Python Software

GT Off-Road Racing | Data Acquisitions

[Benjamin Boeckman]

[06/18/2021]

Table of Contents

[1.0 Overview 2](#_Toc77099290)

[1.1 Introduction 2](#_Toc77099291)

[1.2 Point of Contact 2](#_Toc77099292)

[2.0 Software Needs and Installation Guide 3](#_Toc77099293)

[2.1 Python Installation and Version 3](#_Toc77099294)

[2.2 IDE Options 3](#_Toc77099295)

[2.3 Installing Software Dependencies 3](#_Toc77099296)

[3.0 Style and Structure Guide 4](#_Toc77099297)

[3.1 Python Style 4](#_Toc77099298)

[4.0 Python Development Projects 5](#_Toc77099299)

[5.0 References 6](#_Toc77099300)

[6.0 Revision History 7](#_Toc77099301)

# 1.0 Overview

## 1.1 Introduction

Data is central to what we do here in DAQ (Data Acquisition). Because of this, we need to take advantage of robust programming languages that are better for certain situation than others. On the lower-level side of collection, we use a majority of C++ because of its ease of memory handling and efficiency with weaker computational devices. Once the data is collected on those devices though, we then need to do something with it. This includes separating it, storing it, organizing it, displaying it, and anything else you can think of. This is where Python comes in.

Python is a great programming language because it is extremely versatile and has many different ways of solving the same problem. We use it to parse packets, organize data, and display it through a GUI (graphical user interface) in our data application\*. This is critical as it gives us accurate feedback about the car so that we can assess its actual performance without having to speculate or just trust in the design.

Now that you know you need to use Python to get involved with development for dealing with collected data, you may be wondering where to get experience with it, what IDE you should use, what version of Python to install, briefly how it works, what dependencies you may need to work on the software, and the list goes on and on. I will do my best to tackle and assess each of these roadblocks that keep you from becoming the software engineering machine we all know you can be.

\*For more information on how to use the application, see “DAATA Documentation” in the Software Documentation folder of the Documentation Repository.

## 1.2 Point of Contact

Benjamin Boeckman: benboeck@yahoo.com

# 2.0 Software Needs and Installation Guide

## 2.1 Python Installation and Version

We use Python 3.6 here at DAQ for our data GUI app.

Python 3.6.8 Download Link: <https://www.python.org/downloads/release/python-368/>

Once navigating to this link, find the corresponding link at the bottom of the page for your specific operating system.

An important thing if you are going to be using Python is that it must be referenced to your PATH variables. If you don’t know what those are, don’t worry. These are just applications that your OS can recognize and run when they need to be executed for your code to work. While you’re downloading and installing Python, be sure to specify that you want it to automatically add it to your PATH if it asks. If it does not, or if you’re not sure, don’t worry. Below is a link to walk you through ensuring that Python is in your PATH for whatever operating system you use. Just scroll down until it walks through the PATH for whatever system you use.

<https://www.tutorialspoint.com/python/python_environment.htm>

## 2.2 IDE Options

## 2.3 Installing Software Dependencies

Once you set yourself up in our GIT organization and Repositories, you’ll need to be in the GTORDaata Repository that you can find in the link below or in your clone of the repository on your local machine to run the dependencies batch file.

<https://github.com/Georgia-Tech-Off-Road/GTORDaata>

Search the above page or in the repository on your machine until you locate the “**install\_dependencies.bat**” file on the highest directory of the repository. Download it, if it is not already on your machine, and then double click it to run it and let the command prompt do its magic. Be sure, though, to get the repositories you need cloned and on your local machine because this batch file could change from time to time and you’ll need the most recent dependencies to alter things in the future.

# 3.0 Style and Structure Guide

## 3.1 Python Style

### 3.1.1 Variables and Functions

We use snake case for variables and functions as seen in the examples below:

![Text

Description automatically generated]()

### 3.1.2 Classes

As for class names, we use camel case as seen in the example below:



# 4.0 Python Development Projects

## 4.1 DAATA (Data Acquisition and Telemetry Application)

## 4.2 . . .

# 5.0 References

# 6.0 Revision History

06/18/2021 (Benjamin Boeckman) – First initial write-up of documentation

07/20/2021 (Benjamin Boeckman) – Adding style guide and clarifying sections to be elaborated on