

AUTONOMOUS SMART ROBOT
SURVEILLANCE SYSTEM
(ASRSS)

GUAT SI JIE

Bachelor of Computer Science

UNIVERSITI MALAYSIA PAHANG



SUPERVISOR'S DECLARATION

I/We* hereby declare that I/We* have checked this thesis/project* and in my/our* opinion, this thesis/project* is adequate in terms of scope and quality for the award of the degree of *Doctor of Philosophy/ Master of Engineering/ Master of Science in

(Supervisor's Signature)

Full Name :

Position :

Date :

(Co-supervisor's Signature)

Full Name :

Position :

Date :



STUDENT'S DECLARATION

I hereby declare that the work in this thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at Universiti Malaysia Pahang or any other institutions.

(Student's Signature)

Full Name : GUAT SI JIE

ID Number : 950423075557

Date : 12 December 2018

AUTONOMOUS SMART ROBOT SURVEILLANCE SYSTEM (ASRSS)

GUAT SI JIE

Thesis submitted in fulfillment of the requirements
for the award of the degree of
Bachelor of Computer Science

Faculty of Computer Systems & Software Engineering
UNIVERSITI MALAYSIA PAHANG

December 2018

ACKNOWLEDGEMENTS

I would first like to thank my supervisor Dr. Noraziah Ahmad of the FSKKP at University Malaysia Pahang. Dr. Noraziah Ahmad mentored me during the whole process of developing the project. She consistently guides me with care and wisdom and gave me advice to improve my project.

I also would like to appreciate the critics and suggestion by the lecturers who is involved in judging my project. Without their passionate participation and input, the presentation of project could not have been successfully conducted.

Finally, I must express my big thanks to my parents for providing me support and love throughout my years of studying in UMP and the period of writing this thesis. This achievement would not have been possible without them. Thank you.

ABSTRAK

Tesis ini menerangkan sistem pengawasan robot pintar autonomi yang bertindak sebagai sistem pengawasan rumah. Robot pengawasan secara automatik boleh bergerak dari satu tinjauan ke yang lain dan merakamkan serta menghantar pemberitahuan kepada pengguna jika ada objek yang bergerak yang direkam. Ia akan memuat naik video ke Google Drive. Ia boleh mengelakkan objek bergerak jika ia bergerak ke arah robot.

ABSTRACT

This thesis describes an autonomous smart robot surveillance system which acts as a home surveillance system. The surveillance robot can automatically move from a surveilling spot to another and record plus send a notification to the user if there is a moving object recorded. It will upload the video to Google Drive. It can avoid moving object if it is moving in the direction of the robot.

TABLE OF CONTENT

DECLARATION	
TITLE PAGE	
ACKNOWLEDGEMENTS	ii
ABSTRAK	iii
ABSTRACT	iv
TABLE OF CONTENT	v
LIST OF TABLES	xii
LIST OF FIGURES	xiii
LIST OF SYMBOLS	xvii
LIST OF ABBREVIATIONS	xviii
CHAPTER 1 INTRODUCTION	1
1.1 INTRODUCTION	1
1.2 PROBLEM STATEMENT	2
1.3 GOAL AND OBJECTIVE	2
1.4 SCOPE	2
1.5 SIGNIFICANCE	3
1.6 THESIS ORGANIZATION	3
CHAPTER 2 LITERATURE REVIEW	5
2.1 INTRODUCTION	5
2.2 CONCEPT	5
2.2.1 Autonomous Surveillance Concept	5

2.2.2	Navigation System Concept	5
2.3	TECHNOLOGIES	6
2.3.1	Single Board Computer	6
2.3.1.1	Raspberry Pi Model 3 B+	6
2.3.1.2	Arduino Uno	7
2.3.1.3	Beaglebone Black	8
2.3.1.4	Comparison of Single Board Computer	9
2.3.2	Sensors	9
2.3.2.1	Webcam	10
2.3.2.2	Ultrasonic Sensor	11
2.3.2.3	Infrared Sensor	12
2.3.2.4	Comparison of Single Board Computer	12
2.4	INVESTIGATION OF EXISTING SYSTEM	13
2.4.1	Existing Surveillance System	13
2.4.1.1	DAYTECH IP Camera CCTV	13
2.4.1.2	GOQ Q7 Robot Magnetic Surveillance CCTV	14
2.4.1.3	ESCAM Robot QN02	14
2.4.1.4	Comparison of Single Board Computer	15
2.4.2	Existing Movement Tracking System	16
2.4.2.1	SIP-enabled Surveillance Patrol Robot	16
2.4.2.2	Implementation of Tracking of a Moving Object Based on Camshift Approach with a UAV	17
2.4.2.3	Comparison of Single Board Computer	19
2.5	METHODOLOGY MODELS	20

2.5.1	Waterfall Model	20
2.5.2	Iterative SDLC Model	21
2.5.3	Agile Model	22
2.5.4	Comparison of Methodology Models	23
2.5.5	Conclusion of Methodology Models	24
 CHAPTER 3 METHODOLOGY		25
3.1	Overview	25
3.2	Methodology	26
3.2.1	Planning Phase	26
3.2.2	Analysis Phase	26
3.2.2.1	Verification of Idea and Technologies	26
3.2.2.2	Verification of Hardware Components	27
3.2.2.3	Getting Requirements from Client	27
3.2.2.4	Use Case Definition	28
3.2.2.5	Context Diagram Design	29
3.2.3	Design Phase	30
3.2.3.1	Architecture Pattern	30
3.2.3.2	Motion Detection Algorithm	32
3.2.3.3	Ultrasonic Distance Detection Algorithm	33
3.2.3.4	ASRSS Physical Design	33
3.2.3.5	Static Detection	34
3.2.3.6	Module	35
3.2.3.2.1	ASRSS	36

3.2.3.2.1.1	Main Module	36
3.2.3.2.1.2	Sensor Module	36
3.2.3.2.1.3	Navigation Module	37
3.2.3.2.1.4	Database Module	37
3.2.3.2.1.5	Camera Module	37
3.2.3.2.2	ASRSS_App	38
3.2.3.2.2.1	MainPage Module	38
3.2.3.2.2.2	Database Module	38
3.2.3.2.2.3	ViewVideo Module	39
3.2.3.2.2.4	Notification Module	39
3.2.3.7	Database Design	40
3.2.3.8	Interface Design	41
3.2.3.3.1	MainActivity	42
3.2.3.3.2	ViewSavedVideoActivity	42
3.2.3.3.3	ViewNotificationActivity	43
3.2.3.3.4	Storyboard	44
3.2.4	Development Phase	46
3.2.4.1	First Iteration	46
3.2.4.2	Second Iteration	46
3.2.4.3	Third Iteration	47
3.2.5	Testing Phase	47
3.3	Hardware and Software Requirements	48

3.3.1	Hardware Requirements	48
3.3.1.1	Raspberry Pi Model 3 B+	48
3.3.1.2	Ultrasonic sensor	49
3.3.1.3	Infrared sensor	50
3.3.1.4	Webcam	50
3.3.1.5	DC Motor	51
3.3.1.6	Car frame	51
3.3.1.7	Android Phone	52
3.3.1.8	Computer	53
3.3.1.9	Summary of Hardware Requirements	54
3.3.2	Software Requirements	54
3.3.2.1	Microsoft Word 2016	55
3.3.2.2	Ninja-IDE v2.3	55
3.3.2.3	Microsoft Project 2016	56
3.3.2.4	Android Studio 3.1.4	57
3.3.2.5	Summary of Software Requirements	57
3.4	Gantt chart	58
CHAPTER 4 RESULTS AND DISCUSSION		59
4.1	Introduction	59
4.2	Implementation	59
4.2.1	Hardware Implementation	59
4.2.2	Software Implementation	60

4.2.2.1	Home Activity	60
4.2.2.2	View Saved Videos Activity	61
4.2.2.3	View Notification Activity	65
4.2.2.4	Notification	70
4.2.2.5	Motion Detection	72
4.2.2.6	Static Detection	75
4.2.2.7	Record Video	77
4.2.2.8	Register Notification	77
4.2.2.9	Hardware Failure	79
4.2.2.10	Swap Spot	79
4.2.2.11	Avoiding Object	82
4.2.2.12	Infrared Sensor	86
4.2.2.13	Ultrasonic Sensor	88
4.3	Testing Result	90
4.4	User Manual	90
CHAPTER 5 CONCLUSION		92
5.1	Introduction	92
5.2	Constraint	92
5.2.1	Power Limitation	92
5.2.2	Motor Power Limitation	93
5.2.3	Port 1433 Blocked	93
5.2.4	Android Drive API Methods Depreciated	93
5.2.5	Virtual Environment in Raspberry Pi Does Not Support I2C	93

5.3	Future Improvements	93
5.4	Conclusion	94
	REFERENCES	1
	APPENDIX A Gantt Chart	4
	APPENDIX B Software Requirement Specification (SRS)	8
	APPENDIX C Software Design Description (SDD)	9
	APPENDIX D Functional Requirements Gathering	10
	APPENDIX E FSKKP Information Gathering Approval Letter	11
	APPENDIX F User acceptance test Report	12
	APPENDIX G User Manual	13
	APPENDIX H Turnitin report	14

LIST OF TABLES

Table 2.1	Comparison of Single Board Computer	9
Table 2.2	Comparison of Sensors	12
Table 2.3	Comparison of existing systems	15
Table 2.4	Comparison Existing Movement Tracking System	19
Table 2.5	Comparison of Methodology Models	23
Table 3.1	List of Use Case	29
Table 3.2	Summary of Hardware Requirements	54
Table 3.3	Summary of Software Requirements	57

LIST OF FIGURES

Figure 2.1	Raspberry Pi 3 Model B+	7
Figure 2.2	Arduino Uno	8
Figure 2.3	Beaglebone Black	9
Figure 2.4	Webcam	10
Figure 2.5	Ultrasonic sensor	11
Figure 2.6	Infrared sensor	12
Figure 2.7	DAYTECH IP Camera CCTV	13
Figure 2.8	GOQ Q7 Robot Magnetic Surveillance CCTV	14
Figure 2.9	ESCAM Robot QN02	15
Figure 2.10	Position measurement using the triangulation method	16
Figure 2.11	Positioning formula	16
Figure 2.12	Zeroth moment formula	18
Figure 2.13	First order moment formula and coordination of xc and yc formula	18
Figure 2.14	Second and first moment formula	18
Figure 2.15	Object orientation formula	18
Figure 2.16	Length and width of probability distribution formula	19
Figure 2.17	Waterfall Model	21
Figure 2.18	Iterative SDLC Model	22
Figure 2.19	Agile Model	23
Figure 3.1	Use Case Diagram of ASRSS	29
Figure 3.2	Context Diagram of ASRSS	30
Figure 3.3	MVC architecture design of ASRSS_App	31
Figure 3.4	3 tier layered architecture design of ASRSS	32
Figure 3.5	Top View of ASRSS Sketched Design	34
Figure 3.6	Bottom View of ASRSS Sketched Design	34
Figure 3.7	Main Module	36
Figure 3.8	Sensor Module	36
Figure 3.9	Navigation Module	37
Figure 3.10	Database Module	37
Figure 3.11	Recording Module	38
Figure 3.12	MainPage Module	38
Figure 3.13	LiveStream Module	39
Figure 3.14	ViewVideo Module	39

Figure 3.15	Notification Module	40
Figure 3.16	ERD design of ASRSS	40
Figure 3.17	ERD design of ASRSS_App	41
Figure 3.18	ERD design of Azure SQL database	41
Figure 3.19	Interface design of HomeActivity	42
Figure 3.20	Interface design of ViewSavedVideoActivity	43
Figure 3.21	Interface design of ViewNotificationActivity	44
Figure 3.22	Storyboard of ASRSS_App	45
Figure 3.23	: Raspberry Pi Model 3 B+	49
Figure 3.24	Ultrasonic sensor	49
Figure 3.25	Infrared sensor	50
Figure 3.26	Webcam	50
Figure 3.27	DC Motor	51
Figure 3.28	Car frame	52
Figure 3.29	Redmi 5 Plus	53
Figure 3.30	HP Pavilion Notebook	54
Figure 3.31	Icon for Microsoft Word 2016	55
Figure 3.32	Ninja-IDE icon	56
Figure 3.33	Icon for Microsoft Project 2016	56
Figure 3.34	Android Studio 3.1.4	57
Figure 4.1	ASRSS system hardware configuration	60
Figure 4.2	Home Activity	61
Figure 4.3	View Saved Videos Activity	62
Figure 4.4	View Saved Videos Activity Opening Video	62
Figure 4.5	View Saved Videos Activity Playing Video	63
Figure 4.6	Embedded Code for Searching Video and Getting Result from Search	64
Figure 4.7	View Notification Activity	65
Figure 4.8	View Notification Activity	66
Figure 4.9	Algorithm for Retrieving and Deleting SQLite Database	67
Figure 4.10	Embedded Code for Refreshing Interface	67
Figure 4.11	Algorithm for Connecting Azure SQL Database	68
Figure 4.12	Embedded Code for Retrieving Azure SQL Database	69
Figure 4.13	Embedded Code for Deleting Azure SQL Database	69
Figure 4.14	Notifications	70

Figure 4.15	Embedded Code for PushUserNotification	71
Figure 4.16	Embedded Code for Creating Notification Channel	71
Figure 4.17	Embedded Code for Motion Detection	72
Figure 4.18	Embedded Code for Motion Detection	73
Figure 4.19	Demo for Static Video	74
Figure 4.20	Demo for Detecting Moving Objects	74
Figure 4.21	Embedded Code for Static Detection	75
Figure 4.22	Demo for detecting static image	76
Figure 4.23	Demo for detecting static image after a minute	76
Figure 4.24	Embedded Code for Record Video	77
Figure 4.25	Embedded Code for Register Notification	77
Figure 4.26	Embedded Code for Connecting and Inserting Data to Azure SQL Database	78
Figure 4.27	Embedded Code for Connecting, Inserting Data and Getting Data from SQLite	78
Figure 4.28	Embedded Code for Detect Camera Failure and Internet Connection	79
Figure 4.29	Demo for Swapping Spot	80
Figure 4.30	Demo for Swapping Spot	80
Figure 4.31	Demo for Swapping Spot	81
Figure 4.32	Embedded Code for Swap Spot	81
Figure 4.33	Demo for Avoiding Object and Detecting Object in Front	82
Figure 4.34	Demo for Avoiding Object and Detecting Object in Front	83
Figure 4.35	Demo for Avoiding Object and Detecting Object in Front	83
Figure 4.36	Demo for Avoiding Object and Detecting Object Behind	84
Figure 4.37	Demo for Avoiding Object and Detecting Object Behind	84
Figure 4.38	Demo for Avoiding Object and Detecting Object Behind	85
Figure 4.39	Embedded Code for Avoiding Object and Detecting Object	85
Figure 4.40	Testing the Infrared Sensor with an Object	86

```

21 def setup(self):
22     GPIO.setwarnings(False)
23     GPIO.setmode(GPIO.BCM) # Numbers GPIOs by physical location
24     GPIO.setup(self.Gpin, GPIO.OUT) # Set Green Led Pin mode to output
25     GPIO.setup(self.Rpin, GPIO.OUT) # Set Red Led Pin mode to output
26     GPIO.setup(self.BTMSensorMid, GPIO.IN) # Set BtnPin's mode is input, and pull up to high level(3.3V)
27     GPIO.setup(self.BTMSensorRight, GPIO.IN)
28     GPIO.setup(self.BTMSensorLeft, GPIO.IN)
29     GPIO.setup(self.BackSensorRight, GPIO.IN)
30     GPIO.setup(self.BackSensorLeft, GPIO.IN)
31
32 def LBTMInfra(self):
33     return GPIO.input(self.BTMSensorLeft)
34
35 def RBTMInfra(self):
36     return GPIO.input(self.BTMSensorRight)
37
38 def MBTMInfra(self):
39     return GPIO.input(self.BTMSensorMid)
40
41 def LBackInfra(self):
42     return GPIO.input(self.BackSensorLeft)
43
44 def RBackInfra(self):
45     return GPIO.input(self.BackSensorRight)
46

```

Figure 4.41 Embedded Code for Infrared Sensor 87

Figure 4.42 Testing for Ultrasonic Sensor 88

Figure 4.43 Distance Result of the Ultrasonic Sensor 88

Figure 4.44 Embedded Code for Ultrasonic Sensor 89

LIST OF SYMBOLS

SBPWM	Simple Boost Pulse Width Modulation
ZSI	Z source inverter

LIST OF ABBREVIATIONS

SBPWM	Simple Boost Pulse Width Modulation
ZSI	Z source inverter

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

As we are getting closer to Industrial Revolution 4.0, we are blessed with capable IoT technology that empowers our lives.(Moore, 2018) Soft computing and artificial intelligence are successfully applied to our everyday life, such as machinery control, robot manipulation and engineering application. With the integration of nonlinear system and communication technologies, creating a secure environment for our home is a non-issue now.(Brandon, 2014)

Home safety is one of the most critical issues as we tend to keep our valuable assets at home. The most convenient solution to secure our home is to use Closed Circuit TV (CCTV) technology that films a fixed area at fixed angle all the time. CCTV holds a record of film which is about what was happening in the area for the users. However, there is limitations for CCTV. Because that it only surveillance a fixed area, it is lacking flexibility to film a particular event ongoing. The filmed video can only cover a footage which is filmed at the fixed area.

A robot with sensors for detecting environment and motor for moving around surveillance area which is programmed to watch over an area is a promising solution as a conventional home surveillance system. To make surveillance job possible, ultrasonic sensors, a camera and a navigating algorithm is implemented. To enhance the ability of the robot to record important event, computer vision algorithm is implemented in the system to detect movements. If the system detects a moving person at home, it will send notification to the owner of the device.

1.2 PROBLEM STATEMENT

Thief crimes often happen at a regular basis. It often happens when the victim is away from the house. The thieves steal away assets of the victim and causes massive amount of lost to the victim.

Regular CCTV do have the problem of mobility since that it does not have the motor and the algorithm to adjust the angle. That means that there is always blind spot for crime to be happen. Some important incidents may not be recorded since it happened in the blind spot.

Installing surveillance camera to cover the whole house requires a high cost to implement it. The cost of the setup is mainly cause by the quantity of the components that is required to setup the system.

Often, thief crimes are not reported it is found out later. Normally it may take hours if not days for the victim to realise what is happening in his house. It will cause a long delay in reporting the incident.

1.3 GOAL AND OBJECTIVE

- i. To build a smart surveillance system that detects nearby obstacles.
- ii. To build a raspberry pi system that can move from a point to another.
- iii. To apply the feature of sending notification and video to a phone.

1.4 SCOPE

- i. User

Only one user can interact with the autonomous surveillance robot system at a time. The user can access the system using the ASRSS mobile app. User must be able to operate Linux system and command windows.

ii. Data

The data of video is stored in local sd card and cloud storage. The video recorded is only encoded in H.264 format. The database is only stored in the surveillance robot, Android smartphone and Azure SQL database. The internet connection must be available to the system and port 1433 must be unblock for the communication to the Azure SQL database.

iii. System

Autonomous surveillance robot system is built on Raspberry Pi system. It will have the ability to navigate from a point to another. It can detect moving object. If moving person is detected, it will send notification to the user. The system must be supplied with a 5 Volt 2 Ampere power supply.

iv. Environment

Autonomous surveillance robot system is designed to only work in indoor environment. The indoor environment must a guiding tile to navigate it from a point to another. The path of the guiding tile must be straight and without any obstacle.

1.5 SIGNIFICANCE

- i. Reduce the cost of surveying a large surveillance area.
- ii. Reduce the risk of thief.
- iii. Reduce the time to make a police report.

1.6 THESIS ORGANIZATION

There are five chapters in this report which are introduction, literature review, Methodology, implementation and result discussion and finally conclusion. Each chapter will discuss its own aspects related to the project.

REFERENCES

- Academy, C. M. R. (n.d.). What Is an Ultrasonic Sensor? Retrieved from http://education.rec.ri.cmu.edu/content/electronics/boe/ultrasonic_sensor/1.html
- Academy, C. M. R. (2018). What is an IR Sensor? Retrieved from What is an IR Sensor?
- Arduino. (2018a). Arduino Uno Rev3. Retrieved from <https://store.arduino.cc/usa/arduino-uno-rev3>
- Arduino. (2018b). Download the Arduino IDE. Retrieved from <https://www.arduino.cc/en/Main/Software>
- Beagleboard.org. (2018). BeagleBone Black. Retrieved from <https://beagleboard.org/black>
- Borong, H. (2018). GOQ Q7 Robot Magnetic Wifi Surveillance IP Security Camera CCTV Cam 960P HD. Retrieved from <https://www.lazada.com.my/products/goq-q7-robot-magnetic-wifi-surveillance-ip-security-camera-cctv-cam-960p-hd-i125945584-s138649433.html?spm=a2o4k.searchlist.list.3.38a1d7abfME3b2&search=1>
- Brandon, J. (2014). 5 Uses for the Surveillance Robot of Tomorrow. Retrieved from <https://www.cio.com/article/2463903/robotics/5-uses-for-the-surveillance-robot-of-tomorrow.html>
- Codecademy. (2018). MVC: Model, View, Controller. Retrieved from <https://www.codecademy.com/articles/mvc>
- Coşkun, M., & Ünal, S. (2016). Implementation of Tracking of a Moving Object Based on Camshift Approach with a UAV. *Procedia Technology*, 22(October 2015), 556–561. <https://doi.org/10.1016/j.protcy.2016.01.116>
- DAYTECH. (2018), DAYTECH HD720P/HD960P CCTV IP Camera WiFi Surveillance Network Security Camera Waterproof Outdoor Two Way Audio Night Vision. (2018). Retrieved from <https://www.lazada.com.my/products/daytech-hd720phd960p-cctv-ip-camera-wifi-surveillance-network-security-camera-waterproof-outdoor-two-way-audio-night-vision-i231719854-s307284688.html?spm=a2o4k.searchlist.list.7.dd7d17f2U9djQw&search=1>
- Developers. (2018). androidstudio. Retrieved from <https://developer.android.com/studio/>
- ESCAM. (2018). ESCAM QN02. Retrieved from <http://www.escam.cn/product/showproduct.php?lang=en&id=99>

- Existek. (2017). Sdlc Models Explained: Agile, Waterfall, V-shaped, Iterative, Spiral | Existek Blog. Retrieved from <https://existek.com/blog/sdlc-models/>
- Foundation, R. P. (2018). Raspberry Pi Downloads - Software For the Raspberry Pi. Retrieved from <https://www.raspberrypi.org/downloads/>
- GSMarena. (2017). Xiaomi Mi A1. Retrieved from https://www.gsmarena.com/xiaomi_mi_a1-8776.php
- Layton, J. (2018). How Webcams Work. Retrieved from <https://computer.howstuffworks.com/webcam1.htm>
- McManus, S. (2018). Top 10 Programming Languages Ported To the Raspberry Pi. Retrieved from <http://www.dummies.com/computers/raspberry-pi/top-10-programming-languages-ported-to-the-raspberry-pi/>
- Microsoft. (2018a). Project. Retrieved from <https://products.office.com/en-my/project/project-and-portfolio-management-software?tab=tabs-1>
- Microsoft. (2018b). Word 2016. Retrieved from <https://products.office.com/en-my/word>
- Moore, M. (2018). What is Industry 4.0? Everything you need to know. Retrieved from <https://www.techradar.com/news/what-is-industry-40-everything-you-need-to-know>
- Nayyar, A. (2017). Top 8 Ides For Raspberry Pi. Retrieved from <http://opensourceforu.com/2017/06/top-ides-raspberry-pi/>
- Ninja-IDE. (2018). Ninja-IDE. Retrieved from <http://ninja-ide.org/>
- Richardson, M. (2012). Get Started with BeagleBone. Retrieved from <https://makezine.com/projects/make-32/get-started-with-beaglebone/>
- Sami, M. (2012). Software Development Life Cycle Models and Methodologies. Retrieved from <https://melsatar.blog/2012/03/15/software-development-life-cycle-models-and-methodologies/>
- Tseng, C. C., Lin, C. L., Shih, B. Y., & Chen, C. Y. (2013). SIP-enabled Surveillance Patrol Robot. *Robotics and Computer-Integrated Manufacturing*, 29(2), 394–399. <https://doi.org/10.1016/j.rcim.2012.09.009>
- Watson, J. A. (2017). Hands-on with the Raspberry Pi 3 Model B+. Retrieved from <http://www.zdnet.com/article/hands-on-raspberry-pi-3-model-b/>