

An Automated Microcontroller-based Cocktail Mixer and Dispenser

Raphael M. Asuncion¹

Ryan D. Prodigalidad²

Daniel V. Ramirez³

Luisito R. Tomelden Jr.⁴

Department of Electronics and Communications Engineering
De La Salle University-Manila

Email: ¹ raphael_asuncion@yahoo.com, ² ryeprodie2@yahoo.com,
³ danuelvramirez@yahoo.com, ⁴ louietomelden@yahoo.com

Analene Montesines-Nagayo

Department of Electronics and Communications Engineering
De La Salle University-Manila

Email: nagayoa@dlsu.edu.ph

This paper focuses on the design and implementation of an automated microcontroller-based cocktail mixer and dispenser that can provide mixed drinks with the push of a single button. The machine uses three microcontrollers that functions as the brain of the system to which all the operating functions of each module are chronologically programmed in it. It has predefined programs and instructions that are responsible for the cocktail production processes that the machine will perform as directed by the user through an input device. The machine has two sets of dispensers responsible for dispensing the ingredients to the mixing container and for pouring the final cocktail in a glass. Submersible pumps, which are controlled by relays, are used to transfer the ingredients to the mixing container through hose pipes. The quantity of each ingredient to be dispensed is controlled and monitored by the microcontrollers' programs. After dispensing the right proportions and transferring them into the mixing container, a simple motor is used to mix them. After which, the cocktail is dispensed in a glass through a valve controlled by a servo motor. The ingredient level for every container is monitored by a water level sensor located inside the container. The system is able to serve a drink in just 5 seconds, offering a choice of 9 pre-programmed cocktails and customized drinks out of the six basic ingredients at the touch of a button.

1.0 INTRODUCTION

Nowadays, automated machines are in demand for they make numerous activities not only easier but also efficiently. These machines require minimal human intervention to do its job.

In most cocktail bars, the busiest person is usually the bartender since he is the one who mixes and prepares drinks for customers who are eager to have a drink. Due to time pressure and demands from the costumers, bartenders are prone to making mistakes during the actual cocktail making process.

The automated microcontroller based cocktail mixer and dispenser minimizes the need for bartenders to cater to the needs of the customers. The machine is user-friendly and is very simple to operate. The customers will only have to deal with marked buttons that indicate the kind of cocktail drink to be produced. With this, labor cost will be minimized and it will also give bar owners the opportunity to attract more customers with this innovation.

Furthermore, with the use of the automated mixer and dispenser, accuracy is obtained. The machine controls beverage pour sizes and ensures portioned drinks and cocktails. This machine prevents product loss by eliminating over-pouring and spillage of liquor inventory as well as breakages due to bottle mishandling. This will, in effect, lead to maximizing of the establishment's liquor inventory and increased profit.

The machine makes the cocktail mixing and dispensing process more hygienic. By means of automated machines, manual handling of the ingredients will be eliminated. The use of removable containers, stirrer, mixing container, and plastic hosepipes remove the concern for rust, corrosion and piled-up dirt.

There are existing automated cocktail dispensers in the market today but most of them are PC-Based such as KIScocktail[7] and Bar Monkey[8] system. This paper shows how the implementation of cocktail drinks mixing and dispensing processes are done through the use of PIC microcontrollers.

2.0 DESIGN CONSIDERATION

2.1 System Structure

The system has several mechanical and electronic components - all of which are divided into five groups namely the *QTERM-N15*, the *MICROCONTROLLER BOARD*, the *DISPENSERS*, the *STIRRER* and the *MIXING CONTAINER*. Figure 1 shows the physical

structure of the Automated Microcontroller-based Cocktail Mixer and Dispenser to which all five components are strategically placed.

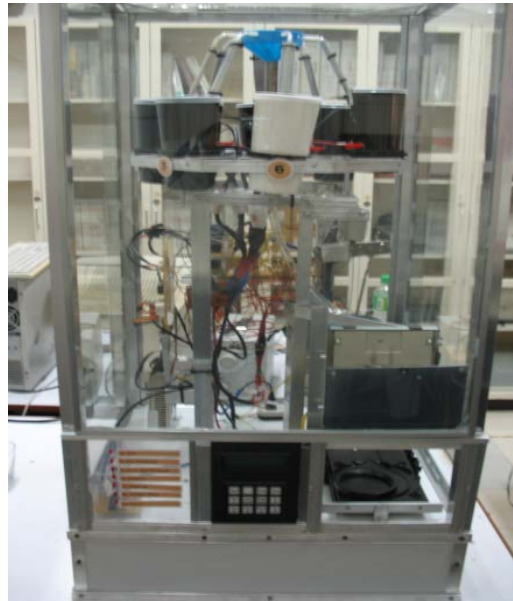
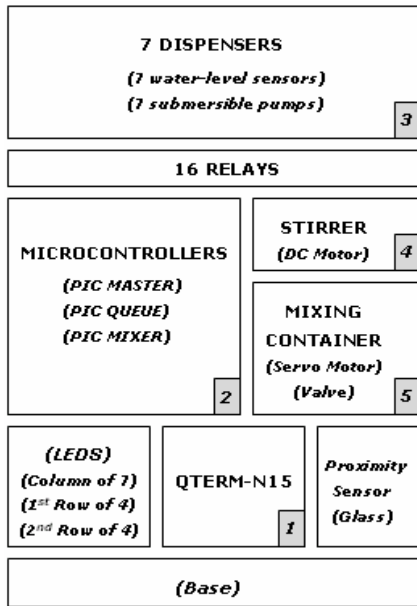


Figure 1: Physical Structure of the Automated Microcontroller-based Cocktail Mixer and Dispenser

The first component is the QTERM-N15 as shown in figure 2. It serves as the interface where the user can access and operate the machine. It is a 3x4 keypad and is capable of displaying data in its 4x20 LCD display. The QTERM-N15 is also used to show the status of the system whether it is ready to accept inputs or not. It is placed in the middle of the system.



Figure 2: The Q-TERM N-15

The microcontroller board is the second component of the system as shown in figure 3. It is the most important component since it serves as the brain of the whole system. The prototype utilizes three microcontrollers – PIC MASTER, PIC QUEUE and PIC MIXER. All microcontrollers are placed in the middle (left side) of the system.

Basically, the PIC MASTER receives data from the QTERM-N15 as inputted by the user and delivers this data to the PIC QUEUE. The PIC QUEUE functions as storage of the data sent by the PIC MASTER to accommodate the features of saving and queuing data to avoid idle time. To make the PIC QUEUE more useful, rather than functioning as storage alone, it was programmed to display in the QTERM-N15 an error message that indicates that one of the queued cocktails is skipped due to an insufficient ingredient. Also, together with the QTERM-N15, the columns and rows of LEDs will be activated, indicating which ingredient needs refill (column of 7 LED), which queued drink is skipped (1st row of 4 LED) and which queued drink is ready to be served (2nd row of 4 LEDs). The one responsible for lighting the LEDs is the PIC MASTER and the PIC MIXER. The last microcontroller, the PIC MIXER, is responsible for controlling all the dispensing and mixing processes done by the machine. The three microcontrollers have utilized more than 30% of its ROM and more than 10% of its RAM leaving a lot of space more for future improvements and additional features to be added. Table 1 shows the memory utilization of the three microcontrollers.

Table 1: ROM and RAM Utilization of the three Microcontrollers

<i>Microcontroller</i>	<i>ROM</i>	<i>RAM</i>
<i>MASTER</i>	40 %	19 %
<i>QUEUE</i>	31 %	81 %
<i>MIXER</i>	32 %	11 %

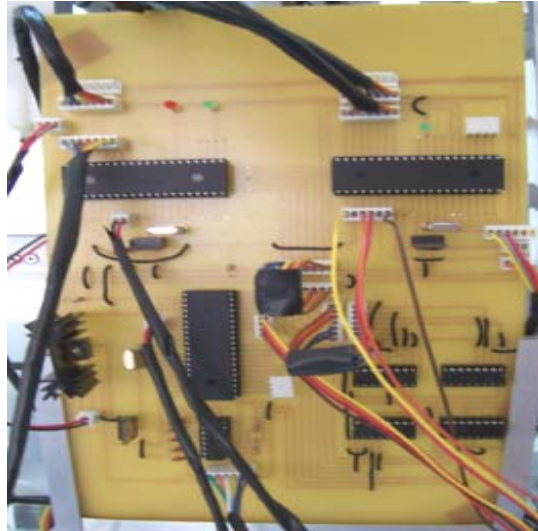


Figure 3: Microcontroller Board

The third component is the dispenser. The system has a total of seven dispensers. Six dispensers contain ingredients used in producing the cocktail; namely, gin, dry vermouth, sweet red vermouth, orange juice, campari and triple sec. The seventh dispenser contains water is intended solely for rinsing purposes. Figure 4 shows the prototype's dispensers for the seven ingredients. Moreover, it can hold an approximately 250 ml. Each dispenser contains a submersible pump and a water-level sensor. The two mentioned subcomponents of the dispenser work together for the transportation and level detection of ingredients respectively. All seven dispensers are placed on top of the system.



Figure 4: Prototype's dispensers for the seven ingredients.

The fourth component is the stirrer which is controlled by a DC motor as shown in figure 5. The stirrer is a bar spoon. It will be responsible for mixing all the ingredients towards the desired quality.



Figure 5: The Mixer comprising the DC Motor and the Bar Spoon

The fifth component is the mixing container as shown in figure 6. It is where the dispensed ingredients are mixed by the stirrer. It comprises of a valve and a servo motor. The servo motor is responsible for opening the valve after the stirrer has finished mixing the ingredients for a short period of time. Both the stirrer and the servo motor are placed in the middle (right) of the system.



Figure 6: The Mixing Container comprising of the valve and the servo motor

A champagne flute glass is placed directly under the valve of the mixing container to receive the final drink. The proximity sensor is used to detect the presence or absence of a glass below the valve before the submersible pumps start dispensing the ingredients towards the mixing container. The base is where the water will be drained after rinsing the hosepipes, the stirrer and the mixing container.

There will be a total of 16 SPDT relays as shown in figure 7: one for each submersible pumps and water-level sensors, one for the stirrer's DC motor and another one for the mixing container's servo motor. These relays are used to trigger the different components as instructed by the microcontroller. Moreover, 15 red LEDs, are used to indicate which container needs refill (column of 7 LEDs), which queued drinks have been skipped (right row of 4 LEDs) and which drink has been processed and ready to be served to the user (left row of 4 LEDs). The two rows of LEDs will display data in binary format. All LEDs are placed beside the QTERM-N15.

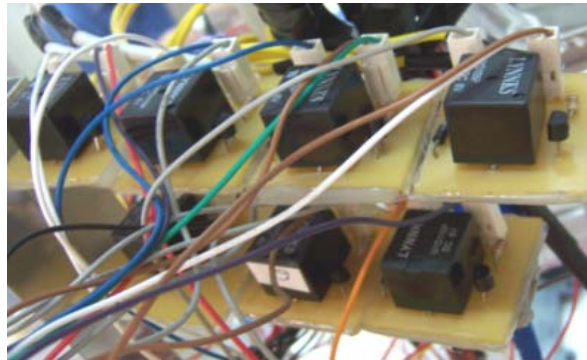


Figure 7: Relay Boards

2.2 System Flowchart

Once the machine is turned-on, water-level sensors inside the dispensers will be activated to determine if the ingredient level in each dispenser is still above or equal the operating level which is 150ml. If the sensors detected a level below 150 ml, the corresponding LED/s will light to tell the user which ingredient/s needs refill.

The system's interface (QTERM-N15) is then accessed by the user to input up to five fixed or customized cocktail drinks which will be processed one at a time. After receiving the input data, the three microcontrollers will accept, queue and process the data respectively. If the ingredients are insufficient, the user will be informed thru the Q-TERM that the drink ordered cannot be processed. The next drink in queue will be processed.

Before the submersible pumps begin dispensing the required ingredients towards the mixing container, a proximity sensor will check if there is a glass placed below the mixing container's valve. If the proximity sensor detected a glass below the mixing container and is properly placed, the machine will proceed with the next process; otherwise, it will wait for the user to place the glass.

If no other problems have been encountered, the stirrer which is powered by a dc motor will mix the dispensed ingredients in the mixing container for a short period of time. The mixing container's valve will be opened by the servo motor once the stirrer has finished operating. Then, final cocktail drink is transferred to a champagne flute glass.

Once the filled glass has been removed by the user, water from the seventh dispenser will be released by the pumps to rinse the hosepipes, stirrer and the mixing container. The water will be drained in a separate container at the base of the prototype which can be accessed by opening its back.

After producing one cocktail drink, the entire process of mixing and dispensing cocktail drink is repeated. The system flowchart of the project is shown in figure 8.

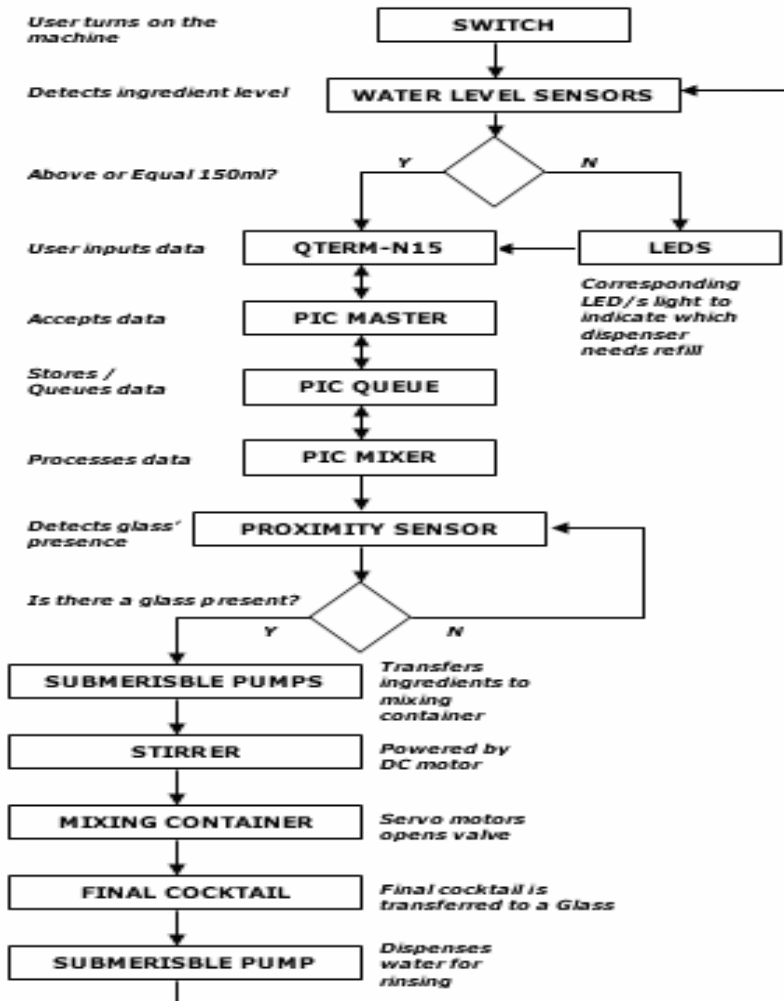


Figure 8: System Flowchart

2.3 Program Algorithm

PIC Master Algorithm

1. Initialize data and variables.
2. Initialize PIC Master to Q-Term connection.
3. Clear drinks data (for temporary drink array memory).
4. Display 'Start Message' and 'Drink Selection Message'.
5. Ask for User Input.
6. If 'Customize Drink' is selected, user inputs amount of each ingredient with a maximum total of 150 mL. If 'Pre-defined Drink' or 'Fixed Drink' is selected, user selects the corresponding drink number.
7. Update drinks data.
8. Display 'Sending Data Message'.
9. Send signal to PIC Queue that PIC Master is ready trying to send data.
10. Initialize PIC Master to PIC Queue connection.

11. Check if PIC Queue is ready to accept data. If not, wait for PIC Queue's availability.
12. Send drink data to PIC Queue.
13. Receive confirmation signal from PIC Queue.
14. If data from PIC Master to PIC Queue is invalid, and confirmation signal from PIC Queue is error data received, resend data to PIC Queue. If data transmission is successful, turn off signal from PIC Master to PIC Queue that PIC Master is trying to send data.
15. Initialize PIC Master to Q-Term connection.
16. Repeat whole process.

PIC Queue Algorithm

1. Initialize Pre-defined array data slots and Queuing array data slots.
2. Initialize idle connection (PIC Queue to: PIC Master, or PIC Mixer, or idle).
3. Send signal to PIC Master that PIC Queue is ready to accept drink data to Queue.
4. Check if there is an available Queue slot.
 - 4.1. If yes, check PIC Mixer is ready to mix drink data.
 - 4.1.1. If yes, initialize PIC Queue to PIC Mixer connection.
 - 4.1.1.1. Send signal to PIC Master that PIC Queue is unable to accept drink data.
 - 4.1.1.2. Send drink data to PIC Mixer.
 - 4.1.1.3. Accept confirmation for validity of data sent.
 - 4.1.1.4. Shift next drink data in queue to be processed.
 - 4.1.1.5. Initialize idle connection.
 - 4.1.1.6. Send signal to PIC Master that he is able to accept drink data.
 - 4.2. If no, go to # 5.
5. Check if PIC Master is trying to send data.
 - 5.1. If yes, check if there is an available queuing slot.
 - 5.1.1. If yes, initialize PIC Queue to PIC Master connection. If no, go to #5.
 - 5.1.2. Clear temporary array to queue.
 - 5.1.3. Accept data from PIC Master.
 - 5.1.4. Initialize idle connection.
 - 5.1.5. Convert data from PIC Master to valid drink data.
 - 5.1.6. Queue data.
 - 5.1.7. Save to Pre-defined drink slot if necessary.
 - 5.2 If no, go to #4.

PIC Mixer Algorithm

1. Initialize constants and variables.
2. Send signal data to PIC Queue that he is ready to accept drink data to mix.
3. Accept drink data if PIC Queue is trying to send drink data.
4. Send signal data to PIC Queue that he is not ready to accept drink data to mix at the moment.

5. Analyze drink data and dispense ingredients according to drink data received and analyzed.
6. Mix ingredients.
7. Dispense final mixed ingredients to glass (final output).
8. Wait for glass to be lifted from its placeholder.
9. If lifted, clean dispensers be dispensing clean water from one of ingredient containers.
10. Repeat instruction #2.

3.0 RESULTS AND ANALYSIS

3.1 Water Level Sensor Test

A water level sensor was placed inside the dispenser together with the pumps. Its operation was based on the shorting of two pins; that is, if the liquid was in contact with the two pins, they were shorted otherwise, they were open. If one of the water level sensors was open, the system informed the user via an LED indicator. To signal the user which one of the containers had insufficient ingredient required for an operation, an array of LEDs labeled from one (1) to seven (7) were placed next to the QTerm N-15 as shown in figure 3.

The test started from emptying a single container to emptying several containers. When one the container was emptied or its content was set below the minimum ingredient level, its corresponding LED indicator turned on. When two or more containers were emptied or their contents were set below the minimum ingredient level, their corresponding LED indicators turned on.

3.2 Proximity Sensor Test

This test verified whether the proximity sensor detected the presence or absence of a glass placed below the mouth of the mixing container's valve. If the proximity sensor detected a glass below the mixing container and was properly placed, the machine proceeded to the final dispensing process; otherwise, the machine did not dispense the cocktail and had to wait until a glass was placed.

No problems were encountered in the performance of the proximity sensor in providing the signal to the machine that there was no glass placed below the mixing container. As a result, the machine did not dispense the final cocktail drink in the absence of a glass below the mixing container's valve. Upon placing of the glass, the machine started the dispensing process.

3.3 Mixing Test

The mixing test determined if the stirrer and the mixing container together with their corresponding components functioned properly. During the testing process, the group observed that the stirrer and the mixing container worked according to their designed functions. The stirrer which was powered by a dc motors rotated for a specific amount of time and the mixing container's valve was opened mechanically by the servo motor. The DC motor was operated at 2 volts while the servo was operated at 5 volts.

3.4 Dispensing Accuracy Test

Dispensing the exact amount of ingredients for each kind of cocktail to be processed by the machine was very important since it significantly affected the taste of the end product. Tables 2 and 3 show the average error in percentage out of 5 trials of each dispensed ingredient for a specific kind of cocktail produced. From the results, it was observed that the densest of the six main ingredients is the Dry Vermouth (2) exhibiting a list of errors ranging from 1.64% to 6.67%. Second was the Sweet Red Vermouth (3). The least dense was the Gin which exhibited errors not more than 1.00%. It can also be concluded that vermouth type of ingredients are dense in nature as shown through the dispensing behavior of the Dry and Sweet Red Vermouth. All average errors have been limited to 10% as stated in the project's objective.

Table 2

Average Errors of Dispensed Ingredients in Fixed Cocktails.

1 – Gin, 2 – Dry Vermouth, 3 – Sweet Red Vermouth,

4 – Campari, 5 – Triple Sec, 6 – Orange Juice

FIXED COCKTAIL	1	2	3	4	5	6
<i>Dry Martini</i>	0%	6.40%	-	-	-	-
<i>Sweet Martini</i>	0.89%	-	1.87%	-	-	-
<i>Perfect Martini</i>	0.89%	3.45%	0.91%	-	-	-
<i>Gin and It</i>	0.22%	-	2.73%	-	-	-
<i>Gin and French</i>	0.67%	4.18%	-	-	-	-
<i>Negroni</i>	0.22%	-	2.73%	0.89%	-	-
<i>Orange Blossom</i>	0.44%	-	-	-	3.00%	1.40%
<i>Gibson</i>	0.44%	6.67%	-	-	-	-
<i>Bronx</i>	0.44%	1.64%	0.45%	-	-	0.91%

Table 3
Average Errors of Dispensed Ingredients in Customized Cocktails

CUSTOMIZED COCKTAIL	1	2	3	4	5	6
# 1	0.83%	4.00%	1.10%	-	-	-
# 2	0.08%	-	3.50%	0.43%	-	-
# 3	0.29%	-	-	-	2.67%	1.00%
# 4	-	4.64%	-	1.00%	-	-
# 5	-	-	1.33%	-	2.14%	0.60%

These discrepancies can be accounted to the following factors: the operation of the pumps, the positioning of the hose pipes, the pulses sent by the PIC to the relays of pumps and the liquids' density. The operation of the pump varied from time to time. We believed that the erratic behavior of pumps was inherent. In addition to that, the position and the attachment of the hosepipes varied differently for every dispenser. Some of the hose pipes were positioned to be more inclined as compared to the others. This inclination provided a higher chance for residue to remain inside the hosepipe. Moreover, the pulses sent by the PIC to the relays to activate the pumps varied from time to time; that is, pulses were delivered at varying speeds. Lastly and more importantly was the liquor's density, some of the ingredients' densities were really high compared to the others. The denser ingredient tends to leave more residues in the hosepipes adding greater error for the next operation.

3.5 Memory Test

This test basically determined whether the program was capable of storing five customized drinks in five memory slots. Note that the memory slots can only store customized drink. After the user customized the contents of his drink, the program asked if the user wanted to save the drink or not, before dispensing it. After which, it asked what slot number the drink was to be saved.

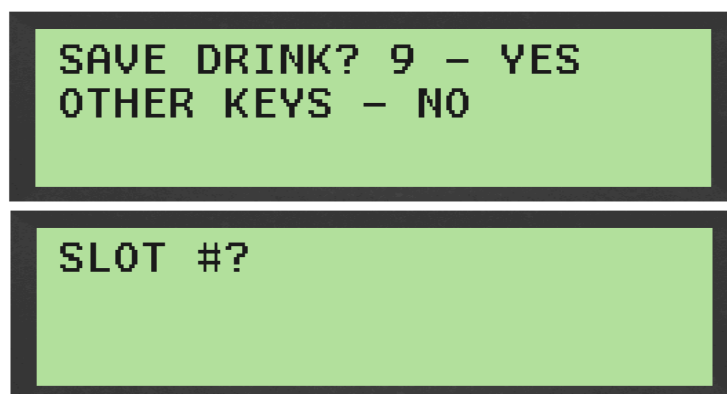


Figure 9: QTERM-N15 displaying the Corresponding Messages during Saving

As seen from Figure 9, the number nine (9) was for the 'YES' option and the other keys were for the 'NO' option. After the questions above were answered, the machine dispensed the said drink. The saved drink can be accessed in the third menu – the "Predefined Drinks" menu. It was also important to note that the five slots can be overwritten without warning.

3.6 Queuing Test Part 1

The user can queue either fixed or customized drink in the available five queuing slots. After the user has selected a drink from the three menus available namely the Fixed Drinks Menu, the Customizable Drinks Menu and the Predefined Drinks Menu, he was able to queue it. Figure 10 shows the three menus of the program.

It was also important to note that once the queuing slots were full, a red LED lit up indicating that the program was no longer accepting queuing operations

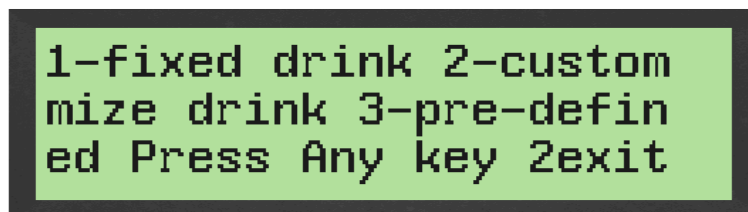


Figure 10: QTERM-N15 Displaying the three Menus of the System

3.7 Queuing Test Part 2

This part tested whether the machine was capable of displaying an error message in the QTERM-N15, and lighting the corresponding LEDs in the two rows of four (4) LEDs which queued drink has been processed, which was ready to be served and which queued drink was skipped due to an insufficient ingredient.

The first row of 4 LEDs (LEFT ROW) indicated the slot number of the drink which was processed and can be served to the user. The second row of 4 LEDs (RIGHT ROW) indicated the slot number of the skipped queued drink as shown in figure 11.

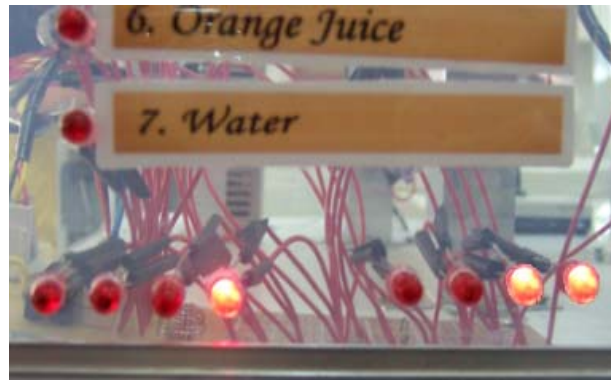


Figure 11: The Second Row of Four LEDs (RIGHT ROW) indicates the slot number (in binary 0011) of the skipped queued drink due to an insufficient ingredient)

It also tested whether the machine was really capable of skipping the queued drink, and proceeded to the next drink if a required ingredient for the queued drink was below the 150ml mark or was insufficient. Figure 12 shows the message that the QTERM-N15 will display once the case above occurred.

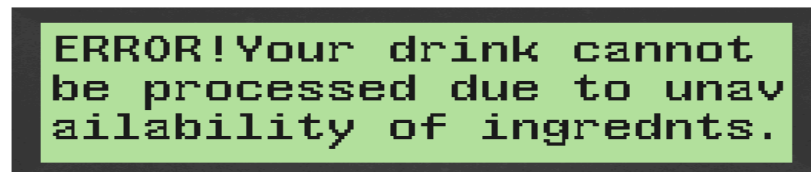


Figure 12: QTERM Displaying the error message

4.0 CONCLUSION AND RECOMMENDATION

The automated microcontroller-based cocktail mixer and dispenser provided a very satisfactory performance with a minimal percentage error. The utilization of a microcontroller has been accomplished in the form of the PIC16F877 microcontroller. The decision to use three microcontrollers was based on the elimination of idle time and the optimization of the cocktail making process. In addition to this, the utilization of the various proposed components such as dispensers, sensors, pumps, relays, dc motor and an input device was also accomplished.

Successful incorporation of intelligence in the machine has been attained by indicating which ingredient needs a refill, which queued drink was skipped, which drink was ready to be served and whether the machine would dispense the cocktail or not. Also, the capability to allot five (5) memory spaces for frequent users to program their customized cocktail and five (5) memory spaces to handle queuing operations has been achieved.

For future studies, we would like to recommend considering additional features that will maximize the use of the microcontroller, such as increasing the container capacity,

adding more ingredients, adding more memory for the saving and queuing slots, incorporating a cooling system, replacing the servo motor with a solenoid valve and the pumps with a carbon dioxide tank.

5.0 REFERENCES

- [1] Johnson, C.D. *Process Control Instrumentation Technology* (5th edition). New Jersey: Prentice Hall, Inc. 1997
- [2] Maloney, T. J. *Modern Industrial Electronics* (5th Edition). New Jersey: Prentice Hall, Inc. 2003
- [3] Nebojsa Matic, "PIC Microcontroller for Beginners, too!", May 30, 2000. [Online] Available:
http://www.mikroelektronika.co.yu/english/product/books/PICbook/0_Uvod.htm
[Accessed: June 15, 2005]
- [4] DeGroff, Dale, "List of Different Cocktail Recipes", 1999. Available:
<http://www.kingcocktail.com/Rcpemenu.htm> [Accessed: July 1, 2005]
- [5] Avery, Steven, "Information on the existing automated cocktail dispenser(Bar Monkey)", 2003. Available: <http://www.barmonkey.net> [Accessed: October 1, 2005]
- [6] MoreFocus Group Incorporated, "History of cocktail drinks", 2000. Available:
<http://www.drinkfocus.com/index.php> [Accessed: October 1, 2005]
- [7] Chocolate Delight Inc. *KisCocktail Bar*. Available: <http://www.kiscocktails.co.uk/>
[Accessed: June 15, 2005]
- [8] Barmonkey.Net. Forums. *Barmonkey Version 4*. Available:
<http://www.barmonkey.net/v5.asp>. [Accessed: June 15, 2005]

ABOUT THE AUTHORS



Raphael M. Asuncion, Ryan D. Prodigalidad, Luisito R. Tomelden Jr. and Danuel V. Ramirez were students of De La Salle University-Manila with degrees in BS Electronics and Communications Engineering. This article was based on their undergraduate thesis that they successfully defended last August 2006.



Analene A. Montesines-Nagayo is an Assistant Professor of Electronics and Communications Engineering Department at De La Salle University-Manila. She obtained her Master of Engineering degree major in Electronics and Communication Engineering from the same university. She specializes in power electronics, analog and digital electronic system design, microprocessor system and interfacing.