

Madison Area Technical College
Electrical Engineering and Electronics Technology Dept
Microcontrollers

Analog to Digital Conversion

Most microcontroller's A/D circuit may be setup to initiate an interrupt after a successful analog translation. The Microchip PIC 18F4520 has a N=10 bit A/D controller controlled by the registers ADCON0, ADCON1 and ADCON2. The registers used to control the A/D interrupt circuit are INTCON and PIE1. In the PCW-CCS compiler the preprocessing command #INT_AD is used to link the user defined interrupt subroutine to the interrupt circuit.

Answer the following question:

- 1) What is the event that triggers an A/D interrupt routine?

- 2) How many A/D conversion units on the 18F4520?

- 3) How many A/D channels on the 18F4520?

Compile and test the following C code

This code should be compiled with the PCW-CCS compiler. The code will take a sample from analog channel 0 and display its value in the LCD panel connected on PORTD. The LCD connection follows the Boogie Board LCD connection design.

1. Test this code both on the simulator AND in the Boogie Board. Carefully verify each pointer and the mask. You should be able to explain any code line in this code.

```
#use delay (clock = 20000000)
#fuses HS, NOWDT, NOLVP
#include "..\Library\modifiedlcd.h"

int *TRISA=0xF92, *TRISB=0xF93, *ADCON0=0xFC2, *ADCON1=0xFC1;
int *ADCON2 = 0xFC0, *ADRESL = 0xFC3, *ADRESH = 0xFC4;

int16 *AD = 0xFC3;
float Vres = 5.0 / 1023.0;

main(){
    *TRISA = 0x01; // Pin A0 Input
    *TRISB = 0x00;
    *ADCON0 = (*ADCON0 | 0x01) & 0xC3; // Ch0, AD-ON
    *ADCON1 = (*ADCON1 | 0x07) & 0xC7; //
    *ADCON2 |= 0x80;
```

```

    lcd_init();
    while(1){
        *ADCON0 |= 0x02;    // Go get one Analog Input
        while(*ADCON0 & 0x02); // Waiting for translation
        printf(lcd_putc, "f Vin = %f", Vres * (float)(*AD));
        delay_ms(500);
    }
}

```

2. It is good practice to make interrupt subroutines efficient. Avoiding printing and delays inside interrupt subroutines is a good place to start. Compare the following code against the previous code. Discuss the differences with your peers. Test this code both on the simulator and in the Boogie Board.

```

#include <18f4520.h>
#use delay (clock = 20000000)
#fuses HS, NOWDT, NOLVP
#include "..\Library\modifiedlcd.h"
#include "..\Library\Registers4520.h"

int16 *AD = 0xFC3;
float Vres = 5.0 / 1023.0;

#INT_AD
void int_ad_isr(){
    printf(lcd_putc, "f Vin = %f", Vres * (float)(*AD));
}

main(){
    *TRISA = 0x01; // Pin A0 Input
    *TRISB = 0x00;
    *ADCON0 = (*ADCON0 | 0x01) & 0xC3; // Ch0, AD-ON
    *ADCON1 = (*ADCON1 | 0x07) & 0xC7; //
    *ADCON2 |= 0x80; //
    *INTCON |= 0xC0; // Peripheral and Global Enable
    *PIE1 |= 0x40; // A/D Int Enable
    lcd_init();
    *ADCON0 |= 0x02; // Start the A/D
    while(1){
    }
}

```

3. Change the voltage value in your potentiometer without turning off your simulation. When you change the potentiometer the A/D value does not change. Why?
4. Modify the seed code to move the A/D input from A/D Channel 0 to A/D Channel 5.