

ระบบตรวจสอบอัตลักษณ์โดยใช้แสง  
IDENTITY CHECKING SYSTEM BY VISIBLE LIGHT

ธนวิชญ์ อนวงต์พิณีจ  
THANAVIT ANUWONGPINIT  
จิรัฐวุฒิ คำรงค์  
JIRATTAWUT DUMBONG

ปริญญาโทขั้นต้นเป็นงานหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมยานยนต์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2556

ระบบตรวจสอบอัตลักษณ์โดยใช้แสง  
IDENTITY CHECKING SYSTEM BY VISIBLE LIGHT

ธนวิชญ์ อนุวงศ์พินิจ  
THANAVIT ANUWONGPINIT

จิรัฐวุฒิ ดำรงค์  
JIRATTAWUT DUMRONG

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
สาขาวิชาวิศวกรรมสารสนเทศ  
คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2556

# IDENTITY CHECKING SYSTEM BY VISIBLE LIGHT

THANAVIT ANUWONGPINIT

JIRATTAWUT DUMRONG

THIS THESIS IS SUBMITTED IN PARTIAL FULFILLMENT  
OF THE REQUIREMENT FOR THE DEGREE OF  
BACHELOR OF ENGINEERING IN INFORMATION ENGINEERING  
FACULTY OF ENGINEERING  
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG  
ACADEMIC YEAR 2013

หัวข้อปริญญาโท  
รายชื่อนักศึกษา

ระบบตรวจสอบอัตลักษณ์โดยใช้แสง

นายธนวิษฐ์ อนุวงศ์พินิจ

รหัสนักศึกษา 53010665

นายจิรัฐภูมิ ดำรงค์

รหัสนักศึกษา 53010227

ปริญญา

วิศวกรรมศาสตรบัณฑิต

สาขาวิชา

วิศวกรรมสารสนเทศ

พ.ศ.

2556

อาจารย์ที่ปรึกษาปริญญาโท ผศ.บุญยชนะ ภูระหงษ์

ปริญญาโทฉบับนี้ ได้รับการอนุมัติให้เป็นส่วนหนึ่งของการศึกษา ตามหลักสูตรวิศวกรรมศาสตรบัณฑิต คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

(ผศ.บุญยชนะ ภูระหงษ์)

อาจารย์ผู้ควบคุมปริญญาโท

หัวข้อปริญญาานิพนธ์  
รายชื่อนักศึกษา

ระบบตรวจสอบอัตลักษณ์โดยใช้แสง

นายธนวิษฐ์ อนุวงศ์พินิจ

รหัสนักศึกษา 53010665

นายจิรัฐภูมิ ดำรงค์

รหัสนักศึกษา 53010227

ปริญญา  
สาขาวิชา

วิศวกรรมศาสตรบัณฑิต

วิศวกรรมสารสนเทศ

พ.ศ.

2556

อาจารย์ที่ปรึกษาปริญญาานิพนธ์ ผศ.บุญยชนะ ภูระหงษ์

## บทคัดย่อ

ปริญญาานิพนธ์นี้เสนอระบบตรวจสอบอัตลักษณ์โดยใช้แสงซึ่งสามารถนำมาใช้เพื่อทดแทนข้อจำกัดบางอย่างของเทคโนโลยี QRCode และ RFID ซึ่งระบบนี้จะสามารถตรวจสอบได้ในระยะที่ไกลกว่าและสามารถใช้ตรวจสอบงานในตอนกลางคืนและมุมการตรวจจับที่หลากหลายกว่า ซึ่งสามารถนำไปประยุกต์ใช้กับระบบต่างๆได้หลากหลาย ในระบบนี้จะประกอบไปด้วยภาคส่งที่ประกอบด้วยไมโครคอนโทรลเลอร์ทำหน้าที่ควบคุมไฟกระพริบรูปแบบต่างๆกัน โดยการกระพริบรูปแบบหนึ่งจะแทนอัตลักษณ์ของสิ่งๆหนึ่ง และทำการบันทึกรูปแบบการกระพริบนั้นเป็นโค้ดแล้วบันทึกลงฐานข้อมูล และภาครับที่ประกอบด้วยกล้องตรวจจับการกระพริบของแสงด้วยการประมวลผลภาพ เมื่อตรวจจับแล้วจะได้โค้ดของอัตลักษณ์ของสิ่งๆนั้นมาแล้วมาทำการตรวจสอบกับฐานข้อมูล ซึ่งระบบนี้เป็นเทคโนโลยีที่ใหม่ และมีแนวโน้มว่าจะเป็นที่น่าสนใจในอนาคต สามารถนำไปประยุกต์ได้หลากหลายไม่ว่าจะเป็นด้านการรักษาความปลอดภัย งานโฆษณา เป็นต้น

Thesis Title Identity Checking System By Visible Light  
Student Mr.Thanavit Anuwongpinit Student ID. 53010665  
Mr.Jirattawut Dumrong Student ID. 53010227  
Degree Bachelor of Engineering  
Program Information Engineering  
Year 2013  
Thesis Advisor Asst.Prof.Boonchana Purahong

## ABSTRACT

A new identity checking system by visible light can fulfill some limitations of QRCode and RFID technology wherewith this system can check further distance. This paper is a model to be applied to other areas. In this system, the communication between LED Blinking and camera consists of transmitter and receiver. Transmitter is a microcontroller to control LED for blinking pattern one pattern for one identity of object and save in database. Receiver consists of camera to detect LED blinking by image processing. When detecting LED blinking receive code for identity of object then get code to check identity with database. This system can be applied to various such as security, advertising, identify system, and etc.

## กิตติกรรมประกาศ

ปริญญานิพนธ์นี้สามารถสำเร็จลุล่วงไปได้ด้วยดีและสมบูรณ์ โดยความเมตตาและความอนุเคราะห์ของ ผศ.บุญชนะ ภูระหงษ์ ซึ่งเป็นอาจารย์ที่ปรึกษาที่คอยให้คำปรึกษา ให้คำแนะนำ และชี้แนะแนวทางแก้ไขปัญหาต่างๆ เสมอมา

ขอขอบคุณศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติหรือเนคเทค (NECTEC) ที่ได้ไปฝึกงาน ภาคฤดูร้อนและได้รับแนวคิด ความรู้ต่างๆจากที่ฝึกงานนี้มาพัฒนาโครงการจนสำเร็จ

ขอขอบคุณเพื่อนๆในห้องแล็บ 1106 ทุกคน ที่คอยช่วยเหลือ ให้คำปรึกษา แนวทางแก้ไขปัญหาดังกล่าวเกี่ยวกับโครงการนี้

ขอขอบคุณ คณาจารย์ภาควิชาวิศวกรรมสารสนเทศทุกท่านที่ให้ความรู้ คำแนะนำ ในทุกๆเรื่อง

ขอขอบคุณ เพื่อนๆในสาขาวิชาวิศวกรรมสารสนเทศทุกคน ที่คอยช่วยเหลือกัน มีอะไรก็ไม่ทิ้งกัน มีเสียงหัวเราะให้กัน ให้กำลังใจกันมาเสมอ

ขอขอบคุณภาควิชาวิศวกรรมสารสนเทศ และสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่เป็นจุดเริ่มต้นของทุกสิ่ง ที่ทำให้พวกผมมีวันนี้

นายธนวิษณุ อนุวงศ์พินิจ

นายจิรัฐวุฒิ คำรงค์

# สารบัญ

หน้า

สารบัญ.....	I
สารบัญตาราง.....	IV
สารบัญรูป.....	V
บทที่ 1 บทนำ.....	1
1.1 แนวคิดและที่มาของปัญหา.....	1
1.2 จุดประสงค์.....	2
1.3 ขอบเขตโครงการ.....	2
1.4 ผลที่คาดว่าจะได้รับ.....	2
1.5 อุปกรณ์ที่ต้องใช้.....	2
1.6 ขั้นตอนการดำเนินงาน.....	3
บทที่ 2 ทฤษฎีพื้นฐาน.....	4
2.1 บทนำ.....	4
2.2 ทฤษฎีพื้นฐานของการประมวลผลภาพดิจิทัลเบื้องต้น.....	4
2.2.1 รูปภาพดิจิทัล.....	4
2.2.2 การประมวลผลภาพ.....	5
2.2.3 องค์ประกอบของการประมวลผลภาพ.....	6
2.2.4 แบบจำลองสี.....	7
2.2.5 ประเภทของภาพ.....	10
2.3 ไลบรารีที่ใช้ในการพัฒนาโปรแกรมประมวลผลภาพ OpenCV , EmguCV , AForge.NET.....	13
2.3.1 OpenCV.....	13
2.3.2 EmguCV.....	15
2.3.3 AForge.NET.....	17
2.4 แสงและแหล่งกำเนิดแสง.....	18
2.4.1 ธรรมชาติของแสง.....	18
2.4.2 หน่วยที่ใช้ในการวัดแสง.....	18
2.4.3 ความสว่าง.....	19
2.5 หลอดไฟและแหล่งกำเนิดแสง.....	20
2.5.1 หลอดไส้ร้อนแบบธรรมดา.....	20
2.5.2 หลอดฮาโลเจน.....	20
2.5.3 หลอดฟลูออเรสเซนต์.....	21

# สารบัญ (ต่อ)

หน้า

2.5.4 หลอดแอลอีดี.....	22
2.6 ไมโครคอนโทรลเลอร์.....	23
2.6.1 Arduino .....	23
2.6.2 ข้อดีของบอร์ด Arduino .....	24
2.6.3 Arduino UNO.....	24
2.7 Channel Coding (การเข้ารหัสช่องสัญญาณ).....	27
2.7.1 รหัสแบบบล็อก.....	28
2.7.2 รหัสแบบคอนโวลูชัน.....	30
บทที่ 3 โครงสร้างของระบบและการออกแบบ.....	32
3.1 บทนำ.....	32
3.2 โครงสร้างโดยภาพรวมของระบบ.....	32
3.3 การออกแบบในส่วนภาคส่ง.....	33
3.4 การออกแบบในส่วนภาครับ.....	37
3.4.1 การออกแบบอินเตอร์เฟส.....	37
3.4.2 การออกแบบฐานข้อมูล.....	38
3.4.3 การออกแบบโปรแกรมประมวลผลภาพ.....	39
3.5 กล้องเว็บแคม.....	41
3.6 การลดอัตราความผิดพลาด และการเพิ่มอัตราการตรวจจับการกระพิบของแสง.....	41
บทที่ 4 ผลการทดลอง.....	42
4.1 การทดลองในส่วนของภาคส่ง.....	42
4.2 การทดลองในส่วนของภาครับ.....	43
4.3 การทดลองเมื่อตัวรับอยู่ห่างจากตัวส่งในระยะต่างๆในเวลากลางวันและเวลากลางคืน.....	44
4.4 การลดอัตราความผิดพลาด และการเพิ่มอัตราการตรวจจับไฟกระพิบของแสง.....	50
4.4.1 การลดความผิดพลาดโดยใช้การเข้ารหัสแบบ Block Code.....	50
4.4.2 การลดความผิดพลาดโดยใช้การเข้ารหัสแบบ Convolutional Code.....	52
4.4.3 การเพิ่มอัตราการตรวจจับโดยใช้เทคนิคการซูมภาพ.....	55
4.4.4 การเพิ่มอัตราการตรวจจับโดยใช้เทคนิคการลดกรอบการประมวลผล.....	58
4.4.5 การลดความผิดพลาดโดยการลดจำนวนบิตที่ส่ง.....	63
4.4.6 การเพิ่มอัตราการตรวจจับโดยการเพิ่มขนาดของตัวส่ง.....	67
4.4.7 การเพิ่มอัตราการตรวจจับโดยทดลองกับหลอดแอลอีดีสีต่างๆ.....	69

## สารบัญ (ต่อ)

	หน้า
4.4.8 การลดความผิดพลาดโดยใช้การตรวจสอบจากการตรวจจับสองครั้ง.....	72
4.5 การจัดการฐานข้อมูลของระบบ .....	73
4.5.1 การเพิ่มข้อมูลลงในฐานข้อมูล.....	73
4.5.2 การลบข้อมูลในฐานข้อมูล.....	74
4.6 การนำไปประยุกต์ใช้ .....	76
บทที่ 5 สรุปและวิจารณ์ผลการทดลอง.....	77
5.1 บทสรุปโครงการ.....	77
5.2 ปัญหาที่พบระหว่างการดำเนินโครงการ.....	77
5.3 แนวทางการแก้ไข.....	77
5.4 การพัฒนาต่อและการนำไปใช้ .....	78
บรรณานุกรม .....	79
ภาคผนวก.....	80
ภาคผนวก ก. โปรแกรมตรวจสอบอัตลักษณ์ด้วยแสงในส่วนของภาคส่ง .....	81
ภาคผนวก ข. โปรแกรมตรวจสอบอัตลักษณ์ด้วยแสงในส่วนของภาครับ .....	88
ภาคผนวก ค. วารสารทางวิชาการที่ได้รับการตีพิมพ์ในการประชุมทางวิชาการ.....	100
ภาคผนวก ง. ข้อมูลและคุณสมบัติต่างๆของอุปกรณ์.....	106

# สารบัญตาราง

ตารางที่	หน้า
1. เปรียบเทียบการทำงานของ EmguCV ในแต่ละเวอร์ชัน.....	15
2. รายละเอียด Arduino UNO.....	24
3. ตัวอย่างข้อมูลในฐานข้อมูลที่ใช้งาน.....	38
4.1 ผลการทดลองการตรวจจับที่ระยะต่างๆในเวลากลางวัน.....	47
4.2 ผลการทดลองการตรวจจับที่ระยะต่างๆในเวลากลางคืน.....	48
4.3 ผลการทดลองการตรวจจับที่ระยะต่างๆในเวลากลางวันโดยใช้การเข้ารหัสแบบ Block Code.....	50
4.4 ผลการทดลองการตรวจจับที่ระยะต่างๆในเวลากลางคืนโดยใช้การเข้ารหัสแบบ Block Code.....	51
4.5 ผลการทดลองการตรวจจับที่ระยะต่างๆในเวลากลางวันโดยใช้การเข้ารหัสแบบ Convolutional Code.....	52
4.6 ผลการทดลองการตรวจจับที่ระยะต่างๆในเวลากลางคืนโดยใช้การเข้ารหัสแบบ Convolutional Code.....	53
4.7 ผลการทดลองการตรวจจับที่ระยะต่างๆในเวลากลางวันโดยใช้เทคนิคการซูมภาพ.....	55
4.8 ผลการทดลองการตรวจจับที่ระยะต่างๆในเวลากลางคืนโดยใช้เทคนิคการซูมภาพ.....	56
4.9 ผลการทดลองการตรวจจับที่ระยะต่างๆในเวลากลางวันโดยใช้เทคนิคการลดกรอบการประมวลผล.....	59
4.10 ผลการทดลองการตรวจจับที่ระยะต่างๆในเวลากลางคืนโดยใช้เทคนิคการลดกรอบการประมวลผล.....	59
4.11 ผลการทดลองการตรวจจับที่ระยะต่างๆในเวลากลางวันโดยใช้เทคนิคการซูมภาพและลดกรอบฯ.....	61
4.12 ผลการทดลองการตรวจจับที่ระยะต่างๆในเวลากลางคืนโดยใช้เทคนิคการซูมภาพและลดกรอบฯ.....	61
4.13 ผลการทดลองการตรวจจับที่ระยะต่างๆในเวลากลางวันโดยลดจำนวนบิตที่ส่งเป็น 21 บิต.....	63
4.14 ผลการทดลองการตรวจจับที่ระยะต่างๆในเวลากลางคืนโดยลดจำนวนบิตที่ส่งเป็น 21 บิต.....	63
4.15 ผลการทดลองการตรวจจับที่ระยะต่างๆในเวลากลางวันโดยลดจำนวนบิตที่ส่งเป็น 16 บิต.....	65
4.16 ผลการทดลองการตรวจจับที่ระยะต่างๆในเวลากลางคืนโดยลดจำนวนบิตที่ส่งเป็น 16 บิต.....	65
4.17 ผลการทดลองการตรวจจับที่ระยะต่างๆในเวลากลางวันโดยการเพิ่มขนาดของตัวส่ง.....	67
4.18 ผลการทดลองการตรวจจับที่ระยะต่างๆในเวลากลางคืนโดยการเพิ่มขนาดของตัวส่ง.....	67
4.19 ผลการทดลองการตรวจจับที่ระยะต่างๆในเวลากลางวันโดยใช้ LED สีแดง.....	69
4.20 ผลการทดลองการตรวจจับที่ระยะต่างๆในเวลากลางคืนโดยใช้ LED สีแดง.....	70
4.21 ผลการทดลองการตรวจจับที่ระยะต่างๆในเวลากลางวันโดยใช้ LED สีม่วง.....	70
4.22 ผลการทดลองการตรวจจับที่ระยะต่างๆในเวลากลางคืนโดยใช้ LED สีม่วง.....	70
4.23 ผลการทดลองการตรวจจับที่ระยะต่างๆในเวลากลางวันโดยใช้ LED สีขาว Super Bright.....	71
4.24 ผลการทดลองการตรวจจับที่ระยะต่างๆในเวลากลางคืนโดยใช้ LED สีขาว Super Bright.....	71

# สารบัญรูป

รูปที่	หน้า
1 การทำงานของระบบ.....	1
2.1 ภาพบิตแมปและภาพเวกเตอร์.....	4
2.2 การกระทำการกับภาพแบบจุด.....	5
2.3 การกระทำการกับภาพแบบบริเวณ.....	6
2.4 องค์ประกอบของการประมวลผลภาพ.....	6
2.5 วงล้อของระบบสีแบบ RGB.....	8
2.6 ระบบสีแบบ YUV.....	8
2.7 ระบบสีแบบ YIQ.....	9
2.8 ระบบสีแบบ YCbCr.....	10
2.9 ภาพระดับความเข้มเทา.....	10
2.10 ภาพสี.....	11
2.11 ภาพไบนารี.....	11
2.12 ภาพดัชนี.....	12
2.13 โลโก้ OpenCV.....	13
2.14 โฟลว์ชาร์ตการทำงานโดยใช้ไลบรารีของ OpenCV.....	14
2.15 โลโก้ EmguCV.....	15
2.16 สถาปัตยกรรมโดยรวมของ EmguCV.....	16
2.17 สัญลักษณ์ของ AForge.NET.....	17
2.18 ค่าย่านความถี่ของคลื่นแม่เหล็กไฟฟ้ารวมไปถึงแสง.....	18
2.19 หลอดไส้ร้อนแบบธรรมดา.....	20
2.20 หลอดไฟแบบฮาโลเจน.....	21
2.21 หลอดแบบฟลูออเรสเซนต์.....	21
2.22 หลอดแบบแอลอีดี.....	22
2.23 บอร์ด Arduino แบบต่างๆที่นิยมใช้.....	23
2.24 บอร์ด Arduino UNO R3.....	25
2.25 ขาพอร์ตของ ATmega328.....	25
2.26 การเข้ารหัสแบบ Convolutional Code.....	30
2.27 Trellis Diagram ที่ K=3.....	31
2.23 State Diagram ที่ K=3.....	31
3.1 โครงสร้างโดยภาพรวมของระบบ.....	32
3.2 วงจรจำลองหลอดไฟ LED จำนวน 49 หลอด.....	33
3.3 หลอดไฟ LED ขนาด 7x7 จำนวน 49 หลอด.....	33
3.4 จำลองวงจรสวิตช์เปิดปิดไฟเชื่อมต่อเข้ากับบอร์ด Arduino UNO.....	34
3.5 วงจรสวิตช์เปิดปิดไฟเชื่อมต่อเข้ากับบอร์ด Arduino UNO.....	34

## สารบัญรูป (ต่อ)

รูปที่	หน้า
3.6 โพล์ชาร์ตการทำงานของโปรแกรมไพเราะพริบ	36
3.7 อินเตอร์เฟสของโปรแกรมโดยใช้โปรแกรม C#	37
3.8 แผนภาพ ER Diagram ฐานข้อมูลของระบบตรวจสอบความเป็นอัตลักษณ์	38
3.9 โพล์ชาร์ตการทำงานของโปรแกรมประมวลผลภาพ	39
3.10 แผนภาพหลักการประมวลผล	40
3.11 การถอดรหัสบีทข้อมูล	40
3.12 กล้องเว็บแคมรุ่น OKER 177 ที่นำไปใช้ในการตรวจจับการกระพริบของแสง	41
4.1 วงจรในส่วนของภาคส่ง	42
4.2 โปรแกรม Arduino 1.0.5	42
4.3 การทำงานของโปรแกรมตรวจสอบอัตลักษณ์โดยใช้แสงที่เขียนโดย C#	43
4.4 การเลือกกล้องที่จะใช้งานและแสดงผลภาพที่กล้องตรวจจับได้	43
4.5 การรับข้อมูลจากการตรวจจับการกระพริบของแสง	44
4.6 การตรวจสอบข้อมูลจากฐานข้อมูล	44
4.7 ตัวส่งที่ระยะห่างจากตัวรับในระยะ 1 เมตรและ 5 เมตรในตอนกลางวัน	45
4.8 ตัวส่งที่ระยะห่างจากตัวรับในระยะ 10 เมตรและ 20 เมตรในตอนกลางวัน	45
4.9 ตัวส่งที่ระยะห่างจากตัวรับในระยะ 30 เมตรและ 40 เมตรในตอนกลางวัน	45
4.10 ตัวส่งที่ระยะห่างจากตัวรับในระยะ 50 เมตรในตอนกลางวัน	46
4.11 ตัวส่งที่ระยะห่างจากตัวรับในระยะ 1 เมตรและ 5 เมตรในตอนกลางคืน	46
4.12 ตัวส่งที่ระยะห่างจากตัวรับในระยะ 10 เมตรและ 20 เมตรในตอนกลางคืน	46
4.13 ตัวส่งที่ระยะห่างจากตัวรับในระยะ 30 เมตรและ 40 เมตรในตอนกลางคืน	47
4.14 ตัวส่งที่ระยะห่างจากตัวรับในระยะ 50 เมตรในตอนกลางคืน	47
4.15 กราฟความสัมพันธ์ระหว่างร้อยละการรับข้อมูลถูกต้องกับระยะจากเครื่องรับเวลากลางวันและกลางคืน	48
4.16 กราฟความสัมพันธ์ระหว่างร้อยละการไม่สามารถรับข้อมูลได้ถูกต้องกับระยะจากเครื่องรับในเวลากลางวันและกลางคืน	48
4.17 กราฟความสัมพันธ์ระหว่างร้อยละการรับข้อมูลผิดพลาดกับระยะจากเครื่องรับเวลากลางวันและกลางคืน	49
4.18 กราฟความสัมพันธ์ระหว่างร้อยละการรับข้อมูลผิดพลาดกับระยะจากเครื่องรับเวลากลางวันและกลางคืน	51
4.19 กราฟความสัมพันธ์ระหว่างร้อยละการรับข้อมูลผิดพลาดกับระยะจากเครื่องรับเวลากลางวันและกลางคืนโดยใช้การเข้ารหัสแบบ Block Code	52
4.20 กราฟความสัมพันธ์ระหว่างร้อยละการรับข้อมูลผิดพลาดกับระยะจากเครื่องรับเวลากลางวันและกลางคืน	53
4.21 กราฟความสัมพันธ์ระหว่างร้อยละการรับข้อมูลผิดพลาดกับระยะจากเครื่องรับเวลากลางวันและกลางคืนโดยใช้การเข้ารหัสแบบ Convolutional Code	54
4.22 การตรวจจับเมื่อยังไม่ได้คุณภาพ	55
4.23 การตรวจจับเมื่อทำการซูมภาพ	55
4.24 กราฟความสัมพันธ์ระหว่างร้อยละการรับข้อมูลถูกต้องกับระยะจากเครื่องรับเวลากลางวันและกลางคืน	56

## สารบัญรูป (ต่อ)

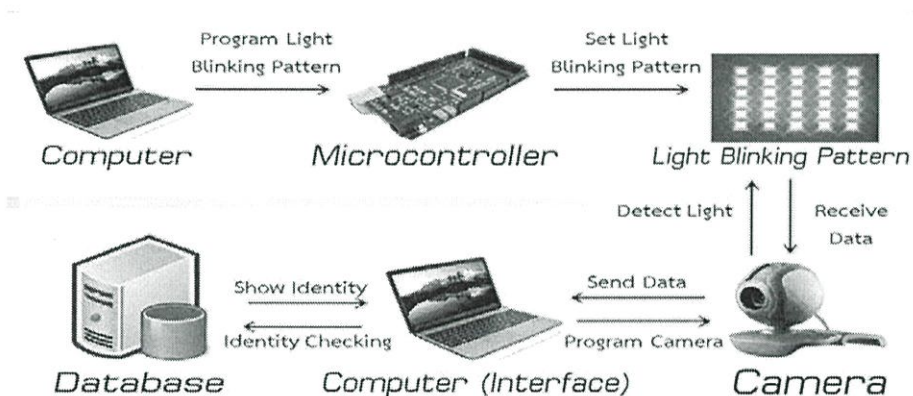
รูปที่	หน้า
4.25 กราฟความสัมพันธ์ระหว่างร้อยละการรับข้อมูลได้ถูกต้องกับระยะจากเครื่องรับในเวลากลางวันและกลางคืน โดยใช้เทคนิคการซูมภาพ .....	57
4.26 กรอบการประมวลผลที่ขนาดต่างๆ .....	58
4.27 กราฟความสัมพันธ์ระหว่างร้อยละการรับข้อมูลถูกต้องกับระยะจากเครื่องรับเวลากลางวันและกลางคืน.....	60
4.28 กราฟความสัมพันธ์ระหว่างร้อยละการรับข้อมูลได้ถูกต้องกับระยะจากเครื่องรับในเวลากลางวันและกลางคืน โดยใช้เทคนิคการลดกรอบการประมวลผล .....	60
4.29 กราฟความสัมพันธ์ระหว่างร้อยละการรับข้อมูลถูกต้องกับระยะจากเครื่องรับเวลากลางวันและกลางคืน.....	62
4.30 กราฟความสัมพันธ์ระหว่างร้อยละการรับข้อมูลได้ถูกต้องกับระยะจากเครื่องรับในเวลากลางวันและกลางคืน โดยใช้เทคนิคการซูมภาพและการลดกรอบการประมวลผล .....	62
4.31 กราฟความสัมพันธ์ระหว่างร้อยละการรับข้อมูลผิดพลาดกับระยะจากเครื่องรับเวลากลางวันและกลางคืน....	64
4.32 กราฟความสัมพันธ์ระหว่างร้อยละการรับข้อมูลผิดพลาดกับระยะจากเครื่องรับเวลากลางวันและกลางคืนโดย ใช้จำนวนบิตที่ส่ง 21 บิต .....	64
4.33 กราฟความสัมพันธ์ระหว่างร้อยละการรับข้อมูลผิดพลาดกับระยะจากเครื่องรับเวลากลางวันและกลางคืน....	66
4.34 กราฟความสัมพันธ์ระหว่างร้อยละการรับข้อมูลผิดพลาดกับระยะจากเครื่องรับเวลากลางวันและกลางคืนโดย ใช้จำนวนบิตที่ส่ง 21 บิต .....	66
4.35 กราฟความสัมพันธ์ระหว่างร้อยละการรับข้อมูลถูกต้องกับระยะจากเครื่องรับเวลากลางวันและกลางคืน.....	68
4.36 กราฟความสัมพันธ์ระหว่างร้อยละการรับข้อมูลได้ถูกต้องกับระยะจากเครื่องรับในเวลากลางวันและกลางคืน โดยการเพิ่มขนาดของตัวส่ง.....	68
4.37 LED สีต่างๆที่ใช้ในการทดลอง .....	69
4.38 GUI ส่วนฐานข้อมูลและการจัดการฐานข้อมูลของโปรแกรม .....	73
4.39 GUI ส่วนการเพิ่มข้อมูล .....	73
4.40 GUI ส่วนการเพิ่มข้อมูลเมื่อเพิ่มข้อมูลสำเร็จ .....	74
4.41 GUI ส่วนการลบข้อมูล.....	74
4.42 GUI ส่วนการลบข้อมูลโดยเลือกข้อมูลที่ต้องการจะลบ .....	75
4.43 GUI ยืนยันการลบข้อมูล.....	75
4.44 GUI แสดงการลบข้อมูลหายไปเรียบร้อยแล้ว.....	76

# บทที่ 1

## บทนำ

### 1.1 แนวคิดและความสำคัญของปัญหา

ในปัจจุบัน ระบบตรวจสอบความเป็นอัตลักษณ์เพื่อระบุข้อมูลของสิ่งต่างๆ ซึ่งทำงานได้ทั้งแบบอัตโนมัติและแบบสั่งการได้ถูกนำมาใช้อย่างแพร่หลายในปัจจุบัน เช่น เทคโนโลยี RFID ที่นำไปใช้กับรถยนต์เพื่อระบุลักษณะรถยนต์หรือผู้ขับขี่ หรือระบบการสแกน QRCode เพื่อตรวจสอบข้อมูลของสิ่งที่ต้องการจะทราบ เช่น ข้อมูลสินค้า เป็นต้น แต่ระบบทั้งสองที่ยกตัวอย่างมานั้นยังมีข้อจำกัดที่สามารถตรวจสอบข้อมูลได้แค่ในระยะใกล้ และหากวิธีตรวจสอบทุกๆ ไปในบริเวณที่มีแสงน้อยก็จะตรวจสอบได้โดยยาก จึงได้มีการศึกษาค้นคว้าสิ่งที่มาแก้ปัญหาลักษณะนี้ โดยใช้กล้องในการตรวจสอบการกระพริบของแสงแทน ซึ่งในโครงการงาน “ระบบตรวจสอบความเป็นอัตลักษณ์ด้วยแสง” จะมีรูปแบบการทำงานตามรูปที่ 1 ดังนี้



รูปที่ 1 การทำงานของระบบ

ในระบบจะใช้ไมโครคอนโทรลเลอร์ซึ่งจะเขียนโปรแกรมในการสร้างรูปแบบการกระพริบของแสง แล้วบันทึกลงในฐานข้อมูล โดยการกระพริบของแสงแต่ละรูปแบบซึ่งจะมีรูปแบบการกระพริบไม่เหมือนกันตามอัตลักษณ์ของข้อมูลแต่ละอย่าง แล้วใช้กล้องโดยเขียนโปรแกรมเพื่อตรวจสอบรูปแบบการกระพริบของแสงว่ารูปแบบการกระพริบนั้นตรงกับข้อมูลในฐานข้อมูลหรือไม่ โดยมีโปรแกรมที่มีอินเตอร์เฟซสำหรับตรวจสอบ

การตรวจสอบความเป็นอัตลักษณ์ด้วยแสงยังสามารถนำไปประยุกต์ใช้ได้อีกหลากหลาย เช่น ในเชิงพาณิชย์ ในอนาคตอาจจะสามารถนำ Smartphone แล้วเปิดแอปพลิเคชันแล้วตรวจสอบหันไปยังป้ายโฆษณาสินค้าที่มีไฟกระพริบอยู่ ก็จะปรากฏข้อมูลสินค้าหรือนำทางไปยังเว็บไซต์ของสินค้านั้นได้อย่างอัตโนมัติ ซึ่งสามารถตรวจสอบได้จากระยะไกล และตรวจสอบได้ที่ละหลายๆ อย่างได้ และในเวลากลางคืนหากมีแสงสว่างก็สามารถตรวจสอบได้โดยง่าย ดังนั้นเทคโนโลยีการตรวจสอบอัตลักษณ์ของข้อมูลด้วยแสงนี้จึงน่าสนใจที่จะศึกษา เพราะเป็นเทคโนโลยีที่ใหม่ และมีแนวโน้มว่าจะเป็นเทคโนโลยีที่น่าสนใจ และอาจจะได้รับความนิยมในอนาคต

## 1.2 จุดประสงค์

1. เพื่อศึกษาหลักการการทำงานในการตรวจสอบความเป็นอัตลักษณ์ด้วยแสง
2. เพื่อศึกษาการประมวลผลภาพโดยใช้กล้องในการตรวจสอบรูปแบบการกระพริบของแสง
3. เพื่อแก้ปัญหาข้อจำกัดบางอย่างของระบบตรวจสอบอัตลักษณ์ที่มีอยู่ในปัจจุบัน
4. เพื่อสร้างความสะดวกสบายให้แก่มนุษย์ ในการตรวจสอบอัตลักษณ์ของสิ่งต่างๆ
5. เพื่อเพิ่มแนวทางในการตรวจสอบอัตลักษณ์ให้มีทางเลือกในการใช้ตรวจสอบมากขึ้น

## 1.3 ขอบเขตของโครงการ

1. มีระบบที่สามารถสร้างรูปแบบการกระพริบของแสงที่แตกต่างกันได้
2. ใช้กล้องเพื่อตรวจสอบอัตลักษณ์ข้อมูลจากการกระพริบของแสงที่แตกต่างกันได้
3. มีโปรแกรมที่มีอินเตอร์เฟซสำหรับแก้ไข เพิ่มหรือลบข้อมูลในฐานข้อมูลได้
4. เมื่อตรวจสอบการกระพริบของแสงแล้วสามารถแสดงในโปรแกรมได้ว่ามีข้อมูลตรงกับฐานข้อมูลหรือไม่

## 1.4 ผลที่คาดว่าจะได้รับ

1. ระบบที่พัฒนาขึ้น สามารถนำไปใช้ในการตรวจสอบความเป็นอัตลักษณ์ได้จริง
2. ระบบที่พัฒนาขึ้น สามารถเก็บข้อมูลแต่ละอย่างที่มีรูปแบบการกระพริบของแสงที่ต่างกัน
3. ได้รับความรู้เพิ่มเกี่ยวกับการตรวจสอบอัตลักษณ์ด้วยแสงและเทคโนโลยีการประมวลผลภาพ
4. ระบบที่ทำการศึกษา สามารถนำความรู้ไปประยุกต์ใช้ในด้านอื่นๆได้
5. ระบบที่พัฒนาขึ้นช่วยสร้างความสะดวกสบายให้แก่มนุษย์มากขึ้น

## 1.5 อุปกรณ์ที่ต้องใช้

### ฮาร์ดแวร์

- |  |                 |
|--|-----------------|
| - เครื่องคอมพิวเตอร์สำหรับพัฒนาโปรแกรมและแสดงข้อมูล    | จำนวน 1 เครื่อง |
| - เครื่องคอมพิวเตอร์สำหรับเป็นเซิร์ฟเวอร์ระบบฐานข้อมูล | จำนวน 1 เครื่อง |
| - ไมโครคอนโทรลเลอร์สำหรับควบคุมการกระพริบของแสง        | จำนวน 1 ชุด     |
| - ไฟ LED สำหรับใช้เป็นต้นแบบในการกระพริบของแสง         | จำนวน 1 ชุด     |
| - กล้องสำหรับตรวจจับการกระพริบของแสง                   | จำนวน 1 เครื่อง |

### ซอฟต์แวร์

- OpenCV, EmguCV และ AForge.net เป็นไลบรารีที่ใช้พัฒนาการประมวลผลภาพ
- Visual Studio เป็นโปรแกรมที่ใช้ภาษา C หรือ C# ในการเขียนโปรแกรม
- Adobe Photoshop สำหรับสร้างรูปภาพ เพื่อตกแต่งโปรแกรมและอินเตอร์เฟซ
- MySQL สำหรับสร้างและจัดการฐานข้อมูลจัดการฐานข้อมูล

## 1.6 ขั้นตอนการดำเนินงาน

No	การดำเนินงาน	2013						2014			
		มี.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.
1	เสนอโครงการ	_____									
2	ศึกษาปัญหา เก็บรวบรวมข้อมูลและค้นคว้าวิจัยข้อมูล	_____	_____	_____							
3	วิเคราะห์และออกแบบระบบ		_____	_____	_____	_____					
4	ออกแบบระบบกระพริบของไฟ		_____	_____	_____	_____					
5	ออกแบบระบบการทำงานของกล้อง			_____	_____	_____					
6	ออกแบบฐานข้อมูลและอินเทอร์เน็ตเฟสผู้ใช้งาน			_____	_____	_____					
7	ดำเนินการทำโครงการ					_____	_____	_____	_____		
8	โมดูลภาคส่ง : ระบบรูปแบบกระพริบของไฟ					_____	_____				
9	โมดูลภาครับ : ระบบการทำงานของกล้องตรวจรูปแบบการกระพริบ						_____	_____			
10	ฐานข้อมูล							_____	_____		
11	อินเทอร์เน็ตเฟสผู้ใช้งาน							_____	_____		
12	ทดสอบระบบ หาปัญหาและหาวิธีการแก้ไข					_____	_____	_____	_____		

## บทที่ 2 ทฤษฎีพื้นฐานที่ใช้

### 2.1 บทนำ

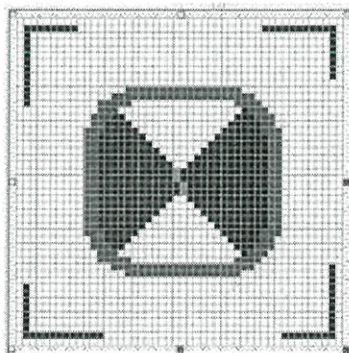
ในระบบการตรวจสอบความเป็นอัตลักษณ์นั้น ปัจจุบันมีเทคโนโลยีมากมายที่นำมาใช้ในปัจจุบันเช่น RFID, NFC และ QR Code เป็นต้น เนื่องจากเป็นอุปกรณ์ที่ติดตั้งง่าย และใช้งานกันอย่างแพร่หลาย แต่ด้วยข้อจำกัดในเรื่องของระยะทางที่สามารถจับได้เพียงระยะที่ตัวส่งกับตัวรับต้องอยู่ไม่ไกลกันเท่านั้น และใช้งานยากในตอนกลางคืน

จึงได้มีการคิดค้นโครงการนี้ขึ้นมาเพื่อเป็นอีกทางเลือกในการตรวจสอบความเป็นอัตลักษณ์ โดยใช้รูปแบบการกระพริบของแสงเป็นข้อมูลแล้วใช้การประมวลผลภาพตรวจจับรูปแบบการกระพริบนั้นเพื่อตรวจสอบความเป็นอัตลักษณ์ของข้อมูล ดังจะกล่าวถึงหลักการต่างๆในหัวข้อถัดไป

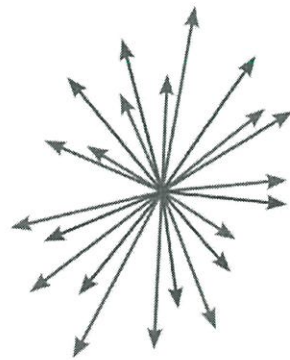
### 2.2 ทฤษฎีพื้นฐานของการประมวลผลภาพดิจิทัลเบื้องต้น

#### 2.2.1 รูปภาพดิจิทัล

โดยปกติแล้ว ข้อมูลภาพต่างๆไปนั้นได้มาจากการที่แสงตกกระทบกับวัตถุแล้วเกิดการสะท้อนผ่านเข้าเลนส์เข้าสู่ตัวบันทึกภาพอาจจะอยู่ในรูปของเซ็นเซอร์ หรือฟิล์ม โดยทั่วไปแล้วสามารถที่จะแบ่งรูปภาพที่ปรากฏและใช้งานบนเครื่องคอมพิวเตอร์ออกเป็นสองประเภท คือ ภาพแบบบิตแมป (Bitmap Image) และ ภาพแบบเวกเตอร์ (Vector Image) โดยรูปภาพแบบบิตแมปจะพิจารณาตัวรูปภาพซึ่งถูกแบ่งออกเป็นส่วนย่อยๆหลายๆส่วนที่เรียกว่าพิกเซล (Pixel) ซึ่งถูกนำมารวมกันและใช้แสดงผลภาพ ส่วนรูปภาพแบบเวกเตอร์จะประกอบด้วยเส้นลายต่างๆที่สร้างขึ้นจากการคำนวณทางคณิตศาสตร์ของลักษณะทางเรขาคณิตเพื่อสร้างรูปทรงต่างๆให้เห็น



(ก) ภาพแบบบิตแมป



(ข) ภาพแบบเวกเตอร์

รูปที่ 2.1 ภาพแบบบิตแมปและภาพแบบเวกเตอร์ [22]

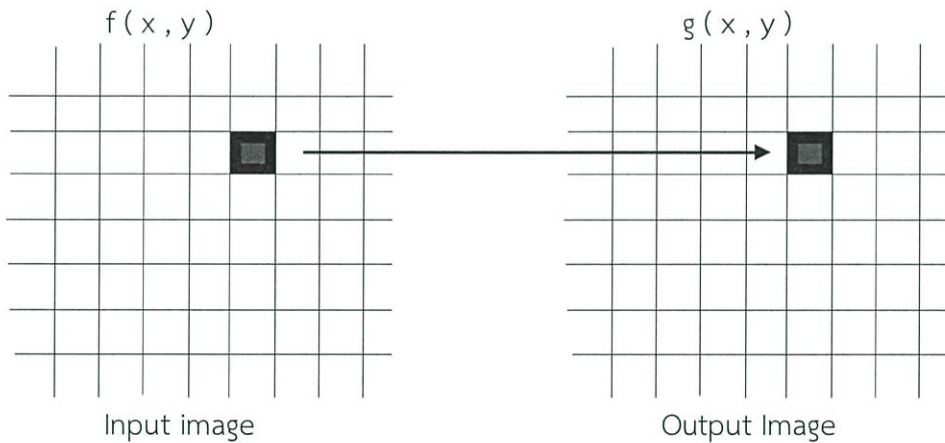
### 2.2.2 การประมวลผลภาพ ( Image Processing )

การประมวลผลภาพ คือ การกระทำอย่างใดอย่างหนึ่งกับภาพต้นฉบับ (Input Image) เพื่อให้ได้ภาพผลลัพธ์ (Output Image) มีลักษณะของภาพเป็นไปตามที่ต้องการ เช่น การปรับภาพให้คมชัดขึ้น ให้ดูดีขึ้น การรู้จำภาพ การแยกแยะบริเวณในภาพ เป็นต้น การกระทำกับภาพในการประมวลผลภาพดิจิทัลสามารถแบ่งได้ออกเป็นสองประเภทคือ การประมวลผลภาพแบบจุด (Point Image Processing) และ การประมวลผลภาพแบบบริเวณ (Local Image Processing)

การประมวลผลภาพแบบจุด (Point Image Processing) เป็นวิธีการกระทำกับภาพต้นฉบับที่ค่าระดับความเข้มเทาในแต่ละพิกเซลของภาพผลลัพธ์จะขึ้นอยู่กับค่าระดับความเข้มเทาของพิกเซลในภาพต้นฉบับพิกเซลต่อพิกเซลที่ตำแหน่งสมนัยกัน โดยที่ค่าเปลี่ยนแปลงของภาพผลลัพธ์ไม่ได้ขึ้นกับค่าพิกเซลที่อยู่บริเวณใกล้เคียงของภาพต้นแบบ กำหนดให้  $f(x,y)$  และ  $g(x,y)$  เป็นภาพต้นฉบับและภาพผลลัพธ์ตามลำดับ สมการที่ใช้ในการประมวลผลของภาพแบบจุดต่อจุดสามารถแสดงได้ดังสมการที่ (1.1)

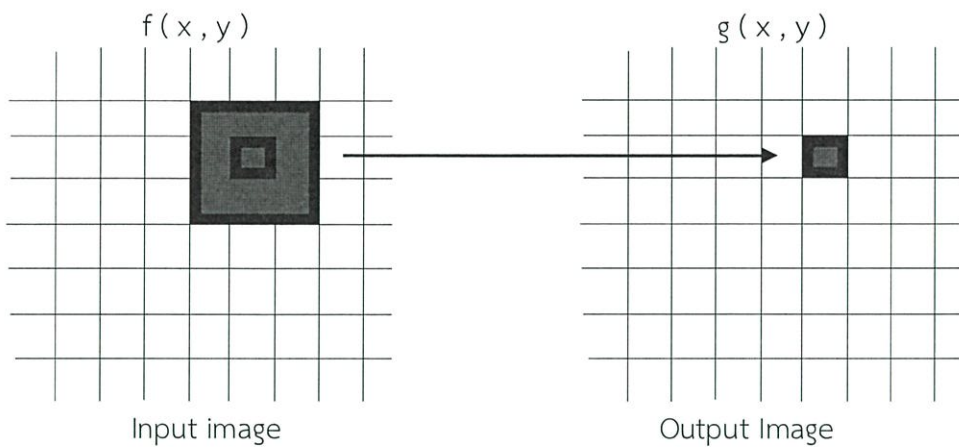
$$g(x,y) = M[f(x,y)] \quad (1.1)$$

เมื่อ  $M[ ]$  เป็นการกระทำกับภาพแบบจุดใดๆหรือเป็นการแทนข้อมูลภาพด้วย Mapping Function ใดๆ โดยค่าระดับความเข้มเทาใหม่ที่ได้ของแต่ละพิกเซลของภาพจะถูกแทนที่ในค่าของแต่ละพิกเซลของภาพต้นฉบับที่พิกัด  $(x,y)$  เดิมของภาพ



รูปที่ 2.2 การกระทำกับภาพแบบจุด ( Point Image Processing )

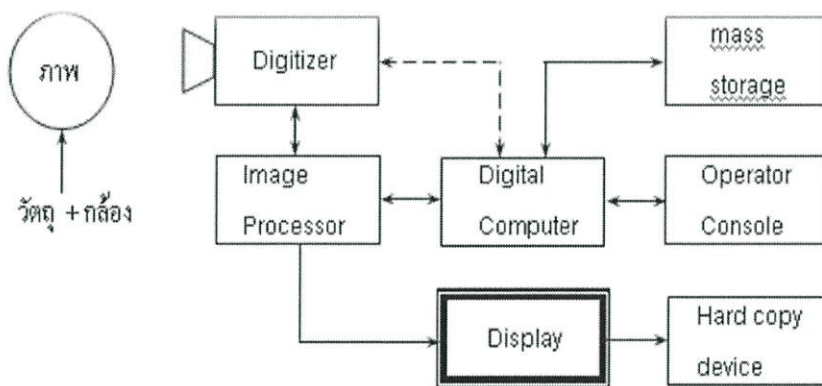
การประมวลผลภาพแบบบริเวณ ( Local Image Processing ) เป็นกระบวนการกระทำกับภาพแบบบริเวณนี้ ค่าระดับความเข้มเทาของพิกเซลในแต่ละจุดในภาพผลลัพธ์จะขึ้นอยู่กับค่าระดับความเข้มเทาของกลุ่มพิกเซลที่อยู่ในบริเวณข้างเคียงกันของภาพต้นฉบับ ตัวอย่างของการประมวลผลภาพทางดิจิทัลแบบนี้ ได้แก่ การกรองสัญญาณภาพในสเปเชียลโดเมน หรือที่เรียกกันว่าการคอนโวลูชัน



รูปที่ 2.3 การกระทำการกับภาพแบบบริเวณ ( Local Image Processing )

### 2.2.3 องค์ประกอบของการประมวลผลภาพ

องค์ประกอบของการประมวลผลภาพแบบดิจิทัลจะประกอบไปด้วยส่วนต่างๆดังรูปที่ 2.4 ซึ่งมีดังนี้



รูปที่ 2.4 องค์ประกอบของการประมวลผลภาพ

- 1) หน่วยประมวลผลภาพ ( Image Processor ) ซึ่งเป็นหัวใจของระบบประกอบไปด้วยฮาร์ดแวร์โมดูลทำงานหลัก 4 อย่าง คือ
  - 1.1) Image acquisition รับสัญญาณวิดีโอเปลี่ยนเป็นข้อมูลดิจิทัลในเชิงตำแหน่งและแอมพลิจูด digitize ภาพทีวินาทีใน 1 เฟรม-เวลา (1/30 ของวินาที ) โมดูลที่ใช้เรียกว่า frame grabber
  - 1.2) Storage module – frame buffer หน่วยความจำสำหรับเก็บข้อมูลทั้งภาพ อาจมีหลายโมดูลที่สามารถเก็บและอ่านภาพที่อัตราของวิดีโอ คือ 30 ภาพ/วินาที
  - 1.3) Low-level (Fast) Processing ALU (Arithmetic-Logic Unit) คำนวณตัวเลขและลอจิกกับพิกเซลแบบขนาน ทำให้การประมวลผลเร็วขึ้น
  - 1.4) Display อ่านข้อมูลภาพจากหน่วยความจำเปลี่ยนเป็นสัญญาณภาพแบบอนาล็อกแสดงบนจอภาพ อาจมีฟังก์ชันเพิ่ม

- 2) Digitizer ทำหน้าที่เปลี่ยนภาพเป็นข้อมูลตัวเลขเข้าคอมพิวเตอร์ เช่น Scanner , Solid-state arrays เป็นต้น
- 3) Digital Computer ชนิดที่ใช้ขึ้นกับขนาดของงานอาจเป็นเพียง Microprocessor device จนถึงคอมพิวเตอร์ขนาดใหญ่ อาจเป็นส่วนหนึ่งหรือเป็นอินเทอร์เฟซกับคอมพิวเตอร์
- 4) Storage device ข้อมูลภาพใช้หน่วยความจำมาก ต้องใช้อุปกรณ์ที่มีความจำขนาดใหญ่เพื่อใช้เก็บข้อมูลภาพ
- 5) Display & Recording Devices อุปกรณ์แสดงผลจอภาพ อาจใช้จอภาพสีเดียว หรือจอภาพสี

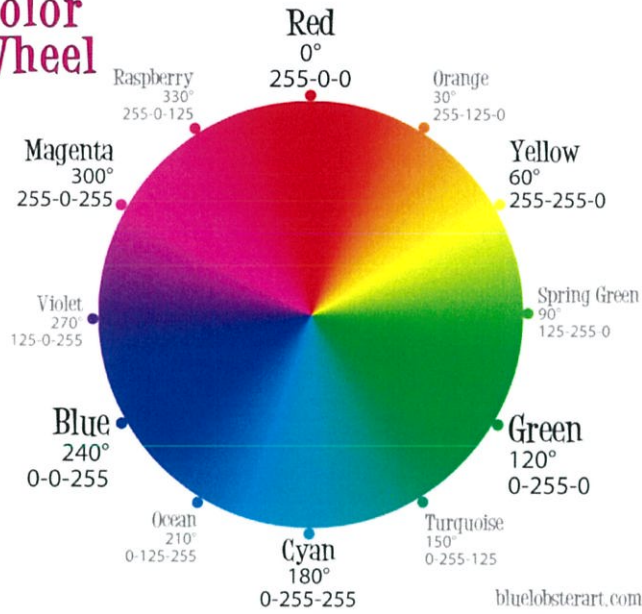
#### 2.2.4 แบบจำลองสี

มาตรฐานของสีที่ใช้อยู่ในปัจจุบันมีอยู่หลายระบบด้วยกันทั้งนี้จะขึ้นอยู่กับการนำไปใช้ แต่โดยทั่วไปแล้วทุกมาตรฐานจะมีแนวคิดเดียวกันคือการแทนจุดสีด้วยจุดที่อยู่ในสเปส 3 มิติ โดยจะมีแกนอ้างอิงสำหรับจุดสีนั้นในสเปสซึ่งแต่ละแกนจะมีความเป็นอิสระต่อกัน ตัวอย่างเช่นในระบบ RGB จะมีแกนสี คือแกนสีแดงเขียวและน้ำเงินในระบบ HLS จะมีแกนเป็นค่าสี (Hue) ความสว่าง (Lightness) และความบริสุทธิ์ของสี (Saturation) ตัวอย่างระบบสีที่นิยมใช้กันได้แก่ระบบ RGB HSV (Hue Saturation Value) และ HLS (Hue Lightness Saturation)

##### 2.2.4.1 แบบจำลองสีแบบ RGB

ระบบสี RGB เป็นระบบสีที่เกิดจากการรวมกันของแสงสีแดงเขียวและน้ำเงินโดยมีการรวมกันแบบ Additive ซึ่งโดยปกติจะนำไปใช้ในจอภาพแบบ CRT (Cathode ray tube) ในการใช้งานระบบสี RGB ยังมีการสร้างมาตรฐานที่แตกต่างกันออกไปที่นิยมใช้งานได้แก่ RGB-CIE และ RGB-NTSC ระบบสีแบบ RGB ของ CIE เป็นระบบสีที่พัฒนาขึ้นโดย CIE (Commission International l 'Eclairage) ซึ่งอ้างอิงสีด้วยสีแดงที่ 700 นาโนเมตร สีเขียวเท่ากับ 546.1 นาโนเมตร และสีน้ำเงิน 435.8 นาโนเมตร ระบบสีแบบ RGB ของ NTSC เป็นระบบที่พัฒนาโดย NTSC (National Television System Committee) เพื่อใช้สำหรับการแสดงภาพของจอภาพแบบ CRT เป็นมาตรฐานสำหรับผู้ผลิตแบบ CRT ให้มีลักษณะเดียวกัน

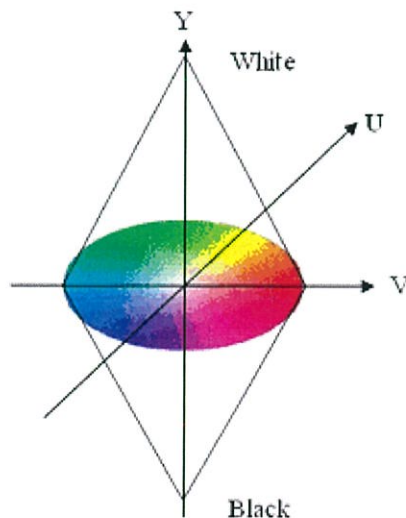
# RGB Color Wheel



รูปที่ 2.5 วงล้อของระบบสีแบบ RGB [18]

## 2.2.4.2 ระบบสีแบบ YUV

ระบบสีแบบ YUV ใช้สำหรับโทรทัศน์แบบ PAL และ SECAM ซึ่งยังมีใช้อยู่ในหลายๆประเทศโดย Y คือค่าความสว่างของภาพส่วนสัญญาณ U และ V เป็นสัญญาณที่เก็บค่าสีของภาพ ต่อมาได้มีระบบ YIQ มาใช้แทนเนื่องจากพบว่าสัญญาณ I และ Q สามารถลด Bandwidth ได้มากกว่าสัญญาณ U และ V ในขณะที่ได้ภาพที่มีคุณภาพเท่ากันโดยแบบจำลองนี้มีความสัมพันธ์กับแบบจำลองสี RGB โดยที่ค่าสีของ RGB ถูกกระทำการด้วยวิธีการนอร์มอลไลซ์ให้มีค่าอยู่ในช่วง  $[0, 1]$



รูปที่ 2.6 ระบบสีแบบ YUV [19]

โดยสมการการแปลงภาพจากภาพ RGB เป็น ภาพ YUV ได้แก่

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.144 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.515 & -0.1 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

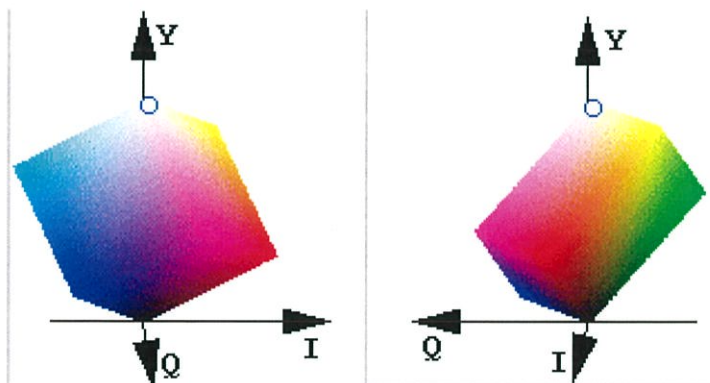
### 2.2.4.3 แบบจำลองสีแบบ YIQ

เป็นระบบที่ใช้ใน TV Broadcasting สำหรับ NTSC ประโยชน์หลักก็เพื่อให้ใช้งาน  
ได้กับโทรทัศน์แบบขาว-ดำโดยที่ Y คือความสว่างของภาพส่วน I และ Q จะเป็นสัญญาณที่เข้ารหัส  
สีของภาพไว้ดังนั้นสำหรับโทรทัศน์ขาว-ดำนั้นสามารถใช้ค่า Y ค่าเดียวก็สามารถได้ภาพที่สมบูรณ์

โดยแบบจำลองนี้มีความสัมพันธ์กับความสัมพันธ์กับแบบจำลองสี RGB ดังสมการ

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.144 \\ 0.596 & -0.273 & -0.321 \\ 0.212 & -0.523 & 0.311 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

และมีระบบสีในแนวแกนต่างๆดังรูปที่ 2.7



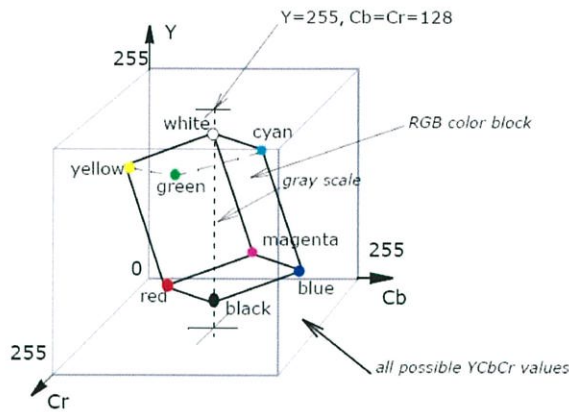
รูปที่ 2.7 ระบบสีแบบ YIQ [20]

### 2.2.4.4 แบบจำลองสีแบบ YCbCr

แบบจำลองสี YCbCr เป็นแบบจำลองสีที่มีความคล้ายคลึงกับแบบจำลองสี YUV  
มาก เพียงแต่มีการเปลี่ยนแปลงสัญญาณซึ่งถูกปรับเปลี่ยนมาจากแบบจำลองสี YUV และ  
แบบจำลองสี YCbCr นี้ยังมีการประยุกต์ใช้งานในกระบวนการบีบอัดข้อมูลของภาพแบบ JPEG  
และขนาดของข้อมูลวิดีโอแบบ MPEG แบบจำลองสี YCbCr นี้ ค่า Y จะใช้เก็บข้อมูลของความเข้ม  
แสง (Luminance) ส่วน Cb , Cr จะใช้เก็บข้อมูลของแสงสี ดังรูปที่ 2.8

โดยแบบจำลองสีนี้มีความสัมพันธ์กับแบบจำลองสี RGB ดังสมการ

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.144 \\ -0.168736 & -0.331264 & 0.5 \\ 0.5 & -0.418688 & -0.081312 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 0 \\ 0.5 \\ 0.5 \end{bmatrix}$$



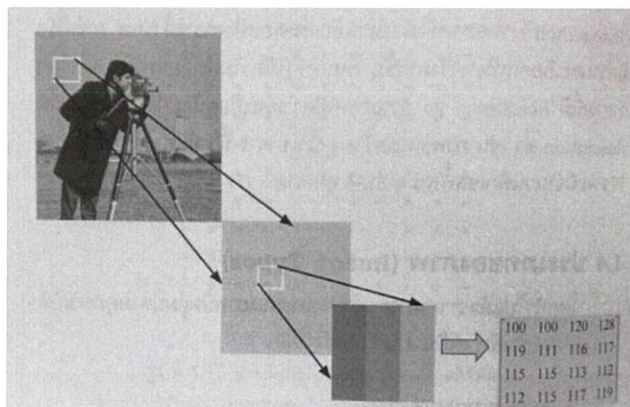
รูปที่ 2.8 ระบบสีแบบ YCbCr [21]

## 2.2.5 ประเภทของภาพ

โดยทั่วไปแล้ว สามารถแบ่งประเภทของภาพบิตแมปตามคุณสมบัติการแสดงผลของสีภาพ เป็น 4 ประเภท ดังนี้

### 2.2.5.1 ภาพระดับความเข้มเทา (Intensity Image or Gray Scale Image)

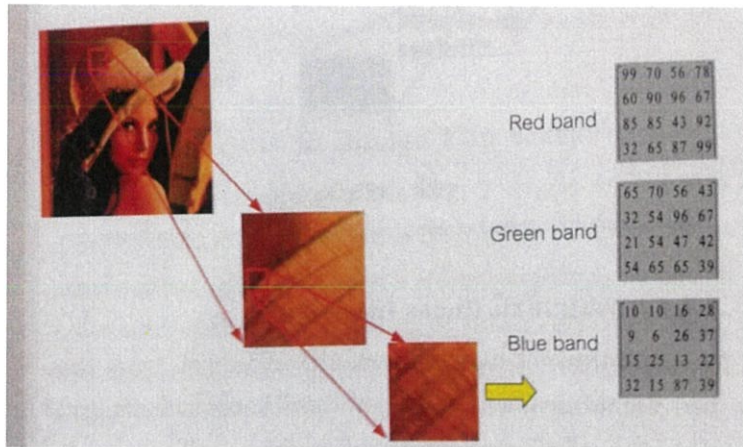
ลักษณะของภาพชนิดนี้ ในแต่ละพิกเซลจะมีค่าความเข้มของแสงในแต่ละระดับที่แตกต่างกันออกไป ตั้งแต่ระดับเทาดำไปยังสีขาว สามารถกำหนดระดับความเข้มของแสงนั้นได้โดยใช้ค่าระดับความเข้มเทา (Gray Scale หรือ Gray Level) โดยปกติทั่วไป ภาพแบบระดับสีเทาจะมีค่าระดับความเข้มเทาเท่ากับ 8 บิต ดังนั้นค่าความเข้มแสงจะถูกแบ่งออกเป็น 256 ระดับ เมื่อค่าระดับความเข้มเทามีค่าเป็น 0 จะหมายถึงจุดภาพนั้นมีค่าความเข้มของแสงต่ำ จะทำให้จุดภาพเป็นสีดำ ในทางกลับกัน หากค่าระดับความเข้มเทามีค่าเป็น 255 จะหมายถึงจุดภาพนั้นมีค่าความเข้มของแสงสูง จะทำให้จุดภาพเป็นสีขาว โดยภาพระดับความเข้มเทาจะแสดงดังรูปที่ 2.9



รูปที่ 2.9 ระดับความเข้มเทา [3]

### 2.2.5.2 ภาพสี (Color Image)

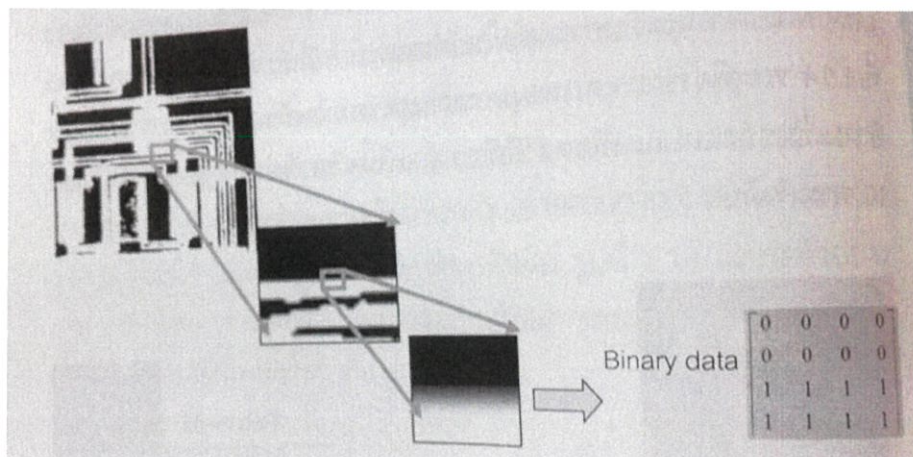
ภาพชนิดนี้ แต่ละจุดภาพหรือพิกเซลของภาพจะเก็บค่าระดับความเข้มเทาของแต่ละแถบแสงของแม่สีหลัก 3 สีที่ซ้อนกันอยู่คือ สีแดง สีเขียว และสีน้ำเงิน ซึ่งในแต่ละพิกเซลนั้นๆ ก็ จะแสดงผลของค่าสีของแต่ละพิกเซลตามระดับความเข้มในแต่ละแถบแสงสีนั้น แสดงดังรูปที่ 2.10



รูปที่ 2.10 ภาพสี [3]

### 2.2.5.3 ภาพไบนารี (Binary Image)

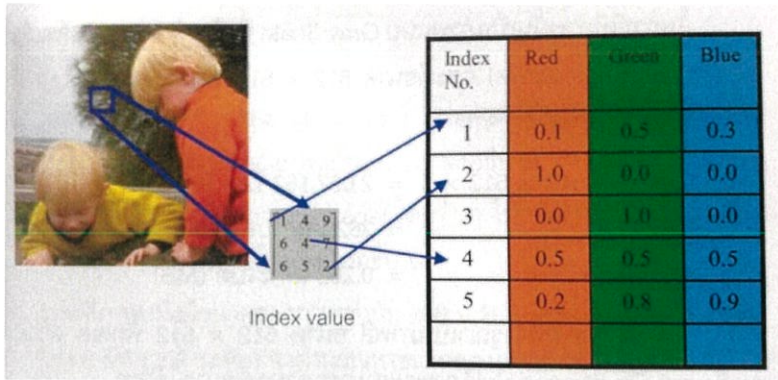
ภาพไบนารีจะแสดงลักษณะของข้อมูลภาพในรูปแบบขาวดำ กล่าวคือในแต่ละพิกเซลของภาพจะถูกแสดงด้วยค่าแบบไบนารี คือมี 1 บิต ประกอบไปด้วยค่า 1 คือจุดสีขาว และ 0 หมายถึงจุดสีดำ ภาพชนิดนี้เหมาะสำหรับภาพที่เกี่ยวกับตัวอักษร (Text) หรือลายนิ้วมือ (Finger Print) โดยภาพไบนารีนั้นจะแสดงดังรูปที่ 2.11



รูปที่ 2.11 ภาพไบนารี [3]

#### 2.2.5.4 ภาพแบบดัชนี (Index Image)

ภาพประเภทนี้ในแต่ละพิกเซลของภาพจะเก็บค่าดัชนี ซึ่งเป็นตัวเลขจำนวนเต็มซึ่งจะถูกนำค่าดัชนีดังกล่าวไปเทียบกับตารางสีซึ่งเป็นตารางสีแสดงค่าแสงสีแดง สีเขียวและสีน้ำเงิน ซึ่งค่าดัชนีนี้จะเป็นตัวบ่งชี้ให้เห็นว่าภาพในแต่ละตำแหน่งพิกเซลใด ๆ มีค่าอัตราส่วนของแม่แสง 3 สีในอัตราส่วนละเท่าไร โดยภาพแบบดัชนีจะแสดงดังรูปที่ 2.12



รูปที่ 2.12 ภาพดัชนี [3]

## 2.3 โลบารี่ที่ใช้ในการพัฒนาโปรแกรมการประมวลผลภาพ : OpenCV, EmguCV และ AForge.Net

### 2.3.1 OpenCV

OpenCV (Open Source Computer Vision) เป็น Library programming functions ที่ช่วยในการเขียนโปรแกรมเกี่ยวกับ Computer vision แบบเรียลไทม์ ซึ่งปัจจุบันสามารถใช้ได้กับระบบปฏิบัติการ Windows, Linux, Mac, iOS และ Android โดยมี Algorithm ที่ถูกแก้ไขวิเคราะห์มาแล้วมากกว่า 2,500 algorithm มียอดดาวน์โหลดมากกว่าสองล้านครั้งจากทั่วโลก และรองรับภาษา C, C++, Java, Python ในการพัฒนาโปรแกรมซึ่ง Open CV นั้นได้ถูกพัฒนาโดยบริษัทอินเทล (Intel) ด้วยความที่ Open CV เป็นโลบารี่ที่ไม่มีข้อจำกัดในเรื่องของฮาร์ดแวร์ (Hardware) เวอร์ชันปัจจุบันอยู่ที่ 2.4.3 ใน Open CV นั้นจะประกอบไปด้วยชุดคำสั่งเกี่ยวกับระบบการรู้จำใบหน้า การจดจำท่าทางการเคลื่อนไหว การจำแนกวัตถุ และการติดตามการเคลื่อนไหวของวัตถุ เป็นต้น

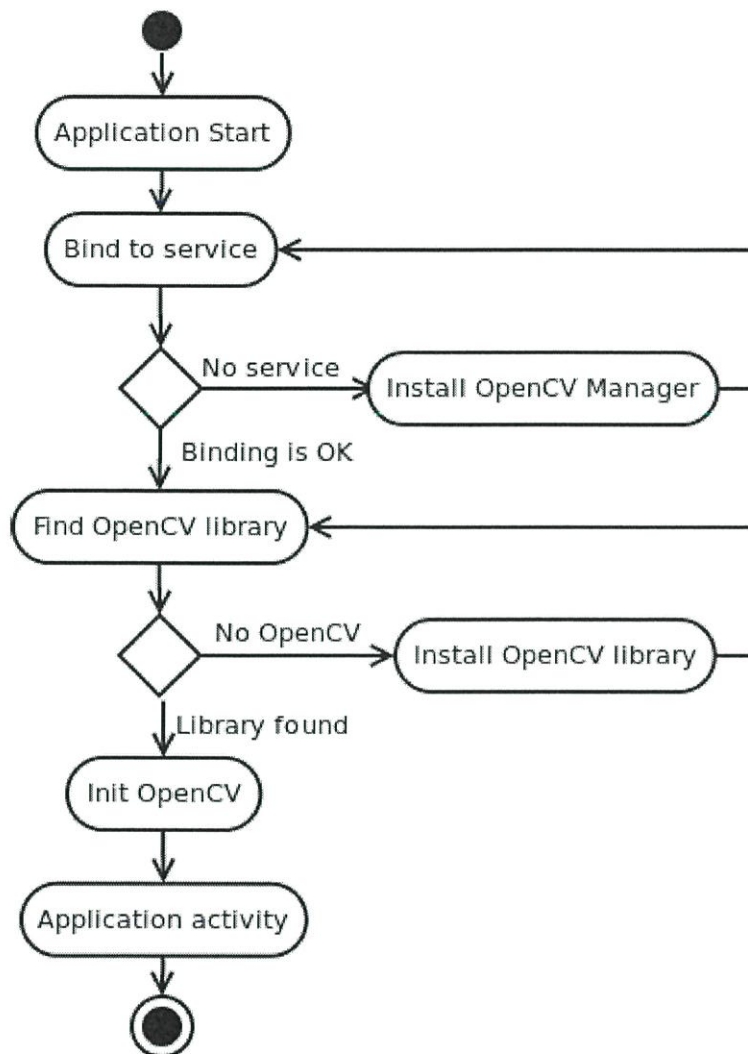


รูปที่ 2.13 โลโก้ของ OpenCV [4]

OpenCV เป็นโครงสร้างของโมดูล ซึ่งหมายความว่าแพ็คเกจต่างๆรวมไปถึงมีหลายโลบารี่ที่คงที่หรือใช้ร่วมกัน ซึ่งโมดูลต่างๆที่สามารถใช้งานได้มีดังนี้

- core เป็นโมดูลที่กำหนดโครงสร้างพื้นฐานของข้อมูล รวมทั้งอาร์เรย์หลายมิติ ฟังก์ชันทางคณิตศาสตร์ และฟังก์ชันพื้นฐานที่ใช้โดยโมดูลอื่นๆทั้งหมด
- imgproc เป็นโมดูลการประมวลผลภาพที่รวมไปถึงแบบทั้งเชิงเส้นและไม่เป็นเชิงเส้น รวมถึงการกรองสัญญาณภาพ ภาพเรขาคณิต การเปลี่ยนแปลงของภาพ เช่น การเปลี่ยนขนาด เปลี่ยนมุมของภาพ รวมไปถึงการแปลงพื้นที่สี ฮิสโทแกรมและอื่นๆ
- video เป็นโมดูลที่ใช้ในการวิเคราะห์วิดีโอ รวมถึงการประเมินการเคลื่อนไหวของวัตถุ อัลกอริทึมการติดตามวัตถุ เป็นต้น
- calib3d เป็นพื้นฐานของอัลกอริทึมทางเรขาคณิตหลายมุมมอง องค์ประกอบของภาพสามมิติ การประมาณค่าวัตถุ อัลกอริทึมการติดต่อกันของสเตอริโอ เป็นต้น
- objdetect โมดูลในการตรวจจับวัตถุ เช่น ใบหน้า ดวงตา แก้ว คน รถยนต์ และอื่นๆ
- highgui โมดูลที่ง่ายต่อการใช้อินเตอร์เฟสในการจับสัญญาณวิดีโอ และการเข้ารหัสวิดีโอ
- gpu มี GPU-accelerated algorithm จากโมดูล OpenCV ที่แตกต่างกันออกไป โมดูลอื่นๆอีกมากมายที่นำมาใช้ เช่น FLANN , Google test wrappers และอื่นๆ

ในการใช้งาน OpenCV จะมีขั้นตอนในการใช้งาน ดังโฟลว์ชาร์ตรูปที่ 2.14



รูปที่ 2.14 โฟลว์ชาร์ตการทำงานโดยใช้ไลบรารีของ OpenCV [4]

### 2.3.2 EmguCV

EmguCV เป็นไลบรารีที่ใช้ช่วยในการเขียนโปรแกรมเกี่ยวกับคอมพิวเตอร์วิทัศน์แบบเรียลไทม์เหมือน OpenCV แต่ต่างกันตรงที่ภาษาที่ใช้ในการพัฒนาโปรแกรม เพราะ Emgu CV จะใช้ภาษาพวก .NET ในการพัฒนา เช่น C# , VB, VC++, IronPython etc. เวอร์ชันปัจจุบันอยู่ที่ 2.4.2 ซึ่ง Emgu CV มีลักษณะเป็น Cross platform คือสามารถใช้ได้บนระบบปฏิบัติการทั้ง Windows , Linux และ MacOSX



รูปที่ 2.15 โลโก้ของ EmguCV [5]

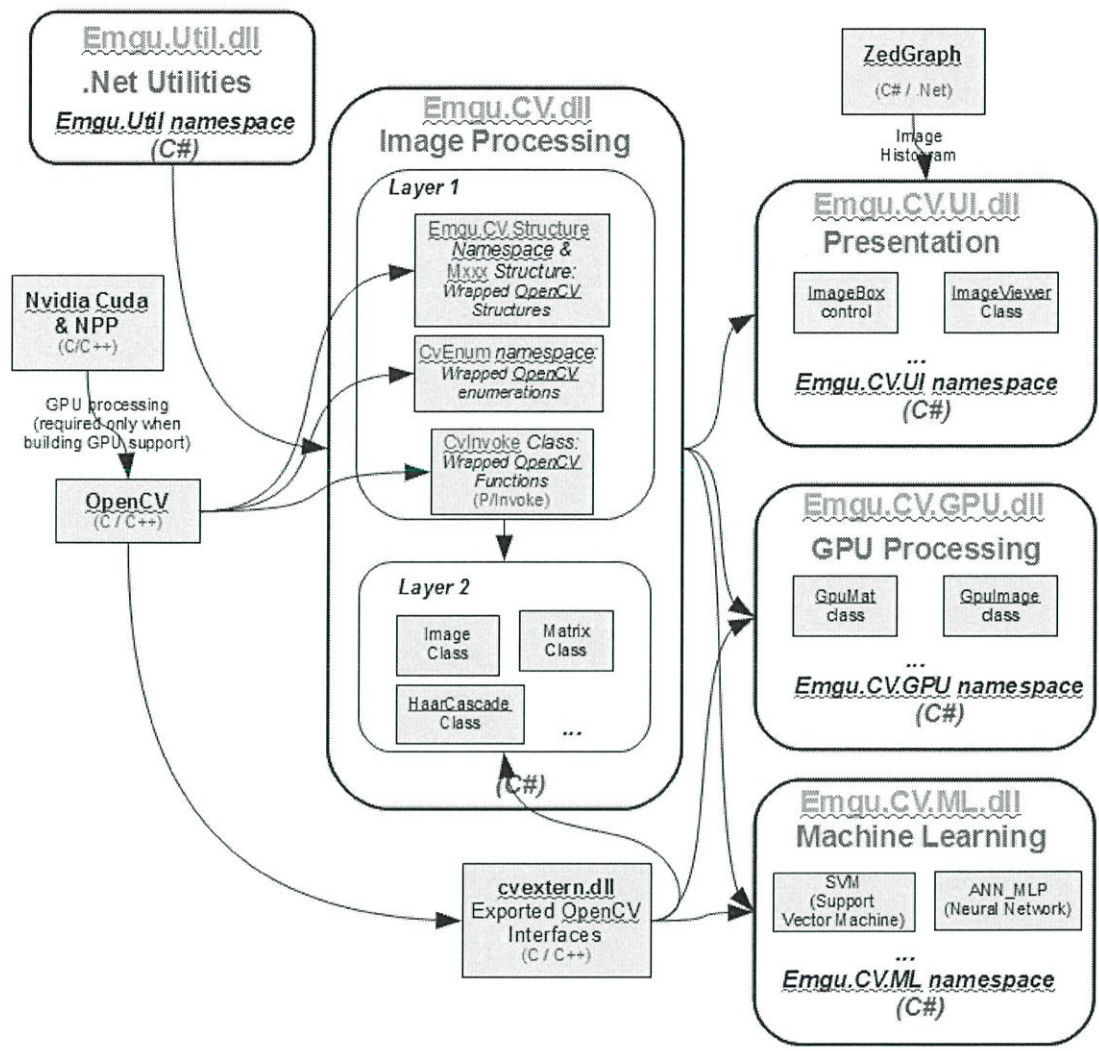
โดย EmguCV แต่ละเวอร์ชัน ก็จะมีความสามารถในการทำงานต่างกันไปดังตารางที่ 1 ดังนี้

ตารางที่ 1 เปรียบเทียบความสามารถในการทำงานของEmguCVแต่ละเวอร์ชัน

Name	Emgu CV (Open Source)	Emgu CV (Commercial Optimized)	Windows Store App *	Emgu CV for iOS (Commercial)	Emgu CV for Android (Commercial)
Platform	Windows, Linux, Mac OSX	Windows	Windows 8 App Store	iOS (iPhone, iPad, iPod Touch)	Android
Supported CPU Architecture	i386, x64	i386, x64	i386, x64	armeabi, armeabi-v7, i386 (Simulator)	armeabi, armeabi-v7a, x86
CUDA GPU Processing	✓	✓	X	X	X
Tesseract OCR †	✓	✓	X	✓	✓
Intel TBB ‡ (multi-thread)	X	✓	X	X	X
Intel IPP ‡ (high performance)	X	✓	X	X	X
Intel C++ Compiler ‡ (fast code)	X	✓	X	X	X
Exception Handling	✓	✓	✓	✓	✓
Debugger Visualizer	✓	✓	X	X	X
Emgu.CV.UI	✓	✓	X	X	X
License	GPL	Commercial License	Commercial License	Commercial License	Commercial License

#### ข้อดีของ EmguCV

- เป็น Cross Platform ซึ่งสามารถทำงานได้ทุกระบบปฏิบัติการไม่ว่าจะเป็น Windows , Linux , MacOSX , iOS หรือ Android
- เป็น Cross Language คือสามารถใช้งานได้จากหลากหลายภาษาไม่ว่าจะเป็น C# , VB.NET , C++ , Python และจะมีซอร์สโค้ดตัวอย่างมาให้ทุกครั้ง
- คลาสของรูปภาพมีทั้งแบบสีทั่วไปและแบบมีความลึก
- มีการจัดการกับโค้ดหรือไฟล์ที่เป็นเหมือนขยะให้อัตโนมัติ
- มีการดำเนินการในระดับพิกเซล
- เป็นทางเลือกในการใช้คลาสของภาพนอกเหนือจากใช้โดยตรงจาก OpenCV



รูปที่ 2.16 สถาปัตยกรรมโดยรวมของ EmguCV [5]

### 2.3.3 AForge.NET

AForge.NET เป็นเฟรมเวิร์คที่ถูกพัฒนาสำหรับภาษา C# สำหรับใช้ในการวิจัยและพัฒนาในด้านคอมพิวเตอร์วิทัศน์และระบบปัญญาประดิษฐ์ (Artificial Intelligence) เช่นการประมวลผลภาพ (Image Processing) โครงข่ายประสาท (Neural Network) การประยุกต์ใช้ในพันธุกรรมคอมพิวเตอร์ (Genetic Algorithms) การเรียนรู้ของเครื่อง (Machine Learning) และวิทยาการหุ่นยนต์ (Robotics) เป็นต้น ปัจจุบันอยู่ที่เวอร์ชัน 2.2.5

# AForge.NET

## F R A M E W O R K

รูปที่ 2.17 สัญลักษณ์ของ AForge.NET [6]

ตัวอย่างของชุดคำสั่งไลบรารีของ AForge.NET ประกอบไปด้วย

- AForge.Imaging เป็นชุดคำสั่งที่ใหญ่ที่สุดในขอบเขตงานประกอบด้วยชุดคำสั่งการประมวลผลภาพต่างๆซึ่งช่วยในการปรับแต่งภาพและการประมวลผลภาพต่างๆ
- AForge.Vision เป็นชุดคำสั่งที่ประกอบด้วยเครื่องมือตรวจจับการเคลื่อนไหวหรือท่าทางต่างๆและชุดคำสั่งการประมวลผลผลการเคลื่อนไหว
- AForge.Video เป็นชุดคำสั่งที่ประกอบด้วยคลาสต่างๆที่ใช้ในการเข้าถึงข้อมูลวิดีโอ
- AForge.Robotics เป็นชุดคำสั่งที่ใช้ควบคุมอุปกรณ์เกี่ยวกับหุ่นยนต์
- AForge.Neuro เป็นชุดคำสั่งที่ประกอบด้วยวิธีการทางโครงสร้างเครือข่ายประสาทเทียมและขั้นตอนวิธีการเรียนรู้ต่างๆ
- AForge.Genetic เป็นชุดคำสั่งที่ประกอบด้วยคลาสที่ใช้แก้ปัญหาทางต่างๆจาก Genetic Algorithms, Genetic Programming และ Gene Expression Programming
- AForge.MachineLearning เป็นชุดคำสั่งที่ประกอบด้วยคลาสที่เกี่ยวกับการเรียนรู้ของเครื่องและมีไลบรารีอื่นๆที่ใช้งานอีกมากมาย

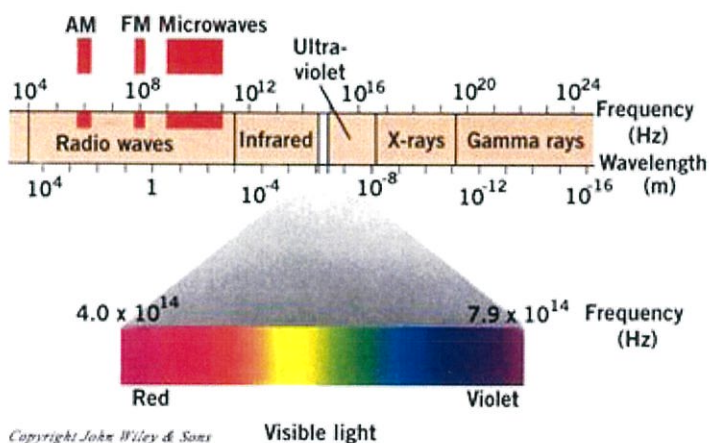
## 2.4 แสงและแหล่งกำเนิดแสง

### 2.4.1 ธรรมชาติของแสง

แสงคือรังสีแม่เหล็กไฟฟ้าที่อยู่ในช่วงสเปกตรัมของคลื่นแม่เหล็กไฟฟ้าที่สามารถมองเห็นได้ คือ อยู่ในย่านความถี่ 380 THz ( $3.8 \times 10^{14}$  เฮิรตซ์) ถึง 750 THz ( $7.5 \times 10^{14}$  เฮิรตซ์) จากความสัมพันธ์ระหว่าง ความเร็ว ( $v$ ) ความถี่ ( $f$  หรือ  $\nu$ ) และ ความยาวคลื่น ( $\lambda$ ) ของแสง :

$$v = f \lambda$$

และ ความเร็วของแสงในสุญญากาศมีค่าคงที่ ดังนั้นจึงสามารถแยกแยะแสงโดยใช้ตามความยาวคลื่นได้ โดยแสงที่มองเห็นได้ข้างต้นนั้นจะมีความยาวคลื่นอยู่ในช่วง 400 นาโนเมตร (ย่อ 'nm') และ 800 นาโนเมตร (ในสุญญากาศ) การมองเห็นของมนุษย์นั้นเป็นผลมาจากภาวะอนุภาคของแสง โดยเฉพาะ เกิดจากการที่ก้อนพลังงาน (อนุภาคโฟตอน) แสง ไปกระตุ้น เซลล์รูปแท่งในจอตา (Rod cell) และ เซลล์รูปกรวยในจอตา (Cone cell) ที่จอตา (Retina) ให้ทำการสร้างสัญญาณไฟฟ้าบนเส้นประสาท และส่งผ่านเส้นประสาทตาไปยังสมอง ทำให้เกิดการรับรู้มองเห็น



รูปที่ 2.18 ค่าย่านความถี่ของคลื่นแม่เหล็กไฟฟ้าวัดรวมไปถึงแสง [12]

### 2.4.2 หน่วยที่ใช้ในการวัดแสง

- ความจ้า (Brightness) หรือ อุณหภูมิ (Temperature)
- ความสว่าง (Illuminance หรือ Illumination) (หน่วย SI: ลักซ์ (lux))
- ฟลักซ์ส่องสว่าง (Luminous flux) (หน่วย SI: ลูเมน (lumen))
- ความเข้มของการส่องสว่าง (Luminous intensity) (หน่วย SI: แคนเดลา (candela))
- ความสุกใสของแสง (Brilliance) หรือ แอมพลิจูด (Amplitude)
- สี (Color) หรือ ความถี่ (Frequency)
- โพลาริเซชัน (Polarization) หรือ มุมการแกว่งของคลื่น (Angle of vibration)

### 2.4.3 ความสว่าง

ความส่องสว่าง (ลูเมนแนช) หมายถึงปริมาณแสงที่กระทบลงบนวัตถุต่อพื้นที่มีหน่วยเป็นลูเมนต่อตารางเมตรหรือลักซ์ (ถ้าหน่วยเป็นลูเมนต่อตารางฟุตความส่องสว่างก็เป็นฟุตแคนเดิล)

$$\text{ลูเมนแนช} = \text{ปริมาณแสง (ลูเมน)} / \text{พื้นที่ (m}^2\text{)}$$

ความสว่าง (ลูเมนแนช) หมายถึงปริมาณแสงที่สะท้อนออกมาจากวัตถุต่อพื้นที่มีหน่วยเป็นแคนเดลาต่อตารางเมตร ปริมาณแสงที่เท่ากันเมื่อตกกระทบลงมาบนวัตถุที่มีสีต่างกันจะมีปริมาณแสงสะท้อนกลับต่างกันนั่นคือลูเมนแนชต่างกัน สาเหตุที่ต่างกันก็เนื่องมาจากสัมประสิทธิ์การสะท้อนแสงของวัสดุต่างกัน

การบอกสีทางการส่องสว่างมักด้วยอุณหภูมิสี ซึ่งหมายถึงสีที่เกิดจากการเผาไหม้วัสดุสีดำซึ่งมีการดูดซับความร้อนได้สมบูรณ์ด้วยอุณหภูมิที่กำหนด เช่นหลอดฟลูออเรสเซนต์คูลไวท์มีอุณหภูมิสี 6500 องศาเคลวินหมายถึงเมื่อเผาวัสดุสีดำให้ร้อนถึงอุณหภูมิ 6500 เคลวินวัตถุนั้นจะเปล่งแสงออกมาเป็นสีคูลไวท์หรือขาวปนน้ำเงินเป็นต้นตัวอย่างอุณหภูมิสีของหลอดต่างๆเป็นดังนี้

- เทียนไข 1900 เคลวิน
- หลอดอินแคนเดสเซนต์ 2800 เคลวิน
- หลอดฟลูออเรสเซนต์ : เดย์ไลท์ (Daylight 6500 เคลวิน คูลไวท์ (Cool White) 4500 เคลวิน วอร์มไวท์ (Warm White) 3500 เคลวิน

แสงสว่างพื้นฐานที่ต้องใช้เพื่อการใช้งานแยกออกได้เป็นระบบต่างๆดังนี้

1. แสงสว่างทั่วไป (General Lighting) คือการให้แสงกระจายทั่วไปทั้งบริเวณพื้นที่ใช้งานซึ่งใช้กับความส่องสว่างที่ไม่มากจนเกินไป
2. แสงสว่างเฉพาะที่ (Locallised Lighting) คือการให้แสงสว่างเป็นบางบริเวณที่ต้องการใช้ไฟแสงสว่างมากเพื่อการประหยัดพลังงาน
3. แสงสว่างเฉพาะที่และแสงสว่างทั่วไป (General and Locallised Lighting) คือการให้แสงสว่างทั้งแบบทั่วไปทั้งบริเวณและเฉพาะที่ที่ทำงานซึ่งมักใช้กับงานที่ต้องการความส่องสว่างสูงซึ่งไม่สามารถให้แสงแบบแสงสว่างทั่วไปได้เพราะเปลืองค่าไฟฟ้ามากแต่ก็ไม่สามารถให้แสงแบบแสงสว่างเฉพาะที่ได้ เพราะเมื่อเงยหน้าจากการทำงานก็จะพบบริเวณข้างเคียงมืดเกินไปทำให้สายตาเสียได้

## 2.5 หลอดไฟและแหล่งกำเนิดแสง

### 2.5.1 หลอดไส้ร้อนแบบธรรมดา

หลอดไส้ร้อนแบบธรรมดา หรือ หลอดความร้อน หรือ หลอดไส้ (Incandescent light bulb, Incandescent lamp หรือ Incandescent light globe) ให้แสงสว่างโดยการให้ความร้อนแก่ไส้หลอดที่เป็นลวดโลหะกระแทงมีอุณหภูมิสูงและเปล่งแสง ไส้ที่ร้อนนั้นถูกป้องกันไม่ให้สัมผัสอากาศโดยหลอดแก้วที่เติมแก๊สเฉื่อยหรือที่เป็นสุญญากาศ ในหลอดฮาโลเจน กระบวนการทางเคมีคืนให้โลหะเป็นไส้หลอด ซึ่งขยายอายุการใช้งาน หลอดไฟฟ้านี้ได้รับกระแสไฟฟ้าโดยเทอร์มินอลต่อสายไฟ (Feed-through terminal) หรือลวดที่ฝังในแก้ว หลอดไฟฟ้าส่วนใหญ่ใช้ในเต้ารับซึ่งสนับสนุนหลอดไฟฟ้าทางกลไกและเชื่อมกระแสไฟฟ้าเข้ากับเทอร์มินัลไฟฟ้าของหลอด

หลอดไส้ร้อนแบบธรรมดาผลิตออกมาในหลายขนาด กำลังส่องสว่าง และอัตราความต่างศักย์ จาก 1.5 โวลต์ไปจนถึงราว 300 โวลต์ หลอดประเภทนี้ไม่ต้องอาศัยอุปกรณ์ควบคุมภายนอก มีค่าบำรุงรักษาต่ำ และทำงานได้ดีเท่ากันทั้งไฟฟ้ากระแสสลับหรือกระแสตรง ด้วยเหตุนี้ หลอดไส้ร้อนแบบธรรมดาจึงใช้กันอย่างกว้างขวางในครัวเรือนและไฟฟ้าใช้ในเชิงพาณิชย์ ตลอดจนไฟฟ้าแบบพกพา อย่างเช่น ไฟตั้งโต๊ะ ไฟนาร์ถยนต์ และไฟฉาย และไฟฟ้าสำหรับตกแต่งและโฆษณาแต่มีอายุการใช้งานสั้น เปลืองไฟมาก การนำไปใช้ประโยชน์บางอย่างของหลอดไส้ร้อนแบบธรรมดาใช้ความร้อนที่เกิดขึ้นจากไส้หลอด อาทิ เครื่องฟักไข่ กล่องฟักไข่สำหรับสัตว์ปีก ไฟความร้อนสำหรับสวนจำลองสภาพแวดล้อม (Vivarium) ของสัตว์เลื้อยคลาน การให้ความร้อนอินฟราเรดในกระบวนการให้ความร้อนและอบแห้งในอุตสาหกรรม ความร้อนส่วนเกินนี้เพิ่มพลังงานที่ต้องใช้ในระบบปรับอากาศของอาคาร



รูปที่ 2.19 หลอดไส้ร้อนแบบธรรมดา [13]

### 2.5.2 หลอดฮาโลเจน

หลอดไฟชนิดนี้ประกอบด้วยไส้ทั้งสแตนเมื่อเทียบกับหลอดทั่วไปจะมีอายุการใช้งานที่นานกว่าเนื่องจากทนความร้อนได้ดีกว่าหลอดไส้ทั่วไปเฉลี่ยที่ 1,500 – 3,000 ชั่วโมง นิยมใช้ในพื้นที่โซลาร์สินค้าเพราะจะสว่างกว่าแต่ก็นำมาใช้กับที่พักอาศัยได้เช่นกันในมุมที่ต้องการแสง เช่น บริเวณมุมอับหรือห้องทำงาน

หลอด HID หรือชื่อเต็มว่า High Intensity Discharge เป็นหลอดที่ให้ความสว่างสูงกว่าฮาโลเจนถึง 4 เท่าแถมมีอายุการใช้งานนานกว่าหลอดชนิดอื่น ๆ ซึ่งถ้าเทียบกับหลอดแบบกลมจะมีอายุการใช้งานนานกว่า 20 เท่า นิยมใช้งานในพื้นที่ต้องการความสว่างเป็นพิเศษ เช่น ลานจอดรถ

หลอดไฮบริดฮาโลเจน หลอดที่พัฒนามาจาก หลอดกลม หลอดตะเกียบและหลอดฮาโลเจนรวมกัน เพื่อให้ได้ระดับความสว่างและอายุการใช้งานที่นานขึ้น เหมาะใช้งานในบ้านเรือนและที่พักอาศัยทั่วไป

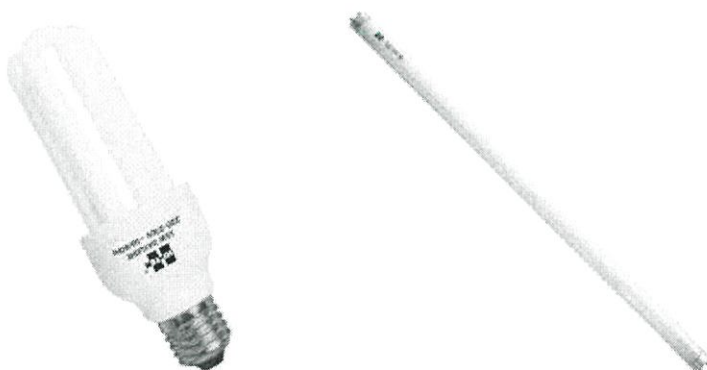


รูปที่ 2.20 หลอดไฟแบบฮาโลเจน [14]

### 2.5.3 หลอดฟลูออเรสเซนต์

**หลอดฟลูออเรสเซนต์** หลอดชนิดนี้นิยมใช้ตามที่พักอาศัย การเกิดแสงจะใช้ประจุจากอิเล็กตรอนผ่านไปยังสารเรืองแสงที่เคลือบหลอดไฟ เป็นกระบวนการที่ใช้พลังงานค่อนข้างมากเช่นกัน ส่วนใหญ่เป็นหลอดประเภท T12 หรือหลอดอ้วน แต่หากเป็นหลอดประเภท T8 จะเรียกหลอดผอมซึ่งมีขนาดเล็กกว่าให้ความสว่างเท่ากันแต่ประหยัดไปกว่าหลอดประเภทแรก

หลอดคอมแพคฟลูออเรสเซนต์ คนไทยส่วนใหญ่รู้จักกันในนาม “หลอดตะเกียบ” ถูกพัฒนามาเพื่อทดแทนไส้หลอดแบบเก่าด้วยคุณสมบัติที่เล็กลงแต่ให้ความสว่างเพิ่มขึ้น อายุการใช้งานเฉลี่ย 8 ปี หรือ 8,000 ชั่วโมง อีกทั้งยังประหยัดได้ถึง 4 เท่าของหลอดแบบเดิม ขนาดของหลอดตะเกียบแบ่งออกเป็น 5 ชนิด คือ U2, U3 เหมาะใช้กับที่พักอาศัย ส่วน U4, U5, U6 เหมาะใช้กับโรงงานและอาคารพาณิชย์

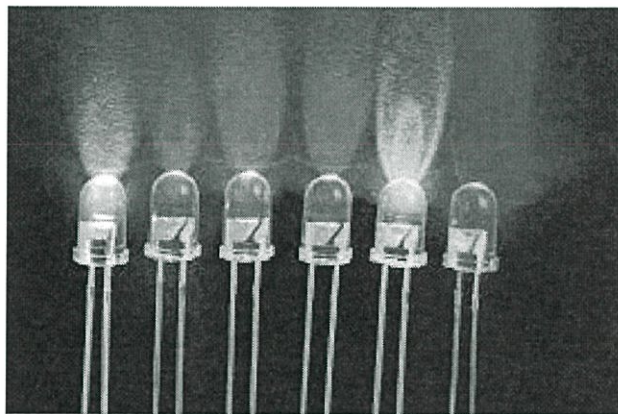


รูปที่ 2.21 หลอดแบบฟลูออเรสเซนต์ [16]

#### 2.5.4 หลอดแอลอีดี (Light Emitting Diode, LED)

เป็นชิ้นส่วนอิเล็กทรอนิกส์ชนิดหนึ่ง สามารถเปล่งแสงสว่างเมื่อให้กระแสไฟผ่าน โดยปกติหลอดชนิดนี้สามารถเปล่งแสงได้เมื่อจ่ายแรงดันไฟตรงเพียงเล็กน้อยเท่านั้น และประสิทธิภาพในการให้แสงก็ยิ่งดีกว่าหลอดไฟขนาดเล็ก แสงที่เกิดขึ้นเป็นการแลกเปลี่ยนอิเล็กตรอนของสารกึ่งตัวนำภายในแอลอีดี เรียกว่าปรากฏการณ์นั้นว่า Electroluminescence ซึ่งแตกต่างจากหลอดทั่วไปที่ใช้กระแสไฟในการจุดไส้หลอดเพื่อให้เกิดแสงสว่าง ผลก็คือตัวหลอดเกิดความร้อนเมื่อใช้งาน แอลอีดีจึงใช้กระแสไฟฟ้าต่ำกว่าในการให้กำเนิดแสงสว่างและความร้อนที่เกิดขึ้นก็ต่ำด้วยเช่นกัน แอลอีดียังสามารถเปล่งแสงได้หลากสี ขึ้นอยู่กับอัตราส่วนของสารกึ่งตัวนำเมื่อทำการผลิตแอลอีดี ปัจจุบันผลิตได้ทุกสีและยังสามารถผลิตแสงชนิดพิเศษอินฟราเรด ที่ตาคนมองไม่เห็นได้อีกด้วย

หลอดแอลอีดีโดยทั่วไปมี 2 ชนิดใหญ่ๆคือหลอดแอลอีดีชนิดที่ตาคนเห็นได้กับชนิดที่ตาคนมองไม่เห็นต้องใช้ทรานซิสเตอร์มาเป็นตัวรับแสงแทนตาคน



รูปที่ 2.22 หลอดแอลอีดี (LED) [15]

ปัจจุบันจากความก้าวหน้าอย่างรวดเร็วของเทคโนโลยีเซมิคอนดักเตอร์ทำให้เทคโนโลยีของ LED ก้าวหน้าอย่างรวดเร็วตามไปด้วย LED ได้ถูกพัฒนาขึ้นเรื่อยๆทั้งในด้านสีของแสงที่เปล่งออกมา ไม่ว่าจะเป็นสีแดง สีเขียว สีส้ม หรือที่ผลิตได้ท้ายสุดและทำให้วงการแอลอีดีพัฒนาขึ้นอย่างรวดเร็ว คือสีน้ำเงินซึ่งการเกิดขึ้นของแอลอีดีสีน้ำเงินนี้ทำให้ครบแม่สี 3 สีคือสีแดง สีเขียว และสีน้ำเงินและเกิดเป็นจุดเริ่มต้นของจอแอลอีดี และแอลอีดีในงานไฟประดับต่างๆ ทั้งยังใช้ประโยชน์แพร่หลายมากขึ้นเรื่อยๆ เช่นในเครื่องคิดเลขสัญญาณจราจรไฟท้ายรถยนต์ป้ายสัญญาณต่างๆไฟฉายไฟให้สัญญาณของประกาศกรจอภาพยนตร์ขนาดใหญ่ ยิ่งไปกว่านั้นหน้าจอ LCD ของโทรศัพท์มือถือที่ใช้กันทั่วไปเกือบทั้งหมดจะให้แสงสว่างด้วย LED

การนำแอลอีดีไปใช้งานในรูปแบบต่าง ๆ สามารถแบ่งการใช้งานได้ 2 รูปแบบใหญ่ๆคือ

1.การใช้งานในด้านการตกแต่ง การใช้งานแอลอีดี การใช้แอลอีดีจะเป็นลักษณะตกแต่ง เปลี่ยนบรรยากาศ ความบันเทิงต่างๆ

2.การใช้งานแสงสว่างทั่วไป ส่วนใหญ่จะเน้นในแนวทางที่ใช้แทนหลอดไฟแบบเดิมๆส่วนใหญ่มักจะใช้เป็นแอลอีดีสีขาว การใช้แอลอีดีในรูปแบบนี้ส่วนใหญ่จะใช้ในจุดที่ติดตั้งยากหรือต้องการความประหยัด

## ข้อดีของแอลอีดี

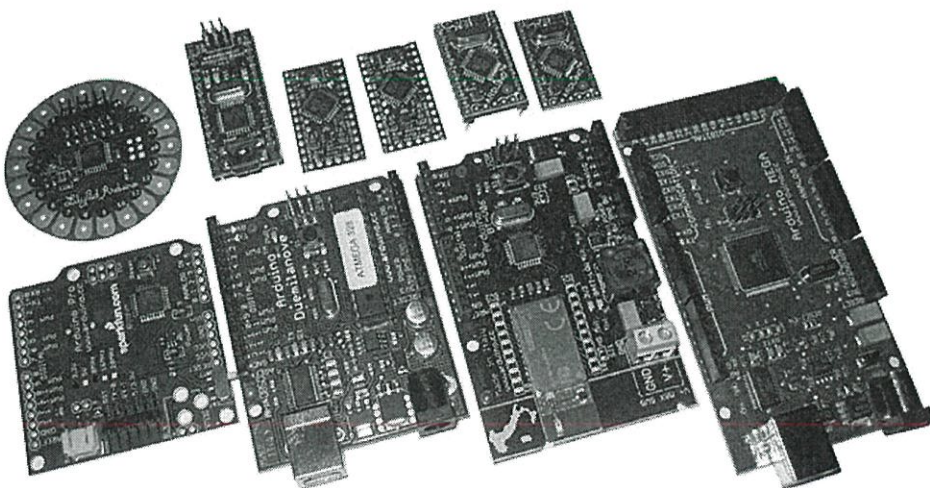
- ประสิทธิภาพในการให้แสงสว่างดีกว่าหลอดไฟธรรมดาทั่วไป.
- ตัวหลอด LED เอง เมื่อทำให้เกิดแสงขึ้นจะกินกระแสไฟน้อยมากประมาณ 1-20 mA
- มีอายุการใช้งานที่ยาวนานประมาณ 50,000 – 100,000 ชั่วโมง ขึ้นอยู่กับคุณภาพของแอลอีดี วงจรขับเคลื่อน สภาพภูมิอากาศ ความชื้น และอุณหภูมิซึ่งก็มีอายุการใช้งานที่ยาวนานกว่าหลอดที่ให้แสงสว่างชนิดอื่นๆมาก
- ทนทานต่อสภาวะอากาศ ทนทานต่อการสั่นสะเทือน และมีหลากหลายสีให้เลือกใช้

## 2.6 ไมโครคอนโทรลเลอร์

โครงการนี้ได้ใช้ไมโครคอนโทรลเลอร์ในการควบคุมการทำงานของวงจรในส่วนของวงจรที่ใช้ไฟกระพริบสร้างรูปแบบการกระพริบของหลอดไฟด้วยไมโครคอนโทรลเลอร์ โดยใช้ Arduino Uno R3

### 2.6.1 Arduino

Arduino คือไมโครคอนโทรลเลอร์ที่สามารถใช้เป็นเครื่องมือในการควบคุมอุปกรณ์อิเล็กทรอนิกส์เพื่อใช้งานตามที่ต้องการ สามารถเชื่อมต่อคอมพิวเตอร์ด้วยการเสียบสาย USB เชื่อมต่อเข้ากับบอร์ดไมโครคอนโทรลเลอร์ Arduino โดยใช้ไฟเลี้ยงจากสาย USB (+5V) ทั้งนี้ Arduino ยังจัดได้ว่าเป็นรูปแบบการพัฒนาประเภทโอเพ่นซอร์ส สามารถเรียกใช้หรือเพิ่มไลบรารีต่างๆเพื่อสะดวกใช้งานตามจุดประสงค์ที่ต้องการ จึงทำให้ Arduino เป็นที่นิยมใช้งานมากในปัจจุบัน คุณสมบัติ Arduino สามารถต่อกับอุปกรณ์อิเล็กทรอนิกส์ เช่น ตัววัดเซ็นเซอร์ ตัววัดอุณหภูมิ มอเตอร์ รีเลย์ แอลอีดี และอื่นๆอีกมากมาย



รูปที่ 2.23 บอร์ด Arduino แบบต่าง ๆ ที่นิยมใช้ [7]

## 2.6.2 ข้อดีของบอร์ด Arduino

- ราคาไม่แพงเมื่อเทียบกับราคาไมโครคอนโทรลเลอร์ชนิดอื่นๆ
- โปรแกรมการใช้งานสามารถใช้งานได้หลายแพลตฟอร์ม ทั้ง Windows , Linux และ MacOSX
- เหมาะสำหรับผู้ใช้งานเริ่มต้นเนื่องจากยืดหยุ่นในการใช้งานยังเพิ่มไลบรารีต่างๆเพื่อสะดวกในการเรียกใช้งานฟังก์ชันต่างๆ
- เปิดเผยแพร่ซอร์สโค้ด และ นำไปพัฒนาต่อยอดได้ โปรแกรม Arduino ดีพิมพ์แบบเปิดเผยซอร์สโค้ด และสามารถเพิ่มเติมความสามารถผ่าน C++ library สามารถข้ามไปเล่น AVR C ซึ่งเป็นต้นแบบของ Arduino และสามารถเพิ่มเติม AVR - C โค้ดได้โดยตรงถ้าต้องการ
- เปิดเผยแพร่วงจร และ นำไปพัฒนาขยาย Hardware ได้
- Arduino ใช้ไมโครคอนโทรลเลอร์ของ Atmel วงจรของบอร์ดดีพิมพ์แบบเปิดเผยวงจรภายใต้ Creative Commons License สามารถนำไปดัดแปลงต่อขยายและเพิ่มประสิทธิภาพ เพื่อศึกษาการทำงานได้ฟรี

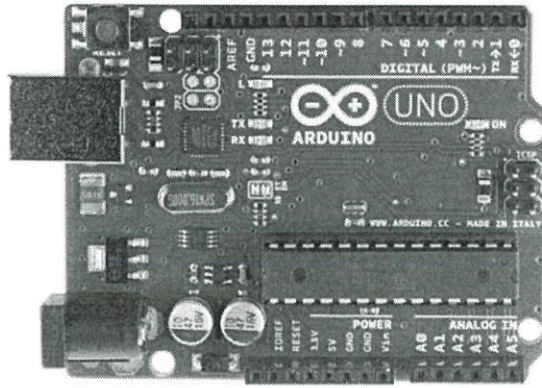
## 2.6.3 Arduino UNO

Arduino UNO เป็นไมโครคอนโทรลเลอร์ที่ใช้ไมโครคอนโทรลเลอร์ ATmega328 ซึ่งมีดิจิตอลอินพุต/เอาต์พุต 14 พอร์ต (มี 6 พอร์ตสามารถใช้เป็น PWM เอาต์พุต), 6 พอร์ตอนาล็อกอินพุต, 16 Mhz ceramic resonator, USB Connection, Power jack, ICSP Header และ Reset Button และต่างจากบอร์ดอื่นๆเพราะเป็นบอร์ดที่ไม่มี Usb-to-serial

สำหรับบอร์ด Arduino UNO R3 มีรายละเอียดของตัวบอร์ดดังนี้

### ตารางที่ 2 แสดงรายละเอียด Arduino UNO

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz



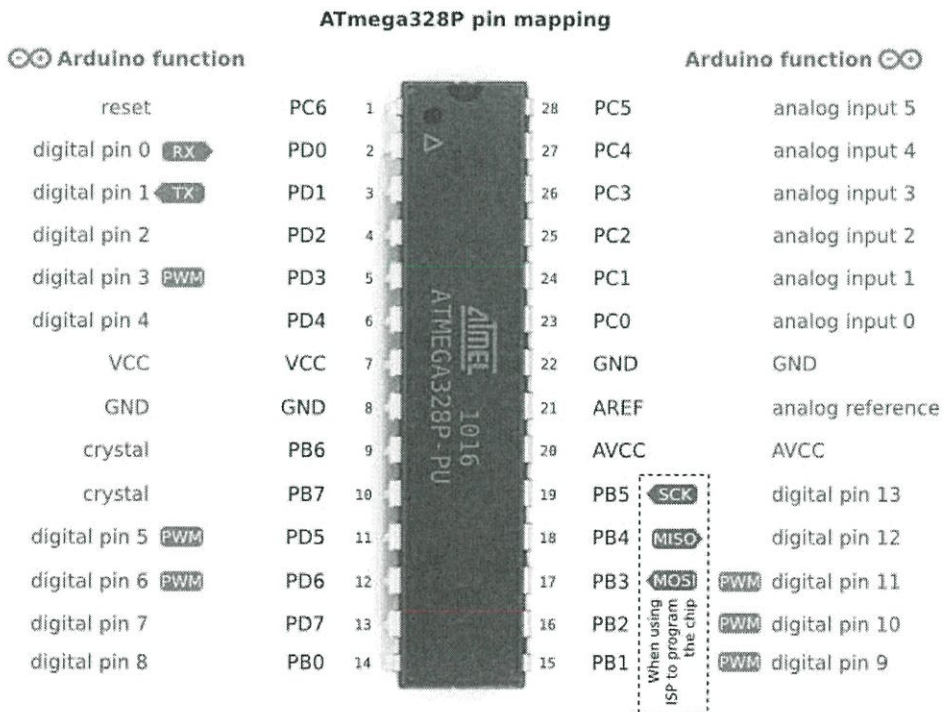
รูปที่ 2.24 บอร์ด Arduino Uno R3 [7]

### 2.6.3.1 หน่วยความจำ

ATmega328 มีหน่วยความจำ 32 KB (โดย 0.5 KB ใช้สำหรับ Bootloader) ซึ่งมี 2KB สำหรับ SRAM และ 1 KB สำหรับ EEPROM

### 2.6.3.2 พอร์ตอินพุตและเอาต์พุต

พอร์ตอินพุตและเอาต์พุตของ Arduino UNO มีทั้งหมด 14 ขา สามารถใช้เป็นที่อินพุตและเอาต์พุตได้ โดยใช้ฟังก์ชัน pinMode(), digitalWrite(), digitalRead() โดยใช้แรงดันไฟ 5V แต่ละขาสามารถรับกระแสได้มากที่สุด 40 mA



รูปที่ 2.25 ขาพอร์ตของ ATmega328 [7]

และมีรายละเอียดในแต่ละขาพอร์ตดังนี้

- Serial : 0 (RX) และ 1 (TX) ใช้สำหรับรับ (RX) และส่ง (TX) TTL Serial data แต่ละขาจะต่อตรงกันกับขาของ ATmega8U2 USB-to-TTL Serial ship
- External Interrupt 2 และ 3 แต่ละขาสามารถปรับเปลี่ยนค่าของอินเตอร์รัพท์ให้เป็น Low value, Rising หรือ Falling edge หรือเปลี่ยนค่าโดยใช้ฟังก์ชัน attachInterrupt()
- PWM 3, 5, 9, 10 and 11 ให้เอาต์พุต 8 บิต PWM โดยใช้ฟังก์ชัน analogWrite()
- SPI 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK) ขาเหล่านี้จะสนับสนุนการสื่อสารโดยใช้ SPI
- LED 13 เป็นไดโอดเปล่งแสงที่ฝังตัวอยู่ในบอร์ดที่พอร์ตที่ 13 เมื่อเป็นลอจิก High จะสว่างและถ้าเป็น Low จะดับ
- TWI A4 หรือ SDA และ A5 หรือ SCL สนับสนุนการสื่อสารแบบ TWI โดยใช้ไลบรารี Wire
- AREF. เป็น Reference Voltage สำหรับ analog input ใช้ฟังก์ชัน analogReference()
- Reset ถ้าเป็นลอจิก LOW จะเป็นการรีเซ็ตไมโครคอนโทรลเลอร์

## 2.7 Channel Coding ( การเข้ารหัสช่องสัญญาณ )

เป็นกระบวนการที่ภาคส่งทำการเพิ่มบิตพิเศษรวมเข้าไปกับบิตข้อมูล เพื่อช่วยให้ภาครับสามารถตรวจจับหรือแก้ไขบิตบางชนิดที่ผิดพลาดในระหว่างการส่งผ่าน การส่งผ่านสัญญาณดิจิทัลในระบบสื่อสารโดยทั่วไป มักจะเกิดปัญหาการผิดพลาดของรูปสัญญาณเนื่องจากคุณสมบัติที่ไม่เป็นอุดมคติของช่องสัญญาณเองหรือเนื่องจากผลกระทบของสัญญาณรบกวนภายนอกในรูปแบบต่างๆ ปัญหาเหล่านี้อาจส่งผลให้ข้อมูลดิจิทัลที่รับได้ที่ภาครับมีความผิดพลาดเกิดขึ้น ด้วยเหตุนี้ระบบสื่อสารปัจจุบันซึ่งต้องการความถูกต้องและความแน่นอนในการส่งข้อมูลสูง มักจะมีการนำข้อมูลดิจิทัลไปผ่านกระบวนการเข้ารหัสช่องสัญญาณก่อนที่จะส่งออก เพื่อให้การรับส่งมีความผิดพลาดน้อยลงและอยู่ในระดับที่ยอมรับได้ในการเข้ารหัสช่อง

สัญญาณนั้นจำเป็นที่จะต้องเพิ่มจำนวนของบิตที่จะส่งออก โดยบิตพิเศษที่เพิ่มเข้ามาจะช่วยให้ภาครับสามารถที่จะตรวจจับความผิดพลาดได้ (Error Detection) หรือหากมีการเพิ่มจำนวนบิตเข้าไปเป็นจำนวนที่มากพอภาครับก็อาจจะสามารถแก้ไขความผิดพลาด (Error Correction) ของข้อมูลได้ด้วย สังเกตว่าการเข้ารหัสช่องสัญญาณมีผลทำให้อัตราบิตข้อมูลที่ต้องส่งจริงมีขนาดสูงซึ่งหมายถึงว่าช่องสัญญาณที่ใช้ส่งจะต้องมีแบนด์วิดท์ที่ใหญ่ขึ้นด้วยหรือถ้าพิจารณา ในทางกลับกันหากช่องสัญญาณมีแบนด์วิดท์ที่จำกัดและต้องการให้การรับส่งของข้อมูลมีความถูกต้องมากขึ้น ก็ต้องลดอัตราการส่งบิตข้อมูลของผู้ใช้ลง ในการที่จะทำให้การรับส่งข้อมูลมีความถูกต้องโดยอาศัยวิธีการเข้ารหัสช่องสัญญาณ

การเข้ารหัสช่องสัญญาณสามารถแบ่งอย่างคร่าวๆ ออกได้เป็น 2 ประเภทใหญ่ๆ คือ การเข้ารหัสแบบบล็อก (Block Codes) และการเข้ารหัสแบบคอนโวลูชัน (Convolutional Codes) การเข้ารหัสแบบบล็อกจะแบ่งบิตข้อมูลที่จะทำการเข้ารหัสออกเป็นกลุ่มหรือที่เรียกว่าบล็อกขนาด  $k$  บิต จากนั้นบิตข้อมูลแต่ละบล็อกก็จะถูกแปลงให้กลายเป็นคำรหัส (Codeword) ที่มีความยาวเท่ากับ  $n$  โดยที่  $n > k$  ดังนั้นจึงมักเรียกการเข้ารหัสนี้ว่า  $(n, k)$  โดยปกติแล้วชุดรหัสที่ได้จากการเข้ารหัสนั้นจะยังคงประกอบด้วยส่วนของบิตข้อมูลเดิม  $k$  บิต และส่วนของบิตพิเศษที่เพิ่มเข้าไปอีก  $n - k$  บิตหรือที่เรียกว่าเช็คบิต (Check bit) เพื่อใช้สำหรับตรวจสอบว่ามีความผิดพลาดในบิตข้อมูลในระหว่างที่ส่งผ่านช่องสัญญาณหรือไม่ ณ ที่ภาครับก็จะมีการทำหน้าที่ถอดรหัสเพื่อดึงบิตข้อมูลเดิมออกมา พร้อมกันนั้นก็ให้ค่าที่เรียกว่า ซินโดรม (Syndrome) ออกมาด้วย โดยค่าซินโดรมนั้นมีไว้สำหรับบ่งบอกว่ามีความผิดพลาดเกิดขึ้นในข้อมูลหรือไม่ หรืออาจใช้ในการบ่งบอกถึงตำแหน่งของบิตที่ผิดด้วย

### 2.7.1 รหัสแบบบล็อก ( Block Code )

รหัสแบบบล็อก (Block Code) เป็นหนึ่งในรหัสที่ใช้ป้องกันความผิดพลาดล่วงหน้า (FEC) ซึ่งในการเข้ารหัสแบบบล็อกนั้น ข้อมูลข่าวสารจะถูกแบ่งออกเป็นชุด หรือบล็อกย่อยๆ ก่อนการเข้ารหัส ซึ่งรหัสแบบบล็อกสามารถตรวจสอบและบอกตำแหน่งข้อมูลที่ผิดพลาดได้ ซึ่งขึ้นอยู่กับเงื่อนไขของการเข้ารหัส รหัสแบบบล็อกนี้มีหลายรูปแบบ เช่น รหัสแบบลิเนียร์บล็อก (Linear Block Code ) รหัสแบบแฮมมิง รหัสแบบ Cyclic และรหัสแบบ BCH เป็นต้น

#### 2.7.1.1 การเข้ารหัสแบบลิเนียร์บล็อก (Linear Block Code)

การเข้ารหัสแบบลิเนียร์บล็อกนั้นเป็นพื้นฐานที่สำคัญของรหัสแบบบล็อก รหัสแบบลิเนียร์บล็อกเป็นการตรวจเช็คพาริตี (Parity) ชนิดหนึ่ง โดยการเข้ารหัสแบบลิเนียร์บล็อกนั้นเป็นการนำข้อมูลข่าวสารที่ต้องการเข้ารหัสมาแบ่งเป็นชุดๆ หรือบล็อกย่อยๆ ซึ่งแต่ละชุดนั้นมีขนาดเท่ากับ  $k$  บิต แล้วนำมาหาค่าคำรหัสหรือ Codeword ขนาด  $n$  บิต โดยทั่วไปแทนด้วยสัญลักษณ์  $(n, k)$  การเข้ารหัสแบบลิเนียร์บล็อกนั้นการคำนวณหาค่าคำรหัสเป็นแบบเชิงเส้น (Linear) ซึ่งการหาค่าคำรหัสจะใช้ Generator matrix ในการคำนวณ ดังนี้

$$U = mG$$

โดย  $U$  คือ ข้อมูลที่ได้หลังจากการเข้ารหัส  $m$  คือ ชุดของข้อมูลจำนวน  $k$  บิต ซึ่งนำมาเข้ารหัส และ  $G$  คือ Generator matrix ของการเข้ารหัส

#### 2.7.1.2 การถอดรหัสแบบลิเนียร์บล็อก

การตรวจจับความผิดพลาดของข้อมูลที่ถูกส่งไปของรหัสแบบบล็อกนั้น วิธีการคือการนำข้อมูลที่รับได้มาทำการคูณกับเมทริกซ์ที่ใช้สำหรับตรวจจับความผิดพลาดของข้อมูล ( $H$ ) ซึ่งผลลัพธ์ที่ได้จากการคูณนั้นเรียกว่าค่า Syndrome ( $S$ ) ซึ่งเป็นค่าแสดงถึงลักษณะความผิดพลาดของข้อมูลที่เกิดขึ้นกับข้อมูลชุดนั้นๆ ค่า Syndrome ทำให้สามารถนำไปใช้แก้ไขความผิดพลาดของข้อมูลให้ถูกต้องได้ โดยลักษณะการถอดรหัสแบบลิเนียร์บล็อกมีการทำงานดังสมการ

$$S = (U + e)H^T$$

โดย  $S$  คือค่าความผิดพลาดที่เกิดขึ้น และ  $H$  จะมีความสัมพันธ์กับ  $G$  ซึ่งทำให้  $U$  มีค่าเป็นศูนย์ ดังนั้นค่า Syndrome ที่ได้นั้นมีค่าเท่ากับ

$$S = eH^T$$

ค่า Syndrome ที่ได้จากการคำนวณนั้นใช้ในการตรวจสอบว่ามีการผิดพลาดเกิดขึ้นที่บิตใด ทำให้สามารถทำการแก้ไขข้อมูลให้เกิดความผิดพลาดให้ถูกต้องได้ แต่รหัสแบบลิเนียร์บล็อกนั้นมีข้อจำกัดในการตรวจสอบหาตำแหน่งผิดพลาดถ้าความผิดพลาดเกิดขึ้นนั้นเกินขอบเขตของรหัส

#### 2.7.1.3 ความสามารถในการตรวจจับและแก้ไขความผิดพลาด

ความสามารถในการตรวจจับและแก้ไขความผิดพลาด เป็นตัวบ่งชี้ประสิทธิภาพของรหัส การวิเคราะห์คุณสมบัติต่างๆ ของการเข้ารหัสต้องมีการพิจารณาถึงองค์ประกอบหลายส่วน ได้แก่ น้ำหนักแฮมมิง (Hamming weight) และระยะห่างแฮมมิง (Hamming distance) สำหรับค่าน้ำหนักแฮมมิงหรือ  $w(U)$  นั้นหมายถึงจำนวนบิตของข้อมูลที่ไม่เท่ากับ "0" ซึ่งในกรณีของข้อมูลไบนารีหมายถึงจำนวนข้อมูลที่เป็น "1" สำหรับค่าระยะห่างแฮมมิงระหว่างข้อมูล  $U$  และ  $V$  หรือ  $d(U,V)$  นั้นหมายถึงจำนวนบิตของข้อมูลทั้งสองที่มีค่าแตกต่างกัน ซึ่งทั้งสองค่านี้มีความสัมพันธ์กันดังสมการ

$$d(U,V) = \sum_{i=1}^n U_i + V_i$$

ข้อมูลที่ผ่านมาการเข้ารหัสนั้นค่าระยะห่างแฮมมิงที่อาจเท่ากันหรือแตกต่างกันก็ได้ แต่ในการวิเคราะห์การทำงานจะพิจารณาเพียงข้อมูลที่มีค่าระยะห่างแฮมมิงที่มีค่าน้อยที่สุด ซึ่งเรียกว่าค่าระยะห่างแฮมมิงต่ำสุด (Minimum hamming distance) หรือถูกนำไปใช้ในการคำนวณความสามารถในการแก้ไขความผิดพลาดของข้อมูล สำหรับการเข้ารหัสแบบบล็อกนั้น ความสัมพันธ์ระหว่างค่า และจำนวนบิตที่มากที่สุดของข้อมูลที่เกิดความผิดพลาดที่ยังสามารถแก้ไขให้ถูกต้องได้ แสดงดังสมการ

$$t = \frac{d_{\min} - 1}{2}$$

ส่วนค่าของจำนวนบิตที่มากที่สุดของข้อมูลที่ยังสามารถตรวจจับได้ว่าเกิดความผิดพลาดขึ้น (Error detecting capability) นั้นแสดงดังสมการ

$$e = d_{\min} - 1$$

## 2.7.2 รหัสแบบคอนโวลูชัน (Convolutional Code)

Convolution Code หรือบางคนอาจเรียกว่า รหัสแบบขด จัดอยู่ในกลุ่มของ การเข้ารหัสผ่านช่องสัญญาณ (Channel Coding) โดยมีจุดประสงค์ เพื่อให้ผู้รับ รับสัญญาณได้ครบถ้วน หรือหากเกิดมีสัญญาณรบกวนขึ้น ผู้รับสามารถรับรู้ได้ถึงความผิดพลาดของข้อมูลที่ได้รับ อีกทั้งยังสามารถแก้ไขข้อผิดพลาดนั้นๆได้ โดยวิธีนี้เรียกว่า Forward Error Correction (FEC)

### 2.7.2.1 การเข้ารหัสแบบ Convolutional Code

การเข้ารหัส Convolution Code นั้น นำไปใช้อย่างแพร่หลาย เช่นใช้ในการส่งสัญญาณวิทยุระบบดิจิทัล โทรศัพท์มือถือ รวมถึงในสัญญาณดาวเทียม และใน BlueTooth อาจเนื่องจากหากเกิดข้อผิดพลาด หรือมี Noise เข้ามากรวน ฝั่งผู้รับสามารถตรวจสอบ และแก้ไขให้ข้อมูลที่ได้รับ กลับมาถูกต้อง แต่ถ้าเกิด Noise แบบแถบ (Burst noise) ละก็ อาจช่วยได้ไม่ทั้งหมด อาจต้องใช้เทคนิค Bit Interleave เข้ามาช่วย

การเข้ารหัส Convolution Code นั้น เริ่มต้นด้วยการสร้าง Generator เพื่อทำการแปลงข้อมูลจากบิตข้อมูล เป็นบิตรหัส Convolution Code โดยที่เริ่มต้นจากกระแสข้อมูลที่ต้องการเข้ารหัสวิ่งผ่าน ตัวสร้างรหัส จะนำแต่ละบิตข้อมูลมาเพิ่มรหัส

การเข้ารหัสแบบคอนโวลูชัน (Convolutional code) มีค่าพารามิเตอร์ทั้งหมด 3 ค่าคือ  $n$ ,  $k$ , และ  $K$  โดย

$K$  คือ Constraint length จำนวนชุดของสัญลักษณ์ข้อมูลที่ใช้ในการเข้ารหัสแต่ละรอบ

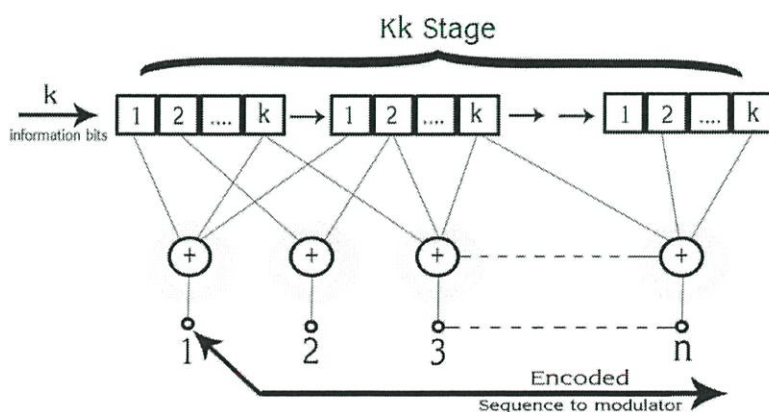
$k$  คือ จำนวนของบิตต่อหนึ่งสัญลักษณ์ เช่น ถ้าข้อมูลเข้าเป็นไบนารี ค่า  $k=1$  และถ้าหากข้อมูลเข้าเป็น 4-ary บิต ค่า  $k=2$

$kK$  คือ จำนวนชิพตรีจิสเตอร์ที่ต้องใช้

$n$  คือ จำนวนข้อมูลบิตที่ผ่านการเข้ารหัสแล้วและออกมาที่ขาออกของวงจรเข้ารหัส

$k/n$  คือ อัตราการเข้ารหัส

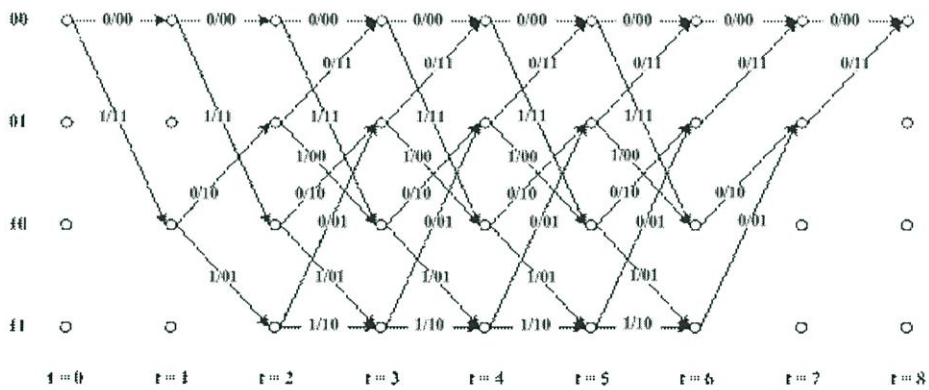
ดังนั้นในกรณีที่ใช้  $k=1$  จะต้องใช้จำนวนชิพตรีจิสเตอร์ทั้งหมด  $K$  ตัว และมีอัตราการเข้ารหัสเท่ากับ  $1/n$



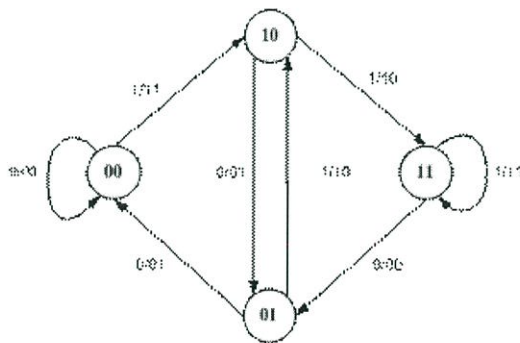
รูปที่ 2.26 แผนภาพการเข้ารหัสแบบ Convolutional Code

### 2.7.2.2 State Diagram และ Trellis Diagram

State Diagram คือ ไดอะแกรมแสดงค่าของข้อมูลใน Shift Register และเอาต์พุตของตัวเข้ารหัส ซึ่ง State Diagram ของวงจรเข้ารหัสในรูปที่ 2 แสดงดังรูปที่ 3 ตัวเลขที่อยู่ในวงกลมแต่ละวงนั้นหมายถึงสภาวะต่างๆ ของข้อมูลที่ถูกเก็บไว้ใน Shift Register ซึ่งในกรณีของวงจรเข้ารหัสที่ K เท่ากับ 3 จะมี State เท่ากับ  $2(K-1)$  หรือ 4 สถานะ (State) และสำหรับลูกศรที่ถูกแสดงไว้ในรูปนั้นจะแสดงถึงลักษณะของการเปลี่ยนแปลงการทำงานจากสถานะหนึ่งไปเป็นอีกสถานะหนึ่ง ซึ่งขึ้นอยู่กับข้อมูลที่ป้อนเข้ามา ณ เวลานั้นๆ นอกจากการแสดงการทำงานของวงจรเข้ารหัสโดยใช้ State Diagram ในการอธิบายการทำงานแล้วยังมีการแสดงการทำงานของวงจรเข้ารหัสในอีกรูปแบบหนึ่ง โดยการนำข้อมูลที่แสดงไว้ใน State Diagram มาแสดง ณ เวลาต่างๆ ซึ่งเรียกว่า Trellis Diagram ซึ่งตัวอย่างของ Trellis Diagram ของวงจรเข้ารหัสในรูปที่ 2 มีลักษณะดังรูปที่ 4 โดยค่า State อยู่ในแนวตั้งส่วนค่าเวลาอยู่ทางแนวนอน ส่วนเส้นทางต่างๆ ที่อยู่ใน Trellis Diagram นั้นแสดงถึงลักษณะการเปลี่ยนแปลงสภาวะของวงจร โดยตัวเลขที่อยู่เหนือหรือทางเดินในแต่ละเส้นทางนั้นแสดงถึงข้อมูลที่ป้อนเข้ามา (k) และข้อมูลที่ถูกส่งออกไป (n) ณ เวลานั้นๆ



รูปที่ 2.26 Trellis Diagram ที่ K=3 [17]



รูปที่ 2.27 State Diagram ที่ K=3 [17]

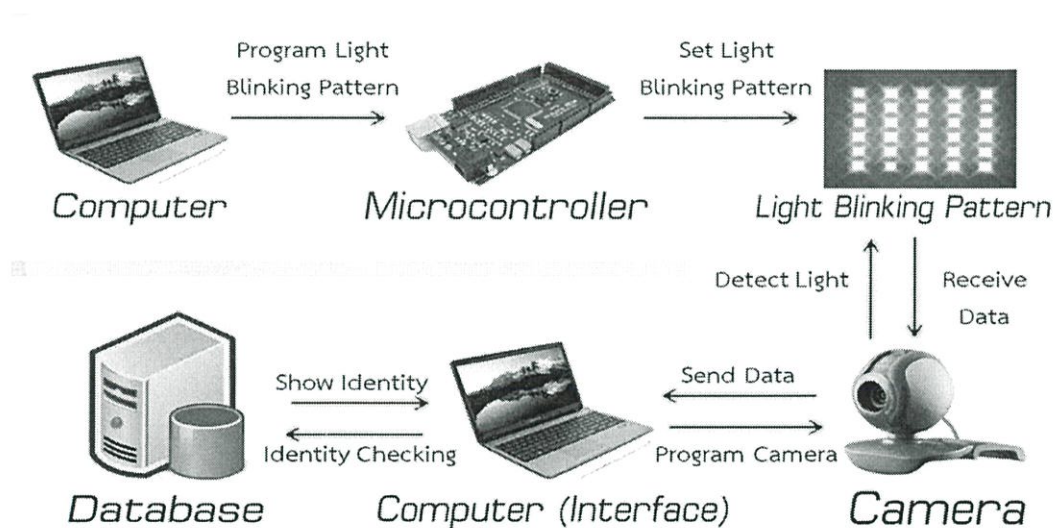
### บทที่ 3

## โครงสร้างของระบบและการออกแบบ

### 3.1 บทนำ

ในหัวข้อนี้จะกล่าวถึงภาพรวมการทำงานของระบบ และรายละเอียดแต่ละส่วนของโครงงาน รวมถึงขั้นตอนการทำงาน โครงสร้างของระบบซึ่งประกอบด้วยในส่วนภาคส่งและภาครับ โดยในภาคส่งจะประกอบไปด้วยไมโครคอนโทรลเลอร์ที่ถูกโปรแกรมการกะพริบของไฟมาต่อกับวงจรของไฟ และในภาครับคือกล้องที่ตรวจจับการกะพริบของแสง และคอมพิวเตอร์ที่มีโปรแกรมคอยแสดงผล และตรวจสอบอัตลักษณ์กับฐานข้อมูล

### 3.2 โครงสร้างโดยภาพรวมของระบบ

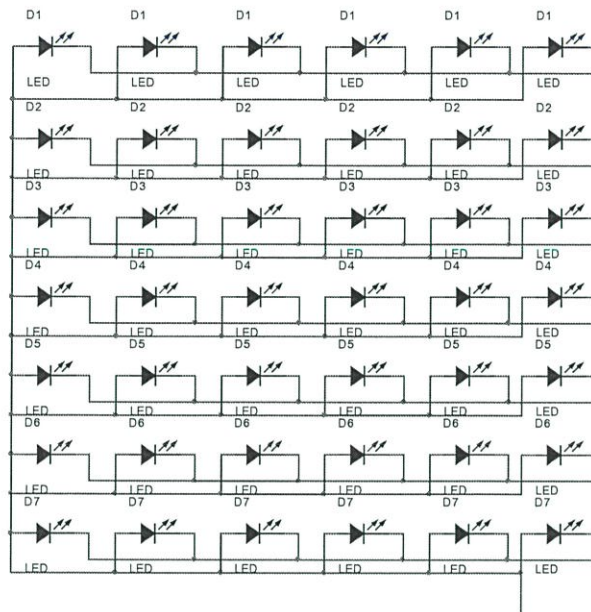


รูปที่ 3.1 โครงสร้างโดยภาพรวมของระบบ

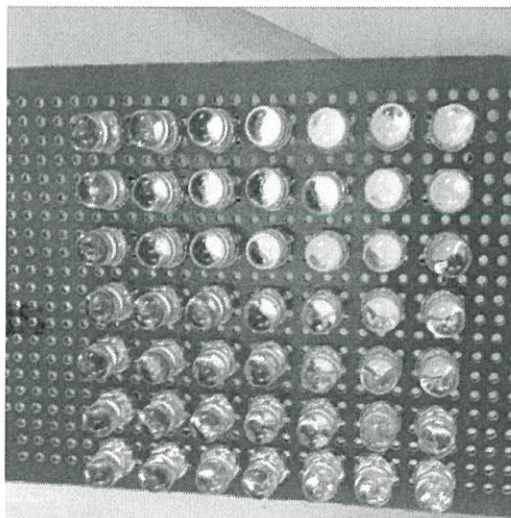
ระบบตรวจสอบอัตลักษณ์โดยใช้แสงนี้ จะประกอบไปด้วยหลักๆสองส่วนคือในส่วนของภาคส่งซึ่งจะมีคอมพิวเตอร์ที่โปรแกรมไมโครคอนโทรลเลอร์ให้หลอดไฟกะพริบเป็นแพทเทิร์นจังหวะติดดับที่ต้องการ และในส่วนของภาครับจะประกอบไปด้วยกล้องที่ทำหน้าที่ตรวจสอบรูปแบบการกะพริบของไฟโดยการเขียนโปรแกรมเชิงประมวลผลภาพโดยใช้โปรแกรม Visual C# ทำงานร่วมกับ OpenCV, EmguCV และ Aforge.NET และเมื่อรับข้อมูลจากการกะพริบของแสงมาได้ก็จะนำข้อมูลไปตรวจสอบอัตลักษณ์กับฐานข้อมูลว่าข้อมูลนี้มีอยู่หรือถูกต้องหรือไม่

### 3.3 การออกแบบในส่วนของภาคส่ง

ในภาคส่งจะประกอบไปด้วยไมโครคอนโทรลเลอร์ซึ่งเลือกใช้บอร์ด Arduino UNO ซึ่งใช้ไมโครคอนโทรลเลอร์ ATmega328 อยู่ และวงจรที่เชื่อมต่อจากบอร์ด Arduino ผ่านวงจรสวิตช์ไปยังหลอดไฟซึ่งเลือกใช้เป็นหลอดไฟ LED จำนวน 49 หลอด

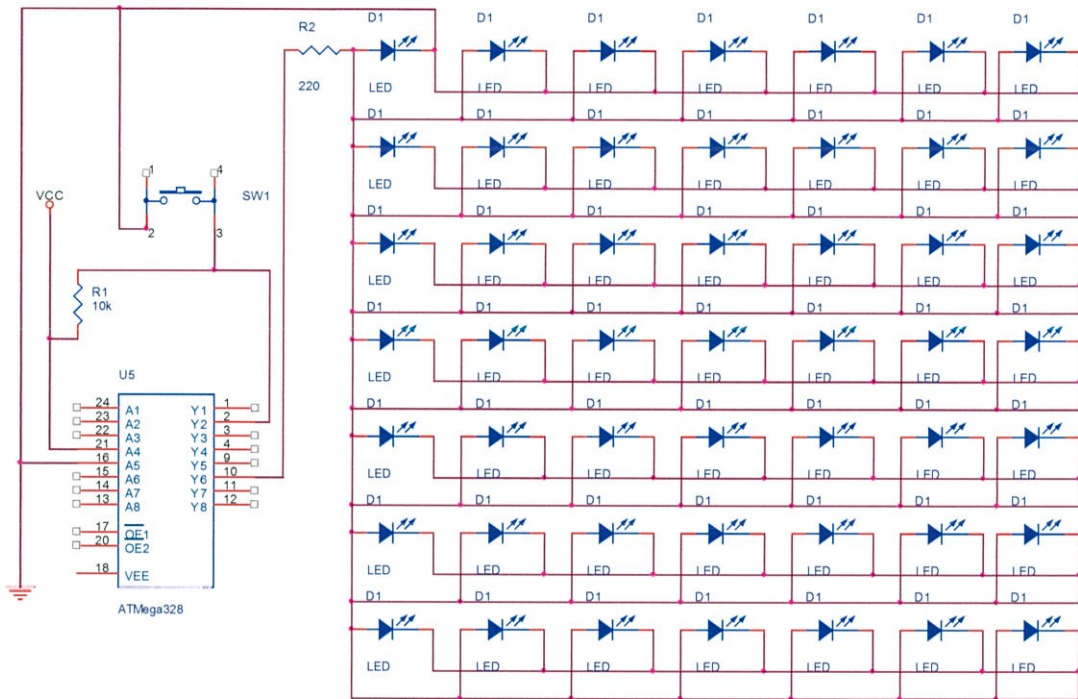


รูปที่ 3.2 วงจรจำลองหลอดไฟ LED จำนวน 49 หลอด

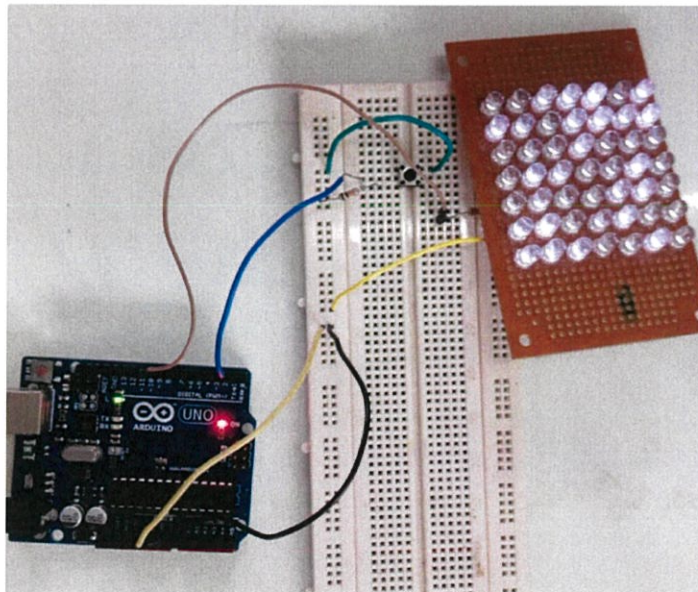


รูปที่ 3.3 หลอดไฟ LED ขนาด 7x7 จำนวนทั้งสิ้น 49 หลอด

สำหรับวงจรสำหรับเชื่อมต่อบอร์ด Arduino UNO เข้ากับวงจรสวิตช์เปิด-ปิดไฟโดยขาที่ 10 ของบอร์ด Arduino จะต่อเข้ากับตัวต้านทาน 220 ohm และ ขาที่ 2 ต่อเข้ากับตัวต้านทาน 10k ohm ที่ต่อเข้ากับขาที่ 1 ของสวิตช์



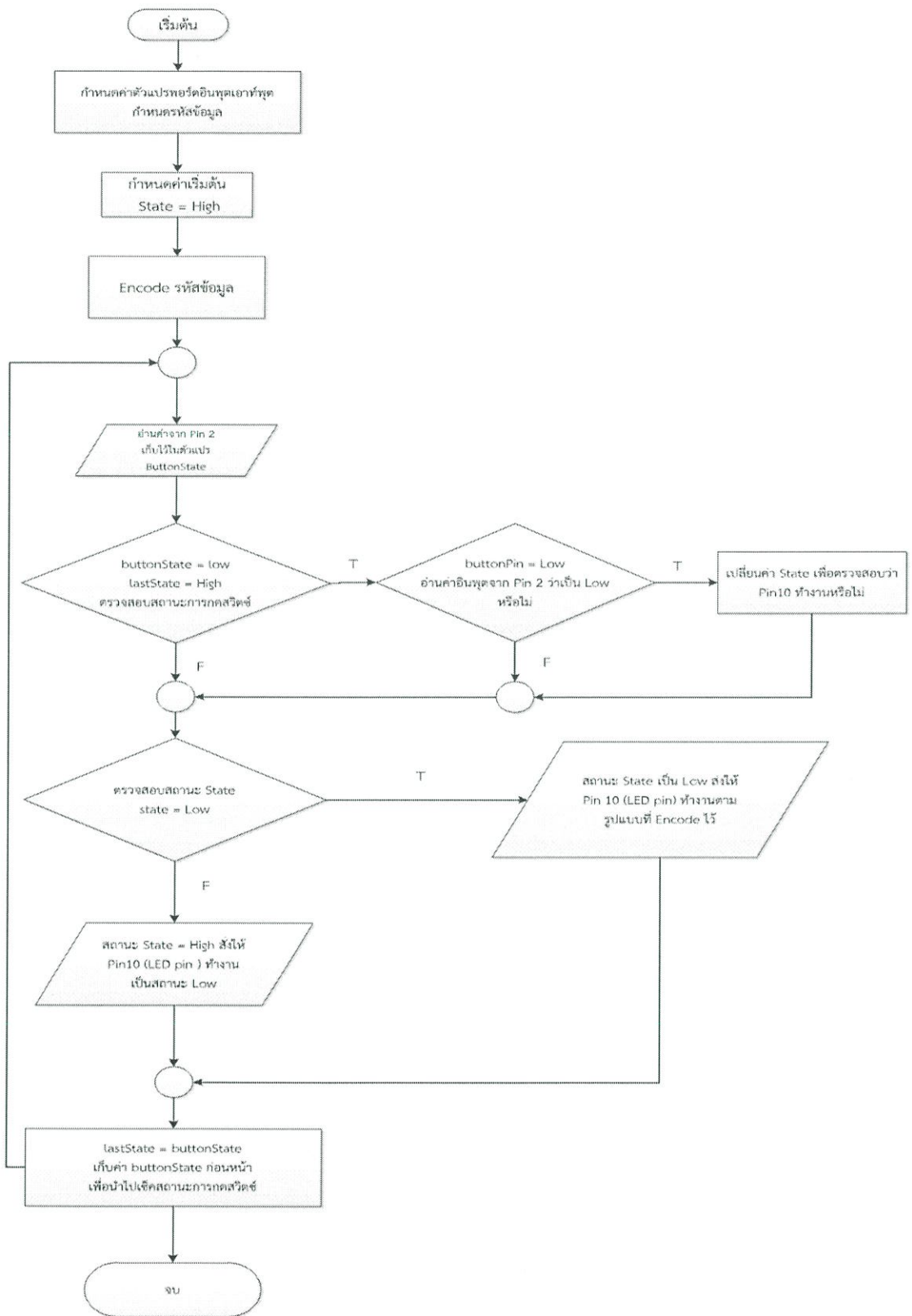
รูปที่ 3.4 การจำลองวงจรสวิตช์เปิดปิดไฟเชื่อมต่อเข้ากับบอร์ด Arduino



รูปที่ 3.5 วงจรสวิตช์เปิดปิดไฟเชื่อมต่อเข้ากับบอร์ด Arduino ( Microcontroller ATmega328 )

สำหรับการออกแบบโปรแกรมวงจรไฟกระพริบจะใช้โปรแกรม Arduino ซึ่งเป็นซอฟต์แวร์สำหรับเขียนโปรแกรมเพื่อควบคุมบอร์ด Arduino โดยตรงมาใช้โดยจะใช้งานพอร์ตของบอร์ด Arduino สองพอร์ต คือ พอร์ตที่ 2 ซึ่งจะใช้ควบคุมสวิทช์การเปิดปิดของไฟ และพอร์ตที่ 10 จะใช้สำหรับควบคุมการกระพริบของหลอดไฟ ซึ่งก่อนจะทำการส่งนั้น ต้องดูที่โปรแกรมในส่วนของภาครับก่อนว่าตอบสนองที่ Frame rate เท่าไหร่ ซึ่งจำนวนบิตที่จะส่งนั้น จะต้องไม่เกินความสามารถของค่า Fps ของกล้องที่สามารถรับภาพได้ ซึ่งภาครับและภาคส่งนั้นจะต้องสัมพันธ์กัน และบิตที่ทำการส่งไปนั้นจะกำหนดการติดดับของหลอดไฟตามบิต ซึ่งบิต 1 จะแทนไฟติด และบิต 0 จะแทนไฟดับ รูปแบบการกระพริบรูปแบบหนึ่ง หรือบิตที่ส่งไปรูปแบบหนึ่งนั้นจะแทนอัตลักษณ์ของสิ่งๆหนึ่ง เช่น คนๆหนึ่งมีอัตลักษณ์แบบหนึ่งแล้วบันทึกลงในฐานข้อมูล แล้วให้มีบิตหรือเรียกเป็นโค้ดประจำตัวคือบิต 1011 ซึ่งจะทำให้การโปรแกรมให้ไฟกระพริบเป็นรูปแบบ ติด-ดับ-ติด-ติด ใน 1 วินาที เมื่อกล้องตรวจจับการกระพริบของแสงแล้วได้บิต 1011 นี้มาก็จะทำการตรวจสอบกับฐานข้อมูลที่มี Primary key เป็นโค้ดนี้ว่าตรงกับโค้ดตัวใดในฐานข้อมูล ก็จะแสดงอัตลักษณ์ของสิ่งๆนั้นขึ้นมา

รูปที่ 3.6 แสดงรูปโฟลว์ชาร์ตการทำงานของโปรแกรมโดยเริ่มจากกำหนดตัวแปรพอร์ต อินพุต เอาท์พุต และบิตที่ต้องการจะส่งก่อนโดยจะเป็นอาร์เรย์บิต 1 คือไฟติดและบิต 0 คือไฟดับ เพื่อให้สามารถควบคุมการทำงานได้ และกำหนดค่าเริ่มต้นตัวแปร State ให้เป็น High แล้วทำการเข้ารหัสข้อมูลเพื่อให้ไฟที่กระพริบออกไปนั้นไม่ใช่ข้อมูลอัตลักษณ์นั้นๆโดยตรง เป็นการรักษาความปลอดภัยเบื้องต้น จากนั้นจะทำการอ่านค่าจาก Pin 2 แล้วเก็บไว้ที่ตัวแปร Button state หาก Button state เป็นค่า Low และ Last state เป็น High จะทำการตรวจสอบสถานะการกดสวิทช์ ว่ามีการกดสวิทช์หรือไม่ หากมีการกดจะให้ตัวแปร Buttonpin เป็น Low และอ่านค่าอินพุตจาก Pin 2 ว่าเป็น :ow หรือไม่ หากเป็นจะทำการเปลี่ยน State เพื่อตรวจสอบว่า Pin10 ทำงานหรือไม่ หากไม่มีการกดสวิทช์หรืออินพุตจาก Pin2 ไม่เป็นค่า Low จะทำการตรวจสอบสถานะของ State ถ้า State = Low จะสั่งการให้ Pin 10 ทำงานควบคุมไฟกระพริบตามรูปแบบที่เข้ารหัสไว้ หาก State = High จะสั่งการให้ Pin 10 ทำงานเป็นสถานะ Low แทน จากนั้นกำหนด Last state = Button state เพื่อเก็บค่า Button state ก่อนหน้าเพื่อตรวจสอบสถานะการกดสวิทช์ ซึ่งการทำงานเหล่านี้มีโฟลว์ชาร์ตการทำงานดังรูปที่ 3.6



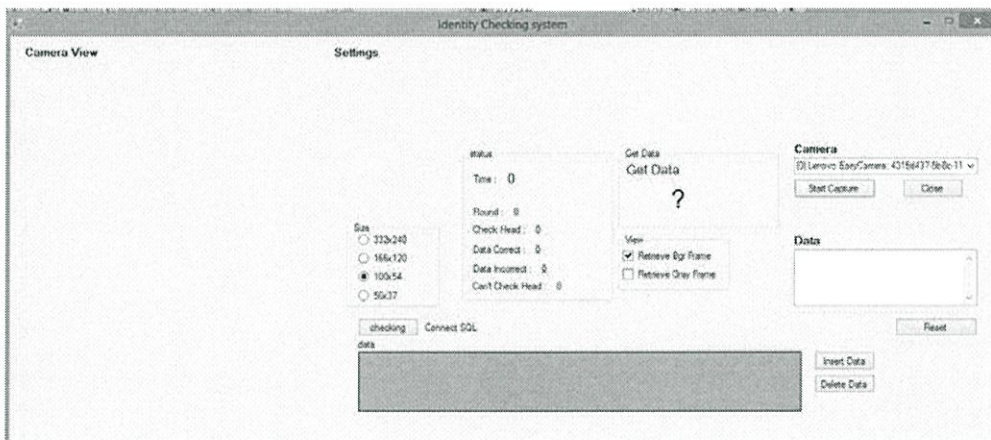
รูปที่ 3.6 โฟลว์ชาร์ตการทำงานของโปรแกรมไฟกระพริบ

### 3.4 การออกแบบในส่วนของภาครับ

ในส่วนของภาครับจะประกอบไปด้วยกล่องที่ตรวจจับการกระพริบของแสงมายังโปรแกรมเชิงประมวลผลภาพที่ใช้ C# ในการเขียนโปรแกรมโดยใช้ร่วมกับไลบรารี OpenCV, EmguCV และ AForge.NET และนำโค้ดที่ตรวจจับได้มาตรวจสอบความเป็นอัตลักษณ์กับฐานข้อมูลว่ามีข้อมูลตรงกันหรือไม่

#### 3.4.1 การออกแบบอินเตอร์เฟซ

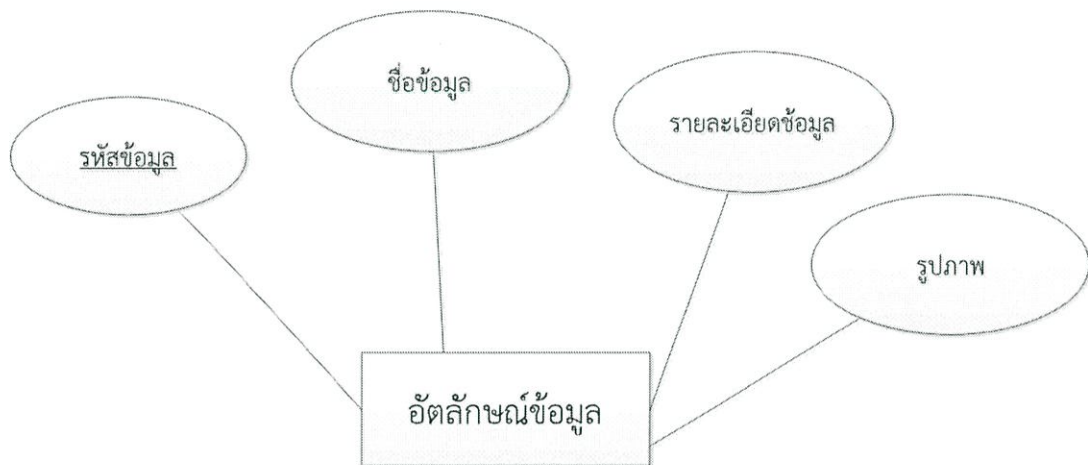
สำหรับอินเตอร์เฟซที่ออกแบบจะแบ่งเป็นสามส่วน คือ ส่วนแสดงผลภาพที่ตรวจจับการกระพริบของแสง ส่วนที่แสดงผลข้อมูลที่รับมาได้ และส่วนที่ทำการเชื่อมต่อกับฐานข้อมูลเพื่อตรวจสอบความเป็นอัตลักษณ์ของข้อมูล



รูปที่ 3.7 ภาพอินเตอร์เฟซของโปรแกรมโดยใช้โปรแกรม C#

### 3.4.2 การออกแบบฐานข้อมูล

สำหรับฐานข้อมูลจะใช้ MySQL ในการจัดการโดยในตารางจะประกอบไปด้วย รหัสข้อมูล ชื่อข้อมูล รายละเอียดข้อมูลและรูปภาพ โดยมี Primary Key คือรหัสข้อมูล ซึ่งเป็นค่าจากการประมวลผลภาพที่รับมาได้จากภาคส่งซึ่งเมื่อรับข้อมูลมาได้อีกจะนำมาตรวจสอบกับฐานข้อมูลว่าข้อมูลที่รับมานั้นมีอัตลักษณ์ที่ตรงกับในฐานข้อมูลหรือไม่ และสามารถมีส่วนที่สามารถจัดการฐานข้อมูลได้ เช่นเพิ่มข้อมูลในฐานข้อมูล ลบข้อมูล และตรวจสอบข้อมูลในฐานข้อมูลได้ แต่อย่างไรก็ตาม หากนำไปใช้กับระบบอื่นฐานข้อมูลนั้นก็ยังสามารถเปลี่ยนแปลงได้ตามความเหมาะสม



รูปที่ 3.8 แผนภาพ ER Diagram ฐานข้อมูลของระบบตรวจสอบความเป็นอัตลักษณ์

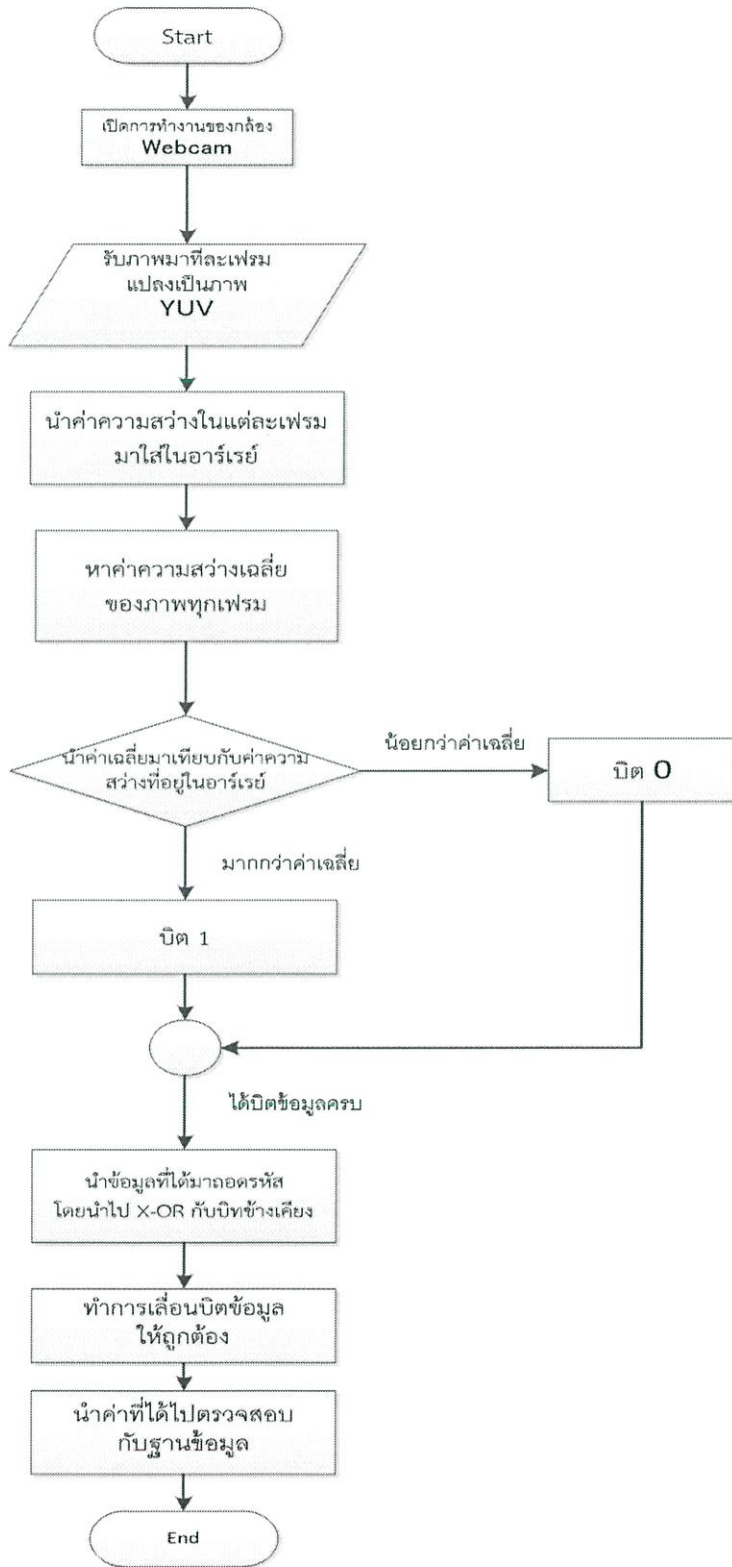
ตารางที่ 3 ตัวอย่างข้อมูลในฐานข้อมูลที่ใช้งาน

ชื่อข้อมูล	รหัสข้อมูล	รายละเอียดข้อมูล	รูปภาพ
KMITL	52	www.kmitl.ac.th	

ดังนั้น ข้อมูล KMITL จะมีรหัสข้อมูลเป็น 52 และมีอัตลักษณ์ดังรายละเอียดข้อมูลและรูปภาพ และทำการโปรแกรมรูปแบบไฟกระพริบตามรหัสข้อมูล 52 ไป เมื่อนำกล้องตรวจจับแล้วจะได้ข้อมูลหรืออัตลักษณ์มาดังตัวอย่าง

### 3.4.3 การออกแบบโปรแกรมประมวลผลภาพ

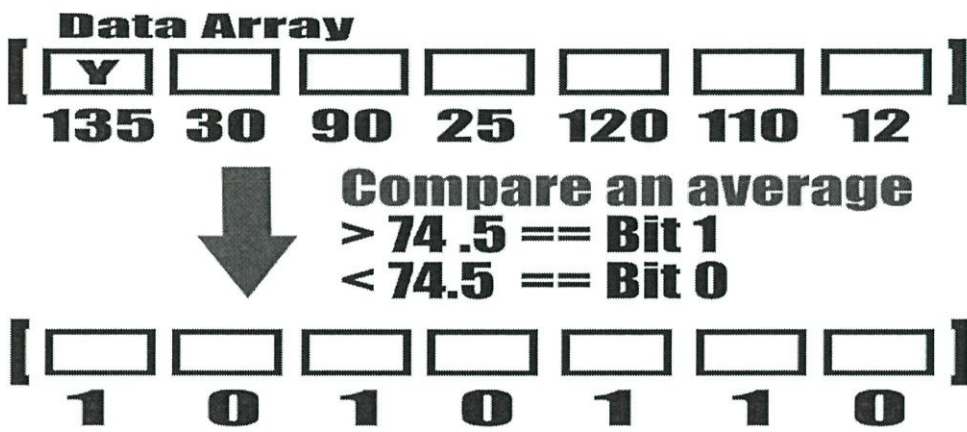
รูปที่ 3.9 แสดงแผนภาพโฟลว์ชาร์ตการทำงานของโปรแกรม ดังนี้



รูปที่ 3.9 โฟลว์ชาร์ตการทำงานของโปรแกรมประมวลผลภาพ

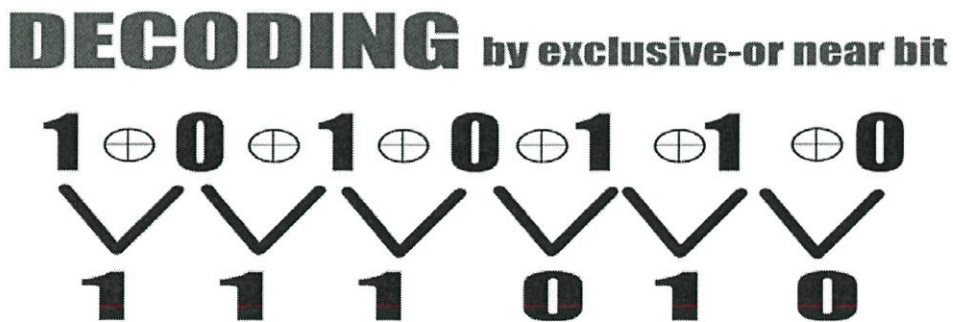
สำหรับโปรแกรมการตรวจจ็กรูปแบบการกระพริบของไฟนั้นจะส่งข้อมูลที่เป็นไบนารีบิต โดยจะสร้างขอบเขตบริเวณที่ต้องการจับการกระพริบในภาพเพื่อลดการประมวลผล ซึ่งจะนำภาพที่รับมาได้นั้นแปลงเป็นข้อมูลแล้วนำข้อมูลไปวิเคราะห์โดยการตรวจจ็กรเปลี่ยนแปลงของแสงโดยวัดจากค่าความสว่างของภาพ เช่น ถ้าภาคส่งส่งด้วยความถี่ 10 Hz. ภาครับจะรับด้วยอัตรา 10 frame/s เมื่อรับภาพมาได้แล้วจำแปลงเป็นระบบสี YUV ก่อน จากนั้นจะนำค่าความสว่างของภาพแต่ละเฟรมมาเก็บไว้ในอาร์เรย์ก่อน แล้วจึงหาค่าความสว่างเฉลี่ยของภาพ

สำหรับการประมวลผลภาพจะใช้หลักการเทียบความสว่างของภาพกับค่าเฉลี่ยที่เก็บไว้ในอาร์เรย์ เช่น หากรับภาพมา 10 เฟรมจะหาค่าเฉลี่ยความสว่างของภาพทั้งหมด 10 เฟรมก่อน แล้วจึงนำค่าความสว่างแต่ละเฟรมมาเปรียบเทียบกับค่าเฉลี่ยถ้ามีค่าน้อยกว่าค่าเฉลี่ยให้เป็นบิต 0 ถ้ามีค่ามากกว่าค่าเฉลี่ยให้เป็นบิต 1



รูปที่ 3.10 แผนภาพหลักการประมวลผล

เมื่อได้รับบิตข้อมูลมาให้ทำการถอดรหัส ด้วยการ Exclusive-OR กับบิตข้างเคียง จะได้บิตข้อมูลที่แท้จริงมา



รูปที่ 3.11 การถอดรหัสบิตข้อมูล

และอาจมีบางกรณีที่ภาครับภาคส่งอาจมีการซิงโครไนส์ระหว่างภาครับภาคส่งไม่เท่ากัน ทำให้ไม่ทราบ Header หรือ Data bit จึงต้องมีการเลื่อนบิตเพื่อหา Header bit เช่น 10 บิต แบ่งเป็น Header bit 3 บิต Databit 5 บิต Close bit 2 บิต กำหนดให้ Header bit เป็น 111 และ Close bit 00 Data bit 10101 เช่น หากข้อมูลที่ได้รับได้มาเป็นซีควเอนซ์ดังนี้ 110101001111010100111101010011 จึงต้องทำการเลื่อนบิตจนกว่าจะพบ Header bit เป็น 111 และ Close bit เป็น 00 นั่นคือบิต 110101001111010100111101010011 ทำการเลื่อนบิต << 1111010100 111101010011 แล้วนำ Data bit 10101 แปลงเป็นเลขฐาน 10 แล้วนำไปตรวจสอบกับฐานข้อมูล

### 3.5 กล้องเว็บแคม

ในการทดลองจะใช้กล้องเพื่อทำการตรวจจับการกระพริบของแสง ซึ่งในการทดลองนี้จะใช้กล้องเว็บแคมรุ่น OKER 177 มาใช้ในการทดลองดังรูป 3.12



รูปที่ 3.12 กล้องเว็บแคมรุ่น OKER 177 ที่นำไปใช้ในการตรวจจับการกระพริบของแสง

### 3.6 การลดอัตราความผิดพลาด และการเพิ่มอัตราการตรวจจับการกระพริบของแสง

ในการทดลองนั้นย่อมมีความผิดพลาดอยู่ของการตรวจจับ และในตอนกลางวันที่มีระยะไกลนั้น จะไม่สามารถตรวจจับได้ จึงได้คิดวิธีการที่จะลดปัญหาดังกล่าวทั้งสอง โดยวิธีต่างๆนี้ได้แก่

1. การลดความผิดพลาดโดยการตรวจสอบความผิดพลาดของบิตโดยใช้การเข้ารหัสแบบ Block Code
2. การลดความผิดพลาดการตรวจสอบความผิดพลาดของบิตโดยใช้การเข้ารหัสแบบ Convolutional Code
3. การลดความผิดพลาดโดยการลดจำนวนบิตที่ใช้ส่ง
4. การเพิ่มอัตราการตรวจจับโดยใช้เทคนิคการซูมภาพ
5. การเพิ่มอัตราการตรวจจับโดยใช้เทคนิคการลดรอบการประมวลผล
6. การเพิ่มอัตราการตรวจจับโดยการเพิ่มขนาดของตัวส่ง
7. การเพิ่มอัตราการตรวจจับโดยทดลองกับหลอดแอลอีดีสีต่างๆ
8. การลดความผิดพลาดโดยใช้การตรวจสอบจากการตรวจจับสองครั้ง

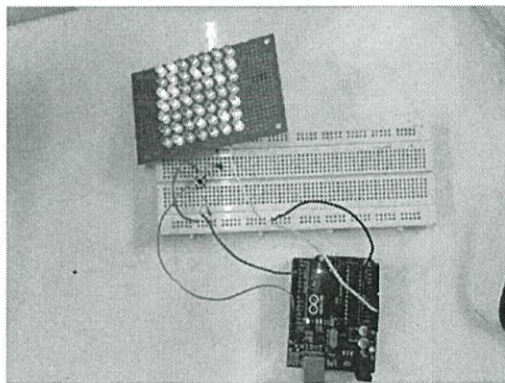
## บทที่ 4

### ผลการทดลอง

ในบทนี้จะกล่าวถึงวิธีการติดตั้งอุปกรณ์ภาครับและภาคส่ง วิธีการทดลองในส่วนของภาครับและภาคส่ง โดยจะมีผลการทดลองจากการตรวจจับการกระพริบของแสง ค่าความผิดพลาด และวิธีในการเพิ่มอัตราการตรวจจับและวิธีในการลดความผิดพลาด

#### 4.1 การทดลองในส่วนของภาคส่ง

ใช้วงจรตัวส่งโดยใช้บอร์ด Arduino ที่ทำการโปรแกรมเรียบร้อยแล้วต่อกับวงจร LED ขนาด 5x5 ตารางเซนติเมตรจำนวน 49 ดังรูปที่ 4.1



รูปที่ 4.1 ภาพวงจรในส่วนของภาคส่ง

ใช้โปรแกรม Arduino 1.0.5 เป็นโปรแกรมในการเขียนโปรแกรมเพื่อควบคุมบอร์ด Arduino จากนั้นจึงทำการกำหนดค่าบิตที่ต้องการในโปรแกรมภาคส่ง โดยกำหนดในอาร์เรย์ข้อมูลว่าโดยให้บิต 1 คือไฟติด และบิต 0 คือไฟดับ ทำการคอมไพล์และเบิร์นโปรแกรมลงบอร์ด Arduino ดังรูปที่ 4.2



รูปที่ 4.2 โปรแกรม Arduino 1.0.5







รูปที่ 4.7 ตัวส่งที่ระยะห่างจากตัวรับระยะ 1 เมตร และ 5 เมตรในตอนกลางวัน



รูปที่ 4.8 ตัวส่งที่ระยะห่างจากตัวรับระยะ 10 เมตร และ 20 เมตรในตอนกลางวัน



รูปที่ 4.9 ตัวส่งที่ระยะห่างจากตัวรับระยะ 30 เมตร และ 40 เมตรในตอนกลางวัน

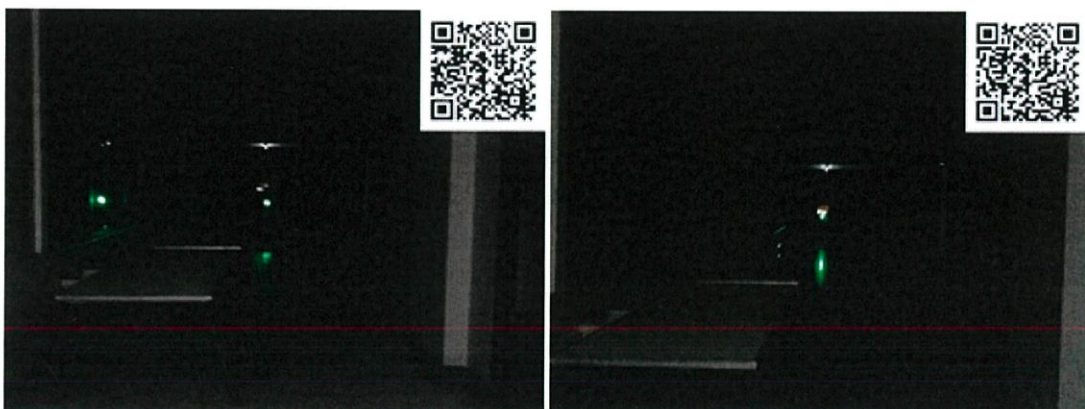


รูปที่ 4.10 ตัวส่งไฟกระพริบที่ระยะห่างจากตัวรับระยะ 50 เมตรในตอนกลางวัน

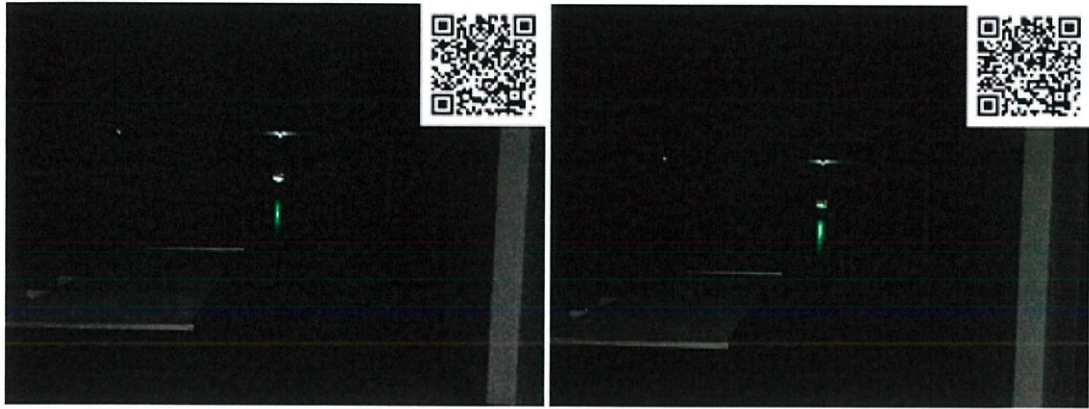
รูปที่ 4.11 – 4.14 ซึ่งแสดงรูปการทดลองระยะทางระหว่างตัวส่งไฟกระพริบกับตัวรับที่ระยะต่าง ๆ ในเวลากลางคืน สถานที่ทดลอง ณ บริเวณโถงชั้น 11 อาคาร 12 ชั้น คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง



รูปที่ 4.11 ตัวส่งที่ระยะห่างจากตัวรับระยะ 1 เมตร และ 5 เมตรในตอนกลางคืน



รูปที่ 4.12 ตัวส่งที่ระยะห่างจากตัวรับระยะ 10 เมตร และ 20 เมตรในตอนกลางคืน



รูปที่ 4.13 ตัวส่งที่ระยะห่างจากตัวรับระยะ 30 เมตร และ 40 เมตรในตอนกลางคืน



รูปที่ 4.14 ตัวส่งที่ระยะห่างจากตัวรับระยะ 50 เมตร ในตอนกลางคืน

ในการทดลองได้ทำการทดลองในเวลากลางวันและกลางคืนซึ่งได้ผลสรุปดังตารางที่ 4.1 และ 4.2 ดังนี้

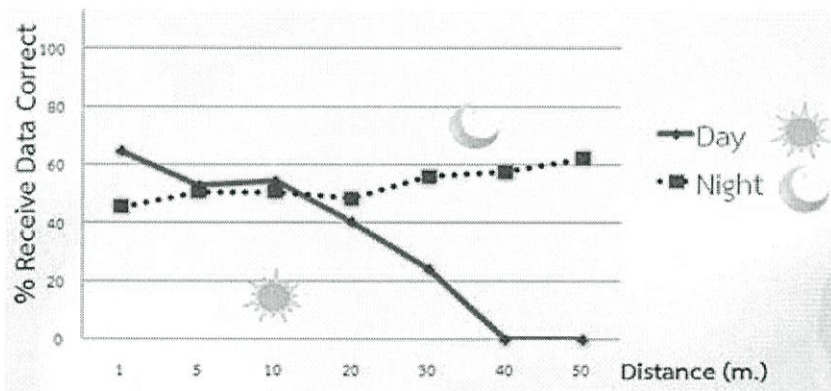
ตารางที่ 4.1 ผลการทดลองการตรวจจับที่ระยะต่างๆในเวลากลางวัน

Distance (m.)	1	5	10	20	30	40	50
Number of times to receive data	65	68	66	67	66	66	66
Correct data	42	36	36	27	16	0	0
Cannot receive data	16	25	23	35	33	56	62
Error data	7	7	7	5	14	10	4
% receive correct data	64.62	52.94	54.55	40.30	24.24	0.00	0.00
% cannot receive data	24.62	36.76	34.85	52.24	50.00	84.85	93.94
% receive error data	10.77	10.29	10.61	7.46	21.21	15.15	6.06

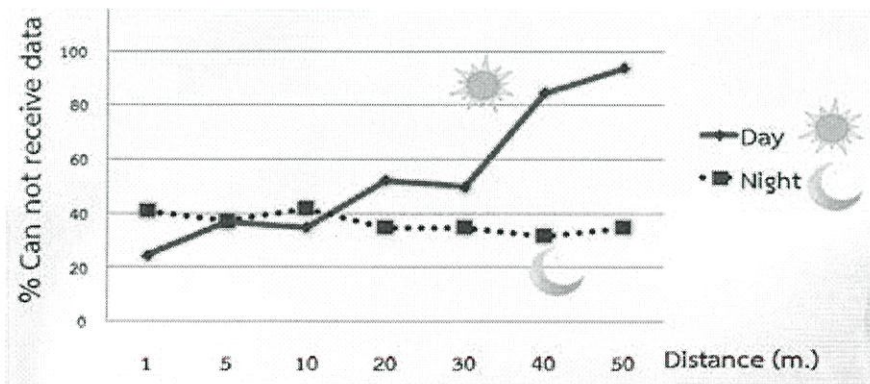
ตารางที่ 4.2 ผลการทดลองการตรวจจับที่ระยะต่างๆในเวลากลางคืน

Distance (m.)	1	5	10	20	30	40	50
Number of times to receive data	66	66	66	66	66	66	66
Correct data	30	34	34	32	37	38	41
Cannot receive data	27	25	28	23	23	21	23
Error data	8	3	7	8	9	6	7
% receive correct data	45.45	50.75	50.75	48.48	56.06	57.58	61.12
% cannot receive data	40.91	37.31	41.79	34.85	34.85	31.82	34.85
% receive error data	12.12	4.48	10.45	12.12	13.64	9.09	10.61

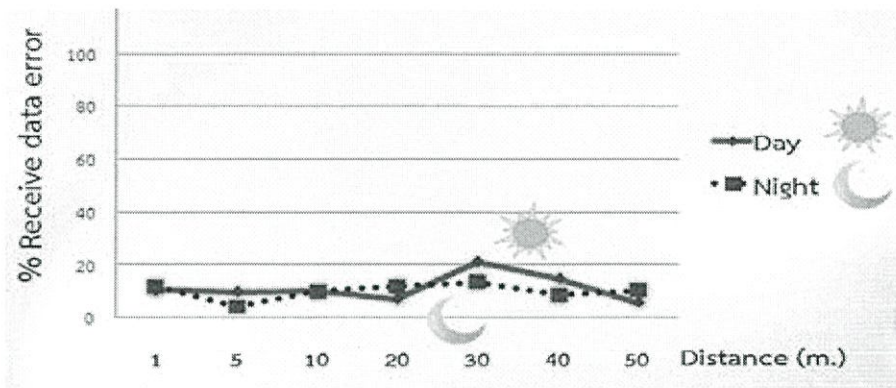
รูปที่ 4.15 – 4.18 แสดงกราฟความสัมพันธ์ระหว่างร้อยละการรับข้อมูลได้ถูกต้อง การไม่สามารถรับข้อมูลได้ และการรับข้อมูลได้ผิดพลาดกับระยะจากเครื่องรับในเวลากลางวันและกลางคืน



รูปที่ 4.15 กราฟความสัมพันธ์ระหว่างร้อยละการรับข้อมูลได้ถูกต้องกับระยะจากเครื่องรับในเวลากลางวันและกลางคืน



รูปที่ 4.16 กราฟความสัมพันธ์ระหว่างร้อยละการไม่สามารถรับข้อมูลได้ถูกต้องกับระยะจากเครื่องรับในเวลากลางวันและกลางคืน



รูปที่ 4.17 กราฟความสัมพันธ์ระหว่างร้อยละการรับข้อมูลได้ผิดพลาดกับระยะจากเครื่องรับในเวลากลางวันและกลางคืน

จากการทดลอง จะเห็นได้ว่าในเวลากลางวัน ซึ่งมีความสว่างของสภาพแวดล้อมมาก เมื่อระยะไกลขึ้นความสามารถในการตรวจจับจะมีค่าลดลง จนกระทั่งไม่สามารถตรวจจับได้เลย และในเวลากลางคืน ซึ่งมีความสว่างของสภาพแวดล้อมน้อย จะมีความสามารถในการตรวจจับมาก และความสามารถในการตรวจจับจะลดลงเพียงเล็กน้อยในระยะที่ไกลขึ้น

ดังนั้น จะเห็นว่าความแตกต่างในเวลากลางวันและกลางคืน คือ ในเวลากลางวันความสามารถในการตรวจจับจะน้อยกว่าเวลากลางคืน เพราะมีค่าความสว่างของสภาพแวดล้อมมาก ทำให้มีผลต่อค่าความสว่างเฉลี่ยที่นำมาใช้ในการคำนวณมากกว่าในเวลากลางคืน จึงได้คิดค้นวิธีการต่างๆขึ้นมาเพื่อลดความผิดพลาด และเพิ่มอัตราการตรวจจับในตอนกลางวันให้มากขึ้นในหัวข้อต่อไป

#### 4.4 การลดอัตราความผิดพลาด และการเพิ่มอัตราการตรวจจับการกระพริบของแสง

จากผลการทดลองข้างต้น จะเห็นว่าผลการทดลองนั้นยังมีความผิดพลาดอยู่ และในตอนกลางวันในระยะไกลนั้น จะไม่สามารถตรวจจับได้ จึงได้คิดวิธีการที่จะลดปัญหาดังกล่าวทั้งสองโดยวิธีต่างๆนี้ได้แก่

1. การลดความผิดพลาดโดยการตรวจสอบความผิดพลาดของบิตโดยใช้การเข้ารหัสแบบ Block Code
2. การลดความผิดพลาดการตรวจสอบความผิดพลาดของบิตโดยใช้การเข้ารหัสแบบ Convolutional Code
3. การลดความผิดพลาดโดยการลดจำนวนบิตที่ใช้ส่ง
4. การเพิ่มอัตราการตรวจจับโดยใช้เทคนิคการซูมภาพ
5. การเพิ่มอัตราการตรวจจับโดยใช้เทคนิคการลดกรอบการประมวลผล
6. การเพิ่มอัตราการตรวจจับโดยการเพิ่มขนาดของตัวส่ง
7. การเพิ่มอัตราการตรวจจับโดยทดลองกับแอลอีดีสีต่างๆ
8. การลดความผิดพลาดโดยใช้การตรวจสอบจากการตรวจจับสองครั้ง

##### 4.4.1 การลดความผิดพลาดโดยใช้การเข้ารหัสแบบ Block Code

ในการเข้ารหัสแบบ Block Code ในภาคส่งจะถูกเข้ารหัส และจะมีการถอดรหัสและแก้ไขบิตที่ผิดพลาดที่ด้านรับ ในการทดลองได้ใช้หลอด LED สีเขียวขนาด 7x7 ตัวส่งจำนวน 26 บิต ทดลองในเวลากลางวันและกลางคืน หาอัตราการตรวจจับใน 1 นาที ซึ่งได้ผลการทดลองดังตารางที่ 4.3 และ 4.4 ดังนี้

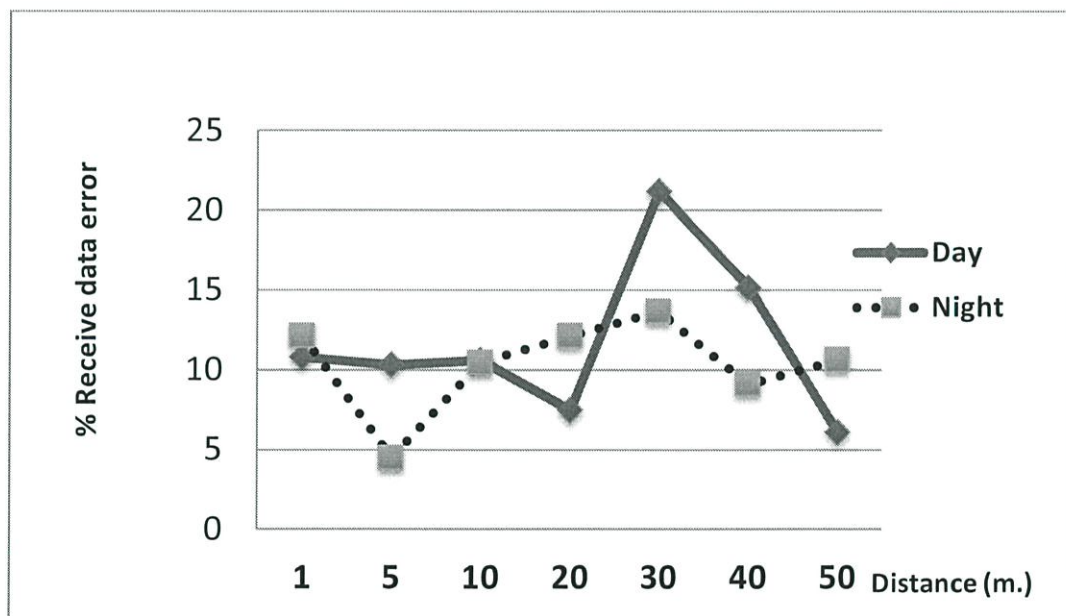
ตารางที่ 4.3 ผลการทดลองการตรวจจับที่ระยะต่างๆในเวลากลางวันโดยใช้การเข้ารหัสแบบ Block Code

Distance (m.)	1	5	10	20	30	40	50
Number of times to receive data	65	65	66	67	66	66	67
Correct data	39	37	35	27	2	0	0
Cannot receive data	20	19	23	30	59	65	66
Error data	6	9	8	10	5	1	1
% receive correct data	60.00	56.92	53.03	40.30	3.03	0.00	0.00
% cannot receive data	30.77	29.23	34.85	44.78	89.39	98.48	98.51
% receive error data	9.23	13.85	12.12	14.93	7.58	1.52	1.49

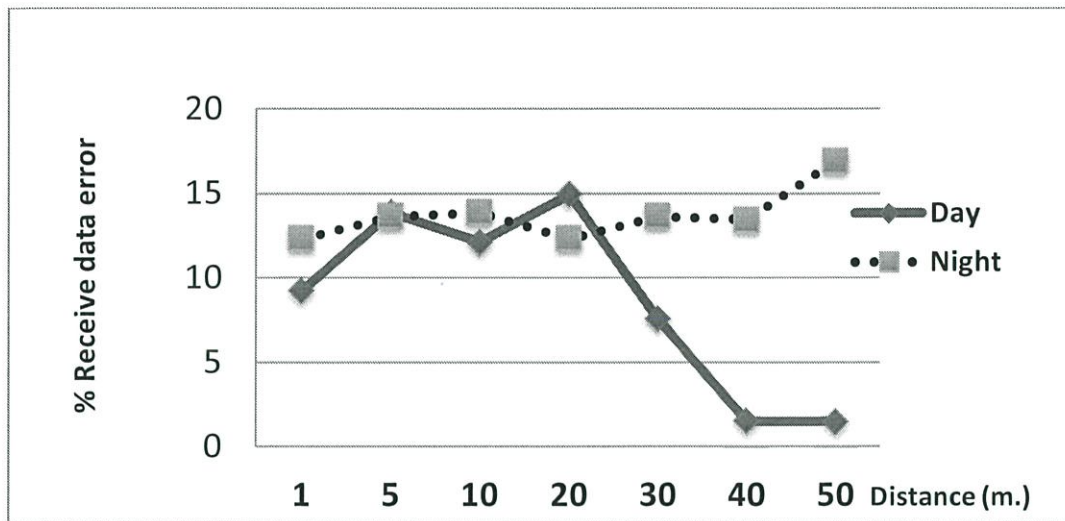
ตารางที่ 4.4 ผลการทดลองการตรวจจับที่ระยะต่างๆในเวลากลางคืนโดยใช้การเข้ารหัสแบบ Block Code

Distance (m.)	1	5	10	20	30	40	50
Number of times to receive data	65	66	65	65	66	67	65
Correct data	37	36	41	34	38	36	31
Cannot receive data	20	21	15	23	19	22	23
Error data	8	9	9	8	9	9	11
% receive correct data	56.92	54.55	63.08	52.31	57.58	53.73	47.69
% cannot receive data	30.77	31.82	23.08	35.38	28.79	32.84	35.38
% receive error data	12.31	13.64	13.85	12.31	13.64	13.43	16.92

รูปที่ 4.18 - 4.19 แสดงกราฟความสัมพันธ์ระหว่างร้อยละการรับข้อมูลได้ผิดพลาดกับระยะจากเครื่องรับในเวลากลางวันและกลางคืน โดยรูปที่ 4.18 จะตรวจจับธรรมดา แต่รูปที่ 4.19 จะใช้การเข้ารหัสแบบ Block Code



รูปที่ 4.18 กราฟความสัมพันธ์ระหว่างร้อยละการรับข้อมูลได้ผิดพลาดกับระยะจากเครื่องรับในเวลากลางวันและกลางคืน



รูปที่ 4.19 กราฟความสัมพันธ์ระหว่างร้อยละการรับข้อมูลได้ผิดพลาดกับระยะจากเครื่องรับในเวลากลางวันและกลางคืนโดยใช้การเข้ารหัสแบบ Block Code

ผลการทดลอง พบว่า โปรแกรมนั้นมีการแก้ไขความผิดพลาดของบิต แต่มีการแก้ไขเพียงบิตเดียว โดยรวมจึงทำให้ไม่มีประสิทธิภาพในการแก้ไขบิตที่ผิดพลาด ทำให้ค่าความผิดพลาดมีอัตราที่สูงอยู่

#### 4.4.2 การลดความผิดพลาดโดยใช้การเข้ารหัสแบบ Convolutional Code

ในการเข้ารหัสแบบ Convolutional Code ในภาคส่งจะถูกเข้ารหัส และจะมีการถอดรหัสและแก้ไขบิตที่ผิดพลาดที่ด้านรับ ในการทดลองได้ใช้ตัวส่ง LED สีเขียวขนาด 7x7 จำนวน 26 บิต ทดลองในเวลากลางวันและกลางคืน ตรวจจับใน 1 นาที ซึ่งได้ผลการทดลองดังตารางที่ 4.5 และตารางที่ 4.6 ดังนี้

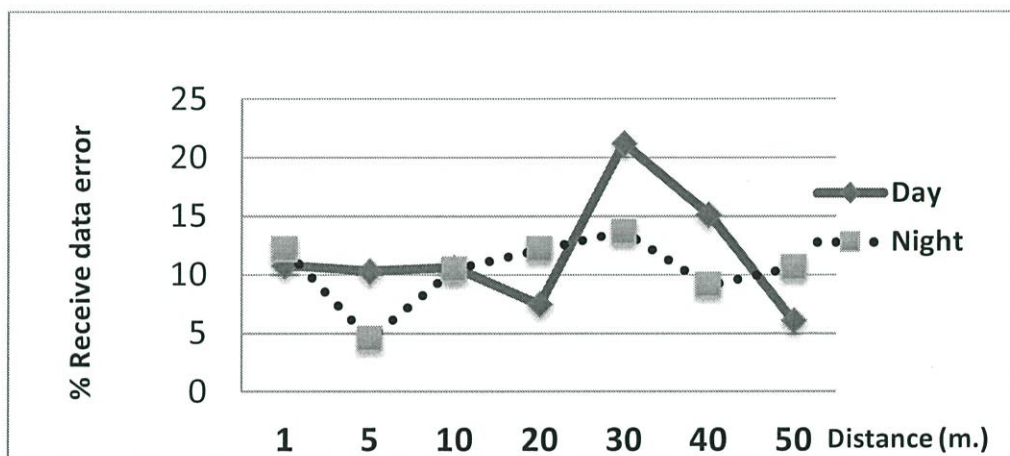
ตารางที่ 4.5 ผลการทดลองการตรวจจับที่ระยะต่างๆในเวลากลางวันโดยใช้การเข้ารหัสแบบ Convolutional Code

Distance (m.)	1	5	10	20	30	40	50
Number of times to receive data	66	66	66	66	66	66	66
Correct data	41	39	32	29	5	0	0
Cannot receive data	20	23	22	17	59	61	64
Error data	4	2	8	8	2	5	1
% receive correct data	62.12	59.09	48.48	43.94	7.58	0.00	0.00
% cannot receive data	30.30	34.85	33.33	25.76	89.39	92.42	96.97
% receive error data	6.06	3.03	12.12	12.12	3.03	7.58	1.52

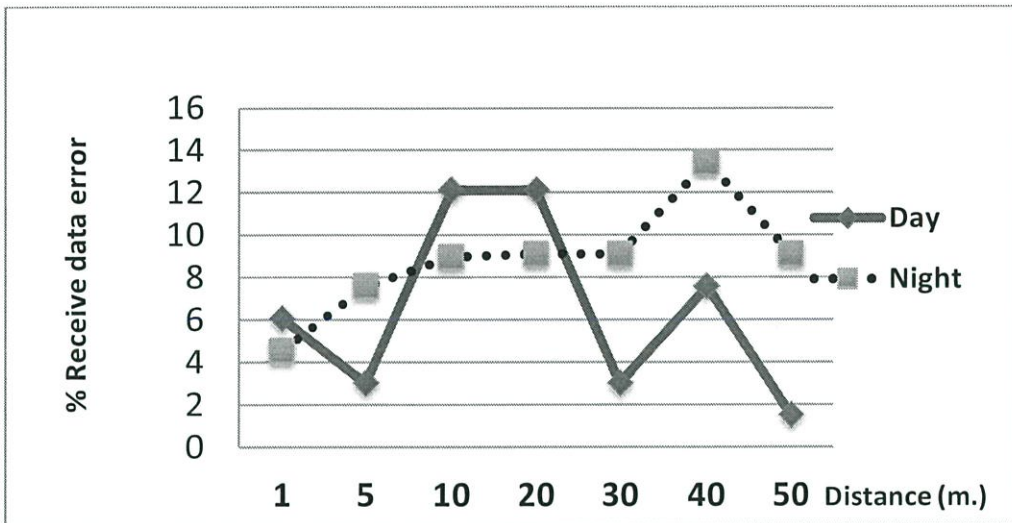
ตารางที่ 4.6 ผลการทดลองการตรวจจับที่ระยะต่างๆในเวลากลางคืนโดยใช้การเข้ารหัสแบบ Convolutional Code

Distance (m.)	1	5	10	20	30	40	50
Number of times to receive data	66	66	67	66	66	67	66
Correct data	34	33	30	33	33	32	27
Cannot receive data	29	25	29	25	26	24	31
Error data	3	5	6	6	6	9	6
% receive correct data	51.52	50.00	44.78	50.00	50.00	47.76	40.91
% cannot receive data	43.94	37.88	43.28	37.88	39.39	35.82	46.97
% receive error data	4.55	7.58	8.96	9.09	9.09	13.43	9.09

รูปที่ 4.20 - 4.21 แสดงกราฟความสัมพันธ์ระหว่างร้อยละการรับข้อมูลได้ผิดพลาดกับระยะจากเครื่องรับในเวลากลางวันและกลางคืน โดยรูปที่ 4.20 จะตรวจจับธรรมดา แต่รูปที่ 4.21 จะใช้การเข้ารหัสแบบ Convolutional Code



รูปที่ 4.20 กราฟความสัมพันธ์ระหว่างร้อยละการรับข้อมูลได้ผิดพลาดกับระยะจากเครื่องรับในเวลากลางวันและกลางคืน



รูปที่ 4.21 กราฟความสัมพันธ์ระหว่างร้อยละการรับข้อมูลได้ผิดพลาดกับระยะจากเครื่องรับในเวลากลางวันและกลางคืนโดยใช้การเข้ารหัส Convolutional Code

ผลการทดลอง พบว่า โปรแกรมนี้มีการแก้ไขความผิดพลาดของบิต ซึ่งมีประสิทธิภาพมากกว่า Block Code แต่ก็ยังมีค่าความผิดพลาดในบางช่วงที่มีค่าสูงอยู่ แต่โดยรวมนั้นสามารถลดความผิดพลาดของบิตได้มากขึ้น

#### 4.4.3 การเพิ่มอัตราการจับได้โดยใช้เทคนิคการซูมภาพ

การซูมภาพ เป็นเทคนิคที่ใช้เพื่อเพิ่มอัตราการตรวจจับไฟกระพริบได้ในระยะที่ไกลจากเดิมที่ไม่สามารถจับได้ โดยจะทำการซูมหรือเพิ่มความยาวโฟกัสเพื่อให้เข้าใกล้ภาพมากขึ้น โดยจะซูมโดยใช้ดิจิทัลซูม หรือซูมที่เลนส์ก็ได้ เพื่อให้เข้าใกล้ไฟกระพริบมากขึ้น



รูปที่ 4.22 การตรวจจับเมื่อยังไม่ได้ซูมภาพ



รูปที่ 4.23 การตรวจจับเมื่อทำการซูมภาพ

โดยการทดลองจะทำการตรวจจับโดยใช้เทคนิคการซูมภาพในเวลากลางวัน และกลางคืน ได้ผลการทดลองในตอนกลางวันดังตารางที่ 4.7 และตอนกลางคืนดังตารางที่ 4.8 นี้

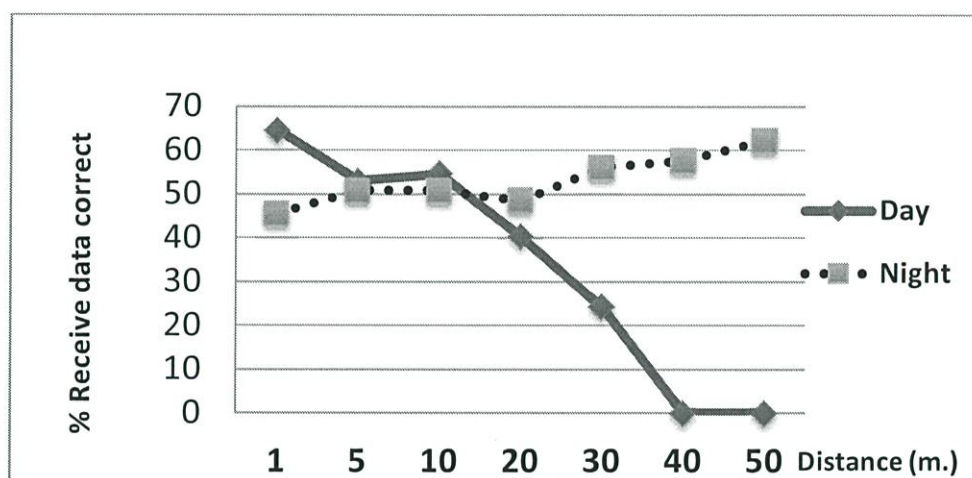
ตารางที่ 4.7 ผลการทดลองการตรวจจับที่ระยะต่างๆในเวลากลางวันโดยใช้เทคนิคการซูมภาพ

Distance (m.)	1	5	10	20	30	40	50
Number of times to receive data	65	68	66	67	66	66	66
Correct data	42	36	36	22	16	12	4
Cannot receive data	16	25	23	37	25	47	56
Error data	7	7	7	3	7	1	3
% receive correct data	64.62	52.94	54.55	32.84	24.24	18.18	6.06
% cannot receive data	24.62	36.76	34.85	55.22	37.88	71.21	84.85
% receive error data	10.77	10.29	10.61	4.48	10.61	1.52	4.55

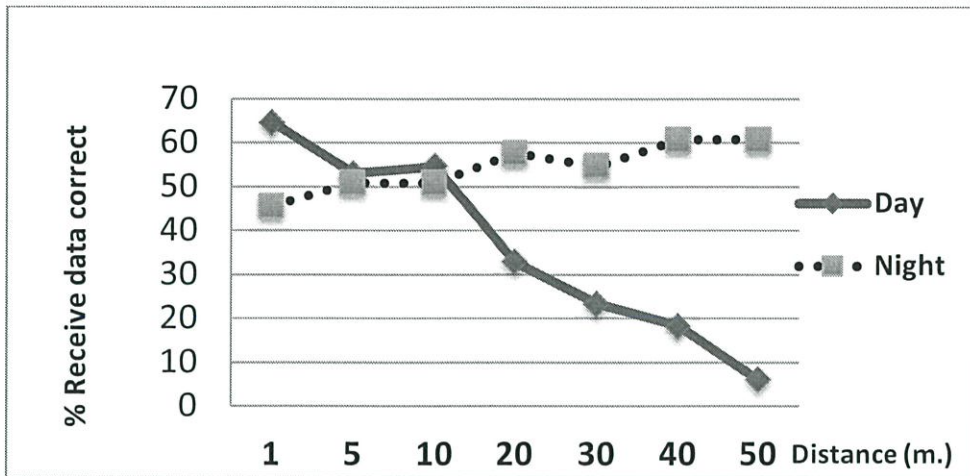
ตารางที่ 4.8 ผลการทดลองการตรวจจับที่ระยะต่างๆในเวลากลางคืนโดยใช้เทคนิคการซูมภาพ

Distance (m.)	1	5	10	20	30	40	50
Number of times to receive data	66	67	67	66	66	66	66
Correct data	30	34	34	38	36	40	40
Cannot receive data	27	25	28	21	20	20	21
Error data	9	8	5	7	9	6	5
% receive correct data	45.45	50.75	50.75	57.58	54.55	60.61	60.61
% cannot receive data	40.91	37.31	41.79	31.82	30.30	30.30	31.82
% receive error data	13.64	11.94	7.46	10.61	13.64	9.09	7.58

รูปที่ 4.24 – 4.25 แสดงกราฟความสัมพันธ์ระหว่างร้อยละการตรวจจับข้อมูลได้ถูกต้องกับระยะจากเครื่องรับในเวลากลางวันและกลางคืน โดยรูปที่ 4.24 จะตรวจจับธรรมดา แต่รูปที่ 4.25 จะใช้เทคนิคการซูมภาพ



รูปที่ 4.24 กราฟความสัมพันธ์ระหว่างร้อยละการรับข้อมูลได้ถูกต้องกับระยะจากเครื่องรับในเวลากลางวันและกลางคืน

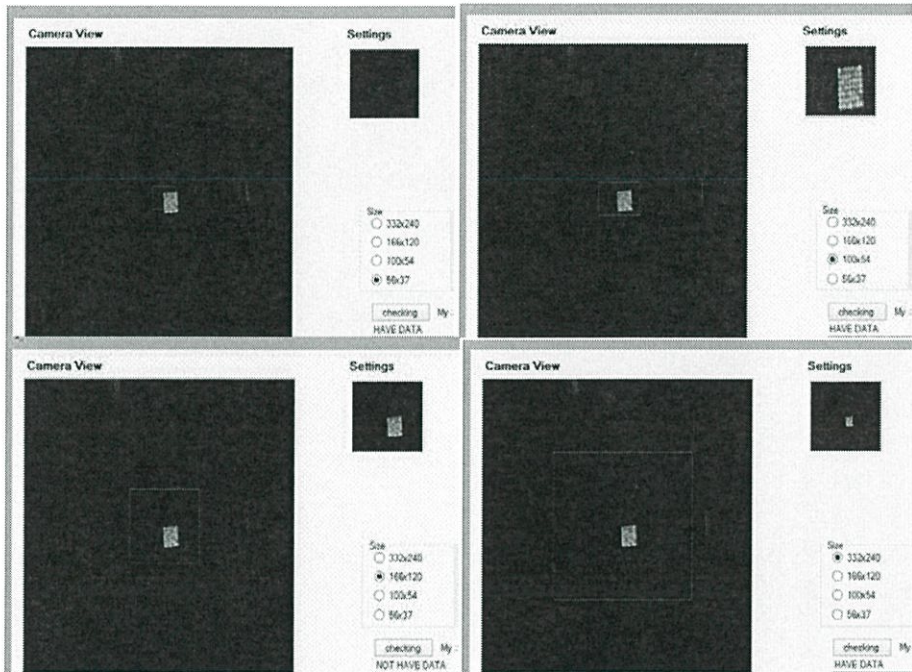


รูปที่ 4.25 กราฟความสัมพันธ์ระหว่างร้อยละการรับข้อมูลได้ถูกต้องกับระยะจากเครื่องรับ  
ในเวลากลางวันและกลางคืนโดยใช้เทคนิคการชุมภาพ

จากผลการทดลอง จะเห็นว่าเมื่อทำการชุมภาพนั้น ในเวลากลางวันนั้นที่ระยะไกล จะสามารถตรวจจับข้อมูลได้ จากเดิมที่ไม่สามารถตรวจจับได้เลย

#### 4.4.4 การเพิ่มอัตราการจับได้โดยใช้เทคนิคการลดกรอบการประมวลผล

โดยปกติการตรวจจับนั้น จะใช้ภาพที่แสดงขึ้นมาในกล่องรับภาพของโปรแกรม และใช้ค่าความสว่างของภาพนั้นมาคำนวณ จึงทำให้ค่าเฉลี่ยของแสงนั้นมากตามไปด้วยทำให้ตรวจจับได้ยาก จึงต้องสร้างกรอบขึ้นมาเพื่อลดการประมวลผลทำให้ใช้การคำนวณแสงสว่างเพียงแค่ภาพภายในกรอบนั้นๆ เพื่อให้เห็นตัวไฟกระพริบมากขึ้น และเพื่อให้สามารถเพิ่มอัตราการตรวจจับได้ โดยมีกรอบขนาดต่างๆ ตามความเหมาะสมของไฟกระพริบในขณะนั้น ดังภาพ



รูปที่ 4.26 กรอบการประมวลผลที่ขนาดต่างๆ

และเมื่อทำการทดลองโดยการลดกรอบการประมวลผลเป็นขนาดเล็กที่สุด (59x37) ทำการตรวจจับในเวลากลางวันและกลางคืน ได้ผลการทดลองในตอนกลางวันดังตารางที่ 4.9 และกลางคืนดังตารางที่ 4.10

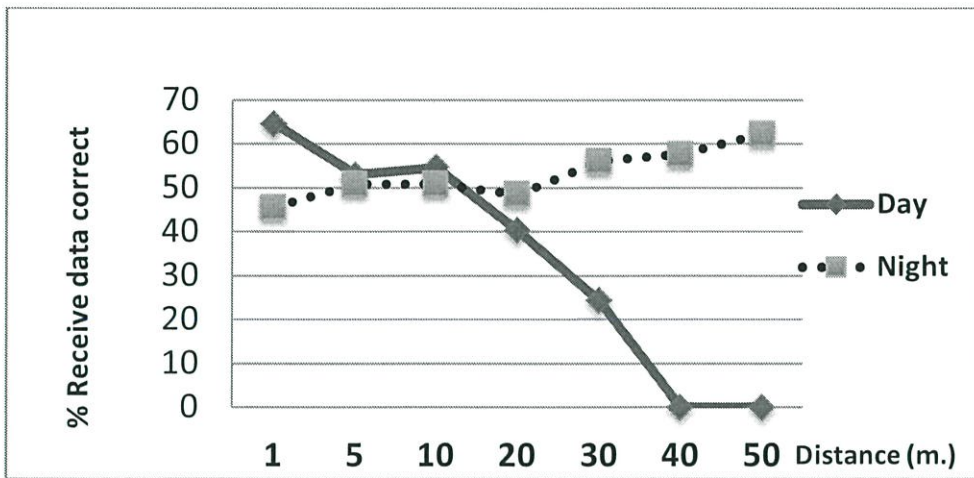
ตารางที่ 4.9 ผลการทดลองการตรวจจับที่ระยะต่างๆในเวลากลางวันโดยใช้เทคนิคการลดกรอบการประมวลผลภาพ

Distance (m.)	1	5	10	20	30	40	50
Number of times to receive data	65	68	66	66	65	66	66
Correct data	42	36	36	34	19	2	0
Cannot receive data	16	25	23	25	35	48	56
Error data	7	7	7	3	7	8	0
% receive correct data	64.62	52.94	54.55	51.52	29.23	3.03	0.00
% cannot receive data	24.62	36.76	34.85	37.88	53.85	72.73	84.85
% receive error data	10.77	10.29	10.61	4.55	10.77	12.12	0.00

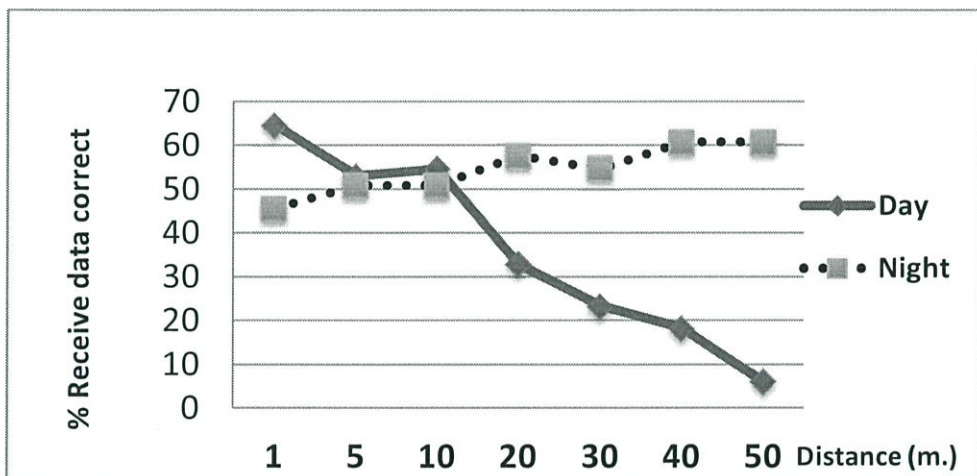
ตารางที่ 4.10 ผลการทดลองการตรวจจับที่ระยะต่างๆในเวลากลางคืนโดยใช้เทคนิคการลดกรอบการประมวลผลภาพ

Distance (m.)	1	5	10	20	30	40	50
Number of times to receive data	66	67	67	66	66	66	67
Correct data	30	34	34	37	36	39	38
Cannot receive data	27	25	28	25	22	25	22
Error data	9	8	5	4	8	2	7
% receive correct data	45.45	50.75	50.75	56.06	54.55	59.09	56.72
% cannot receive data	40.91	37.31	41.79	37.88	33.33	37.88	32.84
% receive error data	13.64	11.94	7.46	6.06	12.12	3.03	10.45

รูปที่ 4.27 – 4.28 แสดงกราฟความสัมพันธ์ระหว่างร้อยละการตรวจจับข้อมูลได้ถูกต้องกับระยะจากเครื่องรับในเวลากลางวันและกลางคืน โดยรูปที่ 4.27 จะตรวจจับธรรมดา แต่รูปที่ 4.28 จะใช้เทคนิคการลดกรอบการประมวลผล



รูปที่ 4.27 กราฟความสัมพันธ์ระหว่างร้อยละการรับข้อมูลได้ถูกต้องกับระยะจากเครื่องรับ  
ในเวลากลางวันและกลางคืน



รูปที่ 4.28 กราฟความสัมพันธ์ระหว่างร้อยละการรับข้อมูลได้ถูกต้องกับระยะจากเครื่องรับ  
ในเวลากลางวันและกลางคืนโดยใช้เทคนิคการลดการกระพริบผล

จากผลการทดลอง จะเห็นว่าเมื่อลดการกระพริบผลนั้น ในเวลากลางวันนั้นที่  
ระยะไกล จะสามารถตรวจจับข้อมูลได้มากขึ้น จากเดิมที่ไม่สามารถตรวจจับได้เลย

และเมื่อนำทั้งสองเทคนิคมารวมกัน ทั้งเทคนิคการชุมภาพ และการลดกรอบการประมวลผล ซึ่งได้ผลการทดลองในตอนกลางวันดังตารางที่ 4.11 และตอนกลางคืนดังตารางที่ 4.12

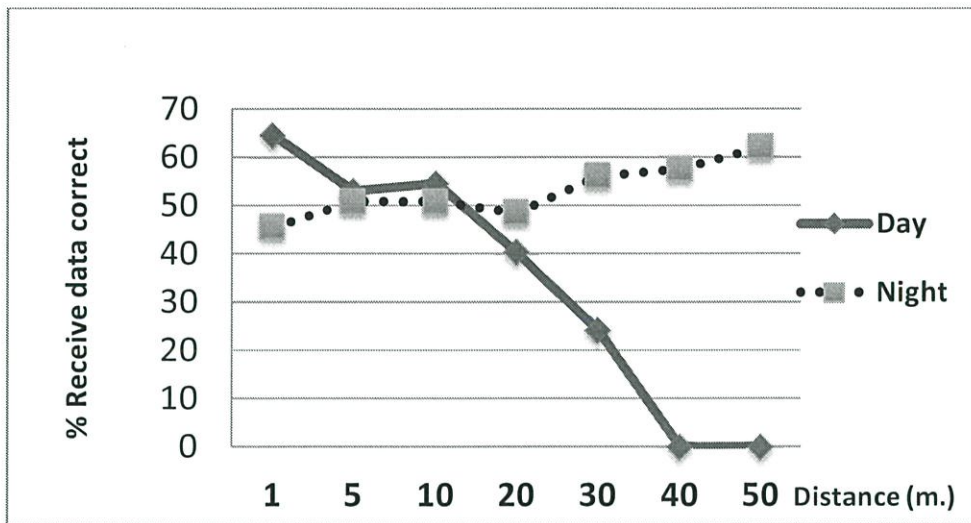
ตารางที่ 4.11 ผลการทดลองการตรวจจับที่ระยะต่างๆในเวลากลางวันโดยใช้เทคนิคการชุมภาพ และการลดกรอบการประมวลผลภาพ

Distance (m.)	1	5	10	20	30	40	50
Number of times to receive data	65	68	66	66	66	66	66
Correct data	42	36	36	31	27	18	10
Cannot receive data	16	25	23	31	27	18	10
Error data	7	7	7	9	10	8	8
% receive correct data	64.62	52.94	54.55	46.97	40.91	27.27	15.15
% cannot receive data	24.62	36.76	34.85	46.97	40.91	27.27	15.15
% receive error data	10.77	10.29	10.61	13.64	15.15	12.12	12.12

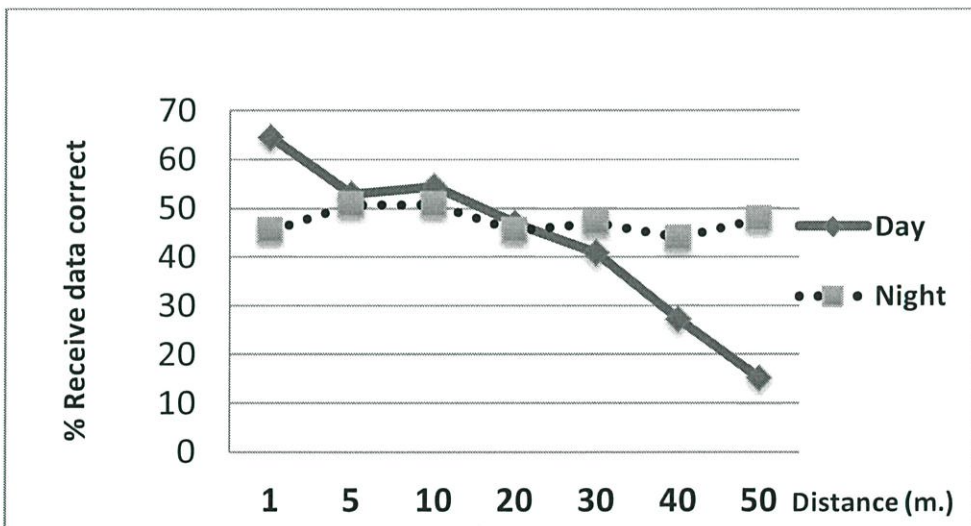
ตารางที่ 4.12 ผลการทดลองการตรวจจับที่ระยะต่างๆในเวลากลางคืนโดยใช้เทคนิคการชุมภาพ และการลดกรอบการประมวลผลภาพ

Distance (m.)	1	5	10	20	30	40	50
Number of times to receive data	66	67	67	66	66	66	67
Correct data	30	34	34	30	31	29	32
Cannot receive data	27	25	28	26	22	24	27
Error data	9	8	5	5	1	7	7
% receive correct data	45.45	50.75	50.75	45.45	46.97	43.94	47.76
% cannot receive data	40.91	37.31	41.79	39.39	33.33	36.36	40.30
% receive error data	13.64	11.94	7.46	7.58	1.52	10.61	10.45

รูปที่ 4.29 – 4.30 แสดงกราฟความสัมพันธ์ระหว่างร้อยละการตรวจจับข้อมูลได้ถูกต้องกับระยะจากเครื่องรับในเวลากลางวันและกลางคืน โดยรูปที่ 4.29 จะตรวจจับธรรมดา แต่รูปที่ 4.30 จะใช้เทคนิคการชุมภาพและการลดกรอบการประมวลผล



รูปที่ 4.29 กราฟความสัมพันธ์ระหว่างร้อยละการรับข้อมูลได้ถูกต้องกับระยะจากเครื่องรับในเวลากลางวันและกลางคืน



รูปที่ 4.30 กราฟความสัมพันธ์ระหว่างร้อยละการรับข้อมูลได้ถูกต้องกับระยะจากเครื่องรับในเวลากลางวันและกลางคืนโดยใช้เทคนิคการชູมภาพและการลดกรอบการประมวลผล

จากผลการทดลอง จะเห็นว่าการชູมภาพและการลดกรอบการประมวลผลนั้นทำให้การตรวจจับในตอนกลางวันในระยะไกลนั้น สามารถตรวจจับได้มากขึ้น จากเดิมที่ไม่สามารถจับได้เลย ดังนั้นการชູมภาพและการลดกรอบการประมวลผลนั้น เป็นวิธีที่สามารถทำให้เพิ่มประสิทธิภาพในการตรวจจับไฟกระพริบในตอนกลางวันได้

#### 4.4.5 การลดความผิดพลาดโดยการลดจำนวนบิตที่ส่ง

โดยปกติการสื่อสารนั้น หากมีจำนวนบิตที่ส่งข้อมูลมาก ย่อมทำให้ได้ข้อมูลจำนวนมากหรือมีความเร็วในการส่งมาก แต่ก็แลกมาด้วยความผิดพลาดที่มากขึ้นเช่นกัน หากบิตที่ส่งน้อยลง ความผิดพลาดก็น่าจะน้อยลงตามไปด้วย จึงได้ทำการทดลองลดบิตที่ใช้ส่งจาก 26 บิต เป็น 21 บิต และ 16 บิต ทำการทดลองตรวจจับในตอนกลางวันและกลางคืน และใช้เทคนิคการซูมภาพและการลดรอบการประมวลผลภาพ โดยได้ผลการทดลอง ส่ง 21 บิต ในตอนกลางวันดังตารางที่ 4.13 และในตอนกลางคืนดังตารางที่ 4.14

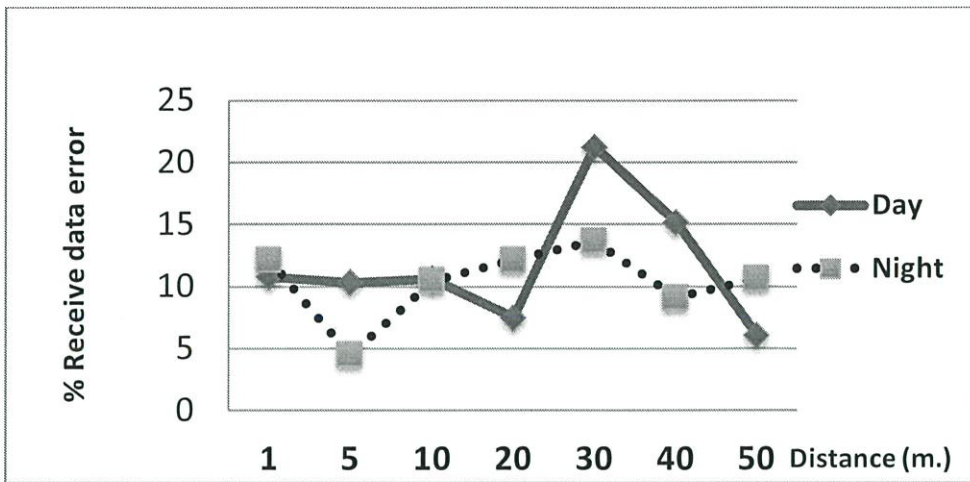
ตารางที่ 4.13 ผลการทดลองการตรวจจับที่ระยะต่างๆในเวลากลางวันโดยลดจำนวนบิตที่ส่งเป็น 21 บิต

Distance (m.)	1	5	10	20	30	40	50
Number of times to receive data	78	78	78	79	79	79	78
Correct data	41	50	45	45	44	25	7
Cannot receive data	30	33	29	31	29	48	61
Error data	7	5	4	3	6	6	10
% receive correct data	52.56	64.10	57.69	56.96	55.70	31.65	8.97
% cannot receive data	38.46	42.31	37.18	39.24	36.71	60.76	78.21
% receive error data	8.97	6.41	5.13	3.80	7.59	7.59	12.82

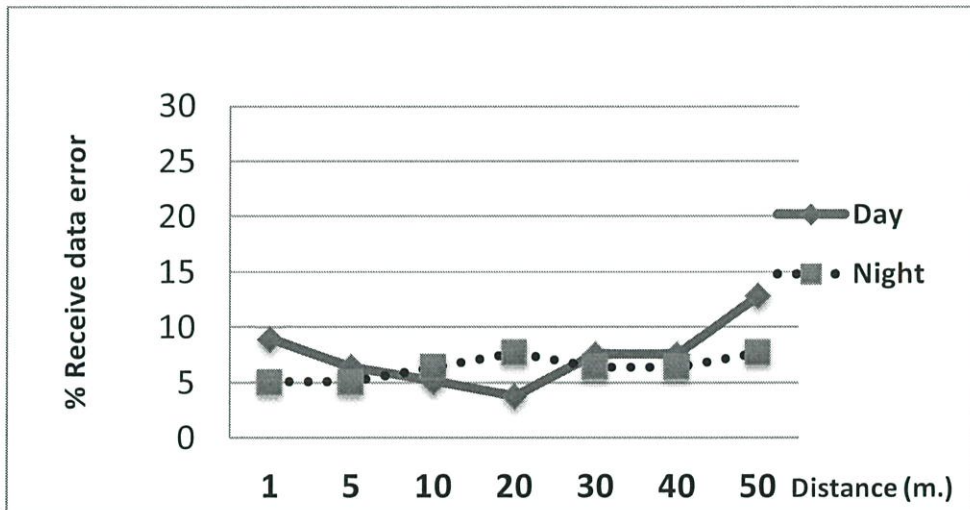
ตารางที่ 4.14 ผลการทดลองการตรวจจับที่ระยะต่างๆในเวลากลางคืน โดยลดจำนวนบิตที่ส่งเป็น 21 บิต

Distance (m.)	1	5	10	20	30	40	50
Number of times to receive data	79	79	78	78	78	78	78
Correct data	44	43	41	44	38	48	44
Cannot receive data	31	32	32	28	35	25	28
Error data	4	4	5	6	5	5	6
% receive correct data	55.70	54.43	52.56	56.41	48.72	61.54	56.41
% cannot receive data	39.24	40.51	41.03	35.90	44.87	32.05	35.90
% receive error data	5.06	5.06	6.41	7.69	6.41	6.41	7.69

รูปที่ 4.31 - 4.32 แสดงกราฟความสัมพันธ์ระหว่างร้อยละการตรวจจับข้อมูลผิดพลาดกับระยะจากเครื่องรับในเวลากลางวันและกลางคืน โดยรูปที่ 4.31 จะตรวจจับธรรมดาตัวส่ง 26 บิต แต่รูปที่ 4.32 จะตรวจจับโดยใช้ตัวส่ง 21 บิต



รูปที่ 4.31 กราฟความสัมพันธ์ระหว่างร้อยละการรับข้อมูลได้ผิดพลาดกับระยะจากเครื่องรับในเวลากลางวันและกลางคืน



รูปที่ 4.32 กราฟความสัมพันธ์ระหว่างร้อยละการรับข้อมูลได้ผิดพลาดกับระยะจากเครื่องรับในเวลากลางวันและกลางคืนโดยใช้จำนวนบิตที่ส่ง 21 บิต

และเมื่อทำการลดจำนวนบิตที่ใช้ส่งลงอีกเป็น 16 บิต จะได้ผลการทดลอง ส่ง 16 บิต ในตอนกลางวันดังตารางที่ 4.15 และ ในตอนกลางคืนดังตารางที่ 4.16

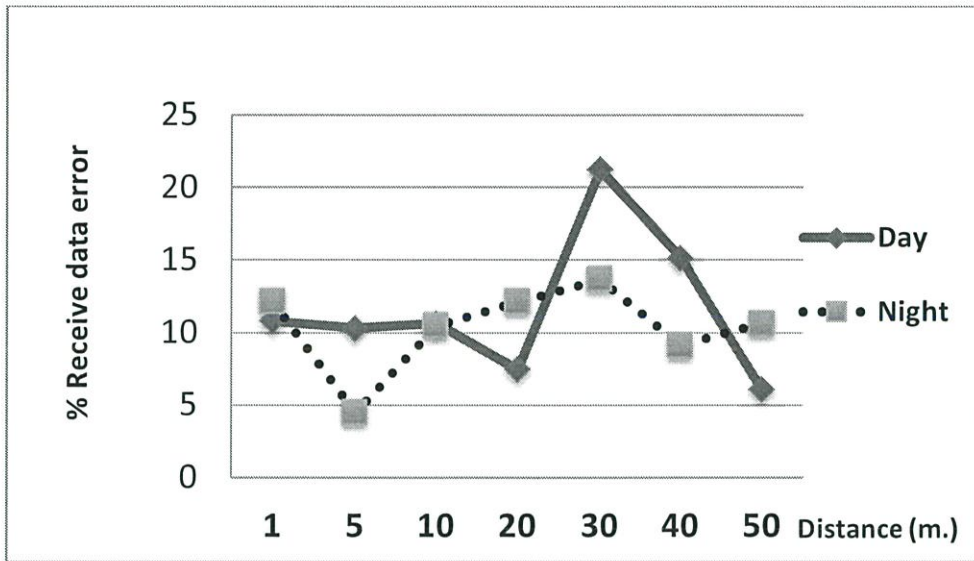
ตารางที่ 4.15 ผลการทดลองการตรวจจับที่ระยะต่างๆในเวลากลางคืนโดยลดจำนวนบิตที่ส่งเป็น 16 บิต

Distance (m.)	1	5	10	20	30	40	50
Number of times to receive data	101	100	101	101	101	102	102
Correct data	60	57	57	56	46	33	4
Cannot receive data	38	41	44	43	48	65	92
Error data	3	2	0	2	7	4	6
% receive correct data	59.41	57.00	56.44	55.45	45.54	32.35	3.92
% cannot receive data	37.62	41.00	43.56	42.57	47.52	63.73	90.20
% receive error data	2.97	2.00	0.00	1.98	6.93	3.92	5.88

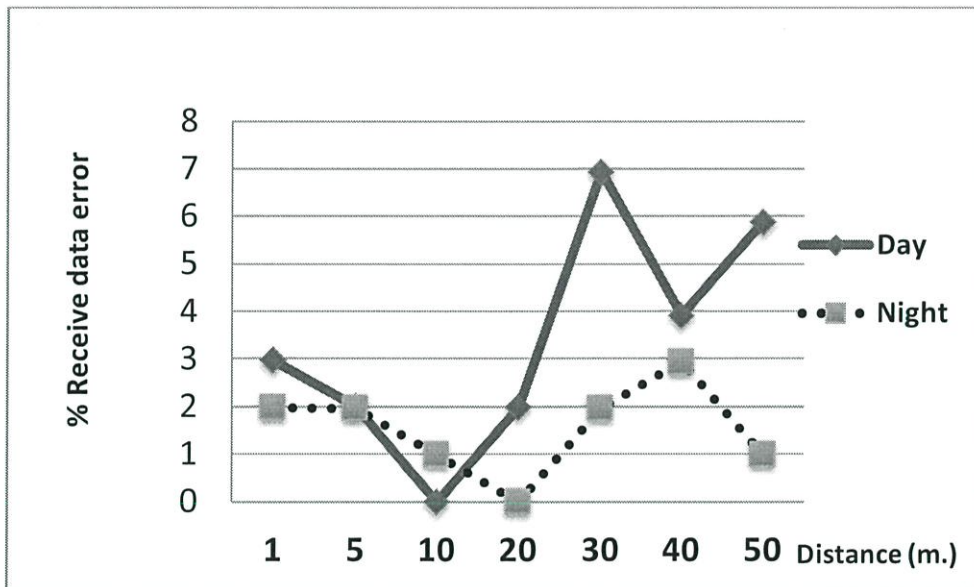
ตารางที่ 4.16 ผลการทดลองการตรวจจับที่ระยะต่างๆในเวลากลางคืนโดยลดจำนวนบิตที่ส่งเป็น 16 บิต

Distance (m.)	1	5	10	20	30	40	50
Number of times to receive data	101	102	101	102	102	102	101
Correct data	54	50	56	55	53	52	54
Cannot receive data	45	50	44	47	47	47	46
Error data	2	2	1	0	2	3	1
% receive correct data	53.47	49.02	55.45	53.92	51.96	50.98	53.47
% cannot receive data	44.55	49.02	43.56	46.08	46.08	46.08	45.54
% receive error data	1.98	1.96	0.99	0.00	1.96	2.94	0.99

รูปที่ 4.33 - 4.34 แสดงกราฟความสัมพันธ์ระหว่างร้อยละการตรวจจับข้อมูลผิดพลาดกับระยะจากเครื่องรับในเวลากลางวันและกลางคืน โดยรูปที่ 4.33 จะตรวจจับธรรมดาตัวส่ง 26 บิต แต่รูปที่ 4.34 จะตรวจจับโดยใช้ตัวส่ง 16 บิต



รูปที่ 4.33 กราฟความสัมพันธ์ระหว่างร้อยละการรับข้อมูลได้ผิดพลาดกับระยะจากเครื่องรับในเวลากลางวันและกลางคืน



รูปที่ 4.34 กราฟความสัมพันธ์ระหว่างร้อยละการรับข้อมูลได้ผิดพลาดกับระยะจากเครื่องรับในเวลากลางวันและกลางคืนโดยใช้จำนวนบิตที่ส่ง 16 บิต

จากผลการทดลองจะเห็นว่า เมื่อจำนวนบิตที่ต้องการส่งนั้น ยังมีค่าน้อยลงจำนวนครั้งที่สามารถรับข้อมูลได้ก็จะมากขึ้น และทำให้เปอร์เซ็นต์ความผิดพลาดนั้นน้อยลงตามไปด้วย จึงเป็นวิธีหนึ่งที่สามารถใช้ในการลดความผิดพลาดได้ แต่ก็จะทำให้ข้อมูลนั้นมีจำนวนน้อยลง เพราะบิตข้อมูลที่น้อยลงตามไปด้วย ทั้งนี้ขึ้นอยู่กับการใช้งาน

#### 4.4.6 การเพิ่มอัตราการจับได้โดยการเพิ่มขนาดของตัวส่ง

ในการทดลองนี้ จะทำการเพิ่มขนาดของตัวส่งจาก จาก LED 49 หลอด เป็น 98 หลอด ขนาด 7x14 ทดลองในเวลากลางวันและกลางคืน 16 บิต และใช้เทคนิคการซูมภาพและการลดการอบการประมวลผล ซึ่งมีผลการทดลองในตอนกลางวันดังตารางที่ 4.17 และตอนกลางคืนดังตารางที่ 4.18

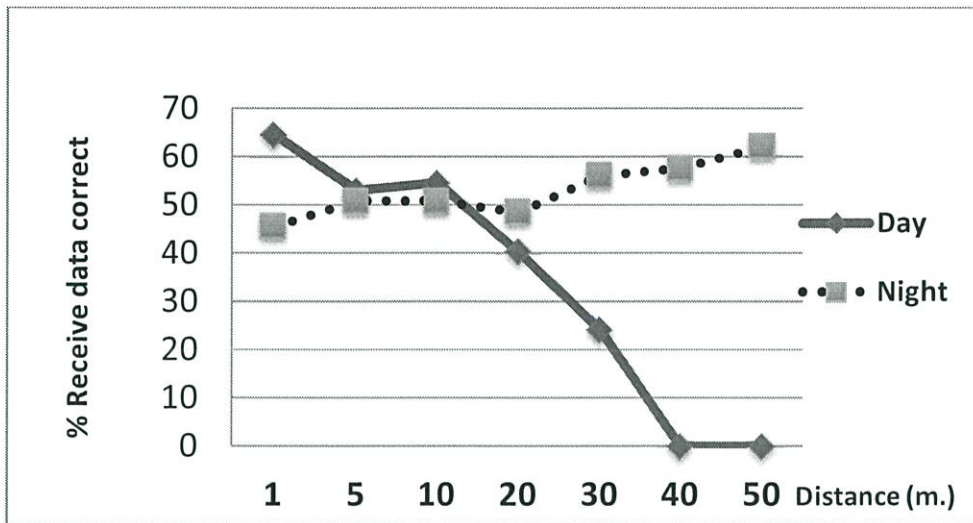
ตารางที่ 4.17 ผลการทดลองการตรวจจับที่ระยะต่างๆในเวลากลางวันโดยการเพิ่มขนาดของตัวส่ง

Distance (m.)	1	5	10	20	30	40	50
Number of times to receive data	101	100	100	100	100	101	101
Correct data	65	51	57	41	51	59	29
Cannot receive data	32	48	40	56	48	36	68
Error data	4	1	3	3	1	6	4
% receive correct data	64.36	51.00	57.00	41.00	51.00	58.42	28.71
% cannot receive data	31.68	48.00	40.00	56.00	48.00	35.64	67.33
% receive error data	3.96	1.00	3.00	3.00	1.00	5.94	3.96

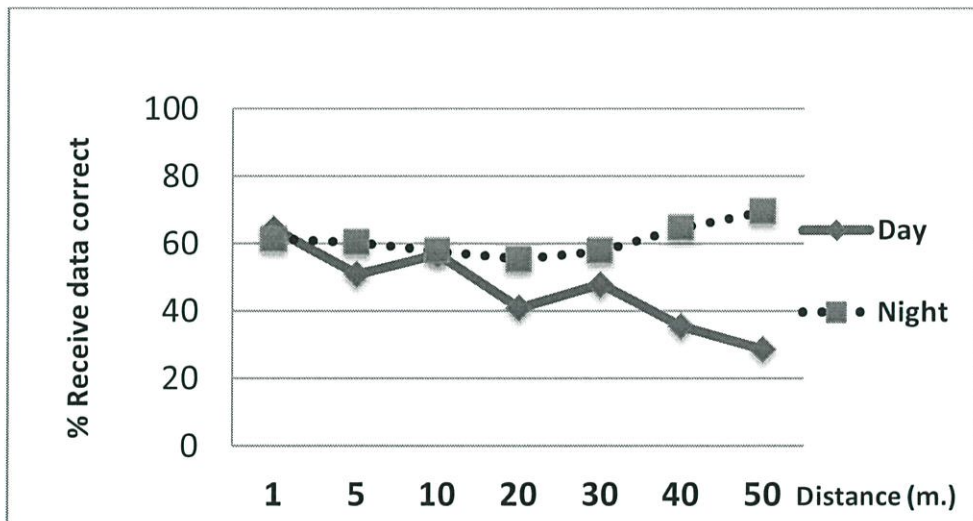
ตารางที่ 4.18 ผลการทดลองการตรวจจับที่ระยะต่างๆในเวลากลางคืนโดยการเพิ่มขนาดของตัวส่ง

Distance (m.)	1	5	10	20	30	40	50
Number of times to receive data	101	101	102	101	102	102	102
Correct data	62	61	59	56	59	66	71
Cannot receive data	34	39	40	44	41	34	30
Error data	5	1	3	1	2	2	1
% receive correct data	61.39	60.40	57.84	55.45	57.84	64.71	69.61
% cannot receive data	33.66	38.61	39.22	43.56	40.20	33.33	29.41
% receive error data	4.95	0.99	2.94	0.99	1.96	1.96	0.98

รูปที่ 4.35 – 4.36 แสดงกราฟความสัมพันธ์ระหว่างร้อยละการรับข้อมูลได้ถูกต้องกับระยะจากเครื่องรับในเวลากลางวันและกลางคืน ตัวส่ง 16 บิต โดยรูปที่ 4.35 จะตรวจจับไฟกระพริบ LED ขนาด 7x7 แต่รูปที่ 4.36 จะตรวจจับโดยใช้ตัวขนาด 7x14



รูปที่ 4.35 กราฟความสัมพันธ์ระหว่างร้อยละการรับข้อมูลได้ถูกต้องกับระยะจากเครื่องรับ ในเวลากลางวันและกลางคืน



รูปที่ 4.36 กราฟความสัมพันธ์ระหว่างร้อยละการรับข้อมูลได้ถูกต้องกับระยะจากเครื่องรับ ในเวลากลางวันและกลางคืนโดยการเพิ่มขนาดของตัวส่ง

จากผลการทดลอง จะเห็นว่าเมื่อเพิ่มขนาดของตัวส่งนั้น ย่อมให้ความสว่างที่มากขึ้นใน การอบการประมวลผลภาพนั้นๆตามไปด้วย ทำให้สามารถตรวจจับได้มากขึ้นจากเดิมที่ใช้ตัวส่งขนาด 7x7 หรือ 49 หลอดในตอนกลางวันในระยะไกล (40m - 50m) การเพิ่มขนาดของตัวส่ง เป็นวิธีที่ สามารถเพิ่มอัตราการตรวจจับได้ เช่นกัน

#### 4.4.7 การเพิ่มอัตราการตรวจจับโดยทดลองกับแอลอีดีสีต่างๆ



รูปที่ 4.37 LED สีต่างๆที่ใช้ในการทดลอง

การทดลองเมื่อเปลี่ยนหลอด LED เป็นสีต่างๆ ได้แก่ สีแดง สีม่วง สีขาว โดยทดลองตรวจจับในเวลากลางวันและกลางคืน 16 บิต และใช้เทคนิคการซุ่มและการลดกรอบการประมวลผล

โดยมีผลการทดลองสีแดงตอนกลางวันดังตารางที่ 4.19 ตอนกลางคืนดังตารางที่ 4.20 และ สีม่วงตอนกลางวันดังตารางที่ 4.21 และตอนกลางคืนดังตารางที่ 4.22 และสีขาว Super Bright ตอนกลางวันดังตารางที่ 4.23 และตอนกลางคืนดังตารางที่ 4.24

ตารางที่ 4.19 ผลการทดลองการตรวจจับที่ระยะต่างๆในเวลากลางวันโดยใช้ LED สีแดง

Distance (m.)	1	5	10	20	30	40	50
Number of times to receive data	100	101	101	102	102	100	101
Correct data	63	69	68	23	13	0	0
Cannot receive data	36	32	33	77	88	100	101
Error data	1	0	0	2	1	0	0
% receive correct data	63.00	68.32	67.33	22.55	12.75	0.00	0.00
% cannot receive data	36.00	31.68	32.67	75.49	86.27	100.00	100.00
% receive error data	1.00	0.00	0.00	1.96	0.98	0.00	0.00

ตารางที่ 4.20 ผลการทดลองการตรวจจับที่ระยะต่างๆในเวลากลางคืนโดยใช้ LED สีแดง

Distance (m.)	1	5	10	20	30	40	50
Number of times to receive data	101	102	102	100	101	101	102
Correct data	59	58	60	63	63	54	54
Cannot receive data	39	40	41	30	35	42	44
Error data	3	4	1	7	3	5	4
% receive correct data	58.42	56.86	58.82	63.00	62.38	53.47	52.94
% cannot receive data	38.61	39.22	40.20	30.00	34.65	41.58	43.14
% receive error data	2.97	3.92	0.98	7.00	2.97	4.95	3.92

ตารางที่ 4.21 ผลการทดลองการตรวจจับที่ระยะต่างๆในเวลากลางวันโดยใช้ LED สีม่วง

Distance (m.)	1	5	10	20	30	40	50
Number of times to receive data	101	100	101	101	101	100	101
Correct data	65	67	68	13	0	0	0
Cannot receive data	34	32	31	86	100	100	101
Error data	2	1	2	2	1	0	0
% receive correct data	64.36	67.00	67.33	12.87	0.00	0.00	0.00
% cannot receive data	33.66	32.00	30.69	85.15	99.01	100.00	100.00
% receive error data	1.98	1.00	1.98	1.98	0.99	0.00	0.00

ตารางที่ 4.22 ผลการทดลองการตรวจจับที่ระยะต่างๆในเวลากลางคืนโดยใช้ LED สีม่วง

Distance (m.)	1	5	10	20	30	40	50
Number of times to receive data	102	101	101	102	101	101	101
Correct data	63	59	59	67	64	64	62
Cannot receive data	38	41	37	29	33	33	38
Error data	1	1	5	6	4	4	1
% receive correct data	61.76	58.42	58.42	65.69	63.37	63.37	61.39
% cannot receive data	37.25	40.59	36.63	28.43	32.67	32.67	37.62
% receive error data	0.98	0.99	4.95	5.88	3.96	3.96	0.99

ตารางที่ 4.23 ผลการทดลองการตรวจจับที่ระยะต่างๆในเวลากลางวันโดยใช้ LED สีขาว Super Bright

Distance (m.)	1	5	10	20	30	40	50
Number of times to receive data	100	101	100	100	101	100	100
Correct data	56	52	50	59	51	0	0
Cannot receive data	42	48	48	39	49	100	100
Error data	2	1	2	2	1	0	0
% receive correct data	56.00	51.49	50.00	59.00	50.50	0.00	0.00
% cannot receive data	42.00	47.52	48.00	39.00	48.51	100.00	100.00
% receive error data	2.00	0.99	2.00	2.00	0.99	0.00	0.00

ตารางที่ 4.24 ผลการทดลองการตรวจจับที่ระยะต่างๆในเวลากลางคืนโดยใช้ LED สีขาว Super Bright

Distance (m.)	1	5	10	20	30	40	50
Number of times to receive data	101	101	99	101	100	100	101
Correct data	55	51	51	62	62	64	58
Cannot receive data	43	45	43	38	35	35	41
Error data	3	5	5	1	3	1	2
% receive correct data	54.46	50.50	51.52	61.39	62.00	64.00	57.43
% cannot receive data	42.57	44.55	43.43	37.62	35.00	35.00	40.59
% receive error data	2.97	4.95	5.05	0.99	3.00	1.00	1.98

จากการทดลอง จะเห็นว่าในตอนกลางคืนทุกสีนั้น มีประสิทธิภาพในการตรวจจับที่ไม่แตกต่างกันมาก แต่ในตอนกลางวันจะเห็นว่าหลอดสีเขียวที่ใช้ในการทดลองปกติ และ Super Bright สีขาวนั้นมีประสิทธิภาพในการตรวจจับดีกว่าสีแดงและสีม่วงตามลำดับ

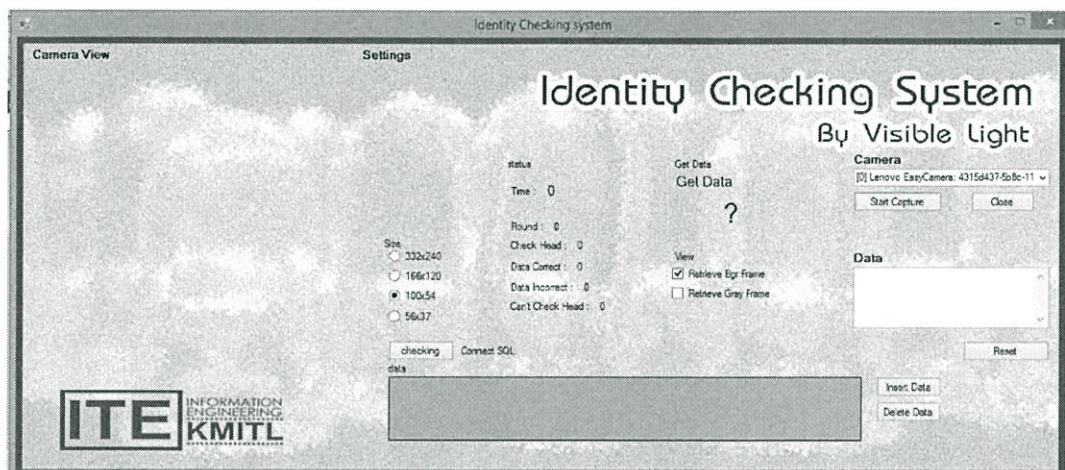
#### 4.4.8 การลดความผิดพลาดโดยใช้การตรวจสอบจากการตรวจจับสองครั้ง

โดยปกติในโปรแกรมการตรวจจับนั้น เมื่อโปรแกรมหา Header bit และ Close bit เ็จ จะทำการแสดง Data bit ขึ้นมาโดยแปลงเป็นเลขฐาน 10 ก่อนนำไปแสดงผล แล้วตรวจสอบกับ ฐานข้อมูลที่หน้าจอ Interface ซึ่งบิตข้อมูลที่ผิดหรือถูกนั้นก็จะถูกแสดงขึ้นมา ซึ่งหากนำไปใช้งานจริงก็จะทำให้เกิดความผิดพลาดได้ จึงได้คิดอัลกอริทึมง่ายๆที่ใช้ในการลดความผิดพลาดนี้ โดยการ ตรวจจับสองครั้งก่อนแสดงผล โดยเมื่อโปรแกรมตรวจจับได้เป็นเลขฐานสิบมาแล้วจะทำการเก็บค่า ไว้ที่ตัวแปรตัวหนึ่งก่อน แล้วพอร์บมาอีกครั้งต่อไป ก็จะนำมาเก็บไว้ที่ตัวแปรอีกตัวหนึ่ง แล้วนำมา เปรียบเทียบว่าเป็นตัวเลขเดียวกันหรือไม่ หากเป็นตัวเลขเดียวกันจึงจะแสดงผลขึ้นมา หากไม่ เหมือนกันก็จะไม่แสดงผลและทำการรับข้อมูลมาใหม่อีกครั้ง ส่วนเรื่องความผิดพลาดที่เกิดขึ้นนั้น เป็นไปได้้น้อยมากเนื่องจากบิตข้อมูลที่ผิดพลาดนั้น เป็นไปได้ยากที่บิตจะผิดที่ตำแหน่งเดียวกันทั้งสอง ครั้ง จึงทำให้มีข้อดีคือข้อมูลที่แสดงผลนั้นมีความถูกต้องแน่นอน แต่ข้อเสียคือทำให้อัตราการ แสดงผลนั้นน้อยลง หากตรวจจับในตอนกลางวันที่ระยะไกล ที่มีอัตราการตรวจจับต่ำนั้น จะทำให้ แสดงผลได้ช้า

ซึ่งผลการทดลองพบว่า ความผิดพลาดในการแสดงผลหลังจากใช้วิธีการตรวจจับสองครั้ง ไม่มีความผิดพลาดแสดงผลเลยแม้แต่ครั้งเดียว ตั้งแต่ทดลองมาในทุกแบบ แต่อัตราการตรวจจับนั้น จะต่ำลง และแสดงผลช้ากว่าเดิม เนื่องจากต้องรอการตรวจจับถึงสองครั้งแล้วนำมาตรวจสอบก่อน

## 4.5 การจัดการฐานข้อมูลของระบบ

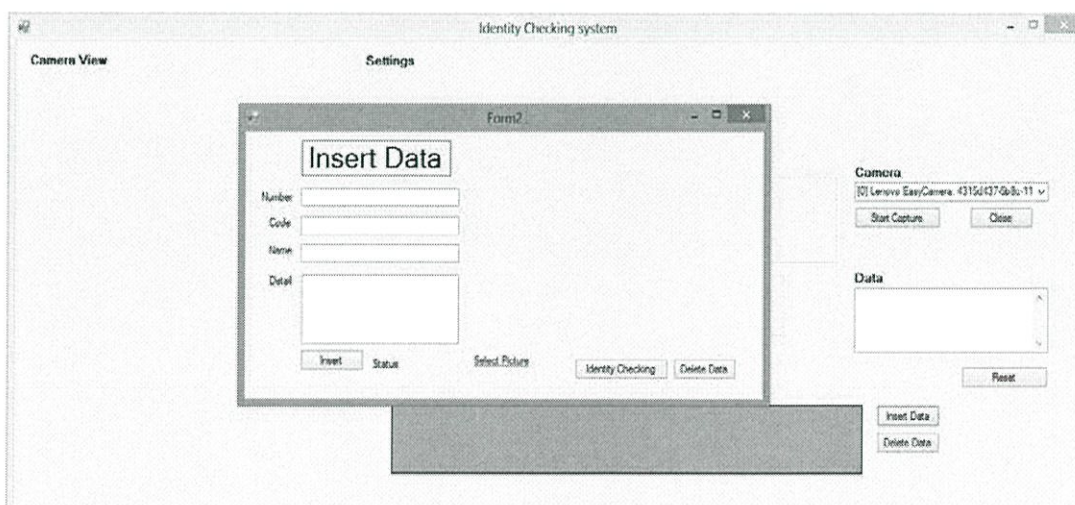
ในระบบตรวจสอบอัตลักษณ์ด้วยแสงนั้น ฐานข้อมูลถือเป็นส่วนสำคัญของระบบ เนื่องจากการใช้ในการเก็บข้อมูลอัตลักษณ์ต่างๆที่ต้องการจะตรวจสอบ เมื่อโปรแกรมตรวจสอบการกระพริบของแสง จะได้โค้ดเป็นตัวเลขฐานสิบมา แล้วจึงนำโค้ดที่ได้นี้ไปเทียบในฐานข้อมูลว่าเป็นโค้ดของอัตลักษณ์ใดๆ ดังนั้นก่อนที่จะมีตรวจสอบได้นั้น ต้องมีการเพิ่มข้อมูลอัตลักษณ์และโค้ดของสิ่งนั้นๆ ลงในฐานข้อมูลก่อน และเพื่ออำนวยความสะดวกจึง มีส่วนของโปรแกรมที่สามารถเพิ่มข้อมูลตรวจสอบข้อมูล อัปเดตข้อมูลและลบข้อมูลในฐานข้อมูลได้ โดยมีอินเตอร์เฟซผู้ใช้งานดังรูปที่ 4.38



รูปที่ 4.38 GUI ส่วนของฐานข้อมูลและการจัดการฐานข้อมูลของโปรแกรมตรวจสอบอัตลักษณ์โดยใช้แสง

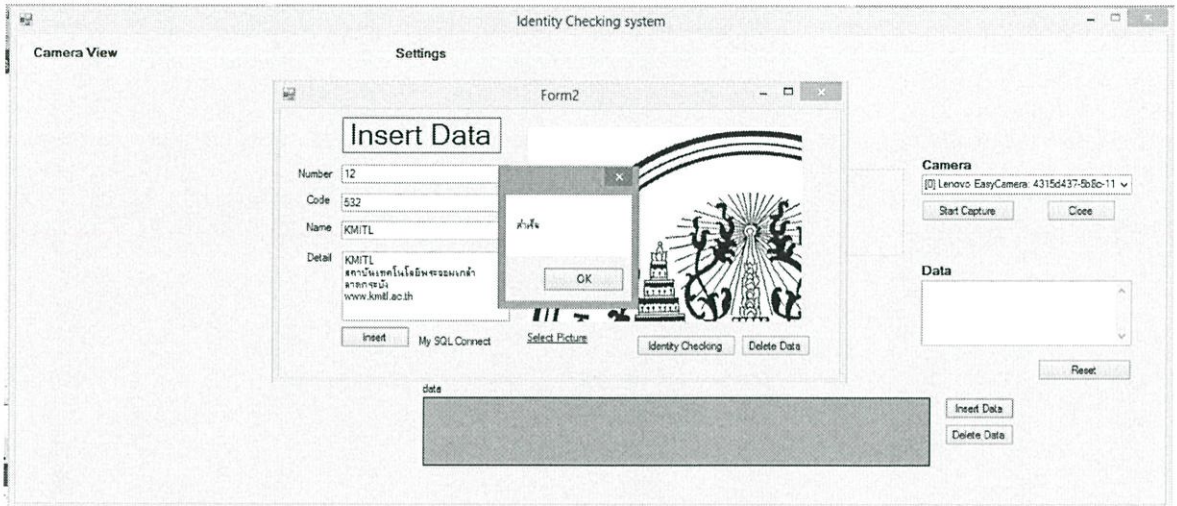
### 4.5.1 การเพิ่มข้อมูลลงในฐานข้อมูล

เมื่อคลิกปุ่ม Insert Data จะแสดงหน้าต่างสำหรับการเพิ่มข้อมูลลงในฐานข้อมูล



รูปที่ 4.39 GUI ส่วนการเพิ่มข้อมูล

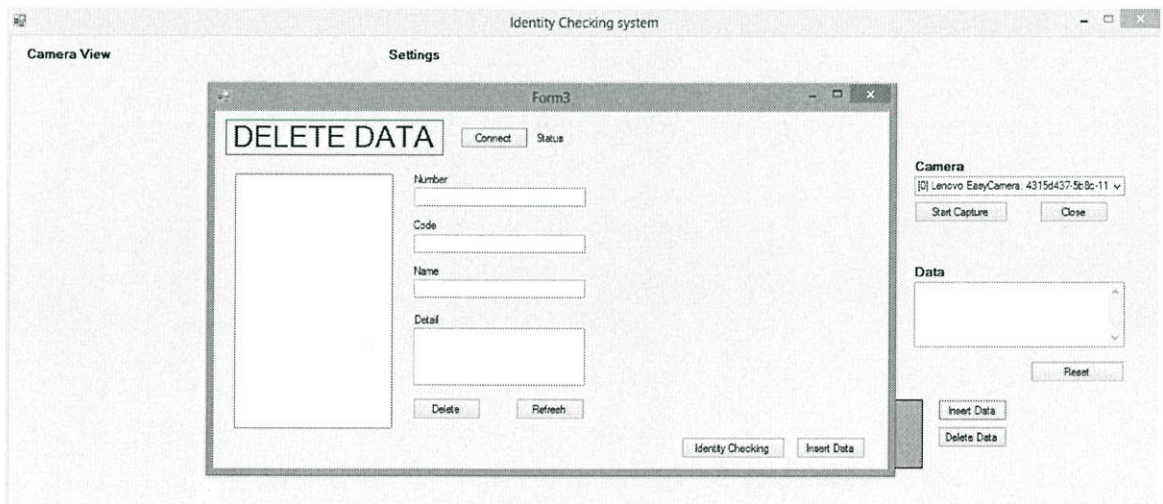
จากนั้น กรอกรายละเอียดสำหรับอัตลักษณ์ของสิ่งนั้นๆลงไป Number คือ หมายเลข, Code คือ รูปแบบไฟกระพริบที่แทนอัตลักษณ์ของสิ่งต่างๆเป็นเลขฐานสิบและไม่ซ้ำกัน, Name คือ ชื่อของสิ่งนั้นๆ, Detail คือรายละเอียดของอัตลักษณ์ และเลือกรูปภาพและเมื่อกรอกรายละเอียดต่างๆเรียบร้อยแล้วกดปุ่ม Insert ก็จะมีกล่องข้อความแสดงว่าการเพิ่มข้อมูลอัตลักษณ์ลงในฐานข้อมูลนั้นเสร็จเรียบร้อยแล้ว



รูปที่ 4.40 GUI ส่วนการเพิ่มข้อมูลเมื่อเพิ่มข้อมูลสำเร็จ

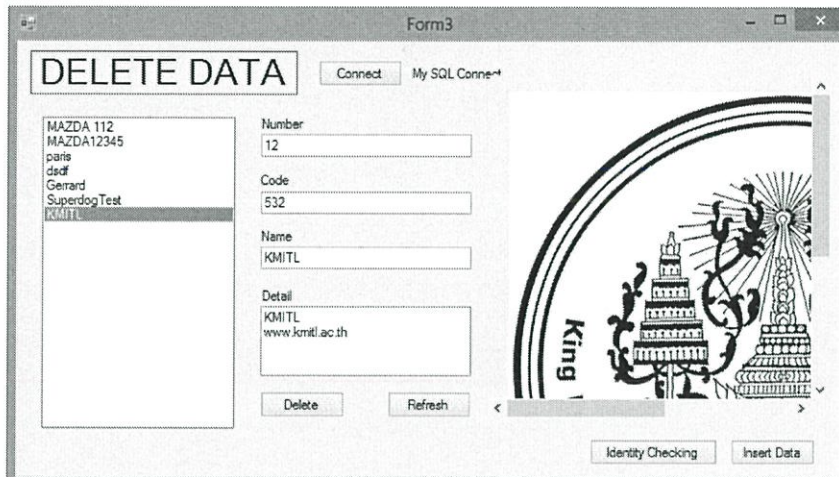
#### 4.5.2 การลบข้อมูลในฐานข้อมูล

จากโปรแกรมนั้นหากต้องการลบข้อมูลออกจากฐานข้อมูลนั้น สามารถทำได้โดยคลิกปุ่ม “Delete Data” จะแสดงหน้าต่างสำหรับลบข้อมูลขึ้นมา ดังนี้



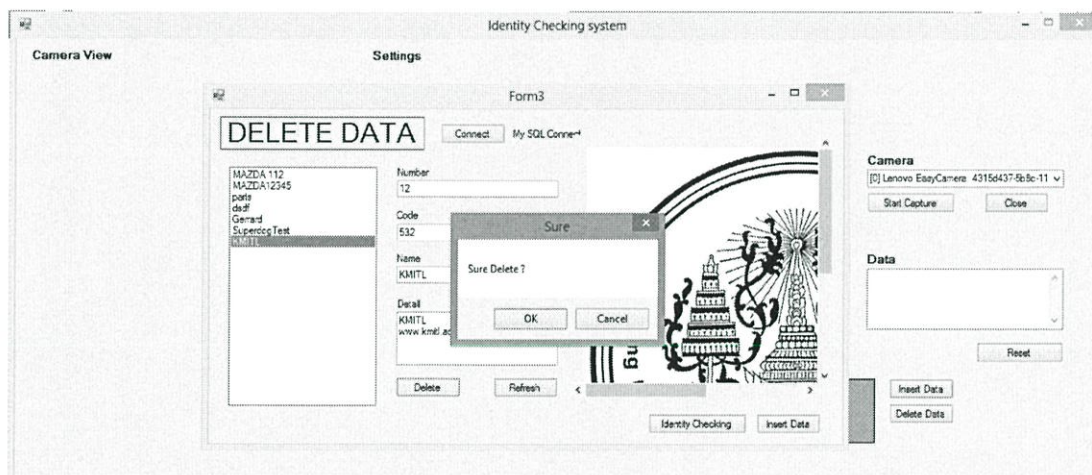
รูปที่ 4.41 GUI ส่วนการลบข้อมูล

กดปุ่ม Connect เพื่อเชื่อมต่อฐานข้อมูลและจะแสดงชื่อข้อมูลที่มีอยู่ในฐานข้อมูลออกมา และเมื่อนำเมาส์ไปคลิกเลือกชื่อนั้นๆก็จะแสดงรายละเอียดต่างๆออกมา ดังภาพ



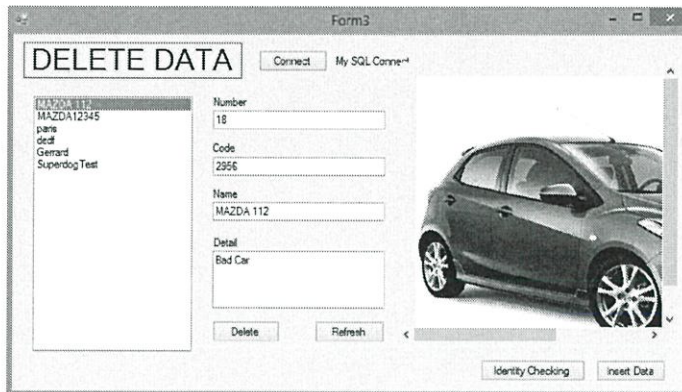
รูปที่ 4.42 GUI ส่วนการลบข้อมูลโดยเลือกข้อมูลที่ต้องการจะลบ

เมื่อเลือกข้อมูลอัตลักษณ์ที่จะทำการลบแล้ว กดปุ่ม Delete จะมีหน้าต่างยืนยันการทำการลบข้อมูล



รูปที่ 4.43 GUI ยืนยันการลบข้อมูล

จากนั้นข้อมูลจะถูกลบทิ้งออกจากฐานข้อมูล ซึ่งสามารถตรวจสอบได้โดยกดปุ่ม Refresh จะพบว่าข้อมูล KMITL นั้นไม่ได้อยู่ในฐานข้อมูลแล้ว



รูปที่ 4.44 GUI การลบข้อมูลเรียบร้อยแล้ว

#### 4.6 การนำไปประยุกต์ใช้

ระบบตรวจสอบอัตลักษณ์ด้วยแสงนั้น สามารถนำไปประยุกต์ใช้ได้หลากหลาย ที่ใช้ทดแทนข้อจำกัดบางอย่างของ QRCode และ RFID สามารถนำไปประยุกต์ใช้ได้หลากหลายงาน เช่น

- การนำไปประยุกต์ใช้กับไฟฉาย เพื่อตรวจสอบอัตลักษณ์ของผู้ใช้ไฟฉาย
- การนำไปประยุกต์ใช้กับกล้องโทรศัพท์มือถือ
- การนำไปประยุกต์ใช้กับงานโฆษณา และในห้างสรรพสินค้าโดยใช้โทรศัพท์มือถือ
- การนำไปประยุกต์ใช้กับงานการบิน ในการส่งสัญญาณการลงจอดของเครื่องบินในขณะที่สภาพอากาศเลวร้าย
- การนำไปประยุกต์ใช้กับระบบตรวจสอบอัตลักษณ์ของพนักงานที่ต้องสวมเครื่องแต่งกายมิดชิดหรือในงานตอนกลางคืน เพื่อตรวจสอบอัตลักษณ์ของพนักงาน
- ฯลฯ

## บทที่ 5

# สรุปและวิจารณ์ผลการทดลอง

### 5.1 บทสรุปโครงการ

ในการทดลองการทำงานของระบบจากการทดลองในบทที่ 4 นั้น ได้ทำการทดลองโดยใช้กล้องเว็บแคมตรวจจับการกระพริบของไดโอดเปล่งแสงโดยใช้บอร์ด Arduino UNO ในการโปรแกรมการกระพริบของแสง ซึ่งกล้องสามารถตรวจจับการกระพริบของแสงได้โดยใช้โปรแกรมที่เขียนขึ้นมาโดยใช้ภาษา C# โดยใช้ไลบรารีของ EmguCV, AForge.NET มาใช้ในการประมวลผลภาพ และนำข้อมูลที่รับได้ไปตรวจสอบกับฐานข้อมูลว่ามีข้อมูลอยู่หรือข้อมูลตรงกันหรือไม่ จากผลการทดลองเมื่อรันโปรแกรมให้กล้องตรวจจับการกระพริบของแสงพบว่าสามารถรับข้อมูลได้ และสามารถนำไปตรวจสอบกับฐานข้อมูลได้ แต่ความสามารถในการรับข้อมูลจะเริ่มลดลงเมื่อระยะทางเริ่มไกลขึ้นเรื่อยๆ และในที่มีจะสามารถตรวจจับการกระพริบของแสง สามารถตรวจจับการกระพริบรับข้อมูลได้ดีกว่าในตอนกลางวัน หรือบริเวณที่มีแสงจากสภาพแวดล้อมมาก และสามารถใช่วิธีต่างๆในการลดความผิดพลาดคือ การเข้ารหัสโดยใช้ Block Code, Convolutional Code, การลดจำนวนบิตที่ใช้ส่ง และการตรวจจับที่ละสองครั้ง และวิธีต่างๆในการเพิ่มอัตราการตรวจจับข้อมูลได้ ทำให้ตรวจจับในตอนกลางวันในระยะไกลได้โดยใช้ เทคนิคการซูมภาพ การลดกรอบการประมวลผลภาพ การเพิ่มขนาดของตัวส่ง และการทดลองกับหลอดแอลอีดีสีต่างๆ ซึ่งสามารถใช้เป็นต้นแบบของระบบต่างๆที่สามารถนำไปประยุกต์ใช้ได้อย่างหลากหลายในอนาคต

### 5.2 ปัญหาที่พบในระหว่างการดำเนินโครงการ

1. แหล่งข้อมูลเพื่อศึกษาหลักการประมวลผลภาพ หลักการใช้งานไลบรารีของ OpenCV หรืออัลกอริทึมในการประมวลผลภาพนั้นมีน้อยมาก ทำให้เป็นอุปสรรคต่อการพัฒนา
2. หากมีภาคส่งพร้อมกันหลายๆตัวจะทำให้การตรวจจับนั้นมีความคลาดเคลื่อนได้
3. สภาพแวดล้อมที่แสงมีการเปลี่ยนแปลงมากๆหรือที่มีแสงสว่างมากๆทำให้ค่าความสว่างภาพมีความคลาดเคลื่อนได้ ทำให้รับข้อมูลได้ผิดพลาด
4. ข้อมูลบางค่าจะมีความซับซ้อนกันของ Header bit และ Data bit ทำให้ได้รับค่าที่ผิดพลาดตลอดเวลาได้
5. ข้อมูลที่รับได้ในบางครั้งยังมีความผิดพลาดอยู่การเลื่อนบิตหรือตรวจสอบความถูกต้องยังไม่มีประสิทธิภาพ ทำให้รับข้อมูลไม่ได้หรือรับมาแล้วมีความผิดพลาดสูง

### 5.3 แนวทางการแก้ไข

1. พยายามศึกษาเกี่ยวกับหลักการประมวลผลภาพ การใช้งานไลบรารีของ OpenCV จากสื่อหรือหนังสือต่างประเทศให้มากขึ้น
2. ค้นคว้าหาข้อมูลเกี่ยวกับการแยกการประมวลผล หาข้อมูลเพื่อที่จะพัฒนาต่อไป

3. ค้นคว้าหาข้อมูลเพิ่มเติมเกี่ยวกับการเขียนโปรแกรมการประมวลผลภาพเพื่อลดสัญญาณรบกวนหรือลดปัญหาปัจจัยภายนอกทางด้านสิ่งแวดล้อม

4. ศึกษาเรื่องการตรวจสอบความถูกต้องของบิตข้อมูลให้มากขึ้น เพื่อลดความผิดพลาดของบิตข้อมูล

#### 5.4 การพัฒนาต่อและการนำไปใช้

1. นำไปพัฒนาต่อเพื่อใช้งานกับระบบต่างๆได้เช่นงานรักษาความปลอดภัย งานตรวจสอบอัตลักษณ์ในเวลากลางคืน หรืองานโฆษณา เป็นต้น

2. นำไปพัฒนากับระบบติดตาม (Tracking) ไฟกระพริบได้

3. ลดการรับข้อมูลได้ผิดพลาดให้น้อยลงและเพิ่มอัตราที่ตรวจจับได้ให้มากขึ้นกว่าเดิม

4. ใช้กล้องโทรศัพท์มือถือในการตรวจจับแทนกล้องเว็บแคมได้

## บรรณานุกรม

- [1] Don Pearson , Image Processing, McGRAW-HILL , 1991
- [2] Milan Sonka and Vaclav Hlavac and Roger Boyle, Image Processing, Analysis and Machine Vision 3<sup>rd</sup> Edition,Thompson , 2008
- [3] รศ.ดร.สมเกียรติ อุดมทรรษกุล, การประมวลผลภาพดิจิทัลเบื้องต้น, สำนักพิมพ์ท็อปพับลิชชิ่ง , 2011
- [4] OpenCV, <http://opencv.org>
- [5] EmguCV, <http://www.emgu.com>
- [6] Aforge.Net, <http://www.aforogenet.com>
- [7] Arduino, <http://arduino.cc/en/Main/arduinoBoardUno>
- [8] นภชา รรวัฒน์ และสิริฉัตร เกตุเรืองโรจน์, การติดตามใบหน้าและลักษณะเด่นบนใบหน้าและการสร้างศีรษะจากลำดับภาพสองมิติ, คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย , 2010
- [9] แสงและความส่องสว่าง, [http:// th.wikipedia.org/wiki/แสง](http://th.wikipedia.org/wiki/แสง)  
<http://www.tieathai.org/know/general/general0.htm>
- [10] Rafael C. Gonzalez and Richard E. Woods , Digital Image Processing 2<sup>nd</sup> edition, Prentice Hall, 2002
- [11] Shinichiro Haruyama, Visible Light Communications : Recent Activities in Japan, Keio University, 2011
- [12] สเปกตรัมแสง, <http://www.rmutphysics.com/physics>
- [13] หลอดไฟแบบต่างๆ, <http://kwaithai.wordpress.com>
- [14] หลอดฮาโลเจน, <http://www.yellowathome.com>
- [15] หลอดแอลอีดี, <http://creativentechno.wordpress.com>
- [16] หลอดฟลูออเรสเซนต์, <http://www.thaitechno.net>
- [17] การเข้ารหัสช่องสัญญาณ, <http://www.kmitl.ac.th/dslabs/Viterbi/convolution.html>
- [18] ระบบสี RGB, <http://webriti.com/cmyk-and-rgb-colorspace/>
- [19] ระบบสี YUV, <http://www.baslerweb.com/>
- [20] ระบบสี YIQ, <http://dcssrv1.oit.uci.edu/~wiedeman/cspace/me/infoyiqr.html>
- [21] ระบบสี YCbCr, <http://software.intel.com>
- [22] ภาพบิตแมพและภาพเวกเตอร์, <http://www.functionx.com>

ภาคผนวก

ภาคผนวก ก.  
โปรแกรมระบบตรวจสอบอัตลักษณ์ด้วยแสงในส่วนภาคส่ง

## โปรแกรมระบบตรวจสอบอัตลักษณ์ด้วยแสงในส่วนของภาคส่ง

```
int c, k;int n = 89;int arr[8];int i_arr=0;
int i=0;int j=0;int size_arr;int data[15];
int data_encode[16];int temp = 0;
int buttonPin = 2;
int ledPin = 10;
boolean buttonState;
boolean lastState;
boolean state = HIGH;
void setup()
{
  pinMode (buttonPin,INPUT);
  pinMode(ledPin, OUTPUT);
  digitalWrite(ledPin ,LOW) ;

  for (c = 7; c >= 0; c--) // 0-7 have 8 digit
  {
    k = n >> c;
    if (k & 1)
      {
        arr[i_arr] = 1;
        printf("1");
      }
    else
      {
        arr[i_arr] = 0;
        printf("0");
      }
    i_arr++;
  }
  size_arr=sizeof(arr)/sizeof(arr[0]);
  for(i=0;i<15;i++){
    if( (i>=0) && (i<=4) ){
      data[i]=1;
    }else if((i>=5)&&(i<=12)){
      data[i] = arr[j];
      j++;
    }else if((i>=13)&&(i<=14)){
```

```

        data[i] = 0;
    }
}
for(j=0;j<=14;j++){
    if(temp!=data[j]){
        data_encode[j]=1;
    }else data_encode[j]=0;
    temp = data_encode[j];
}
}
void loop()
{
    buttonState = digitalRead(buttonPin);

    if ( buttonState == LOW && lastState == HIGH )
    {
        delay (10);
        if (digitalRead(buttonPin) == LOW)
        {
            state = !state;
        }
    }
    if(state == LOW)
    {
        digitalWrite(ledPin,data_encode[i]);
        delay(100);
        i=(i+1)%16;
    }else {
        digitalWrite(ledPin,LOW);
        i=0;
    }

    lastState = buttonState;
}

```

โปรแกรมในส่วนของภาคส่ง โดยมีการเข้ารหัสแบบ Convolutional Code

```
int c, k;
int n = 255;
int arr[9];
int arr_convo[18];
int i_arr=0;
int i=0;
int j=0;
int size_arr;
int data[25];
int data_encode[26];
int temp = 0;
char present_state = 'a';
```

```
int buttonPin = 10;
int ledPin = 5;
boolean buttonState;
boolean lastState;
boolean state = HIGH;
```

```
void setup()
```

```
{
```

```
  pinMode (buttonPin,INPUT);
  pinMode(ledPin, OUTPUT);
  digitalWrite(ledPin ,LOW) ;
```

```
  for (c = 8; c >= 0; c--) // 0-8 have 9 digit
```

```
  {
```

```
    k = n >> c;
```

```
    if (k & 1)
```

```
      {
```

```
        arr[i_arr] = 1;
```

```
        printf("1");
```

```
      }
```

```
    else
```

```
      {
```

```
        arr[i_arr] = 0;
```

```

printf("0");
    }
        i_arr++;
}

size_arr=sizeof(arr)/sizeof(arr[0]);

for( i=0;i<size_arr;i++)
    {
        if(present_state=='a')
            {
                if(arr[i]==1){
                    arr_conv[2*i]=1;
                    arr_conv[(2*i)+1]=1;
                    present_state = 'b';
                }else if(arr[i]==0){
                    arr_conv[2*i]=0;
                    arr_conv[(2*i)+1]=0;
                    present_state = 'a';
                }
            }else if(present_state=='b'){
                if(arr[i]==1){
                    arr_conv[2*i]=1;
                    arr_conv[(2*i)+1]=0;
                    present_state = 'd';
                }else if(arr[i]==0){
                    arr_conv[2*i]=0;
                    arr_conv[(2*i)+1]=1;
                    present_state = 'c';
                }
            }else if(present_state=='c'){
                if(arr[i]==1){
                    arr_conv[2*i]=0;
                    arr_conv[(2*i)+1]=0;
                    present_state = 'b';
                }else if(arr[i]==0){
                    arr_conv[2*i]=1;
                    arr_conv[(2*i)+1]=1;

```

```

        present_state = 'a';
    }
}
else if(present_state=='d'){
    if(arr[i]==1){
        arr_convo[2*i]=0;
        arr_convo[(2*i)+1]=1;
        present_state = 'd';
    }else if(arr[i]==0){
        arr_convo[2*i]=1;
        arr_convo[(2*i)+1]=0;
        present_state = 'c';
    }
}
}

for(i=0;i<25;i++){
    if( (i>=0) && (i<=4) ){
        data[i]=1;
    }else if((i>=5)&&(i<=22)){
        data[i] = arr_convo[j];
        j++;
    }else if((i>=23)&&(i<=24)){
        data[i] = 0;
    }
}

for(j=0;j<=24;j++){
    if(temp!=data[j]){
        data_encode[j]=1;
    }else data_encode[j]=0;
    temp = data_encode[j];
}
}

```

```

void loop()
{
  buttonState = digitalRead(buttonPin);

  if ( buttonState == LOW && lastState == HIGH )
  {
    delay (10);
    if (digitalRead(buttonPin) == LOW)
    {
      state = !state;
    }
  }
  if(state == LOW)
  {
    digitalWrite(ledPin,data_encode[i]);
    delay(33.33333333);
    i=(i+1)%26;
  }else {
    digitalWrite(ledPin,LOW);
    i=0;
  }

  lastState = buttonState;
}

```

ภาคผนวก ข.  
โปรแกรมระบบตรวจสอบอัตลักษณ์ด้วยแสงในส่วนภาครับ

## โปรแกรมในส่วนของภาครับ โดยมีการถอดรหัสแบบ Convolutional Code

```
//+++++ Convolution code +++++//
```

```
public int k;
public int BM_00_to_a;
public int BM_11_to_a;
public int BM_00_to_b;
public int BM_11_to_b;
public int BM_01_to_c;
public int BM_10_to_c;
public int BM_10_to_d;
public int BM_01_to_d;
public int Part_A;
public int Part_B;
public int Part_C;
public int Part_D;
public int min_BM_A;
public int min_BM_B;
public int min_BM_C;
public int min_BM_D;
public string last_state_at_a = "a";
public string last_state_at_b = "b";
public string last_state_at_c = "c";
public string last_state_at_d = "d";
public string last_state;
public string inputdata_A;
public string inputdata_B;
public string inputdata_C;
public string inputdata_D;
public string temp_inputdata_A;
public string temp_inputdata_B;
public string temp_inputdata_C;
public string temp_inputdata_D;
public string new_mess_a;
public string new_mess_b;
public string new_mess_c;
public string new_mess_d;
public string last_new_mess;
public string Last_message;
```

```

public string Present_state;
public string mess_input;
public string inputdata;
public string temp_message_at_a;
public string temp_message_at_b;
public string temp_message_at_c;
public string temp_message_at_d;
public int[,] PM = new int[10, 4];
//+++++++//

//+++++++ convolution code ++++++//
private int viterbi_decode(int mes_decial)
{
    new_mess_a = null;
    new_mess_b = null;
    new_mess_c = null;
    new_mess_d = null;
    inputdata_A = null;
    inputdata_B = null;
    inputdata_C = null;
    inputdata_D = null;
    mess_input = null;
    temp_inputdata_A = null;
    temp_inputdata_B = null;
    temp_inputdata_C = null;
    temp_inputdata_D = null;

    //Convert to binary in a string
    string s_mes_decial = Convert.ToString(mes_decial, 2);

    int[] bits = s_mes_decial.PadLeft(16, '0') // Add 0's from left
        .Select(c => int.Parse(c.ToString())) // convert each char to int
        .ToArray(); // Convert IEnumerable from select to Array

```

```

for (k = 0; k < (bits.Length) / 2; k++)
{
    if (k == 0)
    {
        //at A
        BM_00_to_a = (0 ^ bits[k * 2]) + (0 ^ bits[(k * 2) + 1]);           // 00
        //at B
        BM_11_to_b = (1 ^ bits[k * 2]) + (1 ^ bits[(k * 2) + 1]);           // 11

        PM[k, 0] = 0 + BM_00_to_a;
        PM[k, 1] = 0 + BM_11_to_b;

        if (PM[k, 0] < PM[k, 1])
        {
            last_state = "a";
            Present_state = "a";
        }
        else if (PM[k, 0] > PM[k, 1])
        {
            last_state = "a";
            Present_state = "b";
        }
    }

    new_mess_a = Last_State("a", "a");
    inputdata_A = new_mess_a;
    temp_inputdata_A = inputdata_A;
    new_mess_b = Last_State("a", "b");
    inputdata_B = new_mess_b;
    temp_inputdata_B = inputdata_B;

    last_new_mess = Last_State(last_state, Present_state);

    mess_input += last_new_mess;
}
else if (k == 1)
{

```

```

//at A 00 ---- > last state is a
BM_00_to_a = (0 ^ bits[k * 2]) + (0 ^ bits[(k * 2) + 1]);

//at b 11 ---- > last state is a
BM_11_to_b = (1 ^ bits[k * 2]) + (1 ^ bits[(k * 2) + 1]);

//at c 01 ---- > last state is b
BM_01_to_c = (0 ^ bits[k * 2]) + (1 ^ bits[(k * 2) + 1]);

//at d 10 ---- > last state is b
BM_10_to_d = (1 ^ bits[k * 2]) + (0 ^ bits[(k * 2) + 1]);

PM[k, 0] = PM[k - 1, 0] + BM_00_to_a;
PM[k, 1] = PM[k - 1, 0] + BM_11_to_b;
PM[k, 2] = PM[k - 1, 1] + BM_01_to_c;
PM[k, 3] = PM[k - 1, 1] + BM_10_to_d;

if ((PM[k, 0] < PM[k, 1]) && (PM[k, 0] < PM[k, 2]) && (PM[k, 0] < PM[k,
3]))
{
    last_state = "a";
    Present_state = "a";
}
else if ((PM[k, 1] < PM[k, 0]) && (PM[k, 1] < PM[k, 2]) && (PM[k, 1] <
PM[k, 3]))
{
    last_state = "a";
    Present_state = "b";
}
else if ((PM[k, 2] < PM[k, 0]) && (PM[k, 2] < PM[k, 1]) && (PM[k, 2] <=
PM[k, 3]))
{
    last_state = "b";
    Present_state = "c";
}
else if ((PM[k, 3] < PM[k, 0]) && (PM[k, 3] < PM[k, 1]) && (PM[k, 3] <
PM[k, 2]))
{

```

```

        last_state = "b";
        Present_state = "d";
    }

    new_mess_a = Last_State("a", "a");
    inputdata_A = temp_inputdata_A + new_mess_a;

    new_mess_b = Last_State("a", "b");
    inputdata_B = temp_inputdata_A + new_mess_b;

    new_mess_c = Last_State("b", "c");
    inputdata_C = temp_inputdata_B + new_mess_c;

    new_mess_d = Last_State("b", "d");
    inputdata_D = temp_inputdata_B + new_mess_d;

    temp_inputdata_A = inputdata_A;
    temp_inputdata_B = inputdata_B;
    temp_inputdata_C = inputdata_C;
    temp_inputdata_D = inputdata_D;

    last_new_mess = Last_State(last_state, Present_state);
    mess_input += last_new_mess;

}
else if (k > 1)
{
    //at a 00 ---- > last state is a
    BM_00_to_a = (0 ^ bits[k * 2]) + (0 ^ bits[(k * 2) + 1]);
    //at a 11 ---- > last state is c
    BM_11_to_a = (1 ^ bits[k * 2]) + (1 ^ bits[(k * 2) + 1]);

    //at b 11 ---- > last state is a
    BM_11_to_b = (1 ^ bits[k * 2]) + (1 ^ bits[(k * 2) + 1]);
    //at b 00 ---- > last state is c
    BM_00_to_b = (0 ^ bits[k * 2]) + (0 ^ bits[(k * 2) + 1]);
}

```

```

//at c 01 ---- > last state is b
BM_01_to_c = (0 ^ bits[k * 2]) + (1 ^ bits[(k * 2) + 1]);
//at c 10 ---- > last state is d
BM_10_to_c = (1 ^ bits[k * 2]) + (0 ^ bits[(k * 2) + 1]);

//at d 10 ---- > last state is b
BM_10_to_d = (1 ^ bits[k * 2]) + (0 ^ bits[(k * 2) + 1]);
//at d 01 ---- > last state is d
BM_01_to_d = (0 ^ bits[k * 2]) + (1 ^ bits[(k * 2) + 1]);

//At a
if ((PM[k - 1, 0] + BM_00_to_a) <= (PM[k - 1, 2] + BM_11_to_a))
{
    min_BM_A = PM[k - 1, 0] + BM_00_to_a;
    last_state_at_a = "a";
    temp_message_at_a = temp_inputdata_A;
}
else if ((PM[k - 1, 0] + BM_00_to_a) > (PM[k - 1, 2] + BM_11_to_a))
{
    min_BM_A = PM[k - 1, 2] + BM_11_to_a;
    last_state_at_a = "c";
    temp_message_at_a = temp_inputdata_C;
}

//At b
if ((PM[k - 1, 0] + BM_11_to_b) <= (PM[k - 1, 2] + BM_00_to_b))
{
    min_BM_B = PM[k - 1, 0] + BM_11_to_b;
    last_state_at_b = "a";
    temp_message_at_b = temp_inputdata_A;
}
else if ((PM[k - 1, 0] + BM_11_to_b) > (PM[k - 1, 2] + BM_00_to_b))
{
    min_BM_B = PM[k - 1, 2] + BM_00_to_b;

```

```

last_state_at_b = "c";
temp_message_at_b = temp_inputdata_C;

}

//At c
if ((PM[k - 1, 1] + BM_01_to_c) <= (PM[k - 1, 3] + BM_10_to_c))
{
    min_BM_C = PM[k - 1, 1] + BM_01_to_c;
    last_state_at_c = "b";
    temp_message_at_c = temp_inputdata_B;

}
else if ((PM[k - 1, 1] + BM_01_to_c) > (PM[k - 1, 3] + BM_10_to_c))
{
    min_BM_C = PM[k - 1, 3] + BM_10_to_c;
    last_state_at_c = "d";
    temp_message_at_c = temp_inputdata_D;

}

//At d
if ((PM[k - 1, 1] + BM_10_to_d) <= (PM[k - 1, 3] + BM_01_to_d))
{
    min_BM_D = PM[k - 1, 1] + BM_10_to_d;
    last_state_at_d = "b";
    temp_message_at_d = temp_inputdata_B;

}
else if ((PM[k - 1, 1] + BM_10_to_d) > (PM[k - 1, 3] + BM_01_to_d))
{
    min_BM_D = PM[k - 1, 3] + BM_01_to_d;
    last_state_at_d = "d";
    temp_message_at_d = temp_inputdata_D;

}

```

```
new_mess_a = Last_State(last_state_at_a, "a");
inputdata_A = temp_message_at_a + new_mess_a;
```

```
new_mess_b = Last_State(last_state_at_b, "b");
inputdata_B = temp_message_at_b + new_mess_b;
```

```
new_mess_c = Last_State(last_state_at_c, "c");
inputdata_C = temp_message_at_c + new_mess_c;
```

```
new_mess_d = Last_State(last_state_at_d, "d");
inputdata_D = temp_message_at_d + new_mess_d;
```

```
temp_inputdata_A = inputdata_A;
temp_inputdata_B = inputdata_B;
temp_inputdata_C = inputdata_C;
temp_inputdata_D = inputdata_D;
```

```
PM[k, 0] = min_BM_A;
PM[k, 1] = min_BM_B;
PM[k, 2] = min_BM_C;
PM[k, 3] = min_BM_D;
```

```
Part_A = PM[k, 0];
Part_B = PM[k, 1];
Part_C = PM[k, 2];
Part_D = PM[k, 3];
```

```
if ((PM[k, 0] < PM[k, 1]) && (PM[k, 0] < PM[k, 2]) && (PM[k, 0] < PM[k, 3]))
{
    last_state = last_state_at_a;
    Present_state = "a";
}
else if ((PM[k, 1] < PM[k, 0]) && (PM[k, 1] < PM[k, 2]) && (PM[k, 1] <
PM[k, 3]))
{
    last_state = last_state_at_b;
    Present_state = "b";
}
```

```

else if ((PM[k, 2] < PM[k, 0]) && (PM[k, 2] < PM[k, 1]) && (PM[k, 2] <
PM[k, 3]))
{
    last_state = last_state_at_c;
    Present_state = "c";
}
else if ((PM[k, 3] < PM[k, 0]) && (PM[k, 3] < PM[k, 1]) && (PM[k, 3] <
PM[k, 2]))
{
    last_state = last_state_at_d;
    Present_state = "d";
}

last_new_mess = Last_State(last_state, Present_state);
mess_input += last_new_mess;

}

}

if ((Part_A < Part_B) && (Part_A < Part_C) && (Part_A < Part_D))
{
    Last_message = inputdata_A;
}
else if ((Part_B < Part_A) && (Part_B < Part_C) && (Part_B < Part_D))
{
    Last_message = inputdata_B;
}
else if ((Part_C < Part_A) && (Part_C < Part_B) && (Part_C < Part_D))
{

```

```

        Last_message = inputdata_C;
    }
    else if ((Part_D < Part_A) && (Part_D < Part_B) && (Part_D < Part_C))
    {

        Last_message = inputdata_D;
    }
    else return -1;

    int letter_decode = 0;
    for (int i = 0; i < Last_message.Length; i++)
    {
        // we start with the least significant digit, and work our way to the left
        if (Last_message[Last_message.Length - i - 1] == '0') continue;
        letter_decode += (int)Math.Pow(2, i);
    }

    return letter_decode;

}

private string Last_State(string laststate, string present)
{
    if (present == "a")
    {
        if (laststate == "a")
            inputdata = "0";
        else if (laststate == "c")
            inputdata = "0";
    }
    else if (present == "b")
    {
        if (laststate == "a")
            inputdata = "1";
        else if (laststate == "c")
            inputdata = "1";
    }
}

```

```
}
else if (present == "c")
{
    if (laststate == "b")
        inputdata = "0";
    else if (laststate == "d")
        inputdata = "0";
}
else if (present == "d")
{
    if (laststate == "b")
        inputdata = "1";
    else if (laststate == "d")
        inputdata = "1";
}
return inputdata;
}
```

ภาคผนวก ค.

วารสารทางวิชาการที่ได้รับการตีพิมพ์ในการประชุมทางวิชาการ  
The 4<sup>th</sup> Joint International Conference on Information and  
communication Technology, Electronic and Electrical Engineering  
(JICTEE) 2014, 5-8 March 2014, Chiang Rai, Thailand

# Identity Checking System by Visible Light

Boonchana Purahong, Thanavit Anuwongpinit and Jirattawut Dumrong

Faculty of Engineering,

King Mongkut's Institute of Technology Ladkrabang

Bangkok, Thailand

kpboonch@kmitl.ac.th, mr.neng49@gmail.com, Jirattawut@gmail.com

**Abstract**— A new identity checking system by visible light can fulfill some limitations of QRCode and RFID technology wherewith this system can check further distance. This paper is a model to be applied to other areas. In this system, the communication between LED Blinking and camera consists of transmitter and receiver. Transmitter is a microcontroller to control LED for blinking pattern one pattern for one identity of object and save in database. Receiver consists of camera to detect LED blinking by image processing. When detecting LED blinking receive code for identity of object then get code to check identity with database. This system can be applied to various such as security, advertising, identify system, and etc.

**Keywords**—LED blinking; Identity checking; Blinking pattern

## I. INTRODUCTION

Nowaday there are technologies for identity checking such as QRCode[1] or RFID which are used for specify object and identity but both technology have limitations that cannot communicate far distance. QRCode has a low bit-rate and angle for a receiver is  $90^\circ$  only while RFID is radio wave, having a regulation. LED are expected as next generation lighting sources. There have been some related system in the past. For example, used infared LED in museum[2], ID Recognizing systems with IR-LEDs [3] and some related systems used smart image sensor [4][5] LEDs are superior to conventional incandescent lights due to their low power consumption, strong directivity, good visibility, long life and low heat generation. Also, LEDs are able to control their intensity electrically at a fast rate, because, it is semiconductor devices. LEDs can be used not only as illuminating devices, but also as communication device. Some cases of LED being used as communication device are LED blinking. VLC is a wireless communication method using luminance, transmitting data by emitting light. It is able to transfer by emitting light source, and able to receive them by light receiver. There are several advantages in this communication method compared to other wireless communication methods including radio waves. The visible light is not harmful to human body and it is able to be transmitted with high power. There is no legal limitation for any existing light sources, such as room illuminations and displays. It can be used at the place where radio waves cannot be used namely in hospital and areas around precision machines. Illuminate from LED not only illuminate but also use to send data by different blinking pattern may be blinking in high frequency to frequency is human cannot see how to blinking and utilizes the intensity of light among them[6]. Receiver can use photodiode or image sensor but, this system

using image sensor because image sensor can receive multiple signals, robustness and long distance more than photodiode.

Using an inexpensive webcam camera, we performed an experiment of identifying identity by attaching LEDs. Compared to previous methods, LED periodically blinks in a special pattern. We developed an efficient technique for interpreting the light patterns, hence making it possible to check identity. Although the camera does not guarantee a fixed rate of each frame, it is possible to identify patterns with high accuracy, which will be described in this paper. Since simple cameras are already enclosed in many portable devices such as mobile phones or laptop computers, the proposed system may be practiced without much cost. Moreover, this system can be used in security system, advertising, identifying people in night and in related works. It can be applied with photogrammetric system[7].

This paper consists of five main sections to explain our system. System overview is explained in section II. Section III explains the transmitter and receiver method, pattern analysis and shift bit. The experiment results are described in section IV and the section V is the conclusion.

## II. SYSTEM OVERVIEW

Our system is composed of 2 main points, transmitter and receiver. Transmitter has a microcontroller to LED blinking. LEDs is pattern on-off in 1 second. That one blinking pattern instead of one identity of object by bit 1 is on and bit 0 is off and get code blinking pattern of object to save in database. Receiver consists of camera used for detecting LED blinking by image processing using C# compatible with OpenCV, EmguCV and AForge.NET library. When detect LED blinking, receive code and then check code with database.

Fig. 1 shows overview of this system composed of microcontroller to control blinking pattern and camera to detect LED blinking and computer with GUI for received data from camera and process by image processing and managing database.

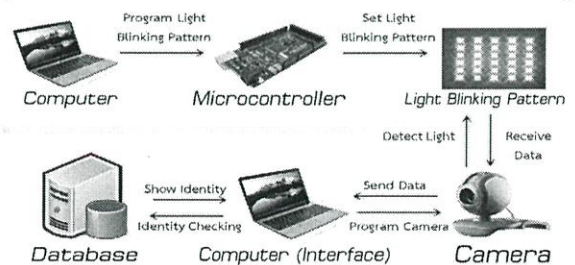


Fig. 1. System overview.

A. Transmitter Method

Transmitter method composed of 2 sections. First, microcontroller is a method to control LED blinking. Second, LEDs size is compatible for using but, this system uses 7x7 LEDs. This is because it is suitable for detecting and using. We use microcontroller adruino uno R3 to control LED blinking. We define LED Blinking pattern bit 1 for LED on and Bit 0 for LED off and blinking one pattern for one identity and save in database.

Fig.2 shows a transmitter composed of microcontroller using adruino uno R3 and blinking LED size 7x7.

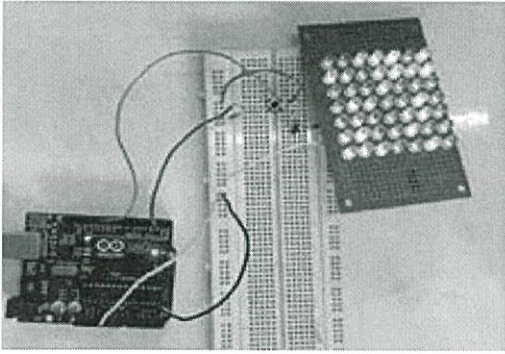


Fig. 2. Transmitter method.

B. Receiver Method

In receiver method consist inexpensive webcam camera to detect LED blinking pattern for image processing using c# with library OpenCV, EmguCV and AForge.NET and then received code from detecting we get code to check code of identity in database. If code received match with code in database that show identity or detail of such. For webcam camera, we used OKER177 in the experiment.

Fig. 3 shows a webcam camera OKER177 in experiment.

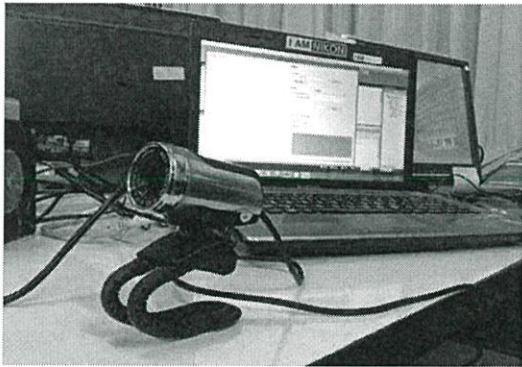


Fig. 3. Webcam camera OKER177.

III. PATTERN ANALYSIS AND SHIFT BIT

In analyzing pattern blinking LED, We used webcam camera (OKER-177) that is inexpensive camera and uses frame rate of 25~30 fps. Therefore, we used illuminance of image in each frame to calculate. When camera and program receive image already. Then convert from RGB channel to

YUV channel in (1) and use Y channel to get illuminance of image store in array. Then calculate average illuminance of all frames in array.

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.144 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.515 & -0.1 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (1)$$

This method can explain by (2). Threshold means an average illuminance, illuminance each array  $Y_i$  and total frame array  $fp$ .

$$Threshold = \frac{\sum_{i=1}^{fp} Y_i}{fp} \quad (2)$$

After that, get illuminance or Y channel from each frame in array to compare with threshold or average illuminance. If more than average or LED ON is called 'bit 1' and less than average or LED OFF is called 'bit 0' in (3) And Fig.(4) shows an algorithm to analyze blinking pattern.

$$Bit \begin{cases} \text{Bit 1 ; } Y_i > Threshold \\ \text{Bit 0 ; } Y_i < Threshold \end{cases} \quad (3)$$

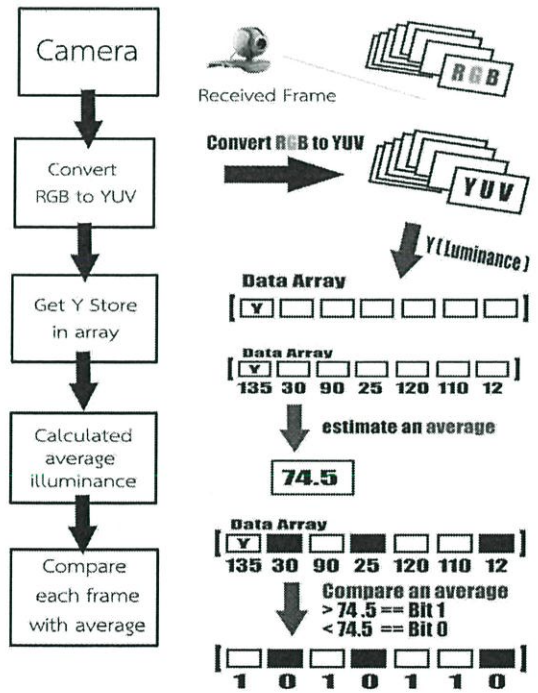


Fig. 4. Algorithm to analyze blinking pattern.

Some case, transmitter may be unequal synchronizing with receiver. Therefore set receive bit to 3 section include header bit, data bit and close bit. When camera detect LED blinking ,receive bit data. Bit data must be shifted to find header bit and close bit by defining in program. When we find header bit and close bit, we used data bits to check identity with database.

Table I shows bit pattern format includes header bits, data bits and close bits for transmitting. Bit '1' is means LED on and Bit 0 is means LED off. LED will blinking this pattern in 1 minute periodically.

TABLE I. PATTERN FORMAT

Header Bits	Data Bits	Close Bits
111110	$d_1d_2d_3\dots d_{18}$	00

#### IV. EXPERIMENTAL RESULTS

We do experiment in day and night and in the same place. Distance from transmitter between 1m – 50m. Data correct means the program can receive Header bits, Data Bits and Close Bits correctly. Being unable to receive data means program cannot find Header bits in 1 second then drop all frame in array and receive again. Data error means program can find header bit, data bits and close bits, but data bits are not correctly.

Table II shows experimental results in day using detect blinking LED in 1 minute to find receive data correct rate, cannot receive data rate and

TABLE II. EXPERIMENTAL RESULTS IN DAY

Distance (m.)	1	5	10	20	30	40	50
Illuminance of environment (lux)	852	1590	1290	1320	1330	1580	1460
Number of times to receive data	65	68	66	67	66	66	66
Data correct	42	36	36	27	16	0	0
Cannot receive data	16	25	23	35	33	56	62
Data error	7	7	7	5	14	10	4
% receive data correct	64.62	52.94	54.55	40.30	24.24	0.00	0.00
% cannot receive data	24.62	36.76	34.85	52.24	50.00	84.85	93.94
% receive error data	10.77	10.29	10.61	7.46	21.21	15.15	6.06

From experimental results in day, you can see a stable data correct rate in 1m – 10m and decrease when distance 20m and cannot detect in distance 40m – 50m because, illuminance of environment is high or you can see in Fig.5(e) and Fig.5(h). It cannot detect blinking LED and cannot get illuminance or Y channel from each frame in array to compare with threshold.

Fig.5 shows example of each distance in day by set blinking LED or transmitter away from receiver each distance 1m – 50m and you can scan QRCode on a pictures to see a motion picture video.

Table III shows experimental results in day using detect blinking LED in 1 minute to find receive data correct rate, cannot receive data rate and receive data error rate.

From experimental results in night, stable data correct rate along 1m – 50m is shown because illuminance of environment is low and can get illuminance or Y channel from each frame in array to compare with threshold.

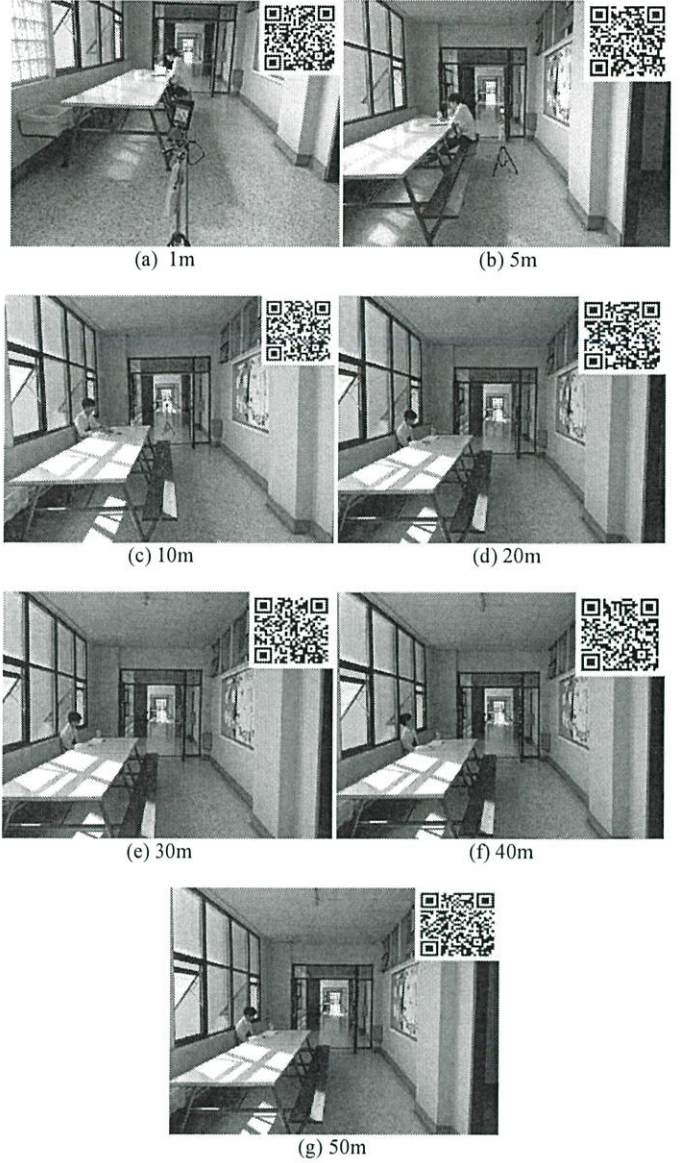


Fig. 5. Example of each distance in day.

TABLE III. EXPERIMENTAL RESULTS IN NIGHT

Distance (m.)	1	5	10	20	30	40	50
Illuminance of environment (lux)	1.03	0.99	0.55	0.33	0.55	0.66	0.99
Number of times to receive data	66	67	67	66	66	66	66
Data correct	30	34	34	32	37	38	41
Cannot receive data	27	25	28	23	23	21	23
Data error	8	3	7	8	9	6	7
% receive data correct	45.45	50.75	50.75	48.48	56.06	57.58	62.12
% cannot receive data	40.91	37.31	41.79	34.85	34.85	31.82	34.85
% receive error data	12.12	4.48	10.45	12.12	13.64	9.09	10.61

Fig.6 shows example of each distance in night by set blinking LED or transmitter away from receiver each distance 1m – 50m and you can scan QRCode on a pictures to see a motion picture video.

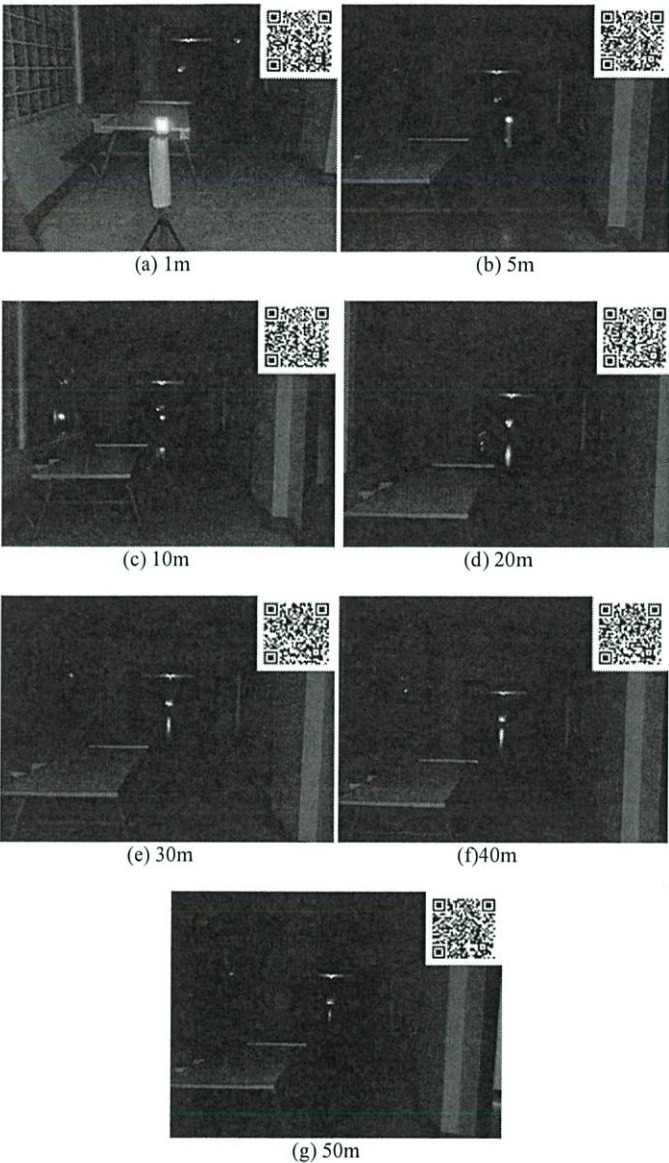


Fig. 6. Example of each distance in night.

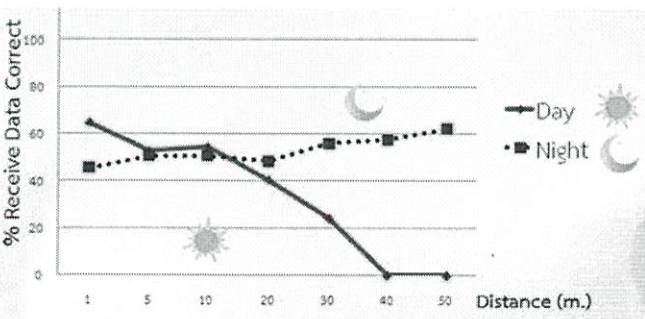


Fig. 7. Rate of received data correct.

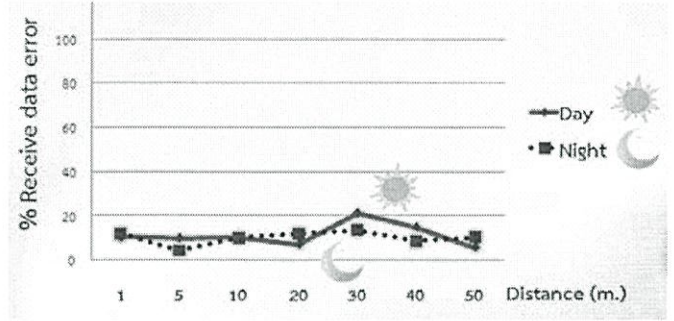


Fig. 8. Rate of received data error.

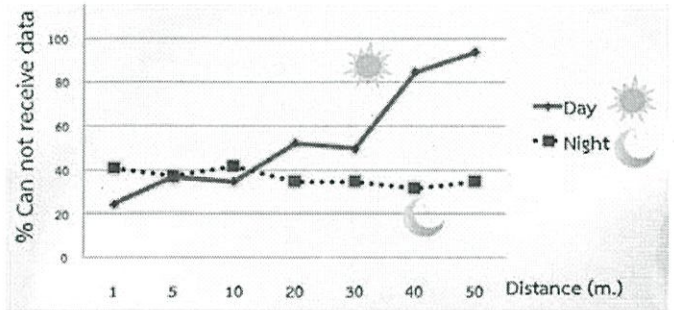


Fig. 9. Rate of can not received data.

Fig.7, Fig.8 and Fig.9 show compared graphs in day and night. Fig.7 is receive data correct rate. Fig.8 is receive data error rate and fig.9 is can not receive data rate.

GUI of system is show in Fig.10. First part of GUI is picture of camera detect and then process only red framework in picture and show received data in decimal number and then check identity with database. You can scan QRCode to see a motion picture video of GUI.

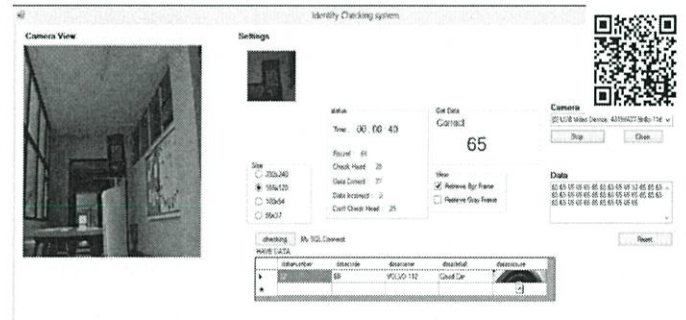


Fig. 10. GUI of system.

From experiment in day and night, it can see that in day, luminance of environment is high so distance 1m – 20m can detect LED Blinking and receive code until distance is far ( 20m – 50m ) effectiveness for camera detection so decreased also can not detect. In night is luminance of environment is low, all distance 1m – 50m have a effectiveness of camera detection along distance. Therefore, in day and night distance 1m – 10m have a same effectiveness to detect LED Blinking but in day distance 20m – 50m so decreased until can not detect because, luminance of environment high is affect to

average luminance used in the calculation more night. Therefore, we find a method to make more efficiency to increase detect rate and decrease error rate. We explain in the next section.

## V. CONCLUSION AND FUTURE WORK

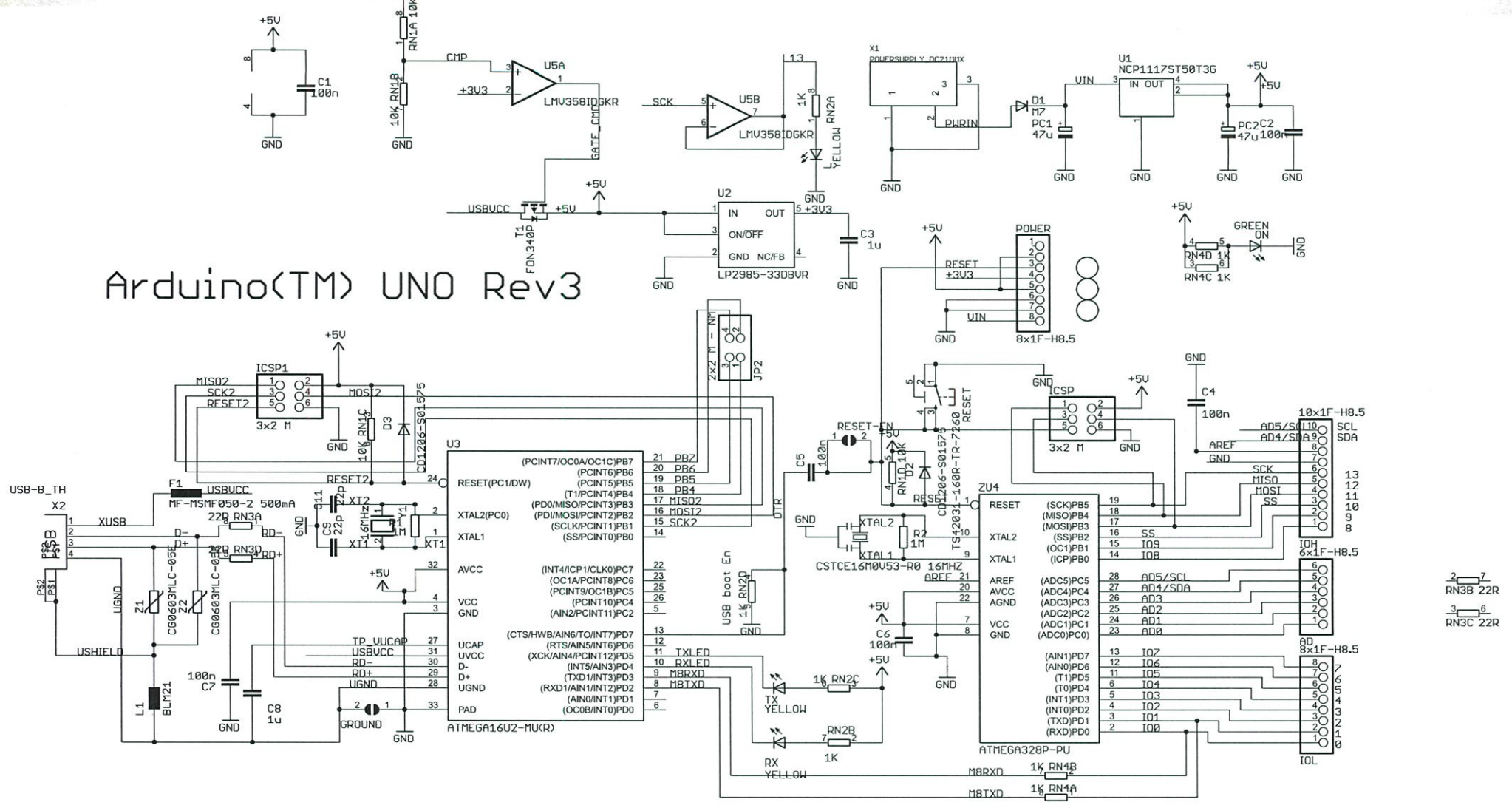
In this paper, we discussed a new identity checking system by visible light and have proposed using camera to detect LED Blinking and process frame by image processing to get a code for check identity with database and have confirm their effectiveness. Experiment results have shown that receiver can detect LED blinking in distance between 1m to 50m in night and 1m to 20m in day and use zoom technique to increase detect rate 1m to 50m in day. This system model can applied for various system such as security, advertising, identify people in night, etc.

Future Work, we will study tracking LED Blinking and detect multiple transmitter and decreased error rate and increase detect rate more than.

## REFERENCES

- [1] Boonchana Purahong and Pikulkaew Tangtisanon: "Motion QR code detecting application" Proceedings 5th AUN/SEED-Net Regional Conference on Information and Communications Technology, October 18-19, 2012, Manila, Philippines.
- [2] Jun Park, Kyu Hwa Lee, Sang Goo Lee, " ID Recognition for Museum Tourists Using Infrared LED's " Second Workshop on Digital Media and its Application in Museum & Heritage p.115-117
- [3] H.Aoki, D.Matsushita., "Balloon Tag (In)visible Marker Which Tells Who's Who", The Fourth International Symposium on Wearable Computers2000,pp.181-182
- [4] Y. Oike, M. Ikeda, and K. Asada, "A Smart Image Sensor With High-Speed Feeble ID-Beacon Detection for Augmented Reality System", Proc. of IEEE European Solid-State Circuits Conference (ESSCIRC) , Sep. 2003, pp.125 - 128.
- [5] Nobuyuki Matsushita, Daisuke Hihara, Teruyuki Ushiro, Shinichi Yoshimura, Jun Rekimoto, Yoshikazu Yamamoto, "ID CAM: A Smart Camera for Scene Capturing and ID Recognition", The Second IEEE and ACM International Symposium on Mixed and Augmented Reality, p.227-236.
- [6] Leonid Naimark & Eric Foxlin, "Encoded LED System for Optical Tracker", Proceedings of the International Symposium on Mixed and Augmented Reality, ISMAR 2005.
- [7] Hideaki Uchiyama, Masaki Yoshino, Hideo Saito, Masao Nakagawa, Shinichiro Haruyama Takao Kakehashi, Naoki Nagamoto, "Photogrammetric System using Visible Light Communication", IEEE 2008.

ภาคผนวก ง.  
ข้อมูลและคุณสมบัติต่างๆของอุปกรณ์



Reference Designs ARE PROVIDED "AS IS" AND "WITH ALL FAULTS. Arduino DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Arduino may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Arduino reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this information. ARDUINO is a registered trademark. Use of the ARDUINO name must be compliant with <http://www.arduino.cc/en/Main/Policy>