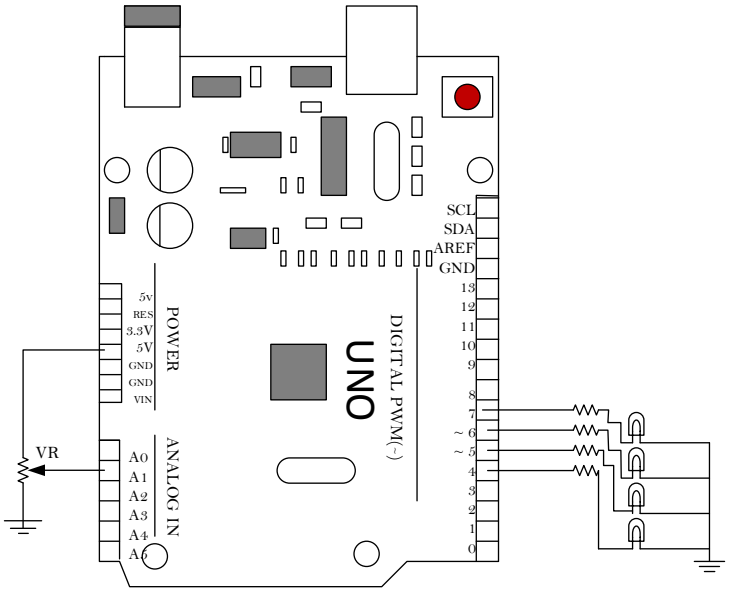
เอกสารประกอบการสอน
วิชา ไมโครคอนโทรลเลอร์ รหัสวิชา 2105 – 2105
ตามหลักสูตรประกาศนียบัตรวิชาชีพ พ.ศ. 2556
ประเภทวิชาช่างอุตสาหกรรม



นายวิโรจน์ กิตติวรปรีดา
ตำแหน่ง ครู วิทยฐานะ ชำนาญการ

สาขาวิชาช่างอิเล็กทรอนิกส์ วิทยาลัยเทคนิคราชบุรี
สำนักงานคณะกรรมการการอาชีวศึกษา กระทรวงศึกษาธิการ

คำนำ

เอกสารประกอบการสอนวิชาไมโครคอนโทรลเลอร์ รหัสวิชา 2105-2105 ตามหลักสูตรประกาศนียบัตรวิชาชีพ พุทธศักราช 2556 สาขาวิชาอิเล็กทรอนิกส์ สำนักงานคณะกรรมการการอาชีวะศึกษา กระทรวงศึกษาธิการ เล่มนี้ ผู้จัดทำได้ศึกษาและเรียบเรียงจากประสบการณ์การสอน การฝึกอบรมสัมมนา จากตำราเอกสารวิชาการต่างๆ และแหล่งข้อมูลทางอินเตอร์เน็ต โดยเรียบเรียงให้ตรงกับ จุดประสงค์รายวิชา สมรรถนะรายวิชาและคำอธิบายรายวิชา ตามที่หลักสูตรกำหนด ซึ่งประกอบด้วยหน่วยการเรียนทั้งหมด 14 หน่วย ประกอบด้วย โครงสร้างและส่วนประกอบของไมโครคอนโทรลเลอร์ เครื่องมือสำหรับพัฒนาไมโครคอนโทรลเลอร์ การเขียนโปรแกรมควบคุมไมโครคอนโทรลเลอร์สำหรับ Arduino การเขียนโปรแกรมภาษาซีสำหรับ Arduino การกำหนด Digital และ Analog Pin การควบคุมหลอดไฟแสดงผล LED การควบคุมตัวแสดงผล LED 7 ส่วน การควบคุมหน้าจอแสดงผล LCD การเชื่อมต่อ Arduino กับ สวิทซ์ การเชื่อมต่อ Arduino กับ Relay การอินเตอร์รัพต์ของไมโครคอนโทรลเลอร์ การใช้งานเซนเซอร์ LDR การใช้งานเซนเซอร์วัดอุณหภูมิ การควบคุมมอเตอร์ไฟฟ้ากระแสตรง

เอกสารประกอบการสอนเล่มนี้สำเร็จลุล่วงไปด้วยดี ผู้จัดทำต้องขอบพระคุณคณะวิทยากรโครงการพัฒนาข้าราชการครูและบุคลากรทางการศึกษาก่อนแต่งตั้งให้มีวิทยฐานะครูชำนาญการพิเศษ สำนักพัฒนาสมรรถนะครูและบุคลากรอาชีวะศึกษา ผู้เชี่ยวชาญทุกท่านที่ตรวจความถูกต้องและให้ข้อเสนอแนะอย่างดียิ่ง ผู้เรียบเรียงคาดหวังว่าเอกสารประกอบการสอนเล่มนี้จะเป็นประโยชน์ต่อผู้เรียน ครูผู้สอน และบุคคลต่างๆ ในวงกว้างต่อไป และขอน้อมรับความผิดพลาดและข้อบกพร่องต่างๆที่เกิดขึ้นกับเอกสารประกอบการสอนเล่มนี้มา ณ ที่นี้ด้วย

.....

(นายวิโรจน์ กิตติวรปรีดา)

ผู้จัดทำ

รายละเอียดรายวิชา

หลักสูตร ประกาศนียบัตรวิชาชีพ พุทธศักราช 2556
ประเภท วิชาอุตสาหกรรม
รหัสวิชา 2105 – 2105 ชื่อวิชา ไมโครคอนโทรลเลอร์
ทฤษฎี 1 ชั่วโมง/สัปดาห์ ปฏิบัติ 3 ชั่วโมง/สัปดาห์ จำนวน 2 หน่วยกิต

จุดประสงค์ของรายวิชา

1. เข้าใจโครงสร้างการทำงาน ชุดคำสั่ง และการเขียนโปรแกรมไมโครคอนโทรลเลอร์
2. มีทักษะในการใช้คำสั่ง และการเขียนโปรแกรมควบคุมไมโครคอนโทรลเลอร์
3. มีกิจนิวัยในการแสวงหาความารู้เพิ่มเติม การทำงานด้วยความประณีต รอบคอบและปลอดภัย

สมรรถนะรายวิชา

1. แสวงหาความรู้เกี่ยวกับการประยุกต์ใช้งานไมโครคอนโทรลเลอร์
2. ประกอบและทดสอบไมโครคอนโทรลเลอร์
3. ประยุกต์ใช้งานไมโครคอนโทรลเลอร์

คำอธิบายรายวิชา

ศึกษาและปฏิบัติเกี่ยวกับโครงสร้างส่วนประกอบและหน้าที่ในส่วนต่างๆ ของไมโครคอนโทรลเลอร์ ชุดคำสั่งและการเขียนโปรแกรมด้วยภาษาแอสแซมบลีและภาษาระดับสูงของไมโครคอนโทรลเลอร์ การควบคุมระบบด้วยไมโครคอนโทรลเลอร์ การต่อวงจรและการประยุกต์ใช้งาน

ตารางที่ 1 วิเคราะห์หน่วยการเรียนรู้และเวลาที่ใช้ในการจัดการเรียนรู้วิชา
รหัสวิชา 2105 – 2105 ชื่อวิชา ไมโครคอนโทรลเลอร์
ทฤษฎี 1 ชั่วโมง/สัปดาห์ ปฏิบัติ 3 ชั่วโมง/สัปดาห์ จำนวน 2 หน่วยกิต

หน่วยที่	ชื่อหน่วยการเรียนรู้	สัปดาห์ที่	เวลา (ชั่วโมง)
1	โครงสร้างและส่วนประกอบของไมโครคอนโทรลเลอร์	1	4 (1)
2	เครื่องมือสำหรับพัฒนาไมโครคอนโทรลเลอร์	2	4 (1)
3	การเขียนโปรแกรมควบคุมไมโครคอนโทรลเลอร์สำหรับ Arduino	3	4 (1)
4	การเขียนโปรแกรมภาษาซีสำหรับ Arduino (1) – (3)	4-6	12 (3)
5	การกำหนด Digital และ Analog Pin	7	4 (1)
6	การควบคุมหลอดไฟแสดงผล LED	8-9	8 (2)
7	การควบคุมตัวแสดงผล LED 7 ส่วน	10	4 (1)
8	การควบคุมหน้าจอแสดงผล LCD	11	4 (1)
9	การเชื่อมต่อ Arduino กับ สวิทซ์	12	4 (1)
10	การเชื่อมต่อ Arduino กับ Relay	13	4 (1)
11	การอินเตอร์รัพต์ของไมโครคอนโทรลเลอร์	14	4 (1)
12	การใช้งานเซนเซอร์ LDR	15	4 (1)
13	การใช้งานเซนเซอร์วัดอุณหภูมิ	16	4 (1)
14	การควบคุมมอเตอร์ไฟฟ้ากระแสตรง	17	4 (1)
-	สอบปลายภาค	18	4 (1)
รวม			72 (18)

ตารางที่ 2 วิเคราะห์ หัวข้อการเรียนรู้หลัก/หัวข้อรอง/หัวข้อย่อย และเวลาจัดการเรียนรู้

รหัสวิชา 2105 – 2105 ชื่อวิชา ไมโครคอนโทรลเลอร์

ทฤษฎี 1 ชั่วโมง/สัปดาห์ ปฏิบัติ 3 ชั่วโมง/สัปดาห์ จำนวน 2 หน่วยกิต

สัปดาห์ที่	หน่วยที่	ชื่อหน่วยการเรียนรู้/หัวข้อการเรียนรู้	เวลาจัดการเรียนรู้		
			ทฤษฎี	ปฏิบัติ	รวม (ชม.)
1	1	โครงสร้างและส่วนประกอบของไมโครคอนโทรลเลอร์ 1.1 แนะนำรายวิชาและเกณฑ์การให้คะแนน 1.2 โครงสร้างไมโครคอนโทรลเลอร์ 1.3 สถาปัตยกรรมของไมโครคอนโทรลเลอร์ 1.4 ไมโครโพรเซสเซอร์กับไมโครคอนโทรลเลอร์ 1.5 หลักการทำงานของไมโครคอนโทรลเลอร์	4	0	4
2	2	เครื่องมือสำหรับพัฒนาไมโครคอนโทรลเลอร์ 2.1 ประวัติของบอร์ด Arduino 2.2 โครงสร้างของ Arduinoบอร์ด Arduino รุ่นต่างๆ	1	3	4
3	3	การเขียนโปรแกรมควบคุมไมโครคอนโทรลเลอร์สำหรับ Arduino 3.1 ภาษาคอมพิวเตอร์ 3.2 การเขียนโฟลว์ชาร์ต	1	3	4
4-6	4	การเขียนโปรแกรมภาษาซีสำหรับ Arduino (1) – (3) 4.1 โครงสร้างภาษาซีสำหรับ Arduino 4.2 คำสั่งเงื่อนไขภาษาซีสำหรับ Arduino 4.3 คำสั่งเงื่อนไขภาษาซีสำหรับ Arduino	3	9	12
7	5	การกำหนด Digital และ Analog Pin 5.1 กลุ่มคำสั่งดิจิตอลอินพุต/เอาต์พุต 5.2 กลุ่มคำสั่งแอนาลอกอินพุต/เอาต์พุต	1	3	4

ตารางที่ 2 (ต่อ) วิเคราะห์ หัวข้อการเรียนรู้หลัก/หัวข้อรอง/หัวข้อย่อย และเวลาจัดการเรียนรู้
รหัสวิชา 2105 – 2105 ชื่อวิชา ไมโครคอนโทรลเลอร์
ทฤษฎี 1 ชั่วโมง/สัปดาห์ ปฏิบัติ 3 ชั่วโมง/สัปดาห์ จำนวน 2 หน่วยกิต

สัปดาห์ที่	หน่วยที่	ชื่อหน่วยการเรียนรู้/หัวข้อการเรียนรู้	เวลาจัดการเรียนรู้		
			ทฤษฎี	ปฏิบัติ	รวม (ชม.)
8-9	6	การควบคุมหลอดไฟแสดงผล LED 6.1 การเขียนโปรแกรมควบคุมการทำซ้ำกับ LED 6.2 โครงสร้างและการทำงานของหลอดไฟแสดงผล LED	2	6	8
10	7	การควบคุมตัวแสดงผล LED 7 ส่วน 7.1 โครงสร้างและการทำงานของหลอดแสดงผล LED 7 ส่วน 7.2 การใช้งาน LED 7 ส่วน	1	3	4
11	8	การควบคุมหน้าจอแสดงผล LCD 8.1 การเชื่องต่อจอแสดงผล LCD กับ Arduino 8.2 การใช้งานหน้าจอแสดงผล LCD	1	3	4
12	9	การเชื่อมต่อ Arduino กับ สวิตช์ 9.1 การทำงานและการต่อสวิตช์	1	3	4
13	10	การเชื่อมต่อ Arduino กับ Relay 10.1 โครงสร้างและการทำงานของ Relay	1	3	4
14	11	การอินเตอร์รัพต์ของไมโครคอนโทรลเลอร์ 11.1 การอินเตอร์รัพท์ 11.2 การควบคุมการอินเตอร์รัพท์กับ Arduino	1	3	4
15	12	การใช้งานเซนเซอร์ LDR 12.1 โครงสร้างและการทำงานของ LDR 12.2 การต่อใช้งาน LDR กับ Arduino	1	3	4
16	13	การใช้งานเซนเซอร์วัดอุณหภูมิ 13.1 การใช้งานของเซนเซอร์วัดอุณหภูมิ 13.2 การต่อใช้งานเซนเซอร์วัดอุณหภูมิกับ Arduino 13.3 การต่อใช้งานเซนเซอร์วัดความเคลื่อนไหวกับ Arduino	1	3	4

ตารางที่ 2 (ต่อ) วิเคราะห์ หัวข้อการเรียนรู้หลัก/หัวข้อรอง/หัวข้อย่อย และเวลาจัดการเรียนรู้

รหัสวิชา 2105 – 2105 ชื่อวิชา ไมโครคอนโทรลเลอร์

ทฤษฎี 1 ชั่วโมง/สัปดาห์ ปฏิบัติ 3 ชั่วโมง/สัปดาห์ จำนวน 2 หน่วยกิต

สัปดาห์ที่	หน่วยที่	ชื่อหน่วยการเรียนรู้/หัวข้อการเรียนรู้	เวลาจัดการเรียนรู้		
			ทฤษฎี	ปฏิบัติ	รวม (ชม.)
17	14	การควบคุมมอเตอร์ไฟฟ้ากระแสตรง 14.1 โครงสร้างและการทำงานของมอเตอร์ไฟฟ้ากระแสตรง 14.2 การต่อใช้งานมอเตอร์ไฟฟ้ากระแสตรงกับ Arduino	1	3	4
18	-	สอบปลายภาค	1	3	4

<table>
<tr><td rowspan="3"> </td><td colspan="2">โครงการสอน</td></tr>
<tr><td colspan="2">รหัสวิชา 2105 – 2105 ชื่อวิชา ไมโครคอนโทรลเลอร์</td></tr>
<tr><td colspan="2">ชื่อหน่วยที่ 5 จำนวน 4 ชั่วโมง</td></tr>
<tr><td colspan="3">เรื่อง การกำหนด Digital และ Analog Pin</td></tr>
<tr><td>จุดประสงค์การสอน</td><td colspan="2">รายการสอน</td></tr>
<tr><td>ทฤษฎี
1. อธิบายการใช้งานของกลุ่มคำสั่งเกี่ยวกับ Digital I/O Pin ได้อย่างถูกต้อง
2. อธิบายการใช้งานของกลุ่มคำสั่งเกี่ยวกับ Analog I/O Pin ได้อย่างถูกต้อง
3. อธิบายหลักการคำนวณค่า ADC ได้อย่างถูกต้อง

ปฏิบัติ
1. เขียนโปรแกรมใช้งานของกลุ่มคำสั่งเกี่ยวกับ Digital I/O ได้อย่างถูกต้อง
2. เขียนโปรแกรมใช้งานของกลุ่มคำสั่งเกี่ยวกับ Analog I/O ได้อย่างถูกต้อง
3. คำนวณค่า ADC ได้อย่างถูกต้อง</td><td colspan="2">1.กลุ่มคำสั่งแอนาลอกอินพุต/เอาต์พุต
- void pinMode(pin,mode)
- void digitalWrite(pin,value)
- int digitalRead(pin)
2.กลุ่มคำสั่งแอนาลอกอินพุต/เอาต์พุต
- analogReference(type)
- int analogRead(pin)
- void analogWrite(Pin,value)
3.การคำนวณค่า ADC</td></tr>
<tr><td colspan="3">วิธีการสอน
1. บรรยายประกอบ power point
2. สาธิต
3. ทดลอง</td></tr>
</table>

สื่อการสอน

1. ใบความรู้ เรื่อง การกำหนด Digital และ Analog Pin
2. Power Point เรื่อง การกำหนด Digital และ Analog Pin
3. ใบงานการทดลอง เรื่อง การกำหนด Digital และ Analog Pin
4. ของจริง โปรแกรม Arduino IDE, ชุดทดลองไมโครคอนโทรลเลอร์

เอกสารอ้างอิง

ทะนงศักดิ์ สัตนาโค.2559.**ไมโครคอยโทรลเลอร์เบื้องต้น**.กรุงเทพมหานคร: สำนักพิมพ์ บริษัท พัฒนาคุณภาพวิชาการ (พว.) จำกัด

ทันพงษ์ ภู่รักษ์.**เอกสารประกอบวิชาไมโครคอนโทรลเลอร์เบื้องต้น.** [Online] Available: http://www.sbt.ac.th/new/sites/default/files/TNP_Unit_1.pdf เข้าถึงวันที่ 11 พฤษภาคม 2560

ปิยะ ศุภวราสุวัฒน์.2559.**ไมโครคอนโทรลเลอร์**.กรุงเทพมหานคร:ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.

ประพันธ์ พิพัฒนสุข และ ธีระพันธ์ พิพัฒนสุข.2557.**ไมโครคอนโทรลเลอร์**.กรุงเทพมหานคร: ศูนย์ส่งเสริมอาชีวะ

สุชิน ชินสีห์.2557.**ไมโครคอนโทรลเลอร์**.นนทบุรี: บริษัทศูนย์หนังสือ เมืองไทย จำกัด

ประจิน พลังสันติกุล.2558.**พื้นฐานภาษา C สำหรับ Arduino**.กรุงเทพมหานคร :บริษัท แอพซอฟต์เทค จำกัด

เอกขัย มะการ.2552.เรียนรู้ เข้าใจ ใช้งาน ไมโครคอนโทรลเลอร์ตระกูล AVR ด้วย Arduino.กรุงเทพฯ: บริษัท อีทีที จำกัด

การประเมิน

1. แบบประเมินผลการปฏิบัติงาน
2. แบบทดสอบหลังเรียน

แหล่งเรียนรู้เพิ่มเติม

-อินเตอร์เน็ต

แผนการจัดการเรียนรู้

รหัสวิชา 2105 – 2105 ชื่อวิชา ไมโครคอนโทรลเลอร์ สอนครั้งที่ 7
หน่วยที่ 5 ชื่อหน่วย การกำหนดดิจิตอลและแอนาลอก อินพุตและเอาต์พุต จำนวน 4 ชม.

หัวข้อเรื่อง

5.1 กลุ่มคำสั่งดิจิตอลอินพุตและเอาต์พุต (digital I/O)
5.2 กลุ่มคำสั่งแอนาลอกอินพุตและเอาต์พุต (Analog I/O)

สาระสำคัญ

ทฤษฎี

สำหรับดิจิตอลอินพุตและเอาต์พุต ใน Arduino (อาดุยโน) นั้น จะมีทั้งหมดจำนวน 14 pin (พิน) โดย Arduino ได้กำหนดรหัสของตัวเลขซึ่งเป็นเลขจำนวนเต็มค่าระหว่าง 0 ถึง 13 สำหรับอ้างถึงดิจิตอลอินพุตและเอาต์พุต ทั้ง 14 pin แต่ตามปรกติแล้วจะมีดิจิตอลอินพุตและเอาต์พุต pin จำนวน 2 pin ซึ่งถูกสงวนไว้สำหรับใช้เป็นพอร์ตสื่อสารอนุกรม RS232 สำหรับใช้ Upload Code ของ โปรแกรมให้กับบอร์ด จึงเหลือดิจิตอลอินพุตและเอาต์พุต สำหรับใช้งานดิจิตอลอินพุตและเอาต์พุต จริงๆจำนวน 12 pin คือดิจิตอลอินพุตและเอาต์พุต หมายเลข 2 ถึง 13 เท่านั้นโดยคำสั่งในกลุ่มนี้จะมีอยู่ด้วยกัน 3 คำสั่ง คือ

- **void pinMode(pin,mode)**
- **void digitalWrite(pin,value)**
- **int digitalRead(pin)**

การใช้งานแอนาลอก (Analog Input / Output)

คำสั่งในกลุ่มนี้เป็นกลุ่มคำสั่งสำหรับใช้งาน Pin อินพุตและเอาต์พุตของแอนาลอกอินพุตและเอาต์พุต ซึ่งตามปกติแล้ว Arduino จะมีขาสัญญาณที่ใช้ทำหน้าที่เป็นแอนาลอกอินพุตได้ทั้งหมด 6 ถึง 8 pin โดยถ้าใช้กับ AVR รุ่นที่เป็นตัวถัง SMD แบบ 32 ขา ซึ่งขาสัญญาณในกลุ่มที่เป็นแอนาลอกอินพุตนี้ จะใช้รหัสตัวเลข 0 ถึง 7 ในการอ้างอิงขาสัญญาณ โดยที่คุณสมบัติของสัญญาณในกลุ่มนี้จะสามารถใช้วัด หรือ อ่านค่าสัญญาณที่เป็นแบบแรงดันไฟฟ้า ซึ่งในทางอิเล็กทรอนิกส์ เรียกสัญญาณแบบนี้ว่า สัญญาณแอนาลอก โดยที่ขาแอนาลอกสามารถรับแรงดันอินพุต ที่มีขนาดแรงดันระหว่าง 0-5 โวลต์ แล้วแปลงค่าขนาดของแรงดันที่อ่านได้มาเป็นข้อมูลแบบตัวเลขจำนวนเต็ม ซึ่งมีค่าความละเอียด 1024 ค่า (0 ถึง 1023) ซึ่งความสามารถของ Arduino ในการอ่านและวัดค่าระดับของแรงดันที่เป็นแอนาลอกนี้จะช่วยทำให้เราสามารถนำไมโครคอนโทรลเลอร์ไปประยุกต์ใช้งาน เกี่ยวกับการตรวจจับอินพุต แบบต่างๆ ได้มากมายหลายแบบโดยใช้งานร่วมกับอุปกรณ์ตรวจจับ (sensor) แบบต่างๆ ที่ให้ค่าออกมาเป็นแรงดัน

ส่วนแอนาลอกเอาต์พุต นั้น ตามปกติแล้วจะต้องใช้วงจรแปลงสัญญาณแบบ Digital to Analog หรือ D/A ซึ่งวงจรแบบนี้ไม่มีบรรจุอยู่ในชิพของ AVR ด้วยเลย แต่อย่างไรก็ตาม Arduino ได้ดัดแปลงเอาความสมารถในการสร้างสัญญาณ PWM ของ AVR มาประยุกต์ดัดแปลงเพื่อใช้สร้างเป็น

สัญญาณแอนาลอกเอาต์พุตได้ด้วย โดยใช้เทคนิคการสร้างสัญญาณ PWM ขึ้นมา แล้วทำการปรับขนาดของ Duty Cycle หรือ ช่วงเวลาที่สัญญาณ PWM มีสถาวะลอจิกเป็น HIGH ให้มีค่ามากน้อยต่างๆ กันไปโดยใช้ค่าความถี่เท่าเดิม ซึ่งผลของการปรับค่าสัญญาณ PWM ด้วยการเปลี่ยนแปลงค่า Duty Cycle นี้จำให้ได้ค่าเฉลี่ยของขนาดแรงดันที่เปลี่ยนแปลงไปตามค่าการปรับของ Duty cycle ของสัญญาณ PWM ด้วยซึ่งผลลัพธ์ก็จะเปรียบเสมือนกับการปรับขนาดของสัญญาณแอนาลอกเอาต์พุต pin ได้จำนวน 6 pin คือ ดิจิตอลอินพุตและเอาต์พุตเท่านั้นเอง โดยคำสั่งของแอนาลอกอินพุตและเอาต์พุตของ Arduino จะมีอยู่ 3 คำสั่งคือ

- **analogReference(type)**
- **int analogRead(pin)**
- **void analogWrite(Pin,value)**

ปฏิบัติ

เป็นการใช้คำสั่งในการควบคุมการทำงานของดิจิตอลอินพุตและเอาต์พุต Pin และแอนาลอกอินพุตและเอาต์พุต Pin ซึ่งการใช้งานดิจิตอลอินพุตและเอาต์พุต Pin เป็นการเขียนโปรแกรมให้เอาต์พุต ควบคุมการทำงานของ LED และอินพุตเป็นการรับค่าการทำงานของสวิตซ์ การใช้งานแอนาลอกอินพุตและเอาต์พุต Pin เป็นการเขียนโปรแกรมให้เอาต์พุต ควบคุมการทำงานของ PWM และอินพุต เป็นการรับค่ามาจากแรงดันภายนอกอ่านค่าไปแสดงผลที่จอแสดงผลของพอร์ตอนุกรม สำหรับอ่านค่าระดับแรงดันที่ขาสัญญาณของแอนาลอกอินพุต Pin ว่าระดับแรงดัน (Analog Input) มีค่าระดับแรงดันเป็นเท่าใด (0 - Analog Reference) ซึ่งทำการแปลงค่ามาเป็นข้อมูลแบบดิจิตอลขนาด 10 บิต เรียบร้อยแล้ว โดยผลลัพธ์ที่ได้จะมีค่า 1024 ระดับ ขนาดของแรงดันต่อขนาดของข้อมูล 1 ระดับ ประมาณเท่ากับ 5v/1024 หรือ ประมาณ 0.0049V (4.9mV)

จุดประสงค์การเรียนรู้

จุดประสงค์ทั่วไป

1. เพื่อศึกษาการใช้งานของกลุ่มคำสั่งเกี่ยวกับ Digital I/O Pin ได้อย่างถูกต้อง
2. เพื่อศึกษาการการใช้งานของกลุ่มคำสั่งเกี่ยวกับ Analog I/O Pin ได้อย่างถูกต้อง

จุดประสงค์เชิงพฤติกรรม

ทฤษฎี

1. อธิบายการใช้งานของกลุ่มคำสั่งเกี่ยวกับ Digital I/O Pin ได้อย่างถูกต้อง
2. อธิบายการใช้งานของกลุ่มคำสั่งเกี่ยวกับ Analog I/O Pin ได้อย่างถูกต้อง
3. อธิบายหลักการคำนวณค่า ADC ได้อย่างถูกต้อง

ปฏิบัติ

1. เขียนโปรแกรมใช้งานของกลุ่มคำสั่งเกี่ยวกับ Digital I/O Pin ได้อย่างถูกต้อง
2. เขียนโปรแกรมใช้งานของกลุ่มคำสั่งเกี่ยวกับ Analog I/O Pin ได้อย่างถูกต้อง

3. คำนวณค่า ADC ได้อย่างถูกต้อง

กิจกรรมการเรียนการสอน

หัวข้อการสอน	เวลา (นาที)	วัตถุประสงค์เชิงพฤติกรรม (เพื่อให้นักศึกษาสามารถ)	วิธีการสอนและกิจกรรม
นำเข้าสู่บทเรียน	5		บรรยาย
ทฤษฎี :			
5.1 กลุ่มคำสั่งดิจิตอล อินพุต/เอาต์พุต 5.2 กลุ่มคำสั่งแอนาลอก อินพุต/เอาต์พุต	55	1. อธิบายการใช้งานของกลุ่มคำสั่งเกี่ยวกับ Digital I/O Pin ได้อย่างถูกต้อง 2. อธิบายการใช้งานของกลุ่มคำสั่งเกี่ยวกับ Analog I/O Pin ได้อย่างถูกต้อง 3. อธิบายหลักการคำนวณค่า ADC ได้อย่างถูกต้อง	บรรยายประกอบ power point
ปฏิบัติ :			
ใบงานที่ 6 การกำหนด Digital และ Analog Pin	165	1. เขียนโปรแกรม**ใช้งานของกลุ่มคำสั่งเกี่ยวกับ Digital I/O Pin ได้อย่างถูกต้อง** 2. เขียนโปรแกรม**ใช้งานของกลุ่มคำสั่งเกี่ยวกับ Analog I/O Pin ได้อย่างถูกต้อง**	บรรยาย สาธิต ทดลอง
สรุป	15		บรรยาย

สื่อการเรียนการสอน

1. ใบความรู้ เรื่อง การกำหนด Digital และ Analog Pin
2. Power Point เรื่อง การกำหนด Digital และ Analog Pin
3. ใบงานการทดลอง เรื่อง การกำหนด Digital และ Analog Pin
4. ของจริง โปรแกรม Arduino IDE, ชุดทดลองไมโครคอนโทรลเลอร์

การวัดผลประเมินผล

1. แบบประเมินผลการปฏิบัติงาน
2. แบบทดสอบหลังเรียน

ใบความรู้
หน่วยที่ 5 เรื่อง การกำหนด Digital และ Analog Pin

5.1 กลุ่มคำสั่งดิจิตอล อินพุต/เอาต์พุต

คำสั่งในกลุ่มนี้เป็นกลุ่มคำสั่งคำสั่งสำหรับใช้งาน Pin I/O ของ Arduino ในแบบของ digital I/O ซึ่งตามปรกติแล้วในการจะกำหนดหน้าที่ใช้งานขาสัญญาณ ของไมโครคอนโทรลเลอร์ AVR ซึ่งนิยมเรียกกันว่า Pin I/O นั้นเราจะต้องเข้าไปกำหนดค่าให้กับรีจิสเตอร์ต่างๆ ในตัว MCU โดยตรง เพื่อเลือกกำหนดรูปแบบของการทำงานของขาสัญญาณ PIN I/O ต่างๆ ให้มีคุณสมบัติตามที่เราต้องการ ซึ่งกรรมวิธี และขั้นตอนดังกล่าวข้างต้นจะมีความยุ่งยากซับซ้อนพอสมควร และ อาจดูเป็นเรื่องลำบากสำหรับผู้เล่นเริ่มใช้ไมโครคอนโทรลเลอร์ด้วยซ้ำไป แต่ใน Arduino นั้นได้มีการสร้างฟังก์ชั่น หรือคำสั่งสำหรับช่วยลดความยุ่งยากซับซ้อนตรงนี้ไว้ได้แล้ว ทำให้ผู้ใช้สามารถกำหนดหน้าที่ของการทำงาน และสั่งงาน Pin I/O ต่างๆ ของ MCU ได้โดยง่าย

สำหรับ digital I/O ใน Arduino นั้น จะมีทั้งหมดจำนวน 14 pin โดย Arduino ได้กำหนดรหัสของตัวเลขซึ่งเป็นเลขจำนวนเต็มค่าระหว่าง 0 ถึง 13 สำหรับอ้างถึง digital I/O ทั้ง 14 pin แต่ตามปรกติแล้วจะมี digital I/O pin จำนวน 2 pin ซึ่งถูกสงวนไว้สำหรับใช้เป็นพอร์ตสื่อสารอนุกรม rs232 สำหรับใช้ Upload Code ของ โปรแกรม ให้กับบอร์ด จึงเหลือ digital I/O สำหรับใช้งาน Digital I/O จริงๆจำนวน 12 pin คือ Digital I/O หมายเลข 2 ถึง 13 เท่านั้นโดยคำสั่งในกลุ่มนี้จะมีอยู่ด้วยกัน 3 คำสั่ง คือ

- void pinMode(pin,mode)

- void digitalWrite(pin,value)

- int digitalRead(pin)

pinMod(pin,Mode)

ใช้ทำหน้าที่สำหรับกำหนดหน้าที่การทำงานของขา I/O ที่เป็น digital I/O pin ของไมโครคอนโทรลเลอร์ AVR ตามที่ Arduino กำหนดไว้ ว่าต้องการกำหนดใช้งานขา digital I/O ขาใดเพื่อใช้งานเป็น INPUT หรือ OUTPUT

```
pinMode (pin,Mode);
```

pin หมายถึง หมายเลข รหัส pin ของขาสัญญาณที่ทำหน้าที่เป็น digital I/O pin ซึ่งจะมีทั้งหมดจำนวน 14 pin คือ 0 ถึง 13 โดยต้องการกำหนดรูปแบบของตัวเลข ให้เป็นแบบจำนวนเต็ม (int) ด้วย

Mode หมายถึง หน้าที่การทำงานของ digital I/O pin ที่ต้องการกำหนด ซึ่งสามารถกำหนดได้ 2 หน้าที่ โดยใช้รหัสข้อความ เป็น INPUT และ OUTPUT

digitalWrite (pin, value)

ใช้ทำหน้าที่สำหรับกำหนดสถานะทาง OUTPUT ให้กับ Digital output pin ว่าต้องการให้มีสภาวะทางโลจิกเป็น High หรือ LOW ซึ่งขาสัญญาณที่จะสั่งงานด้วยคำสั่งนี้ จะต้องถูกกำหนดคุณสมบัติให้ทำหน้าที่เป็น OUTPUT เรียบร้อยแล้ว

```
digitalWrite(pin, value);
```

Pin หมายถึง รหัส pin ของขาสัญญาณที่ทำหน้าที่เป็น Digital I/O pin ซึ่งจะมีทั้งหมด จำนวน 14 pin คือ 0 ถึง 13 โดยต้องกำหนดรูปแบบของตัวเลข ให้เป็นแบบจำนวนเต็ม (int) ด้วย

Value หมายถึง ค่าสถานะทาง OUTPUT ของ digital output pin ที่ต้องการกำหนดซึ่งสามารถกำหนดสถานะให้กับ pin ได้ 2 คำ คือ high และ low

digitalRead(pin)

ใช้ทำหน้าที่ อ่านสถานะ Logic input ของ digital input pin ว่ามีค่าสถานะเป็น high หรือ low ซึ่งขาสัญญาณที่จะสั่งอ่านด้วยคำสั่งนี้ จะต้องถูกกำหนดคุณสมบัติให้ทำหน้าที่เป็น input เรียบร้อยแล้ว

```
Var = digitalRead(pin)
```

Pin หมายถึง รหัส pin ของขาสัญญาณที่ทำหน้าที่เป็น digital input pin ซึ่งจะมีทั้งหมด จำนวน 14 pin คือ 0 ถึง 13 โดยต้องกำหนดรูปแบบของตัวเลข ให้เป็นจำนวนเต็ม (int) ด้วย

Var คือ ตัวแปรแบบ int สำหรับใช้รอรับคำสั่งที่ส่งคืนกลับมาจากฟังก์ชั่น ซึ่งเป็นค่าสถานะทางโลจิกของ digital input pin ซึ่งมีค่าเป็น HIGH หรือ LOW

5.2 กลุ่มคำสั่งแอนาลอก อินพุต/เอาต์พุต

คำสั่งในกลุ่มนี้เป็นกลุ่มคำสั่งสำหรับใช้งาน Pin I/O ของ Analog I/O ซึ่งตามปกติแล้ว Arduino จะมีขาสัญญาณที่ใช้ทำหน้าที่เป็น Analog input ได้ทั้งหมด 6 ถึง 8 pin โดยถ้าใช้กับ AVR รุ่นที่เป็นตัวถัง SMD แบบ 32 ขา ซึ่งขาสัญญาณในกลุ่มที่เป็น Analog input นี้ จะใช้รหัสตัวเลข 0 ถึง 7 ในการอ้างอิงขาสัญญาณ โดยที่คุณสมบัติของสัญญาณในกลุ่มนี้จะสามารถใช้วัด หรือ อ่านค่าสัญญาณที่เป็นแบบแรงดันไฟฟ้า ซึ่งในทางอิเล็กทรอนิกส์ เรียกสัญญาณแบบนี้ว่า สัญญาณ Analog โดยที่ Analog input สามารถรับแรงต้น input ที่มีขนาดแรงดันระหว่าง 0-5 โวลต์ แล้วแปลงค่าขนาดของแรงดันที่อ่านได้มาเป็นข้อมูลแบบตัวเลขจำนวนเต็ม ซึ่งมีค่าความละเอียด 1024 ค่า (0 ถึง 1023) ซึ่งความสามารถของ Arduino ในการอ่านและวัดค่าระดับของแรงดันที่เป็น Analog นี้จะช่วยทำให้เราสามารถนำไมโครคอนโทรลเลอร์ไปประยุกต์ใช้งาน เกี่ยวกับการตรวจจับ input แบบต่างๆ ได้มากมายหลายแบบโดยใช้งานร่วมกับอุปกรณ์ตรวจจับ (sensor) แบบต่างๆ ที่ให้ค่าออกมาเป็นแรงดัน

ส่วน Analog output นั้น ตามปกติแล้วจะต้องใช้วงจรแปลงสัญญาณแบบ Digital to Analog หรือ O/A ซึ่งวงจรแบบนี้ไม่มีบรรจุอยู่ในชิพของ AVR ด้วยเลย แต่อย่างไรก็ตาม Arduino ได้ดัดแปลงเอาความสมารถในการสร้างสัญญาณ PWM ของ AVR มาประยุกต์ดัดแปลงเพื่อใช้สร้างเป็นสัญญาณ Analog output ได้ด้วย โดยใช้เทคนิคการสร้างสัญญาณ PWM ขึ้นมา แล้วทำการปรับขนาดของ Duty Cycle หรือ ช่วงเวลาที่สัญญาณ PWM มีสถาวะลอจิกเป็น HIGH ให้มีค่ามากน้อยต่างๆ กันไปโดยใช้ค่าความถี่เท่าเดิม ซึ่งผลของการปรับค่าสัญญาณ PWM ด้วยการเปลี่ยนแปลงค่า Duty Cycle นี้จำให้ได้ค่าเฉลี่ยของขนาดแรงดันที่เปลี่ยนแปลงไปตามค่าการปรับของ Duty cycle ของสัญญาณ PWM ด้วยซึ่งผลลัพธ์ก็จะเปรียบเสมือนกับการปรับขนาดของสัญญาณ Analog output pin ได้จำนวน 6 pin คือ digital I/O เท่านั้นเอง

ดังนั้นขาสัญญาณที่เป็น Analog output ของ Arduino ในที่นี้ก็คือ ขาสัญญาณที่ทำหน้าที่เป็น Digital Output pin เฉพาะ pin ที่สามารถสร้างสัญญาณ PWM ได้เท่านั้น ไม่สามารถใช้ได้กับ

ทุก pin โดยถ้าเป็น Arduino ที่ใช้ไมโครคอนโทรลเลอร์ AVR รุ่น Mega8 จะสามารถใช้งาน pin สำหรับเป็น Analog output pin ได้จำนวน 3 pin คือ Digital I/O pin หมายเลข 9,10 และ 11 เท่านั้น แต่ถ้าเป็น Arduino ที่ใช้ AVR รุ่น mega88 และ mega168 จะสามารถใช้งาน pin สำหรับเป็น Analog Output pin ได้จำนวน 6 pin คือ Digital I/O Pin หมายเลข 3,5,6,9,10 และ 11 เท่านั้น โดยคำสั่งของ Analog I/O ของ Arduino จะมีอยู่ 3 คำสั่งคือ

- **analogReference(type)**
- **int analogRead(pin)**
- **void analogWrite(Pin,value)**

analogReference(type)

หน้าที่การทำงานของคำสั่ง

คำสั่งนี้ ใช้ทำหน้าที่สำหรับเลือกชนิดของวงจร Analog to Digital หรือ A/O เพื่อใช้งานเป็น Analog output (analogRead) ซึ่งค่าผลลัพธ์ของ Analog input จะมีความสัมพันธ์โดยตรงกันแรงดันอ้างอิง ของ Analog input โดยจะอ่านค่าผลลัพธ์ของ Analog input ได้ค่าเป็นสูงสุด 1023 เมื่อแรงดัน input มีค่าเท่ากับแรงดันอ้างอิงที่กำหนดให้

รูปแบบของคำสั่ง

```
analogReference(type)
```

type หมายถึง ชนิดของค่าแรงดันอ้างอิงที่ต้องการกำหนดให้กับ MCU ซึ่ง Arduino สามารถเลือกกำหนดแรงดันอ้างอิงให้กับ Analog input ได้ 3 แบบด้วยกันคือ

DEFAULT มีค่าแรงดันอ้างอิงเท่ากับ +5v

INTERNAL มีค่าแรงดันอ้างอิง 1.1v สำหรับ ATmega88 หรือ ATmega168 และมีค่าเป็น 2.5v สำหรับ ATmega8

EXTERNAL มีค่าแรงดันอ้างอิงตามขนาดแรงดันที่ป้อนให้กับขา AREF จากภายนอก

Int analogRead(pin)

คำสั่งนี้ ใช้ทำหน้าที่สำหรับอ่านค่าระดับแรงดันที่ขาสัญญาณของ Analog input Pin ว่าระดับแรงดัน (Analog Input) มีค่าระดับแรงดันเป็นเท่าใด (0 - Analog Reference) ซึ่งค่าผลลัพธ์ที่อ่านได้จากฟังก์ชั่น จะเป็นระดับของแรงดันที่อ่านได้ที่ขา Analog input pin ซึ่งทำการแปลงค่ามาเป็นข้อมูลแบบดิจิตอล ขนาด 10 บิต เรียบร้อยแล้ว โดยผลลัพธ์ที่ได้จะมีค่า 1024 ระดับ โดยผลลัพธ์จะมีค่าระหว่าง 0 ถึง 1023 ซึ่งค่า 1 ระดับของข้อมูลที่อ่านได้จะมีขนาดเท่ากับขนาดของแรงดันอ้างอิง /1024 ซึ่งถ้าใช้ค่าแรงดันอ้างอิง หรือ Analog Reference ตามค่า DEFAULT ซึ่งมีค่าเป็น +5v ก็จะได้ขนาดของแรงดันต่อขนาดของข้อมูล 1 ระดับ ประมาณเท่ากับ 5v/1024 หรือประมาณ 0.0049V (4.9mV)

โดยการทำงานของคำสั่งนี้จะใช้เวลาในการทำงานประมาณ 100uS (100 ไมโครวินาที) ดังนั้นเราจะสามารถสั่งอ่านค่า Analog Input ได้เร็วสุดประมาณ 10,000 รอบต่อวินาที

ความสัมพันธ์ของค่า ADC กับแรงดันหาได้จาก

$$\text{ค่า ADC ที่อ่านได้} = \frac{1024}{5} \times \text{แรงดันแอนาล็อก}$$

รูปแบบของคำสั่ง

```
Var = analogRead(pin)
```

Pin หมายถึง หมายเลขรหัส pin ของขาสัญญาณที่ทำหน้าที่เป็น Analog input pin ซึ่งจะมีทั้งหมดจำนวน 6 pin ถ้าใช้ชิพรุ่นที่มีตัวถัง 28 pin และมี 8 pin ถ้าใช้กับชิพรุ่นที่มี 32 ขา โดยมีรหัสตัวเลข 0 ถึง 5 และ 0 ถึง 7 โดยต้องกำหนดรูปแบบของตัวเลข ให้เป็นจำนวนเต็ม (int) ด้วย

ค่าที่คืนกลับจากฟังก์ชั่น

Var คือตัวแปรแบบ int สำหรับใช้รอรับค่าที่ส่งคืนกลับมาจากฟังก์ชั่น ซึ่งเป็นค่าแรงดัน Analog input ที่ถูกแปลงเป็นค่าข้อมูล 10 บิตแล้ว (1024 ค่า) โดยค่าผลลัพธ์ที่เป็นไปได้คือ 0 – 1023 ซึ่งในการใช้งานเราจะต้องสร้างตัวแปรแบบ int ขึ้นมาเพื่อใช้รอรับค่าที่จะส่งกลับมาจากฟังก์ชั่นด้วย

Void analogWrite(pin, value)

คำสั่งนี้ใช้หน้าที่สำหรับกำหนดค่าระดับของสัญญาณ analog output ให้กับ digital output pin บาง pin เพื่อสร้างสัญญาณแรงดันให้มีระดับแรงดันตามที่กำหนดจากคำสั่ง

รูปแบบของคำสั่ง

```
analogWrite(pin,value)
```

Pin หมายถึง หมายเลข pin ของขาสัญญาณที่ทำหน้าที่เป็น digital output pin เฉพาะ pin ที่สามารถสร้างสัญญาณ PWM ได้เท่านั้น ไม่สามารถใช้ได้กับทุก pin โดยถ้าเป็น Arduino ที่ใช้ไมโครคอนโทรลเลอร์ AVR รุ่น mega6 จะสามารถใช้งาน pin สำหรับเป็น analog output pinได้จำนวน 3 pin คือ digital1/0 pin หมายเลข 9,10และ 11เท่านั้นแต่ถ้าเป็น Arduino ที่ใช้ AVR รุ่น mega 168 จะสามารถใช้งาน pin สำหรับเป็น analog output pin ได้จำนวน 6 pin คือ digital1/0 pin หมายเลข 3,5,6,9,10 และ 11 เท่านั้น

Value หมายถึง ค่าระดับสัญญาณ analog output ที่ต้องการ มีค่าความละเอียด 8 บิต หรือ 256 ระดับโดยมีค่าระหว่าง 0-255 ซึ่งมีค่าศูนย์จะเป็นคำต่ำสุดและ 255 เป็นค่าสูงสุด

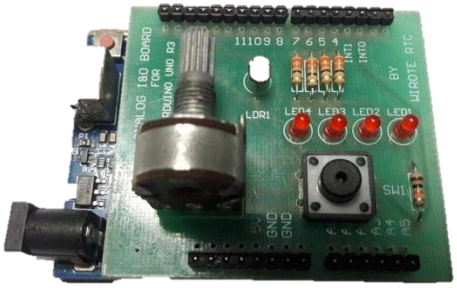
ใบงานที่ 6
เรื่อง การกำหนด Digital และ Analog Pin

เครื่องมือและอุปกรณ์

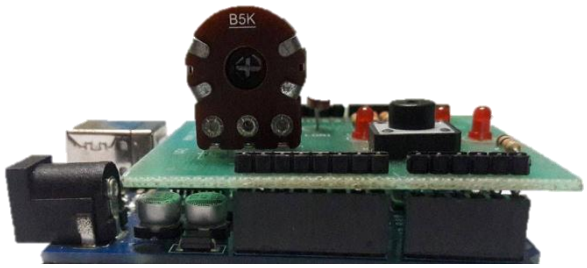
1. บอร์ด Arduino Uno R3 จำนวน 1 ชุด
2. คอมพิวเตอร์ จำนวน 1 เครื่อง
3. Shield Analog I&O จำนวน 1 ชุด
4. สาย USB จำนวน 1 เส้น

ขั้นตอนการทดลอง

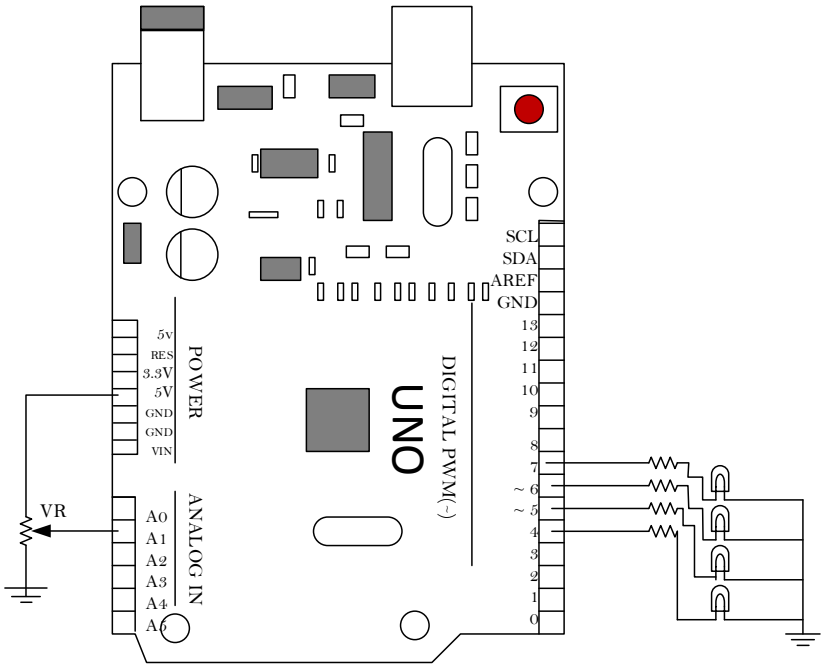
1. ต่อ Shield Analog I&O เข้ากับบอร์ด Arduino UNO R3 ดังรูปที่ 5.1



ก



ข



ค

รูปที่ 5.1 การต่อ Shield Analog I&O เข้ากับบอร์ด Arduino

2. เขียนโปรแกรมดังต่อไปนี้ จากนั้น Verify และ Upload ให้เรียบร้อย แล้วทำการบันทึกผลการทดลองที่สังเกตได้

การทดลองที่ 1 โปรแกรม Test_Analog1.ino

```
int analogPin = A1;
int val=0;
void setup()
{
  Serial.begin(9600);
  pinMode(analogPin,INPUT);
}
void loop()
{
  val = analogRead(analogPin);
  Serial.print("Analog input = ");
  Serial.println(val);
}
```

บันทึกผลการทดลองเมื่อหมุนตัวตานทานปรับค่าได้

.....

.....

.....

การทดลองที่ 2 โปรแกรม Test_Analog2.ino

```
int analogPin = A1;
int led1 = 5;
int val=0;
void setup() {
  Serial.begin(9600);
  pinMode(analogPin,INPUT);
  pinMode(led1,OUTPUT);
}
void loop()
{
  val = analogRead(analogPin);
  Serial.print("Analog input = ");
  Serial.println(val);
```

```
  if(val>512)
  digitalWrite(led1,HIGH);
  else
  digitalWrite(led1,LOW);
}
```

บันทึกผลการทดลองเมื่อหมุนตัวตานทานปรับค่าได้

.....

.....

.....

.....

การทดลองที่ 3 โปรแกรม Test_Analog3.ino

```
int analogPin = A1;
int led1 = 4;
int led2 = 5;
int led3 = 6;
int led4 = 7;
int val=0;
float Vin;
void setup()
{
  Serial.begin(9600);
  pinMode(analogPin,INPUT);
  pinMode(led1,OUTPUT);
  pinMode(led2,OUTPUT);
  pinMode(led3,OUTPUT);
  pinMode(led4,OUTPUT);
}
void loop()
{
  val = analogRead(analogPin);
  Serial.print("Analog input = ");
  Serial.println(val);
  Vin = (0.0048875*val);
  Serial.print("Vin = ");
  Serial.println(Vin,2);
```

```
  if((val>205)&&(val<410))
  digitalWrite(led1,HIGH);
  else digitalWrite(led1,LOW);
  if((val>410)&&(val<614))
  digitalWrite(led2,HIGH);
  else digitalWrite(led2,LOW);
  if((val>614)&&(val<819))
  digitalWrite(led3,HIGH);
  else digitalWrite(led3,LOW);
  if((val>819)&&(val<=1023))
  digitalWrite(led4,HIGH);
  else digitalWrite(led4,LOW);
}
```

บันทึกผลการทดลองเมื่อหมุนตัวตานทานปรับค่าได้

.....

.....

.....

.....

การทดลองที่ 4 โปรแกรม Test_Analog4.ino

```
int analogPin = A1;
int led1 = 4;
int led2 = 5;
int led3 = 6;
int led4 = 7;
int val=0;
float Vin;
void setup()
{
  Serial.begin(9600);
  pinMode(analogPin,INPUT);
  pinMode(led1,OUTPUT);
  pinMode(led2,OUTPUT);
  pinMode(led3,OUTPUT);
  pinMode(led4,OUTPUT);
}
```

```
void loop()
{
  val = analogRead(analogPin);
  Serial.print("Analog input = ");
  Serial.println(val);
  Vin = (0.0048875*val);
  Serial.print("Vin = ");
  Serial.println(Vin,2);
  if((val>205)&&(val<410))
  digitalWrite(led1,HIGH);
  else digitalWrite(led1,LOW);
  if((val>410)&&(val<614))
  {digitalWrite(led1,HIGH);
   digitalWrite(led2,HIGH);}
  else digitalWrite(led2,LOW);
  if((val>614)&&(val<819))
  {digitalWrite(led1,HIGH);
   digitalWrite(led2,HIGH);
   digitalWrite(led3,HIGH);}
  else digitalWrite(led3,LOW);
  if((val>819)&&(val<=1023))
  {digitalWrite(led1,HIGH);
   digitalWrite(led2,HIGH);
   digitalWrite(led3,HIGH);
   digitalWrite(led4,HIGH);}
  else digitalWrite(led4,LOW);
}
```

บันทึกผลการทดลองเมื่อหมุนตัวตานทานปรับค่าได้

.....

.....

.....

.....

การทดลองที่ 5 โปรแกรม Test_Analog5.ino

```
int led1 = 5;
void setup()
{ pinMode(led1,OUTPUT);  }
void loop()
{ for(int fade=0;fade<=255;fade+=5)  {
   analogWrite(led1,fade);
   delay(40);  }
  for(int fade=255;fade>=0;fade-=5)
  {
   analogWrite(led1,fade);
   delay(40);
  }
}
```

บันทึกผลการทดลองที่ได้

.....

.....

.....

.....

การทดลองที่ 6 โปรแกรม Test_Analog6.ino

```
int led1 = 5;
int led2 = 6;
int analogPin = A1;
int InputValue =0;
int OutputValue =0;
void setup()
{
  Serial.begin(9600);
  pinMode(analogPin,INPUT);
  pinMode(led1,OUTPUT);
  pinMode(led2,OUTPUT);
}
void loop()
{
   InputValue = analogRead(analogPin);
```

```
  OutputValue = map( InputValue,0,1023,0,255);
  analogWrite(led1,  OutputValue);
  analogWrite(led2,  OutputValue);
  Serial.print("Input = ");
  Serial.println(InputValue);
  Serial.print("Output = ");
  Serial.println(OutputValue);
 }
```

บันทึกผลการทดลองเมื่อหมุนตัวตานทานปรับค่าได้

.....

.....

.....

.....

คำถามท้ายการทดลอง

1. จงเขียนโปรแกรมใน LED มีความเร็วในการกระพริบช้าเร็วตามการปรับค่าตัวต้านทาน

แบบประเมินผลการปฏิบัติงาน
หน่วยที่ 5 ชื่อหน่วย การกำหนด Digital และ Analog Pin

ชื่อ – สกุล .....รหัสผู้เข้าเรียน.....
ระดับชั้น ปวช.3 ห้อง..... แผนกวิชาช่างอิเล็กทรอนิกส์
วัน/เดือน/ปี.....
เริ่มปฏิบัติงานเวลา.....น. เสร็จเวลา.....น.
รวมปฏิบัติงาน.....ชั่วโมง.....นาที

คะแนนเต็มรวม 10 คะแนน

ลำดับที่	หัวข้อการประเมิน	คะแนนเต็ม	ช่วงระดับคะแนน					ตัวคูณ	คะแนนจริงที่ได้
			ดีมาก	ดี	ปานกลาง	น้อย	น้อยมาก		
			5	4	3	2	1		
1	การเลือกใช้เครื่องมือ วัสดุและอุปกรณ์ในการปฏิบัติงาน	2						0.4	
2	ขั้นตอนการปฏิบัติงาน	2						0.4	
3	จำนวนใบงานที่ทำได้ถูกต้อง	3						0.4	
4	เวลาที่ใช้ในการปฏิบัติงาน	3						0.4	
	รวม	10	รวม						

ลงชื่อ.....ผู้ประเมิน
(.....)

เกณฑ์การให้คะแนนปฏิบัติงาน

คำชี้แจง เกณฑ์การให้คะแนนการปฏิบัติงานนี้ ให้ครูผู้ควบคุมเป็นผู้ประเมินผลตามเกณฑ์การให้คะแนนเท่านั้น

1. **เกณฑ์การให้คะแนน การเตรียมวัสดุและอุปกรณ์ในการปฏิบัติงาน**

หัวข้อประเมิน

1. เลือกใช้วัสดุในการปฏิบัติงานได้อย่างถูกต้อง
2. เลือกใช้อุปกรณ์ในการปฏิบัติงานได้อย่างถูกต้อง
3. เลือกใช้วัสดุและอุปกรณ์ในการปฏิบัติงานได้อย่างถูกวิธี
4. ทำความสะอาดห้องเรียน ห้องปฏิบัติงานเป็นระเบียบเรียบร้อย
5. จัดวางวัสดุและอุปกรณ์ในการปฏิบัติงานได้อย่างเป็นระเบียบเรียบร้อย

ระดับคะแนน

ดีมาก	หมายถึง	มีการเลือกใช้วัสดุ และอุปกรณ์ในการปฏิบัติงานครบทุกหัวข้อและการประเมิน
ดี	หมายถึง	มีการเลือกใช้วัสดุ อุปกรณ์ในการปฏิบัติงาน มีหัวข้อการประเมินได้ 4 หัวข้อ
ปานกลาง	หมายถึง	มีการเลือกใช้วัสดุ อุปกรณ์ในการปฏิบัติงาน มีหัวข้อการประเมินได้ 3 หัวข้อ
น้อย	หมายถึง	มีการเลือกใช้วัสดุ อุปกรณ์ในการปฏิบัติงาน มีหัวข้อการประเมินได้ 2 หัวข้อ
น้อยมาก	หมายถึง	มีการเลือกใช้วัสดุ อุปกรณ์ในการปฏิบัติงาน มีหัวข้อการประเมินได้ 1 หัวข้อ หรือ ปฏิบัติงานไม่ได้เลยตาม หัวข้อการประเมิน

2.ขั้นตอนการปฏิบัติงาน

2.1 เกณฑ์การให้คะแนน การเขียนโปรแกรม

หัวข้อการประเมิน

1. เตรียมวัสดุและอุปกรณ์ในการเขียนโปรแกรม
2. การใช้วัสดุและอุปกรณ์ในการเขียนโปรแกรม
3. การใช้งานโปรม Arduino IDE
4. การเชื่อมต่อชุดฝึกกับคอมพิวเตอร์

ระดับคะแนน

ดีมาก	หมายถึง	ปฏิบัติงานได้สำเร็จ รวดเร็ว และถูกต้องครบถ้วนทุกหัวข้อการประเมิน ตามขั้นตอนในใบงานและขั้นตอนการปฏิบัติงาน
ดี	หมายถึง	ปฏิบัติงานได้ตามหัวข้อประเมิน มีความถูกต้องตามขั้นตอนในใบงานและขั้นตอนการปฏิบัติงานได้ 4 หัวข้อการประเมิน
ปานกลาง	หมายถึง	ปฏิบัติงานได้ตามหัวข้อประเมิน มีความถูกต้องตามขั้นตอนในใบงานและขั้นตอนการปฏิบัติงานได้ 3 หัวข้อการประเมิน
น้อย	หมายถึง	ปฏิบัติงานได้ตามหัวข้อประเมิน มีความถูกต้องตามขั้นตอนในใบงานและขั้นตอนการปฏิบัติงานได้ 2 หัวข้อการประเมิน
น้อยมาก	หมายถึง	ปฏิบัติงานได้ตามหัวข้อประเมิน มีความถูกต้องตามขั้นตอนในใบงานและขั้นตอนการปฏิบัติงานได้น้อยกว่า 2 หัวข้อการประเมิน

3.เกณฑ์การให้คะแนน จำนวนใบงานที่ทำได้ถูกต้อง

ระดับคะแนน		
ดีมาก	หมายถึง	ปฏิบัติงานได้สำเร็จ รวดเร็ว ถูกต้องครบถ้วนทุกใบงาน ประเมิน เป็นไปตามขั้นตอนการปฏิบัติงานของใบงาน
ดี	หมายถึง	ปฏิบัติงานได้สำเร็จ รวดเร็ว ถูกต้อง 5 ใบงาน ประเมิน เป็นไปตามขั้นตอนการปฏิบัติงานของใบงาน
ปานกลาง	หมายถึง	ปฏิบัติงานได้สำเร็จ รวดเร็ว ถูกต้อง 4 ใบงาน ประเมิน เป็นไปตามขั้นตอนการปฏิบัติงานของใบงาน
น้อย	หมายถึง	ปฏิบัติงานได้สำเร็จ รวดเร็ว ถูกต้อง 3 ใบงาน ประเมิน เป็นไปตามขั้นตอนการปฏิบัติงานของใบงาน
น้อยมาก	หมายถึง	ปฏิบัติงานได้น้อยกว่า 3 ใบงาน ประเมิน เป็นไปตามขั้นตอนการปฏิบัติงานของใบงาน

4.เกณฑ์การให้คะแนน เวลาที่ใช้ในการปฏิบัติงาน

ระดับคะแนน

ดีมาก หมายถึงปฏิบัติงานด้วยความรวดเร็ว ถูกต้อง แม่นยำ ทันตามเวลาที่กำหนด

ดี หมายถึงการปฏิบัติงานไม่ทันตามเวลาที่กำหนดไว้แต่ไม่เกิน 5 นาที

ปานกลาง หมายถึงการปฏิบัติงานไม่ทันตามเวลาที่กำหนดไว้แต่ไม่เกิน 10 นาที

น้อย หมายถึงการปฏิบัติงานไม่ทันตามเวลาที่กำหนดไว้แต่ไม่เกิน 20 นาที

น้อยมาก หมายถึงการปฏิบัติงานไม่ทันตามเวลาที่กำหนดไว้เกินกว่า 20 นาที

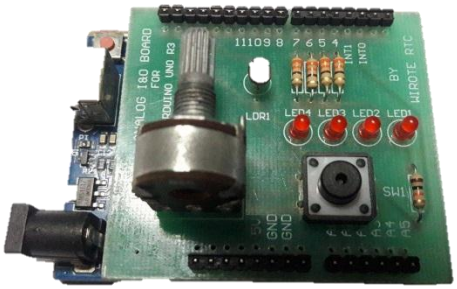
เฉลยใบงานที่ 6
เรื่อง การกำหนด Digital และ Analog Pin

เครื่องมือและอุปกรณ์

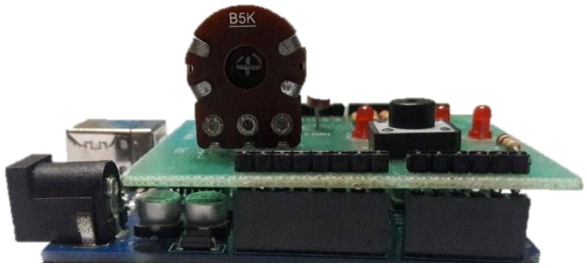
1. บอร์ด Arduino Uno R3	จำนวน 1 ชุด
2. คอมพิวเตอร์	จำนวน 1 เครื่อง
3. Shield Analog I/O	จำนวน 1 ชุด
4. สาย USB	จำนวน 1 เส้น

ขั้นตอนการทดลอง

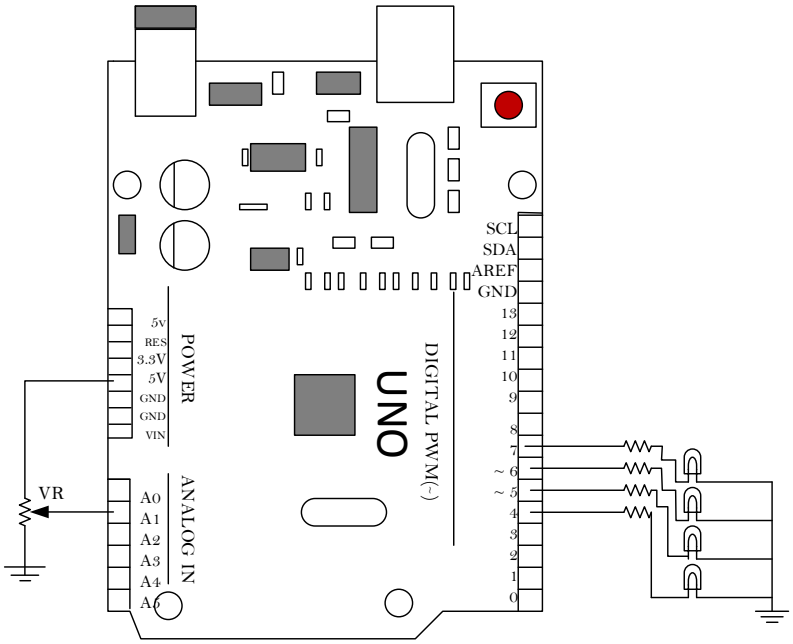
1. ต่อ Shield Analog I/O เข้ากับบอร์ด Arduino UNO R3 ดังรูปที่ 5.1



ก



ข



ค

รูปที่ 5.1 การต่อ Shield Analog I/O เข้ากับบอร์ด Arduino

2. เขียนโปรแกรมดังต่อไปนี้ จากนั้น Verify และ Upload ให้เรียบร้อย แล้วทำการบันทึกผลการทดลองที่สังเกตได้

การทดลองที่ 1 โปรแกรม Test_Analog1.ino

```
int analogPin = A1;
int val=0;
void setup()
{
  Serial.begin(9600);
  pinMode(analogPin,INPUT);
}
void loop()
{
  val = analogRead(analogPin);
  Serial.print("Analog input = ");
  Serial.println(val);
}
```

บันทึกผลการทดลองเมื่อหมุนตัวตานทานปรับค่าได้

จอ Serial Monitor จะแสดงค่า Analog input = 0 เมื่อหมุนไปทางซ้ายสุดและจะเพิ่มขึ้นเรื่อยๆ เมื่อหมุนไปทางขวา สูงสุดจะมีค่า Analog input = 1023

การทดลองที่ 2 โปรแกรม Test_Analog2.ino

```
int analogPin = A1;
int led1 = 5;
int val=0;
void setup()
{
  Serial.begin(9600);
  pinMode(analogPin,INPUT);
  pinMode(led1,OUTPUT);
}
void loop()
{
  val = analogRead(analogPin);
  Serial.print("Analog input = ");
  Serial.println(val);
  if(val>512)
  digitalWrite(led1,HIGH);
  else
  digitalWrite(led1,LOW);
}
```

บันทึกผลการทดลองเมื่อหมุนตัวตานทานปรับค่าได้

จอ Serial Monitor จะแสดงค่า Analog input = 0 เมื่อหมุนไปทางซ้ายสุดและจะเพิ่มขึ้นเรื่อยๆ เมื่อหมุนไปทางขวา สูงสุดจะมีค่า Analog input = 1023 เมื่อ Analog input มีค่ามากกว่า 512 LED จะติด

การทดลองที่ 3 โปรแกรม Test_Analog3.ino

```
int analogPin = A1;
int led1 = 4;
int led2 = 5;
int led3 = 6;
int led4 = 7;
int val=0;
float Vin;
void setup()
{
  Serial.begin(9600);
  pinMode(analogPin,INPUT);
  pinMode(led1,OUTPUT);
  pinMode(led2,OUTPUT);
  pinMode(led3,OUTPUT);
  pinMode(led4,OUTPUT);
}

void loop()
{
  val = analogRead(analogPin);
  Serial.print("Analog input = ");
  Serial.println(val);
  Vin = (0.0048875*val);
  Serial.print("Vin = ");
  Serial.println(Vin,2);
  if((val>205)&&(val<410))
  digitalWrite(led1,HIGH);
  else digitalWrite(led1,LOW);
  if((val>410)&&(val<614))
  digitalWrite(led2,HIGH);
  else digitalWrite(led2,LOW);
  if((val>614)&&(val<819))
  digitalWrite(led3,HIGH);
  else digitalWrite(led3,LOW);
```

```
  if((val>819)&&(val<=1023))
  digitalWrite(led4,HIGH);
  else digitalWrite(led4,LOW);
}
```

บันทึกผลการทดลองเมื่อหมุนตัวตานทานปรับค่าได้

เมื่อ Analog input มีค่ามากกว่า 205 แต่น้อยกว่า 410 LED1 จะติด เมื่อ Analog input มีค่ามากกว่า 410 แต่น้อยกว่า 614 LED2 จะติด เมื่อ Analog input มีค่ามากกว่า 614 แต่น้อยกว่า 819 LED3 จะติด เมื่อ Analog input มีค่ามากกว่า 819 แต่น้อยกว่า 1023 LED4 จะติด

การทดลองที่ 4 โปรแกรม Test_Analog4.ino

```
int analogPin = A1;
int led1 = 4;
int led2 = 5;
int led3 = 6;
int led4 = 7;
int val=0;
float Vin;
void setup()
{
  Serial.begin(9600);
  pinMode(analogPin,INPUT);
  pinMode(led1,OUTPUT);
  pinMode(led2,OUTPUT);
  pinMode(led3,OUTPUT);
  pinMode(led4,OUTPUT);
}
void loop()
{
  val = analogRead(analogPin);
  Serial.print("Analog input = ");
  Serial.println(val);
  Vin = (0.0048875*val);
  Serial.print("Vin = ");
  Serial.println(Vin,2);
  if((val>205)&&(val<410))
  digitalWrite(led1,HIGH);
  else digitalWrite(led1,LOW);
  if((val>410)&&(val<614))
  {digitalWrite(led1,HIGH);
   digitalWrite(led2,HIGH);}
  else digitalWrite(led2,LOW);
  if((val>614)&&(val<819))
  {digitalWrite(led1,HIGH);
   digitalWrite(led2,HIGH);
   digitalWrite(led3,HIGH);}
```

```
  else digitalWrite(led3,LOW);
  if((val>819)&&(val<=1023))
  {digitalWrite(led1,HIGH);
   digitalWrite(led2,HIGH);
   digitalWrite(led3,HIGH);
   digitalWrite(led4,HIGH);}
  else digitalWrite(led4,LOW);
}
```

บันทึกผลการทดลองเมื่อหมุนตัวตานทานปรับค่าได้

เมื่อ Analog input มีค่ามากกว่า 205 แต่น้อยกว่า 410 LED1 จะติด, เมื่อ Analog input มีค่ามากกว่า 410 แต่น้อยกว่า 614 LED1 LED2 จะติด, เมื่อ Analog input มีค่ามากกว่า 614 แต่น้อยกว่า 819 LED1 LED2 LED3 จะติด, เมื่อ Analog input มีค่ามากกว่า 819 แต่น้อยกว่า 1023 LED1 LED2 LED3 LED4 จะติด

การทดลองที่ 5 โปรแกรม Test_Analog5.ino

```
int led1 = 5;
void setup()
{ pinMode(led1,OUTPUT);  }
void loop()
{ for(int fade=0;fade<=255;fade+=5)
  {
   analogWrite(led1,fade);
   delay(40);
  }
  for(int fade=255;fade>=0;fade-=5)
  {
   analogWrite(led1,fade);
   delay(40);
  }
}
```

บันทึกผลการทดลองที่ได้

LED1 จะค่อยๆสว่างเพิ่มมากขึ้นและลดลงสว่างลด ซ้ำๆ

การทดลองที่ 6 โปรแกรม Test_Analog6.ino

```
int led1 = 5;
int led2 = 6;
int analogPin = A1;
int InputValue =0;
int OutputValue =0;
void setup()
{
  Serial.begin(9600);
  pinMode(analogPin,INPUT);
  pinMode(led1,OUTPUT);
  pinMode(led2,OUTPUT);
}
void loop()
{
   InputValue = analogRead(analogPin);
   OutputValue = map( InputValue,0,1023,0,255);
   analogWrite(led1,  OutputValue);
   analogWrite(led2,  OutputValue);
   Serial.print("Input = ");
   Serial.println(InputValue);
   Serial.print("Output = ");
   Serial.println(OutputValue);
  }
```

บันทึกผลการทดลองเมื่อหมุนตัวตานทานปรับค่าได้

LED1 จะค่อยๆสว่างเพิ่มขึ้นและลดลงตามการปรับค่าความต้านทาน ถ้าหมุนความต้านทานไปทางซ้ายสุด Analog input = 0 ไฟจะสว่างน้อย ถ้าหมุนความต้านทานไปทางขวาสุด Analog input = 1023 ไฟจะสว่างมาก

คำถามท้ายการทดลอง

1. จงเขียนโปรแกรมใน LED มีความเร็วในการกระพริบช้าเร็วตามการปรับค่าตัวต้านทาน

```
int led1 = 4;
int led2 = 5;
int led3 = 6;
int led4 = 7;
int analogPin = A1;
int InputValue =0;
int OutputValue =0;
void setup()
{
  Serial.begin(9600);
  pinMode(analogPin,INPUT);
  pinMode(led1,OUTPUT);
  pinMode(led2,OUTPUT);
  pinMode(led3,OUTPUT);
  pinMode(led4,OUTPUT);
}
void loop()
{
   InputValue = analogRead(analogPin);
   digitalWrite(led1,HIGH);
   digitalWrite(led2,HIGH);
   digitalWrite(led3,HIGH);
   digitalWrite(led4,HIGH);
   delay(InputValue);
   digitalWrite(led1,LOW);
   digitalWrite(led2,LOW);
   digitalWrite(led3,LOW);
   digitalWrite(led4,LOW);
   delay(InputValue);}
```

ตารางที่ 1 การวิเคราะห์หัวข้อหลัก

รหัสวิชา 2105 – 2105 ชื่อวิชา ไมโครคอนโทรลเลอร์

ทฤษฎี 1 ชั่วโมง/สัปดาห์ ปฏิบัติ 3 ชั่วโมง/สัปดาห์ จำนวน 2 หน่วยกิต

หน่วยที่	หัวข้อหลัก (ชื่อหน่วย)	แหล่งข้อมูล				
		A	B	C	D	E
1	โครงสร้างและส่วนประกอบของไมโครคอนโทรลเลอร์	✓	✓			
2	เครื่องมือสำหรับพัฒนาไมโครคอนโทรลเลอร์	✓	✓			
3	การเขียนโปรแกรมควบคุมไมโครคอนโทรลเลอร์สำหรับ Arduino	✓	✓			
4	การเขียนโปรแกรมภาษาซีสำหรับ Arduino (1) – (3)	✓	✓			
5	การกำหนด Digital และ Analog Pin	✓	✓	✓		
6	การควบคุมหลอดไฟแสดงผล LED	✓	✓	✓		
7	การควบคุมตัวแสดงผล LED 7 ส่วน	✓	✓	✓		
8	การควบคุมหน้าจอแสดงผล LCD	✓	✓	✓		✓
9	การเชื่อมต่อ Arduino กับ สวิทช์	✓	✓	✓		
10	การเชื่อมต่อ Arduino กับ Relay	✓	✓	✓		
11	การอินเตอร์รัพต์ของไมโครคอนโทรลเลอร์	✓	✓			
12	การใช้งานเซนเซอร์ LDR	✓	✓		✓	
13	การใช้งานเซนเซอร์วัดอุณหภูมิ	✓	✓		✓	✓
14	การควบคุมมอเตอร์ไฟฟ้ากระแสตรง	✓	✓			

แหล่งข้อมูล (Sources)

A : หลักสูตรรายวิชา (Course Description)
B : ตำราและเอกสาร (Literatures)
C : ประสบการณ์ (Experiences)
D : ผู้เชี่ยวชาญ (Experts)
E : อื่นๆ (Other)

ตารางที่ 2 การวิเคราะห์หัวข้อย่อย

รหัสวิชา 2105 – 2105 ชื่อวิชา ไมโครคอนโทรลเลอร์

ทฤษฎี 1 ชั่วโมง/สัปดาห์ ปฏิบัติ 3 ชั่วโมง/สัปดาห์ จำนวน 2 หน่วยกิต

หัวข้อหลัก หน่วยที่ 5 ชื่อหน่วย การกำหนด Digital และ Analog Pin

ลำดับที่	หัวข้อหลัก (ชื่อหน่วย)	แหล่งข้อมูล				
		A	B	C	D	E
1	5.การกำหนด Digital และ Analog Pin					
	5.1 กลุ่มคำสั่งดิจิตอล อินพุต/เอาต์พุต	✓	✓	✓		✓
	5.2 กลุ่มคำสั่งแอนาลอก อินพุต/เอาต์พุต	✓	✓	✓		✓
แหล่งข้อมูล (Sources)		**A : หลักสูตรรายวิชา (Course Description)** **B : ตำราและเอกสาร (Literatures)** **C : ประสบการณ์ (Experiences)** **D : ผู้เชี่ยวชาญ (Experts)** **E : อื่นๆ (Other)**				