

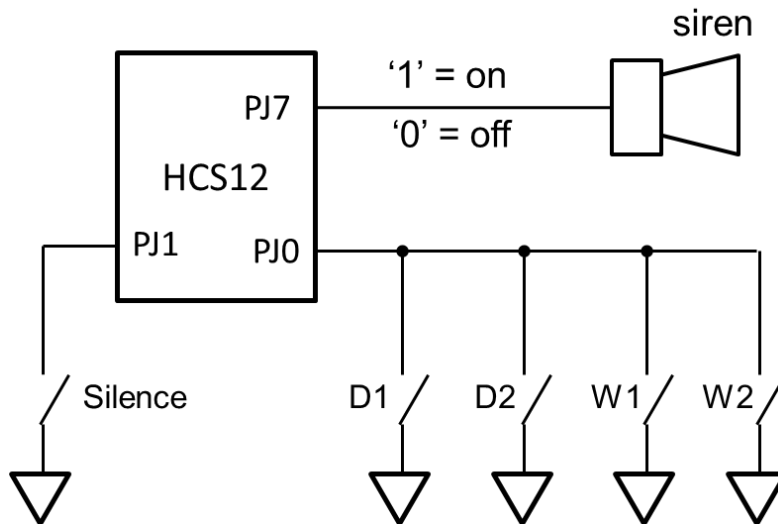
**Problem 11.1:**

Write a subroutine called *echo* to read the 8-bit data on Port S, add a constant to this data and output the sum on Port T. The constant to be added is passed to the subroutine in accumulator B. Check to see that the data on the physical Port T pins is correct. If it is not, complement the constant in accumulator B as a way of notifying the main program that there is something wrong with the external wiring connected to Port T. You can assume that the main program has already set up Port S for input and Port T for output. Save any registers (other than B) that you use.

### Problem 11.2:

An HCS12 microcontroller is to be used to monitor and control a home security system. There are two doors and two windows in the house, each of which has a switch which closes when the door (or window) is opened. These switches are connected in parallel to pin PJ0 as shown in the Figure. A siren is connected to pin PJ7. The siren is activated by setting pin PJ7 to a '1'. The siren is silenced by setting PJ7 back to '0'. A *silence* button is connected to pin PJ1.

Write a main program starting at \$4000 and an interrupt service routine to activate the siren when any of the doors and/or windows are opened. The main program should set up Port J and its interrupts and then branch to another location called "useful\_work". The activation and the silencing of the siren should be achieved using interrupts. Once the main program has branched to "useful\_work", it plays no part in the activation and/or silencing of the siren. Once the siren has been activated, the siren is silenced by first closing the door (or window) that had been opened and then pushing the *silence* button. The code should not allow the siren to be silenced as long as any of the doors or windows are open. The Interrupt Vector Address of Port J is \$FFCE.



### **Solution 11.1:**

; Use register PTIT to examine the state of the port T pins to check that there is no  
; overload or short condition by comparing the pins to the actual data that was output

```
echo:  psha                ; save A on stack
       ldaa  PTS           ; load port S data into A
       aba                    ; add constant to A
       staa  PTT          ; output result to port T
       cmpa  PTIT         ; compare with data on physical pins
       beq   done         ; if equal – OK
       comb                    ; if not - complement acc B
done:  pula                ; restore A
       rts
```

## Solution 11.2:

- ; Need to setup PJ0 and PJ1 with pull-up resistor
- ; Need to enable interrupts on PJ0 and PJ1

```
ORG    $4000

main:  movb  #$80, DDRJ      ; set PJ7 to output, PJ0 and PJ1 to input
       bclr  PTJ, $80       ; clear PJ7 to disable siren
       movb  #$03, PERJ     ; set pull enable on PJ0 and PJ1
       movb  #$00, PPSJ     ; set to PJ0 and PJ1 to pull UP
       bset  PIFJ, $03      ; clear interrupt flags on PJ0 and PJ1
       movw  #PTJ_isr, $FFCE ; set up port J interr. vector with addr of isr
       bset  PIEJ, $03      ; enable interrupts on PJ0 and PJ1
       cli   ; turn on global interrupts
       bra   useful_work
```

- ; For interrupt service routine, need to see if interrupt caused by PJ0 or PJ1
- ; Either way, need to clear appropriate interrupt flag
- ; If interrupt caused by silence button, need to check doors/windows are now closed

```
PTJ_isr: brset  PIFJ, $02, silence ; branch if interr caused by silence button
         brset  PIFJ, $01, alarm   ; branch if interr caused by open door/win
         rti   ; some other interrupt not expected

alarm:   bset   PIFJ, $01          ; clear PJ0 interrupt flag
         bset   PTJ, $80          ; turn on siren
         rti

silence: bset   PIFJ, $02          ; clear PJ1 interrupt flag
         brclr  PTJ, $01, done     ; if still open, do not silence
         bclr  PTJ, $80          ; silence siren

done:    rti
```