LECTURE VI

Communication Protocols II

A Note on Terminology

The words **master** and **slave** pervade modern software and electronics documentation. The terms are inappropriate given their connection to American chattel slavery. Therefore, we are transitioning to new terminology.

There will be a discrepancy between what is discussed in lecture/lab and what is written in our datasheets, but neither the lecture slides nor the OPS team will use the antiquated terminology.

A Note on Terminology (Cont'd)

Use this table for a conversion in terminology wherever you find it in the documentation for our Arduino board and other parts…

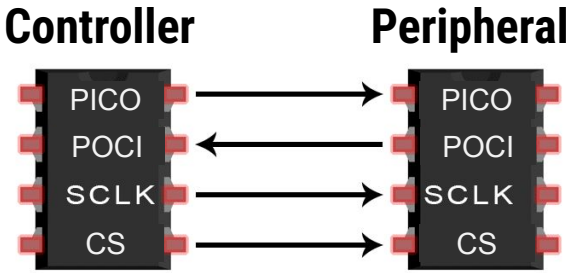
Old Term	New Term
Master	Controller
Slave	Peripheral
MISO	POCI
MOSI	PICO
SS	CS

SECTION I

SPI

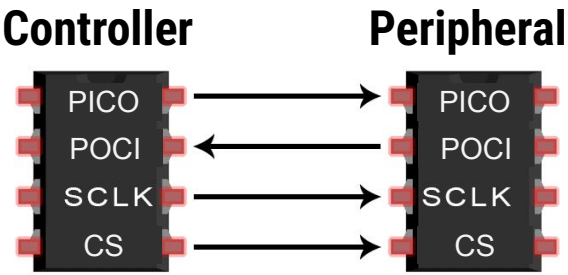
SPI Protocol

- **Serial Peripheral Interface** or **SPI** is a serial communication protocol which is **synchronized with a clock**
- Unlike UART, **data is transferred continuously** in SPI
    - There are no packets, start bits, stop bits, parity bits, or anything else
- One device acts as a **controller** to one or more **peripheral** devices
- Supports **full-duplex** communication

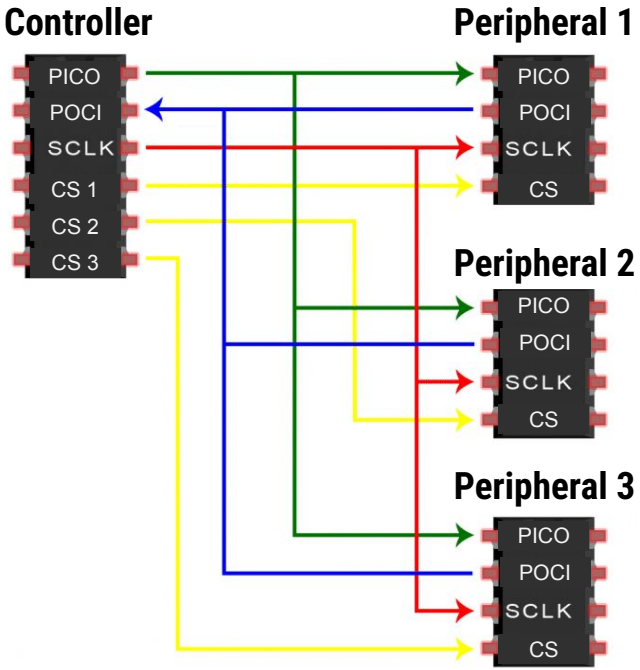


SPI Layout

- PICO (**Peripheral-In/Controller-Out**) - line for the controller to send data to the peripheral
- POCI (**Peripheral-Out/Controller-In**) - line for the peripheral to send data to the controller
- SCLK (**Clock**) - line for the clock signal
- CS (**Chip Select**) - line for the controller to select which peripheral to send data

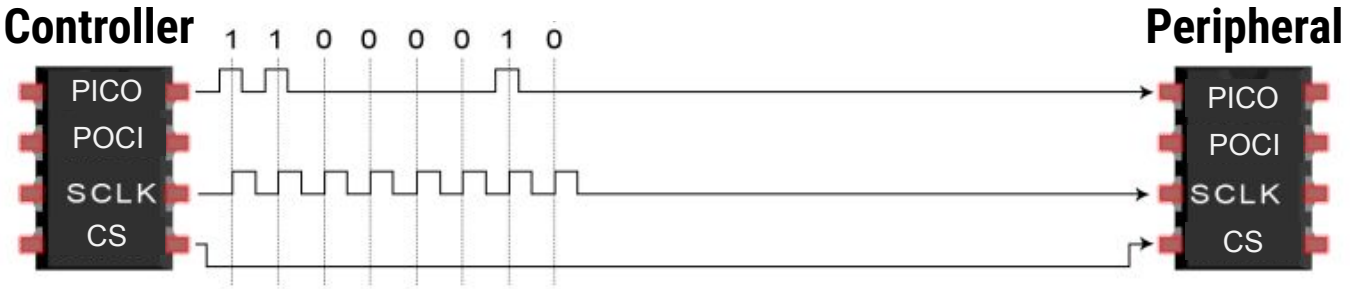


SPI Layout (Cont'd)



- There is **one controller**
  - Often a microcontroller (Arduino)
  - Generates the clock signal
- The controller controls **one or more peripherals**
  - i.e. Radio module, sensor, actuator
  - In the configuration to the left, there is an **additional CS line for each peripheral**

SPI Transmission



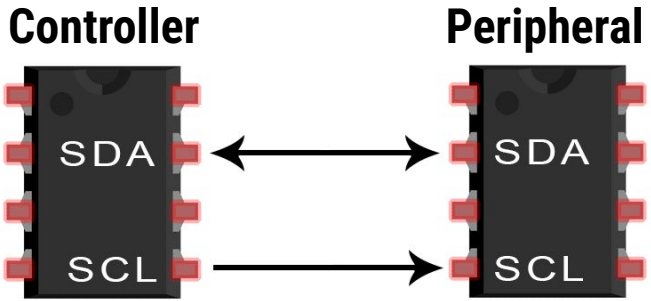
1. The controller first **selects a peripheral** by pulling a HIGH signal to LOW along the CS line
2. The controller **sends data** to the selected peripheral along the PICO line
    - Bits in the PICO signal are **synchronized with the rising edge** of the oscillating clock signal
    - Alternatively, the peripheral sends data along the POCI line

SECTION II

I^2C

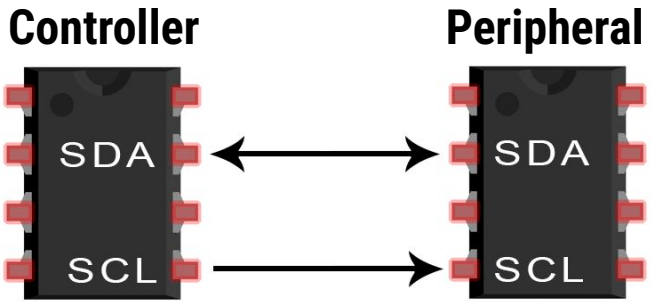
I²C Protocol

- Inter Integrated Circuit or **I²C** is another synchronized, serial protocol
- Unlike UART and SPI, I²C supports **multiple controllers**
- I²C features **acknowledgements** to confirm if messages are received and addressing for peripherals
- Supports **half-duplex** communication



I²C Layout

- **SDA** (**Serial Data**) - the line for the controller and peripheral to send and receive data
- **SCL** (**Serial Clock**) - the line that carries the clock signal
- All peripherals can share the same two SDA and SCL lines



I²C Frame Format

- Data is transferred as **messages**
- A **start condition** from the controller begins the message
- Each message contains an **address frame** with an address to identify the receiving peripheral

I²C Message

I²C Frame Format (Cont'd)

- The **read/write bit** indicates whether the controller is sending or requesting data
- Then, the peripheral with the matching address sends an **acknowledgement** (**ACK**) of the message as a single bit

I²C Message

I²C Frame Format (Cont'd)

- Depending on whether the message is read or write, either the controller or peripheral will send **data frames**
- The device receiving data will send an **ACK** before the transmission of the next data frame

I²C Message

I²C Frame Format (Cont'd)

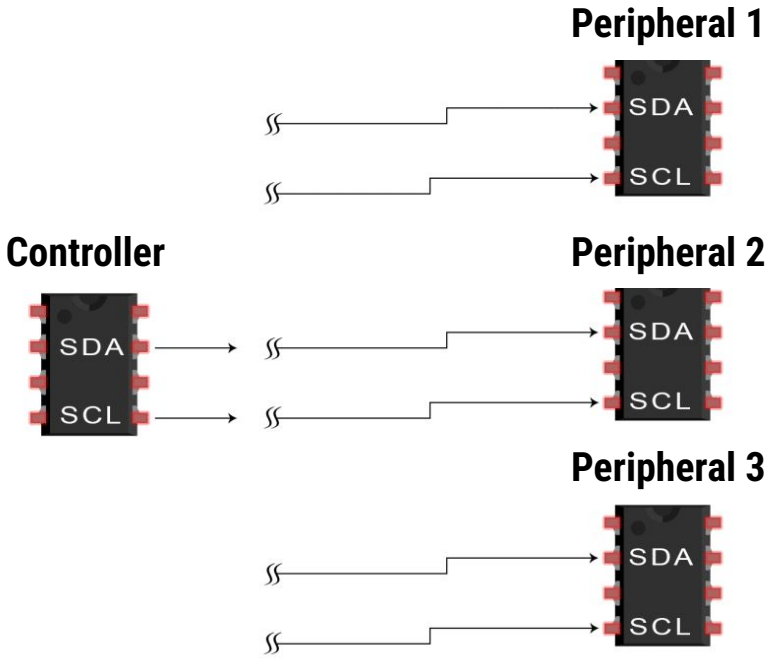
- The devices may transfer as many data frames in one message as desired
- If the controller is reading from the peripheral, it will send a **NACK** to end data transmission
- The controller terminates communication by signalling the **stop condition**
  - Another controller may send a message upon seeing the stop signal

I²C Message

I²C Transmission (Cont'd)

1. The controller outputs the **start condition** to all connected peripherals

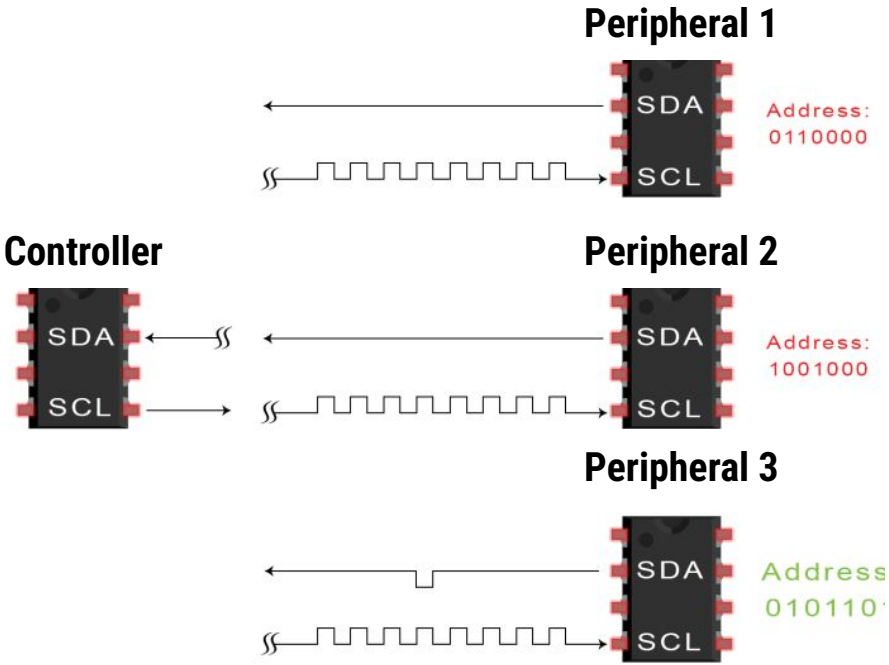
   Starting from a **HIGH** voltage in idle, the controller pulls to the voltage down **LOW** on the SDA line *then* the SCL line



I²C Transmission (Cont'd)

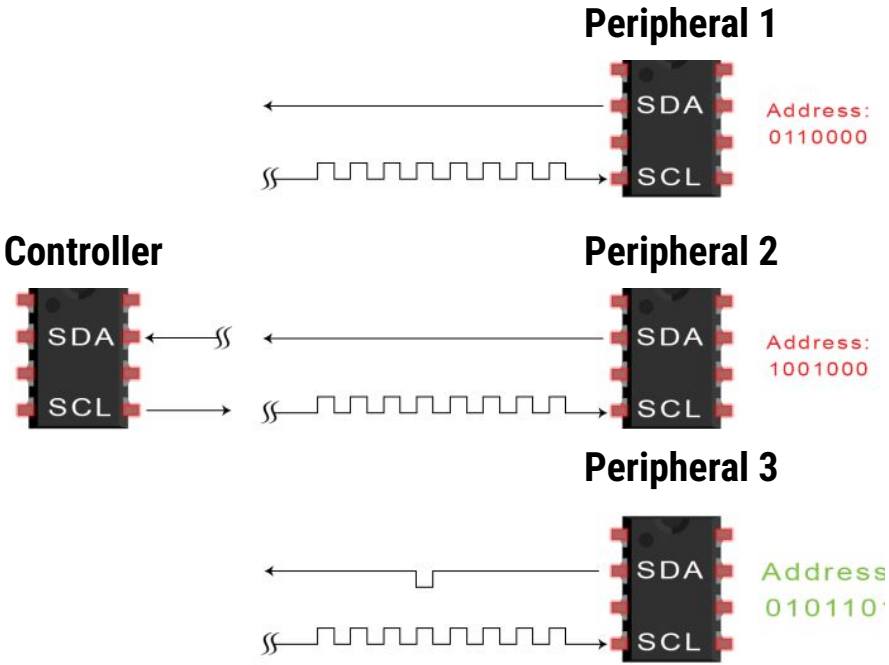
2. The controller then **sends the address** of the peripheral it would like to communicate with

   The **last bit of the address is the read/write bit** that indicates whether the controller is sending or receiving data



I²C Transmission (Cont'd)

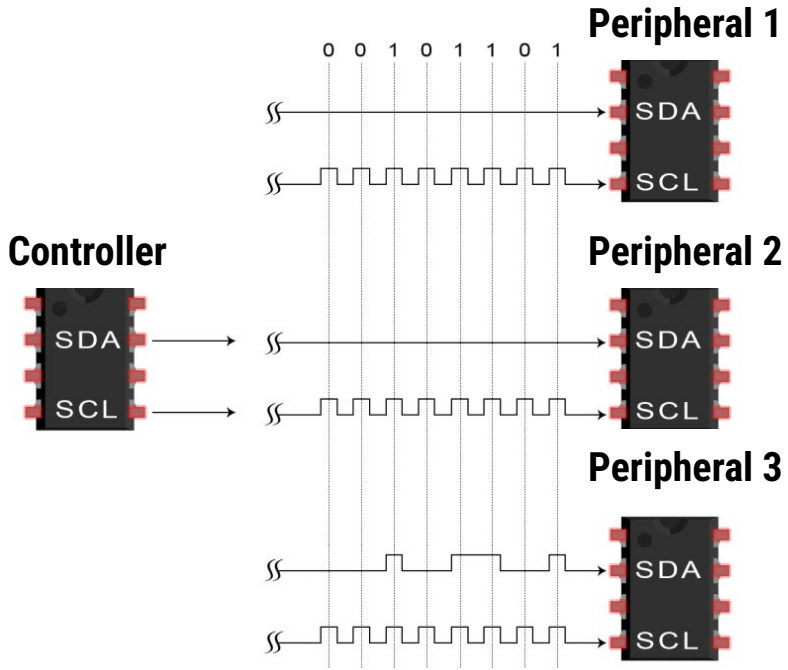
3. The peripheral with the matching address responds with an **ACK**



I²C Transmission (Cont'd)

4. The controller **sends or receives the data frame**, depending on the read/write bit

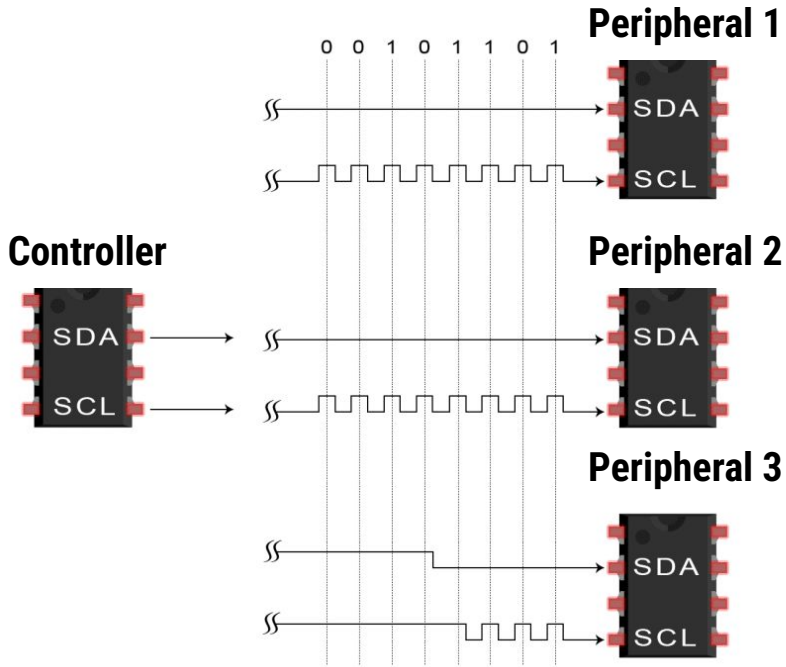
   After each transmission, the receiving device sends an ACK for a successful transmission



I²C Transmission (Cont'd)

4. Finally, the controller switches the voltage of the SCL line then the SDA line to **HIGH**, signalling the **stop condition**

   The message is complete



SECTION III

Comparing Protocols

UART vs SPI vs I^2C

Protocol	UART	SPI	I^2C
# of Lines	2	4+	2
# of Controllers	1	1	1+
# of Peripherals	1	1+	1+
Transmission Type	Full-Duplex	Full-Duplex	Half-Duplex
Error-Checking	✅	❌	✅
Speed	Slowest	Fastest	Slower

Comparing Protocols

I/A

Which one of the following is a full-duplex communication protocol?

A. SPI

B. I^2C

C. Both SPI and I^2C

D. Neither SPI or I^2C

Comparing Protocols (Cont'd)

I/A

Which of the following protocols doesn't require an additional line to support multiple devices on the same bus?

A. SPI

B. I^2C

C. Both SPI and I^2C

D. Neither SPI or I^2C

Comparing Protocols

I/A

Which of the following protocols supports multiple controllers and peripherals?

A. SPI
B. I^2C
C. UART
D. Both UART and I^2C

Comparing Protocols

I/A

Which of the following protocols doesn't have start and stop bits?

A. SPI
B. I^2C
C. UART
D. Both UART and I^2C

Comparing Protocols

I/A

The SCLK, PICO, POCI, CS are the four data lines in _____ protocol.

A. SPI
B. I^2C
C. UART
D. Both SPI and I^2C

Comparing Protocols

I/A

Which of the following communication protocols is a type of synchronous protocol?

A. SPI

B. I^2C

C. UART

D. Both SPI and I^2C

Comparing Protocols

I/A

Which one of the following protocols needs a clock?

A. SPI

B. I^2C

C. UART

D. Both SPI and I^2C

Comparing Protocols

I/A

The receiver and transmitter are the two data lines in _____ protocol.

A. SPI
B. I^2C
C. UART
D. Both SPI and I^2C

Comparing Protocols

I/A

Which of the following protocols needs a chip select line?

A. SPI
B. I^2C
C. UART
D. Both SPI and I^2C

Comparing Protocols

I/A

Which of the following protocols is a single controller, single peripheral communication protocol?

A. SPI

B. I^2C

C. UART

D. Both SPI and I^2C

Comparing Protocols

I/A

How many signal lines does I^2C Protocol require?

A. 1

B. 2

C. 4

Comparing Protocols

I/A

How many signal lines does UART Protocol require?

A. 1

B. 2

C. 4

Comparing Protocols

I/A

Which of the following protocols is best for multiple peripherals over the fewest lines?

A. SPI

B. I^2C

C. UART

Comparing Protocols

I/A

Which of the following protocols is best for multiple peripherals with the highest transmission rate?

A. SPI

B. I^2C

C. UART

Comparing Protocols

I/A

Which of the following protocols is best for full-duplex communication with multiple peripherals?

A. SPI

B. I^2C

C. UART

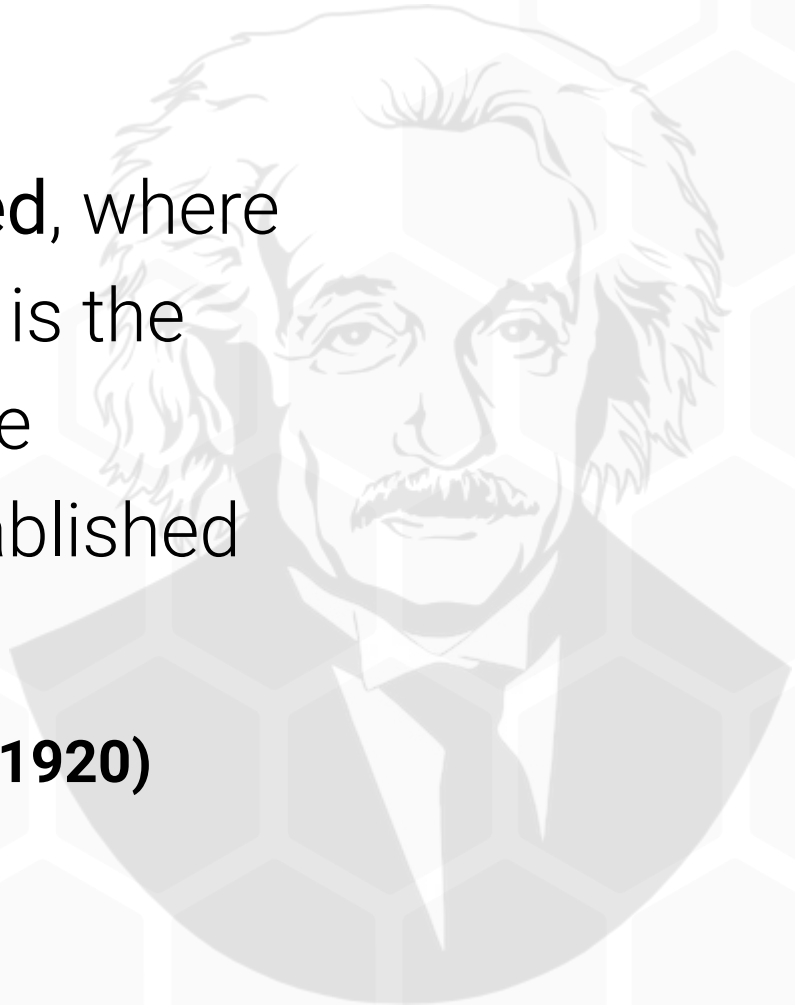
Comparing Protocols

I/A

Which of the following protocols consumes less power?

A. SPI

B. I^2C



“E equals MC serially squared, where M is the mighty data byte, C is the constant bit rate, and E is the enlightenment of a well-established communication protocol.”

Albert Einstein (circa 1920)

Famous Misquotes

FAIR USE DISCLAIMER

Copyright Disclaimer under section 107 of the Copyright Act 1976, allowance is made for “fair use” for purposes such as criticism, comment, news reporting, teaching, scholarship, education and research.

Fair use is a use permitted by copyright statute that might otherwise be infringing.

Non-profit, educational or personal use tips the balance in favor of fair use.