Systemy wbudowane

Sprawozdanie z laboratorium 3

Mariusz Jędrzejewski / 128059 / 28.10.2019r

W trakcie laboratorium wykorzystywana była płytka z wyświetlaczem „STM32F429ZI”.

Spis przydatnych słów / poleceń używanych w kodzie pliku main.c :

#include “math.h” – biblioteka która wprowadza do programu funkcje matematyczne, takie jak exp, sin, cos.
volatile – tzw. zmienna ulotna, oznacza że zmienna może zostać zmieniona niezależnie od kodu program w jakim jest

W sekcji *Private variables* deklarujemy flagi na których będziemy operować. Naszym zadaniem jest przedstawienie czasu wymaganego na wykonanie operacji matematycznych. W naszym zadaniu wykonaliśmy takie operacje jak *dodawanie, odejmowanie, mnożenie, dzielenie, sinus z liczby* ***(math.h),*** *cosinus z liczby* ***(math.h),*** *funkcja exponencjalna* ***(math.h)***.
Do tych zmiennych również wstawiamy zmienną typu *float* która nazywa się *result*. Jest ona podana dla każdej funkcji.

```
44  /* Private variables -----------------------
45
46  /* USER CODE BEGIN PV */
47      volatile int32_t ai, bi, ci;
48      volatile double ad, bd, cd;
49      volatile float as, bs, cs;
50
51      int add_flag = 0; // dodawanie
52      int sub_flag = 0; // odejmowanie
53      int mul_flag = 0; // mnozenie
54      int div_flag = 0; // dzielenie
55      int sin_flag = 0; // sinus
56      int cos_flag = 0; // cosinus
57      int exp_flag = 0; // fun. exponencjalna
58      float result = 0;
59  /* USER CODE END PV */
```

Zmienne takie jak ai, bi itd. nie są używane, zostały zawarte w tym kodzie ze względu na proces przygotowawczy do laboratorium nr 3.

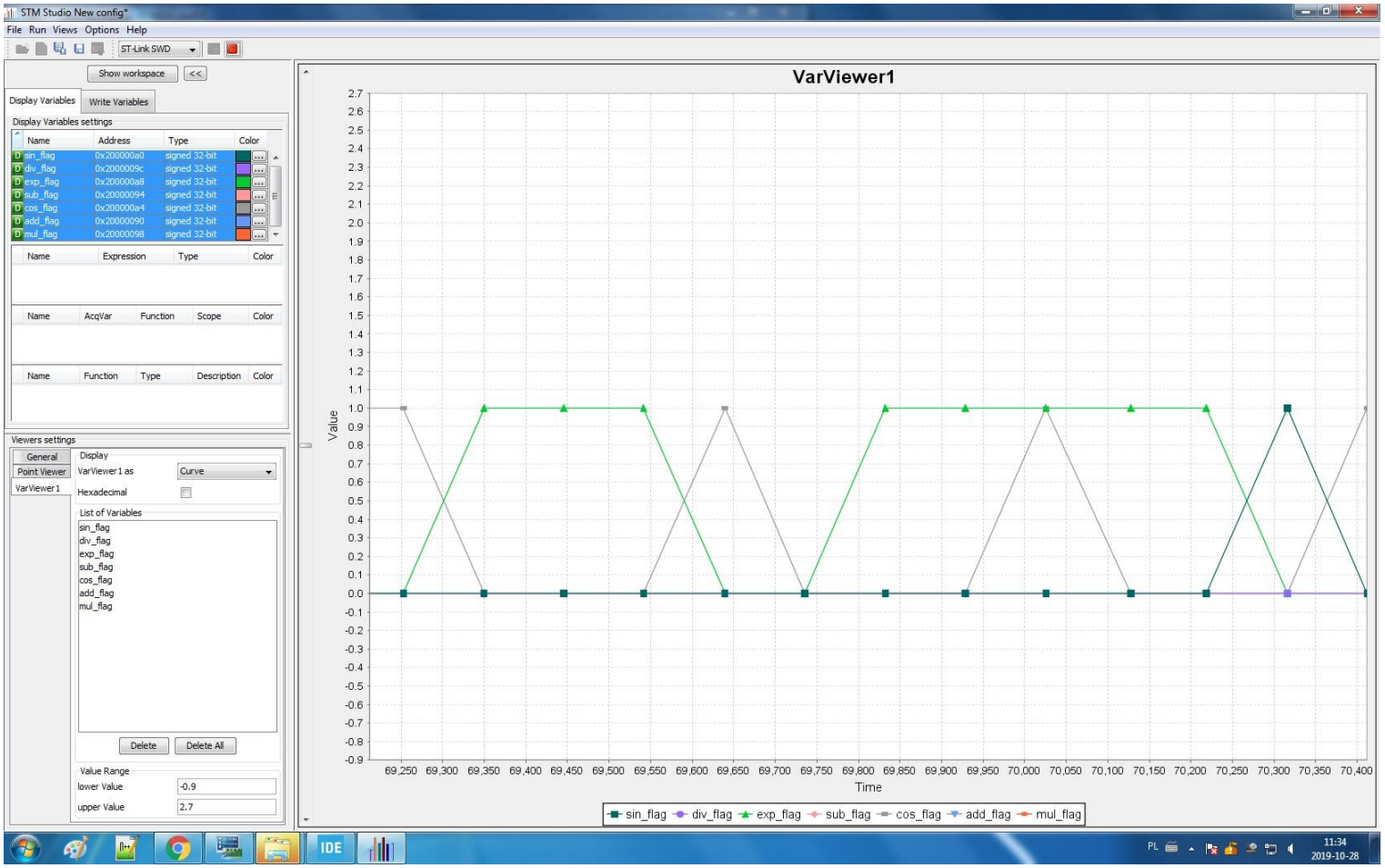
W funkcji *main(void)* deklarujemy w sekcji *USER CODE BEGIN 1* nasze zmienne liczbowe typu *float.*
Są to liczby *float a = 25* oraz *float b = 10.5*. Na tych liczbach wykonywane zostały wszystkie operacje matematyczne.

Nasze zadanie polegało na zaprogramowaniu płytki *STM32F429ZI* do wykonywania obliczeń matematycznych. Wstępnym założeniem było wykonanie zadania na oscyloskopie, ale można było wykonać zadanie w sposób taki, że nie wymagane było korzystanie z oscyloskopu i opuszczaniu oprogramowania dostarczonego nam w sali laboratoryjnej.

Kod wykonujący operacje matematyczne został umieszczony w środku pętli *while*.

```
107   /* Infinite loop */
108   /* USER CODE BEGIN WHILE */
109   while (1)
110   {
111     /* USER CODE END WHILE */
112
113     /* USER CODE BEGIN 3 */
114     add_flag = 1;
115     result = a + b;
116     add_flag = 0;
117
118     sub_flag = 1;
119     result = a - b;
120     sub_flag = 0;
121
122     mul_flag = 1;
123     result = a * b;
124     mul_flag = 0;
125
126     div_flag = 1;
127     result = a / b;
128     div_flag = 0;
129
130     sin_flag = 1;
131     result = sin(a);
132     sin_flag = 0;
133
134     cos_flag = 1;
135     result = cos(a);
136     cos_flag = 0;
137
138     exp_flag = 1;
139     result = exp(a);
140     exp_flag = 0;
```

Na poniższych załącznikach jest ukazane jak te operacje w programie *STM Studio* są wykonywane w czasie rzeczywistym i są powtarzane w nieskończoność. Jest tak dlatego że kod w pętli *while* będzie wykonywał się bez końca, tzw. *infinite loop.*



Na tym załączniku są pokazane kolory też naszych flag i operacji jakie wykonują po zaprogramowaniu płytki.

Na poniższym załączniku jest pokazane działanie programu wraz z wykorzystaniem przycisku który wstrzymuje działanie programu i wykonywania się operacji. Rozpoczyna działanie programu ponownie po opuszczeniu przycisku.
Po lewej stronie wykresu jest fragment czasu gdzie przycisk jest włączony, a po prawej gdy jest opuszczony (wykonuje się od początku i obliczenia są wykonywane).