

EECE 690/890

Digital Radio Hardware Design

Team 3

Assignment 3

Due: Tues 11/3/98

Introduction

This is the third in a series of assignments designed to guide you through the tasks needed to complete the software design. In this assignment the two Firmware Engineers will write most of the main code used in call processing (see flowcharts given out in class), while the Test/Diagnostic Engineer will develop code that will be used during initial test of the phone hardware.

In the next assignment following this one, the Firmware Engineers will test their code by programming a PIC and using the Hardware Test Bench created by the Test/Diagnostic Engineer in assignment 1. The Test/Diagnostic Engineer will test his/her code on the hardware as well. As always, you are encouraged to work ahead if you can. Talk to the Test/Diagnostic Engineer if you wish to try testing some of your code in this assignment.

Firmware Engineer 1 Tasks

Your main task for this assignment is to write code specified in the Call Initialization flowchart discussed in class (or a suitable derivative of this flowchart). A list of suggested subtasks is given below to help you organize your work. **Note that many functions needed will be the same or very similar to those that your teammate needs. Therefore, one of your first tasks is to coordinate who will write what, or if you each want to write the functions and use the best implementation in the final code.**

- ♦ Review the Call Initialization flowchart and modify it as needed to improve it.
- ♦ Talk to the PLD team and review their documentation to determine how to load packets into the transmit queue and begin transmitting them. (Coordinate with the other firmware engineer on this to divide the work and/or agree on a common routine.)
- ♦ Coordinate with your teammate on who will write what code.

- ♦ Develop a flowchart or pseudo-code description of the block in the Call Init flowchart that says “Begin Transmitting Packets”. (Coordinate with the other firmware engineer on this to divide the work and/or agree on a common routine. They have to create a function to “Send Reply ID” which is the same thing.)
- ♦ Code the Transmit Packet functions.
- ♦ Repeat the above three steps to develop code for checking for a valid ID (if not already done in assignment 1).
- ♦ Determine what additional supporting routines you will need. Some of the routines and possible sources for the code are:
 - Function to scan for an available channel. This could be adapted from the code written in Assignment 1. If possible, reuse some of the same functions, or modify the code to allow reuse. It is good programming practice to write a function that is useful in two applications and just call it from those applications. Do not just copy the code and end up with two versions of the same thing. That wastes space and makes code maintenance harder.
 - Function to turn transmitter on. Make this a function, even though the amount of code involved may be small.
 - Function to begin transmitting packets. This will probably be somewhat involved and you will need to write several sub-functions. Do a flowchart or pseudo-code first.
 - Function to check for off-hook. (Pass back a 0 or 1 in W register).
 - Function to turn transmitter off.
 - Function to make a busy signal. (Does this need a modification of the hardware design, or a new output from the PIC? Talk to team 4.)
- ♦ Write a pseudo-code description of the Call Init code using the functions above. You should be more detailed than the flowchart and you should call out the names of the functions identified above that perform the different actions needed. Keep your main Call Init routine simple! Delegate most actions to called functions. Pseudo code for lower level functions is not required at this stage.
- ♦ Write the Call Init code.
- ♦ **DOCUMENT YOUR CODE** with appropriate comments before each function, and for each major section!
- ♦ Check with your teammates to be sure that you all agree on interfaces and are ready to integrate the code together!!

Your deliverables are listed below:

- ♦ A summary of how you divided up your work in writing the functions.
- ♦ Your flowchart and/or pseudo-code description of the Call Init code.
- ♦ Your (documented) code - including any header files you used.

Firmware Engineer 2 Tasks

Your main task for this assignment is to write code specified in the Call Reception flowchart discussed in class (or a suitable derivative of this flowchart). A list of suggested subtasks is given below to help you organize your work. *Note that many functions needed will be the same or very similar to those that your teammate needs. Therefore, one of your first tasks is to coordinate who will write what, or if you each want to write the functions and use the best implementation in the final code.*

- ♦ Get a copy of the Top-Level Executive code written by Firmware Engineer 1 in the first assignment and review the code to understand what is there.
- ♦ Review the Call Reception flowchart and modify it as needed to improve it.
- ♦ Coordinate with your teammate on who will write what code.
- ♦ Talk to the PLD team and review their documentation to determine how to load packets into the transmit queue and begin transmitting them. This will be needed to do the “Send Reply ID” portion of the flowchart.
- ♦ Write psuedo code for the functions needed. (Coordinate with the other firmware engineer on this to divide the work and/or agree on a common routine. They have to create a function to “Begin Transmitting Packets” which is the same thing.)
- ♦ Code the functions.
- ♦ Repeat the above three steps to develop code for checking for a voice packet. This is similar to the code done by Firmware Engineer 1 in checking for a valid ID, but only requires checking the packet type.
- ♦ Determine what additional supporting routines you will need. Some of the routines and possible sources for the code are:
 - Function to check for off-hook. (Pass back a 0 or 1 in W register). Make this a function, even though the amount of code involved may be small. It will make your code more readable and easier to maintain.
 - Function to check for “Still Receiving”. You should probably use the “Valid ID” check code here.
 - Function to activate ringer. This can be adapted from the first assignment, but you may need to add code for checking the hook and checking to see if we are still receiving.
 - Function to turn transmitter on.
 - Function to switch PLD to Voice Mode.
 - Function to turn transmitter off.
- ♦ Write a pseudo-code description of the Call Reception code using the functions above. You should be more detailed than the flowchart and you should call out the names of the

functions identified above that perform the different actions needed. Keep your main Call Rcv routine simple! Delegate most actions to called functions.

- ♦ Write the Call Rcv code.
- ♦ ***DOCUMENT YOUR CODE*** with appropriate comments before each function, and for each major section!
- ♦ Check with your teammates to be sure that you all agree on interfaces and are ready to integrate the code together!!

Your deliverables are listed below:

- ♦ A summary of how you divided up your work in writing the functions.
- ♦ Your pseudo-code description of the Call Rcv code.
- ♦ Your (documented) code - including any header files you used.

Test / Diagnostic Engineer Tasks

Your main task for this assignment is to write diagnostic code for placing the phone in various modes that will be useful when the phone is first built. This will initially be a separate program from that written by your teammates, but will later be integrated into the full code so that we can use it at open house to demonstrate various aspects of digital radio functions.

The diagnostic code will use interface with the user through two pushbuttons and the LCD. The pushbuttons will be used to select functions while the LCD gives confirmation on what the software is doing. When a function is selected, it will cause various actions to be taken such as toggling the Sleep/Wake line to the TX/RX, initializing the PLD, programming the synthesizer, etc.

Note that many functions needed will be the same or very similar to those that your teammates need. Therefore, one of your first tasks is to coordinate who will write what, or if you each want to write the functions and use the best implementation in the final code.

A list of suggested subtasks is given below to help you organize your work.

- ♦ Define two input pins on the uC to connect to pushbutton switches. The first pushbutton will cause the code to cycle through several possible functions, and the second will cause the current function (displayed on the LCD screen) to be activated.

- ♦ Make a list of the functions needed and what will be displayed on the LCD screen. You should talk to the other teams to see what they want to do their initial testing. A reasonable set of functions might include:
 - Initialize TX/RX - Code written in previous assignment by Firmware Engineer 2.
 - Channel UP - Function to increment the synthesizer(s) to the next channel (and wrap around from 9 to 0).
 - Synth Test - Function to program the synth LD/Fo pin to output different signals.
 - Toggle TX - Function to toggle TX line on transceiver hardware.
 - Toggle Sleep - Function to toggle sleep line on transceiver hardware.
 - Ring - Function to activate ringer
 - Busy - Function to create busy signal sound
 - RX ID - Function to place PLD in mode of receiving ID (display results to LCD?)
 - TX ID - Function to place PLD in mode to transmit ID
 - Voice Mode - Function to place PLD into mode of transmitting/receiving voice
- ♦ Decide what will be displayed on the LCD. You should show the function names as you cycle through the choice, but will you also show the current state for toggle functions, or the ID received for the RX ID function?
- ♦ Create a flowchart and/or pseudo-code of your program.
- ♦ Work with teammates on writing the functions.
- ♦ Code the functions.
- ♦ ***DOCUMENT YOUR CODE*** with appropriate comments before each function, and for each major section!
- ♦ Check with your teammates to be sure that you all agree on interfaces and are ready to integrate the code together!!

Your deliverables are listed below:

- ♦ A summary of how you divided up your work in writing the functions.
- ♦ Your flowchart/pseudo-code description of the diagnostic code.
- ♦ Your (documented) code - including any header files you used.

Team 3 Future Assignments

The following gives an overview of the tasks remaining after task 3.

Firmware Engineer 1

- ♦ Develop a test plan for your code.
- ♦ Add code to write messages to LCD that show what mode the phone is in.
- ♦ Test the code first on the simulator, and then on the Hardware Test Bench.
- ♦ Integrate with Firmware Engineer 2's code.

Firmware Engineer 2

- ♦ Develop a test plan for your code.
- ♦ Add code to write messages to LCD that show what mode the phone is in.
- ♦ Test the code first on the simulator, and then on the Hardware Test Bench.
- ♦ Integrate with Firmware Engineer 2's code.

Test/Diagnostic Engineer 1

- ♦ Develop a test plan for your code.
- ♦ Work with other members of your team and with the PLD team to test your code and theirs on the hardware test bench.
- ♦ Integrate code with other team members.
- ♦ Work with RF team to test the frequency synthesizer.