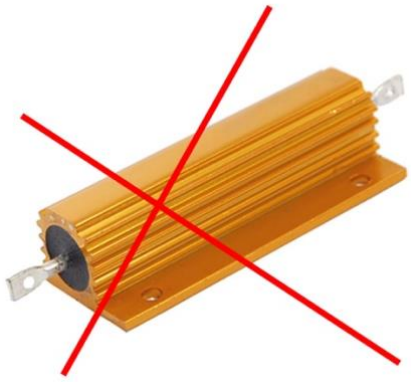
SDC-50A diode controller

User manual

Important note. Please measure output with adequate load only (diodes). Resistive load connected to the output won't destroy the driver, but will severe distort its behavior.	
	

Overview

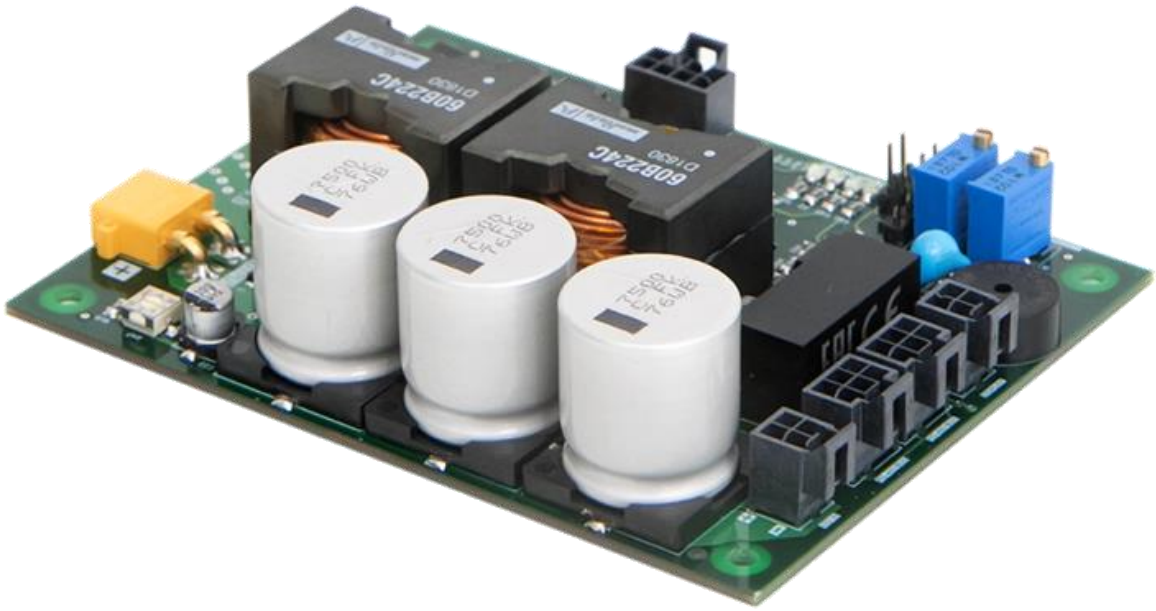
SDC-50A is a pulsed diode driver specially designed for diode pumping of Nd:YAG and similar lasers. The output current is up to 50A in a base modification. There is a powerful Peltier controller onboard.

Module can be controlled either digitally via RS-485 interface or manually (using jumpers and trimpots).

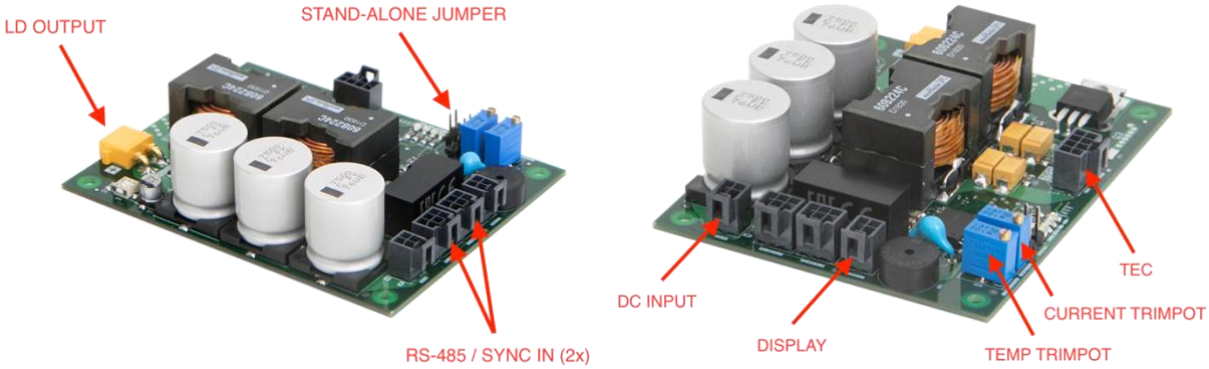
Main parameters are the following:

- Input – 12-15V DC
- Output current – up to 50A
- Compliance voltage – 4-6V (in dependence on input voltage)
- Pulse width – up to 500us
- Repetition rate – up to 50Hz
- Risetime/falltime – <25us/15us respectively

Appearance



Connections, signals, signal descriptions



TYPE	DESIGNATION	DESCRIPTION
Connector	**DC INPUT**	Power input (12-15V DC)
Connector	**LD OUTPUT**	Pulsed output to the laser diode
Connector	**TEC**	Peltier and NTC connection, aux. NTC connection
Connector	**RS-485 / SYNC IN**	RS-485 interface Synchro input (in external synchronization mode)
Connector	**DISPLAY**	External LCD can be connected here
Trimpot	**CURRENT TRIMPOT**	Sets output current if STAND-ALONE JUMPER is on
Trimpot	**TEMP TRIMPOT**	Sets diode temperature if STAND-ALONE JUMPER is on
Jumper	**STAND-ALONE JUMPER**	Defines the way output current and diode temperature are set

DC INPUT: Molex Nano-Fit (105310-1204)

PIN (color)	DESIGNATION	DESCRIPTION
1, 2 (black)	**GND**	12-15V DC input voltage is to be applied here
3, 4 (red)	**DC INPUT**	

LD OUTPUT: Amass (XT30PW-M)

PIN (color)	DESIGNATION	DESCRIPTION
1 (red)	**LD+**	Laser diode positive (anode)
2 (black)	**LD-**	Laser diode negative (cathode)

TEC: Molex Nano-Fit (105310-1208)

PIN (color)	DESIGNATION	DESCRIPTION
1, 2 (black)	**TEC-**	Peltier connection (negative). Both pins are interconnected in parallel.
3, 7 (violet)	**NTC**	10kOhm NTC connection
4, 8 (blue)	**NTC-AUX**	Reserved for the future use
5, 6 (red)	**TEC+**	Peltier connection (positive). Both pins are interconnected in parallel.

RS-485/SYNC IN: Molex Nano-Fit (105310-1206) – **2PCS**

(two identical connectors are connected in parallel, which makes serial connection of several drivers possible)

PIN (color)	DESIGNATION	DESCRIPTION
1 (orange)	**+5V DC**	5V DC voltage powering RS-485 interface to be applied here.
2 (white)	**SYNC IN**	Incoming synchronization pulses should be applied to this pin if controller is run in external synchronization mode.
3, 5 (black)	**GND**	Return of RS-485 interface and synchronization signals.
4 (green)	**B**	Signals of RS-485 interface.
6 (violet)	**A**	

DISPLAY: Molex Nano-Fit (105310-1204)

PIN (color)	DESIGNATION	DESCRIPTION
1	GND	TTL Serial display might be connected here in order to provide graphical UI in stand-alone applications (if interested, please contact us for further details).
2	TX	
3	+5V DC	
4	RX	

STAND-ALONE JUMPER:

When *STAND-ALONE JUMPER* is ON, SDC-50A doesn't need active RS-485 connection for the operations and works as a stand-alone device. In this case the output parameters are set either from memory or by *CURRENT TRIMPOT* and *TEMP TRIMPOT* (see also *Modes of operations* section).

CURRENT TRIMPOT and TEMP TRIMPOT:

When *STAND-ALONE JUMPER* is ON and parameters are set accordingly to trimpots (see also software description), *CURRENT TRIMPOT* defines the output pulse current and *TEMP TRIMPOT* defines the TEC set point.

LEDS:

There are several LEDs indicating state of SDC-50A board.

RS-485 LED (green):

- blinks when the device sends data via RS-485 interface

FAULT LED (red):

- lights steadily if any of Fault conditions is met

TEC LED (green):

- lights steadily if TEC is turned on

AUX LED (blue):

- lights if TEC is turned on and thermal stabilization is achieved

Grounding policy

Most of SDC-50A circuits have common ground.

Only RS-485 interface (including SYNC IN signal) is optically isolated from other circuits.

Modes of operations

RS-485 mode and Stand-alone mode

SDC-50A has two control modes – RS-485 mode и Stand-alone mode:

- RS-485 mode – requires the active RS-485 connection and all the controls are performed by commands sent via RS-485 interface. To operate in this mode Stand-alone jumper should be removed (OFF).
- Stand-Alone mode – doesn't require the active RS-485 connection and SDC-50A starts the operations immediately after 12-15V DC power is applied to the board. To operate in this mode Stand-alone jumper should be set (ON).

Switching between RS-485 mode and Stand-Alone mode is not trivial. Please follow the instructions below.

RS-485 mode to Stand-alone mode

1. Remove Stand-alone jumper (OFF).
2. Apply power to SDC-50A.
3. Establish RS-485 connection between controlling device and SDC-50A.
   Further description supposes, that SDC-50A is connected to PC and controlled via software, although the same can be done from customer's controlling device by sending RS-485 command.
4. Set up the essential parameters and regimes of SDC-50A, e.g.:
   4.1. TEC temperature
   4.2. output current
   4.3. pulse width
   4.4. regimes of operations (see *Software description* and *RS-485 protocol description* sections for the details).
5. Check or uncheck **Params from memory** check box accordingly to your wishes.
6. Save parameters with **SaveParam** button.
7. Remove power from SDC-50A.
8. Install Stand-alone jumper (ON).
9. Apply power to SDC-50A – driver will operate with the saved parameters.

Stand-alone mode to RS-485 mode

1. Remove power from SDC-50A board.
2. Remove Stand-alone jumper (OFF).
3. Connect SDC-50A to the controlling device (PC or another one).
4. Apply power to SDC-50A.
5. Establish RS-485 connection between controlling device and SDC-50A.

Specifications

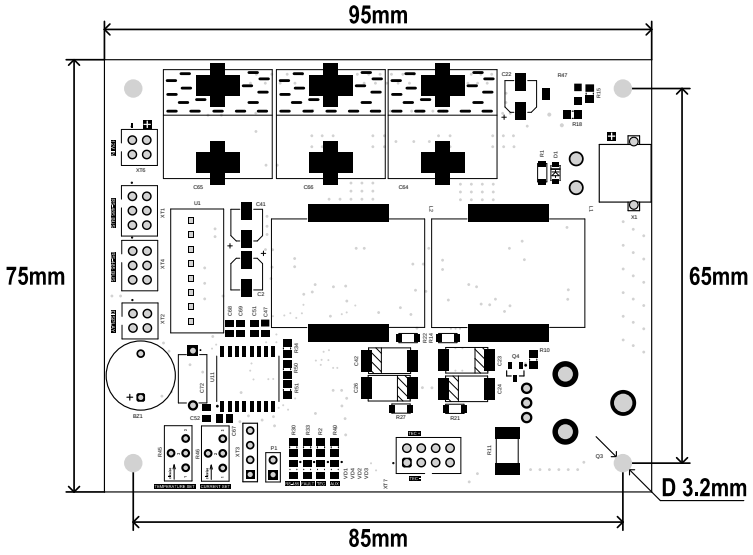
ELECTRICAL SPECIFICATION

INPUT	
Voltage	12-15V DC
Current	6A max
LASER DIODE OUTPUT	
Maximal output current (I_{MAX})	50A
Maximal output voltage (V_{MAX})	4V/6V with 12V DC/15V DC input respectively
Operating regime	Pulsed
Typical pulse width	Up to 500us (other on request)
Typical repetition rate	Up to 50Hz (other on request)
Risetime / falltime	<25us / <15us
Accuracy	<1%
Stability	<1%
TEC OUTPUT	
Max. current / voltage	>5A / 10V
Temperature range	10-40C
Accuracy/Stability	0.1C
Feedback loop	NTC 10kOhm
ENVIRONMENT	
Operating temperature	+10…+40C
Storage temperature	-20…+60C

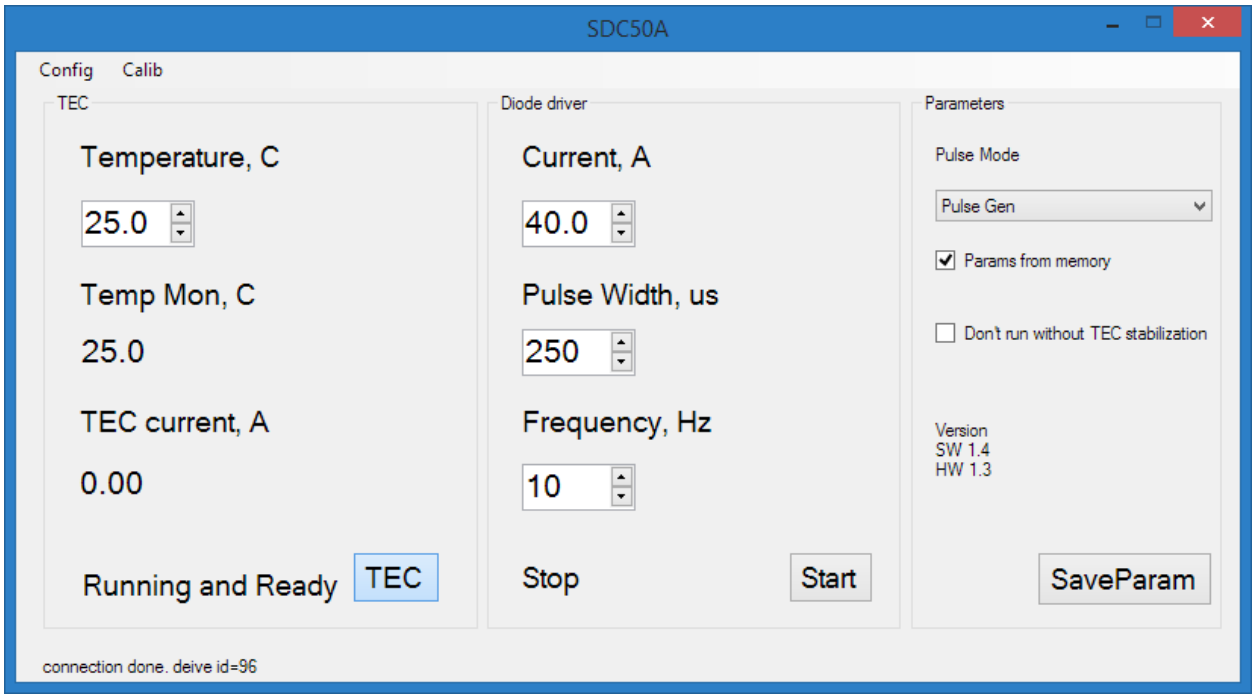
MECHANICAL SPECIFICATION

Size (LxWxH)	95x75x30 mm (see also the dimensional drawing below)
Weight	0.1 kg

DIMENSIONAL DRAWING



Software description



TEC section:

- **Temperature** – temperature set point (10-40°C)
- **Temp Mon** – real temperature measured with NTC
- **TEC Current** – instant current consumed by TEC from 12V DC (A)
- Status of TEC – possible statuses are **Running, Running and Ready, Stop**, **No connection**
- **TEC** button – starts/stops TEC

Diode driver section:

- **Current** – sets pulse current (0-50A)
- **Pulse width** – sets pulse width (50-500us)
- **Frequency** – sets repetition rate (1-50Hz)
- Status of Diode driver – possible statuses are **Running**, **Stop**, **No connection**
- **START** button – starts/stops diode driver
  (please note the driver will not start until TEC is on)

Parameters section:

- **Pulse mode**:
  - **Pulse Gen** – internal synchronization mode, pulses are generated following the internal clock
  - **External pulse width** – external synchronization mode, the duration of the output pulse is equal to the duration of the input pulse, but not more than 500us (regardless of the value set in Pulse Width field).
  - **Internal pulse width** – external synchronization mode, the duration of the output pulse is equal to the value set in Pulse Width field (regardless of the duration of the input pulse).
- **Params from memory**:

    - If checked (ON) – all parameters: **Temperature**, **Current**, **Pulse width**, **Pulse mode** and **Don't run without TEC stabilization** are taken from the internal memory of SDC-50A.
    - If unchecked (OFF) – **Temperature** and **Current** are set with corresponding trimpots. **Pulse width**, **Pulse mode** and **Don't run without TEC stabilization** are still taken from the internal memory of the device.
- **Don't run without TEC stabilization**:
    - If checked (ON) – current pulses appear at LD OUTPUT only after temperature stabilization.
    - If unchecked (OFF) – current pulses appear at LD OUTPUT immediately once diode driver is on, regardless of whether the temperature stabilization is achieved.

SaveParam button:

- Saves all the parameters to internal memory of the device.

Versions:

- SW – software version
- FW – firmware version

RS-485 protocol description

Physical level:

- RS-485, 115200

Logical level:

- Based on «request-response» architecture
- SDC-50A doesn't generate any messages without a request received
- Response is generated for the every request received

Command structure (requests):

```
typedef struct
{
  uint8_t head;
  uint8_t device_ID;
  uint8_t cmd_ID;
  int16_t set_val;
  int16_t get_val;
  uint8_t reserved[4];
  uint8_t tail[3];
} command;
```

head – command header – 1 byte (0x72)

device_ID – receiver identifier – 1 byte

cmd_ID – command identifier – 1 byte

set_val – parameters – 2 bytes

get_val – standardly not used in requests – 2 bytes

reserved – standardly not used in requests – 4 bytes

tail – command end mark – 3 bytes (all 0xFF)

Command structure (responses):

```
typedef struct
{
  uint8_t head;
  uint8_t device_ID;
  uint8_t cmd_ID;
  int16_t set_val;
  int16_t get_val;
  uint8_t reserved[4];
  uint8_t tail[3];
} command;
```

head – command header – 1 byte (0x72)

device_ID – responder identifier – 1 byte

cmd_ID – command identifier (different from cmd_ID of request, see below) – 1 byte

set_val – standardly not used in responses – 2 bytes

get_val – returned value – 2 bytes

reserved – returned value (standardly not used) – 4 bytes

tail – command end mark – 3 bytes (all 0xFF)

Device identifier (device_ID):

- Default value – 0x60
- If several devices are connected to the same RS-485 bus, unique IDs to be assigned to them with SDC_SET_ID command

Command description (cmd_ID, set_val, get_val, reserved) and errors descriptions:

Command	cmd_id	Description / Parameters (request) / Parameters (response)
Requests		
SDC_SET_ID	**0xF0**	**Assigns new device identifier** Parameters (request): • set_val = new_id Parameters (response): • no
SDC_ON	**0x02**	**Enables current pulses at SDC-50A output** TEC may have to be turned ON before using this command (see also SDC_SET_STARTPARAMS command description) Parameters (request): • no Parameters (response): • get_val = 1 - ok • get_val = 0 - error (no TEC, temperature is out of limits)
SDC_OFF	**0x03**	**Disables SDC-50A output** Parameters (request): • no Parameters (response): • no
SDC_SET_CURRENT	**0x05**	**Sets output current (0-50A)** Parameters (request): • set_val = current*10 (e.g. for 34.5A current 345 value to be sent) Parameters (response): • no
SDC_GET_STATUS	**0x07**	**Gets internal status of the device** Parameters (request): • no Parameters (response): • get_val = tec_temp*10 – temperature from the base NTC (example: if temperature is 20.3C, returns 203) • set_val = aux_temp*10 – temperature from the auxiliary NTC • reserved[0]: bit 0 – diode driver status (0- off, 1- on) bit 1 – TEC status (0- off, 1- on) • reserved[1] = fault (see faults section) • reserved[2] = I_{TEC}/255 (I_{TEC} is TEC input current, i.e. current consumed from power supply) • reserved[3] = I_{TEC}%255
SDC_PULSE_SET	**0x09**	**Sets pulse width** Parameters (request): • set_val = pulse_width (1..500us) Parameters (response):

		• no
SDC_PULSE_GET	**0x24**	**Returns pulse width set point** Parameters (request): • no Parameters (response): • get_val = pulse_width
SDC_GET_CURRENT	**0x25**	**Returns current set point** Parameters (request): • no Parameters (response): • get_val = current*10
SDC_TEC_ON	**0x30**	**Turns TEC on** Parameters (request): • no Parameters (response): • get_val = 1 - ok • get_val = 0 - error (temperature is out of limits)
SDC_TEC_OFF	**0x31**	**Turns TEC off** Current pulses are also stopped when the command is sent Parameters (request): • no Parameters (response): • no
SDC_TEC_GET_TEMP	**0x32**	**Returns real temperature and temperature set point** Parameters (request): • no Parameters (response): • get_val = tec_temp*10 – real temperature • set_val = tec_set_temp*10 – temperature set point
SDC_TEC_SET_TEMP	**0x33**	**Sets temperature (target)** Parameters (request): • set_val = tec_temp*10 Parameters (response): • no
SDC_TEC_GET_LIMITS	**0x34**	**Returns temperature limits (absolute minimal and absolute maximal)** Parameters (request): • no Parameters (response): • get_val = tec_low_lim*10 – low limit • set_val = tec_high_lim*10 – high limit
SDC_SAVE_PARAMS	**0x35**	**Save parameters to the device's memory** Parameters (request): • no Parameters (response):

		• no
SDC_SETMODE	**0x36**	**Sets synchronization mode** Parameters (request): • set_val = 0x00 – internal synchronization • set_val = 0x01 – external synchronization, pulse width is defined by internal presets • set_val = 0x02 – external synchronization, repeats SYNC IN pulse Parameters (response): • no
SDC_GETMODE	**0x37**	**Returns synchronization mode** Parameters (request): • no Parameters (response): • get_val = mode (0x00, 0x01, 0x02 – see also SDC_SETMODE command description)
SDC_SET_STARTPARAMS	**0x38**	**Sets start-up parameters (stand-alone mode parameters and TEC stabilization parameters)** Parameters (request): • set_val = self_mode_param self_mode_param = 1 – in stand-alone mode parameters are set from device's memory self_mode_param = 0 – in stand-alone mode parameters are set from onboard trimpots • get_val = tec_stab tec_stab = 0 – current pulses are possible without TEC stabilization tec_stab = 1 – current pulses are impossible until TEC is stabilized Parameters (response): • no
SDC_GET_STARTPARAMS	**0x39**	**Returns start-up parameters** Parameters (request): • no Parameters (response): • get_val = self_mode_param • set_val = tec_stab
SDC_SET_FREQ	**0x40**	**Sets pulse repetition rate (in internal synchronization mode only)** Parameters (request): • set_val = freq*10 Parameters (response): • no
SDC_GET_FREQ	**0x41**	**Returns pulse repetition rate** Parameters (request): • no Parameters (response): • get_val = freq*10
SDC_GET_VERSION	**0xF3**	**Returns firmware version**

		Parameters (request): • no Parameters (response): • get_val = version*10
SDC_CALIB_CLEAR	**0x20**	**Erases current calibration table** Parameters (request): • no Parameters (response): • no
SDC_CALIB_NUM	**0x21**	**Returns number of current calibration points** Parameters (request): • no Parameters (response): • get_val = calib_point
SDC_CALIB_ADD	**0x22**	**Adds current calibration point (20pcs max, points can be between minimal and maximal values only)** Parameters (request): • set_val = set_current (current set point) • get_val = real_current (current measured at the output) Parameters (response): • no
SDC_CALIB_GET	**0x23**	**Gets selected current calibration point** Parameters (request): • set_val = point_num (numeration starts from 0) Parameters (response): • set_val = set_current • get_val = real_current
Responses		
CMD_OK	**0xDE**	**Correct command is detected**
CMD_UNKNOWN	**0xEE**	**Unknown command**
FAULT states		
FAULT_NO	**0x00**	**No faults**
FAULT	**Bit 1**	**Set to 1 in the case of general fault**
FAULT_TEC	**Bit 4**	**Set to 1 if temperature is either >10C below minimal allowed temperature or >10C above maximal allowed temperature**

Other notes:

- If temperature is out of 5..50C limits, the driver will be forcibly stopped. To remove the fault state, send SDC_TEC_ON command after the temperature has returned to the allowed range.
- If no NTC is connected, the returned temperature is -55C.
- When current and temperature trimpots are active, their states are refreshed regularly with approx. 1Hz frequency.
- Commands can be sent independently on SDC-50A state (standby or active).
- Recommended frequency of requests is 3-4Hz or below.
- If the driver doesn't respond the request, it's recommended to repeat it a few times with 1-2ms delays.