

Współpraca STM32 z modułem Bluetooth HC-05

Moduł Bluetooth HC-05 pozwala na łatwe zrealizowanie bezprzewodowej transmisji danych pomiędzy systemem embedded a np. smartfonem lub tabletem. Jego wykorzystanie w systemach z kontrolerem STM32 nie wymaga skomplikowanych procedur. Większość pracy związanej z nawiązaniem połączenia wykonuje oprogramowanie samego modułu. W tym artykule zademonstruję jak skonfigurować moduł, aby można go było użyć we własnej aplikacji.

Bluetooth pozwala na lokalną transmisję danych na odległość (zależnie od klasy modułów) od kilku do 150 m. Transmisja jest przeprowadzana w obrębie pasma 2,4 GHz. Wbudowane w standard procedury pozwalają na automatyczne wyszukiwanie działających w pobliżu i dostępnych urządzeń Bluetooth, nawiązanie połączenia i przesłanie danych.

Moduł HC-05 można wstępnie skonfigurować tekstowymi komendami w trybie AT. Komendy przesyłane są do modułu za pośrednictwem jego portu szeregowego UART. Po zakończeniu konfiguracji moduł przechodzi do normalnego trybu pracy, czyli przesyłania danych łączem radiowym.

Moduł HC-05 jest dostępny w kilku wersjach. Do opisu i testów posłużyłem się modułem zintegrowanym z anteną i zamontowanym na dodatkowej płytce ze złączem 6-stykowym. Oprócz złącza na płytce zamontowano między innymi stabilizator, miniaturowy przycisk i diodę sygnalizacyjną LED. Inne wersje wykonania będą działać podobnie i różnić się jedynie sposobem wprowadzenia modułu w tryb AT.

Złącze

Na 6-szpilkowe złącze wyprowadzono wszystkie dostępne sygnały modułu. W tabeli zebrano oznaczenie kolejnego styku oraz opis pełnionej funkcji.

Tabela 1. Opis złącza modułu HC-05

Numer styku	Nazwa	Opis
1	EN	Wejście z wewnętrznym podciąganiem do poziomu wysokiego. Poziom wysoki – tryb normalnej pracy Poziom niski – tryb wstrzymania funkcjonowania modułu z obniżonym poborem prądu
2	VCC	Wejście napięcia zasilania, nominalnie 3,6...6 V
3	GND	Masa
4	TXD	Wyjście danych do zewnętrznego urządzenia.
5	RXD	Wejście danych z zewnętrznego urządzenia
6	STATE	Wyjście sygnalizujące status modułu

Nominalnie moduł powinien być zasilany napięciem z przedziału 3,6...6V, które przez zamontowany na płytce stabilizator jest obniżane do 3,3 V. Jednak moduł będzie pracować także przy zasilaniu napięciem o wartości 3,3 V. Nie jest to technicznie najlepsze rozwiązanie jednak pozwala skorzystać z zasilania układu kontrolera STM32.

W trakcie parowania moduł pobiera przeciętnie 40 mA, po zainicjowaniu połączenia kilka miliamper. Wszystkie pozostałe linie wejściowe są w standardzie

Listing 1. Definicje nazw symbolicznych

```
#define USART_2 USART2
#define USART_2_GPIO GPIOA
#define USART_2_CLK RCC_APB1Periph_USART2
#define USART_2_GPIO_CLK RCC_APB2Periph_GPIOA
#define USART_2_RxPin GPIO_Pin_3
#define USART_2_TxPin GPIO_Pin_2
#define USART_2_IRQn USART2_IRQn
#define USART_2_IRQHandler USART2_IRQHandler
```

CMOS 3,3 V, więc nie wymagają konwerterów do współpracy z portami kontrolera STM32. Poziom wysoki wyjścia STATE sygnalizuje nawiązanie połączenia ze sparowanym urządzeniem.

Status modułu jest sygnalizowany za pomocą diody LED w następujący sposób:

Szybkie migotanie: gotowość do nawiązania połączenia.

Dwa błyski – przerwa – dwa błyski: sygnalizacja nawiązanego połączenia.

Wolne migotanie: urządzenie w trybie konfiguracji komendami AT.

Profil SPP i tryb pracy modułu

Cechą charakterystyczną standardu Bluetooth jest korzystanie z tzw. profili. Moduły HC-05 obsługują jeden profil

– SPP. Oznacza to, że są zdolne do przesyłania danych w trybie symulacji portu szeregowego UART. Moduł może współpracować z dowolnym innym modulem i urządzeniem z profilem SPP. Urządzenia nieobsługujące tego profilu mogą wykrywać HC-05 jednak nawiązanie połączenia i transmisja pomiędzy urządzeniem a modulem nie będzie możliwa.

Każdy moduł HC-05 można ustawić do pracy jako urządzenie nadrzędne (master) lub podrzędne (slave). Tylko urządzenie nadrzędne może inicjować połączenie i realizować dwustronną wymianę danych z urządzeniem podrzędnym. W lokalnym systemie może funkcjonować wiele urządzeń podrzędnych, ale tylko jedno nadrzędne (master).

Typowe podłączenie modułu do kontrolera STM32

Przykładowe podłączenie modułu HC-05 do kontrolera STM32 pokazano na **rysunku 1**. Do komunikacji z modulem wybrany został port UART2 kontrolera. Wyprowadzeniem EN modułu może sterować dowolny port kontrolera. W **tabeli 2** wymieniono połączenia pomiędzy modulem HC-05 a systemem z mikrokontrolerem STM32.

Listing 2. Deklaracja buforów i wskaźników do obsługi transmisji

```
//zmienne procedur UART2
char bufor_rx_USART2[ROZMIAR_BUF_RX_TX]; //bufor danych odbieranych poprzez UART2
char bufor_tx_USART2[ROZMIAR_BUF_RX_TX]; //bufor danych wysyłanych poprzez UART2

char *p_bufor_rx_USART2; //wskaźnik do początku buforu odbiorczego UARTA2
int bufor_rx_USART2_rozmiar; //zmienna określająca całkowity rozmiar buforu odbiorczego UARTA2
char *p_rx_USART2_in; //wskaźnik bieżącej pozycji w buf. odbiorczym dla danych umieszczonych w buf.
char *p_rx_USART2_out; //wskaźnik bieżącej pozycji w buf. odbiorczym dla danych odczytywanych z buf.

char *p_bufor_tx_USART2; //wskaźnik do początku buforu transmisji UARTA2
int bufor_tx_USART2_rozmiar; //zmienna określająca całkowity rozmiar buforu transmisji UARTA2
char *p_tx_USART2_in; //wskaźnik bieżącej pozycji w buf. nadawczym dla danych umieszczonych w buf.
char *p_tx_USART2_out; //wskaźnik bieżącej pozycji w buf. nadawczym dla danych odczytywanych z buf.
char tx_USART2_restart flag; //gdy ,1' poprzednia transmisja zakończona, nowa musi być zainicjowana
```

Listing 3. Inicjacja wyprowadzeń mikrokontrolera jako wejścia/wyjścia UART

```
// inicjacja i konfiguracja linii I/O będących wyprowadzeniami Rx, Tx portu UART2
void USART2_GPIO_Configuration(void)
{
    GPIO_InitTypeDef GPIO_InitStructure;
    /* Configure USART2 Rx as input floating */
    GPIO_InitStructure.GPIO_Pin = USART_2_RxPin;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_IN_FLOATING;
    //GPIO_InitStructure.GPIO_Mode = GPIO_Mode_IPU;
    GPIO_Init(USART_2_GPIO, &GPIO_InitStructure);
    /* Configure USART2 Tx as alternate function push-pull */
    GPIO_InitStructure.GPIO_Pin = USART_2_TxPin;
    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AF_PP;
    GPIO_Init(USART_2_GPIO, &GPIO_InitStructure);
}
```

Listing 4. Inicjowanie UART do pracy w trybie z przerwaniami

```
//procedury inicjacji portu UART2
void USART2_Inicjacja(uint32_t usart_speed)
{
    /* USART configured as follow:
    - BaudRate = usart_speed baud
    - Word Length = 8 Bits
    - One Stop Bit
    - No parity
    - Hardware flow control disabled (RTS and CTS signals)
    - Receive and transmit enabled
    */

    /* Enable GPIO clock */
    RCC_APB2PeriphClockCmd(USART_2_GPIO_CLK | RCC_APB2Periph_AFIO, ENABLE);
    /* Enable USART2 Clock */
    RCC_APB1PeriphClockCmd(USART_2_CLK, ENABLE);
    /* PCLK1 = HCLK/4 */
    RCC_PCLK1Config(RCC_HCLK_Div4);
    USART_Cmd(USART_2, DISABLE);
    USART_InitStructure.USART_BaudRate = usart_speed;
    USART_InitStructure.USART_WordLength = USART_WordLength_8b;
    USART_InitStructure.USART_StopBits = USART_StopBits_1;
    USART_InitStructure.USART_Parity = USART_Parity_No;
    USART_InitStructure.USART_HardwareFlowControl = USART_HardwareFlowControl_None;
    USART_InitStructure.USART_Mode = USART_Mode_Rx | USART_Mode_Tx;
    /* Configure USART2 */
    USART_Init(USART_2, &USART_InitStructure);
    /* Enable UART Receive interrupts UART2*/
    USART_ITConfig(USART_2, USART_IT_RXNE, ENABLE); //zezwozenie na przerwanie odbioru
    //zezwozenie na przerwanie transmisji dopiero gdy rozpocznie się wysyłanie zawartości buf. TX
    /* Enable the USART2 */
    USART_Cmd(USART_2, ENABLE);
}
```

Procedury inicjacji portu UART2

Mikrokontroler nadzoruje pracę modułu za pomocą interfejsu UART2. Poprzez niego są wysyłane i odbierane dane oraz polecenia konfiguracyjne. Transmisja jest obsługiwana w trybie przerwania. Dane odbierane i wysyłane są umieszczane w osobnych buforach w pamięci RAM mikrokontrolera. Procedury związane z portem UART2 posługują się nazwami, które wcześniej należy umieścić np. w osobnym pliku nagłówkowym (**listing 1**). Następnie należy zadeklarować bufony i wskaźniki do obsługi transmisji danych wysyłanych i odbieranych, co pokazano na **listingu 2**. Procedurę inicjacji portów kontrolera pełniących funkcje wyprowadzeń RxD i TxD UART2 pokazano na **listingu 3**, natomiast inicjacja UART2 do pracy z przerwaniem może wyglądać jak na **listingu 4**. W pliku *stm32f10x_it.c* należy umieścić obsługę przerwania UART2, co pokazano na **listingu 5**. Należy także zainicjować kontroler przerwania NVIC, jak na **listingu 6**. Wszystko razem można obsługiwać jedną procedurą inicjacji, co pokazano na **listingu 7**. Po inicjacji program kontrolera może korzystać z portu UART2 do komunikacji z modułem HC-05. Do kontroli wypełnienia bufora odbiorczego wykorzystuje się porównanie wskaźników *ilosc_danych_odebranych = p_rx_USART2_in - p_rx_USART2_out*.

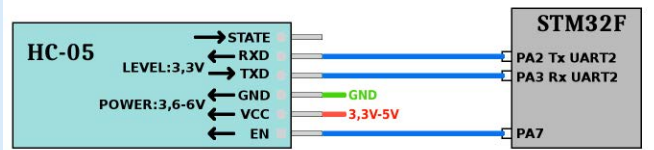
Po wykorzystaniu odebranych danych należy oba wskaźniki ustawić na początek bufora odbiorczego *bufor_rx_USART2[]*. Wysyłając dane do modułu należy najpierw sprawdzić stan flagi *tx_USART2_restart_flag*. Jeżeli jest ustawiona oznacza to, że poprzednia transmisja została zakończona i można wysłać nowe dane. Po zapisaniu danych do wysłania w buforze *bufor_tx_USART2[]* należy wskaźnik *p_tx_USART2_in* ustawić na początek bufora. Wskaźnik *p_tx_USART2_out* powinien wskazywać pozycję *bufor_tx_USART2[0]+ilość danych w buforze nadawczym*. Następnie należy zainicjować transmisję np. tak:

```
if (tx_USART2_restart_flag == 1) // if transmit interrupt is disabled, enable it
{
    tx_USART2_restart_flag = 0;
    // enable TX interrupt
    USART_ITConfig(USART2, USART_IT_TXE, ENABLE);
}
```

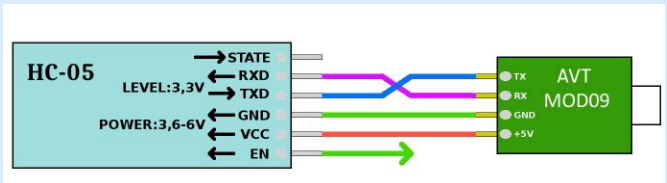
Jeżeli do sterowania wejściem EN modułu użyty zostanie port kontrolera np. PA7 należy go zainicjować jako wyjściowy z wewnętrznym podciąganiem. Następnie należy ustawić port w stanie wysokim.

Tabela 2. Połączenia pomiędzy modułem HC-05 a mikrokontrolerem STM32

Wyprowadzenia HC-05	Port kontrolera STM32F	Ustawienia portu kontrolera
RxD	PA2	Linia TxD UART2
TxD	PA3	Linia RxD UART2
EN	PA7	Port wyjściowy z wewnętrznym podciąganiem do poziomu wysokiego



Rysunek 1. Przykładowe podłączenie modułu HC-05 do kontrolera STM32



Rysunek 2. Połączenie modułu HC-05 do portu komputera poprzez interfejs AVT MOD09

Tryb AT modułu HC-05

Dostęp do modułu w trybie AT pozwala na zmianę ustawień takich parametrów jak: tryb pracy, nazwa, szybkość transmisji portu szeregowego i innych. Komendy konfiguracyjne wysyłane są w trybie tekstowym, więc zmianę ustawień HC-05 można przeprowadzić używając dowolnego programu terminalowego uruchomionego na komputerze. Ja do testów posłużyłem się programem „Br@y++” w wersji 1,93b. Do komunikacji z modułem można wykorzystać port USB komputera i interfejs dopasowujący, którego wyprowadzenia TxD i RxD przystosowane są do pracy z poziomami COMS 3,3 V. Połączenie modułu HC-05 do portu komputera poprzez interfejs AVT MOD09 zostało pokazane na **rysunku 2**.

Listing 5. Obsługa przerwania od UART2

```
// obsługa przerwania portu szeregowego UART2
void USART_2_IRQHandler(void)
{
    //obsługa przerwania wywołanego odebraniem kolejnego bajtu danych
    if(USART_GetITStatus(USART_2, USART_IT_RXNE) != RESET)
    {
        //przepisanie odebranego bajtu do bufora odbiorczego
        *p_rx_USART2_in =USART_ReceiveData(USART_2);
        p_rx_USART2_in++;
        if (p_rx_USART2_in >=(p_bufor_rx_USART2 +bufor_rx_USART2_rozmiar))//kołowy bufor Rx UART2
        {
            p_rx_USART2_in =p_bufor_rx_USART2;
        }
    }
    //obsługa przerwania wywołanego zakończeniem wysłania kolejnego bajtu danych
    if(USART_GetITStatus(USART_2, USART_IT_TXE) != RESET)
    {
        if (p_tx_USART2_in !=p_tx_USART2_out)
        {
            //w buforze transmisji są dane do wysłania
            USART_SendData(USART_2, *p_tx_USART2_out);
            p_tx_USART2_out++;
            if (p_tx_USART2_out >=(p_bufor_tx_USART2 +bufor_tx_USART2_rozmiar))
            {
                p_tx_USART2_out =p_bufor_tx_USART2;
            }
        }
        else
        {
            //cała zawartość bufora transmisji została wysłana,
            //wyłączenie zezwolenia na przerwanie transmisji
            USART_ITConfig(USART_2, USART_IT_TXE, DISABLE);
            tx_USART2_restart_flag = 1;
        }
    }
}
```

Listing 6. Inicjowanie NVIC

```
void NVIC_Configuration(void)
{
    NVIC_InitTypeDef NVIC_InitStructure;
    /* Configure the NVIC Preemption Priority Bits */
    NVIC_PriorityGroupConfig(NVIC_PriorityGroup_2);
    NVIC_InitStructure.NVIC_IRQChannel = USART_2_IRQn;
    NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = 1;
    NVIC_InitStructure.NVIC_IRQChannelSubPriority = 0;
    NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;
    NVIC_Init(&NVIC_InitStructure);
}
```

Moduł wprowadza się w tryb przyjmowania komend AT dwoma sposobami:

Dołączając zasilanie należy naciskać przycisk zamontowany na module.

Przy włączonym zasilaniu należy nacisnąć przycisk na module. Następnie zeruje się i ustawia wejście EN.

Moduł sygnalizuje wejście w tryb przyjmowania komend AT wolnym migotaniem diody LED zamontowanej na płycie modułu. W tym trybie szybkość transmisji portu szeregowego modułu ustawiona jest zawsze na 38400. Można sprawdzić czy moduł znajduje się w trybie AT wysyłając tekst „AT” zakończony znakami CR-LF (dziesiętnie 13 i 10).

Jako potwierdzenie moduł powinien odesłać tekst „OK”.

Ustawienie modułu HC-05 w trybie podrzędnym (slave)

Przykładowa sekwencja rozkazów zeruje dotychczasowe ustawienia modułu, ustawia go w trybie podrzędnym, ustawia standardowy kod dostępu, nadaje nazwę, odczytuje adres oraz ustawia szybkość transmisji portu szeregowego. Użyte rozkazy wraz z opisem zostały wymienione w tabeli 3.

Każdy rozkaz musi być zakończony znakami CR LF (dziesiętnie 13 i 10). Poniżej przykładowa sekwencja

rozkazów i odpowiedzi modułu wyświetlana na pulpicie terminalu:

```
AT
OK
AT+ORGL
OK
AT+RMAAD
OK
AT+PSWD=1234
OK
AT+NAME=Slave
OK
AT+ROLE=0
OK
AT+ADDR?
+ADDR:2015:3:275580
OK
AT+UART=38400,0,0
OK
```

Testowanie transmisji do modułu HC-05 pracującego w trybie podrzędnym

Do przetestowania działania modułu pracującego w trybie podrzędnym potrzebne będzie urządzenie z interfejsem Bluetooth. Może to być komputer lub smartfon. Z reguły telefony nie obsługują profilu SPP, jednak pozwalają na uruchomienie program terminala z programową obsługą profilu. Z dostępnych programów terminalowych dla urządzeń z systemem operacyjnym Android można zastosować jeden z darmowych terminali: *BT Simple Terminal*, *Bluetooth spp pro*, *BlueTooth Serial Controler 16*. W przypadku niektórych z nich może okazać się konieczne samodzielne sparowanie z telefonem nowego urządzenia Bluetooth, którym będzie moduł HC-05.

Do testu moduł HC-05 powinien być połączony z komputerem, na którym uruchomiono zwykły program

Listing 7. Główna procedura inicjująca UART

```
void USART2_Ini(void)
{
    uart_speed = 38400; // 38400 zadeklarowana szybkość transmisji portu szeregowego
    USART2_GPIO_Configuration();
    USART2_Inicjacja(uart_speed);
    NVIC_Configuration();
    USART_ClearFlag(USART_2, USART_FLAG_TC | USART_FLAG_RXNE);
    USART_ClearITPendingBit(USART_2, USART_IT_TC | USART_IT_RXNE);
    p_bufor_rx_USART2 = &bufor_rx_USART2[0];
    bufor_rx_USART2_rozmiar = sizeof(bufor_rx_USART2);
    p_rx_USART2_in = &bufor_rx_USART2[0];
    p_rx_USART2_out = &bufor_rx_USART2[0];
    p_bufor_tx_USART2 = &bufor_tx_USART2[0];
    bufor_tx_USART2_rozmiar = sizeof(bufor_tx_USART2);
    p_tx_USART2_in = &bufor_tx_USART2[0];
    p_tx_USART2_out = &bufor_tx_USART2[0];
    tx_USART2_restart_flag = 1; // ustawienie flagi konieczności startu przerwania transmisji USART2
}
```

Tabela 3. Ustawienie modułu w trybie podrzędnym

Rozkaz AT	Odpowiedź HC-05	Opis
AT	OK	Potwierdzenie gotowości modułu do przyjmowania rozkazów AT
AT+ORGL	OK	Ustawienie parametrów predefiniowanych: ID:0 Working Role: Slave Baudrate: 38400bits/s or 9600bits/s Stop bit: 1 Parity: None Pair Code: 1234
AT+RMAAD	OK	Zerowanie listy sparowanych urządzeń
AT+PSWD=1234	OK	Ustawienie kodu dostępu
AT+NAME=Slave	OK	Nadanie modułowi wybranej nazwy
AT+ROLE=0	OK	Ustawienie modułu do pracy w trybie podrzędnym, dostępne ustawienia: 0 – Slave 1 – Master 2 – Slave Loop
AT+ADDR?	+ADDR<adres>OK	Odczyt unikalnego adresu modułu
AT+UART=38400,0,0	OK	Ustawienie parametrów portu szeregowego modułu: szybkości od 4800 do 1382400, bity stopu 0=1, 1=2, bity parzystości 0=bark, 1=Odd, 2=Even

Tabela 4. Ustawienie modułu w trybie nadrzędnym

Rozkaz AT	Odpowiedź HC-05	Opis
AT	OK	Potwierdzenie gotowości modułu do przyjmowania rozkazów AT
AT+ORGL	OK	Ustawienie parametrów predefiniowanych: ID:0 Working Role: Slave Baudrate: 38400bits/s or 9600bits/s Stop bit: 1 Parity: None Pair Code: 1234
AT+RMAAD	OK	Zerowanie listy sparowanych urządzeń
AT+NAME=Master	OK	Nadanie modułowi wybranej nazwy
AT+ROLE=1	OK	Ustawienie modułu do pracy w trybie nadrzędnym, dostępne ustawienia: 0 – Slave 1 – Master 2 – Slave Loop
AT+RESET	OK	Komenda resetu wysyłana po zmianie trybu pracy
AT+CMODE=1	OK	Zezwolenie na połączenie z urządzeniem o dowolnym adresie
AT+INQM=0,5,5	OK	Ustalenie kryterium wyszukiwania pracujących w pobliżu urządzeń: standardowy sposób wyszukiwania, wyszukiwanie zakończy odebranie sygnału z 5 urządzeń lub po 5 sekundach
AT+PSWD=1234	OK	Ustawienie kodu dostępu, powinien być taki jak w przeznaczonym do połączenia urządzeniu podrzędnym
AT+INIT	OK	Start profilu w trybie transmisji i odbioru
AT+INQ	+INQ<address> +INQ<address>...OK	Przesłanie adresów wszystkich wykrytych w pobliżu urządzeń bluetooth
AT+PAIR=<HC05 Slave address>,20	OK	Parowanie urządzenia o podanym adresie, maksymalny czas oczekiwania na sparowanie ok. 20s
AT+BIND=<HC05 Slave address>	OK	Określenie adresu wybranego urządzenia do automatycznego połączenia
AT+CMODE=0	OK	Automatyczne połączenie po resecie z urządzeniem o podanym w rozkazie BIND adresie
AT+LINK=<HC05 Slave address>	OK	Wymuszenie połączenia z urządzeniem o podanym adresie

terminalowy np. „Br@y+”. Po uruchomieniu na smartfonie wybranego programu terminalowego i wybraniu z listy dostępnych urządzeń modułu, powinno nastąpić samoczynne ustanowienie połączenia, co dioda LED zamontowana na HC-05 zasygnalizuje zmianą rytmu migotania. Od tego momentu wypisane na smartfonie w oknie terminala teksty powinny pojawiać się w oknie terminala uruchomionego na komputerze, do którego jest przyłączony moduł. Trzeba tylko pamiętać by szybkość transmisji portu szeregowego modułu była zgodna z szybkością ustawioną na terminalu komputera.

Ustawienie modułu HC-05 w trybie nadrzędnym (master)

Przykładowa sekwencja rozkazów ustawia moduł do pracy w trybie nadrzędnym. Dodatkowo moduł będzie się zawsze łączył z określonym modułem HC-05 pracującym w trybie podrzędnym. Użyte rozkazy wraz z opisem wymieniono w tabeli 4.

Każdy rozkaz musi być zakończony znakami CR LF (dziesiętnie 13 i 10). Poniżej przykładowa sekwencja rozkazów i odpowiedzi modułu wyświetlana na pulpicie terminalu:

```
AT
OK
AT+ORGL
OK
AT+RMAAD
OK
AT+NAME=Master
OK
AT+ROLE=1
OK
AT+RESET
OK
Ponowne wprowadzenie modułu w tryb AT, od tego momentu przycisk na module HC-05 powinien być naciśnięty bez przerwy.
AT+CMODE=1
OK
```

```
AT+INQM=0,5,5
OK
AT+PSWD=1234
OK
AT+INIT
OK
AT+INQ
+INQ:5500:27:1758A5,5E020C,7FFF □ wykryty
ty telefon
+INQ:2015:3:275538,0,7FFF □ wykryty
pierwszy moduł HC-05
+INQ:2015:3:275580,0,7FFF □ wykryty
drugi moduł HC-05
OK
AT+PAIR=2015,3,275538,20 □ parowanie
z pierwszym modułem HC-05
OK
AT+BIND=2015,3,275538
OK
AT+CMODE=0 □ po resecie automatyczne
łączenie z pierwszym modułem
OK
AT+LINK=2015,3,275538 □ wymuszenie
połączenia z urządzeniem o podanym adresie
OK
```

Testowanie transmisji pomiędzy modułami HC-05

Jeżeli dwa moduły będą ustawione w taki sposób jak to wcześniej opisano, po włączeniu zasilania powinno nastąpić automatyczne połączenie pomiędzy modułem pracującym w trybie nadrzędnym (masterem) a modułem pracującym w trybie podrzędnym (slave). Nawiązanie połączenia będzie sygnalizowane migotaniem diod LED obu modułów w sekwencji błysków: 2 krótkie – przerwa -2 krótkie ... Po nawiązaniu połączenia będzie możliwa jednoczesna dwustronna transmisja danych podawanych na wejścia UART obu modułów.

Ryszard Szymaniak, EP