Xi'an Jiaotong-Liverpool University
西交利物浦大学
DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING

EEE311 Final Year Project

Communication Technology for Multiple Automated Guided Vehicles (AGVs)

In Partial Fulfillment
of the Requirements for the Degree
Bachelor of Engineering

Student Name : Xiangxing Li
Student ID : 1510108
Supervisor : Limin Yu
Assessor : Mark Leach

Abstract

Automated Guided Vehicle (AGV), the portable robots which can drive automatically as instruction, show great potential in the future market. The main objective of this project is to investigate and compare dierent communication mechanisms on the AGV system. In this project, we use Arduino Mega2560 for MCU to build an Omni-wheel car. Considering the comparison between each communication method, we choose to apply WIFI, Zigbee, and Bluetooth and investigate its characteristics based on transmission distance, signal eectiveness, and system stability. The module ESP8266 (wi), Xbee S2C (ZigBee) and CC2541 (Bluetooth) are tested for investigation and experiment.

To measure the maximum transmission distance, a LED prototype is designed, which assists to judge whether the signal is received. To study signal eectiveness and system stability, Received signal strength indication (RSSI) is imported in the comparative experiment and serves as the key indication for measurement. In the experiment, three environmental eects are considered and set for controlled variables: indoor and outdoor eect, electromagnetic shielding eect (EMS), and obstacle attenuation eect. Eventually, the results showed that the Xbee S2C module had the best performance on transmission distance, signal eectiveness and system stability and CC2541 performed worst. Conclude the experiment, since module type limits, the antenna of each module is dierent, and the number of types of parameters studied (only RSSI), the comparison results may not theoretically represent the characteristic of each technology. However, the procedure and method of practical experiments can reference for future study..

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation and Objectives

The Internet of Things (IoT), which is one of the most trending technology, has been in the contemporary era and has a strong capability to reach to a significant share of the global market in the future[1]. Automated Guided Vehicle (AGV), the portable robots which can drive automatically as instruction, is one of the most achievable products of IoT and has been wildly applied in industrial manufacturing facility, warehouse or transportation systems.

In order to realize autonomous AGV driving, the control and navigation problem are the key issues for AGV [2], while a stable and effective communication system is the foundation of the system. The objective of this project is to build a prototype of the AGV system based on different communication mechanisms such as wifi, ZigBee and Bluetooth, and then investigate and compare the characteristics of each technology. In this project, we especially focused on transmission distance, signal effectiveness and stability of the communication system.

1.2 Industry Relevance

Currently, the wildest applicant scenarios of AGV are on logistics. Whatever Taobao or Jingdo, has built many intelligent warehouses for logistic division and transmission. On 10th, December 2017, the World's biggest Automated Cargo Wharf Yangshan Deep-water Port was on operation in Shanghai. There are 130 driverless automatic guided vehicles (AGV) equipped in the port, where the AGV system takes the most important role. Nowadays, stable, effective and efficient wireless communication is the powerful technologies for the rapidly growing manufacturing industry [3].

1.3 Literature Review

1.3.1 Wireless Communication on AGV

Centralized Control

For centralized control, at the beginning of AGV development, AGV move along fixed paths, and realize localization by using Charge Coupled Device (CCD) cameras or Radio Frequency Identification (RFID) [4], meanwhile, all path information collected from each AGV by are stored in one host computer. Then, with the concept "networking" importing in the AGV system as well as computing capability increasing, the application of AGVs in more complex and flexible scenarios increases. Lately, Controller Area Network (CAN) is settled down by using standardized interface cards and Ultra High Frequency (UHF) transceiver modules [4]. Until now, for chasing low cost, less power consuming and more comprehensive performances, the technology WIFI, and ZigBee have been the first choice for controlling method.

Distributed Control

Previously, the control system is based on centralized control, however, when the industrial needs booms, the AGVs system is becoming larger and more complex, the demand for computing capacity dynamically increases. The localization of each AGVs becomes more efficient. In addition, due to the rapid development of hardware and sensors, distributed control is becoming a more accessible and practical solution. In the distributed control system, Cyber-Physical System (CPS) based control approach is a method [4].

WIFI

WIFI, the Wireless Fidelity, is the communication technology based on standard IEEE802.11, which belongs to the physical layer of TCP/IP network layers shown in Fig1.1. In this project, Transmission Control Protocol (TCP), one kind of full-duplex communication, can achieve centralized controlling by utilizing WIFI.

ZigBee

ZigBee is the standard IEEE802.15.4 for wireless sensor networks (WSN). It has been applied to the home automation system, industry control, environment control and so on. It shows the advantage of its low cost but meanwhile provides a low transmission data rate. Same as other electromagnetic waves, it is easily affected by the wave which has the same frequency band of 2.4GHz, such as WIFI, Bluetooth. Meanwhile, even in the environment full of crowds, the signal strength will fluctuate a lot according to research [5].

The ad-hoc characteristic of ZigBee allows the user to design star, tree, and mesh topology network [6]. Three kinds of components, which are coordinator, router and the end device, form the network shown in the figure below.

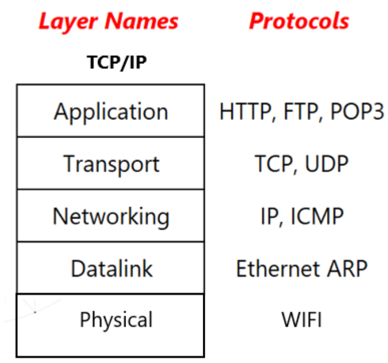

Figure 1.1: TCP/IP Model

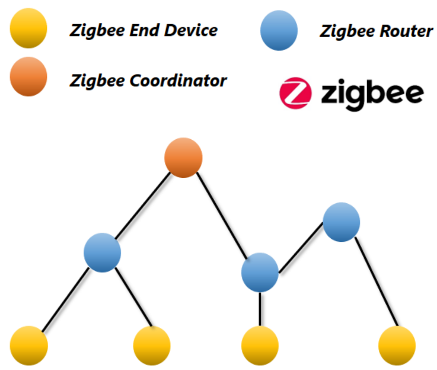

Figure 1.2: ZigBee Network

1.3.2 RSSI

Log-distance Path Loss Model

Received Signal Strength Indication (RSSI) aims to judge the quality of the link, can be defined as

$$RSSI(dBm) = 10 * log10(\frac{P(mW)}{1(mW)}), \tag{1.1}$$

where P means power. Since for wireless signal, the output power is generally less than 1mW, thus the data measured on the hardware always shows negatively. Meanwhile, RSSI can represent the process of signal strength attenuation. There are two models applied on distance measurement, which are space propagation model and log-distance path loss model [7]. Since the previous one sets the working environment in free space, not attenuation or multi-path effect is considered, the log-distance path loss model, which can be applied in both indoor and outdoor environments, is more useful in practical scenarios. The mechanism of the model is shown below.

$$PathLoss_{d_2} = PathLoss_{d_1} + 10 * \gamma * \frac{d_2}{d_1} \tag{1.2}$$

Where d_2 is the distance of position from the transmitter to the receiver. d_1 is the distance at the referenced position. While P_d represents the pass loss at each position and γ means the path loss component [8]. In free space, $\gamma = 2$, while in reality, γ should be larger than 2. If we consider the referenced point which is 1 meter away from the transmitter, we can deduce the RSSI (dBm) is

$$RSSI(dBm) = RSSI_1(dBm) + 10 * \gamma * log10(D), \tag{1.3}$$

where $RSSI_1$ is the RSSI value at the referenced point, and D is the distance between transmitter and receiver.

Ingredients to effect RSSI

There are kinds of ingredients affecting the RSSI. Considering environmental effects, the interference can be caused by neighbor co-frequency signals, background noise, and electromagnetic shield, an obstacle will result in attenuation of the signal. The study in [9] compares the ZigBee Network under Wi-Fi Interference in different situations based on arrival time and RSSI. The results show that the interference of ZigBee and Wi-Fi networks is a serious problem [9]. Expanding the results, we can also deduce that Wifi, ZigBee is actually mutual interference sources.

Considering the inside ingredients of the system, the type of antenna has a great influence on signal strength. Experiment [10] compares the RSSI of four models equipped with a dipole, ceramic, meander PCB and inverted meandered antenna. The result shows that the ESP8266 module based on the dipole antenna has the strongest RSSI, the following is the ceramic antenna, meander PCB antenna and inverted meandered antenna.

Meanwhile, RSSI will fluctuate by crowd behavior in indoor environments [5]. The signal strength will be affected by the density, velocity or disorder of crowds [5].Where d_2 is the distance of position from transmitter to receiver. d_1 is the distance at referenced position. While P_d represents the pass loss at each position, and γ means the path loss component [8]. In free space, $\gamma = 2$, while in reality γ should be larger than 2. If we consider the referenced point which is 1 meter away from transmitter, we can deduce the RSSI (dBm) is

$$RSSI(dBm) = RSSI_1(dBm) + 10 * \gamma * log10(D), \tag{1.4}$$

where $RSSI_1$ is the RSSI value at referenced point, and D is the distance between transmitter and receiver.

Ingredients to effect RSSI

There are kinds of ingredients effecting the RSSI. Considering environmental effects, the interference can be caused by neighbor co-frequency signals, background noise, and electromagnetic shield, and obstacle will result in attenuation of signal. The study in [9] compares the ZigBee Network under Wi-Fi Interference in different situations based on arrival time and RSSI. The results show that the interference of ZigBee and Wi-Fi networks is a serious problem [9]. Expanding the results, we can also deduce that Wifi, ZigBee are actually mutual interference sources.

Considering the inside ingredients of system, the type of antenna has a great influence on signal strength. Experiment [10] compares the RSSI of four models equipped with dipole, ceramic, meander PCB and inverted meandered antenna. The result shows that the ESP8266 module based on dipole antenna has the strongest RSSI, the following is ceramic antenna, meander PCB antenna and inverted meandered antenna.

Meanwhile, RSSI will fluctuate by crowd behavior in indoor environment [5]. The signal strength will be effected by the density, velocity or disorder of crowds [5].

Chapter 2

Methodology and Results

2.1 Methodology

2.1.1 Structure of Car

The main components of the car are shown in table below.

Device	Chasis	Motor	L298N	Mega board	ESP8266	CC2541	Xbee
Number	2	3	2	2	1	1	1

Table 2.1: State of DC motor

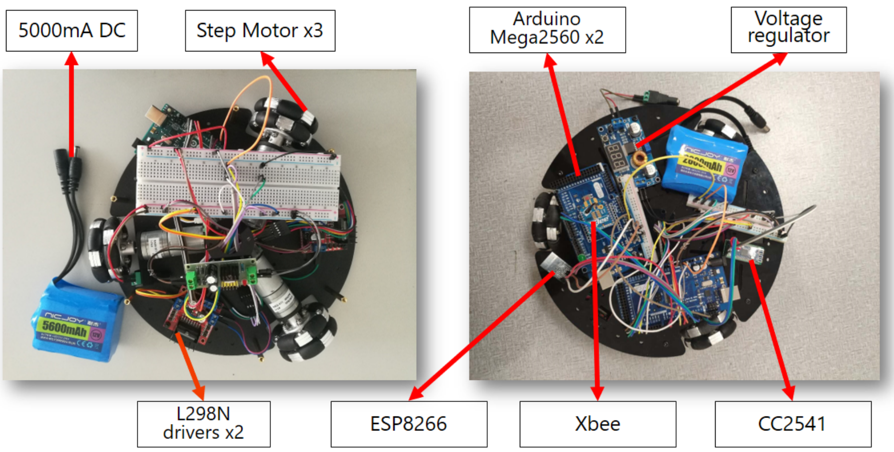

Figure 2.1: Inside and outside structure of the car

Arduino Mega 2560

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560, serving as the main board in this project.

L298N Driver

L298N is the motor driver, which can control the rotation direction by *Enable* and speed of DC motor by *Pulse Width Modulator* (PWN) control. To control the rotation direction, different states can be shown as a table below. Meanwhile, the drive controls the rotation speed by inputting PWN pulse to ENABLE.

ENA	INPUT 1	INTPUT 2	Motor State
0	X	X	STOP
1	0	0	RUN
1	0	1	FWD
1	1	0	REV
1	1	1	RUN

Table 2.2: State of DC motor

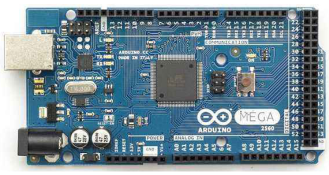

Figure 2.2: Arduino Mega 2560

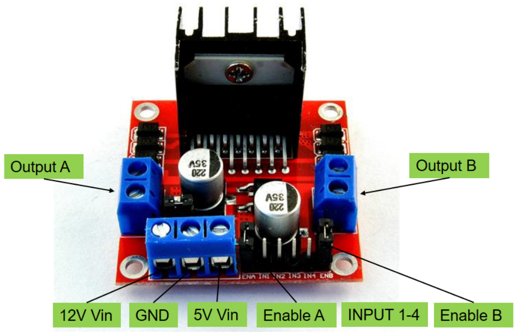

Figure 2.3: M298N Motor Driver

Omni-Wheels

The omnidirectional wheel can roll forward and back as the formal wheel, and realize lateral movement by moving the side rollers embedded in the wheel. Thus, the combination of Omni-wheels allows the car to move straightly without changing the posture of the car. Due to its characteristics, the Omni-wheels can be applied on AGVs operating in specific scenarios.

Figure 2.4: Front side of Omni wheel

Figure 2.5: Side view of Omni wheel

2.1.2 Motion Modality

Consider the motion of Omni-wheel based car, the modality can be diverse and depended on the referenced coordinate. Based on a referenced coordinate, when each wheel rotates toward a specific direction with a specific speed, as figure 2.6, the car can move towards four directions.

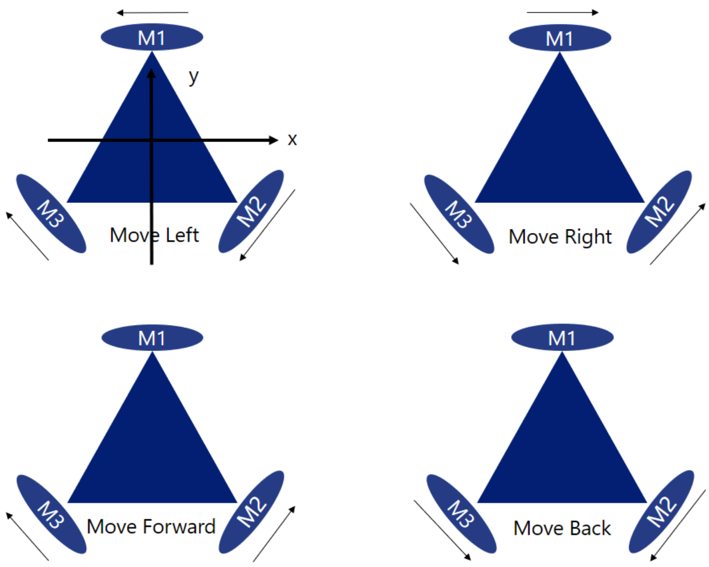

Figure 2.6: Motion analysis

Analyze the stress and speed of wheels, the idealized mathematical model requires:

1 Each wheel can produce fully friction,

2 The center of motor axis is the center of gravity of chassis,

3 Three wheels are handed over to the same centre.

Analyze the normal condition as shown in figure 2.7. There are two coordinates serving for calculation. The red one represents the coordinate system of the site, while blue one represents the coordinate system of car motion. V_1, V_2 V_3 represents line velocity of the wheels, L is the distance from the car center to the wheel center. To find the relation between car velocity (V_x, V_y, V_z) based on site coordinate system and line velocity of the wheels, we should break down the car velocity along to both coordinate systems of site and car motion. Supposed the decomposed velocity on the coordinate system of car motion is $V_{x'}$, $V_{y'}$, $V_{z'}$, the angular velocity is ω.

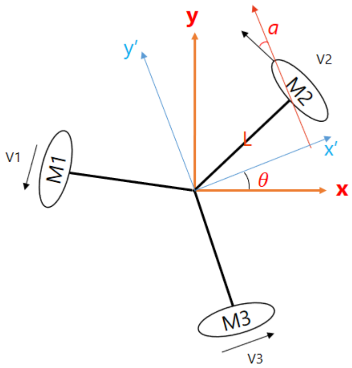

Figure 2.7: Before coordinate rotation

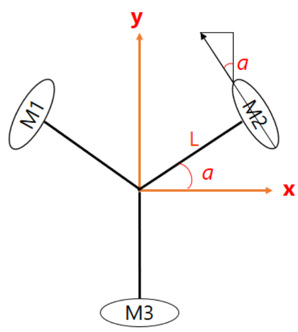

Figure 2.8: After coordinate rotation

Basically, represent the velocity on car motion coordinate system by using velocity on site.

$$\begin{bmatrix} V_{x'} \\ V_{y'} \\ \omega \end{bmatrix} = \begin{bmatrix} sin(\theta) & cos(\theta) & 0 \\ -sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} V_x \\ V_y \\ \omega \end{bmatrix}. \tag{2.1}$$

Then, based on figure 28, considering to break down the line velocity to site coordinate system, we can deduce:

$$\begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} = \begin{bmatrix} -sin(a) & -cos(a+) & L \\ -sin(a) & cos(a) & L \\ cos(a) & sin(a) & L \end{bmatrix} * \begin{bmatrix} V_{x'} \\ V_{y'} \\ \omega \end{bmatrix}. \tag{2.2}$$

Combining previous two matrixes, eventually we can deduce the relationship between line velocity and car velocity as:

$$\begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} = \begin{bmatrix} -sin(a) & -cos(a+) & L \\ -sin(a) & cos(a) & L \\ cos(a) & sin(a) & L \end{bmatrix} * \begin{bmatrix} sin(\theta) & cos(\theta) & 0 \\ -sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} V_x \\ V_y \\ \omega \end{bmatrix}. \quad (2.3)$$

Simplifying the results, we have

$$\begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} = \begin{bmatrix} -sin(a+\theta) & -cos(a+\theta) & L \\ -sin(a+\theta) & cos(a+\theta) & L \\ cos(\theta) & sin(\theta) & L \end{bmatrix} * \begin{bmatrix} V_x \\ V_y \\ \omega \end{bmatrix}. \quad (2.4)$$

Thus, to realize forward, back, moving left and right, the responding $a = 30$, $\theta = 0$ and $\omega = 0$, then we can get the results:

$$V_1 = -0.5 * V_x - \frac{\sqrt{3}}{2} * V_y, \quad (2.5)$$

$$V_2 = -0.5 * V_x + \frac{\sqrt{3}}{2} * V_y, \quad (2.6)$$

$$V_3 = V_x. \quad (2.7)$$

2.1.3 Communication Methods

To realize centralized control, ESP8266-01 module (wifi), Xbee module (ZigBee) and CC2541 module (Bluetooth) are applied to provide communication solutions between central controller and car.

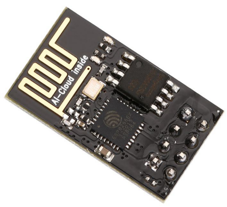

Figure 2.9: ESP8266

Figure 2.10: Xbee

Figure 2.11: CC2541

Foundational Module Parameters

To explore the character in further experiments, we need to have a general sense based on module futures. The parameter shown table below were collected from the product manual of each module.

Parameter\ Module	ESP8266	Xbee	CC2541
Transmission Distance	30 - 120m	30 - 100m	10 - 50m
Receiver Sensitivity	-93dBm	-92dBm	-94dBm
Date Rate	110 - 921600bps	250kbps	250kbps - 2Mbps

Table 2.3: Module Parameters

- ESP8266: PCB Aentenna/ I-PEC Connector, SMA Connector,
- Xbee: PCB Antenna/ U.FL Connector, RPSMA Connector,
- CC2541: PCB Antenna.

System Flow

Apply the previous three modules, the car can be controlled by PC and mobile phone. The system flow is shown below.

Consider the flow of the ESP8266 module application. Basically, programmed the module and enable it to identify the IP address of specific wifi and connect it. Then, the hotspot of the phone provided the wifi for ESP8266 and PC. While, by using TCP/IP Socket, PC can establish a server, which will be detected by ESP8266 and connect with the module. Eventually, the commands of car motion can be sent to Mega for follow-up processing.

Consider the communication built by Xbee. There were two modules, one was set to be a coordinator, and another was set to be a router. In this flow, the coordinator, which was serially connected with PC, served as the transmitter, and the router served as receiver for signal transmission. In addition, as the module is shown in figure 2.16, the "coordinator" was connected by the USB adapter, and the "router" realized the serial connection with one Mega board by using the expansion board.

As for Bluetooth, after setting it up, the module which supports BLE6.0 in the mobile phone would detect the model type and wait for a pair-up connection.

Above all, based on module communication and serial communication methods, we can control the car by using a PC or mobile phone.

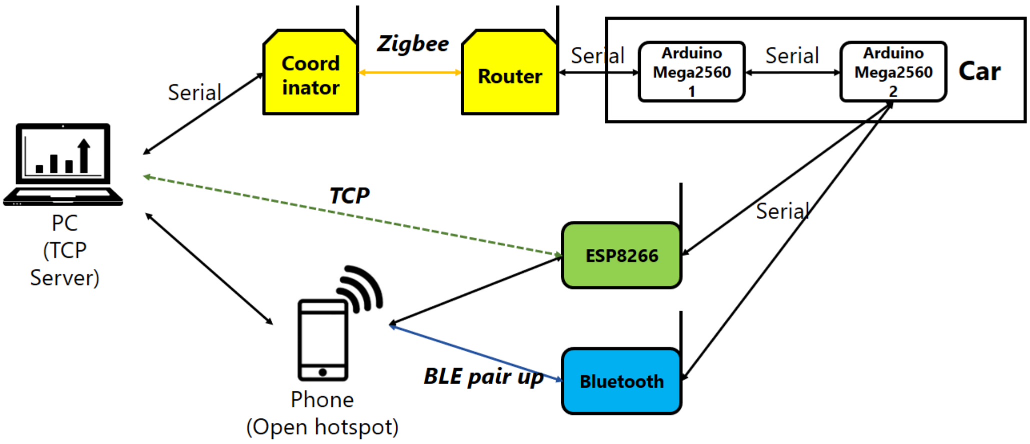

Figure 2.12: Motion analysis

ESP8266 Module

Consider the WIFI modules, ESP8266 has drawn much attention to open source community due to its high-cost performance, including 2.4 GHz transceiver with a power amplifier, CPU, ADCs, timers, GPIOs, etc [10]. It is a low-cost WIFI microchip and allows the microcontroller to access WiFi networks with simple connectivity (SPI / SDIO or I2C interface / UART) [11].

To drive the module, we can use CP2102 serial to USB adapters to connect the module with the computer, and then we can use serial assistants (fig 2.13) to send AT-Command or directly program it at Arduino IDE.

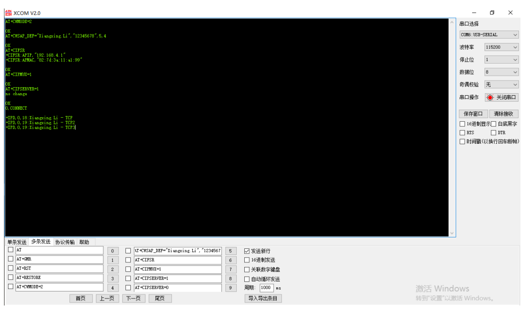

Figure 2.13: XCOM Serial Debuger

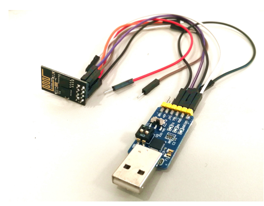

Figure 2.14: ESP8266 with CP2102

Xbee Module

Xbee S2C modules, based on ZigBee standards, can support Wireless Sensor Network (WSN) at low power and low cost. The module works with the 2.4GHz frequency band as well as ESP8266.

To drive the module, we can use the Xbee USB adapter or Arduino Xbee expansion board to connect with PC and use XCTU to set parameters. The network setting can be edited in "Radio Configuration" shown in figure 2.15 or by sending AT-command.

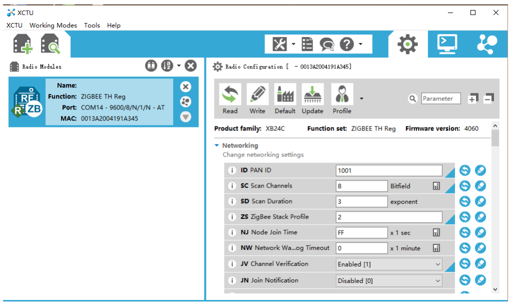

Figure 2.15: XCTU

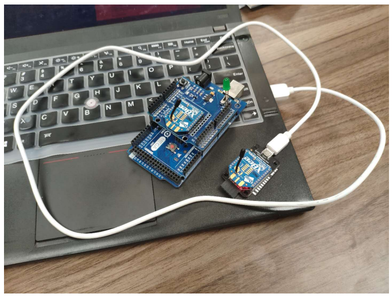

Figure 2.16: Xbee with adapter and expansion board

CC2541 Module

The CC2541 is a low-cost module, based on IEEE 802.15.1 standard, supports BLE4.0. It is suited for ultra-low power consuming systems. The module can be directly connected to the pair up the object, and we can also use CP2102 to realize the serial connection between the module with PC.

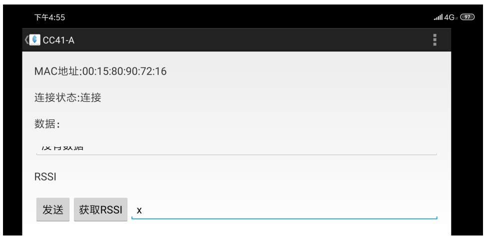

Figure 2.17: BLE Debuger

2.1.4 Comparative Experiment

The objective of the experiment is highly aligned with project aims. It is to study the transmission distance, signal effectiveness, and stability of three communication systems. The main method of this experiment is to measure the Received Signal Strength Indication (RSSI) of three technologies with different environmental effects. The following content will focus on the experiment setting and detailed procedure.

LED Prototype

The LED prototype has the same system flow as the built car, users can control the light on or off by sending the commands through three communication methods. The prototype aims to measure the maximum transmission distance of each module. The procedure is to take the transmitter and leave away the receiver step by step, at the same time, repeat sending commands to control the light. Until the light cannot be controlled, we assume that the signal is not effective, and the current distance from the transmitter to the receiver is the maximum transmission distance. Ultimately, repeat the experiment five times to get an average value.

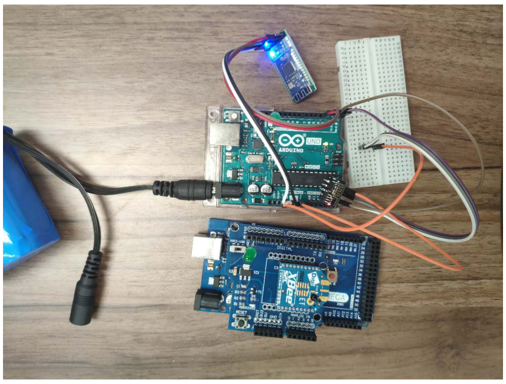

Figure 2.18: LED prototype for measurement

RSSI Measurement

Consider the software for measurement, the ESP8266 module can be programmed to return RSSI by sending command. Thus, we can use TCP/IP debugger to repeat sending the signal at a certain frequency, here we set interval time is 1 second. For the Xbee module, we can use XJTU to send AT command to return a hexadecimal value. As for CC2541, we can read the RSSI on the BLE debugger. The measuring process, showing on the user interface is shown in figure 2.19.

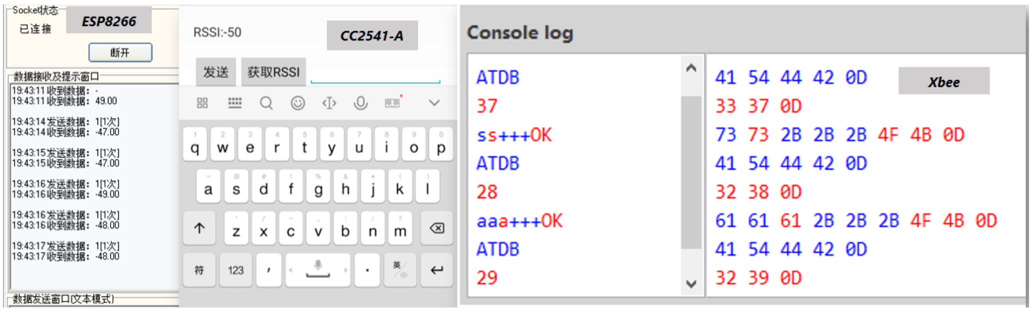

Figure 2.19: The inference of TCP/IP Socket, BLE Debuger, XCTU to get RSSI

Insight of location choice and measured points, there is a different environment for experiments, which will be explained in the following illustration. To generate measured points, take one path shown in figure 2.20 for an explanation. Based on the result of measurement of transmission distance, we set the path to be measured at 80 meters. As previously mentioned, according to the log-distance path loss model, the RSSI can be reduced to:

$$RSSI(dBm) = RSSI_1(dBm) + 10 * \gamma * log10(D). \tag{2.8}$$

Assume that the $RSSI_1 = -30$, $\gamma = 2$, distance interval between each measured point is 1 meter. Then, we can draw the mathematical model shown in figure below.

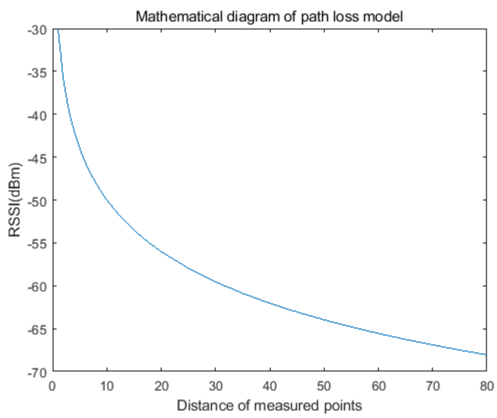

Figure 2.20: Assumed log-distance path loss model

Observe the graph, decreasing trends are slowing down when the distance from the transmitter to the receiver increases. Thus, in order to show a more accurate and completed changing trend of RSSI, we set a tighter distance interval at closer measured points. From 0 to 10 meters, set distance interval to 1 meter. For the region from 10 to 20 meters, the distance interval was set to 2.5 meters. As for the interval from 20 to 80 meters, we set an interval to 5 meters. Eventually, we selected 27 points for measurement and the expression of D in Matlab simulation is shown below.

```
x1=[0:9];
x2=[10:2.5:17.5];
x3=[20:5:80];
D=[x1,x2,x3];
```

Illustrate the procedure of measuring RSSI at each point. As figure 2.21 shows, at each point, we put the device at the right top of the measured point, controlled the light, and measured the RSSI value. Eventually, a group of data can be obtained shown as figure 2.22.

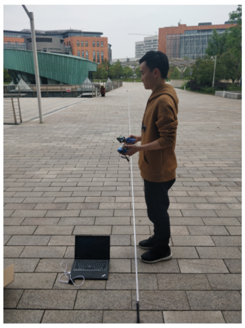

Figure 2.21: Measure process

WIFI	Distance	0	1	2	3	4	5
	RSSI	-21	-39	-50	-51	-47	-49
Distance	17.5	20	25	30	35	40	45
RSSI	-51	-53	-52	-57	-64	-66	-75

5	6	7	8	9	10	12.5	15
-49	-51	-58	-55	-58	-60	-55	-59
45	50	55	60	65	70	75	80
-75	-75	-79	-79	-79	-79	-79	-79

Figure 2.22: Part of RSSI data

Environment Setting

1. To measure maximum transmission distance, the environmental effects such as scattering, reflection, diffraction and refraction [12] should be reduced as much as possible, thus the outdoor environment should be considered. The path in front of the International Research (IR) building in the South campus of XJTLU was chosen. The white line represents the movement of each module in the experiment.

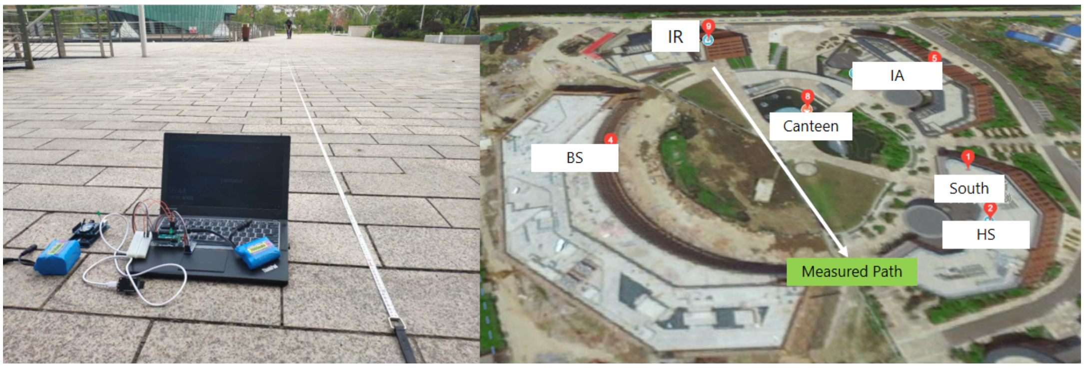

Figure 2.23: Measured path in front of "IR" building

2. To simulate electromagnetic shielding effects, we used an iron box to envelope the module as figure 2.25 shows.

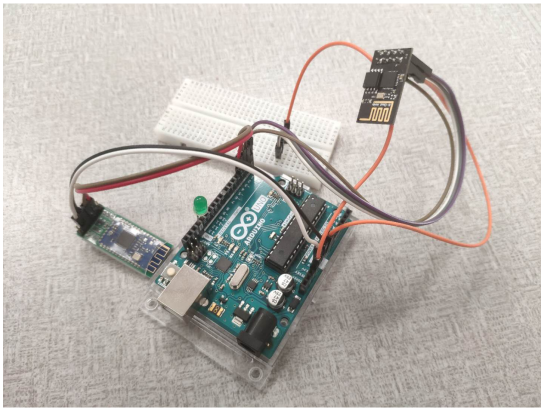

Figure 2.24: Module without EMS effect

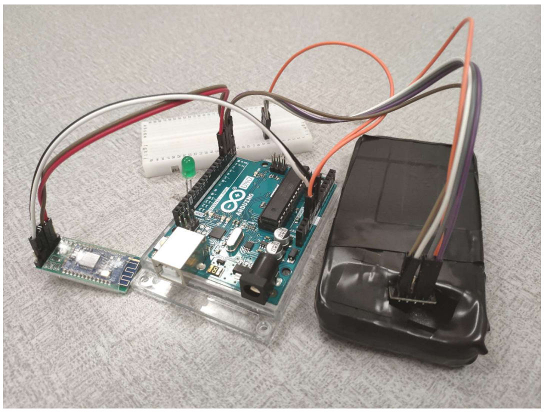

Figure 2.25: Module With EMS effect

3. To explore outdoor and indoor environmental effects, we made outdoor experiments in the path in front of the IR building shown as figure 2.23. As for the indoor environment, besides the wifi distribution, other ingredients we considered were that the environment should provide the long straight path without obstacles and fewer crowds walking around. Thus we chose to make experiments along the corridor on the 4th floor of IBSS building. Due to the limitation of path length, the measured path is from 0 to 55m.

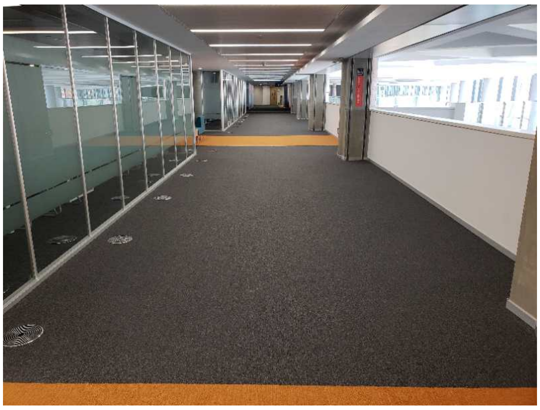

Figure 2.26: IBSS 4th floor: Horizontal version

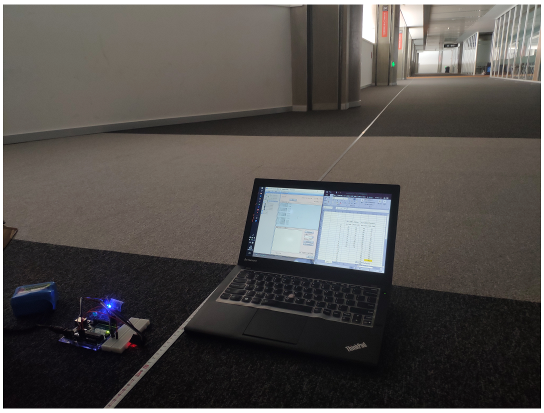

Figure 2.27: Measuring process

4. To analyze the obstacle attenuation effect, we chose IR 6th floor (figure 2.28) for experiments. The procedure of measurement is shown in the following. Basically, we put the transmitter and receiver device at the corner beside the elevator, shown as figure 2.30, and measured the value of RSSI represented by $emphR1$. Then move the receiver to another corner, and measured to get *R2*. The previous two steps aim to simulate the RSSI of two points within a fixed distance.

Then, we need to set obstacles for the comparative experiment. Basically, we moved the transmitter and receiver device to the central point of the wall and measured the RSSI. The relative position of the two devices should be the same as previous. Since the transmitter was close enough to the receiver. The RSSI measured at the center point remains to be $emphR1$. Afterward, put the receiver to the center point of another side of the wall, and made the same measurement as before, we can get $emphR3$. Due to obstacle attenuation effects, there should be $R2 > R3$.

Figure 2.28: IR 6th floor

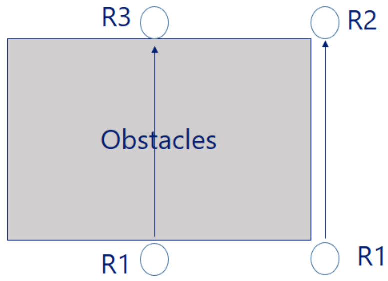

Figure 2.29: Diagram for measurement

Figure 2.30: IR 6th floor

Figure 2.31: Indoor: IBSS 4th floor

2.2 Results and Analysis

2.2.1 Car Movement

The motion of the car can be controlled by sending w, s, a, d, x, where w means forward, s means back, a means moving left, d means moving right, and x is to stop the car.

Figure 2.32: Interface to send message from TCP/IP Socket, BLE Debuger, XCTU

2.2.2 Transmission Distance

After five times of measurement, the maximum transmission distance of each module is shown below. The Xbee module owns the farthest transmission distance, and CC2541 has the shortest.

Figure 2.33: Collected RSSI data

Compare the theoretical parameter with practical results, as shown in table below, the measured data makes sense within the allowable range of error.

Parameter\ Module	ESP8266	Xbee	CC2541
Theoretical Transmission Distance	30 - 120m	30 - 100m	10 - 50m
Measured Transmission Distance	120m	80m	55m

Table 2.4: Comparison between theoretical and practical transmission distance

2.2.3 RSSI in Scenarios

Data Pretreatment

The results of the RSSI of three modules in various environments are collected and shown in figure 2.34. By using Matlab, the data can be analyzed. However, before data analysis, pretreatment needs to be considered for unifying abscissa. Since the transmission distance of CC2541 does not reach to 80 meters. Meanwhile, due to the limitation of indoor distance, the measured data in the indoor environment is based on a 55-meter long path. Thus, to unify abscissa, we used $-100dBm$ to represent ineffective signal and fill up the "BLE" table (red part in figure 2.34), and used the RSSI at the distance of 55 meters away to fill up "WIFI" and "ZigBee "table (green part in figure 2.34).

BLE

	BLE (dBm) Indoor		BLE (dBm) Outdoor	
Distance	No EMS	With EMS	No EMS	With EMS
0	-25	-55	-30	-51
1	-55	-60	-49	-62
2	-70	-72	-60	-71
3	-74	-72	-64	-73
4	-76	-78	-68	-71
5	-75	-77	-65	-80
6	-72	-75	-63	-75
7	-74	-80	-66	-83
8	-81	-82	-68	-87
9	-79	-81	-63	-85
10	-82	-78	-70	-90
12.5	-73	-81	-77	-94
15	-72	-80	-73	-93
17.5	-79	-85	-75	-92
20	-74	-91	-88	-96
25	-77	-93	-90	-99
30	-83	-100	-94	-95
35	-81	-100	-95	-100
40	-87	-100	-93	-100
45	-94	-100	-96	-100
50	-100	-100	-95	-100
55	-100	-100	-100	-100
60	-100	-100	-100	-100
65	-100	-100	-100	-100
70	-100	-100	-100	-100
75	-100	-100	-100	-100
80	-100	-100	-100	-100

WIFI

Wifi (dBm) Indoor		Wifi (dBm) Outdoor	
No EMS	With EMS	No EMS	With EMS
-21	-33	-25	-31
-39	-45	-39	-52
-50	-49	-53	-71
-51	-53	-54	-78
-47	-46	-62	-86
-49	-50	-61	-83
-51	-60	-58	-89
-58	-71	-59	-88
-55	-66	-64	-85
-58	-69	-60	-83
-60	-66	-61	-86
-55	-67	-66	-88
-59	-64	-63	-84
-51	-66	-74	-82
-53	-61	-72	-81
-52	-63	-77	-88
-57	-67	-78	-87
-64	-69	-84	-92
-66	-74	-83	-96
-75	-77	-87	-90
-75	-76	-83	-92
-79	-82	-84	-92
-79	-82	-87	-91
-79	-82	-83	-88
-79	-82	-87	-95
-79	-82	-91	-95
-79	-82	-96	-100

Zigbee

Zigbee (dBm) Indoor		Zigbee (dBm) Outdoo	
No EMS	With EMS	No EMS	With EMS
-26	-33	-34	-45
-43	-51	-48	-60
-56	-53	-52	-62
-59	-62	-51	-61
-62	-59	-56	-54
-61	-58	-62	-55
-60	-64	-62	-63
-57	-65	-61	-63
-60	-58	-64	-64
-61	-62	-63	-61
-63	-63	-61	-68
-66	-74	-60	-73
-62	-63	-72	-75
-61	-66	-82	-74
-70	-76	-78	-85
-68	-73	-81	-80
-63	-79	-82	-81
-62	-65	-75	-86
-65	-68	-85	-84
-74	-74	-80	-83
-75	-74	-76	-90
-75	-74	-83	-79
-75	-74	-88	-86
-75	-74	-89	-90
-75	-74	-87	-92
-75	-74	-84	-85
-75	-74	-79	-89

Figure 2.34: Results of maximum transmission distance of three modules

Module Comparison

Compare the RSSI between each module. To reduced environmental effects, the experiment was set in outdoor. The measured result is shown in figure 2.36. Based on Log-distance Path Loss model [13]:

$$RSSI(dBm) = RSSI_1(dBm) + 10 * \gamma * log10(D), \tag{2.9}$$

(where γ is a path-loss component, in free space, we suppose $\gamma = 2$), the theoretical RSSI along the varying distance of three modules can be obtained, which is shown as figure 2.35. These two graphs illustrate that ESP8266 and Xbee module own the appropriate performance on signal effectiveness, and CC2541 performed worst.

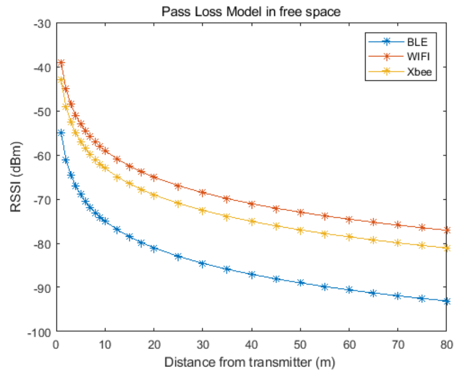

Figure 2.35: Theoretical RSSI

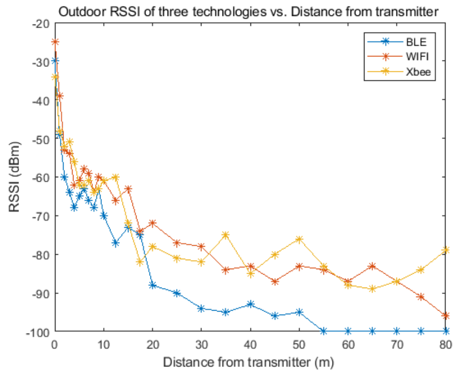

Figure 2.36: Measured RSSI

Compare the experimental and theoretical results of each module. Since in practical scenarios, γ is larger than 2, which is its value in free space. Thus the measured RSSI should be less than the theoretical value. The results graph below is satisfied with the analysis.

Meanwhile, it shows that the measurement on the interval between 0 and 10 meters is highly matched with the theoretical graph, whereas, when the distance exceeds 10 meters, the attenuation is accelerated by increasing distance.

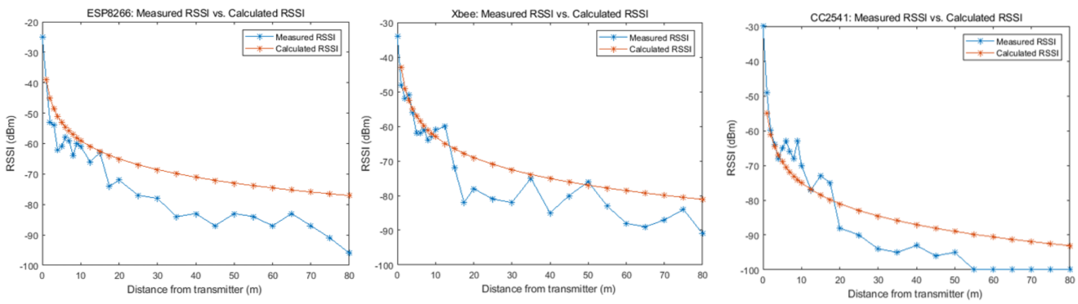

Figure 2.37: Measured and calculated RSSI of three modules

EMS Effects

Comparing the ElectroMagnetic Shielding effects (EMS) to each module, we measured the RSSI of the receiver with or without EMS effects in an outdoor environment. Three graphs are drawn below. The attenuation effects depend on the size, shape, and orientation of the shielding [14]. However, shielding cannot protect against low-frequency magnetic fields. Seen on results, it is apparent that the Xbee module owns the best performance on EMS resistance.

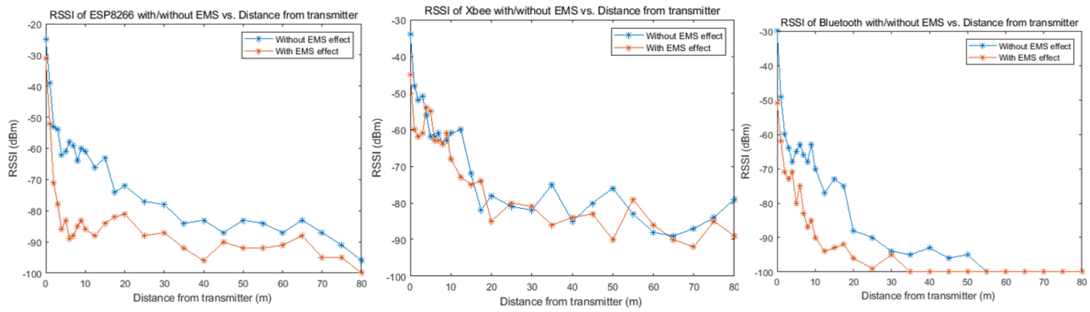

Figure 2.38: RSSI of three modules with or without EMS effect

Indoor and Outdoor Environment

Comparing the indoor and outdoor environmental effects. The main difference lies in the multi-path environment, and the interference caused by other co-frequency signals in indoor scenarios. Multi-path illustrates that the path of the electromagnetic wave is combined with the direct wave and indirect waves such as reflected wave, diffracted wave, transparent wave, and etc [5]. In the experiment, there were about 120 IP address of WIFI detected, which means interference caused by WIFI was severe. Observe the figure results, the common future is that the RSSI in the indoor environment of three modules is all almost larger than which in the outdoor environment. The reason behind the feature may lie on multi-path effects. However, observe the graph of CC2541, the maximum transmission distance in the outdoor is larger than which in indoor. Besides, review the measuring process in the experiment, several connection failures occurred in Xbee. Thus, we can deduce that in the indoor environments, due to multipath, and interference effects, the RSSI of each module increases, nevertheless, it reduces the effectiveness of signal, meanwhile reducing transmission distance.

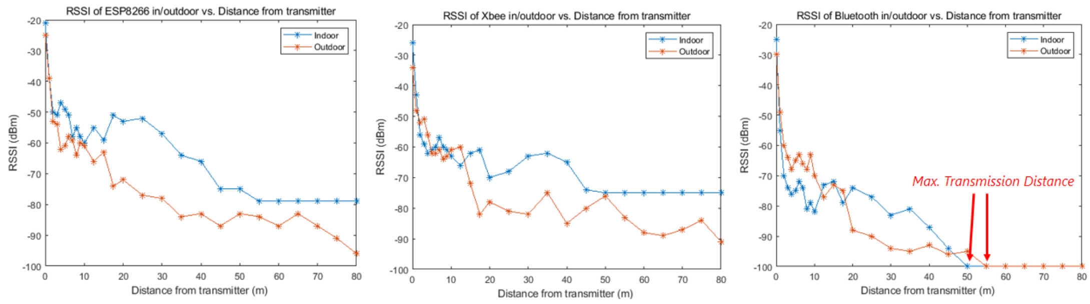

Figure 2.39: RSSI of three modules in outdoor or indoor environment

Obstacle Attenuation Effect

Analyze the obstacle attenuation effect, when an enormous obstacle exists, part of the electromagnetic wave would directly pass through the obstacle, and other parts would arrive by wave reflection and diffraction. Since three modules are all 2.4GHz band based technology, the attenuation should be approachable. The results are shown in figure 2.40. It is clear that each module ran through signal attenuation, and all decreased about 20dBm, which is aligned with the theoretical analysis.

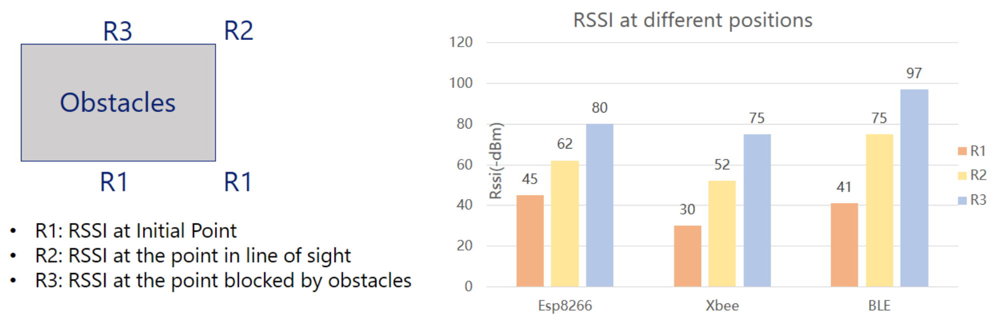

Figure 2.40: RSSI of three modules with or without obstacle attenuation

Chapter 3

Conclusion and Future Work

3.1 Experiment Review

Review the experiments, there are identified points for improvement.

1 Module type limits: There are too few types of modules for experiments. In the experiment, we used ESP8266-01 module for experiments, however, the manufacturer has designed and launched numerous versions of modules, from ESP-01 to ESP-12, etc, each of which offers different added features and configurations [10]. Thus the performance of one module cannot exactly represent the character of the technology behind.

2 Indication limits: In the experiment, we analyzed the character mainly by RSSI. However, RSSI is easily effected, and may not fully reflect the signal quality in the transmission. For wireless signal modulation and coding schemes, the user can adjust the data transfer rate. However, the Modulation scheme supporting high-speed transmission requires high SNR [10], which means the RSSI can be adjusted to support a specific data rate. Meanwhile, the packet arrives time, delay time, data transfer rate, energy efficiency are all ingredients to decide the quality of the communication system. Based on different scenarios, the requirement varies.

3 Other ingredients: The module in the experiment should be equipped with the same antenna. The recording process could be improved from manual recording to automated recording, which can measure the data from more times to get a more accurate average value.

3.2 Project Conclusion

Conclude the project. In this project, the objective of the project is investigating the characteristics of three familiar communication technologies wildly applied to the Automated Guided Vehicle (AGV) system. At the end of the project, the built Omni-wheel car, equipped with Xbee, ESP8266-01, and CC2541 modules were able to realize centralized control by PC or mobile phone. The car could following the instruction to forward, back, move left, move right, and stop. Besides, a series of comparative experiments were made to study the transmission distance, signal, and stability of the module.

The key indication in the experiment was RSSI, and there were three environmental effects set for study: Indoor and outdoor environmental effects, electromagnetic shielding effects and obstacle attenuation effects. The results are shown as the graph of RSSI vs. Transmission distance, and according to theoretical analysis, the graph drawn in the experiment is reasonable.

The results reflect that among Xbee, ESP8266 and CC2541 modules, Xbee seems to own the most comprehensive performance, and the CC2541 perform worst based on transmission distance, signal effectiveness, and system stability. Nevertheless, there are several identified points in the experiment to be improved, the performance of the communication system should be evaluated based on various ingredients and specific requirement.

3.3 Budget

The project totally spent 1623.2 RMB, the detailed device list is shown below.

Semester 1			
Product & Module	**Number**	**Brand**	**Price (RMB)**
Arduino package: Arduino UNO	1	U-MICRO	185.2
Arduino Mega2560 R3 MCU	1	BIHUISHUMA	62.5
Smart Car board x2 & Motors x3	1	JUXINZHICHUANG	437
12V 5600mA lithium batteries	1	Nicjoy	97
12V 2800mA lithium batteries	1	Nicjoy	57
L298N Motor drive	4	Telesky	46.82
L7805 5V Manostat	1	Telesky	7.55
CC2541 Bluetooth 4.0	2	Telesky	50.62
USB to UART Serial module CP2102	1	Telesky	20.33
ESP8266 Module	1	Telesky	16.01
ESP8266 Development Board	1	Telesky	100.37
Total		1085.2	
Semester 2			
Xbee(zigbee module) x2	2	DIGI	254
Xbee usb V2	1	Dfrobtot	112
BLE LINK 4.0	1	Dfrobtot	87
Leonardo Xbee	1	Dfrobtot	85
Total		538	
Total		1623.2	

Figure 3.1: Project Budget

3.4 Future Work

There are many subjects to be considered in the future.

1 Car Motion: The PID control is the essential solution to improve the stability of AGV motion.

2 Distributed Control: The technologies which can realize the communication between AGV, such as the UDP of WIFI, API mode of ZigBee, or Controller Area Network can be considered.

3 Specific Experiment: There are many ingredients influencing the communication system, dig out one issue and conduct in-depth research.

4 Practical Design: The project in the future can be aimed at designing a communication system of AGV for practical scenarios. Study one specific issue, deliver the questions and design the prototype to solve the problems.

Bibliography

[1] K. Routh and T. Pal, "A survey on technological, business and societal aspects of internet of things by q3, 2017," in *2018 3rd International Conference On Internet of Things: Smart Innovation and Usages (IoT-SIU)*, pp. 1–4, Feb 2018.

[2] M. Watanabe, M. Furukawa, and Y. Kakazu, "Acquisition of communication protocol for autonomous multi-agvs driving," in *Proceedings of the Second International Conference on Intelligent Processing and Manufacturing of Materials. IPMM'99 (Cat. No.99EX296)*, vol. 2, pp. 1115–1121 vol.2, July 1999.

[3] V. K. Kongezos and C. R. Allen, "Wireless communication between agvs (autonomous guided vehicles) and the industrial network can (controller area network)," in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, vol. 1, pp. 434–437 vol.1, May 2002.

[4] M. Zhang and K. Yu, "Wireless communication technologies in automated guided vehicles: Survey and analysis," in *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*, pp. 4155–4161, Oct 2018.

[5] M. Arai, H. Kawamura, and K. Suzuki, "Estimation of zigbee's rssi fluctuated by crowd behavior in indoor space," in *Proceedings of SICE Annual Conference 2010*, pp. 696–701, Aug 2010.

[6] T. Vimala and U. Rajaram, "Analysis of rssi in diverse field," in *International Conference on Information Communication and Embedded Systems (ICICES2014)*, pp. 1–3, Feb 2014.

[7] S. Shue, L. E. Johnson, and J. M. Conrad, "Utilization of xbee zigbee modules and matlab for rssi localization applications," in *SoutheastCon 2017*, pp. 1–6, March 2017.

[8] Sukhoon Jung, Choon-oh Lee, and Dongsoo Han, "Wi-fi fingerprint-based approaches following log-distance path loss model for indoor positioning," in

2011 IEEE MTT-S International Microwave Workshop Series on Intelligent Radio for Future Personal Terminals, pp. 1–2, Aug 2011.

[9] K. Nomura and F. Sato, "A performance study of zigbee network under wi-fi interference," in *2014 17th International Conference on Network-Based Information Systems*, pp. 201–207, Sep. 2014.

[10] Yoppy, R. H. Arjadi, H. Candra, H. D. Prananto, and T. A. W. Wijanarko, "Rssi comparison of esp8266 modules," in *2018 Electrical Power, Electronics, Communications, Controls and Informatics Seminar (EECCIS)*, pp. 150–153, Oct 2018.

[11] M. Kusriyanto and A. A. Putra, "Weather station design using iot platform based on arduino mega," in *2018 International Symposium on Electronics and Smart Devices (ISESD)*, pp. 1–4, Oct 2018.

[12] I. Kuzminykh, A. Snihurov, and A. Carlsson, "Testing of communication range in zigbee technology," in *2017 14th International Conference The Experience of Designing and Application of CAD Systems in Microelectronics (CADSM)*, pp. 133–136, Feb 2017.

[13] M. Phunthawornwong, E. Pengwang, and R. Silapunt, "Indoor location estimation of wireless devices using the log-distance path loss model," in *TENCON 2018 - 2018 IEEE Region 10 Conference*, pp. 0499–0502, Oct 2018.

[14] G. Sindura, K. Ram Prakash, and P. Salil, "Control of electromagnetic waves through electromagnetic shielding," in *2011 International Conference on Emerging Trends in Electrical and Computer Technology*, pp. 448–452, March 2011.

Appendices

The code in Arduino Mega which connects with Xbee is shown below.

```
int LED = 13;
char c;

void setup()
{
  pinMode(LED, OUTPUT);//
  Serial.begin(9600);
  Serial1.begin(9600);
}

void loop(){
  if (Serial.available()>0)
  {
  c = Serial.read();
  Serial1.write(c);
  Serial.print(c);
  if (c == 'a')
    {
     digitalWrite(LED,HIGH);
      }
     else if (c == 's'){
      digitalWrite(LED,LOW);
      }
    }
}
```

The code in ESP8266 module is shown as below.

```
#include <ESP8266WiFi.h>

#define led 2 // Control LED
const char *ssid     = "TcpAGV";//Define SSID
const char *password = "12341234";//wifi code
const char *host = "192.168.43.168";//IP of server
const int tcpPort = 1010;//Port of server
WiFiClient client;

void setup()
{
    Serial.begin(115200);
    pinMode(led,OUTPUT);
    delay(10);
    Serial.println();
    Serial.print("Connecting␣to␣");//
    Serial.println(ssid);
    WiFi.begin(ssid, password);//Start
     //Detect connected or not
```

```
    while (WiFi.status() != WL_CONNECTED)
    {
        delay(500);
    }

    Serial.println("");
    Serial.println("WiFi connected");
    Serial.println("IP address: ");
    Serial.println(WiFi.localIP());
}

void loop()//If not connected, make connection again
{
    while (!client.connected())
    {
        if (!client.connect(host, tcpPort))
        {
            Serial.println("connection....");
            delay(500);
        }
    }

    while (client.available())
    {
        double ch = WiFi.RSSI();
        char val = client.read();
        if(val=='a'){
            client.println("LEFT");
        }
        if(val=='d')
        {
            client.println("RIGHT");
        }
        if(val=='w'){
            client.println("FORWARD");
        }
        if(val=='s')
        {
            client.println("BACK");
        }
        if(val=='x'){
            client.println("STOP");
        }
        if(val=='1'){
            client.println(ch);
        }
        Serial.println(val);
    }
}
```

```
}
}
```

The code in Arduino Mega which drives the car is shown as below.

```
#include<SoftwareSerial.h>

#define DIR1A 24    //Motor 1(Motor)
#define DIR1B 25
#define PWM1  6    //Motor 1 (L298N) PWM

#define DIR2A 22    //Motor 2
#define DIR2B 23
#define PWM2  5    //Motor 2 PWM

#define DIR3A 26   //Motor 3
#define DIR3B 27
#define PWM3   7   //Motor 3 PWM

const int xb_rx = 10;
const int xb_tx = 11;
const int ble_rx = 12;
const int ble_tx = 13;
SoftwareSerial Xbee(xb_rx,xb_tx);
SoftwareSerial ble(ble_rx,ble_tx);

float Vx,Vy;
float V1,V2,V3;

char ch;

void setup()
{
  pinMode(ledPin, OUTPUT);
  Serial.begin(115200);
//motor set
 pinMode(DIR1A,OUTPUT);
  pinMode(DIR1B,OUTPUT);
  pinMode(DIR2A,OUTPUT);
  pinMode(DIR2B,OUTPUT);
  pinMode(DIR3A,OUTPUT);
  pinMode(DIR3B,OUTPUT);
  pinMode(PWM1,OUTPUT);
  pinMode(PWM2,OUTPUT);
  pinMode(PWM3,OUTPUT);

  pinMode(PWM1,0);
  pinMode(PWM2,0);
  pinMode(PWM3,0);
```

```
}

void Forward(float speed) //Forward along Y axis
{
  Vy = - speed;

  V1 = - 2 * Vy;
  V2 = 2 * Vy;

  digitalWrite(DIR1A,HIGH);
  digitalWrite(DIR1B,LOW);
  analogWrite(PWM1, V1);

  digitalWrite(DIR2A,LOW);
  digitalWrite(DIR2B,HIGH);
  analogWrite(PWM2,- V2);

  analogWrite(PWM3,0);
}

void Back(float speed) //Move back along Y axis
{
  Vy = - speed;
  V1 = - 2 * Vy;
  V2 = 2 * Vy;

  digitalWrite(DIR1A,LOW);
  digitalWrite(DIR1B,HIGH);
  analogWrite(PWM1, V1);

  digitalWrite(DIR2A,HIGH);
  digitalWrite(DIR2B,LOW);
  analogWrite(PWM2, - V2);

  analogWrite(PWM3,0);
}

void Right(float speed) //Move right along X axis
{
  Vx = 2*speed;
  Vy = 0;
  V1 = - 0.5 * Vx - 0.86 * Vy;
  V2 = - 0.5 * Vx + 0.86 * Vy;
  V3 = Vx;

  digitalWrite(DIR1A,HIGH);
  digitalWrite(DIR1B,LOW);
  analogWrite(PWM1, - V1);

```

```
  digitalWrite(DIR2A,HIGH);
  digitalWrite(DIR2B,LOW);
  analogWrite(PWM2, - V2);

  digitalWrite(DIR3A,LOW);
  digitalWrite(DIR3B,HIGH);
  analogWrite(PWM3,V3);
}

void Left(float speed) //Move left along X axis
{
  Vx = 2 * speed;
  Vy = 0;
  V1 = - 0.5 * Vx - 0.86 * Vy;
  V2 = - 0.5 * Vx + 0.86 * Vy;
  V3 = Vx;

  digitalWrite(DIR1A,LOW);
  digitalWrite(DIR1B,HIGH);
  analogWrite(PWM1,- V1);

  digitalWrite(DIR2A,LOW);
  digitalWrite(DIR2B,HIGH);
  analogWrite(PWM2,- V2);

  digitalWrite(DIR3A,HIGH);
  digitalWrite(DIR3B,LOW);
  analogWrite(PWM3,V3);
}

void Stop()             //Stop moving
{
  analogWrite(PWM1,0);
  analogWrite(PWM2,0);
  analogWrite(PWM3,0);
}

void loop()
{
//while (ble.available()>0)   //For Bluetooth control
//while (Xbee.available()>0)  //For Xbee control
while (Serial.available()>0)  //For ESP8266 control
{
    ch=Serial.read();
    if (ch=='w')
      {
        Forward(100);delay(2000);
        Serial.print("Forward");
        delay(100);
```

```
        }
    if (ch=='s')
      {
        Back(100);    delay(2000);
        Serial.print("Back");
        delay(100);
      }
    if (ch=='a')
      {
        Left(100);    delay(2000);
        Serial.print("Left");
        delay(100);
      }
    if (ch=='d')
      {
        Right(100);   delay(2000);
        Serial.print("Right");
        delay(100);
      }
    if (ch=='x')
      {
        Stop();   delay(2000);
        Serial.print("Stop");
        delay(100);
      }
}
}
```