TRACKING AND HANDS MOTION DETECTION APPROACH FOR MONITORING HAND-HYGIENE COMPLIANCE FOR FOOD HANDLING AND PROCESSING INDUSTRY

By

YONGBO WAN

Bachelor of Science in Electrical Engineering
Shaanxi University of Science and Technology
Xi'an, Shaanxi, China
2001

Master of Science in in Electrical Engineering
Shaanxi University of Science and Technology
Xi'an, Shaanxi, China
2006

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
DOCTOR OF PHILOSOPHY
July, 2013

TRACKING AND HANDS MOTION DETECTION APPROACH FOR MONITORING HAND-HYGIENE COMPLIANCE FOR FOOD HANDLING AND PROCESSING INDUSTRY

Dissertation Approved:

Dr. Ning Wang

Dissertation Adviser

Dr. Marvin Stone

Dr. Paul Weckler

Dr. Tim Bowser

Dr. Weihua Sheng

ACKNOWLEDGEMENTS

This dissertation is a milestone in my academic career. I would like to express my sincere gratitude to my advisor Dr. Ning Wang for her continuous support of my Ph.D. study, her immense knowledge, professional attitude and suggestions to my research, and her kindness.

Besides my advisor, I would like to thank the rest of my thesis committee: Prof. Marvin Stone, Dr. Paul Weckler, Dr. Tim Bowser and Dr. Weihua Sheng for their encouragement, insightful comments, and helpful challenging questions. And also Dr. Philip G. Crandall and his research team in University of Arkansas for brought out the research topic and their cooperation on this project.

My sincere thanks also go to my fellows in BAE Sensor Lab: Aaron Franzen, Yeyin Shi, Jinchang Zhang and Weiming Cai for the stimulating discussions, and being my loyalty volunteers for testing my research project.

Last but not the least; I would like to thank my family: my parents, my parent-in-law and especially my wife Yi Shao and my happy daughter Sophia Wan who was just one year old, for their supporting throughout my Ph.D. study.

Name: YONGBO WAN

Date of Degree: JULY, 2013

Title of Study: TRACKING AND HANDS MOTION DETECTION APPROACH FOR MONITORING HAND-HYGIENE COMPLIANCE FOR FOOD HANDLING AND PROCESSING INDUSTRY

Major Field: BIOSYSTEMS ENGINEERING

Abstract: Hand-hygiene is a very critical issue for both food handling and processing industry and health care service providers. Poor hand-hygiene practice can easily lead to foodborne illness or large scale decease transmission. In this research, an automatic tracking and monitoring system was developed that used a 3D camera for hand washing and hands motion detection and a sensor-based monitoring system for hand-hygiene activities evaluation. An active Wi-Fi portable Radio Frequency Identification (RFID) tag was used for personal ID tracking. The effective hand washing time, soaping time were measured based on the hands motion detection and hand movement tracking. Water temperature, water flow, paper towel, soap and hand sanitizer usage were also measured for each hand washing event. All the data were forwarded to a system server for data recording, storage and management. Preliminary test data were collected to evaluate the system performance. The results showed that the system could effectively collect most of the hand-hygiene related factors including hand-hygiene product usage, hand washing time and soap lathering time for hand-hygiene evaluation.

TABLE OF CONTENTS

Chapter Page

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

1. Statement of the Problems

The food processing industry is putting more and more ready-to-eat food into market providing customers more options for their daily meals. For example, the worldwide fresh-cut produce industry has grown rapidly in recent years to a multi-billion dollar sector. Fresh-cut products are very convenient for customers because they are ready to eat, easy to carry, fresh, nutritious, and time saving for cooking. However, from food safety perspective, studies have shown that improper food handler practices contributed to approximately 97% of foodborne illnesses in food-service establishments and homes in USA (Howes et al., 1996). Mishandling of food, especially fresh-cut food, is the leading reason of the occurrence of foodborne illness (Bean et al., 1996). More than half of the foodborne-disease outbreaks, 3,334 out of 5,429 outbreaks based on USDA statistics from 1988-2002, were sourced to at restaurants or delicatessens (Lynch et al., 2006). Hence, food safety remains a critical issue due to outbreaks of foodborne illness.

Food production, processing, and marketing are now globalized, which may broaden the nature of infectious disease spread. Pathogens can be disseminated from an original point of processing and packaging to locations at the other side of the world rapidly (Egan et al., 2007; Käferstein et al., 1997). Many outbreaks of foodborne illness resulted from faulty food handling practices and food workers' poor personal hygiene (Bryan, 1988; Evans et al., 1998; Guzewich and Ross, 1999). Hence, it is very important to gain an understanding of the current practices of food handlers and processors. This knowledge may then be used to improve the hygiene compliance and to reduce foodborne illness effectively (Clayton et al., 2002)Personal hygiene plays a very important role in reducing foodborne illnesses. Effective personal hygiene, especially hand-hygiene, is critical at every stage of food production, as most work is operated by hands. Most food production workers receive little or no training concerning hand and fingertip washing. Regulatory authorities currently only check if there is a hand wash sink in the food/production/service area, if this hand washing area is supplied with soap, and if the sink functions properly. No devices or procedures are used to monitor actual implementation of the hand-hygiene compliance. Food handling or processing companies are not able to track and record the hand washing behaviors of employees for long term operations just because of high cost and lack of demand. In addition, these companies cannot justify whether employees wash their hands at all or the hand-washing is sufficient to reduce pathogens residue on their hands and fingertips before working on food products.

With the new development of electronic and computer technologies, battery powered microcontrollers with wireless communication modules are now readily available technology. These devices are mainly working under a sleeping mode to conserve energy and awakened to process event handling only when a trigging event happens. The battery used to power these devices can last up to several years. Due to their power-saving working style, they can be used very conveniently to monitor some specific events, e.g. the usage of paper towels or soap usage. Sensors including phototransistor optical interrupter switches, limit switches, Hall Effect devices, etc. can be used to detect physical movements and generate a trigging signal. A real-time event monitoring system can then be developed.

2. Research Objectives

The primary goal of this research was to develop a system to automatically track, monitor, and analyze hand-hygiene compliance. The focus of the development effort was to enhance food safety through the improvement of hand hygiene compliance, which is a highly priority research area of the U.S. Department of Agriculture (USDA) and other related agencies such as Food and Drug Administration (FDA) and the Centers for Disease Control and Prevention (CDC). The designed system was intended to analyze the complex interactions of water usage, water temperature, hand soap/sanitizer usage, paper towel usage, and hand/arm movements during hand washing and relate these factors to hand-hygiene compliance evaluation.

The specific tasks to reach the objective included:

1) to evaluate available sensing technologies which could be used for monitoring hand-hygiene compliance
2) to develop a Radio Frequency Identification (RFID)-based sensing system to gather metrics which could be used to describe hand washing behaviors during a hand-washing activity
3) To build a prototype system for the automatic hand wash monitoring
4) To develop algorithms for location tracking and gesture and behavior monitoring; and
5) To conduct a system test in the pilot plant of Department of Food Science at University of Arkansas to evaluate system performance

The designed system should have the features listed as followings:

- Easy to be deployed in food industry

    - With wireless, light weight, wearable RFID tags
    - Automatic location tracking
    - Easily expandable network structure to cover the desired areas
- Automatic tracking and monitoring of operators' hands motion
    - Provide real-time hand-hygiene (HH) monitoring
    - Provide recorded data on hand-hygiene behaviors of each operators

3. Qualifier to Evaluate the Performance of the Developed System

As discussed above, no qualitative and quantitative criteria are available to evaluate the practice of hand-hygiene compliance. The targeted performance qualifiers used to evaluate the developed HH monitoring system include:

- Every data record needs to be RFID-tagged;
- Every data record should be time-stamped;
- A wearable tag needs to be able to provide RFID tag location information to differentiate two people near hand-washing sink with a distance bigger than 1.5m "social distances" (Changingminds.org, 2013; Cherry and About.com, 2013);
- The image processing algorithm can track hands location and motion with a refreshing speed greater than or equal to 15 frames per second; and
- The sensor measurement errors for hygiene product usages, water temperature, and water flow rate should be less than 5%.

CHAPTER II

LITERATURE REVIEW

1. Impact of Hand-Hygiene in Food Industry

Improper food handling is one of the main reasons for the outbreaks of foodborne diseases. Between January 1st 1995 to December 31st 1996 in England and Wales, 1919 general outbreaks of infectious intestinal disease were reported. Over half of the outbreaks were transmitted from person to person, while twenty-two percent were mainly foodborne (Evans et al., 1998). From 1975 to 1998, seventy-two articles described a total of 81 outbreaks in the U.S. (Guzewich and Ross, 1999). Ninety-three percent of the outbreaks were caused by the pathogens carried by food workers before or during the food processing. The causes of eight-nine percent of the outbreaks were sourced to food processing at process facilities.

According to the journal of Morbidity and Mortality Weekly Report, hands might be the major media that transmitted viruses and pathogens (LeBaron et al., 1990). Foods requiring intensive hand contact like sandwiches and salads had higher chances of carrying foodborne disease compared to foods not requiring direct hand contact (Guzewich and Ross, 1999). The annual cost for the United State to deal with the food outbreaks was estimated in between 6.5 to 35 billion(Buzby and Roberts, 1997).

2. Hand-Hygiene in Hospitals

Similar to food industry, hands play an important role in decease transmission in healthcare institutions. About two to three million deaths occur each year from diarrheal disease worldwide. Many of these outbreaks can be avoided if proper hand-hygiene procedures were used. However, hand-hygiene compliance is still poor worldwide. Jumaa, et al. (2005) summarized that hand-hygiene was the most effective way of interrupting the transmission of microorganism caused infection in both community and the healthcare settings. Though the methods involved in hand-hygiene are simple, implementing them becomes complicated when adding factors such as work load, preparations for hand-hygiene, cultural influences, and different human behavioral reactions etc.

Hand-hygiene compliance remains low in most hospitals (Allegranzi and Pittet, 2009; Boyce and Pittet, 2002; Challenge, 2009). The World Health Organization (Allegranzi and Pittet, 2009) issued Guidelines on Hand-hygiene in Health Care on May 6, 2009. The main objective of this document was to improve hygiene practices in healthcare providers to reduce transmission of pathogenic between patients and healthcare workers. In the guidelines, it said that "health care-associated infection (HCAI) is a major problem for patient safety and health care providers should always put the patients' safety as their first priority. And hand-hygiene is the most important and effective measure to prevent HCAI". But very low compliance rates and low adherence to good hand-hygiene procedures have been reported in health providers in both developed and developing countries. In developed countries, HCAI happened to 5-15% general hospitalized patients, within which up to 37% are the patients in intensive care units (2007). In developing countries, it is several times higher than those in developed countries because of the paucity and unreliability of laboratory data, limited data access, and poor medical record keeping cannot be used to estimate the actual compliance rate accurately.

3. Factors Influencing Hand-Hygiene Compliance

Green et al. (2002) claimed that the factors related to food worker hand-hygiene practices included their activities, characteristics of their working environment, food safety training and physical and social environment. They found that the usage of gloves was another factor that affected the hand-hygiene compliance. Hand-hygiene could be improved with appropriate regulations. For example, the effective glove changing habit could enhance the hand-hygiene. The food workers needed a glove change if he/she touched anything that had been or had potentials of being contaminated. However, wearing gloves could not guarantee hand-hygiene. They suggested that multiple hand-sinks on sight, convenient access to gloves and other supplies required for hand-hygiene improvement be provided to food workers. Clayton et al. (2009) stated that safety training was not enough to improve food safety. Adequate hand-hygiene product resources and appropriate management culture were required for food safety control. Generally, food workers were aware of the importance of the food safety and food hygiene practice. The reasons for low hygiene compliance that prevented them implementing proper hygiene practices included lack of time and shortage of resources.

Challenge (2009) mentioned that a health care worker needed to wash their hands as many as 42 times per shift and up to 15.2 times per hour. The hand washing duration also varied in the range from 6.6 seconds to 30 seconds.

4. Hand-Hygiene Procedures

On May 5th 2009, the World Health Organization issued Hand-hygiene in Health Care to help promote the hand-hygiene practices in health care as a priority measure for infection control (Challenge, 2009). The guideline included the hand-hygiene requirements as follow:

- Apply a palm full of alcohol-based hand rub and cover all surfaces of the hands. Rub hands until dry.
- When washing hands with soap and water, wet hands with water and apply the amount of product necessary to cover all surfaces. Rinse hands with water and dry thoroughly with a single-use towel. Use clean, running water whenever possible. Avoid using hot water, as repeated exposure to hot water may increase the risk of dermatitis. Use towel to turn off tap/faucet Dry hands thoroughly using a method that does not re-contaminate hands. Make sure towels are not used multiple times or by multiple people.
- Liquid, bar or powdered forms of soap are all acceptable. When bar soap is used, small bars of soap in racks that facilitate drainage should be used to allow the bars to dry.

Two charts were provided to explain the right procedure for hand washing procedure by WHO and the Lung association L'assocation Pulmonaire (www.lung.ca) in Figure 1 and Figure 2.

5. Hand-Hygiene Compliance Monitoring

Boscart, et al. (2008) claimed that in healthcare institutions, infection transmission was a significant threat to patients and healthcare workers. Based on the published statistics, the rate for hand-hygiene compliance of healthcare staff was about 40% and various education and training on hand-hygiene only increased the rate to about 50%. To improve effectiveness of hand washing, monitoring systems were needed to provide the first-hand proof of good hygiene practices.

Figure 1 Hand-hygiene technique with soap and water (Challenge, 2009)

Figure 2 Recommended Hand-hygiene practice (Lung association, 2013)

6. Hand-Hygiene Compliance Monitoring

Boscart, et al. (2008) claimed that in healthcare institutions, infection transmission was a significant threat to patients and healthcare workers. Based on the published statistics, the rate for hand-hygiene compliance of healthcare staff was about 40% and various education and training on hand-hygiene only increased the rate to about 50%. To improve effectiveness of hand washing, monitoring systems were needed to provide the first-hand proof of good hygiene practices.

Boscart et al. (2008) reported a hand-hygiene monitoring system developed based on the concept of a wearable gel dispenser and a wearable monitor device. A monitoring device was worn by a person who could activate an alcohol gel dispenser wirelessly. The device also recorded the number of times of the activation of a gel dispenser during a preset time period. Then the usage of the gel dispenser by a specific person could be estimated and used to achieve continuous increases of hand-hygiene compliance. The locations of the healthcare worker were also tracked as they entered or left the specific working area. A prompting signal was provided when a hand cleansing was required. The hygiene monitoring device received good responses from the test group of healthcare workers. More frequent hand wash activities were observed, though there were some concerns about the privacy issues on being watched.

Boyce (2010) stated that hand-hygiene monitoring could be considered as a component of a successful hand-hygiene promotion program. The author compared the advantages and drawbacks of different compliance rating methods, such as direct observation, observation survey, and self-reporting. Direct observation was considered to be the most accurate ways to monitor hand-hygiene compliance, but it was time consuming and required intensive repetitive work. Subjective factors might also affect the results. The observation survey was not very trustworthy because of the lack of standard observation techniques. Self-reporting was simply not a sufficiently reliable method. Hand-hygiene products usage monitoring was easily applicable and required less time for data collection. But the developed components did not provide the quality of information needed.

Edmond et al. (2010) reported an experiment conducted in wards with 35 beds in a hospital to monitor hand-hygiene compliance. Alcohol foam dispensers were installed inside and outside of each patient room. Each healthcare worker with a badge was required to perform an alcohol hand scrub within eight seconds of the entering or leaving the patient room. The badge could enable color changes of an onboard light and beeping signal when a hand cleaning was required. The testing results showed a rapid and significant improvement of hand-hygiene compliance with the use of the monitoring system. On the other hand, a LED light was used to indicate the hand washing status if a hand wash were required. A patient could observe the light changes on the badge to remind the healthcare workers washing their hands.

Sahud et al.(2010) conducted a pilot study in a 700-bed hospital to evaluate the feasibility and effectiveness of an automatic surveillance and monitoring device for hand-hygiene compliance. The device included a room trigger, soap dispenser trigger, and a reader. The reader was worn by a health care worker and recorded the room entry and exit and the changes of soap dispensers. Weekly data for each individual participant was feedback to each participant for improvement. They concluded that the electronic hand-hygiene surveillance device had potential to be an accurate tool for hand-hygiene routine monitoring.

7. Parameters Used to Monitor Hand Washing

Critical hygiene parameters need to be measured and monitored closely. For example, washing vigorously for 20 vs. 40 seconds may produce different hygiene results. The following parameters can be monitored and recorded based on the WHO hand-hygiene guideline:

- Duration and completeness of wetting hands prior to soap usage
- Temperature of water used
- Soap usage
- Duration of washing time
- Aggressiveness of washing with soap
- Rinsing time
- Paper towel usage
- Sanitizer usage

8. Radio-frequency Identification Technology

Radio-frequency identification (RFID) techniques use wireless, non-contact, radio-frequency electromagnetic fields for data transmission. RFID frequency bands cover from 12kHz up to 10 GHz (Malik, 2009). For different frequency bands, an RFID has different coverage ranges and data transmission speeds. Generally, the lower the frequency, the lower the data transmission speed. The tags used in a RFID system are either passive or active tag. The cost of the active tags is much higher than that of passive ones. Passive tags usually have very small size and need no built-in batteries. The cost for each passive tag can be as low as a few cents. Passive tag technology can be easily embedded in solid nonmetallic items such as labels, pallets and cards. But the disadvantages of passive tags include low performance around liquids and metals, crowded frequency band, no global standardization for localization applications, high network traffic, and large number readers required. The active RFID tags can employ many RF-based technologies including Ultra-Wide Band (UWB), ZigBee, Bluetooth, Wi-Fi, and even cellular networks, etc. They can continuously send out signals, scan or listen for the signals from other devices a predefined frequency. The active tags have their own power supply that can support a longer coverage range of up to 200 meters and flexible network topology and structures. As mentioned, active tags have different communication standards available for different wavebands or technology (Rahimi et al., 2005).

9. Human Behavior Recognition

To monitor human behavior and to reliably detect and track a specific person is very challenging. Human behavior is complex and highly dependent on individual habits and social nature. Technology that is capable of sensing human behavior can be categorized into two types: pervasive systems and action and activity recognition systems (Rahimi et al., 2005). A pervasive system requires many sensors to collect human-behavior-related data. Salah et al (Hu and Wo, 2010) mentioned five basic elements to understanding an environment for human behavior recognition, including detection, localization, tracking, recognition and understanding. Various methodologies are needed to increase the recognition rate, for example, a vision based computer recognition classifier method (Mizuno et al., 2007; Sugimoto et al., 2006), and wearable sensor systems with sound motion and pressure sensors to recognize common human behavior such as walking, eating, et al. (Kim et al., 2009; Shim et al., 2011).

All the above methods used "state of the art" technologies or computer vision algorithms to detect or predict possible parameter changes caused by human behavior. Because of the unpredictability of human behavior and environment changes, different techniques should be used for different applications. To understand and represent human behavior, the sensed primitives should be enough to represent and differentiate actions. The more measured parameters, the more accurate the monitoring system can be. But these more complex systems need more training samples for learning, longer time for computation during operation, and large computer resources to store and manipulate the collected data.

10. 3D Image Sensing Techniques

There are several methods for depth measurements with light waves. They include multiple-camera view triangulation, single time-of-flight camera (ToF camera), structured light imaging, and laser scanning. Multiple-camera view based systems need at least two calibrated cameras and multiple computational steps. The qualities of the images are highly dependent on the surface texture. The cameras capture the object from different view angles, all the images processed with radial undistortion, rectification, stereo matching and triangulation to obtain the depth information. The ToF camera only needs one camera with less computation requirements and is independent of surface texture to be measured. But it requires high-accuracy time measurement and usually has limited resolution and accuracy(Kolb et al., 2009). The ToF camera measures the depth by measuring the absolute time interval during which a light pulse travels from a source to a target object and reflects back to measure distance. Laser range sensors also use the time of flight principle to measure distance to a target. The Microsoft Kinect sensor projects an infrared light pattern into a 3D scene and measures a reflected light pattern with its IR camera. The light pattern distortion of the reflected structured light is used to compute the 3D structure(Clayman, 2010).

On June 1, 2009, Microsoft (Microsoft Corporation, Redmond, Washington) first announced its Kinect sensor - a motion sensing input device for the Xbox 360 video game (Microsoft, 2012). The sensing technology behind this sensor was invented in 2005, which was a spatial neighborhood structured light imaging system. The Kinect sensor has three major components for 3D imaging: a laser based Near Infrared (NIR) projector, an IR camera, and a color camera. The laser projector projects a known speckle NIR light pattern. The IR camera collects the reflected speckle light from the targets. The depth data is calculated by triangulation of each speckle between a speckle pattern and the observed pattern. The observed speckle size and shape depends on the distance and orientation with respect to the sensor.

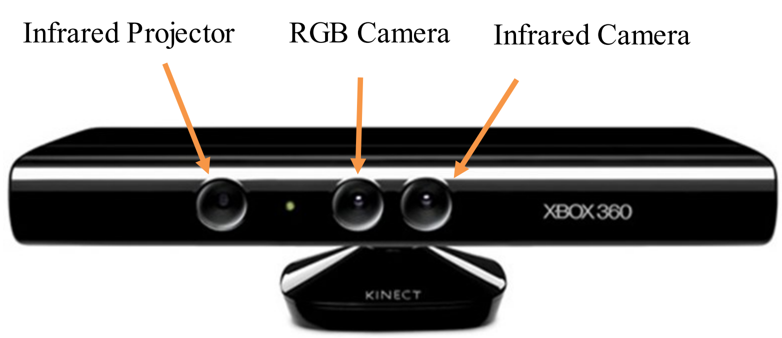


Figure3 Kinect Sensor (Takahashi, 2012)

The feature specifications of the Kinect sensor are listed below:

- Depth resolution: 640x480 pixel @ 30FPS
- RGB resolution: 1600x1200 pixel @ 12FPS; 640x480 pixel @ 30FPS
- Operation range: 0.8m~3.5m
- Spatial X/Y resolution: 3mm @2m distance
- Depth Z resolution: 1cm @2m distance

11. Hand Motion Tracking Using the Kinect Sensor

Beside its major applications on video games, the Kinect sensor has been used to develop various applications. As of January 9, 2012, Microsoft has sold 18 million Kinect sensors since it first launched (Takahashi, 2012).

Gallo et al. (2011) introduced a controller free, highly interactive medical image exploration system . A Kinect sensor was used to track the user's hand and arm gesture. A gesture control interface was designed that relied on the concept of an "activation area" for hand appearance detection in the activation area. By using the activation detection, nine different image operations were performed with the system, including pointing, click, erasing, animating, region of interest extraction, zoom, translation, and rotation etc.

Oikonomidis et al.(2011) developed a marker-less hand articulation 3D tracking system with a Kinect sensor. The system was able to provide a 15Hz continuous solution for hand articulation 3D tracking. The tracking algorithm involved intensive computation of rendering, pixel-wise operation between observation and hypothesis map and overall summation. The algorithm was part of a Particle Swarm Optimization of a stochastic evolutionary algorithm introduced by Kennedy and Eberhart (2011). Beside a high speed quad-core Intel i7 950 CPU, an NVidia GTX 580 GPU with 1581 GFlops processing power video card was used for the computation.

Ren et al.(2011) introduced a hand motion detection method, Finger-Earth Mover's Distance (FEMD), based on a Kinect sensor. They mentioned that the noisy images of Kinect sensor made it very challenging for hand shape recognizing, especially with severe distortions due to movements. In order to accurately detect the fingers, the FEMD method considered each detected finger as a cluster and penalized unmatched fingers as empty finger-holes. In FEMD, a hand and its contour were detected and segmented at first. Then the normalized hand contour was used to measure the distance between predefined signatures or histograms for hands motion detection.

Ten hand gestures were tested with local distortions and changes of orientation and scale. They concluded that their finger decomposition methods were able to accurately and efficiently recognize over 90% of hand gestures with articulations, distortions and orientation change.

Jesse (2012) presented a hand washing assistance system with an image processing method by using flocks of features from a video camera. He claimed that the flock features could maintain a consistent motion when the objects were independently moving. A sequential Markovian process method was used to estimate the flock density over time.

CHAPTER III

SYSTEM HARDWARE DESIGN

1. Overall System Design Concept

The overall design of the hand-hygiene monitoring system was divided into four parts: a sensing unit, a RFID unit, a hands-motion detection unit, and a sensor fusion and data management unit. The sensing unit collected data indicating the usages of water, soap/sanitizer, and paper towel and water temperature. The RFID unit provided identification information of a person and location information in a dedicated hand washing area. The hands motion detection unit monitored the movements of hands and checked if the movements followed the hand-hygiene compliance rules. The sensor data collection and data management unit was used for data integration from different sensing sources, data storage in a data log file, and data interpretation and analysis. Figure 4 provides an overall block diagram of the system design.

2. Major System Components

System Server

Since the system sever needs to process the video stream from the 3D camera sensor, a PC with a relative high computation power was required. The minimum specifications of the PC should be:

- Computer Operating System: Windows 7 or higher

- Desktop: Core 2 Duo or higher, 4G RAM, discrete video card
- Laptop: Intel i5 CPU, 4G RAM

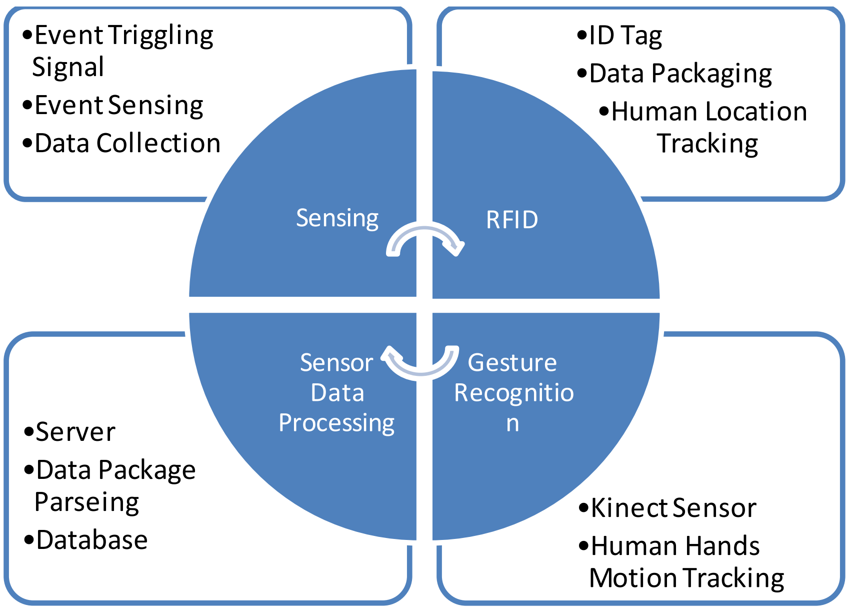


Figure 4 Block Diagram of Overall System Design

Microcontroller

Two types of microcontrollers were used in the prototype system: Seeeduino Mega (www.seeedstudio.com , Seeed Studio Inc, Shenzhen, China) and Aruduino Pro Mini (www.arduino.cc). The Seeeduino Mega was used in a sink node to collect data from wired sensors and Wi-Fi access point module. The Arduino Pro Mini was used for the wearable mobile tag because of its small size and onboard regulator for multiple power options. The Seeedunio Mega had four UART ports for serial data communications, two of which were used for server and Wi-Fi module data communication. One analog input channel and three GPIO ports of the Seeeduino microcontroller board were used to monitor water temperature and hot cold water flow. The Seeeduino controller continuously monitored and collected data from these ports, formed messages with a specific format, and sent the messages to the PC/Server through a UART port.

Table 1 showed the basic features of the two selected microcontrollers.

Table 1 Specifications of Selected Microcontroller

	Seeeduino Mega	Aruduino Pro Mini
Microprocessor	ATmega 1280	ATmega168
Operating Voltage	Selectable 5V/3.3V	3.3V
Input Voltage	7 - 12V	3.35 -12 V
Digital I/O Pins	70(14 PWM)	14(6 PWM)
Analog Input Pins	16	6
UART	4	1
DC Current per I/O Pin	40mA	40 mA
Flash Memory	128KB	16 KB
SRAM	8KB	1 KB
EEPROM	4KB	512 bytes
Clock Speed	16MHz	8 MHz

RFID Module

An ultra-low power Wi-Fi tag was developed to be worn by each operator and used as identification for the operator. This tag was also capable of Wi-Fi based infrastructure network localization service that continuously provided the specific subject's location information. The tag communicated wirelessly to the Wi-Fi access point. There were several different types of Wi-Fi module available for selection. For example, WizFi210 (WIZnet Co., Ltd. Jin Buhm Kim, Korea), Carambola 2 (8devices, Kaunas, Lithuania), WL11-IP (North Pole Engineering, Inc.

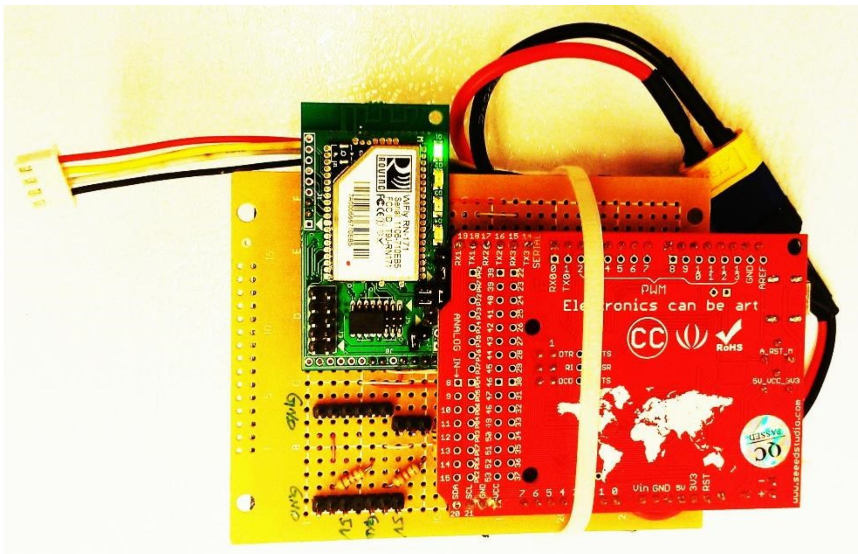
Minneapolis, MN, USA), CC3000 (Texas Instruments, Dallas, TX, USA) and RN174 WiFly (Roving Networks Inc. Los Gatos, CA USA). All of these tags were IEEE 802.11 b/g/n compatible. The detailed comparison among these modules is given in Appendix A.

The RN174 WiFly radio module was chosen for this project because of its flexible configurations and very helpful online design and application documents for system prototype and actual implementation. It was a complete, fully qualified and Wi-Fi certified 2.4-GHz IEEE 802.11 b/g wireless LAN access embedded transceiver. It had an ultra-low power intelligent, built-in power management device that supports a programmable wakeup soft Access Point (AP) with an on-board ceramic chip antenna and/or U.FL connector for external antenna. In this research, seven Wi-Fi modules were used for different purposes including three location sensor (one sink node and two anchor nodes) to provide location reference information, three sensing node sensor for paper towel, hand soap and sanitizer usage monitoring, and one wearable mobile tag sensor for identification and location estimation.

Location Sensor: Sink node

A Wi-Fi module was connected to the Seeeduino microcontroller board(Seeeduino) to form a "sink node" or Central Control Unit (CCU). All the other Wi-Fi modules sank their data to this sink node to be transmitted to the system server (Figure 5). Two UART ports of the microcontroller were used for data communications. One was used between the server and the Seeeduino. The other was between the Seeeduino and the Wi-Fi module on the sink node. This Wi-Fi module served as an access point was configured to listen to a specific TCP/IP port for the data transmissions from other nodes, and was powered on all the time through a USB ports and waiting for the incoming data communication request from the other nodes. The Wi-Fi module was given a Service Set Identifier (SSID) name with a format as "gateway-XXXX", where

XXXX was the last two/four character of the Wi-Fi module's Media Access Control (MAC) address.



a. Sink Node with a wireless AP Module

b. Sink Node (CCU) Block Diagram

Figure 5 Prototype of a Sink Node with Wi-Fi Module: a. Sink node with a wireless AP module; b. Sink node (CCU) block diagram

Location Sensor: Anchor node

An anchor node was a standalone Wi-Fi module powered by two AA batteries for easy sensor deployment. It could also be powered by a DC source from an adaptor. Two anchor nodes were used in the developed system. Since each anchor node was required to be powered on all the time to provide localization reference information for the mobile nodes, an AC-DC adaptor might be a better option. The Wi-Fi module on the anchor node was configured to be booted up in the AP mode and named using a SSID of “gateway-XXXX”, where XXXX was the last two/four character of the Wi-Fi module’s Media Access Control (MAC) address. Figure 6 and figure 7 show a picture of three anchor nodes with two-AA battery packs and the schematic diagram of the anchor node, respectively.

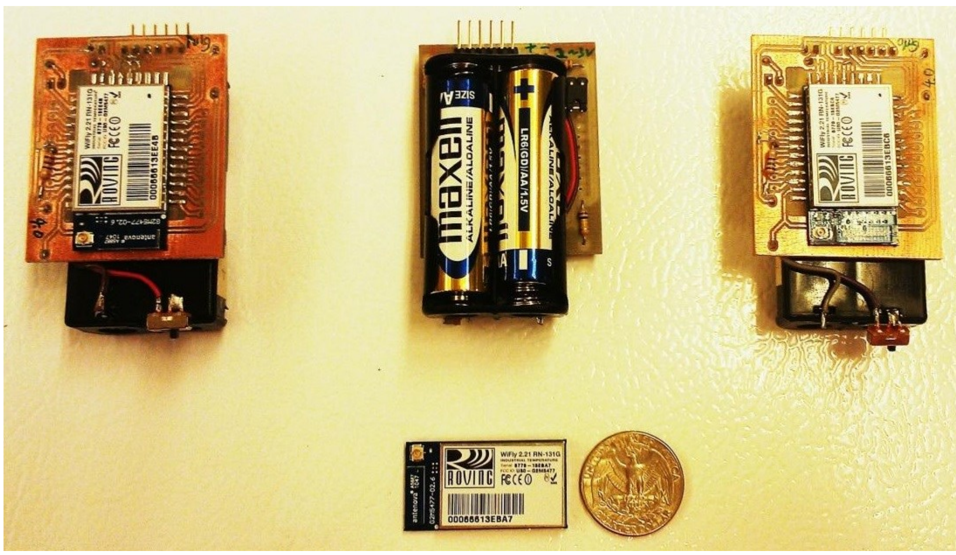


Figure 6 Three Anchor Nodes and its onboard Wi-Fi Chip

Wearable Mobile Sensor Node: Mobile node

The wearable mobile sensor node (mobile node) consisted of a microcontroller, a Wi-Fi module, and an AAA battery pack. The Wi-Fi module was configured to work at client mode rather than AP mode. An Ardunio Pro Mini microcontroller was used. The mobile node sent out a scan package to the sink node and the anchor nodes for localization every 10 seconds. After receiving a successful acknowledgement from the sink node, the Wi-Fi module of the mobile node entered into a deep sleep mode to reduce power consumption. Figure 8 shows two mobile nodes.

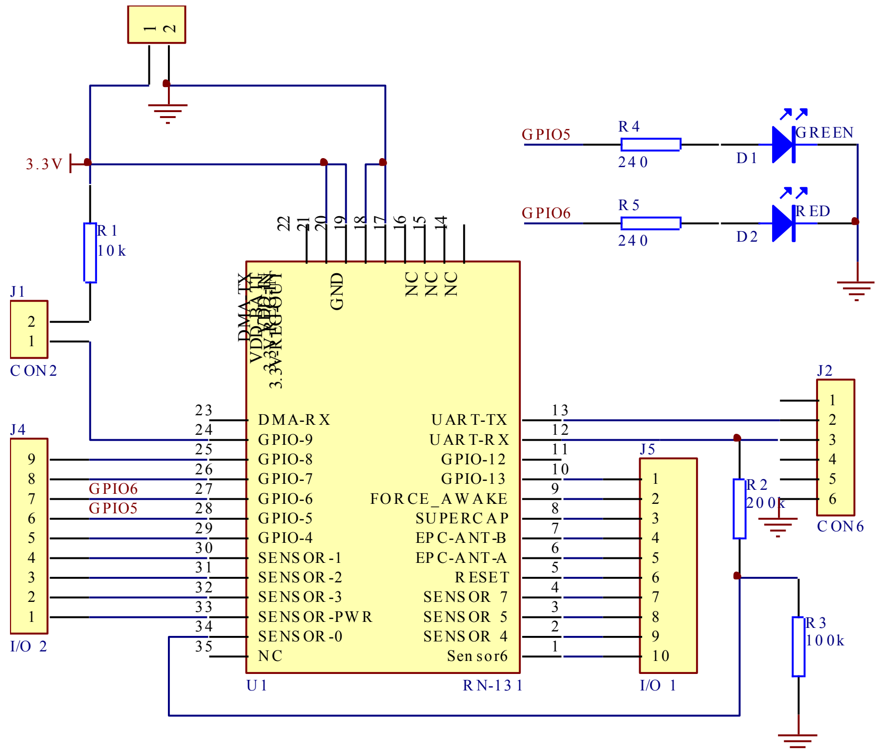


Figure 7 Schematic Diagram of the Anchor Node

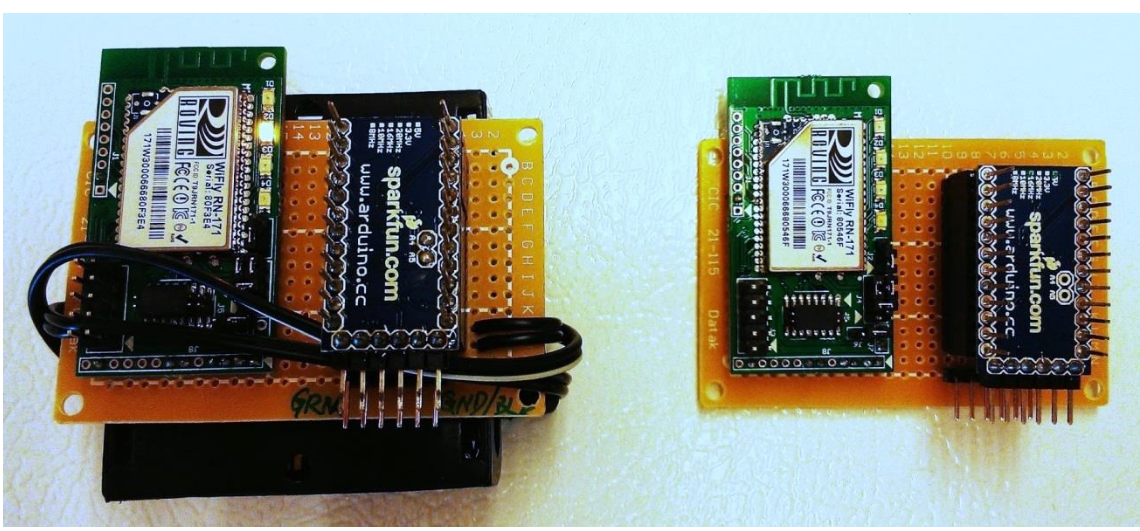


Figure 8 Wearable Mobile Sensor Node

Gesture Measurement Device: Kinect Sensor

The Microsoft Kinect sensor provides a well-developed and cost effective solution for high quality video and depth measurements. It provides a RGB video stream, a depth stream and a skeleton tracking stream with joint locations. With full-body joint-location tracking provided by Microsoft, simple hand-motion detection can be easily realized. The Kinect sensor has an infrared light source that projects light patterns onto the scene. The patterns are randomly distributed with a fixed pattern of spots. The IR camera captures the pattern that is deformed by the geometry of the projected area and compared with a reference image at known depth. The shift of the pattern between the reference image and camera captured pattern can be computed by the Kinect sensor internal processor for the depth frame data. However, the Kinect sensor Windows Software development kit (SDK) package provided by Microsoft has only a full-body tracking algorithm built into the device. It does not provide partial-human-body joint-tracking methods. In order to use the Kinect sensor for hands motion detection beside a hand-washing sink, it is very difficult to mount the sensor to meet the distance and field-of-view requirements for whole body tracking. Hence, a new hand-tracking algorithm was developed to extract only hand locations from the Kinect sensor's raw depth data stream. Because the camera generated a large amount of data, a high computation power computer was required for data processing. The Kinect sensor was connected to the PC/Server through USB port.

The Kinect sensor had two cameras: a normal video camera and an IR camera. The video camera was used to capture a regular color video stream as any general off-the-shelf USB camera or webcam. The color image sensor generated frame-ready events continuously to indicate a new frame of data is available. The data frame of 640*480 at 30 frames per second was used for color image video capture in the hand-hygiene monitoring system. However, the image data from the Kinect video camera was not used for any image processing. They were only used to feed the user interface on the server for displaying and comparison to the depth binary image.

The IR camera provided a third dimension 640*480 resolution depth map output with 30 frames per second, which contained the distance measurements of the scene within a 1cm measurement accuracy at a range of two meters. The camera used an infrared emitter and an IR camera for distance measurement. The Kinect SDK processed the IR data from the IR camera to produce a depth image stream and generate a depth image frame ready event to indicate the depth data was ready for reading. Figure 9 shows the field of view of a Kinect sensor. The normal Kinect depth vision range was from 0.8m to 4m with a pyramid shaped envelope. However, objects farther away from the Kinect sensor had greater depth values and the accuracy degrades as the distance increased. Hence, the recommended range was set from 0.9m to 3.6 m. The pixel data in the depth frame was a 16-bit number with 13 bits of depth data and 3 bits indicating the player index. The player index was a random number from 1 to 6 assigned by the sensor when a person was tracked by the sensor. The player index was at the least significant bits from bit 0 to bit 2 and the depth value was stored in bits 3 to 15. A bit shift operation was required to get the distance value for each pixel. This index stayed the same if the person was continuously tracked by the sensor, but it could change if the tracked person left and reentered the sensor field-of-view. Each pixel depth value was matched to a physical point of the scene in the field-of-view of a depth frame and was stored in a 1-D array. The index of a specific pixel of a frame was calculated using Eq.: 1

$$\text{PixelIndex} = \text{pixelX} + \text{pixelY} * \text{frame.width} \qquad (1)$$

For example, the pixel (20, 36) in a depth frame image's depth data index was stored in a 1D array with the index of 2360 if the depth frame had a resolution of 640*480. The unit of the depth value is in millimeters.

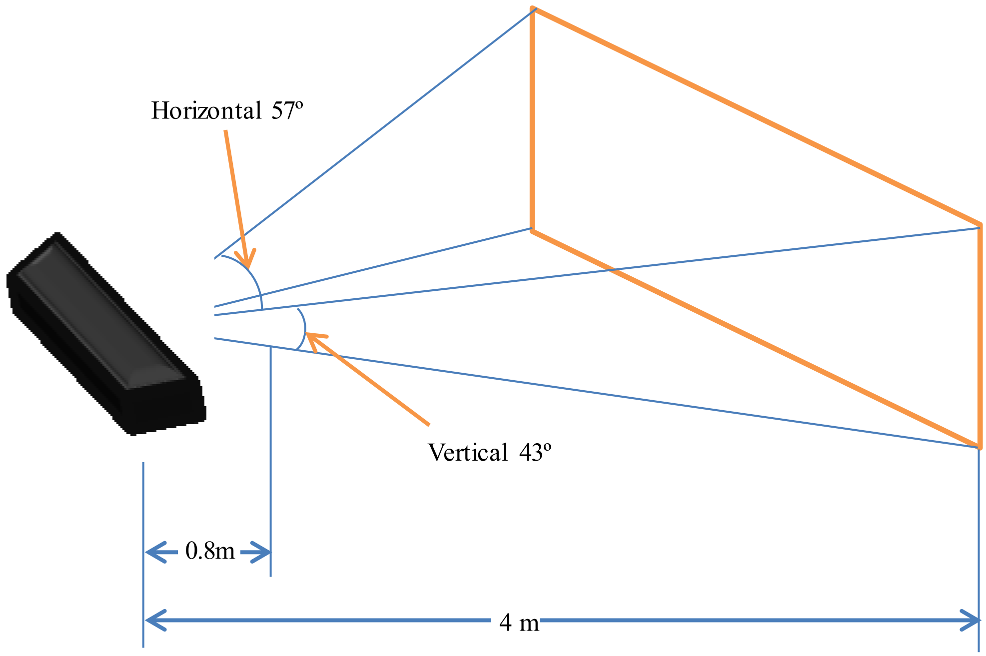


Figure 9 Field of View of the Kinect Sensor

Hands motion detection and tracking with the Kinect Sensor

The Kinect Sensor was used to track and measure the real-time hands location of a human subject that appeared within the hand-washing area. The Kinect sensor was connected to a computer directly via a USB port. The sensor was mounted 1.2 meters above the sink facing the sink counter. Figure 10 shows the Kinect sensor mounting diagram viewed from the side of the hand washing sink.

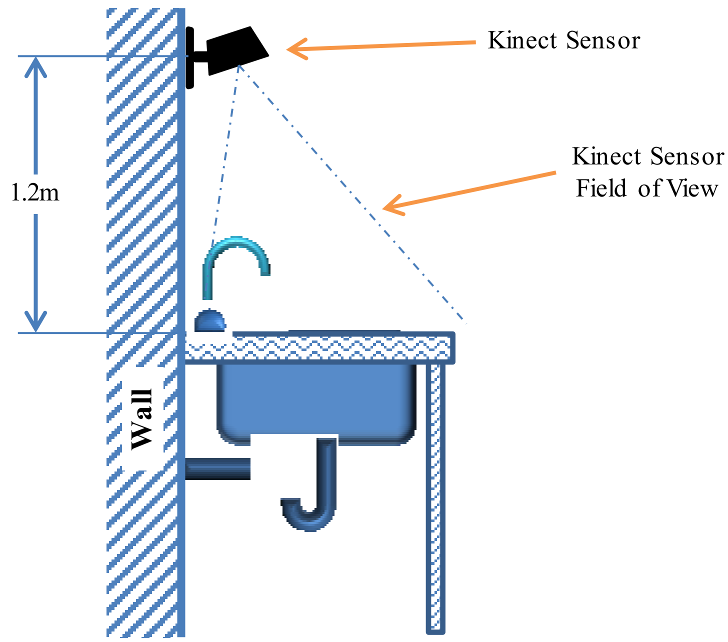


Figure 10 Kinect Sensor Installation Diagram

3. Design of the Sensing Unit

The major task of the hand wash compliance monitoring system was to initiate a data logging process whenever a hand washing activity was about to happen. Hence, it required human location tracking, identification information collecting, hands motion detecting, hand washing environment monitoring including the usage of water, soap, paper towel etc. The system then was designed to record all the related data and store them into a log file.

The sensing system for data collection had two sections, a data collection section with a microcontroller unit (Micro-programmed Control Unit, MCU) and an image and data processing section with a PC/Server. The Wi-Fi modules were put into wearable ID tags (mobile nodes), an automated paper dispenser, and an automated soap dispenser, respectively. Each operator wore a mobile node which periodically scanned surrounding area for the location sensors and forwarded the scanned information to the PC/server.

Figure 11 shows a detailed system block diagram, the data flow and the interconnected data communication methods of the developed hand-hygiene compliance monitoring system.

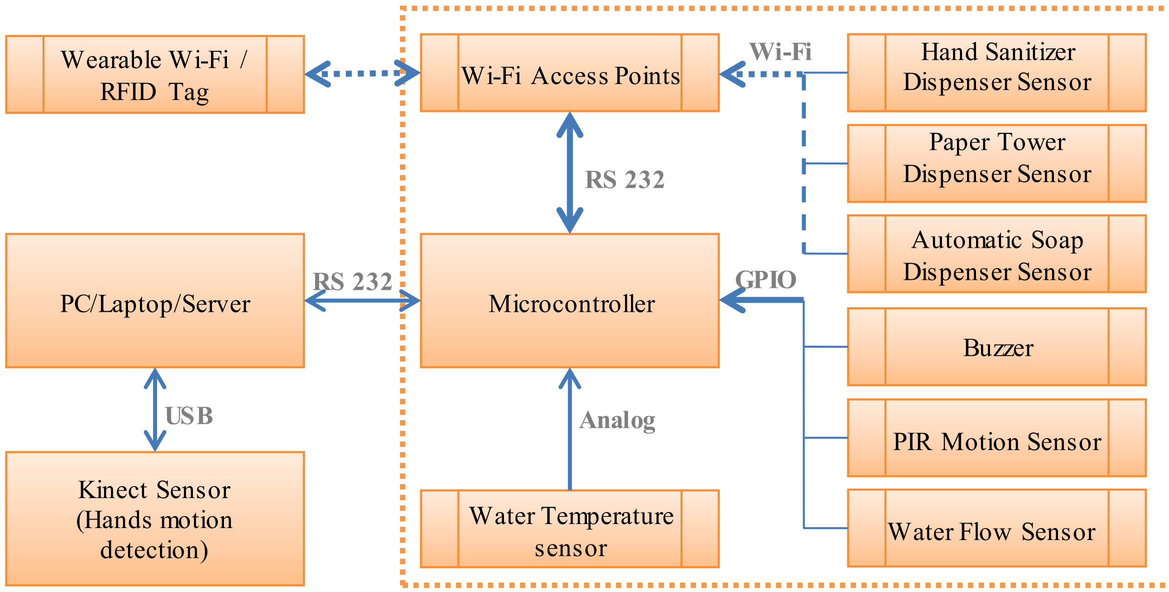


Figure 11 Block Diagram of the Developed Hand Hygiene Compliance Monitoring System

The sensing unit indicated in the dot line box in Figure 11 was used to collect data related to the usage of water, paper towel, hand soap/sanitizer, etc. during a hand washing activity by the Seeeduino microcontroller. Seeeduino then packed the data in a predefined format and forwarded the data packets to the PC/Server.

Figure 12 shows the sensing unit with the microcontroller, two mobile nodes, an anchor node, a modified soap dispenser, and two water flow sensors.

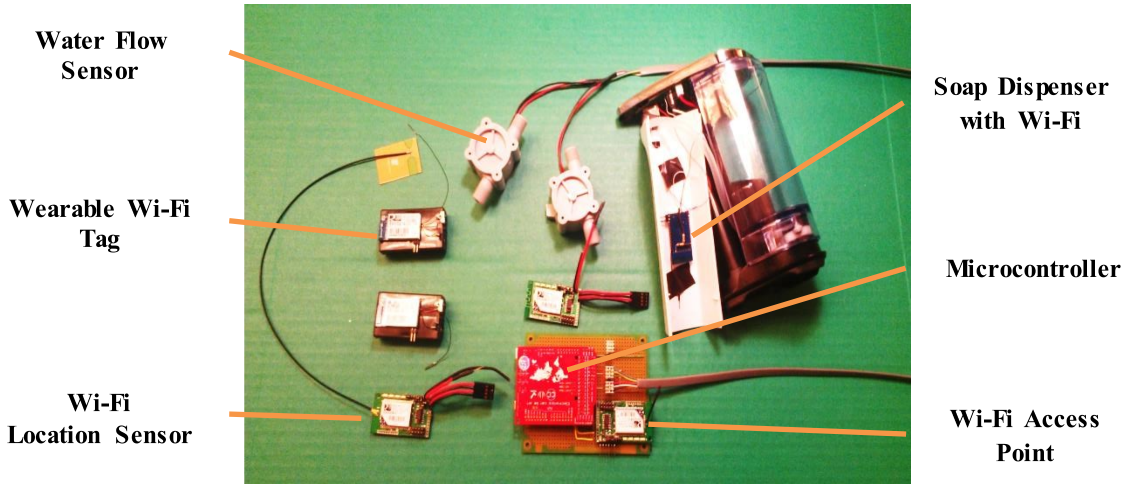


Figure 12 Major components of the developed Hand-hygiene Compliance Monitoring System

Water temperature sensor: Thermistors were low cost and widely used temperature sensors with adequate response time. The thermistor's resistance changed significantly with temperature. Within a known water temperature range, thermistor's nonlinearity could be compensated or corrected by a software curve fitting and calibration. In this design, the Seeedunio microcontroller used an analog input to read the water temperature from a thermistor which was installed in a drain pipe underneath the sink. Water temperature was collected every two seconds if the microcontroller detected a water flow. Figure 13 showed the thermistor's calibration curve in the temperate range of 20 to 60℃ which was a normal water temperature range for hand washing. A voltage divider circuit output for the thermistor was connected to the Seeeduino analog input 0. The ADC reading from the input was mapped from a range of 0 - 5 volts into a range of digital readings of 0 - 1023. This yielded an analog-to-digital conversion resolution of: 5 volts / 1024 units or, 0.0049 volts (4.9 mV) per unit. The temperature measurement resolution was 0.4℃.

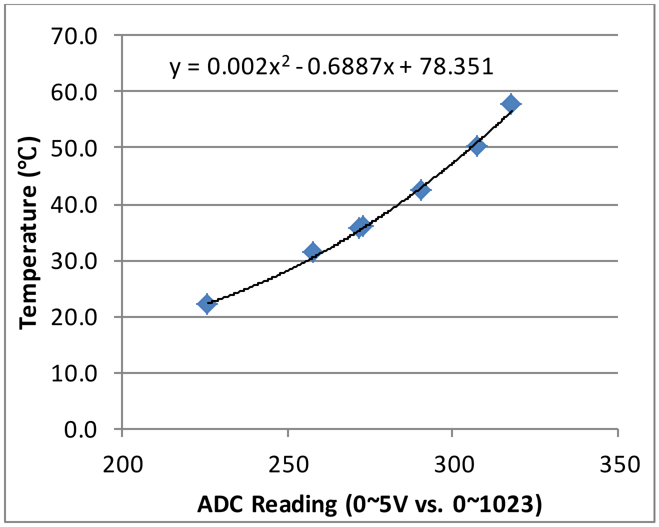


Figure 13 Calibration Curve for the Thermistor

Water flow meter: A Hall Effect flow sensor output a pulse train to measure the water flow rate. In the developed system, two Hall Effect flow sensors were installed into the hot water and cold water pipe lines, respectively, underneath the sink to measure water flow. The flow sensor generated a pulse for about 0.004 liter water flow. Two interrupt inputs of the microcontroller were used to monitor water flow sensor output signal's falling edge. The water usage was accumulated and stored in a temporary variable if the faucet was turned on and off within a time threshold window, e.g. 20 seconds for the developed prototype system.

Hand-hygiene product usage sensor: Each soap, hand sanitizer and paper dispenser had a Wi-Fi module installed inside the dispenser. These Wi-Fi modules were standalone and configured to monitor one of its General Purpose Input Output (GPIO) pins to indicate the hand-hygiene product usage. A relevant triggering signal was connected to the Wi-Fi module's event monitor pin for the dispenser operation status. The Wi-Fi module embedded in the dispensers was set to automatically send a message to the sink node whenever triggered. The Wi-Fi message included its device ID formed by the last four characters of its MAC address and its functionality. The server identified the data source and corresponding event based on the received message. The dispensers were battery operated. The Wi-Fi module consumed very little amount of power and was always under deep sleep mode until the triggering signal was activated to wake it up. The motor control signal of the dispenser was used as the triggering signal which can be easily found in any commercial dispensers.

- Paper towel usage sensors: A commercialized automatic paper towel dispenser (enMotion, Automated Motion-Activated Touch less Paper Towel Dispensers Model: 59462) was modified for hygiene product usage monitoring.
- Soap and hand sanitizer usage sensors: An off-the-shelf battery operated automatic soap and hand sanitizer dispenser (Compact Sensor Pump by Simplehuman) was used and modified with an embedded wireless Wi-Fi module.

4. Design of the Sink Node

As shown in Figure 14, a Seeeduino mega microcontroller board was used to collect hand-hygiene product usage data and maintain corresponding data communications. The Wi-Fi module of the sink node was set to work as a HTTP server and listening on a specific port number for incoming TCP\IP communication request.

Each Wi-Fi module in the mobile node and the dispensers was set to open a TCP/IP connection automatically to the sink node when the sensor was being trigged. A data communication request on the mobile node was managed by its on-board microcontroller; while all the other Wi-Fi modules embedded in the dispensers were set to wake up with a GPIO pin status change signal (trigger).

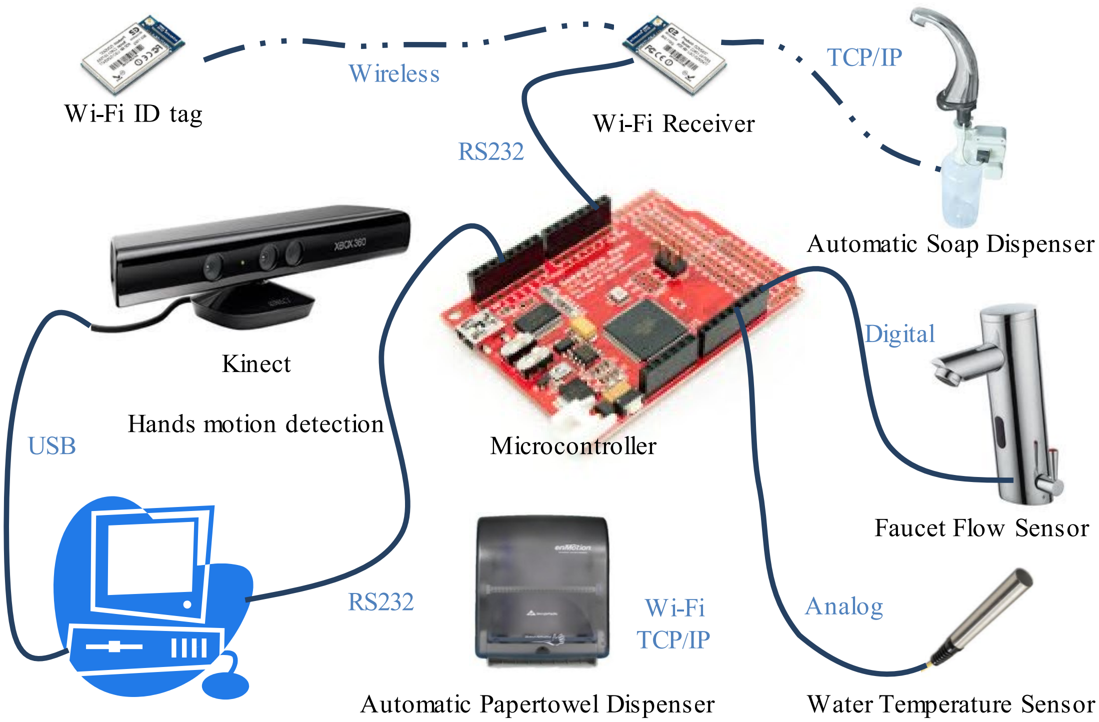


Figure 14 Automatic Hand washing Compliance Monitoring System diagram

As shown in Figure 14, both wired and wireless communication methods were used in the prototype system. The inter-connections between system components were listed below:

- Wired data communication
  - Server to Kinect Sensor: USB
  - Wi-Fi receiver: RS-232
  - Water flow sensor: GPIO (Microcontroller)
  - Temperature sensor: Analog input (Microcontroller)
- Wireless data communication
  - ID tag: Wi-Fi, TCP/IP
  - Soap dispenser Wi-Fi, TCP/IP
  - Hand sanitizer dispenser Wi-Fi, TCP/IP
  - Paper towel dispenser Wi-Fi, TCP/IP

CHAPTER IV

SYSTEM SOFTWARE DESIGN

Two software programs were developed for the developed hand-hygiene monitoring system. One was central control unit software which runs on the Seeeduino single board microcontroller. The second software program was for all the wearable mobile tag Ardunio Mini Pro microcontrollers.

1. Sink node

The sink node collected sensor data and constructed data packets to send to the system server. Figure 15 shows a flowchart of the software program for the sink node. The main routine checked all the sensor status flags to assemble different messages to forward to the system PC\Server. The water flow sensor's signal input was used to calculate the water usage and also used to monitor water ON/OFF status. When the microcontroller detected a water flow, it changed the water flow status to ON and updated water temperature every 2 seconds. When the Seeeduino detected the event of the water faucet being turned off, it then forwarded the water usage data to the server. The water flow sensor signal was measured with interrupt pins of the microcontroller. Two interrupt service routines were used to update the water flow variable, which was also used to reset the water check timer/counter when the water status changed to the OFF state, i.e. water faucet being turned off.

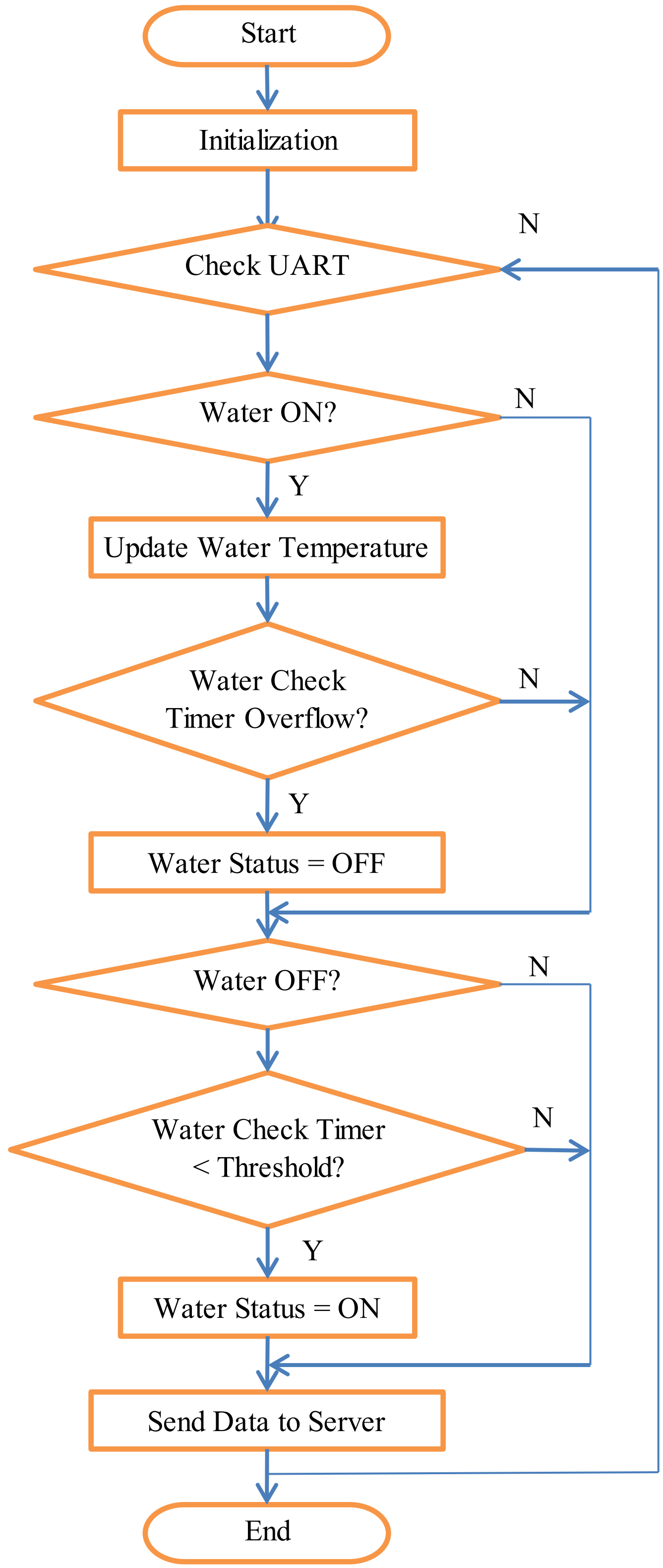


Figure 15 Program Flowchart of the Central Control Unit

2. Mobile node

In the mobile node, the microcontroller, Arduino Mini Pro, periodically awoke the Wi-Fi module, set the module into a command mode, and then sent out an AP scan command to measure the Received Signal Strength Indication (RSSI) value for every surrounding access point. The scanned AP data was processed by the Arduino to remove the non-system based or third party access point data. Only the system related AP data from the anchor nodes was reserved. The processed data was packaged with the reading of each location sensor's information including RSSI, Wi-Fi Channel, MAC, and SSID. The data packet was transmitted to the server via TCP/IP through the sink node for location tracking and human subject identification. Two GPIO pins of the Arduino were used to initiate and monitor the TCP/IP connection. Figure16 shows a software program flowchart for the mobile node.

Most of the time, the Wi-Fi tag received more information from various access points in its surrounding area including the anchor nodes and sink node. To get the RSSI value for its location estimation, the Arduino on the mobile node analyzed the access point list and selected the information from the sink node and anchor node only.

In order to determine the location of the mobile node, the mobile node collected data packages from the Wi-Fi modules on the sink node and the anchor nodes. Each data package was given an ID of its source in a format of "gateway-XXXX", where "XXXX" indicated the last four characters of the Wi-Fi module's MAC address. For each command the Arduino sent to its Wi-Fi module, a specific acknowledge message was replied to confirm the proper execution of the command. Appendix B shows the data communication direction and corresponding example messages between the Arduino and its Wi-Fi module of the mobile node.

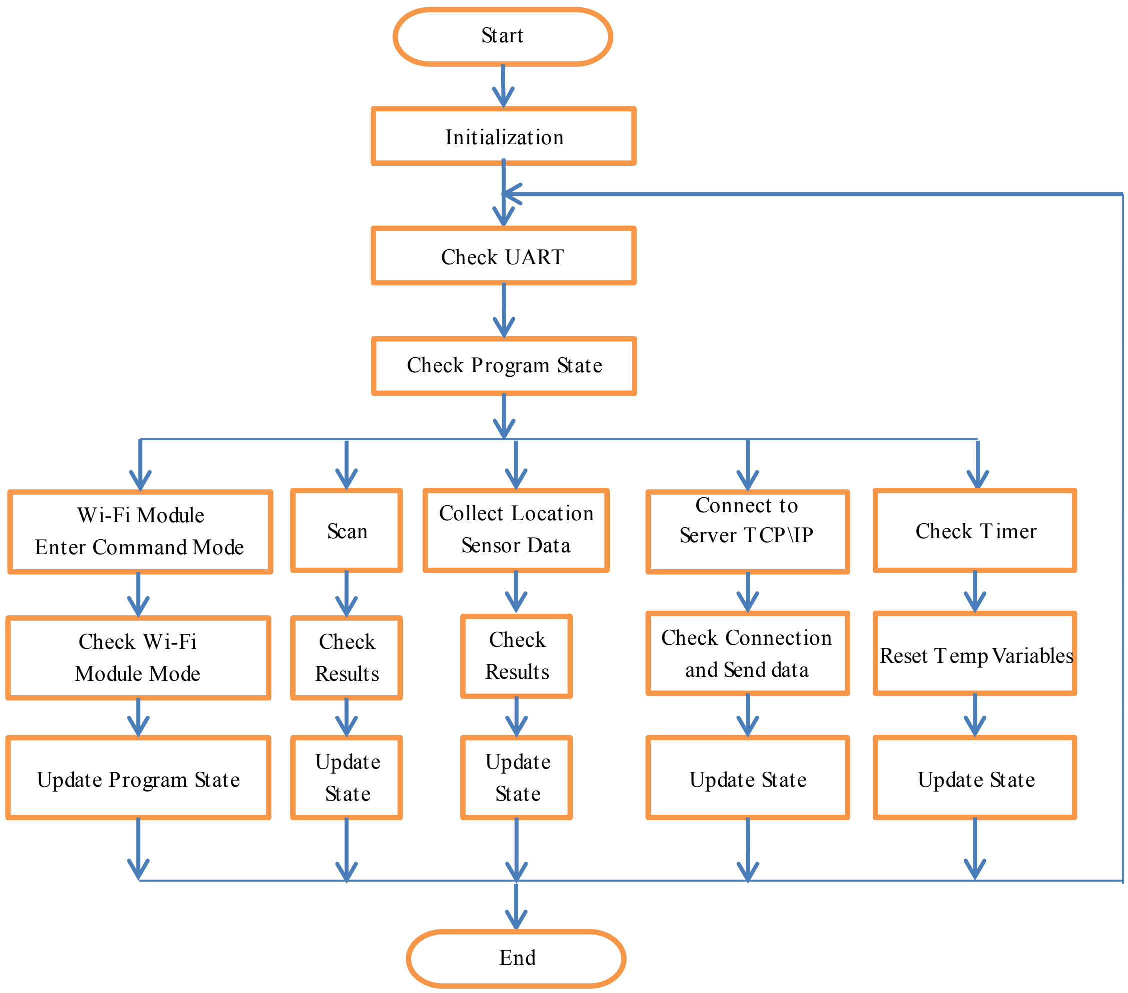


Figure 16 Wearable Mobile ID tag Microcontroller Program Flow Chart

3. Sensor Data Collection

Wifly Wi-Fi Module Configuration

The Wi-Fi modules in the system had four different configurations for different tasks. The Wi-Fi module on the sink node was configured as Access Point (AP mode) which was used to collect data from the mobile node and the sensing nodes (Wi-Fi modules embedded in automatic dispensers). The Wi-Fi module on the anchor nodes was also configured as AP mode used to provide location reference information for the mobile node. The Wi-Fi modules on the mobile node and the sensing nodes were all configures as client mode used to initiate TCP\IP communication request and data transmission to sink node. The detailed configurations of the Wi-Fi modules on different nodes were listed in Appendix C.

RFID-based Localization Algorithm

Besides the Wi-Fi receiver modules on the mobile nodes and the sensing nodes which transmitted the data including ID tag, paper towel usage and soap/sanitizer usage to the sink node, two additional Wi-Fi modules (anchor nodes) were configured as AP mode to be used to track the location of a person. The wearable mobile node was used as an identification device for each individual person. This tag was also capable for Wi-Fi base infrastructure network localization service that could continuously provide the specific subject's location information. The mobile nodes transmitted the data including tag ID to the sink node. As mentioned before, the mobile nodes were set to wake up every 10 seconds, scan for available APs, and send the scan results to the sink node which were then forwarded to system server. The scanned results include all the parameters about the AP in its reachable range, including the AP's Media Access Control (MAC) address, SSID (Service Set Identifier), RSSI (Received Signal Strength Indicator), etc. The RSSI value for each location sensor was used by the server to compute the tag's location with trilateration algorithm.

Room Level Tag Localization

A room level localization system provided a person's location information within a dedicated area, e.g. inside a food processing factory, a restaurant, or a hospital. This room level localization system could be used to estimate the number of possible hand-hygiene opportunities. For example, a hand wash was required if someone entered or left a certain area. This type of location change should be recorded into the system and counted as a hand-hygiene opportunity. An RSSI-based localization system was used for the hand-hygiene opportunity estimate. At least three Wi-Fi

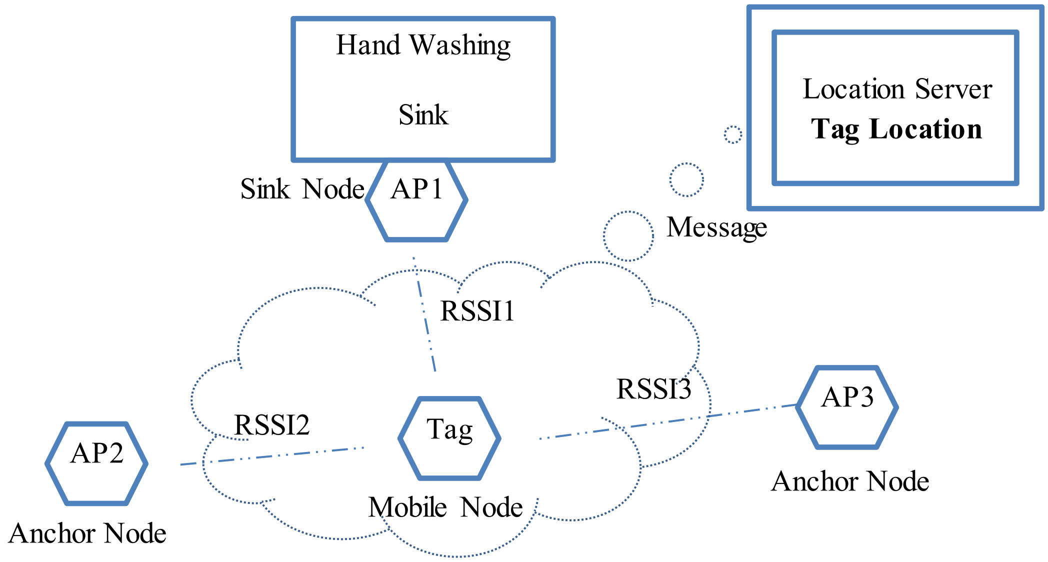


Figure 17 Tag Location Estimation Diagram

modules were used as location sensors for a typical RSSI based localization system. One sink node and two anchor nodes were installed at known locations near the hand washing sink. The sink node served as an access point to transfer data between the Seeeduino microcontroller and the server. As showed in Figure 17, whenever a mobile node entered the covering range of the location nodes, the mobile node collected RSSI values corresponding to each scanned location sensor and forwarded them to the sink node. The sink node then forwarded these RSSI information to the server which calculated the location of the mobile node using the trilateration method described as followings:

- Each location sensor transmitted signals with a fixed preconfigured power at 18mW.
- The RF signals propagation loss was typically given by the following formula:

  $$\text{Path loss in dB} = C + 10 \times n \times \log_{10}(\text{distance}) \qquad [2]$$

  Where n was the path loss exponent, C was a constant which accounts for systems and the signal frequency, and "distance" was the distance between the signal source and the received signal strength checking point. The value of n depended upon the surrounding environment, such as walls or glass windows, carts, and people that could absorb RF energy and cause RF attenuation (for free space propagation, C = 30.6, n = 2).

The distance between the tag and APs could be computed with equation [2], but needed to be combined with empirical actual field test model to find ***C*** and ***n*** with the actual RSSI data readings and with the consideration of tag orientation.

- Trilateration (or multilateration if more than three access point were used)

1) If ***n*** reference nodes were used $AP_1, AP_2, AP_3, \ldots, AP_n$, with known coordinates (X_k, Y_k), and the distances between these APs and the ID tag were calculated using Eq 2, then $\boldsymbol{r_1, r_2, r_3, \ldots, r_n}$. could be found using the following equations [3] where X, Y are the coordinates to be determined:

$$\begin{cases} r_1^2 = (X - X_1)^2 + (Y - Y_1)^2 \\ r_2^2 = (X - X_2)^2 + (Y - Y_2)^2 \\ \quad\quad\quad \ldots \\ r_n^2 = (X - X_n)^2 + (Y - n)^2 \end{cases} \quad [3]$$

2) By subtraction of each of the other equation from the first equation of [3] and writing $\boldsymbol{b_{i1}}$ as:

$$b_{i1} = \frac{1}{2}\left(X_1^2 - X_i^2 + Y_1^2 - Y_i^2 + r_i^2 - r_1^2\right) \quad (i = 2, 3, \ldots, n) \quad [4]$$

Equation [3] could be linearized as:

$$\begin{cases} (X_1 - X_2)X + (Y_1 - Y_2)Y = b_{21} \\ (X_1 - X_3)X + (Y_1 - Y_3)Y = b_{31} \\ \quad\quad\quad \ldots \\ (X_1 - X_n)X + (Y_1 - Y_n)Y = b_{n1} \end{cases} \quad [5]$$

Equation [5] could be written as matrix form:

$$A\mathbf{X} = b$$

Where:

$$A = \begin{pmatrix} (X_1 - X_2) & (Y_1 - Y_2) \\ (X_1 - X_3) & (Y_1 - Y_3) \\ \cdots & \\ (X_1 - X_n) & (Y_1 - Y_n) \end{pmatrix}, \quad X = \begin{pmatrix} X \\ Y \end{pmatrix}, \quad b = \begin{pmatrix} b_{21} \\ b_{31} \\ \cdots \\ b_{n1} \end{pmatrix} \qquad [6]$$

Then Equation system [6] was solved as a linear least squares problem

$$\text{Min}||AX - b||^2 \qquad [7]$$

Equation [7] then was solved by QR-factorization and singular value decomposition to find the location coordinates of the mobile node. However, the RSSI value from each anchor node was very sensitive to the environment, thus could be affected by reflection, refraction or diffusion because of different surface conditions and structural features. There might be also other factors that could affect the propagation of the 2.4G Wi-Fi carrier signal. Hence, the localization based on RSSI was accurate to a resolution of less than 3 to 5 meters which was adequate for room level localization applications.

Identification Location Detection Near Hand Washing Sink Target Area

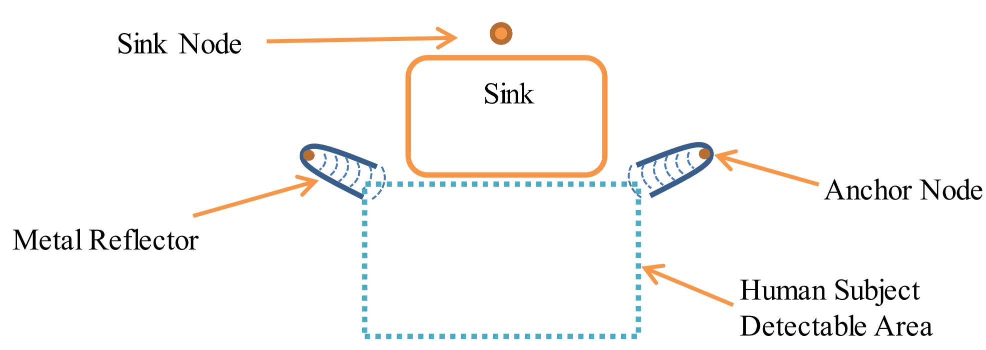


Figure 18 Human Subject Target Area Detection Location Sensor Installation Diagram

In order to differentiate two individuals within a hand washing area, the anchor nodes were put into a metal parabolic reflector that confined the signal propagation to a certain direction and also increased the signal intensity in that direction. Figure18 shows the system location sensor installation diagram viewed from the top over the sink counter. The rectangular area with a dashed line in Figure 18 indicated the target area that was considered as the hand washing area. If more than one person was in the room, the person standing inside the detectable hand washing area should have the distinct signature of RSSI values with respect to the anchor and sink nodes. Thresholds were selected, calibrated and tested to be used to localize a mobile node in the sink area for each location sensor.

4. Hands Tracking and Motion Detection Algorithm

Though the Kinect sensor could provide location information of up to 20 joints from one tracked subject, it required a person to be inside the sensor's field-of-view at least above the waist and with head and shoulder also in view. In hand-washing monitoring, most of the time, a sink was installed right next to a wall. It was difficult to be able to maintain the whole body or even the waist, the head and the shoulders in the field of view of the Kinect sensor. Hence, the built-in Skelton tracking function in the Kinect sensor SDK was not able to track a person with just the hands, arms and body partially in the sensor's field-of-view. In this research, an image processing algorithm was developed to process the depth data stream for hands localization. The user interface of the program is shown in Figure 19. Figure 20 shows a flowchart of the image processing program that explained the procedures of hand tracking and motion detection.

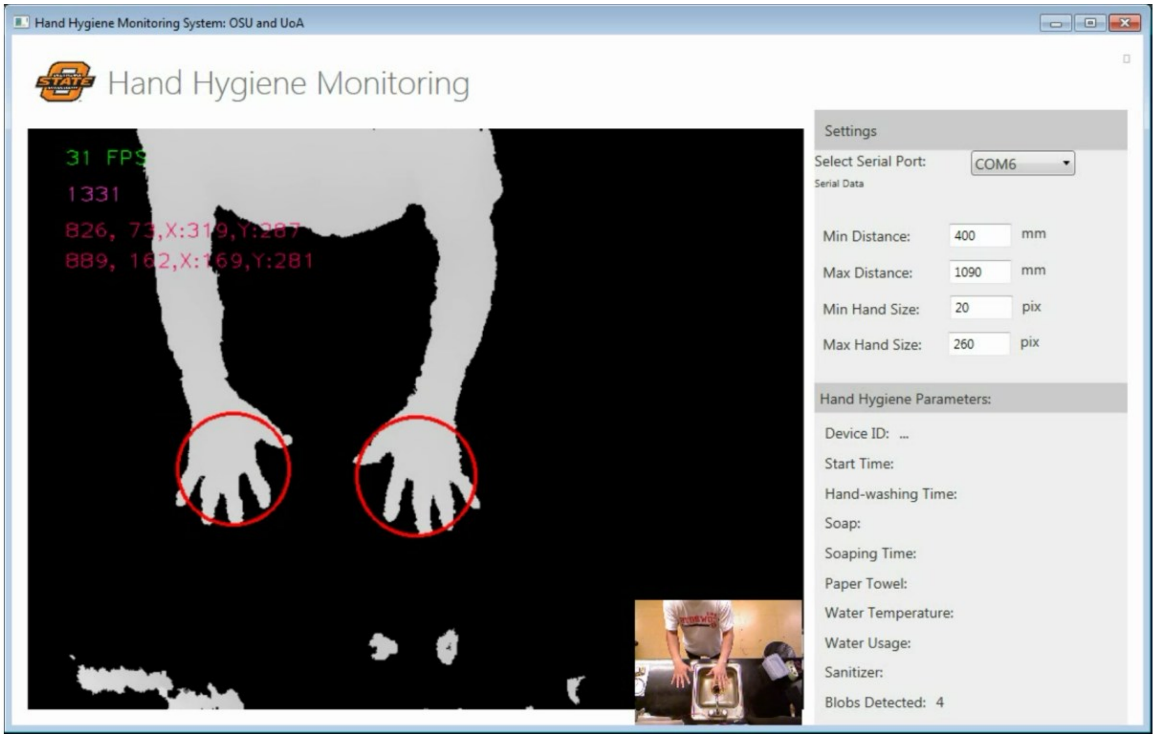


Figure 19 Hand-hygiene Monitoring System Human Interface

The image processing algorithm could be divided into four steps: data preparation, image processing, hands tracking, and feature extraction. The program was developed with Visual Studio 2010 Professional Visual C#2010(Microsoft, Redmond, WA, U.S.) under Window 7 operation system. The required support drive and library including Kinect for Windows SDK v1.6 and emgucv-windows-x86 2.3.0.1416 (Microsoft, Redmond, WA, U.S.). Emgu CV (Free Software Foundation, Inc. Boston, MA USA) was a .Net wrapper for OpencV (Intel Corporation, Willow Garage, Itseez) that allowed programmers to use .Net for image processing with C # instead of C++.

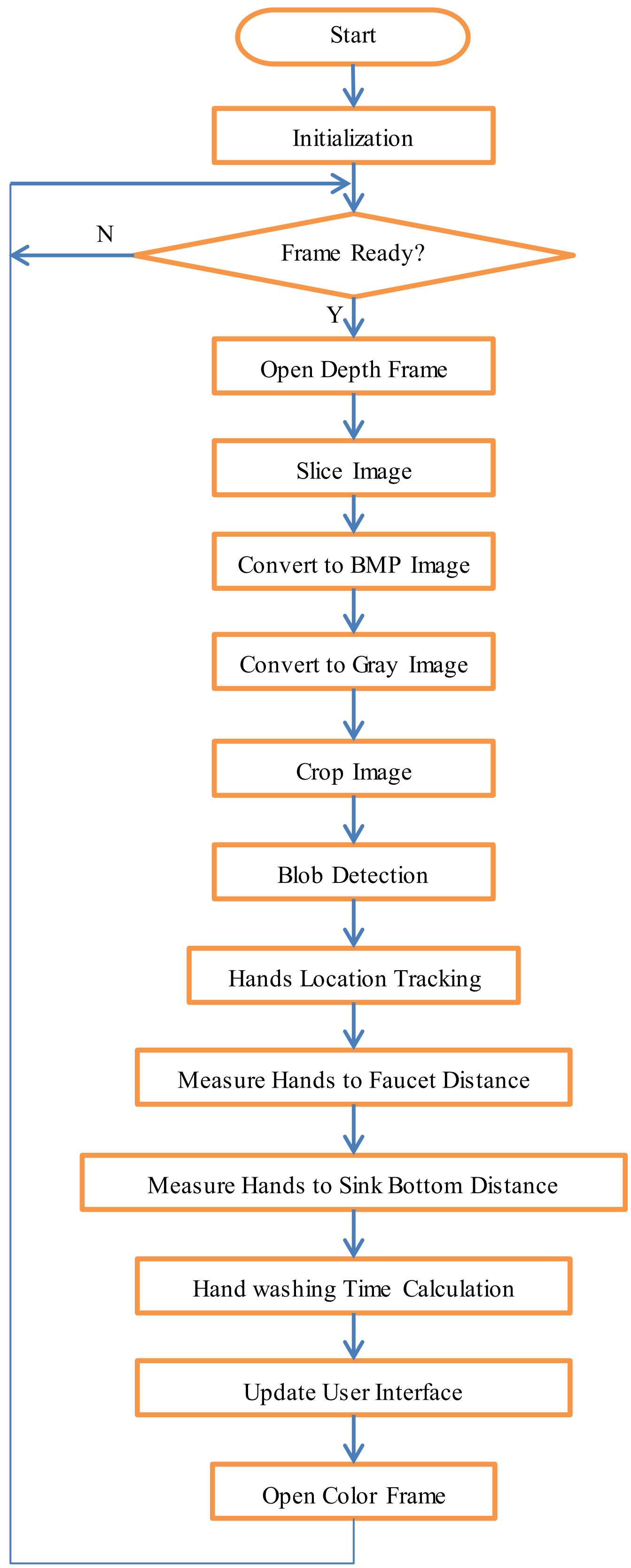


Figure 20 Image Processing and Hands Motion Detection Flow Chart

Data Preparation

Though the raw data from the IR camera was currently not available for user access, the Kinect sensor provided the third dimension depth data stream with the support of the Kinect Windows SDK. The depth data stream could be accessed using the following procedures for further image processing.

- Create an application project with Microsoft visual studio
- Add Microsoft.Kinect.dll reference into the project
- Add the image element as a container for the depth frame data as shown in Figure19, where the image box is shown with a black background.
- Initialize the Kinect sensor object to discover and maintain a reference to the Kinect sensor, and set the data retrieve method and start the sensor.
- Retrieve the data once the data frame event is fired by the Kinect sensor

Image Processing

The raw data of a depth frame contained 640*480 pixels depth data with 16-bits depth data for each pixel. The raw data was then converted to a gray image representing a distance map. The pixel data processing procedures is shown in Figure 21. The left side diagram shows the pixel data manipulation procedure and right column is the corresponding pixel data format.

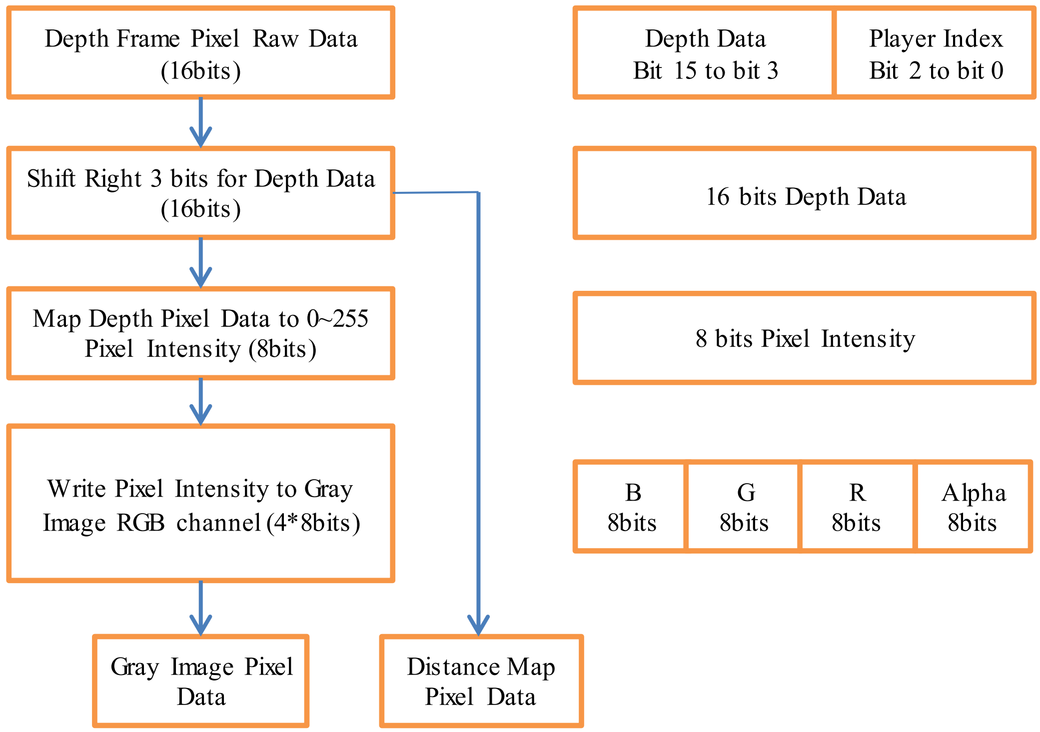


Figure 21 Depth Frame Pixel Data Processing Procedure and Its Data Structure

The depth frame pixel data mapping was also called as a “Slice Image” in Figure 20. Depth frame pixel data contained the distance measure for the object within the IR camera’s field-of-view. It ranged from 80 to 4000 mm for a valid distance measurement. Two thresholds were used to slice the depth frame image to keep the pixel that within the two thresholds that defined as minimal and maximal detectable distance. Pixels with a depth measure outside of the defined threshold were set to 255 which were the black background of the gray image shown in the user interface depth video stream. The minimal distance threshold was defined a little bit greater than 800mm based on the Kinect sensor’s minimal valid measurable range. The maximal distance threshold was set to a little bit smaller than the distance between Kinect sensor and the hand washing sink counter. Any objects outside of this range, from Kinect sensor to sink counter, were removed or sliced out from the depth data frame. Pixels with a depth measure inside the defined range were mapped to a value between 0 to 255 pixel intensities were written into a gray image with four channels RGBA formatted pixels. This gray image was also known as the BitmapSource Image that was used to feed the user interface to display a real-time video from the Kinect IR camera.

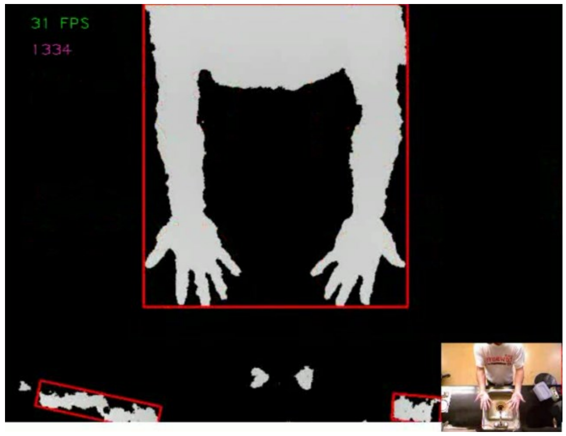
Hands Tracking

Hands tracking methods with a depth image processing from a Kinect sensor was developed based on the mounting setup of the Kinect sensor for this research shown in Figure 10. This mounting method was designed based on practical considerations in the food processing plants.

A blob detection image processing method was used for hands location tracking for the depth image. The blob detection method was based on the OpenCV library's ***findContours*** method. The depth gray image was converted to an 8bit single channel gray image to be used for the find Contours function with Emgu.CV.CvEnum.CHAIN_APPROX_METHOD.CV_CHAIN_APPROX_TC89_L1 method (Teh and Chin, 1989). This method extracted the dominant points and features to a uniquely specify a curve via a scale-space filtering with a Gaussian kernel for a cardinal curvature points finding algorithm.

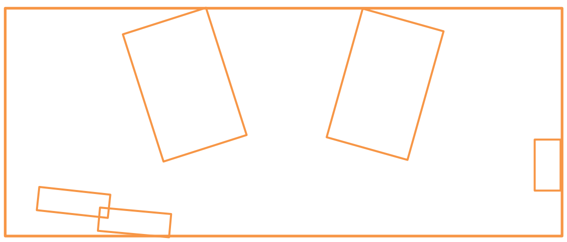
Through the analysis of the captured images from the Kinect sensor, it was found that the hands locations of person inside the camera field-of-view was always closer to the bottom edge of the image during the hand washing based on the mounting position of the Kinect sensor for this research (Figure 10).

In the developed hand tracking algorithm, if a person had his body mostly in the field-of-view of the IR camera, this person's body was recognized as a blob, Figure 22a. An image cropping method was used on the depth image to extract only the hands and partial arms for hand location tracking. Figure 22b showed the depth image after image cropping. Figure 22c showed an illustration diagram of blob detection with minimal wrap rectangle marked for each detected blob

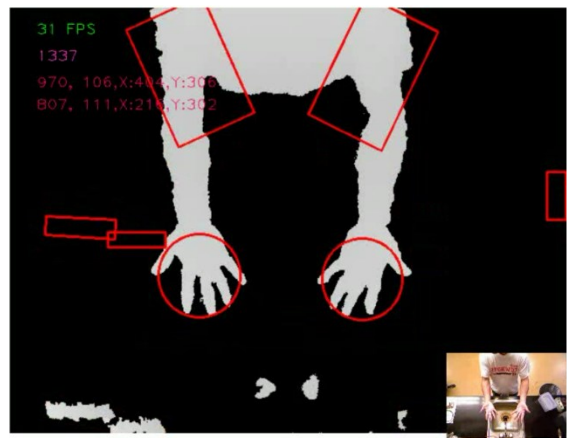


(a) Blob Detection for Original Image

(b) Cropped Image



(c) Illustration Diagram for Blob Detection for the Cropped Image



(d) Hands Location (Circles) Tracking Based on Detected Blobs Locations (Blobs were Drawn on Top of the Original Image)

Figure 22 Depth Image Blob Detection and Hands Location Tracking

for the cropped image. In Figure 22d, the rectangle for each detected blob was drawn over the original gray image. The rectangles were all shifted up after the image cropping. The image cropping area was based on the size of the initial detected blob height. The shift up distance was the pixel number of the 3/5 of the length of full body blob rectangle shorter side (Figure 22a). Hands locations were estimated based on the center location, orientation, size and the shift up pixel distance of the detected blobs. The number of maximal detectable hands in one frame was set as two. If the number of the detected blob was greater than two, based on the blob sizes and locations, only two blobs were kept for hands location tracking, as showed in Figure 22c four blobs were detected but only two of the blobs were kept and marked with circles indicate tracked hand locations. Two size thresholds and location criteria were used to check if each detected blob could be a possible hand or not. If not then it was removed from the detected blob list. The two thresholds were miniHandSize and maxHandSize which could be adjusted on the user interface.

After the image cropping and blob detection, a GetMinAreaRect method was used to find hand locations. The hand location could be found as one of the ends of the rectangle that closer to the bottom edge of the depth image. The center of the hand location was calculated as the center of the rectangle based on its length, height and orientation. Once the hand center location was found, a red color circle was drawn on the depth image with the diameter of the rectangle width to indicate an estimated hand location and size.

The hand depth measure was not a single pixel distance value in the depth map. It was a calculated average distance by count the number of valid distance values inside the red circle of the tracked hand. The distance from the bottom of the sink to the Kinect sensor was also measured the same way with a 20*20 pixel rectangular area that was mapped to the sink bottom right under the faucet, so whenever a hand showed underneath the faucet outlet, the sink bottom distance reading changed and was detected by the system.

The distance was calculated using Equation [8]:

$$Distance = \frac{Sum\ of\ the\ pixel\ distance\ data\ with\ in\ the\ trackde\ hand\ circle}{Number\ of\ valid\ pixels} \qquad [8]$$

where the number of valid pixels was counted based on the number of pixels in which the pixel's data was with in the predefined minimal and maximal distance range.

Feature Extraction

Once the hands location information was acquired, several parameters including the distance between a hand and the faucet; the distance between the Kinect sensor and the faucet; the distance between the Kinect Sensor and the bottom of the sink, and the number of detected hands were all calculated. The location of Kinect sensor, faucet and sink were all fixed, so except the distance between hands and faucet, all the other distance measures did not need the hands location information and could be extracted from distance map directly with the location coordinates of the sink and faucet based on the actual hand washing station distance measure. As showed in Figure 22(a), the text information on the upper left corner was the extracted hand location features with the displayed format as depth image frame rate (frames per second) for the first line, Kinect sensor to sink bottom distance (mm) as the second line, Kinect sensor to hand distance (mm), hand to faucet distance (mm), hand X and Y coordinate (pixel) in depth frame as third line. If two hands were detected the fourth line text would be the second hand's location information with the same format as the third text line.

5. Sensor Data Collection and Data Management Software

For a single hand washing event, the data collected from different sensors might have different formats. All Wi-Fi enabled sensor data packets received by the system server were disassembled and repacked with a predefined format to form a data message by the Seeeduino microcontroller. All these sensor messages were sent to the server in series by the Seeeduino microcontroller based on the actual message receiving time from each sensor. Each message contained the information that could be used to identify the data source. For each hand washing event, the data collected by the server were packed into a data structure format to be stored in a log file. Figure 23 showed the flowchart of the data managing and processing procedure on the server. Each data record of the data log file had a data format of: Device ID, Start Time, Soap Usage(Drops), Soap

Time, Lathering Time (Second),Sanitizer Usage (Drops), Sanitizer Time, Paper towel Usage (Pieces), Paper Time, Water Temperature (F), Hand Washing Time (Second), Recording Time.

By monitoring the hands location of a tracked person near the hand washing station, the server software extracted the parameters including soaping time, rinsing time, total hand washing time beside the data collected from the sensing unit. These information extraction methods were explained in Table 2.The unit, definition and calculation criteria of each parameter in the data log file were listed in the table.

Table 2 Definition and Calculation Criteria for the System Data Log File

Parameter(unit)	Definition	Calculation Criteria
Device ID (IDEN-XXXX)	Device identification information of a wearable mobile tag that detected by the system	Tag ID information was acquired from microcontroller. The server program monitors the serial communication data from MCU for the device ID by checking the received message. The ID packet content always start with "IDEN"
Start Time (MM/DD/YYYY H:M:S AM/PM)	Hand washing start time	The system software monitors the water flow and soap dispenser status. If water being turned on or soap dispenser being activated, the current computer system time was recorded as hand washing start time.
Soap Usage (drops)	The drops of soap used for current hand washing	The server program monitors the serial communication data from MCU for message

	event	that contains "SOAP", which was generated by the Wi-Fi module embedded inside the soap dispenser
Soap Time (MM/DD/YYYY H:M:S AM/PM)	Time of soap being used	Current computer system time, when the server program detects the soap dispenser being activated
Lathering Time (second)	Soap lathering time	Once system software detects the soap dispenser being activated, it will start a timer for this parameter. This timer adds an average frame time (1second/frame rate) for every processed frame, if only one hand was detected (lathering) and the system software detects hand under water running faucet over one second, the lathering timer will be stopped.
Sanitizer Usage (drops)	The drops of sanitizer used for current hand washing event	The server program monitors the serial communication data from MCU for message that contains "SANI", which was generated by the Wi-Fi module embedded inside the sanitizer dispenser
Sanitizer Time (MM/DD/YYYY	Time of sanitizer being used	Current computer system time, when the server program detects the sanitizer dispenser being activated

H:M:S AM/PM)		
Paper towel Usage (piece)	The number of how many pieces of paper towel being used	The server program monitors the serial communication data from MCU for message that contains “PAPE”, which was generated by the Wi-Fi module embedded inside the paper towel dispenser
Paper Time (MM/DD/YYYY H:M:S AM/PM)	Time of paper towel being used	Current computer system time, when the server program detects the paper towel dispenser being activated
Water Temperature (°F)	Water temperature	The MCU reads the temperature sensor once per iteration and stores the reading in to a 128 elements temperature buffer. If water being turned on, MCU will send an averaged buffer temperature reading once every two seconds. The server program monitors the serial communication data from MCU for message that contains “TEMP”, and extracts the water temperature data and put it in to another buffer. One averaged temperature data from this buffer was recorded into the log file for one hand washing event.

Hand Washing Time (second)	Hand washing time including wetting time and rinsing time	The system softer monitors the hand location. Once it detects the hand under a water running faucet, it will start a timer for this parameter. This timer will add an average frame time (1second/frame rate) for every processed frame.
Recording Time (MM/DD/YYYY H:M:S AM/PM)	Logging time of current record	Current computer system time, if the server program detects no event happened for the past 50 seconds.

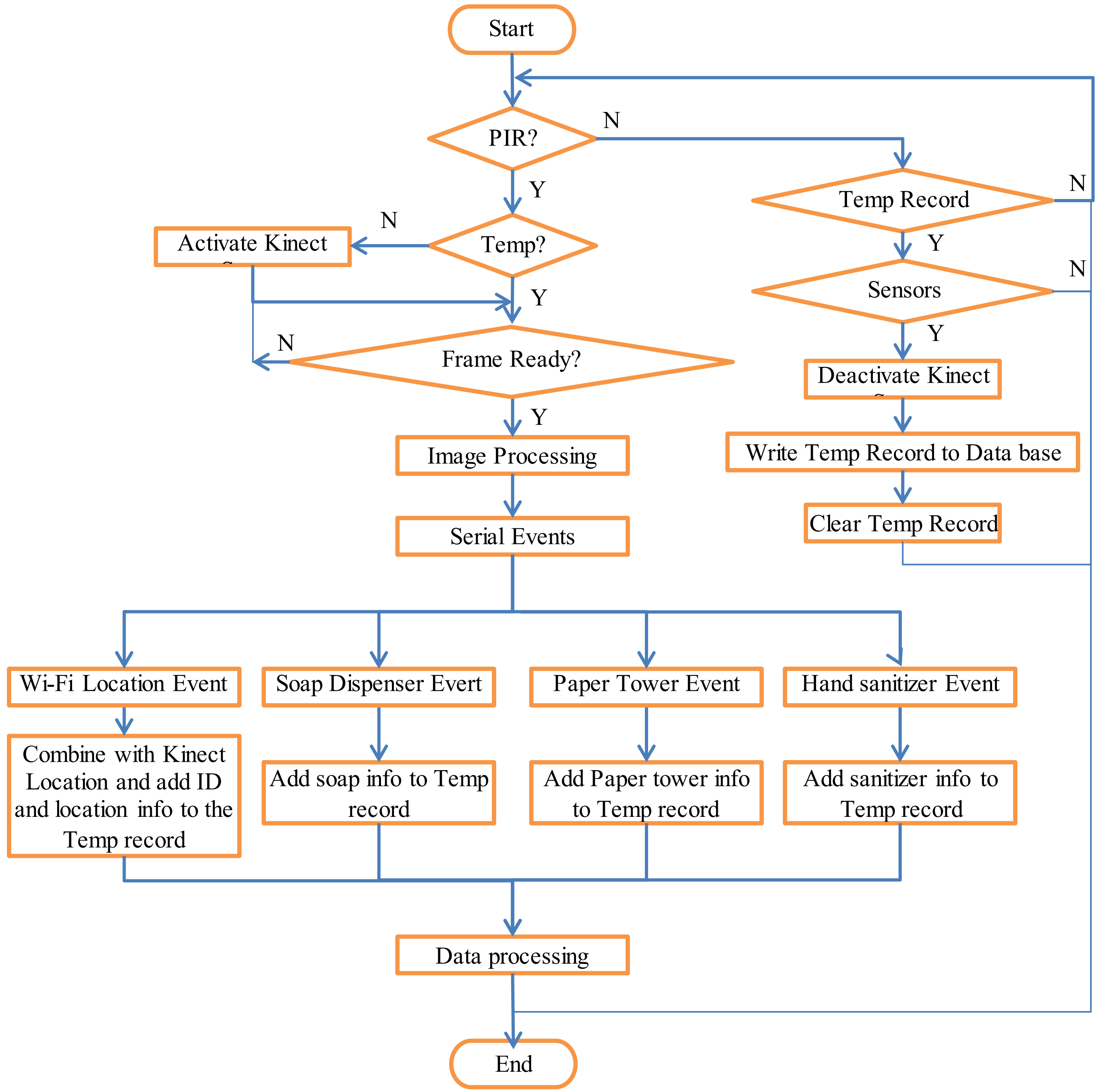


Figure 23 Sensor Fusion and Data Management Flow Chart

CHAPTER V

EXPERIMENT DESIGN

A series of testing experiments was developed to validate the performance of the hand hygiene monitoring system, which included preliminary functionality tests followed by system evaluation tests. Preliminary functionality tests were conducted for the performance evaluation of individual component of the developed hand-hygiene monitoring system. The system evaluation tests were designed to evaluate the performance of the developed system on hygiene monitoring.

1. Preliminary system tests

Evaluation on Water Temperature and Flow Measurements

To evaluate water temperature sensors, five temperature sensors (thermistors) were used to measure water temperature in the range of 24℃ to 54℃ (70~130 F), which covered the appropriate water temperature range for normal hand washing. A digital thermometer with thermocouple temperature sensor was used to provide reference temperature measurements to test the thermistors. The sensor probe tip of the digital thermometer was mechanically banded with each of the five thermistor tips and put into a cup. Different temperature water samples were prepared and placed in the cup to allow variations in temperature from the thermistor and digital thermometer. Temperature difference (ΔT) was calculated by subtracting thermistor temperature measurements from the reference temperature. Error was calculated by taking the ratio of the temperature difference divided by reference temperature.

$$E_T = \frac{T_{Thermocouple} - T_{Thermistor}}{T_{\text{Thermocouple}}} \qquad [9]$$

Two water flow sensors were tested at different water flow rates between 1 and 5 liters per minute. The water flow sensors were installed into hot and cold water pipes horizontally. The microcontroller measured water volume was compared with computed water volume based on water mass delivered divided by water density. Water mass delivered was measured with a digital balance (TR-81020, Denver Instrument Company, Denver USA). The measurement errors of the water flow sensors were calculated by dividing the difference in measured water and balance determined volume by balance determined volume.

$$E_F = \frac{V_{True\ Volume} - V_{Flow\ Sensor\ Measured\ Volume}}{T_{\text{True volume}}} \times 100\% \qquad [10]$$

Evaluation of Distance Measurements Using the Kinect Sensor

To evaluate the accuracy of distance measurements, the distances were measured between an object and the Kinect sensor at three different angles (front, left and right 20º). During the measurements, the object was placed at different position along the three angles, respectively. Since the Kinect sensor was mounted at 1.2 meters above hand wash sink counter, the testing range was between 0.8m to 1.1m. A reference distance was measure with a tape measure. Errors were calculated by subtracting system measured object distance from tape measured distance.

$$E_D = \frac{D_{True\ Distance\ from Tape\ Measure} - V_{Kinect\ Sensor\ Measured\ Volume}}{T_{\text{True Distance}}} \times 100\% \qquad [11]$$

Evaluation of Sensing Node for Event Trigging

To test the event trigging reliability of the sensing nodes, the Wi-Fi modules embedded in the hand-hygiene product auto dispensers were used to simulate the event sensing. A simulated trigging signal was periodically generated every10 seconds by a computer and sent to the Wi-Fi module to make a data transmission with exactly same format message as the prototype system.

Another Wi-Fi module was configured as the sink node to receive the messages. The number of messages sent and received were recorded to check the successful message delivery rate. The successful message receiving rate was calculated as:

$$R_m = \frac{Number\ of\ Message\ Sent - Number\ Message\ Received}{Number\ of\ Message\ Sent} \times 100\% \qquad [12]$$

Message delay time, in second, was also collected to evaluate the system event sensing response. The message delay time was calculated as:

$$T_{Delay} = Message\ Received\ Time - Message\ Sending\ Time \qquad [13]$$

Evaluation of Localization Algorithms Using Wi-Fi Modules

To test the applicability and resolution of the mobile node for location tracking and the relation of mobile tag RSSI readings vs. distance, three Wi-Fi module location sensors were configured as access points and set up in an outdoor open space (one sink node and two anchor nodes). A mobile node (wearable ID tag) was mounted on a tripod and used to collect RSSI values from the three location sensors. The location sensors and mobile node were all set at a height of 1meter above ground. The RSSI values were collected at the distance between the location sensors and mobile Wi-Fi node in the range of 0.9 to 25 meters along with a tape measure laid on the ground as distance reference as shown in Figure 24. The mobile node scanned the surrounding AP and takes the RSSI reading from each AP node then forwards the RSSI values to the sink node which was collected by the laptop as data records. The initial testing distance started from 0.9 meters and increased by 0.9 meters per step.

Figure 24 RSSI vs. Distance Measurement Field Test Setup

To test the target area localization performance, As showed in Figure 25, two anchor nodes were set beside sink counter with a distance of 1.2 meters in between two anchor nodes. A mobile node was set at the same height as the counter (1m) to collect RSSI values from two anchor nodes to estimate the mobile tag location thresholds\signature within the human subject detectable area. Each collected RSSI value was calculated by average five continuous readings.

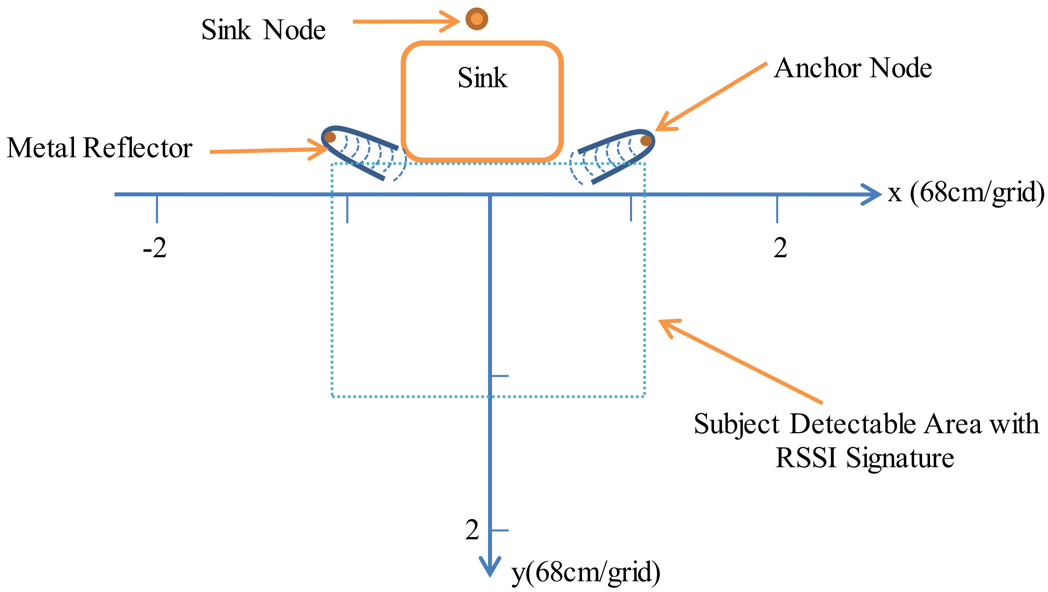


Figure 25 WEARABLE MOBILE TAG RSSI SIGNATURE TESTING SETUP

2. Evaluation on Hand-Hygiene Monitoring

Experiments were designed which included 15 volunteers to conduct 35 hand washing events. During the hand-hygiene events, parameters from all the sensors were recorded for system evaluation. Participants were asked to perform a hand wash without special training. A poster describing recommended hand washing procedures was on the wall beside a sink. Liquid soap and paper towel were provided from the dedicated auto dispensers, respectively. The participants were asked to go through the poster first, and then start to wash their hands. The sensor data were collected. To validate the sensor data, the hand washing time and the time for soap usage were manually recorded for each single hand washing event.

CHAPTER VI

RESULTS AND DISCUSSION

1. Preliminary system tests

Temperature Sensor Measurement Data

Table 3 shows the temperature measurement data from five thermistors in the Sensor columns. The Ref temperature data was measured from the digital thermometer

Table 3 Temperature sensor evaluation data

Sensor1 (℃)	Ref1 (℃)	Sensor2 (℃)	Ref2 (℃)	Sensor3 (℃)	Ref3 (℃)	Sensor4 (℃)	Ref4 (℃)	Sensor5 (℃)	Ref5 (℃)
24.9	25.6	25.0	25.5	24.9	25.4	24.9	25.4	24.8	25.4
28.7	29.4	28.8	29.5	28.9	29.6	29.0	29.7	29.0	29.7
32.8	33.1	32.9	33.1	32.9	32.9	32.7	32.8	32.5	32.7
37.0	37.0	37.3	37.2	37.7	37.4	37.8	37.4	37.8	37.5
41.9	40.9	41.9	40.8	41.4	40.8	41.4	40.3	40.6	40.0
46.4	44.8	46.6	45.0	47.0	45.4	47.4	45.4	47.5	45.6
52.1	49.8	51.9	49.6	51.6	49.4	51.0	48.5	50.8	48.1
55.5	53.1	55.9	53.4	56.5	53.8	56.6	53.9	56.7	54.1
57.9	55.4	57.8	55.1	57.3	54.9	57.5	54.8	57.3	54.3

Table 4 showed the temperature measurement data error compared with measured reference temperature. First column was a reference as temperature range. As shown in table 5, the average temperature measurement error was less than 1.1℃, and the average error was less than 3.1%. The most accurate temperature measuring range was from 32℃ to 37℃ with the accuracy less than 0.4℃ (< 1%).

Table 4 Temperature Sensor Measurement Error

Ref (°C)	Error 1		Error 2		Error 3		Error 4		Error 5	
	Ratio	ΔT	Ratio	ΔT	Ratio	ΔT	Ratio	ΔT	Ratio	ΔT
25.4	2.6%	-0.7	2.0%	-0.52	2.1%	-0.54	2.2%	-0.57	2.5%	-0.63
29.6	2.6%	-0.8	2.3%	-0.69	2.3%	-0.67	2.2%	-0.64	2.2%	-0.64
32.9	0.9%	-0.3	0.6%	-0.18	0.2%	-0.08	0.3%	-0.09	0.7%	-0.23
37.4	0.0%	0.0	0.1%	0.06	0.8%	0.31	1.1%	0.39	0.9%	0.34
40.8	2.4%	1.0	2.5%	1.03	1.6%	0.63	2.7%	1.11	1.6%	0.64
45.4	3.6%	1.6	3.5%	1.59	3.4%	1.56	4.2%	1.91	4.1%	1.86
49.4	4.7%	2.3	4.8%	2.39	4.5%	2.23	5.1%	2.48	5.6%	2.70
53.8	4.5%	2.4	4.7%	2.53	5.0%	2.70	5.0%	2.69	4.8%	2.58
54.9	4.5%	2.5	4.9%	2.73	4.3%	2.36	5.0%	2.73	5.5%	2.98
Average	2.9%	0.89	2.8%	0.99	2.7%	0.95	3.1%	1.11	3.1%	1.07

Water Flow Sensor Measurement Data

Two water flow sensors were tested for cold and hot water flow volume measurements. Table 5 showed the water flow sensor measurement data. For approximate 1.6 liter water volume measurement, the flow sensor measured water volume results was very close to the true water volume that was acquired by converting the water weight measure from a digital balance. The flow sensor measurement error was less than 2%.

Table 5 Water Flow Sensor Measurement Data

Measured Volume (L)	Time(s)	True Volume (L)	Flow Rate (L/min)	Water Pipe Line	Error	Error Ratio
1.58	59	1.61	1.61	c	-0.03	-1.9%
1.61	58	1.61	1.67	h	0	0.0%
1.6	57	1.6	1.68	h	0	0.0%
1.54	51	1.57	1.81	c	-0.03	-1.9%
1.59	52	1.62	1.83	c	-0.03	-1.9%
1.47	48	1.5	1.84	c	-0.03	-2.0%
1.54	35	1.54	2.64	h	0	0.0%
1.59	31	1.62	3.08	c	-0.03	-1.9%
1.47	24	1.47	3.68	h	0	0.0%
1.55	24	1.55	3.88	h	0	0.0%
1.55	24	1.54	3.88	h	0.01	0.6%
1.52	23	1.54	3.97	c	-0.02	-1.3%
1.53	22	1.55	4.17	c	-0.02	-1.3%
1.55	22	1.57	4.23	c	-0.02	-1.3%
1.55	22	1.55	4.23	h	0	0.0%
1.49	20	1.49	4.47	h	0	0.0%
1.58	21	1.6	4.51	c	-0.02	-1.3%
1.66	22	1.67	4.53	c	-0.01	-0.6%

Kinect Sensor Distance Measurement Data

Table 6 shows the distance measurements by the Kinect sensor.

Table 6 Kinect Sensor Distance Measurement Data

True Distance (mm)	Left (mm)		Center front (mm)		Right (mm)	
	Distance	Error	Distance	Error	Distance	Error
865	826	-39	857	-8	817	-48
950	904	-46	943	-7	909	-41
1040	995	-45	1027	-13	996	-44
1103	1067	-36	1069	-34	1063	-40
Average	**-41.5**	**-42**	**-15.5**	**-16**	**-43.3**	**-43**

The results showed that the measured distances were all underestimated compared with the reference distance measure. The tested sensor range was from 0.8m to 1.1m that in between the mounted Kinect sensor and hand washing sink counter. The distance measurement error increased as the object moves away from the Kinect sensor front center line. The possible major reason for the underestimate was the distance estimation algorithm. The total number of the valid pixels calculation method needs to be improved for better distance measure.

Sensing Node Evaluation

A total of 517 simulated trigging messages were generated by a computer with 514 received messages from a simulated sink node. This showed the HH system have a 99.4% event sensing accuracy for hand-hygiene product usage monitoring. However, 54 out of 517 messages have an averaged 2.4 seconds message delay on the receiving time.

Location Sensor Evaluation

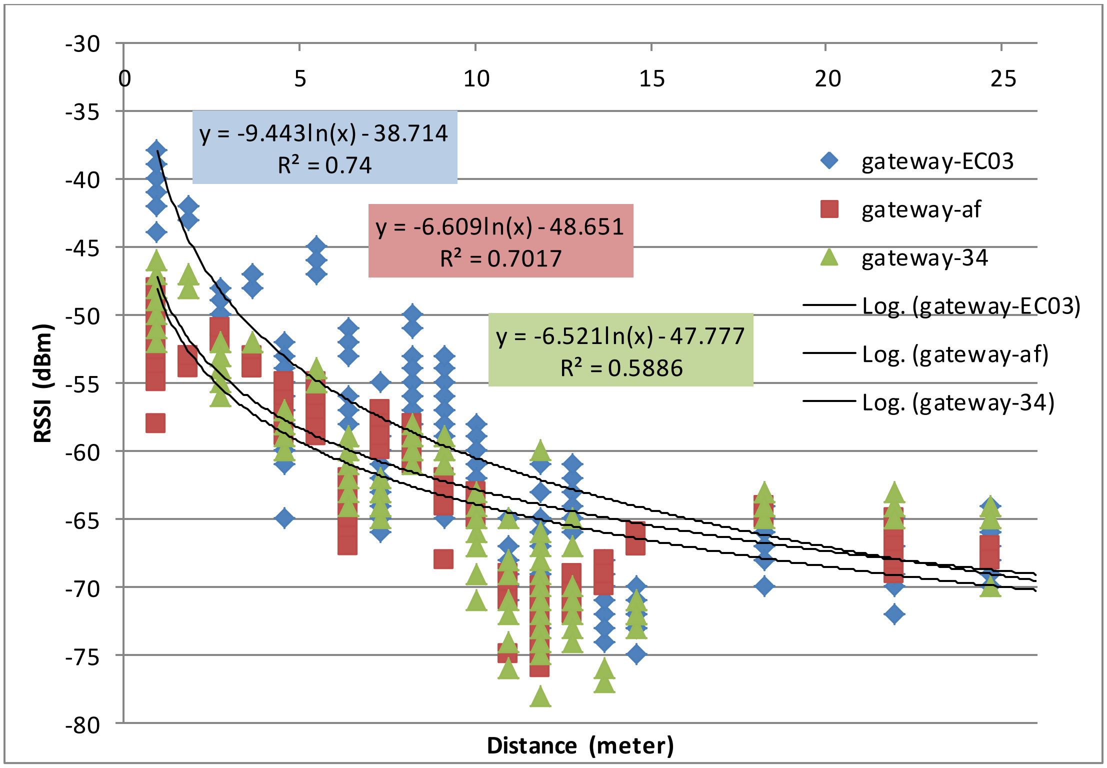


Figure 26 Location Sensor RSSI Value vs. Distance

Figure 26 shows the scatter plot of the collected RSSI value from the access points acquired by the mobile node verses the distance between the access point node and mobile node. The RSSI value rapidly dropped as the distance increased from 1 meter to about 10 meters and the relationship became nearly flat after the distance increased above 15 meters.

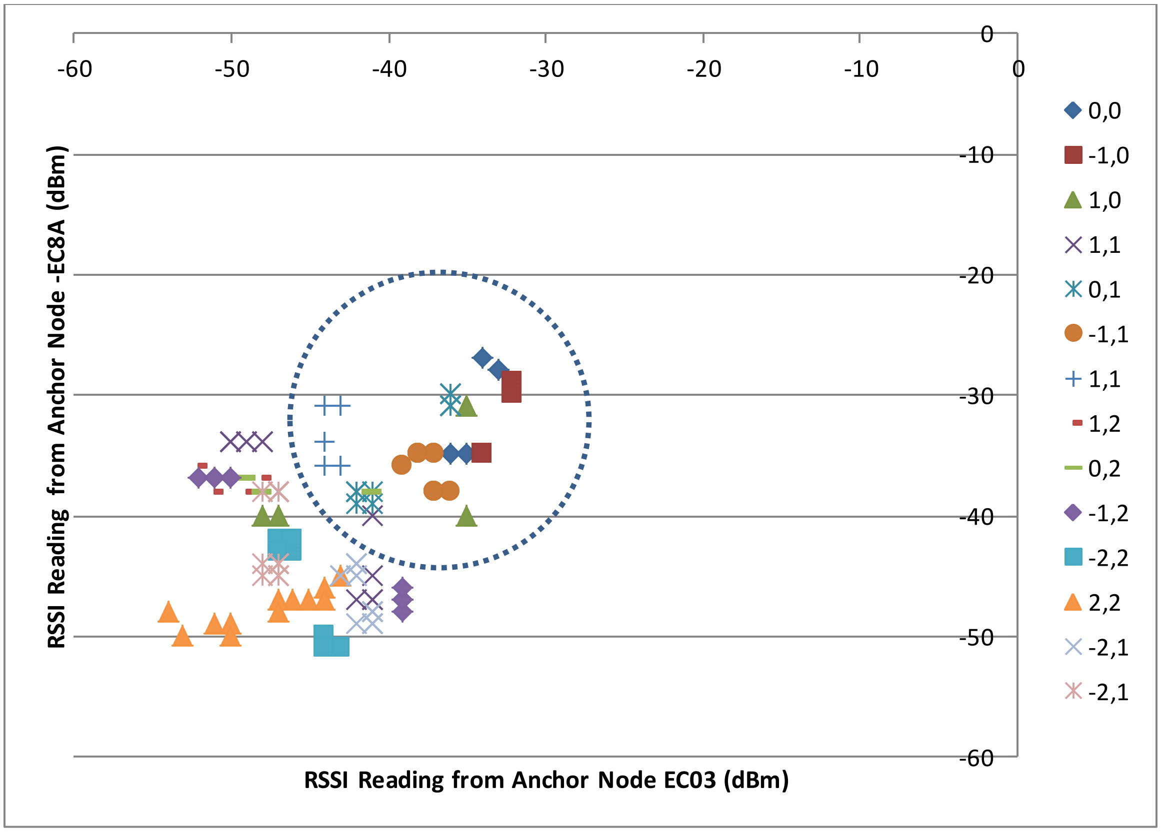


Figure 27 Location Sensor RSSI Value vs. Wearable Tag Location (RSSI Signature)

Figure 27 shows the scatter plot of the RSSI signature from the two anchor nodes. Fourteen locations with the coordinates are shown in the plot of a total of 154 sets of RSSI readings. The mobile node was set at the testing location in random directions. As marked by the dot line circle, 9 out of 91 sets of RSSI readings were within the subject detectable area (marked in figure 27 inside the circle). This means the location sensor can detect weather or not a mobile tag inside an approximate 1.2*0.7 meters area with about 90% estimate accuracy.

2. Hand-Hygiene Related Factor Evaluation

Figures 28 and 29 showed the comparison of manual and automatically collected data of hand washing time and soaping time. Where Soaping Time Error and Hand Washing Time Error were calculated by subtracting computer measured time from manually measure time.

The computer measured soaping time and hand washing times were 19.3 and 19.7 seconds respectively. While the recommended time for these were 20 and 10 seconds. As shown in the two charts, the computer measured time closely followed the manually measured time for soaping and hand washing time. Table 7 shows the descriptive statistics of the soaping and hand washing time error. The mean error for computer estimated times for soaping and hand washing were -0.91 and 3.4 seconds. So the hand washing time and soaping time estimation error were calculated as:

Hand washing time error: $$E_w = \frac{|Error\ Average|}{Hand\ washing\ time\ Average} = \frac{0.91\ s}{19.3\ s} = 4.7\% \qquad [14]$$

Soaping time error: $$E_s = \frac{|Error\ Average|}{Soaping\ time\ Average} = \frac{3.4\ s}{19.7 s} = 17.3\% \qquad [15]$$

There were three peaks for both plots caused by the hand location tracking failure. The reason for the hand tracking failure was due to lost tracking of hands location because of the hand was out of the range of the image slice algorithm thresholds.

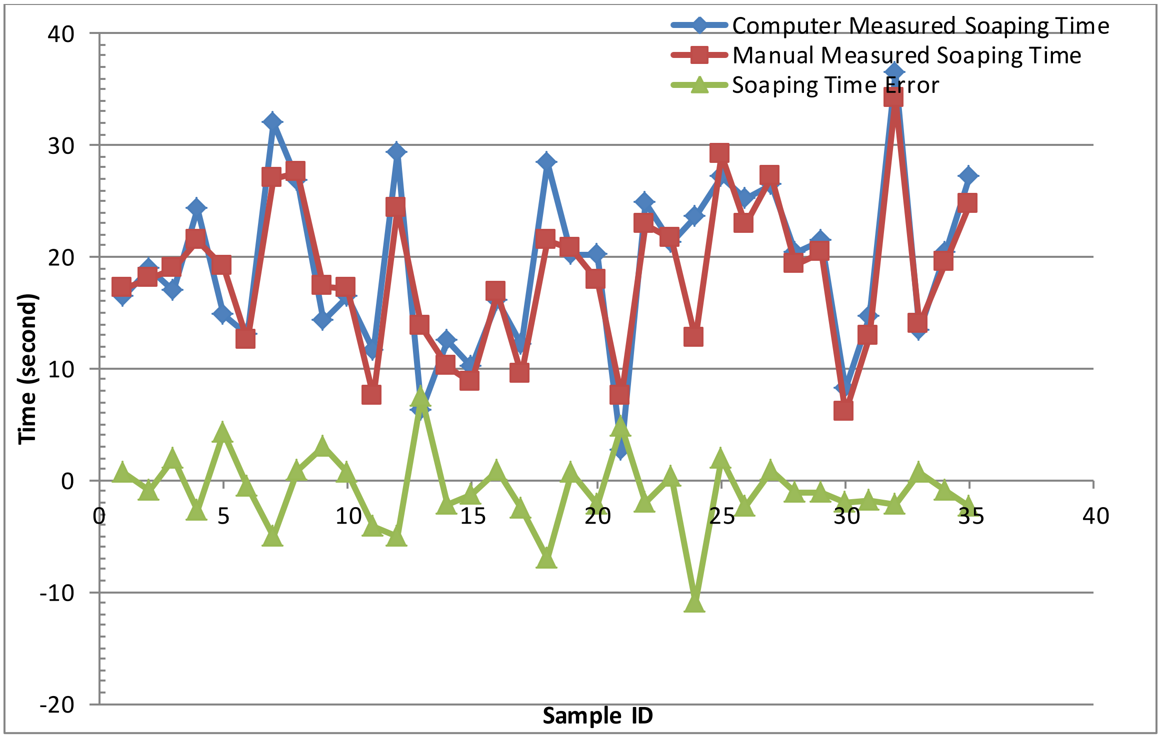


Figure 28 Comparison between Manual Measured and Computer Soaping Time

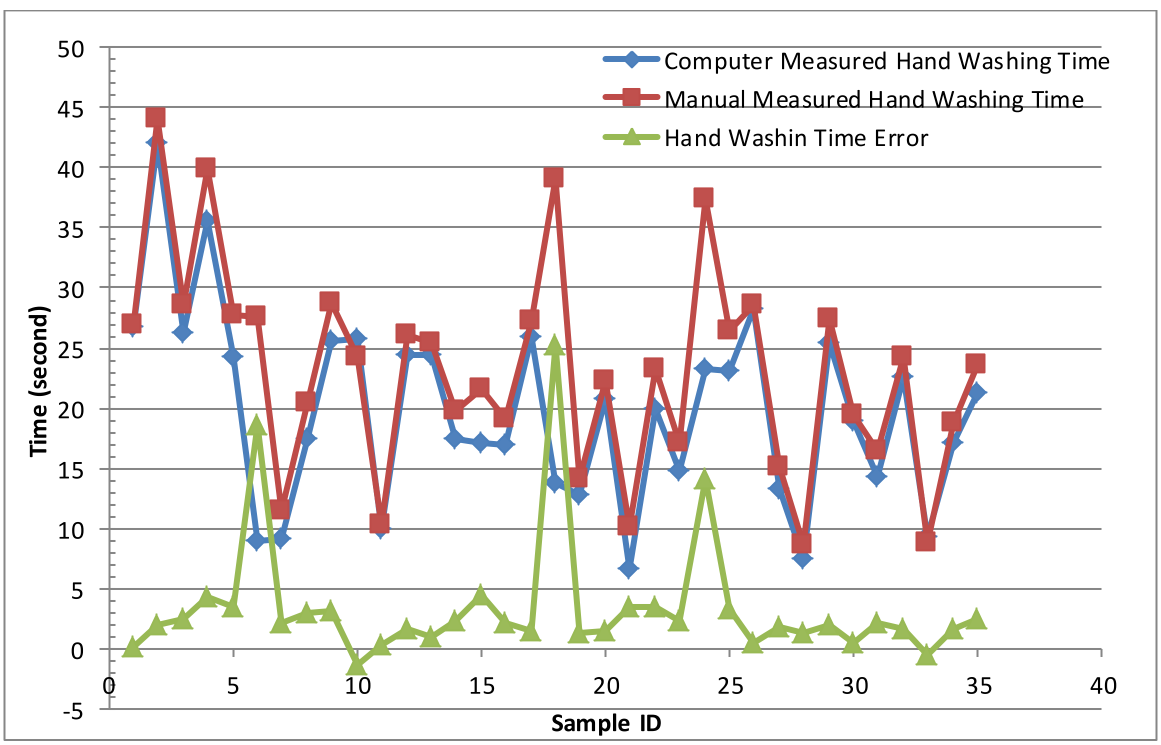


Figure 29 Comparison between Manual Measured and Computer Hand Washing Time

Table 7 Descriptive Statistics of Soaping and Hand Washing Timer Error

	Soaping Time Error	*Hand Washing Time Error*
Mean (s)	-0.91	3.40
Standard Error	0.57	0.89
Standard Deviation	3.36	5.29
Sample Variance	11.29	27.94
Range (s)	18.51	26.70
Minimum (s)	-11.00	-1.40
Maximum (s)	7.51	25.30
Sum (s)	-31.94	119.14
Count	35.00	35.00
Confidence Level (95.0%)	1.15	1.82

CHAPTER VII

CONCLUSIONS

The utmost goal of this research project was to use current advanced technologies to develop a monitoring tool to improve hand-hygiene compliance in the food industry.

This research project provided a platform that implemented human hand tracking techniques based on 3D image sensor to recognize people's behavior to realize an automatic tracking and managing system for hand-hygiene compliance evaluation. An optimized algorithm was developed which could process a large amount data from the 3D camera's video and depth output date stream. By extracting useful information from a raw depth image data with a person's hands location tracking and collecting different testing data for a hand washing station, such a system could have great potential to improve hygiene compliance in food industry.

The empirical data collected by the system for testing and evaluation showed that the physical parameters including temperature, water flow and hand-hygiene product usage monitoring could be effectively and accurately collected by a system server just like the other monitoring systems introduced in chapter II. Wireless Wi-Fi modules were successfully embedded inside the hand-hygiene product dispensers to monitor the usage. The wireless tag also had an embedded Wi-Fi module for identification location tracking. The microcontroller of the system sensing unit provided accurate water temperature and water flow readings through RS-232 communication to system server. The 3D camera and image processing methods provided this system accurate and responsive motion and location tracking performance at a very low cost. Hand location tracking allowed the system segment different procedures or stages for a hand washing event thus make it very straight forward to use a computer to evaluate individual hand-hygiene factors for long term operation. Compared with other traditional hand-hygiene monitoring methods like direct observation survey or self-reporting, this automated hand-hygiene data collection system can provide much more detailed and accurate information for hand-hygiene evaluation.

Based on the testing results, the designed prototype system reached the system design objectives:

- In the developed hand hygiene prototype system:
  - Ultra-low power wireless Wi-Fi modules were used for personal identification, hand-hygiene produce usage monitoring, and data communications;
  - A 3D camera was used to provide color and depth video stream at a hand washing station;
  - An image processing algorithm was developed for hand location tracking by using the depth video stream from the 3D camera
- Wireless Wi-Fi modules were embedded in to hand-hygiene product automatic dispensers for the corresponding product usage monitoring and human subject location estimation and tracking;
- An automated hand-washing monitoring prototype system with a computer visual interface was developed for data collection and data management;
- The developed hands location tracking algorithm could successfully track two hands location with a refreshing rate at 30 frames per second. By using the result of the hand location tracking algorithm, the developed system could segment a hand washing activity into three sections for hand-hygiene related information extracting. Hand washing time

and soaping time could be estimated with more than 83% estimate accuracy based on 35 hand washing events collected data

- Two data log files was generated and managed by the program running on the system server. One was for communication data storage and the other one was the extracted the parameters related hand-hygiene activities. Every data in the data log file was time stamped and attached with an ID number which could be used to track the source of the data;
- The wearable mobile tag could be identified within a 1.2m*0.7m detectable area in front of the hand washing sink;
- The accuracy of water temperature measurements was 99% in the range of 25°C to 55°C;
- The accuracy of the water flow measurements was 98% in the flow rate range of 1.6L/m to 4.6L/m;
- The Wi-Fi module for hand-hygiene produce usage monitoring was able to monitoring the paper/soap usages with an accuracy of over 99.4%. No time delay was achieved for 89.6% events. The average time delay for the delayed messages was 2.4 seconds.

In summary, a prototype hand-hygiene monitoring system was developed for hand wash monitoring which was able to meet the qualifiers listed in chapter one. The system was able to monitor a hand washing event with an image processing algorithm for hand location tracking and an embedded wireless data acquisition system for hand-hygiene product usage monitoring. A data log file was recorded and managed by the system server and could be used for hand-hygiene compliance evaluation for long term operations.

CHAPTER VIII

FUTURE WORKS

The proposed system design and the prototype system was the very first built system that has a lot of opportunity to be improved in future works.

A Passive Infrared Sensor (PIR) could be added into the system to save energy. it can wake up the microcontroller and server from the sleep mode whenever a human appeared within the hand washing area. Additionally, the modified paper towel and soap dispenser could also provide warning signals on content level and battery level signal that can be detected by the embedded Wi-Fi module. So the system server can generate warning signals to refill the corresponding hand-hygiene product and batteries.

The image pre-processing method in chapter 3.4 could be further optimized. The current pre-processing method uses two constant depth thresholds to remove the depth pixel data out of the thresholds range. Under this condition, the hand tracking algorithm was limited with the image data that it can only track a person's hands when they were above the hand washing sink counter. A better solution is to use a background depth reference frame to replace the two thresholds to pre-processes the depth image (background subtraction) for noise removal. Currently the hands location information was only used for hand washing time and soaping time estimation. More complicated image processing methods might be able to combine both color and depth image for hand gesture recognition to extract more detailed factors like soaping aggressiveness. This could have the potential to increase the hand-hygiene evaluation reliability. The system can be a standalone hand washing station for small scale implementations like in a small restaurant. It can also be expanded to large scale operation without too many modifications and provide room level tag location tracking and context based behavior monitoring and location tracking information. An infrastructure based Wi-Fi localization system needs to be built to evaluate the wearable mobile tag localization performance and hand-hygiene opportunity estimation. Thus, a more complicated estimation algorithm can be applied for better hand-hygiene compliance evaluation.

REFERENCES

Allegranzi, B., and D. Pittet. 2009. Role of hand hygiene in healthcare-associated infection prevention. *Journal of Hospital Infection* 73(4):305-315.

Bean, N., J. Goulding, C. Lao, and F. Angulo. 1996. Surveillance for foodborne-disease outbreaks--United States, 1988-1992. *MMWR. CDC surveillance summaries: Morbidity and mortality weekly report. CDC surveillance summaries/Centers for Disease Control* 45(5):1.

Boscart, V., K. McGilton, A. Levchenko, G. Hufton, P. Holliday, and G. Fernie. 2008. Acceptability of a wearable hand hygiene device with monitoring capabilities. *Journal of Hospital Infection* 70(3):216-222.

Boyce, J. M. 2008. Hand hygiene compliance monitoring: current perspectives from the USA. *Journal of Hospital Infection* 70:2-7.

Boyce, J. M., and D. Pittet. 2002. Guideline for hand hygiene in health-care settings: Recommendations of the healthcare infection control practices advisory committee and the hicpac/shea/apic/idsa hand hygiene task force. *Infection Control and Hospital Epidemiology* 23(S12):3-40.

Bryan, F. L. 1988. Risks of practices, procedures and processes that lead to outbreaks of foodborne diseases. *Journal of Food Protection* 51(8):663.

Buzby, J. C., and T. Roberts. 1997. Economic costs and trade impacts of microbial foodborne illness. *World health statistics quarterly. Rapport trimestriel de statistiques sanitaires mondiales* 50(1-2):57.

Challenge, F. G. P. S. 2009. WHO Guidelines on Hand Hygiene in Health Care.

Changingminds.org. 2013. Social Distance. Available at: http://changingminds.org/techniques/body/social_distance.htm.

Cherry, K., and About.com. 2013. Understanding Body Language. Available at: http://psychology.about.com/od/nonverbalcommunication/ss/understanding-body-language_8.htm.

Clayman, D. 2010. E3 2010: Project Natal is "Kinect". Available at: http://www.ign.com/articles/2010/06/13/e3-2010-project-natal-is-kinect.

Clayton, D. A., C. J. Griffith, P. Price, and A. C. Peters. 2002. Food handlers' beliefs and self-reported practices. *International Journal of Environmental Health Research* 12(1):25-39.

Edmond, M., A. Goodell, W. Zuelzer, K. Sanogo, K. Elam, and G. Bearman. 2010. Successful use of alcohol sensor technology to monitor and report hand hygiene compliance. *Journal of Hospital Infection* 76(4):364-365.

Egan, M., M. Raats, S. Grubb, A. Eves, M. Lumbers, M. Dean, and M. Adams. 2007. A review of food safety and food hygiene training studies in the commercial sector. *Food Control* 18(10):1180-1190.

Evans, H., P. Madden, C. Douglas, G. Adak, S. O'Brien, T. Djuretic, P. Wall, and R. Stanwell-Smith. 1998. General outbreaks of infectious intestinal disease in England and Wales, 1995 and 1996. *Communicable Disease and Public Health* 1:165-175.

Green, L. R., V. Radke, R. Mason, L. Bushnell, D. W. Reimann, J. C. Mack, M. D. Motsinger, T. Stigger, and C. A. Selman. 2007. Factors related to food worker hand hygiene practices. *Journal of Food Protection&# 174;* 70(3):661-666.

Guzewich, J., and M. Ross. 1999. Evaluation of risks related to microbiological contamination of ready-to-eat food by food preparation workers and the effectiveness of interventions to minimize those risks. *FDA White Paper, FDA, CFSAN*.

Howes, M., S. McEwen, M. Griffiths, and L. Harris. 1996. Food handler certification by home study: Measuring changes in knowledge and behavior. *Dairy, food and environmental sanitation: a publication of the International Association of Milk, Food and Environmental Sanitarians* 16.

Hu, C. H., and S. L. Wo. 2010. An efficient method of human behavior recognition in smart environments. In *Computer Application and System Modeling (ICCASM), 2010 International Conference on*. IEEE.

Jumaa, P. 2005. Hand hygiene: simple and complex. *International Journal of Infectious Diseases* 9(1):3-14.

Käferstein, F., Y. Motarjemi, and D. W. Bettcher. 1997. Foodborne disease control: a transnational challenge. *Emerging Infectious Diseases* 3(4):503.

Kim, Y. M., C. Theobalt, J. Diebel, J. Kosecka, B. Miscusik, and S. Thrun. 2009. Multi-view image and tof sensor fusion for dense 3d reconstruction. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*. IEEE.

Kolb, A., E. Barth, R. Koch, and R. Larsen. 2009. Time-of-flight sensors in computer graphics. In *Proc. Eurographics (State-of-the-Art Report)*.

LeBaron, C. W., N. Furutan, J. Lew, J. Allen, V. Gouvea, C. Moe, and S. Monroe. 1990. Viral agents of gastroenteritis. Public health importance and outbreak management. *MMWR. Recommendations and reports: Morbidity and mortality weekly report. Recommendations and reports/Centers for Disease Control* 39(RR-5):1.

Lynch, M., J. Painter, R. Woodruff, and C. Braden. 2006. *Surveillance for Foodborne: Disease Outbreaks: United States, 1998-2002*. US Department of Health and Human Services.

Malik, A. 2009. *RTLS for Dummies*. For Dummies.

Microsoft. 2012. Microsoft Xbox 360 + Kinect. Available at: http://www.xbox.com/en-US/kinect.

Mizuno, H., H. Nagai, K. Sasaki, H. Hosaka, C. Sugimoto, K. Khalil, and S. Tatsuta. 2007. Wearable sensor system for human behavior recognition (first report: Basic architecture and behavior prediction method). In *Solid-State Sensors, Actuators and Microsystems Conference, 2007. TRANSDUCERS 2007. International*. IEEE.

Oikonomidis, I., N. Kyriazis, and A. Argyros. 2011. Efficient model-based 3d tracking of hand articulations using kinect. *Procs. of BMVC, Dundee, UK (August 29–September 10 2011)[547]*.

Pittet, D., B. Allegranzi, and J. Boyce. 2009. The World Health Organization guidelines on hand hygiene in health care and their consensus recommendations. *World Health* 30(7):611-622.

Rahimi, M., R. Baer, O. I. Iroezi, J. C. Garcia, J. Warrior, D. Estrin, and M. Srivastava. 2005. Cyclops: in situ image sensing and interpretation in wireless sensor networks. In *Proceedings of the 3rd international conference on Embedded networked sensor systems*. ACM.

Ren, Z., J. Yuan, and Z. Zhang. 2011. Robust hand gesture recognition based on finger-earth mover's distance with a commodity depth camera. ACM.

Sahud, A. G., N. Bhanot, A. Radhakrishnan, R. Bajwa, H. Manyam, and J. C. Post. 2010. An electronic hand hygiene surveillance device: a pilot study exploring surrogate markers for hand hygiene compliance. *Infection Control and Hospital Epidemiology* 31(6):634-639.

Shim, H., R. Adelsberger, J. D. Kim, S. M. Rhee, T. Rhee, J. Y. Sim, M. Gross, and C. Kim. 2011. Time-of-flight sensor and color camera calibration for multi-view acquisition. *The Visual Computer*:1-13.

Sugimoto, C., M. Tsuji, G. Lopez, H. Hosaka, K. Sasaki, T. Hirota, and S. Tatsuta. 2006. Development of a behavior recognition system using wireless wearable information devices. In *Wireless Pervasive Computing, 2006 1st International Symposium on*. IEEE.

Takahashi, D. 2012. Xbox 360 surpasses 66M sold and Kinect passes 18M units. Venturebeat.com. Available at: http://venturebeat.com/2012/01/09/xbox-360-surpassed-66m-sold-and-kinect-has-sold-18m-units/.

Teh, C.-H., and R. T. Chin. 1989. On the detection of dominant points on digital curves. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 11(8):859-872.

APPENDIX

Appendix A: Wi-Fi Module Comparison

	WizFi210	**Carambola 2**	**WL11-IP**	**CC3000**	**RN174**
Protocols	UDP, TCP/IP (IPv4), DHCP, ARP, DNS, HTTP/HTTPS Client		UDP, TCP/IP (IPv4), DHCP, ARP, DNS, HTTP/HTTPS Client and Server		DHCP, DNS, ARP, ICMP, FTP client, HTTP client, TCP, UDP
Supply voltage	3.3 V		3.3 VDC	2.7 to 4.8 VDC	3.3V to 16V DC (depending on mode)
Output power	8 dBm	21 dB	8dBm	18dBm	0dBm to +12dBm
Power consumption	Standby: 34.0 µA Rx:125mA Tx:135mA	3.3 V, power consumption 0.5 W	Standby: 5 µA Deep Sleep:110 µA Rx: 164 mA Tx: 192 mA	Standby:<5 µA Rx: 92mA	4µA sleep, Active: 35mA RX: 180mA

Data rate		150MBps	921.6 kbps over UART	7000k bps	464Kbps over UART, up to 1Mbps over SPI slave
Wi-Fi security	WEP, WPA, WPA2		WEP, WPA/WPA2-PSK,		WEP, WPA, WPA2
Interface	I2C, SPI, UART	USB host/slave, serial port, 2 x Ethernet, i2S, SLIC, SPDIF, 23 x GPIO	2-UART, 2-SPI, I2C, PWM, 2-ADC, up to 22 GPIO		UART, SPI Slave
Antenna	U.FL	U.FL connector	PCB and U.FL connector	Not Integrated	PCB trace, wire, chip, U.FL connector
Size	32 mm x 23.5 mm x 3 mm	28 x 38 mm	36.8mm x 22.9mm	16.3mm x 13.5mm x 2mm	51 mm x 28 mm x 11.6 mm

Appendix B: Data Communication of the Wearable Wi-Fi Tag and Microcontroller

Operation	Direction	Example Message
Wake up Wi-Fi module	MCU to Wi-Fi Module	“Wake\r\n”*
	Wi-Fi Module to MCU	WiFly Ver 2.32. 05-13-2013 on 131G87 ……
Enter Command Mode	MCU to Wi-Fi Module	“$$$”
	Wi-Fi Module to MCU	“CMD\r\n”
Access Points Scan	MCU to Wi-Fi Module	“scan\r\n”
	Wi-Fi Module to MCU with two-way communication	<2.36>Scan Succeed\r\n SCAN:Found 8\r\n 8 Access Point Were Found\r\n 01,06,-56,08,3104,28,00,58:35:d9:9b:24:30,OSUSTUDENT\r\n 02,06,-43,00,2100,00,00,00:06:66:71:44:af,gateway-af\r\n 03,06,-56,00,2104,00,00,58:35:d9:9b:24:32,OSUREGISTERED\r\n 04,06,-52,00,2100,00,00,00:06:66:13:ec:03,gateway-EC03\r\n 05,06,-56,00,2104,00,00,58:35:d9:9b:24:31,OSUGUEST\r\n 06,06,-56,08,3104,28,00,58:35:d9:9b:24:33,OSUSTAFF\r\n 07,06,-

		51,00,2100,00,00,00:06:66:13:ec:8a,gateway -EC8A\r\n 08,11,- 56,02,1104,28,00,00:0f:66:bc:4e:b7,Local\r\n END: \r\n 3 Location Sensor Found\r\n Scan End\r\n
Location Sensor Message Extraction and Forwarding	MCU to System Server	*OPEN**HELLO* ID:IDEN-0EB5\r\n 3,06,-44,00:06:66:13:ec:03,gateway-EC03, \r\n 2,06,-46,00:06:66:71:44:af,gateway-af, \r\n 1,06,-36,00:06:66:13:ec:8a,gateway-EC8A, \r\n *CLOS*
Sleep	MCU to Wi-Fi Module	"sleep\r\n"
	Wi-Fi Module to MCU	No conformation message

* This could be any character sent by the microcontroller for the Wi-Fi module to wake up from deep sleep mode.

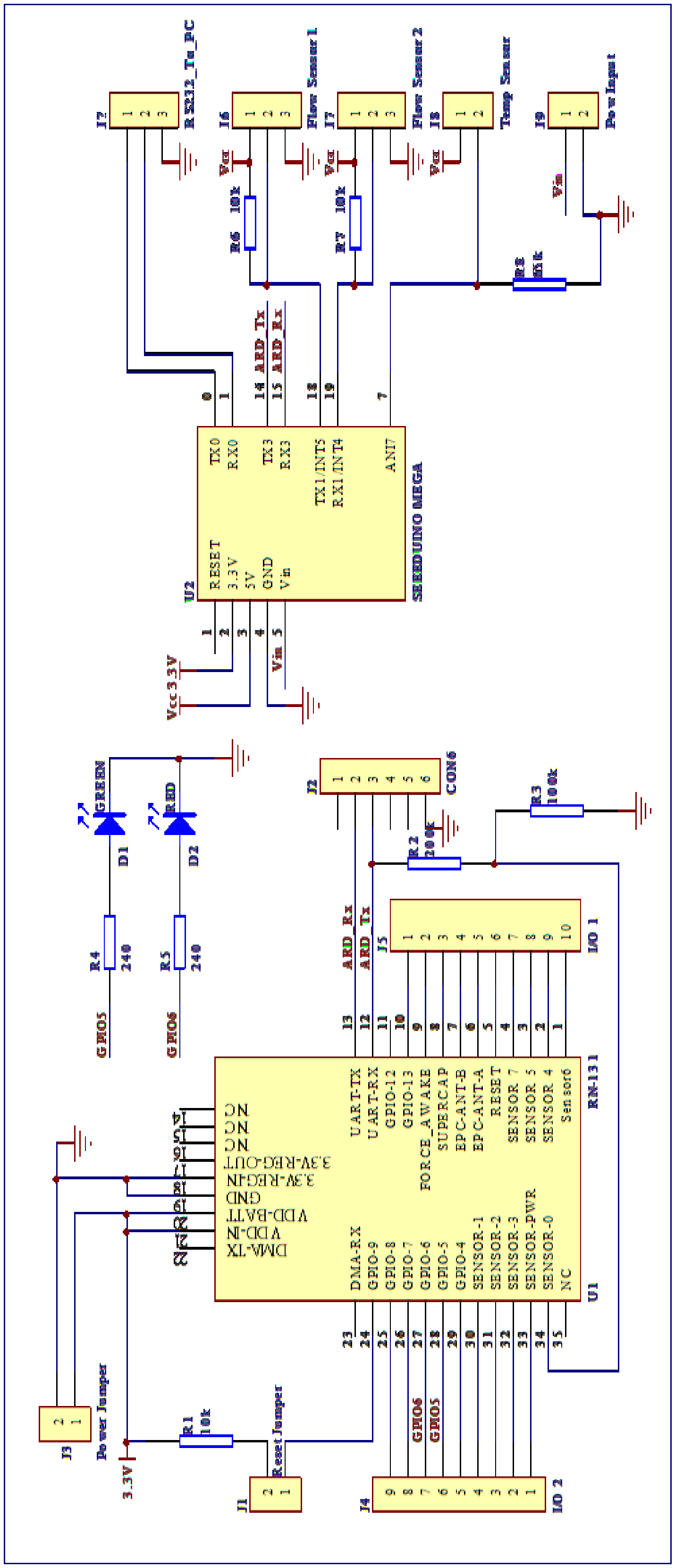
Appendix C: Comparison of Wi-Fi Module Configuration

Node Type	Parameters	Setup
Sink Node	Power	DC powered, always on
	Mode	Soft AP mode
	Association/Network*	DHCP server with SSID as gateway-XX
	Device ID*	gateway-XX
	Connection	Connect with mobile node and sensing node as a local data server/receiver via TCP/IP. Connect to the microcontroller and forwards its received TCP/IP message to the system server via RS-232.
	Data Transmission	TCP/IP message: A new message is always start with: *OPEN**HELLO*+ Space Space + ID: IDEN-XXXX \r\n Followed by multiple access points information with a line format as: **Index, channel, RSSI, MAC,SSID,\r\n** The index indicates how many access points left for this scan package. The message always end with *CLOS* No "\r\n" after *CLOS*
Anchor Node	Power	DC powered , always on

	Mode	Soft AP mode
	Association/Network	No associated network for the prototype system yet.
	Device ID*	gateway-XXXX
	Connection	Not connected to any network for the prototype system yet.
	Data Transmission	No data transmission
Mobile Node	Power	Battery operated with 3VDC (two AA batteries) Wakes up from deep sleep mode on schedule controlled by the microcontroller Go to deep sleep mode after each operation
	Mode	Client mode
	Association/Network	Sink node network / gateway-XX
	Device ID*	IDEN-XXXX
	Connection	TCP/IP with sink node. RS-232 with microcontroller on the mobile ID tag.
	Data Transmission	TCP/IP: see sink node in this table RS-232: see Table 1
Sensing Node	Power	Battery operated by the dispenser batteries (6V) or 5V DC adaptor Wakes up upon sensor pin1 set to high(1.2VDC), which was connected to the dispenser's activation signal

		Go to deep sleep mode after TCP/IP connection closed
	Mode	Client mode
	Association/Network	Sink node network / gateway-XX
	Device ID	Paper towel dispenser: PAPE-XXXX Soap dispenser: SOAP-XXXX Sanitizer dispenser: SANI-XXX
	Connection	TCP/IP to the sink node.
	Data Transmission	Wi-Fi module's Device ID

Appendix D: Schematic Diagram of Sink Node



Appendix E: Source Code of Sink Node Seeeduino Microcontroller

```
// Hand Hygiene Monitoring System
// Sink Node Microcontroller Code
// Yongbo Wan, March 2013
// BAE Oklahoma State University

#include <Streaming.h>
#define PinPaperTower 38
#define PinSoap 39
#define HotWaterSensorCoef 0.006
#define ColdWaterSensorCoef 0.006
#define WaterHot 4    //pin19
#define WaterCold 5   //pin18
#define TempAverageLevel 32
#define TempSensorCoef 0.37;

unsigned int WaterHotUse = 0;
unsigned int WaterColdUse = 0;
unsigned int WaterHotLast = 0;
unsigned int WaterColdLast = 0;

unsigned int ColdFlow  = 0;
unsigned int HotFlow  = 0;
boolean WaterOnOff = false;

// Calculated water usage result: liter
float WaterUsage = 0;
float HotWaterUsage = 0;
float ColdWaterUsage = 0;

boolean MessageReceived = false;
unsigned long CurrentTime  = 0;
unsigned long WaterOnOffCheckTimer = 0;
String ReceivedMessage;
// Water Temperature Sensor output connected to Analog input pin 0;
int WaterTempSensorPin = 7;
int TempReading[TempAverageLevel];

unsigned int i = 0;  //Iteration counter

void setup()
{
 Serial.begin(9600);     //To computer
 Serial3.begin(9600);    //To Wi-Fi module
 Serial << "Hand Hygiene Mornitoring  System MCU" << endl;
 attachInterrupt(WaterHot,WaterHotProcess,FALLING );
 attachInterrupt(WaterCold,WaterColdProcess,FALLING );
}
```

```
///////////////////////////////////////////////////////////////////////
// Main loop for Flow Sensor, Water Temperature Sensor,Faucet State
// Monitoring
///////////////////////////////////////////////////////////////////////
void loop()
{
 CurrentTime = millis();
  i++;

 if (Serial3.available() > 0) WiFi2Arduino();
 if (Serial.available() > 0) Arduion2PC();

 if (WaterOnOff && (CurrentTime - WaterOnOffCheckTimer > 2000))
 {
  Serial << "WAON of" << "\r\n" << "WAFL " << WaterHotUse+WaterColdUse << "\r\n";
  WaterOnOff = false;
  WaterHotUse = 0;
  WaterColdUse = 0;
 }

 if(!WaterOnOff && (CurrentTime - WaterOnOffCheckTimer <= 2000) )
 {
  Serial << "WAON on"<< "\r\n";
  WaterOnOff = true;
 }

 if (CurrentTime % 2000 == 1000 && WaterOnOff)
 {
  Serial << "WATP " <<GetWaterTemp()<< "\r\n";
  delayMicroseconds(50);
 }

 if (HotFlow > 0)
 {
  //Serial << "H";
  WaterHotUse += HotFlow;
  HotFlow = 0;
  WaterOnOffCheckTimer = CurrentTime;
 }
 if (ColdFlow > 0)
 {
  WaterColdUse += ColdFlow;
  ColdFlow = 0;
  WaterOnOffCheckTimer = CurrentTime;
 }

 if (abs(CurrentTime - WaterOnOffCheckTimer ) > 1500)
 {
   HotFlow = false;
   ColdFlow = false;
 }
```

```
 TempReading[i%TempAverageLevel] = analogRead(WaterTempSensorPin);
}


void WaterHotProcess()
{
 HotFlow += 1;
}

void WaterColdProcess()
{
 ColdFlow  += 1;
}

void WiFi2Arduino()
{
 for (int i = 0; i < Serial3.available();  i++)
 {
   Serial << char(Serial3.read());
}

void Arduion2PC()
{
 for (int i = Serial.available();  i > 0; i--)
 {
   Serial<<char(Serial.read());
 }
}

float GetWaterTemp()
{
 float sum = 0.0;
 float Temperature = 0;
 for(int j = 0; j < TempAverageLevel; j++)
 {
  sum += TempReading[j];
 }
 Temperature = sum/TempAverageLevel * TempSensorCoef;
 Temperature = 0.0035*Temperature*Temperature - 1.2397*Temperature + 173.03;
 return Temperature;
}
```

Appendix F: Source Code of Mobile Node Arduion Microcontroller

```
// This code runs on Arduino Mini Pro ATmega328 Board
//Com3 connected to the wifly module

// Yongbo Wan, BAE, Oklahoma State Univeristy, April 15, 2013

// State Machine/*
1:Entercommand Mode
2:Scan
3:Wait Scan Result
4:Open Tcp/Ip
 Sleep
*/

#include <Streaming.h>
#include "AP_Info.h"

unsigned long CurrentTime = 0;
unsigned long LastTime = 0;

#define Conn_TCP 2     // connected to Wifly Module GPIO5
#define Status_TCP 3   // connected to Wifly Module GPIO6

//Found Location Sensor flag
boolean flag_Find_LS = false;
boolean GotDeviceID = false;

// State Machine Var
uint8_t State = 0;

// Wifly CMD mode Status
boolean Wifly_CMD = false;

//Received message buffer
String MessageW2A = "";
String Command;
String DeviceID = "";

boolean WiflyReplyReceived = false;

// The number of founed Access Point
int AP_Quantity;

// The number of founded Locaiton sensor
uint8_t LocationSensorQuant = 0;

// AP list index
uint8_t ListIdx = 0;
```

```
//Array to store Scaned AP information
ScannedAPInfo AP_List[5];

void setup()
{
 //Serial.begin(9600);
 Serial.begin(9600);
 pinMode(Status_TCP, INPUT);
 pinMode(Conn_TCP, OUTPUT);
 digitalWrite(Conn_TCP, LOW);
 CurrentTime = millis();
 LastTime = CurrentTime;
 WiflyEnterComMod();
 GetDeviceID();
 MessageW2A = "";
 delay(3000); //
}

void loop()
{
 //Arduino as bridge between PC and Wifly module
 // UART 3 on Arduino mega is the port connected to Wifly module
 if (Serial.available() > 0) Wifly2PC();
 //if (Serial.available() > 0) PC2Wifly();

 switch (State)
 {
  // Enter Command Mode
  case 1:
  {
   if (Wifly_CMD)
   {
    State = 2;
   }
   else
   {
    Serial << "close" << endl;
    delay(2000);
    MessageW2A = "";
    WiflyEnterComMod();
   }
   if (!GotDeviceID) GetDeviceID();
   delay(10);
   break;
  }
  // Scan
  case 2:
  {
   if(Wifly_CMD)
   {
```

```
      if (WiflyScan())//   SendWiflyCMD("scan",  "<4.00>")
      State = 3;
      //Serial << "State2: " << State << endl;
     }
     else
     {
      WiflyEnterComMod();
      State = 1;
     }
     delay(1);
     break;
    }
      case 3:
    {
     if (WiflyReplyReceived)    WiflyMessageProce();
     if (flag_Find_LS)
     {
      State = 4;
      //Serial << "State: " << State << endl;
      flag_Find_LS  = false;
     }
     delay(1);
     break;
    }
    // Exit  Command Mode
    case 4:
    {
     digitalWrite(Conn_TCP,  HIGH);
     delay(50);
     while(!digitalRead(Status_TCP))
     {
      if (millis()  - CurrentTime  > 4000)
      {
        WiflyEnterComMod();
        Serial << "reboot" << endl;
        Wifly_CMD = false;
      }
     }
     delay(10);
     Serial << "  ID:" << DeviceID << endl;
      for(uint8_t i = 0; i < LocationSensorQuant;  i++)
      {
      Serial << LocationSensorQuant-i << ',' << AP_List[i].Ch << ',' << AP_List[i].rssi << ',' <<
AP_List[i].MAC  << ','<< AP_List[i].ssid << ','<< endl; //
      }
      if (LocationSensorQuant == 0)
      {
       Serial << "No Location Sensor Found" << endl;
      }
      delay(100);
      digitalWrite(Conn_TCP,  LOW);
```

```
      delay(200);
      if (WiflyEnterComMod())
      {
        Serial << "sleep" << endl;
      }
      State = 0;
      Wifly_CMD = false;
      LastTime = CurrentTime;
      //Serial << "State4: " << State << endl;
      break;
    }
    default:
    {
        if (CurrentTime % 10000 == 0)
      {
       State = 1;
       //Serial << "Scan End" << endl;
       delay(2);
      }
    }
  }

  CurrentTime = millis();
  if (LastTime > CurrentTime) LastTime = CurrentTime;
  else if (CurrentTime - LastTime > 30000)
  {
    State = 1;
    WiflyEnterComMod();
    Serial << "reboot" << endl;
    delay(1000);
    LastTime = CurrentTime;
  }
}

void WiflyMessageProce()
{
  int NewLineIdx = MessageW2A.indexOf('\n');

    if (NewLineIdx > 0)
    {
     Command = MessageW2A.substring(0, NewLineIdx);
     MessageW2A = MessageW2A.substring(NewLineIdx + 1);
     int idx = Command.indexOf("Found");
     if (idx >= 0)
     {
       String quantity = Command.substring(idx+ 6, Command.length());
       AP_Quantity = quantity.toInt();
       LocationSensorQuant = 0;

     }
       // Extract AP information
```

```
      //scan
      //5 Access Point Were Found
      //          1         2         3         4
      //01234567890123456789012345678901234567890123456789
      //01,01,-62,08,3104,28,00,58:35:d9:9b:24:33,OSUSTAFF
      //02,01,-62,08,3104,28,00,58:35:d9:9b:24:30,OSUSTUDENT

      else if (AP_Quantity > 0)
      {
        int temp1 = Command.indexOf("gateway");
        //Serial << "List " << Command << " idx:" << temp1 << endl;

        if (temp1 >= 0)
        {
          AP_List[ListIdx].Idx = Command.substring(0,2);
          AP_List[ListIdx].Ch = Command.substring(3,5);
          AP_List[ListIdx].rssi = Command.substring(6,9);
          //AP_List[ListIdx].SecruityMode = Command.substring(10,11);
          //AP_List[ListIdx].cap = Command.substring(13,16);
          //AP_List[ListIdx].WPAconfig = Command.substring(18,19);
          //AP_List[ListIdx].WPSMode = Command.substring(21,22);
          AP_List[ListIdx].MAC = Command.substring(24,41);
          AP_List[ListIdx].ssid = Command.substring(42,Command.length()-1);
          ListIdx += 1;
        }
        AP_Quantity -= 1;
      }
    else if (Command.indexOf("END") >= 0)
    {
        // Set Wifly Scan has Result Flag
        flag_Find_LS = true;
        LocationSensorQuant = ListIdx;
        ListIdx = 0;
        //Serial << LocationSensorQuant << " Location Sensor Found" << endl;
//          for(uint8_t i = 0; i < LocationSensorQuant; i++)
//          {
//            Serial << i+1 << ":" << AP_List[i].Idx << ",Channel:" << AP_List[i].Ch << ", RSSI:"
<< AP_List[i].rssi << ", MAC:" << AP_List[i].MAC << ", SSID:"<< AP_List[i].ssid << endl;
//          }
//          Serial << "Reset AP_List_Index" << endl;
    }
  }
  else
  {
    WiflyReplyReceived = false;
  }
}

//Scan for location sensor
boolean WiflyScan()
{
```

```
 MessageW2A = "";
 if (Serial.available() >0) ClearUart3Buf();
 boolean scan = false;
 unsigned long TempTime = millis();
 Serial << "scan" << endl;
 delay(20);
 while(MessageW2A.length() < 6)
 {
  if(Serial.available() > 0)
  {
   Wifly2PC();
  }
  if (millis() - TempTime > 10000)
  {
   State = 0;
  }
 }
 if (MessageW2A.indexOf("2.36") >= 0)
 {
  scan = true;
  //Serial << "Scan Succeed" << endl;
 }
 delay(1);
 return scan;
}

boolean WiflyEnterComMod()
{
 MessageW2A = "";
 if (Serial.available() > 0) ClearUart3Buf();
 boolean cmd = false;
 unsigned long TempTime = millis();
 Serial << "$$$";
 while (MessageW2A.length() < 3)
 {
  if(Serial.available() > 0)
  {
   Wifly2PC();
   //Serial << MessageW2A << '.';
  }
  if (millis() - TempTime > 1000)
  {
   Serial << "exit" << endl;
   State = 1;
   //Serial << "Enter CMD Time Out" << endl;
   break;
  }
 }

 if (MessageW2A.indexOf("CMD") >= 0 || MessageW2A.indexOf('$$$' >= 0))
 {
```

```
    cmd = true;
    Wifly_CMD = true;
    //Serial << "In Command Mode" << endl;
  }
  return cmd;
}

boolean WiflyQuit_CMD()
{
  boolean quit = false;
  unsigned long TempTime = millis();
  MessageW2A = "";
  if (Serial.available() > 0) ClearUart3Buf();
  Serial << "exit" << endl;
  while (MessageW2A.length() < 4)
  {
    if(Serial.available() > 0)
    Wifly2PC();

    if (millis() - TempTime > 10000)
    {
      State = 0;
      //Serial << "Enter CMD Time Out" << endl;
    }
  }

  if (MessageW2A.indexOf('EXIT') >= 0)
  {
    quit = true;
    Wifly_CMD = false;
    //Serial << "Exit from Command Mode" << endl;
  }
  return quit;
}

void Wifly2PC()
{
  int tempch = Serial.read();
  MessageW2A += char(tempch);
  //Serial << char(tempch);
  if (tempch == 13 || tempch == 10) //|| tempch == 10
  {
    WiflyReplyReceived = true;
  }
  //Serial << "Wifi2Arduino" << endl;
}

void PC2Wifly()
{
  //Serial3 << char(Serial.read());
}
```

```
void GetDeviceID()
{
 MessageW2A = "";
 if (Serial.available() > 0) ClearUart3Buf();
 Serial << "show d" << endl;
 while (MessageW2A.length() < 18)
 {
  if(Serial.available() > 0)
  Wifly2PC();
 }
 //int temp = MessageW2A.indexOf('=');
 DeviceID = MessageW2A.substring(9);
 if (DeviceID.indexOf("IDEN") == 0 && DeviceID.length() == 9)
 {
  GotDeviceID = true;
  MessageW2A = "";
  if (Serial.available() > 0) ClearUart3Buf();
 }
 //Serial << MessageW2A << endl;
}

void ClearUart3Buf()
{
 int bufSize = Serial.available();
 if (bufSize > 0)
 {
  for (int i = 0; i < bufSize; i++)
  {
   char temp = Serial.read();
  }
  //Serial << bufSize << " bytes cleared" << endl;
 }
}
```

Appendix G: IRB Approval

Oklahoma State University Institutional Review Board

Date:	Wednesday, May 29, 2013
IRB Application No	EG132
Proposal Title:	Tracking and Gesture Recognition Approach for Monitoring Hand Hygiene Compliance for Food Handling and Processing Industry
Reviewed and Processed as:	Exempt

Status Recommended by Reviewer(s): Approved Protocol Expires: 5/28/2014

Principal Investigator(s):

Yongbo Wan	Ning Wang
111 Ag Hall	111 Ag Hall
Stillwater, OK 74078	Stillwater, OK 74078

The IRB application referenced above has been approved. It is the judgment of the reviewers that the rights and welfare of individuals who may be asked to participate in this study will be respected, and that the research will be conducted in a manner consistent with the IRB requirements as outlined in section 45 CFR 46.

☑ The final versions of any printed recruitment, consent and assent documents bearing the IRB approval stamp are attached to this letter. These are the versions that must be used during the study.

As Principal Investigator, it is your responsibility to do the following:

1. Conduct this study exactly as it has been approved. Any modifications to the research protocol must be submitted with the appropriate signatures for IRB approval. Protocol modifications requiring approval may include changes to the title, PI, advisor, funding status or sponsor, subject population composition or size, recruitment, inclusion/exclusion criteria, research site, research procedures and consent/assent process or forms.
2. Submit a request for continuation if the study extends beyond the approval period of one calendar year. This continuation must receive IRB review and approval before the research can continue.
3. Report any adverse events to the IRB Chair promptly. Adverse events are those which are unanticipated and impact the subjects during the course of this research; and
4. Notify the IRB office in writing when your research project is complete.

Please note that approved protocols are subject to monitoring by the IRB and that the IRB office has the authority to inspect research records associated with this protocol at any time. If you have questions about the IRB procedures or need any assistance from the Board, please contact Dawnett Watkins 219 Cordell North (phone: 405-744-5700, dawnett.watkins@okstate.edu).

Sincerely,

Shelia M. Kennison

Shelia Kennison, Chair
Institutional Review Board

VITA

Yongbo Wan

Candidate for the Degree of

Doctor of Philosophy

Thesis: TRACKING AND HANDS MOTION DETECTION APPROACH FOR MONITORING HAND-HYGIENE COMPLIANCE FOR FOOD HANDLING AND PROCESSING INDUSTRY

Major Field: Biosystems Engineering

Biographical:

Education:

Completed the requirements for the Doctor of Philosophy in Biosystems Engineering at Oklahoma State University, Stillwater, Oklahoma in July, 2013.

Completed the requirements for the Master of Science in Electrical Engineering at Shaanxi University of Science and Technology, Xi'an, Shaanxi, China in 2006.

Completed the requirements for the Bachelor of Science in Electrical Engineering at Shaanxi University of Science and Technology, Xi'an, Shaanxi, China in 2001.