

PP > WE Studia stacj. I st. Informatyka > sem. 4	Podstawy technik mikroprocesorowych	Rok ak. 2012/2013 Sem. letni
Artur Wujczak Mariusz Wielgus	Odtwarzacz prostych dźwięków sterowany z PC	Grupa: I2-2

Spis treści

Opis tematu projektu.....	2
Opis rozwiązania.....	2
Ważne elementy projektu.....	2
Opis interfejsu.....	4
Opis przeprowadzonych testów.....	4

Opis tematu projektu

Tematem projektu było stworzenie odtwarzacza prostych dźwięków, który jest sterowany z PC. W tym celu użyte zostało wyjście słuchawkowe oraz szeregowy magistrala I²C. W projekcie wykorzystane zostały wybrane dźwięki wirtualnego pianina, a także kilka gotowych, nieco dłuższych efektów dźwiękowych.

Opis rozwiązania

W trakcie tworzenia projektu korzystaliśmy ze środowiska programistycznego Visual C++, w którym stworzony został między innymi interfejs programu (wykorzystanie biblioteki graficznej allegro.h) oraz programu CoCo IDE, za pomocą którego sterujemy mikrokontrolerem STM32.

Pamięć procesora na mikrokontrolerze STM32 wynosi 1MB w związku z czym wielkość wykorzystywanych dźwięków jest ogólnie ograniczona. Wielkość programu to ponad 800kB co powoduje, że czas jego wgrywania jest stosunkowo długi.

Po wgraniu całego programu z komputera przesyłane są wytyczne, mówiące o tym, który z zawartych w pamięci dźwięków ma zostać odtworzony.

Ciekawym rozwiązaniem jest zapis dźwięków w formacie tekstowym, co zostało przedstawione przy prezentacji projektu.

Ważne elementy projektu

```
SPI_I2S_DeInit(CODEC_I2S);
I2S_InitType.I2S_AudioFreq = 24000;//12000//Normal 10000
I2S_InitType.I2S_MCLKOutput = I2S_MCLKOutput_Enable;
I2S_InitType.I2S_DataFormat = I2S_DataFormat_16b;
I2S_InitType.I2S_Mode = I2S_Mode_MasterTx;
I2S_InitType.I2S_Standard = I2S_Standard_Phillips;
I2S_InitType.I2S_CPOL = I2S_CPOL_Low;
```

Powyższy kod przedstawia konfigurację portu I²S. Częstotliwość jaką wybraliśmy wynosi 24KHz.

```
I2C_DeInit(CODEC_I2C);
I2C_InitType.I2C_ClockSpeed = 100000;
I2C_InitType.I2C_Mode = I2C_Mode_I2C;
I2C_InitType.I2C_OwnAddress1 = CORE_I2C_ADDRESS;
I2C_InitType.I2C_Ack = I2C_Ack_Enable;
I2C_InitType.I2C_AcknowledgedAddress = I2C_AcknowledgedAddress_7bit;
I2C_InitType.I2C_DutyCycle = I2C_DutyCycle_2;
```

Tutaj przedstawiony jest zapis konfiguracji portu I²C

```
if( OutBuffer[0]==3)
{
    if (SPI_I2S_GetFlagStatus(CODEC_I2S, SPI_I2S_FLAG_TXE))
    {
        sample=d1[sampleCounter];

        SPI_I2S_SendData(CODEC_I2S, sample);

        if(iter%2==1)
            sampleCounter++;
        iter++;
    }
    if (sampleCounter == d1s)sampleCounter = 0;
}
```

Przykład przesłania informacji, który sampel ma zostać nagrany.

```
if((mouse_x>15 && mouse_y>185 && mouse_x<50 && mouse_y<250 && mouse_b==1) ||
key[KEY_Z])
{
    buffer_out[1]=1;
    if(debug==1)textprintf_ex( bufor, czcionka, x, y, makecol( 0, 0,0 ),-1,
"Wcisnieto klawisz: C1");
}
```

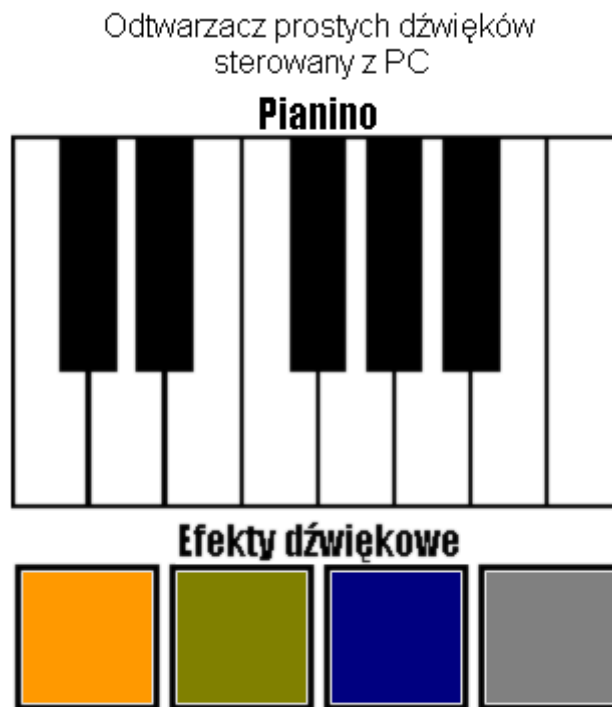
Powyższy fragment kodu ukazuje „kolizję” kursora myszy z przykładowym klawiszem pianina wyświetlonym w interfejsie co pozwala na przesłanie informacji o odtworzeniu jego dźwięku. Wybranymi klawiszami można również sterować za pomocą klawiatury co pokazuje konkretny fragment `key[KEY_Z]`.

Znaczna część kodu przedstawionego powyżej powtarza się z racji tego, że dźwiękami sterujemy za pomocą 13 klawiszy, więc ukazywanie go w tym rozdziale nie ma większego sensu.

Opis interfejsu

Interfejs programu jest bardzo prosty i intuicyjny. Nie zawiera niepotrzebnych „udogodnień”, a tylko konkretne klawisze do przesyłania pojedynczych dźwięków oraz gotowych efektów dźwiękowych, o których było wspomniane wcześniej.

Graficznie interfejs przedstawiony jest na poniższym obrazie:



Mariusz Wielgus, Artur Wujczak

Opis przeprowadzonych testów

Przy wielokrotnym uruchamianiu programu nie dostrzeżliśmy błędów w działaniu, a dźwięki były odtwarzane poprawnie. Problemem okazał się szum w momencie kiedy nie zostawał wywoływany żaden z dźwięków. Rozwiązaniem tego okazało się wgranie dźwięku „ciszy”, który jest domyślnie wywoływany, jeśli użytkownik nie użyje żadnego z przycisków programu. Momentami traciliśmy połączenie z płytką ale przyczyną tego był wadliwy port USB.

Jako że tematem jest odtwarzanie dźwięków, nie można zaprezentować jego działania w inny sposób, jak tylko przy oddawaniu projektu.