Button Firmware
UNIV 1.0.1.1

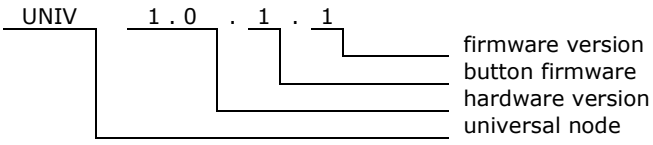
1. Features:

- 6 channel button module. Up to 6 buttons with free voltage contacts can be connected to the module
- 7 types of button behaving is recognized: button pressed, released, pressed for 400ms, pressed for 4s, released within 400ms, released between 400ms and 4s, released after 4s
- There is a 20ms reaction time. Button has to be pressed for at least 20ms to activate module. It avoids contacts bouncing
- LEDs can't be connected to the module
- Allows writing notes in the processor memory
- Uptime counting
- Self-health check
- Receive and transmit FIFO buffers for CAN

2. Compatibility:

- Firmware for **UNIV 1.0.1.0. application.**
- Firmware is compatible with previous versions UNIV 1.0.1.0.
- Firmware can be uploaded into devices with bootloader version 2.5 or compatible.

3. Firmware version



4. Overview

This is a button module firmware. The module can work with up to 6 push buttons or other switches with voltage free contacts. Module is able to distinguish a few types of button behaving: pressed, released, pressed for 400ms, pressed for 4s, released within 400ms, released between 400ms and 4s , released after 4s. For each situation the unique message is sent to the bus. It is possible to choose for each button separately what massages should be sent.

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

Module sends message to the bus, when status of input changes. When LED status changes this frame is not sent. The table below shows meaning of each byte in the button frame.

Table 1. BUTTON frame

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	BUTTON	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 0x01 (button 1) – 0x06 (button 6)

BUTTON - actual input status
0x00 – open
0xFF – closed
0xFE – closed and held for 400ms
0xFD – closed and held for 4s
0xFC – closed and open within 400ms
0xFB – closed and open between 400ms and 4s
0xFA – closed and open after 4s

5.2. Status request

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

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 3). Meaning of bytes is the same as in Table 1.

Table 3. 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	BUTTON	0xFF	0xFF	0xFF	0xFF
0x301	0x1	Node Nr	Group Nr	0xFF	0xFF	0x02	BUTTON	0xFF	0xFF	0xFF	0xFF
...											
0x301	0x1	Node Nr	Group Nr	0xFF	0xFF	0x06	BUTTON	0xFF	0xFF	0xFF	0xFF

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

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 to 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

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

There is nothing to control in the module.

5.6. Configuration

With this version of application parameters below can be configured:
- Module identifier (module number and group number);
- Module description (16 chars);
- Button settings
- Text notes;

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.

5.6.2. 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.

5.6.3. Button settings

For each button it is possible to configure what type of button behaving is recognized by module. Module can recognize when:
- button is pressed,
- button is released,
- pressed and held for 400ms,
- pressed and held for 4s,
- pressed and released within 400ms (quick click),
- pressed and released between 400ms and 4s,
- pressed and released after 4s.

For each behaving a separate message will be sent on the bus.
WARNING: It is very important to choose only messages which will be used on the network to keep traffic on the bus as low as possible.

5.6.4. Text notes.

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

6. Document version

File	Note	Date
univ_v1-0-1-1a.pdf	Original version	March 2012