Laboratory 1 Serial Device Driver and Protocol Communication

Daniel K. Krygsman

University of California, Santa Cruz

Baskin School of Engineering

ECE 121L: Lab Number 45444: Section 01D: Microcontroller System Design Lab

Professor Stephen Petersen

March 16, 2023

Author Note

First paragraph: Abstract

Second paragraph: Introduction

Third paragraph: Uart

Fourth paragraph: Circular Buffers

Fifth paragraph: Uart Functions

Sixth paragraph: Packet State Machine

Seventh paragraph: Protocol Functions

Eighth paragraph: Conclusion

Abstract

The purpose of this lab is to make and understand serial communication between the pic32 development board and another microprocessor, in this case our computer. This is done through a uart chip and a needed uart device driver. To understand the process, the lab consists of three parts, 1) initializing and using the serial port, 2) creating a full duplex interrupt driven rx/tx serial device driver, and 3) utilizing the device driver to create a packet application. The progression of this lab allows for a clear understanding of microcontroller serial communication, including the use of multiple circular buffers for short term storage of information and the use of interrupts when dealing with interrupting the normal running program.

Introduction

Coming into this class with no experience of embedded systems proved to be challenging as interfacing software with the microcontrollers hardware varies with each microcontroller family. In this case an understanding of interfacing the pic32 comes solely from the family reference manual, and from a thousand page document understanding how to find the given information is key. With this understanding we start with the uart registers and simply connecting the transmit buffer register to the receive buffer register while there is data to send. This combined with the proper uart initialization allows us to accurately receive and retransmit single byte characters. With this we move to the send part of the lab and create circular buffer functions to allow short term storage for characters while interrupting code is transmitting. This allows us to send entire strings of characters without losing data. This is the main part of the lab as correct interrupt code configuration is the heart of all serial communication. The last step utilizes this device driver and incorporates it with the packet state machine which verifies if incoming data correctly makes up a packet and if so creates it. These data packets depending on their payload ID perform various tasks such as led control, reading, or endianness conversion. With all parts working properly the microcontroller can predictably receive and interpret data.