รายงานการวิจัยฉบับสมบูรณ์

ระบบบันทึกภาพอัตโนมัติราคาประหยัด
Automatic low cost picture recording system

นาย อรรถสิทธิ์ หล่าสกุล

ได้รับทุนสนับสนุนงานวิจัยจากเงินรายได้ ประจำปีงบประมาณ 2559
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ชื่อโครงการ (ภาษาไทย) ระบบบันทึกภาพอัตโนมัติราคาประหยัด

แหล่งเงิน เงินรายได้คณะวิศวกรรมศาสตร์

ประจำปีงบประมาณ พ.ศ.2559 **จำนวนเงินที่ได้รับการสนับสนุน**.....130,000 **บาท**

ระยะเวลาทำการวิจัย 1 **ปี ตั้งแต่** 1 ตุลาคม 2558.....ถึง.....30 กันยายน 2559

ชื่อ-สกุล หัวหน้าโครงการ และผู้ร่วมโครงการวิจัย พร้อมระบุ หน่วยงานต้นสังกัด

ชื่อ-สกุล (ภาษาไทย) .....อรรถสิทธิ์ หล่าสกุล.....

ชื่อ-สกุล (ภาษาอังกฤษ) .....ATTASIT LASAKUL.....

ตำแหน่งทางวิชาการ ...รองศาสตราจารย์..... **สัดส่วนการวิจัย** .....100 %.....

ภาควิชา .....สาขาวิชาวิศวกรรมคอมพิวเตอร์..... **คณะ** .....วิศวกรรมศาสตร์.....

โทรศัพท์ ...0840270185..... **โทรสาร** .....

E-mail .....klattasi@kmitl.ac.th.....

บทคัดย่อ

ในปัจจุบัน ระบบการเก็บภาพด้วยกล้องวงจรปิด มีใช้อย่างแพร่หลาย โดยเฉพาะกล้องเก็บภาพที่เป็นแบบระบบดิจิตอล ซึ่งสามารภเก็บภาพได้บนอุปกรณ์เก็บข้อมูลขนาดเล็กเช่น จานแม่เหล็กขนาดเล็ก ซึ่งราคานับวันก็จะถูกลงเรื่อยๆ ข้อดีอีกประการของการใช้ระบบดิจิตอลคือ มีการรับส่งหรือควบคุมการทำงานของอุปกรณ์ไม่ว่าจะเป็นกล้องเอง เช่น การขยายภาพ, การหมุนมุมกล้อง เหล่านี้สามารถทำได้ผ่านระบบที่เป็นเครือข่ายอินเตอร์เนท นั่นเอง ซึ่งในสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ก็มีการใช้งานระบบดังกล่าวด้วยเช่นกัน ซึ่งการทำงานในระบบดังกล่าวนั้นต้องใช้งบประมาณที่สูงมากโดยเฉพาะตัวประมวลผลกลาง (Network Video Recorder: NVR) การบำรุงรักษาการปรับปรุงเพื่อให้ตรงกับความต้องการของผู้ใช้ก็ทำได้ยากเพราะต้องมีข้อตกลงหรือค่าใช้จ่ายเพิ่มขึ้นนั่นเอง ด้วยเหตุดังกล่าวนี้ ผู้วิจัยจึงมีแนวคิดว่า ในปัจจุบันนั้น อุปกรณ์ด้านการเชื่อมต่ออินเตอร์เนท, ตัวกล้องดิจิตอล หรือแม้แต่อุปกรณ์ด้านคอมพิวเตอร์ฝังตัวขนาดเล็ก นั้นมีราคาที่ต่ำลงมาก หากนำมาสร้างเป็นระบบเก็บภาพให้สามารถทำงานพื้นฐานได้ และสามารถเพิ่มเติมฟังก์ชั่นหรือหน้าที่พิเศษได้เอง ก็จะเป็นประโยชน์มากเพราะนอกจากจะลดค่าใช้จ่ายลงเพราะราคาที่ต่ำลงแล้ว ในเรื่องการซ่อมบำรุงการพัฒนาระบบต่อไปก็สามารถทำได้อย่างอิสระโดยทันที ดังนั้นงานวิจัยนี้น่าจะเป็นต้นแบบหรือทางเลือกหนึ่งในการนำไปใช้งานได้เช่นกัน

คำสำคัญ : คอมพิวเตอร์ฝังตัว, การสื่อสารผ่านไวฟาย, กล้องวงจรปิดแบบเครือข่ายดิจิตอล

Research Title: ..... Automatic low cost picture recording system .....
Researcher: Assoc.Prof.Dr. Attasit LASAKUL
Faculty: Engineering **Department:** .....Computer engineering

ABSTRACT

Currently, the image digital recorder is widely used in many applications. Especially, in security purpose, because it can record the picture for long of limitation time. The picture data can be saved on Handy drive which is very small size and low cost. KMITL also have the same system configured. However, the system has a high cost for installation. Furthermore, has also high cost for maintenance or changing some configuration to suit of any purposes. In this research, proposed the new idea for constructing the image digital recorder system that has low cost. This system can be constructed by using low cost embedded computer (Raspberry pi3) with a small size of digital camera. The proposed software also was created by open source software (C, Python, JAVA). This makes the system is very low cost but good performance. The system can record the picture as a normal IP camera system, but has some special add-on function. The example of add-on function use for tracking the anything that has the specified color. This special function can freely be adding more by the developer because of all hardware/software are open for development. This could be very useful for any application in the future.

Keywords: Embedded computer, WIFI, IP Camera

กิตติกรรมประกาศ

ในการทำงานวิจัยครั้งนี้ จะไม่อาจสำเร็จได้เลย หากปราศจาก บุคคลและหน่วยงานเหล่านี้ ได้ให้ข้อมูลและคำเสนอแนะ อันเป็นประโยชน์ต่อผู้วิจัยเป็นอย่างมาก คือ ผศ.ไพศาล สิทธิโยภาศกุล คณะวิศวกรรมศาสตร์ พระจอมเกล้าเจ้าคุณทหารลาดกระบัง ซึ่งให้คำแนะนำที่เป็นประโยชน์ต่องานวิจัยนี้เป็นอย่างมาก และอีกคือกลุ่ม นักศึกษาชั้นปีที่ 4 ของหลักสูตรวิศวกรรมสารสนเทศ ที่ช่วยทำการทดลองทดสอบการทำงานของเครื่องที่ได้ทำวิจัยนี้ และขอขอบคุณ คุณพ่อและคุณแม่ ที่ให้กำลังใจ มาตลอดการทำการวิจัยนี้ จนทำให้งานสาเร็จลุล่วงไปด้วยดี และงานนี้จะไม่สำเร็จเลยหากปราศจากผู้ให้ทุนวิจัย ซึ่งในการวิจัยครั้งนี้ผู้วิจัยได้รับทุนสนับสนุนการวิจัย จากแหล่งทุน “เงินรายได้คณะวิศวกรรมศาสตร์ ประจำปีงบประมาณ พ.ศ. 2559” (สัญญาเลขที่ 2559 01 01018)

รองศาสตราจารย์ ดร. อรรถสิทธิ์ หล่าสกุล
(หัวหน้าโครงการวิจัย)

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	ก
บทคัดย่อภาษาอังกฤษ	ข
กิตติกรรมประกาศ	ค
สารบัญ	ง
สารบัญตาราง	ฉ
สารบัญภาพ	ช
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์ของการวิจัย	1
1.3 ขอบเขตของการวิจัย	2
1.4 วิธีดำเนินการวิจัย	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ	2
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง	3
2.1 ทฤษฎีที่เกี่ยวข้อง	3
บทที่ 3 วิธีดำเนินการวิจัย	7
3.1 ส่วนของ Hard ware	8
3.2 ส่วนของ Software	10
3.3 การใช้งานของระบบบันทึกภาพอัตโนมัติราคาประหยัด	14
3.3.1 การปรับเมื่อติดตั้งกล้องครั้งแรก	15
3.3.2 โหมดการบันทึกภาพ	17
3.3.2.1 การโหลดภาพมาตรวจสอบ	18
3.3.3 โหมดการตรวจจับวัตถุ	19
บทที่ 4 ผลการวิจัย	21
4.1 ผลวิจัยต่างๆ	21
4.2 จุดเด่นของงานวิจัยนี้	23
บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ	24
5.1 สรุปผลการวิจัยและคาเสนอแนะ	24
บทที่ 6 สรุปผลผลิตที่ได้จากงานวิจัย	25
บรรณานุกรม เอกสารอ้างอิง	26
ภาคผนวก	27

สารบัญ (ต่อ)

	หน้า
ภาคผนวก ก ส่วนของฮาร์ดแวร์	27
ภาคผนวก ข ส่วนของโปรแกรม	31
ข้อมูลประวัติคณะผู้วิจัย	57

สารบัญตาราง

ตารางที่		หน้า
3.1	แสดงตารางเวลาการดำเนินการวิจัย	13

สารบัญภาพ

ภาพที่	หน้า
2.1 เป็นตัวประมวลผลหลัก Raspberry pi2 ติดตั้งกล้องแล้วที่ใช้ในงานวิจัยนี้	3
2.2 แสดงตัวเชื่อมต่อไร้สายแบบ USB Type ที่ใช้ในงานวิจัย	4
2.3 เป็นเทคโนโลยี่โปรแกรมต่างๆที่นำมาพัฒนาด้านซอร์แวร์	5
2.4 บล็อกไดอะแกรมโปรแกรมหน้าที่พิเศษ (ตรวจจับวัตถุ) ที่ทดลองใช้	6
3.1 แนวคิดของการใช้งานของเครื่องบันทึกภาพอัตโนมัติ	7
3.2 แสดงส่วนประกอบของตัวกล้อง	8
3.3 รูปบอร์ด Raspberry Pi2 และส่วนประกอบเชื่อมต่ออื่นๆ	9
3.4 รูปบอร์ด Raspberry Pi2 ประกอบใส่กล่อง	9
3.5 โฟร์ชาร์ท โปรแกรมของส่วนตัวกล้อง	10
3.6 โฟร์ชาร์ท โปรแกรมของส่วนโปรแรกมควบคุม (GUI) ของผู้ใช้	11
3.7 แสดงการใช้ภาษาโปรแกรมแต่ละส่วน	13
3.8 แสดงตำแหน่งไฟกระพริบเป็นระยะแสดงว่ากล้องพร้อมใช้งาน	14
3.9 แสดงหน้าจอ GUI ที่ออกแบบให้ผู้ใช้สั่งงานบนคอมพิวเตอร์ผู้ใช้	15
3.10 แสดงตำแหน่งปุ่ม Camera Check บน GUI	16
3.11 แสดงการขั้นตอนการเลือก IP ที่ต้องการ	16
3.12 แสดงภาพที่ได้จากกล้องที่เราต้องการปรับตำแหน่งให้ถูกต้อง	17
3.13 แสดงตัวอย่างชื่อไฟล์ภาพที่เก็บไว้ในหน่วยความจำ	18
3.14 หน้าต่างใหม่เพื่อให้เลือกโหลดภาพจากกล้องหมายเลข IP เท่าใด	18
3.15 โหลดภาพในที่เก็บภาพของกล้องตามต้องการ	19
3.16 แสดงส่วนปุ่มของการเลือกสีก่อนแล้วก็เลือกเริ่มการตรวจจับ	19
3.17 แสดงภาพตัวอย่างที่ตรวจจับพบวัตถุสีแดงในขนาดที่ต้องการ	20
4.1 แสดงตำแหน่งติดตั้งกล้องเพื่อการทดสอบอีกสถานที่	21
4.2 ผลตัวอย่างภาพที่ได้จากการทำงานในโหมดตรวจจับวัตถุสีเหลือ	22
4.3 ผลตัวอย่างภาพที่ได้จากการทำงานในโหมดตรวจจับวัตถุสีแดง	22

บทที่ 1
บทนำ

ในรายงานฉบับสมบูรณ์นี้จะได้ เรียงหัวข้อตามรูปแบบมาตฐาน คือ เริ่มจากปัญหาที่มา, ทฤษฎีและงานวิจัยที่ เกี่ยวข้อง, วิธีดำเนินการวิจัย และผลการวิจัย ซึ่งจะได้เน้นส่วนที่เป็นผลการวิจัย เพื่อการพัฒนาปรับปรุงต่อไป ได้อย่างมีประสิทธิภาพต่อไป

1.1 ความเป็นมาและความสำคัญของปัญหา

จากที่ได้กล่าวมาในตอนต้น ของระบบบันทึกภาพที่มีจำหน่ายใช้งานกันอยู่ในปัจุบันนั้น มีมากมายหลายยี่ห้อโดยเฉพาะที่เป็นแบบเครือข่าย ซึ่งเป็นที่นิยมมากเพราะมีความสามารถในการควบคุมการทำงานผ่านระบบอินเทอร์เนทได้ทำให้การใช้งานสะดวกมากขึ้น แต่ระบบกล้องแบบเครือข่ายนี้ราคาการติดตั้ง ยังมีราคาที่สูงอยู่ และนอกจากนี้คุณสมบัติที่มีอาจไม่เหมาะสมกับสถานที่ที่ต้องการ หรือการเพิ่มหน้าที่การทำงานที่มากขึ้นก็จะทำได้ยากมาก เพราะระบบได้ถูกกำหนดมาตฐานหน้าที่พื้นฐานมาแล้วจากโรงงาน การเปลี่ยนแปลงใดๆก็จะทำได้ยาก โดยเฉพาะตัวประมวลผลหลักซึ่งนับเป็นหัวใจระบบที่ใช้ควบคุมการทำงานของระบบคือ Network Video Recorder (NVR) [1] ที่เป็นส่วนสำคัญมากและราคาสูง ดังนั้นผู้วิจัยจึงมีแนวคิด ที่จะสร้างตัวกล้องแบบเครือข่ายนี้ โดยให้สามารถเพิ่มเติมหน้าที่การทำงานได้อย่างอิสระ โดยในตัวอย่างหน้าที่พิเศษที่ทำขึ้นคือการทำเป็นเครื่องติดตามวัตถุโดยใช้สีเป็นเงื่อนไขหลัก เหตุที่สามารถสร้างเป็นระบบต้นแบบได้ก็เพราะว่า ปัจจุบันนี้ อุปกรณ์คอมพิวเตอร์ฝังตัว ต่างมีความสามารถในการเชื่อมต่อเครือข่ายแล้ว และตัวคอมพิวเตอร์เองก็สามารถ สั่งงานเชื่อมต่อกับระบบฮาร์ดแวร์ต่างๆได้เป็นอย่างดี ที่สำคัญราคาถูกลงเป็นอย่างมาก ดังนั้นหากเรานำคอมพิวเตอร์ฝังตัวนี้มาทำการเชื่อมต่อกันกับกล้องขนาดเล็ก แล้วก็ออกแบบเขียนโปรแกรมเพื่อควบคุมการทำงานเอง ก็จะสามารถสร้างระบบกล้องวงจรปิดที่เก็บภาพได้เหมือนกันและยังสามารถเพิ่มหน้าที่พิเศษต่างๆได้ตามต้องการได้อีกต่างหาก ซึ่งจะเกิดประโยชน์มากต่อการนำไปต่อยอดเพื่อใช้งานให้ตรงกับที่หน่วยงานหรือสถานที่ต่างๆนั่นเอง

1.2 วัตถุประสงค์ของโครงการวิจัย

ออกแบบและสร้างระบบกล้องวงจรปิดที่ทำงานได้ด้วยตัวเอง แต่จะสามารถมีการสั่งงานได้จากคอมพิวเตอร์ส่วนบุคคลทั่วไปผ่านระบบเครือข่ายคอมพิวเตอร์ไร้สาย (WIFI) [2] ที่โปรแกรมสำหรับควบคุมการทำงานบนคอมพิวเตอร์นี้ก็จะถูกออกแบบให้เหมาะสมเพื่อใช้งานที่ง่ายที่สุด ซึ่งเครื่องต้นแบบนี้นอกจากจะมีความสามารถในการเก็บภาพได้เหมือนระบบทั่วไปแล้ว ยังสามารถเพิ่มเติมหน้าที่พิเศษได้อีกตามความต้องการของผู้ใช้ โดยในเครื่องตต้นแบบนี้ได้เพิ่มหน้าที่พิเศษคือ หน้าที่ในการตรวจจับวัตถุที่มีสีตามที่กำหนดและแจ้งเตือนออกมาที่เครื่องคอมพิวเตอร์ของผู้ใช้ทันทีที่พบ ซึ่งอันนี้จะเป็นแนวทางที่สามารถนำไปพัฒนาต่อในการตรวจจับวัตถุอื่นๆต่อไปได้

1.3 ขอบเขตของโครงการวิจัย

สร้างระบบบันทึกภาพอัตโนมัติราคาประหยัด จำนวน 1 ชุด
ประกอบไปด้วยดังนี้

1) ส่วนของตัวกล้อง จำนวน 2 เครื่อง
2) ส่วนของโปรแกรมควบคุมการทำงานบนคอมพิวเตอร์ทั่วไป จำนวน 1 ชุด

ให้มีคุณสมบัติเบื้องต้นของแต่ละส่วนเป็นดังนี้

- เก็บภาพได้และเรียกดูไฟล์ภาพได้ผ่านระบบเครือข่าย
- มีโปรแกรมควบคุมการทำงานสั่งงานจากคอมพิวเตอร์ทั่วไปได้ เพื่อการกำหนดการทำงานต่างๆได้โดยผู้ใช้เอง
- การเชื่อมต่อตัวกล้องและคอมพิวเตอร์ของผู้ใช้ สามารถใช้ระบบอินเทอร์เนท (WIFI) ได้

1.4 วิธีดำเนินการวิจัย

ในการวิจัยนี้ ผู้วิจัยจะปฏิบัติการออกแบบและสร้างต้นแบบ ที่ตึก 12 ชั้น ห้อง E12-1110 ซึ่ง เป็นห้องวิจัยของผู้วิจัยเอง หลังจากทำการสร้างต้นแบบได้แล้ว ก็จะนำไปทดสอบกบพื้นที่จริง คือ ทดลองใช้งานที่ตึก 12 ชั้นที่ชั้น 11 (ห้องวิจัยของผู้วิจัยเอง) โดยติดตั้งไว้ที่ทางเดิน และให้ นักศึกษาและอาจารย์ท่านอื่นๆได้ทดลองใช้งานด้วยเช่นกัน เพื่อเก็บข้อมูลไปพัฒนาปรับปรุงต่อไป

1.5 ประโยชน์ที่คาดว่าจะได้รับ

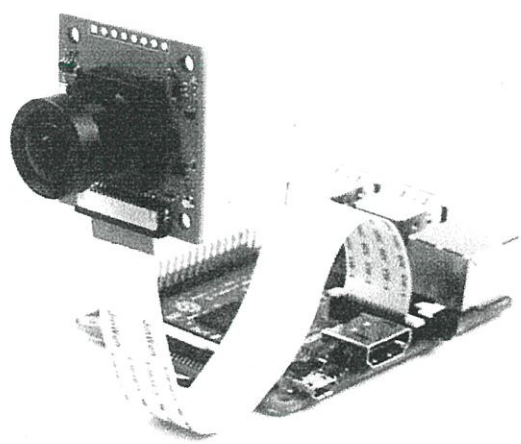
ได้เป็นต้นแบบของ ระบบกล้องวงจรปิดแบบทำงานผ่านเครือข่าย โดยเน้นสร้างเองทุกส่วน ตั้งแต่ตัวกล้อง ที่ใช้คอมพิวเตอร์ฝังตัวเป็นตัวประมวลผลหลัก ทำงานร่วมกันกับโปรแกรมที่ออกแบบสำหรับควบคุมการทำงานของกล้องโดยติดตั้งได้บนคอมพิวเตอร์ทั่วไป ที่สำคัญสามารถเชื่อมต่อระบบอินเทอร์เนทในแบบไร้สายได้ ดังนั้น งานวิจัยนี้จึงน่าจะเป็นประโยชน์ต่อการนำไปใช้งานได้ในหลายรูปแบบ เพราะสามารถจะขยายจำนวนของกล้อง และพัฒนาโปรแกรมควบคุมให้สามารถเพิ่มหน้าที่พิเศษได้เองตามต้องการของหน่วยงานนั้นๆ และด้วยราคาการสร้างที่ไม่แพง และสามารถหาอุปกรณ์ได้ในเมืองไทย ก็จะทำให้ง่ายต่อการซ่อมแซมอีกด้วยเช่นกัน ดังนั้นผลงานนี้ น่าจะเป็นตัวเลือกหนึ่งในการนำไปพัฒนาใช้งานให้ตรงกับความต้องการของแต่ละสถานที่ต่อไป

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ในปัจจุบันนี้ ระบบกล้องวงจรปิดได้มีใช้กันอย่างแพร่หลาย โดยเฉพาะงานด้านการรักษาความปลอดดภัย, งานเฝ้าระวังติดตาม คน, รถยนต์ เหล่านี้เป็นต้น ซึ่งในสถานที่ใหญ่ๆ หรือตึกสูงๆหลายๆชั้น เพราะมีจุดที่จำเป็นต้องเฝ้าระวังหลายตำแหน่ง การมีจำนวนคนเฝ้าไม่พอ ระบบกล้องวงจรปิดจึงเป็นที่นิยมใช้งานมาก และเนื่องจากปัจจุบันได้มีระบบกล้องวงจรปิดที่ใช้การสื่อสารควบคุมได้โดยระบบอินเทอร์เนท และเป็นแบบไร้สาย (WIFI) ทำให้ยิ่งเป็นที่นิยมมากขึ้นเพราะสามารถควบคุมได้จากที่ใดก็ได้ที่สามารถเชื่อมต่ออินเทอร์เนทได้ ในงานวิจัยนี้ก็จะได้ออกแบบระบบกล้องวงจรปิดนี้ให้มีการควบคุมด้วยระบบไร้สายเช่นกันแต่เพิ่มเติมให้มีหน้าที่พิเศษได้ สามารถซ่อมแซมหรือพัฒนาต่อยอดได้ โดยมีทฤษฎีที่เกี่ยวข้องดังจะได้กล่าวต่อไป

2.1 ทฤษฎีที่เกียวข้อง

ในส่วน ทฤษฎีอุปกรณ์ที่เกี่ยวข้องนั้น จะมีอยู่ด้วยกันดังนี้ การออกแบบส่วนของฮาร์ดแวร์จะเป็นการใช้งานตัวประมวลผลแบบคอมพิวเตอร์ฝังตัวโดยผู้วิจัยเลือกใช้ ไมโครคอนโทรลเลอร์ขนาดเล็ก เป็นบอร์ด Rapberry pi2 [3] สาหรับตัวกล้องก็จะใช้ที่ทำมาเฉพาะตัวบอร์ดเลยคือ Picam ซึ่งสามารถกำหนดขนาดของภาพที่ต้องการได้ ตัวกล้องที่ควบคุมโดยหน่วยประมวลผลนี้ทำงานร่วมกับโปรแกรมหลักของผู้ใช้โดยการรับส่งผ่านระบบ WIFI รูปแบบการเชื่อมต่อก็ไม่ยุ่งยากดังแสดงในภาพที่ 2.1 ข้างล่างนี้



ภาพที่ 2.1 เป็นตัวประมวลผลหลัก Raspberry pi2 ติดตั้งกล้องแล้วที่ใช้ในงานวิจัยนี้

ส่วนตัว WIFI ผู้วิจัยเลือกใช้แบบ USB Type ที่สามารถเสียบเข้าช่อง USB ของ Rapberry pi2 ได้เลยซึ่งสะดวกในการติดตั้งมากทั้งฮาร์ดแวร์และซอฟร์แวร์ ดังรูปที่ 2.2

รูปที่ 2.2 แสดงตัวเชื่อมต่อไร้สายแบบ USB Type ที่ใช้ในงานวิจัย

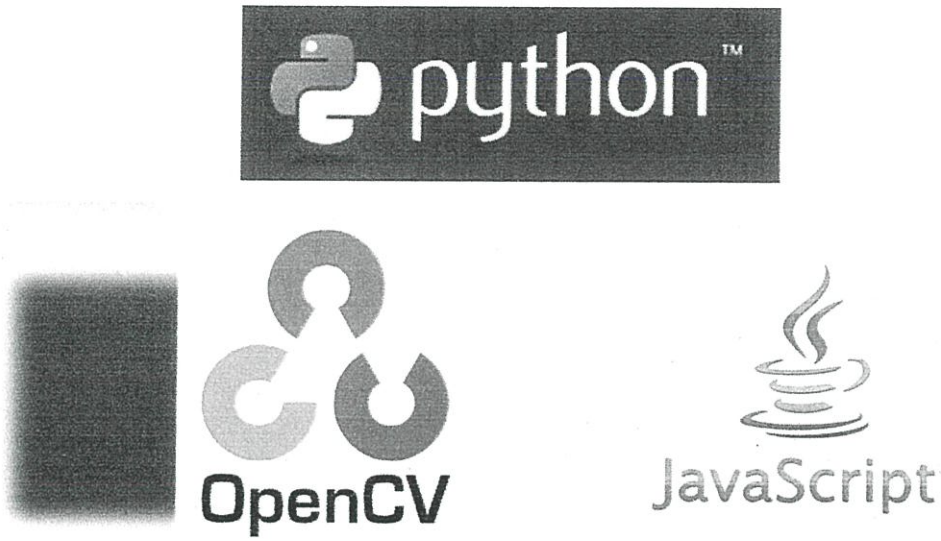
ส่วนในเรื่องหน่วยความจำที่ใช้เก็บภาพของกล้องนั้น เนื่องจากมีแนวคิดในการเก็บภาพแบบภาพนิ่งที่ขนาดภาพ 320x240 ต่อภาพใช้หน่วยความจำประมาณ 15 KB (ใช้การบีบอัดแบบ Jpeg) กำหนดให้มีการเก็บภาพทุก 2 วินาที (ปรับได้) โดยมีการจัดเก็บไว้ที่ ตัวหน่วยความจำแบบ USB (Handy drive) ได้ซึ่งปัจจุบันนี้มีขนาดที่มากถึง 64GB ให้ใช้งาน และตัวโปรแกรมเองก็สามารถกำหนดได้ว่า หากมีจำนวนไฟล์เก่าเกินเวลาเท่าใดหรือจำนวนไฟล์มากเกินกำหนดก็จะมีการบันทึกภาพทับไฟล์ที่เก่าสุดไปเรื่อยๆ ทำให้กล้องสามารถทำงานได้ต่อเนื่องได้ตลอดเวลา

ในส่วนของโปรแกรมควบคุมนั้นผู้วิจัยใช้โปรแกรมอยู่หลายตัวที่ทำหน้าที่ในแต่ละอย่างแบบเหมาะสม โดยตัวโปรแกรมก็เป็นแบบเปิด (Open Source) สามารถนำมาใช้งานได้ทันทีแบบไม่มีค่าใช้จ่าย เช่น JAVA, Python และ OpenCV [4] เป็นต้น ซึ่งหากออกแบบทำงานด้วยกันแล้วก็จะทำให้การทำงานมีประสิทธิภาพและง่ายต่อการพัฒนาปรับปรุง ต่อไป โดยแต่ละโปรแกรมแบ่งการทำงานดังนี้

โปรแกรมภาษา Python ใช้สำหรับการประมวลผลภาพบนตัวคอมพิวเตอร์ฝังตัว (ซึ่งออกแบบให้ใช้ภาษา Python เป็นหลักอยู่แล้ว) โดยโปรแกรมจะมีส่วนของการวิเคราะห์ภาพด้วย โดยการนำ ไลบารี่ ของโปรแกรมแบบเปิด OpenCV (Open Computer Vision) มาใช้งานร่วม

ส่วนของโปรแกรมภาษา JAVA ใช้เขียนใช้งานบนทั้งตัวคอมพิวเตอร์ฝังตัว (กล้อง) และบนตัวคอมพิวเตอร์ส่วนบุคคลที่โปรแกรมเป็นแบบ GUI สำหรับผู้ใช้เพื่อควบคุมการทำงานของกล้องผ่านระบบไร้สาย (WIFI)

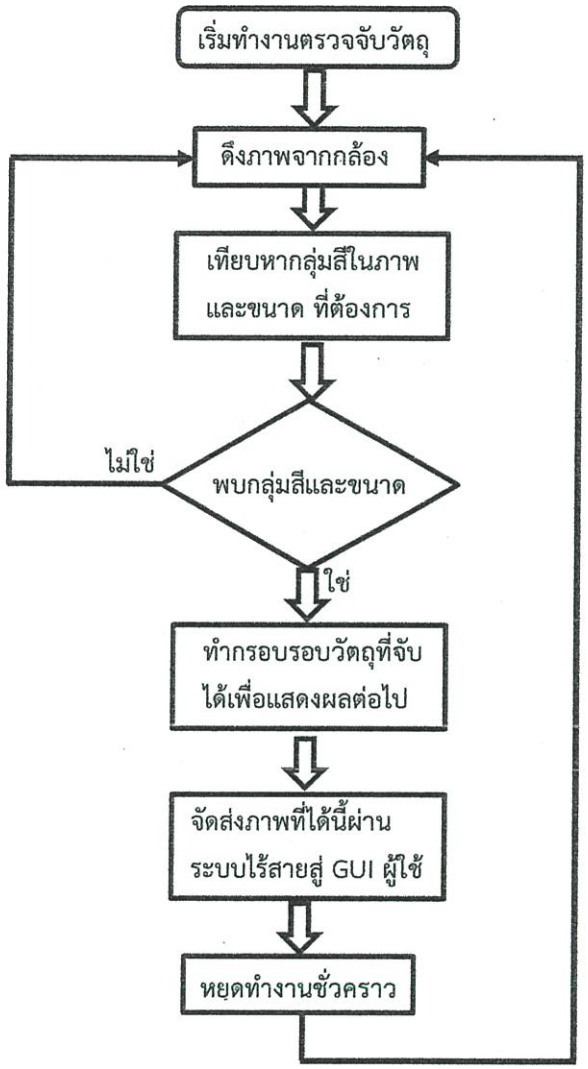
ซึ่งรวมเทคโนโลยี่ด้านโปรแกรมที่นำมาใช้งานมีดังรูปที่ 2.3



ภาพที่ 2.3 เป็นเทคโนโลยีโปรแกรมต่างๆที่นำมาพัฒนาด้านซอร์แวร์

ตัว ไลบารี่ของ OpenCV ที่ทำงานร่วมกับภาษา Python จะใช้เขียนโปรแกรมสำหรับหน้าที่พิเศษคือเป็นการให้กล้องจับภาพวัตถุที่มีสีตรงกับที่ผู้ใช้กำหนดให้จับภาพ (กำหนดสีได้โดยผู้ใช้ผ่านโปรแกรม GUI) โดยตัวโปรแกรมจะมีหลักการทำงานดังแสดงเป็น โฟล์ชาร์ท รูปที่ 2.4 ต่อไปนี้

เริ่มจากการจับภาพจากกล้องวงจรเปิดก่อนโดยเป็นภาพคลื่อนไหวขนาด 320x240 ตัวโปรแกรมจะใช้วิธีเทียบระดับสีหลักคือ RED, GREEN, BLUE โดยตัวสีที่เราต้องการเทียบจะได้มาจากผู้ใช้สั่งงานโดยผ่านระบบ GUI จากทางฝั่งของคอมพิวเตอร์ผู้ใช้นั่นเอง เมื่อพบในภาพมีระดับสีที่ใกล้เคียง (ในทางปฏิบัติจะมีโอกาศน้อยมากที่จะพบระดับสีที่ตรงกับเลย อันมีผลมาจากแสง) และต้องมีขนาดที่มีพื้นที่มากพอด้วย (กำหนดไว้แล้วจากผลการทดลอง) ก็จะบันทึกภาพในแฟรมนั้นไว้ แล้วทำการสร้างกรอบแสดงตำแหน่งของวัตถุที่พบ เพื่อให้ง่ายต่อการสังเกตุด้วยตาผู้ใช้ ส่วนของการจับภาพและหาวัตถุนี้จะเป็นภาษา Python จากนั้นโปรแกรมก็จะใช้ภาษา JAVA [5] ในการส่งภาพนี้ผ่านระบบไร้สาย ไปแสดงภาพนี้บนโปรแกรม GUI บนเครื่อง คอมพิวเตอร์ของผู้ใช้เพื่อให้ผู้ใช้ดูว่าใช่วัตถุที่ต้องการหาหรือไม่ ตัวโปรแกรมของกล้องนี้จะหยุดทำงานชั่วขณะเพื่อให้ผู้ใช้ ตัดสินใจว่าจะหยุดทำงานในโหมดนี้หรือไม่ หากต้องการหาต่อโปรแกรมก็จะวนทำงานเช่นเดิมไปเรื่อยๆ นั่นเอง

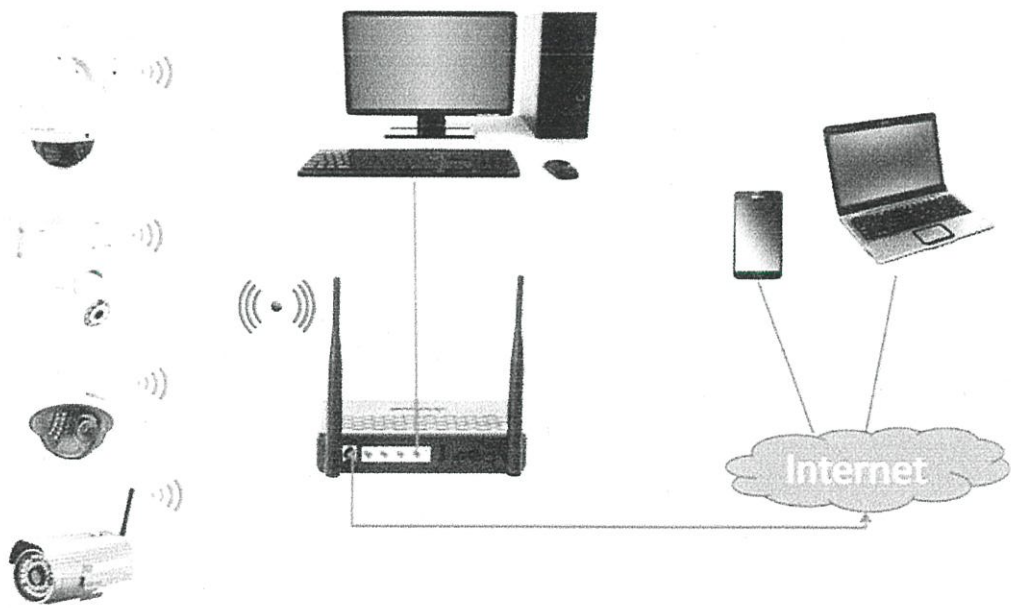


รูปที่ 2.4 บล็อกไดอะแกรมโปแกรมหน้าที่พิเศษ (ตรวจจับวัตถุ) ที่ทดลองใช้

การทำงานร่วมกับโปรแกรมส่วนอื่นๆจะได้กล่าวอีกครั้งอย่างละเอียดในขั้นตอนดำเนินการวิจัย ส่วนตัวโปรแกรมก็จะมีรวมไว้แล้วทั้งหมดในแผ่น CD ROM โดยในส่วนของภาคผนวกจะขอนำบางส่วนเท่านั้นแสดงไว้

บทที่ 3
วิธีดำเนินงานวิจัย

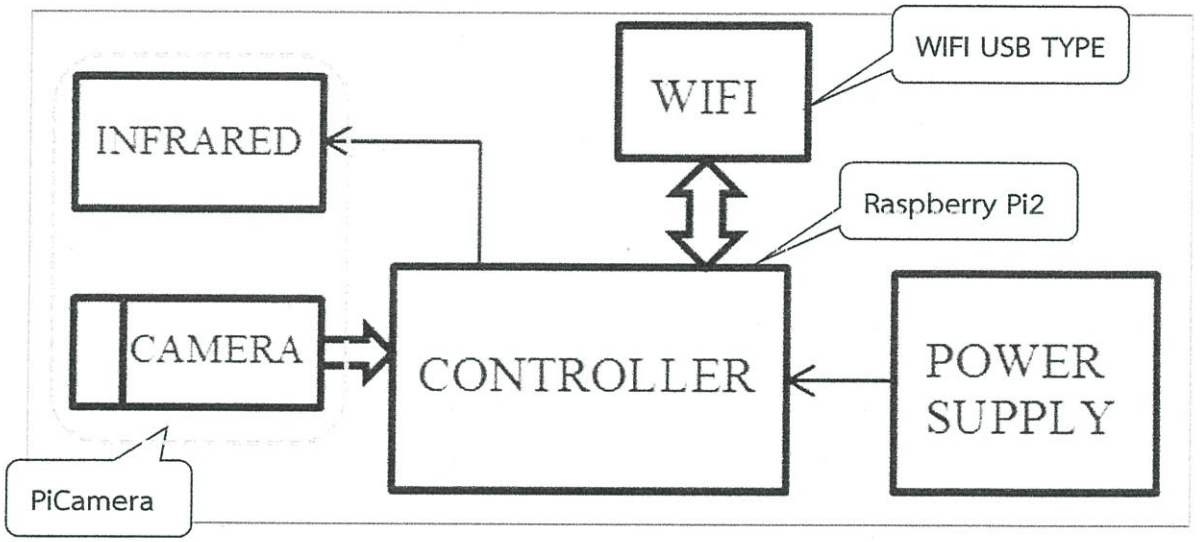
จากภาพรวมการใช้งานของระบบบันทึกภาพอัตโนมัติราคาประหยัด รูปที่ 3.1 ระบบก็จะประกอบไปด้วยตัวกล้องเองซึ่งมีคอมพิวเตอร์ฝังตัวทำงานเป็นแกนหลักภายในกล้องมีการติดต่อรับคำสั่งจากผู้ใช้โดยผ่านระบบไร้สาย (WIFI) โดยปกติตัวกล้องเมื่อถูกสั่งงานแล้วก็จะทำงานนั้นด้วยตัวเองเลย เช่นการสั่งให้กล้องทำงานตรวจจับวัตถุในสีที่กำหนด ต้วกล้องเมื่อได้รับคำสั่งก็จะทำตามและตัดสินใจเองเลย ดังนั้นตัวโปรแกรมควบคุมฝังผู้ใช้หรือคอมพิวเตอร์ของผู้ใช้ก็จะไม่ต้องประมวผลใดๆเลย ทั้งนี้เพราะในระบบไร้สายจริงๆนั้น การส่งข้อมูลไปมาบ่อยๆ จะทำให้การทำงานช้าได้ โดยเฉพาะในระบบไร้สายที่มีอัตรารับส่งต่ำๆ ระบบจึงถูกออกแบบมาให้ทำงานเองแบบอัตมัตินั่นเอง จากรูปโปรแกรมแบบ GUI จะถูกติดตั้งที่เครื่องของผู้ใช้ ในบทนี้ การดำเนินการวิจัยจะได้พูดถึงการสร้างในแต่ละส่วนควบคู่ไปกับการทำงานในส่วนนั้นๆและแสดงรูปภาพของส่วนที่สร้างนั้นพร้อมกันไป



ภาพที่ 3.1 แนวคิดของการใช้งานของเครื่องบันทึกภาพอัตโนมัติ

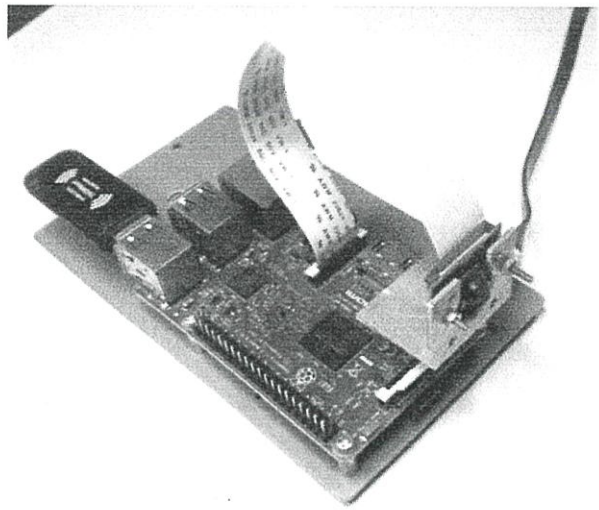
3.1 ส่วนของ Hardware

ทั้งหมดของระบบก็จะมีส่วนที่เป็นฮาร์ดแวร์ก็คือตัวกล้องเท่านั้น ดังนั้นส่วนฮาร์ดแวร์นี้ก็จะเป็นการออกแบบสร้างตัวกล้อง โดยบล็อกไดอะแกรมของตัวกล้องก็จะมีส่วนประกอบ ดังแสดงในภาพที่ 3.2

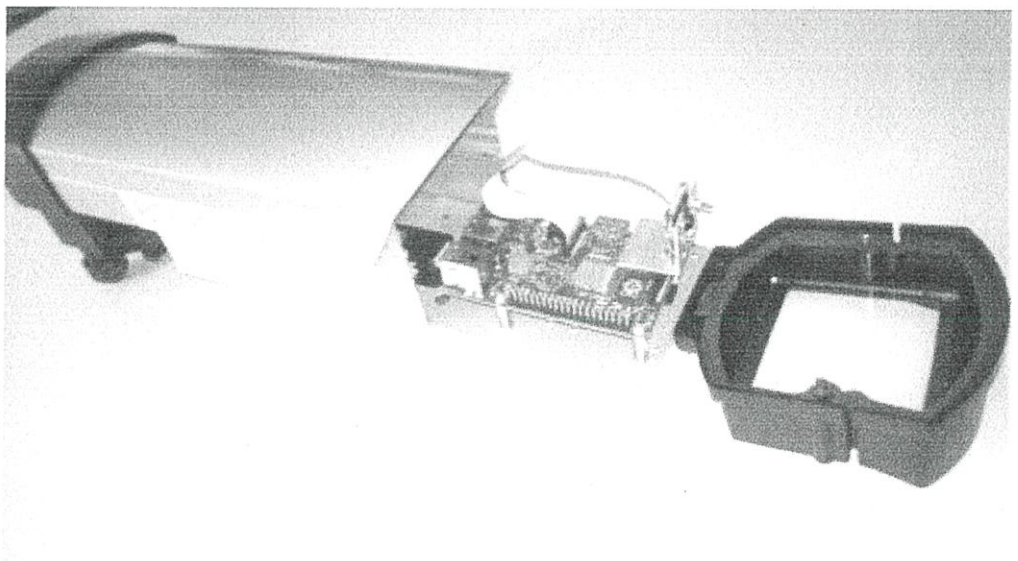


ภาพที่ 3.2 แสดงส่วนประกอบของตัวกล้อง

โดยส่วนของตัวประมวลผลก็ดังที่กล่าวไว้แล้วว่าใช้คอมพิวเตอร์ฝังตัว Raspberry Pi2 ซึ่งสามารถต่อร่วมกันกับกล้อง PiCamera ได้ทันที (สร้างมาใช้งานด้วยโดยเฉพาะกับ Raspberry Pi2) และตัวกล้องเองก็มีแบบที่รวมระบบอินฟาเรดไว้ในตัวแล้วด้วยเช่นกัน และอีกตัวคือส่วนของ WIFI ก็เป็นแบบ USB TPYE ซึ่งขนาดเล็กและง่ายต่อการติดตั้งใช้งาน สุดท้ายคือส่วนของจ่ายกำลังไฟ อันนี้ตัวบอร์ดต้องการไฟเลี้ยงแบบไฟฟ้ากระแสตรง เพียง 5 โวล์ จึงสามารถใช้เครื่องแปลงไฟขนาดเล็กได้ทั่วไป เนื่องจากตัวกล้องต้องติดตั้งไว้ที่สูงและมักจะอยู่นอกอาคาร ดังนั้นตัวบอร์ดทั้งหมดจึงต้องมีกล่องคลุมอีกครั้ง ดังนั้นตัวกล้องที่นำเสนอในงานวิจัยนี้จึงเป็นดังรูปที่ 3.3 และ รูปที่ 3.4 ตามลำดับ



ภาพที่ 3.3 รูปบอร์ด Raspberry Pi2 และส่วนประกอบเชื่อมต่ออื่นๆ



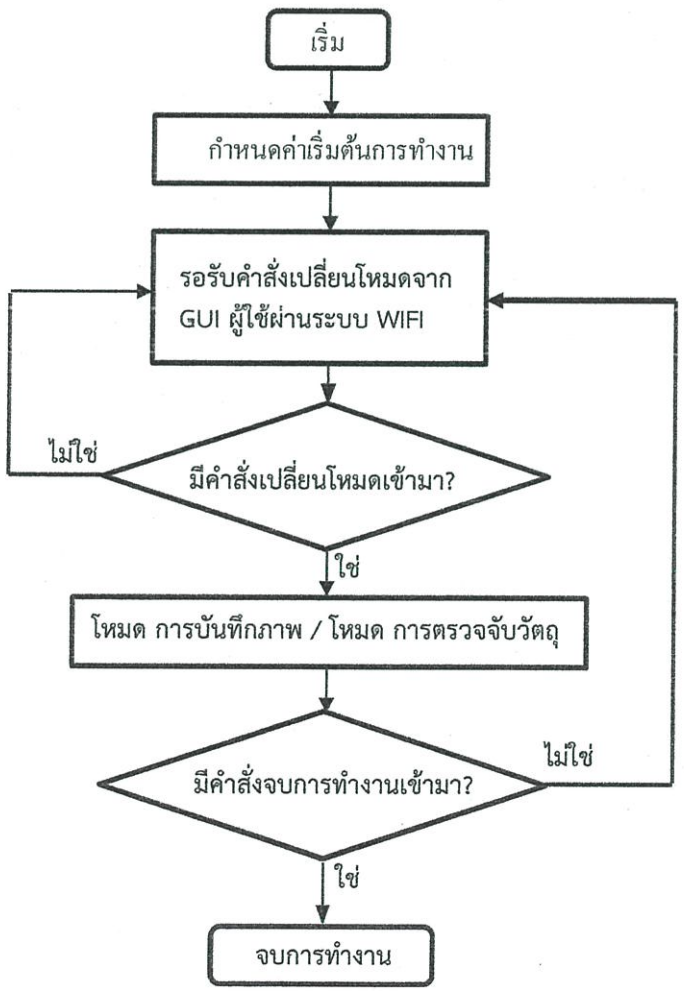
ภาพที่ 3.4 รูปบอร์ด Raspberry Pi2 ประกอบใส่กล่อง

จะเห็นได้ว่าส่วนของ ฮาร์ดแวร์นั้น ไม่มีอะไรที่ยุ่งยากนัก เพราะอุปกรณ์ทุกตัวเป็นแบบสำเร็จแล้วและสามารถ

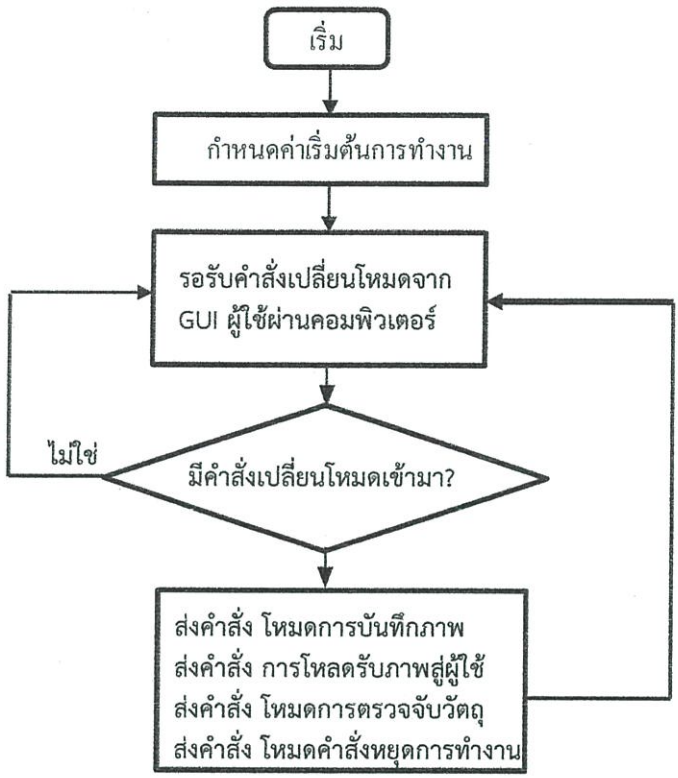
หาได้ในเมืองไทยได้ไม่ยากนัก ส่วนที่สำคัญจึงจะเป็นส่วนของซอฟร์แวร์ ซึ่งมีทั้งส่วนที่ใช้งานบนตัวกล้องเอง และส่วนของตัวโปรแกรมแบบ GUI ที่ต้องติดตั้งที่ตัวคอมพิวเตอร์ของผู้ใช้ ซึ่งทั้งหมดนี้จะได้กล่าวต่อไปในหัวข้อ ส่วนของ Software

3.2 ส่วนของ Software

ส่วนนี้นับว่าสำคัญมาก ดังที่ได้กล่าวมากแล้วว่าในส่วนฮาร์ดแวร์นั้น อุปกรณ์ส่วนใหญ่เป็นแบบสำเร็จรูป ซึ่งปัจจุบันมีจำหน่ายอยู่แล้ว เพียงนำมาประกอบกันเป็นระบบให้เหมาะสม มีการปรับแต่งบ้างเล็กน้อย ก็สามารถได้ฮาร์ดแวร์ที่ต้องการได้โดยไม่ยากนัก แต่ส่วนที่จะต้องพิจารณาให้ดีก็คือ ตัวของโปรแกรมหรือซอฟร์แวร์ ซึ่งต้องควบคุมการทำงานทุกอย่างให้เป็นไปอย่างลำดับที่ถูกต้อง ซึ่งในการทำงานโดยภาพรวมจะแบ่งเป็นสองโหมดการทำงานคือ โหมดการบันทึกภาพปกติ และโหมดการตรวจจับวัตถุด้วยสี โดยไออะแกรมซอฟร์แวร์ของตัวกล้องและตัวโปรแกรมควบคุมบนคอมพิวเตอร์ผู้ใช้ แสดงดังรูปที่ 3.5



ภาพที่ 3.5 โฟร์ชาร์ท โปรแกรมของส่วนตัวกล้อง



ภาพที่ 3.6 โฟร์ชาร์ท โปรแกรมของส่วนโปรแรกมควบคุม (GUI) ของผู้ใช้

ซึ่งจะเห็นได้ว่าการออกแบบส่วนของซอฟร์แวร์นี้นั้น ในภาพที่ 3.5, 3.6 จะเป็นแบบแนวคิด ตรงไฟตรงมา ไม่ซับซ้อนนัก โดยรูปที่ 3.5 เป็นโปรแกรมของตัวกล้องซึ่ง ในหน้าที่หลักคือต้องรอรับคำสั่งการทำงานที่ส่งมาจากฝั่งผู้ใช้ด้วยโปรแกรม GUI ผ่านระบบอินเตอร์เนทไร้สาย (WIFI) แล้วก็ต้องทำงานตามนั้น ในตอนแรกตัวกล้องจะกำหนดค่าพื้นฐานของตัวเองก่อน เมื่อมีคำสั่งก็จะทำงานตามนั้นดังนี้

ในโหมดการบันทึกภาพ ตัวกล้องจะบันทึกภาพจากโมดูลกล้อง ก็จะทำการบันทึกภาพ (ภาพนิ่ง) ในทุก 2 วินาที เก็บไว้ในหน่วยความจำ ไปเรื่อย โดยในแต่ละภาพก็จะมีการบันทึก วัน, เวลา เป็นชื่อของไฟล์ไว้ ทำให้ตอนที่เราจะโหลดภาพไปตรวจสอบโดยผู้ใช้ก็จะง่ายในการหาภาพที่ต้องการนั้นเอง ในการเก็บภาพนั้นจะมีการเก็บภาพสะสมไปเรื่อยๆ จึงต้องมีการตรวจสอบเสมอว่าจำนวนไฟล์ที่ต้องการเก็บมีค่ามากเท่าไหร่ หากถึงกำหนดโปรแกรมก็จะมีการบันทึกทับภาพที่เก่าสุดเพื่อวนบันทึกไปได้เรื่อยๆ ซึ่งสามารถกำหนดได้ว่าจะให้มีจำนวนภาพก่อนที่จะวนเขียนภาพทับเป็นจำนวนเท่านได ในโหมดนี้หากผู้ใช้ต้องการโหลดภาพที่บันทึกเก็บไว้ก็จะหยุดการบันทึกแล้วให้ผู้ใช้สามารถเข้าถึงหน่วยความจำที่เก็บภาพ

เพื่อโหลดดภาพได้ มีโปรแกรมสำหรับโหลดภาพ อีกที

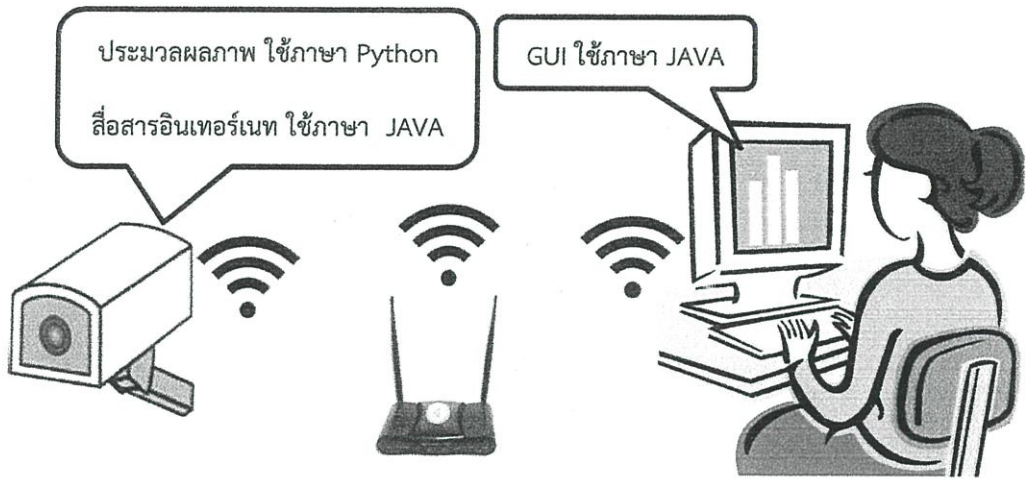
ในโหมดของการตรวจจับวัตถุ อันนี้ในงานวิจัยจะใช้เป็นค่าของสีวัตถุ ในการตรวจจับ โดยจะมีการรับภาพมาวิเคราะห์ในแต่ละเฟรม ว่าพบวัตถุในช่วงของสีที่ต้องการหรือไม่และต้องมีขนาดที่ใหญ่ตามต้องการด้วย เหตุที่เลือกเป็นช่วงก็เพราะว่า สีของวัตถุที่ถ่ายภาพได้อาจมีความผิดเพี้ยนไปจากเป็นจริงหรือสีที่เราต้องการก็ได้เนื่องจากลักษณะของแสง ณ.เวลาที่ถ่ายภาพนั้นแตกต่างกันไป จึงกำหนดเป็นช่วง โดยใช้ช่วงของ ระดับสี แดง, น้ำเงิน และเขียว (RED, BLUE, GREEN) ให้มีช่วงเบี่ยงเบน ค่าหนึ่งที่ได้จากการทดลองจริง เมื่อพบวัตถุดังกล่าวในภาพจากกล้องแล้วก็จะทำการบันทึกภาพนั้นไว้พร้อมกับทำกรอบล้อมรอบกลุ่มวัตถุนั้นไว้ ทั้งนี้เพราะภาพนี้จะถูกส่งผ่านเครือข่ายไร้สายไปสู่หน้าจอ GUI ของผู้ใช้ ซึ่งในการใช้งานโหมดนี้ ตัวกล้องจะทำหน้าที่เพียงส่งภาพวัตถุที่พบพร้อมทำกรอบล้อมรอบไว้ให้ ตัวผู้ใช้เองจะต้องดูที่ภาพว่าใช่วัตถุที่ต้องการหรือไม่ หากใช่ก็จะสามารถทราบตำแหน่งที่อยู่เพราะจะทราบว่ามาจากกล้องตัวใด หากไม่ใช่กล้องกจะส่งภาพที่พบมาให้ดูอีก เป็นเช่นนี้ไปเรื่อยๆ จนกว่าจะมีการเปลี่ยนโหมดการทำงาน

ในรูปที่ 3.6 ก็จะเป็นโฟล์ชาร์ทของโปรแกรมฝั่งผู้ใช้ที่เป็นแบบ GUI ซึ่งส่วนนี้จะถูกเขียนด้วยโปรแกรม ภาษา JAVA โดยจะมีปุ่มให้กดใช้งานที่ง่ายสามารถเข้าใจและใช้ได้อย่างรวดเร็ว การทำงานเริ่มจากตัวโปรแกรมจะรอรับคำสั่งจากผู้ใช้ว่าจะต้องการทำหน้าที่ใดกับกล้องตัวใด (ในตัวอย่างงานวิจัยนี้จะมี 2 ตัว) เมื่อผู้ใช้กดปุ่มเลือกการทำงาน เช่น เลือกโหมดการบันทึกภาพ ตัวโปรแกรมก็จะส่งคำสั่งนี้ไปสู่ตัวกล้องโดยผ่านระบบเครือข่ายไร้สาย หากคำสั่งใดเป็นการโหลดข้อมูลภาพจากกล้องก็จะมีคำสั่งไปเปิดโปรแกรมอื่นให้ทำงานต่อไป โดยมีโปรแกรมสำเร็จรูปที่นำมาใช้ด้วยคือ

โปรแกรม WinSCP ใช้สำหรับการโหลดรูปภาพจากตัวกล้องมาสู่เครื่องผู้ใช้โดยผ่านโปโตคอลแบบ FTP (File Transfer Protocal)

โปรแกรม VLCmedia ใช้สำหรับการดูภาพแบบเวลาจริงจากกล้องต่างๆมาแสดงบน GUI ของผู้ใช้ ส่วนนี้จะใช้ตอนติดตั้งกล้องครั้งแรกเท่านั้น เพราะจะต้องมีการตั้งโฟกัสดภาพที่เหมาะสมในแต่ละสถานที่

ดังที่ได้กล่าวมาข้างต้นว่าในงานวิจัยได้มีการใช้งานภาษาโปรแกรมอยู่ 2 ภาษา คือ Python, JAVA และใช้ไลบารี่ OpenCV สำหรับการเขียนโปรแกรมประมวลผลภาพบน Python ทั้งนี้เพราะแต่ละภาษาก็จะมีจุดเด่นไม่เหมือนกัน ตัวภาษา Python เป็นภาษาที่นับว่าเป็นตัวหลักของการเขียนโปรแกรมบนตัว Raspberry pi2 มีตัวอย่างที่พัฒนาด้วยภาษานี้มากมายสำหรับรันบนตัว Raspberry pi2 อีกทั้งยังสามารถทำงานร่วมกับภาษาอื่นๆได้อย่างไม่ซับซ้อน ผู้วิจัยจึงนำโปรแกรม Python นี้เขียนโปรแกรมประมวลผลภาพโดยใช้ไลบารี่ OpenCV ซึ่งก็มีให้ทำงานร่วมกับ Python ด้วยเช่นกัน ส่วนตัวโปรแกรม JAVA ผู้วิจัยใช้เพื่อพัฒนาตัว GUI สำหรับผู้ใช้บนคอมพิวเตอร์ทั่วไป เพราะทำงานด้านกราฟฟิกได้ดีและที่สำคัญมีคลาสมากมายในการติดต่อข้อมูลผ่านระบบเครือข่ายคอมพิวเตอร์ ดังนั้นตัวภาษา JAVA จึงใช้สำหรับการแสดงผล GUI และการติดต่อสื่อสารที่ติดตั้งไว้ทั้งบนคอมพิวเตอร์ผู้ใช้และฝั่งด้านกล้องวงจรปิดก็ติดตั้งไว้เช่นเดียวกัน รายละเอียดตัวโปรแกรมนั้น จะอยู่ในรูปแบบ CD-ROM ซึ่งจะรวมทุกส่วนไว้แล้วทั้งฮาร์ดแวร์และซอฟร์แวร์ ส่วนที่อยู่ในภาคผนวกจะแสดงบางส่วนที่สำคัญเพราะโปรแกรมยาวมาก รูปแสดงการใช้งานตัวโปรแกรมภาษาต่างๆ ในแต่ละส่วนแสดงดังรูปที่ 3.7



รูปที่ 3.7 แสดงการใช้ภาษาโปรแกรมแต่ละส่วน

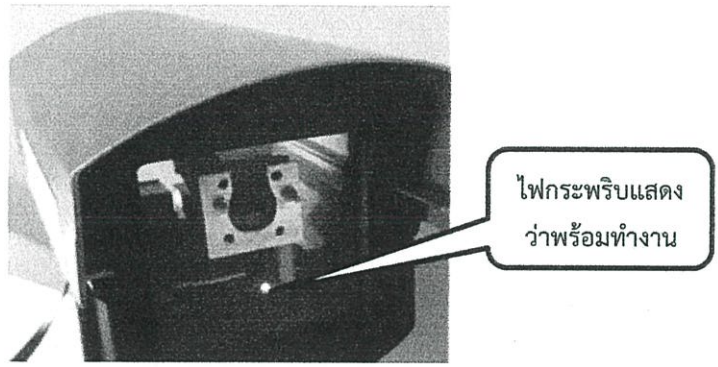
ในส่วนของแผนการทำงานวิจัยในแต่ละเดือนที่ได้ทำมาก็แสดงได้ดังรูปในตารางที่ 3.1 ข้างล่างนี้

การดำเนินงาน	ระยะเวลา												หมายเหตุ
	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	
ศึกษาส่วนของโมดูลต่างๆที่จะต้องใช้งาน													
ออกแบบส่วนของ Hard & Software													
ทดสอบแต่ละส่วน													
จัดทำเอกสาร													

ตารางที่ 3.1 แสดงตารางเวลาการดำเนินการวิจัย

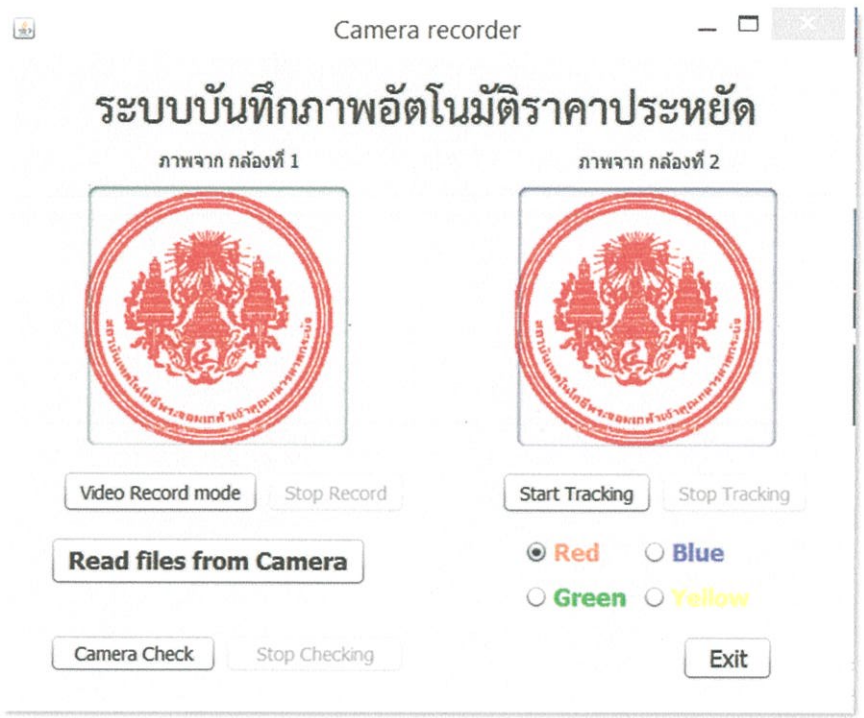
3.3 การใช้งานของระบบบันทึกภาพอัตโนมัติราคาประหยัด

ในส่วนนี้จะได้กล่าวถึงการใช้งานพร้อมรูปภาพประกอบและผลต่างๆ ซึ่งไม่ยุ่งยากเลย เริ่มจากในส่วนของตัวกล้องเนื่องจากว่าได้ทำเสร็จแล้วก็เพียงแต่หลังจากติดตั้งในตำแหน่งที่ต้องการแล้ว ก็เพียงเสียบปลั๊กไฟของแหล่งจ่ายขนาด 5 โวล์ท แล้วรอสักครู่ให้เครื่องพร้อมสำหรับการทำงานก่อน โดยสังเกตุไฟกระพริบที่ตัวกล้องด้านหน้า ดังรูป 3.8 โดยตัวเครื่องจะติดต่อ WIFI เอง (ตั้งไว้ก่อนแล้ว) และคอมพิวเตอร์ที่ใช้ควบคุมก็ต้องอยู่ในวงเครือข่ายเดียวกันด้วย อนึ่งหากต้องการสั่งงานมาจากนอกเครือข่ายก็ต้องมีการกำหนดการเข้าถึงเครือข่ายก่อนที่ตัว Router WIFI ที่เราใช้งาน อันนี้สามารถทำได้ไม่ยากแต่จะไม่กล่าวถึงในที่นี้



รูปที่ 3.8 แสดงตำแหน่งไฟกระพริบเป็นระยะแสดงว่ากล้องพร้อมใช้งาน

หลังจากนั้นก็ทำการเปิดโปรแกรมที่ตัวเครื่องคอมพิวเตอร์ผู้ใช้แล้วรันโปรแกรมที่ใช้ควบคุมก็จะได้หน้าจอ GUI ดังแสดงในรูปที่ 3.9 โดยมีปุ่มต่างๆที่มีความหมายตรงๆอยู่แล้ว

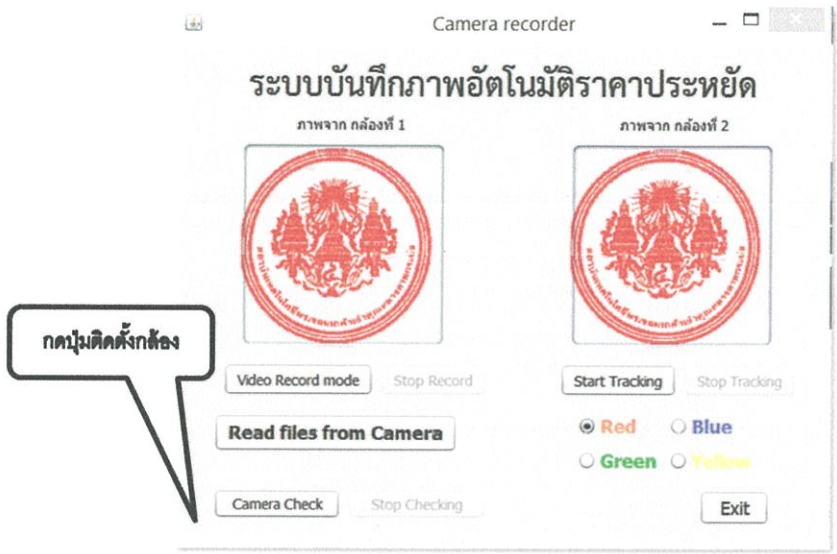


รูปที่ 3.9 แสดงหน้าจอ GUI ที่ออกแบบให้ผู้ใช้สั่งงานบนคอมพิวเตอร์ผู้ใช้

3.3.1 การปรับเมื่อติดตั้งกล้องครั้งแรก

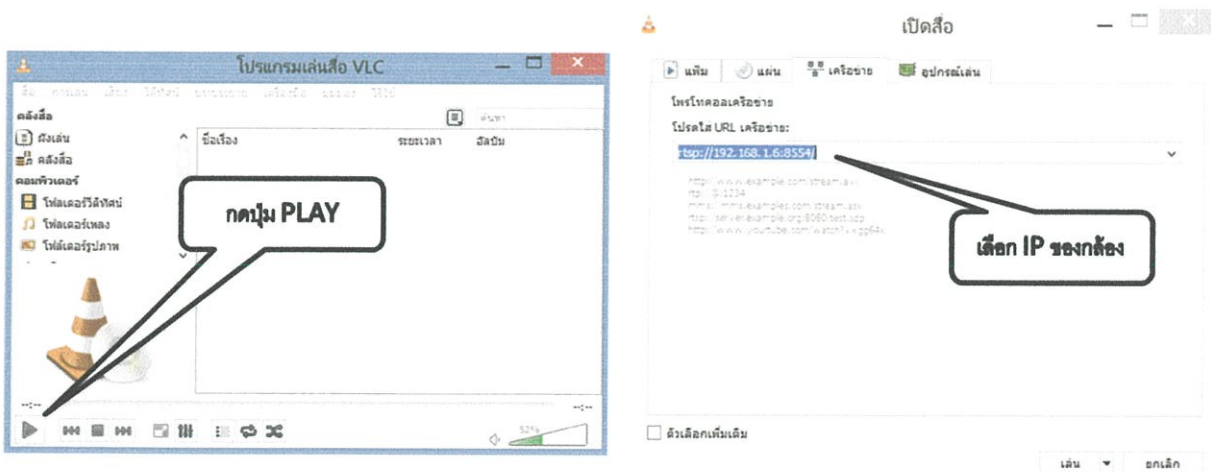
เมื่อเริ่มติดตั้งกล้องในตำแหน่งที่เราต้องการ ในครั้งแรกเราจะต้องทำการปรับให้กล้องมีมุมมองที่ได้ภาพชัดที่สุด เพื่อให้ได้ภาพที่ง่ายต่อการประมวลผล โดยลำดับขั้นดังนี้

ขั้นตอนที่ 1 เมื่อติดตั้งกล้องไว้แล้วก็ทำการเปิดไฟเข้าสู่ตัวกล้องและเปิดโปรแกรมควบคุม GUI บนตัวคอมพิวเตอร์ของผู้ใช้ ที่เมนู GUI ให้เรากดปุ่ม Camera check ดังรูป



รูปที่ 3.10 แสดงตำแหน่งปุ่ม Camera Check บน GUI

เราก็จะได้หน้าต่างใหม่โผล่ออกมาดังรูปที่ 3.11 เพื่อให้เราเลือกหมายเลข IP ของกล้องที่เราจะดูภาพแบบเวลาจริงเพื่อปรับความคมชัด เลือกกดปุ่ม PLAY จะได้เมนูอีกครั้งเลือกที่ เครือข่าย ก็จะได้เลือก IP ที่ต้องการ



รูปที่ 3.11 แสดงการขั้นตอนการเลือก IP ที่ต้องการ

ขั้นตอนที่ 2 จะปรากฏภาพของกล้องที่เราเลือกขึ้นมา (จอใหม่) ดังรูปตัวอย่างที่ รูปที่ 3.12 ให้ทำการปรับตำแหน่งตัวกล้องให้ได้ภาพที่ดีที่สุดตามต้องการ เมื่อได้ภาพตามต้องการแล้ว จากนั้นก็ทำการปิดการใช้งานนี้โดยการกดปุ่ม Stop checking ที่เมนูหลัก GUI เป็นอันเสร็จสิ้น

รูปที่ 3.12 แสดงภาพที่ได้จากกล้องที่เราต้องการปรับตำแหน่งให้ถูกต้อง

3.3.2 โหมดการบันทึกภาพ

ในโหมดนี้เราต้องการสั่งให้กล้องทำการบันทึกภาพนิ่งลงในหน่วยความจำไปเรื่อยๆ ทำได้โดยการ กดที่ปุ่ม Video record mode ซึ่งตัวกล้องทุกตัวก็จะทำการเปลี่ยนโหมดตัวเองให้เป็นการบันทึกภาพนิ่งลงในหน่วยคาจำของตนเองทุก 2 วินาที (ตั้งได้) จะไม่มีภาพมาปรากฏที่หน้าจอของ GUI เราทราบได้ว่ากล้องทุกตัวอยู่ในโหมดนี้เพราะจะเปลี่ยนโหมดไปเป็นอย่างอื่นก็ต้องกด ปุ่ม Stop Record ก่อนเท่านั้น ปุ่มอื่นจะไม่สามารถทำงานได้นั่นเอง ในการทำงานโหมดนี้เราจะมีการเก็บภาพทุกๆ 2 วินาที และเก็บไปเรื่อยๆจนกว่าผู้ใช้จะสั่งหยุดหรือเปลี่ยนโหมดไปเป็นอย่างอื่น และไฟล์ภาพก็จะถูกเก็บไว้ในพื้นที่ที่กำหนด และได้มีการกำหนดไว้แล้วว่าหากไฟล์เกิน 500 รูปก็ให้ทำการบันทึกภาพทับไฟล์ที่เก่าสุดได้ วนไปเรื่อยๆ ดังนั้นโปรแกรมบันทึกก็สามารถทำงานได้ตลอดเพียงแต่ภาพที่เก่ามากๆก็จะถูกทับไป (ซึ่ง 500 ภาพ เป็นตัวอย่างเท่านั้น สามารถเปลี่ยนแปลงได้ขึ้นอยู่กับหน่วยความจำที่เราใช้นั่นเอง) รูปแสดงตัวอย่างภาพที่เก็บไว้ในหน่วยความจำของกล้องแสดงดังรูปที่ 3.13 โดยสังเกตุว่าชื่อไฟล์จะมีการบอกวันเวลาไว้ด้วยดังนี้

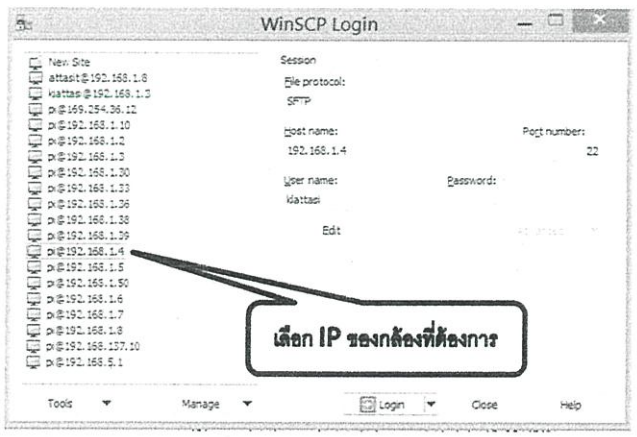
```
pi@raspberrypi:~/Pictures$ ls
2016-Aug-13 14:50:02.jpg  2016-Aug-13 14:50:21.jpg
2016-Aug-13 14:50:05.jpg  2016-Aug-13 14:50:24.jpg
2016-Aug-13 14:50:08.jpg  2016-Aug-13 14:50:27.jpg
2016-Aug-13 14:50:11.jpg  2016-Aug-13 14:50:30.jpg
2016-Aug-13 14:50:15.jpg  2016-Aug-13 14:50:33.jpg
2016-Aug-13 14:50:18.jpg  2016-Aug-13 14:50:36.jpg
pi@raspberrypi:~/Pictures$
```

รูปที่ 3.13 แสดงตัวอย่างชื่อไฟล์ภาพที่เก็บไว้ในหน่วยความจำ

ในการโหลดภาพมา ตรวจสอบนั้น เราใช้โปรแกรม WinSCP ซึ่งเป็นโปรแกรมสำเร็จรูปทำงานแบบ FTP มาใช้งานร่วมกันกับโปรแกรมที่ออกแบบทำให้สามารถโหลดภาพได้รวดเร็ว และที่สำคัญ ใช้งานง่าย

3.3.2.1 การโหลดภาพมาตรวจสอบ

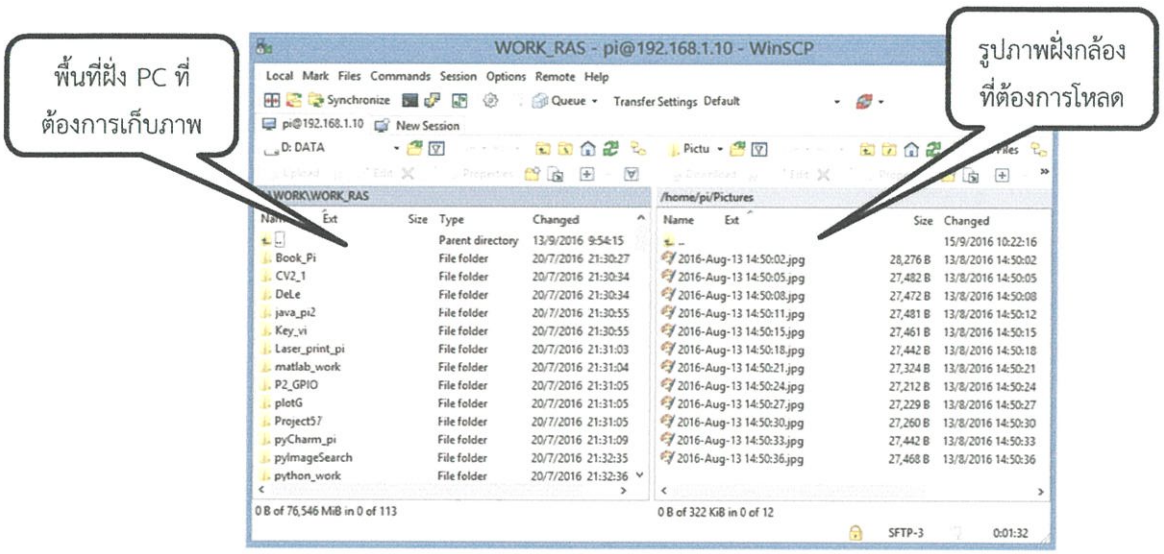
ในการทำงานโหมดการบันทึกภาพนี้จะมีการเก็บภาพไว้ที่หน่วยความจำที่กำหนดตลอดเวลา แต่หากเราต้องการโหลดภาพมาเก็บไว้ที่เครื่องเมื่อไหร่ก็สามารถทำได้โดยกดปุ่ม Stop record ซึ่งหากอยู่ในโหมดบันทึกภาพอยู่ ก็จะมีปุ่มนี้ให้กดอยู่แล้วปุ่ม Read file from camera จะไม่สามารถใช้ได้ เมื่อเรากดปุ่ม Stop record แล้วเราก็สามารถกดปุ่ม Read file from camera ได้ โดยจะมีหน้าต่างใหม่โผล่ออกมาให้เราสามารถเลือกได้ว่าจะไปโหลดภาพจากกล้องหมายเลข IP อะไร ดังรูปที่ 3.14 ทำการเลือกหมายเลข IP กล้องที่ต้องการ



รูปที่ 3.14 หน้าต่างใหม่เพื่อให้เลือกโหลดภาพจากกล้องหมายเลข IP เท่าใด

เมื่อเลือกหมายเลข IP ของกล้องที่ต้องการแล้วก็จะให้เราใส่ Password ซึ่งก็คือ raspberry โปรแกรมก็จะนำ

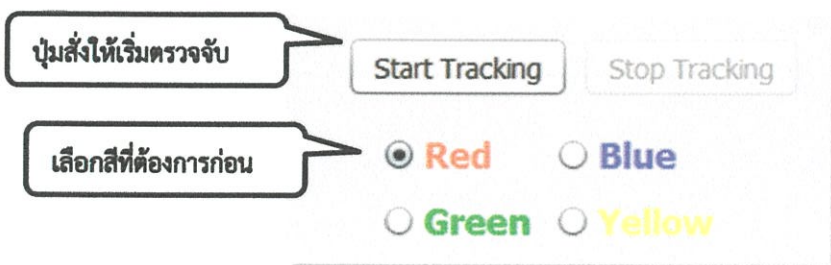
เราไปสู่พื้นที่ Home directory เราก็สามารถไปโหลดภาพในโฟดเดอร์ที่เก็บภาพได้ตามจำนวนที่ต้องการ ดังตัวอย่างแสดงใน รูปที่ 3.15 ข้างล่างนี้



รูปที่ 3.15 โหลดภาพในที่เก็บภาพของกล้องตามต้องการ

3.3.3 โหมดการตรวจจับวัตถุ

ในโหมดนี้ผู้ใช้สามารถสั่งงานได้โดยการกดปุ่ม เลือกสี่ที่ต้องการติดตามตรวจจับก่อนโดยในเบื้องต้นมีให้เลือกอยู่ 4 สีเท่านั้นคือ RED, GREEN , BLUE และ YELLOW (ซึ่งสามารถแก้ไขเพิ่มเติมได้) หากเราไม่เลือกสีก่อนก็จะไม่มีเมนู Start Tracking ให้เราได้สั่งงาน เมื่อเราเลือกสีแล้วก็ให้กดปุ่ม Start Tracking กล้องทุกตัวก็จะทำการเปลี่ยนโหมดไปอยู่โหมดการตรวจจับวัตถุทันที



รูปที่ 3.16 แสดงส่วนปุ่มของการเลือกสีก่อนแล้วก็เลือกเริ่มการตรวจจับ

เมื่อกล้องทำงานในโหมดนี้ก็จะทำการตรวจจับไปเรื่อยๆ หากกล้องตัวใดพบวัตถุที่มีทั้งสีและขนาดที่กำหนดตามต้องการก็จะนำภาพนั้นมาแสดงออกที่จอภาพบน GUI ทันที ผู้ใช้ก็จะพิจารณาว่าเป็นวัตถุที่ต้องการหรือเปล่า โดยตัวโปรแกรมจะรอให้ผู้ใช้พิจารณาประมาณ 10 วินาที จากนั้นก็จะทำการตรวจวัดต่อไป หากพบอีกก็จะส่งภาพมาแสดงอีก เป็นเช่นนี้ไปเรื่อยๆ โดยผู้ใช้ไม่ต้องทำอะไรเลยเพียงนั่งดูหากมีการตรวจพบเท่านั้น และหากพบว่าเป็นวัตถุที่ต้องการและต้องการหยุดภาพไว้หรือหยุดการทำงานก็ทำได้โดยการกดปุ่ม Strop Tracking ได้ทันที รูปที่ 3.17 เป็นรูปภาพทดสอบที่ตรวจสอบวัตถุสีแดง และภาพนี้ก็จะได้ถูกนำมาแสดงที่จอภาพ GUI ของกล้องหมายเลข 1 จะเห็นว่ามีการเขียนกรอบล้อมรอบวัตถุที่ตรวจพบไว้ด้วยเพื่อความชัดเจนในการตรวจสอบ

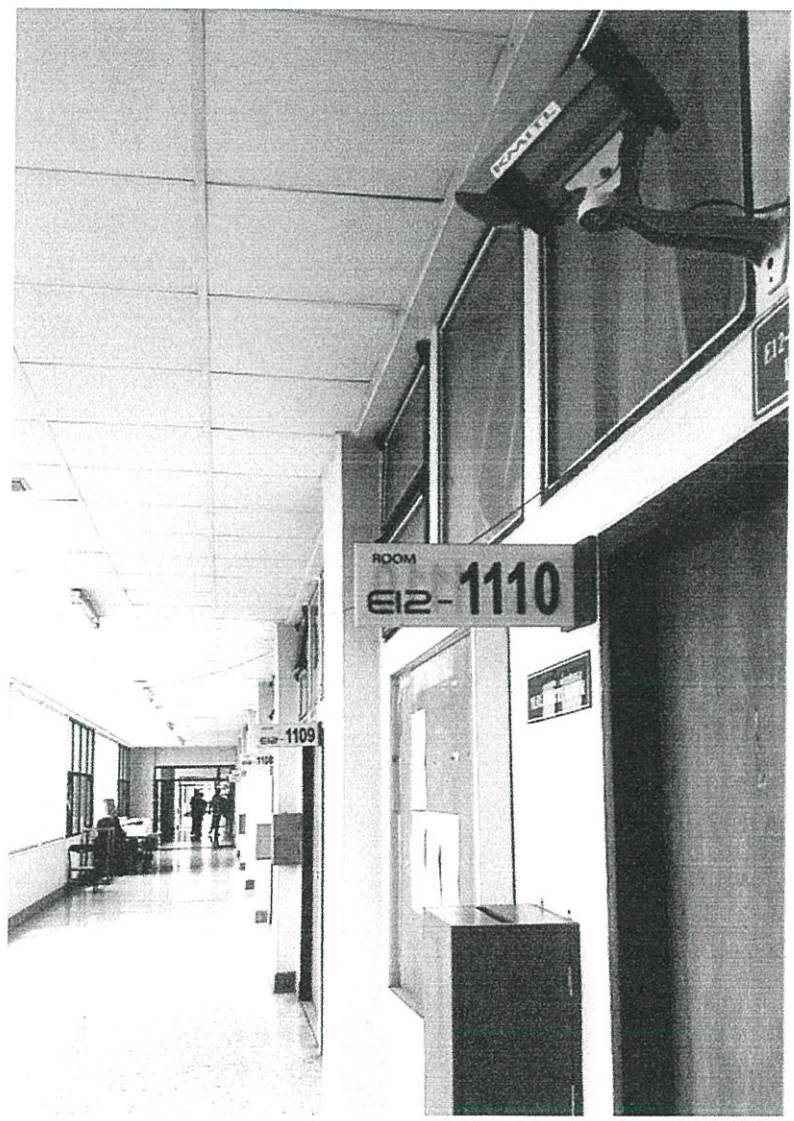
รูปที่ 3.17 แสดงภาพตัวอย่างที่ตรวจจับพบวัตถุสีแดงในขนาดที่ต้องการ

อนึ่ง ในการทดลองนั้นได้ มีการกำหนดค่าของสีไว้ช่วงที่สามารถตรวจจับได้ง่าย ซึ่งอันนี้เป็นประเด็นที่อาจต้องมีการพัฒนาให้มากขึ้นเพราะภาพที่จับได้อาจมีสีที่ผิดเพี้ยนไปทำให้จับภาพไม่ได้ตรงตามต้องการ ซึ่งผลเกิดจากลักษณะของแสงที่สถานที่ๆติดตั้ง ดังนั้นเพื่อให้ปัญหาน้อยที่สุดก็อาจต้องมีการปรับแต่งโหมดการทำงานนี้ไปด้วยในช่วงที่ติดตั้งกล้อง ทั้งนี้เนื่องจากกล้องแต่ละตัวทำงานอิสระต่อกันจึงสามารถมีค่าที่กำหนดช่วงสีได้ต่างกัน ในการวิจัยครั้งนี้เป็น ตัวอย่างต้นแบบเท่านั้น หากจะต้องนำไปใช้งานจริงจังโดยวิธีการนี้จึงควรจะต้องมีการพัฒนาส่วนนี้ให้ทำงานง่ายขึ้น อาจเป็นการปรับแต่งอัตโนมัติเป็นต้น ก็จะสะดวกต่อการติดตั้งนั่นเอง

บทที่ 4
ผลการวิจัย

4.1 ผลต่างๆ

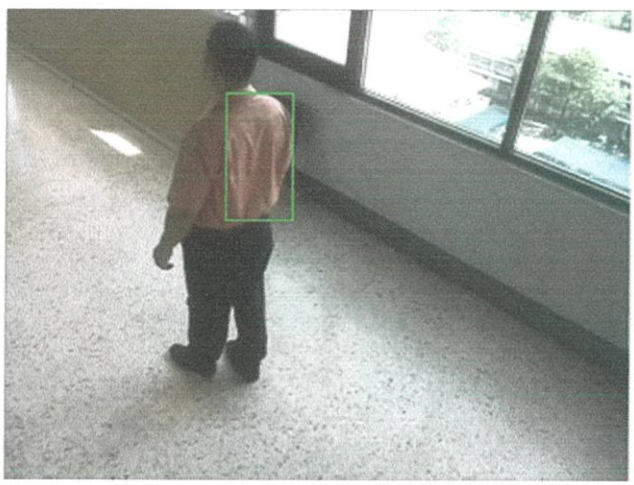
จากหัวข้อที่ผ่านมานั้นได้ อธิบายการใช้งานของระบบบันทึกภาพไปแล้ว ซึ่งก็ได้รวมไปถึงผลทดลองไปด้วยในตัวอย่างใช้ในที่ตึกคณะวิศวกรรมศาสตร์ ชั้น 11 (ตึก 12 ชั้น) โดยติดตั้งไว้ที่หน้าห้องของผู้วิจัยเองซึ่งมีการใช้งานติดกับห้องอาจารย์ ผศ. ไพศาล สิทธิโยภาสกุล ซึ่งมีนักศึกษาโปรเจ็กทำงานอยู่ด้วยหลายคน โดยการติดตั้งไว้ดังรูปที่ 4.1



รูปที่ 4.1 แสดงตำแหน่งติดตั้งกล้องเพื่อการทดสอบอีกสถานที่

ภาพตัวอย่างจากที่บันทึกได้ และภาพตัวอย่างจากการทดลองจากที่ใช้ตรวจจับวัตถุ (คนใส่เสื่อสีเหลือและแดง) แสดงให้เห็นดังรูปที่ 4.2, 4.3 จากผลการทดลอง ก็สามารถสรุปได้ว่าสามารถทำงานได้ดังเป้าประสงค์ แต่ก็ยังมีข้อที่ควรจะต้องพิจารณาปรับปรุงหากจะต้องนำไปใช้งานจริงที่ดีมากขึ้นไปอีก ซึ่งจะได้กล่าวในหัวข้อของข้อเสนอแนะนำ ต่อไป

รูปที่ 4.2 ผลตัวอย่างภาพที่ได้จากการทำงานในโหมดตรวจจับวัตถุสีเหลือง



รูปที่ 4.3 ผลตัวอย่างภาพที่ได้จากการทำงานในโหมดตรวจจับวัตถุสีแดง

4.2 จุดเด่นของงานวิจัยนี้

งานวิจัยนี้เป็นการสร้าง ระบบบันทึกภาพอัตโนมัติราคาประหยัด ที่สามารถควบคุมการทำงานได้ผ่านเครือข่ายไร้สายคอมพิวเตอร์ ซึ่งในปัจจุบันมีใช้กันมากมาย มีขายในท้องตลอดทั่วไปและหลายยี่ห้อ แต่ด้วยที่ระบบยังมีราคาสูง และการปรับเปลี่ยนให้ทำงานได้ตามเป้าประสงค์พิเศษที่ต้องการก็ทำได้ยากมากหรือทำไม่ได้เลย การพัฒนาต่อยอดจึงลำบาก ในงานวิจัยนี้ได้แก้ปัญหาดังกล่าว ดังนั้นจุดเด่นของงานวิจัยนี้ น่าจะอยู่ตรงที่สามารถ สร้างเองได้ในราคาที่ประหยัดมาก ในขณะที่มีระบบการบันทึกภาพได้ตามปกติ และที่สำคัญสามารถเพิ่มเติมหน้าที่การทำงานพิเศษขึ้นได้อีกตาม แล้วแต่ผู้พัฒนาจะทำ ในตัวอย่างเป็นการทำหน้าที่พิเศษคือตรวจจับวัตถุโดยใช้สีเป็นหลัก ดังนั้นผู้พัฒนาจึงน่าจะสามารถนำไปพัฒนาต่อเนื่องได้อย่างรวดเร็วตามความต้องการ จุดเด่นอีกอย่างคือเนื่องจากว่าอุปกรณ์ด้านฮาร์ดแวร์สามารถได้ภายในประเทศ อีกทั้งยังมีการพัฒนาตัวประมวลผลอยู่ตลอดเวลา ทำให้ความเร็วเพิ่มมากขึ้น ในขณะที่ราคาก็ถูกลง อันนี้ก็นับเป็นข้อจุดเด่นหรือข้อได้เปรียบอีกอย่างเพราะสามารถปรับเปลี่ยนฮาร์ดแวร์ได้ทันที แม้ตัวซอฟร์แวร์ก็ยังอาจใช้ตัวเดิมอยู่หรือแม้หากมีการเปลี่ยนแปลงไปบ้างก็ไม่มาก ทำให้ การซ่อม, และการพัฒนา เป็นไปได้อย่างต่อเนื่องรวดเร็วสามารถปรับแต่งหน้าที่พิเศษด้านการใช้งานให้เหมาะสมกับการใช้งานหลากหลายมากขึ้นนั่นเอง

บทที่ 5
สรุปผลการวิจัยและข้อเสนอแนะ

5.1 สรุปผลการวิจัยและข้อเสนอแนะ

งานวิจัยนี้ได้นำไปใช้งานทดสอบใช้งานที่ตึก 12 ชั้น ชั้นที่ 11 โดยมีการใช้งานบันทึกภาพปกติและทดลองการจับวัตถุตามกำหนด ดังที่ได้กล่าวมาแล้วและจากการที่ได้นำไปทดลองทดสอบการใช้งานนั้น ถึงแม้ว่าจะสามารถทางานได้ตามความต้องการก็ตาม แม้จะมีการใช้เวลาในการทดสอบไม่นานนัก แต่ก็ได้มีข้อ พิจารณาจากการทดลองของผู้ใช้ หรือข้อเสนอแนะมาบ้างข้อดังนี้

ข้อที่ 1 ตัวกล้องในการทำงานในโหมดของการบันทึกภาพ มีการเก็บภาพไว้บนหน่วยความจำแบบ USB Drive หรือ บน SD-CARD ซึ่งอาจมีจำนวนหน่วยความจำน้อย หากสามารถเชื่อมต่อ Hard Disk ได้ก็จะสามารถเก็บภาพได้มากขึ้น

ในข้อนี้นั้นผู้วิจัยเห็นด้วย เพียงแต่สังเกตุได้ว่าระบบที่สร้างขึ้นจะไม่มีตัวประมวลผลที่เรียกว่า VPR ซึ่งการเก็บภาพจะเก็บไว้บนตัวกล้องหมด เพราะตัวกล้องจะประมวลผลเองไม่ว่าจะอยู่ในโหมดตรวจจับวัตถุก็ตาม ทั้งนี้เพราะตัดปัญหาการรับส่งข้อมูลที่เป็นจำนวนมากผ่านระบบอินเตอร์เนท เพราะการรับส่งข้อมูลหากเป็นภาพก็จะเสียเวลามาก หากมีข้อผิดพลาดของระบบอินเตอร์เนท ก็จะทำให้ข้อมูลเสียหายได้ ผู้วิจัยจึงมีแนวคิดใช้งาน อินเตอร์เนท เฉพาะการรับส่งคำสั่งเท่านั้น และหากมีการโหลดภาพมาไว้ที่ผู้ใช้ถึงแม้การเชื่อมต่ออินเตอร์เนทจะมีปัญหาก็จะไม่เป็นอะไรเพราะข้อมูลภาพถูกเก็บไว้แล้วในกล้องนั่นเอง แต่แนวทางแก้ไขก็สามารถเป็นไปได้ โดยการเปลี่ยนใช้คอมพิวเตอร์ฝังตัวที่สามารถสูงขึ้น สามารถเชื่อมต่อกับ Hard disk ได้ซึ่งก็มีให้เลือกหลายแบบเช่นกัน

ข้อที่ 2 ตัวเครื่องประมวลผลค่อนข้างช้าในส่วนของการตรวจจับวัตถุ บางครั้งก็ผิดพลาด

สำหรับปัญหาข้อนี้ผู้วิจัยก็พบในการทดลองเอง เช่นกัน แต่ก็เรียกว่าน้อย ซึ่งวิเคราะห์แล้วน่าจะเป็นเพราะความเร็วของตัวโปรเซสเซอร์มีการทำงานที่ช้า (ที่ใช้ปัจจุบันคือ 900 MHz) ณ.เวลาที่ได้สร้างต้นแบบเป็นรุ่น Raspberrpy pi2 ซึ่งก็ได้ใช้ภาพขนาดที่เล็กสุดแล้วคือ 320x240 แต่ก็ยังมีปัญหาดังกล่าว แต่อย่างไรก็ตามก็มีข่าวการพัฒนา Raspberryp pi3 ขึ้นมาในช่วงหลังซึ่งคุณสมบัติดีขึ้นมากและผู้วิจัยคิดว่ายังจะมีการพัฒนามากขึ้นต่อไปอีก ดังนั้นปัญหานี้ก็คงจะแก้ไขได้ดังเช่นปัญหาข้อที่ 1 ที่ผ่านมา คือการเปลี่ยนหน่วยประมวลผลเป็น คอมพิวเตอร์ฝังตัวที่เร็วขึ้นนั่นเอง (นอกจาก Raspberry pi ก็มียี่ห้ออื่นอีกมาก) ซึ่งอันนี้น่าจะได้มีการทดสอบเลือกใช้พัฒนาต่อเนื่องไปได้อีก

บทที่ 6
สรุปผลผลิตที่ได้จากงานวิจัย

จากการทำวิจัยเรื่อง ระบบบันทึกภาพอัตโนมัติราคาประหยัด ที่เกิดเนื่องจากเกิดจากแนวคิดในการออกแบบสร้างตัวกล้องเองและออกแบบซอฟร์แวร์ควบคุมการทำงานเอง เพื่อให้ได้ระบบที่สามารถบันทึกภาพพร้อมกับสามารถเพิ่มเติมหน้าที่การทำงานพิเศษต่างๆได้เอง ให้ทำงานได้ตรงตามต้องการ โดยอุปกรณ์ทุกชิ้นสามารถหาได้ในเมืองไทย เพื่อให้การซ่อม, พัฒนาต่อยอด สามารถทำได้โดยรวด และสุดท้ายราคาที่ถูก ดังนั้นเมื่อเสร็จสิ้นต้นแบบนี้แล้วผลที่ได้ก็จะประกอบไปด้วย

1) ส่วนของตัวกล้อง จำนวน 2 เครื่อง
2) ส่วนของโปรแกรมติดตั้งบนคอมพิวเตอร์ทั่วไป จำนวน 1 ชุด

โดยจะมีรายงานฉบับสมบูรณ์พร้อมกับแผ่น CD-ROM บรรจุเอกสาร, โปรแกรมส่วนต่างๆทั้งหมดเพื่อการนำไปพัฒนาต่อยอดเพิ่มความสามารถในด้านต่างๆ ได้อย่างเหมาะสมกับการใช้งานให้หลากหลายต่อไป

บรรณานุกรม เอกสารอ้างอิง

[1] https://en.wikipedia.org/wiki/Network_video_recorder

[2] https://en.wikipedia.org/wiki/Wi-Fi

[3] https://www.raspberrypi.org/

[4] http://www.opencv.com/

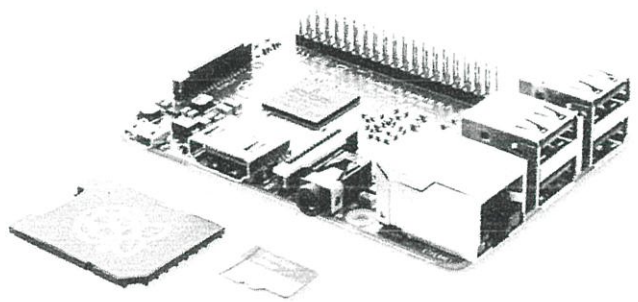
[5] http://www.java.com/en/

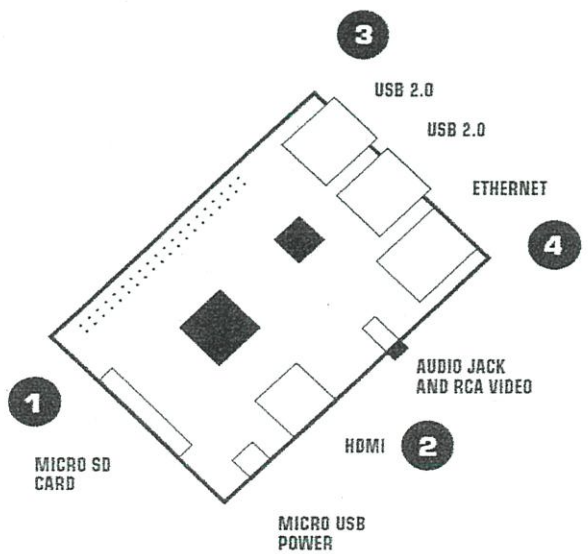
ภาคผนวก ก
ส่วนของฮาร์ดแวร์

ในส่วนของอุปกรณ์ในการสร้าง หรือ ฮาร์ดแวร์ นั้น ดังที่ได้กล่าวมาว่า อุปกรณ์ทุกชิ้นสามารถหาได้ในเมืองไทย จึงข้อแสดงอุปกรณ์ในแต่ชิ้นในการนำมาสร้างตัวกล้อง ดังต่อไปนี้

ตัวบอร์ดประมวลผลหลัก Raspberry pi2

เป็นบอร์ดขนาด 32 บิท ที่สามารถเชื่อมต่ออุปกรณ์ต่างๆ ที่จำเป็นเช่นกล้อง USB หรืออุปกรณ์อื่นๆได้อย่างสะดวก เพราะเป็นคอมพิวเตอร์ฝังตัวขนาดเล็กที่ทำงานได้ อาจใกล้เคียงกับคอมพิวเตอร์ตั้งโต๊ะ แต่ราคาต่ำกว่าและขนาดเล็กกว่า มีระบบปฏิบัติการที่ใช้งานคือ Linux ดังมีรายละเอียดบางอย่างต่อไปนี้ รูปบางส่วนแสดงดังรูปภาพข้างล่างนี้ ส่วนตัวคู่มือบอร์ดและส่วนเอกสารสนับสนุนต่างๆ มีแนบไว้ที่แผ่น CD ROM เรียบร้อยแล้ว



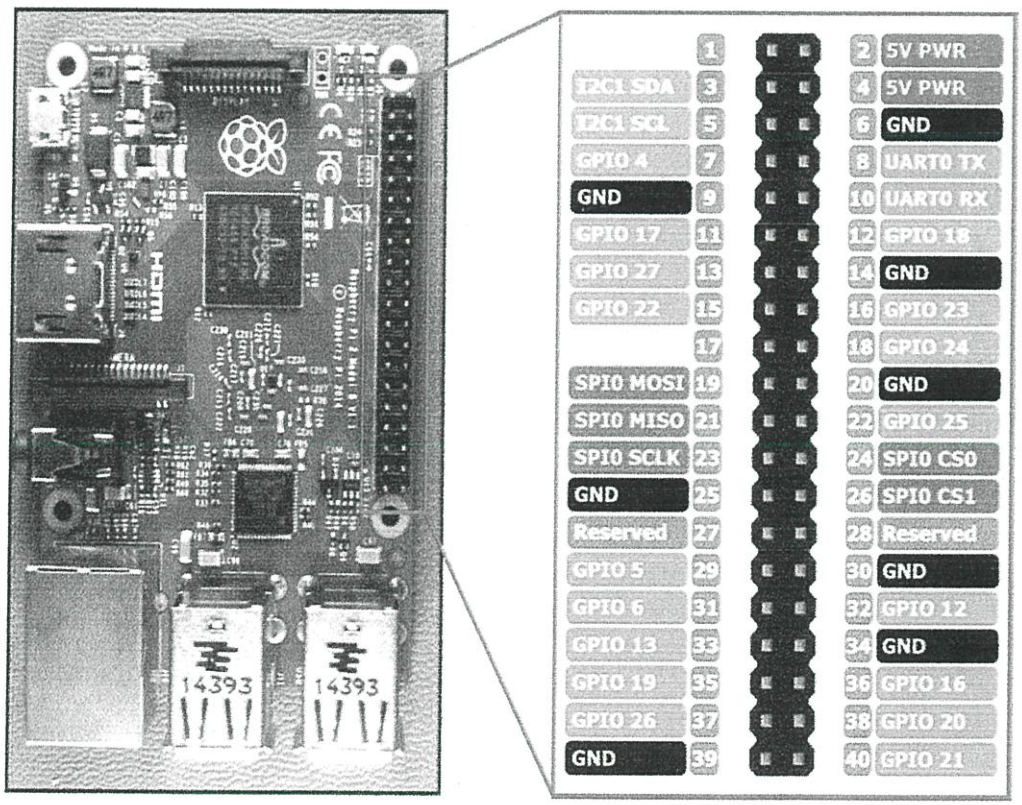


SPEC DATA:

Raspberry Pi 3 Model B

Type: Value, Mini Desktop PC
Processor Family: Broadcom BCM2837
Processor Name: Broadcom BCM2837 ARMv8 Cortex A53
Processor Speed: 1.2 GHz
RAM: 1 GB
Storage Capacity (as Tested): N/A GB
Storage Type: Flash storage

Graphics Card: VideoCore IV
Primary Optical Drive: External
Secondary Optical Drive: None
Monitor Type: None
Touch Screen: No
Screen Size: N/A inches
Operating System: Linux
Built-in TV Tuner: None

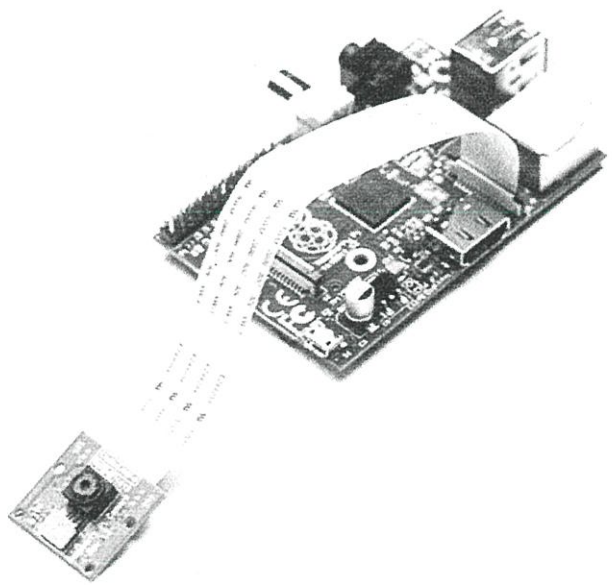


อุปกรณ์โมดูลกล้อง Picamera

เป็นตัวโมดูลกล้องที่มีขนาดที่เล็กและทำมาเฉพาะให้กับบอร์ดคอมพิวเตอร์ฝังตัว Raspberry Pi2 ตัวกล้องมีการเชื่อมต่อแบบสายแพ ยาวประมาณ 1 ฟุต การควบคุมการทำงานทำได้บนภาษาหลายภาษา เช่น C++, Python, C, JAVA เหล่านี้เป็นต้น สามารถทำการปรับให้ได้ความละเอียดของภาพได้มากมายดังตารางข้างล่างและมีคุณสมบัติทางไฟฟ้าดังต่อไปนี้

	Camera V1	Camera V2
Net price	$25	$25
Size	around 25 x 24 x 9 mm	
Weight	3g	
Still resolution	5 Megapixels	8 Megapixels
Video modes	1080p30, 720p60 and 640x480p60/90	1080p30, 720p60 and 640x480p60/90
Linux integration	V4L2 driver available	V4L2 driver available
C programming API	OpenMAX IL and others available	OpenMAX IL and others available
Sensor	Omnivision OV5647	Sony IMX219
Sensor resolution	2592 x 1944 pixels	3280 x 2464 pixels
Sensor image area	3.76 x 2.74 mm	
Pixel size	1.4 µm x 1.4 µm	
Optical size	1/4	
Full-frame SLR lens equivalent	35 mm	
S/N ratio	36 dB	
Dynamic range	67 dB @ 8x gain	
Sensitivity	680 mV/lux-sec	
Dark current	16 mV/sec @ 60 C	
Well capacity	4.8 Ke-	
Fixed focus	1 m to infinity	
Focal length	3.60 mm +/- 0.01	
Horizontal field of view	53.50 +/- 0.13 degrees	
Vertical field of view	41.41 +/- 0.11 degrees	
Focal ratio (F-Stop)	2.9	

#	Resolution	Aspect Ratio	Framerates	Video	Image	FoV	Binning
1	1920x1080	16:9	1-30fps	x		Partial	None
2	2592x1944	4:3	1-15fps	x	x	Full	None
3	2592x1944	4:3	0.1666-1fps	x	x	Full	None
4	1296x972	4:3	1-42fps	x		Full	2x2
5	1296x730	16:9	1-49fps	x		Full	2x2
6	640x480	4:3	42.1-60fps	x		Full	4x4
7	640x480	4:3	60.1-90fps	x		Full	4x4



ตัวกล้อง Picamera และตำแหน่งที่ติดตั้ง

ภาคผนวก ข
ส่วนของโปรแกรม

ในส่วนของโปรแกรมนั้นจะมีอยู่ด้วยกันหลายโปรแกรมทำงานร่วมกัน ในส่วนของภาคผนวกนี้จะขอแสดงบางส่วนเท่านั้นเพราะโปรแกรมค่อนข้างยาว แต่ทุกส่วนโปรแกรมจะมีอยู่ในแผ่น CD-ROM อยู่แล้ว

โปรแกรมส่วนตัวกล้อง ประกอบไปด้วย ส่วนการทำงานโหมดบันทึกภาพ และส่วนของการประมวลผลภาพตรวจจับหาวัตถุ

โปรแกรมส่วนของตัวกล้อง

```
/* MAIN PROGRAM ON RASPBERRY PI  (JAVA) */
package camone;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetSocketAddress;
import java.net.SocketException;
import java.util.Scanner;
/**
 *
 * @author Attasit
 */
public class CamOne {

        /**
         * @param args the command line arguments
         */
        public static void main(String[] args) throws SocketException, IOException {
                CamOne obj = new CamOne();
        byte[] buffer = new byte[1024];
                Scanner scanner = new Scanner(System.in);
                System.out.print("Name : ");
```

```
            System.out.print("Source port : ");
            System.out.print("Destination IP  : ");
            System.out.print("Destination port : ");
    String name =" RPI ";
        int sourcePort = Integer.parseInt("8000");
        String DestinationIP = "192.168.1.5";  //PC Adress
        int DestinationPort = Integer.parseInt("12345"); //PC Port
      InetSocketAddress address = new InetSocketAddress(DestinationIP,DestinationPort);
            DatagramSocket  socket = new DatagramSocket(sourcePort);
            while(true){
               DatagramPacket packet = new DatagramPacket(buffer,buffer.length); //put
here to reset
                    System.out.print("Wait :.. ");
                    socket.receive(packet);
                    String msg = new String(buffer,0,packet.getLength());
                    System.out.println(msg);
                    int choice = Integer.parseInt(msg);
                    switch (choice){
                          case 0 : // Start  Take Picture every 3 seconds
                        Runtime.getRuntime().exec("python GrabImg.py");
                              break;
                        case 1 : // Stop Record image
                              Runtime.getRuntime().exec("./GrabImgStop.sh");
                                  break;
                        case 11 : // Start  Tracking Color RED
                               Runtime.getRuntime().exec("python pi3cam_red.py");
                                 break;
                        case 12 : // Start  Tracking Color GREEN
                              Runtime.getRuntime().exec("sudo ./stopP.sh");
                              Runtime.getRuntime().exec("python pi3cam_green.py");
                                 break;
                        case 13 : // Start  Tracking Color BLUE
                                 Runtime.getRuntime().exec("sudo ./stopP.sh");
                               Runtime.getRuntime().exec("python pi3cam_blue.py");
                                 break;
```

```
                        case 14 : // Start  Tracking Color YELLOW
                                Runtime.getRuntime().exec("sudo ./stopP.sh");
                               Runtime.getRuntime().exec("python pi3cam_yellow.py");
                                break;
                        case 2 : // Stop all
                                Runtime.getRuntime().exec("sudo ./stopP.sh");
                                break;
                        case 3 : // Start send image continue..chane port for each
raspberry pi!!!.
                                Runtime.getRuntime().exec("./video_stream.sh");
                                break;
                        case 4 : // Stop send image continue! No..need by now (use '2'
instread)
                                Runtime.getRuntime().exec("ps -ef | grep raspivid | awk
'{print $2}' | xargs kill");
                                        break;

                        case 5 : // Stop Main program
                                Runtime.getRuntime().exec("./CamOneStop.sh");
                                break;
                        default : // No command and just send black suituation
                                msg=" [1]Take Picture is.....  [2]Track Color is.....";
                                    buffer = msg.getBytes();
                                    buffer = msg.getBytes();
                DatagramPacket S4packet = new DatagramPacket(buffer,buffer.length);
                                S4packet.setSocketAddress(address);
                                socket.send(S4packet);
                                break;
                        }
                }
        }
        private String executeCommand(String command){
                StringBuffer output = new StringBuffer();
                Process p;
                try{
```

```
                    p = Runtime.getRuntime().exec(command);
                    p.waitFor();
                    BufferedReader reader = new BufferedReader(new
InputStreamReader(p.getInputStream()));
                    String line = "";
                while((line = reader.readLine())!=null){
                    output.append(line+"\n");
                }
            } catch(Exception e){
                e.printStackTrace();
            }
                return output.toString();
        }
}
```

โปรแกรมส่วนของผู้ใช้ (GUI)

แสดงเฉพาะส่วนของตัวสร้างเฟรมเมนูเท่านั้น ส่วนอื่นๆที่เหลือจะอยู่ในแผ่น CD-ROM ทั้งหมด

```
/*   MAIN PROGRAM FOR GUI (JAVA)  */
package cam_menu;

import java.awt.Color;
import java.awt.Image;
import java.io.File;
import java.io.IOException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetSocketAddress;
import java.net.SocketException;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.ImageIcon;
import java.util.Timer;
import java.util.TimerTask;

/**
 *
 * @author Attasit
 */
public class MainFrom extends javax.swing.JFrame {

                    DatagramSocket socket;

                public MainFrom() throws SocketException {
                    initComponents();
                    this.setTitle("Camera recorder");
                m_Readfile_Button_1.setEnabled(false);
                m_Stop_Button.setEnabled(false);
                m_Stop_Track_Button.setEnabled(false);
                m_Readfile_Button_1.setEnabled(true);
                m_stopChecking.setEnabled(false);
                m_preview_2.setIcon(new ImageIcon(new
ImageIcon("D:\\WORK\\RESEARCH_KMITL\\research_proposals59\\Camera\\Cam_menu\\logo-
small.gif").getImage().getScaledInstance(200, 200, Image.SCALE_DEFAULT)));
```

```
                    m_preview_1.setIcon(new ImageIcon(new
ImageIcon("D:\\WORK\\RESEARCH_KMITL\\research_proposals59\\Camera\\Cam_menu\\logo-
small.gif").getImage().getScaledInstance(200, 200, Image.SCALE_DEFAULT)));
                    thread_Dis.start();
                    thread_Dis.suspend();
                    }

File f1 = new File("D:\\WORK\\RESEARCH_KMITL\\research_proposals59\\Camera\\Cam_menu\\cam1.jpg");
File f2 = new File("D:\\WORK\\RESEARCH_KMITL\\research_proposals59\\Camera\\Cam_menu\\cam2.jpg");

Runnable DisPack = new Runnable(){
                        @Override
                        public void run() {
                            while(true){
                                if (f1.exists()){
                    m_preview_1.setIcon(new ImageIcon(new
ImageIcon("D:\\WORK\\RESEARCH_KMITL\\research_proposals59\\Camera\\Cam_menu\\cam1.jpg").getImag
e().getScaledInstance(200, 200, Image.SCALE_DEFAULT)));
                        new
File("D:\\WORK\\RESEARCH_KMITL\\research_proposals59\\Camera\\Cam_menu\\cam1.jpg").delete();
                                try {
                                    Thread.sleep(2000);
                                } catch (InterruptedException ex) {

                    Logger.getLogger(MainFrom.class.getName()).log(Level.SEVERE, null, ex);
                                }
                                    try {
                                        Runtime.getRuntime().exec("python
D:\\WORK\\RESEARCH_KMITL\\research_proposals59\\Camera\\Cam_menu\\PCrev.py");
                                    } catch (IOException ex) {

                    Logger.getLogger(MainFrom.class.getName()).log(Level.SEVERE, null, ex);
                                    }
                                }
                                else
                                {
                    m_preview_1.setIcon(new ImageIcon(new
ImageIcon("D:\\WORK\\RESEARCH_KMITL\\research_proposals59\\Camera\\Cam_menu\\logo-
small.gif").getImage().getScaledInstance(200, 200, Image.SCALE_DEFAULT)));
```

```
                                                }
                                                if (f2.exists()){
                                m_preview_2.setIcon(new ImageIcon(new
ImageIcon("D:\\WORK\\RESEARCH_KMITL\\research_proposals59\\Camera\\Cam_menu\\cam2.jpg").getImag
e().getScaledInstance(200, 200, Image.SCALE_DEFAULT)));
                                    new
File("D:\\WORK\\RESEARCH_KMITL\\research_proposals59\\Camera\\Cam_menu\\cam2.jpg").delete();
                                                try {
                                                        Thread.sleep(2000);
                                                } catch (InterruptedException ex) {

                                Logger.getLogger(MainFrom.class.getName()).log(Level.SEVERE, null, ex);
                                                }
                                                }
                                                else
                                m_preview_2.setIcon(new ImageIcon(new
ImageIcon("D:\\WORK\\RESEARCH_KMITL\\research_proposals59\\Camera\\Cam_menu\\logo-
small.gif").getImage().getScaledInstance(200, 200, Image.SCALE_DEFAULT)));
                                                try {
                                                        Thread.sleep(3000);
                                                } catch (InterruptedException ex) {

                                Logger.getLogger(MainFrom.class.getName()).log(Level.SEVERE, null, ex);
                                                }
                                        }
                                    }
                                };
    Thread  thread_Dis = new Thread(DisPack);
//**************** SET IP ADRESS & PORT CAMERA1, CAMERA2
                                InetSocketAddress  address1 = new InetSocketAddress("192.168.1.7",8000);
                                InetSocketAddress  address2 = new InetSocketAddress("192.168.1.16",8000);

                                /**
                                 * This method is called from within the constructor to initialize the
                                 * form. WARNING: Do NOT modify this code. The content of this method is
                                 * always regenerated by the Form Editor.
                                 */
                                @SuppressWarnings("unchecked")
  // <editor-fold defaultstate="collapsed" desc="Generated Code">
  private void initComponents() {
```

```
buttonGroup2 = new javax.swing.ButtonGroup();
jLabel1 = new javax.swing.JLabel();
m_preview_1 = new javax.swing.JLabel();
m_preview_2 = new javax.swing.JLabel();
Label_1 = new javax.swing.JLabel();
Label_2 = new javax.swing.JLabel();
jPanel1 = new javax.swing.JPanel();
m_Record_Button_1 = new javax.swing.JButton();
m_Track_Button_2 = new javax.swing.JButton();
m_Readfile_Button_1 = new javax.swing.JButton();
m_Stop_Button = new javax.swing.JButton();
m_RadioButton_blue = new javax.swing.JRadioButton();
m_RadioButton_yellow = new javax.swing.JRadioButton();
m_RadioButton_red = new javax.swing.JRadioButton();
m_RadioButton_green = new javax.swing.JRadioButton();
m_Exit_Button = new javax.swing.JButton();
m_Stop_Track_Button = new javax.swing.JButton();
m_CamCheck = new javax.swing.JButton();
m_stopChecking = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

jLabel1.setFont(new java.awt.Font("TH SarabunPSK", 1, 48)); // NOI18N
jLabel1.setText("ระบบบันทึกภาพอัตโนมัติราคาประหยัด");

m_preview_1.setBackground(new java.awt.Color(0, 204, 153));
m_preview_1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
m_preview_1.setText(" ");
m_preview_1.setAlignmentY(0.0F);
m_preview_1.setBorder(javax.swing.BorderFactory.createTitledBorder(""));

m_preview_2.setBackground(new java.awt.Color(0, 51, 255));
m_preview_2.setForeground(new java.awt.Color(0, 204, 102));
m_preview_2.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
m_preview_2.setText(" ");
m_preview_2.setBorder(javax.swing.BorderFactory.createTitledBorder(""));

Label_1.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
Label_1.setText("ภาพจาก กล้องที่ 1");
```

```
Label_2.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
Label_2.setText("ภาพจาก กล้องที่ 2");

jPanel1.setLayout(new java.awt.BorderLayout());

m_Record_Button_1.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
m_Record_Button_1.setText("Video Record mode");
m_Record_Button_1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        m_Record_Button_1ActionPerformed(evt);
    }
});

m_Track_Button_2.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
m_Track_Button_2.setText("Start Tracking");
m_Track_Button_2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        m_Track_Button_2ActionPerformed(evt);
    }
});

m_Readfile_Button_1.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N
m_Readfile_Button_1.setText("Read files from Camera");
m_Readfile_Button_1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        m_Readfile_Button_1ActionPerformed(evt);
    }
});

m_Stop_Button.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
m_Stop_Button.setText("Stop Record");
m_Stop_Button.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        m_Stop_ButtonActionPerformed(evt);
    }
});

buttonGroup2.add(m_RadioButton_blue);
m_RadioButton_blue.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N
```

```
m_RadioButton_blue.setForeground(new java.awt.Color(51, 51, 255));
m_RadioButton_blue.setText("Blue");

buttonGroup2.add(m_RadioButton_yellow);
m_RadioButton_yellow.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N
m_RadioButton_yellow.setForeground(new java.awt.Color(255, 255, 0));
m_RadioButton_yellow.setText("Yellow");

buttonGroup2.add(m_RadioButton_red);
m_RadioButton_red.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N
m_RadioButton_red.setForeground(new java.awt.Color(255, 102, 51));
m_RadioButton_red.setSelected(true);
m_RadioButton_red.setText("Red");

buttonGroup2.add(m_RadioButton_green);
m_RadioButton_green.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N
m_RadioButton_green.setForeground(new java.awt.Color(0, 255, 0));
m_RadioButton_green.setText("Green");

m_Exit_Button.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N
m_Exit_Button.setText("Exit");
m_Exit_Button.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        m_Exit_ButtonActionPerformed(evt);
    }
});

m_Stop_Track_Button.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
m_Stop_Track_Button.setText("Stop Tracking");
m_Stop_Track_Button.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        m_Stop_Track_ButtonActionPerformed(evt);
    }
});

m_CamCheck.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
m_CamCheck.setText("Camera Check");
m_CamCheck.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        m_CamCheckActionPerformed(evt);
```

```
        }
    });

    m_stopChecking.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
    m_stopChecking.setText("Stop Checking");
    m_stopChecking.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            m_stopCheckingActionPerformed(evt);
        }
    });

    javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                .addGroup(layout.createSequentialGroup()
                    .addGap(32, 32, 32)
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                        .addGroup(layout.createSequentialGroup()
                            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                                .addGroup(layout.createSequentialGroup()
                                    .addComponent(m_CamCheck, javax.swing.GroupLayout.PREFERRED_SIZE,
126, javax.swing.GroupLayout.PREFERRED_SIZE)
                                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                                    .addComponent(m_stopChecking, javax.swing.GroupLayout.PREFERRED_SIZE,
136, javax.swing.GroupLayout.PREFERRED_SIZE))
                                .addGroup(layout.createSequentialGroup()
                                    .addComponent(m_Record_Button_1)
                                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                                    .addComponent(m_Stop_Button)))
                            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                            .addComponent(m_Exit_Button, javax.swing.GroupLayout.PREFERRED_SIZE, 70,
javax.swing.GroupLayout.PREFERRED_SIZE)
                            .addGap(26, 26, 26))
                        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
                            .addGap(0, 0, Short.MAX_VALUE)
```

```
                        .addComponent(m_Track_Button_2)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addComponent(m_Stop_Track_Button))
                    .addGroup(layout.createSequentialGroup()
                        .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE, 381,
javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addGap(0, 0, Short.MAX_VALUE))
                    .addGroup(layout.createSequentialGroup()
                        .addGap(26, 26, 26)
                        .addComponent(m_preview_1, javax.swing.GroupLayout.PREFERRED_SIZE, 200,
javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 125,
Short.MAX_VALUE)
                        .addComponent(m_preview_2, javax.swing.GroupLayout.PREFERRED_SIZE, 200,
javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addGap(23, 23, 23))
                    .addGroup(layout.createSequentialGroup()
                        .addComponent(m_Readfile_Button_1)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,
false)
                            .addComponent(m_RadioButton_green, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                            .addComponent(m_RadioButton_red, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,
false)
                            .addComponent(m_RadioButton_yellow,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                            .addComponent(m_RadioButton_blue, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
                        .addGap(44, 44, 44))))
                .addGroup(javax.swing.GroupLayout.Alignment.LEADING, layout.createSequentialGroup()
                    .addGap(64, 64, 64)
                    .addComponent(jLabel1)))
            .addGap(25, 25, 25))
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()
            .addGap(112, 112, 112)
```

```
                .addComponent(Label_1)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(Label_2)
                .addGap(94, 94, 94))
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(12, 12, 12)
                .addComponent(jLabel1)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addComponent(Label_2, javax.swing.GroupLayout.Alignment.TRAILING)
                    .addComponent(Label_1))
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(layout.createSequentialGroup()
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                        .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addGap(18, 18, 18))
                    .addGroup(layout.createSequentialGroup()
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                            .addComponent(m_preview_1, javax.swing.GroupLayout.PREFERRED_SIZE, 200,
javax.swing.GroupLayout.PREFERRED_SIZE)
                            .addComponent(m_preview_2, javax.swing.GroupLayout.PREFERRED_SIZE, 200,
javax.swing.GroupLayout.PREFERRED_SIZE))
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 15,
Short.MAX_VALUE)))
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                        .addComponent(m_Stop_Button, javax.swing.GroupLayout.PREFERRED_SIZE, 32,
javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addComponent(m_Track_Button_2, javax.swing.GroupLayout.PREFERRED_SIZE, 32,
javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addComponent(m_Stop_Track_Button, javax.swing.GroupLayout.PREFERRED_SIZE, 32,
javax.swing.GroupLayout.PREFERRED_SIZE))
                    .addComponent(m_Record_Button_1, javax.swing.GroupLayout.PREFERRED_SIZE, 32,
```

```
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(18, 18, 18)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
               .addComponent(m_Readfile_Button_1, javax.swing.GroupLayout.PREFERRED_SIZE, 37,
javax.swing.GroupLayout.PREFERRED_SIZE)
               .addGroup(layout.createSequentialGroup()
                  .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                     .addComponent(m_RadioButton_red)
                     .addComponent(m_RadioButton_blue))
                  .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                  .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                     .addComponent(m_RadioButton_yellow)
                     .addComponent(m_RadioButton_green))))
            .addGap(18, 18, 18)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
               .addComponent(m_Exit_Button, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)
               .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                  .addComponent(m_CamCheck)
                  .addComponent(m_stopChecking)))
            .addGap(18, 18, 18))
      );

      pack();
   }// </editor-fold>

   private void m_Exit_ButtonActionPerformed(java.awt.event.ActionEvent evt) {
                              try {
                                                      //1]Send command '2' -> Stop all program
                                                      this.socket = new DatagramSocket(12345);
                                                } catch (SocketException ex) {

                              Logger.getLogger(MainFrom.class.getName()).log(Level.SEVERE, null, ex);
                                                }
                                    String msg ="5";
                                    byte[] buffer = msg.getBytes();
                                    DatagramPacket packet = new DatagramPacket(buffer,buffer.length);
                                    packet.setSocketAddress(address1);
                                                try {
                                                      socket.send(packet);
```

```
                    } catch (IOException ex) {

                Logger.getLogger(MainFrom.class.getName()).log(Level.SEVERE, null, ex);
                    }
                packet = new DatagramPacket(buffer,buffer.length);
                packet.setSocketAddress(address2);
                    try {
                        socket.send(packet);
                    } catch (IOException ex) {

                Logger.getLogger(MainFrom.class.getName()).log(Level.SEVERE, null, ex);
                    }
System.exit(0);     // TODO add your handling code here:
}

private void m_Record_Button_1ActionPerformed(java.awt.event.ActionEvent evt) {
                st_flag =false;
                try {
                        //1]Send command '2' -> Stop all program
                        this.socket = new DatagramSocket(12345);
                    } catch (SocketException ex) {

                Logger.getLogger(MainFrom.class.getName()).log(Level.SEVERE, null, ex);
                    }
                String msg ="2";
                byte[] buffer = msg.getBytes();
                DatagramPacket packet = new DatagramPacket(buffer,buffer.length);
                packet.setSocketAddress(address1);
                    try {
                        socket.send(packet);
                    } catch (IOException ex) {

                Logger.getLogger(MainFrom.class.getName()).log(Level.SEVERE, null, ex);
                    }
                packet = new DatagramPacket(buffer,buffer.length);
                packet.setSocketAddress(address2);
                    try {
                        socket.send(packet);
                    } catch (IOException ex) {
```

```
Logger.getLogger(MainFrom.class.getName()).log(Level.SEVERE, null, ex);
                }
                try {
                        //[2] Delay 5 seconds and send command '0'
                        Thread.sleep(5000);
                } catch (InterruptedException ex) {

Logger.getLogger(MainFrom.class.getName()).log(Level.SEVERE, null, ex);
                }

                try {
                        //[3] Send comand '0' <-- to Record mode
                        this.socket = new DatagramSocket(12345);
                } catch (SocketFxception ex) {

Logger.getLogger(MainFrom.class.getName()).log(Level.SEVERE, null, ex);
                }
        msg ="0";
        buffer = msg.getBytes();
        packet = new DatagramPacket(buffer,buffer.length);
        packet.setSocketAddress(address1);
                try {
                        socket.send(packet);
                } catch (IOException ex) {

Logger.getLogger(MainFrom.class.getName()).log(Level.SEVERE, null, ex);
                }
        packet = new DatagramPacket(buffer,buffer.length);
        packet.setSocketAddress(address2);
                try {
                        socket.send(packet);
                } catch (IOException ex) {

Logger.getLogger(MainFrom.class.getName()).log(Level.SEVERE, null, ex);
                }

socket.close();

        m_Track_Button_2.setEnabled(true);
        m_Record_Button_1.setEnabled(false);
```

```
                m_Stop_Track_Button.setEnabled(false);
                m_RadioButton_blue.setEnabled(true);
                m_RadioButton_red.setEnabled(true);
                m_RadioButton_green.setEnabled(true);
                m_RadioButton_yellow.setEnabled(true);
                m_Stop_Button.setEnabled(true);
                m_Readfile_Button_1.setEnabled(false);

}

private void m_Track_Button_2ActionPerformed(java.awt.event.ActionEvent evt) {
                if(!st_flag){
                        try {
                                //1]Send command '1' (Start tracking...)
                                this.socket = new DatagramSocket(12345);
                        } catch (SocketException ex) {

                Logger.getLogger(MainFrom.class.getName()).log(Level.SEVERE, null, ex);
                        }
                        String msg="";
                        if(m_RadioButton_red.isSelected()) msg="11";
                        if(m_RadioButton_green.isSelected()) msg="12";
                        if(m_RadioButton_blue.isSelected()) msg="13";
                        if(m_RadioButton_yellow.isSelected()) msg="14";

                byte[] buffer = msg.getBytes();
                DatagramPacket packet = new DatagramPacket(buffer,buffer.length);
                packet.setSocketAddress(address1);
                        try {
                                socket.send(packet);
                        } catch (IOException ex) {

                Logger.getLogger(MainFrom.class.getName()).log(Level.SEVERE, null, ex);
                        }
                packet = new DatagramPacket(buffer,buffer.length);
                packet.setSocketAddress(address2);
                        try {
                                socket.send(packet);
                        } catch (IOException ex) {
```

```
                    Logger.getLogger(MainFrom.class.getName()).log(Level.SEVERE, null, ex);
                        }
                        st_flag=true;
                    socket.close();
                    }
                    m_Record_Button_1.setEnabled(true);
                    m_Stop_Track_Button.setEnabled(true);
        m_Track_Button_2.setEnabled(false);
                    m_RadioButton_blue.setEnabled(false);
                    m_RadioButton_red.setEnabled(false);
                    m_RadioButton_green.setEnabled(false);
                    m_RadioButton_yellow.setEnabled(false);
                    m_Readfile_Button_1.setEnabled(false);
                    m_Stop_Button.setEnabled(false);
                    FileServer fs = new FileServer(12345);
                    fs.start(); //---> loop check received Image file
                try {
                        Runtime.getRuntime().exec("python
D:\\WORK\\RESEARCH_KMITL\\research_proposals59\\Camera\\Cam_menu\\PCrev.py");
                    } catch (IOException ex) {
                    Logger.getLogger(MainFrom.class.getName()).log(Level.SEVERE, null, ex);
                        }       //---> wait for Image file and save to cam1.jpg
      thread_Dis.resume();// Start display img if exist
    }


    private void m_Stop_Track_ButtonActionPerformed(java.awt.event.ActionEvent evt) {
                    m_Stop_Track_Button.setEnabled(false);
                    m_Track_Button_2.setEnabled(true);
                    m_RadioButton_blue.setEnabled(true);
                    m_RadioButton_red.setEnabled(true);
                    m_RadioButton_green.setEnabled(true);
                    m_RadioButton_yellow.setEnabled(true);
                    m_Readfile_Button_1.setEnabled(true);

                    thread_Dis.suspend();
                new
File("D:\\WORK\\RESEARCH_KMITL\\research_proposals59\\Camera\\Cam_menu\\cam1.jpg").delete();
                new
File("D:\\WORK\\RESEARCH_KMITL\\research_proposals59\\Camera\\Cam_menu\\cam2.jpg").delete();
```

```
                try {
                        Runtime.getRuntime().exec("Taskkill /F /IM python.exe");
                } catch (IOException ex) {
                Logger.getLogger(MainFrom.class.getName()).log(Level.SEVERE, null, ex);
                        }
                // Send command stop all "2"
try {
                                //1]Send command '2' -> Stop all program
                                this.socket = new DatagramSocket(12345);
                        } catch (SocketException ex) {

                Logger.getLogger(MainFrom.class.getName()).log(Level.SEVERE, null, ex);
                        }
                String msg ="2";
                byte[] buffer = msg.getBytes();
                DatagramPacket packet = new DatagramPacket(buffer,buffer.length);
                packet.setSocketAddress(address1);
                        try {
                                socket.send(packet);
                        } catch (IOException ex) {

                Logger.getLogger(MainFrom.class.getName()).log(Level.SEVERE, null, ex);
                        }
                packet = new DatagramPacket(buffer,buffer.length);
                packet.setSocketAddress(address2);
                        try {
                                socket.send(packet);
                        } catch (IOException ex) {

                Logger.getLogger(MainFrom.class.getName()).log(Level.SEVERE, null, ex);
                        }
                        st_flag = false;
                        m_preview_2.setIcon(new ImageIcon(new
ImageIcon("D:\\WORK\\RESEARCH_KMITL\\research_proposals59\\Camera\\Cam_menu\\logo-
small.gif").getImage().getScaledInstance(200, 200, Image.SCALE_DEFAULT)));
                m_preview_1.setIcon(new ImageIcon(new
ImageIcon("D:\\WORK\\RESEARCH_KMITL\\research_proposals59\\Camera\\Cam_menu\\logo-
small.gif").getImage().getScaledInstance(200, 200, Image.SCALE_DEFAULT)));
    }
```

```
    private void m_Stop_ButtonActionPerformed(java.awt.event.ActionEvent evt) {
                                    st_flag = false;
                                    m_Record_Button_1.setEnabled(true);
                                    m_Stop_Button.setEnabled(false);
                                    m_Readfile_Button_1.setEnabled(true);
                                try {
                                                //1]Send command '1' -> Stop recording program
                                                this.socket = new DatagramSocket(12345);
                                        } catch (SocketException ex) {

                                Logger.getLogger(MainFrom.class.getName()).log(Level.SEVERE, null, ex);
                                        }
                                    String msg ="1";
                                    byte[] buffer = msg.getBytes();
                                    DatagramPacket packet = new DatagramPacket(buffer,buffer.length);
                                    packet.setSocketAddress(address1);
                                            try {
                                                    socket.send(packet);
                                            } catch (IOException ex) {

                                Logger.getLogger(MainFrom.class.getName()).log(Level.SEVERE, null, ex);
                                            }
                                        packet = new DatagramPacket(buffer,buffer.length);
                                        packet.setSocketAddress(address2);
                                            try {
                                                    socket.send(packet);
                                            } catch (IOException ex) {

                                Logger.getLogger(MainFrom.class.getName()).log(Level.SEVERE, null, ex);
                                            }
                                            m_preview_2.setIcon(new ImageIcon(new
ImageIcon("D:\\WORK\\RESEARCH_KMITL\\research_proposals59\\Camera\\Cam_menu\\logo-
small.gif").getImage().getScaledInstance(200, 200, Image.SCALE_DEFAULT)));
                                m_preview_1.setIcon(new ImageIcon(new
ImageIcon("D:\\WORK\\RESEARCH_KMITL\\research_proposals59\\Camera\\Cam_menu\\logo-
small.gif").getImage().getScaledInstance(200, 200, Image.SCALE_DEFAULT)));
    }

    private void m_Readfile_Button_1ActionPerformed(java.awt.event.ActionEvent evt) {
                                    try {
```

```
                                        Runtime.getRuntime().exec("C:/Program Files
(x86)/WinSCP/WinSCP.exe");
                                } catch (IOException ex) {
                                        Logger.getLogger(MainFrom.class.getName()).log(Level.SEVERE,
null, ex);
                                }
    }

    private void m_CamCheckActionPerformed(java.awt.event.ActionEvent evt) {
                                //Disable all menu except stop checking menu
                            m_Readfile_Button_1.setEnabled(false);
    m_Stop_Button.setEnabled(false);
                            m_Stop_Track_Button.setEnabled(false);
    m_stopChecking.setEnabled(true);
                            m_Record_Button_1.setEnabled(false);
    m_RadioButton_green.setEnabled(false);
    m_RadioButton_red.setEnabled(false);
    m_RadioButton_blue.setEnabled(false);
    m_RadioButton_yellow.setEnabled(false);
                            m_Track_Button_2.setEnabled(false);
                            m_CamCheck.setEnabled(false);
                            // Send command 3 => Camera check
                                    if(!st_flag){
                                    try {
                                            //1]Send command '3' (Start checking camera)
                                            this.socket = new DatagramSocket(12345);
                                    } catch (SocketException ex) {

                            Logger.getLogger(MainFrom.class.getName()).log(Level.SEVERE, null, ex);
                                    }
                                String msg ="3";
                                byte[] buffer = msg.getBytes();
                                DatagramPacket packet = new DatagramPacket(buffer,buffer.length);
                                packet.setSocketAddress(address1);
                                    try {
                                            socket.send(packet);
                                    } catch (IOException ex) {

                            Logger.getLogger(MainFrom.class.getName()).log(Level.SEVERE, null, ex);
                                    }
```

```
                packet = new DatagramPacket(buffer,buffer.length);
                packet.setSocketAddress(address2);
                        try {
                                socket.send(packet);
                        } catch (IOException ex) {

                Logger.getLogger(MainFrom.class.getName()).log(Level.SEVERE, null, ex);
                        }
                        st_flag=true;
                socket.close();
                }
                        try {
                        Runtime.getRuntime().exec("C:\\Program Files
(x86)\\VideoLAN\\VLC\\vlc.exe");
                } catch (IOException ex) {
                Logger.getLogger(MainFrom.class.getName()).log(Level.SEVERE, null, ex);
                                }
                        try {
                        Runtime.getRuntime().exec("C:\\Program Files
(x86)\\VideoLAN\\VLC\\vlc.exe");
                } catch (IOException ex) {
                Logger.getLogger(MainFrom.class.getName()).log(Level.SEVERE, null, ex);
                                }
    }

    private void m_stopCheckingActionPerformed(java.awt.event.ActionEvent evt) {
                        //Open all menu
                m_Readfile_Button_1.setEnabled(true);
    m_Stop_Button.setEnabled(false);
                m_Stop_Track_Button.setEnabled(false);
    m_stopChecking.setEnabled(false);
                m_Record_Button_1.setEnabled(true);
    m_RadioButton_green.setEnabled(true);
    m_RadioButton_red.setEnabled(true);
    m_RadioButton_blue.setEnabled(true);
    m_RadioButton_yellow.setEnabled(true);
                m_Track_Button_2.setEnabled(true);
                m_CamCheck.setEnabled(true);
                // Send command '4'--> stop sending image by python program  on raspberry
pi and
```

```
                // Stop python PCrev.py & sudspen display image
                try {
                                //1]Send command '4' -> Stop sending image python
program
                                this.socket = new DatagramSocket(12345);
                        } catch (SocketException ex) {

Logger.getLogger(MainFrom.class.getName()).log(Level.SEVERE, null, ex);
                        }
                String msg ="4";
                byte[] buffer = msg.getBytes();
                DatagramPacket packet = new DatagramPacket(buffer,buffer.length);
                packet.setSocketAddress(address1);
                        try {
                                socket.send(packet);
                        } catch (IOException ex) {

Logger.getLogger(MainFrom.class.getName()).log(Level.SEVERE, null, ex);
                        }
                packet = new DatagramPacket(buffer,buffer.length);
                packet.setSocketAddress(address2);
                        try {
                                socket.send(packet);
                        } catch (IOException ex) {

Logger.getLogger(MainFrom.class.getName()).log(Level.SEVERE, null, ex);
                        }
                        try {
                        Runtime.getRuntime().exec("Taskkill /F /IM vlc.exe");
                } catch (IOException ex) {
                Logger.getLogger(MainFrom.class.getName()).log(Level.SEVERE, null, ex);
                                }
                        st_flag = false;
}

                /**
                 * @param args the command line arguments
                 */
                public static void main(String args[]) throws SocketException {
                        /* Set the Nimbus look and feel */
```

```
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
                                * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
                                */
                                try {
                                        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                                                if ("Nimbus".equals(info.getName())) {

                                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                                                        break;
                                                }
                                        }
                                } catch (ClassNotFoundException ex) {

                                java.util.logging.Logger.getLogger(MainFrom.class.getName()).log(java.util.logging
.Level.SEVERE, null, ex);
                                } catch (InstantiationException ex) {

                                java.util.logging.Logger.getLogger(MainFrom.class.getName()).log(java.util.logging
.Level.SEVERE, null, ex);
                                } catch (IllegalAccessException ex) {

                                java.util.logging.Logger.getLogger(MainFrom.class.getName()).log(java.util.logging
.Level.SEVERE, null, ex);
                                } catch (javax.swing.UnsupportedLookAndFeelException ex) {

                                java.util.logging.Logger.getLogger(MainFrom.class.getName()).log(java.util.logging
.Level.SEVERE, null, ex);
                                }
        //</editor-fold>

        /*  Set runnable for get file to display on screen
                                This part will check the file.jpg on the desire directory
                                  if  cam1.jpg or cam2.jpg exist it will be load to display
                                  on the screen and then deleted. The picture will display only
                                  5 seconds and loop will go on again.
                                 */
```

```
                java.awt.EventQueue.invokeLater(new Runnable() {
                public void run() {
                        try {
                                new MainFrom().setVisible(true);
                        } catch (SocketException ex) {

        Logger.getLogger(MainFrom.class.getName()).log(Level.SEVERE, null, ex);
                        }
                }
            });
        }

    // Variables declaration - do not modify
    private javax.swing.JLabel Label_1;
    private javax.swing.JLabel Label_2;
    private javax.swing.ButtonGroup buttonGroup2;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JButton m_CamCheck;
    private javax.swing.JButton m_Exit_Button;
    private javax.swing.JRadioButton m_RadioButton_blue;
    private javax.swing.JRadioButton m_RadioButton_green;
    private javax.swing.JRadioButton m_RadioButton_red;
    private javax.swing.JRadioButton m_RadioButton_yellow;
    private javax.swing.JButton m_Readfile_Button_1;
    private javax.swing.JButton m_Record_Button_1;
    private javax.swing.JButton m_Stop_Button;
    private javax.swing.JButton m_Stop_Track_Button;
    private javax.swing.JButton m_Track_Button_2;
    private javax.swing.JLabel m_preview_1;
    private javax.swing.JLabel m_preview_2;
    private javax.swing.JButton m_stopChecking;
    // End of variables declaration
                        //static File f;
private boolean st_flag = false;
}
```

รหัสโครงการ/รหัสสัญญา 2559-01-01018

แบบรายงานการใช้จ่ายเงินโครงการวิจัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

รายงานความก้าวหน้า ครั้งที่ 2 รอบ 12 เดือน ประจำปีงบประมาณ .....

☐ แหล่งงบประมาณแผ่นดิน (แบบปกติ) ☑ แหล่งเงินรายได้

ชื่อโครงการ (ภาษาไทย).....ระบบบันทึกภาพอัตโนมัติราคาประหยัด.....

(ภาษาอังกฤษ) .....Automatic low cost picture recording system .....

ชื่อ-สกุลหัวหน้าโครงการวิจัยผู้รับทุน/ผู้วิจัย (อ./ดร./ผศ./รศ./ศ.) รศ.ดร. อรรถสิทธิ์ หล่าสกุล

รายงานในช่วงตั้งแต่วันที่ 1 เมษายน 2559 ถึงวันที่ 30 กันยายน 2559

ระยะเวลาดำเนินการ 1 ปี .....เดือน ตั้งแต่วันที่ 1 ตุลาคม 2557 ถึงวันที่ 30 กันยายน 2558

ข้อมูลการรายงานค่าใช้จ่ายงบประมาณโครงการวิจัย

1. การเบิกจ่ายงบประมาณ (กรณีการจ่ายเงินถ้าจ่ายงวดเดียวให้ลบข้อที่ไม่เกี่ยวข้องออก)

งวดที่ 1 110,500 บาท 85% วันที่ได้รับอนุมัติให้เบิกจ่ายเงิน (ว/ด/ป) 30 พ.ย. 2558

งวดที่ 2 บาท 15% วันที่ได้รับอนุมัติให้เบิกจ่ายเงิน (ว/ด/ป) 30 เมษายน 2559

2. <u>สรุปงบประมาณค่าใช้จ่าย</u>ที่ใช้นับตั้งแต่เริ่มทำการวิจัยถึงปัจจุบัน (จำแนกตามหมวดค่าใช้จ่าย)

หมวดค่าใช้จ่าย	งบประมาณรวมทั้งโครงการ	ค่าใช้จ่าย (บาท)	คงเหลือ (หรือเกิน)
งบดำเนินงาน			
ค่าใช้สอย	3,000	3,000	
ค่าวัสดุ	127,000	127,000	0
ค่าสาธารณูปโภค			
งบลงทุน: ค่าครุภัณฑ์			
รวม	130,000	130,000	0

.....

(.....รศ.ดร.อรรถสิทธิ์ หล่าสกุล)

ลงนามหัวหน้าโครงการวิจัยผู้รับทุน

.....14/09/2559.....

.....

(.....)

ลงนามเจ้าหน้าที่การเงิน/เจ้าหน้าที่ที่เกี่ยวข้อง

...../.....

ข้อมูลประวัติคณะผู้วิจัย

ประวัติส่วนตัว

ชื่อ-สกุล (ภาษาไทย) .....อรรถสิทธิ์ หล่าสกุล.....

ชื่อ-สกุล (ภาษาองกฤษ) .....ATTASIT LASAKUL.....

ตำแน่งทางวิชาการ.....รองศสตราจารย์.....**สัดส่วนการวิจัย**.....100%

ภาควิชา .....สาขาวิชาวิศวกรรมคอมพวเตอร์..... คณะ .....วิศวกรรมศาสตร์.....

โทรศัพท์ ...0840270185..... โทรสาร .....

E-mail .....klattasi@kmitl.ac.th.....

ประวัติการศึกษา

ระดับการศึกษา	ปี ค.ศ. ที่สาเร็จ	สถาบันการศึกษา	วิชาเอก	ชื่อปริญญา
ปริญญาตรี	1987	KMITL	อิเลกทรอนิกส์	B.Ind.Tech
ปริญญาโท	1990	KMITL	ไฟฟ้า	M.Eng
ปริญญเอก	2000	Tokai University	ไฟฟา	D.Eng

สาขาวิจัยที่มีความชำนาญพิเศษ (แตกต่างจากวุฒิการศึกษา).....

การออกแบบวงจรดิจิตอลและการใช้งาน, ระบบดิจิตอล อัตโนมัติ เช่น ไมโครคอนโทลเลอร์

ทุนการศึกษาและทุนวิจัยที่เคยได้รับ

ปี พ.ศ.	ทุนการศึกษาและทุนวิจัย	สถาบันที่ให้
2544	“เครื่องคอบคุมสวิทช์แบบหลายช่องพร้อมกัน”	สกว.
2551	“อุปกรณ์เสริมคย์บอร์ดสำหรับผู้พิการทางสายตา”	พระจอมเกล้าลาดกระบัง
2552	“เครื่องวัดระยะไกลผ่านวิทยุสื่อสาร”	พระจอมเกล้าลาดกระบัง
2553	“เครื่องติดตามยานพาหนะแสดงผลภาษาไทย”	พระจอมเกล้าลาดกระบัง
2554	“เครื่องบันทึกการสอน”	พระจอมเกล้าลาดกระบัง
2554	“เครื่องวัดสิ่งแวดล้อมระยะไกลผ่านวิทยุสื่อสาร”	วช.
2555	“เครื่องวัดสิ่งแวดล้อมระยะไกลอัจฉริยะ”	พระจอมเกล้าลาดกระบัง
2556	“อุปกรณ์ตรวจวัดระดับน้ำสำหรับระบบเตือนภัยน้ำท่วม”	พระจอมเกล้าลาดกระบัง
2556	“ระบบเฝ้าระวังผู้ป่วยแบบไร้สาย”	พระจอมเกล้าลาดกระบัง
2558	“ระบบเฝ้าระวังผู้ป่วยแบบใหม่”	พระจอมเกล้าลาดกระบัง