
Sinamics CANOpen Documentation

Release 0.1

Bruno Tibério

Nov 12, 2018

CONTENTS:

1	Changelog	3
1.1	Sinamics Class description	3
2	Indices and tables	9
	Python Module Index	11

This documentation describes the class Sinamics developed in Python using CANopen to control the Siemens CU-320-2DP device with one motor module.

Date 24 Aug 2018

Version 0.1

Author Bruno Tibério

Contact bruno.tiberio@tecnico.ulisboa.pt

CHANGELOG

version 0.1 initial release

1.1 Sinamics Class description

class `sinamics.SINAMICS` (*_network=None, debug=False*)

begin (*nodeID, _channel='can0', _bustype='socketcan', objectDictionary=None*)
Initialize SINAMICS device

Configure and setup SINAMICS device.

Parameters

- **nodeID** – Node ID of the device.
- **channel** (*optional*) – Port used for communication. Default can0
- **bustype** (*optional*) – Port type used. Default socketcan.
- **objectDictionary** (*optional*) – Name of EDS file, if any available.

Returns A boolean if all went ok.

Return type bool

changeState (*newState*)
Change SINAMICS state

Change SINAMICS state using controlWord object

To change SINAMICS state, a write to controlWord object is made. The bit change in controlWord is made as shown in the following table:

State	LowByte of Controlword [binary]
shutdown	0xxx x110
switch on	0xxx x111
disable voltage	0xxx xx0x
quick stop	0xxx x01x
disable operation	0xxx 0111
enable operation	0xxx 1111
fault reset	1xxx xxxx

see section 8.1.3 of firmware for more information

Parameters **newState** – string with state witch user want to switch.

Returns boolean if all went ok and no error was received.

Return type bool

checkState()

Check current state of SINAMICS

Ask the StatusWord of SINAMICS and parse it to return the current state of SINAMICS.

State	ID	Statusword [binary]
Start	0	x0xx xxx0 x000 0000
Not Ready to Switch On	1	xxxx xxxx x0xx 0000
Switch on disabled	2	xxxx xxxx x1xx 0000
ready to switch on	3	xxxx xxxx x01x 0001
switched on	4	xxxx xxxx x01x 0011
refresh	5	x1xx xxx1 x010 0011
measure init	6	x1xx xxx1 x011 0011
operation enable	7	xxxx xxxx x01x 0111
quick stop active	8	xxxx xxxx x00x 0111
fault reaction active (disabled)	9	x0xx xxx1 x000 1111
fault reaction active (enabled)	10	x0xx xxx1 x001 1111
Fault	11	xxxx xxxx x0xx 1000

Returns numeric identification of the state or -1 in case of fail.

Return type int

logDebug (*message=None*)

Log a message

A wrap around logging. The log message will have the following structure: [class name : function name] message

the function name will be the caller function retrieved automatically by using `sys._getframe(1).f_code.co_name`

Parameters **message** – a string with the message.

logInfo (*message=None*)

Log a message

A wrap around logging. The log message will have the following structure: [class name : function name] message

Parameters **message** – a string with the message.

printControlWord (*controlword=None*)

Print the meaning of controlword

Check the meaning of current controlword of device or check the meaning of your own controlword. Usefull to check your own controlword before actually sending it to device.

Parameters **controlword** (*optional*) – If None, request the controlword of device.

printCurrentSmoothed()

Print value of smoothed current

printParameter (*parameter=None, isFloat=False*)

Print value of requested SINAMICS parameter.

Request the SINAMICS for the current value of parameter. In CAN, the parameter number, should be converted to hex and added with 0x2000 (for the first drive).

Parameters

- **parameter** – value of Sinamics parameter to be printed.
- **isFloat** – Boolean, if the value to be read is float or not.

printStatusWord()

Print meaning of status word.

See [manual](#) page 30 for meaning of each bit value.

printTorqueSmoothed()

Print value of smoothed torque

printVOFcharFrequency()

Print value of characteristic frequency

printVOFcharVoltage()

Print value of voltage for characteristic frequency

printVOFminVoltage()

Print value of voltage for frequency equal to zero

readControlWord()

Read controlword from device

Returns

A tuple containing:

controlword the current value or None if any error.

Ok A boolean if all went ok or not.

Return type tuple

readObject (*index, subindex*)

Reads an object

Request a read from dictionary object referenced by index and subindex.

Parameters

- **index** – reference of dictionary object index
- **subindex** – reference of dictionary object subindex

Returns message returned by SINAMICS or empty if unsuccessful

Return type bytes

readParameter (*parameter=None*)

Read Sinamics parameter value.

Parameters **parameter** – location to be read.

Returns

A tuple containing:

val the current value or None if any error.

Ok A boolean if all went ok.

Return type tuple

readStatusWord()

Read statusword from device

Returns

A tuple containing:

statusword the current value or None if any error.

Ok A boolean if all went ok or not.

Return type tuple

readVOFcharFrequency()

Read minimum V/F voltage for characteristic frequency.

Returns current value of V/F voltage for characteristic frequency or None if failed

Return type int

readVOFcharVoltage()

Read minimum V/F voltage for characteristic frequency.

Returns current value of V/F voltage for characteristic frequency or None if failed

Return type int

readVOFminVoltage()

Read minimum V/F voltage for frequency equal to zero

Returns current value of V/F voltage for f=0 or None if failed

Return type int

setTargetVelocity(rpm=0)

Set target velocity for sinamics

Parameters **rpm** – velocity in rpms. Must be a signed int32

Returns A boolean if all went ok or not.

setVOFcharFrequency(frequency=None)

Write V/F voltage for characteristic frequency

Returns a boolean if all went ok or not.

Return type bool

setVOFcharVoltage(voltage=None)

Write V/F voltage for characteristic frequency

Returns a boolean if all went ok or not.

Return type bool

setVOFminVoltage(voltage=None)

Write minimum V/F voltage for frequency equal to zero

Returns a boolean if all went ok or not.

Return type bool

writeControlWord(controlword)

Send controlword to device

Parameters **controlword** – word to be sent.

Returns a boolean if all went ok.

Return type bool

writeObject (*index, subindex, data*)

Write an object

Request a write to dictionary object referenced by index and subindex.

Parameters

- **index** – reference of dictionary object index
- **subindex** – reference of dictionary object subindex
- **data** – data to be stored

Returns boolean if all went ok or not

Return type bool

writeParameter (*parameter=None, newData=None, length=2*)

Write Sinamics parameter value

Parameters

- **parameter** – location to be written
- **newData** – value to be written
- **length** – byte length

Returns A boolean if all went ok

Return type bool

INDICES AND TABLES

- `genindex`
- `modindex`
- `search`

PYTHON MODULE INDEX

S

`sinamics`, 3