

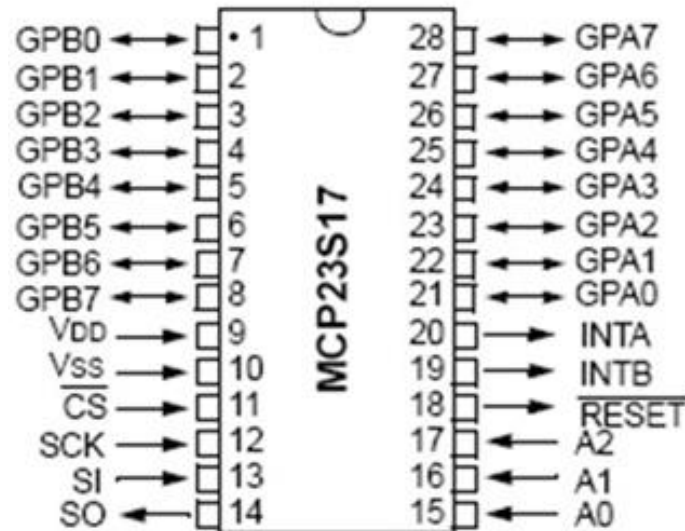
Expandery wejść
MCP23S17 oraz MCP23017

Expander I/O MCP20S17 I2C

Adafruit	UNO/Nano	MCP23017 (I2C Control)		UNO/Nano	Adafruit
D8		GPB0 ↔	1	28 ↔	D7
D9		GPB1 ↔	2	27 ↔	D6
D10		GPB2 ↔	3	26 ↔	D5
D11		GPB3 ↔	4	25 ↔	D4
D12		GPB4 ↔	5	24 ↔	D3
D13		GPB5 ↔	6	23 ↔	D2
D14		GPB6 ↔	7	22 ↔	D1
D15		GPB7 ↔	8	21 ↔	D0
	+5V	VDD →	9	20 →	Interrupt
	GND	Vss →	10	19 →	Interrupt
	---	NC	11	18 →	+5V
	A5 (SCL)	SCL →	12	17 ←	GND
	A4 (SDA)	SDA ↔	13	16 ←	GND
	---	NC	14	15 ←	GND

Piny wyjściowe expanderów MCP23S17 oraz MCP23017

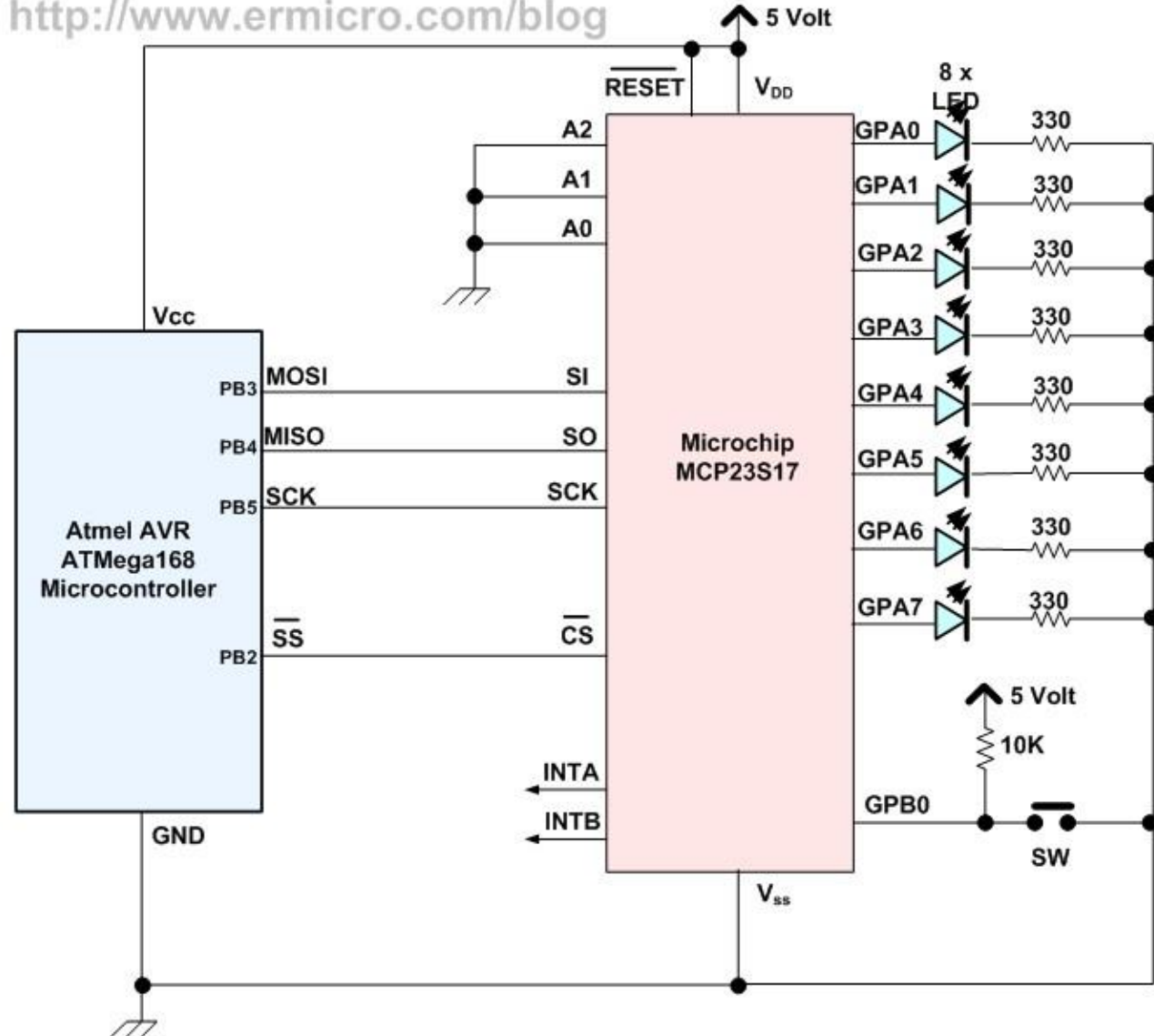
Expander I/O MCP23S17 SPI



Pin Name	Description
GPA0..7	8-bit General I/O Port A
GPB0..7	8-bit General I/O Port B
INTA	Port A Interrupt Signal
INTB	Port B Interrupt Signal
RESET	Reset Signal
A0,A1,A2	Configurable Address
CS	Chip Select (active Low)
CSK	Synchronous Clock
SI	Slave In
SO	Slave Out
V _{DD}	+5 Volt
V _{SS}	GND

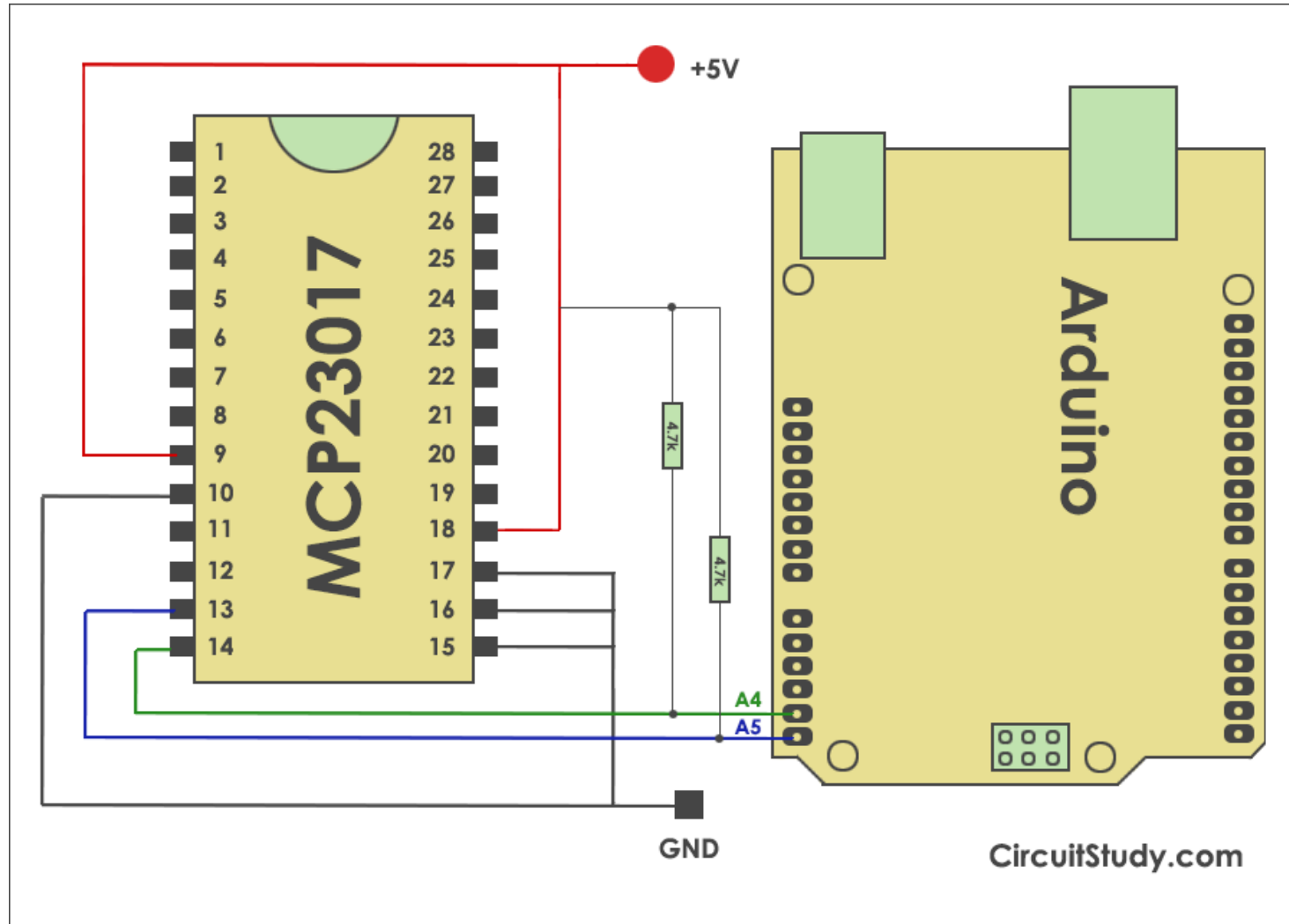
Podłączenie urządzenia na magistrali SPI z płytą Arduino.
Linie adresu hardwareowego są tu połączone z masą.

<http://www.ermicro.com/blog>

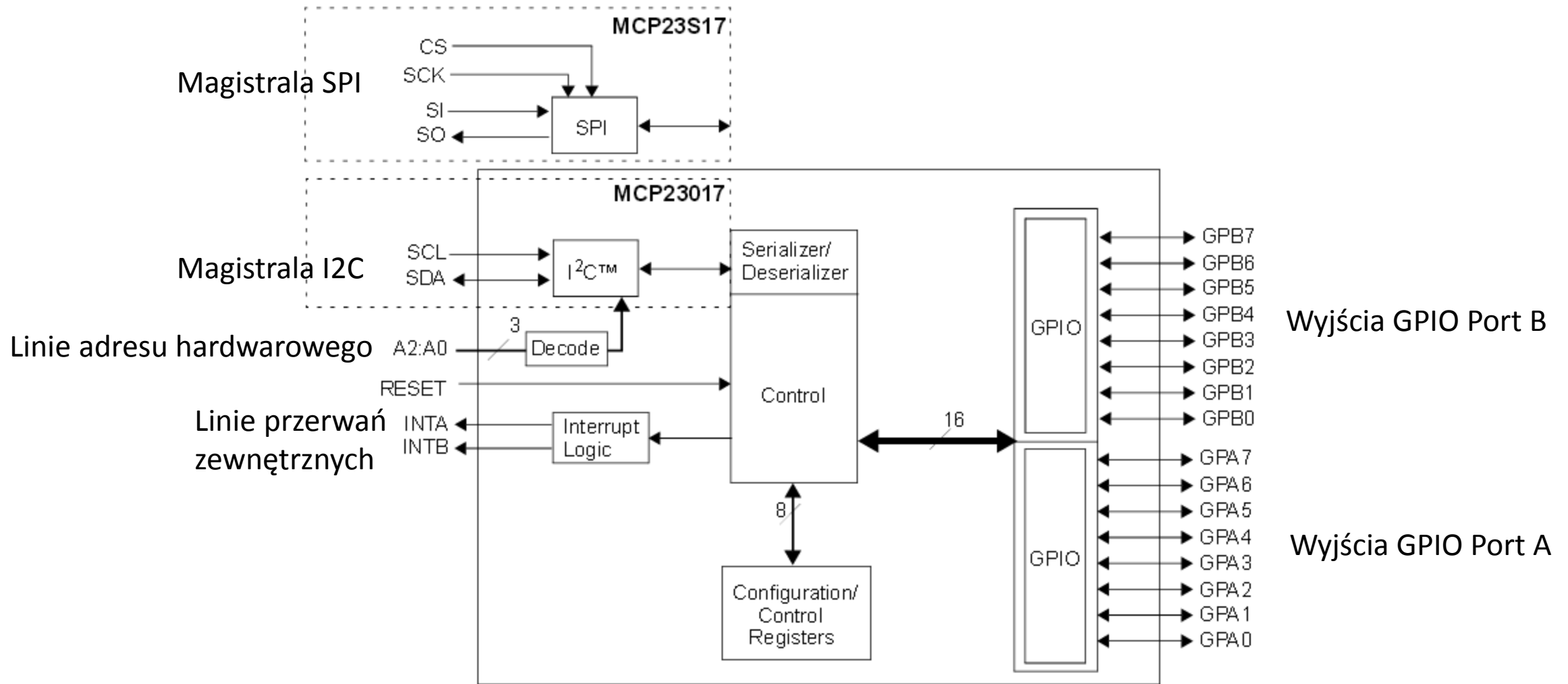


Atmel AVR ATmega168 and Microchip MCP23S17 16-Bit SPI I/O Expander Schema

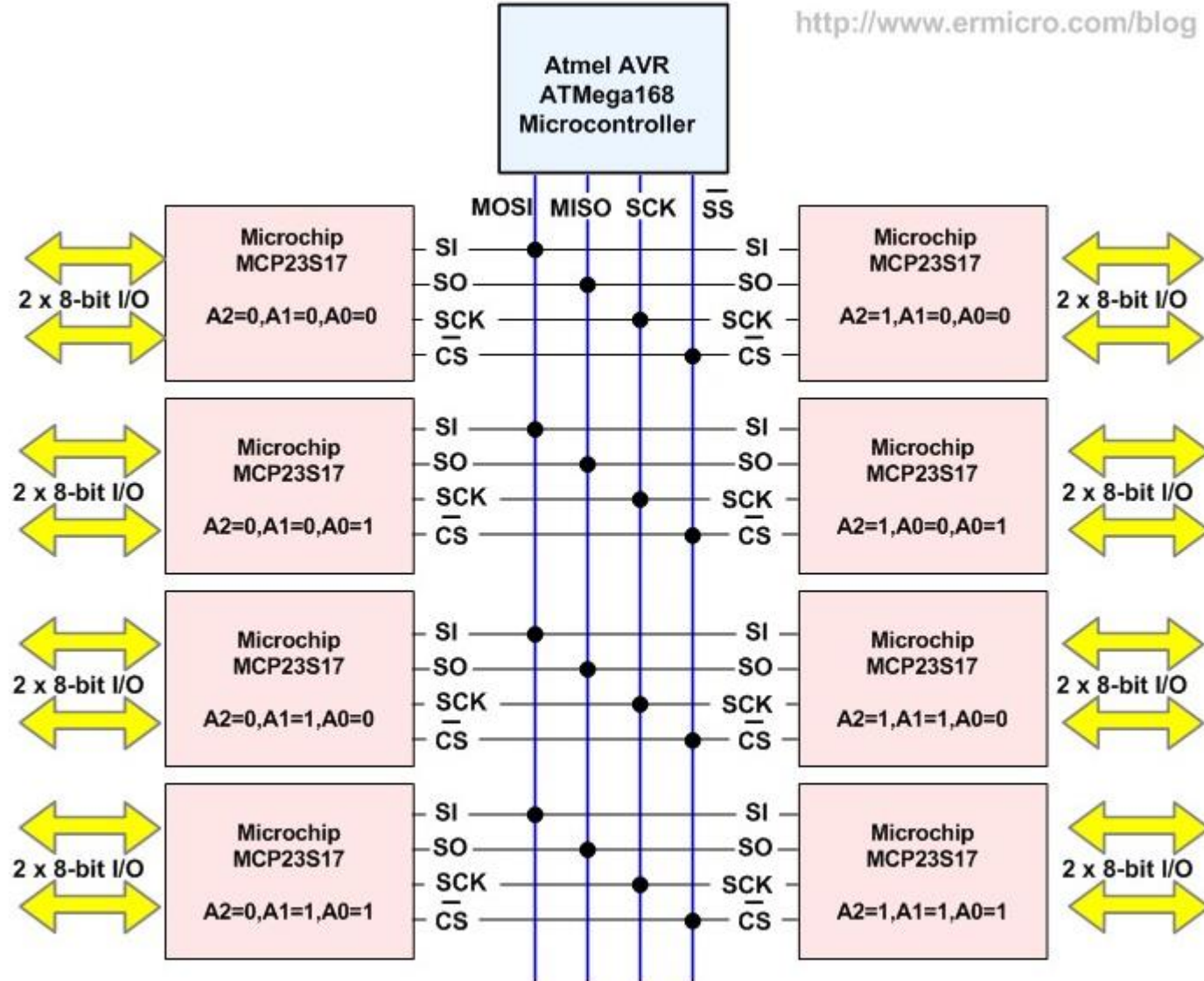
Podłączenie urządzenia na magistrali I2C z płytą Arduino (widoczne podwieszenia linii magistrali)
Linie adresu hardwareowego są tu połączone z masą.



Budowa blokowa expanderów I/O: MCP23S17 SPI oraz MCP23017 I2C

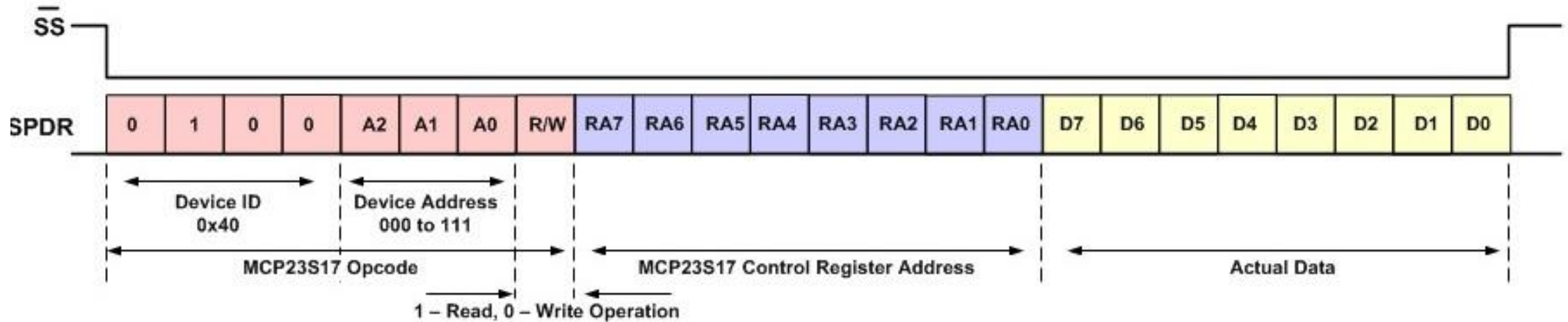


Linie adresu hardwarowego A2, A1, A0 (jednakowe dla MCP23S17 oraz MCP23017) pozwalają na umieszczenie i rozróżnienie w systemie do 8-miu identycznych urządzeń



AVR ATmega168 with 8 x Microchip MCP23S17 16-Bit SPI I/O Expander Connection

Format transmisji danych po magistrali SPI (dotyczy expandera MCP23S17)



Transmisja danych odbywa się w jednostkach po 3 bajty.

W 1-szym bajcie nadawany jest identyfikator urządzenia (Device ID: 0100) oraz adres hardwarowy ustawiony na pinach A2, A1, A0 (Device Address). Ostatni bit R/W decyduje o trybie: zapis lub odczyt.

W 2-gim bajcie nadawany jest adres rejestru expandera, do którego jest adresowana transmisja.

W 3-cim bajcie znajduje się dana dla rejestru expandera (w trybie zapisu) lub z rejestru (w trybie odczytu).

```
#include <SPI.h>
//linie magistrali SPI są zdefiniowane w bibliotece:
//static const uint8_t SS  = 10;
//static const uint8_t MOSI = 11;
//static const uint8_t MISO = 12;
//static const uint8_t SCK  = 13;

//konfiguracja kierunku linii
pinMode (linia, OUTPUT);
digitalWrite(linia,HIGH);

// funkcja SPI.begin(); włącza interfejs w trybie master
SPI.begin();

// funkcja SPI.setDataMode(SPI_MODEEx); ustala tryb x=0-3 związany z polaryzacją i fazą impulsów
zegarowych
SPI.setDataMode(SPI_MODEEx);

// funkcja SPI.setClockDivider(SPI_CLOCK_DIVx); ustala prędkość przez ustawienie dzielnika między
częstotliwością taktowania mikrokontrolera, a częstotliwością taktowania interfejsu x= 4, 8, 16, 32, 64, lub 128
SPI.setClockDivider(SPI_CLOCK_DIVx);

//transmisja jednego bajtu (nadawanie lub odbiór)
int re = SPI.transfer(value);
```

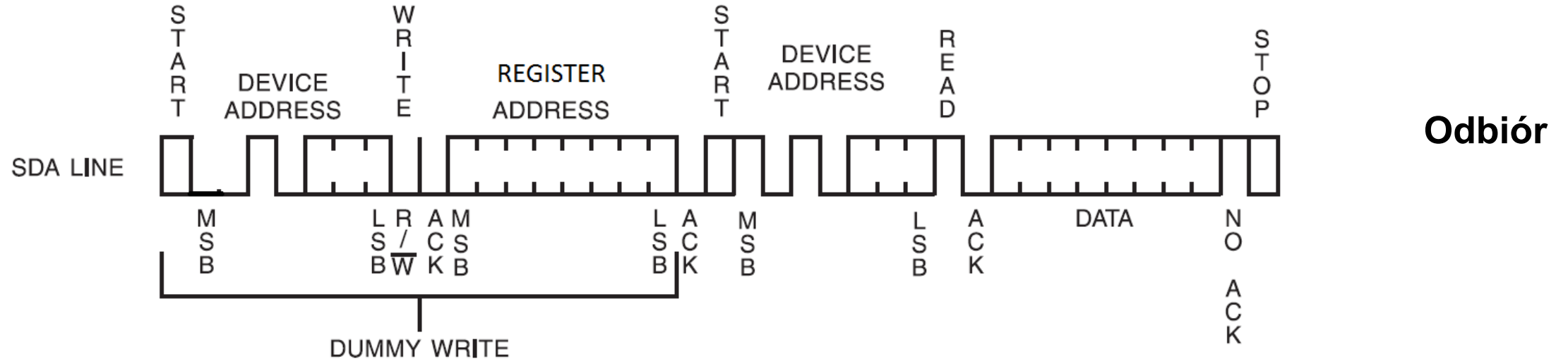
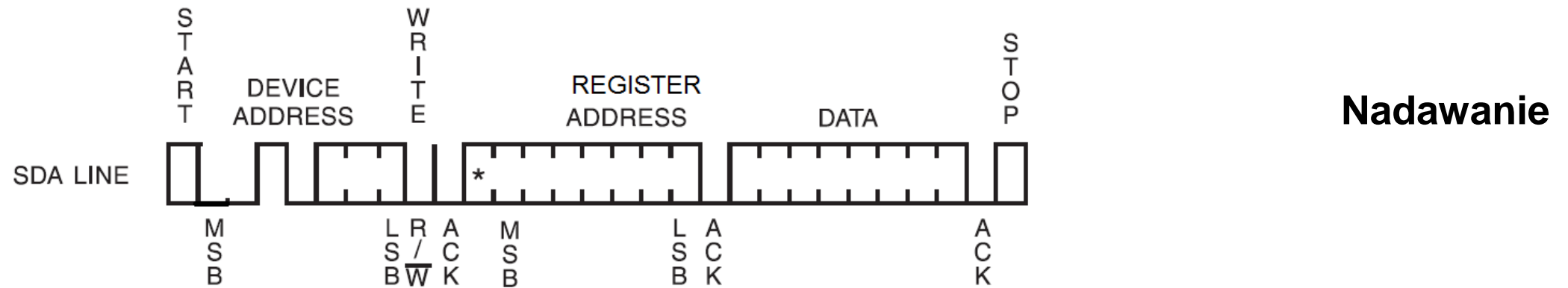

Format transmisji danych po magistrali I2C (dotyczy expandera MCP23017)

Transmisja danych odbywa się w jednostkach po 3 bajty.

W 1-szym bajcie nadawany jest identyfikator urządzenia (Device ID: 0010) oraz adres hardwarowy ustawiony na pinach A2, A1, A0 (Device Address). Ostatni bit R/W decyduje o trybie: zapis lub odczyt.

W 2-gim bajcie nadawany jest adres rejestru expandera, do którego jest adresowana transmisja.

W 3-cim bajcie znajduje się dana dla rejestru expandera (w trybie zapisu) lub z rejestru (w trybie odczytu).



```
#include <Wire.h>
//linie magistrali I2C są zdefiniowane w bibliotece:
//static const uint8_t SDA = A4;
//static const uint8_t SCL = A5;

// funkcja Wire.begin(); włącza interfejs
Wire.begin();

//nadanie bajtu Device Address obróconego o 1 bit w prawo
Wire.beginTransmission(Device Address >1);

//nadanie bajtu x
Wire.write(x);

//zakończenie nadawania
Wire.endTransmission();

//żądanie odbioru N bajtów spod adresu DeviceAddress >1
Wire.requestFrom(DeviceAddress >1, N);

//odbiór bajtów
while(Wire.available())    {
char c = Wire.read();    }
```

Wykaz rejestrów expanderów MCP23S17 oraz MCP23017

IODIR – odpowiedzialne za kierunek linii GPIO: 0 - wyjście, 1 - wejście

GPPUB – podwieszanie wewnętrzne linii wejściowych: 0 - nie podwieszone, 1 - podwieszone

IPOL – odwrócenie polaryzacji bitu sterującego skojarzoną linią (wpisanie zera do rejestru IPOL powoduje sterowanie skojarzonego pinu logiczną jedynką)

GPINTEN – maska przerwania zmiany stanu na skojarzonej linii (1 –maska zdjęta)

DEFVAL – rejestr wartości odniesienia dla porównania, pojawienie się wartości przeciwnej na skojarzonej linii wywoła przerwanie

INTCON – kontroluje z czym porównywany jest aktualny stan na pinie w celu stwierdzenia zmiany na pinie (1: porównanie z korespondującą wartością w rejestrze DEFVAL, 0: porównanie z poprzednim stanem na pinie)

IOCON – rejestr konfiguracyjny (wybór banku, sposobu adresowania, polaryzacji na linii przerwania)

INTF – rejestr flagowy przerwań, jedynka logiczna sygnalizuje przerwanie wywołane zmianą stanu na skojarzonej linii. Rejestr „tylko do odczytu”

INTCAP – rejestr przechwytyjący stan na skojarzonym pinie w chwili przerwania, stan rejestru pozostaje bez zmian aż do wykonania odczytu rejestru INTCAP lub GPIO. Rejestr „tylko do odczytu”

GPIO – odczyt stanu na pinie wejściowym oraz sterowanie stanem pinu wyjściowego

OAT – dla wyjść: sterowanie wejściem wyjściowego przerzutnika D (stan ten po jednym takcie zegarowym pojawia się na pinie wyjściowym), dla wejść: odczyt ostatniego stanu na wejściu przerzutnika (nie na pinie wejściowym)

IOCON - I/O Expander Configuration Register

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0
BANK	MIRROR	SEQOP	DISSLW	HAEN	ODR	INTPOL	—
bit 7							bit 0

bit 7 **BANK:** Controls how the registers are addressed
1 = The registers associated with each port are separated into different banks
0 = The registers are in the same bank (addresses are sequential)

bit 6 **MIRROR:** INT Pins Mirror bit
1 = The INT pins are internally connected
0 = The INT pins are not connected. INTA is associated with PortA and INTB is associated with PortB

bit 5 **SEQOP:** Sequential Operation mode bit.
1 = Sequential operation disabled, address pointer does not increment.
0 = Sequential operation enabled, address pointer increments.

bit 4 **DISSLW:** Slew Rate control bit for SDA output.
1 = Slew rate disabled.
0 = Slew rate enabled.

bit 3 **HAEN:** Hardware Address Enable bit (MCP23S17 only).
Address pins are always enabled on MCP23017.
1 = Enables the MCP23S17 address pins.
0 = Disables the MCP23S17 address pins.

bit 2 **ODR:** This bit configures the INT pin as an open-drain output.
1 = Open-drain output (overrides the INTPOL bit).
0 = Active driver output (INTPOL bit sets the polarity).

bit 1 **INTPOL:** This bit sets the polarity of the INT output pin.
1 = Active-high.
0 = Active-low.

Wybór banku w rejestrze konfiguracyjnym IOCON ustala adresy rejestrów expander dla portów A i B

Najprostszy tryb pracy nie wymaga żadnych ustawień w rejestrze konfiguracyjnym IOCON (defaultowy jest bank 0).

Address IOCON.BANK = 1	Address IOCON.BANK = 0	Access to:
00h	00h	IODIRA
10h	01h	IODIRB
01h	02h	IPOLA
11h	03h	IPOLB
02h	04h	GPINTENA
12h	05h	GPINTENB
03h	06h	DEFVALA
13h	07h	DEFVALB
04h	08h	INTCONA
14h	09h	INTCONB
05h	0Ah	IOCON
15h	0Bh	IOCON
06h	0Ch	GPPUA
16h	0Dh	GPPUB
07h	0Eh	INTFA
17h	0Fh	INTFB
08h	10h	INTCAPA
18h	11h	INTCAPB
09h	12h	GPIOA
19h	13h	GPIOB
0Ah	14h	OLATA
1Ah	15h	OLATB

Wymagania projektowe

//definicje rejestrów wymaganych w projekcie (definicje własne, niekonieczne)

#define IODIRA 0x00 //kierunkowy portu A

#define IODIRB 0x01 //kierunkowy portu B

#define IOCON 0x0A //konfiguracyjny

#define GPPUA 0x0C //podwieszenie linii portu A (gdy są wejściami)

#define GPPUB 0x0D //podwieszenie linii portu B (gdy są wejściami)

#define GPIOA 0x12 //sterowanie liniami portu A (dla wyjść), odczyt linii (dla wejść)

#define GPIOB 0x13 //sterowanie liniami portu A (dla wyjść), odczyt linii (dla wejść)

Wymagane ustawienia wstępne:

- Konfiguracja linii magistrali (SPI: MOSI, MISO, SCK, SS, I2C: SDA, SCL)
- Nadanie kierunku liniom portu A, B
- Podwieszenie linii portów A i B (dla wejść)

Czynności w pętli:

- Cykliczny odczyt wejść oraz sterowanie wyjściami przez rejestry GPIOA oraz GPIOB

