G54MRT Coursework 2 Final Report

Title: The Unit Monitor for Warehouse Inventory Management

Student ID: 14338557
Date: May 2019

1. Summary

Typically, warehouse consists of many standard storage units and it stores items with standard size. The project was designed for managing inventory in these warehouses, which would be able to identify the typical actions (Put-In/Put-Out), the current stock statuses (Shortage or Normal), and the number of items in each unit. There are 3 sensors in each unit: 1 ultrasonic sensor, 1 PIR motion sensor, and 1 button sensor, and the distance detected by the ultrasonic sensor would be used as the core value in this project.

2. Background and motivation

I was inspired by a project that I experienced before, which is about designing the interface of a smart warehouse management system, based on which, I conducted some further table research recently and got the findings that, for these kind of warehouse inventory systems:

1) **Specific Settings:** Standardisation is the main feature of this setting. Storage units is standard (e.g. storage girds in the shelf), and generally the goods are supposed to have the same size, which provides good conditions for sensor detection.

2) **Common Needs:** Presenting the stock status and in-out actions are the core functions of these systems; both the purchasing department and sale departments attach much importance on them for further measurements.

3) **Current Challenges:** Underlying these systems, a majority of warehouses are manually checked and recorded by staffs, which are time-consuming and easy to make mistakes, not to mention that they have to re-check them frequently for confirmation; some other existing smart systems requires smart storage facilities to complete the same process which are money-consuming and might not appropriate for relatively small-sized companies to use.

Therefore, in this project I aimed to accomplish these core functions for each storage unit with cheap and sensitive sensors. It would be smarter, cheaper and particularly, easier to be deployed in existing storage facilities, so it might bring great values for inventory management.

3. Related work

1) The lectures and papers regarding to UniComp system design enabled me to understand how to design seamlessly for specific contexts and how to interpret the sensor data for delivering valuable information. The testing process was also developed from them.

2) The project that I used to participate in about smart warehouse allowed me to understand the setting, recognise needs/challenges and typical scenarios and contributed to my system design framework. (see 2.)

3**)** The existing product Siemens Warehouse System inspired me to think about how to identify the number of goods by collecting and transferring other value in a simpler way. Based on online research, it was often mentioned that to record the Put-In and Put-Out behaviours by scanning the codes in all changing items, which rendered me to consider to use other sensors(e.g. button/motion/NFC tag) to achieve the same.

4**)** Finally, the technology framework for sensing was based on the understanding of the feature of the sensors and projects, in which the application and accuracy of ultrasonic sensors are demonstrated: can detect all types of surface materials and has a relatively high accuracy (1cm) in the range of 2cm-10m (Liu 2015; Köhler 2013). Thus I considered to transfer the distance information to the stock status by the ultrasonic sensor, as well as taking the limitation of its accuracy and testing later.

4. Design

4-1 User Cases

1) **Purchasing Staff:** Who wants to check which items/units are **Shortage**, and then to purchase more these items to **Put-In** their located stock units.

2) **Pick-up Staff:** Who wants to check if the needed items are **Well-Stocked** as well as their located stock units, and then to **Pick-Up** the items.

3) **Warehouse Manager:** Who inputs the value of the standard **Unit-Size** and **Items-Size** with descriptions; wants to inspect the recording of **Put-In** and **Put-Out** actions.

4-2 Logic of a Stock Unit

1) The number of items

- As a stock unit with the standard size (unit_size), generally there is only one single type of items in it. For each storage unit, managers of this system can customise the height for the single item (item_size) as the default setting, so the number of items can be detected by the distance data specifically. Distance will be detected by the Ultrasonic Sensor (ultra_distance).

2) Current Stock Status: 'Shortage', 'Well-Stocked'

- The warehouse manager can set the number for the boundary of 'Shortage', based on the detected items in the unit. Another case is the item is not available to detected the number by a fixed height.In this case, the number won't be displayed, and the 'Shortage' status will be displayed when the remaining space (ultra_distance) is less than a specific value.

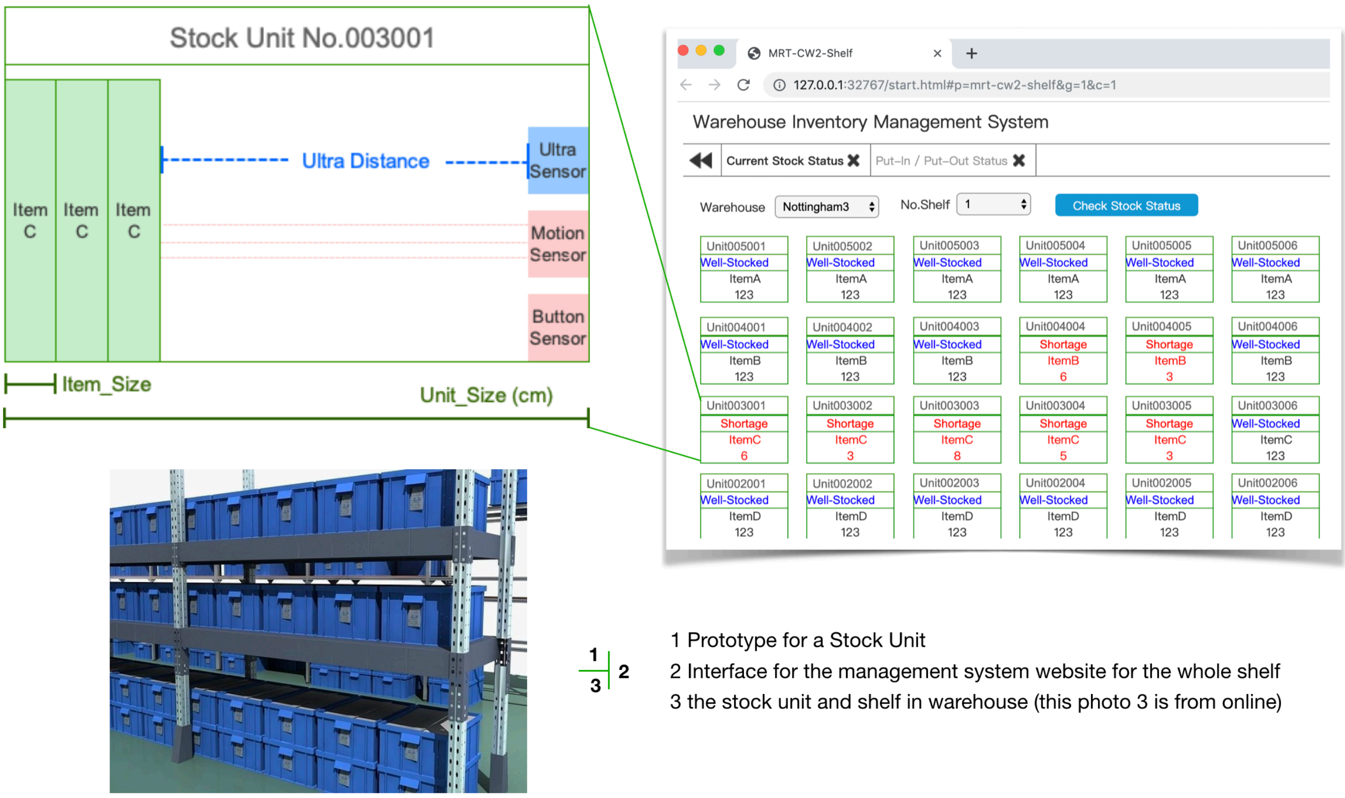
- All units with the current stock numbers and status will be displayed together with corresponding location in the monitoring webpage (the warehouse background management system). This page can inform staffs about the whole inventory status and help them to take further actions.

- When the unit is in the 'Well-stocked Status', this unit will display the current number of storing items with normal colour. When the unit is in the 'Shortage Status', this unit will be red highlighted in the monitoring webpage and the manager will receive a notification for reminding replenishment.

3) Three behaviours: Put-In, Put-Out, No Change

- These two behaviours will be justified by the Positive-Negative Change of the distance detected by Ultrasonic Sensors with the assistance of Motion Sensor or button sensors.

4-3 Whole System Architecture



1 Prototype for a Stock Unit
2 Interface for the management system website for the whole shelf
3 the stock unit and shelf in warehouse (this photo 3 is from online)

5. Implementation. 20'

5-1 What implemented & Real Model

The stock unit is the core uniComp part built by our sensors within this smart warehouse system. In this functional prototype, the main concerns were given to its core functions: **A.**identify the stock statuses; **B.**identify the number of items; **C.**identify the In-Out actions. In order to implement it quickly and simply, I set the default input values based on the reality of my prototype, rather than allowing users to customise this value; Additionally, I didn't output the real texts about the status; instead, I used 0/1/2 to represent the different status / actions for easily outputting graph of our data later.

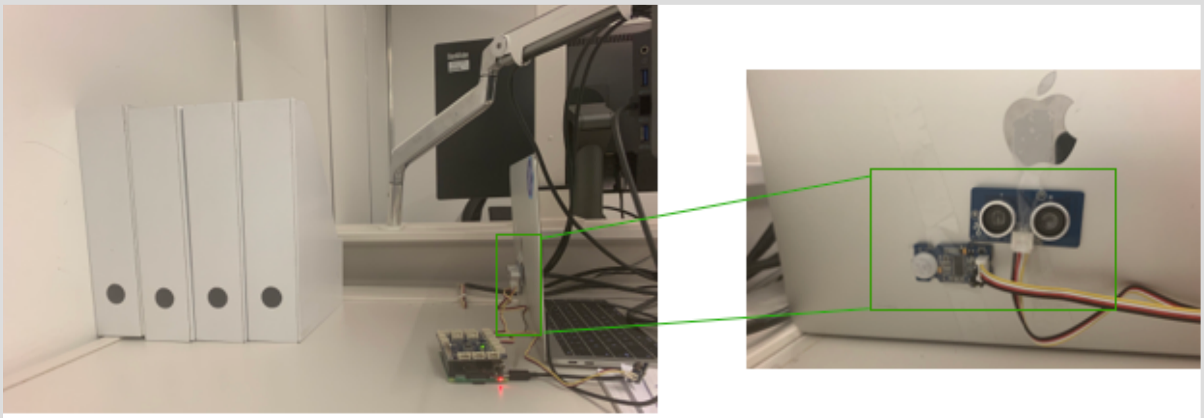


Figure 4 The Functional Prototype for a Stock Unit

5-2 How implemented

My codes are responded to this simulated unit and the item paper box that I use.

1) **Choose the sensors:** Decision for ultrasonic sensors is based on its possibility to reflect the occupied space in the unit, which is undoubtedly. Both of PIR motion & button sensors are deployed in order to simulate the Out-In action and then judge the put-in/out behaviour. For button sensor, logic is to simulate the situation: ①open door; ②put-in/out action; ③close the door and finish this action. Therefore, first click(1) = open the unit door; Second click(1) = close the unit door; the change of items between them will be calculated later. For PIR sensor, logic is simple: move(1) means action is conducting, when it becomes no-move(0) again, means this action is completed.

2) **Median Filter for Ultrasonic Sensor**: This system doesn't attach importance to the variating process of distance but to the steady value it can get to detect the distance. In the initial tests, mostly its raw data are steady with rarely unusual values, I decided to implement a median filter for the ultrasonic sensor, which will return a steady value (Non-Linear data is what I want in this situation), aiming to remove the outliers and meaningless fluctuation.

3) **Counting the number:** Ultrasonic sensor was fixed in the side of the unit and the distance will represent the width of remained space. As the number of items have to be integer (N=1,2,3…) and we tended to ignore the tiny inaccuracy of value caused in reality, the **round()** has been used for getting the nearest integer of this equation. Ideally, **n_items = round ((unit_size - ultra_distance) / item_size).**

4) **Identifying the status & action:** The stock status is a following judgement of current item number, and the action will be judged by the value's positive or negative.

```
#####################################################
# Judge the number of items
n_items = round((unit_width - ultra) / item_width)

# Judge the stock status
if n_items < 0:
    unit_status = 2  # 'Abnormal'
else:
    if n_items<=1:
        unit_status = 1  # 'Shortage'
    else:
        unit_status = 0 # 'Well-stocked'
```

```
#####################
# Use Filter Data of Ultrasonic (Median)
if changeB > 0 :
    behaviorB = 1 #'Put-In'
elif changeB < 0 :
    behaviorB = 2 #'Put-Out'
else: # changeB = 0
    behaviorB = 0 #'No-Change'
```

6. Testing:

There are three rounds of testing with the same prototype with unit_size=51cm and item_size=7cm. I tested the function by simulating the action ‘put items in and put items out’ for several times, and different numbers of changing each time. I set the ‘Shortage’ status to ‘less than 2’ to observe the status.

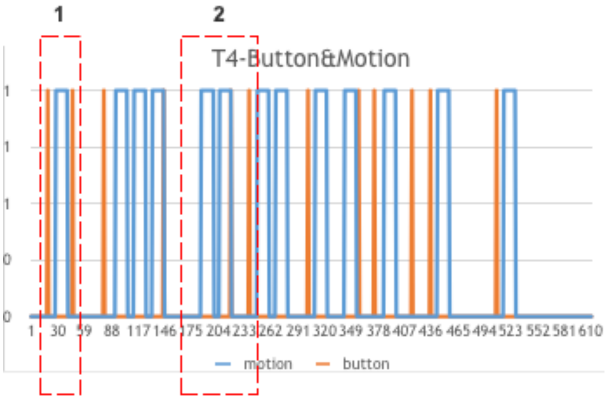


Figure 5

```
history_ultra.append(ultra)
ordered_ultra = sorted(history_ultra)
ultra_median = history_ultra[int(len(ordered_ultra) / 2)]
```

In Test Stage 1, I supposed to use button sensor to identify Put-In&Out actions, and calculate the changes before and after the paired button click, while the system failed to achieve it. I reviewed the raw data, finding that: button sensor didn’t record the real clicking behaviour precisely. To be more specific, in Figure 5, the data slide 1 is a correct typical detection, the movement will be located within the paired ‘1-1’ (between open and close door period), while the data slide 2 is error one due to the missed record of one click. Thus, I decided to remove the button sensor temporarily, but use PIR sensor with new logic and a mean filter. By mean filter, I aim to transfer the non-linear motion points to a continuous line, reflecting the ‘action period’ and ‘no-action period’. Additionally, after the initial tests, the 1cm was recognised to subtracted in the equation due to the distance caused by the height of ultrasonic itself (based on initial test).

n_items = round ((unit_size - 1- ultra_distance) / item_size)

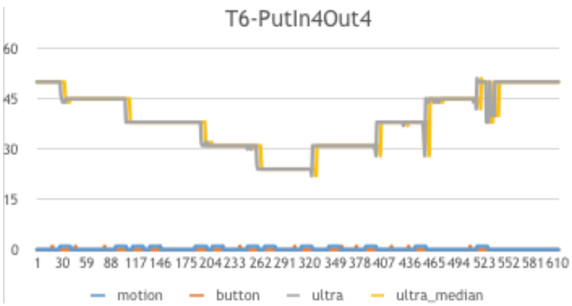


Figure 6

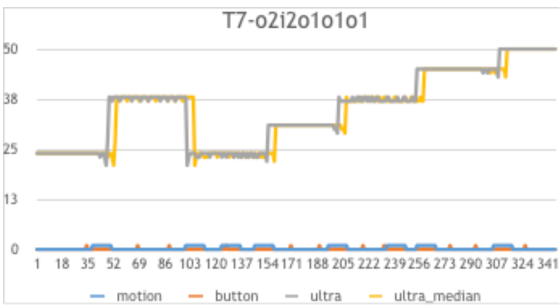


Figure 7

In Test Stage 2, the parallel tests were conducted to get raw data and filtered data to compare the effect of filter. I used to set the collection number for median is 11, while from Figure6 and Figure7 I noticed that this setting made this filter became useless in this scenario. So I decided to adjusted the 11 become 31 (alternative is to remove the filter, as the raw data is actually good enough).

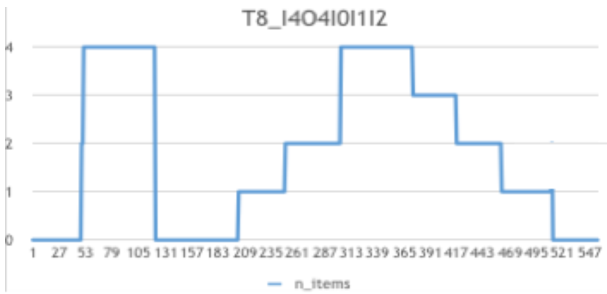


Figure 8

In Test Stage Final / 3, I tested the whole performance of this system for 20 times, the significant output (number of items, storage status, and put-in/out action) showed a 100% correction.

Critical reflection

This project maximised the ubicomp systems' value and highly responded to the real-world needs and scenarios in warehouse inventory management. Although only a single storage unit was implemented technologically, it has been designed as the core part of an overall integrated system and is likely to bring more benefits after further development.

When considering the real application: **a)** ultrasonic sensors might not be able to count items with less-then-2cm width (or heights) and the accuracy will be influenced by surface materials and other environmental factors (e.g. temperature). **b)** In order to reduce the error caused by abnormal situation (e.g. damaged single sensor), two or more Ultrasonic Sensors should be used in each unit (only 1 Ultrasonic in our prototype).

Outcomes of testing is obvious. It enabled me to consider the system error into codes and choose the most appropriate sensor (i.e.use motion sensor to judge the condition rather than button sensor) based on their real effect on the performance of the prototype in simulated environment. Programming has become a process of constantly iteration, in which each time's modification contributed to a better performance of this system.

Limitation of my tests cannot be ignored. I conducted test only with one type of items, paper material and fixed width in one single simulated environments, while the accuracy of ultrasonic sensor will be influenced by them to a large extent, in particular, the surface materials (e.g. metal, wood, plastic). Also the power effect of my tests is low, as the limitation of the number of tests, and the single-size items. Further testing should be conducted more sufficient times with various items in several different environments.

References

Chalmers, M., & MacColl, I. (2003, October). Seamful and seamless design in ubiquitous computing. In *Workshop at the crossroads: The interaction of HCI and systems issues in UbiComp* (Vol. 8).

Liu, J., Han, J., Lv, H., & Li, B. (2015). An ultrasonic sensor system based on a two-dimensional state method for highway vehicle violation detection applications. *Sensors*, *15*(4), 9000-9021.

Köhler, P., Connette, C., & Verl, A. (2013, May). Vehicle tracking using ultrasonic sensors & joined particle weighting. In *2013 IEEE International Conference on Robotics and Automation* (pp. 2900-2905). IEEE.

Appendix A – instructions

Instruction: It should be with the prototype that unit_size = 51cm; the used items should be 7cm width. Run the codes directly and the output number will represent the information that we need. For behaviour: 1 = Put-In; 2 = Put-Out; For Stock Status: 0=Well-Stocked; 1 = Shortage; 2 = Abnormal; For Items: n_items = current stock number of item.