

# Stacjonarny odtwarzacz MP3 z budzikiem (1)

Malkontenci zapewne zakrzykną „po co publikować taki projekt, skoro w sklepie można kupić odtwarzacz MP3 z pamięcią Flash za mniej niż 100 złotych?”. To prawda, jednak prezentowane przez nas urządzenie jest unikatowe ze względu na możliwość przystosowania go do pracy chociażby w samochodzie oraz bardzo dobrą jakością dźwięku. Ponadto satysfakcja z samodzielnie wykonanego urządzenia nie da się porównać z tą pojawiającą się przy zakupie masowo produkowanego urządzenia.

**Rekomendacje:** amatorzy dobrego dźwięku.

**Od redakcji** – Artykuł ma wielkie walory dydaktyczne. Autor w szczególności bardzo interesująco analizuje „bebechy” projektu.

## AVT-5195

W ofercie AVT:  
AVT-5195A – płytką drukowaną

### PODSTAWOWE PARAMETRY

- Układ zbudowany na 3 płytkach: klawiatura, płytka główna, przetwornik C/A
- Zdalnie sterowany pilotem podczerwieni
- Dopasowany do możliwości typowego, domowego zestawu audio
- Mikrokontroler AT91SAM9260
- Odtwarzanie utworów nagranych na karcie SD
- Zegar czasu rzeczywistego z funkcją budzika

### PROJEKTY POKREWNE

wymienione artykuły są w całości dostępne na CD

Tytuł artykułu	Nr EP/EdW	Kit
Kieszonkowy odtwarzacz MP3	EP 3-4/2003	---
Magnetofon cyfrowy DAR-001	EP 6-9/2007	---
Yampp-7 Kieszonkowy odtwarzacz MP3	EP 3-4/2003	---
Yampp-3	EP 12/2002-2/2003	---
Odtwarzacz MP3 z układem STA013	EP 8-9/2003	---
Yampp 3	EP 9-10/2002	---

### Jak robią to inni?

Projekt odtwarzacza różni się od większości publikowanych dotychczas przede wszystkim podejściem do dekodowania plików MP3: w najczęściej opisywanych do tej pory projektach stosowano mało wyszukany mikrokontroler typu AVR lub nawet jakąś pochodną 8051. Zadaniem mikrokontrolera było przede wszystkim odczytywanie danych z urządzenia magazynującego i wysyłanie ich do zewnętrznego, sprzętowego dekodera strumienia danych MP3, np. takiego jak jedna z wersji popularnego układu VS1001.

Przystępując do pracy takie podejście od razu odrzuciłem, ponieważ nie wnosi ono nic nowego – byłaby to tylko kolejna wersja tego, co już kilkakrotnie zostało zrobione i dokładnie opisane. Dużo bardziej ambitny jest projekt odtwarzacza MP3, którego auto-

rem jest Andreas Schwarz. Więcej informacji o tym projekcie znajdziemy pod adresem [http://embdev.net/articles/ARM\\_MP3/AAC\\_Player](http://embdev.net/articles/ARM_MP3/AAC_Player). W tym projekcie odpowiedzialność za całą funkcjonalność odtwarzacza spoczywa na mikrokontrolerze: ma on za zadanie odczytywać dane z karty, dekodować strumień audio,ysterować przetwornik cyfrowo-analogowy oraz zapewnić interfejs użytkownika. Zastosowany tam mikrokontroler to AT91SAM7S256 z rdzeniem ARM7TDMI taktowany zegarem o częstotliwości 54 MHz (wg informacji na stronie autora).

Planując własny odtwarzacz MP3 pierwotnie chciałem zbudować urządzenie takie jak prezentuje Andreas Schwarz. Jest to bardzo dobry projekt, lecz zniechęciło mnie do niego „wyżyłowanie” mikrokontrolera. Zastosowano tam bowiem różne zabiegi mające na celu zwiększenie szybkości przetwarzania programu, np. wykonywanie programu z pamięci RAM zamiast Flash. W wykorzystaniu większości mocy obliczeniowej mikrokontrolera oczywiście nie ma nic złego, lecz niestety zamyka to drogę do poważniejszej rozbudowy urządzenia. Np. trudnym zadaniem byłoby uruchomienie odtwarzania plików MP3 na SAM7, jeśli w programie używalibyśmy małego systemu operacyjnego takiego jak FreeRTOS, a ten z kolei przydaje się do bardziej rozbudowanego interfejsu użytkownika. Implementacja dodatkowych efektów dźwiękowych lub odtwarzania bardziej wymagających formatów plików także mogłaby być trudna.

Z powyższych powodów zdecydowałem się na użycie kompatybilnego, lecz mocniejszego mikrokontrolera: AT91SAM9260. Jego moc obliczeniowa jest wystarczająca zarówno do odtwarzania MP3, dla działania systemu operacyjnego, a także dla interfejsu użytkownika z wyświetlaczem graficznym. W swoim odtwarzaczu zrezygnowałem z formatu AAC, ponieważ na razie jest on znacznie mniej popularny niż „tradycyjne” MP3.

## Możliwości i wybrane rozwiązania układowe

Funkcjonalność prezentowanego urządzenia została dopasowana do typowych wymagań domowego sprzętu audio. Ponadto kierowałem się tym, aby zastosowane w nim części były bez problemów dostępne w handlu w przystępnych cenach przy ilościach detalicznych.

Głównym elementem odtwarzacza jest mikrokontroler AT91SAM9260 firmy Atmel. Wyposażony jest on w rdzeń ARM926EJ-S typowo taktowany częstotliwością zbliżoną do 200 MHz, ma po 8 KB pamięci cache dla instrukcji i dla danych oraz dysponuje bardzo rozbudowanym jak na tę klasę mikrokontrolera zestawem układów peryferyjnych. Dodatkowo jest dość tani, bo jego cena jest porównywalna jest z od dawna znanymi

układami rodziny AT91SAM7 z ARM7TDMI. Jego wady to przede wszystkim: brak wewnętrznej pamięci Flash i bardzo małe ilości wewnętrznej pamięci RAM (2×4 KB)

Odtwarzacz w swojej podstawowej wersji wyposażony jest w czarno-biały, czytelny, a jednocześnie w miarę niedrogi wyświetlacz LCD o rozdzielczości 128×64 pikseli. Mimo, że możliwe było zastosowanie kolorowej matrycy TFT nawet o wysokiej rozdzielczości, zrezygnowałem z takiej opcji przede wszystkim ze względu na: wyższą cenę takich matryc, brak popularnego i „nieśmiertelnego” kontrolera (jak KS0107/KS0108 dla matryc czarno-białych), oraz mniejszy zakres kątów, pod którymi widać wyraźny obraz.

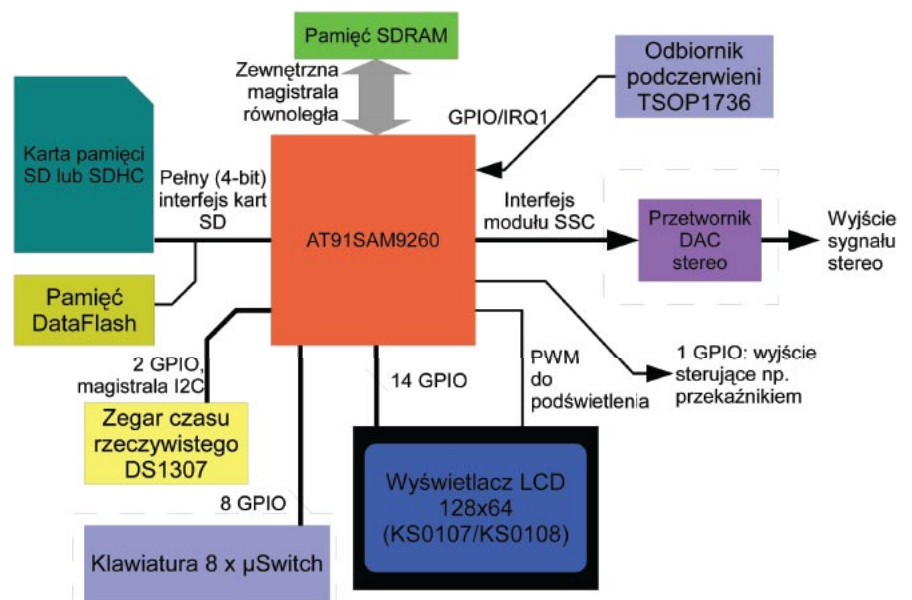
Do sprawowania kontroli nad odtwarzaczem może służyć pilot zdalnego sterowania. Jako standard transmisji danych w podczewieni wybrałem RC5 – do tej pory bardzo popularny także w amatorskich urządzeniach elektronicznych. O piloty pracujące w standardzie RC5 ciągle jest nietrudno, więc nie powinno być problemu z zakupem odpowiedniego nadajnika.

Źródłem danych dla urządzenia są pliki MP3 znajdujące się na karcie SD (*Secure Digital*). Pliki na karcie zapisane powinny być w jednej z odmian systemu plików FAT. Zastosowanie SD zamiast np. dysku twardego czy napędu CD/DVD podyktowane było przede wszystkim łatwością zasilania kart SD i ich obsługi przy pomocy mikrokontrolera. Urządzenie obsługuje karty SD o pojemnościach do 2 GB oraz karty SDHC (*SD High Capacity*) o pojemnościach do 32 GB (testowane do 8 GB). Wraz z pojawieniem się SDHC pojemności kart pamięci wreszcie nie są tak silnie ograniczone samym standardem, więc stosowanie w odtwarzaczu MP3 dużych i ciężkich dysków twardych lub napędów CD/DVD z przestarzałym już interfejsem PATA całkowicie mija się z celem.

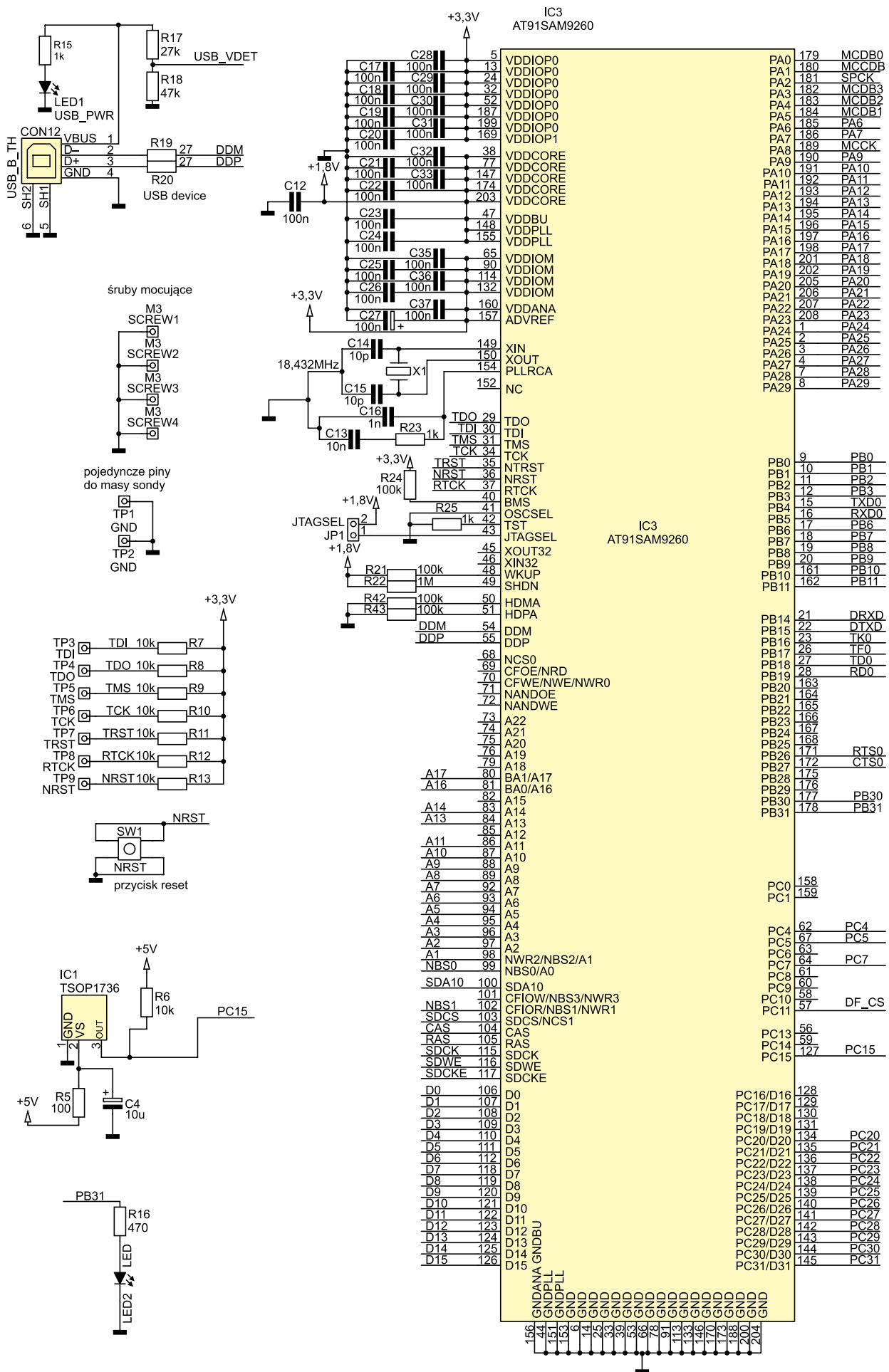
W projekcie odtwarzacza MP3 domyślnie zastosowany jest bardzo prosty moduł niedrogi i łatwo dostępnego przetwornika DAC o oznaczeniu UDA1330ATS produkcji NXP (jego cena detaliczna to kilkanaście złotych). Moduł przetwornika dołączany jest do płytki głównej. Na złączu zewnętrznego przetwornika wystawione są sygnały umożliwiające podłączenie zarówno przetworników DAC, ADC lub nawet całych kodeków audio. Hardware odtwarzacza można zatem z powodzeniem wykorzystać do skonstruowania rejestratora dźwięku otrzymując cyfrowy odpowiednik magnetofonu taśmowego lecz o dużo wyższej jakości.

Subiektywną jakość dźwięku, którą można uzyskać z omawianego odtwarzacza MP3 można określić jako co najmniej dobrą – przewyższa ona tanie bezimienne, przenośne odtwarzacze MP3. Jest tak przede wszystkim ze względu na dość wysoką jakość zastosowanego przetwornika DAC oraz – co także ma znaczenie – jakość zastosowanej biblioteki dekodowania strumienia danych MP3. W projekcie użyto biblioteki Helix MP3 Decoder dostępnej pod adresem <http://datatype.helixcommunity.org/Mp3dec>.

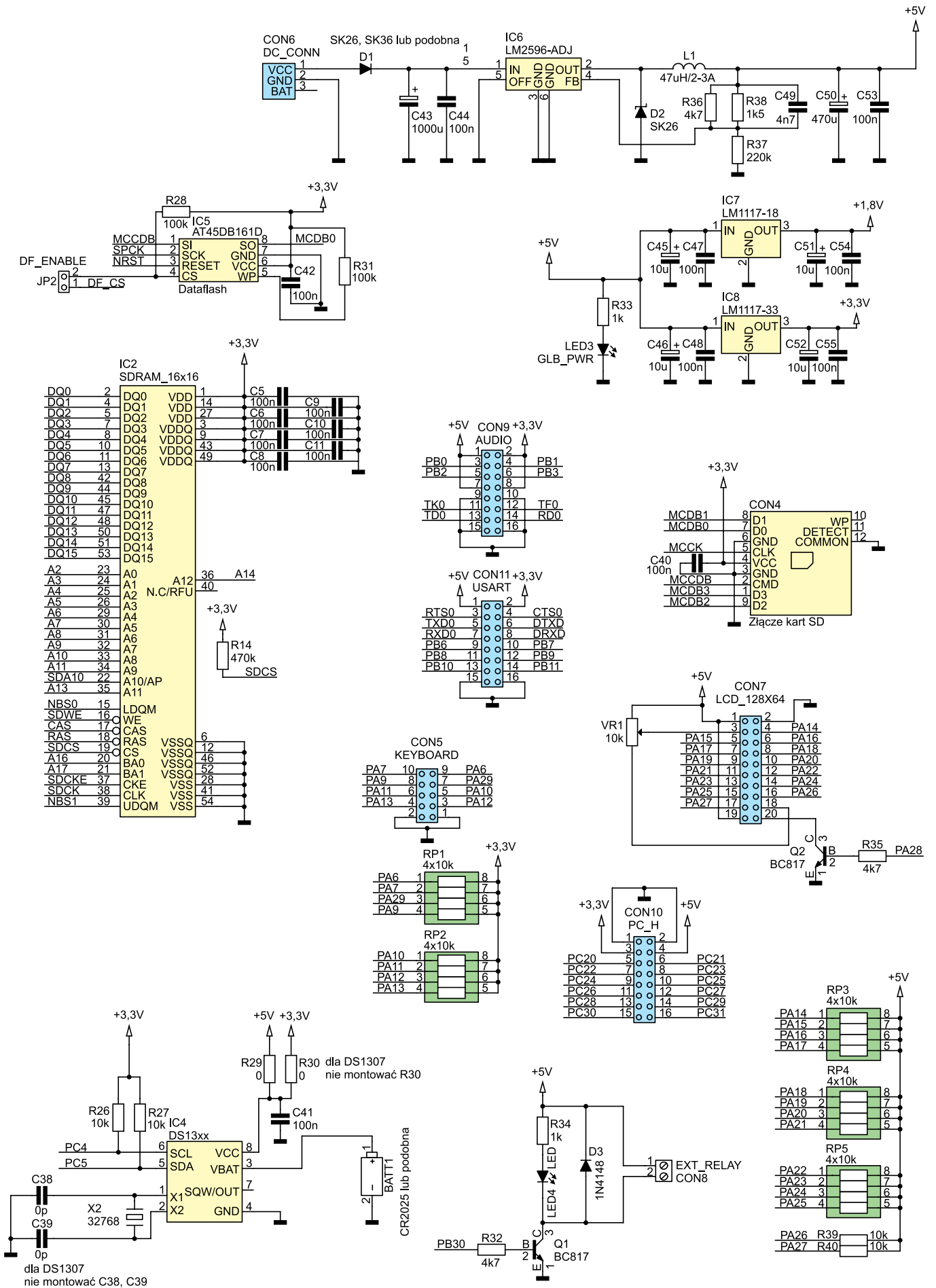
Omawiany odtwarzacz MP3 ma także możliwość odmierzania czasu oraz funkcję budzika. Oczywiście „budzik” oznacza rozpoczęcie odtwarzania od wybranego utworu o zaprogramowanej przez użytkownika godzinie. Dzięki temu pobudka może być znacznie przyjemniejsza niż przy zastosowaniu zwykłego sygnału budzika, a jednocześnie także bardziej przewidywalna niż alarm „radiobudzika”. Z racji, że odtwarzacz w założeniach ma być podłączany do zestawu audio, ma on możliwość sterowania przekaźnikiem włączającym np. wzmacniacz. Dzięki temu wzmacniacz nie będzie musiał być włączony całą noc, aby budzik mógł rano zagrać.



Rys. 1. Schemat blokowy odtwarzacza



Rys. 2. Schemat ideowy odtwarzacza



Rys. 2. c.d.



## WYKAZ ELEMENTÓW

### płytki głównej

#### Rezystory

R29: 0 Ω 0603  
R19, R20: 27 Ω 0603  
R5: 100 Ω 0603  
R16: 470 Ω 0603  
R15, R23, R25, R33, R34: 1 kΩ 0603  
R38: 1,5 kΩ 0603  
R32, R35, R36: 4,7 kΩ 0603  
R6...R13, R26, R27, R39, R40 : 10 kΩ 0603  
R17: 27 kΩ 0603  
R18: 47 kΩ 0603  
R21, R24, R28, R31, R42, R43: 100 kΩ 0603  
R37: 220 kΩ 0603  
R22: 1 MΩ 0603  
R14: 470 kΩ 0603  
R30: nie montować w podstawowej wersji  
RP1...RP5: drabinka rezystorowa 4×10 kΩ, rozmiar 1206  
VR1: potencjometr montażowy 10 kΩ pionowy

#### Kondensatory

C14, C15: 10 pF 0603  
C16: 1 nF 0603  
C13: 10 nF 0603  
C5...C11, C12, C17...C26, C28...C37, C40...C42, C44, C47, C48, C53...C55: 100 nF 0603  
C4, C27, C45, C46, C51, C52: 10 μF tantalowy, rozmiar A C50: 470 μF/10 V, elektrolityczny, przewlekany  
C43: 1000 μF/25 V, elektrolityczny, przewlekany  
C38, C39: nie montować w podstawowej

#### wersji

#### Półprzewodniki

T1, T2: BC817 lub podobny NPN w obudowie SOT-23  
D1: SK26, SK36 lub inna o  $I_F > = 1$  A D2: SK26 lub podobna  
D3: 1N4148 w obudowie LL-34  
LED1, LED3, LED4: dioda LED SMD 0805 (można nie montować)  
LED2: dioda LED do montażu przewlekane  
IC1: TSOP1736  
IC2: K45561632-UC75 w obudowie TSOP(II)  
IC3: AT91SAM9260 w obudowie PQFP-208  
IC4: DS1307 w obudowie SO-8  
IC5: AT45DB161D  
IC6: LM2596-ADJ w obudowie TO-263  
IC7: LM1117-18  
IC8: LM1117-33

#### Inne

TP1, TP2: pojedynczy goldpin  
JP1, JP1: zworka 2-pin  
BATT1: podstawka baterii litowej 3 V (np. CR2032)  
CON4: złącze kart SD bez wyrzutnika  
CON5: goldpin 2×5, box header („wannowe”)  
CON6: złącze zasilania DC  
CON7: goldpin 2×10, box header („wannowe”)  
CON8: złącze 2-pin, „terminal block” (śrubowe)  
CON9...CON11: goldpin 2×8, box header („wannowe”)  
CON12: USB-B kątowe, przewlekane

L1: 47 μH, ok. 2 A, rozmiar 12×12 mm, SMD

X1: oscylator kwarcowy 18,432 MHz, SMD  
X2: oscylator kwarcowy „zegarkowy” 32,768 kHz, przewlekany, montowany „powierzchniowo”  
SW1: przycisk „tact switch” SMD

#### płytki kodeka

#### Rezystory

R1: 0 Ω 0603  
R9, R10: 1 Ω 0603  
R4: 47 Ω 0603  
R7, R8: 100 Ω 0603  
R2, R3: 10 kΩ 0603  
R5, R6: 100 kΩ 0603

#### Kondensatory

C9, C10: 100 pF 0603  
C3...C5: 100 nF 0603  
C6: 10 μF tantalowy, rozmiar A C7, C8: 47...220 μF tantalowy, rozmiar D

#### Półprzewodniki

IC1: UDA1330ATS

#### Inne

CON1: goldpin 2×8, żeńskie montowane od spodu płytki  
CON2: goldpin 1×4

#### płytki klawiatury

#### Inne

CON1: goldpin 2×5, box header („wannowe”), montowane od spodu płytki  
SW1...SW8: przyciski „tact switch”, przewlekane

Program odtwarzacza ma dość złożoną funkcjonalność, dlatego zdecydowałem się na zastosowanie w nim prostego systemu operacyjnego czasu rzeczywistego. Wybrałem FreeRTOS – wybór był dość oczywisty, ponieważ jest to bardzo popularny, przejrzysty, stabilny i świetnie udokumentowany system operacyjny dla urządzeń embedded. Jego port dla mikrokontrolera AT91SAM9260 opracowałem korzystając z fragmentów kodu dla mikrokontrolerów SAM7. Dzięki choćby tak prostemu systemowi operacyjnemu program staje się bardziej przejrzysty, a dodawanie nowych opcji w programie jest dużo bardziej elastyczne niż bez zastosowania systemu. Nie skorzystałem z gotowych rozwiązań na AT91SAM9260 z systemem Linux, ponieważ do takiego zastosowania

jak odtwarzacz MP3, system operacyjny Linux to „armata na muchę”. Gdyby chcieć uruchamiać tutaj Linuxa, to po co byłoby projektować własny hardware – czyż nie prościej jest kupić przemysłowy komputer PC? Można na nim uruchomić system Linux, a równie dobrze nawet Windows, zabijając całą ideę sprzętowego odtwarzacza MP3. Tutaj, z powodu dość nietypowej (bo zoptymalizowanej) konfiguracji sprzętu, Linux mógłby sprawić dużo kłopotów przy uruchamianiu urządzenia. Kłopot ten nie zostałby zrekompenzowany przez zwiększenie funkcjonalności, ponieważ urządzenie jest zbyt proste.

## Sprzęt – płytki głównej

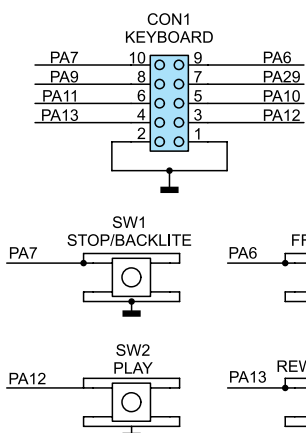
Schemat blokowy odtwarzacza umieszczono na **rys. 1**, a **rys. 2** przedstawia schemat ideowy płytki głównej. Na **rys. 3** zamieszczono schemat modułu prostej klawiatury.

W mikrokontrolerach rodziny AT91SAM9 (SAM9) duża liczba wyprowadzeń zewnętrznych nie powinna dziwić. W naj-

prostszym AT91SAM9260 zastosowanym w odtwarzaczu jest ich aż 208. Same wejścia zasilania i masy zajmują kilkadziesiąt wyprowadzeń. Z racji tego, że AT91SAM9260 nie ma wewnętrznego stabilizatora napięcia 1,8 V, należy do niego dołączyć co najmniej dwa napięcia: typowo 3,3 V służące głównie do zasilania modułów wejścia-wyjścia oraz 1,8 V najczęściej używane do zasilania bloków wewnętrznych.

Do uzyskania napięć 5 V, 3,3 V i 1,8 V zastosowałem popularne stabilizatory, odpowiednio: LM2596-ADJ (IC6), LM1117-33 (IC8) i LM1117-18 (IC7). Jeśli zamiast LM2596-ADJ (napięcie ustalone zewnętrznym dzielnikiem rezystancyjnym) zastosujemy LM2596-5.0 (fabrycznie ustalone napięcie 5 V), to wystarczy, że zamiast R36 lub R37 zamontujemy zworę, a C49 i R38 nie zamontujemy wcale.

Elementy zewnętrzne służące do zapewnienia sygnału zegarowego dla mikrokontrolera to: oscylator kwarcowy X1 o częstotliwości 18,432 MHz wraz kondensatorami C14 i C15 oraz zewnętrzny filtr modułu PLL składający się z elementów R23, C13 i C16. Jest to bardzo podobne rozwiązanie jak stosowane w mikrokontrolerach rodziny SAM7. W przypadku SAM9 mamy także do dyspozycji dodatkową, wewnętrzną pętlę PLL (o nazwie PLLB) przeznaczoną głównie do taktowania modułów USB częstotliwością 48 MHz i nie wymagającą żadnych dodatkowych elementów zewnętrznych.



Rys. 3. Schemat ideowy klawiatury

Wyprowadzenia BMS i OSCSEL to wyprowadzenia konfiguracyjne, od stanu których zależy start mikrokontrolera. W naszym przypadku BMS=1, co oznacza, że pierwszym programem, który wystartuje po włączeniu zasilania będzie kod bootloadera znajdującego się w wewnętrznej pamięci ROM mikrokontrolera (nazwijmy go bootloaderem I poziomu). Natomiast OSCSEL=0 oznaczać będzie, że mikrokontroler startuje taktowany z wewnętrznej pamięci generatora RC o niskiej częstotliwości (wtedy także nie ma potrzeby dołączenia do wyprowadzeń XIN32 i XOUT32 dodatkowego, zewnętrznego oscylatora kwarcowego).

W projekcie nie korzystamy z kilku innych wyprowadzeń: HDMA i HDPDA (odpowiednio D- i D+ kontrolera hosta USB), WKUP (wybudzanie mikrokontrolera) czy SHDN (sygnał informujący o przejściu w stan uśpienia). Interfejs JTAG (nóżki TDI, TDO, TMS, TCK, TRST i RTCK) wyprowadzony został jedynie w postaci „padów” (TP3...TP9) na płytce drukowanej, ponieważ odtwarzacz w swoich założeniach nie jest płytką ewaluacyjną, więc nie będzie też potrzeby intensywnego debugowania kodu przez JTAG (programowanie pamięci Flash można wykonać przez port szeregowy DBGU albo USB). Rezygnacja z typowego, 20-pinowego złącza JTAG w pewnym stopniu przyczyniła się także do redukcji rozmiarów płytki drukowanej, a co za tym idzie kosztów całego odtwarzacza.

Interfejs urządzenia USB (USB Device, moduł UDP) został przewidziany na płytce drukowanej, lecz odtwarzacz praktycznie z niego nie korzysta. Teoretycznie można by zaimplementować w projekcie funkcjonalność klasy magazynującej USB, lecz nie będzie z tego wielkiego pożytku przede wszystkim ze względu na relatywnie powolne transfery USB Full Speed. Szybkość oferowana w trybie USB Speed to jedynie 12 Mbit/s (razem z transferami kontrolnymi, więc na właściwe dane zostaje jeszcze mniej). Dlatego o wiele wygodniejszym rozwiązaniem we „wgrzywaniu” plików MP3 na kartę jest zastosowanie taniego, fabrycznego czytnika kart pracującego z szybkością High Speed, a następnie umieszczenie przygotowanej karty w odtwarzaczu. Zapewnienie plikami MP3 8-gigabajtowej karty SDHC przy USB Full Speed trwałoby naprawdę długo.

Karta SD lub SDHC (w złączu CON4) podłączona jest do mikrokontrolera przy pomocy dedykowanego interfejsu kontrolera MCI (MultiMedia Card Interface). Interfejs ten może posługiwać się 4-bitową magistralą danych (sygnały MCDB3:0), sygnałem komend (MCCDB) i zegarem (MCCK). Część wyprowadzeń modułu MCI jest współdzielona z modułem SPI0, dlatego na schemacie ideowym możemy zobaczyć, że wyprowadzenia SI i SO (nóżki 1 i 8) pamięci DataFlash (IC5) podłączone są do sygnałów MCDB0 i MCCDB. Jed-

nak obie pamięci (karta SD i chip DataFlash) nie przeszkadzają sobie nawzajem.

Mikrokontrolery AT91SAM9260 nie posiadają wewnętrznej pamięci Flash, dlatego cały kod programu odtwarzacza MP3 przechowywany jest w zewnętrznej pamięci DataFlash (IC5). Wewnętrzny program bootloadera pierwszego poziomu (czyli tego, od którego rozpoczyna się praca mikrokontrolera) wykrywa, czy w pamięci DataFlash znajduje się poprawny program. Jeśli tak – program ten ładowany jest do wewnętrznej pamięci SRAM0 (4 KB od adresu 0x200000), a jeśli nie – uruchamiane są procedury komunikacji z programem SAM-BA (przez USB i port szeregowy DBGU). Czasem, gdy w pamięci DataFlash znajduje się już właściwy program, a my chcemy skomunikować mikrokontroler z SAM-BA, zachodzi potrzeba zasymulowania odłączenia DataFlash-a, aby mikrokontroler pomyślał, że nie ma w nim żadnego programu. Do przeprowadzenia tego „oszustwa” służy zworka JP2, przy pomocy której możemy odciąć sygnał aktywacji pamięci DataFlash od wyprowadzenia kontrolującego tę pamięć (sygnał DF\_CS). Efekt rozłączenia tej zworki będzie taki sam, jakby w pamięci DataFlash nie było żadnego programu. Oprogramowanie testowane było z pamięcią DataFlash o oznaczeniu AT45DB161D (2 MB), lecz powinno także poprawnie działać z mniejszymi pamięciami, np. AT45DB081D (1 MB).

Mikrokontrolery AT91SAM9260 posiadają tragicznie małe ilości wewnętrznej pamięci RAM. Dysponują one jedynie dwoma 4-kilobajtowymi blokami tych pamięci o nazwach SRAM0 (na adresie 0x200000) i SRAM1 (0x300000). Jedynymi rozsądnymi programami, które można z nich w całości uruchomić to proste bootloadery drugiego poziomu (uruchamiane bootloaderem pierwszego poziomu zawartym w wewnętrznej pamięci ROM mikrokontrolera). W „dużych programach” można ich używać najwyżej jako szybkie pamięci „pomocnicze”. Dlatego do mikrokontrolera dołączona jest gigantyczna jak na urządzenie embedded pamięć SDRAM. Oczywiście w tak prostym urządzeniu jak odtwarzacz MP3 nie skorzystamy z niej całej. W prototypie odtwarzacza zastosowano popularną pamięć SDRAM firmy Samsung (oznaczenie K4S561632J-UC75) o organizacji 16×16 Mbit (o pojemności 32 MB z 16-bitową magistralą). Bez zmiany oprogramowania i bez większych problemów powinny też zadziałać mniejsze pamięci, np. o organizacji 16×8 Mbit (16 MB przy 16-bitowej magistrali). Pamięć SDRAM, mimo swoich rozmiarów nie jest droga – cena detaliczna pamięci o pojemności 32 MB to kilkanaście zł. Pamięć SDRAM podłącza się do mikrokontrolerów SAM9 najłatwiej przy pomocy dedykowanej magistrali będącej interfejsem modułu peryferyjnego EBI (External Bus Interface). Dzięki kontrolerowi EBI dołączone do mikrokontrolera pamięci równoległe

będą gładko wkomponowane we fragmenty jego przestrzeni adresowej zależnie od ich wielkości, a niezależnie od ich organizacji, czy rodzaju. Do EBI możemy w elastyczny sposób podłączać zarówno SDRAM, NAND- i NOR-Flash, jak też różne odmiany statycznej pamięci RAM czy nawet karty CompactFlash z interfejsem równoległym. Podłączając pamięć SDRAM do kontrolera EBI we własnym programie nie musimy się martwić ani o adresowanie pamięci SDRAM, ani o jej odświeżanie – robi to za nas odpowiednio skonfigurowany moduł EBI oraz stowarzyszony z nim kontroler SDRAM.

W przeciwieństwie do większości płytek ewaluacyjnych z mikrokontrolerami ARM9 firmy Atmel, w odtwarzaczu zdecydowałem się na zastosowanie tylko jednej „kości” pamięci SDRAM (w większości przypadków stosuje się 2 pamięci SDRAM z magistralami 16-bitowymi tworzące razem jakby „jedną pamięć” SDRAM o organizacji 32-bitowej). Jeśli przyjrzymy się bliżej płytkom z dwiema pamięciami SDRAM, to albo są one wielowarstwowe (co najmniej 4-warstwowe), albo pamięci SDRAM nie mogą na nich działać z pełnymi prędkościami, albo... jedno i drugie. W odtwarzaczu wolałem zastosować jedną pamięć SDRAM na zwykłej i dość taniej płytce drukowanej jednak z porządnie zaprojektowaną mozaiką ścieżek. Ostrożne prowadzenie ścieżek na płytce odtwarzacza pozwoliło na uzyskanie pełnej możliwej do osiągnięcia szybkości w dostęпах do SDRAM (sygnał SCKE może mieć częstotliwość nawet 100 MHz) przy zastosowaniu dość niewyszukanej technologii wytwarzania obwodów drukowanych.

Interfejs użytkownika zapewniony jest dzięki 8-przyciskowej klawiaturze, wyświetlaczowi LCD i układowi scalonemu odbiornika podczerwieni. Sama klawiatura nie wymaga żadnego komentarza. Wyświetlacz podłączamy do złącza CON7, a jego kontrast można regulować potencjometrem montażowym VR1. Układ wyprowadzeń CON7 odpowiada wyprowadzeniom miniatury wersji wyświetlacza 128×64. Wyświetlacze 128×64 o standardowej wielkości będą wymagały połączenia wg **tab. 1**. Podświetlenie wyświetlacza LCD sterowane jest z wyprowadzenia PA28 multipleksowanego z sygnałem TIOA2 układu Timer Counter 2. Do uzyskania sygnału PWM użyłem układu Timer Counter, ponieważ w mikrokontrolerze AT91SAM9260 nie ma typowych kontrolerów PWM (modułów PWM) znanych z SAM7.

Jeśli chodzi o podłączenie wyświetlacza 128×64, to zastosowałem tutaj pewien trick, którego używam praktycznie za każdym razem, gdy chcę podłączyć do mikrokontrolera SAM7/9 niezbyt szybkie urządzenie z interfejsem CMOS 5 V. Napięcia wyjściowe z wyprowadzeń PIO mikrokontrolera mogą mieć maksymalny poziom typowo 3,3 V (bez

obciążenia). Może to nie być wystarczające napięcie dla wejść niektórych wyświetlaczy LCD, dlatego stan wysoki wymuszony zostaje rezystorami podciągającymi R39, R40 i drabinkami rezystorowymi RP3...RP5 (podciągają do +5 V) podczas gdy wyjścia mikrokontrolera pracują w trybie „otwarty kolektor” (stan niski to zwarcie do masy, a stan wysoki to w stan wysokiej impedancji). Nie jest to może oszczędne rozwiązanie pod względem zużycia energii, lecz na pewno znacznie upraszcza schemat i projekt płytki drukowanej ze względu na brak konieczności stosowania jakichkolwiek buforów do translacji napięć.

Scalony odbiornik podczerwieni typu TSOP1736 (dla częstotliwości 36 kHz) podłączony jest w typowy sposób, jak zaleca producent. Jego sygnał wyjściowy doprowadzony jest do portu PC15 mikrokontrolera. Sygnał z pilota może zgłaszać przerwanie, ponieważ PC15 multipleksowany jest z wejściem zewnętrznego przerwania IRQ1 (co prawda PIO także może zgłosić przerwanie, ale obsługa pilota staje się bardziej elegancka, jeśli zastосуemy sygnał IRQ).

Przełącznik do włączania wzmacniacza, o którym była mowa wcześniej, można podłączyć do złącza CON8. Jest to złącze, na którym pojawia się napięcie 5 V wtedy, gdy wyprowadzenie PB30 mikrokontrolera jest w stanie wysokim. Napięcie to w założeniach ma służyć do zasilania cewki przełącznika. Przełącznik podłączany do CON8 powinien mieć napięcie znamionowe cewki wynoszące ok. 5 V. Nie znajduje się on na płytce głównej aby uniknąć wprowadzania na nią wysokiego napięcia zasilania i niepotrzebnego powiększania rozmiarów płytki.

Mikrokontroler AT91SAM9260 ma wbudowany zegar czasu rzeczywistego, który może być taktowany zewnętrznym oscylatorem podłączonym do wyprowadzeń XIN32 i XOUT32. Jak mogliśmy się dowiedzieć, wyprowadzenia te w projekcie odtwarzacza nie są używane. Jest tak dlatego, że w mikrokontrolerach AT91SAM9260 w wersji A (rev. A, wg erraty w nocie katalogowej) program bootloadera pierwszego poziomu resetuje ustawienia wewnętrznego licznika czasu (w module RTT). Przez to funkcjonalność zegara czasu rzeczywistego staje się w takiej konfiguracji bezużyteczna, a w momencie pisania tego artykułu mikrokontrolery w wersji A stanowiły większość AT91SAM9260 dostępnych na polskim rynku. Z kolei bootloadera musimy używać (wyprowadzenie BMS=1), ponieważ program do wykonania znajduje się w pamięci DataFlash, która jest odczytywana przez wspomniany bootloader pierwszego poziomu. Errata wspomina o tym mankancie tylko dla AT91SAM9260 w wersji A, milczy natomiast odnośnie wersji B (prawdopodobnie oznacza to, że w rev. B błąd ten nie występuje, lecz nie miałem możliwości sprawdzenia tego w praktyce).

Tab. 1.

Katalog	Pliki	Krótki opis zawartości
Główny	main.c	funkcja main, inicjalizacja podstawowych sterowników, utworzenie zdania głównego
	board.h, common.h	często wykorzystywane i globalne dla całego programu makrodefinicje oraz dyrektywy #include
	syscalls.c	funkcje umożliwiające działanie niektórych bibliotek języka C
	FreeRTOSConfig.h	makrodefinicje konfiguracyjne dla systemu FreeRTOS
	user_if.c, user_if.h	realizacja interfejsu użytkownika odtwarzacza, realizacja innych ważnych zadań takich jak obsługa modułów odtwarzacza
	player_task.c, player_task.h	zadanie systemu operacyjnego realizujące odtwarzanie MP3 oraz funkcje wysokiego poziomu sprawujące kontrolę nad odtwarzaniem
	player_core.c, player_core.h	„rdzeń” odtwarzacza MP3: funkcje realizujące obsługę biblioteki Helix, nadzorujące pracę przetwornika DAC i pobieranie danych z plików MP3
	id3.c, id3.h	odczyt tagów ID3v1 z plików MP3
include	AT91SAM9260.h, lib_AT91SAM9260.h	pliki nagłówkowe z makrodefinicjami specyficznymi dla układów peryferyjnych mikrokontrolera, strukturami danych reprezentującymi te układy i funkcjami inline do niskopoziomowej ich obsługi
	s dram.lids	skrypt linkera GCC (LD) umożliwiający wykonywanie programu i przechowywanie danych w zewnętrznej pamięci SDRAM
mp3dec	biblioteka dekodowania plików MP3 (Helix MP3 decoder)	
	mp3dec.c	wysokopoziomowe funkcje do dekodowania i analizy strumienia danych w formacie MP3 („interfejs użytkownika” biblioteki)
freertos	pliki wchodzące w skład systemu operacyjnego FreeRTOS oraz jego portu dla mikrokontrolerów AT91SAM	
fat	elementy biblioteki FatFS do obsługi systemu plików FAT	
drivers	moduły sterowników układów peryferyjnych	
	backlite.c, backlite.h	obsługa podświetlenia wyświetlacza LCD przy pomocy modułu Timer Counter
	cstartup.S, Cstartup_SAM9.c	niskopoziomowa inicjalizacja mikrokontrolera oraz obsługa niektórych wyjątków
	dbgu.c, dbgu.h	funkcje odpowiedzialne za realizację komunikacji „serwisowej” przez port szeregowy z modułu DBGU
	delay.c, delay.h	opóźnienia w trybie polling uzyskane przy pomocy modułu Timer Counter
	i2c.c, i2c.h	programowa implementacja komunikacji przez I <sup>2</sup> C przy użyciu linii PIO
	keyboard.c, keyboard.h	obsługa klawiatury
	rc5_ir.c, rc5_ir.h	odbiór kodów RC5 z pilota zdalnego sterowania
	remote_control.c, remote_control.h	„most” pomiędzy funkcjami obsługi pilota, a pozostałymi elementami systemu; w tych plikach można „podmienić” obsługę pilota
	rtc.c, rtc.h	obsługa układu odmierzania czasu rzeczywistego (tutaj DS1307)
	stereo_dac.c/h, stereo_dac_isr.c/h	obsługa przetwornika cyfrowo-analogowego
	time.c, time.h	funkcje pomocnicze do odmierzania czasu rzeczywistego

Z powyższych względów, do odliczania czasu rzeczywistego używany jest zewnętrzny układ scalony DS1307 (IC4) wraz z baterią BATT1 podtrzymującą jego działanie także wtedy, gdy zasilanie odtwarzacza jest wyłączone. Do komunikacji z IC4 służy magistrala I<sup>2</sup>C zaimplementowana programowo na wyprowadzeniach PC4 i PC5 używanych jako uniwersalne wejścia-wyjścia. DS1307 zasilany jest napięciem 5 V (zamontowana zwora R29), lecz jego

magistrala I2C jest kompatybilna z urządzeniami zasilanymi z napięcia 3,3 V (niskie minimalne napięcia stanu wysokiego). Gdybyśmy chcieli zastosować w miejscu DS1307 jakiś inny układ RTC pasujący topologią wyprowadzeń, możemy to uczynić (prawdopodobnie jednak nie obejdzie się bez modyfikacji software’u). W zamianie pomocna może być możliwość wyboru napięcia zasilania układu RTC przy pomocy zworek R29 (+5 V) lub R30 (+3,3 V).

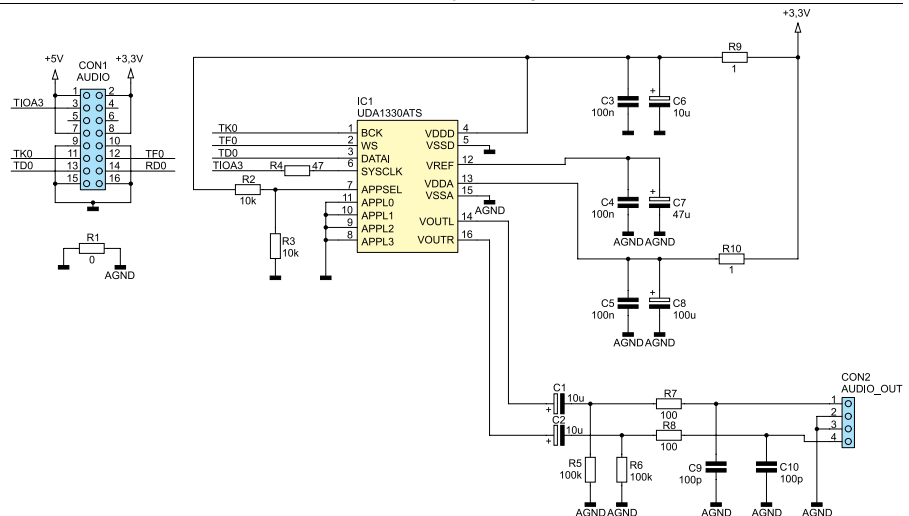
Dwa złącza na płycie głównej odtwarzacza (CON10 i CON11) umieszczone są dla umożliwienia dalszej rozbudowy urządzenia. Złącze CON10 zawiera wyprowadzenia bardziej znaczących bitów portu PC. Są one wolne, ponieważ korzystamy z tylko jednego układu pamięci SDRAM. Złącze CON11 zawiera sygnały kilku portów szeregowych USART, w tym port DBGU. Port DBGU (sygnały DTXD i DRXD) będzie potrzebny do zaprogramowania zawartości pamięci DataFlash na płycie odtwarzacza.

## Cyfrowy sygnał audio i moduł przetwornika DAC

Na złączu CON9 wyprowadzone zostały m.in. sygnały potrzebne do działania przetwornika DAC lub kodeka audio. Przetwornik DAC działający w typowym standardzie I<sup>2</sup>S potrzebował będzie 4 sygnałów: linii danych, sygnału zegarowego synchronizującego poszczególne bity danych, synchronizację całych słów (pełnych próbek kanału lewego i prawego) oraz głównego sygnału taktującego. W mikrokontrolerach rodziny AT91SAM znajdują się specjalne układy peryferyjne o nazwie SSC (Synchronous Serial Controller) zdolne do wysyłania i odbierania danych m.in. do przetworników audio przetwarzających dane w standardzie I<sup>2</sup>S lub podobnym. W **tab. 2** znajdziemy wykaz najważniejszych wyprowadzeń modułu SSC wraz z ich rozmieszczeniem na złączu CON9. Zauważmy, że nie wszystkie wyprowadzenia SSC znalazły się na złączu CON9. Jest tak dlatego, że nawet do podłączenia kodeka audio (przetwornik ADC i DAC w jednym module bądź układzie scalonym) zazwyczaj nie są potrzebne linie synchronizacji bitów i ramek odbiornika (RK0 i RF0), a zamiast nich odbiornik synchronizowany może być z nadajnikiem (nadajnik i odbiornik wspólnie używają linii synchronizacji TK0 i TF0).

Złącze CON9 oprócz podstawowych sygnałów modułu SSC zawiera także dodatkowe wyprowadzenia, które mogą stanowić zarówno wejścia-wyjścia ogólnego przeznaczenia, źródła przebiegów PWM (TIOA, TIOB) lub interfejs SPI0. Przebiegi PWM o wybranej częstotliwości mogą służyć jako sygnały zegarowe m.in. do taktowania przetworników audio lub kodeków.

Moduł z przetwornikiem UDA1330ATS (schemat na **rys. 4**) został w większości oparty na danych aplikacyjnych producenta zaczerpniętych z noty katalogowej i nie ma w nim szczególnej filozofii czy układowych tricków. Po prostu wprowadzamy sygnał cyfrowy audio (DATAI, BCK, WS) oraz sygnał zegarowy (SYSCLK) i przy pomocy układu scalonego IC1 z kilkoma zewnętrznymi elementami biernymi otrzymujemy analogowy sygnał stereo (kanał lewy na VOUTL i kanał prawy na VOUTR). Zasilanie części analogowej i cyfrowej rozdzielone jest rezystorami R9 i R10 o wartości 1 Ω oraz filtrowane przy pomocy kondensatorów C3, C5, C6, C8. Wewnętrzne napięcie odniesienia przetwornika (VREF) także jest filtrowane



Rys. 4. Schemat modułu z przetwornikiem UDA1330ATS

– do tego służą kondensatory C4 i C7. Z racji, że sygnał audio powinien oscylovąć wokół masy układu, a przetwornik zasilany jest pojedynczym napięciem, to zastosowano kondensatory C1 i C2 odcinające składowe stałe w obu kanałach. Za nimi znajdują się rezystory R5 i R6 ustalające składową stałą sygnałów wyjściowych na masie. Elementy R7, R8, C9 i C10 nie są konieczne: kondensator można nie montować, a rezystory można zastąpić zwozami. Zamontowanie tych elementów (lub na ich miejsce innych, o wybranych wartościach) może nieco zmniejszyć skutki przenikania sygnałów o wysokiej częstotliwości do linii pomiędzy przetwornikiem a wzmacniaczem, lecz w warunkach „domowych” taki problem praktycznie nie występuje.

Wyprowadzenia APPSEL i APPL0...3 przetwornika służą głównie do jego konfiguracji. Szczegóły na temat ich działania znajdują się w nocie katalogowej UDA1330ATS. W przypadku odtwarzacza, APPSEL ustawione jest na wartość połowy napięcia zasilania (dzielnik R2, R3), a wszystkie wyprowadzenia APPL0...3 są na potencjale masy. Przy takich ustawieniach przetwornik pracuje w podstawowym konfiguracji, z popularnym interfejsem I<sup>2</sup>S, przy częstotliwości sygnału zegarowego SYSCLK wynoszącej  $384 \times f_s$ , czyli 384 razy wyższej niż częstotliwość próbkowania przetwornika.

Zależności czasowe pomiędzy sygnałami wprowadzanymi do przetwornika są bardzo istotne. W przedstawionej konfiguracji, jak już mieliśmy okazję się dowiedzieć, sygnał SYSCLK musi mieć wartość  $384 \times f_s$ . Jeśli  $f_s$  przyjmujemy typową jak dla większości nagrań w „znośnej” jakości czyli 44100 Hz, to częstotliwość SYSCLK powinna wynosić 16,9344 MHz (czyli  $384 \times 44100$  Hz). Jak na SAM9 nie jest to bardzo wysoka częstotliwość, lecz problemem może być uzyskanie jej dokładnej wartości przy pomocy dostępnego sprzętu (dzielniki, preskalery). Zastosowanie zewnętrznego generatora tylko dla sygnału SYSCLK nie sprawdzi się, ponieważ SYSCLK musi być w pełni zsynchronizowany z sygnałem BCLK (taktowania bitów transmisji uzyskanym z TK0). Oznacza to, że

najlepiej, jakby oba sygnały: SYSCLK i BCLK wygenerowane zostały przez mikrokontroler. Do przetwornika DAC nie są dostarczane sygnały o dokładnych częstotliwościach potrzebnych do próbkowania 44100 kHz. Mimo to, stosunek częstotliwości SYSCLK do BCLK jest idealnie zachowany, ponieważ jest to krytyczne dla działania samego przetwornika DAC. W rzeczywistości odtwarzacz będzie „grał” utwory MP3 praktycznie niezauważalnie wolniej, bo z częstotliwością próbkowania wynoszącą ok. 43111 kHz. Przy takiej częstotliwości, w ciągu jednej minuty utworu rozbieżność pomiędzy poprawnym a rzeczywistym czasem trwania wyniesie niewiele ponad 1 s. Z racji, że praktycznie wszystkie utwory MP3 w jakości nadającej się do komfortowego słuchania mają częstotliwość próbkowania 44100 kHz, to **tylko taka częstotliwość jest poprawnie obsługiwana** przez odtwarzacz. Obsługa innych częstotliwości próbkowania może być bez większych problemów zaimplementowana, lecz wymaga to modyfikacji oprogramowania.

Robert Brzoza-Woch  
rabw@poczta.fm

Nazwa pola	Rozmiar pola w bajtach	Opis
id	3	nagłówek identyfikujący taga ID3v1 – trzyznakowy tekst „TAG”
title	30	tytuł utworu
artist	30	nazwa wykonawcy
album	30	tytuł albumu
year	4	rok wydania
comment	28	komentarz tekstowy
zeroByte	1	jeśli ten bajt zawiera wartość 0, to kolejny bajt (trackNumber) odczytany zostanie jako numer ścieżki
track-Number	1	numer ścieżki w albumie
genre-Code	1	kod gatunku muzyki