ABET Example Low
B+

Electronic Bartender

The electronic bartender is an automated bar that makes and serves mixed drinks. Modification is easy to cater for any type of drink with a simple interchange of ingredients hidden within the machine. With this contraption, serving drinks at parties and get-togethers can be made easy with the simple push of a button.

Andrew Hung
Steven Lee

EECS 129A: Senior Design
Prof. Beatriz Da Costa

EECS 129B: Senior Design
Dr. Raymond Klefstad

Description

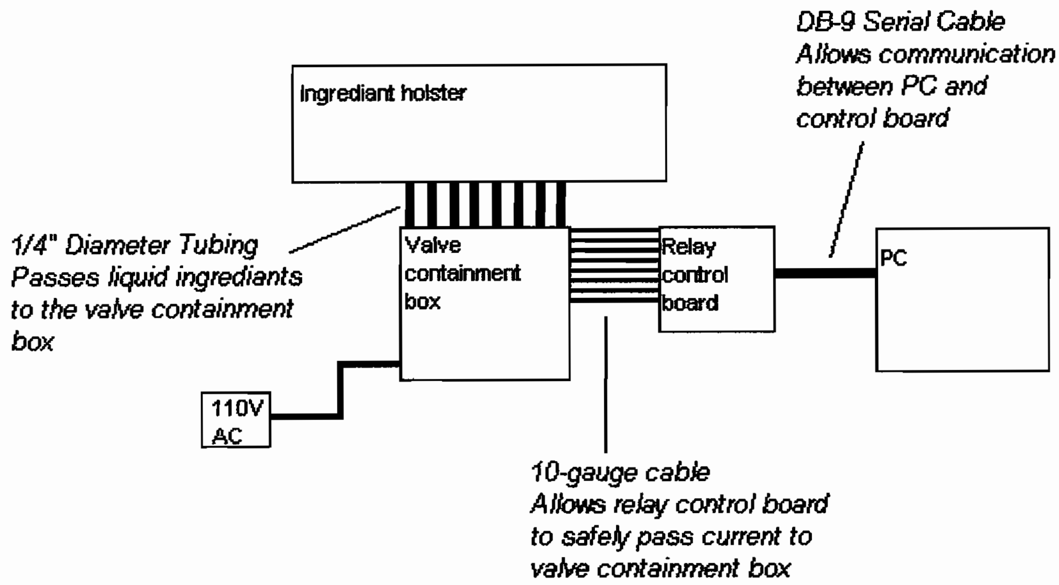
The electronic bartender is a machine that automatically mixes and dispenses alcoholic beverages. Originating from the idea by four college students, the automated bar is aimed more towards the use of large parties. The employment of this machine can greatly cut down on the worry of how to make certain drinks as well as the hassle to pour, mix, and serve them. It can also provide more of a variety to how many different drinks a person can try instead of being limited to what drinks the host knows how to make. Many times, a party only serves a variety of 3-5 different drinks. With the electronic bartender, the range of drinks is only limited by amount of ingredients and one's imagination.

On the program level, the software of this machine is fairly simple. At the forefront of the program, the user should first see a page asking whether he/she would like to create his/her own drink or choose a preset drink stored in the program database. If the user chooses to create their own drink, a list of the available ingredients should be presented on the next screen. Once the preferred ingredients and quantity (in ounces) of each ingredient is chosen, the user finalizes the order by selecting to make the drink. Once this is done, the ingredients for the mixed drink should be dispensed from the appropriate valves. If the user chooses to select one of the preset combinations, a screen appears with a list of all the names of different mixed drinks with ingredient details. Once a certain drink is chosen, the beverage is poured. At the end of each cycle of use, the program will check to see if any ingredient is nearly empty. If so, a pop-up will appear to warn that it is in need of a refill for one of its ingredients and discontinue any use from then on until the ingredient has been refilled. Implementation of this software scheme will be done with Visual Basic .NET 2005 Express edition for communication between the machine and control board.

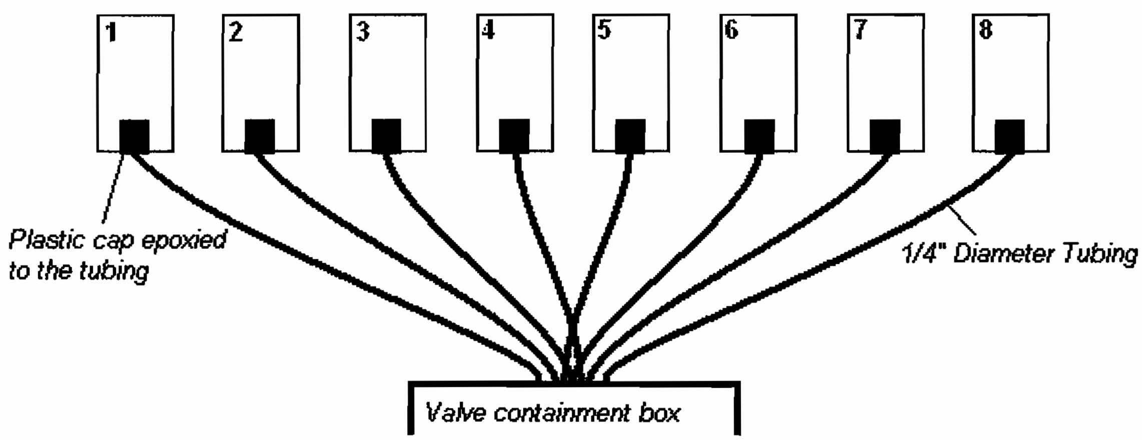
Easily programmed to take in any order of drinks, the automated bar also features easy interchangeability with the ingredients that it uses in terms of types of alcohol and fruit mixers. Liquid ingredients are attached to tubes leading to the valve where the drink is poured. When a drink is ordered, a signal is sent to its respective valves leading to the bottled ingredients, and releases the instructed amount. The bottles holding the liquor and mixers have easy-to-remove caps to ensure quick and simple substitution or replacement if necessary. Also, in terms of the preset combinations, the drink database is simple to update and easily expandable. Maintenance of the machine is also non-complex, with each part being easily reachable, detachable and cleanable.

Targeting young adults, specifically collegians of legal drinking age, the electronic bartender is something to be greatly considered for an easy, more care-free party organization. Ideal because of its mobility and ease of use, this machine features more to parties than what can be considered enough for most.

System Level Block Diagram

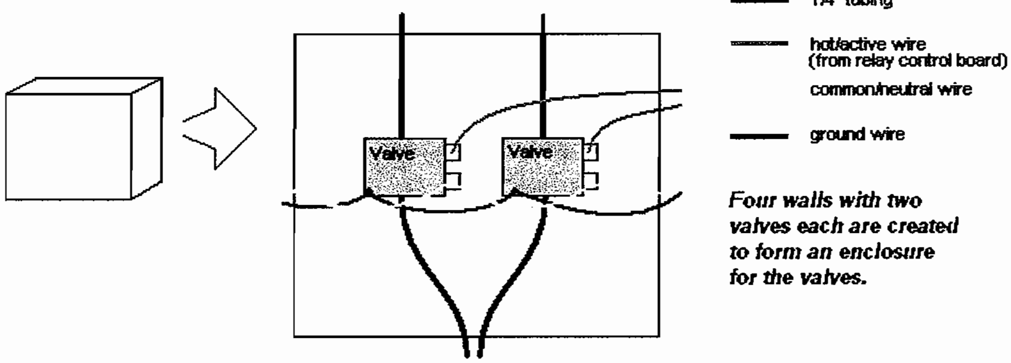


Ingrediant Holster



The ingrediant holster simply holds the 8 different ingrediants. The ingrediants include alcohol, soda, and any other mixer you desire. The liquid is pulled down by gravity so each container is placed upside down.

Valve Containment Box

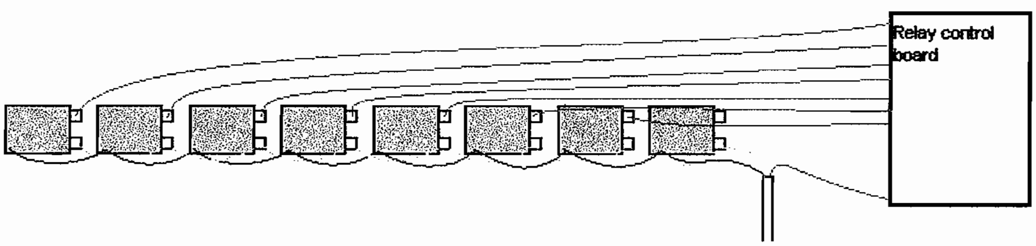


Four walls with two valves each are created to form an enclosure for the valves.

Wiring of all valves:

Each valve has its ground and neutral daisy chained from the previous valve. The last valve does not continue the chain. The first valve gets the ground and neutral connection directly from a single 120V 3-prong outlet cord that has had its wires exposed.

Each valve gets its power from a hot/active wire originating from the relay control board. The relay control board gets a 120VAC supply from the outlet cord. When a valve is supposed to open, the relay control board completes the connection between the valve and the 120VAC supply. A description of the relay control board is provided later in the report.

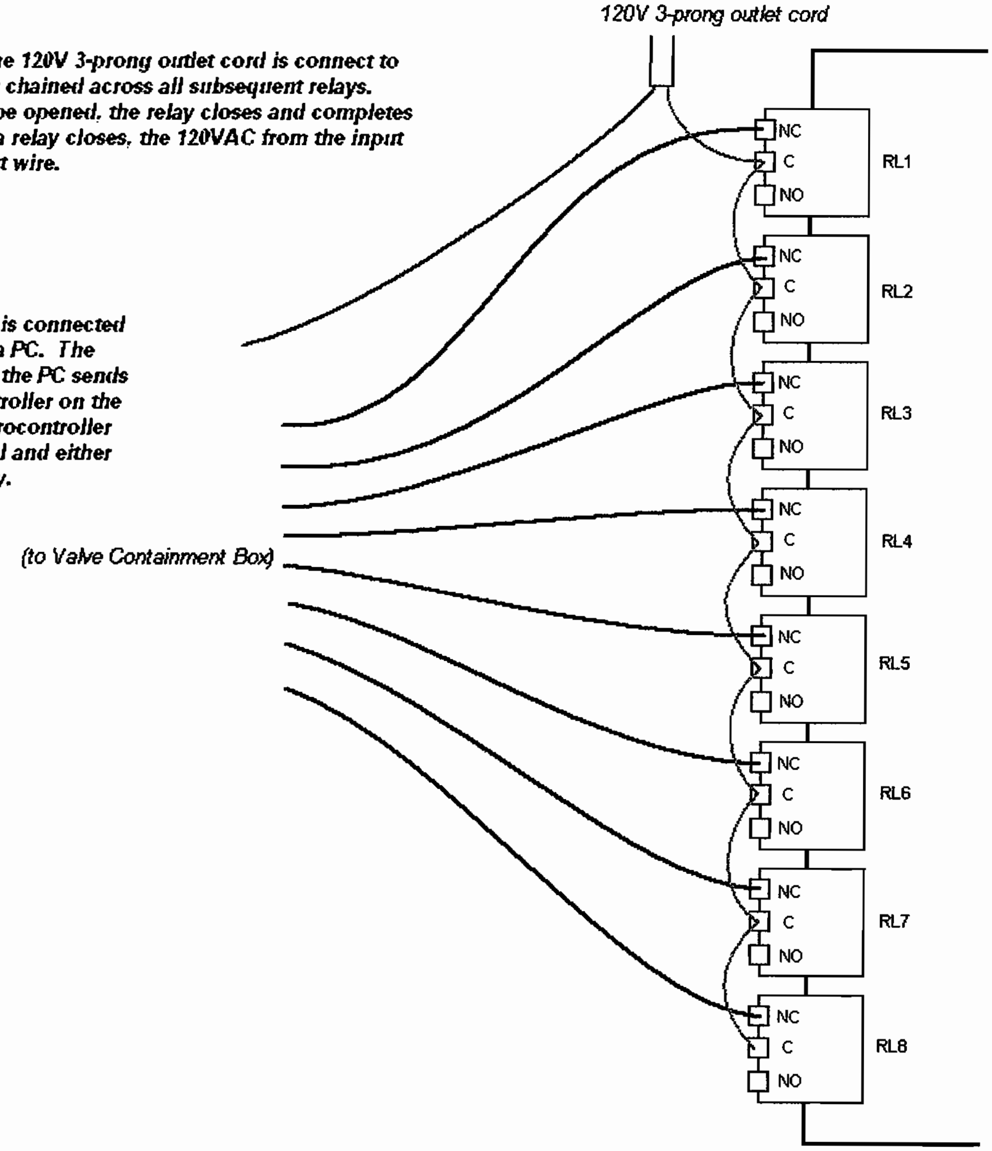


Relay Control Board

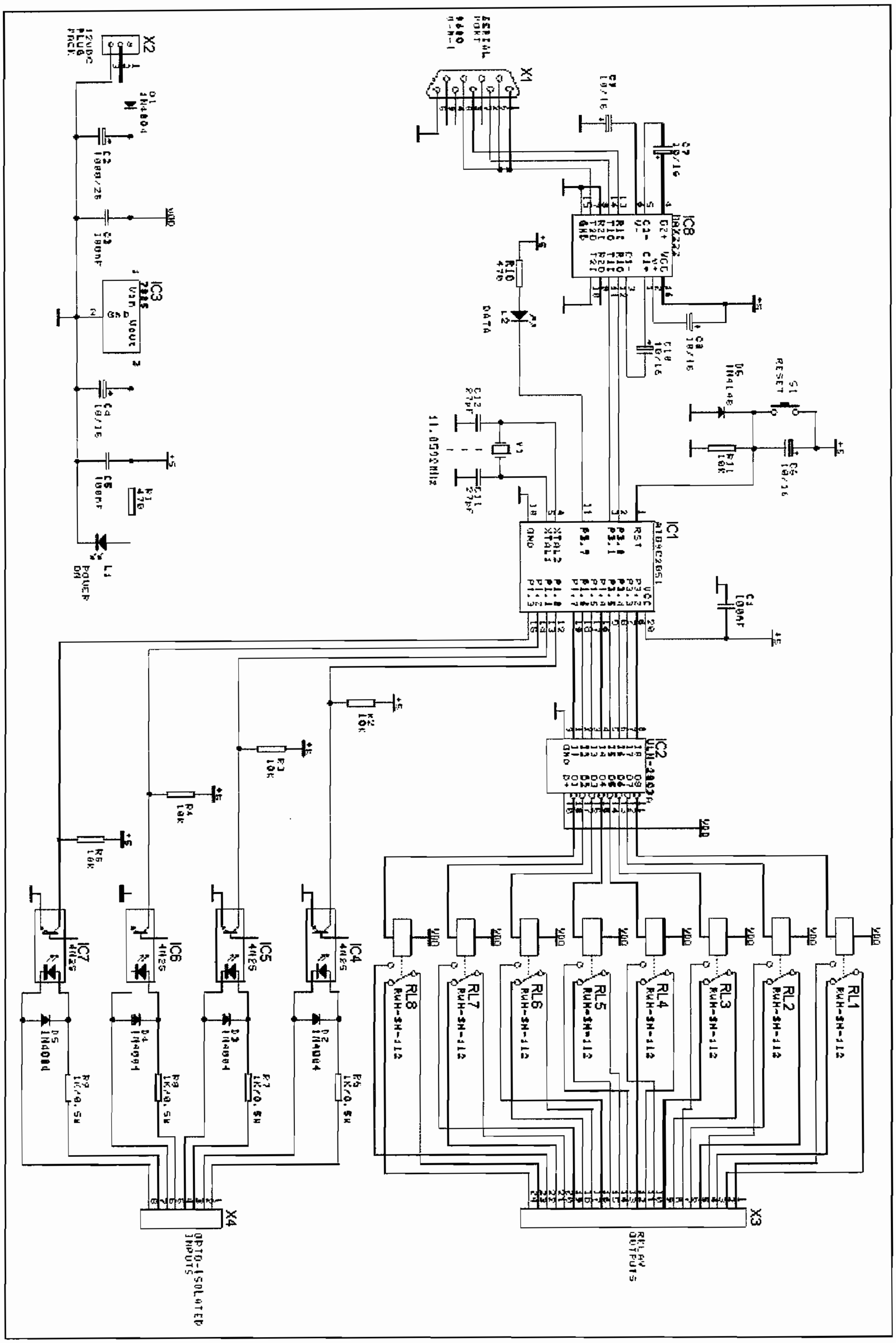
The hot/active wire of the 120V 3-prong outlet cord is connect to the first relay then daisy chained across all subsequent relays. When a valve needs to be opened, the relay closes and completes the connection. When a relay closes, the 120VAC from the input wire is sent to the output wire.

——— input wire
——— output wire

The relay control board is connected by a DB-9 serial port to a PC. The customized software on the PC sends signals to the microcontroller on the control board. The microcontroller then interprets the signal and either closes or opens the relay.



Electronic Schematic of relay control board



PART LIST – KIT 108		
Resistors (0.25W unless specified)		
470R	R1,10	2
1K, 0.5W	R6,7,8,9	4
10K	R2,3,4,5,11	5
Capacitors		
27pF ceramic	C11,12	2
100nF monobloc	C1,3,5	3
10uF 16V electro	C4,6,7,8,9,10	6
1000uF 25V electro	C2	1
Semiconductors		
1N4004 diode	D1,2,3,4,5	5
1N4148	D6	1
4N25 opto-coupler IC	IC4,5,6,7	4
AT89C2051 uC	IC1 pre-programmed	1
ICL232 IC RS232 driver/receiver	IC8	1
ULN2803A Octal open collector driver	IC2	1
7805 regulator, TO-220	IC3	1
LED, panel mounting 3mm red	L1,2	2
Miscellaneous		
Crystal, 11.0592MHz	Y1	1
D9 connector PCB mounting, female	X1	1
Relay, SPDT "Goodsky" RWH-SH-112D	RL1,2,3,4,5,6,7,8	8
2.5mm DC jack	X2	1
Terminal socket 8 way, PCB mtg, "Dinkle" 2EHDRC-08P	X4	1
Terminal socket 24 way, PCB mtg, "Dinkle" 2EHDRC-24P	X3	1
Terminal plug, 3 way, to fit "X4" "Dinkle" 2ESDV-03P		8
Terminal plug, 2 way, to fit "X3" "Dinkle" 2ESDV-02P		4
Heatsink, to fit "IC3"		1
Pushbutton, panel mtg		1
IC socket, 6 pin, for "IC4,5,6,7"		4
IC socket, 16 pin, for "IC8"		1
IC socket, 18 pin, for "IC2"		1
IC socket, 20 pin, for "IC1"		1
Screw, 3 x 8mm, to fit heatsink to "IC3"		1
Nut, 3mm, to fit heatsink to "IC3"		1
Self tapping screws for mounting PCB		4
Plastic case, 140(W) x 110(D) x 35(H)mm		1
PCB, K108V2		1
Set of front & rear panels		1
Hookup wire, twin, 36cm (14")		

- X1: Serial Port 9680 8-N-1
    - DB-9 serial port input that is used to receive and send data between the microcontroller and PC.

- IC8: MAX232
    - Provides conversion between TTL signals and RS-232 signals.
    - TTL signals are transistor to transistor logic signals.
    - RS-232 is a standard for serial binary data interconnection
    - This allows the microcontroller to interpret RS-232 signals

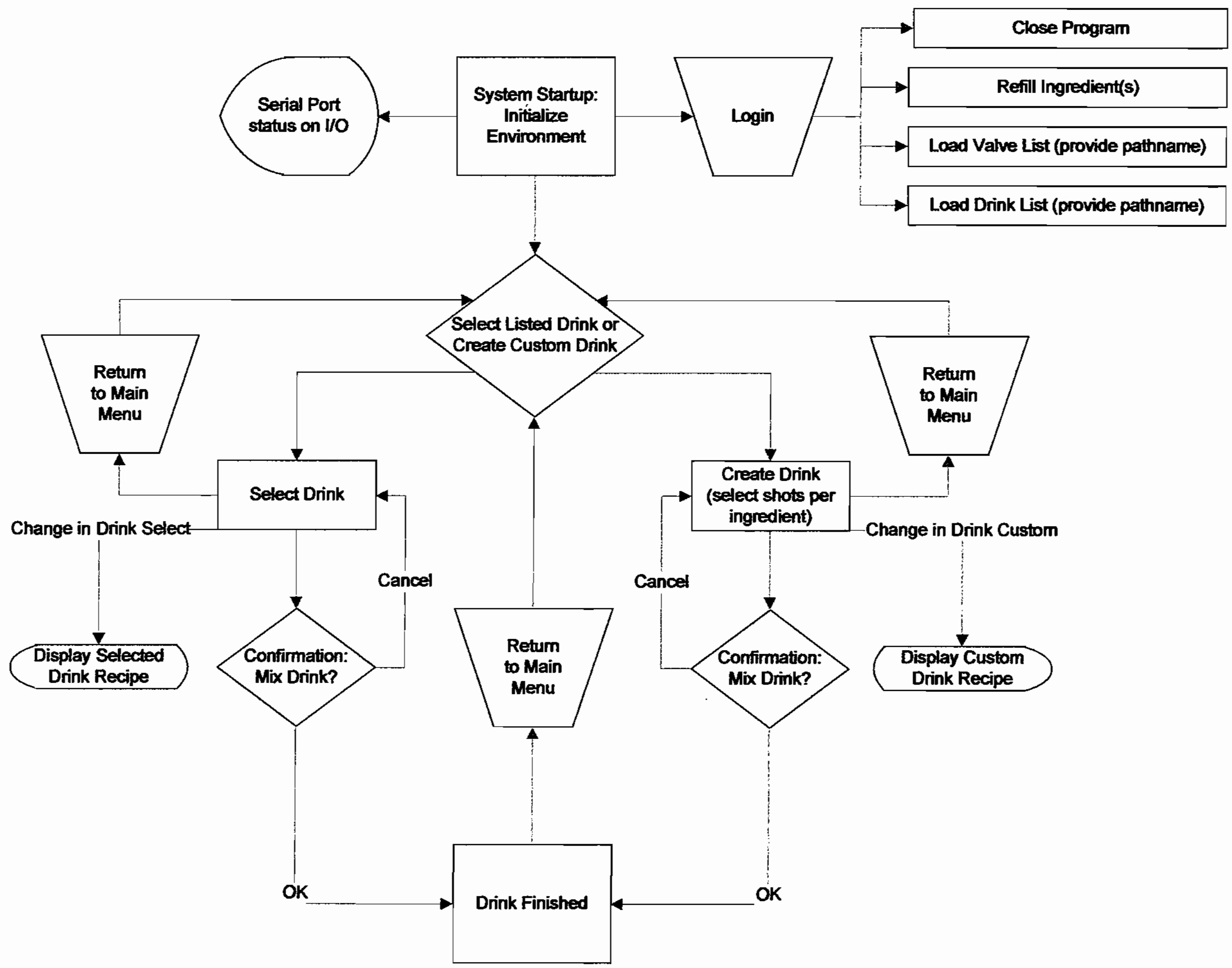
- IC1: AT89C2051
    - Microcontroller from Atmel.
    - It has I/O pins and built in serial port.
    - It is preprogrammed to process commands received from the serial port, control the relays, and receive data from the inputs.
    - There are eight output pins that allow the microcontroller to set the relays
        - When specific relays want to be closed, the microcontroller sends the appropriate signal to IC2, which then drives the appropriate relays.

- IC2: VLN-2803A
    - An octal relay driver
    - It is used to drive each of the relays.

- RL1 to RL8: RWH-SH-112
    - Relays that complete the connection between a "common" voltage at C to either points at NC or NO

- X3: Relay outputs
    - Outputs that allow different devices to be connected to relays RL1-RL8

User-Level Flow Chart



Software Description

As shown in the previously provided flow chart, the program is initialized to start at the main menu without being logged into administrative control. In order upload a drink list text file or valve-ingredient association text file, the correct administrative username and password have to be entered into the login textboxes. Once that is provided, the administrator can also signal that a valve, or set of valves, has been refilled or close the program if necessary. General users do not have access to these controls but are still able to use the program. As part of the main functionality a user may select between selecting a drink from a pre-devised menu and creating a drink of their own by selecting certain ingredients to mix.

If a user clicks on the link to choose a drink from a menu, the user is directed to another screen containing a list of different drinks with a text window to show the liquid contents of the currently selected drink. The user can finalize his/her decision by clicking the button to mix the drink with a final confirmation window just to make sure they did not accidentally click the wrong button. If the user passes an okay to mix the drink, a serial command is communicated through the serial port to the microcontroller to open the valves necessary to create the drink. Once the valves are opened, an arbitrary increment of microseconds is timed to gauge how long each valve should stay open depending on how many "shots" are needed per ingredient/valve (example: 5 shots = 5secs). The program communicates to the microcontroller at each interval until the last valve is closed. Once the process is finished, the user is directed to another window notifying that his/her drink has been dispensed with a button to direct them back to the main menu if they desire to navigate the program more.

The alternate to selecting a drink would be for the user to create his/her own drink out of ingenuity. If he/she clicks on the link to customize their own drink, they are directed to a window containing all the available ingredients with a maximum cap of 5 shots per ingredient and 20 shots total per mixed drink. This is strictly for ingredient rationing and spilling prevention due to the overfilling of cups. A user can keep track of their recipe with a text window providing the amount of shots per ingredient as well as the total amount of shots for the whole drink. If the user tries to mix a drink over the limit, a notification window pops up advising the user to lower the shot total and cancels the drink order. However, if the user is within the shot limitations, the drink is mixed the same way as a drink selected from a menu would be. After the process is complete, the user is again, directed to the window notifying that their drink is done with a navigation button back to the main menu.

If at anytime any ingredient falls bellow the 5 shot buffer restriction (i.e. if an ingredient has 5 shots left before it is empty), a notification window will appear, specifying which ingredient of which valve needs to be refilled. Users will not be able to create or choose any drink using the nearly empty ingredient. The valve can only be reset when the administrator logs in and resets the refilled valve. Easy to use for both user and administrator, no one should have trouble maintaining the electronic bartender through this program.

System Test Plan:

Minimum Working Subset:

Subset consists of one liquid container containing one type of liquid, connected to one valve that will allow the liquid to be dispensed into one measuring cup. Measuring cup allow us to create measure the flow in amount of liquid per unit time. A minimal program will therefore have the ability to control one valve and dispense liquid.

Tests to ensure correct functionality of electronic bar:

1. Software tests
    a. After every drink is dispensed, the program will check the levels of ingredients by taking into account the starting amount and the amount already dispensed.
    b. After every drink is dispensed, the program will check if the microcontroller is responding by using a built-in command.
    c. The program should disallow any drink from being created if that drink requires ingredients that aren't available.

2. If the program shows that an ingredient is available, the user can visually confirm that the ingredient is actually available. Likewise, if the program shows that an ingredient is empty, then the user can confirm it visually.

3. A measuring cup can verify that correct amount of liquids has been dispensed.
    a. Each ingredient can be tested to ensure that the valves, tubing, and wiring are correctly performing and installed
    b. Combinations of several ingredients can be tested to ensure that the microcontroller is properly sending multiple signals. This also ensures that the program is correctly keeping track of the timing.

4. The valves should make a humming noise when activated. The user can acoustically verify that the correct valves are being activated without using any liquids or measuring cups. Likewise, the user can acoustically verify that the valves are being shut off at the correct time.
    a. Valves should responding almost instantaneously without any noticeable delay

5. A provided test program will allow the user to verify the functionality of the hardware.
    a. Once the hardware is verified, the performance of the hardware using the test program can be compared to the performance of the hardware using Electronic Bar program.

Cost analysis

- Laptop
  already obtained
  $0.00

- Visual Basic .NET 2005 Express edition
  Microsoft
  http://msdn.microsoft.com/vstudio/express/vb/
  1 unit x (0.00 each)
  $0.00

- Serial Port I/O Module
  (CK1610)
  Carl's Electronics
  http://www.electronickits.com/kit/complete/elec/ck1610.htm
  1 unit x (59.95 each)
  $59.95

- Icemaker Single Solenoid Water Valve
  Maytag
  http://www.cheapapplianceparts.com/maytag_ref_parts.htm
  8 units x (25.95 each)
  $207.60

- ¼" Polyethylene Tube
  Home Depot
  20 feet
  $3.00

- Cabinet
  Home Depot
  $50.00

Total: $320.55

Iterative Steps

- In order to dispense a drink, software must be written that can utilize an external controller that can operate a series of valves.
  - Idea is to use a Serial Port I/O Module to control a series of valves that are connected to tubing, which in turn are connected to different types of drinks.
- We possess the schematic of the Serial Port I/O Module.
  - The I/O module is the main component in our setup. It directly controls the valves that allow liquid to escape the container.
  - Serial Port I/O Module requires assembly which will include soldering capacitors, diodes, and different IC's to the PCB.
- There will be a valve for each liquid container.
  - The valves must be clean, and be able to handle consumable materials without risk of contamination.
  - Refrigerator water valves, theoretically, should be ideal because they pass drinkable water.
- Liquid is forced down via gravity
- Software is written that can control the Serial Port I/O Module
  - Current language in use: Visual Basic .NET 2005 Express edition
  - Software must be able to run on a Windows XP Professional SP2
- Serial Port I/O Module is a Kit 108 Serial Relay Kit
  - Requires 9-12VDC @ 500mA power supply
  - 8 relay outputs
    - Ratings: 120VAC @ 15A
  - Can be programmed via any language
- ¼" tubing used to connect liquid container to valves

Timeline

Start Date: January 3
End Date: March 16

January

- Week 1:
  - Order all parts
  - Design and plan program in Visual Basic
- Week 2:
  - Compile and test program
- Week 3:
  - Assemble internal liquid pipelining.
  - Assemble relay control board
- Week 4:
  - Assemble preliminary demo
  - Test basic program functionality using demo

February

- Week 5:
  - Continue development of program
  - Continue assembly of hardware
- Week 6:
  - Complete program
  - Complete hardware build
- Week 7:
  - Assemble any new components
  - Test and fix program/hardware bugs.
- Week 8:
  - Testing and modifications.

March

- Week 9:
  - Testing and modifications.
- Week 10:
  - Model finalization.

Task Division

Andrew Hung:

- Order all parts from their respective suppliers.
- Assemble internal pipelining
- Assemble valve enclosure
- Assemble relay control board
    - Solder various parts to a PCB
    - Ensure proper functionality of relay board
        - Verify communication between relay board, valves, and PC
- Assemble ingredient holster.
- Wire and solder all hardware
- Assist in developing software
    - Valve activation
    - Ingredient selection
    - Research other necessary methods

Steven Lee:

- Design a plan for software development
    - Basic functions
        - Single valve activation
        - Single ingredient selection
    - Advanced functions
        - Multiple valve activation with various time intervals
        - Multiple ingredient recipes
        - Customized selections
- Create and implement Visual Basic Program to communicate between machine and computer
- Create and implement Visual Basic GUI
- Conduct compilation, test, and debugging of program
- Assist in developing hardware platform
    - Valve enclosure assembly
    - Relay control board assembly
    - Ingredient holster assembly

Summary

The project worked exactly as we hoped it would. Our goal was to create something that could be used at parties or other get-togethers. The electronic bar allowed the user to create custom drinks, or to select a drink from pre-defined recipes. The user easily interfaced with the electronic bar via our software. Our hardware then successfully created the desired drink.

The hardware aspect of this project was straight-forward, but required some research on what type of valves to employ and some ingenuity regarding the mounting of the liquid containers. Because the consumable liquids were being passed, the valves have to have non-toxic materials and prevent contamination of the ingredients. We settled on refrigerator icemaker valves because of their safety value and also because of their financial aspect. Other valves we found were overly expensive in the 35-50 dollar range. The icemaker valves were approximately 25 dollars. Two-liter soda bottles were used to hold the various liquids. It was necessary to have a universal container to facilitate the ingredient holster construction and allow for a universal cap to be developed. ¼" holes were drilled into the caps and tubes were epoxied to the openings. Because universal caps and containers were used, it was simple and quick to replace and refill any ingredients.

The software aspect of the project was the most complicated and therefore our main focus. Originally, Java or C was the language of choice, but due to the difficulty of creating a GUI, Visual Basic .NET 2005 Express was chosen. Because neither of us had experience using the Visual Basic language, there was a lot of time invested in researching and learning. The most challenging hurdle we faced was commanding specific valves to shut off at the correct time intervals. Originally planning to cycle through each valve that was necessary for the chosen drink, it was found to be less intensive on both the program and microcontroller if all the valves were commanded upon in parallel. Enabled through a special set of commands that came with our microcontroller, this was possible and turned out to work better with much faster response time in turning on and off in congruency with the time intervals of the computer program.

Overall, the project was challenging because it forced us to utilize our software development skills that we learned while taking Computer Engineering courses at UCI. Even though the relay control board came as a kit, it was fully disassembled. So we were required to read the schematic and solder resistors, diodes, capacitors, and IC's to a PCB. Once assembled, we had to fully understand how the relay control board worked in order to develop an effective program. Other than our computer engineering skills, we were forced to think outside the box to develop an enclosure and method to hold the liquids and valves. This project has refined and developed our technical skills and therefore allowed us to develop into better engineers.