Another example of using a subroutine

Using a subroutine to wait for an event to occur, then take an action.

- Wait until bit 7 of address `$00C4` is set.
- Write the value in ACCA to address `$00C7`.

```
; This routine waits until the HC12 serial
; port is ready, then sends a byte of data
; to the HC12 serial port

putchar:      brclr    $00C4,#$80
              staa     $00C7
              rts
```

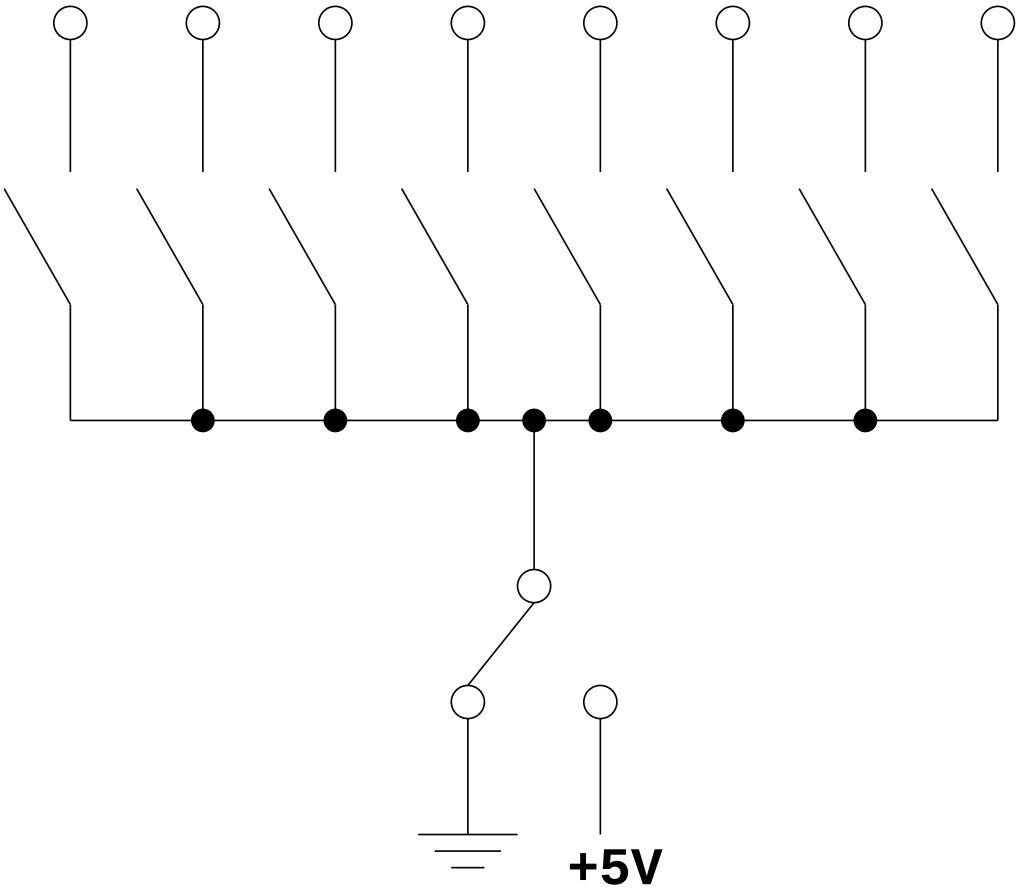
- Program to send the word `hello` to the HC12 serial port

```
; Program fragment to write the word "hello" to the
; HC12 serial port

              ldx      $str
loop:         ldaa     1,x+    ; get next char
              beq      done    ; char == 0 => no more
              jsr      putchar
              bra      loop
              swi

str:          dc.b     "hello",0
```

Using DIP switches to get data into the HC12

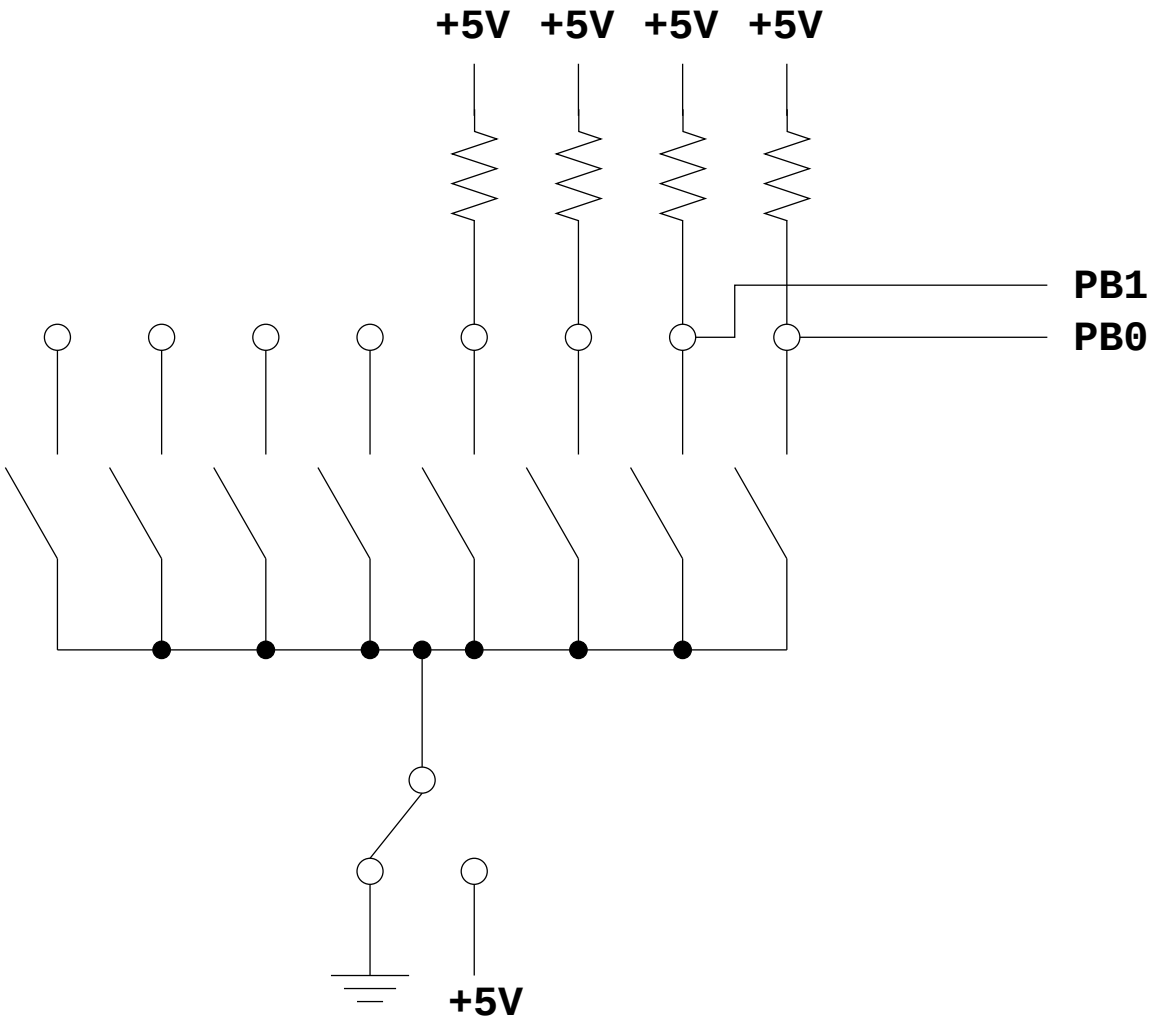
- DIP switches make or break a connection (usually to ground)

DIP Switches on Breadboard



- To use DIP switches, connect one end of each switch to a resistor
- Connect the other end of the resistor to +5 V
- Connect the junction of the DIP switch and the resistor to an input port on the HC12

Using DIP Switches



- When the switch is open, the input port sees a logic 1 (+5 V)
- When the switch is closed, the input sees a logic 0 (0 V)

Looking at the state of a few input pins

- Want to look for a particular pattern on 4 input pins
  - For example want to do something if pattern on PB3-PB0 is `0110`
- Don't know or care what are on the other 4 pins (PB7-PB4)
- Here is the wrong way to do it:

```
        ldaa    PORTB
        cmpa    #b0110
        beq     task
```

- If PB7-PB4 are anything other than `0000`, you will not execute the task.
- You need to mask out the Don't Care bits **before** checking for the pattern on the bits you are interested in

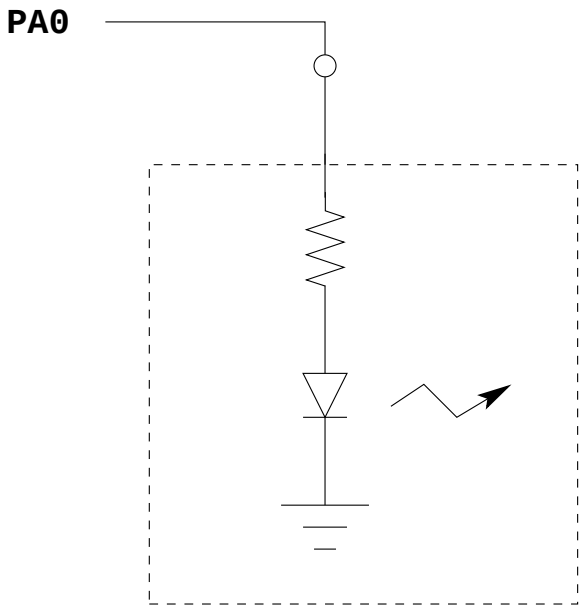
```
        ldaa    PORTB
        anda    #b00001111
        cmpa    #b00000110
beq     task
```

- Now, whatever pattern appears on PB7-4 is ignored

Using an HC12 output port to control an LED

- Connect an output port from the HC12 to an LED.

Using an output port to control an LED

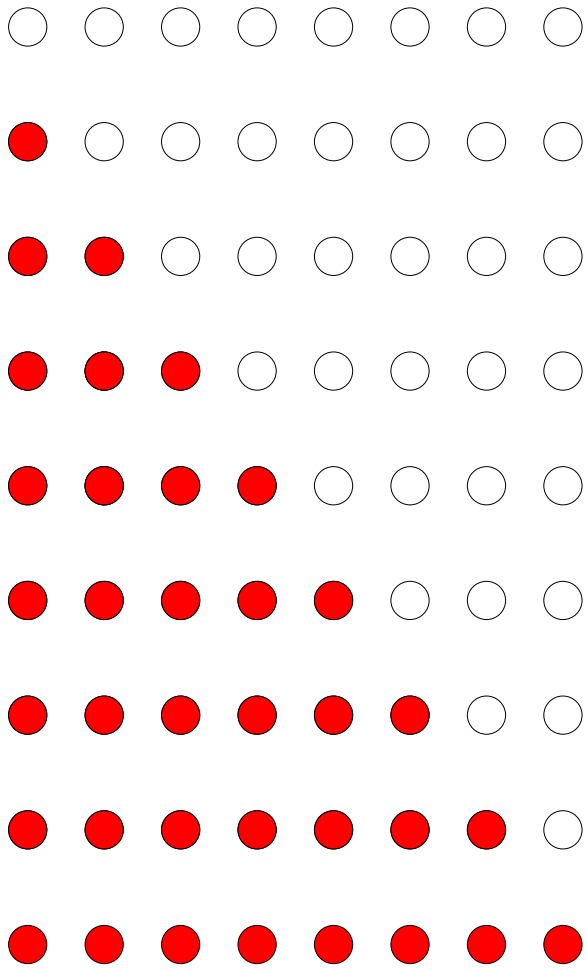


Resistor, LED, and ground connected internally inside breadboard

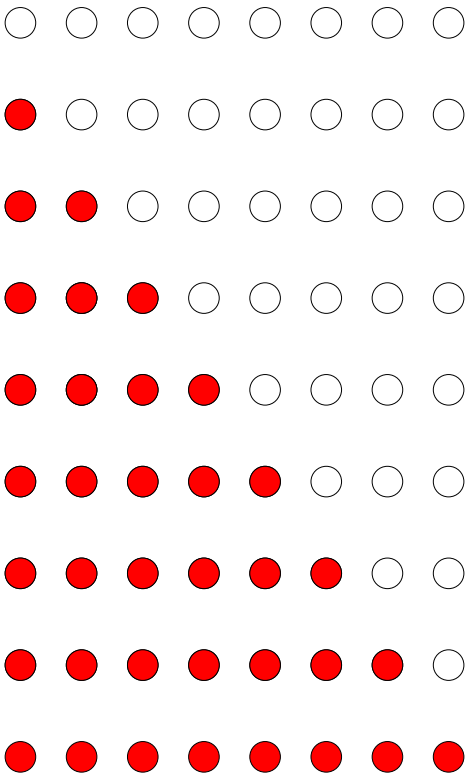
When a current flows through an LED, it emits light

Making a pattern on a set of LEDs

- Want to generate a particular pattern on a set of LEDs:

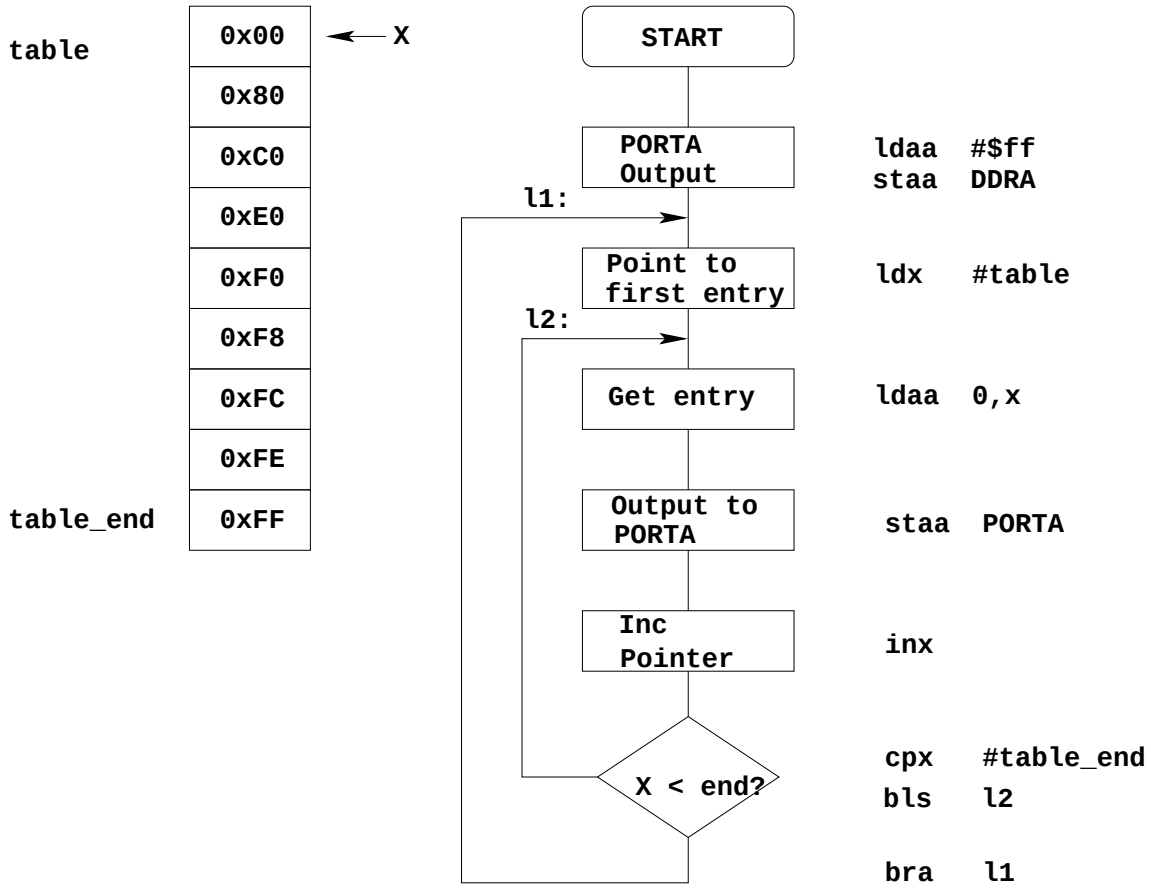


- Determine a number (hex or binary) which will generate each element of the pattern
- Put these numbers in a table
- Go through the table one by one to display the pattern
- When you get to the last element, repeat the loop




```
Set up a table.
Successively load data from table
Start over again when at end of tab
```

Flowchart to display a pattern of lights on a set of LEDs



```
; Program using subroutine to make a time delay

prog:       equ     $0800
data:       equ     $0800
stack:      equ     $0A00
PORTA:      equ     $0000
DDRA:       equ     $0002


CODE:       section .text
            org     prog

            lds     #stack      ; initialize stack pointer
            ldaa    #$ff        ; Make PORTA output
            staa    DDRA        ;   0xFF -> DDRA
l1:         ldx     #table      ; Start pointer at table
l2:         ldaa    1,x+        ; Get value; point to next
            staa    PORTA       ; Update LEDs
            jsr     delay       ; Wait a bit
            cpx     #table_end  ; More to do?
            bls     l2          ; Yes, keep going through
                                ;   table
            bra     l1          ; At end; reset pointer

DATA:       section .data
            org     data
table:      dc.b    $00
            dc.b    $80
            dc.b    $C0
            dc.b    $E0
            dc.b    $F0
            dc.b    $F8
            dc.b    $FC
            dc.b    $FE
table_end:  dc.b    $FF
```

```
; Subroutine to wait for 100 ms

delay:      psha
            pshx
            ldaa    #250
loop2:      ldx     #800
loop1:      dex
            bne     loop1
            deca
            bne     loop2
            pulx
            pula
            rts
```