

เซนเซอร์ตรวจจับขโมยแบบไร้สาย

Wireless Thief Sensor



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์

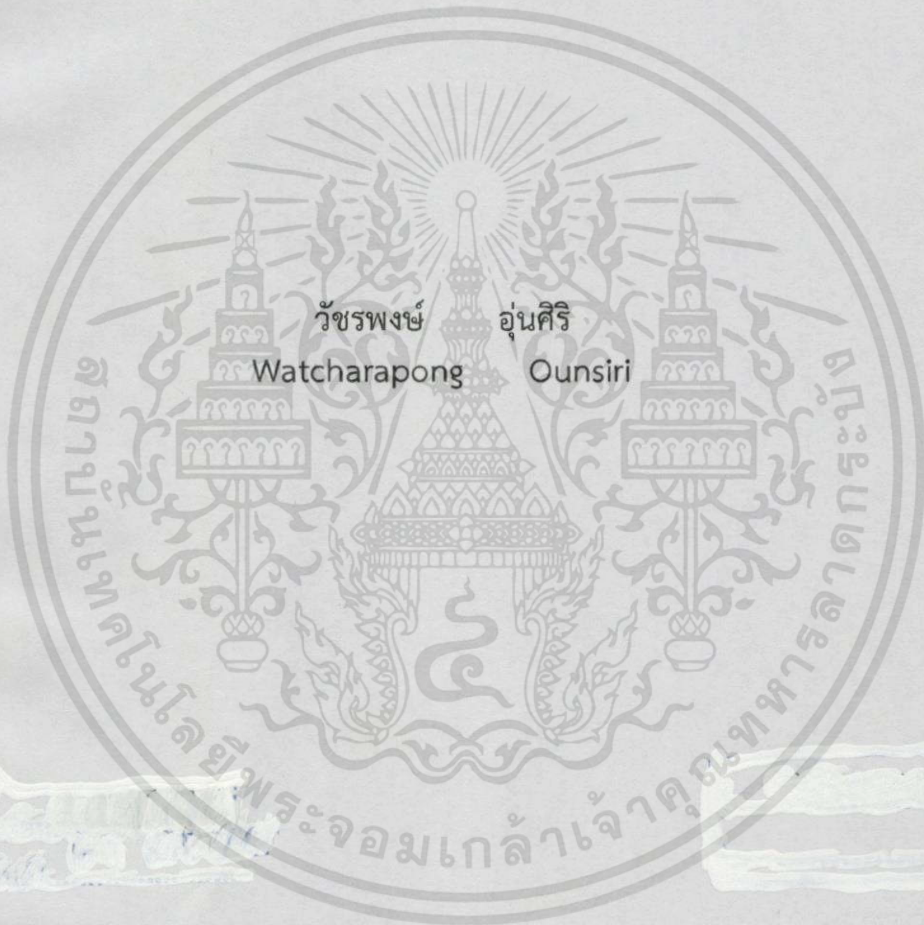
คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2556

เซนเซอร์ตรวจจับขโมยแบบไร้สาย

Wireless Thief Sensor



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์  
คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2556

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เซนเซอร์ตรวจจับขโมยแบบไร้สาย

Wireless Thief Sensor



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2556

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ปีการศึกษา 2556  
สาขาวิชา วิศวกรรมอิเล็กทรอนิกส์  
คณะ วิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
เรื่อง เซนเซอร์ตรวจจับขโมยแบบไร้สาย  
Wireless Thief Sensor  
ผู้จัดทำ นายวัชรพงษ์ อุ๋นศิริ รหัส 52011084

ปริญญาานิพนธ์นี้ผ่านการตรวจสอบโดยอาจารย์ที่ปรึกษาแล้ว



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์

นักศึกษา

ปริญญา

สาขาวิชา

ปีการศึกษา

อาจารย์ที่ปรึกษาปริญญานิพนธ์

เซนเซอร์ตรวจจับขโมยแบบไร้สาย

นายวัชรพงษ์ อุ่นศิริ 52011084

วิศวกรรมศาสตรบัณฑิต

วิศวกรรมอิเล็กทรอนิกส์

2556

ดร.อิทธิภูมิ บุญพิคำ

### บทคัดย่อ

โครงการอุปกรณ์ตรวจจับขโมยแบบไร้สายนี้ มีขึ้นเพื่อเฝ้าระวังเหตุการณ์ที่ผิดปกติที่อาจเกิดได้ภายในที่พักอาศัย ในขณะที่ไม่มีผู้อาศัยในที่พักอาศัยนั้น ๆ โดยการส่งสัญญาณแจ้งเตือนในรูปแบบข้อความสั้นผ่านทางโทรศัพท์เคลื่อนที่ ซึ่งมีหลักการทำงานดังนี้ เมื่อมีผู้บุกรุกผ่านย่านการตรวจจับของอุปกรณ์ตรวจจับขโมยแบบไร้สายนี้ เซนเซอร์จะเป็นตัวรับสัญญาณการเคลื่อนไหวของผู้บุกรุก และจะทำการประมวลผลด้วยไมโครคอนโทรลเลอร์ Arduino Module โดยไมโครคอนโทรลเลอร์ชนิดนี้จะทำงานร่วมกับ GSM Module ซึ่งสามารถประมวลผลคำสั่งเพื่อสั่งการให้ GSM Module สามารถส่งข้อความแจ้งเตือนไปยังหมายเลขโทรศัพท์ของเจ้าของบ้านหรือเจ้าของสถานที่นั้น ๆ ได้ โดยอุปกรณ์ที่นำมาใช้ในโครงการนี้ก็เป็นอุปกรณ์ที่ใช้กันอย่างแพร่หลาย โดยเฉพาะ Arduino Uno R3 ที่เนื่องจากราคาที่สมเหตุสมผลกับประสิทธิภาพแล้ว ยังมีโค้ดที่สามารถนำไปใช้งานได้โดยไม่ต้องเสียค่าใช้จ่ายใด ๆ เพิ่มเติม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	Wireless Thief Sensor
Student	Mr.Watcharapong Ounsiri Student ID 52011084
Degree	Bachelor of Engineering
Program	Electronics Engineering
Year	2013
Thesis Advisor	Dr. Ittibhoom Boonpikum

### Abstract

This project is entitled "The Wireless Sensor for Thieves Detection". It is the tool that detects an abnormal incident which may happen in your accommodation. By sending alert in SMS form via your mobile phone. The principle is when intruders move through the detector, the sensor can detect moving. After detection the sensor will receive the signal and process by the microcontroller, Arduino Module, this microcontroller is connect to GSM Module which can send the alert to the home owner devices. This project uses the microcontroller, Arduino Uno R3, control the system because it is widely used and there are many opensource platform that can download without payment.

## กิตติกรรมประกาศ

โครงการนี้จะสำเร็จลุล่วงไปด้วยดีมิได้ หากไม่ได้รับความอนุเคราะห์อย่างยิ่งจากท่านอาจารย์ที่ปรึกษาคือ ดร.อิทธิภูมิ บุญพิศา ที่ได้ให้คำแนะนำ คำสั่งสอน ให้ความรู้ และความช่วยเหลือต่างๆ ตลอดระยะเวลาในการทำโครงการ

ขอขอบพระคุณทุกท่านในการช่วยเหลือและมอบความห่วงใยให้ ขอขอบคุณคณาจารย์ประจำภาควิชาอิเล็กทรอนิกส์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ทุกท่านที่ได้อบรมสั่งสอน และประสิทธิ์ประสาทวิชาความรู้ให้แก่ข้าพเจ้า ขอขอบคุณผู้ที่มีส่วนเกี่ยวข้องทั้งหลายที่ได้เอ่ยถึงมา ณ ที่นี้ ที่ให้กำลังใจแก่ข้าพเจ้าเสมอมา จนกระทั่งทำให้โครงการในครั้งนี้สำเร็จลุล่วงไปได้ด้วยดี ความดีอันเกิดจากการทำโครงการในครั้งนี้ ข้าพเจ้าขอมอบแต่บิดา มารดา ครูบาอาจารย์ เพื่อน พี่ น้องนักศึกษาและผู้มีพระคุณทุกท่าน

วัชรพงษ์ อุ่นศิริ  
ผู้จัดทำ



# สารบัญ

	หน้า
บทคัดย่อ .....	I
Abstract .....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VI
สารบัญรูป .....	VII
<b>บทที่ 1 บทนำ.....</b>	<b>1</b>
1.1 ความเป็นมาและความสำคัญของปัญหา .....	1
1.2 วัตถุประสงค์ของการศึกษา.....	1
1.3 สมมติฐานของการศึกษา.....	1
1.4 ขอบเขตการวิจัย.....	1
1.5 ประโยชน์ที่คาดว่าจะได้รับ .....	1
<b>บทที่ 2 หลักการและทฤษฎีที่เกี่ยวข้อง .....</b>	<b>2</b>
2.1 เซนเซอร์ (Sensor).....	2
2.2 รังสีอินฟราเรด (Infrared).....	2
2.2.1 อินฟราเรดเซนเซอร์แบบพาสซีฟ (PIR Sensor).....	3
2.3 ไมโครคอนโทรลเลอร์ (Microcontroller).....	3
2.3.1 อาร์ดูอโน (Arduino Module).....	3
2.4 GSM Modem .....	4
2.5 แหล่งจ่ายไฟตรง (DC Power Supply).....	5
2.5.1 แหล่งจ่ายแรงดันไฟฟ้ากระแสตรงแบบเชิงเส้น (Linear Power Supply).....	5
2.5.2 แหล่งจ่ายแรงดันไฟฟ้ากระแสตรงแบบสวิตซ์ซิ่ง (Switching Power Supply).....	5
<b>บทที่ 3 การออกแบบและการจัดทำโครงงาน .....</b>	<b>7</b>
3.1 การออกแบบ.....	7
3.1.1 ส่วนประกอบของระบบ.....	7
3.2 การรับสัญญาณด้วยเซนเซอร์.....	7
3.2.1 อินฟราเรดเซนเซอร์แบบพาสซีฟ (PIR Sensor).....	7
3.3 การออกแบบการส่งสัญญาณแจ้งเตือน .....	9
3.3.1 รายละเอียดของภาษาที่ใช้ในการโปรแกรมคำสั่งและ Arduino UNO R3.....	9
3.3.2 รายละเอียดของส่วนต่างๆ.....	11
3.3.3 GSM Modem.....	13
<b>บทที่ 4 ผลการทดลอง.....</b>	<b>14</b>
4.1 การทดสอบบอร์ด Arduino Uno R3 ก่อนการใช้งาน .....	14
4.2 ทดลองการรับสัญญาณด้วยเซนเซอร์ .....	14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ (ต่อ)

	หน้า
4.3 ระยะการทำงานของ PIR Sensor .....	18
4.4 ส่งสัญญาณแจ้งเตือนในรูปแบบของ SMS.....	18
4.5 การแจ้งเตือนไปยังเลขหมายอื่น.....	19
บทที่ 5 บทสรุป.....	20
บรรณานุกรม .....	21
ภาคผนวก.....	22



# สารบัญตาราง

หน้า

ตาราง 4.1 ผลการทดลองย่านการทำงานของ PIR Sensor .....	18
--	----



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูป

รูปที่	หน้า
2.1 คลื่นแม่เหล็กไฟฟ้าในช่วงความยาวคลื่นต่างๆ.....	2
2.2 อินฟราเรดเซนเซอร์แบบพาสซีฟ (PIR Sensor).....	3
2.3 Arduino Uno R3.....	4
2.4 GSM Module Siemens TC35.....	4
2.5 วงจรจ่ายไฟตรงจากไฟบ้าน.....	5
3.1 ส่วนประกอบของระบบ .....	7
3.2 การตรวจจับของอินฟราเรดเซนเซอร์.....	8
3.3 วงจรของ PIR Motion Sensor .....	8
3.4 บอร์ด Arduino Uno R3.....	10
3.5 คุณสมบัติทั่วไปของ ATmega328 .....	10
3.6 การต่อบอร์ด Arduino เข้ากับ GSM modem .....	13
3.7 รูปของ GSM Module Siemens TC35 .....	13
4.1 โปรแกรม Arduino Version 1.5.2.....	14
4.2 หน้าเริ่มต้นของโปรแกรม Arduino .....	14
4.3 แสดงการเข้าสู่คำสั่ง Blink .....	15
4.4 ตรวจสอบความถูกต้องของคำสั่ง.....	15
4.5 การอัปเดตโปรแกรมลงบอร์ด.....	16
4.6 วงจรที่ใช้ในการทดลอง.....	17
4.7 ขณะที่ยังไม่เกิดการเคลื่อนที่ของวัตถุ.....	17
4.8 หลอดไฟLED สว่างเมื่อวัตถุเคลื่อนไหวในย่านการตรวจจับ.....	18
4.9 เครื่องตรวจจับขโมยนี้ทำการส่ง SMS เพื่อแจ้งเตือนการบุกรุก.....	18
4.10 แสดงการแจ้งเตือนไปยังเลขหมายอื่น.....	19

# บทที่ 1

## บทนำ

### 1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันการโจรกรรมและอาชญากรรมถือเป็นปัญหาที่สำคัญปัญหาหนึ่งซึ่งส่งผลกระทบต่อการพัฒนาประเทศในแง่เศรษฐกิจเพราะอุปกรณ์หรือเครื่องมือเครื่องใช้ที่ใช้ป้องกันการโจรกรรม ส่วนใหญ่จะต้องนำเข้าเทคโนโลยีถึงแม้ว่าอุปกรณ์จะผลิตได้ในประเทศแต่เมื่อประกอบออกมาเป็นผลิตภัณฑ์แล้ว มูลค่าของผลิตภัณฑ์จะถูกเพิ่มขึ้นอีกมากซึ่งมาจากการที่เราจะต้องเสียค่าลิขสิทธิ์ทางปัญญาให้กับผู้ผลิตรายนั้นๆ ซึ่งก็เป็นผู้ประกอบการชาวต่างชาติเสียเป็นส่วนใหญ่ สำหรับโครงการนี้ผู้จัดทำได้ทำการศึกษาถึงวิธีการที่จะใช้ทฤษฎีต่างๆ เพื่อลดการปัญหา

ดังนั้นผู้จัดทำโครงการนี้จึงได้นำเสนอโครงการเซนเซอร์ตรวจจับขโมยแบบไร้สาย โดยการตรวจจับการเคลื่อนไหวด้วย PIR Sensor ประมวลผลการเปลี่ยนแปลงของสัญญาณด้วย Arduino Uno R3 และส่งการแจ้งเตือนผ่าน GSM Modem ในรูปของ SMS ( Short Message Service )

### 1.2 วัตถุประสงค์ของการศึกษา

1.2.1 สามารถนำความรู้ที่ได้ศึกษาในชั้นปีที่ 1-4 มาประยุกต์ใช้ในการทำโครงการในวิชา Project 1 และ 2 ซึ่งเป็นไปตามหลักสูตรของนักศึกษาชั้นปีที่ 4

1.2.2 สามารถทำให้อุปกรณ์ที่ใช้ตรวจจับค่าสัญญาณที่เปลี่ยนแปลงแล้วสามารถส่งสัญญาณแจ้งเตือนได้ในรูปแบบ SMS ในโทรศัพท์เคลื่อนที่

1.2.3 สามารถเชื่อมต่ออุปกรณ์ในแต่ละส่วนให้ทำงานได้ตามที่เราต้องการโดยการโปรแกรมด้วยภาษาทางคอมพิวเตอร์ ( Computer Language )

### 1.3 สมมติฐานของการศึกษา

เมื่อมีผู้บุกรุกผ่านเข้ามาในย่านการตรวจจับ เซนเซอร์ชนิด Passive Infrared จะเกิดการเปลี่ยนแปลงลอจิกในโหมดที่กำหนด Arduino จะทำการประมวลผลและส่งให้ GSM Module ทำการส่งการแจ้งเตือนในรูปของข้อความผ่านทางโทรศัพท์เคลื่อนที่

### 1.4 ขอบเขตการวิจัย

งานวิจัยนี้จะครอบคลุมการป้องกันการบุกรุกที่จะเกิดขึ้นเมื่อห้อง,บ้านหรืออาคารนั้น เซนเซอร์ที่เราทำการติดตั้งไว้ จะสามารถแจ้งเตือนได้อย่างทันที

### 1.5 ประโยชน์ที่คาดว่าจะได้รับ

สามารถนำความรู้จากงานนี้ไปต่อยอดในการคิดค้นสิ่งประดิษฐ์ที่เป็นประโยชน์อื่นๆได้ต่อไป

## บทที่ 2

# หลักการและทฤษฎีที่เกี่ยวข้อง

### 2.1 เซนเซอร์ (Sensor)

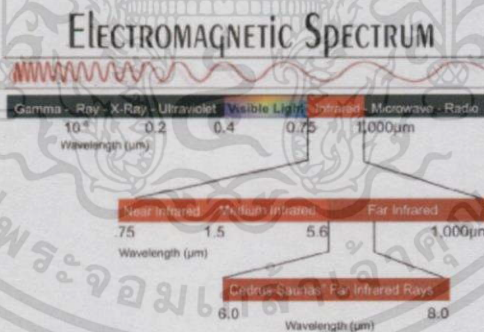
เซนเซอร์ (Sensor) คือตัวอุปกรณ์ตรวจวัดตัวแรกในระบบการวัด ซึ่งใช้ตรวจจับหรือรับรู้การเปลี่ยนแปลงปริมาณทางกายภาพของตัวแปรต่างๆ เช่น ความร้อน แสง สี เสียง ระยะทาง การเคลื่อนที่ ความดัน การไหล เป็นต้น แล้วเปลี่ยนให้อยู่ในรูปของสัญญาณหรือข้อมูลที่สอดคล้องและเหมาะสมกับส่วนของการกำหนดเงื่อนไขทางสัญญาณ โดยที่ชนิดของเซนเซอร์แบ่งโดยอาศัยหลักเกณฑ์ต่างๆ เช่น แบ่งตามความต้องการพลังงาน ดังนี้

1.แบบแอคทีฟ (Active sensors) เป็นทรานสดิวเซอร์ที่สามารถปล่อยพลังงานเองได้ เช่น เทอร์โมคัปเปิ้ล เทียโซ เซลล์แสงอาทิตย์ ออปโตไดโอด เป็นต้น อุปกรณ์เหล่านี้ไม่ต้องมีแหล่งจ่ายกำลังจากภายนอกให้ก็สามารถให้สัญญาณแรงดันหรือกระแสที่แปรตามตัวแปรได้เอง

2.แบบพาสซีฟ (Passive sensors) แบบนี้จะต้องใช้แหล่งจ่ายจากภายนอกจึงจะทำการตรวจวัดได้ เช่น เซนเซอร์ที่ใช้หลักการเปลี่ยนค่าความต้านทาน ค่าความจุ ค่าความเหนียว ฯลฯ เป็นต้น

### 2.2 รังสีอินฟราเรด (Infrared)

รังสีอินฟราเรด หรือ รังสีใต้แดงเป็นคลื่นแม่เหล็กไฟฟ้าชนิดหนึ่งที่มีความถี่อยู่ระหว่าง 10<sup>11</sup>-10<sup>14</sup> Hz และมีความยาวคลื่นระหว่าง 0.75 ไมโครเมตรถึง 1000 ไมโครเมตร สสารที่มีอุณหภูมิมากกว่า 0 องศาเคลวิน จะปล่อยรังสีอินฟราเรดออกมาจากตัวมันเองเสมอ

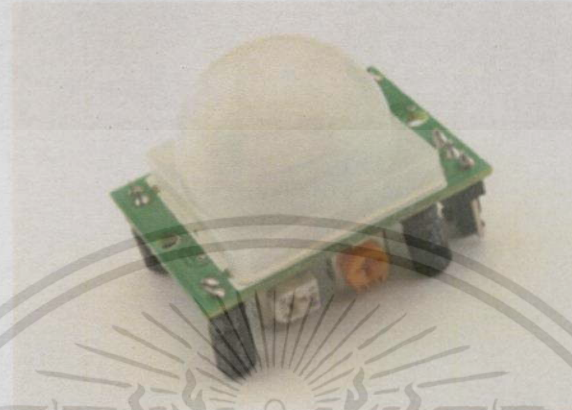


รูปที่ 2.1 คลื่นแม่เหล็กไฟฟ้าในช่วงความยาวคลื่นต่างๆ

จากรูปแสดงคลื่นแม่เหล็กไฟฟ้าในช่วงความยาวคลื่นต่างๆ จะเห็นได้ว่ารังสีอินฟราเรดจะมีความยาวคลื่นที่มากกว่าแสงที่ตามองเห็นที่ยาวที่สุด ซึ่งก็คือสีแดง ด้วยเหตุนี้เองจึงเรียกรังสีชนิดนี้ว่า รังสีอินฟราเรด หรือ รังสีใต้แดง

### 2.2.1 อินฟราเรดเซนเซอร์แบบพาสซีฟ (PIR Sensor)

คือ อุปกรณ์ Sensor ชนิดหนึ่งที่ใช้ตรวจจับคลื่นรังสี Infrared ที่แผ่มาจาก มนุษย์ หรือ สัตว์ที่มีการเคลื่อนไหว ทำให้มีการนำเอา PIR มาประยุกต์ใช้งานกันเป็นอย่างมากใช้เพื่อตรวจจับการเคลื่อนไหวของสิ่งมีชีวิต หรือ ตรวจจับการบุกรุกในงานรักษาความปลอดภัย



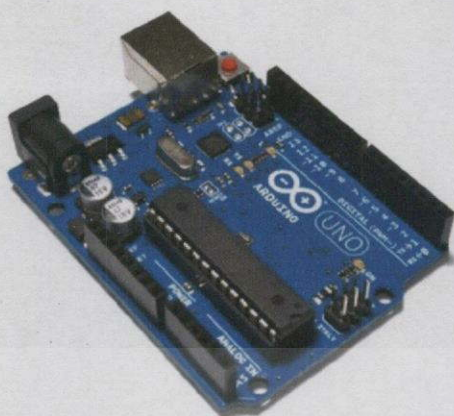
รูปที่ 2.2 อินฟราเรดเซนเซอร์แบบพาสซีฟ (PIR Sensor)

### 2.3 ไมโครคอนโทรลเลอร์ (Microcontroller)

ไมโครคอนโทรลเลอร์ คือ อุปกรณ์ที่มีหน่วยประมวลผลและความจำขนาดเล็กภายในตัวเอง สามารถรับ-ส่ง ข้อมูลได้ทั้งแบบดิจิทัลและอนาล็อก ใช้พลังงานน้อย ทำให้เป็นที่นิยมในการใช้งานในรูปแบบที่เรียกว่า Embedded เช่น เครื่องใช้ไฟฟ้าอัจฉริยะทั้งหลาย

#### 2.3.1 อาร์ดูอโน้ (Arduino Module)

Arduino คือ เครื่องมือที่จะทำให้คอมพิวเตอร์สามารถรับสัญญาณจากภายนอกและส่งสัญญาณไปควบคุมอุปกรณ์ภายนอกได้อย่างมีประสิทธิภาพ เป็นไมโครคอนโทรลเลอร์ชนิดหนึ่ง ซึ่งเป็นแบบที่เรียกว่า Open Hardware หรือ Open Source กล่าวคือ Arduino อุปกรณ์ที่มีแบบส่วนประกอบเป็นมาตรฐานที่เปิดเผยมุ่งหมายว่า สามารถทำเองโดยใช้แบบที่มีการเปิดเผยทั่วไป ข้อดีของการใช้ Arduino Module ก็คือซื้อหาได้ง่าย มีราคาถูก มีซอฟต์แวร์ให้ประยุกต์ใช้งานได้ อย่างหลากหลาย สามารถนำไปใช้งานทั่วไปหรือแบบธุรกิจได้โดยไม่ต้องเสียค่าลิขสิทธิ์ เป็นรูปแบบที่มีข้อมูลมากที่สุดบนอินเทอร์เน็ต การพัฒนาง่าย เพราะมีตัวอย่างมากมาย และไม่ต้องเขียนโปรแกรมในรูปแบบ Low Level คือ สามารถใช้คำสั่งเขียนโปรแกรมได้เสมือนโปรแกรมภาษาขั้นสูงทั่วไป เช่น ภาษาซี จาวา เป็นต้น โดยในปัจจุบัน Arduino module นั้นมีหลายรุ่นให้เลือกใช้งาน



รูปที่ 2.3 Arduino Uno R3

## 2.4 GSM Modem

GSM modem ใช้สำหรับการส่งข้อความในรูปแบบของ SMS ในการแจ้งเตือนการบุกรุก เมื่อใดก็ตามที่มีการตรวจจับวัตถุได้โดยเซนเซอร์อัลตราโซนิก GSM modem ที่ใช้ในโครงการนี้คือ Siemens TC35 GSM module



รูปที่ 2.4 GSM Module Siemens TC35

สำหรับ Siemens TC35 GSM Module นี้ใช้งานที่ความถี่ 850 MHz, 900 MHz, 1800 MHz และ 1900 MHz มีขนาดกะทัดรัด และง่ายต่อการใช้ ได้รับการออกแบบพร้อมวงจรเชื่อมต่อ 3V3/5V TTL ที่ให้เราต่อเข้ากับไมโครคอนโทรลเลอร์ 5V(PIC, Arduino, AVR และอื่นๆ) เช่นเดียวกับไมโครคอนโทรลเลอร์ 3V3(ARM, ARM, Cortex XX, และอื่นๆ) baud rate สามารถตั้งค่าได้จาก 9600-115200 ผ่านทางคำสั่ง AT เดิมที Modem นี้อยู่ในโหมด Autobaud ส่วน GSM TTL Modem นี้มี TCP/IP stack ภายในเพื่อให้เราต่อเข้ากับ internet ผ่านทาง GPRS ได้ ซึ่งเหมาะสมสำหรับ SMS เช่นเดียวกับการส่ง DATA ใน M2M interface นอกจากนี้ modem นี้ใช้สายไฟเพียงสองเส้นเท่านั้น(Tx,Rx) ยกเว้นแหล่งจ่ายไฟเพื่อต่อเข้ากับ microcontroller/Host พร้อมทั้ง Low Dropout Linear voltage regulator ที่มาด้วยทำให้เราต่อแหล่งจ่ายที่ไม่ได้รับการ regulated(4.2V-13V) ได้ในช่วงกว้าง

## 2.5 แหล่งจ่ายไฟตรง(DC Power Supply)

เครื่องจ่ายไฟตรง (DC Power Supply) เป็นอุปกรณ์ที่ใช้ในการจ่ายกำลังงานให้แก่วงจรอิเล็กทรอนิกส์ ซึ่งถือเป็นหัวใจสำคัญอย่างหนึ่ง เพราะในอุปกรณ์อิเล็กทรอนิกส์ต่างชนิดกันก็ต้องการแหล่งจ่ายแรงดันไฟฟ้ากระแสตรงแตกต่างกันด้วย และในปัจจุบันได้มีการนำเอาเทคโนโลยีสมัยใหม่มาปรับปรุงเปลี่ยนแปลงทำให้แหล่งจ่ายไฟกระแสตรงมีประสิทธิภาพดียิ่งขึ้น แต่โดยทั่วไปแล้วจะมีโครงสร้างพื้นฐานคล้าย ๆ กัน โดยแบ่งออกเป็น 2 ประเภทใหญ่ ๆ คือ

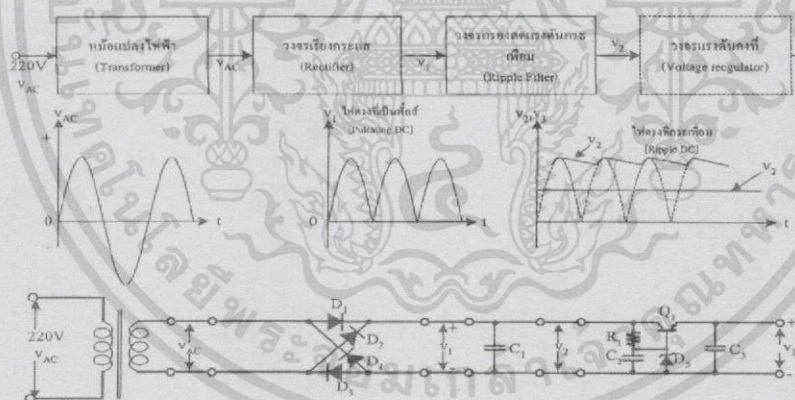
### 2.5.1. แหล่งจ่ายแรงดันไฟฟ้ากระแสตรงแบบเชิงเส้น (Linear Power Supply)

แหล่งจ่ายแรงดันไฟฟ้ากระแสตรงแบบเชิงเส้นเป็นแหล่งจ่ายไฟกระแสตรงที่นิยม ใช้มาตั้งแต่อดีต เพราะออกแบบได้ง่าย และรวดเร็วไม่ซับซ้อน แต่จะเกิดการสูญเสียกำลังไฟฟ้า ในรูปของความร้อน ทำให้ได้ประสิทธิภาพของแหล่งจ่ายไฟต่ำ

### 2.5.2. แหล่งจ่ายแรงดันไฟฟ้ากระแสตรงแบบสวิตซ์ซิ่ง (Switching Power Supply)

เป็นแหล่งจ่ายแรงดันไฟฟ้ากระแสตรงที่มีประสิทธิภาพสูง มีน้ำหนักเบา แหล่งจ่ายไฟฟ้ากระแสตรงแบบนี้มีแนวโน้มที่จะใช้มากในอนาคตเพราะมีขนาดเล็ก จึงทำให้อุปกรณ์อิเล็กทรอนิกส์ที่ใช้แหล่งจ่ายไฟฟ้าแบบสวิตซ์ซิ่งมีขนาดเล็กลงด้วย แต่มีข้อเสียตรงที่มีวงจรมีความยุ่งยากซับซ้อน เสี่ยงต่อการผิดพลาดได้ง่าย

ส่วนประกอบพื้นฐานของแหล่งจ่ายแรงดันไฟฟ้ากระแสตรง



รูปที่ 2.5 วงจรจ่ายไฟตรงจากไฟบ้าน

ความหมายและคำอธิบายส่วนต่างๆพื้นฐานของเครื่องจ่ายไฟตรงหลายทางออก

#### 1. หม้อแปลงไฟฟ้า (Transformer)

หม้อแปลงไฟฟ้าจะทำหน้าที่เปลี่ยนค่าแรงดันไฟฟ้ากระแสสลับให้เป็นแรงดันที่เหมาะสมกับการใช้งาน โดยสามารถเปลี่ยนได้ทั้งมากขึ้นและลดลง ซึ่งโดยปกติจะเป็นการลดแรงดันเพื่อใช้ในวงจรอิเล็กทรอนิกส์

#### 2. วงจรเรียงกระแส (Rectifier)

ทำหน้าที่เปลี่ยนไฟฟ้ากระแสสลับให้เป็นไฟฟ้ากระแสตรงโดยผ่านชุดของไดโอด ลักษณะของไฟตรงมีรูปร่างเป็นพัลส์ (Pulsating DC) ซึ่งจะมีการกระเพื่อม (Ripple Voltage) สูงมาก

#### 3. วงจรกรองสัญญาณ (Ripple Filter)

ทำหน้าที่กรองแรงดันกระเพื่อมให้มีการกระเพื่อมน้อยลงโดยวงจรอิเล็กทรอนิกส์จะสามารถทำงานได้ดีเมื่อมีค่าเล็กน้อยเท่านั้น ซึ่งเปอร์เซ็นต์การกระเพื่อมควรน้อยกว่า 10 % โดยสามารถสร้างขึ้นมาจากอุปกรณ์ที่เป็นรีแอคทีฟ (Reactive) ได้แก่ ตัวเก็บประจุ และ ตัวเหนี่ยวนำ ซึ่งจะต่อในลักษณะวงจรกรองความถี่ต่ำผ่าน (Low pass filter)

#### 4. วงจรรักษาแรงดันคงที่ (Voltage Regulator)

ใช้สำหรับวงจรอิเล็กทรอนิกส์ที่ต้องการความคงที่ของแรงดันไฟตรงที่เรียบมาก ๆ โดยจะทำหน้าที่กำจัดแรงดันกระเพื่อมที่ยังหลงเหลืออยู่เล็กน้อยให้หมดไป ไฟตรงที่ได้จะมีคุณภาพใกล้เคียงกับแบตเตอรี่ สามารถสร้างได้จากอุปกรณ์อิเล็กทรอนิกส์พื้นฐานหรือใช้วงจรรวม (IC)

#### 5. วงจรป้องกัน (Protection)

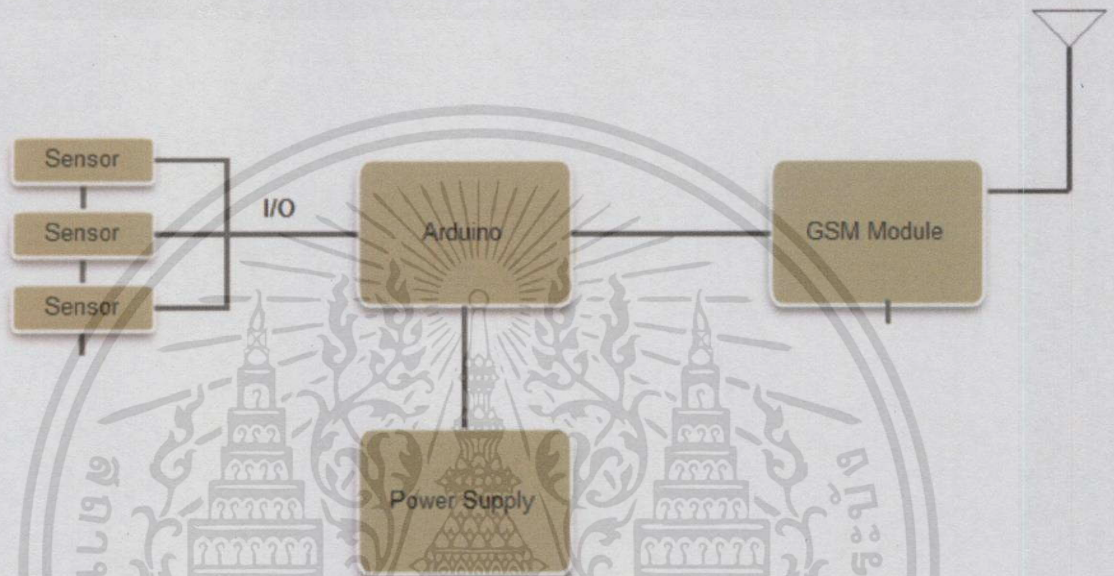
เป็นวงจรที่เพิ่มเติมขึ้นมาเพื่อป้องกันไม่ไห้แหล่งจ่ายไฟฟ้าเสียหายอันเนื่องมาจากการลัดวงจร (Short Circuit) หรือกระแสเกิน (Overload) ที่เอาท์พุท ที่อาจจะทำให้อุปกรณ์บางตัวเสียหายได้ จึงเป็นวงจรที่มีความสำคัญในแหล่งจ่ายไฟกระแสตรง

## บทที่ 3

### การออกแบบและการจัดทำโครงงาน

#### 3.1 การออกแบบ

##### 3.1.1 ส่วนประกอบของระบบ



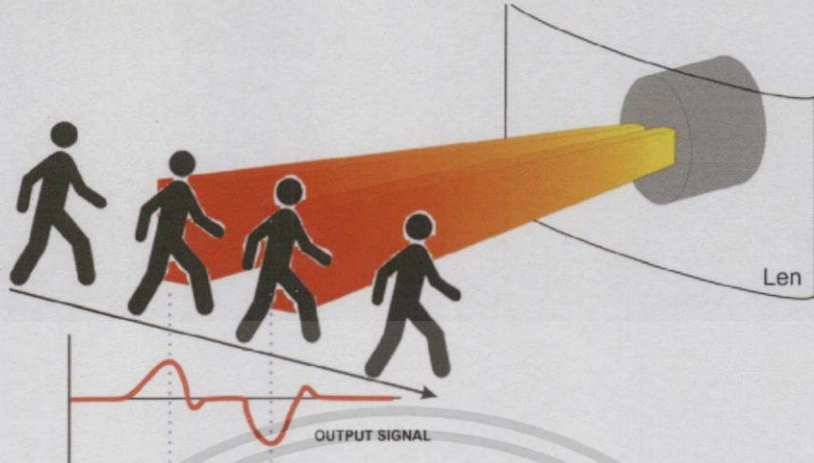
รูปที่ 3.1 ส่วนประกอบของระบบ

เซนเซอร์รับอินพุตในรูปของการเปลี่ยนไปของสัญญาณ จากนั้นจะทำการประมวลผลด้วย Arduino Module เพื่อ เปรียบเทียบสัญญาณที่เป็นดิจิตอล ( โลจิก 0 , 1 ) ว่าเมื่อค่าที่เป็นศูนย์จะไม่เกิดการแจ้งเตือนในรูปแบบต่างๆ แต่เมื่อชุดการทำงานของเซนเซอร์ให้ค่าที่เป็นหนึ่ง Arduino จะทำการประมวลผลส่งสัญญาณแจ้งเตือนโดยประมวลผลด้วยการโปรแกรมทางคอมพิวเตอร์ที่ต่อกับซีเรียลพอร์ท(Serial Port) เพื่อให้ GSM Module ทำการส่ง SMS แจ้งเตือนผู้ใช้งานว่าเกิดการบุกรุกขึ้นภายในบริเวณย่านการตรวจจับ

#### 3.2 การรับสัญญาณด้วยเซนเซอร์

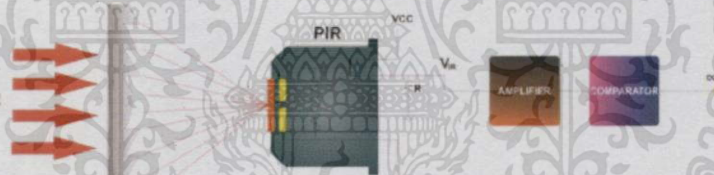
##### 3.2.1 อินฟราเรดเซนเซอร์แบบพาสซีฟ (PIR Sensor)

ภายใน PIR จะมีอุปกรณ์ตรวจจับรังสี Infrared อยู่ เมื่อมี คน หรือ สัตว์ ที่มีความอบอุ่นในร่างกายเคลื่อนที่ผ่านเข้ามาใน พื้นที่โซนที่ PIR สามารถตรวจจับคลื่นรังสี Infrared ที่แผ่ออกมาจากสิ่งมีชีวิตได้ PIR จะเปลี่ยนคลื่นรังสี Infrared ให้กลายเป็น กระแสไฟฟ้าดังรูป จะเห็นว่าเมื่อมีสิ่งมีชีวิตเคลื่อนที่ผ่าน อุปกรณ์ตรวจจับรังสี Infrared ตัวที่ 1 จะได้สัญญาณ Output ออกมาสูงกว่าแรงดันปรกติ และ เมื่อสิ่งมีชีวิตเคลื่อนที่ผ่าน อุปกรณ์ตรวจจับรังสี Infrared ตัวที่ 2 จะได้แรงดัน Output ต่ำกว่าค่าแรงดันปรกติ



รูปที่ 3.2 การตรวจจับของอินฟราเรดเซนเซอร์

ความร้อนวัดได้จากการเปลี่ยนแปลงระดับรังสีอินฟราเรดที่ปล่อยออกมาจากวัตถุ เมื่อวัตถุเคลื่อนที่ (สิ่งมีชีวิตทุกชนิด จะแผ่รังสีอินฟราเรดออกมาจากตัวเอง การแผ่รังสีดังกล่าวเกิดจากการเคลื่อนที่ของอิเล็กตรอนในอะตอม ปริมาณรังสีจะมีมากน้อยตามแต่โครงสร้างทางเคมี และอุณหภูมิของวัตถุหรือสิ่งมีชีวิตนั้นๆ) จึงทำให้สามารถตรวจจับสัญญาณลจิกที่เปลี่ยนแปลงที่ขาเอาต์พุตได้



รูปที่ 3.3 วงจรของ PIR Motion Sensor

สำหรับ PIR Motion Sensor เป็นโมดูลที่มีวงจรอยู่ภายในทำให้ผู้ใช้สามารถใช้งานได้สะดวก โดยการต่อใช้งานจะใช้สายสัญญาณเพียงแค่ 3 เส้น คือ

- DC Power สามารถใช้กับไฟเลี้ยงตั้งแต่ 5-12 V
- GND
- Alarm

คุณสมบัติของ PIR Motion Sensor

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## โหมดการใช้งานของสัญญาณทางเอาต์พุต

โหมดสัญญาณเอาต์พุตสามารถเลือกใช้งานได้ 2 แบบ คือ

1. สัญลักษณ์ H (HIGH) หมายถึง เอาต์พุตเป็นลอจิก 0 เมื่ออยู่ในสภาวะปกติ และเอาต์พุตเป็นลอจิก 1 เมื่อตรวจจับความเคลื่อนไหวได้
2. สัญลักษณ์ L (LOW) หมายถึง เอาต์พุตเป็นลอจิก 0 เมื่ออยู่ในสภาวะปกติ และเอาต์พุตเป็นลुकคลื่นลอจิก 1 สลับกับ 0 อย่างต่อเนื่อง (pulse) เมื่อตรวจจับความเคลื่อนไหวได้

ในโครงการนี้ได้เลือกใช้สัญญาณเอาต์พุตในโหมด H (แบบที่1)

## 3.3 การออกแบบการส่งสัญญาณแจ้งเตือน

### 3.3.1 รายละเอียดของภาษาที่ใช้ในการโปรแกรมคำสั่งและ Arduino UNO R3

โปรแกรมที่ใช้กับ Arduino คือ open source environment เขียนด้วย Java โปรแกรมนี้คล้ายกับภาษา C++ ที่มีคำสั่งที่มีโครงสร้างสถาปัตยกรรมเดียวกัน แต่มีฟังก์ชันเฉพาะในการโปรแกรม Arduino ตัวอย่างเช่น `pinMode(x,y)` เพื่อกำหนดขาสำหรับเอาต์พุต หรือ `Inputs, analog//digital write and read, pulseIn(x,y), delay(xxxx)` และ `serial.print()`.

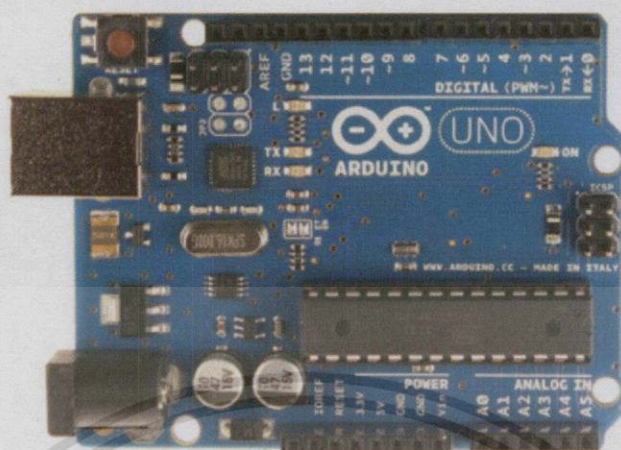
ระบบรักษาความปลอดภัยภายในบ้านนี้ใช้ Arduino Uno R3 เป็นตัวควบคุมส่วนกลาง ที่มีบทบาทหลักของหน่วยประมวลผลกลาง บอร์ด Arduino นี้ควบคุมขบวนการทั้งหมดที่เกี่ยวข้องกับเซนเซอร์ และ GSM modem ซึ่งถือเป็นส่วนที่สำคัญมากในโครงการนี้

Arduino Uno คือบอร์ดไมโครคอนโทรลเลอร์ที่มีพื้นฐานบน ATmega328 มีขา input/output ทางดิจิทัลทั้งหมด 14 ขา (ที่ขา 6 ใช้เป็นเอาต์พุต PWM) ขาอินพุตแบบอนาล็อก 6 ขา, resonator แบบเซรามิกความถี่ 16 MHz ขาเชื่อมต่อ USB และสายไฟเลี้ยง รวมถึง ICSP header พร้อมปุ่มรีเซ็ต ซึ่งมีทุกสิ่งที่ต้องใช้เพื่อรองรับไมโครคอนโทรลเลอร์ เพียงต่อเข้ากับคอมพิวเตอร์ด้วยสาย USB หรือสายไฟเลี้ยงด้วยตัวแปลง AC-to-DC

ATmega328P คือไมโครคอนโทรลเลอร์ CMOS กำลังต่ำขนาด 8 บิต มีพื้นฐานบนสถาปัตยกรรม AVR enhanced RISC โดยปฏิบัติตามคำสั่งในวัฏจักร single clock ATmega328P จะให้ความเร็วถึง 1 MIPS ต่อความถี่ MHz ทำให้ผู้ออกแบบระบบใช้การกินไฟต่อความเร็วในการประมวลผลต่ำสุด Arduino นี้ผลิตโดยใช้เทคโนโลยีหน่วยความจำชนิด non-volatile ความถี่สูงของ Atmel ส่วนของ ISP Flash บนชิปให้หน่วยความจำโปรแกรมสามารถนำมาโปรแกรมได้อีก หรือ In-System ผ่านทางการเชื่อมต่อ SPI แบบอนุกรม โดยผู้ออกแบบโปรแกรมแบบ non-volatile ทั่วไป หรือโดยโปรแกรม On-chip Boot บน AVR core. ส่วนของ Boot program นี้สามารถใช้ในการเชื่อมต่อใดๆ เพื่อลงโปรแกรมใช้งานใน Application Flash memory ได้ โปรแกรมในส่วนของ Boot Flash จะทำงานต่อเมื่อขณะที่มีการปรับปรุงส่วนของ Application Flash ให้การทำงานสำหรับ Read-While-Write ที่แท้จริง

โดยรวม CPU ชนิด RISC ขนาด 8 bit เข้ากับ In-System Self-Programmable Flash บน monolithic chip, Atmel ATmega48PA/88PA/168PA/328P จะกลายเป็นไมโครคอนโทรลเลอร์ศักยภาพสูงที่ให้คำตอบทางด้านประสิทธิภาพของราคาและมีความยืดหยุ่นสูง ให้กับงานควบคุมแบบ embedded จำนวนมาก ATmega328P AVR นั้นรองรับโดยโปรแกรมการพัฒนาโปรแกรมและระบบ รวมถึง C Compilers, Macro Assemblers, Program Debugger/Simulators, In-Circuit Emulators และ Evaluation kits.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 บอร์ด Arduino Uno R3

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

รูปที่ 3.5 คุณสมบัติทั่วไปของ ATmega328

### 3.3.2 รายละเอียดของส่วนต่างๆ

#### 1.) กำลัง

จ่ายไฟเลี้ยงให้แก่ Arduino Uno ผ่านสาย USB หรือจากแหล่งจ่ายไฟภายนอก แหล่งจ่ายไฟจะถูกเลือกโดยอัตโนมัติ บอร์ดนี้สามารถทำงานโดยอาศัยแหล่งจ่ายภายนอกขนาด 6 ถึง 20 โวลต์ได้ อย่างไรก็ตามหากจ่ายไฟต่ำกว่า 7 โวลต์ ขา 5 โวลต์อาจจะจ่ายไฟให้น้อยกว่า 5 โวลต์และบอร์ดนี้จะไม่เสถียร หากใช้ไฟสูงกว่า 12 โวลต์ ตัว regulator แรงดันอาจจะร้อนเกิน และทำให้บอร์ดเสียหายได้ ช่วงระดับแรงดันแนะนำคือ 7-12 โวลต์ ซึ่งขากำลังไฟมีดังต่อไปนี้

- Vin. แรงดันอินพุตสำหรับบอร์ด Arduino เมื่อใช้แหล่งจ่ายไฟจากภายนอก (ตรงข้ามกับ 5 โวลต์ จากสาย USB หรือแหล่งจ่ายไฟที่ได้รับการ regulated อื่นๆ) เราสามารถจ่ายแรงดันผ่านขา Vin หรือหากจ่ายแรงดันผ่านสายกำลัง ก็จะใช้ขานี้
- 5V. ขาเอาต์พุตนี้ ใช้ไฟ 5 โวลต์ที่ได้รับการ regulate จาก regulator บนบอร์ด สามารถจ่ายไฟให้แก่บอร์ดนี้โดยไฟจาก DC power jack(7-12 โวลต์) ปลาย USB(5 V) หรือขา VIN ของบอร์ด(7-12 โวลต์) จ่ายไฟผ่านทางขา 5 โวลต์ หรือ 3.3 โวลต์ ที่ bypasses regulator นี้ ที่สามารถทำให้บอร์ดเสียหายได้ จึงไม่แนะนำในส่วนนี้
- 3V3. คือแหล่งจ่ายไฟ 3.3 โวลต์ที่สร้างจาก regulator บนบอร์ด ดึงกระแสสูงสุด 50 mA
- GND ขากราวด์
- IOREF ขาบนบอร์ด Arduino ให้แรงดันอ้างอิงที่ซึ่งไมโครคอนโทรลเลอร์ทำงาน การตั้งค่าที่ถูกต้องสามารถอ่านแรงดันขา IOREF และเลือกแหล่งจ่ายไฟที่เหมาะสม หรือทำให้ทรานซิสเตอร์แรงดันทางด้านเอาต์พุตให้ใช้งานกับแรงดัน 5 โวลต์ หรือ 3.3 โวลต์ ได้

## 2.) หน่วยความจำ

ATmega 328 มี 32 kB(ใช้ 0.5 kB สำหรับ boot loader) นอกจากนี้ยังมี SRAM ขนาด 2kB และ EEPROM ขนาด 1kB(ที่สามารถอ่านและเขียนด้วย EEPROM library)

## 3.) อินพุตและเอาต์พุต

TWI : ขา A4 หรือขา SDA และ A5 สำหรับขาดิจิตอล 14 ขาแต่ละขา บน Uno สามารถใช้เป็นอินพุต หรือเอาต์พุต โดยใช้ pinMode(), digitalWrite(), และ digitalRead() ทำงานที่ 5 โวลต์ ขาแต่ละขาให้หรือรับกระแสสูงสุด 40 mA และมีตัวต้านทาน pull-up ภายใน(ไม่ต่อโดย default) ขนาด 20-50 โอห์ม นอกจากนี้ขาบางขายังมีหน้าที่พิเศษด้วย

Serial : 0(RX) และ 1(TX) ใช้เพื่อรับ(RX) และส่ง(TX) ข้อมูล TTL แบบอนุกรม ขาเหล่านี้ต่อเข้ากับขาที่สอดคล้องกันของ ATmega8U2 USB-to-TTL Serial chip

External Interrupts : 2 และ 3 สามารถตั้งค่าขาเหล่านี้เพื่อกระตุ้นขา interrupt โดยใช้สภาวะ "low" ทั้งขอบขึ้นและขอบลง หรือเมื่อมีการเปลี่ยนค่า

PWM : 3,5,6,9,10 และ 11 ให้เอาต์พุต PWM ขนาด 8 bit พร้อมการทำงาน analogWrite()

SPI : 10(SS), 11(MOSI), 12(MISO), 13(SCK) ขาเหล่านี้รองรับการสื่อสารแบบ SPI โดยใช้ SPI library

LED : 13 มี LED ภายในต่อเข้ากับขา 13 ทางดิจิตอล เมื่อขานี้มีสภาวะเป็น "HIGH" LED นี้จะติด ขณะที่หากขานี้มีสภาวะเป็น "LOW" LED จะดับ

สำหรับ Arduino Uno R3 มีอินพุตทางอนาล็อกด้วยกัน 6 อินพุต แสดงด้วย A0 ถึง A5 ที่แต่ละขาจะให้ความละเอียด 10 bits (เช่น ค่าที่แตกต่างกัน 1024 ค่า) โดย default จะวัดระดับแรงดัน 5 โวลต์จากกราวด์ แม้ว่าเป็นไปได้ที่จะเปลี่ยนระดับแรงดันนี้โดยใช้ขา AREF และการทำงาน

analogReference() นอกจากนี้ ขาบางขายังทำหน้าที่พิเศษอีกด้วย หรือขา SCL รองรับการสื่อสารแบบ TWI โดยใช้ Wire library นอกจากนี้ยังมีขาอื่นๆอีก บนบอร์ด ดังนี้

- AREF แรงดันอ้างอิงสำหรับอินพุตแบบอนาล็อก ใช้กับ analogReference()
- Reset นำสายนี้ให้มีสถานะเป็น “LOW” เพื่อรีเซ็ตไมโครคอนโทรลเลอร์ ใช้กันทั่วไปเพื่อเพิ่มปุ่มรีเซ็ตเข้ากับ shields ที่ป้องกันการรบกวนบนบอร์ด

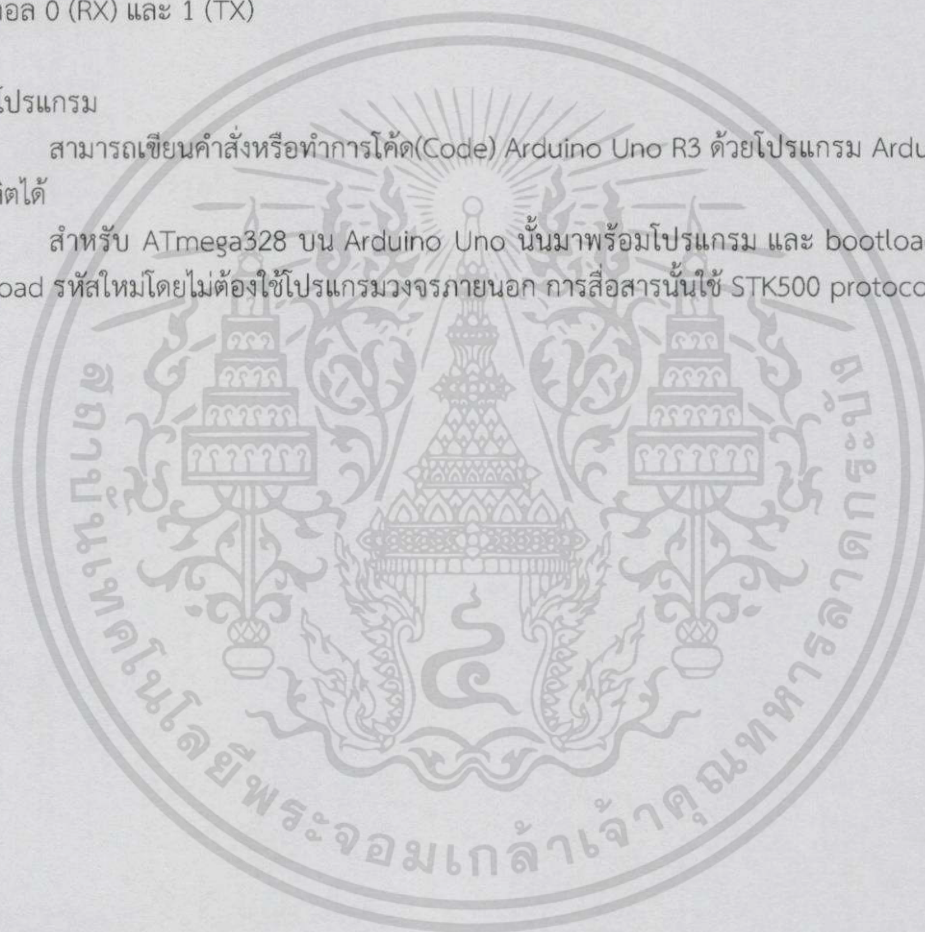
#### 4.) การสื่อสาร

Arduino Uno มีอุปกรณ์จำนวนมากสำหรับสื่อสารกับคอมพิวเตอร์ Arduino อื่นๆ หรือไมโครคอนโทรลเลอร์อื่นๆ ATmega328 ให้การสื่อสารแบบอนุกรม UART TTL(5V) ที่มีอยู่บนขา ดิจิตอล 0 (RX) และ 1 (TX)

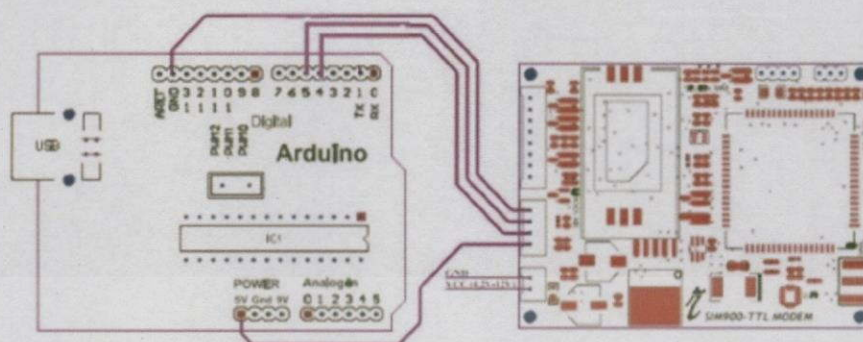
#### 5.) โปรแกรม

สามารถเขียนคำสั่งหรือทำการโค้ด(Code) Arduino Uno R3 ด้วยโปรแกรม Arduino จากผู้ผลิตได้

สำหรับ ATmega328 บน Arduino Uno นั้นมาพร้อมโปรแกรม และ bootloader ที่ให้ upload รหัสใหม่โดยไม่ต้องใช้โปรแกรมวงจรรภายนอก การสื่อสารนั้นใช้ STK500 protocol ดังเดิม

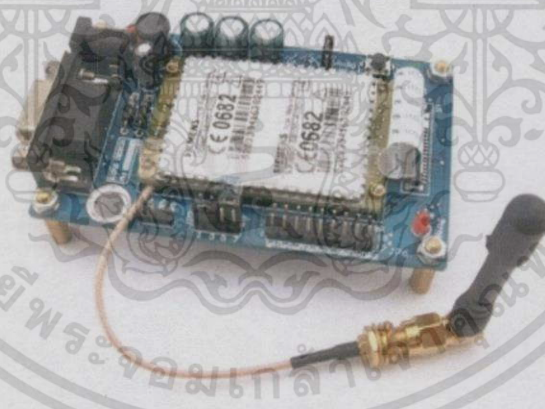


### 3.3.3 GSM Modem



รูปที่ 3.6 การต่อบอร์ด Arduino เข้ากับ GSM modem

ขา Rx ของ GSM Module Siemens TC35 ต่อเข้ากับขาที่ 4 ของ Arduino และขา Tx ของ GSM Module Siemens TC35 ต่อเข้ากับขาที่ 5 ของ Arduino ต่อ Vinterface เข้ากับขา 5V บนบอร์ด Arduino และ GND เข้ากับขา GND บน Arduino ส่วนของ GSM modem นี้ต้องใช้ไฟเลี้ยงจากภายนอก เนื่องจากไฟเลี้ยงจากบอร์ด Arduino นั้นไม่พอ จึงต่อแบตเตอรี่ขนาด 9V เพื่อจ่ายไฟเลี้ยงให้แก่ modem นี้ ส่วนขา Vin และ GND ต่อเข้ากับแบตเตอรี่



รูปที่ 3.7 รูปของ GSM Module Siemens TC35

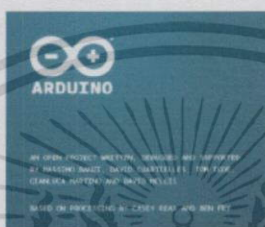
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 4

### ผลการทดลอง

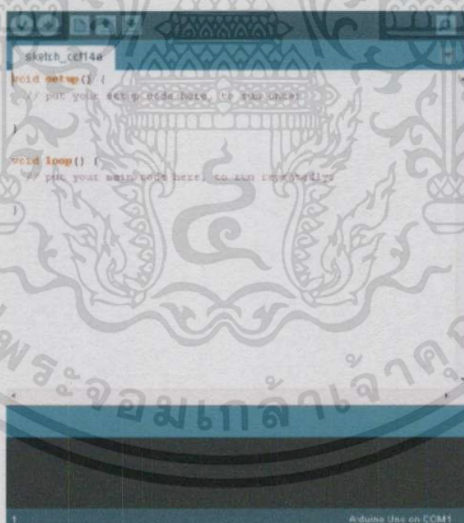
#### 4.1 การทดสอบบอร์ด Arduino Uno R3 ก่อนการใช้งาน

เราสามารถตรวจสอบได้ว่าบอร์ด Arduino Uno R3 ของเราสมบูรณ์หรือไม่ ขาเอาต์พุตมีปัญหาหรือไม่ได้จากไลบรารีพื้นฐานที่มาในตัวโปรแกรม Arduino Version 1.5.2 ได้โดยมีขั้นตอนดังนี้



รูปที่ 4.1 โปรแกรม Arduino Version 1.5.2

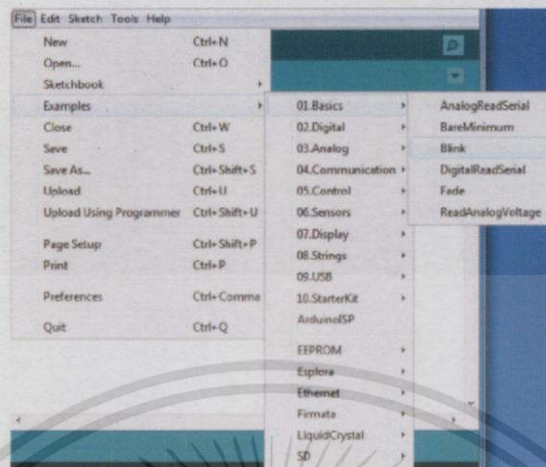
1.เปิดโปรแกรม Arduino Version 1.5.2 ขึ้นมา จะได้ดังรูปที่ 4.2



รูปที่ 4.2 หน้าเริ่มต้นของโปรแกรม Arduino

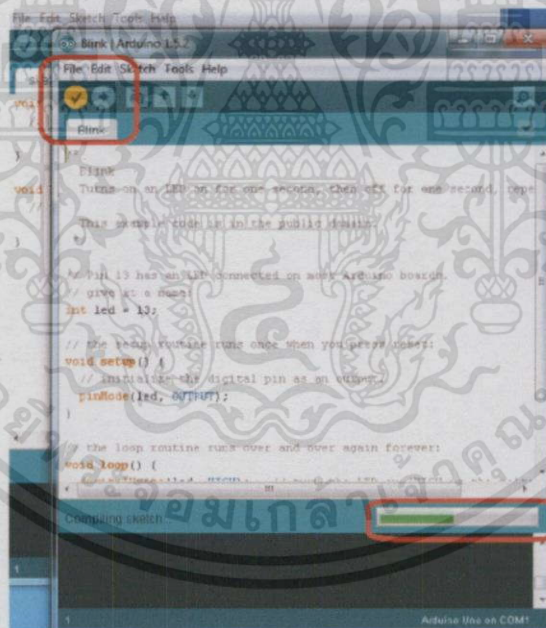
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. เข้าเมนู File เลือกที่ Examples ไปที่ 01.Basics แล้วเลือก Blink ดังแสดงในรูปที่ 4.3



รูปที่ 4.3 แสดงการเข้าสู่คำสั่ง Blink

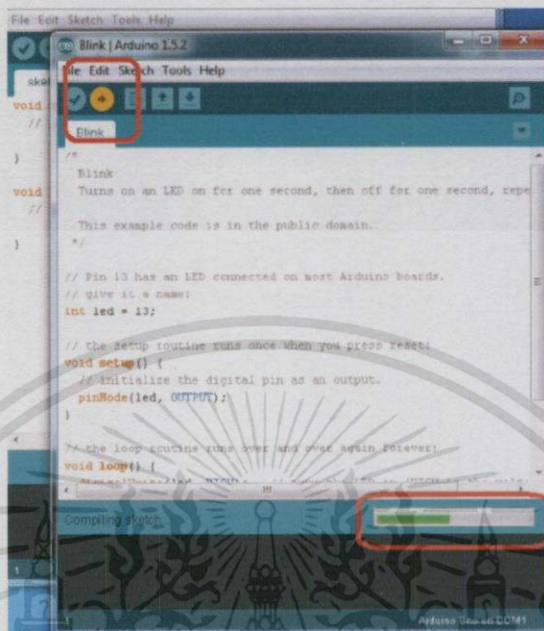
3. เมื่ออยู่ในหน้าต่างของคำสั่ง Blink แล้ว ให้ทำการ Verify ดังรูปที่ 4.4 เพื่อตรวจสอบความถูกต้องของโค้ดซึ่งโค้ดนี้มีความถูกต้องอยู่แล้วเนื่องจากมีอยู่ในไลบรารีที่ตัวโปรแกรมให้มา



รูปที่ 4.4 ตรวจสอบความถูกต้องของคำสั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ทำการอัปโหลดโปรแกรม (ดังรูปที่ 4.5) ลงบอร์ด ขั้นตอนนี้เป็นเสมือนการบอกให้ Arduino board รู้ว่าผู้กำหนดจะกำหนดให้บอร์ดทำงานอย่างไร

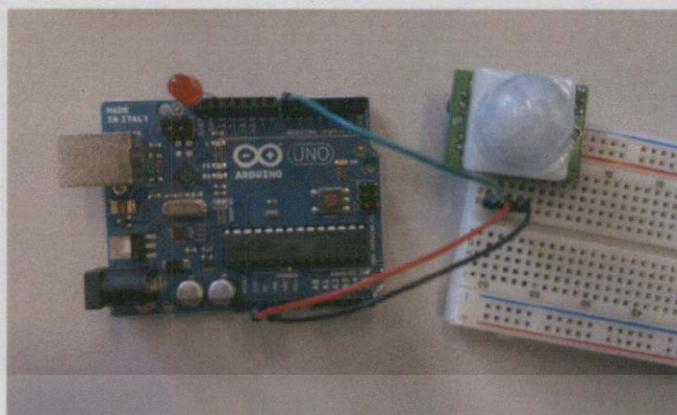


รูปที่ 4.5 การอัปโหลดโปรแกรมลงบอร์ด

5. ต่อหลอดไฟ LED เข้ากับ Arduino board พอร์ตที่ 13 ซึ่งเป็น Output Port กับพอร์ต GND สังเกตที่หลอดไฟ LED ว่ามีการกระพริบติด - ดับเป็นจังหวะหรือไม่ ถ้าไม่ติดก็แสดงว่าตัว Arduino board หรือตัวหลอดไฟ LED มีปัญหา ซึ่งวิธีนี้เป็นการที่ง่ายต่อตรวจสอบ อีกทั้งยังเป็นการสร้างความคุ้นเคยกับการใช้งานทั้งในส่วนของฮาร์ดแวร์และซอฟต์แวร์ให้กับผู้ใช้งาน

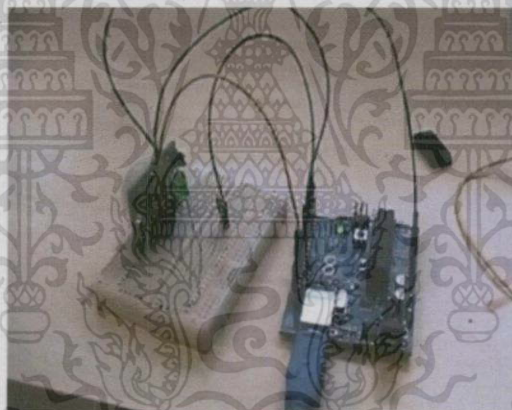
#### 4.2 ทดลองการรับสัญญาณด้วยเซนเซอร์

หลังจากที่ได้ออกแบบการตรวจจับเป็นที่เรียบร้อยแล้ว โดยได้แสดงดังรูปที่ 3.1 ได้ทำการทดลองโดยการต่ออินฟราเรดเซนเซอร์แบบพาสซีฟเข้ากับวงจรที่ประกอบไปด้วย Arduino Uno R3 และตัวต้านทาน โดยที่ Arduino Module นั้นจะต่อเข้ากับคอมพิวเตอร์ผ่าน Serial Port ซึ่งไฟเลี้ยงที่เลี้ยงวงจรนั้นจะมาจากเครื่องคอมพิวเตอร์ ในส่วนของการแสดงผลแจ้งเตือนหรือส่วนแสดง Output นั้นจะมาจากหลอดไฟ LED ที่ทำการต่อเข้ากับพอร์ตที่ 13 กับ GND บนบอร์ดของ Arduino Uno R3



รูปที่ 4.6 วงจรที่ใช้ในการทดลอง

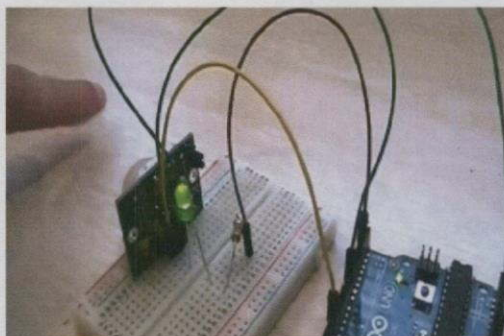
จากการทดลองเราจะพบว่า ในสภาวะปกติที่ไม่มีการเคลื่อนไหวใดๆ หลอดไฟLED จะยังคงดับอยู่ซึ่งแสดงดังรูปที่ 4.7 แต่เมื่อผู้ทดลองทำการเคลื่อนไหววัตถุหรือร่างกายซึ่งอยู่ในย่านของการตรวจจับ ก็จะทำให้หลอดไฟLED ติดในช่วงหนึ่ง และดับลง จนกว่าจะมีการเคลื่อนไหวอีกอันเป็นผลของการปรับ Delay Time ที่อยู่บนตัวเซนเซอร์



รูปที่ 4.7 ขณะที่ยังไม่เกิดการเคลื่อนที่ของวัตถุ

ซึ่งเราสามารถสรุปผลที่เกิดจากการทดลองครั้งนี้ได้ว่า ในสภาวะปกติ เอาต์พุตเป็นลอจิกจะเป็น 0 และให้เอาต์พุตเป็นลอจิก 1 เมื่อตรวจจับความเคลื่อนไหวได้อันเป็นผลมาจากการทำงานของสัญญาณเอาต์พุตในโหมดH

135106



รูปที่ 4.8 หลอดไฟLED สว่างเมื่อวัตถุเคลื่อนไหวในย่านการตรวจจับ

### 4.3 ระยะเวลาการทำงานของ PIR Sensor

ตาราง 4.1 ผลการทดลองย่านการทำงานของ PIR Sensor

ระยะทาง (เมตร)	LED ครั้งที่ 1	LED ครั้งที่ 2
0.5	ติด	ติด
1	ติด	ติด
1.5	ติด	ติด
2	ติด	ไม่ติด
2.5	ไม่ติด	ติด
3	ติด	ติด
5	ไม่ติด	ไม่ติด

### 4.4 ส่งสัญญาณแจ้งเตือนในรูปแบบของ SMS



รูปที่ 4.9 เครื่องตรวจจับขโมยนี้ทำการส่ง SMS เพื่อแจ้งเตือนการบุกรุก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.5 การแจ้งเตือนไปยังเลขหมายอื่น



รูปที่ 4.10 แสดงการแจ้งเตือนไปยังเลขหมายอื่น

ในการทดลองที่ 4.4 และ 4.5 เมื่อมีการบุกรุกในย่านตรวจจับของ PIR Sensor ตัวอุปกรณ์นี้จะทำการแจ้งเตือนในรูปแบบของ SMS และเราสามารถกำหนดข้อความหรือเลขหมายที่จะแจ้งเตือนได้ด้วยโปรแกรมที่ควบคุมการทำงานของ Arduino

ในการทดลองที่ 4.4 เราจะทำการส่งการแจ้งเตือนไปที่หมายเลข 089-6987775 ด้วยข้อความแจ้งเตือน “my Project is complete” จากรูปที่ 4.9 เราจะเห็นว่าโทรศัพท์ดังกล่าว ก็ได้รับการแจ้งเตือนจากหมายเลข 092-7599585 ซึ่งเป็นหมายเลขที่ใช้กับ GSM Module นั้นเอง

ส่วนรูปที่ 4.10 นั้นเราจะเห็นว่าโทรศัพท์ที่ใช้ในการส่ง ก็ได้ทำการแจ้งเตือนการบุกรุกด้วยข้อความ “SOS BoyZZZ” มาที่เบอร์ที่เราเปลี่ยนเลขหมายใหม่คือ 089-6845915 ได้อีกด้วย

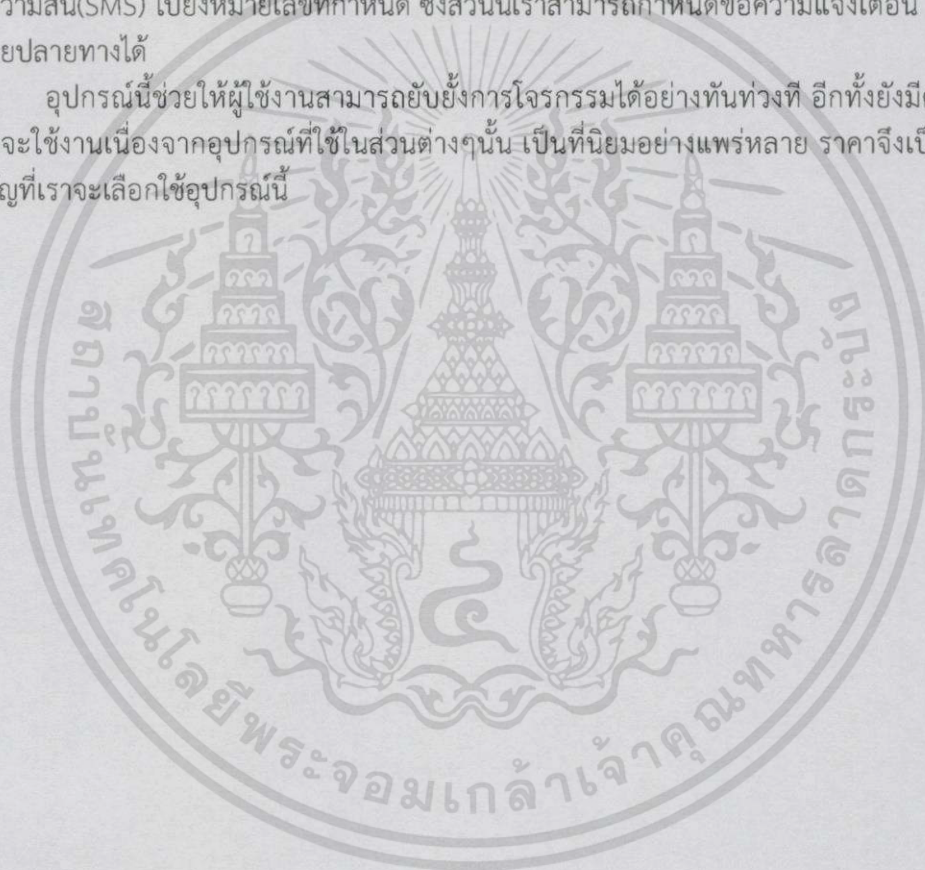
## บทที่ 5

### บทสรุป

เซนเซอร์ตรวจจับโมยแบบไร้สายเป็นอุปกรณ์รักษาความปลอดภัยภายในที่พักอาศัย ช่วยในการเฝ้าระวังการโจรกรรมในสถานที่นั้นๆ ยามปลอดภัย โดยอุปกรณ์นี้อาจจะถูกติดตั้งไว้ที่ประตูหรือหน้าต่าง เมื่อมีผู้บุกรุกผ่านเข้ามาในย่านการตรวจจับของอินฟราเรดเซนเซอร์ ก็จะทำให้เซนเซอร์ส่งสัญญาณเป็นการเปลี่ยนแปลงลอจิก 0,1

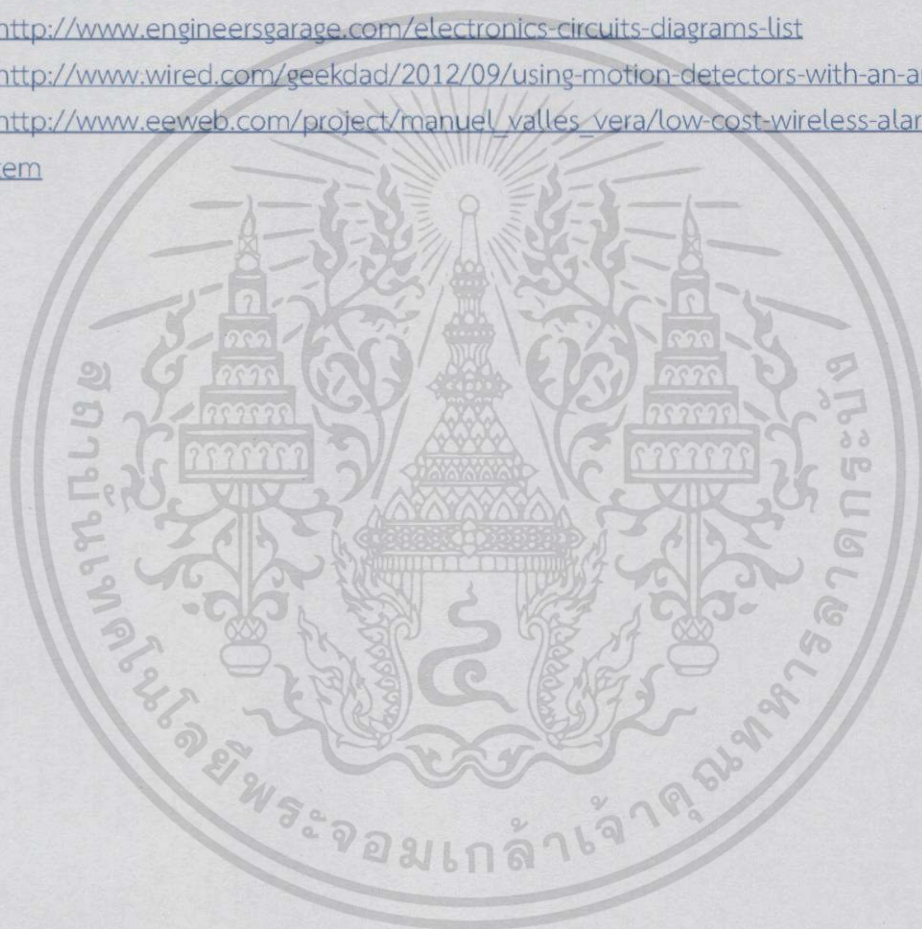
ส่วนการควบคุมหลักที่จะคอยประมวลผลสัญญาณและสั่งงานนั้นจะอยู่ที่ Arduino Uno R3 ที่จะประมวลผลลอจิกจากเซนเซอร์แล้วทำการควบคุมให้ GSM Module แจ้งเตือนด้วยการส่งข้อความสั้น(SMS) ไปยังหมายเลขที่กำหนด ซึ่งส่วนนี้เราสามารถกำหนดข้อความแจ้งเตือน และเลขหมายปลายทางได้

อุปกรณ์นี้ช่วยให้ผู้ใช้งานสามารถยับยั้งการโจรกรรมได้อย่างทันท่วงที อีกทั้งยังมีความคุ้มค่าที่จะใช้งานเนื่องจากอุปกรณ์ที่ใช้ในส่วนต่าง ๆ นั้น เป็นที่นิยมอย่างแพร่หลาย ราคาจึงเป็นปัจจัยสำคัญที่เราจะเลือกใช้อุปกรณ์นี้



## บรรณานุกรม

- [1] ZP. Horowitz, W. Hill: "The Art of Electronics", 2nd Edition, Cambridge University Press 1989, USA.
- [2] R.C. Jaeger: "Microelectronic Circuit Design", International Edition, The McGraw-Hill Companies, Inc. NY.
- [3] [https://docs.google.com/viewer?a=v&pid=sites&srcid=ZGVmYXVsdGRvbWFpbnxlbmduODQ0YXx\\_hcm1zeXN0ZW18Z3g6NmI4N2M2Mjg1NTOyZGMzYw](https://docs.google.com/viewer?a=v&pid=sites&srcid=ZGVmYXVsdGRvbWFpbnxlbmduODQ0YXx_hcm1zeXN0ZW18Z3g6NmI4N2M2Mjg1NTOyZGMzYw)
- [4] <http://people.ece.cornell.edu/land/courses/eceprojectsland/STUDENTPROJ/>
- [5] <http://www.engineersgarage.com/electronics-circuits-diagrams-list>
- [6] <http://www.wired.com/geekdad/2012/09/using-motion-detectors-with-an-arduino/>
- [7] [http://www.eeweb.com/project/manuel\\_valles\\_vera/low-cost-wireless-alarm-system](http://www.eeweb.com/project/manuel_valles_vera/low-cost-wireless-alarm-system)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ภาคผนวก



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.

ส่วนของคำสั่งที่ใช้ในการควบคุมอุปกรณ์ Arduino Uno R3



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Arduino 1.5.2

```
const int sensorPin = 2;
const int switchPin = 3;
const int ledPin = 13;
const int ledBlinkTime = 500;
const unsigned int calibrationTime = 60000;

void setup() {
  Serial.begin(9600);

  pinMode(sensorPin, INPUT);
  pinMode(switchPin, INPUT);
  pinMode(ledPin, OUTPUT);

  for (unsigned int i=0; i<calibrationTime; i+=ledBlinkTime*2) {
    digitalWrite(ledPin, HIGH);
    delay(ledBlinkTime);
    digitalWrite(ledPin, LOW);
    delay(ledBlinkTime);
  }

  if (digitalRead(sensorPin) == HIGH && digitalRead(switchPin) == HIGH)
  {
    digitalWrite(ledPin, HIGH);
    delay(5000);
    Serial.print("\r");
    delay(1000);
    Serial.print("AT+CMGF=1\r");
    delay(1000);
    Serial.print("AT+CMGS=\"+66896845915\"\r");
    delay(1000);
    Serial.print("SOS BoyZZZ\r");
    delay(1000);
    Serial.write(0x1A);
    delay(1000);
    myfunction();
    delay(1000);
  }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}  
}  
void myfunction (void) {  
  digitalWrite(ledPin, LOW);  
}  
void loop()  
{  
}
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข.

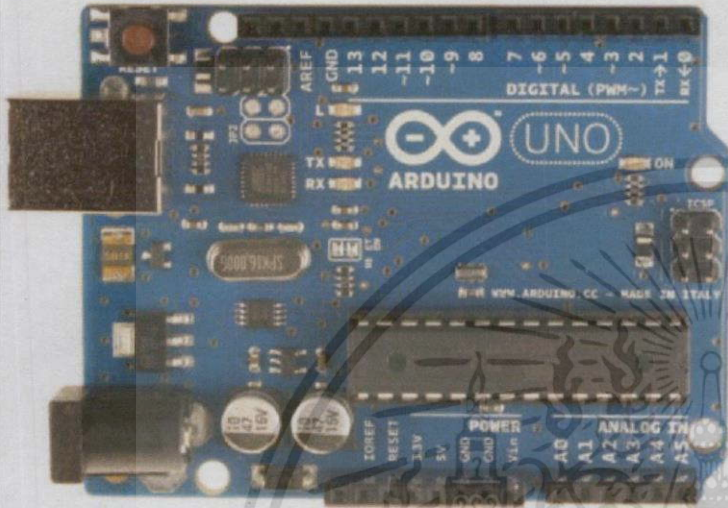
ข้อมูลของไมโครคอนโทรลเลอร์ ATmega 328



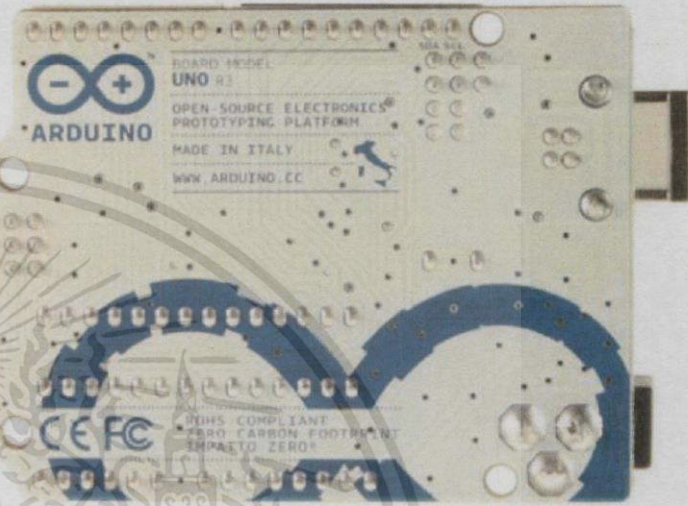
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



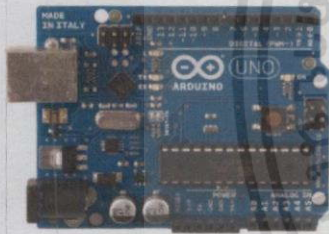
## Arduino Uno



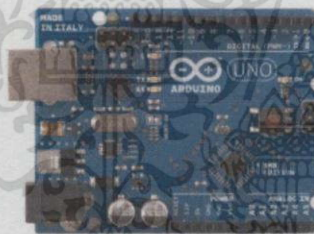
Arduino Uno R3 Front



Arduino Uno R3 Back



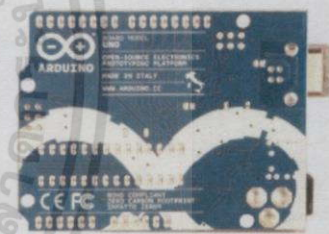
Arduino Uno R2 Front



Arduino Uno SMD



Arduino Uno Front



Arduino Uno Back

### Overview

The Arduino Uno is a microcontroller board based on the ATmega328 ([datasheet](#)). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.

Revision 2 of the Uno board has a resistor pulling the 8U2 HWB line to ground, making it easier to put into DFU mode.

Revision 3 of the board has the following new features:

- 1.0 pinout: added SDA and SCL pins that are near to the AREF pin and two other new pins placed near to the RESET pin, the IOREF that allow the shields to adapt to the voltage provided from the board. In future, shields will be compatible both with the board that use the AVR, which operate with 5V and with the Arduino Due that operate with 3.3V. The second one is a not connected pin, that is reserved for future purposes.
- Stronger RESET circuit.
- Atmega 16U2 replace the 8U2.

"Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions, see the [index of Arduino boards](#).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Summary

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

## Schematic & Reference Design

EAGLE files: [arduino-uno-Rev3-reference-design.zip](#)

Schematic: [arduino-uno-Rev3-schematic.pdf](#)

**Note:** The Arduino reference design can use an Atmega8, 168, or 328. Current models use an ATmega328, but an Atmega8 is shown in the schematic for reference. The pin configuration is identical on all three processors.

## Power

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

## Memory

The ATmega328 has 32 KB (with 0.5 KB used for the bootloader). It also has 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the [EEPROM library](#)).

## Input and Output

Each of the 14 digital pins on the Uno can be used as an input or output, using [pinMode\(\)](#), [digitalWrite\(\)](#), and [digitalRead\(\)](#) functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **External Interrupts: 2 and 3.** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the [attachInterrupt\(\)](#) function for details.
- **PWM: 3, 5, 6, 9, 10, and 11.** Provide 8-bit PWM output with the [analogWrite\(\)](#) function.
- **SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).** These pins support SPI communication using the [SPI library](#).
- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though it is possible to change the upper end of their range using the AREF pin and the [analogReference\(\)](#) function. Additionally, some pins have specialized functionality:

- **TWI: A4 or SDA pin and A5 or SCL pin.** Support TWI communication using the [Wire library](#).

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with [analogReference\(\)](#).
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

See also the [mapping between Arduino pins and ATmega328 ports](#). The mapping for the ATmega8, 168, and 328 is identical.

## Communication

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, [on Windows, a .inf file is required](#). The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A [SoftwareSerial library](#) allows for serial communication on any of the Uno's digital pins.

The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino software includes a [Wire library](#) to simplify use of the I2C bus; see the [documentation](#) for details. For SPI communication, use the [SPI library](#).

## Programming

The Arduino Uno can be programmed with the Arduino software ([download](#)). Select "Arduino Uno" from the **Tools > Board** menu (according to the microcontroller on your board). For details, see the [reference](#) and [tutorials](#).

The ATmega328 on the Arduino Uno comes preburned with a [bootloader](#) that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol ([reference](#), [C header files](#)).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see [these instructions](#) for details.

The ATmega16U2 (or 8U2 in the rev1 and rev2 boards) firmware source code is available. The ATmega16U2/8U2 is loaded with a DFU bootloader, which can be activated by:

- On Rev1 boards: connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2.
- On Rev2 or later boards: there is a resistor that pulling the 8U2/16U2 HWB line to ground, making it easier to put into DFU mode.

You can then use [Atmel's FLIP software](#) (Windows) or the [DFU programmer](#) (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader). See [this user-contributed tutorial](#) for more information.

## Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Uno is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2/16U2 is connected to the reset line of the ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

be well-coordinated with the start of the upload.

This setup has other implications. When the Uno is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

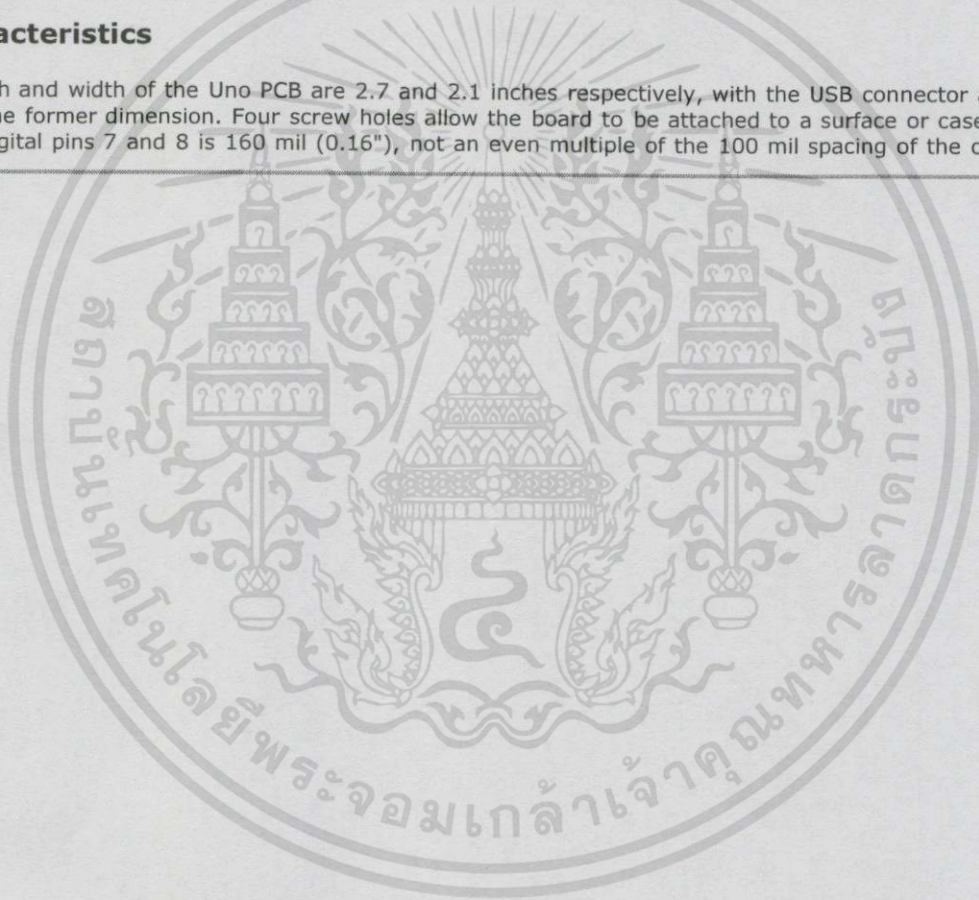
The Uno contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see [this forum thread](#) for details.

## USB Overcurrent Protection

The Arduino Uno has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

## Physical Characteristics

The maximum length and width of the Uno PCB are 2.7 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Four screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Features

- High Performance, Low Power AVR<sup>®</sup> 8-Bit Microcontroller
- Advanced RISC Architecture
  - 131 Powerful Instructions – Most Single Clock Cycle Execution
  - 32 x 8 General Purpose Working Registers
  - Fully Static Operation
  - Up to 20 MIPS Throughput at 20 MHz
  - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory Segments
  - 4/8/16/32K Bytes of In-System Self-Programmable Flash program memory (ATmega48PA/88PA/168PA/328P)
  - 256/512/512/1K Bytes EEPROM (ATmega48PA/88PA/168PA/328P)
  - 512/1K/1K/2K Bytes Internal SRAM (ATmega48PA/88PA/168PA/328P)
  - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
  - Data retention: 20 years at 85°C/100 years at 25°C<sup>(1)</sup>
  - Optional Boot Code Section with Independent Lock Bits
    - In-System Programming by On-chip Boot Program
    - True Read-While-Write Operation
  - Programming Lock for Software Security
- Peripheral Features
  - Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode
  - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
  - Real Time Counter with Separate Oscillator
  - Six PWM Channels
  - 8-channel 10-bit ADC in TQFP and QFN/MLF package
    - Temperature Measurement
  - 6-channel 10-bit ADC in PDIP Package
    - Temperature Measurement
  - Programmable Serial USART
  - Master/Slave SPI Serial Interface
  - Byte-oriented 2-wire Serial Interface (Philips I<sup>2</sup>C compatible)
  - Programmable Watchdog Timer with Separate On-chip Oscillator
  - On-chip Analog Comparator
  - Interrupt and Wake-up on Pin Change
- Special Microcontroller Features
  - Power-on Reset and Programmable Brown-out Detection
  - Internal Calibrated Oscillator
  - External and Internal Interrupt Sources
  - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby
- I/O and Packages
  - 23 Programmable I/O Lines
  - 28-pin PDIP, 32-lead TQFP, 28-pad QFN/MLF and 32-pad QFN/MLF
- Operating Voltage:
  - 1.8 - 5.5V for ATmega48PA/88PA/168PA/328P
- Temperature Range:
  - -40°C to 85°C
- Speed Grade:
  - 0 - 20 MHz @ 1.8 - 5.5V
- Low Power Consumption at 1 MHz, 1.8V, 25°C for ATmega48PA/88PA/168PA/328P:
  - Active Mode: 0.2 mA
  - Power-down Mode: 0.1 µA
  - Power-save Mode: 0.75 µA (Including 32 kHz RTC)



8-bit **AVR<sup>®</sup>**  
Microcontroller  
with 4/8/16/32K  
Bytes In-System  
Programmable  
Flash

ATmega48PA  
ATmega88PA  
ATmega168PA  
ATmega328P

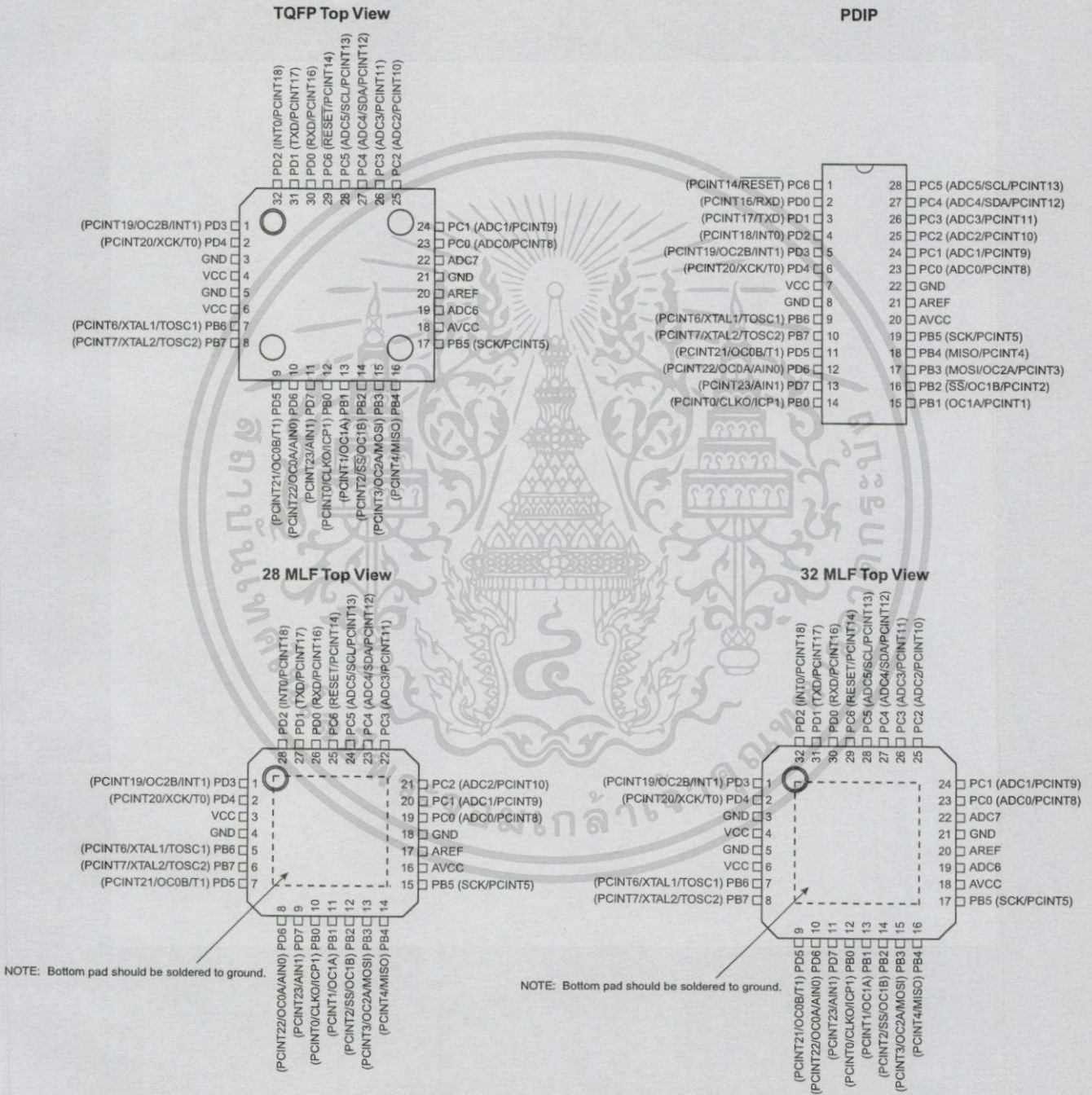
Rev. 8161D-AVR-10/09



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 1. Pin Configurations

Figure 1-1. Pinout ATmega48PA/88PA/168PA/328P



## 1.1 Pin Descriptions

### 1.1.1 VCC

Digital supply voltage.

### 1.1.2 GND

Ground.

### 1.1.3 Port B (PB7:0) XTAL1/XTAL2/TOSC1/TOSC2

Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Depending on the clock selection fuse settings, PB6 can be used as input to the inverting Oscillator amplifier and input to the internal clock operating circuit.

Depending on the clock selection fuse settings, PB7 can be used as output from the inverting Oscillator amplifier.

If the Internal Calibrated RC Oscillator is used as chip clock source, PB7..6 is used as TOSC2..1 input for the Asynchronous Timer/Counter2 if the AS2 bit in ASSR is set.

The various special features of Port B are elaborated in "Alternate Functions of Port B" on page 82 and "System Clock and Clock Options" on page 26.

### 1.1.4 Port C (PC5:0)

Port C is a 7-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The PC5..0 output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running.

### 1.1.5 PC6/RESET

If the RSTDISBL Fuse is programmed, PC6 is used as an I/O pin. Note that the electrical characteristics of PC6 differ from those of the other pins of Port C.

If the RSTDISBL Fuse is unprogrammed, PC6 is used as a Reset input. A low level on this pin for longer than the minimum pulse length will generate a Reset, even if the clock is not running. The minimum pulse length is given in Table 28-3 on page 318. Shorter pulses are not guaranteed to generate a Reset.

The various special features of Port C are elaborated in "Alternate Functions of Port C" on page 85.

### 1.1.6 Port D (PD7:0)

Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.



The various special features of Port D are elaborated in "Alternate Functions of Port D" on page 88.

## 1.1.7 $AV_{CC}$

$AV_{CC}$  is the supply voltage pin for the A/D Converter, PC3:0, and ADC7:6. It should be externally connected to  $V_{CC}$ , even if the ADC is not used. If the ADC is used, it should be connected to  $V_{CC}$  through a low-pass filter. Note that PC6..4 use digital supply voltage,  $V_{CC}$ .

## 1.1.8 AREF

AREF is the analog reference pin for the A/D Converter.

## 1.1.9 ADC7:6 (TQFP and QFN/MLF Package Only)

In the TQFP and QFN/MLF package, ADC7:6 serve as analog inputs to the A/D converter. These pins are powered from the analog supply and serve as 10-bit ADC channels.

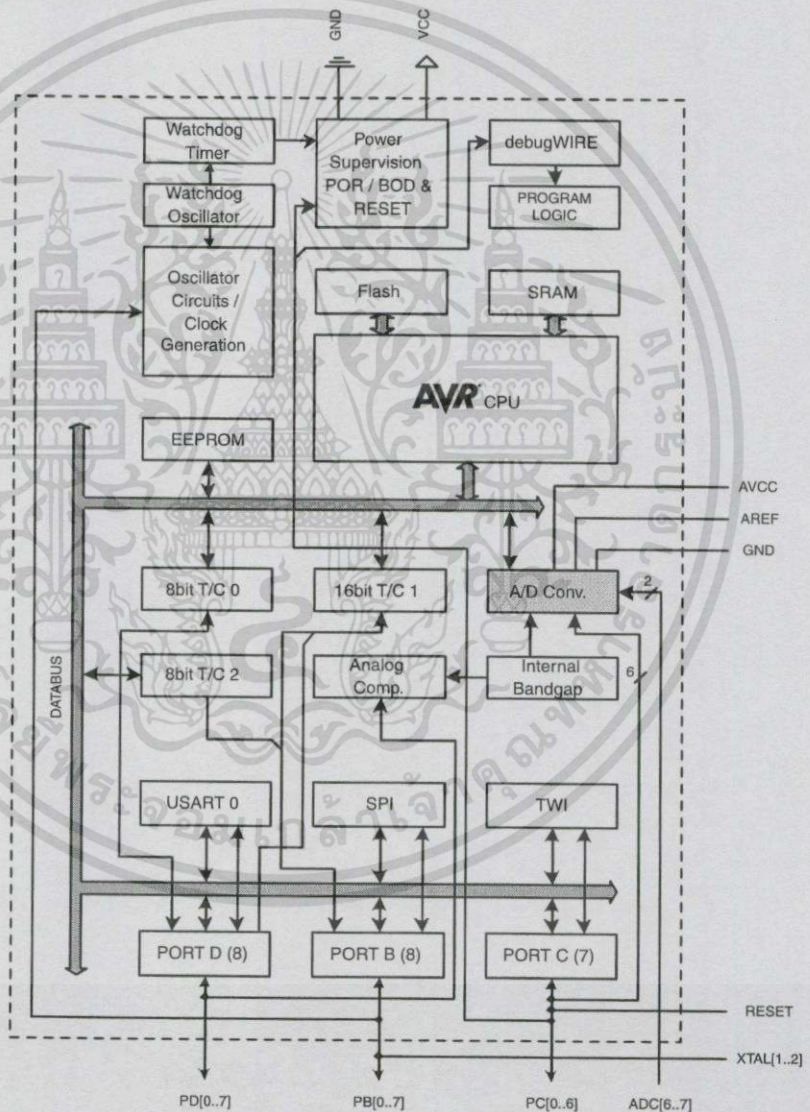


## 2. Overview

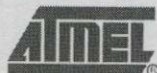
The ATmega48PA/88PA/168PA/328P is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega48PA/88PA/168PA/328P achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

### 2.1 Block Diagram

Figure 2-1. Block Diagram



The AVR core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting



architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

The ATmega48PA/88PA/168PA/328P provides the following features: 4K/8K bytes of In-System Programmable Flash with Read-While-Write capabilities, 256/512/512/1K bytes EEPROM, 512/1K/1K/2K bytes SRAM, 23 general purpose I/O lines, 32 general purpose working registers, three flexible Timer/Counters with compare modes, internal and external interrupts, a serial programmable USART, a byte-oriented 2-wire Serial Interface, an SPI serial port, a 6-channel 10-bit ADC (8 channels in TQFP and QFN/MLF packages), a programmable Watchdog Timer with internal Oscillator, and five software selectable power saving modes. The Idle mode stops the CPU while allowing the SRAM, Timer/Counters, USART, 2-wire Serial Interface, SPI port, and interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the Oscillator, disabling all other chip functions until the next interrupt or hardware reset. In Power-save mode, the asynchronous timer continues to run, allowing the user to maintain a timer base while the rest of the device is sleeping. The ADC Noise Reduction mode stops the CPU and all I/O modules except asynchronous timer and ADC, to minimize switching noise during ADC conversions. In Standby mode, the crystal/resonator Oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low power consumption.

The device is manufactured using Atmel's high density non-volatile memory technology. The On-chip ISP Flash allows the program memory to be reprogrammed In-System through an SPI serial interface, by a conventional non-volatile memory programmer, or by an On-chip Boot program running on the AVR core. The Boot program can use any interface to download the application program in the Application Flash memory. Software in the Boot Flash section will continue to run while the Application Flash section is updated, providing true Read-While-Write operation. By combining an 8-bit RISC CPU with In-System Self-Programmable Flash on a monolithic chip, the Atmel ATmega48PA/88PA/168PA/328P is a powerful microcontroller that provides a highly flexible and cost effective solution to many embedded control applications.

The ATmega48PA/88PA/168PA/328P AVR is supported with a full suite of program and system development tools including: C Compilers, Macro Assemblers, Program Debugger/Simulators, In-Circuit Emulators, and Evaluation kits.

## 2.2 Comparison Between ATmega48PA, ATmega88PA, ATmega168PA and ATmega328P

The ATmega48PA, ATmega88PA, ATmega168PA and ATmega328P differ only in memory sizes, boot loader support, and interrupt vector sizes. Table 2-1 summarizes the different memory and interrupt vector sizes for the three devices.

Table 2-1. Memory Size Summary

Device	Flash	EEPROM	RAM	Interrupt Vector Size
ATmega48PA	4K Bytes	256 Bytes	512 Bytes	1 instruction word/vector
ATmega88PA	8K Bytes	512 Bytes	1K Bytes	1 instruction word/vector
ATmega168PA	16K Bytes	512 Bytes	1K Bytes	2 instruction words/vector
ATmega328P	32K Bytes	1K Bytes	2K Bytes	2 instruction words/vector

ATmega88PA, ATmega168PA and ATmega328P support a real Read-While-Write Self-Programming mechanism. There is a separate Boot Loader Section, and the SPM instruction can only execute from there. In ATmega48PA, there is no Read-While-Write support and no separate Boot Loader Section. The SPM instruction can execute from the entire Flash.



## 3. Resources

A comprehensive set of development tools, application notes and datasheets are available for download on <http://www.atmel.com/avr>.

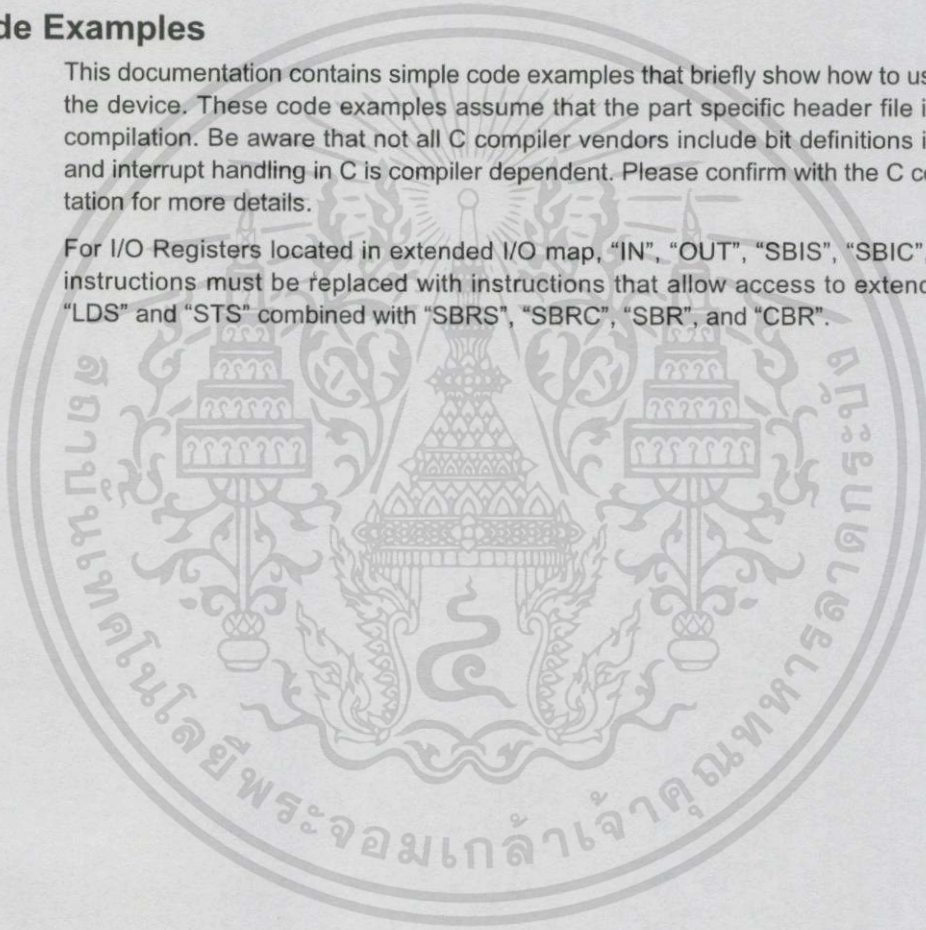
## 4. Data Retention

Reliability Qualification results show that the projected data retention failure rate is much less than 1 PPM over 20 years at 85°C or 100 years at 25°C.

## 5. About Code Examples

This documentation contains simple code examples that briefly show how to use various parts of the device. These code examples assume that the part specific header file is included before compilation. Be aware that not all C compiler vendors include bit definitions in the header files and interrupt handling in C is compiler dependent. Please confirm with the C compiler documentation for more details.

For I/O Registers located in extended I/O map, "IN", "OUT", "SBIS", "SBIC", "CBI", and "SBI" instructions must be replaced with instructions that allow access to extended I/O. Typically "LDS" and "STS" combined with "SBRS", "SBRC", "SBR", and "CBR".



## 28. Electrical Characteristics

### 28.1 Absolute Maximum Ratings\*

Operating Temperature.....	-55°C to +125°C
Storage Temperature .....	-65°C to +150°C
Voltage on any Pin except $\overline{\text{RESET}}$ with respect to Ground .....	-0.5V to $V_{CC}+0.5V$
Voltage on $\overline{\text{RESET}}$ with respect to Ground.....	-0.5V to +13.0V
Maximum Operating Voltage .....	6.0V
DC Current per I/O Pin .....	40.0 mA
DC Current $V_{CC}$ and GND Pins.....	200.0 mA

\*NOTICE: Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

### 28.2 DC Characteristics

$T_A = -40^\circ\text{C}$  to  $85^\circ\text{C}$ ,  $V_{CC} = 1.8V$  to  $5.5V$  (unless otherwise noted)

Symbol	Parameter	Condition	Min.	Typ.	Max.	Units
$V_{IL}$	Input Low Voltage, except XTAL1 and $\overline{\text{RESET}}$ pin	$V_{CC} = 1.8V - 2.4V$ $V_{CC} = 2.4V - 5.5V$	-0.5 -0.5		$0.2V_{CC}^{(1)}$ $0.3V_{CC}^{(1)}$	V
$V_{IH}$	Input High Voltage, except XTAL1 and $\overline{\text{RESET}}$ pins	$V_{CC} = 1.8V - 2.4V$ $V_{CC} = 2.4V - 5.5V$	$0.7V_{CC}^{(2)}$ $0.6V_{CC}^{(2)}$		$V_{CC} + 0.5$ $V_{CC} + 0.5$	V
$V_{IL1}$	Input Low Voltage, XTAL1 pin	$V_{CC} = 1.8V - 5.5V$	-0.5		$0.1V_{CC}^{(1)}$	V
$V_{IH1}$	Input High Voltage, XTAL1 pin	$V_{CC} = 1.8V - 2.4V$ $V_{CC} = 2.4V - 5.5V$	$0.8V_{CC}^{(2)}$ $0.7V_{CC}^{(2)}$		$V_{CC} + 0.5$ $V_{CC} + 0.5$	V
$V_{IL2}$	Input Low Voltage, $\overline{\text{RESET}}$ pin	$V_{CC} = 1.8V - 5.5V$	-0.5		$0.1V_{CC}^{(1)}$	V
$V_{IH2}$	Input High Voltage, $\overline{\text{RESET}}$ pin	$V_{CC} = 1.8V - 5.5V$	$0.9V_{CC}^{(2)}$		$V_{CC} + 0.5$	V
$V_{IL3}$	Input Low Voltage, $\overline{\text{RESET}}$ pin as I/O	$V_{CC} = 1.8V - 2.4V$ $V_{CC} = 2.4V - 5.5V$	-0.5 -0.5		$0.2V_{CC}^{(1)}$ $0.3V_{CC}^{(1)}$	V
$V_{IH3}$	Input High Voltage, $\overline{\text{RESET}}$ pin as I/O	$V_{CC} = 1.8V - 2.4V$ $V_{CC} = 2.4V - 5.5V$	$0.7V_{CC}^{(2)}$ $0.6V_{CC}^{(2)}$		$V_{CC} + 0.5$ $V_{CC} + 0.5$	V
$V_{OL}$	Output Low Voltage <sup>(3)</sup> except $\overline{\text{RESET}}$ pin	$I_{OL} = 20\text{ mA}$ , $V_{CC} = 5V$ $I_{OL} = 10\text{ mA}$ , $V_{CC} = 3V$			0.9 0.6	V
$V_{OH}$	Output High Voltage <sup>(4)</sup> except $\overline{\text{RESET}}$ pin	$I_{OH} = -20\text{ mA}$ , $V_{CC} = 5V$ $I_{OH} = -10\text{ mA}$ , $V_{CC} = 3V$	4.2 2.3			V
$I_{IL}$	Input Leakage Current I/O Pin	$V_{CC} = 5.5V$ , pin low (absolute value)			1	$\mu\text{A}$
$I_{IH}$	Input Leakage Current I/O Pin	$V_{CC} = 5.5V$ , pin high (absolute value)			1	$\mu\text{A}$



## 28.2.4 ATmega328P DC Characteristics

$T_A = -40^{\circ}\text{C}$  to  $85^{\circ}\text{C}$ ,  $V_{CC} = 1.8\text{V}$  to  $5.5\text{V}$  (unless otherwise noted)

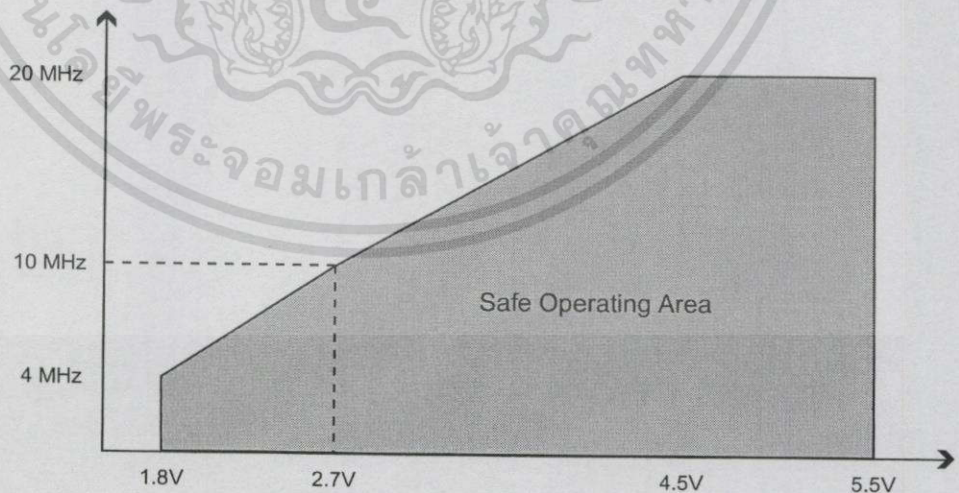
Symbol	Parameter	Condition	Min.	Typ. <sup>(2)</sup>	Max.	Units	
$I_{CC}$	Power Supply Current <sup>(1)</sup>	Active 1 MHz, $V_{CC} = 2\text{V}$		0.3	0.5	mA	
		Active 4 MHz, $V_{CC} = 3\text{V}$		1.7	2.5	mA	
		Active 8 MHz, $V_{CC} = 5\text{V}$		5.2	9	mA	
		Idle 1 MHz, $V_{CC} = 2\text{V}$		0.04	0.15	mA	
		Idle 4 MHz, $V_{CC} = 3\text{V}$		0.3	0.7	mA	
		Idle 8 MHz, $V_{CC} = 5\text{V}$		1.2	2.7	mA	
	Power-save mode <sup>(3)(4)</sup>	32 kHz TOSC enabled, $V_{CC} = 1.8\text{V}$			0.8	1.6	$\mu\text{A}$
		32 kHz TOSC enabled, $V_{CC} = 3\text{V}$			0.9	2.6	$\mu\text{A}$
	Power-down mode <sup>(3)</sup>	WDT enabled, $V_{CC} = 3\text{V}$			4.2	8	$\mu\text{A}$
		WDT disabled, $V_{CC} = 3\text{V}$			0.1	2	$\mu\text{A}$

- Notes:
1. Values with "Minimizing Power Consumption" enabled (0xFF).
  2. Typical values at  $25^{\circ}\text{C}$ . Maximum values are test limits in production.
  3. The current consumption values include input leakage current.
  4. Maximum values are characterized values and not test limits in production.

## 28.3 Speed Grades

Maximum frequency is dependent on  $V_{CC}$ . As shown in Figure 28-1, the Maximum Frequency vs.  $V_{CC}$  curve is linear between  $1.8\text{V} < V_{CC} < 2.7\text{V}$  and between  $2.7\text{V} < V_{CC} < 4.5\text{V}$ .

Figure 28-1. Maximum Frequency vs.  $V_{CC}$



## 28.4 Clock Characteristics

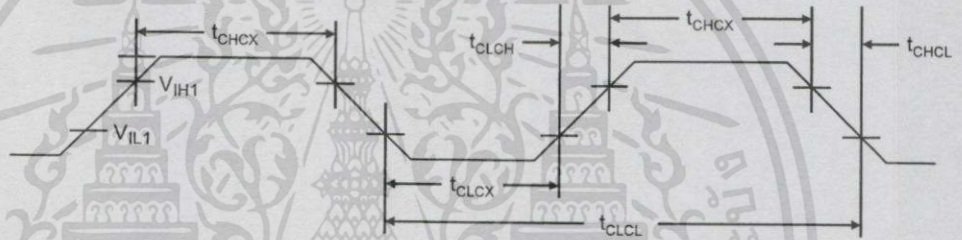
### 28.4.1 Calibrated Internal RC Oscillator Accuracy

**Table 28-1.** Calibration Accuracy of Internal RC Oscillator

	Frequency	V <sub>CC</sub>	Temperature	Calibration Accuracy
<b>Factory Calibration</b>	8.0 MHz	3V	25°C	±10%
<b>User Calibration</b>	7.3 - 8.1 MHz	1.8V - 5.5V	-40°C - 85°C	±1%

### 28.4.2 External Clock Drive Waveforms

**Figure 28-2.** External Clock Drive Waveforms



### 28.4.3 External Clock Drive

**Table 28-2.** External Clock Drive

Symbol	Parameter	V <sub>CC</sub> = 1.8 - 5.5V		V <sub>CC</sub> = 2.7 - 5.5V		V <sub>CC</sub> = 4.5 - 5.5V		Units
		Min.	Max.	Min.	Max.	Min.	Max.	
1/t <sub>CLCL</sub>	Oscillator Frequency	0	4	0	10	0	20	MHz
t <sub>CLCL</sub>	Clock Period	250		100		50		ns
t <sub>CHCX</sub>	High Time	100		40		20		ns
t <sub>CLCX</sub>	Low Time	100		40		20		ns
t <sub>CLCH</sub>	Rise Time		2.0		1.6		0.5	μs
t <sub>CHCL</sub>	Fall Time		2.0		1.6		0.5	μs
Δt <sub>CLCL</sub>	Change in period from one clock cycle to the next		2		2		2	%

**Note:** All DC Characteristics contained in this datasheet are based on simulation and characterization of other AVR microcontrollers manufactured in the same process technology. These values are preliminary values representing design targets, and will be updated after characterization of actual silicon.

## 28.5 System and Reset Characteristics

**Table 28-3.** Reset, Brown-out and Internal Voltage Characteristics<sup>(1)</sup>

Symbol	Parameter		Min	Typ	Max	Units
V <sub>POT</sub>	Power-on Reset Threshold Voltage (rising)		1.1	1.4	1.6	V
	Power-on Reset Threshold Voltage (falling) <sup>(2)</sup>		0.6	1.3	1.6	V
SR <sub>ON</sub>	Power-on Slope Rate		0.01		10	V/ms
V <sub>RST</sub>	$\overline{\text{RESET}}$ Pin Threshold Voltage		0.2 V <sub>CC</sub>		0.9 V <sub>CC</sub>	V
t <sub>RST</sub>	Minimum pulse width on $\overline{\text{RESET}}$ Pin				2.5	μs
V <sub>HYST</sub>	Brown-out Detector Hysteresis			50		mV
t <sub>BOD</sub>	Min Pulse Width on Brown-out Reset			2		μs
V <sub>BG</sub>	Bandgap reference voltage	V <sub>CC</sub> =2.7 T <sub>A</sub> =25°C	1.0	1.1	1.2	V
t <sub>BG</sub>	Bandgap reference start-up time	V <sub>CC</sub> =2.7 T <sub>A</sub> =25°C		40	70	μs
I <sub>BG</sub>	Bandgap reference current consumption	V <sub>CC</sub> =2.7 T <sub>A</sub> =25°C		10		μA

- Notes: 1. Values are guidelines only.  
 2. The Power-on Reset will not work unless the supply voltage has been below V<sub>POT</sub> (falling)

**Table 28-4.** BODLEVEL Fuse Coding<sup>(1)</sup>

BODLEVEL 2:0 Fuses	Min V <sub>BOT</sub>	Typ V <sub>BOT</sub>	Max V <sub>BOT</sub>	Units
111				V
110	1.7	1.8	2.0	
101	2.5	2.7	2.9	
100	4.1	4.3	4.5	
011				Reserved
010				
001				
000				

- Notes: 1. V<sub>BOT</sub> may be below nominal minimum operating voltage for some devices. For devices where this is the case, the device is tested down to V<sub>CC</sub> = V<sub>BOT</sub> during the production test. This guarantees that a Brown-Out Reset will occur before V<sub>CC</sub> drops to a voltage where correct operation of the microcontroller is no longer guaranteed. The test is performed using BODLEVEL = 110, 101 and 100.



## 28.6 SPI Timing Characteristics

See Figure 28-3 and Figure 28-4 for details.

**Table 28-5. SPI Timing Parameters**

	Description	Mode	Min	Typ	Max	
1	SCK period	Master		See Table 18-5		
2	SCK high/low	Master		50% duty cycle		
3	Rise/Fall time	Master		3.6		
4	Setup	Master		10		
5	Hold	Master		10		
6	Out to SCK	Master		$0.5 \cdot t_{sck}$		
7	SCK to out	Master		10		
8	SCK to out high	Master		10		
9	$\overline{SS}$ low to out	Slave		15		
10	SCK period	Slave	$4 \cdot t_{ck}$		ns	
11	SCK high/low <sup>(1)</sup>	Slave	$2 \cdot t_{ck}$			
12	Rise/Fall time	Slave				1600
13	Setup	Slave	10			
14	Hold	Slave	$t_{ck}$			
15	SCK to out	Slave		15		
16	SCK to $\overline{SS}$ high	Slave	20			
17	$\overline{SS}$ high to tri-state	Slave		10		
18	$\overline{SS}$ low to SCK	Slave	20			

- Note:
- In SPI Programming mode the minimum SCK high/low period is:
    - $2 t_{CLCL}$  for  $f_{CK} < 12$  MHz
    - $3 t_{CLCL}$  for  $f_{CK} > 12$  MHz
  - All DC Characteristics contained in this datasheet are based on simulation and characterization of other AVR microcontrollers manufactured in the same process technology. These values are preliminary values representing design targets, and will be updated after characterization of actual silicon.



## 29. Typical Characteristics

The following charts show typical behavior. These figures are not tested during manufacturing. All current consumption measurements are performed with all I/O pins configured as inputs and with internal pull-ups enabled. A square wave generator with rail-to-rail output is used as clock source.

All Active- and Idle current consumption measurements are done with all bits in the PRR register set and thus, the corresponding I/O modules are turned off. Also the Analog Comparator is disabled during these measurements. The "ATmega88PA: Supply Current of IO Modules" on page 356 and page 380 shows the additional current consumption compared to  $I_{CC}$  Active and  $I_{CC}$  Idle for every I/O module controlled by the Power Reduction Register. See "Power Reduction Register" on page 42 for details.

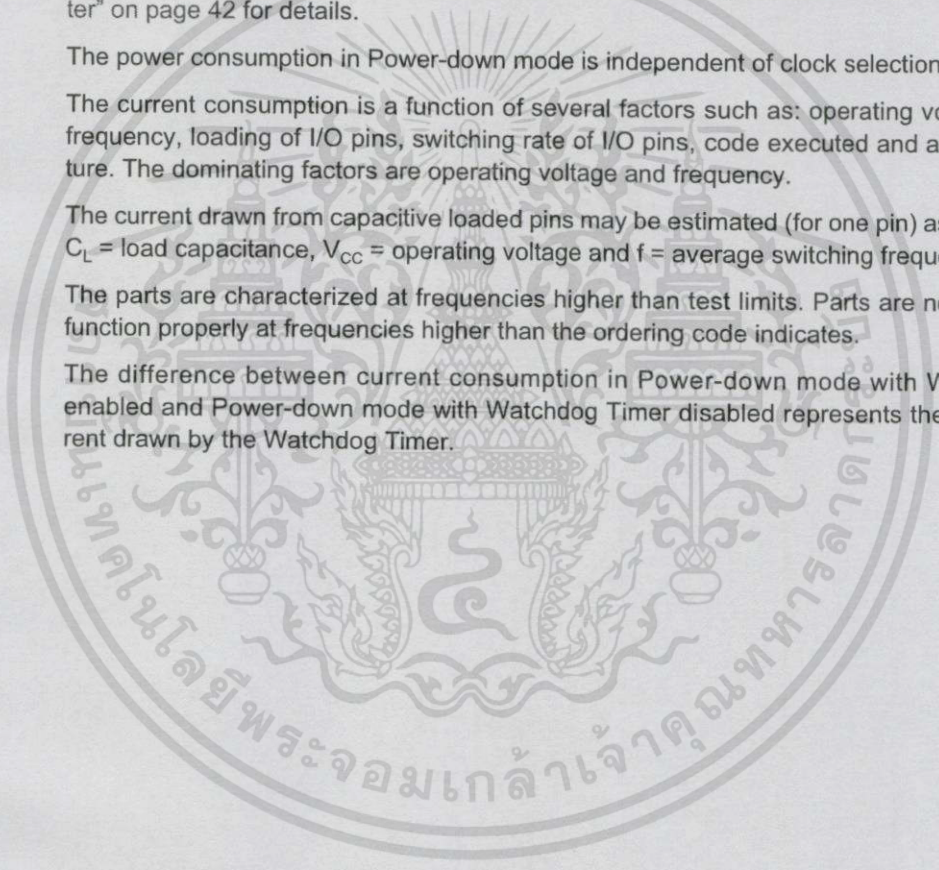
The power consumption in Power-down mode is independent of clock selection.

The current consumption is a function of several factors such as: operating voltage, operating frequency, loading of I/O pins, switching rate of I/O pins, code executed and ambient temperature. The dominating factors are operating voltage and frequency.

The current drawn from capacitive loaded pins may be estimated (for one pin) as  $C_L \cdot V_{CC} \cdot f$  where  $C_L$  = load capacitance,  $V_{CC}$  = operating voltage and  $f$  = average switching frequency of I/O pin.

The parts are characterized at frequencies higher than test limits. Parts are not guaranteed to function properly at frequencies higher than the ordering code indicates.

The difference between current consumption in Power-down mode with Watchdog Timer enabled and Power-down mode with Watchdog Timer disabled represents the differential current drawn by the Watchdog Timer.



## 29.4 ATmega328P Typical Characteristics

### 29.4.1 Active Supply Current

Figure 29-139. ATmega328P: Active Supply Current vs. Low Frequency (0.1-1.0 MHz)

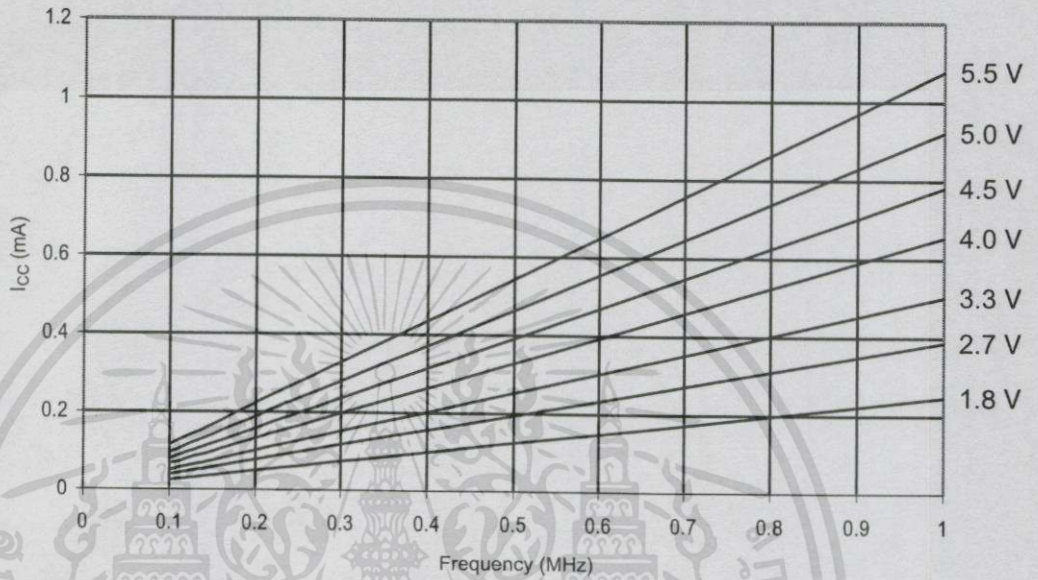
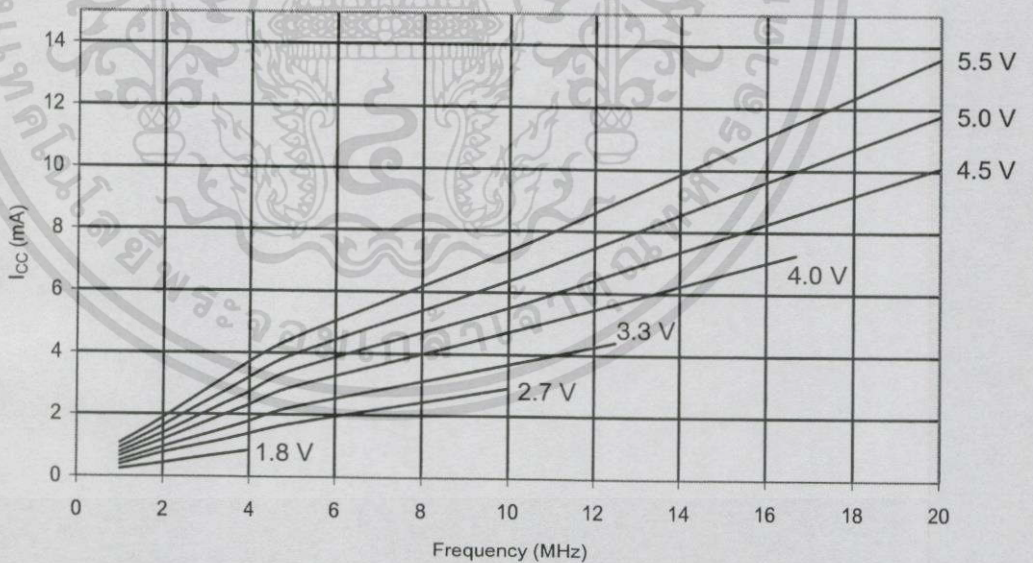
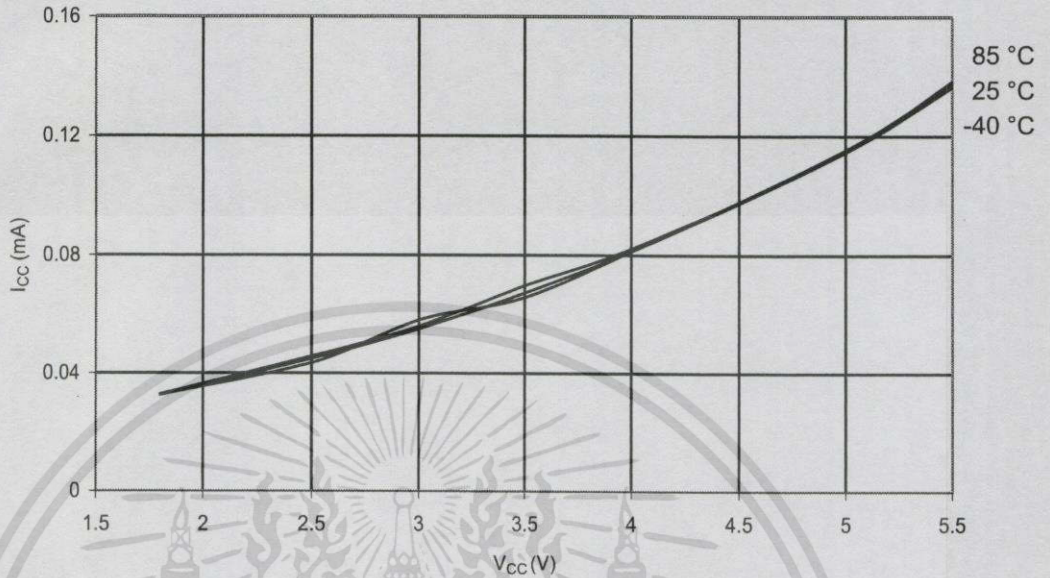


Figure 29-140. ATmega328P: Active Supply Current vs. Frequency (1-20 MHz)



**Figure 29-141.** ATmega328P: Active Supply Current vs.  $V_{CC}$  (Internal RC Oscillator, 128 kHz)



**Figure 29-142.** ATmega328P: Active Supply Current vs.  $V_{CC}$  (Internal RC Oscillator, 1 MHz)

