Interfacing with other devices

Lectorial 5

Data transfer between devices

- Parallel data transfer
  - Allows many bits of data to be transmitted simultaneously
  - Requires more pins
  - Usually synchronous
- Serial data transfer
  - Only one bit of data to be transmitted at a time
  - Uses less pins
  - Can be synchronous or asynchronous

Data transfer interfaces available on the ATmega328P

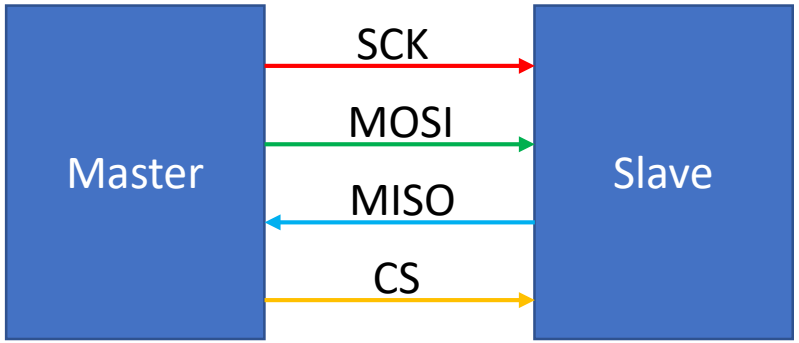
- Serial Peripheral Interface (SPI)
- Inter-Integrated Circuit (I^2C, I2C, IIC)
- Universal asynchronous receiver-transmitter (UART)

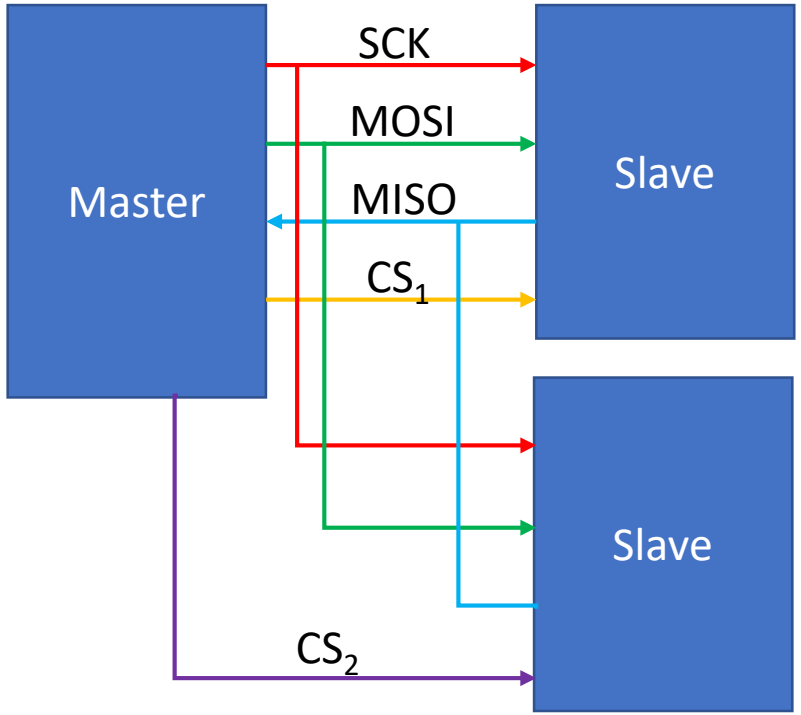
Serial Peripheral Interface (SPI)

- Synchronous – a clock signal determines when a bit of data is sent
- Uses a master/slave configuration where the master device provides clock signal and controls the slave device(s)
- Full duplex communication – simultaneous data transfer between master and slave

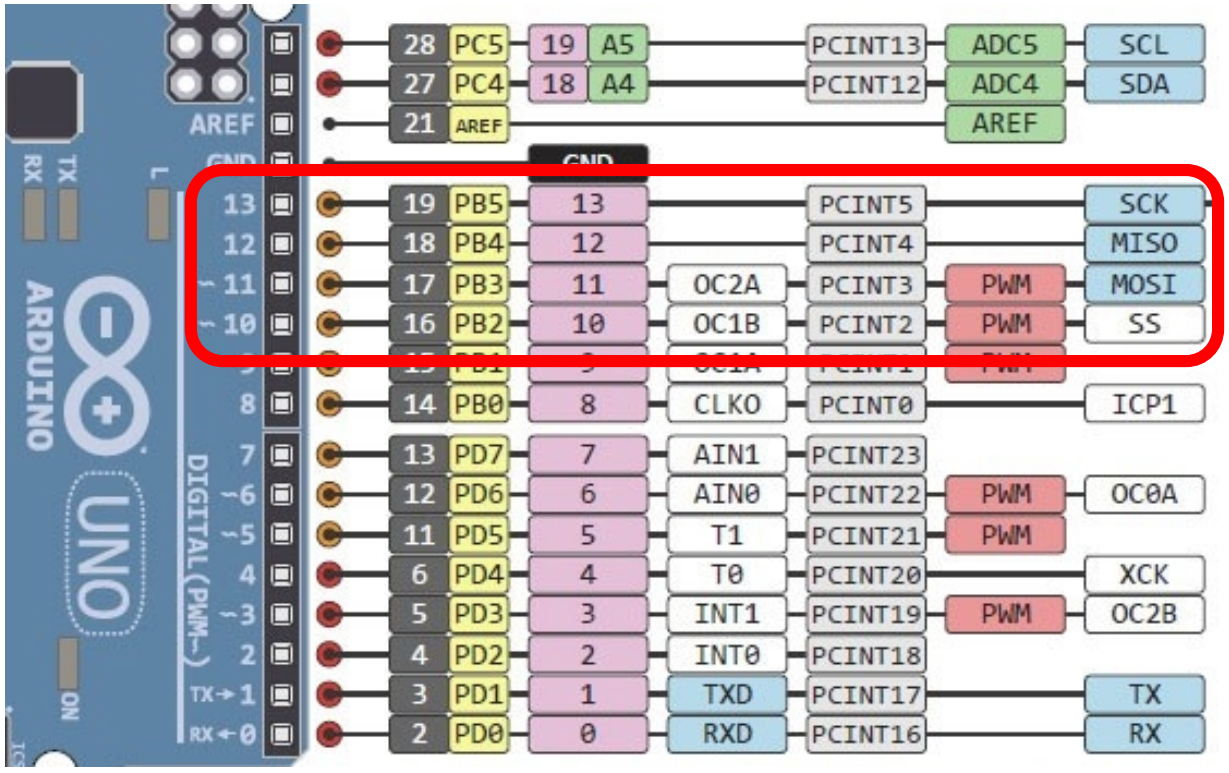
SPI configuration

- SCK – clock
- MOSI (master out slave in) – data transmission line from master to slave
- MISO (master in slave out) – data transmission line from slave to master
- CS/SS (chip/slave select) – line for master to select which slave it is communicating with (usually active low)





ATmega328P pins



Setting up SPI interface

- Set up Port B data direction register appropriately
- No pullup resistor for input
- Chip select signal(s) should be initialised high if acting as Master
- Set up SPI control register (SPCR)

19.5.1 SPCR – SPI Control Register

Bit	7	6	5	4	3	2	1	0	
0x2C (0x4C)	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	SPCR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – SPIE: SPI Interrupt Enable**

This bit causes the SPI interrupt to be executed if SPIF bit in the SPSR Register is set and the if the Global Interrupt Enable bit in SREG is set.

- **Bit 6 – SPE: SPI Enable**

When the SPE bit is written to one, the SPI is enabled. This bit must be set to enable any SPI operations.

- **Bit 5 – DORD: Data Order**

When the DORD bit is written to one, the LSB of the data word is transmitted first.

When the DORD bit is written to zero, the MSB of the data word is transmitted first.

- **Bit 4 – MSTR: Master/Slave Select**

This bit selects Master SPI mode when written to one, and Slave SPI mode when written logic zero. If $\overline{SS}$ is configured as an input and is driven low while MSTR is set, MSTR will be cleared, and SPIF in SPSR will become set. The user will then have to set MSTR to re-enable SPI Master mode.

SPI clock rate (SPR1, SPR0)

19.5.1 SPCR – SPI Control Register

Bit	7	6	5	4	3	2	1	0	
0x2C (0x4C)	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	SPCR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 1, 0 – SPR1, SPR0: SPI Clock Rate Select 1 and 0**

These two bits control the SCK rate of the device configured as a Master. SPR1 and SPR0 have no effect on the Slave. The relationship between SCK and the Oscillator Clock frequency f_{osc} is shown in the following table:

Table 19-5. Relationship Between SCK and the Oscillator Frequency

SPI2X	SPR1	SPR0	SCK Frequency
0	0	0	$f_{osc}/4$
0	0	1	$f_{osc}/16$
0	1	0	$f_{osc}/64$
0	1	1	$f_{osc}/128$
1	0	0	$f_{osc}/2$
1	0	1	$f_{osc}/8$
1	1	0	$f_{osc}/32$
1	1	1	$f_{osc}/64$

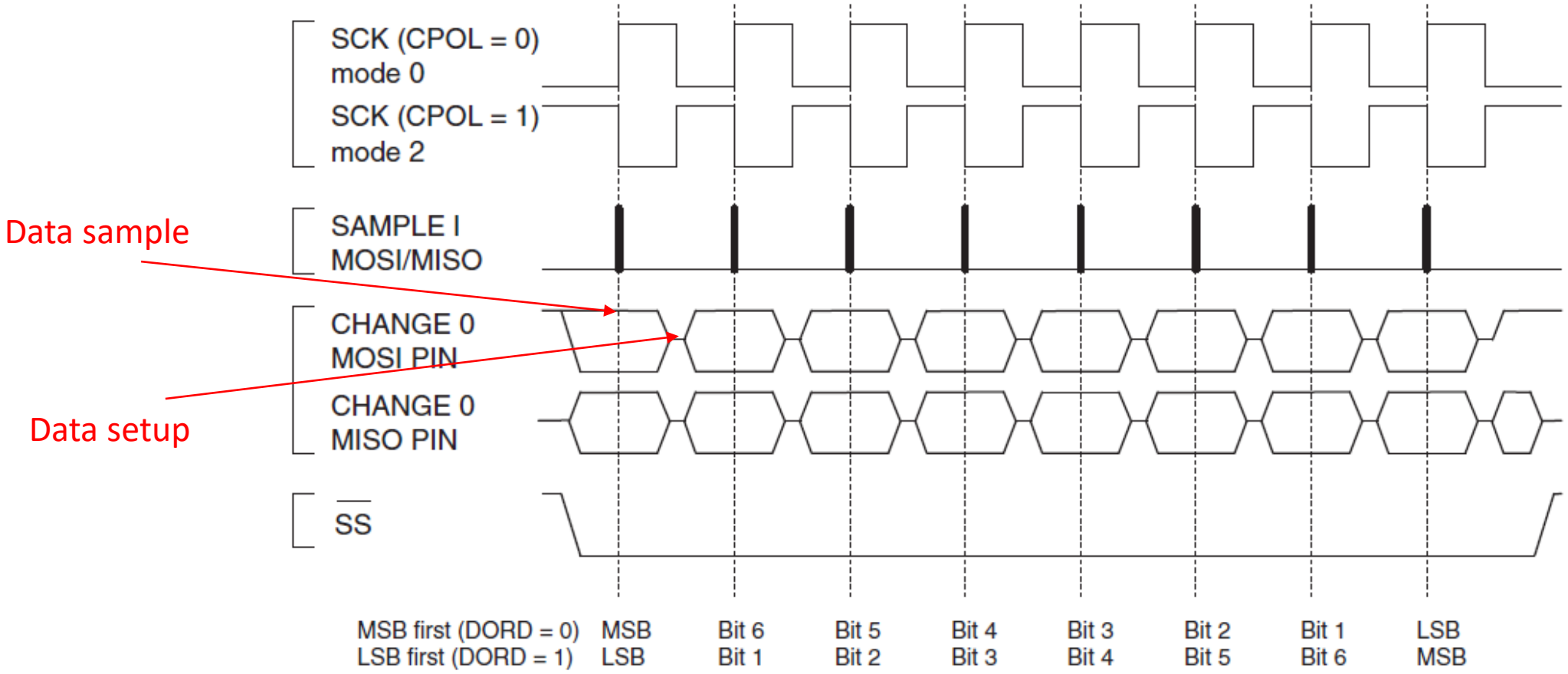
CPOL/CPHA

- Two clock signal features determine mode of operation
  1. Clock Polarity (CPOL)
     - Refers to idle (non-transmitting) state of clock signal (either low or high)
  2. Clock Phase (CPHA)
     - Refers to edge of clock signal when data is sampled (either rising or falling edge)

SPI Mode	CPOL/CPHA	Data Sample	Data setup
0	CPOL = 0, CPHA = 0	Rising Edge	Falling Edge
1	CPOL = 0, CPHA = 1	Falling Edge	Rising Edge
2	CPOL = 1, CPHA = 0	Falling Edge	Rising Edge
3	CPOL = 1, CPHA = 1	Rising Edge	Falling Edge

SPI Mode	CPOL/CPHA	Data Sample	Data setup
0	CPOL = 0, CPHA = 0	Rising Edge	Falling Edge
2	CPOL = 1, CPHA = 0	Falling Edge	Rising Edge

Figure 19-3. SPI Transfer Format with CPHA = 0



How to set up SPCR?

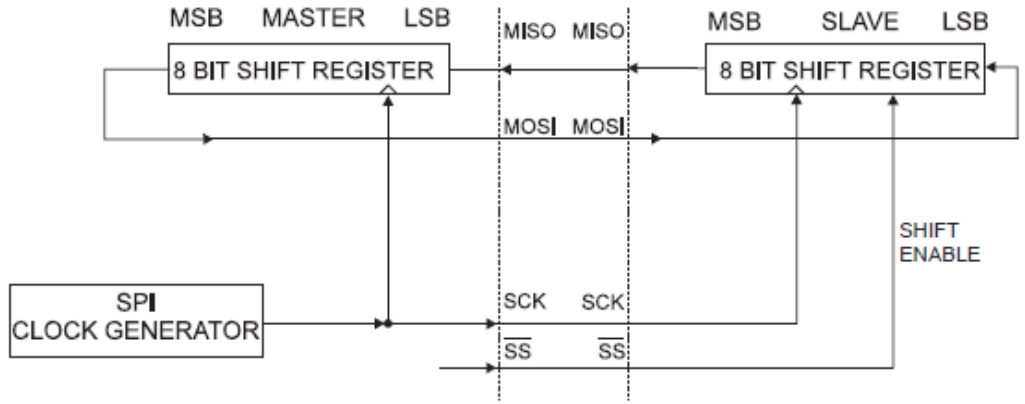
- Depends on what is supported by connected device
- In this unit, you will connect to the MCP23S17 Port Expander via SPI
  - Use mode 0 for CPOL/CPHA
  - Read the MCP23S17 datasheet to determine:
    - Whether the ATmega328P should be master or slave
    - Which bit to send first

SPI Data and Status Registers

- SPDR is the data register
  - This is where you put data for sending and read from when data is received
- SPSR is the status register
  - In normal operation, SPIF flag will indicate whether data is available for reading, or SPDR is available for writing
  - If data is written to SPDR during data transfer, the WCOL bit will be set

Transferring/Receiving data

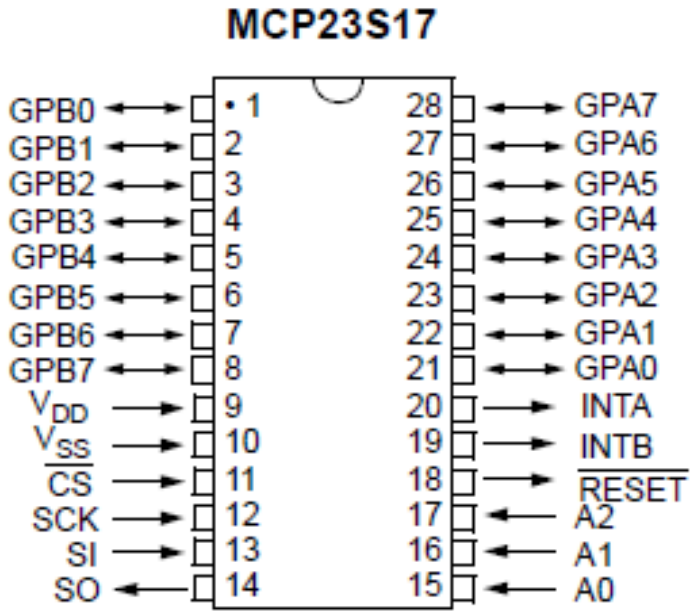
Figure 19-2. SPI Master-slave Interconnection



- Write 8-bits to SPDR (data if transferring/0 if receiving)
- Wait until SPIF goes high
- Read SPDR

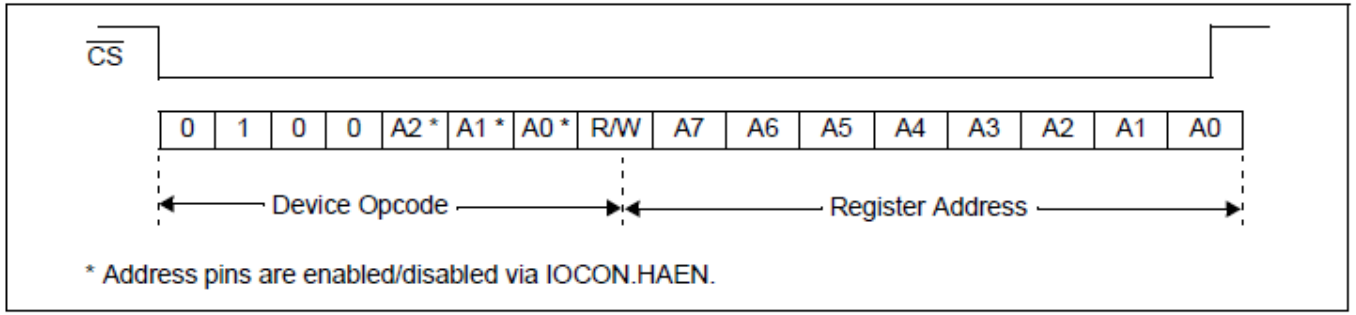
MCP23S17 Port Expander

- Two 8-bit GPIO ports (PORT A, PORT B)
- INTA and INTB are interrupts generated from PORT A and B, respectively
- V_{DD} connect to 5V, V_{SS} connect to GND
- SI – data input
- SO – data output
- A0, A1, A2 are used for addressing
  - Connect these to ground



Accessing MCP23S17 registers

FIGURE 3-7: SPI ADDRESSING REGISTERS



- You will need to send three bytes
  - Byte 1 = Opcode (0b0100000{R/W})
    - R/W – 0 for write, 1 for read
  - Byte 2 = Register Address
  - Byte 3 = Data
    - Send 0 if reading from register
- Send bytes one at a time because SPDR is 8-bit only

REGISTER ADDRESSES

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

Pin input/output

- IODIRA (0x00), IODIRB (0x01) determines whether a pin is an input or output
  - 1 = input, 0 = output (this is opposite to ATmega328P)
- GPPUA (0x0C), GPPUB (0x0D) controls pullup resistors
  - 1 = enable, 0 = disable
- Read input from GPIOA (0x12), GPIOB (0x13)
- Write output to OLATA (0x14), OLATB (0x15)

Example Psuedocode

Set PORT B pin 0 to 3 as output and pin 4 to 7 as input

- Assume a function has already been written to handle sending and receiving 8 bits of data via SPI (see slide on transferring/receiving data):

```
uint8_t spi_transfer(uint8_t data);
```

1. Set chip select line low
2. spi_transfer(0x40) // Opcode + write bit
3. spi_transfer(0x01) // register address of IODIRB
4. spi_transfer(0xf0) // set direction of pins
5. Set chip select line high

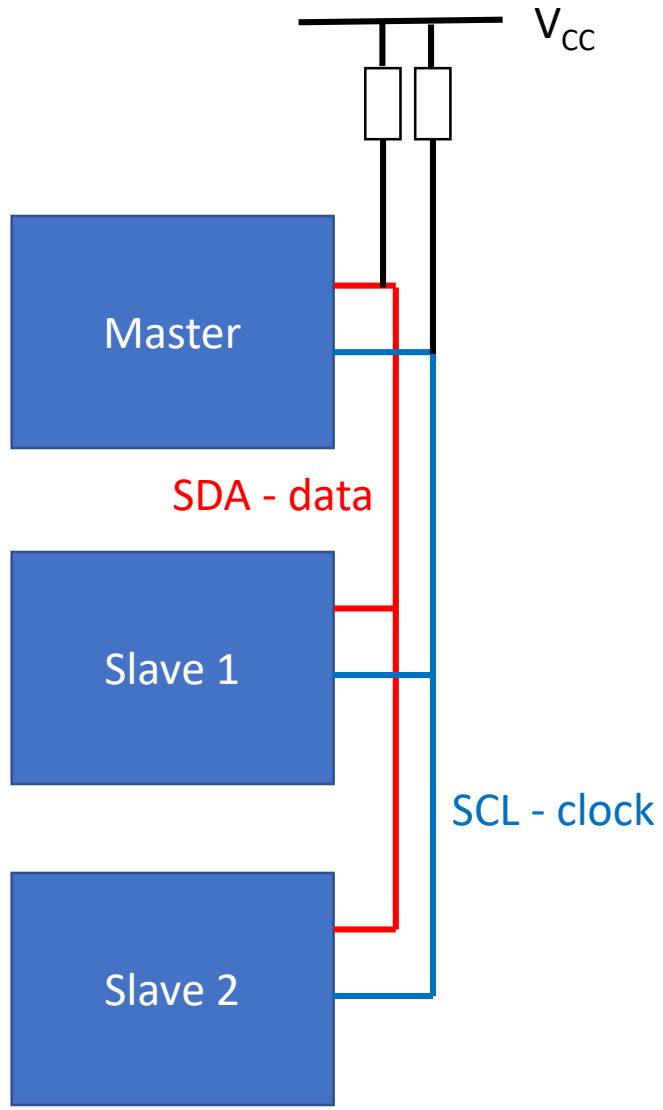
Example Psuedocode

Read values in PORT B

1. Set chip select line low
2. spi_transfer(0x41) // Opcode + read bit
3. spi_transfer(0x13) // register address of GPIOB
4. portb_values = spi_transfer(0) // send 0s to get values
5. Set chip select line high

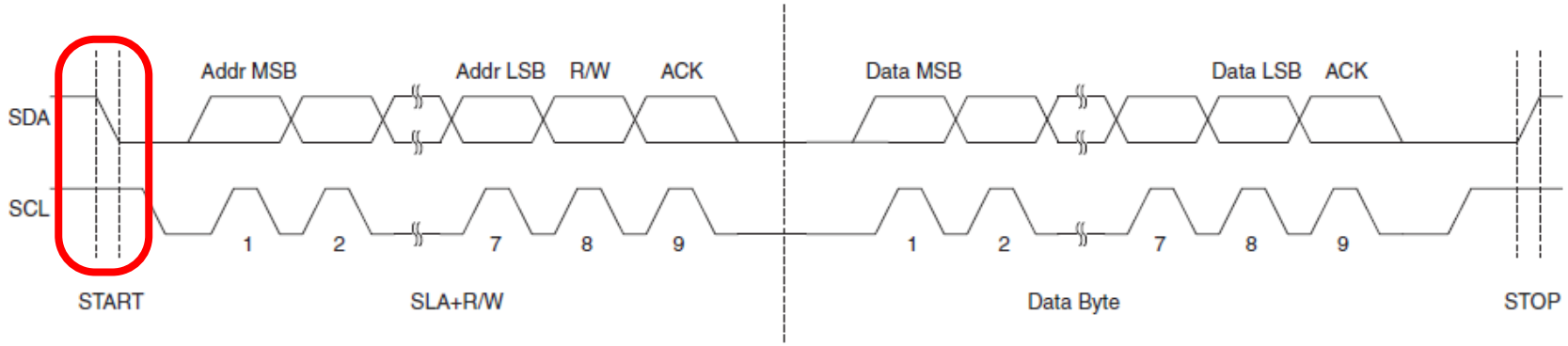
Inter-Integrated Circuit (I^2C, I2C, IIC)

- Called 2-Wire Serial Interface in datasheet
- Synchronous
- Requires only two wires and pull up resistors
- Allows communication between multiple masters and slaves on the same bus
  - Uses unique 7-bit address for each device
  - Masters cannot communicate with each other on the same bus
- Half-duplex – communication can only occur in one direction at a time



I^2C communications

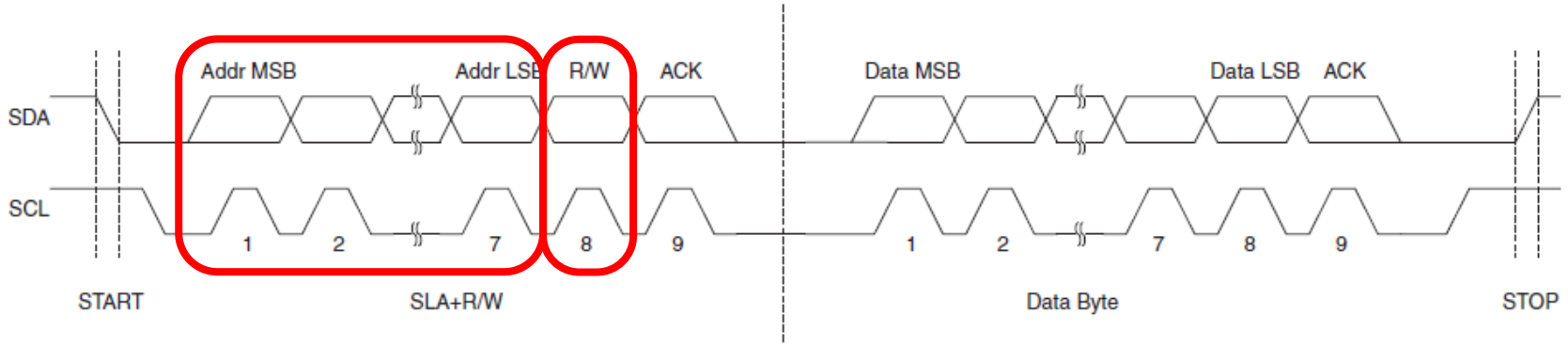
Figure 22-6. Typical Data Transmission



- Master generates start condition on bus
  - Pulls SDA low while SCL high
- If two masters tries to take ownership of the bus, the first device to pull SDA low gains control

I²C communications

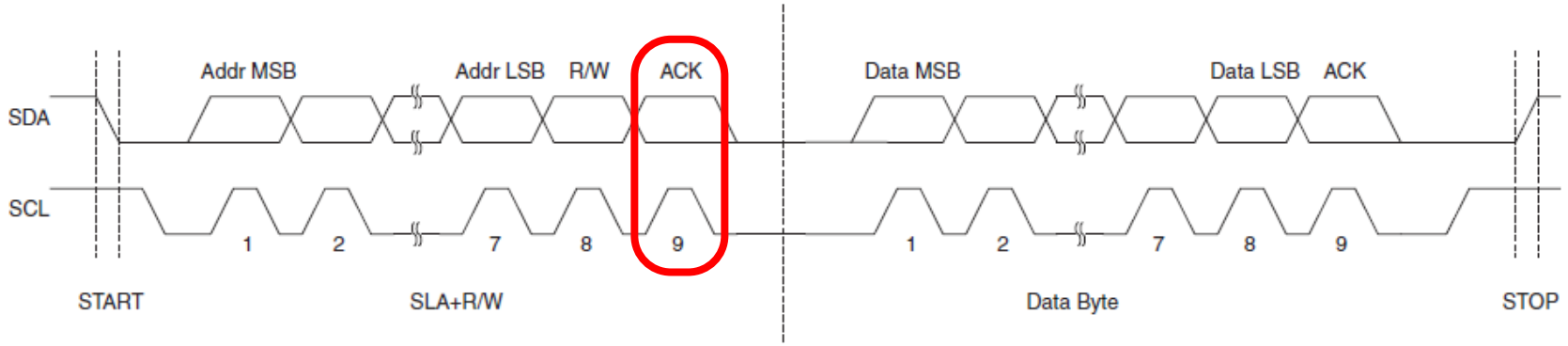
Figure 22-6. Typical Data Transmission



- Send 7-bit slave address, MSB first
- Follow by R/W bit indicating whether this is a read (1) or write (0) operation
- During data transfer, SDA should not change value when SCL is high

I²C communications

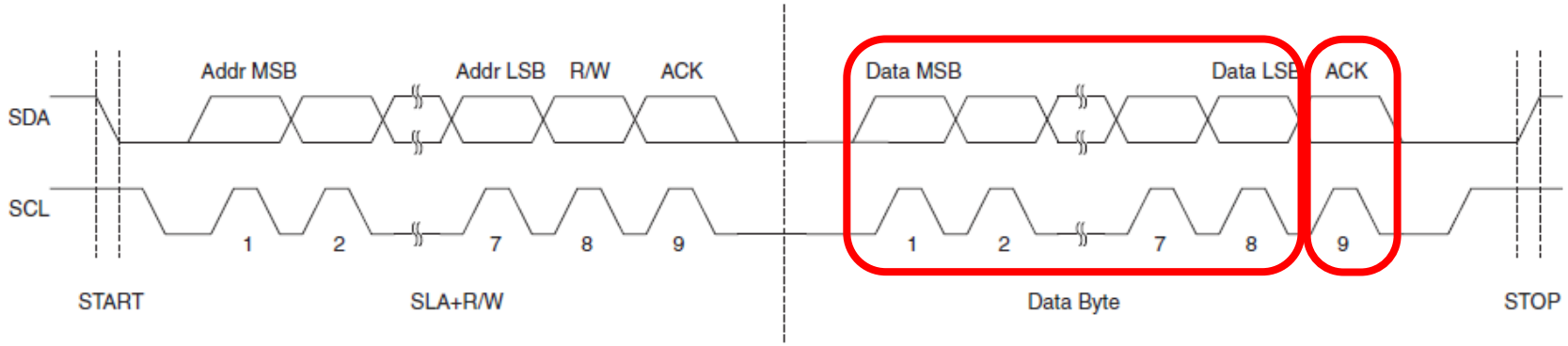
Figure 22-6. Typical Data Transmission



- Receiving device is given control of SDA (not SCL)
- Receiving device will pull down SDA if message received and understood (ACK)
- If no ACK (NACK) bit received (SDA remains high), data exchange is stopped
  - Master needs to decide what to do next

I²C communications

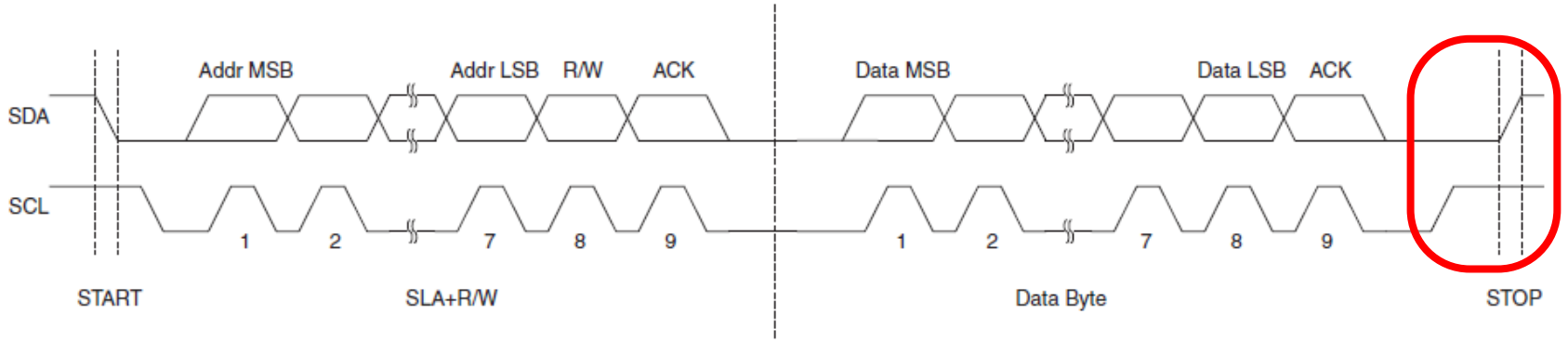
Figure 22-6. Typical Data Transmission



- If ACK received, data is sent
- Data can be any number of bits
- For every 8 data bits, the transmitter will wait for the ACK/NACK bit from the receiver before continuing transmission
- MSB is transmitted first

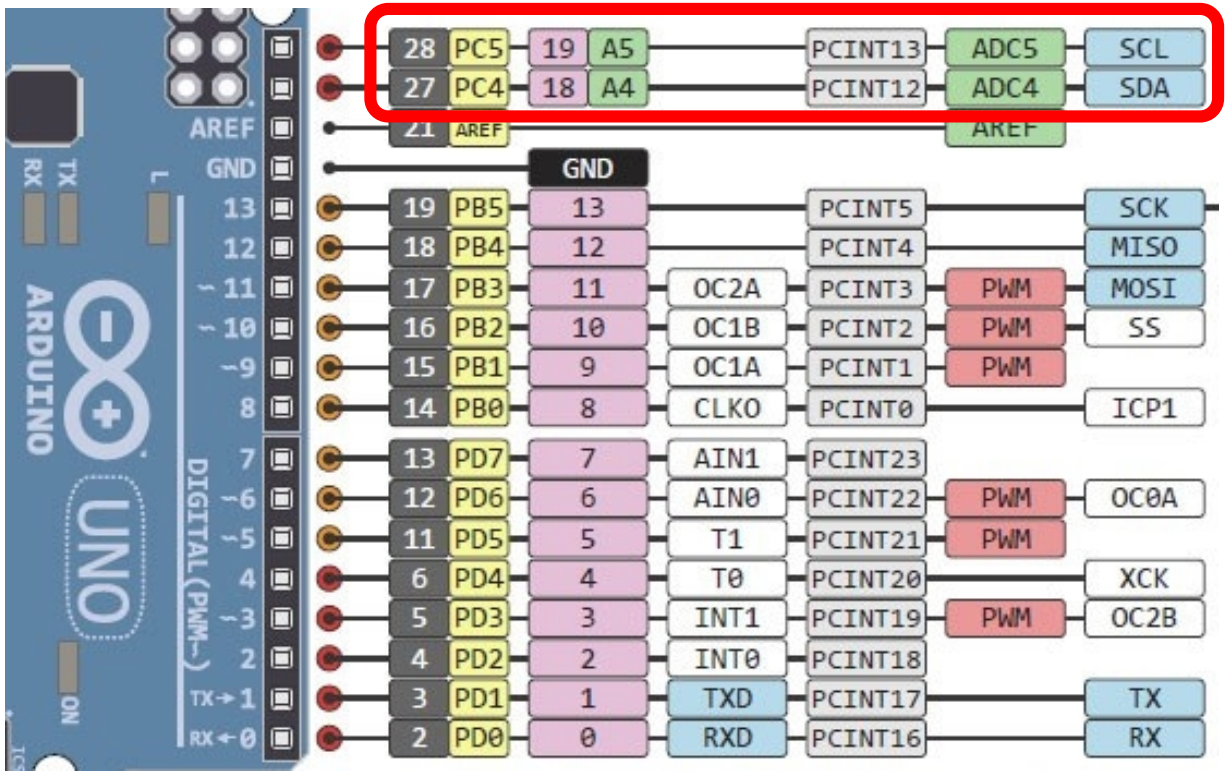
I²C communications

Figure 22-6. Typical Data Transmission



- Transmission is stopped by master creating stop condition
  - Positive transition on SCL followed by positive transition on SDA while SCL remains high

ATmega328P pins



Configuring for I^2C communications

- Set Port C pins 4 & 5 as inputs with pullup resistors
- Set data transmission rate (SCL frequency)
- Calculate TWBR value by choosing prescaler value and setting SCL frequency to be ~40 kHz
  - This is slow enough to allow for interference of jumper wires
  - Can go faster if using printed circuitry and shielded

$$\text{SCL frequency} = \frac{\text{CPU Clock frequency}}{16 + 2(\text{TWBR}) \cdot (\textit{PrescalerValue})}$$

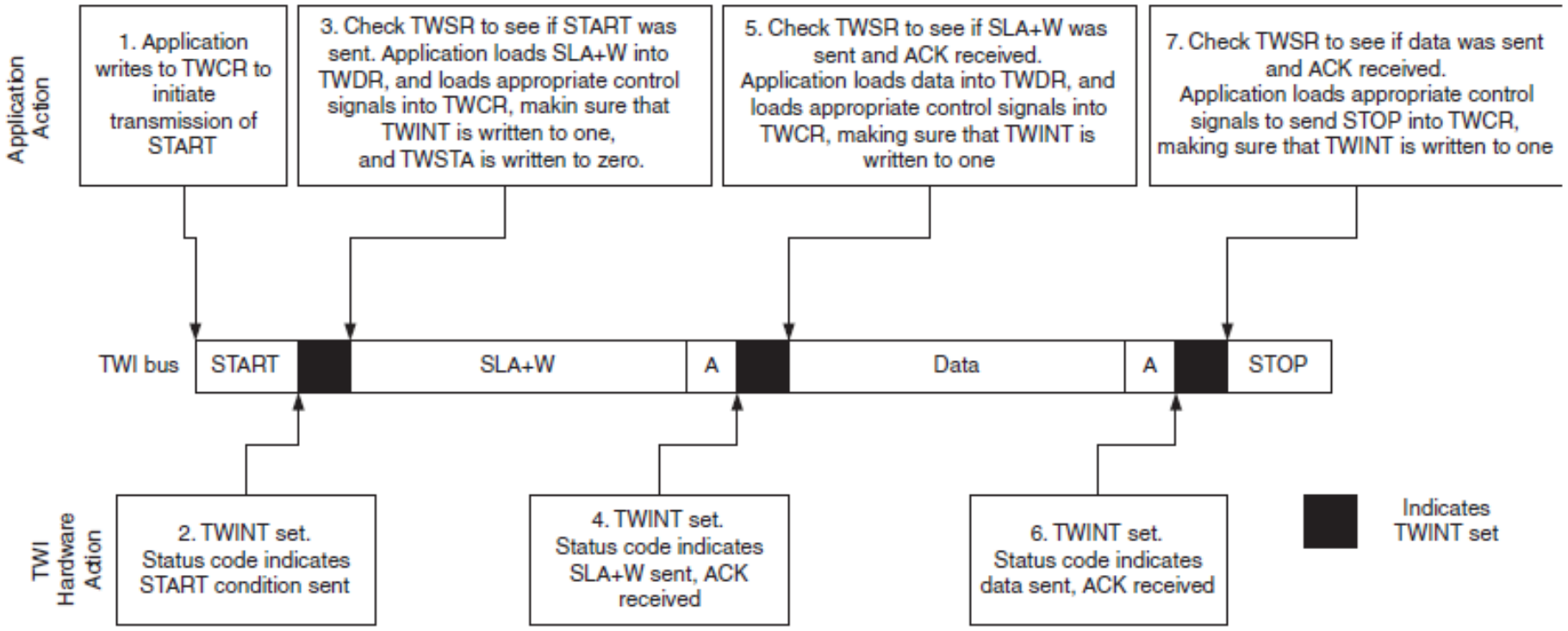
- TWBR = Value of the TWI Bit Rate Register.
- *PrescalerValue* = Value of the prescaler, see Table 22-7 on page 241.

Table 22-7. TWI Bit Rate Prescaler

TWPS1	TWPS0	Prescaler Value
0	0	1
0	1	4
1	0	16
1	1	64

Typical interaction with I²C hardware

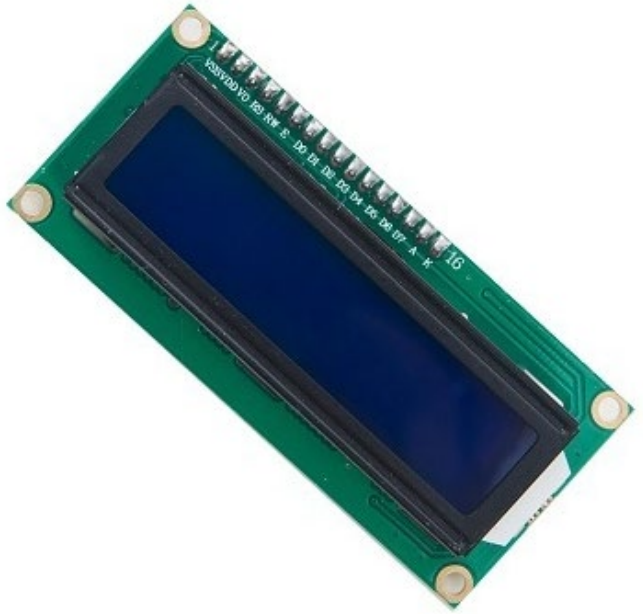
Figure 22-10. Interfacing the Application to the TWI in a Typical Transmission



- TWINT flag is set after every bus events (can also generate an interrupt)

I^2C LCD display

- Consists of HD44780 based character LCD display and I2C LCD backpack
- Displays 2 rows of 16 characters
- Depending on which PCF8574 chip is on the backpack, the I2C address will be different
  - 0x27 for Texas Instrument
  - 0x3F for NXP Semiconductors



Datasheets

- PCA8574_PCA8574A.pdf
  - PCF8574 & PCA8574 are equivalent
  - for understanding I2C communications between ATmega328p & backpack
- HD44780.pdf
  - for understanding how to display characters on LCD display

PCF8575

- Converts serial I^2C data to 8 parallel bits

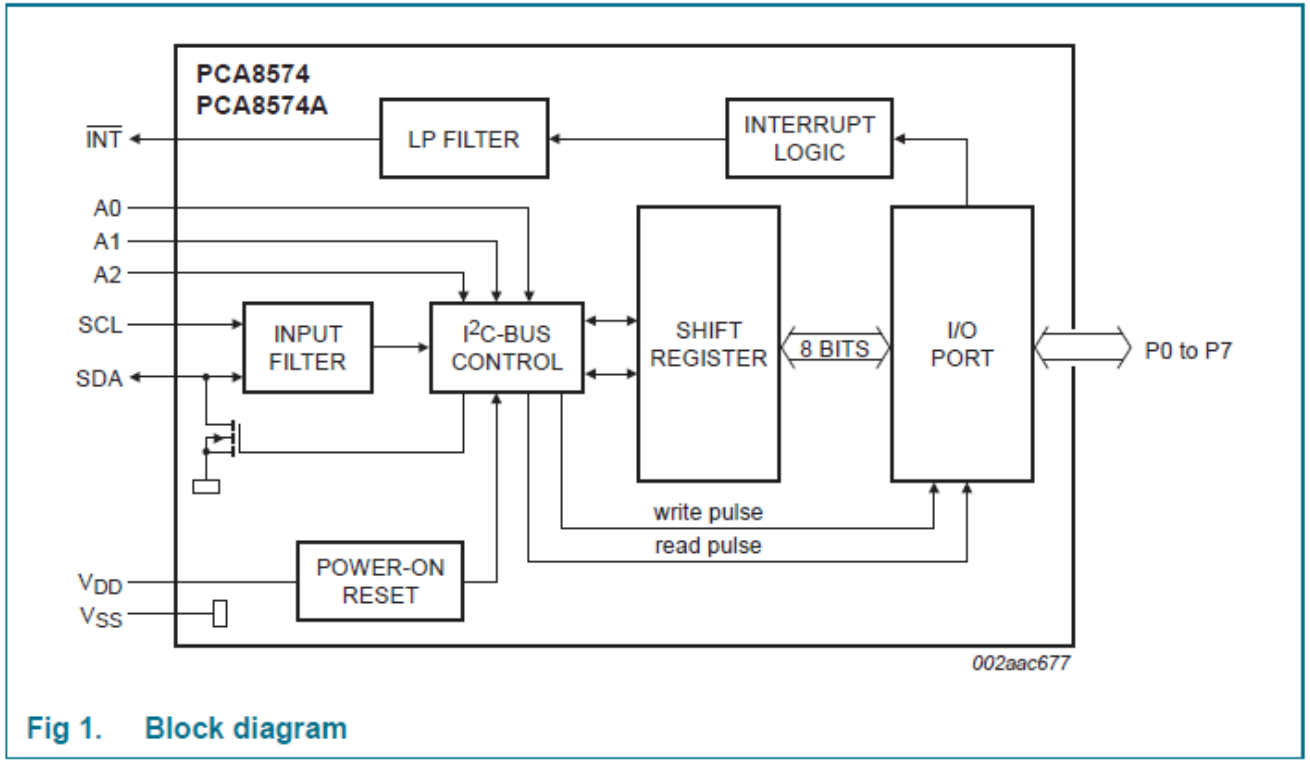


Fig 1. Block diagram

HD44780U communications

- Can work in 8-bit or 4-bit mode
- PCF8575 limits us to 4-bit mode. Higher 4-bits used for data, lower 4-bits used for RS, R/W, E, and backlight
- For 4-bit mode, the 8-bit operations is split and four most significant bits are sent first

Need to create negative transition to latch data

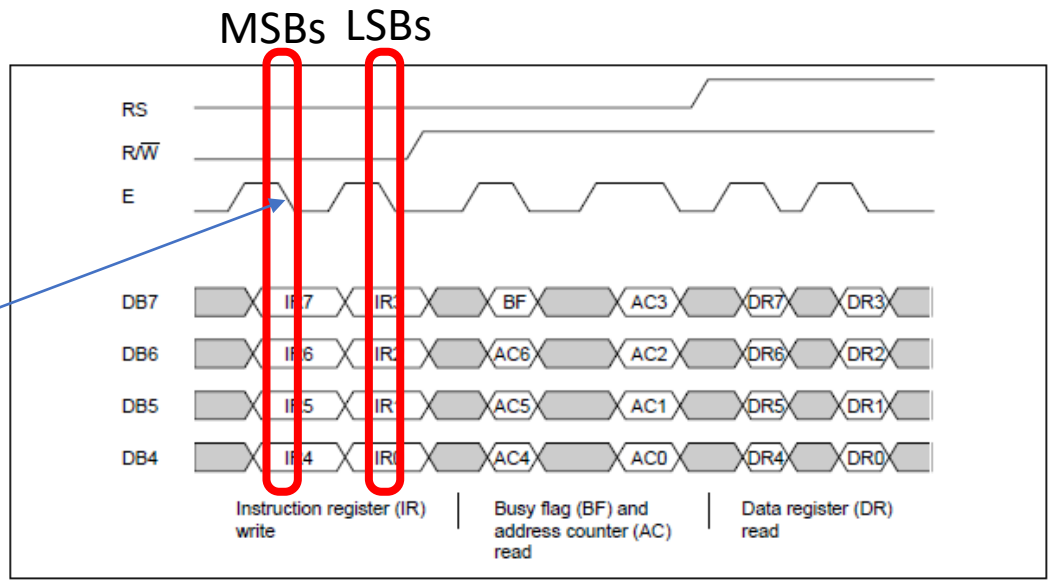


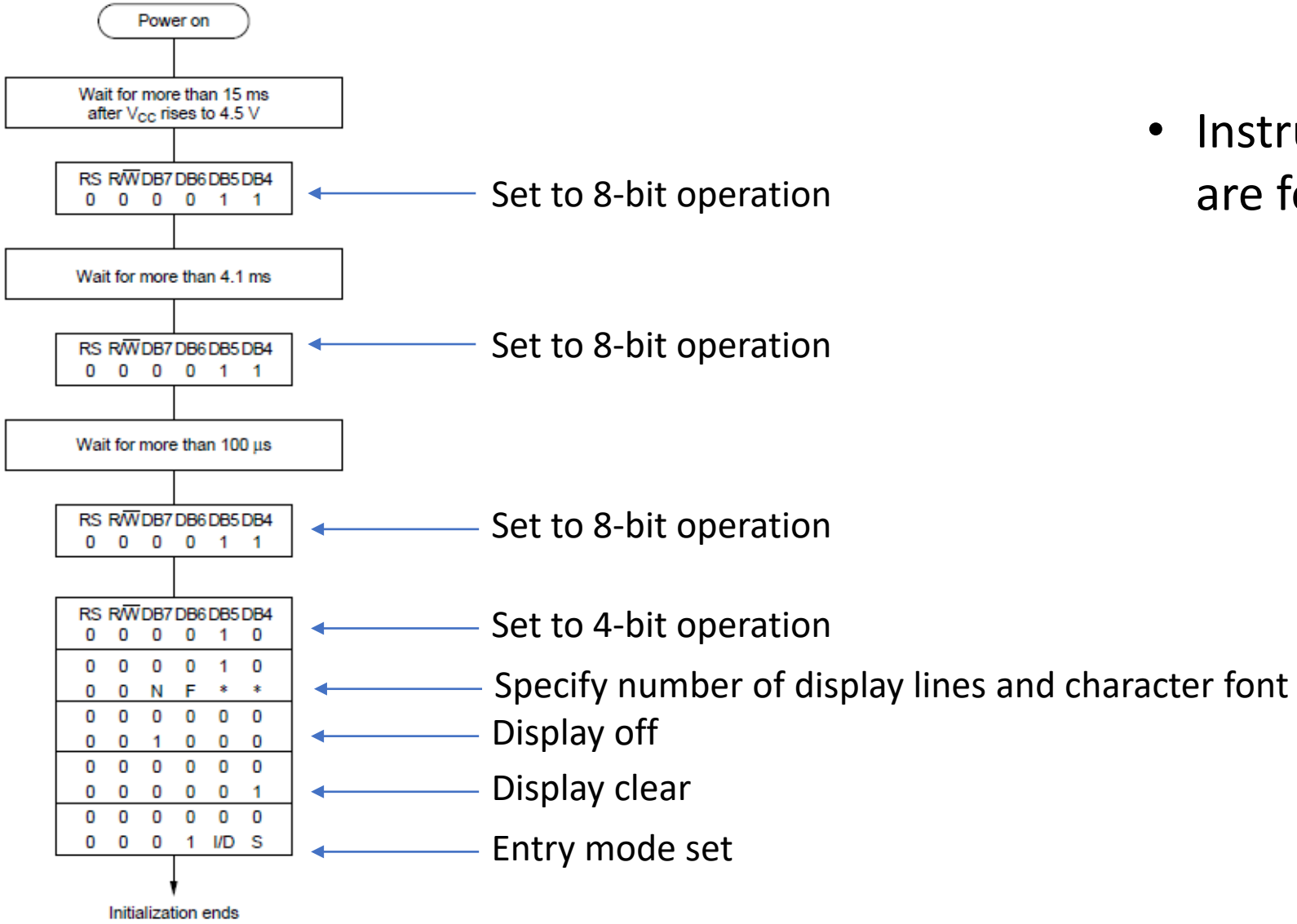
Figure 9 4-Bit Transfer Example

RS (register select)
0 = instruction register
1 = data register
R/W (read/write)
0 = write
1 = read
E (enable)
DB7:DB4 (data bits)

HD44780 registers

- Has two 8-bit registers
  - Instruction register – stores instruction codes; e.g. display clear, cursor shift and address information for display data RAM (DDRAM) and character generator RAM (CGRAM)
  - Data register – temporarily stores data to be written into (or read from) DDRAM and CGRAM
- DDRAM – corresponds to positions on display
- CGRAM – consists of character generator ROM and RAM
  - ROM stores permanent fonts that can be displayed using corresponding ASCII code
  - RAM provides memory for user-defined characters

HD44780 Initialisation (Fig. 24)



- Instruction code definitions are found in Table 6

Writing character(s) to the display

1. Set cursor position using DDRAM address
   - Note starting address of second row

Display position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
DDRAM address	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F

2. Send ASCII code of character(s)

Table 6 Instructions

	Code										
Instruction	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Description
Clear display	0	0	0	0	0	0	0	0	0	1	Clears entire display and sets DDRAM address 0 in address counter.
Return home	0	0	0	0	0	0	0	0	1	—	Sets DDRAM address 0 in address counter. Also returns display from being shifted to original position. DDRAM contents remain unchanged.
Entry mode set	0	0	0	0	0	0	0	1	I/D	S	Sets cursor move direction and specifies display shift. These operations are performed during data write and read.
Display on/off control	0	0	0	0	0	0	1	D	C	B	Sets entire display (D) on/off cursor on/off (C), and blinking of cursor position character (B).
Cursor or display shift	0	0	0	0	0	1	S/C	R/L	—	—	Moves cursor and shifts display without changing DDRAM contents.
Function set	0	0	0	0	1	DL	N	F	—	—	Sets interface data length (DL), number of display lines (N), and character font (F).
Set CGRAM address	0	0	0	1	ACG	ACG	ACG	ACG	ACG	ACG	Sets CGRAM address. CGRAM data is sent and received after this setting.
Set DDRAM address	0	0	1	ADD	ADD	ADD	ADD	ADD	ADD	ADD	Sets DDRAM address. DDRAM data is sent and received after this setting.

Questions about minor project?