

POLITECHNIKA POZNAŃSKA

WYDZIAŁ AUTOMATYKI, ROBOTYKI I ELEKTROTECHNIKI

INSTYTUT ROBOTYKI I INTELIGENCJI MASZYNOWEJ

ZAKŁAD STEROWANIA I ELEKTRONIKI PRZEMYSŁOWEJ



SPRAWOZDANIE

SYSTEMY MIKROPROCESOROWE (LABORATORIUM)
[WARiE_2021-22_AiR_Dz_1_5_D_LUCZAK_21/22]

WYJŚCIA ANALOGOWE (DAC - PRZETWORNIK C/A;
GENERACJA SYGNAŁÓW ANALOGOWYCH O ZADANYCH
PARAMETRACH Z WYKORZYSTANIEM PRZERWAŃ I DMA)
(TEMAT ZAJĘĆ)

KAROL DĘBSKI

(AUTOR I: KAROL.DEBSKI@STUDENT.PUT.POZNAN.PL)

FORMA ZAJĘĆ: LABORATORIUM

PROWADZĄCY:

DR INŻ. DOMINIK ŁUCZAK

DOMINIK.LUCZAK@PUT.POZNAN.PL

POZNAŃ 06-12-2021 9-45

(DATA I GODZINA ZAJĘĆ)

Spis treści

2	Zadanie #2	3
2.1	Specyfikacja	3
2.2	Implementacja	3
2.3	Wyniki testów.....	3
2.4	Wnioski	3
2.5	Pytania	3
3	Zadanie #3	4
3.1	Specyfikacja	4
3.2	Implementacja	4
3.3	Wyniki testów.....	4
3.4	Wnioski	4
4	Zadanie #4	5
4.1	Specyfikacja	5
4.2	Implementacja	5
4.3	Wyniki testów.....	6
4.4	Wnioski	6
5	Zadanie #5	7
5.1	Specyfikacja	7
5.2	Implementacja	7
5.3	Wyniki testów.....	7
5.4	Wnioski	7
7	Podsumowanie.....	8

Zadanie #2

1.1 Specyfikacja

Program generuje sinusoidę o $f=1$ Hz z szybkością $fs=100$ Hz. Do weryfikacji poprawnego działania programu zostanie użyty SWV.

1.2 Implementacja

Listing 1 Callback dla przerwania na TIM7

```
void HAL_TIM_PeriodElapsedCallback (TIM_HandleTypeDef * htim) {  
    if(htim == &htim7) {  
        faza=fmod((n*2*3.14*f/fs), (2*3.14));  
        voltage=1+sin(faza);  
        DAC_SetVoltage(voltage);  
        n++;  
    }  
}
```

Listing 2 Funkcja zadająca napięcie na pinie DAC

```
void DAC_SetVoltage(float voltage) {  
    HAL_DAC_Start(&hdac, DAC1_CHANNEL_1);  
    uint32_t RegDAC=4096*voltage/2.94;  
    HAL_DAC_SetValue(&hdac, DAC1_CHANNEL_1, DAC_ALIGN_12B_R,  
RegDAC);  
}
```

Po wykryciu przerwania obliczany jest argument dla sinusa a następnie wartość tego sinusa jest zadawana na pinie DAC.

1.3 Wynik testów



Rys. 1 Podgląd SWV

1.4 Wnioski

Sinus ma zakładaną częstotliwość 1 Hz. Częstotliwość sygnału z treści polecenia została zmieniona by poprawnie wyświetlać zadany sygnał w SWV.

Zadanie #3

1.1 Specyfikacja

Program na podstawie odczytu z potencjometru zadaje napięcie na pinie DAC. Do weryfikacji poprawnego działania programu zostanie użyty SWV.

1.2 Implementacja

Listing 3 Callback dla przerwania na TIM7

```
void HAL_TIM_PeriodElapsedCallback (TIM_HandleTypeDef * htim){  
    if(htim == &htim7){  
        voltage=2.9*ADC_raw/4096.0;  
        DAC_SetVoltage(voltage);  
    }  
}
```

Listing 4 Główna pętla programu

```
while (1)  
{  
    HAL_ADC_Start_DMA(&hadc1, &ADC_raw, 1);  
    __delay_ms(10);  
    /* USER CODE END WHILE */  
    MX_USB_HOST_Process();  
  
    /* USER CODE BEGIN 3 */  
}
```

W pętli głównej odczytywany jest pomiar ADC z potencjometru a gdy na timerze zostanie wywołane przerwanie zadawane jest napięcie na DAC na podstawie odczytu z ADC.

1.3 Wynik testów



Rys. 2 Podgląd SWV

1.4 Wnioski

DAC generuje zadane napięcie na podstawie odczytu ADC z potencjometru.

Zadanie #4

1.1 Specyfikacja

Na podstawie danych z pliku txt generowany jest sygnał na pinie DAC. Do zweryfikowania poprawnego działania programu zostanie użyte SWV.

1.2 Implementacja

Listing 5 Funkcja zwrotna dla przerwania na timerze

```
void HAL_TIM_PeriodElapsedCallback (TIM_HandleTypeDef * htim) {
    if(htim == &htim7) {
        voltage=data[n];
        DAC_SetVoltage(voltage);
        n++;
        if (sizeof(data)/sizeof(int)==n) n=0;
    }
}
```

Listing 6 Główna pętla programu

```
while (1)
{
    __delay_ms(10);
    /* USER CODE END WHILE */
    MX_USB_HOST_Process();

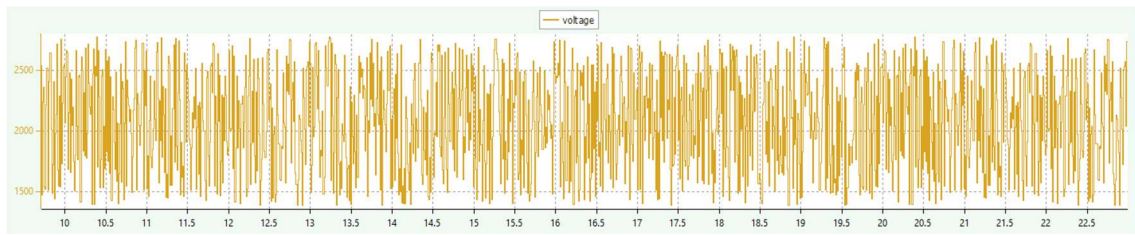
    /* USER CODE BEGIN 3 */
}
```

Listing 7 Skrypt MATLAB'a generujący 3 pliki

```
clear all, close all, clc
fid1=fopen('data.h','w+t')
fid2=fopen('data.c','w+t')
fid3=fopen('data.txt','w+t')
text1='#ifndef INC_TEST_H_\n#define INC_TEST_H_'
%fid1
fprintf(fid1,'#ifndef INC_TEST_H_\n#define INC_TEST_H_\n')
fprintf(fid1,'#define DATA_SIZE 1000\n')
fprintf(fid1,'int data[DATA_SIZE];\n')
fprintf(fid1,'#endif /* INC_TEST_H_ */')
fclose(fid1)
%fid3
values=int32(4096*((rand(1,1000)*1+1)')/2.95)
plot(values)
for i=1:size(values)
    fprintf(fid3,'%d,\n',values(i))
end
fclose(fid3)
%fid2
fprintf(fid2,'#include "data.h"\nint data[DATA_SIZE]={\n#include
"data.txt"\n};\n')
fclose(fid2)
```

Po każdym minięciu okresu timera pobierana jest jedna próbka z danych i następnie za jej pomocą jest generowane napięcie na pinie DAC.

1.3 Wynik testów



Rys. 3 Podgląd SWV

1.4 Wnioski

Wygenerowane próbki za pomocą MATLAB'a są wiernie odtwarzane na pinie DAC.

Zadanie #5

1.1 Specyfikacja

Program za pomocą DMA generuje napięcie na pinie DAC w oparciu o przekazane dane z pliku txt. Do weryfikacji poprawnego działania programu zostanie użyte SWV.

1.2 Implementacja

Listing 8 Inicjalizacja timera i DAC

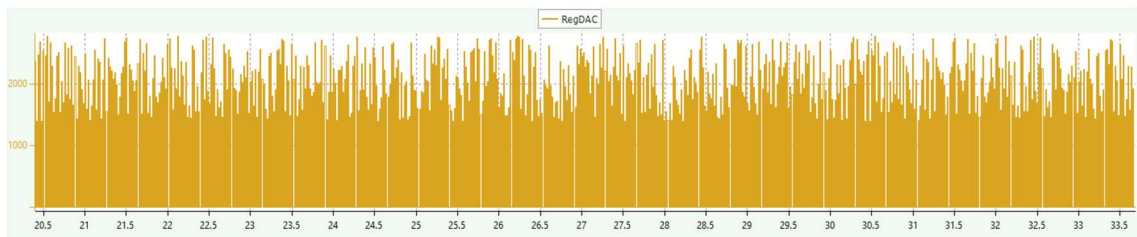
```
HAL_TIM_Base_Start_IT(&htim7);  
HAL_DAC_Start_DMA(&hdac, DAC1_CHANNEL_1, data,  
sizeof(data)/sizeof(int), DAC_ALIGN_12B_R);
```

Listing 9 Callback dla przerwania na TIM7

```
void HAL_TIM_PeriodElapsedCallback (TIM_HandleTypeDef * htim) {  
    if(htim == &htim7) {  
        RegDAC=HAL_DAC_GetValue(&hdac, DAC1_CHANNEL_1);  
    }  
}
```

Za pomocą DMA przesyłane są próbki a w callbacku dla timera, odczytywana jest bieżąca wartość rejestru danych DAC – czyli wartość próbki.

1.3 Wynik testów



Rys. 4 Podgląd SWV

1.4 Wnioski

Wartości rejestru DAC dla danych są takie same jak z pliku data.txt

Podsumowanie

Zadanie 1 zostało zrealizowane na zajęciach.