

Piotr Radecki

## Raport z projektu:

# „Stream audio przez bluetooth z użyciem modułu HC-05 i płytki FRDM-KL05Z”

## 1. Opis projektu

Celem projektu było stworzenie układu transmitującego dźwięk z mikrofonu elektretowego na komputer i odtwarzanie tego dźwięku „na żywo”- czyli w praktyce z niewielkim opóźnieniem.

Projekt składa się z trzech znaczących komponentów:

- Wzmacniacz analogowy
- Płytką FRDM-KL05 z modułem HC-05
- Komputer PC

Każdy z komponentów zostanie bardziej szczegółowo omówiony w kolejnych punktach sprawozdania.

## 2. Układ Analogowy

Układ został zaprojektowany przeze mnie. Z uwagi na brak możliwości zmierzenia napięcia na mikrofonie parametry wzmocnienia zostały dobrane metodą prób i błędów. Układ składa się ze wzmacniacza, wtórnika emiterowego i filtra dolnoprzepustowego RC. Głównym założeniem było uzyskanie napięcia stałego na wyjściu wtórnika rzędu 1.5V, co stanowi około połowę napięcia referencyjnego przetwornika AC(2.91V) w mikrokontrolerze. Uzyskane napięcie stałe na wyjściu to około 1.7-1.8V co stanowi zadowalający efekt. Użyte tranzystory bipolarne to modele BC546B. Częstotliwość 3dB filtra to 8kHz. Dzięki temu próbkowanie z częstotliwością 16kHz nie powinno spowodować znaczącego aliasingu. Mikrofon modelowany jest źródłem napięcia AC i kondensatorem. Układ zasilany jest bateryjnie z 12V. Poniżej schemat układu analogowego(z uwagi na dostępne komponenty w mojej dystrybucji programu spice, na schemacie przedstawione są tranzystory BC547B- model o podobnej charakterystyce):

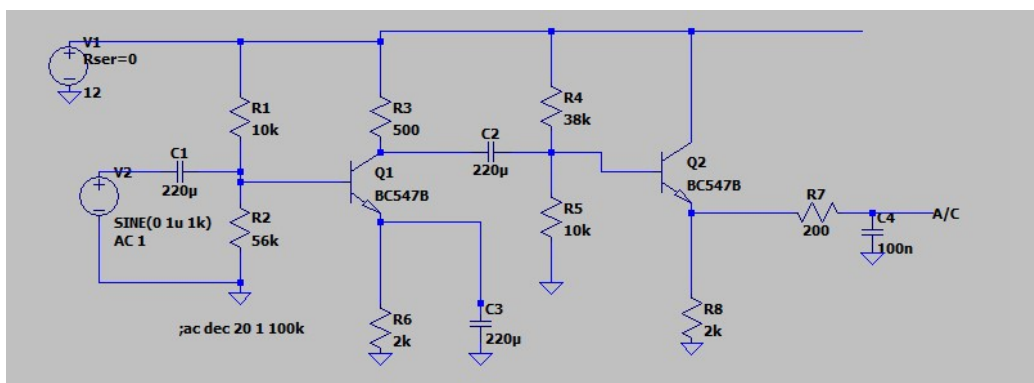


fig. 1

Wzmocnienie układu w funkcji częstotliwości:

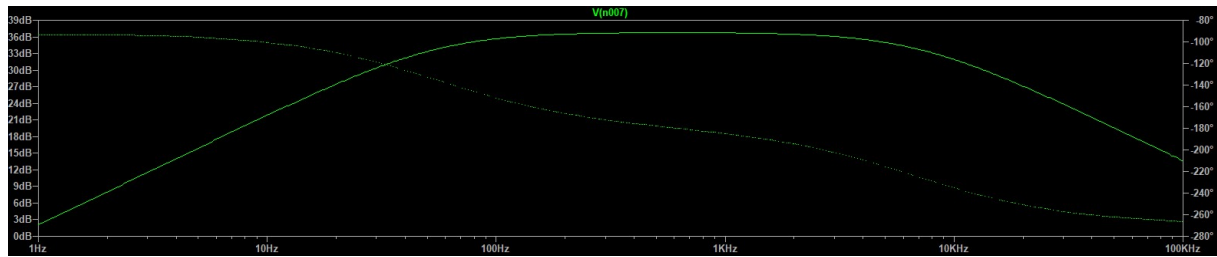


fig.2

Jak widać na powyższym wykresie wzmacniacz ma przyzwoitą charakterystyką w paśmie akustycznym.

### 3. Płytką FRDM-KL05Z z modułem HC-05

Projekt obejmuje dwa projekty napisane na płytce:

- Programator
- Program właściwy przerażający dane

Poza modułem HC-05 używany jest jeszcze wyświetlacz LCD z expanderem i2c, z którego korzystano na zajęciach laboratoryjnych.

**Uwaga! To sprawozdanie odnosi się do posiadanej przeze mnie płytki z modułem HC-05. W tej konkretnej dystrybucji znajduje się przycisk do programowania a także wejście Tx należy połączyć z wyjściem Tx płytki FRDM i analogicznie Rx należy połączyć z Rx. Te parametry mogą się różnić dla innych płytek!**

#### 3.1 Moduł HC-05

Moduł korzysta z komunikacji uart. Możliwe jest używanie go w dwóch trybach:

- Tryb komend
- Tryb komunikacji

W trybie komend używana jest komunikacja z baud rate 38400. W tym trybie przesyłane są komunikaty AT programujące pamięć nieulotną modułu. Możliwe jest ustawienie między innymi takich parametrów jak nazwa, prędkość transmisji, tryb master/slave, hasło. Każdy z komunikatów to string zakończony „\r\n”. Lista komend:

[HC-05 Bluetooth AT Command List | Microcontroller Tutorials \(teachmemicro.com\)](https://teachmemicro.com/hc-05-bluetooth-at-command-list/)

W trybie komunikacji używany jest baud rate ustawiony w trybie komend(domyślnie 9600). Wysyłane są dowolne strumienie bitów w protokole uart. Moduł z punktu widzenia mikrokontrolera to urządzenie połączone za pomocą uart, które nadaje i odbiera dane.

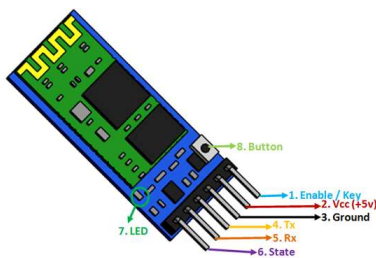


fig. 3- moduł HC-05

### 3.2 Programator

Programator znajduje się w folderze **programmer** w repozytorium. Schemat połączeń znajduje się poniżej:

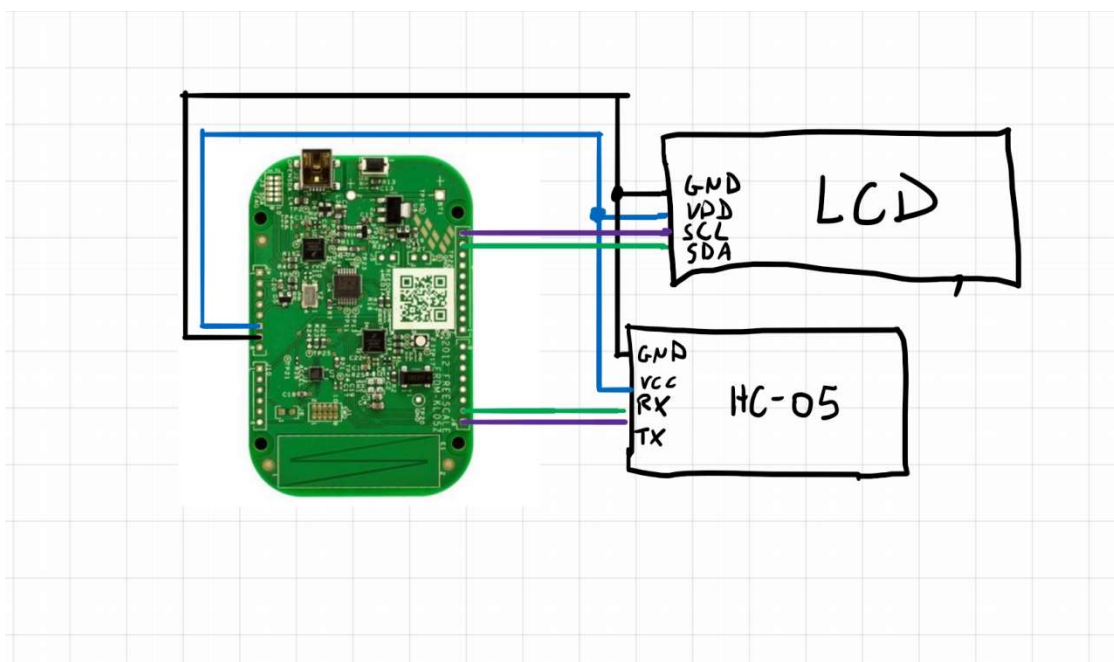


fig. 4

**Podczas włączania zasilania modułu HC-05 należy trzymać wciśnięty przycisk aby przejść do trybu komend! Aby wyjść z tego trybu należy zresetować moduł.**

Po wgraniu programu na płytkę zostaną ustawione parametry modułu HC-05. Program powinien ustawić kolejno: `uart(230400,0,0)`, nazwę urządzenia(TMP2) i hasło(9999). Ponadto powinny zostać przysłane potwierdzenia i wartość ustawionych parametrów. Kolejne etapy i otrzymane odpowiedzi będą wyświetlane na wyświetlaczu LCD. Format ustawionego `uart` oznacza: baud rate=230400, jeden bit stopu, brak kontroli parzystości.

Najważniejsze funkcje:

- `uart_receive()`- odbiera daną z `uart` gdy będzie gotowa, czyści ekran LCD i wyświetla daną jako stringa na ekranie.
- `uart_hc_05_ms(char* string)`- jest odpowiedzialna za wysłanie pojedynczego komunikatu, odebranie odpowiedzi i wyświetleniu odpowiedzi na ekranie
- `UART_Init(uint32_t baud_rate)`- pozwala ustawić żądany baud rate i aktywować moduł `uart`.

Program główny znajduje się w zakładce **stream** w repozytorium. Schemat połączeń znajduje się poniżej:

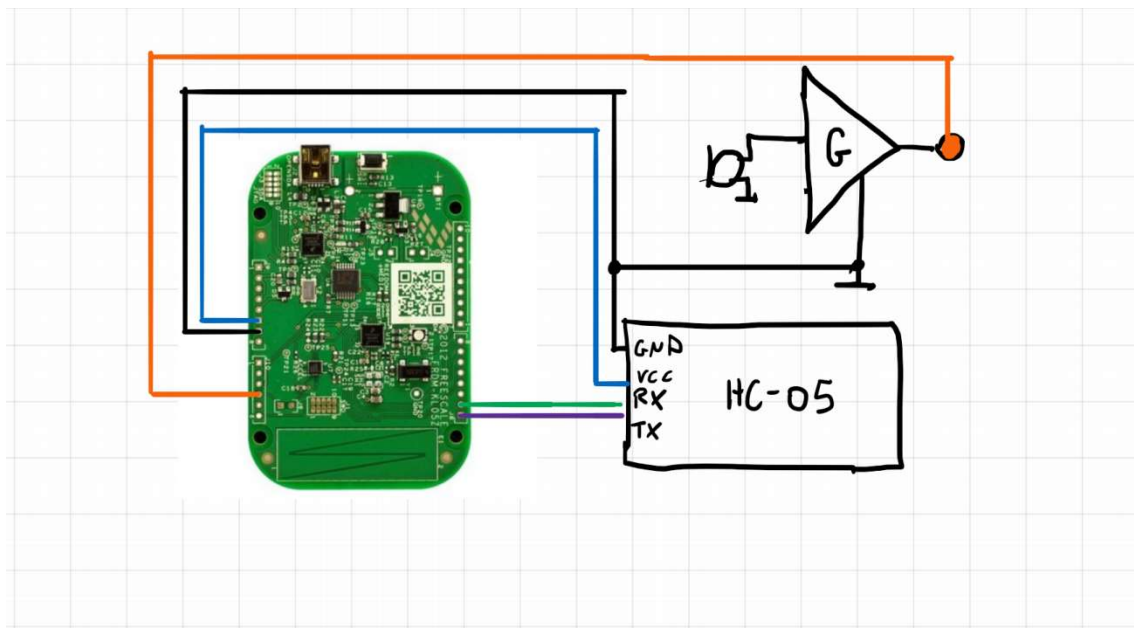


fig. 5

Program jest taktowany licznikiem PIT. 16 tysięcy razy na sekundę PIT przysyła przerwanie. Wówczas ostatnia dana odebrana z przerwania przetwornika A/C jest zapisywana do zmiennej wynik, przesuwana o 4 miejsca w prawo (w celu zrzutowania danej 12-bitowej na daną 8-bitową) i wysyłana przez UART z baud rate 230400. Zmienna pit\_ok odpowiada za handshake pomiędzy przerwaniem a pętlą główną. W praktyce układ stanowi zatem układ próbkujący z częstotliwością 16kHz. Z uwagi na filtr antyaliasingowy na wyjściu wzmacniacza ( $f_{3dB} = 8\text{kHz}$ ) i odpowiednio dobraną częstotliwość próbkowania możliwe powinno być uzyskanie przyzwoitej jakości dźwięku.

Ponownie korzysta się z funkcji `UART_Init ()` pozwalającej ustawić baud rate. Program korzysta z 12 kanału przetwornika A/C.

4. Komputer PC z systemem windows i modułem bluetooth pełniący rolę odbiornika

Aplikacja zna

## 4.1 Konfiguracja

Aby odpowiednio odbierać dane na komputerze z systemem windows należy najpierw odpowiednio skonfigurować porty w systemie.

Aby to zrobić należy:

- Użyć programatora w celu ustawienia odpowiednich parametrów modułu hc-05
- Podłączyć moduł do zasilania
- wejść w Bluetooth & other devices > Add Bluetooth or other device > Bluetooth a następnie wybrać urządzenie TMP2 i wpisać PIN: 9999, urządzenie zostanie sparowane.

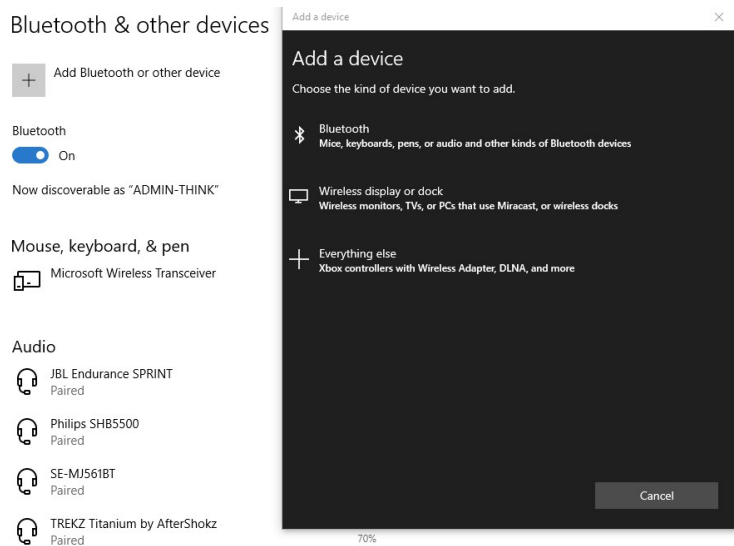


fig. 6

- Następnie należy wejść w zakładkę More Bluetooth options

Related settings

[Devices and printers](#)

[Sound settings](#)

[Display settings](#)

[More Bluetooth options](#)

[Send or receive files via Bluetooth](#)

fig. 7

- Po wyświetleniu się menu należy wejść w zakładkę COM ports, wybrać opcję outgoing a następnie wybrać z rozwijanej listy urządzenie TMP2 i Service: SPP Dev. Po naciśnięciu „ok” urządzenie zostanie przypisane do odpowiedniego portu(w moim przypadku jest to COM5).

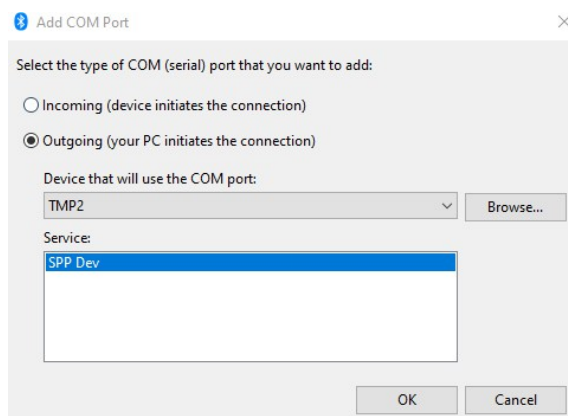


fig. 8

## 4.2 Odbiornik

Odbiornik został napisany w języku python. Jego zadaniem jest odczyt danych z portu szeregowego i transmisja tych danych na głośnik. Cały tor transmisyjny powoduje bardzo niewielkie opóźnienia, w związku z czym dźwięk jest transmitowany na żywo.

Oprócz transmisji, program realizuje też zapis dźwięku. Podczas wykonywania programu dane przechowywane są w dynamicznej tablicy. Po zakończeniu programu użytkownik ma możliwość zapisu sesji w pliku o formacie .wav.

Odbiornik dostarcza prosty interfejs graficzny użytkownika. Po włączeniu programu i inicjalizacji portu użytkownik ma dostęp do okienka z trzema opcjami: start, stop, quit.

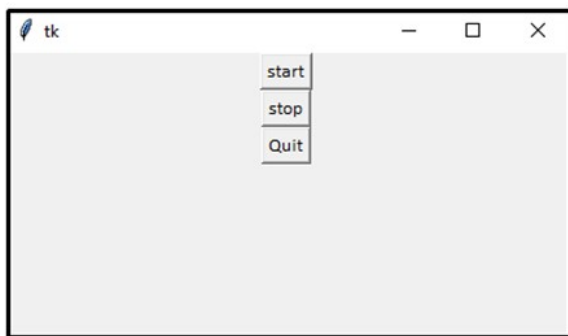


fig. 9

- Opcja start:

Po wystartowaniu program zaczyna transmitować i zapisywać dźwięk, trwa to aż do momentu naciśnięcia przycisku stop.

- Opcja stop:

Program czyta dane z bufora jednak nie zapisuje ich ani nie transmituje. Czytanie ma na celu jedynie opróżnienie bufora.

- Opcja quit:

Po wybraniu tej opcji zamykany jest port a użytkownikowi wyświetla się przebieg czasowy dźwięku. Po zamknięciu wykresu użytkownik jest pytany o chęć zapisu dźwięku do pliku. W przypadku wpisania ścieżki do pliku .wav zostanie utworzony nowy plik lub nadpisany stary o tej samej nazwie. W przypadku wybrania opcji cancel dane nie zostaną zapisane.