

Systemy wbudowane - wykład 7

Przemek Błaśkiewicz

4 kwietnia 2019

- UART

- UART
 - ramka danych (STOP, START, DATA, PARITY)

- UART
 - ramka danych (STOP, START, DATA, PARITY)
 - parametry (bod, tryby transmisji, szybkość, (a)synchroniczność)

- UART
 - ramka danych (STOP, START, DATA, PARITY)
 - parametry (bod, tryby transmisji, szybkość, (a)synchroniczność)
 - RTS/CTS, USART, DUART, OCTART

- UART
 - ramka danych (STOP, START, DATA, PARITY)
 - parametry (bod, tryby transmisji, szybkość, (a)synchroniczność)
 - RTS/CTS, USART, DUART, OCTART
- RS-232

- UART
 - ramka danych (STOP, START, DATA, PARITY)
 - parametry (bod, tryby transmisji, szybkość, (a)synchroniczność)
 - RTS/CTS, USART, DUART, OCTART
- RS-232
 - rodzina MAX232

- UART
 - ramka danych (STOP, START, DATA, PARITY)
 - parametry (bod, tryby transmisji, szybkość, (a)synchroniczność)
 - RTS/CTS, USART, DUART, OCTART
- RS-232
 - rodzina MAX232
 - PL2302

- SCLK – linia zegarowa

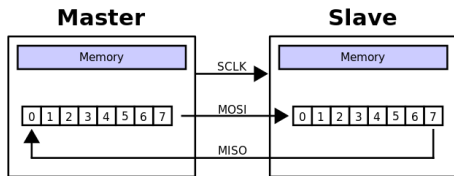
- SCLK – linia zegarowa
- MOSI – master output, slave input

- SCLK – linia zegarowa
- MOSI – master output, slave input
- MISO – master input, slave output

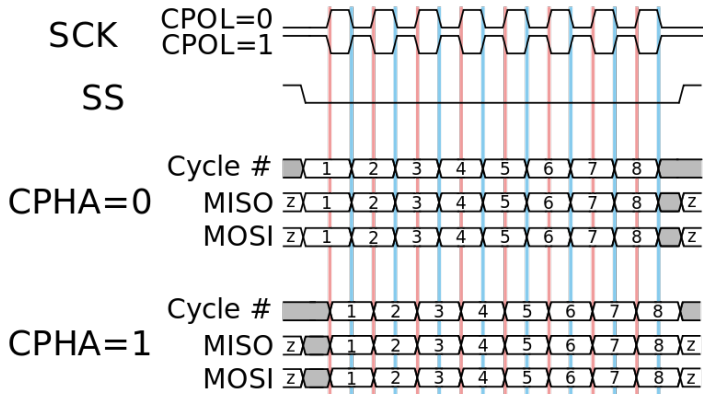
- SCLK – linia zegarowa
- MOSI – master output, slave input
- MISO – master input, slave output
- SS – slave select – wybór jednoski slave

SPI - kabelkologia sygnalizowana

- SCLK – linia zegarowa
- MOSI – master output, slave input
- MISO – master input, slave output
- SS – slave select – wybór jednoski slave



SPI - czasowość



CPOL=0 → podstawa czasu 0

- CPHA=0 → dane zatrzymywane na rosnącym zboczach
- CPHA=1 → dane zatrzymywane na opadającym zboczach

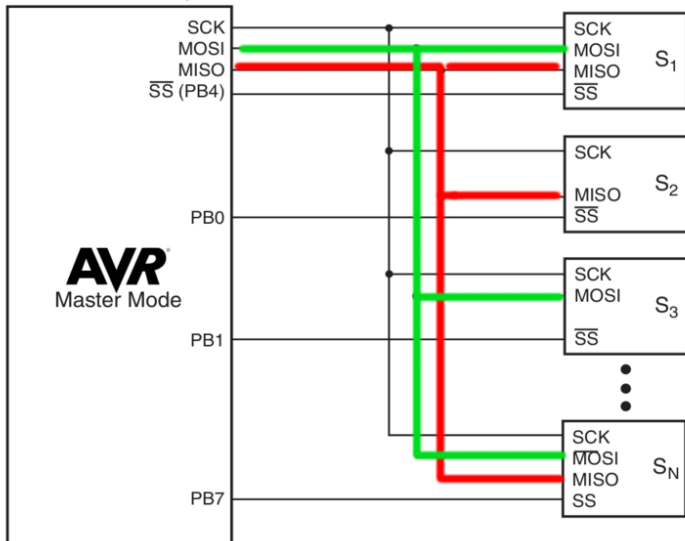
CPOL=1 → podstawa czasu 1

- CPHA=0 → dane zatrzymywane na opadającym zboczach
- CPHA=1 → dane zatrzymywane na rosnącym zboczach

Na przeciwnym zboczach następuje propagacja danych (shift w rejestrach).

SPI - połączenie wielu urządzeń slave

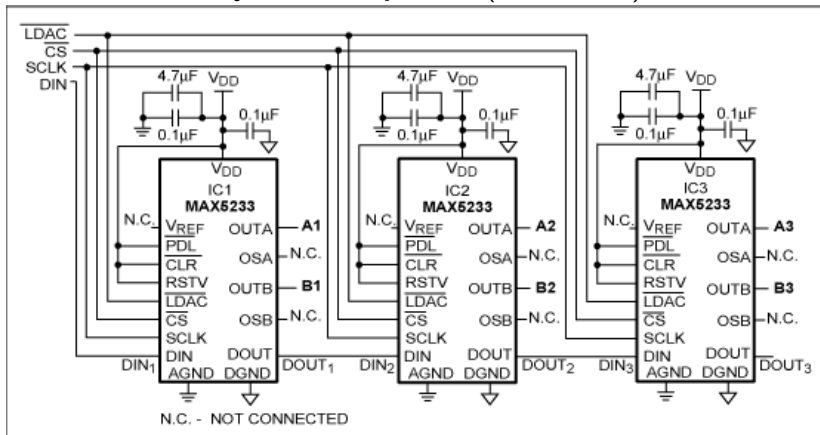
Standardowe połączenie: 1 master steruje każdym slave osobno:





SPI - połączenie wielu urządzeń slave

Połączenie "daisy-chain" (stokrotka...)



- wariacja: mSPI - wszystkie urządzenia podłączone 4 liniami, slave mają adres;

- wariacja: mSPI - wszystkie urządzenia podłączone 4 liniami, slave mają adres;
- tylko jedna dedykowana linia SS, reszta wspólna;

- wariacja: mSPI - wszystkie urządzenia podłączone 4 liniami, slave mają adres;
- tylko jedna dedykowana linia SS, reszta wspólna;
- linie jednokierunkowe – można odseparować galwanicznie (ACSL-6400);

- wariacja: mSPI - wszystkie urządzenia podłączone 4 liniami, slave mają adres;
- tylko jedna dedykowana linia SS, reszta wspólna;
- linie jednokierunkowe – można odseparować galwanicznie (ACSL-6400);
- nie ma ograniczonego rozmiaru ramki

- wariacja: mSPI - wszystkie urządzenia podłączone 4 liniami, slave mają adres;
- tylko jedna dedykowana linia SS, reszta wspólna;
- linie jednokierunkowe – można odseparować galwanicznie (ACSL-6400);
- nie ma ograniczonego rozmiaru ramki
- nie trzeba konwertować poziomów ani synchronizować urządzeń

- wariacja: mSPI - wszystkie urządzenia podłączone 4 liniami, slave mają adres;
- tylko jedna dedykowana linia SS, reszta wspólna;
- linie jednokierunkowe – można odseparować galwanicznie (ACSL-6400);
- nie ma ograniczonego rozmiaru ramki
- nie trzeba konwertować poziomów ani synchronizować urządzeń
- brak kontroli komunikacji po stronie slave

1-wire

- dwukierunkowy protokół pół-dupleksowy



1-wire

- dwukierunkowy protokół pół-dupleksowy
- dwie linie: DATA, GND



1-wire

- dwukierunkowy protokół pół-duplexowy
- dwie linie: DATA, GND
- każde urządzenie ma zakodowany unikalny identyfikator



1-wire

- dwukierunkowy protokół pół-dupleksowy
- dwie linie: DATA, GND
- każde urządzenie ma zakodowany unikalny identyfikator
- struktura master-slave



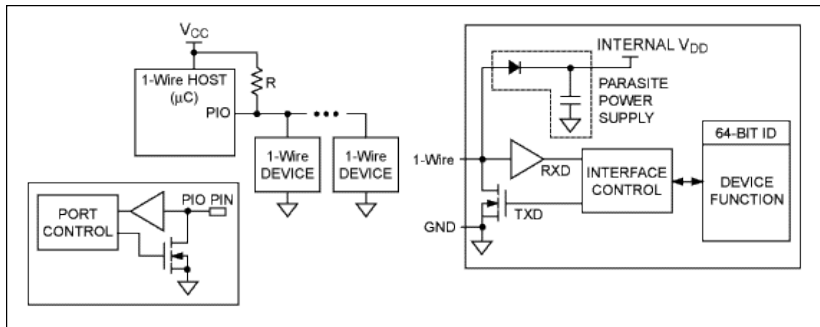
1-wire

- dwukierunkowy protokół pół-dupleksowy
- dwie linie: DATA, GND
- każde urządzenie ma zakodowany unikalny identyfikator
- struktura master-slave
- prędkości: 15.4kbps, 125kbps (overdrive)



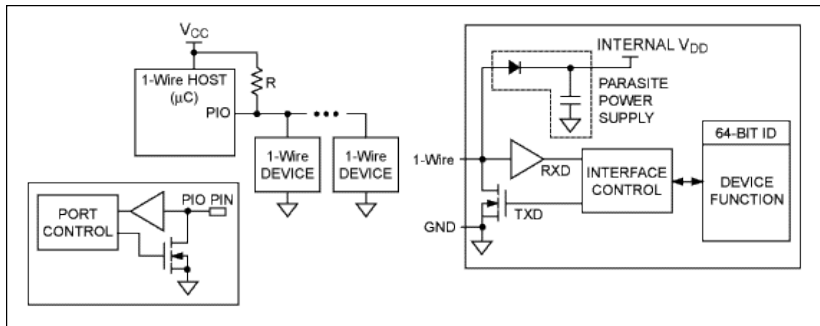
1-wire szczegóły

- master rozpoczyna i kontroluje komunikację (collision avoidance)



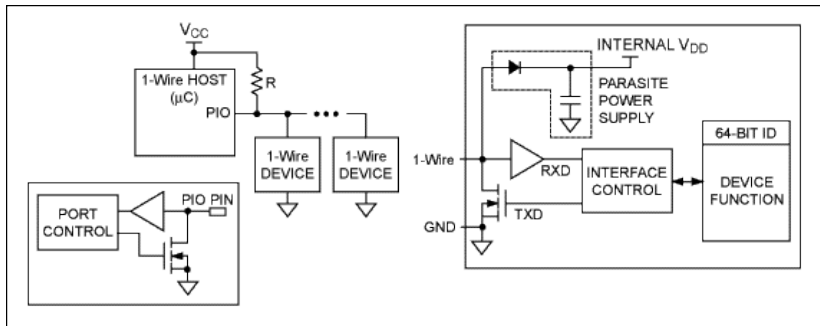
1-wire szczegóły

- master rozpoczyna i kontroluje komunikację (collision avoidance)
- pompa ładunkowa + kondensator zapewnia zasilanie



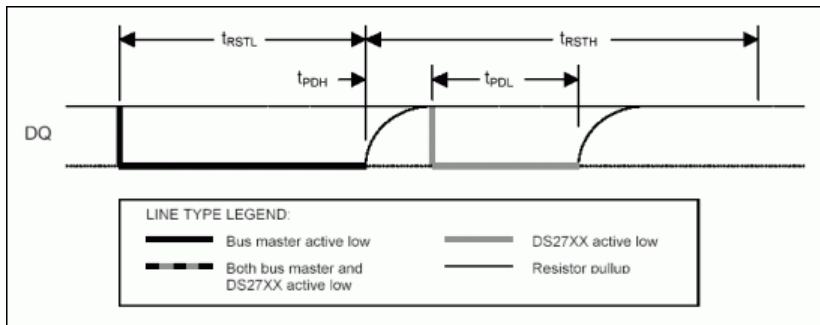
1-wire szczegóły

- master rozpoczyna i kontroluje komunikację (collision avoidance)
- pompa ładunkowa + kondensator zapewnia zasilanie
- master nadaje "sygnał" CLK



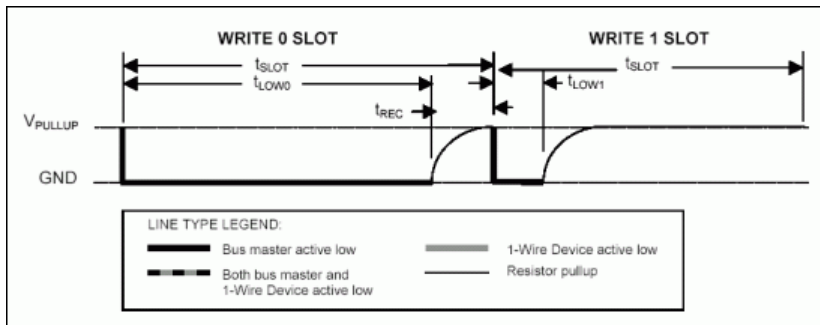
1-wire szczegóły

- master rozpoczyna i kontroluje komunikację (collision avoidance)
- pompa ładunkowa + kondensator zapewnia zasilanie
- master nadaje "sygnał" CLK
- binarna komunikacja w slotach czasowych



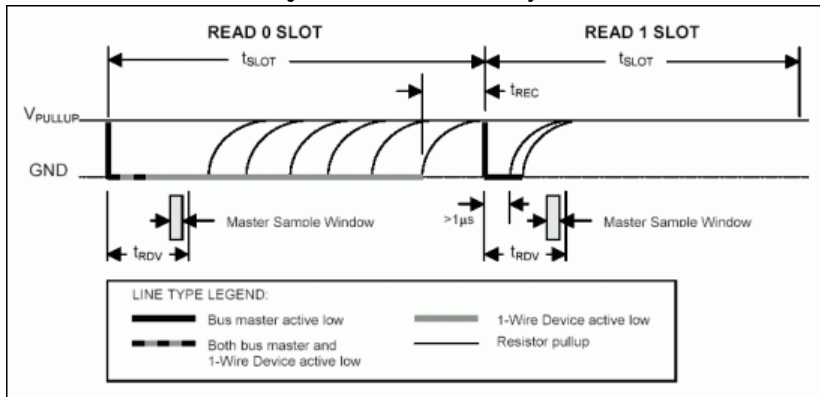
1-wire szczegóły

- master rozpoczyna i kontroluje komunikację (collision avoidance)
- pompa ładunkowa + kondensator zapewnia zasilanie
- master nadaje "sygnał" CLK
- binarna komunikacja w slotach czasowych

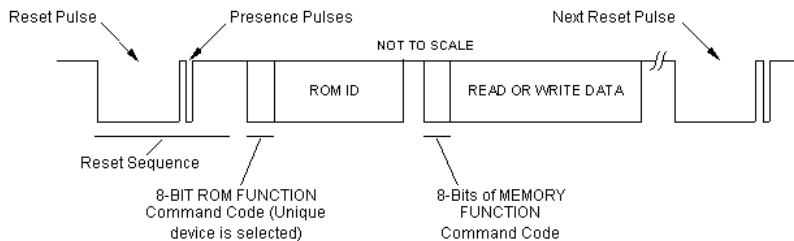


1-wire szczegóły

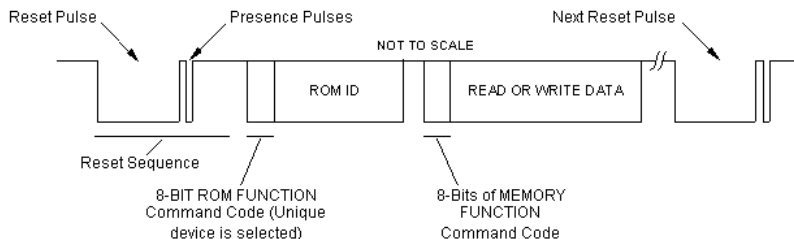
- master rozpoczyna i kontroluje komunikację (collision avoidance)
- pompa ładunkowa + kondensator zapewnia zasilanie
- master nadaje "sygnał" CLK
- binarna komunikacja w slotach czasowych



1-Wire - komunikacija

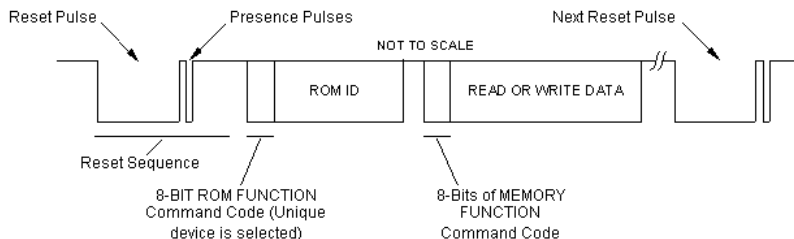


1-Wire - komunikacja



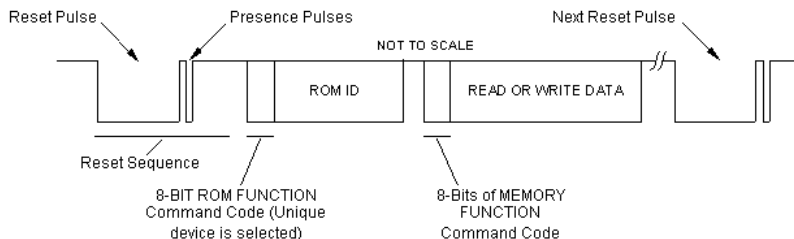
- ROM-ID : 8 bitów określających typ urządzenia + 48 bitów identyfikatora + 8 bitów 8bit CRC

1-Wire - komunikacja



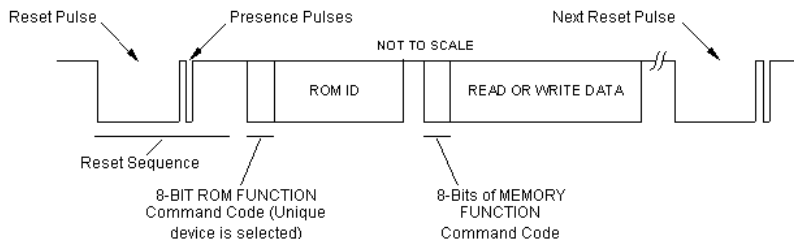
- ROM-ID : 8 bitów określających typ urządzenia + 48 bitów identyfikatora + 8 bitów 8bit CRC
- ROM-CMD:

1-Wire - komunikacja



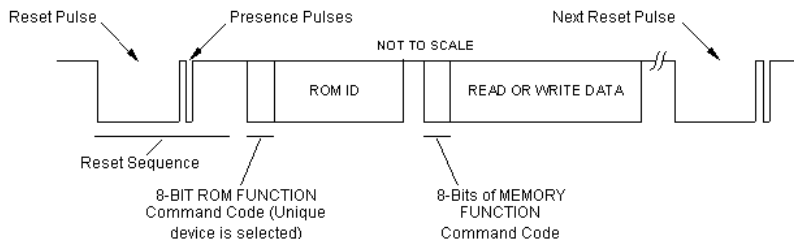
- ROM-ID : 8 bitów określających typ urządzenia + 48 bitów identyfikatora + 8 bitów 8bit CRC
- ROM-CMD:
 - *skip ROM* (dla pojedynczego urządzenia)

1-Wire - komunikacja



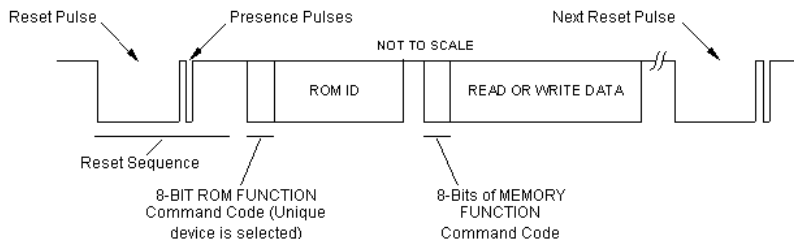
- ROM-ID : 8 bitów określających typ urządzenia + 48 bitów identyfikatora + 8 bitów 8bit CRC
- ROM-CMD:
 - *skip ROM* (dla pojedynczego urządzenia)
 - READ ROM (przeczytaj ID wybranego)

1-Wire - komunikacja



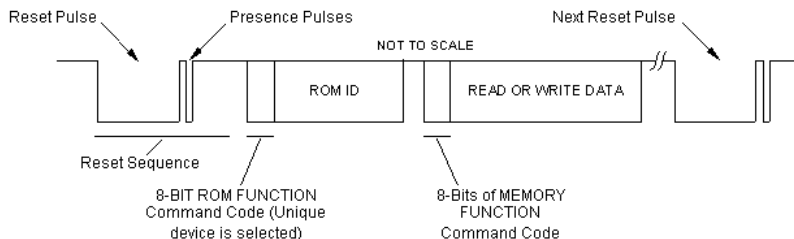
- ROM-ID : 8 bitów określających typ urządzenia + 48 bitów identyfikatora + 8 bitów 8bit CRC
- ROM-CMD:
 - *skip ROM* (dla pojedynczego urządzenia)
 - READ ROM (przeczytaj ID wybranego)
 - match ROM (kolejne dane tylko dla tego)

1-Wire - komunikacja



- ROM-ID : 8 bitów określających typ urządzenia + 48 bitów identyfikatora + 8 bitów 8bit CRC
- ROM-CMD:
 - *skip ROM* (dla pojedynczego urządzenia)
 - READ ROM (przeczytaj ID wybranego)
 - match ROM (kolejne dane tylko dla tego)
 - resume ROM (kontynuacja), overdrive-skip ROM, search ROM(*)

1-Wire - komunikacja



- ROM-ID : 8 bitów określających typ urządzenia + 48 bitów identyfikatora + 8 bitów 8bit CRC
- ROM-CMD:
 - *skip ROM* (dla pojedynczego urządzenia)
 - READ ROM (przeczytaj ID wybranego)
 - match ROM (kolejne dane tylko dla tego)
 - resume ROM (kontynuacja), overdrive-skip ROM, search ROM(*)
- MEMORY-FUNC: zależne od urządzenia (np. czytanie konkretnych danych, konfiguracja)

Inne protokoły... I2C, w następnym odcinku

Rzeczy do zapamiętania

- rodzaje transmisji, Bd,
- UART: tryby, działanie, ramka danych
- SPI: timing, sygnały, sposoby łączenia
- 1-Wire: linie, sygnały