School of Engineering Science
Simon Fraser University, Burnaby, B.C. V5A 1S6
temperogroup@yahoogroups.com

January 14, 2005

Dr. Andrew Rawicz
School of Engineering Science
Simon Fraser University
Burnaby, B.C. V5A 1S6

Re: ENSC 340 Post-Mortem Report for *TemperSure*

Dear Dr. Rawicz,

Please find an attached copy of the Tempero Systems Post-Mortem Report for *TemperSure*. This document outlines the overall process in the development of the temperature control system for the shower.

The following report addresses the current state of system development, the deviation of the proof-of-concept system from the design specifications, budget and scheduling deviations, future plans, and personal experiences resulting from the project.

Tempero is composed of Stephen Au-Yeung, Francis Merin, Victor Tai, and Jason Wong, four highly competent fourth year electronics and computer engineering students with diverse strengths and technical experiences. If you have any questions or comments please to contact us by e-mail at temperogroup@yahoogroups.com. Alternatively, you may contact me directly by telephone at 604-254-7492.

Sincerely,

Jason Wong
CEO
Tempero Systems

Enclosure: Tempero Systems Post-Mortem Report for *TemperSure*

Tempero Systems

take control.

Post Mortem for a Shower Temperature Control System

Project Team:
Stephen Au-Yeung
Francis Merin
Victor Tai
Jason Wong

Contact personnel:
Jason Wong
jwongk@sfu.ca

Submitted to:
Andrew Rawicz
Mike Sjoerdsma
School of Engineering Science
Simon Fraser University

Date: January 13th , 2005

Table of Contents

List of Figures

List of Tables

1. Introduction

Tempero Systems envisioned designing, developing, and testing an electronic temperature control system for showers/baths. The goals of our project were to reduce the need for the "adjust-and-feel" method of temperature control, and to develop a solution to ensure comfort and most of all, safety, for shower/bath users. We aimed at providing a solution for automatically adjusting to water temperature changes for commercial/institutional uses to reduce the need for manual valve maintenance.

The following document assesses the actual process of developing the *TemperSure* system and assesses how well we attained our project goals.

2. Current State of the System

The block diagram of the current *TemperSure* system is presented in Figure 1.

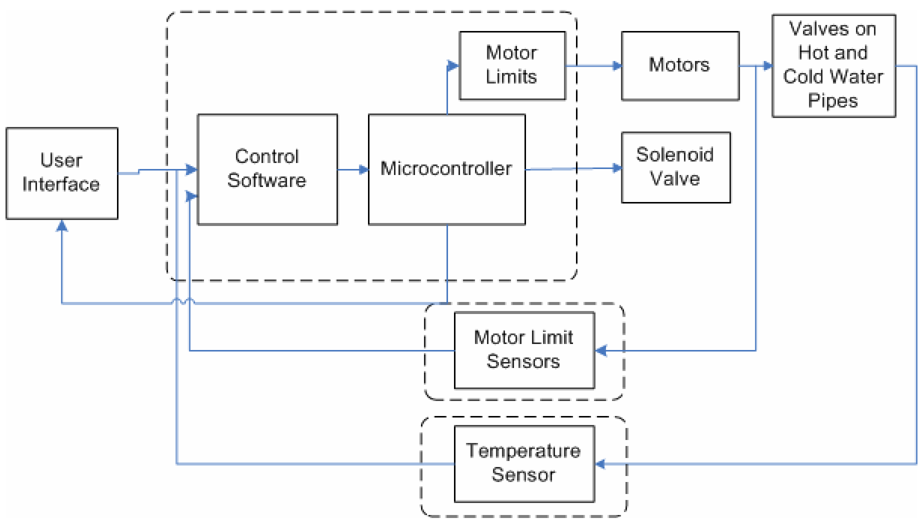

Figure 1: Block Diagram of Current TemperSure System

The *TemperSure* system provides automatic adjustment of water temperature within the 25°C to 33°C temperature range. The motors must be manually calibrated to a fully closed cold water valve and a hot water valve one and a half turns open from its fully closed position. From our test bench and testing facilities, we have concluded that the system is limited in attaining a desired temperature whose difference in large compared to the current temperature because of the physical limits of the valves, and the limited hot water temperature of the input water flowing through the customized plumbing of the system.

2.1. Temperature-Sensing Unit

The functionality of the current temperature-sensing unit closely resembles the functionality in the final product. It can provide accurate temperature feedback to the temperature controller module and acts as an anti-scalding unit at the same time. Users can adjust their cut off temperature from 20°C to 60°C. In addition, a manual override button is available for disabling the whole anti-scalding feature.

2.2. Valve Position Sensing Unit

The valve sensor module is used to prevent the motor from allowing the valve to reach its physical limits. This circuit is built using an infrared emitter and multiple receivers for positioning. Currently, the valve sensors are functioning as they are designed to. They accurately, within the devices' specifications, turn on when the receiver receives the emitter's infrared light. The construction of this circuit was relatively simple. We had used a circuit design suggestion from a web conference hosted by National Semiconductor [1]. With this design, we did not have many problems with the valve sensor unit.

2.3. Valve/Motor Control Unit

The valves we planned to use, half-inch stop valves, proved to be incompatible with our functionality. The stop valve will not control the water flow, but rather stop the water flow after reaching a certain point. We changed our valve selection to a half-inch gate valve, which works similar to that of a plunger valve. The gate valve has a metal gate that limits the flow of water rather than a rubber plunger.

Figure 2 shows the current design for the motor/valve control unit.

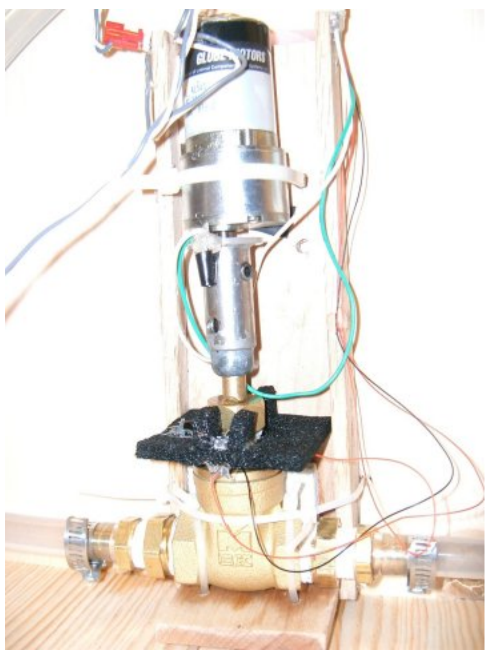

Figure 2: Current design of the motor/valve unit

Because the optical sensors we are using have a large sensitivity area, we narrowed the area by limiting the angles at which the infrared is emitted and received. Although not obvious, Figure 2 shows the emitter covered with electrical tape on all surfaces but the ones facing the receivers and the receivers are encased in a tunnel-like construction to narrow the receiving angle.

The DC motors are of the same type outlined in the Design Specification.

2.4. Temperature Control Unit

The temperature controller unit attains and maintains the user specified temperature output and consists of a software program that runs on a Microchip PIC18F452 microcontroller. As detailed in the Design Specification, by moving the valves to control the flow of water, the temperature controller unit can modify the output temperature of the unit.

The interaction of the firmware modules of the temperature controller unit are presented in Figure 3.

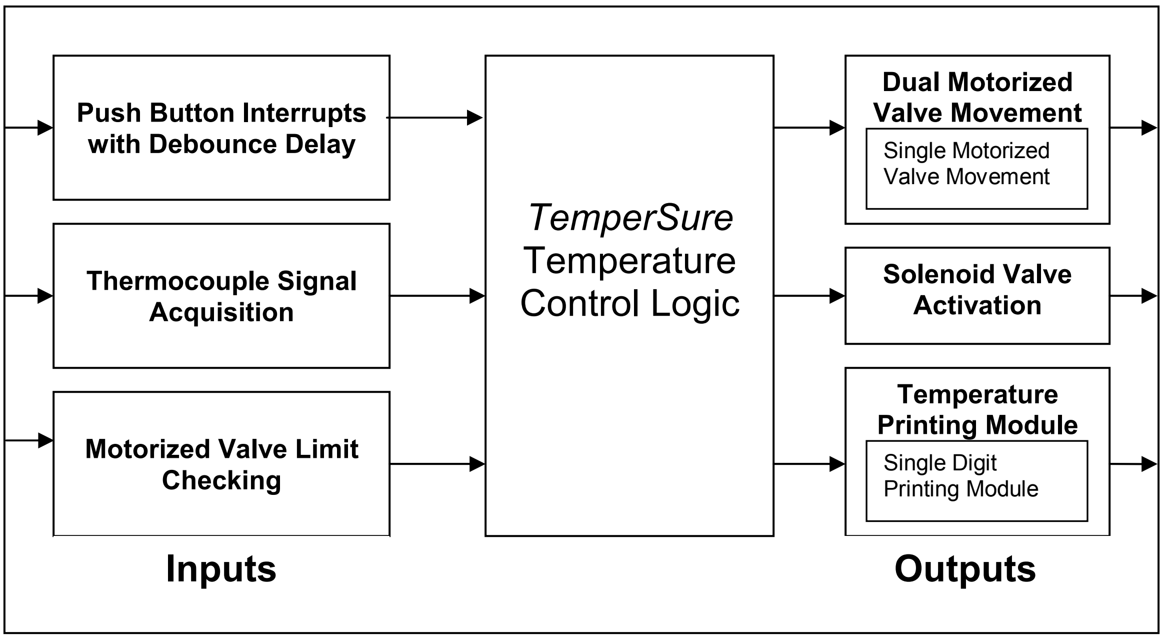

Figure 3: Interaction of Firmware Modules

Currently, this unit functions as planned in the Design Specification. The temperature will change according to the user's desired temperature, with an accuracy of ±1°C. Although the performance of this unit is limited by the valve unit, we can deduce from independent and integrated unit tests that the unit is functioning to our expectations. However, we found several issues during our software development. When the current water temperature is approaching the user-desired temperature, the outputs of the controls to the motor unit will oscillate between logic 1 to a logic 0. We are not able to determine the source of this problem, but we suspect that it may be a software related bug. Another issue we have had is the lack of timers on the microcontroller. In our original design, we did not expect the number of timers needed to exceed four. Currently, the blinking of the display, motor movement module, button debounce delay, and A/D capture delay are using timers to accomplish their tasks. The A/D capture delay is one of the uses of timers that we did not expect.

2.5. User Interface Unit

The user interface unit of the proof-of-concept system is presented in Figure 4 below.

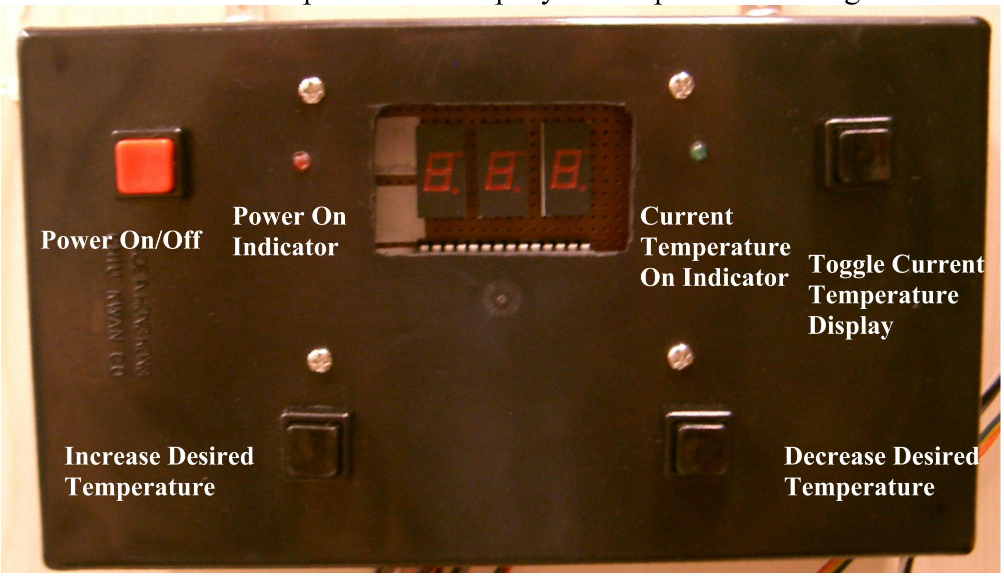

Figure 4: User Interface Unit

The user interface unit was constructed with the seven-segment LED displays, four push buttons, two LEDs, and an inexpensive plastic case. The four push buttons provide the following features: Power on/off, Toggle current temperature, Increase desired temperature, and Decrease desired temperature. The display blinks when the system is attempting to achieve the desired temperature entered by the user.

Interfacing the display with the microcontroller was a straightforward task. However, debouncing the push buttons for a quick yet accurate response proved to be a non-trivial task, which required trial-and-error testing with different scale factors of the timer module allocated for this task.

3. Deviation of the System

Throughout the development process, some hardware and software modules did not behave as initially anticipated in the Design Specification. We discuss changes made in the design of system components to compensate for the deficiencies discovered during the implementation phases.

3.1. Temperature-Sensing Unit

While the functionality remains closely the same, the method of implementing the circuit changes completely from our original design. As specified in the function specification, our original approach to implement this unit is through the use of a Schmitt trigger. However, while the unit can be implemented with a Schmitt trigger, the random initial condition of the Schmitt trigger makes the circuit very unstable.

As the result, we have redesigned our circuit and the new design is shown in Figure 5.

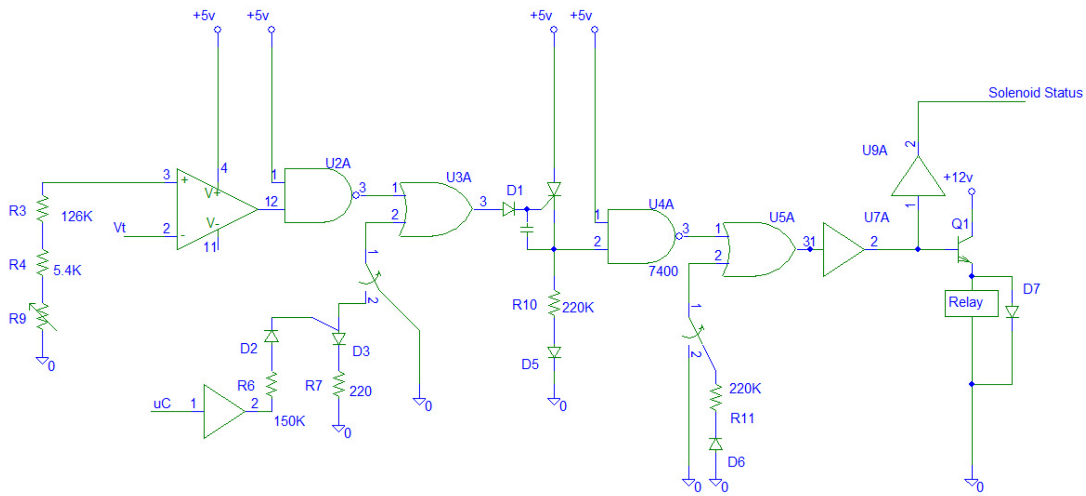

Figure 5: New circuit for the temperature-sensing / anti-scalding unit

The new implementation utilizes a couple of SCR (Silicon Controlled Rectifier) as a one-time switch trigger. When the current temperature reaches above the threshold, a voltage will be sent to the gate of an SCR to introduce a logic low at the output of the second OR-gate, which shuts off the solenoid valve and stops the water flow.

Besides the implementation difference, another major function deviation from our test unit is the non-independent anti-scalding unit. A major issue we have encountered while constructing this unit is the high amount of power consumed by the solenoid valve. This power restriction dramatically reduces our options on the sources of power for this unit. As the result, we have decided to tie this unit with the rest of the system so power can be drawn from the main power supply of the system.

3.2. Valve Position Sensing Unit

The valve sensor unit did not deviate from our previous design document.

3.3. Valve/Motor Control Unit

The design of the motor/valve control unit is very similar to that outlined in the Design Specification. We deviate slightly in terms of the position of the emitter sensor. Note that in the Design Specification the emitter sensor is located on the opposite end of the coupler.

The mechanical coupler deviates slightly from the original design. The coupling mechanism to the motor stayed the same. On the valve end, we have attached a metal piece with a square hole that fits into the square shape of the valve shaft. The coupling principle to the valve is similar to that of a screwdriver to a screw.

Figure 6 shows the modified design for the H-Bridge circuit.

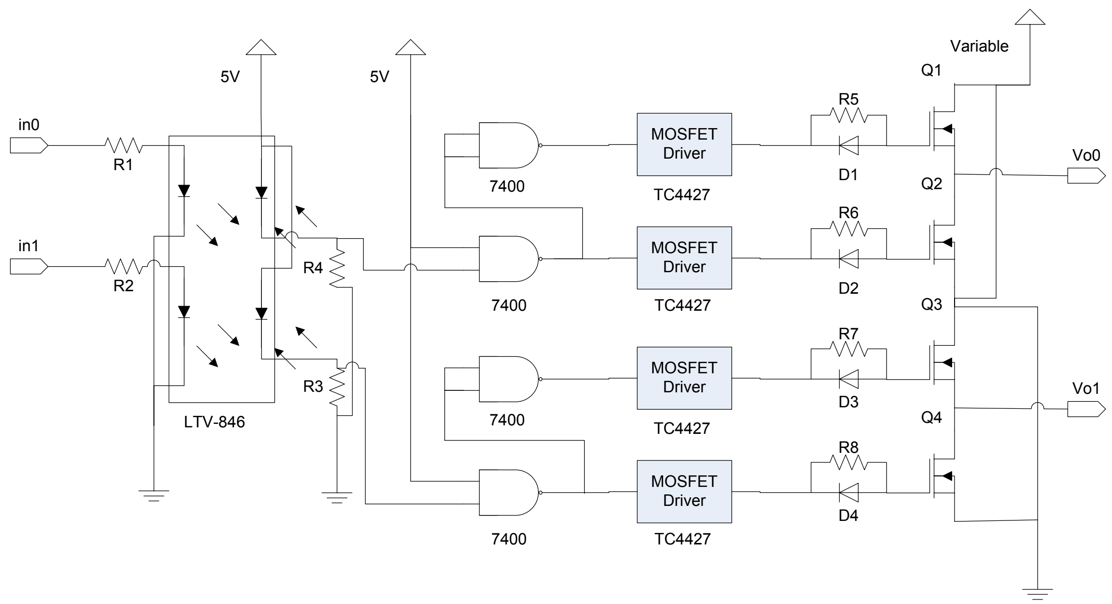

Figure 6: H-Bridge circuit.

The H-Bridge circuit shown in the Design Specification contained the basic components but modifications were made to create a stable circuit and a circuit that would suit our needs. We eliminated the enable control pin from the microcontroller, as we did not see a need for it. As a result, the NAND gate inputs previously connected to Enable were routed to 5V. To operate the optoisolator properly, we added resistors R1-R4. In order to operate the MOSFETs, we added the TC4427 MOSFET drivers powered by the Variable rail. To prevent excessive voltage at the gate of the MOSFETs due to switching, we added the free-wheeling diodes D1-D4 and resistors R4-R8.

To be able to test and fine-tune our motor response, we added a Variable rail. A LM317 variable regulator and a potentiometer allow us to change the motor speeds as needed.

3.4. Temperature Control Unit

The implementation of the temperature controller deviates slightly from our original design given in Figure 9 found in Appendix A.

The modified general software flowchart is shown in Figure 10 of Appendix A. The modified firmware separates the A/D conversion and motor limit checking from the temperature checking logic. The temperature checking logic is performed in the main loop of the firmware. As presented in Figure 11 of Appendix A, these operations are instead performed periodically in response to a timer overflow in an interrupt service routine.

As noted in the flowchart in Figure 10 of Appendix A, the software program will only move the motor in a direction that will attain the desired temperature. We feel that this is a better arrangement because we cannot accurately determine the position of the valves. Also, the valves we used do not have linear flow characteristics, making the software program to control the water flow impossible without other hardware additions. The decision to keep moving or stopping the motors has been moved into a timer. The main reason is to allow time for the motors to move so that temperature changes can occur. Since the water temperature requires time to change, the delay in making the decision can help reduce the chances of pulsating the motor units.

3.5. User Interface Unit

The functionality of the user interface unit was reduced because of problems encountered in firmware programming and speed of the water temperature changes influenced by the system.

A feature not included in the actual system was the incrementing and decrementing of the desired temperature by 2 or more degrees when the appropriate button is pressed for more than 2 seconds. The rationale behind the feature was to provide the user with a less tedious approach to entering his or her desired temperature if it differed from the default of 25°C by a significant amount. As mentioned previously, we had used up all the microcontroller timer modules for essential features. Therefore, since there were no more timer modules available to provide at least a 2 second delay, we omitted this feature from the proof-of-concept system.

Another user interface feature that was omitted from the actual system was the printing of error status messages on the display to indicate system errors. The code required for printing error messages could be derived from the code used to print the current and desired temperatures.

However, the problem with displaying error messages was based on the limitation of the seven-segment display driver. The display driver can operate in one of two decoding modes, or fonts: Hexadecimal, and Code-B. The decoding modes are linked to a driver pin with the shutdown display function. Hence, high, low, and floating (microcontroller pin set as an input) states indicate Hexadecimal, Code-B, and Shutdown functions, respectively.

Because we required the display of a “C” to indicate the temperature, in degrees Celsius, we used the Hexadecimal font. However, some of the desired characters for the error messages such as “E”, “H”, “L”, “P”, and “-“ could only be accessed in Code-B mode. Therefore, a potential solution was to remain in Hexadecimal mode during normal operation and then quickly switch to Code-B when flashing an error message. This solution would require that the mode pin be set as an input for a few milliseconds. However, such a delay would require a timer module on the microcontroller. As mentioned previously, no additional timer modules were available and hence the feature was omitted.

Finally, a push button to toggle between displaying the desired temperature and displaying the current temperature on the display was added along with a green LED placed next to the button to indicate that the current temperature is being displayed. Both of these functional additions accommodate the preferences of different users and reduce the need for another seven-segment display.

4. Assessment of Project Management

During the course of the project, discrepancies arose between the proposed and desired budgets and schedules. In this section, we assess the project management aspects involved in developing our system.

4.1. Budget

Table 1 shows the estimated cost for the development of *TemperSure* to be $671 however, the actual cost is $722.

Table 1: Project Budget

Part	Estimated Cost ($CND)		Actual Cost ($CND)	
Motorized Valves (two units)		**236**		**256**
· ½ inch valves	16		30	
· DC geared motors	135		175	
· IR sensors	20		15	
· Mounting equipment	50		0	
· H-bridge	15		36	
User Input		**15**		**25**
· LED display	10		8	
· Push buttons	5		9	
· Box			8	
Temperature Sensing Unit		**100**		**115**
· Pipe	5		5	
· Temperature sensors	85		89	
· Thermometer circuit	10		21	
Power Supply		**30**		**10**
Solenoid Stop Valve		**135**		**130**
Microcontroller		**50**		**25**
Piping		**50**		**61**
Miscellaneous		**55**		**100**
Total Cost		**$671**		**$722**

In the cost estimation, we did not foresee numerous shipping charges and costs for placing our circuits neatly in boxes. The effects of our oversight caused the cost overrun for the DC geared motors, the user input and the miscellaneous costs. We did not account for the cost of connectors and basic electronic components (such as buffers and NAND gates), which ballooned the cost for miscellaneous items. For our choice of plastic piping and valves, we required connectors to resize fittings to the appropriate diameter. Although our costs have surpassed our estimations, the discrepancy is less than 10%.

4.2. Schedule

Figure 7 presents the Gantt chart of the original project schedule as given in the *Tempero Systems Project Proposal for TemperSure*.

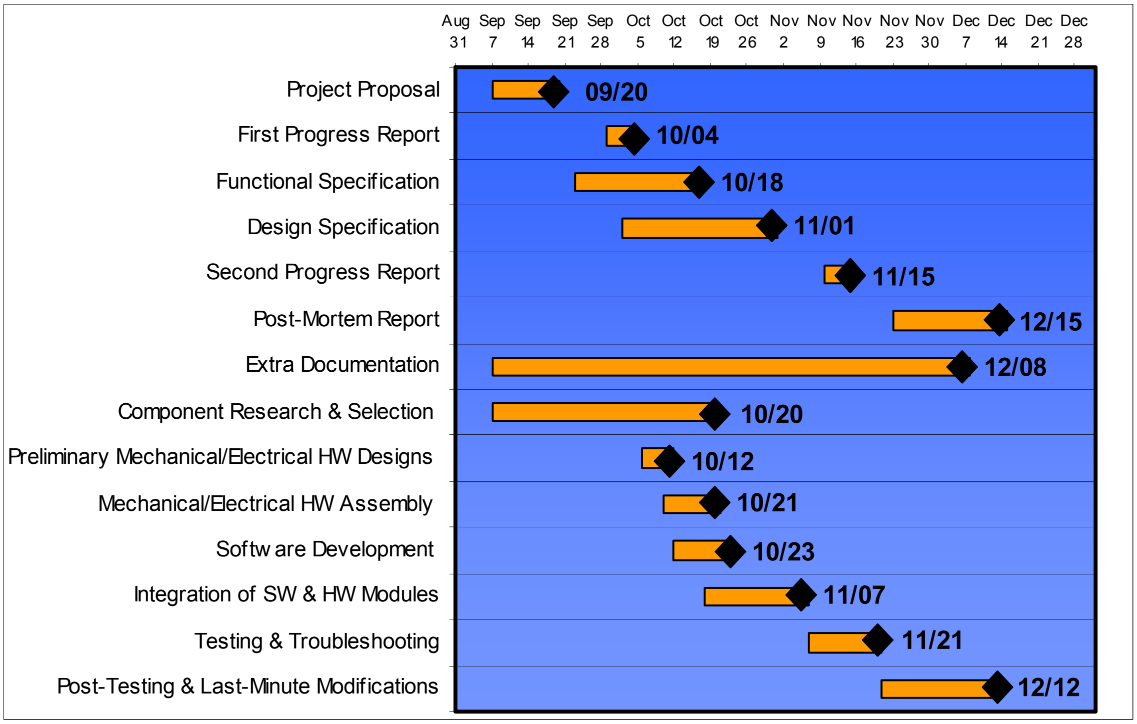

Figure 7: Gantt Chart of Project Deliverables and Major Milestones for Proposed Schedule

We intended to complete the proof-of-concept system in late November 2004 and provide the presentation in mid-December 2004.

Figure 8 presents a Gantt chart of the actual project schedule followed.

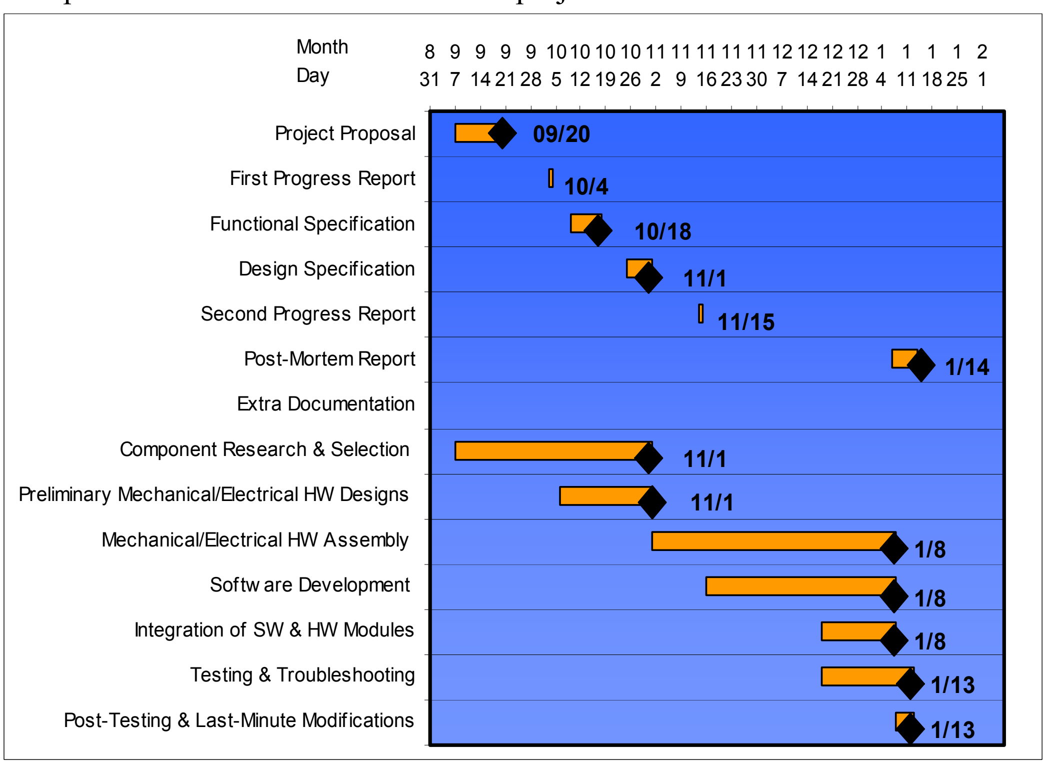

Figure 8: Gantt Chart of Project Deliverables and Major Milestones for Actual Schedule

Comparing Figure 7 to Figure 8, the time and effort per set of tasks was compressed, and milestones were delayed because we underestimated the difficulty of electromechanical construction and the time commitment involved with course work. The integration of software and hardware modules, mechanical/electrical hardware assembly, and testing activities were performed until the end of the project. The software-hardware integration relied heavily on the implementation of the motorized valve assembly and required more effort than initially allocated. Consequently, the testing and troubleshooting period was also extended. Due to time constraints, our team decided to opt out of composing extra documentation.

5. Future Plans and Recommendations

From the design and development process of our system, we determined our system could be improved and that future plans could be undertaken to achieve more of our functional requirements.

5.1. Temperature-Sensing Unit

The temperature-sensing/anti-scalding unit can be further improved in the future by constructing it as a separate unit. A lower power consuming solenoid valve can be used in conjunction with PCB and surface board devices to reduce the total power consumption of the unit. With a reduction in power consumption, the unit can then be battery powered and it can then be developed as an independent unit, which further increases the marketing potential.

5.2. Valve Position Sensing Unit

In the future, we believe that we can make the sensor unit more accurate and more precise. Currently, the infrared emitter has a viewing angle of approximately 20°. This large angle will allow the receiver to turn on even if the emitter is not directly facing the receiver. The large viewing angle will decrease the operational range of our motors because the motors will stop before the motors have actually reached its limiting point.

5.3. Valve/Motor Control Unit

A future system could include medical valves with a more linear water flow versus valve position relationship to provide for a more consistent water temperature change under firmware control. However, a compromise needs to be determined between consistency and cost. Further research and development of the medical valves needs to be performed before applying them to our system. Consequently, a change in valve type would also require the design of a new mechanical coupler.

5.4. Temperature Control Unit

In the future, we plan to develop more complex algorithms for changing and maintaining the temperature. With a more complex algorithm, the temperature can be maintained more accurately and quickly. With a complex algorithm, we feel that the current microcontroller may be inadequate to handle the tasks. A more capable microcontroller, such as the Motorola HC12, may be needed. Also, we also consider switching to programming with high level programming languages such as C or C++. Assembly was used throughout the development phase. With high level languages, we believe we can shorten development and debugging time because of the familiarity and ease of use we have with high level languages.

5.5. User Interface Unit

A few functional enhancements for the user interface unit could be included in the next-generation system. The casing of the unit could be reduced while increasing the display area and the size of the push buttons. This enhancement would assist users, especially the elderly and young children with more accessible control and feedback. In addition, a preset/memory feature could be added that provides the user with his or her favourite water temperature settings. Such a feature would reduce the need for the user to increment/decrement the desired temperature by 1 degree when a temperature change of 20 degrees is desired. Alternatively, a numerical keypad allowing the user to enter the desired temperature directly could be incorporated in the next version of the *TemperSure* system.

6. Interpersonal and Technical Experiences

6.1. Stephen Au-Yeung

The past 5 months has been very frustrating yet a very rewarding experience for me. I have learned a lot in project management such as task scheduling, planning while I continue to gain unique and valuable experience both in expanding my technical knowledge and strengthening my interpersonal skills. In addition, I also got a chance to work in the business aspects of the project such as developing a financing strategy and researching on the marketing potential of our final product.

I quickly realized everyone has a different method and approaches to different aspects of the project. Fortunately, healthy discussions and frequent communication between each member of the group allow us to cope with the differences quickly. Regular team meeting allows members of the group to voice their opinion and concerns which improves team dynamics as the semester goes on.

The technical experience I gained is invaluable. I am now very familiar with water system and gain unforgettable plumbing expertise. I would like to thank Chao Cheng for providing this amazing idea and my group members for a fun, rewarding project experience.

6.2. Francis Merin

I found the project to be an extremely trying yet necessary learning experience. Many of the lessons and experiences learned from previous Engineering Science courses seemed to be magnified exponentially during the course of this semester.

My main contribution to the project was developing firmware for the user interface, particularly for the push buttons and display components, developing motor limiting logic with Victor Tai, and coordinating project documentation. Despite having not programmed in assembly language since ENSC 151 and having chosen an unfamiliar microcontroller, given a couple weeks of research, long nights in the lab, and an in-circuit debugger, Victor Tai and I managed to develop functional firmware for the *TemperSure* system.

If I were to undertake a project of similar scale, I would have not taken the full 17 credits for this semester. This would have allowed me to become more involved in the design of the hardware and plumbing aspects of the project. Through this project, I have learned to better cope with stress, deadlines, and unexpected issues, or at least become more aware of them. In addition, I have realized that being less ambitious and more realistic helps when it comes to task scheduling. I also learned that effective group work is achieved not by taking ownership of particular tasks but rather by sharing them with the overall goal of solving the problems at hand.

Regardless of group size and differences in opinion, I learned that effective group dynamics are fostered through respect and communication. Most often than not, one cannot properly resolve any issue, be it hardware, software, or team related, without a cool head and an honest look at the situation. I have been fortunate to work with such a determined and talented group and look forward to working with them in the future.

6.3. Victor Tai

During this project, I have learned several aspects of project planning, research and time management. When the project first started, I was initially assigned to the hardware components of the team. During that time, I helped with designing the valve sensor circuit. I have expected to develop more of the hardware components, but one of our team members decided to leave the group, therefore I migrated to the software development team, with Francis Merin. As a Computer Engineer, it was natural that I be assigned to the software development group as I have experience in programming and software design. At this point, much of the real work began. Because we worked with a Microchip PIC controller, it was a completely different architecture than the Motorola HC12, which I have previously worked on. Although some assembly instructions are similar, the programming model is completely new to me. I have forgotten many assembly programming knowledge as my last assembly programming course was almost 3 years ago. During this phase, I developed the temperature changing code and assisted with the sensor limit codes. Although I would have preferred to develop in a high level language, such as C/C++, to reduce development time, the experience though programming in assembly gives me a refresher in assembly programming.

One of the most frustrating parts of the project was the debugging of the software program. As the debugger, the Microchip ICD2 module, was limited in speed and the number of breakpoints, much of the time spent debugging was guessing where the problem may be and waiting for the ICD2 to finish a group of instructions. There were also cases where a wire to an essential part of the circuit was loose, leading to a malfunction in the software program.

Throughout the project, each group member has had a difference of opinion towards the project. Through calm discussion, we have been able to find a common ground and proceed smoothly with the project. Although many parts of the project were delayed and we had to work very hard during the last several weeks, we managed to complete the project with a working model in time for the demonstration.

6.4. Jason Wong

The project course, Ensc 340, has the reputation of being a challenging course and it lives up to its reputation. To complete the course and project without much emotional and physical damage, I relayed to my group to work hard and steady while staying away from late nights. I have found that late nights mean that everyone shows up to work later the next day. A guideline I adopted is called the *12 hour rule*. If we are to be at school the following day at a specified time, then we would all leave school 12 hours before that specified time. Keeping everyone well rested increased endurance and productivity and decreased human error.

Leading my team though the many stages of the project involved coordination. As a group, we decided what needed to be done and I would make sure tasks were completed. I made sure everyone had a chance to voice their opinion regarding every aspect of the project. With such a group environment, I believe everyone felt that the project was a group effort and not a work environment where they are being told what to do. Each group member had their area of expertise. No one person was the one in control.

As the project leader, my vision of how the working model should behave and look never faltered but I had difficulty keeping my motivation high. It did not help that we were working though the holidays. Luckily, I had an excellent group. Whenever one person's motivation would fall, someone would pick up the slack and push everyone else to work. We had great team dynamics. Although most of us had never worked with each other on any project or labs, I found that we got along very well and became friends in the process. Because of the team members I worked with, developing a model of the *TemperSure* system was enjoyable at the best of times and bearable at the worst of times.

7. Conclusion

The past semester has been a challenging yet rewarding experience for the Tempero Systems team. We applied most of the knowledge and skills acquired in our engineering careers to date, as well as patience and determination, to research, design, implement, test, and document hardware and software modules to achieve a working temperature control system for the shower.

8. References

[1] National Semiconductor. 2004. *Photodiode Amplifiers – Turning Light into Electricity*. 2005 Jan 14
< http://www.national.com/onlineseminar/2004/photodiode/PhotodiodeAmplifers.pdf>.

Appendix A: Flow Charts of Temperature Controller Firmware

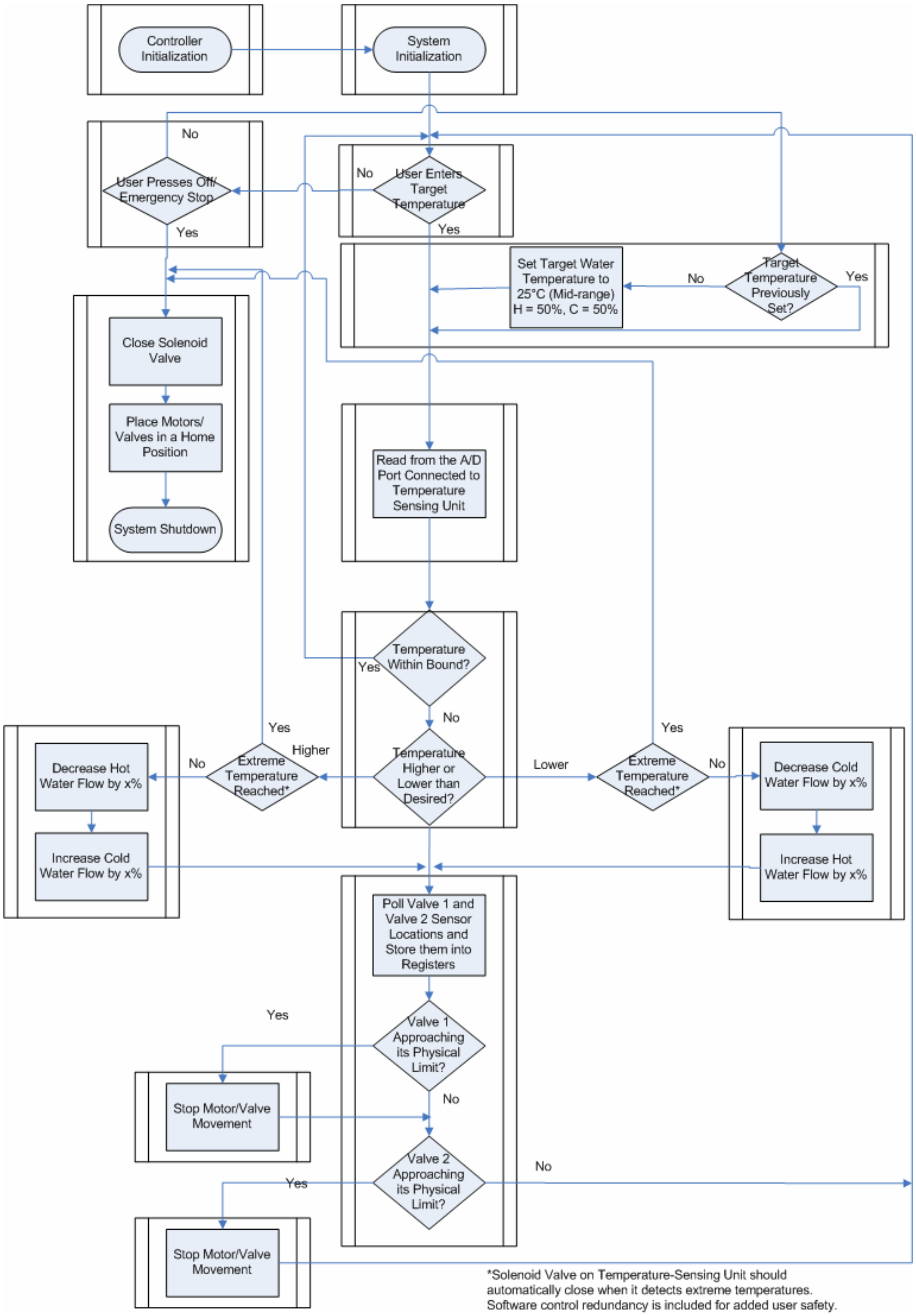

Figure 9: Original Flow Chart for Temperature Control Firmware

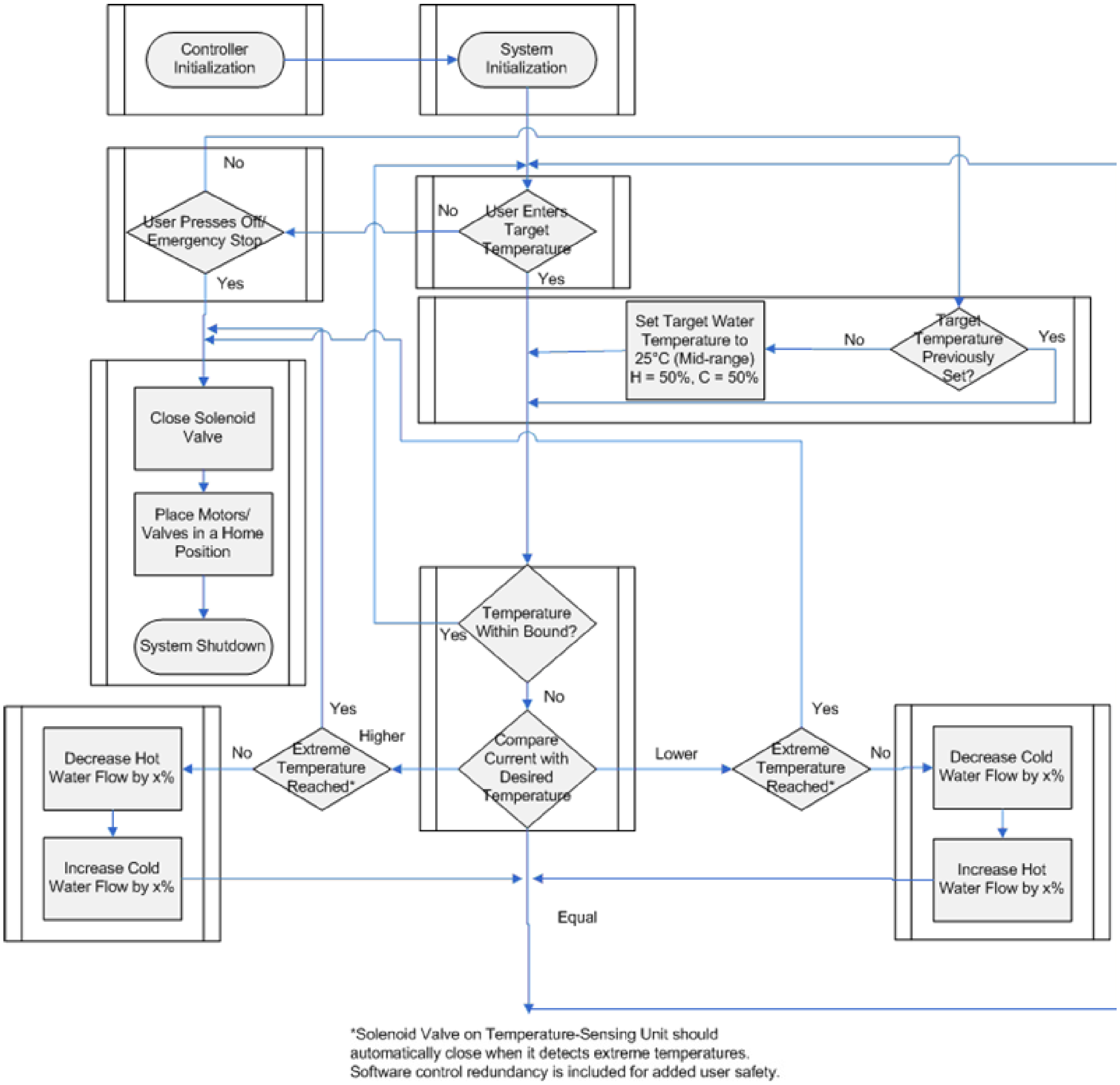

Figure 10: Modified Flow Chart of Temperature Controller Firmware without Periodic A/D Conversion and Motor Limit Checking.

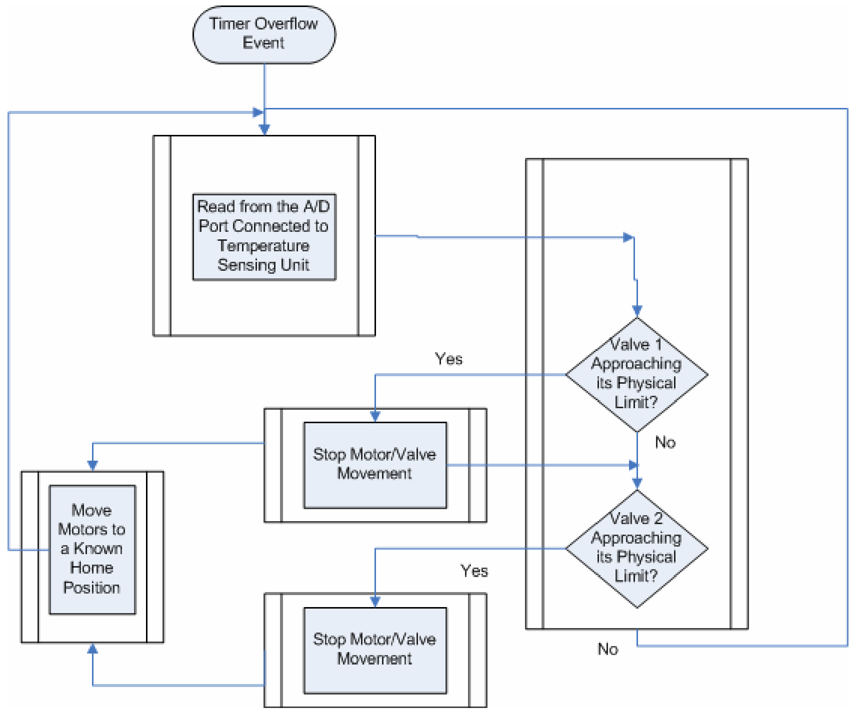

Figure 11: Flow Chart of Actual Periodic A/D Conversion and Motor Limit Checking