

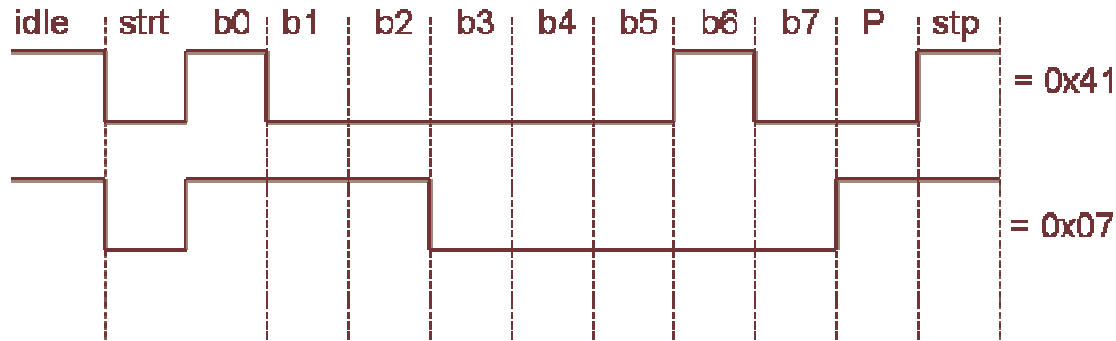
**HOMEWORK SOLUTION #7**

Due Friday, May 8th, 2009

**1. (10 points) Serial Data Frames**

Draw the 11-bit (8 data, even parity, 1 stop) frames for the asynchronous transmission of the ASCII character 'A' and the ASCII bell character (BEL).

**Note: Clearly indicate bit intervals in the waveform.**



'A' = 0x41 (RS232 goes LSB to MSB)

BEL = 0x07

**2. (20 points) Dumb Terminal using the UART**

Write a program that will configure the ADuC7026 UART for operation at 38,400 baud, 8 data bits, no parity, 2 stop bits, using pins P1.0 and P1.1.

Configure so that UART reception will cause an IRQ. All other UART interrupts will be turned off (For this problem we will ignore error conditions, and handle transmit complete through polling).

In this problem we will implement a dumb terminal. In Keil  $\mu$ Vision under: **View**→**Serial Window**→**UART #1** you can get what is like a **very** dumb terminal. Everything you type in this window is sent to the UART at the configured baud rate. Everything you send from the UART will show up in this terminal.

However, what you type, does not automatically show in the terminal. So one thing we need to do is every time we receive a character we need to echo it back. This will be done in the IRQ ISR. Read the received character and transmit it out (write to COMTX).

However, we want to make a prompt to. You should define a prompt string using DCB in the flash AREA...

DCD "\nDumb Terminal>\0"

Note the newline character at the beginning, and the fact that it is a null terminated ASCII string.

You will need to write a routine (*displayPrompt*) in your main source file to display this prompt. It should read the characters one by one and send them over the UART. After each character is written to COMTX you have to wait for it to actually get transmitted. You can do this by monitoring the TEMT bit of the COMSTA0 register. Do not send the null termination character `\0`.

Your main routine should initialize the UART as mentioned above. Then it should call the *displayPrompt* routine once. Then it should enter an infinite loop doing nothing, because everything else will be handled by the UART receive interrupt.

The UART receive interrupt will grab the received character and compare it to a carriage return. If it is a carriage return then it should call *displayPrompt*. If it is not a carriage return it should simply echo it to terminal (write it to COMTX).

**Important:** Submit your program source code files *teamX\_prob2\_main.s* and *teamX\_prob2\_exceptions.s* using the homework #7 dropbox in Learn@UW. Also, submit a **paper copy** of *teamX\_prob2\_main.s* and *teamX\_prob2\_exceptions.s* with the rest of the assignment.

See solution posted on Web.

### 3. (10 points) INL of a A2D

On the website there is a Excel file called A2D\_INL.xls. It contains 17 measurements of a 14-bit A2D converter. Using these measurements find the “best fit” INL of this A2D. Report the answer in whole LSB’s (this is for a datasheet so should round up to be conservative). All work should be done in the .xls file itself. Simply turn in your renamed and completed .xls file when you are done. **Hint:** There are functions called SLOPE and INTERCEPT in Excel that can be used to get the best fit coefficients for a set of data.

**Important:** Submit your .xls file *teamX\_A2D\_INL.xls* using the homework #7 dropbox in Learn@UW.

See solution posted on Web.

### 4. (10 points) ADC System Design

A. Assume that a system has to measure an input signal that can vary between 0mV and 400mV. If you are using an ADC with a voltage reference of 2.5V (ADC nominal input range of 0V to Vref) to measure the signal directly, how many bits of resolution do you need to have in order to know the value of the input with a maximum error of 1mV? (Assume that the ADC and voltage reference are ideal.) Then, consider the case where you add an ideal amplifier with a voltage gain of 6 between the signal source and the ADC. What ADC resolution is now required to be able to measure the input with a maximum error of 1mV? Provide calculations to support your answers.

Max error of an ideal ADC is  $\frac{1}{2}$  and LSB. ( $\frac{1}{2}$  a quanta). If your signal is coming straight through un amplified into a ADC with a reference of 2.5V then:

$$Error = \frac{V_{ref}}{2 \cdot 2^n} = \frac{2.5}{2 \cdot 2^n} \leq 1mV \rightarrow 2^n \geq 1250 \rightarrow n \geq 11$$

Now if we are amplifying the input signal by 6 prior to the ADC then 1mV of error at the input (this is called "input referred") is now 6mV of error at the input to the ADC

$$Error = \frac{V_{ref}}{2 \cdot 2^n} = \frac{2.5}{2 \cdot 2^n} \leq 6mV \rightarrow 2^n \geq 209 \rightarrow n \geq 8$$

Many teams did this problem with 1LSB error for an ideal ADC instead of  $\frac{1}{2}$  LSB. I was very lenient on this mistake.