



Politechnika Łódzka

Instytut Elektroniki

Programowanie Mikrokontrolerów

Magistrala SPI

mgr inż. Paweł Poryzała

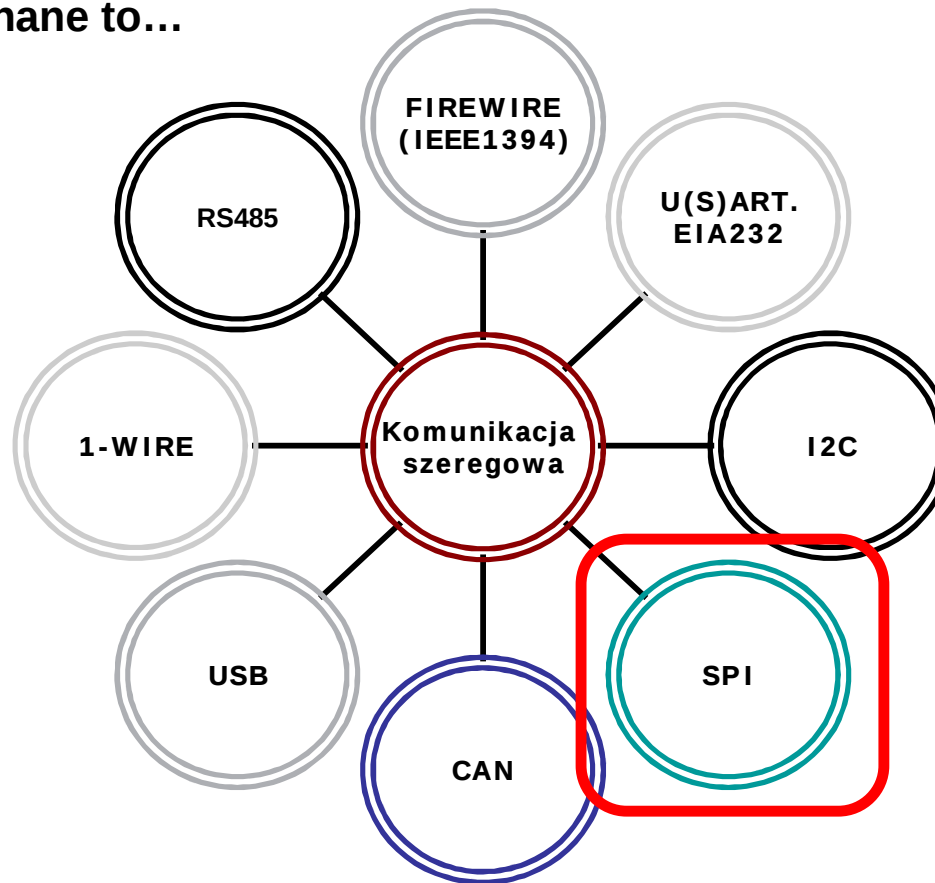
Zakład Elektroniki Medycznej



Komunikacja szeregowa

- Jakie znamy typy komunikacji szeregowej?

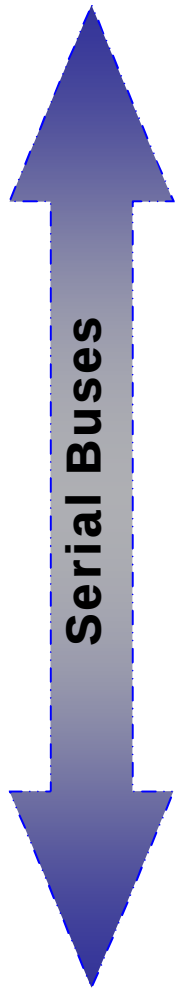
Prawdopodobnie najbardziej znane to...



Zagadnienia

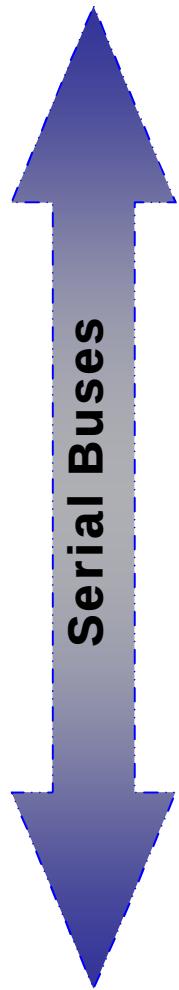
- Wprowadzenie,
- Główne cechy, zalety i wady magistrali SPI,
- Przykłady zastosowań,
- Przykładowa konfiguracja, tryby pracy magistrali,
- Dane na magistrali SPI,
- Podsumowanie,
- Przykład komunikacji – ATmega32.

SPI



- Serial Peripheral Interface:
 - Jest to „3+n”-przewodowy, szeregowy, synchroniczny interfejs typu full-duplex zawierający linie:
 - SCLK: *Serial Clock*,
 - MOSI: *Master Out Slave In*, ew. SIMO, dane z urządzenia Master do Slave,
 - MISO: *Master In Slave Out*, ew. SOMI, dane z urządzenia Slave do Master,
 - SS: *Slave Select*,
 - Opracowany przez firmę Motorola,
 - Używany do połączeń pomiędzy uC i szybkimi, szczególnie strumieniowymi, peryferiami,

SPI



- Rejestry przesuwne do szeregowego wysyłania oraz odbioru bitów (magistrala zorientowana bajtowo, lecz istnieją wyjątki),
- Komunikacja typu *Master* – *Slave*, tylko jedno urządzenie *Slave* aktywne w danej chwili,
- *Master* nadzoruje transmisję, zapewnia sygnał zegarowy SCLK,
- Różne prędkości transmisji w zależności od zegara SCLK – typowo w zakresie 1MHz – 70 MHz.

Alternatywne nazewnictwo linii SPI

- SCLK:
 - SCK, CLK — Serial Clock (output from master)
- MOSI/SIMO:
 - SDI, DI, SI — Serial Data In
- MISO/SOMI:
 - SDO, DO, SO — Serial Data Out
- SS:
 - nCS, CS, nSS, STE — Chip Select, Slave Transmit Enable (active low; output from master)

Cechy

- **Zalety**

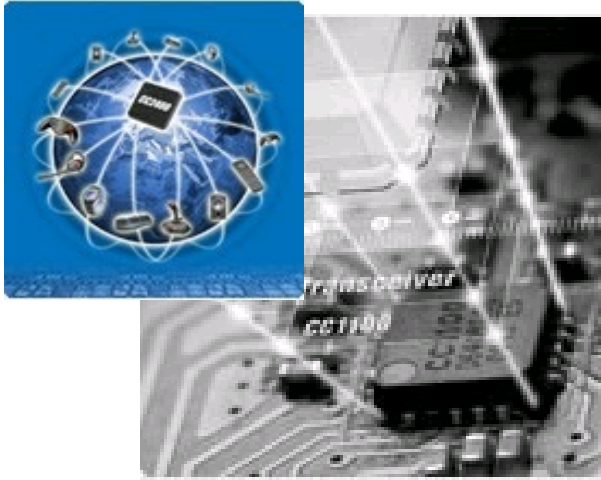
- Komunikacja ful-duplex
- Szybkość transmisji znacznie wyższa niż w I2C, 1-Wire lub podobnych
- Pełna dowolność definicji protokołu wymiany danych
 - Brak ograniczeń do 8-bitów danych,
 - Brak ograniczeń związanych z protokołem (adresowanie, formaty wymiany danych), brak konieczności definiowania adresów,
- Prosty hardware, prosta emulacja poprzez „bit-banging”:
 - Brak złożonych protokołów arbitrażu,
 - Urządzenie *Slave* synchronizują transmisję do sygnału SCLK – mogą być proste i tanie,
- Znacznie mniej połączeń niż w trybie transmisji równoległej

Cechy

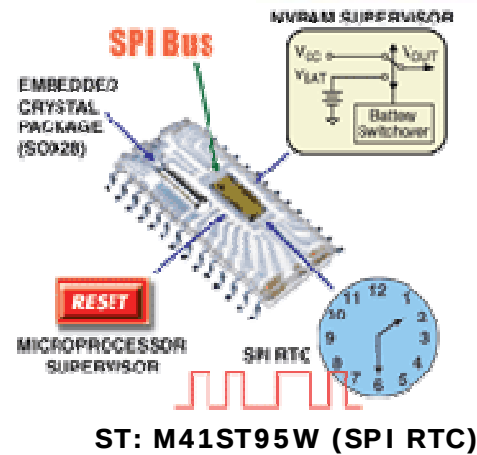
- **Wady**

- Wymaga użycia większej ilości wyprowadzeń niż I2C czy 1-Wire,
- Brak sprzętowej kontroli przepływu danych,
- Brak potwierdzeń ze strony urządzenia *Slave* – *Master* może wysyłać „w powietrze” nic o tym nie wiedząc
- Tylko jedno urządzenie *Master* na magistrali,
- Komunikacja na niewielkiej odległości – w ramach PCB bądź urządzenia.

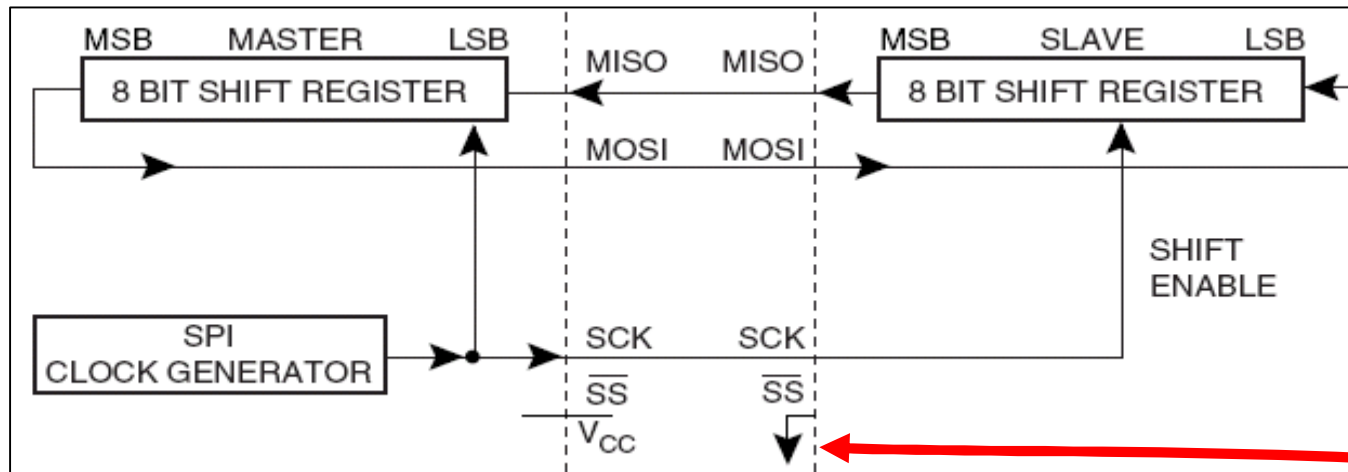
Zastosowania



Canon EF Lenses



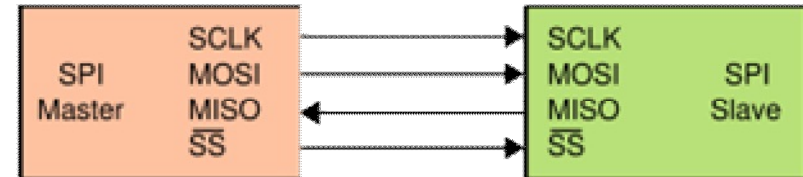
Przykładowa konfiguracja



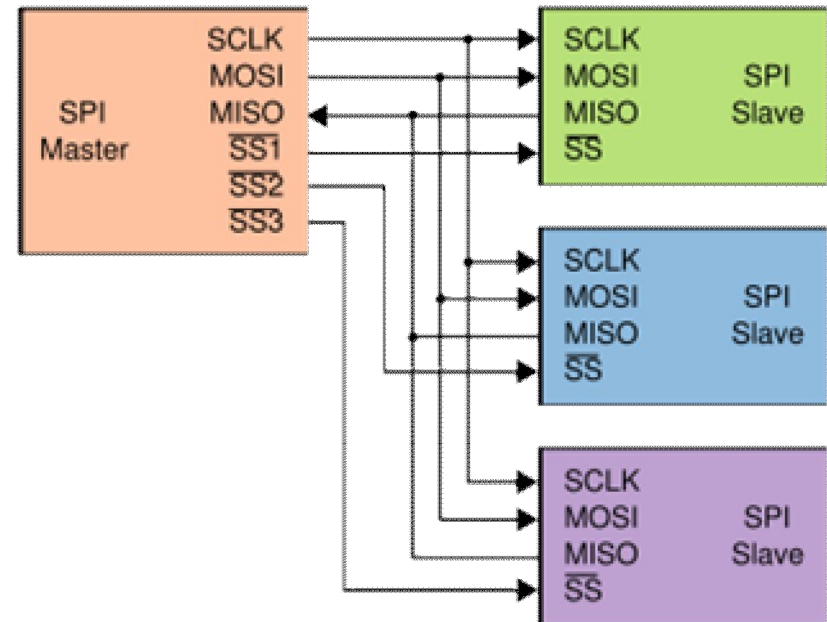
- Konfiguracja pojedynczy *Master* – pojedynczy *Slave*:
 - Linia nSS powinna być połączona, choćby ze względu na możliwość przyszłej rozbudowy magistrali.

Możliwe konfiguracje magistrali

Single Master – Single Slave

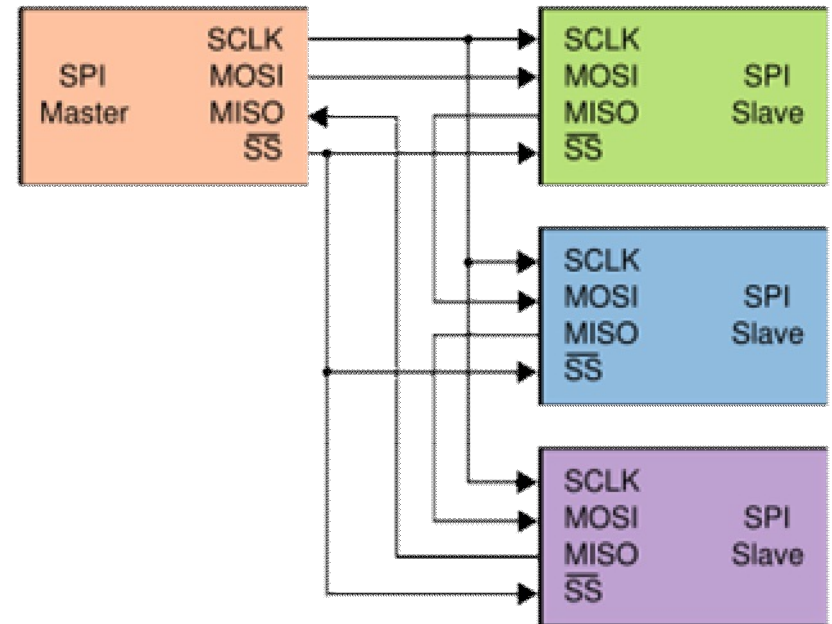


Single Master – Multiple Slave
(independent)

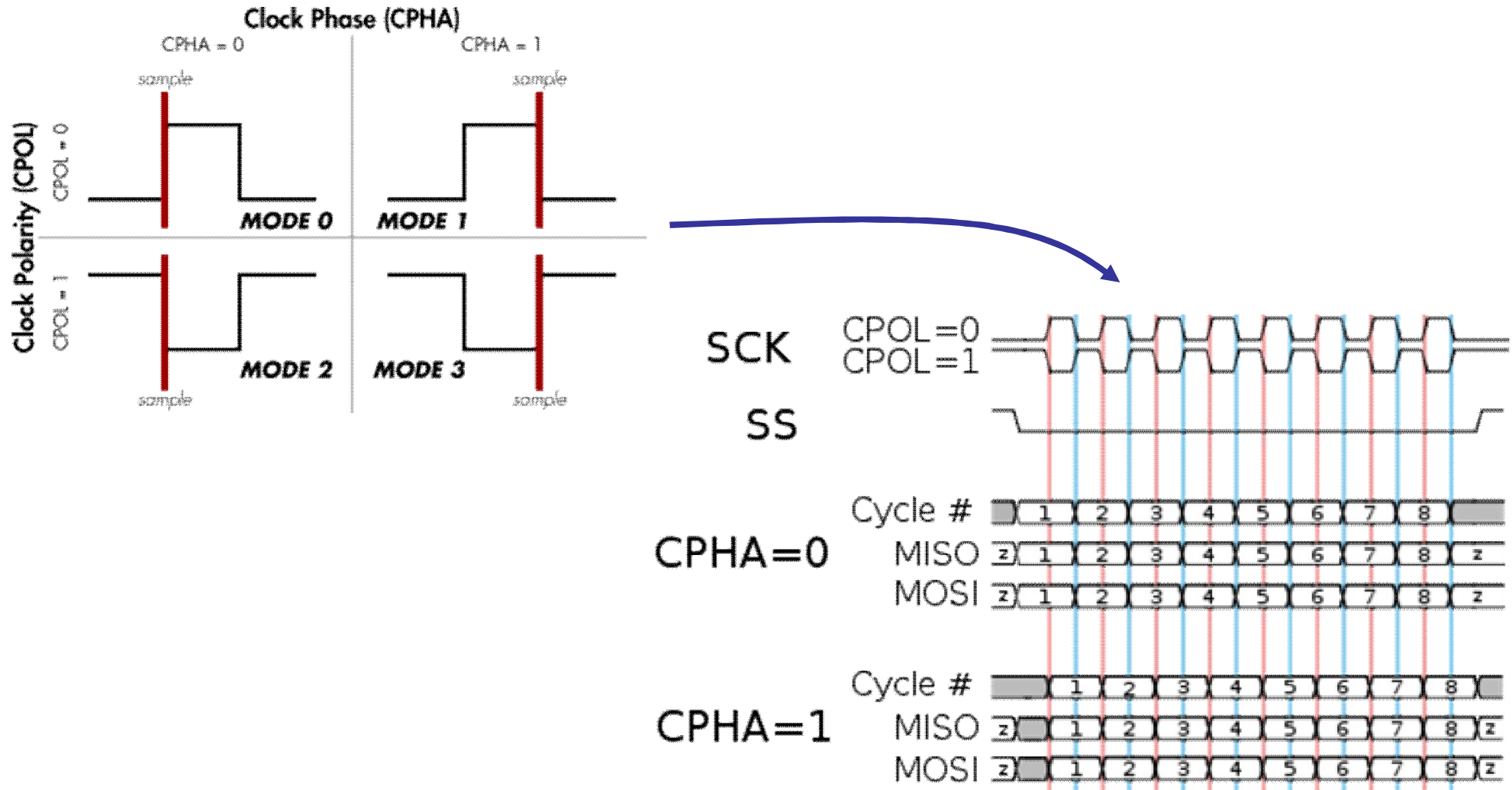


Możliwe konfiguracje magistrali

Single Master – Multiple Slave
(cooperative)



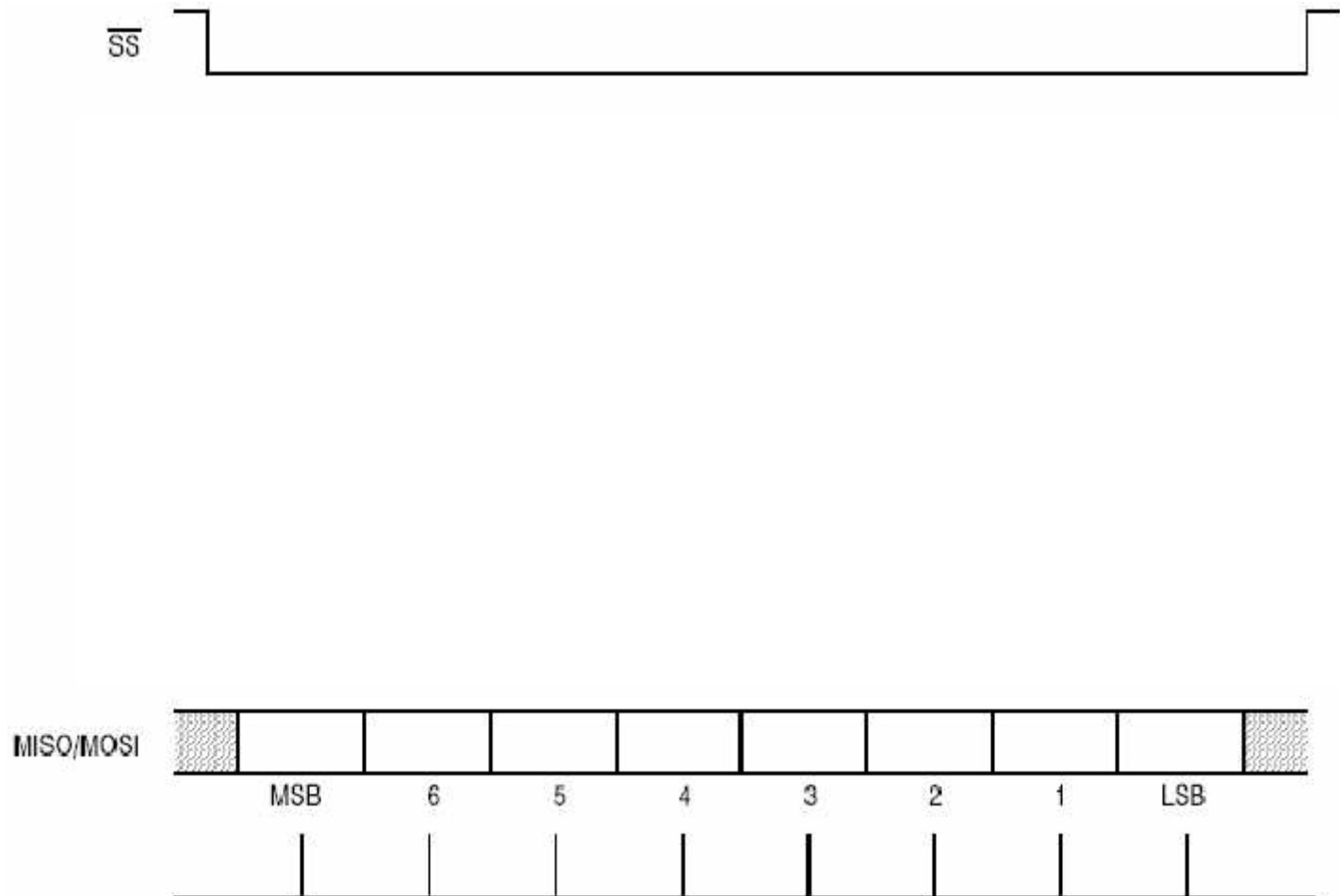
Tryby komunikacji SPI



Właściwości

- Układ *Master*:
 - Inicjuje transfer danych na magistrali – ustala częstotliwość zegara SCLK (tym samym prędkość transmisji),
 - Zapewnia generowanie sygnału zegarowego SCK dla danych przesyłanych w obu kierunkach (MOSI oraz MISO).
- Układ *Slave*:
 - Urządzenie aktywne jedynie, gdy wybrane przez układ nadrzędny za pomocą linii nSS (osobna linia dla każdego urządzenia),
 - Gdy urządzenie Slave nie jest wybrane odłącza się od magistrali – linia MISO urządzenia przechodzi w stan wysokiej impedancji.

Dane na magistrali SPI



Podsumowanie

- Both SPI, I2C or 1-Wire provide good support for communication with slow peripheral devices that are accessed intermittently. EEPROMs and real-time clocks are examples of such devices.
- SPI is better suited than others for applications that are naturally thought of as data streams (as opposed to reading and writing addressed locations in a slave device). An example of a "stream" application is data communication between microprocessors or digital signal processors. Another is data transfer from analog-to-digital converters.

Źródło: <http://belhob.wordpress.com>

Podsumowanie

- SPI can achieve significantly higher data rates than I2C or 1-Wire. SPI-compatible interfaces often range into the tens of megahertz. SPI really gains efficiency in applications that take advantage of its duplex capability, such as the communication between a "codec" (coder-decoder) and a digital signal processor, which consists of simultaneously sending samples in and out.
- SPI devices communicate using a master-slave relationship. Due to its lack of built-in device addressing, SPI requires more effort and more hardware resources than I2C when more than one slave is involved.
- SPI tends to be simpler and more efficient than I2C in point-to-point (single master, single slave) applications for the very same reason; the lack of device addressing means less overhead.

Źródło: <http://belhob.wordpress.com>

Podsumowanie

- Slaves can be thought of as input/output devices of the master. SPI does not specify a particular higher-level protocol for master-slave dialog. In some applications, a higher-level protocol is not needed and only raw data are exchanged. An example of this is an interface to a simple codec. In other applications, a higher-level protocol, such as a command-response protocol, may be necessary. **Note that the master must initiate the frames for both its command and the slave's response.**
- Far from being just a dumb "byte port," SPI is often an elegant solution for modest communication needs. It can also serve as a platform on which to create higher-level protocols.

Źródło: <http://belhob.wordpress.com>

Podsumowanie

- SPI does not have an acknowledgement mechanism to confirm receipt of data. In fact, without a communication protocol, the SPI master has no knowledge of whether a slave even exists. SPI also offers no flow control. If you need hardware flow control, you might need to do something outside of SPI.
- SPI is a very simple communication protocol. It does not have a specific high-level protocol which means that there is almost no overhead. Data can be shifted at very high rates in full duplex. This makes it very simple and efficient in a single master single slave scenario.

Źródło: <http://belhob.wordpress.com>

Przykład komunikacji SPI

```
void spiInit(void) {
    //kierunki linii interfejsu - wyjścia:
    SPI_DDR &= SPImask;
    SPI_DDR |= (1 << MOSI) | (1 << SCK) | (1 << nSS);

    //stany logiczne linii portu:
    SPI_PORT &= SPImask;
    SPI_PORT |= (1 << MOSI) | (1 << SCK) | (1 << nSS) | (1 << MISO);

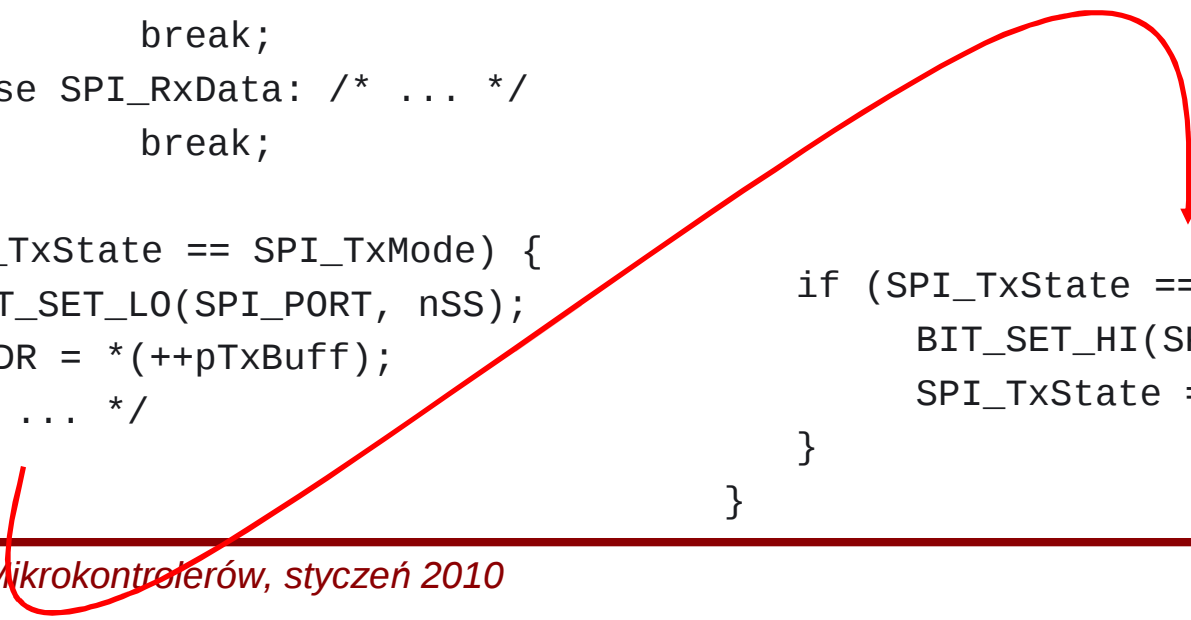
    //konfiguruj interfejs SPI:
    //SPI Interrupt Enabled, SPI Master, SCLK: fclk /2, MSB first,
    //SPI mode 0 (CPOL = 0, CPHA = 0):
    SPCR = (1 << SPIE) | (1 << SPE) | (1 << MSTR) | (1 << SPR1) | (1 << SPR0);
    SPSR = (1 << SPI2X);

    //stan początkowy nadajnika oraz odbiornika:
    SPI_TxState = SPI_TxIdle;
    SPI_RxState = SPI_RxIdle;
}
```

Przykład komunikacji SPI

```
ISR(SPI_STC_vect) {  
    uint8_t RcvByte = SPDR;  
    switch (SPI_RxState) {  
        case SPI_RxIdle: /* ... */  
            break;  
        case SPI_RxSync: /* ... */  
            break;  
        case SPI_RxPcktClass: /* ... */  
            break;  
        case SPI_RxDataCount: /* ... */  
            break;  
        case SPI_RxData: /* ... */  
            break;  
    }  
    if (SPI_TxState == SPI_TxMode) {  
        BIT_SET_LO(SPI_PORT, nSS);  
        SPDR = *(++pTxBuff);  
        /* ... */  
    }
```

```
    if (SPI_TxState == SPI_TxLastByte) {  
        BIT_SET_HI(SPI_PORT, nSS);  
        SPI_TxState = SPI_TxIdle;  
    }  
}
```



Więcej o SPI

- Introduction to Serial Peripheral Interface:
 - <http://embedded.com/columns/beginnerscorner/9900483?requestid=28276>
- SPI Intro:
 - <http://www.mct.net/faq/spi.html>
- Microchip SPI Information:
 - <http://ww1.microchip.com/downloads/en/DeviceDoc/spi.pdf>



Politechnika Łódzka

Instytut Elektroniki

Dziękuję za uwagę.

