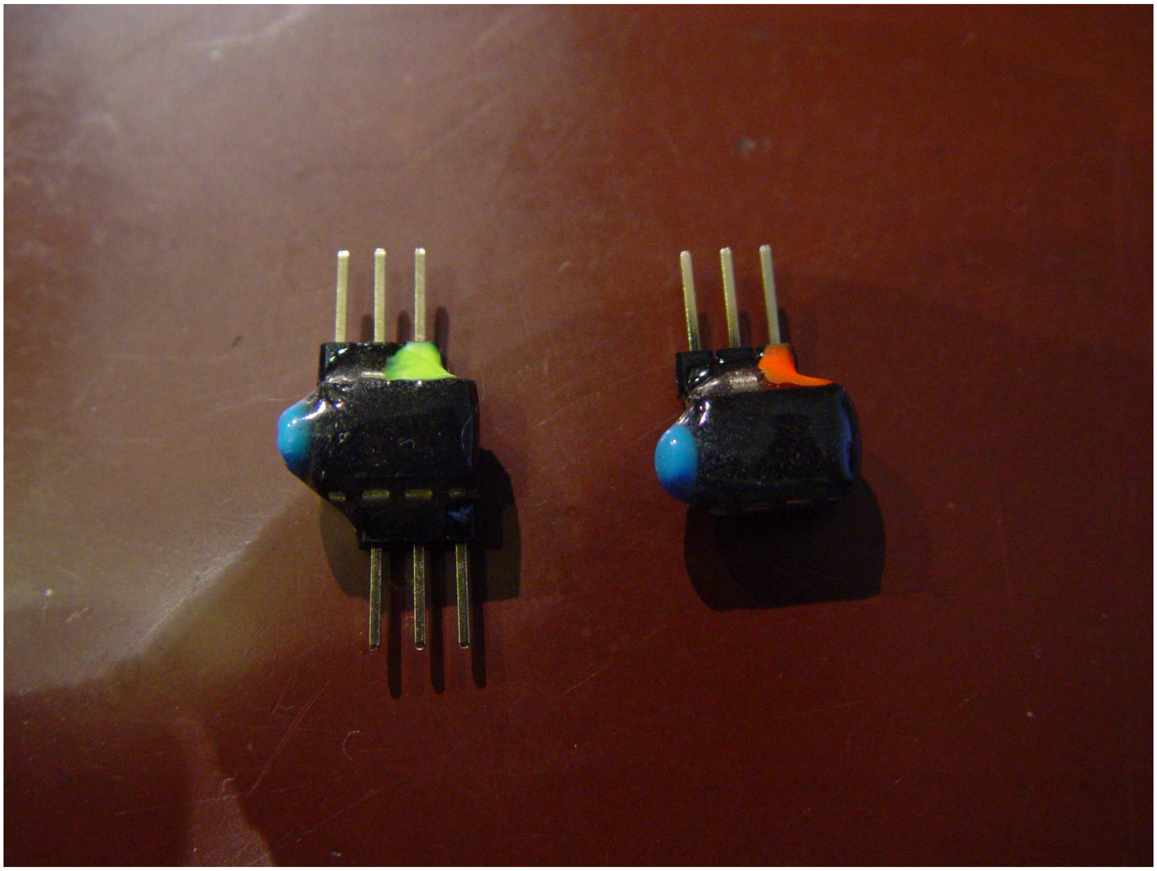
iTimer v 2.0

The timer (sequencer) autonomously creates commands in standards of R/C equipment. It makes one servo control signal (for R/C servo motor or regulator), a logical (bit) output for switching on / off other electrical devices and it has a logical (bit) input from sensors, switches or other timer.

Parameters:

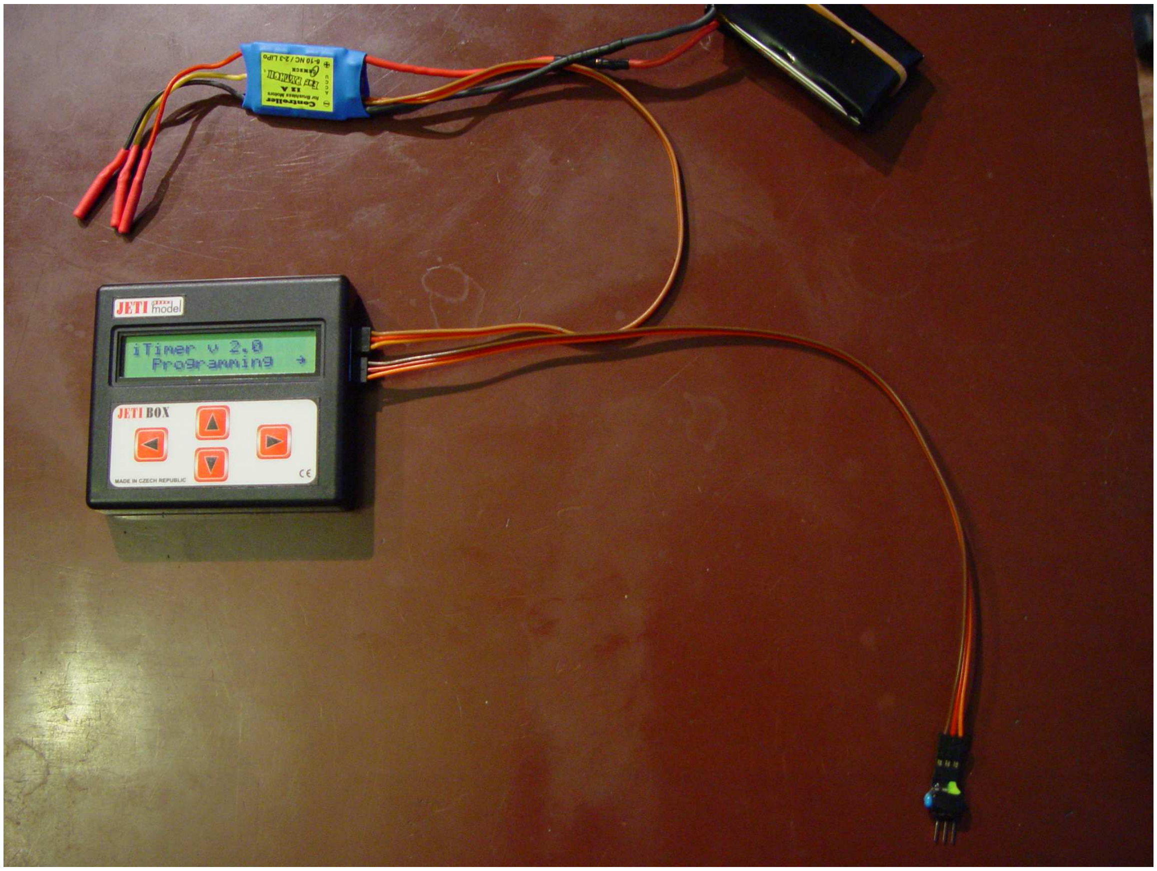
- power supply 3V to 5.5V for runnig (4.8V to 5.5V for programming)
- 64 programming steps
- programmable time intervals are up to 1 hour, the minimum time step 0.1s
- standard R/C servo programmable output range 1ms to 2ms in 200 steps (100 is neutral)
- maximum current of the logic outputs 20 mA
- the instruction set includes commands for waiting; setting of the logic output; testing logic input; conditioned jumps; interrupts; logic inputs and an asynchronous timeout
- all programmable times are controlled on base of the calibrated RC oscillator

Timer has two versions:



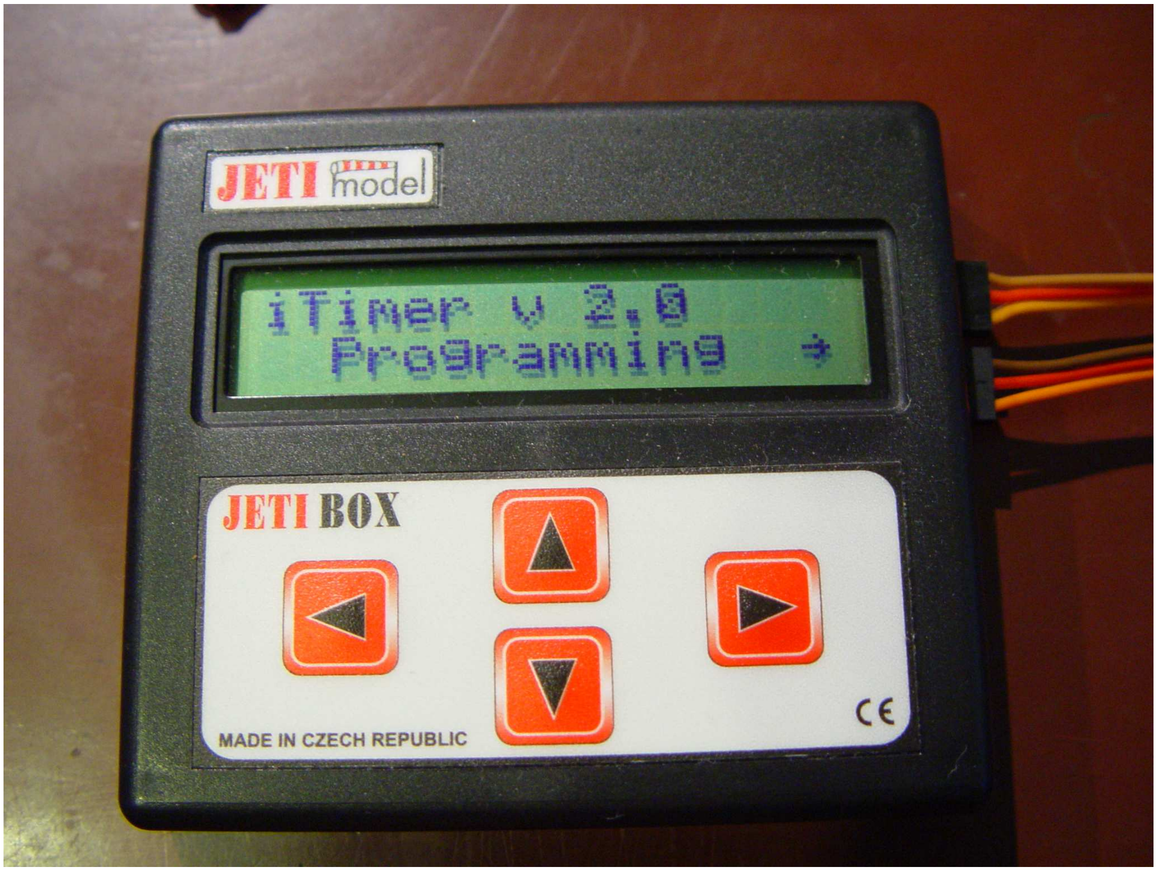
They are functionally identical but smaller version has no logical interface. It means that the smaller version creates servo signal only. Servo output signal pin is marked by color.

Programming is possible by Jetibox (product of company Jeti). To do that, connect the upper connector of the Jetibox to the 4.8 to 5.5V power supply (the picture shows power supply from regulator) and connect the timer to the lower connector by an extension cable.



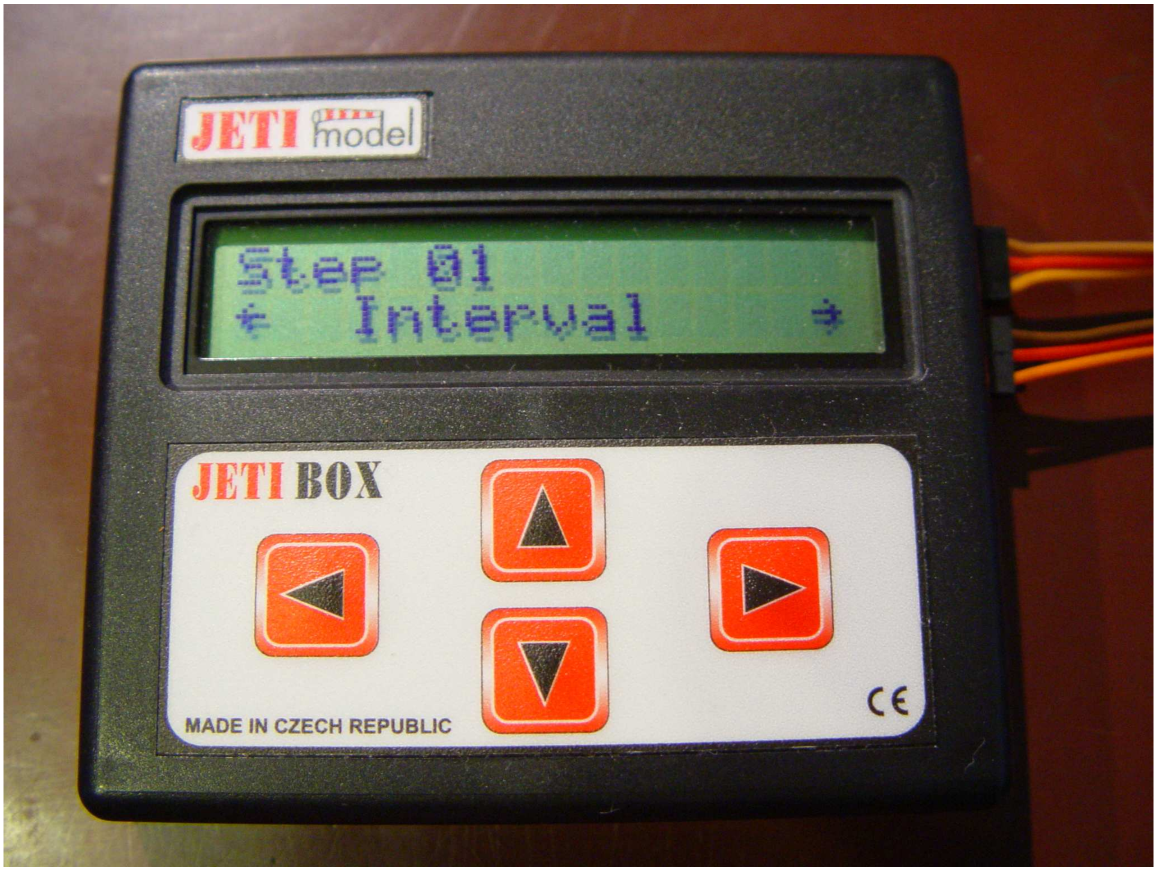
Programming steps are selected by arrows up and down, their parameters are selected by arrows left and right.

Examples:

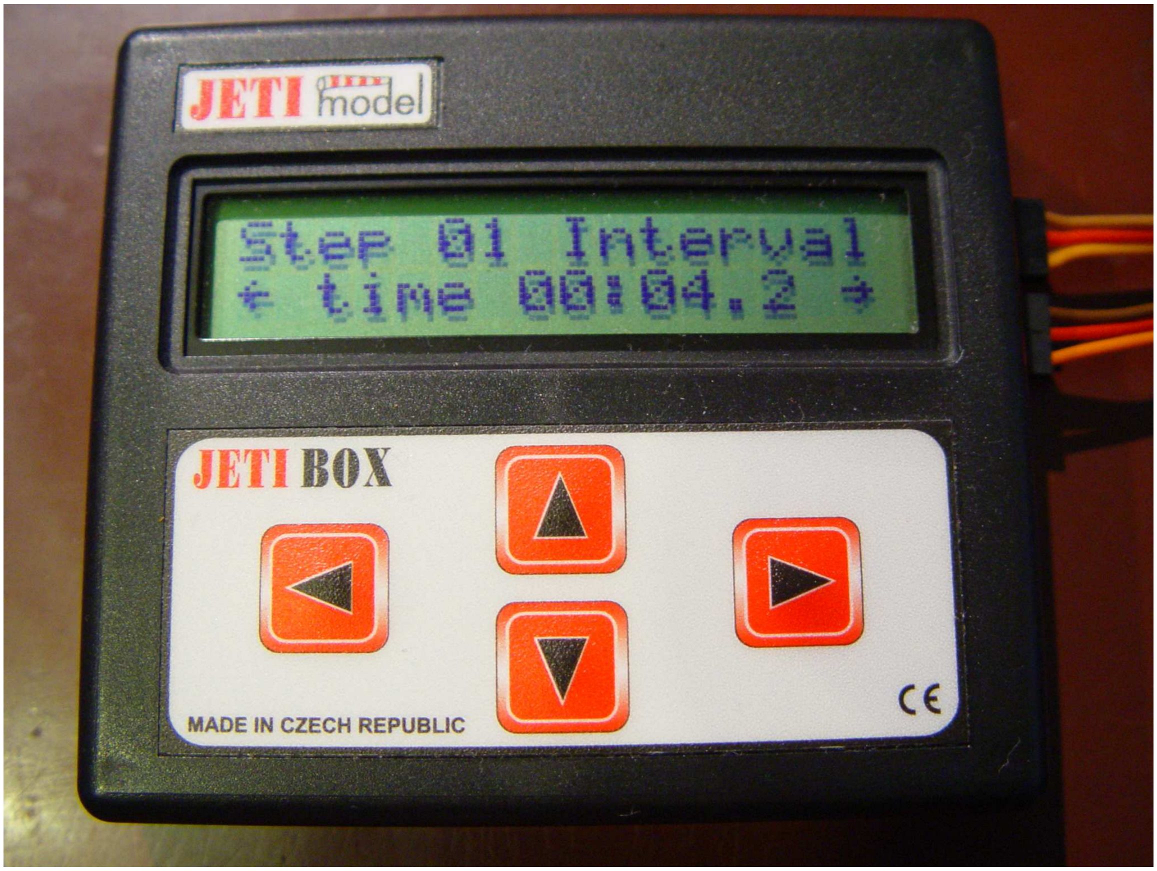


```
iTimer v 2.0
  Programming  →
```

This is first (splash) screen, right arrow goes to clearing of the entire program, the down arrow to the first step of the program.

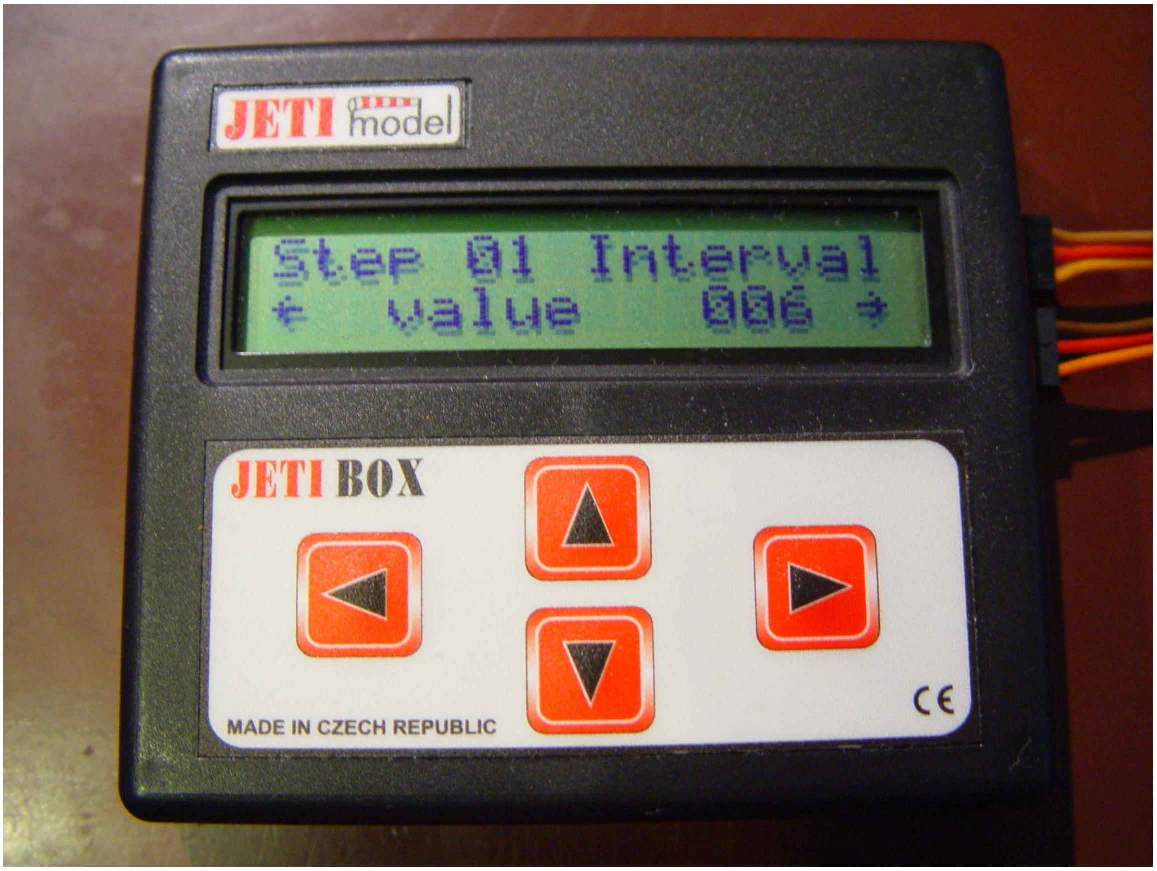


The first command is a command of the time interval.



```
Step 01 Interval
← time 00:04.2 →
```

Interval takes 4.2s.

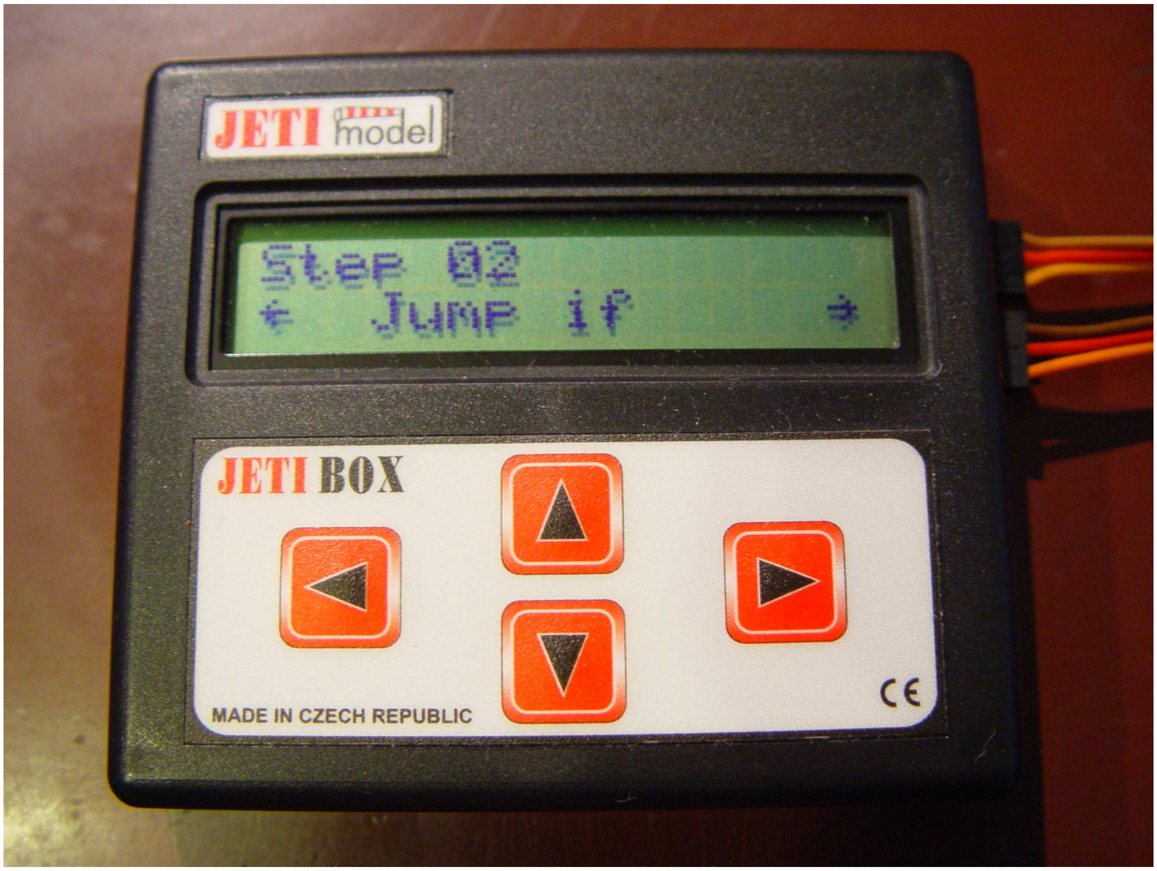


```
Step 01 Interval
← value    006 →
```

The attached servo will go to the position 6 from range 0 to 200, it means that the servo impulse will take 1ms + (6 / 200) * 1 ms = 1.03 ms. The value of 0 will be on one side, the value of 200 will be opposite side and the value of 100 will be the center of the range.

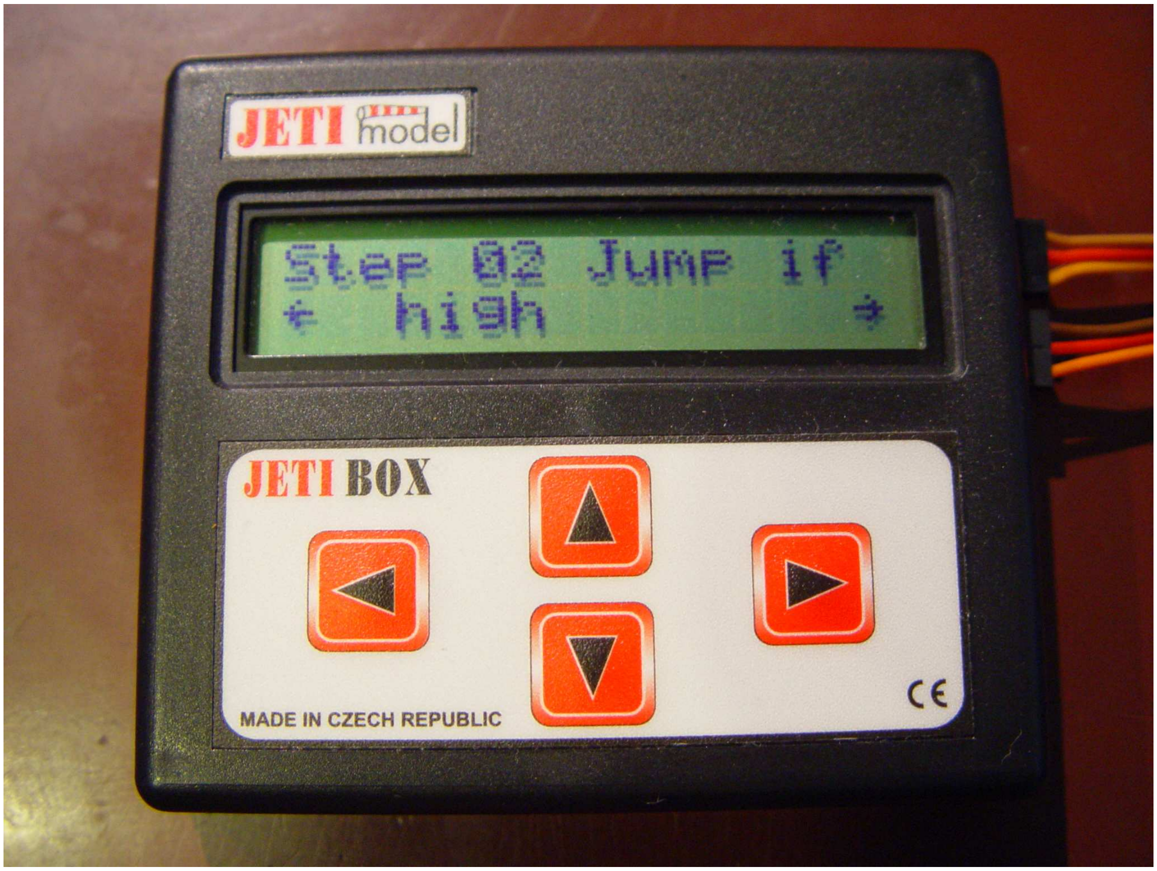
Usual electric regulator will interpret this value as 6 / 200 = 3% throttle, the value of 0 as iddle, the value of 200 will be full gas.

The servo position is set at beginning of the time interval.



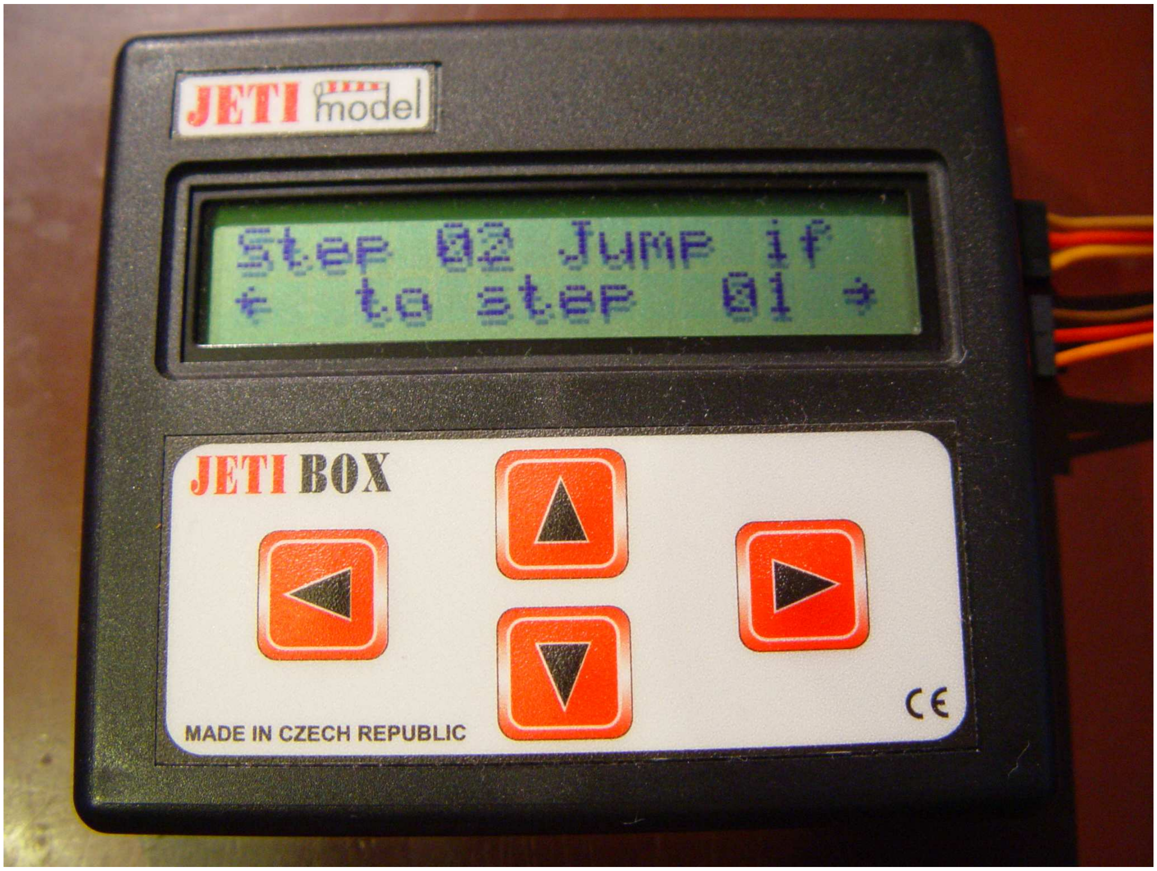
```
Step 02
←  Jump if       →
```

Conditional jump command.


```
Step 02 Jump if
←  high         →
```

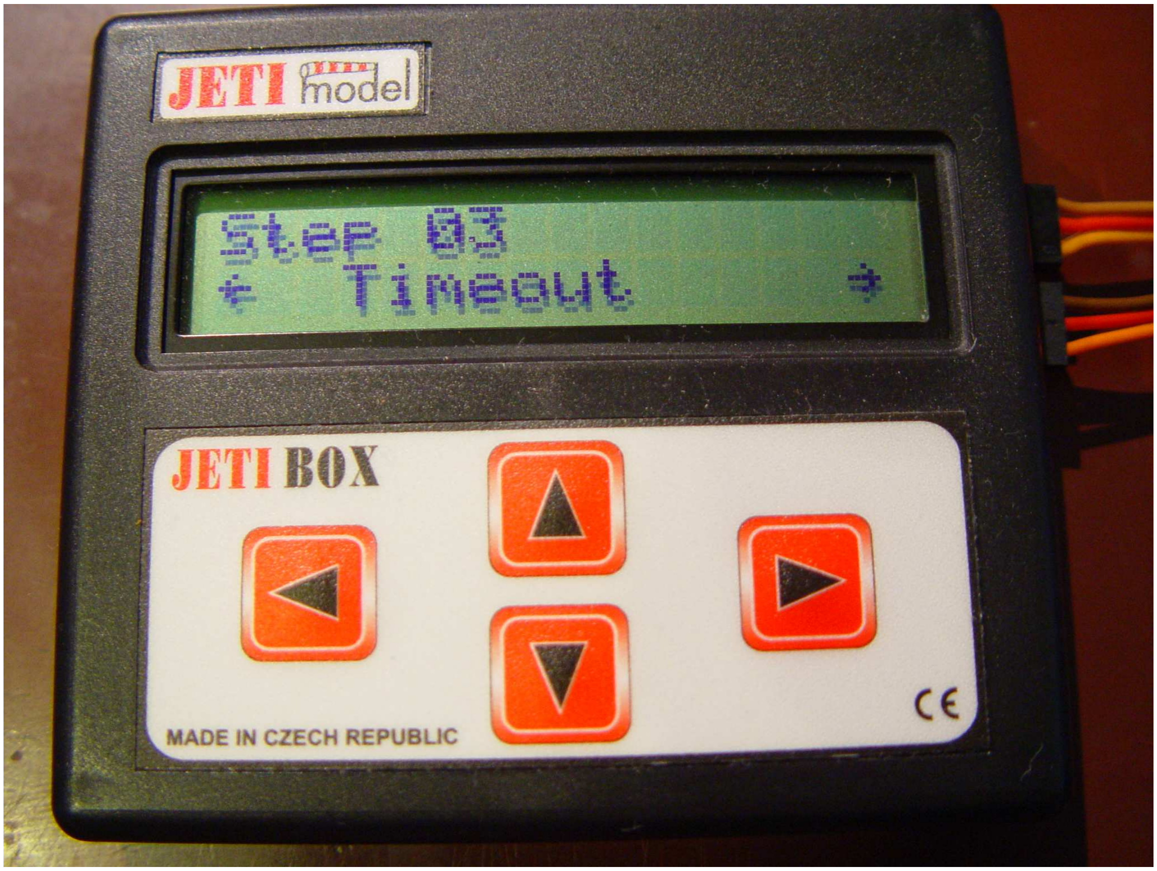
Jump is executed if "high" is on input pin of the timer. Logical input is treated by pullup resistor, so logical "high" is automatically present on input pin.

Option "int" instead of "high" allows interrupt during following commands and does not cause immediate jump – program will continue on next command.



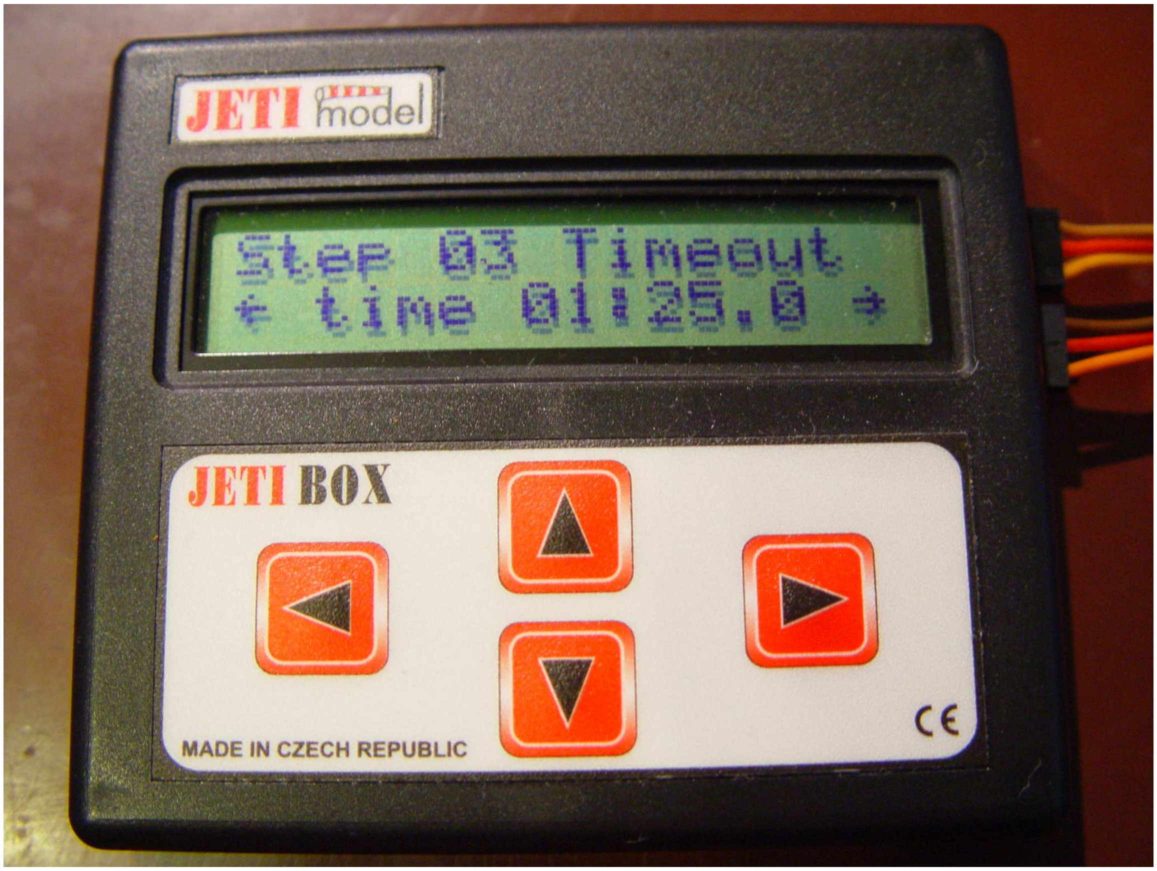
```
Step 02 Jump if
←  to step  01 →
```

Destination address is number of destination command (1 of 64)



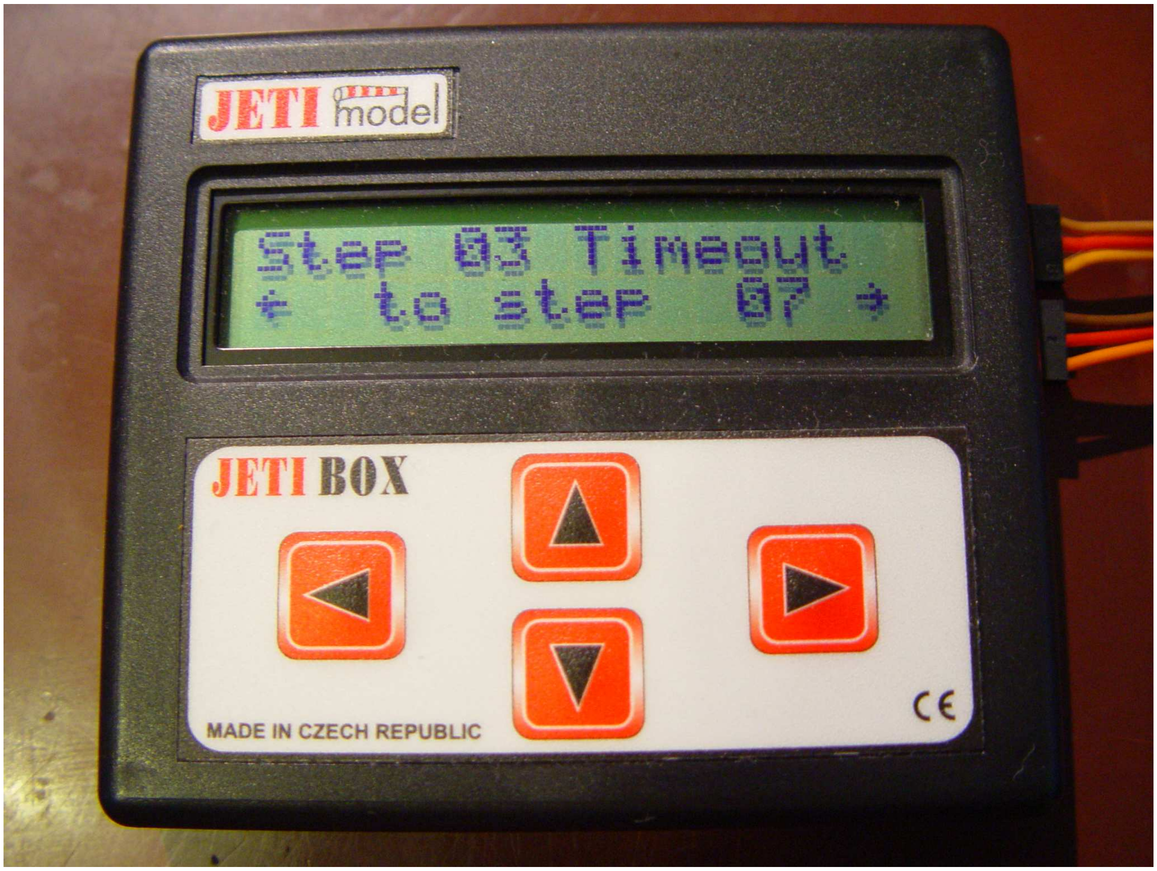
```
Step 03
←  Timeout       →
```

Command "timeout" interrupts the program in the case of the timeout. It is used for example for waiting for change on the logical input.



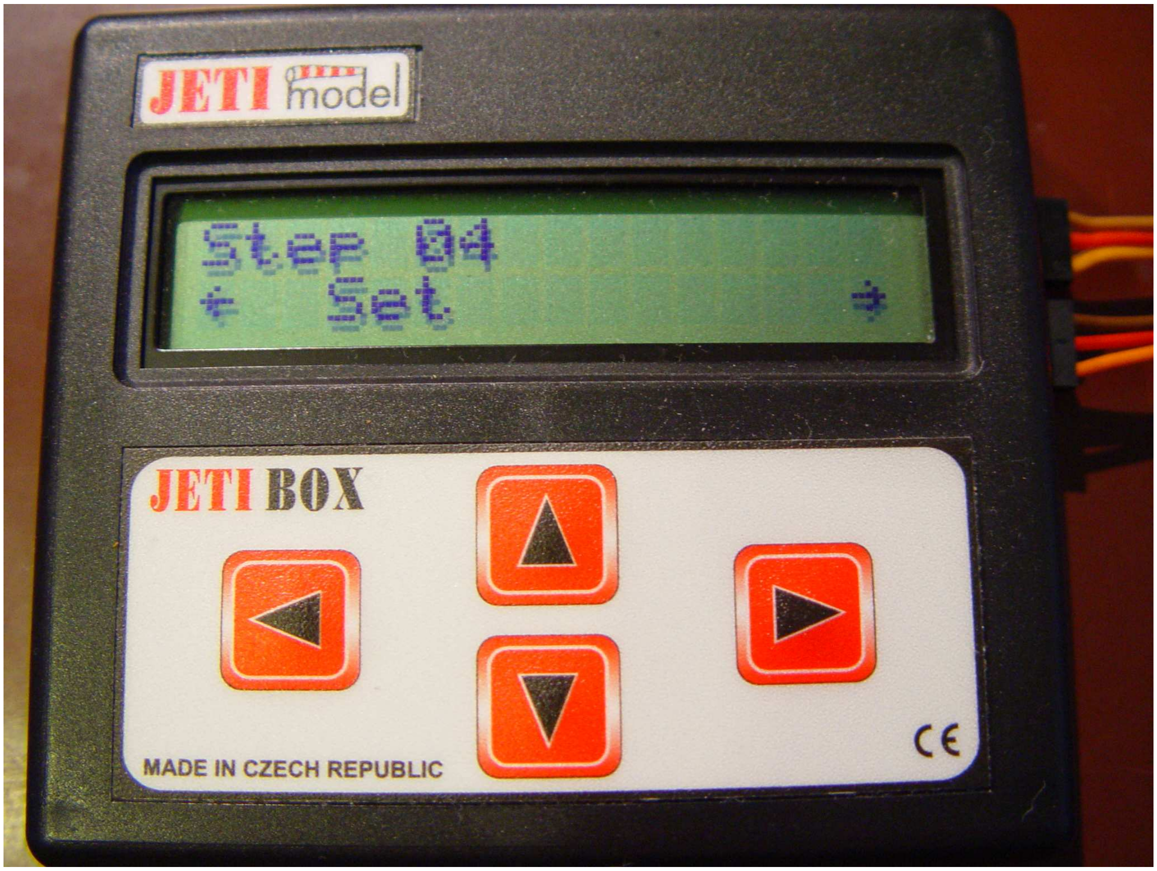
Step 03 Timeout
← time 01:25.0 →

Timeout interval, counted from actual time, when the command was executed. It is possible to cancel or modify the timeout by following command “timeout”.

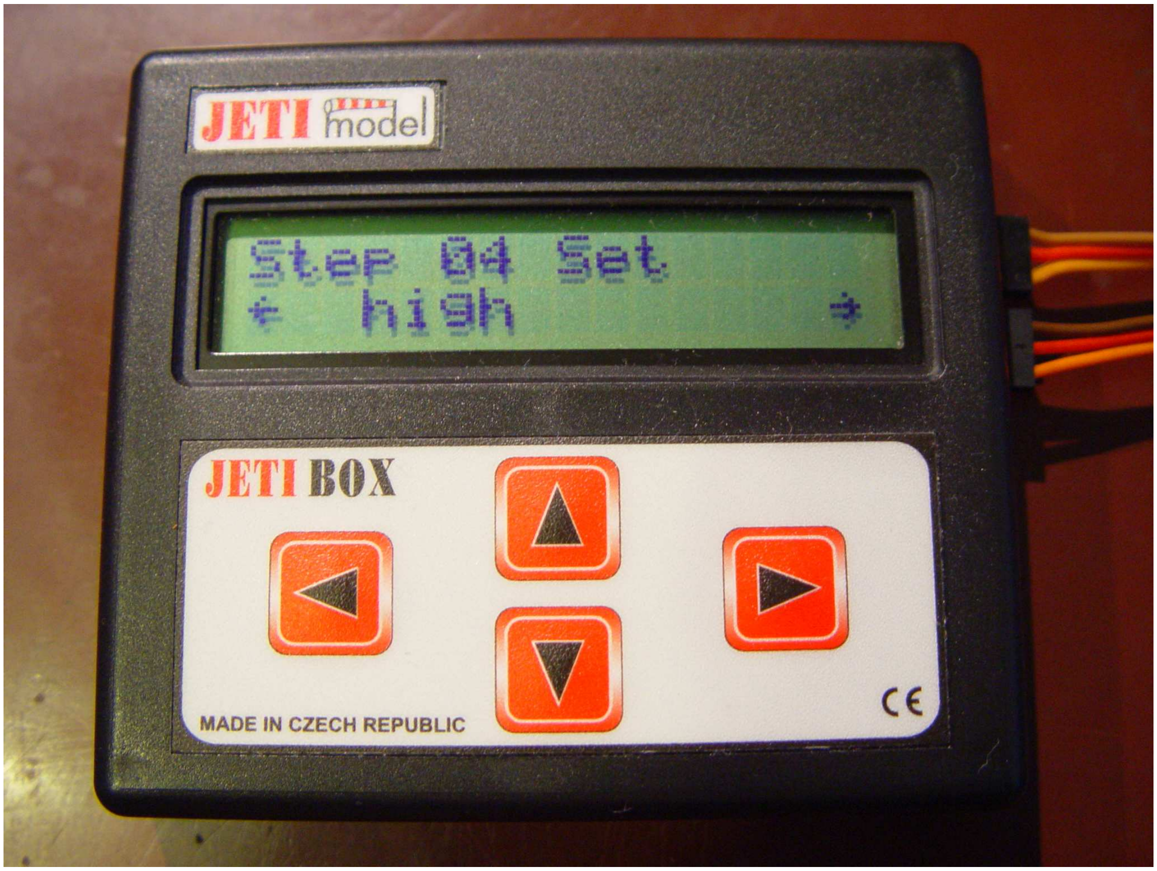


```
Step 03 Timeout
←  to step   07 →
```

Destination address after timeout.



Set of the logic output.



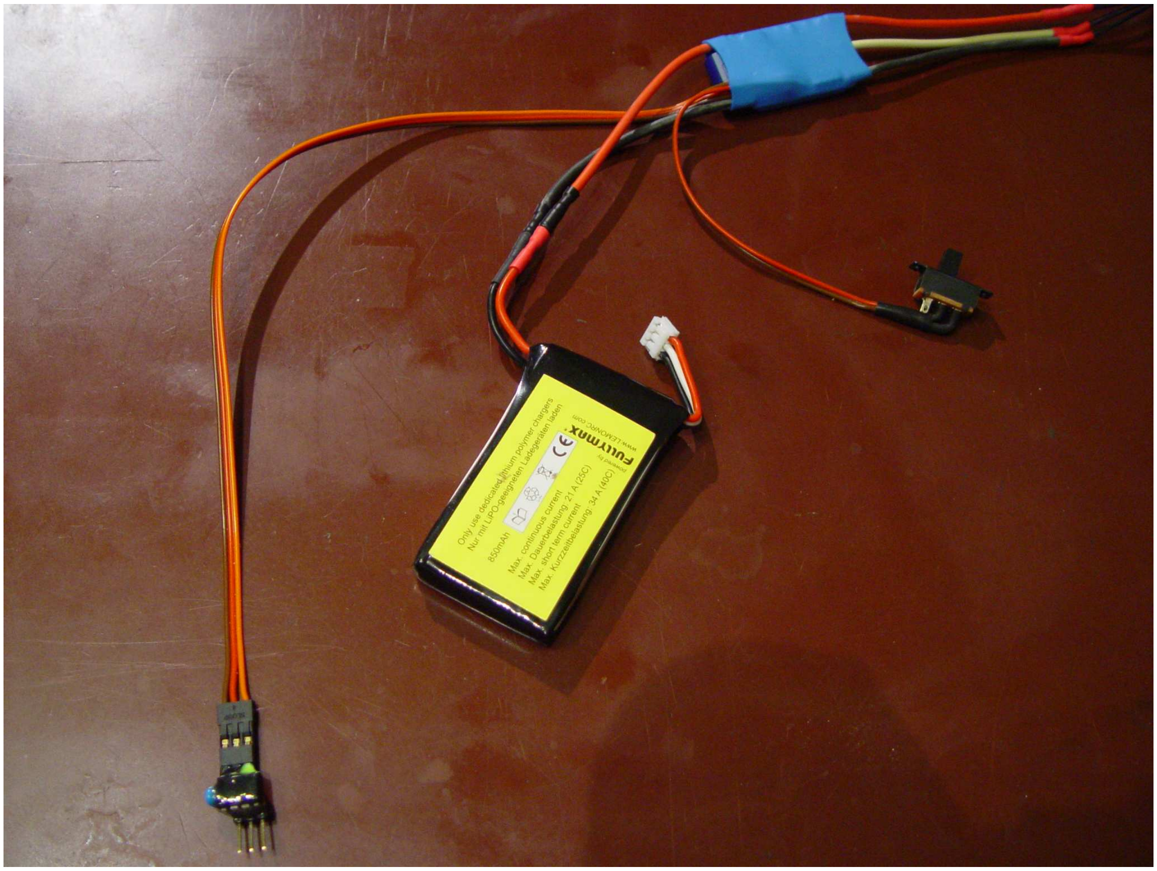
Step 04 Set
← high →

The value of the logic output after the execution.

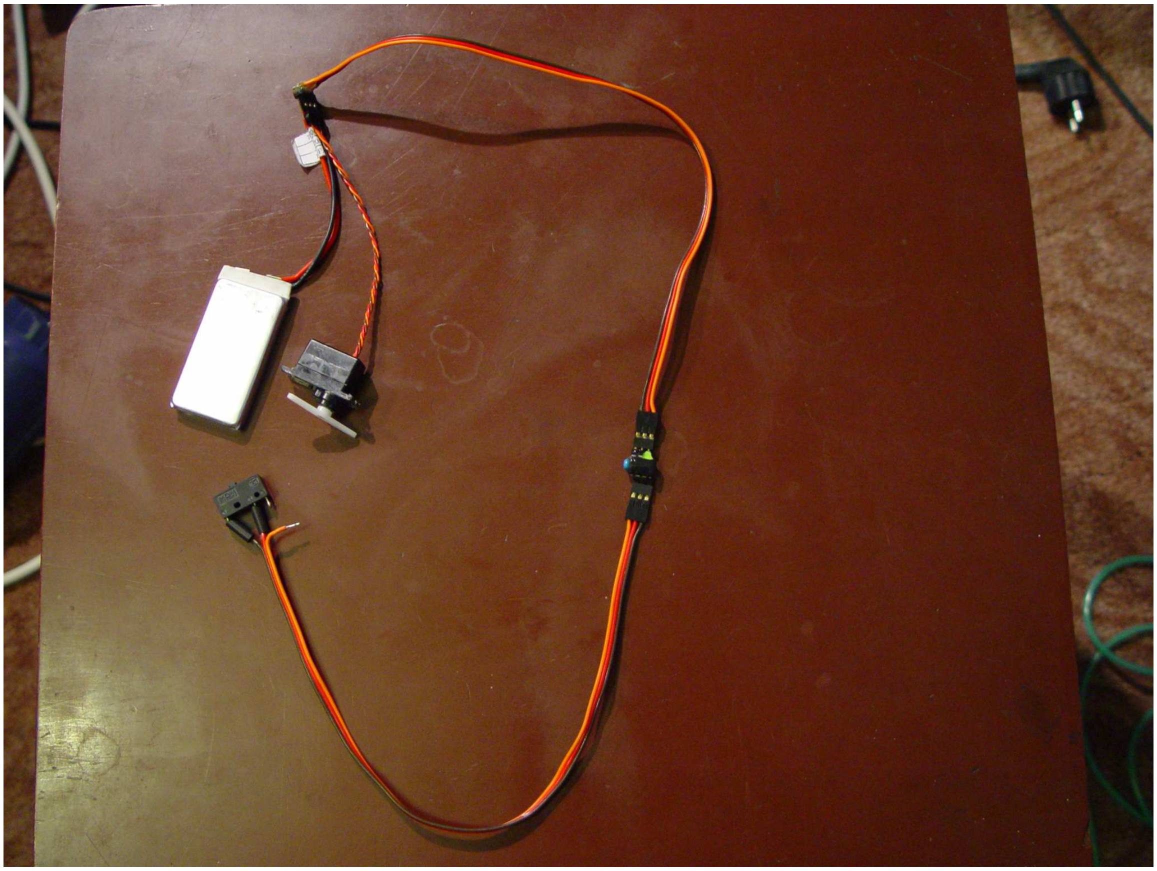
Timer usage.

Connect the timer to the servo connector so that the pin marked by colored dot, connects to the servo signal pin. The same connector is also power supply.

Wiring example



to the regulator of the C/L model

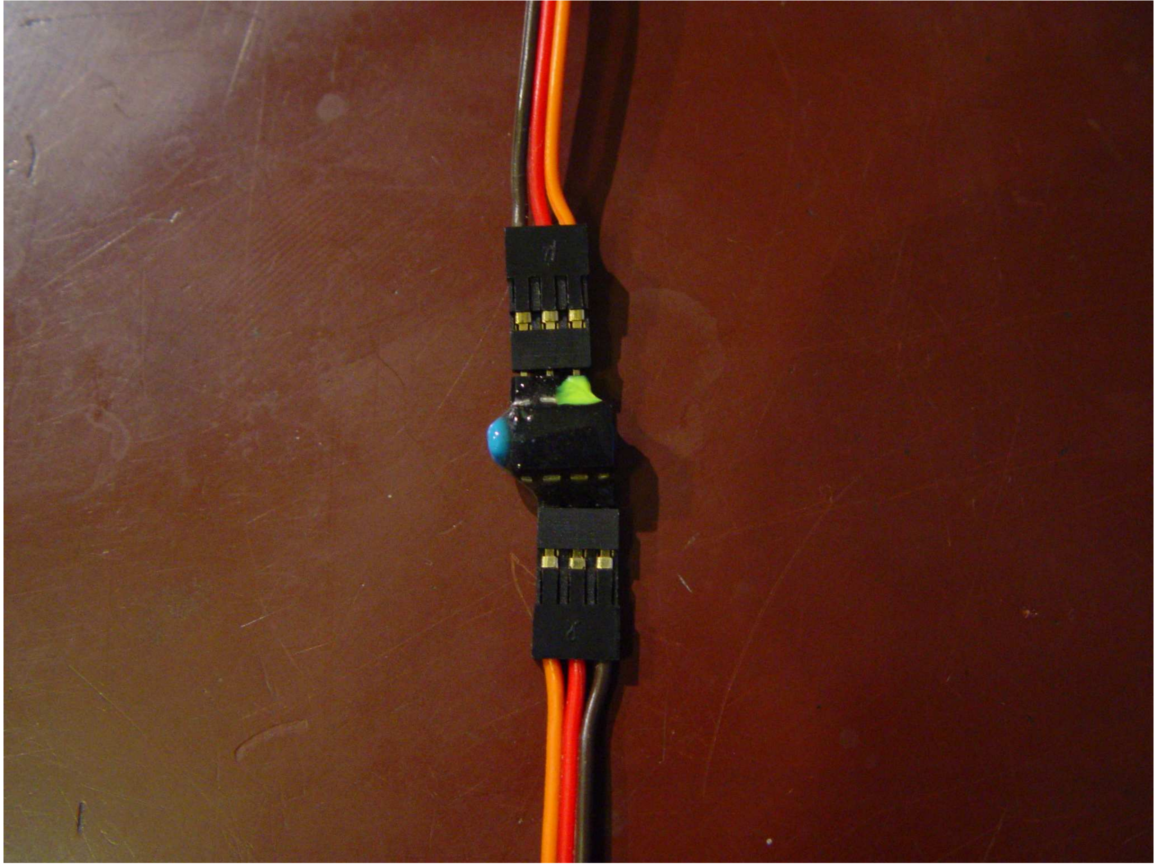


to the free flight model with controlled elevator and the power supply from one LiPol cell. (power supply voltage must not exceed 5.5 V)

External inputs and outputs are on unmarked connector. Right most pin is 0V, the next logical input and third pin is the logical output.

Pins on upper side:

0V +5 V servo signal



Lower side:

Logic output logic input 0V

Instruction set.

Command: "Interval"
Parameter: "Time", "value"

Command is executed so as to adjust the servo position (or engine throttle) to "value" and the program waits for the "time" period. "Time" can be set as "unlimited", so program will stop there. It is used as the last command of program or method of waiting for an asynchronous event ("int", "timeout").

Command: "Jump if"
Parameter "condition", "Address"

Command is executed, so that the program jumps to the target "address" (destination command number) if condition for the jump is true. Conditions of "high" and "Low" values are a logical entry at which the jump has to be made. If the set condition is not met, the program immediately follows to the following command.

The value of "int" does not mean any direct effect, but in the event that on logical input appears logical "low" timer cuts running program, and continues on step in parameter "address". This results in disabling of interruptions, so if necessary, interrupt must be adjusted again.

Logical input is internally supplied to 'high' by internal pullup resistor, so interrupting needs only connection of the logical input to 0V.

Command: "Timeout"
Parameter: "Time", "Address"

Command has no direct effect, but in case of a timeout timer makes a jump from any place of the program to "address". This results in eliminating timeouts settings, so if necessary, it must be adjusted again.

The following command "timeout" resets the previous one.

The value of "time" unlimited" means eliminating timeouts.

Command: "Set"
Parameter: "value"

Command is executed so that the logic output is set to "value".

Example of program for C/L model.

01: startup delay 30s before take off
Interval - time: 00:30 s - value: 000

02: 5 minutes flight time at 75% power
Interval - time: 5:00 s - value: 150

03: landing with stopped engine
Interval - time: unlimited - amount: 000

Example of program for the C/L model with start button.

01: waiting for pressing of the push button connected between logical input and 0V
Jump if - value: high - address: 01

02: short gentle prop spin up
Interval - time: 00:02 s - value: 050

03: startup delay 30s before take off
Interval - time: 00:30 s - value: 000

04: 5 minutes flight time at 75% power
Interval - Time: 5:00 s - value: 150

05: landing with stopped engine
Interval - time: unlimited - amount: 000

Example of program for FF model F1A.

01: starting timeout 10 minutes for the case of line breakage (if line does not disconnect till 10 minutes from start)
Timeout - time: 10:00,0 sec - address: 06

02: set elevator to position for flight with line (time in this case is not important)
Interval - time: 0.1 s - value: 020

03: waiting for disconnecting – input is "high" during the lift with line (hook is closed), opened hook after disconnecting connects input to 0V
Jump if - value: high - address: 03

04: set the tail of the normal flight
Interval - time: 03:00,0 sec - Value: 010

05: siren on
Set - Value: High

06: determalizator and end run
Interval - time: unlimited - amount: 200

Expanding.

If necessary, several timers can be connected in parallel and thus their programs start the same time. Power supply is connected from one source and servo signals control several devices. In this case, the synergy depends only on properly timed programs.

Parallel connected timers can also share a common input, for example pin actuated by the same contact, sensor or button. This allows synchronization together with by interrupts or conditioned jumps.

Logical input and output can be also used for communication between timers. It allows building master / slave systems of timers.