

# REDALS – 01

## *Rocket Engine Data Acquisition and Logging System*

### Description and User's guide

#### **Table of content:**

1.	Device description.....	2
2.	Hardware .....	3
3.	Front panel.....	4
4.	Connectors / Interface .....	4
5.	Sensors .....	5
6.	RECON Software .....	6
7.	Communication protocol.....	7
	List of commands: .....	7
8.	Firmware upgrade .....	9

Copyright © by Authors:

REDALS	Libor Ulčák	<a href="mailto:ulcak@runbox.com">ulcak "at" runbox.com</a>
RECON	Luboš Kobes	<a href="mailto:kobes@via.cz">kobes "at" via.cz</a>

## 1. Device description

REDALS is an electronic data logger. It was designed for the purpose of logging thrust data measured during rocket motor testing, but it is suitable for other purposes as well. It continuously measures electrical signal and records the measured data as a function of time. Recorded data is later uploaded into PC where it is processed and displayed with the RECON software. During the recording session no connection to the PC is required.

Measured data is recorded into nonvolatile memory (EEPROM) and is available there even the battery has been temporarily disconnected. The input value is measured every 10ms i.e. the sampling rate is 100 measurements/second. The maximum measured value is dependent only on the connected sensor e.g. the load cell. The environmental temperature is measured internally at the start of each individual recording session.

The maximum recording time depends on the memory capacity and is 40 seconds for the standard 64kb memory module and up to 160 seconds for the optional 256kb memory module. If the recording session does not use all available memory, another session can be recorded in sequence. Up to 10 recording sessions can follow, before the recorded data is cleared or uploaded into the PC.

The recording is triggered either manually or automatically, according to the position of the "**Trigger**" selector on the front panel. In automatic mode the recording starts when the input value changes at least 1/650 of the measured range (approx. 0.15%) between two measurements (within 10ms) and this increase continues for at least 8 subsequent measurements (i.e. during 0.08 sec.). This algorithm eliminates accidental triggering by generated noise or discontinuous measurement. Before the triggering event is detected any sensor value is stored as the idle value. This is used to detect the end of recording when the measured value drops back to or below this value. This value also serves to tare (bring to zero-level) the graph when the uploaded data is processed in RECON. The time between activation and triggering is limited only by the battery lifetime.

In case the motor thrust increase at startup is slow, or the thrust is too low for the output range of the connected load cell, the start may not be detected correctly. In such case the recording must be started manually by pressing the "**start**" button. The following example shows how to calculate load cell parameters for reliable automatic triggering:

*Thrust measurement of the flare rocket motor 40-R-Os. Expected average thrust 20N, burn time 2 second. Estimated motor ignition in 1/10 of burn time = 0.2 second.*

*In the start-up phase thrust will change from 0 to 20 N in 0.2 seconds. (Peak thrust at the startup not considered) Thus the change of 1N will occur in 0.01second. Just to be sure considered only half of this value i.e. changes of 0.5N in 0.01second.*

*The logger will be triggered by change of >0.15% of the sensor range within 0.01second. When given  $0.15\% = 0.5N$  then the  $100\% = 325N$ .*

*Result: in order to use automatic triggering when testing this motor, the maximal load cell range should not exceed 325N. Otherwise the start will not be detected reliably.*

In order to prevent data loss at startup, measured data is stored into internal RAM buffer before it is flushed to the nonvolatile memory. This buffer holds last 50 measurements. As

soon as the start is detected, the buffer is flushed and so the values of the last 0.5 second before detection are saved. In case the motor startup takes longer than 0.5 second, manual triggering is recommended.

In automatic and manual mode the recording session ends when the memory is full or when the "*start*" button is pressed again. In automatic the recording session ends also when the measured value drops below 0.15% above the idle value captured at start detection and lasts in this state for longer than 0.5 seconds. Power breakdown during the recording session leads to invalid data. Such data can not be uploaded and is ignored.

The memory cleanup is done either via the connected PC or manually by pressing the "*start*" button before the power switch is turned on and holding the button pressed for at least next 5 seconds until the green LED lights up. When releasing the button, the LED goes off again.

A single 9V battery block supplies the device with power. LED on the front panel will light up when battery voltage is too low (below 7,7V). Alkaline or rechargeable NiCd or NiMh batteries are preferred against low cost carbon-zinc battery.

## 2. Hardware

The device utilizes 8-bit microcontroller Atmel ATmega8 with RISC architecture and 8MHz clock. See <http://www.atmel.com/> for detailed description. The program is stored as firmware in 8kB EEPROM memory. Firmware upgrade is possible from PC connected via standard cable to the serial port. The boot loader, necessary to perform the upgrade function, is stored in protected memory during the assembly process.

The analog signal from the sensor (load cell) is amplified by AD623 amplifier up to the required level and digitalized in 12-bit A/D converter LTC1286. This value is periodically sampled by the Atmel microcontroller, digitally filtered and recorded into memory. The temperature is measured by the internal sensor LM355 and digitalized in a 10-bit A/D converter. The precision of this sensor is  $\pm 4^{\circ}\text{C}$  and thus the measured value is for information purpose only.

One internal 5V regulator supplies the digital circuits: microcontroller, RAM, EEPROM memory and TTL/RS-232 interface. Another stabilized 5V regulator supplies the analog circuits: amplifier and A/D converter. The separation and stabilization of the power supply ensures high measurement precision and resolution.

Characteristic:

- Sensor power supply: 5,0 V
- Sensor minimal input resistance 120 Ohm
- Sensor output voltage: 0 – 5,0 mV.
- Dynamic range of the A/D converter: 12 bit. (0 – 4095)
- Effective measured range: 250 - 4050.
- Power consumption (analog circuits): 5,0 V /  $\sim 2,5$  mA
- Power consumption (digital circuits): 5,0 V /  $\sim 26$  mA.
- Battery voltage: min 7,7 – 12 Vss.
- Dimensions (BHL): 74x29x118 mm

### 3. Front panel

**Power** switch On / Off

**Trigger** switch between automatic and manual triggering method.

**Com.TX** orange LED flashes when data is transmitted to or from PC over the serial port.

**Battery Low** red LED lights up when the battery voltage drops below 7,7V

**Start** button,

- starts immediate recording in manual mode
- starts waiting for triggering event in automatic mode
- stops recording in automatic or manual mode

**Start** green LED (in the button socket)

- Flashes slowly when the automatic triggering is turned on and the start is not detected yet.
- Lights up when data is recorded or during firmware upgrade.
- Flashes fast on internal or on communication error.
- Lights off when recording is finished.

### 4. Connectors / Interface

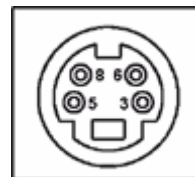
The interface to the PC is standard RS-232, full duplex, 19.2kb, no parity, 8 bit, 1 stop bit.

The cable is standard serial **extension** cable with DSUB9F – DSUB9M (male/female) connectors. Only pins 2,3 and 5 are used.

The connector to the sensor is 4 female pin miniDIN-4 connector as shown below.

The pin layout is:

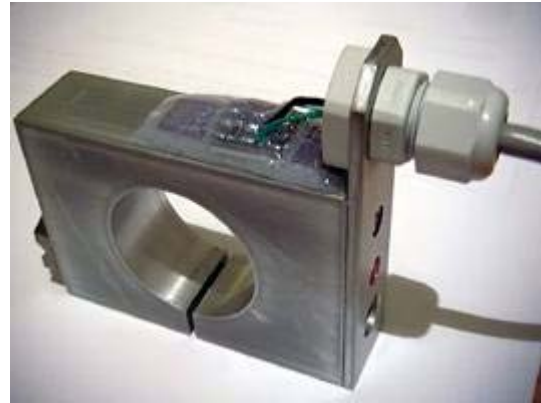
- Pin 3 sensor power supply (-5V)
- Pin 5 sensor power supply (+5V)
- Pin 6 sensor signal (+)
- Pin 8 sensor signal (-)



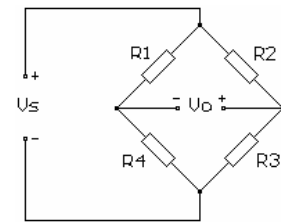
## 5. Sensors

The device was developed to measure force (thrust) produced by a rocket motor. Therefore a load cell was used as sensor to measure the force and convert it to electrical signal. Other applications for example pressure or temperature measurement are possible but require other sensors matching the device input specifications.

Load cells are commonly constructed as a metal block with a predefined elasticity. They are deformed in one axis by a mechanical stress of an acting force. The maximal force must be limited, otherwise the deformation is permanent and the load cell is destroyed. The deformation is converted into electrical signal with strain gage (tensometer), a thin wire strip, which changes its resistance while it is strained. Resistance of a wire is dependent on its length and temperature. At constant temperature and cross-section, the resistance is proportional to the wire length.



The resistance change is very small. In order to increase the sensitivity, the Wheatston bridge circuit is used. The resistors  $R_1, R_2, R_3$  must have the exactly the same temperature coefficient as the tensometer  $R_4$ , otherwise the measurement precision is not sufficient. In order to compensate the negative impact of the temperature, special materials and 4 identical tensometers are used instead. The linearity and precision depends on the design and many electrical and mechanical factors. Every load cell must be calibrated before the first use. The construction of an own load cell requires mature skills in electrical and mechanical engineering. The picture above shows a semi-professional load cell built by the authors. If you really want to build your own one, read these documents:



- <http://www.erde.se/lcguide.pdf>,
- [http://www.seas.upenn.edu/~meam247/fall/straingage/gage\\_manual.pdf](http://www.seas.upenn.edu/~meam247/fall/straingage/gage_manual.pdf)
- [http://www.vishay.com/brands/measurements\\_group/guide/index.htm](http://www.vishay.com/brands/measurements_group/guide/index.htm)

You can also buy one "off the shelf". Many load cells for different purposes are commercially available, industrial weighing is the most common one. You can choose out of dozens of products and compare prices. The advantage of a commercial product is its guaranteed quality and reproducibility. The disadvantage is its much higher price. Do not buy used one! You never know if it was overloaded. If you want to buy one, look at:



- <http://www.hbm.com/>
- <http://www.omega.com/>
- <http://www.lcmssystems.com/>
- <http://www.sensotec.com/>

You will find many others. Look for products with characteristic matching the REDALS specification.

## 6. RECON Software

The RECON - (REDALS CONtrol) software is the recommended and most comfortable way to communicate with the device and to process the recorded data. It is available as free download from <http://raketky.ebox.cz/redals/recon.zip>.

The installation is simple, just extract the recon folder and its subfolders structure to any location on your disk and click on recon.exe. The graphical user interface and help are available in English and Czech language. (The default it is Czech, use the Menu "Konfigurace" field "Jazyk" and switch the language to "en". You have to do this change only once.) Help is available in HTML format and is readable without starting the program by opening the file *rcnhelpn.htm*.

For execution of the Menus "Load data", "Erase Data" and "Test" the connection to the device via serial **extension** cable is required.

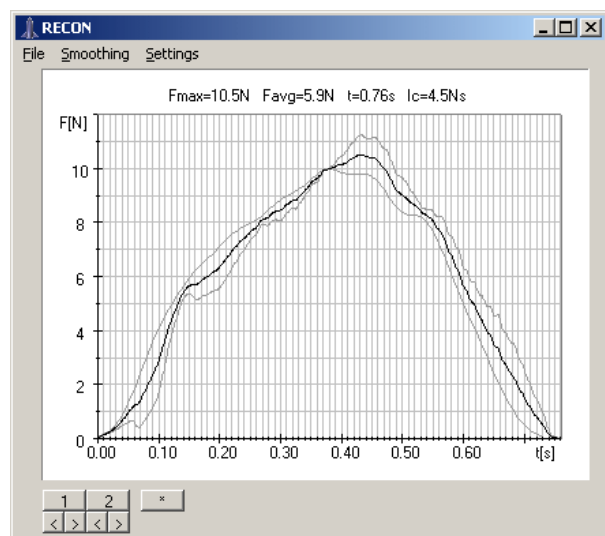
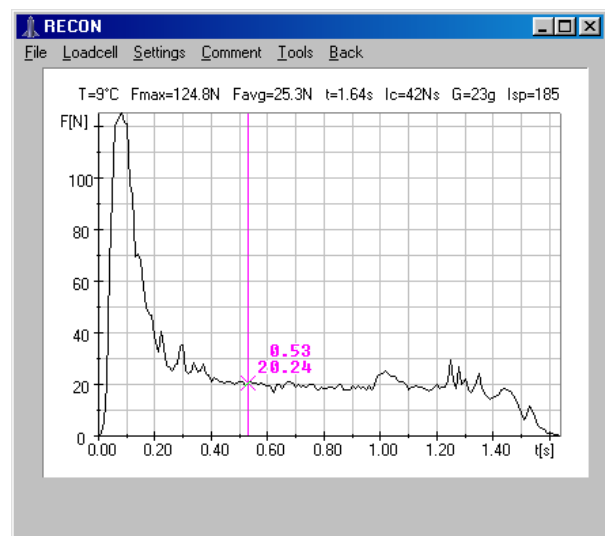
The main features of this program are:

- Upload of recorded sessions from the device into the program. One session is uploaded at once
- Processing of the uploaded data like:
  - o correcting, cutting, smoothing, zooming of the raw data
  - o correlation of data with the load cell characteristic
  - o adding comments and a ruler to the graph
  - o creation of average graph from multiple recording sessions
- Save of raw uploaded, processed or final graph data into files which can be loaded later.
- Memory Cleanup
- Test mode with display of actually measured data in 0.2 seconds interval.

Other features:

- Configurable size of graphs produced
- Export in BPM and GIF format
- Examples of real measured data and graphs to play with

More detailed description is available in the help file.



## 7. Communication protocol

It is also possible to use standard terminal program and basic textual commands to communicate with the device. Connect the serial cable to the PC and to the device and start the terminal program. Take care to set the characteristic of the serial line correctly (full duplex, 19.2kb, no parity, 8 bit, 1 stop bit)

After the device is powered the ComLED will flash several times and the following or similar message will appear on your PC.

```
REDALS-01 v1.3 * Bootloader v1.0 * (c) L.Ulcak, 2005  
Supported EEPROM size: 64kb
```

The control commands are case sensitive and have always three characters. Each command begins with "#" sign, the two following characters designate the action and the command is sent to the device by pressing <CR> ("Carriage Return"). Regular answers are ended with <CR>, so the cursor stays on a new line when the action is finished. RADALS answers with the "?" sign to unknown commands. And with "!" sign for errors. The <BS> ("Backspace") character is not accepted, thus typos cannot be corrected. Just hit twice <CR> and ignore the answer. The following example shows the come dialogs (red = commands, blue = device response).

```
#Rt  
?  
#RT  
+27°C  
#RI  
Recorded logs: 1  
Rec time left: 20.6 sec  
#EA  
Erasing...  
Verifying...  
OK  
#R4  
Log not available!
```

When you press the "**Start**" button and the automatic triggering is turned on, the green LED in the start button will start flashing. The environmental temperature has been captured and the logger is awaiting the triggering event. If the green LED does not flash two situations are possible:

- a. The device is in internal test mode
- b. The EEPROM is full

### List of commands:

*#TE* – Start test mode

From now on the device sends continuously actual values measured by the A/D converter in the interval of 200ms. Each value is on separate line and has 4 digits.

0250  
 0250  
 0251  
 0253  
 ...

**#TX** – *End test mode*  
 Switch back to normal mode.

**#RI** – *Read session information*  
 Reads the number of recorded sessions (logs) and the remaining time available for recording.

*Recorded logs: 0*  
*Rec time left: 40.8 sec*

Possible errors are:  
*I2C error!* – hardware error on data bus EEPROM.  
*I2C timeout!* – communication error on data bus EEPROM.

**#Rx** – *Read session x*  
 Reads the recorded session x where x = 0-9. Data is sent back in two byte hexadecimal numbers separated by comma. The first four numbers have special meaning.

- 1-st number is the number of input channels (0001h)
- 2-nd number is the sampling rate in Hz (0064h = 100Hz)
- 3-rd number is the temperature in K at recording begin (0125h = 293K = 20°C)
- 4-th number is the idle value at recording begin

The number series ends with <CR>.

*0001,0064,0125,00B9,00BA,00BB,00BA,00BA,00BB,00BA,00BB,00BA,00BA,00BA,00BA,00BA,00BA,00BC,00BA,00C4,00E6,014A,01DC,01BD,01DE,0203,020C.....*

Possible errors are:  
*Log not available!* – This recording session does not exist.  
*I2C error!*  
*I2C timeout!*

**#RT** – *Read temperature*  
 Reads the actual measured temperature in Celsius .

*+27°C*

**#EA** – *Erase all*  
 Deletes all data in EEPROM.

*Erasing...*  
*Verifying...*  
*OK*

Possible errors are:



*I2C error!*

*I2C timeout!*

*Verify error!* – Data in EERPOM cannot be deleted.

#### **#VI - Read version info**

Reads the actual firmware version and the EEPROM memory size in **kilobits**. Installation of larger memory requires firmware upgrade. Otherwise the new memory is not used.

*REDALS-01 v1.3 \* Bootloader v1.0 \* (c) L.Ulcak, 2005*

*Supported EEPROM size: 64kb*

If the boot loader is missing, the first line reads

*REDALS-01 v1.3 \* Bootloader N/A \* (c) L.Ulcak, 2005*

and the firmware upgrade cannot be done. In such case AVRprog software and a special harness is required to install the boot loader.

## **8. Firmware upgrade**

The firmware upgrade is possible with the AVRprog program and the serial communication cable. This service program is part of AVR Studio and is free download from the Atmel home page. <http://www.atmel.com/>

- In order to start firmware upgrade, you must open the device by unscrewing the 4 top crews and removing the front panel.
- Then short cut the jumper "**Prog**" beside of the microcontroller and power on the device. This will bring the device in firmware upgrade mode.
- After power on the green Start-LED will light up continuously signaling the firmware upgrade mode.
- Start the AVRprog program. It will automatically detect the communication port where the device connected.
- Switch to program mode and you should see "*Atmega8 BOOT*" on the bottom window border.
- Use the browser to locate the file (.hex) with the firmware and then use button "Programm" in under the heading "Flash" to send the file to the microcontroller.