CS 5422: Physical Computing

Common Digital I/O Protocols

Slides will be available through Blackboard. There is no textbook.

Digital I/O

- Voltages must be compatible
- Current levels must be compatible

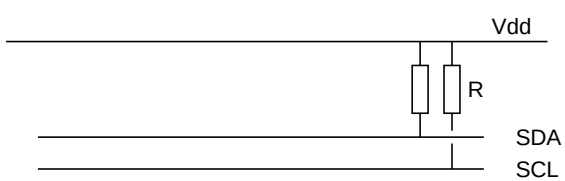
A number of different standards for communication between devices.

Common ones for microcontrollers and I/O devices:

- SPI: Serial peripheral interface
- I^2C: Inter-integrated circuit

Digital I/O

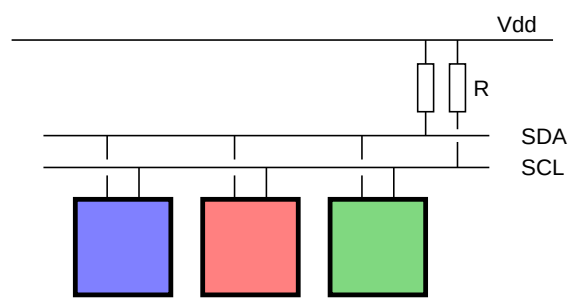
I^2C bus and protocol



- "Open drain" lines
- SDA: serial data
- SCL: serial clock

Digital I/O

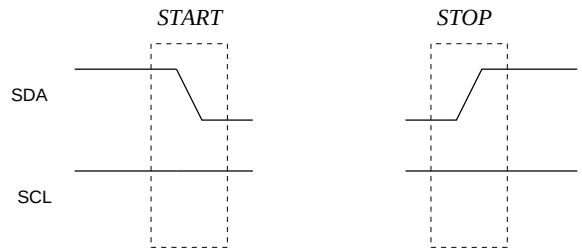
Multiple devices can be connected to the bus



- "Master": initiates data transfer
- "Slave": device addressed by the master

Bus supports multiple masters and multiple slaves

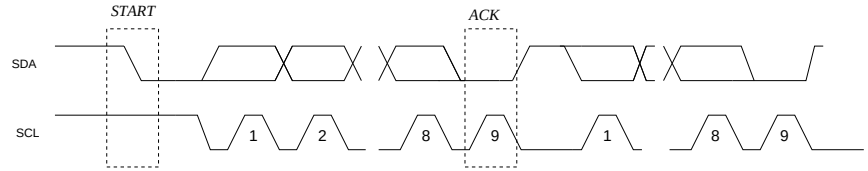
Digital I/O



(SCL is high)

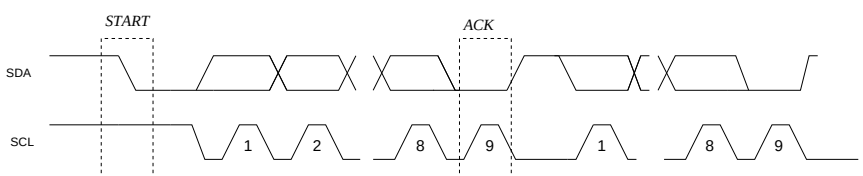
- Master creates the start condition by pulling SDA low
- After START, the bus is considered busy until a STOP

Digital I/O



- 8-bit transfers, with 9th acknowledge bit
- Data transferred MSB-first
- SDA must be stable when SCL is high
- Receiver generates the acknowledge
- If receiver cannot accept next byte
  - receiver holds SCL low
  - transfers can continue after the receiver releases SCL

Digital I/O



- If no acknowledge: master can generate a STOP and abort
- If the master is the receiver, to end the transfer it can use the acknowledge period to generate a STOP

Digital I/O

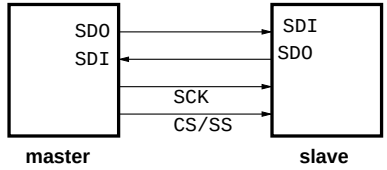
What about multiple devices?

- Standard format
  - First byte: command
    - First 7 bits: address
    - 8th bit: “read or write”
  - Next byte: data
  - ...

Often the device will contain information about exactly how many bytes to expect, etc.

Digital I/O

SPI protocol: 4-wire interface



- `SCK`: serial clock
- `SDI`: serial data in
- `SDO`: serial data out (shared `SDI`/`SDO` is 3-wire variant)
- `CS`: chip select (`SS`: slave select)
- Can specify clock polarity, edge for data sampling

Digital I/O

Word size can vary

- Typical is 8-bit
- 16-bit words, 12-bit words also possible
- Determined from data sheet of communicating devices

SPI with multiple slaves

- Multiple `SS` outputs from the master
  - Master selects one slave for communication
  - Input and output pins are wired together
  - Slaves must ignore inputs when not selected
  - Slave must not drive output when not selected