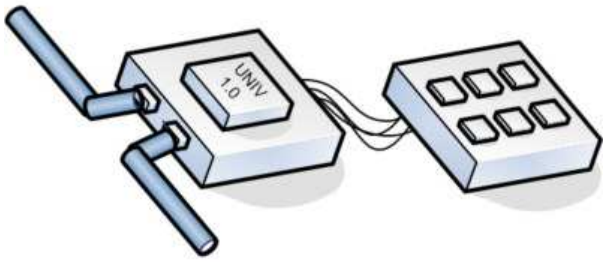
Button Firmware
UNIV 1.0.1.2

1. Features:

- 6 channel button module. Up to 6 buttons with free voltage contacts can be connected to the module
- 3 control instructions
- 3 blocking instructions
- There is a 20ms reaction time. Button has to be pressed for at least 20ms to send a message. It avoids contacts bouncing.
- Allows to set up 128 conditions for receiving bus messages



2. Compatibility:

- Firmware for **UNIV 1.0.1.2. application**.
- Firmware can be uploaded into devices with bootloader version 2.5 or compatible.

3. Firmware version

UNIV 1 . 0 . 1 . 2

- 2 – firmware version
- 1 – button firmware
- 0 – hardware version
- UNIV – universal node

4. Operation overview

Node sends message to the bus saying which button was pressed. It sends another message, when button is released. Firmware controls LEDs connected to switches. LED can be toggled by message received from the bus or locally by pressed button.

5. Firmware

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

5.1. Button message

It sends message to the bus, when the status of input changed and was held for at least 20ms. The table below shows meaning of each byte in the button frame.

Table 1. BUTTON MESSAGE frame – input state.

Frame type	Flags				Module	Group	D0	D1	D2	D3	D4	D5	D6	D7
0x301	3	2	1	0	Node Nr	Group Nr	0xFF	0xFF	CHANNEL	STATUS	0xFF	0xFF	0xFF	0xFF

- 0x301 – universal module frame, button 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 input has just changed.
- Node Nr - node number on the network
- Group Nr - group number of the node on the network
- CHANNEL - input channel
- STATUS - actual status of input 0x00 – open, 0xFF - close

Table 2. BUTTON MESSAGE frame – LED state

Frame type	Flags	Module	Group	D0	D1	D2	D3	D4	D5	D6	D7
0x301	3 2 1 0	Node Nr	Group Nr	0xFF	0xFF	CHANNEL	STATUS	0xFF	0xFF	0xFF	0xFF

0x301 – universal module frame, button 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. For LED always equals "1"

Node Nr - node number on the network

Group Nr - group number of the node on the network

CHANNEL - diode number 0x21 (diode1) – 0x26 (diode6)

STATUS - LED status 0x00 – off 0xFF - on

The LED states are only sent as response for STATUS REQUEST.

5.2. Status request

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

Table 3. 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

0x109 – 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 status frames (table 4). Meaning of bytes is the same as in table 1.

Table 4. Response to STATUS REQUEST.

Frame type	Flags	Module	Group	D0	D1	D2	D3	D4	D5	D6	D7
0x301	0x1	Node Nr	Group Nr	0xFF	0xFF	0x01	STAN	0xFF	0xFF	0xFF	0xFF
0x301	0x1	Node Nr	Group Nr	0xFF	0xFF	0x02	STAN	0xFF	0xFF	0xFF	0xFF
0x301	0x1	Node Nr	Group Nr	0xFF	0xFF	0x03	STAN	0xFF	0xFF	0xFF	0xFF
0x301	0x1	Node Nr	Group Nr	0xFF	0xFF	0x04	STAN	0xFF	0xFF	0xFF	0xFF
0x301	0x1	Node Nr	Group Nr	0xFF	0xFF	0x05	STAN	0xFF	0xFF	0xFF	0xFF
0x301	0x1	Node Nr	Group Nr	0xFF	0xFF	0x06	STAN	0xFF	0xFF	0xFF	0xFF
0x301	0x1	Node Nr	Group Nr	0xFF	0xFF	0x21	STAN	0xFF	0xFF	0xFF	0xFF
0x301	0x1	Node Nr	Group Nr	0xFF	0xFF	0x22	STAN	0xFF	0xFF	0xFF	0xFF
0x301	0x1	Node Nr	Group Nr	0xFF	0xFF	0x23	STAN	0xFF	0xFF	0xFF	0xFF
0x301	0x1	Node Nr	Group Nr	0xFF	0xFF	0x24	STAN	0xFF	0xFF	0xFF	0xFF
0x301	0x1	Node Nr	Group Nr	0xFF	0xFF	0x25	STAN	0xFF	0xFF	0xFF	0xFF
0x301	0x1	Node Nr	Group Nr	0xFF	0xFF	0x26	STAN	0xFF	0xFF	0xFF	0xFF

5.3. Module control

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

5.3.1. Control instruction

The module understands 3 instructions shown below in the Table 5.

Table 5. Coding of instructions

Instruction	Instruction code			Description
	INSTR1	INSTR2	INSTR3	
Turn diode off	0x00	DIODEs	0xXX	Turns chosen diodes off
Turn diode on	0x01	DIODEs	0xXX	Turns chosen diodes on
Toggle diode	0x02	DIODEs	0xXX	Toggles chosen diodes

0xXX - any value

INSTR2	Description
<00000001>	- diode 1 only
<00000010>	- diode 2 only
<00000011>	- diodes 1 & 2
<00000100>	- diode 3 only
…	…
<00111111>	- diodes 1,2,3,4,5,6

5.3.3. 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 6. 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	INSTR3

0x10A – 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)

INSTR3 - not used

0xXX - any value

5.3.4. 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.

5.4. Configuration

With this version of application parameters below can be configured:

- Module identifier (module number and group number);
- Module description (16 chars);
- Linking device with other modules (indirect control of module)

Configuration process can be done by using HAPCAN Programmer.

5.4.1. 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.

5.4.2. Module description

Every module can have 16 char description, which makes easier for user (programmer) to distinguish nodes. Examples of node descriptions: living-button, living-lamp, bed2-button etc.

5.4.6. Linking devices

The module has 128 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 7. 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
…	…
0x7F	Box 128

INSTR3	Description
0x00	+ 0 -(and not anyone)
0x01	+ 1 -(and 1 following)
…	…
0x7F	+ 127 -(and 127 following)

6. Document version

File	Note	Date
univ_v1-0-1-2a.pdf	Original version	May 2010