

Mikroelektronika w Technice i Medycynie  
Podstawy projektowania systemów wbudowanych  
Instrukcja do ćwiczeń laboratoryjnych  
**Akwizycja sygnałów - pomiar “On DMA”**  
Miroslaw Żołędź 2017

Spis treści

|  |          |
|--|----------|
| <b>RAM-&gt;UART</b>                                      | <b>2</b> |
| <b>RAM-&gt;DMA-&gt;RAM-&gt;UART</b>                      | <b>3</b> |
| <b>Microphone-&gt;ADC-&gt;DMA-&gt;RAM-&gt;UART</b>       | <b>4</b> |
| <b>UltrasonicModule-&gt;ADC-&gt;DMA-&gt;RAM-&gt;UART</b> | <b>5</b> |

## RAM->UART

1. Do poprzedniego rozwiązania dodać projekt o nazwie “ADC\_OnDMA”
2. Dodać moduły ASF jak w poprzednim projekcie oraz moduł DMA
3. Podmienić zawartość pliku main na zawartość na poniższą.

```
#include <asf.h>

#define UART_TXPIN IOPORT_CREATE_PIN(PORTE, 3)
#define SAMPLES_NR 10000

uint8_t Samples[SAMPLES_NR];

int main(void){

    // CLK
    sysclk_init();

    // UART
    ioport_set_pin_dir(UART_TXPIN, IOPORT_DIR_OUTPUT);
    static usart_rs232_options_t USART_SERIAL_OPTIONS = {
        .baudrate = 921600,
        .charlength = USART_CHSIZE_8BIT_gc,
        .paritytype = USART_PMODE_DISABLED_gc,
        .stopbits = false
    };
    usart_serial_init(&USARTE0, &USART_SERIAL_OPTIONS);

    // DATA
    for(int16_t i=0;i<SAMPLES_NR;i++) Samples[i]=i;

    while(1) {
        uint8_t c;
        usart_serial_getchar(&USARTE0,&c);

        for(int16_t i=0;i<SAMPLES_NR;i++) Samples[i]=Samples[i]+1;

        usart_serial_write_packet(&USARTE0,Samples, SAMPLES_NR);
    }
}
```

4. Skompilować program i zrozumieć program a następnie przetestować jego działanie z użyciem terminala.

**Zapisać plik main pod nazwą “main\_DMA\_1\_Intro”.**

## RAM->DMA->RAM->UART

Zapoznać się z poniższym kodem a następnie uzupełnić go od inicjalizację DMA.

W aplikacji aplikację nadrzędnej:

- dodać wysyłanie znaku przed odczytem,
- ustawić odczyt na 5000 znaków,
- usunąć kod i kontrolkę do FFT,
- włączyć automatyczne skalowanie osi X i Y pozostałej kontrolki ("Waveform graph").

Sprawdzić działanie programu (zawartość kontrolki powinna się zmieniać). Sprawdzić jak zmieni się działanie programu po usunięciu *dma\_channel\_enable* z pętli głównej.

**Zapisać plik main pod nazwą "main\_DMA\_2\_Mem2Mem".**

```
#include <asf.h>

#define UART_TXPIN IOPORT_CREATE_PIN(PORTE, 3)
#define DMA_CHANNEL 0
#define SAMPLES_NR 5000

uint8_t SamplesSrc[SAMPLES_NR], SamplesDst[SAMPLES_NR];

int main (void) {

    for(int16_t i=0;i<SAMPLES_NR;i++) SamplesSrc[i]=i;

    // CLK
    sysclk_init();

    // UART
    ioport_set_pin_dir(UART_TXPIN, IOPORT_DIR_OUTPUT);
    static usart_rs232_options_t USART_SERIAL_OPTIONS = {
        .baudrate = 921600,
        .charlength = USART_CHSIZE_8BIT_gc,
        .paritytype = USART_PMODE_DISABLED_gc,
        .stopbits = false
    };
    usart_serial_init(&USARTE0, &USART_SERIAL_OPTIONS);

    // DMA
    // ...

    while(1) {
        uint8_t c;

        usart_serial_getchar(&USARTE0,&c);

        for(int16_t i=0;i<SAMPLES_NR;i++) SamplesSrc[i]=255-SamplesSrc[i];

        dma_channel_enable(DMA_CHANNEL);
        dma_channel_trigger_block_transfer(DMA_CHANNEL);
        while(dma_channel_is_busy(DMA_CHANNEL));

        usart_serial_write_packet(&USARTE0,SamplesDst, SAMPLES_NR);
    };
}
```

## Microphone->ADC->DMA->RAM->UART

Zapoznać się z poniższym kodem a następnie uzupełnić go od inicjalizację DMA i ADC przy czym ADC powinno pracować z zegarem 1 MHz i w trybie "freerun" (pozostałe parametry tak jak w poprzednim programie wykorzystującym ADC).

Sprawdzić działanie programu używając modułu z mikrofonem.

**Zapisać plik main pod nazwą "main\_DMA\_3\_Microphone".**

```
#include <asf.h>

#define UART_TXPIN IOPORT_CREATE_PIN(PORTE, 3)
#define DMA_CHANNEL 0
#define SAMPLES_NR 5000

uint8_t Samples[SAMPLES_NR];

int main (void) {

    // CLK
    sysclk_init();

    // UART
    ioport_set_pin_dir(UART_TXPIN, IOPORT_DIR_OUTPUT);
    static usart_rs232_options_t USART_SERIAL_OPTIONS = {
        .baudrate = 921600,
        .charlength = USART_CHSIZE_8BIT_gc,
        .paritytype = USART_PMODE_DISABLED_gc,
        .stopbits = false
    };
    usart_serial_init(&USARTE0, &USART_SERIAL_OPTIONS);

    // ADC

    // DMA

    // Main loop
    while(1) {
        uint8_t c;
        usart_serial_getchar(&USARTE0, &c);

        dma_channel_enable(DMA_CHANNEL);
        while(dma_channel_is_busy(DMA_CHANNEL));

        usart_serial_write_packet(&USARTE0, Samples, SAMPLES_NR);
    };
}
```

## UltrasonicModule->ADC->DMA->RAM->UART

1. Zapoznać się z opisem ultradźwiękowego czujnika odległości HC-SR04 (np. [www.bootland.com.pl](http://www.bootland.com.pl))  
Wejście wyzwalające modułu zostało podłączone do pinu 7 potu A. Do wejścia DAC1 przetwornika podpięto sygnał z głowicy odbiorczej (po wzmacnieniu).
2. **Przy wyłączonym zasilaniu zestawu uruchomieniowego** podłączyć moduł ultradźwiękowego czujnika odległości. Sprawdzić poprawne działanie modułu z użyciem programu referencyjnego *main\_DMA\_4\_Ultrasonic.hex* oraz aplikacji nadrzędnej *4\_DMA\_1Msps.exe*
3. Dodać w odpowiednie miejsce pętli głównej generowanie impulsu wyzwalającego o czasie trwania 1 ms (*delay\_ms*) oraz odwrócenie stanu pinu podłączonego do leda na zestawie uruchomieniowym.
4. Zmienić ilość sampli na 10000 i zmodyfikować odpowiednio aplikację nadrzędną.
5. Sprawdzić działanie programu.  
Na podstawie liczby próbek przypadających na jeden okres obliczyć częstotliwość przebiegu.
6. **Zapisać plik main pod nazwą "main\_DMA\_4\_Ultrasonic".**  
**Zapisać aplikację nadrzędną (nowy vi) pod nazwą "OnDMA"**