TO: Prof. Pierre-Emmanuel Gaillardon, Course Instructor
FROM: David Venegas
DATE: February 29th, 2024
SUBJECT: Pre-Lab 05 (I2C)

1. **Describe two differences between I2C master and slave devices?**

- Master devices initiate communication with slave devices.
- Each slave device has an unique hardware I2C address.
- A master selects a specific slave by sending its address on the bus.
- Slave devices can respond to a master device when requested, but can't start a new transaction on their own.

2. **What are the two connections in an I2C bus? Describe their purpose:**

- SDA (Serial Data): Depending on the direction of communication, both the master and slave produce data on the shared SDA line. When receiving, both slave and master devices acknowledge each communication frame to notify the other that the data was received.

- SCL (Serial Clock): When communicating, a master device produces clock transitions on the SCL line; the slave device uses this clock signal for both receiving and transmitting data. A slave can also hold the clock line low to pause the master if it needs more processing time, in a processes called clock-stretching.

3. **What is the difference between open-drain and push-pull outputs?**

- Open-drain: Open-Drain outputs have a single transistor and can only pull the output to a low state. Because of this, open-drain systems require an external connection—such as a pull-up resistor—to return the line to a high state when no device is pulling it low.

- Push-Pull: Push-Pull outputs have drive transistors that allow the device to push the output line "high" by connecting to the supply rail of the device, as well as pulling it "low" by connecting to ground. A push-pull output can source or sink current depending on the voltage of the external system.

4. **What is the purpose of the I2C restart condition?**

Because I2C is a half-duplex bus, master and slave devices cannot transmit simultaneously. If a master wishes to both write and read from a slave, it must begin a new transaction with the appropriate read/write bit set for each transaction. In the case of chained transactions, ending the current transaction with a stop condition would release the bus and allow other devices to steal control before the master begins again; to prevent this, devices may issue new start conditions

without properly ending the previous transaction with a stop. We refer to this event as a restart condition.

5. **What peripheral register would you use to set the read/write direction of the next I2C transaction?**

Control Register 2 (I2C_CR2) --- > RD_WRN: This bit sets the direction of data transfer for the next transaction; its state controls the read/write bit in the address frame.

6. **The 10-bit SADD bit-field holds the slave device address. Since standard I2C addresses only use 7 bits, to which bits in the bit-field would you write the shorter address?**

The default 7-bit addressing mode uses bits [7:1] within the center of the bit field.

7. **Name one thing you found confusing or unclear in the lab:**

Nothing looks confusing. But I will be interested in knowing the differences between UART and I2C serial communication protocols. And when one would be needed over the other.