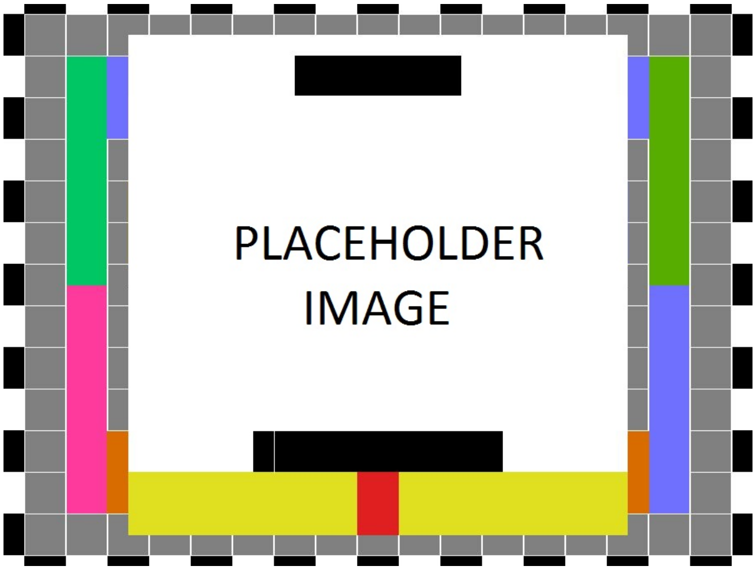
Department of Computer Science & Engineering
The University of Texas at Arlington

Architectural Design Specification
CSE 4316: Senior Design I
Fall 2015



Team Name
Product Name

Alan Turing
Grace Hopper
John Von Neumann
Ada Lovelace
Charles Babbage

REVISION HISTORY

Revision	Date	Author(s)	Description
0.1	10.01.2015	GH	document creation
0.2	10.05.2015	AT, GH	complete draft
0.3	10.12.2015	AT, GH	release candidate 1
1.0	10.20.2015	AT, GH, CB	official release
1.1	10.31.2015	AL	added design review requests

Contents

List of Figures

List of Tables

1 Introduction

The "Back Burner Brew" is built with the sole purpose of brewing large batch of beer in the home environment. This product provides home brewers with a low-cost electric home brewing system that allow them to have precise control over the brewing process. The brewing process can be automated with the help of micro-controllers like the ESP32 which is then hosted to a local website or an app interface.

ESP32 is a micro-controller that can receive data such as current temperature of the water or mash from the heat sensors located inside the kettles which can be converted to either analog or digital input. The heating coil can be controled using the input from the user as per their desired either to increase or to decrease the temperature. The electric pump can also be controlled by the user through micro-controllers to regulate the flow of the water in the kettles. The user will be able to communicate with the brewing system through a web interface or app interface.

The user should expect to input desired commands, controls, and specific settings such as temperature and length of time by a easily accessible touchscreen. The touchscreen will be attached to a Raspberry Pi that will handle communications between the user and the various sensors and heating elements. The user can expect that whichever temperature they set for their desired application, that the temperature will remain constant.

The intended audiences for this product would be home brewers or person interested in brewing beer only. Provided that the user manual would be present in the product, any person who wants to brew beer in his local environment can easily use this product. This product is made focusing on how effortless can the brewing process gets simply with the use of micro-controller.

2 System Overview

This section should describe the overall structure of your software system. Think of it as the strategy for how you will build the system. An architectural "layer" is the top-level logical view, or an abstraction, of your design. Layers should be composed of related elements of similar capabilities, and should be highly independent of other layers, but should have very clearly defined interfaces and interactions with other layers. Each layer should be identified individually and should be unique as to its function and purpose within the system. This section should also contain the high-level block diagram of the layers, as shown in the example below, as well as detailed descriptions of the functions of each layer.

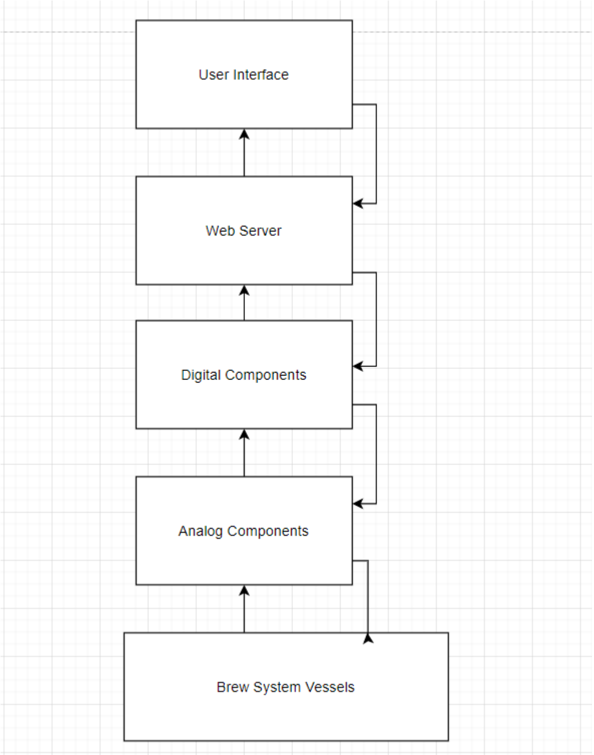


Figure 1: Simple ADS

2.1 Brew System Vessel

Each layer should be described separately in detail. Descriptions should include the features, functions, critical interfaces and interactions of the layer. The description should clearly define the services that the layer provides. Also include any conventions that your team will use in describing the structure: naming conventions for layers, subsystems, modules, and data flows; interface specifications; how layers and subsystems are defined; etc.

2.2 Layer Y Description

Each layer should be described separately in detail. Descriptions should include the features, functions, critical interfaces and interactions of the layer. The description should clearly define the services that the layer provides. Also include any conventions that your team will use in describing the structure: naming conventions for layers, subsystems, modules, and data flows; interface specifications; how layers and subsystems are defined; etc.

2.3 Layer Z Description

Each layer should be described separately in detail. Descriptions should include the features, functions, critical interfaces and interactions of the layer. The description should clearly define the services that the layer provides. Also include any conventions that your team will use in describing the structure: naming conventions for layers, subsystems, modules, and data flows; interface specifications; how layers and subsystems are defined; etc.

3 Subsystem Definitions & Data Flow

This section breaks down your layer abstraction to another level of detail. Here you grapically represent the logical subsytems that compose each layer and show the interactions/interfaces between those subsystems. A subsystem can be thought of as a programming unit that implements one of the major functions of the layer. It, therefore, has data elements that serve as source/sinks for other subsystems. The logical data elements that flow between subsystems need to be explicitly defined at this point, beginning with a data flow-like diagram based on the block diagram.

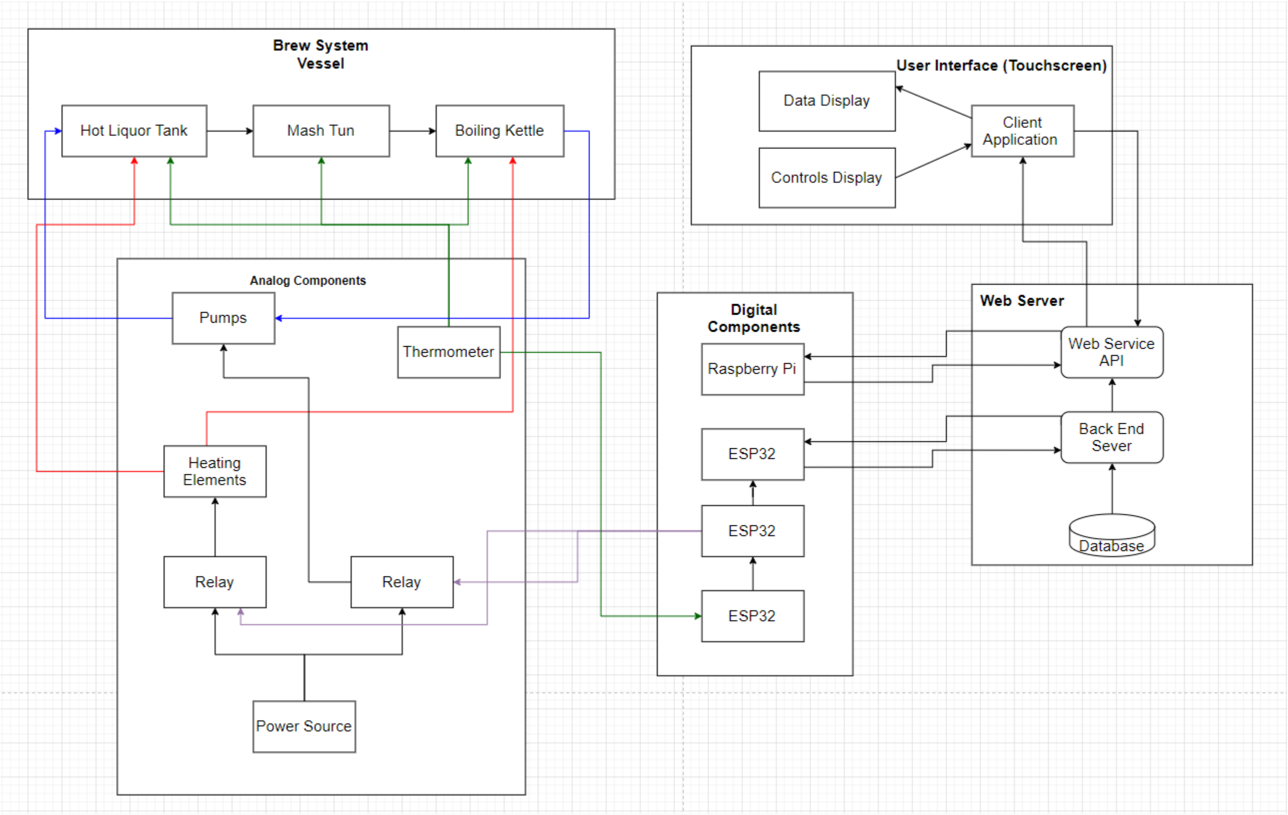


Figure 2: Subsystem ADS

4 Brew System Vessel Layer Subsystems

In this section, the layer is described in some detail in terms of its specific subsystems. Describe each of the layers and its subsystems in a separate chapter/major subsection of this document. The content of each subsystem description should be similar. Include in this section any special considerations and/or trade-offs considered for the approach you have chosen.

The Brewing Vessel layer consists of three major components which plays a vital role in each part for maintaining a good brew. This is the initial phase and the most important layer which should be observed properly. Any mistake made on this layer can drastically affects the taste of beverages. Brew System Vessel layer is directly connected to the some of subsystems from the Analog Components layer which help to regulate the Brewing process. "Brew in A Bag", an alternative for this system was considered which would only contains two subsystems; however, the system chosen in this design provided users with better control and evaluation on overall brewing process. The three subsystems of this layer are explained below:

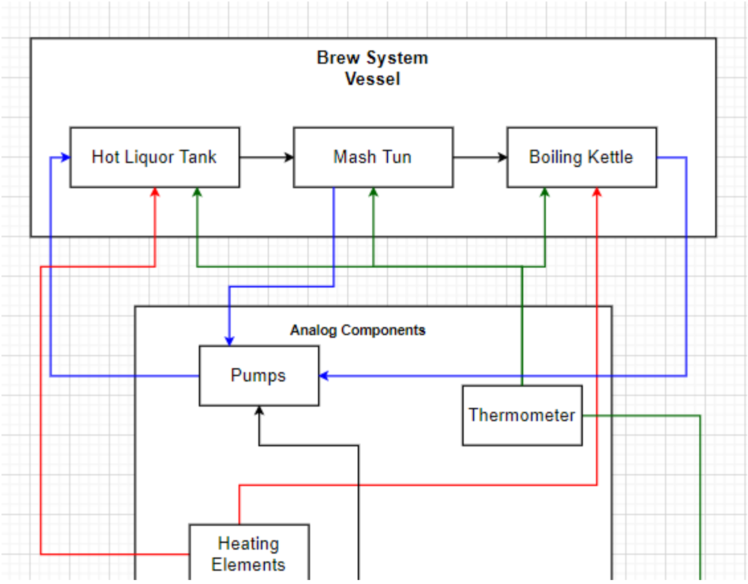


Figure 3: Brew System Vessel Layer Representation

4.1 Hot Liquor Tank Subsystem (HLT)

This section should be a general description of a particular subsystem for the given layer. For most subsystems, an extract of the architectural block diagram with data flows is useful. This should consist of the subsystem being described and those subsystems with which it communicates.

The main purpose of Hot Liquor Tank subsystem is to monitor and heat the water to the desired temperature. This subsystem is directly connected to three subsystems from Analog Components layer which are pumps, thermometer, and heating elements.

4.1.1 Assumptions

Any assumptions made in the definition of the subsystem should be listed and described. Pay particular attention to assumptions concerning interfaces and interactions with other layers.

In the initial phase, the vessel should be filled with water by the brewer. The subsystem is controlled by the micro-controller based on the current temperature of the water measured from thermometer and sent through micro-controller to the user.

4.1.2 Responsibilities

Each of the responsibilities/features/functions/services of the subsystem as identified in the architectural summary must be expanded to more detailed responsibilities. These responsibilities form the basis for the identification of the finer-grained responsibilities of the layer's internal subsystems. Clearly describe what each subsystem does.

The Hot Liquor Tank subsystem is responsible for providing informations such as current temperature of the water to the brewer, current status of pump and heating elements. Then the subsystem should heat the water to the desired temperature as per the brewer wants. When the water is reached to desired temperature, then it is responsible to pass the water to the Mash Tun subsystem.

4.1.3 Subsystem Interfaces

Each of the inputs and outputs for the subsystem are defined here. Create a table with an entry for each labeled interface that connects to this subsystem. For each entry, describe any incoming and outgoing data elements will pass through this interface.

Table 2: Hot Liquor Tank Subsystem interfaces

ID	Description	Inputs	Outputs
#01	Thermometer Subsystem	User input to display temperature	Current Temperature of the water
#02	Pump Subsystem	User input collected from the micro controller	Open/Close the pump based on the input
#03	Heating Element Subsystem	User input in temperature collected from the micro controller	Turn on/off heating elements in order to reach user desired temperature

4.2 Mash Tun Subsystem

The main purpose of Mash Tun subsystem is to mix the hot water with grains to produce wort. This subsystem is directly connected to Thermometer and Pump subsystems from Analog Components layer which is used to provide temperature of the mash.

4.2.1 Assumptions

When the Mash Tun is filled with hot water from the HLT, then the Mash Tun is manually loaded with the crushed grains. The user is supposed to set the temperature and time through web or app interface he want the mash to be in this subsystem. If the temperature inside the Mash Tun reached to the different temperature than the user set, the water is sent back to HLT and heated. The mash is rinse several time to rinse the sugar from the mash which is then passed to Boiling Kettle subsystem.

4.2.2 Responsibilities

The Mash Tun is responsible to remove sugars, mostly maltose, sucrose and maltotriose from the crushed grains to produce the final product called wort, also known as un-fermented beer. The Mash Tun should provide temperature of the current mash to the brewer. The Mash Tun is responsible to pump out the water to HLT Subsystem when it doesn't maintain required temperature. And when the water is heated enough, then it is again passed to Mash Tun subsystem. By this way, the mash is rinse to get the sugars out. The wort obtained is then passed to Boiling Kettle subsystem in this layer.

4.2.3 Subsystem Interfaces

Table 3: Mash Tun Subsystem interfaces

ID	Description	Inputs	Outputs
#01	Thermometer Subsystem	User input to display and set temperature	Current Temperature of the mash
#02	Pump Subsystem	User input collected from the micro controller	Open/Close the pump based on the temperature of the mash

4.3 Boiling Kettle Subsystem

The main purpose of Boiling Kettle subsystem is to boil wort for the required amount of time. This subsystem is directly connected to Thermometer, Pump, and Heating elements subsystems from Analog Components layer which is used to maintain the amount of heat provided to the wort.

4.3.1 Assumptions

The Boiling Kettle subsystem is almost an automated process. The brewer have to add some hops after each beer cycle in the wort as needed because it add bitterness to the beer. The brewer must be careful with adding appropriate amount of hops as too much can ruin the taste and very little won't bring any taste to beer.

4.3.2 Responsibilities

The Boiling Kettle is only responsible for heating the wort with appropriate heat for the appropriate amount of time. The wort should be heated properly to maintain the good flavor of beer. Once the boil is done, the Boiling Kettle subsystem is responsible for sending beer to the chiller for fermentation.

4.3.3 Subsystem Interfaces

Table 4: Boiling Kettle Subsystem interfaces

ID	Description	Inputs	Outputs
#01	Thermometer Subsystem	User input to display temperature	Current Temperature of the wort
#02	Pump Subsystem	User input collected from the micro controller	Open/Close the pump based on the input
#03	Heating Element Subsystem	User input in temperature collected from the micro controller	Turn on/off heating elements in order to reach user desired temperature

5 Y Layer Subsystems

In this section, the layer is described in some detail in terms of its specific subsystems. Describe each of the layers and its subsystems in a separate chapter/major subsection of this document. The content of each subsystem description should be similar. Include in this section any special considerations and/or trade-offs considered for the approach you have chosen.

5.1 Subsystem 1

This section should be a general description of a particular subsystem for the given layer. For most subsystems, an extract of the architectural block diagram with data flows is useful. This should consist of the subsystem being described and those subsystems with which it communicates.

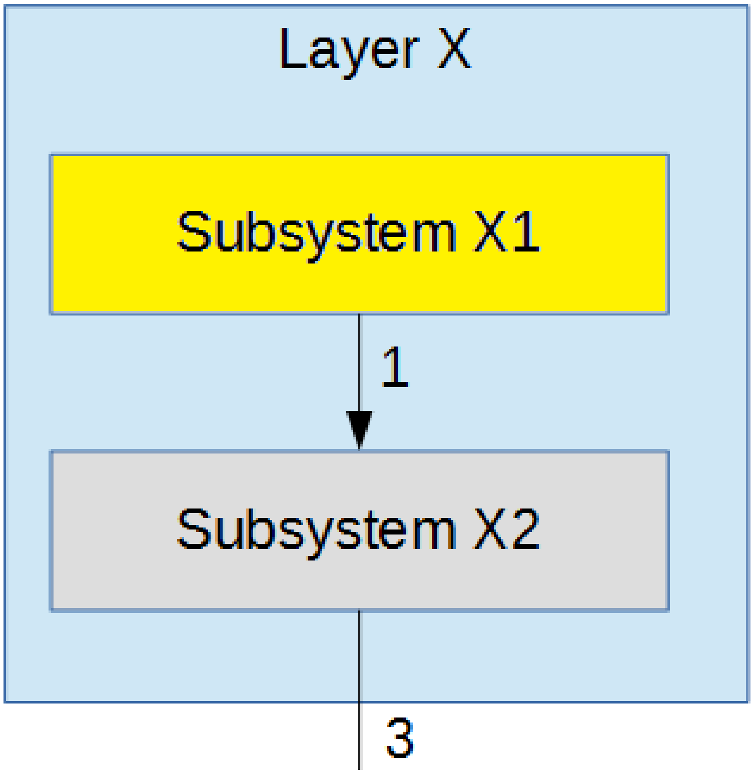


Figure 4: Example subsystem description diagram

5.1.1 Assumptions

Any assumptions made in the definition of the subsystem should be listed and described. Pay particular attention to assumptions concerning interfaces and interactions with other layers.

5.1.2 Responsibilities

Each of the responsibilities/features/functions/services of the subsystem as identified in the architectural summary must be expanded to more detailed responsibilities. These responsibilities form the basis for the identification of the finer-grained responsibilities of the layer's internal subsystems. Clearly describe what each subsystem does.

5.1.3 Subsystem Interfaces

Each of the inputs and outputs for the subsystem are defined here. Create a table with an entry for each labelled interface that connects to this subsystem. For each entry, describe any incoming and outgoing data elements will pass through this interface.

Table 5: Subsystem interfaces

ID	Description	Inputs	Outputs
#xx	Description of the interface/bus	input 1 input 2	output 1
#xx	Description of the interface/bus	N/A	output 1

5.2 Subsystem 2

Repeat for each subsystem

5.3 Subsystem 3

Repeat for each subsystem

6 Z Layer Subsystems

In this section, the layer is described in some detail in terms of its specific subsystems. Describe each of the layers and its subsystems in a separate chapter/major subsection of this document. The content of each subsystem description should be similar. Include in this section any special considerations and/or trade-offs considered for the approach you have chosen.

6.1 Subsystem 1

This section should be a general description of a particular subsystem for the given layer. For most subsystems, an extract of the architectural block diagram with data flows is useful. This should consist of the subsystem being described and those subsystems with which it communicates.

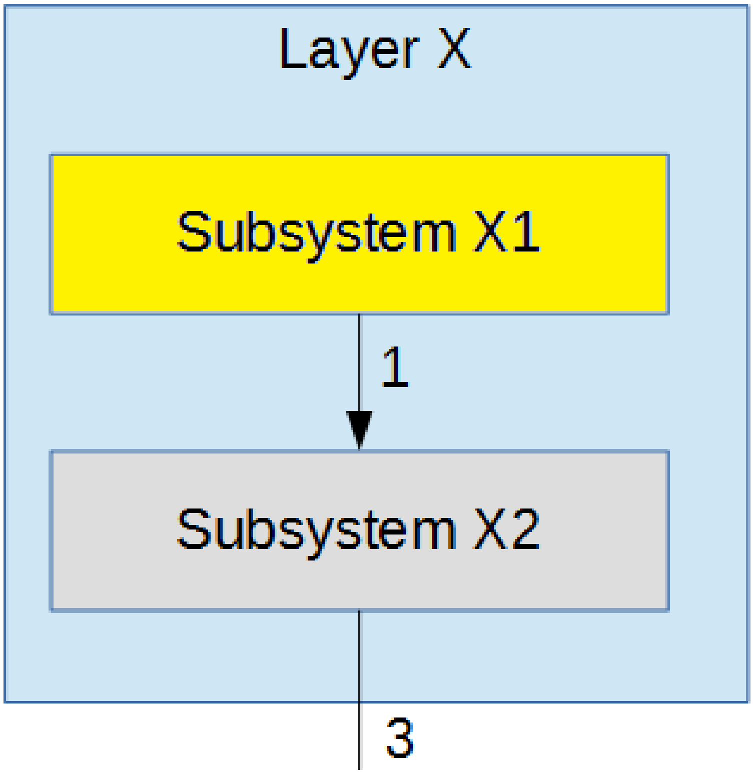


Figure 5: Example subsystem description diagram

6.1.1 Assumptions

Any assumptions made in the definition of the subsystem should be listed and described. Pay particular attention to assumptions concerning interfaces and interactions with other layers.

6.1.2 Responsibilities

Each of the responsibilities/features/functions/services of the subsystem as identified in the architectural summary must be expanded to more detailed responsibilities. These responsibilities form the basis for the identification of the finer-grained responsibilities of the layer's internal subsystems. Clearly describe what each subsystem does.

6.1.3 Subsystem Interfaces

Each of the inputs and outputs for the subsystem are defined here. Create a table with an entry for each labelled interface that connects to this subsystem. For each entry, describe any incoming and outgoing data elements will pass through this interface.

Table 6: Subsystem interfaces

ID	Description	Inputs	Outputs
#xx	Description of the interface/bus	input 1 input 2	output 1
#xx	Description of the interface/bus	N/A	output 1

6.2 Subsystem 2

Repeat for each subsystem

6.3 Subsystem 3

Repeat for each subsystem

References