Monostable Relay Firmware

UNIV 1.0.2.11

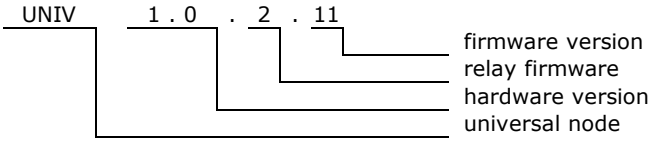
1. Features

- Controller of 6 monostable relays
- 3 control instructions (on, off, toggle)
- 3 blocking instructions
- 6 timers (1 for each relay) for instruction execution delay 1s-20h
- Allows to set up messages that can toggle the module
- Settable Power up value.
- Allows writing notes in processor memory
- Uptime counting
- Self health check
- Receive and transmit FIFO buffers for CAN

2. Compatibility:

- Firmware for **UNIV 1.0.2.1 application**.
- Firmware can be uploaded into devices with bootloader version 2.5 or compatible.
- This firmware is not compatible with previous one UNIV 1.0.2.1. The node will need reconfiguring.

3. Firmware version



4. Operation overview

This is a controller of 6 monostable relays. The module can react to 96 messages received from the bus. These rules are saved in the non-volatile memory. When module receives a message it can change status of relays immediately or with delay. The new status of relays will be sent to the bus. In this firmware blocking instructions have been implemented. They can block executing of other instructions save in module memory. They can be used for simple logical functions.

5. Firmware

Firmware can be uploaded into the node by using HAPCAN Programmer, which can be downloaded from site http://siwilo.com/hapcan/software.

5.1. Relay message

The module sends message to the bus, when the status of relays changes.

Table 1. RELAY frame (0x302)

Frame type	Flags	Module	Group	D0	D1	D2	D3	D4	D5	D6	D7
0x302	3 2 1 0	Node Nr	Group Nr	0xFF	0xFF	CHANNEL	STATUS	0xFF	INSTR1	INSTR2	TIMER

0x302 – universal module frame, relay application

3	- not used flag, read as "0"
2	- not used flag, read as "0"
1	- not used flag, read as "0"
0	RE - response flag, flag is equal "1" if node was requested. If flag is equal „0" it means that status of output has just changed.

Node Nr - node number on the network

Group Nr - group number of the node on the network

CHANNEL - output channel

STATUS - actual status of outputs 0x00 – relay off, 0xFF – relay on

INSTR1 - instruction that is waiting for execution, or 0xFF if none instruction

INSTR2 - second byte of instruction that is waiting for execution, or 0xFF

TIMER - delay value of waiting instruction, or 0x00 if none waiting

5.2. Status request

Status of module can be checked by sending from computer STATUS REQUEST frame (0x109) (Table 2).

Table 2. STATUS REQUEST frame (0x109).

Frame type	Flags	Module	Group	D0	D1	D2	D3	D4	D5	D6	D7
0x109	0x0	COMP ID1	COMP ID2	0xXX	0xXX	Node Nr	Group Nr	0xXX	0xXX	0xXX	0xXX

0x1090 – STATUS REQUEST frame

COMP ID1 - computer identifier (must be unique on the network)

COMP ID2 - computer identifier (must be unique on the network)

Node Nr - node number of requested module

Group Nr - group number of requested module

0xXX - inessential data

As response the module will send RELAY frames. The meaning of bytes is the same as in Table1.

Table 3. Response to STATUS REQUEST

Frame type	Flags	Module	Group	D0	D1	D2	D3	D4	D5	D6	D7
0x302	0x1	Node Nr	Group Nr	0xFF	0xFF	0x01	STATUS	0xFF	INSTR1	INSTR2	TIMER
0x302	0x1	Node Nr	Group Nr	0xFF	0xFF	0x02	STATUS	0xFF	INSTR1	INSTR2	TIMER
0x302	0x1	Node Nr	Group Nr	0xFF	0xFF	0x03	STATUS	0xFF	INSTR1	INSTR2	TIMER
0x302	0x1	Node Nr	Group Nr	0xFF	0xFF	0x04	STATUS	0xFF	INSTR1	INSTR2	TIMER
0x302	0x1	Node Nr	Group Nr	0xFF	0xFF	0x05	STATUS	0xFF	INSTR1	INSTR2	TIMER
0x302	0x1	Node Nr	Group Nr	0xFF	0xFF	0x06	STATUS	0xFF	INSTR1	INSTR2	TIMER

5.3. Uptime request

Table 4. UPTIME REQUEST (0x113).

Frame type	Flags	Module	Group	D0	D1	D2	D3	D4	D5	D6	D7
0x113	0x0	COMP ID1	COMP ID2	0xXX	0xXX	Node Nr	Group Nr	0xXX	0xXX	0xXX	0xXX

0x1130 – UPTIME REQUEST frame

COMP ID1 - computer identifier (must be unique on the network)

COMP ID2 - computer identifier (must be unique on the network)

Node Nr - node number of requested module

Group Nr - group number of requested module

0xXX - inessential data

Table 5. Response to UPTIME REQUEST (0x113).

Frame type	Flags	Module	Group	D0	D1	D2	D3	D4	D5	D6	D7
0x113	0x1	Node Nr	Group Nr	0xFF	0xFF	0xFF	0xFF	UPTIME3	UPTIME2	UPTIME1	UPTIME0

0x1131 – Response to UPTIME REQUEST frame

Node Nr - node number on the network

Group Nr - group number of the node on the network

UPTIME - (UPTIME3*256^3+UPTIME2*256^2+UPTIME1*256^1+UPTIME3*256^0) in seconds

5.4. Health check request

Table 6. HEALTH CHECK - STATUS REQUEST (0x115).

Frame type	Flags	Module	Group	D0	D1	D2	D3	D4	D5	D6	D7
0x115	0x0	COMP ID1	COMP ID2	0x01	0xXX	Node Nr	Group Nr	0xXX	0xXX	0xXX	0xXX

0x1150 – HEALTH CHECK REQUEST frame – STATUS REQUEST

COMP ID1 - computer identifier (must be unique on the network)

COMP ID2 - computer identifier (must be unique on the network)

0x01 - status request

Node Nr - node number of requested module

Group Nr - group number of requested module

0xXX - inessential data

As response the module will send two frames (Table 7).

Table 7. Response to HEALTH CHECK - STATUS REQUEST (0x115).

Frame type	Flags	Module	Group	D0	D1	D2	D3	D4	D5	D6	D7
0x115	0x1	Node Nr	Group Nr	0x01	RXCNT	TXCNT	RXCNTMX	TXCNTMX	CANINTCNT	RXERRCNT	TXERRCNT

0x1151 – Response HEALTH CHECK REQUEST frame

Node Nr - node number on the network

Group Nr - group number of the node on the network

0x01 - frame 1 (current values)

RXCNT - current level of receive FIFO buffer

TXCNT - current level of transmit FIFO buffer

RXCNTMX - maximum level of receive FIFO buffer since power up

TXCNTMX - maximum level of transmit FIFO buffer since power up

CANINTCNT - number of CAN interface restarts since power up

RXERRCNT - current receive errors register

TXERRCNT - current transmit errors register

Frame type	Flags	Module	Group	D0	D1	D2	D3	D4	D5	D6	D7
0x115	0x1	Node Nr	Group Nr	0x02	0xFF	0xFF	RXCNTMXE	TXCNTMXE	CANINTCNTE	RXERRCNTE	TXERRCNTE

0x1151 – Response HEALTH CHECK REQUEST frame

Node Nr - node number on the network

Group Nr - group number of the node on the network

0x02 - frame 2 (maximum values saved in eeprom memory)

RXCNTMXE - maximum ever level of receive FIFO buffer

TXCNTMXE - maximum ever level of transmit FIFO buffer

CANINTCNTE - maximum ever number of CAN interface restarts

RXERRCNTE - maximum ever receive errors

TXERRCNTE - maximum ever transmit errors

To clear maximum values saved in eeprom memory the frame shown in Table 8 must be sent. There is no response to this message.

Table 8. HEALTH CHECK - CLEAR REQUEST (0x115).

Frame type	Flags	Module	Group	D0	D1	D2	D3	D4	D5	D6	D7
0x115	0x0	COMP ID1	COMP ID2	0x02	0xXX	Node Nr	Group Nr	0xXX	0xXX	0xXX	0xXX

0x1150 – HEALTH CHECK REQUEST frame – CLEAR REQUEST

COMP ID1 - computer identifier (must be unique on the network)

COMP ID2 - computer identifier (must be unique on the network)

0x02 - clear request

Node Nr - node number of requested module

Group Nr - group number of requested module

0xXX - inessential data

5.5. Module control

The module can be controlled directly from PC, or indirectly by other modules. In both situation all 3 instruction can be used.

5.5.1. Control instruction

Table 9 shows all instructions, which can be executed by relay controller. The byte INSTR1 defines instruction, byte INSTR2 defines relays.

Table 9. Coding of relay instructions

Instruction	Instruction code		Description
	INSTR1	INSTR2	
TURN OFF	0x00	X	It will turn off chosen relays, and the rest will stay unchanged.
TURN ON	0x01	X	It will turn on chosen relays, and the rest will stay unchanged.
TOGGLE	0x02	X	It will toggle chosen relays, and the rest will stay unchanged.

X – chosen relays (see table below)

INSTR2	Description
<00000001>	- 0x01 – only relay K1
<00000010>	- 0x02 - only relay K2
<00000011>	- 0x03 - relays K1 & K2
<00000100>	- 0x04 - only relay K3
…	…
<00111111>	- 0x3F - relays K1,K2,K3,K4,K5,K6

bit <0> - relay 1
bit <1> - relay 2
bit <2> - relay 3
bit <3> - relay 4
bit <4> - relay 5
bit <5> - relay 6

5.5.2. Timer

All instructions can be executed with delay 1s-20h set by TIMER register. Drawing below shows time dependence of TIMER register.

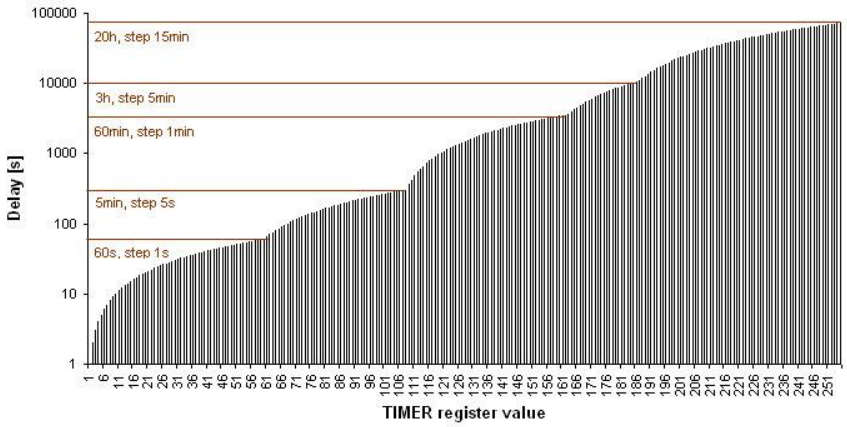


Figure 1. Delay/timer register relationship

5.5.3. Direct control

It is possible to control module by sending DIRECT CONTROL message. The message contains instruction, which will be executed by module. The module can be also controlled from HAPCAN Programmer.

Table 10. DIRECT CONTROL frame (0x10A).

Frame type	Flags	Module	Group	D0	D1	D2	D3	D4	D5	D6	D7
0x10A	0x0	COMP ID1	COMP ID2	0xXX	0xXX	Node Nr	Group Nr	0xXX	INSTR1	INSTR2	TIMER

0x10A0 – DIRECT CONTROL frame

COMP ID1 - computer identifier (must be unique on the network)

COMP ID2 - computer identifier (must be unique on the network)

Node Nr - node number of requested module

Group Nr - group number of requested module

INSTR1 - instruction to be executed (byte1)

INSTR2 - instruction to be executed (byte2)

TIMER - instruction delay

0xXX - any value

5.5.4. Indirect control

Indirect control means that module will react to messages sent by other modules on the network. It depends on configuration programmed into the module.

5.6. Configuration

With this version of firmware parameters below can be configured:

- Module identifier (module number and group number);
- Module description (16 chars);
- Power up values;
- Text notes;
- Linking device with other modules (indirect control of module).

Configuration process can be done by using HAPCAN Programmer.

5.6.1. Module identifier

Every module on the network must have unique identifier. The identifier is made of two bytes, module number (1 byte) and group number (1 byte). Belonging to particular group might be important when linking devices; e.g. some devices can react to messages sent by modules belonging to particular group.

5.6.2. Module description

Every module can have 16 char description, which makes easier for user (programmer) to distinguish nodes.

5.6.3. Power up values

It is possible to configure relay states at startup after power loss. At startup relays can be set to ON, OFF or to the last set value. The last set value must be unchanged for at least 6s before power failure.

5.6.4. Text notes.

Up to 1024 characters can be written into processor's memory.

5.6.5. Linking devices

The module has 96 memory cells (boxes). Each box can contain information about message sent by other node, and instruction which will be executed when that message is received.

This firmware has feature to set simple conditions of executing instruction. To do so, you can use blocking instruction shown in the table below. As an example of simple condition can be situation when light has to be turned on by PIR when someone enters room, but should not be during a day. The HAPCAN Programmer simplifies configuration process.

Table 11. Coding of conditional instructions

Instruction	Instruction code			Description
	INSTR1	INSTR2	INSTR3	
ENABLE BOX	0xDD	X	Y	It enables chosen boxes – these boxes will be compared with next received message from the bus.
DISABLE BOX	0xDE	X	Y	It disables chosen boxes – these boxes will be passed when next message arrives from the bus.
TOGGLE BOX	0xDF	X	Y	It toggles boxes – enables when they are disabled and vice versa

INSTR2	Description
0x00	Box 1
0x01	Box 2
…	…
0x5F	Box 96

INSTR3	Description
0x00	+ 0 -(and not anyone)
0x01	+ 1 -(and 1 following)
…	…
0x5F	+ 95 -(and 95 following)

6. Document version

File	Note	Date
univ_v1-0-2-11a.pdf	Original version	September 2011