

Płyta ewaluacyjna i klawiatura muzyczna MIDI AVT-960

Zgodnie z zapowiedziami w cyklu artykułów kursowych na temat MIDI, nadszedł czas na omówienie narzędzi ułatwiających implementację MIDI we własnych projektach. Zaprezentujemy bibliotekę MIDI w języku C dla mikrokontrolerów AVR. Jako przykład jej zastosowania zostanie przedstawiona prosta klawiatura muzyczna, zbudowana na bazie nieskomplikowanej płyty rozwojowej MIDI, opracowanej specjalnie na potrzeby kursu.

Rekomendacje: płyta ewaluacyjna MIDI z pewnością zainteresuje muzykujących elektroników, jako że podobnych projektów nie ma za wiele. Jest łatwa w montażu i uruchomieniu, ale wymaga umiejętności pisania programów w języku C dla mikrokontrolerów AVR (AVRGCC).

UWAGA!

Wszystkie listingi publikujemy na CD-EP12/2006B

PODSTAWOWE PARAMETRY

- Płytki o wymiarach: 93x106 mm (płyta bazowa), 161x80 mm (płytki klawiatury)
- Zasilanie ≥ 8 VDC
- Gniazda we/wy standardu MIDI: MIDI IN, MIDI THRU, MIDI OUT (wszystkie typu DIN)
- Optoizolacja wejść
- Gniazda uniwersalnego zastosowania: PA0...PA7, PB0...PB7, PD2...PD7, PC6, PC7
- Złącze programowania ISP (zgodne z STK200/300)
- Gniazdo wyświetlacza LCD (LCD zgodny z HD44780)
- Klawiatura: 1 oktawa (12 półtonów), możliwość łączenia w łańcuch (max 12 oktaw)
- Biblioteka MIDI dostępna na zasadzie licencji GNU LGPL

Biblioteka MIDI w języku C

Prawdę powiedziawszy, „biblioteka” to w przypadku MIDI dość duże słowo. Obsługa tego interfejsu jest bowiem prosta, a potrzebnych funkcji jest w rzeczywistości niewiele (w przeciwieństwie do ilości wymaganych definicji). Tak, czy inaczej, opracowanie biblioteki to kilka godzin spędzonych przy klawiaturze, a oszczędności czasu przecież nigdy zbyt wiele.

Przedstawianą dalej bibliotekę dedykowano mikrokontrolerom AVR i stworzono pod kątem wykorzystania z darmowym kompilatorem AVRGCC. Kod programu (w szczególności nazwy funkcji i definicji) napisano w języku angielskim, by biblioteka współgrała ze specyfikacją MIDI i by udostępnić ją wszystkim zainteresowanym na zasadach licencji GNU LGPL. Potrzebne pliki zamieszczono w Sieci [1].

Zagospodarowanie zasobów

Założono, że obsługę interfejsu MIDI będzie sprawował wbudowany w mikrokontroler AVR moduł USART (lub UART), który doskonale nadaje się do tego celu. Poza modułem sprzętowym, biblioteka zagospodarowuje pewną przestrzeń pamięci danych na bufor odbiorczy i nadawczy (o określonych przez programistę wielkościach) oraz wskaźniki globalne (3 zmienne bajtowe).

Parametry biblioteki

Właściwości biblioteki można zmienić, definiując w programie głównym kilka parametrów. Możliwe definicje zebrano w **tab. 1**.

Sposób buforowania

Obsługa interfejsu MIDI zachodzi w sposób buforowany, z zastosowaniem tzw. bufora cyklicznego (jego objętość można przy tym dobrać według potrzeb – patrz **tab. 1**). Funkcje nadające komunikaty korzystają z podfunkcji zapisującej bufor nadawczy, natomiast funkcje odbierające z podfunkcji odczytującej bufor odbiorczy.

W przypadku przepełnienia bufora nadawczego wykonywanie programu wpisu do bufora zostaje wstrzymane do momentu faktycznego przesłania kolejnego bajtu (dane nie są gubione). W przypadku przepełnienia bufora odbiorczego następuje natomiast ładowanie zmiennej `midi_in_status` wartością 2 (OVERRUN), fakt wykrycia błędu ramki jest sygnalizowany wartością 1 (ERROR), a normalnie zmienna ta przyjmuje wartość zerową. W razie wystąpienia jednego z wymienionych błędów dane nadchodzące są tracone, a zmienna statusu powinna być zerowana w programie głównym.

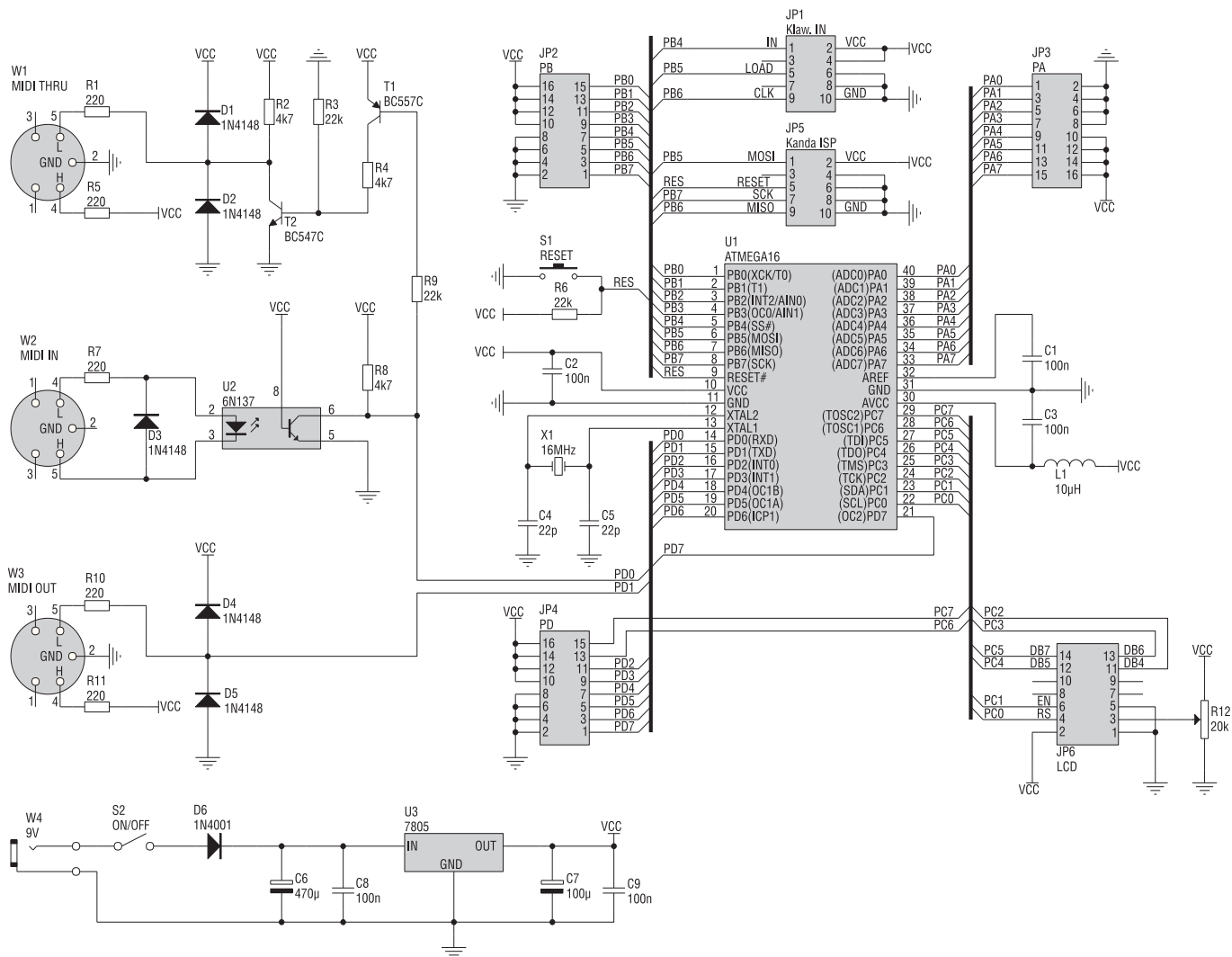
Definicje biblioteczne

Aby uczynić programy nieco bardziej czytelnymi, zamiast posługiwać się liczbami symbolizującymi bajty statusu, nuty, sterowniki itp., można korzystać z gotowych mnemoników. Zbiór zdefiniowanych etykiet przedstawiono w **tab. 2**. Mogą być one wykorzystywane jako parametry dalej opisywanych funkcji bibliotecznych.

Funkcje biblioteczne

Dostępne funkcje wraz z krótkimi opisami przedstawiono w **tab. 3**. Dokładniejszych informacji na temat ich praktycznego znaczenia należy szukać





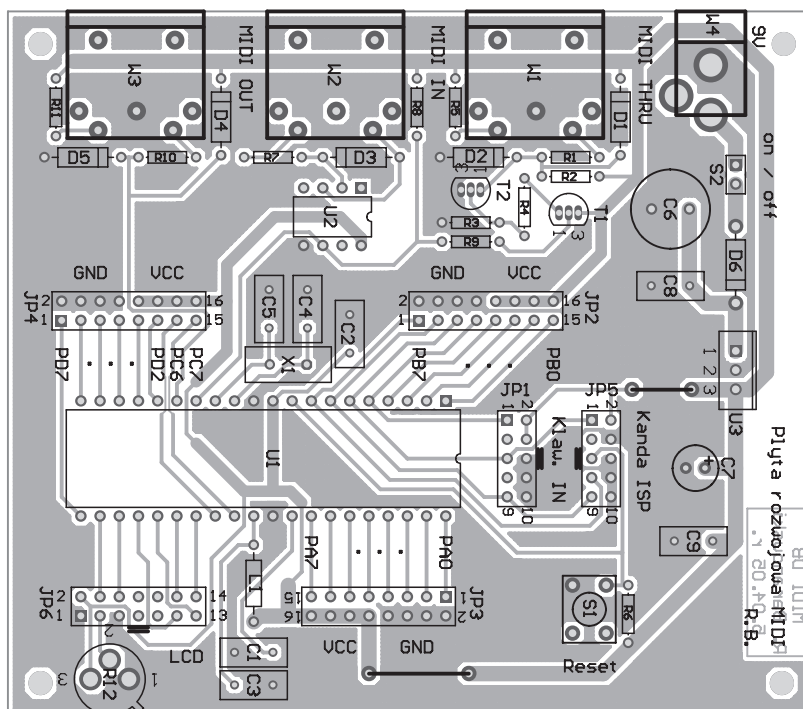
Rys. 1. Schemat płyty ewaluacyjnej MIDI

w kursie MIDI zamieszczonym w EP11/2005, EP12/2005 i EP1/2006.

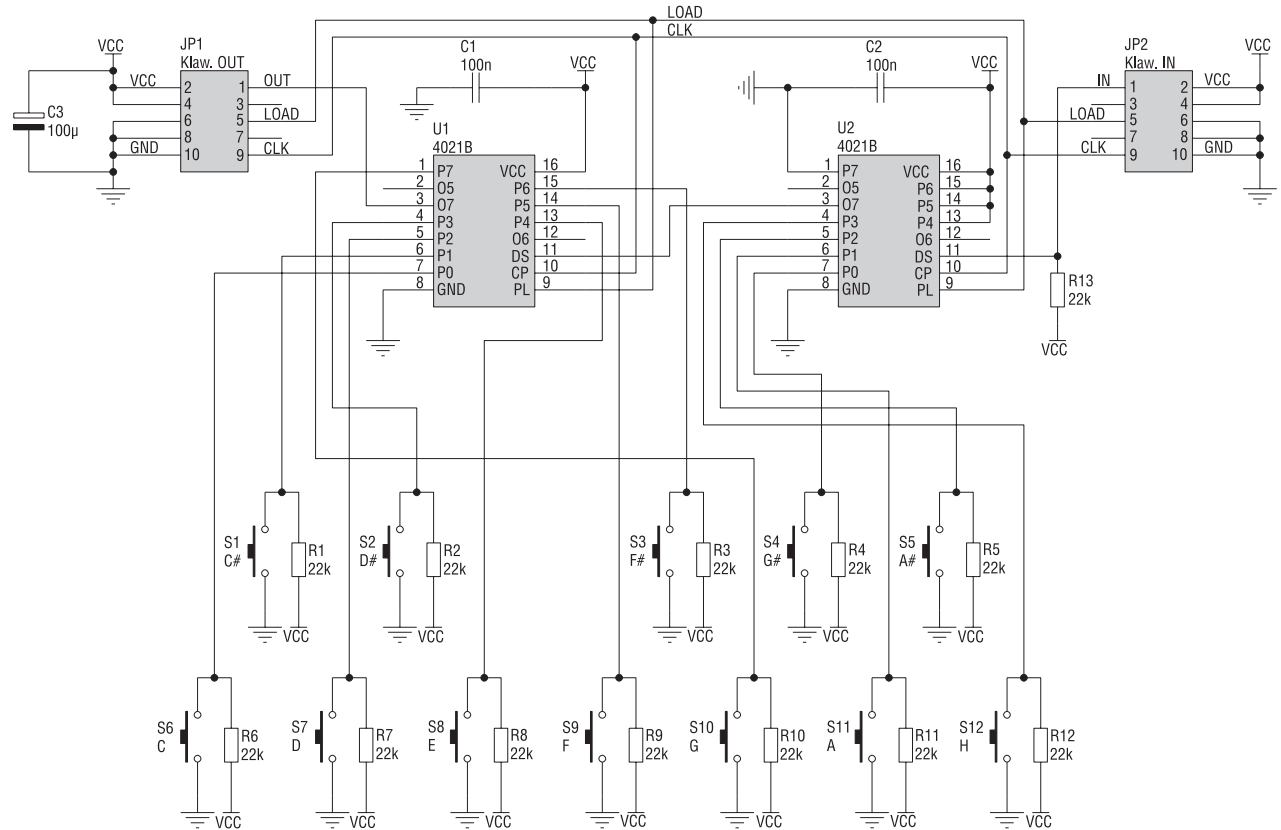
Należy zauważyć, że większość przedstawionych funkcji stanowią w rzeczywistości makra, zastępowane w procesie kompilacji zestawem odwołań do funkcji zapisującej bufor nadawczy. W przypadku częstego korzystania z danego makra warto przekształcić je w funkcję, w celu zmniejszenia wielkości programu wynikowego.

Budowa płyty rozwojowej MIDI

Na chwilę zapomnijmy o kwestii oprogramowania i przyjrzymy się stronie sprzętowej. Uniwersalny układ interfejsowy MIDI powinien posiadać zestaw trzech gniazd DIN i optoizolowane układy polaryzacji. Jeśli dodożymy do tego mikrokontroler programowany w systemie i zaopatrzymy go w zestaw złączy ogólnego przeznaczenia, otrzymamy płytę rozwojową, której schemat przedstawiono na **rys. 1**.



Rys. 2. Obwód drukowany płyty ewaluacyjnej MIDI



Rys. 3. Schemat elektryczny klawiatury

Układ został wyposażony w mikrokontroler AVR ATmega16. Jego najważniejszymi cechami są: wbudowany moduł USART, przetwornik A/C, 32 linie portów ogólnego przeznaczenia, możliwość pracy z sygnałem zegarowym o częstotliwości do 16 MHz i interfejs programowania ISP. Porty wejścia-wyjścia tego układu zostały wyprowadzone na złącza szpilkowe, natomiast linie modułu USART przyporządkowano „na stałe” interfejsowi MIDI.

Poza złączami MIDI DIN i wyprowadzeniami portów, płyta rozwojowa została wyposażona w złącze programowania ISP (dla programatorów zgodnych z STK200/300 firmy Kanda), wyjście wyświetlacza LCD (rozkład wyprowadzeń jak dla standardowych wyświetlaczy zgodnych z HD44780) i złącze klawiatury. Za pośrednictwem tego ostatniego można dołączyć klawiaturę muzyczną, opisywaną w dalszej części artykułu.

Wygląd obwodu drukowanego przedstawiono na rys. 2. Kwestia montażu jest tutaj oczywista, jednak należy zwrócić uwagę na to, aby dla wyjść LCD (JP6), ISP (JP5) i klawiatury (JP1) zastosować odpowiednie złącza wannowe (z wcięciem) i wlotować je w odpowiedniej orientacji.

Napięcie zasilania powinno być niemniejsze niż 8 V. Użytkowanie płyty ułatwiają opisy umieszczone na wierzchniej warstwie. Zaznaczono m.in. przeznaczenie poszczególnych złączy, numery portów procesora (przy JP2, JP3, JP4 i JP6) i opisano linie zasilające.

Zakres zastosowania tego układu jest dość szeroki – można wykorzystać go zarówno w celach edukacyjnych, jak i jako bazę dowolnego urządzenia MIDI. Układ może być

przydatny do budowy urządzeń interfejsowych (kierunkowanie wiadomości i ich translacja), układów sterujących (automatyzacja przedstawień), instrumentów muzycznych i syntezatorów. W ostatniej części tego artykułu zostanie zamieszczona prosta aplikacja klawiatury muzycznej MIDI.

Budowa płyty klawiatury

W celu rozszerzenia funkcjonalności płyty rozwojowej opracowano też płytę klawiatury, której schemat przedstawiono na rys. 3. Pojedyncza płyta odpowiada tylko jednej oktawie muzycznej (12 półtonów), lecz można łączyć ją w łańcuch z kolejnymi, identycznymi modułami. Złącze początku utworzonego w ten sposób łańcucha można doprowadzić do płyty rozwojowej, uzyskując klawiaturę muzyczną o maksymalnie 12 oktavach.

Urządzenie zostało zbudowane w oparciu o rejestry przesuwające typu 4021B (U1 i U2). Do równoległych wejść tych układów doprowadzono klawisze zwierające do masy, a podciągane do napięcia zasilania. Przy tworzeniu łańcucha modułów rejestry są łączone szeregowo, tworząc jeden długi rejestr przesuwający. Jego ładowanie zachodzi przy dodatnim impulsie sygnału LOAD, a przesuwanie taktuje zbocze narastające sygnału CLK.

Tab. 1. Definicje parametrów wykorzystywanych przez biblioteczne funkcje MIDI

Parametr	Znaczenie	Wartość domyślna	Zakres wartości
F_CPU	Częstotliwość taktująca mikrokontroler [Hz]	16000000UL (16 MHz)	
USART_NO	Numer przyporządkowanego interfejsowi MIDI modułu USART (definiowany, jeśli mikrokontroler posiada więcej niż jeden moduł tego typu)	0	0...3
MIDI_BUF_OUT_LENGTH	Wielkość bufora nadawczego [B]	50	1...255
MIDI_BUF_IN_LENGTH	Wielkość bufora odbiorczego [B]	50	1...255

Tab. 2. Etykiety zdefiniowane w bibliotece MIDI

Dotyczy	Etykieta	Wartość (hex)
wiadomości dźwiękowych	STATUS_NOTE_OFF	0x80
	STATUS_NOTE_ON	0x90
	STATUS_AFTERTOUCH	0xA0
	STATUS_CTL_CHG	0xB0
	STATUS_PROG_CHG	0xC0
	STATUS_CHAN_AFTERTOUCH	0xD0
wiadomości zmiany trybu	STATUS_PITCH_CHG	0xE0
	STATUS_MODE_CHG	0xB0
	CTL_ALL_SOUND_OFF	0x78
	CTL_RESET_ALL_CTL	0x79
	CTL_LOCAL_CTL	0x7A
	CTL_ALL_NOTES_OFF	0x7B
	CTL_OMNI_OFF	0x7C
	CTL_OMNI_ON	0x7D
wiadomości systemowych	CTL_MONO_ON	0x7E
	CTL_POLY_ON	0x7F
	STATUS_SYSEX_START	0xF0
	STATUS_QUARTER_FRAME	0xF1
	STATUS_SONG_PTR	0xF2
	STATUS_SONG_SELECT	0xF3
wiadomości czasu rzeczywistego	STATUS_SONG_SELECT	0xF3
	STATUS_TUNE_REQ	0xF6
	STATUS_SYSEX_END	0xF7
	STATUS_TIMING_CLOCK	0xF8
	STATUS_TIMING_TICK	0xF9
	STATUS_START	0xFA
numerów sterowników	STATUS_CONTINUE	0xFB
	STATUS_STOP	0xFC
	STATUS_ACT_SENS	0xFE
	STATUS_RESET	0xFF
	CTL_MSB_BANK_SEL	0x00
	CTL_MSB_MODUL_WHEEL	0x01
	CTL_MSB_BREATH	0x02
	CTL_MSB_FOOT	0x04
	CTL_MSB_PORTAMENTO_TIME	0x05
	CTL_MSB_DATA_ENTRY	0x06
	CTL_MSB_CHANNEL_VOL	0x07
	CTL_MSB_BALANCE	0x08
	CTL_MSB_PAN	0x0A
	CTL_MSB_EXPRESSION	0x0B
	CTL_MSB_EFFECT_1	0x0C
	CTL_MSB_EFFECT_2	0x0D
	CTL_MSB_GP_1	0x10
	CTL_MSB_GP_2	0x11
	CTL_MSB_GP_3	0x12
	CTL_MSB_GP_4	0x13
	CTL_LSB_BANK_SEL	0x20
	CTL_LSB_MODUL_WHEEL	0x21
	CTL_LSB_BREATH	0x22
	CTL_LSB_FOOT	0x24
	CTL_LSB_PORTAMENTO_TIME	0x25
	CTL_LSB_DATA_ENTRY	0x26
	CTL_LSB_CHANNEL_VOL	0x27
	CTL_LSB_BALANCE	0x28
	CTL_LSB_PAN	0x2A
	CTL_LSB_EXPRESSION	0x2B
	CTL_LSB_EFFECT_1	0x2C
	CTL_LSB_EFFECT_2	0x2D
	CTL_LSB_GP_1	0x30
CTL_LSB_GP_2	0x31	
CTL_LSB_GP_3	0x32	
CTL_LSB_GP_4	0x33	

Tab. 2. cd

Dotyczy	Etykieta	Wartość (hex)
numerów sterowników	CTL_SUSTAIN_ON_OFF	0x40
	CTL_PORTAMENTO_ON_OFF	0x41
	CTL_SUSTENUTO_ON_OFF	0x42
	CTL_SOFT_ON_OFF	0x43
	CTL_LEGATO_ON_OFF	0x44
	CTL_HOLD_2_ON_OFF	0x45
	CTL_SOUND_1	0x46
	CTL_SOUND_2	0x47
	CTL_SOUND_3	0x48
	CTL_SOUND_4	0x49
	CTL_SOUND_5	0x4A
	CTL_SOUND_6	0x4B
	CTL_SOUND_7	0x4C
	CTL_SOUND_8	0x4D
	CTL_SOUND_9	0x4E
	CTL_SOUND_10	0x4F
	CTL_GP_5	0x50
	CTL_GP_6	0x51
	CTL_GP_7	0x52
	CTL_GP_8	0x53
	CTL_PORTAMENTO	0x54
	CTL_EFFECT_1	0x5B
	CTL_EFFECT_2	0x5C
	CTL_EFFECT_3	0x5D
	CTL_EFFECT_4	0x5E
	CTL_EFFECT_5	0x5F
	CTL_DATA_INC	0x60
	CTL_DATA_DEC	0x61
	CTL_LSB_NRPN	0x62
	CTL_MSB_NRPN	0x63
	CTL_LSB_RPN	0x64
	CTL_MSB_RPN	0x65
	numerów sterowników RPN	RPN_MSB_PITCH_BEND_SENS
RPN_LSB_PITCH_BEND_SENS		0x00
RPN_MSB_CHANNEL_FINE_TUNE		0x00
RPN_LSB_CHANNEL_FINE_TUNE		0x01
RPN_MSB_CHANNEL_COARSE_TUNE		0x00
RPN_LSB_CHANNEL_COARSE_TUNE		0x02
RPN_MSB_TUNING_PROG_CHG		0x00
RPN_LSB_TUNING_PROG_CHG		0x03
RPN_MSB_TUNING_BANK_SEL		0x00
RPN_LSB_TUNING_BANK_SEL		0x04
RPN_MSB_MODULATION_RANGE		0x00
RPN_LSB_MODULATION_RANGE		0x05
RPN_MSB_NULL		0x7F
RPN_LSB_NULL	0x7F	
przykłady nut	N_F0 (nuta F oktawy 0)	0x11
	N_GH1 (nuta G# oktawy 1)	0x20

Obwód drukowany przedstawiono na **rys. 4**. Został on zaprojektowany na podobieństwo (z zachowaniem wymiarów) typowej klawiatury muzycznej, aby mógł posłużyć za bazę urządzenia tego

typu lub zastąpić obwody sprzętu istniejącego. Również w tym przypadku montaż nie pozostawia wątpliwości, lecz warto zwrócić uwagę na konieczność stosowania kątowych złączy wannowych (JP1 i JP2), a w przypadku złączy prostych – uważać na ich orientację.

Choć w urządzeniu modelowym w roli klawiszy zastosowano typowe mikroprzyciski, sens ich stosowania jest w tym przypadku dość wątpliwy.

Tab. 3. Funkcje i makra biblioteki MIDI

Dotyczy	Prototyp funkcji lub deklaracja makra	Komentarz
inicjalizacji	<code>void midi_init(void);</code>	inicjalizuje interfejs MIDI
obsługi buforów	<code>void midi_buf_write(uint8_t data);</code>	zapisuje bufor nadawczy
	<code>uint8_t midi_buf_read(uint8_t *data);</code>	odczytuje bufor odbiorczy (zwraca 0, jeśli bufor jest pusty; zwraca 1 w przypadku powodzenia)
wiadomości dźwiękowych	<code>#define midi_note_off(channel, note, velocity)</code>	generuje dźwięk
	<code>#define midi_note_on(channel, note, velocity)</code>	wygasza dźwięk
	<code>#define midi_aftertouch(channel, note, pressure)</code>	moduluje dźwięk
	<code>#define midi_ctl_chg(channel, controller, value)</code>	zmienia wartość sterownika
	<code>#define midi_prog_chg(channel, program)</code>	zmienia brzmienie
	<code>#define midi_chan_aftertouch(channel, pressure)</code>	moduluje kanał
	<code>#define midi_pitch_chg_word(channel, pitch_chg_word)</code>	zmienia ton kanału
wiadomości zmiany trybu	<code>#define midi_all_sound_off(channel)</code>	wyłącza wybrzmiewające dźwięki
	<code>#define midi_reset_all_ctl(channel)</code>	zeruje wszystkie sterowniki
	<code>#define midi_local_ctl_off(channel)</code>	wyłącza syntezytor lokalny
	<code>#define midi_local_ctl_on(channel)</code>	włącza syntezytor lokalny
	<code>#define midi_all_notes_off(channel)</code>	wygasza wybrzmiewające dźwięki
	<code>#define midi_mode_1(channel)</code>	wybierają jeden z czterech dostępnych trybów MIDI
	<code>#define midi_mode_2(channel)</code>	
	<code>#define midi_mode_3(channel)</code>	
	<code>#define midi_mode_4(channel, number_of_channels)</code>	
wiadomości systemowych	<code>void midi_sysex(uint8_t id, uint8_t sysex_channel, uint8_t *data, uint8_t length);</code>	nadaje wiadomość specjalną
	<code>#define midi_quarter_frame(msg_number, SMPTE_part)</code>	nadaje komunikat ćwiartki ramki (SMPTE)
	<code>#define midi_song_ptr_word(song_ptr_word)</code>	przesuwa miejsce odtwarzania utworu
	<code>#define midi_song_select(song_number)</code>	wybiera utwór
	<code>#define midi_tune_req()</code>	żąda dostrojenia
wiadomości czasu rzeczywistego	<code>#define midi_timing_clock()</code>	generuje wiadomość zegara muzycznego
	<code>#define midi_timing_tick()</code>	generuje wiadomość zegara absolutnego
	<code>#define midi_start()</code>	uruchamia odtwarzanie utworu
	<code>#define midi_continue()</code>	kontynuuje odtwarzanie utworu
	<code>#define midi_stop()</code>	przerwa odtwarzanie utworu
	<code>#define midi_act_sens()</code>	nadaje wiadomość aktywnego połączenia
	<code>#define midi_reset()</code>	zeruje dołączone urządzenia
	sterowników	<code>#define midi_ctl_chg_word(channel, controller_msb, value_word)</code>
<code>#define midi_ctl_on(channel, controller)</code>		włącza sterownik dwupołożeniowy
<code>#define midi_ctl_off(channel, controller)</code>		wyłącza sterownik dwupołożeniowy
sterowników (N)RPN	<code>#define midi_rpn_chg_word(channel, rpn_msb, rpn_lsb, value_word)</code>	zmienia 2-bajtową wartość sterownika RPN
	<code>#define midi_rpn_inc(channel, rpn_msb, rpn_lsb)</code>	zwiększa wartość sterownika (N)RPN
	<code>#define midi_rpn_dec(channel, rpn_msb, rpn_lsb)</code>	zmniejsza wartość sterownika (N)RPN
	<code>#define midi_nrpn_chg_word(channel, nrpn_msb, nrpn_lsb, value_word)</code>	zmienia 2-bajtową wartość sterownika NRPN

Tab. 4. Sposób programowania bitów konfiguracyjnych mikrokontrolera U1 płyty rozwojowej

OCDEN	JTAGEN	SPIEN	CKOPT	EESAVE	BOOTSZ1	BOOTSZ0	BOOTRST
		*	*		*	*	
BODLEVEL	BODEN	SUT1	SUT0	CKSEL3	CKSEL2	CKSEL1	CKSEL0
*	*	*					

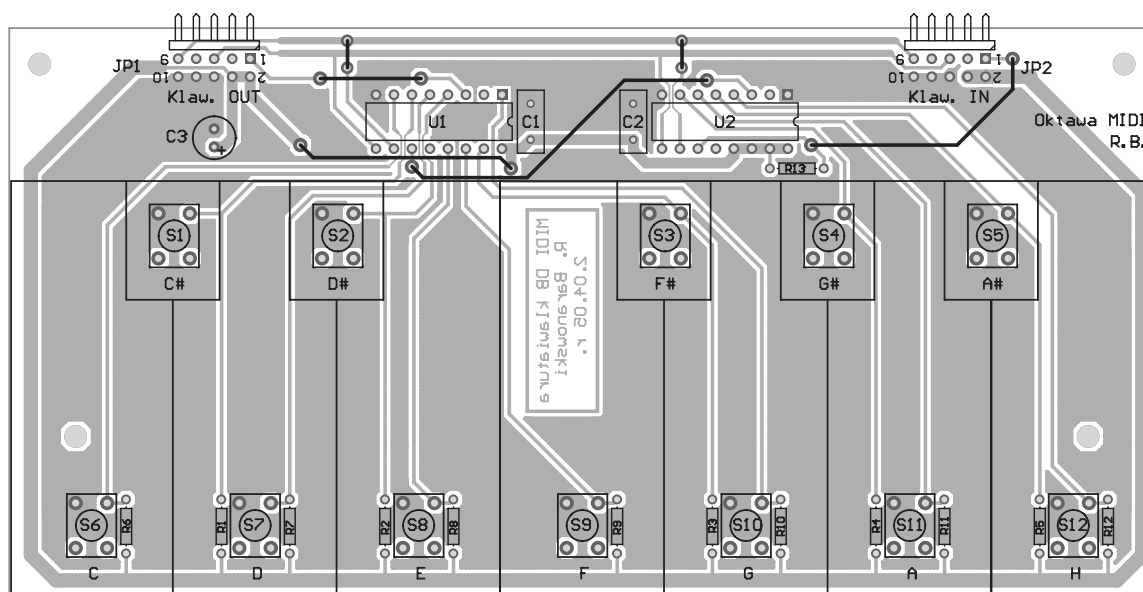
* – bit zaprogramowany (o zerowej wartości logicznej)

Otóż elementy tego typu mają zbyt mały skok, są przy tym zbyt „twarde”, a na domiar złego generują charakterystyczny „klik”, który jest tutaj niedopuszczalny. W praktyce należałoby zastosować bezgłośne, miękkie przyciski o większym skoku, lub wykonać je ze sprężystych blaszek, wlotowywanych w płytę.

Układ dołącza się do płyty rozwojowej MIDI poprzez złącze JP1 pierwszej klawiatury łańcucha, które należy doprowadzić do złącza JP1 płyty rozwojowej, z której układ jest zasilany. Połączenia mogą być wykonywane za pośrednictwem 10-żyłowej taśmy AWG28, zakończonej wtykami zaciskowymi.

Przykładowa aplikacja

W ostatniej części artykułu pokazemy na prostym przykładzie sposób wykorzystania omawianej wcześniej biblioteki. Jak już wspomniano, łącząc łańcuch klawiatur z płytą rozwojową można stworzyć prostą klawiaturę muzyczną MIDI. Oczywiście, wymagany jest jeszcze program sterujący pra-



Rys. 4. Obwód drukowany płytki klawiatury

cają mikrokontrolera, lecz wykorzystując język C z zastosowaniem gotowej biblioteki MIDI jest to kwestia dosłownie kilku chwil.

Zakładając, że poza biblioteką MIDI posiadamy również podprogramy obsługujące klawiaturę, cały program mógłby wyglądać tak, jak przedstawiono na **list. 1**. Jego zadaniem jest porównywanie aktualnego stanu klawiatury ze stanem sprzed chwili. W razie braku zgodności określa się, który klawisz zmienił swoje położenie i przesyła odpowiedni komunikat dźwiękowy (generacji dźwięku lub jego wygaszenia).

Na **list. 2** przedstawiono dwie funkcje ułatwiające obsługę łańcucha klawiatur. Funkcja inicjalizująca (`keyb_init`) dokonuje nastawy kierunku portów wejścia-wyjścia i określa liczbę dołączonych modułów klawiatury. Wartość ta jest zapisywana w zmiennej globalnej `keyb_len`. Funkcja sprawdzająca stan klawiatury (`keyb_check`) odświeża natomiast tablicę aktualnego stanu (`keyb_state[12]`). Każdemu bitowi tej 12-elementowej tablicy bajtów odpowiada pojedynczy klawisz klawiatury. Jeśli dany klawisz jest wciśnięty (puszczony), przyporządkowany mu bit jednego z elementów tablicy jest ustawiony (wyzerowany).

Przedstawiony program został dołączony jako przykład do omawianej wcześniej biblioteki MIDI i można go odnaleźć pod adresem [1]. Załączono zarówno pliki źródłowe, jak i odpowiedni `makefile` oraz plik wynikowy. Przy programowaniu mikrokontrolera należy zaprogramować bity sterujące według **tab. 4**.

I co dalej?

Aplikacja przykładowa, tj. program, biblioteka oraz płyta rozwojowa wraz z klawiaturą mogą posłużyć za bazę do budowy pełnowartościowej klawiatury muzycznej MIDI. Duża liczba portów zastosowanego mikrokontrolera pozwala stworzyć rozbudowany interfejs użytkownika. Ponieważ mikrokontroler jest wyposażony w wielokanałowy przetwornik C/A, można pokusić się o jego wykorzystanie do badania ciśnienia wywieranego przez muzyka na klawiaturę za pośrednictwem czujników tensometrycznych (w celu implementacji wiadomości modulacji

kanalu – *channel aftertouch*). Jednak na zagadnienie można spojrzeć z innej strony – przecież płyta rozwojowa może równie dobrze służyć do budowy syntezatora czy elementu automatyki. Pomysłów i możliwości można byłoby mnożyć wiele – ich realizację pozostawia się Czytelnikowi.

Rafał Baranowski, EP
Rafal.Baranowski@ep.com.pl

Adresy internetowe:

[1] <http://www.polsl.gliwice.pl/~rufus/midi.zip> – biblioteka MIDI dla AVRGCC

WYKAZ ELEMENTÓW

Płyta rozwojowa MIDI

Rezystory

R1, R5, R7, R10, R11: 220 Ω/0,125 W
 R2, R4, R8: 4,7 kΩ/0,125 W
 R3, R6, R9: 22 kΩ/0,125 W
 R12: potencjometr montażowy stojący 20 kΩ

Kondensatory

C1...C3, C8, C9: 100 nF
 C4, C5: 22 pF
 C6: 470 µF/25 V
 C7: 100 µF/25 V

Półprzewodniki

D1...D5: 1N4148
 D6: 1N4001
 T1: BC557C
 T2: BC547C
 U1: ATmega16, DIP40
 U2: transoptor 6N137
 U3: 7805

Inne

W1...W3: gniazdo DIN5 kątowe do druku
 W4: gniazdo zasilające do druku

JP1, JP5: gniazdo proste 2x5 pin do druku

JP2...JP4: gniazdo proste 2x8 pin do druku

JP6: gniazdo proste 2x7 pin do druku

S1: microswitch

S2: włącznik zasilania

L1: dławik 10 µH

X1: kwarc 16 MHz

Płyta klawiatury

(pojedyncza oktawa)

Rezystory

R1...R13: 22 kΩ/0,125 W

Kondensatory

C1, C2: 100 nF
 C3: 100 µF/25 V

Półprzewodniki

U1, U2: 4021B, DIP-16

Inne

JP1, JP2: gniazda kątowe 2x5 pin do druku + wtyki zaciskane na taśmę + taśma 10 żyłowa AWG28

S1...S12: klawisz muzyczny (ew. microswitch)