**Project Documentation**

**Looney Code**

**EC327 - Final Project**

**December 12, 2022**

Beren Donmez, Jiahe Niu, Christian So

**Intro (Marketability and Current solutions)**

According to an article by *Insider*, large amounts of harmful bacteria were discovered in both small and chain supermarkets. However, the most detrimental bacteria were found on the handlebar of shopping carts. In certain scenarios, the shopping carts contain 270x more germs than the average toilet handle. In addition to this, 75 percent of the germs identified in grocery shopping carts were found as harmful. Another article from t*oday.com* provides evidence of the 10 germiest places in a supermarket; not surprisingly, the first one is the handle of the shopping cart and the second one is the child seat of the shopping cart. This means children are being exposed to many kinds of bacteria as well.

The current solutions seem to be lackluster. Having researched an article from *WCNC* dating back to 2020, we discovered a solution marketed as “CartSafe”, which is a cover made for shopping carts. This solution seems to be unpopular amongst customers based on the low amount of Amazon product reviews. Another prior solution to this problem is increasing the number of hand sanitizers throughout the store. However, According to CBS News Medical Correspondent Dr. Jennifer Ashton, hand sanitizers not only fail to provide 100 percent protection, but also lasts for only 2 minutes.

**Our Solution:**

Based on our research and analysis the major problems of the prior solutions are they are either inconvenient, costly, time-consuming, or non reusable. Hence the solution we need to innovate, and implement must have the following attributes:

* Dramatically decrease spread of germs without additional supermarket employee manpower
* Reusable to make sure it is environmentally friendly
* Low-cost so users are motivated to invest in it (time and money)
* Convenient so that users are inclined to use it

Therefore our innovative solution is: Contactless Motorized Cart (CMC). Using Arduino, a CPU, motorized wheel, and the Leap Motion the CMC will allow the shopper or user to control the cart without making physical contact and spreading germs.

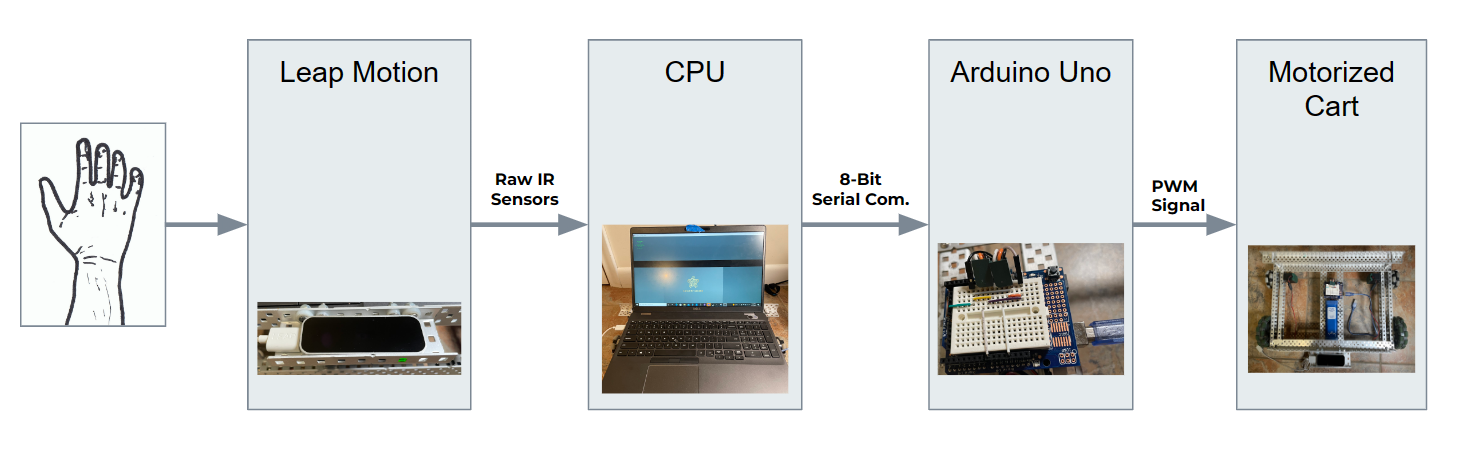


Figure 1. System block diagram of the CMC

**Processing Code with leap motion**

All the global variables are declared in addition to the setup of the graphical user interface. The five commands that will be sent to the Arduino to process are Stop, Forward, Backward, Left turn, and Right turn. In order to decide which command is sent, hand detection is necessary, which is indicated by a flag variable that is only set to 1 when the hand is grabbed. The hand is known to be grabbed when the grab strength is larger than 0.9 as when it is tight as a fist the strength is 1. PVector is a class that contains x, y, and z coordinates. When the hand is first detected to be grabbed, the initial position of the hand and the angle of the palm to the fingers are stored and will be compared to the eventual movement of the hand. The threshold for change in direction (forward and backward) was chosen to be 10, which we experimentally found would be the most natural. We wanted the hand to be able to move slightly but not make the machine twitch back and forth. We allowed for a threshold distance of around 1 inch. If the hand is more than 10 units(~1 inch) away from the stored initial position, then the Forward command will be sent to the Arduino to correlate the motors to the command. Similarly, with the Forward and Back, a threshold was experimentally obtained for turning commands. Coincidentally, the threshold for turning was also 10. If the angle from the palm to the fingers was more than 10 from the initial angle then the corresponding left or right command will be sent to the Arduino.

**Processing Graphical Interface**:

As engineers, while we have to come up with a solution for a specific problem or need, we also need to think of the visuals and graphics since they are a big part of the integration of technology and humans. For us, displaying the status of the device and showcasing the hand movement, using the if-else statement while checking the value of “send”, created a better user experience and easier to debug. First, when the user’s hand gets into the scope of our sensor, we map the X,Y, Yaw difference values from -100 to 100. The shopping cart moves until the detected hand is back in the center. On the screen, three dots appear when a hand is detected, the chosen color being #00E310. Draw() function draws the gray handle bar on the screen using the Processing code. The statement, for(Hand hand : leap.getHands () is an important component which loops through the hand objects seen by the sensor. It lets the handlebar turn white, storing the initial position of the hand so that it knows where the center is. The setup of the initial states can be found in slide 11 of Project Architecture.

**Arduino Uno:**

For the Arduino code, we have two classes: PID and motor. We overloaded the PID class for PD control, where the only inputs are kP and kD. In the PD, the kI is set to 0. In the Arduino Uno code, motor object is created which has inputs to instantiate what Arduino Uno pins are connected to the motors. This class controls commands and power sent to the motor which will be discussed later.

**PID Class:**

A class was created for Proportional Integral Derivative (PID) Control in order to smooth out the motion of the chassis, Tweety. There are three components to PID (Proportional control, Integral control, and Derivative control) which stabilizes the behavior of the system, in this case, the movement of the Tweety.

The Proportional control results in a more stable system as it reduces steady state error. Proportional derivative control allows the system to react fast and suddenly to an unexpected change. The Proportional integral control resets the system so that it is back on track. The Proportional Integral Derivative is a combination of all three constants (Kp, Ki, Kd) in the PID control. PID control guides the system so that it can respond to and gives good error tracking. Ultimately Proportional derivative control was used as we wanted the system to react to the sudden hand movements of the user. Additional error tracking was unnecessary to the project at this time.

**Motor Class:**

The motor class has a default constructor where it sets the starting command to stop. The command is a private variable in the motor class because we did not want it to be easily overwritten outside the class. The other private variables are Servo which is an object from an imported library. Changed values including the pulse width modulation (PWM) were mapped to the Servo in order to control the motors. We tuned the PWM so that it works with the 393 Vex motors on the chassis.

According to the command that it has received from the Processing Code, the motion of the direction of each motor is determined when the command is changed.

The two motors are initially screwed in opposite directions. Therefore, when the command is forward, the speed of the motors are opposites of each other.

**GitHub:**

Multiple branches were created as the code could only be tested one at a time. There were multiple times that we worked remotely, and therefore the use of github was essential. However, due to everyone working on the same code at the same time, branches were made so that the latest versions of each person's code could be stored without overriding each other’s. Ultimately, after the code was tested and seen to work, it was all combine

**Conclusion:**

Overall we have successfully designed a product that is:

* Reusable to make sure it is environmentally friendly - it is electric
* Low-cost so users are motivated to invest in it - one time cost
* Convenient so that users are inclined to use it - ease of use and intuitive

Due to the rise of infectious diseases like COVID over the past two decades, public places such as supermarkets have become a breeding ground for bacteria. Specifically, the handlebars of shopping carts are places where almost every customer has to touch. In order to combat this challenge, we have designed a hands free motion detecting chassis CMC. Using both hardware and software implementation, the sensor is able to detect a hand when it is grabbed and analyze the continuous hand motion. Depending on movement of the hand, or the lack thereof, the corresponding command will be sent to the Arduino code which controls the movement of the 393 Vex motors. Thus, the Contactless Motorized Cart creates a safer environment for shoppers and workers alike.

**Refer:**

Fantozzi, J. (2017, November 14). *Your supermarket is filled with germs — here are 7 facts that will make you dread grocery shopping*. Insider. <https://www.insider.com/dangerous-supermarket-germs-study-2017-11>

*How dirty is your shopping cart?* (2020, October 30). Wcnc. <https://www.wcnc.com/article/life/shopping-cart-cover/275-b9786264-4919-40b1-b63a-c4d7bcaf034a>

Gibbs, K. B. (2017, March 24). *10 germiest things in the grocery store — and how to stay safe*. TODAY.com. https://www.today.com/food/10-germiest-things-grocery-store-how-stay-safe-t109564

CBS News. (2010, October 25). *How Long Do Hand Sanitizers Really Work?* <https://www.cbsnews.com/news/how-long-do-hand-sanitizers-really-work/>

*Servo - Arduino Reference*. (n.d.). <https://www.arduino.cc/reference/en/libraries/servo/>