

### 1. Features

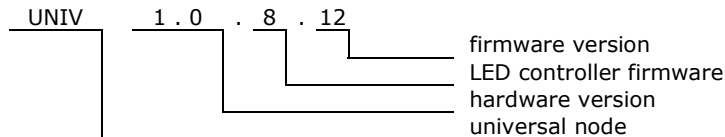
- Voltage control in 256 steps (0-255) for each channel
- Channel MASTER changeable in 256 steps as well
- Possibility to set up minimum and maximum value for each channel
- Last MASTER state memory
- Adjustable dimming time for each channel
- 33 control instructions
- 1s-20h timer for some instruction execution
- Allows to set up 128 conditions for receiving bus messages
- 3 blocking instructions
- Settable Power up value.
- Allows to write notes into processor



### 2. Compatibility:

- Firmware for **UNIV 1.0.8.1. application.**
- Firmware can be uploaded into devices with bootloader version 2.5 or compatible.

### 3. Firmware version



### 4. Operation overview

The module is a three channel LED controller with PWM (pulse width modulation). Each channel can be controlled in 256 steps. There is a fourth virtual MASTER channel, which dims all channels proportionally to keep set colour. The controller gives possibility to display one of 16 millions colours. It can react to 128 messages received from the bus. When message is received, the module can execute one of 33 instructions. Some of them can be executed with delay 1s-20h. After that the module sends information to the bus about new status. In this firmware blocking instructions have been implemented. They can block executing of other instructions save in module memory.

### 5. Firmware

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

#### 5.1. LED Controller message

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

Table 1. LED CONTROLLER frame

Frame type	Flags	Module	Group	D0	D1	D2	D3	D4	D5	D6	D7
0x308	3   2   1   0	Node Nr	Group Nr	0xFF	0xFF	CHANNEL	STATUS1	STATUS2	INSTR1	INSTR2	INSTR3

0x308	- universal module frame, led controller										
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 (0x01 - red, 0x02- green, 0x03 - blue, 0x04 - master)

**STATUS1** - actual status of channel from 0x00 to 0xFF

**STATUS2** - actual status of channels (0x00 - if all channels are switched off, 0xFF - if any of channels is turn on) - this byte can be used to turn on/off the relay, which toggles LED power supply.

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

**INSTR2** - second byte of instruction that is waiting for execution

**INSTR3** - third byte of instruction that is waiting for execution

**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	0xFF	0xFF	Node Nr	Group Nr	0xFF	0xFF	0xFF	0xFF

**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

**0xFF** - inessential data

As response the module will send LED CONTROLLER frame.

Table 3. LED CONTROLLER frame as response to STATUS REQUEST

0x308	0x1	Node Nr	Group Nr	0xFF	0xFF	0x01	STATUS1	STATUS2	INSTR1	INSTR2	INSTR3
0x308	0x1	Node Nr	Group Nr	0xFF	0xFF	0x02	STATUS1	STATUS2	INSTR1	INSTR2	INSTR3
0x308	0x1	Node Nr	Group Nr	0xFF	0xFF	0x03	STATUS1	STATUS2	INSTR1	INSTR2	INSTR3
0x308	0x1	Node Nr	Group Nr	0xFF	0xFF	0x04	STATUS1	STATUS2	INSTR1	INSTR2	INSTR3

The meaning of bytes is the same as in Table 1.

**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	0xFF	0xFF	Node Nr	Group Nr	0xFF	0xFF	0xFF	0xFF

**0x113** - 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

**0xFF** - 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

0x113 - UPTIME REQUEST frame

Node Nr - node number on the network  
 Group Nr - group number of the node on the network

UPTIME - 4-byte time in seconds UPTIME3 - MSB, UPTIME0 - LSB

**5.4. Module control**

The module can be controlled directly from PC, or indirectly by other modules. In both situation all 33 instruction can be used.

**5.4.1. Control instruction**

The Table 6 shows all instructions, which can be executed by LED controller. The byte INSTR1 defines instruction. Bytes INSTR2 and INSTR3 are additional for some instructions.

Table 6. LED Controller Instructions

Instruction	Instruction code			Description
	INSTR1	INSTR2	INSTR3	
Set LED1 (R) to...	0x00	0x00-0xFF	TIMER	Sets channel immediately to value shown by INSTR2.
Set LED2 (G) to...	0x01	0x00-0xFF	TIMER	
Set LED3 (B) to...	0x02	0x00-0xFF	TIMER	
Set MASTER to...	0x03	0x00-0xFF	TIMER	If channel was on it turns it off, if it was off it turns it on to max value
Toggle LED1	0x04	0xFF	TIMER	
Toggle LED2	0x05	0xFF	TIMER	
Toggle LED3	0x06	0xFF	TIMER	
Toggle MASTER	0x07	0xFF	TIMER	
Step down LED1	0x08	0xFF	TIMER	
Step down LED2	0x09	0xFF	TIMER	
Step down LED3	0x0A	0xFF	TIMER	Decreases value by one
Step down MASTER	0x0B	0xFF	TIMER	
Step up LED1	0x0C	0xFF	TIMER	
Step up LED2	0x0D	0xFF	TIMER	Increases value by one
Step up LED3	0x0E	0xFF	TIMER	
Step up MASTER	0x0F	0xFF	TIMER	
Set LED1 (R) softly to...	0x10	0x00-0xFF	TIMER	Sets channel with programmed speed to value shown by INSTR2
Set LED2 (G) softly to...	0x11	0x00-0xFF	TIMER	
Set LED3 (B) softly to...	0x12	0x00-0xFF	TIMER	
Set MASTER softly to...	0x13	0x00-0xFF	TIMER	
Stop LED1	0x14	0xFF	0xFF	Stops instruction that is being executed (e.g. instruction START)
Stop LED2	0x15	0xFF	0xFF	
Stop LED3	0x16	0xFF	0xFF	
Stop MASTER	0x17	0xFF	0xFF	
Start LED1	0x18	0xFF	0xFF	Instruction START begins typical control process. If within 400ms from START instruction, controller receives STOP instruction then it toggles channel's state (exactly the same as TOGGLE instruction). If after that time instruction STOP is not received then channel dims up (if previous state was min or 0x00), or dims down (if previous state was max). It gives possibility to control with one button. If button is pressed for less than 400ms the channel will turn on or off, If button is pressed for longer than 400ms, then channel will dim.
Start LED2	0x19	0xFF	0xFF	
Start LED3	0x1A	0xFF	0xFF	
Start MASTER	0x1B	0xFF	0xFF	
Set RGB to...	0x1C	0x00-0xFF	0x00-0xFE	This sets values of three channels at the same time. Values are defined by INSTR2 & INSTR3 according to rule: value LED1 = INSTR2<7:3> - 5 most significant bits of INSTR2; value LED2 = INSTR2<2:0>INSTR3<7:6> - 3 less significant bits of INSTR2 & 2 most significant bits of INSTR3; value LED3 = INSTR3<5:1> - bits from 1 to 5 of INSTR3. The value of each channel has 5 bit resolution. The less significant bit of INSTR3<0> is not used.
Set RGB speed to...	0x1D	0x00 - 1s 0x01 - 5s 0x02 - 10s 0x03 - 15s 0x04 - 20s 0x05 - 25s 0x06 - 30s 0x07 - 35s 0x08 - 40s 0x09 - 45s 0x0A - 50s 0x0B - 55s 0x0C - 60s	0xFF	It set dimming speed for all three channels.
RGB speed up	0x1E	0xFF	0xFF	It changes dimming speed for all three channels to next lower value

RGB speed down	0x1F	0XXX	0XXX	It changes dimming speed for all three channels to next higher value
Program	0x20	0x00	0XXX	It stops any running program
		0x01	0XXX	Program1 – every program calling causes colour changing with sequence: 1- red 2- green 3- blue 4- yellow 5- cyan 6- violet 7- white
		0x02	0XXX	Program 2 – first program calling runs automatic colour changing with set dimming speed; second program calling stops colour.

0XXX – any value

**5.3.2. Timer**

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

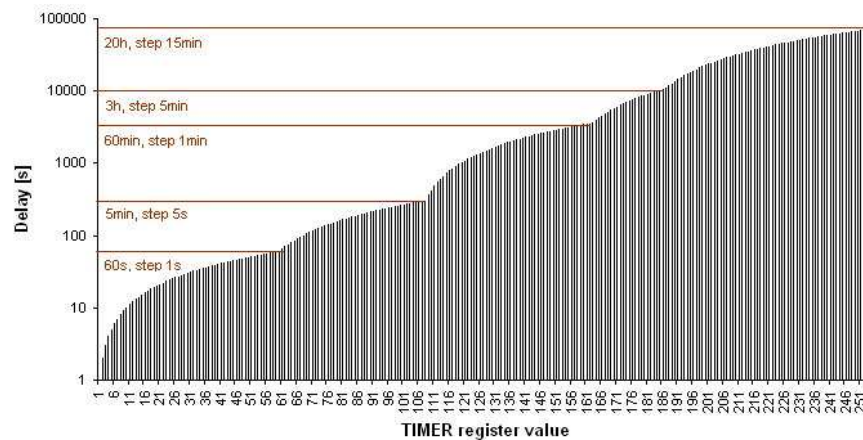


Figure 1. Delay/timer register relationship

**5.4.3. 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 7. 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 – instruction to be executed (byte3)
- 0XXX – any value

**5.4.4. 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.

**5.5. Configuration**

With this version of firmware parameters below can be configured:

- Module identifier (module number and group number);
- Module description (16 chars);
- Minimum and maximum values;
- Dimming speed;
- Power up values;
- Last state memory;
- Linking device with other modules (indirect control of module).

Configuration process can be done by using HAPCAN Programmer.

**5.5.1. 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.

**5.5.2. Module description**

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

**5.5.3. Minimum and maximum values**

It is possible to set minimum and maximum values for each channel.

**5.5.4. Dimming speed**

This parameter defines how fast channel goes from value 0x00 to value 0xFF.

**5.5.5. Power up values**

It is possible to configure value for each channel on power up.

**5.5.6. Last state memory**

The last state of MASTER channel can be remembered. In memory mode, when MASTER channel is switched on, it goes to the value that was before switching off. In no memory mode it goes to the maximum value.

**5.5.7. Notes.**

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

**5.5.8. 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 8. 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_v1-0-8-12a.pdf	Original version	November 2010