

**Optimized MASS device for
synchronous measurements with
Paranal DIMM
Electronics and Device control**

Kornilov V., Shatsky N., Shugarov A., Voziakova O.

December 4, 2003

Contents

1	Electronics overview	6
1.1	Common characteristics of electronics and RS-485 line	6
1.2	Detectors unit	6
1.3	Auxiliary electronics	8
1.4	Star centering unit	8
1.5	Requirements to power supply	9
1.6	Data exchange protocol (physical level)	10
1.7	RS485/LPT converter	11
2	Electronic modules	12
2.1	Detectors unit	12
2.2	Bicounter module	12
2.3	Auxiliary module	19
2.4	Stepper motor module	22
2.5	Connections	22
2.6	RS485/LPT converter	24
3	Microprogramming and Instructions set	29
3.1	Logical level of interaction between computer and modules	29
3.2	Bicounter module	30
3.3	Auxiliary module	32
3.4	Stepper motor controller module	35
3.5	Updating a microcode	36

List of Figures

1.1	Schematic view of electronics	7
1.2	Principles of the centering unit function	9
1.3	Illustration of data exchange between host computer and modules.	10
2.1	Circuit diagram of the Detectors unit electronics.	14
2.2	Circuit diagram of the analog part of the Bicounter module electronics	15
2.3	Circuit diagram of the digital part of the Bicounter module electronics	16
2.4	Placement of the components on the PCB of the PMT voltage divider	17
2.5	Placement of the components on the PCB of the Pulse amplifier and discriminator	18
2.6	Placement of the components on the PCB of the digital part of the Bicounter.	18
2.7	Circuit diagram of the main part of the auxiliary electronics.	20
2.8	Placement of the components on the PCB02A of the auxiliary electronics.	21
2.9	Circuit diagrams of the separate parts of the auxiliary electronics	21
2.10	Placement of the components on the PCB02C and PCB02D of the auxiliary elec- tronics	22
2.11	Circuit diagram of the stepper motor controller electronics.	23
2.12	Placement of the components of the stepper motor controller electronics.	25
2.13	Connections between the separate boards of the device	26
2.14	Printed circuit boards for the RS485/LPT	27
2.15	Circuit diagram of the RS485/LPT converter.	28

List of Tables

2.1	Specification for detectors unit electronics.	13
2.2	Specification for an analog part of the Bicounter module.	13
2.3	Specification for a digital part of the Bicounter module.	17
2.4	Specification for the Auxiliary electronics.	19
2.5	Specification for the Stepper motor module.	24
2.6	Bus colors table	24
2.7	Arrangement of line cable	25
2.8	Specification for the RS485/LPT converter (see SCH02).	27
3.1	Used signal bytes.	29

Bibliography

- [1] Kornilov V., The optimization of MASS device for synchronous measurement with Paranal DIMM. A Proposal to European Southern Observatory (ESO). December 11, 2002
- [2] Kornilov V., Potanin S., Shatsky N., Voziakova O., Zaitsev A. *Multi-Aperture Scintillation Sensor (MASS). Final design report.* February 2002.
- [3] Kornilov V., Potanin S., Shatsky N., Voziakova O., Shugarov A. *Multi-Aperture Scintillation Sensor (MASS) Upgrade. Final report.* January 2003.
- [4] Kornilov V., Shatsky N., Voziakova O., *The optimization of MASS device for synchronous measurement with Paranal DIMM. First stage project report* April 2003.
- [5] Kornilov V., Tokovinin A., Voziakova O., Zaitsev A., Shatsky N., Potanin S., Sarazin M. MASS: a monitor of the vertical turbulence distribution. Proc. SPIE, V. 4839, p. 837-845, 2003
- [6] A.Tokovinin, V.Kornilov, N.Shatsky, O.Voziakova, *Restoration of turbulence profile from scintillation indices*, MNRAS 2003, V. 343, P. 891

Introduction

The document presents description of the electronic modules of the optimized MASS device. The base of the electronics design is the same as in original MASS: modular structure, data exchange via RS-485 interface, PMTs as light detectors. Meanwhile, the real electronics were significantly redesigned: number of separate modules was reduced from 7 (in original MASS [2]) to 3 in the current design.

First Chapter of this document contains general description of the electronics as well as overview of data exchange protocol between the device and PC.

Detailed design of the MASS electronic modules (circuit diagrams, printed circuit board views, their specifications) is provided in the second Chapter. Although the description of RS485/LPT converter is available in [3], this information is included in the document also.

Last Chapter contains detailed description of the low level command set, which can be used during handling of non-standard situations, which can arise in test, adjustment or repair process. The description of the microcode update procedure is presented, too.

The document contains the detailed information on MASS device, which may be needed in case of device malfunctions or fault. The information will be useful for exact understanding of the device possibilities and potentials.

Chapter 1

Electronics overview

The MASS electronics inherits the modular principle of the original MASS device; the photometric modules were united in one detectors unit, some functions of other modules were redistributed due to partial modification of the device construction. In order to promote the device reliability, the number of external and inter-PCB connectors was minimized. The basic Atmel AVR microcontroller model was changed from AT90S2313 to more powerful and advanced AT-Mega8, meanwhile the communication protocol of data transfer from MASS to PC was retained unchanged to maintain the software compatibility with the original MASS system.

1.1 Common characteristics of electronics and RS-485 line

As in original design, the architecture of all MASS electronic modules is similar. The kernel of any module is an AVR micro-controller ATMega8 from Atmel company running at 14.746 MHz frequency. In principle, such a clock frequency provides the standard transmitting rate as large as 1840 Kbit/sec instead of 460.8 Kbit/sec in original MASS. A big merit of these controllers is the possibility of their re-programming using the data exchange line. All modules are designed to support this possibility.

Schematic view of the MASS electronics is presented in Fig. 1.1. All information exchange between the host computer and the individual modules is executed via RS-485 line working in half-duplex mode. The line connects to LPT port of PC via a special RS485/LPT converter.

A pure RS-485 interface is used in the segment "host computer — MASS". Balanced data lines A and B are properly terminated and biased from both ends of the long cable to provide safe-fault data transmission. For inter-communication between the electronic modules, RS-485 itself and 2 additional lines are used. These additional lines do not have constant specification and are used to transmit specific local signals: SYNCHRO — for common hardware synchronization of the modules, and OVERLIGHT — for fast hardware protection of PMTs against light overflow.

1.2 Detectors unit

MASS detectors must measure the intensity of light in four channels synchronously with a very short (1 ms) exposure time and high duty cycle. The number of photons detected in an elementary exposure in each area goes up to tens of thousand depending on the channel and star brightness. Preliminary investigation shows that most suitable PMT is bi-alkali Hamamatsu

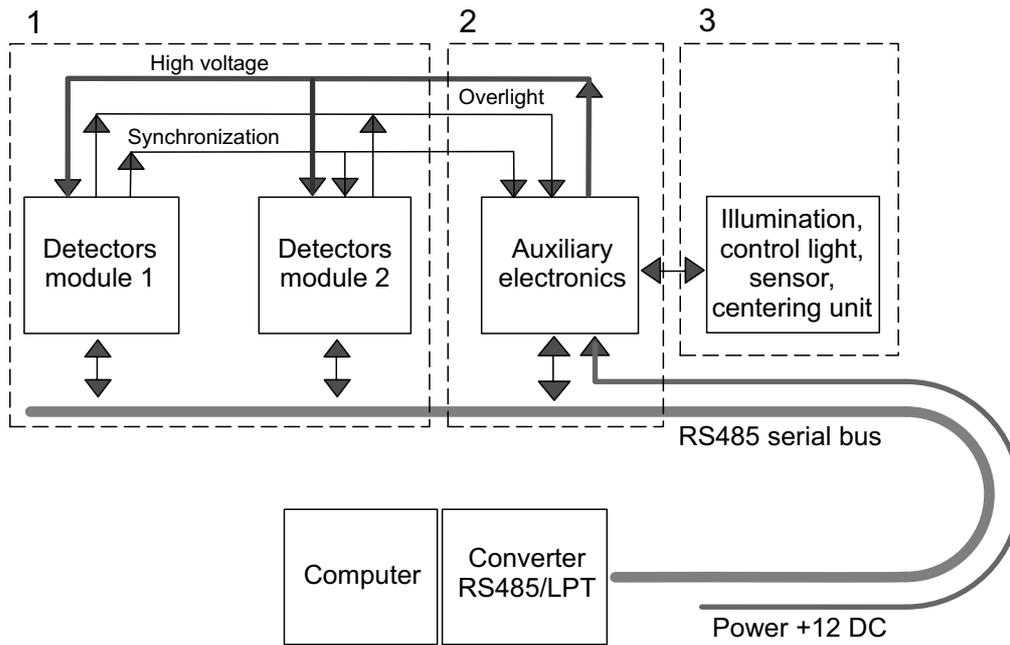


Figure 1.1: Schematic view of electronics for MASS device. 1 — detectors unit, 2 — electronics box, 3 — electronic elements in main MASS case. Centering module connects with RS-485 line via auxiliary electronics

R7400P (see Report [4]). It has a low dark current (< 100 pulse/s), high sensitivity in blue-green spectral region, suitable temporal characteristics and very compact size.

Contrary to original MASS, where the detectors were implemented as four separate photometric modules (PMs), four MASS device detectors are united in one detector unit, including both PMTs and the associated electronics. Detector electronics is subdivided in two independent two-channel modules (further — Bicounter module). Nevertheless, these modules are placed at one PCB, which permits to diminish a number of inter-connections.

Each bicounter consists of two voltage dividers for PMT, two very fast amplifier- discriminators, two counters driven by a single micro-controller and an interface circuit. Both data from the module to a host computer and commands from a computer to the module are transmitted via RS-485 line. Interaction between PM and computer is described in Section 1.6. Besides, an additional line connects the photometric (bicounter) and power supply modules and immediately shuts down the HV when the PMT flux exceeds the maximum rating. This feature assures the safety of PMTs.

The bicounter module executes the following functional commands: set level of the pulse discrimination, run series of microexposures with a preset exposures number and integration time, set needed integration time or work by external synchro, set length of series, and so on. Data packet from bicounter contains always counts from both PMT.

Physically, the detector unit is attached to the electronics box and they can be removed from the MASS device together. Therefore, electrical connections between these both units are soldered. Whole electronics unit is fixed at the main box with help of plug elements and locked

by screws.

1.3 Auxiliary electronics

Auxiliary electronics is mainly placed in the electronic box, although a few elements are placed in the main case of the device. The electronics provides following functions:

- Device powering by DC +5 V
- PMT powering by high voltage
- Measurement of internal temperature
- Control of illumination of FOV in viewer
- Polling of viewer mirror position
- Management of control light
- Monitoring of RS-485 line status

From the side of external control, the electronics is one multi-functional module connected to RS-485 line. The module executes the following functional commands: set brightness of both light sources (control and viewer), modulates the control light synchronously with microexposures for statistic test, which is used to test the normal operation of the detectors and to control the parameters of the photometric channels.

Position of viewer mirror is checked to prevent measurements when the mirror is occasionally left on axis.

The high voltage converter TA-1.0N-12LS from WME company produces a voltage from 0 to 1000 V for PMT powering. The converter is powered by +12 DC and supplies 1 mA current with low ripple. The control of the output voltage is done by software as well.

The presence of high voltage is indicated by a special red LED placed near other indicating LEDs. In an emergency (light overflow) the special signal from detector unit turns off the high voltage immediately.

Additional functions provided by this electronics are the controls of internal temperature of MASS and of the status of RS-485 line. Line status (data is transmitted or not) is indicated by a yellow LED.

The secondary power unit (such as DC/DC converter TEM2-1211 from TRACO Power Company) produces DC +5 V for device electronics and for RS485/LPT converter (powered thus via the line).

1.4 Star centering unit

In order to provide effective work of the optimized MASS in situations when DIMM is not used, the star centering unit is included in the design. It also helps to check whether the DIMM and piggy-back MASS devices are coaligned on the same (DIMM's) mounting. Since the new MASS device does not have an aperture wheel, the construction of the centering unit differs from the centering mechanism of the original MASS. The basic scanning idea is similar, but scanning of the stellar image is made with help of a triangle knife rather than by a triangle hole. This method has an important advantage: the star position can be measured with respect to the real

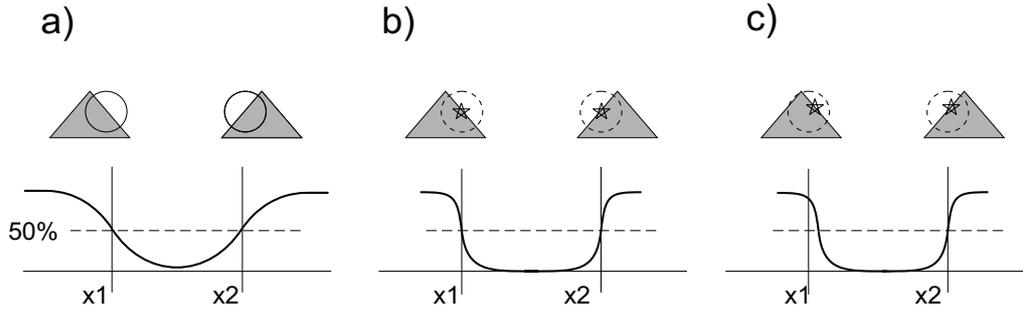


Figure 1.2: Principles of the centering unit function. **a)** — scan of uniform aperture, **b)** — scan of the star image in the aperture enter, **c)** — the same for offset star.

center of aperture (see below), not to some initial star position which needs additional (manual) calibration.

The Fig. 1.2 shows the light curves during scanning of uniformly illuminated aperture, star in the aperture center, and an offset star. Uniformly illuminated field aperture (twilight scanning) serves for defining of the exact aperture center.

To move the triangle knife, a little stepper motor from FDD drive with its native brass worm is used. Bipolar stepper motor has two 18 ohm windings powered by maximal current 200 mA. The motor has 20 step per revolution and can be driven in quarter step mode only, not finer. Pitch of worm equals to about 3 mm. So, the knife shift step as small as 0.04 mm is provided. This value (taking into account the 45° slope of the knife edge) corresponds to the angular step $3''$, which is less than 0.01 of the field aperture size.

A separate electronic module is used to control the centering unit. The stepper motor controller shifts of triangle knife with a preset speed at the needed distance, checks the left and right motion limits, turns off and on motor powering between centering procedures.

Stepper motor is powered by DC +12 V. The module is connected to RS-485 line as shown in Fig. 1.1.

1.5 Requirements to power supply

The power supply +12 DC may be either a battery or a line regulator or a switching converter. The maximal total MASS power consumption is 300 mA when the high voltage is on. The main requirements for the characteristics of the power supply are following:

- Output voltage: +12 V (Min +11.5 V, Max +13 V)
- Max output current: greater than 0.6 A
- Max output voltage pulsations: less than 100 mV
- Operating temperature from -10° to $+35^\circ$.

A 2-wires cable which powers the MASS device, must have 0.5 mm^2 cross-section (AWG20) and the length not longer than 15 m. Voltage drop at the cable must be less than 0.2 V per one wire.

1.6 Data exchange protocol (physical level)

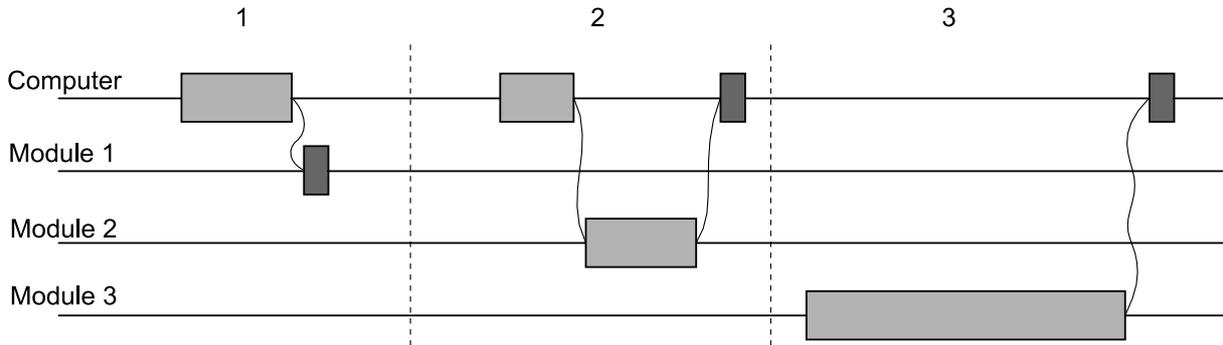


Figure 1.3: Illustration of data exchange between host computer and modules. Light grey — packet with data or command information, dark grey — signal which confirms packet reception. Three possibilities are shown: 1) computer sends a command to module 1, 2) computer sends a request to module 2 and receives the reply, 3) an active module 3 sends a data block to computer.

As described above, all information exchange between the host computer and MASS is executed via RS-485 line. MASS operation needs an informational flow as large as 8 Kb/sec or 110 Kbit/sec on a serial line. So, the line with ≈ 0.5 Mbit/sec is sufficient for our purpose. Although ATmega controller provides more faster exchange rate, the possibility of usage of a faultless line of 50–100 m length at lower rates is more preferable.

Data transmission through RS-485 line is executed serially. Each serial byte contains 1 start bit, 8 data bits, 1 parity bit (used for special needs only), 1 stop bit. It is the standard protocol, defined at the hardware level of micro-controllers. The method of interaction between receiver and transmitter to provide faultless and effective data exchange is called an exchange protocol. The main features of the used protocol of data exchange are following:

- Data and commands are transmitted in binary (non-symbolic) form
- The bytes of information for transmission are merged in a packet
- The inter-module exchange is excluded from protocol, computer always participates in any data exchange
- The packet can have a length from 1 to 31 bytes
- The packet with length of 1 byte has a special function – signal
- Each packet is started with a header byte (except signals). Header byte is marked by 1 in the parity (ninth) bit
- Header contains address of the destination or departure module
- The last byte of the packet is 8 bits cyclic residual control
- Receptor confirms the packet acceptance by sending an acknowledgment signal
- A non-confirmed packet is treated as lost and is re-transmitted until reception

All MASS modules work commonly as passive devices — they can transmit data only in response to a request from the host computer. The photometric module can work as an active

one, i.e. it can activate the packet transmission. This ability permits to reach more effective data acquisition than polling method. In Fig. 1.3 the three possible variants of data exchange are illustrated.

Since the same lines of the interface is used for both reception and transmission, a collision of packets is possible. To avoid collisions, some modification of time windowing method is used. For the needed data flow and transmission rate, the used time window is about 500 mksec.

In general, the collision problem for an exchange rate of 20% of line capacity and for three active devices (two bicoounter modules and host computer) is severe. It was solved in case of MASS by special induce procedure, when the data transmission from next module is started after passing data packet from the so called inductor module. This way is used in MASS, too.

1.7 RS485/LPT converter

To solve a data exchange problem with a needed rate (460 Kbit/s) we use standard LPT port working in EPP mode and a special RS485/LPT converter (see [3]) with a packet processing feature and a large FIFO capacity (512 bytes). Such bufferization permits to reduce by more than two orders the PC reaction requirements and to reduce a processor load at the interrupt service by few tens times.

In order to send a packet to the MASS device, the driver program writes sequentially all the packet bytes to the LPT data register except the address byte. Then, an address byte is written in the LPT address register which signals that the packet is fully loaded in the converter. Converter computes the CRC byte, adds it to the packet and transmits serially the packet via an RS-485 line. Then it waits for the module response and, if no fault results, asks the computer to read the replied data.

When a transmission is activated by a (bi)counter module, the converter receives a full packet, checks its CRC, sends back a signal ACK (or NAK in the fault case) to the module and asks the computer to read received data. If computer is busy and can not read data immediately, the received data are placed into the internal FIFO buffer. The buffer can keep up to 15 full packets and provides thus the bufferization as long as 30 ms. When the computer is finally able to read data, all the data, packet by packet, are input during 1 ms.

The converter uses optical coupling that insulates electrically the PC computer from the MASS instrument. The converter is connected to the LPT port of the PC directly and does not require any separate power supply. It is powered from the line voltage +5 DC fed by the MASS device.

Chapter 2

Electronic modules

The electronics design is performed as the base of the modular conception explained in the Main Document earlier. In the next sections the circuit diagrams of the modules are presented. Generally accepted designations of schematic elements are used, except designation for resistors (we used a russian symbolic for them).

The connectors are also marked in a special way. Connectors to external cable and wires are denoted by a letter "X", internal connectors are divided into four groups: 1) Soldered connectors are marked with a letter "S", 2) Inter-board connectors, which link different parts of the same module — with "I", 3) Connectors to internal bus, which links the different modules — "Y", 4) Special connectors for In System Programming technique are denoted as "ISP".

The nominal values of passive elements are shown on the schemes. The active element features are shown in specification tables which are included, too. Component manufacturer are not shown in cases of widespread parts. Connectors, which are parts of PCB, don't include in the tables.

Also, this chapter contains the schematic views of the module PCBs with the component placement for easy identification of the schematic element with the real component used.

2.1 Detectors unit

As described above, detectors (photometric) unit consists from two identical bicounter modules, placed in parallel at the same PCBs. In the Fig. 2.1 a circuit diagram for whole unit is presented. Bicounter modules are shown schematically by blocks. Components which do not belong to bicounter modules are shown explicitly. Further, these components are marked by prefix * at the PCB01C.

2.2 Bicounter module

The circuit diagrams of the analog and digital parts of the bicounter are shown in Fig. 2.2 and Fig. 2.3. PCB views are presented in Fig. 2.4, Fig. 2.5 and Fig. 2.6.

Table 2.1: Specification for detectors unit electronics.

Item	Part	Name	Manufacturer	Q-ty	Rem
1	D1	IC NC7SZ32M5	Fairchild	1	SOT-23, Gate OR
2	Z1	Quartz 14.746 MHz	—	1	HC49S
3	C1	SMD capacitor 2KV	—	1	1208 size
4	C2, C3	SMD capacitors	—	1	0805 size
5	R1	SMD resistor	—	2	0805 size
6	Y1	Line conn.	—	1	Soldered
7	PCB01A	Printed board	Custom	1	Fig. 2.4
8	PCB01B	Printed board	Custom	1	Fig. 2.5
9	PCB01C	Printed board	Custom	1	Fig. 2.6

Table 2.2: Specification for an analog part of the Bicoounter module.

Item	Part	Name	Manufacturer	Q-ty	Rem
1	D1, D2	IC SA5205AD	Philips	2	
2	D3	IC AD1580ART	Analog Dev.	1	
3	D4, D5	IC AD8400AR1	Analog Dev.	2	
4	D6, D7	IC AD8611AR	Analog Dev.	2	
5	R1-R4	SMD resistors	—	4	0805 size
6	R5-R22	SMD resistors	—	18	1208 size
7	C1, C2	SMD capacitors	—	2	0805 size
8	C3, C4	SMD capacitors	—	2	1208 size, 250V
9	C5-C8	SMD capacitors	—	4	0603 size
10	C9-C19	SMD capacitors	—	10	0805 size
11	C20-C22	SMD Tantal capac.	—	3	A size
12	L1-L4	SMD inductances	Bourns	4	1812 size
13	R23-R33	SMD resistors	—	11	0805 size
14	PMT1,2	E678-12	Hamamatsu	2	PMT sockets
15	I1-I3	Pins connector	—	3	PBS2-2, soldered
16	I4	Pins connector	—	1	PLD2-12

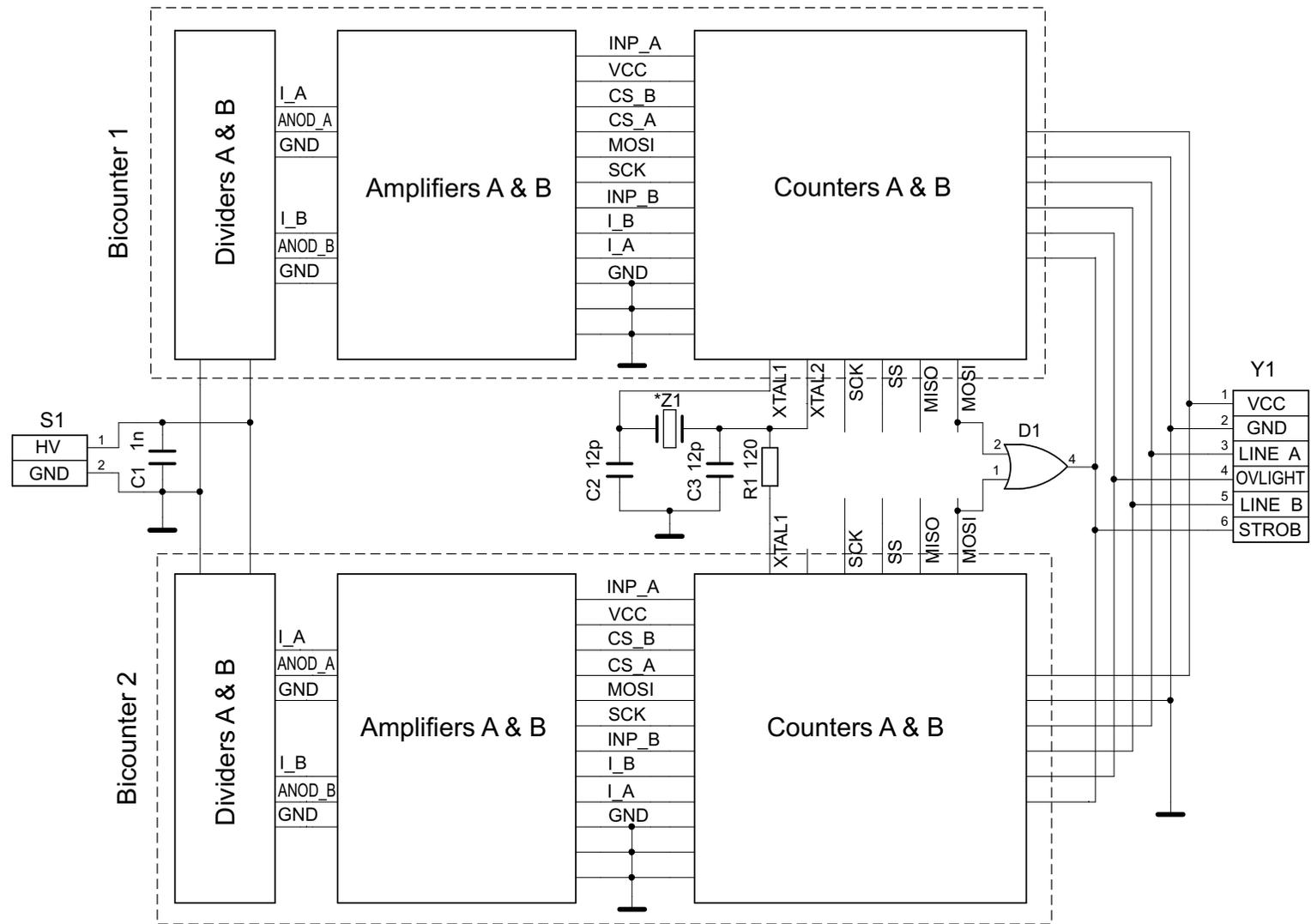


Figure 2.1: Circuit diagram of the Detectors unit electronics. See circuit diagram for a bicounter in the further figures.

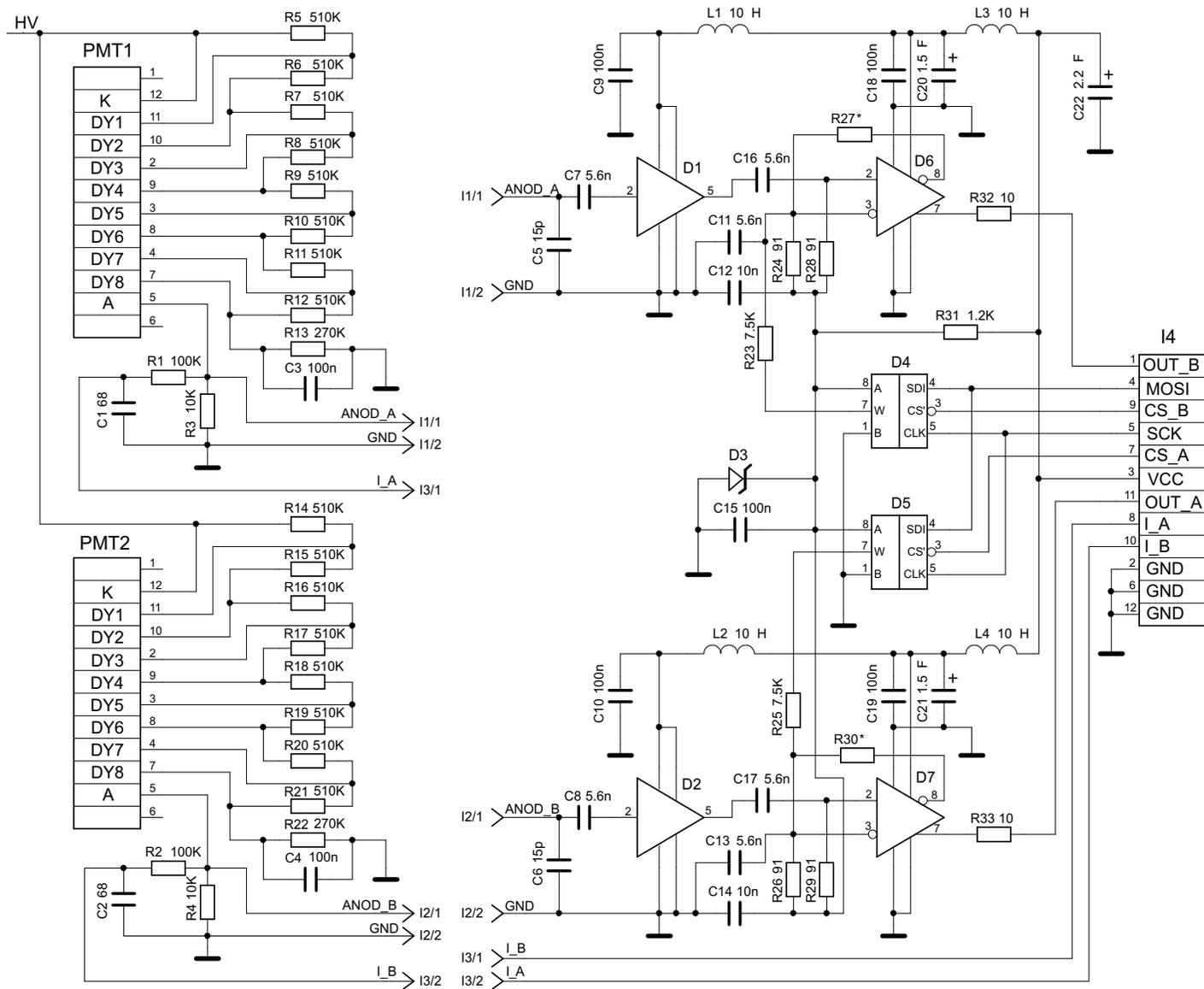


Figure 2.2: Circuit diagram of the analog part of the Bicounter module electronics. PMT divider, pulse amplifier and discriminator with level control.

Table 2.3: Specification for a digital part of the Bicounter module.

Item	Part	Name	Manufacturer	Q-ty	Rem
1	D1	IC 74HC4520D	Philips	1	
2	D2	IC LM2903M	Nat.Semicond.	1	
3	D3	IC ATmega8-16AI	Atmel	1	
4	D4	IC ADM1485AR	Analog Dev.	1	
5	R1-R5	SMD resistors	—	5	0805 size
6	C1, C2	SMD capacitors	—	2	0805 size
7	C3	SMD Tantal capac.	—	1	B size
8	I4	Pins connector	—	1	PBD2-12

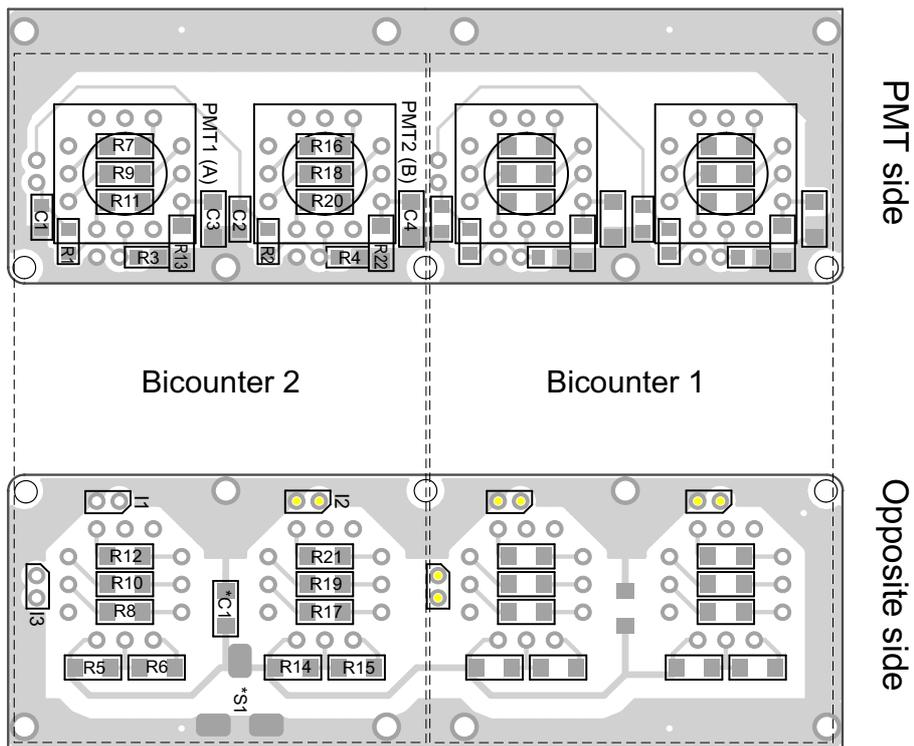


Figure 2.4: Placement of the components on printed circuit board of the PMT voltage divider. Designations are the same as in circuit diagrams in Fig. 2.2.

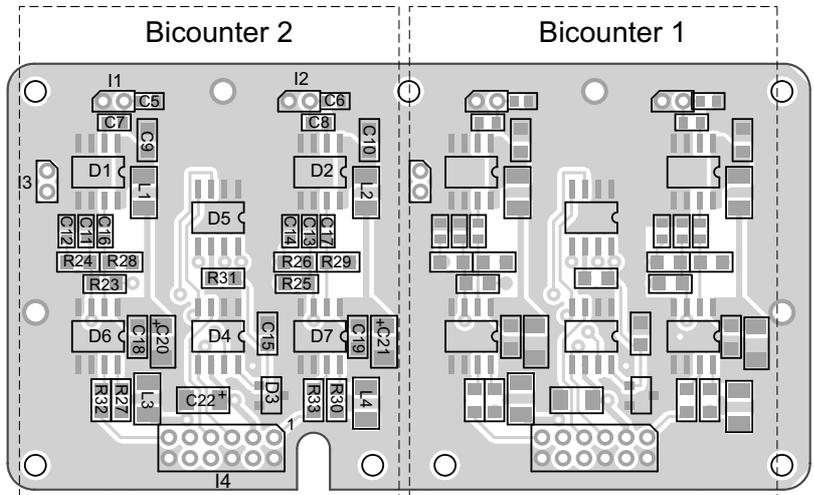


Figure 2.5: Placement of the components on printed circuit board of the Pulse amplifier and discriminator with level control. Designations are the same as in circuit diagrams in Fig. 2.2.

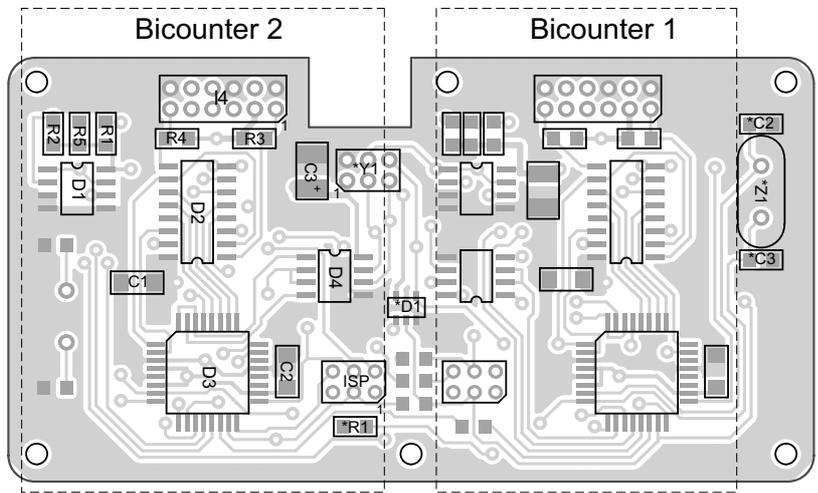


Figure 2.6: Placement of the components on printed circuit board of the digital part of the Bicounter. Designation are the same as in circuit diagrams in Fig. 2.3.

2.3 Auxiliary module

The circuit diagrams of this module are shown in Fig. 2.7 and Fig. 2.9. PCB views are presented in Fig. 2.8 and Fig. 2.10. On the PCB2A main part of the auxiliary module is placed. So as this PCB is installed in removable electronics box, it is connected with other parts of auxiliary electronics, placed in the main case of the device, with help of connector I5. The connector is mounted on cross-plate PCB02B. Further connections made soldered.

Table 2.4: Specification for the Auxiliary electronics.

Item	Part	Name	Manufacturer	Q-ty	Rem
1	D1	Mod. TEM2-1211	Traco Power	1	DIP-24 size
2	D2	IC LM2904M	Nat.Semicond.	1	
3	D3	Mod. TA-1.0N-12LS	WME	1	See spec.
4	D4	IC TMP36	Analog Dev.	1	TO-93
5	D5	IC LM7101BM5	Nat.Semicond.	1	
6	D6	IC ATmega8-16AI	Atmel	1	
7	D7	IC AD8402AR10	Analog Dev.	1	
8	D8	IC ADM1485AR	Analog Dev.	1	
9	D9	IC SS441	Honeywell	1	
10	Z1	Quartz 14.746 MHz	—	1	HC49S
11	R1-R22	SMD resistors	—	22	0805 size
12	C1	Alum.capacitor	—	1	16V, D=8 mm
13	C2, C4	SMD tantal capac.	—	2	B size
14	C5	SMD tantal capac.	—	1	A size
15	C3-C11	SMD capacitors	—	7	0805 size
16	V1-V3	L513(SV,SG,SY)	Kingbright	3	LEDs
17	V4	KPL3015G	Kingbright	1	Green
18	V5, V6	KPL3015R	Kingbright	2	Red
19	ISP	ISP connector	Custom	1	PCB holes
20	X1	Power conn. DJK-02B	—	1	d=2.5mm
21	X2	Line conn. DB9F	—	1	
22	Y2	Pins conn. (pins)	—	1	PLD2-6
23	I5(A)	Pins conn.(pins)	—	1	PLD-12
24	I5(B)	Pins conn.(sockets)	—	1	PBD-12
25	PCB02A	Printed board	Custom	1	
26	PCB02B	Printed board	Custom	1	
27	PCB02C	Printed board	Custom	1	
28	PCB02D	Printed board	Custom	1	

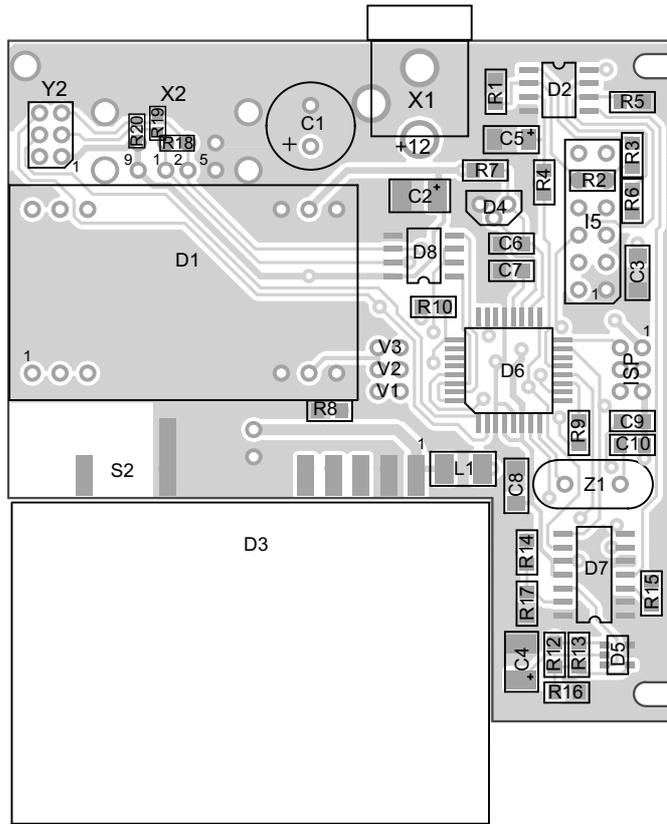


Figure 2.8: Placement of the components on the main printed circuit board PCB02A of the auxiliary electronics. Designations are the same as in circuit diagrams in Fig. 2.7.

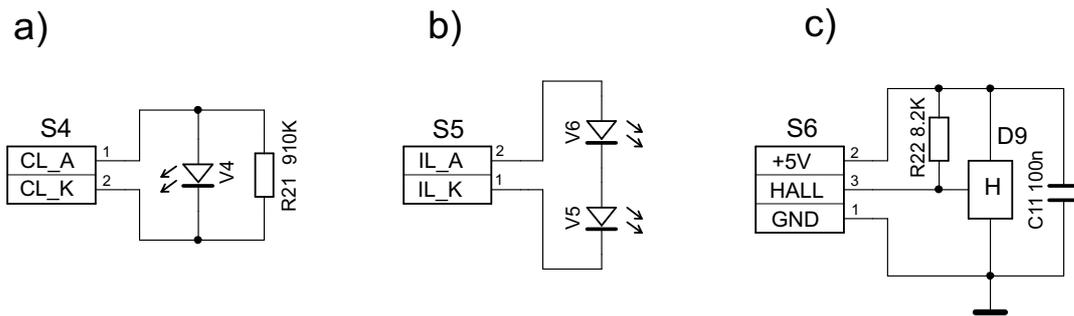


Figure 2.9: Circuit diagrams of the separate parts of the auxiliary electronics: a) control light LED, b) field of view illumination LEDs, c) Hall sensor of the position of viewer mirror.

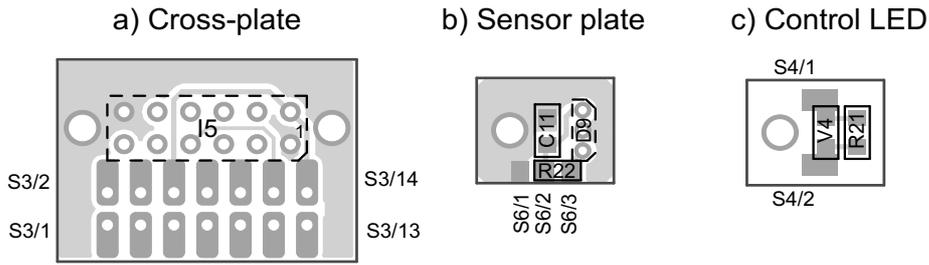


Figure 2.10: Placement of the components on the separate printed circuit boards PCB02C and PCB02D of the auxiliary electronics. Additional cross plate PCB02B is shown, too. Designations are the same as in circuit diagrams in Fig. 2.9.

2.4 Stepper motor module

The circuit diagram of stepper motor module for centering unit are shown in Fig. 2.11. PCB3 view are presented in Fig. 2.12a and Fig. 2.12b On the PCB3A main part of the module — motor controller, is placed. This PCB is installed inside of main case of the device. The PCB3B contains the optosensors limiting the movement of triangle knife. The connections between these PCB are made soldered. Table 2.5 shows a specification for the Stepper motor module.

2.5 Connections

Arrangement of cable connections between separate PCBs of the device is shown in Fig. 2.13. Connections between cross-plate PCB02B and PCB02C, PCB02D, and LEDs of FOV illumination (no PCB) are made by single-core wire. Length of each wire is defined in place.

High voltage cable (Fig. 2.13b) is a multi-cored cable with teflon insulation which shield serves as a common wire. It has an outer diameter 3 mm and length 100 mm.

The bus between Y1 and Y2 connectors (Fig. 2.13c) consists of 6 multi-core wires placed in a plastic tube. This cable has a length 100 mm. The conductors are labeled by color tube according to Table:

Next Table contains information on the line cable between the device and the host computer (RS485/LPT converter). Special RS-485 cable Belden 8132 is used.

Table 2.5: Specification for the Stepper motor module.

Item	Part	Name	Manufacturer	Q-ty	Rem
1	D1	IC LM2904M	Nat.Semicond.	1	
2	D2	IC PBL3777SI	Ericsson	1	
3	D3	IC AD8402AR10	Analog Dev.	1	
4	D4	IC ADM1485AR	Analog Dev.	1	
5	D5	IC ATmega8-16AI	Atmel	1	
6	Z1	Quartz 14.746 MHz	—	1	HC49S
7	R2, R4	SMD resistors	—	2	1206 size
8	R1-R10	SMD resistors	—	8	0805 size
9	C1, C4	SMD capacitors	—	2	1206 size
10	C2, C5	Alum.capacitor	—	2	16V, D=5 mm
11	C7	SMD tantal capac.	—	1	A size
12	C10	SMD tantal capac.	—	1	B size
13	C3-C12	SMD capacitors	—	6	0805 size
14	V1, V2	SDP8406	Honeywell	2	Phototrans.
15	V3, V4	SEP8506	Honeywell	2	IR LEDs
16	ISP	ISP connector	Custom	1	PCB holes
17	S7, S8	Power conn. PLS-2	Custom	2	Soldered
18	S9	Conn. PLS-4	Custom	1	PLS2-4
19	S10	Line conn. PLS2-4	Custom	1	Soldered
20	S11(A)	Pins conn. PLS2-5	—	1	PLS2-5
21	S11(B)	Pins conn. PLS2-5	—	1	Soldered
22	PCB03A	Printed board	Custom	1	
23	PCB03B	Printed board	Custom	1	

Table 2.6: Bus colors table

Conn. pin	Signal	Color
1	+5V	red
2	GND	black
3	Line A	white
4	Overlight	yellow
5	Line B	green
6	Synchro	black

2.6 RS485/LPT converter

The circuit diagram of the converter is shown in Fig. 2.15. PCB views are presented in Fig. 2.14.

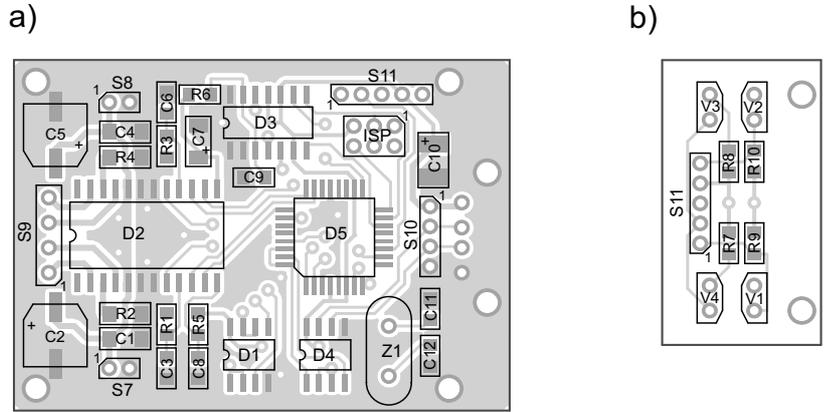


Figure 2.12: Placement of the components on the printed circuit board PCB3A (a) and on the printed circuit board PCB3B (b). Designations are the same as in circuit diagrams in Fig. 2.11.

Table 2.7: Arrangement of line cable

DB9F pin device side	Signal	Wire color	DB9M pin PC side	Rem.
1	Line A	blue	1	pair 1
2	Line B	blue-white	2	pair 1
5	GND	orange	5	pair 2
9	+5V	orange-white	9	pair 2
5	Shield		—	

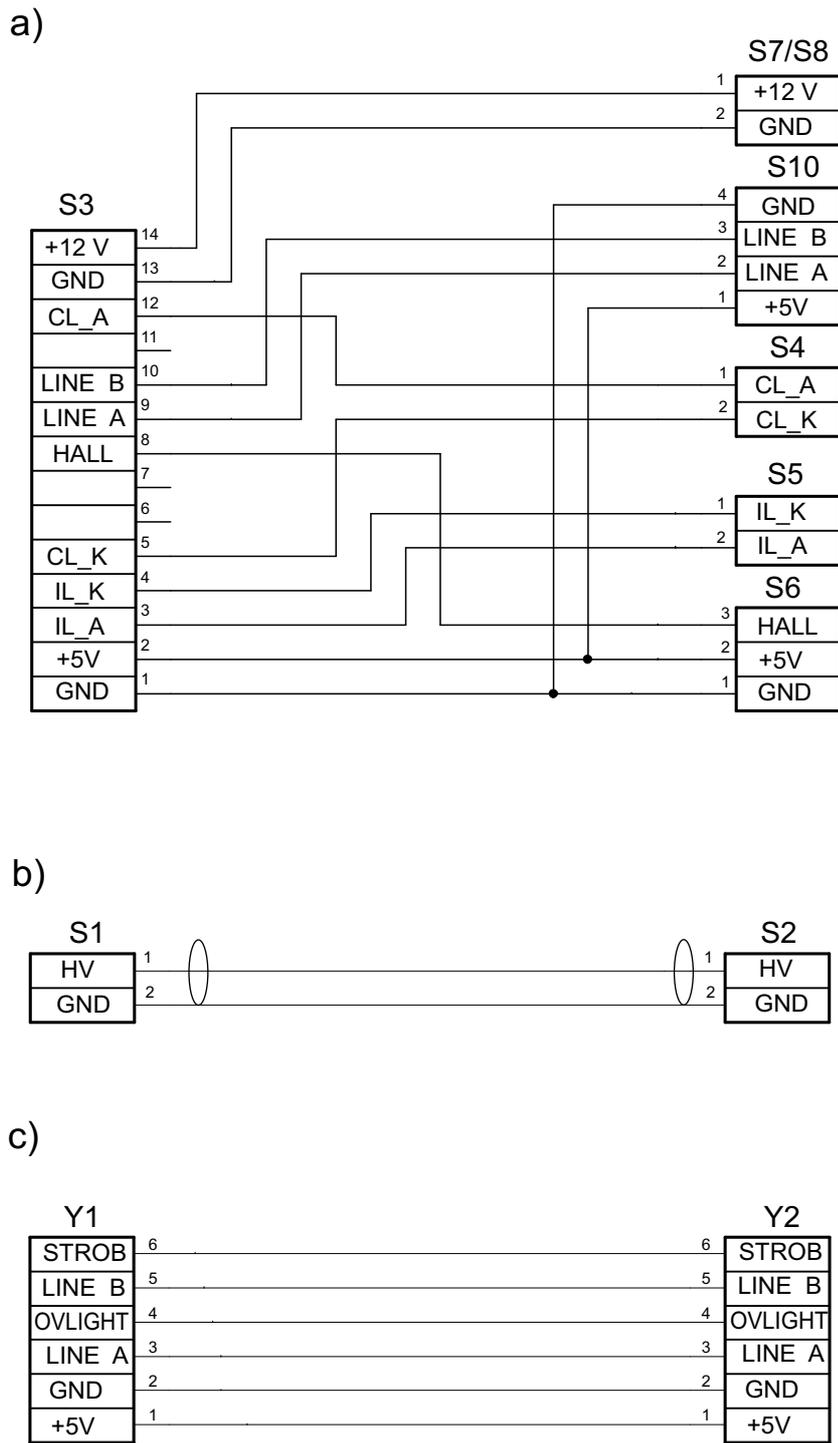


Figure 2.13: Connections between **a)** cross-plate and separate parts of the auxiliary electronics, **b)** main PCB and PMT dividers and **c)** main PCB and bicomputers PCB. Designations correspond to the circuit diagrams in Fig. 2.1 and Fig. 2.9.

Table 2.8: Specification for the RS485/LPT converter (see SCH02).

Item	Part	Name	Manufacturer	Q-ty	Rem
1	D1	IC ADM1485AR	Analog Dev.	1	
2	D2	IC ATmega8-16AI	Atmel	1	
3	D3	IC HCPL0611	HP	1	
4	D4	IC HCPL0630	HP	1	
5	R1–R9	Chip resistors	—	1	0805 size
6	C2	Tantal chip capac.	—	1	A size
7	C4	Alum.chip capac.	—	3	B size
8	C2, C3	Chip capacitors	—	2	1208 size
9	C5	Chip capacitors	—	2	1208 size
10	X1	DB25M connector	—	1	LPT
11	X2	DB9M	—	1	Line
12	PCB	Printed board	Custom	1	

LPT/RS485

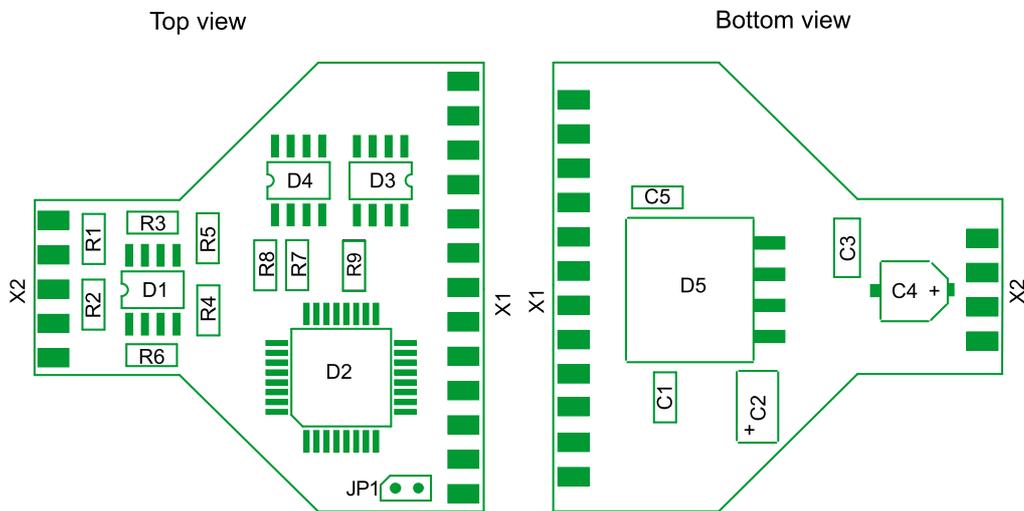


Figure 2.14: Printed circuit boards for the RS485/LPT. Placement of the components is shown. Designations correspond to circuit diagrams in Fig. 2.15

Chapter 3

Microprogramming and Instructions set

3.1 Logical level of interaction between computer and modules

As described in Section 1.6, the command and/or data exchange between the computer and the modules is implemented by byte packet transmissions. Each full cycle of exchange is started with a header byte and terminated with acknowledgment signal. Header byte contains 5-bits address of the destination or departure module and 2-bits cyclic packet number to distinguish a transmission repeat from a new packet. Both header and signal bytes are marked by the parity bit. For distinction between them, the header byte is a positive number and the signal is a negative.

Normally two signals are only used: ACK — acknowledgment of successful data receiving and NAK — non acknowledgment, which is equivalent to unsuccessful data receiving message. The used logical protocol includes several additional signals to provide a larger information flow rate. The signals contain a self-checked code with 8 possible values. These signals are listed in Table 3.1. Last two signals are not used in MASS exchange protocol and included for a further extension.

Table 3.1: Used signal bytes.

Signal	Hex. code	Meaning
ACK	0x87	Successful receiving of data
NAK	0x96	Damaged packet was received
NOD	0xA5	Required data are not ready
ACN	0xB4	Command is successfully received but such a command does not exist
ACY	0xC3	Command is successfully received and executed
ACW	0xD2	Command is successfully received but can not be executed right now
SINC	0xE1	Synchronization signal
DNG	0xF0	Danger signal

Within the packet, the command byte follows after the header byte. This byte contains a command to the module, or both a command and operand, or length of the following data block, depending on its code. The encoding procedure is simple and fast. Most of commands are read/write operations of working variables with or without execution of associated functions. Rest part only run some functions. Naturally, the same function for different modules has the same command. For example, request of the module identification has the command code 0xA2 for all modules.

If the command byte is less than 0x20 then following bytes represent a block of data with the length equal to this command byte value. Contrary to the command transmissions (from computer to module only, see an exception below), the data transmission can go in both directions. The maximal length of the data block is 31 bytes. This permits to provide a good buffered data transmission leaving the line free during long time.

Since a cyclic residual control byte (CRC byte) is appended during transmission and cut off during receiving by a system driver, in the further description of the commands (instructions), CRC byte is omitted. Also, the first byte of a transmitted packet (header byte) is not shown as a rule. Additionally, fault transmission cases are not considered, correct situations are presented only.

The instruction sets for the all the modules in use are presented below. For each instruction, a name (and its code in parentheses), arguments used (if exist) are written. Normal reply from a module is shown after a left arrow. The response alternatives are put in brackets.

3.2 Bicounter module

Main problem for photometric modules is the synchronization of integration and subsequent transmissions. To solve this, one module (Master) generates a synchronization clock. Another module (Slave) uses this signal to organize its work at the same rate.

To avoid line collisions, the data transmission procedure works in inductive mode. One module is set in Active mode of data transmission and starts the transmission as soon as the block of data is ready. Another Bicounter is set in Inductive mode and starts the transmission of integrated data after the packet from its module-Inductor have passed the line.

Current status of the Bicounter module is indicated by its Status word accessible for reading. It has the following format:

```
Bit 0 - active mode on
Bit 1 - inductive mode on
Bit 2 - short format of data transmission on
Bit 3 - integration using external clock
Bit 4 - decremental test on
Bit 5 - data block is ready for transmission
Bit 7 - integration in progress.
```

The instructions set is presented below.

- Pulse discrimination threshold setting for counter A:
SET_LEVEL_A (0x41) level <- ACY,
where level is obtained from threshold T in mV with help of 2 constants programmed in

the module:

```
level = low(max(0,min(255,255+const2-T*(255+const2-const1))))
```

- Pulse discrimination threshold setting for counter B:
SET_LEVEL_B (0x42) level <- ACY
- Current threshold request for counter A:
GET_LEVEL_A (0xE1) <- level,
then $T = (255+const2-level)/(255+const2-const1)$.
- Current threshold request for counter B:
GET_LEVEL_B (0xE2) <- level
- Micro-exposure (integration time) setting:
SET_EXPOS (0x54) low(exposure) high(exposure) <- ACY
where exposure is calculated from microexposure t in ms with help of 2 constants programmed in the module: $exposure = (t*(const3+const4<<8)-1)/8$
- Current micro-exposure request:
GET_EXPOS (0xF4) <- low(exposure) high(exposure)
then t in ms is equal $(8*exposure+1)/(const3+const4<<8)$
- Series length setting:
SET_NUMBER (0x36) low(number) high(number) <- ACY
where number from 1 to 32767, in the case of number = 0 the infinite series is set.
- Current series length request:
GET_NUMBER (0xF6) <- low(number) high(number).
- Module status request (see the meaning of the Status bits above):
GET_STATUS (0xE0) <- status.
- EEPROM CRC check:
GET_CRC (0xEF) <- crc.
There are no errors if $crc = 0$.
- Data block size setting:
SET_BLSIZE (0x28) size <- ACY,
where size can be from 1 to 16, recommended value is 16.
- Data block size request:
GET_BLSIZE (0xE8) <- size.
- Inductor address setting:
SET_INDUC (0x29) address <- ACY.
- Inductor address request:
GET_INDUC (0xE9) <- address.
- Request of a new data block:
GET_DATA (0xA0) <- data block (NOD).
Returns NOD if no new data ready.
- Module identification request:
GET_IDENT (0xA2) <- id1 id2 id3 id4.
Returns unique identification of the module.

- Module constants request :
`GET_CONST (0xA3) <- const1 const2 const3 const4.`
Returns four constants for threshold and exposure calculations.
- Start up exposure series:
`RUN (0x80) <- ACY (ACW).`
Returns `ACW` when integration does not finish yet.
- Halt of exposure series:
`STOP (0x81) <- ACY.`
Used to terminate infinite series or to break current integration.
- Generation of synchro clock on:
`MASTER_ON (0x83) <- ACY (ACW).`
- Use an external synchro clock on:
`MASTER_OFF (0x82) <- ACY (ACW).`
- Active mode on:
`ACTIVE_ON (0x88) <- ACY (ACW).`
- Active mode off:
`ACTIVE_OFF (0x89) <- ACY (ACW).`
- Inductive mode on:
`INDUCE_ON (0x8A) <- ACY (ACW).`
- Inductive mode off:
`INDUCE_OFF (0x8B) <- ACY (ACW).`
- One byte per count format on:
`SHORTER (0x84) <- ACY (ACW).`
- Two byte per count format on:
`LONGER (0x85) <- ACY (ACW).`
- Start up decremental test:
`RUN_TEST (0x86) <- ACY (ACW).`
The numbers from `number-1` to 0 are generated instead of normal counts to check exchange faultiness.

Previous instructions return `ACW` signal if the integration is in progress.
- Software restart:
`RESET (0x87) <- ACY`

3.3 Auxiliary module

A brightness of the aperture illumination is changed step by step and is described by a following equation:

$$\text{illumination} = \text{floor}(2^{(8*n/15.01)+0.5}) \text{ if } n \neq 0,$$

$$\text{illumination} = 0 \text{ if } n = 0.$$

The procedure of a repetitive turning on of the high voltage after the overlight detection is following: HV turn off, Safety turn off, Safety turn on and then HV turn on. This procedure is made in such a complex way to protect from occasional turning on of the high voltage.

Current status of this module is indicated by its Status word accessible for reading. It has the following format:

Bit 0 - high voltage turned on
Bit 1 - safety on (overlight protection on)
Bit 2 - overlight indicator
Bit 3 - high voltage locking
Bit 4 - control light on
Bit 5 - illumination on
Bit 6 - modulation of the control light on
Bit 7 - viewer mirror of axis

The instructions set is presented below.

- Illumination brightness setting:
SET_ILLUM (0x41) illumination <- ACY
illumination is calculated from relative brightness IL (from 0 to 1.0):
illumination = low(max(0,min(255,255*IL)))
- Illumination brightness request:
GET_ILLUM (0xE1) <- illumination
where IL = illumination/256
- Control light brightness setting:
SET_LIGHT (0x42) light <- ACY
light is calculated from relative brightness CL (from 0 to 1.0):
light = low(max(0,min(255,255*IL))).
The brightness setting instructions do not turn on light.
- Control light brightness request:
GET_LIGHT (0xE2) <- light
where CL = light/256
- Module status request (see the meaning of the Status bits above).
GET_STATUS (0xE0) <- status
- EEPROM CRC check
GET_CRC (0xEF) <- crc.
There are no errors if crc = 0.
- Control light modulation amplitude setting:
SET_VAMPL (0x23) delta <- ACY
where delta = low(max(0,min(255,255*DL*CL))) and DL is a relative amplitude.
- Control light modulation amplitude request:
GET_VAMPL (0xE3) <- delta,
then DL = (delta/256)/CL.
- Illumination turn on:
ILLUM_ON (0x80) <- ACY (ACW)
- Illumination turn off:
ILLUM_OFF (0x81) <- ACY (ACW).

- Control light turn on:
LIGHT_ON (0x82) <- ACY (ACW).
- Control light turn off:
LIGHT_OFF (0x83) <- ACY (ACW).
- Control light modulation turn on:
VARY_ON (0x84) <- ACY (ACW).
- Control light modulation turn off:
VARY_OFF (0x85) <- ACY (ACW).
- High voltage value setting:
SET_VOLTAGE (0x44) high <- ACY,
where high is calculated from needed voltage in V with help of 2 programmed constants:
high = low(max(0, min(255, 0.001*U*const1 -const2))).
The instruction does not turn on the high voltage.
- High voltage value request:
GET_VOLTAGE (0xE4) <- high.
Then $U = 1000*(high+const2)/const1$.
- High voltage turn on:
HIGH_ON (0x88) <- ACY (ACW).
Returns ACW in a case of locking, and does not turn anything on the high voltage.
- High voltage turn off:
HIGH_OFF (0x89) <- ACY.
- Safety turn on:
SAFETY_ON (0x8A) <- ACY (ACW).
Executes only if the HV is turned off.
- Safety turn off:
SAFETY_OFF (0x8B) <- ACY (ACW).
Executes only if the HV is turned off.
- Device temperature request:
GET_TEMPER (0xE5) <- temperature.
Temperature in °C is equal to $-20+(temperature)/4$.
- Module identification request:
GET_IDENT (0xA2) <- id1 id2 id3 id4.
Returns unique identification of the module. Also, this instruction switches module from autonomous work to under computer control.
- Module constants request:
GET_CONST (0xA3) <- const1 const2 const3 const4.
Returns four constants.
- Software restart:
RESET (0x87) <- ACY.

3.4 Stepper motor controller module

Current status of this module is indicated by its Status word accessible for reading. It has the following format:

```
Bit 0 - new microstep is done
Bit 1 - left stop is achieved
Bit 2 - right stop is achieved
Bit 5 - disable motor powering
Bit 6 - forward motion
Bit 7 - motor is moving
```

The instructions set is presented below.

- Module status request (see the meaning of the Status bits above):
`GET_STATUS (0xE0) <- status`
- EEPROM CRC check
`GET_CRC (0xEF) <- crc.`
There are no errors if `crc = 0`.
- Current position request:
`GET_POSITION (0xF2) <- low(position) high(position).`
`position` changes during motion.
- Relative shift start up:
`SHIFT_AT (0x54) low(shift) high(shift) <- ACY (ACW),`
where `shift` is a signed integer value.
- Motion speed setting:
`SET_SPEED (0x56) low(speed) high(speed) <- ACY (ACW),`
where `speed` is positive integer, and can be calculated from step period $T(\text{ms})$
 $\text{speed} = (T * (\text{const3} + \text{const4} \ll 8) - 1) / 8$
- Motion speed request:
`GET_SPEED (0xF6) <- low(speed) high(speed),`
then T in ms is equal $(8 * \text{speed} + 1) / (\text{const3} + \text{const4} \ll 8)$
- Turn on stepper motor powering:
`TURN_ON (0x80) <- ACY (ACW).`
ACW when motor is moved, ACY otherwise.
- Turn off stepper motor powering:
`TURN_OFF (0x81) <- ACY (ACW).`
ACW when motor is moved, ACY otherwise.
- Motor status check:
`TEST_MOTION (0x82) <- ACY (ACW).`
ACW when motor is moved, ACY otherwise.
- Start up motion to left stop:
`AT_LEFT (0x83) <- ACY (ACW).`
ACW when motor is moved already, ACY otherwise.

- Start up motion to right stop:
`AT_RIGHT (0x84) <- ACY (ACW).`
ACW when motor is moved already, ACY otherwise.
- Emergency stop:
`STOP (0x85) <- ACY (ACW).`
ACW when motor isn't moved, ACY otherwise.
- Sensor LED turn on:
`LED_ON (0x88) <- ACY (ACW).`
- Sensor LED turn off:
`LED_OFF (0x89) <- ACY (ACW).`
- Clear absolute position:
`CLEAR_ABS (0x8A) <- ACY (ACW).`
Here and above ACW returned when motor is moved.
- Module identification request:
`GET_IDENT (0xA2) <- id1 id2 id3 id4.`
Returns unique identification of the module.
- Module constants request:
`GET_CONST (0xA3) <- const1 const2 const3 const4.`
Returns four constants.
- Software restart:
`RESET (0x87) <- ACY.`

3.5 Updating a microcode

The ATmega8 microcontroller has a useful feature — Boot Loader Support (BLS), which permits to change part of microcode for application with help of another part of the microcode - so called boot loader. In practice, the microcontroller is programmed for the first time only by the boot loader software with help of any ISP programmer. This loader program starts each time after power on. This code supports a number of special commands which permit to load the new version of the needed application software in the flash memory.

In order to switch from the microcode update state to the working state, the module must receive the RESET command. In the normal work when Turbina software is started, it sends RESET as first command for each module. After that, module goes into application mode and operates further in this mode. There is a possibility to return into boot loader mode with a special command: `BOOT (0xC0) <- ACY.`

Boot loader itself can't be reprogrammed this way, what guarantees the possibility to repeat the application software update even if a serious problem has occurred during the updating process. Since the updating procedure uses the same RS-485 line for data transfer, no additional cabling or device disassembling are required.

MASS software distribution contains an utility `avrup` in the directory `avrup` of the RS-485 driver package. Before its usage the needed steps for RS-485 driver installation must be done and MASS device must be connected and powered on. To reprogram any module, its application code `Naprimer.hex` in *Generic* format must be prepared. For example, conversion

of the assembler application text into the generic (hex) format may be done with help of Atmel AVR assembler `avrasm32.exe` started via `wine` software under Linux OS directly.

The `avrup` utility has 3 options:

- `-b bootaddress` — indicates that the module will be updated with the boot loader address = `bootaddress` (hex).
- `-a appladdress` — indicates that module will have address = `appladdress` (hex) after updating.
- `-r baudrate` — indicates that module will have new exchange rate = `baudrate` (Kbaud) after updating.

We recommend to always save `appladdress` the same as `bootaddress`, both corresponding to the module address in `device.cfg` file. For example, to update `Bicounter1` with new code, execute:

```
.\avrup -a1 -b1 bicounter.hex
```

Note, that the same microcode `bicounter.hex` is used for both `Bicounters`.

Being started, the utility detects the state of the module (application or update) accessing it via the boot address. Note that problem may arise if the boot and application addresses of this module differ. If needed, the module is switched into update state. In case the application address or baudrate are given in command options, the respective code words are modified before uploading. Then `avrup` loads the application code page by page from the zero address and checks it by back reading and comparison. This progress of this uploading and checking is shown in console. After end of uploading the module is reset into application state.

Note, that for control needs, the utility modifies the 3rd and 4th bytes of the module identification replacing them with the encoded current programming date. This new identification is displayed in the end of reprogramming and should be written in respective section of `device.cfg`. If forgotten, the next start of `Turbina` will output the warning message about the invalid identification of the reprogrammed module.