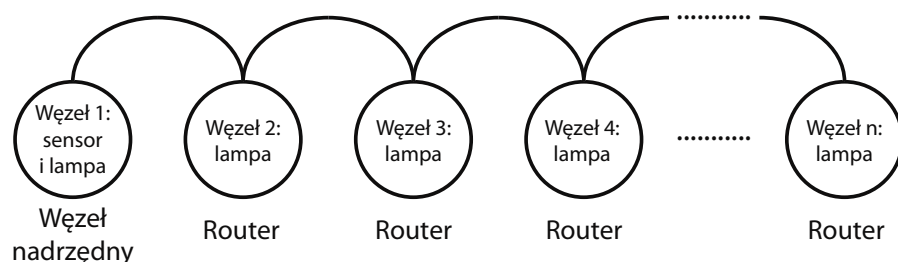
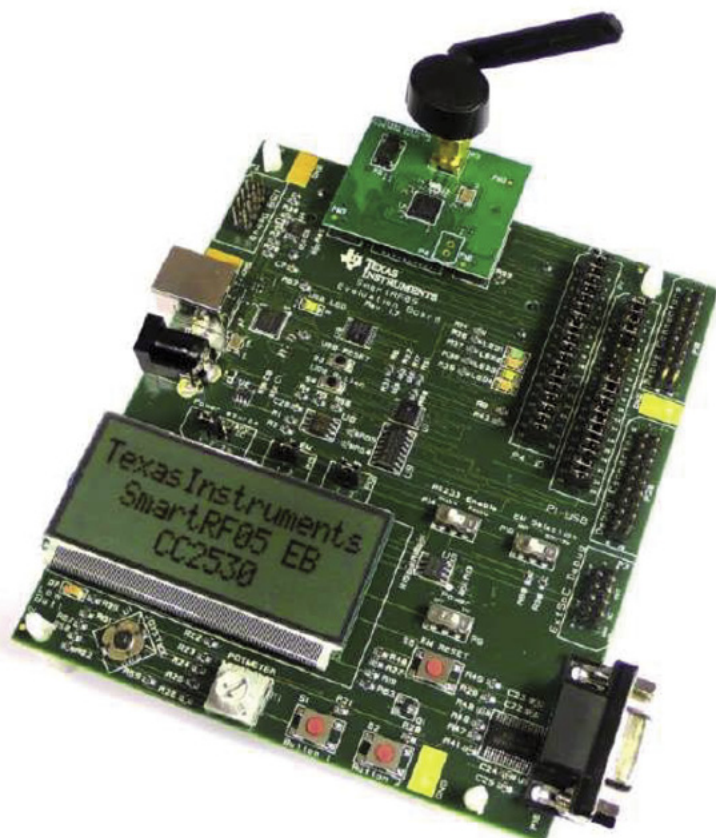


# Sterowanie latarniami ulicznymi z użyciem ZigBee

Lubimy, aby ulice w miastach były oświetlone. Poprawia to nasze poczucie bezpieczeństwa i komfort życia. Tradycyjnie stosowaną od lat metodą oświetlenia ulic jest montaż lamp i ich zasilanie z sieci energetycznej. Wymaga to ułożenia pod powierzchnią ziemi dużej ilości kabli. W związku z postępem technologicznym, coraz większa popularność zdobywają lampy zasilane akumulatorami ładowanymi za pomocą energii słonecznej. Zazwyczaj takie latarnie uliczne pracują niezależnie. Brak połączenia pomiędzy nimi oraz różnice w zacienieniu miejsc, w których są zamontowane a także rozrzut wykonania sensorów oświetlenia otoczenia sprawiają, że latarnie nie zaświecają się jednocześnie, co nie wygląda dobrze. Rozwiązaniem tego problemu może być sterowanie z użyciem ZigBee.



Rysunek 1. Węzły sieci ZigBee sterującej latarniami ulicznymi



Fotografia 2. Zestaw SmartRF05 z modulem CC2530EM

**Dodatkowe informacje:**  
 Arrow Electronics Poland  
 ul. Rzymowskiego 53, 02-697 Warszawa  
 tel. 22 856 90 90, faks 22 558 82 83  
[www.arroweurope.com](http://www.arroweurope.com)

Sens stosowania niezależnych lamp ulicznych zasilanych energią słoneczną pobieraną dostrzega coraz większa liczba zarządców miast. Jeszcze do niedawna koszt zakupu takiej lampy był na tyle duży, że musiało upłynąć wiele czasu, zanim zwrócił się. Problemem było również nierównoczesne włączanie się lamp stojących w szeregu. Poszczególne czujniki mogą być nierównomiernie oświetlone ze względu na chmury, liście drzew, zabrudzenia, czy nawet ptaki. Dodatkowo, tolerancja wykonania tych sensorów sprawia, że nawet przy jednolitym oświetleniu, niektóre latarnie będą włączały się wcześniej niż inne. Skutkuje to niepożądanym efektem, w którym część lamp stojących przy ulicy jest włączonych, a część nie, co sprawia wrażenie, jakby te wyłączone nie działały.

Obecnie przy wzrastających cenach energii elektrycznej oraz miedzi, a także przy malejących kosztach zakupu ogniw słonecznych, koszt instalacji latarni zasilanych energią słoneczną dosyć szybko zwraca się. Synchronizację włączania i wyłączania lamp można natomiast przeprowadzić z użyciem ZigBee. Standard ten określa komunikację w otwartym paśmie ISM i pozwala na wymianę informacji pomiędzy latarniami, które zazwyczaj oddalone są od siebie o od 40 do 60 m. Najważniejszą cechą ZigBee, która ma znacznie w omawianym zastosowaniu, jest możliwość tworzenia sieci o topologii kraty (mesh), w której wszystkie węzły mogą funkcjonować jako routery. Dzięki temu latarnie można połączyć bezprzewodowo i przysyłać polecenia włączenia i wyłączenia światła wydawane przez węzeł sterujący wyposażony w czujnik oświetlenia otoczenia (rysunek 1).

## Przykładowa aplikacja

Aby zbudować system sterowania oświetleniem w omówiony powyżej sposób, jest konieczne utworzenie dwóch rodzajów węzłów: nadrzędnego i podrzędnego. Zadaniem pierwszego z nich są:

- badanie natężenia światła otoczenia z użyciem odpowiedniego czujnika, generowanie sygnału PWM, sterującego

Tworząc podobne aplikacje ZigBee warto zapoznać się z opracowanym przez Texas Instruments dokumentem *Create New Applications for SmartRF05 and CC2530*, udostępnianym pod numerem katalogowym SWRA231.

natężeniem światła lampy, na której jest zamontowany; dzięki temu jasność latarni zwiększa się wraz z ściemnianiem się otoczenia, oszczędzając energię,

- przesyłanie informacji o parametrach sygnału PWM do pozostałych latarni, by tamte dostosowały swój poziom światła do węzła nadrzędnego.

Zadania drugiego typu węzłów sprowadzają się do:

- odbierania informacji o sygnale PWM,
- sterowania podłączoną lampą za pomocą sygnału PWM zgodnego z odebrany,
- przekazywania dalej informacji o sygnale PWM do kolejnych węzłów; warto zauważyć, że ta część oprogramowania również może być identyczna dla wszystkich węzłów podrzędnych, gdyż nie ma potrzeby rozróżniania ich adresów.

W omawianym, przykładowym projekcie zastosowano komponenty firmy Texas Instruments. Użyto płytki deweloperskiej *SmartRF05EB* oraz modułu *CC2530DB* (fotografia 2). Obsługę ZigBee zaimplementowano ze stosem Z-Stack w wersji 2.4.0-1.4.0 udostępnianego również przez Texas Instruments. Aplikacje utworzono w oprogramowaniu *IAR Embedded Workbench for 8051* w wersji 7.60.1.

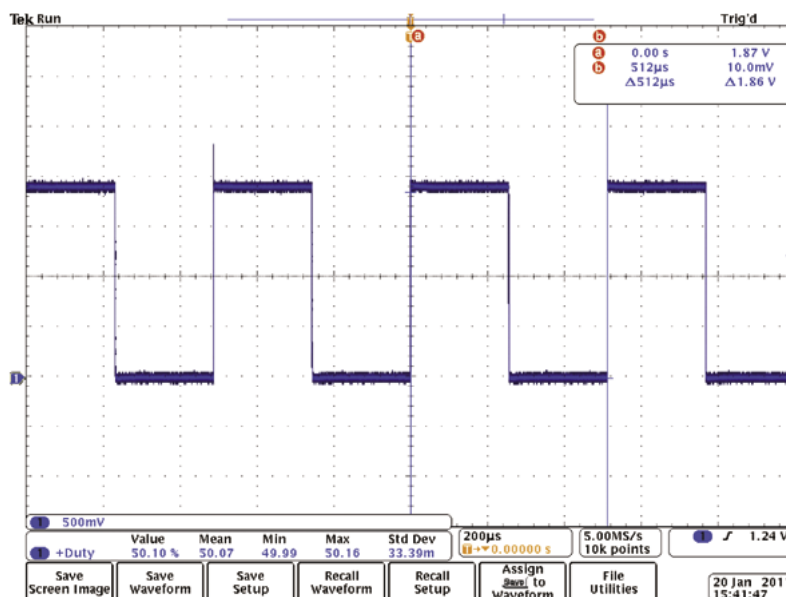
W niniejszym artykule wykorzystano przykład *GenericApp* dostarczony przez Texas Instruments, nieco go jednak modyfikując. W wersji oryginalnej wysłał on zdanie „Hello World” co 5 sekund i umożliwia transmisję w obie strony. W omawianym zastosowaniu węzeł nadzorujący sieć nie może odbierać komunikatów.

### Krok 1 – czujnik światła

Zmiany natężenia oświetlenia otoczenia symulujemy za pomocą potencjometru umieszczonego na płytce deweloperskiej. Napięcie z potencjometru jest podawane na port P0.7 układu CC2530 skonfigurowanego jako wejście. Przed konfiguracją konwertera A/C, który będzie korzystał z wejścia P0.7, należy poprawnie skonfigurować port wyjściowy, przez który będzie wyprowadzany sygnał z generatora PWM. Do taktowania sygnału PWM można użyć dowolnego z wbudowanych timerów, za wyjątkiem timera 2, z którego korzysta stos ZigBee. W tym przykładzie wybrano pierwszy timer (16-bitowy).

Sygnał PWM będzie dołączony do nóżki 11 układu (port P1.0). Dołączono ją do diody LED na płytce SmartRF05EVB. Do rejestru P1DIR musi być zapisana wartość 0x01.

Zgodnie z dokumentacją „HAL Driver API” firmy Texas Instruments potrzebne są



Rysunek 3. Przebiegi PWM przy współczynniku wypełnienia równym 50%

dwa wywołania funkcji do odczytu wartości z przetwornika A/C:

1. *HalAdcSetReference* (*HAL\_ADC\_FER\_AVDD*), która służy do ustalenia napięcia odniesienia przetwornika ADC, np. w stosunku do AVDD. Oznacza to, że napięcie 0 V zostanie przekonwertowane na wartość 0, a napięcie 3 V na 255 (przy 8-bitowej rozdzielczości przetwornika).
2. *HalAdcRead* (*HAL\_ADC\_CHN\_AIN7*, *HAL\_ADC\_RESOLUTION\_12*) powoduje odczyt kanału 7, który jest podłączony do potencjometru i ustala rozdzielczość na 12 bitów. Trzeba zwrócić uwagę na fakt, że układ CC2530 podaje na wyjściu wartość z zakresu od 0 do 2047.

Obie funkcje muszą być dołączone za pomocą instrukcji `#include <hal_adc.h>`, aby można było ich użyć.

Aby skonfigurować timer 1 należy kolejno:

- skierować wyjście timera 1 na pin 12 (`PERCFG |= 0x40`),
- ustawić odpowiednio rejestr peryferiów (`PSEL |= 0x01`),
- ustawić rejestr sterujący timerem 1, tak aby dzielił sygnał zegarowy o częstotliwości 32 MHz przez 8 i aby timer pracował w trybie Modulo, tj. odliczał od 0 do zadanej wartości (`T1CTL = 0x06`),
- ustawić zawartość rejestru timera 1 na 2048 taktów zegarowych (`T1CC0H = 0x08`; `T1CC0L = 0x00`), co pozwoli uzyskać okres 512 µs (rysunek 3),
- ustawić rejestr Capture/Compare Control Register tak na „set when equal T1CC0 and clear when equal T1CC2” (`T1CCTL2 = 0x34`).

Powyższe kroki spowodują wygenerowanie sygnału PWM na pinie 12. Współczynnik wypełnienia będzie się zmieniał w odniesieniu do wartości w rejestrach T1CC2H i T1CC2L. Pozostałe operacje są względnie proste: po odczytaniu wskazania potencjo-

metru funkcją *HalAdcRead()* należy odpowiednio ustawić wartości tych rejestrów. 12-bitowa liczba może być przekonwertowana przy użyciu predefiniowanych makr *HI\_UINT8* i *LO\_UINT8* (`T1CC2H = HI_UINT8(wartość)` i `T1CC2L = LO_UINT8(wartość)`).

Należy zaznaczyć, że wysyłany łańcuch sterujący musi być typu znakowego. Z tego względu wartość liczbowa po odczytaniu z przetwornika A/C musi być zamieniona na łańcuch znaków np. za pomocą funkcji *itoa(wartość odczytana z ADC, opis stringu wyjściowego, precyzja)*. Na koniec, w środowisku IAR IDE program węzła nadrzędnego musi zostać skompilowany i zapamiętany jako *ZigBee PRO Coordinator*.

### Krok 2 – Router

Co oczywiste, w pozostałych węzłach, przetwornik A/C nie jest potrzebny. Poza tym, te węzły mogą używać tych samych ustawień timera, co w węzle nadrzędnym. Po otrzymaniu informacji o współczynniku wypełnienia sygnału PWM, wartości te powinny być odpowiednio zapisane do rejestrów T1CC2H i T1CC2L. Programy tych węzłów muszą być zbudowane jako ZigBee PRO Router.

### Podsumowanie

Zademonstrowana aplikacja pokazuje jak łatwo można zmodyfikować dostarczone przez producenta kody źródłowe i przykładowe programy pracujące z użyciem Z-Stack firmy Texas Instruments do opracowania własnych projektów. Omówione zastosowanie pozwala stworzyć rozległą sieć ZigBee sterującą pracą latarni ulicznych zasilanych światłem słonecznym.

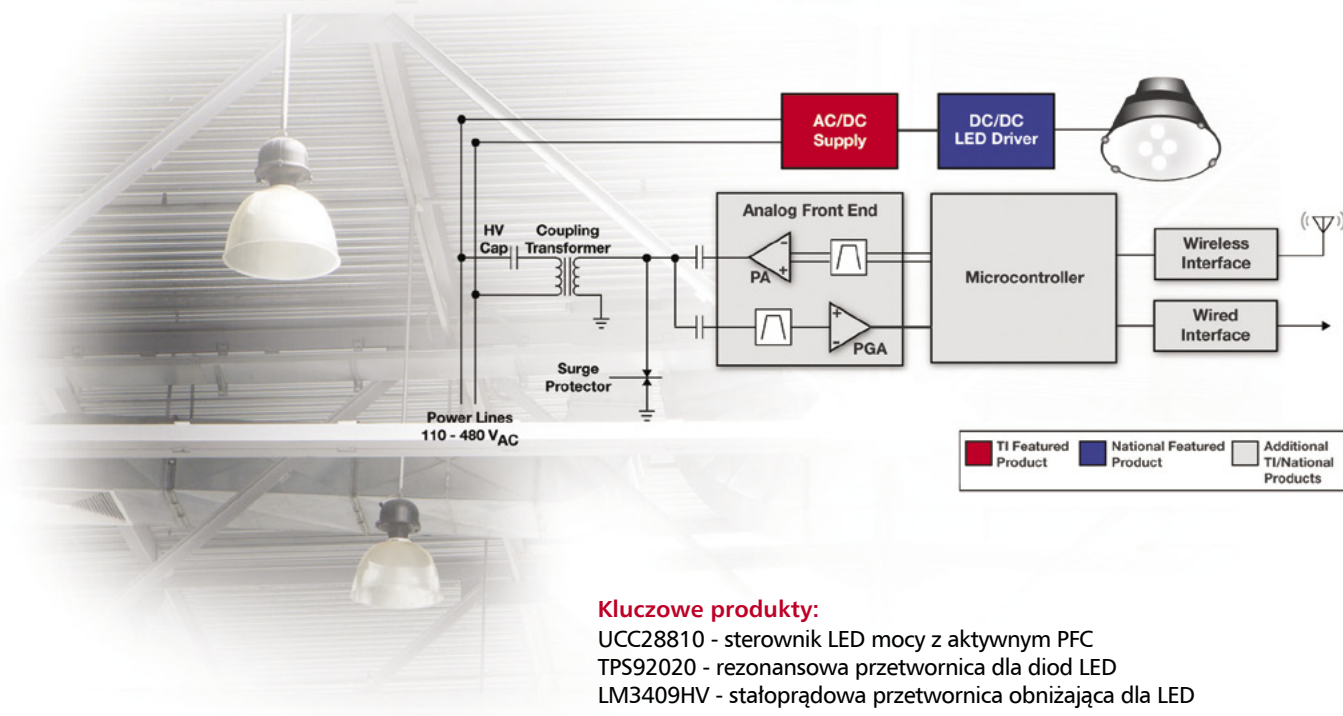
**Marcin Karbowiczek, EP**

Artykuł przygotowano na podstawie materiałów firmy Texas Instruments, których autorem jest Hans Gunter Kremser.

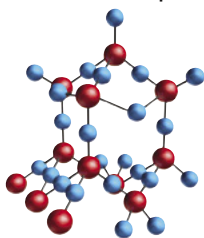
# roz • wia • za • nie

n l. sposób rozstrzygnięcia problemu

## Rozwiązanie TI dla lamp LED High Bay



Możliwość współpracy z firmą, która rozumie wyzwania, przed jakimi stają projektanci jest nieoceniona. Poznaj inteligentne **rozwiązania** firmy Texas Instruments przeznaczone do **lamp LED-owych typu High Bay**, cechujące się najwyższą sprawnością, rewelacyjnym mechanizmem korekcji współczynnika mocy i dużą uniwersalnością. Pozwalają one sterować pracą wielu łańcuchów LED-ów za pomocą pojedynczego projektu zasilacza.



Bezpłatne zestawy próbne dostępne pod adresem: [www.ti.com/highbayled](http://www.ti.com/highbayled)  
Texas Instruments i National Semiconductor - Razem dostarczamy **WIĘCEJ**

 **TEXAS  
INSTRUMENTS**

The stylized red and blue silicon dioxide molecule and the platform bar are trademarks of Texas Instruments. © 2011 TI

Oficjalny dystrybutor firmy Texas Instruments

  
**ARROW ELECTRONICS**

Arrow Electronics Poland Sp. z o.o.  
ul. Rzymowskiego 53, 02-697 Warszawa  
tel. +48 22 55 88 282  
SalesOffice.Warsaw@arroweurope.com