

Zegar dla szachistów

Nowoczesna platforma dla STM32



W listach do redakcji EP Czytelnicy domagają się zamieszczenia większej ilości artykułów, które pomogłyby im się rozwijać. Publikując ten projekt postaraliśmy się wyjść naprzeciw tym oczekiwaniom.

Nowoczesny mikrokontroler STM32, jego otoczenie oraz duży, czytelny wyświetlacz graficzny tworzą urządzenie, na bazie którego można wykonać wiele różnych projektów. Można powiedzieć, że implementacja zegara szachowego na bazie niniejszego projektu, to tylko przykład zastosowania, ponieważ możliwości mikrokontrolera znacznie przekraczają wymagania takiej aplikacji.

Rekomendacje: osoby zainteresowane budową niedrogiej, uniwersalnej platformy sprzętowej dla STM32.

Cortex-M3 znaczy nowoczesne

Procesory z rodziny STM32 cieszą się coraz większą popularnością. Nic w tym dziwnego, skoro wewnątrz procesora jest bardzo dużo nowoczesnych i dojrzałych rozwiązań. Moim zdaniem jednym z najważniejszych jest bez wątpienia możliwość sprzętowego zagnieżdżenia przerwania o różnych priorytetach. Można powiedzieć, że opracowanie NVIC (Nested Vector Interrupt Controller) przez inżynierów z firmy ARM jest naprawdę przełomem w porównaniu do np. starej już architektury ARM9. System przerwania w tym procesorze działa bardzo sprawnie i bez wątpienia to główna jego zaleta. Jeśli chodzi o peryferia, to 5 portów UART, CAN, USB oraz szereg timerów to już chyba standard.

W projekcie nie będziemy zagłębiać się w nic skomplikowanego. Cały projekt bazuje na obsłudze portów procesora, pętli głównej oraz jednego przerwania wywoływanego co 1 ms, a służącego do odmierzenia czasów w programie. Wszyscy projektanci z pewnością zauważą, że poznanie takiej konfiguracji procesora to swoisty „Hello word” w świecie programowania systemów wbudowanych. Napisany i skompilowany program to dopie-

AVT-5199

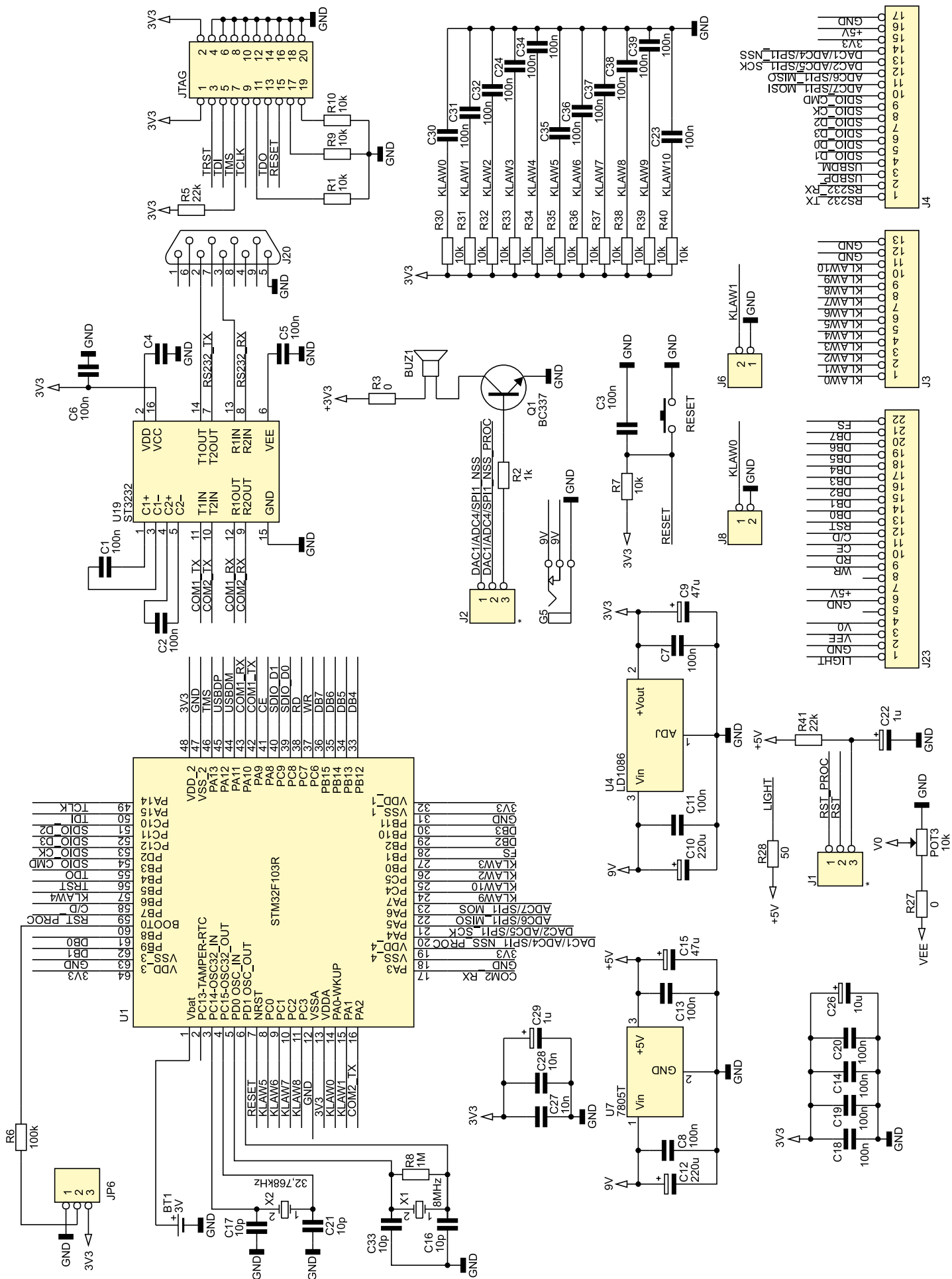
W ofercie AVT:
AVT-5199A – płytka drukowana

PODSTAWOWE PARAMETRY

- Doskonały układ bazowy do projektów z STM32
- Płytkę dwustronną o wymiarach 172x115 mm
- Mikrokontroler STM32F103R z zegarem 72 MHz
- Zasilanie 9...12 VDC
- Wyświetlacz graficzny 240x128 pikseli
- Złącze RS232, wbudowany buzzer, złącze JTAG
- Interfejsy dostępne na złączu szplikowym (w tym USB, drugi RS232, interfejs kart SD i microSD, SPI)

PROJEKTY POKREWNE wymienione artykuły są w całości dostępne na CD

Tytuł artykułu	Nr EP/EdW	Kit
Zegar szachowy	EP 2/2005	AVT-378



Rys. 1. Schemat ideowy zegara

WYKAZ ELEMENTÓW**Rezystory**

R3, R27: 0 Ω

R6: 100 kΩ

R1, R7, R9, R10, R30...R40: 10 kΩ

R2: 1 kΩ

R8: 1 MΩ

R5, R41: 22 kΩ

R28: 50 Ω

POT2: 10 kΩ (potencjometr montażowy)

Kondensatory

C1...C24, C30...C39: 100 nF (MKSE)

C27, C28: 10 nF (MKSE)

C16, C17, C21, C33: 10 pF

C26: 10 μF/25 V

C22, C29: 1 μF/25 V

C10, C12: 220 μF/25 V

C9, C15: 47 μF

Półprzewodniki

U1: STM32D103R (TQFP64)

U4: LD33CV

U7: 7805

U19: ST3232

T1: BC337

Inne

X1: rezonator 8 MHz

X2: rezonator 32768 Hz

BT1: podstawka baterii CR2032

G5: gniazdo zasilania

JTAG: wtyk IDC20 do druku

J20: złącze do druku DSUB9/F

BUZ1: buzzer 5 V

RESET: przycisk do druku

Listwy goldpin

ro połowa sukcesu – trzeba jeszcze zaprogramować procesor.

Wgrywanie oprogramowania (ISP In system Programming)

Mogę zaryzykować twierdzenie, że sposobów zapisu programu wewnątrz mikrokontrolera jest tak wiele, że praktycznie programista ograniczony jest tylko własną wyobraźnią. Każdy może napisać własny program, który wgra właściwy program aplikacyjny używając do tego celu dowolnego interfejsu procesora. No tak, ale pojawia się pytanie: jak wgrać pierwotny program czyli tzw. bootloader? Dotychczas jedyną możliwością było zastosowanie interfejsu JTAG. Firma STM poszła o krok dalej i dała możliwość wgrania bootloadera również za pomocą interfejsu UART. Jest to bardzo wygodne w przypadkach, w których budowane jest urządzenie, dla którego krytyczna jest ilość wyprowadzeń i nie można poświęcić kilku pinów procesora do podłączenia JTAG. Gdy stosowany jest UART, to do wgrania wystarczy dwie nóżki procesora, które podczas normalnego użytkownika urządzenia mogą pełnić rolę interfejsu RS232.

Ta metoda ma jednak pewne ograniczenie. Bootloader może być wgrywany tylko za pomocą USART1 (doprowadzenia PA8 i PA9). Zostało to ustalone przez producenta i zapisane w firmwarze procesora. Dlatego też nie można go uszkodzić ani wykasować i jest on równie niezawodny, co złącze JTAG. Żeby procesor uruchomił bootloader należy na do-

prowadzenie BOOT0 podać napięcie 3,3 V, a BOOT1 zewrzeć do masy. W takiej konfiguracji, po zresetowaniu procesor skoczy do adresu, pod którym umieszczony jest bootloader.

Aby wgrać program pierwotny należy zewrzeć zworkę JP6 podłączoną do BOOT0 procesora. Następnie przylutować krótki odcinek przewodu z jednej strony do masy, a z drugiej do nóżki 22 złącza wyświetlacza (jest ono podłączone do BOOT1). Po tym można włączyć zasilanie mikrokontrolera. Powinien uruchomić się wewnętrzny bootloader. Następnie łączymy płytke z interfejsem RS232 komputera PC i uruchamiamy program o nazwie „FLASH LOADER”. Można go pobrać ze strony firmy ST lub skopiować z płyty CD_EP09/2009B.

Kolejną, nieco bardziej złożoną metodą jest wgranie programu pierwotnego za pomocą JTAG. Na płycie PCB przewidziano miejsce na wtyk IDC20 służący do podłączenia JTAG-a. Zajmuje ono aż 5 doprowadzeń procesora. Wybór programatora z interfejsem JTAG pozostawiono użytkownikowi, ponieważ prawie każde środowisko programistyczne wspiera jakiś własny programator i często oferowane jest wraz z nim. Zastosowanie takiego kompletu jest zwykle bardzo wygodne, ponieważ możliwe staje się nie tylko wgrywanie programów, ale również debugowanie programu.

Można również rozważyć zastosowanie złącza SWD (*Serial Wire Debug*). Jest to nowoczesny sposób na debugowanie procesora, wymagający minimalnej liczby połączeń. Do pracy potrzebuje tylko dwóch sygnałów zegara oraz dwukierunkowego złącza danych. Niestety, to rozwiązanie dostępne jest na razie tylko z płatnymi środowiskami, takimi jak R-Link sprzedawany przez Raisonance oraz J-LINK sprzedawany przez IAR Systems. Najprostszy, niemalże darmowy Wiggler nie ma możliwości dwuprzewodowej komunikacji z rdzeniem Cortex-M3.

Jeśli już zdecydowaliśmy się na sposób wgrania programu, to nadszedł czas wyboru odpowiedniego programu pierwotnego lub całkowitej rezygnacji z IAP na rzecz pracy z zastosowaniem programatora JTAG.

Wgrywanie oprogramowania (IAP In Application Programming)

Oprogramowanie może wgrywać i uruchamiać inne oprogramowanie... Skądś to znamy, nieprawdaż? Brzmi to jak zapowiedź pisanie wirusa. Ta możliwość jest bez wątpienia bardzo ciekawa, ale warto się zastanowić nad ograniczeniami. Zaczniemy od organizacji pamięci w procesorze STM32.

STM32 ma architekturę 32-bitową. Wielkość przestrzeni adresowej to 4 GB. To naprawdę dużo zważywszy, że cała pamięć RAM i FLASH to niewiele ponad 0,5 MB. Procesor STM32 po włączeniu zasilania wykonuje instrukcje spod adresu 0x8000000.

Ten unikalny adres to początek pamięci FLASH. Pod tym adresem powinien być zapisany program, bo od tego miejsca rozpocznie się jego wykonanie. Jeśli w programie procesor napotka instrukcję skoku, to skacze do podanego adresu i zaczyna wykonywać kolejne instrukcje. Wszyscy zapewne znamy tą elementarną zasadę działania procesora. Pozwoliłem sobie ją przypomnieć, ponieważ jest ona potrzebna do zrozumienia jak działa IAP. Pamięć flash w procesorze STM32 zorganizowana jest w postaci strony o wielkości 1 kB lub 2 kB, zależnie od wersji. Procesor pozwala na swobodną jej obsługę. Można skasować całą stronę, a następnie zaprogramować ją. Po zakończeniu zapisu pamięci należy wykonać rozkaz skoku do obszaru wgrzanego i procesor zaczyna wykonywać wgrany kod programu. Tak naprawdę to powinno wystarczyć do zrozumienia zasady działania bootloadera. Trzeba tylko uważać aby nie uszkodzić własnej aplikacji.

Firma STM udostępnia bootloader, który wgrywa program za pomocą interfejsu USART1. Program jest bardzo dobrze udokumentowany i ma kompletne źródła. Bazując na przykładzie można program rozbudować lub napisać własny. Kolejnym krokiem jest wybór odpowiedniego środowiska programistycznego.

Środowisko programistyczne (IDE Integrated Development Environment)

W poprzednim rozdziale opisaliśmy wszystkie możliwości wgrania kodu wynikowego. Teraz czas na stworzenie programu. Do tego celu potrzebujemy kompilator, edytor tekstu oraz sterowniki do obsługi programatora. Wszystkie te elementy (kompilator, edytor tekstu oraz obsługa programatora), dostarczone są jako środowisko programistyczne. Przystosowanie każdego środowiska pod nową platformę sprzętową jest zadaniem trudnym i nie warto tego robić samemu. W tym miejscu przydaje się wsparcie producenta danego pakietu.

Moim zdaniem do najlepszych IDE przeznaczonych dla STM32 należą pakiety programów następujący producentów:

- RIDE7 firmy Raisonance.
- RVMDK firmy Keil.
- EWARM firmy IAR Systems.
- HITOP firmy Hitex.

Wszystkie wymienione środowiska są wspierane przez biblioteki dostarczane przez firmę STM. Każdy przykładowy projekt ma oprócz kodu źródłowego również pliki konfiguracyjne dla tych środowisk. Każdy przykład jest od razu gotowy do kompilacji. Takie przykłady są bardzo pomocne, szczególnie jeśli w projekcie niewystarczająca jest sama zmiana kodu. Często w projektach obsługi np. kontrolerów pamięci, konieczna jest również odpowiednia modyfikacja plików linker. Przy wyborze innych środowisk jesteśmy

skazani na samodzielne, żmudne konfigurowanie środowiska programistycznego.

Wybór odpowiedniego środowiska pozostawiam Czytelnikowi. Jestem pewien, że istnieje jeszcze wiele innych środowisk, które mogą współpracować z procesorami STM32. Powyższy projekt został napisany w udostępnianym przez firmę Raisonance darmowym środowisku RIDE7. Płacimy dopiero wtedy jeśli chcemy używać debugera.

Do obsługi popularnego Wigglera służy program HJTAG.

Platforma sprzętowa (PCB)

Projekt „Zegar „dla szachistów” został wykonany w oparciu o mikrokontroler STM32F103R i może posłużyć jako zestaw startowy do dowolnych projektów wykorzystujących wyświetlacz graficzny oraz interfejs RS232. Na płytce dostępne są również doprowadzenia (J4) interfejsów SPI, SDIO, ADC oraz DAC. Na szczególną uwagę zasługuje interfejs SDIO, który umożliwia podłączenie kart microSD w trybie 4 przewodowym. Wszyscy na pewno docenią również bardzo wygodny dostęp do przetworników A/C oraz C/A. Wszystkie te cechy czynią projekt uniwersalnym zestawem startowym, który bardzo łatwo dostosować do własnych potrzeb.

Zasilanie

Płytkę zasilana jest napięciem stałym o wartości 9...12 V. Do jego podłączenia służy gniazdo G5. Przed podłączeniem należy sprawdzić czy polaryzacja napięcia z zasilacza jest prawidłowa. Stabilizator U7 dostarcza napięcie zasilające 5 V, natomiast U4 3,3 V. Kondensatory C18...C20 oraz C26...C29, są zalecane przez STM. Ich zadaniem jest filtrowanie napięcia podawanego na układ zasilania procesora.

Otoczenie procesora

Procesor do pracy potrzebuje rezonatora kwarcowego. Na płytce zastosowano kwarc o częstotliwości 8 MHz, który pracuje w układzie generatora częstotliwości odnie-

sienia dla pętli PLL. Rdzeń mikrokontrolera taktowany jest sygnałem o częstotliwości 72 MHz.

Płytkę wyposażono w przycisk zerowania RESET. Układ zerowania jest sprzęgnięty ze złączem JTAG, który dzięki temu może wyzerować procesor programowo. Na płytce zamontowano też gniazdo baterii litowej CR2032. Służy ona do podtrzymania pracy wewnętrznego zegara.

Wspomniana wcześniej zworka JP6 służy do zmiany metody uruchamiania układu i jest potrzebna do uruchomienia wewnętrznego bootloadera.

Klawiatura

W układzie można zastosować dowolną klawiaturę, którą podłączamy pod złącze J3. Stosowanie układu filtrowania zakłóceń nie jest konieczne, chociaż upraszcza programową obsługę klawiatury.

Buzzer

Układ buzzera jest podłączany za pomocą zwory J2 do portu PA4. Zwora była praktyczna ze względu na założenie wyprowadzenia kompletu sygnałów interfejsu SPI podłączonego do złącza J4.

Wyświetlacz

W układzie zastosowano monochromatyczny wyświetlacz graficzny o rozdzielczości 240×128 pikseli. Wyświetlacz ma kontroler Toshiba T6964C. Interfejs wyświetlacza wymaga przesyłania 8 bitów danych, WR, RD, CE (można podłączyć na stałe) oraz C/D (informuje on kontroler o tym, czy przesyłane są dane instrukcja). Stan doprowadzenia 22 wyświetlacza steruje wielkością czcionki generowaną przez wewnętrzny generator znaków. Do wyboru jest czcionka 8×8 lub 8×6 pikseli. Kontroler wyświetlacza LCD może być zerowany na dwa sposoby: przez procesor (J1: 1–2) lub za pomocą układu RC zbudowanego z C22-R41 (J1: 2–3). Potencjometr POT3 służy do regulacji kontrastu. W przypadku wyświetlacza użytego w projekcie podanie na V0 napięcia o wartości około –14 V pozwalało na czyste wyświetlanie znaków. Napięcie ujemne

zostało podłączone do generatora napięcia ujemnego znajdującego się na wyświetlaczu i podzielne za pomocą potencjometru POT3. Złącze wyświetlacza jest dostosowane tylko do jednego ze standardów wyprowadzeń wyświetlaczy. Niestety, w przypadku wyświetlaczy graficznych istnieją duże różnice w sposobach wyprowadzenia sygnałów, więc przed zakupem warto sprawdzić, czy dany LCD ma wyprowadzenia kompatybilne z płytką.

Interfejsy komunikacyjne

Układ ma złącze J4, na którym dostępne są interfejsy procesora. Oprócz tego do złącza doprowadzone są również napięcia 3,3 V oraz 5 V, które można wykorzystać do zasilania układów dołączanych z zewnątrz.

2×RS232

Układ ma złącze J20 zgodne z RS232 (tylko sygnały RxD, TxD). Za jego pomocą można go podłączyć do komputera PC wyposażonego port szeregowy. Interfejs RS232 jest podłączony wewnętrznie do doprowadzeń PA8 oraz PA9 i może służyć do zaprogramowania wewnętrznego bootloadera ISP lub bootloadera IAP. Oba dostępne są na stronie internetowej firmy ST. Do doprowadzeń 1 i 2 złącza J4 podłączono drugi interfejs szeregowy. Jego napięcia są zgodne ze standardem RS232.

USB

Procesor ma jedno złącze USB. Jego sygnały podano na doprowadzenia 3 i 4 złącza J4. Do doprowadzeń 16 i 17 podłączono, odpowiednio, napięcie zasilające oraz masę


SDIO

Interfejs SDIO służy do obsługi kart SD, które podłączone do niego pracują szybciej i pewniej niż przy współpracy poprzez SPI. Oprócz kart SD możliwe jest także stosowanie kart microSD, które nie zawsze mają tryb SPI.

SPI

Interfejs SPI jest bardzo dobrze znanym interfejsem szeregowym. Można go użyć do współpracy z różnymi układami zewnętrznymi.

R E K L A M A

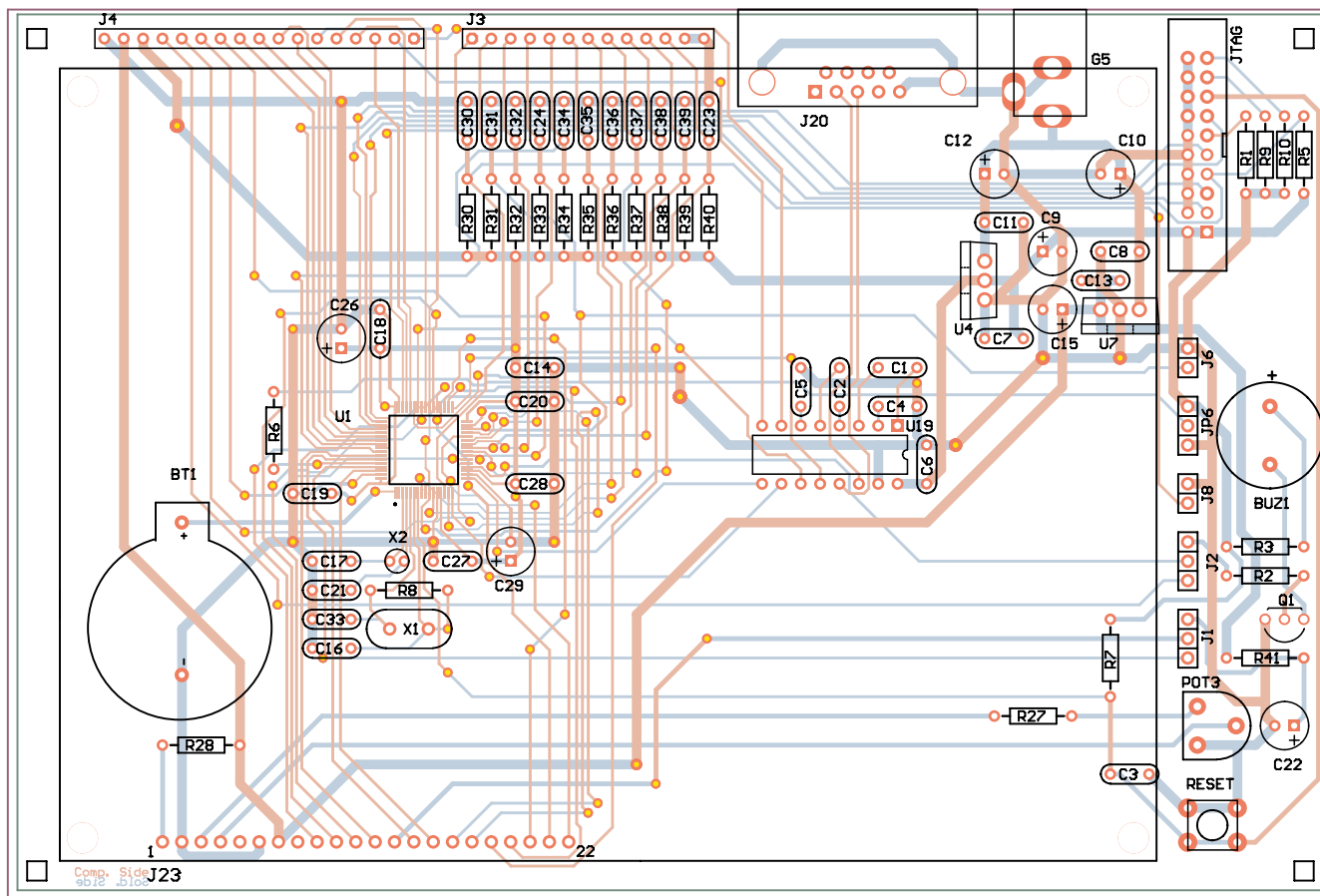


przełączniki dźwigniowe
przełączniki klawiszowe
przełączniki klawiszowe podświetlane
przełączniki hebelkowe

przełączniki krańcowe
przełączniki z bezpiecznikiem
przyciski
przełączniki dip-switch
przełączniki tact switch




www.micros.com.pl



Rys. 2. Rozmieszczenie elementów na płycie zegara

ADC oraz DAC

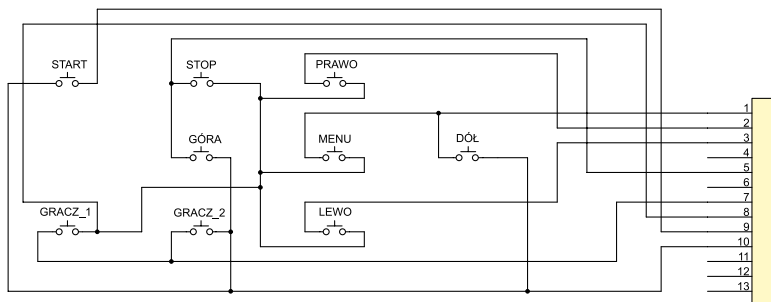
Wszystkich, którzy potrzebują wygodnego sposobu na sprzęgnięcie procesora z układami analogowymi, z pewnością ucieszy bardzo wygodny dostęp do interfejsów ADC oraz DAC.

Montaż i uruchomienie

Schemat ideowy zegara umieszczono na rys. 1, natomiast montażowy na rys. 2. Na rys. 3 umieszczono schemat przykładowej klawiatury. Montaż układu nie powinien sprawić problemów. Płytkę PCB zaprojektowano specjalnie z myślą o montażu przez osoby nieposiadające specjalistycznego sprzętu. Wystarczą zwykła lutownica i obcinaczki do drutu. Jedynym problemem może być przylutowanie samego procesora, który dostarczany jest w obudowie TQFP64.

Uruchomienie rozpoczynamy od wgrania programu aplikacyjnego umieszczonego na płycie CD_EP9/2009B. Skompilowany program znajduje się w pliku *Zegar.hex* lub *Zegar.bin*. Program można wgrać za pomocą programatora JTAG lub interfejsu RS232. Opis tej czynności podano wcześniej.

Poprawnie zmontowany zegar działa natychmiast po załączeniu zasilania.



Rys. 3. Schemat ideowy klawiatury

Program

Program napisano w języku C za pomocą środowiska RIDE7 firmy Reasonance. Do pamięci mikrokontrolera wgrano go za pomocą programatora Wiggler. Do obsługi programatora służy darmowy program HJTAG.

Program wyświetla dwa zegary, które podają czasy dla poszczególnych graczy. Po naciśnięciu „Menu” program umożliwia dokonanie wyboru jednego z trzech trybów gry (całkowity, pojedynczy oraz tryb Fishera). Możemy również ustawić imiona graczy oraz czasy rozgrywek. Program pamięta parametry po wyłączeniu zasilania. Menu jest proste i przejrzyste, podobnie

jak wyświetlane komunikaty. Obsługa urządzenia nie powinna nastręczać żadnych trudności.

Niestety, z własnego doświadczenia wiem, że prawie żaden układ nie działa od razu po załączeniu zasilania. Nie należy się tym zniechęcać. Czasami od celu ważniejsze jest to, przez co się przechodzi, a rozwiązując problemy nabywa się wiedzy i doświadczeń. Nagrodą będzie uruchomienie płytki z dużym, czytelnym wyświetlaczem, z klawiaturą oraz powiadomieniem dźwiękowym. Korzystając z tej platformy można napisać wiele własnych programów użytkowych.

Jarosław Ośka,
jaroslaw.oska@gmail.com