Scott Jackson

CECS 447

11/19/09

Projects #4 and #5

A-to-D and D-to-A Conversion

In this project, the goal is to program the LPC2148 board to use A-to-D and D-to-A conversion. For A-to-D conversion, we use the on-board ADC. For this part of the project, we want to take an analog voltage value and convert it to a digital value and display it on the HyperTerminal using the UART. In order to accomplish this, we must program the A/D Control Register 0 (AD0CR) to do A/D conversion. For our particular project, we want to use the AD0.3 pin, because that is the pin connected to the on-board potentiometer that we can use to adjust the analog voltage. In order to use the pin, we must enable it by setting pin P0.30 to select the AD0.3 pin. We do this by setting bit 28 on PINSEL1 to 1. Once AD0.3 is enabled, we must program the AD0CR. In this register, we set many parameters to make A/D conversion possible. We use the AD0CR to set the input pin for A/D conversion, the ADCLK, the BURST bit, the number of bits used for conversion, the PDN bit, as well as the START and EDGE bits. For this particular project, we are going to use AD0.3 as the input pin, also we want to set the BURST and PDN bits to 1, and the EDGE bit to 0. As for the ADCLK, we want to set this to a maximum of 4.5 MHz or the next closest lower value. The AD clock is set by taking the PCLK value and dividing it down using the CLKDIV bits. For our project, the PCLK is 30 MHz, so we need to divide it by a factor of 7 to get the desired AD clock value. To do this we set the CLKDIV value to 6 because the PCLK is divided by this value plus one, so it will divide it be 7, giving us a ADCLK value of 4.29 MHz which is close to the 4.5 MHz we need. To set all these parameters, we set AD0CR = 0x00210608. Once these parameters are set, we can OR the START bit to tell it to start conversion now, by setting AD0CR |= 0x01000000.

When AD conversion begins, the target value we want is placed in the A/D Global Data Register, which transfers the value to the A/D Channel Register for the pin we enabled. In this case, it is transferred to A/D Channel 3 Data Register (AD0DR3). So this is the register we must read to get the value we want. When we read this register, however, the value is represented as a binary fraction representing the voltage on the AD0.3 pin divided by the voltage on the VREF pin (V/VREF). The read will be between 0-0x3FF. We want a voltage value between 0-3.3V, so in order to get a value in this range, we must multiply the value by 3.3 and divide by 1023 (for 10-bit conversion). The value is also shifted to the left 6 bits, so we must first shift it to the right 6 bits before converting the data. On this is done, we can take the voltage value and translate it to ascii characters so that it can be sent to the UART and outputted on the HyperTerminal. Also, the output to the HyperTerminal is updated in real-time, so if we change the voltage using the potentiometer, we see the corresponding voltage change on the HyperTerminal.

The second part of the project is D-to-A conversion. D-to-A conversion is relatively simple as it only uses one register, the D/A Control Register (DACR) for digital to analog conversion. First we must enable the Aout pin for D-to-A conversion by setting P0.35 to select the Aout pin. We do this by setting bit 19 on PINSEL1 to 1. Once this is done, all we have to do is read the value into the DACR and conversion will begin automatically. For this project, we want to do it output a cosine wave on the oscilloscope, and the voltage value we read into the DACR will be a value that oscillates between 0 and 3.3V. To make the cosine wave, we use a for loop that goes between 0 and 360 (corresponding to the 360⁰ of the unit circle). We then convert this value to radians and enter it in this equation (cos(val1)\*511)+512, to convert it to a value between 0 and 0x3FF. We then must shift the value to left 6 bits because the DACR reads values into bits [15:6] and then read it into the DACR. The DACR will then convert the value to a voltage between 0 and 3.3V, and output the value on the Aout pin. We connect our scrope probe to the Aout pin on the LPC2148, and connect that to the oscilloscrope to see the waveform. Also, we can use the value from the ADC as a delay value to adjust the frequency of the wave. That way we can speed or slow the frequency of the waveform by simply adjusting the potentiometer on the board.