Parallel Input/Output

N. Narasimhamurthi

1 Objective

To become familiar with hardware interfacing and input/output. Also introduces polling timer overflow flag.

2 Getting started

It is extremely important that you become familiar with your HC11. Since most of you will be using the `CMD11E1` board made by Axiom Manufacturing, this write-up is specific to the board.

1. Locate the jumper `JP13` and make sure that it is open.

2. Locate the MCU port. This is a dual row 17-pin Berg style connector. Locate pin #1 on the port. This is identified with the number 1 on the front of the board. On the other side (solder side), pin #1 is identified by a square solder.

3. Connect a ribbon cable to the port. Use a continuity tester to locate and identify the following pins: pin 1 (PA0), pin 5 (PA4), pin 6 (PA5), pins 9 and 10 (5 Volts), pins 11 and 12 (Ground). (See page 15 of the User's guide that came with your board.)

4. Make two test light emitting diode probes. It is a good idea to have several probes handy. To make a test probe, connect a 3.3K resistor to the **anode** of a light emitting diode (See figure 1). Use a light emitting diode with an operating voltage of 1.7 volts and a current rating in the 1-5 mA range. If you use an light emitting diode with 20+ mA operating current, the light emitting diode would be dim when it lights up and you may have to look carefully to see if it is on. Test the probe by connecting the free end of the resistor to pin 9 and the free end of the light emitting diode to pin 11. The light emitting diode should light-up. If not, the chances are you have

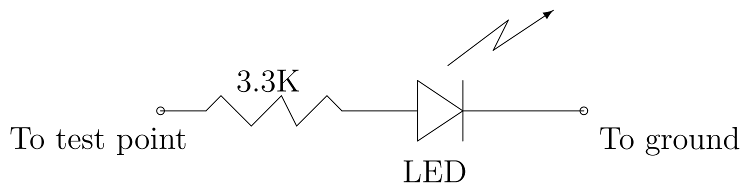

Figure 1: *Test Probe. Make sure that the resistor is connected to the anode of the light emitting diode. The longer lead of the light emitting diode is the anode. To test a pin, connect the cathode to ground and the resistor to the pin.*

connected the resistor to the cathode of the light emitting diode. Redo your circuit by connecting the resistor to the anode.

3 Experiment

3.1 Hardware connection

1. **Disconnect the power to HC11.**
2. Connect a (LEFT) test probe between pin 5 and pin 11. The cathode of the light emitting diode should be connected to pin 11 and the free end of the resistor to pin 5.
3. Connect a (RIGHT) test probe between pin 6 and pin 12. The cathode of the light emitting diode should be connected to pin 12 and the free end of the resistor to pin 6.
4. Label the two light emitting diodes as LEFT and RIGHT so that it is easy to identify them (you can use a magic tape and small pieces of paper).
5. Connect the input circuit as shown in figure 2.
6. Check and recheck all your connections before you connect the power to the HC11.

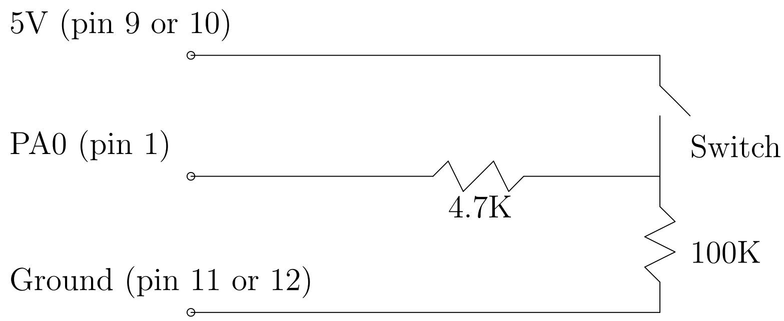

Figure 2: *Typical input connection*

3.2 Theory of operation

Port A in HC11 appears at memory location `$1000`. In port A, the bits 0,1 and 2 (referred to as PA0, PA1 and PA2) are inputs. They appear as pins 1, 2, and 3 of the MCU port (this is specific to the Axiom Board. If you use a different board, check the documentation that came with your board). Thus if you read the memory location `$1000`, then the value in the bits depends on the voltage applied to the corresponding pins. If the voltage is 5 Volts, the bit will be one, and if the voltage is zero, the bit would be zero.

Bits 3,4,5 and 6 of port A (referred to as PA3, PA4, PA5 and PA6) are outputs. This means your program can control the voltage in the corresponding pins 4,5,6 and 7 by writing a 1 or a zero to the appropriate bit in location `$1000`.

3.3 Experiment 1

Use memory modify, `MM $1000`, command to modify the location `$1000`. Change the value in the location to `$00`, `$10`, `$20`, and `$30`. (Note: When you communicate directly with the HC11 using BUFFALO commands, you do not type the `$`.) After each change, look at the state of the light emitting diodes and write down what you see. Provide a brief explanation of what you see.

Using the methods you learnt in earlier labs, write a program that will read the keyboard and depending on what the user types, perform the following:

Key	Action
`Q` or `q`	Turn on the LEFT led. The state of other led should not change.
`Z` or `Z`	Turn off the LEFT led. The state of other led should not change.
`E` or `e`	Turn on the RIGHT led. The state of other led should not change.
`C` or `c`	Turn off the RIGHT led. The state of other led should not change.

3.4 Experiment 2

Use memory dump to see the contents of `$1000`. Close the input switch and dump the contents of location `$1000`. Repeat the experiment with the switch open. Write down what you see and provide a brief explanation of your observation.

Write a program that will, in an infinite loop, print either `CLOSED` or `OPEN`, depending on the state of the switch. The program should continually monitor the state of the switch, and print `CLOSED` is the switch is closed. Or else it should print `OPEN`.

4 Timing

In this part of the lab, we will monitor the `TOF` flag. This flag is controlled by the *free running counter*. The free running counter is a 16 bit counter that counts the clock ticks. It is set to `$0000` on power up. It counts up to `$FFFF`and then rolls over to `$0000` and the process is repeated until you power off the HC11. Every time the counter rolls over, it sets the `TOF` flag. This flag is not automatically reset and it is your responsibility to reset it in your code depending on your need. To reset the flag you have to write a 1 to it [1].

Your HC11 most likely uses a 8 MHz crystal which means that the processor speed is 2 MHz, or you get 2×10^6 clock ticks every second. The free running counter rolls over every 2^{16} ticks, or you get

$$\frac{2 \times 10^6}{2^{16}} = 30.52 \;\; \text{overflows every second}$$

We can use this to create a $\frac{30.52}{2}$ Hz square wave by toggling an output pin every time counter rolls over. Here is the code (you have to fill in the details!)

```
; Various defines go here ...
       ORG $3000 don't forget the $
ME     FCC /Your name/
       FCB 10
       FCC /ECE 372/
       FCB 10
       FCC /Date the program was last changed/
       FCB 10, 10, 4

       ORG $2000 DONT FORGET THE $
       LDX #ME
```

[1] This is true of all HC11 flags. You reset a flag, i.e. make it go to zero, by writing a 1 to it!

```
       JSR OUTSTRG ; MAKE SURE YOU HAVE EQU FOR OUTSTRG

LOOP1
; CLEAR THE FLAG. NO HARM IS DONE IF IT IS ALREADY CLEARED
       LDAA #%10000000
       STAA TFLAG12

;WAIT FOR THE FLAG TO BE SET
LOOP2
      LDAA TFLAG2
      ANDA #%10000000
      BEQ LOOP2

;NOW TOGGLE PA4
      LDAA #%00010000
      EORA PORTA
      STAA PORTA

;DO IT ALL OVER AGAIN
      BRA LOOP1
```

Type and run the above program. Connect the LED to PA4 (pin #5) and you should see it flicker. Connect the pin to a oscilloscope and verify that you are generating a square wave. Verify that the frequency is correct.