

Protokół komunikacyjny		
Przedmiot	Mikroprocesory	 POLITECHNIKA BYDGOSKA Wydział Telekomunikacji, Informatyki i Elektrotechniki
Student	Mikołaj Kołodziejski	
Indeks	121192	

1. Protokół komunikacyjny

1.1. Wstęp

Komunikacja pomiędzy użytkownikiem, a urządzeniem będzie odbywać się poprzez program terminal, za pomocą którego użytkownik będzie mógł wysłać odpowiednie komunikaty do urządzenia STM32.

Protokół będzie służył do odbierania poleceń użytkownika, zwracał będzie informacje zwrotne (ramkę), takie jak, czy komenda zawarta w ramce będzie prawidłowa.

Użytkownik będzie wprowadzał dane w terminalu znaki, które będą odbierane przez urządzenie, które będzie przechowywać je w formacie ASCII. Po prawidłowym odebraniu ramki i rozpoznaniu komendy, urządzenie wyśle ramkę zwrotną z zamienionymi adresami nadawcy i odbiorcy oraz ze zmienionymi danymi (w otrzymanej ramce dane to komenda, a w zwracanej, informacje o rozpoznaniu lub nierozpoznaniu komendy).

Obsługa błędów:

- Jeżeli zostanie wysłanych kilka znaków rozpoczęcia ramki pod rząd, to ramka rozpocznie się od ostatniego znaku w powtórzeniu, np. :::::, to ramka rozpocznie się od piątego znaku ':', a reszta zostanie zignorowana.
- Jeżeli zostanie wysłanych kilka znaków końca ramki pod rząd, to ramka zakończy się na pierwszym takim znaku w powtórzeniu, np. ;;;;, to ramka zakończy się na pierwszym znaku ';', a reszta zostanie zignorowana.
- Jeżeli ramka zostanie wprowadzona prawidłowo (z prawidłowym adresem odbiorcy i będzie miała prawidłową konstrukcję), a komenda nie zostanie rozpoznana, zostanie zwrócona ramka informująca o wpisaniu błędnej komendy.
- W sytuacji, gdy do urządzenia zostanie wysłana prawidłowa ramka, w środku której znajdują się znaki rozpoczęcia lub zakończenia ramki (np. w danych lub adresach), nie zostanie wysłana informacja zwrotna, co trzeba przyjąć jako błędną ramkę.
- Jeżeli użytkownik poda dłuższą długość komendy, niż wpisana podana komenda będzie posiadała w rzeczywistości oraz gdy jednocześnie zostaną

podane dane w ilości większej, niż długość komendy, wtedy urządzenie wykryje błąd i zwróci komunikat o błędnej długości komendy.

1.2. RAMKA PROTOKOŁU

Początek Ramki	Nadawca	Odbiorca	Długość Komendy	Dane	Koniec Ramki
:	000	001	000	[znaki ASCII]	;
0x3A	wszystkie znaki poza 0x3A i 0x3B	wszystkie znaki poza 0x3A i 0x3B	znaki od 0x30 do 0x39	wszystkie znaki poza 0x3A i 0x3B	0x3B
1	3	3	3	0-256	1

- **Maksymalna długość komendy:** Wynosi 256 bajtów (znaków). Jest to maksymalna długość pola "Dane".
- **Minimalna długość komendy:** Najkrótsza komenda musi zawierać co najmniej jeden znak. Zatem, minimalna niezerowa długość komendy, to 001. Ilość znaków w danych (czyli w polu "Dane") nie może wynosić 0, jeżeli długość komendy jest różna od 0.
- **Minimalna długość ramki:** Minimalna długość całej ramki (bez znaków początku i końca) wynosi 8 znaków ASCII: 3 (Nadawca) + 3 (Odbiorca) + 3 (Długość) = 9. Znak początku i końca to dodatkowe dwa znaki. *Jednakże*, jeśli pole "Długość Komendy" ma wartość "000", cała ramka jest traktowana jako *pusta* i jest ignorowana (z wyjątkiem wysłania komunikatu o błędzie).
- **Adres urządzenia (płytki STM32):** Na potrzeby tego projektu adres płytki STM32 to "MTW" (zamiast numerycznego "001" czy innego). To ułatwia identyfikację urządzenia. Adres komputera PC to "000".
- **Budowa komendy:** Komenda (zawarta w polu "Dane") składa się z *nazwy komendy*. W tej uproszczonej wersji protokołu, komendy *nie mają* dodatkowych parametrów w polu "Dane". Przykładowe komendy to: SNAKE, PULSE, BLINK, SET. Komenda SET ma parametry oddzielone przecinkami: SET,R,G,B. Gdzie R,G,B to trzypozycyjne wartości od 000 do 255.

Dozwolone znaki:

- **Dane:** Dowolne znaki ASCII z *wyjątkiem* znaków początku ramki (':', 0x3A) i końca ramki (';', 0x3B).
- **Nadawca i Odbiorca:** Dowolne znaki ASCII z *wyjątkiem* znaków początku ramki (':', 0x3A) i końca ramki (';', 0x3B). Zalecane jest używanie adresów numerycznych ("000" - "999") lub symbolicznych jak "MTW".
- **Długość Komendy:** Wyłącznie cyfry ASCII ('0' - '9', 0x30 - 0x39).

1.3. Tabela Rozkazów

Komenda	Długość w ramce [ASCII]	Opis	Przykład Ramki	Przykład ramki zwrotnej
SNAKE	005	Uruchamia efekt "węża" świetlnego z domyślnymi parametrami (kolor, prędkość).	:000MTW005SNAKE;	:MTW000002OK; lub :MTW000005ERROR; w przypadku błędu
PULSE	005	Uruchamia efekt pulsowania światła z domyślnymi parametrami.	:000MTW005PULSE;	:MTW000002OK; lub :MTW000005ERROR;
BLINK	005	Uruchamia efekt migania diod LED z domyślnymi parametrami.	:000MTW005BLINK;	:MTW000002OK; lub :MTW000005ERROR
SET	Zmienna	Ustawia statyczny kolor dla wszystkich diod LED. Przyjmuje parametry R, G, B (czerwony, zielony, niebieski) jako trzy 3-znakowe liczby ASCII (000-255).	:000MTW012SET,255,000,000; (czerwony)	:MTW000002OK; lub :MTW000005ERROR; (np. ERR_RGB dla złych wartości RGB)
STATUS	006	Żądanie statusu urządzenia. STM32 może zwrócić informacje o aktualnym trybie, kolorze, itp. (szczegóły odpowiedzi do zdefiniowania).	:000MTW006STATUS ;	:MTW000xxx...; (gdzie xxx... to dane statusu, np. 008RUN,SNAKE
RESET	005	Resetuje urządzenie (STM32). Może wyłączyć wszystkie efekty i przywrócić domyślne ustawienia.	:000MTW005RESET;	:MTW000002OK;
PING	004	Sprawdza, czy urządzenie jest dostępne.	:000MTW004PING;	:MTW000004PONG;

1.4. Tabela błędów

Kod Błędu	Długość w ramce [ASCII]	Opis	Przyczyna
OK	002	Operacja zakończona sukcesem.	Komenda została poprawnie odebrana, zinterpretowana i wykonana.
ERROR	005	Ogólny błąd. Używany, gdy wystąpił błąd, ale nie można go zakwalifikować do bardziej szczegółowej kategorii. Zaleca się unikanie tego kodu na rzecz bardziej precyzyjnych kodów błędów.	Niesprecyzowany błąd.
ERR_CMD	007	Nieznana komenda. STM32 nie rozpoznało nazwy komendy przesłanej w polu "Dane".	W polu "Dane" ramki z komendą znajduje się ciąg znaków, który nie odpowiada żadnej zdefiniowanej nazwie komendy (np. literówka, nieprawidłowa komenda).
ERR_ARG	007	Nieprawidłowe argumenty komendy. Używane, gdy komenda (np. SET) przyjmuje argumenty, a podane argumenty są nieprawidłowe (zła liczba argumentów, nieprawidłowy format, brakujące argumenty).	Dla komendy SET: mniej lub więcej niż 3 argumenty (R, G, B), argumenty nie są oddzielone przecinkami, argumenty nie są 3-znakowymi liczbami ASCII.
ERR_RGB	007	Nieprawidłowe wartości RGB. Specyficzny błąd dla komendy SET, gdy wartości R, G lub B są poza zakresem 0-255.	Wartość R, G lub B w komendzie SET jest mniejsza niż 0 lub większa niż 255, lub nie jest liczbą.
ERR_LEN	007	Nieprawidłowa długość komendy. Wartość w polu "Długość Komendy" nie zgadza się z faktyczną długością pola "Dane" lub wartość w polu długości jest większa niż 256.	Wartość w polu "Długość Komendy" jest mniejsza lub większa niż liczba znaków w polu "Dane", w polu długości podano więcej niż 3 znaki, podana wartość jest większa niż 256.
ERR_EMP	007	Pusta ramka	Wysłano pustą ramkę z wartością 000 w polu "Długość Komendy".

1.4.1. Działanie ramek zwrotnych

Ramki zwrotne to komendy które będą wysyłane z odbiory do nadawcy (STM32 → PC). Są one według mnie niezbędne w każdym protokole komunikacji ze względu na potwierdzenie czy ramka została poprawnie odebrana.

1.4.2. Konstrukcja ramki nadawczej (PC→ STM32)

To jest proces, który musi wykonać oprogramowanie na PC *przed* wysłaniem komendy do STM32.

1. **Wybierz Komendę:** Zdecyduj, którą komendę chcesz wysłać (np. SNAKE, SET, STATUS, RESET).
2. **Przygotuj Pole "Dane":**
 - **Dla komend bez parametrów (SNAKE, PULSE, BLINK, STATUS, RESET, PING):** Pole "Dane" zawiera *tylko* nazwę komendy (np. "SNAKE").
 - **Dla komendy SET:** Pole "Dane" zawiera nazwę komendy i parametry, oddzielone przecinkami: SET,R,G,B. Wartości R, G, B muszą być 3-znakowymi liczbami ASCII (np. "255", "000", "128").
Przykład: "SET,255,000,100"
 - **Dla komendy PONG:** PONG.
3. **Oblicz Długość:** Oblicz *liczbę znaków* w polu "Dane". To będzie wartość pola "Długość Komendy".
 - **Przykład (SNAKE):** Długość = 5 (bo "SNAKE" ma 5 znaków).
 - **Przykład (SET,255,000,100):** Długość = 12 (bo "SET,255,000,100" ma 12 znaków).
4. **Sformatuj Pole "Długość Komendy":** Zapisz obliczoną długość jako 3-znakową liczbę ASCII, z zerami wiodącymi, jeśli trzeba.
 - **Przykład (Długość = 5):** "005"
 - **Przykład (Długość = 12):** "012"
 - **Przykład (Długość = 256):** "256"
5. **Złóż Ramkę:** Połącz wszystkie pola w jeden ciąg znaków, w odpowiedniej kolejności:

Ramka = ":" + Nadawca + Odbiorca + Długość_Komendy + Dane + ";"

- **Nadawca:** "000" (zakładamy, że PC to "000").
- **Odbiorca:** "MTW" (adres STM32).

6. **Wyślij Ramkę:** Wyślij utworzony ciąg znaków przez port szeregowy (UART) do STM32.

Przykład Ramki zwrotnej:

:000MTW005BLINK;

1.4.3. Konstrukcja ramki nadawczej (STM32→ PC)

To jest proces, który wykonuje STM32 po odebraniu i przetworzeniu komendy od PC.

1. **Określ Wynik:** Po przetworzeniu komendy, ustal, czy operacja się powiodła (np. "OK"), czy wystąpił błąd (np. "ERR_CMD", "ERR_RGB").
2. **Przygotuj Pole "Dane":** Wpisz odpowiedni kod wyniku do pola "Dane" (np. "OK", "ERR_CMD").
3. **Oblicz Długość:** Oblicz długość pola "Dane" (np. "OK" ma długość 2, "ERR_CMD" ma długość 7).
4. **Sformatuj Pole "Długość Komendy":** Zapisz długość jako 3-znakową liczbę ASCII (np. "002", "007").
5. **Złóż Ramkę:** Połącz wszystkie pola w jeden ciąg znaków, w odpowiedniej kolejności:

Ramka = ":" + Nadawca + Odbiorca + Długość_Komendy + Dane + ";"

- **Nadawca:** "000" (zakładamy, że PC to "000").
- **Odbiorca:** "MTW" (adres STM32).

6. **Wyślij Ramkę:** Wyślij utworzony ciąg znaków przez port szeregowy (UART) do STM32.

Przykład Ramki zwrotnej:

:MTW000002OK; lub :MTW000005ERROR;