

The Memrowave

A UCF Senior Design Project

Fall 2014

Group 31

Winston Todd

Jack Gulick

Joseph Serritella

Darren Armstrong

Table of Contents

Table of Contents	ii
List of Figures	iv
List of Tables	vi
1 Executive Summary	1
2 Project Description	2
2.1 Project Motivation	2
2.2 Objectives	4
2.2.1 Automatic Timer	4
2.2.2 Food Product Database	6
2.3 Project Requirements and Specifications	9
2.3.1 Automatic Timer and Power Level	9
2.3.2 Local Product Database with Web Update	9
2.3.3 Touch Screen Interface	11
2.3.4 Barcode and QR Code Reader	12
2.3.5 SMS Notifications	12
2.3.6 User Profiles	12
2.3.7 Manual Operation	13
3 Research	13
3.1 Similar Projects and Products	13
3.1.1 Retail and Other Smart Appliances	13
3.1.2 Previous Senior Design Projects	14
3.1.3 Current Microwave Interfaces and Functionality	15
3.2 Relevant Technologies	28
3.2.1 Magnetron Control	29
3.2.2 ARM Microprocessors	34
3.2.3 Computer Vision for Barcodes and QR Codes	35
3.2.4 Output Display Technologies	35
3.2.5 Input Touch Screen Technologies	36
3.2.6 Operating Systems	37
3.2.7 Wi-Fi Interface	38
4 Project Design Details	40
4.1 Initial Design Approaches	40
4.2 Hardware	41
4.2.1 Power Supply	41
4.2.2 Magnetron Controller	47
4.2.3 Miscellaneous Microwave Controller	52
4.2.4 ARM Microprocessor	57
4.2.5 LCD with Capacitive Touch Screen Display	61
4.2.6 Wi-Fi Interface	62
4.2.7 Camera	63
4.3 Software	63
4.3.1 The Android Operating System	64
4.3.2 Drivers, Libraries, Dependencies	64

4.3.3	Microwave Control	64
4.3.4	Barcode and QR Code Software	68
4.3.5	Local Product Database	68
4.3.6	User Profiles	70
4.3.7	User Interface	72
4.3.8	SMS Notifications	90
5	Design Summary	90
5.1	Software Design Summary	90
6	Project Prototype Construction and Coding	97
6.1	Parts Acquisition and BOM	100
6.2	PCB Vendor and Assembly	104
6.3	Final Coding Plan	106
7	Project Prototype Testing	107
7.1	Hardware Test Environment	107
7.2	Hardware Specific Testing	111
7.3	Software Test Environment	118
7.4	Software Specific Testing	118
7.4.1	Unit Tests	118
7.4.2	End-to-end Tests	122
8	Administrative Content	128
8.1	Milestone Discussion	128
8.2	Finance and Budget Discussion	131
	Appendix A – Copyright Permissions	A-1
	Appendix B – Datasheets	A-1
	Appendix C – References	A-1

List of Figures

Figure 2.2.1-1: Automatic timer flowchart	5
Figure 2.3.2-1: Database flow chart	10
Figure 2.3.2-2: Local database update service flowchart	11
Figure 3.1.3.1-1: Whirlpool control panel	16
Figure 3.1.3.1-2: Panasonic Control panel	16
Figure 3.2.1-1: System control diagram	29
Figure 3.2.1-2: Analog PWM	31
Figure 3.2.1-3: Digital PWM	31
Figure 4.1-1: Possible integration of designs	41
Figure 4.2.1-1: PCB Corner Design	42
Figure 4.2.1-2: Beaglebone with built in power supply	43
Figure 4.2.1-3: Camera example	44
Figure 4.2.1-4: Wi-Fi Module	44
Figure 4.2.1-5: AC-to-DC power supply	45
Figure 4.2.1-6: Voltage regulator	46
Figure 4.2.1-7: Power Flow Chart	47
Figure 4.2.2-1: Interactions between PWM and source signal	47
Figure 4.2.2-2: LM555 pins	48
Figure 4.2.2-3: LM555 layout and example output	49
Figure 4.2.2-4: MSP430 counting options	49
Figure 4.2.2-5: Controller placement	51
Figure 4.2.3-1: Standing wave depiction	52
Figure 4.2.3-2: Pin Diagram	56
Figure 4.2.3-3: Switching circuit.	57
Figure 4.2.4.2-1: Beaglebone Black with Sitara AM3358 ARM Microprocessor	58
Figure 4.2.4.2-2: Simple Memrowave system diagram	60
Figure 4.2.7-1: Logitech C270 Webcam	63
Figure 4.3.3-1: Microwave control board I2C command poll sequence	66
Figure 4.3.3-2: MicrowaveController UML Diagram	67
Figure 4.3.4-1: BarcodeScanner class diagram	68
Figure 4.3.5-1: Local product database structure	69
Figure 4.3.5-2: Local product database class diagrams	70
Figure 4.3.6-1: User profile database schema	71
Figure 4.3.6-2: Class diagram of User Profiles	71
Figure 4.3.7-1: Main menu/home screen	72
Figure 4.3.7-2: Main menu flowchart	73
Figure 4.3.7-3: Manual cook screen	74
Figure 4.3.7-4: Manual control flowchart	75
Figure 4.3.7-5: Scanning barcode flowchart	76
Figure 4.3.7-6: Scanning a barcode	77
Figure 4.3.7-7: Scanning successful	78
Figure 4.3.7-8: Successful barcode scan processing	79
Figure 4.3.7-9: Automatic operation complete	80
Figure 4.3.7-10: Auto operation complete	81

Figure 4.3.7-11: Adjust auto settings	81
Figure 4.3.7-12: Automatic/scanned settings adjustments	82
Figure 4.3.7-13: Database record not found	83
Figure 4.3.7-14: Product not found	84
Figure 4.3.7-15: Entering new product, upper case	84
Figure 4.3.7-16: Entering new product, lower case	84
Figure 4.3.7-17: Product cook settings	84
Figure 4.3.7-18: Use new local record	85
Figure 4.3.7-19: Quick Cook settings	86
Figure 4.3.7-20: Quick Cook menu	87
Figure 4.3.7-21: Settings menu	88
Figure 4.3.7-22: Memrowave profiles and settings	89
Figure 4.3.8-1: SMSNotifier class diagram	90
Figure 5.1-1: Class diagrams home screen, manual cook screen, and microwave controller class	91
Figure 5.1-2: Class diagrams for the barcode scan screen, product cook screen, and quick cook screens	93
Figure 5.1-3: Class diagrams for the new user screen, modify user screen, and user selection screen	95
Figure 5.1-4: Class diagrams for the new product screen, modify product screen, and the product database	96
Figure 5.2-1: Microwave control board schematic	98
Figure 5.2-2: Microwave control circuit 'Cape' for the BeagleBone Black	99
Figure 5.2-3: Microwave hardware control diagram	99
Figure 7.1-1: Frequency response	108
Figure 7.1-2: Noise on DC offset	109
Figure 7.4.1-1: The BarcodeScanner class	119
Figure 7.4.1-2: The ProductDatabase class	119
Figure 7.4.1-3: The UserDatabase class	121
Figure 7.4.1-4: The SMSNotifier class	122
Figure 8.2-1: Budget bar graph	134
Figure 8.2-2: Pie chart of Budget	135

List of Tables

Table 3.1.3.3-1: Whirlpool dedicated button preset functions	20
Table 3.1.3.3-2: Panasonic dedicated button preset functions	21
Table 3.1.3.3-3: Whirlpool Cook button preset functions	22
Table 3.1.3.3-4: Panasonic Sensor Cook button preset functions	23
Table 3.1.3.4-1: Whirlpool Reheat function	25
Table 3.2.1-1: Manufacturers	33
Table 3.2.1-2: Manufacturers, Types, and Descriptions	33
Table 3.2.7.1-1: Common Wi-Fi standards	39
Table 4.2.1-1: Microwave component power specifications	46
Table 4.2.2-1: LM555 pins	48
Table 4.2.3-1: Illumination table	54
Table 4.2.3-2: Power table	55
Table 4.2.4.2-1: Sitara AM3358 ARM Microprocessor Details	59
Table 4.2.4.2-2: Beaglebone Black Details	60
Table 4.2.5.2-1: Chipsee CS-BBB-EXP50C Display Cape Details	62
Table 4.2.6-1: Netgear WNA1000M specifications	63
Table 4.3.3-1: Microwave control circuit commands	65
Table 6.1-1: Rough Bill of Materials Estimate	103
Table 7.1-1: Testing equipment list	110
Table 7.1-2: Testing Overview	111
Table 8.2-1: Sponsors	132
Table 8.2-2: Budget	133

1 Executive Summary

The Memrowave is a UCF engineering Senior Design project for the Fall 2014-Spring 2015 semesters. The main purpose for this project is to make a smarter and more autonomous microwave capable of cooking multiple foods with ease and limited user input. Typical microwaves incorporate the use of tactile buttons and segmented displays to relay information to the user. The user is then required to stand-by and monitor the cooking process. Some of the most common issues with this process are proper cooking and consistency of the food being produced and being limited to close proximity to the microwave. The Memrowave will have the ability to overcome these issues.

The Memrowave is being designed to be a microwave that is more connected and autonomous. An automatic timer will be incorporated into the system which will make the user input much simpler and more efficient. Through the use of bar code scanning, cooking configurations based on the code of the food package will be sent directly to the Memrowave, allowing it to begin the cooking process once the door is closed. This approach eliminates the hassle of identifying proper cooking times based on different wattage microwave appliances. Along with bar code scanning, a food product database will be setup to log all the specific information needed to cook foods. The already established UPC code will act as an index to identify the product.

The connection to the internet will be implemented through WIFI, in addition to using this connection for the web database, the Memrowave will have the ability to send notifications to the user's phone or mobile device, allowing the user to roam as the food is being prepared. The main user interface will be a LCD touchscreen on the front of the microwave. The user will be able to access all the features of the Memrowave, to use it as a regular microwave, or to use the smart features.

Other senior design teams and appliance manufacturers have made efforts to make appliances smarter by creating mobile applications that can serve as a remote control for appliances such as washing machines, and dishwashers to report status changes and when the user's input is needed. When it comes to smart microwaves, most enhances in the industry are done through the ability to having touch-type interfaces and more specific and numerous preset configurations to cook multiple foods. Memrowave will be able to add to these enhancements and also be an internet connected device that will be simple to use and desirable for those who with no time to waste.

The basic components needed for designing Memrowave include the magnetron, internal lights, internal beeper, power switch, turntable motor, and power supply. The more complex components include the processing unit, printed circuit board, camera module, WIFI module, and LCD touchscreen. Divisions between

hardware and software form the basis for the design plan: powering devices, power operation of switches, PCB assembly, and final enclosure assembly are in the hardware side of the design, controlling of the multiple peripherals, constructing the user interface, web database development, and bar code scanning implementation are in the software side of the design. Integration of both the hardware and software is the final step in developing the Memrowave. After initial integration any design changes will be made and implemented within the development time frame.

Overall the Memrowave will be a go to device for serving any food preparing needs when time is limited, not simply for cooking frozen package foods, but also reheating, defrosting, and simmering with the ease and consistency of an autonomous system.

2 Project Description

In this section, we will describe the project in detail, including our objectives, requirements, specifications, and motivation for designing the Memrowave.

2.1 Project Motivation

Today's society calls for ever more growing technologies. Consumers demand up-to-date devices that push the envelope of what we can achieve through technologies. These Technologies may provide us with revolutionary devices that shape the world, some may make the hardest of tasks simple with the push of a button, and some new technology is just developed to simplify own lives even further. The ideal to make everyday tasks quicker and more efficient is the main driving force behind our design the Memrowave. The Memrowave seeks to take a common household appliance and create a better more innovative appliance that is smart, fast, and just plain simple for everyone to use. The end goal is allow the user to quickly set up his morning meal or dinner on the go with a quick barcode scan, having the user spend much less time around the kitchen, granting the user more time to prepare for his day while his food cooks its self. In order to create this appliance of the future there are certain task to be completed; The Memrowave must learn the cook times for each food, it must adapt to the preferences of the user, it will alert the user to completed food times and estimated cool down times for food, the Memrowave will keep track of amount of food eaten by the users, and lastly there must be a simple friendly user interface.

In order for the Memrowave to become a "smart: microwave it must first gather the prospective cook times of each food the user may want to consume. There are many possible ways this can be achieved however, we are going to consider two and focus on one of the methods. The first method is to manually store the data containing information such as cook time to the memory of the device. Once

the barcode is scanned the Memrowave will retrieve the information stored in memory. This method, simple, yet unfortunately doesn't stick to the theme of being efficient and seamless. The proper approach to our problem is to connect Memrowave to the internet via Wi-Fi. Upon scanning for the first time a product's barcode, the Memrowave will search the internet for the manufacture's cooking instructions. Once obtained these instructions such as cook time, power level, and cool time will be saved into memory. The Memrowave will then begin cooking the food based on the recommended settings and shut after the process is completed. Once the product is stored in memory the Memrowave will no longer need to use resources to search the internet. However if an internet scanning algorithm proves to be too complicated our team will devise another method capable of achieving a similar goal. The final method the Memrowave can use to provide a database can be creating an online file in which the Memrowave will access. Our team can manually store initial products to the online spreadsheet. This spreadsheet will contain the barcode's identity, cooking times, cool down times, and product names. Once the device is scanned the Memrowave will retrieve information via Wi-Fi through the webpage created by our design team. The barcode scanned will be matched and compared to a specific barcode from the online spreadsheet. After conformation the Memrowave will retrieve the cooking time. The cooking time will be feed to the hardware and the proper amount of power and on time will be supplied. If the spreadsheet search returns a cool down time a text message will be sent via Wi-Fi to notify completion. This Online database we create also will have the ability to edit and write to. For example if the barcode cannot be found in the database, it will be saved in a new cell. The user will then be prompted to input cooking time, cool down time, power level, and verify correctness. This information will be saved and ready for use next time the new barcode is scanned. Another example of editing the smart microwave's online database can be accomplished by changing preexisting cooking times. By allowing this acting the online database can be molded to the users liking. This edit however may be a problem outside of prototype phase because our prototype will only use one online database. To combat this flaw our production models will need to utilize individual databases. These individual databases will all have separate online locations therefore isolating each Memrowave from each other.

Keeping to the theme of making things better at what they do a focus of the Memrowave is to adapt to the user and their likings. To accomplish this goal the Memrowave will allow the user to change the stored cook times previously used. For example, if a meal was deemed to be overcook a user can alter the timing in arbitrary integer increments to achieve the desired taste. Also the same will be true for the reverse case thus, allowing for positive or negative integer increments with respect to the original timing stored in memory. This information will then be stored as the new time for later usage. This option will be referenced for a particular item every time that item is processed.

The next task of the Memrowave is allowing the user to not be present during cooking of the product. Utilizing the microwave's ability to access the internet, we create notifications via text messaging. This system will ping the user once a product has completed its cook and cool down cycles. This feature keeps the microwave new and innovative compared to standard units. Texting will provide the user with the ability to continue on with daily tasks while avoiding waiting for the microwave to ping. The system is particularly affective when a user is out of hearing range of the Memrowave.

Eating provides a person with the energy needed to focus throughout the day however, over eating can be detrimental to a person health. This is where the Memrowave comes in. It can function as a daily calorie counter. Information about a product's nutrition is stored online, which the Memrowave can retrieve and track. This feature isn't supposed to stop the microwave from cooking once it reaches a boundary condition but, it will simply display the amount consumed for the day thus helping the user make a healthy choice.

Lastly, a product can only successful if and only if the mass consumer finds it simple, appealing, and has ease of use. The Memrowave seeks to make this a key aspect of its design by integrating a seamless LCD touch screen as the user interface. The LCD will display all the features mentioned previously such as the barcode scanner, the clock, the calorie counter, and user adjustable controls to change times. When implanted into the Memrowave navigation of the menus will be a breeze, thus link the whole thing together.

The Memrowave seeks to be the next must have house hold appliance. The Memrowave came into design because we wanted to make preparing our meals even simpler. It draws on the ideal that simpler is better and time is important. These goals are achieved by adapting to the preferences of the user, text message alerts to the user of completed food times, a seamless LCD touch screen user interface, and a barcode system that cooks our food for us.

2.2 Objectives

There are a number of objectives we will be trying to achieve with the Memrowave. In this section, we will describe each of these objectives in detail.

2.2.1 Automatic Timer

The main focus of the Memrowave as stated before is to make preparing food much simpler and faster for the user on the go, this is what we call the Automatic timer. Automatic timing cuts out all the user thought on food preparation. The user simply scans the barcode of the product and closes the door. Once closed the user won't have to stress over reading instructions and punching in numbers because the Memrowave does that for him/her. This Automatic timing is what

sets our product apart from standard microwaves. Timing is begun once the Memrowave obtains the cooking information (this is discussed in more detail in a later chapter) and activates the radio frequency (RF) power. Power is kept on until the automatic timer reaches the desired cook time. Timing of products can even be taken a step further by achieving a cool down timer once the first timer reaches zero. This cool down time won't be user operated either thus making it the second automatic timer on the microwave. The function of this automatic timer is cooling of recently cooked products. This information can also be found online for a given product.

Displaying of the automatic timer is through a store bought LCD screen interfaced onto the front of the microwave (this is discussed in more detail in a later chapter). The timer will start at the top of the cook time for the product and count down till zero. Once reached, the time will flash until the microwave door has been opened by the user thus resting the automatic timer and the LCD screen to a stand-by state. The only interface the user will have with the timer is if small incrementally changes in time are needed in timing.

Timing will be kept by using a microcontroller. The microcontroller will deactivate the RF power once the automatic timer has reached zero. From an overhead view the timer, magnetron, and microcontrollers interact together as depicted in Figure 2.2.1-1.

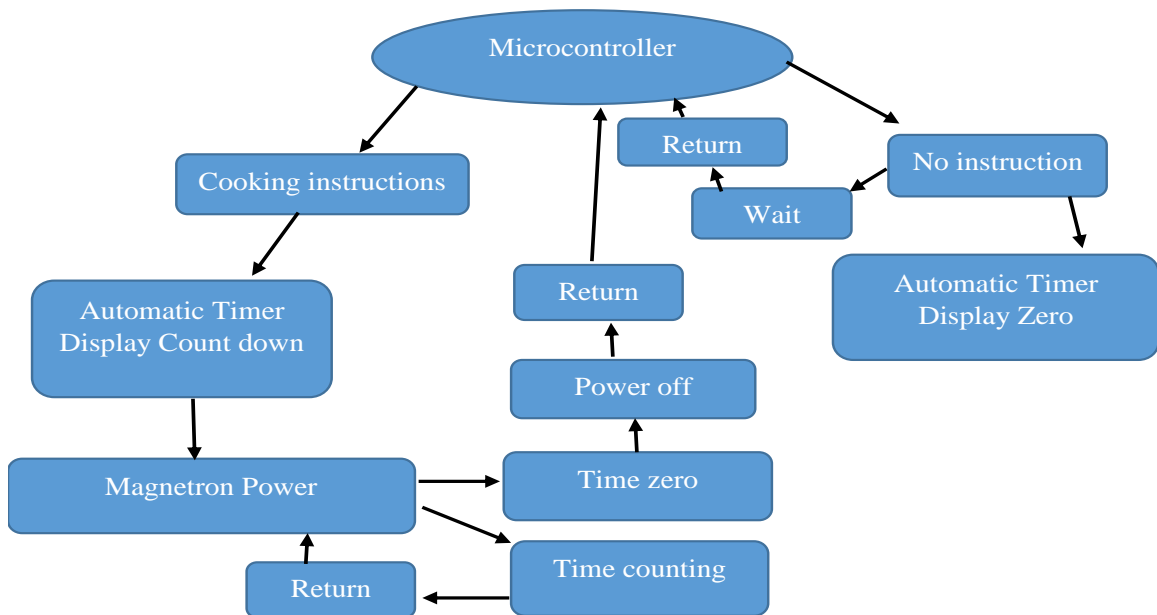


Figure 2.2.1-1: Automatic timer flowchart

2.2.2 Food Product Database

The Memrowave should include a database of all of the food products that it knows how to cook, and it should be able to learn how to cook any new food product. This will be accomplished using a database that indexes food products based on their UPC, or Universal Product Code. The UPC is convenient because most food products already include a unique UPC that is easily scannable, that is, the barcode.

The food product database would store all of the relevant information for a particular food product, including the UPC, name, description, preparation instructions, cook time, and power level. This information could be provided by us or by the user. For example, suppose the user scans a food product that is not in the database. The user must manually enter the cook time and power level, but they should not have to do this more than once for a particular product. The Memrowave should remember this information for the next time the user scans this product. This will be accomplished by storing the information in the product database.

Sometimes the cook time provided by food manufacturers isn't optimal for a particular microwave or altitude. The food may come out still cold, or it may be a little burned, etc. Because of this, we would like the Memrowave to provide the option for the user to modify the cook time of any product. The user's modification will be stored in the database and given priority over the manufacturer's information.

2.2.2.1 Cook Times and Power Levels

In this section we will discuss in more detail the process of obtaining cook times and power levels. In order to properly cook product inside a microwave the user must follow the instructions labeled on the package. These instructions are as follow; cooking time, power Level, and cooling down time. While obviously a simple task for all people to complete it is the Memrowave's goal to get rid of this step.

The process begins with the user scanning the barcode of the product into the camera mounted to the surface of the Memrowave. Once the barcode is scanned into the processing unit and read the Memrowave will begin an internet search via built in Wi-Fi capabilities. The barcode stores information such as the product's title, using the title we can retrieve the manufactures web-site. An algorithm (discussed in more detail in a later chapter) can then be used to search the web page and find key words such as cook time, power level, and cool time. The information is then stored in the Memrowave's food product database for use later.

Once stored in the food product database the cook times and power levels will be accessed next time they are needed. In order to access the food product database the user must scan the barcode of the product. The barcode binary is then matched with the information stored in the food product database thus retrieving the cook time, power level, and cool time. The information is then used to have the microcontroller set up and start the automatic time and power the magnetron. Once the cooking cycle is complete the system is powered down and the product is ready to be removed

The interaction between power levels and the system can be broken down as follows. Proper cooking of many products requires changes in the RF power delivered to the system. It is one of the goals of the Memrowave to regulate this in order to best cook foods. Once a product's information is stored in the food product database we can refer to a given product's power requirement. This requirement will be passed to the microcontroller which will then be feed through a system to regulate and control the power output of the microwave. This control system will be built on a Printed Circuit Board (PCB) which is discussed in more detail in a later chapter. This system can be modeled as shown in Figure 2.2.2.1-1.

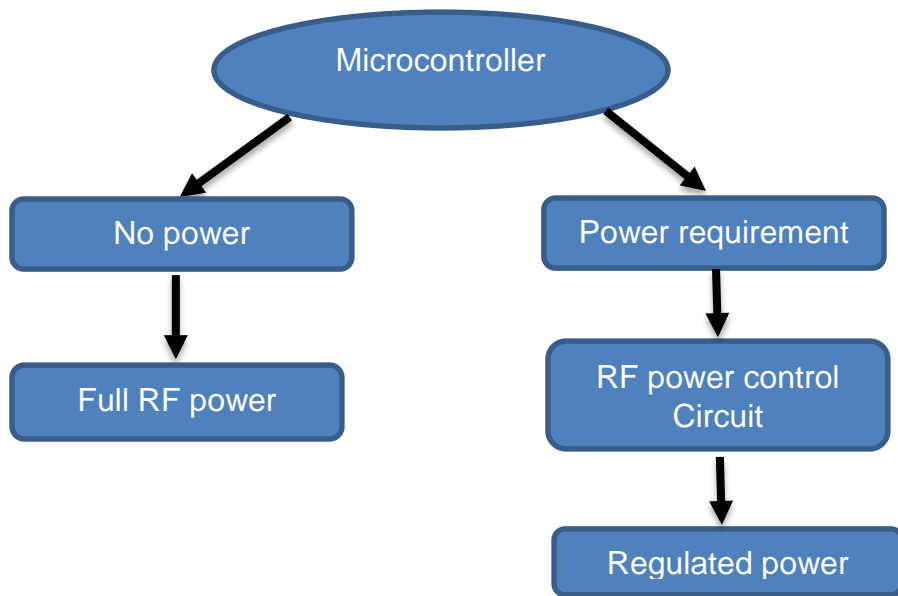


Figure 2.2.2.1-1: Power level flowchart

2.2.2.2 Preparation Instructions

Operation instructions describe the step by step process for the user to utilize the Memrowave while Preparation Instructions refer to microwave recipes the Memrowave will store for the user.

Operation instructions for the Memrowave are simple and very quick to follow; here is a step by step process for Operation. First the user begins by holding up a products barcode to the camera on the Memrowave. The LCD screen will display the image in real time to aid in the placing of the barcode. Once the Memrowave scanned the barcode a time is displayed and the user is prompted to close the door of the microwave. Once closed the LCD screen will display start and the cooking process begins. This period of time will allow the user to leave the microwave and carry on with other tasks during the waiting period. When the cooking is finished, the user will receive a text message informing that the time is up. This text is sent through the Memrowave using Wi-Fi. When the user opens the microwave door the LCD will display a question asking if the user would like to add or subtract 5 seconds from the cook time. If yes the user can adjust the time to his/her desire. Finally the microwave will return into the stand-by position until the user returns to cook more. Overall operation must contain a minimum amount of steps to make the Memrowave as simple as possible to use with much less time spent in front of it than a standard microwave.

If a product isn't currently saved the Memrowave will proceed as follows. The user is notified from the LCD that the product isn't saved and requires internet access to obtain the information. Once the information is retrieved the user is notified and the cooking process will begin as normal. However if there is no information online about the cooking instructions the user will be prompted to store the information personally. If the user would like to view the calorie counter for the day there will be an option on the touch screen LCD when it is in stand-by mode. If chosen the counted will be displayed and remain on screen for 30 seconds or until the user selects a return home option. These features will be discussed in much greater detail in the remaining chapters.

The Memrowave will also consist of another feature to help the user with preparing meals both quick and fancy. This feature is called "Quick Prep" and its function is to help the user create meals based off of microwave only recipes. The feature can be achieved by storing a list of ingredients and preparation instructions in an online database created by our design team. The Memrowave will provide the option via the LCD display screen. Once activated the screen will display a list of meals the user can make. Our preparation feature will also be active upon scanning of an items barcode. Once the item is scan the user can be prompted if they would like to see recipes that involve the product scanned. If the user selects "yes" the Memrowave will retrieve the possible recipes that include the current product and display them for the user. This feature scans the online database for the proper barcode contained in a recipe. If the user would like to use the preparation instructions for a given product the recipe will be displayed on screen during the duration of the session. The recipe will not be displayed during cooking and will be deactivated upon selection of the "finished option" displayed on the LCD.

2.3 Project Requirements and Specifications

In the previous section, we discussed our objectives for the Memrowave project. In this section, we will outline the specifications that must be met to achieve our objectives for the project.

2.3.1 Automatic Timer and Power Level

After a user scans a product's barcode, the Memrowave must automatically set the timer and power level, provided the Memrowave has access to this information. If, for whatever reason, the Memrowave cannot determine the appropriate cook time or power level, it will ask the user to enter this information; however, this should only occur once, because the Memrowave will store this information for the next time this product is scanned.

2.3.2 Local Product Database with Web Update

We would like the Memrowave to continue to work in situations where there is no internet access. To achieve this, we would need to include a local product database. This database would store all of the relevant information for a particular food product, including the UPC, name, description, preparation instructions, cook time, and power level.

When a user scans a food item using the barcode scanner on the Memrowave, it will first search the local database for an entry with a matching UPC. If the UPC is not found in the local database, the Memrowave will then attempt to connect to the internet and retrieve this information from the master product database, which will be made available through some web API. If the product cannot be found in the master database or if the Memrowave does not have internet access, then the user will be asked to manually enter the cook time of the product. This manual entry will be saved in the local product database for later use. A flowchart of this process is shown in Figure 2.3.2-1.

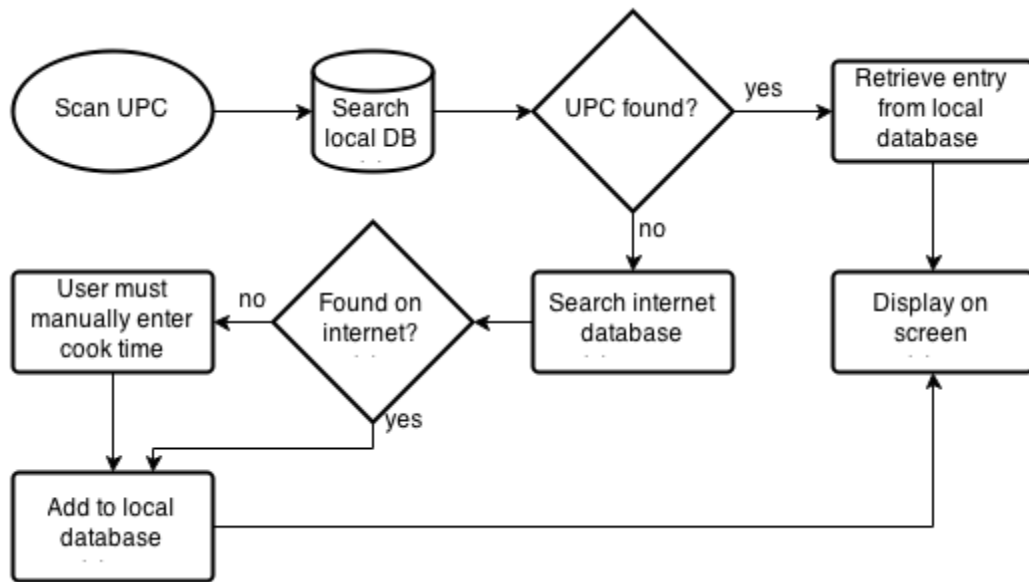


Figure 2.3.2-1: Database flow chart

In situations where internet access is intermittent, it is important to utilize the internet connection whenever it is available, so we've decided to implement an automatic update service for the local product database. This service, when enabled by the user, will periodically connect to the internet to check the master product database for updates. If any update is found, the Memrowave will download it and apply it to the local product database. This service will ensure that the local product database is synchronized to the master database, so the user will not need to create manual product entries in the case of an internet service failure. A flowchart demonstrating the update service routine is shown in Figure 2.3.2-2.

The automatic database update feature could have some conflicts with user created entries. For instance, the update could include an entry with a UPC that matches that of a user created entry. If this occurs, both entries should be saved, and the next time a user scans that product, the Memrowave should ask the user which entry to keep.

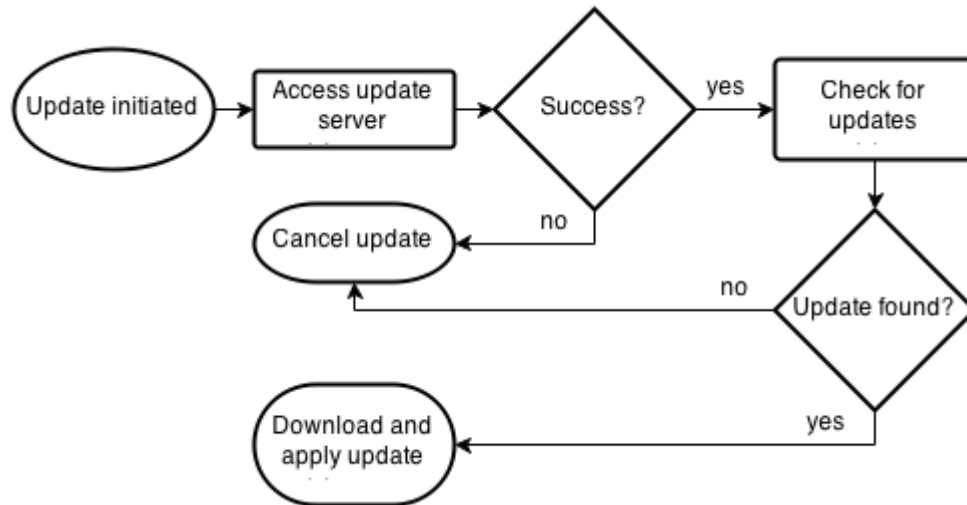


Figure 2.3.2-2: Local database update service flowchart

2.3.3 Touch Screen Interface

We want the Memrowave to feel like a real smart device, so it needs a touchscreen interface. The touchscreen must be large enough to read without straining, and it must be large enough for the user to easily interact with the software; however, it cannot be too large that it will not fit into the control panel of the microwave. An LCD in the range of 4” to 7” diagonally should meet these size requirements.

The LCD must be backlit, so that it can be seen clearly in low light, and the brightness should be adjustable by the user. Both the display interface and the touch interface must be compatible with the ARM microcontroller we chose to run the Memrowave.

The software interface must have a clean, ‘smart’ look. It must be easy to use, and the control elements must be large enough to select easily on the touchscreen. It should be quick and responsive, providing some form of feedback whenever it is loading or processing so that it never appears to be frozen.

The interface must have a well-defined, intuitive layout. The software should provide an interface for every available feature; the user should not have to navigate any external piece of software. All of the Memrowave’s settings should be accessible from one central location within the software. The settings should be stored on some non-volatile storage, so that they do not reset after a power-cycle.

2.3.4 Barcode and QR Code Reader

The Memrowave must have some method of reading barcodes to get a product's UPC. This barcode reader must be relatively quick, and it must be accurate. It should be able to read any barcode, regardless of any differences in colors, packaging, etc.

2.3.5 SMS Notifications

With a normal microwave, the user must wait within earshot of the microwave to hear when it is finished. With the Memrowave, we'd like to enable the user to know exactly when the microwave finishes cooking, even out of earshot. To achieve this, we decided that the Memrowave will have an SMS notification feature.

The SMS notification feature will allow the Memrowave to send text messages to the user's cell-phone. These messages will keep the user informed about the state of the Memrowave as it cooks the food. For instance, if the user is defrosting some food, the Memrowave could send an SMS notification instructing the user to flip over the food to defrost the other side.

This feature could also be used to notify the user when their food is finished cooking, when the food's cool down time has been reached, when to stir the food, etc. Basically, this feature can be used to send any message from the Memrowave to the user.

2.3.6 User Profiles

In most cases, microwaves are used by more than one person, so the Memrowave must support multiple users as well. The Memrowave must know who is currently cooking food so that it knows where to send an SMS notification and so it can keep track of nutritional information. The simplest way to accomplish this would be to implement some sort of user profile system.

Each user profile will store all of the settings for a particular user, including the cell phone number, notification settings, and any other user-specific setting that may be required. It must be easy to switch between profiles to make the cooking process as simple as possible.

The profiles are a feature, not a requirement, so the Memrowave must also include the option to cook without using any profiles. This could be accomplished by including a generic guest profile, or by having the user select a profile only when they would like to use this feature.

2.3.7 Manual Operation

Sometimes a user may not want to use the smart features of the Memrowave, or the smart features may not work well for certain types of cooking and recipes. For cases such as these, standard manual operation features are necessary, so the Memrowave must provide these features.

The manual operation features include basic cook time selection, start and stop buttons, power level selection, and any of the basic features you might find on a standard microwave. These features are described in detail in section 3.1.3.

3 Research

Before we can design any systems of the Memrowave, we must research a number of things pertaining to the project, including similar projects and products and the relevant technologies that we will be using to design the Memrowave.

3.1 Similar Projects and Products

The Memrowave is not entirely unique, but every product builds off its predecessors. In this section, we will be researching products and projects that are similar to the Memrowave to gain a better understanding of how we will design the Memrowave.

3.1.1 Retail and Other Smart Appliances

Most microwaves deemed smart by its manufacturers set these products apart from the rest by adding features to the microwave rather than changing its primary functions. Instead of physical buttons, touch-type buttons are used. Power levels are much more graduated. The microwave is transformed into a device made to cook all types of meals with different preset configurations to make it act as a grill, fermentation device, vegetable steamer, and defroster. Some of the microwaves are also self-cleaning, a feature usually for the high-heat capacity of a conventional oven. Many of these features we want to add to our smart microwave, but we want the Memrowave to also be more connected.

More of these connected features are used in smart refrigerators, washers and dryers, and dishwashers. Whirlpool has implemented in its smart appliances what is called 6th Sense Live Technology which allows the users to manage and control these devices. Energy usage, time left on machines, and the condition of all your appliances are available to you through a phone application. The smart appliances application also makes use of cellular push notifications. Samsung

has also implemented touch screens into their refrigerators for temperature and other controls. These types of systems seem to be implemented where the appliance is a stand-alone autonomous machine, giving the user status reports or whether some action is needed by the owner.

Most smart appliances make use of touch-enabled interfaces, WIFI, and mobile device applications. Device which can be left alone for hours or are more autonomous than a microwave benefit from those features, these include washers and dryers, air conditioning units, and water heaters.

Other smart technologies include also include system diagnostics, error detection, system failure reports. These can also implemented with a connected system to keep the user more aware of the condition of their appliances. Whirlpool in their Energy Smart water heater has uses system diagnostics to check for dry fire, upper or lower element condition, upper or lower thermistor condition, water temperature limits, and thermostat functionality. These diagnostics provide error codes which are readable on the LCD screen of the user interface; there is not a mobile solution to accessing such information, which we would like to implement in the Memrowave.

3.1.2 Previous Senior Design Projects

A project similar to what we plan on implementing was created developed by a Senior Design group in the Fall 2013 - Spring 2014 semesters. The project is called N.O.M.S., the Nutritional Object-identifying Microwave System. It “takes the idea of a traditional microwave and advances the technology to be on par with technology today.” This project for a smart microwave incorporates many of the “smart” ideas we wish to incorporate into our project. N.O.M.S. incorporates the use of a camera as a QR code reader to scan barcodes that contain the information for cooking times and power levels. A touch screen on the microwave is used as the main interface. The LCD touch screen was used to display the time, for UPC code scanning, and proving the status of the product while cooking. For storing the information necessary for cooking configurations, they used the caching system for short term storage then if it was not stored in the cache then a web search would be executed. WIFI was used to provide an internet connection to the microwave, which was configurable through PC or mobile software.

Comparing the N.O.M.S. project to Memrowave, the idea of a more connected appliance is the ultimate goal while the implementation and overall features have many similarities and differences. Both will use a touchscreen as the main user interface for accessing all the features, a camera will be used to collect bar code information on cooking configurations, but different systems for reading and saving cooking configurations are being used, N.O.M.S. used the bar code as a ID tag to go and search a local and/or web database for the cooking

configuration, with the Memrowave, the entire cooking information configuration will be contained within the scannable code. The use of WIFI will be mainly the same, as a connection to the internet for Memrowave's web database. In terms of other features, the N.OM.S. project added an advertising scheme on the LCD screen that shows advertising from manufacturers, consumer specific coupons, other forms of entertainment that would be shown as the food was being cooked, with the Memrowave, phone or mobile device notifications through either Bluetooth, WIFI, or SMS mobile messaging is a feature being considered.

3.1.3 Current Microwave Interfaces and Functionality

Before beginning the design for the smart microwave interface, it is important to recognize that not all functionality will be able to be met by the smart features. Many food items that a consumer may want to prepare in a microwave, for example baked potatoes and restaurant leftovers, will not come with a barcode. Therefore manual features and controls must be provided for the smart microwave in order for it to provide a complete solution. To understand what features and controls exist on current microwaves, two existing microwaves were examined and their features were evaluated for inclusion in the features set of the smart microwave.

3.1.3.1 Microwaves Examined

The first microwave evaluated was a Whirlpool over-the-range microwave hood combination (model MH1150MXT-2). This microwave was inexpensive enough that it was used in new home construction, but has more than just basic features. Physical features include a size of 1.5 cubic feet, 1000 watts cooking power, a turntable, and a cooking rack. The microwave includes many quick control features such as Popcorn, Baked Potato, Pizza, Defrost, and Reheat. Figure 3.1.3.1-1 shows the Whirlpool control panel.

The second microwave used for evaluation was a Panasonic Genius Inverter microwave (model NN-H965X ABH). This microwave is a mid- to upper-range countertop microwave. Physical features include 2.2 cubic feet interior, 1250 watts cooking power, and a temperature sensor that can be utilized during cooking or reheating food. The microwave includes quick control features such as Popcorn, Defrost, Keep Warm, Sensor Reheat, and Sensor Cook which covers nine commonly microwaved foods. The Panasonic control panel is shown in Figure 3.1.3.1-2.

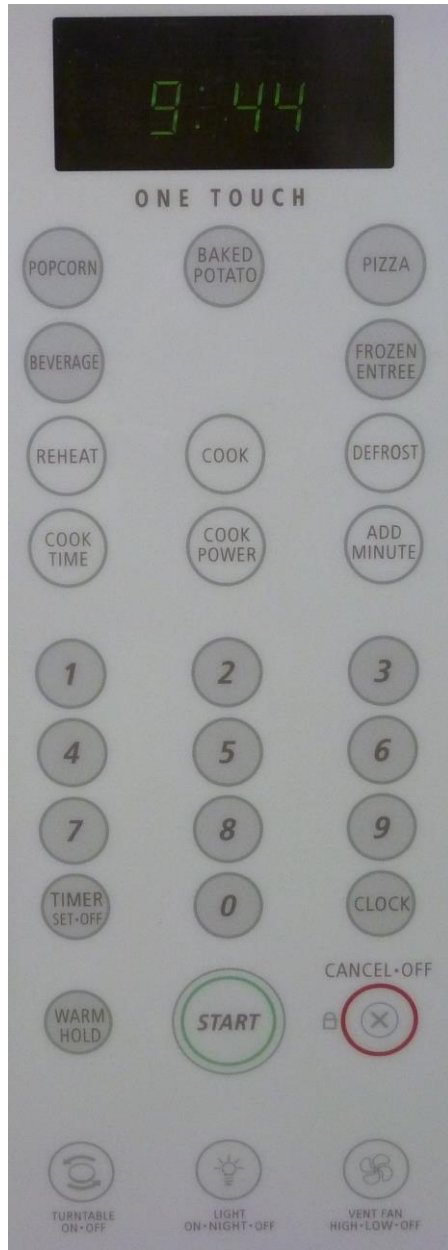


Figure 3.1.3.1-1:
Whirlpool control panel



Figure 3.1.3.1-2
Panasonic control panel

3.1.3.2 Basic Microwave Functionality

The basic control functionality of a microwave is as follows:

1. Select the amount of time to cook the food
2. Optionally, select the power level to cook the food at
3. Start the microwave

As simple as this basic functionality is, the microwaves being examined vary in their implementations. The Whirlpool has separate Cook Time and Cook Power buttons while the Panasonic does not have a distinct cook time button. Setting cook time on the Whirlpool microwave can be accomplished in two ways. The first method is to select the Cook Time button followed by the appropriate numbers on the control panel number pad. The second method assumes that no other buttons have been pressed, i.e. the microwave control panel is in its idle state. With this method, pressing the numbers on the control panel number pad is all that is required. In both cases the power level is displayed as a small number to the right of the input cooking time. Setting the power level starts with pressing the Cook Power button, but diverges to two methods after that point. The user can either keep pressing the Cook Power button to reduce the power in steps or the user can enter the power directly using the control panel number pad. The user can also do both or neither. The Whirlpool microwave basic functionality is shown in Figure 3.1.3.2-1.

The Panasonic microwave takes a narrower approach. There is no cook time button. Number keypad presses always results in a change in the cook time. This can occur because at no time is the control pad number pad used to enter the cook power. Cook power is only adjusted by repeatedly pressing the Power Level button. The result of this design decision is both good and bad. The good result is there is less opportunity for a mistake such as pressing a number pad button that unintentionally changes the power level. The negative consequence is there is less flexibility in how the user inputs the cooking parameters. The Panasonic microwave basic functionality is shown in Figure 3.1.3.2-2.

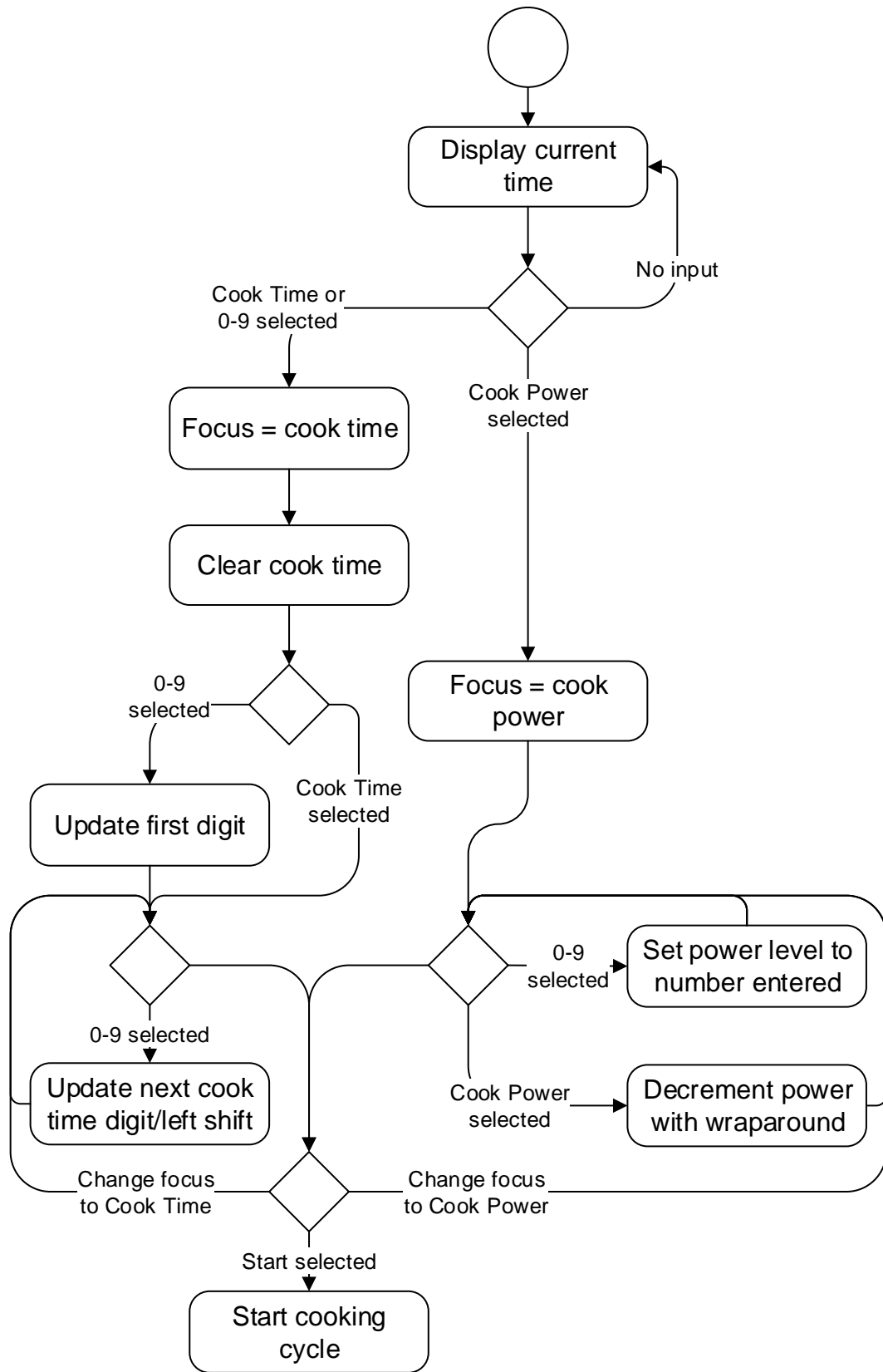


Figure 3.1.3.2-1: Whirlpool basic control

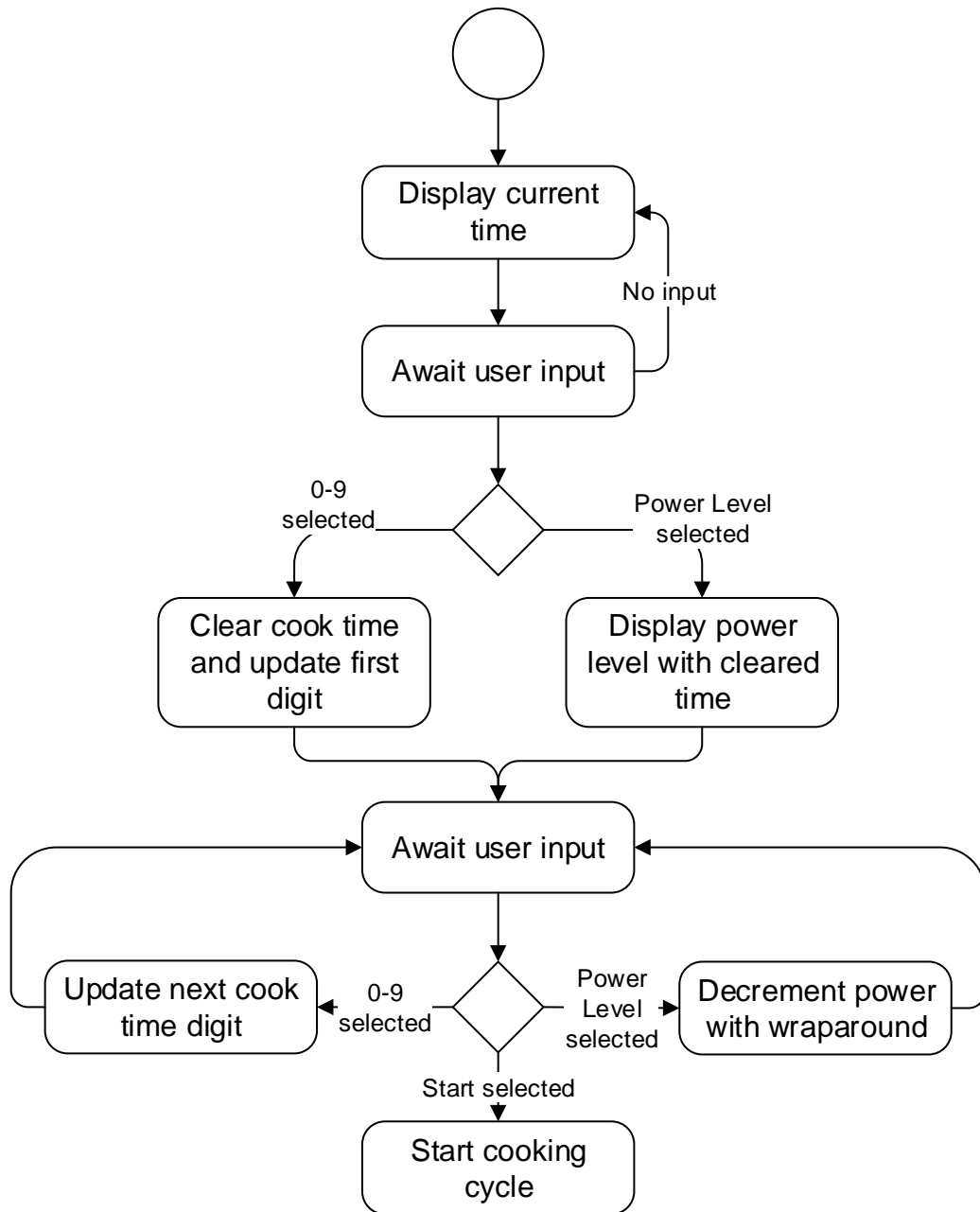


Figure 3.1.3.2-2: Panasonic basic control

3.1.3.3 Advanced Microwave Functionality

With the lessons learned while examining basic functionality in mind, the functional differences and similarities of the more advanced control features were compared and evaluated for possible inclusion in the smart microwave interface. The first subset of features shown in Table 3.1.3.3-1 are from the Whirlpool microwave.

Dedicated button	Power (%)	# of Key Presses	Preset Options	Time (mm:ss)
Beverage	70	1	1 cup	1:25
		2	2 cups	2:45
Frozen Entrees	100	1	10 oz	7:45
		2	20 oz	12:30
Frozen Pizza	70	1	1 slice	2:00
		2	2 slices	3:40
		3	3 slices	5:30
Potatoes	100	1	1 pcs	7:00
		2	2 pcs	9:00
		3	3 pcs	12:00
		4	4 pcs	15:00
Popcorn	100	1	3.50 oz	2:10
		2	3.00 oz	2:05
		3	1.75 oz	1:30

Table 3.1.3.3-1: Whirlpool dedicated button preset functions

These features have dedicated buttons and are designed for quick access to preparing specific foods. The user simply presses a dedicated button multiple times to cycle through the number of servings supported. Some of these foods will likely be handled by the automated feature of the smart microwave, but not all of these foods can be handled by reading a barcode.

The Panasonic microwave only has a single dedicated button to prepare a specific food: popcorn. While the Panasonic microwave is considered a better featured model than the Whirlpool microwave, it appears to be a deliberate design decision to bury most of the advanced features behind a menu system in order to present a cleaner panel. The popcorn preset options are shown in Table 3.1.3.3-2. Note that there is no standard time since this advanced function relies on the built-in sensor to determine the length of time to apply cook power.

Dedicated button	Power (%)	# of Key Presses	Preset Options
Popcorn	100	1	3.50 oz
		2	3.00 oz
		3	1.75 oz

Table 3.1.3.3-2: Panasonic dedicated button preset functions

Dedicated buttons are convenient, but it is not possible to provide a dedicated button for every preset function that a manufacturer wants to include in their microwave. This leads to the next set of advanced functions which are the general food preparation presets. These functions are usually accessed by first pressing a single master button multiple times to cycle through the various types of foods that have identified presets. In the case of the Whirlpool microwave there are two ways to select the preset function; cycling through presets using multiple presses of the Cook button and also pressing a number on the keypad to jump directly to the preset. The second option requires that the user has memorized which number selects the desired preset function. This is too onerous considering there are only five Cook preset functions, but it seems unlikely that a user will memorize an option unless it is used regularly. After the user selects the type of food to prepare, a quantity is entered using the keypad. The Cook button preset functions are shown in Table 3.1.3.3-3. The number in the Type column indicates the number of times to press the Cook button again or the number to press on the keypad after the Cook button is pressed just once.

Cook button	Type	Power (%)	Servings Supported	Time (mm:ss)
Rice	1	100	0.5 oz	21:30
			1.0 oz	23:40
			1.5 oz	26:00
			2.0 oz	27:40
Fresh Vegetables	2	100	1 cup	2:00
			2 cups	3:15
			3 cups	5:00
			4 cups	7:30
Frozen Vegetables	3	100	1 cup	3:00
			2 cups	5:15
			3 cups	7:00
			4 cups	9:15
Canned Vegetables	4	100	1 cup	2:30
			2 cups	4:30
			3 cups	6:15
			4 cups	8:15
Bacon	5	100	1 slice	1:05
			2 slices	1:45
			3 slices	2:10
			4 slices	2:45
			5 slices	3:30
			6 slices	4:15

Table 3.1.3.3-3: Whirlpool Cook button preset functions

The Panasonic microwave has a Sensor Cook button that takes the simpler approach of only cycling through the supported preset functions. If a user

frequently prepared pasta using this microwave, it would likely be annoying to have to press the Sensor Cook button nine times every time. All preset functions also rely on the built-in sensor to determine the cook time. Utilizing the sensor removes the need to indicate the quantity of food that the user places in the microwave. The preset functions are shown in Table 3.1.3.3-4.

Note the Whirlpool's Cook function requires a bit more time using the control panel number pad. After Cook is pressed, the desired food is selected using the number pad, or Cook may be pressed repeatedly to cycle through all five of the available options. Finally, the quantity of food is entered using the number pad. Designing a feature that requires that many steps, given the limited display of this microwave, will likely result in reduced use of that feature.

Only some items such as Frozen Entree, Frozen Pizza, Canned Vegetables, and Popcorn stand out as candidates for the automated side of the smart microwave. Other items such as Oatmeal and Bacon may be candidates for automation depending on the form they take. The remaining items will need to be evaluated individually for inclusion in the manual side of the control software developed for this project. Functions that are shared by both microwaves are particularly good candidates for inclusion into the Memrowave interface since these are likely expected by many consumers. The challenge will be to include this functionality without resorting to a selection menu system that is too deep.

Sensor Cook button	# of Key Presses	Power (%)
Oatmeal	1	80
Breakfast Sausage	2	60
Omelet	3	60
Frozen Entrees	4	60
Frozen Pizza	5	80
Potatoes	6	80
Fresh Vegetables	7	80
Frozen Vegetables	8	80
Pasta	9	80

Table 3.1.3.3-4: Panasonic Sensor Cook button preset functions

3.1.3.4 Reheat Functionality

A reheat function is provided by each microwave. The Whirlpool's reheat function shown in Table 3.1.3.4-1 utilizes steps similar to those that were used with the Cook button. The user first presses the Reheat button, followed by a number from 1 to 4 to select the food type, and finally a number indicating the number of servings being reheated. Once again the limited display of the microwave makes these functions a bit more difficult to use and therefore too easy to just ignore. Oddly, the final option, Dinner Plate, requires the user to input the quantity of 1 even though that is the only option available. Perhaps this is an idiosyncrasy of this model and larger models that use the same software have the option for more than one serving.

The Panasonic leverages its sensor for this function. The user simply presses the Sensor Reheat button and then Start. The time and power settings are completely controlled by the microwave. There are no other options to set. This is the kind of simple functionality that is the goal of the smart microwave just extended to all types of foods and food preparation requirements. Reheating food is a vital function of any microwave and a robust reheat feature needs to be included in the smart microwave's software.

<i>Reheat</i> button	Type	Power (%)	Servings	Time (mm:ss)
Soup/Sauce	1	70	1	2:15
			2	4:15
			3	6:00
			4	7:20
Casserole	2	70	1	3:45
			2	6:30
			3	9:00
			4	11:30
Baked Goods	3	70	1	0:10
			2	0:15
			3	0:30
			4	0:40
			5	0:50
			6	1:00
Dinner Plate	4	70	1	5:15

Table 3.1.3.4-1: Whirlpool Reheat function

3.1.3.5 Defrost Functionality

The last major microwave feature is the defrost function. The Panasonic cannot use its sensor for this function since that sensor relies on the generation of steam from the food being prepared. Both microwaves rely on the user to enter the weight of the food to defrost, but the Whirlpool further subdivides the foods into three categories: meat, poultry, and fish. The Panasonic makes no distinction and relies solely on weight. Using the defrost feature on the Panasonic microwave requires that the user press the Inverter Turbo Defrost button followed by the appropriate weight using the control panel number pad. No other options are available. For the Whirlpool microwave, the user first presses the Defrost button. The user can repeatedly press the Defrost button to cycle through the three options - Meat, Poultry, Fish - or press the corresponding number on the number pad. The user then enters the appropriate weight using the number pad. Defrost functionality is important for any microwave. It is simple enough that

software should be written to mimic the functionality that comes with the microwave that will be modified for this project.

3.1.3.6 *Miscellaneous Microwave Functionality*

There are several miscellaneous features that are common to both microwaves although implementations vary. These features are:

- Keep warm: Uses low/intermittent power to keep cooked food warm until it is removed from the microwave
- Timer (hold): After cooking, a timer will continue counting down and notify the user the food is ready after the timer expires.
- More/less: Allows the user to make minor adjustments as needed based on past experience/use of the microwave and specific product. This feature appears on other kitchen appliances such as single serve coffee makers and so should be fairly familiar to many users.
- Quick (add) minute: Used for short and/or quick microwave jobs such as warming a cup of coffee or slightly reheating a meal that has sat out on the table too long.

3.1.3.6.1 Keep Warm

The keep warm feature on the Whirlpool microwave is an all or nothing proposition. Pressing the Warm Hold button turns the microwave on without further user input. The microwave continues to run, but the magnetron only comes on for a few seconds at a time. The Warm Hold can be used in conjunction with regular cooking by pressing the Warm Hold button after the regular cook time and power selections have been made and before Start is pressed.

The Panasonic microwave's Keep Warm feature allows the user to specify the warming time. The microwave uses a low power setting to keep the cooked food warm. The Keep Warm feature can be used in conjunction with regular cooking times by pressing the Keep Warm feature after cook time and power selections have been made and before Start is pressed. The warming time is set after pressing the Keep Warm button.

3.1.3.6.2 Timer (Hold)

The timer feature on the Whirlpool microwave is not what a typical user might expect. It is an independent timer that cannot be incorporated into a food preparation sequence. Since this is an independent timer, use of the microwave is not hindered by a timer counting down. The design decision to make the Whirlpool microwave's timer an independent timer was likely due to its location.

As an over-the-range microwave, it is likely that users will want to utilize the timer for more than just preparing food in the microwave. If a user wanted to cook food in the microwave for 20 minutes followed by a 5 minute hold, the only way to accomplish this would be to start the timer at 25 minutes followed immediately by starting the 20 minutes cooking process.

The Panasonic microwave's timer feature is not an independent timer and can be utilized as part of the cooking process. Using the same example used for the Whirlpool, a user would set the 20 minute cooking time then press the Timer button and set its separate time for 5 minutes. The downside is that the microwave cannot be used while the timer is running. A solution that provides both an independent countdown timer like the Whirlpool microwave and a hold timer than can be inserted as part of the food preparation sequence like the Panasonic microwave is preferable.

3.1.3.6.3 More/Less

The Whirlpool implements this feature without using a separate button and does so in an unintuitive way by using the Cook Power button rather than the Cook Time button. After pressing an automatic cooking option, for example Popcorn, the Cook Power button can be pressed to change the cooking time to more, less, or back to normal. This is likely a very often missed and unused feature of this microwave.

The Panasonic microwave has a separate More/Less button that allows the user to make slight adjustments to the cooking times based on past experience. For example, if the previous bag of popcorn prepared in the microwave was still furiously popping when the microwave stopped, the user may choose to add just a little more time the next time popcorn is prepared. The time change that the More/Less button has on cooking times varies depending on the selection. Popcorn can be +/-10 or 20 seconds while other selections such as Sensor Reheat may adjust by +/-10%.

3.1.3.6.4 Quick (Add) Minute

The Whirlpool microwave has an Add Minute which functions by adding a minute of time for each button press and, if the microwave is already not running, will also start the microwave cooking cycle. Additional minutes can then be added as desired while the microwave is running.

The Panasonic microwave implements this feature using the Quick Minute button. Each press of this button increments the cook time by a full minute. The cook time can be increased whether the microwave is running or has not yet started cooking. Pressing the Quick Minute button does not start the microwave.

3.1.3.7 Additional Functionality

A Cancel/Clear button is included with each microwave. This button cancels the cooking cycle if the microwave is running just as would happen if the door was opened. The button also is used to clear any options currently selected. The Whirlpool microwave includes buttons that are necessary for options that are included to fulfill its role as an over-the-range hood. These functions are Vent Fan High/Low/Off and Light On/Night/Off. This functionality will only have to be planned for if a similar over-the-range combination microwave is used for the Memrowave. The Whirlpool microwave also includes a button to turn the turntable off and on. This is likely a desired feature to have in a smaller microwave that may need to handle platters or trays that won't fit completely on the turntable. This could also be a good feature to include in a larger microwave such as the size of the Panasonic microwave. Both microwaves include a Clock button. It is the intention of the smart microwave to keep its time updated using its internet connectivity so this button will likely not be needed and may only be included in an Options section of the software. Finally, the Panasonic microwave has a Function button. All of these are software related options and are not used in normal microwave operations. These could be specified and included in an Options section of the software or not as desired.

Listed under Function (Panasonic) are:

1. Language choice
2. Pound/kilogram choice
3. Word speed (LCD display)
4. Menu action on/off (prompt display)
5. Child lock on/off
6. Beep on/off
7. Reminder beep on/off
8. Daylight saving on/off
9. Clock on/off
0. Demo mode on/off

3.2 Relevant Technologies

The Memrowave will be a conjunction of many different technologies into one 'smart' device. These technologies include magnetron control, ARM microprocessors, computer vision, display technologies, etc. In the following sections, we will go into detail about each relevant technology we researched while designing the Memrowave.

3.2.1 Magnetron Control

Our system brings together all of our subsystems through a centralized system controller. This controller will handle tasks such as interfacing with the food product database, communicating with user interfaces, managing Wi-Fi access, and most importantly the magnetron. Controlling of the magnetron is vital to the Memrowave. This section will discuss the possible ways of controlling a microwave's magnetron and how to best incorporate a system to control the magnetron. Figure 3.2.1-1 shows a diagram of the system collaboration.

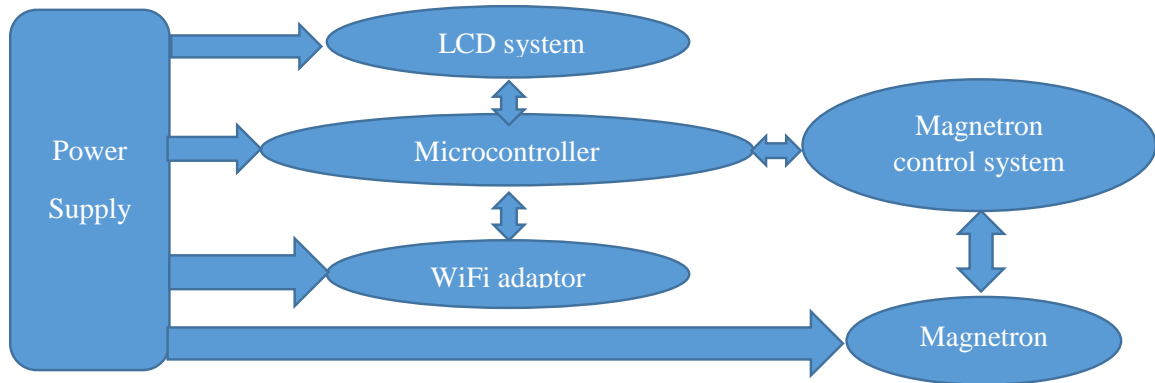


Figure 3.2.1-1: System control diagram

The magnetron is essentially an oscillator that produces a magnetic field and simultaneously an electric field. Average frequency ranges generated from the magnetron are around 0.5 GHz to 30 GHz however a specific magnetron will work a sole frequency. Magnetrons are used in many applications from military to consumer. They can be used as a sputtering system or a cooking system. The Magnetron is made up of a cathode, anode, resonant cavities, permanent magnet, and filament. The anode servers as the positive source in the system while the cathode servers as the negative. The Anode is usually made from a copper block. The filament leads are used to keep structure order. The resonance cavities of the magnetron are what determine the center frequency of the magnetron. With all these elements working simultaneously electric and magnetic fields can be controlled and created by the magnetron interacting with electrons. In order to operate the magnetron current must flow through the filament and the cathode. Electron flow through the filament will transport electrons between the anode and cathode. Current is generated from this interaction; with a presence of current a magnetic field is formed. The permanent magnet will alter the flow of current causing a circular path of flow. This path is a function of magnetic field and electron mobility. Oscillations are created from the resonance cavities in the device. This will oscillate the charges thus cooking the food stored in the path of the generated RF power.

When reaching this possible solution to magnetron control. A place to begin with is whether or not we will create our own magnetron or if the Memrowave will incorporate the built in magnetron system from the originally purchased microwave. Building our own magnetron for our device is not practical or safe for our needs. It requires our team to acquire unnecessary parts that doesn't add to the overall goal of the device. The Memrowave's focus is to make the microwave a more effective home appliance and not to redefine the microwave. Another negative aspect of attempting to create our own magnetron is the addition of more research time and man hours. Time spent working to create a magnetron can be better used in other places of our design. There is also no foreseeable engineering advantage to constructing a magnetron. Finally this option will add unnecessary cost to the group's budget. Keeping cost down will allow us to add more features to the Memrowave thus adding better parts. Such parts may be; higher end Wi-Fi adaptor, a larger order of surplus materials, and larger storage devices for the food product database. Excluding the creation of our own magnetron will increasing the overall effectiveness of the device. Without a benefit or incentive to producing a magnetron the Memrowave will not follow this route.

It is determined that salvaging the magnetron system from the originally purchased microwave yields the more effective outcome to our design. The next step is to review the possible ways of controlling the Memrowave's magnetron. If the design only required the magnetron to run at max power the system would only require a signal to turn on and off the magnetron. However to best meet our design we require the Memrowave to cook off the exact instructions of the product being considered. For example if the product requires a cooking time of 10 minutes for a 1000 Watts microwave at three fourths power. When scanned the Memrowave prepares the instructions and begins to cook at 750 Watts, the only way to achieve this is to regulate the power into the magnetron. In order to regulate our power we must first decide on what type of signal modulation we will choose.

The three general types of modulation methods that are available are Analog modulation methods, Digital Modulation methods, and Pulse modulation methods. In regulating our magnetrons input power we will stick with Pulse modulation methods. The reasoning is based off of research conducted showing that a pulse modulation is effective for many applications that require power delivery such as motor controllers and in this case magnetrons. Pulse modulation can also be broken down into sub divisions. These are Pulse-amplitude modulation (PAM), Pulse-width modulation (PWM) similar to Pulse-depth modulation (PDM), and Pulse-position modulation (PPM). Our task is to determine which method is best for power regulation of the Memrowave's magnetron.

Out of the three pulse-width modulation satisfies the solution of regulating the power to the magnetron. The benefit of using pulse-width modulation is from

being able to control the amount of power transferred to a load and not having any losses from regulation by a resistive method. Using pulse width modulation to control our power input to the magnetron will allow us to implement the feature of changing power levels and gives us better control of the magnetron.

Using pulse width modulation as a possible way of implanting magnetron control is proving to be one of the better options based off of our research on power controlling. The next step in designing a magnetron control unit is to weight the options between analog pulse width modulation and digital pulse width modulation. Pulse width modulation has been primarily dominated by the analog version for a substantial amount of time; however, the digital version is beginning to become more popular. Since Intel has begun utilizing digital PWMs, the digital version is bond to only grow more. With that said a reason to pick digital over analog is to follow the trends in larger companies. Please note that the difference between digital and analog is how the chip interprets the data. Either a voltage signal (analog) of data in terms of ones and zeros (digital).

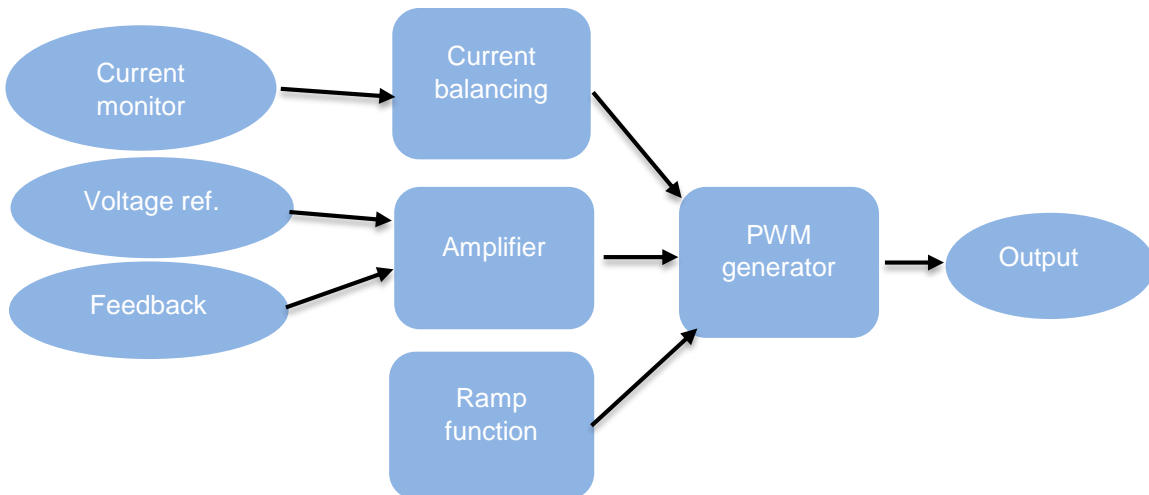


Figure 3.2.1-2:: Analog PWM

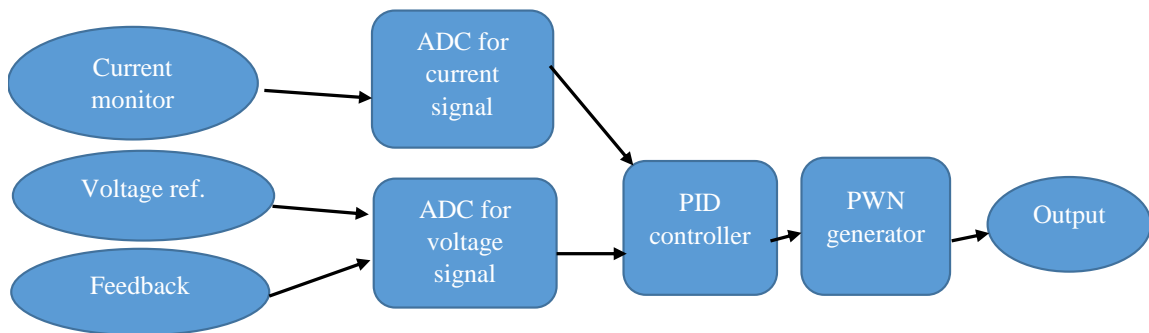


Figure 3.2.1-3: Digital PWM

With a layout of the differences of the two methods in a flow chart we can now make a better decision between the two. So what are the direct benefits of analog versus digital and vice versa? When designing a system the question about speed is very important, we want to be able to process our data as fast as possible. Keeping our speed up certainly has positive benefits that is why speed is a positive aspect of an analog PWM. Analog PWM's provide one hundred percent faster hardware design. The Implantation of our design is another huge factor that goes into the Memrowave. We want to incorporate an effective yet simple design that gets the job done. Choosing materials that are easier to work with is a deciding factor for what is placed in our design. A benefit of using an analog PWM is that they are pre-programmed therefore, much easier for us to implement into our design. Having a pre-programmed chip adds a considerable amount of time for work elsewhere because coding development time for pulse width modulation is completely taken out of the process. Having a pre-programmed chip however, may hinder the project because it limits what we customize on the chip. With analog PWM's calculation errors are few and far in between. Having minimal errors in calculations means a more effective product. If for instance the power to the magnetron is incorrect food products may be overcooked or undercooked. No possible calculation errors is one of the reasons we may use an analog chip to control the magnetron. It is for these reasons using an analog PWM chip is a possibility for our magnetron control.

Now that we have listed the possible reasons for using an analog PWM to control the magnetron here are some of the reasons to use a digital PWM chip for magnetron control. Data commutation is of up most importance when it comes to a design, using materials that provide the Memrowave with the best possible outputs is a necessity. When using a digital design all information is interpreted as series of ones and zeros, therefore allowing for more certainty. Using digital pulse width modulation chips will provide the Memrowave with a higher precision and accuracy when carrying out calculations. Keeping a high precision and accuracy means that the majority of our data will return with similar results to the expected answer and within a very small margin of error compared to all the data calculated. Digital PWM's also have the ability to predict transient loads. Another benefit of using a digital PWM is from its ability to control a larger majority of the Voltage regulator module (VRM) for any possible time and process all given inputs. Having the ability to edit and change a design on the move is a top priority for the Memrowave, we need a design that can adapt when need and adapt for very little cost in a short amount of time. Be able to use a device that allows our design to meet these requirements sets that product at the top of the list. Having a PWM chip that can meet this requirement will help with any obstacles we face. Using a digital pulse width modulation chip grants the ability to do this. For our design to be most effective it must meet whatever obstacles it faces therefore, having the ability to re-program the digital pulse width modulation chip in the field and tailor it to the individual needs of each situation provides the biggest possible benefit for using it. With all the advantages of using each method known we can

now proceed in deciding which method better fits the Memrowave's needs. Before we make a decision we will look at possible vendors for each type.

ANALOG	DIGITAL
Intersil (Ziker Labs)	International Rectifier(IR)
uPI Semiconductor	CHiL
Analog Devices	Volterra

Table 3.2.1-1: Manufacturers

This gives us six options to use when designing the Memrowave's magnetron control. Here are examples of specific chips that we can use in our design.

DEVICE MANUFACTOR	TYPE	CHARACTERISTICS	COMMONLY FOUND
STMicroelectronics	Analog	L6718 – a 4+1 phase analog PWM	ASRock Z77 Pro4
Intersil	Analog	ISL6366 – a 6+1 phase analog PWM	GIGABYTE Z68X-UD7
uPI Semiconductor	Analog	uP1618 – a 6+2 phase analog PWM	MSI X79
Internation Rectifier	Digital	IR3567 – a 6+2 phase Digital PWM	GIGABYTE Z77X-UD5H
CHiL	Digital	CHL8328 – a 7+1 phase digital PWM	ASUS Maximus 4 Extreme
Texas Instruments	Digital	MSP430	LaunchPad

Table 3.2.1-2: Manufacturers, Types, and Descriptions

Now that we have gone through the types of PWM we may possible use in controlling the Memrowave's magnetron we can decide on what might be used. The first option to use will be using digital pulse width modulation. This is because of the benefits that it provides. Being able to re-program the PWM in the field is the main reason for using this method. Since the Memrowave is in

prototype this feature is a must. Along with that advantage utilizing digital's higher precision and accuracy will yield better consistency in our prototype.

The first option for utilizing a digital PWM is to use an MSP430 microcontroller to program the PWM. A reason for using this option first is because this microcontroller is readily available and our team has the most experience using this microcontroller. Microcontrollers are also low-cost, low power, small, and easy to implant as a solution. There are also other types of microcontrollers that can be used. First we can use an Arduino; this device is simple and effective for the use as our PWM generator. These are cheap and easy to obtain thus adding to the possibility of use. The Arduino also has many tutorials on usage and creation. However not everyone in our team does have as much experience as needed. Since the MSP430 is a tool used by everyone in our design team this makes it a better option than the Arduino. Next we can use a Raspberry Pi, these devices are more expensive than the Arduino and the MSP430. Also not everyone in our team does have as much experience as needed. Since the MSP430 is a tool used by everyone in our design team this makes it a better option than the Raspberry Pi. Finally an option would be to use a BeagleBone Black as the source of the magnetron power controller. These devices are much more expensive than the Arduino, Raspberry Pi, and the MSP430. They are also very powerful and would not fit to being only a source for a PWM. We could use the same BeagleBone Black that we use for the camera and LCD screen however we would like to limit as much functions on one device as possible. It is for these reasons the optimal choice for our Memrowave will be using digital pulse width modulation, a microcontroller as the main housing, and the MSP430 as the final product to create a PWM and control magnetron output power. With all these products and options our decision on magnetron control is complete. The next discussion on the magnetron and its controller will be the implementation and description on how we will create the PWM, order the parts and compare it to using an LM555 timer.

3.2.2 ARM Microprocessors

The Memrowave's interface will be handled by an ARM microprocessor. ARM microprocessors are the most widely used microprocessors in the world of mobile handheld devices. They are reduced instruction set computing (RISC) processors that use fewer transistors than typical desktop computing processors resulting in lower cost and lower power requirements which is desirable for the Memrowave. Features and processing power allow ARM processors to run operating systems such as embedded Linux and Android which increases functionality and usability in applications such as the Memrowave. The prevalence of ARM processors in the marketplace should allow for multiple choices in software and connecting hardware during the design phase.

3.2.3 Computer Vision for Barcodes and QR Codes

Since we've decided to implement the barcode and QR code reader using a camera, we need some kind of computer vision solution to read them. Luckily, there is an open source, multi-format barcode image processing library available for Android, called ZXing. This library supports UPC-A, UPC-E, QR Codes, and a few other barcode formats.

3.2.4 Output Display Technologies

The Memrowave's primary interface will need to display the software interface including pictures of food products, bar codes, and QR codes. The output display can be handled by a number of technologies. It is possible at the start to conclude that since the device will reside within the Memrowave it should be a direct display and not a projected display. Direct display technologies include emissive technologies such as organic light emitting diode as well as non-emissive technologies such as liquid crystal display and electronic ink. Since it is desirable to have the Memrowave display color photos of food products, e-ink is eliminated. That leaves two possible displays: OLED and LCD.

3.2.4.1 Organic Light Emitting Diode Display Overview

Most emissive display technologies don't have the capability to display a satisfactory interface for the Memrowave due to lack of colors or pixel density. This is not the case for OLED displays which have recently seen increasing use in many applications such as mobile devices and televisions. As an emissive display technology, OLED displays don't require external/backlighting in order to be viewed. Lack of backlighting usually results in a thinner display and more energy efficiency. The viewing angle for OLED displays is extremely wide so that a Memrowave user could be standing to the side of the device and still be able to clearly view the display. Response time is outstanding in the sub-millisecond range. Contrast and color display capability are both excellent which would enhance the display of food product photos used by the Memrowave interface. OLED displays also have excellent black levels which would make display of the bar and QR codes look sharper. Some OLED negatives include variations in the aging of the different colors. Unfortunately OLED displays tend to be more expensive and availability is limited at this time.

3.2.4.2 Liquid Crystal Display Overview

The liquid crystal display is a mature technology that is currently prevalent in the smart device market as well as other markets. As a non-emissive technology, LCD must utilize an external light source in order to view the display. This backlighting does add some additional thickness and weight to the hardware, but this shouldn't be a problem since the Memrowave is not a handheld device. The

backlight also reduces black level representation and can cause colors to look a bit more washed out. Many backlit screens can also be hard to see in bright lights, and kitchens are usually brightly lit, so visibility could be an issue. Some clarity issues can be countered by using thin-film transistor (TFT) technology, an active-matrix technology, which provides a brighter and sharper display and would be desirable at small display sizes such as used in the Memrowave. Another problem with LCD visibility can be a somewhat limited viewing angle. This can be highly dependent on the quality of the LCD. The number of colors that can be represented is quite good and should be more than adequate to display the pictures used by the Memrowave. Response time is fair and should be acceptable since the Memrowave will not be utilizing moving images like a television

3.2.4.3 Output Display Technologies Summary

OLED displays have more favorable features such as better visibility in bright light and wider viewing angles than LCD displays. OLEDs are also thinner since they do not require a backlight. Unfortunately OLED displays are currently have limited availability and are higher priced than LCD displays. Overall price is a factor for the Memrowave therefore a liquid crystal display will be chosen for the project.

3.2.5 Input Touch Screen Technologies

Primary input for the Memrowave will be accomplished with a touch screen device. These input devices are normally integrated with a display, such as a liquid crystal display, but the technology is distinct from the display technology that it may be used with. It is even possible to add a touch screen input device to a display that does not have integrated touch input. The two major touch screen technologies that could be used for the Memrowave are resistive and capacitive.

3.2.5.1 Resistive Touch Screen Overview

Resistive touch screens consist of two thin layers of material, usually transparent, that are separated by a thin gap so they are not touching. The inward facing surfaces of both layers are electrically resistive. The concept is that when a user presses on the top layer using a finger or stylus, the two layers make contact and the location of that contact can be determined. This design is an advantage since the location of the press can be determined regardless of the item pressing the layers together. A finger in an oven mitt or covered in flour is just as capable of being detected as a clean finger or a stylus. This would be desirable for the Memrowave since it is a kitchen appliance. Resistive touch screens are also fairly inexpensive technology and cost is a factor for this project. Disadvantages of resistive touch screens are that the two additional layers of material reduces

the contrast of whatever display they are placed over and they can easily be damaged by sharp objects.

3.2.5.2 Capacitive Touch Screen Overview

Capacitive touch screens consist of a hard insulating layer such a glass with a transparent electrically conductive coating. The electrostatic field of the touch screen distorts when another electrical conductor such as a human finger comes in contact with the surface coating. This distortion can be processed to determine the location of the touch including the ability to sense multiple simultaneous touches. Since the surface is not a soft, flexible layer it is more resistant to damage by sharp objects. Contrast of an underlying display is not negatively impacted. However, since the technology requires a touch from an electrically conductive source, it would not detect a touch from a finger in an oven mitt and might not detect a touch by a finger covered with flour. Despite their prevalence in the market, capacitive touch screens tend to be more expensive than resistive touch screens.

3.2.5.3 Input Touch Screen Technologies Summary

Besides price, the biggest advantage of capacitive touch screens is the ability to detect multi-touch events which is something that resistive touch screens generally do not support. Multi-touch is not a feature that will be used by the Memrowave, but swiping will be. Swiping is the movement of a finger or stylus across the surface as an input to a device. It is extremely prevalent in many mobile devices today. Swiping is extremely difficult to do across a resistive touch screen unless the user is utilizing a stylus. This is because the surface is soft and it takes a bit of pressure to bring the two layers in contact with each other. The Memrowave will need to support swiping in order to make its interface as familiar and user friendly as possible. The ability to support swiping really is the decision point for the project; therefore a capacitive touch screen will be used.

3.2.6 Operating Systems

We have two options for the operating system of the Memrowave. We could design a custom operating system, or we can modify an existing operating system to suit our purposes.

Designing a custom operating system would most likely be too complex, because of all of the advanced features of the Memrowave. It includes a screen, so we'd presumably need to implement some kind of display driver. It includes internet access, so we would need to implement some kind of network interface driver. It has a touchscreen, so we'd need to implement some kind of driver for that. With the availability of open source operating systems designing a custom OS would really just add unnecessary work.

This leads to the selection of a pre-existing open-source operating system. The most obvious choice would be some Linux-based operating system, because of its ubiquitous support for ARM microcontrollers. In fact, there are a number of Linux-based operating system options available for the Beaglebone Black, including Ubuntu, Angstrom, and Android.

The main advantage of Ubuntu is its large support base. The Ubuntu software repositories include mostly everything we would need to design the software for the Memrowave. The Angstrom distribution is a relatively new distribution developed by a smaller team, so it generally cannot match the level of support Ubuntu offers; however, the Angstrom distribution was created with embedded systems in mind, so it will most likely be more lightweight than Ubuntu and optimized for embedded use.

The problem with using Ubuntu or Angstrom is that Linux is generally lacking in support for touch screen interfaces. Most graphical Linux interfaces are based on window systems, which generally do not work very well for touch screens. This means we would have to either develop our own touchscreen GUI framework or configure some third party GUI library before we could even begin to implement our user interface. Generally, programming the user interface without such a framework in place is not a good idea. While it definitely could be done, it would make it much more difficult to keep a consistent style and flow throughout the entire user interface. This is where Android really shines.

Android was created specifically for touch screen devices, so it includes all of the necessary libraries to implement a touch screen user interface with consistent styling and control flow. It also includes a network stack, local database support, and mostly every other feature we need in an operating system. If we were to use either the Ubuntu or Angstrom distributions, we would need to carefully select all of the different libraries that we would need to write all of the software for the system, and try to fit them together into a working product. With Android, this is not necessary, because it already includes all of the software libraries necessary to accomplish what we want. This is why we've chosen Android for the Memrowave's operating system.

3.2.7 Wi-Fi Interface

The Memrowave is intended to be a connected kitchen appliance. The primary use of this interface is to retrieve product records from a central database located on the Internet. This activity will only be necessary for scanned items that are not already in the local database. Given this small requirement, it is not necessary to have a high bandwidth solution. While it is possible to connect the Memrowave to a network using the Beaglebone Blacks' built-in Ethernet port, it is highly unlikely that most houses will have a twisted-pair network port located in the kitchen. Therefore, Wi-Fi will be used to achieve the required connectivity. Wi-Fi is a

widely used and supported wireless technology that connects many different computing devices to a local network.

3.2.7.1 IEEE 802.11

The standard associated with Wi-Fi is IEEE 802.11. This standard has been updated numerous times over the years with the specific version designated by the suffix, e.g. 802.11n. This suffix is used to distinguish Wi-Fi device support and capabilities. The more common versions in use today are shown in Table 3.2.7.1-1. While there are other published versions, some earlier and some more recent, they are not commonly available and there will be no effort to support them. Bandwidth values are not included in the table. Bandwidth is not of vital importance to the Memrowave since the actual amount of data being exchanged should be small.

802.11x	Freq (GHz)	Range (ft)
b	2.4	115
g	2.4	125
n	2.4	230
	5	230
ac	5	115

Table 3.2.7.1-1: Common Wi-Fi standards

While 802.11b and 802.11g are older standards dating back to 1999 and 2003 respectively, they still see some use in low cost consumer products and in older equipment that has not been replaced due to cost. For example, it is quite likely that some Internet service providers may still supply wireless access points that only support the 802.11b/g standards. IEEE 802.11n is unique among the version shown because it supports both 2.4GHz and 5GHz frequencies. This capability is called dual band support. It also has double the usable range of the other versions, even the newer 802.11ac. The latest standard that is seeing increasing deployment, 802.11ac, has the capability for higher bandwidth compared to the earlier versions, but bandwidth is not a major consideration for this project. Fortunately, 11ac devices normally also support 11n Wi-Fi at a minimum.

3.2.7.2 Wi-Fi Interference

One complaint often heard regarding microwaves is that they interfere with wireless communications, including Wi-Fi connections. The Memrowave will not be utilizing the Wi-Fi connection during cooking operations. The only time that

Internet communication will be needed is during the scan and identify process. Once a product record has been received from the online database and the settings are sent to the Memrowave's control board (by the user pressing Start), the user interface including the Wi-Fi will be placed on standby except for a cancel capability. The Memrowave control board will handle all microwave operations including time display and cancel via opening the cabinet door. Therefore, microwave interference with Wi-Fi connectivity is not an issue.

3.2.7.3 Wi-Fi Interface Summary

After reviewing the options, the best range of Wi-Fi support for the Memrowave is 802.11g/n. This gives a good range of access point support, provides adequate bandwidth, and provides the option of dual band via 11n. Having legacy support of 11g is important to remove a roadblock to possible adoption in the marketplace. Most older devices support both 11b and 11g therefore there is no explicit requirement to support 11b. The biggest reason to include 11n support is because of the increased range which is important since kitchen distance to a wireless access point, which may be at the opposite end of many houses, is likely not considered by many consumers. Dual band support is welcome, but not required since interference is not a concern. 11ac, while new and high bandwidth, just adds cost without any real benefit given the Memrowave's low bandwidth needs. The final consideration is the Wi-Fi adapter needs to be supported in the version of Android used for this project.

4 Project Design Details

In this section, we will begin to describe all of the details of the design process of the Memrowave, starting with our initial design approaches, continuing on to the hardware, and finishing with the software.

4.1 Initial Design Approaches

The Memrowave's main function is to integrate a high powered processing unit with a printed circuit board of our own design. The benefit of using this approach is that it requires a more powerful processor to accurately run the barcode scanning to determine what product is ready for use. If we chose to use a smaller embedded processor we predict that the overall design will be hindered. However we first did look into perusing a single embedded processor to be the main unit of the Memrowave. The main advantage of this route is form factor. Using one embedded system allows for only one overall printed circuit board design. Compared to using a BeagleBone Black integration which uses three separate layers as depicted in Figure 4.1-1.

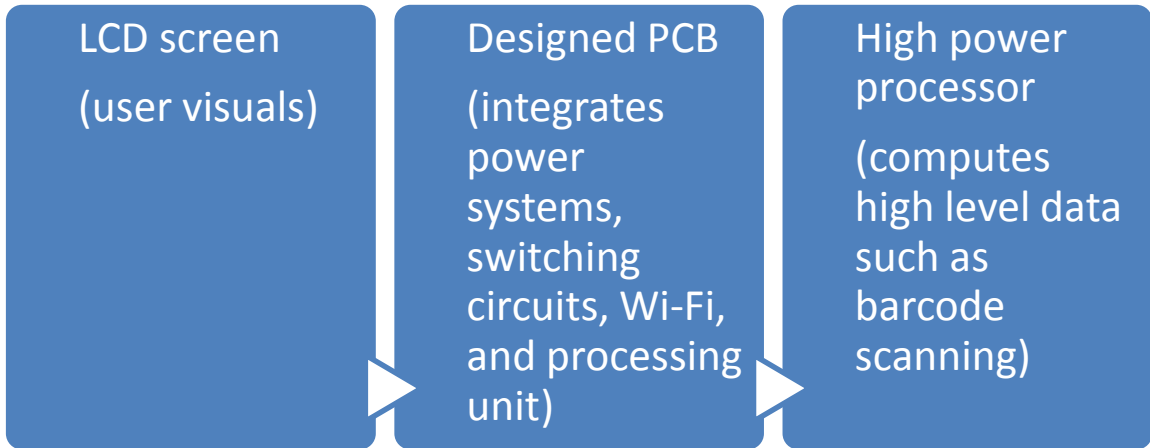


Figure 4.1-1: Possible integration of designs

However, the single benefit of using only one printed circuit board with the main processor integrated to it is not enough to merit use. The main disadvantage of it being a slower and less accurate method is too large. Therefore the Memrowave design team will be using a high powered process in combination with a team designed PCB.

Using this method requires our team to design the PCB that functions as the integration of all subsystems. The PCB integrates with the power supply, LCD screen, Wi-Fi, and the processing unit we use. This PCB also must function as a sandwich layer between our subsystems because the Processing unit and LCD screen will only have elements on one side. This forces our team to make the PCB which connects them from the middle.

Overall our design meets the requirements to create a new innovative microwave to help make daily life easier. This is accomplished through a unique integration of high level architecture and systems constructed by our team.

4.2 Hardware

In this section, we will detail the design process of the Memrowave's hardware. While designing the hardware, we be sure to meet all of our objectives and specifications outlined previously.

4.2.1 Power Supply

The power supply for the microwave must be able to power the microwave as well as the additional electronics and components we plan to add to the system. This includes the LCD display, processor unit, PCB circuitry, and WIFI adapter. The typical microwave has a power specification between 850-1800 watts of total

power, we would like our microwave to fall into these guidelines. The power supply must provide AC power to the magnetron, turntable, fans, and lights. DC power must be provided to the processing unit, camera, and LCD screen. Since the magnetron typically uses 4000 V at 300mA when at full power, a HV transformer circuit must be implemented. This can be done through using a printed circuit board with a voltage handling in the proper range since we want to keep a space-saving form-factor for a microwave. Considerations for developing this HV circuit are the dangers of direct arc over and corona production. When voltage potentials exceed the dielectric's ability to withstand it, arc over occurs. Corona discharge is a failure mechanism which causes degradation of the insulation system, essentially a precursor to direct arc over occurrence.

A solution to limit these issues is to use smooth round curves in the circuit design, in Figure 4.2.1-1 ideal and non-ideal ways of designing parts of the printed circuit are shown:

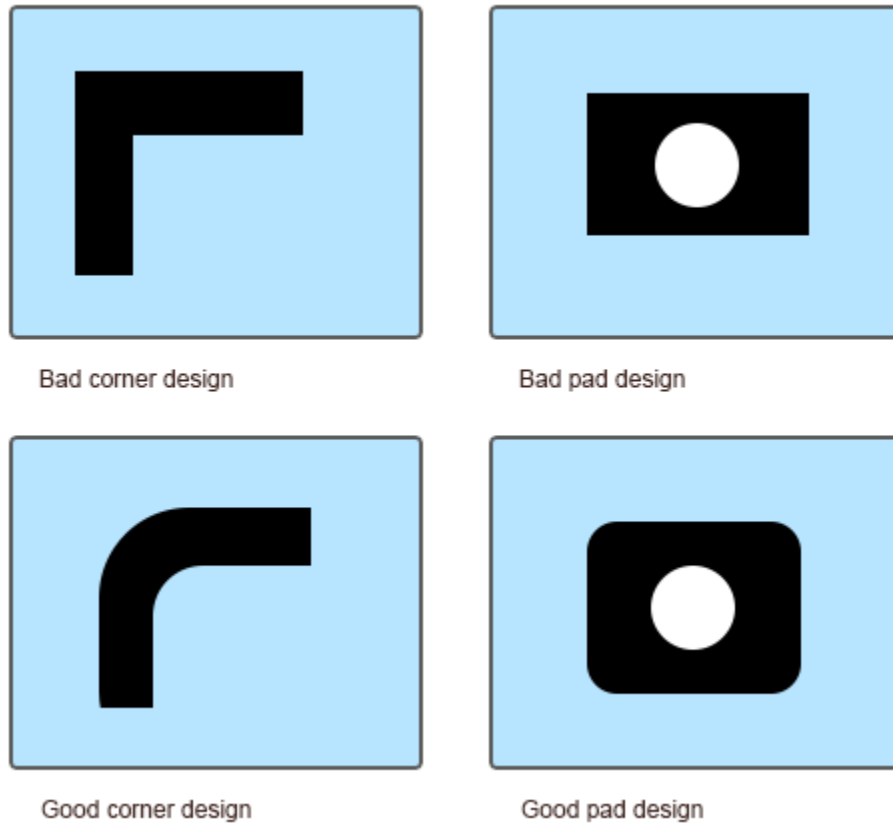


Figure 4.2.1-1: PCB Corner Design

All sharp edges should be avoided; rounded corners and pads should be designed.

Also in the printed circuit board for the power supply, a section for powering the DC components is necessary. Depending on the specific processing unit we plan on using, the voltages needed will range from 5-10 volts. Most processing units can be powered through a USB connection which usually also serves the purpose of porting information to the unit. Power can also be delivered through a power jack which is also integrated onto the boards. This is ideal for keeping the USB port free for possible software changes and updates through a wired connection. Shown below in Figure 4.2.1-2 is a picture of an BeagleBone Black development board, a possible processing unit, and the item boxed is the 2.1mm positive pole DC power jack which can be connected to the PCB board while leaving the USB port available for data transfer.

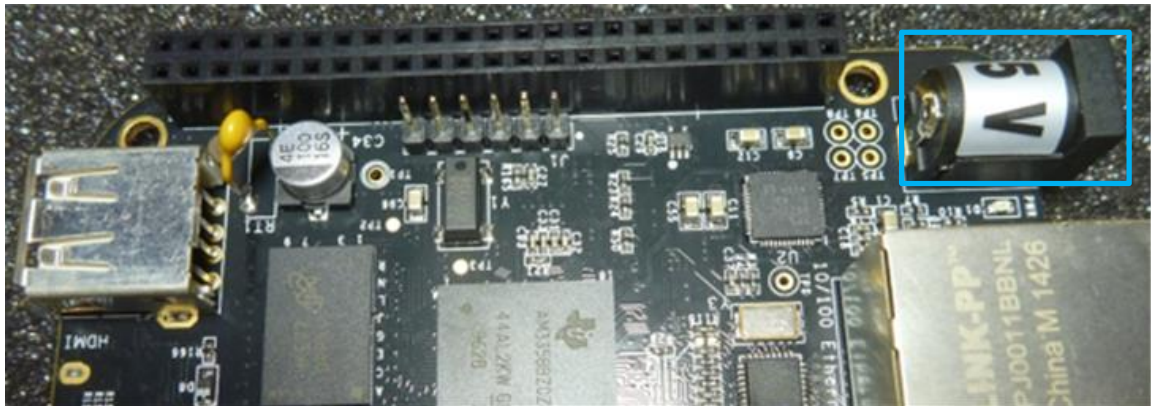


Figure 4.2.1-2: BeagleBone with built in power supply

A typical camera module which we could use as the Memrowave's QR code reader and scanner input voltages range from 6-20 volts DC and requires 50 milliamps typically. The camera can be wired directly to the PCB power supply board with current limiting or through a processing unit. Since the camera is used infrequently and under certain conditions, power saving opportunities are available. This would be ideal and easier to implement if power can be supplied through the processing unit. The camera can be implemented under the control of the processing unit only to be turned on or active while food products are about to be scanned or if the user comes into close proximity to the microwave, the microwave can be activated out of a power save mode for less power consumption. On the next page in Figure 4.2.1-3 is a picture of a usable camera module that fits into the power specifications.



Figure 4.2.1-3: Camera example (used with permission from SparkFun Electronics)

Powering a 4-7 inch LCD touchscreen will require a range of 4.0-6.0 volts DC and a minimum of 500 milliamps. Based on the processing unit we use for the Memrowave, a power will either need be power through PCB circuitry or the processing unit. Limits on the voltages the processor can supply will be the deciding factor in this decision but use of the processing unit as the power delivery system directly would make power saving much easier, due to the LCD touchscreen being another component that is used intermittently and also be adjusted for brightness, power savings would be possible.

A possible wireless network processor for creating WIFI connections will require 3-5 volts DC and a max current of 100 milliamps. Due to the necessary voltages and the close integration needed between the processing unit and the Wi-Fi module, power will most likely be provided through the processor. Our ideal Wi-Fi module will fit into these power specifications while providing the best integration and WIFI range and speed possible in our budget. Below in Figure 4.2.1-4 is a WIFI module made by Texas Instruments that fits into the power specifications.



Figure 4.2.1-4: Wi-Fi Module (used with permission from SparkFun Electronics)

Common parts of the typical microwaves such as the internal lights, beeper, and turntable will be powered through the AC outlet voltages and specific current loadings. Since each of these internal parts work intermittently, the task for switch them on and off will be done through the processing unit or other means, such as opening the microwave door to turn on the internal light.

Since both AC and DC power is needed within the Memrowave, a way for converting AC to DC is needed. An AC DC inverter will be used to convert the 120V AC domestic outlet to 12V-20V DC. The inverter must also be able to meet our total DC current needs as well. To acquire the multiple voltages needed for the DC devices, voltage regulation will used based on the specific DC devices that we will chose, ideally a maximum of three different voltage regulators should be sufficient. Another possible way of obtaining the dried voltages is to build the voltage dividers as well on the PCB, this option would save space as well as money and more accurate voltage could be produced. The use of voltage regulator components brings the advantage of a more robust system, a current loading limits can observed much easier based on the device specifications and with voltage regulators a more precision DC voltage can be obtained. Below in Figure 4.2.1-5 is a schematic of an AC-to-DC power supply. In Figure 4.2.1-6 is a 1A voltage regulator made with multiple voltage options which fits into the power specifications.

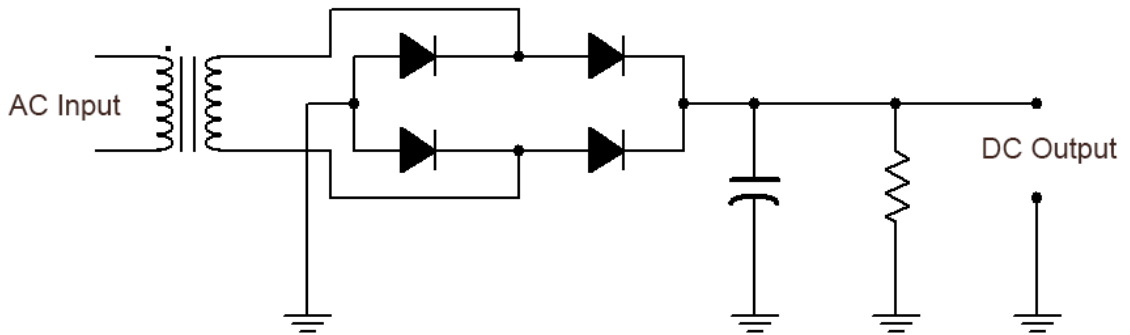


Figure 4.2.1-5: AC-to-DC power supply

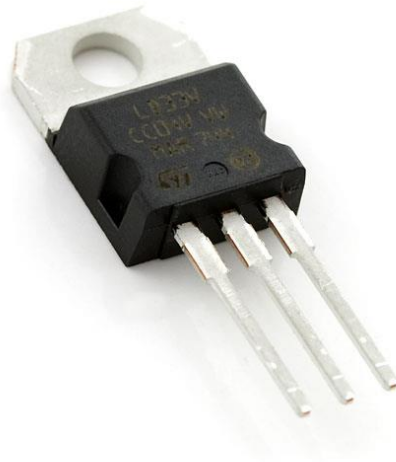


Figure 4.2.1-6: Voltage regulator (used with permission from SparkFun Electronics)

Below in Table 4.2.1-1 is a working summary of the main parts and component of the Memrowave including details on required voltages and currents.

Components	Voltage(V)	Current(mA)
Magnetron	4000 AC	300
Processing Unit	5-12 DC	30
LCD Touchscreen	4-6DC	500
Camera	6-20 DC	50
WIFI Module	3-5 DC	100
Internal Light	120DC	200
Beeper	120 DC	20
Turntable Motor	21 AC	150

Table 4.2.1-1: Microwave component power specifications

In Figure 4.2.1-7 is a power flow chart for all the electrical parts and components of the Memrowave.

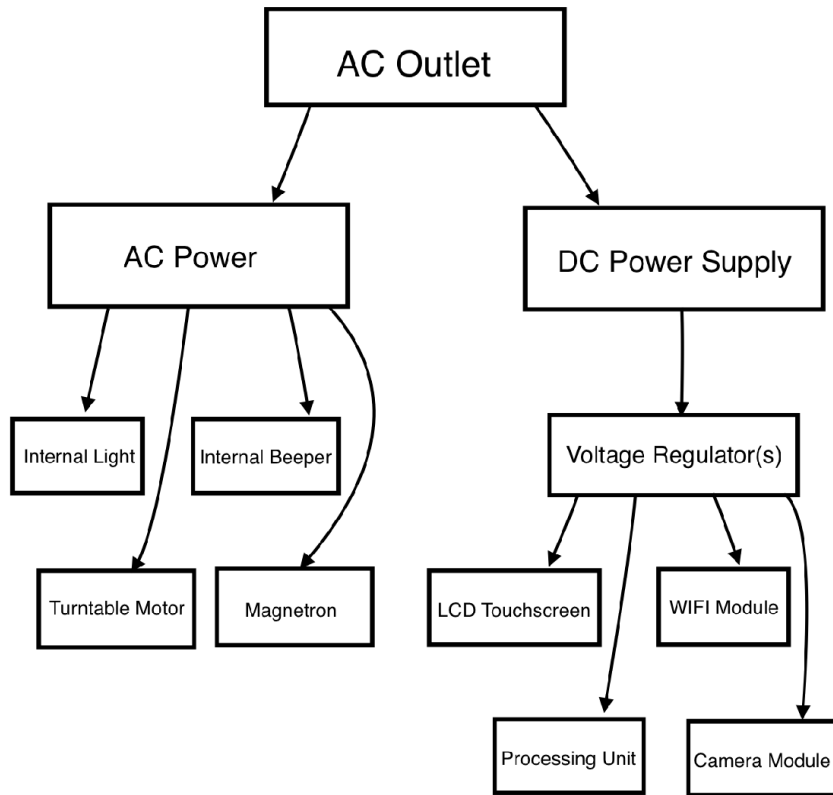


Figure 4.2.1-7: Power Flow Chart

4.2.2 Magnetron Controller

The controller that is most effective and practical for the Memrowave is the MSP430 created by Texas Instruments. Reasoning behind using the MSP430 to directly control the power delivered to the Magnetron is its cheap cost to obtain (Starting at \$9.99 according to Texas Instrument’s website), ease of use, availability, already used for other features in the Memrowave, and ability to program Pulse width modulation functionality.

MAIN BOARD

- Analog signal controlling the duty cycle of the PWM

MSP430

- PWM (power delivered)

Figure 4.2.2-1: Interactions between PWM and source signal

Before we continue another method to create a PWM signal other than programming a MSP430 is to use the LM555 timer. Figure 4.2.2-2 depicts the packaging and pin layout of the LM555 while Table 4.2.2-1 shows each pin's function.

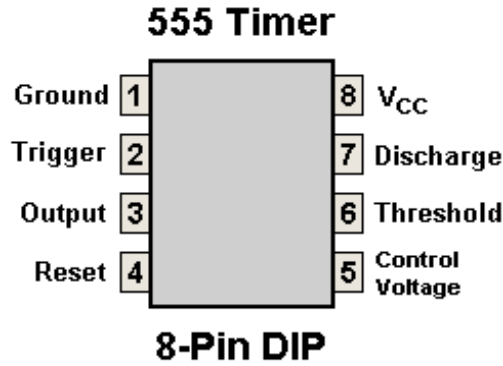


Figure 4.2.2-2: LM555 pins (used with permission from Texas Instruments)

Pin	Function
1	Connection to Ground on PCB
2	Connection to PCB
3	Connection to Magnetron
4	Connection to BeagleBone
5	Regulation from BeagleBone
6 7	Connection to PCB to BeagleBone
8	Connection to Power supply

Table 4.2.2-1: LM555 pins

This device is noted for being highly stable and therefore can generate accurate time delays or oscillations by applying a constant pulse with on and off phases. We can then modulate the pulse width with the signal. This signal is coupled to the control voltage pin of the LM555. The resulting output is then modulated. We can depict this using the following Figure 4.2.2-3.

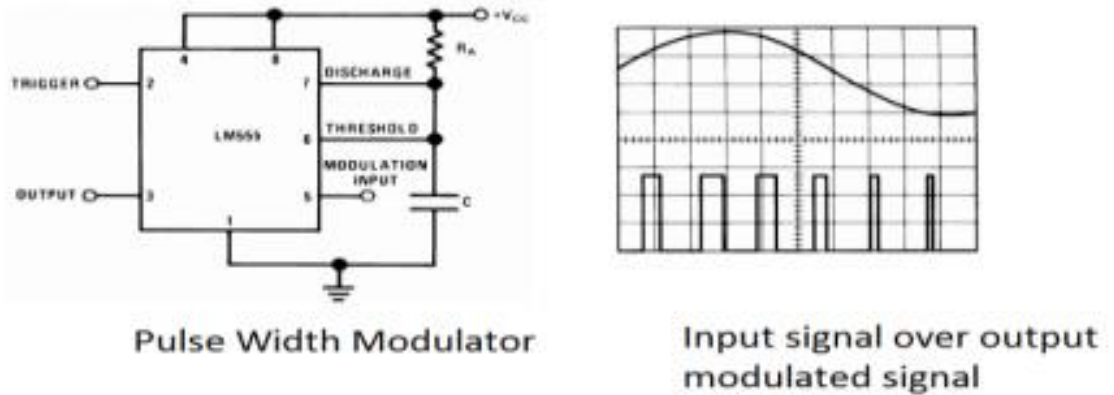


Figure 4.2.2-3: LM555 layout and example output (used with permission from Texas Instruments)

Mounting of the LM555 timer will be done using a PCB fabricated from a chosen vendor. The circuit in Figure 4.2.2-3 will be implemented directly to a PCB using surface mounted parts. Using surface mounted resistors and capacitors gives the board and circuit a sleeker and slimmer look. Having a slim board is helpful since space is limited in the microwave. The Table 4.2.2-1 lists the pin locations that will be connected to the PCB and where the connection wires will lead.

Due to ease of implementation however we have elected to use the MSP430 to control power to the magnetron. The MSP430 generates the PWM signal, as well as receive an analog signal from the main board used in the system to fluctuate the duty cycle of the PWM which is generated. The duty cycle for pulse width modulation determines the amount of time for which the current is allowed to flow during each cycle for every period. This output signal is then sent to the magnetron's input which dimming or increasing power.

Implementing pulse width modulation on the MSP430 is achieved by using the timer incorporated in the microcontroller. The timer works by incrementing each clock cycle by one. The timer modes on the MSP430 are as follows in figure 4.2.2-4

MC_0	Stop timer
MC_1	Count up to the value in register CCR0
MC_2	Counts to 0xFFFF and resets
MC_3	Counts up to CCR0 then Down

Figure 4.2.2-4: MSP430 counting options

The best implementation for the Memrowave is to have the counter store in the CCR0 register (method MC_1). When the device is activated and required to

obtain cook times and power levels for products information pertaining power levels. This will determine the duty cycle needed for pulse-width modulation. For example a product that requires only 50 percent power would call for a 50 percent duty cycle. Using the CCR0 method allows for dynamic changes to the timer giving better control for the magnetron's power. The MSP430 controller will be mounted onto a fabricated printed circuit board. This board will be purchased from a supplier at a later date. Connection to the board is accomplished by surface mounting the MSP430. This controller will then be connected directly to the magnetron via the PCB lines. If using an MSP430 the code will be produced by the Memrowave team. Examples of PWM coding can be found from many locations, this example will serve as guide lines to producing our working code. The code for the MSP430 will be constructed using the Texas instruments software Code Composer Studio Version 5.3. The reason for using this software is ease of use. This software comes pre-installed with all the necessary libraries for the MSP series, allowing for easy compilation of software. Another benefit of using Code Composer Studio Version 5.3 is the whole Memrowave team has experience using this software. Since the full team has this experience we can produce a working prototype much more quickly. Once written the code will be then compiled and stored to the MSP430 microcontroller via USB connection from the computer. This allows for easy installation and testing. Once verified the Chip is then removed from the original board. This microcontroller will be surface mounted onto a PCB once testing has certified a working PWM using a bread board to connect it to a BeagleBone Black. This testing produce will be disused in much greater detail in Chapter 6 which pertains to hardware and software testing.

MSP430 acquisitioning is conducted as follows. For obtaining the MSP430 Launchpad microcontroller we will be ordering directly from the Texas Instruments website or using a pre-purchase model if available. The webpage offers an option to buy directly from the distributor or from a contract distributor. However there is also a possibility to obtain a free sample from the distributor. The Memrowave will likely use a pre-purchased model from previous usage. In addition to this we may also acquire a second MSP430 in case of damage to the chip in installation. Also ordering a second MSP430 may be beneficial in testing of secondary systems and or equipment. The second microcontroller accounts for the addition of Murphy's Law. No matter the case we will be acquiring directly from Texas Instruments and will be supported by a specific member in the group. However, if the Memrowave group decides to use the LM555 timer and circuit to construct a PWM purchasing will be either directly from the Texas instruments website or from an Amazon outlet.

Placement of the controller system is curial to the design of the Memrowave. Our PCB must be form fitting enough for proper placement in the microwave. This PCB must also house the controller for the magnetron, the power supply connections, camera connections, and connections to the BeagleBone black. The location the first comes to mind is behind the standard push buttons on the

microwave. The area provides little interaction from pre-existing hardware since we will remove the unneeded materials. This area also houses a large free space to fit components as seen in multiple models such as Kenmore, GE, and etc. Once gutted the controller will be mounted in the desired location and sealed upon completion. We can create a flow chart of the controller's layout as follows in Figure 4.2.2-5.

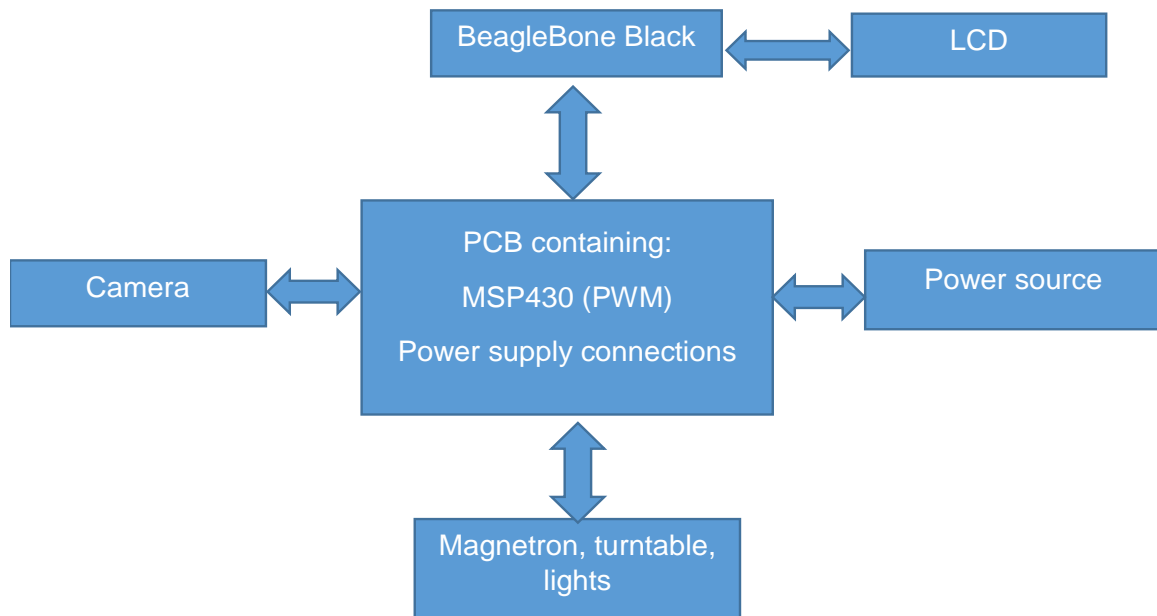


Figure 4.2.2-5: Controller placement

As Figure 4.2.2-5 shows the goal is to sandwich the boards together behind the shell of the microwave. By doing this method we are able to add a form factor that greatly improves the free space in our design. However using this method of placement can be detrimental to our design because it may cause interference from nearby RF components. This noise maybe caused from the magnetron's RF power output. This RF power may interfere with Wi-Fi, timing, and other components. In order to prevent this we may need to shield our parts from interference. This topic will be discussed in a later section on the overall microwave hardware controller (section 4.3.3). Lastly our magnetron controller will ideally be interfaced with other controllers that work with the other subsystems. All controllers will be organized onto the PCB on completion. As stated before form factor is a must in our design since we are limited by space constraints.

4.2.3 Miscellaneous Microwave Controller

Along with powering the magnetron we must also be able to interact and control two other important features of the microwave. These are controlling the turntable and powering the interior light during cooking and opening of the main door. An option to simplify our design can be to scratch the turntable and have the food cook stationary. However using the turntable provides a more thoroughly cooked product. The turntable's purpose in the microwave is to help balance and even the cooking of a product. Microwaves cook by launching standing waves into the product and exciting atoms thus heating food. Figure 4.2.3-1 depicts such a standing wave.

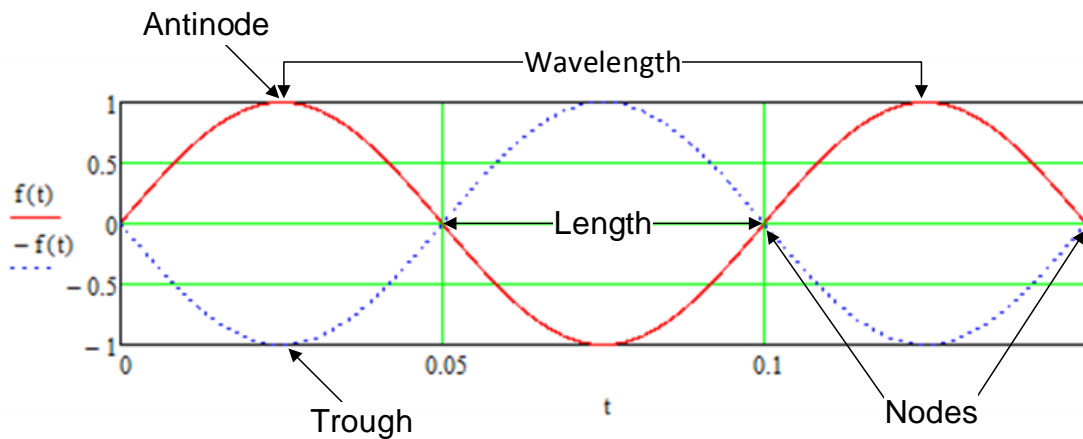


Figure 4.2.3-1: Standing wave depiction

For parts of the product that are located in the nodes of the wave the amplitude of energy is at a minimum. Therefore in these locations the product heats slower. Using the turntable provides a solution to this problem because we will be able to rotate and allow different areas to interact with the wave nodes. The goal of the Memrowave is to provide the user with the best possible experience therefore we will be keeping the turntable as part of the final product. This will require us to control power to the turntable. Lighting is also another must have feature of the Memrowave. We will be keeping the lighting system that came preinstalled with the original microwave. Since we are gutting the internals of the microwave's systems we will need a way to control power flow to the lights inside the microwave, which calls for another system to be integrated.

When working with the turntable it is important to figure out the method in which we will achieve controlling of it. The first option we are deciding to use is to remove the stock microwave motor and use a store bought or build our own motor. For this option we will need to discuss the possible types of motors we will be using. The first type of motor that we can use in the Memrowave is a DC stepper motor. The stepper motor works similar to most motors by converting electrical energy into mechanical energy. These stepper motors differ by rotating

in steps and not in a continuous movement. Stepper motors will also be powered by a series of digital pulses instead of a constant AC or DC signal. This motor contains a set of permanent magnets and the stator has the coils. The coils in combination with the permanent magnets will cause rotation. This rotation can be manipulated to create higher or lower degrees of movement. For example the higher amount of pulses the higher the resolution of stop positions. What this implies is that higher resolutions correlate to a smaller differential change in theta. The advantages of using a stepper motor are as follows. The motor is considered a digital motor so it will be very consistent when turning on and off. Have a highly responsive motor is key to our design. This also provides us with a long term solution that won't need replacing. This motor does not need to receive information of motor position. Not having a feedback loop will increase the overall reliability and effectiveness. The stepper also doesn't contain brushes which will reduce product maintenance. A stepper motor is highly recommend in the Memrowave due to its ease of use, digital nature, and yields precise control of the turntable. The next possible motor is a brushed motor. Similar to a stepper motor the brushed motor contains permanent magnets. These magnets likes like the stepper motor are in fixed locations. The rotor is moved using the magnets. The brushes in the motor are used to change the polarity upon contacting with the electrodes on the spinning arm. Reasons for using this motor are as follows. Ease of implantation, requires a DC power supply, and simply parts. However this motor does have disadvantages. First the motor consists of a mechanical brush that is in constant contact with a surface during rotation. Having this contact will cause deterioration in the product. We want a product that is long lasting. Next the motor has the potential to arc and produces a large amount of electrical noise from the brush. We would like to avoid this possible case from occurring inside the Memrowave. The final motor we will consider is the brushless DC motor. Brushless motors consist of permanent magnets, a spinning arm (rotor), and are powered by a DC input. Similar to stepper motors and brush motors in these fact yet lack the brushes of brush motors and spin continuously unlike stepper motors. The brushless motor has a large advantage by not using brushes. We have a lesser chance to spark and consist of less electrical noise compared to the brush motor. Based off of magnet placement the rotor is much easier to control. However the initial cost of brushless motors is more substantial compared to the brush motor. Price is a large factor in our designs but, since there are less mechanical parts to wear down in the brushless motor overall maintenance prices are dropped. The Brushless motor is a much better candidate than the brush motor; however, our product will most likely use a stepper motor if we decide to proceed with this route.

Since the microwave we purchase will already contain a motor for the turntable we have considered using this motor and connecting it to a controller of our choosing. This option is beneficial because it will save on the cost of our design and can save time on building. If we decide to remove the original motor we lose production time on removal and installation of a new motor system. Being able to integrate the original motor with the power supply we implement and the

controller we use simplifies our design. The negatives of this route may be; incompatibility with our subsystems, the motor may draw a wasteful amount of power, and the original motor may not fit when we add our PCB, BeagleBone Black, and Wi-Fi system. Another reason we would like to use the original motor is our turntable does not have to spin at varying speeds. The table will be set to only one slower speed allowing for a simple motor. Using a motor that only relies on one speed is also beneficial because a circuit to control speed does not have to be implemented. For example if our motor needed to be capable of changing speeds and directions, a possible solution is an H-bridge circuit in combination with a PWM. Without this constraint our system saves on parts and design hours. Since we do not need to add circuitry to change speeds our task is simplified. Therefore using the original motor is a more likely candidate. However our first option for the prototype will be using the original motor in combination with an MSP430 of our choosing since it is already in place in the microwave. If successful the Memrowave will only consider using a new motor if time permits towards the end of the semester.

Other piece of hardware that must be powered is the microwave's pilot light. The light needs to be supplied with an input signal from our power supply we decide to use. It is also imperative that the light is allowed to switch to an on and off state. In Table 4.2.3-1 we provide a true table to show states.

Condition	Status
Microwave door open	Light is on
Microwave door closed and not cooking	Light is off
Microwave door is closed and currently cooking	Light is on

Table 4.2.3-1: Illumination table

To save room however it is an option to disconnect the light from the power source to save on pin space and power consumption. However, the light during the cooking process does provide feedback to the user. This feedback can help the user verify that microwave is working and allows the user to see the process occurring. The light also consumes a negligible amount of power therefore removing the light will provide no large benefit. Table 4.2.3-2 provides possible lights used in microwaves. As the table shows, most microwaves provide similar power consumptions of around 20 Watts. We are also in no danger of running out of pin space so this reason will allow us to keep the light as part of the Memrowave.

Maker	Product number	Power consumption
General Electric	AP4380308	20 Watts
General Electric	WB25X10019	20 Watts
Globe	Z187	20 Watts
General Electric	JVM1860SF001	20 Watts
Whirlpool	8206232A	40 Watts

Table 4.2.3-2: Power table

Based on the presented reasons the Memrowave will be keeping the microwave light. The light controller we use will be an option between the MSP430G2553, MSP430F5529, or the Texas Instruments TM4C123G. The next hardware we need to control is the microwaves fans. The microwave will only convert and use part of its microwave energy in the heat of products. For the majority of user microwaves around 1100 W of electricity is consumed. This produces 700 W of microwave power, yielding an efficiency of sixty four percent. The left over 400 W are dissipated as heat, primarily going to the magnetron, along with subsequent power being used to control lamps, the AC transformer, food turntable motor, the magnetron's fan and the control circuits. This wasted heat coupled with heat from the product being microwaved, while be exhausted as heated air through cooling vents. This verifies that we need to keep the magnetron fan to prevent overheating of our design. Another fan we may when to power is the exhaust fan. The exhaust fan is important to keep in our design because it will vent our unwanted fumes caused by cooking. The exhaust fan also serves as a space saver. Using the exhaust fan allows you to store the microwave over an oven. If the fan isn't used a gas range may exhaust fumes into the microwave when cooking. Now that we have discussed what hardware we are using we can now proceed to acquiring a specific microcontroller. Controlling the Memrowave will require 7 input/output pins to function. These are listed as; turntable motor, lights, fan, magnetron, and a seven segment display. The seven segment display uses 3 pins to operate. Because we only need seven pins we are able to use the MSP430G2553. This chip will be able to control power to the magnetron, turntable motor, lights, and fans. The MSP430G2553 also has a smaller amount of pins then the MSP430F5529. It will be better to use closer to the exact number of pins. This saves space on any fabricated PCBs. The TM4C123G would not be needed in our design because it is too complex for what we need. Our design is already using the BeagleBone Black to run higher level functions so the TM4C123G is considered overkill. Therefore our main microcontroller of choice is the MSP430G2553. A low level diagram of the layout can be shown in Figure 4.2.3-2

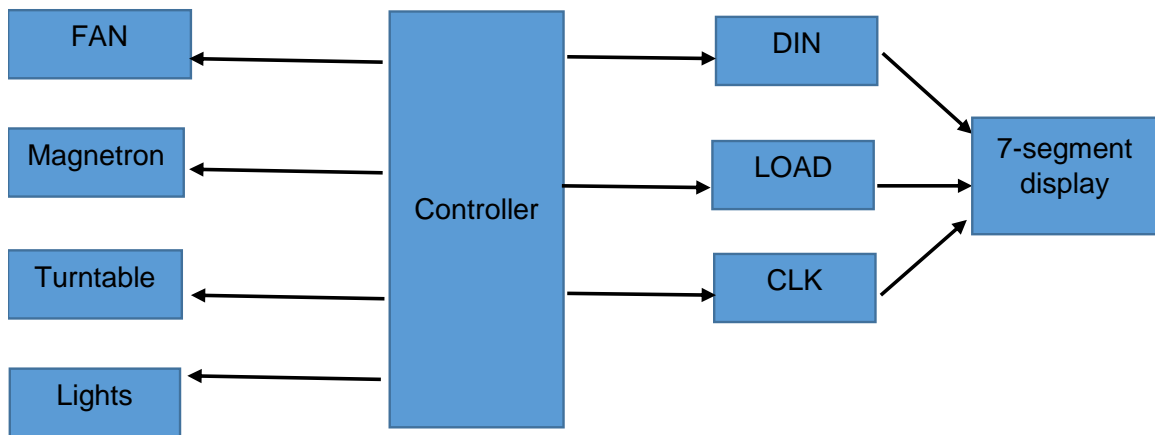


Figure 4.2.3-2: Pin Diagram

In order to have these parts function with the microcontroller at the correct times we need to create switching circuits. These switching circuits will function in unison during the operation of the microwave and during ideal use. The switching circuits will be created using MOSFETs. We will also be making four of these circuits; one for the fan motor, the turntable motor, the light, and the magnetron. Each will be placed on a separate pin of the microcontroller. These unfortunately take up four locations. Prototyping of circuitry will be conducted on a breadboard found in one of the various laboratories at our disposal. Once prototyping is completed we will then fabricate a printed circuit board which will house the circuits. The circuit will be implemented by our design team using purchased passive components, transistors, wires, and solder. For better time management our design team may purchase a solder station. This allows for consist working and reworking of our printed circuit board. In Figure 4.2.3-3 we depict an example of a basic switching circuit using MOSFETs.

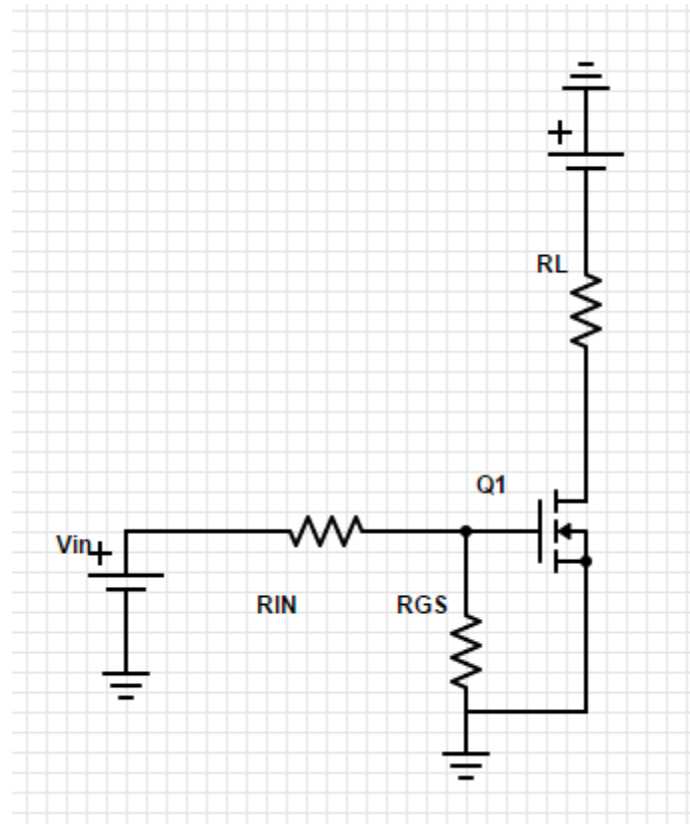


Figure 4.2.3-3: Switching circuit.

4.2.4 ARM Microprocessor

In this section, we will discuss the selection of the ARM microprocessor for the Memrowave.

4.2.4.1 ARM Microprocessor Specifications

While most ARM microprocessors will have adequate computing power for the Memrowave, there are two general and fundamental requirements that will be used to determine which ARM microprocessor will be used. The first requirement is that it must be capable of running Android out of the box, preferably Android 4.x or higher. Android was selected to aid software development such as the user interface and food preparation database. The second requirement is that the microprocessor must be available in an experimenter board that can in turn easily connect to add-on expansion boards similar to Arduino shield add-on boards. This requirement is needed since this will be the method to connect the touch screen display. The board designed and built for the Memrowave will also be connected like any other expansion board. Using these two requirements immediately eliminates some ARM microprocessors such as the Broadcom

BCM2835 used in the popular Raspberry Pi board which doesn't have a supported, stable version of Android available. The Texas Instruments OMAP4460 ARM Cortex-A9 MPCore used on the Pandaboard ES looked very promising at first due to its Android support, graphics capabilities, and built-in camera interface which would be useful for this project. However, the Pandaboard is more of a standalone development platform that doesn't use standard add-on boards. The Pandaboard is also expensive compared to most experimenter boards. The Texas Instruments Sitara AM335x ARM Cortex-A8 Microprocessor used on the Beaglebone Black seems to be the right fit.

4.2.4.2 TI Sitara AM335x Microprocessor

The only ARM microprocessor that meets the requirements of Android support and expansion board capability is the Texas Instruments Sitara AM335x ARM Cortex-A8 Microprocessor used in the Beaglebone Black. Table 4.2.4.2-1 shows specifications for the Sitara AM3358 microprocessor. Table 4.2.4.2-2 contains the specifications and features of the Beaglebone Black and Figure 4.2.4.2-1 is a picture of the Beaglebone Black. Note the 46-pin headers on opposite edges of the board. These headers will be used to connect the Memrowave PCB to the Beaglebone. A matching pair of these headers on the Memrowave board will then connect the LCD touch screen display. It is important that the LCD touch screen display chosen is a Beaglebone Black expansion board, known as a "cape", or can be modified to fit the Beaglebone Black's headers. The Memrowave PCB will be sandwiched between the LCD and the Beaglebone Black using matching 46-pin headers.

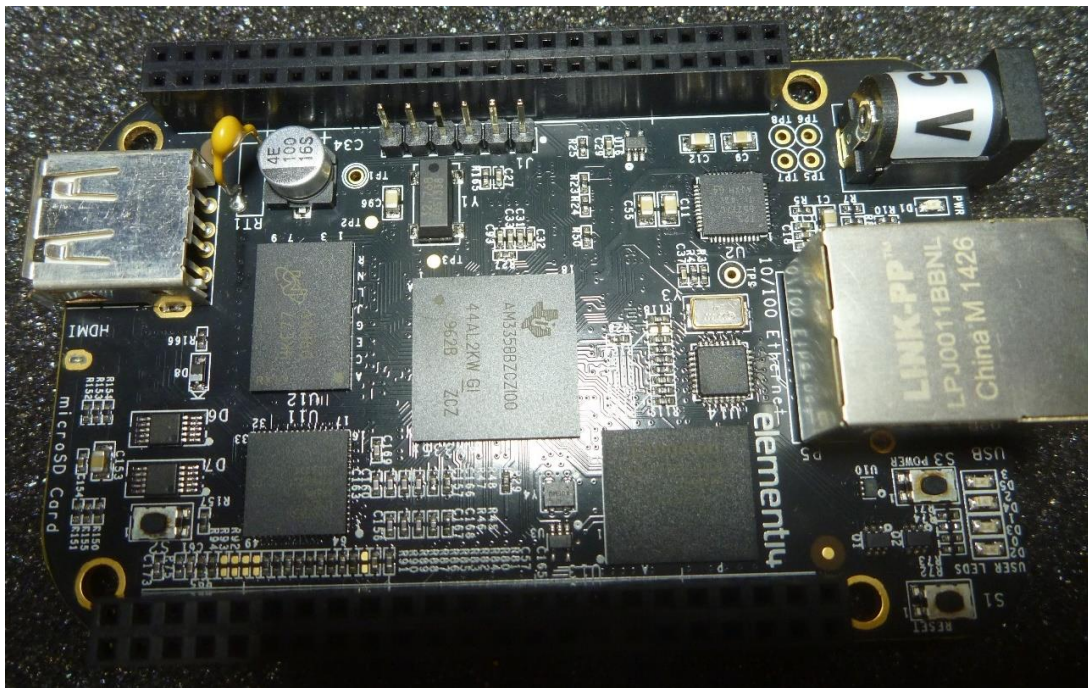


Figure 4.2.4.2-1: Beaglebone Black with Sitara AM3358 ARM Microprocessor

ARM CPU	ARM Cortex-A8
ARM MHz (Max.)	1000
ARM MIPS (Max.)	2000
On-Chip L1 Cache	64 KB (ARM Cortex-A8)
On-Chip L2 Cache	256 KB (ARM Cortex-A8)
Other On-Chip Memory	128 KB
Graphics Acceleration	3D
Display Options	LCD
DRAM	1 16-bit (GPMC NAND flash NOR Flash SRAM)
USB	2
MMC/SD	3
UART (SCI)	6
PWM (Ch)	3
Real Time Clock	1
I2C	3
SPI	2
DMA (Ch)	64-Ch EDMA
Standby Power	7 mW
IO Supply (V)	3.3
Operating Temp Range (C)	-40 to 105 C

Table 4.2.4.2-1: Sitara AM3358 ARM Microprocessor Details

4GB 8-bit eMMC on-board flash storage
512MB DDR3L 800MHZ SDRAM
3D graphics accelerator
NEON floating-point accelerator
2x PRU 32-bit microcontrollers
SD/MMC Connector for microSD
USB host, Ethernet, HDMI
2x 46 pin headers

Table 4.2.4.2-2: Beaglebone Black Details

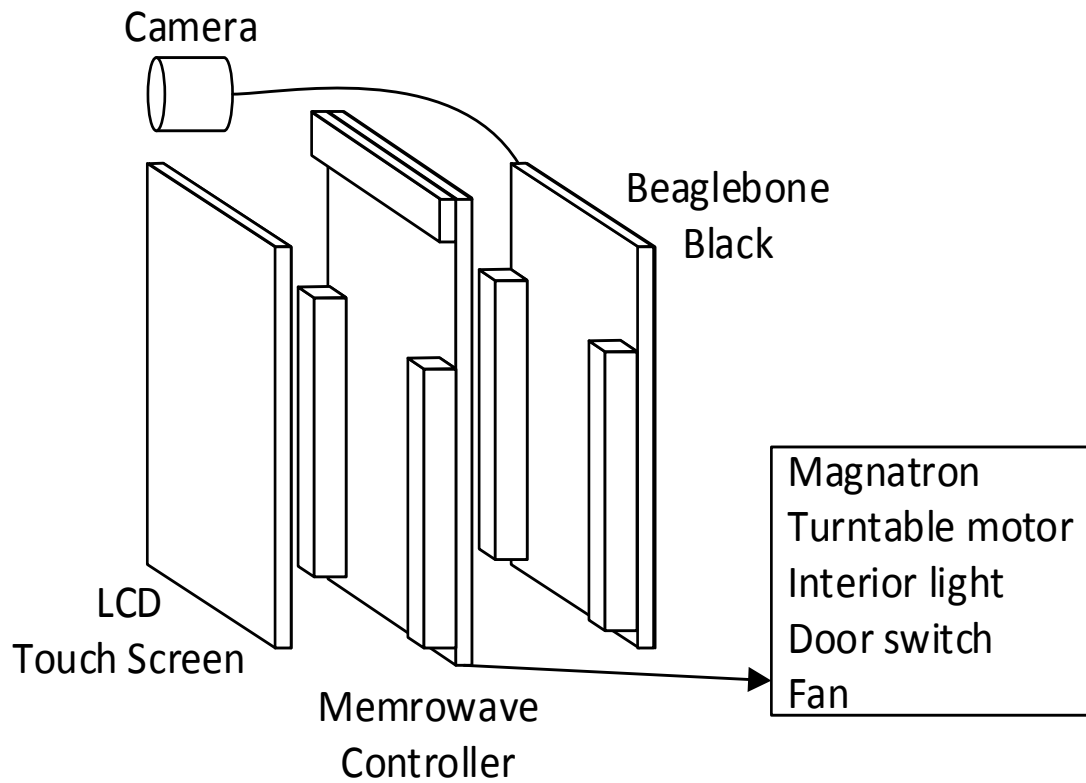


Figure 4.2.4.2-2: Simple Memrowave system diagram

4.2.5 LCD with Capacitive Touch Screen Display

In this section, we will discuss the various options for the output display and touch screen hardware.

4.2.5.1 Specifications

The Memrowave's display will handle the input and output of the software interface. The display should be a thin film transistor (TFT) liquid crystal display (LCD) utilizing an integrated capacitive touch screen. The physical size of the display is determined by potential microwave chassis space. Of the standard microwaves available and evaluated, the Panasonic NN-H965X ABH has the narrowest available panel space of approximately 2.82 inches in width. This narrow space would only be able to accommodate a 4:3 ratio, 4.3-inch diagonal display with the display in the portrait orientation. A 5-inch display could be used if it had a widescreen ratio such as 16:10. The preferred display would use a widescreen ratio since this would allow about a 0.8 to 1.2-inch increase in the height dimension for the same width with the display in portrait. The display must be compatible with the Beaglebone Black with Android 4 running as the operating system. The display should get its power directly from the Beaglebone black. Mounting holes are desirable, but not a hard requirement.

4.2.5.2 LCD with Capacitive Touch Screen Display

After extensive research there does not appear to be much demand for LCD capacitive touch screen displays for the Beaglebone Black at this time. Most displays at this small size use resistive touch screens. The only display that comes close to the specifications is a 5-inch display from Chipsee. See specifications in Table 4.2.5.2-1. Note that the normal display width/height dimensions are shown for the rotated portrait orientation that will be used for the Memrowave. Also note that the overall size of the part is larger than the display since the PCB extends beyond the actual display boundaries on all 4 sides.

Display size	5-inch
Display resolution	800 x 480 pixels
Aspect ratio	16:9
Touch type	5-point capacitive
Display width	3.02"
Display height	4.77"
Overall width	3.51"
Overall height	5.28"
Power	Provided by Beaglebone Black
OS support	Android 4.2 Angstrom Debian TI Linux Windows CE7

Table 4.2.5.2-1: Chipsee CS-BBB-EXP50C Display Cape Details

4.2.6 Wi-Fi Interface

Internet connectivity gives the Memrowave's software access to an online products database that can be queried for items that are not found in the local database. This keeps the local database as small as possible, a desirable outcome when working with a limited amount of memory. Only the user interface software running on Android will require Internet connectivity. This connectivity will be provided via a USB Wi-Fi device attached to the Beaglebone Black's USB port. Kitchen appliances tend to remain in use for many years and are not replaced nearly as frequently as smart phones and other smart devices. It is likely that Wi-Fi standards will change during the typical lifespan of a microwave. Using a USB device has the advantage of being easy to replace or upgrade. An example of an acceptable USB adapter with Linux/Android support is the Netgear WNA1000M. Table 4.2.6-1 shows the adapter's specifications.

Controller	Realtek RTL8188CUS
Standards	IEEE 802.11b/g/n
Frequency	2.4GHz
USB	1.1, 2.0, 3.0
Linux driver	rtl8192cu

Table 4.2.6-1: Netgear WNA1000M specifications

4.2.7 Camera

For the camera, we will be using the Logitech HD C270 USB webcam, shown in Figure 4.2.7-1. It has a 1280 x 720 resolution, which should be more than enough to read barcodes and QR-codes reliably.



Figure 4.2.7-1: Logitech C270 Webcam

4.3 Software

The Memrowave cannot be 'smart' without well developed software, so in this section we will be discussing the software design details and considerations for the Memrowave.

4.3.1 The Android Operating System

In section 3.2.6, we compared the various operating systems that we could use for the Memrowave and decided that Android was the best choice. Android support for the BeagleBone Black is provided by the open-source Rowboat project, which enables Android Jelly Bean v4.2.2 to run on TI Sitara ARM Cortex-A processors.

4.3.2 Drivers, Libraries, Dependencies

There are a number of drivers that are needed for Android to support the hardware of the Memrowave. A driver is needed for the LCD to work properly, but luckily, this driver is included in the Android distribution we will be using. The driver for the capacitive touch screen is included as well. However, the driver for the camera is not included, so it will need to be obtained separately and installed on the system, as will the driver for the Wi-Fi interface.

Android includes most of the libraries we will need to implement the software of the Memrowave, with a few exceptions. Android does not have a native library for reading barcodes or QR-codes using a camera. To handle this, we will be using the open source ZXing multi-format barcode image processing library. This software library is licensed under the Apache License 2.0, which gives us permission to use the software in our project. A link to the full text of the Apache License 2.0 is shown in Appendix A.

4.3.3 Microwave Control

The microwave hardware will be controlled by a circuit separate from the BeagleBone Black, with its own microcontroller. The microwave control circuit will take care of the timing, power level, magnetron control, fan control, turntable control, beeper, and the light control. The BeagleBone Black will communicate with the microwave control circuit via the I2C communication protocol, with the BeagleBone Black configured as the master and the microwave control circuit configured as the slave. In Table 4.3.3-1, we outline all of the features of the microwave control circuit and assign the I2C commands that will be used to invoke these features.

Command	Description	I2C Command
set_timer	Clears the timer and sets it to the specified time	0x00
get_timer	Requests the current state of the cook timer	0x01
clear	Clears the current time from the timer display	0x02
cook_start	If the timer is set, this command will start the microwave, which will then proceed to cook the food at the current power level setting.	0x03
cook_stop	Pauses the timer and stops the microwave from cooking	0x04
timer_start	Starts the timer without cooking	0x05
set_power_level	Sets the power level to the specified value	0x06
add_time	Adds the specified amount of time to the current timer state	0x07
set_clock	Sets the real-time clock to the specified value	0x08

Table 4.3.3-1: Microwave control circuit commands

For commands that are expected to return a value, the I2C master must write the I2C command to the slave device, and then read from the slave device to retrieve the result. Simultaneously, on the slave device, after receiving a command that expects some return value, the slave must write to the master. An activity diagram of the I2C command sequence on the microwave control board is shown in Figure 4.3.3-1.

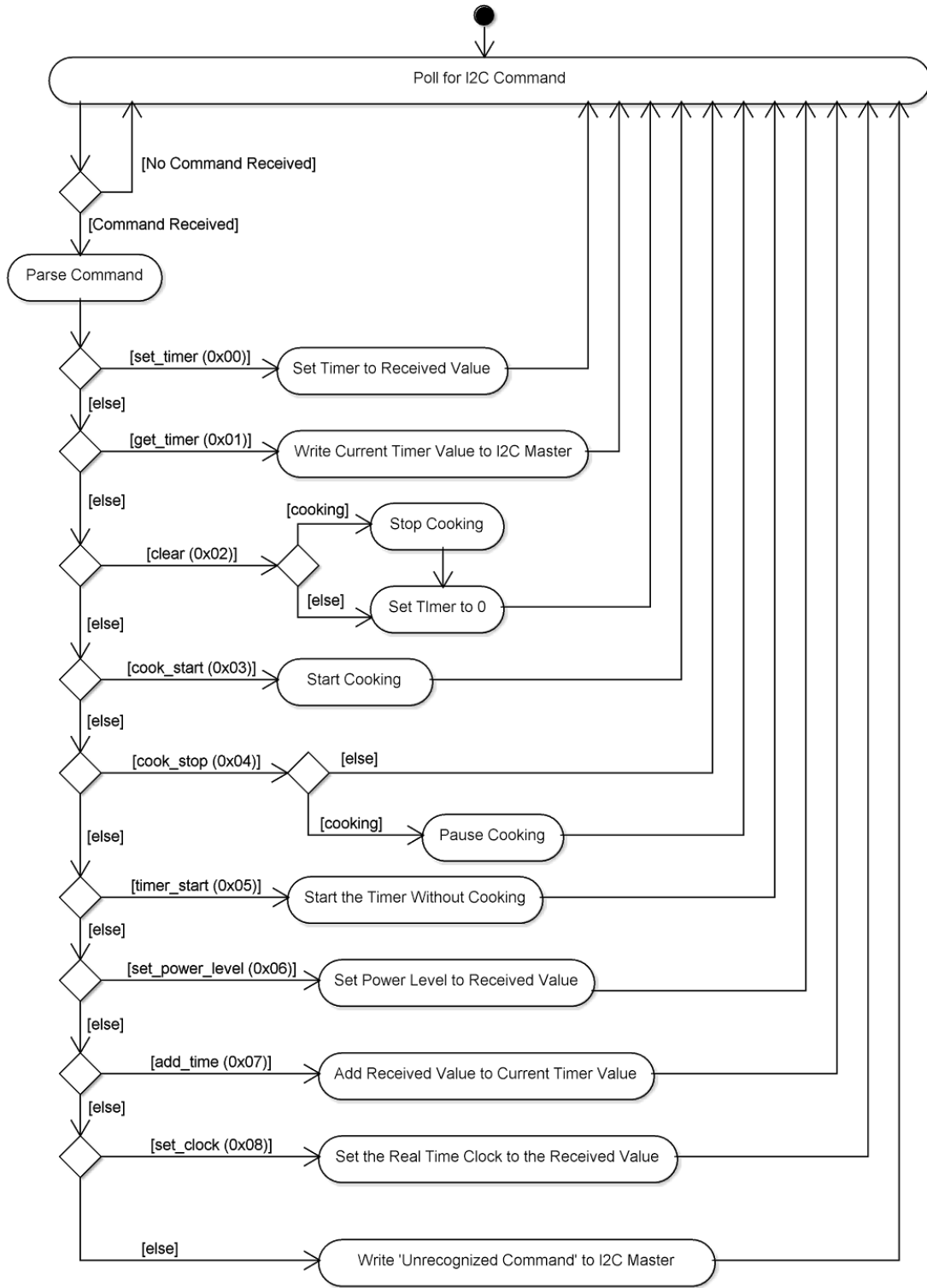


Figure 4.3.3-1: Microwave control board I2C command poll sequence

On the Beaglebone Black, we will have a single class that will allow the rest of the software to interface with the microwave control board. This class will implement all of the I2C communications between the Beaglebone Black and the microwave control board and provide a simple interface for the rest of the software to control the microwave hardware. A UML diagram of this class is shown in Figure 4.3.3-2. The UML diagram also describes an enumeration that contains the possible states of the microwave.

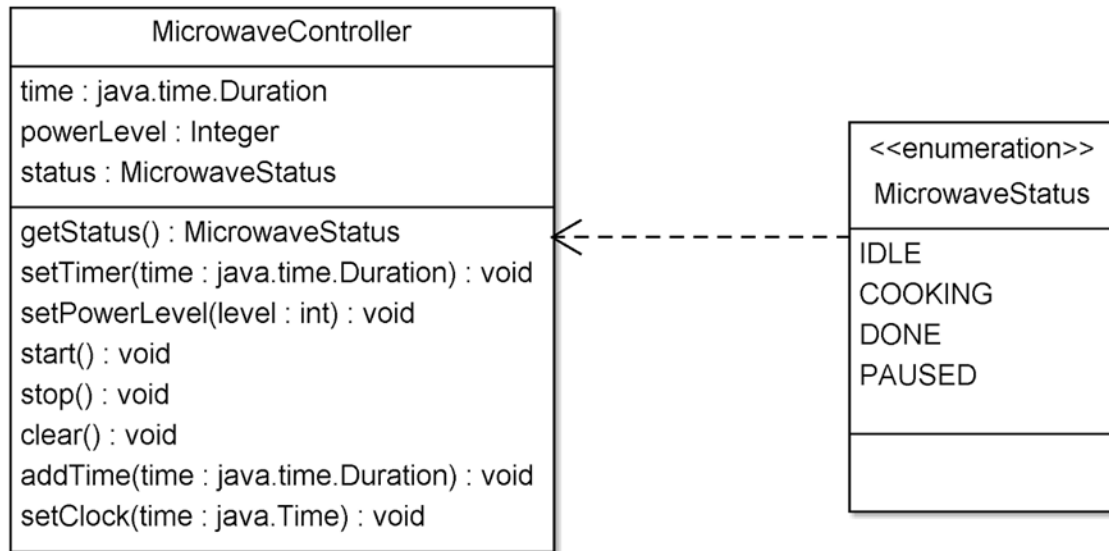


Figure 4.3.3-2: MicrowaveController UML Diagram

There is only one microwave control circuit, so we have to ensure that there can only be one instance of the MicrowaveController class. This could be accomplished by making all of the methods static, but this would allow any class in any thread to gain access to the microwave hardware. Another option would be to implement the class in a singleton pattern. Instead of having a public constructor like a normal class, this class will store a single instance of itself as a private static variable. When another class would like to gain access to this instance, it will call a method that returns a reference to this instance, or the method will create the instance if it has not yet been created. Implementing the class in this way will allow us to control what parts of the software can gain access to the microwave hardware.

4.3.4 Barcode and QR Code Software

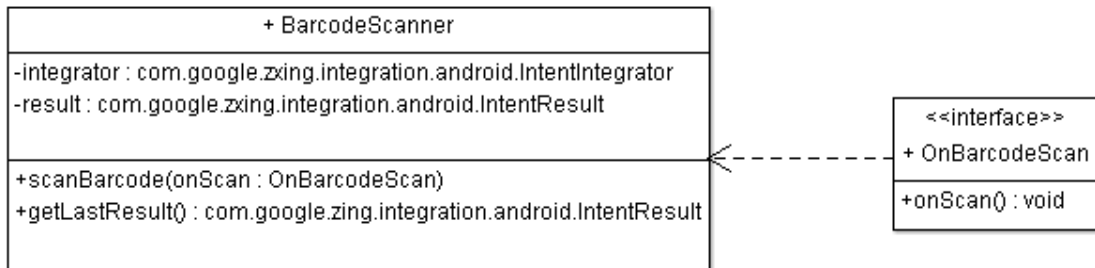


Figure 4.3.4-1: BarcodeScanner class diagram

The barcode scanning will be done using the `BarcodeScanner` class, which will use the ZXing barcode library to scan barcodes and QR-codes using the Memrowave’s camera. A class diagram of the `BarcodeScanner` class is shown in Figure 4.3.4-1, along with an interface for the `OnBarcodeScan` callback.

4.3.5 Local Product Database

Android uses local SQLite databases to store application data. SQLite is a software library that implements a self-contained, server less SQL database engine.

The structure we’ve designed for the Memrowave’s local product database is shown in Figure 4.3.5-1. The database called *Products* has a table called *Product*. This table will have a row for each product. This row will have a unique id, the product’s upc, the product’s name, etc. The *has_steps* column defines whether or not the product has any steps. If this is false, the specified *power_level* and *cook_time* will be used. If it is set to true, each step will be stored as a row in a unique table in the database. The *steps_id* column in the *Product* table will store the id of the table that contains the steps. Each step will have an *id*, *name*, *description*, *cook_time*, and *power_level*.

The figure also shows an example product, oatmeal. For purposes of this example, the oatmeal product will require multiple steps to prepare, so *has_steps* is set to true, and the *cook_time* and *power_level* fields are left blank. The *steps_01* table holds all of the steps needed to prepare the oatmeal. The first row of the *steps_01* table holds the first step, with an example id of 0001. The name is given as “Step 1”, and the description field stores the instructions for the step. The Memrowave will be cooking for this step, as indicated by the provided *cook_time* and *power_level*. If there is another step in the *steps_01* table, it will be shown on the Memrowave after the first step finishes cooking.

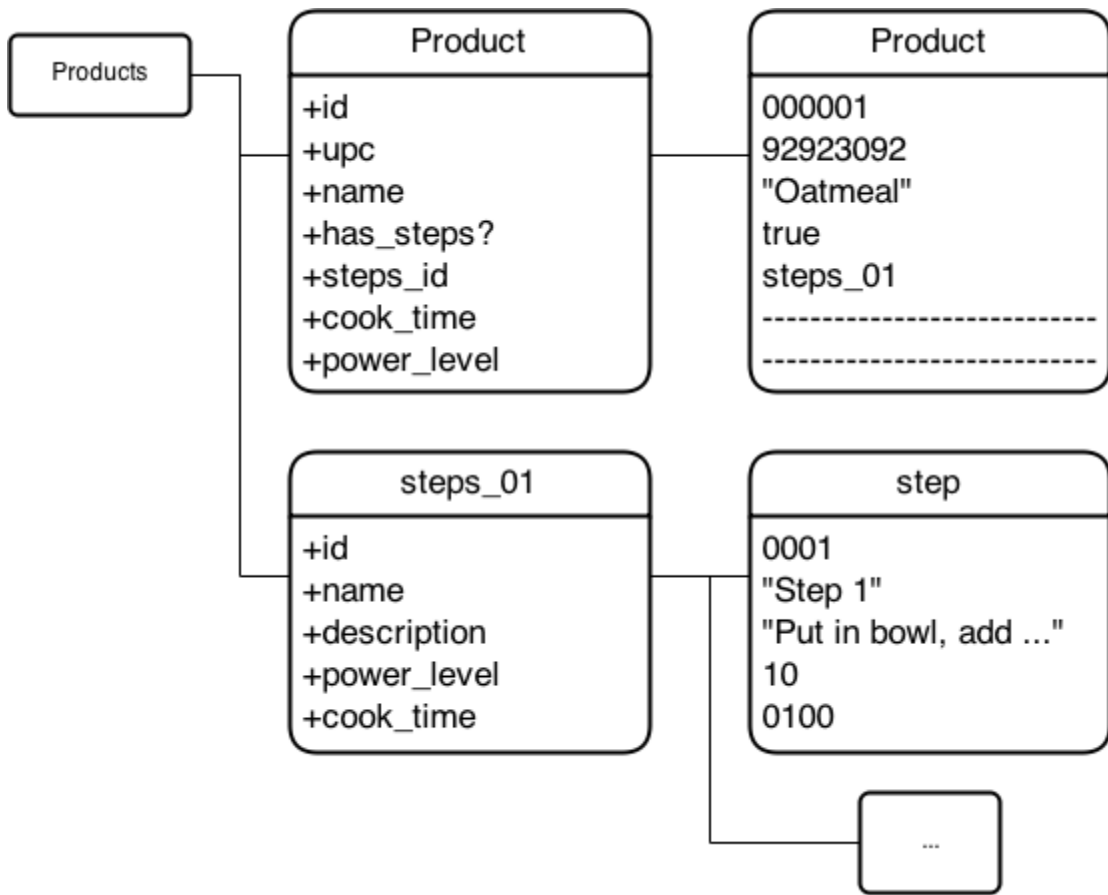


Figure 4.3.5-1: Local product database structure

Now, we need to define how this database will be implemented in our software. We will create a Product class that stores the information about a product and a Step class that holds the information about a single step. If a product has any steps, they will be stored as a list of Step objects. Then we will create a singleton class that will provide access to the product database. This class will provide methods to search the database and add, remove, and update any records. Figure 4.3.5-2 shows how we will implement these classes.

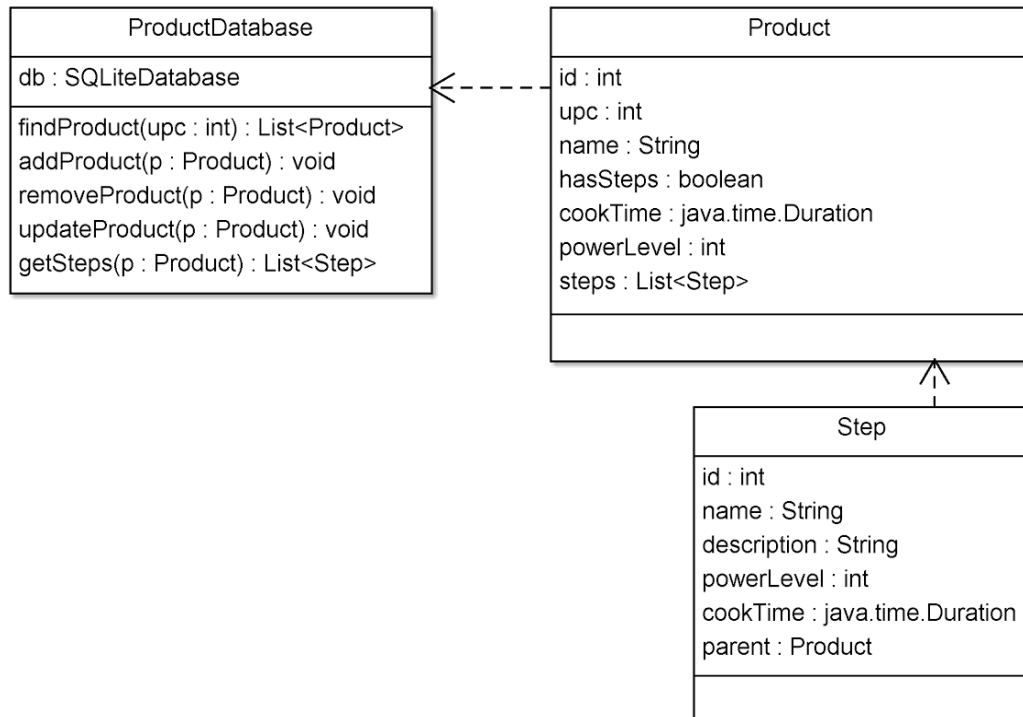


Figure 4.3.5-2: Local product database class diagrams

The ProductDatabase class has several methods to allow the Memrowave software to interface with the SQLite database. The findProduct method takes an integer UPC code as an argument and returns a list of products from the database that matches this UPC code. UPC codes are supposed to be unique, so it would make sense for this method to return only a single Product instance, however, a List is used to ensure any duplicate results are shown as well. The addProduct method takes an instance of the Product class as an argument and returns nothing. This class will verify the information in the Product class is formatted properly and then it will add the product to the database. The removeProduct class will remove the specified product from the database. The updateProduct will replace an entry in the database with the specified Product instance, provided that the ids match. Finally, the getSteps will retrieve the list of steps for a particular product from the database and return it.

4.3.6 User Profiles

The user profiles for the Memrowave will be used mainly as a way of changing the SMS notification recipient quickly between several different people, but they also serve as a convenient way to store different settings for each user. In section 4.3.7, we will discuss how the User Profiles will be accessed by the User Interface, from a procedural viewpoint. In this section, we will discuss how the

User Profiles will be stored in the database and how they will be implemented at the class level.

To design the database Schema, we must list all of the unique data that will be stored in a profile, and then use this list to design the Schema.

User Profile Data:

- Id
- Username
- Cellphone number
- If notifications should be sent

Using this list, we designed the database Schema shown in Figure 4.3.6-1.

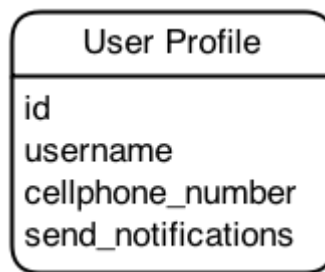


Figure 4.3.6-1: User profile database schema

In addition to the database schema, we must design the class structure that will be used to implement the User Profiles in software.

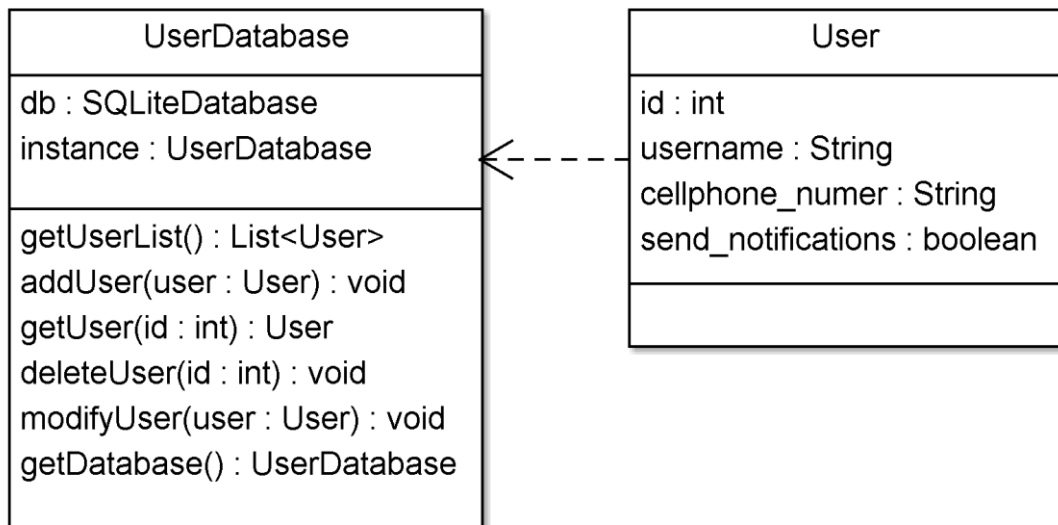


Figure 4.3.6-2: Class diagram of User Profiles

4.3.7 User Interface

The primary means of using the Memrowave is a touch screen interface and not physical buttons like older microwaves. Touch screen interfaces should be reasonably familiar with the majority of users today due to the proliferation of handheld smart devices such as cellular phone and tablets. With this in mind, conventions that are used by modern communication devices, such as the swiping motion to advance through a list, will be used for the Memrowave interface where feasible. Interface design should be kept simple while also trying to avoid driving too deeply into menus or requiring too many steps to accomplish a simple task. Breaking conventions used for current microwaves should be avoided unless there is a clear advantage to make a change. For example, there is no need to change the orientation of the number keypad from what is currently used on microwaves to the keypad used for calculators.

The first screen that a user will see is the Memrowave's main menu or home screen. This screen should remain simple with a clear focus on what makes the Memrowave unique, that is the ability to scan and automatically prepare food products. Figure 4.3.7-1 is a simple view of the main menu as seen by the user.

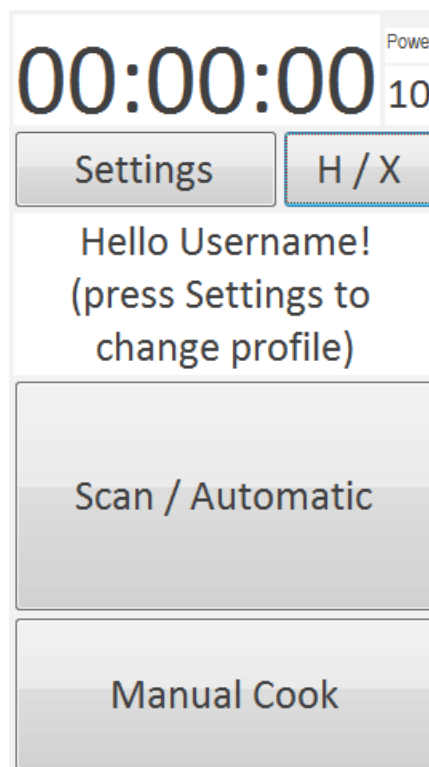


Figure 4.3.7-1: Main menu/home screen

The Scan / Automatic button is prominent and easily accessible, but the Manual Cook button, which may see as much use as the Scan button, does not look like

an afterthought. The H / X button will appear on all interface screens. The H stands for home and the X stands for cancel. The H / X text will be replaced by a custom graphic showing a house for home and a red X for cancel. This button has different functionality depending on where the user is, but generally pressing it once will cancel the current action and pressing it a second time will take the user all the way back to the main menu. A 3-line output display is used give information when required and prompt the user. The Settings button will change to support other functions as necessary and will be disabled if no other function is required. The time display is shown on the screen to give a general sense of how the screen will appear with the Memrowave's 7-segment time display. Figure 4.3.7-2 shows the main menu software activity

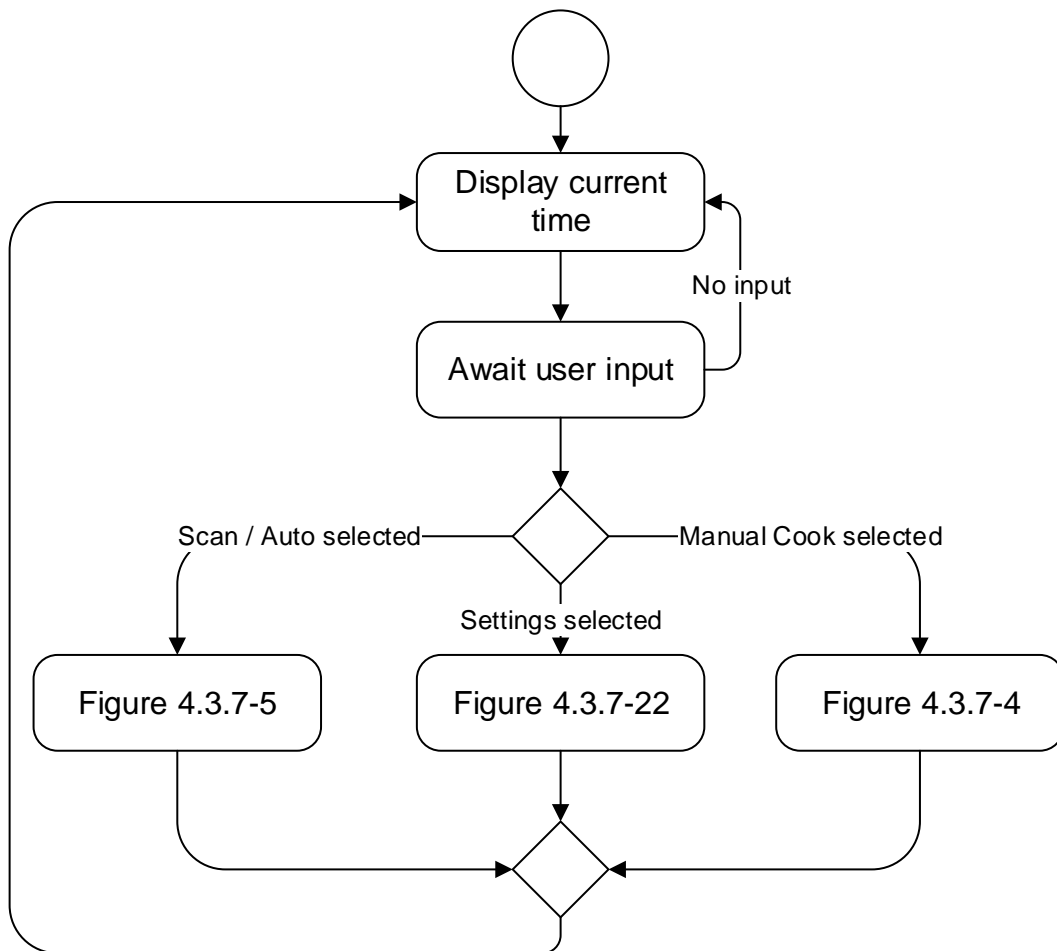


Figure 4.3.7-2: Main menu flowchart

Despite the automatic nature of the Memrowave, it should have a capable manual control option. Figure 4.3.7-3 shows the manual cook screen.

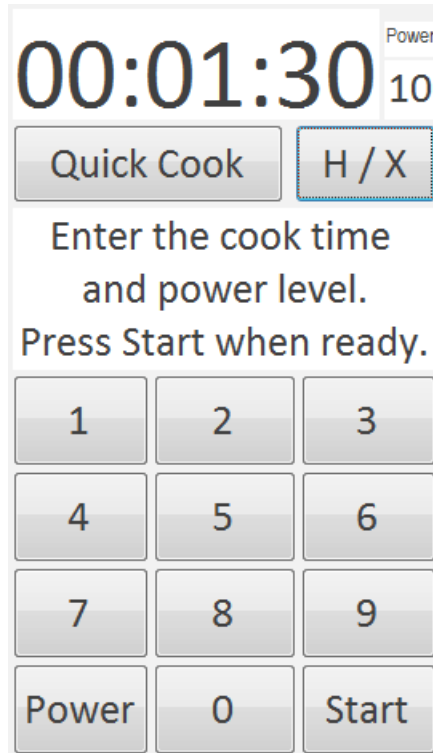


Figure 4.3.7-3: Manual cook screen

This screen and the software control shown in Figure 4.3.7-4 are patterned generally after the simpler operation of the Panasonic model NN-H965X ABH microwave. The manual control is straight forward without a lot of confusing options. For example, there is only one way to change the power level; press the Power button until the desired level is displayed.

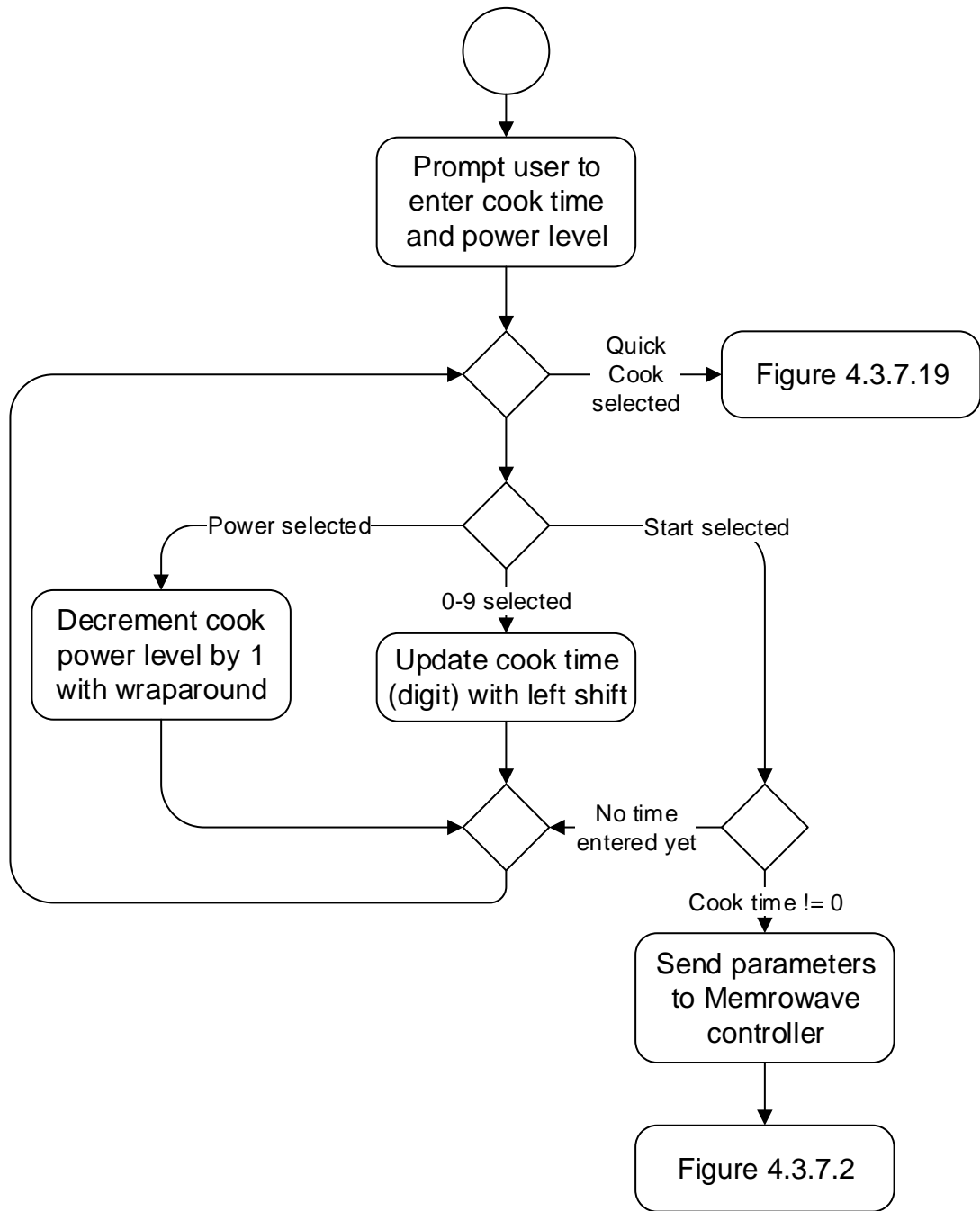


Figure 4.3.7-4: Manual control flowchart

The strength of the Memrowave is the scanning and automatic preparation of food products. This process needs to be kept as simple and clean as possible in order to provide the best user experience. The first thing to note in the software flow shown in Figure 4.3.7-5 is that an automatic scanning feature that does not require a user interface action is included. It is the intent to have this feature start scanning and processing, looking for barcodes shortly after the user presses the Scan button on the main menu.

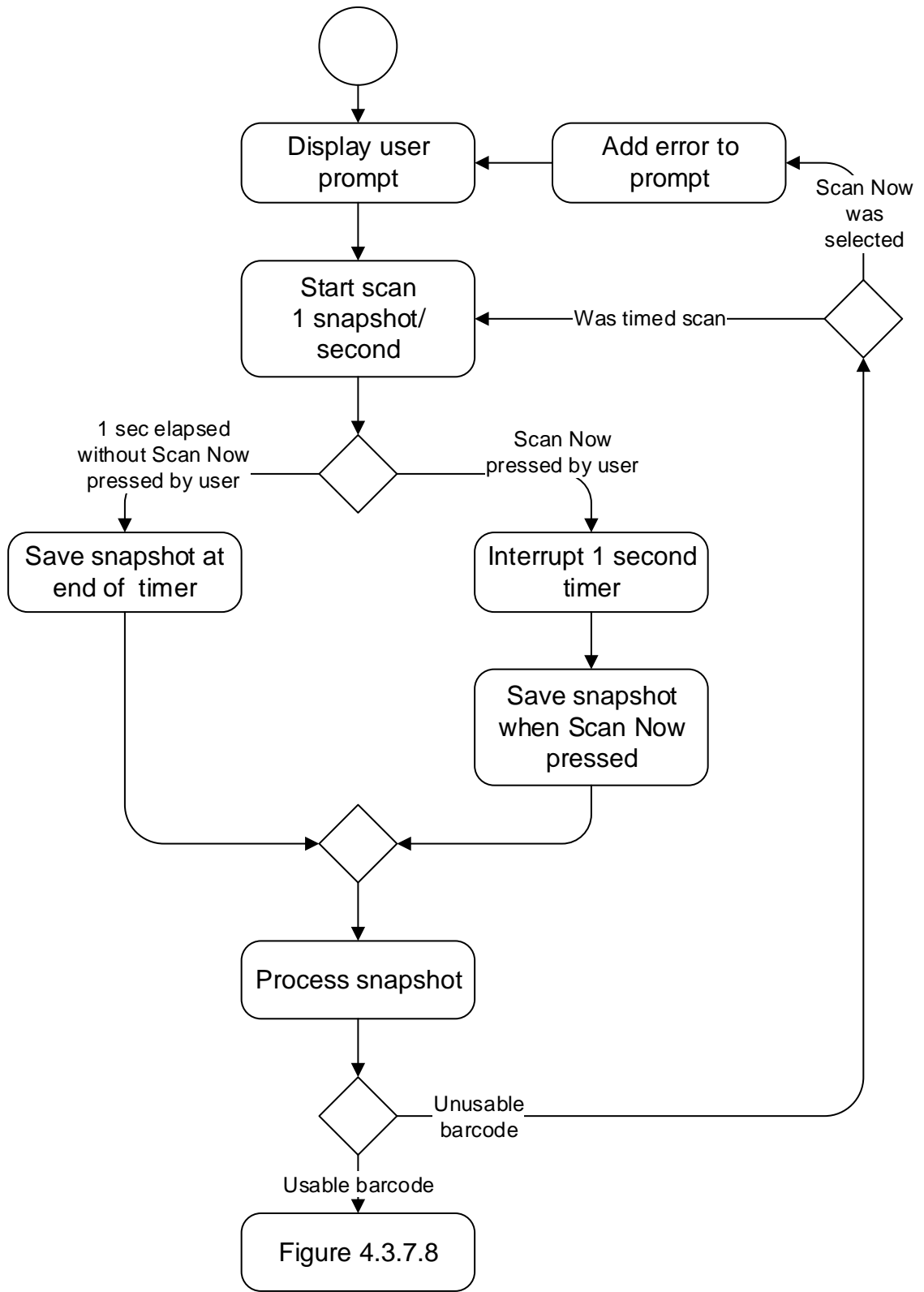


Figure 4.3.7-5: Scanning barcode flowchart

The user simply holds the barcode up to the Memrowave's camera. The Memrowave will display what the camera sees as shown in Figure 4.3.7-6. This display will be as near real time as possible given the capabilities of the hardware. A 1 second per snapshot rate is indicated only for the purposes of design and will likely change during testing. If a usable barcode can be identified during this automatic process, the user will not need to press the Scan Now button on the scanning screen. The other possibility is that a user may discover, through extensive use and experience, the best way to present a barcode to the Memrowave's camera. Conditions may certainly vary based on room lighting and other factors. In these cases, the user may be able to present the barcode to the camera and be able to press the Scan Now button in one fluid motion that often beats the automatic process. In this way the possibility of a faster result using the automated process is provided for while also giving the user ultimate control with the Scan Now button.

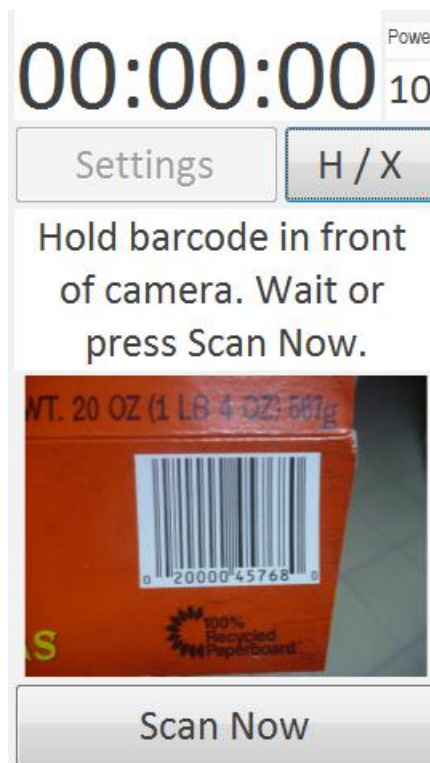


Figure 4.3.7-6: Scanning a barcode

When a usable barcode is processed from a snapshot, a clean barcode is generated displayed to the user. This presentation as shown in Figure 4.3.7-7 is intended as a verification of a successful scan since any problems with the snapshot processing could result in the wrong food product and settings being identified.



Figure 4.3.7-7: Scanning successful

If the wrong barcode is displayed, the user has a Rescan button available to immediately try again without cancelling back to the home screen. As shown in Figure 4.3.7-8, any valid barcode results in a local database lookup to locate the settings record. If this record is not found locally, the internet database is queried for the record. When that record is found, it is saved to the local database. The software displays the name of the product and updates the cook time and power level based on the product record. After reviewing all this information, the user may use the Continue button to proceed to the next step.

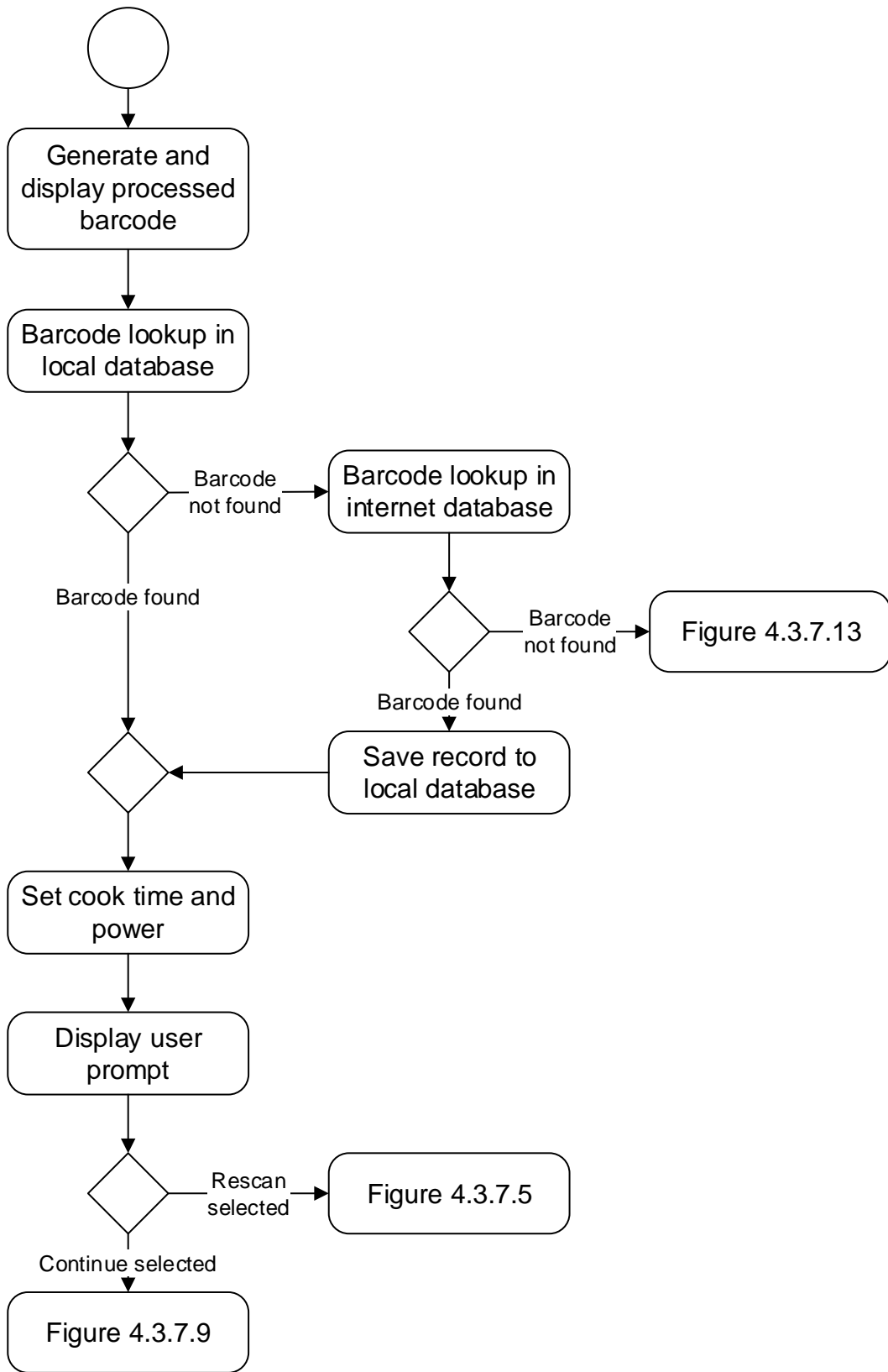


Figure 4.3.7-8: Successful barcode scan processing

The final step in the normal automated process is to give the user some final options as shown in Figure 4.3.7-9. The first is to simply give the option to rescan just in case the user erroneously pressed Continue at the previous screen.

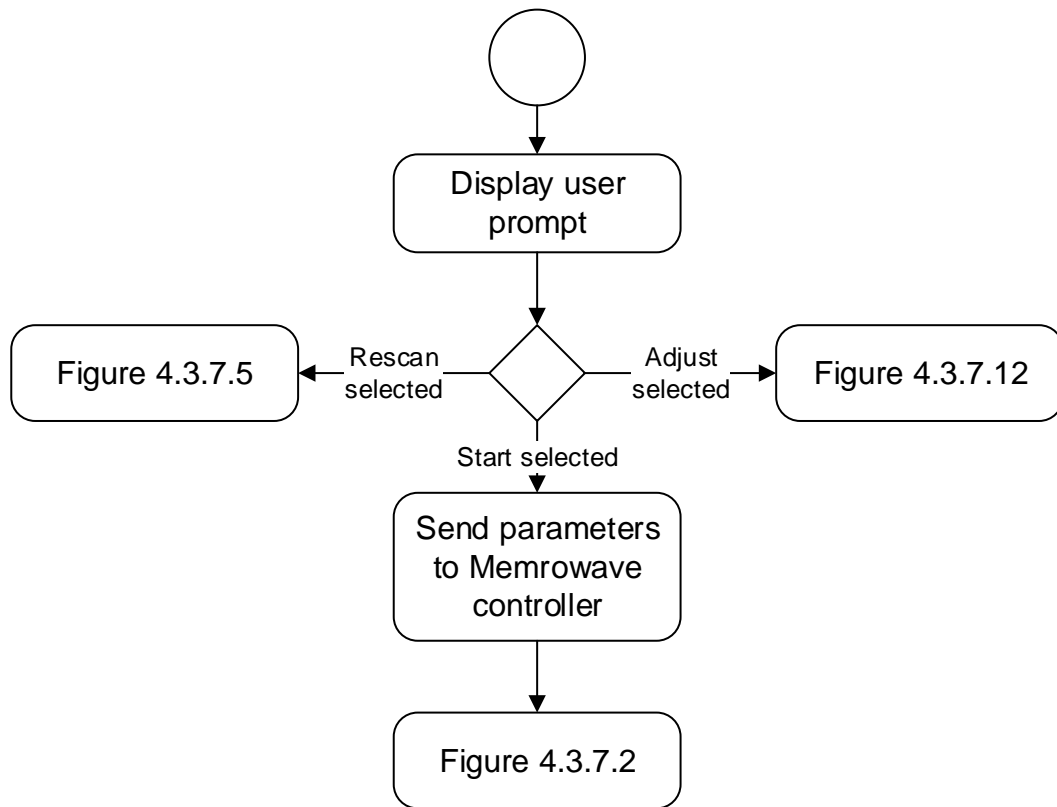


Figure 4.3.7-9: Automatic operation complete

The second option is the ability to make adjustments to the settings before preparing the food. This is accomplished with the Adjust Time/Power button shown in Figure 4.3.7-10. This brings up the screen in Figure 4.3.7-11 which looks similar to the manual control, but with the time and power already set. The user can make changes on this screen and, if it's just a one-time event, hit Start. The settings are not saved to the local database. However, if the user wants the changes to be saved and used for every subsequent use for this food product, then the Save to Database button is used to accomplish that.

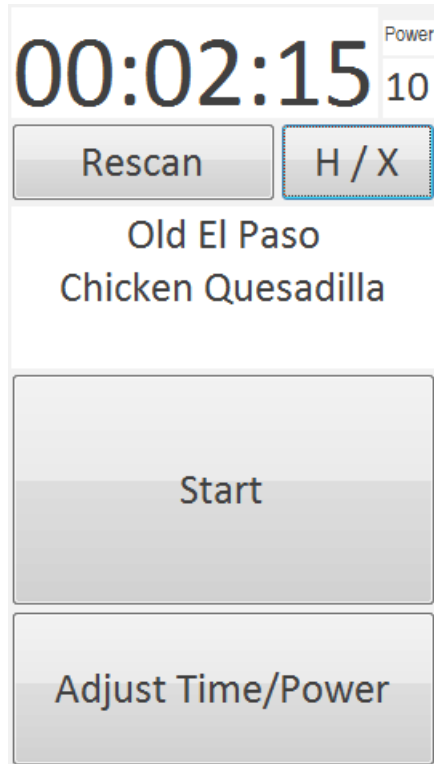


Figure 4.3.7-10:
Auto operation complete

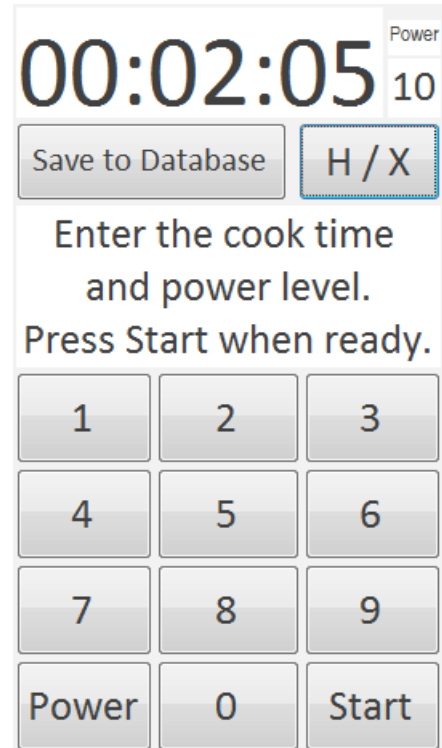


Figure 4.3.7-11:
Adjust auto settings

Figure 4.3.7-12 shows the software activity for these options. The process of changing the cook time and power levels, shown as a single activity in this flowchart, is identical to the process for manual control previously shown in Figure 4.3.7-4. The primary difference here is the optional ability to save the altered settings to the local database. Finally, if the user erroneously pressed the Adjust Time/Power, then Start can be pressed without making any changes and the microwave will proceed using the settings already retrieved from the database.

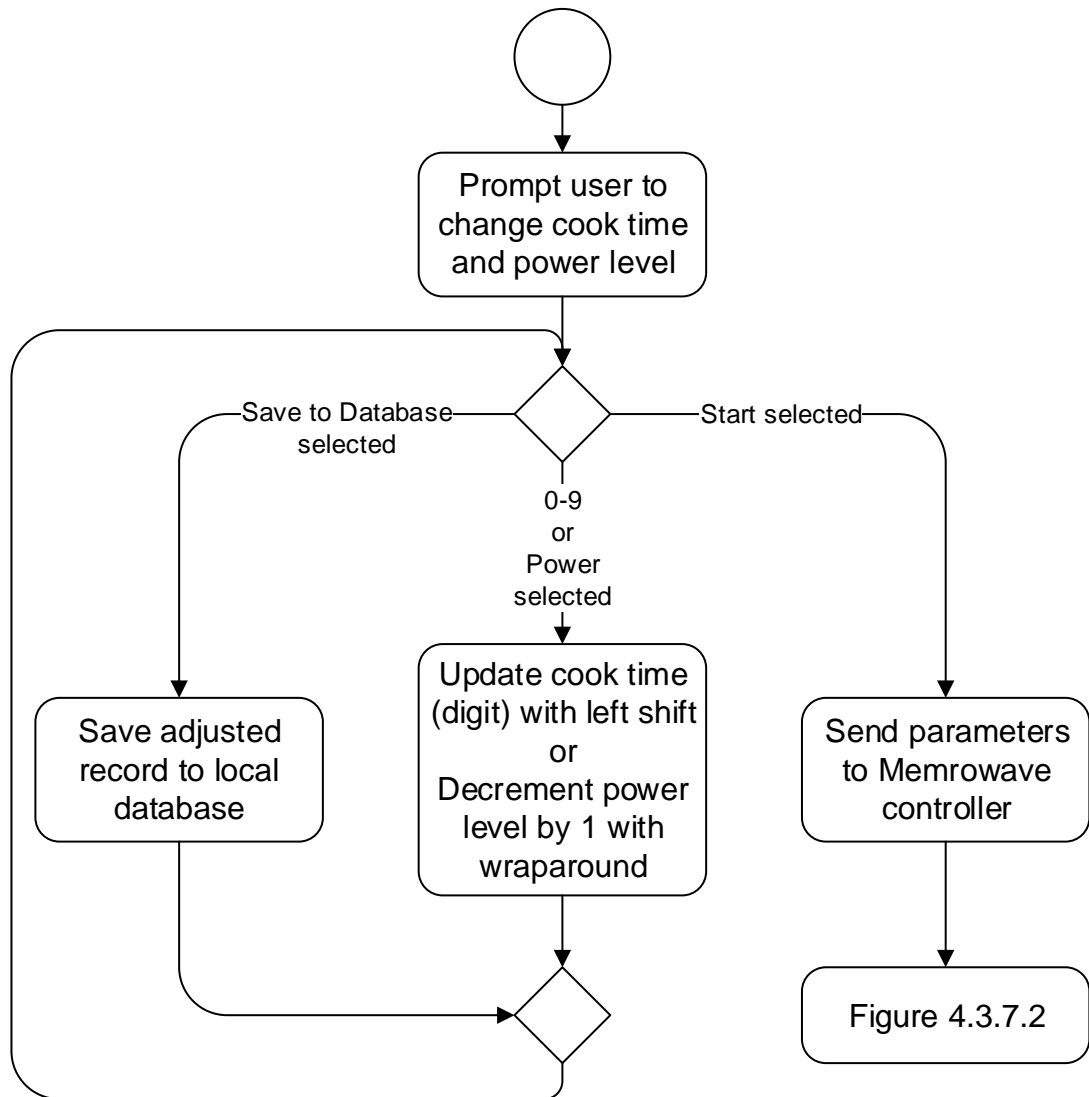


Figure 4.3.7-12: Automatic/scanned settings adjustments

Figure 4.3.7-13 shows the software activity if the processed barcode is not found in both the local and internet databases. The user is given a couple of options as shown in Figure 4.3.7-14. The first option is to just use manual control. By selecting manual control, the user intent is to prepare the product without saving any settings to the local database. The second option is to create a new local database record for future use. Since the barcode is unique and displayed each time it is scanned, the product name is not a requirement. Serving information is also optional. Therefore, it is possible to create a new record using just the scanned barcode, the cook time, and the power level. The product information is entered using various screens as shown in Figures 4.3.7-15, 4.3.7-16, and 4.3.7-17. The user may cancel at any point during this process and switch to manual control.

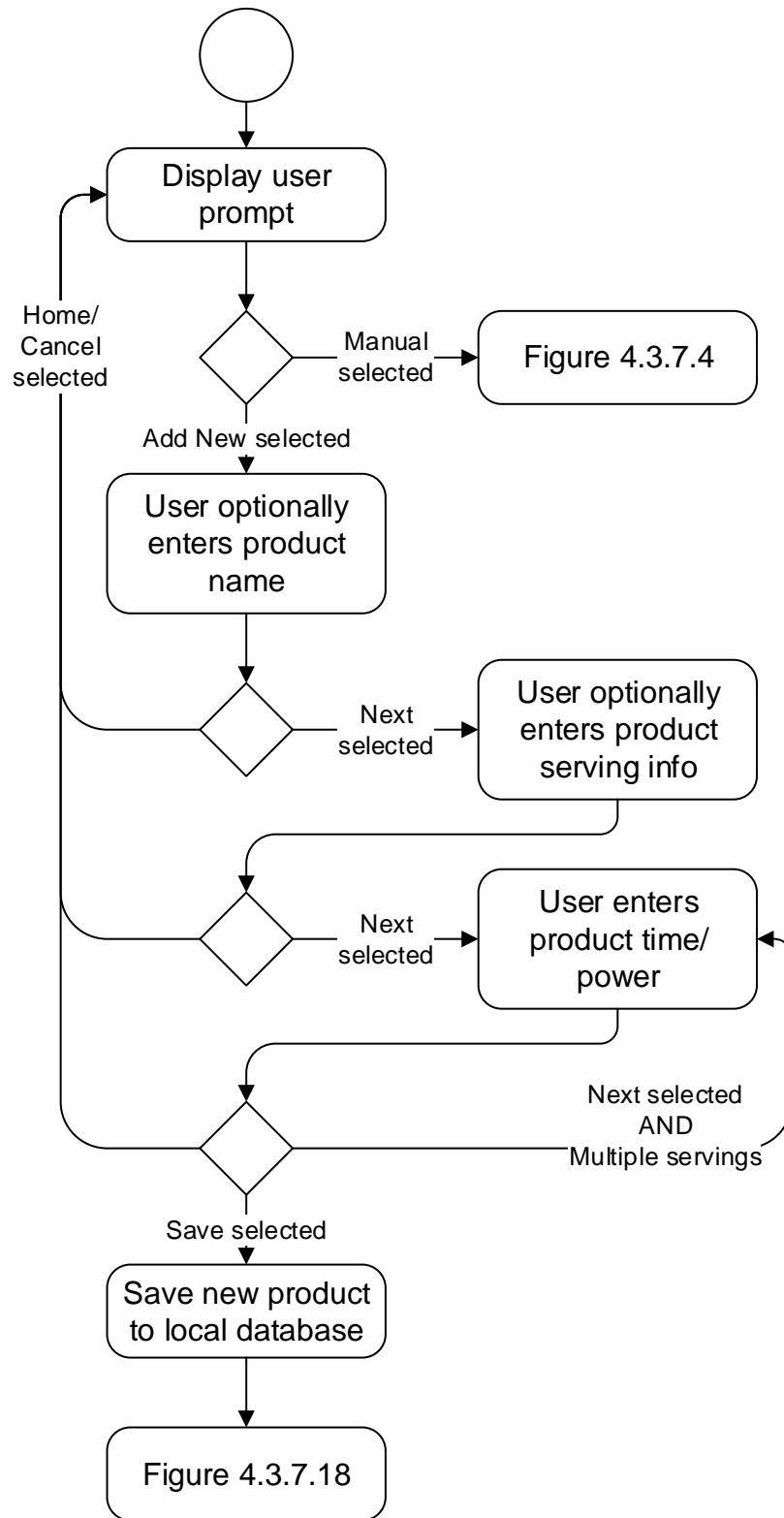


Figure 4.3.7-13: Database record not found

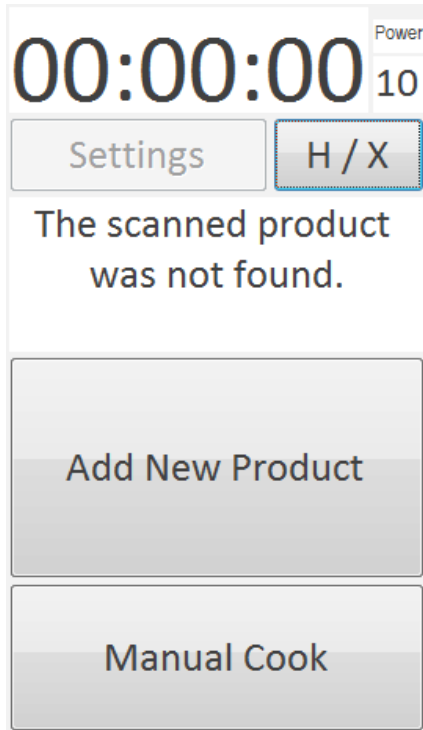


Figure 4.3.7-14:
Product not found

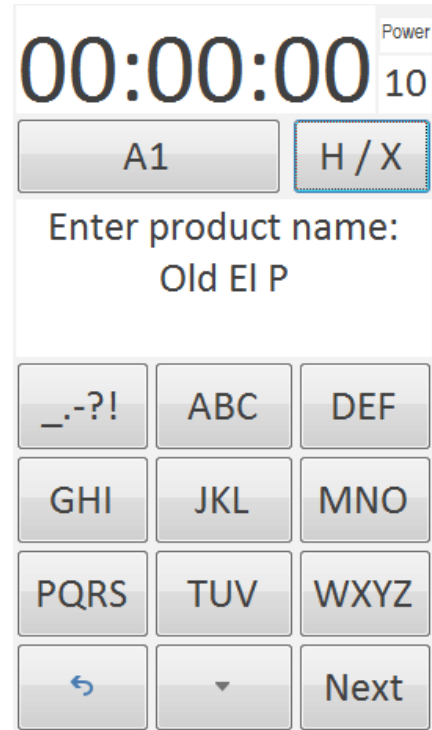


Figure 4.3.7-15:
Entering new product, upper case

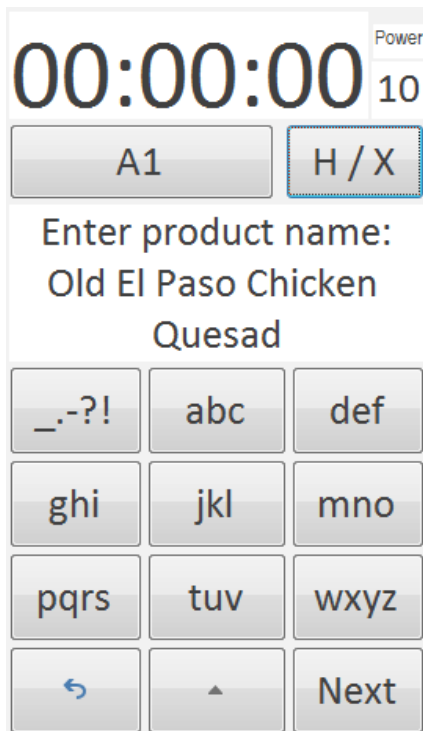


Figure 4.3.7-16:
Entering new product, lower case

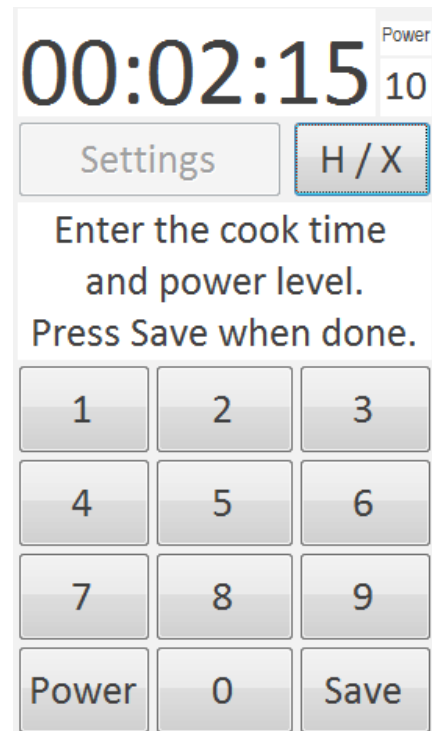


Figure 4.3.7-17:
Product cook settings

If the user has chosen to save the new product's settings, the product can be prepared immediately without returning to the scan menu and rescanning the barcode as shown in 4.3.7-18. However, it is also possible to cancel preparation at this point since it may have been the user's desire simply to enter the product in the database for later use without preparing the product at that moment. For example, this sequence could be used by a parent who wants to ensure a meal their child will prepare later is already included in the Memrowave's database. After this sequence, the interface returns to the main menu.

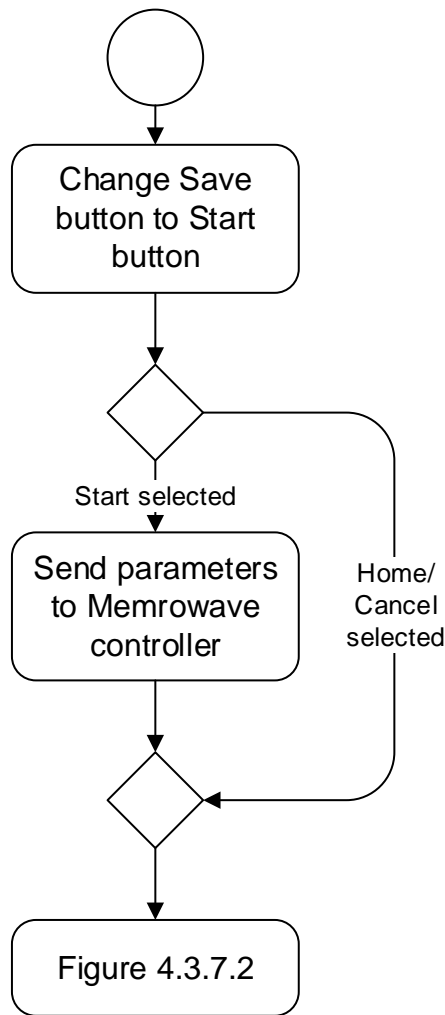


Figure 4.3.7-18: Use new local record

Along with a classic manual cook mode, the Memrowave will also provide quick presets similar to the dedicated buttons and quick settings found in the current microwaves analyzed for this project. These presets are called Quick Cook and are initially accessed from the manual control screen shown in Figure 4.3.7-3. The quick cook software activity is shown in Figure 4.3.7-19.

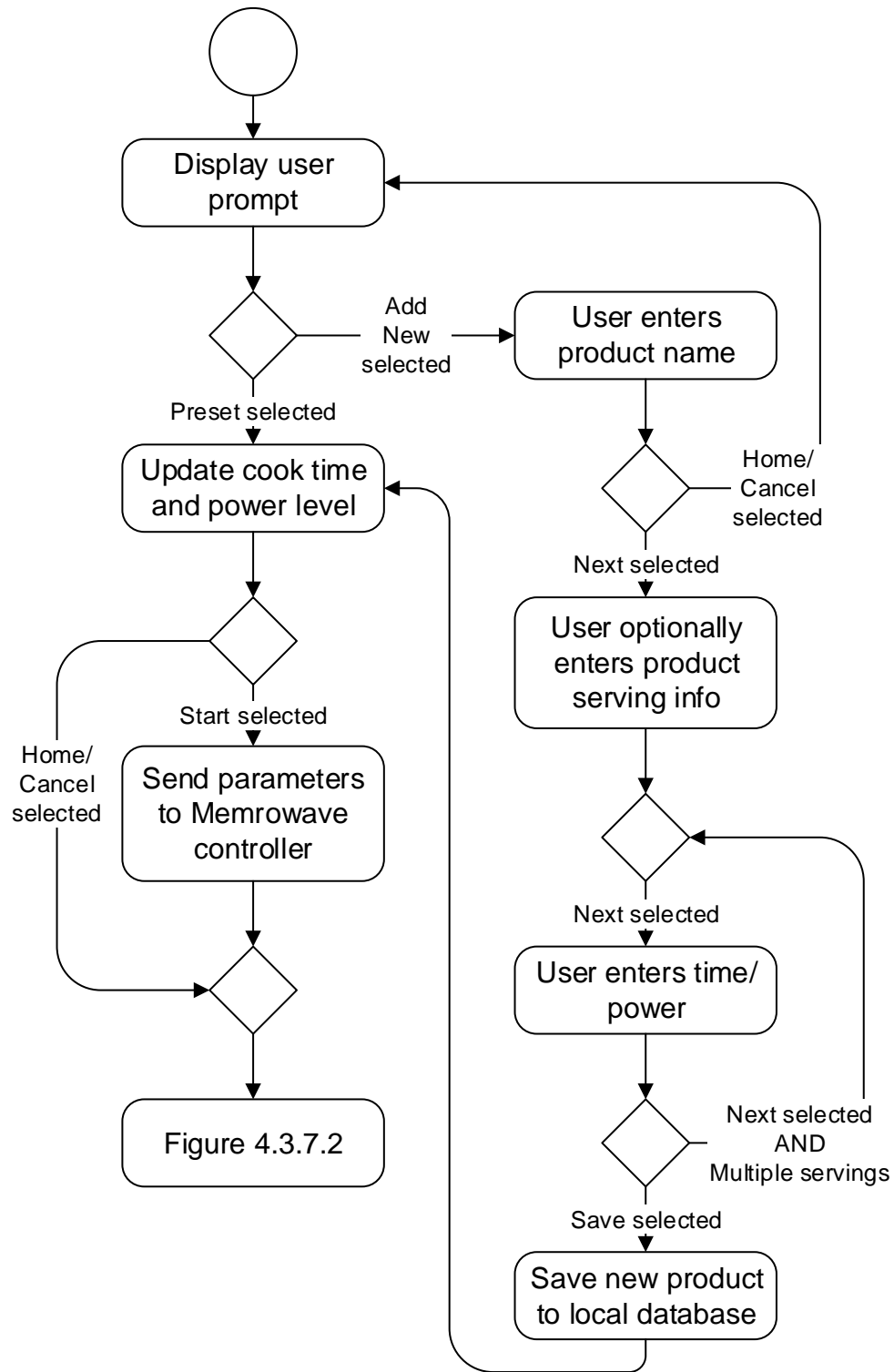


Figure 4.3.7-19: Quick Cook settings

Many of the presets seen in the research microwaves are unnecessary in the Memrowave since the barcode scanning feature will handle products such as

popcorn, canned vegetables, and frozen pizza. However, the quick cook feature will come with a number of standard presets for foods do not normally come with barcodes including potatoes and other fresh vegetables or when using a barcode might be inconvenient such as with large bags of frozen vegetables. This is a very flexible feature that will allow a user to use existing presets or add custom presets as they choose. The user selects the preset by swipe scrolling through a list of available presets as shown in Figure 4.3.7-20. When the desired preset is located, the user selects the preset button and the cook time and power levels are automatically set. The Start button starts the cooking process.

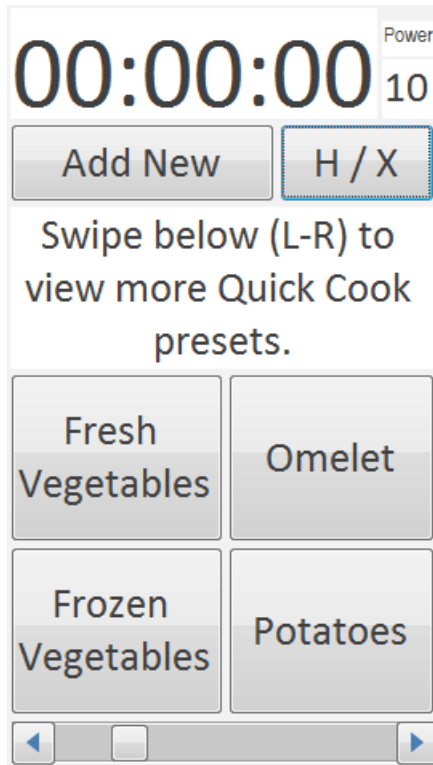


Figure 4.3.7-20: Quick Cook menu

The user can add a new preset at any time on this screen by selecting the Add New button. The steps and screens are similar to adding a new barcode record (Figures 4.3.7-15, 4.3.7-16, and 4.3.7-17) with the exception that the product name must be designated for quick cook presets since no barcode is associated with the record. The new preset can be entered without actually starting the Memrowave's cooking process. Since this feature is so flexible, the defrost and reheat functions will be consolidated with the Quick Cook function. This results in fewer menus and fewer actions that a user must remember in order to access all functionality of the microwave.

The final button on the main menu is the Settings button which is used for Memrowave settings as well as working with user profiles. A nominal user menu is shown in Figure 4.3.7-21. Selecting the Change Profile button opens a screen

similar to the Quick Cook menu where a user can swipe scroll through the various profiles and select one to set active. If the user creates a new profile, the option to swap to the new profile will be available immediately after saving the user record.

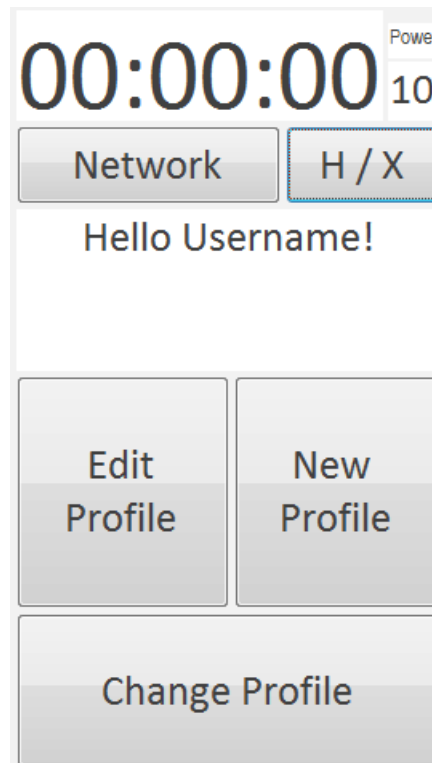


Figure 4.3.7-21: Settings menu

The software activity flow for the settings menu is shown in Figure 4.3.7-22. While not explicitly shown, selecting the home button at any time takes the user back to the main menu. Besides profiles, only network options are currently anticipated to be included, but it is likely that other options will need to be added as software is built and tested. If these other settings remain limited, then it will be possible to add them in a similar manner as the network settings, just changing the name of the provided button. However, a change to a swipe scrolling menu of settings similar to the Quick Cook screen in Figure 4.3.7-20 can be added if settings requirements increases.

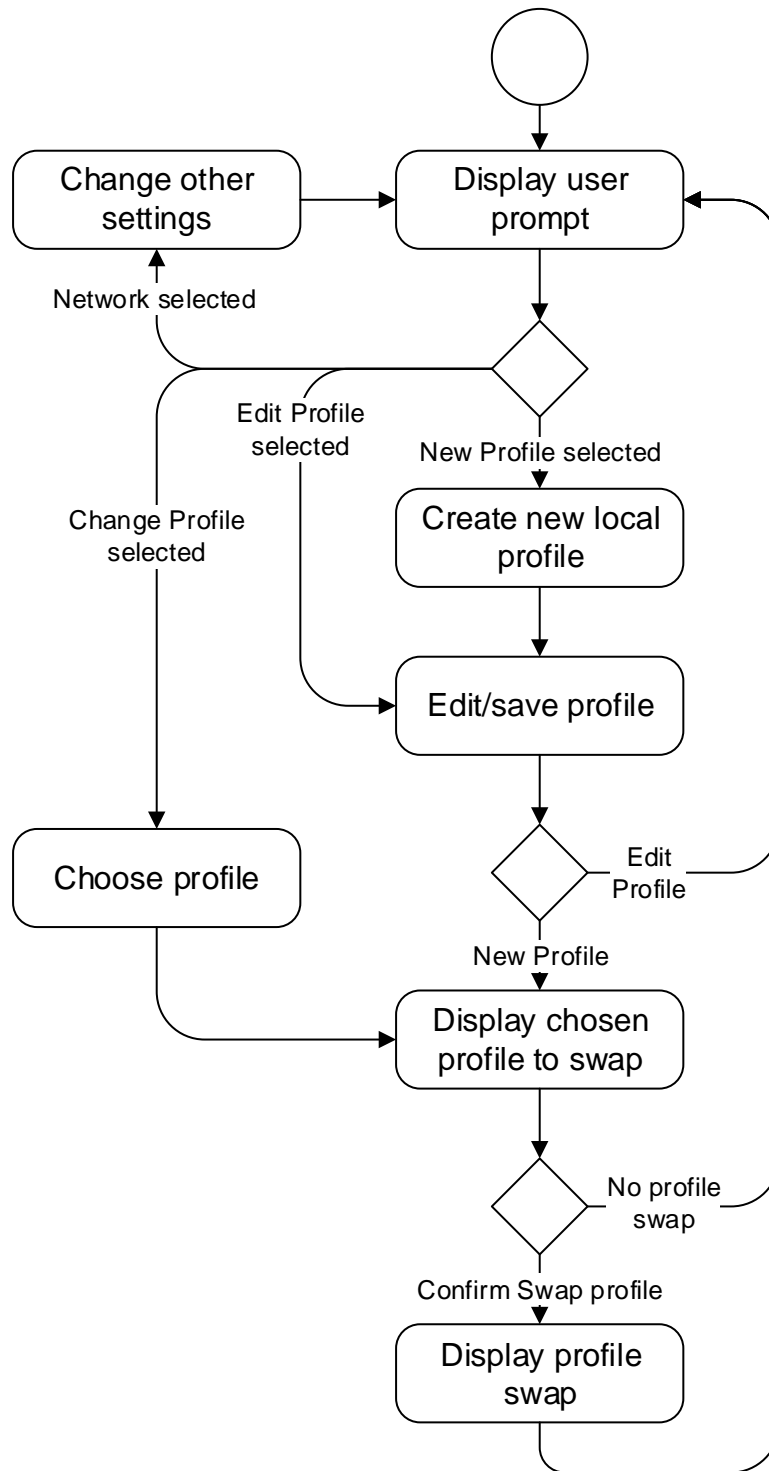


Figure 4.3.7-22: Memrowave profiles and settings

This design does not cover every possible scenario or minor feature, but it includes enough examples of software activity flow and user interface design that other minor or unforeseen functionality can be rapidly developed and

implemented, possibly using already implemented screens with only minor changes.

4.3.8 SMS Notifications

Sending SMS messages over the internet requires some type of SMS gateway server. There are commercial options available, but these usually require a monthly fee and charge for each message sent. For our project, a self-hosted SMS gateway server would be sufficient. The PlaySMS project is an open-source SMS gateway server that is licensed under the GPLv3 license, which gives us permission to use the software in our project. A link to the full text of the GPLv3 license can be found in Appendix A. Figure 4.3.8-1 specifies the notifier class.

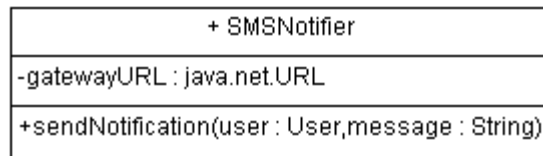


Figure 4.3.8-1: SMSNotifier class diagram

5 Design Summary

In section 4, we discussed the design details of the Memrowave for each subsystem. In this section, we will discuss the design of the overall system that will integrate all of the subsystems.

5.1 Software Design Summary

In section 4.3.6, we outlined the software design details of each part of the system. In this section, we will combine all of the subsystems into an overall design summary of the Memrowave's software. This design summary will show how we will implement the main program, which ties all of the separate parts together.

The main entry point to the program will act as an Android 'Launcher'. This will bring up the Memrowave's main interface whenever the Home button is pressed on the device's screen, effectively blocking the user from encountering the standard Android interface. Figure 5.1-1 shows class diagrams for several of the Android 'Activities' that make up the program.

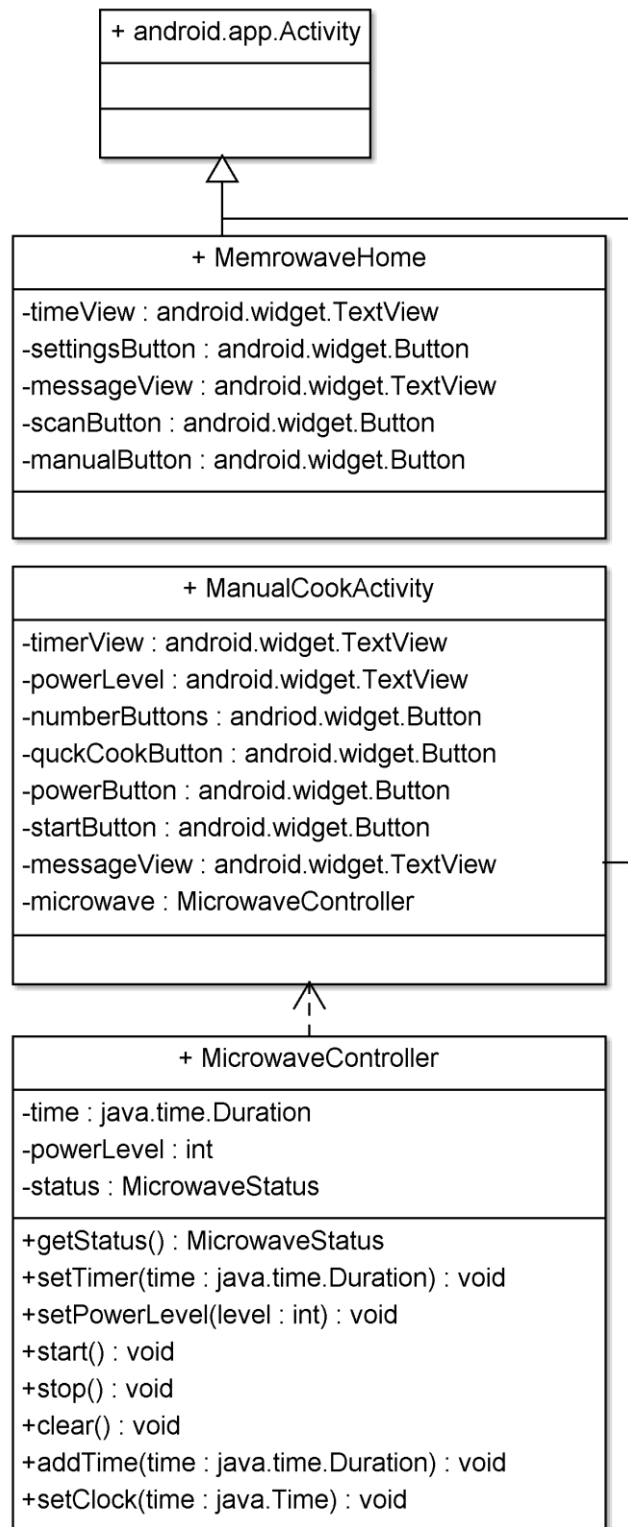


Figure 5.1-1: Class diagrams home screen, manual cook screen, and microwave controller class

The MemrowaveHome class will act as the Launcher replacement, so anytime the Home button is pressed, this Activity will be displayed on the screen. The user interface of the Activity was described in detail in section 4.3.7, and a mock-up of the screen was shown in Figure 4.3.7-1. Each element of the mock-up has its corresponding android-specific UI control element as an attribute in the class diagram. Also in Figure 5.1-1 is a class diagram for the ManualCookActivity. This class describes the screen that will be displayed when a user selects the 'Manual Cook' option on the main screen. Like the MemrowaveHome class, this class has UI control elements for each of the control elements of the mock-up shown in Figure 4.3.7-3. In addition to these attributes, the ManualCookActivity will have a reference to the MicrowaveController class, which will allow it to send commands to the microwave hardware. A class diagram of the MicrowaveController class is shown in Figure 5.1-1 as well. This class will use the I2C interface of the Beaglebone Black to send commands to the microwave control board, as described in detail in section 4.3.3.

Figure 5.1-2 shows three different Android Activity classes, including the BarcodeScanActivity class, ProductCookActivity class, and QuickCookActivity class. The BarcodeScanActivity class will be used to implement the barcode scanning screen of the Memrowave. A mock-up of this screen is shown in Figure 4.3.7-6 and Figure 4.3.7-7. The class diagram lists all of the Android-specific UI control elements that will be needed to implement this class, and it also includes a reference to the BarcodeScanner class, which will process the camera images to scan the user's barcode.

After a user scans a barcode and a product match is found, the ProductCookActivity will be displayed on the screen. A mock-up of this screen is shown in Figure 4.3.7-10. The ProductCookActivity class will implement the product cook screen. It will display information about a particular product, including the cook time, power level, name, etc. The ProductCookActivity will also give an option to modify the product's database entry to change the cook time and power level. Finally, the ProductCookActivity will have a button that will start the cooking process.

The QuickCookActivity class, which is also shown in Figure 5.1-2, will display a list of food products that may not necessarily have a barcode, and yet can be cooked automatically by the Memrowave, nonetheless. The class has a reference to the QuickCookDatabase class, which will access the local database to retrieve all of the quick cook product options. The list of quick cook items will then be displayed on the android.widget.ListView UI control element. When the user selects an item from this list, the callback method defined in the OnItemClickListener will be executed. This will load the ProductCookActivity with the selected quick cook product.

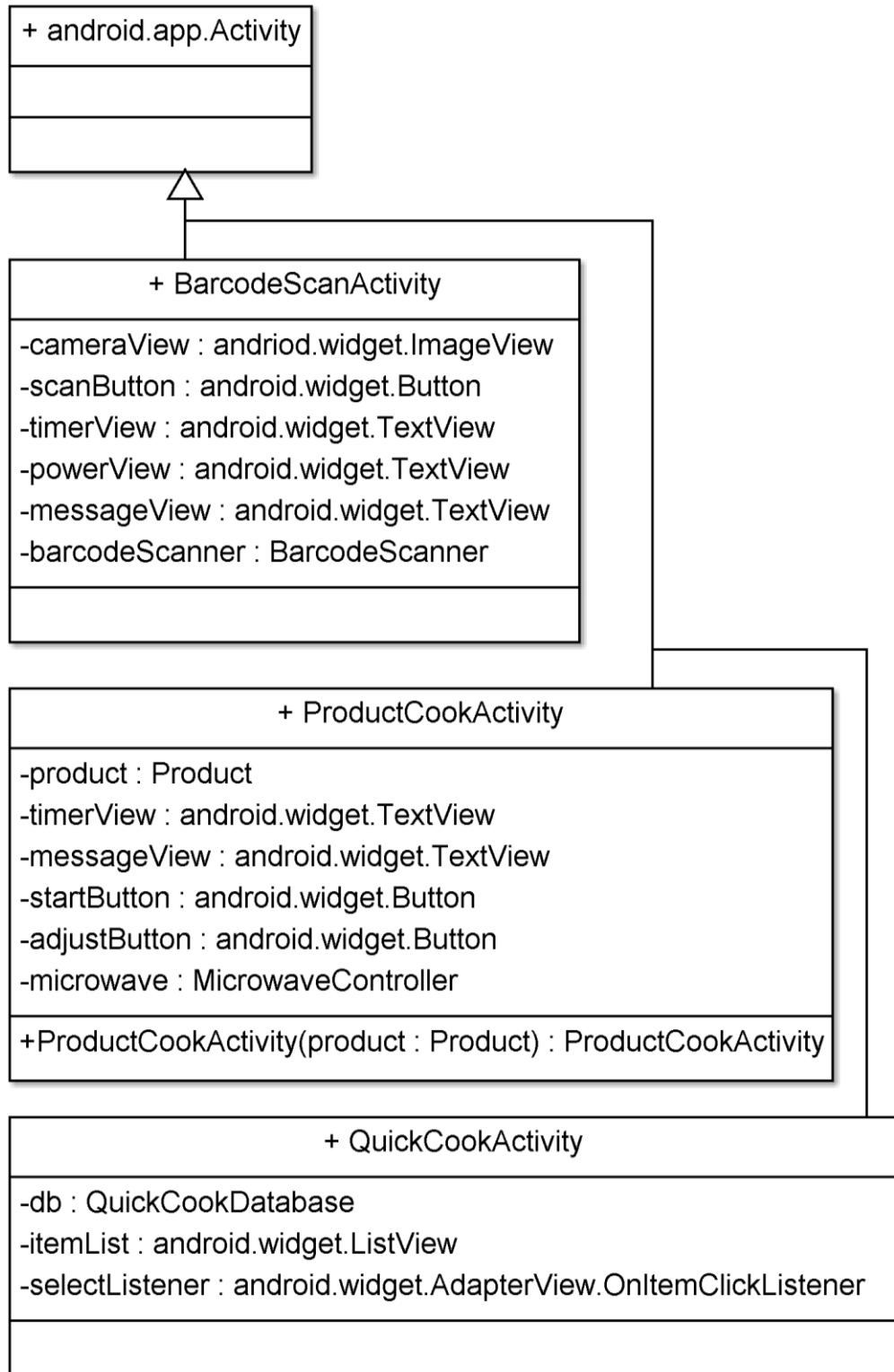


Figure 5.1-2: Class diagrams for the barcode scan screen, product cook screen, and quick cook screens

Figure 5.1-3 shows all of the Activities that pertain to the user profile feature. The first class shown in the diagram is the `NewUserActivity` class. This class will display a screen to create a new user profile. The class contains attributes for all of the UI control elements that will be needed to implement this functionality, and it also has a reference to the `UserDatabase` so that the new user can be saved to the local database. The next class in the diagram is the `ModifyUserActivity`. This class will be very similar to the `NewUserActivity`, except instead of having blank fields, they will be populated by the user's current settings. Finally, Figure 5.1-3 shows a class diagram for the `SelectUserActivity`, which will access the `UserDatabase` class to obtain a list of users. The list of users will be displayed on the `userList` `ListView` element. After a user is selected, the callback in the `OnItemClickListener` will be executed.

The `NewProductActivity` class, shown in Figure 5.1-4, will display a screen that allows the user to enter a new product into the local product database. A mock-up of this screen was shown in Figure 4.3.7-15. The class has references to a `TextView` UI element for the product name, a `TimePicker` UI element for the cook time selection, a `NumberPicker` UI element for the power level selection, and a few buttons, including a button to save the new product to the database and a button to scan the barcode of the new product. When the user presses the save button, the information entered in the UI elements will go through a verification process to be sure it's formatted correctly, and then the product will be added to the local product database using the `ProductDatabase` class reference.

The next class in Figure 5.1-4 is the `ProductAdjustmentActivity`. This class will give the user the option of modifying the cook time and power level of a particular product. A mock-up of this screen is shown in Figure 4.3.7-17. This class is similar to the `NewProductActivity` class, except the only user-modifiable options will be the cook time and power level.

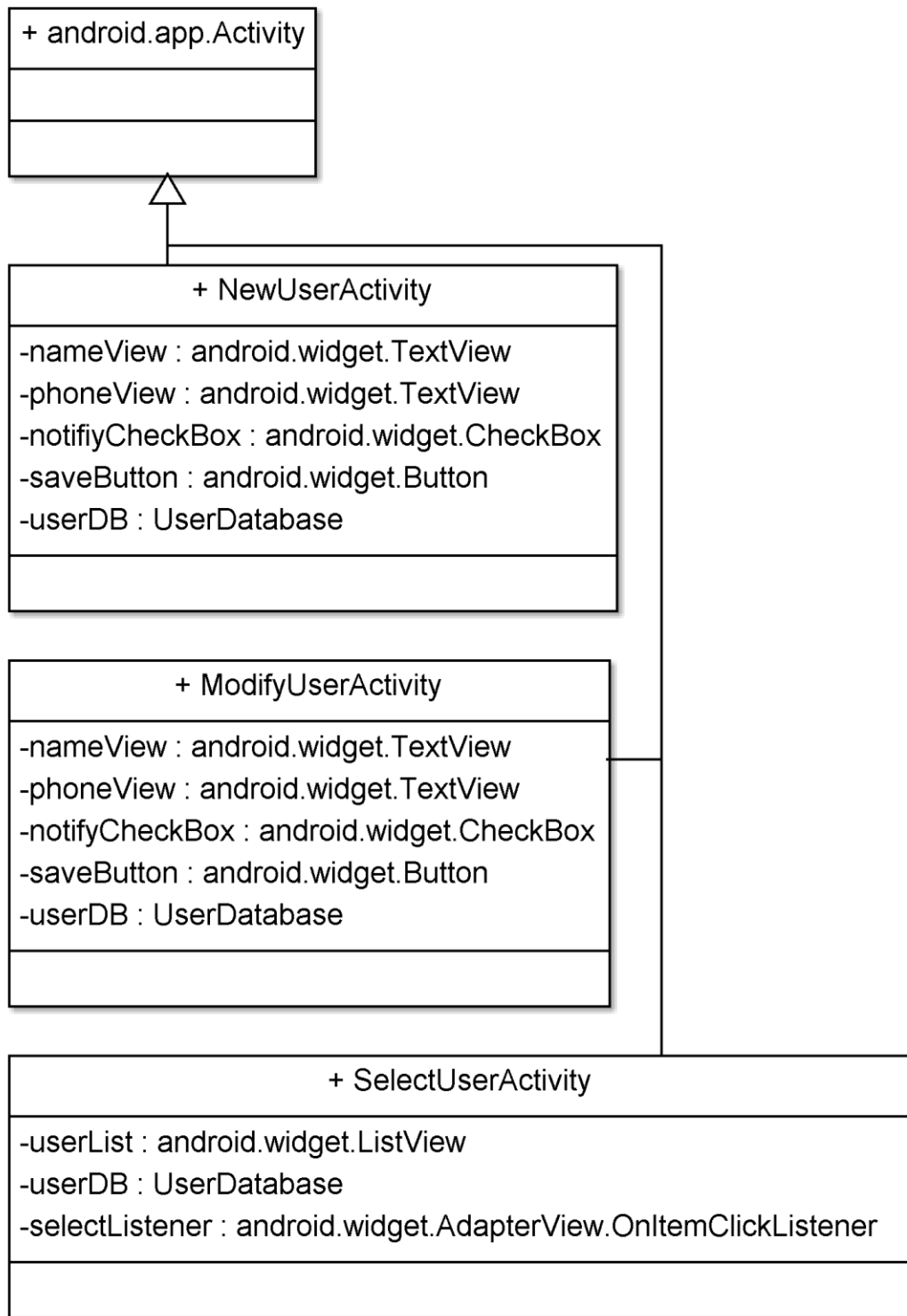


Figure 5.1-3: Class diagrams for the new user screen, modify user screen, and user selection screen

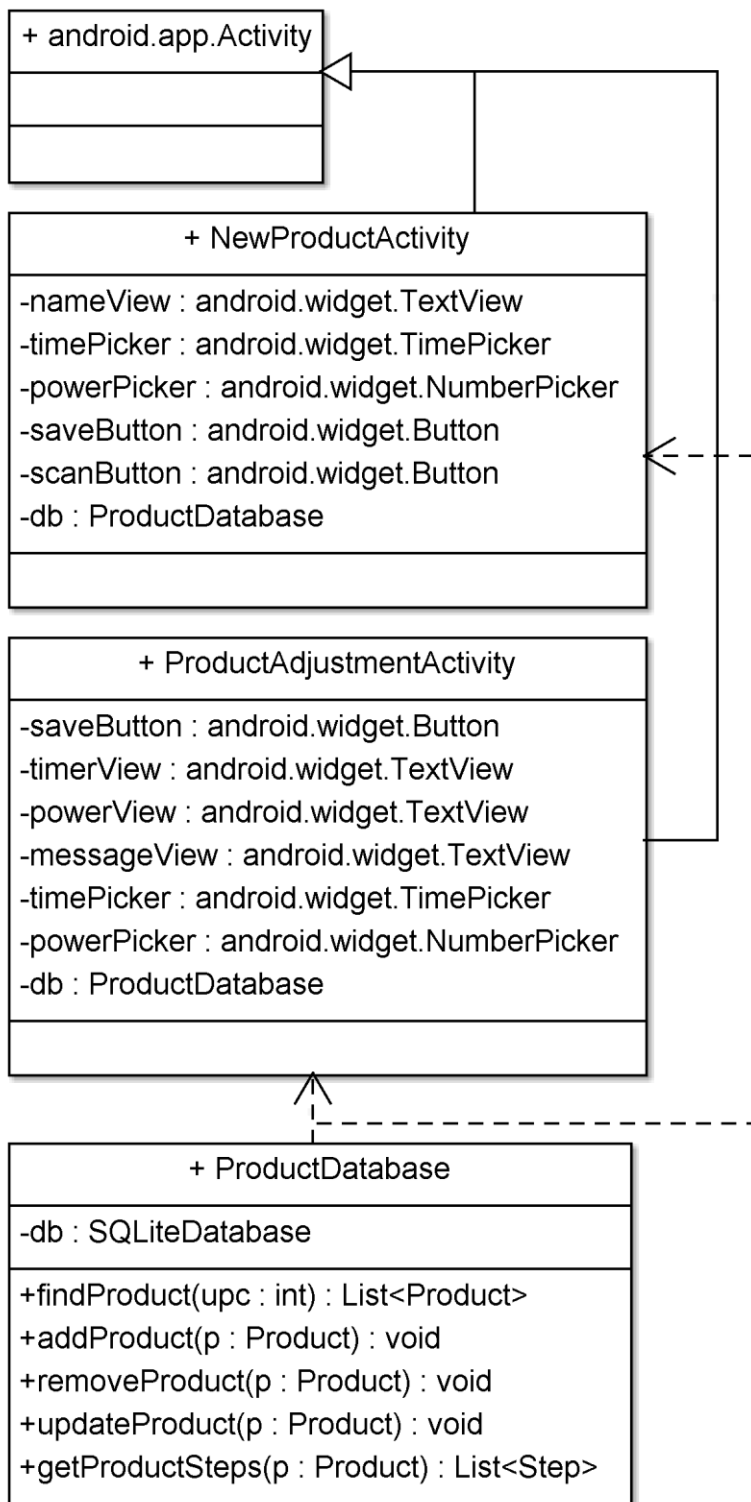


Figure 5.1-4: Class diagrams for the new product screen, modify product screen, and the product database

5.2 Hardware Design Summary

In this section, we will provide a high-level overview of the hardware of the Memrowave.

A schematic diagram for the microwave control board is shown in Figure 5.2-1. The schematic has the MSP430 microcontroller that will control the microwave hardware. The I2C pins on the MSP430 will be connected to the I2C pins of the BeagleBone via the header shown in the schematic. The MAX7219CNG LED display driver is also shown in the schematic. This part uses 5V logic for the serial communications, while the BeagleBone Black uses 3.3V. To make the two work together, a logic level shifter circuit was used for each of the three serial communication lines. The LED display will be connected to the board via a ribbon cable, so the relevant pins of the MAX7219CNG were connected to a 12 pin header in the schematic.

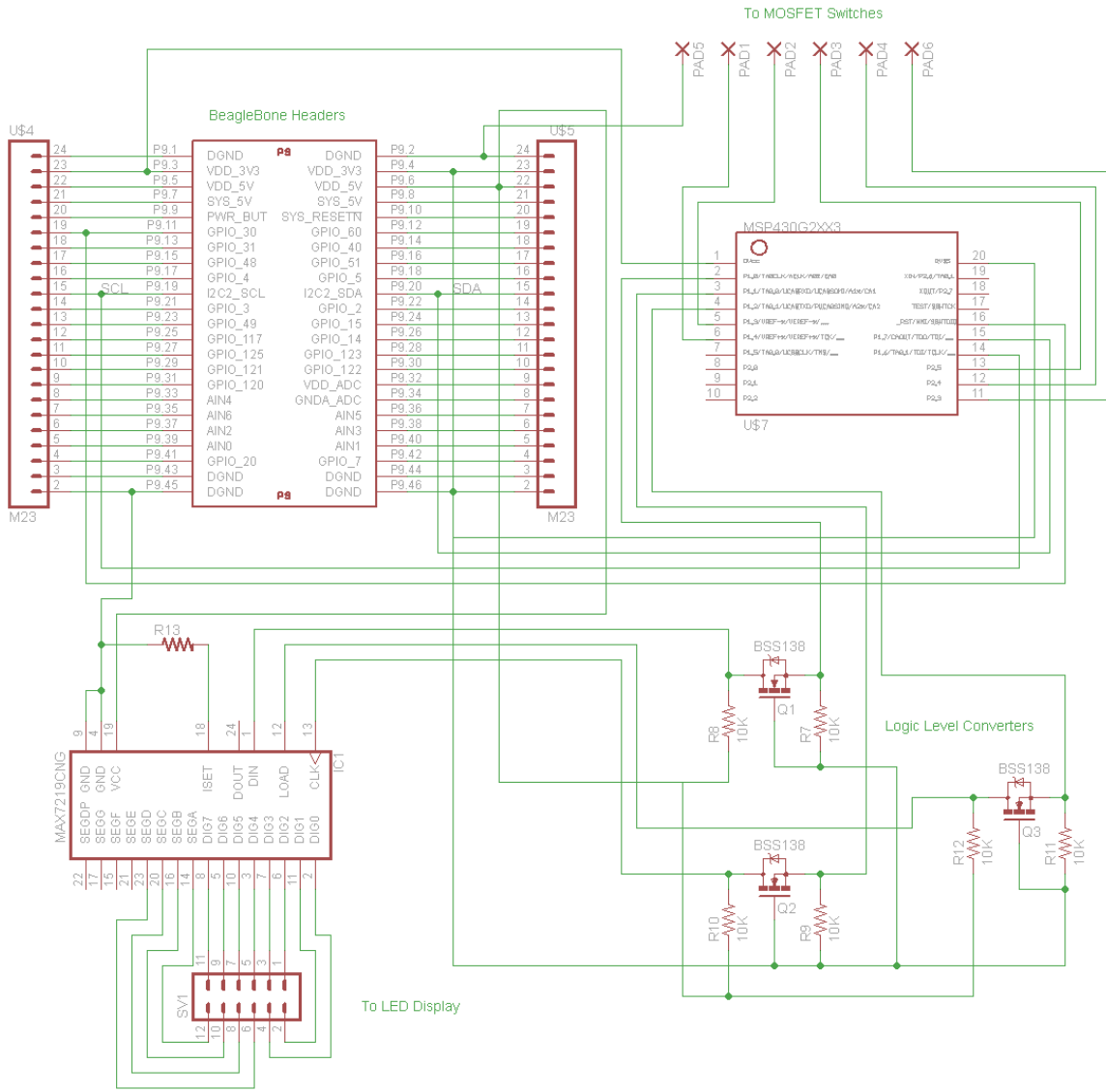


Figure 5.2-1: Microwave control board schematic

Figure 5.2-2 shows a PCB design for the schematic in Figure 5.2-1. Design files from SparkFun Electronics' BeagleBone Black prototyping cape were modified to create our own custom board design. SparkFun Electronics made their design files for the prototyping cape available under the CC BY-NC-SA 3.0 license, which gives us permission to use the design files in our project. All of the parts that were not relevant to our project were removed from the design files, and our own circuits were then inserted in their place. We then positioned the parts on the board and routed the appropriate connections.

Figure 5.2-3 shows how the circuit from Figures 5.2-1 and 5.2-2 will be connected to the various microwave components. The switches shown in the figure represent MOSFET based solid-state relay circuits that will be used to control the microwave hardware.

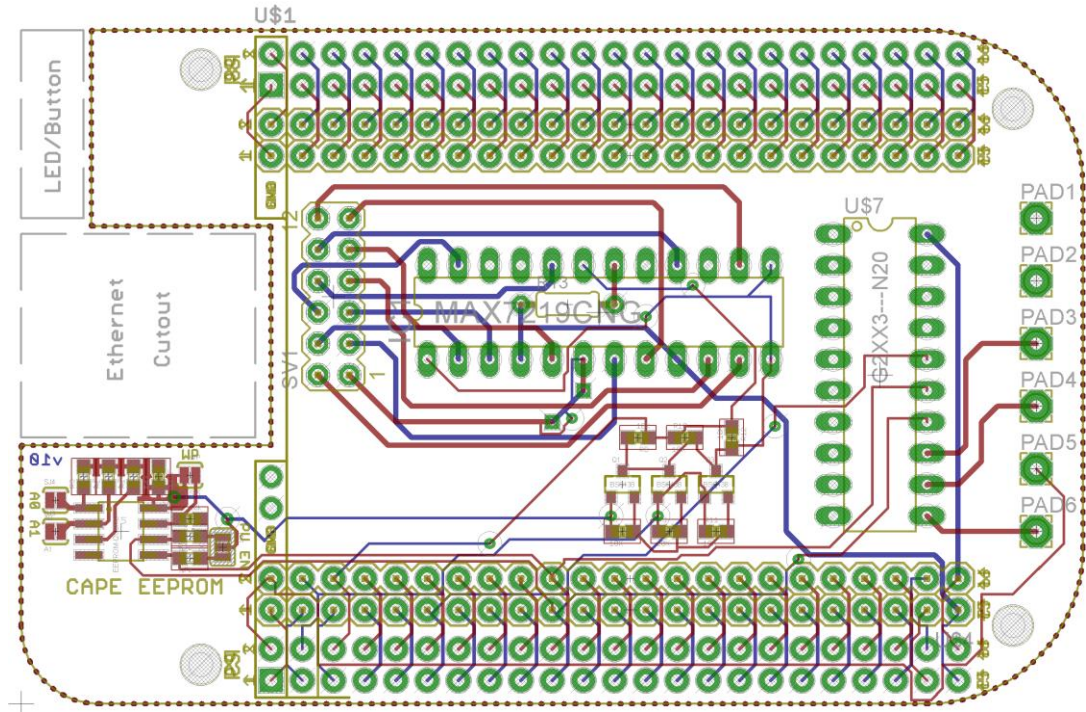


Figure 5.2-2: Microwave control circuit 'Cape' for the BeagleBone Black

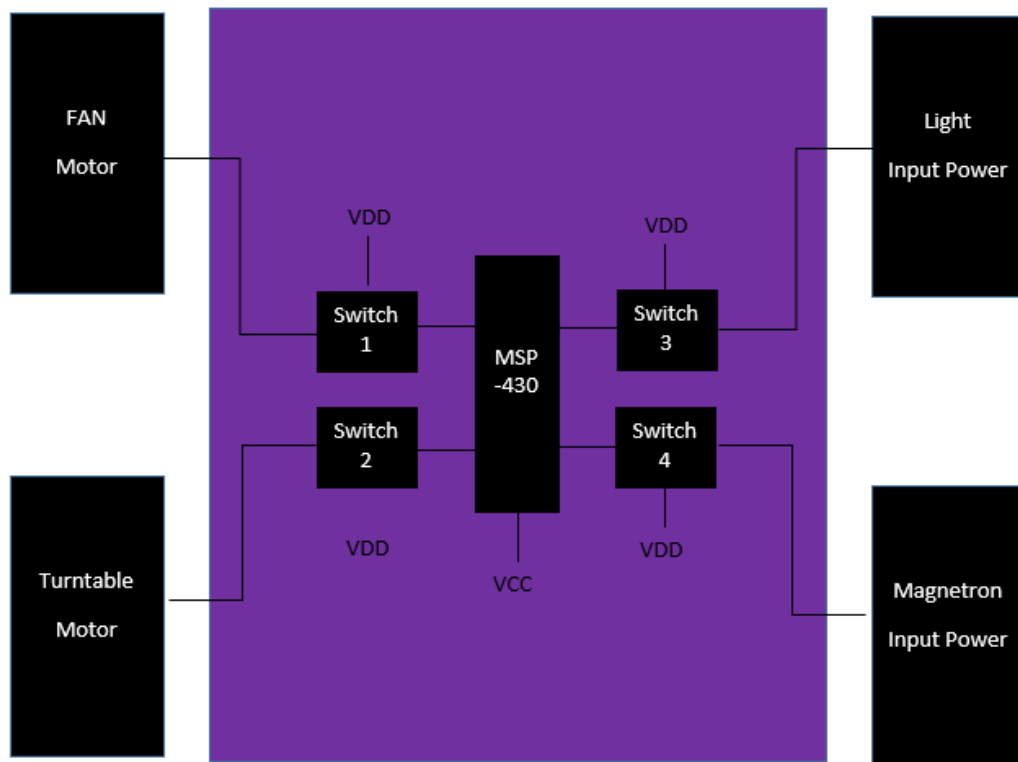


Figure 5.2-3: Microwave hardware control diagram

6 Project Prototype Construction and Coding

In section 6, we will discuss how the design of the Memrowave will be implemented.

6.1 Parts Acquisition and BOM

An outline for the specific parts and components for our project has been developed. All the parts will be divided into sections and will have detailed information based on the following criteria:

- Division between hardware and software application.
- Relative importance of the specific item.
- Difficulty in acquiring the item.
- Part manufacturers and locations.
- Engineering time frame for receiving the item.

This outline will serve the basis for what and who is responsible for the certain components and parts are required for the Memrowave. The division between software and hardware is obvious due to the need for separate hardware and software design that is necessary and later integrated together. The hardware will include mainly all the electrical components needed to power the Memrowave, the many parts needed on the printed circuit board which must be designed, and any external and internal parts needed to mount any of the active components of the entire system together to fit in or on the microwave. In our project many of the components are definitely necessary such as the power supply and processing unit, while other are of lesser importance like a specific type of mounting bracket. Giving each part or component a certain level of importance will help in setting a certain value for getting the product developing a plan for acquiring it in a timely fashion. Difficulty in finding and receiving certain parts is also an issue which must be addressed. This could be the difference in getting a part or not to meet our deadline and/or budget. Part manufacturers and locations also play a part in acquiring certain items. Ideally we would like all our parts to be available within the United States in order to secure relatively quick turnaround shipping times and to limit any issues with receiving goods from overseas. An engineering time frame for the each part and component is laid out in order to setup a starting block for each item as it is needed for integration into the final engineering the Memrowave and to keep an up to date schedule throughout the development phase.

The working outline on the next page shows many of the parts and components needed for the Memrowave as well as some of the specifics details about acquiring such items:

Part Acquisition Outline

- 1) Processing Unit
 - a) Division
 - i) Software Application
 - b) Priority
 - i) High
 - c) Part Manufacturers/Retailers
 - i) Texas Instruments - United States
 - ii) Sparkfun Electronics - United States
 - iii) Arduino - available in the United States
 - iv) Intel - United States
 - d) Acquisition Difficulty
 - i) Low
 - e) Engineering Time Frame
 - i) Early
- 2) LCD Touchscreen Module
 - a) Division
 - i) Software
 - b) Priority
 - i) High
 - c) Part Manufacturers/Retailers
 - i) Sparkfun Electronics - United States
 - ii) Arduino - available in the United States
 - iii) 4D Systems - United States
 - iv) Digi-Key - United States
 - d) Acquisition Difficulty
 - i) Low
 - e) Engineering Time Frame
 - i) Early
- 3) Camera Module
 - a) Division
 - i) Software
 - b) Priority
 - i) High
 - c) Part Manufacturers/Retailers
 - i) Sparkfun Electronics - United States
 - ii) Digi-Key - United States
 - iii) e-con Systems - United States
 - iv) Byd IT - available in the United States
 - d) Acquisition Difficulty
 - i) Low
 - e) Engineering Time Frame
 - i) Early

- 4) WIFI Module
 - a) Division
 - i) Software
 - b) Priority
 - i) High
 - c) Part Manufacturers/Retailers
 - i) Sparkfun Electronics - United States
 - ii) Digi-Key - United States
 - iii) WhizNets - United States
 - iv) Microchip - United States
 - d) Acquisition Difficulty
 - i) Low
 - e) Engineering Time Frame
 - i) Early
- 5) AC DC Converter
 - a) Division
 - i) Hardware
 - b) Priority
 - i) High
 - c) Part Manufacturers/Retailers
 - i) Delta Electronics - United States
 - ii) Digi-Key - United States
 - iii) Sanken Electric -available in the Unites States
 - iv) CUI Inc. - United States
 - d) Acquisition Difficulty
 - i) Low
 - e) Engineering Time Frame
 - i) Early/Middle
- 6) Voltage Regulator
 - a) Division
 - i) Hardware
 - b) Priority
 - i) High
 - c) Part Manufacturers/Retailers
 - i) Delta Electronics - United States
 - ii) Digi-Key - United States
 - iii) Sanken Electric -available in the Unites States
 - iv) CUI Inc. - United States
 - d) Acquisition Difficulty
 - i) Low
 - e) Engineering Time Frame
 - i) Early/Middle

Many of the common place parts in or on a regular microwave such as the turntable and lights will either be salvaged from existing microwave assemblies

or replacements parts will be used in their place. A list of these items are shown below:

- Internal Lights
- Internal Beeper
- Clock Display
- Magnetron
- Power switch
- RF shielding
- Door assemblies

Bill of Materials

The bill of materials will include all items that will need to be purchased in the course of the conceptualizing phase of the Memrowave. Details about certain miscellaneous items will be hard to pin down due to unforeseeable changes and alterations which may need to be done such as additional supports for assembling and mounting parts, wiring details. Due to the nature of some of the components and parts, whether to design a part itself or to buy the part also plays a factor in the bill of material list based on available options, the flexibility, usable, and cost effective must be considered. Below in Table 6.1-1 is a working outline of all the components and parts that need to be purchased with typical prices, as well as money which will be allocated to certain items were a specific monetary amount can't be given at this time.

Bill of Materials			
Item(s)	Cost Per Item	Quantity	Total
Processing Unit	\$60	1	\$60
WIFI Module	\$23	1	\$23
Camera Module	\$32	1	\$32
LCD Touchscreen	\$140	1	\$140
Power Switch	\$17	2	\$34
Wire	\$30	1	\$30
Total Estimated Cost			\$353

Table 6.1-1: Rough Bill of Materials Estimate

6.2 PCB Vendor and Assembly

For the Memrowave a reliable printed circuit board assembly is needed for the additional components we will be adding to the Memrowave. Since the PCB will be mainly for the power supply and power regulation, the design of the PCB will most likely be of one layer, measuring at most a maximum of 4 inch by 3 inch to keep as compact a design as possible while allowing space to fit all the needed board-level parts. A minimum of three boards would be desirable for design and testing purposes. In the final stages of the design and assembly another board would be desirable after any changes that need to be made. The options we have in making this PCB is either to do self-fabrication or to use a PCB vendor and/or assembler to do the fabrication and/or assembly for us. Through self-fabrication we have the quickest turnaround time, the ability to make minute changes as we see fit even during the process, there is some opportunity for less waste in that regard. Disadvantages to self-fabrication of the PCB is ideally the lack of precision which can be acquired through PCB vendors the PCB will undoubtedly need to be larger to avoid error, also on the assembly side as well problems can arise with soldering parts which could weaken the reliability and functionality of the final PCB. Through the PCB vendors there is the ability to have them create multiple boards with great accuracy and precision, with limit errors due to their rigorous testing, experience and equipment. Also some vendors will also perform assembling of parts as well as fabrication of the PCB. The available PCB vendors are compared by the following criteria:

- Location
- Turn around/Delivery times
- Cost based on our needs
- PCB design software choices
- Assembling capabilities

These criteria will set the basis for making the preliminary and later final decision on which PCB vendor will be most ideal for this project. The turnaround time will be crucial as we have a limited time frame for testing and assembling the final appliance. To manage the budget properly a cost based on a quantity of three 4 inch by 3 inch PCBs comparison will aid in allowing for the proper allocation of money for multiple fabricated printed circuit boards. Many PCB vendors have different requirements and options as far PCB design software is concerned, some allow CAD files, other allow the use of the Eagle freeware PCB design software, and some have their own proprietary software. Assembling in addition to fabrication of our PCB would be ideal due to the possibility of making the PCB as small as possible with great accuracy.

The following outline is a list of the possible vendors and the details described to make a decision which PCB vendor is ideal.

PCB Vendor Outline

- 1) PCB-Pool
 - a) Location
 - i) In the USA
 - b) Turn-Around time
 - i) 1-8 days based on needs
 - c) Cost
 - i) \$60-250 based on delivery time needs
 - d) PCB Software choices
 - i) Eagle CAD
 - ii) PCAD
 - iii) Target
 - iv) Sprint
 - e) Assembling capabilities
 - i) No
 - f) Other Services
 - i) Electrical testing
- 2) OSH Park
 - a) Location
 - i) In the USA
 - b) Turn-Around time
 - i) 12 days
 - c) Cost
 - i) \$5 per square inch= \$60
 - d) PCB Software choices
 - i) Eagle CAD
 - ii) Gerber CAM files
 - e) Assembling Capabilities
 - i) No
- 3) ExpressPCB
 - a) Location
 - i) In the USA
 - b) Turn-Around time
 - i) 3 days
 - c) Cost
 - i) \$97
 - d) PCB Software choices
 - i) ExpressPCB - proprietary but free
 - ii) ExpressSCH - proprietary but free
 - e) Assembling Capabilities
 - i) No

- 4) Advanced Circuits
 - a) Location
 - i) In the USA
 - b) Turn-Around time
 - i) 5 days, next day delivery available
 - c) Cost
 - i) \$99
 - d) PCB Software choices
 - i) PCB Artist - proprietary but free
 - e) Assembling Capabilities
 - i) Yes
 - f) Other Services
 - i) Electrical test

6.3 Final Coding Plan

For the final coding plan, we will be splitting the software into sections and assigning each section to a group member. Winston Todd and Jack Gulick are in charge of the software, so the sections will be split between them. Jack will be put in charge of developing the software for the microwave control board described in section 4.3.3. This will include developing software to control all of the microwave components in the correct sequence and developing the software to meet the I2C interface specifications listed in section 4.3.3. He will also be in charge of programming the user interface described in section 4.3.7. This will include programming all of the UI controls for each Activity listed in the class diagrams shown in Figure 5.1-1.

Winston will be in charge of implementing the barcode scanner class described in section 4.3.4. This will entail interfacing the Zxing barcode software library with the Memrowave's software. He will also be in charge of implementing the software to interface with the microwave control board, as described in the class diagram in Figure 4.3.3-2. Section 4.3.5 describes the local product database and provides class diagrams in Figure 4.3.5-2. Winston will be in charge of implementing these classes to meet those specifications. In addition to the local database, Winston will be in charge of implementing a master database server and a database update feature to sync the local database to the master database. Finally, Winston will implement the user profiles as described in the class diagrams shown in Figure 4.3.6-2.

After implementing those software sections, we will work together to combine all of those subsystems into the final software system. Section 5.1 shows class diagrams of all of the class that need to be implemented. These classes will tie together all of the subsystems into the final product.

7 Project Prototype Testing

It is important to create well designed tests for the project to ensure that any bugs in either the hardware or software will be caught. With this in mind, we will proceed to detail the testing of both the Memrowave's hardware and its software.

7.1 Hardware Test Environment

In this section we will discuss the testing environment for our prototype's hardware. We will provide information on the testing equipment available to our design group, what equipment will be used in testing, the labs we have access to, and go over possible beneficial test to run. Understanding what we have at our disposal will better help our testing during the prototype design phase. We need to present a clear list of possible equipment that can aid in troubleshooting problems with our design. Having the proper equipment will allow us to determine characteristics such as signal to noise ratio (SNR), device noise floor, and check for proper input/output levels. This list continues much further and is imperative to troubleshooting any issues we may encounter.

The first device to consider is a signal generator. The signal generator function as a source for input signals. It allows us to test to analyze multiple different input responses such as trigonometric, sync functions, square waves, and much more. This source also allows testing of different frequencies. The importance of the signal generator in our testing environment is its ability to input these wide range of inputs. For example the signal generator is needed to test the output of our MOSFET switching circuits. Using built in functions from the signal generator we can predict what the output from our switching circuits will be. This type of testing is crucial to prototyping our design.

Testing the outputs of the microcontrollers can be accomplished by using a DC power supply. This simple equipment allows for our design group to testing input/output pins on our controllers. We also need the DC power supply during testing of our switching circuits. These circuits require biasing power to operate. During testing of these subsystems via bread board the DC power supply is required to bias elements.

Understanding the frequency response of the Memrowave's hardware and circuits can also prove to be very important when characterizing our system. When characterizing the system we want to understand how different frequencies may affect our switches, filters, and possibly couplers. In order to run these test we need access to a sweep generator and a spectrum analyzer. The sweep generator functions by creating a series for frequency changing impulses of non-varying magnitude. This allows for a large measurement of frequency response over a short time. To measure the frequency response of our systems we need to acquire a spectrum analyzer. The benefit of this device in our testing

is its ability to span a large range of frequency a one time. Running the spectrum analyzer can give us information like filter pass bands, how switches are affected by frequency changes, and the change in dB from the pass band to the noise floor of filters. Using these two pieces of equipment will also allow us to determine if our system provides any unwanted radio frequency feed through. An example of a test we will like to run is if the magnetron energy can couple to our circuits or controllers. Being able to characterize this issue will determine whether or not we need shielding on any of our subsystems. Figure 7.1-1 shows a frequency response.

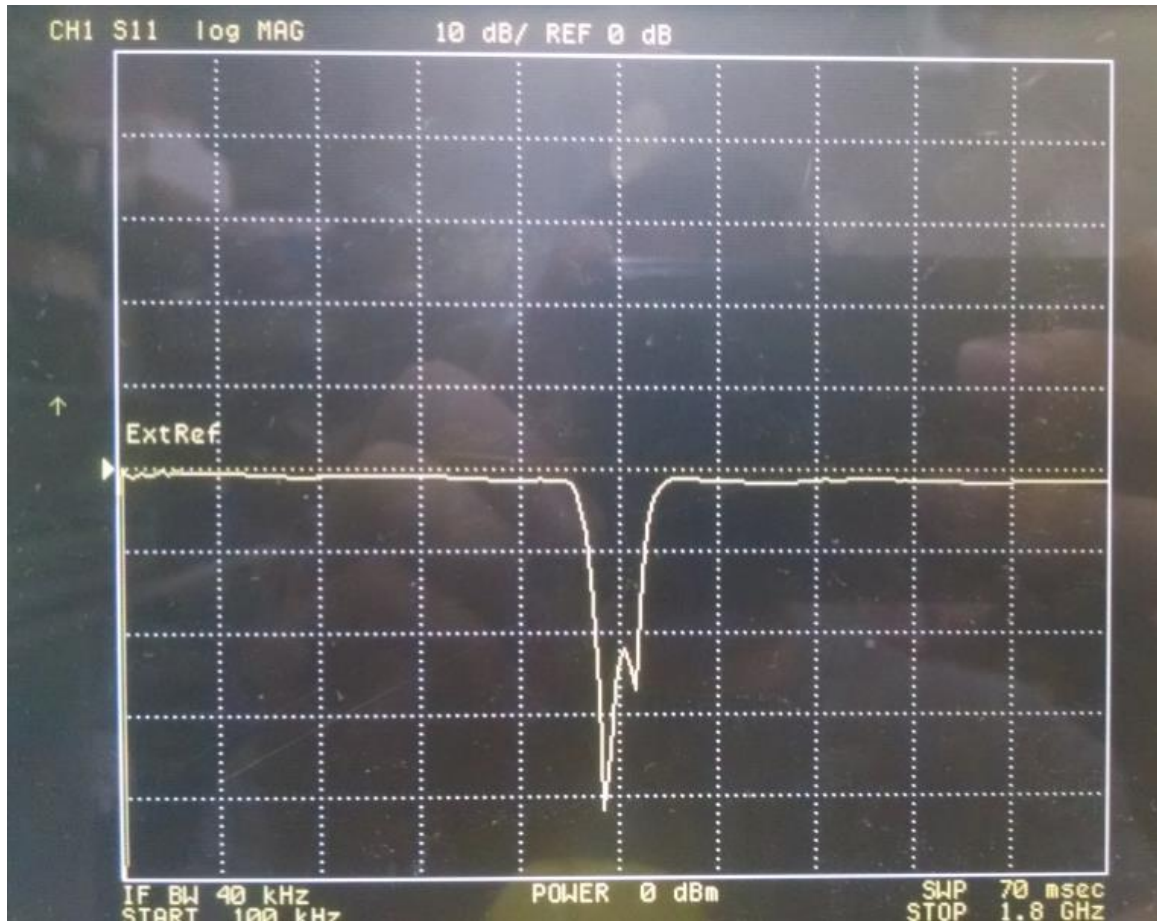


Figure 7.1-1: Frequency response

The final piece of equipment that will be beneficial to our hardware testing is an oscilloscope. The oscilloscope will be used to measure and record output signals from our sub systems. For example the oscilloscope can be used to characterize noise on subsystems. Figure 7.1-2 depicts the noise from a 5 volt offset. As shown the ripple (V_{pp}) is measured in decibels.

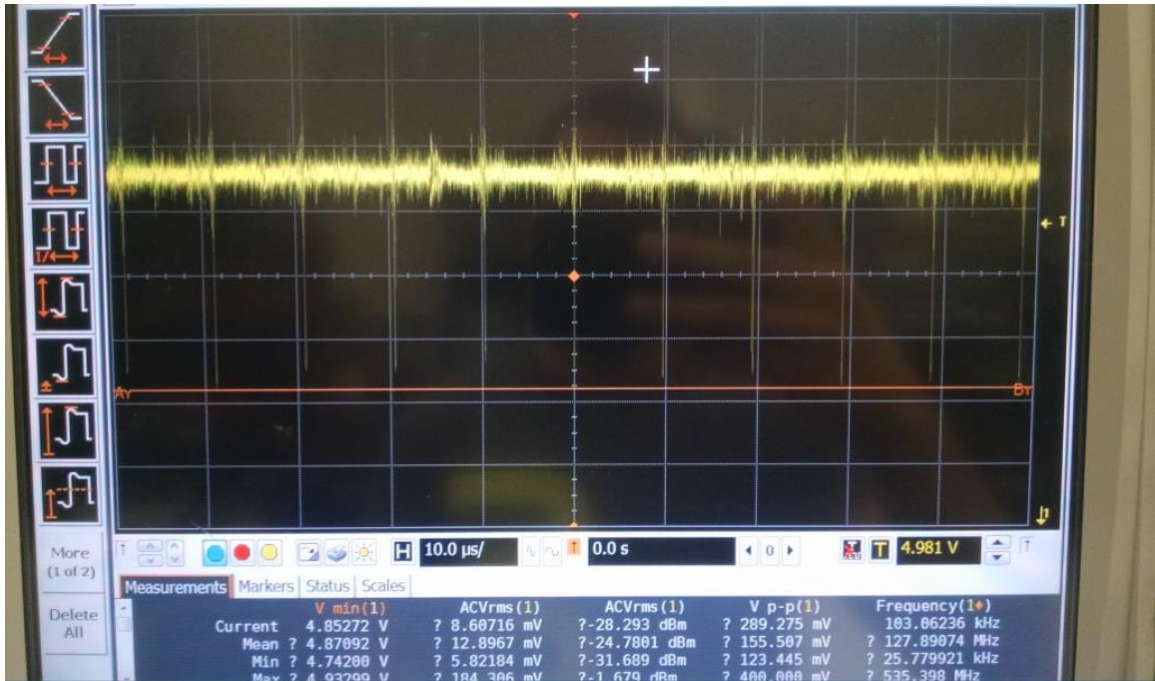


Figure 7.1-2: Noise on DC offset

Now that we have a proper list of equipment that is useful for characterizing our design, testing the hardware, and subsystem integration we can discuss briefly our testing locations. These laboratory locations will provide the equipment necessary for testing our design. Table 7.1-1 lists the laboratories and the equipment available.

Laboratory	Equipment
Senior Design Lab (Eng 1, Room 456)	Tektronix MSO 4034B Digital Mixed Signal
	Oscilloscope, 350 MHz, 4 Channel
	Tektronix AFG 3022 Dual Channel Arbitrary Function Generator, 25 MHz
	Tektronix DMM 4050 6 ½ Digit Precision Multimeter
	Agilent E3630A Triple Output DC Power Supply
	Dell Optiplex 960 Computer
EECS Smart Lab (HEC, Room 384)	Oscilloscope, 350 MHz, 4 Channel
	Tektronix AFG 3022 Dual Channel Arbitrary Function Generator, 25 MHz
	Tektronix DMM 4050 6 ½ Digit Precision Multimeter
	Agilent E3630A Triple Output DC Power Supply
	Dell Optiplex 990 Computer
CAAT Lab (Eng 1, Room 234, limited access to members of CAAT)	Spectrum Analyzer
	Sweep Generator
	Tektronix DPO4104B Oscilloscope
	RF power meter

Table 7.1-1: Testing equipment list

A list of the possible test can be shown in Table 7.1-2

Test	Equipment
Voltage Regulators variation	Agilent E3630A Triple Output DC Power Supply
	Tektronix DPO4104B Oscilloscope
MOSFET Switch responses	Oscilloscope, 350 MHz, 4 Channel
	Tektronix AFG 3022 Dual Channel Arbitrary Function Generator, 25 MHz
	Tektronix DMM 4050 6 ½ Digit Precision Multimeter
Filter Responses (If needed)	Spectrum Analyzer
	Sweep Generator
	Tektronix DPO4104B Oscilloscope
Magnetron Power Output	Tektronix AFG 3022 Dual Channel Arbitrary Function Generator, 25 MHz
	RF power meter

Table 7.1-2: Testing Overview

7.2 Hardware Specific Testing

This section enclosed specific test the design team will conduct during our prototyping stages.

Test Name:

Characterization of Memrowave's voltage regulators

Objective:

The objective of this experiment is to characterize any voltage regulators used in our system. The voltage regulator will be tested to determine if the output is the desired value. As well as testing to see if the output value is consistently where we would expect it to be. DC output may still vary by a percent and we want to know how often and how far off our output voltages may be.

Supplies:

- Tektronix DPO4104B Oscilloscope
- Agilent E3630A Triple Output DC Power Supply
- Tektronix DMM 4050 6 ½ Digit Precision Multimeter

Preparation:

1. The voltage regulator is removed from packaging and place on an antistatic pad.
2. The Tektronix DPO4104B Oscilloscope is powered up.
3. The Agilent E3630A Triple Output DC Power Supply is powered up.
4. The Tektronix DMM 4050 6 ½ Digit Precision Multimeter is powered up

Procedure:

1. Voltage regulator is bias with the required input voltage V_{in}
2. The regulator is manually adjusted till the required output voltage V_r is received using the Tektronix DMM 4050 6 ½ Digit Precision Multimeter
3. V_r is inputted to the Tektronix DPO4104B Oscilloscope
4. The Oscilloscope is set to display a histogram of V_r from port 1

Expected Result:

The histogram is expected to display a large frequency around V_r . If V_r is highly consistent the voltage regulator will be used. If the histogram is not frequent around V_r we will replace the voltage regulator.

Test Name:

Characterizing MOSFET switches

Objective:

The objective of this experiment is to characterize any MOSFET switches used in our system. The MOSFET switch will be tested to determine if the output is the desired value set. As well as testing to see if the output value is consistently where we would expect it to be. Each MOSFET may vary by a percent and we want to know how often and how much our output voltages vary.

Supplies:

- Tektronix MSO 4034B Digital Mixed Signal
- Tektronix AFG 3022 Dual Channel Arbitrary Function Generator, 25 MHz
- Oscilloscope, 350 MHz, 4 Channel
- Tektronix DMM 4050 6 ½ Digit Precision Multimeter
- Agilent E3630A Triple Output DC Power Supply

Preparation:

1. The MOSFET and resistors are removed from packaging and place out for assembly
2. The Tektronix AFG 3022 Dual Channel Arbitrary Function Generator is powered up.
3. The Agilent E3630A Triple Output DC Power Supply is powered up.
4. The Tektronix DMM 4050 6 ½ Digit Precision Multimeter is powered up

Procedure:

1. The MOSFET switch is constructed on a bread board and hooked up to equipment required to test
2. MOSFET switch is bias with the required input voltage V_{in}
3. V_{out} is inputted to the Oscilloscope, 350 MHz, 4 Channel
4. The Oscilloscope is set to display the time plot of the output voltage required

Expected Result:

The output is expected to display a switching nature. V_{out} is expected to be present when the MOSFET receives its required gate voltage to switch on. This will be observed via Oscilloscope.

Test Name:

Verifying motor controllers active turntable correctly

Objective:

The objective of this experiment is to check if the subsystems are integrated correctly. MOSFET switches used in our system as well a DC power supply and the turntable motor. The system will be tested to determine if integration was a success. Each subsystem will be tested in a similar manor.

Supplies:

- Tektronix MSO 4034B Digital Mixed Signal
- Tektronix AFG 3022 Dual Channel Arbitrary Function Generator, 25 MHz
- Oscilloscope, 350 MHz, 4 Channel
- Tektronix DMM 4050 6 ½ Digit Precision Multimeter
- Agilent E3630A Triple Output DC Power Supply

Preparation:

1. The motor is removed from packaging.
2. The Tektronix AFG 3022 Dual Channel Arbitrary Function Generator is powered up.
3. The Agilent E3630A Triple Output DC Power Supply is powered up.
4. The Tektronix DMM 4050 6 ½ Digit Precision Multimeter is powered up

Procedure:

1. The motor is connected to the switching circuit on the bread board.
2. MOSFET switch is bias with the required input voltage V_{in}
3. The output voltage will be monitored by the multimeter
4. The motor is observed during testing to verify correct outputs.

Expected Result:

The Motor is monitor visually by testers, once the MOSFET switching circuit is biased and verified to be correctly switching. The motor is expected to be powered during the changing of our switch. If the motor is nothing powered we can expect an unwanted voltage on the output of the switch.

Test Name:

Checking for proper webcam integration to BeagleBone.

Objective:

The objective of this experiment is to check if the subsystems are integrated correctly. The BeagleBone black will be connected to the webcam and powered to check for a connection. The system will be tested to determine if integration was a success. LCD monitors and Wi-Fi will be tested similarly.

Supplies:

- Logitech Webcam C270
- BeagleBone Black Dev board
- LCD screen
- Tektronix DMM 4050 6 ½ Digit Precision Multimeter
- Agilent E3630A Triple Output DC Power Supply

Preparation:

1. The Beagle bone placed in testing area.
2. The Board is removed from packaging and placed on antistatic pad.
3. Power supply is set to the required voltage to power the Dev board.
4. The multimeter is used to verify the correct value.

Procedure:

1. The BeagleBone and webcam are connected together.
2. The BeagleBone is powered up from the power supply.
3. Expected to receive input from the Logitech C270 webcam.
4. A design member will hold up a barcode to the camera.
5. A design member will verify that the product is in sight on the LCD screen.

Expected Result:

The outcome for the test is to verify that all the involved subsystem are communicating with each other after integration. We expect for the webcam to be displayed on the LCD screen.

Test Name:

Characterization of system noise floor

Objective:

The objective of this experiment is to characterize the noise floor of our system. The noise floor will be tested to determine the output power in dB caused by the thermal noise of the environment. Having information on the noise floor of our system will greatly improve our ability to predict outputs.

Supplies:

- Tektronix DPO4104B Oscilloscope
- Agilent E3630A Triple Output DC Power Supply
- Tektronix DMM 4050 6 ½ Digit Precision Multimeter

Preparation:

1. The system is not connected to any power supply
2. The Tektronix DPO4104B Oscilloscope is powered up.
3. The Agilent E3630A Triple Output DC Power Supply is powered up.
4. The Tektronix DMM 4050 6 ½ Digit Precision Multimeter is powered up
5. Isolation of the system from anything other than thermal noise.

Procedure:

1. The System is left off and connected to the DPO4104B Oscilloscope.
2. The Scope is set to measure in dB
3. The output of a given element is inputted to the Tektronix DPO4104B Oscilloscope
4. The value is measured and determined to be the noise floor of the system.

Expected Result:

The expectation of the experiment is to determine the noise caused by leaving the system open to energy being coupled in from the environment. This is what we can predict will caused our idealized values to be off by a certain percent. The thermal noise will affect the signal to noise ratio of the system.

Test Name:

Characterization of energy coupling from magnetron to controllers.

Objective:

The objective of this experiment is to characterize the noise energy coupled from our magnetron to our elements such as controllers. The energy coupled to controllers may affect the inputs of circuits thus changing values. This will be tested to determine the output power in dB caused by the energy coupled from the magnetron. Having information on the energy coupling of our system will greatly improve our ability to predict how much precautions we will need to take to shield components from the RF noise from the magnetron.

Supplies:

- Tektronix DPO4104B Oscilloscope
- Agilent E3630A Triple Output DC Power Supply
- Tektronix DMM 4050 6 ½ Digit Precision Multimeter

Preparation:

1. The system is not connected to any power supply
2. The Magnetron is connected to an isolated power supply
3. The Tektronix DPO4104B Oscilloscope is powered up.
4. The Agilent E3630A Triple Output DC Power Supply is powered up.
5. The Tektronix DMM 4050 6 ½ Digit Precision Multimeter is powered up
6. Apply shielding to elements as necessary.

Procedure:

1. The System is left off and connected to the DPO4104B Oscilloscope.
2. The Scope is set to measure in dB
3. The output of a given element is inputted to the Tektronix DPO4104B Oscilloscope
4. Power the Magnetron and observe if energy is coupled in the form of ripple in expected ideal outputs.
5. The value is measured and determined to be the noise floor of the system.

Expected Result:

The expectation of the experiment is to determine the noise caused by leaving the system open to energy being coupled in from the environment and the magnetron. This is what we can predict will caused our idealized values to be off by a certain percent and determine if we need to shield our elements. The thermal noise and magnetron will affect the signal to noise ratio of the system.

7.3 Software Test Environment

The software testing will be done in two separate environments. We will be designing unit tests for each class in the software. These tests will be run on the Beaglebone Black through the Android Unit Testing Framework. These tests will ensure that changes to the software do not break any feature of any of the classes. In section 7.3.1, we will be describing the unit tests for each class described in section 5.1.

In addition to unit testing, we will design end-to-end tests that will be used to test the functionality of the entire system. These tests will simulate most actions that a user might take, to be sure the software works consistently. The tests will describe a physical sequence of actions that must be performed on the Memrowave to test the software functionality. These tests will not be automated, so the environment will be different than the unit tests. More specifically, the test will be performed in the UCF senior design lab and the test procedures will be performed by a member of the group.

7.4 Software Specific Testing

In this section, we will outline the various tests we will use to verify the correct functionality of the software. In section 7.4.1, we will outline the unit tests that will be applied to each class in the Memrowave's software. In section 7.4.2, we will outline the end-to-end tests that will be performed by a group member to verify that all of the software subsystems are integrated properly.

7.4.1 Unit Tests

Each of the critical classes in the Memrowave's software must be tested to ensure that it is working properly. In the following sections, we will describe the unit tests that will be applied to each class.

The BarcodeScanner Class

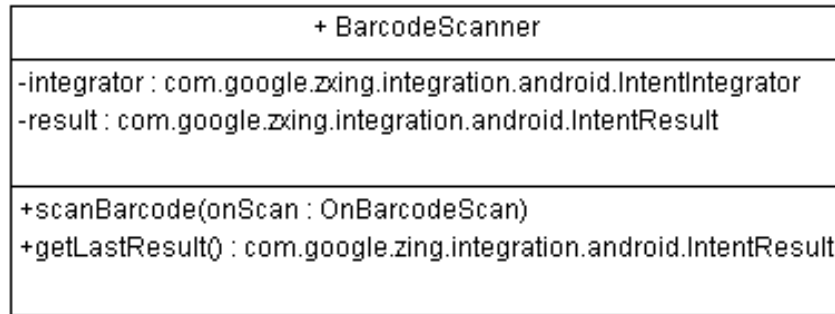


Figure 7.4.1-1: The BarcodeScanner class

This test will ensure that the BarcodeScanner class shown in Figure 7.4.1-1. To test the barcode scanner class, we will gather a list of barcode images and their respective UPC codes. The images will be stored in one array and the UPC codes will be stored in another. The test will go through each image in the array, use the BarcodeScanner class to read the barcode in the image, and then check whether or not the scanned UPC matches the expected value. If the class obtains the correct result for all of the barcodes, it passes the test.

The ProductDatabase Class

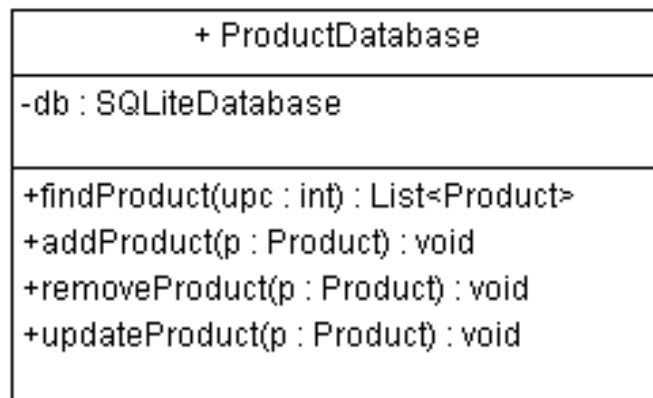


Figure 7.4.1-2: The ProductDatabase class

To test the ProductDatabase class shown in Figure 7.4.1-2, we will test each method of the class to ensure that they work as expected. We will supply the ProductDatabase class with a test SQLiteDatabase to do the operations on.

The first method listed in the class diagram shown in Figure 4.3.5-2 is the findProduct method. This method takes an integer UPC code as an argument

and returns a list of all the products in the database that have that UPC code. To test this method, we will populate the database with a number of example products and create a corresponding instance of the Product class for each. The test will then call the findProduct method for each of these products and ensure that each Product that is returned by the method matches the expected product.

The second method listed is the addProduct method. This method takes an instance of the Product class as an argument and adds it to the database. To test this method, we will create an example product instance and attempt to add it to the database using the addProduct method. The findProduct method will then be used to retrieve the entry. If the product retrieved from the database matches the product added to the database, then the test is passed successfully.

The next method is the removeProduct method. This method takes an instance of the Product class as an argument and removes this entry from the database, if it exists. To test this method, we will first add a test product to the database using the addProduct method, then we will verify that the product was added using the findProduct method, and finally, we will call the removeProduct method on this product. If a subsequent call to the findProduct method returns no results, then the product will be considered to be removed successfully.

The updateProduct method takes an instance of the Product class as an argument and changes the entry in the database with a matching id. To test this method, we will first add a test product to the database using the addProduct method. We will then modify the original Product instance and call the updateProduct method with the modified product as the argument. We will then call the findProduct method to get the product with the matching id from the database. If the Product instance returned by the findProduct method matches the modified Product instance, then the method works as expected.

The UserDataBase Class

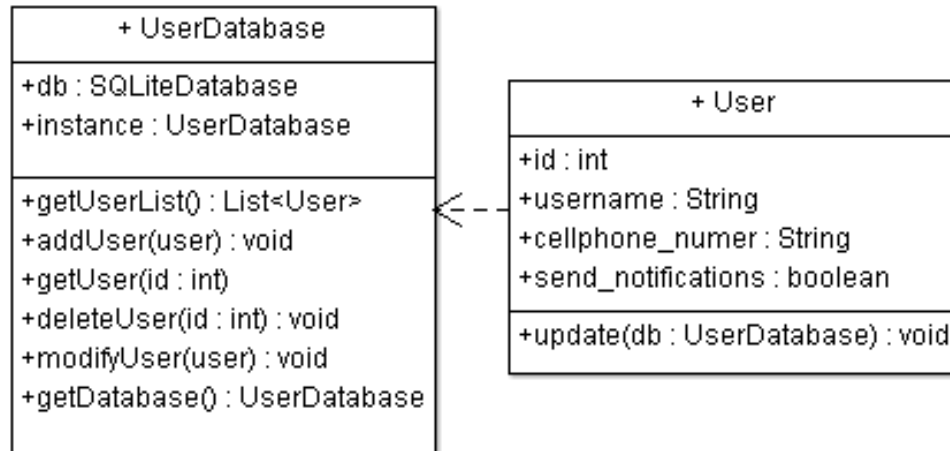


Figure 7.4.1-3: The UserDataBase class

The UserDataBase class is shown in Figure 7.4.1-3. To test this class, we will first create an empty SQLite database. After each test, this database will be erased so that any subsequent tests are conducted from a blank state.

The addUser method takes an instance of the User class as an argument and adds this user to the database. To test this method, we will create a number of test users and add them to the database using this method. We will then use SQL operations on the database to verify that the test users were added correctly.

The getUserList method takes no arguments and returns a list of all the users in the database. To test this method, we will first call the method on a blank database and verify that the method returns an empty list. We will then add a number of test users to the database using the addUser method. Finally, we will call the getUserList method and verify that the resulting list of users matches the original list.

The getUser method takes an integer ID as an argument and returns the corresponding user from the database. To test this method, we will add a test user to the database using the addUser method. We will then call the getUser method with the appropriate id as the argument. Finally, we will verify that the user retrieved by the getUser method matches the original test user.

The deleteUser method takes an integer ID as an argument and deletes the corresponding entry from the database. To test this method, we will add a test user to the database using the addUser method, then we will verify that the user was added correctly using the getUser method. We will then call the deleteUser

method to delete the user from the database. The delete operation will be verified by calling the `getUser` method again and verifying that no user is returned.

The SMSNotifier Class

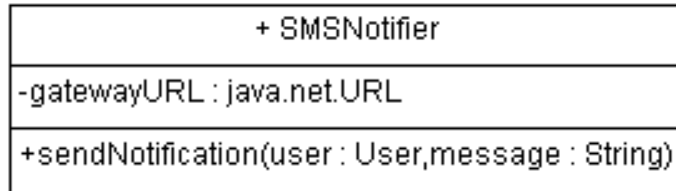


Figure 7.4.1-4: The SMSNotifier class

To test the SMSNotifier class shown in Figure 7.4.1-4, we will create a test user with one of the group member's cell phone number. We will then send a test message to this user and verify that the message was sent to the phone properly. We will then disable notifications in the user's settings and attempt to send the message again. In this case, we will verify that the message was not sent.

7.4.2 End-to-end Tests

Unit testing is helpful to ensure that all of the subsystems work properly, but it is difficult to verify that all of the subsystems will work together properly using unit testing. To test the integration of all of the software subsystems, we will design end-to-end tests that will be performed by a group member.

Test Name:

Main Screen Test

Objective:

The objective of this test will be to verify that the main screen of the Memrowave's software works correctly.

Supplies:

- Memrowave

Preparation:

1. Ensure that the Memrowave hardware is assembled and working properly
2. Press the home button to bring up the main screen of the Memrowave

Procedure:

1. Press the 'Settings' button and verify that the main settings screen is displayed. Press the back button and verify that the Memrowave returns to the main screen.
2. Press the 'Scan' button and verify that the barcode scanning screen is displayed. Press the back button and verify that the Memrowave returns to the main screen.
3. Press the 'Manual Cook' button and verify that the manual operation screen is displayed. Press the back button and verify that the Memrowave returns to the main screen.
4. Repeat steps 1 through 3, pressing the home button to return to the main screen instead of the back button. Verify that the Memrowave returns to the main screen for each.

Test Name:

Manual Operation Screen Test

Objective:

The objective of this test is to verify that the manual operation screen works correctly.

Supplies:

1. Memrowave

Preparation:

1. Ensure that the Memrowave hardware is assembled and working properly
2. Press the home button to bring up the main screen of the Memrowave

Procedure:

1. Press the 'Manual Cook' button on the main screen.
2. On the manual cook screen, press the 'Quick Cook' button. Verify that the quick cook screen is displayed, and then press the back button. Verify that the manual cook screen is displayed again.
3. Enter a cook time on the number pad. Verify that the entered time is shown on the timer display.
4. Press the 'Power' button and verify that the power level menu is shown. Enter a power level and verify that the software returns to the manual cook screen with the appropriate power level shown.
5. Press the clear button and verify that the entered cook time is reset to zero.
6. Enter another cook time and press the start button. Verify that the Memrowave begins to cook and counts down the time properly. When the timer reaches zero, verify that the Memrowave stops cooking and produces the appropriate notifications.

Test Name:

Product Scan Screen Test

Objective:

This test is designed to verify that the product scanning screen works correctly.

Supplies:

1. Memrowave

Preparation:

1. Ensure that the Memrowave hardware is assembled and working properly.
2. Insert a new test product into the local product database.
3. Press the home button to bring up the main screen of the Memrowave.

Procedure:

1. Press the 'Scan' button on the main screen. Verify that the product scan screen is displayed.
2. Verify that the camera preview is shown on the screen.
3. Hold the barcode for the test product up to the camera so that the entire barcode is shown on the preview. Press the 'Scan Now' button and verify that the barcode is scanned properly and that the product details screen is shown.
4. Press the back button and verify that the product scan screen is displayed again.

Test Name:

Product Cook Test

Objective:

This test is designed to verify that the Memrowave can scan and cook a product automatically.

Supplies:

1. Memrowave

Preparation:

1. Ensure that the Memrowave hardware is assembled and working properly.
2. Insert a new test product into the local product database.
3. Press the home button to bring up the main screen of the Memrowave.

Procedure:

1. Press the 'Scan' button on the main screen. Verify that the product scan screen is displayed.
2. Verify that the camera preview is shown on the screen.
3. Hold the barcode for the test product up to the camera so that the entire barcode is shown on the preview. Press the 'Scan Now' button and verify that the barcode is scanned properly and that the product details screen is shown.
4. Verify that the correct cook time and power level are shown for the test product.
5. Put the product into the Memrowave and press the 'Start' button. Verify that the Memrowave begins to cook the food. When the timer reaches zero, verify that the Memrowave stops cooking and sends the appropriate notifications.

Test Name:

Quick Cook Test

Objective:

This test is designed to verify that the Quick Cook feature works correctly.

Supplies:

1. Memrowave

Preparation:

1. Ensure that the Memrowave hardware is assembled and working properly.
2. Verify that the Quick Cook database has been populated with example products.
3. Press the home button to bring up the main screen of the Memrowave.

Procedure:

1. On the main screen, press the 'Manual Cook' button, then press the 'Quick Cook' button. Verify that the 'Quick Cook' screen is displayed.
2. Verify that the Quick Cook products inserted in the Preparation section are displayed in the product list.
3. Select one of the quick cook products and verify that the product cook screen is displayed with the correct information.
4. Press the 'Back' button to return to the Quick Cook screen.
5. Press the 'Add New' button. Verify that the 'New Quick Cook Product' screen is displayed. Enter the name, cook time, and power level, and press the save button.
6. Press the 'Home' button to return to the main screen, and then navigate to the 'Quick Cook' screen. Verify that the new Quick Cook item is shown in the list.

Test Name:

User Profile Test

Objective:

This test is designed to verify that the User Profiles feature works correctly.

Supplies:

1. Memrowave

Preparation:

1. Ensure that the Memrowave hardware is assembled and working properly.
2. Press the home button to bring up the main screen of the Memrowave.

Procedure:

1. Press the 'Settings' button on the main screen. Verify that the settings screen is displayed.
2. Press the 'New Profile' button and enter the information for the new profile, including the cellphone number for SMS notifications. When finished, press the save button.
3. Return to the Settings page. Press the 'Change User' button. Verify that the user list is shown, and that the user created in step 2 is shown.
4. Select the user created in step 2 and proceed to cook a test product in the Memrowave. Verify that an SMS notification is received when the cooking finishes.

8 Administrative Content

In this section we will discuss the administrative content of the project, including project milestones and budget considerations.

8.1 Milestone Discussion

The milestone discussion details about who the Memrowave project will move from the initial conceptual stage to the final product stage will be outlined with important key sections to keep track of our progress as well as when it is necessary to move forward in the face of any delays or setbacks. A rough outline of what needs to be done to reach the final end product is shown below.

- I. Conceptual Stage
- II. Design Stage
- III. Part and Component Collection
- IV. Initial Assembly and Testing
- V. Redesign
- VI. Final Assembly and Testing

Within each major section, division between software and hardware must be outlined. The completion of the first areas will be finished with the initial writing of this document. Decisions on specific parts and components have been decided and detailed in their corresponding sections while others have been specified without exact part information. For the part and component collection, all parts will be collected based on the importance and how extensive any further development on the item is necessary such as the processing unit and LCD touchscreen which will need continuous development throughout the project to make updates, changes, and to find and fix any bugs. On the hardware side, crucial components such as the magnetron, power supply, and voltage regulators will be needed as soon as possible to make sure such parts will fit into the overall assembly and scheme of the Memrowave. Initial assembly and testing will be the most extensive and time-consuming sections of the overall project. After some development by both hardware and software sections of the Memrowave, we will start the integration of both divisions. Not particularly into an enclosure, but a checking of the overall system's abilities to do certain functions and the robustness. A list of the specific checks will be done:

- Software functionality
- Power supply working properly
- Electrical component check
- LCD touchscreen functionality
- Processing unit operation
- Switch tests
- Camera operation
- WIFI operation
- Notification operation
- Total power consumption

Further tests will be done after assembling into the microwave enclosure takes place. After the initial round of testing any changes, errors, or potential delays will be addressed. Based on the available time and options decisions will be made on whether to fix certain issues, go alternate design routes, or to abandon certain ideas to move the project forward. Final design conclusions will be drawn in the following categories and will dictate how to progress:

- Successes
- Failures
- Additions
- Withdrawals

Once final design decisions have been made, a final assembly and development period will ensue to create the final product before the end of the time frame. At this point both hardware and software have been fully developed and integration issues and fine tuning will be the only issues left to address. Below is an outline of the major milestones with a rough time line on the completion of these tasks.

I. Memrowave

A. Hardware

1. Part Collection
 - a) Printed Circuit Board Design
 - b) DC Power Supply
 - c) Voltage Regulators
 - d) Power Switch
 - e) Internal Lights
 - f) Internal Beeper
 - g) Turntable Motor
 - h) Magnetron
2. Part Testing and Characterization
 - a) Functionality
 - b) Usability
3. Initial System Development
 - a) PCB fabrication and assembly
 - b) Proper operation of power devices
 - c) Power specifications are within design
 - d) Robustness of internal connections

B. Software

1. Part Collection
 - a) Processing Unit
 - b) LCD touchscreen
 - c) WIFI Module
 - d) Camera Module
2. Part Testing and Characterization
 - a) Functionality
 - b) Usability
3. Initial System Development
 - a) Integration between Processing Unit and LCD touchscreen
 - b) Integration between Processing Unit and WIFI Module
 - c) Integration between Processing Unit and Camera Module

C. Hardware/Software

1. Initial integration of Hardware and Software
 - a) Proper function of the power switch
 - b) Proper operation of software part collection under power supply
 - c) Desired control of internal lights by processing unit
 - d) Desired control of internal beeper by processing unit
 - e) Desired control of magnetron by processing unit
 - f) Desired control of turntable motor by processing unit
 - g) Available space to fit components into microwave enclosure
2. Redesign
 - a) Successes
 - b) Failures
 - c) Additions
 - d) Withdrawals
3. Final Integration of Hardware and Software
 - a) Proper operation of any alternative functions or processes
 - b) Proper function of the power switch
 - c) Proper operation of software part collection under power supply
 - d) Desired control of internal lights by processing unit
 - e) Desired control of internal beeper by processing unit
 - f) Desired control of magnetron by processing unit
 - g) Desired control of turntable motor by processing unit
 - h) Available space to fit components into microwave enclosure
 - i) Cosmetic fine tuning
 - j) Proper function of the user interface
 - k) Overall functionality of the Memrowave

8.2 Finance and Budget Discussion

The Memrowave development is only possible with a strong financial backing. There were multiple options to obtain financing for our project. The first option was to seek financial backing from companies that would like to sponsor our design. Our largest sponsor options were Duke Energy and Boeing. These companies requested documentation contain information on project summary, applications, and financing requests. Using these sponsors however limited our group to only receiving reimbursements for our money invested. The Memrowave design team decided on financing the project fully out of pocket. The reasoning for this is due to the fact that the Memrowave overall cost was deemed to be low enough for self-purchase, the design didn't fit the needs of Boeing, and our team did not want to risk signing over parts endorsed by sponsors. Table 8.2-1 depicts sponsors available and if requested.

Sponsor	Request
Boeing	No Request
Duke Energy	No longer sponsors
Soar Tech	No Request
Camera Stability	No Request

Table 8.2-1: Sponsors

Finance will then be handled completely internally by group members; Joseph Serritella, Darren Armstrong, Jack Gulick, and Winston Todd. Finance might also not be split evenly by a divisor of 4. Some members maybe willing or can afford a bigger contribution towards the total cost of the Memrowave. The Benefits of out of pocket financing are; Parts can be ship to a destination of our choosing, Project will be able to remain in our control afterwards, and dealing with sponsor will not be an issue. Because our project is personally financed parts are not required to be shipped to a specific location. The benefit of this is found in personal leisure time. The Memrowave design team will be able to acquire parts at any time and have them shipped to a location of our choosing. For example our project will not be limited to being stored with other group design parts. The next big influence on personal financing is that we will be allowed without question to have our prototype remain in our control. For example if the Memrowave received one hundred percent financing from Boeing incorporated our design would be owned by that company. Likewise if only a portion of our design was sponsored by a company that company may be entitled to that said portion. This however is not a guarantee, a sponsored project may be deemed unsalvageable hence would remain in the design team’s possession. This is not an option out team would like to entertain because it is in our best interests to retain the Memrowave. Lastly because we are personally financing the Memrowave we will not have to adhere to a sponsor’s requests. For example, one sponsor requested a self-correcting camera with full finical sponsor. This required a team to follow the project goals of the sponsor. It is in the best interest of the Memrowave to remain independent of a sponsor due to the possibility of being required to follow request not in the best interest of the Memrowave and the Memrowave design group.

The Memrowave’s budget will consist of parts, LCD screens, Wi-Fi, printed circuit boards, and more along these lines. Our design which can be heavily influenced by the microwave we purchase may be cheap or expensive. For instance if our design requires us to remove most parts from the microwave we will end up spending more on replacing the parts. However if our design allows us to reuse the original parts such as motors and fans, our budget will run lower. The budget calculated in this section will cover replacing motors, the magnetron and others. This will give our budget a worst case scenario.

The budget will include the medium price listing for cameras, Wi-Fi Chips, development boards, Touch screen panels, Microcontrollers, resistors, transistors, and so on. To save time in broken parts and PCB we will also be ordering from vendors that will supply more than one board upon purchase. The Best PCB manufacturer for this will be OCH Park because they supply 3 copies of the printed circuit board with the purchase.

The Following Table 8.2-2 provides information on our budget.

Development Board	\$70
Microcontrollers (x4)	\$40
Logitech Webcam C270	\$40
Microwave	\$200
Microwave Turntable replacement	\$25
Turntable motor replacement	\$20
Microwave fan replacement	\$30
Fan motor replacement	\$30
Transformer DC/AC	\$10
Voltage Regulators	\$20
Ringer	\$5
Lights	\$5
Wi-Fi Card	\$40
PCB fab	\$60
Magnetron replacement	\$60
Power cord	\$5
LCD screen	\$100
TOTAL:	\$760

Table 8.2-2: Budget

Table 8.2-2 provides the layout for the possible budget we will encounter during our design. This budget was calculated to be seven hundred and sixty dollars. Figure 8.2-1 provides a bar graph to help depict this budget visually.

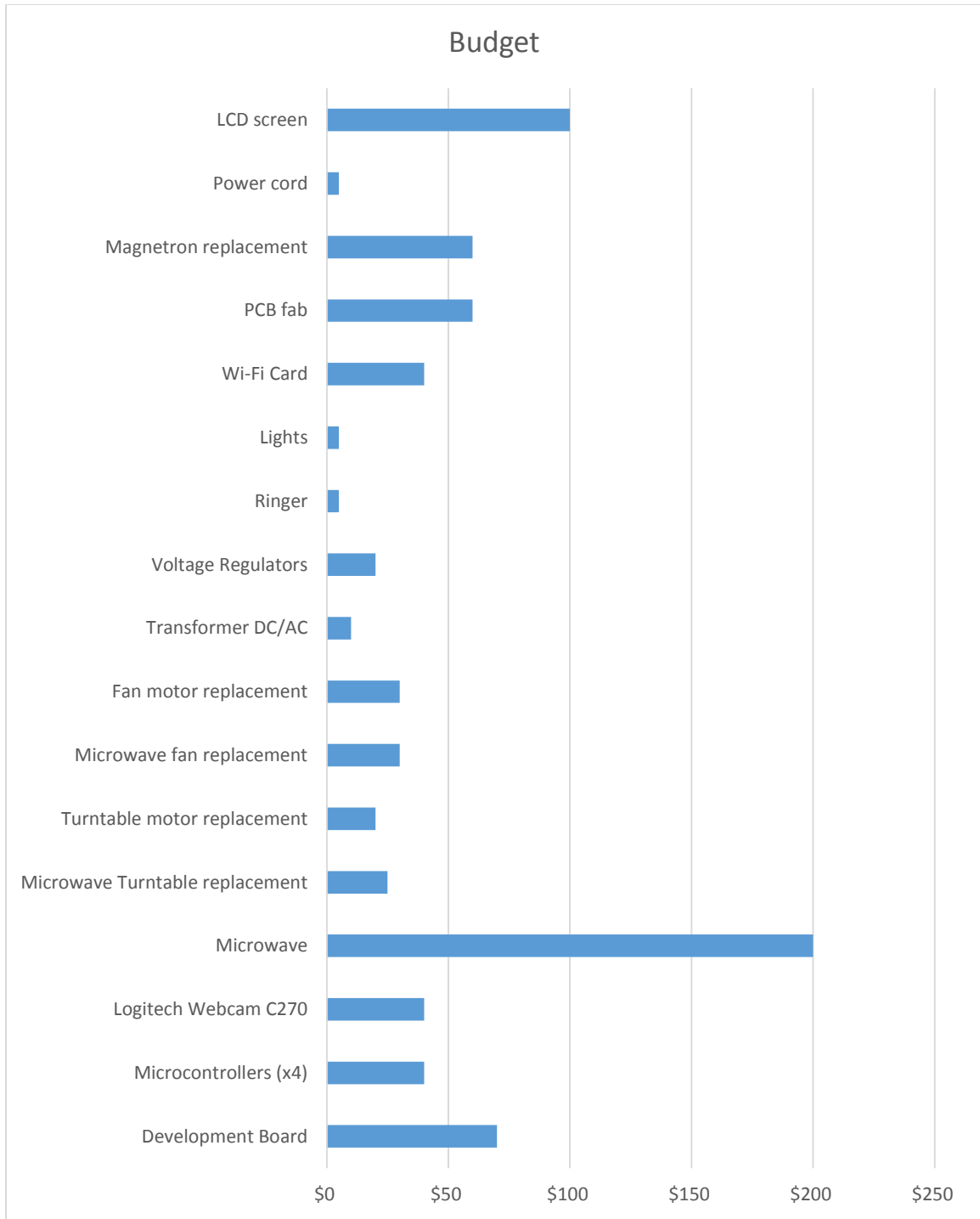


Figure 8.2-1: Budget bar graph

To better depict what cost the most in our budget Figure 8.2-2 shows a pie chart of the budget.

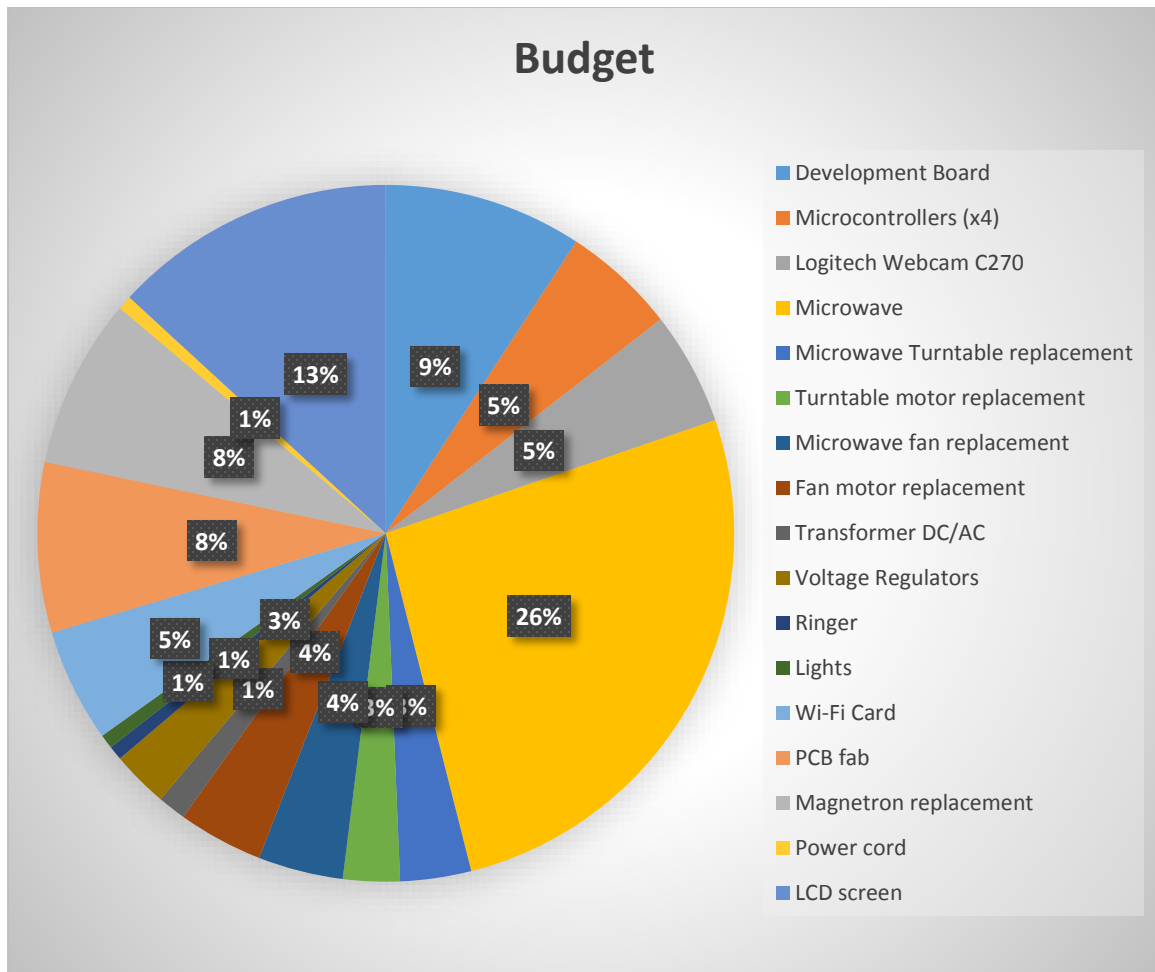


Figure 8.2-2: Pie chart of Budget

As the chart show the largest portion of our budget will be devoted to purchasing a microwave to use. However if we are able to use a pre-purchased microwave (which is 26% of our budget) our largest contributor will be the LCD screen (which is 13% of our budget). Based on our budget and financing plan the Memrowave will be devoting one hundred percent of our own resources into creating the Memrowave. However this allows us to work at our own pace without constraints from a sponsor. It also allows the design team to keep the project upon completion. The total Budget is also expected to go under budget because our team will likely acquire a pre-purchased microwave that has a working magnetron, turntable, fan, and other standard microwave accessories. This will bring our final contribution down to three hundred and seventy dollars. Therefore we will be three hundred and ninety dollars under budget.

Appendix A – Copyright Permissions

“Apache License, Version 2.0.” Internet:

<http://www.apache.org/licenses/LICENSE-2.0> [12/1/2014]

“GNU GENERAL PUBLIC LICENSE.” Internet: <http://www.gnu.org/licenses/gpl-3.0.html> [12/1/2014].

“Texas Instruments Terms of Use.” Internet:

<http://www.ti.com/corp/docs/legal/termsofuse.shtml> [12/1/2014].

Images from Sparkfun Electronics are licensed under the CC BY-NC-SA 3.0 license. Internet: <http://creativecommons.org/licenses/by-nc-sa/3.0/> [12/1/2014].

Appendix B – Datasheets

“MAX7219CNG 8-Digit LED Display Driver.” Internet:

<https://www.sparkfun.com/datasheets/Components/General/COM-09622-MAX7219-MAX7221.pdf> [12/1/2014].

“7-Segment Display - 20mm (Blue)” Internet:

<http://dlnmh9ip6v2uc.cloudfront.net/datasheets/Components/LED/1LEDBLUC C.pdf> [12/1/2014].

“MSP430G2553 Datasheet.” Internet:

<http://www.ti.com/lit/ds/symlink/msp430g2553.pdf> [12/1/2014].

“BeagleBone Black Reference Manual RevC.1.” Internet:

https://github.com/CircuitCo/BeagleBone-Black/blob/master/BBB_SRM.pdf?raw=true [12/1/2014].

“TI AM335X Sitara Processor.” Internet:

<http://www.ti.com/lit/ds/symlink/am3359.pdf> [12/1/2014].

Appendix C – References

“Android API Package Index.” Internet:

<http://developer.android.com/reference/packages.html> [11/28/2014].

“Android API Guides.” Internet: <http://developer.android.com/guide/index.html> [11/28/2014].

"Play SMS: Free and Open Source SMS Software." Internet: <http://playsms.org> [12/2/2014].

"ZXing Barcode Software Library." Internet: <https://github.com/zxing/zxing/> [12/2/2014].

"OLED." Internet: <http://en.wikipedia.org/wiki/OLED> [12/1/2014].

"Liquid Crystal Display." Internet: http://en.wikipedia.org/wiki/Liquid-crystal_display [12/1/2014].

"Touchscreen." Internet: <http://en.wikipedia.org/wiki/Touchscreen> [12/1/2014].

"Wi-Fi." Internet: <http://en.wikipedia.org/wiki/Wi-Fi> [12/1/2014].

"Raspberry Pi Frequent Asked Questions." Internet: <http://www.raspberrypi.org/help/faqs/> [12/1/2014].

"Raspberry Pi Wiki." Internet: http://en.wikipedia.org/wiki/Raspberry_Pi [12/1/2014].

"Beagleboard:BeagleBoneBlack Wiki." Internet: <http://elinux.org/Beagleboard:BeagleBoneBlack> [12/1/2014].

"BeagleBoard." Internet: <http://en.wikipedia.org/wiki/BeagleBoard> [12/1/2014].

"PandaBoard." Internet: <http://www.omappedia.com/wiki/PandaBoard> [12/1/2014].

"PandaBoard." Internet: <http://en.wikipedia.org/wiki/PandaBoard> [12/1/2014].

"PWM type." Internet: <http://sinhardware.com/index.php/vrm-articles/113-digital-vs-analog-pwms/74-analog-vs-digital-pwms> [12/1/2014].

"Timers." Internet: <http://homepages.ius.edu/RWISMAN/C335/HTML/msp430Timer.HTM> [12/1/2014].

"Coding PWM." Internet: <http://www.egr.msu.edu/classes/ece480/capstone/fall10/group04/Dong,%20Roy%20-%20Application%20Note.pdf> [12/1/2014].

"Stepper Motors." Internet: http://www.pcbheaven.com/wikipages/How_Stepper_Motors_Work/ [12/1/2014].

“Brushless motors” Internet: <http://electronics.howstuffworks.com/brushless-motor.htm> [12/1/2014].

“Brush motors” Internet: <http://electronics.howstuffworks.com/brush-motor.htm> [12/1/2014].

“Switching circuits” Internet: http://www.electronicstutorials.ws/transistor/tran_7.html [12/1/2014].

“Brushless motors” Internet: <http://electronics.howstuffworks.com/brushless-motor.htm> [12/1/2014].