Introduction

The ATMEL AT89LP microcontrollers feature on-chip Flash, enhanced single cycle execution, and an integrated ADC/DAC module. This application note describes how to initialize the on-board ADC and gives useful suggestions in improving the sampling accuracy.

Analog – To – Digital Converter

For our AT89LP devices, our ADC supports up to 8 input analog channels that can be configured in single ended mode or differential mode. In addition, resolution of the AT89LP ADC can be selected to a standard 8-bit resolution or a high precision 10-bit resolution.

As shown in Figure 2-1, the AT89LP51ED2 will be shown with basic passive components to allow users to manually adjust the voltage input and display the value through an array of LEDs.
The analog input is controlled by using a variable resistor (R1) that can be increased / decreased by the user. Since the device is being used in single ended mode, the resistor (R1) will be able to deliver a signal that can be from one end (0V) to the other (Vcc). The push-button switch allows to the system to trigger the on-board timer which when times out, triggers the conversion of the ADC. Port 2 of the AT89LP51ED2 is connected to an array of LEDs to give a visual 8-bit representation of the newly converted output. A status LED (D9) has been added to P3.5 of the AT89LP51ED2 to show when the ADC is busy/done.

The AT89LP51ED2 I/O ports have 4 unique modes that can be configured:

- Quasi-bidirectional
- Push-Pull Output
- Input Only (High Impedance)
- Open-Drain

For this application, that analog input channel, P0 will be configured as “Input Only” while all others will be set to quasi-bidirectional. For P0, “Input Only” offers the best port structure that rejects outside EMI that could cause accuracy of the ADC to be compromised.
As stated earlier, Timer 0 of the AT89LP51ED2 will be used to trigger an ADC conversion. The software will count from 0 to 255 after which the Timer 0 overflag will set and cause the ADC interrupt to be serviced. The ADC interrupt routine will move the newly converted value from the DADH register to Port 2 and set the status led (D9).

Timer 0 will begin counting as soon as the push-button switch (SW1) is pressed. The software is waiting for a pulse to be generated on the P1.1 before it starts Timer 0 and ADC conversion starts after which the new value will be displayed on Port 2. Each subsequent pulse detected on P1.1 will issue another conversion of the ADC of the AT89LP51ED2 device.

### Noise Considerations

Generally, digital systems generate EMI which could cause accuracy issues when analog values are being samples and converted. If accuracy of the system is critical, the following are useful tips in reducing the EMI in most applications.

- **Disable ALE**: In normal operation, ALE is pulsed at a constant rate of 1/3 the oscillator frequency in compatibility mode (1/2 the oscillator speed in “Fast Mode”). This pin is used to latch the low byte of the address during access to external memory. By setting the DISALE bit in the AUXR (Auxiliary Control) register, ALE will only be pulsed when external memory access is performed.

- **IDLE Mode**: The mode is one of two power saving modes our 8051 device come equipped with (other being Power-Down). Idle mode stops the internal CPU clock and holds the values of the RAM, stack pointer, program counter, program status word, and accumulator. In addition, all ports pins hold the logic state at the time Idle was invoked. Idle mode of the device is achieved by setting IDL bit in the PCON (Power Control) register.

- **Analog Channel Input**: The user can disconnect the analog inputs from the ADC to further decrease the amount of EMI that is being absorbed by the device. By clearing the ACON bit in the DADI (Input Control Register), the analog inputs are disconnected to from the ADC to not allow any type of noise to affect accuracy of the ADC during conversion.
Example Source Code

; ADC Example for AT89LPxx devices
; Analog Input on P0.0, Single Ended Mode, 8-bit resolution
; External Reference Vdd Select
; Internal RC Clock Selected
; Hardware Trigger via Timer 0 overflow
; Converted Result will be in the DADH Register

org 0x0063 ; Analog Interrupt Routine
mov P2, DADH ; Output value to port
setb P4.7 ; ADC completed
reti

start:
mov P0M0, #0xFF ; P0 tristate
mov P0M1, #0x00 ; Other Ports bidirectional
mov P1M0, #0x00
mov P2M0, #0x00
mov P3M0, #0x00
mov P4M0, #0x00
orl AUXR, #0x01 ; Disable ALE
mov TMOD, #0x01 ; T0 Mode1
mov IEN0, #0x82 ; EA
mov IEN1, #0x20 ; EAD
mov DADC, #0x18 ; Enable, Left-Adjust, Int RC Osc
mov DADI, #0x10 ; IREF=0, DIFF=0, Channel 0

convert:
mov TCON, #0x00 ; Initialize Timer
mov TH0, #0xFF ; Reload Values
mov TL0, #0x00
mov DADH, #0x00 ; Clear ADC result registers
mov DADL, #0x00
orl DADI, #0x80 ; Connect
clr P4.7 ; ADC ready
jb P1.1, $ ; Wait for user to issue pulse
jnb P1.1, $
setb TR0 ; Overflow will trigger ADC
mov PCON, #0x11 ; Enter Idle
clr TR0
anl DADI, #0x7F ; Disconnect
sjmp convert

END