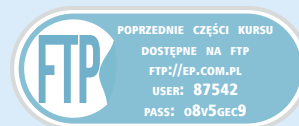


Pierwszy projekt dla mikrokontrolera z rodziny MSP432



W artykule zaprezentujemy sposób rozpoczęcia pracy z mikrokontrolerami z rodziny MSP432 – zainstalujemy odpowiednie oprogramowanie narzędziowe oraz utworzymy pierwszy projekt. Pokażemy też metody optymalizacji poboru mocy przez mikrokontroler MSP432. Za bazę sprzętową posłużą nam zestaw startowy MSP-EXP432P401R LaunchPad.

Nowa seria mikrokontrolerów MSP432 jest obsługiwana przez środowisko programistyczne IDE: Code Composer Studio (wersja 6.1+), IAR Embedded Workbench for ARM (wersja 7.40.2+), ARM Keil MDK (wersja 5+), Energia (wersja 15+). W artykule będziemy korzystać z rozwijanego przez Texas Instruments środowiska programistycznego Code Composer Studio wersja 6.1.

Środowisko CCSv6

Code Composer Studio to środowisko programistyczne IDE, którego najnowsze wersje są oparte na Eclipse (praca w systemach operacyjnych Windows, Linux, Mac OS X). Oprogramowanie zapewnia wsparcie dla mikrokontrolerów i procesorów produkowanych przez Texas Instruments. Najnowsza wersja oprogramowania jest dostępna w wersji stacjonarnej (do zainstalowania na dysku twardym) oraz w wersji Web (w chmurze dostępnej przez Internet). Wersję webową CCSv6 udostępniono pod adresem dev.ti.com. Aby móc korzystać z oprogramowania należy mieć konto w serwisie ti.com.

Wersja webowa oprogramowania CCSv6 obsługuje wybrane produkty z oferty Texas Instruments, w tym nową serię mikrokontrolerów MSP432.

Oprogramowanie CCSv6 w wersji webowej 1.1 w porównaniu z wersją stacjonarną 6.1 ma ograniczoną funkcjonalność. Różnice pomiędzy wersjami oprogramowania prezentuje **rysunek 1**. Najważniejszy mankament wersji webowej jest taki, że korzystając z niej nie możemy debugować/emulować pracy mikrokontrolera. Brak jest możliwości korzystania z pułapek sprzętowych. Nie mamy dostępu do rejestrów oraz do pamięci mikrokontrolera. Biorąc pod uwagę wspomniane ograniczenia w artykule będziemy korzystać z wersji stacjonarnej (instalacyjnej) oprogramowania CCSv6.

Oprogramowanie CCSv6 w wersji stacjonarnej można pobrać ze strony <http://ti.com/tool/ccstudio>. W sekcji *Download* są dostępne wersje CCSv6 dla systemów

Mikrokontrolery z serii MSP432 charakteryzują się dużą wydajnością (Coremark 3.41) oraz niskim poborem mocy (ULPBench 167.4). W ofercie Texas Instruments jest dostępny zestaw startowy MSP-EXP432P401R LaunchPad wyposażony w mikrokontroler z serii MSP432. O nowej serii mikrokontrolerów MSP432 pisaliśmy w EP 05/15. Zestaw startowy LaunchPad prezentowaliśmy w EP 08/15.

operacyjnych Windows oraz Linux. W artykule korzystać będziemy z wersji dla systemu Windows i tę wersję należy pobrać i zainstalować. W tym celu, w sekcji *Download* wybieramy pole z oznaczeniem systemu operacyjnego Windows. Automatycznie uruchomiony zostanie panel logowania do konta użytkownika w serwisie ti.com. Należy się zalogować a następnie wypełnić i wysłać formularz informacyjny. Po wysłaniu formularza na adres e-mail podany w jego treści otrzymamy link do pobrania oprogramowania (aktywny przez 6 dni).

Pobieramy oprogramowanie CCSv6 oraz uruchamiamy instalator. W kolejnych krokach konfigurujemy parametry instalacji. Akceptujemy licencję (**rysunek 2a**). Wybieramy miejsce instalacji oprogramowania. Zalecana lokalizacja to `c:\ti` (**rysunek 2b**). Zaznaczamy produkty dla których ma być przeprowadzona instalacja. Nam potrzebne będą: *MSP432 Ultra Low Power MCUs* oraz kompilatory *TI ARM Compiler* i *GCC ARM Compiler* (**rysunek 2c**). Następnie potwierdzamy instalację emulatora *XDS* oraz *FET* (**rysunek 2d**). W tym momencie rozpoczyna się instalacja oprogramowania (**rysunek 2e**). Na zakończenie jesteśmy pytani o uruchomienie aplikacji oraz o utworzenie skrótów do aplikacji na pulpicie i w menu start. Odznaczamy opcję uruchomienia aplikacji i zatwierdzamy koniec instalacji (**rysunek 2f**). Na pulpicie i w menu start jest tworzony skrót do oprogramowania *Code Composer Studio 6.1*. Praca instalatora jest kończąca. Oprogramowanie jest gotowe do użycia.

Zainstalowaliśmy darmowe, w pełni funkcjonalne środowisko programistyczne CCSv6 dla mikrokontrolerów MSP432. Darmowa licencja jest aktywna przez 90 dni. Po tym czasie jest uruchamiane ograniczenie wielkości kodu wynikowego projektu. Dla MSP432 limit ten wynosi 32 kB. Ograniczenie wielkości kodu obowiązuje, jeśli korzystamy z kompilatora *TI ARM Compiler*. Jeśli korzystamy z kompilatora *GCC ARM Compiler*, to ograniczenie rozmiaru kodu wynikowego nie obowiązuje. **Wówczas możemy bezterminowo korzystać z pełnej wersji oprogramowania CCSv6 dla mikrokontrolerów MSP432!**

Feature	CCS Desktop	CCS Cloud
Code Editing/Building	✓	✓
Support for CCS Projects	✓	✓
Support for Energia (Wiring) framework	✓	✓
Flash/load program to local LaunchPad	✓	✓
Stepping & Breakpoints	✓	
View Registers and Memory	✓	
Advanced Debug and trace	✓	
Support for high performance debug probes	✓	

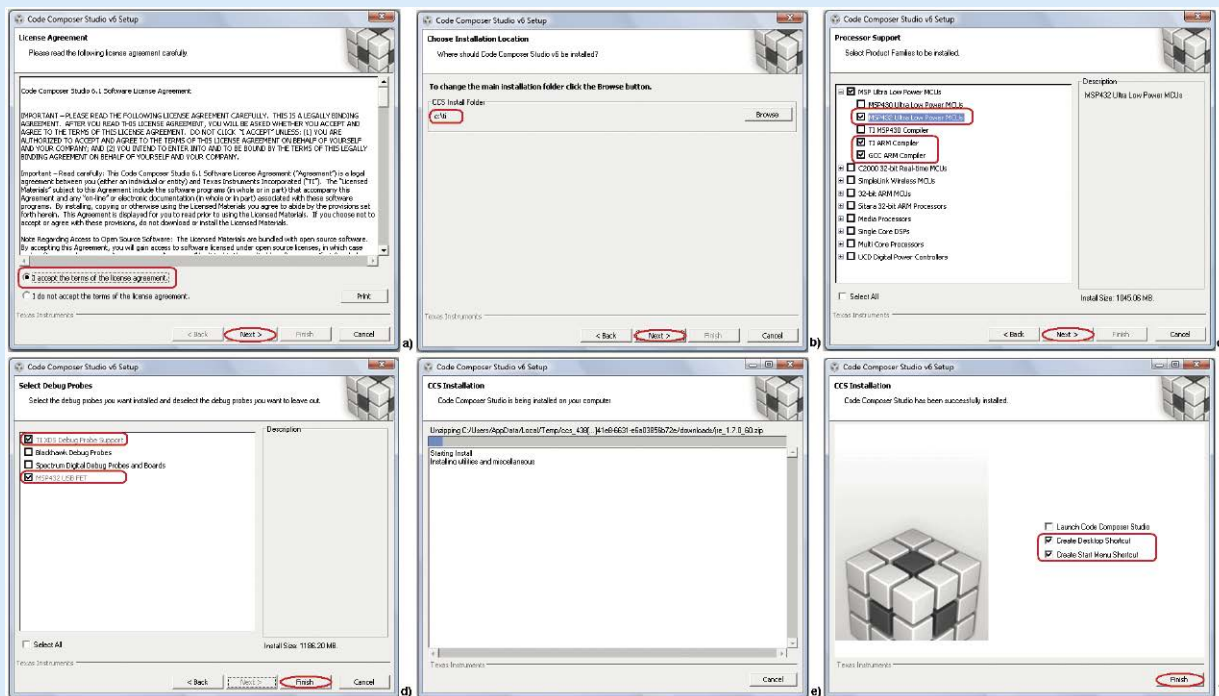
Rysunek 1. Funkcjonalność CCS. Wersja stacjonarna (desktop 6.1) a wersja webowa (cloud 1.1)

W celu pobrania oprogramowania CCSv6 oraz umożliwienia korzystania z webowych aplikacji dla MSP432 należy mieć konto w serwisie ti.com. Konto tworzymy na stronie <http://ti.com>. W prawym, górnym rogu strony wybieramy zakładkę Login/Register. Następnie wypełniamy formularz rejestracyjny. Rejestracja w serwisie ti.com jest bezpłatna.

Programator XDS110

Dla potrzeb aplikacji opisywanych w artykule będziemy korzystali z zestawu startowego MSP-EXP432P401R LaunchPad.

Moduł LaunchPad dla mikrokontrolerów MSP432 ma wbudowany układ programatora/emulatora XDS110-ET. Dodatkowo, został on przystosowany do pracy z zewnętrznym programatorem/emulatorem dołączanym



Rysunek 2. Instalowanie Code Composer Studio 6.1

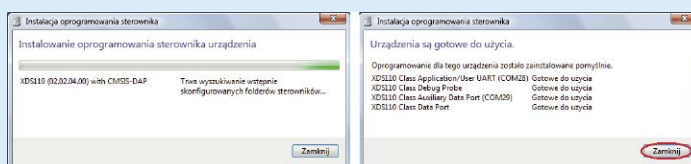
do złącza J102. Selekcji dokonujemy za pomocą przełącznika *JTAG Switch*. Dla naszych potrzeb ustawiamy przełącznik *JTAG Switch* w pozycji *XDS-ET* oraz dołączamy moduł LaunchPad do portu USB komputera PC. Rozpoczyna się instalowanie sterowników. Pliki źródłowe sterowników zostały skopiowane na dysk komputera PC podczas instalacji oprogramowania CCSv6. Instalator odnajduje folder ze sterownikami oraz instaluje drivery w systemie Windows (rysunek 3).

Po zakończeniu instalacji sterowników przechodzimy do panelu sterowania i uruchamiamy aplikację Menedżer Urządzeń. Jeśli instalacja sterowników przebiegła prawidłowo, to w systemie powinno pojawić się nowe urządzenie o nazwie *Texas Instruments Debug Probes*. Dodatkowo, w zakładce *Porty (COM i LPT)* powinny pojawić się dwa wirtualne porty COM. Wygląd okna Menedżera Urządzeń z poprawnie zainstalowanymi sterownikami dla modułu MSP432 LaunchPad pokazano na rysunku 4 (zależnie od systemu numery portów COM mogą różnić się).

Wbudowany programator/emulator XDS110-ET jest włączony po przyłączeniu modułu LaunchPad do portu USB komputera PC. Poza podstawową funkcją programowania i emulowania mikrokontrolera MSP432P401RIPZ moduł XDS110-ET pozwala na realizację komunikacji szeregowej UART pomiędzy mikrokontrolerem a komputerem PC (transmisja UART przez USB). Dodatkowo, układ XDS110-ET zapewnia wsparcie dla technologii Energy Trace (tzw. *power debugging*). Komunikacja UART pomiędzy zainstalowanym na płycie modułu LaunchPad mikrokontrolerem MSP432P401RIPZ a komputerem PC jest realizowana za pomocą wirtualnego portu COM o nazwie: *XDS110 Class Application/User UART*. Obsługiwany jest mechanizm sprzętowej kontroli przepływu (sygnały CTS, RTS). Parametry transmisji są konfigurowane w aplikacji mikrokontrolera (moduł *eUSCI_A0*).

Pierwszy projekt

Uruchamiamy środowisko programistyczne CCSv6. Na wstępie jesteśmy prośzeni o podanie lokalizacji, w której będą zapisywane projekty (zakładka



Rysunek 3. Moduł LaunchPad – instalowanie sterowników

Select a workspace). Wybieramy folder *msp432* w lokalizacji *c:\ti*. Wpisujemy zatem *c:\ti\msp432* w polu *workspace* i zatwierdzamy nasz wybór (rysunek 5).

W kolejnym kroku przechodzimy do utworzenia nowego projektu dla mikrokontrolera MSP432. W tym celu z menu programu wybieramy opcję *Project → New CCS Project*. Zaznaczamy rodzinę mikrokontrolerów MSP432. Wybieramy mikrokontroler MSP432P410R. Podajemy nazwę projektu. Zatwierdzamy ustawienia przyciskiem *Finish* (rysunek 6).

Utworzony zostanie projekt dla mikrokontrolera MSP432P410R. W projekcie będzie używany programator/emulator *XDS110* (wbudowany moduł programatora/emulatora) oraz kompilator Texas Instruments w wersji 5.2.2 (wersja komercyjna). Ustawienia projektu możemy konfigurować w zakładce *Project → Properties*.

Pierwszy program

Korzystając z utworzonego projektu stworzymy pierwszy program dla mikrokontrolera MSP432. Będzie to program „Hello Word” w wersji dla elektroników, czyli mrugająca dioda LED. W programie będziemy sterowali linią P1.0, do której w module LaunchPad dołączono diodę LED w kolorze czerwonym. Dioda będzie włączana raz na sekundę przez 500 ms. Odmierzanie czasu będzie realizowane za pomocą oprogramowania.

Otwieramy plik *main.c* projektu i rozpoczynamy tworzenie oprogramowania. W pierwszych instrukcjach programu zatrzymujemy pracę układu Watchdog:

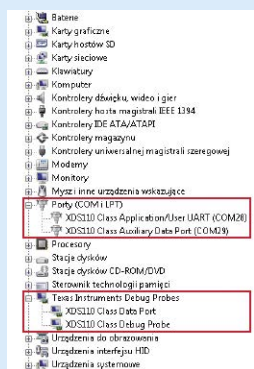
```
WDTCTL = WDTPW + WDTHOLD;
```

Następnie linię P1.0 sterującą diodą LED ustawiamy w stanie niskim i konfigurowujemy w kierunku wyjścia.

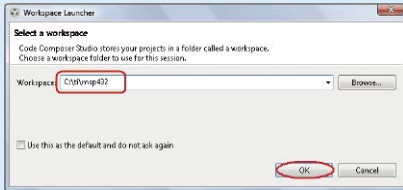
```
P1OUT &=~ BIT0;
P1DIR |= BIT0;
```

Konfigurowujemy taktowanie sygnału zegarowego ACLK z wewnętrznego generatora REF0 (32768 Hz), a sygnał zegarowy MCLK i SMCLK z wewnętrznego generatora DCO (3 MHz):

```
CSCTL1 = SELA_2 + SELS_3 + SELM_3;
```



Rysunek 4. Moduł LaunchPad. Menedżer urządzeń systemu Windows



Rysunek 5. CCSv6. Wybór lokalizacja projektu

W pętli głównej programu wprowadzamy programowe opóźnienie o czasie trwania około 500 milisekund oraz zmieniamy stan wyjścia sterującego diodą LED na przeciwny:

```
__delay_cycles(3000000/2/4);
P1OUT ^= BIT0;
```

Dioda będzie migała z częstotliwością 1 Hz i będzie włączona przez połowę okresu pracy. Kod źródłowy programu pokazano na **listingu 1**.

Następnie dołączamy moduł LaunchPad do portu USB komputera PC (przełącznik JTAG Switch w pozycji XDS-ET). Kompilujemy projekt (*Project* → *Build All*) i uruchamiamy moduł debugera (*Run* → *Debug*). Rozpoczyna się programowanie pamięci Flash mikrokontrolera. Po załadowaniu programu do pamięci mikrokontrolera uruchamiamy aplikację (*Run* → *Resume*). Dioda LED zamontowana w module LaunchPad zaczyna migać. W kolejnym kroku kończymy pracę debugera (*Run* → *Terminate*).

Pobór mocy

Aby ułatwić optymalizację poboru mocy firma Texas Instruments przygotowała oprogramowanie narzędziowe: *ULP Advisor* oraz *EnergyTrace Technology*. Oba programy zostały dołączone do środowiska programistycznego Code Composer Studio.

Aplikacja *ULP Advisor* (Ultra Low Power Advisor) analizuje kod programu i prezentuje wskazówki dotyczące optymalizacji kodu pod kątem zmniejszenia poboru mocy mikrokontrolera MSP432. Sprawdzane jest 12 reguł kontrolnych. Opis reguł jest dostępny w opcjach projektu w zakładce *Build* → *MSP432 Compiler* w polu *ULP Advisor* (**rysunek 7**).

Możemy wyłączyć sprawdzanie wszystkich reguł (przycisk *None*) albo pojedynczej reguły (checkbox przy nazwie reguły). W praktyce wyłączenie sprawdzania reguł nie jest zalecane. Wyniki pracy aplikacji *ULP Advisor* są **prezentowane w oknie Advice** edytora projektu CCSv6.

Aby uruchomić analizę kodu programu musimy skompilować projekt (*Project* → *Build All*). W naszym programie aplikacja *ULP Advisor* wykrywa 11 nieprawidłowości. Wskazówki aplikacji dotyczą trzech reguł:

- Numer 1: brak użycia trybu uśpienia LPM w programie (mikrokontroler MSP432 to układ energooszczędny i powinien korzystać z trybów uśpienia).
- Numer 2: zastosowanie programowego opóźnienia (zamiast programowego opóźnienia do odmierzenia czasu należy stosować liczniki sprzętowe w połączeniu z trybami uśpienia).
- Numer 4: brak inicjalizacji portów wejścia-wyjścia mikrokontrolera (nieużywane porty wejścia-wyjścia mikrokontrolera MSP432 powinny być skonfigurowane jako wyjścia na poziomie niskim).

Zatem poprawmy nasz program zgodnie z zaleceniami.

W programie jest używana tylko linia wejścia/wyjścia P1.0, do której w module LaunchPad jest dołączona dioda LED. Pozostałe linie wejścia-wyjścia są nieużywane – należy je wyzerować. Do odmierzenia czasu użyjemy licznika TAR. Licznik będzie odmierzał półsekundowe odcinki czasu i będzie obsługiwany w trybie przerwań. Mikrokontroler będzie wprowadzany w tryb uśpienia z którego będzie budzony przez przerwania od licznika TAR (raz na 0,5 s). Po wyjściu z uśpienia będziemy zmieniali stan wyjścia sterującego diodą LED (jedno mignięcie diody na sekundę).

Na początku ustawiamy taktowanie licznika TAR sygnałem ACLK, zatrzymujemy pracę licznika i **włączamy obsługę trybu przerwań**:

```
TAOCTL = TASSEL_1 + MC_0 + TAIE;
```

Następnie zerujemy licznik i konfigurujemy czas pracy licznika TAR. Licznik będzie pracował w trybie *Up Mode* (*licz do*). Żeby skonfigurować czas pracy licznika musimy zaprogramować wartość 16-bitowego rejestru

Listing 1. Pierwszy program - wersja 1

```
#include "msp.h" // pliki nagłówkowe

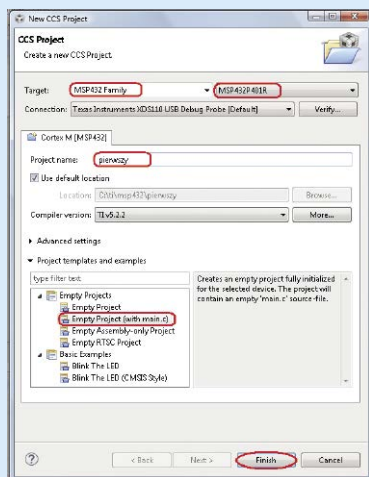
void main(void) // program główny
{
    // konfiguracja układu watchdog
    WDTCTL = WDTPW + WDTHOLD; // zatrzymaj układ
    watchdog
    // konfiguracja linii we-wy
    P1OUT &= ~BIT0; // wyzeruj P1.0
    P1DIR |= BIT0; // P1.0 ustaw linię w kierunku
    wyjścia
    // konfiguracja systemu zegarowego
    CSKEY = CSKEY_VAL; // odblokuj dostęp do modułu
    // ACLK= REFOCLK, SMCLK=MCLK= DC0CLK
    CSCTL1 = SELA_2 + SELS_3 + SELM_3;
    CSKEY = 0; // zablokuj dostęp do modułu
    // pętla główna programu
    while(1)
    {
        __delay_cycles(3000000/2/4); // czekaj około
        0.5 sekundy
        P1OUT ^= BIT0; // zmień stan linii sterującej
    } // dioda LED na przeciwny
}
```

Listing 2. Pierwszy program - wersja numer 2

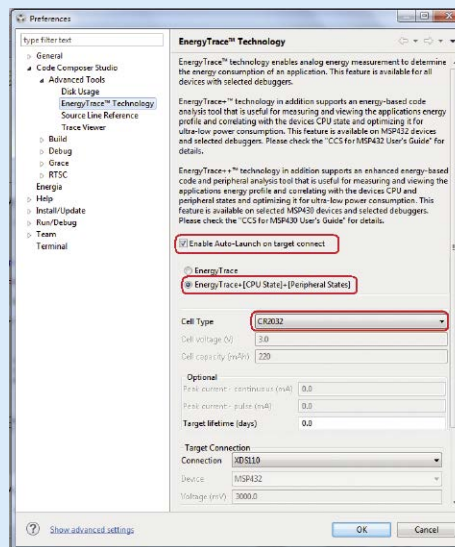
```
#include "msp.h" // pliki nagłówkowe

void main(void) // program główny
{
    // konfiguracja układu watchdog
    WDTCTL = WDTPW + WDTHOLD; // zatrzymaj układ watchdog
    // konfiguracja linii we-wy
    P1OUT = 0x00; P2OUT = 0x00; P3OUT = 0x00; // nieużywane linie we-wy
    P4OUT = 0x00; P5OUT = 0x00; P6OUT = 0x00; // zeruj wyjścia
    P7OUT = 0x00; P8OUT = 0x00; P9OUT = 0x00;
    P10OUT = 0x00; P10OUT = 0x00; // linia P1.0 również jest konfigurowana
    P1DIR = 0xFF; P2DIR = 0xFF; P3DIR = 0xFF; // wyjście, wyzerowane
    P4DIR = 0xFF; P5DIR = 0xFF; P6DIR = 0xFF;
    P7DIR = 0xFF; P8DIR = 0xFF; P9DIR = 0xFF;
    P10DIR = 0xFF;
    // konfiguracja systemu zegarowego
    CSKEY = CSKEY_VAL; // odblokuj dostęp do modułu
    CSCTL1 = SELA_2 + SELS_3 + SELM_3; // ACLK = REFO, SMCLK = MCLK = DC0CLK
    CSKEY = 0; // zablokuj dostęp do modułu
    // konfiguracja licznika TA0R
    TAOCTL = TASSEL_1 + MC_0 + TAIE; // licznik TA0R taktuj sygnałem
    TA0R = 0; // zegarowym ACLK, zatrzymaj pracę
    TA0CCR0 = 16383; // licznika, włącz przerwania od przepełnienia
    // następnie wyzeruj licznik, modyfikuj TA0CCR0
    // (czas pracy 0.5 sekundy)
    enable_interrupt(); // włącz obsługę przerwań
    NVIC_ISER0 = 1 << ((INT_TA0_N - 16) & 31);
    SCB_SCR &= ~SCB_SCR_SLEEPONEXIT; // po zakończeniu obsługi przerw.
    // opuść procedurę obsługi
    TAOCTL |= MC_1; // uruchom licznik (tryb Up Mode)
    // pętla główna programu
    while(1)
    {
        __sleep(); // aktywuj tryb uśpienia
        P1OUT ^= BIT0; // po przebudzeniu z uśpienia
    } // zmień stan linii sterującej
} // dioda LED na przeciwny

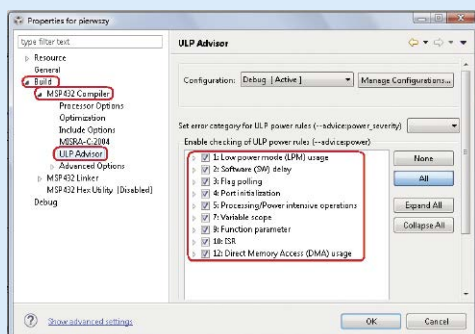
// procedura obsługi przerwania licznika TA0R
void TimerA0_NISRHandler(void)
{
    // wyzeruj flagę przepełnienia licznika
    TAOCTL &= ~TAIFG; // wyjdź z procedury
    // do pętli głównej programu
}
```



Rysunek 6. CCSv6. Tworzenie projektu dla MSP432



Rysunek 8. EnergyTrace Technology. Konfiguracja aplikacji



Rysunek 7. CCSv6. Okno konfiguracji modułu ULP Advisor

W kolejnym kroku włączamy obsługę przerwania maskowalnych mikrokontrolera MSP432 oraz przerwania od licznika TAR:

```
_enable_interrupt();
NVIC_ISER0 = 1 << ((INT_TA0_N - 16) & 31);
Dodatkowo, definiujemy regułę, aby po zakończeniu wykonania procedury obsługi przerwania mikrokontroler opuścił tryb uśpienia:
```

```
SCB_SCR &= ~SCB_SCR_SLEEPONEXIT;
```

Następnie uruchamiamy licznik TAR (wybieramy tryb pracy *Up Mode*):

```
TAOCTL |= MC_1;
```

W pętli głównej programu wprowadzamy mikrokontroler w tryb uśpienia. Z uśpienia mikrokontroler jest budzony przez przerwanie od licznika TAR. Rozpoczyna się wykonanie kodu procedury obsługi przerwania (czyszczenie flagi przerwania):

```
TAOCTL &=~ TAIFG;
```

Następnie mikrokontroler wraca do wykonania pętli głównej programu. Zgodnie z wcześniejszą konfiguracją mikrokontroler opuszcza tryb uśpienia i wykonuje kolejną instrukcję programu. W naszym przypadku jest to zmiana stanu wyjścia sterującego diodą LED na przeciwny.

```
PIOUT ^= BIT0;
```

Po zmianie stanu diody LED mikrokontroler jest ponownie usypiany. Cykl pracy programu powtarza się. Zmodyfikowany program zamieszczono na **listingu 2**.

TACCR0. Wartość wpisywaną do rejestru TACCR0 obliczmy ze wzoru 1.1:

$$wRejestr0 = tPracy \times fZegara - 1, \quad (1.1)$$

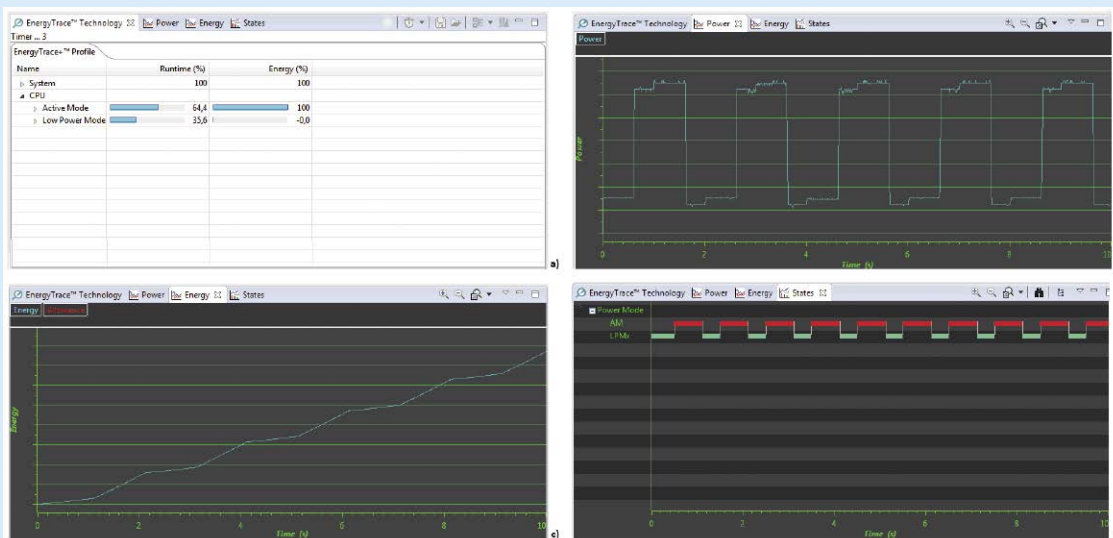
gdzie:

- tPracy – czas pracy licznika TAR w trybie Up Mode (s),
- fZegara – częstotliwość sygnału taktującego licznik TAR (Hz),
- wRejestr0 – wartość wpisywana do rejestru TACCR0.

Ponieważ częstotliwość sygnału ACLK taktującego licznik TAR wynosi 32768 Hz, to zgodnie ze wzorem 1.1 do rejestru TACCR0 wypisujemy wartość 16383:







$$TAOR = 0;$$

$$TAOCCR0 = 16383;$$



Rysunek 9. EnergyTrace+. Okna: a) Profile b) Power c) Energy d) States

Tabela 1. EnergyTrace Technology. Sterowanie pracą aplikacji

Przycisk	Opis
	Włącz/wyłącz działanie aplikacji. Gdy aplikacja jest wyłączona ikona koloru szarego.
	Ustaw czas pracy aplikacji. Dostępne nastawy to: 5 sekund, 10 sekund, 30 sekund, 1 minuta, 5 minut, On halt – praca ciągła do moment zatrzymania aplikacji (Run → Suspend).
	Zapisz profil energetyczny aplikacji.
	Wczytaj profil energetyczny aplikacji. Opcja wykorzystywana podczas porównywania parametrów zużycia energii urządzenia.
	Uruchom okno konfiguracji aplikacji albo przywróć zamknięte zakładki aplikacji.
	Przełącz tryb pracy (z EnergyTrace+ na EnergyTrace oraz z EnergyTrace na EnergyTrace+).

Kod źródłowy programu pokazano na listingu 2. Jeśli teraz skompilujemy program (Project → Build All) to w oknie *Advice* brak będzie informacji od aplikacji *ULP Advisor*. Dzieje się tak ponieważ reguły tworzenia aplikacji o niskim poborze mocy zostały spełnione.

EnergyTrace Technology

Oprogramowanie *EnergyTrace Technology* mierzy i wyświetla profil energetyczny aplikacji oraz pomaga zoptymalizować aplikację pod kątem poboru mocy. Pomiar poboru prądu odbywa się w nietypowy sposób, gdyż jest mierzona charakterystyka pracy przetwornicy DC/DC zasilającej mikrokontroler, a zwykle – w tradycyjnej metodzie – mierzy się spadek napięcia na rezystorze pomiarowym.

Oprogramowanie *EnergyTrace Technology* dla mikrokontrolerów MSP432 jest dostępne w dwóch wersjach: EnergyTrace oraz EnergyTrace+. Wersja podstawowa bez „plusa” w nazwie dostarcza informacji o profilu energetycznym aplikacji, zużyciu mocy oraz zużyciu energii. Wersja z plusem w nazwie jest bardziej zaawansowana i dodatkowo dostarcza informacji o zmianach stanów pracy jednostki CPU mikrokontrolera MSP432 (prezentacja pracy w trybie aktywnym oraz w trybach uśpienia).

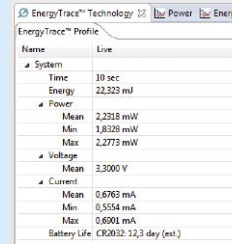
W ustawieniach domyślnych oprogramowania CCSv6 aplikacja *EnergyTrace Technology* jest wyłączona. Aby włączyć aplikację, przechodzimy do ustawień CCSv6 (*Window* → *Preferences*). Wybieramy zakładkę *Code Composer Studio*, pole *Advanced Tools*, pozycja *EnergyTrace™ Technology*. Następnie zaznaczamy opcję *Enable Auto-Launch on target connect*. Dodatkowo, możemy wybrać domyślny tryb pracy aplikacji oraz rodzaj zasilania baterijnego (istnieje możliwość wprowadzenia własnych parametrów baterii). Wygląd okna konfiguracji aplikacji *EnergyTrace Technology* pokazano na **rysunku 8**.

Uwaga! Moduł LaunchPad dla mikrokontrolerów MSP432 w trybie debugowania ma maksymalną wydajność prądową 75 mA. Przekroczenie dopuszczalnej wartości poboru prądu spowoduje zatrzymanie pracy debugera oraz aplikacji EnergyTrace Technology.

Aplikacja *EnergyTrace Technology* jest uruchamiana razem ze startem debugera/emulatora oprogramowania CCSv6 (*Run* → *Debug*). Pracę aplikacji kontrolujemy za pomocą **przycisków z paska sterowania**. Ikony przycisków oraz ich funkcje zostały opisane w **tabeli 1**.

W trybie pracy EnergyTrace+ dostępne są zakładki: *Profile*, *Power*, *Energy*, *States*. Przykład działania aplikacji pokazano na **rysunku 9**. W trybie pracy EnergyTrace brak jest zakładki *States*, a zakładka *Profile* wyświetla bardziej szczegółowe dane niż w przypadku EnergyTrace+. Prezentowane są informacje o czasie pomiaru, zużyciu energii, średniej maksymalnej oraz minimalnej wartości mocy i prądu. Dodatkowo, jest prezentowany przybliżony czas działania urządzenia przy zasilaniu z baterii o parametrach zdefiniowanych przez użytkownika (konfiguracja *EnergyTrace Technology* pole *Cell Type*). Przykład działania aplikacji pokazano na **rysunku 10**.

W obu trybach pracy mikrokontroler MSP432 jest cyklicznie odpytywany przez oprogramowanie *EnergyTrace Technology*. Interfejs komunikacyjny (JTAG

**Rysunek 10. EnergyTrace+. Okno Profile**

```

C:\td\er\ti\ccsv6\ccs_base\common\usci\if\xds110>
C:\td\ccsv6\ccs_base\common\usci\if\xds110>xdsdfu -n
USB Device Firmware Upgrade Utility
Copyright (C) 2000-2015 Texas Instruments Incorporated. All rights reserved.
Scanning USB buses for supported XDS110 devices...

<<<< Device 0 >>>>
VID: 040451 PID: 040ef3
Device Name: XDS110 with CMSIS-DAP
Version: 2.2.4.0
Manufacturer: Texas Instruments
Serial Num: 050000000
Model: JTAG line

Switching device into DFU mode.
C:\td\ccsv6\ccs_base\common\usci\if\xds110>xdsdfu -f firmware_bin -r
USB Device Firmware Upgrade Utility
Copyright (C) 2000-2015 Texas Instruments Incorporated. All rights reserved.
Scanning USB buses for supported XDS110 devices...
Downloading Firmware.bin to device...
C:\td\ccsv6\ccs_base\common\usci\if\xds110>

```

Rysunek 11. XDS110. Aktualizacja oprogramowania debugera/emulatora

lub Spy-Bi-Wire) jest cały czas aktywny. W związku z tym, wartości w zakładkach *Profile*, *Power* oraz *Energy są zawyżone o pobór mocy debugera. Żeby zmierzyć rzeczywisty pobór mocy (bez narzutu debugera), należy wyłączyć komunikację debugera z mikrokontrolerem (Run → Free Run)*. Wówczas w trybie EnergyTrace+ zakładka *State* przestanie udostępniać dane (brak komunikacji z mikrokontrolerem), ale pozostałe dane będą zbliżone do rzeczywistych.

Power debugging

Aby zaprezentować power debugging (debugowanie programu powiązane z pomiarem mocy) skorzystamy z oprogramowania *EnergyTrace*. Do celów prezentacji wykorzystamy moduł LaunchPad oraz dwa prezentowane programy.

Przed rozpoczęciem pracy z aplikacją *EnergyTrace* musimy zaktualizować oprogramowanie wewnętrznego debugera/emulatora XDS110. Po podłączeniu modułu LaunchPad do portu USB komputera PC uruchamiamy wiersz poleceń systemu operacyjnego Windows (*Start* → *Uruchom* → *cmd*). Następnie przechodzimy do lokalizacji `c:\ti\ccsv6\ccs_base\common\usci\if\xds110` i wprowadzamy polecenia, jak na **rysunku 11**. Rozpocznie się aktualizacja oprogramowania wbudowanego debugera/emulatora XDS110. Po jej zakończeniu uruchamiamy pierwszy projekt w wersji numer 1. Otwieramy okno konfiguracji aplikacji *EnergyTrace Technology*. Włączamy aplikację oraz ustawiamy tryb pracy EnergyTrace. Rodzaju baterii nie konfigurujemy pozostając przy domyślnej opcji z baterią CR2032. Następnie zamykamy okno konfiguracji aplikacji *EnergyTrace Technology*, kompilujemy oprogramowanie (*Project* → *Build All*) i uruchamiamy debugger (*Run* → *Debug*). Rozpocznie się programowanie pamięci Flash mikrokontrolera MSP432.

Po wgraniu oprogramowania uruchamiamy pracę debugera w trybie *Free Run (Run → Free Run)*. Wówczas aplikacja *EnergyTrace* rozpocznie gromadzenie danych. Gdy minie 10 sekund (domyślny czas pracy aplikacji EnergyTrace), działanie aplikacji zostanie zakończone.

Aby ułatwić konfigurowanie linii wejścia/wyjścia mikrokontrolerów MSP432 firma Texas Instruments udostępniła narzędzie o nazwie TI PinMux wersja 4.0. W odróżnieniu od poprzednich wersji oprogramowania TI PinMux wersja 4.0 jest dostępna wyłącznie w wersji webowej (w chmurze). Aplikacja została udostępniona pod adresem <http://dev.ti.com>. Aby móc korzystać z oprogramowania, należy mieć konto w serwisie ti.com.

Aplikacja *EnergyTrace Technology* poza mikrokontrolerami MSP432 zapewnia wsparcie dla mikrokontrolerów MSP430. W przypadku MSP430 jest dostępny trzeci tryb pracy EnergyTrace++ z dwoma plusami w nazwie. Tryb ten umożliwi monitorowanie i wizualizację poboru mocy modułów peryferyjnych mikrokontrolera MSP430. Szczegółowy opis *EnergyTrace Technology* jest dostępny na stronie [www.projektu http://ti.com/tool/energytrace](http://ti.com/tool/energytrace).

Wydanie specjalne

„Raspberry Pi”

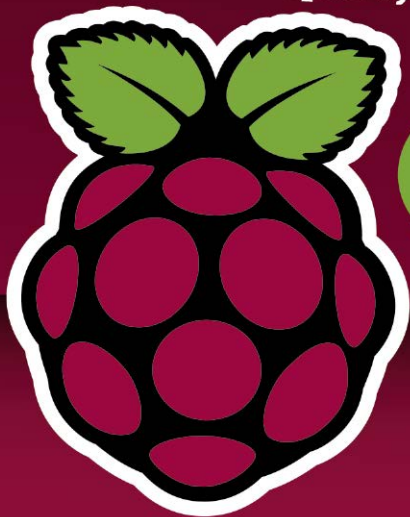
to polski przekład światowego bestsellera
na temat słynnego minikomputera

WYDANIE SPECJALNE „MŁODEGO TECHNIKA” NR. 1/2015

Raspberry Pi

Ależ to bardzo proste!

Jak w pełni wykorzystać możliwości
minikomputera Raspberry Pi



196
pomysłów
i
porad

KOMPENDIUM DLA NIEELEKTRONIKÓW

- ROZPOCZĘCIE PRACY
- PODSTAWOWE UMIEJĘTNOŚCI
- PROGRAMOWANIE
- PROJEKTY

To kompendium wiedzy o konfiguracji

i sposobach programowania tego uniwersalnego urządzenia
oraz prawie dwieście pomysłów i sztuczek aplikacyjnych

Nie będziesz rozczarowany!

Nie musisz być elektronikiem, aby zaprzęgnąć Raspberry Pi
do wykonywania niezliczonych rodzajów funkcji i aplikacji

Z tym poradnikiem możesz to osiągnąć!

Szukaj na

www.UlubionyKiosk.pl

(przesyłka GRATIS)

Zapisujemy profil energetyczny programu w wersji numer 1 do pliku i kończymy pracę debugera (*Run* → *Terminate*). Następnie, w edytorze projektu oprogramowania CCSv6, zmieniamy program z wersji numer 1 na wersję numer 2. Kompilujemy oprogramowanie i uruchamiamy debuger. Po wgraniu oprogramowania do pamięci Flash mikrokontrolera MSP432 uruchamiamy pracę debugera w trybie *Free Run*. Gdy minie czas pracy aplikacji EnergyTrace (10 sekund) nie zapisujemy profilu energetycznego programu w wersji numer 2, a wczytujemy profil energetyczny programu w wersji numer 1. Oba profile zostaną porównane. Wynik porównania pokazano na rysunku 12.

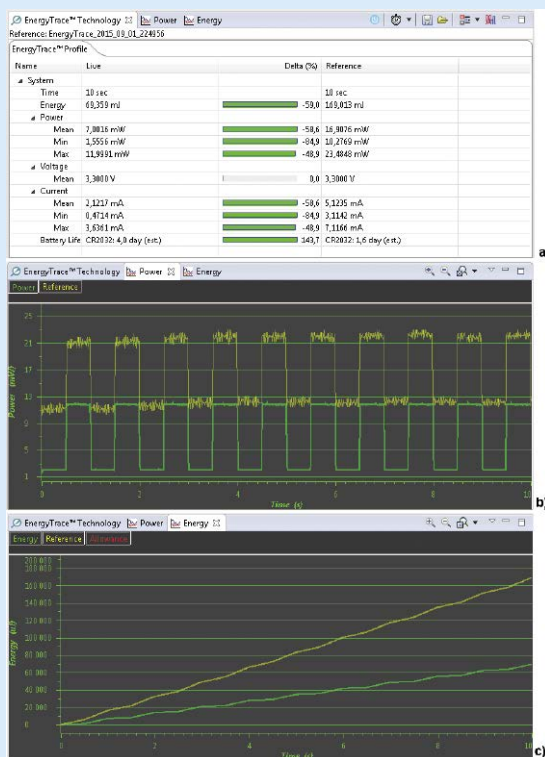
Analiza danych jednoznacznie potwierdza, że zmiany, które wprowadziliśmy do programu w wersji numer 1 sprawiły, iż program w wersji numer 2 jest znacznie bardziej energooszczędny. Aby uzyskać jeszcze lepsze parametry poboru mocy, możemy nadal modyfikować program. Przykładowo, zamiast trybu uśpienia LPM0 możemy zastosować tryb uśpienia LPM3, w którym pobór prądu mikrokontrolera MSP432 wynosi zaledwie 850 nA.

Dalsze modyfikacje oprogramowania i analizę poboru mocy z wykorzystaniem aplikacji *EnergyTrace Technology* pozostawiamy dociekliwym czytelnikom.

Podsumowanie

W artykule zaprezentowaliśmy sposób rozpoczęcia pracy z nową serią mikrokontrolerów MSP432. Zainstalowaliśmy oprogramowanie narzędziowe IDE Code Composer Studio. Utworzyliśmy pierwszy projekt oraz napisaliśmy pierwszy program dla mikrokontrolera MSP432. Dodatkowo, wykonaliśmy optymalizację poboru mocy korzystając z udostępnianych przez Texas Instruments narzędzi: *ULP Advisor* i *EnergyTrace Technology*. Kod źródłowy projektu prezentowanego w artykule (wersja numer 1 i wersja numer 2) został zamieszczony w materiałach dodatkowych na FTP. Dodatkowo udostępniona została dokumentacja dla mikrokontrolerów MSP432, modułu LaunchPad i środowiska programistycznego IDE CCSv6 – noty aplikacyjne, przykłady konfiguracji, kody źródłowe przykładowych programów.

Łukasz Krysiewicz, EP



Rysunek 12. EnergyTrace. Porównanie profili energetycznych a) Live – wersja 2, b) Reference – wersja 1, c) kolor zielony – wersja 2, kolor żółty – wersja 1