ECE 414 – Embedded Systems
Implementation Plan for Lab 04 – UART

GuoYuan Li (Scribe)
Harry Zhu
Sep 15, 2020

1. Introduction

The objective of this lab is to learn the basics of UART by programming the UART in PIC32 microcontroller with blocking I/O. In addition, functions for configuring data transmission and receiving are also essential in this lab. Furthermore, learning to use the interrupt-driven “ztimer” module discussed in class to work with longer timing intervals than allowed by the “timer1” module. Finally, it is equally important to learn how design choices impact code performance.

2. Requirements

For this task you will create a module named uart1 that provides a clean interface for writing and reading characters to/from UART1.
1. The UART driver shall be named “uart1” and include a header file “uart1.h” and code file “uart1.c”.
2. The module shall include a function to initialize the UART to communicate using eight data bits and no parity bits at a specific baud rate: `void uart1_init(uint32_t baudrate);`
3. The module shall include a function that returns a true value when the UART transmitter is ready to accept a character for transmission: `uint8_t uart1_txrdy();`
4. The module shall include a function to write a character to the UART: void `uart1_txwrite(uint8_t c);`
5. The module shall include a function to write a null-terminated string to the UART: `void uart1_txwrite_str(char *c);` This function must not overflow the transmitter – if the transmitter is not ready, it should suspend execution and wait until ready before sending more characters (this is known as blocking I/O).
6. The module shall include a function that returns true when the UART receiver has a character ready for `reading: uint8_t uart1_rxrdy();`
7. The module shall include a function that reads a received character from the UART. `uint8_t uart1_rxread();`
8. The module shall be tested by a unit test that performs the following tasks:
    a. Print out a greeting message including the names of the lab partners.
    b. Enter an infinite loop where it waits for a character to be received; reads the character, inverts the case of alphabetic characters (i.e. ‘A’->’a’ and viceversa) waits for the transmitter to be ready, and writes the character back to the transmitter.
    c. Unit test shall be performed using input strings ranging in length from a single character up to 100 characters.
    d. Unit test shall be performed using baud rates ranging from 1200-19200 baud.

3. Architectural Description

This section describes the planned architecture for a design, generally partitioned into a hardware architecture and a software architecture.

3.1 Hardware Implementation

No hardware implementation for this lab because it is for testing the UART on PIC 32 and only 3 pins for transmission(U1TX), receive(U1RX), and ground(GND) should be connected properly.

3.2 Software Implementation

The main software block consists of two modules: the UART1 module and a Test module. Test module sends an input array of characters(String) for testing to UART1 and UART1 receives character by character serially. Once a full character is received, required operation is performed on the character, in this case inverting the case of the alphabetic character, and it is transmitted to the Test module serially, or bit-by-bit. Meanwhile, UART1 is receiving another character from the String.

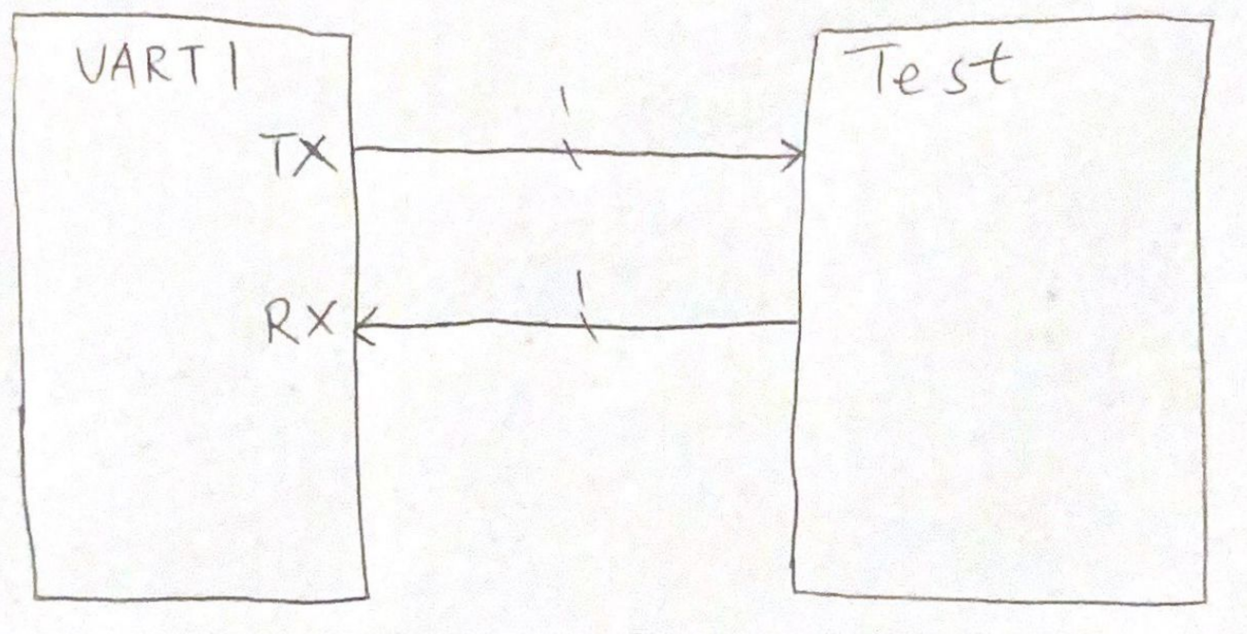

Figure 1. Software High Level Block Diagram

4. Test Plan

4.1 Unit Test of UART1

Unit test is performed by the test module to see if UART1 module behaves correctly

- Print out a greeting message including the names of the lab partners.
    - **Pass**: a string of characters of names is transmitted successfully and appear on the test module

    - **Fail**: transmission fails, nothing shows up on the test module
- Enter an infinite loop where it waits for a character to be received; reads the character, inverts the case of alphabetic characters, waits for the transmitter to be ready, and writes the character back to the transmitter.
    - **Pass**: Test module shows every of alphabetic character sent has its case inverted
    - **Fail**: Case is not inverted or transmission/reception fails
- input strings ranging in length from a single character up to 100 characters.
    - **Pass**: Test module shows every of alphabetic character sent has its case inverted
    - **Fail**: Case is not inverted or transmission/reception fails
- baud rates ranging from 1200-19,200 baud.
    - **Pass**: higher baud rate has faster transmission and reception period, lower baud rate has lower transmission and reception period
    - **Fail**: Error or no change in period

4.2 Acceptance Test

- **T1: set the baud rate to 1200 baud**
    - **Pass**: baud rate is 1200 with transmission period of 833us
    - **Fail**: Baud rate is not set or to the incorrect baud
- **T2: Type ‘B’ on Test module for UART1 reception**
    - **Pass**: ‘B’ is transmitted to UART1 bit-by-bit serially and ‘b’ is received by Test module
    - **Fail**: character is not received or nothing/wrong character is returned
- **T3: print the lab partners’ names**
    - **Pass**: a string is transmitted and received by the test module
    - **Fail**: wrong or no string is received
- **T4: type “HelloWorld” on the test module**
    - **Pass**: string is received by UART1 and cases are inverted and then transmitted by to test module; final string is “hELLOwORLD”
    - **Fail**: string is incorrect or transmission/reception error

	Requirements							
Tests	**2.1**	**2.2**	**2.3**	**2.4**	**2.5**	**2.6**	**2.7**	**2.8**
T1.	**x**	**x**						
T2.	**x**	**x**	**x**	**x**		**x**	**x**	
T3.	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**
T4.	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**

Figure 2. Traceability Matrix

5. Schedule.

Preliminary design is finished and corrected in actual implementation in the first weekend. Guoyuan will be writing the software code for the UART1 module Harry will check the correctness and efficiency of this module and checking the program syntax and help to debug potential issues. Both of us will perform the unit test and acceptance accordingly on the weekend and the beginning of the next week.