TemperoSystems take control.

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

November 1, 2004

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

Re: ENSC 340 Design Specification for *TemperSure*

Dear Dr. Rawicz,

Please find an attached copy of the Tempero *TemperSure* design specifications. Our design specification provides a set of detailed technical descriptions for system design and development of our preliminary model. The *TemperSure* model will be able to provide a constant shower temperature for varying input water levels and provide an emergency stop to showers that reach scalding temperatures.

The following design specifications are only for the proof-of-concept model. Therefore, we will only discuss the functional requirements pertaining to proof-of-concept and both proof-of-concept and final production models.

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 Design Specification for *TemperSure*

# Tempero Systems

take control.

## Design Specification for a Shower Temperature Control System


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

**Contact personnal:** Jason Wong
jwongk@sfu.ca

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


**Date:** November 1 st , 2004

# Abstract

The Tempero *TemperSure* design specification provides a set of detailed technical descriptions for system design and development of our preliminary proof-of-concept model. Since the design specifications are solely for the proof-of-concept model, we will only discuss the functional requirements pertaining to proof-of-concept and both proof-of-concept and final production models.

We will be discussing the design of *TemperSure* and justifying our design choices against other designs we considered. The temperature of the water will be captured by a thermocouple interfaced to the A/D of a PIC18F452. The mixture of incoming water will be controlled by a set of DC motors attached to valves. The positioning of the valves will be captured by infrared sensors and emitters placed strategically around the valve. The user will interface to the unit through a panel composed of an LED display and push buttons. If at any time, the shower water enters into the scalding temperature range, a solenoid valve will immediately end the shower. The water of the system will be routed when possible by PEX piping and minimal copper piping. We will include a detailed section on the microcontroller detailing the use of resources and general program flow. The algorithms for controlling the water temperature will also be included.

Each section will detail a test plan for the particular unit. Once all tests have passed, and the individual parts are deemed to be working we will integrate the system and perform system tests.

# Table of Contents

## List of Figures

# List of Tables



# Glossary

| | |
|---|---|
| AC | Alternating Current |
| A/D | Analog-to-Digital |
| DC | Direct Current |
| GPM | Gallons per Minute |
| K-Type | Thermocouple Composite with Nickel Chromium and Nickel Aluminum Alloy |
| PEX | Cross-linked Polyethylene |
| PID | Proportional-Integral-Derivative |
| Response time | Time elapsed to attain 63% of the final temperature |
| *TemperSure* | Shower Temperature Control System by Tempero Systems |

# 1. Introduction

Tempero Systems envision designing, developing, and testing an electronic temperature control system for showers/baths. The goals of our project are 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 also aim 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 provides technical details on the design of each component. Both hardware and software components will be discussed in detail.

## *1.1. Scope*

This document is the design specification that meets the requirements outlined in *Functional Specification for a Shower Temperature Control System*. The design specification includes all requirements for a proof-of-concept system and a partial set of requirements for a production device. The appendices of this document include schematics and test plans to facilitate development of the *TemperSure* System.

## *1.2. Intended Audience*

This document is intended for use by all members of Tempero Systems. Design engineers shall refer to these specifications as overall design goals to ensure all requirements are met in the final product. Test engineers shall use this document to implement their test plan and confirm the behaviour of this product.

# 2. System Specifications

*TemperSure* will be able to provide a range of temperature between 5 to 55°C. Above 55 °C the anti-scalding feature will activate and the shower will stop. The justification for the 55°C maximum temperature is discussed in Section 3.

Ultimately *TemperSure* is limited by the hot and cold water input from the shower piping system. If the incoming water cannot be combined to output the desired temperature, the shower will continue until either the user changes their desired water temperature or the incoming water temperature change. If, in the previous case, the output water temperature reaches a scalding 55°C, the anti-scalding device will stop the shower.

Since 5°C is the typical household cold water temperature, we chose 5°C to be our lower bound.

The system response for our system is defined as response with the time between user input and motor/valve activation less than 1 s. An accurate system response is a 1°C difference between the actual temperature and the desired temperature set by the user.

### 2.1. Power Supply

We have chosen to use a computer power supply from ASTEC (AA16320) since it was made available to us at no cost. The AA16320 is an 85 Watt power supply which has a ±12 and ±5 VDC terminal. In addition, it also provides us with abundance of current (2.5A from +12, 10.00A from +5, 0.5A for –12 and -5) which is more than enough for our system.

## 3. Temperature-Sensing Unit

### 3.1. Overview

The three main functions of the temperature sensor unit are to:

1) Provide a constant temperature feedback to the central microprocessor unit.
2) Accept command for closing the solenoid valve from the microprocessor in the case of emergency.
3) Act as an anti-scald device that will turn off water supply when water temperature rises over 55°C.

The main components of the unit contain a K-Type precision thermocouple, a thermocouple amplifier, and a solenoid valve for shutting off water supply.

The design and the construction of the temperature sensor is the most crucial part of the system. Without an accurate measurement of the water temperature, the whole system will not function properly.

### 3.2. Physical Design

The proposed design solution is shown in Figure 1.

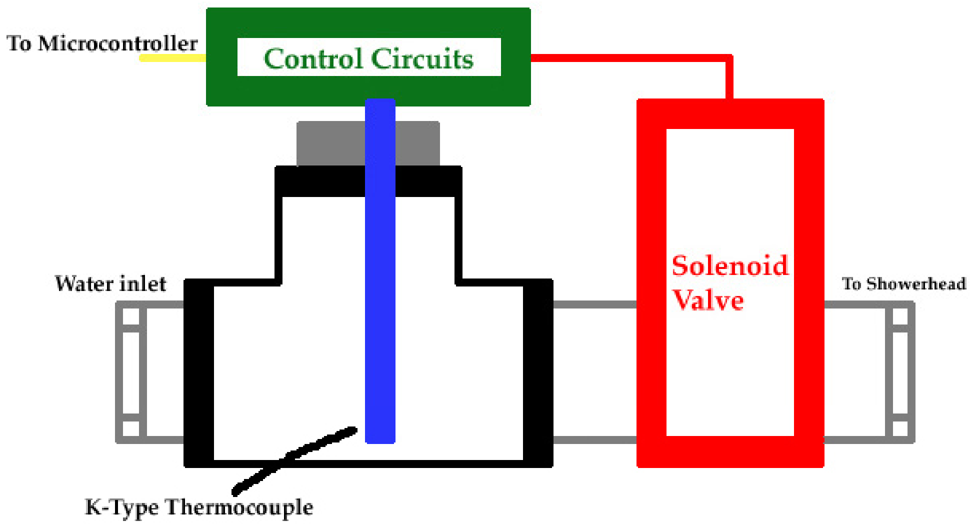


**Figure 1: High level diagram of the Temperature Sensing Unit.**

The whole temperature sensing unit is to be installed as an attachment before the showerhead. It is implemented as an isolated unit in order to cut down its dependence on the rest of the system (mainly the microprocessor). The anti-scalding function is implemented independently from the rest of the system, which allows for future opportunity to develop the unit as a portable anti-scalding device.

As illustrated in Figure 1, the whole unit contains a solenoid valve, a control circuit and a thermocouple wire. A thermocouple wire is a sensor for measuring temperature. It consists of two dissimilar metals, which produce a small unique voltage for given temperature. The voltage is then converted into temperature readings in the control circuit.

The thermocouple wire is thread through a pre-drilled hole in the water stopper. The water stopper is then screwed on tightly to one of the openings in the T-pipe, the outlined black object seen in Figure 1, which provides a stable housing of the thermocouple.

## 3.3. Thermocouple Specifications

In order to insure that water stops flowing within one second after a scalding temperature is detected, a thermocouple with fast response time is needed to provide a quick and accurate feedback to the system. To meet the performance requirement and also provide easy prototyping, we’ve chosen to use a K-Type 36 AWG thermocouple from Omega Engineering Ltd. which has a response time of 0.04 seconds. A K-Type thermocouple is used due to its durability and its wide range of measurement temperature.

## 3.4. Thermocouple Amplifier Specifications

We used a thermocouple amplifier from Analog Devices (AD595) in conjunction with the thermocouple. The AD595 IC combines an ice point reference with a pre-calibrated amplifier to produce a high level (10 mV/°C) output directly from a thermocouple signal, which will dramatically simplify the process of translating the thermocouple output voltage into temperature in degrees Celsius [1].

## 3.5. Solenoid Valves Specifications

A 12 VDC solenoid valve (8210G1) from ASCO Ltd. is used for this design to provide a quick and fast shut off of the water supply in the case of emergency. Due to tight budget constraints, a normally closed valve is chosen instead of a normally open valve. A normally closed valve means that the valve is in the off position until power is applied, and the converse with a normally open valve.

## 3.6. Electronic Design

The 5VDC powered temperature control circuit is the main block that interacts with the rest of the system by accepting commands and providing temperature feedback to the microprocessor. Figure 2 is a schematic of the control circuit.

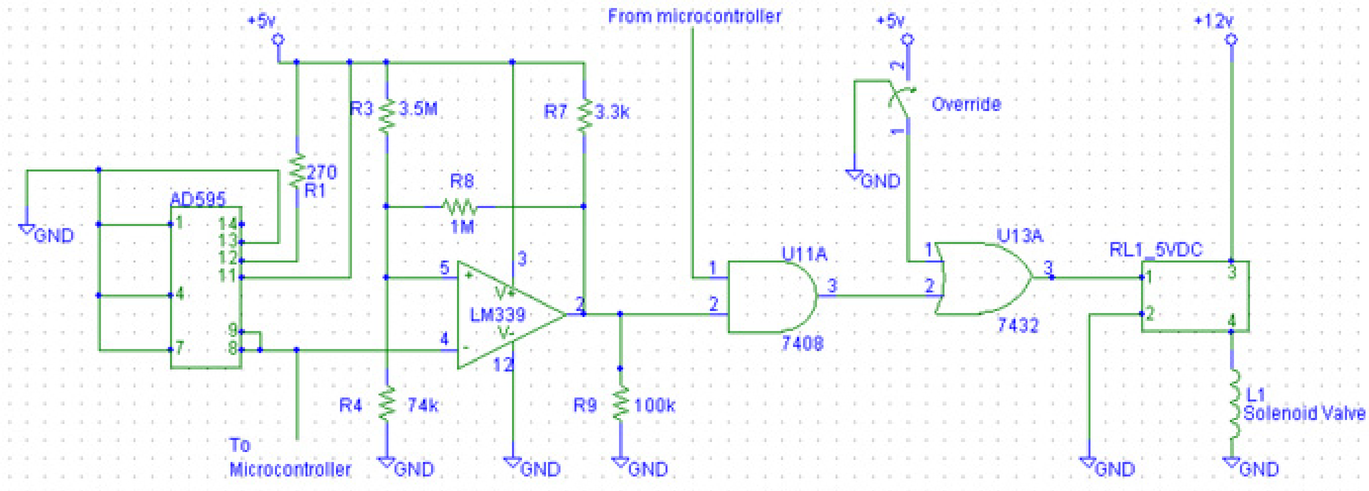


**Figure 2: Schematic of the temperature control circuit.**

The AD595 provides output on the water temperature from the thermocouple through pin 9. The information is relayed to the microcontroller and into a LM339 comparator implemented as an inverting comparator with hysteresis. The upper threshold voltage represents the high cut-off temperature voltage. We have decided to set this value to 550mV (thus 55°C) because according to research from EnergyGuide, a person will sustain a serious burn when exposed to water with a temperature of 55°C for 30 seconds [4].

In order to implement a hysteresis characteristic, a low threshold voltage will also need to be defined. To assure the solenoid valve will remain closed once the water temperature goes above the high cut-off limit; a very low threshold voltage is used. We have decided to set this voltage to around 100mV (thus 10°C) in order to prevent the hysteresis from switching on again; thus reopening the solenoid valve. By implementing the hysteresis this way, we can almost assume that the only way to reopen the solenoid valve is through a reset of the whole system.

The LM339 comparator appears to be an ideal choice to implement such Schmitt Trigger[1] due to its low offset voltage (±3mV) and ability to operate in a wide range of supply voltages (2 to 36VDC). Figure 3 shows the Schmitt trigger from Figure 2 and Figure 4 illustrates its behavior.

[1] A Schmitt Trigger is another name for an amplifying device with hysteresis.

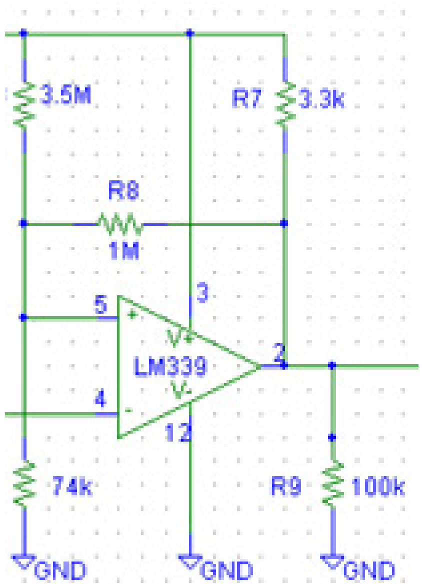


**Figure 3: The LM339 implemented as an comparator with hysteresis, also known as a Schmitt trigger.**

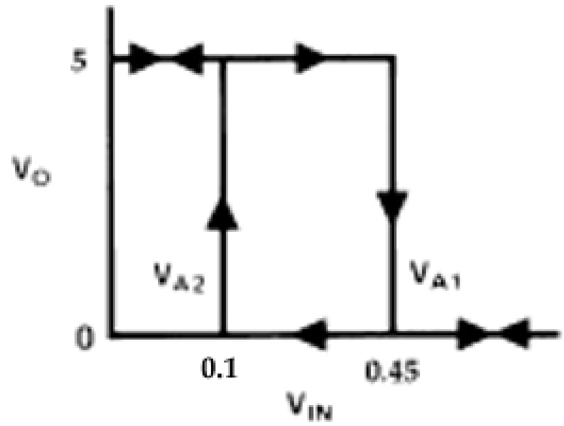


**Figure 4: The transfer characteristics of the LM339 Schmitt trigger.**

The threshold voltage of the Schmitt trigger can be made adjustable in the future to allow user defined cut-off temperature. A 0-20k potentiometer in series with a 60k resistor can be used to replace R4 (74k resistor in Figure 3) to provide user adjustable cut-off temperature range from 48-55°C. A minimum of 48°C is chosen because household with small children are recommended to have a hot water temperature of 48°C [4].

The output of the Schmitt trigger in Figure 2 is then fed into an AND gate so that the microcontroller can force the solenoid valve to close when an emergency button is pressed. The output of the AND gate is ORed with the manual override button in order to implement the system override feature. Once activated, the override feature allows a constant flow of water regardless of the microcontroller's command to initiate the emergency stop for scalding water temperatures. Hence, the user can manually turn off the anti-scalding feature; however, when the temperature unit is used along with the rest of the *Tempersure* system, the maximum temperature is still bounded by the predefined range of the system.

Finally, the output of the OR gate is used to turn on/off of a relay, which is used to initiate the solenoid valve. A relay will isolate the solenoid valve, powered at 12V, from the rest of the circuit, powered at 5V.

# 4. Valve Position Sensing Unit

To accurately determine the current position of the motor unit, a set of photo emitter/ photodiode pairs are used. Figure 5 shows the circuit diagram for one pair of photo emitter and photodiode pair.

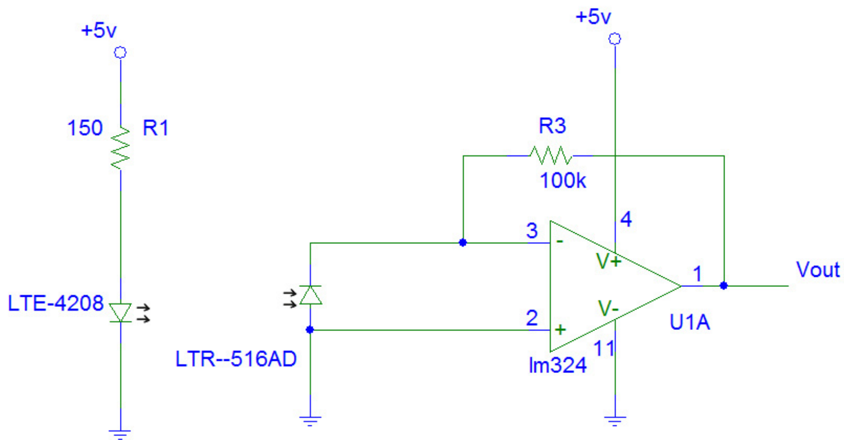


**Figure 5: Circuit diagram for one photo emitter/photodiode pair.**

## *4.1. Optical Sensor Specifications*

The photo emitter used in the above circuit is the LTE-4208 manufactured by Lite-On Electronics Inc. This device's specifications are shown in Table 1. The photodiode used is the fast response LTR-516AD, which is also manufactured by Lite-On Electronics Inc. The photodiode receives light and translates it into a current flowing from the cathode to the anode of the diode. The current flow is opposite of a normal diode, which current flows from the anode and to the cathode. The device parameters of the LTR-516AD are shown in Table 2.

**Table 1: Device parameters of the LTE-4208.**

| Parameter | Value(typical) | Units |
|---|---|---|
| Peak Emission Wavelength | 940 | nm |
| Forward Voltage | 1.6 | V |
| Viewing Angle | 20 | degree |
| Reverse Current | 100 | µA |

**Table 2: Device parameters of the LTR-516AD.**

| Parameter | Value | Units |
|---|---|---|
| Reverse Dark Current | 30 | nA |
| Rise Time | 50 | nsec |
| Fall Time | 50 | nsec |
| Wavelength of Max Sensitivity | 900 | nm |

Comparing Table 1 and Table 2, it is evident that the LTE-4208's peak wavelength does not match the LTR-516AD. However, since the LTR-516AD is able to attain 85% of its sensitivity at 940 nm, it will not be a problem for the circuit. As noted in Table 1, the wavelength that the circuit is operating on is 940 nm. This wavelength is chosen as it provides good immunity against environmental infrared sources. A potential infrared source is the human body. The human body radiates heat because of body temperature, which translates to infrared radiation with wavelength of approximately 9 µm.

Another potential infrared source is the hot water pipe in the system. Using Wien's Displacement Law[13] and assuming a water temperature of 80°C, it can be shown that the infrared radiation that radiates from the water pipe has a wavelength of approximately 8 µm. With the circuit operating at 940 nm, interference from other sources of infrared radiation can be reduced.

The operational amplifier LM324, shown in Figure 5, serves as an amplifier to increase the voltage detected from the photodiode. The feedback resistor, R3, provides gain control and also dictates the sensitivity of the photodiode. A bigger feedback resistor increases the sensitivity of the receiver because it allows amplification of smaller signals. The increase in sensitivity allows a longer range between the photo emitter and the photodiode. The trade-off, however, is that the photodiode can be overly sensitive, leading to incorrect detection. As a balance between range and sensitivity, a 100KΩ resistor gives a detector range of approximately 6 cm and viewing angle of approximately 20°.

To interface with the microcontroller, the LM324 will amplify the signal to 4V, giving standard logic levels to the microcontroller. When a photo emitter shines light onto a photodiode, the sensor is triggered and will generated 4V at the output of that particular photodiode circuit. Each photodiode will be connected to its own microcontroller I/O pin, allowing the microcontroller to determine the triggering photodiode. Table 3 summarizes the outputs of the valve sensor system.

**Table 3: Outputs of the valve sensor unit.**

| Photodiode Status | Output Voltage |
|---|---|
| Illuminated | 4V |
| Not illuminated | 0V |

## *4.2. Mounting Sensor Elements*

As part of the motor/valve assembly, the photo emitters will be placed on a spinning disk along the motor motor/valve coupler. As seen in Figure 6, when the photo emitter passes over a photodiode, the corresponding photodiode will be triggered. The wire connection to power the photo emitter will come from the motor side (from the right side in Figure 6). Since our motor rotates only 1.5 turns, the wire will not tangle with the motor and motor shaft.

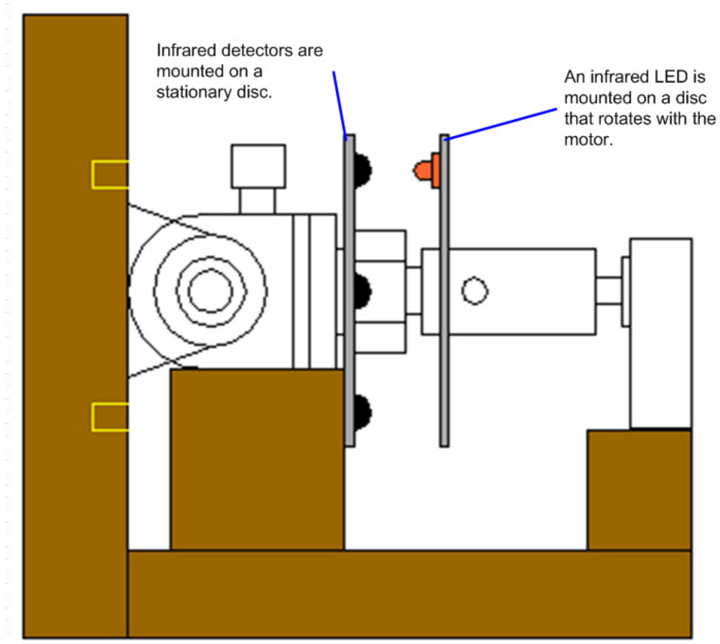


**Figure 6: Sketch of Sensor Mounting**

### *4.3. Sensor Verification*

The sensor unit requires testing in two phases. Phase 1 of testing is performed with the sensor unit not mounted onto the final motor/valve assembly.

1. Verify the ability of the photodiode to detect infrared light from the photo emitter. The range of the photodiode must be at least 6 cm and the view angle cannot exceed 10° on both sides of the photodiode.
2. Verify the output voltage is 4 V when the photo emitter is shining on the photodiode and 0 V otherwise.
3. Verify the response time of the photodiode be at most 50 µs. This test involves ensuring the output will change from logic low to logic high or vice versa within 50 µs.

All Phase 1 tests should be performed before continuing on to Phase 2.
During phase 2 testing, the sensor unit will be mounted onto the motor/valve assembly. Phase 2 testing involves testing the sensor unit in a environment that will be the same as the unit's operating environment. The tests in Phase 2 will include all the tests done in Phase 1. In addition, the following tests will also be performed:

1. Verify the operation of the photo emitter power wire to ensure the wire does not interfere with the operation of the motor and valve.
2. During a full open and close sweep of the valve, verify all photodiode states are reached.
3. When the motor movement passes a sensor, we must be able to stop with motor within 5° of passing the sensor.
4. When a photo emitter passes the photodiode, ensure the output is at 4V when the photo emitter is within ±10° of the photodiode. At any other position, the output should remain at 0V.

# 5. Valve/Motor Control Unit

The valve/motor control unit is responsible for controlling the mixture of hot and cold water for the *TemperSure* system. The unit is composed of a DC motor attached to a half inch valve via a coupler and an H-Bridge. The H-Bridge takes inputs from the microcontroller and controls the movement and direction of movement of each motor accordingly.

Other designs for the unit were considered such as using stepper motors, DC motors with built in encoders, and motorized valves . Motorized valves and DC motors with built in encoders would have increased our budget at least two fold. Stepper motors, although costly, are unreliable in keeping track of the valve positions. Stepper motor will skip a step if the torque becomes too great for the motor, thereby, introducing an error in the valve positions. A diagram of the plunger valve, coupler, and motor integration is shown in Figure 15 in Appendix A.

## *5.1. Plunger Valve Specifications*

We are using CSA approved half inch stop and waste plunger valves from Waterline Products. The valves we are using are fitted for PEX piping. Our justification for using PEX piping can be found in Section 7. The particular valve chosen was competitively priced and readily available at Home Depot, which avoids the cost for shipping. The piping in homes have an outer diameter of a half inch, so, we used a half inch valve for compatibility purposes. A schematic of the valve used is shown in Figure 13 in Appendix A.

Through experimental tests, we found the torque to operate the valve to be 3.5Ncm or 4.95oz-in and 1.5 turns is required to turn the valve from an open to a close position (without entering the valve's extremities in both cases). Consequently, we set out to find motors with at least twice the torque we required.

## *5.2. Geared DC Motor Specifications*

To narrow down the selection of DC motors, we concluded that we needed a geared and reversible DC motor. Geared motors reduce the speed of the motor but increase the torque. A reversible motor is required to turn the valve in both directions. We further bound our search by the required torque we needed and by the limitations of our power supply to deliver the necessary current and voltage to the motor.

We decided on a set of Globe motors which operates at 110RPM at 24VDC and provides 30oz-in. of torque at 330mA. The consideration for the speed of the motor is the most difficult. With a faster motor, we gain response time, but we lose accuracy in motor control. We decided a speed of 55RPM at 12VDC will give us a good response. With the power supply we are using, our motor, at 330mA, gives us the option of using either the positive or negative 12V supply. The Globe motors we used were discounted and less costly when compared to equivalent motors. A disadvantage of using the particular model of Globe motors is their bulkier size (irrelevant to the construction of the model). A schematic of the motor used in shown in Figure 12 in Appendix A.

## 5.3. Mechanical Coupler Specifications

There are many methods to transfer the torque of a motor shaft to the shaft of a valve. Our solution is simply to use a hollowed metal bar with a hole at each end for the shafts. To secure the motor and valve shaft, we used set screws. A schematic of the coupler used is shown in Figure 14 in Appendix A.

## 5.4. The H-Bridge

The H-Bridge is a common circuit design used to reverse the polarity of a voltage supply to DC motors. Once the polarity is reversed, the connected DC motor can be rotated in the opposite direction. The H-Bridge circuit allows a microcontroller to easily control the movement and direction of a DC motor. Figure 7 shows the H-Bridge circuit diagram. The MOSFETs are connected to the 12V supply to meet the motor voltage requirements.

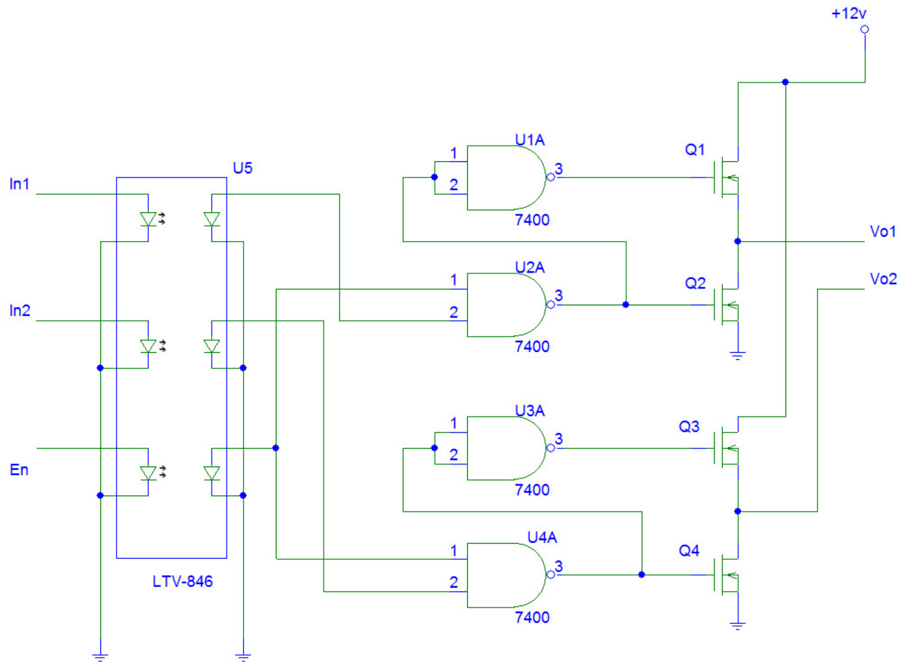


**Figure 7: H-Bridge circuit diagram.**

The implementation of the H-Bridge design in Figure 7 is similar to many other existing designs. Following common designs, an optoisolater is used to protect the output pins of the microcontroller from any destructive feedback. The optoisolater used is a LTV-846, manufactured by Lite-On Electronics Inc. The LTV-846 is a four channel device, capable of providing protection for up to four signals. The device is capable of providing an isolation voltage of 5000 Vrms and a current transfer ratio of at least 50%. Some important parameters of the LTV-846 are summarized in Table 4.

**Table 4: Device parameters of the LTV-846**

| Parameter Name | Value | Units |
|---|---|---|
| Isolation Voltage | 5000 | Vrms |
| Input Reverse Current | 10 | µA |
| Current Transfer Ratio | 50~600 | % |
| Isolation Resistance | 10^11 | Ω |
| Response Time(rise) | 4 | µs |
| Response Time(fall) | 3 | µs |

The NAND gates' primary function is to allow the use of an enable signal and also to invert the output to provide an inverted signal to the MOSFETs. Although a logic inverter can be used, it will increase the chip count of the system and also add additional delays. A typical NAND gate integrated chip contains four NAND gates in a package; therefore the H-Bridge will only need one chip.

The MOSFET in the H-Bridge are the switches that control the polarity of the output voltage. The MOSFETs will be required to deliver significant amount of current to the DC motor, so the N-Channel BUZ73s manufactured by Infineon Technologies are used. Several parameters of the BUZ73 is listed in Table 5.

**Table 5: Device parameters of the BUZ73**

| Parameter Name | Value | Units |
|---|---|---|
| Maximum Drain current | 7 | A |
| Maximum $V_{ds}$ | 200 | V |
| Maximum $V_{gs}$ | 20 | V |
| Response Time(rise) | 40 | ns |
| Response Time(fall) | 30 | ns |

The H-Bridge uses three inputs and two outputs for its operation. The digital inputs are In1, In2, and En. The outputs are Vo1 and Vo2. Vo1 and Vo2 are connected to the DC motor and it is these two outputs' polarity will be varied by the H-Bridge. The operation of the H-Bridge is summarized in Table 6.

**Table 6: Summary of H-Bridge operation. X denotes an unknown value.**

| In1 | In2 | En | Vo1 (V) | Vo2 (V) |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 12 |
| 1 | 0 | 1 | 12 | 0 |
| 1 | 1 | 1 | X | X |

Table 6 shows that the inputs In1 and In2 cannot be at logic high concurrently. If both inputs are at logic high, unexpected behaviour will result. Therefore, it is required that the microcontroller does not drive the logic inputs high at the same time.

### 5.5. Valve/Motor Control Verification

The H-Bridge requires the following tests to be performed:

1. Verify correct operation by supplying the inputs and observing the outputs, as depicted in Table 6 with the exception of the last case.
2. Verify the ability of the H-Bridge to provide enough current for the DC motor
3. Verify response time of the H-Bridge to be at most 10 µs so that it does not limit the operation of the motors.

The whole unit can be tested by:

1. Supplying inputs from Table 6 and observing the correct movement and direction of the correct motor.
2. Assure that mechanical coupler is still firmly securing both shafts after motor movement.

## 6. Temperature Control Unit

### 6.1. Overview

The temperature control unit is the central controller for the *TemperSure* control system. The unit consists of the Microchip PIC18F452 high-performance, enhanced flash microcontroller with 10-bit A/D as the primary hardware element. Control software written in Microchip assembly language provides the control logic used to synchronize, monitor, and activate all other system units.

### 6.2. Microchip PIC18F452 Microcontroller

#### 6.2.1. Selection Criteria

We considered many options during the *TemperSure* microcontroller selection process. The criteria applied for microcontroller selection included:

1. Cost
2. Memory Quantity
3. Input-Output (I/O) Pin Quantity
4. Interrupt Capabilities
5. External Circuitry
6. Development Learning Curve

Based on our prior course and project experience, the Motorola M68HC12 was seen as a favourable option because of the reduced learning curve involved. Consequently, the basis of comparison for microprocessor capabilities also relied on our experience with the M68HC12. Based on experience, we concluded that the M68HC12, and similarly the M68HC11 would require a lot of external circuitry to construct an evaluation board.

Adding external circuitry would increase the component costs, the development learning curve, and the prototyping and troubleshooting time.

Following technology trends and the experience of past engineering teams led us to consider the PICmicro® family microcontrollers from Microchip.

Each of the PICmicro® microcontrollers that we considered contained:

- 20 MHz clock inputs for the 16F models and with 40 MHz clock inputs for the 18F models
- On-board analog-to-digital (A/D) converters
- Timers
- On-board serial and parallel communications capabilities
- Simple instruction set

The microcontrollers were also available in dual-inline packages (DIPs) for ease of prototyping with solderless breadboards and vector boards. For ease of software development, an integrated development environment (IDE), MPLAB IDE, and an assembler, MPASM, were both available for free from the Microchip web site.

We narrowed down our choices to the microcontrollers given in Table 7.

**Table 7: PICmicro(R) Microcontroller Comparison**

| Device Model | Cost (US Dollars)[2] | Flash Program Memory | Data Memory (RAM) | EEPROM Data Memory | I/O Pin Quantity | Interrupt Capabilities |
|---|---|---|---|---|---|---|
| PIC16F873 | $4.33 | 4K | 192 bytes | 128 bytes | 3 ports | Up to 13 sources |
| PIC16F877 | $5.11 | 8K | 368 bytes | 256 bytes | 5 ports | Up to 14 sources |
| PIC16F877A | $4.68 | 8K | 368 bytes | 256 bytes | 5 ports | Up to 15 sources |
| PIC18F452 | $5.70 | 32K | 1536 bytes | 256 bytes | 5 ports | Up to 18 sources |

### 6.2.2. Instruction Set

The PIC18F452 contains 55 16-bit core instructions that are backward compatible with the PIC16 and PIC17 instruction sets. A set of 16-bit enhanced instructions provide extended capabilities. The table lookup, stack, and multiplication instructions are the most notable of the 16-bit enhanced instructions. The instructions typically use as their operands the working register, W, as well as general-purpose registers, and special function registers. The general-purpose and special function registers are denoted by F in the instruction mnemonics. Operands may be accessed via access bank direct addressing, direct addressing via the bank select register (BSR), or indirect addressing via file select registers (FSRs).

[2] According to Microchip, "the budgetary price is representative" and "does not reflect final pricing". [8]

### 6.2.3. On-board Peripherals

The 10-bit A/D converter is used to convert the voltage output from the comparator of the temperature-sensing unit. The PIC18F452 contains four timers whose specifications are given in Table 8. The parallel port, serial port, and universal synchronous/asynchronous receiver/transmitter (USART) modules are left unused for our temperature control system application.

**Table 8: Timer Specifications**

| Timer Module | Function |
|---|---|
| Timer0 | 8-bit/16-bit timer/counter with 8-bit programmable prescaler |
| Timer1 | 16-bit timer/counter |
| Timer2 | 8-bit timer/counter with 8-bit period register (time-base for PWM) |
| Timer3 | 16-bit timer/counter |

### 6.2.4. I/O Allocation

I/O pins are to be allocated for interfacing with system units as given in Table 9 to Table 13.

PORTA contains analog inputs used by the on-board A/D converter. We choose to sample the output from the comparator of the temperature-sensing unit on PORTA bit 0.

**Table 9: PORTA I/O Allocation**

| Pin Name | Pin Number | Direction | Allocated Resource |
|---|---|---|---|
| RA0/AN0 | 2 | Analog Input | Thermocouple Amplifier Output from Temperature-sensing Unit |
| RA1/AN1 | 3 | Digital Output | Motor/Valve 1 Control: IN1 |
| RA2/AN2/VREF- | 4 | Analog Input | External Negative Reference Voltage |
| RA3/AN3/VREF+ | 5 | Analog Input | External Positive Reference Voltage |
| RA4/TOCKI | 6 | Digital Output | Motor/Valve 1 Control: IN2 |
| RA5/NOT(SS)/AN4/ LVDIN | 7 | Digital Output | Motor/Valve 2 Control: IN1 |
| OSC2/CLK0/RA6 | 14 | Digital Output | Motor/Valve 2Control: IN2 |

PORTB provides pins for external interrupt sources, and interrupt-on-change sources.

**Table 10: PORTB I/O Allocation**

| Pin Name | Pin Number | Direction | Allocated Resource |
|---|---|---|---|
| RB0/INT0 | 33 | Digital Input to GND | UNALLOCATED |

| | | | |
|---|---|---|---|
| RB1/INT1 | 34 | Digital Input to GND | UNALLOCATED |
| RB2/INT2 | 35 | Digital Input to GND | UNALLOCATED |
| RB3/CCP2 | 36 | Digital Output | Solenoid Valve On/Off Control |
| RB4 | 37 | Digital Input | Interrupt-on-change pin: Emergency Stop Push Button |
| RB5/PGM | 38 | Digital Input | Interrupt-on-change pin: Increase Temperature Push Button |
| RB6/PGC | 39 | Digital Input | Interrupt-on-change pin: Decrease Temperature Push Button |
| RB7/PGD | 40 | Digital Input | Interrupt-on-change pin: Solenoid Valve Status |

PORTC acts as the valve position status register. The valve position register is updated asynchronously by the control circuitry of the motor/valve unit.

**Table 11: PORTC I/O Allocation**

| Pin Name | Pin Number | Direction | Allocated Resource |
|---|---|---|---|
| RC0/T1OSO/T1CLI | 15 | Digital Input | Valve 1 Position Sensor 0 |
| RCI/T1OSI/CCP2 | 16 | Digital Input | Valve 1 Position Sensor 1 |
| RC2/CCP1 | 17 | Digital Input | Valve 1 Position Sensor 2 |
| RC3/SCK/SCL | 18 | Digital Input | Valve 1 Position Sensor 3 |
| RC4/SDI/SDA | 23 | Digital Input | Valve 2 Position Sensor 0 |
| RC5/SDO | 24 | Digital Input | Valve 2 Position Sensor 1 |
| RC6/TX/CK | 25 | Digital Input | Valve 2 Position Sensor 2 |
| RC7/RX/DT | 26 | Digital Input | Valve 2 Position Sensor 3 |

PORTD has been allocated as the address and data register for use with the ICM7218D seven segment display driver.

**Table 12: PORTD I/O Allocation**

| Pin Name | Pin Number | Direction | Allocated Resource |
|---|---|---|---|
| RD0/PSP0 | 19 | Digital Output | Seven Segment Display: DA0 |
| RD1/PSP1 | 20 | Digital Output | Seven Segment Display: DA1 |
| RD2/PSP2 | 21 | Digital Output | Seven Segment Display: DA2 |
| RD3/PSP3 | 22 | Digital Output | Seven Segment Display: ID0 |
| RD4/PSP4 | 27 | Digital Output | Seven Segment Display: ID1 |
| RD5/PSP5 | 28 | Digital Output | Seven Segment Display: ID2 |
| RD6/PSP6 | 29 | Digital Output | Seven Segment Display: ID3 |
| RD7/PSP7 | 30 | Digital Output | Seven Segment Display: ID7 |

PORTE bits 0 and 1 provide the WRITE and MODE bits required to control the ICM7218D display driver.

**Table 13: PORTE I/O Allocation**

| Pin Name | Pin Number | Direction | Allocated Resource |
|---|---|---|---|
| RE0/NOT(RD)/AN5 | 8 | Digital Output | Seven Segment Display: WRITE |
| RE1/NOT(WR)/AN6 | 9 | Digital Output | Seven Segment Display: MODE |
| RE2/NOT(CS)/AN7 | 10 | Digital Input to GND | UNALLOCATED |

## *6.3. Control Software*

### 6.3.1. Overview

The control software is written using the Microchip PIC18F assembly instruction set. Control algorithms are organized into modules written assembly programming constructs known as subroutines. Modules are interconnected through subroutine calls, global lookup data tables, global registers, external interrupts from other units, and internal interrupts from timers.

The flow chart in Figure 8 presents the general flow of control logic for the temperature controller software. The vertically bordered boxes represent separate modules, or subroutines.

The default temperature for the temperature control system is based on the mid-range temperature that approximately corresponds to 50% hot water flow and 50% cold water flow, assuming equal flow through both shower water pipes. The variable percentage of water flow, x, by which each of the water pipes are modified by the coupled motors/valves shall be determined by rigorous testing. A relationship shall be determined between the motor/valve activation time and change of water temperature exiting the shower head. The correspondence will be incorporated into non-volatile program memory and provide a suitable estimate of motor activation time that will provide the desired target temperature. The percentage of change in temperature versus motor/valve activation time relationship is calibrated during installation.

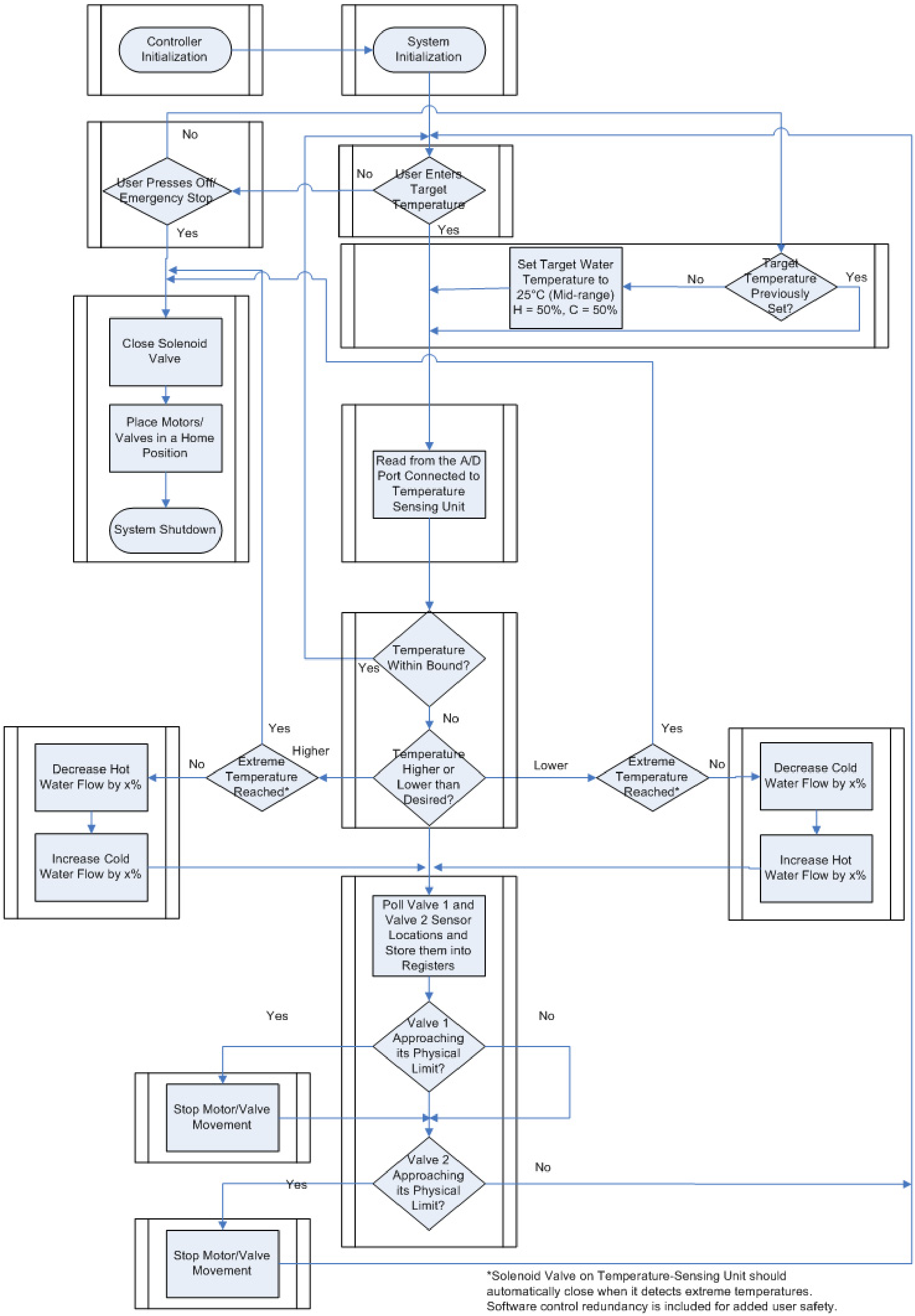


**Figure 8: The main control software flow chart.**

The following modules given in Table 14 are included in the temperature control unit.

**Table 14: Temperature Control Unit Software Modules**

| Module Name | Purpose |
|---|---|
| Controller Initialization | Place the temperature control unit into a known state and prepare the controller logic for normal system operation. |
| System Initialization | Adjust motors/valves into predefined states and initialize target temperature to a mid-range temperature. |
| Main Control | Main software arbitration through a series of subroutine calls and interrupt service routines. |
| A/D Conversion | Convert the analog voltage from the comparator of the temperature-sensing unit into a digital value. |
| Temperature Error | Compare the result of the A/D conversion with the desired user temperature. |
| Motor/Valve Activation | Activate the motor/valve to influence a change in water temperature. |
| Motor/Valve Position Monitoring | Keep track of the position of the motors/valves with respect to predefined digital states. |
| Motor/Valve Limiter | Limit the activation of motors/valves in each direction to prevent motor/valve control unit failure. |
| System Shutdown | Place the motor/valves into a known, home position, for the next shower session and de-energize the solenoid valve. |

### 6.3.2. Memory Allocation

The PIC18F452 has its memory organization based on the Harvard architecture, which provides separate buses and memory storage for program instructions and data, unlike the single memory storage and single bus organization provided by the Von Neumann architecture.

32 K of flash program memory is available for constants and stable program code consisting of the modules given above.

1536 bytes of RAM is available for variables including but not limited to the most recent value sampled from the temperature-sensing unit, temporary loop variables used by modules, and values sampled from the motor/valve positioning unit.

256 bytes of EEPROM memory is available for calibrating the system to a shower system with a particular flow rate. The ability for a data table to be created once during installation will be included in the temperature control software to determine the

relationship between motor/valve activation times versus the percentage of total flow. Such a data table provides estimates of the control parameters to be applied to the outputs of the temperature control unit that will reduce the search time for the system to modify the water temperature to the user desired target temperature.

The given memory allocation describes the maximum capacity of the PIC18F452. We shall organize modules within memory by adhering to the capacity limits of flash program memory, RAM memory, and EEPROM memory. The actual details of memory allocation are not specifically outlined because of the dynamic nature of the software implementation.

### 6.3.3. Controller Initialization Module

The controller initialization module for specifying the I/O pin directionality (Input/Output) of each I/O pin. For PORTA pins only, I/O pin type (Analog/Digital) should be selected and is achieved by writing to bits 3 to 0 in the ADCON1 register with semantics described in [6]. For all pins, directionality is set by writing to the data directionality register TRIS*x*, as described in Table 15 [6].

**Table 15: TRISx Data Directionality Register Parameters [3]**

| Parameter | Meaning |
|---|---|
| *x* | PORTx being configured where *x* = {A, …, E} |
| Bit Position | Pin within PORT*x* being configured |
| Bit Value | ‘1’ denotes Input; ‘0’ denotes Output |

### 6.3.4. System Initialization Module

Data such as last A/D digital result, last motor/valve positions, and other state data shall be cleared and set to default values given below in Table 16.

**Table 16: Default System State Variable upon System Initialization**

| State Data | Default Value |
|---|---|
| Default Temperature | Mid-range temperature value (25°C) |
| User Pressed Power Off/Emergency Stop | False |
| User Inputs Desired Temperature | False |
| Target Temperature Previously Set | False |
| Motor Position | Position based on a predefined algorithm that yields 50% hot water and 50% cold water flow (Refer to Section 6.3.11 for details) |
| Motor 1 at Limit | False |
| Motor 2 at Limit | False |
| A/D Temperature, Analog Input | Disabled |
| A/D Temperature, Digital Result | B‘00 00000000’[3] |
| Solenoid Valve Closed | False |
| Temperature Within Bound | False |
| Temperature Higher than Target | False |
| Extreme Low Temperature Reached | False |
| Extreme High Temperature Reached | False |

### 6.3.5. Main Control Module

The main control module is responsible for specifying the microcontroller options to be used upon controller startup. Although not explicitly shown on Figure 8, the main control module calls upon all the other subroutines, looping at the user input modules. The main control module is used to define global variables used by all control software modules.

The user input modules including button debouncing, button interrupt service routines, and button identification are discussed in the section on the *User Interface Unit*.

### 6.3.6. A/D Conversion Module

A/D conversion involves setting the ADCON1 register as mentioned in Section 6.3.3 for the desired pins to be tagged as analog inputs. Using the ADCON0 register, the analog input pin to be sampled is selected and A/D conversion is initiated [5]. By configuring the A/D interrupt enable bit, ADIE, of register PIE1, priority interrupt enable 1 [10], an interrupt can occur upon completion of A/D conversion. The result, that may be up to 10-bits, is stored in the registers ADRESH:ADRESL with justification of the bits controlled by the ADFM bit of the ADCON1 register [10]. One of the on-board timer modules may be scaled to regularly initiate A/D conversion of a selected channel.

For our purposes, A/D conversion shall occur during each loop and the results averaged over $2^n$ samples to facilitate simple division by two. A value of $n$ will be determined during initial and acceptance testing that will yield quick and accurate system response.

[3] B’XXXXXXXX’ is the notation for a binary number in PIC18F Instruction Set.

### 6.3.7. Temperature Error Module

A simple approach to handling the difference between the target temperature desired by the user and the actual shower temperature will be taken. The temperature error is given by

$$T_{err} = T_{target} - T_{actual}$$

where $T_{err}$ is the temperature error, $T_{target}$ is the target temperature desired by the user and $T_{actual}$ is the actual temperature measured by the temperature-sensing unit. Based on this error, the temperature error module is responsible for providing a motor/valve activation time[4] that is proportional to the target temperature and the actual temperature. As mentioned in Section 6.3.1, a manual calibration between activation time and change in temperature will be determined via installation and acceptance tests. The nature of the temperature error feedback classifies the *TemperSure* as a proportional gain control system. A sample proportional-integral-derivative (PID) controller module is available from Microchip and is a design alternative that can be used to optimize system response [7] at the cost of increased system complexity.

The motor/valve activation time determined by testing is entered into a data table. The calibration data table is accessed by a table index corresponding to the temperature error determined by the temperature error module. The motor/valve activation time in milliseconds ultimately translates into a discrete number of clock cycles. Hence, a clock cycle to milliseconds conversion ratio will be incorporated in the motor/valve activation module.

### 6.3.8. Motor/Valve Activation Module

Motor/valve activation occurs by writing a logic high bit to the bit of PORTA corresponding to the desired motor allocated in Table 9.

### 6.3.9. Motor/Valve Position Monitoring Module

The motor/valve positioning unit is read by the temperature controller via the motor/valve position-monitoring module. Each of the sensors of the motor/valve positioning unit is connected to each bit of PORTC. The area between two optical sensors is considered a discrete state that gives the temperature controller an indication that the position of the motor is bounded by two optical sensors.

### 6.3.10. Motor/Valve Limiter Module

The motor/valve limiter module relies on the motor/valve position-monitoring module described above. Polling the lower bits of PORTC associated with Motor 1 indicates to the controller that Motor 1 is approaching the lower limit and the upper limit of the physical rotational constraints of the motor/valve. A similar approach can be taken to poll the upper bits of PORTC that are associated with Motor 2. The motor/valve limiter module stops a motor from moving when a limit is reached.

[4] Motor/valve activation time is defined as the amount of time a DC current is applied to the motor.

If the control system is unable to reach the target temperature of the user because a motor/valve is being limited, either the temperature is shutoff if an extreme temperature is reached, or the water continues to flow but the display continues blinking indicating that the desired temperature entered by the user cannot be reached.

### 6.3.11. System Shutdown Module

The system shutdown module will call the motor/valve activation module, the motor/valve position-monitoring module, and the motor/valve limiter module to place the motor into a home position. During this homing sequence, the analog input to the A/D converter will be disabled and will only be enabled once the motors/valves have reached the end of the homing sequence. The homing sequence will be based on the feedback given by the motor/valve position sensors only, not the feedback from the temperature sensor. Lastly, system shutdown module sends a command to the temperature-sensing unit control circuitry to close the solenoid valve.

## *6.4. Software Test Plan*

### 6.4.1. General Test Plan for Software Modules

A general test plan for testing *TemperSure* software modules consists of white box and black box module testing.

### 6.4.2. Black Box Test Plan

Individual modules will be tested with the inputs given in Table 17.

**Table 17: Black Box Test Inputs and Expected Outputs.**

| Module Name | Inputs | Expected Outputs |
|---|---|---|
| Controller Initialization | None | I/O ports initialized according to temperature control unit: I/O Allocation |
| System Initialization | None | State variables initialized according to system initialization specifications |
| A/D Conversion | Thermocouple amplifier output of temperature-sensing unit connected to pin RA0/AN0<br>Range: 0 to 1 V | Value proportional to voltage input that is updated every clock cycle |
| Temperature Error | Values representing actual temperature and desired target temperature | Difference between the actual and desired temperature converted into a value representing activation time |

| | | |
|---|---|---|
| Motor/Valve Activation | Number representing motor activation time | Proportional motor/valve movement |
| Motor/Valve Position Monitoring | Motor/valve movement | Numbers in the motor/valve position register corresponding to activated motors |
| Motor/Valve Limiter | Motor/valve movement approaching a limit | Motor/valve movement halted |
| System Shutdown | 1. Temperature at 55°C<br>2. Temperature at 0°C<br>3. Normal User Power Off/Emergency Stop | Solenoid valve closed; motor/valve homing sequence completed with last motor position located between the same two sensors for each case. |
| Main Control | Controller Initialization and System Initialization successfully completed | Normal system operation as per temperature controller flowchart |

#### 6.4.3. White Box Test Plan

White box testing of software modules is dependent on implementation. The following procedure will guide the general white box testing:

1. Record the values of the following parameters:
    a. System state variables
    b. Status register
    c. General-purpose and special function registers related to the module
    d. Input/Output voltages and currents of external circuits used by the module
2. Stimulate the software module with analog/digital inputs listed in Table 17.
3. Repeat Step 1.
4. Compare parameter changes in Step 3 and determine the correctness of outputs in Table 17.

# 7. Piping

## 7.1. Overview

The *TemperSure* customized motor-driven valves are to be integrated into the existing shower plumbing system. For convenient construction of a shower testbench, cross-linked polyethylene (PEX) will be used as a substitute for copper piping. However, copper piping cannot be entirely avoided, as the solenoid valve of the temperature-sensing unit requires connections to copper pipes. To connect the hot and cold water pipes, a T valve will be used to mix the two water streams. Other connectors and clamps will be used as necessary to complete the system.

### 7.2. Plastic Piping

For the majority of the system, Cross-linked polyethylene (PEX) plastic piping will be used. This type of piping has the advantage of being easy to install and does not require welding for pipe connections. The PEX piping used is manufactured by BOW Plastics with a maximum pressure rating of 100 PSI at a temperature of 82°C [11].

### 7.3. Copper Piping

For the connection to and from the solenoid valve, copper piping is used. The copper piping used is manufactured by Cerro Copper Canada, rated at a pressure maximum of 150PSI [2].

## 8. User Interface Unit

### 8.1. Overview

The user interface unit allows user to start a shower session, to stop a shower session under normal and emergency conditions, and to increase/decrease water temperature. The simple unit consists of three seven-segment displays and a set of three push buttons.

### 8.2. User Interface Hardware

#### 8.2.1. User Interface Display

The user input display consists of a series of three MAN74A 7-segment LED displays (red coloured). We considered using LCD displays, but showers can be a dim area. LEDs offer a brighter and more visible solution. The MAN74As are standard IC size and a little smaller than our projected ideal size, however the standard IC size makes for an easier implementation in our preliminary model.

To decrease the burden of the microcontroller we decided to use the ICM7218D 7-segment display driver. The ICM7218D has a parallel interface and can control up to eight 7-segment displays. We considered using a serial interface driver, however that would result in a longer software program to display the digits. Since the microcontroller I/O pins are not used to capacity, extra pins are readily available. The circuit diagram for the LED display circuit is shown in Figure 9.

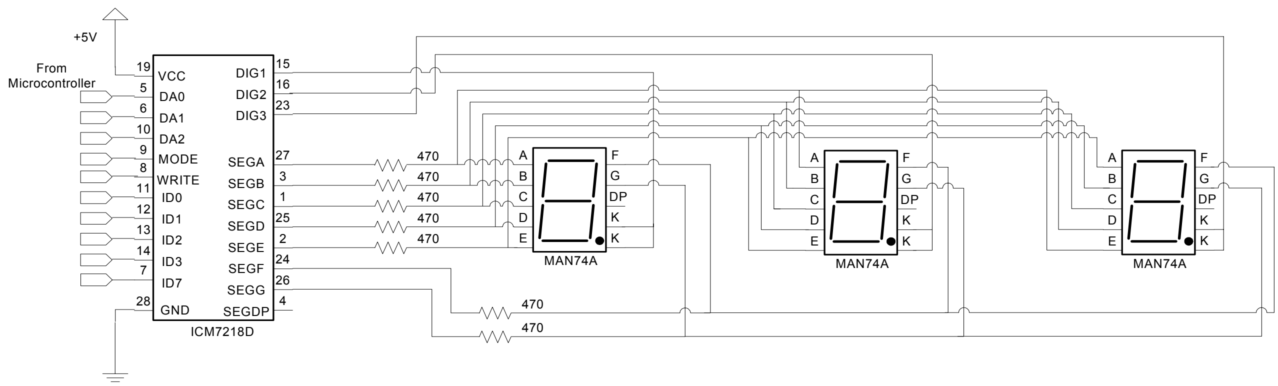


**Figure 9: The circuit diagram for the user interface display.**

The ICM7218D's MODE pin (pin 9) needs special consideration. The pin can have states of HIGH for hexadecimal decoding, FLOATING for binary decoding, and LOW for Shutdown mode. To achieve all three states, we could use a tri-state buffer. Since we will be keeping to a binary decoding, we can use the configuration shown in Figure 10.

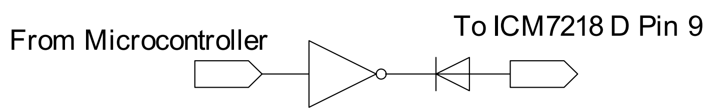


**Figure 10: The interface to the ICM7218D's MODE pin [3].**

For the circuit in Figure 10, a high from the microcontroller will cause the ICM7218D to enter SHUTDOWN, and a low will result in a BINARY decoding mode.

### 8.2.2. Buttons

User input is provided by a set of Circuit-Test standard SPST pushbutton switches; two of type 456-701 and one of type 456-702. The buttons will be used to control the following system functions given in Table 18.

**Table 18: Button Specifications [12]**

| Function | Button Colour | Part Number | Electrical Specifications | Physical Dimensions |
|---|---|---|---|---|
| On/Off/ Emergency Stop | Red | Circuit-Test 456-702 | 3A/125VAC | 13mm (1/2") Mounting Hole |
| Increase Water Temperature | Black | Circuit-Test 456-701 | 3A/125VAC | 13mm (1/2") Mounting Hole |
| Decrease Water Temperature | Black | Circuit-Test 456-701 | 3A/125VAC | 13mm (1/2") Mounting Hole |

Different button colours have been chosen to differentiate the buttons. To differentiate between the two black buttons, white arrows pointing upward and downward are to be added to the *Increase Water Temperature* and *Decrease Water Temperature* buttons, respectively. Alternatively, a plus sign and minus sign could be used as labels to the *Increase Water Temperature* and *Decrease Water Temperature* buttons, respectively. Figure 11 shows the configuration of the push buttons connected to the microcontroller.

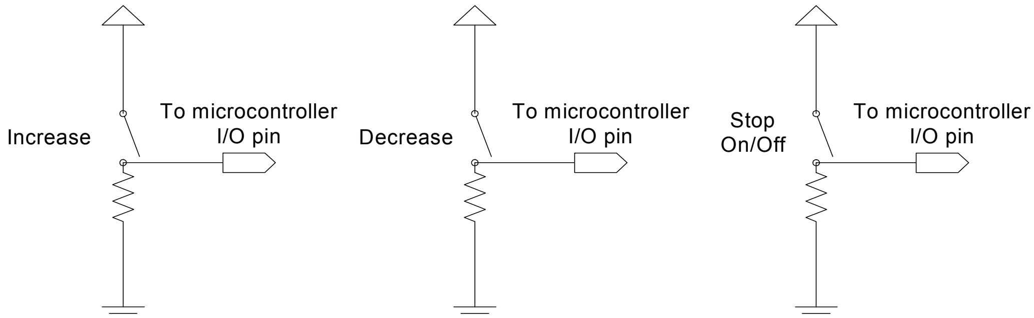


**Figure 11: The configuration of the push buttons connected to the microcontroller.**

The push buttons are configured for active high signal sent to the microcontroller.

## 8.3. User Interface Hardware Verification

To assure the user interface is functioning properly, we must manually enter input signals to the display driver and observe the following:

1. Display digits 0-9 on the first display.
2. Display digits 0-9 on the second display.
3. Display digits 0-9 on the third display.
4. Toggle in and out of shutdown mode.
5. When each button is pressed, a logic high is observed in the appropriate connection to the microcontroller.

## 8.4. User Interface Software

### 8.4.1. Overview

The User Interface software will take input from the buttons pushed from the user display and increase the desired temperature, decrease the desired temperature or turn on/off *TemperSure*. The software is also responsible for displaying the appropriate message to the 7-segement display giving feedback to the user on the current state of our system. A detailed flow chart of the User Interface software can be seen in Appendix B. Because of the temperature limitations of our system, the user will only be able to select desired temperature ranges from 5-55°C.

### 8.4.2. Button Identification Module

To avoid a constant polling of the button states, the buttons will be mapped to interrupts. Once pressed, an interrupt will occur and we must identify which button was pressed. A transition period exists when a push button is depressed where the state of the button is undefined. In order to sample the state of the button in a defined state, we must debounce the button, or introduce a delay before sampling the button state. The delay time required will be determined through testing.

#### 8.4.3. Seven Segment Display Module

The user interface displays the desired temperature at all times; the value is also stored in the microcontroller memory. To display the appropriate number on the user interface, a look-up table of constants is used, which contains the appropriate data to send to display driver. As the desired temperature is changed through button presses, the microcontroller will look-up the new data value to send to the display driver.

In the case when the target temperature is not equal to the actual temperature, the display will flash to indicate the system is in transition to the desired temperature. The flashing display is accomplished by initiating the SHUTDOWN condition to the MODE pin. A logic high from the microcontroller to the circuit in Figure 10 will cause a SHUTDOWN condition. The display will strobe off and on when entering in and out of SHUTDOWN mode.

The look-up table will also contain states for the error codes. In the case of an error code, the desired temperature will be replaced with an error code value in memory, resulting in the error code displayed on the user interface.

### 8.5. User Interface Software Verification

To test our software portion we must perform the following tests.

1. Display digits 0-9 on the first display.
2. Display digits 0-9 on the second display.
3. Display digits 0-9 on the third display.
4. Display the numbers 5-55 on the user interface.
5. Display error codes on the user interface.
6. When the buttons are pressed, an interrupt occurs.
7. When the *Increase Water Temperature* button is pressed, the displayed temperature increases in value.
8. When the *Decrease Water Temperature* button is pressed, the displayed temperature decreases in value.
9. When the desired (displayed) water temperature is different then the actual water temperature, the display will flash.
10. The pressing of the *On/Off/Emergency Stop* button is noted.

## 9. System Test Plan

### 9.1. Normal Case: Nominal Water Flow Rate

User Input: Desired Temperature of 30°C
Conditions: Nominal water flow through hot and cold pipes
Expected Observation: System operation as per flow chart

### 9.2. Extreme Case 1: Reduced Cold Water Flow Rate

User Input: Desired Temperature of 30°C
Conditions: Reduce flow rate in cold water pipe to 20% of nominal
Expected Observation: The water temperature increases because of a reduction in cold water flow.

If water temperature goes above 55°C:
1. The solenoid valve of the anti-scald unit is closed
2. The system shuts down.
3. Motors return to a known, home position

If water temperature remains below 55°C:
1. Display blinks to indicate the temperature is changing.
2. System compensates by adjusting both valves within valve physical limit.
3. As valves approach their physical limit, motors stop.
4. Display continues to blink.
5. If nominal cold water flow rate is restored, then the system adjusts the motors to maintain the desired temperature within the valve limits.

### 9.3. Extreme Case 2: User Enters a Desired Temperature Higher than High Cutoff Temperature of System

User Input: Desired Temperature of 55°C
Conditions: Nominal water flow through hot and cold pipes
Expected Observation:
1. Display blinks as user presses the *Increase Temperature* button.
2. Display limits user to entering 55°C.

## 10. Conclusion

The proposed hardware and software design solutions used to implement functional requirements of the Tempero *TemperSure* shower temperature control system have been discussed in this design specification. During the development phases, the design requirements are evaluated against the corresponding functional requirements. Testing requirements discussed for each module are provided to verify design approaches and desired system functionality. The design specification provides a guideline of design goals and test approaches for developing and verifying the functionality of a proof-of-concept system.

# 11. References

[1] Analog Devices Inc. 1999. *Monolithic Thermocouple Amplifiers with Cold Junction Compensation*. 2004 Oct 30. <http://www.analog.com/UploadedFiles/Data_Sheets/421725987AD594_5_c.pdf>.

[2] Cerro Flow Products, Inc. 2004. *Plumbing Tube – Technical Data*. 2004 Oct 31 <http://www.cerrocu.com/pdfs/CFP-2418_Cerrotube_FINAL_LORES.pdf>

[3] Dallas Semiconductor. 1996. *8 Digit LED Display Driver*. 2004 Oct 20 <http://pdfserv.maxim-ic.com/en/ds/ICM7218-ICM7228.pdf>.

[4] EnergyGuide, *Water Heater Temperature Safety*. 2004 Oct 29 <http://library.energyguide.com/EnergyLibraryTopic.asp?bid=minnpower&prd=10&TID=13361&SubjectID=7746>.

[5] Microchip, Inc. 2000. *PICmicro® 18C MCU Family Reference Manual*. 2004 Oct 31 <http://ww1.microchip.com/downloads/en/DeviceDoc/39500a.pdf>.

[6] -----. 2002. *PIC18FXX2 Data Sheet*. 2004 Oct 31 <http://ww1.microchip.com/downloads/en/DeviceDoc/39564b.pdf>.

[7] -----. 2004. *Microchip Application Note 937: Implementing a PID Controller Using a PIC18 MCU*. 2004 Nov 01 <http://ww1.microchip.com/downloads/en/AppNotes/00937a.pdf>.

[8] -----. *Microchip PIC 16 Comparison Chart*. 2004 Oct 30 <http://www.microchip.com/ParamChartSearch/chart.aspx?mid=10&lang=en&branchID=1002>.

[9] National Semiconductor. 1973. *A Quad of Independently Functioning Comparators*. 2004 Nov 1 <www.national.com/an/AN/AN-74.pdf>.

[10] Peatman, J.B. 2003. *Embedded Design with the PIC18F452 Microcontroller*. Upper Saddle River, NJ: Pearson Education, Inc.

[11] Plastic Pipe and Fittings Association (PPFA). *PEX Products.* 2004 Oct 31 <http://www.ppfahome.org/pex/index.html>.

[12] RP Electronics – Online Catalog – Switches & Relays: Push Buttons. 2004 Oct 30 < http://www.rpelectronics.com/English/Content/Divisions/Div_40_140.asp>.

[13] Weisstein, Eric W. 2003. *Wien's Displacement Law – Eric Weisstein's World of Physics*. 2004 Oct 30 <http://scienceworld.wolfram.com/physics/WiensDisplacementLaw.html>.

# Appendix A: Mechanical Schematics

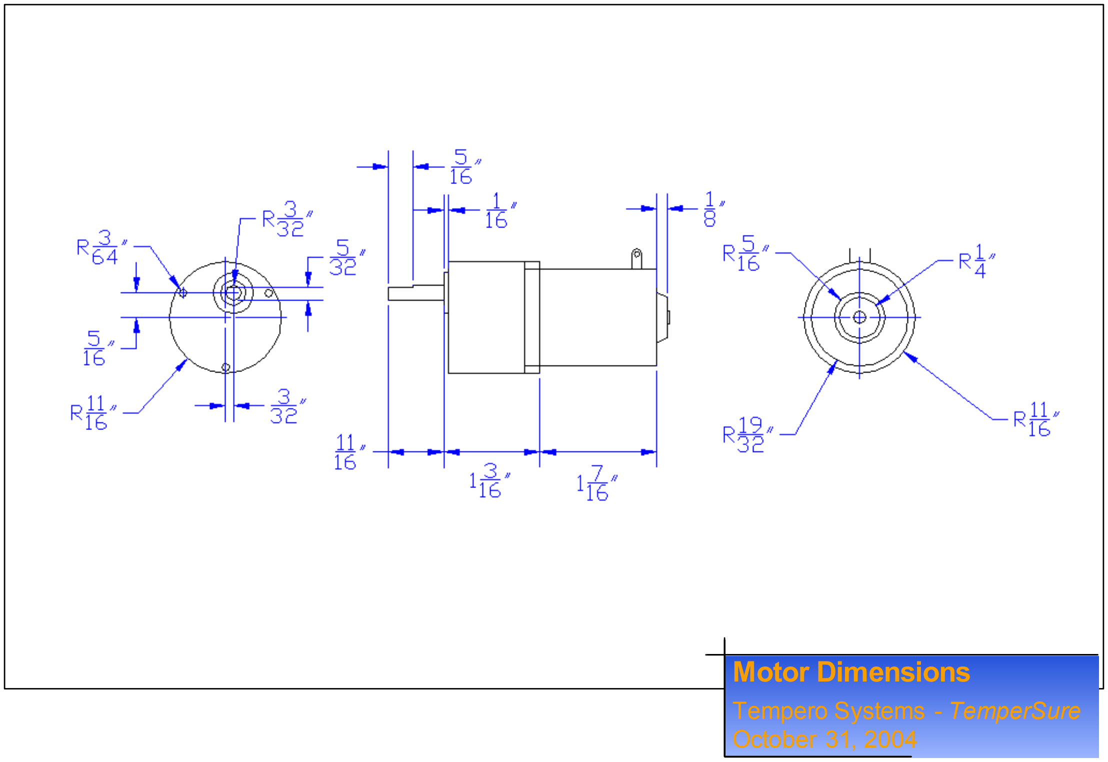


**Figure 12: Motor Dimensions.**

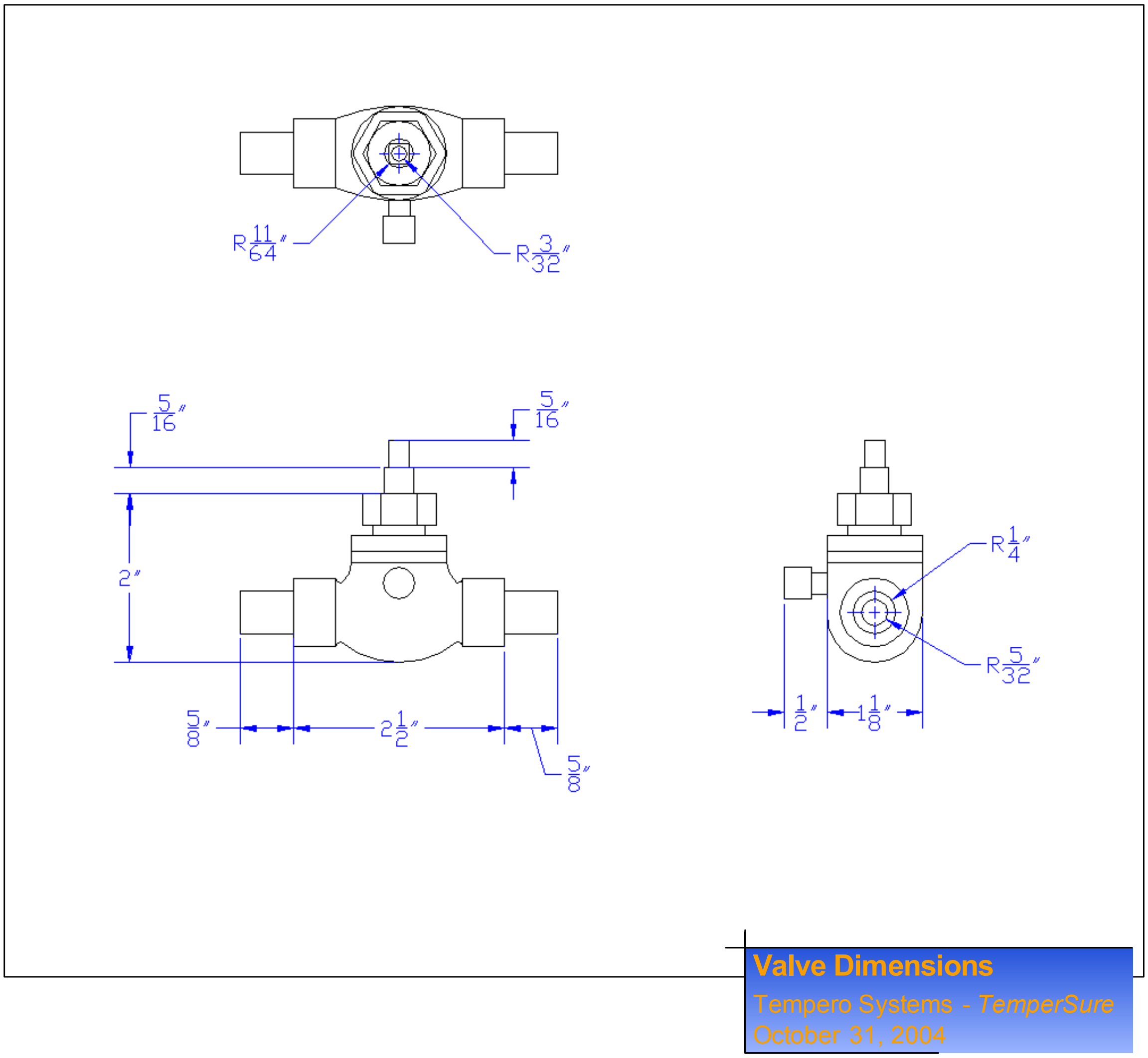


**Figure 13: Valve Dimensions.**

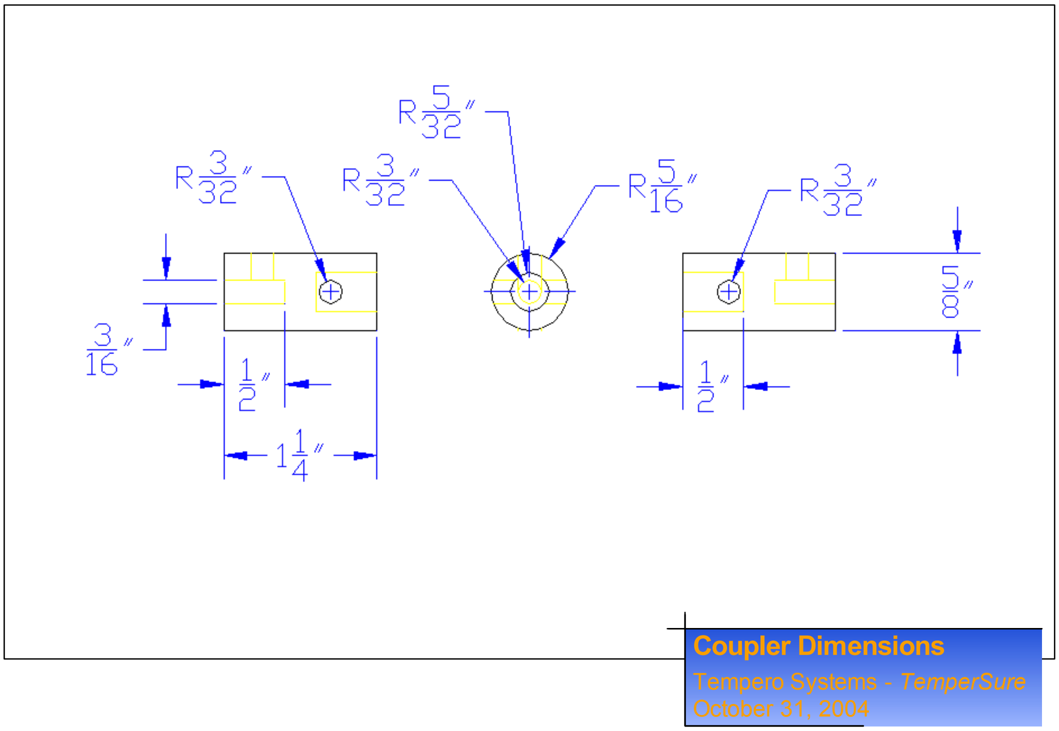


**Figure 14: Coupler Dimensions.**

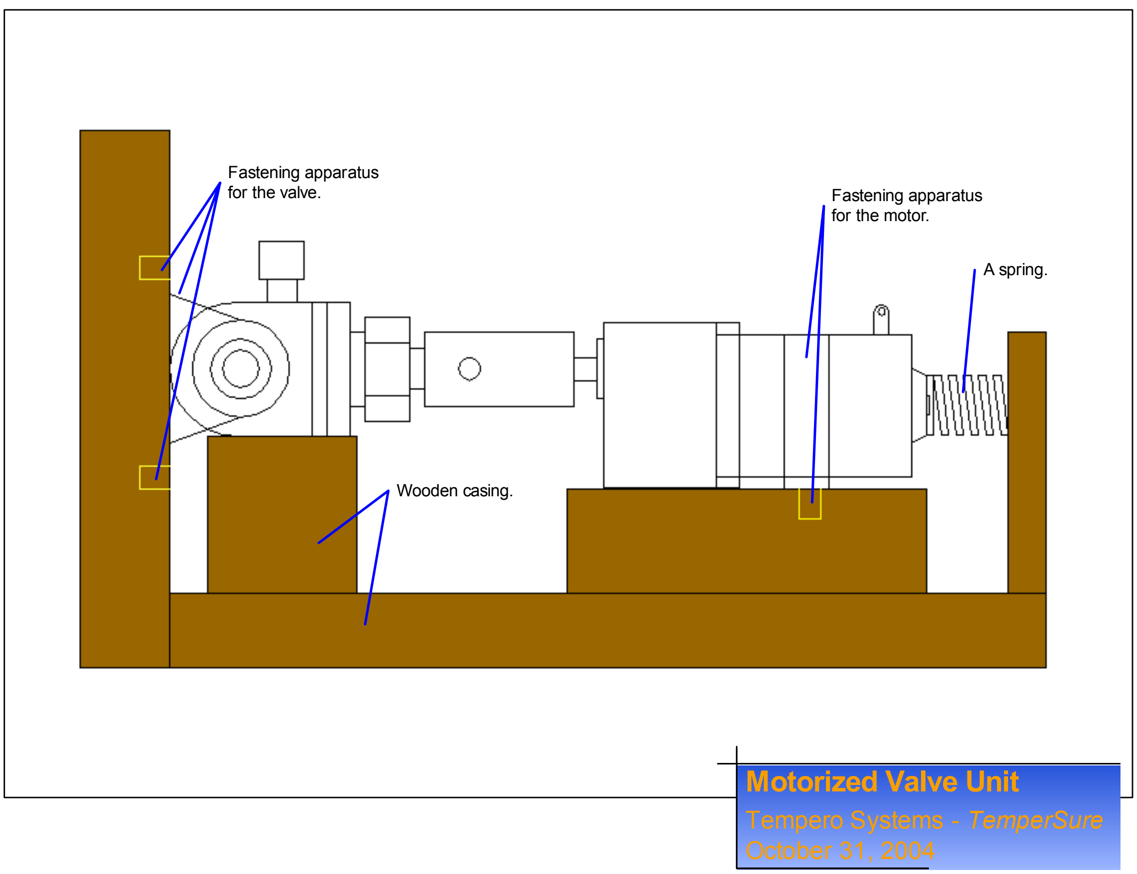


**Figure 15: The motorized valve unit.**

# Appendix B: Software Flow Charts

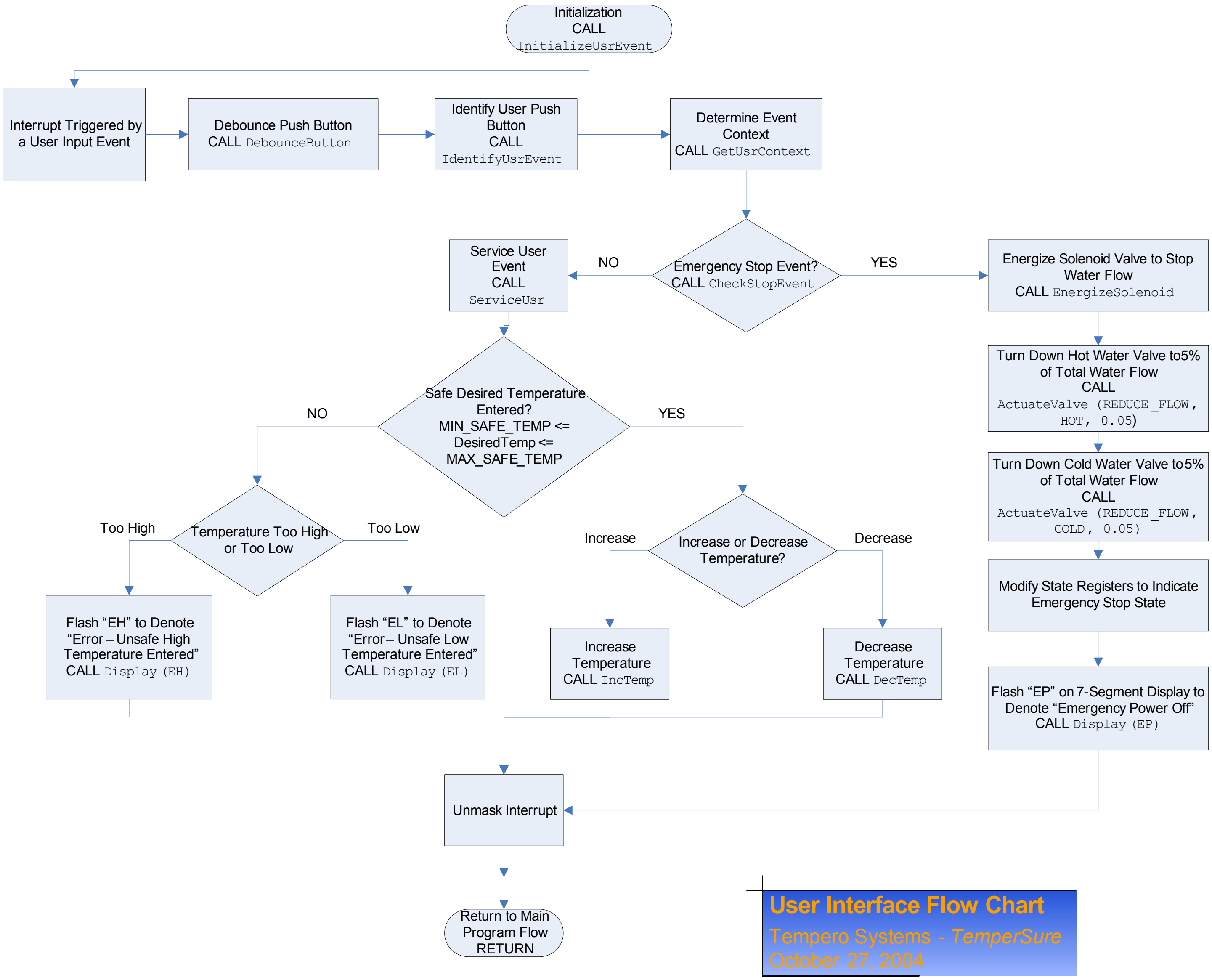


**Figure 16: The user interface flow chart.**

# Appendix C: Hysteresis Threshold Calculation [9]

The inverting Schmitt Trigger has the configuration seen in Figure 17.

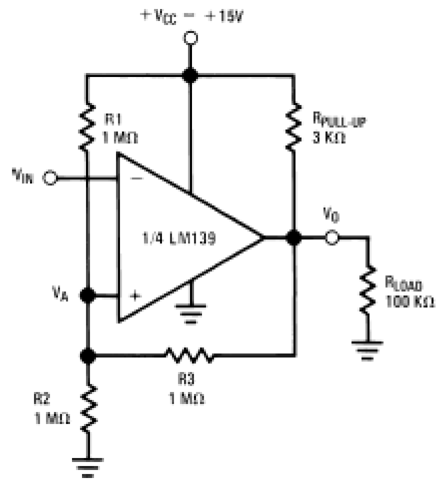


**Figure 17: Configuration of an inverting Schmitt trigger.**

To ensure Vo will swing fully from Vcc to ground, its important to choose,

$$R_{pull\text{-}up} < R_{load} \text{ and } R_3 > R_{pull\text{-}up}.$$

When output is high, the output transistor of the comparator is a open circuit. So, the whole circuit can be reduced to Figure 18.

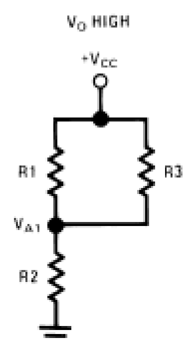


**Figure 18: The equivalent circuit for the Schmitt trigger high output.**

We can see that in Figure 18,

$$V_{A1} = V_{cc}\ (R_2/(R_1||R_3)+R_2).$$

When output is low, the transistor in the comparator output conducts. Therefore, the circuit can be simplified to Figure 19.

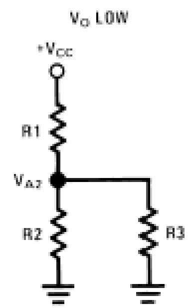


**Figure 19: Equivalent circuit for a Schmitt trigger low output.**

Resulting from Figure 19,

$$V_{A2} = V_{cc} (R_2||R_3)/((R_2||R_3)+R_1).$$

# Appendix D: Thermocouple Reference [1]

## AD594/AD595

### THERMOCOUPLE BASICS

Thermocouples are economical and rugged; they have reasonably good long-term stability. Because of their small size, they respond quickly and are good choices where fast response is important. They function over temperature ranges from cryogenics to jet-engine exhaust and have reasonable linearity and accuracy.

Because the number of free electrons in a piece of metal depends on both temperature and composition of the metal, two pieces of dissimilar metal in isothermal and contact will exhibit a potential difference that is a repeatable function of temperature, as shown in Figure 14. The resulting voltage depends on the temperatures, T1 and T2, in a repeatable way.

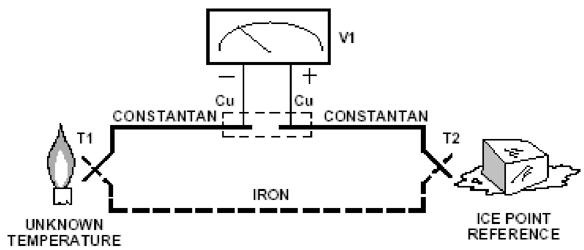


*Figure 14. Thermocouple Voltage with 0°C Reference*

Since the thermocouple is basically a differential rather than absolute measuring device, a know reference temperature is required for one of the junctions if the temperature of the other is to be inferred from the output voltage. Thermocouples made of specially selected materials have been exhaustively characterized in terms of voltage versus temperature compared to primary temperature standards. Most notably the water-ice point of 0°C is used for tables of standard thermocouple performance.

An alternative measurement technique, illustrated in Figure 15, is used in most practical applications where accuracy requirements do not warrant maintenance of primary standards. The reference junction temperature is allowed to change with the environment of the measurement system, but it is carefully measured by some type of absolute thermometer. A measurement of the thermocouple voltage combined with a knowledge of the reference temperature can be used to calculate the measurement junction temperature. Usual practice, however, is to use a convenient thermoelectric method to measure the reference temperature and to arrange its output voltage so that it corresponds to a thermocouple referred to 0°C. This voltage is simply added to the thermocouple voltage and the sum then corresponds to the standard voltage tabulated for an ice-point referenced thermocouple.

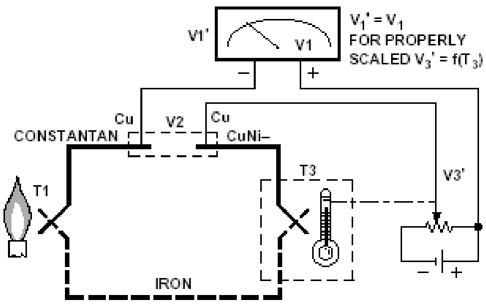


*Figure 15. Substitution of Measured Reference Temperature for Ice Point Reference*

The temperature sensitivity of silicon integrated circuit transistors is quite predictable and repeatable. This sensitivity is exploited in the AD594/AD595 to produce a temperature related voltage to compensate the reference of "cold" junction of a thermocouple as shown in Figure 16.

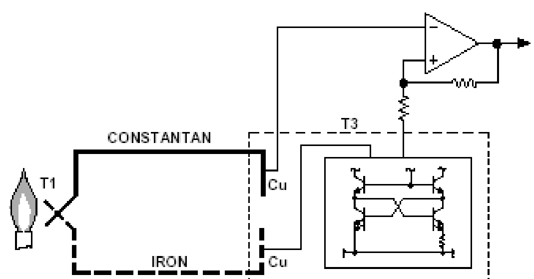


*Figure 16. Connecting Isothermal Junctions*

Since the compensation is at the reference junction temperature, it is often convenient to form the reference "junction" by connecting directly to the circuit wiring. So long as these connections and the compensation are at the same temperature no error will result.