

Internet Rzeczy w przykładach (5)



Sterownik inteligentnej skrzynki na listy

Prezentujemy projekt sterownika inteligentnej skrzynki na listy. Po wrzuceniu listu do skrzynki sterownik powiadomi jej właściciela wysyłając wiadomość e-mail. Do budowy urządzenia wykorzystamy moduł startowy CC3200 LaunchPad.

Skrzynka na listy jest w każdym domu, firmie, instytucji publicznej. Dynamiczny rozwój technologii Internet Rzeczy sprawia, że coraz częściej zamiast skrzynek tradycyjnych montowane są skrzynki inteligentne. Inteligentne skrzynki na listy powiadamiają użytkownika o nowej korespondencji wrzuconej do skrzynki. Użytkownik nie musi niepotrzebnie sprawdzać zawartości skrzynki na listy. Otwiera skrzynkę tylko, gdy jest w niej nowy listy.

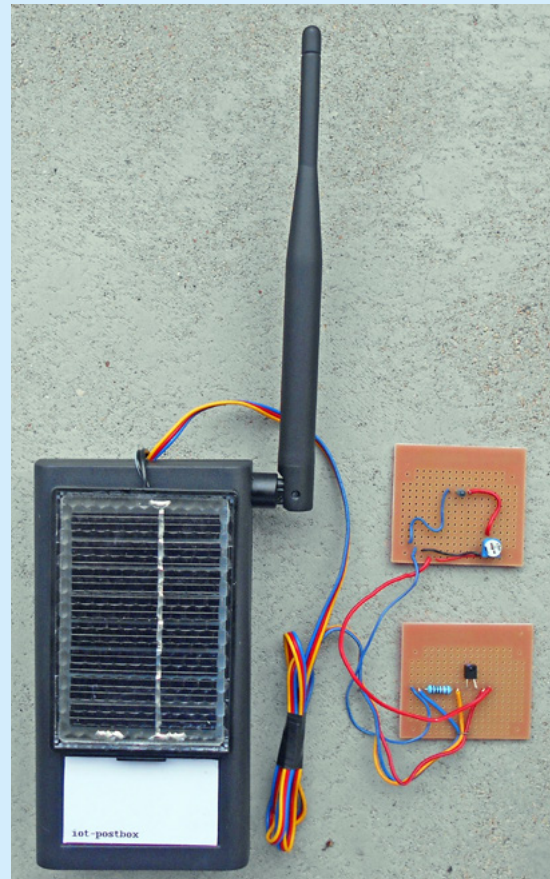
Budowa

Podstawowym elementem sterownika inteligentnej skrzynki na listy jest moduł startowy **CC3200 LaunchPad**. Za wykrywanie listów w skrzynce jest odpowiedzialna bariera optyczna IR. Zbudowano ją z podczerwonej diody nadawczej L-934SF4BT (3 mm, 880 nm, 7 mW) oraz z fototranzystora QSE113 (50°, 880 nm). Sterownik jest zasilany z dwóch akumulatorek AA (NiMH o napięciu 1,2 V i pojemności 2500 mAh każdy). Akumulatorki są ładowane za pomocą niewielkiego fotoogniwa. Moduł CC3200 LaunchPad oraz akumulatorki zostały umieszczone w obudowie Z-44 o wymiarach 150 mm×80 mm×33 mm. Na obudowie zamontowano fotoogniwo (3 V/80 mA) oraz zewnętrzną antenę Wi-Fi (TRF1002 pracującą w paśmie 2,4 GHz, o zysku energetycznym 5 dBi). Elementy bariery optycznej zostały wyprowadzone na zewnątrz sterownika (docelowo montaż w skrzynce na listy). Wygląd urządzenia pokazano na **fotografii 1**.

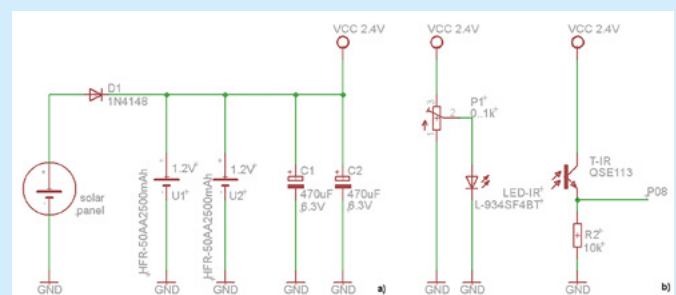
Diody nadawczą IR włączono w kierunku przewodzenia. Dioda jest cały czas włączona. Prąd przewodzenia diody ogranicza zamontowany potencjometr. Kolektor fototranzystora IR dołączono do napięcia zasilania, a emiter do masy. Prąd emitera ogranicza zamontowany rezystor o wartości 10 kΩ. Do emitera jest dołączone wejście mikrokontrolera CC3200 (wyprowadzenie P08). Akumulatorki połączono szeregowo, dzięki utworzono ogniwo zasilające o napięciu 2,4 V i pojemności 2500 mAh. Na wyjściu ogniwa zasilającego zamontowano dwa kondensatory o pojemności 470 μF. Mają one za zadanie buforowanie spadków napięcia zasilania. Panel słoneczny został podłączony bezpośrednio do ogniwa zasilania. Schemat ideowy dodatkowych obwodów pokazano na **rysunku 2**.

Budowę sterownika rozpoczynamy od zmian w module CC3200 LaunchPad. Aby zmniejszyć pobór prądu modułu, odlutowujemy diody D1 oraz D4. Żeby aktywować zewnętrzną antenę Wi-Fi usuwamy rezystor

R111 oraz wykonujemy punkt lutowniczy w miejscu rezystora R110 (**rysunek 3**). Po dokonaniu zmian, do złącza J18 dołączamy antenę zewnętrzną Wi-Fi. Następnie do wejścia P08 mikrokontrolera CC3200 doprowadzamy



Fotografia 1. Sterownik inteligentnej skrzynki na listy



Rysunek 2. Schemat elektryczny urządzenia: a) moduł zasilania, b) bariera optyczna

Do poprawnej pracy urządzenia pomiarowe z modulem CC3200 LaunchPad wymaga dostępu do sieci Wi-Fi (2,4 GHz, standard IEEE 802.11 b/g/n). Należy zapewnić „widoczność” skrzynki i Access Point. Punktem dostępu do sieci może być router, komputer PC, telefon komórkowy. Parametry transmisji (nazwa SSID dla Access Point, szyfrowanie transmisji, ew. hasło dostępu do Access Point) wpisuje się w plikach konfiguracyjnych projektu.

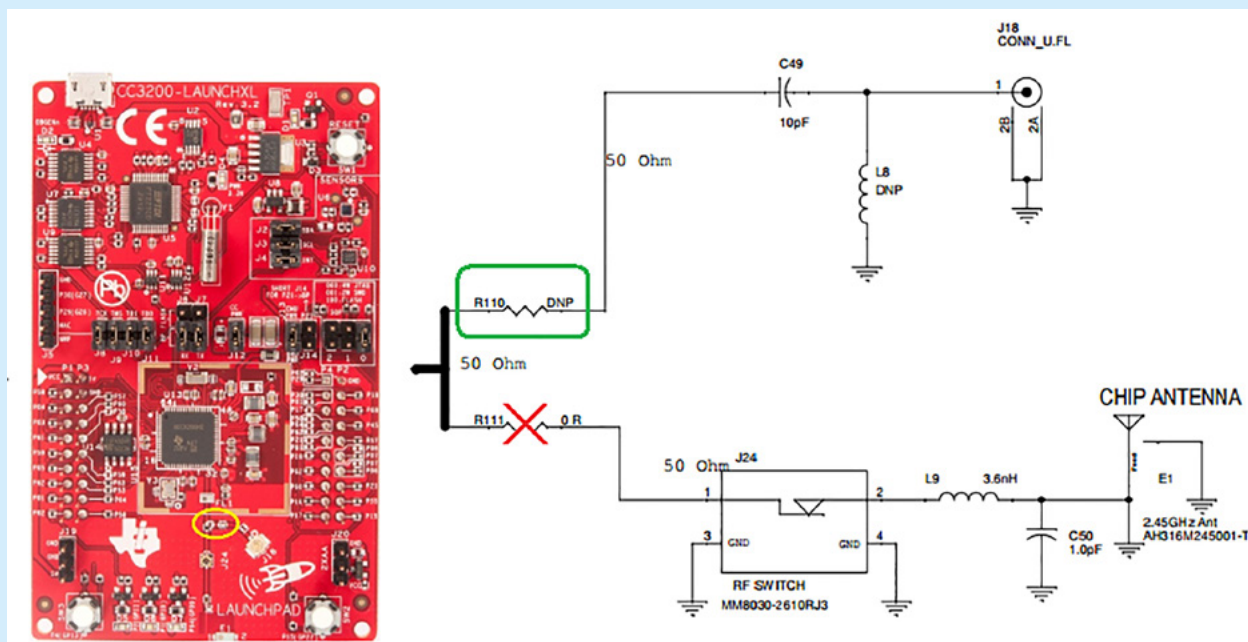
sygnał z fototranzystora IR. Usuwamy zworkę z złącza J13 i do wyprowadzeń Vcc oraz GND złącza J20 doprowadzamy zasilanie urządzenia.

W obudowie urządzenia montujemy moduł CC3200 LaunchPad, akumulatora i kondensatory buforujące. Na obudowie montujemy mini panel słoneczny i antenę Wi-Fi. Elementy bariery optycznej montujemy wewnątrz skrzynki na listy. Diodę nadawczą IR oraz fototranzystor IR montujemy na spodzie skrzynki na listy na jej przeciwległych ścianach. Elementy bariery optycznej muszą być skierowane ku sobie (dioda nadawcza w kierunku

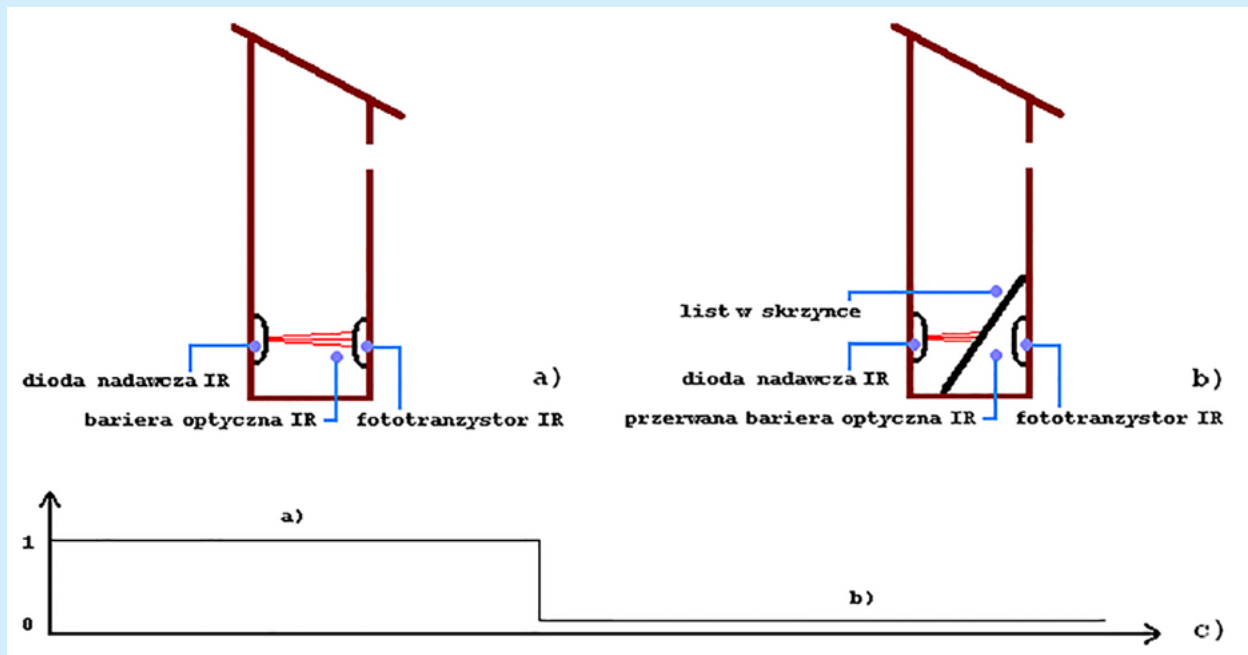
fototranzystora). Wówczas fototranzystor jest otwarty, a na wejściu P08 mikrokontrolera CC3200 występuje poziom wysoki. Gdy do skrzynki zostanie wrzucony list, bariera optyczna zostanie przerwana, fototranzystor zatkany, a na wejście P08 mikrokontrolera CC3200 zostanie wyzerowane. Opisaną zasadę działania zilustrowano na rysunku 4.

Hibernacja

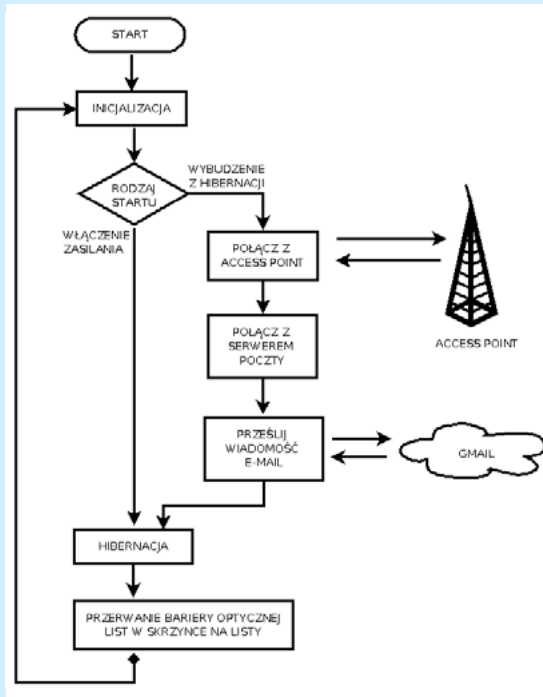
W sterowniku skrzynki na listy został oprogramowany tryb hibernacji mikrokontrolera CC3200. W trybie hibernacji pobór prądu mikrokontrolera wynosi jedynie około 4 μ A. Wyłączony jest rdzeń procesora, nie działają moduły urządzeń peryferyjnych, nie jest odświeżana pamięć RAM (za wyjątkiem dwóch rejestrów 32-bitowych). Aktywny jest jedynie 48-bitowy licznik *Slow Clock Counter* taktowany za pomocą rezonatora „zegarkowego” o częstotliwości 32768 Hz oraz wybrane wejścia mikrokontrolera



Rysunek 3. Podłączenie zewnętrznej anteny Wi-Fi. Zmiany w module CC3200 LaunchPad



Rysunek 4. Skrzynka na listy przekrój boczny a) brak listu w skrzynce b) list w skrzynce c) stan wejścia P08



Rysunek 5. Uproszczony schemat pracy sterownika

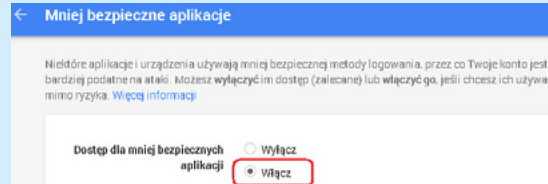
(2, 4, 11, 13, 17, 24) tzw. wejścia *HibWakeUp*. Ze stanu hibernacji mikrokontroler może być wyprowadzony przez licznik oraz przez zmianę poziomu na wejściu *HibWakeUp*. Po wyjściu z trybu hibernacji mikrokontroler jest ponownie uruchamiany. Kod programu jest uruchamiany od początku (jak po zerowaniu).

W oprogramowaniu sterownika skrzynki na listy mikrokontroler CC3200 jest wyprowadzany ze stanu hibernacji za pomocą zmiany poziomu na wejściu *HibWakeUp* – licznik *Slow Clock Counter* jest nieużywany. Użyte wejścia numer 17 (P08), do którego jest doprowadzony fototranzystor IR. Zmiana poziomu logicznego na wejściu z wysokiego (aktywna bariera optyczna, brak listu w skrzynce) na niski (przerwana bariera optyczna, list w skrzynce) sprawia, że mikrokontroler CC3200 przechodzi do trybu aktywnego.

Funkcjonalność

Zadaniem sterownika inteligentnej skrzynki na listy jest informowanie użytkownika o nowych listach w skrzynce. Sterownik informuje użytkownika przesyłając wiadomość e-mail.

Po włączeniu zasilania urządzenia konfigurowane są parametry pracy modułów peryferyjnych mikrokontrolera CC3200 (inicjalizowanie UART i Watchdog, konfigurowanie trybu hibernacji). Następnie mikrokontroler jest wprowadzany w tryb hibernacji. Wrzucenie listu do skrzynki (równoważne przerwaniu bariery optycznej IR) powoduje przejście mikrokontrolera do trybu aktywnego. Inicjowane są parametry pracy modułów peryferyjnych mikrokontrolera. Następnie jest uruchamiana część programu odpowiedzialna za przesłanie wiadomości e-mail. Sterownik łączy z punktem dostępu do sieci *Access Point*, loguje się do serwera poczty SMTP (tu wykorzystalem gmail) i przesyła do użytkownika wiadomość e-mail. Na zakończenie mikrokontroler CC3200 ponownie jest wprowadzany w tryb hibernacji. Jeśli użytkownik wyjmie list ze skrzynki, to bariera optyczna IR zostanie załączona. Wrzucenie nowego listu do skrzynki przerwie



Rysunek 6. Konfiguracja konta pocztowego Gmail

barierę optyczną IR. Mikrokontroler CC3200 obudzi się z uśpienia i rozpocznie wysłanie wiadomości e-mail. Cykl pracy urządzenia powtórzy się. Sposób działania sterownika zilustrowano na **rysunku 5**.

Przy braku połączenia z *Access Point* bądź błędzie podczas wysyłania wiadomości e-mail mikrokontroler bezwarunkowo przechodzi w tryb hibernacji. Nad ciągłością pracy mikrokontroler CC3200 czuwa układ *Watchdog*. Domyślnie timer *Watchdog* jest wyłączony. Po zainicjowaniu czas pracy układu *Watchdog* wynosi 50 sekund.

E-Mail

Wiadomości e-mail do użytkownika są wysyłane przy wykorzystaniu serwera pocztowego Gmail za pomocą protokołu SMTP. Mikrokontroler CC3200 łączy się z serwerem Gmail, loguje się do konta użytkownika poczty i korzystając z protokołu SMTP przesyła wiadomość dla użytkownika skrzynki na listy. W ustawieniach projektu należy podać adres oraz hasło konta e-mail w serwisie Gmail (plik *configure.h*). Dodatkowo, w ustawieniach konta należy włączyć dostęp dla mniej bezpiecznych aplikacji. Opcję, którą należy skonfigurować pokazano na **rysunku 6**.

Oprogramowanie

Oprogramowania sterownika inteligentnej skrzynki na listy wykonano w środowisku CCSv6. Utworzony został projekt o nazwie *iot_postbox*. W projekcie użyto systemu czasu rzeczywistego freeRTOS, frameworka *SimpleLink*, driverów dla CC3200, biblioteki *libmail* do obsługi klienta SMTP (biblioteka w folderze SDK w lokalizacji */netapps/sntp/client*). Dodatkowo, wykorzystano przygotowane przez Texas Instruments interfejsy do obsługi urządzeń peryferyjnych i sieci (uart, udma, wdt, network). Projekt został skonfigurowany zgodnie z opisem publikowanym w poprzednich częściach kursu. Pliki źródłowe projektu dostępne są w materiałach dodatkowych dołączonych do artykułu. W katalogu *source* zostały umieszczone pliki z konfiguracją linii wejścia-wyjścia mikrokontrolera CC3200. Konfiguracja linii wejścia-wyjścia została wygenerowana przy użyciu oprogramowania *Pin Mux Tool*. W katalogach *hardware* i *system* zostały umieszczone pliki źródłowe oprogramowania. W katalogu *hardware* pliki do obsługi modułów sprzętowych mikrokontrolera CC3200 (*UART*, *Watchdog*). W katalogu *system* pliki do obsługi logiki pracy urządzenia (konfiguracja, obsługa sieci, wysyłanie wiadomości e-mail, obsługa trybu hibernacji itp.).

W oprogramowaniu uruchomiony zostały system czasu rzeczywistego freeRTOS. Utworzony został wątek o nazwie *system* (plik *system.c*). W wątku wywoływane są procedury *EmailSendTask* oraz *HibernateEnter*. W procedurze *EmailSendTask* jest zestawiane połączenie z *Access Point* oraz jest uruchamiana obsługa wysyłania wiadomości e-mail. Konfigurowane są parametry serwera poczty (numer portu SMTP, adres IP serwera, metoda

Listing 1. Procedury do obsługi e-mail

```

long SetAdvancedEmailParameters(unsigned long destinationIp)
{
    S1NetAppEmailOpt_t    eMailParameters;
    // Set Email Server Parameters
    eMailParameters.Family = AF_INET;
    eMailParameters.Port   = GMAIL_HOST_PORT;
    eMailParameters.Ip     = destinationIp;
    eMailParameters.SecurityMethod = SL_SO_SEC_METHOD_SSLV3;
    eMailParameters.SecurityCypher = SL_SEC_MASK_SSL_RSA_WITH_RC4_128_MD5;
    return (sl_NetAppEmailSet(SL_NET_APP_EMAIL_ID, NETAPP_ADVANCED_OPT, \
                             sizeof(S1NetAppEmailOpt_t), (unsigned char*)&eMailParameters));
}

long SetSourceEmail()
{
    S1NetAppSourceEmail_t sourceEmailId;
    memcpy(sourceEmailId.Username, GMAIL_USER_NAME, strlen(GMAIL_USER_NAME)+1);
    return (sl_NetAppEmailSet(SL_NET_APP_EMAIL_ID, NETAPP_SOURCE_EMAIL, \
                             strlen(GMAIL_USER_NAME)+1, (unsigned char*)&sourceEmailId));
}

long SetSourceEmailPassword()
{
    S1NetAppSourcePassword_t sourceEmailPwd;

    memcpy(sourceEmailPwd.Password, GMAIL_USER_PASS, strlen(GMAIL_USER_PASS)+1);
    return (sl_NetAppEmailSet(SL_NET_APP_EMAIL_ID, NETAPP_PASSWORD, \
                             strlen(GMAIL_USER_PASS)+1, (unsigned char*)&sourceEmailPwd));
}

long SetDestinationEmail()
{
    S1NetAppDestination_t destEmailAdd;
    memcpy(destEmailAdd.Email, RCPT RFC, strlen(RCPT RFC)+1);
    return (sl_NetAppEmailSet(SL_NET_APP_EMAIL_ID, NETAPP_DEST_EMAIL, \
                             strlen(RCPT RFC)+1, (unsigned char *)&destEmailAdd));
}

long SetEmailSubject()
{
    S1NetAppEmailSubject_t emailSubject;
    memcpy(emailSubject.Value, EMAIL_SUB, strlen(EMAIL_SUB)+1);
    return (sl_NetAppEmailSet(SL_NET_APP_EMAIL_ID, NETAPP_SUBJECT, \
                             strlen(EMAIL_SUB)+1, (unsigned char *)&emailSubject));
}

long SetEmailMessage(char * message)
{
    char const value[] = „Hi\r\n” \
                        „From this site your postbox\r\n” \
                        „\r\n” \
                        „New letter is Available !”;
    if(strlen(value)>63) return -1;
    memcpy(message, value, strlen(value));
    *(message + strlen(value)) = 0;
    return 1;
}

```

kodowania transmisji, algorytm kodowania transmisji). Jest ustawiany adres oraz hasło konta Gmail używanego do wysyłania wiadomości. Wprowadzane są adres odbiorcy, tytuł oraz treść wiadomości e-mail. Kod źródłowy procedur do obsługi e-mail pokazano na **listingu 1**. W procedurze *HibernateEnter* jest włączany tryb uśpienia.

W programie można aktywować obsługę timera Watchdog. Czas pracy Watchdoga wynosi 50 sekund. Przy „zawieszeniu się” programu układ Watchdog wykona restart mikrokontrolera CC3200.

Uruchomienie

Projekt urządzenia pomiarowego dostępny jest w materiałach dodatkowych dołączonych do artykułu (folder *iot_potboxes*). Kopiujemy katalog z projektem do lokalizacji *c:/ti/ep/*. Następnie uruchamiamy oprogramowanie *Code Composer Studio* i importujemy projekt (*Project*

Podczas pracy sterownik inteligentnej szafy na ubrania wysyła komunikaty serwisowe (procedura *DBG_PRINT*). Komunikaty są wysyłane za pomocą UART. Aby odebrać informacje wysyłane przez sterownik należy dołączyć moduł *LaunchPad* do portu USB komputera PC, a zworki *JP6* i *JP7* ustawić w pozycji *Flash*. Wówczas w systemie operacyjnym *Windows* pod nazwą *CC3200LP Dual Port* zostanie aktywowany port *COM* do obsługi modułu *LaunchPad*. Parametry transmisji UART to: 115200, n, 8, 1.

→ *Import CCS Projects*). W kolejnym kroku zmieniamy ustawienia oprogramowania. W pliku konfiguracyjnym *configure.h* ustawiamy nazwę *SSID* dla *Access Point*, hasło dostępu do *Access Point* oraz algorytm szyfrowania transmisji danych. Podajemy nazwę i hasło do konta Gmail, jeśli nie mamy konta pocztowego Gmail, to należy je założyć. Podajemy adres odbiorcy wiadomości. Żeby aktywować układ *Watchdog* w opcjach projektu w zakładce *Predefined Symbols* z pola *Undefine NAME* usuwamy wpis *WATCHDOG_ON* a następnie do pola *Pre-define NAME* dodajemy wpis *WATCHDOG_ON*. Po dokonaniu zmian kompilujemy projekt (*Project* → *Build All*). W wyniku kompilacji jest tworzony plik binarny o nazwie *iot_postbox.bin*. Oprogramowanie wgrujemy korzystając z aplikacji *CCS UniFlash* (opis w poprzednich częściach kursu). Podczas programowania w złączu *J13* należy zamontować zworkę.

Następnie w skrzynce na listy montujemy elementy bariery optycznej. Moc diody nadawczej IR regulujemy za pomocą potencjometru. Im większa moc diody nadawczej, a co za tym idzie – zasięg działania bariery optycznej, tym większy pobór prądu (dla 100 Ω zasięg działania bariery optycznej to około 4 centymetrów, pobór prądu to około 7 mA). Żeby zmniejszyć pobór prądu urządzenia elementy bariery optycznej należy umieścić możliwie jak najbliżej siebie oraz za pomocą potencjometru

(zwiększając rezystancję) zmniejszyć moc diody nadawczej. Zaprogramowany moduł CC3200 LaunchPad umieszczamy w obudowie. Usuwamy zworę z złącza J13. Montujemy panel słoneczny. Przyłączamy elementy bariery optycznej IR i włączamy zasilanie. Na zakończenie montujemy sterownik na ogrodzeniu. Przykład instalacji sterownika pokazano na **fotografii 7**.

Podsumowanie

W kolejnym, ostatnim odcinku kursu zaprezentujemy projekt serwera http. Zadaniem serwera będzie sterowanie automatyką domu. Do budowy serwera wykorzystamy moduł startowy CC3200 LaunchPad. Działanie urządzenia zostanie zoptymalizowane pod kątem poboru mocy.

Łukasz Krysiewicz, EP



Fotografia 7. Sterownik inteligentnej skrzynki na listy podczas pracy

Wydanie specjalne

„Raspberry Pi”

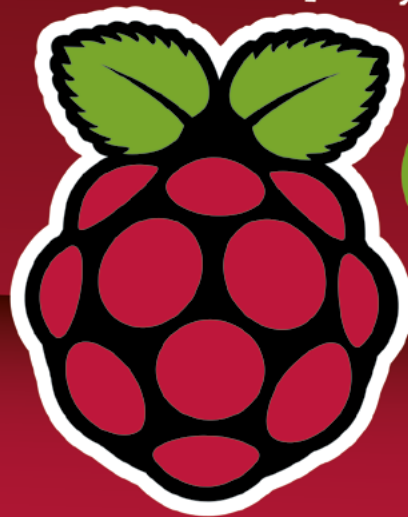
to polski przekład światowego bestsellera na temat słynnego minikomputera

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 UMIĘJĘTNOŚCI

✓ 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 w salonach prasowych oraz na www.UlubionyKiosk.pl

(przesyłka GRATIS)