92-94 salesgermany@aim-online.com

AIM USA LLC

Seven Neshaminy Interplex Suite 211 Trevose, PA 19053 Phone 267-982-2600 Fax 215-645-1580 salesusa@aim-online.com

© AIM GmbH 2022

Notice: The information that is provided in this document is believed to be accurate. No responsibility is assumed by AIM GmbH for its use. No license or rights are granted by implication in connection therewith. Specifications are subject to change without notice.



TABLE OF CONTENTS

1	Introduction	1
2	Low-Level API Access Using ctypes Bindings	2
List	of Abbreviations	ı



IV



1 Introduction

This Programmer's Guide has been developed to guide users that want to control their AIM mil bus devices using custom applications written in Python scripting language. This can be done through low level Application Programming Interface (API) access using ctypes.

The interface is provided with a special Python package called *aim_mil*. At this time, the package supports Python 2.7 as well as Python 3.x.

Note:

Make sure the package aim_mil is found by your Python interpreter if you want to import it in your application. On some Operating System (OS) you have to manually add the path to the package to your *PYTHONPATH* environment variable.

AIM is also a leading designer and manufacturer of other high performance test and simulation modules, data bus analyzer software and systems for MIL-STD-1553 A/B, AFDX/ARINC664, ARINC429, MIL-STD-1760 and CAN/ARINC825 Applications, PANAVIA Serial Link and Fibre Channel. Supported hardware platforms include Peripheral Component Interconnect (PCI), PCI Express (PCIe), Compact PCI (cPCI), Versa Module Eurocard (VME), VME eXtensions for Instrumentation (VXI), PC104+, PC-Card, PCI Mezzanine Card (PMC), Express Card and Universal Serial Bus (USB). Information about all AIM products can be found at http://www.aim-online.com.



2 Low-Level API Access Using ctypes Bindings

At the core, the interface between Python scripts and AIM's mil bus C\C++ based API library is implemented via the *ctypes* library of Python. The mandatory interface for Python scripts is the AIM API Python bindings module *aim_mil.mil_bindings.py*. This one is binding C API definitions like data types, constants, defines and structures to ctypes objects. This allows to call the C library functions directly from a Python script. For the general programming the standard *Reference Manual* applies.

See below the calls necessary for initializing the ctypes bindings:

```
# ctypes is a foreign function library for Python. It provides C
# compatible data types, and allows calling functions in C libraries.
# It can be used to wrap these libraries in pure Python.
import ctypes
import sys

# import from aim_mil API bindings all
# AIM mil bus API C-Structures as Python Classes
from aim_mil.mil_bindings import *

# Load the AIM mil bus C library via ctypes.

if sys.platform.startswith("linux"):
        c_library = ctypes.cdll.LoadLibrary("libaim_mil.so")
else:
        c_library = ctypes.cdll.LoadLibrary("api_mil.24.dll")
```

The *c_library* object can now be used to issue API function calls via Python ctypes.

Refer to the Python ctypes library documentation https://docs.python.org/2.7/library/ctypes.html and the sample Python scripts for a full overview of the ctypes functionality.



List of Abbreviations

API Application Programming Interface

cPCI Compact PCI

OS Operating System

PCI Peripheral Component Interconnect

PCIe PCI Express

PMC PCI Mezzanine Card

USB Universal Serial Bus

VME Versa Module Eurocard

VXI VME eXtensions for Instrumentation