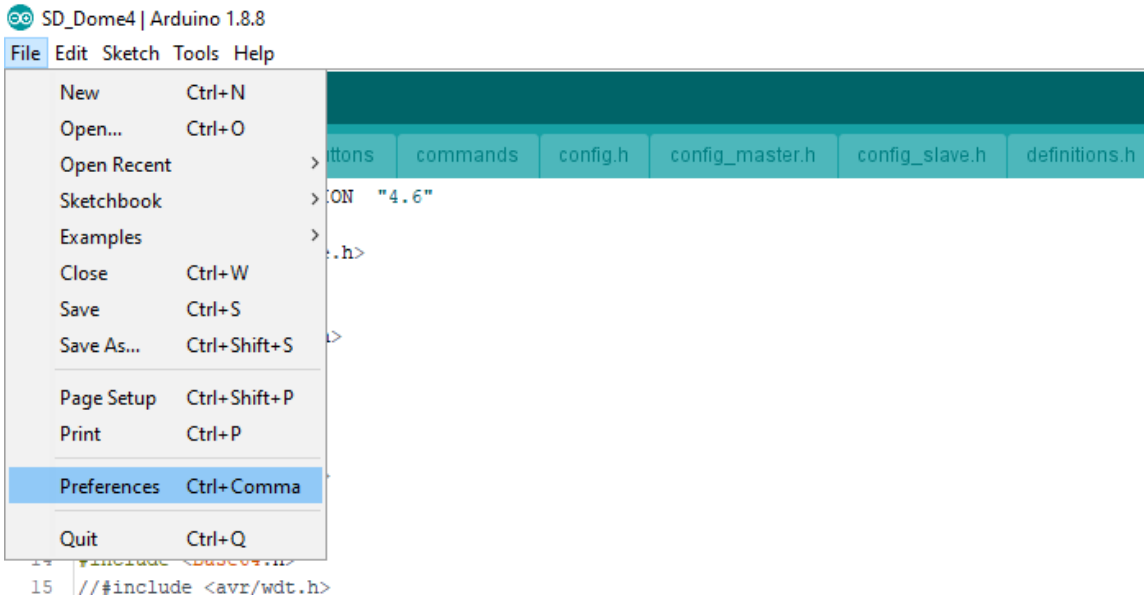
ScopeDome Arduino 5.2

Table of contents

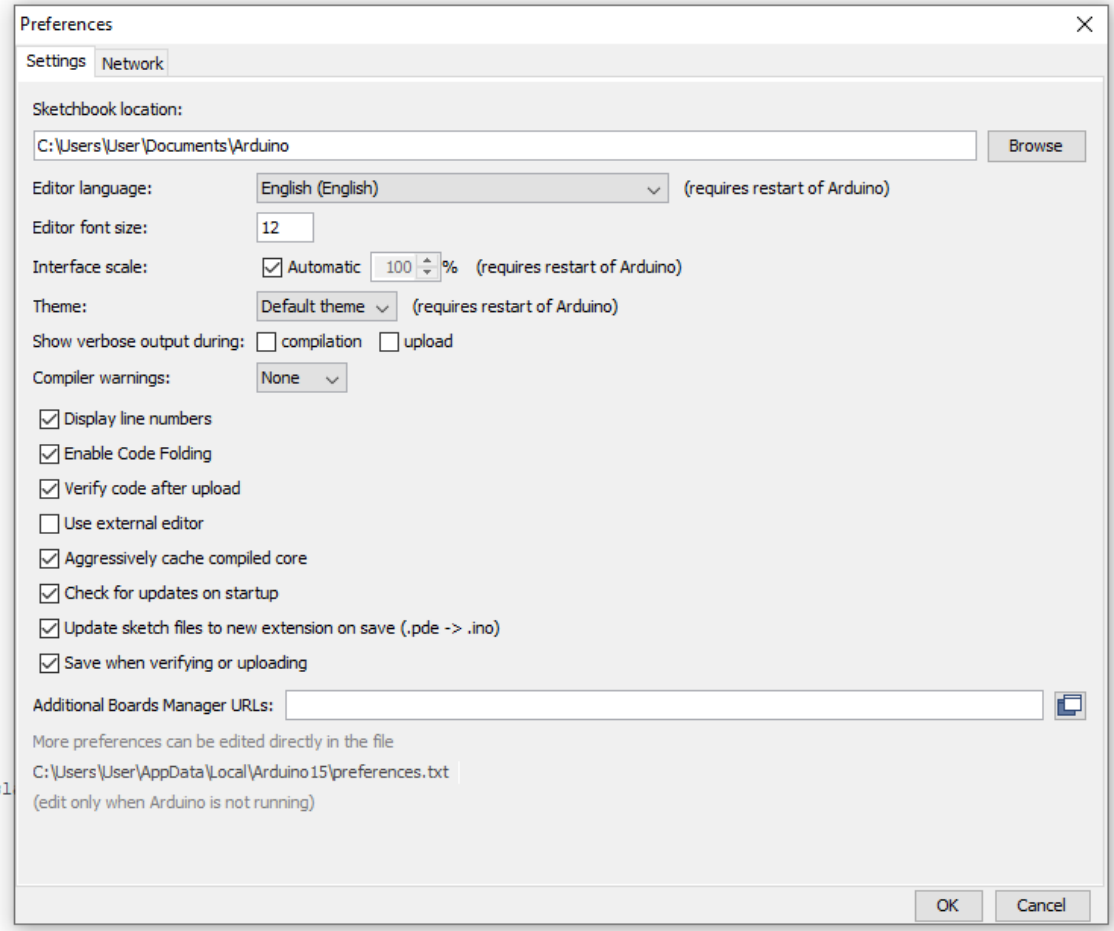
Setting-up Arduino IDE

We recommend to use the latest version of the Arduino IDE available. The current version tested is: 1.8.8.

Before you compile and upload the sketch to your board, make sure that you have placed all the required libraries in the libraries folder. The path to libraries folder can be changed in the [File]→[Preferences] window:



Arduino IDE looks for the libraries in [Sketchbook location]/libraries folder and all the custom libraries used in the sketch have to be present there to be able to compile and upload the sketch to the board.



Pre-setting the devices

All devices should be pre-set before its first use with your dome or roll-off roof. The minimum recommended configuration for each mode is listed below.

We strongly recommend resetting your device after pre-setting with either *reset* command or toggling the power off and on again.

Dome

Master	**Slave**
resetEncoderAPositionCounter clearThermostats clearRelayInputSwitches clearDigitalInputModes clearThermometerToggles clearEmergencys setMode=<master/slave/clamshell> setDomeDelay=<milliseconds> setMac=<MAC address, small letters, no captions> setEth=<true/false> setDhcp=<true/false> setHttpPassword=<password> setIp=<IP address> setLogging=<true/false>	resetEncoderAPositionCounter clearThermostats clearRelayInputSwitches clearDigitalInputModes clearThermometerToggles clearEmergencys setMode=<master/slave/clamshell> setDomeDelay=<milliseconds> setEth=<true/false> setLogging=<true/false>
Optional:	
setEncoderADebounce=<milliseconds> setEncoderATimeout=<milliseconds> setEncoderARange=<integer encoder range> setHomeSignalLow=<true/false> setCloudSensorIndex=<float value> setOpenOnlyOnHome=<true/false> setOpenOnlyOnTelescopeAtHome=<true/false> setEthWatchdog=<true/false>:<(optional) ip>:<(optional) port>:<(optional) relay> setEmergencyShutterTimeout=<seconds> addEmergency=<mode>:<input>:<sample timespan>:<sample triggering number>:<triggering value>	

Example:

Master	**Slave**
resetEncoderAPositionCounter clearThermostats clearRelayInputSwitches clearDigitalInputModes clearThermometerToggles clearEmergencys setMode=master setDomeDelay=2000	resetEncoderAPositionCounter clearThermostats clearRelayInputSwitches clearDigitalInputModes clearThermometerToggles clearEmergencys setMode=slave setDomeDelay=2000

setMac=DA3949D4FC3C setEth=true setDhcp=false setHttpPassword=default setIp=192.168.1.119 setEthWatchdog=0:0.0.0.0:0:0 setLogging=false setEncoderADebounce=10 setEncoderATimeout=1000 setEncoderARange=64000 setHomeSignalLow=true setCloudSensorIndex=10.0 setOpenOnlyOnHome=false setOpenOnlyOnTelescopeAtHome=false	setEth=false setLogging=false setCloudSensorIndex=10.0

Roll off roof

resetEncoderAPositionCounter clearThermostats clearRelayInputSwitches clearDigitalInputModes clearThermometerToggles clearEmergencys setMode=<master/slave/clamshell> setDomeDelay=<milliseconds> setMac=<MAC address, small letters, no captions> setEth=<true/false> setDhcp=<true/false> setHttpPassword=<password> setIp=<IP address> setLogging=<true/false>
Optional:
setEthWatchdog=<true/false>:<(optional) ip>:<(optional) port>:<(optional) relay> setEmergencyShutterTimeout=<seconds> addEmergency=<mode>:<input>:<sample timespan>:<sample triggering number>:<triggering value>

Example:

resetEncoderAPositionCounter clearThermostats clearRelayInputSwitches clearDigitalInputModes clearThermometerToggles clearEmergencys setMode=slave setDomeDelay=2000 setMac=DA3949D4FC3C setEth=1

setDhcp=0
setHttpPassword=default
setIp=192.168.1.119
setLogging=1
setEthWatchdog=1:0.0.0.0:0:0

Clamshell

clearThermostats
clearRelayInputSwitches
clearDigitalInputModes
clearThermometerToggles
clearEmergencys
setMode=<master/slave/clamshell>
setDomeDelay=<milliseconds>
setMac=<MAC address, small letters, no captions>
setEth=<true/false>
setDhcp=<true/false>
setHttpPassword=<password>
setIp=<IP address>
setLogging=<true/false>

Optional:

setEthWatchdog=<true/false>:<(optional) ip>:<(optional) port>:<(optional) relay>
setEmergencyShutterTimeout=<seconds>
addEmergency=<mode>:<input>:<sample timespan>:<sample triggering number>:<triggering value>

Example:

clearThermostats
clearRelayInputSwitches
clearDigitalInputModes
clearThermometerToggles
clearEmergencys
setMode=clamshell
setDomeDelay=2000
setMac=DA3949D4FC3C
setEth=1
setDhcp=0
setHttpPassword=default
setIp=192.168.1.119
setLogging=1
setEthWatchdog=1:0.0.0.0:0:0

Pairing master device and slave device

After pre-setting your dome and shutter devices (i.e. master and slave) they have to be paired to be able to communicate.

To pair the devices:

- Press and release „PAIR" button on your master and slave device at the same time (i.e. within seconds).

OR

- Send ***pair*** command to both master and slave device at the same time (i.e. within seconds).

Calibrating the dome

To calibrate the dome, i.e. to find full dome rotation encoder span:

1. Move your dome out of home position (it shouldn't be at home position).

2. Send command ***calibrate*** to your master device.

The dome should now start to rotate and make at least 360 degrees rotation, after which it should stop close to its home position.

Note: Sending any movement command (e.g. ***moveDome*** or ***moveShutter***) will stop callibrating!

Typical commands

Check *Available commands* section for more details on using commands listed below.

Reading status

To read your device status use command: ***getStatus*** or ***getLocalStatus***

Dome and shutter movement

To open or close your dome use command: ***moveShutter=OPEN*** or ***moveShutter=CLOSE***
To stop your shutter use command: ***stopShutter***

To rotate your dome clockwise or counter-clockwise use command: ***moveDome=CW*** or ***moveDome=CCW***
Optionally add parameter, e.g. ***moveDome=CW:1000***, to rotate the dome by encoder value of 1000.
To stop your dome use command: ***stopDome***

Clamshell movement

To fully open or close your dome use command: ***moveClamshell=OPEN*** or ***moveClamshell=CLOSE***
To stop your shutter use command: ***stopClamshell***

Switching relays

To switch free relay on use command: ***switchOnFreeRelay=<relay index, 1...number of free relays>***
To switch free relay off use command: ***switchOffFreeRelay=<relay index, 1...number of relays>***

If you know what you are doing and want to switch any selected relay on or off use commands:
switchOnRelay=<relay index, 1...number of relays>
switchOffRelay=<relay index, 1...number of relays>

Check *Relays* part of *Available commands* section for more relay commands.

Available commands

Common commands (available in master and slave mode)

Command	Description	Notes
Resetting		
reset	hard reset using reset pin (works only with compatible shield)	Not available on some devices
resetSoft	soft reset	
resetEthernet	hard reset the Ethernet	Not available on some devices
setFresh	setFresh=<value> set "fresh" flag to value 1 or 0	
Diagnostics		
getFirmwareVersion	get firmware version (from #define FIRMWARE_VERSION)	
getDeviceName	get device name (from #define DEVICE_RESPONSE)	
getMode	get device mode master or slave	
getMac	get MAC address of the Ethernet	
getIp	get IP address of the Ethernet	
getFreeRAM	get free RAM size	Returns three measurements in bytes
getRelaysNames	get relays names in "name1:name2:name3:..." format	
getInputsNames	get inputs names in "digital_input_name1:digital_input_name2:digital_input_name3:...;analog_input_name1:analog_input_name2:..." format	
getButtonsNames	get buttons names in "name1:name2:name3:..." format	
getThermometersNames	get thermometers names in "name1:name2:name3:..." format	
Configuration		
setMode	set master or slave mode, e.g. setMode=master	USB only Format: setMode=<master/slave/clamshell>
setMac	set MAC address of the ethernet shield	USB only Format: setMac=DEADBEEFFEED
setIp	set IP for no DHCP mode	USB only Format: setIp=123.123.123.123
setLogging	turn logging on serial on or off,	USB only

	e.g. setLogging=true	Format: setLogging=<true/1/false/0>
setEth	set if should use Ethernet, e.g. setEth=true	USB only Format: setEth=<true/1/false/0>
setDhcp	set if should use DHCP, e.g. setDhcp=true	USB only Format: setDhcp=<true/1/false/0>
setEthWatchdog	set if the device should use Ethernet watchdog, which has two modes: **MODE 1)** *setEthWatchdog=1* resets the Ethernet when there is no ethernet commands for a long time **MODE 2)** *setEthWatchdog=1:<ip>:<port>:<relay>* resets the Ethernet when there is no ethernet commands for a long time and can't set a connection to ip <ip> on port <port>; if <relay> != 0 watchdog also toggles this relay for 1 sec.	Format: setEthWatchdog=<true/1/false/0> OR setEthWatchdog=<true/1/false/0>:<remote ip to check connection>:<remote port>:<relay to reset (optional)>
setDomeDelay	set dome delay (for changing movement direction)	Format: setDomeDelay=<delay in millis>
resetEncoderAPositionCounter	reset EEPROM encoder A calibration: set counter to 0 and encoder A position to initial value 32000	
setEncoderADebounce	set debounce time for encoder A	Format: setEncoderADebounce=<debounce time in millis>
setEncoderATimeout	set timeout for encoder A (moveDome has to change encoder in that timeout)	Format: setEncoderATimeout=<timeout in millis>
setEncoderARange	set range for encoder A	Format: setEncoderARange=<range>
setEncoderA	set encoder A counter to <integer value>	Format: setEncoderA=<integer value>
setHomeSignalLow	set to 1 if home signal is low, 0 if home signal is high, e.g. setHomeSignalLow=true	Format: setHomeSignalLow=<true/1/false/0>
setCloudSensorIndex	set cloud sensor index (i.e. temperature difference)	Format: setCloudSensorIndex=<float>
setHttpPassword	set password for HTTP authentication	Format: setHttpPassword=<password>
clearDigitalInputModes	clears (resets) all digital inputs modes to normal (not opposite)	
setDigitalInputMode	set the digital input mode to normal (0) or reverted/opposite	Format: setInputMode=<digital input index (from 1 to the number of digital

	(1); opposite means HIGH becomes LOW and vice versa	inputs)>:<0/1>
clearThermometerToggles	clears (resets) all thermometer toggles	
setThermometerToggle	set thermometer toggle to on (1) or off (0)	Format: setThermometerToggle=<thermometer index (from 1 to the number of thermometers)>:<0/1>
clearEmergencys	clears (resets) all emergency triggers	
addEmergency	adds new emergency trigger *MASTER only:* if input is negative, slave input of abs(input) index is checked	Format: addEmergency=<DIGITAL/ANALOG/CLOUD/NONE>:<input [for master in master+slave configuration can be negative to check slave input]>:<sample timespan in milliseconds>:<sample triggering number>:<triggering value [for ANALOG: negative if less than abs(value), positive if more than value]>
saveEmergency	update existing or add new emergency trigger *MASTER only:* if input is negative, slave input of abs(input) index is checked	Format: saveEmergency=<emergency index from 1 to ...>:<DIGITAL/ANALOG/CLOUD/NONE>:<input [for master in master+slave configuration can be negative to check slave input]>:<sample timespan in milliseconds>:<sample triggering number>:<triggering value [for ANALOG: negative if less than abs(value), positive if more than value]>
setEmergencyShutterTimeout	set emergency shutter closing timeout	Format: setEmergencyShutterTimeout=<timeout in **seconds**>
pair	start pairing with second device	
Thermostats		
clearThermostats	clears (resets) all thermostats	
setThermostat	set a thermostat	Format: setThermostat=<thermometer index (from 1 to the number of thermometers)>:<enabled=true/1/false/0>:<mode=HEATER/COOLER/NONE>:<desiredTemperature (as float, e.g. 29.0)>:<width (as float, e.g. 2.0)>:<relay index (from 1 to m)>:<(optional) thermometer index (from 1 to the number of thermometers>
Status		
getStatus	get status of:	**In master mode:**

	- master and slave (in dome configuration) - slave only (in roll off roof configuration)	get master+slave status in format: <master digital inputs>;<master analog inputs>;<master encoders>;<master temperatures>;<master+slave clouds>;<master Vcc>;<master buttons>;<master relays>;<master loop time>;<master fresh flag>#<slave digital inputs>;<slave analog inputs>;<slave encoders>;<slave temperatures>;<slave clouds>;<slave Vcc>;<slave buttons>;<slave relays>;<slave loop time>;<slave fresh flag>#<flags> **In slave and clamshell mode:** get slave or clamshell status in format: <slave/clamshell digital inputs>;<slave/clamshell analog inputs>;<slave/clamshell encoders>;<slave/clamshell temperatures>;<slave/clamshell clouds>;<slave/clamshell Vcc>;<slave/clamshell buttons>;<slave/clamshell relays>;<slave/clamshell loop time>;<slave/clamshell fresh flag>
getLocalStatus	get local status (without the slave connected device status)	Format: <digital inputs>;<analog inputs>;<encoders>;<temperatures>;<Vcc>;<buttons>;<relays>;<loop time>;<fresh flag>
getRelays	get relays status in "1:0:0:1:0.5" format	Float values stands for PWM relays
getButtons	get buttons status in "1:1:0:..." format	
getInputs	get digital and analog inputs status in <digital inputs>;<analog inputs> format e.g. "1:1:1:0:...;1.0:0.5:0.1:..."	
getInputDs	get digital inputs status in "1:1:1:0:..." format	
getInputAs	get analog inputs status in "1:1:1:0:..." format	
getTemperatures	get thermometers value in "1.0:2.0:3.0:4.0:..." format	First and secondary measured thermometers values one by one
getInputsAndEncoder	get input and counter statuses "1:1:1:0:...:323"	
getCloudy	get info about clouds: returns 0 = not cloudy; 1 = cloudy; -1 = unknown	-1 never returned. Default value (with no inputs connected) is 0 = “not cloudy”.
Relays		
switchOnFreeRelay	switch ON free relay; <relay> index from 1	Format: switchOnFreeRelay=<relay>:<(optional)

		analog value>
switchOffFreeRelay	switch OFF free relay; <relay> index from 1	Format: switchOffFreeRelay=<relay>
switchRelayWithInput	switch on/off relay whenever input is on/off (or opposite); send input=0 for turning off	Format: switchRelayWithInput=<save=true/1/false/0>:<relay>:<input>:<(optional) opposite=true/1/false/0>
clearRelayInputSwitches	clears (resets) all RelayInputSwitches	
switchOnRelay	switch on relay nr <relay> (from 1 to n)	Only for firmware or special use Format: switchOnRelay=<relay>:<(optional) analog value>
switchOffRelay	switch off relay nr <relay> (from 1 to n)	Only for firmware or special use Format: switchOffRelay=<relay>
switchOnRelayTillInput	switch on relay till input is on (switches on relay and turns it off when given input is on)	Only for firmware or special use Format: switchOnRelayTillInput=<relay>:<input>:<(optional) analog value>
switchOnRelayTillTimeout	switch on relay till timeout	Only for firmware or special use Format: switchOnRelayTillTimeout=<relay>:<timeout>:<(optional) analog value>
switchOnRelayTillInputOrTimeout	switch on relay till input is on or timeout passed	Only for firmware or special use Format: switchOnRelayTillInputOrTimeout=<relay>:<input>:<timeout>:<(optional) analog value>
switchOnRelayTillEncoder	switch on relay till encoder counter counted to <count>	Only for firmware or special use Format: switchOnRelayTillEncoder=<relay>:<count>:<(optional) analog value>
Movement		
moveShutter	shutter movement; <direction> is 'OPEN' or 'CLOSE', <timeout> and <release timeout> in milliseconds are optional	Format: moveShutter=<direction>:<(optional) timeout>:<(optional) release timeout>
stopShutter	stop shutter movement	
setOpenOnlyOnHome	set if slave should open only if it's on home	Format: setOpenOnlyOnHome=<true/1/false/0>
setOpenOnlyOnTelescopeAtHome	set if slave should open only if it's telescope is at home (at park)	Format: setOpenOnlyOnTelescopeAtHome=<true/1/false/0>
Other		
emergencyStop	in case of emergency turns off all the relays	

Master commands (available in master mode only)

Command	Description	Notes
Diagnostics		
getCalibratedRotation	get counted rotation span	
Movement		
moveDome	dome movement; <direction> is 'CW' or 'CCW', <encoder> is optional (relative value)	Format: moveDome=<direction>:<encoder>
stopDome	stop dome movement	
Other		
findHome	find home	Start finding home, the direction in which to search is automatically calculated
calibrate	calibrate (count encoder during rotation)	Start finding home-to-home encoder span (full dome rotation encoder span), the direction in which to search for home is automatically calculated
slave	send command to the slave device	Format: slave=<command>

Clamshell commands (available in clamshell mode only)

Command	Description	Notes
Movement		
moveClamshell	clamshell dome movement; <direction> is 'OPEN' or 'CLOSE', <shutter> is optional (the default is "0" for full dome open or close), timeouts in milliseconds are optional	Format: moveClamshell=<direction>:<(optional) shutter>:<(optional) timeout>:<(optional) release timeout>
stopClamshell	stop clamshell dome movement	

Status format

Note:
...INPD... = digital input
...INPA... = analog input
...REL... = relay

Master mode

In **master** mode the getStatus command returns device status in format:

<master digital inputs>;<master analog inputs>;<master encoders>;<master temperatures>;<master other instruments>;<master clouds check (taking slave clouds check into account)>;<master Vcc>;<master buttons>;<master relays>;<master emergency flag>;<master loop time>;<master fresh flag>#<slave digital inputs>;<slave analog inputs>;<slave encoders>;<slave temperatures>;<slave other instruments>;<slave clouds check>;<slave Vcc>;<slave buttons>;<slave relays>;<slave emergency flag>;<slave loop time>;<slave fresh flag>#<other>

i.e.: (new lines added for readability)

```
MASTER_INPD_DETECT_230LOSS:
MASTER_INPD_ENCODERA:
MASTER_INPD_HOMESENSOR:
MASTER_INPD_FREE1:
MASTER_INPD_FREE2:
MASTER_INPD_RAINSENSOR:
MASTER_INPD_CLOUDSENSOR:
MASTER_INPD_TELESCOPE_A_H;

MASTER_INPA_VOLTAGE064:
MASTER_INPA_T_PT100:
MASTER_INPA_T_PCB;

MASTER_ENCODER_A;

MASTER_THERMOMETER_PCB:
MASTER_BAROMETER_TEMPERATURE:
MASTER_THERMOMETER_ ONEWIRE_MOTOR:
MASTER_THERMOMETER_ ONEWIRE_OUTSIDE:
MASTER_THERMOMETER_ ONEWIRE_MIRROR_1:
MASTER_THERMOMETER_ ONEWIRE_MIRROR_2:
MASTER_HIGROMETER_TEMPERATURE:
MASTER_PIROMETER_AMBIENT:
MASTER_PIROMETER_SENSOR;

MASTER_BAROMETER_PRESSURE:
MASTER_HIGROMETER_HUMIDITY;

MASTER_CLOUDS;

MASTER_Vcc;

MASTER_BUTTON_CW:
MASTER_BUTTON_CCW:
MASTER_BUTTON_FREE1:
```

```
MASTER_BUTTON_FREE2:
MASTER_BUTTON_FREE3:
MASTER_BUTTON_PAIR;

MASTER_REL_CW:
MASTER_REL_CCW:
MASTER_REL_INBOX:
MASTER_REL_MOTOR:
MASTER_PWM_1:
MASTER_PWM_2:
MASTER_REL_FREE1:
MASTER_REL_FREE2:
MASTER_REL_FREE3;

MASTER_EMERGENCY;

MASTER_LOOP_TIME;

MASTER_FRESH_FLAG

#

SLAVE_INPD_DETECT_230LOSS:
SLAVE_INPD_ENCODERA:
SLAVE_INPD_HOMESENSOR:
SLAVE_INPD_OPEN1:
SLAVE_INPD_CLOSE1:
SLAVE_INPD_RAINSENSOR:
SLAVE_INPD_CLOUDSENSOR:
SLAVE_INPD_TELESCOPE_A_H;

SLAVE_INPA_VOLTAGE064:
SLAVE_INPA_T_PT100:
SLAVE_INPA_T_PCB;

SLAVE_ENCODER_A;

SLAVE_THERMOMETER_PCB:
SLAVE_BAROMETER_TEMPERATURE:
SLAVE_THERMOMETER_ONEWIRE_MOTOR:
SLAVE_THERMOMETER_ONEWIRE_OUTSIDE:
SLAVE_THERMOMETER_ONEWIRE_MIRROR_1:
SLAVE_THERMOMETER_ONEWIRE_MIRROR_2:
SLAVE_HIGROMETER_TEMPERATURE:
SLAVE_PIROMETER_AMBIENT:
SLAVE_PIROMETER_SENSOR;

SLAVE_BAROMETER_PRESSURE:
SLAVE_HIGROMETER_HUMIDITY;

SLAVE_CLOUDS;

SLAVE_Vcc;

SLAVE_BUTTON_CW:
SLAVE_BUTTON_CCW:
SLAVE_BUTTON_FREE1:
SLAVE_BUTTON_FREE2:
```

```
SLAVE_BUTTON_FREE3:
SLAVE_BUTTON_PAIR;

SLAVE_REL_OPEN1:
SLAVE_REL_CLOSE1:
SLAVE_REL_INBOX:
SLAVE_REL_MOTOR:
SLAVE_PWM_1:
SLAVE_PWM_2:
SLAVE_REL_FREE1:
SLAVE_REL_FREE2:
SLAVE_REL_FREE3;

SLAVE_EMERGENCY;

SLAVE_LOOP_TIME;

SLAVE_FRESH_FLAG

#

IS_SLAVE_ONLINE:
IS_COMPLEX_COMMAND_FINDING_HOME:
IS_COMPLEX_COMMAND_MOVE_SHUTTER_ON_HOME:
IS_COMPLEX_COMMAND_CALIBRATING
```

Slave mode

In **slave** mode the getStatus command returns device status in format:

`<slave digital inputs>;<slave analog inputs>;<slave encoders>;<slave temperatures>;<slave other instruments>;<slave clouds check>;<slave Vcc>;<slave buttons>;<slave relays>;<slave emergency flag>;<slave loop time>;<slave fresh flag>`

i.e.: (new lines added for readability)

```
SLAVE_INPD_DETECT_230LOSS:
SLAVE_INPD_ENCODERA:
SLAVE_INPD_HOMESENSOR:
SLAVE_INPD_OPEN1:
SLAVE_INPD_CLOSE1:
SLAVE_INPD_RAINSENSOR:
SLAVE_INPD_CLOUDSENSOR:
SLAVE_INPD_TELESCOPE_A_H;

SLAVE_INPA_VOLTAGE064:
SLAVE_INPA_T_PT100:
SLAVE_INPA_T_PCB;

SLAVE_ENCODER_A;

SLAVE_THERMOMETER_PCB:
SLAVE_BAROMETER_TEMPERATURE:
SLAVE_THERMOMETER_ONEWIRE_MOTOR:
SLAVE_THERMOMETER_ONEWIRE_OUTSIDE:
SLAVE_THERMOMETER_ONEWIRE_MIRROR_1:
SLAVE_THERMOMETER_ONEWIRE_MIRROR_2:
```

```
SLAVE_HIGROMETER_TEMPERATURE:
SLAVE_PIROMETER_AMBIENT:
SLAVE_PIROMETER_SENSOR;

SLAVE_BAROMETER_PRESSURE:
SLAVE_HIGROMETER_HUMIDITY;

SLAVE_CLOUDS;

SLAVE_Vcc;

SLAVE_BUTTON_CW:
SLAVE_BUTTON_CCW:
SLAVE_BUTTON_FREE1:
SLAVE_BUTTON_FREE2:
SLAVE_BUTTON_FREE3:
SLAVE_BUTTON_PAIR;

SLAVE_REL_OPEN1:
SLAVE_REL_CLOSE1:
SLAVE_REL_INBOX:
SLAVE_REL_MOTOR:
SLAVE_PWM_1:
SLAVE_PWM_2:
SLAVE_REL_FREE1:
SLAVE_REL_FREE2:
SLAVE_REL_FREE3;

SLAVE_EMERGENCY;

SLAVE_LOOP_TIME;

SLAVE_FRESH_FLAG
```

Clamshell mode

In **clamshell** mode the getStatus command returns device status in format:

`<clamshell digital inputs>;< clamshell analog inputs>;< clamshell encoders>;< clamshell temperatures>;< clamshell other instruments>;<clamshell clouds check>;< clamshell Vcc>;<clamshell buttons>;<clamshell relays>;<clamshell emergency flag>;<clamshell loop time>;<clamshell fresh flag>`

i.e.: (new lines added for readability)

```
CLAMSHELL_INPD_DETECT_230LOSS:
CLAMSHELL_INPD_SE_C_3:
CLAMSHELL_INPD_SE_O_3:
CLAMSHELL_INPD_SE_O_1:
CLAMSHELL_INPD_SE_C_1:
CLAMSHELL_INPD_RAINSENSOR:
CLAMSHELL_INPD_CLOUDSENSOR:
CLAMSHELL_INPD_TELESCOPE_A_H:
CLAMSHELL_INPD_SE_C_2:
CLAMSHELL_INPD_SE_O_2;

CLAMSHELL_INPA_VOLTAGE064:
```

CLAMSHELL_INPA_DUMMY:
CLAMSHELL_INPA_T_PCB;

CLAMSHELL_ENCODER_A;

CLAMSHELL_THERMOMETER_PCB:
CLAMSHELL_BAROMETER_TEMPERATURE:
CLAMSHELL_THERMOMETER_ONEWIRE_MOTOR:
CLAMSHELL_THERMOMETER_ONEWIRE_OUTSIDE:
CLAMSHELL_THERMOMETER_ONEWIRE_DUMMY:
CLAMSHELL_THERMOMETER_ONEWIRE_MIRROR_2:
CLAMSHELL_HIGROMETER_TEMPERATURE:
CLAMSHELL_PIROMETER_AMBIENT:
CLAMSHELL_PIROMETER_SENSOR;

CLAMSHELL_BAROMETER_PRESSURE:
CLAMSHELL_HIGROMETER_HUMIDITY;

CLAMSHELL_CLOUDS;

CLAMSHELL_Vcc;

CLAMSHELL_BUTTON_OPEN:
CLAMSHELL_BUTTON_CLOSE:
CLAMSHELL_BUTTON_FREE1:
CLAMSHELL_BUTTON_FREE2:
CLAMSHELL_BUTTON_FREE3:
CLAMSHELL_BUTTON_PAIR;

CLAMSHELL_REL_OPEN1:
CLAMSHELL_REL_CLOSE1:
CLAMSHELL_REL_INBOX:
CLAMSHELL_REL_MOTOR:
CLAMSHELL_PWM_1:
CLAMSHELL_PWM_2:
CLAMSHELL_REL_S2:
CLAMSHELL_REL_S1:
CLAMSHELL_REL_S3;

CLAMSHELL_EMERGENCY;

CLAMSHELL_LOOP_TIME;

CLAMSHELL_FRESH_FLAG

Notes

[MASTER/SLAVE/CLAMSHELL]_EMERGENCY is a number indicating the type of emergency:

- 1 – DIGITAL
- 2 – ANALOG
- 3 - CLOUD

Button shortcuts

- During start-up:
  [CW] + [CCW] OR **[OPEN] + [CLOSE]**
  logging parameter set to 0

- After start-up (while running):
  [CW] + [CCW] + [FREE] OR **[OPEN] + [CLOSE] + [FREE]**
  start pairing in master/slave

LEDs

- **[BLUE] blinking**
  Ethernet is initializing or is being re-initialized by watchdog.
- **[BLUE] on**
  Ethernet communication detected.
- **[RED] on**
  Voltage loss.
- **[GREEN] blinking**
  Checking Ethernet.
- **[GREEN] on**
  Complex command is executing.

Watchdogs

There are two types of watchdogs being used in the devices: main loop watchdog and Ethernet watchdog.

- **Main loop watchdog**
  Default watchdog that checks if the device is frozen and if so – performs software reset. Maximum loop time is 5 seconds.

- **Ethernet watchdog** (optional)
  Optional watchdog, that can be enabled with the one of two commands:
  MODE 1): *setEthWatchdog=1*
  MODE 2): *setEthWatchdog=1:<ip>:<port>:<(optional) relay>*
  and can be disabled with the command *setEthWatchdog=0*
  In both modes the watchdog starts checking the Ethernet communication after the first successful Ethernet command receipt.
  It checks the correctness of the Ethernet communication differently in each mode by:
  MODE 1): checking if there was at least one Ethernet command in the last 60 seconds,
  MODE 2): checking if there was at least one Ethernet command in the last 60 seconds and if not – trying to connect to a given remote IP/port (if the connection is successful, no reset action is performed).
  If the check is unsuccessful the Ethernet module is re-initialized and optionally in MODE 2) the given relay is switched on and off (1 second HIGH, 1 second LOW). The device itself is not reset.
  The relay should control the power of the local Ethernet router/switch/etc. (note: NOT the power of the device itself!).

Web interface

ScopeDome Arduino 5.2 has a simple web interface built in. To use it, please open URL: http://[your card ip]