Sample a waveform on PAD1 at approximately 125Hz (8ms)
Each time take 4 quick samples and record the average.
I'll use the RTI system to generate timing needed.
Assume a 4.0 MHZ Oscillator => E=2MHz
(Note: Simulator assumes a 16MHz Oscillator => E=8MHz)

Symbol definitions

ATDCTL2 EQU $0062 ; A/D Control Register 2
ATDCTL5 EQU $0065 ; A/D Control Register 5
ATDSTATH EQU $0066 ; A/D Status Register (High)
ATDSTATL EQU $0067 ; A/D Status Register (Low)
ATDR0H EQU $0070 ; A/D Result Register 0 (High)
ATDR1H EQU $0072 ; A/D Result Register 1 (High)
ATDR2H EQU $0074 ; A/D Result Register 2 (High)
ATDR3H EQU $0076 ; A/D Result Register 3 (High)
RTICTL EQU $0014 ; RTI Interrupt Control Register
RTIFLG EQU $0015 ; RTI Interrupt Flag Register
RAM EQU $0900 ; Start of Ram
STK EQU $0B00 ; User Stack #### MOVED to $B000!!
RAMPROG EQU $0970 ; Program space in RAM
IV_RTI EQU $FFF0 ; RTI Interrupt Vector Location
IPV_RTI EQU $0833 ; RTI Interrupt Pseudo-Vector Location

Bit definitions
BIT7 EQU %10000000
INV7 EQU %01111111
BIT0 EQU %00000001
RTIE EQU BIT7
RTIF EQU BIT7
INV_RTIF EQU INV7

Constants
NSAMP EQU 100
PAD1 EQU %00000001 ; Value for ATDCTL5:

   S8CM=SCAN=MULT=CD=CC=CB=0, CA=1

Data Area
ORG RAM
CNT DS.B 1
FINISH DS.B 1
POINTER DS.W 2
EEG DS.B NSAMP

Main Program

Data Area
ORG RAM
CNT DS.B 1
FINISH DS.B 1
POINTER DS.W 2
EEG DS.B NSAMP

Initialization

Main Program

Initialization
*-----------------------------------------------------
* Turn on A/D System
LDA #%10000000 ; Bit 7 (ADPU) of ATDCTL2
STA ATDCTL2 ; ..turns on A/D system
* Wait 100 microsec = 200 E-cycles (at 0.5us/E-clock)
* for charge pump to stabilize.
* Can't really calculate this precisely on 68HC12. Too much is best.
LDA #100 ; 1 E cycles
DEC
BNE WAIT1 ; 1 E cycles
(WAIT1)
* Set up RTI for 8.192 ms
LDA #00000010 ; Interrupt at 8.192 ms, 0's (E=2MHz)
STA RTICTL ; ..(2.048 ms if E=8MHz)
* Initialize Constants
LDA #NSAMP ; CNT has the number of samples
STA CNT
LDY #EEG ; Y will point to the EEG output table
STY POINTER
CLR FINISH ; When FINISH = 1, done
* Enable RTI
BCLR RTIFLG INV_RTIF ; Clear RTIF
BSET RTICTL RTIE ; Enable RTI interrupt
CLI
HERE
LDA FINISH ; When FINISH = 0, not done;
BNE HERE ; when FINISH = 1, done
BCLR RTICTL RTIE ; Disable RTI interrupt
DONE
BRA DONE
*-----------------------------------------------------
** RTI INTERRUPT SERVICE ROUTINE
** * Collect a sample
** * Get sample counter & exit if count=0
** * Store sample & decrement sample counter
** * Clear RTI flag
***********************************************
ORG $09C0
RTI_ISR
BRCLR RTIFLG RTIF, BYE ; Ignore illegal interrupt
BCLR RTIFLG INV_RTIF ; Clear RTIF
JSR SAMPLE ; Get the sample value (stored in B)
LDY POINTER ; Save the value in the table
STA 0,Y
DEC CNT ; If Cnt = done; else not done
BNE NotDone ;
INC FINISH ; done with sampling so set stop flag
BRA BYE
NotDone LDY POINTER ; Calculate address for next sample
INY
STY POINTER
BYE
***********************************************
** Subroutine to sample the A/D PAD1
** Result is in register B
***********************************************
SAMPLE LDA #PAD1 ; S8CM=SCAN=MULT=CD=CC=CB=0, CA=1
STAA ATDCTL5 ; Using PAD1

* Wait for 4 new samples to be available
WAIT2 BRCLR ATDSTATH BIT7, WAIT2

* Get the average of the four values read and return the average in B
CLRA
LDAB ATDR0H ; Get new data into D
XGDX ; Put data into X

* Add 3 other values read to first value (then divide by 4 to get average)
LDAB ATDR1H
ABX
LDAB ATDR2H
ABX
LDAB ATDR3H
ABX
XGDX ; Put sum into D
LSRD ; Divide by 2 twice to get a divide by 4
LSRD ; (which gives us the average in B)

RTS