SMART SHOPPING CART

By

Di Fan

Ying He

Xuyang Yao

Final Report for ECE 445, Senior Design, Spring 2012

TA: Mustafa Mukadam

26 April 2013

Project No. 22

Abstract

As technology progressed, lives have been significantly improved due to the emergence of laborsaving and intelligent utilities. However, the shopping carts in major stores have experienced little changes and serve only simple purposes ever since they were first manufactured. Pushing these carts around becomes noticeably painful when they carry heavy loads. Even though generally people are not moving fast in stores, there is a risk of collision if obstacles arise out of sight. Moreover, products can be extremely difficult to locate before people get familiar with the store layout. Considering all the disadvantages discussed above, we aim to implement a smart shopping cart that provides great convenience and efficiency to customers. This report documents the design, construction, verification, and cost of this project.

Contents

1. Introduction

1.1 Purpose

The purpose of this project is to implement an automatic shopping cart that provides great convenience and efficiency to customers, which eliminates human labor to push heavy loaded carts and help customers easily locate the desired products. Our goal includes avoiding collisions with obstacles and detecting accessible routes, collecting user input and optimizing the path required to accommodate the shopping list, and informing customer when the cart is blocked. To achieve these goals, we designed each module based on our choice of microcontroller, Arduino, since it has a user-friendly working platform and adequate processing speed that satisfies the need of our project.

1.2 Functions

Our project is able to move automatically under the drive of two 12 V DC motors, whose voltage input is controlled by Arduino to adjust moving speed. Three ultrasonic sensors are embedded in front, left and right of the cart, detecting which path is accessible and alarming the customer when obstacles found in 20 cm range. The communication between the cart and customer is achieved by two RF transceivers operated at 433 MHz band: one attached to the cart and one carried by the customer. The cart receives customer's shopping list through the user interface, which consists of a keypad, a LCD and an ATmega328, then generates the shortest path covering all products selected. Our main microcontroller, Arduino, works as the information transfer and processing center. It is connected to most subcomponents and is programmed with the logic that responses to all possible situations.

1.3 Blocks

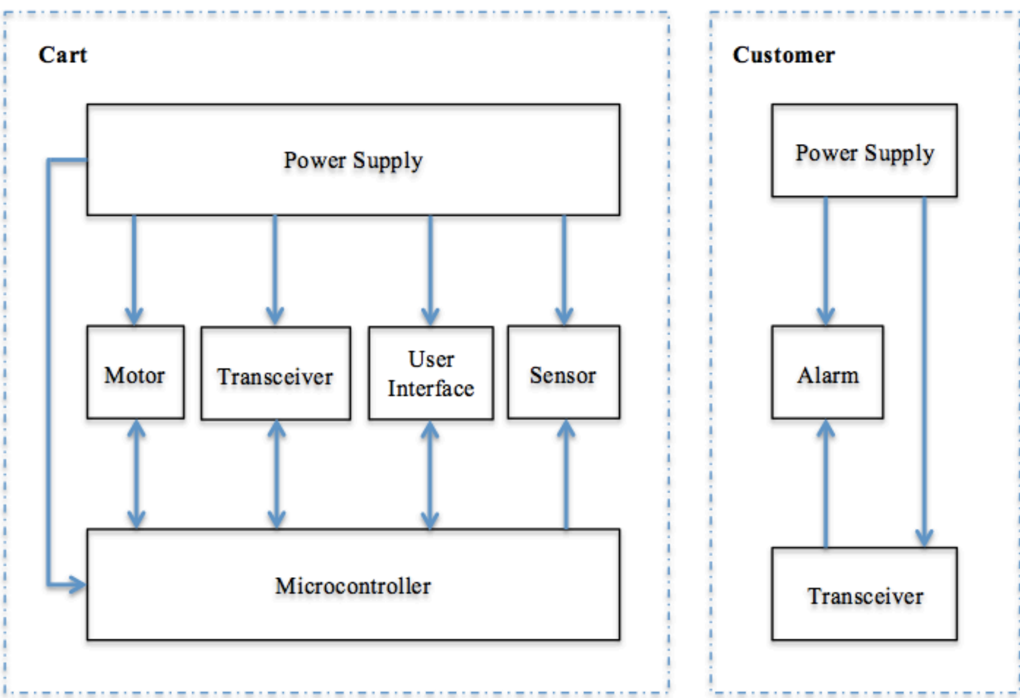

Figure 1 Block diagram of the entire project

1.3.1 Power Supply

The power supply module consists of 3 sets of four 1.5 V AA batteries. Four 1.5 V batteries provide power to Arduino, which can output both 3.3 V and 5 V voltage. The remaining eight batteries, altogether 12 V, provide power to two DC motors.

1.3.2 Transceiver

One RF transceiver is connected to Arduino, which works as transmitter to send out information collected from sensors. The other transceiver is controlled by a separate ATmega328, the processing unit of Arduino, to receive signals and determine whether to alarm the customer. RF transmitter and receiver pair is responsible to cover a range of 10 m and maintain fast and reliable communications.

1.3.3 Motor

Two DC motors are connected between 12 V power supply and two PWM output pins of Arduino, which adjusts the rotation speed of motors based on different duty cycles. Since Arduino outputs a maximum of 40 mA current, two TIP120 transistors are used to amplify the current to the scale that is sufficient to motors.

1.3.4 Sensor

Three 40 kHz ultrasonic sensors are powered by the 5 V output of Arduino. Each sensor has one input pin, trigger and one output pin, echo, which are both connected to Arduino digital pins. Based on the pulse width of the reflected sound wave, Arduino is able to calculate the distance between obstacles and the cart with known sound speed. Sensors should effectively cover a range of at least 20 cm.

1.3.5 User Interface

Our user interface is made up of a 12-button keypad, a LCD and an ATmega328 as microcontroller. The keypad reflects which product is selected by the customer and transfers this information to ATmega328. Then ATmega328 prints the corresponding names onto LCD to double check with the customer before they confirm their choices.

1.3.6 Microcontroller

Our main microcontroller, Arduino, is installed on the cart and connected to two motors, one RF transmitter and three sensors. One ATmega328, which is carried by the customer, is wired to the other RF transceiver and a LED alarm light. Another ATmega328, which is used in use interface, talks to the LCD display and keypad. Additionally, Arduino will output 3.3 V and 5 V voltages to all other components.

2 Design

2.1 Design Procedure

2.1.1 Power Supply and Motor

When we first designed the power supply, we had to determine all voltages required by individual components: two motors need 12 V DC voltage, while sensors, LCD display, keypad and RF transceivers need 5 V DC voltage. Our initial design was to use several 12 V batteries and a 5 V regulator to scale the 12 V down. Later we found out that Arduino has a regulator built inside and can output 5V DC voltage directly. Moreover, the input voltage of Arduino is 6 V to 20 V. In this case, we decided to use a 12 V voltage source to power two motors and a 6 V voltage source to power the Arduino.

We initially chose A23 12 V battery as our power supply. However, when we got our battery, we discovered that the batteries were too small to provide enough current and power since motors require relatively large currents. Thus, we switched to use eight 1.5 V AA batteries to power the motors and another four 1.5 V batteries to power Arduino.

In design review, we set cart speed to be 1.2m/s. In our later implementation, considering human walking speed and safety issue, we considered slowing down the cart speed. There were two types of motors available for our project: 19:1 motor and 62.5:1 motor. The 19:1 motor had higher rotary speed but lower power than 62.5:1 motor. We finally decided on 62.5:1 motor considering that our cart could be quite heavy and motors with higher power could drive the cart more efficiently.

2.1.2 Transceiver

Operating the RF transceivers was one of the most challenging parts in our design. Initially we wanted to use RF transceivers to implement the indoor positioning system. We planned to place four transceivers at all four corners of the store, one transceiver on the cart and one carried by customer. The RFM12B-S2 transceiver we chose had three channels, high data rate and stable performance. Compared to other transceivers, this model is low-cost and small.

After we began to work on the positioning system, problems occurred. RF transceivers communicate via wireless signals, in other words, electromagnetic waves, which are almost impossible to measure the time delay. Furthermore, the internal delay of data processing was much longer than the time taken to transmit data. We tried our several methods to solve these problems, such as sending the same signal for a thousand times and subtracting the internal time delay. However, since the internal time delay was not consistent all times, we could not get a precise transmission time in the end. We had to admit that RF transceiver failed in implementing the positioning system. Despite its poor performance in measuring delays, RF transceiver worked extremely reliably in transmitting data.

2.1.3 Sensor

Two types of sensors are common in student projects, small and inexpensive: infrared sensor and ultrasonic sensor. Infrared sensor and ultrasonic sensor use electromagnetic wave and sound wave to detect distance, respectively. As widely known, speed of light (3×10^{8} m/s) is much faster than that of sound (340 m/s). In determining distance between sensor and object, infrared sensor measures the strength of signal and ultrasonic sensor measures the time delay which sound wave travels. Since our store is designed to be relatively small and needs good precision, we turned to ultrasonic sensor in our design. A large number of sensors are available in the market and we finally chose the most popular one, HC-SR04 ultrasonic sensor, which has a detective range of 2cm - 5m with resolution of 0.3cm. The data-processing time is also very fast. In later verification, our sensors can detect object accurately.

2.1.4 User Interface

The user Interface includes a LCD display and a keypad. When choosing the model of LCD, we need to know how much information we want to display on it. After deciding that the LCD only displayed the most recent item chosen by customer, we went for a small-sized LCD. The most common LCD is 16×2, which could display 2 lines with 16 characters each. This model is perfect for our design since we could display one line of commend and one line of item entered. Moreover, this type of LCD is relatively inexpensive and we could find a lot of resources online describing how to connect it with Arduino.

Before choosing keypad, we should decide how many buttons the customer would need to use. In design review, we promised that our navigation system has a maximum of ten categories, so at least ten buttons are needed. Additional two buttons would be useful to implement the “clear” and “enter” functions. In this case, 12-buttons keypad meets all our requirements. After searching resources online, we finally decided on the COM-08653 keypad.

2.2 Design Details

2.2.1 Motor

This module aims to control a high-current DC load from a microcontroller. We used PWM pins from Arduino to control the motor speed. However, PWM only output a voltage of 5 V with different duty cycles, but the motor can only reach its full speed at 12V. In this case, we need a transistor to enlarge the current. The model we chose for our design is an NPN-type TIP120 transistor, which allowed a maximum of 4 A current. The base of transistor was connected to the microcontroller output pin, the collector was connected to one side of the motor, and the emitter was grounded. After adding the transistor, we applied 12V to the other end of the motor. As the last step, we added a d1N4007 diode in parallel with the collector and emitter of the transistor in order to protect the transistor from back voltage generated when the motor shut off. Detailed connection of motor is included in Figure 2.

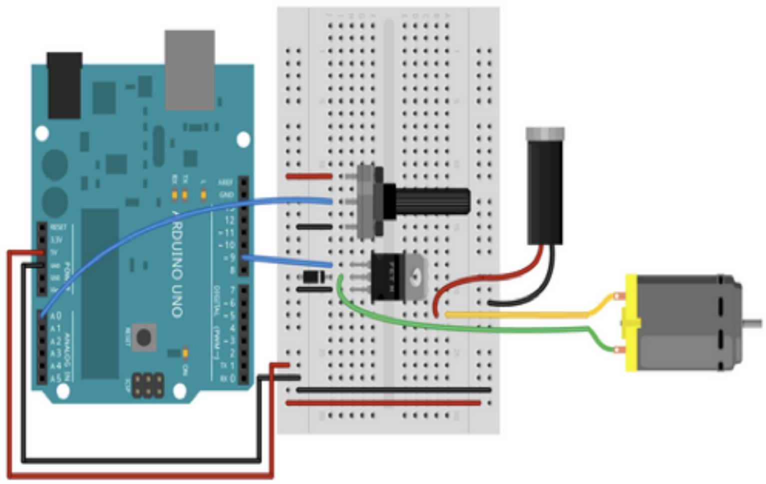

Figure 2 Circuit diagram of motors

2.2.2 Transceiver

The RF transceiver we chose had 14 pins. Among all the 14 pins, we only connected 7 pins to the microcontroller. The hardware part was relatively easy when implementing RF transceiver. When programming the RF transceivers, we used different methods for coding transmitter and coding receiver. We used the same network ID to make sure that the transmitter and receiver could communicate with each other and labeled each transceiver a node ID to so that the transmitter could know which receiver it was sending signal to. The transmission of data was fast and reliable, so the performance of alarm system was very satisfactory.

2.2.3 Sensor

Each sensor consists of an ultrasonic transmitter and an ultrasonic receiver. When the sensor is working, the ultrasonic transmitter kept sending out a series of 40kHz square wave. The square wave keeps travelling until meets some object in front. The wave would bounce back, and the ultrasonic receiver would receive the wave and then transformed it to a pulse. Measuring the pulse width gives us the time delay of the sound wave. If we multiply the time by the speed of sound and then divide by two, we could get the distance between sensor and object. Each sensor had four pins: V_{CC}, GND, Trig and Echo. V_{CC} was connected to 5 V DC. Trig pin was used to make the ultrasonic transmitter send out eight 40kHz square wave, while Echo pin was used to measure the time delay between sensor and object. A timing diagram of the sensor is shown in Appendix C.1.

2.2.4 User Interface

Six pins of LCD display should be connected to the microcontroller as Figure 3 shows. One 10K potentiometer should be used to adjust the brightness of the words when displaying data. The voltage between potentiometer and ground should be 4.5V for best performance. The LCD we chose had three backlight colors: red, green and blue. One resistor should be used to limit current when using backlight: it could not be directly connected to 5V, as it would burn the backlight. The appropriate backlight voltage was 4V after trial and error. Since the I-V curve of LCD was very sensitive, a 15Ω resistor was enough for desirable backlight.

A 12-button keypad has only 7 pins. If one button were pressed, the resistor between specific two pins would decrease significantly. When implementing the keypad module, we need to make sure that the connection between pins and Arduino matched our coding. If one pin is connected wrongly, the whole keypad would not initialize. In conclusion, the keypad module was not very difficult.

2.2.5 Microcontroller

Besides Arduino, we also have two ATmega328 chips, which could not work independently: a standalone circuit is required. We needed one 16 MHz crystal oscillator and two 22 pF capacitors to complete this circuit. One 10 KΩ resistor is connect between the reset pin and V_{CC} to make manually pull it low.

One major feature in our design is the navigation system. Below is the flow chart for our algorithm in Arduino:

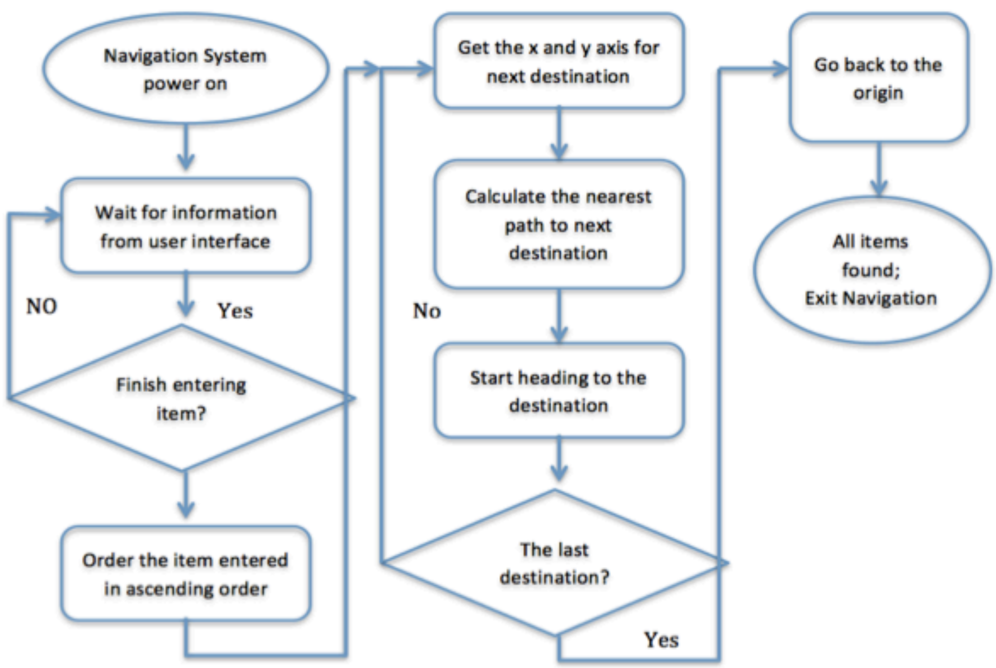

Figure 3 Flow chart of navigation system

We labeled the shelves in such an arrangement so that when we ordered the number in ascending order, the route would automatically be the optimized route. In the real store, we could also use the same method: number the shelf nearest to the origin the smallest number and the shelf farthest to the origin the biggest number, as shown in Figure 4.

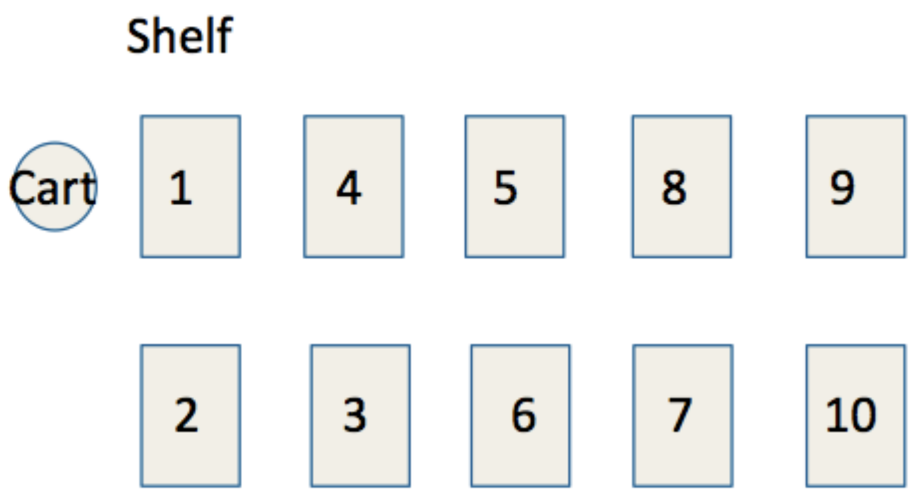

Figure 4 Store layout

When coding the algorithm in Arduino, we need to tell the cart where it should go after determining the next destination. For example, if the cart is originally beside shelf 1 and its next destination is shelf 4, the algorithm would tell the cart first go straight for 0.7m, turn 90 degrees left and go straight for the distance in y direction between 1 and 4, then turn 90 degrees left and go straight for 0.7m. Detailed algorithm is included in Appendix E.

3. Design Verification

Detailed design requirements and verification table is presented in Appendix A. In this section, we are going to cover testing procedure and verification results. Each module is tested independently before the overall performance of the project is examined.

3.1 Power Supply

Three different of levels of voltage supply is required in this project: 12 V for two DC motors, 5 V for Arduino microcontroller and ATmenga328, and 3.3 V for RFM12B transceivers. Our initial design includes two voltage regulators for 5 V and 3.3 V, respectively, since we only provide direct power supply to the motors. As the project advanced, we realized that 5 V and 3.3 V regulators are embedded in Arduino and Arduino can be operated at a voltage input of 6 V to 12 V, which conveniently eliminated the voltage regulators. All power supplies were measured by multimeters in the senior design lab. Twelve 1.5 V AA batteries were used: eight to provide power for motors read 12.42 V in total; the other four for Arduino read 6.23 V.

3.2 Transceiver

In our initial design, transceivers were used to implement an indoor positioning system. However, the plan was not successful due to the immeasurable time delay of wireless signals. Additionally, transceivers need around 3 ms to generate a single character and this time overwhelms the actual delay time. Then we modified the usage of RF transceivers to add more features to our project: updating the sensor results and informing the customer if the cart is blocked. One transceiver is attached to the cart and operated by our main microcontroller, Arduino, while the other is carried by the customer and operated by ATmega328. We tested the coverage of the transceiver by connecting the receiver end to a computer monitor and checked if the information it received matched the information sent by the transmitter while we increased the distance between them. The result shows that the RFM12B-S2 transceivers we chose can efficiently cover the entire senior design lab, which satisfies our requirement. However, when the signal needs to pass through a wall or door, its strength is decreased noticeably. When the two transceivers are placed in adjacent rooms, signal coverage is less than 5 m. Fortunately, our project only applied to the condition that the cart and customer are in the same room. Next, the transceivers were examined on the bandwidth through which they communicated. In the RFM12B library, we can assign the transceiver channel by the following command:

radio.Initialize(NODEID, RF12_433MHZ, NETWORKID);

As stated in the design review, both transceivers work at 433 MHz band. Since the communications were fluent and accurate, 433 MHz should be the working bandwidth of our RF transceivers. Finally, we tested the data rate since the cart's position needs to be updated once per second. The *micros()* command in Arduino was used to determine the time required to generate one bit. From the computer monitor, we read that the transceiver produced a single ASCII character that contains 3 bits in 3006 us (average of 100 transmissions). Therefore, the average data rate is given by

$$data\ rate = \frac{3\times10^6}{3006} = 998 = 0.998\ kbps$$

which fulfills our design requirement. The wired transceiver with ATmega328 is included in Appendix B.2.

3.3 Motor

In this project, two GM9236S025 12V DC motors are used to drive the shopping cart and make arbitrary turning angle by adjusting the relative speeds of two wheels. Since Arduino cannot output analog voltage other than 0 or 5 V, the PWM function serves as the speed indicator. By assigning different duty cycles from 0 to 255 (0 represents 0 duty cycle and 255 represents full duty cycle), Arduino is able to control the input voltage of the DC motors and ultimately, controls their speeds. Note that the two motors are not exactly the same, we need to determine the duty cycle of each PWM pin output to make the cart go straight, turn 90 and 15 degrees in left and right and make U turns. We also measured the distance the cart covered in 1 s to get its moving speed. The result showed that when the right and left wheel received 195 duty cycle (76.5%) and 169 duty cycle (66.3%), respectively, the cart went straight at a speed of 52 cm/s. Unfortunately, this speed is much slower than the speed specified in design review. The problem is that the weight of the cart became much heavier than expected after we added all modules and the motors only have limited powers. The turning function was implemented by stopping one wheel and maintaining the speed of the other for a specific period of time. We marked the start point of the cart and measured the distance it covered in 5 s under each of the right-turn and left-turn commands for 5 times and took the average. Results showed that the largest error occurred at 90 degrees left turns and the percentage error was 8.9%.

3.4 Sensor

Three HC-SR04 ultrasonic sensors are attached to the front, left and right side of the cart to keep track of the obstacles around the cart. Once objects are detected within 20 cm in any direction, the cart needs to make proper response given by the situation. Each sensor is connected to two digital pins of Arduino: one is used to trigger eight 40 kHz pulses and the other is used the read the pulse width reflected, which represents the delay time. We first checked the working frequency of the sensors by connecting the trigger pin to an oscilloscope, which gave a series of pulse at 39.6 kHz. Then the performance of each sensor was read from a computer monitor when Arduino was connected. Testing results are shown in Appendix C.2.

3.5 User Interface

Our user interface consists of a 12-button keypad, a 16×2 LCD and an ATmega328 microcontroller. Costumers use this interface to enter their shopping list then the interface decodes the input information and sends it to Arduino for optimal route determination.

3.5.1 Keypad

The numerical button from 0 to 9 represent 10 different catalogs of products and "#" is used as "enter" and "*" as "clear". The COM-08653 keypad has 7 output pins where 4 pins indicate which row the pressed button is in and 3 pins indicate which column. The keypad is basically a combination of 12 bush buttons that gives different resistance when different button is pressed. How each button is projected by one row pin and one column pin is included in Appendix D.1. To verify the keypad, we first connected it with Arduino since Arduino can be directly monitored by the computer. Each button was pushed at least 5 times to see if Arduino receives correct information about which button is pressed. All trials succeeded. After we tested the ATmega328, the keypad is connected to that microcontroller.

3.5.2 LCD

A 16×2 LCD is RGB backlight is used to reflect the product names entered by the customer from keypad, which is also connected with ATmega328. Each numerical button on the keypad is assigned a unique product name, and this information is stored in ATmega328. After the microcontroller determines which button is pressed, it transfers the product name to the LCD. We tested the performance of LCD by connecting it with the microcontroller and check if the names sent from ATmega328 were accurately reflected on the LCD screen. Each name was tested three times and the responses of LCD were perfect. We also measured the time required to update a new name, which should be less than 0.5 s specified in the requirement table in Appendix A. It turned out that this time was less than 0.1 s and all requirements about LCD are satisfied.

3.5.3 ATmega328

ATmega328 is the main processor of Arduino and has most of its functionality. It only costs 1/6 of an Arduino but requires special circuit to operate. In the user interface module, ATmega328 serves as the information transit and processing center, connecting both the keypad and LCD, and also sends the shopping list to the main microcontroller of the project, Arduino. However, as mentioned previously, ATmega328 cannot generate arbitrary voltage output to represent different products. It has to use PWM waves to encode the information into different duty cycles. Then we add a low-pass filter to convert the duty cycle into voltages within 0 to 5 V, which Arduino can read from analog input pins. Finally, Arduino maps the input into numbers from 0 to 1023. We divided the entire range of Arduino into 10 intervals to represent all products. How each product sent from ATmega328 is received by Arduino is shown in Appendix D.3. This connection was examined by checking if the name received by Arduino was actually the name sent by ATmega328. By slightly adjusting the intervals, Arduino can 100% correctly receive the product names.

3.6 Microcontroller (Arduino)

As the main controller of project, Arduino controls the speed of motors, receives information from sensors and user interface, and sends cart position to customer through RF transceiver. After each module is tested independently, we wrote coded to combine all components with Arduino to see if it could make the right decision with respect to each condition. Arduino codes are included in Appendix E.

3.7 Ultrasonic Transmitter & Receiver

After we found that RF transceivers cannot be used to implement the positioning system, we quickly switched to ultrasonic transmitter and receiver because of the desirable speed of sound. Since ultrasonic transmitter and receiver in the market are usually assembled together to make ultrasonic sensors as the ones we chose, we need to build our own circuits that support bare transmitter and receiver. Circuit diagrams are shown in Appendix F.1 and F.2. Since the transmitter works better at larger source voltage, we amplified the 5 V output of Arduino using a “H-bridge”. The principle of the entire circuit is pretty similar to an ultrasonic sensor: we generated 40 kHz pulses using a inner timer of Arduino and read the amplitude of voltage at the receiver side to determine the distance. Two amplifiers in series in the receiver circuit are needed because of the small voltage it receives for large distances. We used an oscilloscope to watch the voltage waveforms sent and received by the circuits, respectively, whose results are included in Appendix F.3 and F.4.

Unfortunately, we failed to implement the positioning system with ultrasonic transmitter and receiver, either. One of the reasons is that the receiver can only receive reliable signals when it is in front of the transmitter since sound is not omni-directional. Another reason is that sound wave has significant reflections indoor and our system cannot distinguish if a signal is being reflected.

4. Costs

A list of parts used in our project is included in Table 4.1 and the labor cost for all group members is included in Table 4.2. The total cost of the project is $235.41 + $43200 = $43435.41.

4.1 Parts

Table 4.1 Parts Cost

Name	Part Number	Unit Price	Quantity	Total Price
Power Supply				
Energizer Max Alkaline Batteries	AA	$0.88	12	$10.49
Battery Holder - 4xAA Square	PRT-00552	$1.95	3	$5.85
Microcontroller				
Arduino Uno	DEV-1102	$29.95	1	$29.95
Atmega328	DEV-10524	$5.50	3	$16.5
User Interface				
Keypad 12-button	COM-08653	$3.95	1	$3.95
Basic 16x2 Character LCD - RGB	LCD-10862	$14.95	1	$9.99
Push Button Switch	COM-09177	$0.99	2	$1.98
Motor				
12V DC Gearmotor/Encoders	GM9236S025	$37.95	2	$75.9
Transceiver				
RFM12B-S2 Wireless Transceiver	WRL-09582	$6.95	4	$27.8
Ultrasonic Transmitter and Receiver	139492	$7.95	3	$23.85
1K Ohm Potentiometer	023-510	$1.79	1	$1.79
Sensor				
Ultrasonic Distance Sensor	HC-SR04	$5.59	4	$22.36
Miscellaneous				
Resistors, capacitors, inductors, transistors, wires, PCB and etc.			20	$5
Total				$235.41

4.2 Labor

Table 4.2 Labor Cost

Members	$/hour	Hours/week	Total hours	Subtotal	Subtotal * 2.5
Ying He	40	12	144	$5760	$14400
Di Fan	40	12	144	$5760	$14400
Xuyang Yao	40	12	144	$5760	$14400
Total					$43200

5. Conclusion

5.1 Accomplishments

Our project is successful in many aspects: we were able to precisely control the speed of motor; RF transceivers communicated fast and reliably; sensors detect obstacles accurately; user interface stored the shopping list and transferred to Arduino; and finally Arduino combines all information to develop the optimal route.

We were particularly proud of the algorithm we developed for optimal route determination. Given a fixed store layout, each shelf was labeled in a way such that the optimal path is automatically generated when the shelf number is sorted in ascending order. This requires the cart to start at a specific point each time, so we required to cart to return to the origin after it covered all products selected by the customer. Considering that shelves are usually arranged in a symmetric manner, sometimes there exists multiple optimal route to cover certain points. So we assign positive direction is each row such that the cart must rearrange itself in that direction when heading to another destination.

Even though we did not accomplish the indoor positioning system that helps the cart to locate itself and the customer, we found useful functions for the RF transceivers we bought. Based on the time resolution of Arduino (4 us) and speed of light, our plan should work when the distance between transmitter and receiver is larger than 150 m. We also had great success building our own ultrasonic transmitter and receiver circuit, which is a better option to implement the positioning system. With further improvements, they should ultimately achieve the goal.

5.2 Uncertainties

The biggest uncertainty of our project is the lack of PCB or vector boards. We tried to solder several vector boards but they all had severe short-circuited parts and failed to work. Our current design use only breadboards, which functions properly, but the jumper wires sometimes have poor connections. A significant disadvantage occurred when we built the ultrasonic transmitter circuit: the capacitance of breadboard is no longer negligible when operated at such high frequency as 40 kHz. The big time constant slowed down sharp transitions and caused the system to behave unexpectedly.

Another uncertainty is that in our design, sensors only check obstacles at specific time points, which means there is still a small chance of collision when the cart is moving. However, if we make the sensor check more frequently, it will take a significant amount of effort of Arduino to process that information and slow down the entire system. This problem needs to be addressed in future work.

5.3 Ethical considerations

Our project aims to build a shopping cart that provides great convenience and efficiency to customers in store. We are also concerned about general safety problems and the order of the entire store. Thus, we need to address several items in the IEEE Code of Ethics, which are presented in Table 5.1.

Table 5.1 Ethical Considerations

IEEE Code of Ethics	Our concerns
1. To accept responsibility in making decisions consistent with the safety, health, and welfare of the public, and to disclose promptly factors that might endanger the public or the environment;	We need to ensure that the future use of our project in stores will not contain any potential danger.
2. To avoid real or perceived conflicts of interest whenever possible, and to disclose them to affected parties when they do exist;	Our interest of shopping cart project has no conflict with any other existing object by research.
3. To be honest and realistic in stating claims or estimates based on available data;	We will be honest and realistic with TA and Instructor while giving explanation and presentation of our project.
4. To reject bribery in all its forms;	Bribery should not be a problem in the project.
5. To improve the understanding of technology; it's appropriate application, and potential consequences;	We will learn and improve our understanding of any project related knowledge.
6. To maintain and improve our technical competence and to undertake technological tasks for others only if qualified by training or experience, or after full disclosure of pertinent limitations;	We will try our best to make the project perfect by using the knowledge we have obtained by far.
7. To seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, and to credit properly the contributions of others;	We promise to guarantee that all the technical works are honest, and will admit the error if there is any. We will not take credits from other people's work.
8. To treat fairly all persons regardless of such factors as race, religion, gender, disability, age, or national origin;	We will treat all people fairly regardless of their race, religion, gender, disability, age, or national origin.

5.4 Future work

For future work, we will mainly concentrate on the implementation of the indoor positioning system. Now we already have the ultrasonic transmitter and receiver that are physically separated and can be used to measure the distance between two arbitrary points. Next, we need to handle the direction problem of sound wave by actively adjusting the direction in which the receivers face. Inspired by a group we peer reviewed, we realized this can be done by comparing the signal strength from all directions and picking the strongest signal, then we can turn the receivers that way. However, this method cannot distinguish if the signal is original or reflected. And we have three transmitters at that are operated at different times, which may also interfere with the reliability of the signal strength.

Another improvement we could achieve is to obtain the wheel speed automatically when the cart is moving. The rotary encoders we have now only have information about whether the cart is moving forward or backward. So we have to manually measure the speed from continuous tests. If the system can have this information, it could determine the time at each speed it needs to run and adjusts its position properly.

References

[1] "Arduino Playground: Keypad". Available: http://playground.arduino.cc/Code/Keypad/.

[2] "Simple Arduino and HC-SR04 Example". Available:

http://www.instructables.com/id/Simple-Arduino-and-HC-SR04-Example/.

[3] "LiquidCrystal – 'Hello World!'". Available: http://arduino.cc/en/Tutorial/LiquidCrystal.

[4] "A Sensitive DIY Ultrasonic Range Finder". Available:

http://www.kerrywong.com/2011/01/22/a-sensitive-diy-ultrasonic-range-sensor/.

[5] "RFM12b Library". Available: http://lowpowerlab.com/blog/tag/rfm12b/.

[6] "Arduino – Control a DC motor with TIP120, potentiometer and multiple power supplies". Available:

http://luckylarry.co.uk/arduino-projects/arduino-control-a-dc-motor-with-potentiometer-and-multiple-power-supplies/.