# Air Force Research Laboratory
# Space Vehicles Directorate

# Cold-Atom Physics Laboratory
# Software, Firmware, & Hardware

–Public Release Summary–

**Version: release-1.0-4-ga1048f0**

This Document Last Revised 2018-03-20

Dr. Spencer E. Olson, AFRL/RVBYC
Dr. Brian L. Kasch, AFRL/RVBYC
Dr. Matthew B. Squires, AFRL/RVBYC
Dr. James A. Stickney, Space Dynamics Laboratory
Dr. Benjamin K. Stuhl, Space Dynamics Laboratory
Dr. Rudolph N. Kohn, Space Dynamics Laboratory

# PREFACE

This report documents the public release process and history for the software belonging to the Air Force Research Laboratory (AFRL) Cold-Atom physics family of laboratory software. This software is developed by the AFRL Cold-Atom Laboratory. Contributions to this software are made as a collaboration between government personnel, for which copyright-law does not apply, and contractor personnel with whose respective employers this software-release is coordinated according to contractual rights and obligations between the government and the contractor.

This material released to the public, as documented here, has been reviewed and is deemed to (1) be publicly useful, (2) be pertinant to basic-research efforts at the academic or national-laboratory level, (3) not to consist of or contain **any** critical military technology, (4) not to involve or release any International Traffic in Arms Regulations (ITAR) or other similarly restricted information with respect to the USA, and (5) anticipate public contributions (bug fixes and other improvements) from which AFRL and the basic-research community can benefit.

# TABLE OF CONTENTS

# LIST OF TABLES

**Table**

# CHAPTER I

# Overview

## 1.1 Purpose

This report documents the public release process and history for the tools belonging to the Air Force Research Laboratory (AFRL) Cold-Atom physics family of laboratory software, firmware, & hardware. These tools are developed by the AFRL Cold-Atom Laboratory. Contributions to this software are made as a collaboration between (1) US government personnel, for which copyright-law does not apply, (2) contractor personnel with whose respective employers this software-release is coordinated according to contractual rights and obligations between the government and the contractor, and (3) academia, and/or other non-governmental collaborators.

## 1.2 Benefit to AFRL/RV

To paraphrase Ref. [1], the government runs various risks when it develops new software but does not release it as Open Source Software (OSS). These risks include

- Greatly increased cost, due to having to bear the entire burden of development costs
- Inability to use improvements (including security patches and innovations) by others, since they do not have the opportunity to aid in its development
- The development and release of a competing project (either commercial or OSS). In this case, the government has the unenviable choice of (1) spending possibly large sums to switch to the OSS project (which would typically have a radically different interface and goals), or (2) continuing to use the government-unique custom solution, leaving the U.S. systems far less capable than otherwise possible
- Questions about why the government—who represents "the people"—is not releasing software that they paid for back to "the people"

It is anticipated that this family of Cold-Atom Physics tools will benefit from the use, testing, and contribution from others that are external to the government.

## 1.3 Scope

This summary documents the publicly released software, firmware, & hardware components of the family of Cold-Atom Physics Laboratory tools. These tools have been reviewed are deemed to (1) be publicly useful, (2) be pertinent to basic-research efforts at the academic or national-laboratory level, (3) not to consist of or contain **any** critical military technology, (4) not to involve or release any International Traffic in Arms Regulations (ITAR) or other similarly restricted information with respect to the USA, and (5) anticipate public contributions (bug fixes and other improvements) from which AFRL and the basic-research community can benefit.

## 1.4 Release Mechanisms

In order to best contribute to the cold-atom physics community, the family of Cold-Atom Physics Laboratory Software, Firmware, & Hardware will be released using OSS licenses. Following guidance from DoD CIO, these licenses are be chosen for each specific project to maximize the benefit to both the community while also ensuring maximum re-entry of fixes and improvements into the AFRL/RVB code. Furthermore, licenses will be used where appropriate in order to satisfy copyright and/or other legal constraints.

The items released within the family of Cold-Atom Physics Laboratory Software, Firmware, & Hardware are released using the Git[1] repository hosting service known as `GitHub.com`. This information is released routinely under the `GitHub.com` organization named afrl-quantum.

[1] Git is a revision control system invented for the Linux kernel and widely used in the Open Source world.

# CHAPTER II

# Packages

This chapter identifies and describes the various components of the family of Cold-Atom Physics Laboratory Software, Firmware, & Hardware that are released together as reported here. For each package, the following items are identified

- **Background:** Provides a very brief description of the reason why this component was developed.
- **Overview:** Provides a brief description of the function of this component.
- **License:** Identifies the conditions or license under which this component is released.
- **Description of Released Contents:** Any information that describes the content or layout of content that is to be released with this component.
- **Major Release Versions:** A table of major revisions that are released through the process documented here. Note that major revisions are defined as revisions that bring new capabilities and facilities to the function of the component. This table records the component version number, revision date, date of release, and a brief summary of capabilities changes included with the particular revision.

## 2.1 Cold-Atom Physics Laboratory Package: Arbwave (as of 2017-12-11)

### 2.1.1 Background

Atomic physics experiments typically involve various voltages, currents, magnetic fields, laser fields, and mechanical devices that must be manipulated and altered according to very precise relative timing relationships. It is necessary to use hardware and software that provides for coordinating these tight timing relationships. The extent to which software and hardware are able to present these timing relationships to a user in a succinct and consistent fashion directly determines how well a configuration can be tailored to the specific experimental situation.

### 2.1.2 Overview

Arbwave provides a means for a user to clearly coordinate groups of transitions for various control signals that can be used to run a complicated experimental timing sequence. In Arbwave, a series of channels are defined where each channel represents some control signal used to supervise an experimental parameter such as voltage, current, magnetic field, etc. Using the information of the defined channels, Arbwave allows one to coordinate heirarchical sets of transitions for each of those channels. Arbwave also graphically represents each of the signal transitions as complete waveforms where the time for each transition is accurately shown. Furthermore, Arbwave facilitates execution of these waveforms in various operation patterns, such as control-variable nested loop iteration and multi-variable optimization. Finally, user scripting, via Python, can be inserted into the execution to customize a particular experimental procedure, customize an optimization procedure, and hook into various external data recorders processors.

### 2.1.3 License

GNU GENERAL PUBLIC LICENSE; Version 3, 29 June 2007

### 2.1.4 Description of Released Contents

The contents of this released software package include

1. The Python source code that implements the bulk of the Arbwave functionality
2. run.py: as script that should be used to launch Arbwave with various commandline options
3. The Arbwave manual in LaTeX form and also as a compiled PDF
4. The entire history of changes made for this package since Arbwave came into being in 2012.

### 2.1.5 Major Release Versions

Tab. 2.1 of this section documents the major changes of Arbwave that warrant a new full release based on a major change in capabilities.

| Version | Revision Date | Release Date | Summary of Contributions/Changes |
|---|---|---|---|
| 1.0.0 | 11 Dec 2017 | 20 Dec 2017 | Initial Release |

Table 2.1: Released major versions of Arbwave

## 2.2 Cold-Atom Physics Laboratory Package: CHIMP (as of 2017-06-05)

### 2.2.1 Background

Currently, it is very common for different simulations to use slightly, if not drastically, different physical data for the same materials. This creates both confusion as to how to compare the results of different simulations, but also questions the validity of any one simulation.

### 2.2.2 Overview

The Chemical Interactions, Materials, and Particles Database and Simulation Framework (CHIMP) was developed to meet the need in the computational physics community for a common repository of physical data and a means to deliver that data to simulation tools. This work represents a collaborative effort to develop a database and library to provide physical data and associated model calculations in a consistent, simple yet flexible manner.

### 2.2.3 License

GNU LESSER GENERAL PUBLIC LICENSE; Version 3, 29 June 2007

### 2.2.4 Description of Released Contents

The contents of this released software package include

1. c++ implementation for accessing/using CHIMP databaes (released via source code).
2. Various c++ and Python examples for accessing/using CHIMP database.
3. Initial Python implementation for accessing/using CHIMP data.
4. Build/install files implemented as Boost.Build configuration files.
5. Documentation of the c++ interface
6. The entire history of changes made for this package

### 2.2.5 Major Release Versions

Tab. 2.2 of this section documents the major changes of CHIMP that warrant a new full release based on a major change in capabilities.

| Version | Revision Date | Release Date | Summary of Contributions/Changes |
|---|---|---|---|
| 0.1.3 | 30 Mar 2011 | 4 Apr 2011 | Last release by AFRL/RD |
| 0.1.4 | 20 Dec 2016 | 20 Dec 2017 | • Adds arbitrary collision expressions<br>• Adds Inverse and Log cross section models<br>• Initial code for three body interactions<br>• Initial code for inelastic collisions<br>• Add threshold energies to Lotz, Constant, and VHS X-section models<br>• Add initial python interface to CHIMP database |

Table 2.2: Released major versions of CHIMP

## 2.3 Cold-Atom Physics Laboratory Package: Digital Signal Distribution (as of 2017-10-17)

### 2.3.1 Background

Experimental work often involves digital signals for various devices in order to control the function, timing, and state of the output for these various devices. Additionally, laboratory environments typically contain various electromagnetic interference (EMI) sources, including the digital signals themselves.

### 2.3.2 Overview

AFRL's digital signal distribution hardware mitigates some of the challenges of EMI that may originate either from digital signals or corrupt digital signal transmission. The digital signal distribution hardware supports the low-noise transport of asynchronous timing signals through a laboratory-grade hostile EMI environment by converting single-ended TTL signals to LVDS for transmission over standard Cat-5e/Cat-6 twisted pair (Ethernet) cables to 4-channel receiver boxes which return the signals to 3.3 or 5 V (LV)TTL signals suitable for interfacing with rack or bench instrumentation. The receiver boxes are locally powered and decoupled from the LVDS ground potential to minimize the creation of ground loops compared to direct wiring of TTL signals from the source instrument. Inputs to the transmitter board are connected through a 68-pin VHDCI connector.

### 2.3.3 License

The hardware schematics are made available under the CERN Open Hardware Licence version 1.2 or later.

### 2.3.4 Description of Released Contents

The contents of this released software package include

1. Design files as used/defined by the electronics design automation software *KiCad EDA*
2. PDF generated representations of the design schematics
3. Excel spreadsheet of parts-list
4. Gerber generated representations of the design schematics and useful for sending to a PCB fabrication facility for construction
5. The entire history of changes made for the receiver and transmitter designs

### 2.3.5 Major Release Versions

Tab. 2.3 of this section documents the major changes of Digital Signal Distribution that warrant a new full release based on a major change in capabilities.

| Version | Revision Date | Release Date | Summary of Contributions/Changes |
|---|---|---|---|
| 1.1 | 9 Jan 2017 | 20 Dec 2017 | Initial Release |

Table 2.3: Released major versions of Digital Signal Distribution

## 2.4 Cold-Atom Physics Laboratory Package: Gridless Direct Simulation Monte Carlo (as of 2017-06-05)

### 2.4.1 Background

Physical systems are typically simulated in a manner where time and space are both discretized. This discretization has two major categories of side-effects: simulation fidelity and computational load. The fidelity of a simulation is affected in so far as the size of a spatial/temporal grid dictates the size/rate of physical processes that can be fully captured in a particular simulation, e.g., a spatial grid of size $\delta x$ cannot be used to simulate a physical process with features much smaller than $\delta x$. The level of discretization also has strong implications on requirements for computational resources. For example, the memory footprint of a fine-resolution grid for a large system can be quite extreme.

### 2.4.2 Overview

This library represents the development of a gridless algorithm for modeling inter-particle interactions. In other words, this library provides a technique by which inter-particle interactions can be simulated without requiring a fixed grid in order to truncate the collision integral of the Boltzmann equation. By using a gridless algorithm, it is possible to mitigate the effects of a discretized simulation, such that computational requirements are lessened, while at the same time a desired level of fidelity is reached throughout the entire simulation domain. Furthermore, a gridless algorithm can be used to simulate various physical systems without the need to perform time-consuming grid-mesh optimization [3–5].

### 2.4.3 License

GNU LESSER GENERAL PUBLIC LICENSE; Version 3, 29 June 2007

### 2.4.4 Description of Released Contents

The contents of this released software package include

1. c++ implementation of Gridless and Gridded DSMC algorithm.
2. Various c++ examples for using DSMC library for simulations.
3. Build/install files implemented as Boost.Build configuration files.
4. Documentation of the c++ interface.
5. The entire history of changes made for this package.

### 2.4.5 Major Release Versions

Tab. 2.4 of this section documents the major changes of Gridless Direct Simulation Monte Carlo that warrant a new full release based on a major change in capabilities.

| Version | Revision Date | Release Date | Summary of Contributions/Changes |
|---|---|---|---|
| 0.1.0 | 12 June 2016 | 20 Dec 2017 | • Many interfaces re-written as c++ templates for better optimization<br>• Multi-species simulation<br>• gridless function without tracking global macroscopic properties<br>• fast octree sorting algorithms. |

Table 2.4: Released major versions of Gridless Direct Simulation Monte Carlo

## 2.5 Cold-Atom Physics Laboratory Package: Fields (as of 2017-08-28)

### 2.5.1 Background

Simulations for atomic physics typically includes the calculation of various magnetic, electric, optical, and gravitational fields and forces. A generic representation of fields and forces allows simulations to be written in an abstracted fashion, such that code-reuse is possible and separation of physics from simulation is possible.

### 2.5.2 Overview

The **Fields** package provides a generic representation of fields and forces as well as a mechanism to calculate such in a generic fashion. Using this interface, it is possible to create optimized c++ compiled code with the field/force calculations being abstracted from the use.

### 2.5.3 License

GNU LESSER GENERAL PUBLIC LICENSE; Version 3, 29 June 2007

### 2.5.4 Description of Released Contents

The contents of this released software package include

1. c++ implementation of generic fields/forces library.
2. Various c++ examples for using this generic interface.
3. Build/install files implemented as Boost.Build configuration files.
4. Documentation of the c++ interface
5. The entire history of changes made for this package

### 2.5.5 Major Release Versions

Tab. 2.5 of this section documents the major changes of Fields that warrant a new full release based on a major change in capabilities.

| Version | Revision Date | Release Date | Summary of Contributions/Changes |
|---|---|---|---|
| 0.1.0 | 27 May 2017 | 20 Dec 2017 | Initial Release of AFRL contributions<br>• Replaced build system with Boost.build<br>• Improved/add documentation and examples<br>• Further extracted code from older “olson-tools” package<br>• Various minor improvements and bug fixes |

Table 2.5: Released major versions of Fields

## 2.6 Cold-Atom Physics Laboratory Package: LabEnviron (as of 2017-08-28)

### 2.6.1 Background

Simple laboratory environmentals, such as temperature, pressure, and humidity, often play a role in the stability of table-top experiments.

### 2.6.2 Overview

This is a small webapp to record and display environmental data from various instances of environmental sensor development boards sold by Adafruit as interfaced by Raspberry Pi devices.

### 2.6.3 License

MIT License

### 2.6.4 Description of Released Contents

The contents of this released software package include

1. Python-Django models and views for storing and displaying data
2. Simple python script to record data
3. Simple documentation for quickstart usage
4. The entire history of changes made for this package

### 2.6.5 Major Release Versions

Tab. 2.6 of this section documents the major changes of LabEnviron that warrant a new full release based on a major change in capabilities.

| Version | Revision Date | Release Date | Summary of Contributions/Changes |
|---|---|---|---|
| 0.1.0 | 12 Jun 2017 | 20 Dec 2017 | Initial Release |

Table 2.6: Released major versions of LabEnviron

## 2.7 Cold-Atom Physics Laboratory Package: LightPipes (as of 2017-06-05)

### 2.7.1 Background

Frequenty, when applying light forces to atoms, it is necessary to more completely model the optical field that is used to manipulate atoms. One method of doing this is it implement a finite element optical field solver, such as represented by this package.

### 2.7.2 Overview

**LightPipes** is designed to model the propagation of light in coherent optical devices in scalar approximation. This toolbox is based on efficient propagation algorithms, providing extended possibilities of beam manipulation. Simulations of interferometers, holographic setups, laser resonators, lasers, Fourier optics, waveguides are possible.

### 2.7.3 License

GNU GENERAL PUBLIC LICENSE; Version 2, June 1991

### 2.7.4 Description of Released Contents

The contents of this released software package include

1. The c++ source code, where most of the actual computational work is performed.
2. Source code for the Octave functions that allow the c++ routines to be called from Octave.
3. Source code for the python functions that allow the c++ routines to be called from Python.
4. Build/install files implemented as Boost.Build configuration files.
5. The original documentation of the UNIX command line programs from Gleb Vdovin. This documentation also discusses many aspects of the theory implemented by the algorithms in the code.
6. Documentation of the new interfaces: c++, Octave, and Python.
7. The entire history of changes made for this package since Dr. Olson began modifications in 2008.

### 2.7.5 Major Release Versions

Tab. 2.7 of this section documents the major changes of LightPipes that warrant a new full release based on a major change in capabilities.

| Version | Revision Date | Release Date | Summary of Contributions/Changes |
|---|---|---|---|
| 0.2.0 | 27 May 2017 | 20 Dec 2017 | Initial Release of AFRL/RV contributions<br>• Replaced build system with Boost.build<br>• Improved/add documentation and examples<br>• Adds Python interface<br>• Replace local fftn with fftw3 third-party library<br>• Created new NxM (nonsquare) pixelated optical field |

Table 2.7: Released major versions of LightPipes

## 2.8 Cold-Atom Physics Laboratory Package: Physical (as of 2017-06-05)

### 2.8.1 Background

Physical units and constants are easily known but frequently present a problem in simulations if care is not taken. A common need for physics simulations and calculations is a means to calculate various items using proper units, with special care taken for dimensional analysis and error handling when mismatched units are used.

### 2.8.2 Overview

Physical units, constants, conversion, parsing, and runtime dimensional analysis package for C++, python, and others. This package provides a common access point for units and constants definitions with dimensional analysis and units mismatch testing provided.

### 2.8.3 License

MIT License

### 2.8.4 Description of Released Contents

The contents of this released software package include

1. c++ implementation of units and dimensional analysis package (released via source code).
2. Example c++ programs including an interactive units calculator.
3. Python implementation of units and dimensional analysis package.
4. Octave m-file that defines a multitude of units' and constants' values.
5. Gnuplot file that defines a multitude of units' and constants' values.
6. Build/install files implemented as Boost.Build configuration files.
7. Documentation of the interfaces: c++ and Python.
8. The entire history of changes made for this package.

### 2.8.5 Major Release Versions

Tab. 2.8 of this section documents the major changes of PHYSICAL that warrant a new full release based on a major change in capabilities.

| Version | Revision Date | Release Date | Summary of Contributions/Changes |
|---|---|---|---|
| 0.1.5 | 2 Feb 2011 | 4 Apr 2011 | Last release by AFRL/RD |
| 0.1.6 | 21 Dec 2016 | 20 Dec 2017 | • Several documentation corrections<br>• Improves build configuration<br>• Adds examples for expression parsing<br>• Improves Python implementation data, error handling, & interaction<br>• Upgrades parser/grammar to bison 3.0<br>• add (+) unary operator in Python |

Table 2.8: Released major versions of Physical

## 2.9 Cold-Atom Physics Laboratory Package: Python-FlyCapture2 (as of 2017-06-08)

### 2.9.1 Background

Camera data is typically necessary in order to analyze any cold-atom experiment. The PointGrey brand of cameras are particularly useful for cold-atom experiments with the various controls and timing capabilities they provide.

### 2.9.2 Overview

This library provides Python access to the PointGrey FlyCapture2 capable cameras.

### 2.9.3 License

MIT License

### 2.9.4 Description of Released Contents

The contents of this released software package include

1. Python object-oriented bindings for FlyCapture2 library.
2. Various Python examples for accessing/using FlyCapture2 cameras.
3. API Documentation of the Python interface
4. The entire history of changes made for this package

### 2.9.5 Major Release Versions

Tab. 2.9 of this section documents the major changes of Python-FlyCapture2 that warrant a new full release based on a major change in capabilities.

| Version | Revision Date | Release Date | Summary of Contributions/Changes |
|---|---|---|---|
| 0.1.0 | 12 Jun 2017 | 20 Dec 2017 | Initial Release |

Table 2.9: Released major versions of Python-FlyCapture2

## 2.10 Cold-Atom Physics Laboratory Package: PyLibNIDAQmx—OSS Contributions (as of 2017-10-17)

### 2.10.1 Background

PyLibNIDAQmx is a Python library that provides Python access to the proprietary National Instruments DAQmx c-library. This library exsists as an OSS project hosted at `https://github.com/pearu/pylibnidaqmx`. National Instruments' DAQmx software is freely distributed in binary form (with c-headers) for use with their commercially available hardware. One major market for their hardware is in instrumentation for academic and industry laboratories.

### 2.10.2 Overview

In order for AFRL's Arbwave to use the DAQmx hardware, it must access the DAQmx c-interface libraries. PyLibNIDAQmx provides this interface, but did not implement access to some key DAQmx functionality that Arbwave requires. AFRL's contributions to the PyLibNIDAQmx package implement the required changes for Arbwave to take advantage of the DAQmx libraries and hardware.

### 2.10.3 License

Public domain contributions

### 2.10.4 Description of Released Contents

The contents of this released software package include

1. Python-code in the form of patches as mandated by the maintainer of PyLibNIDAQmx.

### 2.10.5 Major Release Versions

Tab. 2.10 of this section documents the major changes of PyLibNIDAQmx—OSS Contributions that warrant a new full release based on a major change in capabilities.

| Version | Revision Date | Release Date | Summary of Contributions/Changes |
|---|---|---|---|
| 1 | 6 May 2016 | 20 Dec 2017 | Initial Release |

Table 2.10: Released major versions of PyLibNIDAQmx—OSS Contributions

## 2.11 Cold-Atom Physics Laboratory Package: Python-Viewpoint DIO64 (as of 2017-06-08)

### 2.11.1 Background

Arbitrary timing is nearly always required for cold-atom experiments in order to control various lasers, shutters, currents, etc. Viewpoint USA sells a digital input/output board (the DIO-64) that allows for arbitrary digital output signals to be specified without significant memory or CPU overhead.

### 2.11.2 Overview

This is a python library for interacting with Viewpoint's DIO-64 digital board in such a way as to provide arbitrary digital timing signals.

### 2.11.3 License

MIT License

### 2.11.4 Description of Released Contents

The contents of this released software package include

1. Python object-oriented bindings for FlyCapture2 library.
2. Various Python examples for accessing/using FlyCapture2 cameras.
3. API Documentation of the Python interface
4. The entire history of changes made for this package

### 2.11.5 Major Release Versions

Tab. 2.11 of this section documents the major changes of Python-Viewpoint DIO64 that warrant a new full release based on a major change in capabilities.

| Version | Revision Date | Release Date | Summary of Contributions/Changes |
|---|---|---|---|
| 0.1.0 | 12 Jun 2017 | 20 Dec 2017 | Initial Release |

Table 2.11: Released major versions of Python-Viewpoint DIO64

## 2.12 Cold-Atom Physics Laboratory Package: VXI-11 for Python (as of 2017-10-13)

### 2.12.1 Background

Laboratory experiments, such as those for cold-atom physics, generally require the use of various instruments and measurement devices. One common standard for communication for instruments is the set of Versa Module Europa (VME) eXtensions for Instrumentation (VXI) protocols. This set of protocols ultimately allows standard GPIB style commands to be executed over various types of interfaces. One such interface is the ethernet interface. VXI over ethernet is implemented as a set of Remote Procedural Calls (RPC) and is called VXI-11.

### 2.12.2 Overview

This package implements the VXI-11 protocols in pure Python for use in Python programs/applications.

### 2.12.3 License

GNU GENERAL PUBLIC LICENSE; Version 2, June 1991

### 2.12.4 Description of Released Contents

The contents of this released software package include

1. Python implementation of VXI-11, portmap, and Sun RPC
2. Various Python examples for using VXI-11 for Python
3. API Documentation of the VXI-11 for Python, portmap, and Sun RPC
4. The entire history of changes made for this package

### 2.12.5 Major Release Versions

Tab. 2.12 of this section documents the major changes of VXI-11 for Python that warrant a new full release based on a major change in capabilities.

| Version | Revision Date | Release Date | Summary of Contributions/Changes |
|---|---|---|---|
| 0.1.0 | 14 Jul 2017 | 20 Dec 2017 | Initial Release of AFRL/RV contributions<br>• Updated RPC lib from pynfs (UMICH CITI project)<br>• Add portmapper client implementation (integral to VXI-11)<br>• Automatically create specialized instrument links for known instruments<br>• Add vxi11.Link specializations for<br>– Anritsu Optical Spectrum Analyzer (OSA)<br>– Keysight Radio-Frequency (RF) Spectrum Analyzer<br>– Keysight Network Analyzer<br>• Add API documentation |

Table 2.12: Released major versions of VXI-11 for Python

## 2.13 Cold-Atom Physics Laboratory Package: Timing Generator (as of 2017-10-17)

### 2.13.1 Background

Arbitrary timing is nearly always required for cold-atom experiments in order to control various lasers, shutters, currents, etc. Some commercial solutions to provide arbitrary timing do exist, but none were found that did not include limitations in the number of timed digital transitions supported for continous-timing-generation mode. Commercial providers refused to address these limitations and also placed some of their products on a path of end-of-life.

### 2.13.2 Overview

This **Timing Generator** package implements an arbitrary timing control using firmware running on a Field-Programmable Gate Array (FPGA). For this effort, we chose to use an FPGA development board by Marvin Test, Inc. In principle, we could also deploy our code on a similar board from a different vendor.

### 2.13.3 License

Firmware: GNU GENERAL PUBLIC LICENSE; Version 3, 29 June 2007
Python Software: MIT License

### 2.13.4 Description of Released Contents

The contents of this released software package include

1. Python interfaces to the Marvin Test Gx* libraries
2. Python interface to AFRL TimingBoard firmware
3. Quartus-II firmware source for Marvin Test Gx3500 implementing AFRL's arbitrary timing unit
4. The entire history of changes made for this package
5. Compiled firmware:
   (a) TimingGenerator128b-0.97-1-g7e2d373.rpd : used store firmware in boot-loadable, non-volatile memory.
   (b) TimingGenerator128b-0.97-1-g7e2d373.svf : used to load firmware directly to FPGA configuration memory.

### 2.13.5 Major Release Versions

Tab. 2.13 of this section documents the major changes of Timing Generator that warrant a new full release based on a major change in capabilities.

| Version | Revision Date | Release Date | Summary of Contributions/Changes |
|---|---|---|---|
| 0.97-1 | 1 Dec 2016 | 20 Dec 2017 | Initial Release |

Table 2.13: Released major versions of Timing Generator

## 2.14 Cold-Atom Physics Laboratory Package: Xylose Utility Library (as of 2017-06-05)

### 2.14.1 Background

Many common algorithms and routines are needed in a common library in order to facilitate reuse. Such algorithms are frequently provided by standard libraries, although standard libraries do not generally address common needs for physics simulations.

### 2.14.2 Overview

This library represents a collection of utility routines that are frequently useful for larger computational physics projects. The routines and data structures defined in this collection do not perform any particular computational function of themselves, but rather ease the implementation of such.

### 2.14.3 License

GNU LESSER GENERAL PUBLIC LICENSE; Version 3, 29 June 2007

### 2.14.4 Description of Released Contents

The contents of this released software package include

1. c/c++/assembly implementation of various computational tools.
2. Various c/c++ examples for using Xylose computational tools.
3. Build/Install files implemented as Boost.Build configuration files.
4. Documentation of the c/c++ interface.
5. The entire history of changes made for this package.

### 2.14.5 Major Release Versions

Tab. 2.14 of this section documents the major changes of Xylose Utility Library that warrant a new full release based on a major change in capabilities.

| Version | Revision Date | Release Date | Summary of Contributions/Changes |
|---|---|---|---|
| 0.1.2 | 4 Apr 2011 | 4 Apr 2011 | Last release by AFRL/RD |
| 0.1.3 | 21 Dec 2016 | 20 Dec 2017 | • Add examples for thread pool control<br>• Improve Vector implementation<br>• Improve Runge-Kutta usability and tests<br>• Improve documentation<br>• Improve timing simulator |

Table 2.14: Released major versions of Xylose Utility Library

# DEFINITIONS AND ACRONYMS

**DoD** Department of Defense . . . 1

**AFRL** Air Force Research Laboratory . . . ii

**SANDS** Strategic Atomic Navigation Devices and Systems.

**ITAR** International Traffic in Arms Regulations . . . ii

**AFRL/RV** AFRL Space Vehicles Directorate . . . iii

**AFRL/RVB** AFRL Battlespace Environment Division . . . iii

**OSS** Open Source Software . . . iii

**DoDD** Department of Defense Directive.

**DoDI** Department of Defense Instruction.

**AFI** Air Force Instruction.

**CIO** Chief Information Officer . . . 1

**FAQ** Frequenty Asked Questions.

**SME** Subject Matter Expert.

**IA** Information Assurance.

**SDL** Space Dynamics Laboratory.

**USRA** Universities Space Research Association.

**AFRL/RD** AFRL Directed Energy Directorate . . . 5

**Comedi** Control and Measurement Device Interface – an OSS project to interface with data acquisition and experimental control hardware.

**DAQ** Data Acquisition.

**EMI** Electromagnetic interference . . . 6

**VME** VMEbus (Versa Module Europa bus) is a computer bus standard, originally developed for the Motorola 68000 line of CPUs, but later widely used for many applications and standardized by the IEC as ANSI/IEEE 1014-1987. [quoted from Wikipedia] . . . 15

**VXI** VME eXtensions for Instrumentation . . . 15

**RPC** Remote procedural calls . . . 15

**OSA** Optical spectrum analyzer . . . 15

**RF** Radio frequency . . . 15

**FPGA** Field-Programmable Gate Array . . . 16

# Bibliography


[1] Chief Information Officer; U.S. Department of Defense. DoD Open Source Software (OSS) FAQ. URL http://dodcio.defense.gov/Open-Source-Software-FAQ/.

[2] Comedi: Control and Measurement Device Interface. http://www.comedi.org. Accessed: 2017-10-13.

[3] S. E. Olson and A. J. Christlieb. Gridless dsmc. *J. Comp. Phys.*, 227(17):8035–8064, 2008. doi: 10.1016/j.jcp.2008.04.038.

[4] Spencer E. Olson, Andrew J. Christlieb, and Fredrik K. Fatemi. PID feedback for load-balanced parallel gridless DSMC. 181(12):2063–2071, 2010. doi: 10.1016/j.cpc.2010.06.045.

[5] S. E. Olson, A. J. Christlieb, and G. Raithel. Pressure-driven evaporative cooling in atom guides. *Phys. Rev. A*, 90(4):043612, 2014. doi: 10.1103/PhysRevA.90.043612.