Đây là một ứng dụng bộ chuyển đổi analog số ADC của AVR để đo nhiệt độ, sau đó giá trị nhiệt độ đo được sẽ được cập nhật thông qua một chương trình trên máy tính.  
  
  
  
  
  
Chương trình cho avr- atmega8  
  
;-------------------------------------------------------------------------  
;Read ADC channel  
;  
;-------------------------------------------------------------------------  
  
  
.nolist  
.include "m8def.inc"  
.list  
  
.def rlpm = R0  
  
.def rbin1L = R10  
.def rbin1H = R11  
.def rbin2L = R12  
.def rbin2H = R13  
  
.def mpr = R16  
.def var1 = R17  
.def var2 = R18  
.def var3 = R19  
  
  
;constants  
.equ fq = 8000000 ;xtal frequency, CKSEL3..0 V\_VVV  
.equ baud = 19200 ;baud rate  
.equ baudgen = (fq/(16\*baud))-1 ;baud divider  
  
  
;Reset-/Interrupt Vector  
.CSEG  
.org 0x0000  
rjmp start  
  
  
;---------------  
;Subroutine for string transfer  
txstr: sbis USR,UDRE ;wait Tx buff empty  
rjmp txstr  
lpm ;read next char from prog mem  
and rlpm,rlpm ;null = end of string  
brne txsend  
ret  
txsend: lpm ;read the same char again  
out UDR,rlpm ;Tx character read  
adiw ZL,1 ;point to next char in memory  
rjmp txstr  
  
;---------------  
;Transmit character  
txchar: sbis USR,UDRE  
rjmp txchar  
out UDR,mpr  
ret  
  
;---------------  
;Receive character  
rxchar: sbis USR,RXC  
rjmp rxchar  
in mpr,UDR  
ret  
  
;---------------  
;Read ADC (var2 + 1) times and get average  
readadcave: sbi ADCSR,ADSC  
ldi var2,3  
  
\_adcave1: sbis ADCSR,ADIF  
rjmp \_adcave1  
sbi ADCSR,ADIF ;clear flag by setting bit  
in mpr,ADCL  
mov rbin2L,mpr  
in mpr,ADCH  
mov rbin2H,mpr  
  
\_adcave2: sbi ADCSR,ADSC  
\_adcave3: sbis ADCSR,ADIF  
rjmp \_adcave3  
sbi ADCSR,ADIF ;clear flag by setting bit  
in mpr,ADCL  
mov rbin1L,mpr  
in mpr,ADCH  
mov rbin1H,mpr  
  
add rbin2L,rbin1L ;add adc(n)+adc(n-1)  
adc rbin2H,rbin1H  
dec var2  
brne \_adcave2  
lsr rbin2H  
ror rbin2L  
lsr rbin2H  
ror rbin2L  
\_adcend: mov rbin1H,rbin2H  
mov rbin1L,rbin2L  
ret  
  
;-------------------------------------------------------------------------  
;Start of program  
start: ldi mpr,low(RAMEND)  
out SPL,mpr  
ldi mpr,high(RAMEND)  
out SPH,mpr  
ldi mpr,baudgen ;baud generator  
out UBRRL,mpr ;set divider  
ldi mpr,0b00011000 ;enable TX and RX  
out UCR,mpr ;to UART Control Register  
  
  
;PORTB IIOOOIII  
ldi mpr,0b00111000  
out DDRB,mpr  
ldi mpr,0b11000111 ;pull-up input  
out PORTB,mpr  
  
;PORTC --IIIIII  
ldi mpr,0b00000000  
out DDRC,mpr  
ldi mpr,0b00000000 ;cut pull-up input  
out PORTC,mpr  
  
;PORTD IIIOOOIO  
ldi mpr,0b00011101  
out DDRD,mpr  
ldi mpr,0b11100010 ;pull-up input  
out PORTD,mpr  
  
sbi PORTD,4  
ldi var1,2  
delaystartup: rcall delay500ms  
dec var1  
cpi var1,0  
brne delaystartup  
  
intro: ldi mpr,12  
rcall txchar  
ldi ZH,high(adcparstr\*2)  
ldi ZL,low(adcparstr\*2)  
rcall txstr  
  
ldi mpr,0b10000010 ;set ADC enable  
out ADCSR,mpr  
  
mainloop: sbis USR,RXC ;test RXC bit  
rjmp mainloop ;none available, wait  
  
in mpr,UDR  
cpi mpr,'>'  
brne \_cmdr  
rcall txchar  
rjmp mainloop  
\_cmdr: cpi mpr,'r'  
brne mainloop  
  
readadc: cbi PORTD,4  
ldi mpr,13  
rcall txchar  
ldi mpr,'D'  
rcall txchar  
  
  
\_readadc0: ldi mpr,0b01000000 ;set ADMUX int ref ch0  
out ADMUX,mpr  
rcall readadcave  
rcall sendvalue  
  
\_readadc1: ldi mpr,0b01000001 ;set ADMUX int ref ch1  
out ADMUX,mpr  
rcall readadcave  
rcall sendvalue  
  
\_readadc2: ldi mpr,0b01000010 ;set ADMUX int ref ch2  
out ADMUX,mpr  
rcall readadcave  
rcall sendvalue  
  
\_readadc3: ldi mpr,0b01000011 ;set ADMUX int ref ch3  
out ADMUX,mpr  
rcall readadcave  
rcall sendvalue  
  
\_readadc4: ldi mpr,0b01000100 ;set ADMUX int ref ch4  
out ADMUX,mpr  
rcall readadcave  
rcall sendvalue  
  
\_readadc5: ldi mpr,0b01000101 ;set ADMUX int ref ch5  
out ADMUX,mpr  
rcall readadcave  
rcall sendvalue  
  
\_readadc6: ldi mpr,0b01000110 ;set ADMUX int ref ch6  
out ADMUX,mpr  
rcall readadcave  
ldi mpr,0xFF ;these 4 lines to disable ADC6  
mov rbin1H,mpr ;(in DIP version)  
ldi mpr,0xFF ;ADC6 = 0xFFFF  
mov rbin1L,mpr ;dummy value  
rcall sendvalue  
  
\_readadc7: ldi mpr,0b01000111 ;set ADMUX int ref ch7  
out ADMUX,mpr  
rcall readadcave  
ldi mpr,0xFF ;these 4 lines to disable ADC7  
mov rbin1H,mpr ;(in DIP version)  
ldi mpr,0xFF ;ADC7 = 0xFFFF  
mov rbin1L,mpr ;dummy value  
rcall sendvalue  
  
sbi PORTD,4  
  
rjmp mainloop  
  
  
adcparstr: .db "ADC parameter :",13,10,0  
  
  
;---------------  
;ASCII to 2HEX  
ascii2hex: push mpr  
push mpr  
swap mpr  
andi mpr,0x0F  
cpi mpr,10 ;nibble > 9?  
brcs \_ascii2hex1 ;no  
ldi var1,7 ;add 7 to get hex A to F  
add mpr,var1  
\_ascii2hex1: ldi var1,'0'  
add mpr,var1  
rcall txchar  
pop mpr  
andi mpr,0x0F  
cpi mpr,10 ;nibble > 9?  
brcs \_ascii2hex2 ;no  
ldi var1,7 ;add 7 to get hex A to F  
add mpr,var1  
\_ascii2hex2: ldi var1,'0'  
add mpr,var1  
rcall txchar  
pop mpr  
ret  
  
;---------------  
;Delay 500ms @8mhz  
delay500ms: push R23  
push R24  
push R25  
ldi R23,0x6B  
\_wg500loop0: ldi R24,0x46  
\_wg500loop1: ldi R25,0xB1  
\_wg500loop2: dec R25  
brne \_wg500loop2  
dec R24  
brne \_wg500loop1  
dec R23  
brne \_wg500loop0  
pop R25  
pop R24  
pop R23  
ret  
  
;---------------  
;Bin2ToBCD5  
;converts a 16-bit-binary to a 5-digit-BCD  
;in rbin1H:rbin1L  
  
sendvalue: mov mpr,rbin1H  
rcall txchar  
mov mpr,rbin1L  
rcall txchar  
ret  
  
;sendvalue:  
bin2tobcd5: push rbin1H ;save number  
push rbin1L  
ldi mpr,high(10000) ;start with 10.000  
mov rbin2H,mpr  
ldi mpr,low(10000)  
mov rbin2L,mpr  
rcall bin2todigit ;calculate digit  
  
ldi mpr,high(1000) ;next with 1.000  
mov rbin2H,mpr  
ldi mpr,low(1000)  
mov rbin2L,mpr  
rcall bin2todigit ;calculate digit  
rcall txchar  
  
ldi mpr,high(100) ;next with 100  
mov rbin2H,mpr  
ldi mpr,low(100)  
mov rbin2L,mpr  
rcall bin2todigit ;calculate digit  
rcall txchar  
  
ldi mpr,high(10) ;next with 10  
mov rbin2H,mpr  
ldi mpr,low(10)  
mov rbin2L,mpr  
rcall bin2todigit ;calculate digit  
rcall txchar  
mov mpr,rbin1L ;remainder are ones  
ori mpr,0x30  
rcall txchar  
pop rbin1L  
pop rbin1H  
ret  
  
;---------------  
;Bin2ToDigit  
;converts one decimal digit by continued substraction of a BCD  
bin2todigit: clr mpr ;digit count is zero  
\_bin2todigita: cp rbin1H,rbin2H ;number bigger than decimal?  
brcs \_bin2todigitc ;MSB smaller than decimal  
brne \_bin2todigitb ;MSB bigger than decimal  
cp rbin1L,rbin2L ;LSB bigger or equal decimal  
brcs \_bin2todigitc ;LSB smaller than decimal  
\_bin2todigitb: sub rbin1L,rbin2L ;substract LSB decimal  
sbc rbin1H,rbin2H ;substract MSB decimal  
inc mpr ;increment digit count  
rjmp \_bin2todigita ;next loop  
\_bin2todigitc: ori mpr,0x30  
ret  
  
  
;end

Sau đây là file hex nạp cho vi điều khiển và phần mềm trên máy tính  
Do file hơi lớn nên chia làm 2 phần PCphanemmmaytinh\_1 và PCphanemmmaytinh\_2, các bạn tải về rồi giải nén 2 file này vào cùng 1 thư mục.

Attached Files

* [PCphan mem may tinh\_1.rar (537.7 KB, 276 lượt xem)](http://www.dientuvietnam.net/forums/filedata/fetch?id=1333251)
* [PCphan mem may tinh\_2.rar (616.3 KB, 269 lượt xem)](http://www.dientuvietnam.net/forums/filedata/fetch?id=1333252)
* [file\_HEX\_templog.rar (678 Bytes, 71 lượt xem](http://www.dientuvietnam.net/forums/filedata/fetch?id=1333253)