EN2533 Robot Design & Competition

Robot Communication

Peshala Jayasekara, PhD

Dept. of Electronic and Telecommunication Engineering

Faculty of Engineering

University of Moratuwa

Outline

- Robot Digital Communication
  - Universal Asynchronous Receiver Transmitter (UART)
  - Serial Peripheral Interface (SPI)
  - Inter-Integrated Circuit (I2C)

Introduction

Introduction

- Inter device communication occurs over digital signals using different communication methods
  - Data is transferred as 0s and 1s

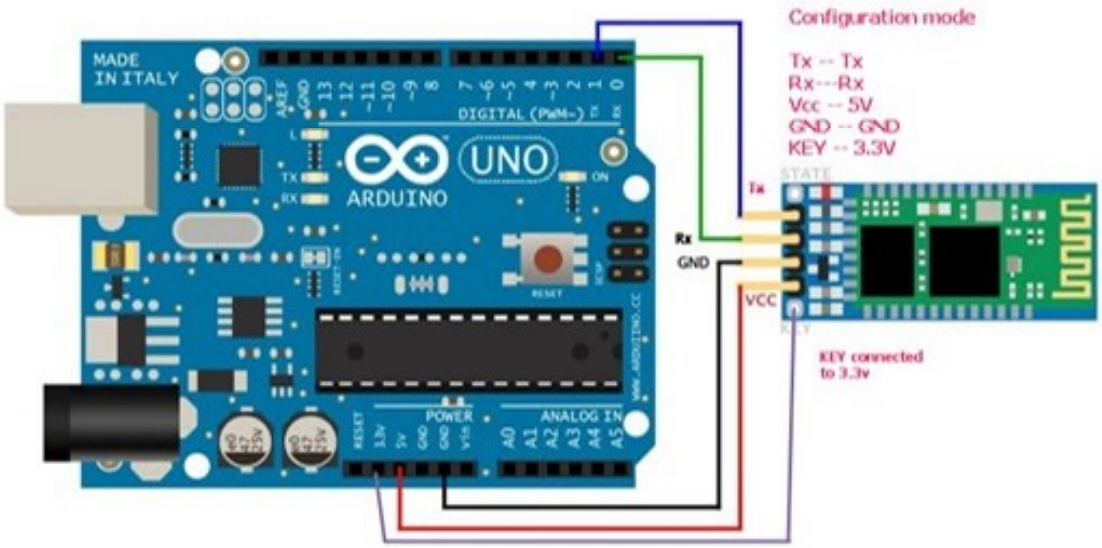
- Protocols/Interfaces
  - Universal Asynchronous Receiver-Transmitter (UART)
  - Serial Peripheral Interface (SPI)
  - Inter-integrated Circuit (I2C) / Two Wire Interface (TWI)

differ in their implementation, but ultimately serve the same purpose: transferring data at high speeds to any compatible device.

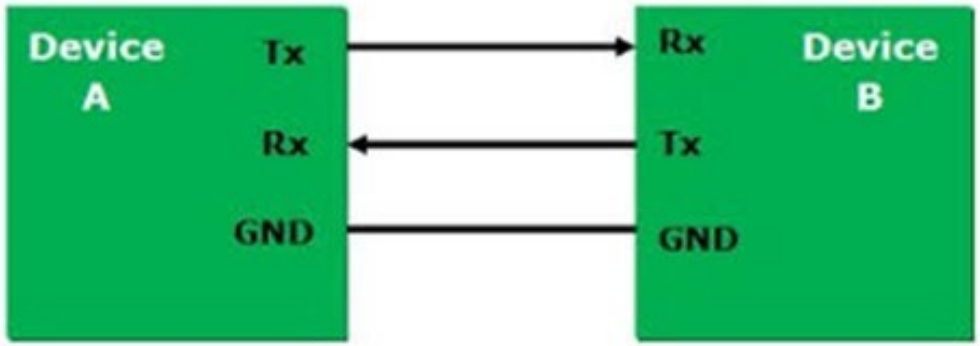
Universal Asynchronous Receiver Transmitter (UART)

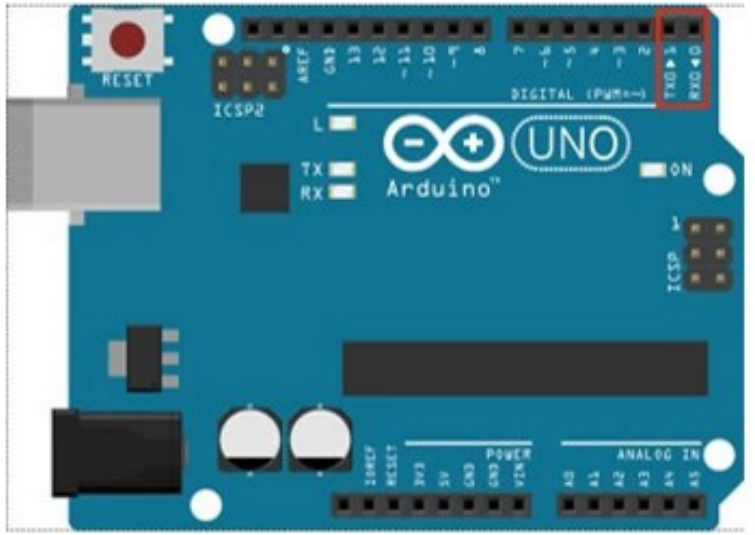
UART

- With UART, robots can be equipped with
  - LCD
  - Bluetooth wireless
  - Debug code
  - Test sensors etc.



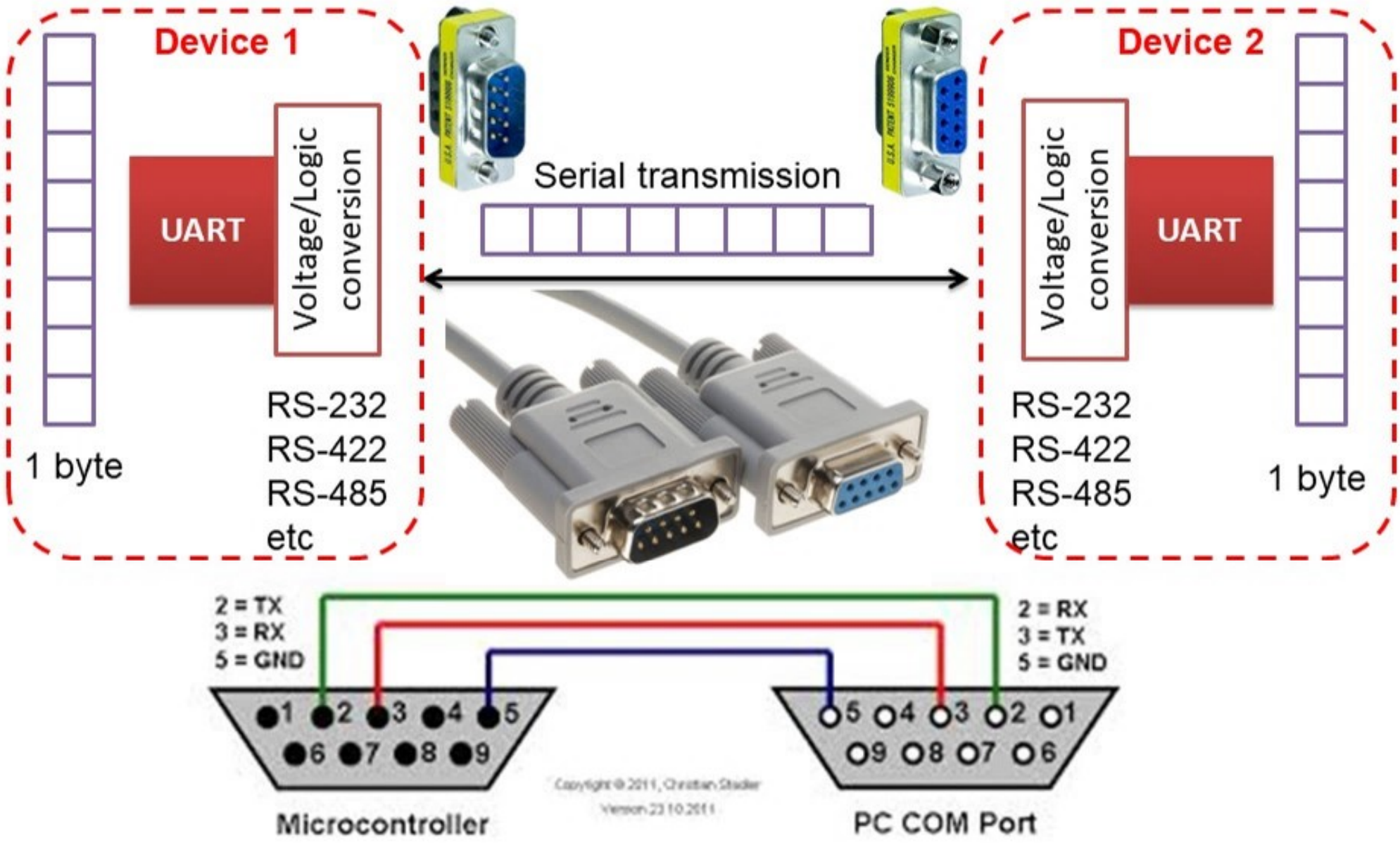
UART





- UART is actually a hardware that implements asynchronous serial communication
  - ➢ Either dedicated pins or any other pins using software serial
- Serial communication protocol
  - ➢ Takes bytes of data and transmits bits in a sequential fashion
- Asynchronous: No clock
  - ➢ Two devices need to agree on baud rate and data format

UART


Device 1
UART
Voltage/Logic conversion
1 byte
RS-232
RS-422
RS-485
etc
Serial transmission
Device 2
Voltage/Logic conversion
UART
RS-232
RS-422
RS-485
etc
1 byte
2 = TX
3 = RX
5 = GND
2 = RX
3 = TX
5 = GND
1 2 3 4 5
6 7 8 9
5 4 3 2 1
9 8 7 6
Microcontroller
PC COM Port

UART

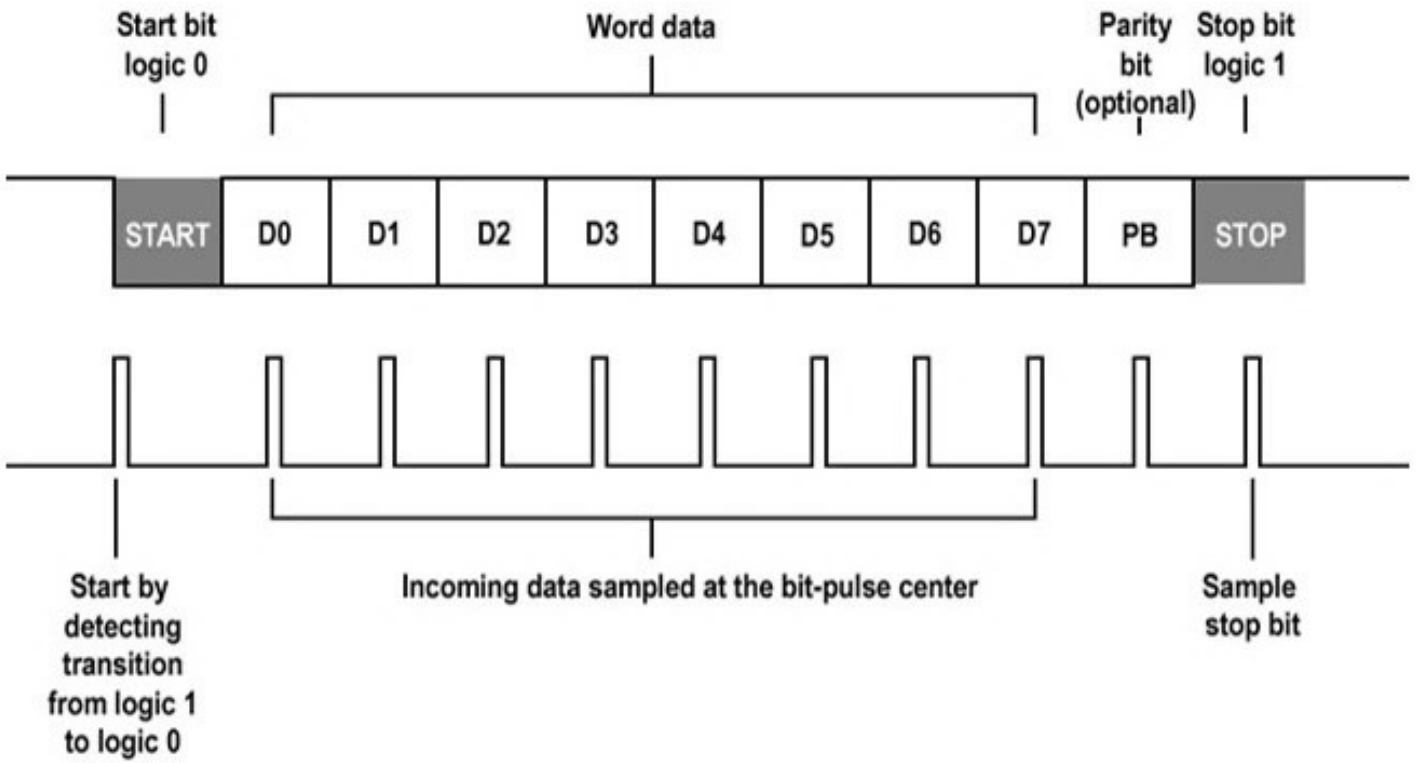
- RS232
  - ➢ Pinout
    1. DCD-data carrier detect
    2. Receive Data
    3. Transmit data
    4. Data terminal ready
    5. GND
    6. Data set ready
    7. Request to send
    8. Clear to send
    9. Ring indicator

Pin 1	DCD
Pin 2	RXD
Pin 3	TXD
Pin 4	DTR
Pin 5	GND
Pin 6	DSR
Pin 7	RTS
Pin 8	CTS
Pin 9	RI

Pins 1,4,6,7,8 & 9 (handshaking pins) are rarely used

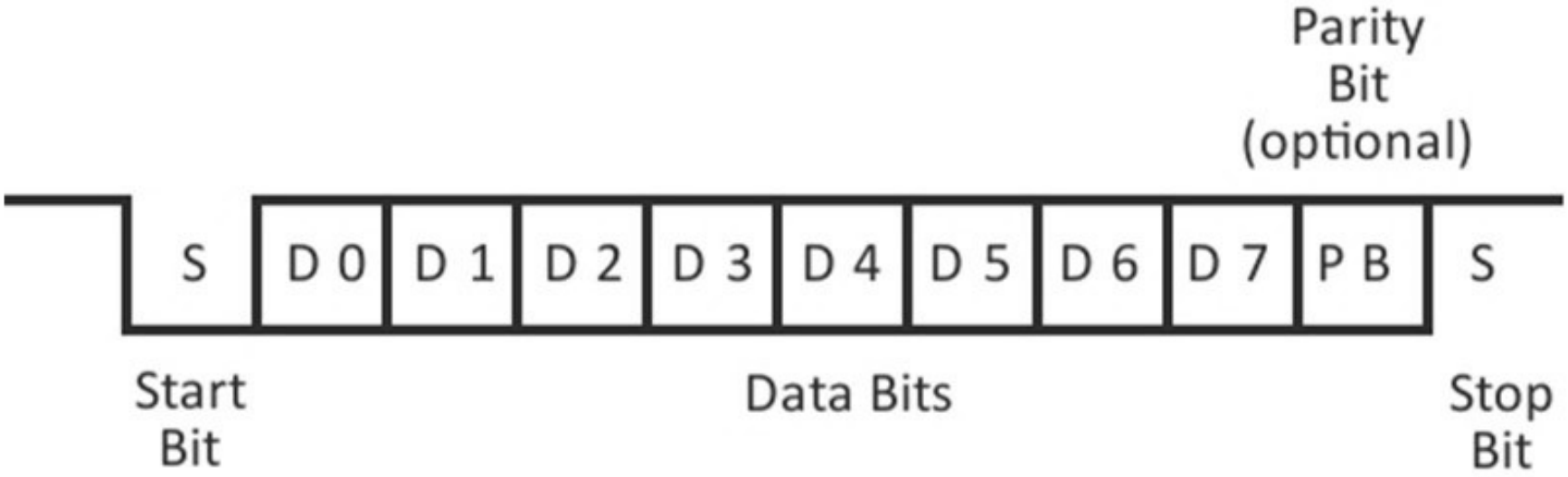
  - ➢ RS232 is 1-to-1

UART

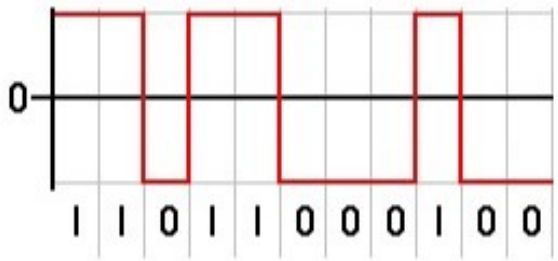


- RS232 Data Format
  - ➢ Start bit + data bits + parity bit (optional) + stop bit
    - Parity bit considers number of “high” data bits. For “odd” parity 10011100: parity bit = 1
    - E.g. 8N1: 8 data bits, no parity bit, 1 stop bit

UART



- RS232 Encoding
  - ➢ Non-Return To Zero is used

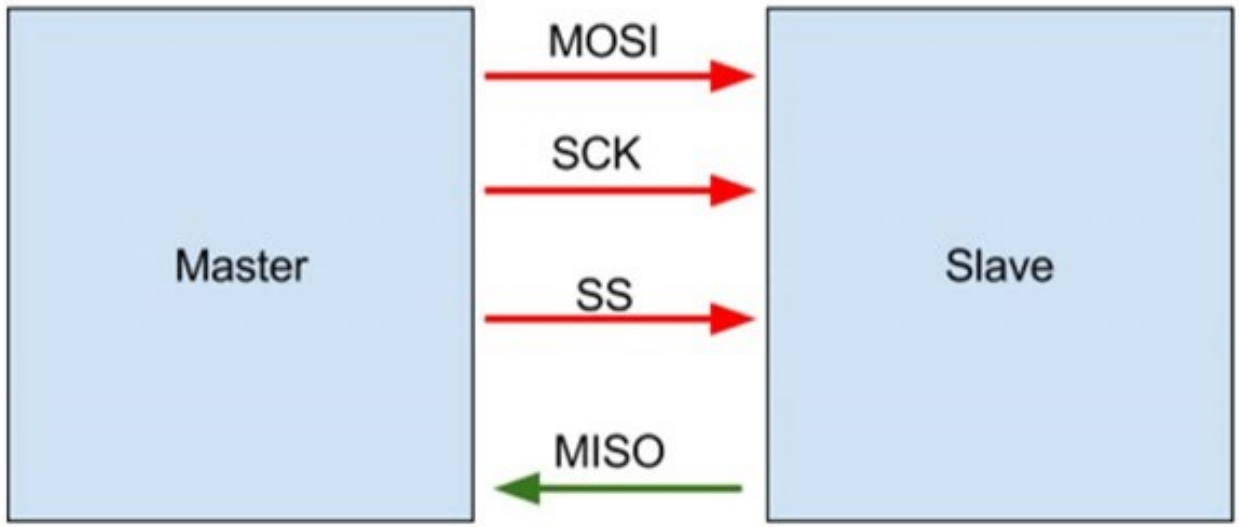


Return To Zero encoding will have a neutral or rest condition

Serial Peripheral Interface (SPI)

SPI

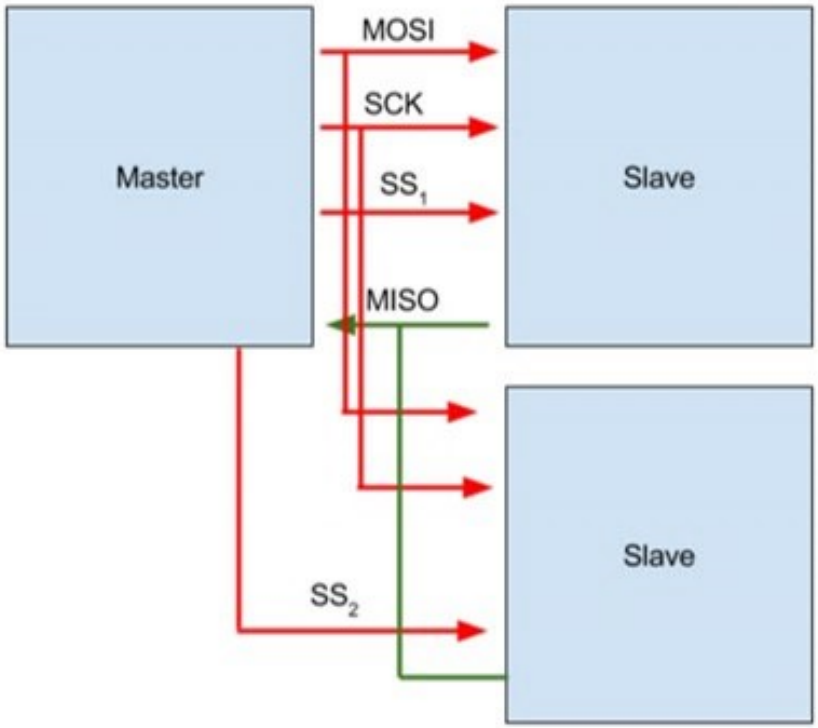
- Synchronous, full-duplex serial data com. protocol
- It follows a master-slave model
  - One master and multiple slave devices



MOSI ("Master Out Slave In"): Data transmission line from master to slave
SCK ("Clock"): Clock line defining transmission speed and transmission start/end characteristics
SS ("Slave Select"): Line for master to select a particular slave to communicate with
MISO ("Master In Slave Out"): Data transmission line from slave to master

SPI

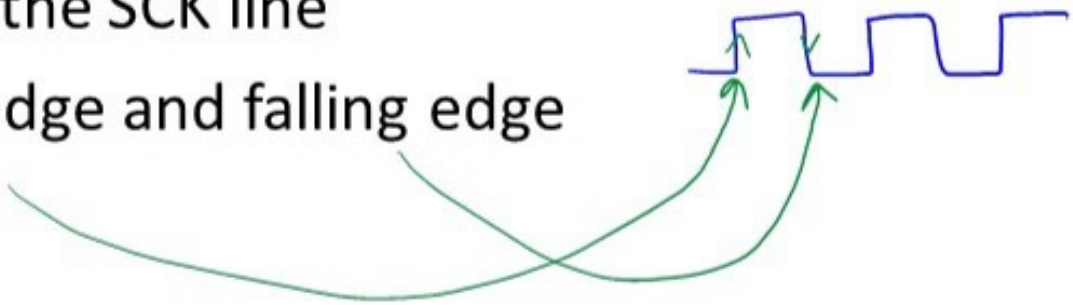
- Multiple slaves can be connected to a single master



- ➢ No separate Tx/Rx lines needed for each slave
  - Common MOSI, MISO and SCK lines
- ➢ Master device decides which slave it is communicating with through a separate SS line for each slave

SPI

- Clock signal
  - Square wave signal on the SCK line
  - Has two edges: rising edge and falling edge



- Clock Polarity CPOL
  - To conserve power, devices will put the clock line at the idle state when not communicating with any slaves
  - If CPOL=0
    - The clock will idle at 0
    - Leading edge is the rising edge 0→1, trailing edge is the falling edge 1→0

  - If CPOL=1
    - The clock will idle at 1
    - Leading edge is the falling edge 1→0, trailing edge is the rising edge 0→1

- Clock Phase CPHA
  - ➢ Edge of the clock signal upon which data is captured

	CPHA=0 IN captures data at leading edge OUT changes data at trailing edge	CPHA=1 IN captures data at trailing edge OUT changes data at leading edge
CPOL=0 Leading = rising Trailing = falling	IN (0→1) OUT (1→0)	IN (1→0) OUT (0→1)
CPOL=1 Leading = falling Trailing = rising	IN (1→0) OUT (0→1)	IN (0→1) OUT (1→0)

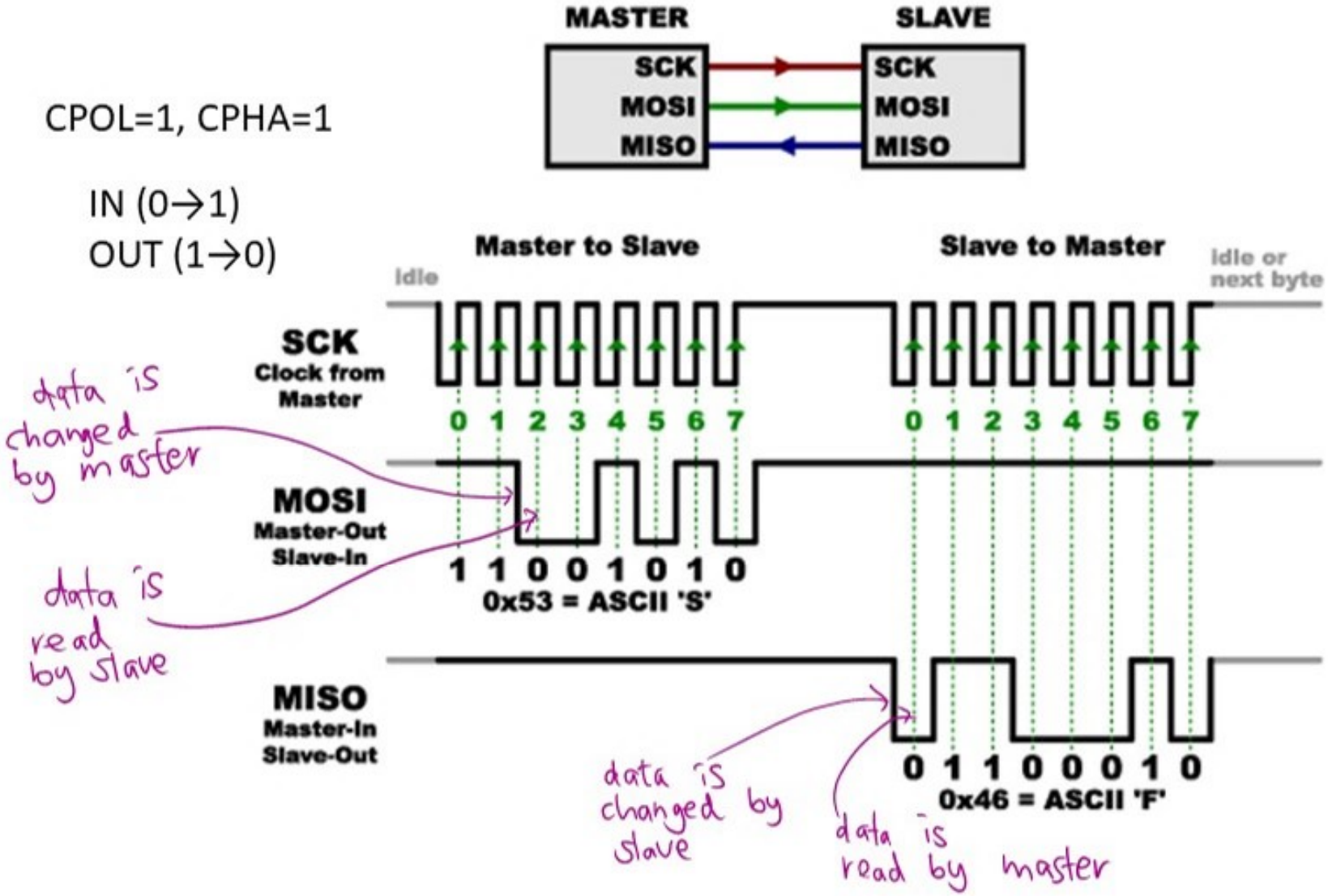
MOSI: Master OUT, Slave IN
MISO: Slave OUT, Master IN

SPI

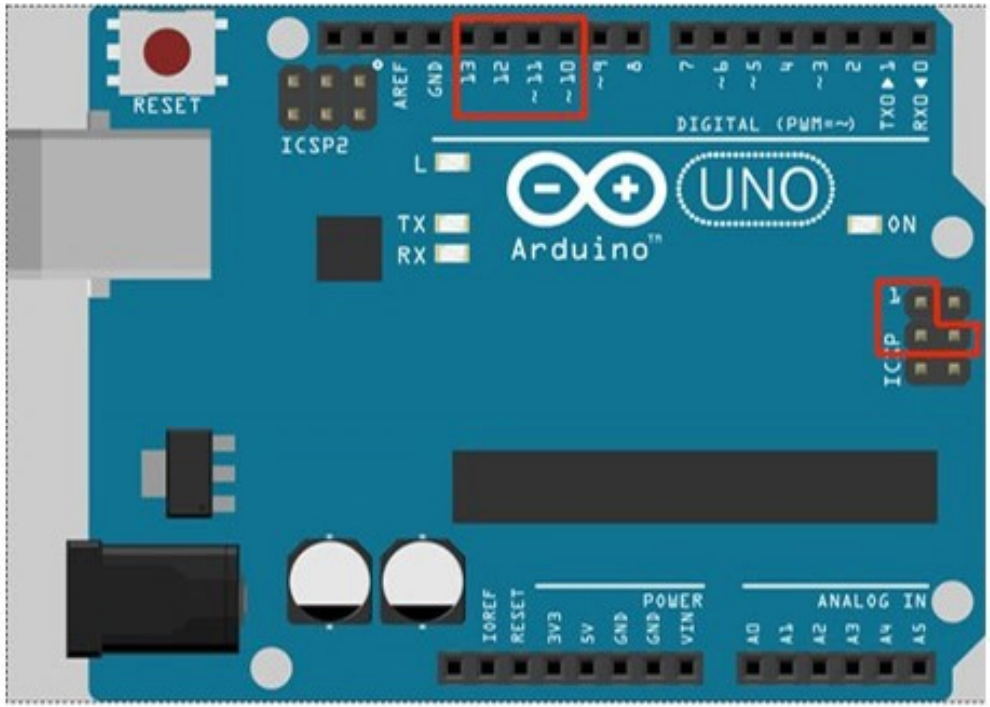
- Example

CPOL=1, CPHA=1

IN (0→1)
OUT (1→0)



- Arduino SPI Implementation
  - ➢ The SPI digital pin connections for SCK, MOSI, and MISO are predefined on Arduino boards.



- **SCK:** GPIO 13 or ICSP 3
- **MOSI:** GPIO 11 or ICSP 4
- **MISO:** GPIO 12 or ICSP 1
- **SS:** GPIO 10

Any digital pin can be also used for the SS pin. To select the device, this digital pin must be driven low.

Inter-Integrated Circuit (I2C)

- Drawbacks of UART and SPI

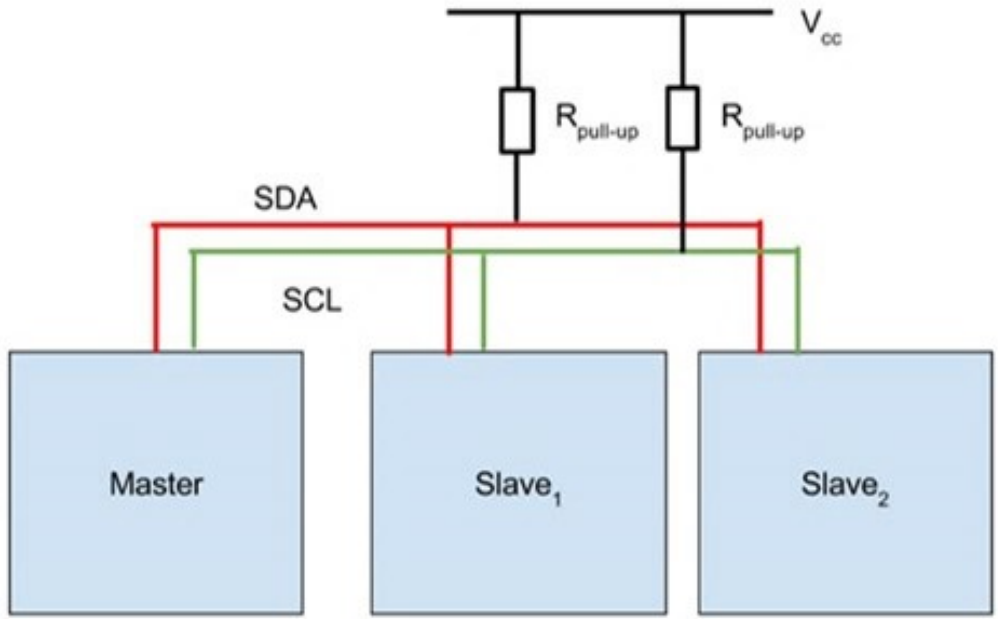
➢ UART
  - Asynchronous: 2 devices must have clocks that are close to the same rate
  - Communication between only 2 devices
  - Slow data rates

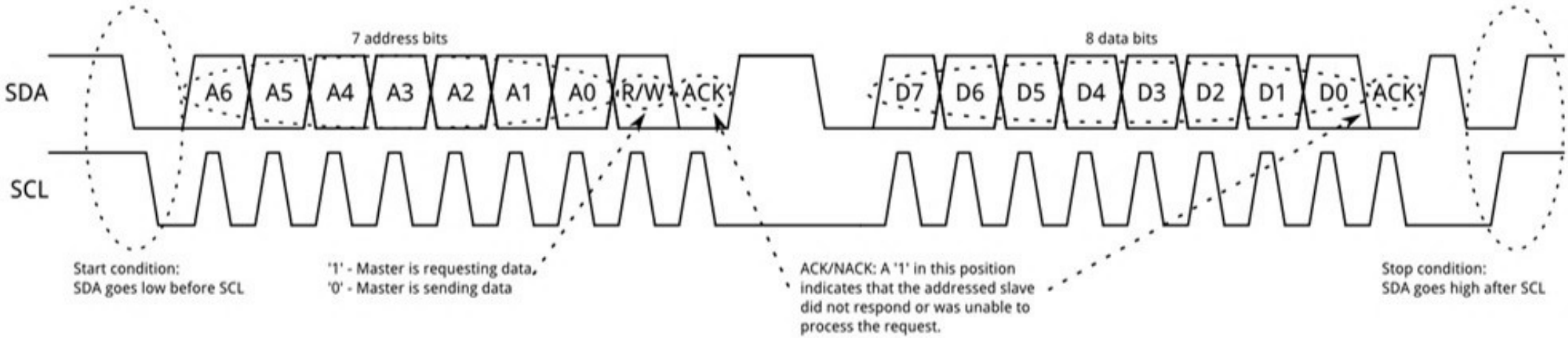
➢ SPI
  - Number of pins required grow with number of slaves
  - Only single master is allowed

I2C

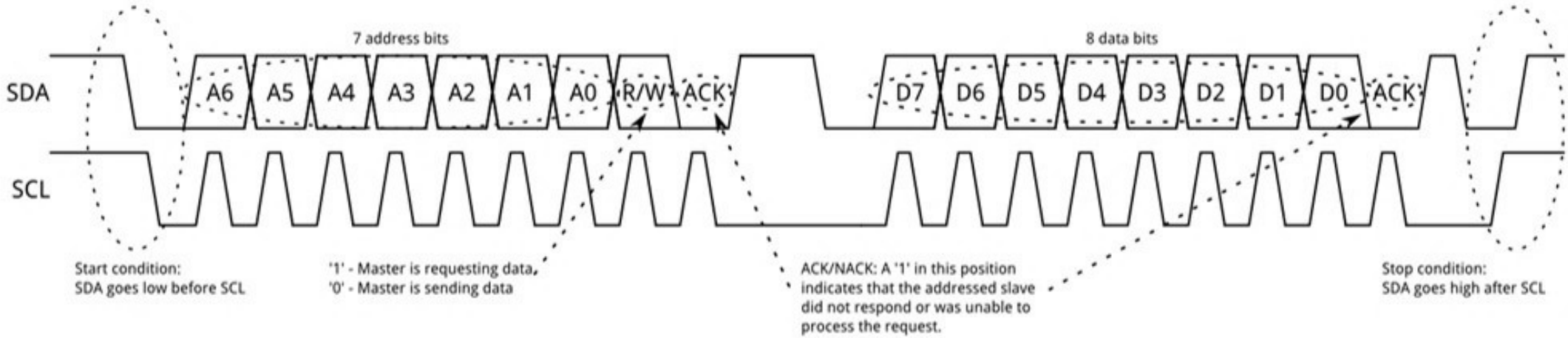
- Master-slave model
- Features
  - ➢ Ability to connect multiple masters to multiple slaves
  - ➢ High speed communication
  - ➢ Simple to implement: two wires and some resistors

- I2C / Two Wire Interface (TWI)
  - ➢ Data line: SDA, Clock line: SCL
  - ➢ SDA and SCL are pulled *high* in the idle state and the devices will make the lines *low* when data is transmitted

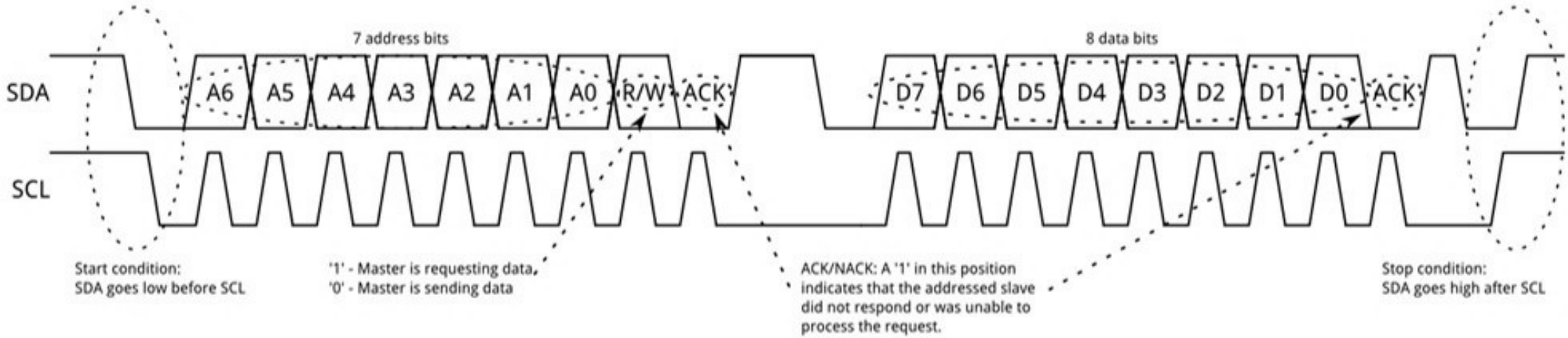




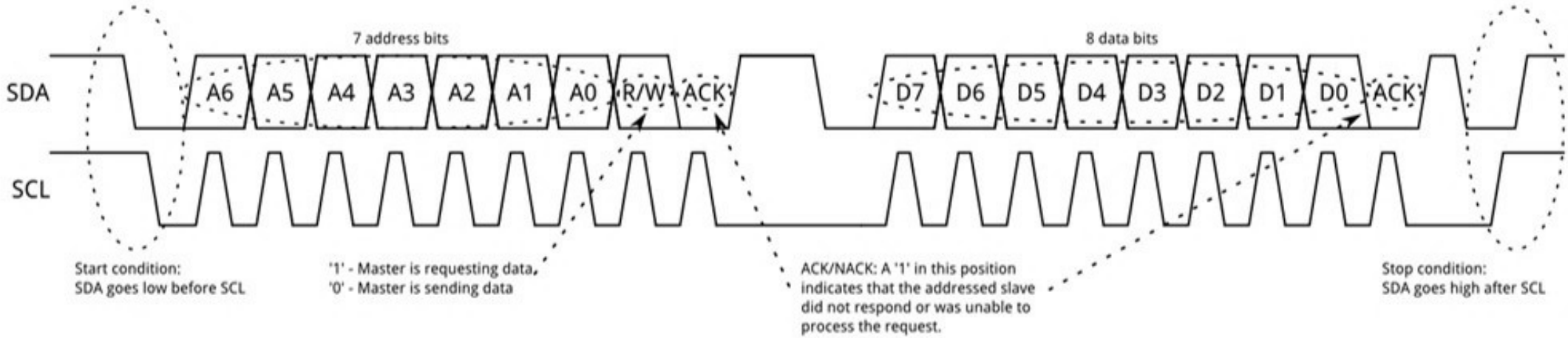
- Start condition
  - To initiate communication, the master device leaves SCL high and pulls SDA low
  - This puts all slave devices on notice that a transmission is about to start



- Address Frame
  - Any new communication starts with an address frame
  - For a 7-bit address, the 8th bit R/W is read (1) or write (0)
  - Once the first 8 bits of the frame are sent, the receiving device is given control over SDA and it should pull the SDA line low



- Data Frames
  - After the address frame, data frames can be transmitted either by the master or slave
  - Number of data frames is arbitrary



- Stop Condition
  - ➢ After all data is transmitted master will generate the stop condition, which is

0->1 (low to high) transition on SDA after a 0->1 transition on SCL, with SCL remaining high

Summary

Summary

- Different digital communication protocols were discussed
  - UART
  - SPI
  - I2C