

Protokół komunikacji z procesorem będącym interfejsem z pamięcią SRAM.

Komunikacja po UART, wysyłanie odpowiednich sekwencji bajtów, które będą tworzyły całe ramki komend.

Stałe komendy komunikacji:

0x00 – START. Pierwszy bajt, który wybudzi odbiorcę wiadomości i przygotowuje go na resztę informacji

0x01 – ACK. Potwierdzenie otrzymania prawidłowej wiadomości (pasuje jakieś CRC na końcu liczyć żeby potwierdzać poprzednie bajty)

0x02 – NACK. Informacja że nie otrzymano całej ramki lub checksuma się nie zgadza

Wymagane komendy do testów:

1. Ustawienie pamięci w różne stany (READ, WRITE, STANDBY, OUTPUT DISABLE)
2. Sprawdzenie w jakim stanie jest pamięć
3. Zapis danej pod adres
4. Zapis danej na całą pamięć
5. Odczyt danej spod adresu
6. Odczyt całej pamięci po kolei
7. Sterowanie zasilaniem pamięci

Proponowane rozwiązanie:

Rozpocząć wysyłanie od 0x00, następny bajt będzie oznaczał funkcje np. 0x10 to zapis, a 0x11 odczyt. Dla obu kolejne 2 bajty będą oznaczały adres, a 3 bajt to dane.

Przykład komunikacji

1. PC → uC 0x00 | 0x10 | 0x00 | 0x05 | 0xff | (coś można jakieś opcje) | Checksum
([Rozpoczynam komunikację z PC, chcę zapisać bajt 0xff w miejscu 0x0005])
2. uC → PC 0x01 (ACK)
3. uC → PC 0x00 | (potwierdzam że zapisało) | ... (coś jeszcze?) ... | Checksum
4. PC → uC 0x01 (ACK)

Checksum to np. xor każdego bajtu po kolei, albo najprościej suma wszystkich bajtów