ECS502U Lab: Week 11

The purpose of this lab is to look into debugging tools when working with the development board. On the QMPlus module page you can find 8051 assembler instruction documentation, including a brief "cheat sheet." We suggest you create a directory for all your code for ECS502U, and then a sub-directory for this week.

This is the final marked lab sheet for ECS502U. Once you have completed this lab sheet, combine the answers into a single PDF document and submit via QMPlus.

QUESTION 1 In week 8, you implemented write and read procedures to work with the 8255 peripheral interface adapter. The instructions included the following statement:

> *Add an* **NOP** *instruction before returning to permit the 8255 to actually send the data across.*

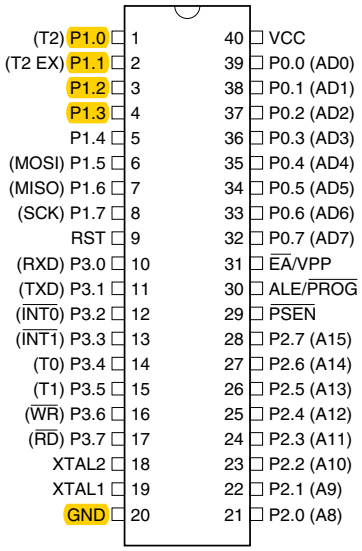
As engineers, however, you should not need to rely on such vague instructions. Therefore you will be using an oscilloscope to actually measure these delays. To do so, compile and load the following program (available as `measure.asm` from QMPlus) onto the development board, and run it.

```
1  RDpin equ 0xB3
2  WRpin equ 0xB2
3  A0pin equ 0xB5
4  A1pin equ 0xB4

6  ORG 8000h

8  ; set WRpin, RDpin high (off)
9  SETB RDpin ; read signal
10 SETB WRpin ; write signal
11 ; set address to control register
12 SETB A0pin
13 SETB A1pin
14 ; set control word
15 MOV P1,#81H
16 LCALL write

18 ; set port to C
19 CLR A0pin
20 SETB A1pin

22 repeat:
23     MOV P1,#0F0H
24     LCALL write
25     LCALL read
26     LCALL delay
27     SJMP repeat

29 write:
30     CLR WRpin
31     NOP
32     SETB WRpin
33     NOP
34     RET

36 read:
37     MOV P1, #0FFh
38     CLR RDpin
39     NOP
40     MOV A, P1
41     SETB RDpin
42     RET

44 delay:
45     MOV A, #0ffh
46 dly:
47     DJNZ A, dly
48     RET

50 END
```

The pin-out shown on the left may help to identify the appropriate pins for the following tasks. You will be using both channels of the oscilloscope. As first step, connect both channels to ground (pin 20). Work in teams of two, or with the help of a TA, to perform the following measurements:

1. Record the signal of one of pins P1.0, P1.1, P1.2, or P1.3 (and continue doing so for the next tasks). Which lines of the above code are responsible for these?

2. Use the other channel of the oscilloscope to record the signal of P3.3. Which lines of the code are responsible for these?



3. Press some buttons on the keypad, such that an effect on the pin of port P1 being measure becomes observable. What is the time between changing RDpin's value and the result becoming available on one of port P1's pins?

QUESTION 2 In C programming, `printf` is a useful debugging tool. In absence of, e.g., an LCD display, debugging on the micro-controller may be more cumbersome. Yet the serial port comes to the rescue. Start from the following program, available as `serial.asm` form QMPlus.

```
1  ORG 8000H
2
3  start:
4      JB P3.2, b2
5      MOV A, #'Q'
6      SJMP out
7  b2:
8      JB P3.3, b3
9      MOV A, #'M'
10     SJMP out
11 b3:
12     JB P3.4, b4
13     MOV A, #'U'
14     SJMP out
15 b4:
16     JB P3.5, start
17     MOV A, #'L'
18
19 out:
20     LCALL send
21     SJMP start
22
23 send:
24     ; TODO
```

You may simulate the program using virtual hardware, or go straight to the development board (connecting the LED/4-button board). Address the following tasks:

(a) Describe the expected behaviour of the program, assuming that the procedure "send" transmits the contents of the accumulator via the serial port.

(b) Which higher-level language (think of C or Java) statement is modelled by the code from lines 3 to 19?

(c) Implement the procedure "send". It should take the contents of the accumulator and transmit it via the serial port. You can safely assume that the serial port has been configured already.

(d) Optional: If you were to transmit hexadecimal numbers (e.g., the contents of a register) via the serial port for display on a remote terminal, what further procedure is required?

(e) Optional: Implement a procedure "receive" and some simple parsing to control the LEDs from the terminal. For example, on input "1" the first LED should be switched on.

Practice Projects The following mini-projects are for training purposes only and do *not* need to be submitted as part of any coursework. Instead, bring your (partial) solutions with you to the lecture on Friday, where they will be discussed.

(a) *popcount:* Count the number of 1-bits of an input in register R0, and store the result in the accumulator.

(b) *LED bar:* Assume an analogue-to-digital converter with a resolution of 3 bits providing its result on P3.0, P3.1 and P3.2. Use 8 LEDs connected to P0 to indicate the amplitude of the signal, i.e., for an ADC result of n, n bits should light up.