Intelligent Shopping Trolley (IST) System by WSN to Support Hypermarket IoT Service

You-Chiun Wang and Chang-Chen Yang
Department of Computer Science and Engineering
National Sun Yat-sen University
Kaohsiung, Taiwan, R.O.C.
Email: ycwang@cse.nsysu.edu.tw; m003040070@student.nsysu.edu.tw

***Abstract*—The Internet of Things (IoT) technology allows physical objects integrated with sensors to monitor and communicate with the outside world. The paper develops an Intelligent Shopping Trolley (IST) system to provide IoT service in a hypermarket. Each trolley is equipped with sensing modules to cooperatively monitor its customer's behavior. The sensing data is sent to a server through the ZigBee network to analyze the customer's preference. Then, the retailer can notify the customer of sales information about interesting products in real time. Also, a customer can query the trolley for product data such as where it is. Then, the system schedules an obstacle-free shortest path to guide that customer. The paper reports both system design and implementation experience. A prototype is also deployed in a real-life hypermarket to verify the feasibility of IST system.**

***Keywords*–customer behavior; indoor navigation; Internet of Things (IoT); shopping trolley; ZigBee.**

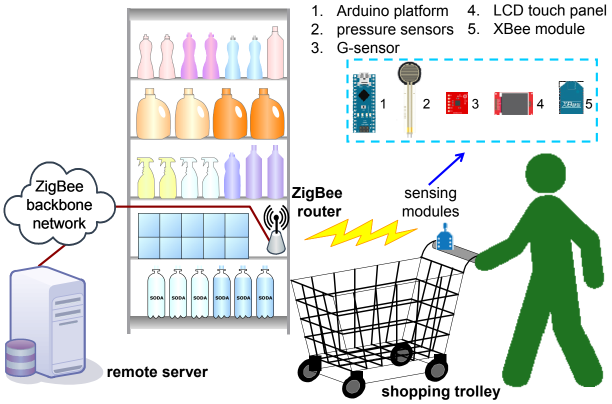


Figure 1. The architecture of our IST system.

I. Introduction

Today, Internet of Things (IoT) has attracted considerable attention from academic, industrial, and commercial communities. It is a burgeoning technology to let devices and physical objects connect with each other by their sensing, communicating, and processing capabilities. Wireless Sensor Network (WSN) is a critical technique of IoT [1], which uses many sensor nodes to jointly monitor the environment. WSN has various applications such as animal tracking, oceanic exploration, traffic control, and urban monitoring [2]. The great progress of WSN, together with other techniques like big data, social network, and cloud computing, promote the development of IoT [3].

This paper introduces IoT to hypermarkets by exploiting WSN's sensing ability, as shopping plays an important role in our economic activity. Many retailers use member cards to record the products sold to customers, and analyze their preference accordingly. However, this method is *offline*, in the sense that the retailer has no idea what products the customers may have interest (but did not buy) when they were shopping in the hypermarket. Some systems such as video surveillance could monitor a customer's activity [4]. However, it requires huge computation to conduct analysis [5]. Besides, such systems cannot provide real-time interaction with customers. Therefore, our goal is to develop an online, interactive system to detect the preference of customers and feed them back in real time, so as to assist customers when they are shopping.

Figure 1 gives the architecture of our Intelligent Shopping Trolley (IST) system by WSN to provide hypermarket IoT service. We equip a trolley with multiple sensing modules to monitor the behavior of its customer. Each product shelf is also placed with wireless routers. These routers, together with trolleys, connect with each other via the ZigBee protocol, and organize a *hybrid WSN* [6] in the hypermarket for communication and detection purposes. Specifically, when a customer is using the trolley, its sensing modules can monitor the customer's activities (e.g., is the customer walking or stopping?). The sensing data is sent to the remote server via a ZigBee router placed on the product shelf that the customer visits. Thus, the retailer can analyze the consumer's behavior and provide feedback to him/her about sales information. For example, when the trolley detects that its customer is stopping by some product shelf for a while, the server can quickly notify the customer of the promotion or discount for the products on that shelf. It can be realized by showing information on the trolley's Liquid Crystal Display (LCD) touch panel. In this way, the retailer could increase the opportunity that a customer is willing to buy the product, since the information of sales promotion is sent to the customer at the right time.

Moreover, the IST system also helps a customer find the desired product and guide him/her to the product's location. Specifically, conventional hypermarkets are operated by self-service. Customers search for the products on their own, place them in trolleys, and proceed to the checkout counters. However, without any assistance, finding the products may be time-consuming [7]. Although Global Positioning System (GPS) is a mature technology to guide people, it is hard to use GPS in a hypermarket due to satellite signal loss. Thus, we develop a lightweight guiding scheme modified from the popular A* algorithm [8]. Our scheme models the hypermarket by grids and computes the shortest path to help a customer detour obstacles and crowds, so that he/she can easily get the desired product. Consequently, we could improve the shopping experience of that customer.

Our contribution is to introduce WSN's sensing capability to common shopping trolleys, and apply IoT to the daily shopping in hypermarkets. The retailer can use our IST system to analyze customers' behavior in real time, and send various information like sales promotion, catalogue, or advertisement to customers to increase potential sales. Besides, IST can suggest the shortest, obstacle/crowd-free path for a customer to get the interesting product in an easy way. Both system design and implementation detail are presented in this paper. We will also develop a prototype and deploy it in the Carrefour hypermarket to demonstrate the proposed idea.

The rest of this paper is organized as follows. The related work is given in Section II. Section III proposes our IST system. Section IV details the hardware implementation. Our prototyping experience is described in Section V. Finally, Section VI concludes this paper and gives some future work.

II. Related Work

Several studies use image or signal processing to facilitate shopping. Tomizawa et al. [9] develop a shopping robot to purchase things for the people who cannot go by themselves to the hypermarket. A user can select the desired product and ask the robot to pick up that product for checkout. To do so, the robot recognizes the target object by a laser range finder and a camera, and uses its hand to grab the recognized object. In [10], volunteers wear electroencephalogram (EEG) caps and eye-tracking systems to do shopping. EEG helps neurologists to observe the brain activity, while the eye-tracking system allows them to see the volunteer's field of view. The data is used to measure the effect of marketing campaigns. Obviously, these studies have different goals with our work.

In [7], store items are tagged with Radio Frequency IDentification (RFID) to identify their positions. This data is registered to an information retrieval system. Then, customers can query the system to search and locate their desired items. Kitazawa et al. [11] design an agent-based in-store simulator to model the shopping behavior of customers. They equip market trolleys and shopping places with RFID tags. Then, purchased items are collected with the point of sales information gathered at the cash desk. The shopping paths and purchase data can be fed as parameters of the simulator to improve its performance. However, [7] and [11] aim at back-end management (i.e., item administration or simulation), rather than front-end management to monitor the behavior of customers.

Some work uses RFID to guide shoppers. In [12], customers carry public cards for identification and mobile phones for navigation. As products are tagged with RFID, each customer can query the product data (e.g., name, manufacturer, and price) via the phone. The system also guides a customer to get the interesting product by sending a route to the phone. Then, the customer can use the card to do checkout. The work of [13] aims at assisting visually impaired people in shopping. To do so, RFID is used to identify products, and people are navigated by audio instructions to obtain products. Besides, an autonomous billing system is built by using ZigBee to send the information. However, neither [12] nor [13] consider analyzing customers' preference and proving them feedback.

Using sensors to enhance shopping trolleys is also discussed. In [14], trolleys are equipped with sensors to record their moving distances and orientations, so as to track customers. Besides, when detecting obstacles, the trolley's velocity can be slowed down for safety. Rupanagudi et al. [15] attach a camera on each trolley to guide a customer based on the shopping list. To simplify image processing, the hypermarket is mapped with only three colors (i.e., blue, light orange, and dark orange). Then, each trolley identifies the color read from its camera to estimate the position. However, these studies do not use trolleys to analyze customers' preference on products.

On the contrary, our work discusses how to apply IoT to hypermarkets through WSN-based trolleys. By detecting the activity of a customer related to the trolley, we can analyze the preference of that customer. Moreover, an efficient guiding scheme is developed to help customers detour obstacles and beat the crowds, so as to improve their shopping experience. These features distinguish our work from others.

III. IST System Design

This section presents the IST system for hypermarket IoT service. We first give the whole picture of our system. Then, we discuss how to analyze customers' behavior via shopping trolleys, and guide them in the hypermarket.

A. System Architecture

Figure 1 presents the IST architecture with three components: *shopping trolleys*, *ZigBee routers*, and a *remote server*. Through ZigBee communication, they form a hybrid WSN in the hypermarket. Specifically, each trolley is a *mobile sensor* that moves in the WSN and produces sensing data, while ZigBee routers act as *static sensors* to provide connectivity and relay messages [16]. Then, the server is a *sink node* to gather data from trolleys and send commands to the network. Below, we detail the function of each component.

1) Shopping Trolley: A trolley plays the role of *user interface* for each customer. It detects the customer's movement and checks if he/she is griping the trolley's handle. Such data is sent to the remote server to analyze the preference of that customer. Based on the analysis result, the server sends relevant information (e.g., sales promotion or activities) to the customer. Moreover, the customer can query the location of desired product via the trolley, and our system then schedules a walking path to help the customer detour obstacles and crowds.

To do so, each trolley has five sensing modules shown in Figure 1. First, the *Arduino platform* works like a motherboard to connect all modules. Specifically, both pressure sensors and G-sensor periodically send their sensing values to Arduino. It also establishes full-duplex links with the LCD touch panel and XBee module to interact with the customer and connect with ZigBee routers, respectively. Besides, Arduino processes input and sensing data, deals with the server's commands, and sends its computation results to other modules for action. Thus, Arduino is also viewed as the main controller of a trolley.

Second, two *pressure sensors* are installed on the trolley's handle to detect whether the customer is using the trolley. Such a sensor can dynamically change its electric resistance based on the pressure exerted on it. When the customer is griping the handle, he/she is exerting a large amount of pressure to the sensor, which results in resistance decrease and voltage increase. Thus, by detecting the voltage change, the trolley can determine whether the customer is griping (and using) it.

Third, the *G-sensor* detects the acceleration G_X, G_Y, and G_Z on the X, Y, and Z axes, respectively. When the trolley stops, its G-sensor reports that $G_X = 0$, $G_Y = 0$, and $G_Z < 0$ (i.e., the acceleration direction on the Z axis is toward earth). When the trolley is moving, we have $G_X \neq 0$ or $G_Y \neq 0$, depending on the direction. With this property, the G-sensor can determine the status of a trolley (i.e., moving or stopping), and also estimate its moving direction.

Fourth, the *LCD touch panel* is an I/O device for a customer to interact with the trolley. The customer can use LCD to search for desired products. In addition, when the server replies the sales promotion or suggested path to the trolley, this information will be shown on LCD.

Finally, the *XBee module* provides communication for a trolley. It follows the IEEE 802.15.4 standard [17] to support low-power transmission ($<$2mW), and adjusts transmission power to provide different range (40$\sim$120m). The trolley uses its XBee to join the hypermarket WSN and exchange messages with a nearby ZigBee router.

2) ZigBee Routers: We place each product shelf with ZigBee routers to provide full coverage [18] of the hypermarket. A router is given one unique ID (e.g., MAC address) for identification. Once a router finds that a trolley is stopping at its shelf, it can notify the server that the customer may have interest in some products. Besides, all routers organize themselves into WSN backbone for communication. A trolley can report its sensing data to the server via the visiting router. The server also uses the backbone to send commands to an individual trolley, or broadcast information to the WSN. We use the Ad-hoc On-demand Distance Vector (AODV) protocol [19] to route packets in the network. Moreover, the routers help estimate the positions of trolleys. So, our system can track every trolley in the hypermarket.

3) Remote Server: The server is the decision center of our system. It connects to the retailer's database to access product information (e.g., prices, discounts, and activities). The server has four missions. First, it checks the health of all components. When there is something wrong with a component, the server notifies an employee to examine that component. To do so, the server records for each component the most recent message sent from it. If a component did not send any message for a threshold time (e.g., 1 hour), the server sends a Hello message to that component and checks if any response is returned. After several failed tries (e.g., 3 times), the server marks that component as broken, and sends a notification to employees. The duration of two successive Hello messages can be set to, for example, 1 minute.

Second, when a trolley transmits its sensing data to the server, the server will analyze the customer's behavior accordingly. The analysis is conducted based on the combination of two actions: 1) the customer is griping the trolley's handle and 2) the trolley is moving. Such analysis result can be recorded in the retailer's database. For example, the retailer can count the number of times that each kind of products have ever been reviewed by customers. Then, the retailer can gather statistics for the reference to adjust its marketing strategy. The analysis is online, which means that the retailer can observe the preference of customers when they are shopping in the hypermarket. Then, the retailer can find out potentially popular products. Moreover, since each trolley reports only its status and there is no linkage between the customer's personal data and that trolley, we can protect the privacy of customers.

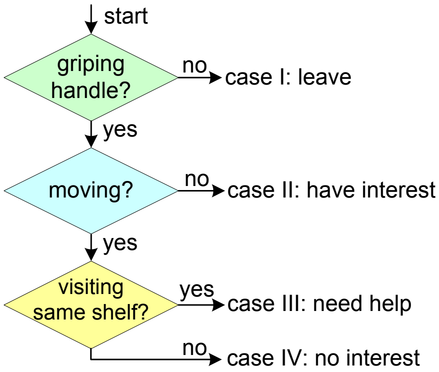


Figure 2. The flowchart to determine the behavior of a customer.

Third, if the server finds that a customer has interest in some product (e.g., he/she stops by a product shelf for a while), it can refer to the database to see if there is sales promotion for the product. If so, the server can unicast such information to the trolley via AODV. Moreover, the server can announce messages to a subset of nodes in WSN. In this case, the server can construct a multicast tree [20] to send its data more efficiently.

Finally, as ZigBee routers will estimate the position of each trolley, it allows the server to keep track of the distribution of trolleys in the hypermarket. Then, once a customer queries a walking path to get the desired product, the server can take such distribution as reference to find a shortest path to detour obstacles and beat the crowds. In Section III-C, we will discuss how to efficiently find such a path.

B. Determining Behavior of Customers

Without using cameras to monitor the behavior of customers, which may involve in complex image processing, we adopt both pressure sensors and G-sensor equipped on each trolley to conduct the monitoring job.

Figure 2 gives the flowchart to determine the behavior of a customer by the trolley. In each case, the remote server can feed back some information to the customer, which is shown on the trolley's LCD panel. Below, we list the four cases:

I. The customer may leave to do something (e.g., looking for products). In this case, the relationship between customer and trolley is temporarily broken. When the customer returns, the trolley can show the advertisement of nearby products to draw his/her attention. However, if the trolley's handle is not gripped for a long time (e.g., ≥ 30 minutes), we treat the trolley as unused, so the server will notify an employee to retrieve that trolley.

II. There is a high possibility that the customer is checking some product, which means that he/she may have interest in that product. Thus, the server can send the customer sales promotion or special activity (according to the retailer's database), so as to increase the purchase intention.

III. The customer has visited the same product shelves multiple times. It means that the customer is searching for some products and may require help. So, the server can ask nearby employees to assist that customer.

IV. The customer may be walking aimlessly and have no interest in nearby products. Therefore, we can show the product catalogue for his/her reference.

When the trolley detects that the customer has interest in certain products, the server can also record the statistics for these products in the database. It helps the retailer to adjust sales policy or product arrangement in the future.

C. Navigating Customers in the Hypermarket

There are *fixed* and *movable* obstacles in a hypermarket. Fixed obstacles include walls and product shelves, while movable obstacles are crowds (i.e., other trolleys). A customer cannot cross fixed obstacles, but he/she may pass through a region with only one or two trolleys. To help a customer detour obstacles and fast get the desired product, we model the hypermarket into grids, where the grid length allows two trolleys to pass through. Since each trolley periodically reports its position to the server, the server can keep track of the number of trolleys in each grid. Then, we define a *crowded grid* to be a grid with more than two trolleys. It is difficult for the customer to move the trolley through such grids.

Given the customer's location (denoted by grid g_s) and the destination (denoted by grid g_e), we modify the A* algorithm to find an obstacle-free shortest path. Specifically, we compute the cost to reach each grid g_k by:

$$\hat{C}(k) = \hat{A}(k) + \hat{M}(k), \tag{1}$$

where $\hat{A}(k)$ is the accumulative cost from g_s to g_k, and $\hat{M}(k)$ is the minimum cost from g_k to g_e. We define the cost of a crowded grid and a grid with fixed obstacle to ∞. Thus, the walking path will not contain such grids, and the customer can detour obstacles and crowds. Then, we conduct the following steps to find the shortest path:

1. Let sets L_C and L_S contain candidate and selected grids, respectively. Initially, $L_C = \{g_s\}$ and $L_S = \emptyset$.
2. Select the grid g_k from L_C that has the minimum cost $\hat{C}(k)$. However, if no grid can be found, there is no path between g_s and g_e. So, the scheme terminates.
3. We calculate the cost of each adjacent grid by (1). If the grid with a finite cost is not included in L_S, we add it to L_C. Then, we view g_k as the starting grid, remove it from L_C, and add it to L_S.
4. Steps 2 and 3 are repeated until we select g_e from L_C. Then, we add g_e to L_S.
5. The walking path is included in L_S. However, it may contain some redundant grids. Therefore, we traverse all grids in L_S to obtain the final (shortest) path.

Notice that trolleys will dynamically move in the hypermarket. When a customer is walking along the suggested path but some grids on the residual path become crowded, we can adaptively compute a new path by the above scheme for the customer.

IV. Hardware Implementation

In this section, we discuss the hardware implementation of IST. Figure 3 shows the sensing modules installed on a shopping trolley. We use Arduino Mega2560 [21] to coordinate all modules, which has a 16MHz ATmega2560 microcontroller, 8KB Static Random Access Memory (SRAM), 4KB Electrically Erasable Programmable Read Only Memory (EEPROM), and 256KB flash memory. Its operating voltage is 5V, and the input voltage is 7∼12V. Arduino also supports multiple protocols for IC communication, such as Inter Integrated Circuit (I^{2}C) and Serial Peripheral Interface (SPI).

Arduino uses 54 digital I/O pins and 16 analog pins for communication, where the direct current for an I/O pin and a 3.3V pin is 40 and 50mA, respectively. Figure 4 shows the circuit connections between Arduino and other modules.

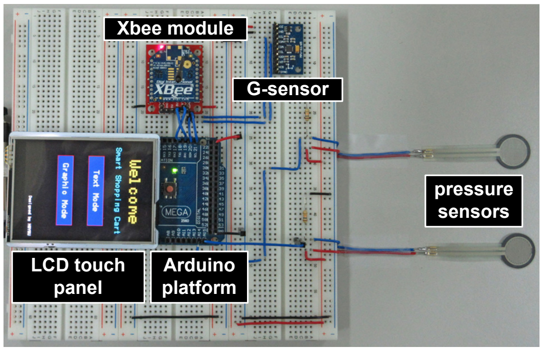
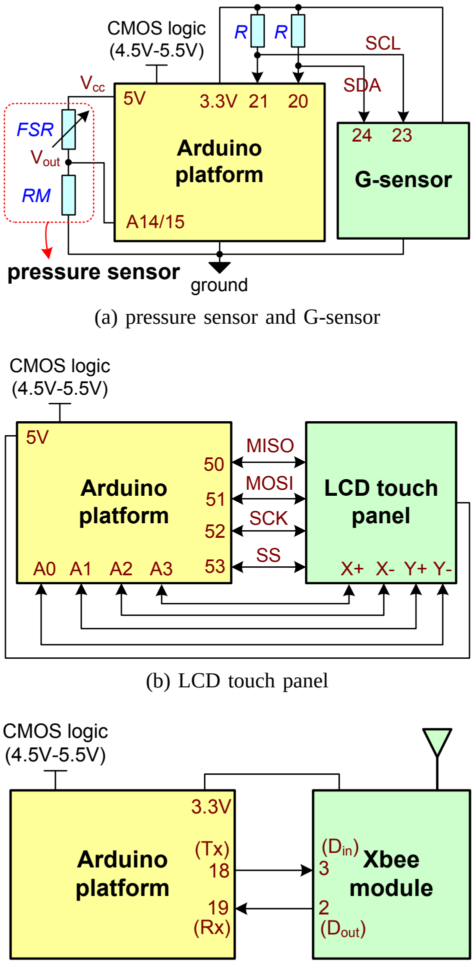


Figure 3. The sensing modules installed on a shopping trolley.



(a) pressure sensor and G-sensor

(b) LCD touch panel

(c) XBee module

Figure 4. The circuit connections between the Arduino platform and other modules, where 'CMOS logic' denotes the external circuit.

Specifically, Arduino supports a *reference voltage* V_{cc} to the two pressure sensors, and uses A14 and A15 analog pins to get the *output voltage* V_{out} from them, as shown in Figure 4(a). The operating principle of a pressure sensor is the *voltage divider rule* [22]:

$$V_{out} = \frac{RM}{FSR + RM} \times V_{cc}, \qquad (2)$$

where FSR and RM are changeable and fixed resistances, respectively. When a customer gives a force to the pressure sensor, FSR will decrease. Based on (2), it results in the increase of V_{out}. Then, by reading V_{out} from A14 and A15 pins, Arduino can detect whether the customer is holding the trolley. In the implementation, we use Sparkfun FSR402 force-sensitive resistor [23] to be the pressure sensor, where $FSR = 1\text{M}\Omega$ (initially) and $RM = 10\text{K}\Omega$ $(\pm 0.25\%)$.

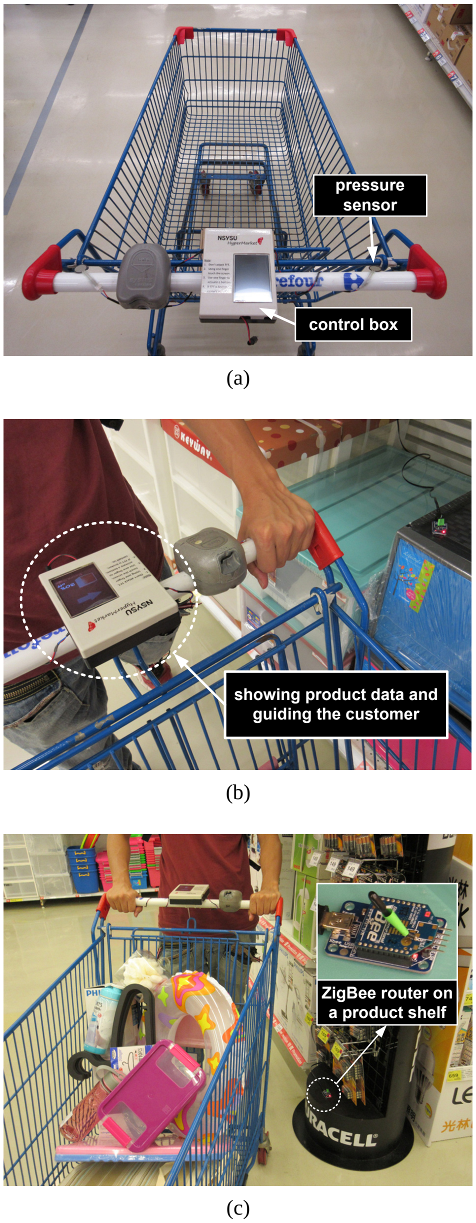
Figure 4(a) also shows how to connect Arduino with a G-sensor, where 'R' denotes a resistance. G-sensor is used to detect the trolley's movement by estimating the acceleration along three axes. To obtain the estimation, Arduino uses its 20th and 21st pins to read the data from both Serial Data Line (SDA) and Serial Clock Line (SCL) of the G-sensor, respectively. The communication between Arduino and G-sensor follows the I^2C protocol. Besides, Arduino supports 3.3V operating voltage to trigger the G-sensor. In our implementation, we adopt InvenSense MPU9250 accelerometer [24] as G-sensor.

Each trolley has an LCD touch panel to interact with its customer. Here, we use Thin Film Transistor (TFT) touch shield [25]. It has a 2.8-inch display with 240×320 resolution and 18bit colors. The display is a 4-wire resistive touch screen, and has a visual angle of $60^\circ \sim 120^\circ$. The shield has an ILI9341 controller with built-in RAM buffer, and 8bit digital interface plus 4 control lines. Arduino uses the SPI protocol to communicate with LCD, where Arduino serves as the master while LCD acts as the slave. To instruct LCD, Arduino uses its 50th, 51st, 52nd, and 53rd pins to respectively connect LCD's controlling pins: Master Input Slave Output (MISO), Master Output Slave Input (MOSI), Serial ClocK (SCK), and Slave Select (SS). LCD uses four data pins, $\text{X}+$, $\text{X}-$, $\text{Y}+$, and $\text{Y}-$, to report the position where the customer touches the screen. Thus, Arduino connects these pins via its A0, A1, A2, and A3 analog pins to get the data, as shown in Figure 4(b).

Through the XBee module, Arduino can communicate with a nearby router. We use XBee Series2 [26] in our implementation. It provides a transmission distance of 40m and 120m indoors and outdoors, respectively. The supply voltage is 2.1∼3.6V, and the transmission power is 2mW. XBee operates on the 2.4GHz band, and supports point-to-point, star, and mesh topologies. The receiver sensitivity is -96dBm and the data rate is 250Kbps. We use two XBee pins for communication: *data out* (D_{out}, the 2nd pin) and *data in* (D_{in}, the 3rd pin). Thus, Arduino uses the 18th and 19th pins to connect with XBee, as shown in Figure 4(c).

V. Prototyping Experience

We demonstrate our IST system in Carrefour, as shown in Figure 5. All sensing modules are encapsulated in a control box for easy deployment, and we entwine pressure sensors on the trolley's handle to detect the usage by a customer, as illustrated in Figure 5(a). A customer can push the trolley to do shopping in the hypermarket. Then, the trolley will not only show the product-discount information when the customer stops to check products, but also navigate the customer via an arrow sign, as presented in Figure 5(b). Besides, when the customer moves close to a product shelf, the trolley can exchange the data with a ZigBee router, as shown in Figure 5(c). Therefore, the server can obtain the up-to-date position of that trolley. Through this way, we can apply the IoT technology to our daily shopping in hypermarkets.


pressure sensor
control box
showing product data and guiding the customer
ZigBee router on a product shelf

(a)

(b)

(c)

Figure 5. IST system deployment in Carrefour: (a) encapsulate all components into a control box, (b) the LCD touch panel shows the product data and guide the customer (by the moving direction), and (c) the shopping trolley communicates with the ZigBee router placed in a product shelf.

Moreover, we develop a Java-based interface at the server to calculate the walking path for each customer, as Figure 6 shows. The layout of product shelves refers to Carrefour and we divide the hypermarket into 22×15 grids. A customer can select two non-obstacle grids to be g_s and g_e, which are

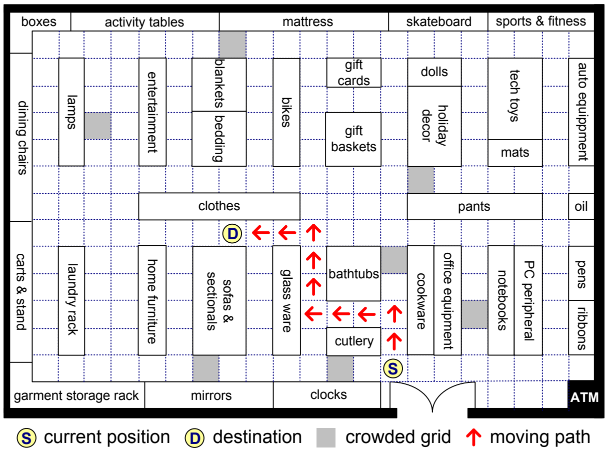


Figure 6. An interface (at the server) to show the walking path for a customer.

respectively denoted by ‘S’ and ‘D’. Crowded grids (i.e., ≥ 3 trolleys) are marked by gray, and our scheme will then prevent the customer from passing through these grids. Finally, the walking path is shown on the interface by arrows.

VI. Conclusion and Future Work

Many retailers use member cards to record the preference of customers, but the method is offline and non-interactive. Therefore, this paper embeds WSN’s sensing ability in common shopping trolleys, and brings the IoT technology to our daily shopping. Our sensor-based trolley can monitor the behavior of each customer when he/she is shopping in a hypermarket. The retailer can thus quickly provide necessary information to the customers who have interest in some products. Besides, we also develop a navigation scheme from A* to help customers easily get their desired products. Both the designing architecture and implementation detail of our IST system are reported. Furthermore, we also test a prototype in Carrefour to demonstrate its practicability.

For the future direction, we will integrate IST with some existing systems such as visual surveillance or RFID to improve the accuracy of monitoring customers’ behavior and preference. In addition, we can apply the energy harvesting technique to extend the system lifetime. One possibility is to obtain energy from the trolley wheel’s movement.

Acknowledgements

You-Chiun Wang’s research is co-sponsored by the Ministry of Science and Technology under Grant No. MOST 104-2221-E-110-036-MY2 and MOST 104-2628-E-110-001-MY2, Taiwan.

References

[1] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, “Internet of Things: a survey on enabling technologies, protocols, and applications,” IEEE Communications Surveys & Tutorials, vol. 17, no. 4, 2015, pp. 2347–2376.

[2] Y. C. Wang, “Mobile sensor networks: system hardware and dispatch software,” ACM Computing Surveys, vol. 47, no. 1, 2014, pp. 12:1–12:36.

[3] F. daCosta, Rethinking the Internet of Things: A Scalable Approach to Connecting Everything. New York: Apress, 2013.

[4] W. Lao, J. Han, and P. H. N. de With, “Automatic video-based human motion analyzer for consumer surveillance system,” IEEE Transactions on Consumer Electronics, vol. 55, no. 2, 2009, pp. 591–598.

[5] Y. C. Tseng, Y. C. Wang, K. Y. Cheng, and Y. Y. Hsieh, “iMouse: an integrated mobile surveillance and wireless sensor system,” IEEE Computer, vol. 40, no. 6, 2007, pp. 60–66.

[6] Y. C. Wang, “A two-phase dispatch heuristic to schedule the movement of multi-attribute mobile sensors in a hybrid wireless sensor network,” IEEE Transactions on Mobile Computing, vol. 13, no. 4, 2014, pp. 709–722.

[7] D. Hicks, K. Mannix, H. M. Bowles, and B. J. Gao, “SmartMart: IoT-based in-store mapping for mobile devices,” in International Conference on Collaborative Computing: Networking, Applications and Worksharing, 2013, pp. 616–621.

[8] H. Wang, J. Zhou, G. Zheng, and Y. Liang, “HAS: hierarchical A-star algorithm for big map navigation in special areas,” in International Conference on Digital Home, 2014, pp. 222–225.

[9] T. Tomizawa, K. Ohba, A. Ohya, and S. Yuta, “Remote food shopping robot system in a supermarket–realization of the shopping task from remote places,” in IEEE International Conference on Mechatronics and Automation, 2007, pp. 1771–1776.

[10] S. F. Sands and J. A. Sands, “Recording brain waves at the supermarket: what can we learn from a shopper’s brain?” IEEE Pulse, vol. 3, no. 3, 2012, pp. 34–37.

[11] M. Kitazawa, M. Takahashi, T. Yamada, and T. Terano, “Analyzing supermarket shopping paths from indirect observation and simulation study,” in IEEE International Conference on Signal-Image Technology & Internet-Based Systems, 2013, pp. 939–943.

[12] R. Chen, L. Peng, and Y. Qin, “Supermarket shopping guide system based on Internet of things,” in International Conference on Wireless Sensor Network, 2010, pp. 17–20.

[13] M. Mathankumar and N. Sugandhi, “A low cost smart shopping facilitator for visually impaired,” in International Conference on Advances in Computing, Communications and Informatics, 2013, pp. 1088–1092.

[14] S. Gai, E. J. Jung, and B. J. Yi, “Localization algorithm based on Zigbee wireless sensor network with application to an active shopping cart,” in IEEE International Conference on Intelligent Robots and Systems, 2014, pp. 4571–4576.

[15] S. R. Rupanagudi et al., “A novel video processing based cost effective smart trolley system for supermarkets using FPGA,” in International Conference on Communication, Information & Computing Technology, 2015, pp. 1–6.

[16] Y. C. Wang, F. J. Wu, and Y. C. Tseng, “Mobility management algorithms and applications for mobile sensor networks,” Wireless Communications and Mobile Computing, vol. 12, no. 1, 2012, pp. 7–21.

[17] IEEE Std 802.15.4-2011, “IEEE standard for local and metropolitan area networks–part 15.4: low-rate wireless personal area networks (LR-WPANs),” 2011.

[18] Y. C. Wang, C. C. Hu, and Y. C. Tseng, “Efficient deployment algorithms for ensuring coverage and connectivity of wireless sensor networks,” in IEEE Wireless Internet Conference, 2005, pp. 114–121.

[19] C. E. Perkins and E. M. Royer, “Ad-hoc on-demand distance vector routing,” in IEEE Workshop on Mobile Computing Systems and Applications, 1999, pp. 90–100.

[20] X. Xiang, X. Wang, and Y. Yang, “Supporting efficient and scalable multicasting over mobile ad hoc networks,” IEEE Transactions on Mobile Computing, vol. 10, no. 4, 2011, pp. 544–559.

[21] Arduino, http://www.arduino.cc/, retrieved: June 2016.

[22] C. K. Alexander and M. N. O. Sadiku, Fundamentals of Electric Circuits. New York: McGraw-Hill Education, 2012.

[23] Sparkfun, https://www.sparkfun.com/, retrieved: June 2016.

[24] InvenSense, http://store.invensense.com/, retrieved: June 2016.

[25] Adafruit, http://www.adafruit.com/, retrieved: June 2016.

[26] Digi International, http://www.digi.com/, retrieved: June 2016.