Eugene Ngo

11/17/2022

EE 474

Lab 3

Procedure

Task 1a

This lab was focused on Analog to Digital Convertes or ADC for short. I first read through the lab specification to get a better understanding of the ADC module. Then I used the datasheets to configure the GPIO, the timer, and the ADC module. For the ADC module, the phase-locked loop should be enabled in order to clock the ADC. The ADC is then configured using a timer which also has to be configured to trigger at a rate of 1 Hz. Instead of using timer interrupts in this lab, we used the interrupts corresponding to the ADC. The onboard LEDs are enabled and configured using the same pins as in past labs. To control the LEDs with a potentiometer, I hooked one end to the 3.3V and the other end of the potentiometer to the ground on the board and the middle connects to the ADC. Then using the given resistance equation and the voltage, the resistance of the potentiometer can be calculated and constant thresholds can be set.

Task 1b

This task was pretty straight forward from part A. The board has a built-in internal temperature sensor which converts the temperature into a voltage. We configure the ADC module to read this voltage and convert it into a temperature value. Then prints the temperature to the terminal. The on board switches are also coded to change the clock frequency to 12 MHz or 120 MHz and based on the clock frequency the temperature readings will either be hotter if it’s a faster frequency which heats up the system or colder if it’s the slower frequency which cools the system. The change is reflected in the temperature reading outputs to terminal and is shown in the demonstration video.

Task 2a

Task 2 of this lab used the Universal Asynchronous Receiver and Transmitter (UART). Using the user manual, the UART module is configured to communicate via the virtual port. To configure the UART we look at the datasheet and find the correct GPIO pins to enable the UART module. Then using the provided PuTTY for UART communication document, we set up PuTTY with all of the specified settings. Then using the same pin setups from Task 1b we had the onboard temperature sensor read the temperature and instead of printing the temperature to the terminal, it instead is read into PuTTY.

Task 2b

For task 2b, I used the Bluetooth for UART Communication document and the datasheets to set up the Bluetooth module. Looking at the pictures, I hooked the Bluetooth module up to the microcontroller. Then configured PuTTY according to the previous document to take in single character inputs. The result is the PuTTY terminal displaying the user input into the PuTTY terminal.

Results

The overall results were as specified in the spec and each task worked as intended a shown by the submitted demonstration videos. Task 1a turned on the LEDs sequentially being controlled by a potentiometer. Task 1b used the on-board temperature sensor to read and print the temperature to the terminal. Then Task 2a built upon task 1b and printed the temperature out to PuTTY. The last task, task 2b used a Bluetooth module to take in user input and display it on the PuTTY terminal.