Komputerek eMeSPek

Proste eksperymenty

Otrzymaliśmy sporo ciekawych projektów w naszym konkursie na aplikacje z mikrokontrolerem MSP430. Cztery najwyżej ocenione projekty przedstawiamy na CD. Oprócz tego Czytelnicy mają okazję zapoznać się z dwoma innymi projektami wykonanymi na płytce eMeSPek. Te dość proste eksperymenty mają tę cechę, że nie wymagają dołączania do płytki elementów zewnętrznych, poza wyświetlaczem alfanumerycznym, jaki każdy eksperymentator na pewno znajdzie w szufladzie.

Pierwszym projektem jest termometr pokojowy. Wykorzystuje on wbudowany w moduł przetwornika analogowo-cyfrowego czujnik temperatury. Czujnik półprzewodnikowy, przewidziany jedynie do kontroli temperatury struktury procesora, nie posiada wystarczającej precyzji ale po kalibracji jednopunktowej (np. w temperaturze pokojowej od dowolnego innego termometru) zapewnia akceptowalną dokładność wskazania.

Drugi projekt to dzwonek do drzwi z pozytywką. To wyjątkowo prosta aplikacja, której zadaniem jest przekonanie Czytelników do eksperymentowania z energooszczędnością. Dzwonek z procesorem MSP430 zasilany jest z własnej baterii typu CR2032 i jest na tyle energooszczędny, że jego pobór mocy nie jest dużo większy niż straty związane ze zjawiskiem samorozładowania się baterii.

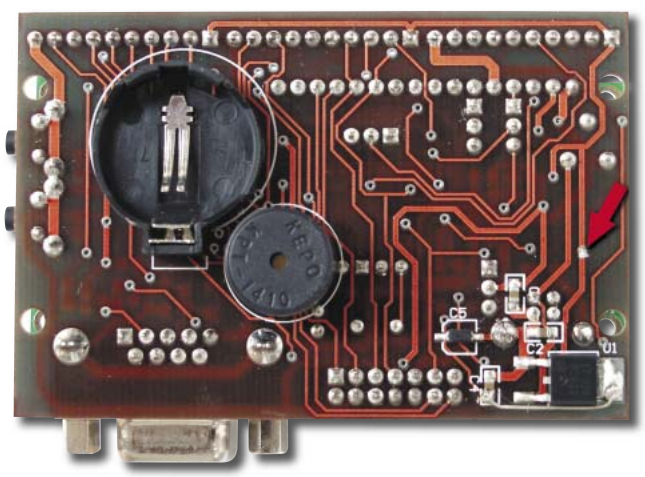
Mikrokontroler MSP430F1232

Na początek kilka słów o mikrokontrolerze, który jest do dyspozycji. MSP430F1232 należy do rodziny mikrokontrolerów o uniwersalnym przeznaczeniu. Jak wszystkie MSP430, posiada 16-bitowy rdzeń typu RISC, interfejs JTAG i mechanizm redukcji poboru mocy. Ponadto wyposażony jest w dwa 16-bitowe liczniki, 10-bitowy przetwornik analogowo-cyfrowy i uniwersalny port szeregowy. Podstawowe parametry mikrokontrolera zawiera **tab. 1**.

Procesor należy do grupy najmniejszych spośród rodziny MSP430. Od większych braci, poza ograniczoną ilością wyprowadzeń, różni się sposobem podłączenia interfejsu JTAG, który współdzieli wyprowadzenia ze standardowymi portami I/O. Może to być nieco kłopotliwe, gdy konieczne jest użycie tych wyprowadzeń do podłączenia układów peryferyjnych. W płytce eMeSPek wyprowadzenia te współdzielone są z liniami danych wyświetlacza alfanumerycznego. Nie oznacza to jednak, że podczas debugowania w systemie nie można korzystać z wyświetlacza. Nie trzeba go też odłączać. Środowisko projektowe umożliwia podczas emulacji (tryb „Go") programowe odłączenie linii JTAG i przypisanie im funkcji standardowej (linii portu lub funkcji specjalnej). Jedyna niedogodność jest taka, że po zatrzymaniu wykonywania programu na pułapce, oprogramowanie automatycznie nie wykrywa tego faktu i nie odświeża danych o stanie procesora. Kiedy użytkownik jest pewien, że procesor zatrzymał się, powinien użyć polecenia Stop. Wtedy połączenie zostanie ponownie nawiązane a dane o stanie procesora wyświetlone na ekranie komputera. W przypadku środowisko IAR Embedded Workbench trzeba zaznaczyć opcję Release JTAG on Go w menu Emulator. W środowisku AQ430 opcję tę umieszczono w menu Debug.

Tab. 1. MSP430F1232 – podstawowe parametry

CPU	16-bit RISC
Pamięć Flash	8 kB (pamięć programu) + 256 B (Info Memory)
Pamięć RAM	256 B
Porty GPIO	22 (14 współpracuje z przerwaniem)
Przetwornik A/C, rozdzielczość, ilość wejść, inne	10-bitów SAR, 8 wejść, czujnik temperatury, pomiar napięcia baterii, referencja 1,5 V, 2,5 V i 1/2 Vcc, autoscan, Direct Transfer Controller
Liczniki	TimerA – 16-bitowy, 3 jednostki capture/compare, WDT – 16-bitowy
System zegarowy	Generator XT1 z rezonatorem 32768 kHz lub rezonatorem kwarcowym/ceramicznym 455 kHz...8 MHz, generator DCO programowalny lub z zewnętrznym rezystorem ustalającym
Port szeregowy	USART – port asynchroniczny/synchroniczny



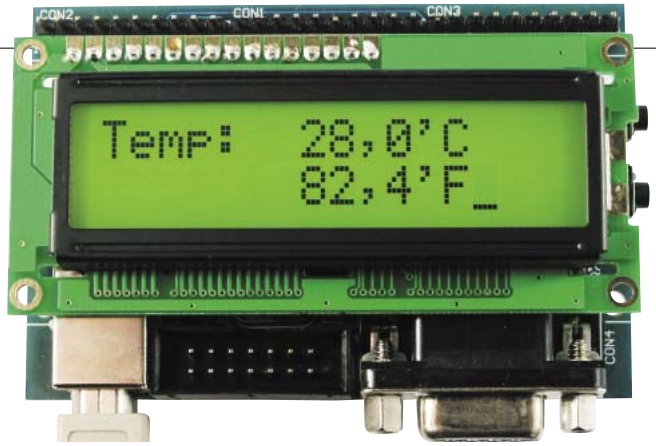
Fot. 1. Poprawki obwodu drukowanego

Modyfikacja płytki eMeSPek

Nim przejdziemy do omówienia projektu, trzeba będzie jeszcze wykonać niewielką poprawkę w mozaice obwodu drukowanego. Należy w tym celu przeciąć ścieżkę drukowaną wskazaną na **fot. 1** i wykonać dodatkowe połączenie przewodowe. Modyfikacja pozwala zworce JP4 pełnić jej funkcję. Nowe połączenie jest szczególnie istotne, ponieważ bez niego stabilizator nie może pracować poprawnie. W większości przypadków może nie ma to większego znaczenia, ale oscylacje w obwodzie zasilania procesora, które wytwarza stabilizator, mogą powodować niestabilność pracy generatora z kwarcem zegarkowym. Efekt zakłóceń jest również widoczny w wynikach pomiarów przetwornika analogowo-cyfrowego.

Termometr z wbudowanym czujnikiem temperatury

Uruchomienie termometru (fot. 1) wymaga odpowiedniego przygotowania płytki eMeSPek. W tym celu należy ustawić zworki JP1 i JP2 w pozycji 1-2, by podłączyć linie wyświetlacza do procesora i ustawić zworkę JP7 w pozycji 2-3, by podłączyć kondensator C10 do wyprowadzenia P2.4 (REFout). Kondensator C10 zostanie dołączony równolegle do wbudowanego w moduł



Fot. 2. Termometr podczas pracy

przetwornika źródła napięcia referencyjnego. Jest on niezbędny dla obniżenia impedancji dynamicznej źródła i zapewnienia odpowiedniej precyzji pomiarów.

Czujnik temperatury to element półprzewodnikowy o dodatnim współczynniku temperaturowym i charakterystyce zbliżonej do opisanej wzorem: Vtemp=3,55×TCelc+986 mV. Tolerancja wartości współczynnika temperaturowego wynosi 3%, co w zastosowaniach domowych jest zupełnie wystarczające. Tolerancja wartości offsetu wynosi 5% i dlatego offset należy skorygować podczas kalibracji jednopunktowej.

Sam przetwornik analogowo-cyfrowy ma interesującą budowę. Umożliwia pomiar pojedynczy lub według zaprogramowanej sekwencji, na wielu kanałach. Dodatkowo wyposażony jest w kontroler DTC – Data Transfer Controller, odpowiedzialny za przesyłanie wyników do RAM-u bez udziału jednostki centralnej. Ciekawostką jest bufor źródła napięcia referencyjnego, konieczny przy szybkich pomiarach i wyłączany dla zaoszczędzenia prądu przy niskiej częstotliwości próbkowania.

Na **list. 1** przedstawiono program termometru. Rozpoczyna się on instrukcją wyłączającą działanie watchdoga. Następnie wykonywana jest konfiguracja przetwornika analogowo-cyfrowego. Przetwornik pracuje w trybie pojedynczego pomiaru, z niskim zegarem taktującym i najdłuższym możliwym czasem sample&hold. Referencja ustalona została na 1,5 V a zewnętrzny kondensator dołączony do obwodu źródła referencyjnego. W przyjętym trybie pracy przetwornik zgłasza przerwanie po zakończeniu pomiaru.

List. 1. Źródło program głównego Termometru

```
  void main( void )
{
  WDTCTL = WDTPW + WDTHOLD;                    // wyłączenie watchdog-a
  // Setup ADC10
  ADC10CTL0 = ADC10ON + ADC10IE + SREF_1 + ADC10SHT_3 + REFON + ADC10SR +
REFOUT;
  ADC10CTL1 = ADC10SSEL_0 + INCH_10 + CONSEQ_0 + ADC10DIV_7;
  LCDInit();                                    // inicjalizacja LCD
  _EINT();                                     // Globalne włączenie przerwań
  while (1)
  {
    ADC10CTL0 |= ENC + ADC10SC;              // włączenie konwersji i start
próbkowania
    _BIS_SR(LPM0_bits);                     // LPM0, wybudzi przerwanie od ADC10
    _NOP();                                   // na wszelki wypadek
    temp=ADC10MEM;
    IntDegF = ((temp - 630) * 7610) >>10;    //  konwersja na oF
    temp=ADC10MEM;
    IntDegC = ((temp - 673) * 4230) >>10;    //  konwersja na oC
    ShowTempOnLCD(IntDegC,IntDegF);          // wyświetlanie temperatury na
LCD
          for(char j = 0; j < 3; j++)         // delay ok 1,8s
            for(unsigned int i = 0; i < 50000; i++);
  }
}
#pragma vector=ADC10_VECTOR
__interrupt void ADC10_ISR (void)
{
    _BIC_SR_IRQ(LPM0_bits);                    // wybudzenie uP
}
```

List. 2. Funkcja odtwarzania dźwięku.

```
void play_note (unsigned short whichnote)
{
  unsigned short duration=0; unsigned short note=0;
  note = whichnote & 0x3fff;             // maska na definicję tonu
  duration = whichnote & 0xC000;         // maska na długość dźwięku
  duration = duration >> 10;             // stopień przesunięcia definiuje tempo
  duration--;                            // dźwięk skrócić, żeby nie był legato
  CCR0 = note;
  while (duration)
  {
     duration--;
     TACTL |= MC0;
     LPM0;                              // Wprowadzenie LPM0 - działa MCLK
  }
  TACTL &=~ MC0;                        // chwila przerwy między dźwiękami
  CCR0=0;                               // (nie gramy legato)
  LPM3;                                 // Wprowadzenie LPM3 - tylko ACLK
}
```

Po inicjacji wyświetlacza alfanumerycznego, wykonywany jest w pętli nieskończonej fragment programu odpowiedzialny za pomiar oraz obliczenie temperatury i prezentację wyników.

Program nie wykorzystuje technik redukcji poboru mocy. Użyto jedynie polecenia LPM0, czyli polecenia zatrzymania CPU na czas pomiaru napięcia na czujniku temperatury. To pożyteczny zabieg, zmniejszający niekorzystny wpływ pracy CPU i magistral na jakość pomiaru. Dzięki temu prostemu zabiegowi uzyskuje się precyzję pomiaru porównywalną z przetwornikami zewnętrznymi.

Dzwonek do drzwi z zasilaniem bateryjnym

W celu uruchomienia projektu należy odłączyć wyświetlacz. Po zakończeniu prób należy również usunąć zworkę JP4 i umieścić baterię w koszyku.

Odtwarzanie dźwięku (melodii) wymaga kontroli co najmniej dwóch parametrów: czasu trwania i wysokości tonu. Oba parametry są pośrednio związane z pomiarem czasu. Metody jego pomiaru mogłyby być dowolne, gdyby nie jedna z zasad budowy energooszczędnych aplikacji: unikanie nadmiernego obciążania CPU, co w pierwszym rzędzie wyklucza odmierzanie czasu w pętli oczekiwania.

Procesor eMeSPeka posiada jeden licznik ogólnego użytku współpracujący z generatorem PWM, którym posłużono się do generowania dźwięku. Czas trwania dźwięku można odmierzyć zliczając okresy PWM, ale można też do tego celu użyć licznika WDT. Watchdog mikrokontrolera pozwala na ograniczone użycie go w formie zwykłego licznika.

Program (**list. 2**) po starcie wykonuje istotną dla ograniczenia poboru mocy operację: ustawienie wszystkich portów w tryb wyjścia. Licznik TimerA przygotowany jest do taktowania z sygnału MCLK, natomiast licznik WDT z sygnału ACLK. TimerA pracuje w trybie zliczania do określonej wartości (wartość wynika z okresu trwania dźwięku) i współpracuje z blokiem *Capture-Compare*. Blok porównujący pracuje w trybie set/reset, w którym generuje krótki pobudzający brzęczyk (o długości trwania zdefiniowanej w rejestrze CCR0).

Melodia zapisana jest w postaci tablicy 16-bitowych wartości, których 12 młodszych bitów definiuje wysokość tonu a 4 starsze bity decydują o wartości nuty.

Program używa polecenia LPM3, wprowadzające mikrokontroler w tryb, w którym aktywny pozostaje jedynie generator z kwarcem zegarkowym. Ogranicza to pobór prądu w trybie spoczynkowym do ok. 2 μA. Podczas odtwarzania dźwięku, kiedy do pracy licznika TimerA potrzebny jest aktywny sygnał taktujący MCLK, używany jest tryb LPM0.

Oba programy przygotowano z wykorzystaniem wersji ewaluacyjnej środowiska IAR EW 5.0 dla MSP430.

Sylwester Nowocień
Mariusz Kaczor