

C2000 Piccolo LanuchPad (5)

Łatwa obsługa interfejsu I²C

Moduł peryferyjny I2C (Typ0) układów procesorowych F2802x/3x Piccolo jest bardzo rozbudowany. Aby go skutecznie zastosować trzeba dobrze zrozumieć jego konstrukcję i zasadę działania. W artykule jest opisane ćwiczenie praktyczne z zastosowaniem biblioteki driverlib pakietu programowego controlSUITE v.3.x oraz środowiska Code Composer Studio v5.x i zestawu ewaluacyjnego C2000 Piccolo LaunchPad. Celem ćwiczenia jest poznanie i zastosowanie obsługi modułu peryferyjnego I2C układu procesorowego serii F2802x Piccolo.

Do tworzenia w środowisku CCSv5 programów przeznaczonych dla procesorów rodziny F2802x Piccolo jest potrzebny pakiet programowy controlSUITE. Zawiera on oprogramowanie firmware, biblioteki, opisy zestawów sprzętowych oraz projekty przykładowe dla wszystkich serii procesorów rodziny C2000.

Użyto przykładowego programu z projektu *Example_F2802xI2C_eeprom* pakietu *controlSUITEv3.x*. Ćwiczenie jest zorganizowane tak, że działania są wykonywane w kolejnych punktach i krokach uzupełnionych o szczegółowe opisy. Do wykonania ćwiczenia jest potrzebny komputer z zainstalowanym (darmowym) oprogramowaniem:

- Środowisko *Code Composer Studio* v5.3.0 firmy Texas Instruments [2, 4, 8, 9, 11]. Umożliwia tworzenie programów przeznaczonych dla procesorów serii Piccolo TMS320F2802x.
- Pakiet programowy controlSUITE v3.2 firmy Texas Instruments [2, 4, 8, 9, 11]. Zawiera oprogramowanie „firmware”, biblioteki, opisy zestawów sprzętowych oraz projekty przykładowe dla wszystkich serii procesorów rodziny C2000.

Platforma sprzętowa obejmuje następujące elementy:

- Zestaw ewaluacyjny *C2000 Piccolo LaunchPad* firmy Texas Instruments z układem procesorowym Piccolo TMS320F28027 firmy Texas Instruments (zawiera kabel USB-A USB-mini) [1, 7].
- Moduł sprzętowy KAmoDEXP2 firmy Kamami [13] z układem scalonym MCP23008 ekspandera szyny I²C firmy Microchip [15].
- Przewody połączeniowe, standard złącza IDC (np. CAB_A firmy Kamami) [14].

Instalowanie i użytkowanie środowiska CCSv5 oraz pakietu programowego controlSUITEv3 zostało opisane w artykule [2].

W folderze *C:\home_dir* komputera zostanie utworzony nowy folder *work_I2C*. Wymagane są prawa dostępu (zapisu i modyfikacji) dla tej ścieżki dyskowej. Możliwe jest umieszczenie foldera *home_dir* na innym wolumenie dyskowym z prawami dostępu.

Opisy

Dane techniczne i parametry elektryczne modułu peryferyjnego I2C układu procesorowego serii F2802x

Dodatkowe informacje:

Dotychczas w EP na temat zestawu ewaluacyjnego C2000 Piccolo LaunchPad:

- „Zestaw ewaluacyjny C2000 Piccolo LaunchPad”, EP 01/2013
- „C2000 Piccolo LanuchPad (1) – Pierwszy program w środowisku programowym CCS v5”, EP 02/2013
- „C2000 Piccolo LanuchPad (2) – Łatwe programowanie z pakietem controlSUITE”, EP 03/2013
- „C2000 Piccolo LanuchPad (3) – Łatwe programowanie do pamięci Flash”, EP 04/2013
- „C2000 Piccolo LanuchPad (4) – Łatwa obsługa szyny SPI”, EP 05/2013

Pliki źródłowe (CD/FTP):

<ftp://ep.com.pl>, user: 52617, pass: 30lct328

Piccolo_F2802x_KAmoDEXP2.c, Piccolo_F2802x_KAmoDEXP2.h oraz Example_2802xI2C_KAmoDEXP2.c

Piccolo są zamieszczone w dokumencie Texas Instruments TMS320F28027, TMS320F28026, TMS320F28023, TMS320F28022, TMS320-F28021, TMS320F280200, Piccolo Microcontrollers, Data Sheet [7].

Opis modułu peryferyjnego I2C układu procesorowego serii F2802x Piccolo jest zamieszczony w dokumencie *TMS320x2802x, 2803x Piccolo Inter-Integrated Circuit (I2C) Module Ref. Guide* [12].

Dokładne omówienie budowy modułu peryferyjnego I2C układu procesorowego serii F2802x Piccolo jest zamieszczone w książce Henryk A. Kowalski, „Procesory DSP dla praktyków”, BTC, Warszawa, 2011 [5].

Dokładne omówienie przykładowego projektu *Example_F2802xI2C_eeprom* jest zamieszczone w książce Henryk A. Kowalski, „Procesory DSP w przykładach”, BTC, Warszawa, 2012 [6].

Skonfigurowanie modułu KAmoDEXP2

W module KAmoDEXP2 został zastosowany układ scalony ekspandera MCP23008 firmy Microchip pracujący w standardzie łącza I²C [15]. Do poprawnej pracy programu z ćwiczenia jest wymagana podstawowa (standardowa) konfiguracja przełączników płytki drukowanej modułu KAmoDEXP2 [13]:

- Założona zwora JP0 w pozycji „0” (JP0.1-JP0.2). Oznacza to bit adresowy A0=0.
- Założona zwora JP1 w pozycji „0” (JP1.1-JP1.2). Oznacza to bit adresowy A1=0.
- Założona zwora JP2 w pozycji „0” (JP2.1-JP2.2). Oznacza to bit adresowy A2=0.

Niestety, moduł KAmoDEXP2 nie zawiera rezystorów podciągających dla sygnałów SCL i SDA szyny I²C. Rezystory o rezystancji ok 4,7 kΩ należy dołączyć zewnętrznie (np. na złączu CON1 lub CON2).

Dołączanie modułu KAmoDEXP2 do zestawu C2000 Piccolo LaunchPad

Dołącz moduł KAmoDEXP2 do zestawu ewaluacyjnego C2000 Piccolo LaunchPad.

Uwaga! Połączenia należy wykonywać bez włączonego zasilania, czyli przy odłączonym kablu USB. Najlepiej najpierw połączyć masę obu płytek drukowanych. Zmniejszy to niebezpieczeństwo uszkodzenia układów ze względu na ładunki elektrostatyczne. Połączenia należy wykonywać przewodami z końcówkami zgodnymi ze standardem złącza IDC [14].

Należy podać masę, zasilanie 3,3 V oraz dwa sygnały łącza I²C, dwa dodatkowe sygnały sterujące i cztery sygnały danych (tabela 1).

Wybór skonfigurowania wyprowadzeń układu procesorowego F28027 Piccolo został zaznaczony kolorem.

Należy zauważyć, że wyprowadzenie GPIO32 układu procesorowego F28027 Piccolo jest połączone z wyprowadzeniem GPIO16 na płytce drukowanej zestawu ewaluacyjnego C2000 Piccolo LaunchPad. Tak samo jest połączone wyprowadzenie GPIO33 z wyprowadzeniem GPIO17. Dlatego bez modyfikacji płytki drukowanej nie można w tym projekcie używać wyprowadzenia GPIO16 i GPIO17.

Konfigurowanie zestawu C2000 Piccolo LaunchPad

Po zainstalowaniu środowiska CCSv5 [2] można pierwszy raz dołączyć zestaw ewaluacyjny C2000 Piccolo LaunchPad [1] kablem USB do wolnego portu USB komputera. System Windows automatycznie rozpoznaje układ. Zostaną zainstalowane sterowniki systemu Windows dla emulatora XDS100v2 [6]. Należy poczekać aż system potwierdzi, że sprzęt jest gotowy do pracy.

Do poprawnej pracy programu przykładowego wymagana jest podstawowa (standardowa) konfiguracja przełączników płytki drukowanej zestawu [1]:

- Założone zwory JP1 („3V3”), JP2 („5V”) i JP3 („GND”). Oznacza to zasilanie układu procesorowego z gniazda USB.
- Przełącznik S1 („Boot”) skonfigurowany następująco: S1.1 - do góry (ON), S1.2 - do góry, S1.3 - do góry. W praktyce oznacza to bootowanie z pamięci Flash.
- Przełącznik S4 („Serial”) skonfigurowany w pozycji do góry (ON). Oznacza dołączenie portu UART układu procesorowego do układu emulatora, a tym samym do wirtualnego portu COM na PC.

Pliki źródłowe

Do wykonania ćwiczenia potrzebny jest kod źródłowy zawarty w dodatkowych plikach *Piccolo_F2802x_KAmoDEXP2.c* oraz *Piccolo_F2802x_KAmoDEXP2.h*. Ponadto zostanie zmodyfikowany kod w głównym pliku *Example_2802xI2C_eeprom.c* projektu przykładowe-

Tabela 1 Dołączenia modułu KAmoDEXP2 do zestawu C2000 Piccolo LaunchPad

Płytką KAmoDEXP2	Płytką C2000 Piccolo LaunchPad
Con3.4 (GND)	J3.2 (GND)
Con3.1 (V+)	J3.1 (+3.3V)
Con3.2 (SCL)	J2.7 (SCLA/GPIO33)
Con3.3 (SDA)	J2.6 SDAA/GPIO32)
Con5.3 (INT)	J1.7 (GPIO18)
Con5.2 (RES)	J1.8 (AIO2/ADCINA2)
Con4.1 (IO0)	J6.1 (GPIO0)
Con4.3 (IO1)	J6.2 (GPIO1)
Con4.5 (IO2)	J6.3 (GPIO2)
Con4.7 (IO3)	J6.4 (GPIO3)


go. Wersja końcowa zmian jest zamieszczona w pliku *Example_2802xI2C_KAmoDEXP2.c*.

Uruchomienie środowiska CCSv5

1. W oknie *Workspace* wpisz ścieżkę i nazwę folderu roboczego. Powinna być ona krótka i musi być zlokalizowana w miejscu, dla którego są uprawnienia dostępu (zapisu). Dla indywidualnej pracy proponowana jest ścieżka `<C:/home_dir>`. Dla tego ćwiczenia proponowana jest nazwa folderu *work_I2C*.

Po kliknięciu na przycisk *OK* okna *Workspace Launcher* otwierane jest okno startowe środowiska CCSv5 (i ładowane są poszczególne elementy środowiska). Można to obserwować na pasku postępu. Może to trwać dosyć długo i należy koniecznie poczekać na zakończenie inicjalizacji środowiska przed rozpoczęciem dalszej pracy.

Projekty przykładowe pakietu controlSUITE

W oknie *TI Resource Explorer* perspektywy *CCS Edit* pokazywana jest strona *Welcome* (w html). Zawiera ona graficznie menu główne. Istotne informacje są zgrupowane na stronie *Home*. Można ją otworzyć po kliknięciu oknie *TI Resource Explorer* na ikonkę *Home* . Po kliknięciu na odnośnik *Examples* pokazywane jest po lewej stronie okna drzewo dokumentacji i dostępnych projektów przykładowych. Jeśli pokazywana jest tylko jedna linia *controlSUITE* z gałęzią *English* to udostępni ona dokumentację pakietu. Aby dodać przykłady należy na stronie *Home* kliknąć na odnośnik *Configure Resource Explorer*. W oknie dialogowym *Package Configuration* trzeba kliknąć na *Add*, wskazać folder `C:\ti\controlSUITE` i kliknąć *OK*. Nazwa *controlSUITE* pojawi się w oknie wyboru i należy kliknąć *OK*. Po dłuższej chwili pojawi się w drzewie okna *TI Resource Explorer* druga linia *controlSUITE* zawierająca pozycje: *development kits*, *device_support* oraz *libs*.

Zastosowanie projektu Example_F2802xI2C_eeprom

2. Dla pracy z rodziną F2802x Piccolo rozwiń w oknie *TI Resource Explorer* drugą pozycję *controlSUITE*. Następnie należy rozwinąć drzewo *controlSUITE* → *device_support* → *f2802x* → *v210* → *f2802x_examples*. Potem trzeba kliknąć na nazwę wybranego projektu *Example_F2802xI2C_eeprom*. W prawym oknie zostanie wyświetlona instrukcja jak krok po kroku zbudować i uruchomić projekt.

Krok1: Importowanie projektu Example_F2802xI2C_eeprom do CCS

Krok1 umożliwia zaimportowanie wybranego projektu do CCSv5.

3. Kliknij na odnośnik kroku 1.

Po poprawnym wykonaniu importowania drzewo projektu pojawia się w oknie *Project Explorer* i zielony znaczek ✓ jest pokazywany na prawo od linii nazwy kroku.

Projekt *Example_F2802xI2C_eeprom* został zaimportowany z kopiowaniem pliku *Example_2802xI2C_eeprom.c* do foldera roboczego projektu.

Krok2: Budowanie projektu Example_F2802xI2C_eeprom

Krok2 umożliwia wykonanie budowania wybranego projektu.

4. Kliknij na odnośnik kroku 2.

W oknie *Console* pokazywane są bieżące informacje o postępie budowania. W oknie *Problems* pokazywane są opisy błędów, ostrzeżeń i informacji. Po poprawnym wykonaniu budowania zielony znaczek ✓ pokazywany jest na prawo od linii nazwy kroku (rysunek 6).

5. W oknie *Project Explorer* rozwiń drzewo projektu i kliknij na jego nazwę. Został zbudowany projekt w konfiguracji budowania o nazwie RAM.

Budowanie projektu *Example_F2802xI2C_eeprom* zostało zakończone poprawnie. Został utworzony wynikowy plik binarny *Example_2802xI2C_eeprom.out*. Jednak Zostały zgłoszone ostrzeżenia. Na razie są one nieistotne.

Krok3: Definiowanie konfiguracji sprzętowego systemu docelowego

Krok3 umożliwia zdefiniowanie konfiguracji sprzętowego systemu docelowego dla projektu. Pole *Connection* pokazuje typ „none”.

6. Kliknij na odnośnik kroku 3.

W oknie dialogowym *Debugger Configuration* rozwiń listę i wybierz pozycję *Texas Instruments XDS100v2 USB Emulator*. Kliknij OK. Pole *Connection* pokazuje teraz typ *Texas Instruments XDS100v2 USB Emulator*. Zielony zna-

czek ✓ pokazywany jest na prawo od linii nazwy kroku. Utworzony plik konfiguracji sprzętowej TMS320F28027.ccxml jest teraz pokazany w gałęzi /targetConfigs drzewa projektu w oknie *Project Explorer*. Jest on ustawiony jako Active/Default (aktywny i domyślny).

Krok4: Uruchamianie sesji debugowej dla projektu Example_F2802xI2C_eeprom

Krok4 umożliwia uruchomienie sesji debugowej dla projektu. Dotychczas praca środowiska CCSv5 nie wymagała fizycznej obecności sprzętu docelowego. Wykonanie kroku 4 wymaga wcześniejszego dołączenia zestawu ewaluacyjnego *C2000 Piccolo LaunchPad* do komputera z zainstalowanym środowiskiem CCSv5.

7. Kliknij na odnośnik kroku 4.

Kliknięcie na odnośnik kroku 4 powoduje automatyczne rozpoczęcie sesji debugowej – podobnie jak po przyciśnięciu przycisku *Debug*.

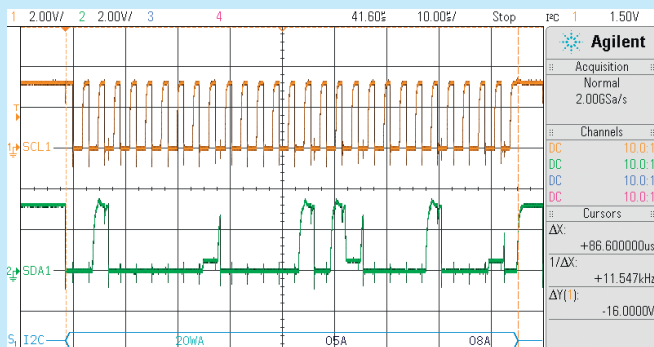
Modyfikacja skonfigurowania wyprowadzeń I2C procesora Piccolo F28027

Z funkcji *main()* wywoływana jest funkcja *InitI2CGpio()* inicjalizacji modułu peryferyjnego I2C. W projekcie domyślnie zostały zastosowane wyprowadzenia GPIO28 i GPI29 układu procesorowego F28027 Piccolo. Te wyprowadzenia są dołączone do izolatora galwanicznego konwertera USB-RS232 na płycie zestawu ewaluacyjnego *C2000 Piccolo LaunchPad*. Jednak można to zmienić. 8. Przełącz się do perspektywy *CCS Edit*. Zauważ, że ikonka na lewo od nazwy pliku *F2802x_I2C.c* zawiera nałożoną strzałkę. Sygnalizuje ona, że plik jest tylko dołączony do projektu. Kliknij prawym klawiszem na nazwę pliku i wybierz *Properties*. Pokazana jest dokładna lokalizacja pliku. Wszelkie zmiany wprowadzone w tym pliku będą widoczne przez inne projekty tego pakietu firmware. Usuń plik *F2802x_I2C.c* z projektu. W systemie Windows skopiuj plik *F2802x_I2C.c* do foldera projektu *C:/home_dir/work_I2C/Example_F2802xI2C_eeprom*.

9. W oknie *Project Explorer* dwukrotnie kliknij na nazwę pliku *F2802x_I2C.c*. Odkomentuj wszystkie linie kodu dotyczące wyprowadzeń GPIO32 i GPIO33. Opatrz komentarzem wszystkie linie kodu dotyczące wyprowadzeń GPIO28 i GPI29.

Zauważ, że nazwa pliku na zakładce okna edycji jest poprzedzona gwiazdą. Sygnalizuje ona, że zostały wykonane zmiany i nie są one zapisane do pliku. Aby je zachować należy kliknąć na przycisk *Save*. Również kliknięcie na przycisk *Build* powinno spowodować zapisanie wszystkich plików ze zmianami przed rozpoczęciem budowania projektu. Tak się typowo dzieje. Czasami występują jednak kłopoty z plikami nagłówkowymi.

Skorzystanie z podglądu stanu pól bitowych rejestrów sterowania modułów peryferyjnych wymaga dołączenia do projektu pliku definicyjnego struktur modelu bezpośredniego dostępu do rejestrów. Zestawienie nazw zmiennych (struktur) udostępnianych przez plik jest podane w tab. 2.12 (str. 32) dokumentacji pakietu firmware [8]. Dla modułu peryferyjnego I2C jest udostępniana struktura rejestrów sterowania *I2cRegs*.



Rysunek 1. Wpisywanie przez moduł peryferyjny I2C układu procesorowego F28027 Piccolo nowej zawartości 0x08 do rejestru konfiguracyjnego IOCON (adres 0x05) układu scalonego MCP23008 pod adresem 0x20 na szynie I²C.

10. Wykonaj samo budowanie projektu (bez ponownego startowanie sesji debugowej). Kliknij na przycisk *Build*. Nie używaj przycisku *Debug*. Na pytanie czy załadować plik wynikowy kodu przyciśnij przycisk *Yes*.

11. Przełącz się do perspektywy *CCS Debug*. Zauważ w oknie edytora, że praca program została zatrzymana na pierwszej linii kodu funkcji *main()*.

12. Otwórz okno *Disassembly* z menu *View* → *Disassembly*.

Wgląd w projekt **Example_F2802xI2C_eeprom**

13. Zapoznaj się z komentarzem na początku pliku *Example_2802xI2C_eeprom.c*.

Krótki opis projektu przykładowego oraz założenia i wymagania sprzętowe są zamieszczone na początku głównego pliku każdego projektu przykładowego z pakietu programowego controlSUITE.

Program dobrze działa z dołączoną pamięcią EEPROM. Szczegółowe omówienie działania przykładowego projektu jest zamieszczone w książce [6].

Zbudowanie projektu **F2802xI2C_KAmodEXP2**

14. Zmień nazwę projektu.

W oknie *Project Explorer* kliknij prawym klawiszem myszy na linię nazwy projektu *Example_F2802xI2C_eeprom*. Z podręcznego menu wybierz z menu *Rename*, wpisz nową nazwę *F2802xI2C_KAmodEXP2* i kliknij *OK*. Zostanie również zmieniona nazwa folderu projektu w folderze roboczym (C:\home_dir\work_KAmodEXP2\F2802xI2C_KAmodEXP2).

15. Dodaj pliki *KAmodEXP2.c* oraz *KAmodEXP2.h* do folderu projektu C:\home_dir\work_KAmodEXP2\F2802xI2C_KAmodEXP2. Oba pliki zostaną automatycznie dodane do projektu.

16. Dodaj na początku pliku *Example_2802xI2C_eeprom.c* dodatkowe definicje z **listingu 1**.

17. W pliku *Example_2802xI2C_eeprom.c* dodaj po linii kodu *InitI2CGpio()*; skonfigurowanie dodatkowych wyprowadzeń GPIO układu procesorowego F28027 Piccolo z **listingu 2**.

Wobec małej liczby dostępnych wyprowadzeń cyfrowych GPIO układu procesorowego F28027 Piccolo zostało użyte wyprowadzenie analogowe ADCINA4 skonfigurowane jako cyfrowe AIO2 i ustawione jako wyjście z poziomem wysokim.

18. Dodaj na początku funkcji *main()* w pliku *Example_2802xI2C_eeprom.c* dodatkowe definicje

```
Uint16 Output = 1, Input=0;
int16 tmp;
```

19. Zamień kod programu rozpoczynając od linii następnej po *EINT*; aż do końca funkcji *main()* na kod pokazany na **listingu 3**.

20. Zmodyfikuj kod funkcji *I2CA_Init(void)*; do postaci pokazanej na **listingu 4**.

21. Zakomentuj prawie cały kod funkcji *i2c_int1a_isr(void)*. Pozostaw aktywną tylko ostatnią linię kodu *PieCtrlRegs.PIEACK.all = PIEACK_GROUP8*;

```
Listing 1. Dodatkowe definicje dla funkcji main() w pliku
Example_2802xI2C_eeprom.c//
//KAmodEXP2 modifications
#include „KAmodEXP2.h”
// MCP23008 hardware address
//Hardware address of KAmodEXP2 (JP0=0 JP1=0 JP2 = 0)
#define DEV_ADDR 0
//Hardware address of KAmodEXP2 (JP0=1 JP1=0 JP2 = 0)
#define DEV_ADDR 1
//Hardware address of KAmodEXP2 (JP0=0 JP1=1 JP2 = 0)
#define DEV_ADDR 2
//Hardware address of KAmodEXP2 (JP0=1 JP1=1 JP2 = 0)
#define DEV_ADDR 3
//Hardware address of KAmodEXP2 (JP0=0 JP1=0 JP2 = 1)
#define DEV_ADDR 4
//Hardware address of KAmodEXP2 (JP0=1 JP1=0 JP2 = 1)
#define DEV_ADDR 5
//Hardware address of KAmodEXP2 (JP0=0 JP1=1 JP2 = 1)
#define DEV_ADDR 6
//Hardware address of KAmodEXP2 (JP0=1 JP1=1 JP2 = 1)
#define DEV_ADDR 7
```

22. Wykonaj samo budowanie projektu (bez ponownego startowania sesji debugowej). Z menu *Project* wybierz pozycję *Clean*. W oknie *Clean* kliknij *OK*.

23. Zauważ brak pytania czy załadować plik wynikowy kodu. Spowodowane jest to zmianą ustawienia ścieżki dostępu do pliku wynikowego. Nazwa foldera roboczego jest inna.

24. Przełącz się do perspektywy *CCS Debug*.

załaduj kod wynikowy do układu procesorowego F28027 Piccolo.

25. Kliknij na przycisk *Load*. W oknie *Load Program* kliknij na przycisk *Browse project*.

26. Kliknij na projekt *F2802xI2C_KAmodEXP2* oraz przycisk *OK*.

27. W oknie *Load Program* kliknij na przycisk *Browse* i w folderze RAM wybierz plik *Example_F2802xI2C_eeprom.out* i kliknij przycisk *Otwórz* a potem *OK*.

Zauważ, że nazwa pliku z kodem źródłowym oraz z wynikowym nie została zmieniona.

Inicjalizowanie modułu peryferyjnego I2C

Wstępna inicjalizacja zawartości rejestrów modułu peryferyjnego I2C układu procesorowego jest wykonywana w funkcji *I2CA_Init*. Kod funkcji znajduje się w pliku głównym *Example_2803xI2C_eeprom.c* projektu. Moduł peryferyjny I2C w trakcie obsługi wysyłania i odbierania danych jest kilkakrotnie reprogramowany.

Adres układu pamięci na szynie I²C jest ustawiony w rejestrze I2CSAR na 0x20. Preskaler sygnału zegarowego w rejestrze I2CPSC jest ustawiony tak, aby częstotliwość zegara szyny I²C wynosiła około 400 kHz. Czas poziomu niskiego sygnału zegarowego w rejestrze I2CCLKL jest ustawiony na ok. 1,5 <m>s. Czas poziomu wysokiego sygnału zegarowego w rejestrze I2CCLKL jest ustawiony na ok. 1 <m>s.

W rejestrze I2CIER włączone jest zezwolenie na zgłaszanie przerw SCD i ARDY, bez obsługi XRDY i RRDY. Moduł peryferyjny jest skonfigurowany w rejestrze I2CMMDR jako „slave”, odbiornik oraz 8 bitów na słowo danych. Dalej moduł peryferyjny I2C zostaje wyprowadzony ze stanu Reset, czyli zostaje włączony. Włączane jest rozszerzenie FIFO. Włączona jest obsługa FIFO nadajnika w rejestrze I2CFFTX, z sygnalizacją zapelnienia 0 pozycji oraz bez zgłaszania przerw. Włączona jest obsługa FIFO odbiornika w rejestrze I2CFFRX, z sygnali-

Listing 2. Skonfigurowanie dodatkowych wyprowadzeń GPIO układu procesorowego F28027 Piccolo

```
//KAmoDEXP2 modifications
//KAmoDEXP2 - Additional GPIO
EALLOW;
GpioCtrlRegs.GPAPUD.bit.GPIO18 = 0; // Enable pull-up for GPIO18
GpioCtrlRegs.GPAMUX2.bit.GPIO18 = 0; // as GPIO18
GpioCtrlRegs.GPADIR.bit.GPIO18 = 0; // GPIO18 = input
//KAmoDEXP2 - Enable an ADCINA4 pin as AIO2 digital output, set it high
GpioDataRegs.AIOSET.bit.AIO2 = 1; // Set high output latch
GpioCtrlRegs.AIOMUX1.bit.AIO2 = 0; // ADCINA4 = AIO2
GpioCtrlRegs.AIODIR.bit.AIO2 = 1; // AIO2 = output
EDIS;
```

zacją zapełnienia 0 pozycji oraz bez zgłaszania przerwania. Zerowany jest znacznik przerwania RXFFINT.

W funkcji *main()* wstawiany jest adres procedury *i2c_int1a_isr()* obsługi przerwania I2CINT1A modułu peryferyjnego I2C-A. Jest jeszcze drugie przerwianie I2CINT2A modułu peryferyjny I2C-A, które w programie nie jest używane.

```
EALLOW;
PieVectTable.I2CINT1A = &i2c_int1a_isr;
EDIS;
```

Dalej w funkcji *main()* ustawiane jest zezwolenie na obsługę przerwania INT8.1 grupy 8 w module PIE. Następnie ustawiane jest w CPU zezwolenie na obsługę przerwania INT8. Na koniec włączane jest w CPU globalne zezwolenie na obsługę przerwania maskowalnych.

```
//Enable interrupts required for this example
//Enable I2C interrupt 1 in the PIE: Group 8 interrupt 1
PieCtrlRegs.PIEIER8.bit.INTx1 = 1;
//Enable CPU INT8 which is connected to PIE group 8
IER |= M_INT8;
EINT;
```

Obecnie w programie nie jest używane przerwianie I2CINT1A modułu peryferyjnego I2C-A. Jednak może ono zostać w prosty sposób dostosowane do obsługi transmisji.

28. Zapoznaj się z kodem funkcji w pliku *Piccolo_F2802x_KAmoDEXP2.c* oraz definicjami w pliku *Piccolo_F2802x_KAmoDEXP2.h*.

Wpis danych do modułu KAmoDEXP2

Funkcja *Uint16 KAmoDEXP2_WriteReg(Uint16 idx, Uint16 RegAddr, Uint16 RegValue)* realizuje wpis danych do wskazanego rejestru układu scalonego MCP23008 firmy Microchip zastosowanego w module KAmoDEXP2 firmy Kamami.

Funkcja *KAmoDEXP2_WriteReg* realizuje najpierw skonfigurowanie modułu peryferyjnego I2C a następnie wstawienie danych do zapisu do układu MCP23008. Parametr *idx* podaje adres układu scalonego MCP23008 na szynie I²C. Adres ten jest definiowany poziomem logicznym wejść A0, A1 i A2 układu scalonego. Tak zdefiniowany zmienny numer jest przesuwany w lewo o jeden bit i konkatelowany ze stałą częścią adresu na szynie wynoszącą 0x20. Parametr *RefAddr* podaje adres rejestru wewnętrznego układu scalonego MCP23008. Jest on numerem od 0x00 do 0x0A. Parametr *RegValue* przekazuje wartość do wpisania do rejestru.

W funkcji *KAmoDEXP2_WriteReg* najpierw jest sprawdzane czy został zakończony poprzedni transfer na szynie (czy był wygenerowany stan STOP). Adres

układu scalonego MCP23008 (0x20) wpisywany jest do rejestru I2CSAR. Sprawdzane jest czy szyna nie jest zajęta (bit BB [I2CSTR[12]]). Jeśli szyna jest wolna to do rejestru I2CCNT wpisywana jest liczba bajtów do wysłania. Jest ona równa liczbie bajtów danych plus bajt adresu rejestru wewnętrznego układu scalonego MCP23008.

Do FIFO nadajnika (rejestr I2CDXR) jest wpisywany bajt adresu rejestru. Następnie wpisywane jest bajt danych. Na koniec ustawiana jest nowa zawartość rejestru I2CMDR. Ustawienie poszczególnych bitów tego rejestru powoduje skonfigurowanie modułu peryferyjnego I2C:

- FREE=1 – Moduł pracuje dalej po przejściu układu procesorowego w tryb debugowy czasu rzeczywistego, np. po aktywowaniu pułapki.
- STT=1 – Jeśli moduł pracuje jako MASTER to powoduje wygenerowanie stanu START na szynę. Następnie stan bitu jest automatycznie zerowany.
- STP=1 – Powoduje wygenerowanie stanu STOP na szynę po zliczeniu licznika bajtów na zero. Następnie stan bitu jest automatycznie zerowany.
- MST=1 – Praca modułu jako MASTER.
- TRX=1 – Praca modułu jako nadajnik.
- IRS=1 – Zezwolenie na pracę.

Po wpisie ustawień do rejestru I2CMDR wysyłanie danych na szynę I²C jest wykonywane automatycznie, sprzętowo przez moduł peryferyjny I2C układu procesorowego F2802x Piccolo.

W programie zostało zastosowane opóźnienie o 90 <m>s, czyli trochę więcej, niż typowy czas wysłania trzech bajtów na szynie I²C (**rysunek 1**) z zegarem o częstotliwości 400 MHz [10].

Na zapisie transmisji (rys. 1) dokładnie widać momenty wystawiania stanu ACK na linii SDA przez układ scalony MCP23008. Podczas wymuszania niskiego poziomu napięcia na linii SDA wynosi ok. +0.5V w porównaniu do poziomu prawie zerowego podczas sterowania stanem linii przez moduł peryferyjny I2C układu procesorowego F28027 Piccolo.

Po zakończeniu przez moduł peryferyjny I2C transmisji i wystawieniu stanu STOP na szynę moduł peryferyjny I2C wykrywa ten stan i zostaje zgłoszone przerwianie I2CINT1A z ustawionym kodem SCD w rejestrze I2CISRC. Może być ono obsługiwane w procedurze *i2c_int1a_isr*. Jednocześnie kasowany jest bit znacznika STP w rejestrze I2CMDR.

Odczyt danych z modułu KAmoDEXP2

Funkcja *int16 KAmoDEXP2_ReadReg(Uint16 idx, Uint16 RegAddr)*; realizuje wpis odczyt ze wskazanego rejestru układu scalonego. Funkcja *KAmoDEXP2_ReadReg* realizuje pracę w dwóch fazach:

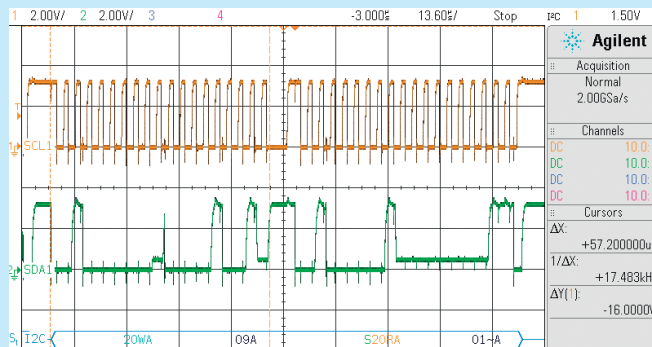
- Skonfigurowanie modułu peryferyjnego I2C jako nadajnik i wysłanie bajtu adresu rejestru do układu MCP23008.
- Skonfigurowanie modułu peryferyjnego I2C jako odbiornik i odebranie bajtu danych.

Parametr *idx* podaje adres układu scalonego MCP23008 na szynie I²C. Adres ten jest definiowany poziomem logicznym wejść A0, A1 i A2 układu scalonego. Tak zdefiniowany zmienny numer jest przesuwany w lewo o jeden bit i konkatelowany ze stałą częścią adresu na szynie wynoszącą 0x20. Parametr *RegAddr* podaje adres rejestru wewnętrznego układu scalonego MCP23008. Jest on numerem od 0x00 do 0x0A.

Faza 1

W funkcji *KAmodeXP2_ReadReg* najpierw jest sprawdzane czy został zakończony poprzedni transfer na szynie (czy był wygenerowany stan STOP). Adres układu scalonego MCP23008 (0x20) wpisywany jest do rejestru I2CSAR. Sprawdzane jest czy szyna nie jest zajęta (bit BB (I2CSTR[12])). Jeśli szyna jest wolna to do rejestru I2CCNT wpisywana jest liczba bajtów do wysłania. Jest ona równa liczbie bajtów danych plus bajt adresu rejestru wewnętrznego układu scalonego MCP23008.

Do FIFO nadajnika (rejestr I2CDXR) jest wpisywany bajt adresu rejestru. Następnie wpisywane jest bajt danych. Na koniec ustawiana jest nowa zawartość rejestru I2CMR. Na koniec ustawiana jest nowa zawartość reje-



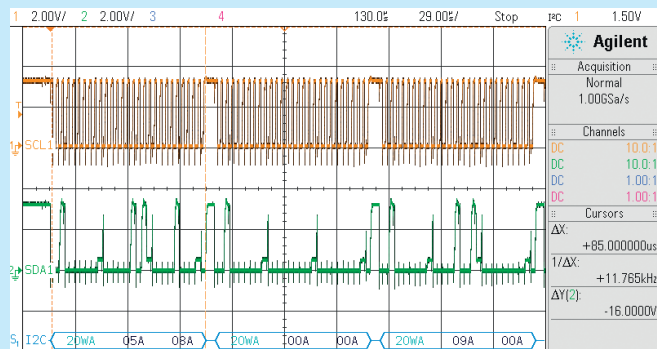
Rysunek 2. Odczytywanie przez moduł peryferyjny I2C układu procesorowego F28027 Piccolo zawartości (0x01) z do rejestru portu GPIO (adres 0x09) układu scalonego MCP23008 pod adresem 0x20 na szynie I²C.

stru I2CMR. Ustawienie poszczególnych bitów tego rejestru powoduje skonfigurowanie modułu peryferyjnego I2C:

- STT=1 – Jeśli moduł pracuje jako MASTER to powoduje wygenerowanie stanu START na szynę. Następnie stan bitu jest automatycznie zerowany.
- MST=1 – Praca modułu jako MASTER.
- TRX=1 – Praca modułu jako nadajnik.
- IRS=1 – Zezwolenie na pracę.

Listing 3. Kod głównej części funkcji *main()*.

```
//KAmodeXP2 modifications
KAmodeXP2_Reset();
DELAY_US(1); // 1 us
//KAmodeXP2 Init(); // Init the processor I2C module hardware
Error = KAmodeXP2_WriteReg(DEV_ADDR, IOCON, 0x08); //(5) Enable the MCP23008 hardware address
if (Error != I2C_SUCCESS) fail();
Error = KAmodeXP2_WriteReg(DEV_ADDR, IODIR, 0x00); //GPIO as output
if (Error != I2C_SUCCESS) fail();
Error = KAmodeXP2_WriteReg(DEV_ADDR, GPIO, 0x00); //Write GIO = 0x00
if (Error != I2C_SUCCESS) fail();
//infinite loop
for(;;)
{
    Error = KAmodeXP2_WriteReg(DEV_ADDR, GPIO, 0x00); //Write GPIO = 0x00
    DELAY_US(1000000); // 1000ms
//-----
// effect 1
//-----
for(Output = 1; Output != 0x20; Output <<= 1) //0x100;
{
    // (9) Write GPIO = Output
    Error = KAmodeXP2_WriteReg(DEV_ADDR, GPIO, Output);
    if (Error != I2C_SUCCESS) fail();
    //SetOutScreen(Output);
    tmp= KAmodeXP2_ReadReg(DEV_ADDR, GPIO);
    if (tmp < I2C_SUCCESS) fail();
    Input= (Uin16)tmp;
    DELAY_US(700000); // 700 ms
}
//-----
// effect 2
//-----
for(Output = 1; Output < 0xF; Output <<= 1)
{
    Output |= 1;
    KAmodeXP2_WriteReg(DEV_ADDR, GPIO, Output);
    // SetOutScreen(Output);
    DELAY_US(700000); // 700ms
}
//-----
// effect 3
//-----
for(Counter = 0; Counter < 4; Counter++)
{
    Output = 0x55;
    KAmodeXP2_WriteReg(DEV_ADDR, GPIO, Output);
    //SetOutScreen(Output);
    DELAY_US(700000); // 700ms
    Output = 0xAA;
    KAmodeXP2_WriteReg(DEV_ADDR, GPIO, Output);
    //SetOutScreen(Output);
    DELAY_US(700000); // 700ms
}
} //for(;;)
```



Rysunek 3. Transmisja konfigurowania rejestru konfiguracyjnego IOCON (adres 0x05), rejestru kierunku IODIR (adres 0x00) oraz rejestru portu GPIO (adres 0x09) układu scalonego MCP23008 pod adresem 0x20 na szynie I²C.

Oznacza to, że po zakończeniu transferu nie jest wystawiany stan STOP na szynę. Po wpisaniu ustawień do rejestru I2CMDR wysyłanie danych na szynę I²C jest wykonywane automatycznie, sprzętowo przez moduł peryferyjny I²C układu procesorowego F2802x Piccolo. W programie zostało zastosowane opóźnienie o 60 μ s, czyli trochę więcej niż typowy czas wysłania dwóch bajtów na szynie I²C z zegarem o częstotliwości 400 MHz (**rysunek 2**).

Stan ACK po wysłaniu ostatniego bajtu zostanie wykryty przez moduł peryferyjny I²C. Zgłaszane jest wtedy przerwanie I2CINT1A z ustawionym bitem znacznika ARDY w rejestrze I2CISRC. Może być ono obsługiwane w procedurze `i2c_int1a_isr`.

Faza 2

Adres układu scalonego MCP23008 (0x20) wpisywany jest do rejestru I2CSAR. Sprawdzane jest czy szyna nie jest zajęta (bit BB [I2CSTR[12]]). Do rejestru I2CCNT wpisywana jest liczba bajtów do odczytu. Ustawiana jest nowa zawartość rejestru I2CMDR. Ustawienie poszcze-

gólnych bitów powoduje skonfigurowanie modułu peryferyjnego I²C:

- NACKMOD=0 – Jeśli moduł pracuje jako odbiornik MASTER to po zliczeniu wewnętrznego licznika odebranych bajtów do zera wysyłany jest stan NACK.
- STT=1 – Jeśli moduł pracuje jako MASTER to powoduje wygenerowanie stanu START na szynę. Następnie stan bitu jest automatycznie zerowany.
- STP=1 – Powoduje wygenerowanie stanu STOP na szynę po zliczeniu licznika bajtów na zero.
- MST=1 – Praca modułu I²C jako MASTER.
- TRX=0 – Praca modułu jako odbiornik.
- IRS=1 – Zezwolenie na pracę.

Moduł peryferyjny I²C wymusza na szynie stan ponowny START (czyli RESTART) a następnie wysyła na szynę bajt adresu pamięci EEPROM z bitem R/W ustawionym na poziom wysoki (READ). Po odebraniu bitu potwierdzenia ACK z szyny I²C (wystawionego przez układ scalony MCP23008) moduł peryferyjny I²C zwalnia szynę. Zaczyna on wtedy pracować jako odbiornik bajtów wysyłanych układ scalony MCP23008.

Odbiór danych z szyny I²C jest wykonywany automatycznie, sprzętowo. Odbiór kolejnych bajtów przez moduł peryferyjny I²C jest potwierdzany wystawieniem przez moduł peryferyjny I²C bitu potwierdzenia ACK na szynę I²C. Jedynie po odebraniu ostatniego bajtu (wewnętrzny licznik bajtów zliczył do zera) wystawiany jest na szynę bit NACK oraz następnie stan STOP. W programie zostało zastosowane opóźnienie o 65 μ s, czyli trochę więcej niż typowy czas wysłania dwóch bajtów na szynie I²C (rys. 2) z zegarem o częstotliwości 400 MHz [10]. Na zapisie transmisji (rys. 2) dokładnie widać momenty wystawiania stanu ACK na linii SDA przez układ scalony MCP23008. Oraz wystawianie niskiego poziomu podczas transmisji bajtu danych.

Listing 4. Funkcja inicjowania modułu peryferyjnego I²C układu procesorowego F2802x Piccolo.

```
void I2CA_Init(void)
{
    // Initialize I2C
    // Slave address - MCP23008 control code
    I2caRegs.I2CSAR = 0x0020 | (DEV_ADDR << 1);
    // I2CCLK = SYSCLK/(I2CPSC+1)
    #if (CPU_FRQ_40MHZ|CPU_FRQ_50MHZ)
        I2caRegs.I2CPSC.all = 4; // Prescaler - need 7-12 Mhz on module clk
    #endif
    #if (CPU_FRQ_60MHZ)
        I2caRegs.I2CPSC.all = 6; // Prescaler - need 7-12 Mhz on module clk
    #endif
    I2caRegs.I2CCLKL = 10; // NOTE: must be non zero
    I2caRegs.I2CCLKH = 5; // NOTE: must be non zero
    I2caRegs.I2CIER.all = 0x24; // Enable SCD & ARDY interrupts
    I2caRegs.I2CMDR.all = 0x0020; // Take I2C out of reset
    // Stop I2C when suspended

    // 15 NACKMOD 0 sends a NACK bit to the transmitter
    // during the next acknowledge cycle on the bus
    // 14 FREE 0 continues to operate when a breakpoint occurs
    // 13 STT 1 STT is automatically cleared after the START condition has been generated
    // 12 rsv
    // 11 STP 0 STP is automatically cleared after the STOP condition has been generated
    // 10 MST 0 Master mode
    // 9 TRX 0 Transmitter mode
    // 8 XA 0 7-bit addressing mode
    // 7 RM 0 Nonrepeat mode
    // 6 DLB 0 Digital loopback mode is disabled
    // 5 IRS 1 The I2C module is enabled
    // 4 STB 0 not in the START byte mode
    // 3 FDF 0 Free data format mode is disabled
    // 2:0 BC 000 8 bits per data byte
    I2caRegs.I2CFITX.all = 0x6000; // Enable FIFO mode and TXFIFO
    I2caRegs.I2CFRX.all = 0x2040; // Enable RXFIFO, clear RXFFINT,
    return;
} // I2CA_Init(void)
```

Wystawienie stanu STOP na szynę powoduje zgłoszenie przerwania I2CINT1A z ustawionym bitem znacznika SCD w rejestrze I2CISRC. Może być ono obsługiwane w procedurze *i2c_int1a_isr*. Jednocześnie kasowany jest bit znacznika STP w rejestrze I2CMDR. Odebrane dane są odczytywane z FIFO odbiornika (rejestr I2CDRR).

Uruchamianie projektu F2802xI2C_KAmodEXP2

29. Przełącz się do perspektywy *CCS Debug*. Kliknij na przycisk Resume.

Program zostanie uruchomiony. Obserwuj diody D2, D3, D4, D5 na zestawie ewaluacyjnym C2000 Piccolo LaunchPad. Diody prezentują trzy wzory świecenia.

Najprostszym sposobem inicjalizacji scalonego ekspandera MCP23008 dla szyny I²C jest pokazany na **rysunku 3**. Najpierw do rejestru konfiguracyjnego IOCON (adres 0x05) układu wpisywany jest bajt danych (0x08). Są to ustawienia zgodne z domyślnymi po operacji *Reset*. Następnie do rejestru kierunku IODIR (adres 0x00) układu jest wpisywany bajt danych (0x00). Oznacza to ustawienie wszystkich bitów portu układu MCP23008 jako wyjścia. Na koniec do rejestru portu GPIO (adres 0x09) jest wpisywana nowa zawartość 0x00.

Bibliografia

- [1] Henryk A. Kowalski, „Zestaw ewaluacyjny C2000 Piccolo LaunchPad”, *Elektronika Praktyczna* 01/2013
- [2] Henryk A. Kowalski, „C2000 Piccolo LanuchPad – pierwszy program w środowisku programowym Code Composer Studio v5”, *Elektronika Praktyczna* 02/2013.
- [3] Code Composer Studio, strona produktu <http://www.ti.com/ccs>
- [4] controlSUITE, strona pobierania <http://www.ti.com/tool/controlsuite>
- [5] Henryk A. Kowalski, „Procesory DSP dla praktyków”, Wydawnictwo BTC, Warszawa, 2011 <http://ii.pw.edu.pl/kowalski/dsp/book/>
- [6] Henryk A. Kowalski, „Procesory DSP w przykładach”, Wydawnictwo BTC, Warszawa, 2012 <http://ii.pw.edu.pl/kowalski/dsp/book/>
- [7] TMS320F28027, TMS320F28026, TMS320F28023, TMS320F28022, TMS320-F28021, TMS320F280200, *Piccolo Microcontrollers, Data Sheet, SPRS523I*, 31 Jul 2012
- [8] F2802x Firmware Development Package USER'S GUIDE v. 210 [f2802x-FRM-EX-UG.pdf], pakiet controlSUITE
- [9] F2802x Peripheral Driver Library USER'S GUIDE v. 210 [f2802x-DRL-UG.pdf], pakiet controlSUITE
- [10] TMS320x2802x/TMS320F2802xx Piccolo System Control and Interrupts Reference Guide (Rev. D) 13 Feb 2013, [SPRUFN3D]
- [11] controlSUITE Getting Started Guide (Rev. B), SPRUGU2B, 09 June 2011
- [12] TMS320x2802x, 2803x Piccolo Inter-Integrated Circuit (I2C) Module Ref. Guide (Rev. D) [SPRUFZ9D.pdf]
- [13] KAmodEXP2, Adresowalny ekspander GPIO z interfejsem I2C www.kamami.pl
- [14] CAB_A, Przewody połączeniowe, www.kamami.pl
- [15] MCP23008/MCP23S08, 8-Bit I/O Expander with Serial Interface, Data Sheet (08/24/2007) www.microchip.com

m.ElektronikaB2B.pl

teraz zawsze pod ręką w Twoim smartfonie



Wejść

Bądź dobrze poinformowany