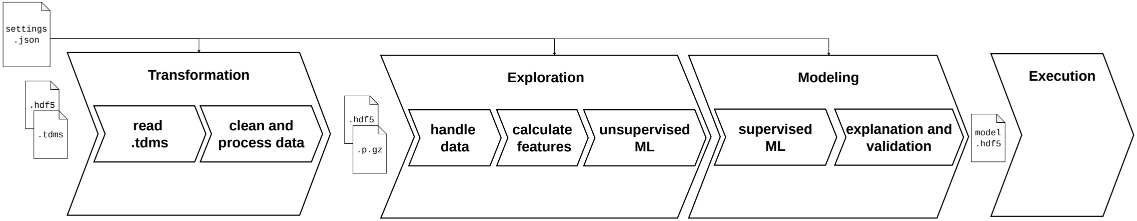
CERN

Transformation ML Framework

on basis of the XBox2 Data Set

Lorenz Fischl

Introduction



Introduction

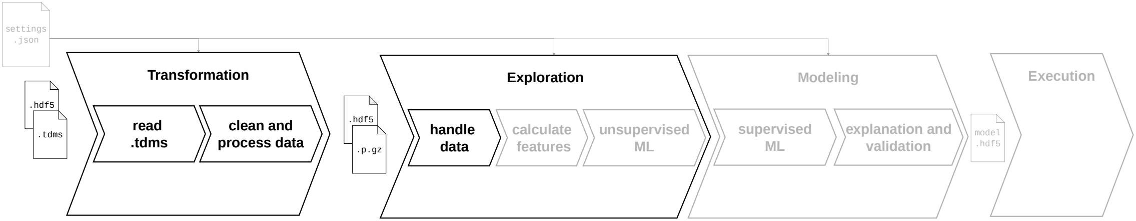


Table of Contents

XBox2 File Structure

- Event-&TrendData

Choosing a Data Format

- reading tdms files
- pandas + compressed pickle
- Conclusion

Table of Contents

XBox2 File Structure

Event-&TrendData

Choosing a Data Format

reading tdms files

pandas + compressed pickle

Conclusion

Table of Contents

XBox2 File Structure

Event-&TrendData

Choosing a Data Format

reading tdms files

pandas + compressed pickle

Conclusion

XBox2 Data

XBox2 Data

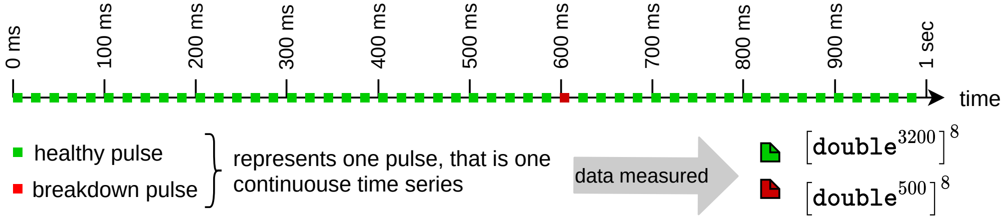
Every 20ms a pulse is sent into the RF cavity for particle acceleration.

XBox2 Data

Every 20ms a pulse is sent into the RF cavity for particle acceleration. Sometimes an arc forms. Those events are called breakdown.

XBox2 Data

Every 20ms a pulse is sent into the RF cavity for particle acceleration. Sometimes an arc forms. Those events are called breakdown. vspace1cm


0 ms
100 ms
200 ms
300 ms
400 ms
500 ms
600 ms
700 ms
800 ms
900 ms
1 sec
time
🟩 healthy pulse
🟥 breakdown pulse
represents one pulse, that is one continuouse time series
data measured
$\left[\texttt{double}^{3200}\right]^{8}$
$\left[\texttt{double}^{500}\right]^{8}$

XBox2 EventData

XBox2 EventData

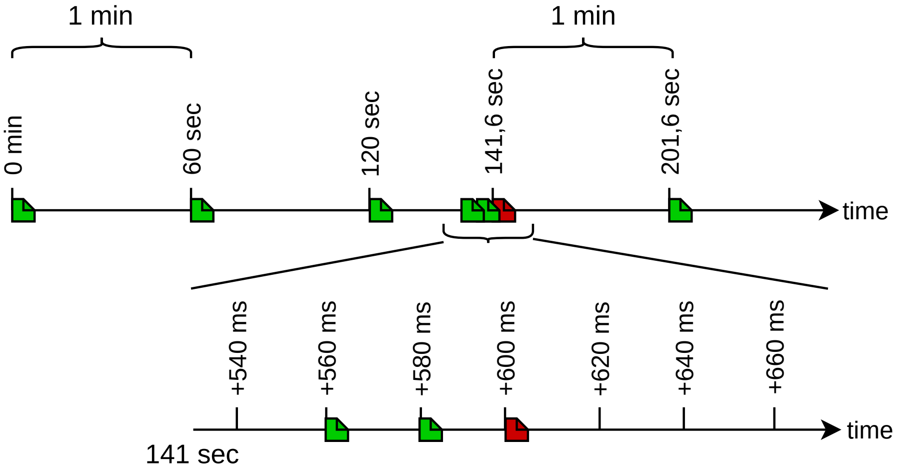
A log group of one pulse is stored every minute.

XBox2 EventData

A log group of one pulse is stored every minute.
When a breakdown happens the corresponding log group + the two prior log groups are stored.

XBox2 EventData

A log group of one pulse is stored every minute.
When a breakdown happens the corresponding log group + the two prior log groups are stored.



XBox2 TrendData

XBox2 TrendData

35 values about the environmental conditions (that don't change rapidly) are stored roughly every 1,5 sec.

XBox2 TrendData

35 values about the environmental conditions (that don't change rapidly) are stored roughly every 1,5 sec.
All TrendData of one day is stored in 1-2 groups.

XBox2 TrendData

35 values about the environmental conditions (that don't change rapidly) are stored roughly every 1,5 sec.
All TrendData of one day is stored in 1-2 groups.

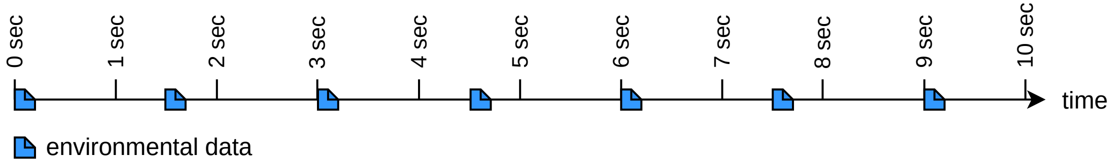


Table of Contents

XBox2 File Structure

- Event-&TrendData

Choosing a Data Format

- reading tdms files
- pandas + compressed pickle
- Conclusion

Table of Contents

XBox2 File Structure
- Event-&TrendData

Choosing a Data Format
- reading tdms files
- pandas + compressed pickle
- Conclusion

nptdms

nptdms

- package nptdms can read and handle .tdms files

`nptdms`

- package `nptdms` can read and handle `.tdms` files
- very slow (ex.: read of a 100MB file can take $>$30sec)

`nptdms`

- package `nptdms` can read and handle `.tdms` files
- very slow (ex.: read of a 100MB file can take >30sec)
- very space inefficient (ex. TrendData: 20,5 GB in .tdms → 2.8 GB of data)

Table of Contents

XBox2 File Structure

- Event-&TrendData

Choosing a Data Format

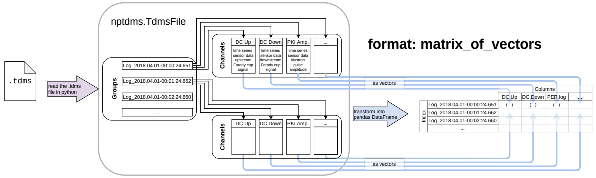
- reading tdms files
- pandas + compressed pickle
- Conclusion

Transformation

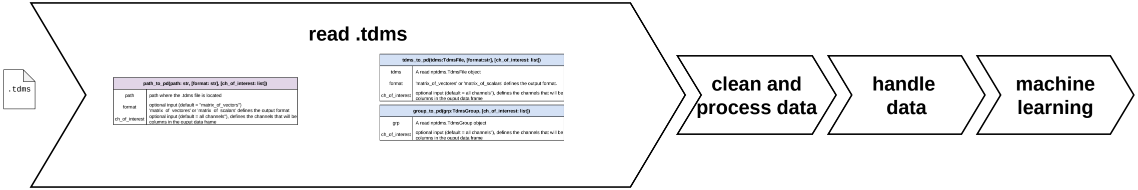
Transformation

`.tdms` ⟶ `pd df/ dictionary` ⟶ `pickle` ⟶ `.gzip`

Transformation: read.tdms


.tdms
read the .tdms file in python
nptdms.TdmsFile
Groups
Log_2018.04.01-00:00:24.651
Log_2018.04.01-00:01:24.662
Log_2018.04.01-00:02:24.660
...
Channels
DC Up
time series sensor data upstream Farady cup signal
DC Down
time series sensor data downstream Farady cup signal
PKI Amp.
time series sensor data klystron pulse amplitude
...
Channels
DC Up
DC Down
PKI Amp.
...
format: matrix_of_vectors
as vectors
as vectors
transform into pandas DataFrame

Index	Columns			
	DC Up	DC Down	PER log	...
Log_2018.04.01-00:00:24.651	(...)	(...)	(...)	
Log_2018.04.01-00:01:24.662				
Log_2018.04.01-00:02:24.660				
...				



read .tdms

path_to_pd(path: str, [format: str], [ch_of_interest: list])	
path	path where the .tdms file is located
format	optional input (default = "matrix_of_vectors") 'matrix of vectores' or 'matrix of scalars' defines the output format
ch_of_interest	optional input (default = all channels"), defines the channels that will be columns in the ouput data frame

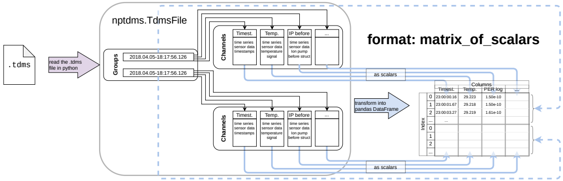
tdms_to_pd(tdms:TdmsFile, [format:str], [ch_of_interest: list])	
tdms	A read nptdms.TdmsFile object
format	'matrix_of_vectores' or 'matrix_of_scalars' defines the output format.
ch_of_interest	optional input (default = all channels"), defines the channels that will be columns in the ouput data frame

group_to_pd(grp:TdmsGroup, [ch_of_interrest: list])	
grp	A read nptdms.TdmsGroup object
ch_of_interest	optional input (default = all channels"), defines the channels that will be columns in the ouput data frame

clean and process data

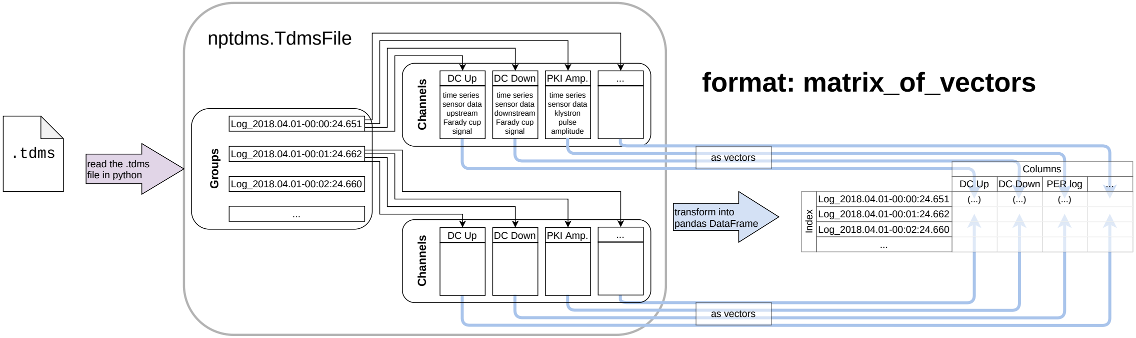
handle data

machine learning


.tdms
read the .tdms file in python
nptdms.TdmsFile
Groups
2018.04.05-18:17:56.126
2018.04.05-18:17:56.126
Channels
Timest.
time series sensor data timestamps
Temp.
time series sensor data temperature signal
IP before
time series sensor data Ion pump before struct
...
Channels
Timest.
time series sensor data timestamps
Temp.
time series sensor data temperature signal
IP before
time series sensor data Ion pump before struct
...
format: matrix_of_scalars
as scalars
as scalars
transform into pandas DataFrame

Index		Columns			
		Timest.	Temp.	PER log	...
	0	23:00:00.16	29.223	1.50e-10	
	1	23:00:01.67	29.218	1.50e-10	
	2	23:00:03.27	29.219	1.61e-10	
	...	...			
	0				
	1				
	2				
	...				

Transformation: read.tdms

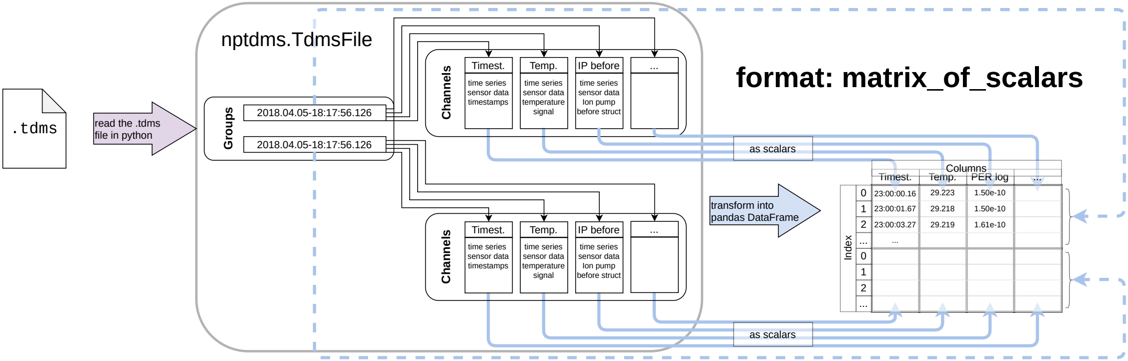


format: matrix_of_vectors

.tdms
read the .tdms file in python
nptdms.TdmsFile
Groups
Log_2018.04.01-00:00:24.651
Log_2018.04.01-00:01:24.662
Log_2018.04.01-00:02:24.660
...
Channels
DC Up
time series sensor data upstream Farady cup signal
DC Down
time series sensor data downstream Farady cup signal
PKI Amp.
time series sensor data klystron pulse amplitude
...
Channels
DC Up
DC Down
PKI Amp.
...
as vectors
as vectors
transform into pandas DataFrame
Index
Columns

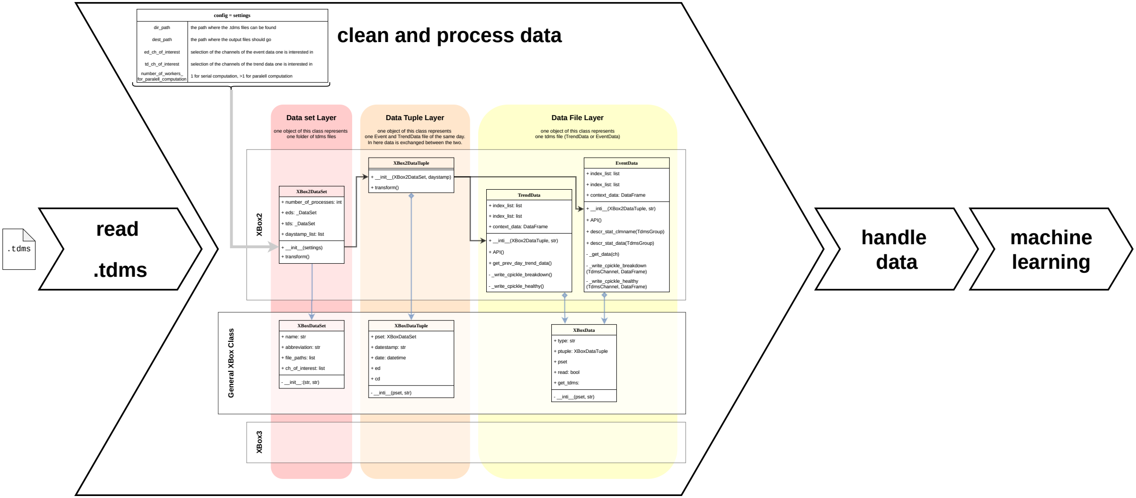
	DC Up	DC Down	PER log	...
Log_2018.04.01-00:00:24.651	(...)	(...)	(...)	
Log_2018.04.01-00:01:24.662				
Log_2018.04.01-00:02:24.660				
...				

Transformation: read.tdms


.tdms
read the .tdms file in python
nptdms.TdmsFile
Groups
2018.04.05-18:17:56.126
2018.04.05-18:17:56.126
Channels
Timest.
time series sensor data timestamps
Temp.
time series sensor data temperature signal
IP before
time series sensor data Ion pump before struct
...
Channels
Timest.
time series sensor data timestamps
Temp.
time series sensor data temperature signal
IP before
time series sensor data Ion pump before struct
...
format: matrix_of_scalars
as scalars
as scalars
transform into pandas DataFrame
Columns
Index

	Timest.	Temp.	PER log	...
0	23:00:00.16	29.223	1.50e-10	
1	23:00:01.67	29.218	1.50e-10	
2	23:00:03.27	29.219	1.61e-10	
...	...			
0				
1				
2				
...				

Transformation: Clean and Process data with classes



.tdms

read

.tdms

clean and process data

config = settings	
dir_path	the path where the .tdms files can be found
dest_path	the path where the output files should go
ed_ch_of_interest	selection of the channels of the event data one is interested in
td_ch_of_interest	selection of the channels of the trend data one is interested in
number_of_workers_for_paralell_computation	1 for serial computation, >1 for paralell computation

Data set Layer

one object of this class represents one folder of tdms files

Data Tuple Layer

one object of this class represents one Event and TrendData file of the same day. In here data is exchanged between the two.

Data File Layer

one object of this class represents one tdms file (TrendData or EventData)

XBox2

XBox2DataSet
+ number_of_processes: int
+ eds: _DataSet
+ tds: _DataSet
+ daystamp_list: list
+ __init__(settings)
+ transform()

XBox2DataTuple
+ __init__(XBox2DataSet, daystamp)
+ transform()

TrendData
+ index_list: list
+ index_list: list
+ context_data: DataFrame
+ __inti__(XBox2DataTuple, str)
+ API()
+ get_prev_day_trend_data()
- _write_cpickle_breakdown()
- _write_cpickle_healthy()

EventData
+ index_list: list
+ index_list: list
+ context_data: DataFrame
+ __inti__(XBox2DataTuple, str)
+ API()
+ descr_stat_clmname(TdmsGroup)
+ descr_stat_data(TdmsGroup)
- _get_data(ch)
- _write_cpickle_breakdown (TdmsChannel, DataFrame)
- _write_cpickle_healthy (TdmsChannel, DataFrame)

General XBox Class

XBoxDataSet
+ name: str
+ abbreviation: str
+ file_paths: list
+ ch_of_interest: list
- __init__:(str, str)

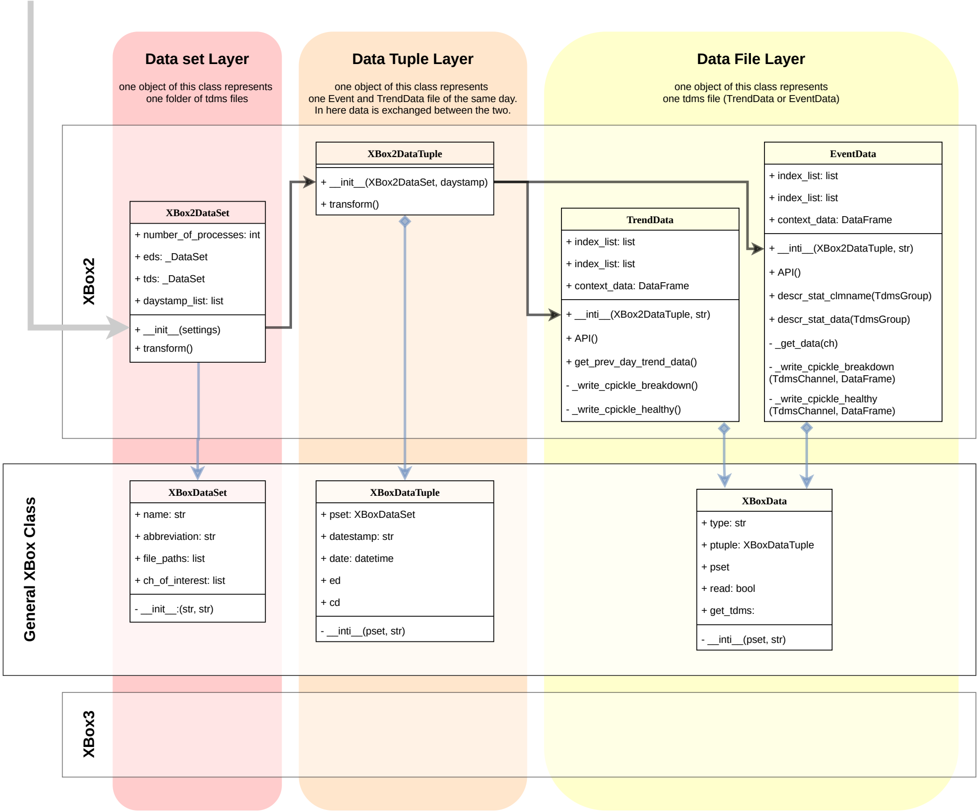
XBoxDataTuple
+ pset: XBoxDataSet
+ datestamp: str
+ date: datetime
+ ed
+ cd
- __inti__(pset, str)

XBoxData
+ type: str
+ ptuple: XBoxDataTuple
+ pset
+ read: bool
+ get_tdms:
- __inti__(pset, str)

XBox3

handle data

machine learning



Data set Layer

one object of this class represents one folder of tdms files

Data Tuple Layer

one object of this class represents one Event and TrendData file of the same day. In here data is exchanged between the two.

Data File Layer

one object of this class represents one tdms file (TrendData or EventData)

XBox2

XBox2DataSet
- + number_of_processes: int
- + eds: _DataSet
- + tds: _DataSet
- + daystamp_list: list
- + __init__(settings)
- + transform()

XBox2DataTuple
- + __init__(XBox2DataSet, daystamp)
- + transform()

TrendData
- + index_list: list
- + index_list: list
- + context_data: DataFrame
- + __inti__(XBox2DataTuple, str)
- + API()
- + get_prev_day_trend_data()
- - _write_cpickle_breakdown()
- - _write_cpickle_healthy()

EventData
- + index_list: list
- + index_list: list
- + context_data: DataFrame
- + __inti__(XBox2DataTuple, str)
- + API()
- + descr_stat_clmname(TdmsGroup)
- + descr_stat_data(TdmsGroup)
- - _get_data(ch)
- - _write_cpickle_breakdown (TdmsChannel, DataFrame)
- - _write_cpickle_healthy (TdmsChannel, DataFrame)

General XBox Class

XBoxDataSet
- + name: str
- + abbreviation: str
- + file_paths: list
- + ch_of_interest: list
- - __init__:(str, str)

XBoxDataTuple
- + pset: XBoxDataSet
- + datestamp: str
- + date: datetime
- + ed
- + cd
- - __inti__(pset, str)

XBoxData
- + type: str
- + ptuple: XBoxDataTuple
- + pset
- + read: bool
- + get_tdms:
- - __inti__(pset, str)

XBox3

Summary

Summary

- pandas DataFrame are easy to use in notebooks

Summary

- `pandas` DataFrame are easy to use in notebooks
- pickle speeds up reading time

Summary

- pandas DataFrame are easy to use in notebooks
- pickle speeds up reading time
- with compression takes up less space

Summary

- pandas DataFrame are easy to use in notebooks
- pickle speeds up reading time
- with compression takes up less space

- version issues with pickle protocol

Summary

- pandas DataFrame are easy to use in notebooks
- pickle speeds up reading time
- with compression takes up less space

- version issues with pickle protocol
- part of the process should not be in the Datahanlder instead of the Transformation

Summary

- pandas DataFrame are easy to use in notebooks
- pickle speeds up reading time
- with compression takes up less space

- version issues with pickle protocol
- part of the process should not be in the Datahanlder instead of the Transformation
- EventData and TrenadData are stored differently

Summary

- `pandas` DataFrame are easy to use in notebooks
- pickle speeds up reading time
- with compression takes up less space

- version issues with pickle protocol
- part of the process should not be in the Datahanlder instead of the Transformation
- EventData and TrenadData are stored differently
- channel properties are lost

Summary

- pandas DataFrame are easy to use in notebooks
- pickle speeds up reading time
- with compression takes up less space

- version issues with pickle protocol
- part of the process should not be in the Datahanlder instead of the Transformation
- EventData and TrenadData are stored differently
- channel properties are lost
- data was changed in place in notebooks in retrospect

Table of Contents

XBox2 File Structure

Event-&TrendData

Choosing a Data Format

reading tdms files

pandas + compressed pickle

Conclusion

Conclusion

Conclusion

- I implemented a generic class for converting `.tdms` files into `pd.df+cpickle`

Conclusion

- I implemented a generic class for converting `.tdms` files into `pd.df+cpickle`
- is there a better data format, maybe `.hdf5`?

Conclusion

- I implemented a generic class for converting `.tdms` files into `pd.df+cpickle`
- is there a better data format, maybe `.hdf5`?

Conclusion

- I implemented a generic class for converting `.tdms` files into `pd.df+cpickle`
- is there a better data format, maybe `.hdf5`?

	`nptdms`	`pd.df+cpickle`		`.hdf5`	
		w/o zip	w zip	w/o zip	w/ zip
space (GByte)	20.5GB	2.8GB	1GB	2.8GB	1GB
read (TD 1 channel)	$\sim$ 60min	4sec	12sec	0.5 sec	
read (TD 3 channels)	$\sim$ 60min	4sec	12sec	1 sec	
read (TD 15 channels)	$\sim$ 60min	4sec	12sec	4 sec	
feature calc. (ED)	$>$ 15min		7sec	8 sec	

CERN

home.cern