KiGN Servo Tester

Hardware Revision 0.2

Firmware Version 0.1.x

Safety Instructions

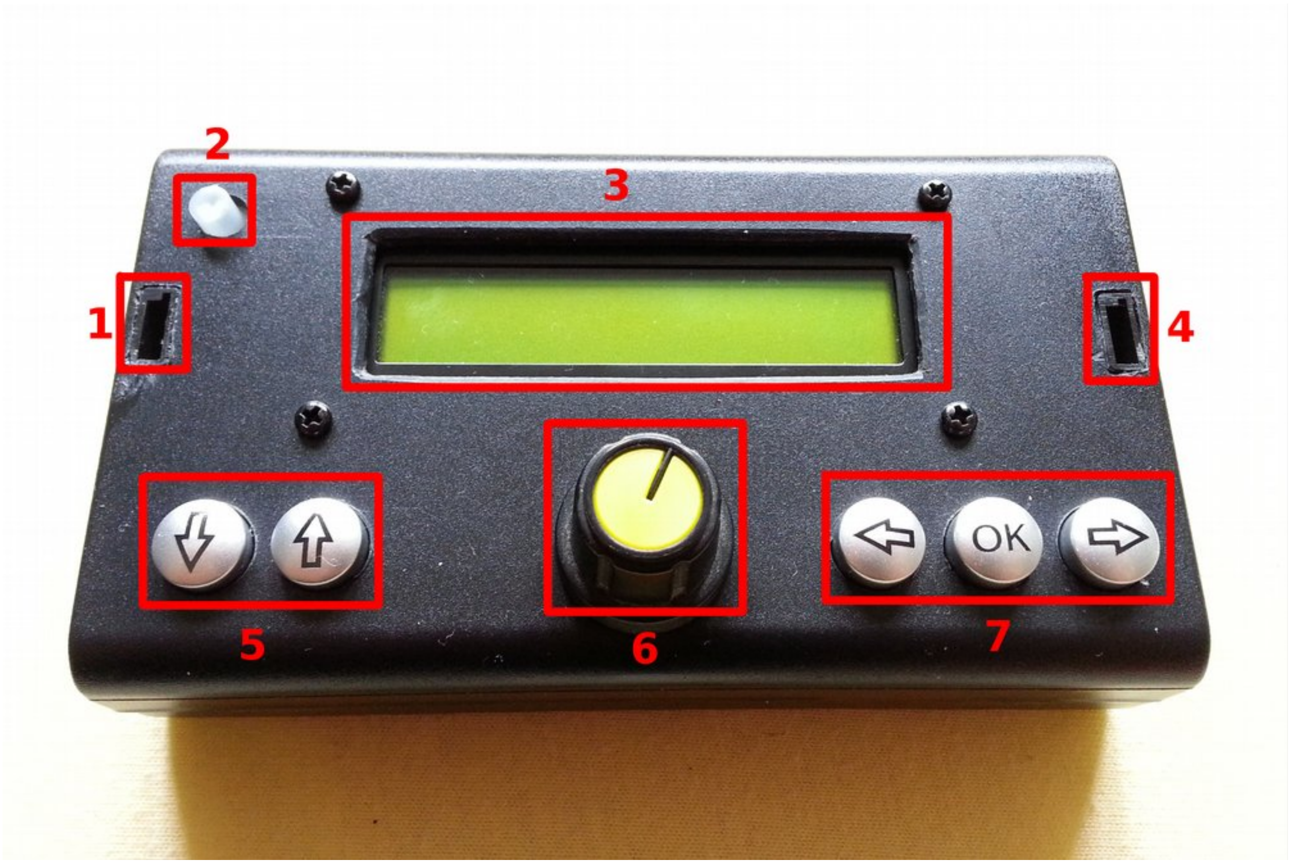
Always ensure **correct voltage rating and polarity** before connecting power source and servo! If your **servo behaves unexpectedly** - moves unintentionally, audibly strains or moves erratically etc, immediately **disconnect** it from tester to prevent damage! The **tester can be set up to generate non-standard PWM signals that can force a servo beyond its mechanical limits which may cause permanent damage**! Please carefully read your servo's datasheet to find correct PWM range, and **set up PWM parameters to match your servo's specications before connecting.** If you suspect that your servo could be **damaged or malfunctioning, do not use it on RC models** until fixed or replaced! **KiGN takes no responsibility from any damage / harm caused by neglecting these safety measures.** Although we tested this appliance as thoroughfully as we can, there are servos, power sources, circumstances or unknown bugs which may cause malfunction.

Technical Specifications

• Operating Voltage*:	4.2 to 5.5 V DC
• Operating Temperature:	0 to 40ºC
• Humidity:	5% to 85% non-condensing
• Current Drawn:	0.4 A @ 4.5 V
• Dimensions:	125 x 72 x 49 mm
• Display:	LCM1602 2x16 character LCD display
• Microcontroller Unit:	Atmel ATMEGA328
• Internal Clock Speed	3 MHz
• Servo Signal mode:	PWM
• PWM Frequency Range:	50 to 300 Hz
• Minimum Pulse Width:	50 µs
• Servo Memory:	20 slots
• PWM Resolution:	0.33 µs
• Potmeter Resolution:	10 bit (0 - 1024)

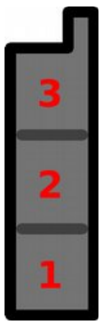
* The tester and most analog servos will work from around 3.8 V – for example a single cell LiPo battery charged to at 2/3 of its capacity – but LCD screen will be barely readable even with highest contrast setting

Front Panel



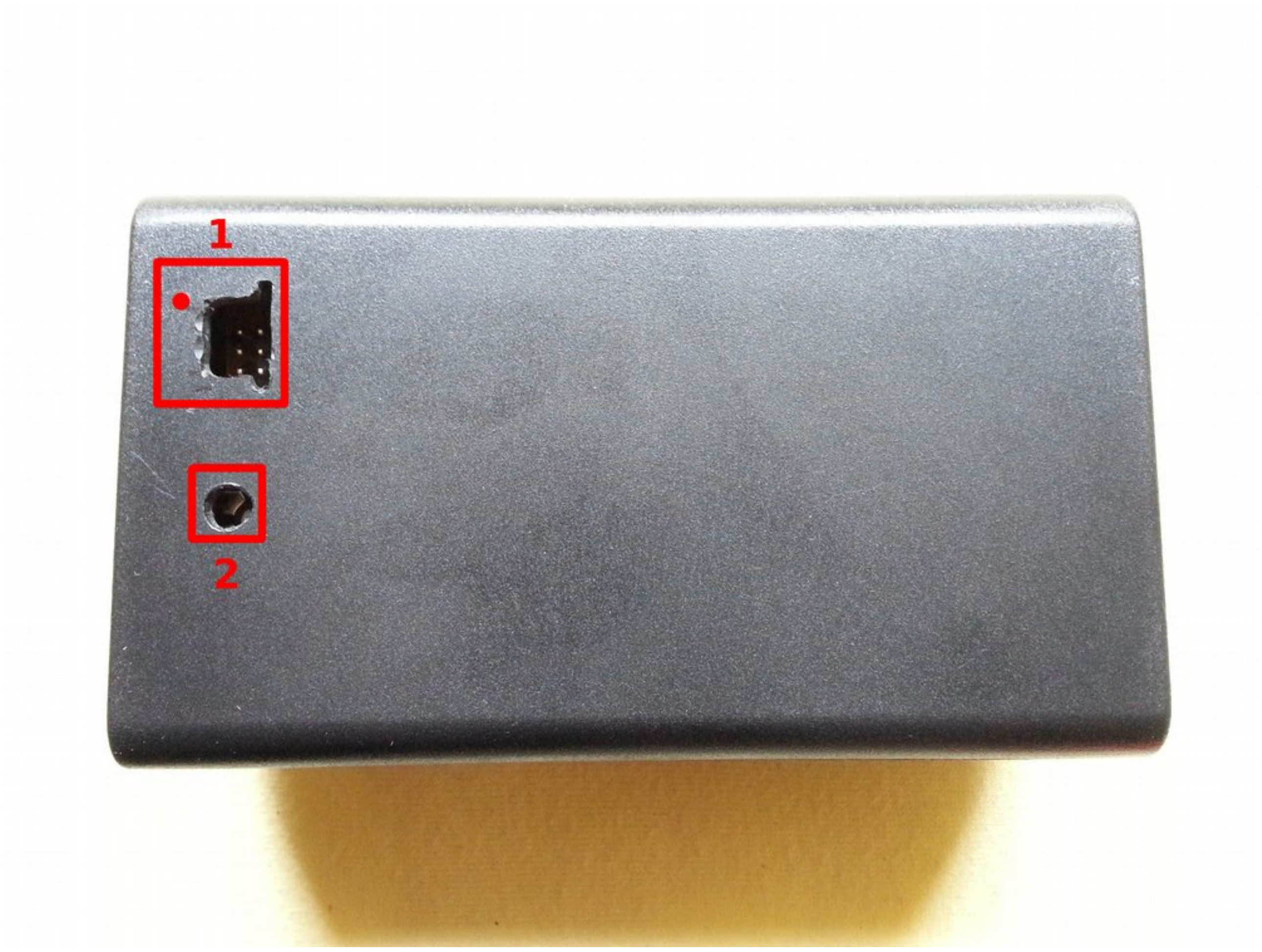
1. **Power Input Socket**: pin 1: GND, 2: +5V, 3: Not Used
2. **Error LED** / BiStable Switch: error LED indicates an impossible / not supported PWM range setting. Switch is reserved for future functions and is currently ignored.
3. **LCD Display**
4. **Servo Socket**, pin 1: GND, 2: +5V, 3: signal
5. **Menu / Screen Select**:
   - ↓: previous, ↑: next
6. **Potmeter**: pulse width / character select
7. **Multi-Function Buttons**:

• ←	left	/ decrease	/ previous item
• OK	center	/ quit	/ apply
• →	right	/ increase	/ next item

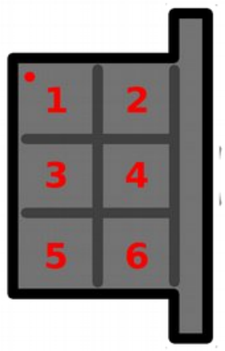


Power / Servo Socket

Back Panel



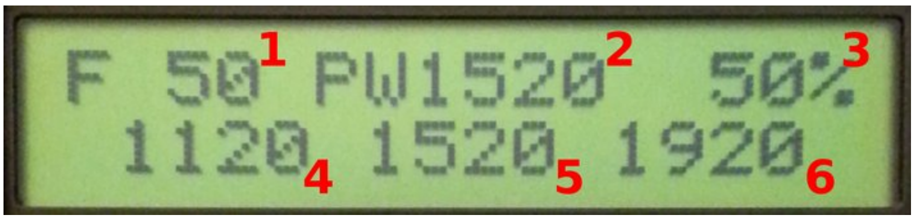
1. **ISP port**: 6-pin Atmel In-System Programming interface for firmware upgrade
2. **LCD display contrast**: use a small + or – head narrow screwdriver to adjust. Display contrast also depends on power source voltage.



ISP port

Menus and functions

1. Live PWM data (default view)



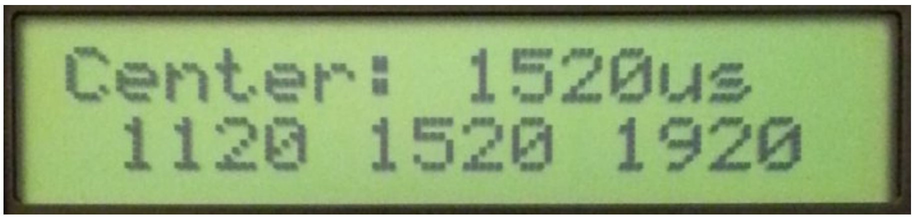
```
F 50 PW1520  50%
 1120 1520 1920
```

1. **PWM Frequency** (Hz)
2. **Current Pulse Width** (µs)
3. **Duty Cycle** (%)
4. **Minimum Pulse Width** (0%, µs)
5. **Center Pulse Width** (50%, µs)
6. **Maximum Pulse Width** (100%, µs)

Button functions:

- ← generate minimum PW
- OK generate center PW
- → generate maximum PW

2. Center Pulse Width Adjustment



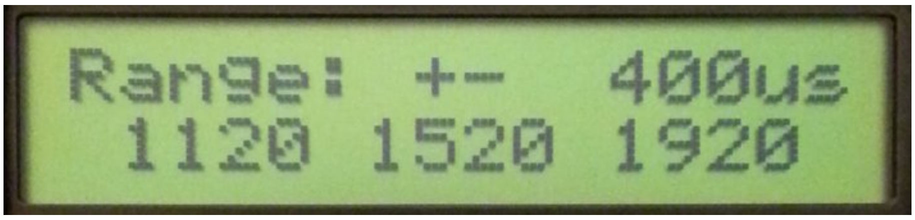
```
Center: 1520us
 1120 1520 1920
```

Button functions:

- ← decrease center PW by 10 µs
- OK return to live data
- → increase center PW by 10 µs

Servo settings are not stored automatically to conserve EEPROM memory life.

3. Pulse Width Range

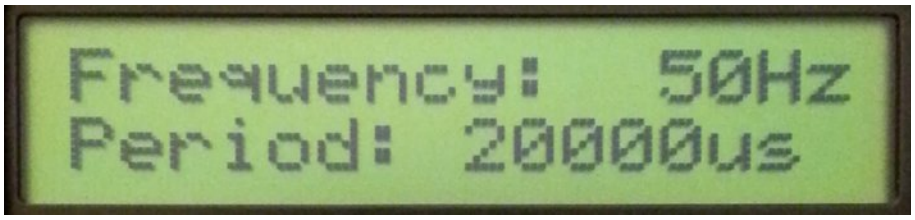

```
Range: +-  400us
 1120 1520 1920
```

For example ±400 µs means a total range of 1920 – 1120 = 800 µs, symmetrical around center pulse width.

Button functions:

- ← decrease range by ±10 µs
- OK return to live data
- → increase range by ±10 µs

4. PWM Frequency

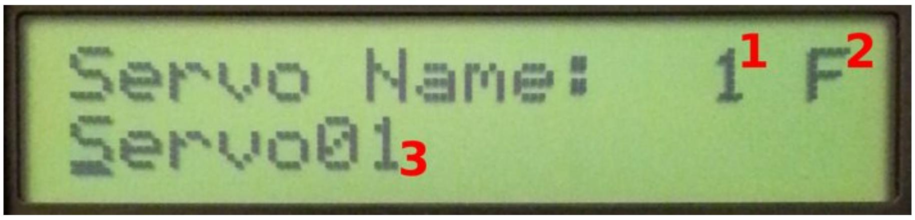

```
Frequency:  50Hz
Period: 20000us
```

PWM Frequency setting in Hz. PWM Period is 1 / PWM Frequency.

Button functions:

- ← decrease frequency by 10 Hz
- OK return to live data
- → increase frequency by 10 Hz

5. Servo Name


```
Servo Name:   1 F
Servo01
```

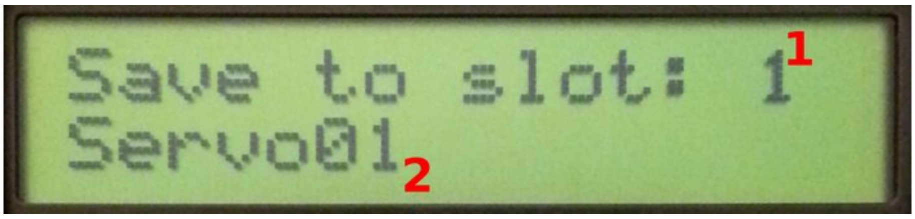
1. **Memory Slot** (1-20): slot number where the current servo settings are loaded from
2. **Selected Character**: use potmeter to select a-z A-Z 0-9 and space characters
3. **Servo Name**: maximum 16 characters

Please note that potmeter is disconnected from PWM signal generator while in Servo Name menu, and is reconnected when exiting. Make sure that potmeter is in a position that is within the mechanical limits of the servo when exiting.

Button functions:

- ← move cursor one step left
- OK Insert selected character at current position and advance cursor
- → move cursor one step right

6. Save Servo Settings


```
Save to slot: 1
Servo01
```

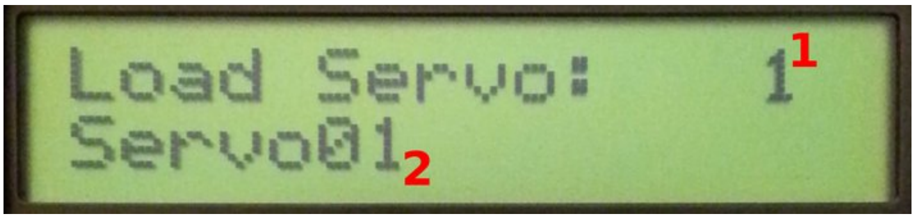
1. **Memory Slot**: slot number to save servo settings to
2. **Name of Servo To Be Overwritten**

Please note that servo settings are not stored automatically to conserve EEPROM life.

Button functions:

- ← select previous slot
- OK Save servo data to currently selected slot and return to live data
- → select next slot

7. Load Servo Settings


```
Load Servo:    1
Servo01
```

Button functions:

- ← select previous slot
- OK Load servo data from currently selected slot and return to live data
- → select next slot

Firmware Upgrade

Programmer Hardware

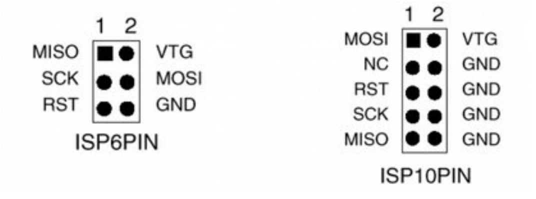
Firmware upgrade can be carried out with any **Atmel 6-pin ISP or compatible programmer**. The best is of course Atmel original ISP programmer, but it comes with a price tag. Most programmers will work on Windows, Linux, MacOS. A slight modification to the instrument case may be needed to accomodate different cable heads and plugs. Most cheap USB programmers are only available with 10-pin cable, so make sure the programmer comes with 6-pin interface or converter before checking out at your favourite webshop.

Example 3rd party programmers*:

URL	Price	Remark>
https://www.pololu.com/product/1300	$19.95	seems like a decent programmer with extra features
http://www.ebay.com/itm/10-Pin-Convert-to-Standard-6-Pin-Adapter-Board-USBASP-USBISP-AVR-Programmer-USB-/310730828729	$5.18	El Cheapo 10-pin programmer that comes with a 6-pin converter

Example DIY 10-pin to 6-pin converter

You may build your own converter plug yourself as 10-pin headers only utilize 6 pins – actually the same pins as 6-pin header but in different order (NC: Not Connected)



* These programmers have not been tested, just serve as an example where to look for. Please do not buy the cheap-est-est out there.

Software

Open Source: avrdude

Will work with most programmers.

Linux:	http://www.nongnu.org/avrdude/
Windows:	http://sourceforge.net/projects/winavr/

Original Atmel Software: Atmel Studio

http://www.atmel.com/tools/atmelstudio.aspx

Before Programming

Firmware is stored in so-called flash memory, while servo settings are stored in EEPROM in the microcontroller unit. **Writing the flash memory requires an erase cycle that will destroy the contents of both flash and EEPROM, so please make sure that you backup the EEPROM in advance** (may not be necessary with Atmel Studio, please check the manual), and restored afterwards.

Atmel Studio

To program firmware image with Atmel Studio, please refer to the documentation available at Atmel's.

AVRDude

These are the exact command lines your very servo tester has been programmed using an Atmel original JTAG3ISP programmer*:

1. Make a backup of EEPROM memory to eeprom_save_file.bin

   ```
   avrdude -c jtag3isp -p m328 -U eeprom:r:eeprom_save_file.hex:i
   ```

2. Flash new firmware

   ```
   avrdude -c jtag3isp -p m328 -U flash:w:firmware_filename.hex
   ```

3. Restore EEPROM from eeprom_save_file.bin

   ```
   avrdude -c jtag3isp -p m328 -U eeprom:w:eeprom_save_file.hex:i
   ```

You are strongly encouraged to make a backup of your EEPROM regularly. Should your EEPROM become corrupt or need to replace the microcontroller unit, you can always restore it from backup.

* avrdude parameter `-c` specifies programmer hardware, please replace it with what is appropriate for your programmer.

Customer Replaceable Parts

Actually, you may replace any parts of the tester you feel the need to, or customize it to your needs. You may even replace the microcontroller itself to another ATMega328 or 328p microcontroller itself as it is seated in a socket – should that become necessary, for instance, after applying reverse polarity voltage. You only need a good soldering iron, desoldering pump, spare parts, some basic soldering skills and a little patience. The schematics will be available on the homepage later this year.

Contact

Please feel free to contact us should you have any feedback to give – feature requests, problems, know-to-work power sources and servos etc.

Product page with latest documentation and firmware:

http://www.kign.org/servo_tester

E-mail: lengyakb@kign.org