

Goal for exercise – “Processor Communication”

The goal of this exercise was to become familiar with I2C and develop basic software that can carry out communication between the Pi, Arduino, and an LCD Display.

Are you aware of any kind of communication protocols? If so which ones and specify one instance of their applications.

I was already somewhat familiar with I2C as it was used in EENG 383. Beside I2C, another protocol of choice is SPI. Both protocols can be used in similar applications. I2C only requires two bus lines, while SPI needs 4. SPI offers the ability to transfer at higher speeds and two-way simultaneous communication. However, SPI does not have an official standard. I2C on the other hand is simpler/easier to implement and can have multiple master and slave device on the same line.

List all resources and what specifically you used or learnt from that resource to complete the challenge exercises.

<https://www.lifewire.com/selecting-between-i2c-and-spi-819003>

This source was helpful in pointing out the differences between the two protocols.

Compile a list of all documentation created. Provide file name and a short description of that file.

README.md contains documentation that describes how to run the program and answers the questions.

README.pdf is a pdf export for submission to canvas

Processor Communication RL.pdf is this word document which is the reflection log.

List any challenges you faced when trying to establish communication between the LCD and the Pi or 2-way communication between Pi and Arduino. Clearly explain how you solved the problems you faced and debugging methods you used to resolve the issue.

The main challenge was understanding how the SMBus library (python) was sending information depending on which function was used. For example, using `write_i2c_block_data` was different from `write_byte` in that the former could send multiple bytes.

Furthermore, using Serial print functions on the Arduino for debugging brought about issues that are not immediately apparent. This is because printing to console takes a lot of time and as a result, puts the I2C process on hold and disrupts communication.

On a scale of 1-5, how comfortable are you with the I2C communication protocol? (1 being least comfortable and 5 being most comfortable).

1

2

3

4

5

Name Pooya Aghanoury