ECE4871 – Culminating Design Project

Evaluation Form – Technical Background Review

Student Name: Dimitry Jean-Laurent

Project Advisor: TBD

Team Name: Blindle

Team Members: Emmy Perez, Annie Luo, Jordan Altaffer, Cameron Davis

_______ / 30

Technical Content

- Current state-of-the-art and commercial products
- Underlying technology
- Implementation of the technology
- Overall quality of the technical summary

_______ / 30

Use of Technical Reference Sources

- Appropriate number of sources (at least six)
- Sufficient number of source types (at least four)
- Quality of the sources
- Appropriate citations in body of text
- Reference list in proper format

_______ / 40

Effectiveness of Writing, Organization, and Development of Content •

Introductory paragraph

- Clear flow of information
- Organization
- Grammar, spelling, punctuation
- Style, readability, audience appropriateness, conformance to standards

_______ / 100 Total - Technical Review Paper

Dimitry Jean-Laurent
TBD
Blindle
Microcontroller Communication Protocols

Introduction

As technology becomes more complex, how separate devices communicate with each other become ever more important. Microcontrollers and device modules utilize communication protocols to communicate with each other. The devices a controller can communicate with range from sensors to Bluetooth modules. This technical review will examine three common microcontroller communication protocols: SPI (Serial Peripheral Interface), I2C (Integrated-Integrated Circuit), UART (Universal Asynchronous Receiver Transmitter). More specifically how they are used commercially, how they function, how they are implemented, and important tradeoffs to consider.

Commercial Applications

Due to its simple implementation I^2C is found in several embedded devices. For example, the Analog Devices AD5243 potentiometer ($1.03-$1.14) is a variable resistor with four end-to-end resistance values (2.5 kΩ, 10 kΩ, 50 kΩ, and 100 kΩ). It also has wiper settings which are controllable via the I^2C as well as extra package address decode pins which allow for multiple parts to share the same I^2C wire bus on a PCB [1]. The potentiometer can be useful for tasks such as dimming lights or volume control.

UART is another popular communication protocol used in many devices. The Raytac's MDBT40-P256V3 ($7.80) is a BT 4.2 stack module which utilizes UART to communicate serially with a microcontroller [2]. This module gives the microcontroller the ability to communicate with other Bluetooth devices wirelessly. Examples of these devices include sensors, wireless peripherals, and medical devices.

The last popular protocol is SPI. It is found in the *Maxim* MAX31855K. This is a device which performs cold-junction compensation and digitizes the signal from a thermocouple to read temperatures. Data is sent out in a 14-bit signal which is compatible with SPI. It is capable of reading temperatures ranging from -270 C to 1800 C. This device is meant to communicate with an external microcontroller to perform thermostatic, process-controls, or monitoring applications [3][4].

Technology of Microcontroller Communication Protocol

In I^2C the message is broken into two frames: address and data. The address frame contains the address of the slave the master intends to communicate with, and the data frame contains the data to be sent. To signal a start to communication, the master sends out an address frame to the slaves which compare it with their own address. The slave which detects a match sends the master an acknowledgement signal. Following this, either the master or the slave sends data on the SDA line. After data is transmitted, the receiving device sends an ACK bit on the SDA line to acknowledge successful delivery of the data. To halt data transmission, the master sends a stop condition by switching SCL high before switching the SDA high [5].

In the UART protocol, two UART modules, are required to communicate with each other. First, the sender UART receives data from the data bus in parallel form. Once the data is loaded in, the sender UART appends a start bit to the beginning, and parity and stop bits to the end of the signal. The new data is then output serially from the transmitter pin the sender UART module and sent to the receiver pin of the receiving UART module. The receiving UART module then removes the extra bits (start, parity, stop), converts the data to parallel form, and pushes it onto the data bus [6].

In the SPI protocol one device (master) controls one or more other devices (slaves). The master initiates communication by activating the SS (Slave Select) signal to select a slave to communicate with. The master then drives data onto the MOSI (Master Out Slave In) line. Depending on the command, slave responds by driving data to the MISO (Master in Slave Out) lines [7].

Implementation

The following section briefly reviews how I^2C, UART, and SPI are implemented. The I^2C implementation allows for multiple masters and multiple slaves. It requires only 2 wires: an SDA (serial data) line and an SCL (Serial Clock) line. The software stack is also required to control the protocol. UART is implemented with two UART modules which are responsible for converting parallel data to serial and vice versa. This is also a two-wire system which requires a connection from the Tx (Transmitter) pin of each UART module to the Rx (Receiver) pin of the other. In SPI there is one master which can communicate with multiple slaves. For one slave this implementation requires only four wires: SCLK (Serial Clock), MISO (Master In Slave Out), MOSI (Master Out Slave In), and SS (Slave Select). For each additional slave, another SS line needs to be added.

Tradeoffs

I^2C requires only 2 wires and allows for multiple master and multiple slave implementations. Furthermore, simple addressing means that additional wires are not required for additional slaves. Even so, multiple masters and slaves can result in complex systems and the interface only supports half duplex data transmission. UART is a more popular protocol as there is UART support in almost all devices with 9 pin connectors. However, it can only support communication between two devices and must have a fixed data rate prior to data transmission. It is also limited to a range of 50 ft. SPI boasts the highest range and data transmission speeds of the three and supports full duplex communication. However, the hardware complexity increases heavily as more slaves are added and only single master implementations are supported [8].

References

[1] Analog Devices, “Dual 256-Position, I2C- Compatible Digital Potentiometers”, AD5243/AD5248, April. 2016

[2] Raytac Corporation, “Bluetooth Low energy Module”, MDBT40-P Series, Jun. 2020

[3] Maxim, “Cold-Junction Compensated Thermocouple-to-Digital Converter”, Feb. 2012

[4] S. Arora. ECE 4321. Class Lecture, Topic:” Communication using MCU” School of Mechanical Engineering, , Indian Institute of Technology, Kanpur.

[5] Jonathan Valdez and Jared Becker, “Understanding the I^2C Bus”, Texas Instruments, Texas, United States, 2015

[6] Kumar, “UART Communication Protocol – How it works?” codrey.com, para. 2, Jun. 16, 2020 [Online]. Available: https://www.codrey.com/embedded-systems/uart-serial-communication-rs232 [Accessed Oct. 5, 2020]

[7] C. Vonk, “Math Talk” October 2015. [Online]. Available https://coertvonk.com/hw/logic/connecting-fpga-and-arduino-using-spi-13067. [Accessed Oct. 4 2020].

[8] R. Mitchell, “Common Communication Peripherals on the Arduino: UART, I2C, and SPI” maker.pro August 30, 2018. [Online]. Available: https://maker.pro/arduino/tutorial/common communication-peripherals-on-the-arduino-uart-i2c-and-spi. [Accessed Oct. 6, 2020].