

Universal serial bus musical instrument digital interface hardware

Dalibor Slovák

Abstract— this article explores possibilities of coordination between computer made music and other stage devices. Purpose of develop was to create simple universal hardware for using in music software communicating via MIDI protocol and USB. This hardware is meaning for professional music live performance. At the input this device communicates via MIDI and USB protocols. At the output is tension. Its value is managing by pulse-width modulation. This modulation value is given to value of the third byte in MIDI message. This value is sound power of control note. There is device is connected as one of USB interfaces at the input. There is device is connected to power transformation device on output. There are artificial smokes, lights and fountains are connected to power transformation device as ending equipment.

Keywords—USB, MIDI, pulse width modulation, Microchip, power transformation, MIDI message

I. INTRODUCTION

Usage of computers in music is standard how for live performance so for music recording. MIDI protocol is the most used for the live performance. It is basic for some setting of musical instruments on the stage. For other equipment, e.g. sound effects too. Some musicians can use MIDI and use it. Some musicians don't know MIDI, but use MIDI too, because MIDI is standard in electronic musical instruments. MIDI is often one of inner components in keyboards, electrical string instruments or wind MIDI instruments. These MIDI components don't need human control.

Today, MIDI is a part of effects units for all music instruments. Whether are electric guitars, keyboards or some effects for singers or acoustical instrument. MIDI protocol is part of other stage equipment. Consequently MIDI is part of stage lights and artificial smokes. Current development and usage of this protocol especially focuses to music instruments settings and other equipment on stage. Thus, time coordination

Manuscript received May 28, 2011: This work was performed with financial support of research project NPV11-2C06007, by the Ministry of Education of the Czech Republic. This work was supported by the grant Internal Grant Agency of Tomas Bata University under the project No. IGA SV3011148020.

Dalibor Slovák Author is with the Department of Computer and Communication systems, Tomas Bata University in Zlin, Faculty of Applied Informatics nám. T.G.Masaryka 5555, 760 01 Zlin Czech republic slovak@fai.utb.cz

with other visual effects without human control. In this development, the MIDI protocol is used in combination with the most modern technologies. Very often is used with USB protocol and devices.

Research target was to design simple, the most inexpensive, simply programmable high variable device. Variability was intended for various lights systems. Important for design was real-time application. Accordingly zero MIDI USB transfer delay for live music performance. Simply put, visual effect must correspondence music performance in real-time. From the begging was evident that it will be one chip device with small requirements for voltage, maintenance and space. There is also a small percentage of failure as an additional criterion for the development of the equipment described below.

II. AVAILABLE SOLUTIONS

Between existing solutions in this area at present belongs to the following applications

A. Harvey Twyman device

Harvey Twyman hardware module consists of an integrated circuit HC11, produced by Motorola. This circuit sends MIDI messages to another device, which is Altera 8254 FPGA (Field Programmable Gate Array). This module provides a total of 12 channels. Rated power of each channel can be up to 300W. Thus conceived control provides 128 levels of light for each light. Setting these levels can then be carried out directly from the editing programs such as Cubase. This setting can be used specifically Key Graphics Editor, List Editor, or Mixer Maps Editor. Author uses the last of the editors. The editors are part of the music software Cubase.

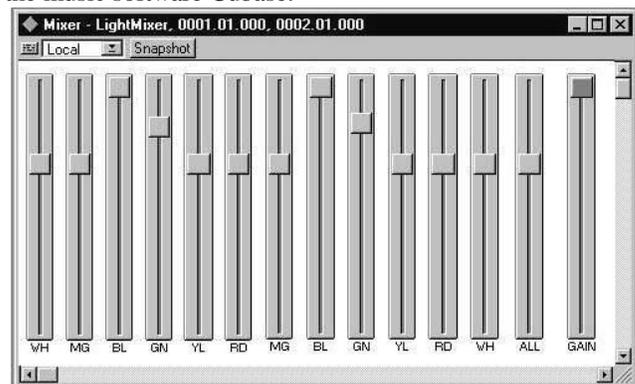


Fig. 1. Setting the color intensity MixerMaps edit in Cubase

Each light channel has three parameters, which can control the channel. There are the level of the channel (channel level), channel gain (Gain channel) and the overall level (Master Gain):

Channel Level - the range is from zero up to 127. Settings this parameter but the parameter is closely related to the overall level (see below).

Gain Channel - serves as the setting of the luminance channel, as they are, according to the author, often seems lighter than it should be.

Master Gain - adjust the overall brightness of all channels[2].

B. BOTEX company and its solution

Other devices that use the MIDI protocol to control stage lighting equipment is of Botex company. These devices are combining MIDI with DMX512 protocol. The MIDI protocol is used primarily by Note On event and Note Off event. As a demonstration model I picked a four-channel dimmer Botex MPX – 405. Among its features include:

- Control via MIDI and DMX

- infrared remote control
- 4 channel output for dimming
- the possibility of linking multiple units (Link Up)
- Last setting memory[3].

C. Tom Scarff MIDI hardware

The author of this design comes from Dublin, where he works in Dublin Institute of Technology. The board is equipped with a chip, MIDI connectors, necessary LEDs, and the connector for the nine-volt battery, which serves as the power supply.

The PIC16C84 microcontroller is centre of this module, along with the option to select those channels. Then select the output signal from the MIDI channel is not resolved through programmed utilities, but is dealt with firmly, hardware. This is the simplest case for lights, extension of other electrical equipment, using only the Note On command, respectively On, Off command. Controlled lights are only switched. Intensity is not controlled [4].

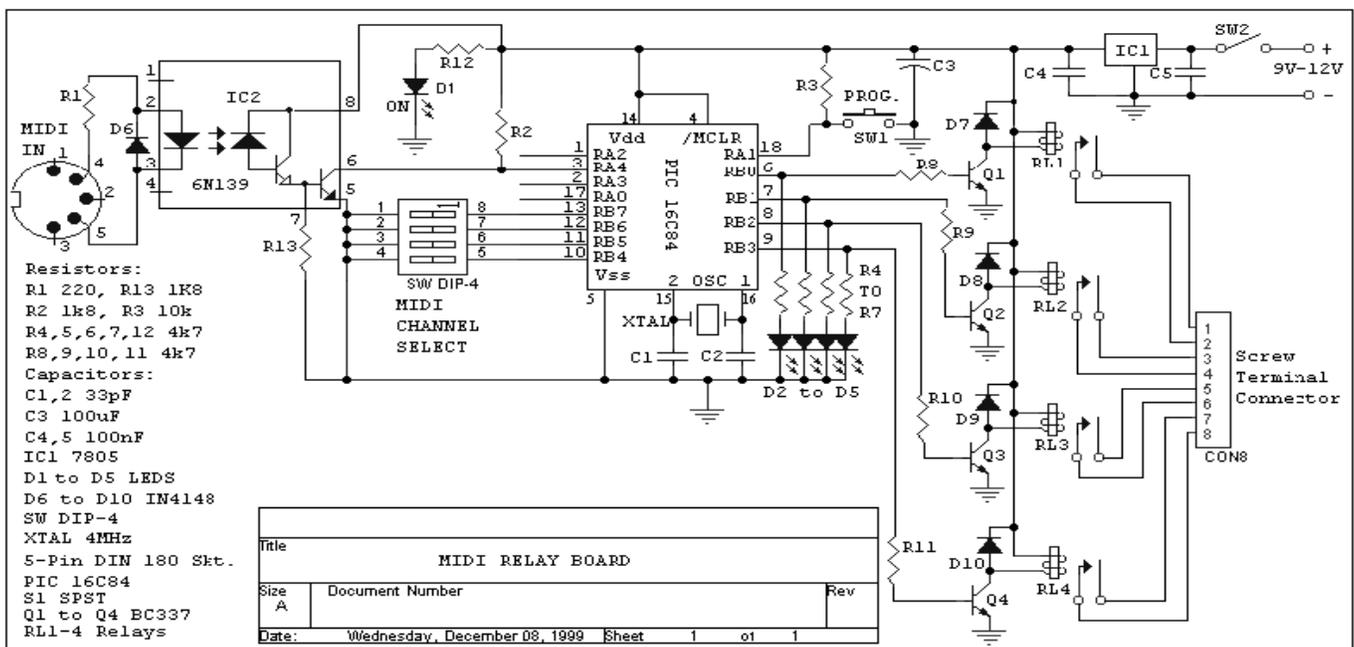


Fig. 2. Tom Scarff MIDI hardware solution scheme [4]

D. Darys solution

This polish company develops MIDI lights drivers. They have a few models of these drivers [13].

1) ML-80P2

MIDI messages manage this equipment. There are Note On, Note Off, Program Change and Control Change kinds of MIDI messages. ML-80P2 is available to manage even eight lights channels. Output tension is from 0 to 10 volts. You can set lights intensity from 0 to 127 values. 127 is maximum value,

because one MIDI message has 7 bits for this value. This is necessary external power supply for ML-80P2.



Fig. 3. MIDI controller Darys ML-80P2

2) *Modul ML-640 PS*

This is leading product of Darys. There is DMX512 protocol is on output. You need lights, which are able to communicate this protocol. The controller is able to work in one of four various modes, as the: 8, 16, 32 or 64 channel device. It has the memory of 32 programs at the work mode operating 64 DMX channels. The number of available programs is increasing along with the choice of the mode operating the smaller number of DMX channels (table below).

Table I. VIEW OF ML-640 PS MODES

Mode	Number of programs	Number of scenes	Program numbers
mode 8	256	2048	A01-A64, B01-B64, C01-C64, D01-D64
mode 16	128	1024	A01-A32, B01-B32, C01-C32, D01-D32
mode 32	64	512	A01-A16, B01-B16, C01-C16, D01-D16
mode 64	32	256	A01-A08, B01-B08, C01-C08, D01-D08

Applications are divided into 4 banks: A, B, C and D. Each program is consisting of 8 scenes with the possibility to shorten then, id est every program can have from 1 to 8 scenes. All programs are being programmed by the user according to own needs. It is possible in the given mode to gather available programs to sequences. There is 16 sequences (from S01 to S16), every sequence is able to combine from at most 8 programs (it is possible to reduce the quantity of programs)[13].



Fig. 4. MIDI controller Darys ML-640 PS

Apart from programs making it possible to arrange and remembering particular scenes in programs characterised above, the controller possesses two programs more: P01 and P02. The P01 is making it possible to steer MIDI orders: Note On, Note Off, Program Change and some Control Change orders belonging to the MLS group (MIDI Light System). The P02 is working similarly to P01 but doesn't react to Note On and Note Off orders. Such a division of programs makes it possible to steer the controller with MIDI orders for various methods in dependence on the source of the control (the musical instrument, the sequencer, the computer) and the kind of effects which we intend to obtain. Levels are being regulated in these programs in all DMX channels in the real time in accordance to received MIDI orders.

III. OWN SOLUTION

This device consists of a CPU, which is in this case, the PIC 18F2550 by Microchip company, 8 LEDs, crystal and necessary stabilizing capacitors. An integral part is a USB type B connector, for connection to a PC [12] [11].

A. *Microchip PIC 18F2550*

Heart of this module is microcontroller by Microchip company with type signature PIC 18F2550. It is single chip, which is compatible with USB protocol version 2.0. It also supports both USB transfer types, both low speed (1.5 Mbps) and full speed at 12 Mbit / s. It allows all types of USB transfers, so to ensure all possible available functionality in the USB protocol. The processor supports full number of two-way endpoints. In Run mode is controller, when running the processor and peripherals. In the Run modes, clocks to both the core and peripherals are active. The difference between these modes is the clock source. In Idle mode runs only the peripherals. Sleep mode is set, even when is not running CPU or peripherals. The device can be connected to two external oscillator frequency of up to 48 MHz. The controller has its internal oscillator too. The user can choose from a total of eight oscillation frequencies between 31 kHz to 8 MHz. The device is also possible to change the polarity, which has used during creation the final project. The first in a series of devices have switching diodes in a logical 1, and then there was a reversal of values taken by the Velocity parameter, which is contained in the third byte of MIDI message. The processor has 100 000x rewritable memory for program and 1000 000x rewritable EEPROM. It also includes 32 tier stack and instruction set, which contains 105 of system instructions. Function Code Protection prevents entry into selected areas of memory. There is also support for ICSP programming that allows you to program the processor embedded in the PCB. There are of the CPU PIC18F2550 16 - bit comparator (SPI - Serial Interface Peripheral) and last but not least, EUSART [10]. The serial communication module, based on the standard RS-232

protocol provides support for the LIN bus standard. EUSART also includes automatic detection of baud rate and 16-bit baud rate generator. If the microcontroller used in the internal oscillatory block EUSART is used in place of communication, where nurses access to unused external oscillator, avoiding mistakes in the requirements for induction [9].

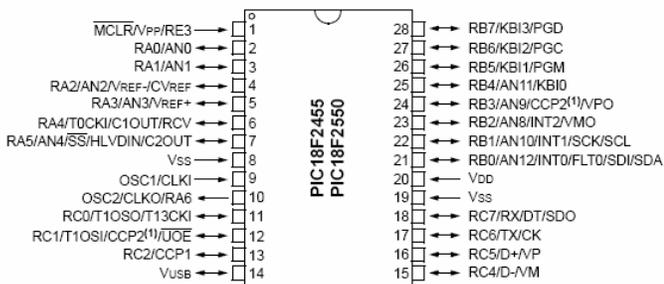


Fig. 5. Microchip 18F2550 pin schema

1) Microchip 18F2550 the most important parts

In this chapter I will closely describe important Microchip PIC18F2550 parts. These parts have an important influence on development of my device. Due to these characteristics I chose this MCU.

a) Universal serial bus

Some special hardware features have been included to improve performance. Dual port memory in the device's data memory space (USB RAM) has been supplied to share direct memory access between the microcontroller core and the SIE. Buffer descriptors are also provided, allowing users to freely program endpoint memory usage within the USB RAM space. A Streaming Parallel Port has been provided to support the uninterrupted transfer of large volumes of data, such as isochronous data, to external memory buffers.

b) USB Status and control

The operation of the USB module is configured and managed through three control registers. In addition, a total of 22 registers are used to manage the actual USB transactions. The registers are:

- USB Control register (UCON)
- USB Configuration register (UCFG)
- USB Transfer Status register (USTAT)
- USB Device Address register (UADDR)
- Frame Number registers (UFRMH:UFRML)
- Endpoint Enable registers 0 through 15 (UEPn)

c) External clock input

The EC, ECIO, ECPLL and ECPIO Oscillator modes require an external clock source to be connected to the OSC1 pin. There is no oscillator start-up time required after a Power-on Reset or after an exit from Sleep mode. In the EC and ECPLL Oscillator modes, the oscillator frequency divided by 4 is available on the OSC2 pin. This signal may be used for test purposes or to synchronize other logic. Figure 2-4 shows the pin connections for the EC Oscillator mode.

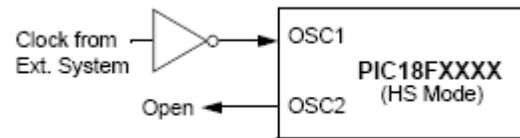


Fig. 6. External Clock input – HS mode

d) nanoWatt TECHNOLOGY

All of the devices in the PIC18F2455/2550/4455/4550 family incorporate a range of features that can significantly reduce power consumption during operation.

Key items include:

- **Alternate Run Modes:** By clocking the controller from the Timer1 source or the internal oscillator block, power consumption during code execution can be reduced by as much as 90%.
- **Multiple Idle Modes:** The controller can also run with its CPU core disabled but the peripherals still active. In these states, power consumption can be reduced even further, to as little as 4% of normal operation requirements.
- **On-the-Fly Mode Switching:** The power-managed modes are invoked by user code during operation, allowing the user to incorporate power-saving ideas into their application's software design.
- **Low Consumption in Key Modules:** The power requirements for both Timer1 and the Watchdog Timer are minimized [9].

e) MULTIPLE OSCILLATOR OPTIONS AND FEATURES

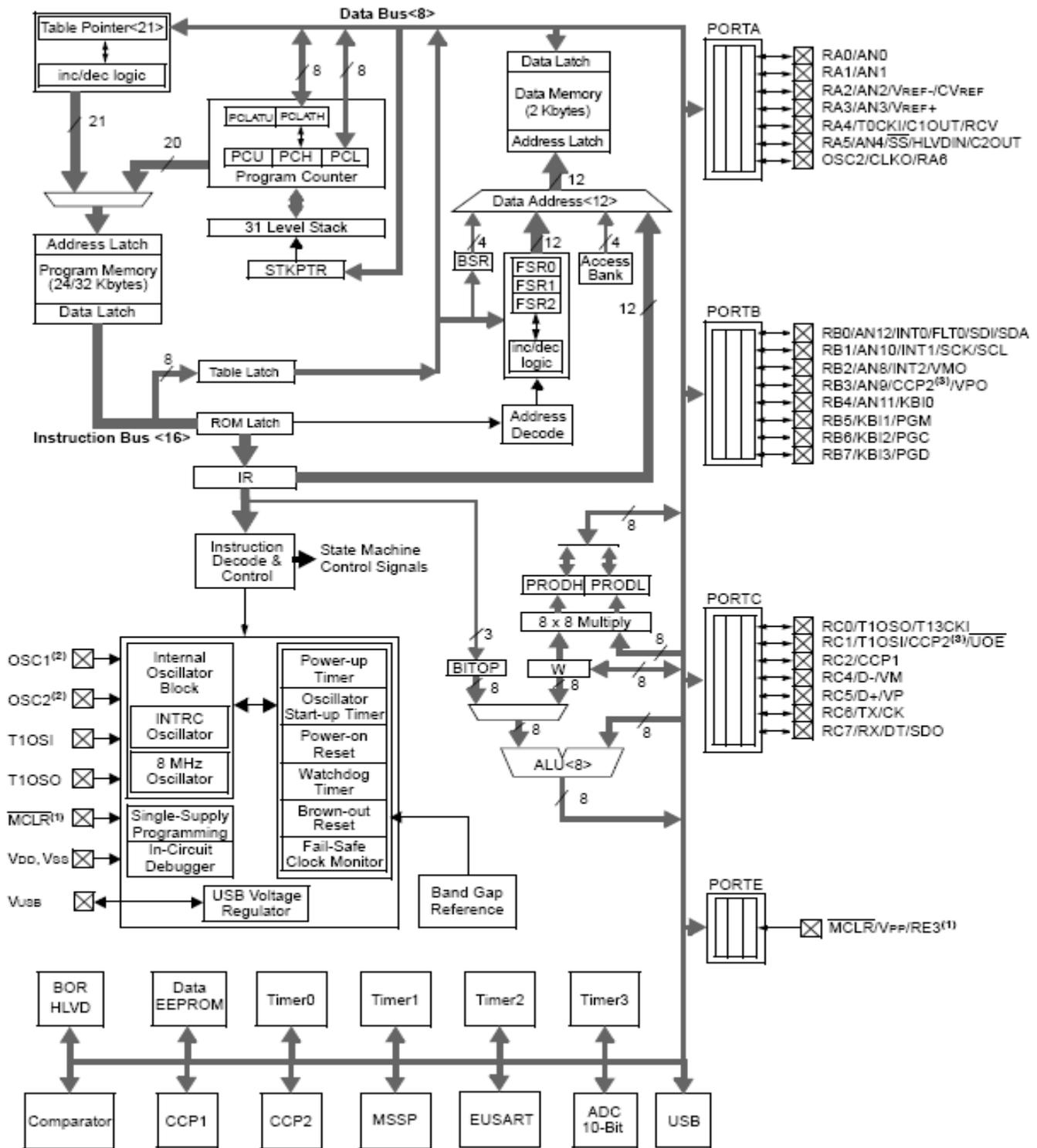
PIC18F2550 offer twelve different oscillator options, allowing users a wide range of choices in developing application hardware. These include:

- Four Crystal modes using crystals or ceramic resonators.
- Four External Clock modes, offering the option of using two pins (oscillator input and a divide-by-4 clock output) or one pin (oscillator input, with the second pin reassigned as general I/O).
- An internal oscillator block which provides an 8 MHz clock ($\pm 2\%$ accuracy) and an INTRC source (approximately 31 kHz, stable over temperature and V_{DD}), as well as a range of 6 user-selectable clock frequencies, between 125 kHz to 4 MHz, for a total of 8 clock frequencies. This option frees an oscillator pin for use as an additional general purpose I/O.
- A Phase Lock Loop (PLL) frequency multiplier, available to both the High-Speed Crystal and External Oscillator modes, which allows a wide range of clock speeds from 4 MHz to 48 MHz.
- Asynchronous dual clock operation, allowing the USB module to run from a high-frequency oscillator while the rest of the microcontroller is clocked from an internal low-power oscillator. Besides its availability as a clock source, the internal oscillator block provides a stable reference source that gives the family additional features for robust operation:
- **Fail-Safe Clock Monitor:** This option constantly monitors the main clock source against a reference signal provided by the internal oscillator. If a clock failure occurs, the controller is

switched to the internal oscillator block, allowing for continued low-speed operation or a safe application shutdown.

• *Two-Speed Start-up*: This option allows the internal oscillator to serve as the clock source from Power-on Reset, or

wake-up from Sleep mode, until the primary clock source is available [9].



Note 1: RE3 is multiplexed with MCLR and is only available when the MCLR Resets are disabled.
Note 2: OSC1/CLKI and OSC2/CLKO are only available in select oscillator modes and when these pins are not being used as digital I/O. Refer to Section 2.0 "Oscillator Configurations" for additional information.
Note 3: RB3 is the alternate pin for CCP2 multiplexing.

Fig. 7. Microchip 18F2550 block diagram [9]

f) *OSCILLATOR CONFIGURATIONS*

Devices in the PIC18F2550 family incorporate a different oscillator and microcontroller clock system than previous PIC18F devices. The addition of the USB module, with its unique requirements for a stable clock source, make it necessary to provide a separate clock source that is compliant with both USB low-speed and full-speed specifications. To accommodate these requirements, PIC18F2550 devices include a new clock branch to provide a 48 MHz clock for full-speed USB operation. Since it is driven from the primary clock source, an additional system of prescalers and postscalers has been added to accommodate a wide range of oscillator frequencies. Other oscillator features used in PIC18 enhanced microcontrollers, such as the internal oscillator block and clock switching, remain the same.

(1) *OSCILLATOR CONTROL*

The operation of the oscillator in PIC18F2550 devices is controlled through two Configuration registers and two control registers. Configuration registers, CONFIG1L and CONFIG1H, select the oscillator mode and USB prescaler/postscaler options. As Configuration bits, these are set when the device is programmed and left in that configuration until the device is reprogrammed. The OSCCON register (Register 2-2) selects the Active Clock mode; it is primarily used in controlling clock switching in power-managed modes. The OSCTUNE register (Register 2-1) is used to trim the INTRC frequency source, as well as select the low-frequency clock source that drives several special features.

g) *Oscillator Types*

PIC18F2455/2550/4455/4550 devices can be operated in twelve distinct oscillator modes. In contrast with previous PIC18 enhanced microcontrollers, four of these modes involve the use of two oscillator types at once.

Users can program the FOSC3:FOSC0 Configuration bits to select one of these modes:

1. XT Crystal/Resonator
2. XTPLL Crystal/Resonator with PLL enabled
3. HS High-Speed Crystal/Resonator
4. HSPLL High-Speed Crystal/Resonator with PLL enabled
5. EC External Clock with Fosc/4 output
6. ECIO External Clock with I/O on RA6
7. ECPLL External Clock with PLL enabled and Fosc/4 output on RA6
8. ECPIO External Clock with PLL enabled, I/O on RA6
9. INTHS Internal Oscillator used as microcontroller clock source, HS
Oscillator used as USB clock source
10. INTXT Internal Oscillator used as microcontroller clock source, XT
Oscillator used as USB clock source
11. INTIO Internal Oscillator used as microcontroller clock source, EC
Oscillator used as USB clock source, digital I/O on RA6

12. INTCKO Internal Oscillator used as microcontroller clock source, EC

Oscillator used as USB clock source, Fosc/4 output on RA6

(1) *OSCILLATOR MODES AND USB OPERATION*

Because of the unique requirements of the USB module, a different approach to clock operation is necessary. In previous PIC® devices, all core and peripheral clocks were driven by a single oscillator source; the usual sources were primary, secondary or the internal oscillator. With PIC18F2455/2550/4455/4550 devices, the primary oscillator becomes part of the USB module and cannot be associated to any other clock source. Thus, the USB module must be clocked from the primary clock source; however, the microcontroller core and other peripherals can be separately clocked from the secondary or internal oscillators as before.

Because of the timing requirements imposed by USB, an internal clock of either 6 MHz or 48 MHz is required while the USB module is enabled. Fortunately, the microcontroller and other peripherals are not required to run at this clock speed when using the primary oscillator. There are numerous options to achieve the USB module clock requirement and still provide flexibility for clocking the rest of the device from the primary oscillator source [9].

h) *PWM Mode*

In Pulse-Width Modulation (PWM) mode, the CCPx pin produces up to a 10-bit resolution PWM output. Since the CCP2 pin is multiplexed with a PORTB or PORTC data latch, the appropriate TRIS bit must be cleared to make the CCP2 pin an output. Figure 8 shows a simplified block diagram of the CCP module in PWM mode.

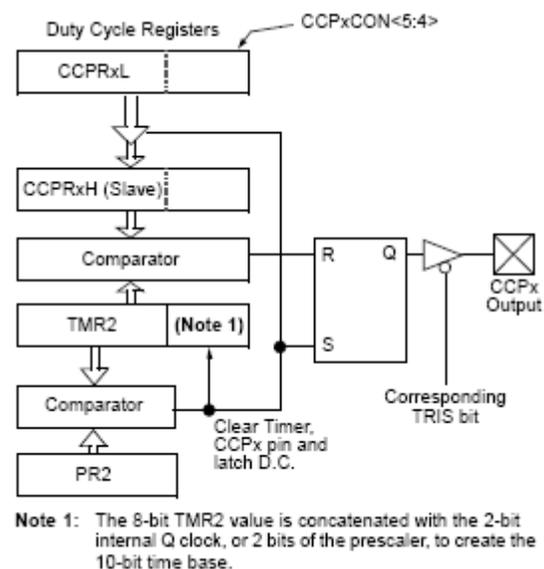


Fig. 8. Simplified PWM block diagram

A PWM output (Figure 9.) has a time base (period) and a time that the output stays high (duty cycle). The frequency of the PWM is the inverse of the period (1/period).

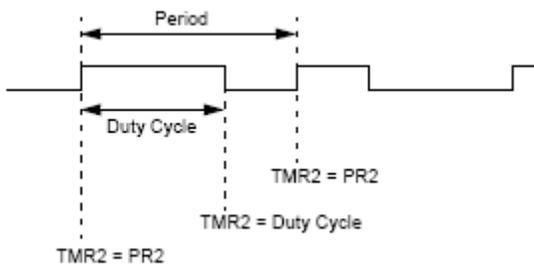


Fig. 9. PWM Output

(1) PWM Period

The PWM period is specified by writing to the PR2 register. The PWM period can be calculated using the following formula:

$$PWM\ Period = [(PR2) + 1] \cdot 4 \cdot TOSC \cdot (TMR2\ Prescale\ Value)$$

PWM frequency is defined as $1/[PWM\ period]$. When TMR2 is equal to PR2, the following three events occur on the next increment cycle:

- TMR2 is cleared
- The CCPx pin is set (exception: if PWM duty cycle = 0%, the CCPx pin will not be set)
- The PWM duty cycle is latched from CCPRxL into CCPRxH.

i) Universal Serial Bus

The PIC18F2550 device contains a full-speed and low-speed compatible USB Serial Interface Engine (SIE) that allows fast communication between any USB host and the PIC® microcontroller. The SIE can be interfaced directly to the USB, utilizing the internal transceiver, or it can be connected through an external transceiver. An internal 3.3V regulator is also available to power the internal transceiver in 5V applications. Some special hardware features have been included to improve performance. Dual port memory in the device's data memory space (USB RAM) has been supplied to share direct memory access between the microcontroller core and the SIE. Buffer descriptors are also provided, allowing users to freely program endpoint memory usage within the USB RAM space. A Streaming Parallel Port has been provided to support the uninterrupted transfer of large volumes of data, such as isochronous data, to external memory buffers [5].

B. USB MIDI Lights Device

USB MIDI Lights Device plate was designed as a versatile development board for PIC18F2550 microcontroller applications with emphasis on the use of USB microcontrollers [1]. Power board is done via the USB port, which provides a stable 5V. To filter this voltage is added inductance (ferrite bead) and 100µF electrolytic capacitor. Microcontroller itself also contains a stabilizer 3.3V, which is used to stabilize the output 1µF ceramic capacitor. The power supply also connected blocking capacitor 100nF located between GND (ground) and VCC pin microcontroller [8]. The source for generating the clock signal is 20MHz XTAL

with two 15pF ceramic capacitors. The value of the crystal was chosen because of its availability, the actual microcontroller allows the use of crystal in the values (4, 8, 12, 16, 20, 24, 40, 48 MHz). Microcontroller is also equipped with an internal RC oscillator, but using the USB connection you must use the exact source of the clock frequency - crystal. The connectors J2, J3 and J5 are connected input / output pins of the microcontroller. The J2 connector pins are connected RC6 and RC7 - microcontroller serial port and ground. The J3 connector pins are connected to gate B (RB0 - RB7), which are parallel connected to 5V LED series resistor. J5 connector includes pins Gate A (RA0-RA5) and GND. Other connectors on the board's jumper, which is used to activate the bootloader in the normal mode is disconnected, the short-circuited in bootloader mode. The basic program board loader - bootloader has been programmed into the board using an external programmer connected to connector J4, which is represented on the board pads for soldering programmer wires.

Reset the microcontroller is connected through resistor 10 k ohms or the power supply. The device has a total of eight functional outputs (channels).

MIDI Protocol

This text part will be describes the MIDI protocol and USB. The basis of the MIDI communication is called a MIDI message, which consists of three bytes. Each MIDI message (event) is therefore presented as three eight-digit binary values, which are made up of zeros and ones. Each MIDI message can then contain in the each byte value from 0 to 255, for a total of 256 different values. MIDI messages are divided into two basic categories: Status messages and Data messages. Status message determines the type of information that is sent via MIDI. Indicates a device that receives a message that the event belongs to which channel the MIDI event belongs and what it is. It may be an event: Note On, pitch change, Program Change (patch change) and after touch (the last event occurs when it is developed further pressure on already depressed note). Data bytes contained in the device again informed about what values are assigned to events, which carries the status byte [7].

C. USB MIDI Event packet

MIDI data is transmitted via USB using 32 - bit MIDI Event Packet. Data transmission is performed using the standard reports of four bytes. With this USB MIDI Event Packet is to create a virtual connection between the endpoints USB host and USB MIDI devices. This method of connection is advantageous for its low, which does not require a large number of endpoints, like other types of USB devices. Each MIDI event has its own USB MIDI packet, which prevents creation of many mistakes.

Table II. USB MIDI EVENT PACKET STRUCTURE

Byte 0		Byte 1	Byte 2	Byte 3
CN cable number	CIN Code Index - Number. MIDI generic classification	MIDI First byte	MIDI Second byte	MIDI Third byte

The first four bytes starting at the MSB position contains information on the number of virtual MIDI cable, which it is transmitted by given MIDI information. The value of CN is an indication of the range 0x0 through 0xF indicates the number of the embedded jack, through which there is a link with appropriate MIDI functionality. Second nibl LSB has ended, then identification of the MIDI message. Table 1 shows how different byte write MIDI messages from the USB MIDI packet, which must be submitted if it will be to communicate and receive MIDI information via the USB protocol.

Table III. CODE INDEX NUMBER AND CORRESPONDING MIDI MESSAGES

CIN	MIDI_x Size	Description
0x0	1, 2 or 3	Vacant value, reserved for others enlargement.
0x1	1, 2 or 3	Vacant value, reserved for others enlargement.
0x2	2	Two bytes MIDI messages. E.g. Time Clock or SongSelect
0x3	3	Three bytes MIDI messages
0x4	3	MIDI messages System Exclusive type run or continue
0x5	1	One byte System common or SysEx messages with command end and next one byte information.
0x6	2	SysEx messages with nex two bytes information
0x7	3	SysEx messages with nex three bytes information
0x8	3	Nota off
0x9	3	Nota On
0xA	3	Common pressure sensitivity
0xB	3	MIDI Control Change
0xC	2	Program change
0xD	2	Pressure sensitivity
0xE	3	Tune change
0xF	1	Single Byte

USB MIDI Event Packet is the final part MIDI and USB communication. In the beginning it is necessary that the device is configured and programmed especially visible as a USB interface for the system. Individual sub-categories for this USB device as USB Audio and USB Audio sub-MIDI interface. Various communication interfaces, therefore must be programmed according to these standards [6].

IV. CONCLUSION

The proposal USB MIDI Lights Device was to create a financially optimized, the cheapest possible, programmable robust equipment. The basic advantage of the device is manufacturing cost. This cost is very low. The device will be available only to the USB connector and a jumper or switch for easy re-programming the application itself for changes in light assembly. An important benefit is the ability to exploit any effect devices, regardless of which protocol is or is not in their software toolkit. The only one prerequisite is to manage the output of the light kit with inductors for large wattage because of reasons of intensity control by pulse width modulation. In this time is developed other new version of hardware device and its management software.

V. ACKNOWLEDGEMENTS

This work was performed with financial support of research project NPVII-2C06007, by the Ministry of Education of the Czech Republic. This work was supported by the grant Internal Grant Agency of Tomas Bata University under the project No. IGA SV30111148020.

REFERENCES:

- [1] R. Grossmann, "Quartz crystals as remote sensors or tire pressure," *Proceedings of the 16th IEEE instrumentation and Measurement Technology conference, IMTC/99, May 1999.*
- [2] Harvey Twyman MIDI hardware [available online] <<http://www.twyman.org.uk/midilite.htm>>
- [3] Botex MPX405 [available online]: <<http://eshop.prodance.cz/Files/MPX405%20MultiPak%20cz.pdf>>
- [4] MIDI lights settings [online]. [available online]: <<http://tomscarff.tripod.com>>
- [5] *Universal Serial Bus specification.pdf* [available online] <<http://www.usb.org>>
- [6] *Universal Serial Bus Device Class Definition for MIDI Devices.pdf* [available online]. <<http://www.usb.org>>
- [7] *Universal Serial Bus Device Class Definition for Audio Devices.pdf* [available online]. <<http://www.usb.org>>
- [8] R. Glaschick, B. Oesterdieckhoff, C. Loeser, "Service Oriented Interface Design for Embedded Devices", 10th IEEE International Conference on Emerging Technologies and Factory Automation –ETFA, 2005
- [9] *PIC18F2455/2550/4455/4550 Data Sheet.pdf* [available online]. <<http://www.microchip.com>>
- [10] *MPLAB C18 C Compiler User's guide.pdf* [available online]. <<http://www.microchip.com>>
- [11] Huertas JL, Sanchez-Solano S, Barriga A, "hardware implementation using AID VLSI techniques" In: *Proceedings of the 8th Int. Confer. OnFuzzy Logic and Neural Networks*, Iizuka, Japan, July 17-22, 2004. p. 535-8
- [12] Slovák, D. *Protocol MIDI and stage equipment: Tomas Bata University*, Faculty of Applied Informatics, Department of Applied Informatics, 2008, 64 p. Thesis supervisor prof. Ing. Vladimír Vašek, CSc.
- [13] *Darys company products* [available online]. <<http://www.konin.pl/darys/ainfo.htm>>