ระบบรักษาความปลอดภัยและแจ้งเตือนผ่าน SMS
SECURITY SYSTEM AND ALERTS VIA SMS

โดย

นางสาวธนาวรรณ พันธุ์สวัสดิ์

นางสาวประกายแก้ว ศรีกระชา

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2556

ระบบรักษาความปลอดภัยและแจ้งเตือนผ่าน SMS
SECURITY SYSTEM AND ALERTS VIA SMS

โดย

นางสาวธนาวรรณ พันธุ์สวัสดิ์

นางสาวประกายแก้ว ศรีกระชา

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2556

ระบบรักษาความปลอดภัยและแจ้งเตือนผ่าน SMS
SECURITY SYSTEM AND ALERTS VIA SMS

โดย

นางสาวธนาวรรณ พันธุ์สวัสดิ์ 53010688

นางสาวประกายแก้ว ศรีกระชา 53010923

อาจารย์ที่ปรึกษา

ดร.สิรภพ ตู้ประกาย

รศ.ดร.กอบชัย เดชหาญ

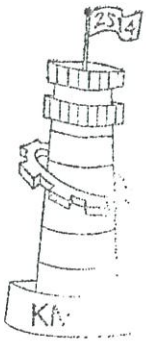
ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2556

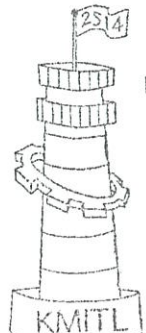


ผ่านการตรวจรูปเล่มแล้ว

(.....)

อาจารย์ที่ปรึกษา

13/3/57



ผ่านการตรวจชิ้นงานแล้ว

(.....)

กรรมการผู้ตรวจชิ้นงาน

1/เม.ย 57

ปริญญานิพนธ์ปีการศึกษา 2556
สาขาวิชาวิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
เรื่อง ระบบรักษาความปลอดภัยและแจ้งเตือนผ่าน SMS
SECURITY SYSTEM AND ALERTS VIA SMS

ผู้จัดทำ
1. นางสาวธนาวรรณ พันธุ์สวัสดิ์ 53010688
2. นางสาวประกายแก้ว ศรีกระชา 53010923

..... อาจารย์ที่ปรึกษา
(ดร.สิรภพ ตู้ประกาย)

..... อาจารย์ที่ปรึกษาร่วม
(รศ.ดร.กอบชัย เดชหาญ)

กิตติกรรมประกาศ

โครงงานฉบับนี้เป็นประสบการณ์ที่สำคัญสำหรับผู้จัดทำเป็นอย่างมากและงานชิ้นนี้จะสำเร็จไม่ได้เลย ถ้าไม่มีบุคคลที่คอยสนับสนุนและช่วยเหลือ

ขอขอบพระคุณ ดร.สิรภพ ตู้ประกาย อาจารย์ที่ปรึกษา และ รศ.ดร.กอบชัย เดชหาญ อาจารย์ที่ปรึกษาร่วม ที่ได้ให้ความรู้ คำแนะนำและแก้ไขข้อบกพร่องต่าง ๆ ด้วยความเอาใจใส่ทุกขั้นตอน เพื่อให้การทำโครงงานฉบับนี้สมบูรณ์ที่สุด

ท้ายที่สุดขอขอบพระคุณครอบครัวที่คอยอบรมสั่งสอน รวมทั้งสนับสนุนและคอยให้กำลังใจเสมอมา และทุกท่านที่มีส่วนร่วมในความสำเร็จของโครงงานฉบับนี้ที่ไม่สามารถกล่าวไว้ ณ ที่นี้ได้หมด

นางสาวธนาวรรณ พันธุ์สวัสดิ์
นางสาวประกายแก้ว ศรีกระชา
ผู้จัดทำ

ระบบรักษาความปลอดภัยและแจ้งเตือนผ่าน SMS
SECURITY SYSTEM AND ALERTS VIA SMS

โดย นางสาวธนาวรรณ พันธุ์สวัสดิ์ 53010688
นางสาวประกายแก้ว ศรีกระชา 53010923

อาจารย์ที่ปรึกษา ดร. สิรภพ ตู้ประกาย
รศ. ดร. กอบชัย เดชหาญ

บทคัดย่อ

โครงงานนี้จัดทำขึ้นเพื่อรักษาความปลอดภัยภายในบริเวณบ้าน อาคารหรือสำนักงาน โดยการทำงานเริ่มจากเซ็นเซอร์ตรวจจับความเคลื่อนไหวเมื่อพบสิ่งผิดปกติและทำการส่งสัญญาณเอาท์พุตไปยังไมโครคอนโทรเลอร์ควบคุมให้ทำการบันทึกเก็บไว้ในเอสดีการ์ดและส่งข้อความสั้นไปยังโทรศัพท์เคลื่อนที่ของผู้ใช้งานหรือเจ้าของอาคาร นอกจากนั้นนำข้อมูลลงในฐานข้อมูล และผู้ใช้งานหรือเจ้าของอาคารสามารถตรวจสอบข้อมูลได้จากระบบอินเตอร์เน็ท

ABSTRACT

This project presents the security for home, building or office. When motion sensor detects by infrared from something that it sends output signal into microcontroller and saves in SD card. Then it sends the short messages to user's mobile phone. The user can check its on the Internet.

สารบัญ

		หน้า
กิตติกรรมประกาศ		I
บทคัดย่อ		II
สารบัญ		III
สารบัญรูป		VIII
สารบัญตาราง		XIII
บทที่ 1	**บทนำ**	**1**
	1.1 ความเป็นมาและความสำคัญของปัญหา	1
	1.2 วัตถุประสงค์	1
	1.3 ขอบเขตของโครงงาน	1
บทที่ 2	**ทฤษฎีและหลักการที่เกี่ยวข้อง**	**2**
	2.1 ไมโครคอนโทรลเลอร์ AVR	2
	2.1.1 คุณสมบัติของ AVR ATmega328	2
	2.1.2 ขาพอร์ตอินพุตเอาท์พุต	6
	2.1.3 การสื่อสารข้อมูลอนุกรมผ่านโมดูล USART	9
	2.1.3.1 คุณสมบัติที่สำคัญของโมดูล USART	12
	2.1.3.2 รีจิสเตอร์ที่เกี่ยวข้องกับการทำงานในโหมด USART	13
	2.1.4 การใช้งานโมดูล USART โหมดอะซิงโครนัส	13
	2.1.4.1รีจิสเตอร์ UDR (USART I/O Data Register)	13
	2.1.4.2 รีจิสเตอร์ UCSRA (USART Control and status Register A)	14
	2.1.4.3 รีจิสเตอร์ UCSRB (USART Control and status Register B)	15
	2.1.4.4 รีจิสเตอร์ UCSRC (USART Control and status Register C)	16

สารบัญ (ต่อ)

	หน้า
2.1.4.5 รีจิสเตอร์ UBRRL และ UBRRH (USART Baud Rate Register)	18
2.1.5 การเชื่อมต่ออุปกรณ์อนุกรม (SPI)	18
2.1.5.1 คุณสมบัติของขาพอร์ต SS	20
2.1.5.2 รีจิสเตอร์ที่เกี่ยวข้องกับการทำงานในโหมด SPI	21
1. รีจิสเตอร์ SPCR (SPI Control Register)	21
2. รีจิสเตอร์ SPSR (SPI Status Register)	23
3. รีจิสเตอร์ SPDR (SPI Data Register)	23
2.1.5.3 โหมดการรับส่งข้อมูล	24
2.1.6 ไมโครคอนโทรลเลอร์ Arduino	26
2.1.6.1 รูปแบบการเขียนโปรแกรมบน Arduino	26
2.1.6.2 Layout & Pin out Arduino Board (Model: Arduino UNO R3)	31
2.2 รังสีอินฟราเรด (Infrared Radiation)	32
2.2.1 ความหมายของรังสีอินฟราเรด	32
2.2.2 ประเภทของคลื่นแม่เหล็กไฟฟ้า	33
2.2.2.1 รังสีแกมมา (Gamma ray)	34
2.2.2.2 รังสีเอ็กซ์ (X-ray)	34
2.2.2.3 รังสีอุลตราไวโอเล็ต (Ultraviolet radiation)	34
2.2.2.3 รังสีอุลตราไวโอเล็ต (Ultraviolet radiation)	34
2.2.2.4 รังสีอินฟราเรด (Infrared radiation)	34
2.2.2.5 คลื่นไมโครเวฟ (Microwave)	34
2.2.2.6 คลื่นวิทยุ (Radio wave)	34
2.2.3 ความสัมพันธ์ระหว่างความยาวคลื่นและอุณหภูมิ	35
2.3 เซ็นเซอร์ตรวจจับความเคลื่อนไหว (Motion Sensor)	37
2.3.1 ส่วนประกอบหลักของอุปกรณ์ตรวจจับการเคลื่อนไหว	37

สารบัญ (ต่อ)

	หน้า
2.3.1.1 Pyro Electric	37
2.3.1.2 Fresnel Lens	38
2.3.1.3 Circuit Control	38
2.3.2 หลักการทำงานของ Passive infrared sensors (PIR sensor)	38
2.3.3 โมดูลตรวจความเคลื่อนไหว KC7783R	39
2.4 อุปกรณ์แถบแม่เหล็ก	41
2.5 SD Card (Secure Digital Card)	41
2.5.1 คุณลักษณะของ SD Card	42
2.5.2 การติดต่อสื่อสาร SD Card	44
2.6 Infrared JPEG Camera	44
2.6.1 คุณสมบัติเบื้องต้น	45
2.6.2 จุดเชื่อมต่อการใช้งาน Infrared JPEG Camera	45
2.7 ET-GSM SIM300CZ	46
2.7.1 คุณสมบัติของบอร์ด ET-GSM SIM300CZ V1.0	47
2.8 AT-COMMAND	48
2.9 มาตรฐานการสื่อสารระบบคอมพิวเตอร์	48
2.9.1 OSI Model (Open System Interconnection Model)	48
2.9.1.1 Physical Layer	50
2.9.1.2 Data Link Layer	51
2.9.1.3 Network Layer	52
2.9.1.4 Transport Layer	53
2.9.1.5 Session Layer	54
2.9.1.6 Presentation Layer	55
2.9.1.7 Application Layer	56
2.10 FTP (File Transfer Protocol)	57
2.10.1 วิธีการทำงานของ FTP	57

สารบัญ (ต่อ)

		หน้า
	2.10.2 ประเภทข้อมูล (Data Type)	58
	2.10.3 วิธีการรับ – ส่ง	58
	2.10.4 ระบบเสริมอื่นๆในการส่งไฟล์ข้ามเครื่อง	59
	2.10.4.1 GetRight	59
	2.6.4.2 WepNFS	60
	2.11 XAMPP	60
	2.11.1 ขั้นเตรียมพร้อมก่อนติดตั้ง XAMPP	61
	2.11.2 การติดตั้ง	61
	2.11.3 การใช้งานเบื้องต้น	61
	2.12 Forward Port	63
	2.12.1 Internal IP	63
	2.13 Domain Name System (DNS)	63
	2.13.1 การทำงานของระบบ DNS	64
บทที่ 3	**การออกแบบ**	**66**
	3.1 การออกแบบ	66
	3.1.1 การออกแบบระบบการทำงานทั้งหมด	66
	3.1.2 ด้านฮาร์ดแวร์	67
	3.1.2.1 การเชื่อมต่อระหว่างเซ็นเซอร์ตรวจจับการเคลื่อนไหว (PIR Motion Sensor) กับไมโครคอนโทรเลอร์	67
	3.1.2.2 การเชื่อมต่อเซ็นเซอร์แถบแม่เหล็ก (Magnetic Sensor) กับไมโครคอนโทรเลอร์	68
	3.1.2.3 การเชื่อมต่อโมดูลกล้อง Infrared JPEG Camera กับไมโครคอนโทรเลอร์	69
	3.1.2.4 การเชื่อมต่อโมดูล GSM SIM 300CZ กับไมโครคอนโทรเลอร์	69

สารบัญ (ต่อ)

		หน้า
	3.1.2.5 การเชื่อมต่อ Ethernet shield Arduino กับไมโครคอนโทรเลอร์	70
	3.1.2.6 วงจรรวมของระบบ	71
	3.1.3 ด้านซอฟต์แวร์	72
	3.1.2.1 โปรแกรม Arduino 1.0.5-r2	72
	3.1.2.2 การติดต่อระหว่างเซ็นเซอร์ตรวจจับการเคลื่อนไหว (PIR Motion Sensor) กับไมโครคอนโทรเลอร์	72
	3.1.3.3 การติดต่อระหว่างเซ็นเซอร์แถบแม่เหล็ก (Magnetic Sensor) กับไมโครคอนโทรเลอร์	73
	3.1.3.4 การติดต่อระหว่างโมดูลกล้อง Infrared JPEG Camera กับไมโครคอนโทรเลอร์	75
	3.1.3.5 การส่งข้อความโดยใช้โมดูล GSM SIM300CZ	76
	3.1.3.6 การติดต่อสื่อสารระหว่าง FTP Server กับไมโครคอนโทรเลอร์	76
	3.1.3.7 การแสดงผลบนหน้าเว็บ Web Server (Show Image)	78
	3.1.3.8 การเชื่อมต่อฐานข้อมูล (Data base) บน FTP Server	79
	3.2 เครื่องมือที่ใช้ในการทดลอง	80
	3.3 การจัดเก็บผลการทดลอง	80
บทที่ 4	**ผลการทดลอง**	**81**
	4.1 การทดสอบสัญญาณที่ออกจากเซ็นเซอร์ตรวจจับการเคลื่อนไหว (PIR motion sensor)	81
	4.1.1 การทดสอบขอบเขตของเซ็นเซอร์ตรวจจับการเคลื่อนไหว	81

สารบัญ (ต่อ)

		หน้า
	4.2 การทดสอบสัญญาณที่ออกจากเซ็นเซอร์แถบแม่เหล็ก (Magnetic Sensor)	83
	4.2.1 วัดระยะห่างของแรงดึงดูดเซ็นเซอร์แถบแม่เหล็ก	84
	4.3 การทดสอบสัญญาณที่ออกจากโมดูลกล้อง	86
	4.4 การทดสอบการส่งข้อมูลออกทางพอร์ต Serial ผ่านวงจร MAX 232	89
	4.5 การทดสอบส่งข้อความโดยใช้ GSM Module Sim300CZ ผ่านโปรแกรม HyperTerminal	90
	4.6 ผลการทดลองส่งข้อมูลไปยังเซิร์ฟเวอร์	92
บทที่ 5	**สรุปผลและข้อเสนอแนะ**	**96**
	5.1 สรุปผล	96
	5.2 ข้อเสนอแนะ	97
	5.3 ปัญหาของโครงงาน	97
บรรณานุกรม		**98**
ภาคผนวก		
	ภาคผนวก ก PIR Motion Sensor Module - KC7783R	**100**
	ภาคผนวก ข โค้ดที่ใช้ในการทำงาน	**103**

สารบัญรูป

รูปที่		หน้า
2.1	บล็อกไดอะแกรม ATmega328	5
2.2	บล็อกไดแกรมโมดูล USART	10
2.3	รูปแบบเฟรมข้อมูลอนุกรม	11
2.4	บล็อกไดอะแกรม SPI	19
2.5	รูปแบบการรับส่งข้อมูล SPI เมื่อกำหนด CPHA = 0	24
2.6	รูปแบบการรับส่งข้อมูล SPI เมื่อกำหนด CPHA=1	25
2.7	โปรแกรมบนคอมพิวเตอร์ ผ่านทางโปรแกรม Arduino โดยใช้ USB	26
2.8	เลือกรุ่นบอร์ด Arduino ที่ต้องการ upload	27
2.9	เลือกหมายเลข Comport ของบอร์ด	28
2.10	กดปุ่ม Verify เพื่อตรวจสอบความถูกต้องและ Compile โค้ดโปรแกรม	29
2.11	Upload โค้ดโปรแกรม	30
2.12	Layout & Pin out Arduino Board (Model: Arduino UNO R3)	31
2.13	คลื่นแม่เหล็กไฟฟ้าในช่วงความยาวคลื่นต่างๆ	32
2.14	ความสัมพันธ์ระหว่างความยาวคลื่นกับอุณหภูมิ	35
2.15	การถ่ายเทพลังงานคลื่นแม่เหล็กไฟฟ้าของวัตถุในรูปแบบรังสีอินฟราเรด	36
2.16	Fresnel Lens	38
2.17	การทำงานของ Passive infrared sensors (PIR sensor)	39
2.18	โมดูลตรวจความเคลื่อนไหว KC7783R	39
2.19	จุดเชื่อมต่อการใช้งานของ โมดูลตรวจความเคลื่อนไหว KC7783R	40
2.20	แถบแม่เหล็ก	41
2.21	ไมโคร SD Card	42
2.22	รูปแบบของ SD Card	43
2.23	ขาของ SD Card	43
2.24	Infrared JPEG Camera	45
2.25	ขาที่ใช้งาน Infrared JPEG Camera	45
2.26	ET-GSM SIM300CZ	46
2.27	การส่งข้อมูลผ่านระหว่างชั้น	49
2.28	Physical Layer	51
2.29	Data Link Layer	52

สารบัญรูป (ต่อ)

รูปที่		หน้า
	2.30 Network Layer	53
	2.31 Transport Layer	54
	2.32 Session Layer	55
	2.33 Presentation Layer	56
	2.34 Application Layer	57
	2.35 XAMPP Control Panel	62
	2.36 XAMPP Control Panel กดปุ่ม start เพื่อให้ Apache ทำงาน	62
	3.1 บล็อกไดอะแกรมของระบบ	66
	3.2 การเชื่อมต่อเซ็นเซอร์ตรวจจับการเคลื่อนไหว (PIR Motion Sensor) กับไมโครคอนโทรเลอร์	67
	3.3 การเชื่อมต่อเซ็นเซอร์แถบแม่เหล็ก (Magnetic Sensor) กับไมโครคอนโทรเลอร์	68
	3.4 การเชื่อมต่อโมดูลกล้อง Infrared JPEG Camera กับไมโครคอนโทรเลอร์	69
	3.5 การเชื่อมต่อโมดูล GSM SIM300CZ กับไมโครคอนโทรเลอร์	70
	3.6 ภาพรวมของระบบ	71
	3.7 โฟลว์ชาร์ตการติดต่อระหว่างเซ็นเซอร์ตรวจจับการเคลื่อนไหว (PIR Motion Sensor) กับไมโครคอนโทรเลอร์	73
	3.8 โฟลว์ชาร์ตการติดต่อระหว่างเซ็นเซอร์แถบแม่เหล็ก (Magnetic Sensor) กับไมโครคอนโทรเลอร์	74
	3.9 ขั้นตอนการติดต่อและสั่งงานของโมดูลกล้อง Infrared JPEG Camera	75
	3.10 โฟลว์ชาร์ตการทำงานของ FTP Server	77
	3.11 โฟลว์ชาร์ตการแสดงผลบนหน้าเว็บ web server (show Image)	78
	3.12 โฟลว์ชาร์ตการเชื่อมต่อฐานข้อมูล (Data base) บน FTP Server	79
	4.1 สัญญาณเอาท์พุตเซ็นเซอร์ตรวจจับการเคลื่อนไหว เมื่อมีการตรวจจับ	81
	4.2 ระยะทางที่เซ็นเซอร์สามารถตรวจจับได้	82
	4.3 สัญญาณเอาท์พุตเซ็นเซอร์แถบแม่เหล็ก เมื่อเซ็นเซอร์แถบแม่เหล็กหลุดออกจากกัน	83
	4.4 ระยะห่างของแรงดึงดูดเซ็นเซอร์แถบแม่เหล็กกับแรงดัน	85
	4.5 ผลที่ได้จากการส่งคำสั่ง Take Picture	86
	4.6 ผลที่ได้จากการส่งคำสั่ง Read JPEG file size	86

สารบัญรูป (ต่อ)

รูปที่		หน้า
	4.7 ผลที่ได้จากการส่งคำสั่ง Stop taking picture	87
	4.8 ข้อมูลภาพผ่านทางโปรแกรม Docklight	88
	4.9 ข้อมูลภาพผ่านทางโปรแกรม Docklight (ต่อ)	89
	4.10 ผลการทดลองวัดสัญญาณที่ส่งจากไมโครคอนโทรลเลอร์และผ่านวงจร MAX232	90
	4.11 วิธีการทดลองส่งข้อความโดยใช้ GSM Module Sim300cz ผ่านโปรแกรม HyperTerminal	91
	4.12 ข้อความสั้นที่แจ้งเตือนผู้ใช้งานแจ้งเตือนเมื่อมีสิ่งผิดปกติผ่านเซ็นเซอร์	92
	4.13 ผลการทดลองที่ออกจาก serial monitor	93
	4.14 ผลการทดลองในโปรแกรม FileZillaFTP เช็คสถานะการส่งข้อมูลเข้าสู่ฐานข้อมูล	93
	4.15 ฐานข้อมูลเมื่อไม่มีการส่งข้อมูลเข้ามา	94
	4.16 ฐานข้อมูลเมื่อมีข้อมูลเข้ามา	94
	4.17 การแสดงผลบนหน้าเซิร์ฟเวอร์แสดงเป็นรูปภาพถ่ายโดยโมดูลกล้อง	95

สารบัญตาราง

ตารางที่	หน้า
2.1 หน้าที่ของขาสัญญาณของพอร์ต B (PB0-PB7)	7
2.2 หน้าที่ของสัญญาณพอร์ต C (PC0-PC6)	8
2.3 หน้าที่ของสัญญาณพอร์ต D (PD0-PD7)	9
2.4 การคำนวณอัตราบอตเรต	12
2.5 รีจิสเตอร์ UDR (USART I/O Data Register)	14
2.6 รีจิสเตอร์ UCSRA (USART Control and status Register A)	14
2.7 รีจิสเตอร์ UCSRB (USART Control and status Register B)	15
2.8 รีจิสเตอร์ UCSRC (USART Control and status Register C)	16
2.9 ตารางแสดงการเซตบิต กำหนดพาริตี้บิต	17
2.10 ตารางแสดงการเซตบิต กำหนดจำนวนบิตข้อมูล	17
2.11 ตารางแสดงการเซตบิต กำหนดขอบขาสัญญาณ XCX	18
2.12 รีจิสเตอร์ UBRRL และ UBRRH (USART Baud Rate Register)	18
2.13 ขาพอร์ต	20
2.14 รีจิสเตอร์ SPCR (SPI Control Register)	21
2.15 การกำหนดขอบขาสัญญาณ SCK	22
2.16 การกำหนดการอ่านตัวอย่างของข้อมูล	22
2.17 การกำเนิดสัญญาณความถี่	22
2.18 รีจิสเตอร์ SPSR (SPI Status Register)	23
2.19 รีจิสเตอร์ SPDR (SPI Data Register)	23
2.20 การกำหนดโหมดสัญญาณนาฬิกาในการรับส่งข้อมูล	24
2.21 เปรียบเทียบรูปแบบ SD Card	42
2.22 แสดงหน้าที่แต่ละขาของ SD Card	44
2.23 การใช้งานขาของ Infrared JPEG Camera	46
4.1 การทดสอบระยะทางที่เซ็นเซอร์สามารถตรวจจับได้	82
4.2 การทดสอบระยะห่างของแรงดึงดูดเซ็นเซอร์แถบแม่เหล็ก	84

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

เนื่องจากโลกปัจจุบันได้มีความเปลี่ยนแปลงในด้านเทคโนโลยีอย่างรวดเร็ว ในส่วนการติดต่อสื่อสารนั้นเป็นส่วนสำคัญอย่างยิ่ง โดยเฉพาะโทรศัพท์มือถือเข้ามามีบทบาทในการใช้ชีวิตประจำวันอย่างมากและได้นำโทรศัพท์มือถือมาใช้ให้เกิดประโยชน์มากขึ้น รวมทั้งในปัจจุบันมีมิจฉาชีพที่ทำการโจรกรรมสิ่งต่างๆในบ้านเรือน อาคารหรือสำนักงาน และมีแนวโน้มจะมากยิ่งขึ้น

ดังนั้น โครงงานนี้จึงจัดทำขึ้นเพื่อรักษาความปลอดภัยภายในบ้าน อาคารและสำนักงาน โดยใช้อุปกรณ์ที่เป็นเซ็นเซอร์ตรวจจับความเคลื่อนไหวช่วยในการรักษาความปลอดภัยของบ้าน อาคารและสำนักงาน จะมีการส่งสัญญาณเมื่อตรวจจับความเคลื่อนไหวได้ไปยังเจ้าของบ้าน อาคารหรือสำนักงานนั้น อีกทั้งยังมีระบบเตือนภัยโดยการส่งข้อความสั้น (SMS) ถึงโทรศัพท์เคลื่อนที่ของเจ้าของได้โดยตรง เพื่อให้รับรู้ถึงเหตุการณ์ที่เกิดขึ้นได้ทันท่วงที

1.2 วัตถุประสงค์

1) เพื่อเพิ่มความปลอดภัยภายในบริเวณที่ต้องการ
2) เพื่อศึกษาไมโครคอนโทรเลอร์และนำมาใช้อย่างมีประสิทธิภาพ
3) เพื่อศึกษาและออกแบบการใช้งานเซ็นเซอร์และกล้อง
4) เพื่อนำเทคโนโลยีมาใช้ประโยชน์และประยุกต์ใช้ในชีวิตประจำวัน

1.3 ขอบเขตของปริญญานิพนธ์

โครงงานนี้จัดขึ้นเพื่อใช้เป็นระบบรักษาความปลอดภัยโดยใช้ไมโครคอนโทรเลอร์ควบคุมการทำงาน ประกอบด้วยเซ็นเซอร์ (Sensor) 2 ชนิด คือ เซ็นเซอร์ชนิดแรกเรียกว่าเซ็นเซอร์ตรวจจับการเคลื่อนไหว (PIR Sensor) ทำการตรวจจับความเคลื่อนไหวได้ในระยะ 5 เมตร เซ็นเซอร์ชนิดที่สองเรียกว่า เซ็นเซอร์แถบแม่เหล็ก (Magnetic Sensor) ทำการตรวจจับความผิดปกติการเปิดปิดประตูหรือหน้าต่าง การทำงานของระบบเริ่มต้นจากเซ็นเซอร์ มีการตรวจจับได้เมื่อมีผู้บุกรุกผ่านเข้ามา เซ็นเซอร์ทำการส่งสัญญาณเอาท์พุตไปยังไมโครคอนโทรเลอร์ควบคุมให้ GSM โมดูลส่งข้อความสั้น (SMS) ไปยังโทรศัพท์เคลื่อนที่ของผู้ใช้งาน หลังจากนั้นกล้องจับภาพบันทึกลงเอสดีการ์ด (SD Card) โดยภาพมีขนาด 3560 ไบต์ และส่งข้อมูลไปยังเซิร์ฟเวอร์ (Server) เพื่อสามารถตรวจสอบข้อมูลได้จากระบบอินเตอร์เน็ท (Internet)

บทที่ 2
ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 ไมโครคอนโทรลเลอร์ AVR

ไมโครคอนโทรเลอร์ AVR เป็นหนึ่งในไมโครคอนโทรลเลอร์ที่ผลิตโดยบริษัท ATMEL (ผู้นำทางด้านไมโครคอนโทรลเลอร์ตระกูล MCS-51) AVR เป็นไมโครคอนโทรลเลอร์ตระกูลใหม่จาก ATMEL มีสถาปัตยกรรมแบบ RISC (Reduce Instruction Set Computer) คือหนึ่งคำสั่งทำงานใช้สัญญาณนาฬิกาเพียง 1 ลูก (instructions in a single clock cycle) เป็นไมโครคอนโทรลเลอร์ที่มีประสิทธิภาพและความสามารถสูง แบ่งออกเป็นหลายอนุกรม ในแต่ละอนุกรมยังแบ่งออกเป็นหลายเบอร์ เพื่อรองรับความต้องการที่แตกต่างของผู้ใช้งาน ในขณะที่ยังคงประสิทธิภาพที่เท่ากัน

2.1.1 คุณสมบัติของ AVR ATmega328

2.1.1.1 เป็นไมโครคอนโทรลเลอร์ขนาด 8 บิต ประสิทธิภาพสูงแต่ใช้พลังงานต่ำ

2.1.1.2 สถาปัตยกรรมแบบ RISC (Reduce Instruction Set Computer)

1) มีคำสั่งควบคุม 131 คำสั่ง โดยมีความเร็วในการประมวลผล 1 คำสั่งต่อ 1 สัญญาณนาฬิกา
2) มีรีจิสเตอร์ใช้งานทั่วไปขนาด 8 บิต จำนวน 32 ตัว
3) ทำงานได้สูงสุดที่ 20 ล้านคำสั่งต่อวินาที (MIPS) เมื่อใช้สัญญาณนาฬิกา 20 เมกะเฮิตร์ (MHz)

2.1.1.3 หน่วยความจำ

1) หน่วยความจำแฟลชสำหรับโปรแกรมขนาด 32 กิโลไบต์ เขียน/ลบได้ 10,000 ครั้ง
2) หน่วยความจำแบบ EEPROM ขนาด 1 กิโลไบต์ เขียน/ลบได้ 100,000 ครั้ง
3) หน่วยความจำแรมชนิดเอสแรม (SRAM) ขนาด 2 กิโลไบต์
4) เก็บข้อมูลได้ 20 ปีที่อุณหภูมิ 85°C และกว่า 100 ปีที่อุณหภูมิ 25°C

2.1.1.4 มีระบบโปรแกรมตัวเองอยู่ในตัวชิพ

2.1.1.5 สามารถทำการอ่านขณะเขียนได้จริง โดยสามารถล็อกการทำงานได้เพื่อความปลอดภัยของซอฟต์แวร์

2.1.1.6 คุณสมบัติการเชื่อมต่อกับอุปกรณ์ภายนอก

1) มีตัวตั้งเวลาและตัวนับขนาด 8 บิต จำนวน 2 ตัว ที่สามารถแยกโหมดการทำงานจากกันได้ 2 โหมด คือโหมดปรีสเกลเลอร์และ โหมดเปรียบเทียบ

2) มีตัวตั้งเวลาและตัวนับขนาด 16 บิต จำนวน 1 ตัว ที่แยกโหมดการทำงานได้ 3 โหมด คือโหมดปรีสเกลเลอร์, โหมดเปรียบเทียบและโหมดตรวจจับสัญญาณอินพุต

3) มีตัวนับแบบเวลาจริง (Real Time Counter) ที่แยกวงจรกำหนดความถี่ได้

4) มี PWM (Pulse Width Modulator) จำนวน 8 ช่องสัญญาณ

5) มีตัวแปลงสัญญาณอะนาล็อกเป็นดิจิตอล (ADC) ขนาด 10 บิตจำนวน 6 ช่องสัญญาณ

6) มีพอร์ตสื่อสารอนุกรม USART (Universal Asynchronous Receiver Transmitters)

7) เชื่อมประสานอนุกรมแบบ SPI (Serial Peripheral Interface) ได้ทั้งการเป็นมาสเตอรและสเลฟ (Master/Slave)

8) มีการเชื่อมประสานแบบอนุกรมด้วยสายสัญญาณ 2 เส้น (I^2C) แบบส่งข้อมูลแบบเรียงไบต์ (Byte Oriented)

9) มีตัวตั้งเวลาแบบวอตช์ด็อกที่สามารถกำหนดการทำงานได้โดยสามารถแยกสัญญาณนาฬิกาได้จากตัวชิพ

10) มีตัวเปรียบเทียบสัญญาณแบบอนาล็อกอยู่ในตัว

11) มีการรองรับการขัดจังหวะและการเวก-อัพ (Wake-up) เมื่อมีการเปลี่ยนแปลงเกิดขึ้นกับขาของชิพ

2.1.1.7 คุณสมบัติพิเศษ

1) มีระบบเริ่มระบบเมื่อมีการรีเซ็ตและมีระบบตรวจจับการเกิดบราวน์เอาท์ (Brown-out) ที่สามารถกำหนดการทำงานได้

2) มีตัวตรวจหาความเที่ยงตรงของออสซิเลเตอร์อยู่ในตัว (Internal Calibrated Oscillator)

3) มีแหล่งการขัดจังหวะทั้งภายในและภายนอก (External and Internal Interrupt Sources)

4) มีโหนดการทำงานสลีป 6 แบบ คือ Idle, ADC Noise Reduction, Power save, Power-Down, Standby, และ Extended Standby

2.1.1.8 ขาอินพุต/เอาท์พุต และตัวถัง

1) มีขาของอินพุต/เอาท์พุต ที่สามารถกำหนดการทำงานได้ 23 ขา

2) ตัวถังแบบ PDIP ชนิด 28 ขา

2.1.1.9 ช่วงอุณหภูมิที่ชิพทำงานได้ -40°C ถึง 85°C

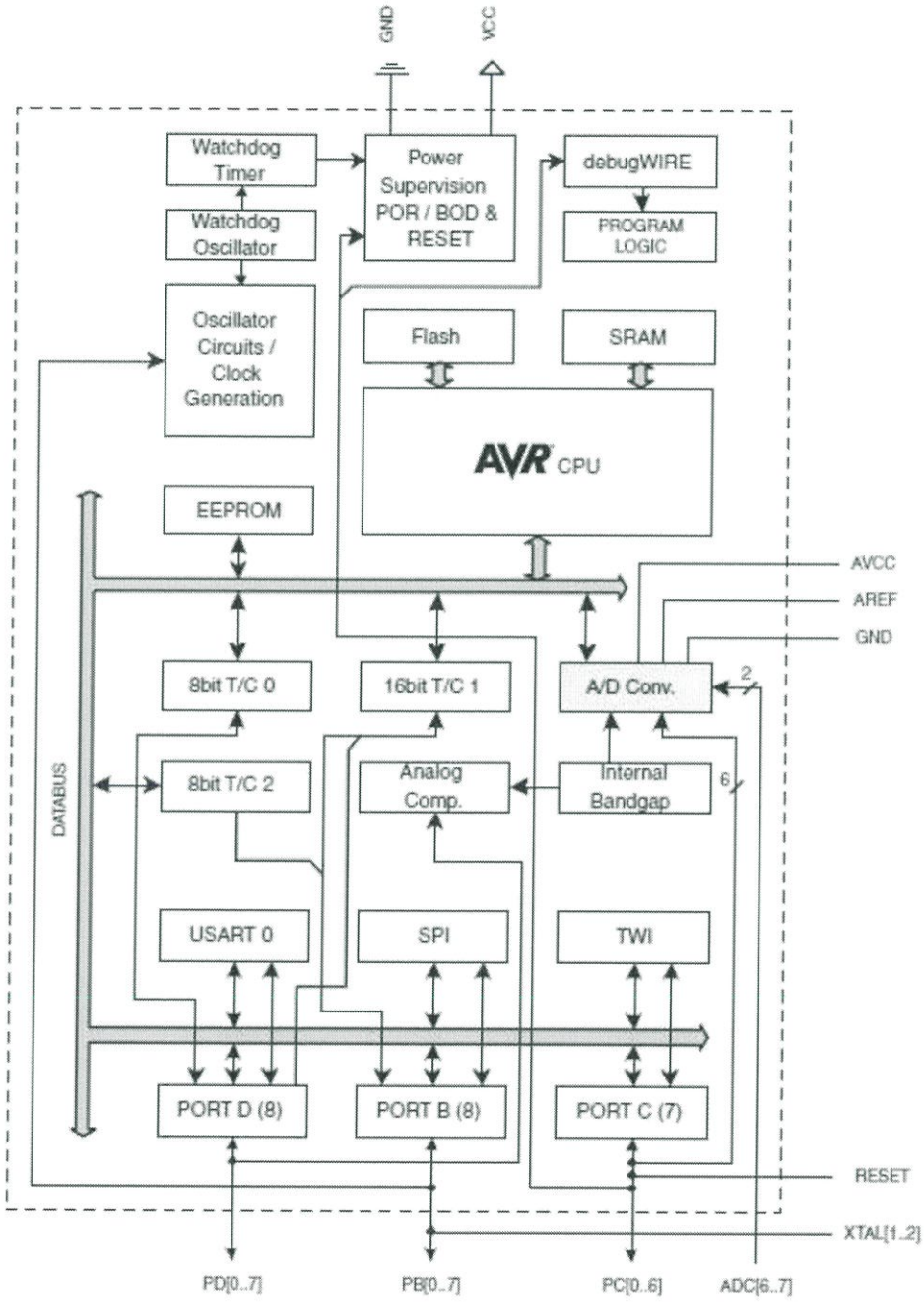
2.1.1.10 การใช้พลังงานที่ 1 MHz แรงดัน 1.8V ที่อุณหภูมิ 25°C

1) โหมดแอกทีฟ (Active Mode) ต้องการกระแส 0.2 mA

2) โหมดเพาเวอร์ดาวน์ (Power-down) ต้องการกระแส 0.1 μA

3) โหมดเพาเวอร์เซฟ (Power-save Mode) ต้องการกระแส 0.75 μA

สถาปัตยกรรมของชิพ



รูปที่ 2.1 บล็อกไดอะแกรม ATmega328 [1]

2.1.2 ขาพอร์ตอินพุตเอาท์พุต

จากรูปที่ 2.1 มีรายละเอียดแต่ละพอร์ตดังนี้

2.1.2.1 VCC ขาแรงดันไฟตรง

2.1.2.2 GND ขากราวด์

2.1.2.3 Port B (PB0...PB7) พอร์ต B เป็นอินพุต/เอาท์พุตพอร์ตแบบทำงาน 2 ทิศทาง (bi-directional I/O port) ขนาด 8 บิต ที่มีวงจรพูลอัพอยู่ภายใน (เลือกได้เป็นรายบิต) สามารถทางานเป็นเอาต์พุตไดทั้งแบบซิงค์ และซอร์สจุดเด่นของพอร์ตนี้คือเป็นพอร์ตที่มีความสามารถในการขับกระแสได้ดีกว่าพอร์ตอื่นๆ หน้าที่ของขาสัญญาณของพอร์ต B (PB0-PB7) ดังตารางที่ 2.1

2.1.2.4 Port C (PC0...PC5) พอร์ต C เป็นอินพุต/เอาท์พุตพอร์ตแบบทำงาน 2 ทิศทาง (bi-directional I/O port) ขนาด 8 บิตที่มีวงจรพูลอัพอยู่ภายใน (เลือกได้เป็นรายบิต) จึงสามารถทำงานเป็นเอาต์พุตได้ทั้งแบบซิงค์และซอร์ หน้าที่ของสัญญาณพอร์ต C (PC0-PC6) ดังตารางที่ 2.2

2.1.2.5 PC6/ $\overline{\text{RESET}}$ ขารีเซ็ทวงจร

2.1.2.6 Port D (PD0...PD7) พอร์ต D เป็นอินพุต/เอาท์พุตพอร์ตแบบทำงาน 2 ทิศทาง (bi-directional I/O port) ขนาด 8 บิตที่มีวงจรพูลอัพอยู่ภายใน (เลือกได้เป็นรายบิต) จึงสามารถทำงานเป็นเอาต์พุตได้ทั้งแบบซิงค์และซอร์ หน้าที่ของสัญญาณพอร์ต D (PD0-PD7) ดังตารางที่ 2.3

2.1.2.7 AVCC ขาแรงดันสำหรับพอร์ต C และโมดูลแปลงสัญญาณอนาล็อกเป็นดิจิตอล

2.1.2.8 AREF ขาแรงดันอนาล็อกอ้างอิงสำหรับโมดูลแปลงสัญญาณอนาล็อกเป็นดิจิตอล

ตารางที่ 2.1 หน้าที่ของขาสัญญาณของพอร์ต B (PB0-PB7)

ขาพอร์ต	ขาพอร์ตฟังก์ชั่นพิเศษ
PB7	XTAL2 (ขาต่อคริสตอลออสซิเลเตอร์ช่องที 2) TOSC2 (ขา RTC ออสซิเลเตอร์ 2 สำหรับไทเมอร์/เคาน์เตอร์ 0) PCINT7 (ขาอินพุตอินเตอร์รัปเนื่องจากสัญญาณภายนอกช่องที่ 7)
PB6	XTAL1 (ขาต่อคริสตอลออสซิเลเตอร์ช่องที 1) TOSC1 (ขา RTC ออสซิเลเตอร์ 1 สำหรับไทเมอร์/เคาน์เตอร์ 0) PCINT6 (ขาอินพุตอินเตอร์รัปเนื่องจากสัญญาณภายนอกช่องที่ 6)
PB5	SCK (ขาสัญญาณนาฬิกาสำหรับระบบของระบบบัส SPI) PCINT5 (ขาอินพุตอินเตอร์รัปเนื่องจากสัญญาณภายนอกช่องที่ 5)
PB4	MISO (ขาสัญญาณอินพุตมาสเตอร์/ขาสัญญาณเอาท์พุตสเลฟ สำหรับบัส SPI) PCINT4 (ขาอินพุตอินเตอร์รัปเนื่องจากสัญญาณภายนอกช่องที่ 4)
PB3	MISI (ขาสัญญาณเอาท์พุตมาสเตอร์/ขาสัญญาณอินพุตสเลฟ สำหรับบัส SPI) PCINT3 (ขาอินพุตอินเตอร์รัปเนื่องจากสัญญาณภายนอกช่องที่ 3)
PB2	$\overline{SS}$ (ขาสัญญาณอินพุตเลือกสเลฟ สำหรับบัส SPI) OC1B (โมดูลเปรียบเทียบสัญญาณเอาท์พุต B กับไทเมอร์/เคาน์เตอร์ 1) PCINT2 (ขาอินพุตอินเตอร์รัปเนื่องจากสัญญาณภายนอกช่องที่ 2)
PB1	OC1A (โมดูลเปรียบเทียบสัญญาณเอาท์พุต A กับไทเมอร์/เคาน์เตอร์ 1) PCINT1 (ขาอินพุตอินเตอร์รัปเนื่องจากสัญญาณภายนอกช่องที่ 1)
PB0	ICP1 (โมดูลอินพุตตรวจจับสัญญาณ กับไทเมอร์/เคาน์เตอร์ 1) CLKO (เอาต์พุตสัญญาณนาฬิกาสำหรับระบบการหาร) PCINT0 (ขาอินพุตอินเตอร์รัปเนื่องจากสัญญาณภายนอกช่องที่ 0)

ตารางที่ 2.2 หน้าที่ของสัญญาณพอร์ต C (PC0-PC6)

ขาพอร์ต	ขาพอร์ตฟังก์ชั่นพิเศษ
PC6	$\overline{\text{RESET}}$ (ขารีเซ็ตวงจร) PCINT14 (ขาอินพุตอินเตอร์รัปเนื่องจากสัญญาณภายนอกช่องที่ 14)
PC5	ADC5 (ขาพอร์ตอินพุตสัญญาณอนาล็อกช่องที่ 5) SCL (ขาสัญญาณนาฬิกาสำหรับบัสข้อมูลอนุกรม 2 สาย หรือ I^2C) PCINT13 (ขาอินพุตอินเตอร์รัปเนื่องจากสัญญาณภายนอกช่องที่ 13)
PC4	ADC4 (ขาพอร์ตอินพุตสัญญาณอนาล็อกช่องที่ 4) SDA (ขาสัญญาณข้อมูลอินพุตอินพุต/เอาท์พุตสำหรับบัสข้อมูลอนุกรม 2 สาย หรือ I^2C) PCINT12 (ขาอินพุตอินเตอร์รัปเนื่องจากสัญญาณภายนอกช่องที่ 12)
PC3	ADC3 (ขาพอร์ตอินพุตสัญญาณอนาล็อกช่องที่ 3) PCINT11 (ขาอินพุตอินเตอร์รัปเนื่องจากสัญญาณภายนอกช่องที่ 11)
PC2	ADC2 (ขาพอร์ตอินพุตสัญญาณอนาล็อกช่องที่ 2) PCINT10 (ขาอินพุตอินเตอร์รัปเนื่องจากสัญญาณภายนอกช่องที่ 10)
PC1	ADC1 (ขาพอร์ตอินพุตสัญญาณอนาล็อกช่องที่ 1) PCINT9 (ขาอินพุตอินเตอร์รัปเนื่องจากสัญญาณภายนอกช่องที่ 9)
PC0	ADC0 (ขาพอร์ตอินพุตสัญญาณอนาล็อกช่องที่ 0) PCINT8 (ขาอินพุตอินเตอร์รัปเนื่องจากสัญญาณภายนอกช่องที่ 8)

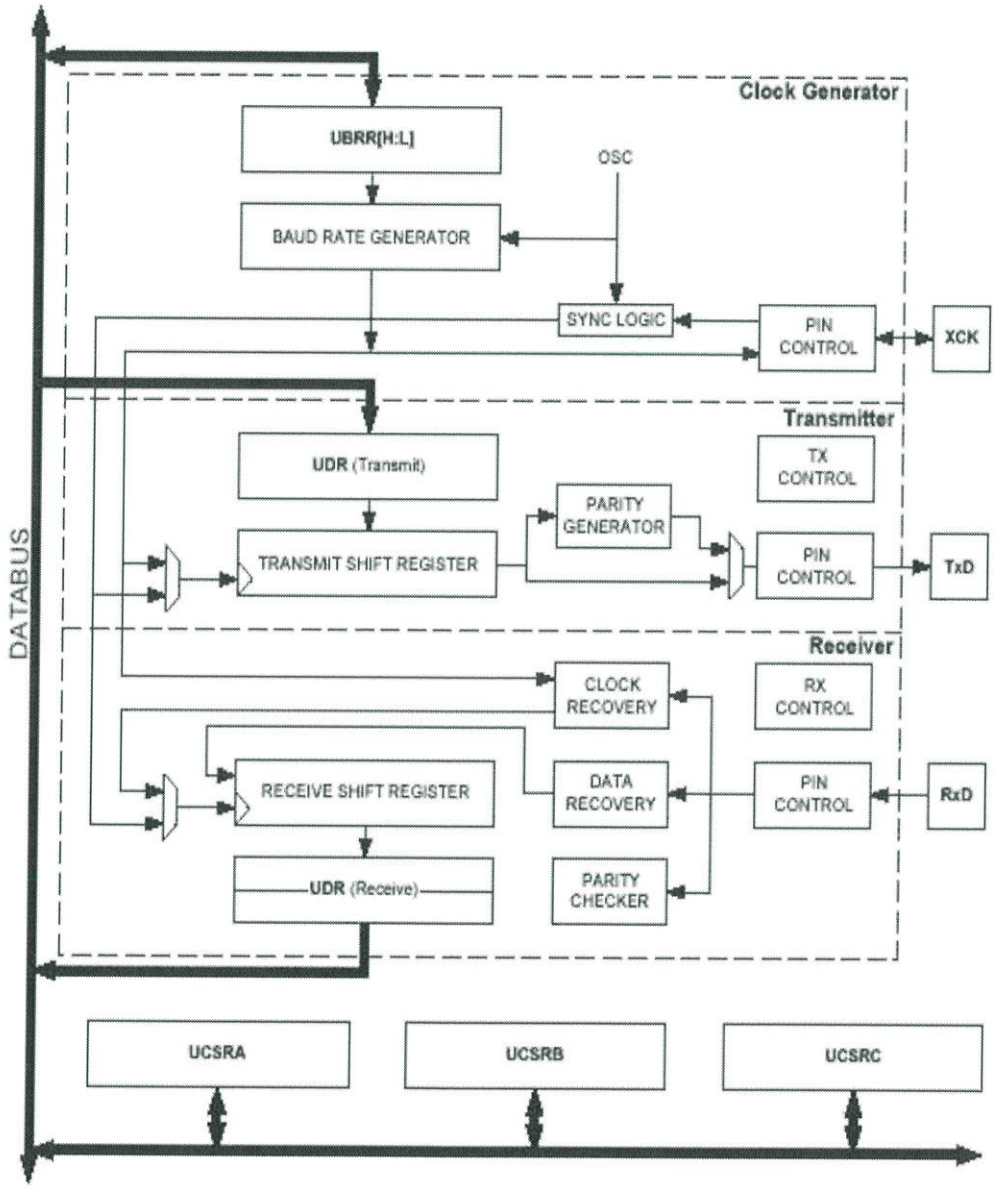
ตารางที่ 2.3 หน้าที่ของสัญญาณพอร์ต D (PD0-PD7)

ขาพอร์ต	ขาพอร์ตฟังก์ชั่นพิเศษ
PD7	AIN1 (อินพุตสัญญาณด้านลบสำหรับโมดูลเปรียบเทียบสัญญาณอนาล็อกอินพุต) PCINT23 (ขาอินพุตอินเตอร์รัปเนื่องจากสัญญาณภายนอกช่องที่ 23)
PD6	AIN0 (อินพุตสัญญาณด้านบวกสำหรับโมดูลเปรียบเทียบสัญญาณอนาล็อกอินพุต) OC0A (โมดูลเปรียบเทียบสัญญาณเอาท์พุต A กับไทเมอร์/เคาน์เตอร์ 0) PCINT22 (ขาอินพุตอินเตอร์รัปเนื่องจากสัญญาณภายนอกช่องที่ 22)
PD5	T1 (ไทเมอร์/เคาน์เตอร์ 1 และอินพุตรับสัญญาณจากภายนอกในโหมดเคาน์เตอร์ 1) OC0B (โมดูลเปรียบเทียบสัญญาณเอาท์พุต B กับไทเมอร์/เคาน์เตอร์ 0) PCINT21 (ขาอินพุตอินเตอร์รัปเนื่องจากสัญญาณภายนอกช่องที่ 21)
PD4	XCK (อินพุต/เอาต์พุตสัญญาณนาฬิกาจากภายนอกสำหรับโมดูล USART) T0 (ไทเมอร์/เคาน์เตอร์ 0 และอินพุตรับสัญญาณจากภายนอกในโหมดเคาน์เตอร์ 0) PCINT20 (ขาอินพุตอินเตอร์รัปเนื่องจากสัญญาณภายนอกช่องที่ 20)
PD3	INT1 (อินเตอร์รัปต์เนื่องจากสัญญาณภายนอกช่องที่ 1) OC2B (โมดูลเปรียบเทียบสัญญาณเอาท์พุต B กับไทเมอร์/เคาน์เตอร์ 2) PCINT19 (ขาอินพุตอินเตอร์รัปเนื่องจากสัญญาณภายนอกช่องที่ 19)
PD2	INT0 (อินเตอร์รัปต์เนื่องจากสัญญาณภายนอกช่องที่ 0) PCINT18 (ขาอินพุตอินเตอร์รัปเนื่องจากสัญญาณภายนอกช่องที่ 18)
PD1	TXD (ขาเอาต์พุตสัญญาณสำหรับโมดูล USART) PCINT17 (ขาอินพุตอินเตอร์รัปเนื่องจากสัญญาณภายนอกช่องที่ 17)
PD0	RXD (ขาอินพุตสัญญาณสำหรับโมดูล USART) PCINT16 (ขาอินพุตอินเตอร์รัปเนื่องจากสัญญาณภายนอกช่องที่ 16)

2.1.3 การสื่อสารข้อมูลอนุกรมผ่านโมดูล USART

ไมโครคอนโทรลเลอร์ AVR สามารถจะสื่อสารข้อมูลแบบอนุกรมได้โดยการใช้โมดูล USART (Universal Synchronous and Asynchronous serial Receiver and Transmitter) เพื่อสื่อสารข้อมูลผ่านทางพอร์ตอนุกรมได้ทั้งแบบซิงโครนัส (ข้อมูลมีความต่อเนื่อง มีการกำหนดสัญญาณมาตรฐานที่เหมือนกันทั้งทางด้านรับและด้านส่ง เพื่อให้การรับส่งมีความสัมพันธ์กัน) และอะซิงโครนัส (ข้อมูลไม่จำเป็นต้องต่อเนื่องมีบิตเริ่มต้น (Start bit) บิตข้อมูล (Data bit) และบิตหยุด (Stop bit) มีบิตพาริตี้ (Parity Bit) หรือไม่มีก็ได้) สำไมโครคอนโทรลเลอร์

ATmega328 ขาพอร์ตอนุกรมกำหนดไว้ที่ขาพอร์ต PD0 (RXD) ใช้ในการรับข้อมูลอนุกรมและขาพอร์ต PD1 (TXD) ใช้ในการส่งข้อมูลอนุกรม บล็อกไดอะแกรมของข้อมูล USART แสดงดังรูปที่ 3



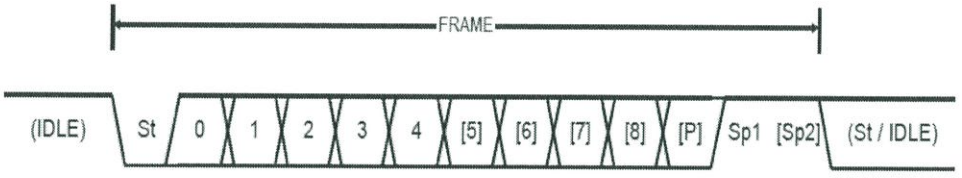
รูปที่ 2.2 บล็อกไดแกรมโมดูล USART [1]

จากรูปที่ 2.3 จะเห็นว่าโมดูล USART แบ่งออกเป็น 3 ส่วนด้วยกันคือ

- ส่วนสร้างสัญญาณนาฬิกา (Clock Generator) เพื่อใช้กำหนดอัตราบอตในการรับส่งข้อมูล โดยสามารถกำหนดได้ทั้งภายในและภายนอก ผ่านางขา XCK (Transfer Clock)
- ส่วนส่งข้อมูลอนุกรม (Transmitter) โดยส่งข้อมูลออกทางพอร์ต TxD
- ส่วนรับข้อมูลอนุกรม (Receiver) โดยการรับข้อมูลจากขาพอร์ต RxD

และมีรีจิสเตอร์ควบคุมการทำงาน 3 ประการประกอบไปด้วย UCSRA, UCSRB และ UCSRC การส่งข้อมูลอนุกรมในรูปแบบอะซิงโครนัส จะเป็นการส่งข้อมูลเป็นเฟรม ลักษณะของเฟรมข้อมูลอนุกรมนี้ประกอบด้วยแสดงดังรูปที่ 2.4

1) บิตเริ่มต้นข้อมูล (Start bit)
2) บิตข้อมูล (Data bit)
3) พาริตี้บิต (Parity bit)
4) บิตหยุดข้อมูล (Stop bit)



รูปที่ 2.3 รูปแบบเฟรมข้อมูลอนุกรม [1]

St Start bit, always low.

(n) Data bits (0 to 8).

P Parity bit. Can be odd or even.

Sp Stop bit, always high.

IDLE No transfers on the communication line (RxDn or TxDn). An IDLE line must be high.

2.1.3.1 คุณสมบัติที่สำคัญของโมดูล USART มีดังนี้

1) การสื่อสารข้อมูลแบบฟลูดูเพล็กซ์ (Full Duplex) ตัวรับและตัวส่งแยกอิสระต่อกันสามารถรับและส่งข้อมูลในเวลาเดียวกัน (พร้อมกัน)
2) ทำงานได้ทั้งในโหมดซิงโครนัสและอะซิงโครนัส
3) มีคุณสมบัติของพอร์ตอนุกรมครบถ้วน เช่น การกำหนดบิตข้อมูล การกำหนดบิตหยุดและการกำหนดบิตพาริตี้ เป็นต้น
4) มีส่วนตรวจสอบความผิดพลาดของเฟรมข้อมูลและข้อมูลโอเวอร์รัน (Framing Error and Data OverRun Detection)
5) โหมดการสื่อสารแบบมัลติโปรเซสเซอร์
6) โหมดทวีคูณความเร็วในการสื่อสารข้อมูลแบบซิงโครนัส

การสื่อสารข้อมูลอนุกรมโหมดอะซิงโครนัสเป็นการสื่อสารข้อมูลระหว่างไมโครคอนโทรลเลอร์กับพอร์ตอนุกรมหรือที่นิยมนำไมโครคอนโทรลเลอร์มาเชื่อมต่อกับคอมพิวเตอร์ผ่านทางพอร์ตอนุกรม (RS-232) นอกจากการกำหนดจำนวนบิตข้อมูล ยิตหยุดและพาริตี้บิตแล้ว จะต้องมีการกำหนดอัตราเร็วในการรับส่งข้อมูลที่เรียกว่า อัตราบอตหรือบอตเรต (baud rate) หรือการเปลี่ยนแปลงของสัญญาณใน 1 วินาที ซึ่งคำนวณหาได้ดังตารางที่ 2.4

ตารางที่ 2.4 การคำนวณอัตราบอตเรต

โหมดการทำงาน	การคำนวณหาอัตราบอตเรท	การคำนวณหาค่า UBRR
โหมดอะซิงโครนัสปกติ (U2X=0)	BAUD=fosc /16(UBRR+1)	BAUD=(fosc /16BAUD)-1
โหมดอะซิงโครนัสทวีคูณ (U2X=1)	BAUD=fosc /8(UBRR+1)	BAUD=(fosc /8BAUD)-1
โหมดมาสเตอร์ซิงโครนัส	BAUD=fosc /2(UBRR+1)	BAUD=(fosc /2BAUD)-1

โดย

BAUD : อัตราบอตเรทในหน่วยบิตต่อวินาที (bps)

Fose : ความถี่สัญญาณนาฬิกาหลักของระบบ

UBRR : รีจิสเตอร์ UBRRH และ UBRRL (0-4095)

การทำงานของโมดูล USART กับสัญญาณนาฬิกา หากเป็นการใช้งานนาฬิกาภายใน (Internal Clock) หรือสัญญาณนาฬิกาหลัก (fose) จะใช้งานกับอะซิงโครนัสโหมดและมาสเตอร์ซิงโครนัส และสัญญาณนาฬิกาภายนอก (External Clock) ใช้งานในโหมดสเลฟซิงโครนัส โดยความถี่สัญญาณนาฬิกาภายนอกจะต้องน้อยกว่าความถี่สัญญาณนาฬิกาหลักหารด้วย 4 (f_{XCK}<fosc/4)

2.1.3.2 รีจิตเตอร์ที่เกี่ยวข้องกับการทำงานในโหมด USART

1) รีจิสเตอร์ UDR (USART I/O Data Register)

รีจิตเตอร์อ่านเขียนข้อมูลขนาด 8 บิตโดยแบ่งออกเป็น 2 ตัว คือ RXB ใช้รับข้อมูลจากภายนอกเข้ามาในไมโครคอนโทรลเลอร์และ TXB ใช้สำหรับส่งข้อมูลออกจากไมโครคอนโทรลเลอร์ การอ่านเขียนข้อมูลจะทำกับรีจิสเตอร์ UDR โดยตรง

2) รีจิสเตอร์ UCSRA (USART Control and status Register A)

รีจิสเตอร์ควบคุมการทำงานและแสดงสถานการณ์ทำงานของโมดูล USART ชุด A เกี่ยวข้องกับสถานะของการสื่อสารข้อมูล

3) รีจิสเตอร์ UCSRB (USART Control and status Register B)

รีจิสเตอร์ควบคุมการทำงานและแสดงสถานการณ์ทำงานของโมดูล USART ชุด B เกี่ยวข้องกับบิตกำหนดอินเตอร์รัปต์และการกำหนดขนาดของข้อมูลแบบ 9 บิตข้อมูล (Data bit)

4) รีจิสเตอร์ UCSRC (USART Control and status Register C)

รีจิสเตอร์ควบคุมการทำงานและแสดงสถานการณ์ทำงานของโมดูล USART ชุด C เกี่ยวข้องกับการกำหนดอัตราบอตในการรับส่งข้อมูล

2.1.4 การใช้งานโมดูล USART โหมดอะซิงโครนัส

การใช้งานโมดูล USART ในโหมดอะซิงโครนัส หรือการรับส่งข้อมูลในรูปแบบมาตรฐาน RS-232 (Comport) จะต้องมีการกำหนดค่าในรีจิสเตอร์ต่อไปนี้

2.1.4.1 รีจิสเตอร์ UDR (USART I/O Data Register)

รีจิสเตอร์ UDR ประกอบไปด้วยบิตที่ 0-7 ดังต่อไปนี้และแสดงดังตารางที่ 2.5

ตารางที่ 2.5 รีจิสเตอร์ UDR (USART I/O Data Register)

บิตที่	7	6	5	4	3	2	1	0
(Read)	RXB[7:0]							
(Write)	TXB[7:0]							
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W
ค่าเริ่มต้น	0	0	0	0	0	0	0	0

รีจิสเตอร์รับส่งข้อมูล เมื่อมีการเขียนข้อมูลไปที่รีจิสเตอร์ UDR ข้อมูลจะถูกเขียนไปที่ TXB (Transmit Data Buffer Register) และเมื่อมีการอ่านข้อมูลจากรีจิสเตอร์ UDR จะไปอ่านข้อมูลที่ เก็บอยู่ที่ RXB (Receive Data Buffer Register)

2.1.4.2 รีจิสเตอร์ UCSRA (USART Control and status Register A)

รีจิสเตอร์ UCSRA ประกอบไปด้วยบิตที่ 0-7 แสดงดังตารางที่ 2.6

ตารางที่ 2.6 รีจิสเตอร์ UCSRA (USART Control and status Register A)

บิตที่	7	6	5	4	3	2	1	0
ชื่อบิต	RXC	TXC	UDRE	FE	DOR	PE	U2X	MPCM
Read/Write	R	R/W	R	R	R	R	R/W	R/W
ค่าเริ่มต้น	0	0	0	0	0	0	0	0

1) บิตที่ 7: บิต RXC (USAART Receive Complete) บิตที่ RXC ถูกเซ็ตเมื่อยังไม่ได้อ่านข้อมูลในบัฟเฟอร์รับข้อมูลและเคลียร์ เมื่อบัฟเฟอร์ว่าง (จะไม่สามารถเก็บข้อมูลใหม่ได้หากข้อมูลยังไม่ได้อ่านออกไป)

2) บิตที่ 6: บิต TXC (USART Transmit Complete) บิต TXC ถูกเซ็ตเมื่อข้อมูลในบัฟเฟอร์ส่งข้อมูลได้ถูกส่งออกไปแล้วและไม่มีข้อมูลในบัฟเฟอร์ส่งข้อมูล (UDR) บิต TXC จะเคลียร์โดยอัตโนมัติเมื่อการส่งข้อมูลในอินเตอร์รัปต์ส่วนส่งข้อมูลทำงานเสร็จสมบูรณ์ หรือโดยการเขียนค่าไปที่บิตโดยตรง

3) บิตที่ 5: บิต UDRE (USART Data Register Empty) หากบิต UDRE ถูกเซ็ตเป็นหนึ่งแสดงว่า รีจิตเตอร์ UDR ว่างพร้อมรับข้อมูลใหม่แล้ว (บัฟเฟอร์ข้อมูลว่าง)

4) บิตที่ 4: บิต FE (Frame Error) บิต FE จะถูกเซ็ตเมื่อเฟรมข้อมูลผิดพลาดหรือบิตหยุดข้อมูลเป็นศูนย์ และบิต FE จะถูกเคลียร์เมื่อบิตข้อมูลจะเป็นหนึ่ง

5) บิตที่ 3: บิต DOR (Data OverRun) บิต DOR ถูกเซ็ตเมื่อเกิดเงื่อนไขข้อมูล Over Run หรือบัฟเฟอร์รับข้อมูลเต็มและมีข้อมูลรอเข้ามาในบัฟเฟอร์รับข้อมูล

6) บิตที่ 2: บิต PE (Parity Error) บิต PE ถูกเซ็ตเมื่อเกิดข้อผิดพลาดของบิตพาริตี้

7) บิตที่ 1: บิต U2X (Double the USART Transmission Speed) บิต U2X แสดงการอัตราการทวีคูณของการสื่อสารข้อมูล มีผลกับโหมดอะซิงโครนัส (เคลียร์บิต U2X เป็นศูนย์เมื่อใช้งานในโหมดซิงโครนัส)

8) บิตที่ 0: บิต MPCM (Multi-processor Communication Mode) เซตบิต MPCM เมื่อต้องการใช้งานการสื่อสารข้อมูลแบบมัลติโปรเซสเซอร์

2.1.4.3 รีจิสเตอร์ UCSRB (USART Control and status Register B)

รีจิสเตอร์ UCSRB ประกอบไปด้วยบิตที่ 0-7 แสดงดังตารางที่ 2.7

ตารางที่ 2.7 รีจิสเตอร์ UCSRB (USART Control and status Register B)

บิตที่	7	6	5	4	3	2	1	0
ชื่อบิต	RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCSZ2	RXB8	TXB8
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W
ค่าเริ่มต้น	0	0	0	0	0	0	0	0

1) บิตที่ 7: บิต RXCIE (RX Complete Interrupt Enable) เซตบิต RXCIE เป็นหนึ่งเพื่อเปิดใช้งานอินเตอร์รัปต์ที่เกี่ยวข้องกับแฟลก RCX (บิตที่ 7 ใน รีจิสเตอร์ UCSRA) เมื่อมีการรับข้อมูลในอินเตอร์รัปต์เสร็จสมบูรณ์

2) บิตที่ 6: บิต TXCIE (TX Complete Interrupt Enable) เซตบิต TXCIE เป็นหนึ่งเพื่อเปิดใช้งานอินเตอร์รัปต์ที่เกี่ยวข้องกับแฟลก TCX (บิตที่ 6 ใน รีจิสเตอร์ UCSRA) เมื่อมีการรับข้อมูลในอินเตอร์รัปต์เสร็จสมบูรณ์

3) บิตที่ 5: บิต UDRIE (USART Data register Empty Interrupt Enable) เซ็ตบิต UDRIE เป็นหนึ่งเพื่อเปิดใช้งานอินเตอร์รัปต์ที่เกี่ยวข้องกับแฟลก UDRE (บิตที่ 5 ในรีจิสเตอร์ UCSRA) เมื่อรีจิสเตอร์ UDR ว่าง

4) บิตที่ 4: บิต RXEN (Receiver Enable) เซตบิต RXEN เป็นหนึ่งเมื่อต้องการใช้งานอินเตอร์รัปต์ เนื่องจากการรับข้อมูล

5) บิตที่ 3: บิต TXEN (Transmitter Enable) เซตบิต TXEN เป็นหนึ่งเมื่อต้องการใช้งานอินเตอร์รัปต์ เนื่องจากการส่งข้อมูล

6) บิตที่ 2: บิต UCSZ2 (Character Size) บิต UCSZ2 ทำงานร่วมกับบิต UCSZ1 (ในรีจิสเตอร์ UCSRC) ในการกำหนดจำนวนข้อมูล (Data Bit) ในการรับส่ง

7) บิตที่ 1: บิต RXB8 (Receive Data Bit 8) บิต RXB8 บิตที่เก้าของจำนวนข้อมูลรับ เมื่อทำงานในโหมดเก้าบิตข้อมูล (ต้องอ่านก่อนที่จะอ่านบิตต่ำสุดของข้อมูลในรีจิสเตอร์ UDR)

8) บิตที่ 0: บิต TXB8 (Transmit Data Bit 8) บิต TXB8 บิตที่เก้าของจำนวนข้อมูลส่ง เมื่อทำงานในโหมดเก้าบิตขข้อมูล (ต้องอ่านก่อนที่จะอ่านบิตต่ำสุดของข้อมูลในรีจิสเตอร์ UDR)

2.1.4.4 รีจิสเตอร์ UCSRC (USART Control and status Register C)

รีจิสเตอร์ UCSRC ประกอบไปด้วยบิตที่ 0-7 แสดงดังตารางที่ 2.8

ตารางที่ 2.8 รีจิสเตอร์ UCSRC (USART Control and status Register C)

บิตที่	7	6	5	4	3	2	1	0
ชื่อบิต	URSEL	UMSEL	UPM1	UPM0	USBS	UCSZ1	UCSZ0	UCPOL
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W
ค่าเริ่มต้น	1	0	0	0	0	1	1	0

1) บิตที่ 7: บิต URSEL (Register Select) บิต URSEL เลือกการเข้าถึงรีจิสเตอร์ UCSRC หรือ UBRRH เซตบิต URSEL เป็นหนึ่งเมื่อเขียนข้อมูลไปที่รีจิสเตอร์ UCSRC

2) บิตที่ 6: บิต UMSEL (USART Mode Select) บิต UMSEL เลือกโหมดการทำงานระหว่างอะซิงโครนัสโหมด (UMSEL=1) กับซิงโครนัสโหมด (UMSEL=0)

3) บิตที่ 5:4: บิต UPM1:0 (Parity Mode) บิต UPM กำหนดพาริตี้บิต การเซตบิต กำหนดพาริตี้บิตแสดงดังตารางที่ 2.9

ตารางที่ 2.9 ตารางแสดงการเซตบิต กำหนดพาริตี้บิต

UPM1	UPM2	พาริตี้บิต
0	0	ไม่ใช้งานพาริตี้
0	1	สงวนไว้
1	0	พาริตี้คู่ (Even Parity)
1	1	พาริตี้คี่ (Odd Parity)

4) บิตที่ 3: บิต USBS (Stop Bit Select) บิต USBS บิตกำหนดจำนวนบิตหยุด (Stop Bit) โดย USBS=0 บิตหยุดข้อมูล 1 บิต, USBS=1 บิตหยุดข้อมูล 2 บิต

5) บิตที่ 2:1: บิต UCSZ1:0 (Character Size) บิตกำหนดจำนวนบิตข้อมูลรับส่ง การเซตบิต กำหนดจำนวนบิตข้อมูล ดังตารางที่ 2.10

ตารางที่ 2.10 ตารางแสดงการเซตบิต กำหนดจำนวนบิตข้อมูล

UCSZ2	UCSZ1	UCSZ0	จำนวนบิตข้อมูล (บิต)
0	0	0	5
0	0	1	6
0	1	0	7
0	1	1	8
1	0	0	สงวนไว้
1	0	1	สงวนไว้
1	1	0	สงวนไว้
1	1	1	9

6) บิตที่ 0: บิต UCPOL (Clock Polarity) เซตบิต UCPOL เมื่อมีการใช้งานโหมดซิงโครนัส จะสัมพันธ์กับการเปลี่ยนแปลงตัวอย่างข้อมูลเอาต์พุต ข้อมูลอินพุต และสัญญาณนาฬิกาซิงโครนัส (XCX) การเซตบิต กำหนดขอบขาสัญญาณ XCX แสดงดังตารางที่ 2.11

ตารางที่ 2.11 ตารางแสดงการเซตบิต กำหนดขอบขาสัญญาณ XCX

UCPOL	การส่งข้อมูล (ขา TxD)	การรับข้อมูล (ขา RxD)
0	ขอบขาขึ้น ของ XCX	ขอบขาลง ของ XCX
1	ขอบขาลง ของ XCX	ขอบขาขึ้น ของ XCX

2.1.4.5 รีจิสเตอร์ UBRRL และ UBRRH (USART Baud Rate Register)

รีจิสเตอร์ UBRRL ประกอบไปด้วยบิตที่ 0-15 แสดงดัง

ตารางที่2.12

ตารางที่ 2.12 รีจิสเตอร์ UBRRL และ UBRRH (USART Baud Rate Register)

บิตที่	15	14	13	12	11	10	9	8
ชื่อบิต	URSEL	-	-	-	UBRR[11:8]			
	UBRR[7:0]							
	7	6	5	4	3	2	1	0
Read/Write	R/W R/W	R R/W	R/W R/W	R R/W	R/W R/W	R/W R/W	R/W R/W	R/W R/W
ค่าเริ่มต้น	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0

1) บิตที่ 15: บิต URSEL (Register Select) บิต URSEL เลือกการเข้าถึงรีจิสเตอร์ UBRRH หรือ UCSRC เคลียร์บิต URSEL เป็นศูนย์เมื่อเขียนข้อมูลไปที่รีจิสเตอร์ UBRRH

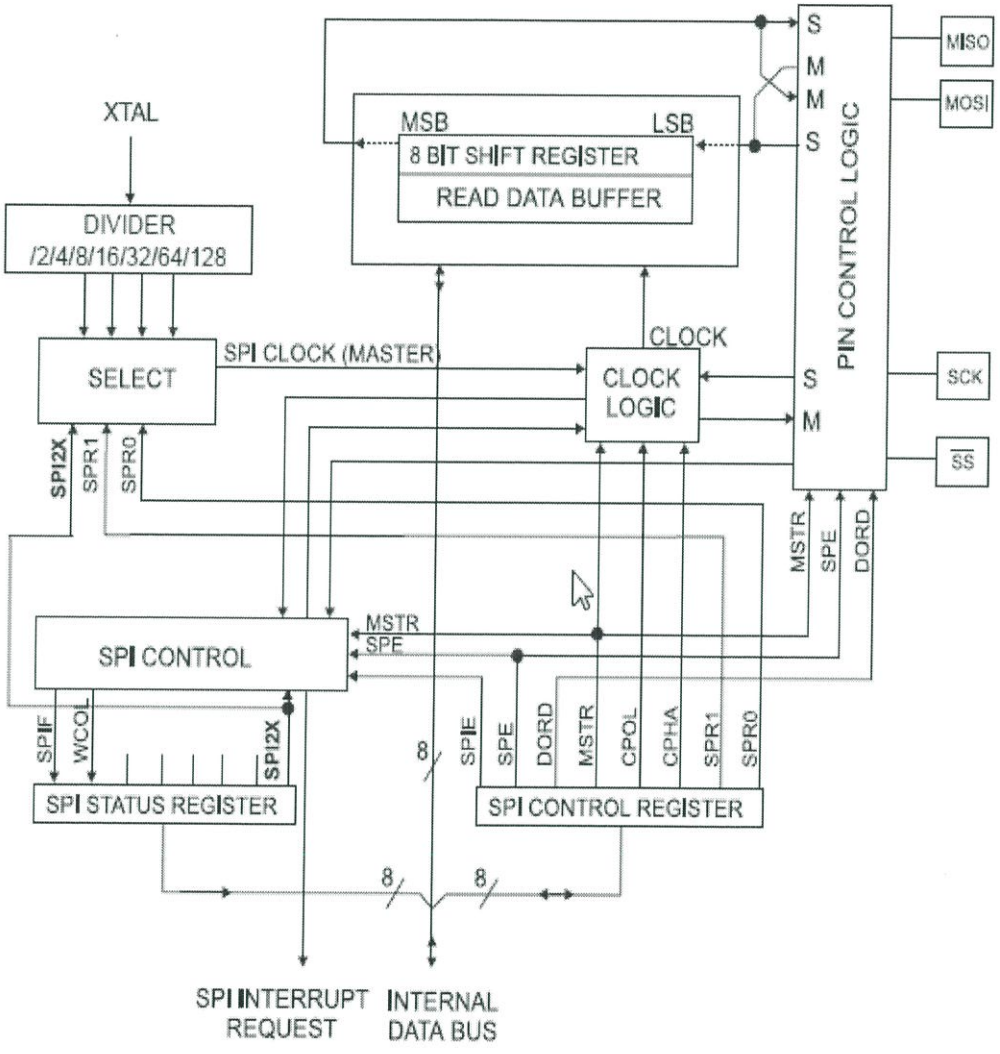
2) บิตที่ 14:12: Reserved Bits บิตที่ 12 ถึง 14 สงวนการใช้งาน

3) บิตที่ 11:0 บิต UBRR11:0 (USART Baud Rate Register) บิต UBRR ใช้ในการกำหนดอัตราบอดในการรับส่งข้อมูลอนุกรม คำนวณหาค่า UBRR ได้

2.1.5 การเชื่อมต่ออุปกรณ์อนุกรม (SPI)

การเชื่อมต่ออุปกรณ์อนุกรม (Serial Peripheral Interface) หรือ SPI เป็นการเชื่อมต่อกับอุปกรณ์เพื่อรับส่งข้อมูลแบบซิงโครนัส (มีสัญญานาฬิกาเข้ามาเกี่ยวข้อง) ระหว่าง

ไมโครคอนโทรลเลอร์ AVR หรือจะเป็นอุปกรณ์ภายนอกที่มีการรับส่งข้อมูลแบบ SPI โดยโครงสร้างบล็อกไดอะแกรมการรับส่งข้อมูลแบบ SPI แสดงดังรูปที่ 2.5



รูปที่ 2.4 บล็อกไดอะแกรม SPI [1]

จากรูปที่ 2.5 การสื่อสารข้อมูล SPI ประกอบไปด้วยสายสัญญาณ 4 เส้น ดังนี้

- MOSI (Master Out-Slave IN) ใช้ในการส่งข้อมูลออกจากอุปกรณ์มาสเตอร์ไปยังอุปกรณ์สเลฟ

- MISO (Master in-Slave Out) ในการรับข้อมูลออกจากอุปกรณ์สเลฟหรือมาสเตอร์อ่านข้อมูล

- SCK (Serial clock) สายสัญญาณนาฬิกา ที่ใช้งานร่วมกันระหว่างมาสเตอร์กับสเลฟ

- (Slave Select) สายสัญญาณสำหรับเลือกอุปกรณ์สเลฟ

อุปกรณ์ที่ทำหน้าที่เป็นมาสเตอร์ โดยปกติแล้วจะเป็นไมโครคอนโทรลเลอร์ หรืออาจกล่าวได้ว่าอุปกรณ์มาสเตอร์จะต้องควบคุมอุปกรณ์สเลฟได้ อุปกรณ์สเลฟมักจะเป็นไอซีที่พิเศษต่างๆ เช่น ไอซีวัดอุณหภูมิ ไอซีฐานเวลาจริง (Real-Time Clock) หรืออาจเป็นไมโครคอนโทรลเลอร์ที่ทำงานในโหมดสเลฟ ก็ได้เช่นกัน

2.1.5.1 คุณสมบัติของขาพอร์ต SS

เมื่อ SPI ทำงานในโหมดสเลฟ ขาพอร์ต SS (Slave Select) จะต้องกำหนดเป็นอินพุต หากจำนวนในโหมดมาสเตอร์ ผู้ใช้งานสามารถกำหนดได้ว่าต้องการให้ขาพอร์ต SS เป็นอินพุตหรือเอาต์พุต หากกำหนดเป็นเอาต์พุตขาพอร์ต SS จะไม่มีผลกับโหมด SPI แต่หากกำหนดเป็นอินพุตจะต้องกำหนดให้ขาพอร์ตมีสถานะเป็นลอจิก “1” หรือ High เพื่อให้ทำงานในโหมดมาสเตอร์ โดยการทำงานของขาพอร์ตในแต่ละโหมดแสดงดังตารางที่ 2.13

ตารางที่ 2.13 ขาพอร์ต

ขาพอร์ต	ทิศทางขาพอร์ตโหมดมาสเตอร์	ทิศทางขาพอร์ตโหมดสเลฟ	ขาพอร์ต ATmega328
MOSI	ผู้ใช้กำหนด	อินพุต	PB3
MOSO	อินพุต	ผู้ใช้กำหนด	PB4
SCK	ผู้ใช้กำหนด	อินพุต	PB5
SS	ผู้ใช้กำหนด	อินพุต	PB2

2.1.5.2 รีจิสเตอร์ที่เกี่ยวข้องกับการทำงานในโหมด SPI

1) รีจิสเตอร์ SPCR (SPI Control Register)

รีจิสเตอร์ควบคุมการทำงานของโมดูล SPI เพื่อกำหนดโหมดการทำงานการใช้งานอินเตอร์รัปต์รวมถึงการกำหนดความถี่สัญญาณนาฬิกาในการรับส่งข้อมูล รีจิสเตอร์ SPCR ประกอบไปด้วยบิตที่ 0-7 แสดงดังตารางที่ 2.14

ตารางที่ 2.14 รีจิสเตอร์ SPCR (SPI Control Register)

บิตที่	7	6	5	4	3	2	1	0
ชื่อบิต	SPIE	SPE	DORE	MSTR	CPOL	CPHA	SPR1	SPR0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
ค่าเริ่มต้น	0	0	0	0	0	0	0	0

- บิตที่ 7: บิต SPIE (SPI Interrupt Enable) บิตเปิดการใช้งานอินเตอร์รัปต์ SPI เมื่อบิต SPIF ถูกเซ็ตจะเกิดอินเตอร์รัปต์ขึ้น หากมีการเปิดการใช้งานอินเตอร์รัปต์โดยรวม (บิต I ในรีจิสเตอร์ SREG)

- บิตที่ 6: บิต SPE (SPI Enable) เซตบิต SPE เป็น "1"เพื่อเปิดการใช้โมดูล SPI

- บิตที่ 5: บิต DORE (Data Order) บิตกำหนดการส่งข้อมูล LSB หรือ MSB ก่อนเป็นอันดับแรก เซตบิต DORE เป็น "1" บิตLSB ของข้อมูลจะถูกส่งออกไปก่อน เซตบิต DORE เป็น "0" บิต MSB ของข้อมูลจะถูกส่งออกไปก่อน ดูรูปที่ 2.6 และ 2.7

- บิตที่ 4: บิต MSTR (Master/Slave Select) บิตกำหนดการทำงานของมาสเตอร์หรือสเลฟ เซตบิต MSTR เป็น "1" เมื่อต้องการทำงานในโหมดมาสเตอร์ และเซตบิต MSTR เป็น "0" เมื่อต้องการทำงานในโหมดสเลฟ

- บิตที่ 3: บิต CPOL (Clock Polarity) บิตกำหนดลักษณะของขาขอบสัญญาณเมื่ออยู่ในสถานะไอเดิล (Idle) เมื่อเซตเป็น "1" ขาสัญญาณ SCK จะเป็น Low เมื่ออยู่ในโหมดไอเดิล เซตเป็น "0" ขาสัญญาณ SCK จะเป็น High เมื่ออยู่ในโหมดไอเดิล การกำหนดขอบขาสัญญาณ SCK แสดงดังตารางที่ 2.15

ตารางที่ 2.15 การกำหนดขอบขาสัญญาณ SCK

CPOL	ต้นของสัญญาณ 1 ลูก	ปลายของสัญญาณ 1 ลูก
0	ขอบขาขึ้น	ขอบขาลง
1	ขอบขาลง	ขอบขาขึ้น

- บิตที่ 2: บิต CPHA (Clock Phase) เซตบิต CPHA เพื่อกำหนดรูปสัญญาณนาฬิกาหรือเฟสของสัญญาณนาฬิกาในการอ่านตัวอย่างข้อมูล (Sample) จากการเปลี่ยนแปลงสัญญาณนาฬิกาจาก High เป็น Low หรือ จาก Low เป็น High อย่างใดอย่างหนึ่ง การกำหนดบิต CPHA แสดงดังตารางที่ 2.16

ตารางที่ 2.16 การกำหนดการอ่านตัวอย่างของข้อมูล

CPHA	ต้นของสัญญาณ 1 ลูก	ปลายของสัญญาณ 1 ลูก
0	อ่านข้อมูล	เปลี่ยนแปลงข้อมูล
1	เปลี่ยนแปลงข้อมูล	อ่านข้อมูล

- บิตที่ 1:0: บิต SPR1, SPR0 (SPI Clock Rate Select 1 and 0) บิตกำหนดความถี่สัญญาณนาฬิกา (fosc) ในการรับส่งข้อมูลระหว่างอุปกรณ์ในโหมด SPI แสดงดังตารางที่ 2.17

ตารางที่ 2.17 การกำเนิดสัญญาณความถี่

SPI2X	SPR1	SPR0	ความถี่ SCK
0	0	0	fosc/4
0	0	1	fosc/16
0	1	0	fosc/64
0	1	1	fosc/128
1	0	0	fosc/2
1	0	1	fosc/8
1	1	0	fosc/32
1	1	1	fosc/64

2) รีจิสเตอร์ SPSR (SPI Status Register)

รีจิสเตอร์แสดงสถานการณ์ทำงานของโมดูล SPI เช่น การเซต แฟลช SPFI การตรวจสอบการชนกันของข้อมูล (Collision Flag) แสดงดังตารางที่ 2.18

ตารางที่ 2.18 รีจิสเตอร์ SPSR (SPI Status Register)

บิตที่	7	6	5	4	3	2	1	0
ชื่อบิต	SPIF	WCOL	-	-	-	-	-	SPI2X
Read/Write	R	R	R	R	R	R	R	R/W
ค่าเริ่มต้น	0	0	0	0	0	0	0	0

- บิตที่ 7: บิต SPIF (SPI Interrupt Flag) บิต SPIF จะถูกเซตเมื่อการรับส่งข้อมูลใน SPI เสร็จสมบูรณ์ หากมีการกำหนดใช้งานอินเตอร์รัปต์ SPI จะเกิดอินเตอร์รัปต์ขึ้น

- บิตที่ 6: บิต WCOL (Write COLlision Flag) บิต WCOL จะถูกเซตเมื่อตรวจพบว่าเกิดการเขียนข้อมูลในรีจิสเตอร์ SPDR ระหว่างการส่งข้อมูล จะถูกเคลียร์เมื่อมีการอ่านรีจิสเตอร์ SPSR บิต WCOL และการเข้าถึงรีจิสเตอร์ SPDR (SPI Data Register)

- บิตที่ 5:1 Res (Reserved Bits) บิตที่1 ถึง 5 สงวนไว้อ่านได้เป็น “0”

- บิตที่ 0: บิต SPI2X (Double SPI Speed Bit) หากกำหนดบิต SPI2X เป็น “1” จะเป็นการเพิ่มความถี่สัญญาณนาฬิกาเป็น 2 เท่า (2fosc) เมื่อทำงานในโหมดมาสเตอร์

3) รีจิสเตอร์ SPDR (SPI Data Register)

รีจิสเตอร์ใช้ในการรับส่งข้อมูลของโมดูล SPI ข้อมูลการรับการส่งจะถูกกำหนดในรีจิสเตอร์นี้แสดงดังตารางที่ 2.19

ตารางที่ 2.19 รีจิสเตอร์ SPDR (SPI Data Register)

บิตที่	7	6	5	4	3	2	1	0
ชื่อบิต	MSB							LSB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
ค่าเริ่มต้น	x	x	x	x	x	x	x	x

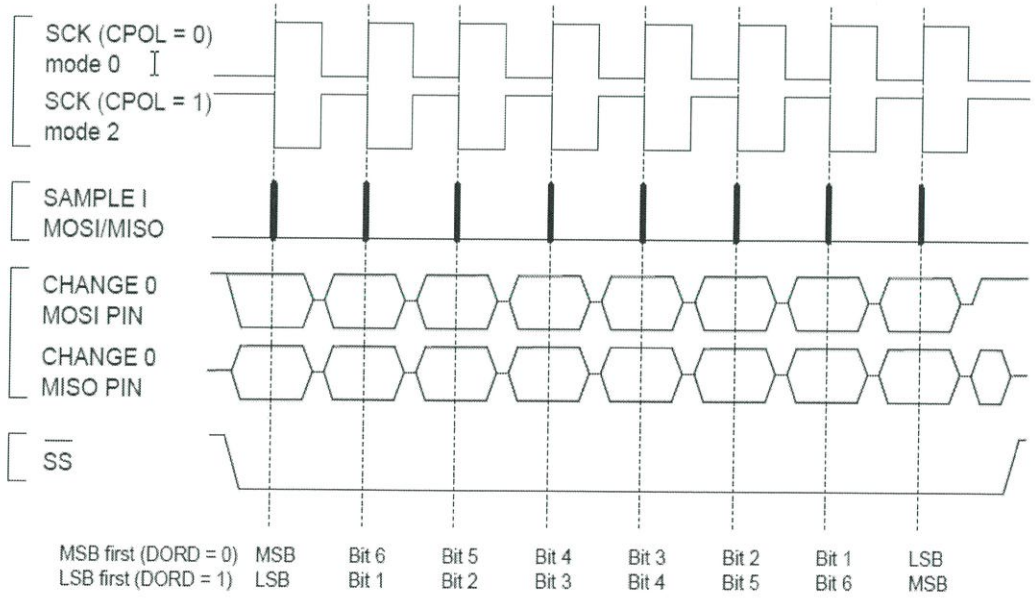
จากตารางที่ 2.19 จะเห็นว่ารีจิสเตอร์ข้อมูล ไว้สำหรับเขียนอ่านเพื่อรับส่งข้อมูลในโหมด SPI ทั้งส่วนของมาสเตอร์และสเลฟ

2.1.5.3 โหมดการรับส่งข้อมูล

การกำหนดรูปแบบของเฟสและลักษณะขอบขาของสัญญาณนาฬิกา จะถูกควบคุมผ่านทางบิต CPHA และ CPOL โดยสามารถกำหนดได้ 4 รูปแบบหรือ 4 โหมด สัญญาณนาฬิกาแสดงดังตารางที่ 2.20 และรูปแบบสัญญาณดังรูปที่ 5 และ 6

ตารางที่ 2.20 การกำหนดโหมดสัญญาณนาฬิกาในการรับส่งข้อมูล

SPI โหมด	การกำหนดค่าบิต	ต้นสัญญาณ	ปลายสัญญาณ
0	CPOL=0, CPHA=0	อ่านข้อมูล (ขอบขาขึ้น)	เปลี่ยนข้อมูล (ขอบขาลง)
1	CPOL=0, CPHA=1	เปลี่ยนข้อมูล (ขอบขาขึ้น)	อ่านข้อมูล (ขอบขาลง)
2	CPOL=1, CPHA=0	อ่านข้อมูล (ขอบขาลง)	เปลี่ยนข้อมูล (ขอบขาขึ้น)
3	CPOL=1, CPHA=1	เปลี่ยนข้อมูล (ขอบขาลง)	อ่านข้อมูล (ขอบขาขึ้น)



รูปที่ 2.5 รูปแบบการรับส่งข้อมูล SPI เมื่อกำหนด CPHA = 0 [1]

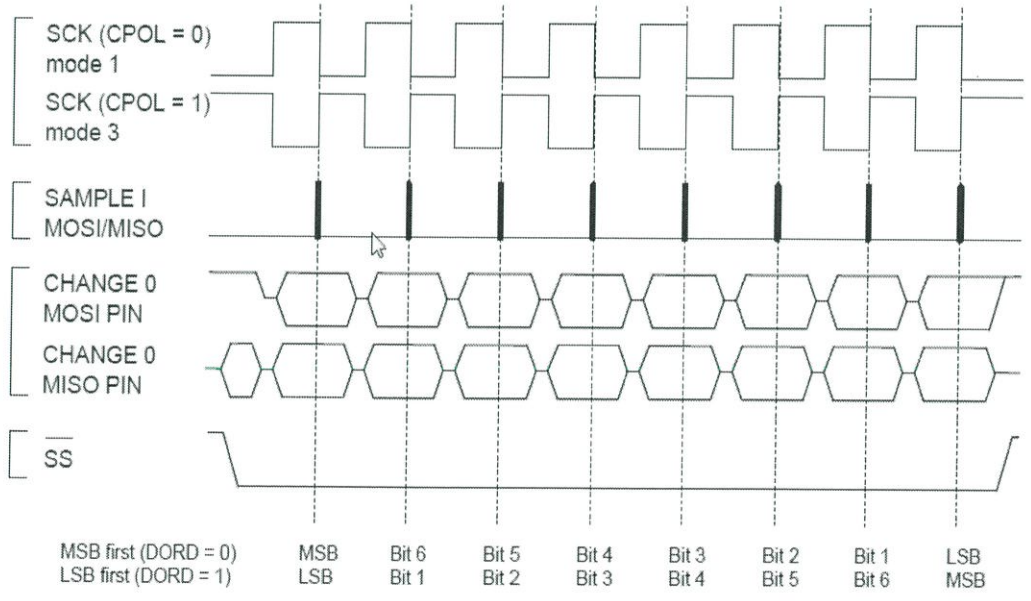
จากรูปที่ 2.5 สัญญาณ SPI ในโหมด 0 และ 2 จะเริ่มอ่านข้อมูลที่ต้นสัญญาณ SCK โดยที่มีรูปแบบการทำงานดังนี้

- เมื่อ SPI ทำงานโหมด 0 (CPOL=0 และ CPHA=0) การอ่านข้อมูล (Sample I MOSI/MISO) ที่ขา MOSI/MISO จะเริ่มต้นอ่านข้อมูลเมื่อขา SCK ที่ขอบขาขึ้น (เปลี่ยนจาก Low เป็น High) โดยการเปลี่ยนแปลงข้อมูลที่ขา MOSI และ MISO จะอยู่ระหว่างสัญญาณ SCK เป็น High เป็น Low (ระหว่างเปลี่ยนจาก High เป็น Low)

- เมื่อ SPI ทำงานในโหมด 2 (CPOL=0 และ CPHA=1) การอ่านข้อมูล ข้อมูลที่ขา MOSI/MISO จะเริ่มต้นอ่านข้อมูลเมื่อขา SCK ที่ขอบขาลง (เปลี่ยนจาก High เป็น Low) โดยการเปลี่ยนแปลงข้อมูลที่ขา MOSI และ MISO จะอยู่ระหว่างสัญญาณ SCK เป็น Low เป็น High (ระหว่างเปลี่ยนจาก Low เป็น High)

- การเลื่อนข้อมูลจะถูกกำหนดในบิต DORD ว่าต้องการให้บิต MSB หรือ บิต LSB ของข้อมูลถูกเลื่อนเข้าไปก่อนหรือหลัง

- ขา SS จะต้องเป็น Low ระหว่างที่มีการส่งข้อมูล



รูปที่ 2.6 รูปแบบการรับส่งข้อมูล SPI เมื่อกำหนด CPHA=1 [1]

จากรูปที่ 2.6 สัญญาณ SPI ในโหมด 1 และ 3 จะเริ่มอ่านข้อมูลที่ปลายสัญญาณ SCK โดยมีรูปแบบการทำงานเช่นเดียวกับรูปที่ 2.5

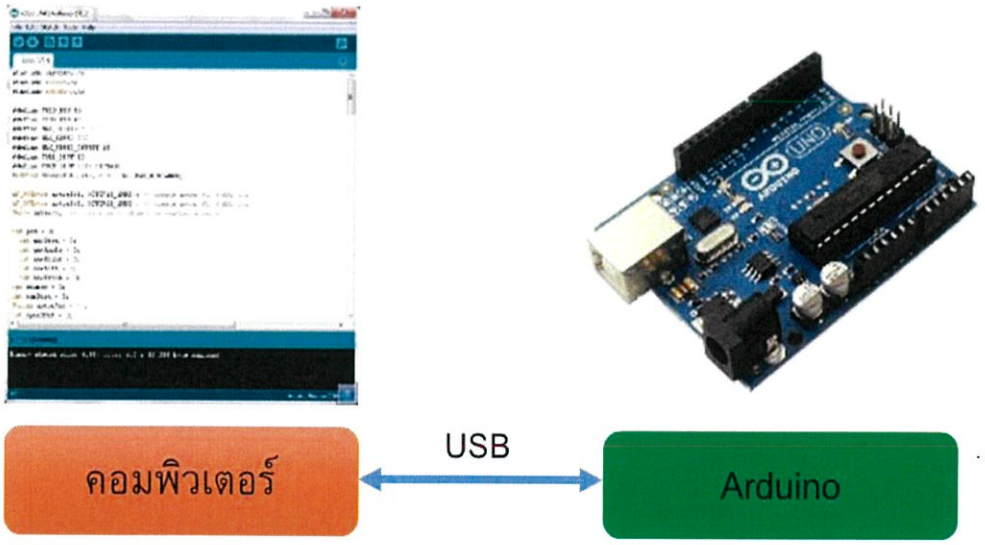
2.1.6 ไมโครคอนโทรเลอร์ Arduino

Arduino คือ ไมโครคอนโทรลเลอร์ ชนิดหนึ่ง ซึ่งเป็นแบบที่เรียกว่า Open Hardware กล่าวคือ Arduino อุปกรณ์ที่มีแบบส่วนประกอบเป็นมาตรฐานที่เปิดเผย หมายความว่า สามารถทำเองโดยใช้แบบที่มีการเปิดเผยทั่วไปก็ได้ หรือสามารถซื้อหาได้ง่าย มีราคาถูก มีซอฟแวร์ให้ใช้งานฟรี สามารถนำไปใช้งานทั่วไปหรือแบบธุรกิจได้โดยไม่ต้องเสียค่าลิขสิทธิ์ เป็นรูปแบบที่มีข้อมูลมากที่สุดบนอินเตอร์เน็ต มีการพัฒนาง่าย เพราะมีตัวอย่างมากมาย และไม่ต้องเขียนโปรแกรมในรูปแบบ Low Level หมายความว่า สามารถใช้คำสั่งเขียนโปรแกรมได้เสมือนโปรแกรมภาษาชั้นสูงทั่วไป

ความง่ายของบอร์ด Arduino ในการต่ออุปกรณ์เสริมต่างๆ คือผู้ใช้งานสามารถต่อวงจรอิเล็กทรอนิคส์จากภายนอกแล้วเชื่อมต่อเข้ามาที่ขา I/O ของบอร์ด หรือเพื่อความสะดวกสามารถเลือกต่อกับบอร์ดเสริม (Shield) ประเภทต่างๆ เช่น XBee Shield, Music Shield, Relay Shield, Wireless Shield, GPRS Shield เป็นต้น มาเสียบกับบอร์ดบนบอร์ด Arduino แล้วเขียนโปรแกรมพัฒนาต่อได้เลย

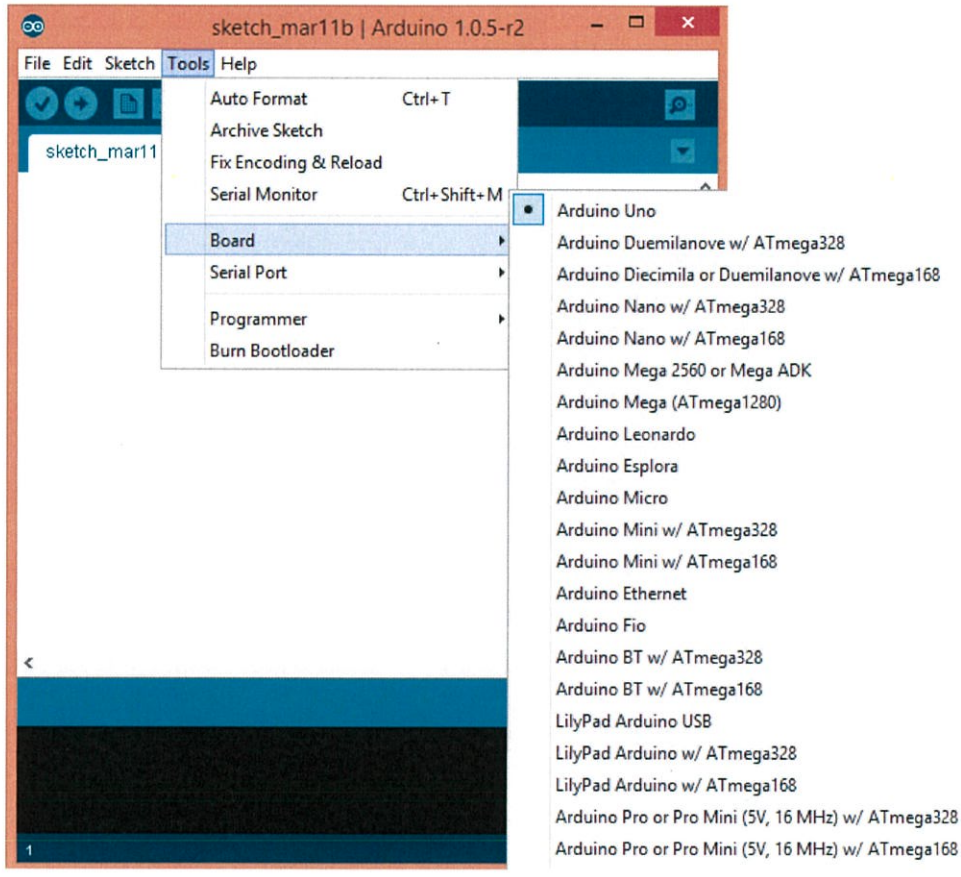
2.1.6.1 รูปแบบการเขียนโปรแกรมบน Arduino

1) เขียนโปรแกรมบนคอมพิวเตอร์โดยใช้ภาษา C++ ในการเขียนโปรแกรม ผ่านทางโปรแกรม Arduino IDE การโปรแกรม Arduino โดยผ่านสาย USB แสดงดังรูปที่ 2.7

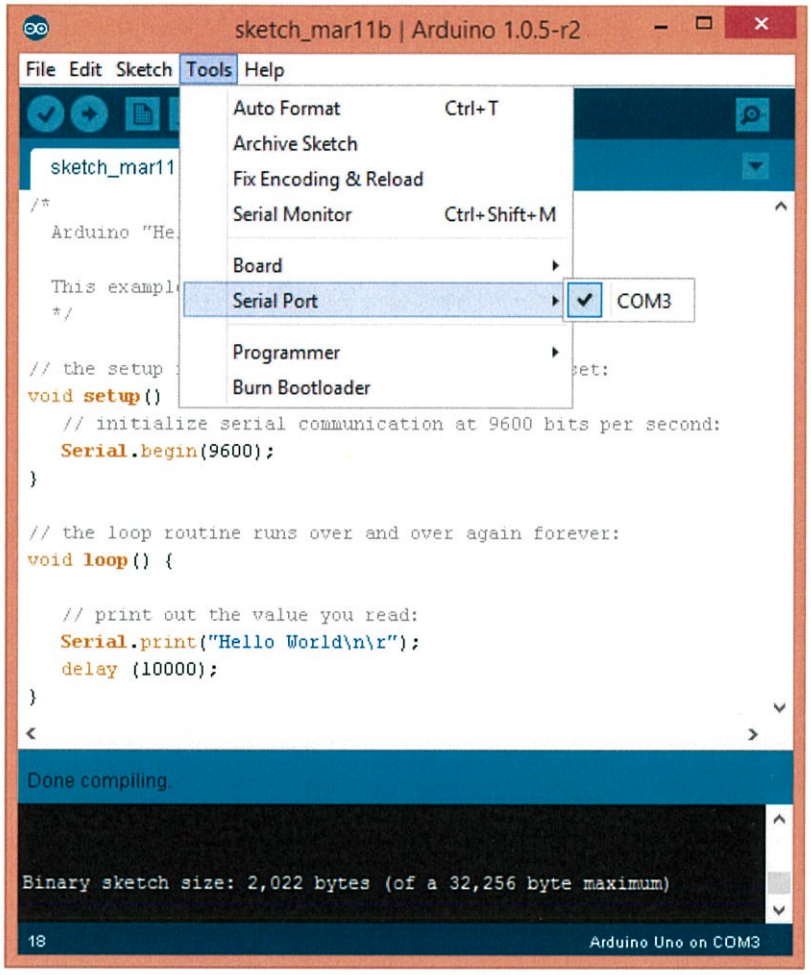


รูปที่ 2.7 โปรแกรมบนคอมพิวเตอร์ ผ่านทางโปรแกรม Arduino โดยใช้ USB [2]

2) หลังจากที่เขียนโค้ดโปรแกรมเรียบร้อยแล้ว ให้ผู้ใช้งานเลือกรุ่นบอร์ด Arduino เป็นบอร์ด Arduino Uno แสดงดังรูปที่ 2.8 โดย และหมายเลข Com port ที่ 3 ดังแสดงในรูปที่ 2.9

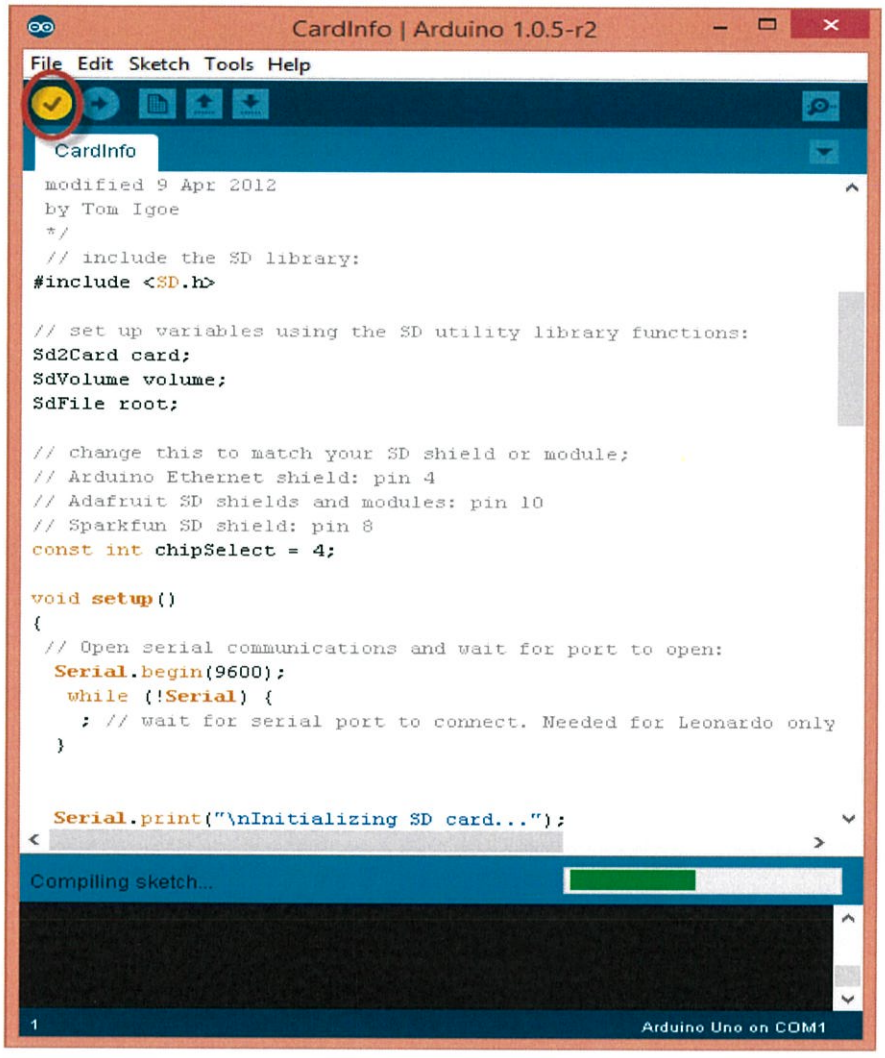


รูปที่ 2.8 เลือกรุ่นบอร์ด Arduino ที่ต้องการ upload [2]

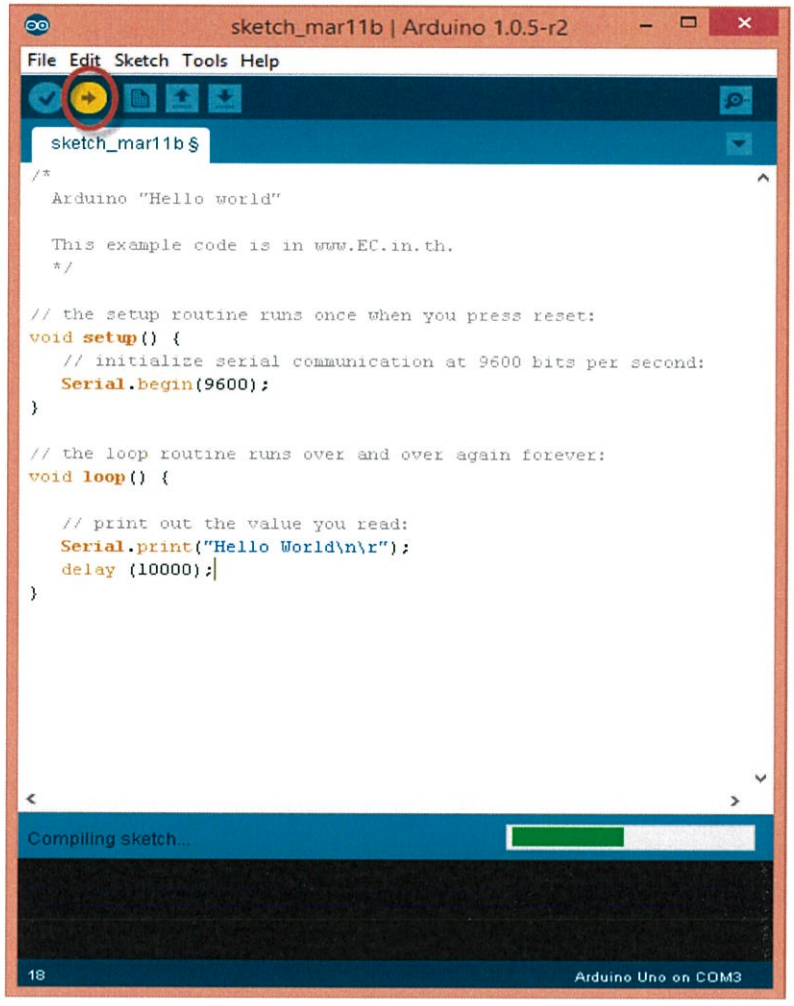


รูปที่ 2.9 เลือกหมายเลข Comport ของบอร์ด [2]

3) กดปุ่ม Verify เพื่อตรวจสอบความถูกต้องและ Compile โค้ดโปรแกรมดังแสดงในรูปที่ 2.10 จากนั้นกดปุ่ม Upload โค้ดดังแสดงในรูปที่ 2.10 โปรแกรมไปยังบอร์ด Arduino ผ่านทางสาย USB เมื่ออัพโหลดเรียบร้อยแล้ว จะแสดงข้อความแถบข้างล่าง “Done uploading” และบอร์ดจะเริ่มทำงานตามที่เขียนโปรแกรมไว้ได้ทันที

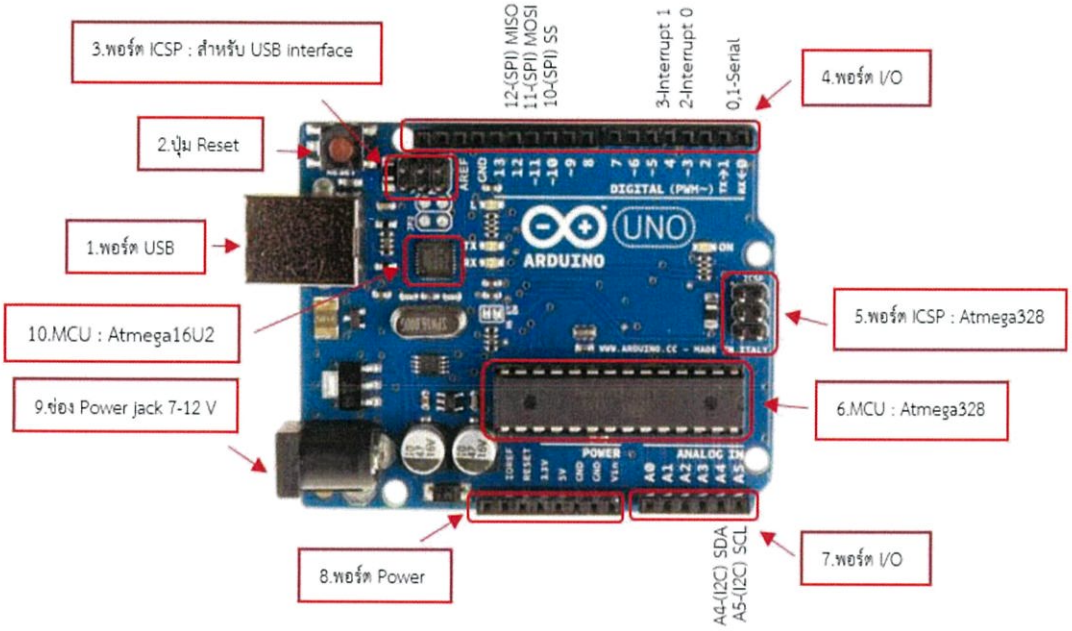


รูปที่ 2.10 กดปุ่ม Verify เพื่อตรวจสอบความถูกต้อง [2]



รูปที่ 2.11 Upload โค้ดโปรแกรม [2]

2.1.6.2 Layout & Pin out Arduino Board (Model: Arduino UNO R3)



รูปที่ 2.12 Layout & Pin out Arduino Board (Model: Arduino UNO R3) [2]

จากรูปที่ 2.12 อธิบาย Layout & Pin out Arduino Board ได้ดังต่อไปนี้

1) USB Port: ใช้สำหรับต่อกับ Computer เพื่ออับโหลดโปรแกรมเข้า MCU และจ่ายไฟให้กับบอร์ด

2) Reset Button: เป็นปุ่ม Reset ใช้กดเมื่อต้องการให้ MCU เริ่มการทำงานใหม่

3) ICSP Port ของ Atmega16U2 เป็นพอร์ตที่ใช้โปรแกรม Visual Com port บน Atmega16U2

4) I/O Port: Digital I/O ตั้งแต่ขา D0 ถึง D13 นอกจากนี้ บาง Pin จะทำหน้าที่อื่นๆ เพิ่มเติมด้วย เช่น Pin 0,1 เป็นขา Tx,Rx Serial, Pin 3,5,6,9,10 และ 11 เป็นขา PWM

5) ICSP Port: Atmega328 เป็นพอร์ตที่ใช้โปรแกรม Bootloader

6) MCU: Atmega328 เป็น MCU ที่ใช้บนบอร์ด Arduino

7) I/O Port: นอกจากจะเป็น Digital I/O แล้ว ยังเปลี่ยนเป็นช่องรับสัญญาณอนาล็อก ตั้งแต่ขา A0-A5

8) Power Port: ไฟเลี้ยงของบอร์ดเมื่อต้องการจ่ายไฟให้กับวงจรภายนอก ประกอบด้วยขาไฟเลี้ยง +3.3 V, +5V, GND, V_{in}

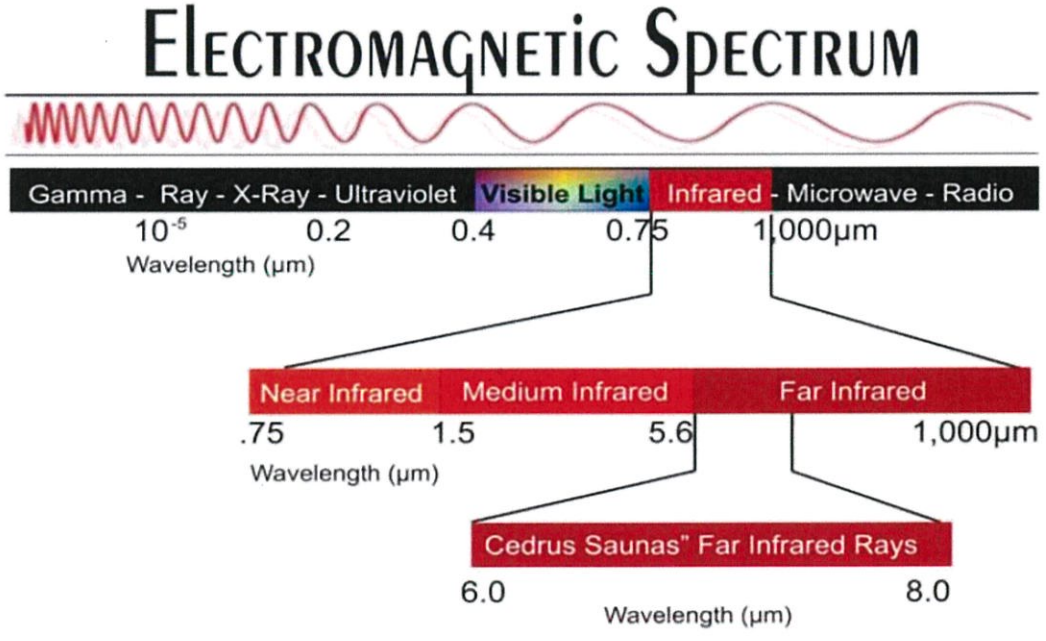
9) Power Jack: รับไฟจาก Adapter โดยที่แรงดันอยู่ระหว่าง 7-12 V

10) MCU ของ Atmega16U2 เป็น MCU ที่ทำหน้าที่เป็น USB to Serial โดย Atmega328 จะติดต่อกับ Computer ผ่าน Atmega16U2

2.2 รังสีอินฟราเรด (Infrared Radiation)

2.2.1 ความหมายของรังสีอินฟราเรด

รังสีอินฟราเรด หรือ รังสีใต้แดงเป็นคลื่นแม่เหล็กไฟฟ้าชนิดหนึ่งที่มีความถี่อยู่ระหว่าง 1011-1014 Hz และมีความยาวคลื่นระหว่าง 0.75 ไมโครเมตรถึง 1000 ไมโครเมตร สสารที่มีอุณหภูมิมากกว่า 0 องศาเคลวิน จะปล่อยรังสีอินฟราเรดออกมากจากตัวมันเองเสมอ



รูปที่ 2.13 คลื่นแม่เหล็กไฟฟ้าในช่วงความยาวคลื่นต่างๆ [4]

จากรูปแสดงที่ 2.13 คลื่นแม่เหล็กไฟฟ้าในช่วงความยาวคลื่นต่างๆ จะเห็นได้ว่า รังสีอินฟราเรดจะมีความยาวคลื่นที่มากกว่าแสงที่ตามองเห็นที่ยาวที่สุด ซึ่งก็คือสีแดง ด้วยเหตุนี้เอง จึงเรียกรังสีชนิดนี้ว่า รังสีอินฟราเรด หรือ รังสีใต้แดง

รังสีอินฟราเรดสามารถแบ่งช่วงความยาวคลื่นได้เป็น 3 ช่วงคือ

2.2.1.1 รังสีอินฟราเรดช่วงคลื่นสั้น (NIR) ช่วงคลื่นสั้นของรังสีอินฟราเรดจะมีความยาวคลื่นประมาณ 0.7 ไมโครเมตรจนถึง 1.5 ไมโครเมตร รังสีอินฟราเรดช่วงคลื่นสั้นมักจะประยุกต์ใช้ในงานถ่ายภาพความร้อน

2.2.1.2 รังสีอินฟราเรดช่วงคลื่นกลาง (MIR) ช่วงคลื่นกลางของรังสีอินฟราเรดจะมีความยาวคลื่นประมาณ 1.5 ไมโครเมตรจนถึง 5.6 ไมโครเมตร อินฟราเรดระยะกลางมักประยุกต์ใช้กับระบบนาวิถีของจรวด Missile

2.2.1.3 รังสีอินฟราเรดช่วงคลื่นยาว (FIR) ช่วงคลื่นยาวของรังสีอินฟราเรดจะมีความยาวคลื่นประมาณ 5.6 ไมโครเมตรขึ้นไป รังสีประเภทนี้เป็นช่วงคลื่นยาวจึงมีพลังงานความร้อนไม่มากนักจึงจานิยมใช้ในการบาบัดผู้ป่วย เช่น อาการปวดเมื่อยเรื้อรัง และผู้ป่วยด้วยโรคความดันโลหิต รวมถึงการควบคุมน้าหนัก เป็นต้น

2.2.2 ประเภทของคลื่นแม่เหล็กไฟฟ้า

แสงที่ตามองเห็น (Visible light) เป็นเพียงส่วนหนึ่งของคลื่นแม่เหล็กไฟฟ้าในช่วงซึ่งประสาทตาของมนุษย์สามารถสัมผัสได้ ซึ่งมีความยาวคลื่นอยู่ระหว่าง 400 – 700 นาโนเมตร (1 เมตร = 1,000,000,000 นาโนเมตร) หากนำแท่งแก้วปริซึม (Prism) มาหักเหแสงอาทิตย์เราจะเห็นว่าแสงสีขาวถูกหักเหออกเป็นสีม่วง คราม น้ำเงิน เขียว เหลือง แสด แดง คล้ายกับสีของรุ้งกินน้ำ เรียกว่า “สเปคตรัม” (Spectrum) แสงแต่ละสีมีความยาวคลื่นแตกต่างกัน สีม่วงมีความยาวคลื่นน้อยที่สุด สีแดงมีความยาวคลื่นมากที่สุด

นอกจากแสงที่ตามองเห็นแล้วยังมีคลื่นแม่เหล็กไฟฟ้าชนิดอื่นๆ ได้แก่ รังสีที่มีความยาวคลื่นถัดจากสีแดงออกไป เรียกว่า เราเรียกว่า “รังสีอินฟราเรด” หรือ “รังสีความร้อน” เรามองไม่เห็นรังสีอินฟราเรด แต่เราก็รู้สึกถึงความร้อนได้ สัตว์บางชนิด เช่น งู มีประสาทสัมผัสรังสีอินฟราเรด มันสามารถทราบตำแหน่งของเหยื่อได้ โดยการสัมผัสรังสีอินฟราเรดซึ่งแผ่ออกมาจากร่างกายของเหยื่อ รังสีที่มีความยาวคลื่นน้อยกว่าแสงสีม่วงเรียกว่า “รังสีอุลตราไวโอเล็ต” แม้ว่าเราจะมองไม่เห็น แต่เมื่อเราตากแดดนานๆ ผิวหนังจะไหม้ด้วยรังสีชนิดนี้ นอกจากรังสีอุลตราไวโอเล็ตและรังสีอินฟราเรดแล้ว ยังมีคลื่นแม่เหล็กไฟฟ้าประเภทอื่นๆ ซึ่งเรียงลำดับตามความยาวคลื่นได้ดังนี้

2.2.2.1 รังสีแกมมา (Gamma ray) เป็นคลื่นแม่เหล็กไฟฟ้าที่มีความยาวคลื่นน้อยกว่า 0.01 นาโนเมตร โฟตอนของรังสีแกมมามีพลังงานสูงมาก กำเนิดจากแหล่งพลังงานนิวเคลียร์ เช่น ดาวระเบิด หรือ ระเบิดปรมาณู เป็นอันตรายมากต่อสิ่งมีชีวิต

2.2.2.2 รังสีเอ็กซ์ (X-ray) มีความยาวคลื่น 0.01 - 1 นาโนเมตร มีแหล่งกำเนิดในธรรมชาติมาจากดวงอาทิตย์ เราใช้รังสีเอ็กซ์ในทางการแพทย์ เพื่อส่องผ่านเซลล์เนื้อเยื่อ แต่ถ้าได้ร่างกายได้รับรังสีนี้มากๆ ก็จะเป็นอันตราย

2.2.2.3 รังสีอุลตราไวโอเล็ต (Ultraviolet radiation) มีความยาวคลื่น 1 - 400 นาโนเมตร รังสีอุลตราไวโอเล็ตมีอยู่ในแสงอาทิตย์ เป็นประโยชน์ต่อร่างกาย แต่หากได้รับมากเกินไปก็จะทำให้ผิวไหม้ และอาจทำให้เกิดมะเร็งผิวหนัง

2.2.2.4 รังสีอินฟราเรด (Infrared radiation) มีความยาวคลื่น 700 นาโนเมตร – 1 มิลลิเมตร โลกและสิ่งชีวิตแผ่รังสีอินฟราเรดออกมา ก๊าซเรือนกระจก เช่น คาร์บอนไดออกไซด์ และไอน้ำ ในบรรยากาศดูดซับรังสีนี้ไว้ ทำให้โลกมีความอบอุ่น เหมาะกับการดำรงชีวิต

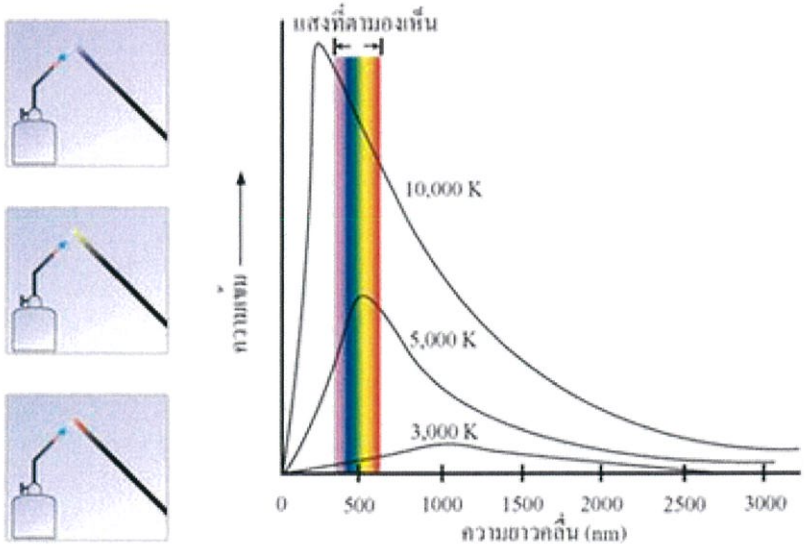
2.2.2.5 คลื่นไมโครเวฟ (Microwave) มีความยาวคลื่น 1 มิลลิเมตร – 10 เซนติเมตร ใช้ประโยชน์ในด้านโทรคมนาคมระยะไกล นอกจากนั้นยังนำมาประยุกต์สร้างพลังงานในเตาอบอาหาร

2.2.2.6 คลื่นวิทยุ (Radio wave) เป็นคลื่นแม่เหล็กไฟฟ้าที่มีความยาวคลื่นมากที่สุด คลื่นวิทยุสามารถเดินทางผ่านชั้นบรรยากาศได้ จึงถูกนำมาใช้ประโยชน์ในด้านการสื่อสารโทรคมนาคม

รังสีอินฟราเรด (infrared) คือ แสงที่มนุษย์เราไม่สามารถมองเห็นได้ เป็นคลื่นแม่เหล็กไฟฟ้าที่มีความถี่ต่ำกว่าแสงสีแดง แหล่งกำเนิดของรังสีอินฟราเรด คือ ความร้อน (Heat) จะเกิดการแผ่รังสีความร้อนกับวัตถุใดก็ตามที่มีอุณหภูมิสูงกว่าค่า Absolute Zero หรืออุณหภูมิศูนย์สัมบูรณ์ (-273.15 องศาเซลเซียล หรือ 0 องศา เคลวิน) เท่านั้น โดยจะมีการแผ่รังสีในย่านอินฟราเรดหรือรังสีความร้อนที่แผ่ออกมาจากพื้นผิว ของวัตถุ แม้วัตถุนั้นจะเป็นไนโตรเจนเหลวที่มีอุณหภูมิอุณหภูมิ -196 องศาเซลเซียล ก็ ยังมีการแผ่รังสีอินฟราเรดออกมาได้ ถึงแม้จะมีค่าน้อยก็ตาม ซึ่งในความเป็นจริงและในชีวิตประจำวัน เราจะสัมผัสและเกี่ยวข้องกับรังสีอินฟราเรดอยู่ตลอดเวลา กล่าวคือ เรารู้สึกร้อนเมื่ออยู่กลางแสงอาทิตย์หรือใกล้ไฟ ถึงแม้ว่าตามนุษย์จะไม่สามารถมองเห็นรังสีอินฟราเรดได้ แต่ผิวเราก็สามารถรู้สึกสัมผัสถึงความร้อนได้ นั่นแสดงให้เห็นว่าวัตถุที่ร้อนย่อมแผ่พลังงานรังสีความร้อนหรือรังสี อินฟราเรดออกมาได้มากกว่าวัตถุที่เย็น

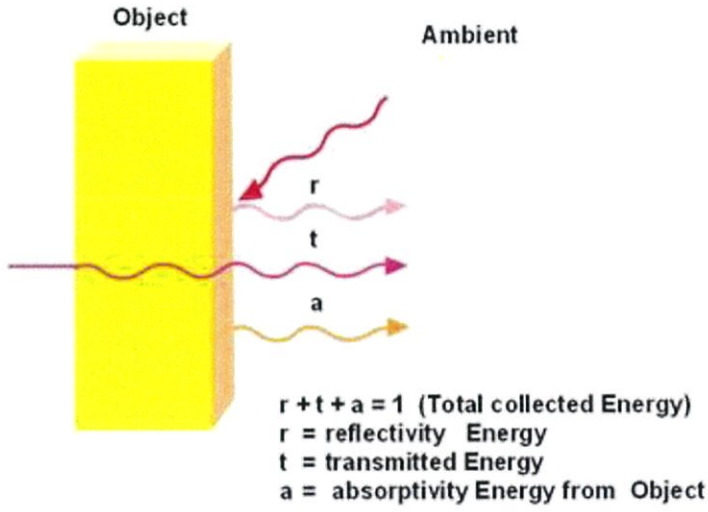
2.2.3 ความสัมพันธ์ระหว่างความยาวคลื่นและอุณหภูมิ

วัตถุทุกชนิดที่มีอุณภูมิสูงกว่า 0 เคลวิน (-273°C) มีพลังงานภายในตัว และมีการแผ่รังสีคลื่นแม่เหล็กไฟฟ้า ความยาวของคลื่นแม่เหล็กไฟฟ้าแปรผกผันกับอุณหภูมิ มิใช่มีเพียงสิ่งที่มีอุณหภูมิสูง ดังเช่น ดวงอาทิตย์ และไส้หลอดไฟฟ้า จึงมีการแผ่รังสี หากแต่สิ่งที่มีอุณหภูมิต่ำ ดังเช่น ร่างกายมนุษย์ และน้ำแข็ง ก็มีการแผ่รังสีเช่นกัน เพียงแต่ตาของเรามองไม่เห็น เมื่อพิจารณาในภาพที่ 2.8 เมื่อเราให้พลังงานความความร้อนแก่แท่งโลหะ เมื่อมันเริ่มร้อน มันจะเปล่งแสงสีแดง (สามารถเห็นได้จากขดลวดของเตาไฟฟ้า) เมื่อมันร้อนมากขึ้น มันจะเปล่งแสงสีเหลือง และ.ในที่สุดมันจะเปล่งแสงสีขาวอมน้ำเงิน เมื่อพิจารณาเส้นกราฟ จะเห็นว่า เมื่อโลหะมีอุณหภูมิ 3,000 K ความยาวคลื่นสูงสุดที่ยอดกราฟจะอยู่ที่ 1,000 nm (นาโนเมตร) ซึ่งตรงกับย่านรังสีอินฟราเรด ซึ่งสายตาเราไม่สามารถมองเห็นรังสีชนิดนี้ เราจึงเห็นแท่งโลหะแผ่แสงสีแดง เนื่องจากเป็นความยาวคลื่นที่ต่ำที่สุดแล้วที่เราสามารถมองเห็นได้ เมื่อแท่งเหล็กมีอุณหภูมิ 5,000 K ความยาวคลื่นสูงสุดที่ยอดกราฟจะอยู่ที่ 580 nm เราจึงมองเห็นแท่งโลหะเปล่งแสงสีเหลือง เมื่อแท่งเหล็กมีอุณหภูมิ 10,000 K ความยาวคลื่นสูงสุดที่ยอดกราฟจะอยู่ที่ 290 nm ซึ่งตรงกับย่านรังสีอุลตราไวโอเล็ก ซึ่งสายตาเราไม่สามารถมองเห็นรังสีชนิดนี้ เราจึงเห็นแท่งโลหะแผ่แสงสีม่วง เนื่องจากเป็นความยาวคลื่นที่สูงที่สุดแล้ว ที่เราสามารถมองเห็นได้



รูปที่ 2.14 ความสัมพันธ์ระหว่างความยาวคลื่นกับอุณหภูมิ [4]

จากรูปที่ 2.14 แสดงให้เห็นว่า วัตถุร้อน จะมีพลังงานสูง และแผ่รังสีคลื่นสั้น ส่วนวัตถุเย็นจะมีพลังงานต่ำ แผ่รังสีคลื่นยาว สำหรับรังสีอินฟราเรดหรือรังสีความร้อนนั้น จะแผ่ออกมาจากพื้นผิวของวัตถุชั้นแรกเท่านั้น ในรูปแบบของพลังงานคลื่นแม่เหล็กไฟฟ้า โดยรังสีอินฟราเรดหรือรังสีความร้อนที่ตกกระทบบนพื้นผิววัตถุ บางส่วนจะถูกดูดกลืน (Absorbed) บางส่วนจะสะท้อน (Refected) และบางส่วนจะส่งผ่านหรือทะลุผ่านออกไป (Transmitted) ดังแสดงในรูปที่ 2.15 ซึ่งสัดส่วนเหล่านี้แสดงได้ในรูปของสมการที่ 1



รูปที่ 2.15 การถ่ายเทพลังงานคลื่นแม่เหล็กไฟฟ้าของวัตถุในรูปแบบรังสีอินฟราเรด [4]

$$r + t + a = 1 \tag{2.1}$$

เมื่อวัตถุอยู่ในสภาพภาวะแวดล้อมที่สมดุล การแผ่พลังงานของรังสีอินฟราเรดของวัตถุจะมีค่าเท่ากับปริมาณที่ดูดกลืน เป็นผลให้วัตถุที่สามารถจะดูดกลืนรังสีได้ดีก็จะแผ่รังสีได้ดีด้วย สรุปได้ว่า ตัวดูดกลืนรังสีในอุดมคติ (Ideal Absorbor) คือ วัตถุที่มีค่า a (Absorptivity) เป็น 1 และเมื่อเรากล่าวถึงการแผ่รังสีที่แผ่ออกจากวัตถุ โดยต่างจากการดูดกลืนจะใช้ค่าว่า e (Emissivity) แทนที่จะใช้ Absorptivity (a) อย่างไรก็ตาม ค่าทั้งสองมีความสัมพันธ์กันโดยตรง ตามกฎของ Kirchhoff's Law ฉะนั้น e = a จากสมการที่ 1 สามารถเขียนใหม่ได้ว่าดังสมการที่ 2.2

$$e + t + r = 1 \qquad (2.2)$$

2.3 เซ็นเซอร์ตรวจจับความเคลื่อนไหว (Motion Sensor)

การตรวจจับความเปลี่ยนแปลงของอินฟราเรดซึ่งจากหลักการแล้ว วัตถุที่มีอุณหภูมิที่สูงกว่า -273 องศาเซลเซียส จะมีการเปล่งรัศมีของอินฟราเรดออกมา คลื่นรังสีความร้อนที่แผ่กระจายออกมาจะครอบคลุมแถบความถี่ในย่านประมาณ 0.74 ไมโครเมตร ถึง 100 ไมโครเมตร ซึ่งปกติเราจะเรียกแถบความถี่ในช่วงนี้ว่า แถบอินฟราเรด (Infrared Region) ด้วยหลักการนี้เองเราจึงนำมาใช้แยกความแตกต่างระหว่างการเคลื่อนไหวของ คน สัตว์ ต้นไม้ ได้ คือ เมื่อมีการแผ่รังสีของวัตถุ รังสีที่แผ่ออกมาจะมีระบบการรวบรวมแสงโดย เฟรสเนลเลนส์ (Fresnel Lens) อาจจะเป็นเลนส์ หรือกระจกต่างๆ สำหรับทำให้เกิดการแผ่รังสีมาตกกระทบยังอุปกรณ์ที่เรียกว่าไพโร (Pyro Electric)

2.3.1 ส่วนประกอบหลักของอุปกรณ์ตรวจจับการเคลื่อนไหวมีอยู่ 3 ส่วน

2.3.1.1 Pyro Electric

โครงสร้างภายในของตัวตรวจจับแบบไพโรอิเล็กตริกมีส่วนประกอบที่สำคัญๆ คือ

1) ตัวไวแสงที่ทำจากผลึกของลิเทียมซัลเฟต

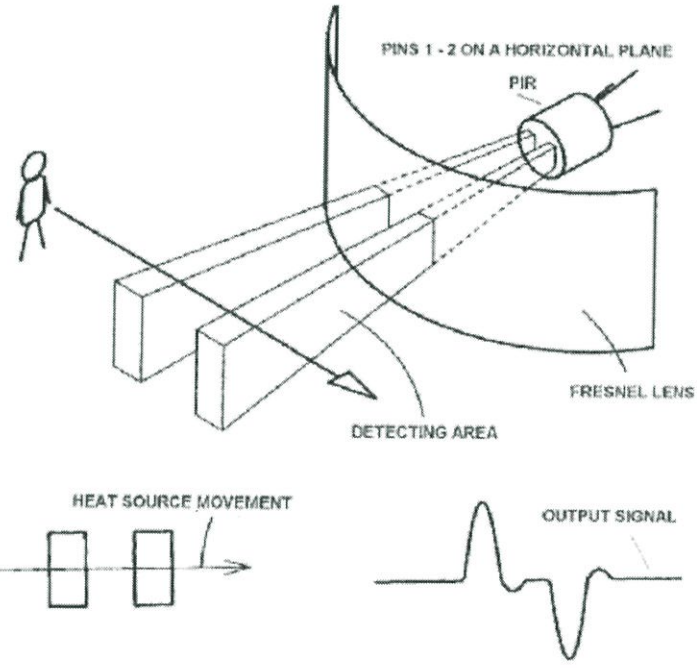
2) เฟ็ต (Fet)

ชิ้นของผลึกแร่ ในแต่ละตัวจะมีขนาดประมาณ 2 × 1 ตาราง มิลลิเมตร โดยต่ออนุกรมกันอยู่แต่ต่อกลับขั้วกัน ซึ่งจากลักษณะสมบัติของผลึกชนิดนี้ เมื่อมันถูกทำให้ร้อนจะเกิดการประจุไฟฟ้าที่ผิวของทั้งสองด้านที่อยู่ตรง ข้ามกัน ดังนั้นเมื่อมีสัญญาณใดๆ มาตกกระทบตัวมันเข้าอย่างต่อเนื่อง จะเป็นเหตุให้สัญญาณลบและบวกถูกผลิตขึ้น ซึ่งสัญญาณที่ถูกผลิตขึ้นนี้จะเกิดการเปลี่ยนแปลงเป็นช่วงกว้างมากจากยอดถึง ยอดของสัญญาณ แต่เนื่องจากผลึกทั้งสองต่อกลับขั้วกันอยู่ จึงทำให้ผลรวมของทั้งสองสัญญาณหักล้างกันหมดไป ลักษณะดังกล่าวนับเป็นผลดีอย่างมาก ในอันที่จะป้องกันการทำงานในสภาวะที่เราไม่ต้องการ เช่น

- แสงรบกวนจากภายนอก เช่น แสงแดดซึ่งจัดเป็นแสงที่ต่อเนื่อง
- การเปลี่ยนแปลงอุณหภูมิในสภาวะแวดล้อม ซึ่งโดยปกตินั้นจะไม่ได้เปลี่ยนแปลงแบบทันทีทันใด
- เปลี่ยนแปลงเนื่องจากการสั่นสะเทือน ที่ไม่ได้เกิดจากคนหรือสัตว์

2.3.1.2 Fresnel Lens

เลนส์ที่ทำจากพลาสติกหรือแก้วก็ได้ เพียงแต่มีร่องรอยเป็นวงกลมที่มีรัศมีไม่เท่ากัน ซ้อนทับกันอยู่ เพื่อที่จะ Focus ความร้อนจากตำแหน่งต่างๆ ให้มาตกกระทบที่ Pyro ดังแสดงในรูปที่ 2.16



รูปที่ 2.16 Fresnel Lens [4]

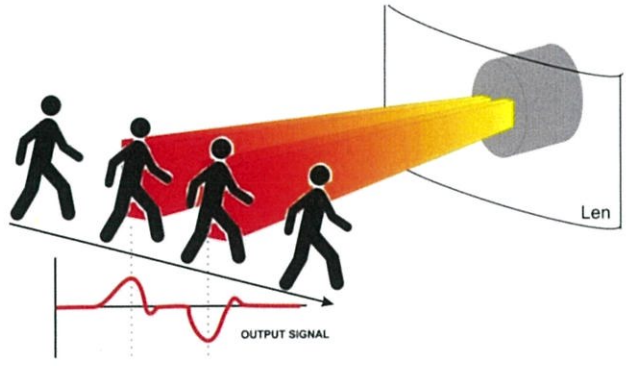
2.3.1.3 Circuit Control

โดยทั่วไปจะใช้การประมวลผลสัญญาณแบบ DSP (Digital Signal Processing) เพื่อควบคุมและวิเคราะห์สัญญาณ

2.3.2 หลักการทำงานของ Passive infrared sensors (PIR sensor)

หลักการทำงานของ Passive infrared sensors แสดงดังรูปที่ 2.17 ซึ่งภายใน PIR sensor จะมีอุปกรณ์ตรวจจับรังสีอินฟราเรด (Infrared) อยู่ 2 ชุดด้วยกันดังรูป เมื่อมีคน หรือ สัตว์ ที่มีความอบอุ่นในร่างกายเคลื่อนที่ผ่านเข้ามาใน พื้นที่โซนที่ PIR สามารถตรวจจับคลื่นรังสีอินฟราเรด (Infrared) ที่แพร่ออกมาจากสิ่งมีชีวิตได้ Passive infrared sensors (PIR sensor) จะเปลี่ยนคลื่นรังสีอินฟราเรด (Infrared) ให้กลายเป็นกระแสไฟฟ้าดังรูป จะเห็นว่าเมื่อมี

สิ่งมีชีวิตเคลื่อนที่ผ่านอุปกรณ์ตรวจจับรังสีอินฟราเรด (Infrared) ตัวที่ 1 จะได้สัญญาณส่งออก (Output) ออกมาสูงกว่าแรงดันปรกติและเมื่อสิ่งมีชีวิตเคลื่อนที่ผ่านอุปกรณ์ตรวจจับรังสีอินฟราเรด (Infrared) ตัวที่ 2 จะได้แรงดันส่งออก (Output) ต่ำกว่าค่าแรงดันปกติ



รูปที่ 2.17 การทำงานของ Passive infrared sensors (PIR sensor) [4]

2.3.3 โมดูลตรวจความเคลื่อนไหว KC7783R

ในโครงงานนี้ได้เลือกใช้โมดูลตรวจความเคลื่อนไหว KC7783R ดังรูปที่ 2.18 ของบริษัท Comedia

รูปที่ 2.18 โมดูลตรวจความเคลื่อนไหว KC7783R [4]

โมดูลตรวจความเคลื่อนไหว KC7783R เป็นแผงวงจรตรวจจับความเคลื่อนไหวด้วยการตรวจวัดความร้อน สามารถวัดได้ไกลถึง 6 เมตร มีขนาดเล็ก ถูกออกแบบมาให้ใช้งานกับไมโครคอนโทรลเลอร์ได้ง่ายโดยใช้ขาเชื่อมต่อเพียง 1 ขา และสามารถเลือกโหมดสัญญาณเอาต์พุตได้

คุณสมบัติ

- ใช้ไฟเลี้ยง +3 ถึง +5 โวลต์ กระแสไฟฟ้ามากกว่า 3 มิลลิแอมป์
- สามารถตรวจจับความเคลื่อนไหวได้ในช่วง 6 เมตร
- รัศมีในการตรวจจับ 70 องศา
- สัญญาณเอาต์พุต 1 บิต
- อุณหภูมิในการทำงานอยู่ในช่วง 0 ถึง 50 องศาเซลเซียส (ใช้ในที่ร่ม)
- ใช้เวลาในการเรียนรู้สภาพแวดล้อม 10 ถึง 60 วินาที ในช่วงเวลานี้ควรจะมีตามการเคลื่อนไหวน้อยที่สุดในพื้นที่ที่มีการตรวจจับ เพื่อให้สามารถทำงานได้อย่างถูกต้อง

โหมดสัญญาณเอาต์พุต

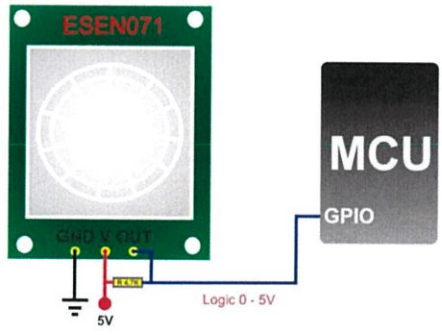
โหมดสัญญาณเอาต์พุตสามารถเลือกใช้งานได้ 2 แบบ คือ

1) สัญลักษณ์ H (HIGH) หมายถึง เอาต์พุตเป็นลอจิก 0 เมื่ออยู่สภาวะปกติและเอาต์พุต เป็นลอจิก 1 เมื่อตรวจจับความเคลื่อนไหวได้

2) สัญลักษณ์ L (LOW) หมายถึง เอาต์พุตเป็นลอจิก 0 เมื่ออยู่ในสภาวะปกติ และเอาต์พุตเป็นลูกคลื่นลอจิก 1 สลับกับ 0 อย่างต่อเนื่อง (pulse) เมื่อตรวจจับความเคลื่อนไหวได้

ในโครงงานนี้ได้เลือกใช้สัญญาณเอาต์พุตในโหมด H

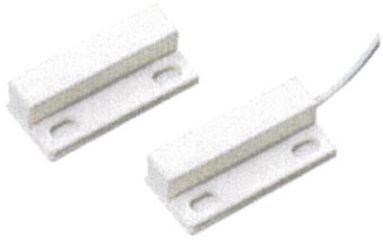
จุดเชื่อมต่อการใช้งานของ โมดูลตรวจความเคลื่อนไหว KC7783R แสดงดังรูปที่ 2.19



รูปที่ 2.19 จุดเชื่อมต่อการใช้งานของ โมดูลตรวจความเคลื่อนไหว KC7783R [4]

จากรูปที่ 2.19 จุดเชื่อมต่อสำหรับใช้งานมีทั้งหมด 3 จุด
- ขาไฟเลี้ยง (+) สำหรับต่อไฟเลี้ยงแรงดัน +5 โวลต์
- ขาเอาต์พุต (OUT) สำหรับต่อเข้ากับขาอินพุตของไมโครคอนโทรเลอร์
- ขากราวด์ (-) สำหรับต่อกราวด์ 0 โวลต์

2.4 อุปกรณ์แถบแม่เหล็ก



รูปที่ 2.20 แถบแม่เหล็ก

จากรูปที่ 2.20 แสดงแถบแม่เหล็กที่ใช้ในการทำโครงงาน เป็นอุปกรณ์ที่ใช้ร่วมกับวงจรรักษาความปลอดภัย ทำงานโดยต่อเข้ากับระบบเพื่อใช้กระแสไฟฟ้าไหลผ่านให้ครบวงจร ซึ่งจะแสดงผลโดยหลอด LED หากกระแสไฟฟ้าไหลผ่านครบวงจร หลอด LED จะติด หากแถบแม่เหล็กหลุดออกจากกันกระแสไฟฟ้าจะไหลไม่ครบวงจร หลอด LED ก็จะดับ จากนั้นระบบจะทำการส่งข้อความสั้น (SMS) แจ้งไปให้ผู้ใช้ทราบ

2.5 SD Card (Secure Digital Card)

หน่วยจำ SD Card นี้เป็นหน่วยความจำที่ได้รับความนิยมมากในปัจจุบันและยังเป็นอุปกรณ์ที่พกพาสะดวก อาจถือได้ว่า SD Card เป็นไมโครคอนโทรลเลอร์ตัวหนึ่งได้เลย เนื่องจาก SD Card นั้นประกอบด้วยชิปและอุปกรณ์ที่จำเป็นของไมโครคอนโทรเลอร์ สามารถถ่ายโอนข้อมูลได้ถึง 512B ต่อ Block สามารถนำไปประยุกต์ใช้งานต่างๆ ในการเก็บข้อมูลได้ โดยทั่วไประบบไฟล์ที่ใช้นิยมใช้ FAT 16 และ FAT 32 ขึ้นอยู่กับขนาดของหน่วยความจำที่ใช้ โดยในโครงงานนี้ได้เลือกใช้ ไมโคร Sd Card ที่มีขนาดความจุ 2 จิกะไบต์ ของบริษัท SanDisk

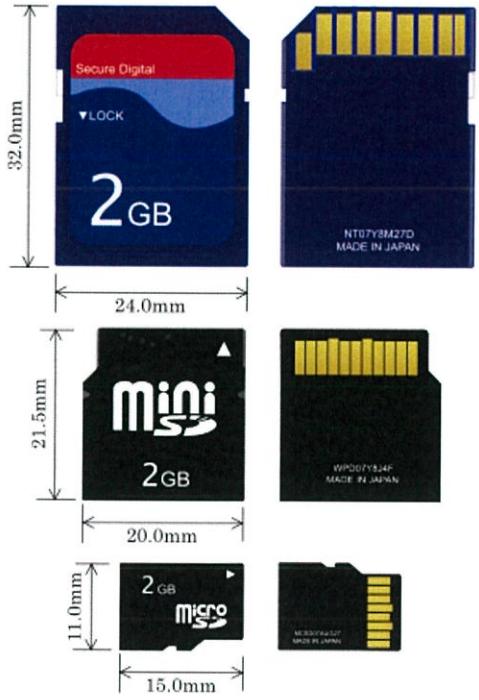
รูปที่ 2.21 ไมโคร SD Card [7]

2.5.1 คุณลักษณะของ SD Card

SD Card (Secure Digital Card) ถูกพัฒนาโดยบริษัท Matsushita, SanDisk และ Toshiba ในปี 2000 ในปัจจุบันถือว่าเป็นอุปกรณ์หน่วยความจำที่เป็นที่รู้จักและนิยมใช้กันอย่างกว้างขวาง SD Card ไม่ว่าจะเป็น กล้องดิจิตอล โทรศัพท์มือถือ SD Card นั้นสามารถแบ่งได้หลายชนิดซึ่งอาจจะใช้ ความจุ ขนาด หรือชนิดในการแบ่ง ลักษณะโดยทั่วไป SD Card จะมีด้วยกัน 3 แบบ ดังตารางที่ 2.21 และลักษณะรูปร่างของแต่ละชนิดแสดงดังรูปที่ 2.22

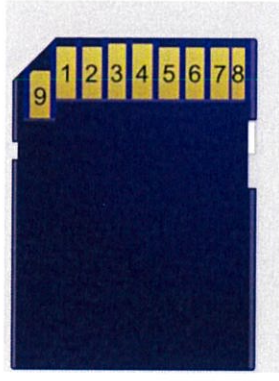
ตารางที่ 2.21 เปรียบเทียบรูปแบบ SD Card

คุณลักษณะ	SD	Mini SD	Micro SD
กว้าง (mm)	24	20	11
ยาว (mm)	32	21.5	15
หนา (mm)	2.1	1.4	1
น้ำหนัก (g)	2	1	0.5
ไฟเลี้ยง (V)	2.7-3.6	2.7-3..6	2.7-3.6
จำนวนขา	9	11	8
อัตราเร็ว (Mb/s)	15-20	15	15



รูปที่ 2.22 รูปแบบของ SD Card [7]

ขาของ SD Card มีทั้งหมด 9 ขา แสดงดังรูปที่ 2.23 โดยแต่ละขามีหน้าที่แตกต่างกันออกไปแสดงดังตารางที่ 2.22



รูปที่ 2.23 ขาของ SD Card [7]

ตารางที่ 2.22 แสดงหน้าที่แต่ละขาของ SD Card

หมายเลขขา	ชื่อ	SPI	อธิบาย
1	DAT3/CS	CS	Chip select
2	CMD/DI	MOSI	Data In
3	Vss1	GND	Ground
4	Vcc	VCC	Supply Voltage
5	CLK	SCLK	Clock
6	Vss2	GND	Ground
7	DAT0/DO	MISO	Data Out
8	DAT1	X	Reserved
9	DAT2	X	Reserved

2.5.2 การติดต่อสื่อสาร SD Card

ในการติดต่อสื่อสารหรือการอินเตอร์เฟส SD Card สามารถทำได้ 2 วิธี

1) SD Bus

2) SPI Bus

ในโครงงานนี้ใช้การติดต่อแบบ SPI เนื่องจากไมโครคอนโทรลเลอร์ส่วนใหญ่จะรองรับการสื่อสารแบบ SPI (Serial Peripheral Interface) ในการประยุกต์ใช้งานสามารถเชื่อมต่อกับ SD Card ได้โดยตรง

2.6 Infrared JPEG Camera

ในโครงงานนี้เลือกใช้กล้อง Infrared JPEG Camera รุ่น LS-Y201-IR-TTL ยี่ห้อ Linksprite Linksprite ดังรูปที่ 2.24 กล้อง Infrared JPEG Camera ควบคุมผ่านทาง UART (TTL) โดยกล้องจะมี Ambient light ในการปรับ Mode infrared อัตโนมัติ ทำให้กล้องสามารถถ่ายในที่มืดได้ ความละเอียดสูงสุด VGA (640x480) จากการทำงานของโมดูลซึ่ง รับ-ส่งข้อมูลผ่านทาง Serial interface ทำให้ สามารถนำไปใช้ งานในด้านต่างๆได้ อย่างสะดวก เช่น การนำไปต่อใช้งานกับ Microcontroller UART หรือใช้ งานกับ PC เป็นต้น

รูปที่ 2.24 Infrared JPEG Camera [8]

2.6.1 คุณสมบัติเบื้องต้น

- VGA (640*480)/ QVGA (320*240)/ 160*120 Resolution
- Frame Rate 640*480 => 30fps
- Default baud rate 38400
- ใช้แหล่งจ่ายไฟกระแสตรง 3.3V หรือ 5V

2.6.2 จุดเชื่อมต่อการใช้งาน Infrared JPEG Camera

Infrared JPEG Camera มีขาใช้งาน 4 ขา แสดงดังรูปที่ 2.25 และรายละเอียดแต่ละขาที่ใช้งานแสดงดังตารางที่ 2.25

รูปที่ 2.25 ขาที่ใช้งาน Infrared JPEG Camera [8]

ตารางที่ 2.23 การใช้งานขาของ Infrared JPEG Camera

สี	รายละเอียด
แดง	RX (TTL)
น้ำตาล	TX (TTL)
ม่วง	GND
เทา	5V DC

2.7 ET-GSM SIM300CZ

โครงงานเล่มนี้เลือกใช้ ET-GSM SIM300CZ แสดงดังรูปที่ 2.26 ซึ่งเป็นชุดเรียนรู้และพัฒนาระบบการสื่อสารไร้สาย โดยใช้โมดูล GSM/GPRS รุ่น SIM300CZ ของ "SIMCom Ltd." เป็นอุปกรณ์หลัก ซึ่ง SIM300CZ เป็นโมดูลสื่อสารระบบ GSM/GPRS ขนาดเล็ก รองรับระบบสื่อสาร GSM ความถี่ 900/1800/1900MHz โดยสั่งงานผ่านทางพอร์ตสื่อสารอนุกรม RS232 ด้วยชุดคำสั่ง AT Command สามารถประยุกต์ใช้งานได้มากมายหลายรูปแบบ ไม่ว่าจะเป็นการรับส่งสัญญาณแบบ Voice, SMS, Data, FAX และยังรวมถึงการสื่อสารด้วย Protocol TCP/IP ด้วย ซึ่งตามปรกติแล้ว ถึงแม้ว่าโมดูล SIM300CZ จะมีวงจร และ Firmware บรรจุไว้ภายในตัวเป็นที่เรียบร้อยแล้วก็ยังไม่สามารถนำไปใช้งานได้โดยตรงทันที เนื่องจากในการใช้งานจริงๆนั้น ผู้ใช้งานเองจำเป็นต้องออกแบบวงจรรอบนอกที่จำเป็นมาเชื่อมต่อกับขาสัญญาณของตัวโมดูลอีกในบางส่วน ไม่ว่าจะเป็นวงจรภาค Power Supply,วงจรเชื่อมต่อกับ SIM Card รวมไปถึงวงจร Line Driver ของ RS232 เป็นต้น

รูปที่ 2.26 ET-GSM SIM300CZ [10]

2.7.1 คุณสมบัติของบอร์ด ET-GSM SIM300CZ V1.0

- มีสวิตช์แบบ Push-Button สำหรับใช้สั่ง เปิด-ปิด การทำงานของโมดูลภายในบอร์ด

- มี Socket SIM รองรับ SIM Card พร้อมวงจร ESD ป้องกัน SIM เสียหาย

- มีวงจร Regulate แยกอิสระ จำนวน 2 ชุด สามารถใช้กับแหล่งจ่ายภาย Adapter ขนาดตั้งแต่ +5V ขึ้นไป สามารถจ่ายกระแสให้กับโมดูล SIM300CZ และอุปกรณ์เชื่อมต่อต่างๆได้อย่างเพียงพอ

- มีวงจร Regulate ขนาด 4.2V / 3A สำหรับจ่ายให้กับโมดูล SIM300CZ ได้อย่างเพียงพอ สามารถใช้กับ SIM ของระบบ GSM900MHz แบบ 2-Watt ได้อย่างไม่เกิดปัญหา

- มีวงจร Regulate ขนาด 3.3V / 1A สำหรับจ่ายให้กับวงจรเชื่อมต่อภายนอกโดยไม่ต้องไปดึงไฟจากตัวโมดูลมาใช้ ป้องกันปัญหาโมดูลเสียหายจากวงจรภายนอกดึงกระแสเกินพิกัดและสะดวกต่อการออกแบบวงจรเชื่อมต่อเพิ่มเติม โดยไม่ต้องกังวลว่ากระแสจะไม่พอจ่ายให้กับอุปกรณ์

- มีวงจร Line Driver สำหรับแปลงระดับสัญญาณโลจิกจากโมดูล SIM300CZ ให้เป็น RS232 ระดับมาตรฐานครบทุกเส้นสัญญาณ ทั้งพอร์ตที่ใช้ในการสื่อสารสำหรับสั่งงานโมดูล และ พอร์ตสำหรับใช้ในการพัฒนาโปรแกรม (Debug) สามารถเชื่อมต่อกับพอร์ต RS232 มาตรฐานได้ทันที

- มี LED แสดงสถานะพร้อมในบอร์ด สำหรับแสดงสถานะของแหล่งจ่ายไฟ สถานะพร้อมทำงานของโมดูล สถานะในการเชื่อมต่อกับ Network และ สถานะ Power-On/Power-OFF ของโมดูล

- มีขั้วสำหรับเชื่อมต่อกับ Handset (ชุดปากพูด และหูฟัง ของโทรศัพท์บ้าน) โดยใช้ขั้วต่อแบบ RJ11พร้อมวงจร Voice Filter สามารถนำชุด Handset ของโทรศัพท์บ้านต่อเข้ากับบอร์ดทางขั้วต่อแบบ RJ11 สำหรับใช้พูดคุย โทรออก และ รับสายได้โดยสะดวก

- มี Buzzer พร้อมวงจรขับเพื่อสร้างสัญญาณเสียง ในกรณีมีการโทรเรียกเข้ามายังโมดูล

- มีจุดยึดเสาอากาศ สำหรับใช้เป็นจุดพักสำหรับเชื่อมต่อกับเสาอากาศแบบต่างๆได้โดยสะดวก

- มีขั้วต่อสำหรับติดตั้งโมดูล SIM300CZ พร้อมเสารองและสกรูยึดโมดูลกับตัวบอร์ด

- มีจุดต่อสัญญาณอื่นๆที่เหลือจากโมดูล เช่น Keyboard, Display, GPIO, Battery Charger ฯลฯ

2.8 AT-COMMAND

AT-COMMAND คือ ชุดคำสั่งมาตรฐาน ที่สามารถใช้ติดต่อสื่อสารกับอุปกรณ์สื่อสารต่างๆ เช่น โมเด็ม หรือ อุปกรณ์ DTE (Data Terminal Equipment) เพื่อโต้ตอบตั้งค่าหรือสั่งงานอุปกรณ์เหล่านั้น ให้ทำงานตามที่ต้องการ และสำหรับการติดต่อกับโทรศัพท์มือถือ จะใช้ชุดคำสั่งที่เรียกว่า GSM AT COMMAND

โดยการเชื่อมต่อด้านฮาร์ดแวร์นั้นจะนิยมใช้ 2 ช่องทาง

1) Serial Port

2) IrDA (Infrared Port)

2.7.1 คำสั่ง AT Command กับ โทรศัพท์เคลื่อนที่

การสื่อสารกับอุปกรณ์สื่อสารต่างๆ เช่น โมเด็มหรืออุปกรณ์ DTE (Data Terminal Equipment) นั้นสามารถใช้ชุดคำสั่งที่เป็นมาตรฐานที่เรียกว่า AT Command ในการติดต่อเพื่อโต้ตอบตั้งค่าหรือสั่งอุปกรณ์เหล่านั้น ให้ทำงานตามที่ต้องการ โดยชุดคำสั่งพื้นฐานจะถูกกำหนดไว้ใน Hayes AT Command ซึ่งบริษัท Hayes เป็นผู้คิดค้นชุดคำสั่ง เพื่อใช้กับโมเด็มของตนและต่อมาได้กลายเป็นมาตรฐานสำหรับผู้ผลิตโมเด็มรายอื่นๆ โดยอาจจะมีชุดคำสั่งขยาย (Extended AT Command) เพื่อใช้เป็นการเฉพาะสำหรับผู้ผลิตรายนั้นๆ

การติดต่อกับมือถือก็เช่นกันสามารถใช้ชุดคำสั่งที่กำหนดไว้ใน GSM AT Command ซึ่งมีคำสั่งเพิ่มเติมที่เหมาะสมสำหรับการใช้งานและควบคุมมือถือ

2.9 มาตรฐานการสื่อสารระบบคอมพิวเตอร์

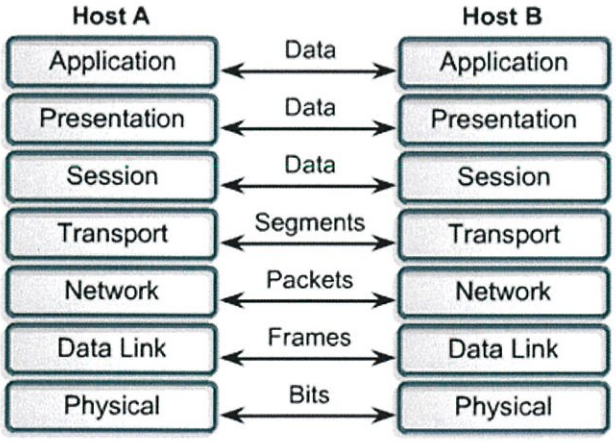
2.9.1 OSI Model (Open System Interconnection Model)

OSI Model เป็น medel มาตรฐานในการสื่อสารดังแสดงรูปที่ 2.27 ซึ่งมีวัตถุประสงค์ ใช้สำหรับการสื่อสารระหว่างระบบ 2 ระบบ ระบบจะเปิดการติดต่อสื่อสารในเค้าโครงสำหรับออกแบบ ระบบเครือข่าย จะอนุญาตให้สื่อสารข้ามทุกรูปแบบของระบบคอมพิวเตอร์แยก

เป็น 7 ชั้นแต่เกี่ยวเนื่องกันและเป็นรูปแบบมาตรฐาน ISOแบบจำลองนี้จะทำการจับกลุ่มรูปแบบฟังก์ชันการสื่อสารที่คล้ายกันให้อยู่ใน ชั้นใดชั้นหนึ่งในเจ็ดชั้นตรรกะ ชั้นใดๆจะให้บริการชั้นที่อยู่บนและตัวเองได้รับบริการจากชั้นที่อยู่ด้าน ล่าง ตัวอย่างเช่นชั้นที่ให้การสื่อสารที่ error-free ในเครือข่ายจะจัดหาเส้นทางที่จำเป็นสำหรับแอปพลิเคชันชั้นบน ในขณะที่มันเรียกชั้นต่ำลงไปให้ส่งและรับแพ็คเก็ตเพื่อสร้างเนื้อหาของเส้น ทางนั้น งานสองอย่างในเวลาเดียวกันที่ชั้นหนึ่งๆจะถูกเชื่อมต่อในแนวนอนบนชั้นนั้นๆ ตามรูปผู้ส่งข้อมูลจะดำเนินงานเริ่มจากชั้นที่ 7 จนถึงชั้นที่ 1 ส่งออกไปข้างนอกผ่านตัวกลางไปที่ผู้รับ ผู้รับก็จะดำเนินการจากชั้นที่ 1 ขึ้นไปจนถึงชั้นที่ 7 เพื่อให้ได้ข้อมูลอันนั้น

โมเดลนี้ได้ถูกแบ่งย่อยออกเป็น 7 ชั้น ตามลำดับ ได้แก่

1) Physical
2) Data Link
3) Network
4) Transportation
5) Session
6) Presentation
7) Application



รูปที่ 2.27 การส่งข้อมูลผ่านระหว่างชั้น [11]

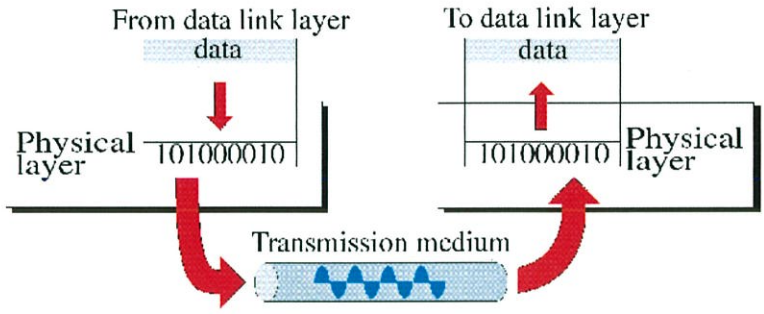
เหตุผลที่โมเดลนี้ถูกแบ่งออกเป็น 7 ชั้นดังรูปที่ 2.7 ก็เพื่อให้ง่ายต่อการทำความเข้าใจว่าแต่ละชั้นนั้นมีความสำคัญอย่างไร และสัมพันธ์กันอย่างไรระหว่างชั้น ซึ่งโดยหลักๆแล้วแต่ละชั้นจะมีความสัมพันธ์โดยตรงกับชั้นที่อยู่ติดกันกับ ชั้นนั้นๆ

2.9.1.1 Physical Layer

ชั้น Physical Layer แสดงดังรูปที่ 2.28 เป็นชั้นล่างที่สุดของการติดต่อสื่อสาร แต่เป็นชั้นแรกของสื่อที่ใช้ในการติดต่อสื่อสาร ทำหน้าที่ส่ง-รับข้อมูลจริง ๆ จากช่องทางการสื่อสาร (สื่อกลาง) ระหว่างคอมพิวเตอร์เครื่องหนึ่งกับคอมพิวเตอร์เครื่องอื่น ๆ มาตรฐานสำหรับ เลเยอร์ ชั้นนี้จะกำหนดว่าแต่ละคอนเนคเตอร์ (Connector) เช่น RS-232-C มีกี่พิน(pin) แต่ละพินทำหน้า ที่อะไรบ้าง ใช้สัญญาณไฟกี่โวลต์ เทคนิคการมัลติเพล็กซ์แบบต่างๆ ก็จะถูกกำหนดอยู่ในเลเยอร์ชั้นนี้ ซึ่งอาจจะเป็นทั้งแบบที่ใช้สายหรือไม่ใช้สาย ตัวอย่างของสื่อที่ใช้ได้แก่ Shield Twisted Pair (STP), Unshield Twisted Pair (UTP), Fibre Optic และอื่นๆ ชั้นกายภาพกำหนดคุณสมบัติทางกายภาพและทางไฟฟ้าสำหรับอุปกรณ์ โดยเฉพาะอย่างยิ่งจะกำหนดความสัมพันธ์ระหว่างอุปกรณ์และสื่อกลางในการส่ง ผ่านเช่นสายทองแดงหรือเส้นใยแก้วนำแสง ซึ่งรวมถึงรูปแบบของขา, แรงดันไฟฟ้า, ความต้านทานสาย, รายละเอียดเคเบิล, เวลาสัญญาณ, ฮับ, รีพีทเทอร์, อะแดปเตอร์เครือข่าย, อะแดปเตอร์บัสโฮสต์ (HBA ใช้ในเครือข่ายพื้นที่จัดเก็บ) และอื่น ๆ ฟังก์ชันที่สำคัญและการบริการที่ดำเนินการโดยชั้นกายภาพดังต่อไปนี้:

- เริ่มและจบการเชื่อมต่อการสื่อกลางการสื่อสาร
- การมีส่วนร่วมในกระบวนการ เพื่อใช้ทรัพยากรของการสื่อสารร่วมกันได้อย่างม

ประสิทธิภาพระหว่างผู้ใช้ ทั้งหลาย เช่นความพอใจ การแก้ปัญหาและการควบคุมการไหล การมอดดูเลทหรือการแปลงระหว่างข้อมูลดิจิตอลในอุปกรณ์ของผู้ใช้กับสัญญาณที่สอดคล้องกันเพื่อส่งผ่านช่องทางสื่อสาร สัญญาณเหล่านี้เดินทางในสายเคเบิลทางกายภาพ (เช่นทองแดงและใยแก้วนำแสง) หรือผ่านทางสัญญาณวิทยุบัส SCSI แบบขนานทำงานในชั้นนี้ แม้ว่า logical SCSI protocol เป็นโพรโทคอลของชั้นขนส่ง ที่วิ่งอยู่บนบัสนี้ มาตรฐานต่างๆของชั้นกายภาพของอีเธอร์เนท ก็อยู่ในชั้นนี้เช่นกัน; อีเธอร์เน็ตรวมชั้นนี้และชั้นเชื่อมโยงข้อมูลเข้าด้วยกัน วิธีนี้นำไปใช่เหมือนกันกับเครือข่ายท้องถิ่นพื้นที่อื่น ๆ เช่น token ring, FDDI, ITU-T G.hn และ IEEE 802.11 เช่นเดียวกับเครือข่ายพื้นที่ส่วนบุคคลเช่นบลูทูธ และ IEEE 802.15.4

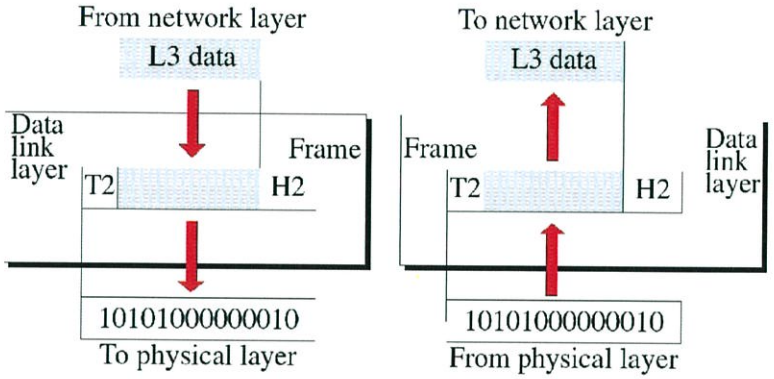


รูปที่ 2.28 Physical Layer [11]

2.9.1.2 Data Link Layer

ชั้น Data Link Layer แสดงดังรูปที่ 2.29 จะเป็นเสมือนผู้ตรวจสอบ หรือควบคุมความผิดพลาดในข้อมูลโดยจะแบ่งข้อมูลที่จะส่งออกเป็นแพ็กเกจหรือ เฟรม ถ้าผู้รับได้รับข้อมูลถูกต้องก็จะส่งสัญญาณยืนยันกลับมาว่า ได้รับข้อมูลแล้ว เรียกว่า สัญญาณ ACK (Acknowledge) ให้กับผู้ส่ง แต่ถ้าผู้ส่งไม่ได้รับสัญญาณ ACK หรือได้รับ สัญญาณ NAK (Negative Acknowledge) กลับมา ผู้ส่งก็อาจจะทำการส่งข้อมุลไปให้ใหม่ อีกหน้าที่หนึ่ง ของเลเยอร์ชั้นนี้คือป้องกันไม่ให้เครื่องส่งทำการส่งข้อมูลเร็วจนเกินขีด ความสามารถของเครื่องผู้รับจะรับข้อมูลได้ ชั้นเชื่อมโยงข้อมูลให้ฟังชั่นการทำงานและขั้นตอนการถ่ายโอนข้อมูล ระหว่างหน่วยงานเครือข่ายและการตรวจสอบและแก้ไขข้อผิดพลาดอาจเป็นไปได้ที่ อาจเกิดขึ้นในชั้นกายภาพ แต่เดิมชั้นนี้มีไว้สำหรับสื่อจุดต่อจุดและจุดต่อหลายจุดหรือลักษณะของสื่อ บริเวณกว้างในระบบโทรศัพท์ พื้นที่สถาปัตยกรรมของแลนซึ่ง รวมถึงสื่อการเข้าถึงที่สามารถการออกอากาศได้จะได้รับการพัฒนาต่างหากใน ISO IEEE 802.2 ในทางปฏิบัติที่ทันสมัย การตรวจสอบข้อผิดพลาดเท่านั้นที่ปรากฏในโพรโทคอลการเชื่อมโยงข้อมูลเช่น Point-to-Point Protocol (PPP) ไม่ใช่ flow control ที่ใช้ sliding window. และในเครือข่ายท้องถิ่น IEEE 802.2 ชั้น LLC ไม่ได้ถูกใช้เป็นโพรโทคอลส่วนใหญ่บน Ethernet และบนแลนอื่น ๆ flow control และกลไกการรับรู้ก็ไม่ค่อยมีการใช้ Sliding window flow control และ acknowledgment ถูกนำไปใช้ที่ชั้นการขนส่งโดยโพรโทคอลเช่น TCP แต่ยังคงใช้ในช้องว่างที่ X.25 เสนอข้อได้เปรียบด้านความสามารถ มาตรฐาน ITU-T G.hn ซึ่งให้บริการเครือข่ายพื้นที่ท้องถิ่นความเร็วสูงบนสายเคเบิลที่มีอยู่ (สายไฟฟ้า, สายโทรศัพท์และสาย coaxial) จะรวมถึงชั้นเชื่อมโยงข้อมูลอย่างสมบูรณ์ซึ่งจัดให้มีทั้งการแก้ไขข้อผิด พลาดและการควบคุมการไหลโดยใช้วิธีการคัดเลือก ซ้ำ ของ Sliding Window Protocol บริการของทั้งแวนและ แลน จัดวางบิตจากชั้นกายภาพให้เป็นลำดับทางตรรกะที่เรียกว่าเฟรม ไม่ใช่ทุกบิตในชั้น

กายภาพที่จำเป็นต้องใส่ไปลงในเฟรม เพราะบางส่วนของบิตเหล่านี้มีความตั้งใจที่จะใช้สำหรับฟังก์ชันในชั้นกายภาพ เท่านั้น ตัวอย่างเช่นทุกบิตที่ห้าของกระแสบิต FDDI ไม่ได้ถูกใช้โดยชั้นนี้



รูปที่ 2.29 Data Link Layer [11]

2.9.1.3 Network Layer

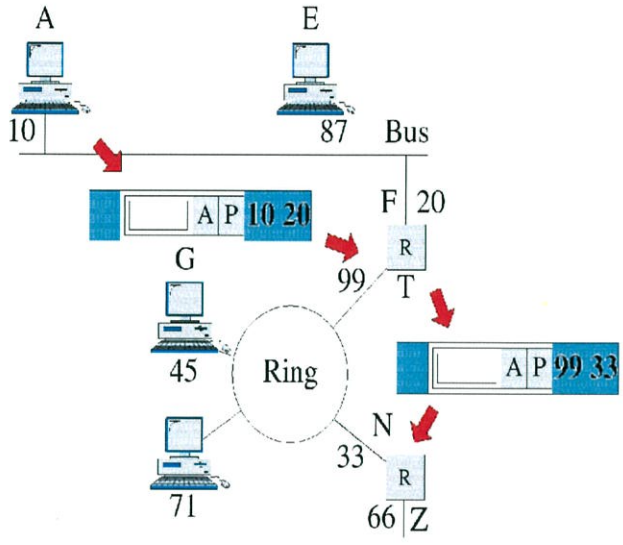
ชั้น Network Layer แสดงดังรูปที่ 2.30 เป็นชั้นที่ 3 เป็นชั้นที่ออกแบบหรือกำหนดเส้นทางการเดินทางของข้อมูลที่จะส่ง-รับในการส่ง ผ่านข้อมูลระหว่างต้นทางและปลายทาง ซึ่งแน่นอนว่าในการสื่อสารข้อมูลผ่านเครือข่ายการสื่อสารจะ ต้องมีเส้นทางการส่ง-รับข้อมูลมากกว่า 1 เส้นทาง ดังนั้นเลเยอร์ชั้น Network นี้จะทำหน้าที่เลือกเส้นทางที่ ใช้เวลาในการสื่อสารน้อยที่สุด และระยะทางสั้นที่สุดด้วย ข่าวสารที่รับมาจากเลเยอร์ชั้นที่ 4 จะถูกแบ่งออกเป็น แพ็กเกจ ๆ ในชั้นนี้ เลเยอร์เครือข่ายให้กรรมวิธีและขั้นตอนการถ่ายโอนข้อมูลความยาวแปรจาก โฮสต์ต้นทางบนเครือข่ายหนึ่งไปยังโฮสต์ปลายทางบนอีกเครือข่ายหนึ่งที่แตก ต่างกัน (ในทางตรงกันข้ามกับชั้น data link layer ที่เชื่อมต่อกับโฮสต์ภายในเครือข่ายเดียวกัน) ในขณะที่รักษาคุณภาพของการบริการที่ถูกร้องขอโดยชั้นของการขนส่ง เลเยอร์เครือข่ายทำหน้าที่หาเส้นทางและยังอาจดำเนินการแยกข้อมูลออกเป็นชิ้น เล็กๆและรวมกลับมาใหม่ และรายงานข้อผิดพลาดในการส่งข้อมูล เราเตอร์ทำงานอยู่ในชั้นนี้ จะการส่งข้อมูลไปทั่วเครือข่ายและทำให้อินเทอร์เน็ตเป็นไปได้ address ที่ใช้เป็นแบบลอจิคคอล -คือค่าจะถูกเลือกโดยวิศวกรเครือข่าย การใช้ address ไม่ได้เป็นแบบลำดับชั้นสูงต่ำ

เลเยอร์เครือข่ายอาจจะถูกแบ่งออกเป็นสาม sublayers:

1) Subnetwork access - จะพิจารณาโพรโทคอลที่จัดการกับอินเตอร์เฟซกับเครือข่ายเช่น X.25

2) Subnetwork dependent convergence - เมื่อจำเป็นที่จะต้องยกระดับของเครือข่ายการขนส่งไปที่ระดับของเครือข่ายข้างใดข้างหนึ่ง

3) Subnetwork independent convergence – จัดการการโอนข้ามหลายเครือข่าย



รูปที่ 2.30 Network Layer [11]

2.9.1.4 Transport Layer

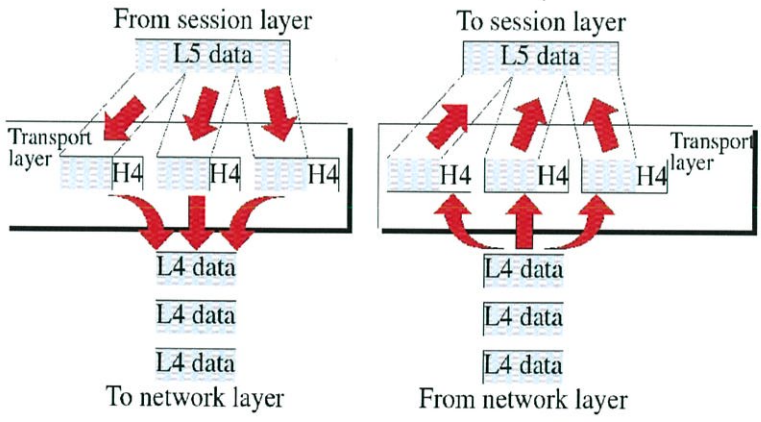
ชั้น Transport Layer แสดงดังรูปที่ 2.31 ทำหน้าที่ดูแลจัดการเรื่องของความผิดพลาดที่เกิดขึ้นจากการสื่อสาร ซึ่งการตรวจสอบความผิดพลาดนั้นจะพิจารณาจากข้อมูลส่วนที่เรียกว่า checksum และอาจมีการแก้ไขข้อผิดพลาดนั้นๆ โดยพิจารณาจาก ฝั่งต้นทางกับฝั่งปลายทาง (End-to-end) โดยหลักๆแล้วชั้นนี้จะอาศัยการพิจารณาจาก พอร์ต (Port) ของเครื่องต้นทางและปลายทาง เลเยอร์การขนส่งทำการโอนอย่างโปร่งใสของข้อมูลระหว่าง end users, ให้บริการการโอนข้อมูลที่เชื่อถือได้ไปยังเลเยอร์ชั้นบน, ควบคุมความน่าเชื่อถือของการเชื่อมโยงโดยการควบคุมการไหลของข้อมูล, การทำ segmentation/desegmentation, และควบคุมความผิดพลาด. บางโพรโทคอลเป็น state- and connection-oriented ซึ่งหมายความว่าชั้นของการขนส่งสามารถติดตาม segment และส่งอีกครั้งสำหรับ segment ที่ล้มเหลว ชั้นของการขนส่งยังมีการรับรู้ของการส่งผ่านข้อมูลที่ประสบความสำเร็จและส่ง ข้อมูลตัวต่อไปถ้าไม่มีข้อผิดพลาดเกิดขึ้น

วิธีที่ง่ายที่จะเห็นภาพชั้นการขนส่งก็คือการเปรียบเทียบกับที่ทำการไปรษณีย์ซึ่งเกี่ยวข้องกับการจัดส่งและการจัดหมวดหมู่ของจดหมายและพัสดุที่ ส่ง อย่างไรก็ตามพึงจำไว้ว่า ที่ทำการไปรษณีย์จัดการกับซองจดหมายที่อยู่ด้านนอก เลเยอร์ที่สูงขึ้นอาจจะเทียบเท่า

กับซองจดหมายสองซอง เช่นการบริการที่นำเสนอการเข้ารหัสลับที่สามารถอ่านได้โดยผู้รับเพียงอย่าง เดียว พูดอีกอย่างคือ โพรโทคอล tunneling ทำงานที่ชั้นการขนส่ง เช่นการดำเนินการที่เป็น Non-IP โพรโทคอล เช่น SNA ของไอบีเอ็ม หรือ IPX บนเครือข่ายไอพีของโนเวลล์หรือการเข้ารหัสแบบ end-to-end ด้วย IPsec ในขณะที่ Generic Routing Encapsulation (GRE) อาจดูเหมือนจะโพรโทคอลชั้นเครือข่าย, ถ้า encapsulation ของ payload ที่ใช้งานที่ปลายทาง, GRE กลายเป็นใกล้ชิดกับโพรโทคอลการขนส่งที่ใช้ IP headers แต่ประกอบด้วยเฟรมหรือแพ็กเกตเพื่อส่งมอบให้กับปลายทาง, L2TP ขนส่งเฟรม PPP ภายในแพ็คเก็ตการขนส่ง

แม้ว่าจะไม่ได้รับการพัฒนาภายใต้ OSI Model อ้างอิงและไม่เคร่งครัดในการสอดคล้องกับคำนิยามของ OSI ชั้นการขนส่ง, Transmission Control Protocol (TCP) และ User Datagram Protocol (UDP) ของชุด Internet Protocol ถูกจัดให้อยู่ในชั้น 4 ของ OSI

OSI กำหนดการกู้คืนเป็นห้าชั้นของโพรโทคอลการขนส่งแบบ connection-mode ตั้งแต่ class 0 (ซึ่งเป็นที่รู้จักกันว่าเป็น TP0 (Transport Protocol 0) และมีคุณสมบัติน้อยสุด) จนถึง class 4 (TP4 มีคุณสมบัติมากสุด, ได้รับการออกแบบสำหรับเครือข่ายที่เชื่อถือได้น้อย, คล้ายกับอินเทอร์เน็ต) ชั้น 0 ไม่มีการกู้คืนและถูกออกแบบมาสำหรับการใช้งานบน network layers ที่ให้การเชื่อมต่อปราศจากข้อผิดพลาด ชั้น 4 ใกล้เคียงกับ TCP ถึงแม้ว่า TCP มีฟังก์ชันเช่น graceful close ซึ่ง OSI กำหนดให้อยู่ใน session layer นอกจากนี้ทุก OSI TP connection-mode protocol classes ให้ข้อมูลเร่งด่วนและการเก็บรักษาของ record boundaries ลักษณะรายละเอียดของ class TP0-4

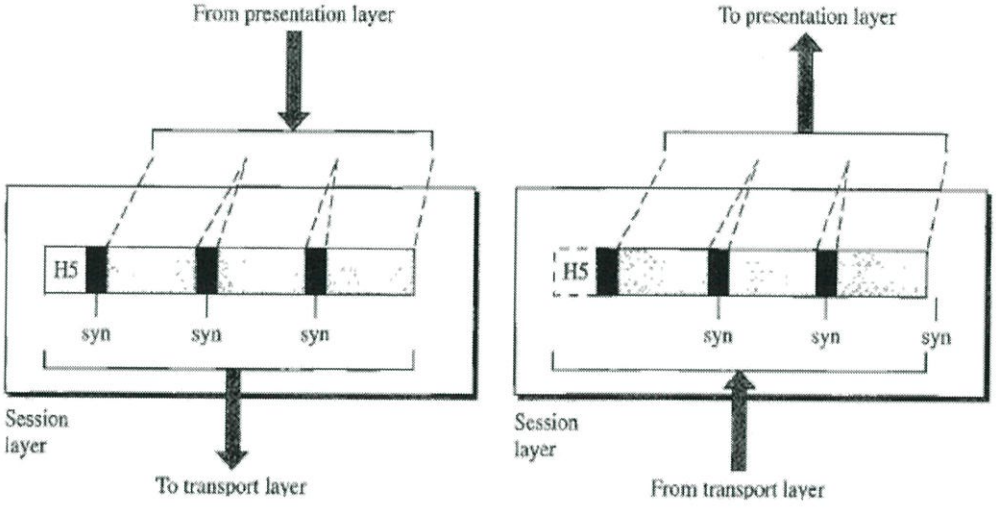


รูปที่ 2.31 Transport Layer [11]

2.9.1.5 Session Layer

ชั้น Session Layer แสดงดังรูปที่ 2.32 ทำหน้าที่ในการจัดการกับเซสชั่นของโปรแกรม ชั้นนี้เองที่ทำให้ในหนึ่งโปรแกรมยกตัวอย่างเช่น โปรแกรมค้นดูเว็บ (Web

browser) สามารถทำงานติดต่ออินเทอร์เน็ตได้พร้อมๆกันหลายหน้าต่าง เลเยอร์เซสชั่นจะควบคุมการหารือ (การเชื่อมต่อ) ระหว่างคอมพิวเตอร์หลายตัว มันจัดทำ, บริหารจัดการ, และยุติการเชื่อมต่อการใช้งานระหว่างภายในกับระยะไกล มันทำงานแบบดูเพล็กเต็ม,ดูเพล็กซ์ครึ่ง หรือ ซิมเพล็กซ์และจัดทำวิธีการตรวจสอบ, การเสร็จสิ้น, การเลิกดำเนินการ และการเริ่มต้นใหม่แบบจำลอง OSI ทำให้เลเยอร์นี้ทำหน้าที่รับผิดชอบในการปิดเซสชั่นอย่างสง่างาม ซึ่งเป็นทรัพย์สินอย่างหนึ่งของ Transmission Control Protocol และยังรับผิดชอบสำหรับเซสชั่น checkpointing and recovery ที่ไม่ค่อยได้ถูกนำมาใช้ใน Internet Protocol สวีท เลเยอร์เซสชั่นปกติจะดำเนินการร่วมกันอย่างชัดเจนกับแอปพลิเคชันในสภาพแวดล้อมที่เรียกกระบวนการจากระยะไกล

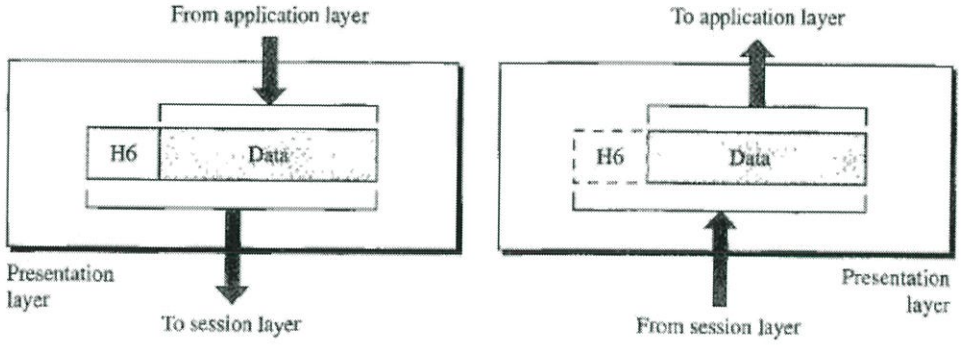


รูปที่ 2.32 Session Layer [11]

2.9.1.6 Presentation Layer

ชั้น Presentation Layer แสดงดังรูปที่ 3.33 เป็นชั้นที่รับผิดชอบเรื่องรูปแบบของการแสดงผลเพื่อโปรแกรมต่างๆที่ใช้งาน ระบบเครือข่ายทำให้ทราบว่าข้อมูลที่ได้เป็นประเภทใด เช่น [รูปภาพ, เอกสาร, ไฟล์วีดีโอ] ชั้นของการนำเสนอจะจัดทำอรรถาธิบายระหว่าง entities ในชั้นแอปพลิเคชัน ในที่ซึ่ง entities ในเลเยอร์ที่สูงกว่าอาจใช้ไวยากรณ์และกฎเกณฑ์ในแอปพลิเคชันที่แตกต่างกัน ชั้นการนำเสนอจึงต้องจัดหาการ mapping ระหว่าง entities เหล่านั้น ถ้าการทำ mapping มีอยู่ ข้อมูลจะถูก encapsulate ให้อยู่ในรูปโพรโทคอลของเลเยอร์เซสชันแล้วส่งต่อลงไปในชั้นต่อไป ชั้นการนำเสนอนี้เป็นอิสระจากการแสดงข้อมูล (เช่นการเข้ารหัส) โดยการแปลระหว่างรูปแบบของแอปพลิเคชันเลเยอร์และรูปแบบของเครือข่ายเลเยอร์ ชั้นการนำเสนอจะแปลงข้อมูลให้อยู่ในรูปแบบที่แอปพลิเคชันเลเยอร์ยอมรับ เลเยอร์นี้จะ format และเข้ารหัสข้อมูลที่จะส่งไปในเครือข่าย บางครั้งถูกเรียกว่าชั้นไวยากรณ์ (syntax layer)

โครงสร้างการนำเสนอเดิมใช้ syntax แบบ Basic Encoding Rules of Abstract Syntax Notation One(ASN.1) ด้วยความสามารถเช่นการแปลงไฟล์ข้อความแบบ EBCDIC ให้เป็นแฟ้มแบบ ASCII หรือการแปลงโครงสร้างแบบอนุกรมของวัตถุหรือโครงสร้างข้อมูลอื่น ๆ ให้เป็นแบบ XML หรือกลับกัน



รูปที่ 2.33 Presentation Layer [11]

2.9.1.7 Application Layer

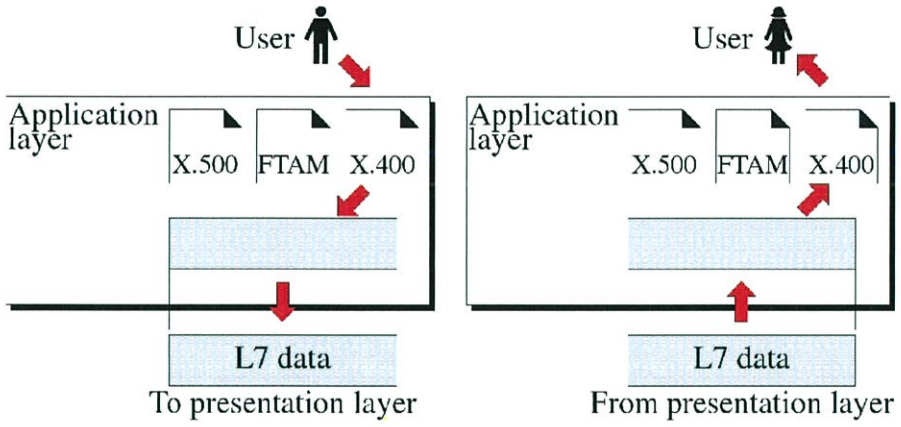
ชั้น Application Layer แสดงดังรูปที่ 2.34 เป็นชั้นที่อยู่ใกล้ผู้ใช้มากที่สุดโดยเป็นชั้นแอปพลิเคชันของ OSI มีปฏิสัมพันธ์กันโดยตรงกับผู้ใช้ด้วยซอร์ฟแวร์แอปพลิเคชัน ฟังก์ชันของชั้นนี้จะรวมถึงการระบุคู่ค้าการสื่อสาร โดยพิจารณาตัวตนและความพร้อมของคู่ค้าสำหรับการประยุกต์ใช้กับข้อมูลที่จะ ส่ง เมื่อพิจาณาถึงความพร้อมของทรัพยากร, แอปพลิเคชันเลเยอร์จะต้องตัดสินใจว่ามีเครื่อข่ายเพียงพอหรือมีเครือข่ายที่ ได้ร้องขอไปอยู่แล้วหรือไม่ ในการสื่อสารให้ตรงกัน, ทุกการสื่อสารระหว่างแอปพลิเคชันทั้งหมดต้องการความร่วมมือที่จะถูกบริหารจัดการโดยแอปพลิเคชันเลเยอร์นี้ ตัวอย่างบางส่วนของการใช้งานแอปพลิเคชันเลเยอร์ ได้แก่

บน OSI stack:

- File Transfer and Access Management Protocol (FTAM)
- X.400 Mail
- Common Management Information Protocol (CMIP)

บน TCP/IP stack:

- Hypertext Transfer Protocol (HTTP),
- File Transfer Protocol (FTP),
- Simple Mail Transfer Protocol (SMTP)
- Simple Network Management Protocol (SNMP)



รูปที่ 2.34 Application Layer [11]

2.10 FTP (File Transfer Protocol)

โปรโตคอลมาตรฐานในอินเตอร์เน็ต ซึ่งเป็นวิธีที่ง่ายที่สุดในการแลกเปลี่ยนไฟล์ระหว่างคอมพิวเตอร์บนอินเตอร์เน็ต คล้ายกับ Hypertext Transfer Protocol (HTTP) ที่ใช้ในการส่งเว็บเพ็จและไฟล์ที่เกี่ยวข้อง และ Simple Mail Transfer Protocol (SMPT) ที่ใช้ส่งผ่าน e-mail ซึ่ง FTP เป็นโปรโตคอลประยุกต์ที่ใช้โปรโตคอล TCP/IP โดย FTP ใช้ในการส่งไฟล์เว็บเพจจากแหล่งที่เก็บไปยังเครื่องคอมพิวเตอร์ โดยแสดงฐานะเป็นเครื่องแม่ข่าย สำหรับทุกคนบนอินเตอร์เน็ตและนิยมใช้ในการ download โปรแกรมและไฟล์มายังเครื่องคอมพิวเตอร์จากเครื่องแม่ข่ายอื่น ใน ฐานะผู้ใช้ การใช้ FTP สามารถใช้คำสั่งติดต่อแบบ command line (เช่นเดียวกับคำสั่งของ MS_DOS) web browser สามารถสร้างคำขอ FTP เพื่อ download โปรแกรมที่เลือกจากเว็บเพจ นอกจากนี้ FTP สามารถใช้ปรับปรุงไฟล์บนเครื่องแม่ข่าย โดยต้อง logon ไปที่ FTP server การสนับสนุน FTP โดยพื้นฐานในฐานะที่เป็นส่วนหนึ่งโปรแกรมที่มากับ TCP/IP อย่างไรก็ตามโปรแกรม FTP แบบ client ที่มีการติดต่อแบบ GUI ต้อง download จากบริษัทที่ผลิต

2.10.1 วิธีการทำงานของ FTP

FTP จะทำงานในแบบไคลเอนต์เซิร์ฟเวอร์ โดยพัฒนาขึ้นตามโปรโตคอลพื้นฐาน TCP ซึ่งจะต้องมีการติดต่อเพื่อจองช่องสื่อสาร (Connection Establishment) ก่อนทำการสื่อสารจริง ซึ่งเรียกว่าเป็นการติดต่อแบบที่ต้องขอเชื่อมต่อก่อน (Connection - oriented) ในการใช้งาน FTP เพื่อเริ่มการติดต่อสื่อสารนั้น จะต้องระบุหมายเลข IP ปลายทางและต้องผ่านการแจ้งรหัส Login และ Password ของเซิร์ฟเวอร์ที่จะติดต่อก่อนจึงจะเข้าใช้งานได้

ข้อมูลของ FTP ที่สื่อสารระหว่างกันมี 2 ประเภทคือ

2.10.1.1 ข้อมูล (data) หมายถึงข้อมูลต่าง ๆ ที่ต้องการรับส่ง รวมทั้งไฟล์ที่รับมาจากเซิร์ฟเวอร์ หรือส่งมาจากไคลเอนต์แล้วไปเก็บไว้ที่เซิร์ฟเวอร์ก็ได้

2.10.1.2 ข้อมูลที่เป็นคำสั่ง (command) FTP จะมีคำสั่งที่ใช้สั่งงานต่าง ๆ เช่น dir เป็นคำสั่งที่ใช้แสดงชื่อไฟล์หรือไดเรคทอรีในเครื่องเซิร์ฟเวอร์ หรือ get ใช้โหลดไฟล์มายังเครื่องไคลเอนต์ เป็นต้น โดยผู้ใช้จะสั่งงานที่ไคลเอนต์ผ่านโปรแกรม FTP

2.10.2 ประเภทข้อมูล (Data Type)

คุณสมบัติที่สำคัญของ FTP คือความสามารถในการแปลงข้อมูลให้ถูกต้องตามความเหมาะสม มีประโยชน์กรณีที่เครื่องเซิร์ฟเวอร์และไคลเอนต์เป็นคอมพิวเตอร์ที่ต่างระบบกัน ซึ่ง FTP ก็จะแปลงข้อมูลให้ถูกต้องตามความเหมาะสมได้โดยที่ FTP จะมีคำสั่ง TYPE เพื่อกำหนดประเภทของข้อมูล ซึ่งประกอบด้วย

2.10.2.1 NVT - ASSCII (Network Virtual Terminal ASSCII) เป็นข้อมูลทั่ว ๆ ไปที่ใช้รหัส ASSCII

2.10.2.2 EBCDIC เป็นข้อมูลที่ใช้ในเครื่องคอมพิวเตอร์เมนเฟรมไอบีเอ็ม

2.10.2.3 IMAGE เป็นข้อมูลที่เป็นไบนารี อาจจะเป็นโปรแกรม หรือรูปภาพ โดยมีข้อมูลเป็น 8 บิต

2.10.2.4 LOCAL เป็นข้อมูลที่เป็นไบนารีเช่นเดียวดับ IMAGE แต่จำนวนบิตต่อหนึ่งไบต์อาจจะแตกต่างกันไปตามประเภทของเครื่องคอมพิวเตอร์

2.10.3 วิธีการรับ – ส่ง

2.10.3.1 FTP กำหนดวิธีการรับส่งข้อมูลได้ดังนี้

Stream Mode เป็นวิธีการที่จะรับส่งข้อมูลเรียงลำดับไบต์ส่งต่อกันไปเรื่อย ๆ ดังนั้นการรับส่งวิธีนี้จึงสามารถใช้ได้กับไฟล์ทุกประเภท ส่วนไฟล์ที่มีโครงสร้างต้องมีรหัสตัวอักษรพิเศษที่กำหนดการสิ้นสุดเรคอร์ด (End of Record หรือ EOR) และจบไฟล์ (End of File หรือ EOF) ด้วย ซึ่งการรับส่งข้อมูลจะสิ้นสุดโดยตรวจสอบจากค่าของ EOF นี้

2.10.3.2 Error Recovery ในการรับส่งไฟล์แบบ Stream Mode การแก้ไขกรณีข้อมูลสูญหายระหว่างการรับส่ง จะใช้ความสารถของโปรโตคอล TCP แต่จะไม่มีการส่งข้อมูลซ้ำ แต้ถ้าหากเป็นแบบ Block Mode หรือ Compressed Mode ข้อมูลสามารถส่งใหม่เฉพาะส่วนที่ตรวจสอบพบข้อผิดพลาดได้

2.10.3.3 Anonymous FTP เซิร์ฟเวอร์ของ FTP บางแห่งจะทำหน้าที่ให้บริการแก่ผู้ใช้ทั่วไปเพื่อดาวน์โหลดไฟล์ ไม่ว่าจะเป็นข้อมูล รูปภาพ หรือโปรแกรมได้อย่างเสรี ซึ่งเซิร์ฟเวอร์เหล่านั้นจะมีรหัสผู้ใช้กลาง ๆ ที่ยอมให้ผู้ใช้สามารถเข้าระบบไปใช้งานเซิร์ฟเวอร์ FTP ได้ทุกคน ส่วนชื่อกลาง ๆ ในการเข้าสู่ระบบที่ส่วนใหญ่จะตั้งให้คือ anonymous (หมายถึง "นิรนาม" คือ ไม่ระบุชื่อ) ดังนั้นผู้ใช้เพียงแต่ทราบชื่อของเซิร์ฟเวอร์ ชื่อไฟล์ และชื่อไดเร็คทอรีที่ต้องการ แล้วเข้าระบบด้วยชื่อ anonymous ก็มารถจะดาวน์โหลดไฟล์ที่ต้องการได้

2.10.3.4 TFTP (Trivial File Transfer Protocal) TFTP เป็นกระบวนการรับส่งไฟล์ที่เรียบง่ายกว่า FTP ทั่วไป โดยใช้กลไกการสื่อสารแบบ UDP (User Datagram Protocal) ซึ่งเป็นโปรโตคอลที่ทำงานแบบ Connectionless ซึ่งผู้ใช้ไม่จำเป็นต้องใส่รหัสหรือ Password แต่จะทำได้เพียงโอนข้อมูลที่จัดเตรียมไว้แล้วเท่านั้น แต่จะไม่มีฟังก์ชันอื่น ๆ เช่น การแสดงรายชื่อไฟล์ การเปลี่ยนไดเร็คทอรี เป็นต้น

2.10.3.5 Connectionless และ Connection- oriented หลักการรับส่งข้อมูล 2 แบบที่เรียกว่า Connectionless และ Connection -oriented นี้มีความแตกต่างกันโดยที่ Connectionless เป็นการส่งข้อมูลโดยไม่สนใจว่าผู้รับปลายทางจะได้รับข้อมูลหรือไม่ เปรียบได้กับว่าการส่งไปรษณีย์แบบธรรมดา ที่ต้องนำจดหมายไปหย่อนลงตู้ โดยผู้จะไม่ทราบเลยว่าผู้รับได้รับจดหมายหรือไม่ ถ้าผู้รับไม่ตอบจดหมายกลับ ส่วน Connection- oriented จะเป็นการสื่อสารที่มีกลไกที่ทำให้ผู้ส่งทราบว่าผู้รับได้รับข้อมูลต่าง ๆ หรือไม่ ซึ่งถ้าไม่ได้รับก็จะต้องส่งไปใหม่ ก็จะคล้ายกับการส่งไปรษณีย์ลงทะเบียนนั่นเอง

2.10.4 ระบบเสริมอื่นๆในการส่งไฟล์ข้ามเครื่อง

ในการรับส่งไฟล์ด้วยโปรแกรมที่ใช้โปรโตคอล FTP โดยทั่ว ๆ ไปนั้น มักจะประสบปัญหาในกรณีที่เป็นไฟล์ขนาดใหญ่ ๆ และต้องหยุดการรับส่งกลางคัน เช่น สายเชื่อมต่ออินเตอร์เน็ตหลุดในขณะที่รับส่งไฟล์ยังไม่เรียบร้อย ทำให้ต้องย้อนไปเริ่มติดต่อเข้าอินเตอร์เน็ตและเริ่มรับส่งไฟล์ใหม่ตั้งแต่ต้น ถึงแม้ในปัจจุบันระบบสื่อสารและโมเด็มมีความเร็วสูงมากขึ้นก็ตาม แต่เหตุการณ์ลักษณะนี้ก็มีโอกาสเกิดขึ้นได้อยู่ตลอดเวลา ดังนั้นจึงมีซอร์ฟแวร์ที่พัฒนาขึ้นเพื่อให้รับข้อมูลส่วนที่ต่อจากข้อมูลเดิมที่ยังไม่ครบถ้วนได้ ซึ่งมีทั้งที่ใช้โปรโตคอล FTP เช่น GetRight และโปรโตคอลใหม่ เช่น WepNFS WepDAV เป็นต้น

2.10.4.1 GetRight

เป็นซอฟต์แวร์ที่ทำงานร่วมกับบราวเซอร์ โดยปกติเมื่อโปรแกรมบราวเซอร์จะดาวน์โหลด จะแสดงไดอะล็อกซ์บ็อกซ์ให้เลือกไดเร็คทอรีที่จะเก็บไฟล์หลังจากดาวน์

โหลด ซึ่งเมื่อติดตั้งโปรแกรม GetRight เรียบร้อยแล้ว มันก็จะเป็นอีกโปรแกรมหนึ่งแฝงตัวอยู่ใน Startup และเมื่อบราวเซอร์มีการใช้โปรโตคอล FTP เพื่อรับส่งข้อมูล โปรแกรม GetRight ก็จะถูกเรียกขึ้นมาทำงานโดยอัตโนมัติ โดยจะมาแทนที่ไดอะล็อกบ็อกซ์ปกติของบราวเซอร์ โดยจะบันทึกชื่อ URL ที่ดาวน์โหลดไฟล์นั้น ๆ และชื่อไฟล์เอาไว้ หากมีปัญหาเกิดขึ้นและทำให้การดาวน์โหลดต้องหยุดไปกลางคัน เมื่อติดต่อเข้าสู่อินเตอร์เน็ตใหม่ GetRight ก็จะสามารถตรวจสอบได้ว่าไฟล์ที่ดาวน์โหลดค้างอยู่ในไฟล์ใด และจะดาวน์โหลดไฟล์เฉพาะส่วนที่ยังขาดอยู่ต่อไปจนเสร็จสมบูรณ์ได้

การทำงานของ GetRight จะอำนวยความสะดวกให้แก่ผู้ใช้งานได้มาก โดยถ้ากำหนดชื่อของ Dail-up Networking ที่ใช้งานไว้ โปรแกรม GetRight ก็จะติดต่อเข้าสู่อินเตอร์เน็ตให้ใหม่โดยอัตโนมัติ หลังจากที่เกิดขัดข้องในการดาวน์โหลดครั้งก่อนหน้านี้รวมทั้งสามารถกำหนดได้ว่า จะให้วางสายโทรศัพท์หรือปิดเครื่องคอมพิวเตอร์โดยอัตโนมัติเมื่อดาวน์โหลดเรียบร้อยแล้ว จึงเหมาะสำหรับดาวน์โหลดไฟล์ขนาดใหญ่ที่ต้องใช้เวลานาน โดยไม่ต้องเฝ้าอยู่ตลอดเวลา รวมทั้งสามารถลดความเร็วในการดาวน์โหลดได้ ถ้าหากต้องการเซร์สายโทรศัพท์กับงานอื่น ๆ เช่น ต้องการเข้าไปที่เว็บไซต์พร้อม ๆ กับดาวน์โหลดข้อมูล เป็นต้น

2.10.4.2 WepNFS

พัฒนาต่อเนื่องมาจากเทคโนโลยีในการแชร์ไฟล์ข้ามเครื่องบนระบบปฏิบัติการ Unix คือ NFS (Network File System) และระบบการเรียกใช้โปรแกรมข้ามเครื่องที่เรียกว่า RPC (Remote Procedure Call) ของบริษัท Sun Microsystems ซึ่งมีคุณสมบัติที่สามารถจำลองดิสก์ของระบบ Unix ให้เครื่องอื่นที่ต่อผ่านเน็ตเวิร์กเข้ามาเรียกใช้ข้อมูล หรือส่งงานมาพิมพ์ และในทางกลับกันก็ขอใช้ไฟล์หรือส่งงานไปพิมพ์ที่เซิร์ฟเวอร์อื่นในเน็ตเวิร์กได้ด้วย โดยขยายให้สามารถแชร์ไฟล์ผ่านอินเตอร์เน็ตได้

2.11 XAMPP

เป็นโปรแกรม Apache web server ไว้จำลอง web server เพื่อไว้ทดสอบ สคริปหรือเว็บไซต์ในเครื่องของเรา โดยที่ไม่ต้องเชื่อมต่ออินเตอร์เน็ตและไม่ต้องมีค่าใช้จ่ายใดๆ ง่ายต่อการติดตั้งและใช้งานโปรแกรม Xampp จะมาพร้อมกับ PHP ภาษาสำหรับพัฒนาเว็บแอพลิเคชั่นที่เป็นที่นิยม , MySQL ฐานข้อมูล, Apache จะทำหน้าที่เป็นเว็บ เซิร์ฟเวอร์, Perl อีกทั้งยังมาพร้อมกับ OpenSSL , phpMyadmin (ระบบบริหารฐานข้อมูลที่พัฒนาโดย PHP เพื่อใช้เชื่อมต่อไปยังฐานข้อมูล สนับสนุนฐานข้อมูล MySQL และ SQLite โปรแกรม Xampp จะอยู่ในรูปแบบของไฟล์ Zip, tar, 7z หรือ exe โปรแกรม Xampp อยู่ภายใต้ใบอนุญาตของ GNU General Public License แต่บางครั้งอาจจะมีการเปลี่ยนแปลงเรื่องของลิขสิทธิ์ในการใช้งาน จึงควรติดตามและตรวจสอบโปรแกรมด้วย

2.11.1 ขั้นเตรียมพร้อมก่อนติดตั้ง XAMPP

2.11.1.1 ก่อนติดตั้ง XAMPP ควรอย่างยิ่งที่จะทำการ Uninstall ระบบจำลอง Webserver ตัวอื่นๆ ในระบบ Windows เราเสียก่อน เช่น Appserv / IIS เป็นต้น

2.11.1.2 RAM ไม่ต่ำกว่า 128 MB HardDisk มีเนื้อที่ว่างมากกว่า 320 MB ส่วน CPU ไม่กำหนดระบบขั้นต่ำ

2.11.1.3 เครื่องที่จะทำการติดตั้ง ลงระบบปฏิบัติการเรียบร้อย Windows 2000, XP (Server 2003), Vista (Server 2008), Windows 7 แบบ 32 bit หรือ 64 bit (อันที่จริง XAMPP ลองรับทั้ง Windows Linux MacOS และ Solaris นะครับแต่ผมจะขอเน้นไปที่ Windows เพียงอย่างเดียว)

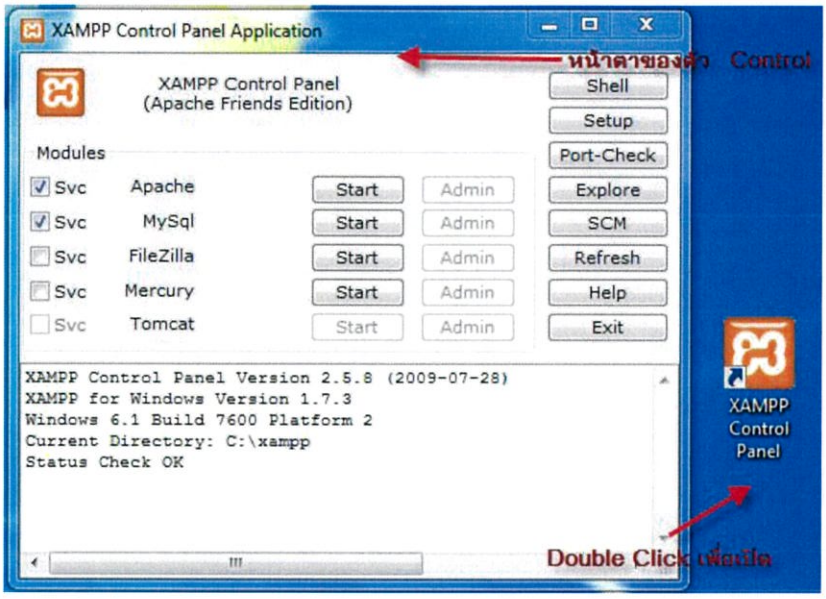
2.11.1.4 Download ชุดติดตั้งจาก http://www.apachefriends.org/en/xampp-windows.html (เฉพาะ Windows) ซึ่งมีให้โหลดหลายชุด ทั้งแบบ Full ขนาดประมาณ 50 mb ซึ่งจะรวม Sever อื่นๆเข้าไปด้วย เช่น Fileserver และ mailserver เป็นต้น และแบบ Lite ขนาดประมาณ 20 mb ซึ่งจะประกอบด้วย Webserver และ Database Sever

2.11.2 การติดตั้ง

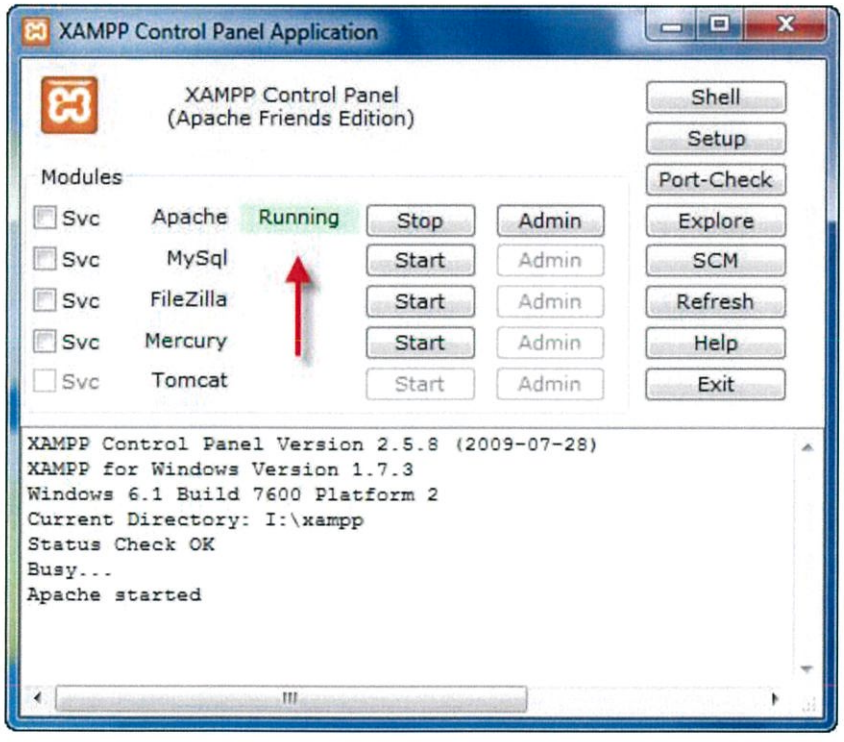
เปิดไฟล์ที่ใช้ในการติดตั้ง xampp-win32-1.7.3.exe (โหลดตัว Full ถ้า DL รุ่นอื่นชื่ออาจจะเปลี่ยนแปลงไปจากนี้เล็กน้อย)

2.11.3 การใช้งานเบื้องต้น

หลังจากการติดตั้งและกำหนดค่าเรียบร้อยแล้ว ที่ จะมี shortcup ที่เป็นตัวควบคุมของ XAMPP ปรากฏอยู่กดเข้าไปในโปรแกรมดังรูปที่ 2.35 และ Control ตัวนี้ถือเป็นจุดเด่นของ XAMPP ก็ว่าได้ Apache ซึ่งเป็น Webserver ถูกกำหนดให้เปิดทุกครั้งที่ Start Windows และ MySql ก็เช่นเดียวกัน แต่ขณะนี้ยังอยู่ในสถานะ Stop คือยังไม่ทำงาน ถ้าเรากดที่ปุ่ม Start ดังรูปที่ 2.36 ก็จะทำงานทันที หากเราไม่ต้องการให้ sever ทำงานทุกครั้งที่เปิดเครื่องก็เอา เครื่องหมายถูกที่ Svc ด้านหน้าของ service ที่ต้องการออก



รูปที่ 2.35 XAMPP Control Panel [14]



รูปที่ 2.36 XAMPP Control Panel กดปุ่ม start เพื่อให้ Apache ทำงาน [14]

2.12 Forward Port

Port forwarding คือ การกำหนดเส้นทางของข้อมูลสำหรับติดต่อกับเครื่องคอมพิวเตอร์ที่อยู่ในระบบ LAN ซึ่งเส้นทางนี้เองทำให้ข้อมูลจากภายนอกหรือจากระบบ Internet สามารถสื่อสารและส่งข้อมูลเข้ามายังอุปกรณ์ต่างๆในระบบ ที่มีการระบุค่าของ Port ไว้บน Router ซึ่งในการทำดังกล่าวจะอาศัยการทำงานร่วมกับการทำ DDNS ด้วยทุกอุปกรณ์ที่ใช้เชื่อมต่ออินเทอร์เน็ต จะถูกกำหนดอย่างน้อยที่สุดหนึ่ง

- แต่ละไอพีสามารถเลือกใช้พอร์ต (port) ใดๆพอร์ตหนึ่ง เพื่อ รับ-ส่ง ข้อมูล
- ไอพีหนึ่ง ไม่สามารถเลือกใช้พอร์ตซ้ำ หรือพอร์ตที่ถูกไอพีอื่นเลือกใช้อยู่ก่อนในคอมพิวเตอร์เครื่องเดียวกัน
- NAT (Network Address Translation) สำหรับเครือข่ายใช้หนึ่งไอพีอ้างอิงตำแหน่ง เพื่อจัดการกับตำแหน่งที่อยู่ไอพีอื่นๆจำนวนมากมาย

สัญญาณ ADSL จะส่งมาเข้าที่ Router โดยมี External IP เปลี่ยนแปลงตามการเชื่อมต่อแต่ละครั้ง และ External IP นี้ บอกให้ทราบถึงที่อยู่ของเรา ไว้เพื่อติดต่อกับ External IP ของผู้อื่นที่ห่างไกล

2.12.1 Internal IP

Router จะสร้าง Internal IP ออกมา 2 ชุด ดังนี้

2.11.1.1 ชุดแรกหรือ Internal IP1 มีชื่อเรียกโดยทั่วไปว่า Gateway เป็น Fix IP (ไอพีเปลี่ยนแปลงตามบริษัทผู้ผลิต Router) หน้าที่ของ Gateway คือ รับ-ส่ง ข้อมูลระหว่าง Client กับ Gateway และ Gateway กับ External IP

2.11.1.2 ชุดสองหรือ Internal IP2 มีชื่อเรียกโดยทั่วไปว่า Client ยังแบ่งย่อยออกไปได้อีกตามจำนวนเครื่องคอมพิวเตอร์ที่เชื่อมต่อใช้อินเทอร์เน็ตร่วมกัน ตัวอย่างเช่น Client PC IP1, Client PC IP2, ต่อๆไป โดยกำหนดไอพีตามลำดับ และตามชนิดของบริษัทผู้ผลิต Router หน้าที่ของ Client คือ รับ-ส่ง ข้อมูลระหว่างเครื่องคอมพิวเตอร์ กับ Gateway

2.13 Domain Name System (DNS)

DNS เป็นระบบจัดการในการแปลงจากชื่อไปเป็นหมายเลข IP Address (name-to-IP Address mapping) หรือในทางกลับกันมันก็สามารถแปลงจาก IP Address ไปเป็นชื่อที่มีการตั้งไว้ DNS ถือเป็นฐานข้อมูลแบบกระจายชนิดหนึ่งที่มีการถูกเรียกใช้งานมากที่สุดและมีการ

เปลี่ยนแปลงมากที่สุดบนโลกใบนี้ ในระบบ DNS จะมีการเก็บชื่อและ IP Address ของเครื่องที่เกี่ยวข้องอย่างเป็นระบบลักษณะการทำงานจะเป็นแบบ Client/Server โดยที่ตัว Server จะเป็นตัวเก็บข้อมูลต่าง ๆ ที่เรียกว่าเป็น DNS Server

ระบบ DNS แบ่งออกได้เป็น 3 ส่วน คือ

- Name Resolvers โดยเครื่อง Client ที่ต้องการสอบถามหมายเลขไอพี เรียกว่า Resolver ซึ่งซอฟต์แวร์ที่ทำหน้าที่เป็น Resolvers นั้นจะถูกสร้างมากับแอพพลิเคชันหรือเป็น Library ที่มีอยู่ใน Client

- Domain Name Space เป็นฐานข้อมูลของ DNS ซึ่งมีโครงสร้างเป็น Tree หรือเป็นลำดับชั้น แต่ละโหนดคือ โดเมนโดยสามารถมีโดเมนย่อย (Sub Domain) ซึ่งจะใช้จุดในการแบ่งแยก

- Name Servers เป็นคอมพิวเตอร์ที่รันโปรแกรมจัดการฐานข้อมูลบางส่วนของ DNS โดย Name Server จะตอบการร้องขอทันที โดยการหาข้อมูลตัวเอง หรือส่งต่อการร้องขอไปยัง Name Server อื่น ซึ่งถ้า Name Server มีข้อมูลของส่วนโดเมนแสดงว่า Server นั้นเป็นเจ้าของโดเมนเรียกว่า Authoritative แต่ถ้าไม่มีเรียกว่า Non-Authoritative

2.13.1 การทำงานของระบบ DNS

การทำงานของระบบชื่อโดเมนนั้นทำงานอยู่บนชั้น Physical เริ่มต้นจากเครื่องคอมพิวเตอร์ซึ่งทำหน้าที่เป็น DNS Server ซึ่งทำงานด้วยซอฟแวร์พิเศษชื่อว่า BIND ที่ทำหน้าที่ในการรับส่งข้อมูลระหว่าง DNS Server แต่ละเครื่องผ่าน DNS Photocal เมื่อมีคำร้อง ขอให้สืบค้นหาหมายเลข ไอพี

2.12.1.1 กระบวนการทำงานของ DNS ประกอบด้วย รีโซลเวอร์ (Resolver) ซึ่งเป็นโปรแกรมในเครื่อง Client ที่ขอบริการ DNS หน้าที่ของรีโซลเวอร์ คือรับคำสั่งจากโปรแกรมประยุกต์ ที่ขอสอบถามข้อมูลจากเนมเซอร์เวอร์ และนำผลลัพธ์ที่ได้ส่งคืนให้แก่โปรแกรมประยุกต์

2.12.1.2 รีโซลเวอร์ในยูนิกส์จะอาศัยแฟ้ม /etc/resolv.conf ที่กำหนดว่าเครื่องที่ใช้งานนั้นอยู่ในโดเมนใดและต้องติดต่อกับเนมเซอร์เวอร์ใด

ตัวอย่างเช่น

Domain net.in.th	157.109.2.67
Name server	157.109.2.71

2.12.1.3 แฟ้มข้างต้นบรรจุไดเรคทีฟ domain กำหนดชื่อโดเมนและไดเรคทีฟ name server กำหนด IP Address ของ name server รีโซลเวอร์จะขอบริการจากเซอร์เวอร์ 157.109.2.67 เป็นอันดับแรก หากไม่สามารถให้บริการได้ รีโซลเวอร์ จะพยายามขอบริการจาก 157.109.2.71เป็นลำดับถัดไป

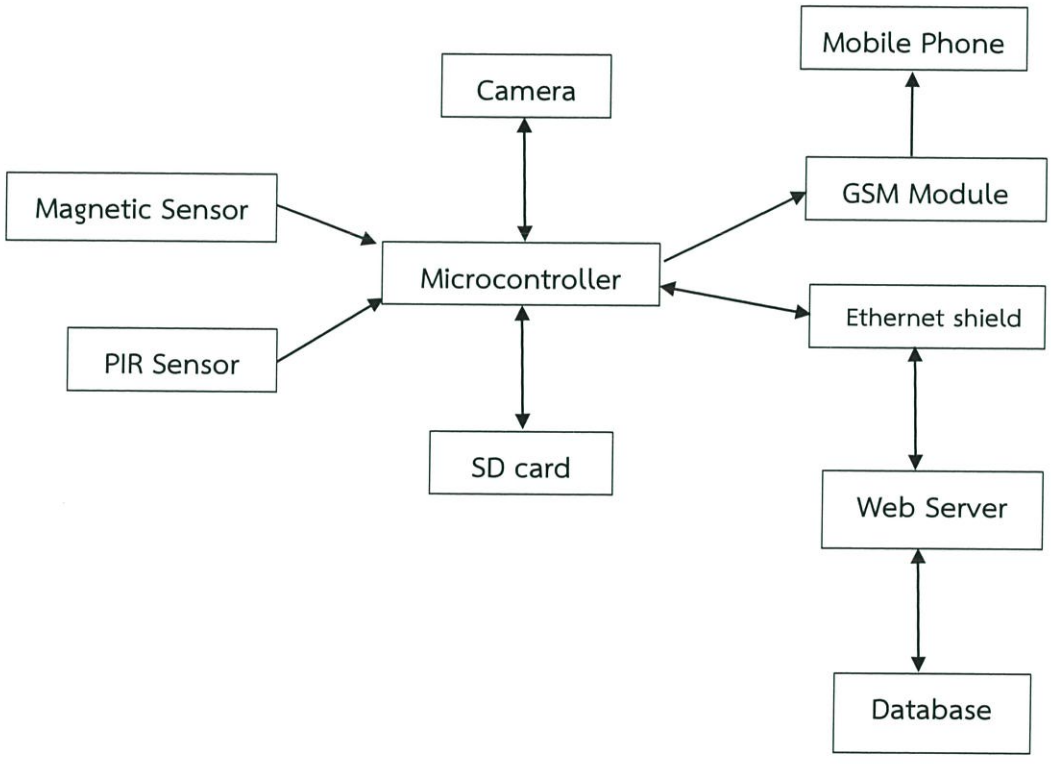
2.12.1.4 ด้าน name server จะมีเดมอน named ทำหน้าที่เป็นเซอร์เวอร์รอรับการร้องขอจากรีโซลเวอร์ เมื่อ named เริ่มทำงานจะอ่านข้อมูลจากแฟ้มบูต /etc/named.Boot หรือ /etc/named.conf

บทที่ 3

การออกแบบ

3.1 การออกแบบ

3.1.1 การออกแบบระบบการทำงานทั้งหมด



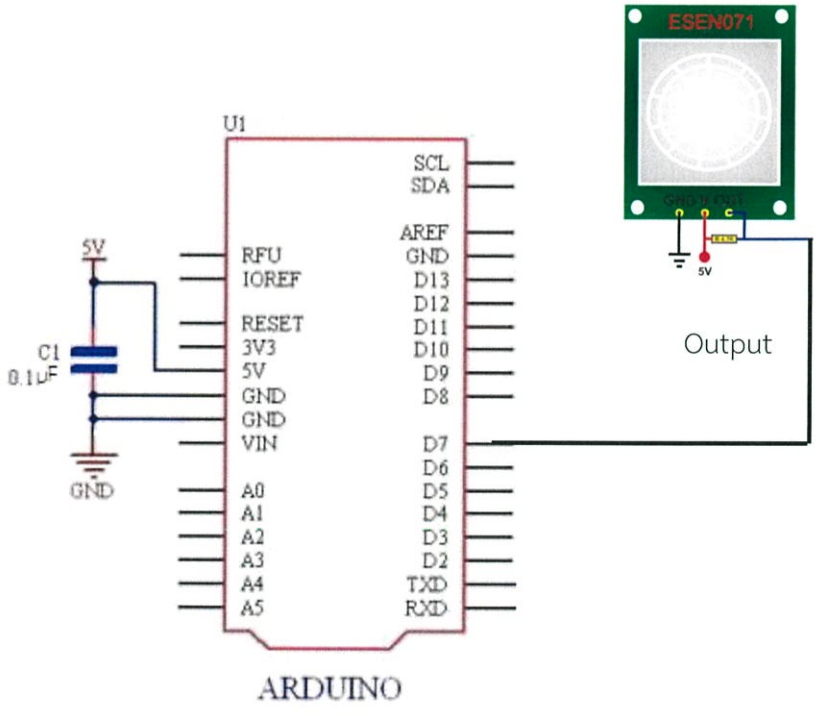
รูปที่ 3.1 บล็อกไดอะแกรมของระบบ

จากรูปที่ 3.1 ระบบจะเริ่มทำงานจากเซ็นเซอร์ตรวจจับความเคลื่อนไหว (PIR sensor) และเซ็นเซอร์แถบแม่เหล็ก (Magnetic sensor) ทำงานโดยตรวจจับสิ่งผิดปกติที่ตรวจจับได้ ส่งสัญญาณไปยังไมโครคอนโทรเลอร์เพื่อสั่งงานให้กล้องถ่ายรูป โดยการถ่ายภาพแล้วบันทึกข้อมูลลงใน SD Card หลังจากนั้นส่งข้อความสั้น (SMS) ไปยังผู้ใช้งาน และไมโครคอนโทรลเลอร์ทำการดึงรูปภาพที่เก็บใน SD Card เก็บในฐานข้อมูล (database) แล้วขึ้นสู่เว็บเซิร์ฟเวอร์ (web server)

3.1.2 ด้านฮาร์ดแวร์

3.1.2.1 การเชื่อมต่อระหว่างเซ็นเซอร์ตรวจจับการเคลื่อนไหว (PIR Motion Sensor) กับไมโครคอนโทรเลอร์

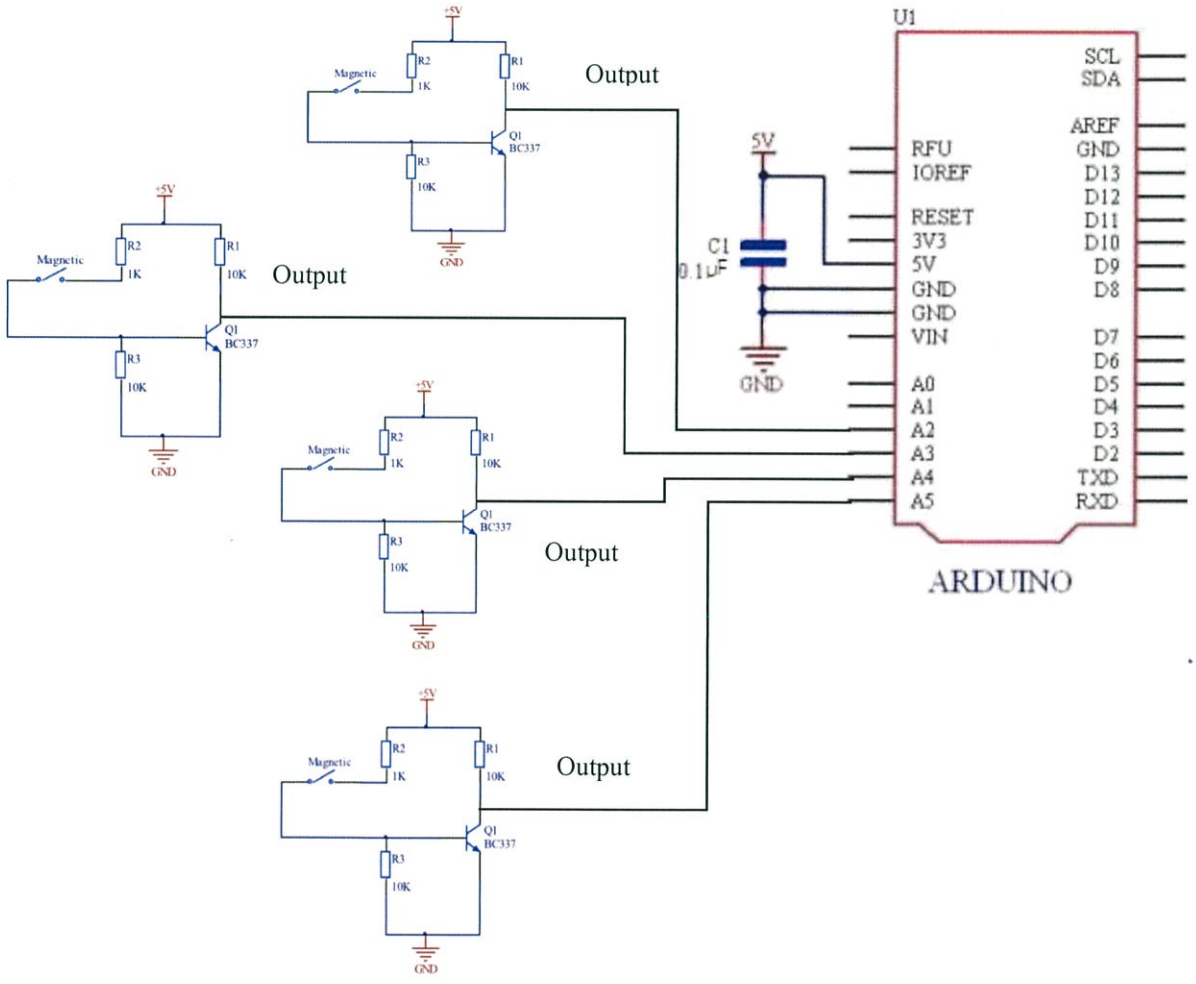
ในโครงงานเล่มนี้เลือกใช้เซ็นเซอร์ตรวจจับการเคลื่อนไหว (PIR Motion Sensor) ซึ่งสามารถตรวจจับการเคลื่อนไหวได้ในระยะ 5 เมตรและรัศมีในการตรวจจับ 60 องศา ซึ่งมีลักษณะเด่นดังนี้ สามารถตรวจจับการเคลื่อนไหวได้ในระยะไกล ประยุกต์ใช้งานง่ายกับไมโครคอนโทรเลอร์ อีกทั้งไวต่อการตรวจจับการเคลื่อนไหวของมนุษย์ด้วยและอุณหภูมิในการทำงานอยู่ในช่วง -20 ถึง 50 องศาเซลเซียส อีกทั้งยังกินกระแสเพียง 300 mA โดยต่อเซ็นเซอร์ตรวจจับการเคลื่อนไหวจากเอาท์พุตของเซ็นเซอร์ตรวจจับการเคลื่อนไหวเข้าพอร์ต D7 ของไมโครคอนโทรเลอร์ ดังรูปที่ 3.2



รูปที่ 3.2 การเชื่อมต่อเซ็นเซอร์ตรวจจับการเคลื่อนไหว (PIR Motion Sensor) กับไมโครคอนโทรเลอร์

3.1.2.2 การเชื่อมต่อเซ็นเซอร์แถบแม่เหล็ก (Magnetic Sensor) กับไมโครคอนโทรเลอร์

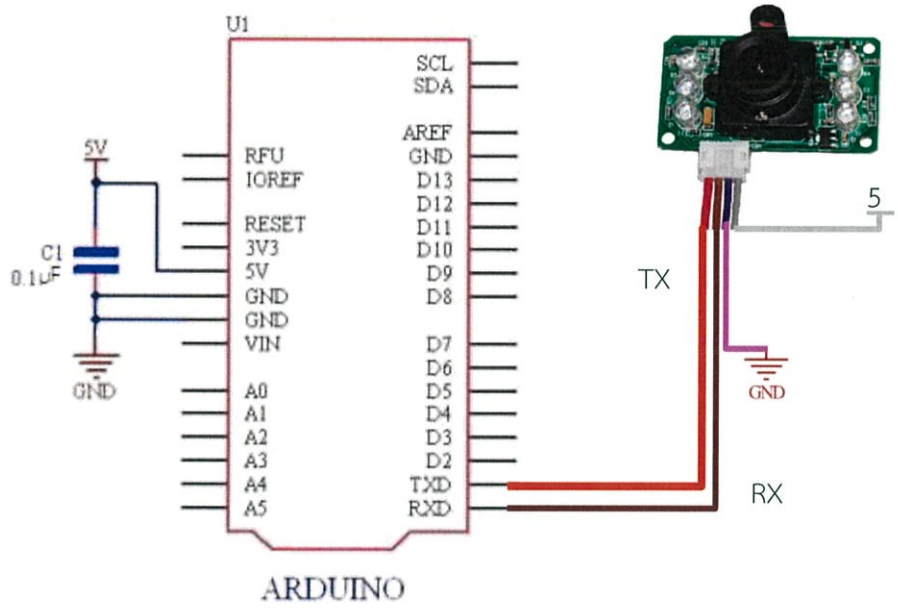
โครงงานนี้เลือกใช้เซ็นเซอร์แถบแม่เหล็ก (Magnetic Sensor) ทั้งหมด 4 ตัว ประกอบด้วยติดหน้าต่าง 3 บานและประตู 1 บาน ต่อเข้ากับไมโครคอนโทรลเลอร์ที่พอร์ต A2, A3, A4 และ A5 ตามลำดับดังแสดงในรูปที่ 3.3 ซึ่งเซ็นเซอร์แถบแม่เหล็ก สามารถประยุกต์ใช้ในการติดประตูและหน้าต่างที่ต้องการตรวจจับ ซึ่งเซ็นเซอร์แถบแม่เหล็ก (Magnetic Sensor) ทำหน้าที่เหมือนสวิตช์ เมื่อมีการเปิดประตูหรือหน้าต่าง วงจรเซ็นเซอร์แถบแม่เหล็ก (Magnetic Sensor) จะส่งลอจิกสูงไปยังไมโครคอนโทรเลอร์ วงจรทำงานที่ไฟเลี้ยง 5 โวลต์



รูปที่ 3.3 การเชื่อมต่อเซ็นเซอร์แถบแม่เหล็ก (Magnetic Sensor) กับไมโครคอนโทรเลอร์

3.1.2.3 การเชื่อมต่อโมดูลกล้อง Infrared JPEG Camera กับไมโครคอนโทรเลอร์

โครงงานเล่มนี้เลือกใช้ โมดูลกล้อง Infrared JPEG Camera ที่มีความสามารถรับส่งข้อมูลผ่านการสื่อสารแบบอนุกรม ดังนั้นจึงสามารถนำไปต่อใช้งานโดยตรงกับไมโครคอนโทรเลอร์ และที่น่าสนใจโดยกล้องจะมี Ambient light ในการปรับ Mode infrared อัตโนมัติ ทำให้กล้องสามารถถ่ายในที่มืดได้ ความละเอียดสูงสุดถึง VGA (640x480) ในการเชื่อมต่อโมดูลกล้องเข้ากับไมโครคอนโทรเลอร์โดยการต่อสามารถดูการใช้งานขาได้จากตารางที่ 2.25 โดยขา TX และ RX ต่อเข้ากับพอร์ต D2 และ D3 ของไมโครคอนโทรเลอร์ตามลำดับ ดังแสดงในรูปที่ 3.4

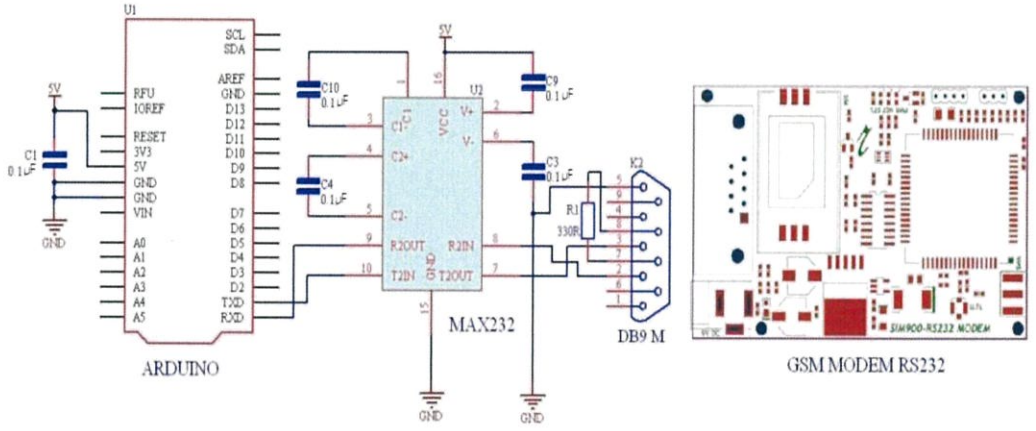


รูปที่ 3.4 การเชื่อมต่อโมดูลกล้อง Infrared JPEG Camera กับไมโครคอนโทรเลอร์

3.1.2.4. การเชื่อมต่อโมดูล GSM SIM300CZ กับไมโครคอนโทรเลอร์

โครงงานเล่มนี้เลือกใช้โมดูล GSM SIM300CZ เพราะเป็นโมดูลสื่อสารระบบ GSM/GPRS ขนาดเล็ก รองรับการสื่อสาร GSM ความถี่ 900/1800/1900 MHz โดย

สั่งงานผ่านพอร์ตอนุกรม RS232 ด้วยชุดคำสั่ง AT Command ดังนั้นจึงนำไปประยุกต์ต่อกับไมโครคอนโทรเลอร์ได้ โดยง่าย แต่ทว่าไมโครคอนโทรเลอร์ที่ใช้เป็นไมโครคอนโทรเลอร์ที่มีสถานะเป็นลอจิกหรือสัญญาณ TTL ขนาด 5 โวลต์ ซึ่งโมดูล GSM SIM300CZ เป็นการสื่อสารอนุกรม RS232 มีขนาดแรงดัน -15 ถึง 15 โวลต์ ดังนั้นจึงจำเป็นต้องใช้ IC MAX 232 เพื่อปรับแรงดันให้เท่ากัน มิฉะนั้นอุปกรณ์จะเกิดความเสียหายขึ้นมาได้ จากรูป 3.5 ทำการที่เชื่อมไมโครคอนโทรเลอร์เข้ากับ GSM SIM300CZ โดยผ่านวงจร MAX232 เพื่อปรับแรงดันให้เท่ากัน โดยการเชื่อมต่อขา 9 และ 10 ของไอซี MAX232 เข้ากับพอร์ต RX และ TX ตามลำดับ



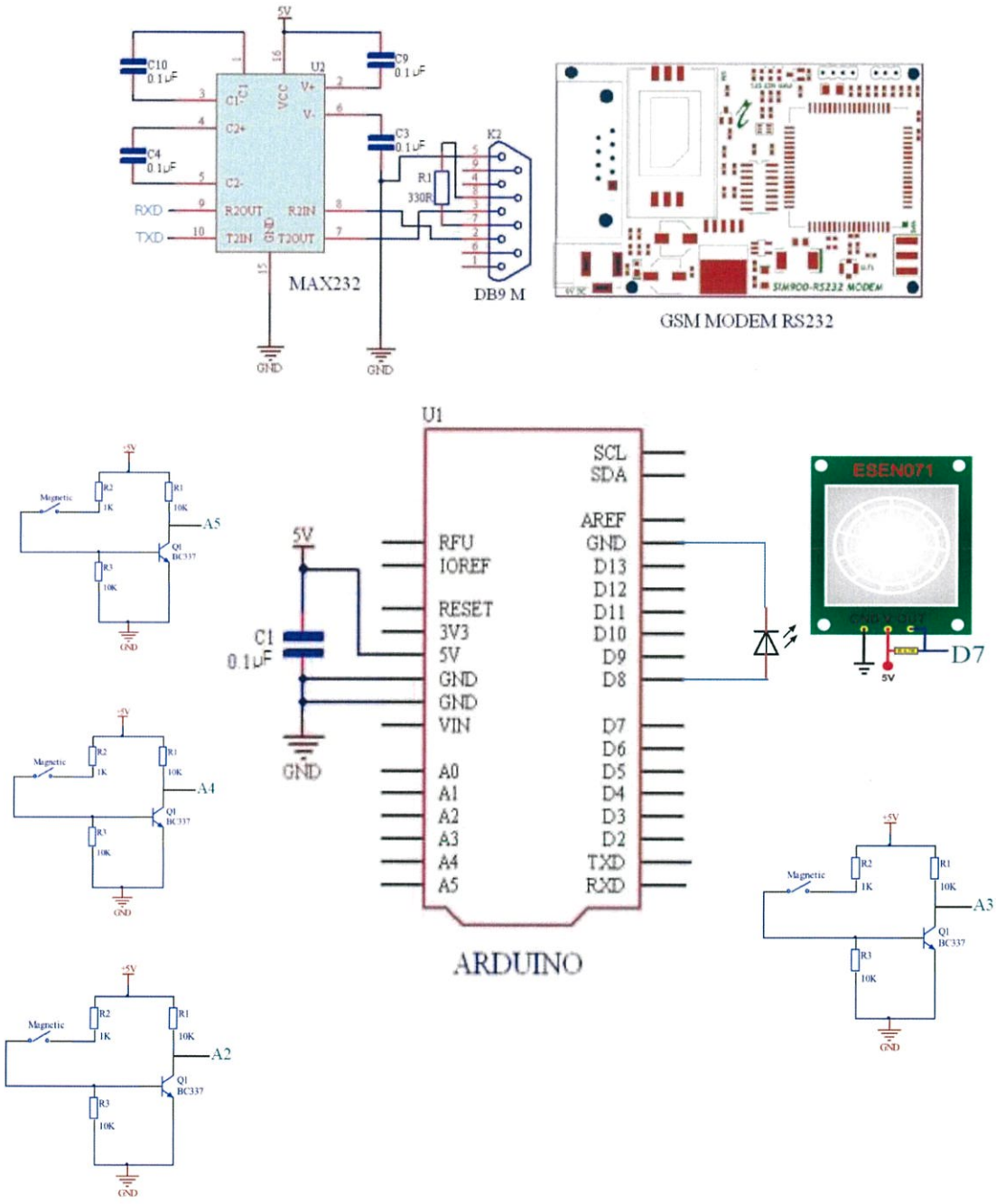
รูปที่ 3.5 การเชื่อมต่อโมดูล GSM SIM300CZ กับไมโครคอนโทรเลอร์

3.1.2.5 การเชื่อมต่อ Ethernet shield Arduino กับไมโครคอนโทรเลอร์

โครงงานเล่มนี้เลือกใช้ Ethernet shield Arduino เนื่องจากว่า Ethernet shield Arduino นั้นสามารถต่อเข้ากับไมโครคอนโทรเลอร์ได้เลย ซึ่ง Ethernet shield Arduino ทำให้ไมโครคอนโทรเลอร์สามารถเชื่อมต่อเข้ากับอินเทอร์เน็ตได้และมีคุณสมบัติการสื่อสารด้วย Protocol TCP และ UDP นอกจากนั้นยังมีช่องเสียบ micro-SD อีกด้วย

3.1.2.6 วงจรรวมของระบบ

ทำการเชื่อมต่ออุปกรณ์ทั้งหมดเข้าด้วยกันด้วยรูปที่ 3.7



รูปที่ 3.6 ภาพรวมของระบบ

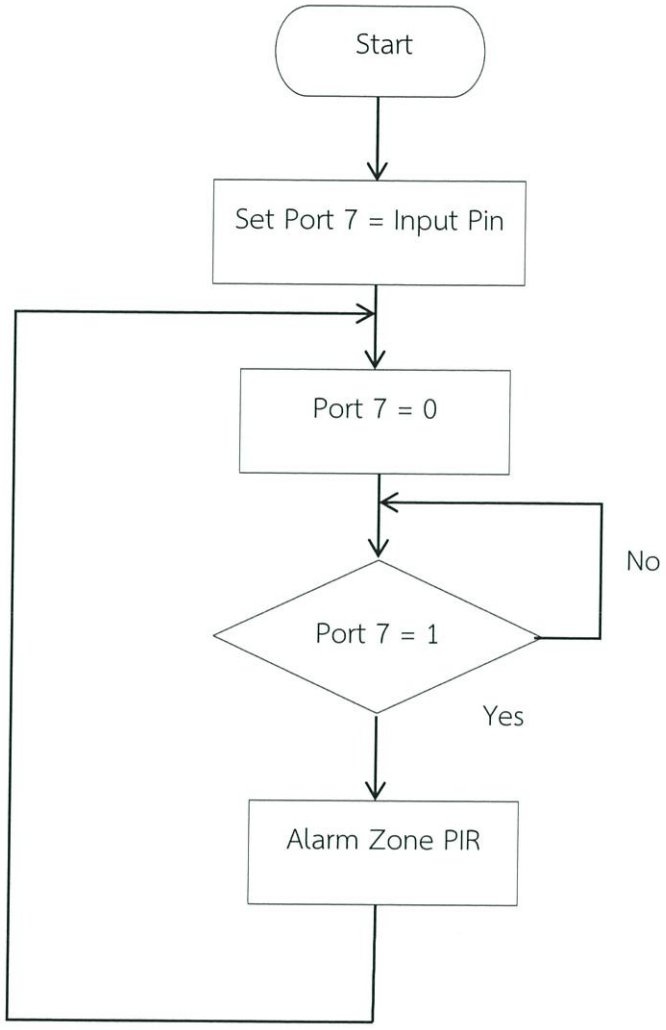
3.1.3 ด้านซอฟต์แวร์

3.1.3.1 โปรแกรม Arduino 1.0.5-r2

โครงงานนี้ใช้โปรแกรม Arduino 1.0.5-r2 ในการเขียนภาษา C และ C++ ใช้ในการอัพโหลดโค้ดลงไมโครคอนโทรเลอร์

3.1.3.2 การติดต่อระหว่างเซ็นเซอร์ตรวจจับการเคลื่อนไหว (PIR Motion Sensor) กับไมโครคอนโทรเลอร์

เริ่มต้นเมื่อเซ็นเซอร์ตรวจจับการเคลื่อนไหวได้ จะส่งสัญญาณเป็น TTL ลอจิกสูง ให้กับไมโครคอนโทรเลอร์ การออกแบบไมโครคอนโทรเลอร์ติดต่อกับเซ็นเซอร์ตรวจจับการเคลื่อนไหว(PIR Motion Sensor) โดยการกำหนดพอร์ต 7 เป็นอินพุตเพื่อรอรับค่าลอจิกจากเซ็นเซอร์ตรวจจับการเคลื่อนไหว(PIR Motion Sensor) แสดงโฟลว์ชาร์ตการทำงานดังรูปที่ 3.9



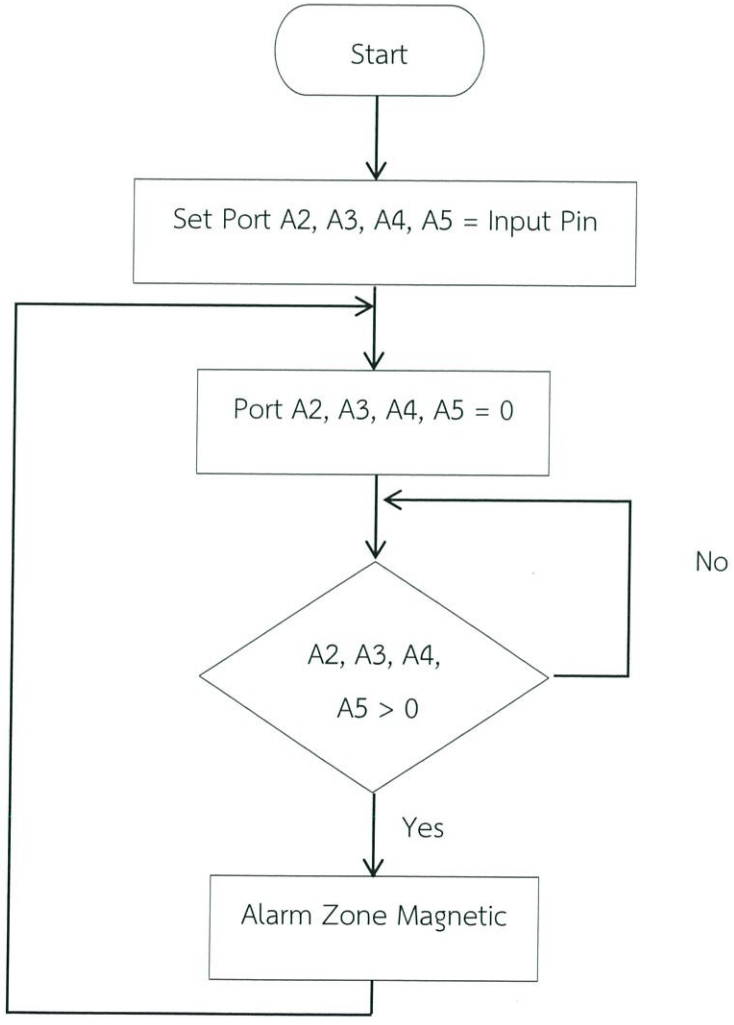
รูปที่ 3.7 โฟลว์ชาร์ตการติดต่อระหว่างเซ็นเซอร์ตรวจจับการเคลื่อนไหว (PIR Motion Sensor) กับไมโครคอนโทรเลอร์

3.1.3.3 การติดต่อระหว่างเซ็นเซอร์แถบแม่เหล็ก (Magnetic Sensor) กับไมโครคอนโทรเลอร์

การออกแบบไมโครคอนโทรเลอร์ติดต่อกับเซ็นเซอร์แถบแม่เหล็ก (Magnetic Sensor) โดยการกำหนดพอร์ต A2,A3,A4 และ A5 ตามลำดับ เป็นอินพุตเพื่อรอรับค่าแรงดันจากเซ็นเซอร์แถบแม่เหล็ก (Magnetic Sensor) แสดงโฟลว์ชาร์ตการทำงานดังรูปที่ 3.10

การทำงานของเซ็นเซอร์แถบแม่เหล็ก(Magnetic Sensor) จะเป็นเหมือนสวิตช์ทั่วไป เมื่อทำการจ่ายไฟ 5 โวลต์ เข้าไปในวงจร ถ้าแถบแม่เหล็กมีแรงดึงดูดซึ่งกัน

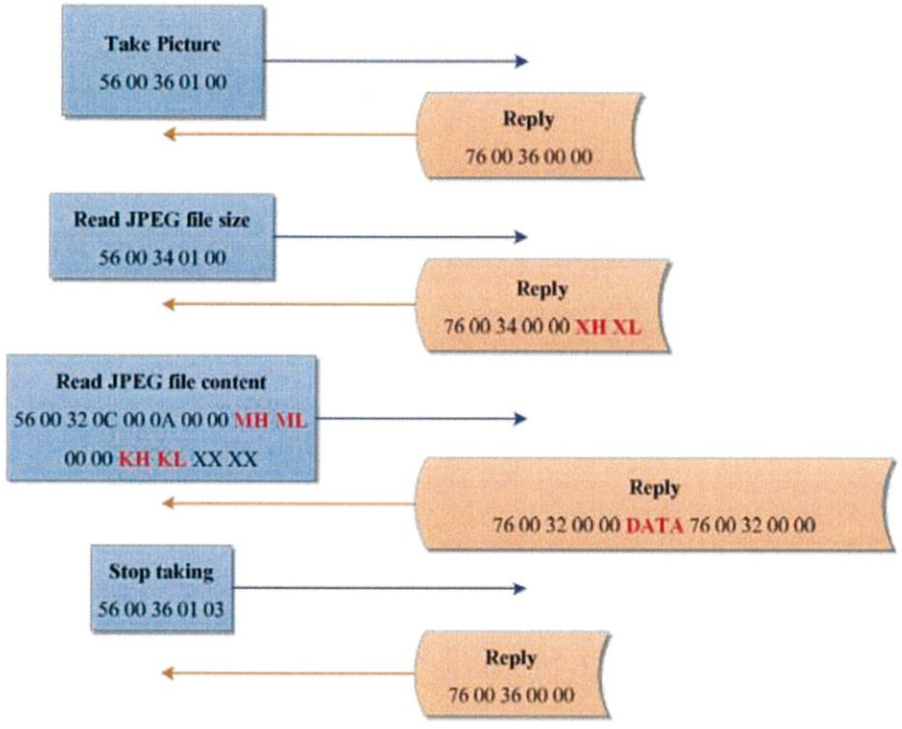
และกันกระแสไฟฟ้าก็จะไหลครบวงจรทำให้ขาพอร์ต A2,A3,A4 และ A5 ตามลำดับ มีสถานะเป็นลอจิกต่ำซึ่งมีแรงดันน้อยกว่าหนึ่ง แต่ถ้าแถบแม่เหล็กอยู่ห่างกันจนไม่มีแรงดูดซึ่งกันและกันแล้ว จะทำให้กระแสไฟฟ้าไหลไม่ครบวงจร ทำให้ขาพอร์ต A2,A3,A4 และ A5 มีสถานะเป็นลอจิกสูงซึ่งมีแรงดันมากกว่าศูนย์



รูปที่ 3.8 โฟลว์ชาร์ตการติดต่อระหว่างเซ็นเซอร์แถบแม่เหล็ก (Magnetic Sensor) กับไมโครคอนโทรเลอร์

3.1.3.4 การติดต่อระหว่างโมดูลกล้อง Infrared JPEG Camera กับไมโครคอนโทรเลอร์

กล้อง Infrared JPEG Camera ควบคุมผ่านทาง UART (TTL) โดยจะใช้คำสั่งในการติดต่อโมดูลกล้อง Infrared JPEG Camera ดังรูปที่ 3.9



รูปที่ 3.9 ขั้นตอนการติดต่อและสั่งงานของโมดูลกล้อง Infrared JPEG Camera

จากรูปที่ 3.9 สรุปคำสั่งที่ใช้ติดต่อโมดูลกล้อง Infrared JPEG Camera ได้ดังนี้

1) ไมโครคอนโทรลเลอร์จะส่งคำสั่ง Take Picture เพื่อสั่งให้โมดูลจับภาพขณะนั้น โดยส่งคำสั่งเป็นเลขฐานสิบหก 56 00 36 01 00 ไป โมดูลกล้องจะตอบกลับมา 76 00 36 00 00 แสดงว่าพร้อมที่จะถ่ายรูป

2) ไมโครคอนโทรลเลอร์จะคำสั่ง Read JPEG file size เพื่อถามหาขนาดของรูปภาพจากโมดูล โดยส่งคำสั่งเป็นเลขฐานสิบหก 56 00 34 01 00 โมดูลกล้องจะตอบกลับมา 76 00 34 00 04 00 00 XH XL ตรง XH และ XL คือขนาดของรูปภาพที่ถูกถ่ายได้ (XH คือไบต์สูง (MSB) และ XL คือไบต์ต่ำ (LSB))

3) ไมโครคอนโทรลเลอร์จะส่งคำสั่ง Read JPEG file content เพื่อดึงภาพจากโมดูล เมื่อโมดูลได้รับคำสั่ง ก็จะตอบกลับโดยส่งข้อมูลรูปภาพตามจำนวนที่กำหนดไว้ โดยส่งคำสั่งเป็นเลขฐานสิบหก 56 00 32 0C 00 0A 00 00 MH ML 00 00 KH KL XX XX โมดูลกล้องตอบกลับมา 76 00 32 00 00 DATA 76 00 32 00 00 โดย MH ML คือ Address เริ่มต้นที่ต้องการจะอ่าน โดยค่าตั้งต้นคือ 00 00 อ่านข้อมูลจาก Buffer ของโมดูลทั้งหมดภายในครั้งเดียว ให้กำหนดเป็น 00 00

4) ไมโครคอนโทรลเลอร์จะส่งคำสั่ง Stop taking picture เพื่อแจ้งให้โมดูลหยุดทำงาน โดยส่งคำสั่งเป็นเลขฐานสิบหก 56 00 36 01 03 โมดูลกล้องจะตอบกลับมา 76 00 36 00 00 แสดงว่าโมดูลกล้องหยุดทำงานแล้ว

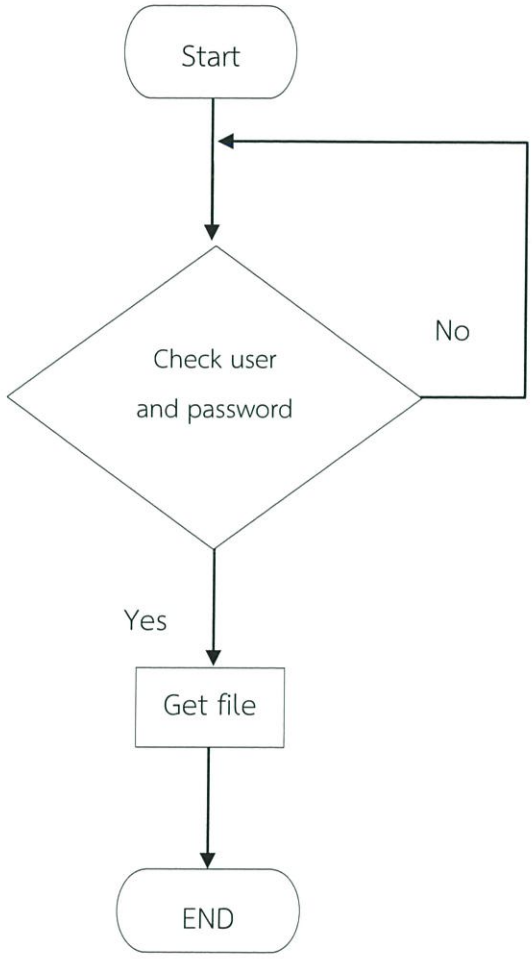
3.1.3.5 การส่งข้อความโดยใช้โมดูล GSM SIM300CZ

ในการส่งข้อความไปยังโทรศัพท์ผู้ใช้งานจะใช้คำสั่งที่เรียกว่า AT Command ในการส่งข้อโดยใช้คำสั่งต่อไปนี้

1. AT+IFC=1,1
2. AT+CMGF=1
3. AT+CMGS="+66843560552"
4. Hello Test SMS
5. กด Ctrl+Z

3.1.3.6 การติดต่อสื่อสารระหว่าง FTP Server กับไมโครคอนโทรเลอร์

FTP เป็นโปรโตคอลประยุกต์ที่ใช้โปรโตคอล TCP/IP โดย FTP ใช้สำหรับแลกเปลี่ยนและจัดการไฟล์บนเครือข่าย TCP/IP เช่นอินเทอร์เน็ต FTP ถูกสร้างขึ้นด้วยสถาปัตยกรรมแบบระบบรับ-ให้บริการ (client-server) ใช้ในการส่งไฟล์เว็บเพจจากแหล่งที่เก็บ (SD card ที่อยู่ในไมโครคอนโทรเลอร์) ไปยังเครื่องคอมพิวเตอร์ โดยแสดงฐานะเป็นเครื่องแม่ข่าย ซึ่งเป็นวิธีที่ง่ายที่สุดในการแลกเปลี่ยนไฟล์ ระหว่างคอมพิวเตอร์บนอินเตอร์เน็ต คล้ายกับ Hypertext Transfer Protocol (HTTP) และใช้การเชื่อมต่อสำหรับส่วนข้อมูลและส่วนควบคุมแยกกันระหว่างเครื่องลูกข่ายกับเครื่องแม่ข่าย

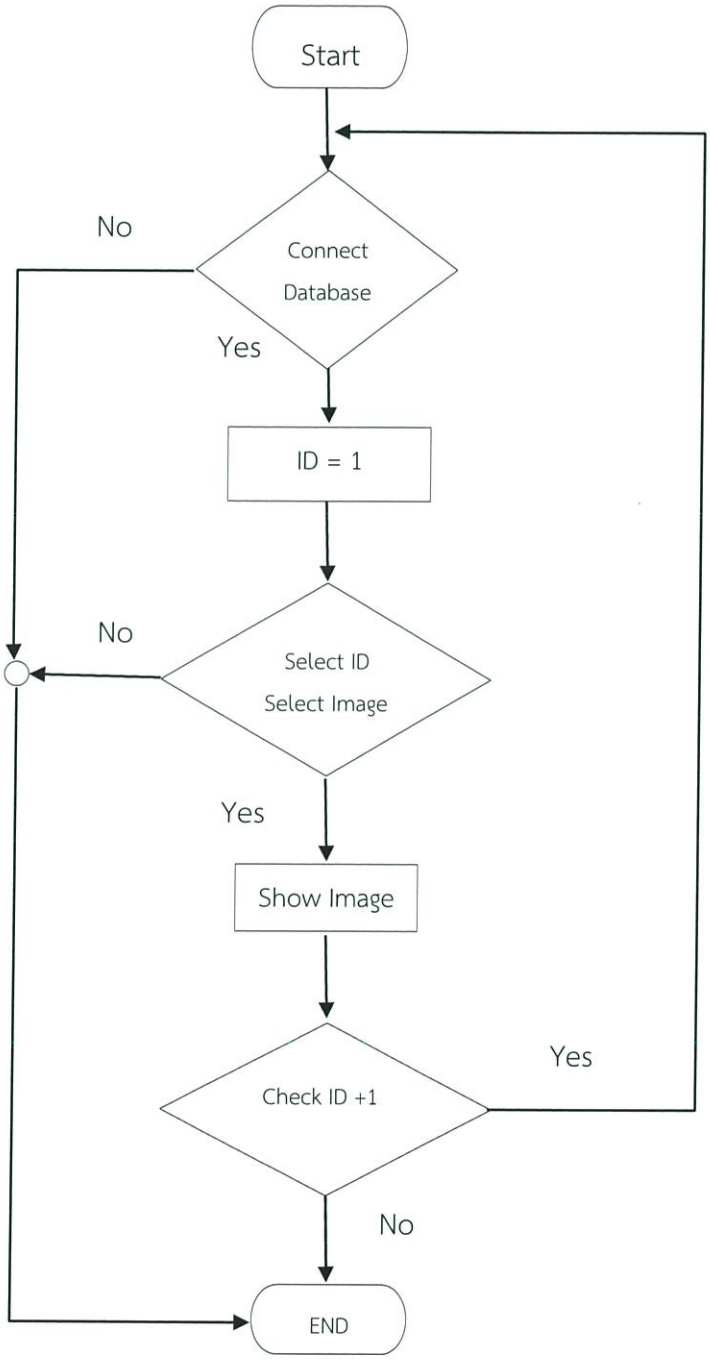


รูปที่ 3.10 โฟลว์ชาร์ตการทำงานของ FTP Server

จากรูปที่ 3.10 การทำงานของ FTP Server เริ่มต้นด้วย Check user and password ถ้าเช็คผ่านก็เข้าไปรับไฟล์ภาพ แล้วจบการทำงาน

การทำงานของ FTP (File Transfer Protocol) เป็นระบบโอนย้ายไฟล์ข้ามระบบเครื่องคอมพิวเตอร์ที่มีความปลอดภัยพอสมควร โดยใช้โปรโตคอล TCP เป็นกลไกขนส่งข้อมูล การเข้าใช้งานผู้ใช้จะต้องแนะนำตนเองต่อเซิร์ฟเวอร์ด้วยชื่อผู้ใช้และรหัสผ่าน จากนั้นจะแสดงชื่อโฟล์เดอร์และชื่อไฟล์ที่มีอยู่ออกมา ความสามารถของ FTP ทำให้ไคลเอนต์โอนย้ายไฟล์ ระหว่างไคลเอนต์ และ FTP Server ได้ รวมทั้งระหว่างเครื่องสองเครื่องที่อยู่ห่างไกลกัน

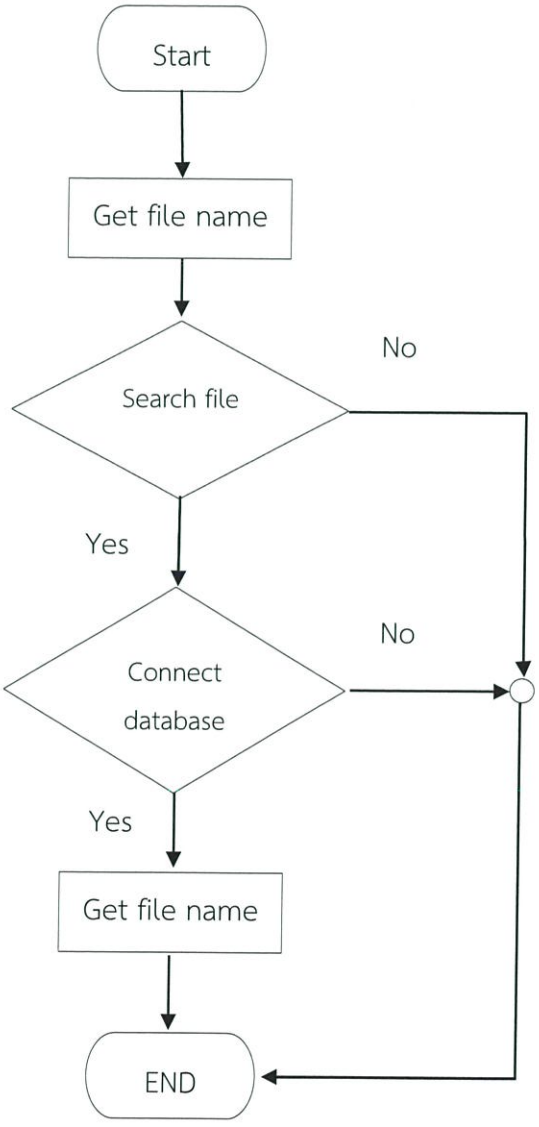
3.1.3.7 การแสดงผลบนหน้าเว็บ Web Server (Show Image)



รูปที่ 3.11 โฟลว์ชาร์ตการแสดงผลบนหน้าเว็บ

จากรูปที่ 3.11 การแสดงผลบนหน้าเว็บ Web Server (Show Image) จะรับข้อมูลจากโมดูลกล้องเป็นรูปภาพทำการเชื่อมต่อด้วยโปรโตคอล FTP และนำแสดงผลเมื่อผู้ใช้งานเข้ามาตรวจสอบ โดยทำการเชื่อมต่อกับฐานข้อมูล กำหนดค่าเมื่อ ID=1 จึงนำข้อมูลเข้ามา หลังจากนั้นเลือกชื่อผู้ใช้งาน (ID) และเลือกรูปภาพ (Image) แสดงรูปภาพที่ได้รับมาขึ้นหน้าเว็บ

3.1.3.8 การเชื่อมต่อฐานข้อมูล (Data base) บน FTP Server



รูปที่ 3.12 โฟลว์ชาร์ตการเชื่อมต่อฐานข้อมูล (Data base) บน FTP Server

จากรูปที่ 3.12 การเชื่อมต่อกับฐานข้อมูล (Data base) เป็นการรับข้อมูลที่ถูกส่งมาจาก web server และเก็บลงเป็นไฟล์รูปภาพในฐานข้อมูล โดยเริ่มทำงานเมื่อมีไฟล์เข้ามาทำการรับไฟล์ ค้นหาไฟล์ที่ต้องการ หลังจากนั้นเชื่อมต่อกับฐานข้อมูลและรับชื่อไฟล์ที่ต้องการเก็บลงในฐานข้อมูล (Data base) เมื่อเข้าไปในเช็คในฐานข้อมูลก็จะมีไฟล์รูปที่รับเข้ามาตามลำดับ

3.2 เครื่องมือที่ใช้ในการทดลอง

1. คอมพิวเตอร์
2. ออสซิลโลสโคป (Oscilloscope)
3. แหล่งจ่ายไฟแรงดัน (Power Supply)
4. โปรแกรม Docklight
5. โปรแกรม HyperTerminal

3.3 การจัดเก็บผลการทดลอง

3.3.1 การทดลองสัญญาณที่ออกจากเซ็นเซอร์ตรวจจับความเคลื่อนไหว

1. ใช้ออสซิลโลสโคปวัดสัญญาณที่ออกจากเอาต์พุตของเซ็นเซอร์ขณะเซ็นเซอร์ตรวจจับความเคลื่อนไหว

3.3.2 การทดลองสัญญาณที่ออกจากเซ็นเซอร์แถบแม่เหล็ก

1. ใช้ออสซิลโลสโคปวัดสัญญาณที่ออกจากเอาต์พุตของเซ็นเซอร์ขณะเซ็นเซอร์แถบแม่เหล็กแยกออกจากกัน

3.3.3 การทดลองสัญญาณที่ออกจากโมดูลกล้อง

1. ใช้ออสซิลโลสโคปวัดสัญญาณตอบสนอง หรือ ACK แต่ละคำสั่งที่กล้องส่งกลับมาตามลำดับ
2. ใช้โปรแกรม Docklight แสดงผลในส่วนข้อมูลภาพที่รับได้จากกล้อง

3.3.4 การทดลองสัญญาณจาก GSM Module

1. ใช้โปรแกรม HyperTerminal ในการส่งข้อความไปยังโทรศัพท์มือถือ

3.3.6 การทดลองส่งข้อมูลไปยังเซิร์ฟเวอร์

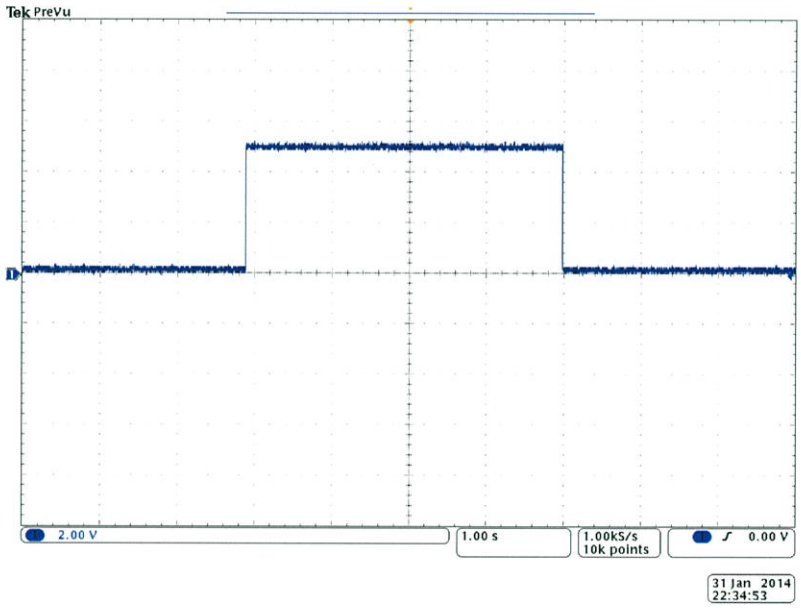
1. เช็คในฐานข้อมูล (Database) เมื่อมีการส่งข้อมูลจากไมโครคอนโทรเลอร์

บทที่ 4

ผลการทดลอง

4.1 การทดสอบสัญญาณที่ออกจากเซ็นเซอร์ตรวจจับการเคลื่อนไหว (PIR motion sensor)

วางเซ็นเซอร์ตรวจจับการเคลื่อนไหว ในบริเวณที่ต้องการตรวจจับ จ่ายไฟเลี้ยง 5 โวลต์ ให้กับเซ็นเซอร์ จากนั้นใช้ออสซิลโลสโคปวัดสัญญาณเอาท์พุตของเซ็นเซอร์ เมื่อมีคนเดินผ่านบริเวณขอบเขตที่เซ็นเซอร์สามารถตรวจจับได้สัญญาณเอาท์พุต ดังรูปที่ 4.1



รูปที่ 4.1 สัญญาณเอาท์พุตเซ็นเซอร์ตรวจจับการเคลื่อนไหว เมื่อมีการตรวจจับ

จากรูปที่ 4.1 เมื่อมีคนเดินผ่านเซ็นเซอร์ สัญญาณที่ได้จะเป็นสัญญาณรูปพัลส์ มีขนาด 5 โวลต์ตามไฟเลี้ยงที่ป้อนเข้าไปให้กับเซ็นเซอร์

4.1.1 การทดสอบขอบเขตของเซ็นเซอร์ตรวจจับการเคลื่อนไหว

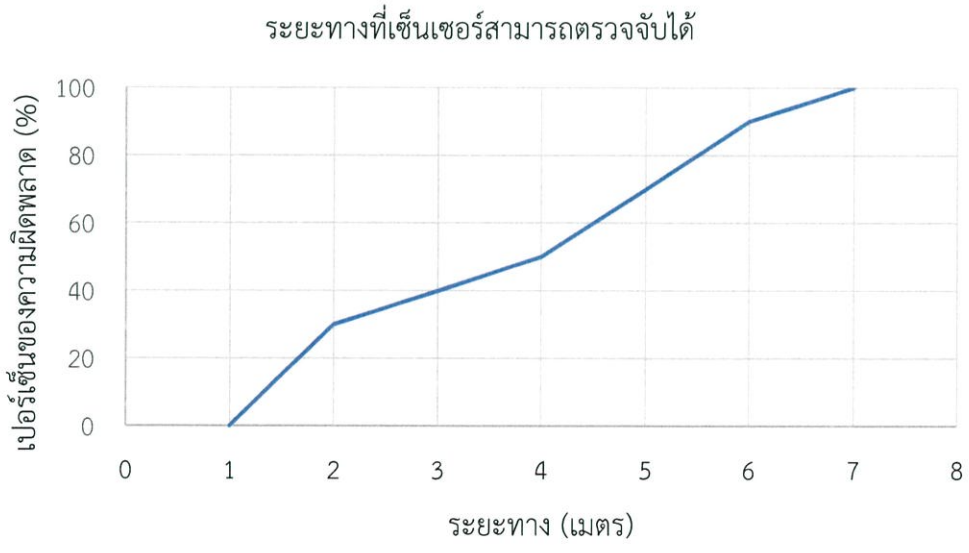
การทดสอบและติดตั้งประสิทธิภาพของเซ็นเซอร์ตรวจจับการเคลื่อนไหว เพื่อให้รู้ว่าขอบเขตการตรวจจับของเซ็นเซอร์นั้นอยู่ที่ระยะตรวจสอบได้

4.1.1.1 วัดระยะทางที่เซ็นเซอร์สามารถตรวจจับได้ ทำการทดลองโดยเดินผ่านเซ็นเซอร์ เป็นจำนวน 20 ครั้ง ดังตารางที่ 4.1

ตารางที่ 4.1 การทดสอบระยะทางที่เซ็นเซอร์สามารถตรวจจับได้

ระยะทาง (เมตร)	จำนวนครั้งที่ตรวจจับได้	เปอร์เซ็นต์ของความผิดพลาด (%)
1	20	0
2	14	30
3	12	40
4	10	50
5	6	70
6	2	90
7	0	100

จากตารางที่ 4.1 เมื่อทดสอบระยะทาง พบว่าเมื่อเข้าใกล้เซ็นเซอร์ในระยะ 1 เมตร จะสามารถตรวจจับได้ 20 ครั้งจากการทดสอบทั้งหมดจำนวน 20 ครั้ง คิดเป็นเปอร์เซ็นของความผิดพลาดเป็น 0 เปอร์เซ็นต์และในระยะ 7 เมตร จะไม่สามารถตรวจจับได้เลย เนื่องจากเซ็นเซอร์ไม่สามารถรับรังสีอินฟราเรดแล้วนำไปประมวลผลได้ คิดเป็นเปอร์เซ็นความผิดพลาดได้ 100 เปอร์เซ็นต์ และนำมาเขียนกราฟได้ดังรูปที่ 4.2

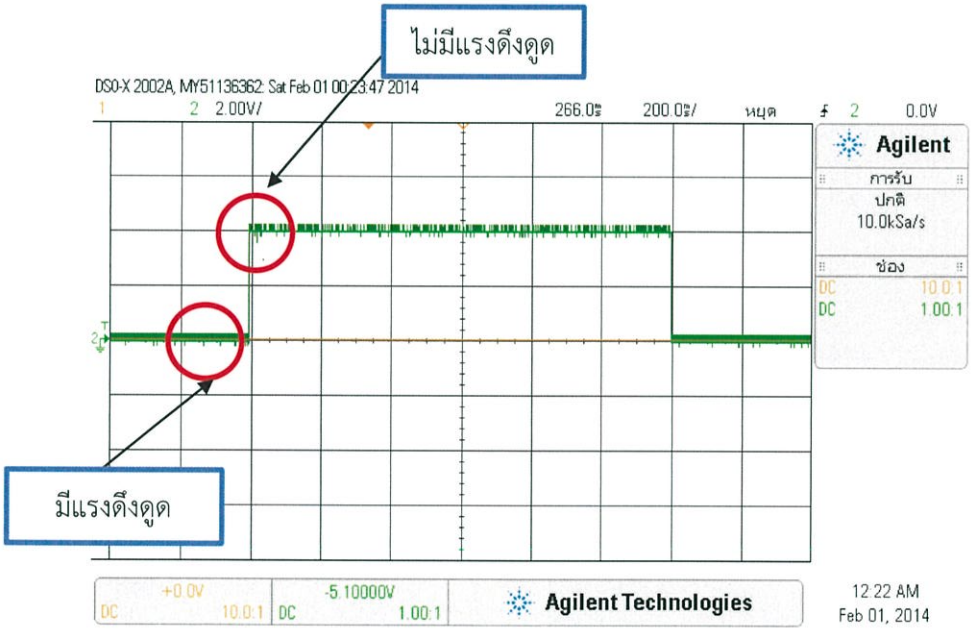


รูปที่ 4.2 ระยะทางที่เซ็นเซอร์สามารถตรวจจับได้

จากรูปที่ 4.2 เป็นกราฟแสดงระยะทางที่เซ็นเซอร์ตรวจจับได้ แกน x คือ ระยะทาง แกน Y คือจำนวนครั้งที่ตรวจจับได้ จากกราฟจะเห็นว่ายิ่งระยะการตรวจจับไกลมากขึ้น เซ็นเซอร์จะมีการตรวจจับได้น้อยลงจนถึงระยะ 7 เมตร เซ็นเซอร์จะไม่สามารถตรวจจับได้

4.2 การทดสอบสัญญาณที่ออกจากเซ็นเซอร์แถบแม่เหล็ก (Magnetic Sensor)

ทำการติดเซ็นเซอร์แถบแม่เหล็กกับประตู โดยทำการปิดประตูซึ่งทำให้เซ็นเซอร์แถบแม่เหล็กมีแรงดึงดูดกัน จ่ายไฟเลี้ยง 5 โวลต์ ให้กับเซ็นเซอร์ จากนั้นใช้ออสซิลโลสโคปวัดสัญญาณเอาท์พุตของแถบแม่เหล็ก (Magnetic Sensor) และเมื่ออุปกรณ์เซ็นเซอร์แถบแม่เหล็ก ไม่มีแรงดึงดูดกันจะได้สัญญาณเอาท์พุต ดังรูปที่ 4.3



รูปที่ 4.3 สัญญาณเอาท์พุตเซ็นเซอร์แถบแม่เหล็ก เมื่อเซ็นเซอร์แถบแม่เหล็ก หลุดออกจากกัน

จากรูปที่ 4.3 แสดงได้ว่าเซ็นเซอร์ทำงานที่ 2 สถานะ โดยสถานะแรกเมื่อไม่ได้ทำการเปิดประตูทำให้เซ็นเซอร์มีแรงดึงดูดซึ่งกันและกัน ทำการวัดสัญญาณเอาท์พุทของเซ็นเซอร์ จะได้สัญญาณ มีขนาด 0 โวลต์ซึ่งแสดงว่าเซ็นเซอร์ไม่มีการตรวจจับ และสถานะที่สองเมื่อมีการเปิดประตูจะทำให้เซ็นเซอร์ไม่มีแรงดึงดูด ทำการวัดสัญญาณเอาท์พุทของเซ็นเซอร์ จะได้สัญญาณขนาด

4 โวลต์ซึ่งแสดงว่าเซ็นเซอร์มีการตรวจจับ ไปจนกว่าจะมีการปิดประตูอีกครั้ง ซึ่งจะให้สัญญาณเอาท์พุตขนาด 0 โวลต์ ออกมา

4.2.1 วัดระยะห่างของแรงดึงดูดเซ็นเซอร์แถบแม่เหล็ก

ทำการวัดระยะห่างของแรงดึงดูดเซ็นเซอร์แถบแม่เหล็กเพื่อทำให้รู้ว่าระยะห่างที่เหมาะสม ทำการวัดโดยเอาท์พุตเมื่อเซ็นเซอร์แถบแม่เหล็กมีแรงดึงดูดกัน และเมื่อเซ็นเซอร์แถบแม่เหล็กหลุดออกจากกัน แสดงดังตารางที่ 4.2

ตารางที่ 4.2 การทดสอบระยะห่างของแรงดึงดูดเซนเซอร์แถบแม่เหล็ก

ระยะห่างของเซนเซอร์แถบแม่เหล็ก (เซนติเมตร)	สัญญาณเอาท์พุต (โวลต์)
0	0
0.1	0
0.2	0
0.4	0
0.6	0
0.8	0
1.0	0
1.2	0
1.4	0
1.6	0
1.8	0
2.0	0
2.2	4
2.4	4

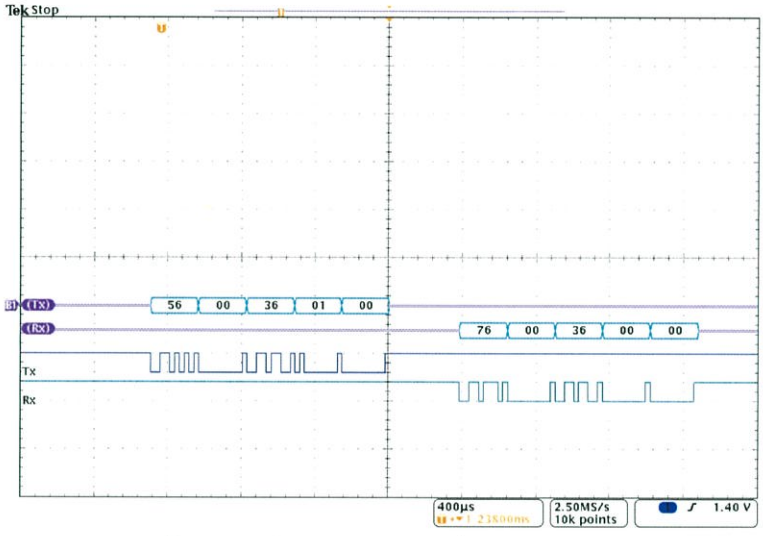
จากตารางที่ 4.2 ทำการวัดระยะห่างของเซนเซอร์แถบแม่เหล็ก โดยทำการแยกแถบแม่เหล็กออกตามระยะดังกล่าวและวัดสัญญาณเอาท์พุต พบว่า ที่ระยะ 2.2 เซนติเมตร ซึ่งแสดงว่าเซ็นเซอร์มีการตรวจจับได้ จากตารางที่ 4.2 นำมาเขียนกราฟระหว่างระยะห่างของเซนเซอร์แถบแม่เหล็กกับสัญญาณเอาท์พุตจะได้ดังรูปที่ 4.4

รูปที่ 4.4 ระยะห่างของแรงดึงดูดเซ็นเซอร์แถบแม่เหล็กกับแรงดัน

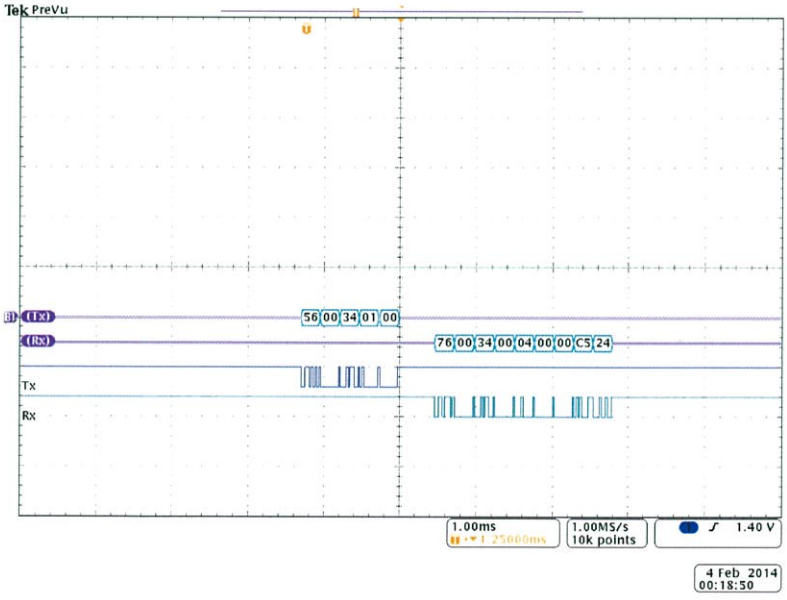
จากรูปที่ 4.4 แสดงกราฟระยะห่างของแรงดึงดูดเซ็นเซอร์แถบแม่เหล็กกับแรงดัน แกน x คือ ระยะห่างของเซ็นเซอร์แถบแม่เหล็ก แกน Y คือ แรงดันที่วัดได้ จากกราฟจะเห็นว่าเซ็นเซอร์แถบแม่เหล็กจะเริ่มตรวจจับได้ที่ระยะ 2.2 เซนติเมตร

4.3 การทดสอบสัญญาณที่ออกจากโมดูลกล้อง

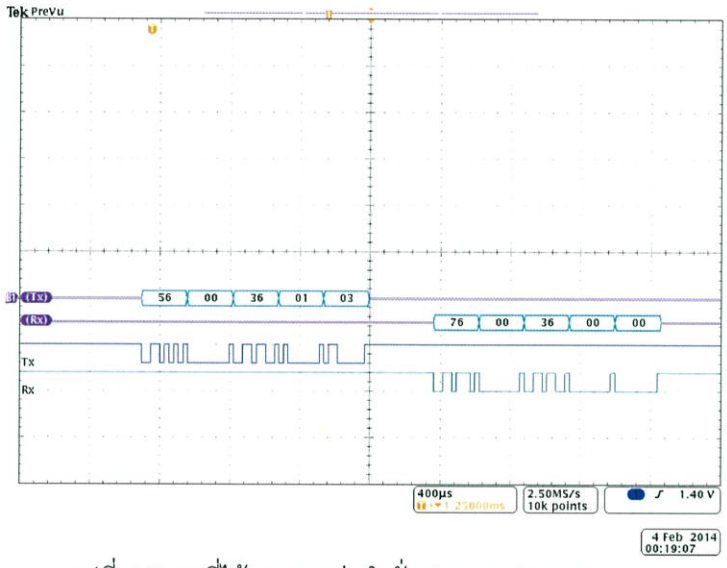
การทดสอบส่งคำสั่งจากโปรแกรม Docklight เพื่อส่งคำสั่งที่ใช้ติดต่อกับโมดูลกล้อง จะทำการบันทึกสัญญาณที่ขา TX และ RX ของโมดูลกล้อง ซึ่งจะได้ผลดังนี้



รูปที่ 4.5 ผลที่ได้จากการส่งคำสั่ง Take Picture



รูปที่ 4.6 ผลที่ได้จากการส่งคำสั่ง Read JPEG file size



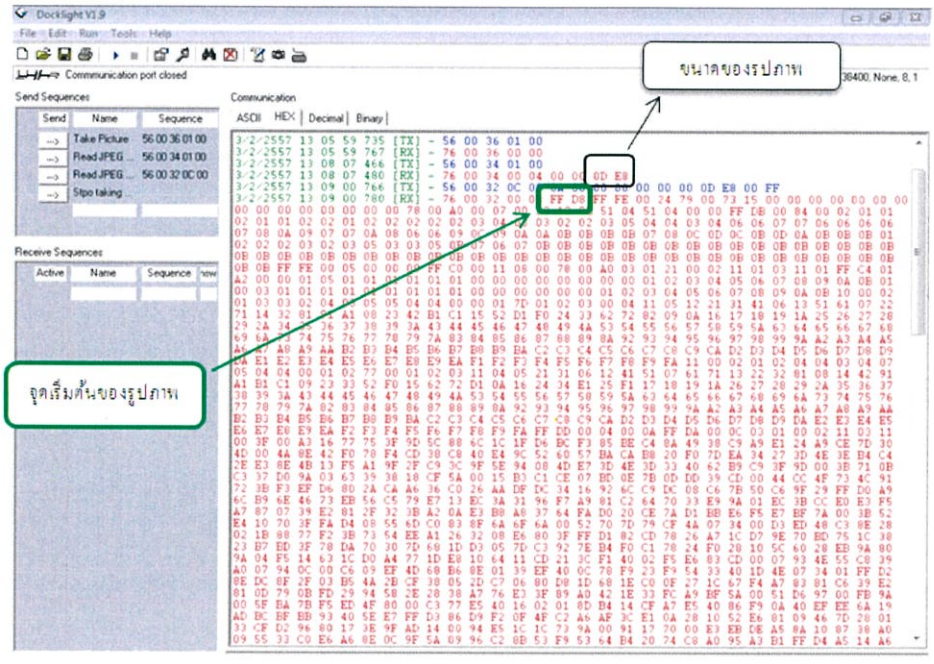
รูปที่ 4.7 ผลที่ได้จากการส่งคำสั่ง Stop taking picture

ในการทดสอบสัญญาณที่ออกจากโมดูลกล้อง ส่งคำสั่งจากโปรแกรม Docklight เพื่อใช้ติดต่อกับโมดูลกล้องและใช้ออสซิลโลสโคปวัดสัญญาณที่ TX และ RX โดยใช้คำสั่งดังนี้

1) Take picture ไปยังโมดูลกล้อง เพื่อสั่งให้โมดูลจับภาพขณะนั้น โดยคำสั่ง (TX) คือ 56 00 36 01 00 และฝั่งตอบกลับ (RX) คือ 76 00 36 00 00 ดังรูปที่ 4.5

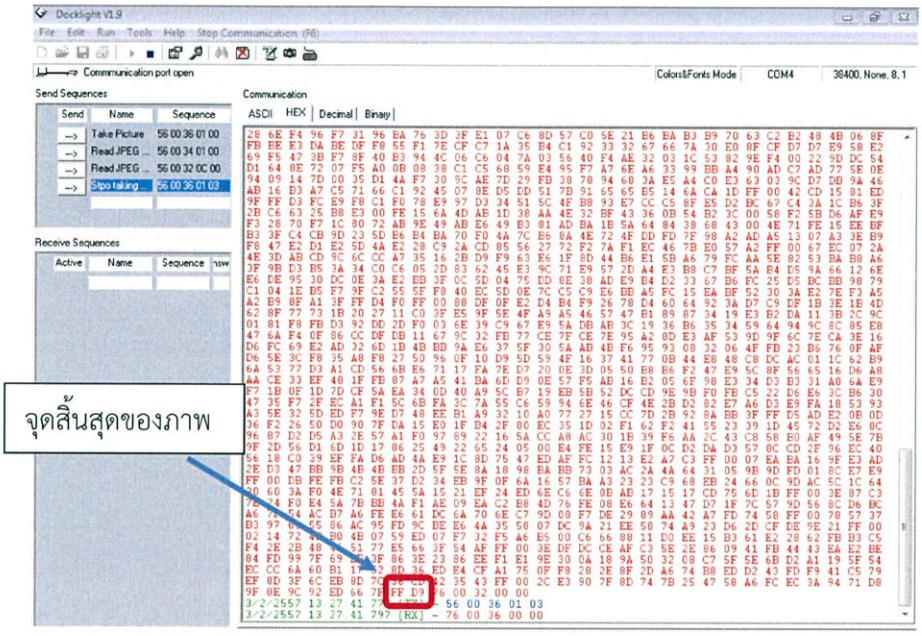
2) Read JPEG file size ไปยังโมดูลกล้อง เพื่อถามค่าขนาดจากโมดูล ทางฝั่งโมดูลจะตอบกลับมา (RX) 76 00 34 00 04 00 00 C5 24 ซึ่ง C5 และ 24 คือขนาดภาพที่ถูกถ่ายได้ (C5) คือ ไบต์ซ้ายมือสุด (MSB) และ XL คือ ไบต์ขวามือสุด (LSB)) ดังรูปที่ 4.6

3) Stop taking picture ไปยังโมดูลกล้อง เพื่อแจ้งโมดูลให้หยุดการเชื่อมต่อโดยคำสั่ง (TX) คือ 56 00 36 01 03 และฝั่งตอบกลับ (RX) คือ 76 00 36 00 00 ดังรูปที่ 4.7



รูปที่ 4.8 ข้อมูลภาพผ่านทางโปรแกรม Docklight

เริ่มจากส่งคำสั่ง Take picture ไปยังโมดูลกล้อง โดยใช้คำสั่ง (TX) คือ 56 00 36 01 00 และฝั่งโมดูลตอบกลับ (RX) คือ 76 00 36 00 00 แสดงว่าโมดูลกล้องมีการจับภาพแล้ว และทำการส่งคำสั่ง Read JPEG file size ไปยังโมดูลกล้องอีกครั้ง เพื่อทราบถึงขนาดของภาพ โดยคำสั่ง (TX) คือ 56 00 34 01 00 และฝั่งโมดูลตอบกลับ (RX) คือ 76 00 34 00 04 00 00 0D E8 ซึ่ง 0D E8 คือขนาดของรูปภาพ หรือเท่ากับ 3560 ไบต์ คำนวณได้จากนำ 0DE8 มาแปลงเป็นเลขฐานสิบ ข้อมูลรูปภาพทั้งหมดที่ได้จะเป็นเลขฐาน 16 และจุดเริ่มต้นของภาพ เริ่มต้นที่จุด FF D8 ตามรูปที่ 4.8



รูปที่ 4.9 ข้อมูลภาพผ่านทางโปรแกรม Docklight (ต่อ)

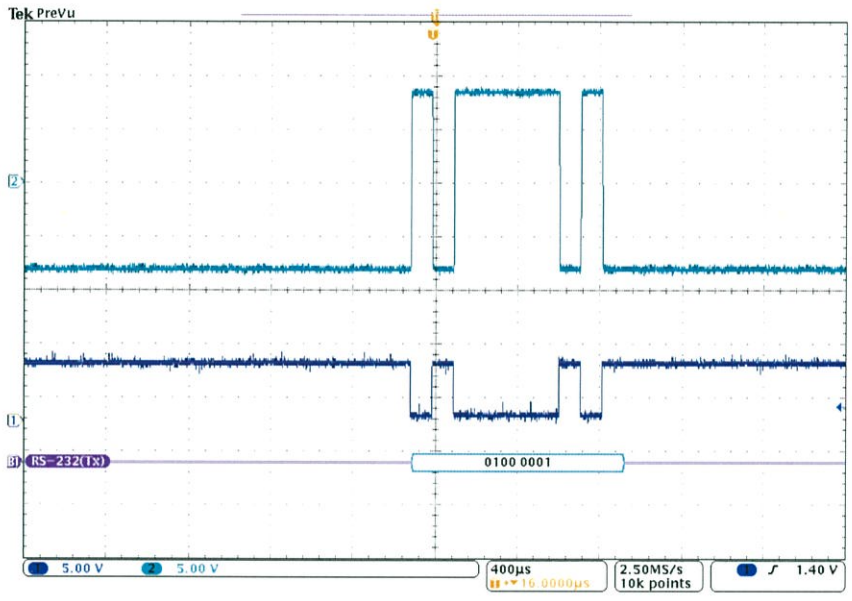
จากข้อมูลภาพข้างต้นจะแสดงจุดสิ้นสุดของรูปภาพนี้ที่จุด FF D9 และทำการส่งคำสั่ง Stop taking picture ไปยังโมดูลกล้อง เพื่อแจ้งโมดูลให้หยุดการเชื่อมต่อ โดยคำสั่ง (TX) คือ 56 00 36 01 03 และฝั่งตอบกลับ (RX) คือ 76 00 36 00 00 ดังรูปที่ 4.9

4.4 การทดสอบการส่งข้อมูลออกทางพอร์ต Serial ผ่านวงจร MAX 232

เมื่อทำการทดสอบส่งข้อมูล โดยการส่งตัวอักษร ‘A’ ออกทางขา TX ของไมโครคอนโทรเลอร์ ออกไปทางพอร์ต Serial ผ่านวงจร MAX232 โดยกำหนดให้ส่งข้อมูลด้วยอัตราบิต 9600 บิต/วินาที ข้อมูลของตัวอักขระ 8 บิต ไม่มีบิตพาริตี้

รหัสแอสกี้ของตัวอักษร ‘A’ คือ 41 ซึ่งเป็นเลขฐาน 16 ไมโครคอนโทรลเลอร์จะส่งบิตออกมาเป็น 0100000101 โดยส่งข้อมูลจากซ้ายไปขวาตามลำดับ ข้อมูลจากบิตแรก คือ Start bit มีค่าเท่ากับ ‘1’ และบิตสุดท้ายคือ Stop bit มีค่าเท่ากับ ‘0’ ส่วนค่าของ ‘A’ คือ 01000001ในเลขฐานสอง ซึ่งถูกส่งมาหน้าเป็นบิต 10000010 ตามลำดับ

ทำการวัดสัญญาณที่ออกจากพอร์ต Serial ของไมโครคอนโทรเลอร์ (Ch.1) สัญญาณหลังผ่านวงจร MAX232 เพื่อส่งเข้าไป GSM Module Sim300CZ (Ch.2) จะได้สัญญาณจากออสซิลโลสโคป ดังนี้



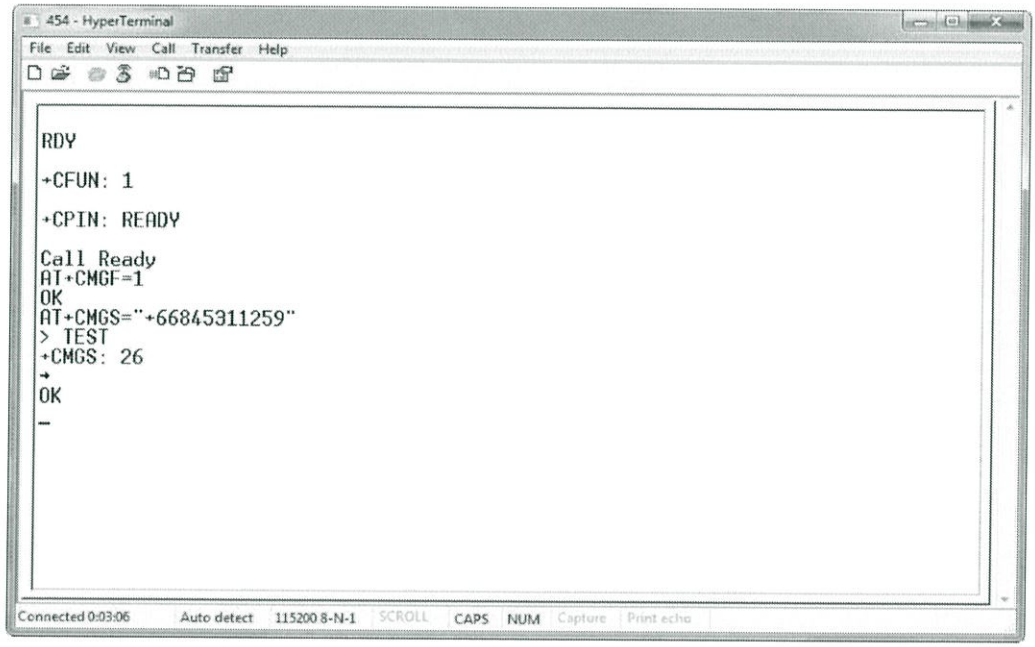
รูปที่ 4.10 ผลการทดลองวัดสัญญาณที่ส่งจากไมโครคอนโทรลเลอร์และผ่านวงจร MAX232

จากรูปที่ 4.10 จะพบว่าที่ Ch.1วงจร MAX232 รับค่าจาก พอร์ต Serial โดยจะใช้สัญญาณไฟฟ้า +5 โวลต์ แทนบิต '1' สัญญาณไฟฟ้า 0 โวลต์ เพื่อแทนบิต '0' โดยจะส่งสัญญาณ 01000001 แทนตัวอักษร 'A' ที่ Ch.2 จะพบว่าหลังผ่านวงจร MAX232 จะได้สัญญาณด้านบวกเพื่อแทนบิต '0' และจะได้สัญญาณด้านลบแทนบิต '1' เพื่อส่งค่าสัญญาณ 01000001 แทนตัวอักษร 'A' ให้ GSM Module SIM300CZ

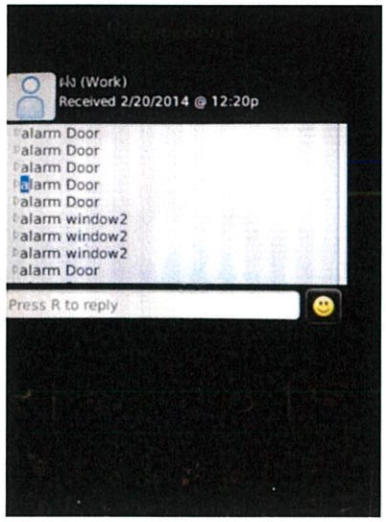
4.5 การทดสอบส่งข้อความโดยใช้ GSM Module Sim300CZ ผ่านโปรแกรม HyperTerminal

ในการทดลองส่งข้อความไปยังโทรศัพท์ผู้ใช้โดยการป้อนคำสั่ง AT Command ในการสั่ง การทดลองนี้จะใช้คำสั่งดังนี้เริ่มต้น กำหนด AT+CMGF=1 เป็นกำหนดค่าอยู่ในโหมดการส่งข้อความ โดยในการส่งข้อความจะใช้คำสั่ง AT+CMGS= "+เบอร์ผู้รับ" โดยเบอร์ผู้รับต้องใส่รหัสประเทศนำหน้าแทนศูนย์ด้วยเสมอ ซึ่งในกรณีเป็นประเทศไทยรหัสประเทศเป็น "66" เมื่อโมดูล

ได้รับคำสั่ง AT+CMGS เรียบร้อยแล้วจะตอบรับด้วยการส่งเครื่องหมาย “>” กลับมาบอก หลังจากนั้นพิมพ์ข้อความที่ต้องการส่งไป แล้วปิดท้ายด้วยข้อความรหัส Ctrl+Z ตามด้วย “Enter” ดังรูปที่ 4.11



รูปที่ 4.11 วิธีการทดลองส่งข้อความโดยใช้ GSM Module SIM300CZ ผ่านโปรแกรม HyperTerminal

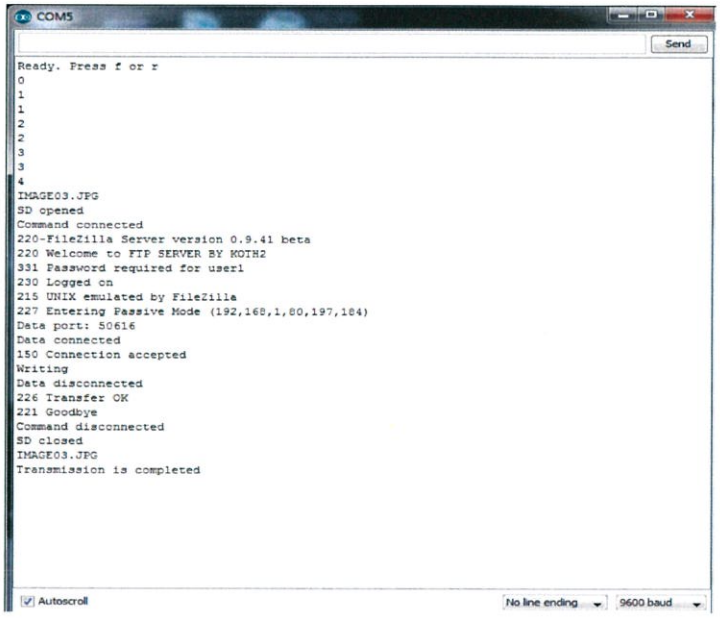


รูปที่ 4.12 ข้อความสั้นที่แจ้งเตือนผู้ใช้งานแจ้งเตือนเมื่อมีสิ่งผิดปกติผ่านเซ็นเซอร์

จากรูปที่ 4.12 เป็นการแจ้งเตือนข้อความสั้น (SMS) โดยส่งข้อความคำว่า alarm Door และ alarm window ไปยังผู้ใช้งานหรือเจ้าของบ้าน จะแจ้งเตือนก็ต่อเมื่อมีสิ่งผิดปกติหรือคนเดินผ่านเซ็นเซอร์ตรวจจับความเคลื่อนไหวที่บริเวณประตูและเมื่อเซ็นเซอร์แถบแม่เหล็กแยกออกจากกันซึ่งติดไว้บริเวณหน้าต่างของบริเวณบ้าน

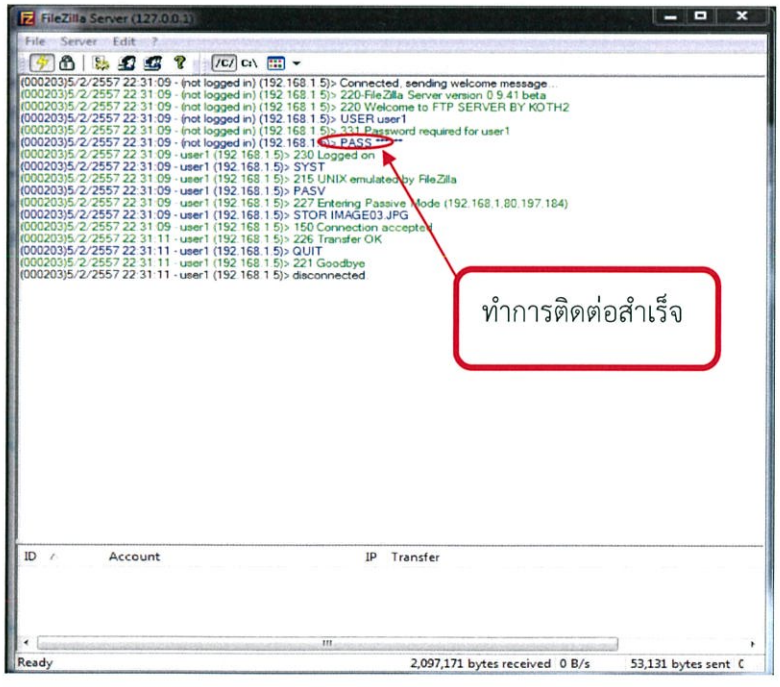
4.6 ผลการทดลองส่งข้อมูลไปยังเซิร์ฟเวอร์

ในการทดลองส่งข้อมูลไปยังเซิร์ฟเวอร์จะทำการส่งข้อมูลเป็นไฟล์รูปภาพที่ถูกถ่ายจากโมดูลกล้อง นำขึ้นแสดงหน้าเว็บเซิร์ฟเวอร์และจะเก็บข้อมูลที่เป็นไฟล์รูปภาพนี้ลงในฐานข้อมูล (Data base) เพื่อป้องกันการสูบหายของข้อมูลเมื่อนำขึ้นแสดงบนหน้าเว็บเซิร์ฟเวอร์ไม่ได้



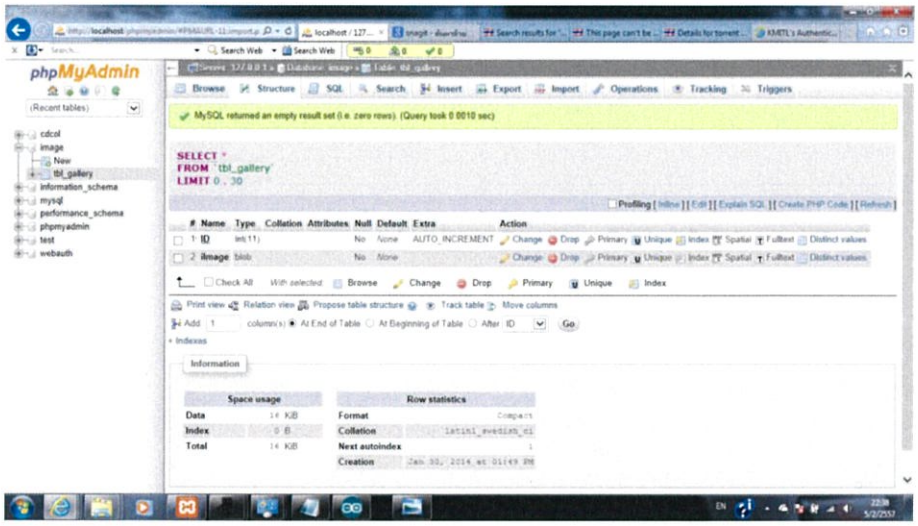
รูปที่ 4.13 ผลการทดลองที่ออกจาก serial monitor

จากรูปที่ 4.13 ผลการทดลองที่ออกจาก serial monitor แสดงรายละเอียดเมื่อมีการเชื่อมต่อกับเซิร์ฟเวอร์ว่ามีข้อมูลพร้อมติดต่อกับฝั่งเซิร์ฟเวอร์และทำการติดต่อสำเร็จ



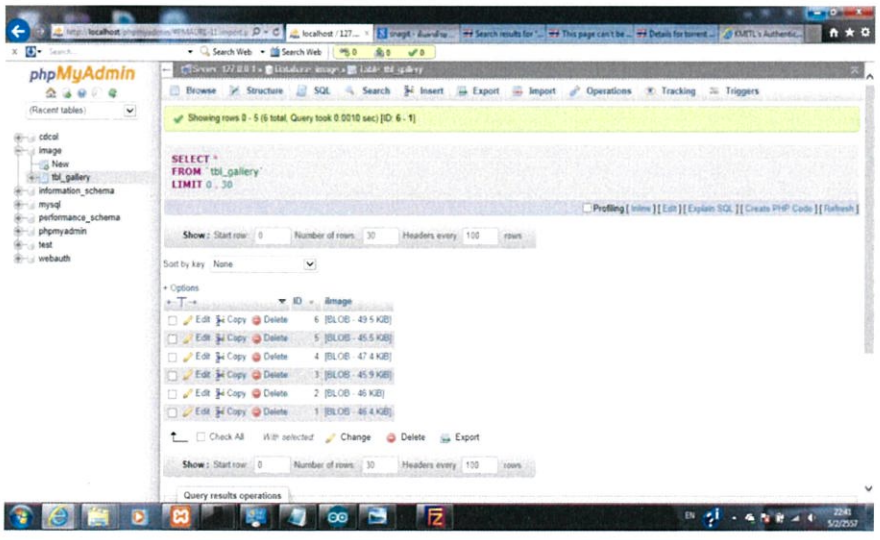
รูปที่ 4.14 ผลการทดลองในโปรแกรม FileZillaFTP เช็คสถานะการส่งข้อมูลเข้าสู่ฐานข้อมูล

จากรูปที่ 4.14 การทดลองเมื่อมีการมีการส่งข้อมูลเข้าสู่เซิร์ฟเวอร์ โดยใช้ FileZilla FTP ในการเช็คสถานะการส่ง จะบอกถึงรายละเอียดการติดต่อเมื่อมีการติดต่อสำเร็จและจะบอกถึงจำนวนไบต์ของรูปภาพที่ทำการส่งที่มุมด้านขวาล่าง



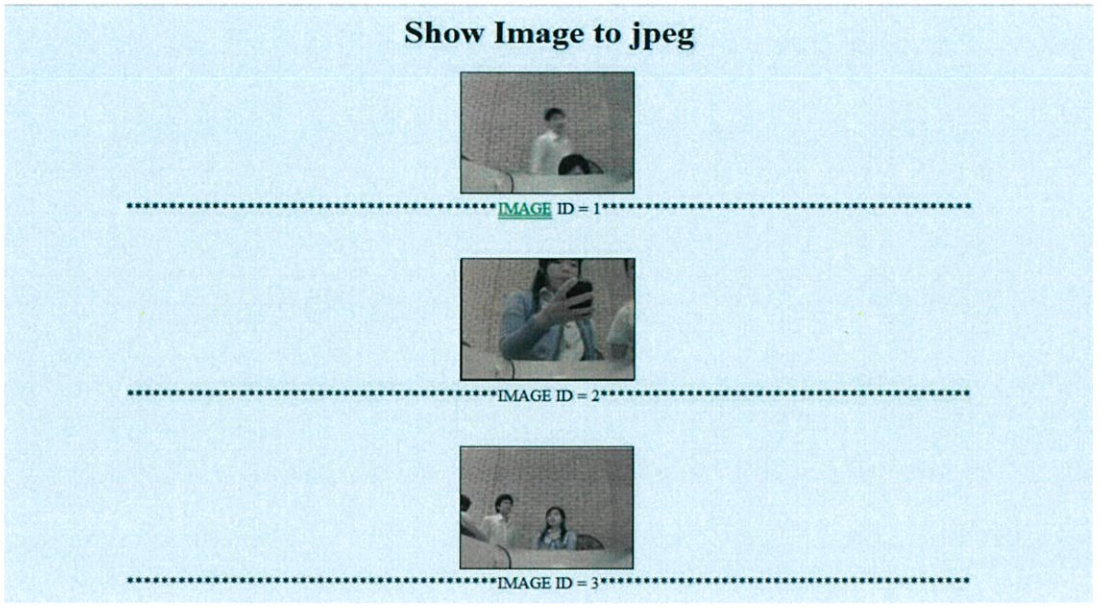
รูปที่ 4.15 เมื่อไม่มีการส่งข้อมูลเข้ามาในฐานข้อมูล

จากรูปที่ 4.15 แสดงถึงไม่มีข้อมูลเข้ามาในฐานข้อมูล พบว่าว่างเปล่าพร้อมรับข้อมูลเข้ามาเก็บไว้ต่อไป



รูปที่ 4.16 เมื่อมีข้อมูลเข้ามาในฐานข้อมูล ข้อมูลที่ได้รับคือไฟล์รูปภาพ

จากรูปที่ 4.16 ฐานข้อมูลที่ได้รับข้อมูลเข้ามา ซึ่งข้อมูลนั้นจะเป็นไฟล์รูปภาพที่ถูกถ่ายโดยโมดูลกล้อง จะเรียงตามลำดับของการถ่ายภาพ



รูปที่ 4.17 การแสดงผลบนหน้าเซิร์ฟเวอร์แสดงเป็นรูปภาพถ่ายโดยโมดูลกล้อง

รูปที่ 4.17 แสดงตัวอย่างการแสดงผลบนหน้าเซิร์ฟเวอร์แสดงเป็นรูปภาพถ่ายโดยโมดูลกล้อง เมื่อทำการส่งข้อมูลเข้าสู่เซิฟเวอร์แล้วนั้น จะสามารถทำการแสดงผลได้ที่หน้าเว็บเซิร์ฟเวอร์ เจ้าของบ้านหรือผู้ใช้งานสามารถเข้ามาตรวจสอบเมื่อมีการแจ้งเตือนข้อความสั้นผ่านโทรศัพท์มือถือแล้ว

บทที่ 5

สรุปผลและข้อเสนอแนะ

จากบทที่ได้กล่าวมาก่อนหน้านี้ เป็นเรื่องเกี่ยวกับขอบเขตและทฤษฎี เนื้อหา ผลการทดลองต่างๆที่เกี่ยวข้องกับโครงงาน และบทนี้จะกล่าวถึงการสรุปผลและข้อเสนอแนะต่างๆในการทำงาน

5.1 สรุปผล

จากการทดลองนั้น ใช้เซ็นเซอร์ในการทำงาน คือ เซ็นเซอร์ตรวจจับความเคลื่อนไหวและเซ็นเซอร์แถบแม่เหล็ก ทำการทดลองโดย

5.1.1 เซ็นเซอร์ตรวจจับความเคลื่อนไหว

วางเซ็นเซอร์ตรวจจับการเคลื่อนไหว ในบริเวณที่ต้องการตรวจจับ จ่ายไฟเลี้ยง 5 โวลต์ ให้กับเซ็นเซอร์ จากนั้นใช้ออสซิลโลสโคปวัดสัญญาณจะได้เป็นสัญญาณพัลส์ มีขนาด 5 โวลต์ตามไฟเลี้ยงที่ป้อนเข้าไป และทำการทดสอบขอบเขตของเซ็นเซอร์ตรวจจับการเคลื่อนไหว พบว่าที่ระยะ 1 เมตร สามารถตรวจจับได้ คิดเป็น 100 เปอร์เซ็นต์ และที่ระยะ 7 เมตร ไม่สามารถตรวจจับได้ คิดเป็น 0 เปอร์เซ็นต์

5.1.2 เซ็นเซอร์แถบแม่เหล็ก

วางอุปกรณ์เซ็นเซอร์แถบแม่เหล็กให้มีแรงดึงดูดกัน จ่ายไฟเลี้ยง 5 โวลต์ ให้กับเซ็นเซอร์ จากนั้นใช้ออสซิลโลสโคปวัดสัญญาณจะได้เป็นสัญญาณพัลส์แสดงออกเป็น 2 สถานะ คือ สถานะแรกเมื่อไม่ได้ทำการเปิดประตูทำให้เซ็นเซอร์มีแรงดึงดูดซึ่งกันและกัน ทำการวัดสัญญาณเอาท์พุทของเซ็นเซอร์ จะได้สัญญาณ มีขนาด 0 โวลต์ซึ่งแสดงว่าเซ็นเซอร์ไม่มีการตรวจจับ และสถานะที่สองเมื่อมีการเปิดประตูจะทำให้เซ็นเซอร์ไม่มีแรงดึงดูด ทำการวัดสัญญาณเอาท์พุทของเซ็นเซอร์ จะได้สัญญาณขนาด 4 โวลต์ซึ่งแสดงว่าเซ็นเซอร์มีการตรวจจับ ไปจนกว่าจะมีการปิดประตูอีกครั้ง ซึ่งจะให้สัญญาณเอาท์พุตขนาด 0 โวลต์ ออกมา

5.1.3 การทดสอบสัญญาณที่ออกจากโมดูลกล้อง

ส่งคำสั่งจากโปรแกรม Docklight เพื่อส่งคำสั่งที่ใช้ติดต่อกับโมดูลกล้อง ซึ่งคำสั่งที่ส่ง มีดังนี้

1. Take picture ไปยังโมดูลกล้อง เพื่อสั่งให้โมดูลจับภาพขณะนั้น
2. Read JPEG file size ไปยังโมดูลกล้อง เพื่อถามค่าขนาดจากโมดูล
3. Stop taking picture ไปยังโมดูลกล้อง เพื่อแจ้งโมดูลให้หยุดการเชื่อมต่อ

5.1.4 การส่งข้อความสั้นและแสดงบนเซิร์ฟเวอร์

เมื่อเซ็นเซอร์ตรวจจับสิ่งผิดปกติ ทำการส่งข้อความสั้น (SMS) คำว่า alarm Door เมื่อเซ็นเซอร์แถบแม่เหล็กแยกออกจากกันที่ประตู คำว่า alarm window เมื่อเซ็นเซอร์แถบแม่เหล็กแยกออกจากกันที่หน้าต่างและคำว่า alarm pir เมื่อเซ็นเซอร์ตรวจจับความเคลื่อนไหวตรวจจับการเคลื่อนไหวภายในห้องได้ไปยังผู้ใช้งานหรือเจ้าของบ้านเพื่อแจ้งเตือน หลังจากนั้นโมดูลกล้องทำการถ่ายภาพและบันทึกภาพลงเอสดีการ์ด (SD card) และจะสามารถตรวจสอบรูปภาพที่โมดูลกล้องทำการถ่ายภาพสิ่งผิดปกตินั้นบนเว็บเซิร์ฟเวอร์ได้

5.2 ข้อเสนอแนะ

จากการทดลองข้างต้นนั้นยังมีข้อบกพร่องในบางส่วนและมีข้อเสนอแนะ ดังนี้

1. การตรวจจับของเซ็นเซอร์ตรวจจับความเคลื่อนไหวยังไม่ครอบคลุมพื้นที่ได้ทั้งหมด ดังนั้นจึงควรติดตั้งเซ็นเซอร์เพิ่มในตำแหน่งอื่น เช่น บริเวณเพดานเพื่อให้ขอบเขตการตรวจจับครอบคลุมมากขึ้น
2. ควรทำการติดตั้งกล้องให้มีการทำงานแบบทันที (real time) เพื่อจับภาพผู้บุกรุกได้ตลอดเวลาและพร้อมตรวจสอบได้ทุกเมื่อ
3. เมื่อมีบุคคลเข้ามาไม่สามารถแยกแยะได้ว่าบุคคลนั้นเป็นผู้บุกรุกหรือไม่ ควรติดตั้งระบบที่มีการใส่รหัสผ่านก่อนเข้ามาในบริเวณนั้น เพื่อให้ทราบว่าบุคคลที่เข้ามาไม่ใช่ผู้บุกรุก

5.3 ปัญหาของโครงงาน

จากการผลการทดลองข้างต้น ยังมีความคลาดเคลื่อน เนื่องจาก

1. เซ็นเซอร์ตรวจจับความเคลื่อนไหวมีความบอบบางและเสียหายได้ง่าย ดังนั้นขณะทำการทดลองต้องใช้ความระมัดระวัง
2. การป้อนไฟจากแหล่งจ่ายให้กับเซ็นเซอร์ทั้ง 2 ตัว ควรใช้คนละแหล่งจ่ายเพื่อให้การทำงานของเซ็นเซอร์มีประสิทธิภาพ (ระบบไม่รวน)
3. พบว่าโมดูลกล้อง มีปัญหาเรื่องกราวด์ ในการรีเซตระบบใหม่จะต้องถอดกราวด์ออกแล้วต่อกราวด์ใหม่จึงจะสามารถทำงานได้
4. เซ็นเซอร์ตรวจแถบแม่เหล็ก ไม่ควรใช้กราวด์ร่วมกับอุปกรณ์ตัวอื่น

บรรณานุกรม

[1] ประจิน พลังสันติกุล.*การเขียนโปรแกรมควบคุมไมโครคอนโทรเลอร์ AVR ด้วยภาษา C กับ WinAVR (C Complier).*กรุงเทพฯ: แอพซอฟต์เทค, 2549.

[2] ThaiEasyElec.2010.*แนะนำเพื่อนใหม่ที่ชื่อ Arduino.*กรุงเทพฯ. http://www.thaieasyelec.com/basic-electronics/บทความ-Arduino-ตอนที่1-แนะนำเพื่อนใหม่ที่ชื่อ-Arduino.html

[3] เอกชัย มะการ.*เรียนรู้เข้าใจใช้งานไมโครคอนโทรเลอร์ตระกูล AVR ด้วย Ardunio.*กรุงเทพฯ: บริษัท อีทีที จำกัด , 2552

[4] ชิสทรอนิกส์.*คุณรู้หรือไม่ "รังสีอินฟราเรด" คืออะไร.*เข้าถึงได้จาก http://thailandthermography.igetweb.com/index.php?mo=3&art=155648

[5] 2556.*การตรวจจับการเคลื่อนไหวแบบไฟโร (Pyroelectric sensor).*http://bme3g7.blogspot.com/p/blog-page_7954.html

[6] 2556. *เซ็นเซอร์ตรวจจับความเคลื่อนไหว (Motion Sensor).*เข้าถึงได้จาก http://application-with-embedded-linux.blogspot.com/2010/12/motion-sensor.html

[7] อารัมภีย์ จันทร์ใย. "เทคนิคการเชื่อมต่อ SD/MMC."*เซมิคอนดักเตอร์อิเล็คทรอนิกส์.* (สิงหาคม 2552): 141-149.

[8] ThaiEasyElec.2010.*การใช้งานเบื้องต้น JPEG Camera Module (VC706).*เข้าถึงได้จาก http://www.thaieasyelec.net/archives/Manual/_ESEN113_%20Linksprite%20-%20Serial%20JPEG%20Camera%20Module.pdf

[9] wikispaces. 2556. *Arduino Duemilanove / YourDuino328.*เข้าถึงได้จาก http://arduino-info.wikispaces.com/QuickRef

[10] บริษัท อีทีที จำกัด.2556. *ET-GSM SIM300CZ V1.0.*เข้าถึงได้จาก http://www.ett.co.th/product/intf/ET-GSM-SIM300CZ-V1.0.html

[11] 2556. *OSI Model.*เข้าถึงได้จาก http://cptd.chandra.ac.th/selfstud/datacom2/Contents/ Chapters/OSI-Model.htm

[12] 2556.*บทที่ 15 การรับส่งไฟล์ (FTP).*เข้าถึงได้จาก http://www.elearning.ac.th/~boonchoay/Network/lesson15.htm

[13] 2556.*กรมวิทยาศาสตร์และเทคโนโลยีกลาโหม.*เข้าถึงได้จาก http://www.detd.mi.th/board/index/php?topic=474.0

บรรณานุกรม (ต่อ)

[14] Knowledge base.2556. *การติดตั้งและใช้งาน XAMPP เบื้องต้น*.เข้าถึงได้จาก
http://www.phar. ubu.ac.th/km/?p=350

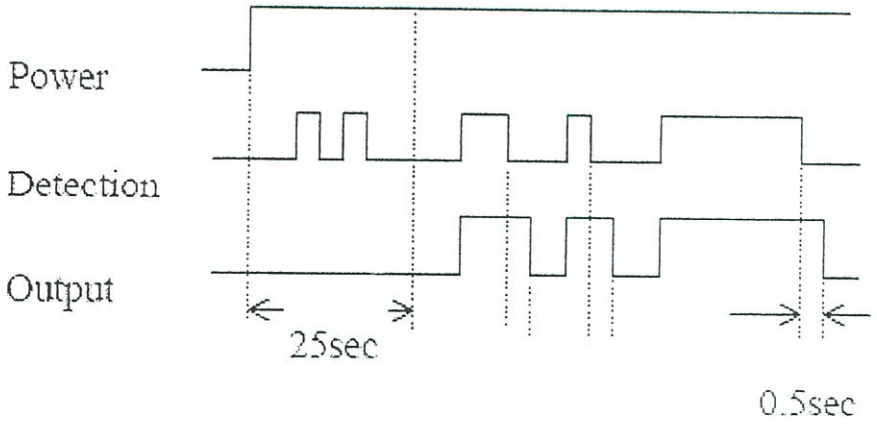
[15] 2556.*DNS คือ*.เข้าถึงได้จาก
http://guru.google.co.th/guru/thread?tid=5756aead6742da22

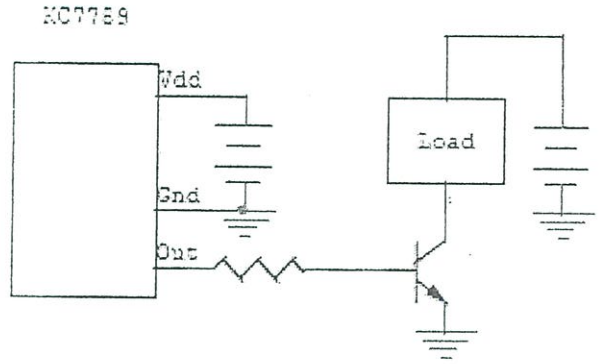
[16] 2556.*ความรู้ในการ Forward Port*.เข้าถึงได้จาก
http://www.mastersatcom.com/pages/how-to-forward-port.html

ภาคผนวก ก

PIR Motion Sensor Module - KC7783R

Timing Chart



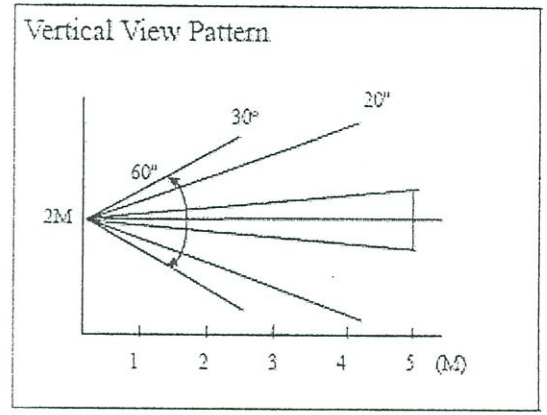


Wiring Diagram

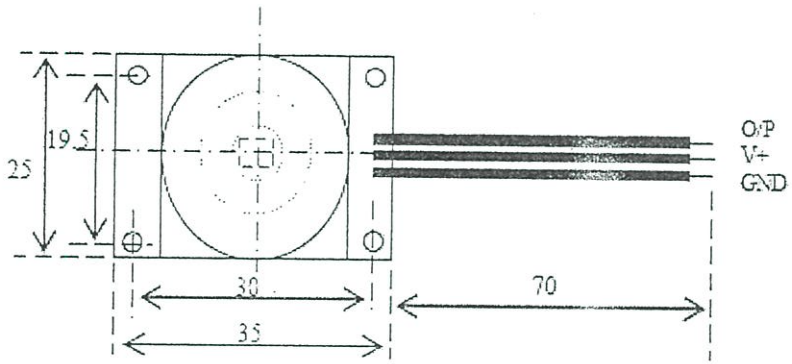
Specification

	Min	Typ	Max	Unit
Operation Voltage	4.7	5	12	V
Standby Current (no load)		300		μA
Output Pulse Width	0.5			Sec
Output High Voltage		5		V
Detection Range		5		M
Operation Temperature	-20	25	50	°C
Humidity Range			95	%

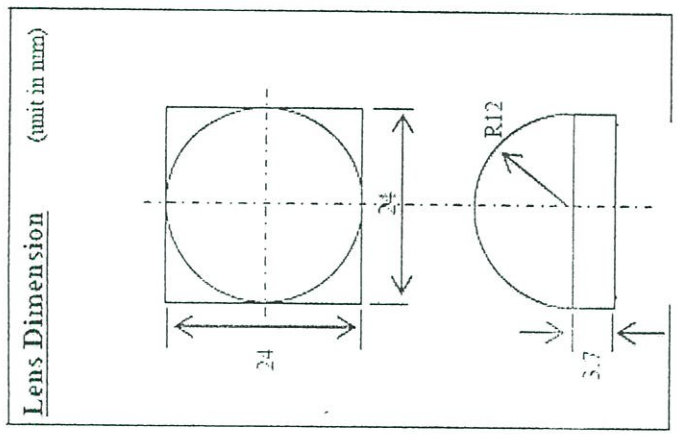
Note: 1. All other features and specification, please refer to KC778B
2. Minimum output pulse width can be customer specified.



Mechanical Dimension



Front View



ภาคผนวก ข

โค้ดที่ใช้ในการทำงาน

```
#include <Adafruit_VC0706.h>
#include <SD.h>
#include <SoftwareSerial.h>
#include <SPI.h>
#include <Ethernet.h>
#define FTPWRITE
byte mac[] = { 0x90, 0xA2, 0xDA, 0x00, 0xBD, 0xB9 };
inti,a;
IPAddressip( 192, 168, 1, 5 );
IPAddress gateway( 192, 168, 1, 80 );
IPAddress subnet( 255, 255, 255, 0 );
IPAddress server( 192, 168, 1, 80 );
EthernetClient client;
EthernetClientdclient;
EthernetClientpclient;
char outBuf[128];
char outCount;
char filename[13];
intledPin =8;
int inputPin1 = 59;
int inputPin2 = 58;
int inputPin3 = 57;
int inputPin4 = 7;
int pirState1 =0;
int pirState2 =0;
int pirState3 =0;
int pirState4 = LOW;
int val1 = 0;
int val2 = 0;
```

```
int val3 = 0;
int val4 = 0;


constintchipSelect = 4;
SoftwareSerialcameraconnection(69, 3);
Adafruit_VC0706 cam = Adafruit_VC0706(&cameraconnection);
void setup(){
Ethernet.begin(mac, ip, gateway, gateway, subnet);
digitalWrite(10,HIGH);
pinMode(10,OUTPUT);
pinMode(ledPin,OUTPUT);
pinMode(inputPin1,INPUT);
pinMode(inputPin2,INPUT);
pinMode(inputPin3,INPUT);
pinMode(inputPin4,INPUT);
Serial.begin(115200);
  delay(1000);
Serial.print("AT+IFC=1,1\r");
  delay(1000);
Serial.print("AT+CMGF=1\r");
  if (!SD.begin(chipSelect)) {
Serial.println("Card failed, or not present");
        return;
  }
    if (cam.begin()) {
Serial.println("Camera Found:");
  } else {
Serial.println("No camera found?");
```

```
    return;
     }
      char *reply = cam.getVersion();
  if (reply == 0) {
Serial.print("Failed to get version");
  } else {
Serial.println("-----------------");
Serial.print(reply);
Serial.println("-----------------");
  }
cam.setImageSize(VC0706_160x120);
     uint8_t imgsize = cam.getImageSize();
Serial.println("Image size: 160x120");
  }
void loop(){
  val1 = analogRead(inputPin1);
   if (val1 <1){
digitalWrite(ledPin,LOW);
      if (pirState1 >0){
         pirState1 =0;
      }
   }
   else
    {
digitalWrite(ledPin, HIGH  );
    if (pirState1 <1){
Serial.println("alarm magnetic1 ");
      pirState1 = 1;
       delay(1000);
```

```
Serial.print("AT+CMGS=\"+66843560552\"\r");
delay(1000);
Serial.print("alarm window1\r");
delay(1000);
Serial.write(0x1A);
delay(1000);
    }
    }
    val2 = analogRead(inputPin2);
   if (val2 <1){
digitalWrite(ledPin,LOW);
      if (pirState2 >0){
        pirState2 =0;
      }
   }
   else
    {
digitalWrite(ledPin, HIGH  );
    if (pirState2 <1){
Serial.println("alarm magnetic2 ");
      pirState2 = 1;
      delay(1000);
Serial.print("AT+CMGS=\"+66843560552\"\r");
delay(1000);
Serial.print("alarm Door \r");
delay(1000);
Serial.write(0x1A);
        delay(1000);
takepic();
```

```
  delay(1000);
    }
    }
    val3 = analogRead(inputPin3);
   if (val3 <1){
digitalWrite(ledPin,LOW);
      if (pirState3 >0){
         pirState3 =0;
      }
   }
   else
    {
digitalWrite(ledPin, HIGH  );
    if (pirState3 <1){
Serial.println("alarm magnetic3 ");
      pirState3 = 1;
      delay(1000);
Serial.print("AT+CMGS=\"+66843560552\"\r");
delay(1000);
Serial.print("alarm window2\r");
Serial.write(0x1A);
delay(1000);
    }
    }
     val4 = digitalRead(inputPin4);
   if(val4 == HIGH){
digitalWrite(ledPin,HIGH);
    if (pirState4 == LOW){
Serial.println("Alarm pir motion");
```

```
    pirState4 = HIGH;
        delay(1000);
Serial.print("AT+CMGS=\"+66843560552\"\r");
delay(1000);
Serial.print("alarm zonepir\r");
delay(1000);
Serial.write(0x1A);
    delay(1000);
takepic();
  delay(1000);
  }
  }
  else{
digitalWrite(ledPin,LOW);
  if (pirState4 == HIGH){
    pirState4 = LOW;
    }
   }
}

void takepic(){
int e;
  for(e=0;e<3;e++){
Serial.println("Snap in 1 secs...");
  delay(1000);
if (! cam.takePicture())
Serial.println("Failed to snap!");
  else
Serial.println("Picture taken!");
```

```
strcpy(filename, "IMAGE00.JPG");
  for (int i = 0; i < 1000; i++) {
    filename[5] = '0' + i/10;
    filename[6] = '0' + i%10;
      if (! SD.exists(filename)) {
      break;
    }
  }
   File imgFile = SD.open(filename, FILE_WRITE);
    uint16_t jpglen = cam.frameLength();
Serial.print("Storing ");
Serial.print(jpglen, DEC);
Serial.print(" byte image.");
   int32_t time = millis();
   byte wCount = 0;
  while (jpglen> 0) {
      uint8_t *buffer;
    uint8_t bytesToRead = min(32, jpglen);
    buffer = cam.readPicture(bytesToRead);
imgFile.write(buffer, bytesToRead);
    if(++wCount>= 64) {
Serial.print('.');
wCount = 0;
    }
jpglen -= bytesToRead;
  }
imgFile.close();
Serial.println("Success");
   delay(5000);
```

```
  byte inChar;
Serial.println(filename);
inChar = 'f';
  if(inChar == 'f')
  {
    if(doFTP()) Serial.println(F("FTP OK"));
    else Serial.println(F("FTP FAIL"));
  }
  if(inChar == 'r')
  {
readSD();
  }
  delay(1000);
  }
}
File fh;
byte doFTP()
{
#ifdef FTPWRITE
fh = SD.open(filename,FILE_READ);
#else
SD.remove(filename);
fh = SD.open(filename,FILE_WRITE);
#endif
  if(!fh)
  {
Serial.println(F("SD open fail"));
    return 0;
  }
```

```
#ifndef FTPWRITE
  if(!fh.seek(0))
  {
Serial.println(F("Rewind fail"));
fh.close();
    return 0;
  }
#endif
Serial.println(F("SD opened"));
  if (client.connect(server,21)) {
Serial.println(F("Command connected"));
  }
  else {
fh.close();
Serial.println(F("Command connection failed"));
    return 0;
  }
  if(!eRcv()) return 0;
client.println(F("USER user1\r"));
  if(!eRcv()) return 0;
client.println(F("PASS 123456\r"));
  if(!eRcv()) return 0;
client.println(F("SYST\r"));
  if(!eRcv()) return 0;
client.println(F("PASV\r"));
  if(!eRcv()) return 0;
  char *tStr = strtok(outBuf,"(,");
intarray_pasv[6];
  for ( int i = 0; i < 6; i++) {
```

```
tStr = strtok(NULL,",");
array_pasv[i] = atoi(tStr);
    if(tStr == NULL)
    {
Serial.println(F("Bad PASV Answer"));

    }
  }
  unsigned inthiPort,loPort;
hiPort = array_pasv[4] << 8;
loPort = array_pasv[5] & 255;
Serial.print(F("Data port: "));
hiPort = hiPort | loPort;
Serial.println(hiPort);
  if (dclient.connect(server,hiPort)) {
Serial.println(F("Data connected"));
  }
  else {
Serial.println(F("Data connection failed"));
client.stop();
fh.close();
    return 0;
  }
#ifdef FTPWRITE
Serial.println(filename);
client.print(F("STOR "));
client.println(filename);
#else
client.print(F("RETR "));
```

```
client.println(filename);
#endif
  if(!eRcv())
  {
dclient.stop();
    return 0;
  }
#ifdef FTPWRITE
Serial.println(F("Writing"));
  byte clientBuf[64];
intclientCount = 0;
  while(fh.available())
  {
clientBuf[clientCount] = fh.read();
clientCount++;
    if(clientCount> 63)
    {
dclient.write(clientBuf,64);
clientCount = 0;
    }
  }
  if(clientCount> 0) dclient.write(clientBuf,clientCount);
#else
  while(dclient.connected())
  {
    while(dclient.available())
    {
      char c = dclient.read();
fh.write(c);
```

```
Serial.write(c);
    }
  }
#endif
dclient.stop();
Serial.println(F("Data disconnected"));
  if(!eRcv()) return 0;
client.println(F("QUIT"));
  if(!eRcv()) return 0;
client.stop();
Serial.println(F("Command disconnected"));

fh.close();
Serial.println(F("SD closed"));
     if(pclient.connect(server, 80)>0) {  // Se conecta al servidor
     delay(1000);
pclient.print("GET http://192.168.1.80/phpSaveBlob.php?pname="); // Envia los
datosutilizando GET
pclient.print(filename);
Serial.println(filename);
   // pclient.print("&ch1="); // Envia los datosutilizando GET
   // pclient.print(dht11_dat[0], DEC);
pclient.println(" HTTP/1.0");
pclient.println("User-Agent: Arduino 1.0");
pclient.println();
Serial.println("Transmission is completed");
  }
  else
  {
```

```
Serial.println("Transmission is Faill");
  }
  if (pclient.connected()) {}
  else {
Serial.println("Unable to connect to server");
  }
pclient.stop();
pclient.flush();

  delay(10000);
  return 1;
}
byte eRcv()
{
  byte respCode;
  byte thisByte;
  while(!client.available()) delay(1);
respCode = client.peek();
outCount = 0;
  while(client.available())
  {
thisByte = client.read();
Serial.write(thisByte);
    if(outCount< 127)
    {
outBuf[outCount] = thisByte;
outCount++;
outBuf[outCount] = 0;
    }
```

```
  }
  if(respCode>= '4')
  {
efail();
    return 0;
  }
  return 1;
}
void efail()
{
  byte thisByte = 0;
client.println(F("QUIT"));
  while(!client.available()) delay(1);
  while(client.available())
  {
thisByte = client.read();
Serial.write(thisByte);
  }

client.stop();
Serial.println(F("Command disconnected"));
fh.close();
Serial.println(F("SD closed"));
}
void readSD()
{
fh = SD.open(filename,FILE_READ);
  if(!fh)
  {
```

```
Serial.println(F("SD open fail"));
    return;
  }
  while(fh.available())
  {
Serial.write(fh.read());
  }
fh.close();
}
```