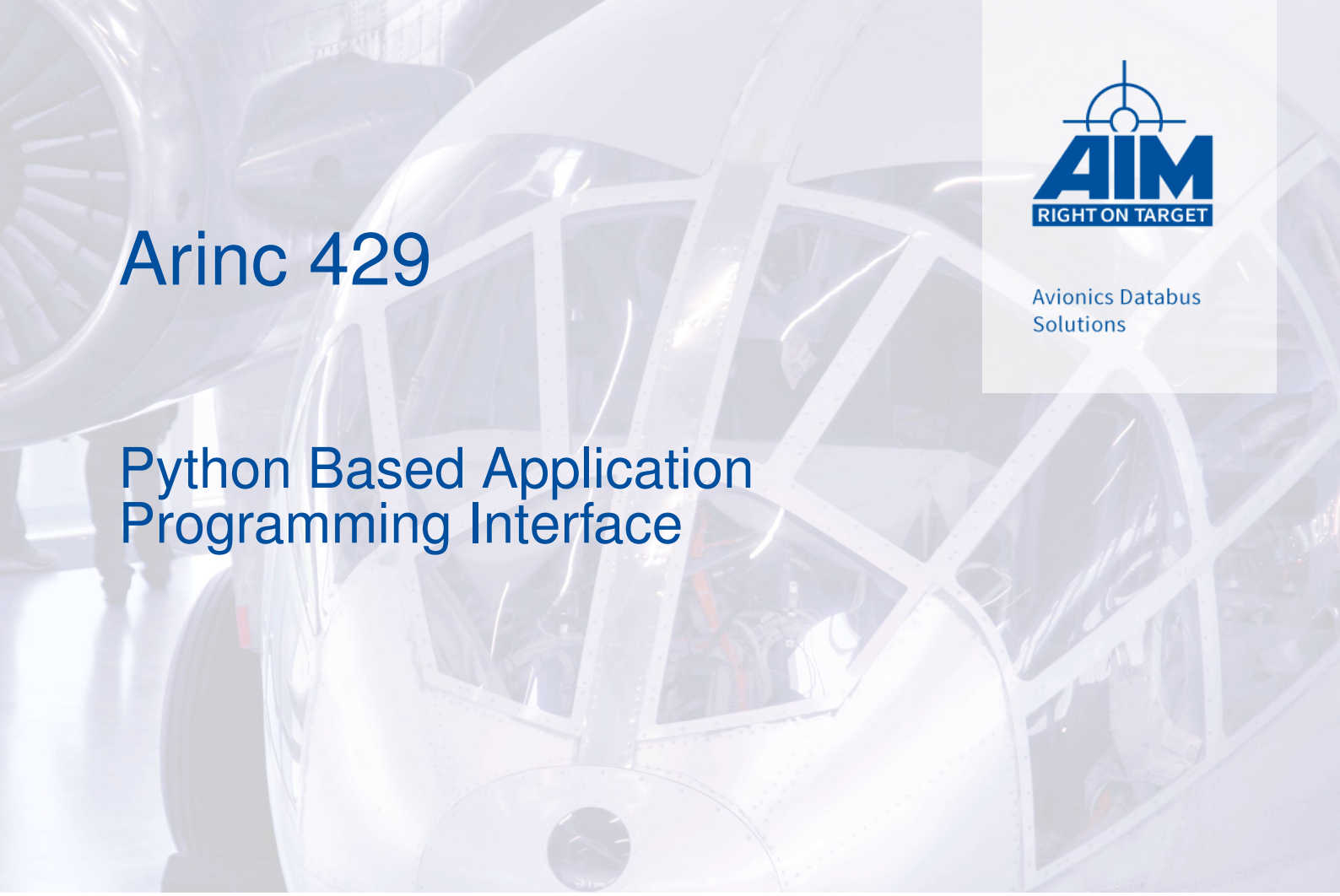

Arinc 429

Avionics Databus Solutions

Python Based Application Programming Interface

Programmer's Guide

Version 9.8.0
June, 2022

www.aim-online.com

Arinc 429

Python Based Application Programming Interface

Programmer's Guide

Version 9.8.0
June, 2022

AIM NO. 60-12945-37-9.8.0

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

TABLE OF CONTENTS

1 Introduction

This Programmer's Guide has been developed to guide users that want to control their AIM Arinc 429 devices using custom applications written in Python scripting language. Two different ways to do this are supported and are covered in this document:

- Low-level C Application Programming Interface (API) access using ctypes
- Programming using object-oriented C API abstraction

All of these interfaces are provided with a special Python package called *aim_429*. At this time, the package supports Python 2.7. Support for Python 3.x is not available yet.

Note:
Make sure the package aim_429 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), 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 Arinc 429 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_429.apibindings.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 API documentation applies, especially the *Programmer's Guide* and *Reference Manual*.

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_429 API bindings all
# AIM 429 API C-Structures as Python Classes
from aim_429.apibindings import *


# Load the AIM 429 C library via ctypes.

if sys.platform.startswith("linux"):
    c_library = ctypes.cdll.LoadLibrary("libaim_civ.so")
else:
    c_library = ctypes.cdll.LoadLibrary("api_civ.9.dll")
```

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

See following code snippet for the *Api429BoardInfoGet* command from a Python script as an basic example for calling 429 API library functions.

```
    # Tell ctypes that Api429LibErrorDescGet returns a string
    c_library.Api429LibErrorDescGet.restype = ctypes.c_char_p

    board_info = TY_API429_BOARD_INFO()

    res = c_library.Api429BoardInfoGet(0, ctypes.byref(board_info))
    if res != API_OK:
```

```
        raise Exception("Error: Api429BoardInfoGet: %s."
            % c_library.Api429LibErrorDescGet(res))
```

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.

3 The Object-Oriented Python API Library

The *aim_429* Python package encapsulates the low-level API access in an object-oriented approach offering a set of Python classes. It also contains sample scripts that explain some basic usage of the library and its capabilities for controlling your Arinc 429 devices.

Note:
The object-oriented API is not considered stable and can be subject to changes that may break applications which make use of it. Hence the package is not documented here. Refer to the documentation in the package's source code for more information.

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