

EECE 690/890

Digital Radio Hardware Design

Team 3

Assignment 1

Due Thurs 10/1/98

Introduction

This is the first of a series of assignments designed to guide you through the tasks needed to complete the software design. The first part of the assignment is common for all team members to get each of you familiar with the PIC processor and software development environment. Each team member should do the tasks under “Common Tasks” below.

Following these, specific tasks are detailed for each team member so that each person has a well-defined job and deliverable (material to be turned in by the due date). However, the tasks are also interdependent, so you need to work together.

Attached to this assignment is an additional page giving a (preliminary) preview of future tasks that will need to be completed before the PDR. You are encouraged to work ahead if you can. This will ease your workload when things like tests stack up in other classes in the coming weeks.

Common Tasks for all Team Members

Each team member should perform the following tasks on his/her own. You may help each other out, but each person should go through the full exercise and make their own printouts:

You should perform the following tasks:

- ♦ Download the data sheet on the PIC17C44 microcontroller and scan through it to understand what information it contains (it is a very long document - you are not expected to read every page!).
- ♦ Download the MPLAB development environment software from the Microchip web site, (or if you prefer, identify a computer in Rathbone Hall where you can do your software development).
- ♦ Start up the MPLAB software and look over the help pages. Then:
 - Select each drop-down menu to see what is there.
 - Create a new project (Project>NewProject), being sure to specify “Simulator Mode”.
 - Create a new source file (File>NewSource), and enter the code from class. Then add the file to your project (Project>EditProject). Make sure you understand where your

- project file and source file reside! - E.g. save each to a known location such as a new directory you have created to store your work in.
- Compile the code (Project>MakeProject) and print a listing of the object code generated (Window>AbsoluteListing followed by File>Print). Be sure to configure Print Setup to include margins and use a reasonable font (e.g. 10 point courier).
 - Check to be sure that the simulator is enabled (Options>DevelopmentMode), and then set breakpoints (right mouse click) in your code at 'start_main', and at the beginning of the 'wait100ms' routine. (Recall that an easy way to do this is to position the cursor on the line, and right-click the mouse to bring up a menu).
 - Create windows to monitor the File Registers and the Special Function Registers (See Window menu).
 - Single-step through the code (or run to a breakpoint) and watch the register windows for the expected changes.
 - Print out the debug windows (select window, followed by File>Print)
 - Save your work.

Firmware Engineer 1 Tasks

- ♦ Write a function that can be called to initialize all PIC I/O port registers to act as outputs, and an additional function that can be used to set a selected bit of a selected port to be an input. These functions will be useful to your teammates (Firmware Engineer 2 and Diagnostic Software Engineer), so document them well.
 - Study the PIC17C44 data sheet sections pertaining to the I/O ports.
 - Plan your code with a flowchart, or with pseudocode as appropriate.
 - Write (and comment!) your functions and a small test program to call them. (You may want to reuse some of the code from the class example and you should use the standard port and bit definitions in p17c44.inc whenever possible).
 - Compile your program, and test your functions by stepping through the program.

Your deliverables are listed below:

- ♦ The printouts specified in the common task above. Each team member should provide his or her own printouts.
- ♦ The pseudocode or flowchart of your program and a brief writeup on how it works (and whether your tests on it were successful).
- ♦ An email message sent to your company giving the program code you wrote, and the description of how it works. (This is primarily intended to help your team members, but other employees may be interested).

Firmware Engineer 2 Tasks

- ♦ Write a function that can be called to make the phone “ring”. Assume that a buzzer is attached to a suitable I/O pin on the device and that the pin is already programmed to be an output (this is being worked on by another member of your team). You make the phone ring by alternately setting this pin high and then low, at a suitable rate. When the pin is high, the buzzer sounds, and when low, it is silent. Your program should make it go buzz, buzz, buzz, etc. until someone picks up the receiver (check the “hook” bit).
 - Open the “p17c44.inc” include file (File>OpenSource) located in the MPLAB directory, print it out (File>Print), and study it. Provide a copy to your teammates for their use.
 - Plan your code with a flowchart, or with pseudocode. (You may want to reuse some of the code from the class example and you should use the standard port and bit definitions in p17c44.inc whenever possible).
 - Write (and comment!) your function and a small test program to call it. Compile it, and test it by stepping through it and monitoring its execution and the values of the ports you are reading and writing.

Your deliverables are listed below:

- ♦ The printouts specified in the common task above. Each team member should provide his or her own printouts.
- ♦ The pseudocode or flowchart of your program and a brief writeup on how it works (and whether your tests on it were successful).
- ♦ An email message sent to your company giving the program code you wrote, and the description of how it works. (This is primarily intended to help your team members, but other employees may be interested).

Test / Diagnostic Engineer Tasks

- ♦ Get started on defining the I/O pins to be used and the diagnostic/test functions to be written as follows:
 - Get a copy of the LCD manual from your company’s manager and review the signals needed to interface to it.
 - Study the top-level block diagram of the phone and make a list of all the I/O signals needed. For things like the synthesizer, and the PLD you may need help from other teams and/or your company manager to identify how many pins will be needed.
 - Work with other members of your team (send lots of email, or meet/call each other!) to do a preliminary specification of what ports and pins will be used for each function

and then create an include file that contains identifiers for all of the signals expected to be used.

Your deliverables are listed below:

- ♦ The printouts specified in the common task above. Each team member should provide his or her own printouts.
- ♦ The list of I/O signal that will be needed to control the phone, together with a description of the signal (e.g. “The TX line is an active high signal that turns on the transmitter”)
- ♦ The (typed-in) include file (with comments!) you created that gives definitions of the ports and bits that will be needed in later coding.
- ♦ An email message sent to your company giving the include file you wrote, together with the description of the I/O signals. (This is primarily intended to help your team members, but other employees may be interested).

Team 3 Future Assignments

(Preliminary)

The following gives an overview of the tasks remaining after task 1. These will be broken down into assignments like task 1, with recommended subtasks and deliverables. We will also have periodic “mini-design reviews” in which your team will meet with the instructors. In these informal reviews, you will need to explain your design decisions, and we will try to find “holes” in the design that need to be addressed.

Firmware Engineer 1

- ♦ Write top-level executive. (Some routines will be stubbed out, but interfaces and functionality of these routines should be defined.)
- ♦ Study/document the interface to the PLD and write code for sending/receiving packet type and security ID.
- ♦ Write call init code and develop a test plan for it.

- ♦ Test the code.
- ♦ Integrate with Firmware Engineer 2's code.

Firmware Engineer 2

- ♦ Develop initialization code for the microcontroller.
- ♦ Study/document interface to synthesizer (getting required programming data from Team 1).
- ♦ Port code for synthesizer programming.
- ♦ Write Call reception code and develop test plan for it.
- ♦ Test the code.
- ♦ Integrate with Firmware Engineer 1's code.

Test/Diagnostic Engineer 1

- ♦ Study/document the LCD display programming needed.
- ♦ Port code for displaying messages on the LCD (work with other team members to define the best interface for this).
- ♦ Build a test board to connect a programmed 17C44 to the LCD.
- ♦ Test the LCD display routines.
- ♦ Work with RF team to test synthesizer interface.
- ♦ Write diagnostic code for placing phone in various modes that will be useful when phone is first activated.