

Mikroelektronika w Technice i Medycynie
Podstawy projektowania systemów wbudowanych
Instrukcja do ćwiczeń laboratoryjnych
Akwizycja sygnałów - pomiar “On Demand”
Mirosław Żołądź 2017

Informacje wstępne	2
1. Zestaw ćwiczeniowy	2
2. Wgrywanie programu	4
3. Tworzenie projektu	5
4. Dodawanie modułu z biblioteki ASF	5
GPIO	6
1. Wyjście	6
2. Wejście	6
3 Opóźnienie	6
UART (stdio)	7
1. Echo znakowe	7
2. Echo łańcuchowe	7
3. Echo łańcuchowe, c.d.	7
4. Przesyłanie liczb	7
ADC	8
1. Pomiar “single ended” (czujnik temperatury)	8
2. Pomiar różnicowy	9
2.1 Dwa czujniki temperatury (wykorzystanie napięcia odniesienia)	9
2.2 Wzorcowe źródło napięcia (przeliczanie na napięcie)	9
2.3 Belka tensometryczna (wpływ wzmocnienia i nadpróbkowania na SNR)	10

Informacje wstępne

1. Zestaw ćwiczeniowy

Zestaw ćwiczeniowy składa się połączonych ze sobą modułu pomiarowego oraz modułu czujników.

Moduł pomiarowy zawiera min.:

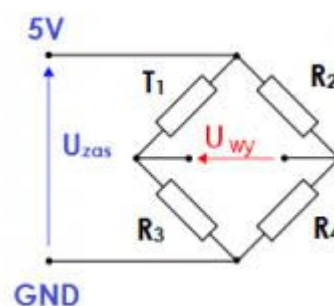
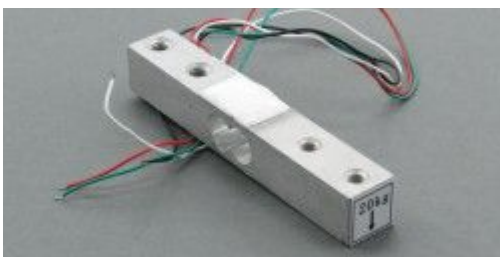
- mikrokontroler atxmega256A3BU
- przyciski do:
 - resetu mikrokontrolera
 - uruchamiania bootloadera (Port E, pin 5)
- diody LED do:
 - sygnalizacji obecności napięcia zasilającego
 - sygnalizacji transmisji danych za pośrednictwem UART-a (oddzielnie Tx/Rx)
 - ogólnego stosowania (Port D, pin 5)
- złącza usb do:
 - wgrywania programu ("USB_PROG")
 - komunikacji za pośrednictwem UART-a (usb, "USB_COM")

Moduł może być zasilany z jednego z wymienionych gniazd USB.

Moduł czujników zawiera:

- dwa czujniki temperatury (symbol na obudowie czujnika)
- ultradźwiękowy miernik odległości (symbol na płytce miernika)
- mikrofon mems (symbol na płytce mikrofonu)
- belkę tensometryczną
- potencjometr (zasilany napięciem 0-3.3V)
- źródło napięć stałych z dwoma wyjściami, między którymi różnica potencjałów wynosi ok. 140 mV

Belka tensometryczna



Sposób podłączenia czujników oraz innych źródeł napięcia do mikrokontrolera

PA.1 (we. przetwornika A/C, port A, pin 1):

- czujniki temperatury 1
- źródło napięć stałych - wyjście 1
- belka tensometryczna - wyjście 1
- potencjometr
- mikrofon
- ultradźwiękowy miernik odległości - wyjście odbiornika ultradźwięków

PA.5 (we. przetwornika A/C):

- czujniki temperatury 2
- źródło napięć stałych - wyjście 2
- belka tensometryczna - wyjście 2

Wybór źródła sygnału odbywa się za pomocą zwerek na module czujników.

Zewnętrzne **napięcie referencyjne** przetwornika A/C (pin PA.0) wynosi 3.3V

Do **monitorowania sygnałów** wchodzących na piny PA.1 i PA.5 mikrokontrolera służą wyprowadzeni znajdujące się w pobliżu potencjometru.

Testy zestawu

W celu sprawdzenia poprawnego działania zestawu uruchomieniowego należy skorzystać z programów testowych (załącznik ProgramyTestowe.zip).

Poprawne działanie mikrokontrolera należy testować programem blinky.hex (pulsowanie LED-em na płytce mikrokontrolera).

Poprawne działanie źródeł rejestrowanych sygnałów należy testować za pomocą odpowiednich zestawów programów, składających się z programu na mikrokontroler oraz aplikacji użytkownika (plik .exe, wymaga LabVIEW 2016, środowisko lub runtime). W podkatalogach Dostępne są zestawy do testów:

- czujników temperatury, źródła napięcia referencyjnego oraz belki tensometrycznej
- potencjometru i mikrofonu
- ultradźwiękowego miernika odległości.

Uwaga: przeprowadzenie powyższych testów wymaga połączenia złącza usb COM z PC, oraz ustawienia odpowiedniego numeru portu COM w aplikacji LabVIEW.

2. Wgrywanie programu

Zastosowany w zestawie ćwiczeniowym mikrokontroler posiada dwa dedykowane interfejsy do programowania/debugowania. Wymagają one jednak odpowiedniego programatora. Zastosowany mikrokontroler pozwala również na wgrywanie programu za pośrednictwem interfejsów komunikacyjnych (uart, i2c, spi, usb). Pozwala to w przypadku interfejsów uart i usb uniknąć konieczności stosowania dodatkowego urządzenia do programowania. Programowanie z użyciem wspomnianych interfejsów wymaga wcześniejszego załadowania (jednorazowo) do mikrokontrolera programu tzw. bootloadera. Programu bootloadera uruchamiany jest jeżeli w trakcie resetu na odpowiednim pinie mikrokontrolera wymuszono stan "0".

Mikrokontrolery w zestawach posiadają wgrany program bootloadera.

Do wgrywania programu z do mikrokontrolera wyposażonego w bootloader służy aplikacja FLIP. Komputery w laboratorium posiadają zainstalowaną aplikację FLIP (szczegóły: nota aplikacyjna AVR1916; uwaga: Jeżeli występuje problem z aktualizacją drivera opisaną w rozdz. 6.2.1 należy usunąć aktualny driver (menu\uninstall) a następnie powtórzyć instalację).

Programowanie mikrokontrolera z użyciem FLIP-a wymaga połączenia zestawu ćwiczeniowego z komputerem PC za pośrednictwem gniazda "USB_PROG".

Przed każdym zaprogramowaniem mikrokontrolera przy użyciu FLIP-a należy połączyć się z mikrokontrolerem.

Przed każdym połączeniem z programu FLIP z mikrokontrolerem należy uruchomić moduł w trybie bootloadera, tj.:

- nacisnąć przycisk "programowania" mikrokontrolera ("BOOT")
- nacisnąć a następnie zwolnić przycisk resetu mikrokontrolera ("RST")
- zwolnić przycisk "programowania" mikrokontrolera ("BOOT")

Do testu wgrywania można użyć programu blinki.hex (strona przedmiotu\załączniki).

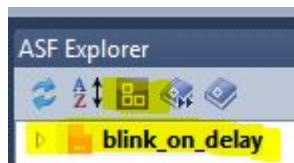
Po wgraniu programu blinki.hex powinien zacząć migać niebieski LED na module (pod płytką z sensorami).

3. Tworzenie projektu

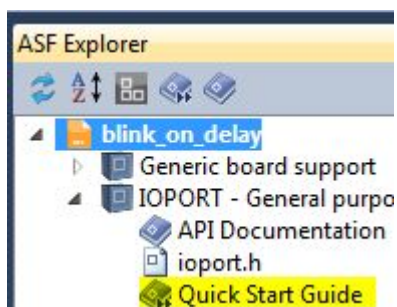
1. Uruchomić środowisko Atmel Studio (6.2)
2. Wybrać tworzenie nowego projektu
3. Wybrać opcję "C/C++" i "ASF board project"
4. Ustawić:
 - a. nazwę projektu (Name) na "GPIO"
 - b. katalog projektu (Location) na własny katalog
 - c. nazwę rozwiązania (Solution name) na "acquisition"
5. Nacisnąć OK
6. W okienku wyszukiwania wpisać: "256a3bu"
7. Z dolnej listy wybrać: "User board template ..."
8. Nacisnąć OK
9. Uruchomić kompilację projekt (F7) a następnie sprawdzić jej wynik (powinno być "succesfull")

4. Dodawanie modułu z biblioteki ASF

1. W okienku górnym okienku po prawej stronie przeciągnąć zakładki "Solution Explore" i "ASF Explorer" odpowiednio na pozycję pierwszą i drugą licząc od lewej.
2. Dodać biblioteki do obsługi GPIO
 - a. W tym celu Wybrać projekt a następnie uruchomić ASF Wizzard



- b. W oknie wizarda wybrać: Show: "Services" a następnie w okienku wyszukiwania wpisać "IOPORT"
3. Z listy dostępnych modułów wybrać jedyną dostępną pozycję, dodać ją do wybranych modułów (przycisk "Add") a następnie dodać wybrany moduł do projektu (przycisk "Apply")
 4. Otworzyć przykład użycia modułu



GPIO

1. Wyjście

1. Na podstawie przykładu użycia modułu IOPORT zmodyfikować main.c tak aby program zapalał LED (port D, pin 5) na module Main. W tym celu:
2. usunąć z pliku main.c wszystkie zakomentowane linie oraz wywołanie funkcji "board_init"
3. dodać do pliku main.c nieskończoną pętlę główną
4. dodać kod ustawiający kierunek oraz stan pinu sterującego LED-em ('1')
5. skompilować a następnie wgrać program

2. Wejście

Na podstawie przykładu użycia modułu IOPORT zmodyfikować main.c tak aby LED świecił się tylko wtedy jeżeli naciśnięty jest przycisk (port E, pin 5) po prawej stronie modułu (patrzac od strony LED-ów). Należy włączyć podciąganie pinu przycisku do Vcc ("pull up").

3 Opóźnienie

1. Zmodyfikować program z poprzedniego zadania tak aby pulsował LED-em. W tym celu:
2. Dodać moduł "Delay" ASF (użyć opcji wyszukiwania i keyworda "delay")
3. Użyć funkcji delay_ms z argumentem 100
4. Użyć funkcji negowania stanu pina (...toggle...)
5. Sprawdzić działanie
6. Zmodyfikować program tak aby pulsowanie miało miejsce tylko jeżeli jest wciśnięty przycisk

UART (stdio)

1. Echo znakowe

1. Dodać do rozwiązania "acq_on_demand" projekt o nazwie "UART" (rozwiązanie\menukontekstowe\addnewproject)
2. Dodać do projektu moduły IOPORT i "standard serial I/O" (search\stdio)
3. Podmienić zawartość pliku main.c na poniższą. Przeanalizować kod i sprawdzić jego działanie.

```
#include <asf.h>
#define UART_TXPIN IOPORT_CREATE_PIN(PORTE, 3)

int main (void) {

    static usart_rs232_options_t USART_SERIAL_OPTIONS = {
        .baudrate = 9600,
        .charlength = USART_CHSIZE_8BIT_gc,
        .paritytype = USART_PMODE_DISABLED_gc,
        .stopbits = false
    };

    stdio_serial_init(&USARTE0, &USART_SERIAL_OPTIONS);
    ioport_set_pin_dir(UART_TXPIN, IOPORT_DIR_OUTPUT);

    while(1) {
        char ch;
        scanf("%c", &ch);
        printf("%c", ch);
    }
}
```

2. Echo łańcuchowe

Zmodyfikować program z poprzedniego zadania tak aby odsyłał wpisane łańcuchy
Docelowo odesłany łańcuch powinien pojawiać się w nowej linii

UWAGI:

- zapoznać się z podstawami formatowania biblioteki "stdio"
- ustawić terminator terminala na CR (terraterm\menu\setup\terminal\new-line)

3. Echo łańcuchowe, c.d.

Zmodyfikować program z poprzedniego zadania tak aby po wysłaniu określonego znaku (bez terminatora) odsyłał określony łańcuch:

b - "binary"

v - "voltage"

t - "temperstuire"

inne - ???

4. Przesyłanie liczb

Zmodyfikować program z poprzedniego zadania tak aby wysyłał po kolei liczby typu int począwszy od 0 w formacie dziesiętnym.

ADC

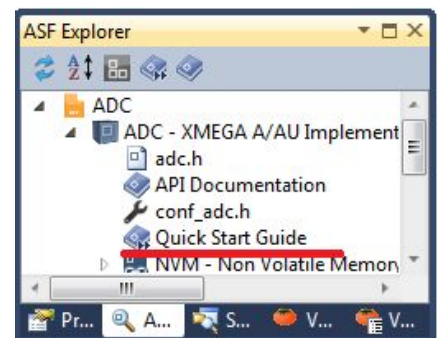
Przydatne materiały:

- “XMEGA AU MANUAL” (8331F–AVR–04/2013)
- “ATxmega256A3BU datasheet” (Atmel-8362 G-AVR-ATxmega-07/2014), szczególnie rozdz. 33. (Pinout and Pin Functions)
- Application note “Using the XMEGA ADC”
<http://www.microchip.com/wwwAppNotes/AppNotes.aspx?appnote=en591303>

1. Pomiar “single ended” (czujnik temperatury)

1. Dodać do rozwiązania “acq_on_demand” projekt o nazwie “ADC” oraz moduły: IOPORT, “standard serial I/O” oraz “ADC”
2. Skopiować zawartość pliku main.c z poprzedniego projektu i sprawdzić czy projekt się kompiluje.
3. Przeanalizować a następnie wstawić do pliku main.c zawartość przykładu użycia modułu ADC (Basic use case) tak aby program wysyłał przez port szeregowy wynik konwersji po odebraniu dowolnego znaku.

Sprawdzić działanie programu (powinien coś odsyłać).



4. Ustawić jumpery na pomiar napięcia z czujników temperatury.

Zmodyfikować program tak aby konwertował napięcie z czujnika temperatury (patrz Atmel-8362, rozdz 29 i 33).

Zmierzyć multimetrem wartość napięcia z czujnika (powinna wynosić ok 250 mV) a następnie sprawdzić czy wartość zmienia się po podgrzaniu czujnika (przyłożeniu palca). UWAGA: do pomiaru używać przelotek a nie bolców.

Powtórzyć test przy użyciu programu i terminala.

5. Powtórzyć test przy użyciu jednej z aplikacji testowych

Opcja “Clear” z menu kontekstowego kontrolki z wykresem (Waveform Chart) pozwala zresetować pamięć kontrolki (również w czasie działania programu).

6. Oszacować na podstawie pomiaru amplitudę szumów (peak to peak) wyrażoną w LSB
Dodać do programu 16-krotne nadpróbkowanie, tj. odsyłać uśredniony wynik 16 konwersji.
Ponownie oszacować amplitudę szumów.

7. Zapisać plik main pod nazwą “main_OnDemand_1_SE” (“Save as”)

2. Pomiar różnicowy

2.1 Dwa czujniki temperatury (wykorzystanie napięcia odniesienia)

1. Przy użyciu poprzedniego programu oszacować na podstawie pomiaru jaka część zakresu pomiarowego przetwornika jest wykorzystywana podczas pomiaru temperatury otoczenia i palca.
UWAGA: dotykać tylko jednego z czujników.
2. Zmodyfikować program tak aby przetwornik pracował w trybie ze znakiem, różnicowym, ze wzmocnieniem 4. Jako źródła napięcia podanego na wejście ujemne wzmacniacza różnicowego użyć napięcia z drugiego przetwornika temperatury (Wejście ADC5 przetwornika A/C).
UWAGA: do odczytu rezultatu konwersji użyć funkcji odpowiedniej dla konwersji ze znakiem.
3. Ponownie oszacować jaka część zakresu pomiarowego przetwornika jest wykorzystywana podczas pomiaru temperatury otoczenia i palca
UWAGA: dotykać tylko jednego z czujników.
4. Oszacować jaka część zakresu pomiarowego przetwornika jest wykorzystywana podczas dotykania na przemian obu czujników do momentu stabilizacji (w razie konieczności skrócić długość historii kontrolki z wykresem, możliwe tylko przy wyłączonym programie)
5. Dobrać maksymalne możliwe wzmocnienie czyli takie dla którego nie dochodzi do przekroczenia zakresu pomiarowego przetwornika.
6. **Zapisać plik main pod nazwą “main_OnDemand_2_Diff_Lsb.c”**

2.2 Wzorcowe źródło napięcia (przeliczanie na napięcie)

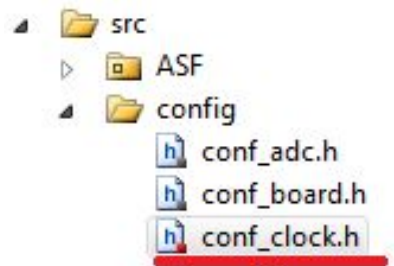
1. Ustawić jumpery na pomiar ze wzorcowego źródła napięcia.
2. Zmierzyć wartość napięcia używając multimetru.
3. Zmodyfikować program tak aby wartość wzmocnienia oraz ilość konwersji na pomiar (uśrednianie) były określane dyrektywą preprocesora (GAIN, SAMPLES_PER_MEASUREMENT). Sprawdzić wynik konwersji dla wzmocnienia równego jeden oraz jednej konwersji na pomiar.
4. Przeliczyć ile mV przypada na jeden LSB i porównać z wartością wynikającą z dokumentacji
5. Zmienić typ zmiennej używanej do uśredniania wyników konwersji z int na float (pamiętać o modyfikacji printf). Przy użycia **terminala** sprawdzić odsyланą wartość.

Zmodyfikować konfigurację kompilatora w sposób opisany w artykule: google: “print-float-atmel-studio-7”

Ponownie sprawdzić odsyланą wartość.
6. Zmodyfikować program tak aby odsyłał wartość w miliwoltach. W tym celu należy min. zdefiniować wartość napięcia referencyjnego REF_VOLTAGE_mV.
7. **Zapisać plik main pod nazwą “main_OnDemand_2_Diff_mV.c”**

2.3 Belka tensometryczna (wpływ wzmocnienia i nadpróbkowania na SNR)

1. Ustawić jumpery na pomiar z belki tensometrycznej.
2. Zmienić częstotliwość taktowania mikrokontrolera na 32 MHz (generator RC).



Ustawić częstotliwość taktowania ADC na 500 000.

Zarejestrować wykres napięcia podczas dwukrotnego obciążenia belki najcięższym odważnikiem.

Pomiaru dokonać przy następujących parametrach akwizycji:

```
#define GAIN 64
#define SAMPLES_PER_MEASUREMENT 2048
#define REF_VOLTAGE_mV 1000
```

Powtórzyć pomiar dla najlżejszego odważnika.

3. Używając najcięższego odważnikiem zmierzyć zanotować amplitudę sygnału (S) i odchylenie standardowe szumu (N) dla różnych wzmocnień i ilości konwersji użytych do wyliczenia średniej (nadpróbkowanie). Jako amplitudę sygnału przyjąć różnicę pomiędzy napięciami średnimi odczytanymi dla belki obciążonej i nieobciążonej. Odchylenie standardowe szumu mierzyć przy belce nieobciążonej. Wykorzystać kontrolki dostępne w oknie aplikacji użytkownika.

- a) Zależność SNR od wzmocnienia
SAMPLES_PER_MEASUREMENT = 1 (SPM)
GAIN = 1, 8, 16, 64

Wyniki zapisać w następującej formie:

```
SPM: 1
GAIN: 1, N: 0.75, S: 1.60
GAIN: 8, ...
GAIN: 16, ...
```

- b) Zależność SNR od ilości konwersji użytych do wyliczenia średniej
GAIN = 64
SAMPLES_PER_MEASUREMENT = 1, 2, 4, 16, 256, 1024, 2048

Wyniki zapisać w następującej formie:

```
SPM: 1, N: 0.75, S: 1.60
SPM: 2, ...
SPM: 16, ...
```

4. Zmodyfikować program tak aby zwracał wynik w miligramach (do kalibracji użyć najcięższego odważnika).
Sprawdzić liniowość przetwarzania, tj. zważyć dwa najlżejsze odważniki osobno oraz razem i porównać wynik. (wyniki zanotować)
5. Zapisać plik main pod nazwą "main_OnDemand_2_Diff_mg"