Western Washington University

Electrical Engineering Technology

Etec 471

Professor Morton

Senior Project Definition

Automated Aquarium

Greg Rose

12/10/03
**Introduction**

Having a large aquarium in a room provides a peaceful form of moving art, but monitoring and maintaining an aquarium can be time consuming. I am proposing a fish aquarium monitoring and control system that monitors the temperature and pH of an aquarium and controls the lights, temperature, filtration, and feeding. This will be done through a user interface with a LCD display.

Some fish are expensive. So to keep them alive it is important to monitor their water conditions. Also you must remember to feed them and change the water every few months. This project would control feeding and remind the owner to change the water at a scheduled time or if the pH of the water changed to outside acceptable programmed range. Over feeding is harmful to aquarium water quality, bad for filtration, and stressful for fish. This project would control feeding as well as turn off the filter during feeding to increase life of filters. Having a microcontroller monitor and control an aquarium would mean less time maintaining your fish and more time to enjoy them.

**General Description**

The block diagram for this project is Figure 1 on page 1. It shows that the microprocessor will receive inputs from the two sensors, temperature and pH. Each signal will go through signal conditioning of op-amps and a low-pass filter into the microcontroller’s A/D converter. The user input will allow the operator to change many settings, such as: lighting times, feeding times, water temperature, and water change reminders (warnings.) This will be done through four keys located just under the LCD screen. When in the main screen, the LCD will display the current time, temperature, pH,
as well as any reminder that is set. From the main screen the operator can chose the menu and select options to change.

The heater, lights, filter, and auto feeder are controlled by the microcontroller. They will plug into the external box on the floor and be controlled by the microcontroller through a relay. Standard heaters, lights, and filters may be used. The lights and filter will just be left in the on position and the microcontroller will control them. By doing this a normally open relays will drive these devices on when the microcontroller energizes relay. The filter will be stopped for a set time when the microcontroller feeds the fish. The automatic fish feeder will be control by a D.C. stepper motor connected to the center of a round lid with many holes cuts out (a single feeding.) When the stepper motor turns, the holes will rotate until lined up with single hole on the bottom and the food will fall through.

![Figure 1: Basic System Block Diagram](image-url)
The maximum dimensions for this project are shown in Figure 2 on page 3. These dimensions are large enough to ensure all components can be easily placed into their appropriate box. The final design for this project would be much smaller because the evaluation board for the Star 12 microcontroller would not be present, like in this prototype. I will assemble the two boxes for this project when more circuitry for this project is laid out and tested. The control box will house the display, LEDs, control buttons, the microcontroller, and all 5 volt DC devices. The input lines from my sensors will come into this box for signal conditioning and processing. This control box will be supplied by a 5 volt DC wall wart which I already have. It supplies 5 amps of current (max) which is much more current than this project will ever draw. Wires exiting this box will be control lines for the stepper motor and relays. The control lines for the relays will enter the external box on the floor. This box will house the relays and the plug-ins for the heater, filter, and hood lamps. To energize common heaters, filters, and hood lamps, 120 volts AC will supply this box.
**Functional Description**

For this project I have chosen to use the Motorola MC9S12DP256B (Star 12) microcontroller. I chose to use this microcontroller because I will be using it in ETEC 454 and I have access to all of the developmental tools. Also because of its onboard memory and many I/O (input/output) lines will allow my to add additional features (redux sensor, auto water change, or data logger) to this project in the future. The major peripheral hardware will include a 2 X 16 LCD (liquid crystal display), 4 pushbuttons, a high accuracy 16MHz crystal, 2 LEDs (light emitting diodes), a stepper motor, a pH sensor, and a temperature sensor. Figure 3 on page 5 shows the functional hardware diagram for this system.

The Star 12 will have more than enough onboard memory to store and run the program to control this automated aquarium. The program will be stored on 32K bytes of fixed flash. The 12K bytes of RAM (random access memory) will hold all of the variables of the program and the kernel. I plan to use Micro/COS for my real-time kernel.

The 2 X 16 LCD will require 11 output bits, which include all of Port A and part of Port K (PK2-PK0.)

Bits (PK6-PK3) will be outputs to ULN2003 (stepper motor driver.) By rotating one or two bits high I can achieve either a full or half step respectively.

Part of Port B (PB3-PB0) will be the user input buttons (UP, DOWN, ENTER, MENU.) With these four buttons, a user will be able to completely set and control this system.
The inputs from the pH and temperature sensors will be conditioned and input into ATD0 (PAD00, and PAD01.)

Port T will output to the relays and LEDs. When the microcontroller energizes the bit for the relay it is controlling on, the relay will close. This will turn on the device connected to that relay. The reset circuitry will ensure that the Star 12 can only operate at a constant 5 volts since low voltage can cause calculation errors.

Ports M and K will control and receive RTC (Real Time Clock) information for the DS1511 which is a battery backed RTC. The DS1511’s crystal should keep the real time clock fairly accurate. It should keep the time within 1 minute per month. Also the DS1511 is battery backed so the real time will not be lost in a power failure.
Figure 3: Functional Hardware Diagram
**Software Description**

I plan on writing code for this project in a combination of C and assemble. Since this project will have a lot of complex code, I plan to write most of it in the higher level programming language C. I plan to use assemble language for my simpler tasks like stepping the motor. The software for this system is composed of the modules below.

<table>
<thead>
<tr>
<th>Module</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>KERNEL</td>
<td>MircoC/OS is a real-time preemptive multitasking kernel.</td>
</tr>
<tr>
<td>MAIN</td>
<td>Main Module that monitors system state.</td>
</tr>
<tr>
<td>DISPLAY</td>
<td>Initializes and writes data to the LCD.</td>
</tr>
<tr>
<td>ATOD</td>
<td>Gets digital words from A/D converter.</td>
</tr>
<tr>
<td>REFRESH</td>
<td>Update real time clock.</td>
</tr>
<tr>
<td>RUNRELAY</td>
<td>Update the state of the relays.</td>
</tr>
<tr>
<td>DEBOUNCE</td>
<td>Denounce pushbuttons and monitor their states.</td>
</tr>
</tbody>
</table>

**User Interface**

Four pushbuttons will allow the user to control the automated aquarium. They will be labeled UP=U, DOWN=D, ENTER=E, and MENU=M. See Figure 4 State Diagram on page 8.
Figure 4: State Diagram
After power up and an initialization screen the LCD will display the main operating screen (see Figure 5.) This screen will display the current date, time, water temperature, and pH. From this screen the only button that will change screen’s display is MENU.

![Figure 5: Main Operating Screen](image)

Pressing the MENU will enter the system into the menu mode (see Figure 6.) When in the menu mode the user can scroll up or down through the menu options with UP or DOWN. When the user has found the setting they wish to change, they then press ENTER to change that setting. Menu settings will include set time, lamp time, set temp, pH range, # feeding, and change H2O. At any point the user can use the MENU button to toggle between the main operating mode and the menu mode.

![Figure 6: Menu Mode First Screen](image)

If the user presses DOWN at this point, the system will remain in the menu, but give option of next menu setting to select. It will scroll through all setting options listed above (see Figure 7.)

![Figure 7: Menu Mode](image)
If the user selects the *set time* option this screen with display allowing them to set time (see Figure 8 and 9.) Bold Italics denote flashing characters.

![Set Time Screen](image)

*Figure 8: Set Hour*

Once the correct hour is chosen by using UP and DOWN the user would press ENTER and select the appropriate minutes. After the appropriate minutes are selected the user would press enter again and the system would return to its main mode with new set time displayed.

![Set Time Screen](image)

*Figure 9: Set Minutes*

If the Menu option *lamp time* is selected the user can set what time the system will turn the lamps on and off (see Figure 10 and 11.) Setting the lamp’s on and off time is much like setting the clock twice (described above), once for an on time and once for an off time.

![Set Time Screen](image)

*Figure 10: Setting Hood Lamp Time*

![Set Time Screen](image)

*Figure 11: Setting Hood Lamp Time*
If the menu option *set temp* is selected the user will be able to set the temperature of the aquarium water (see Figure 11.) The automated aquarium system will control the heater in the aquarium to heat to that temperature. While the heater is on the LED in the control box will be lit.

![Image of temperature setting](image.png)

*Figure 11: Setting Hood Lamp Time*

When the user selects *pH range* from the menu mode the user will be able to select a safe pH range for the aquarium water (See Figure 12 and 13.) Should the pH of water change to outside of that safety range the system will light the change water LED.

![Image of pH range setting](image.png)

*Figure 12: Setting pH Range*

![Image of pH range setting](image.png)

*Figure 13: Setting pH Range*

If the user selects *# feeding* from the menu mode the user be able to select the number of feeding per day that the system will dispense to the fish (see Figure 14.) The system will automatically feed the fish when the lights are on. Also the system will turn off the filter in the aquarium for about 3 minutes after feeding to increase the life of the filter.

![Image of number of feedings](image.png)

*Figure 14: Setting Number of Feedings*
If the user selects change H2O from the menu mode the system will show the day set for next water change in the top left of the LCD display. Also it will allow user to select how long until next water change, if water was just changed. To denote it is time for a water change the change water LED will light.

![Figure 14: Setting Water Change Time](image)

**Development Plan**

This quarter I have been busy researching and ordering parts for this project. I have already received the pH and temperature sensors that I ordered. Also I already have tested the stepper motor that I have, and it works great. I plan on working mostly on hardware and testing it during the winter quarter, but I would also like to get started on the code as soon as possible. All development tools that I require for this project are in the ET340 lab at WWU. This lab contains BDM pods, development software, digital oscilloscopes, and a sink which I will require.

Since, this project requires 120 volt ac, I will need to do all wiring with the power off. Thankfully all testing and functionality test will not require 120 volts. Only the final product will require 120 volts and never will 120 volts be supplied to the floor box with the cover off. My class demonstration will consist of showing my design and maybe a chemical endo or exothermic reaction with chemicals of different pH to show the temperature and pH change, since this is an environment controller and doesn’t have the wow factor of other projects.
This is my development plan for work to be done in the weeks to come.

Winter Quarter 2004

Week 1: Make or find feeder
Week 2: Construct signal conditioning circuits for sensors
Week 3: Construct signal conditioning circuits for sensors
Week 4: Test and calibrate stepper motor to feeder
Week 5: Test hardware
Week 6: Test hardware
Week 7: Get parts laid out for box designs
Week 8: Make boxes
Week 9: Begin Software design
Week 10: Dead Week

Spring Quarter 2004

Week 1: Modify MicroC/OS module
Week 2: Modify Debounce module, All Hardware completed
Week 3: Modify / write display module, Hardware Review
Week 4: Modify / write A\D module test and calibrate sensor with the microcontroller
Week 5: Testing and finishing floor box, Software system presentation
Week 6: Put all together / Prototype testing, Software system presentation
Week 7: Prototype testing
Week 8: Prototype testing

Week 9: Code Review

Week 10: Dead Week

**Electrical Specification**

5 Volt Supply @ 5.0 Amps

Calculated worst case maximum current dissipation by wallwort **208.26 mA**

**Accuracy**

Time $^{+}1$ second/month

Temperature of water $^{+}0.5$ degrees Fahrenheit

pH .1 unit

**Range**

Temperature of water 50 – 100 degrees Fahrenheit

pH 1 – 14 units

Water change time 1 – 6 months

**AC max current dissipation when relays energized (closed)**

Lights 1.2 A

Heater 2.5 A

Filter 420 mA

**Expected Environment - Indoors**

**Operating Temperature**

60 – 100 degrees Fahrenheit
<table>
<thead>
<tr>
<th>Qty</th>
<th>Description</th>
<th>Part #</th>
<th>Source</th>
<th>Price</th>
<th>Power (max)</th>
<th>Lead time (Wks)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Microcontroller</td>
<td>MC9S12DP256B</td>
<td>Motorola</td>
<td>$ 14.08</td>
<td>65 mA</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>pH sensor</td>
<td>PHE1301-NB</td>
<td>Omega</td>
<td>$ 32.00</td>
<td>.5 mA</td>
<td>have</td>
</tr>
<tr>
<td>1</td>
<td>temp sensor</td>
<td>DCT-DIN</td>
<td>Vernier</td>
<td>$ 28.00</td>
<td>.15 mA</td>
<td>have</td>
</tr>
<tr>
<td>1</td>
<td>stepper motor</td>
<td>K82701-P2</td>
<td>USBid</td>
<td>$ 19.00</td>
<td>55 mA</td>
<td>have</td>
</tr>
<tr>
<td>2</td>
<td>instrumentation amp</td>
<td>AD-620</td>
<td>Analog Dev.</td>
<td>$ 6.54</td>
<td>0.1 mA</td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td>resistors</td>
<td>1/4W %1</td>
<td>DigiKey</td>
<td>$ 0.60</td>
<td>2.5 mA</td>
<td>have</td>
</tr>
<tr>
<td>3</td>
<td>relays</td>
<td>not chosen yet</td>
<td>n/a</td>
<td>$ 15.00</td>
<td>20 mA</td>
<td>4</td>
</tr>
<tr>
<td>1</td>
<td>feeder</td>
<td>n/a</td>
<td>E-Bay/ make</td>
<td>$ 30.00</td>
<td>n/a</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>boxes</td>
<td>n/a</td>
<td>Make</td>
<td>$ 30.00</td>
<td>n/a</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>120V 3 prong male plug</td>
<td>n/a</td>
<td>Home Depot</td>
<td>$ 1.00</td>
<td>n/a</td>
<td>have</td>
</tr>
<tr>
<td>3</td>
<td>120V 3 prong female plugs</td>
<td>n/a</td>
<td>Home Depot</td>
<td>$ 3.00</td>
<td>n/a</td>
<td>have</td>
</tr>
<tr>
<td>1</td>
<td>2x16 LCD</td>
<td>3-1033-ND</td>
<td>DigiKey</td>
<td>$ 15.33</td>
<td>3 mA</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>16 MHz Crystal</td>
<td>73-XT49U1600-20</td>
<td>Mouser</td>
<td>$ 0.41</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>LED</td>
<td>67-1096-NB</td>
<td>DigiKey</td>
<td>$ 0.10</td>
<td>5 mA</td>
<td>have</td>
</tr>
<tr>
<td>1</td>
<td>stepper motor driver ic</td>
<td>ULN2003</td>
<td>DigiKey</td>
<td>$ 0.40</td>
<td>40 mA</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>4.7V zener</td>
<td>1N4732</td>
<td>Jameco</td>
<td>$ 0.04</td>
<td>2 mA</td>
<td>have</td>
</tr>
<tr>
<td>1</td>
<td>MAX6314 reset circuitry</td>
<td>MAX6314US4S101-T-ND</td>
<td>DigiKey</td>
<td>$ 1.04</td>
<td>.012 mA</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>pushbuttons</td>
<td>JF15SP1H</td>
<td>Allied</td>
<td>$ 10.20</td>
<td>n/a</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>battery backed RTC</td>
<td>DS1511</td>
<td>DigiKey</td>
<td>$ 9.44</td>
<td>15 mA</td>
<td>2</td>
</tr>
</tbody>
</table>

For 5V wall wort total $216.18 208.26 mA