

Systemy cyfrowe i podstawy elektroniki

Adam Szmigielski

aszmigie@pjawst.edu.pl

materiały: *ftp(public) : //aszmigie/SYC*

Metody komunikacji - wykład 14

Komunikacja z urządzeniami zewnętrznymi

- *Porty*
- *Łączy równoległe*
- *Łączy szeregowo*

Wymiana informacji - procesor, pamięć oraz urządzenia wejścia-wyjścia

- Większość mikrokontrolerów (Intel, AVR, PIC) używa jednego *oktetu* (8 bitów) do przesłania lub odbioru danych w jednym cyklu rozkazowym,
- Dla operacji wejścia-wyjścia używa się podobnych mechanizmów przesyłania informacji jak dla pamięci (dla niektórych mikroprocesorów te same mnemoniki są używane do wymiany informacji zarówno z pamięcią jak i urządzeniami In-Out).

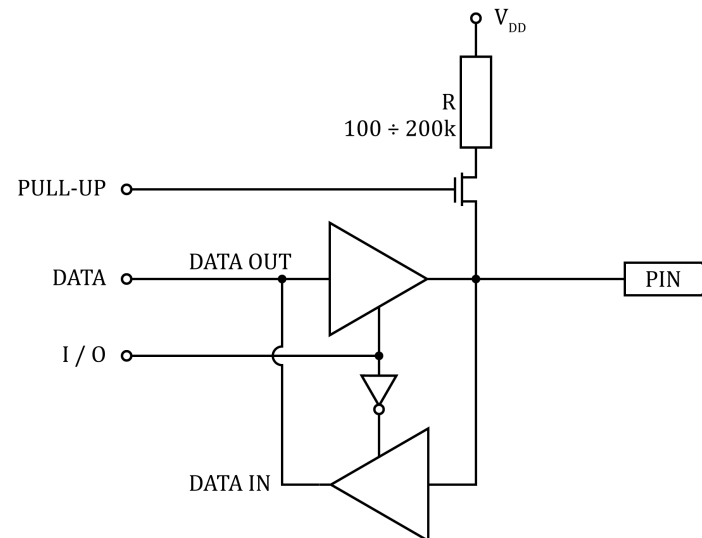
Porty

- Ogólnie, *port* jest interfejsem pomiędzy komputerem a innym komputerem lub urządzeniami peryferyjnymi,
- *porty sprzętowe* - jest fragmentem sprzętu umożliwiającego podłączenie urządzeń zewnętrznych i wymianę informacji pomiędzy nimi a kontrolerem,
- *porty programowe* (ang. software port) - jest wirtualnym (logicznym) połączeniem, które może być używane przez programistów do bezpośredniej wymiany danych (z pominięciem plików wymiany lub innych struktur przechowujących tymczasowo dane) np. porty TCP, UDP i inne.

Wejście i wyjście bezpośrednie

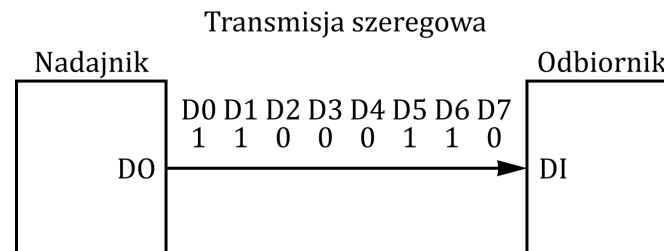
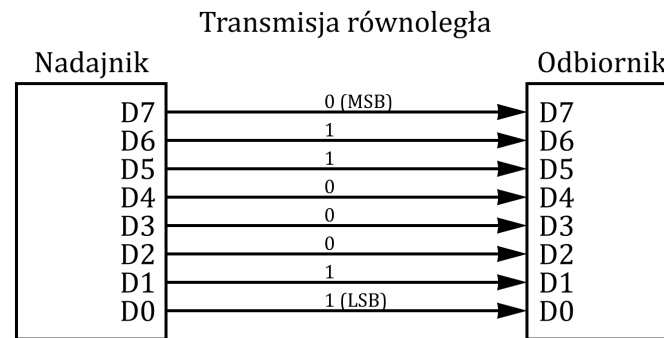
- Transmisja danych poprzez linie portów wejścia-wyjścia – sygnały cyfrowe zgodne ze standardem CMOS/TTL,
- Porty wejścia-wyjścia – organizacja bajtowa z możliwością ustawienia odczytu/zapisu pojedynczych linii (adresowanie bitowe),
- Po resecie poziom linii portów jest w stanie wysokiej impedancji,
- Linie portów mogą pełnić funkcje alternatywne,

Port wejścia-wyjścia - wybór trybu pracy



- Możliwe stany wejścia:
 - stan niski,
 - stan wysoki,

Komunikacja szeregową i równoległą



Ze względu na sposób przesyłania danych:

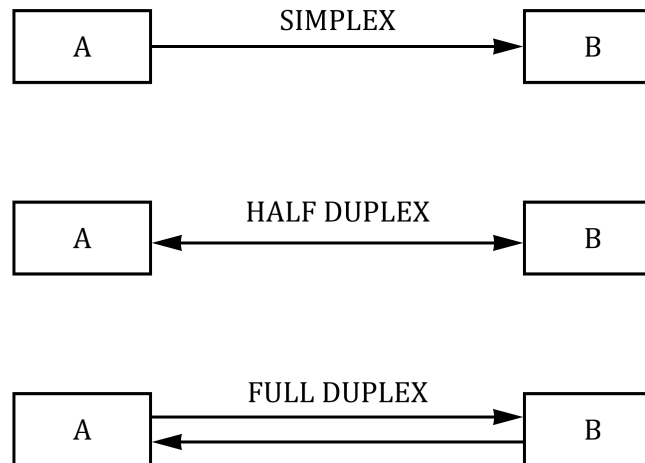
- *komunikacja szeregową* - bity przesyłane są kolejno po sobie,
- *komunikacja równoległą* - kilka bitów, zwykle krotność bajtu, przysyłanych jest jednocześnie.

Komunikacja synchroniczna i asynchroniczna

Ze względu na rodzaj synchronizacji:

- *Transmisja synchroniczna* - przesyłanie danych odbywa się w określonych chwilach, wyznaczonych poprzez sygnał synchronizujący (zegara),
- *Transmisja asynchroniczna* - transmisja danych może być rozpoczęta w dowolnej chwili czasowej.

Simplex, Duplex i Full Duplex

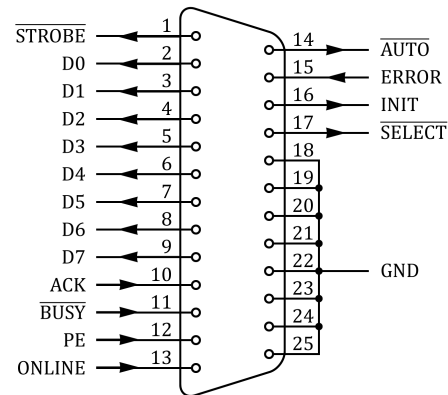


- *Simplex* - Informacja przesyłana jest tylko od nadajnika do odbiornika,
- *Duplex* - Informacja może być przesyłana w obu kierunkach. W danej chwili czasowej możliwy jest tylko jeden kierunek transmisji,
- *Full Duplex* - Informacja może być przesyłana w obu kierunkach jednocześnie.

Komunikacja równoległa

1. *Wyjście/wyjście bezpośrednie* - zazwyczaj port może pełnić wiele różnych funkcji. Wyboru funkcji jaki ma pełnić port dokonuje się programowo.
2. *Port równoległy* (ang. Parallel Port) - port, w którym dane są przesyłane jednocześnie kilkoma przewodami, z których każdy przenosi jeden bit informacji.

Port LPT



- Interfejs IEEE 1284 - nazwa 25-pinowego złącza w komputerach osobistych w standardzie TTL.
- IEEE 1284 jest portem równoległym wykorzystywanym w głównej mierze do podłączenia urządzeń peryferyjnych: drukarki, skanery, plotery.
- Został opracowany w 1984 r.
- Port obsługuje układ 8255, składający się z *rejestr danych* 00H, *rejestr wejściowy* (Status) 01H, *rejestr sterujący* (Control) 02H

Komunikacja szeregową

- *Transmisja szeregową* - dane są przesyłane jednym przewodem (albo jedną parą), i poszczególne bity informacji są przesyłane kolejno.
- Zwykle pod tą nazwą rozumie się transmisję opracowaną do komunikacji komputera z modemem zewnętrznym (standard RS-232).
- Transmisja szeregową może być *synchroniczna* albo *asynchroniczna*

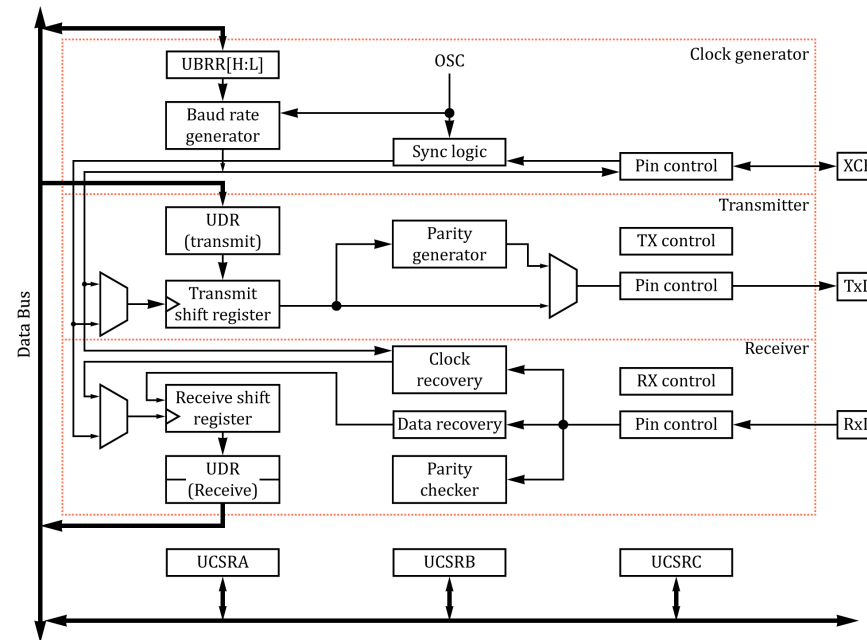
Rodzaje interfejsów szeregowych

- *RS-232* – standardowy interfejs szeregowy, standard został zaprojektowany do komunikacji z modemem. W standardzie tym musi nastąpić konwersja poziomów napięć.
- *I²C* – Interfejs *Inter-Integrated Circuit* oznacza “pośredniczący pomiędzy układami scalonymi”. Standard *I²C* określa dwie najniższe warstwy modelu odniesienia OSI: warstwę fizyczną i warstwę łącza danych.
- *SPI* – interfejs o możliwościach i właściwościach podobnych do I2C, o większej szybkości transmisji danych sięgającej kilka MB/s. Interfejs ten jest wbudowywany w wiele mikrokontrolerów.
- *D²BUS* – Interfejs Digital Data Bus opracowany przez firmę Philips, stworzony z myślą o łączeniu niewielkiej liczby urządzeń na małym obszarze, umożliwiającą transmisję danych z szybkością $100 \frac{kbit}{s}$ pomiędzy urządzeniami oddalonymi od siebie o 150m. Pozwala

zaadresować 4096 jednostek.

- *CAN* - Controller Area Network (CAN) jest szeregową magistralą komunikacyjną powstałą w latach 80-tych w *Bosch GmbH* z myślą o zastosowaniach w przemyśle samochodowym (ABS, sterowanie silnika).
- *IEEE 1394* - FireWire to standard łączy szeregowego umożliwiającego synchroniczną komunikację. Opracowany w roku 1995 (przez firmę Apple Inc.) dla komputerów osobistych i cyfrowych urządzeń optycznych.
- *USB* (ang. Universal Serial Bus) – Zaawansowany interfejs szeregowy, opracowany głównie z myślą o wykorzystaniu w komputerach klasy PC, realizujący koncepcję *plug and play* w odniesieniu do urządzeń zewnętrznych. Może obsługiwać do 127 urządzeń peryferyjnych, stosując dużą szybkość transmisji danych - $1,5 \frac{Mbit}{s}$ w przypadku obniżonej prędkości i $12 \frac{Mbit}{s}$ przy prędkości pełnej.

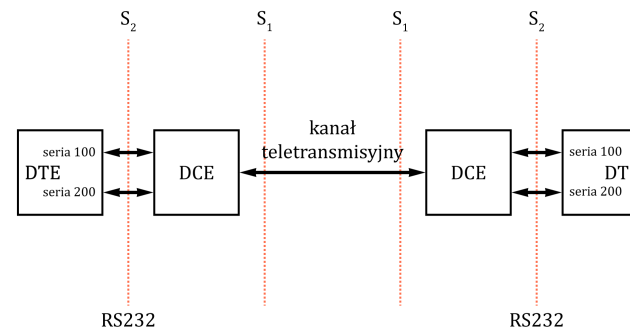
Interfejs szeregowy USART



- Full Duplex Operation, transmisja synchroniczna i asynchroniczna,
- Możliwość transmisji danych 5, 6, 7, 8 lub 9 bitowych z 1 lub 2 bitami stopu, bit parzystości wykrywany sprzętowo, wykrywanie błędów transmisji,
- Przerwania: TX Complete, TX Data Register Empty, RX Complete.

Interfejs RS232

- Pierwotnie standard RS232 został opracowany w celu ujednolicenia szeregowego interfejsu pomiędzy terminalem a *modemem* w instalacjach zdalnego dostępu,
- *Interfejs RS232* obecnie jest często stosowany także do bezpośredniego łączenia różnego rodzaju urządzeń (komputery z urządzeniami pomiarowymi, sterownikami, telefonami gsm, odbiornikami GPS, itp).
- W przyjętej terminologii wyróżnia się dwa rodzaje współpracujących urządzeń:



- urządzenie końcowe dla danych DTE (ang. Data Terminal Equipment),
- urządzenie komunikacyjne DCE (Data Communication Equipment).

Główne sygnały interfejsu *RS232*



Złącze *RS232* można spotkać jako DB9 i DB25.

Główne sygnały interfejsu *RS232*:

- *TxD* – *Transmitted Data* - dane nadawane,
- *RxD* – *Received Data* - dane odbierane,
- *SG* – *Signal Ground* - masa sygnałowa - ustala wspólny potencjał odniesienia dla wszystkich linii sygnałowych,

Pozostałe sygnały interfejsu RS232

- *DSR – Data Set Ready* - gotowość DCE - potwierdza fakt zestawienia połączenia,
- *DTR – Data Terminal Ready* - gotowość DTE - gotowość DTE do współpracy z DCE,
- *RTS – Request To Send* - żądanie nadawania,
- *CTS – Clear To Send* - gotowość do nadawania,
- *DCD – Data Carrier Detected* - obecność fali nośnej w kanale transmisyjnym,
- *RI – Ring Indicator* - sygnał wystawiany przez DCE (często niewykorzystany),
- *PG – Protective Ground* - masa ochronna - obniża poziom zakłóceń, w złączu DB-9 rolę masy pełni metalowa osłona.

Poziomy napięć interfeju RS232

- Na liniach danych (RxD, TxD) obowiązuje *logika ujemna*, tzn. logicznej “1” odpowiada stan niski

stan logiczny	napięcie
0	$+3V \leq U_{Rx,Tx} \leq +15V$
1	$-15V \leq U_{Rx,Tx} \leq -3V$

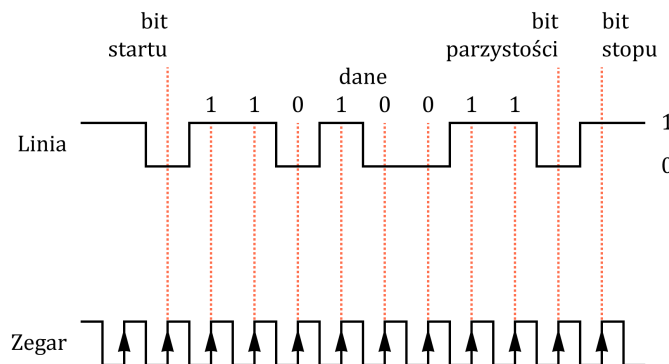
- Na pozostałych liniach stosuje się *logikę dodatnią*

stan logiczny	napięcie
0	$-15V \leq U_{Rx,Tx} \leq -3V$
1	$+3V \leq U_{Rx,Tx} \leq +15V$

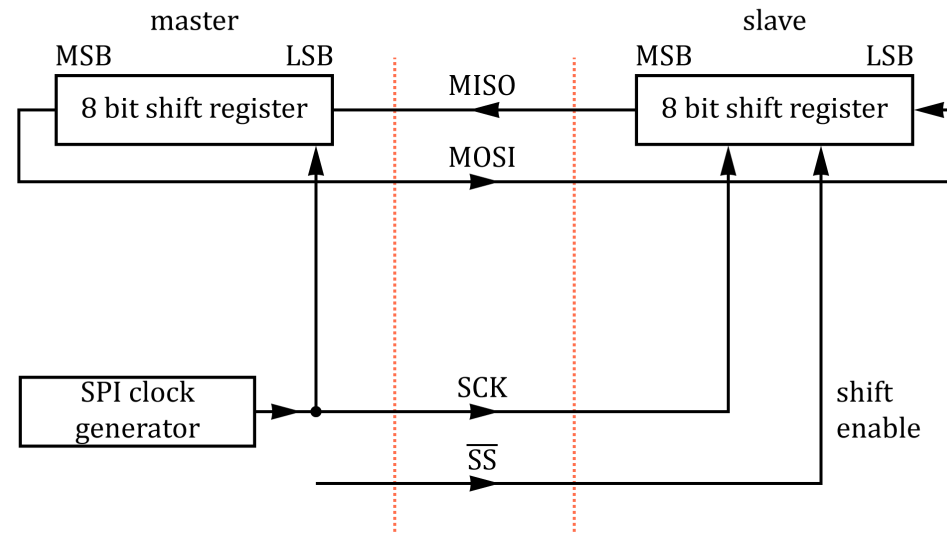
- Przedział $< -3V, +3V >$ jest przejściowy i nie określa stanu obwodu,
- Nie należy łączyć linii interfejsu RS232 bezpośrednio z liniami w standardzie TTL.**
Specjalistyczny układów - MAX232 dopasowuje poziomy napięć.

RS232 - transmisja synchroniczna i asynchroniczna

- W *trybie synchronicznym* nadajnik i odbiornik są taktowane z tego samego źródła sygnału zegarowego. Sygnał zegarowy pomiędzy DTE i DCE jest przesyłany bezpośrednio po wydzielonych liniach,
- W *trybie asynchronicznym* nadajnik i odbiornik korzystają z osobnych, niezależnych generatorów sygnału zegarowego. Do każdego słowa danych dodaje się START i STOP, wskazujące moment kiedy rozpoczyna się i kończy transmisja.

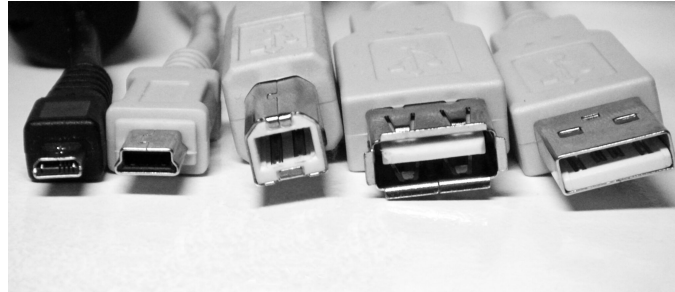


Interface szeregowy SPI - ang. Serial Peripheral Interface



- Full-duplex, synchroniczny transfer danych, 7 programowalnych prędkości transmisji,
- Możliwość pracy w trybie Master lub Slave,
- Koniec transmisji identyfikowany flagą przerwań.

Universal Serial Bus USB 2.0



- Sygnał przesyłany jest z wykorzystaniem dwóch linii $D-$ i $D+$,
- Na linii $D-$ jest zanegowany sygnał $D+$ - podwaja to amplitudę sygnału,
- Dane przesyłane są w trybie half-duplex,
- Wyprowadzone źródło zasilania $+5V$.

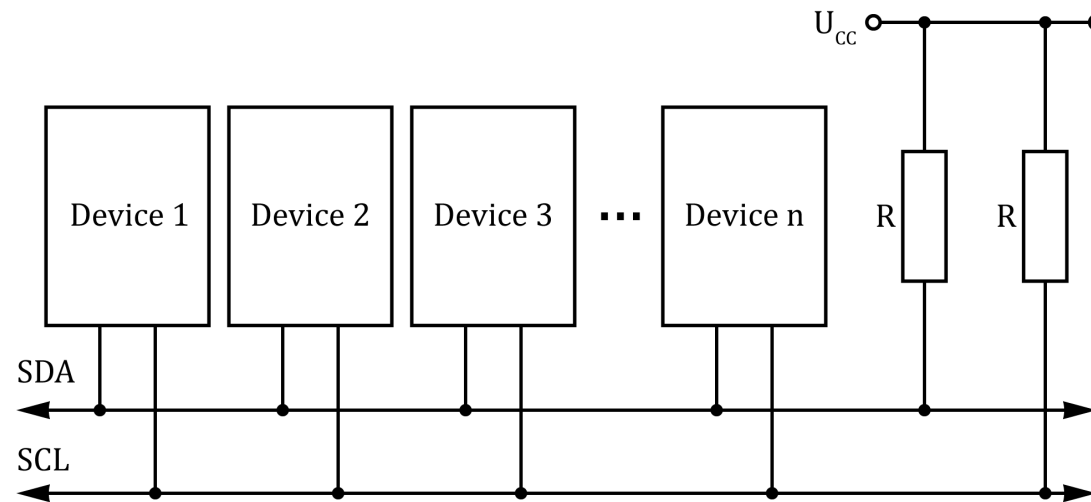
kabel	kolor	opis
V_{CC}	czerwony	$+5V$
$D-$	biały	dane-
$D+$	zielony	dane+
GRD	czarne	masa $0V$

Transmisja pakietowa USB

Istnieje pięć podstawowych typów pakietów, które służą do komunikacji w transmisji USB:

- *Handshake packets* - Odpowiedź składa się tylko z ośmiobitowego pola PID,
- *Start of Frame SOF* - Wskazuje na początek ramki transmisyjnej,
- *Token packets* - Zapowiedź jest wysyłany przez komputer, żeby zainicjować wymianę,
- *Data packets* - Dane mogą być wysyłane zarówno przez komputer jak i urządzenie odbiorcze,
- *Special packets* - wyróżnia się *Start-split (SSPLIT) Token* i *Complete-split (CSPLIT) Transaction Token*.

Interface szeregowy - Two-wire

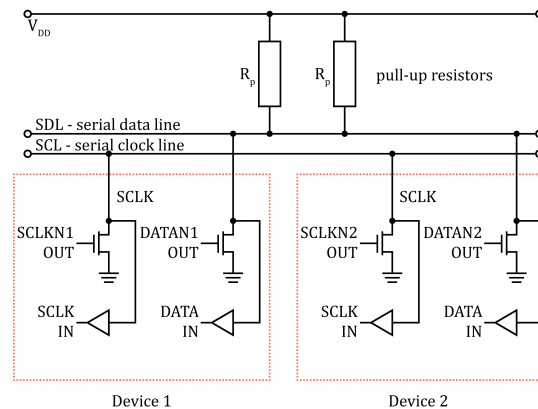


- Obsługa trybów Master i Slave, arbitraż Multi-master,
- 7-bitowy adres (128 adresów Slave), w pełni programowalny adres Slave-a w ramach General Call Support
- Prędkość transmisji do $400kHz$

Interfejs I^2C

- Standard został opracowany na początku lat 80. (określany obecnie jako tryb standardowy pracy) i cechowały go: prędkość transmisji 100 kbps 7-bitowa przestrzeń adresowa
- W 1992 roku została opracowana wersja 1.0: dodanie trybu pracy z prędkością transmisji 400 kbps (Fast Mode) rozszerzenie standardu o możliwość adresowania 10-bitowego
- W 1998 roku opracowana została wersja 2.0: dodanie trybu High Speed Mode, pozwalającego na prędkość transmisji 3,4 Mbps
Zwiększenie zakresu tolerancji napięcia w stanie wysokim: 2,3 – 5,5 V
- Standard I^2C określa dwie najniższe warstwy modelu odniesienia OSI: warstwę fizyczną i warstwę łącza danych.

Interfejs I^2C - warstwa fizyczna



- I^2C do transmisji wykorzystuje dwie dwukierunkowe linie:
 - *SDA* - linia danych (ang. Serial Data Line)
 - *SCL* - linia zegara (ang. Serial Clock Line).

Obydwie linie są na stałe podciągnięte do źródła zasilania poprzez rezystory podciągające (pull-up).

- I^2C używa logiki dodatniej, a więc stan niski na magistrali odpowiada “0” logicznemu, natomiast stan wysoki “1” logicznej.
- Wszystkie nadajniki są typu otwarty kolektor.

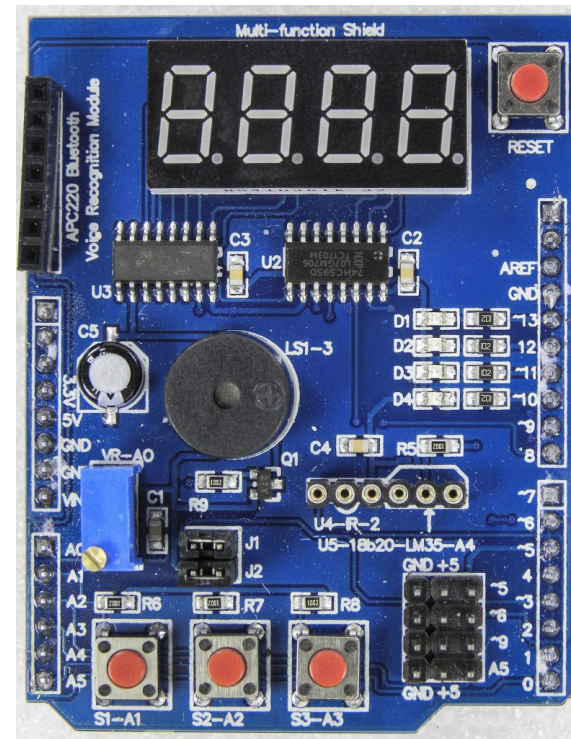
- Podstawowa wersja I^2C zakłada istnienie tylko jednego urządzenia, które może inicjować transmisję (master),
- *Mechanizmu detekcji kolizji*, możliwa jest praca w trybie multi-master. Ponieważ dane nadawane są w kolejności od najstarszego bitu do najmłodszego, w przypadku jednoczesnego nadawania, urządzenie nadające adres o wyższym numerze wycofa się pierwsze.
- *Arbitraż* ze stałym przydziałem priorytetów. Urządzenia o niższych adresach mają wyższy priorytet.
- Zmiana na linii danych podczas transmisji może następować jedynie, gdy linia zegara znajduje się w stanie niskim.
- Bit startu występuje, gdy linia danych zmienia swój stan z “1” na “0”.
- Po zakończeniu transmisji generowany jest bit stopu, czyli przejście linii danych w stan wysoki przy wysokim stanie linii zegara.
- Długość linii ograniczona jest jedynie jej maksymalną pojemnością, która wynosi 400 pF.

Interfejs I^2C - warstwa łącza danych

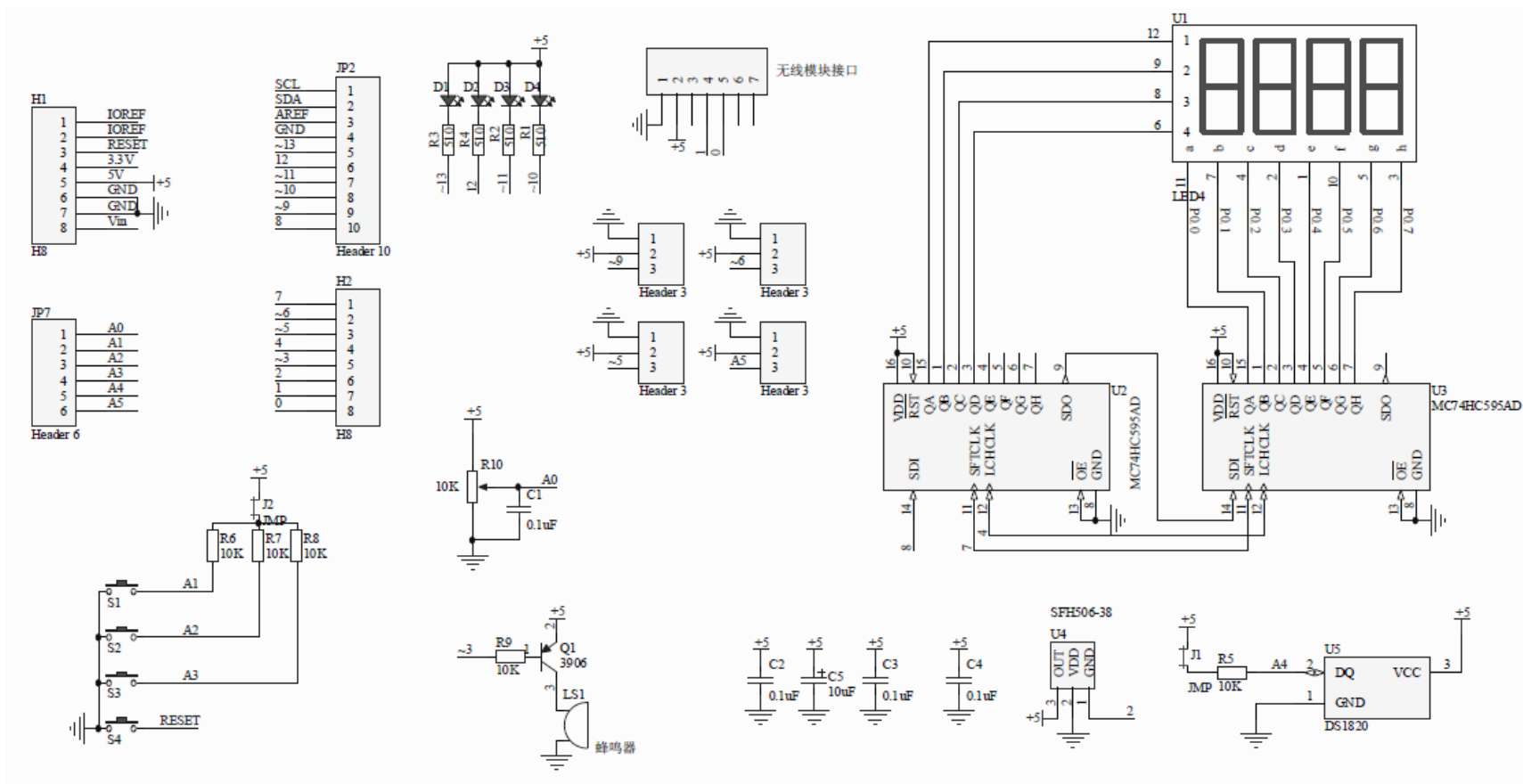
- I^2C jest magistralą zorientowaną bajtowo (bity grupowane po 8),
- Po przesłaniu 8 bitów przesyłany jest dodatkowy bit potwierdzenia odebrania danych ACK,
- Pierwszym bajtem jest zawsze nadawany przez urządzenie master adres urządzenia slave, który oprócz 7 bitów właściwego adresu zawiera bit kierunku transmisji (na najmłodszej pozycji),
- Wartość “0” tego bitu oznacza transmisję od mastera do slave’a (zapis), podczas gdy wartość “1” kierunek przeciwny (odczyt). Po pierwszym bajcie przesyłane zostają dane,
- Standard zakładał 7-bitową przestrzeń adresową, czyli możliwość zaadresowania do 128 urządzeń (w praktyce 112 urządzeń),
- Jednym z zarezerwowanych adresów jest tzw. General call (adres 0), który powoduje wysłanie danych do wszystkich urządzeń,

Wielofunkcyjny moduł ćwiczeniowy

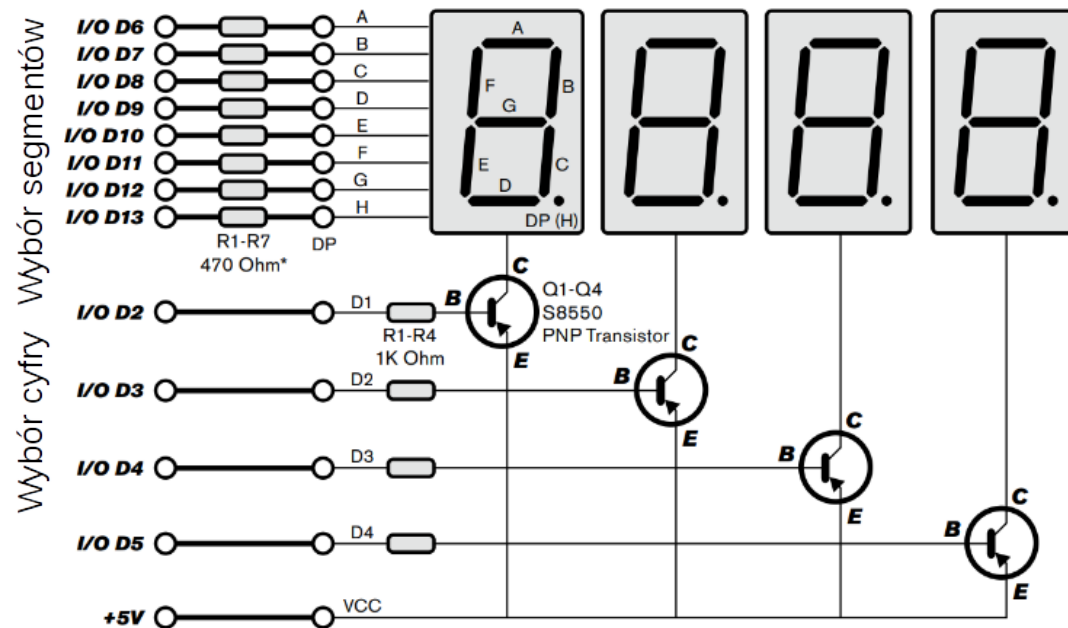
- A0 - wejście analogowe z potencjometru,
 - A1, A2, A3 - wejścia przycisków (z rezystorami podciągającymi)
 - 10, 11, 12, 13 diody
 - A4, A5 - wejście i2c
 - 5, 6, 8, A5 piny (VCC + GND)
 - 2 - przycisk do czujników (int0)
 - 3 buzzer
 - 4 LATCH (rejestr)
 - 7 CLK (rejestr)
 - 8 DIO (rejestr)
-



Wielofunkcyjny moduł ćwiczeniowy - schemat



Wyświetlacz LED



- rejestr pierwszy - wybór wyświetlanej cyfry,
- rejestr drugi - rodzaj wyświetlanej cyfry

SPI - Zapis cyfry do wyświetlacza

```
#define LATCH_DIO 4
#define CLK_DIO 7
#define DATA_DIO 8
const byte SEGMENT_MAP[] = {0xC0,0xF9,0xA4,0xB0,0x99,0x92,0x82,0xF8,0X80,0X90};
const byte SEGMENT_SELECT[] = {0xF1,0xF2,0xF4,0xF8};

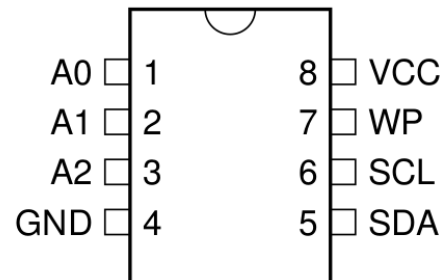
digitalWrite(LATCH_DIO,LOW);
shiftOut(DATA_DIO, CLK_DIO, MSBFIRST, SEGMENT_MAP[ Value ] );
shiftOut(DATA_DIO, CLK_DIO, MSBFIRST, SEGMENT_SELECT[ Segment ] );
digitalWrite(LATCH_DIO,HIGH); ;
```

I2C memory chip

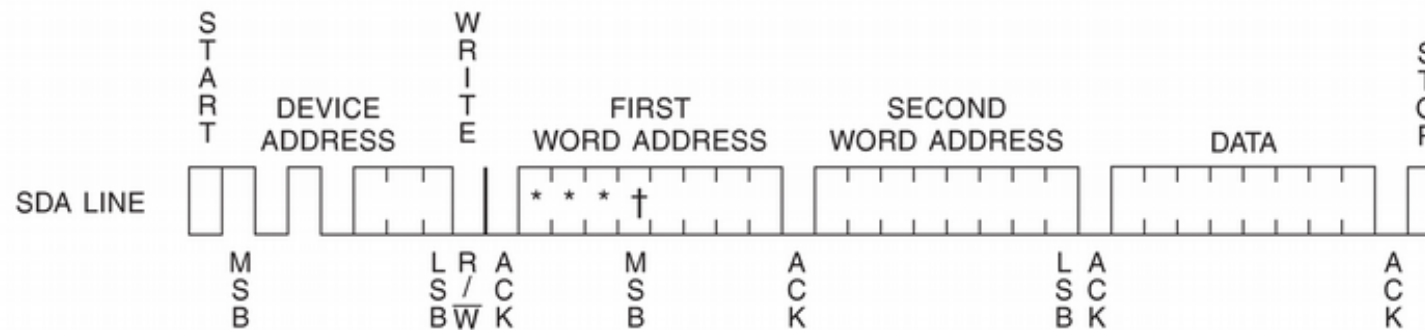
1	0	1	0	A ₂	A ₁	A ₀	R/W
MSB				LSB			

Pin Name	Function
A0 - A2	Address Inputs
SDA	Serial Data
SCL	Serial Clock Input
WP	Write Protect

8-Pin PDIP

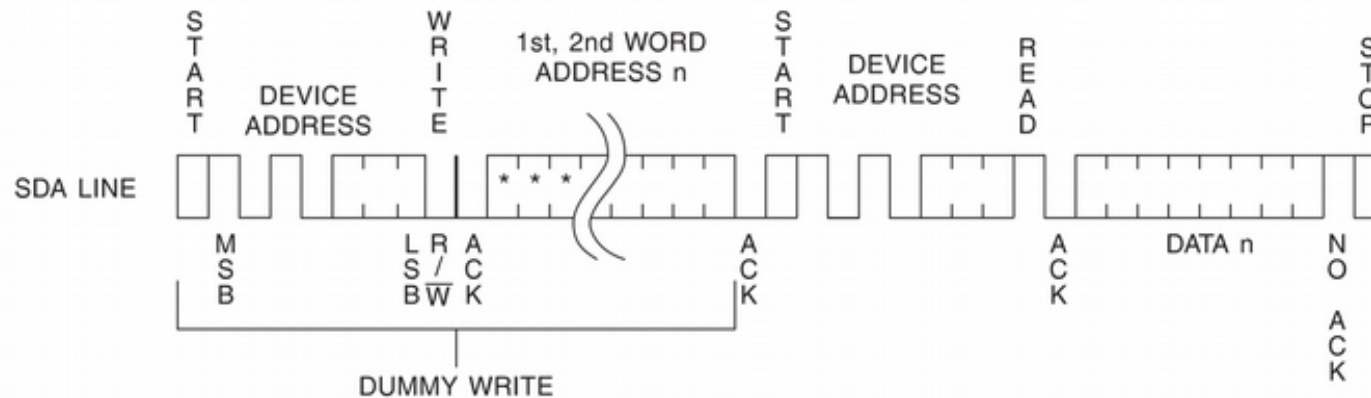


I2C - Zapis bajtu do pamięci



```
Wire.beginTransmission(deviceaddress);  
Wire.write(eeaddress >> 8); // MSB  
Wire.write(eeaddress & 0xFF); // LSB  
Wire.write(data);  
Wire.endTransmission();
```

I2C - Odczyt bajtu z pamięci (Random Read)



```
Wire.beginTransmission(deviceaddress); //dummy write
Wire.write(eeaddress >> 8);           // MSB
Wire.write(eeaddress & 0xFF);         // LSB
Wire.endTransmission();
```

```
Wire.requestFrom(deviceaddress, 1);
if (Wire.available()) byte rdata = Wire.read();
```

I2C - Odczyt bajtu z termometru DS3231

```
byte displayTemperature()  
{  
    byte data = 0xFF;  
  
    Wire.beginTransmission(0x68);  
    Wire.write(0x11); // the integer portion  
    Wire.endTransmission();  
  
    Wire.requestFrom(0x68, 1);  
    if (Wire.available()) { data = Wire.read();}  
  
    return data;  
}
```

Zadania na ćwiczenia

1. SPI (rejestr przesuwany) - Napisz program wyświetlający na wyświetlaczu napis "HELP"
2. I2C - Napisz program odczytujący z termometru (adres urządzenia 0x68, adres rejestru 0x11) wartość całkowitoliczbową temperatury. Wynik wyświetl na wyświetlaczu.
3. I2C - Napisz program odczytujący z termometru temperaturę z wartościami ułamkowymi (adres urządzenia 0x68, adres rejestru 0x11, adres rejestru części ułamkowej 0x12). Wynik wyświetl na wyświetlaczu wraz z kropką dziesiętną.