

tlmc Version 1 format specification

October 29, 2020

This file describes the content of the so-called `tlmc` format. `tlmc` stands for compressed telemetry: it is simply a standard HDF-5 file with compression enable, that can be opened with any HDF-5 reader. This document specifies the organisation of data in this file.

The examples in this document are made using the `h5py` library ; `file` is an `h5py.File` object.

The telemetry of the robot outputs two different types of object: constants, wich are (key, value) pairs, and variables. Variables each have a unique name, and consists of two time series: one for time, one for values. Variables can have various basic types, and can have metadatas associated to them.

The `tlmc` will be organised as follow:

- The root group shall contain an attribute ‘`TLMC_VERSION`’, which stores an int specifying the version of the `tlmc` standard use. This document describes `VERSION=1`.
- The root group shall contain an attribute ‘`START_TIME`’, which stores a float specifying the absolute start time of the log, in second relative to the UNIX epoch.
- A group `constants` will store the original telemetry constants in its attribute dictionary.
- A second group `variables` will store the variables.
 - Each subgroup `variableName` represents a variable, originally named `variableName`. Each variable group contains:
 - * A `value` 1D dataset representing the variable’s values through time.
 - * A `time` 1D dataset representing the time instants relative to the ‘`START_TIME`’ file constant. This dataset will contain an attribute `unit` specifying the ratio to SI unit (i.e. 1 second). For instance when using nanoseconds, `file["variables/myvariable/time"].attrs["unit"]` evaluates to `1e-9`.
 - * Variable-specific metadatas stored in the group’s attribute.

For storage efficiency, all datasets will be stored using the ‘`gzip`’ filter with compression level of 4, and the ‘`shuffle`’ filter. These are enabled in `h5py` using the following flags:

```
f.create_dataset(name, data=data_array, compression='gzip', shuffle=True)
```

Examples

Here is a (simplified) view of a tlmc file using the h5dump

```
HDF5 "data/20200921T101310Z_LogFile.tlmc" {
GROUP "/" {
  ATTRIBUTE "START_TIME" {
    DATATYPE H5T_IEEE_F64LE
    DATASPACE SCALAR
    DATA {
      (0): 1.60068e+09
    }
  }
  ATTRIBUTE "VERSION" {
    DATATYPE H5T_STD_I64LE
    DATASPACE SCALAR
    DATA {
      (0): 1
    }
  }
  GROUP "constants" {
    ATTRIBUTE "ButtonMapper.INSTALLATION_LEFT" {
      DATATYPE H5T_STD_I64LE
      DATASPACE SCALAR
      DATA {
        (0): 256
      }
    }
    ATTRIBUTE "Telemetry.StartTime" {
      DATATYPE H5T_STD_I64LE
      DATASPACE SCALAR
      DATA {
        (0): 1600683190350165284
      }
    }
    ...
  }
  GROUP "variables" {
    GROUP "HighLevelController.currentPositionLeftSagittalHip" {
      DATASET "time" {
        DATATYPE H5T_STD_I64LE
        DATASPACE SIMPLE { ( 338623 ) / ( 338623 ) }
        ATTRIBUTE "unit" {
          DATATYPE H5T_IEEE_F64LE
          DATASPACE SCALAR
          DATA {
            (0): 1e-09
          }
        }
      }
      DATASET "value" {
        DATATYPE H5T_IEEE_F32LE
        DATASPACE SIMPLE { ( 338623 ) / ( 338623 ) }
```

```
    }  
  }  
  ...  
}  
}  
}
```

And here is an example python code for browsing a tlmc file:

```
import h5py  
  
file = h5py.File('my_file.tlmc', 'r')  
  
print(file.attrs['VERSION']) # Prints 1  
print("The log contains the following constants:")  
for k, v in file['constants'].attrs.items():  
    print(k, v)  
print(f"Log start time: {file['START_TIME']}")  
print("The log contains the following variables:")  
for variable_name in file['variables']:  
    print(variable_name)
```