

**Names**

**Fall 2012**

**ECE 362 Final Project (boat): USS Neversail**

Abstract: The objective of this project was to incorporate the assembly language elements learned this semester into a comprehensive project which demonstrates the various elements of a programmable boat on the Motorola HC12 microcontroller. This project uses interrupts to drive a variety of subroutines as each desired task as requested by the user. A password is required to start, which can then be reset by the user with a variable length of up to 7 places. In addition, two motors are used. One for propulsion and one for rudder control with both clockwise and counter clockwise rotation.

**IUPUI**

**Indianapolis, IN**

**Table of Contents**

1. **List of Figures and Tables**
2. **Introduction**
   1. **Goals**
   2. **Assumptions**
3. **Design**
   1. **Required Elements**
   2. **Additional Elements**
   3. **Usage of Peripherals**
      1. Stepper-Motor
      2. DC-Motor
      3. Keypad
      4. Potentiometer
      5. LCD
      6. Menu
      7. Display the fuel level
      8. Display the speed
      9. View time
      10. Push Button
      11. Switches
      12. Speaker (Port T)
      13. LEDs
      14. IRQ
      15. RTI
   4. **Software implementation**
      1. Setup/Initial values
      2. Switches
      3. Subroutines
      4. Tables
      5. Variables
4. **Changes made**
   1. Changed clock divider
   2. Keypad debounce value
5. **Additions to the project**
   1. Alarm system
   2. Variable password
   3. Light functionality
6. **Division of work**
7. **Functional and Non-functional portions of the Design**
   1. **Functional**
   2. **Non-functional**
8. **Conclusion**

**List of Figures and Tables**

**Attachment 1 (main.asm)**

**Attachment 2 (lcddisp.c)**

**Attachment 3(Potentiometer.c)**

**Attachment 4(SendsChr.c)**

**Attachment 5 (Keypress.asm)**

**Introduction**

The “USS Neversail” (our boat) program incorporates many of the elements learned during this semester in ECE 362. The boat has a password enabled system, programmable clock function, speed control, rudder control, alarm system and horn. These elements are enabled by Boolean switches and the use of interrupts. The design requirements called for the incorporation of some elements which were not fully implemented in the time allotted. These will be reviewed in the Discussion section of this report. In addition problem areas and the methods which were employed will be discussed as well. The system was programmed using assembly language through the Freescale Codewarrior IDE and the use of the Motorola HCS12 microcontroller.

**Goals**

The objective of this project was to incorporate interrupts, variables, logic switches, pulse width modulation, subroutines, bit wise functions and the various ports available to emulate the performance characteristics of a boat. The user would be able to control desired functions through the keypad, a push button, IRQ and dipswitches. Interrupts were used to control timing elements required in the performance of the various functions of the microcontroller.

For example, the main interrupt (rti) was used to count. This value is used in the pulse width modulation of the motor which regulates the motor speed. Motor speed is increased by sending a longer duty on cycle to the motor. Our design also incorporated logic switches in this function to control turning on and off the motor through the menu and keypad.

**Assumptions**

Several assumptions were made to setup the conditions for the program. Because the timing of the rti was of critical import to the entire program, a clock division was required. A divider of 2^20 was used to get a 1 ms value for the clock. For the keypress portion of the code a debounce value (called length) was used of approximately 4 ms. Some delays were used through the rti routine (as required). One delay affected the rudder which was controlled via the potentiometer. The delay was named “rudder\_delay” and had a value of 25000. This was determined by trial and error to account for the sensitivity of the particular microcontroller board that we were using. To give the desired blinking effect for the LEDs, an arbitrary value of 1000 was used and saved to the **RTI** as “light\_delay”. Because this worked well, it was never modified but could easily be changed for the desired user affect. To simulate fuel usage, a timing device was programmed into the rti routine which was called “fuel\_delay”. Because this was a simulated rate of fuel consumption, the program was modified to give a quick but measured rate of decrease. The fuel\_delay was set at a value of 500 and this was combined in an algorithm with “fuel per” to get a marked decrease in fuel for the presentation. To account for the 1 ms clock speed used for the eclock, a conversion value called “clock\_delay” was used to increase the time period to 1 second which was used in the clock display function in the LCD clock menu. Prop\_Delay was intended to be used to control the variable motor speed for the propulsion motor. There were three variable pairs that were coded to control the Flank, Full and Slow speed of the engine. For the Flank speed the Prop\_Delay value was 10000 and a prop\_delay\_off value of 0 was used. For Full speed Prop\_Delay and prop\_delay\_off were each set to 5000. For slow speed Prop\_Delay was set to 2500 and prop\_delay\_off was set to 7500. For the horn an arbitrary value of 750 was stored to the variable Emer\_Horn\_Delay in the **RTI** routine. This was chosen for its appropriate alarm and horn sound characteristics.

**Design**

The overall design was in conformance with the project requirements. These requirements are specified below. Additional design elements were also employed such as a variable length user defined password and a low fuel alarm protocol.

**Required Elements**

The provided requirements called for several design elements. These included the following:

* The user should first start the engine, by pushing start and entering the password.
* The vehicle should start in a “stop” mode.
* The vehicle should be at “stop” to shut down the engine.
* After starting, the user must select the boat’s speed.
* The boat will have 4 speeds: Stop, High, Medium, and Low.
* DC motor with 3 distinct speeds: High, Medium, and Low.
* The DC motor will gradually increase, or decrease, to the given speed.
* The direction is controlled by the potentiometer.
* The center position of the POT will be forward and any change will turn the stepper motor in that direction at a speed based on the distance from center.
* The speed of the boat (High, Medium, Low, and Stop) should be displayed.
* After starting the boat, the user can use the horn, menu, and spot lights. (Spot lights can be used in any situation).
* There must be two separate spot lights.
* The boat should display the fuel level.
* The boat shuts down if the fuel tank is empty.
* The boat must refill with gas before it can start again.
* The fuel latch must be open to refill.
* If the fuel latch is open, and the boat is moving the boat shuts down.
* The system has a timer that displays the time (hh:mm:ss Day). The user should be able to change the time and the date.
* The main light will be controlled by the clock, turning on for the time period 6pm to 6am.
* No delay loops are allowed, you must utilize the Real Time Interrupt (**RTI**).

**Additional Elements**

In addition, the “USS Neversail” team implemented a couple more design elements listed below:

* An audible alarm and an alarm message would occur upon running out of fuel.
* Also, a programmable military (24 hour) clock was also available from the menu.
* The user password could be of variable length (up to 7 chars).
* The control of the rudder would allow for stable control of rudder at the center position for the potentiometer.
* Boolean switches would be used to allow for the programmer to incorporate changes to the program for adding in different functionality as the worked progressed.
* The debounce function of the keypad would not be such that the user needed to modify their pressing of the buttons to get the desired result.
* A menu of 5 “sea shanties” would be playable from the menu.
* The IRQ button could be used to test the alarm for the engine shutdown condition.

**Usage of Peripherals**

1. Stepper-Motor:
   1. As discussed, the rudder direction, rate of turn and direction of rotation was based on the potentiometer’s distance from the center value of full range of the potentiometer. This was controlled via a subroutine called “Sub\_Rotor”.
   2. Additionally, a calibration (implemented via code) of the sensitivity of the potentiometer was needed because the board used for the demonstration was “touchy”.
2. DC-Motor:
   1. Main engine was coded to have Pulse Width Modulation for 4 speeds (Stop, Slow, Full and Flank). However, our implementation was faulty and could not implement the speed change function successfully in time for presentation. See functional section for additional information.
3. Keypad:
   1. Select different options and provide functionality for:
4. Enter Password (worked)
5. Setting the time (worked: for am/pm and military time)
6. Select song (not implemented successfully)
7. Start Button / Shutdown Button (worked)
8. Potentiometer:
   1. Direction of the motor (worked perfectly)
   2. Sensitivity of potentiometer tailored to the microcontroller in use (worked).
   3. Rate of turn adjustable (worked).
9. LCD
   1. Displayed required menu items. (worked)
10. Menu
    1. Initial setup menu
    2. System password menu
    3. User password change menu
    4. Setup menu
    5. Clock set menu (12 hour)
    6. Clock set menu (24 hour)
    7. Some additional menus were not incorporated in time
11. Display the fuel level
    1. Fuel level was displayed as a percentage from full.
    2. Stop displaying fuel usage upon reaching empty (worked).
12. Display the speed
    1. Speed indication was associated with the propulsion subroutine (did not work).

1. View time
   1. Time in 12 hour clock form (worked).
   2. Time in 24 hour (military) clock form (worked).
2. Push Button:
   1. Horn (worked)
3. Switches:
   1. Fueling switch and fuel latch (had trouble).
   2. Switch used to simulate key for the Boat
4. Speaker (Port T):
   1. Horn and music (5 different songs >5seconds each)
5. LEDs
   1. Lights and spotlights on the boat.(3 bit and 4 bit position)
   2. Lights blinked on emergency condition at zero and 7 bit position. (worked)
6. IRQ:
   1. Engine Shut Down. (worked)
   2. Added Horn, message on LCD display and blinking lights.(worked simultaneously)
7. RTI
   1. Implemented with motors, menus, keypad, IRQ, horn and dipswitches.

**Software implementation**

To implement the sundry functions of the microcontroller, a number of variables, initial values, tables, subroutines and interrupts were used. Among the variables a special type of variable was utilized, the Boolean operator or switch. For ease in implementation and in troubleshooting, a variety of subroutines were used to track the different functions in the code.

1. Setup/Initial values
   1. The initial values for the various ports were provided by the instructor. These values established the ports in the appropriate configuration for later functionality.
   2. Changes to the .prm file were also coded in to incorporate the 2 interrupts in use (Vector 6 and Vector 7) for the IRQ and the RTI.

1. Switches (used as Boolean terms)
   1. A number of switches were used to coordination the operation of the RTI, establish logic conditional operations and to use in the testing of the subroutines. The switches are described in the commented code in the main program.
   2. A list of the switches used is provided in Table 1.
2. Subroutines
   1. Subroutines were used to break the extensive code into smaller manageable pieces for ease of troubleshooting and coalescing into a cohesive project.
   2. A copy of the subroutine codes is provided as Attachments 2 through X.
3. Tables

The description of the tables used in the project are provided below:

* 1. “Rows”:provides the values for reading the keypad ( $70,$B0,$D0,$E0,$FF)
  2. “Values”:provides the entire set of values need to read the keypad with FF as the terminator($EB,$77,$7B,$7D,$B7,$BB,$BD,$D7, $DB,$DD,$E7,$ED,$7E,$BE,$DE,$EE,$FF)
  3. “Step\_values” were used in conjuction with the stepper motor.($00,$0A,$12,$14,$0C,$FF)

1. Variables
   1. The variables used can be seen in Attachment 1. Most of the variables are already mentioned as part of the switch listing in Table 1. The remaining variables are described and commented in Attachment 1.

**Changes made**

A number of changes were made throughout the design, testing and implementation portions of this project. Only the most significant changes will be discussed since most changes were minor and there is little to be learned from discussing those.

1. Changed clock divider. Initially started with a clock divider value that yielded a 128 ms frequency. By dividing the original clock setting by 2^7, the frequency was reduced to a 1 ms value. This was crucial in coordinating the clock display and had no negative impact on the performance of the main or subroutines.
2. Keypad debounce value. A 4 ms debounce value was used to prevent inadvertent repeated key presses from being registered. This was changed to accommodate the change in clock speed from the earlier (128 ms) value.
3. The remainder of the project was consistent with the goals set forth with the exception of the portions which were non-functional.

**Additions to the project**

The additions to the project were consistent with real life performance of boats.

1. Alarm system: Added an audible alarm, a visual warning and an engine shutdown when the fuel indicator became zero were all incorporated to be more realistic.
2. Variable password: Added the ability to have a variable length password (up to 7 values)
3. Light functionality: Added blinking maneuvering lights when in motion.

**Division of work**

The work was divided up according to each individual’s schedule. Because of offset class periods and the Thanksgiving break, both team members contributed individually during the week and in a combined effort on Saturdays. Since the member skills were similar, collaboration was necessary throughout the project to overcome hurdles. Also, when one person was not active in coding, other tasks were undertaken in parallel when possible. Additionally, forceful backup was provided by both persons and the inputs of both members were incorporated as much as possible into the finished project.

**Functional and Non-functional portions of the Design**

**Functional**

1. The clock displays for both 12 hour and 24 hour was functional and programmable.
2. The password (both system and user) setup worked correctly.
3. The horn worked well via the push button through the RTI.
4. The keypad input worked correctly without issues.
5. The IRQ interrupt correctly caused an emergency shutdown.
6. The rudder controls were perfected. This had been a problem in the last few hours before presentation but the fix was found in time. Previously, the rudder controls were difficult to hold at a center straight position.
7. The dipswitches (position 4) turned the engine on as expected to simulate a key in the ignition.
8. The lights (maneuvering and rudder light) behaved as intended.

**Non-functional**

1. The 4 speed propulsion system did not work. The reason is not clear. As our project approached the presentation time, it was decided to address any elements of the project that could be quickly implemented. The engine ran in one speed and off. The code was written to support but the team was unable to determine where the bug in the code was.
2. The fuel latch did not work as expected. During testing prior to the final day, it did cause the engine to stop. However, this did not occur on the board that was used during the presentation. This could be in the haste to correct the engine problem; a switch related to the fuel latch could have been left commented out or changed inadvertently.
3. The USS Neversail could not play the prescribed music. Code was attempted to do this but the troubleshooting of the engine took priority.

**Final Project Report Requirement**

The final project should include the following sections:

1. Cover letter:
   1. The project title
   2. The name of the team members
   3. Project date (Spring 2012)
   4. Should include a picture of the equipment and/or a picture relevant to the project
2. Table of contents
3. List of figures and tables
4. Introduction:
   1. Describe the goal and the purpose of the project.
   2. Assumptions made
5. Design:
   1. Show the peripherals that are used in the project and what they are used for
   2. Software implementation of the project:
      1. Give a high level description and discussion
      2. General system flow chart
      3. Flow charts for each module
      4. Error handling and fail safe techniques
   3. Changes made in the design
   4. Additions to the project
6. Description of the division of work between team members
7. Description of which parts of the proposed project is working and which part is not working
8. Conclusion
9. Discussion and suggestions for future improvements on your project
10. Appendixes:
    1. User Manual
    2. Code:
       1. The code should be commented (useful and meaningful comments)
       2. Description of each file, subroutine and procedure:
          * Name
          * Inputs/Outputs and method of passing the parameters
          * General Description
11. References

**Note:**

* Use headings and subheadings throughout the report
* Pay attention to grammatical and spelling errors
* All figures and flow charts should be done using software (paint, Visio etc.)
* The report should be consistent in style
* Fonts and sizes:
  + Single spaced
  + Use the “Times New Roman” font or any similar font
  + Use font size 14 bold for headings
  + Use font size 12 for subheadings
  + Use font size 12 for text
  + Use the “Courier New” font for the code and the size should be 10

**Project requirements:**

**Boat**

**Objective:**

Write an assembly program to simulate a Boat.

**Requirements:**

* The user should first start the engine, by pushing start and entering the password.
  + The vehicle should start in a “stop” mode.
  + The vehicle should be at “stop” to shut down the engine.
* After starting the user must select the boat’s speed.
  + The boat will have 4 speeds Stop, High, Medium, and Low
    1. DC motor with 3 distinct speeds for High, Medium, and Low.
    2. The DC motor will gradually increase, or decrease, to the given speed.
* The direction is control by the potentiometer.
* The center position of the POT will be forward and any change will turn the stepper motor in that direction at a speed based on the distance from center.
* The speed of the boat (High, Medium, Low, and Stop) should be displayed
* After starting the boat the user can use the horn, menu, and spot lights (Spot lights can be used in any situation).
  + There must be two separate spot lights.
* The boat should display the fuel level.
  + The boat shuts down if the fuel tank is empty.
  + The boat must refill with gas before it can start again.
  + The fuel latch must be open to refill.
    1. If the fuel latch is open, and the boat is moving the boat shuts down.
* The system has a timer that displays the time (hh:mm:ss Day). The user should be able to change the time and the date.
* The main light will be controlled by the clock, turning on for the time period 6pm to 6am.
* No delay loops are allowed, you must utilize the Real Time Interrupt.

**Use of peripherals:**

1. Stepper-Motor:

Boat motor direction, speed and direction of turn is based on pot’s distance from a center value.

1. DC-Motor:

Main engine (Pulse Width Modulated to for 4 speeds) must have smoothed chance between speeds

1. Keypad:

Select different options and provide functionality for:

1. Enter PW
2. Setting the time
3. Select song
4. Start Button / Shutdown Button
5. Potentiometer:

Direction of the motor

1. LCD:
2. Menu
3. Display the fuel level
4. Display the speed
5. View time
6. Push Button:

Horn

1. Switches:

Fueling switch and fuel latch. Key for the Boat

1. Speaker (Port T):

Horn and music (5 different songs >5seconds each)

1. LEDs

Lights and spotlights on the boat.

1. IRQ:

Engine Shut Down

1. RTI

**Note:**

You are encouraged to be creative and make this project your own. You can make reasonable assumptions in the development of this project, but keep in mind that the assumptions **must** make sense to the user (and to the Lab TA).

If you have any questions pertaining to this project please discuss it with your Lab TA as early as possible.

You may use any C code provided through the lab but all other code must be written in assembly.

CODE: