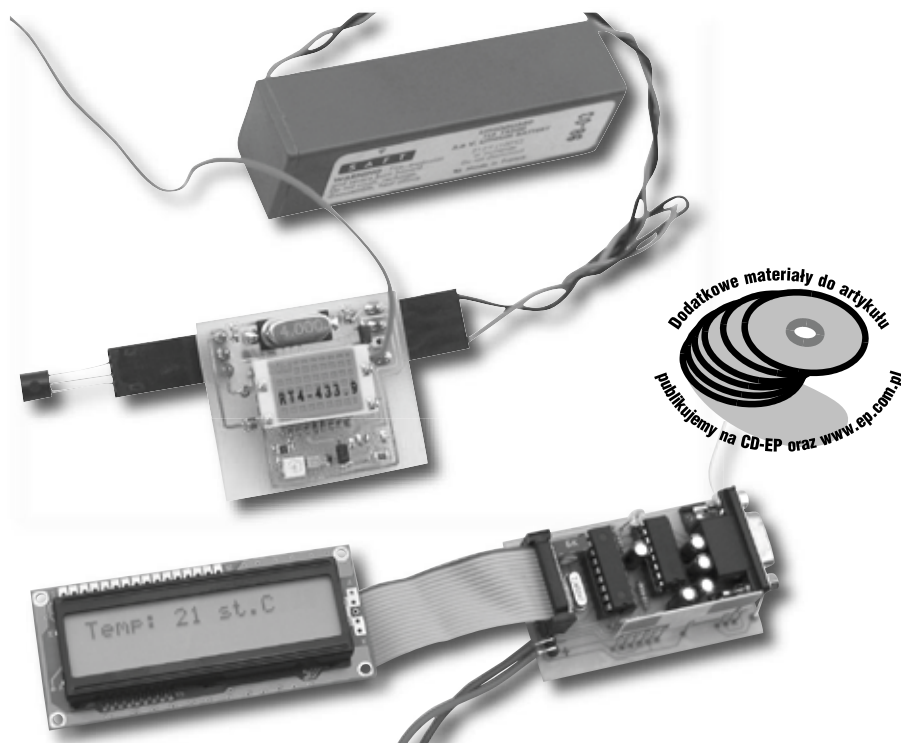


Bezprzewodowy rejestrator temperatury

AVT-542

W artykule autor prezentuje bezprzewodowy system odczytu temperatury mierzonej przez czujnik. Jest to prosty i łatwy do wykonania układ zrealizowany w celu zademonstrowania, że cały świat nie kończy się na '51 i Bascomie.

Rekomendacje: niezależnie od motywacji autora urządzenie spełnia swoją rolę i może być polecone wszystkim borykającym się z problemem pomiaru temperatury w warunkach, w których realizacja odczytu przewodowego jest niewygodna lub niemożliwa.



Urządzenie opisane w artykule powstało z dwóch powodów: pierwszym była potrzeba bezprzewodowego pomiaru temperatury kolejno w kilku miejscach obiektu. Drugą, determinującą sam wybór mikrokontrolera i tworzenie jego firmware'u, był wstępniak sierpniowego numeru EP (z roku 2003), w którym wypowiedziano opinię całkowicie zgodną ze zdaniem autora tego projektu, na temat sytuacji szeroko pojętej elektroniki w Polsce. Postawiono za cel udowodnienie, że bez Bascoma i bez 8051 można szybko i efektywnie stworzyć układ oparty na mikrokontrolerze i wykorzystujący peryferia, które tak łatwo jest oprogramować w Bascomie.

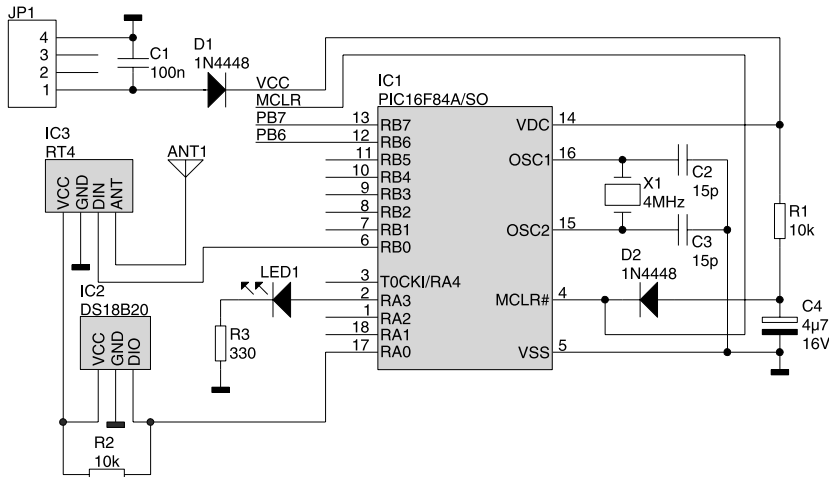
Opis układu

Bezprzewodowy termometr składa się z nadajnika (schemat elektryczny pokazano na **rys. 1**) z czujnikiem temperatury DS18B20 oraz odbiornika (schemat elektryczny pokazano na **rys. 2**) z interfejsem RS232 oraz możliwością podłączenia wyświetlacza alfanumerycznego (HD44780). Nadajnik komunikuje się z odbiornikiem za pośrednictwem hybrydowych

modułów radiowych Telecontrolli o częstotliwości nośnej 433 MHz. Odświeżenie pomiaru temperatury następuje w konfigurowalnych w pewnych granicach odstępach czasu. W układzie modelowym wynosił on ok. 10s. Aby zapewnić pełną mobilność układu nadajnika zasilany jest on z baterii. Użyto litowej baterii 3,6V stosowanej do podtrzymania zegara BIOS-u komputera.

Nadajnik

Schemat elektryczny nadajnika pokazano na **rys. 1**. Najważniejszym elementem jest mikrokontroler PIC16F84A (użyto przemysłowej wersji układu ze względu na szeroki zakres pomiarowy układu DS18B20) taktowany kwarem 4MHz. Do wprowadzenia 0 portu A mikrokontrolera podłączono układ DS18B20 z rezystorem podciągającym linię danych do plusa zasilania. Producent czujnika zaleca stosowanie rezystora o wartości 4,7kΩ jednak w celu obniżenia konsumpcji energii przez układ nadajnika zastosowałem rezystor 10kΩ. Przeprowadzone testy nie wykazały żadnych problemów w związku z tą modyfikacją. Do wypro-



Rys. 1. Schemat elektryczny nadajnika

wadzenia 0 portu B dołączono hybrydowy nadajnik RT4. Wybór tego układu został wymuszony dość niskim napięciem zasilania całego układu, a jak wynika z noty katalogowej, RT4 może pracować w zakresie napięć od 2 do 14V. Jedynym mankamentem RT4 jest konieczność stosowania zewnętrznej anteny. Jako, że z założenia układ nadajnika miał mieć małe gabaryty zdecydowano na wykonanie go wyłącznie z elementów SMD, co spowodowało konieczność programowania mikrokontrolera w systemie. PIC16F84A wspiera tę metodę programowania, nazwaną przez Microchipa *In Circuit Serial Programming*. Z programowaniem ICSP powiązany jest dość nietypowy obwód zerowania. Konieczne było zastosowanie diody D2 separującej linię MCLR od reszty układu w momencie programowa-

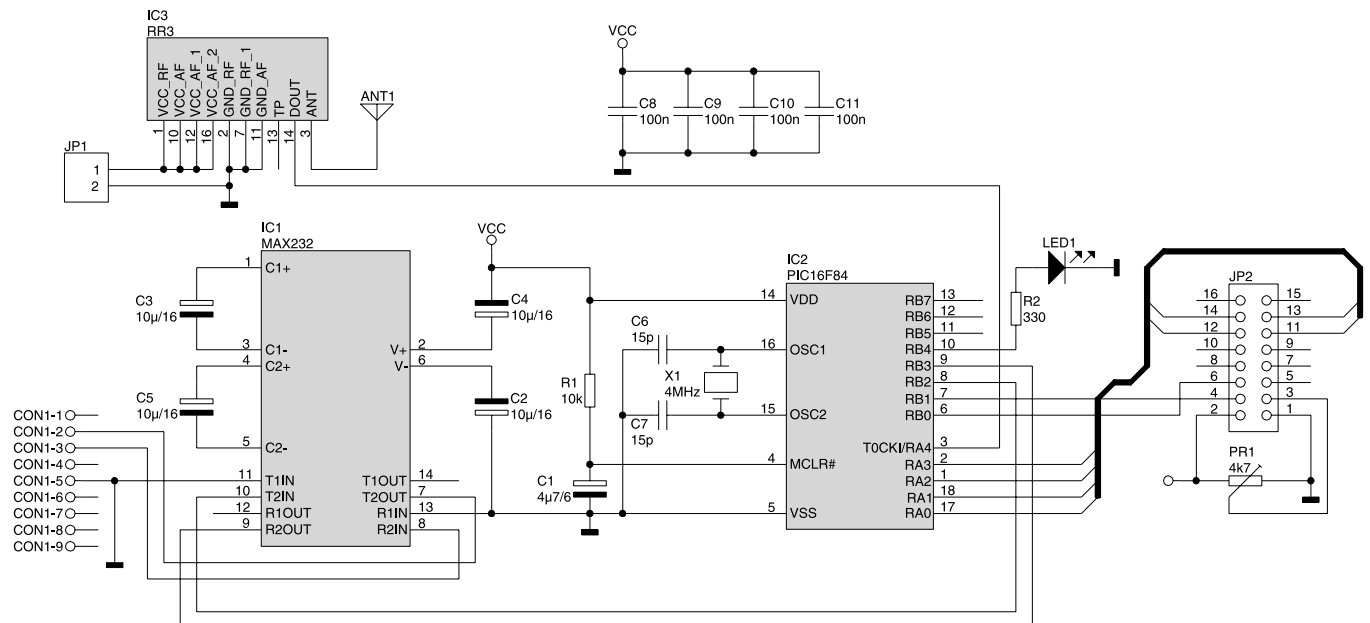
nia. Dioda D1 służy do oddzielenia obwodu zasilania mikrokontrolera w przypadku, gdy podczas programowania napięcie zasilające pobierane jest z programatora. Ma to znaczenie wówczas gdy stosujemy programatory zasilane bezpośrednio z portów komputera (np. JDM), których wydajność prądu jest niewystarczająca do zasilenia całego układu. W praktyce najpewniejszym rozwiązaniem, dającym 100% poprawność programowania okazało się stosowanie zewnętrznego zasilania całego układu. Na płytce drukowanej nadajnika znajdują się pola lutownicze służące do programowania w systemie. Dioda LED w nadajniku sygnalizuje rozpoczęcie kolejnej transmisji danych. Antenę, tak nadajnika, jak i odbiornika stanowi odcinek przewodu miedzianego o długości około 17 cm.

Odbiornik

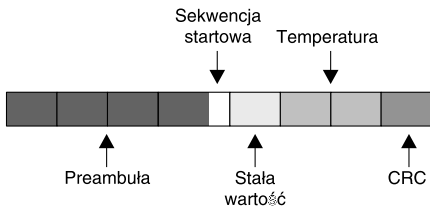
Schemat elektryczny odbiornika przedstawiono na rys. 2. Podobnie jak w przypadku nadajnika, odbiornikiem steruje PIC16F84A, tym razem w obudowie DIL. Za konwersję poziomów logicznych przy transmisji szeregowej pomiędzy odbiornikiem a komputerem odpowiada MAX232, a za odbiór danych z nadajnika moduł RR3. Złącze służy do podłączenia opcjonalnego wyświetlacza LCD (HD44780), a potencjometr montażowy PR1 umożliwia regulację jego kontrastu. Dioda LED sygnalizuje poprawny odbiór ramki danych. Całość zamontowana jest na jednostronnej płytce drukowanej, z montażem przewlekany. Podobnie jak w nadajniku, mikrokontroler jest taktowany z częstotliwością 4MHz. Układ zasilany jest z zewnętrznego zasilacza 5V. Możliwe jest stosowanie napięć zasilających w przedziale od 3 do 5V, jednak gdy jest używany wyświetlacz LCD, napięcie zasilania powinno wynosić 5V. W modelowym urządzeniu napięcie to musiało być wyższe niż 4V ze względu na stałe podświetlenie użytego wyświetlacza.

Program(y)

Algorytm działania tak nadajnika, jak i odbiornika jest następujący. Po włączeniu zasilania mikrokontroler znajduje się w trybie sleep (ze względu na mały pobór prądu, wynoszący w tym trybie około 7µA), z którego cyklicznie wychodzi co zadany (za pomocą timera-watchdog) czas, po czym przesyła do układu



Rys. 2. Schemat elektryczny odbiornika



Rys. 3. Budowa ramki danych

DS18B20 komendę ustalenia rozdzielczości przetwornika na 11 bitów (domyślnie jest to 12 bitów, jednak dla zminimalizowania poboru energii przez układ zdecydowano się na rozdzielczość 11-bitową, co skróciło czas konwersji o połowę tzn. do 375 ms) oraz rozpoczęcia konwersji (44h) temperatury na postać cyfrową. Jest to możliwe jedynie w ustalonej sekwencji zdefiniowanej dla magistral 1-wire (inicjalizacji, komenda ROM-u, komenda operacji). Po wydaniu komendy konwersji mikrokontroler przechodzi znów w tryb *sleep* na około 0,8s (czas potrzebny na przeprowadzenie przez DS18B20 konwersji temperatury na słowo 12-bitowe). Po „przebudzeniu” wysyłana jest komenda odczytu zawartości rejestrów DS18B20 (BEh). Odczytuje się jedynie dwa pierwsze bajty, zawierające informacje o temperaturze. Następnie formowana jest ramka danych wysyłana do odbiornika. Strukturę ramki przedstawiono na rys. 3.

Transmisja realizowana jest z użyciem kodowania Manchester, co oznacza w praktyce, że logiczna jedynka transmitowana jest jako sekwencja 10, natomiast logiczne zero jako 01 (kod (2n,n)). Preambuła składa się z czterech bajtów o wartości FFh. Bezpośrednio po niej jest trans-

mitowana sekwencja startowa, informująca kontroler aby rozpoczął odczyt danych z odbiornika. Sekwencją tą jest stan wysoki trwający przez okres dwóch bitów. Informacja ta jest dla kontrolera jednoznaczna, ze względu na kod Manchester, w którym taka sytuacja nigdy nie wystąpi i jest interpretowana jako nadchodząca nowa ramka danych. Odbiornik czeka w pętli na tę sekwencję i po jej zakończeniu rozpoczyna odbiór pierwszego bajtu ramki, którym jest ustalona wcześniej i nigdy nie zmieniana wartość (w modelu było to 73). Jeśli odbiornik stwierdzi że wartość ta jest inna, to cała ramka zostaje odrzucona. Kolejne miejsca w ramce zajmują kolejno mniej i bardziej znaczący bajt wartości temperatury, a po nich znajduje się jednobajtowa suma kontrolna.

Odbiornik po odebraniu ramki oblicza sumę kontrolną z jej trzech pierwszych bajtów i porównuje ją z czwartym bajtem ramki, czyli z odebraną CRC. Jeśli obie wartości są identyczne, cała ramka (oprócz preambuły i sekwencji startowej) zostaje przetransmitowana do portu szeregowego.

Na koniec cyklu wartość temperatury zostaje obliczona w kontrolerze (z dokładnością do jedności) i wysłana do wyświetlacza LCD.

Aplikacja

Program zbierający wyniki przesyłane z odbiornika do komputera został „napisany” w LabVIEW 5.1. Pomimo tego, iż aktualna wersja LabVIEW nosi numer 7.1, to na potrzeby rejestratora wersja 5.1 jest w

WYKAZ ELEMENTÓW

Nadajnik

Rezystory

- R1,R2: 10kΩ
- R3: 330Ω

Kondensatory

- C1: 100nF
- C2, C3: 15 pF
- C4: 4,7µF/6V

Półprzewodniki

- D1, D2: 1N 4448
- LED1: LED SMD0805
- IC1: PIC16F84A/SO (zaprogramowany)
- IC2: DS18B20
- IC3: RT4 Telecontrolli

Różne

- JP1: Goldpin 4x1
- X1: rezonator kwarcowy 4MHz

Odbiornik

Rezystory

- R1: 10kΩ
- R2: 330Ω
- PR1: 4,7kΩ

Kondensatory

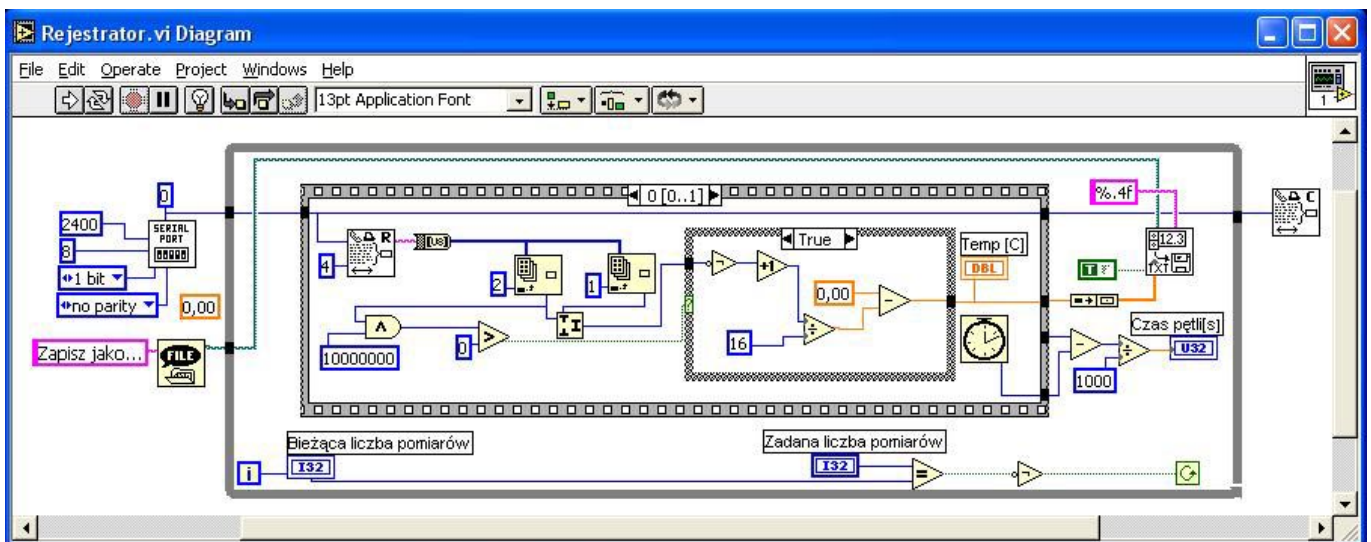
- C1: 4,7µF/6V
- C2...C5: 10µF/16V
- C6,C7: 15pF
- C8...C11: 100nF

Półprzewodniki

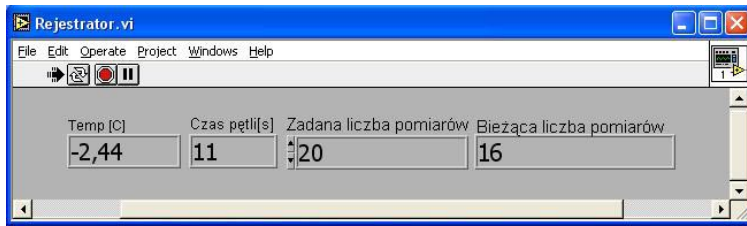
- LED1: LED 3mm
- IC1: MAX232
- IC2: PIC16F84 (zaprogramowany)
- IC3: RR3 Telecontrolli

Różne

- JP1: Goldpin 2x1
- JP2: Goldpin 8x2
- X1: Rezonator kwarcowy 4MHz



Rys. 4. Kod aplikacji przygotowanej za pomocą programu LabView



Rys. 5. Wygląd panelu czołowego aplikacji

zupełności wystarczająca. Kod programu przedstawiono na **rys. 4**.

Po inicjalizacji portu szeregowego (ikona *Serial Port Init.vi*) i wybraniu pliku do którego będą zapisywane dane z rejestratora, program wchodzi w pętlę główną, w której komponent *Serial Port Read.vi* oczekuje na cztery bajty danych. Po odebraniu ramki komponent *Read* formuje wyjściowy strumień stringów, który następnie konwertowany jest to postaci tablicy bajtów. Tablica ta zostaje następnie dwukrotnie zaadresowana w celu wyłuskania dwóch środkowych bajtów niosących informacje o temperaturze. Następnie te bajty zostają połączone w jedną 16 – bitową zmienną. Najstarszy bit testowany jest na obecność jedyńki w celu zidentyfikowania znaku tempe-

ratury. Identyfikacja ta decyduje, który z wariantów struktury *case* zostanie zastosowany do zdekodowania wartości temperatury. Na **rys.4** jest widoczny przypadek, kiedy mamy do czynienia z liczbą ujemną (wartość temperatury zapisana jest w kodzie U2), liczba ta zostaje zanegowana, a następnie zwiększona o jeden. Ponieważ odczytujemy wartość temperatury z dokładnością do 4 cyfr po przecinku, uzyskana wartość zostaje podzielona przez 16 (dziesiętnie), a następnie odjęta od zera. Tak zdekodowana liczba zostaje wysłana na panel użytkownika (**rys.5**) jako wskaźnik o nazwie *Temp* (z precyzją dwóch miejsc po przecinku). W tym samym czasie wartość temperatury zostaje dopisana do wybranego wcześniej pliku. Cała pętla powtarza się zadaną na pa-

nelu czołowym liczbą cykli. Program należy uruchamiać w cyklu jednokrotnym (strzałka w prawo). Do przerwania działania programu w dowolnym momencie służy czerwony przycisk po prawej stronie strzałki. Po zakończeniu komponent *Serial Port Close.vi* zamyka port szeregowy komputera. Widoczne komponenty timera służą do obliczania czasu trwania kolejnej iteracji, a więc odstępu od poprzedniego pomiaru.

Uwagi końcowe

Układ był testowany w całym dopuszczalnym dla DS18B20 zakresie temperatur. Pracował stabilnie zarówno przy temperaturze -20°C , jak i $+125^{\circ}\text{C}$ (w tej temperaturze umieszczony był jedynie czujnik). Należy jednak pamiętać, że błąd pomiarowy DS18B20 w takim zakresie temperatur wynosi $\pm 2^{\circ}\text{C}$, a zapewniana przez producenta dokładność $\pm 0,5^{\circ}\text{C}$ obowiązuje jedynie w przedziale od -10 do $+85^{\circ}\text{C}$. Nadajnik pracuje poprawnie z napięciem zasilania od 3 do 5V.

Marcin Chruściel
chrusciel2@wp.pl

Szukasz niezawodnej, stałej partnerki, której możesz ufać? Oto ona.

Przetwornica DATEL DC/DC

ROPLA®

Przetwornica DATEL DC/DC doskonale nadaje się do układów telekomunikacyjnych, elektroniki medycznej oraz do zasilania systemów LEDowych. Wykorzystuje ją wojsko, medycyna i przemysł produkujący urządzenia o wysokim stopniu pewności działania i niezawodności.



- - niezawodne źródło napięcia o bardzo dużej stabilności,
- - możliwość uzyskania dużej mocy z niewielkiej powierzchni,
- - izolacja galwaniczna od układu wykonawczego,
- - kompatybilne wyprowadzenie w stosunku do innych producentów,
- - wykonywana w technologii wielowarstwowej - do 16 warstw,
- - niska awaryjność (ppm)

Ropla Elektronik Sp. z o.o., 53-011 Wrocław, ul. Wyścigowa 3, tel. +48 (71) 339 7229, fax. +48 (71) 339 7230, info@roppla.pl, www.roppla.pl

Przed wszystkim kondensatory...