Authors: Chunyang Gao, Dominik Mühlematter, Nick Teutschmann
Lecture: Geospatial Data Acquisition
02.11.2023

Automatic Measurement with a Total Station via GeoCOM

Introduction

In geospatial data acquisition, a crucial task involves automating data collection and analysis by developing software capable of controlling sensors autonomously. This is particularly essential for continuous monitoring over longer periods of time. This exercise aims to master the operation of a Leica TS60 total station through serial communication utilizing the Leica GeoCOM protocol. First, the developed software enables the measurement of an arbitrary number of points in a semi-automated fashion. This involves manually aligning the total station approximately with each point to be measured in a single face for the first set.

Subsequently, the software allows the total station to perform automated measurements of these points in two faces for an arbitrary number of sets. The obtained measurements are then analyzed using the simplified test procedure for horizontal directions as outlined in the ISO 17123-3 standard [1].

Furthermore, the software possesses the capability to save the acquired measurements, analysis results, and software logs locally on the computer in use, all without the need for user interaction at the total station.

Software Architecture and Data Flow

The software architecture includes three main parts as introduced below and displayed in *Figure 1*. The software is primarily based on the Python libraries NumPy [2], Pandas [3], and PySerial [4].

Modules

First, the software starts with initializing the configuration variables, the default values of which are presented in *Table 1*. The user should modify these variables located at the top of the Python script according to the task. A log file is created to record the measuring procedure and settings configured by the user. In this module, the user is asked to input the desired output file name, select a port, and specify the number of sets and points to be measured. The software then automatically establishes the connection to the total station. Afterwards, the software configures the settings of the total station according to the configuration variables.

Variable Name	Default Value	Request ID
ATR_mode	1 (ATR on)	18005
ATR_window	10 gon	9041
tolerances	0.00063662, 0.00063662 gon	9007
prism_type	0 (Circular)	17008
target_type	0 (Reflector)	17021
reflector_height	0 m	2012
Face	1,1,0 (POSMode, ATRMode)	9028

Table 1: Configuration settings for the total station, including default values.

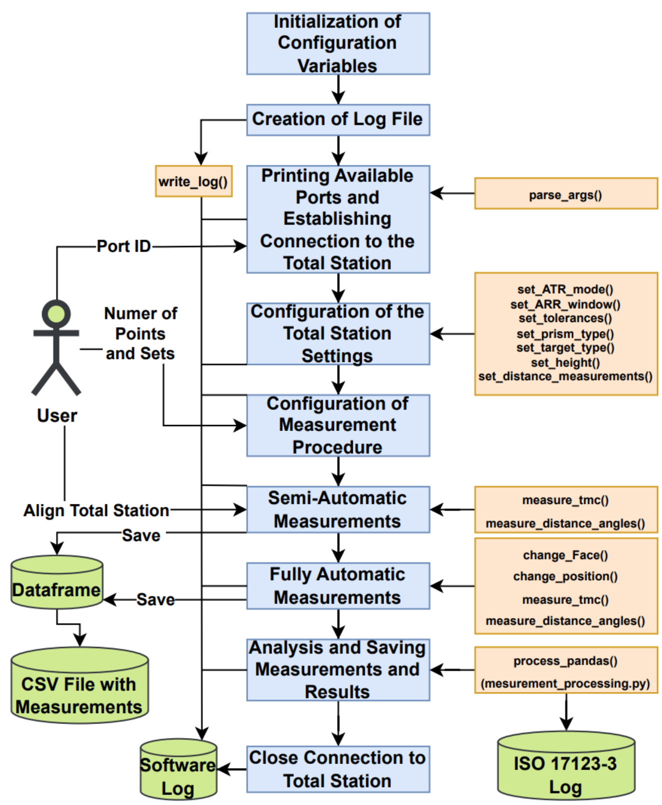

Figure 1: Proposed Software's Software Architecture.

The user needs to first measure the points manually in the measurement module. The software will prompt the user to align the total station with the desired measurement point. Subsequently, the user can perform the measurement from the Python terminal. This procedure is repeated for each point in the first face. Next, the software will take control of the total station and automatically measure the same points in the second face for the first set without requiring user interaction. Similarly, the point measurements are automatically repeated in the two faces for the remaining sets that were specified.

The measurements are stored in a dataframe and saved as a CSV file. This file includes the point name/ID, horizontal and vertical angles in gon, distance in meters, as well as the face and set ID of each measurement. A simplified test procedure for horizontal directions following the ISO 17123-3 norm [1] is conducted and the result is saved in a "_iso_17123_3.log" file. Finally, the connection to the total station is closed to release the resources.

Results and Discussion

By using the interface protocol GeoCOM for the communication between a computer and a total station (TS60), it is possible to write applications like ours running on operating systems supporting ASCII-based communication through serial ports. Horizontal and vertical angles, as well as distances can be measured automatically by sending requests and receiving the response.

In order to test our application we performed 5 sets of measurements, each of which consists out of 5 points in 2 faces. The obtained result documents the point name/ID, horizontal and vertical angles in gon, distance in meters, as well as the face and set number of each measurement. The simplified test procedure shows that the software works well for measuring the horizontal directions with highest accuracy, as shown in *Table 2*. The superior accuracy of our measurements compared to official instrument specifications may be attributed to well-calibrated instruments.

Measurements	Standard Deviation [mgon]
Our Measurements	0.07
Specification by Leica [5]	0.15

Table 2: Comparison of standard deviations for horizontal angles according to ISO 17123-3 [1].

While important instrument and measurement settings can be specified by the user in the script's header section as shown in *Table 1*, our software utilizes command-line inputs to enhance the user experience for frequently changing settings. This approach allows users to specify the measurement and log file names, the port for establishing the connection, the number of data points and measurement sets, the names of the points and also provides instructions for the manual measurements. This form of human-computer interaction has already proven successful during the practical applications we have conducted. For future improvements of the software, it can be enhanced to have an even higher flexibility (more measurement modes, deleting or adding points freely, graphical user interface).

Another enhancement can be implemented by improving the error tolerance. Currently, the software stops its execution in the event of an error during the measurements and the user can access the error message in the log file or through the Python terminal. However, the software could be improved to handle such errors more efficiently by providing options to either retry failed measurements or skip the problematic measurement entirely and continue with the overall measurement procedure. This would be particularly crucial for monitoring tasks conducted over longer time periods.

Challenges and Learning Experience

We encountered several challenges during software development and testing, primarily related to the GeoCOM documentation. Firstly, it was often difficult to identify the appropriate function for a given task. For instance, there are functions that only retrieve the last measurement conducted with the total station, and others that actively perform the measurements, leading to confusion during development.

However, the most problematic challenge we faced during debugging was a perplexing issue in the GeoCOM documentation. Specifically, the function used to rotate the total station's telescope to the other face (ASCII-Request ID: 9028) includes a dummy parameter, which, according to the documentation, should be of data type boolean. However, this method only functions correctly when provided with a value of either "0" or "1" as input, whereas using the standard Python booleans "True" and "False" resulted in an error message indicating transmission cable issues. Such feedback made the debugging of the software time demanding.

Despite the challenges and problems we encountered, we enjoyed the process of developing the software and gained valuable working experience with the total station.

Conclusion

In conclusion, our software enables communication with external devices and is fully capable of controlling a Leica TS60 total station using the GeoCOM interface protocol. The conducted measurements meet high accuracy standards. Our software can serve as a framework for developing more advanced applications for measurement and monitoring tasks with a total station.

Time Management

Table 3 displays the time required to complete this assignment. As discussed in the prior chapter, our project encountered delays due to the perplexing issue in the GeoCOM documentation and the challenge of identifying this issue within the code.

Task	**Time Invested [h]**
Study Background Material and Task	2
Software Development	6
Debugging and Measurements	21
Report Drafting	4.5
User Manual Creation	1.5
Presentation Preparation	2
Total Time Invested	37
Planned Time Budget	27
Time Variance	+10

Table 3: Time allocated to each work package within the project.

References

[1] International Organization for Standardization. (2001). ISO 17123-3: Optics and optical instruments – Field procedures for testing geodetic and surveying instruments – Part 3: Theodolites. Geneva, Switzerland: ISO.

[2] Harris, C.R., Millman, K.J., van der Walt, S.J. et al. Array programming with NumPy. Nature 585, 357–362 (2020). DOI: 10.1038/s41586-020-2649-2.

[3] The pandas development team. (2020). pandas-dev/pandas: Pandas. Zenodo. DOI: 10.5281/zenodo.3509134 (source)

[4] PySerial. (n.d.). PySerial Documentation. Retrieved October 28, 2023, from https://pyserial.readthedocs.io/en/latest/pyserial.html

[5] Leica Geosystems AG. (n.d.). Leica Nova TS60 Total Station Datasheet. Leica Geosystems AG. https://leica-geosystems.com/-/media/files/leicageosystems/products/datasheets/leica_nova_ts60_ds.ashx?la=de-de&hash=2CCBBCF7639E1F6647DAA23A6D10EB87