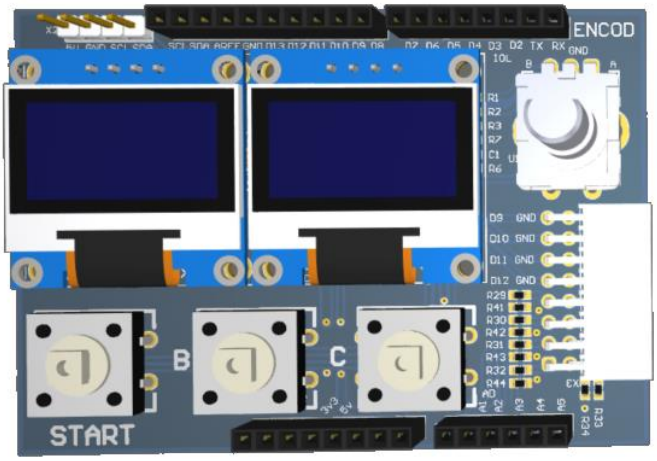
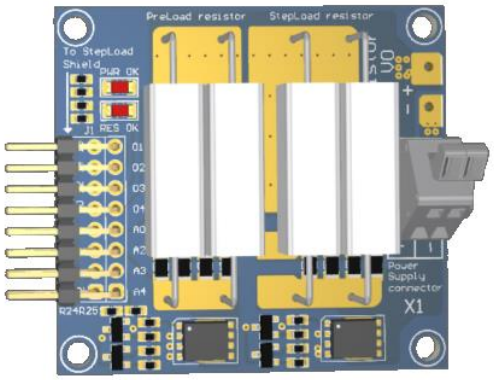
Step load V2 -


ENCOD
START
B
C


PreLoad resistor
StepLoad resistor
To StepLoad Shield
PWR OK
RES OK
Power Supply connector
X1

Content

General description

One of essential power supply stability testing is Step Load response. Idea is quickly change the load value and capture relaxation process. During the testing process it is useful to be able to quickly adjust On/Off time to get an accurate scope image of transient process. This tool was designed with the aim of simplify and increase speed of testing. Project was completely DIY based with low cost components and useful functionality. Electronic schematic itself very simple.
StepLoad Generator consist of 3 parts:

- Arduino UNO (base layer and software processing)
- UNO compatible shield (have 2x oled displays, encoder, 3x buttons, mosfet driver)
- Resistor box board (have two MOSFETs, solder pleases for different resistors)

Software was written in C# using Arduino IDE 1.8.x. Board has manual control as well as full features command line interface over serial port.

Main Board description.

This is UNO compatible shield which have:

- 2x OLED displays
- rotary encoder
- 3x tactile buttons
- MOSFET driver

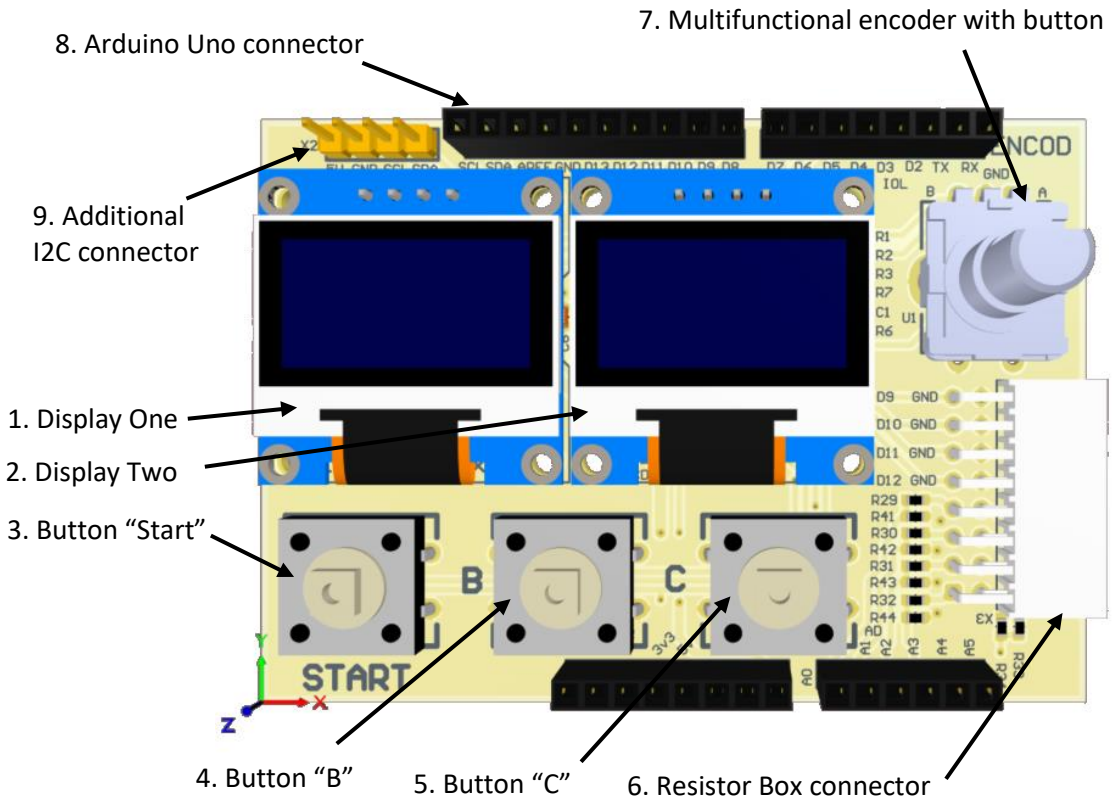
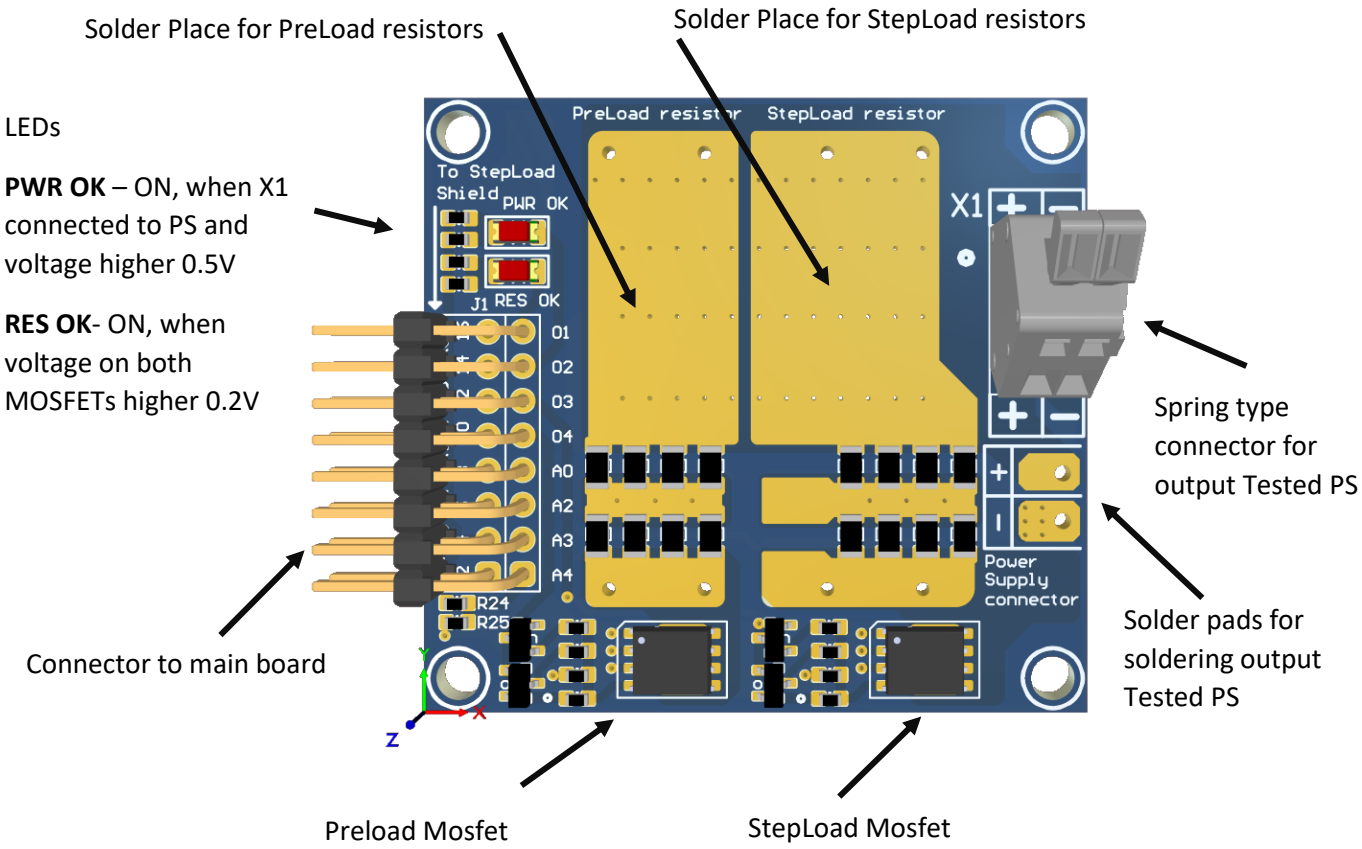


Figure 1 - Front view of StepLoad shield and location of main control components

#	Name	Description and functions
1.	**Display One**	Oled display 128x64 shows current parameters of generator.  **T_on** – Step load resistor to be connected to power supply during this time. Time in microseconds **T_of** – Step load resistor to be disconnected during this time. Time in microseconds. **N_rep** – number of pulsed to be generated on a single press button “Start” N Runs – Run counter since last power up. Each press “start” will increment counter value.
2.	**Display Two**	Oled display 128x64 shows measurement.  **U_ps** – Voltage that connected to resistor box board. Voltage indication is used to ensure that power supply is connected to resistor box with proper polarity and running normally. This voltage value can’t be use for measurement. **Step-Load res** – feedback from MOSFET indicates resistor present. **Pre-Load res** – feedback from MOSFET indicates resistor present. Indication: “**NOT detected**” – shows that there is no input voltage or resistor is not connected to mosfet. “**Connected**” – shows that resistor is connected to MOSFET. Indication has no influence on work cycle.
3.	**Button “Start”**	Runs generator a single time
4.	**Button “B”**	In “Stand-by” decreases On and Off time by 100 us In “Edit T_On” or “Edit T_OFF” modes decrease time value by 1000 us for quick adjustment. In “Edit N_rep” mode decrease value by 10. To enter this “Edit ….” mode hold Endcoder button for 3 sec.
5.	**Button “C”**	In “Stand-by” increase On and Off time by 100 us In “Edit T_On” or “Edit T_OFF” increase On and Off time by 1000 us for quick adjustment. To enter this mode hold Endcoder button for 3 sec. In “Edit N_rep” mode increase value by 10. To enter this “Edit ….” mode hold Endcoder button for 3 sec.
6.	**Resistor Box connector**	Connector for a secondary board that contain a two Mosfets and load resistors.
7.	**Multifunctional encoder with button**	Rotary encoder with button. Rotation to the right increments value, rotation to the left decrement value. Button supports single click, double click and hold functions. In “Stand-by” rotation will increase **N_rep** by 1. To enter “Edit ….” mode hold Endcoder button for 3 sec. Inside “Edit mode” click single time to go navigate for editing next value. Rorate encoder to change value. To exit “Edit ….” mode hold Endcoder button for 3 sec. In “Waiting start” state double click on encoder button will reset all setting to default values.
8	**Arduino Uno connector**	Standard Arduino UNO connector
9	**Additional I2C connector**	Additional i2C connector for connecting additional peripheral devices

Resistor Box Board description

Resistor Box Board is a removable board which have two MOSFETs, solder places for different combination of resistors, spring type connector for clipping bare wires and two LEDs for indication



State diagram

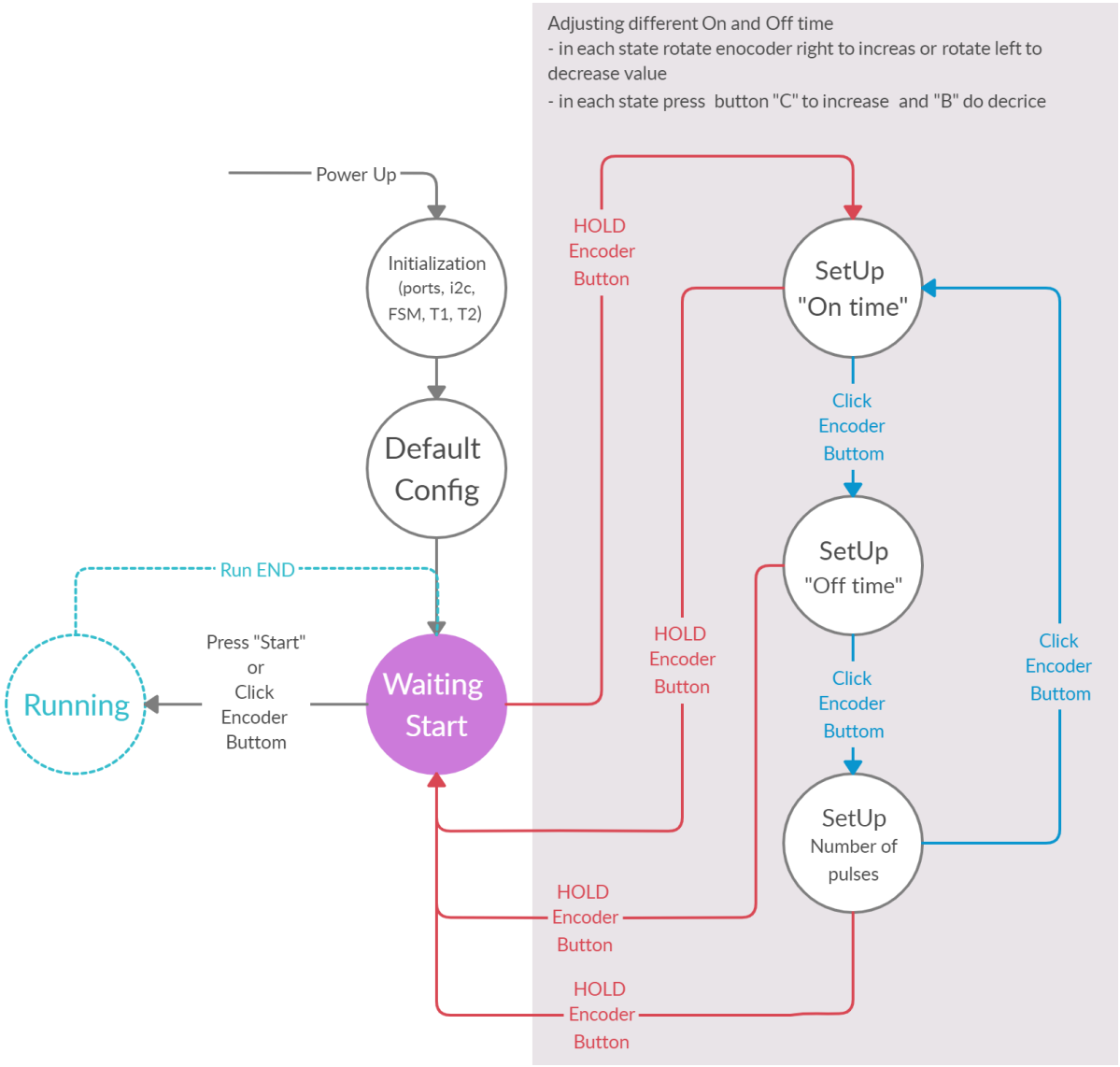


Figure 2 – Software state diagram. Software is based on FSM (Finite-State Machine)

Pulse description

Typical waveform is shown on Figure 3. Generator ran with Nrep=2. PreLoad resistor is connected first. There is a fixed delay of 10 ms delay for relaxation process. After 10 ms step load generator starts with first Off_time, then On_time, and number of repetition. After last pulse, there is another fixed delay of 10 ms before Preload resistor will be disconnected.

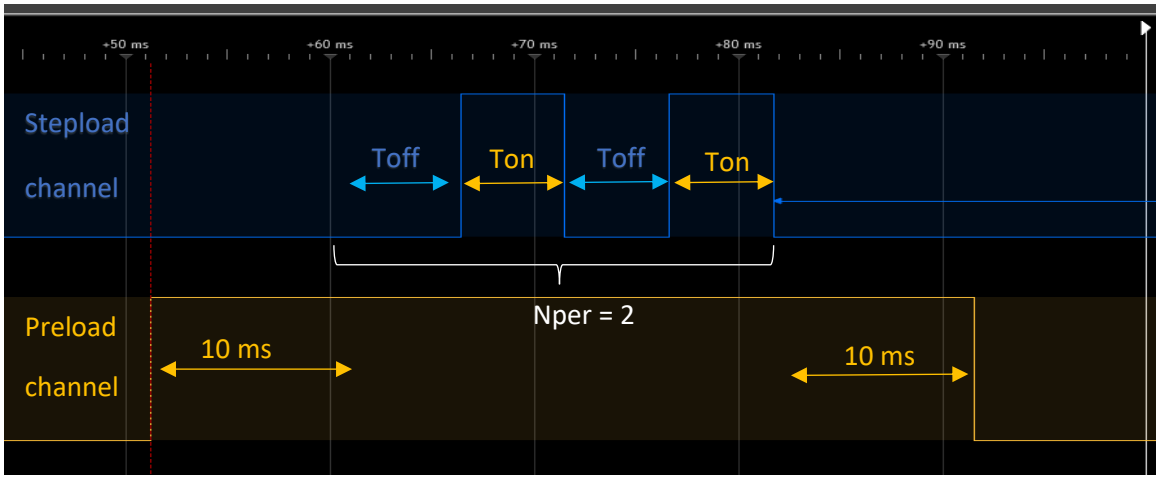


Figure 3 – Typical waveform of a single run with 2 pulses.

Command line interface

Driver Installation

Arduino UNO have built in USB-to-UART converter that is used to control StepLoad board.

Drivers available for Windows XP or later; Linux X86; Linux ARM; MacOS X 10.8 or later.

If Arduino Installer was used, Windows - from XP up to 10, will install drivers automatically as soon as you connect your board.

To install serial driver only, without Arduino IDE, follow the steps below:

- Download Arduino IDE Zip package [link: https://www.arduino.cc/en/Main/Software].
- Unzip package.
- Click on the Start Menu and open up the Control Panel.
- While in the Control Panel, navigate to System and Security. Next, click on System. Once the System window is up, open the Device Manager.
- Look under Ports (COM & LPT). You should see an open port named "Arduino UNO (COMxx)". If there is no COM & LPT section, look under "Other Devices" for "Unknown Device".
- Right click on the "Arduino UNO (COmxx)" port and choose the "Update Driver Software" option.
- Next, choose the "Browse my computer for Driver software" option.
- Finally, navigate to and select the driver file named "arduino.inf", located in the "Drivers" folder of the Arduino Software download (not the "FTDI USB Drivers" sub-directory). If you are using an old version of the IDE (1.0.3 or older), choose the Uno driver file named "Arduino UNO.inf"

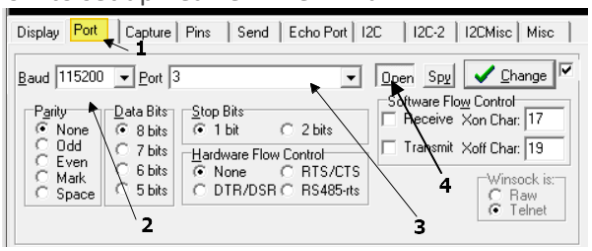
- Windows will finish up the driver installation from there.

UART Set-Up

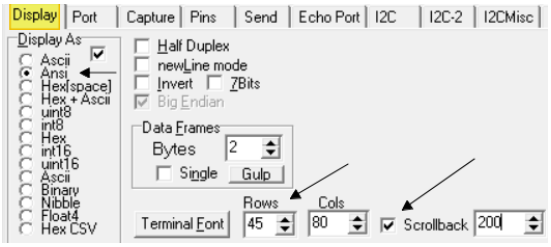
Default UART speed is: 115200.
Actual UART speed is always displaying in boot massage on left display in first 3 seconds of booting.

RealTerm SetUp example

Command line interface accessible over any kind of Serial terminals. It is working with ASCII symbols.
How to set up RealTerm Terminal:

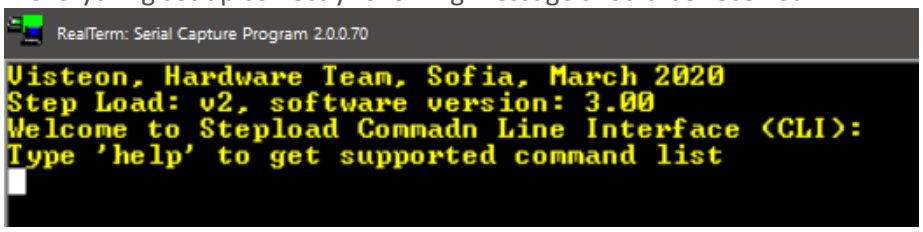


Set Up port number, speed and Open Port



[Optional]: Set Up for better view.

If everything set up correctly following message should be received:



Command list

To control board using Serial interface, board should be in “Waiting start” state. This is default state after boot. If board in “Edit …” setting mode, please hold encoder button to exit “Edit …” state and return to “Waiting start”. UART available only in this state. In any other state commands would be unavailable. All commands should be written on LOW case.

Command 1:	Description:	
start	Start generating pulse. Works the same way as hardware “Run”. Start - Command could send without parameter to execute “Run” immediately Start <delay ms> - to execute command with delay. Delay value = 0 ms – 5000 ms After execution current value of run counter will be returned type: **start**	
	Return Value:	**Run counter:1**
	Examples 1:	**start 100**
	Examples 2:	**start**

Command 2:	Description:	
set	Setup pulse generator parameters, like: ON time, Off time, Nrep. Parameters separated by space. *Parameters list:* **-ontime** – set On time value in us. Value should be in range 100 us - 6500 us **-offtime** – set Off time value in us. Value should be in range 100 us - 6500 us **-nrep** – set number of repetitions value. Value should be in range 1-500 type: **set -ontime <val us> -offtime <val us> -nrep <val>**	
	Return Value:	Command has no return value
	Examples:	**set -ontime 250 -offtime 300 -nrep 5**

Command 3:	Description:	
setontime	Set the ON pulse duration time. Value should be in range 100 us - 6500 us type: **setontime <val us>**	
	Return Value:	Command has no return value
	Examples:	**setontime 500**

Command 4:	Description:	
setofftime	Set the OFF pulse duration time. Value should be in range 100 us - 6500 us type: **setofftime <val us>**	
	Return Value:	Command has no return value
	Examples:	**setofftime 500**

Command 5:	Description:	
setofftime	Set number of repetitions value. Value should be in range 1-500 type: **setnrep <val>**	
	Return Value:	Command has no return value
	Examples:	**setnrep 10**

Command 6:	Description:	
getstatus	Returns current generator setting and load voltage type: **getstatus**	
	Return Value:	**ONtime[us]= 500; OFFtime[us]= 500; Ncycles = 500; Nruns = 1; Vload[V]= 0.000**
	Examples:	**getstatus**

Command 7:	Description:	
getvload	Returns measured voltage on load connector type: **getvload**	
	Return Value:	**Vload [V]= 0.000**
	Examples:	**getvload**

Command 8:	Description:	
help	Returns supported command list to terminal window type: **getvload**	
	Return Value:	See Figure 4
	Examples:	**getvload**

```
# help
----------Supported command list:----------
1 | start | Start generating, trigger run button via uart
* | type: start  - 'start immideatly'
* | type: start <delay_value=0-5000ms>; example: start 100
-------------------------------------------
2 | set   | Setup pulse generator parameters ON time, Off time, Nrep
* | type: set -ontime <val us> -offtime <val us> -nrep <val>
* | example: set -ontime 250 -offtime 300 -nrep 5
-------------------------------------------
3 | set ON time | Set the ON pulse duration time, in us
* | type: setontime <val us>; example: setontime 250 [100-6500 us]
-------------------------------------------
4 | set OFF time | Set the OFF pulse duration time, in us
* | type: setofftime <val us>; example: setofftime 500 [100-6500 us]
-------------------------------------------
5 | set Nrep | Set nubmer of pulses to be geenerated on a signle run
* | type: setnrep <val times>; example: setnrep 20  [1-500]
-------------------------------------------
6 | status|Returns current generator setting and load volatge
* | type: getstatus
-------------------------------------------
7 | Vload |Returns measured voltage on load connector
* | type: getvload
-------------------------------------------
8 | help  | Display this massage
* | type: help
-------------------------------------------
```

Figure 4 – Return massage of help command (software version 3.00).

Command line error processing.

Command with wrong spelling will result in error massage:

*# **hlp***
ERROR: Command not found at 'hlp'

Arguments that is out of the range will automatically be fit into the range without notification

*# set -ontime **250000** -offtime **300000** -nrep **5000***
 # getstatus
ONtime[us]= 65000; OFFtime[us]= 65000; Ncycles = 500; Nruns = 0; Vload[V]= 0.000

Command with wrong argument spelling will result in error massage

*# set -**ortime** 250000 -offtime 300000 -nrep 5000*
ERROR: Unknown argument at command 'set' at '-ortime'
Did you mean "set -ontime <value> -offtime <value> -nrep <value> -ortime 250000 -offtime 300000 -nrep 5000"?

Automatization and Scripting support

StepLoad V2 and RealTerm support simple scripting that allow simple automation to be made in measurement or triggering event.

Writing script

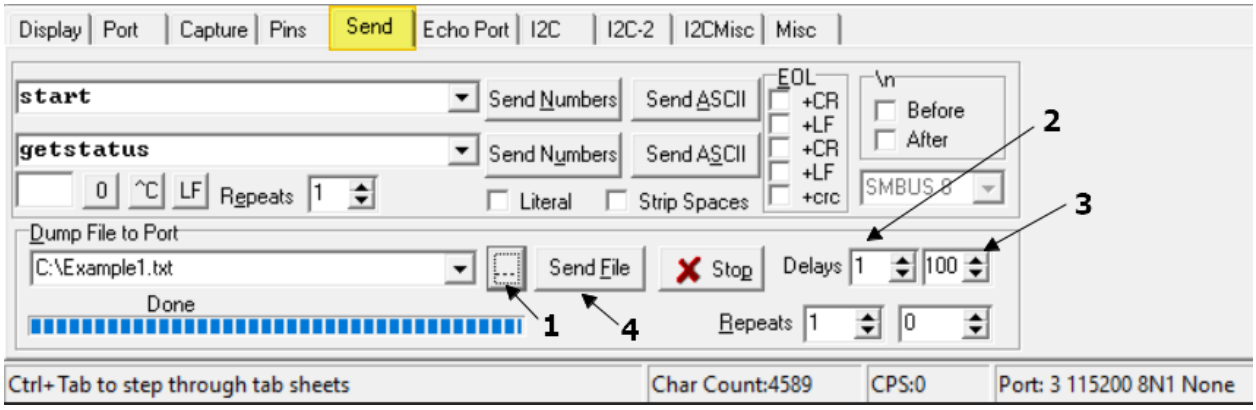
Simple way and yet useful to write a script is to make a *.txt file with the list of command that need to be executed. Bellow the example of script that generate two pulses with different duration time:

```
set -ontime 250 -offtime 250 -nrep 2
start
set -ontime 500 -offtime 500 -nrep 2
start
set -ontime 1000 -offtime 1000 -nrep 2
start
set -ontime 5000 -offtime 5000 -nrep 2
start
```

Copy and save this text into Exmaple1.txt file.

RealTerm SetUp

In Serial terminal **RealTerm** you need set up port and speed first. Take a look **RealTerm SetUp example** part. If communication is established go to “Send” tab and make following adjustments:



1. Navigate to saved script in txt format
2. Setup delay between sending each symbol
3. Setup delay between sending each string.
4. Press “Send File” to execute script.