Liquid Detection Cup

By
Alfredo Sanchez (alfredo7)
Francis Mui (fmui2)
Ran Wang (ranwang2)

Final Report for ECE 445, Senior Design, Spring 2020
TA: Ruhao Xia

8, May 2020

Team No. 72

Abstract

We want to eliminate unnecessary hassle for waiters by telling the waiters that a cup at any table they are serving is running low. The original solution was a weight sensor within the coaster, which relays information to the server when the weight goes below a threshold. Our solution is an optical sensor within the cup, which relays information when the water level goes under a threshold. The key difference is that the location of our system allows for a much different sensor, which eliminates issues with different weights and solids, and provide real-time updates on whether or not the
cup is running low.

1. Second Project Motivation

1.1. Problem Statement

The objective of our system is to be able to detect that a cup is running low on water, and convey this information to a nearby server. Places such as Red Robin offer free refills on their bottomless drinks [1], meaning multiple different cups are necessary for multiple different drinks. Red Robin's current solution is to flag down a waiter and tell them what to refill it with so that the waiter can comply.

1.2. Solution

Our solution is to implement a sensor directly onto the cup which will detect the water level of the cup. When the water threshold falls below 25%, a multicolor LED will light up to tell the waiter of the specific location of the cup, and a message will be sent directly to the serving station so an idle waiter can immediately fill their cup.

The previous project used a weight sensor within the coaster to detect when drinks run low. They use a load sensor to detect the amount of beer in a glass while the beer is sitting on a coaster, as well as LED lights placed on the coaster to signal to a server when a beverage needs to be refilled. This data is also sent to a computer system, which in their case is an iPad, which records the beer consumption rate and also sends the information to the waiter.

Our project intends to place the entire system as an add on to standard cups instead of the coasters. By moving the system to the cups, we can also switch from load sensors to optical sensors. Outside of this key difference, we follow the original project pretty similarly. We send any information from our system to a computer at a serving station, which is a computer system in our case, which will record the consumption rate and notify any waiters.

Our change allows the cup to effectively ignore any solids in the cup that impact its weight, such as bubble tea, or ice cubes. Furthermore, different liquids have different densities, which may lead to incorrect readings from a sensor, as a drink like grenadine is ~18% more dense than water and vodka is ~8.5% less dense than water. Because of this, we can use these cups and differentiate the drinks via color coding in the LEDs while still using the same sensor regardless of drink.

1.3. High-Level Requirements

1.3.1. Accuracy

The system needs to be able to respond to a cup running low on water with a 99% accuracy rate, without incurring more than a 5% false alarm rate, such as when a person is actively drinking from the water or swishing it around.

1.3.2. Size & weight

The system band on the cup will not be thicker than 5mm. The system's weight also will have to be smaller than 15% of the full glass. This requirement is for the customer comfortness.

1.3.3. Longevity

The battery life should be at least 2 hours when fully charged. The average current in 1 year must be 200uA.

1.4. Visual Aid

Figure 1 shows that when the water level goes down, a notification will be sent and light will turn on

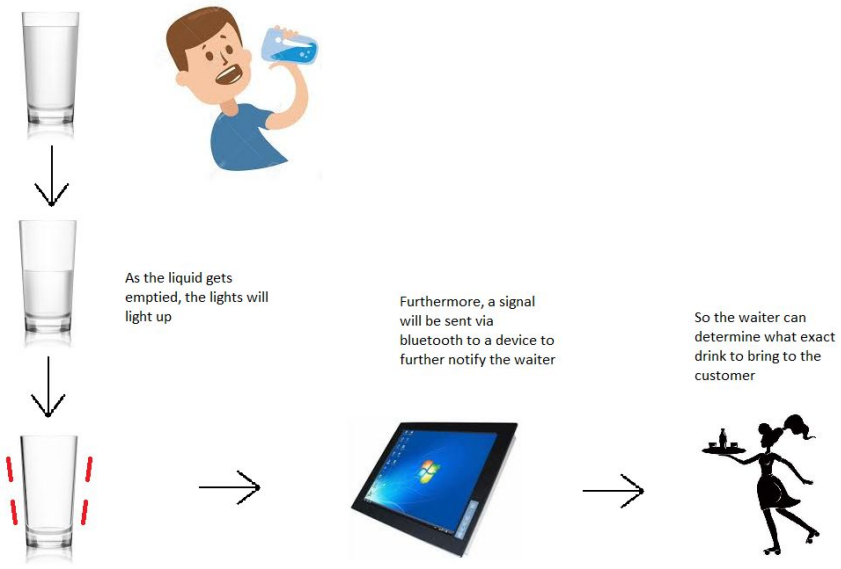


Fig.1 System Functionality

1.5. Block Diagram

Figure 2 shows how the detection module sends info into the control module, which sends it to both the bluetooth and light module, which finally sends information to the UI.

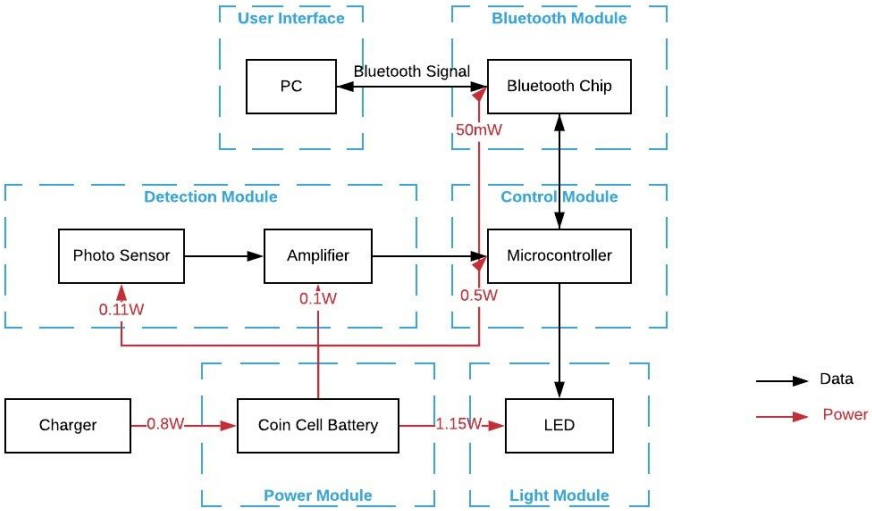


Fig.2 Block Diagram

2. Second Project Implementation

2.1. Implementation Details and Analysis

2.1.1. Detection Module

This module contains two optical level sensors embedded at opposite sides of the cup to account for the cup's tilting and accurately detect the drink level, and an amplifier with a gain of 2 to amplify the sensor's 0.5V/1V logical output to a 2.0V for later processing.

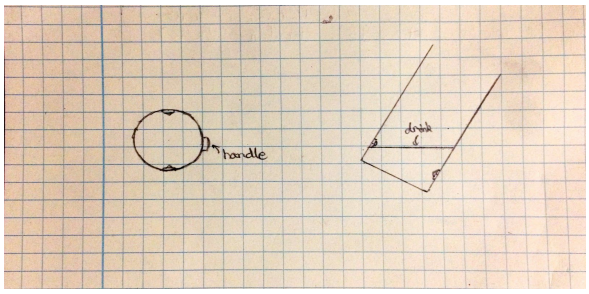


Fig.3 Double Sensor Detection

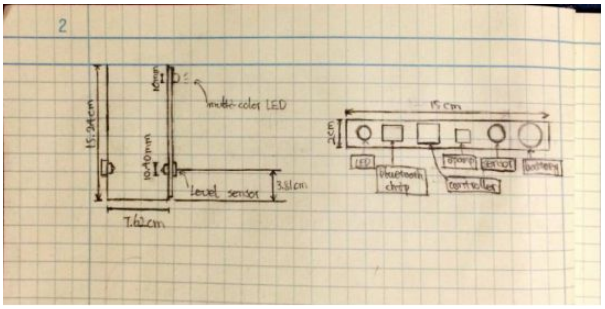


Fig.4 System Setup

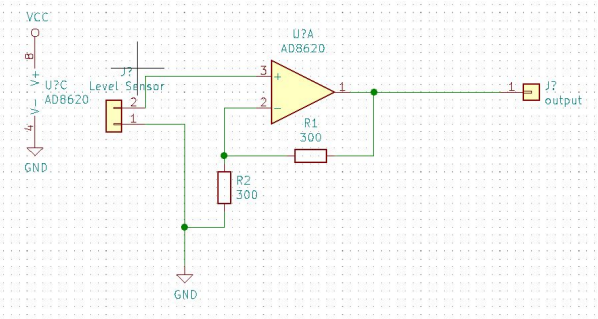


Fig.5 Detection Module Schematic

Sensor Choice Justification

In our original design for the liquid detection cup, we planned to implement a double sensor system for the detection module using the OLS200D3SH optical sensor.[2][3] The consideration was that with two sensors, the cup's tilting when the customer drinks from it will be better accounted for, and would not send out false alarms. Also, the sensor's system is relatively small. It would not take much space inside the cup, and it would be not so obvious for the user to notice, which means a better user experience.

However, with further discussion, some problems arise. The first is the price. An OLS200D3SH sensor would cost 40.67 dollars. With a double sensor system implemented, it would mean 81.34 dollars used purely for one part of the system.

Another problem is that once installed, the system will not be so easily removed. When the cup needs washing, for example, the sensors, along with the other parts of the system, need to be removed to avoid being damaged. However, since the sensors need to be installed inside the cup, two holes need to be drilled, and a secure installation is required to keep the sensors in place. To compensate for that, a possible solution is that we design the system to be flexible. For example, we can create spiral flutes on the cup. When needed, the sensors can be spiraled in, and secured by nuts. Another potential solution is a hand wash labor system. When washing/wiping the cups, extra caution is needed to avoid damaging the system. However, these two potential solutions both require extra labor, and would mean additional costs to the customer.

The third problem arises from the system's functionality. The sensor consists of an IR transmitter and a phototransistor. When used, the IR transmitter sends out signals that reflect inside the plastic cone of the sensor and are received by the transistor. The functionality is based on the refractive index difference between the plastic and the outer medium. With drinks of small particles, like water, this difference will be small, meaning it's easily recognized when the system is submerged. But with drinks of large particles, like milk, the difference will be greater, and closer to the difference between the plastic and the air. In this case, the system might not be able to accurately detect if it's submerged or exposed. But in reality, the refractive indices of different drinks are very similar, and will not create such drastic inaccuracies. Another issue is temperature. Hotter objects emit more IR rays than colder objects. The temperature of the drinks served ranges from 4 to 60 degrees Celsius, which means different levels of IR emission. The extra IR emissions that go into the sensor when submerged might result in an inaccurate reading. However, this will not mean such great differences, either. The IR emissions from the environment have far less power than the IR emitter embedded in the sensor. Also, the standard working temperature of the sensor, as stated in the datasheet, ranges

from -25 to +80 degrees Celsius, indicating the proper functionality under the desired conditions. As a solution to both issues, we can program the controller to account for the minor inaccuracies. That being said, we will still need to set up experiments to examine the influences these variables have on the system's functionality.

An alternate choice to the optical sensor would be a capacitive sensor, which contains a pair of inductive plates, shown in figure 6 below. When the sensor is fully exposed in the air, it has a certain capacitance. When it's partly exposed in a liquid, the capacitance varies due to the liquid being a dielectric material. Compared to the optical sensor system, this sensor has certain advantages in that it can be designed to be highly flexible, and that it does not depend on the properties of the liquid so much. In the design, a capacitor tube can be mounted at the outer wall of the cup, and a small hole is drilled on the bottom of the cup to connect the liquid in the cup to the tube. This design doesn't involve the sensor going directly inside the cup, so it can be removed relatively easily. Also, it does not depend on the size of particles of the drink, or the temperature of the drink, since it only depends on the dielectric property of the liquid. However, this choice has its own weaknesses.

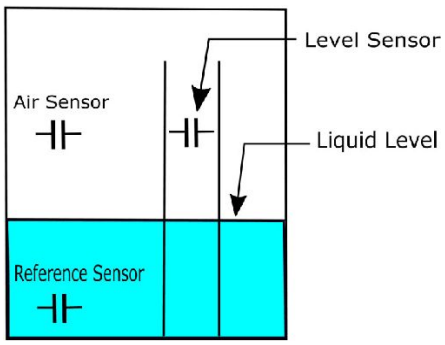


Fig.6 Capacitive Level Sensor[4]

The first is that it cannot account for the tilting of the cups as well as the double optical sensor system. In the original design, the system will send out alarms only when both of the sensors are exposed. This way, when the customer tilts the cup to drink, if only one sensor is exposed, the system will not send out alarms, because it's highly likely that the drink's level is still not low. With the capacitive sensor, on the other hand, the data output from the sensor will be quite unstable, reacting to all the disturbances. A potential solution to this is to design a timer in the control module. For example, when the drink level in the sensor drops below the 25% threshold for more than 20 seconds, it sends out signals. However, this solution will depend on how quickly the customer finishes the drink after the threshold is reached. If the customer finishes it within 2

seconds after the threshold is reached, then the alarm will be too late. So the choice of the time between the detection and the alarm requires cautiousness, and the time gap can vary between customers in reality.

The second is that it requires more calculations. In the original design, the optical sensors will only send out a logical on/off output, which is easily recognized and processed. The capacitive sensor, on the other hand, outputs a variable voltage based on how much liquid is in the sensor. After receiving this data, the microcontroller needs to first calculate the precise liquid level, then decide whether or not an alarm needs to be sent. This process will introduce a larger latency than that of the original design, added to the waiting time, and as a result, may not be able to accurately reflect the real-time liquid level. Aside from that, a more powerful and costly microcontroller will be required for this design.

The third weakness comes down to the size. Compared to the optical sensor, which is just a tiny probe, the capacitive tube will need to be as tall as the cup to fully measure the liquid level variations. Since it will be mounted outside the cup, it will be quite obvious to the customer, and may sometimes obscure the customer when grabbing the cup, which means a bad user experience.

Our targeted customer group will be high-end restaurant owners. After all, this system is designed purely to decrease the time gap between the drink running out and the waiter refilling, thus improving the customer experience. In a pub, the customers will not care that much about the time gap, and the owner will most likely not have enough budget to invest in such a system. Taking this into consideration, we decide to stick to the original design, since it will account for the real time effectiveness and the tilting issue better than a capacitive sensor system; secondly, it will be more unnoticeable to the customer, and thus more comfortable to use, which is what high-end restaurant owners care the most. As for the other issues, a flexible installing method can be implemented to improve the installation issue, but the system and labor cost issues are a compromise we will need to make.

2.1.2. Control Module

We want to design the control module to send information to the bluetooth module and LED module once the liquid in the cup has gone below a certain threshold, which can be done with a microchip controller. One possible alternative was to immediately send any data from our system to the computer which talks to the waiter. This allows the microcontroller to do less work, and be cheaper as a result. However, this will create a bottleneck towards the computer and the bluetooth itself, which will need to send information as well as receive it again just to light up. Because Radio transmission takes 12 mA while the microcontroller will only take 4.6 mA, we will overall save energy by using a microchip controller to perform all necessary computations. Thus, we would use

the ATPMEGA328P-AU QFP microcontroller, which can perform the simple yet necessary computations for our project.

2.1.3. Bluetooth Module

In the part of transferring data, we have changed the way of doing it, instead of using wifi, as we did in the first project, for this second project we have chosen bluetooth. There are several reasons why we have changed from wifi to bluetooth.

To begin with, for the first project, wifi was more available as the Raspberry Pi has installed a wifi component inside it, also it is more reliable, as it can transfer information via a TCP connection that will ensure that the connection is established and secure. In addition, for the whole project we were only needing one transmitter and receiver station, so we could afford having a most costly and less affordable component. Also as time was crucial in our first project, we needed a fast way of transmitting data and wifi, once connection is established is faster and, as I indicated before, more reliable than Bluetooth.

For this second project, scenario was different, we need a lot of substations, one for each cup of water that will be transmitting, so we will need a cheaper component than wifi and Bluetooth is, also latency was not such a problem as it was in the previous project, as we have a margin of some time to let the waiter know if a cup is empty or not. Furthermore, the Bluetooth module is more manageable than Wifi and it is more compatible and more frequently used with the microcontroller that we will need in each cup.

2.1.4. Light Module

This module, shown in figure 7 below, will consist of a multi-color LED which has the three basic colors red, green and blue, and these colors can be combined to display various other colors. It should be noted that the driving voltage for the R, G, and B LEDs are not the same: 2.0V for red, 3.2V for green, 3.1V for blue. The driving current will be 20mA for all three LEDs. The brightness intensity of the LED will be 250-700mcd, which is sufficient for the waiter to notice, but not too bright to annoy the customer.

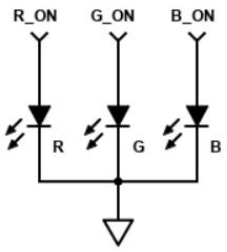


Fig.7 Multi-color LED schematic

2.1.5. Power Module

Components in the system that require powering:

-OLS200D3SH Optical Level Sensor*2: 15.4V, 2.5mA each
-LM741 op amp: 22V, 500mW
-ATMEGA328P-AU TQFP microcontroller: 1.8V, 0.2mA
-COM11120 multi-color LED: max 5.2V, 20mA
-RN4870 bluetooth chip: 3V, 10mA

Total power needed: 15.4*2.5*2+500+1.8*0.2+5.2*20+3*10=711.36mW
It's required to support the system for 2 hours: 1211.36*2=1422.72mWh
Considering the size and weight requirements of the system, a coin cell battery should be used.
Battery choice: RJD3555HPPV30M rechargeable Li-ion coin cell battery

-Power output: 3.7V, 500mAh
-recharge time: <3 hours

Total energy stored is 3.7*500=1850mWh, sufficient for the system usage. The recharge time is reasonable.

2.1.6. User Interface

This is a laptop app implemented to allow the customer to manually assign each cup with a color to represent the drink, continually monitor the alarm signals sent from the control modules, and allow the waiter to manage all the cups and see the alarms. To achieve this, the design of the app should follow these principles:
-Manageable. Create a list that allows the waiter to keep track of all the cups.
-Alarming. Create alarm signals associated with a cup easy to spot.
-Quick. Latency between a reception and a visual alarm should be no more than 200ms.

2.1.7. Software-based simulation

Bluetooth part.

For the bluetooth module, we will configure a UART module called HC-06 that will allow us to communicate easily with the microprocessor via Bluetooth. HC-06 will work as a bridge between the computer which will be connected via Bluetooth and a microcontroller that will be connected to with an asynchronous

interface in series. By default the interface in series will work at 9600 bauds in a 8N1 format with two voltage levels from 0 to 5 V.

In order to configure the PC we will need to use the program Putty , that is a versatile program that allows us, among other things, to connect ourselves to a serial port. We will need to open it as “Serial” as the type of connection and introduce the port at which HC-06 is connected.

To connect the module, we will need to connected it to Tx port and to the Voltage source and ground connection, as figure 8 shows:

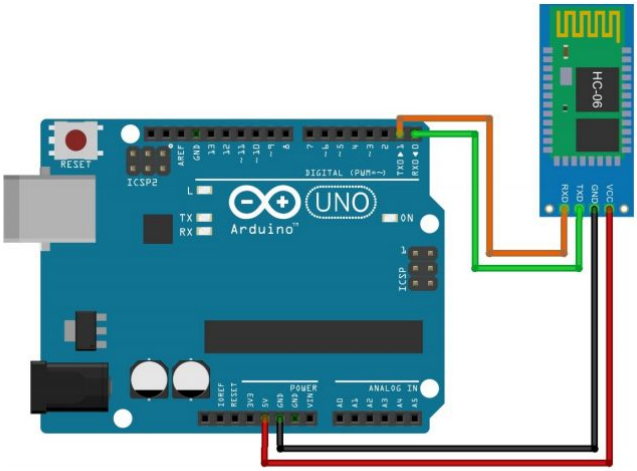


Fig.8 Arduino Bluetooth Chip Connection

For testing purposes, we will send characters with the microcontroller and must be received by the computer and shown in the Serial window of the Putty program. The transmission will be managed by polling, and the transmitter will be enabled when the chain is going to be sent and will be disabled when it is finished sending.

A code example of the program will be:

```
#include <xc.h>
// Configuration bits: selected in the GUI (MCC)
// DEVCFG3
#pragma config PMDL1WAY = ON    // Peripheral Module Disable Configuration->Allow only one reconfiguration
#pragma config IOL1WAY = ON     // Peripheral Pin Select Configuration->Allow only one reconfiguration
#pragma config FUSBIDIO = ON    // USB USID Selection->Controlled by the USB Module
#pragma config FVBUSONIO = ON   // USB VBUS ON Selection->Controlled by USB Module
// DEVCFG2
#pragma config FPLLIDIV = DIV_2   // PLL Input Divider->2x Divider
#pragma config FPLLMUL = MUL_20         // PLL Multiplier->20x Multiplier
```

```
#pragma config UPLLIDIV = DIV_12 // USB PLL Input Divider->12x Divider
#pragma config UPLLEN = OFF        // USB PLL Enable->Disabled and Bypassed
#pragma config FPLLODIV = DIV_2  // System PLL Output Clock Divider->PLL Divide by 2
// DEVCFG1
#pragma config FNOSC = PRIPLL    // Oscillator Selection Bits->Primary Osc w/PLL
(XT+,HS+,EC+PLL)
#pragma config FSOSCEN = OFF     // Secondary Oscillator Enable->Disabled
#pragma config IESO = ON    // Internal/External Switch Over->Enabled
#pragma config POSCMOD = XT      // Primary Oscillator Configuration->XT osc mode
#pragma config OSCIOFNC = OFF   // CLKO Output Signal Active on the OSCO Pin->Disabled
#pragma config FPBDIV = DIV_8      // Peripheral Clock Divisor->Pb_Clk is Sys_Clk/8
#pragma config FCKSM = CSDCMD // Clock Switching and Monitor Selection->Clock Switch
Disable, FSCM Disabled
#pragma config WDTPS = PS1048576         // Watchdog Timer Postscaler->1:1048576
#pragma config WINDIS = OFF         // Watchdog Timer Window Enable->Watchdog Timer is in
Non-Window Mode
#pragma config FWDTEN = OFF       // Watchdog Timer Enable->WDT Disabled (SWDTEN Bit
Controls)
#pragma config FWDTWINSZ = WINSZ_25 // Watchdog Timer Window Size->Window Size is
25%
// DEVCFG0
#pragma config DEBUG = OFF         // Background Debugger Enable->Debugger is Disabled
#pragma config JTAGEN = ON         // JTAG Enable->JTAG Port Enabled
#pragma config ICESEL = ICS_PGx1           // ICE/ICD Comm Channel Select->Communicate
on PGEC1/PGED1
#pragma config PWP = OFF  // Program Flash Write Protect->Disable
#pragma config BWP = OFF  // Boot Flash Write Protect bit->Protection Disabled
#pragma config CP = OFF     // Code Protect->Protection Disabled

void InicializarReloj(void)
{
        SYSKEY = 0x0;         // Nos aseguramos que OSCCON está bloqueado
        SYSKEY = 0xAA996655;  // Se escribe la primera clave en SYSKEY
        SYSKEY = 0x556699AA;  // Se escribe la segunda clave en SYSKEY
        //  Ahora OSCCON está desbloqueado y podemos modificarlo

        // Se configura el reloj para usar el oscilador externo usando PLL.
        // El oscilador está dividido entre 2, pues el PLL ha de tener una entrada
        // entre 4 y 5 MHz (DEVCFG3.FPLLIDIV = 001).
        // Al PLL entran por tanto 8MHz/2 = 4 MHz, luego el PLL lo
        // Multiplica por 20 (80 MHz) y esta salida se divide entre 2 para
        // obtener un reloj principal a 40 MHz. Este reloj se divide por 8 para
        // obtener el reloj del bus de periféricos. Para ello, en el
```

```
        // registro OSCCON se hace:
        // PLLODIV = 001 (div. por 2)
        // PBDIV = 11 (div. por 8)
        // PLLMULT = 101 (mult. por 20)
        // COSC = 011 (Oscilador principal con PLL)
        // NOSC = 011 (Oscilador principal con PLL)
        OSCCON = 0x081D3300;
        // Una vez hemos terminado, lo volvemos a bloquear
        SYSKEY = 0x0;
}
int main(void){
  char cadena[] = "Hello, world?\n I am fine";
        int cont_cad=0;
        int pulsador_ant, pulsador_act;
        ANSELB &= ~ ( (1<<5) | (1<<7) );   //Se configuran como digitales
        TRISB = (1<<5);   //Se pone el pulsador activo
        SYSKEY=0xAA996655; // Se desbloquean los registros de configuración para hacer el
mapeo de puerto
        SYSKEY=0x556699AA;
        RPB7R = 1;    // Puerto RB7 conectado a salida de la UART1 (U1TX)
        SYSKEY=0x1CA11CA1;        // Se vuelven a bloquear.
        U1BRG = 32;             // 9600 baudios
        U1MODE = 0x8000;   // UART: arranque (ON=1), configuracion 8N1, velocidad estándar
        pulsador_ant = (PORTB>>5) & 1;
        while (1) {
        pulsador_act = (PORTB>>5) & 1;
        if ( (pulsador_act != pulsador_ant) && (pulsador_act == 0) ) {
        U1STAbits.UTXEN = 1;   //Cuando se pulsa el pulsador se habilita el receptor.
        cont_cad=0;
        do {
                U1TXREG = cadena[cont_cad];  //Se va leyendo caracter a caracter la cadena y
se van enviando
                cont_cad ++;
                while (U1STAbits.TRMT == 0)
                ;
        }while (cadena[cont_cad] != '\0');  //Mientras que la cadena no este acabada.
        U1STAbits.UTXEN = 0;
        }
        pulsador_ant = pulsador_act;
        }
}
```

As we receive the characters sent, we will know that we can send and receive characters and therefore change the characters that we send to whatever

we want to send, in this case a yes or no saying if the cup is empty or not and the cup number. Here is an example of the result of the previous code:

3. Second Project Conclusions

3.1. Implementation summary

We created functional code to use for our UART module, which allows the computer and our device to communicate with each other, as well as constructed a cost-analysis and setup analysis for our module, were it successfully created and implemented into a cup in a restaurant. Finally, we also constructed a PCB to use for our detection module and microcontroller. This allows us to, hardware permitting, immediately begin working on building our device, as most of the bluetooth and detection software has already been worked upon. This means that with a physical model to attach it to, we could have the sensor send information to the microcontroller and then to the computer, and warn the waiter of the liquid running low as a result.

3.2. Unknown, uncertainties, testing needed

We are unable to complete many of the hardware components because we do not have access to the lab. This also means that we have very little physical usage of the optical sensor outside of the theoretical. In order to solve this problem, we would need to purchase the sensor, PCB, etc. So that we could wire them together and test if our current setup will work. This includes wiring the microcontroller to the Bluetooth module, LED module, and Detection module. We would also need to further develop any container for our project, to ensure both the customer and the device's safety when exposed to large quantities of liquids and physical contact. Both these problems are, unfortunately, only solved by having access to the lab in order to purchase, sodder, and test our equipment.

3.3. Ethics and Safety

The general goal of both the IEEE Code of Ethics [5] and the ACM Code of Ethics [6] is to ensure quality without either intentionally or unintentionally causing harm. Our design does not appear to break any laws; the device's only detection module is a capacitive sensor, which will not invade anyone's privacy since it is only detecting capacitance to determine the water level. Furthermore, the sensor requires very little power and would not be harmful to a person. The only wireless connections are made via bluetooth, so there is little to no possibility of invading personal privacy or interfering with other signals.

Every subsystem aside is contained within one compartment, which only comes in contact with the cup, a human intending to drink with it, and a liquid. We will take our due diligence to ensure that the device will be able to withstand hot conditions such as hot coffee. We also need to ensure that every module that requires power is encased in plastic to ensure that it will not harm any human contacting it. This will double as protection from water during either a wash or usage. Overall, the power required of the system is minimal, so risk is also at a minimum, and coin cell batteries are generally safe to use aside from swallowing concerns.

Water safety laws by the EPA[7] limits many chemicals and materials from being inside drinking water, as they create hazards. One notable material that is limited is copper, which will cause gastrointestinal distress and eventually liver and kidney damage. However, our priority is to ensure that our product will be encased in a waterproof layer that will both prevent any electronics from entering the drinking water and the drinking water from ruining our product. Because all metals and inorganic materials will be shielded from the drinking water, our product complies with EPA water safety laws.

Final safety concerns occur in the creation of our device in the Senior Design Lab. We have already performed standard safety training, and understand how to use the equipment while avoiding electrical shorts, shocks, and burns. The most dangerous would probably be soldering, as forgetting something as simple as forgetting to turn it off will create safety hazards to us and everyone in the lab. As such, safety is our priority when it comes to physical work in the design lab and we will work with at least one other person in case either find the situation dangerous, in order to adhere to rule 9 of the IEEE code of ethics [5].

Overall, we believe that we are following both Codes of Ethics [5], as we are not breaching any regulations or standards. We will keep careful note to prevent our primary controller from overheating, but otherwise no safety concerns or breaches of privacy arise.

3.4. Project Improvements

3.4.1. Bluetooth Module Setup Analysis

Since the bluetooth chip's maximum working range is approximately 10 meters, which obviously does not cover the entire restaurant, it is likely that some cups might not be able to send signals to the laptop. Further design is required for the physical setup of the system. Extra components might be required.

For analysis, take for example a restaurant floor plan of 5000 square feet, or 464.51 square meters. For a fine dining plan, (suitable for our case since our primary customer group will be high-end restaurant owners) the recommended dining area is 18-20 square feet, or 1.67-1.86 square meters, per person. [8] Take 1.86 square meters for our design, and assume the restaurant's floor plan and the dining areas are square. This means that the width of the floor plan is about 21.55 meters, and each dining area has a width of 1.36 meters. 21.55/1.36 = 15.80, meaning there will be at most 15*15 dining tables in the restaurant. The system has a maximum communication range of 10 meters, approximately 10/1.36 = 7.33 areas long.

Under these assumptions, the tables at the center area are highly likely to lose contact with the waiting stations located at the corners of the restaurant, displayed in figure 9 below.

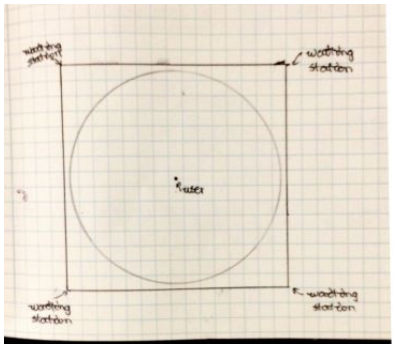


Fig.9 Center Table Example

The solution is to divide the 15*15 dining areas into smaller 5*5 areas, and add relay bluetooth modules at the conjunction points, shown in figure 10 below.

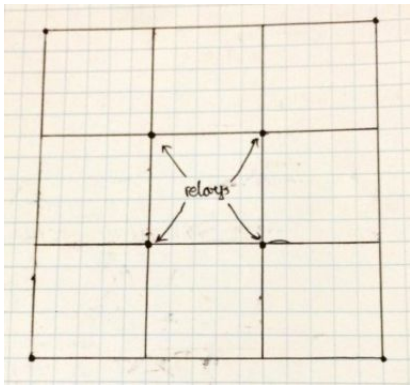


Fig.10 Relay bluetooth modules

After this design is introduced, the distance between a cup and a nearest bluetooth reception point (either the waiting station laptop or the relay module) will be greatly reduced. Considering the farthest possible distance, namely the diagonal distance, is (5-0.5)*sqrt(2) = 6.36 units length. (the 0.5 units length is subtracted to account for the cup being at the center of a dining area) This is shorter than the 7.33 units length which was calculated before. As a conclusion, this design will cover all the areas in the restaurant. The Compromise is having to add additional components to the system, increasing the system's cost.

However, with this design, more problems arise: there is a possibility that multiple waiting stations or multiple relay modules will receive the same cup's warning. How can repetitive alarms be avoided?

Firstly, for the waiting station reception: consider a table located at the center along a line between the two neighboring laptops. The distance to either laptop is: Sqrt(0.5^2+7.5^2) = 7.52 units > 7.33 units. Thus it's actually impossible that there are two laptops receiving the alarm from the same cup.

Secondly, what if a relay module and a station receives the same alarm? The answer is: it doesn't matter. The relay will transmit the alarm to the station anyways.

Lastly, is it possible to have one station and multiple relays receiving the same alarm? It could be that one station is alarmed, and the relay communication routes the relayed alarm to another station. This is shown in figure 11.

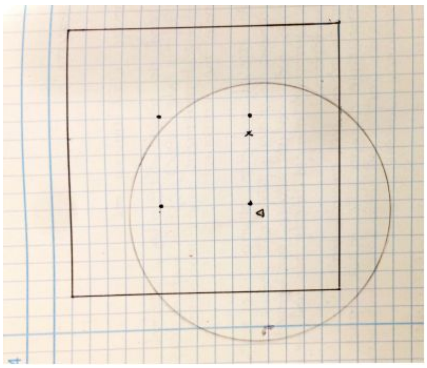


Fig.11 Station and multiple relays reception

As shown in the figure (drawn to scale), this will cause a real problem. The solution will be to assign a certain order to the station laptops. For example, the stations would be numbered 1-4. If multiple stations receive alarms from more than one relays about the same cup, then the station with the smallest assigned number will alarm the waiter. Long distance wireless communication between laptops is achieved using wifi protocol.

Bluetooth chips typically have shorter working ranges than wifi chips, but are less power consuming and smaller in size. Considering the power and weight requirements of the system, a bluetooth module is chosen for the system instead of wifi chips. The relay modules need only receive and transmit control module signals. Low-power microcontroller chips with built-in bluetooth protocol will be used.

3.4.2. Cost-Benefit Analysis

Bluetooth chip cost: $7.00 each
Controller chip cost: $2.00 each
Optical sensor cost: $40.67 each
Op amp cost: $0.87 each
Resistor cost: $0.20 each
Battery cost: $18.78 each
LED cost: $1.05 each
Sum: 7+2+40.67*2+0.87+0.2*2+18.78+1.05=$111.44 for each system.

Consider a purchase of 100 such cups. Four additional bluetooth chips and microcontrollers will need to be purchased to build the relay module.
Total cost: 111.44*100+(7+2)*4=$11,180.00

Compared to the system’s cost, the extra labor cost introduced by having to remove the system before washing the cup is negligible.

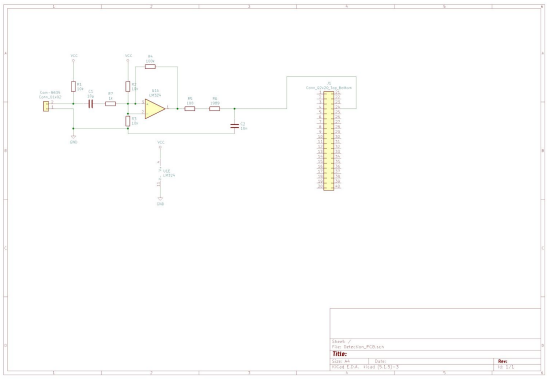
A successful restaurant's net avenue rate can range between 4% and 6%. Our targeted customer group is high end restaurant owners, so Le Diplomate, a mid-to-high end chain restaurant practicing a waiter system, is examined. Its annual sales avenue was $16,308,810 for 2018 [9], translating to a net avenue income of $652,352 to $978,529. This means it will take Le Diplomate at most 11180/652352 = 0.017 years, or roughly 6.2 days to balance the cost of using this system.

Compared to the cost aspect, which is quickly recovered, the benefit aspect introduced by using the system is more subtle, yet very valuable. Consider the time after the cup is emptied and before the waiter comes to refill it. If the waiter is about 10 meters away from you, it would take some 7-10 seconds for them to get to you. If it happens that there are no waiters nearby, or they didn't notice your cup is emptied, it would probably take longer than that. Also, the waiter may not know what drink to refill with, and might make some detours to finally get you the drink. However, after using the system, this time gap can be greatly reduced, and ideally, even eliminated. Since the system informs the waiter to start preparing when the drink level is still relatively high, and the type of drink is displayed on both the waiting station and the cup, there would be no time wasted for the waiter to come to the correct table, with the desired drink. This will no doubt improve the customer experience, and increase their chance of returning and recommending the restaurant to other people. How much more benefit this will bring to the restaurant, however, will only be obvious after long term market experiments.

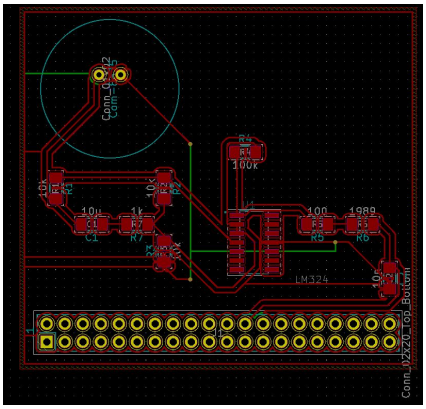
4. Progress made on First Project

We constructed the PCB for our first project, as well as its associated footprints, which functioned to receive the input from the microphone and sent it to our microcontroller to perform its computations. The audit was approved by betway but unfortunately never shipped due to the coronavirus.

Schematic Layout:



PCB Layout:



5. References

1) “Bottomless,” Red Robin, 2020. [Online]. Available: https://www.redrobin.com/pages/everyday-value/bottomless/. [Accessed: 5-8-20].
2) OLS200D3SH product specs. [Online]. Available: https://www.digikey.com/product-detail/en/cynergy-3/OLS200D3SH/725-1281-ND/4209104 . [Accessed 5-8-20].
3) “Optical liquid level sensors: how they work”. [Online]. Available: https://www.youtube.com/watch?v=ZByijUX_TDY&t=27s . [Accessed 5-8-20]
4) “Three capacitive sensor-based-level measurements”. [Online]. Available: https://www.researchgate.net/figure/Three-capacitive-sensor-based-level-measurements-32_fig4_325702981 . [Accessed 5-8-20]
5) “IEEE Code of Ethics,” IEEE, Jun-2019. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: 5-8-20].
6) “Code of Ethics,” Code of Ethics, 22-Jun-2018. [Online]. Available: https://www.acm.org/code-of-ethics. [Accessed: 5-8-20]
7) “National Primary Drinking Water Regulations,” EPA, 14-Feb-2020. [Online]. Available: https://www.epa.gov/ground-water-and-drinking-water/national-primary-drinking-water-regulations#one. [Accessed: 5-8-20]
8) “How to Create a Restaurant Floor Plan”, totalfood.com. [Online]. Available: https://totalfood.com/how-to-create-a-restaurant-floor-plan/. [Accessed 5-8-20]
9) “These are the 50 highest grossing restaurants in the US”. [Online]. Available: https://www.usatoday.com/story/money/restaurants/2018/08/09/highest-grossing-restaurants-in-america/37227589/ . [Accessed 5-8-20]