“Massachusetts Institute of Technology
Department of Electrical Engineering and Computer Science
6.115 Microprocessor Project Laboratory Spring 2024

DEMO Laboratory
What we would like to see in a CI-M approved lab solution and an interview check-off

Issued: February 6, 2024 Due: Never, just read it

GOALS: Learn, by reading a “DEMO” lab with solution, what we generally expect to see in a lab solution that you write.
Learn, again by reading this document, what types of questions might come up in a check-off interview with a 6.115 staff member.

PRIOR to starting this lab:
You would typically read appropriate data sheets explaining how LEDs work, recall basic TTL gates, and read about or already understand basic 8051 assembly language programming.

NOTE: Please don’t panic if you don’t understand every technical trick in this DEMO lab. The DEMO solution uses some methods and techniques that we will discuss in a few weeks in 6.115. For now, the purpose of this demo is simply to convey to you a sense of what a good lab might look like.

YOUR WORK PRODUCT: In completing a 6.115 lab, you will produce four “results” that you should have ready for us at checkoff:

1. Your working hardware and software for each lab exercise. Do not disassemble any of your circuits before you check off your entire lab!
2. Your knowledgeable brain, ready to demonstrate your completed exercises to a staff member at checkoff and to answer questions about your work.
3. Your lab notebook, filled in as indicated in this demonstration lab, for example. We will review your lab notebook with you.
4. Your “appendices,” a complete printout of your commented code on white copy paper, stapled and ready to hand in to us at your checkoff. We may also request that you submit some code online.

GRADING: Your lab work is worth a total of 10 points for each lab. Generally (there may be variations depending on lab): You can earn up to eight points for the quality of your answers to checkoff questions, demonstration of your work, and presentation of your lab notebook. You can earn two more points for your well-commented and complete code.

EXERCISE 1: Blinking light.

Your R31JP port 1 output pins cannot provide enough current to safely drive a typical LED. Let’s begin by making a circuit to help the R31JP with the very important problem of making a blinking light.

Please do the following:

- Design and construct a circuit to drive an LED using the five-volt power supply on your kit. Your circuit should take in a TTL-level command signal. When the command signal is “high,” the LED should glow. When the command signal is “low,” the LED should be off or dark.

- Test your circuit by driving it with a TTL square wave from the signal generator on your kit. Describe what the LED looks like when you drive the circuit with a 1 Hz TTL square wave from the signal generator. What does the LED look like when you use a 100 Hz TTL square wave drive?
- Be prepared to demonstrate your drive circuit in a lab checkoff.

EXERCISE 2: Let your R31JP do the driving.

Assume that your R31JP is running from a 12 Mhz clock.
Please do the following:

- Use the Port 1.0 pin on your R31JP to provide the TTL-level command signal to your LED circuit. That is, let your R31JP replace the signal generator that we used in Exercise 1.
- Write a program that turns the LED on for approximately a quarter of a second, then off for a quarter of a second, repeating forever.
- Be prepared to demonstrate your code by showing your R31JP driving the LED circuit.
- Submit your ASCII text file (ASM code) with your final code for Exercise 2 to the course web site.

On the following pages, we have attached a hypothetical solution to this lab.

NOTE: We are trying an EXPERIMENT this term to make your life easier and the lab more fun. Rather than requiring formal, typed lab writeups completed "after" the lab work, we are going to use laboratory notebooks to document your design work and also your lab work at the bench. This will hopefully minimize the time that you spend in front of a computer summarizing your work. This approach of using a lab notebook as a "scratchpad" to design your system, and then as a record documenting your construction's performance, is how you would typically work in industry or in a research laboratory. Keeping a proper, useful lab notebook is an essential skill to practice. We will give you a lab notebook that you will return to us at the end of the term.

In order for this experiment to work, you must be NEAT. Draw neat schematics, preferably with a computer, and paste them in the book. Neatly scratch over errors in your notebook with one line, and move on to a fresh page as needed. Write your name and date at the top of every page. Paste in scope photos and use boxes and underlines to indicated key conclusions and results of calculations. PRINT – do not use cursive or illegible handwriting.

The solution attached to this lab is not meant to be a "flawless" vision of ideal beauty that you could conceivably achieve. Rather, it is a solid, thoughtful, acceptable solution that would garner a good score for the lab check-off. In general, please strive to make your solution brief and clear. Think about the solution you would want to be given if you were purchasing these answers for part of a product design in your company. Your tolerance for ambiguity, missing information, or negligent error would probably be very low if you actually paid for this information. When you write a solution, you are crafting a product that is similar to reports you will make when you write papers in graduate school, file for a patent, work in industry, or when you run your own company.

When you think you have completed a lab report, ask yourself how much you'd be willing to pay for the report as documentation for your company. If the answer is "not much," you'll find that your grade on the lab report will probably improve if you revise the report for improved technical clarity and proper grammar, accuracy and neatness.

Your lab reports should include COMPLETE schematics (with all pins and part numbers and values) and carefully labeled graphs (with correct labels on the x and y axes, and with any pertinent features of the graph, e.g., peaks, zero-crossings, etc., clearly labeled). The web site WWW.EXPRESSPCB.COM offers a free, easy to use schematic drawing program. Consider using it this term. We will also accept neat, carefully labeled, hand-written schematics, but they must be legible and complete. Use concise written descriptions to explain what you did in the lab with reference to your graphs, schematics, and tables. Most of our laboratory oscilloscopes have the capability to download screen prints as postscript or jpeg files to a floppy disk or compact flash card. Take advantage of this for report documentation.

You will find a bin in the laboratory with glue sticks, scissors, and compact flash cards. Use this gear to collect and paste scope photos and schematics in your lab notebook.

At the end of each laboratory, print all of your commented code and staple it together. Print each program separately, and label it as an appendix as shown here:

Appendix A: Lab 1, Exercise 2, Blinky Code, YOUR NAME

In your lab writeup in your notebook, refer to the code appendices to indicate which code goes with which sections of your work. At checkoff, you may be asked to make modifications to your code and demonstrate the effects of requested changes. Make sure that your code is neatly and clearly commented. Slide the stapled pack of commented code into your lab notebook at the end of your lab work for that lab. A staff member will collect your code at your checkoff. You also may be asked to submit some or all of your code for a given laboratory on-line on the course web site. We will inspect code personally and with a software agent to compare it against other submissions in the class and against code from past years.

Please do not make the foolish mistake of testing us by cheating.

Your final laboratory grade would also depend on your overall knowledge as demonstrated by your oral answers to questions like those on the "check-off criteria" sheet attached at the end of the solution. A TA might ask you these questions in lab.

Here are a couple of "online" references available for helping you to improve your writing if you need them:

Mayfield Handbook of Technical and Scientific Writing,	https://web.mit.edu/21.guide/www/home.htm
LabWrite, North Carolina State University	http://www.ncsu.edu/labwrite/

This patent, below, is an example of technical writing for communicating an engineering idea in a commercially relevant form, a US Patent. Lab notebooks and the dated data and ideas that they contain often serve as the basis for lucrative patent filings. Use your experience in 6.115 to practice making complete lab notebook observations that would support or serve as useful written communications.

Professor Leeb's US Patent 6,198,230 available from: http://web.mit.edu/6.115/www/ideas.shtml

Your lab report would be the text on this side of the page.

INSTRUCTOR COMMENTS:

Note the nice clear title with name and date.

This page would be a permanent page in your lab notebook.

A brief introduction establishes what's in the report. No need for long-winded literary introductions about how wonderful microcontrollers are, or what philosophical purpose the lab might have. Just say what the goals are and what sections are in your report to demonstrate your mastery of the goals.

USE FULL ENGLISH SENTENCES. Use a clear paragraph format.

Draw NEAT schematics for all of your circuits, with complete connections and component values shown. I use the computer drawing package from expresspcb. You can hand draw your circuits, but you must use a template or ruler to make neat, clear symbols.

Briefly and clearly explain how the circuit works.

Show any relevant design calculations that explain how you sized components like resistors in the circuit.

6.115 DEMOLAB WRITEUP STEVEN LEEB

GOALS: DATE

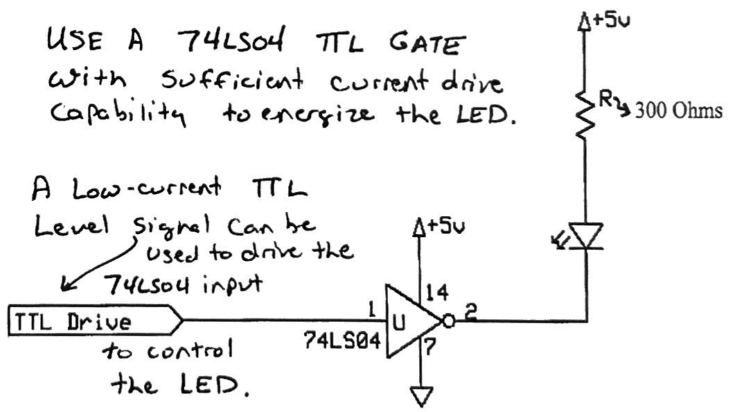
The goal of this laboratory was to provide an opportunity to explore basic techniques that I will learn more about in 6.115.

Another goal of this lab was for me to practice writing a clear set of lab notebook entries, explaining what I did and observed in the lab in clear English grammar.

TWO EXERCISES: (1) Construct an LED driver
(2) Write "blinky" code for the R31JP

EXERCISE 1:

- My driver circuit design is shown below:



NOTES:

- Use an LED/Resistor combination to limit LED current.
- Connect the LED/Resistor combination to the +5 rail and the gate output to create the required logical sense requested in the lab.

INSTRUCTOR COMMENTS, Continued:

Here is the next "permanent" page in your lab notebook.

Show any relevant design calculations that explain how you sized components like resistors in the circuit.

DATE STEVEN LEEB

- Input → High leads to LED "ON"
- Input → Low leads to a high output on the 74LS04 grate and LED "OFF"

Sizing the resistor value R:

- LED Data sheet indicates max LED current of 10 mA at a foward voltage of 1.5 V.
- I want my LED bright so I'll size my resistor for 10 mA of forword current.
- At the "Low" output voltage, the 74LS04 DATASHEET indicates that the 74LS04 OUTPUT will drop to 0.5V (not zero!).

So:

$$R = (5\text{ Volts} - \underbrace{1.5\text{ Volts}}_{\text{LED DROP}} - \underbrace{0.5\text{V}}_{\text{74LS04 "Low"}}) / 10\text{mA}$$

or

$$\boxed{R = 300\text{ Ohms}}$$

I connected my drive circuit input to a TTL Signal Generator as requested in the Lab.

- With a 1 Hz period, the LED glowed brightly for about 0.5 seconds, stayed dark for about 0.5 seconds, and then repeated this behavior.
- With a 100 Hz period, the "blinking" was too fast for my eye to see. The LED appeared to be on at about half of the full brightness all of the time.

INSTRUCTOR COMMENTS, Continued:

The final, permanent page in your lab notebook:

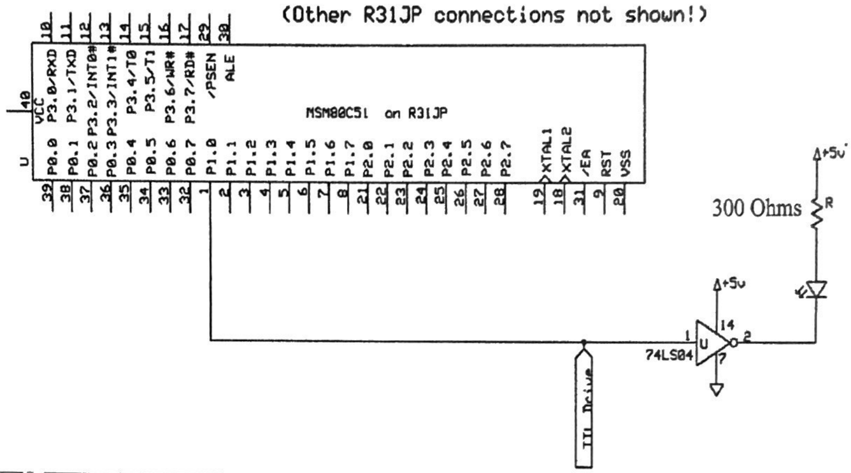
Another neat, clear schematic showing **everything** that **you** built or hooked up for Exercise 2, and to what it is connected. Note that it is not necessary to draw every single R31JP wire and connection. Just show the relevant connection to your LED driver circuit that <u>you</u> added, and make a note of what's missing. Try to develop a clear sense of what's important to include and what's confusing. In this case, for example, we don't need to see the RAM and ROM connections on the R31JP board. You did not build these for Exercise 2, so a simple note that the "other" R31JP connections (that were not built by you) are not shown is sufficient.

Explain and <u>highlight</u> essential calculations that make clear how you designed your code.

DATE STEVEN LEEB

EXERCISE 2:

- In this exercise, I connected my LED driver circuit to PORT 1, PIN 0, as requested in the lab. This connection is shown in the figure below:



- My R31JP Code to make the light blink on and off with a 0.5 second period is listed in Appendix A. Appendix A includes comments, and has been submitted to the course web site as requested.

- The DELAY Subroutine in my code decrements an inner counter R1 from 255 to 0. This happens 255 times, once for each count of an <u>outer</u> counter R0 counting from 255 to 0. The DJNZ command in my count loop uses <u>2 cycles</u> or 2 microseconds on an R31JP with a 12 MHz clock. Therefore, <u>two</u> calls to Delay in the main loop results in a time delay of approximately:

<u>Two calls</u>

$$\rightarrow 2 \cdot \underbrace{255 \cdot 255}_{\text{Loops}} \cdot \underbrace{2e\text{-}6}_{\text{DJNZ time}} = 0.26 \text{ seconds as required.}$$

INSTRUCTOR COMMENTS,
Continued:

CLEARLY comment your code listed in Appendix A!
Provide code listings with an easy to read font and indentation that indicates structure in the program.

The code comments should tell a running story about your program. That is, a comment should not simply say what the line does functionally, e.g. "Calls subroutine Delay." Rather, it should explain the role of the line in the program, e.g., "waits for a quarter second by calling Delay."

Explain briefly how you would use/call (e.g., use MON/RUN) your code.

YOU WOULD TURN IN THE APPENDIX SHEET BELOW TO THE STAFF MEMBER AT YOUR LAB CHECKOFF. DO NOT GLUE OR AFFIX IT TO THE LAB NOTEBOOK!

APPENDIX A: Lab 1, Exercise 2, BLINKY CODE — Steven B. Leeb

NOTE: My program starts at ORG 0000h, so I run it using the RUN/MON switch after loading it through WinSCP.

CODE:

```
main:                  ;Here's the main blinking program
   mov  P1, #00h       ;First, make all port 1 pins low (LED off)
   lcall delay         ;Wait for a quarter of a second using delay
   lcall delay
   mov P1, #01h        ;Turn a port 1 pin on to make the LED glow
   lcall delay         ;Wait for a quarter of a second using delay
   lcall delay
sjmp main              ;Do it over again for blinking effect

.org 500h              ;locate the subroutine DELAY away from main
delay:
   mov R0, #0ffh             ;load R0 and R1 for 255 counts
   mov R1, #0ffh

   loop1:
     djnz R1, loop1      ;decrement 255 counts in R1
   mov R1, #0ffh         ;reload R1 in case
                         ;       we need to loop again.
   djnz R0, loop1        ;count R1 down again until R0
                         ;     counts down
ret
```

Demo Laboratory: Possible Check-off Criteria, i.e., questions you might be asked at the lab check-off interview.

During check-off, we will be verifying that you have completed the lab. We will also attempt to determine if you have learned anything, and if you are able to think critically about the material at hand. The live check-off with a staff member will be a significant means for determining grades. Note that the verbal check-off counts for 80% of the lab grade!

If you were being checked off on this demonstration lab, you might be expected to answer or do the following for a 6.115 staff member:

Exercise 1:

- How would you change the circuit to double the current through the LED?
- How would you change the circuit to reverse the logical sense of the driver, i.e., make the LED glow when the input signal was a logical "low" signal?
- Demonstrate at a bench with an oscilloscope and a signal generator the behavior of your LED driver.
- Could you replace your 74LS04 with an LM358 op-amp? If so, draw a quick sketch of your op-amp circuit.

Exercise 2:

- (If we were farther along in the course:) How would you use an interrupt to create the timing loop?
- Conduct an exact timing analysis of your code – exactly how long is the blinking period?
- Recompute the blinking rate for your code assuming that the R31JP is running from an 11.059Mhz crystal.
- What would you change in your program, if anything, so that, after loading it into the R31JP, we could execute it using a MINMON G8000 command ?
- Modify your code to make the LED flash twice as quickly. <u>Make your changes, recompile your code, load it and run it, all during the checkoff.</u>
- Suppose the 8051 was running your code from internal memory and that the Port 0 pins were free and available. Would you need any additional components to run your LED driver circuit from Pin 0 on Port 0?