

Lab 3 – Serial and Timers: Datalogging

Overview

This 2 week lab will have you creating a program that can take acceleration measurements and transmit them to a PC using serial.

Prelab

1. What do the following register and bit names stand for in reference to the AT90USB TIMERO (pages 100 to 116 of the datasheet found on the lab webpage)? Also for each write one sentence, in your own words, what each does.
 - a. TCCR0B
 - b. OCR0A
 - c. TIFR0
 - d. OCF0A
 - e. CS02:0
2. Explain in words what the bit OCF0A in TIFR0 does. How could this be used to perform some operation at a specific time (frequency)?
3. Give the proper C syntax for an enum with the options of: idle, running, walking
4. Using internet research, understand what a state machine is and draw a 5 state, state diagram for going out on a date. You can be funny, but keep it clean. Be sure to label what is needed to move from one state to another. Be careful, this is not a flow chart, it is a state diagram.
5. Research the function itoa(). What does it do?

Procedure

1. Download the skeleton code for this lab from the lab webpage. In the main.c file you will find 3 function stubs that need to be filled out to allow for your program to operate based on the internal timer 0. Read below to understand what each function does and to get some hints for each function:
 - unsigned char InitializeTIMERO()
This function should setup the TIMERO on the Wunderboard to be in a Clear Timer on Compare match (CTC) mode with a base clock frequency of $\text{clk}_{\text{IO}}/1024$. This function should return a 1 if it fails and a 0 if it does not.

HINT: The variables you NEED to manipulate are already listed in the skeleton code. Look each of them up in turn to understand them.
 - unsigned char checkTIMERO()
This function should immediately clear the OCF0A flag if the timer has elapsed and return a 1 so that it is ready for

the timer to elapse again. If the timer has not elapsed, it should immediately return a 0.

HINT: Look at the OCF0A bit in the TIFR0 register to see if you can write a new value. Don't forget you have to manually reset this flag every time the timer elapses.

- unsigned char setTIMER0(unsigned char clock, unsigned char count)
This function takes two values, clock and count. The value of count should be copied into OCR0A and the value of clock should be used to set CS02:0. The TCNT0 variable should also be reset to 0 so that the new timer rate starts from 0. This function should return a 1 if it fails and a 0 if it does not.

2. Once you have written each of the functions, be sure to test them and show that you can blink LEDs at a certain speed. You will need to show the TA that you can change the speed of blinking based on the input values of the setTIMER0() function.
3. In a previous lab you implemented serial transmission. To start the data logging you need to be able to receive a byte from the PC and receive a different byte to stop. You must create a state machine in your program using an enum. Figure 1 show the state machine you need to implement. To start sampling, use the character 's' to stop sampling you must also use 's'.

When you create the serial receive function, be careful to not make it 'blocking' since you want to be taking measurements at certain times even if the user has not sent a new character.

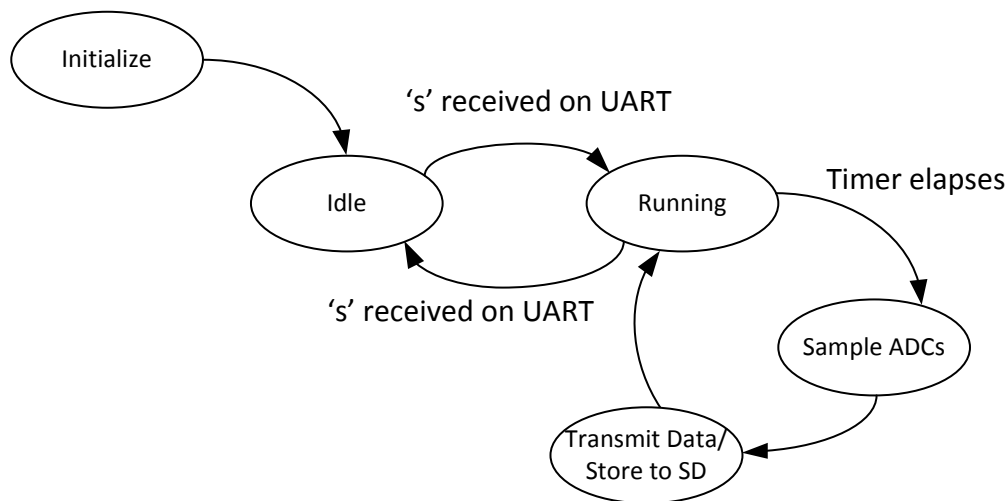


Figure 1: Lab 3 State Machine

4. Next combine all of the parts together to make a program that reads from the accelerometer and transmits the values to the PC. These values should be transmitted exactly 2 times a second. Once the values have been stored for about a minute, you should cut and paste the data in your terminal program into a file and graph it using excel or similar.

To include the functions for reading from the ADC, copy the function `read_adc()` from `adc.c` found on the course webpage.

Demonstrate Code

Show your code and program to your TA. They will watch your system run for 30 seconds and count the number of samples in 30 seconds (it should be 60 ± 1). They will come back and look at your graphed output of the samples you took in their presence. Be sure you bring a printed copy of your code to give to your TA.

Study Questions

1. Develop and draw a possible state diagram for a program that can send and receive start and stop commands over the serial port and display animations on the LED array loaded from the SD card. It should be very obvious from your diagram that program would be able to update the display quickly enough.
2. For your diagram above, write pseudo code version for the main function. Don't worry about what each function would do, just use descriptive names in the pseudo code.

Lab Summary

Task	Completed?
Prelab	2pts.
Timers Work	3pts.
Final Code uses an ENUM type and is a state machine	3pts.
Final code Works	10pts.
Study Questions	2pts.