Final Project: Pontoon

Names

Spring 2014



**Table of Contents:**

List of Figures and Tables 3

Introduction 4

Design: 5

a. Peripherals 5

b. Software implementation 9

c. Design Changes 12

d. Project Additions 13

Division of Work 14

Functionality of Project 14

Conclusion 15

Future Improvements 15

Appendix I: User Manual 16

Appendix II: Code 18

Appendix III: Project and Report Requirements 91

**List of Figures and Tables**

Figure 1: Motorola 68HC12 Microcontroller 5

Figure 2: Stepper Motor / DC Motor / Speaker 6

Figure 3: Keypad 7

Figure 4: LCD Screen 7

Figure 5: Potentiometer / Push Button / Switches / LEDs / IRQ 8

Figure 6: General System Flowchart 10

Figure 7: Main Menu Flowchart 11

Figure 8: Engine Menu Flowchart 11

Figure 9: Options Menu Flowchart 11

Figure 10: Status Menu Flowchart 12

Figure 11: Radio Menu Flowchart 12

**Introduction**

Purpose:

The purpose of this project is to serve as a final test of the comprehension of the knowledge learned throughout the Spring 2014 semester in ECE 362: Microprocessing Systems and Interfacing. Ranging from the basic understanding of assembly language to an understanding and comprehension of the 68HC12 microcontroller to a thorough understanding of interfacing separate programs in order to run a functional, sophisticated system, this project is the culmination of the ideas and lessons learned throughout the semester. This pontoon is equipped with an ignition, general motor controls, speed control, a status menu, clock, and a radio as per the general requirements. It is also equipped with a password confirmation, engine start up sequence, gear display, a horn, and an LCD display screensaver. This system was mainly programmed in assembly language with the Freescale CodeWarrior IDE with some select coding in C. The hardware used was the Motorola 68HC12 Microcontroller.

Assumptions:

Basic assumptions used throughout this project include specific times for the different RTIs in order to secure the functionality of the project. Multiple timing variables had to be changed and manipulated in order to do so.

All timing variables are located in the Variables.asm code along with all other significant variables including password values, keypad sequences, songs, etc.

Other basic assumptions included the fuel values and RPM in regards to how quickly fuel is used as different RPM values. Modeling the engine off of real pontoon engine data did this. The potentiometer value was multiplied in order to scale the value to the maximum RPM value of 5250. The fuel used in based off of RPMs that the motor is running at any given time with more fuel being used at higher RPMs.

**Design**

**Peripherals:**

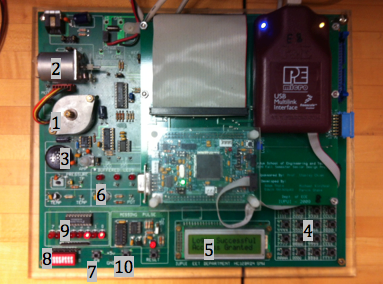


Figure 1: Motorola 68HC12 Microcontroller

The Motorola 68HC12 microcontroller, as seen above, was the main piece of equipment used in the design and programming of the pontoon. The 68HC12 has a set of 8-bit accumulators, A and B, which cascade into a 16-bit accumulator, D. It also has two 16-bit registers, X and Y that we used mainly for calculations and storing variables. The microcontroller also has a diverse set of peripherals that allow us to effectively represent a multitude of different systems on the pontoon. Each separate peripheral and its use in the pontoon is described below.

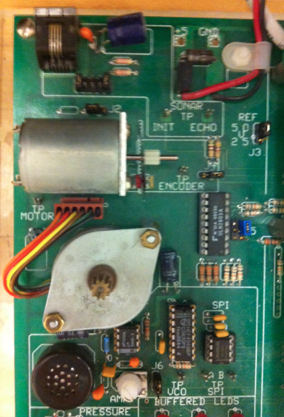


Figure 2: Stepper Motor / DC Motor / Speaker

1. Stepper Motor: Serves as the transmission of the pontoon. When rotating clockwise the engine is in forward gear, when the motor is still the engine is in neutral and when the motor is rotating counter-clockwise the engine is in reverse.

2. DC-Motor: Serves as the engine itself. The speed of the motor directly correlates to the speed of the engine. The higher the RPMs of the engine, the faster the DC motor rotates.

3. Speaker: Serves as the speaker for the radio on board the pontoon as well as an indicator that the engine is broken.

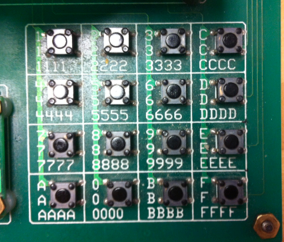


Figure 3: Keypad

4. Keypad: Serves as the user interface to the pontoon computer system as well as the engine controls.



Figure 4: LCD Screen

5. LCD Screen: Serves as the monitor for the user to be able to navigate the menus and options of the pontoon computer system and engine controls

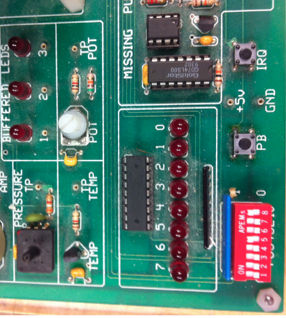


Figure 5: Potentiometer / Push Button / Switches / LEDs / IRQ

6. Potentiometer: Serves as the speed control for the engine. Rotate the potentiometer clockwise to increase speed and counter-clockwise to decrease the speed.

7. Push Button: Serves as the engine ignition. Once the system is turned on and safety checks are satisfied, pushing the button starts the engine.

8. Switches: Serve as selectors in regards to forward/neutral/reverse as well as turning the engine power on and adding fuel.

Switch 7: serves as the power to the engine.

Switch 6: refuels engine

Switch 5:

Switch 4: serves as DC Motor power, always high.

Switch 3:

Switch 2:

Switch 1: serves as forward if high, reverse if low

Switch 0: serves as neutral gear, overrides Switch 1

9. LEDs: Serve as indicators for the power to the engine as well as the engine itself.

LED 7: engine cylinder

LED 6: engine cylinder

LED 5: engine cylinder

LED 4: engine cylinder

LED 3: Reverse Gear

LED 2: Neutral Gear

LED 1: Forward Gear

LED 0: shows power to the engine

10. IRQ: Serves as an engine “malfunction” that needs to be fixed.

**Software Implementation:**

**Discussion:**

The software implementation was laid out in the fashion shown below in the general and module flow charts. Each individual portion of the system had its own .asm file. All of the variables are located in a single file, Variables.asm. The initial values of these, which are set when the program is started, are located in the rInitVals.asm file.

Following the general flow chart seen below, the main program simply calls upon the initial variables and waits for an interrupt in order to open and use all of the other programs.

The menu programs serve as the user interface for the pontoon systems. The user navigates the options of the pontoon through the menu selections. Each program within the pontoon project is based on the hierarchy in the flow charts with all of the interrupts representing the parts of the system that do not require anything from the user, i.e. the consumption of gas.

**General Flow Chart:**

rInitVars.asm

Main.asm

Variables.asm

Wait for RTI

Keypad.asm

menuDisplay.asm

4radioMenu.asm

3statusMenu.asm

2optionsMenu.asm

1engineMenu.asm

0mainMenu.asm

Figure 6: General Flow Chart

**Module Flow Charts:**

0mainMenu.asm

Keypad.asm

Figure 7: Main Menu Flow Chart

1engineMenu.asm

Keypad.asm

Engine.asm

CheckEngine.asm

Potentiometer.asm

Figure 8: Engine Menu Flow Chart

2optionsMenu.asm

Keypad.asm

rLogin.asm

repairEngine.asm

Figure 9: Options Menu Flow Chart

setPassword.asm

setTime.asm

Keypad.asm

3statusMenu.asm

Variables.asm

Figure 10: Status Menu Flow Chart

Keypad.asm

4radioMenu.asm

Variables.asm

Figure 11: Radio Menu Flow Chart

**Error Handling and Failsafe Techniques**

Failsafe techniques implemented into the program include:

* Engine will not start unless in Neutral
* If the engine breaks, it must be fixed before it can start again
* Password confirmation for initialization sequence
* No access to engine controls or other menus until logged in with proper password

**Design Changes:**

No significant design changes were implemented into the design of our pontoon beyong the additions listed below and the

**Project Additions:**

Beyond the required design of the pontoon we have incorporated the following additions:

Horn

The horn acts as a general horn when pressing the Push Button (PB) with the pontoon running. It also serves as the warning signal for the engine breaking down.

Scrolling status screensaver

The scrolling status screensaver runs on the LCD screen once no action has been taken for a certain amount of time. This screensaver displays the fuel level in the pontoon.

Gear display

The gear display is located on the LED display. LED1 represents Forward, LED2 represents Neutral, and LED3 represents Reverse. The Neutral gear overrules the Forward and Reverse gears, meaning that if the Neutral gear is on—LED2 is lit—the pontoon is in Neutral disregarding the other two gears.

Password confirmation

As another security measure, any new password that is entered has to go through a redundant check to make sure that the user truly put in the new password that they wanted. This guards against the user mistakenly putting in a wrong password and locking the system indefinitely.

Engine Startup Sequence

When the engine is starting, the LEDs light up in a pattern and then LEDs 4-7 light up in order to represent the engine cylinders while the engine is running.

**Division of Work**

This project was divided into two parts in order to divide the work to be done. Both Anthony Short and Scott Milam worked on the layout and organization of the project with input as to direction and additions to the project being determined by both. Anthony completed the vast majority of the coding with help on specific debugging and coding ideas. Scott completed the vast majority of the report with specific help with wording and organizational layout.

**Functionality of Project**

Working:

Every part of the project is virtually functioning correctly. In order for this to happen the different RTI timing variables had to be manipulated multiple times.

The LCD display as well as the LEDs and the motor are working perfectly for each of the different functions.

Not Working:

The only section of the project that could be said to not be functioning properly is the variable speed of the pontoon. Whereas we had initially intended to have the speed have a continuous spectrum, due to the timid nature of the potentiometer we had to set discrete values for the speed. We did this with a surrounding range of values that would effectively show the same value for the speed. The largest reason for this was trying to make the fuel calculations more accurate.

**Conclusion**

In conclusion, this final project and report has challenged us to the extent of the topics learned in ECE 362 this spring. Everything from the basic understanding of assembly language and the microcontroller to the integration of multiple programs into a working system have been implemented into this project.

**Future Improvements**

Future improvements to the pontoon program that we have discussed include cruise control, an extended password, password stipulations that have to be met in order to guarantee a secure password, and….

Cruise Control

Cruise Control would work in much the same way it does in a car. Setting the control at the current speed, and then keeping the speed constant until deselected.

Extended Password

The extended password would simply take a larger memory space to hold.

Password Stipulations

The password stipulations would be dependent on the extended password, but would include such stipulations as having at least 1 of each, letter and number, or other stipulations. This would be checked with a number and a letter counter. Going bit by bit, once a letter is found in the sequence then it starts looking for numbers. If at any time the entire password has been stepped through and either counter is still 0, then the system rejects the new password and asks for another. The check is done before the redundant password check and the rejection occurs after.

**Appendix I: User Manual**

Contents:

Engine 15

Start

Stop

Change Gears

Speed Control

Cruise Control

Show RPMs

Refuel Engine

Repair Engine

Clock 16

Show Time

Set Time

Security 16

Set Password

Radio 16

Power ON

Power OFF

Change Song

**Engine**

Start Engine

Prior to starting the engine multiple stipulations must be met.

1. Transmission must be in Neutral. (See “Change Gears”)

2. Power to engine must be turned ON. This is indicated by LED 0 and is turned ON/OFF by Switch 7.

Switch 7 (high): Engine Power ON

Switch 7 (low): Engine Power OFF

Once these stipulations are met, push the push button (“PB”) on the 68HC12 to start the engine.

Stop Engine

To stop the engine, simply push the push button (“PB”) on the 68HC12 while the engine is ON.

Change Gears

The gear that the engine is currently on is indicated by LEDs 1-3.

LED 1 (ON): Forward

LED 2 (ON): Neutral

LED 3 (ON): Reverse

Neutral (Switch 0) overrides Forward/Reverse (Switch 1).

Switch 0 (high): Neutral

Switch 0 (low): Forward/Reverse

To change the gear between Forward and Reverse, change Switch 1.

Switch 1 (high): Forward

Switch 1 (low): Reverse

Speed Control

The speed of the pontoon is determined by the RPMs of the engine. In order to increase/decrease the RPMs of the engine, rotate the potentiometer

(See “Design: Peripherals”) on the 68HC12.

Rotate Clockwise: Increase RPMs (speed)

Rotate Counter-Clockwise: Decrease RPMs (speed)

Show RPMs

From “Main Menu,” select the “Status Menu” by hitting 3 on the keypad.

Refuel Engine

First, turn off the engine. (See “Stop Engine”)

Next, flip Switch 6 from low to high.

>>If Switch 6 is already high, first set to low, then reset to high.

Repair Engine

From “Main Menu,” select the “Options Menu” by hitting 2 on the keypad.

Next, select “Repair Engine” by hitting 3 on the keypad.

**Clock**

Show Time

From “Main Menu,” select the “Status Menu” by hitting 3 on the keypad.

Set Time

From “Main Menu,” select the “Options Menu” by hitting 2 on the keypad.

Next, select “Set Time” by hitting 1 on the keypad.

Next, enter the day of the week (1-7) with Monday being 1.

Next, enter the hour, minute, and second values.

>>Military Time (13:00 = 1:00 pm)

>>Enter two numbers for each value (“02” = 2)

**Security**

Set Password

From “Main Menu,” select the “Options Menu” by hitting 2 on the keypad.

Next, select “Set Password” by hitting 2 on the keypad.

Enter your new 4-character password.

**Radio**

Power ON

From the “Main Menu” select the “Radio Menu” by hitting 4 on the keypad. From there hit 1 on the keypad to start playing a song

Next Song

From the “Radio Menu” select 2 on the keypad.

Previous Song

From the “Radio Menu” select 3 on the keypad.

Power OFF

From the “Radio Menu” select 4 on the keypad to stop the songs from playing.

**Appendix II: Code**

***Main Code***

**main.asm**

**Variables.asm**

**Appendix III: Project and Report Requirements**

**Project Requirements:**

***Pontoon***

***Objective:***

*Write an assembly program to simulate a Pontoon. (Engine and Transmission)*

***Peripheral Simulations:***

1. *Stepper-Motor:*

*Output of transmission. Clockwise for forward, still for neutral, Counter-Clockwise for reverse.*

1. *DC-Motor:*

*Engine. (should change speed based on speed of boat, (rpms))*

1. *Keypad:*

*Select different options and provide functionality for:*

1. *Set the time: The user should be able to set the time (hh:mm:ss Day).*
2. *Enter the password for starting*
3. *Be able to change password*
4. *Select to repair engine through the menu options*
5. *Potentiometer:*

*Controls engine rpm(rotation per minute).*

1. *LCD:*
2. *Menu*
3. *Display: Fuel Level, RPMs*
4. *View time and mileage*
5. *Push Button: Start engine.*
6. *Switches: Shift gears 1, N, and reverse. Turn power on to engine. Add Fuel.*
7. *Speaker (Port T):*

*music (3 different songs >5seconds each)*

1. *LEDs: 4 LEDs will simulate the sparkplug of a 6 cylinder engine, 1 LED will show power to engine.*
2. *IRQ: Throws an engine rod and a knocking sound must be sent to speaker. If engine does not shut off within 6 seconds then the engine breaks and automatically shuts off.*
3. *RTI control timing.*

***Requirements:***

1. *The engine (DC motor) used to simulate engine rpms.*
   * *In order to start the pontoon, the transmission must be in neutral.*
   * *The DC motor will increase in Revolutions Per Minute when the potentiometer resistance increases.*
   * *If the IRQ is pressed then the engine throws a rod and a knocking sound must be sent to speaker. If engine does not shut off within 5 seconds then the engine breaks and automatically shuts off.*
   * *If this happens threw the menu, using the hex keypad, select to perform a repair before the engine can restart.*
2. *The transmission (Stepper motor) will turn clockwise for forward or counterclockwise for reverse. In order for the pontoon to move it must start in first gear or reverse. If in forward direction as the gears increase the stepper motor rotation speed will increase.*
3. *No delay loops are allowed, you must utilize the Real Time Interrupt. DELAY LOOPS ARE ONLY ALLOWED OR HEX KEYPAD DE-BOUNCE*.

**Report Requirements:**

***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 (Fall 2013)*
   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*