

# Cyfrowa gitara MIDI, część 1

PROJEKT  
Z OKŁADKI



*Elektronika od początku swojego istnienia zajmowała się obróbką i przesyłaniem dźwięku. Pierwotnie wszystkie zagadnienia - transmisja dźwięku, jego wzmacnianie, nagrywanie i odtwarzanie - były realizowane metodami analogowymi. Czas pokazał, że rola odtwórcza nie stanowi jedynej możliwości, jaką daje elektronika. Druga połowa zeszłego wieku to okres rozwoju twórczej roli urządzeń elektronicznych i muzyki elektronicznej - przez jednych pogardzanej, przez drugich uwielbianej, a w każdym razie stale się rozwijającej.*

**Rekomendacje:** projekt polecamy wszystkim miłośnikom zastosowań elektroniki w muzyce, w szczególności tym, którzy lubią eksperymentować. Na przedstawionym w artykule instrumencie można wydobywać dźwięki jak z prawdziwej gitary, choć nie posiada ona prawdziwych strun.

Właściwie chyba nikt już dziś nie wątpi w jakość cyfrowego zapisu muzycznego czy cyfrowych odtwarzaczy i wzmacniaczy audio (np. zapis CD, wzmacniacze klasy D). Inaczej ma się jednak kwestia twórcza urządzeń cyfrowych - instrumenty elektroniczne jak dotąd nie zastąpiły w pełni żadnego instrumentu klasycznego, znalazło się przy tym względnie niewielu artystów, którzy zajmują się tworzeniem muzyki w pełni elektronicznej. Cóż, to dość zniechęcające dla nas, elektroników. Bynajmniej nie wszystko stracone i chyba zawsze warto próbować - przecież nic nie trwa wiecznie!

## Historii tylko trochę

Przedstawiając Czytelnikom projekt w pełni elektronicznego instrumentu, nie sposób nie zacząć od opisanego tego, co nasz instrument będzie tworzył. Tak też zrobimy, bowiem dźwięk wydobywający się z każdego instrumentu, czy to jako drgania struny przenoszone przez powietrze w postaci fal akustycznych, czy w postaci zerojedynkowych pulsacji prądu w kablu sygnałowym, jest sprawą nadrzędną. Będziemy mieli ułatwione zadanie, gdyż praktycznie mamy jeden standard, w którym „rozmawiają” ze sobą cyfrowe instrumenty muzyczne. To dość osobliwe, szczególnie jeśli spojrzeć na elektronikę w ogóle - wszędzie mnóstwo rozwiązań, często zupełnie niekompatybilnych, a tutaj jedno? Tak się

złożyło, że dawno temu w tej kwestii doszło do porozumienia między kilkoma większymi firmami zajmującymi się produkcją urządzeń muzycznych. Powołano specjalne stowarzyszenie i zaprowadzono porządek. Nie znaczy to jednak, że nasz standard był zawsze wolny od wszelkich nieścisłości.

Standard MIDI (*The Musical Instrument Digital Interface* - cyfrowy interfejs instrumentów muzycznych), bo o nim mowa, ma już swoje lata i jest w wieku niektórych z nas. Początek MIDI datuje się na 1983r., kiedy to projektanci z różnych firm rozpoczęli współpracę, której wynikiem była prezentacja gotowych urządzeń mających możliwość komunikowania się między sobą. Bardzo szybko standard ten zyskał sobie uznanie, bowiem w końcu urządzenia stawały się ze sobą kompatybilne i można było je łączyć nawet wówczas, kiedy były wyprodukowane przez różne firmy. Standaryzacji zostały poddane złącza, typ interfejsu, protokół transmisji i znaczenie poszczególnych kodów i sekwencji. Okazało się jednak, że pierwotnie nie wszystko zostało dostatecznie określone, dlatego w roku 1991

**Tab. 1. Ważniejsze bajty informacyjne i następujące po nich bajty danych**

Wiadomość	Bajt informacyjny	I bajt danych	II bajt danych
Włączenie nuty (Note On Event)	1001kkkk k - kanał	0aaaaaaa a - nuta*	0bbbbbbb b - siła
Wyłączenie nuty (Note Off Event)	1000kkkk k - kanał	0aaaaaaa a - nuta	0bbbbbbb b - siła
Zmiana instrumentu (Program Change)	1100kkkk k - kanał	0aaaaaaa a - instrument	[brak]
Zagięcie tonu (Pitch Bend)	1110kkkk k - kanał	0aaaaaaa a - zagięcie [LSB]	0bbbbbbb b - zagięcie [MSB]
Zmiana ustawień (Control Change)	1011kkkk k - kanał	0aaaaaaa a - ustawienie**	0bbbbbbb b - wartość
Zmiana trybu pracy (Channel Mode)	1011kkkk k - kanał	0aaaaaaa a - tryb***	0bbbbbbb b - wartość

\* - patrz tabela 2

\*\* - patrz tabela 3

\*\*\* - patrz tabela 4

wprowadzono rozszerzenie w postaci General MIDI, a następnie, w 1999 - General MIDI II. W tym miejscu warto zaznaczyć jedną sprawę. Skrót MIDI przez większość użytkowników domowych komputerów kojarzona jest z plikami dźwiękowymi o formacie MID lub SMF (*Standard Midi File*), przy czym z powodu słabej jakości generowanego dźwięku pliki tego typu są zazwyczaj pogardzane. Jeśli tak myślisz - odrzuć to zdanie! MIDI jest standardem interfejsu muzycznego, a pliki o wspomnianym formacie są jedynie zapisem transmisji. Ich brzmienie zależy od osprzętu, który je interpretuje i nie ma nic wspólnego z „jakością MIDI”. Ciągły rozwój urządzeń syntetyzujących dźwięk powoduje, że cyfrowa interpretacja MIDI staje się coraz trudniej odróżnialna od dźwięków generowanych przez tradycyjne instrumenty, czego dowodem może być fakt, że coraz częściej tworzy się w ten sposób choćby podkłady muzyczne do filmów czy nawet elementy muzyki pop.

MIDI posiada jeszcze jeden ciekawy zakres zastosowań, który polega nie na generacji dźwięku, ale na sterowaniu urządzeniami oświetleniowymi i ogólnie scenografią. Wyobrazić sobie można przykładowo spektakl teatralny z podkładem muzycznym generowanym z pliku MIDI. Jeśli w pliku takim wprowadzimy dodatkową ścieżkę (wykorzystamy dodatkowy kanał, o którym później), na której będą przesyłane komendy dla urządzeń sterujących oświetleniem, ruchomymi częściami de-

koracji itp., to zautomatyzujemy całe przedstawienie! I nie jest to wcale jedynie nasz wymysł - tak robi się w praktyce.

### MIDI - muzyczne gadu-gadu

Omawiany standard ma w rzeczywistości postać prostego, szeregowego interfejsu asynchronicznego, opartego na pętli prądowej z optoizolacją. Działa on z prędkością 31,25kbps, a przesyłane słowo składa się z bitu startu, ośmiu bitów danych i bitu stopu, co w sumie trwa 320 mikrosekund. Jest to czas na tyle mały, że słuchacz nie słyszy niejednoczesności uderzeń w klawisze instrumentu, które zresztą nigdy nie są przecież jednoczesne. Jeśli wyobrazić sobie jednak skomplikowany utwór muzyczny z wieloma instrumentami grającymi współbieżnie, można pomyśleć, że prędkość transmisji może się okazać niewystarczająca. Będzie to po części prawda, z drugiej jednak strony, jak dotąd, w praktyce interfejs MIDI świetnie radzi sobie nawet

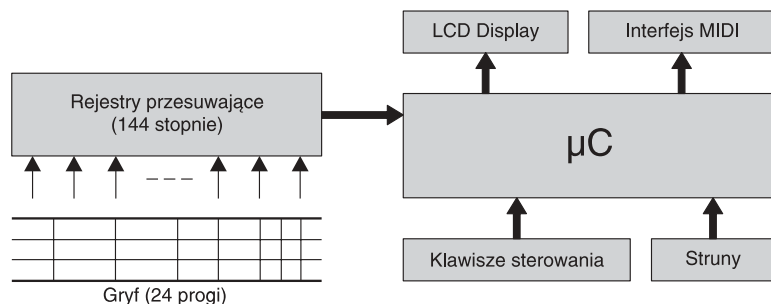
z kilkoma instrumentami działającymi wspólnie.

Przechodząc do meritum, instrumenty wyposażone w złącza MIDI posiadają najczęściej aż trzy gniazda, przy czym każde z nich ma różne przeznaczenie. Są to: MIDI IN, MIDI OUT i MIDI THRU. To ostatnie jest kopią MIDI IN i służy do łączenia instrumentów w łańcuch. Jak nazwa wskazuje, na wejściu IN urządzenie otrzymuje polecenia z zewnątrz - instrument może otrzymywać informacje na temat synchronizacji i ustawień, natomiast syntezator otrzymuje przez to złącze na przykład informacje o uderzonych w instrumencie klawiszach. OUT natomiast to w instrumencie wyjście zakodowanego zapisu muzycznego lub w przypadku urządzenia nadzorującego, przykładowo, źródło sygnałów synchronizujących.

Aspektem fizycznym interfejsu zajmiemy się w dalszej części artykułu, tymczasem przyjrzymy się protokołowi MIDI. Na transmisję składają się bloki danych o nieokreślonej długości. Początkowy bajt takiego bloku informuje o jego zawartości i o „kanale” (jest ich 16 możliwych), na którym jest on przesyłany (kanał jest oczywiście abstrakcją, zresztą bardzo wygodną i jak się później okaże, dającą wiele ciekawych możliwości). Najczęściej występuje sytuacja, kiedy w pierwszej kolejności przesyłany jest bajt informacyjny, a zaraz po nim jeden lub dwa bajty danych. Bajt informacyjny dostarcza wiedzy o kanale, na którym przesyłana jest informacja i na temat tego, czym jest treść następujących po nim danych. Warto byłoby dowiedzieć się, jak

**Tab. 2. Kody nut dla wiadomości dźwiękowych**

Numer oktawy	Kod nuty											
	C	C#	D	D#	E	F	F#	G	G#	A	A#	B
-1	0	1	2	3	4	5	6	7	8	9	10	11
0	12	13	14	15	16	17	18	19	20	21	22	23
1	24	25	26	27	28	29	30	31	32	33	34	35
2	36	37	38	39	40	41	42	43	44	45	46	47
3	48	49	50	51	52	53	54	55	56	57	58	59
4	60	61	62	63	64	65	66	67	68	69	70	71
5	72	73	74	75	76	77	78	79	80	81	82	83
6	84	85	86	87	88	89	90	91	92	93	94	95
7	96	97	98	99	100	101	102	103	104	105	106	107
8	108	109	110	111	112	113	114	115	116	117	118	119
9	120	121	122	123	124	125	126	127				



Rys. 1. Schemat blokowy gitary

nakazać syntezatorowi, posługując się jego złączem IN, żeby zagrał naszą ulubioną melodię. W zasadzie jest to bardzo proste, należy znać jedynie kilka słów-zakłęb, na które urządzenia MIDI są wyczułone. Słowa te, nazwane wcześniej bajtami informacyjnymi, podzielić można na: wiadomości dźwiękowe (*Channel Voice Message*), zmiany trybu pracy kanału (*Channel Mode*), wiadomości wspólne (systemowe) dla wszystkich kanałów (*System Common*), sygnały synchronizacji (*System Real Time*) i wiadomości zastrzeżone (*System Exclusive*). Po wszystkich tych słowach następuje transmisja bajtów danych, które

czasem mogą stanowić po prostu jakąś wartość, a czasem adres lub komendę. W dalszej części przyjrzymy się ramkom najważniejszym, które później wykorzystamy w budowanym przez nas instrumencie.

### Magiczne zakłęcia

Z praktycznego punktu widzenia, najważniejszymi bajtami informacyjnymi są wiadomości dźwiękowe. Za ich pomocą możemy zmusić dany syntezator do „odezwania się” - możemy bowiem zażądać zainicjowania (lub przerwania) gry wybranej nuty z daną siłą i odpowiednią głośnością. Możliwa jest także zmiana

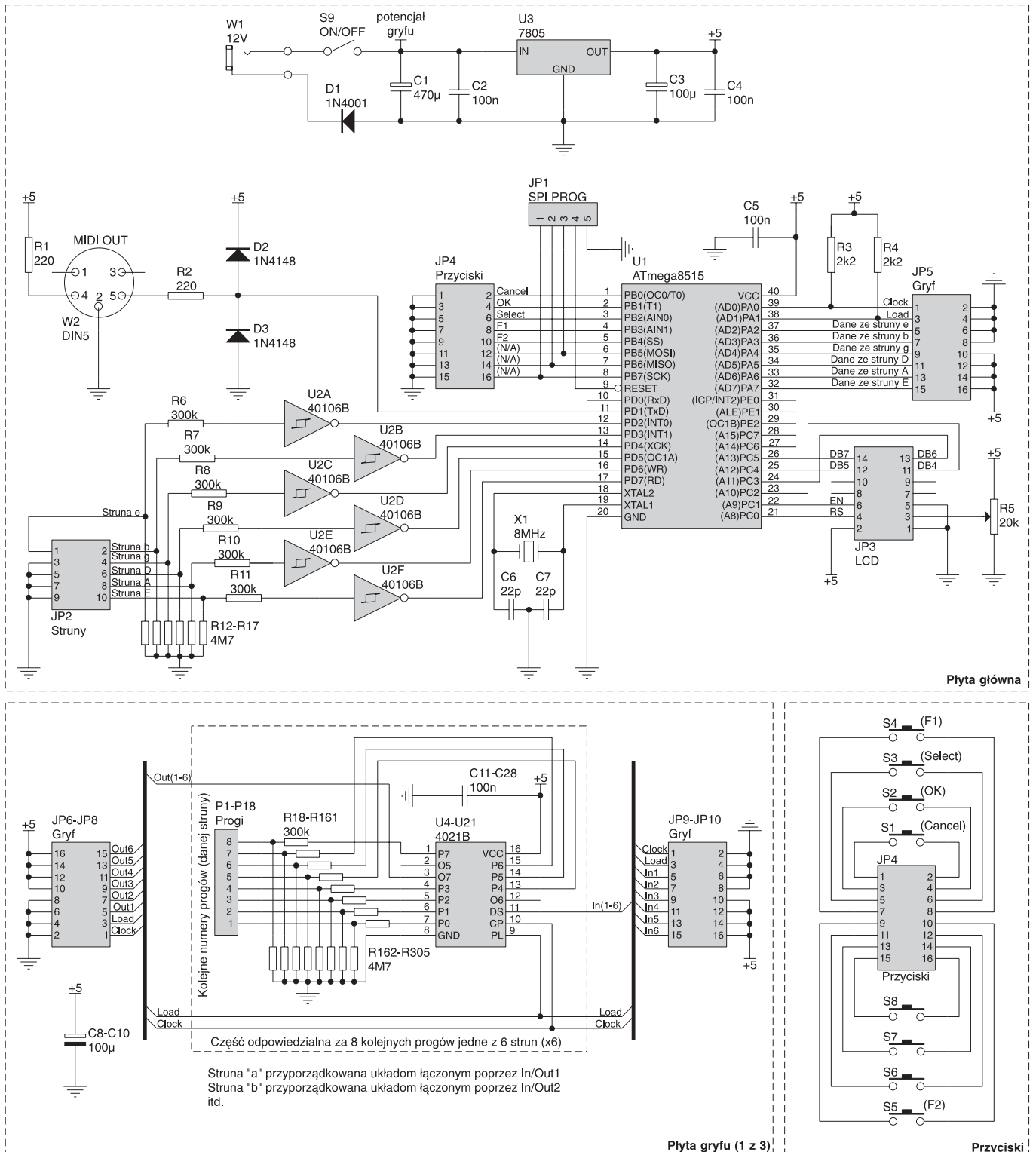
instrumentu przypisanego do danego kanału lub zagięcie tonu wybrzmiewającego już dźwięku (*Pitch Bend*). Do wiadomości dźwiękowych zalicza się również (choć jedynie formalnie) wiadomość zmiany ustawień (*Controller Change*). Za jej pomocą, adresując odpowiednie ustawienie, możemy zmienić bardzo wiele parametrów, od których zależy dźwiękowa interpretacja strumienia nut płynącego danym kanałem. Do ustawień takich zaliczyć można między innymi: głośność, balans, położenie „stopki”, numer banku dźwięków (z którego dany instrument wybiera się później odpowiednią wiadomością dźwiękową), położenie koła modulacji dźwięku i wiele innych. Istnieją także bardziej zaawansowane parametry, które należy najpierw zaadresować wiadomością wyboru parametru specjalnego, a dopiero później, za pomocą wiadomości wpisu ustawień, ustalić jego nową wartość. Ustawienia tego typu dzielone są na parametry zdefiniowane i niezdefiniowane (*Registered i Non-Registered Parameters*), przy czym te pierwsze wyróżniają się tym, że ich adresy i zastosowanie zostało określone w standardzie GM i powinny być one jednakowo interpretowane w urządzeniach wszystkich marek. Przykładem takiego specjalnego parametru zdefiniowanego jest np. zakres zaginania tonu dla wiadomości dźwiękowej - *Pitch Bend* (jego adresem są dwa bajty: 00H (LSB), 00H (MSB), natomiast wartość wpisywana to dwa, 7-bitowe bajty, z których bajt wyższy określa liczbę półtonów, natomiast niższy liczbę setnych części półtonów, w sumie określające bezwzględny zakres zagięcia tonu).

Tab. 3. Ważniejsze adresy ustawień i ich możliwe wartości dla wiadomości zmiany ustawień

Ustawienie	I bajt danych	II bajt danych	
Głośność kanału	7	0-127 [MSB]	głośność
	39	0-127 [LSB]	
Balans	8	0-127 [MSB]	balans
	40	0-127 [LSB]	
Bank dźwięków	0	0-127 [MSB]	numer banku
	32	0-127 [LSB]	
Oddech	2	0-127 [MSB]	wartość wdechu
	34	0-127 [LSB]	
Ekspresja	11	0-127 [MSB]	wartość ekspresji
	43	0-127 [LSB]	
Koło modulacji dźwięku ("pitch bend" całego kanału)	1	0-127 [MSB]	wartość przesunięcia
	33	0-127 [LSB]	
Pedał podtrzymania	64	0 - off, 127 - on	
Pedał tłumienia	67	0 - off, 127 - on	
Pedał legato	68	0 - off, 127 - on	
Portamento	65	0 - off, 127 - on	
Czas portamento (płynne przejście pomiędzy tonami)	5	0-127 [MSB]	długość czasu
	37	0-127 [LSB]	
Stopka	4	0-127 [MSB]	siła nacisku
	36	0-127 [LSB]	
Wybór parametru specjalnego (niezdefiniowanego)	98	0-127 [LSB]	adres parametru
	99	0-127 [MSB]	
Wybór parametru specjalnego (zdefiniowanego)	100	0-127 [LSB]	adres parametru
	101	0-127 [MSB]	
Wpis danych (dla parametrów specjalnych)	6	0-127 [MSB]	wartość wpisywana
	38	0-127 [LSB]	

Tab. 4. Ważniejsze tryby pracy dla wiadomości zmiany trybu pracy

Ustawienie	I bajt danych	II bajt danych
Wyłączenie granych dźwięków	120	0
Zresetowanie ustawień	121	0
Lokalne zarządzanie	122	0 - off 127 - on
Wygaszenia granych nut	123	0
Tryb OMNI Off	124	0
Tryb OMNI On	125	0
Tryb monofoniczny	126	liczba kanałów
Tryb polifoniczny	127	0

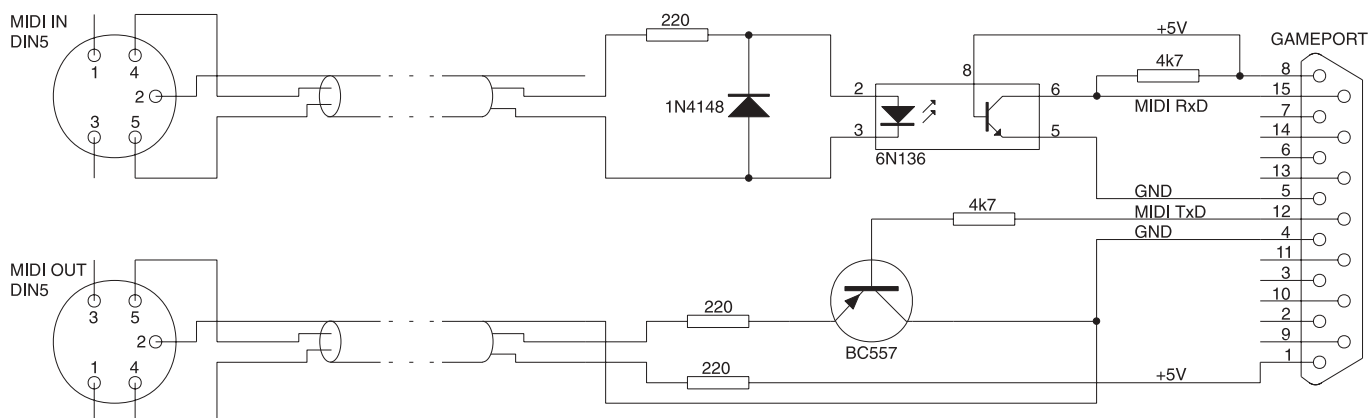


Rys. 2. Schemat ideowy instrumentu

Wiadomość określająca tryb pracy kanału jest pochodną wiadomości zmiany ustawień i pozwala między innymi ustawić dany kanał jako mono- lub polifoniczny, co oznacza, że w jednej chwili może być na nim grany jeden dźwięk (granie nowego zastępuje poprzedni) lub wiele dźwięków jedno-

nocześnie. Możliwe jest również wyciszenie wszystkich dźwięków wybrzmiewających na danym kanale i zerowanie ustawień. Zmiana trybu pracy urządzenia na OMNI powoduje natomiast, że syntezator odbiera dane ze wszystkich kanałów jednocześnie. Funkcja ta została zaimplementowana w obawie

przed sytuacją, w której potencjalny klient w czasie wyboru urządzenia w sklepie mógłby zrazić się ewentualnymi problemami przyłączenia instrumentu z urządzeniem zewnętrznym. W rzeczywistości tryb ten nie jest praktycznie wykorzystywany i zazwyczaj przy-sparza samych kłopotów.



Rys. 3. Schemat konwertera MIDI <=> GamePort

Wiadomości synchronizacji są przydatne przy połączeniu wielu syntezatorów w łańcuch i umożliwiają zachowanie przez wszystkie jednakowego tempa oraz przykładowo nadanie sygnału startu i stopu. Wiadomości wspólne, do których należą wiadomości zastrzeżone, są natomiast znacznie bardziej złożone i ich zastosowanie jest dość wszechstronne. Za ich pomocą możliwe jest chociażby przesyłanie pomiędzy urządzeniami większych bloków danych (np. całych banków próbek dźwiękowych), a ich wykorzystanie różni się w zależności od producenta. Każda większa firma produkująca urządzenia muzyczne posiada bowiem unikalny kod, który wraz z nagłówkiem wiadomości zastrzeżonej tworzy bajt informacyjny, akceptowany najczęściej wyłącznie w urządzeniach produkowanych przez danego producenta. Jest to więc sposób rozszerzenia standardu MIDI i przyznanie większej swobody konstruktorom sprzętu muzycznego. Jak to już wcześniej zostało zaznaczone, blok danych składa się zazwyczaj z wyżej wymienionych bajtów informacyjnych, po których następują bajty danych o różnorodnym znaczeniu. Najważniejsze z nich zostały podane i objaśnione w **tab. 1**. Należy jeszcze wspomnieć o możliwości wielokrotnego wysyłania danych po jednokrotnym wysłaniu bajtu informacyjnego. Przykładowo, chcąc wysłać informację o trójdźwiękowym akordzie, wystarczy jedynie raz wysłać bajt informacyjny wiadomości dźwiękowej (np. *Note On*), a następnie trzykrotnie przesłać po dwa bajty

zawierające dane o wysokości dźwięku i jego sile. Możliwość ta pozwala skrócić czas potrzebny na przesłanie danego bloku nawet o 30%.

### Budowa instrumentu

Po przedstawionym wyżej skrótownym opisie interfejsu możemy w końcu przejść do głównego tematu artykułu - nadszedł czas na budowę instrumentu! W zasadzie większość klasycznych instrumentów została już w jakiś sposób tak zmodyfikowana, aby było możliwe ich łączenie w sposób cyfrowy. Do tych instrumentów należy także gitara, przy czym instrument ten nie poddaje się łatwo „ucyfrowieniu” i z tego powodu produkuje się specjalne przystawki wykorzystujące dosyć zaawansowaną technikę analizy dźwięku generowanego przez instrument klasyczny i dopiero w kolejnym kroku zamienia się ten dźwięk na postać zerojedynkową zapisu nutowego. Jak można się domyślić, tego rodzaju przystawki są dosyć skomplikowane (wymagają stosowania procesorów sygnałowych) i, co za tym idzie, drogie. Przedstawiony w artykule model gitary jest praktycznie „czysto” cyfrowy, co daje wiele różnych ciekawych możliwości, które są niedostępne przy stosowaniu przystawek do gitar strunowych. Co więcej, układ jest tani i prosty w budowie, a przy tym poddaje się łatwym modyfikacjom.

Schemat blokowy urządzenia został przedstawiony na **rys. 1**. Jak łatwo można było przewidzieć, centralnym elementem układu jest mikrokontroler, przy czym zadowolimy się nawet układem

o niewielkiej liczbie peryferii. Elektronika gitary jest niezmiernie prosta - mikrokontroler przyjmuje dane o miejscach nacisku danej struny na gryfie, sprawdza, które struny są pobudzone do grania, a następnie generuje odpowiednie kody MIDI przesyłane poprzez tenże interfejs. Dodatkowo, w celu wprowadzania możliwości zmian ustawień i konfiguracji gitary, została dołączona klawiatura i wyświetlacz, dzięki którym ustawienia mogą być łatwiej dokonywane. Znamy już teorię, czas zobaczyć, jak to zostało wykonane w praktyce. Schemat ideowy instrumentu został przedstawiony na **rys. 2**. Z powodów konstrukcyjnych podzielono go na dwie zasadnicze części - płytę główną z kontrolerem i złączami oraz trzy identyczne układy gryfu. Jako serce układu został wybrany mikrokontroler rodziny AVR - ATmega8515. Wybór ten był w zasadzie prosty - potrzebny był układ średnio szybki, posiadający względnie dużą liczbą portów wyjściowych. Mniej istotnymi cechami była wielkość pamięci programu, wyposażenie w rozbudowane funkcje specjalne - kryteria te świetnie spełnia nasz scalak.

W zasadzie nie ma potrzeby wyjaśniania działania budowanego układu od podstaw, gdyż jest ono bardzo proste. Omówienia wymaga jedynie sposób, w jaki układ wykrywa chwyt i jego pobudzenie. Struny, które są najzwyklejszymi kawałkami przewodu, przeciągniętymi jedynie w części korpusu gitary (czyli w miejscu, w którym gitarzysta je pociąga), są dołączane do układu poprzez rezystory R6-R11 zabez-

pieczające przed przepięciami i poprzez inwertery Schmitta wykonane w technologii CMOS. Aby potencjał strun „nie pływał“, przewody te są ściągane do masy poprzez rezystory R12-R17 o dużych wartościach rezystancji. Dzięki temu osiągamy możliwość „dotykowej“ gry na gitarze pod warunkiem, że artysta znajduje się na potencjale zasilania. Można to uzyskać, stosując przewodzącą obudowę gryfu, podłączoną do najwyższego potencjału w układzie. Zaskoczeniem może być to, że osobną częścią, rozłączną ze strunami służącymi do pociągania jest gryf, na którym nie interesują nas progi, ale odcinki strun przeciągnięte pomiędzy nimi. A zatem, mając na gryfie po sześć odcinków strun pomiędzy każdymi progami, mamy w sumie (dla 24 progów) aż 144 miejsca (odcinki strun), których dociśnięcie (dotyk) musimy zbadać. Niestety nie ma innego, równie niezawodnego sposobu, jak zastosowanie rejestrów przesuwających z wpisem równoległym i wyjściem szeregowym. Wszelkie konkurencyjne pomysły posiadały wady, które wykluczały możliwość ich zastosowania w omawianym urządzeniu. O jakimkolwiek multipleksowaniu nie można nawet myśleć, natomiast metody analogowe wymagałyby stosowania przetworników analogowo-cyfrowych i byłyby mniej niezawodne i bardziej kłopotliwe. Jak widać ze schematu, każdy odcinek struny jest dołączony zatem do osobnego wejścia rejestru szeregowego poprzez rezystory spełniające funkcje identyczne jak te wcześniej omawiane. Schemat płyt gryfu został narysowany w uproszczeniu - w rzeczywistości każda płyta (potrzebne są trzy) zawiera po sześć rejestrów przesuwających (każdy dla osobnego ciągu odcinków strun) o osobnych wejściach i wyjściach szeregowych. Płyty gryfu łączone są szeregowo poprzez złącza JP5-JP10 w taki sposób, że rejestry przesuwające połączone są po trzy w szeregu tak, aby każdej takiej trójce (24 stopniom) przypadał jeden ciąg odcinków strun. W ten sposób mikrokontroler odczytuje dane szeregowo-równoległe (szeregowo w sensie pobierania danych o kolejnych progach, rów-

noległe w sensie, że dla sześciu strun jednocześnie), co przyspiesza ten proces.

Ponieważ nasz instrument nie ma możliwości generowania dźwięku „od siebie“, to nie ma potrzeby implementowania w nim gniazda MIDI IN (ani MIDI THRU), za pomocą którego syntezatory mogą otrzymywać polecenia grania (zmiany ustawień, trybów pracy, synchronizacji itd.) z zewnątrz. Wystarczy jedynie MIDI OUT, którego budowa, jak widać, jest niezwykle prosta (co najważniejsze, do interfejsu MIDI możemy wykorzystywać wbudowany UART mikrokontrolera!). Jak to już wcześniej zostało powiedziane, interfejs MIDI opiera się na pętli prądowej, przy czym prąd wpuszczany w linię powinien wynosić około 5mA (przepływowi prądu odpowiada stan niski, zaś brak przepływu oznacza stan wysoki). Ponieważ wydajność prądowa portów mikrokontrolera jest kilkukrotnie wyższa niż 5mA, to wyjściem interfejsu może być wyjście mikrokontrolera, zabezpieczone jedynie rezystorem ograniczającym prąd i diodami chroniącymi przed wyładowaniami elektrostatycznymi. Po stronie odbiorczej podłączane urządzenie (syntezator, komputer) powinno być wyposażone w optoizolację.

Najpowszechniej dostępnym urządzeniem, które mogłoby „zrozumieć“ żargon budowanej przez nas gitary, jest komputer domowy wyposażony w kartę dźwiękową z GamePortem. Wykorzystamy go więc do współpracy z instrumentem. Na **rys. 3** przedstawiono schemat prostego układu konwertującego sygnały izolowanej pętli prądowej na poziomy napięcie TTL (z którymi kompatybilny jest GamePort) i odwrotnie. Interfejs taki można wykorzystać zarówno do podłączenia naszej gitary, jak i jakiegokolwiek innego urządzenia MIDI. Część MIDI OUT jest opcjonalna i w naszym wypadku po prostu zbędna (MIDI IN GamePort-u łączymy oczywiście z MIDI OUT gitary). Przedstawiony układ można z powodzeniem wykonać „na pajęczka“ i zmieścić w obudowie wtyku DB-15.

Program sterujący pracą mikrokontrolera U1 został napisany w assemblerze i dzięki temu zajął

## WYKAZ ELEMENTÓW

### Rezystory

R1, R2: 220Ω  
R3, R4: 2,2kΩ  
R5: 20kΩ pot. montażowy, stojący  
R6...R11, R18...R16: 300kΩ (SMD 0805)  
R12...R17, R162...R305: 4,7MΩ (SMD 0805)

### Kondensatory

C1: 470μF/25V  
C2, C4, C5, C11...C28: 100nF  
C3, C8...C10: 100μF/16V  
C6, C7: 22pF

### Półprzewodniki

D1: 1N4001  
D2, D3: 1N4148  
U1: Atmega8515 zaprogramowany  
U2: CD40106B  
U3: μA7805  
U4...U21: CD4021B

### Różne

X1: kwarc 8MHz  
W1: gniazdo zasilania  
W2: gniazdo DIN5, kątowe, do druku  
JP1: SIP5  
JP2: gniazdo IDC 10-pin + wtyk zaciskany na taśmie  
JP3: gniazdo IDC 14-pin  
JP4...10: gniazdo IDC 16-pin  
P1...P18: 8-pin złącza szpilkowe + nasadki z DIL16  
S1...S8: przyciski chwilowe  
S9: wyłącznik zasilania  
LCD alfanumeryczny 2\*16

niespełna połowę pamięci programu zastosowanego układu (około 3kB, plus dane w pamięci EEPROM), a przy tym jest odpowiednio szybki. Ze względu na stosunkowo dużą liczbę wykonywanych zadań, zrezygnowano z zastosowania przerwań, co wydaje się być wbrew sztuce, ale w naszym wypadku jest uzasadnione. W budowanym instrumencie nie występują bowiem zdarzenia nadrzędne, które wymagałyby natychmiastowej obsługi - wszystkie zadania są w praktyce równoprawne (przykładowo: ciągnięcie strun, ich tłumienie, zmiana chwytu - zagięcie tonu, zmiany ustawień itd.). A zatem część główna programu to pętla, która, sprawdzając stan poszczególnych podzespołów gitary i wybranych przez użytkownika ustawień, podejmuje odpowiednie działania.