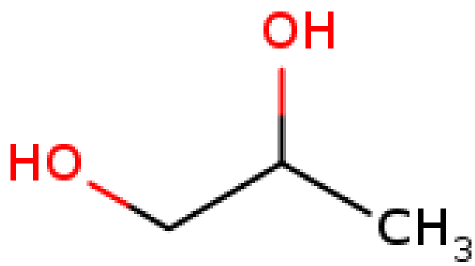
Department of Computer Science & Engineering
The University of Texas at Arlington

Detailed Design Specification
CSE 4317: Senior Design II
Spring 2021



Penny Pitcher
Glycol Chiller Fermentation

Janine Batongmalaki
Abhishek Dhital
Connor Glasgow
Elmer Rivera-Molina
Joshua Venter

REVISION HISTORY

Revision	Date	Author(s)	Description
0.1	3.05.2021	JB	Document Created

Contents

List of Figures

List of Tables

1 INTRODUCTION

Our product is a glycol Chiller for use during the fermentation phase of brewing beer. With our product, a user will be able to chill fermenting alcoholic intermediaries. The product will achieve this by cooling a reservoir of propylene glycol, and piping the fermenting wort through pipes in the reservoir. The product will be made with the cooling element from a consumer grade AC unit. The system will be controlled with a Raspberry PI with software pre-loaded onto the PI. The user will be able to interact with system via a touch screen display. On the display, the user will be able to set the desired temperature of the glycol reservoir, declare preset temperatures to use, and select a preset from the user defined presets. The product will only be able to chill the contents of one fermentation vessel at a time.The product will be a cooler used as the reservoir for the glycol, with the "internals" in an enclosure attached to the cooler.

The key requirements of our product include: chilling the contents of one vessel, user controlled temperature setting, and temperature reading accuracy.

If time allows the user will be allowed to interact with the system via a smartphone application and recipe files that allow automatic fermentation steps to be loaded and executed onto the glycol chiller without a user present at all.

2 SYSTEM OVERVIEW

The Glycol Fermentation system will consist of three main layers: The Control System, Glycol Management System, and the Fermentation Management System. The Control System will communicate with both the Glycol Management System and the Fermentation Management System.

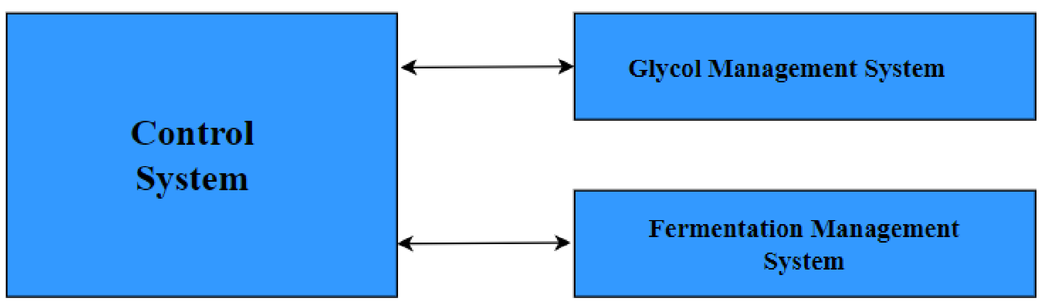


Figure 1: Penny Pitcher's Glycol Fermentation System architectural layer diagram

2.1 GLYCOL MANAGEMENT DESCRIPTION

When fermenting wort into beer, heat is released in the chemical reaction that ensues. For the beer to come out the way the brewer wants, the temperature needs to be held at a specific range of temperatures. This temperature range is managed by pumping glycol through a coil held inside the fermenter. The Glycol Management layer is in charge of cooling the glycol down to a low enough, constant temperature that allows for simpler calculations by the Fermentation management layer. This layer consists of a single subsystem, the glycol Pump system.

2.2 FERMENTATION MANAGEMENT DESCRIPTION

Part of the brewing process involves fermenting wort into beer. This is done in a separate system called the fermenter. The Fermentation Management layer will be interacting with said fermenter. In this layer we will be keeping track of the temperature inside of the fermenter as well as the specific gravity of the beer. These measurements will be taken with a Bluetooth Hydrometer and sent to the Control

Layer. The subsystems of this layer include: Temperature sensor, Specific Gravity Sensor, and ac power controller(Bluetooth plug that functions as an on and off switch).

2.3 Control Description

The Control layer will regulate the functionality of the overall system and communicate with most of the project's subsystems, including that of the Fermentation Management and Glycol Management layers. This layer will be facilitated through a Raspberry Pi and will come pre-installed with our custom built software for this purpose. Regarding user interaction, manual controls will be through the use of the touchscreen raspberry pi, and semi-automated controls will be applied through a mobile application.

3 Glycol Management Layer Subsystem

This section describes the details of the Glycol Management Layer. This layer is responsible for chilling the glycol reservoir down to a temperature that can be used to cool the fermenting wort in the fermenter. This layer consists of a single high level layer, the AC Unit.

3.1 Layer Hardware

This subsystem will be a re-purposed AC Unit. The AC unit has been removed from its casing to allow access to the Evaporator and Condenser coils. The Evaporator coils, here on out referred to as the 'cool coils', take the refrigerant within the AC itself and cool the coils. These cool coils are placed inside the glycol reservoir, lowering the temperature of the glycol. The Condenser coils, here on to be referred to as the 'hot coils', take the heat from the process and is exhausted outside of the system via a fan in the AC unit itself.

3.2 Layer Operating System

This Layer has no operating system.

3.3 Layer Software Dependencies

This layer has no software dependecies.

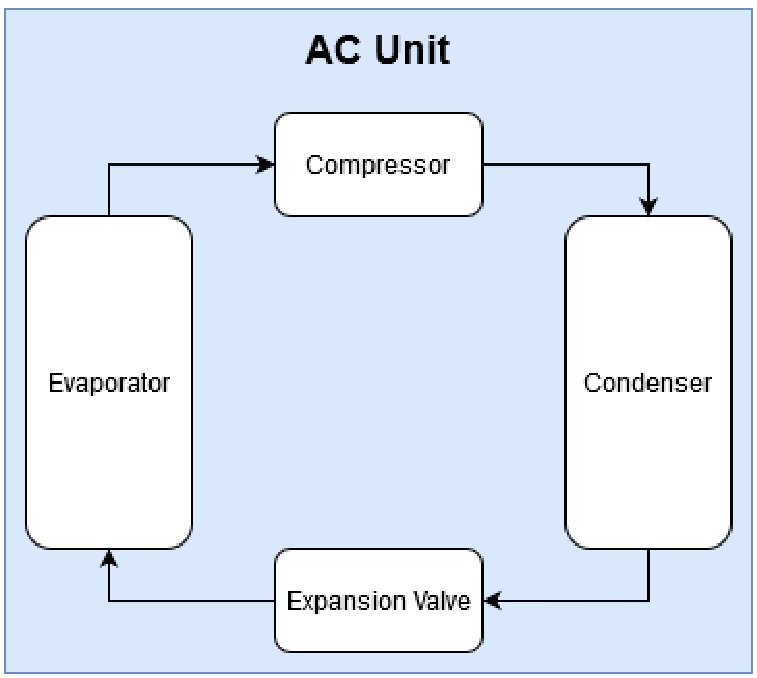


Figure 2: Subsystem Diagram of an AC Unit

3.4 Compressor

The compressor is a piece of hardware that circulates the refrigerant necessary for heat exchange in the system coils. A motor runs that condenses the refrigerant, raising the temperature of the refrigerant. The refrigerant releases its heat in the hot coils and turns into a liquid. Then the refrigerant moves to the cool coils and travels to the cool coils, absorbs the heat from the glycol, and travels back to the compressor.

3.4.1 Subsystem Hardware

The hardware of this subsystem is simply the compressor itself.

3.5 Condenser

The condenser coils are the hot coils. The hot coils release the heat absorbed from the cool coils and releases said heat outside of the system via a blower fan.

3.5.1 Subsystem Hardware

The hardware of this subsystem is a set of coils attached to a large heat sink, and a fan to blow the air past the coils, exhausting the heat.

3.6 Expansion Valve

The expansion valve removes pressure from the refrigerant when in liquid form, to allow it to change its state from liquid to solid.

3.6.1 Subsystem Hardware

The hardware in this subsystem is only the expansion valve.

3.7 Evaporator

The evaporator coils are the cool coils. The cool coils absorb the heat from the reservoir of glycol, lowering the temperature of the glycol in the process.

3.7.1 Subsystem Hardware

The hardware of this subsystem is a set of coils attached to a large heat sink.

4 Fermentation Management Layer Subsystem

The Fermentation layer contains three subsystems: Temperature sensor, Specific Gravity Sensor, and pump system. All of these subsystems help monitor the temperature in the fermentation container as well as manage the glycol being sent into the fermentation container.

4.1 Layer Hardware

The hardware in the fermentation management system consists of a fermentation tank, pump with special piping to transfer glycol, and a Bluetooth hydrometer. The Bluetooth hydrometer consist of a temperature sensor and specific gravity sensor. Additionally, the Raspberry Pi 3 runs a software process in order to interface with the hydrometer and the pump.

4.2 Layer Operating System

The Fermentation Management requires Raspbian OS in order to interface with the three subsystems.

4.3 Layer Software Dependencies

The fermentation layer uses a python library- pytilt, which is specifically used to get the temperature and specific gravity readings that the bluetooth hydrometer constantly transmits. This library also supports sending the captured data to a online database in case we need to keep records of the readings we get over time. The readings from this layer is forwarded to the control unit.

4.4 Temperature Sensor

The first subsystem of this layer will be the temperature sensor of the Bluetooth hydrometer unit. It measures and sends the temperature reading from inside the reservoir to the Control Unit when the temperature changes beyond 2%.

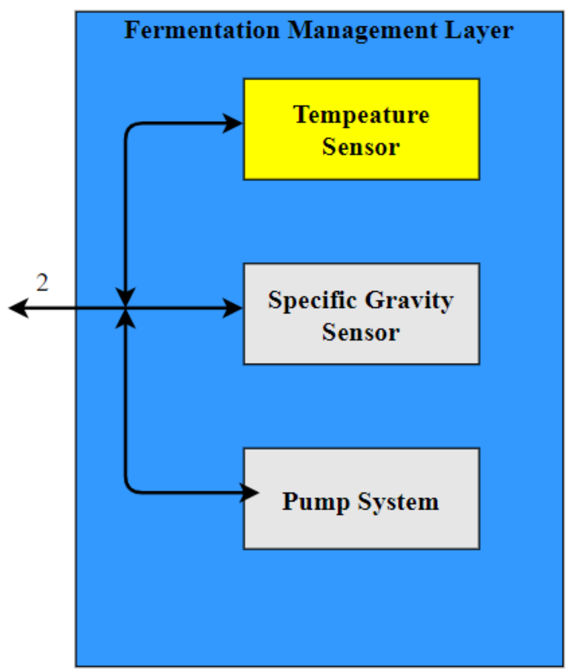


Figure 3: Temperature Sensor subsystem description diagram

4.4.1 Subsystem Hardware

The temperature sensor subsystem consists of an thermometer sensor along with other circuit components such as resistors, capacitors, etc. Additionally, all of these components are housed on a small PCB board. Additionally, a transmitter is used to send data.

4.4.2 Subsystem Operating System

The system requires Raspbian OS in order to be able to compile and run the python libraries to interface with the temperature sensor in the Bluetooth hydrometer.

4.4.3 Subsystem Software Dependencies

This subsystem sends the temperature readings via bluetooth to the control unit from where the functioning of the overall system is manipulated. The software of this subsystem comes preinstalled with the bluetooth hydrometer.

4.4.4 Subsystem Programming Languages

The System uses Python in order to be programmed to get temperature readings.

4.4.5 Subsystem Data Structures

The temperature reading is transmitted to the Raspberry Pi 3 through a Bluetooth transmitter on the hydrometer. The packets of data are structured into chunks and sent using frequency-hopping.

4.4.6 Subsystem Data Processing

The data is processed and transmitted by the software that is built-in with the bluetooth hydrometer that we use for this purpose. We did not implement any extra data processing algorithm for this subsystem.

4.5 Specific Gravity Sensor

This subsystem is similar to the temperature sensor, and is also the part of a Bluetooth hydrometer. It measures and sends the readings of specific gravity of fermented beer in the reservoir to the Control Unit. This allows us to monitor the level of glycol in the fermenter as well as estimate the alcohol content of the beer being made.

4.5.1 Subsystem Hardware

The Specific Gravity subsystem consist of an accelerometer and a gyroscope along with other common circuit components such as resistors, capacitors, etc. Like the temp sensor, it is also on a PCB board. Additionally, a transmitter is used to send data.

4.5.2 Subsystem Operating System

The system requires Raspbian OS in order to be able to compile and run the python libraries to interface with the accelerometer and gyroscope in the Bluetooth hydrometer.

4.5.3 Subsystem Software Dependencies

This subsystem sends the specific gravity readings via bluetooth to the control unit from where the functioning of the overall system is manipulated. The software of this subsystem comes preinstalled with the bluetooth hydrometer.

4.5.4 Subsystem Programming Languages

The System uses Python in order to be programmed to get specific gravity readings.

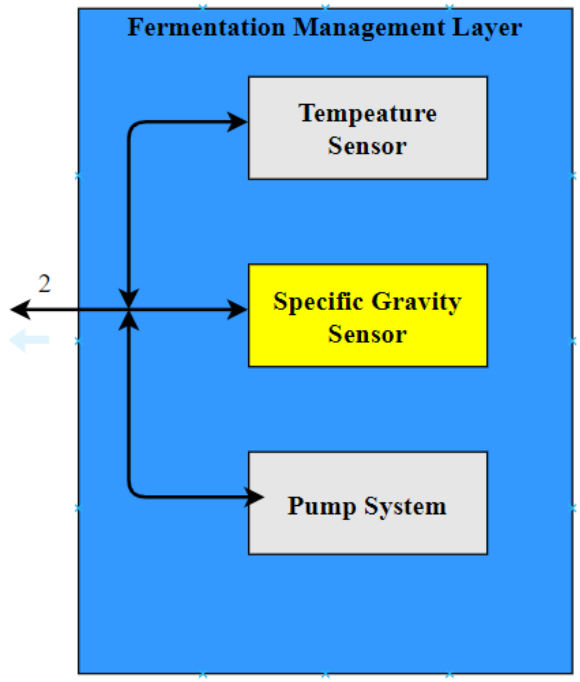


Figure 4: Specific Gravity Sensor subsystem description diagram

4.5.5 Subsystem Data Structures

The specific gravity reading is transmitted to the Raspberry Pi 3 through a Bluetooth transmitter on the hydrometer. The packets of data are structured into chunks and sent using frequency-hopping.

4.5.6 Subsystem Data Processing

Similar to the temperature sensor subsystem, the data in this subsystem is also processed and transmitted by the software that is built-in with the bluetooth hydrometer that we use for this purpose. We did not implement any extra data processing algorithm for this subsystem.

4.6 Pump System

This subsystem is the one that pumps the chilled glycol into the reservoir that contains the fermented beer. The transfer occurs through special types of pipes that facilitate the transfer of glycol.

4.6.1 Subsystem Hardware

As of now, assuming pump uses a DC power source and does have not have Bluetooth it will need a transistor-switch circuit and a pin used from the Raspberry Pi 3 to turn the pump on/off.

4.6.2 Subsystem Operating System

The system requires Raspbian OS in order to be able to compile and run the python libraries to drive the transistor circuit to turn the pump on/off.

4.6.3 Subsystem Software Dependencies

This subsystem will be dependent on the program written to control the specific GPIO pin, which is connected to the base of a transistor in the hardware circuit, and the software determines when to turn the pump on/off by toggling the GPIO pin in the Raspberry pi.

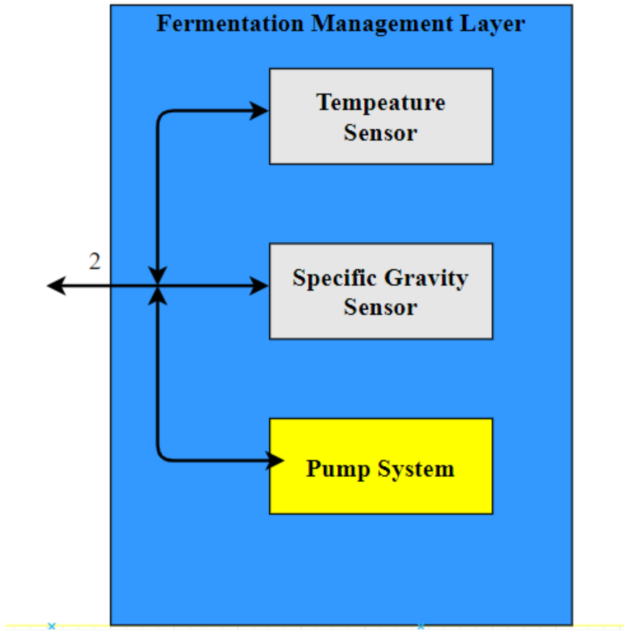


Figure 5: Pump System subsystem description diagram

4.6.4 Subsystem Programming Languages

The System uses Python in order to be programmed to drive the transistor circuit used to turn the pump on/off.

4.6.5 Subsystem Data Structures

The only data structure used is the bit corresponding to the GPIO pin used to drive the pump transistor-switch circuit.

4.6.6 Subsystem Data Processing

The only software requirement for implementing this subsystem is to toggle a Raspberry Pi GPIO pin connected to the base of transistor in the circuit ON and OFF in order to control the pump. Therefore, no data processing is involved in this subsystem.

5 Control Layer Subsystem

This section describes the details of the Control Layer. The subsystems that we have determined to make up the input section of the control layer include the manual control Touchscreen Display, the smart phone Mobile Application, and Recipe File Reading Subsystems. Each of these subsystems are different methods of inputting commands towards the controller application subsystem that will be running on the Raspberry Pi.

5.1 Layer Hardware

Raspberry pi 3 computer system along with a touchscreen to allow for user input and an easy to use display.

5.2 Layer Operating System

The default raspbian operating system will be used since it perfectly fulfills our needs as a generic python environment with Bluetooth

5.3 Layer Software Dependencies

We will be incorporating the pytilt python library in our control script to communicate directly with the Bluetooth hydrometer that acts as both our temperature sensor and specific gravity sensor and the tiny-tuya python library to control and communicate with the AC power controller. In order for both of these libraries to work, we have also installed the python BLE library to allow those libraries to connect the to the Bluetooth module on the pi.

5.4 Programming Languages

The main programming language for our script is python, however we may end up also using a bash shell scripts just to stitch everything together.

5.5 Data Structures

The Bluetooth tilt hydrometer sends a message structure every time a substantial change in it's measurements occurs that includes color of the hydrometer, gravity, date, and temperature. The tiny-tuya library has a control object that allows you call functions associated with the Bluetooth plug such as "on" and "off" functions. Those are our main data structures that we interact with and more information can be found on their websites which we have listed in the appendix.

5.6 Data Processing

The main data processing of this layer is a loop in which the data from the hydrometer is read. On new input the controller makes a decision to turn off or on the ac unit and a decision on the pump speed of the glycol. It then logs the hydrometers message data for future use in recipe calculations (not yet implemented). If the time period for the recipe is reached or the temperature is too cold the pi should automatically turn off the system.

5.7 Touchscreen Display Subsystem

The first implemented subsystem will be a touchscreen display mounted to the Glycol Chiller system and connected directly to the Raspberry Pi control unit. This subsystem serves as the primary method for user input via manual control and communication between the different layers. To do this, we are using a Raspberry Pi Touch Display for easier integration of connecting our Raspberry Pi 3 to the display,

5.7.1 Assumptions

This assumes the controller application has a GUI displayed on the touchscreen that can be used to easily string commands. Additionally, it is assumed that commands initiated through the touchscreen

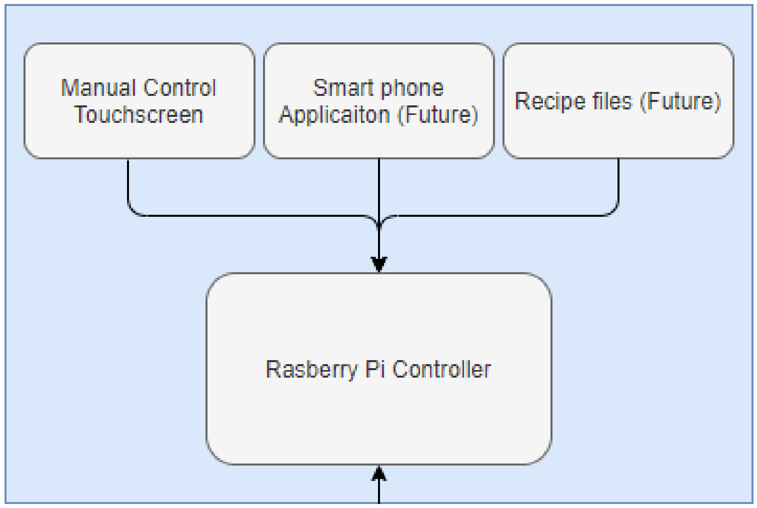


Figure 6: Control Layer subsystem description diagram

display will activate functions within the Fermentation Management Layer and the Glycol Management Layer.

5.7.2 Responsibilities

The main two responsibilities of this subsystem will be to handle manual user input by forwarding it to controller application on the Raspberry Pi and display the controller application's output. The screen will receive input via touchscreen gestures on a GUI display that will send commands to the control system.

5.7.3 Subsystem Interfaces

Table 2: Subsystem interfaces

ID	Description	Inputs	Outputs
#01	Description of the interface/bus	Touchscreen based commands	Commands to the controller application via a GPIO connector Status display

5.8 Mobile Application Subsystem

The secondary user input method will be a software application made for smartphones that will forward commands over the internet to the controller application. Controls would be similar to that of the system's built-in display but on a mobile user interface.

5.8.1 Assumptions

This assumes that the smartphone's Internet connection is stable enough to submit coherent commands, assumes that the user has a smartphone at all, and lastly assumes that the controller application has a GUI that can be displayed on the smartphone.

5.8.2 Responsibilities

This subsystem is responsible for relaying commands to the controller application. It is also responsible for displaying the results of those commands and the GUI interface required to input those commands. In this way it will be very similar to the touchscreen subsystem but just on a mobile application.

5.8.3 Subsystem Interfaces

Table 3: Subsystem interfaces

ID	Description	Inputs	Outputs
#01	TCP packets sent over LTE to the controller application	Touchscreen based commands	Commands sent in packets to the control application

5.9 Recipe File Reading Subsystem

The last input method will be customized recipe files that the control unit will use to completely automate the fermentation process.

5.9.1 Assumptions

This assumes the user has an easy way to place recipe files on the Raspberry pi or to create them within our controller application's GUI. The communication route would probably require a thumb drive or an Ethernet connection of some kind.

5.9.2 Responsibilities

This subsystem is responsible for saving fermentation recipes and the ability to parse those recipes to be passed to the controller application.

5.9.3 Subsystem Interfaces

Table 4: Subsystem interfaces

ID	Description	Inputs	Outputs
#01	Logical Bus Connection	Recipes XMLs	Recipe instructions passed to the controller application

5.9.4 Controller Application

The controller application itself that be in charge of managing the other layers.

5.9.5 Assumptions

-assumes communication between layers happens perfectly.

5.9.6 Responsibilities

This subsystem is responsible for communication between the different layers and any level of automation that will be implemented.

5.9.7 Subsystem Interfaces

Table 5: Subsystem interfaces

ID	Description	Inputs	Outputs
#01	GPIO wires	Touchscreen	Fermentation and Glycol layer control signals
#02	Description of the interface/bus	Mobile Application	Fermentation and Glycol layer control signals
#03	Description of the interface/bus	Recipe Files	Fermentation and Glycol layer control signals

6 APPENDIX A

Include any additional documents (CAD design, circuit schematics, etc) as an appendix as necessary.

Libraries: https://pypi.org/project/tinytuya/ https://github.com/atlefren/pytilt

REFERENCES