วิชา อินเทอร์เน็ตของทุกสรรพสิ่ง
(Internet of Things : 4123412)

ภาคการศึกษา 1/2568

ผู้ช่วยศาตราจารย์ ดร. นัฐพงศ์ ส่งเนียม
Asst. Prof. Dr. Nattapong Songneam

สาขาวิชาวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์และเทคโนโลยี มรภ.พระนคร

Program in Computer Science , Faculty of Science and Technology
Phranakhon University

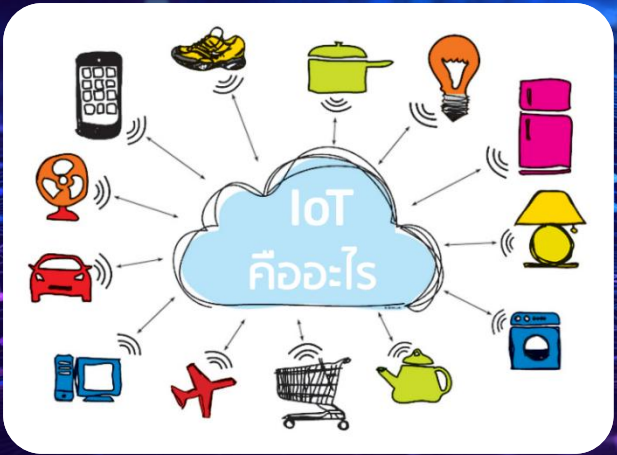
วิชา อินเทอร์เน็ตของทุกสรรพสิ่ง (Internet of Things : 4123412)

บทที่ 2

ความรู้พื้นฐานเกี่ยวกับอุปกรณ์ IoT และการใช้งานโปรแกรม Arduino IDE

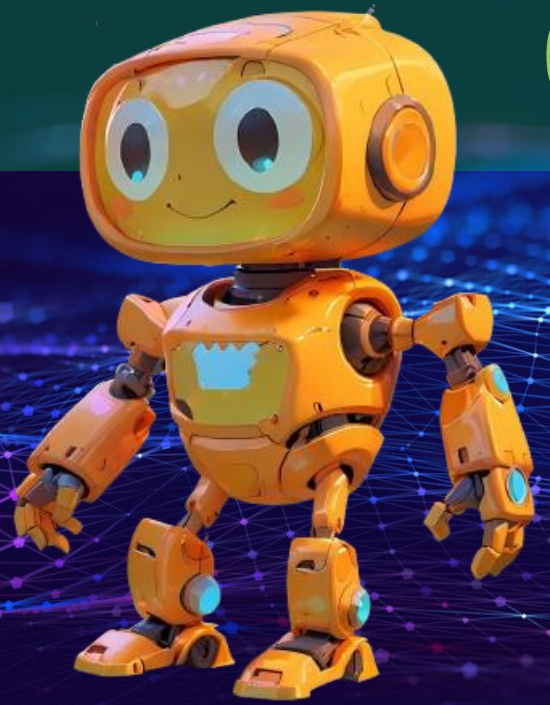
บทที่ 2 ความรู้พื้นฐานเกี่ยวกับอุปกรณ์ IoT และการใช้งานโปรแกรม Arduino IDE

หัวข้อ



1. แนะนำอุปกรณ์พื้นฐานเกี่ยวกับ Internet of Things
2. การเขียนโปรแกรมภาษา C ของ Arduino
3. การใช้งาน Arduino คำสั่งตัดสินใจเลือก if else
4. การใช้งาน Arduino คำสั่งทำซ้ำ
5. ตัวอย่างโปรแกรม

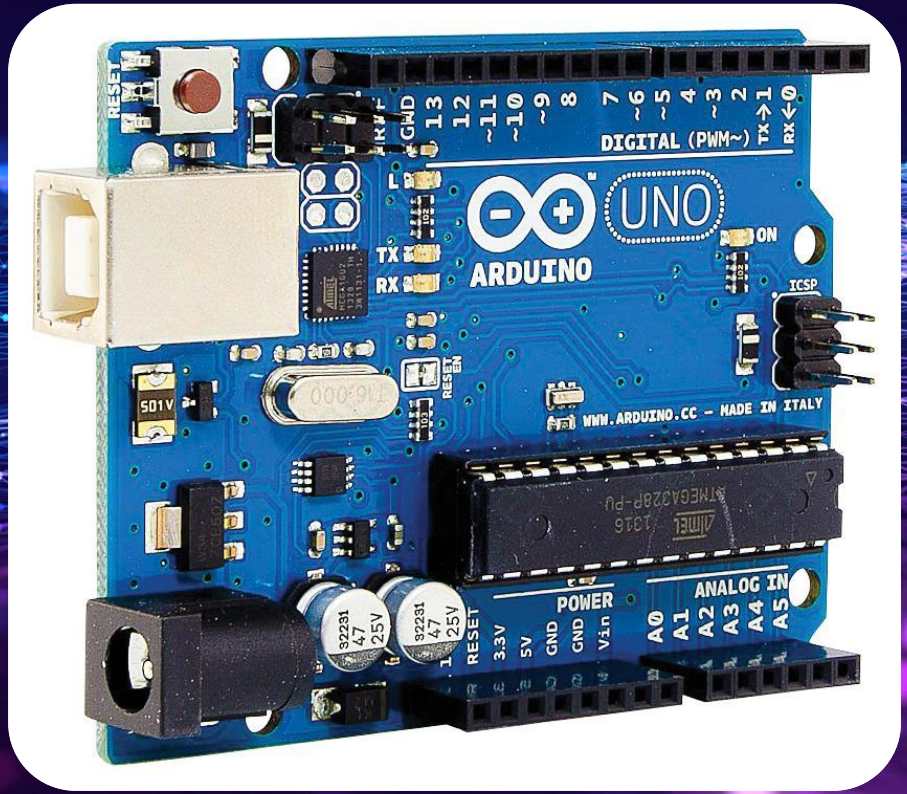
วิชา อินเทอร์เน็ตของทุกสรรพสิ่ง (Internet of Things : 4123412)



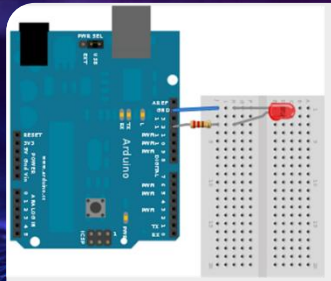
2.1 แนะนำอุปกรณ์พื้นฐานเกี่ยวกับ Internet of Things

2.1 แนะนำอุปกรณ์พื้นฐานเกี่ยวกับ Internet of Things

แนะนำ Arduino

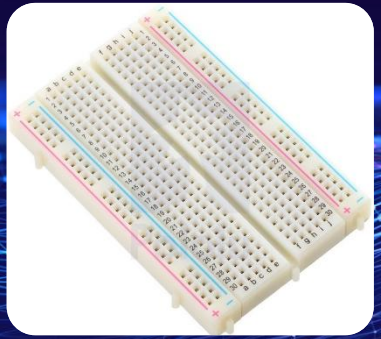


2.1 แนะนำอุปกรณ์พื้นฐานเกี่ยวกับ Internet of Things



รูปที่1 บอร์ด Arduino ต่อกับ LED

รูปที่2 บอร์ด Arduino ต่อกับบอร์ด XBee Shield



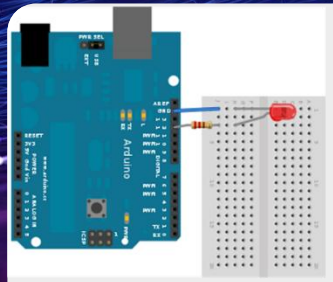
More...

2.1 แนะนำอุปกรณ์พื้นฐานเกี่ยวกับ Internet of Things

Arduino คืออะไร

Arduino อ่านว่า (อา-ดู-อิ-โน่ หรือ อาดุยโน่) เป็นบอร์ดไมโครคอนโทรเลอร์ตระกูล AVR ที่มีการพัฒนาแบบ Open Source คือมีการเปิดเผยข้อมูลทั้งด้าน Hardware และ Software ตัวบอร์ด Arduino ถูกออกแบบมาให้ใช้งานได้ง่าย ดังนั้นจึงเหมาะสำหรับผู้เริ่มต้นศึกษา ทั้งนี้ผู้ใช้งานยังสามารถดัดแปลง เพิ่มเติม พัฒนาต่อยอดทั้งตัวบอร์ด หรือโปรแกรมต่อได้อีกด้วย

ความง่ายของบอร์ด Arduino ในการต่ออุปกรณ์เสริมต่าง ๆ คือผู้ใช้งานสามารถต่อวงจรอิเล็กทรอนิคส์จากภายนอกแล้วเชื่อมต่อเข้ามาที่ขา I/O ของบอร์ด (ดูตัวอย่างรูปที่ 1) หรือเพื่อความสะดวกสามารถเลือกต่อกับบอร์ดเสริม (Arduino Shield) ประเภทต่าง ๆ (ดูตัวอย่างรูปที่ 2) เช่น Arduino XBee Shield, Arduino Music Shield, Arduino Relay Shield, Arduino GPRS Shield เป็นต้น มาเสียบกับบอร์ดบนบอร์ด Arduino แล้วเขียนโปรแกรมพัฒนาต่อได้เลย



รูปที่1 บอร์ด Arduino ต่อกับ LED

รูปที่2 บอร์ด Arduino ต่อกับบอร์ด XBee Shield

แหล่งที่มา: https://www.thaieasyelec.com/article-wiki/basic-electronics/what-is-arduino-ch1.html

2.1 อุปกรณ์พื้นฐานเกี่ยวกับ Internet of Things

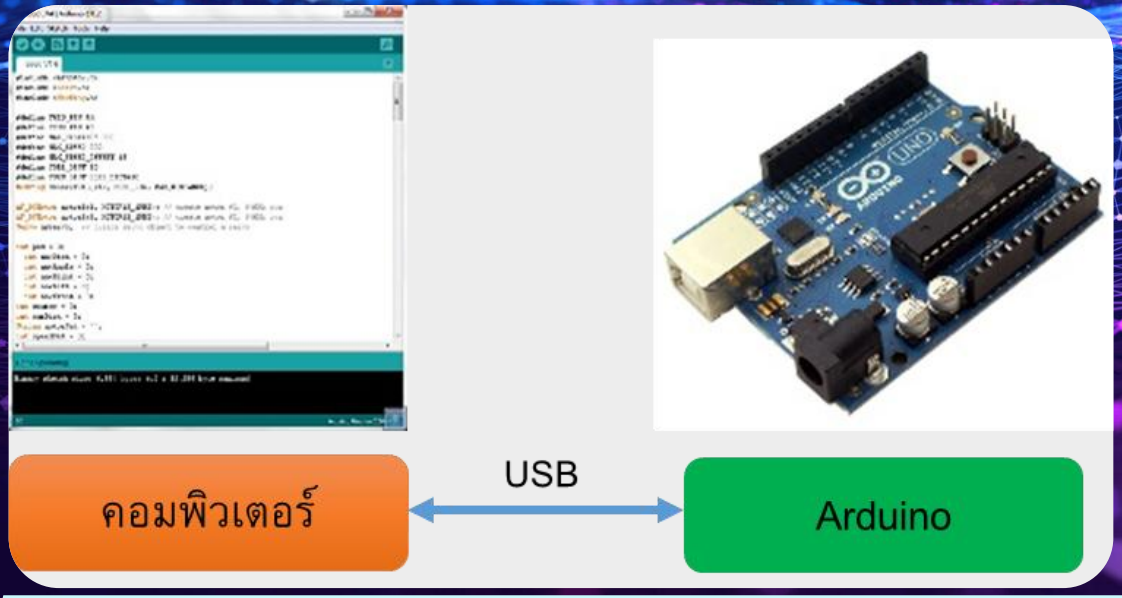
จุดเด่นที่ทำให้บอร์ด Arduino เป็นที่นิยม

- ง่ายต่อการพัฒนา มีรูปแบบคำสั่งพื้นฐาน ไม่ซับซ้อนเหมาะสำหรับผู้เริ่มต้น
- มี Arduino Community กลุ่มคนที่ร่วมกันพัฒนาที่แข็งแรง
- Open Hardware ทำให้ผู้ใช้สามารถนำบอร์ดไปต่อยอดใช้งานได้หลายด้าน
- ราคาไม่แพง
- Cross Platform สามารถพัฒนาโปรแกรมบน OS ใดก็ได้

แหล่งที่มา: https://www.thaieasyelec.com/article-wiki/basic-electronics/what-is-arduino-ch1.html

2.1 อุปกรณ์พื้นฐานเกี่ยวกับ Internet of Things

รูปแบบการเขียนโปรแกรมบน Arduino

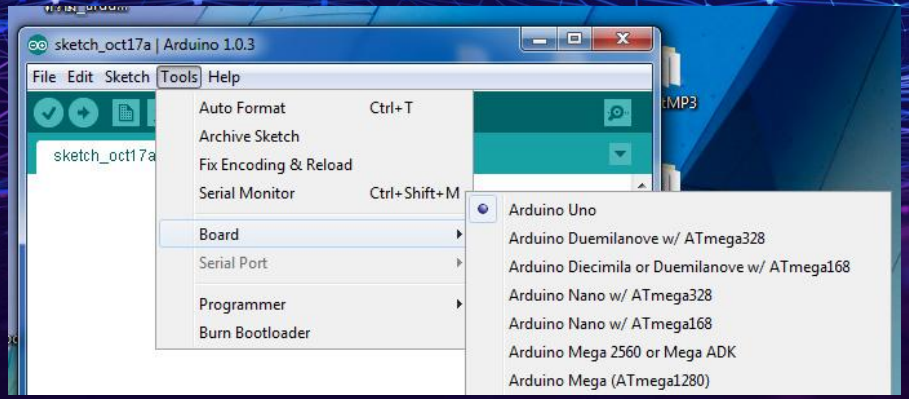


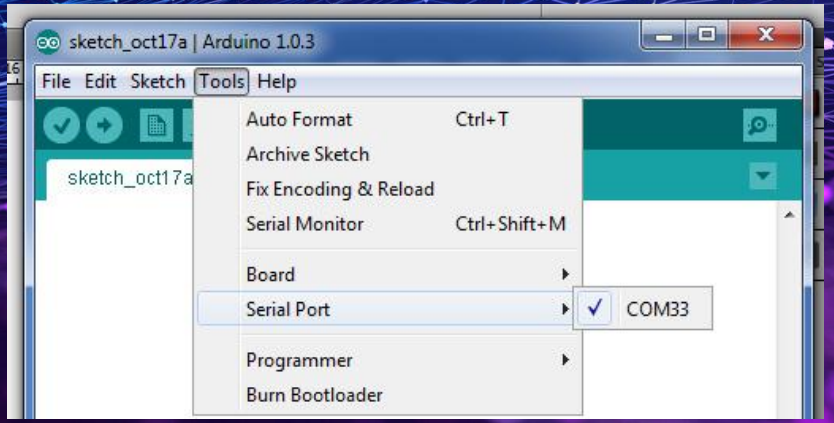
แหล่งที่มา: https://www.thaieasyelec.com/article-wiki/basic-electronics/what-is-arduino-ch1.html

2.1 อุปกรณ์พื้นฐานเกี่ยวกับ Internet of Things

รูปแบบการเขียนโปรแกรมบน Arduino

- 1. เขียนโปรแกรมบนคอมพิวเตอร์ ผ่านทางโปรแกรม Arduino IDE ซึ่งสามารถดาวน์โหลดได้จาก Arduino.cc/en/main/software
- 2. หลังจากที่เขียนโค้ดโปรแกรมเรียบร้อยแล้ว ให้ผู้ใช้งานเลือกรุ่นบอร์ด Arduino ที่ใช้และหมายเลข Com port



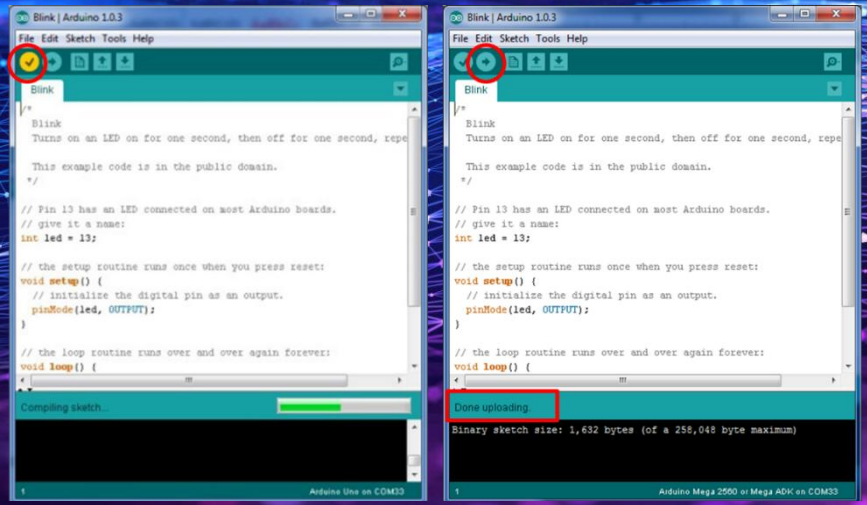


แหล่งที่มา: https://www.thaieasyelec.com/article-wiki/basic-electronics/what-is-arduino-ch1.html

2.1 อุปกรณ์พื้นฐานเกี่ยวกับ Internet of Things

รูปแบบการเขียนโปรแกรมบน Arduino

- 3. กดปุ่ม Verify เพื่อตรวจสอบความถูกต้องและ Compile โค้ดโปรแกรม จากนั้นกดปุ่ม Upload โค้ด โปรแกรมไปยังบอร์ด Arduino ผ่านทางสาย USB เมื่ออัปโหลดเรียบร้อยแล้ว จะแสดงข้อความแถบข้างล่าง “Done uploading” และบอร์ดจะเริ่มทำงานตามที่เขียนโปรแกรมไว้ได้ทันที

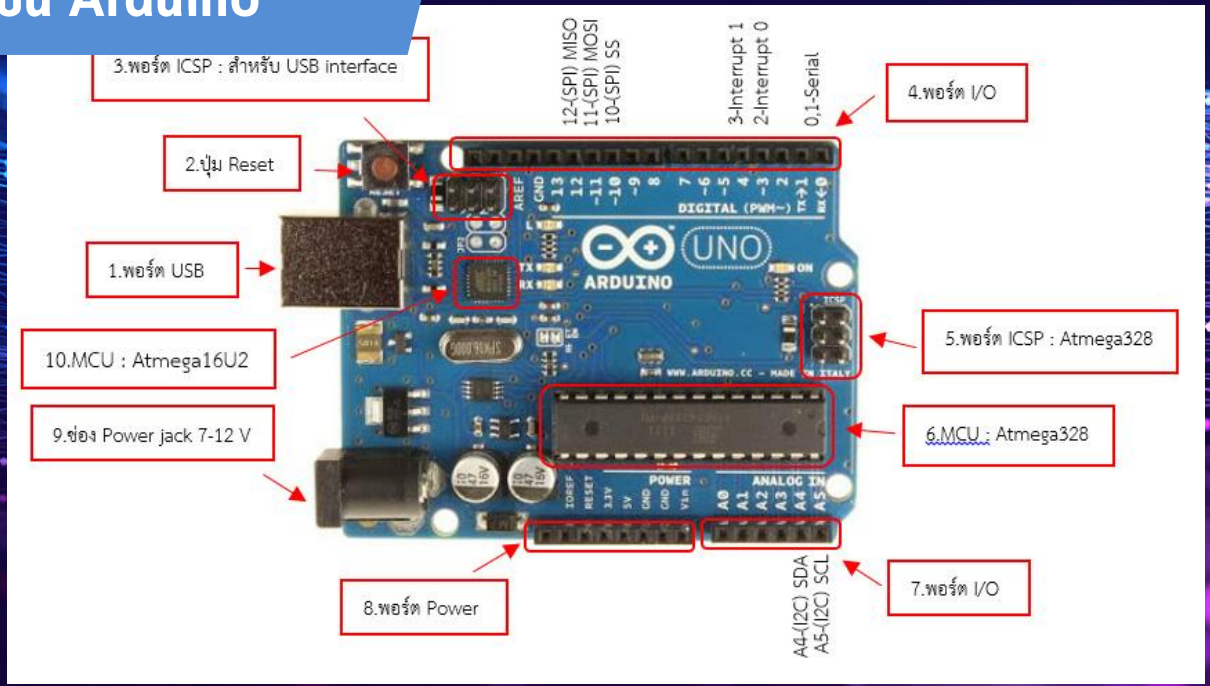


แหล่งที่มา: https://www.thaieasyelec.com/article-wiki/basic-electronics/what-is-arduino-ch1.html

2.1 อุปกรณ์พื้นฐานเกี่ยวกับ Internet of Things

รูปแบบการเขียนโปรแกรมบน Arduino

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



แหล่งที่มา: https://www.thaieasyelec.com/article-wiki/basic-electronics/what-is-arduino-ch1.html

2.1 อุปกรณ์พื้นฐานเกี่ยวกับ Internet of Things

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

- 1. USB Port: ใช้สำหรับต่อกับ Computer เพื่ออับโหลดโปรแกรมเข้า MCU และจ่ายไฟให้กับบอร์ด
- 2. Reset Button: เป็นปุ่ม Reset ใช้กดเมื่อต้องการให้ MCU เริ่มการทำงานใหม่
- 3. ICSP Port ของ Atmega16U2 เป็นพอร์ตที่ใช้โปรแกรม Visual Com port บน Atmega16U2
- 4. I/OPort:Digital I/O ตั้งแต่ขา D0 ถึง D13 นอกจากนี้ บาง Pin จะทำหน้าที่อื่นๆ เพิ่มเติมด้วย เช่น Pin0,1 เป็นขา Tx,Rx Serial, Pin3,5,6,9,10 และ 11 เป็นขา PWM
- 5. ICSP Port: Atmega328 เป็นพอร์ตที่ใช้โปรแกรม Bootloader
- 6. MCU: Atmega328 เป็น MCU ที่ใช้บนบอร์ด Arduino
- 7. I/OPort: นอกจากจะเป็น Digital I/O แล้ว ยังเปลี่ยนเป็น ช่องรับสัญญาณอนาล็อก ตั้งแต่ขา A0-A5
- 8. Power Port: ไฟเลี้ยงของบอร์ดเมื่อต้องการจ่ายไฟให้กับวงจรภายนอก ประกอบด้วยขาไฟเลี้ยง +3.3 V, +5V, GND, Vin
- 9. Power Jack: รับไฟจาก Adapter โดยที่แรงดันอยู่ระหว่าง 7-12 V
- 10. MCU ของ Atmega16U2 เป็น MCU ที่ทำหน้าที่เป็น USB to Serial โดย Atmega328 จะติดต่อกับ Computer ผ่าน Atmega16U2

แหล่งที่มา: https://www.thaieasyelec.com/article-wiki/basic-electronics/what-is-arduino-ch1.html

2.1 อุปกรณ์พื้นฐานเกี่ยวกับ Internet of Things

รุ่นต่าง ๆ ของ Arduino

1. Arduino Uno R3 เป็นบอร์ด Arduino ที่ได้รับความนิยมมากที่สุด เนื่องจากราคาไม่แพง ส่วนใหญ่โปรเจคและ Library ต่างๆ ที่พัฒนาขึ้นมา Support จะอ้างอิงกับบอร์ดนี้เป็นหลัก และข้อดีอีกอย่างคือ กรณีที่ MCU เสีย ผู้ใช้งานสามารถซื้อมาเปลี่ยนเองได้ง่าย

แหล่งที่มา: https://www.thaieasyelec.com/article-wiki/basic-electronics/what-is-arduino-ch1.html

2.1 อุปกรณ์พื้นฐานเกี่ยวกับ Internet of Things

รุ่นต่าง ๆ ของ Arduino

2. Arduino Uno SMD เป็นบอร์ดที่มีคุณสมบัติและการทำงานเหมือนกับบอร์ด Arduino UNO R3 ทุกประการ แต่จะแตกต่างกับที่Package ของ MCU ซึ่งบอร์ดนี้จะมี MCU ที่เป็น Package SMD (Arduino UNO R3 มี MCU ที่เป็น Package DIP)

แหล่งที่มา: https://www.thaieasyelec.com/article-wiki/basic-electronics/what-is-arduino-ch1.html

2.1 อุปกรณ์พื้นฐานเกี่ยวกับ Internet of Things

รุ่นต่าง ๆ ของ Arduino

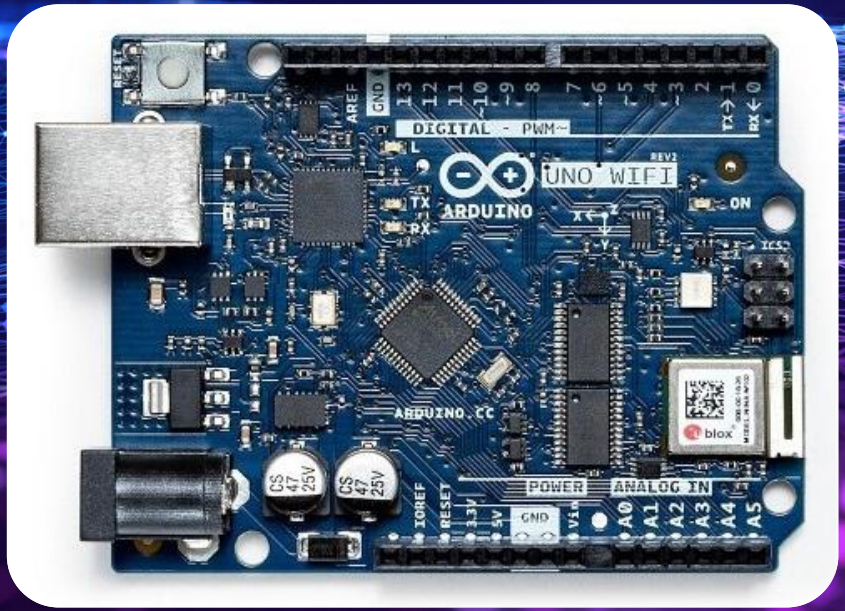
3. Arduino Mega 2560 R3 เป็นบอร์ด Arduino ที่ออกแบบมาสำหรับงานที่ต้องใช้ I/O มากกว่า Arduino Uno R3 เช่น งานที่ต้องการรับสัญญาณจาก Sensor หรือควบคุมมอเตอร์ Servo หลายๆ ตัว ทำให้ Pin I/O ของบอร์ด Arduino Uno R3 ไม่สามารถรองรับได้ ทั้งนี้บอร์ด Mega 2560 R3 ยังมีความหน่วยความจำแบบ Flash มากกว่า Arduino Uno R3 ทำให้สามารถเขียนโค้ดโปรแกรมเข้าไปได้มากกว่า ในความเร็วของ MCU ที่เท่ากัน

แหล่งที่มา: https://www.thaieasyelec.com/article-wiki/basic-electronics/what-is-arduino-ch1.html

2.1 อุปกรณ์พื้นฐานเกี่ยวกับ Internet of Things

รุ่นต่าง ๆ ของ Arduino

4. Arduino UNO WiFi Rev2 ใช้ชิพ ATmega4809 ซึ่งมี Flash memory 48KB, SRAM 6KB และยังประหยัดพลังงานมากกว่าตัวเวอร์ชั่นเก่า อีกทั้งตัวบอร์ด Arduino Uno Wifi Rev2 ยังมาพร้อมกับชิพ ESP32 u-Blox NiNA-W13 Wifi ที่ทำให้ตัวบอร์ดรองรับใช้งานร่วมกับ Wifi และมีฟังก์ชั่นการใช้งานที่หลากหลายมากขึ้น



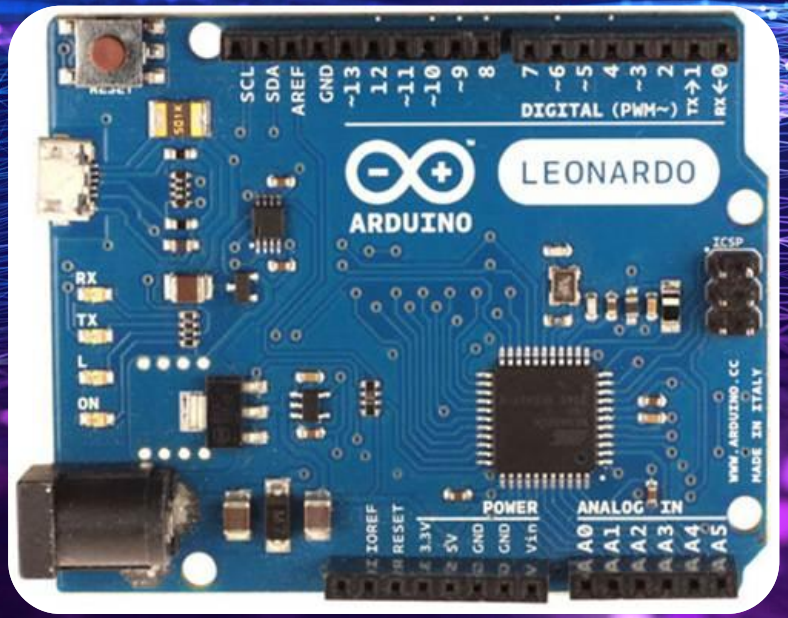
แหล่งที่มา: https://www.thaieasyelec.com/article-wiki/basic-electronics/what-is-arduino-ch1.html

2.1 อุปกรณ์พื้นฐานเกี่ยวกับ Internet of Things

รุ่นต่าง ๆ ของ Arduino

5. Arduino Leonardo การทำงานจะคล้ายกับบอร์ด Arduino Uno R3 แต่มีการเปลี่ยน MCU ตัวใหม่เป็น ATmega32U4 ซึ่งมีโมดูลพอร์ต USB มาด้วยบนชิป (แตกต่างจากบอร์ด Arduino UNO R3 หรือ Arduino Mega 2560 ที่ต้องใช้ชิป ATmega16U2 ร่วมกับ Atmega328 ในการเชื่อมต่อกับพอร์ต USB)

ข้อควรระวัง: เนื่องจาก MCU เป็นคนละเบอร์กับ Arduino Uno R3 อาจจะทำให้บอร์ด Shield บางตัวหรือ Library ใช้ร่วมกันกับบอร์ดArduino Leonardo ไม่ได้ ผู้ใช้งานจำเป็นต้องตรวจสอบก่อนใช้งาน



แหล่งที่มา: https://www.thaieasyelec.com/article-wiki/basic-electronics/what-is-arduino-ch1.html

2.1 อุปกรณ์พื้นฐานเกี่ยวกับ Internet of Things

รุ่นต่าง ๆ ของ Arduino

6. Arduino MKR WIFI 1010 เป็นบอร์ดในตระกูล MKR ของ Arduino ปรับปรุงจากบอร์ด MKR 1000 WIFI ประกอบด้วย 3 ส่วน ได้แก่ (1) ชิพหลัก SAMD21 ARM Cortex-M0+ 32-bit Low Power MCU 48 MHz SRAM 32 KB และ Flash Memory 256 KB มี Digital I/O 8 ขา Analog Input 7 ขา Analog Output 1 ขา UART 1 ชุด SPI 1 ชุด I2C 1 ชุด รองรับ External Interrupt 8 ขา รองรับ PWM 12 ขา แรงดันขา I/O ทำงานที่ 3.3 โวลต์เท่านั้น (2) ชิพ ESP32 U-BLOX NINA-W10 Series Low Power 2.4 GHz 802.11 bgn (3) ชิพ ECC508 Crypto Authentication

2.1 อุปกรณ์พื้นฐานเกี่ยวกับ Internet of Things

รุ่นต่าง ๆ ของ Arduino

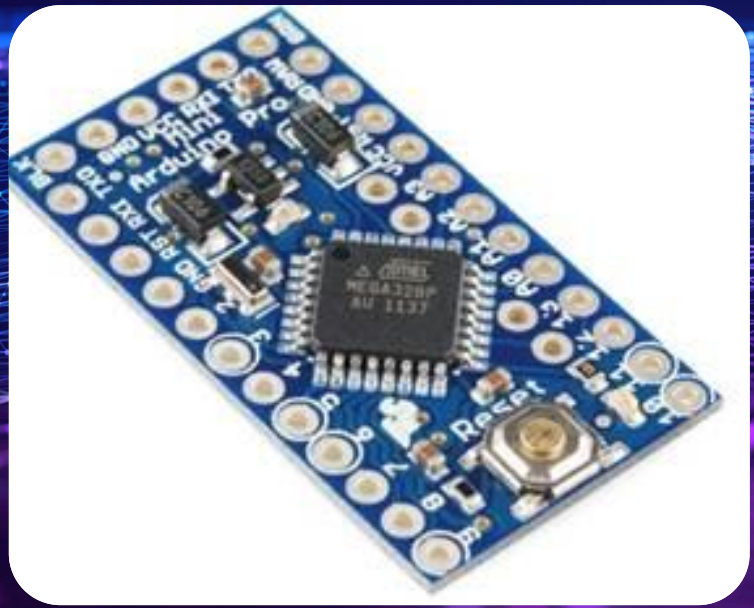
7. Arduino Pro Mini 328 3.3V เป็นบอร์ด Arduino ขนาดเล็ก ที่ใช้ MCU เบอร์ ATmega328 ซึ่งจะคล้ายกับบอร์ด Arduino Mini05 แต่บนบอร์ดจะมี Regulator 3.3 V ชุดเดียวเท่านั้น ระดับแรงดันไฟที่ขา I/O คือ 3.3V

แหล่งที่มา: https://www.thaieasyelec.com/article-wiki/basic-electronics/what-is-arduino-ch1.html

2.1 อุปกรณ์พื้นฐานเกี่ยวกับ Internet of Things

รุ่นต่าง ๆ ของ Arduino

8. Arduino Pro Mini 328 5V เป็นบอร์ด Arduino ขนาดเล็ก ที่ใช้ MCU เบอร์ ATmega328 เช่นเดียวกับบอร์ด Arduino Mini 05 แต่บนบอร์ดจะมี Regulator 5V ชุดเดียวเท่านั้น ระดับแรงดันไฟที่ขา I/O คือ 5V

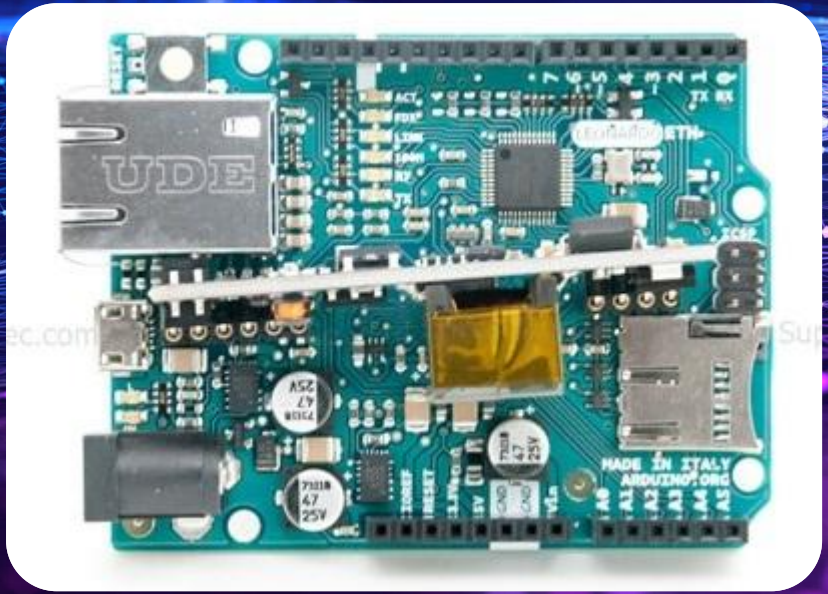


แหล่งที่มา: https://www.thaieasyelec.com/article-wiki/basic-electronics/what-is-arduino-ch1.html

2.1 อุปกรณ์พื้นฐานเกี่ยวกับ Internet of Things

รุ่นต่าง ๆ ของ Arduino

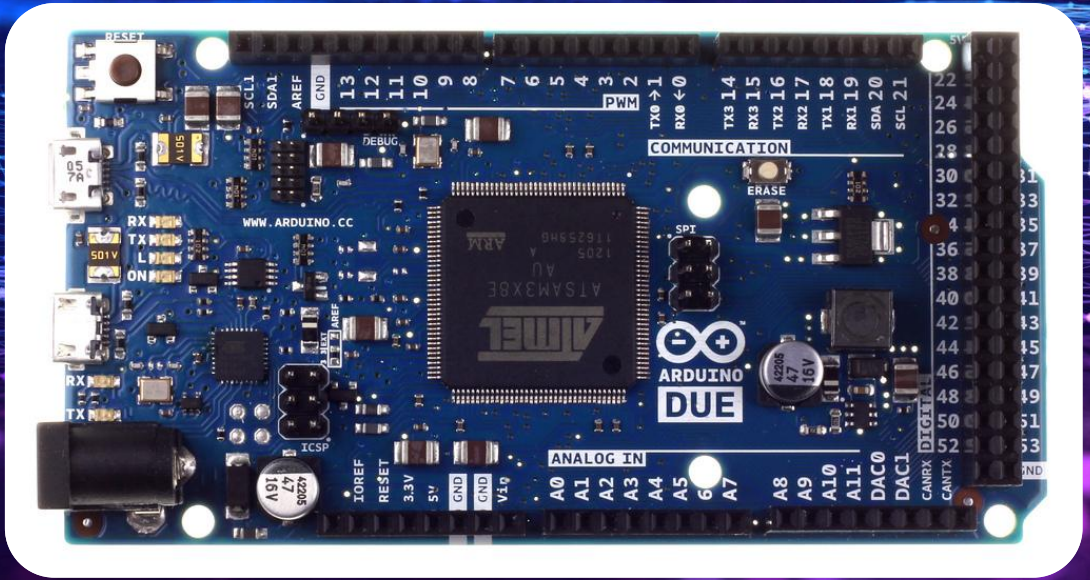
9. Arduino Ethernet with PoE module Arduino Leonardo ETH with PoE เป็นบอร์ดจาก Arduino.org ที่มี ATmega32U4 พร้อมกับโมดูล Ethernet ในตัว เหมาะสำหรับผู้ที่ต้องการใช้งานควบคุมผ่าน Network (TCP/IP)



2.1 อุปกรณ์พื้นฐานเกี่ยวกับ Internet of Things

รุ่นต่าง ๆ ของ Arduino

10. Arduino Due เป็นบอร์ด Arduino ที่เปลี่ยนชิป MCU ใหม่ ซึ่งจากเดิมเป็นตระกูล AVR เปลี่ยนเป็นเบอร์ AT91SAM3X8E (ตระกูลARM Cortex-M3) แทน ทำให้การประมวลผลเร็วขึ้น แต่ยังคงรูปแบบโค้ดโปรแกรมของ Arduino ที่ง่ายอยู่ ข้อควรระวัง: เนื่องจากMCU เป็นคนละเบอร์กับ Arduino Uno R3 อาจะทำให้บอร์ด Shield บางตัวหรือ Library ใช้ร่วมกันกับบอร์ด Arduino Leonardo ไม่ได้ ผู้ใช้งานจำเป็นต้องตรวจสอบก่อนใช้งาน



2.1 อุปกรณ์พื้นฐานเกี่ยวกับ Internet of Things

รุ่นต่าง ๆ ของ Arduino

จากตารางจะเห็นได้ว่าเนื่องจากบอร์ด Arduino UNO R3 เป็นรุ่นที่ได้รับความนิยมมากที่สุด ทำให้ Library และบอร์ด Shield ส่วนใหญ่จะรองรับกับบอร์ดรุ่นนี้ ดังนั้น จึงขออธิบายรายละเอียดต่าง ๆ โดยอิงจากบอร์ด Arduino UNO R3 เป็นหลัก

ตารางเปรียบเทียบคุณสมบัติของบอร์ด Arduino แต่ละรุ่น

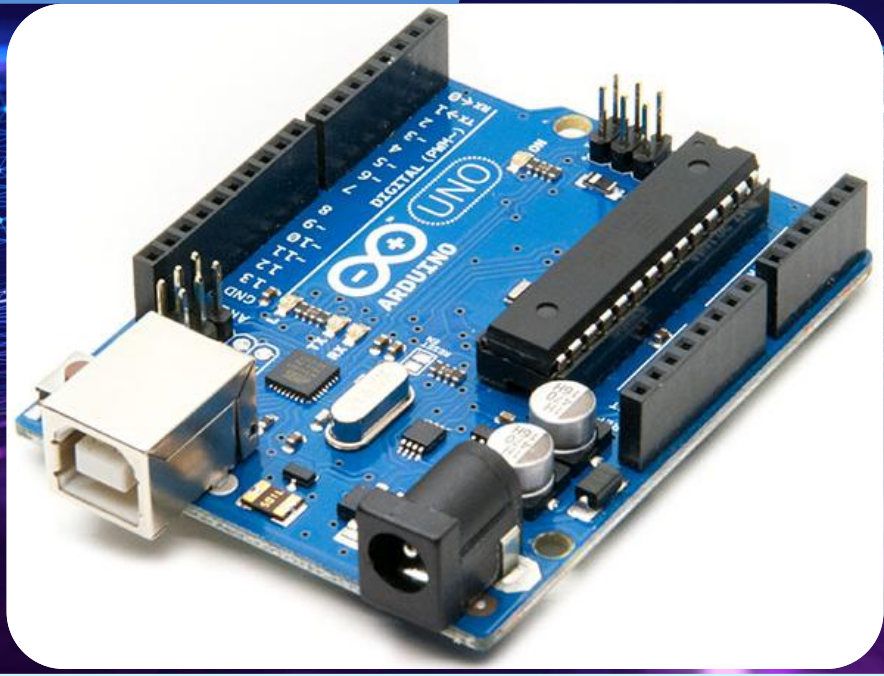
	Processor					Input / Output							Power				Connectivity				
	Family	SRAM	FLASH	EEPROM	Clock	Digital I/O	Analog In	ADC Bits	PWM	UART	Analog Out	DAC Bits	VCC	Vin Range	5V	3V3	USB-Serial	I2C	Ethernet	USB-Host	SD Card
Arduino UNO R3	ATmega328	2k	32k	1k	16MHz	14	6	10	6	1	N/A	N/A	5V	7-12V	Yes	Yes	ATmega16U2	1	No	No	No
Arduino UNO SMD	ATmega328	2k	32k	1k	16MHz	14	6	10	6	1	N/A	N/A	5V	7-12V	Yes	Yes	ATmega16U2	1	No	No	No
Arduino Mega 2560 R3	ATmega2560	8k	256k	4k	16MHz	54	16	10	14	4	N/A	N/A	5V	7-18V	Yes	Yes	ATmega16U2	1	No	No	No
Arduino Mega ADK	ATmega2560	8k	256k	4k	16MHz	54	16	10	14	4	N/A	N/A	5V	7-18V	Yes	Yes	ATmega16U2	1	MAX3421E	No	No
Arduino Leonardo	ATmega32U4	2.5k	32k	1k	16MHz	25	12	10	7	1	N/A	N/A	5V	7-12V	Yes	Yes	Built-In	1	No	No	No
Arduino Mini 05	ATmega328	2k	32k	1k	16MHz	14	6	10	6	1	N/A	N/A	5V	7V-9V	Yes	No	N/A	1	No	No	No
Arduino Pro Mini 328 - 3.3V	ATmega328	2k	32k	1k	8MHz	14	6	10	6	1	N/A	N/A	3.3V	5V-12V	No	Yes	N/A	1	No	No	No
Arduino Pro Mini 328 – 5V	ATmega328	2k	32k	1k	16MHz	14	6	10	6	1	N/A	N/A	5V	7V-12V	Yes	No	N/A	1	No	No	No
Arduino Ethernet with PoE module	ATmega328	2k	32k	1k	16MHz	9	6	10	4	1	N/A	N/A	5V	6-18V	Yes	Yes	N/A	1	No	No	No
Arduino Ethernet without PoE module	ATmega328	2k	32k	1k	16MHz	9	6	10	4	1	N/A	N/A	5V	6-18V	Yes	Yes	N/A	1	No	No	No
Arduino DUE	SAM3X8E	96kb	512k	N/A	84MHz	70	12	12	12	4	2	12	3.3V	7-12V	No	VCC	Built-In	2	No	Yes	No

แหล่งที่มา: https://www.thaieasyelec.com/article-wiki/basic-electronics/what-is-arduino-ch1.html

2.1 อุปกรณ์พื้นฐานเกี่ยวกับ Internet of Things

Arduino Uno R3

Arduino Uno R3 เป็นบอร์ด Arduino ที่ได้รับความนิยมมากที่สุด เนื่องจากราคาไม่แพง ส่วนใหญ่โปรเจคและ Library ต่างๆ ที่พัฒนาขึ้นมา Support จะอ้างอิงกับบอร์ดนี้เป็นหลัก เนื่องจากเป็นขนาดที่เหมาะสำหรับการเริ่มต้นเรียนรู้ Arduino และมี Shields ให้เลือกใช้งานได้มากกว่าบอร์ด Arduino รุ่นอื่นๆที่ออกแบบมาเฉพาะมากกว่า โดยบอร์ด Arduino Uno ได้มีการพัฒนาเรื่อยมา ตั้งแต่ R2 R3 และรุ่นย่อยที่เปลี่ยนชิปไอซีเป็นแบบ SMD และข้อดีอีกอย่างคือ กรณีที่ MCU เสีย ผู้ใช้งานสามารถซื้อมาเปลี่ยนเองได้ง่าย

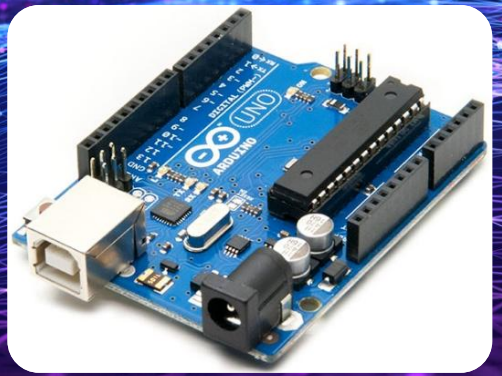


แหล่งที่มา: https://www.thaieasyelec.com/article-wiki/basic-electronics/what-is-arduino-ch1.html

2.1 อุปกรณ์พื้นฐานเกี่ยวกับ Internet of Things

ข้อมูลของบอร์ด Arduino Uno R3

ชิปไอซีไมโครคอนโทรเลอร์	ATmega328
ใช้แรงดันไฟฟ้า	5V
รองรับการจ่ายแรงดันไฟฟ้า (ที่แนะนำ)	7 – 12V
รองรับการจ่ายแรงดันไฟฟ้า (ที่จำกัด)	6 – 20V
พอร์ต Digital I/O	14 พอร์ต (มี 6 พอร์ต PWM output)
พอร์ต Analog Input	6 พอร์ต
กระแสไฟที่จ่ายได้ในแต่ละพอร์ต	40mA
กระแสไฟที่จ่ายได้ในพอร์ต 3.3V	50mA
พื้นที่โปรแกรมภายใน	32KB พื้นที่โปรแกรม, 500B ใช้โดย Booloader
พื้นที่แรม	2KB
พื้นที่หน่วยความจำถาวร (EEPROM)	1KB
ความถี่คริสตัล	16MHz
ขนาด	68.6x53.4 mm
น้ำหนัก	25 กรัม



แหล่งที่มา: https://www.thaieasyelec.com/article-wiki/basic-electronics/what-is-arduino-ch1.html

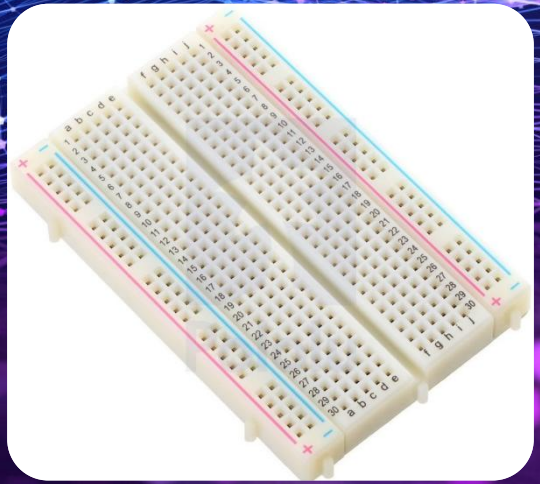
2.1 อุปกรณ์พื้นฐานเกี่ยวกับ Internet of Things

Breadboard บอร์ดทดลอง

Breadboard หรือบอร์ดทดลอง คือแผ่นพลาสติกหรือไฟเบอร์กลาสที่มีรูเล็ก ๆ เรียงเป็นแถวและคอลัมน์ ใช้สำหรับต่อวงจรอิเล็กทรอนิกส์แบบชั่วคราวโดยไม่ต้องบัดกรี

ลักษณะและโครงสร้าง:

- มีรูเล็กๆ เรียงเป็นตาราง โดยแต่ละรูสามารถเสียบขาของอุปกรณ์อิเล็กทรอนิกส์หรือสายไฟได้
- ภายในมีแผ่นโลหะเชื่อมต่อกันในแต่ละแถวหรือคอลัมน์ ทำให้เกิดการนำไฟฟ้า
- ส่วนกลางมักมีร่องแบ่งเพื่อแยกด้านซ้ายและขวา
- ด้านข้างมีรางสำหรับจ่ายไฟ (Power Rail) สำหรับ VCC และ GND



ประโยชน์:

- ทดลองวงจรได้อย่างรวดเร็วและง่ายดาย
- ไม่ต้องใช้ความร้อนในการบัดกรี
- สามารถปรับเปลี่ยนหรือแก้ไขวงจรได้ทันที
- ใช้ซ้ำได้หลายครั้ง
- เหมาะสำหรับการเรียนรู้และพัฒนาต้นแบบ

ข้อจำกัด:

- ไม่เหมาะสำหรับวงจรที่ใช้กระแสหรือความถี่สูง
- การเชื่อมต่อไม่แน่นหนาเท่ากับการบัดกรี
- ไม่เหมาะสำหรับใช้งานถาวร

Breadboard เป็นเครื่องมือพื้นฐานที่สำคัญมากสำหรับคนที่เรียนรู้อิเล็กทรอนิกส์ วิศวกร และนักประดิษฐ์

แหล่งที่มา: https://www.thaieasyelec.com/article-wiki/basic-electronics/what-is-arduino-ch1.html

2.1 อุปกรณ์พื้นฐานเกี่ยวกับ Internet of Things

หัวแร้ง,บัดกรี

หัวแร้ง (Soldering Iron) และ บัดกรี (Solder) คือเครื่องมือและวัสดุสำคัญในงาน IoT ที่ใช้เชื่อมต่อชิ้นส่วนอิเล็กทรอนิกส์อย่างถาวร

หัวแร้ง (Soldering Iron)

คืออะไร:

- เครื่องมือที่ใช้ความร้อนในการหลอมเหลวบัดกรี
- มีหัวแร้งที่ทำจากทองแดงหรือเหล็ก เมื่อร้อนจะใช้หลอมบัดกรี
- ชนิดที่ใช้ในงาน IoT:
- หัวแร้งไฟฟ้า 25-40 วัตต์ (เหมาะสำหรับงานละเอียด)
- หัวแร้งแบบควบคุมอุณหภูมิได้
- หัวแร้งแบบปากกา (สำหรับ SMD components)

บัดกรี (Solder)

คืออะไร:

- โลหะผสมที่หลอมเหลวที่อุณหภูมิต่ำ ใช้เชื่อมต่อชิ้นส่วนอิเล็กทรอนิกส์
- ส่วนใหญ่ผสมระหว่างดีบุก (Tin) และตะกั่ว (Lead) หรือแบบไร้ตะกั่ว
- ชนิดที่ใช้ในงาน IoT:
- บัดกรีแบบมีไส้ฟลักซ์ (Flux Core) - ง่ายต่อการใช้งาน
- ขนาด 0.6-1.0 มม. เหมาะสำหรับงานละเอียด
- บัดกรีไร้ตะกั่ว (Lead-free) - ปลอดภัยกว่า

แหล่งที่มา: https://www.thaieasyelec.com/article-wiki/basic-electronics/what-is-arduino-ch1.html

2.1 อุปกรณ์พื้นฐานเกี่ยวกับ Internet of Things

ตัวต้านทาน

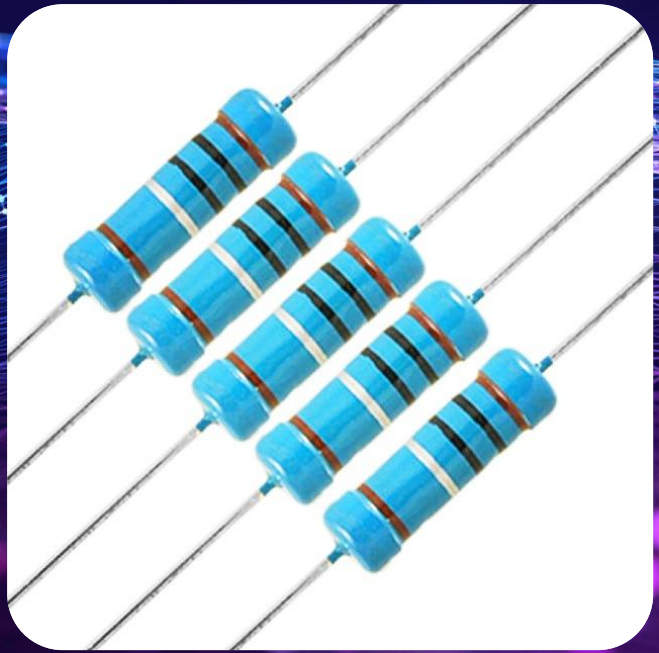
ตัวต้านทาน (Resistor) คือชิ้นส่วนอิเล็กทรอนิกส์พื้นฐานที่ใช้ต้านทานการไหลของกระแสไฟฟ้า โดยแปลงพลังงานไฟฟ้าเป็นความร้อน

หลักการทำงาน

- ขัดขวางการไหลของกระแสไฟฟ้า ทำให้กระแสลดลง
- สร้างการตกแรงดันในวงจร
- แปลงพลังงานไฟฟ้าเป็นความร้อน

ประเภทของตัวต้านทาน

1. ตัวต้านทานคาร์บอน (Carbon Resistor)
   - ราคาถูก ใช้งานทั่วไป
   - ความแม่นยำปานกลาง (±5%, ±10%)
2. ตัวต้านทานฟิล์ม (Film Resistor)
   - ความแม่นยำสูง (±1%, ±0.1%)
   - เสียงรบกวนต่ำ
3. ตัวต้านทานแปรค่าได้ (Variable Resistor/Potentiometer)
   - ปรับค่าได้ด้วยการหมุน
   - ใช้ควบคุมระดับเสียง, ความสว่าง



แหล่งที่มา: https://www.thaieasyelec.com/article-wiki/basic-electronics/what-is-arduino-ch1.html

2.1 อุปกรณ์พื้นฐานเกี่ยวกับ Internet of Things

สายไฟจั๊มเปอร์

สายไฟจั๊มเปอร์ ตัวผู้-ตัวเมีย

สายไฟจั๊มเปอร์ ตัวผู้- ตัวผู้

สายไฟจั๊มเปอร์ ตัวเมีย- ตัวเมีย

แหล่งที่มา: https://www.thaieasyelec.com/article-wiki/basic-electronics/what-is-arduino-ch1.html

2.1 อุปกรณ์พื้นฐานเกี่ยวกับ Internet of Things

หลอดไฟ LED 5mm

หลอดไฟ LED 5mm คือหลอดไฟ LED แบบ Through-Hole ที่มีขนาดเส้นผ่านศูนย์กลาง 5 มิลลิเมตร เป็นหลอดไฟ LED ขนาดมาตรฐานที่ใช้กันอย่างแพร่หลาย

ลักษณะทางกายภาพ

โครงสร้าง:

- ตัวหลอดทำจากพลาสติกใส หรือสีต่างๆ
- มีขา 2 ขา (Anode และ Cathode)
- เส้นผ่านศูนย์กลาง 5 มิลลิเมตร
- ความสูงประมาณ 8-9 มิลลิเมตร

การจำแนกขั้ว:

- ขาบวก (Anode) - ขายาวกว่า
- ขาลบ (Cathode) - ขาสั้นกว่า และตัวหลอดจะมีด้านแบนเล็กน้อย

วิชา อินเทอร์เน็ตของทุกสรรพสิ่ง (Internet of Things : 4123412)

2.2 การเขียนโปรแกรมภาษา C ด้วย Arduino IDE

2.2 การเขียนโปรแกรมภาษา C ด้วย Arduino IDE

โครงสร้างของ Arduino มี 2 ส่วนหลัก ๆ คือ

```
void setup()
{
    // ทำงานเพียงแค่ครั้งแรก ครั้งเดียว
}
void loop()
{
    // ทำงานซ้ำเรื่อย ๆ จนกว่าจะถอดปลั๊ก
}
```

2.2 การเขียนโปรแกรมภาษา C ด้วย Arduino IDE

ชนิดและประเภทของตัวแปร

ถ้าหากว่าเราจะเปรียบเทียบว่า ตัวแปร คือ ภาชนะสำหรับบรรจุสิ่งของ และ ข้อมูล คือ สิ่งของที่เรา ต้องการจะเก็บ จะเห็นได้ว่า สิ่งของต่าง ๆ รอบ ๆตัวเรานั้น จะมีคุณสมบัติที่แตกต่างกันไป ดังนั้นในการเลือก ภาชนะสำหรับใช้บรรจุสิ่งของ เราก็จำเป็นต้องเลือกชนิดของภาชนะให้มีความเหมาะสมที่จะใช้เก็บสิ่งของ ด้วย ซึ่งสิ่งแรกที่ต้องพิจารณาคือ เราจะต้องรู้จักคุณสมบัติของสิ่งของที่ต้องการจะจัดเก็บ และ จุดประสงค์ การใช้งาน ก่อน จากนั้นจึงจัดหาภาชนะที่มี ขนาด และ รูปทรงของภาชนะ เหมาะสมที่จะใช้เก็บสิ่งของ เพื่อให้สามารถจัดเก็บ และ นำสิ่งของออกมาใช้งานได้อย่างเขประหยัดมากที่สุด

ในภาษาซีนั้น มีการกำหนด และ จำแนก ชนิดของตัวแปร ไว้เป็น 5 ชนิดด้วยกัน โดยแต่ละชนิดจะมี คุณสมบัติการใช้งานที่ต่างกัน เพื่อใช้ในการเก็บข้อมูลที่มีรูปแบบแตกต่างกัน คือ

- char ใช้เก็บข้อมูลที่เป็นตัวอักษร (character) ใช้เก็บข้อมูลที่เป็นเลขจำนวนเต็มได้ 256 ค่า
- int ใช้เก็บข้อมูลที่เป็นเลขจำนวนเต็ม (integer) ใช้เก็บข้อมูลที่เป็นเลขจำนวนเต็มได้ 65536 ค่า
- float ใช้เก็บข้อมูลที่เป็นเลขทศนิยมแบบ Single Precision
- double ใช้เก็บข้อมูลที่เป็นเลขทศนิยมแบบ Double Precision ซึ่ง สามารถเก็บค่าตัวเลขทศนิยมที่มีความละเอียดและถูกต้องของทศนิยมมากว่าแบบ float ถึง 2 เท่า
- void ใช้เก็บตัวแปรที่ไม่มีค่า

2.2 การเขียนโปรแกรมภาษา C ด้วย Arduino IDE

ตารางชนิดตัวแปร

ตัวแปร	ขนาด	ค่าของตัวแปร	การประกาศค่า
boolean	1 bit	True, fales	boolean a = true;
byte	8 bit	0 – 255	byte a = 10;
char	8 bit	ASCII	char a = ‘a’;
int	16 bit	-32,768 – 32,767	int a = 1000;
unsigned int	16 bit	0 ถึง 65,535	unsigned int a = 13;
float	32 bit	-3.4028235E+38 ถึง 3.4028235E+38.	Float a = 10.77;

2.2 การเขียนโปรแกรมภาษา C ด้วย Arduino IDE

คุณสมบัติเฉพาะของตัวแปร

สำหรับตัวแปรชนิดที่ใช้เก็บค่าเลขจำนวนเต็ม (char และ int) นั้น ในภาษาซี ไม่ได้มีการจำแนก ชนิด ของตัวแปรเพื่อใช้เก็บค่าตัวเลขที่เป็น ค่าบวก หรือ ค่าลบ เป็นการเฉพาะ แต่ภาษาซี จะใช้วิธีการเพิ่ม คำสั่ง สำหรับกำหนดคุณสมบัติเฉาพะให้กับตัวแปรไว้อีก 4 คำสั่งสำหรับใช้กำหนดคุณสมบัติของตัวแปรแบบนี้ ให้มีคุณสมบัติที่เฉพาะเจาะจงลงไปอีกเป็นต้นว่า จะใช้ตัวแปรในการเก็บค่าตัวเลขที่เป็นค่าบวกอย่างเดียว หรือต้องการเก็บค่าตัวเลขแบบคิดเครื่องหมายด้วย เพื่อให้ผู้ใช้สามารถปรับแก้คุณสมบัติในการใช้งานของตัวแปรให้มีคุณสมบัติใกล้เคียงกับความต้องการใช้งานมากขึ้นไปอีก และเพื่อจำกัดขอบเขตการใช้งานของ ตัวแปรให้ตรงกับจุดประสงค์มากยิ่งขึ้น และยังเป็นการช่วยให้ประหยัดจำนวนของหน่วยความจำที่ใช้สร้าง ตัวแปรด้วยและทำให้โปรแกรมทำงานได้เร็วขึ้นกว่าเดิมอีกด้วย โดยในภาษาซีจะมีคำสั่ง ที่ใช้สำหรับระบุ คุณสมบัติเฉพาะของตัวแปรที่ใช้เก็บค่าเลขจำนวนเต็ม 4 คำสั่ง คือ

- unsigned ใช้ระบุให้เก็บค่าเลขจำนวนเต็มในตัวแปรเฉพาะค่าที่เป็นบวกเท่านั้น
- signed ใช้ระบุให้เก็บค่าเลขจำนวนเต็มในตัวแปรทั้งค่า บวก และ ลบ
- short ใช้ระบุให้เก็บค่าเลขจำนวนเต็มในตัวแปรที่มีค่าน้อยกว่า int
- long ใช้ระบุให้เก็บค่าเลขจำนวนเต็มในตัวแปรที่มีค่ามากกว่า int เป็น 2 เท่า

2.2 การเขียนโปรแกรมภาษา C ด้วย Arduino IDE

คุณสมบัติเฉพาะของตัวแปร

ชนิดข้อมูล	การเก็บข้อมูล	ขนาด
boolean	จริง (True) หรือ เท็จ (False)	1 บิต
char	ตัวเลข หรือตัวอักษร	1 ไบต์ ใส่ค่าได้ตั้งแต่ -128 ถึง 127
unsigned char	ตัวเลข หรือตัวอักษร	1 ไบต์ ใส่ค่าได้ตั้งแต่ 0 ถึง 255
byte	ไบต์	1 ไบต์ ใส่ค่าได้ตั้งแต่ 0 ถึง 255
int	ตัวเลขจำนวนเต็ม	2 ไบต์ ใส่ค่าได้ตั้งแต่ -32,768 ถึง 32,767
unsigned int	ตัวเลขจำนวนเต็ม	2 ไบต์ ใส่ค่าได้ตั้งแต่ 0 ถึง 65,535 (2^16) - 1)
long	ตัวเลขจำนวนเต็มที่มีความยาว	4 ไบต์ ใส่ค่าได้ตั้งแต่ -2,147,483,648 ถึง 2,147,483,647
unsigned long	ตัวเลขจำนวนเต็มที่มีความยาว	4 ไบต์ ใส่ค่าได้ตั้งแต่ 0 ถึง 4,294,967,295 (2^32 - 1)
float	ตัวเลขทศนิยมใช้ในการคำนวน	4 ไบต์ ใส่ค่าได้ตั้งแต่ 3.4028235E+38 ถึง -3.4028235E+38 มีทศนิยมได้ 6 ถึง 7 ตำแหน่ง
double (เฉพาะบอร์ด Arduino Due)	ตัวเลขทศนิยมที่มีความยาวและต้องการความแม่นยำ	8 ไบต์ ใช้ในการคำนวนที่ต้องการประสิทธิ์ภาพสูง
String	ข้อความ	ไม่ระบุ

2.2 การเขียนโปรแกรมภาษา C ด้วย Arduino IDE

การประกาศตัวแปร กำหนดค่าตัวแปร

การประกาศตัวแปรจะเหมือนกับภาษา C โดยปกติ คือ TYPE KEY;
โดย TYPE เป็นชนิดของข้อมูล ส่วน KEY เป็นชื่อตัวแปร การประกาศตัวแปรข้างต้นคือการประกาศตัวแปรแบบไม่กำหนดค่า ดังนั้นค่าปกติที่อ่านจากตัวแปรจะเป็น 0

TYPE KEY = VAL;

จากตัวอย่างด้านบน เป็นลักษณะของการประกาศตัวแปรแบบกำหนดค่า เมื่ออ่านค่าของตัวแปรออกมา จะได้เป็นค่าที่กำหนดไว้ตอนประกาศ

2.2 การเขียนโปรแกรมภาษา C ด้วย Arduino IDE

ตัวอย่างการประกาศตัวแปร int

```
int i;
int a = 10, b = 20;
```

ตัวอย่างการประกาศตัวแปร int
int i;
int a = 10, b = 20;
จากตัวอย่าง จะเห็นว่าเราสามารถกำหนดชนิดของข้อมูลให้ทีเดียวหลายๆตัวแปรก็ได้ โดยใช้เครื่องหมาย , คั่นไว้

```
int i;
int a = 10, b = 20;
i = a + b;
```

จากตัวอย่างด้านบน เราสามารถกำหนดค่าให้กับตัวแปรเมื่อไรก็ได้ โดยใช้เครื่องหมายเท่ากับ = เป็นตัวเชื่อม ชื่อตัวแปรจะอยู่ทางซ้าย และจะกำหนดค่าเป็นอะไร ให้อยู่ทางขวา ค่าที่อยู่ทางขวาจะถูกนำมาใส่ในค่าที่อยู่ทางซ้ายเสมอ

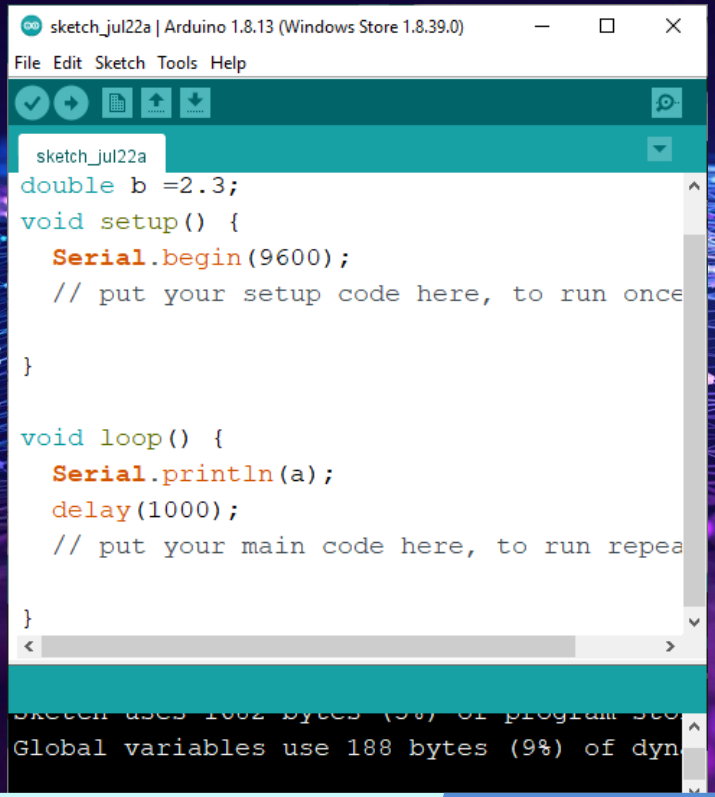
```
int i = 10, a;
a = i;
```

จากตัวอย่างด้านบน จะเห็นว่า a ไม่ได้กำหนดค่าไว้ตอนประกาศ ทำให้ค่าที่อ่านได้จาก a คือ 0 แต่บรรทัดถัดมา มีการกำหนดค่าให้ a เท่ากับ i ซึ่งตอนประกาศ i ได้ประกาศไว้ว่าค่าเท่ากับ 10 เมื่อนำมาใส่ a ค่าที่อ่านได้จาก a จึงเป็น 10 ด้วยเช่นกัน

2.2 การเขียนโปรแกรมภาษา C ด้วย Arduino IDE

Lab01

ตัวอย่าง การประกาศตัวแปรแบบที่ 1



2.2 การเขียนโปรแกรมภาษา C ด้วย Arduino IDE

Lab02

sketch_jul22a | Arduino 1.8.13 (Windows Store 1.8.39.0)

File Edit Sketch Tools Help

sketch_jul22a

```
int a =10;
double b =2.3;
double c = a+b;
void setup() {
  Serial.begin(9600);
  // put your setup code here, to run once:

}

void loop() {
  Serial.println(c);
  delay(1000);
  // put your main code here, to run repeatedly:

}
```

```
Sketch uses 3114 bytes (9%) of program storage spac
Global variables use 204 bytes (9%) of dynamic memo
```

ตัวอย่าง การประกาศตัวแปรแบบที่ 2

COM3

Send

```
12.30
12.30
12.30
12.30
12.30
12.30
```

Autoscroll Show timestamp Newline 9600 baud Clear output

2.2 การเขียนโปรแกรมภาษา C ด้วย Arduino IDE

Lab02

การประกาศตัวแปร boolean และคลาส string

```
boolean is = false;

is = !is;
```

จากตัวอย่างด้านบน มีการประกาศตัวแปร boolean ซึ่งเป็นตัวแปรทางลอจิก มีค่าเป็น True (1) หรือ False (0) ได้เท่านั้น ในบรรทัดแรกได้ประกาศว่าตัวแปร is เป็นตัวแปรชนิด boolean และมีค่าเป็น false หรือลอจิก 0 บรรทัดต่อมา ได้มีการกำหนดให้ is เท่ากับ !is การที่เครื่องหมายนิเสธไปอยู่หน้าตัวแปร หมายถึงการกลับเป็นค่าตรงข้าม จากบรรทัดแรก ตัวแปร is มีค่าเป็น false เมื่อเจอ !is ค่าจึงถูกกลับเป็น true และถูกนำไปเซ็ตในตัวแปร is ทำให้สุดท้ายแล้วตัวแปร is มีค่าเป็น true

```
String text = "Myarduino";
```

จากตัวอย่างด้านบน ได้มีการประกาศตัวแปรชื่อ text เป็นชนิด String เมื่ออ่านค่าที่ได้จาก text จึงได้ค่าออกมาเป็น "Myarduino" เลย

* การกำหนดค่าแบบข้อความให้กับตัวแปร จะต้องอยู่ภายใต้เครื่องหมาย "" เท่านั้น มิฉะนั้นโปรแกรมจะแสดงข้อความผิดพลาดออกมา

2.2 การเขียนโปรแกรมภาษา C ด้วย Arduino IDE

สรุปชนิดตัวแปร Arduino ที่มีการใช้งานบ่อย

1. boolean ใช้เก็บค่าข้อมูล เพียง 2 จำนวน คือ TRUE (จริง) และ FALSE (เท็จ)
2. char ใช้เก็บค่าข้อมูลขนาด 8 บิต ใช้สำหรับเก็บค่ารหัสของตัวอักษร ซึ่งสามารถกำหนดเป็นค่า หรือ เขียนตัวอักษรไว้ภายใต้เครื่องหมาย ฟันเดียวก็ได้ เช่น 'A' หรือ 0x41 หรือ 65
3. byte ใช้เก็บค่าข้อมูลขนาด 8 บิตที่เป็นค่าจำนวนเต็มแบบไม่คิดเครื่องหมาย เหมือนกันกับunsigned char ในภาษาซี) ซึ่งสามารถเก็บค่าข้อมูลได้ 256 ค่า คือ 0-255
4. int หรือ Integer ใช้เก็บค่าข้อมูลขนาด 16บิต ที่เป็นค่าจำนวนเต็ม แบบคิดเครื่องหมาย โดยสามารถใช้เก็บข้อมูลได้ 65536 ค่า คือ -32768 ถึง +32767
5. unsigned int ใช้เก็บค่าข้อมูลขนาด 16บิต ที่เป็นค่าจำนวนเต็ม แบบไม่คิดเครื่องหมาย โดยสามารถใช้เก็บข้อมูลได้ 65536 ค่า คือ 0-65535

2.2 การเขียนโปรแกรมภาษา C ด้วย Arduino IDE

สรุปชนิดตัวแปร Arduino ที่มีการใช้งานบ่อย

6. long ใช้เก็บค่าข้อมูลขนาด 32บิต ที่เป็นค่าเลขจำนวนเต็มแบบคิดเครื่องหมาย โดยสามารถใช้เก็บข้อมูลได้ 4294967296 ค่า คือ -2,147,483,648 ถึง 2,147,483,647

7. unsigned long ใช้เก็บค่าข้อมูลขนาด 32 บิต ที่เป็นค่าเลขจำนวนเต็มแบบไม่คิดเครื่องหมาย โดยสามารถใช้เก็บข้อมูลได้ 4294967296 ค่า คือ 0 ถึง 4,294,967,295

8. float ใช้เก็บค่าข้อมูลที่เป็นเลขทศนิยมแบบคิดเครื่องหมายขนาด 32 บิต โดยสามารถเก็บค่าได้ ระหว่าง 3.4E-38 ถึง 3.4E+38 (-3.4028235E+38 ถึง 3.4028235E+38)

9. double ใช้เก็บค่า เลขทศนิยมเช่นเดียวกันกับ float แต่มีค่าความละเอียดกว่า float ถึง 2 เท่า สามารถเก็บค่าได้มากถึง 1.7E+308

10. void เป็นตัวแปรแบบที่ไม่มีการเก็บค่าใด ๆ คือ ไม่มีค่านั้นเอง

2.2 การเขียนโปรแกรมภาษา C ด้วย Arduino IDE

สรุปชนิดตัวแปร Arduino ที่มีการใช้งานบ่อย

11. arrays เป็นตัวแปรที่ใช้เก็บข้อมูลหลายๆค่าไว้ในตัวแปรตัวเพียงชื่อเดียว แต่มีตัวเลขสำหรับชี้ ตำแหน่งการเก็บข้อมูลต่างกัน โดยตัวเลขที่ใช้ทำหน้าที่เป็นตัวชี้ตำแหน่งของข้อมูล เรียกว่า Index Number โดยค่าลำดับของข้อมูลในตัวแปร array ตำแหน่งแรกจะมีค่าเป็น ศูนย์เสมอ

12. string เป็นตัวแปรใช้เก็บข้อความ หรือ ตัวอักษรหลายๆตัว ซึ่ง string ก็คือ array ของตัวแปร แบบ char นั่นเอง

13. pointer เป็นตัวแปรที่ไม่ได้ใช้เก็บข้อมูล แต่ใช้เก็บค่าตำแหน่งแอดเดรสของหน่วยความจำที่ใช้ สร้างเป็นตัวแปรสำหรับเก็บข้อมูล ซึ่งตัวแปรแบบนี้จะใช้ทำหน้าที่เป็นตัวชี้ไปยังตำแหน่งแอดเดรส ของตัวแปรอื่น ๆ อีกที่หนึ่ง

วิชา อินเทอร์เน็ตของทุกสรรพสิ่ง (Internet of Things : 4123412)

2.3 การใช้งาน Arduino คำสั่งตัดสินใจเลือก if else

2.3 การใช้งาน Arduino คำสั่งตัดสินใจเลือก if else

การใช้งาน Arduino คำสั่งตัดสินใจเลือก if else

คำสั่ง if เป็นคำสั่งใช้สำหรับตรวจสอบเงื่อนไข เพื่อสั่งให้โปรแกรมเลือกทำงานในปีกกาต่าง ๆ ตามผลลัพธ์ที่ได้จากการตรวจสอบเงื่อนไขของคำสั่ง โดยมีรูป2แบบดังนี้

```
รูปแบบที่ 1 เป็นแบบ If
if (เงื่อนไข) {
คำสั่งที่ต้องการให้ทำเมื่อเงื่อนไขเป็นจริง
}
ตัวอย่างคำสั่ง if
int i = 1;
if (i == 1) {
ถ้า i เท่ากับ 1 จะทำใน ปีกกานี้
}
```

2.3 การใช้งาน Arduino คำสั่งตัดสินใจเลือก if else

การใช้งาน Arduino คำสั่งตัดสินใจเลือก if else

รูปแบบที่ 2 เป็นแบบ If...else

```
if (เงื่อนไข) {
คำสั่งที่ต้องการให้ทำเมื่อเงื่อนไขเป็นจริง
} else {
คำสั่งที่ต้องการให้ทำเมื่อเงื่อนไขเป็นจริงเท็จ
}
```

ตัวอย่างคำสั่ง If...else

```
int x = 1 ;
if (x == 1) {
ถ้า x เท่ากับ 1 จะทำในปีกกานี้ ถ้าไม่ใช่จะข้ามไปทำในปีกกา Else
}
else {
ถ้า X ไม่เท่ากับ 1 จะทำในปีกกานี้
}
```

2.3 การใช้งาน Arduino คำสั่งตัดสินใจเลือก if else

คำสั่งที่ใช้ตรวจสอบเงื่อนไข

- == หมายความว่า เท่ากับ
- != หมายความว่า ไม่เท่ากับ
- < หมายความว่า น้อยกว่า
- > หมายความว่า มากกว่า
- <= หมายความว่า น้อยกว่าหรือเท่ากับ
- >= หมายความว่า มากกว่าหรือเท่ากับ
- && หมายความว่า และ
- || หมายความว่า หรือ

2.3 การใช้งาน Arduino คำสั่งตัดสินใจเลือก if else

ตัวอย่างการใช้งาน if

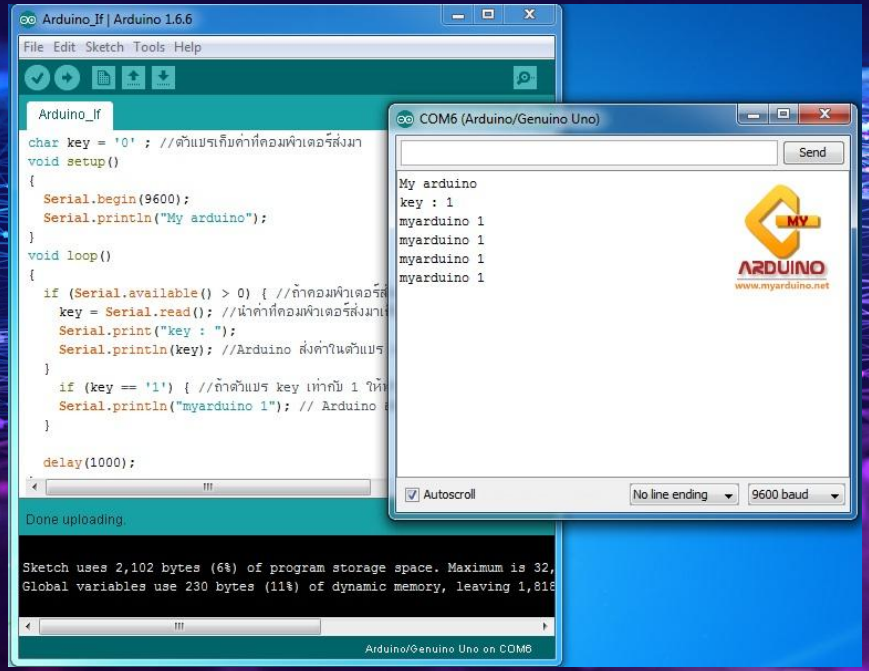
```
char key = '0' ; //ตัวแปรเก็บค่าที่คอมพิวเตอร์ส่งมา
void setup() {
  Serial.begin(9600);
  Serial.println("My arduino");
}
```

```
void loop() {
  if (Serial.available() > 0) { //ถ้าคอมพิวเตอร์ส่งข้อมูลมาให้จะทำใน if นี้
    key = Serial.read(); //นำค่าที่คอมพิวเตอร์ส่งมาเก็บในตัวแปร key
    Serial.print("key : ");
    Serial.println(key); //Arduino ส่งค่าในตัวแปร key เข้าคอมพิวเตอร์ Serial
Monitor
  }
    if (key == '1') { //ถ้าตัวแปร key เท่ากับ 1 ให้ทำในปีกกานี้
    Serial.println("myarduino 1"); // Arduino ส่งข้อความตอบกลับมาทาง Ser
ial Monitor "myarduino 1"
  }
  delay(1000);
}
```

2.3 การใช้งาน Arduino คำสั่งตัดสินใจเลือก if else

ตัวอย่างการใช้งาน if

เป็นการใช้ คำสั่ง if ในการตรวจสอบเงื่อนไข ให้พิมพ์ 1 แล้วกด Send ส่งค่า 1 จากคอมพิวเตอร์มาเก็บไว้ในตัวแปร key แล้วให้ Arduino ตรวจสอบค่า key ถ้าตัวแปล key เท่ากับ 1 ให้ Arduino ส่งตอบกลับมาทาง Serial Monitor ว่า myarduino 1 ตามรูปด้านล่าง



2.3 การใช้งาน Arduino คำสั่งตัดสินใจเลือก if else

ตัวอย่างการใช้งาน if

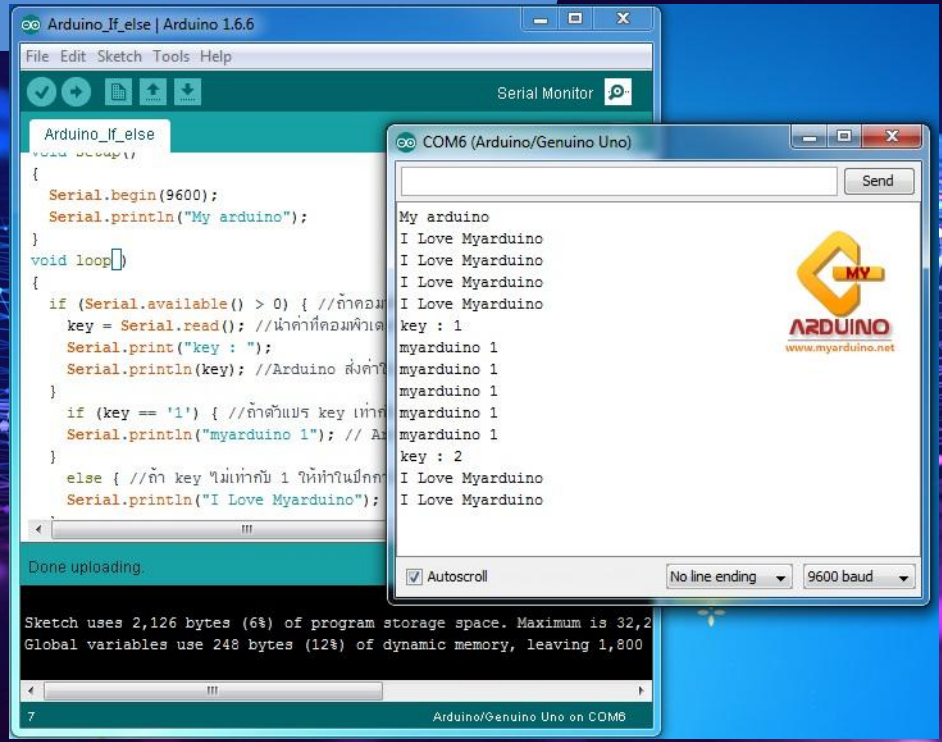
```
char key = '0' ;
//ตัวแปรเก็บค่าที่คอมพิวเตอร์ส่งมา
void setup() {
  Serial.begin(9600);
  Serial.println("My arduino");
}
```

```
void loop() {
  if (Serial.available() > 0) { //ถ้าคอมพิวเตอร์ส่งข้อมูลมาให้จะทำใน if นี้
    key = Serial.read(); //นำค่าที่คอมพิวเตอร์ส่งมาเก็บในตัวแปร key
    Serial.print("key : ");
    Serial.println(key); //Arduino ส่งค่าในตัวแปร key เข้าคอมพิวเตอร์ Serial Monitor
  }
    if (key == '1') { //ถ้าตัวแปร key เท่ากับ 1 ให้ทำในปีกกานี้
    Serial.println("myarduino 1"); // Arduino ส่งข้อความตอบกลับมาทาง Serial Monitor "mya
rduino 1"
  }
    else { //ถ้า key ไม่เท่ากับ 1 ให้ทำในปีกกานี้
    Serial.println("I Love Myarduino");
  } delay(1000);
}
```

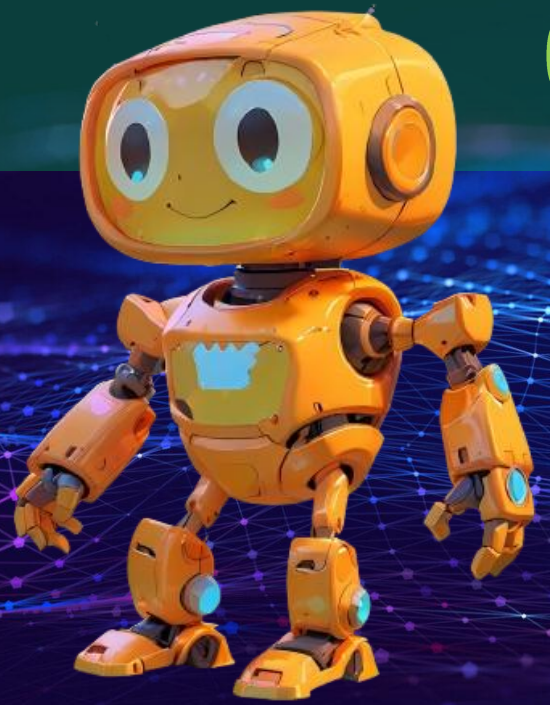
2.3 การใช้งาน Arduino คำสั่งตัดสินใจเลือก if else

ตัวอย่างการใช้งาน if

เป็นการใช้ คำสั่ง if..else ในการตรวจสอบเงื่อนไข ให้พิมพ์ 1 แล้วกด Send ส่งค่า 1 จากคอมพิวเตอร์มาเก็บไว้ในตัวแปร key แล้วให้ Arduino ตรวจสอบค่า key ถ้าตัวแปล key เท่ากับ 1 ให้ Arduino ส่งตอบกลับมาทาง Serial Monitor ว่า myarduino 1 ตามรูปด้านล่าง ต่อมาให้ลอง พิมพ์ 2 แล้วกด Send ถ้าค่า key ไม่เท่ากับ 1 ให้ทำในปีกกา else ให้ Arduino ส่งตอบกลับมาทาง Serial Monitor ว่า I Love Myarduino ตามรูปด้านล่าง



วิชา อินเทอร์เน็ตของทุกสรรพสิ่ง (Internet of Things : 4123412)



2.4 การใช้งาน Arduino คำสั่งทำซ้ำ

2.4 การใช้งาน Arduino คำสั่งทำซ้ำ

การใช้งาน Arduino คำสั่งทำซ้ำ

คำสั่งทำซ้ำ คือ การสั่งให้โปรแกรมทำงานวนซ้ำ ๆ จนกว่าจะไม่อยู่ในเงื่อนไข เช่น การสั่งให้บวกเลขตั้งแต่ 1 – 10 การแสดงข้อความหลาย ๆ ข้อความ ในการใช้คำสั่งการทำซ้ำทุกครั้งจะต้องมีการตั้งเงื่อนไขของการทำงานทุกครั้ง เช่น สร้างตัวแปร x = 1 กำหนดเงื่อนไข ถ้า x มากกว่า 10 ให้ทำการออกจากลูป ถ้าไม่ตรงเงื่อนไข ให้ทำการเพิ่มค่า หรือลดค่า ตามลำดับ โดยการทำซ้ำสามารถแบ่งวิธีการทำซ้ำได้ 2 วิธี คือ

1) การทำซ้ำแบบรู้จำนวนรอบ คือ การใช้งานคำสั่งทำซ้ำโดยที่รู้จำนวนรอบที่แน่นอน คำสั่งที่เหมาะแก่การใช้งานคือ for ยกตัวอย่าง เช่น การทำสูตรคูณตั้งแต่แม่ 1 ถึง แม่ 12
2) การทำซ้ำแบบไม่รู้จำนวนรอบ คือ การใช้งานคำสั่งทำซ้ำ โดยที่ยังไม่ทราบจำนวนรอบที่แน่นนอน คำสั่งที่เหมาะแก่การใช้งานคือ while กับ do while

2.4 การใช้งาน Arduino คำสั่งทำซ้ำ

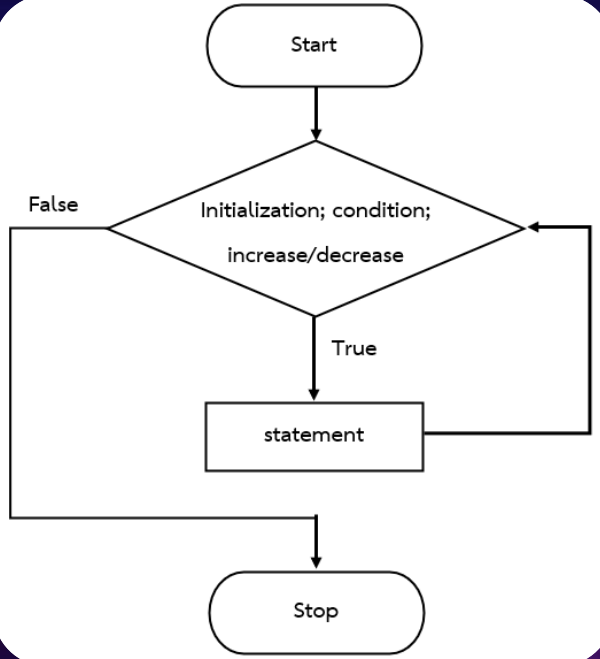
คำสั่ง for

คำสั่ง for คือ คำสั่งทำซ้ำที่จะทำงานวนเป็นลูป ในการใช้งาน for จะเป็นการใช้งานในกรณีที่รู้จำนวนครั้ง หรือจำนวนรอบที่แน่นอน ภายใน for จะมีการกำหนดเงื่อนไขหากยังอยู่ในเงื่อนไขโปรแกรมจะทำงานวนซ้ำจนกว่าจะไม่ได้อยู่ในเงื่อนไข โดย for จะมีองค์ประกอบแบ่งออกเป็น 4 ส่วน ดังนี้

- initialization; คือ ตัวแปรของการทำซ้ำ
- condition; คือ เงื่อนไขที่ให้ออกจากการทำซ้ำ
- increase / decrease คือ คำสั่งที่ใช้ในการนับจำนวน
- statement คือ ชุดคำสั่ง

รูปแบบการเขียน คำสั่งทำซ้ำ for

```
for(initialization; condition; increase / decrease) {
// เขียนโปรแกรมที่นี่
}
```



2.4 การใช้งาน Arduino คำสั่งทำซ้ำ

ตัวอย่างคำสั่งทำซ้ำ for เพื่อแสดงข้อความ www.siam2dev.com 5 ข้อความ

```
void setup() {
  Serial.begin(9600);
  for (int i = 0; i < 5; i++) {
    Serial.println("www.siam2dev.com");
  }
}
void loop() {
 }
```

2.4 การใช้งาน Arduino คำสั่งทำซ้ำ

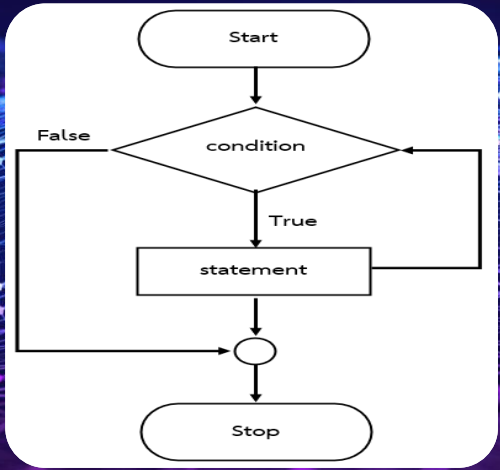
คำสั่ง while

คำสั่ง while คือ คำสั่งทำซ้ำที่ใช้ในกรณีที่ไม่ทราบจำนวนรอบ หรือครั้งในการทำงาน ในคำสั่งทำซ้ำ while จะเป็นการทำงานโดยตรวจสอบเงื่อนไขก่อน ถ้าเงื่อนไขถูกต้องให้ชุดคำสั่งทำงาน ถ้าไม่อยู่ในเงื่อนไขให้หยุดการทำงาน โดย while มีองค์ประกอบแบ่งออกเป็น 2 ส่วน ดังนี้

- condition คือ เงื่อนไขการทำซ้ำ ซึ่งประกอบด้วยตัวแปร หรือค่าคงที่ 2 ตัว และตัวดำเนินการ เพื่อใช้ในการตั้งเงื่อนไข เช่น x < 10 หรือ x < n เป็นต้น
- statement คือ ชุดคำสั่งของ while ภายใน statement จะต้องมีการเพิ่ม หรือลดค่าตัวแปรของ condition เพื่อให้ตัวแปรมีการอัพเดตตามขอบเขตของโปรแกรม

คำสั่งทำซ้ำ while

```
while(condition){
// statement
}
```

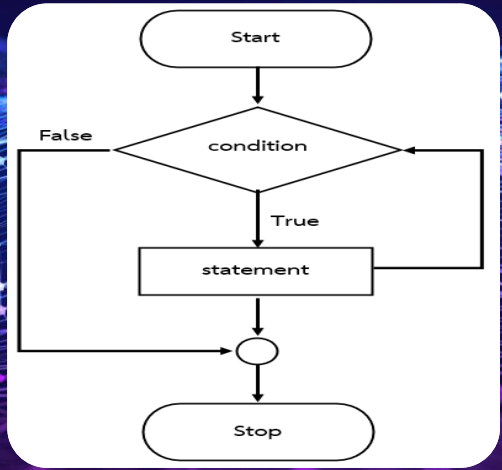


2.4 การใช้งาน Arduino คำสั่งทำซ้ำ

ตัวอย่างคำสั่งทำซ้ำ while เพื่อแสดงข้อความ www.siam2dev.com 5 ข้อความ

ตัวอย่างคำสั่งทำซ้ำ while เพื่อแสดงข้อความ www.siam2dev.net 5 ข้อความ

```
void setup() {
  int i = 0;
  Serial.begin(9600);
  while (i < 5) {
    Serial.println("www.siam2dev.com");
    i++;
  }
}
void loop() {
}
```



2.4 การใช้งาน Arduino คำสั่งทำซ้ำ

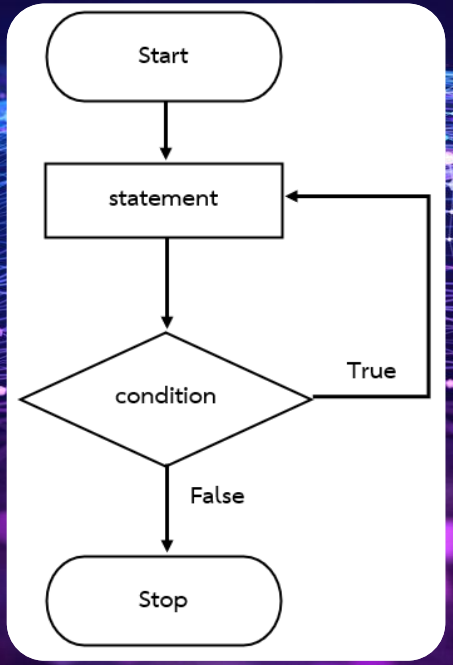
คำสั่ง do while

คำสั่ง do while คือ คำสั่งทำซ้ำที่ใช้ในกรณีที่ไม่ทราบจำนวนรอบเหมือนกับ while แต่จะแตกต่างกันตรงที่ while จะเช็คเงื่อนไขก่อนค่อยทำ แต่ do while จะทำก่อน 1 ครั้งแล้วค่อยกลับมาเช็คเงื่อนไข โดยคำสั่ง do while มีองค์ประกอบแบ่งออกเป็น 3 ส่วนดังนี้

- do คือ คำสั่งทำซ้ำที่มีชุดคำสั่งอยู่ภายใน
- statement คือ ชุดคำสั่งของ do ภายในชุดคำสั่งจะต้องมีการเพิ่ม หรือลดตัวแปรของเงื่อนไข เช่น x++; หรือ x--; เป็นต้น
- condition คือ เงื่อนไขของคำสั่ง do จะประกอบด้วยตัวแปร หรือค่าคงที่ 2 ตัว และตัวกำเนินการ

คำสั่งทำซ้ำ do while

```
do{
// statement
} while (condition)
```



2.4 การใช้งาน Arduino คำสั่งทำซ้ำ

ตัวอย่างคำสั่งทำซ้ำ do while เพื่อแสดงข้อความ www.siam2dev.com 5 ข้อความ

ตัวอย่างคำสั่งทำซ้ำ do while เพื่อแสดงข้อความ www.siam2dev.com 5 ข้อความ

```
void setup() {
  int i = 0;
  Serial.begin(9600);
  do {
    Serial.println("www.siam2dev.com");
    i++;
  } while (i < 5);
 }
void loop() {
}
```

วิชา อินเทอร์เน็ตของทุกสรรพสิ่ง (Internet of Things : 4123412)

2.5 ตัวอย่างโปรแกรม ควบคุมการวิ่งของหลอดไฟ LED

2.5 ตัวอย่างโปรแกรม ควบคุมการวิ่งของหลอดไฟ LED

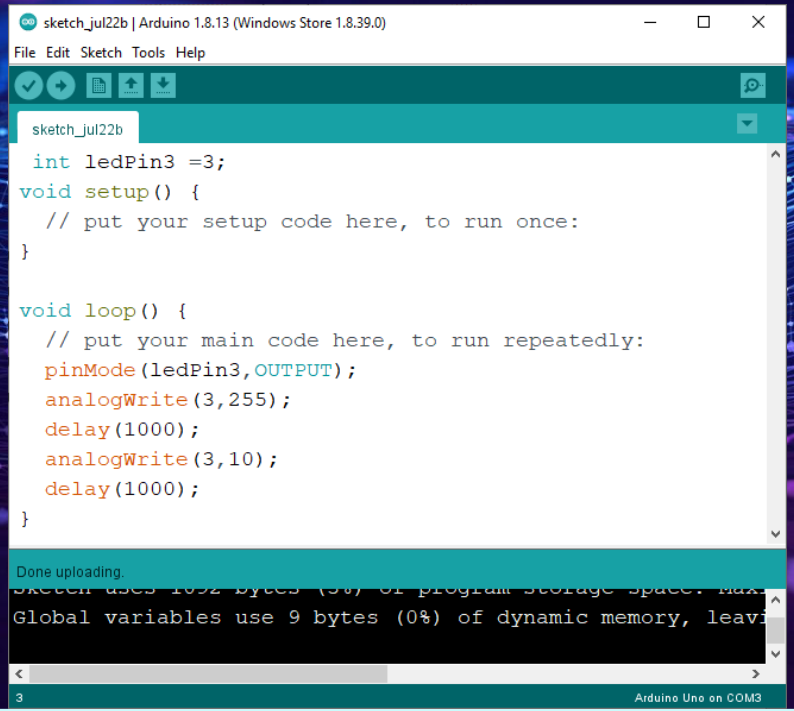
คำสั่ง Digital Write

คำสั่ง Digital Write
เป็นคำสั่งที่ใช้กำหนดสัญญาณ HIGH LOW ของขาดิจิตอลของ Arduino
HIGH คือลอจิก 1 ปล่อยไฟออกแรงดัน 5V
LOW คือลอจิก 0 กำหนดขานั้นให้เป็นกราว์ด 0V
digitalWrite(PiN,Status)
PiN หมายถึง ขา Digital ของ Arduino ที่จะสั่งงาน ให้เป็น HIGH หรือ LOW
Status หมายถึง สถานะ HIGH หรือ LOW

ตัวอย่างคำสั่ง Digital Write
ต้องการให้ขา Digital ขา 13 เป็นสถานะ HIGH
digitalWrite(13,HIGH)

2.5 ตัวอย่างโปรแกรม ควบคุมการวิ่งของหลอดไฟ LED

Lab03



2.5 ตัวอย่างโปรแกรม ควบคุมการวิ่งของหลอดไฟ LED

อุปกรณ์ที่ต้องใช้

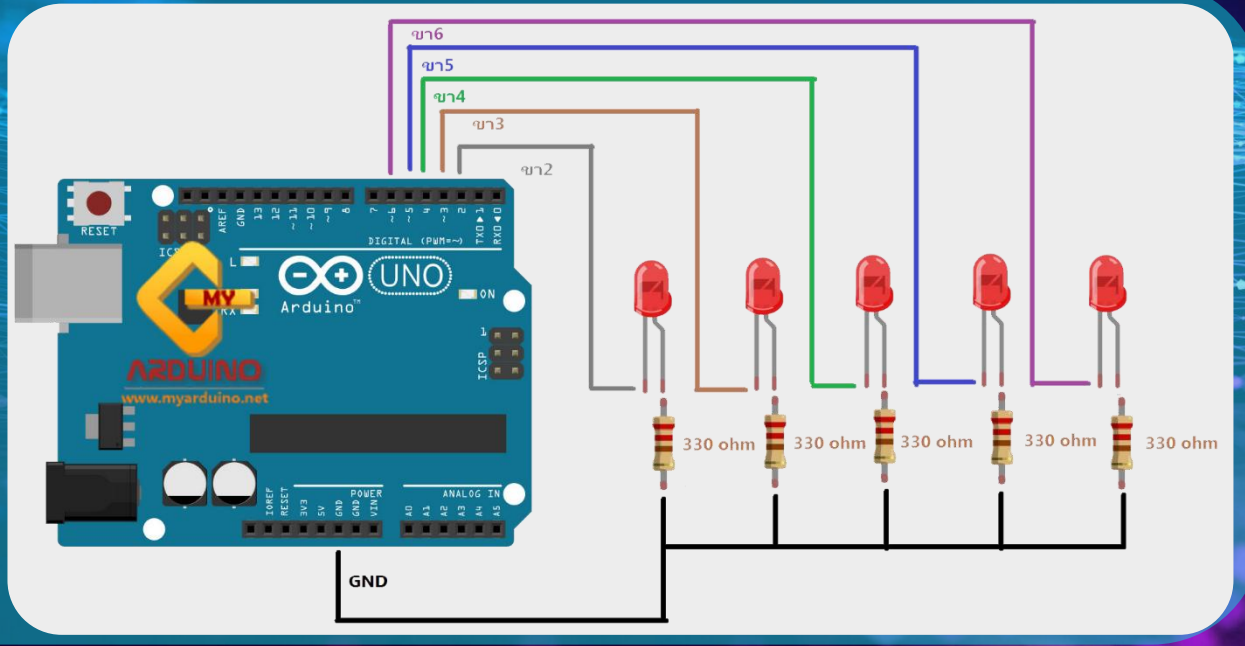
- Arduino UNO R3 พร้อม สายUSB
- LED ขนาด 5mm สีแดง
- บอร์ดทดลอง Breadboard 830 Point
- Resistor ตัวต้านทาน 330 Ohm 1/4W Metal film 1%
- สายไฟจัมเปอร์ ผู้-ผู้

2.5 ตัวอย่างโปรแกรม ควบคุมการวิ่งของหลอดไฟ LED

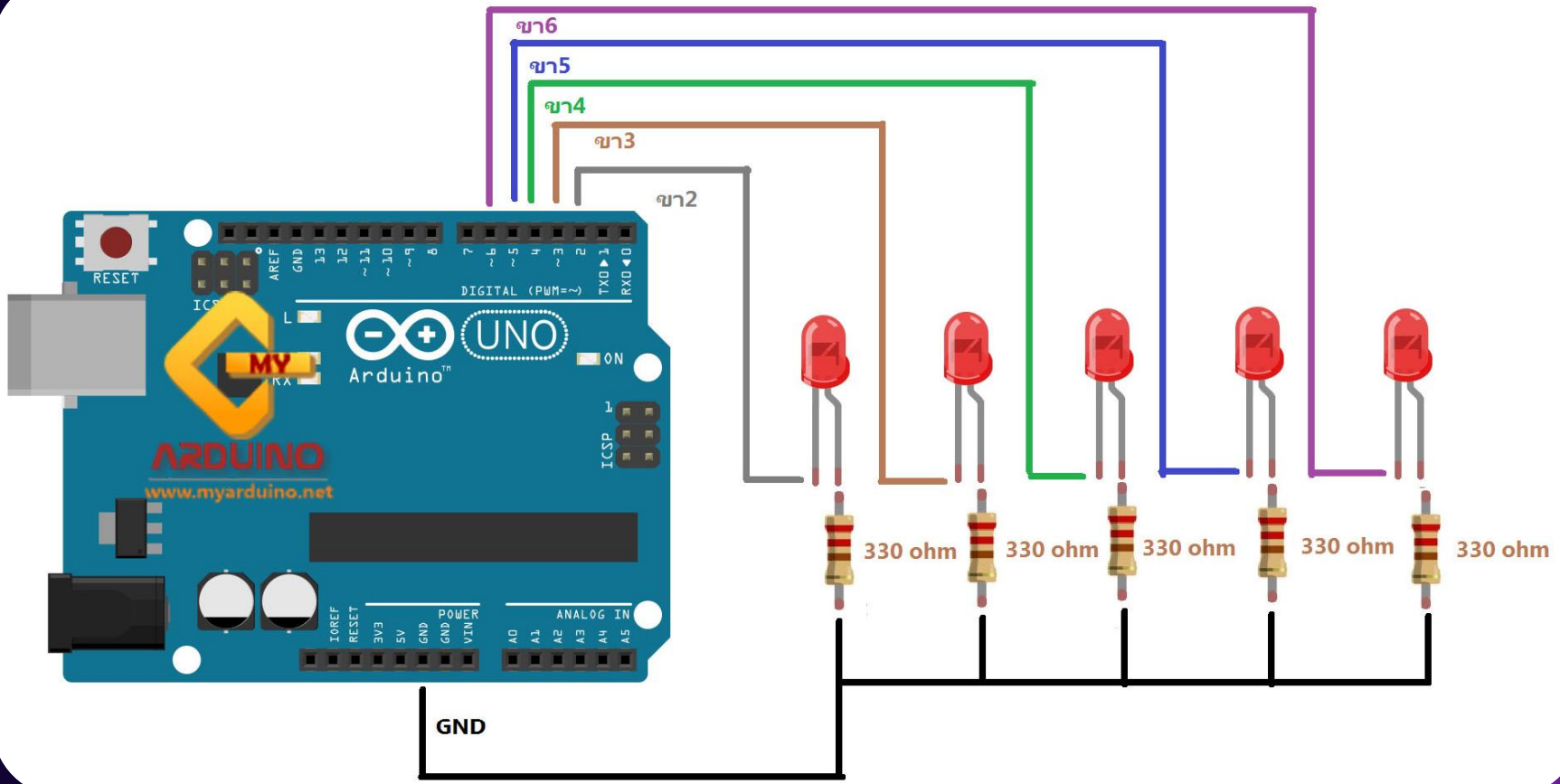
วิธีการต่ออุปกรณ์ การเขียนโปรแกรมเบื้องต้นกับ Arduino สัญญาณ Digital OutPut ควบคุมไฟวิ่ง

Arduino uno r3 -> หลอดไฟ LED

- ขา2 -> LED1
- ขา3 -> LED2
- ขา4 -> LED3
- ขา5 -> LED4
- ขา6 -> LED5



วิธีการต่ออุปกรณ์ การเขียนโปรแกรมเบื้องต้นกับ Arduino สัญญาณ Digital OutPut ควบคุมไฟวิ่ง



2.5 ตัวอย่างโปรแกรม ควบคุมการวิ่งของหลอดไฟ LED

วิธีการต่ออุปกรณ์ การเขียนโปรแกรมเบื่องต้นกับ Arduino สัญญาณ Digital OutPut ควบคุมไฟวิ่ง

1

```
int led1 = 2; // กำหนดขาใช้งาน
int led2 = 3;
int led3 = 4;
int led4 = 5;
int led5 = 6;
void setup()
{
pinMode(led1, OUTPUT); // กำหนดขากำหน้าที่ให้ขา 2 เป็น OUTPUT
pinMode(led2, OUTPUT); // กำหนดขากำหน้าที่ให้ขา 3 เป็น OUTPUT
pinMode(led3, OUTPUT); // กำหนดขากำหน้าที่ให้ขา 4 เป็น OUTPUT
pinMode(led4, OUTPUT); // กำหนดขากำหน้าที่ให้ขา 5 เป็น OUTPUT
pinMode(led5, OUTPUT); // กำหนดขากำหน้าที่ให้ขา 6 เป็น OUTPUT
digitalWrite(led1,LOW);
digitalWrite(led2,LOW);
digitalWrite(led3,LOW);
digitalWrite(led4,LOW);
digitalWrite(led5,LOW);
}
```

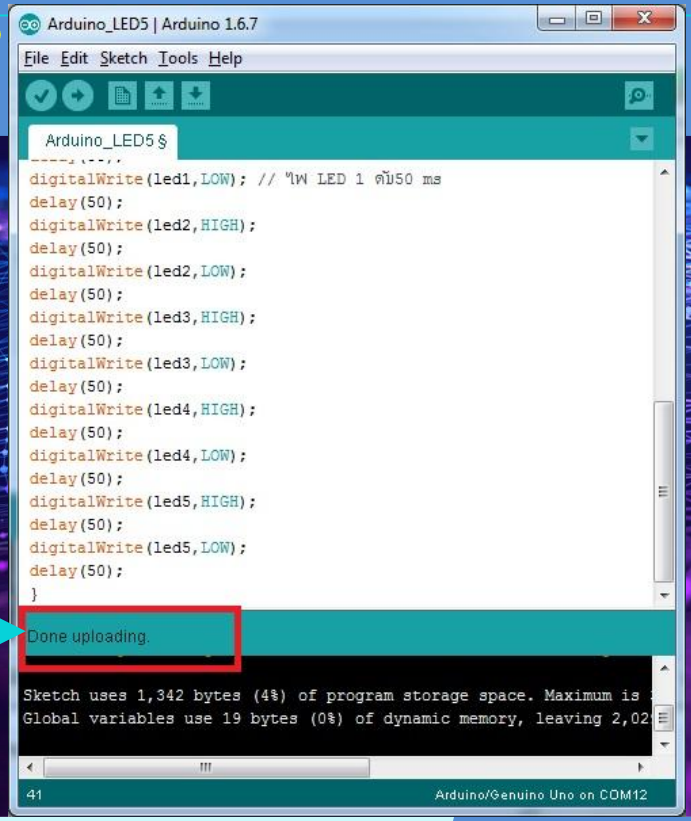
2

```
void loop() {
digitalWrite(led1,HIGH); // ไฟ LED 1 ติด 50 ms
delay(50);
digitalWrite(led1,LOW); // ไฟ LED 1 ดับ50 ms
delay(50);
digitalWrite(led2,HIGH);
delay(50);
digitalWrite(led2,LOW);
delay(50);
digitalWrite(led3,HIGH);
delay(50);
digitalWrite(led3,LOW);
delay(50);
digitalWrite(led4,HIGH);
delay(50);
digitalWrite(led4,LOW);
delay(50);
digitalWrite(led5,HIGH);
delay(50);
digitalWrite(led5,LOW);
delay(50);     }
```

2.5 ตัวอย่างโปรแกรม ควบคุมการวิ่งของหลอดไฟ LED

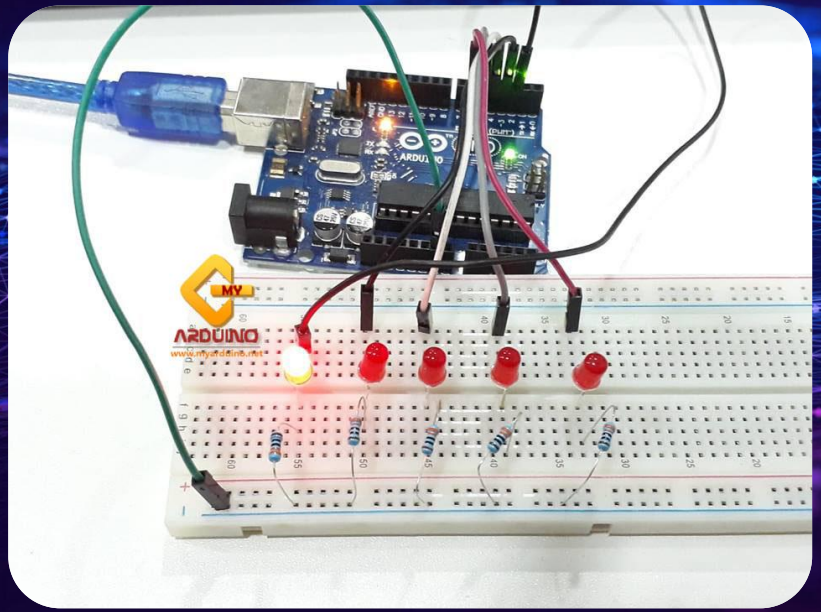
วิธีการต่ออุปกรณ์ การเขียนโปรแกรมเบื่องต้นกับ Arduino ไฟวิ่ง

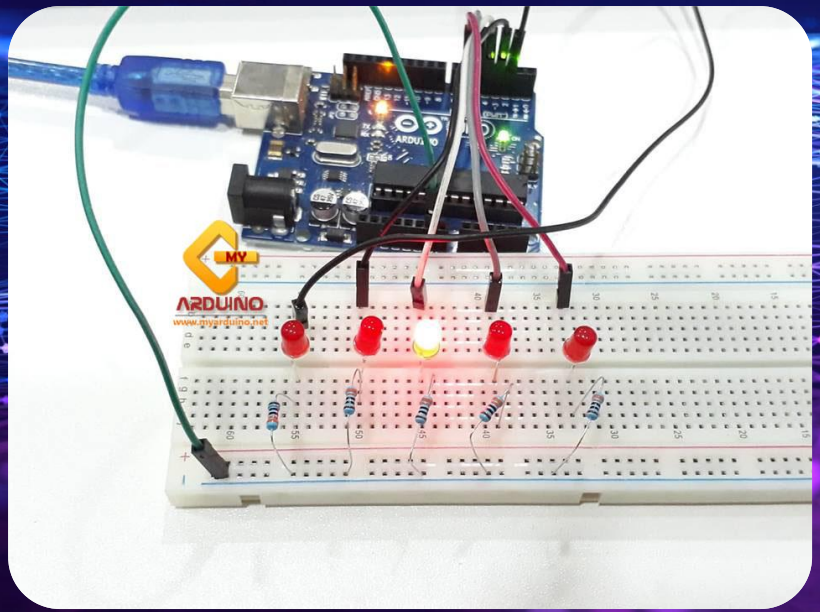
ถ้าอัพโหลดสำเร็จ จะขึ้นตามรูปด้านล่าง



2.5 ตัวอย่างโปรแกรม ควบคุมการวิ่งของหลอดไฟ LED

หลอดไฟ LED จะติดสว่างจากซ้ายไปขวา





2.5 ตัวอย่างโปรแกรม ควบคุมการวิ่งของหลอดไฟ LED

หลอดไฟ LED จะติดสว่างจากซ้ายไปขวา

```
pinMode(led1, OUTPUT); // กำหนดขากำหน้าที่ให้ขา 2 เป็น OUTPUT
pinMode(led2, OUTPUT); // กำหนดขากำหน้าที่ให้ขา 3 เป็น OUTPUT
pinMode(led3, OUTPUT); // กำหนดขากำหน้าที่ให้ขา 4 เป็น OUTPUT
pinMode(led4, OUTPUT); // กำหนดขากำหน้าที่ให้ขา 5 เป็น OUTPUT
pinMode(led5, OUTPUT); // กำหนดขากำหน้าที่ให้ขา 6 เป็น OUTPUT
```

จาก Code เป็นการ ประกาศ ให้ ขา 2 3 4 5 6 ของ arduino uno r3 เป็น Output เพื่อส่งสัญญาณ Digital ไปควบคุม เปิดปิดไฟ LED
การกำหนดหลอดไฟ LED ติด จะใช้คำสั่ง HIGH
digitalWrite(led1,HIGH); สั่งให้ขา2 arduino เป็น HIGH ทำให้ไฟไหลผ่าน LED ลง GND ได้
digitalWrite(ขาของarduinoที่จะควบคุม,ตำแหน่งควบคุมลอจิ H L);
การกำหนดหลอดไฟ LED ดับ จะใช้คำสั่ง LOW
digitalWrite(led1,LOW); สั่งให้ขา2 arduino เป็น LOW ทำให้ไฟไหลผ่าน LED ไม่ได้
เราจะใช้สั่ง delay ในการคงสถานะการติดของไฟ
delay(50); หน่วงเวลา 50ms

5.2 ตัวอย่างโปรแกรม ควบคุมความสว่าง หรี่ไฟ หลอดไฟ LED

วิชา อินเทอร์เน็ตของทุกสรรพสิ่ง (Internet of Things : 4123412)

2.6 ตัวอย่างโปรแกรม ควบคุมความสว่าง หรี่ไฟ หลอดไฟ LED

2.6 ตัวอย่างโปรแกรม ควบคุมความสว่าง หรี่ไฟ หลอดไฟ LED

Arduino สัญญาณ Analog OutPut PWM ควบคุมความสว่าง หรี่ไฟ หลอดไฟ LED

การส่งสัญญาณ Analog OutPut ของ Arduino โดยใช้ขาสัญญาณ PWM 0-5v มาควบคุมความสว่างของหลอดไฟ LED ขาของ Arduino UNO ที่ส่งสัญญาณ PWM ได้จะมีขา 3 5 6 9 จะมีตัวหนอนอยู่ใกล้ๆ

รูปแบบของคำสั่ง: analogWrite

analogWrite(Pin,vaiue)

Pin คือ หมายเลขขาหรือตัวแปรที่แทนขาของ Arduino

value คือ ค่าระดับสัญญาณ Analog Output มีค่าความละเอียด 8 บิต หรือ 256 มีค่าระหว่าง 0-255 ค่าสูงสุดจะให้แรงดันไฟฟ้ามากสุด

ตัวอย่างคำสั่ง analogWrite

analogWrite(9,255)

ต้องการให้ขา 9 ของ Arduino ส่งสัญญาณ PWM 255 ออกมา

2.6 ตัวอย่างโปรแกรม ควบคุมความสว่าง หรี่ไฟ หลอดไฟ LED

อุปกรณ์ที่ต้องใช้

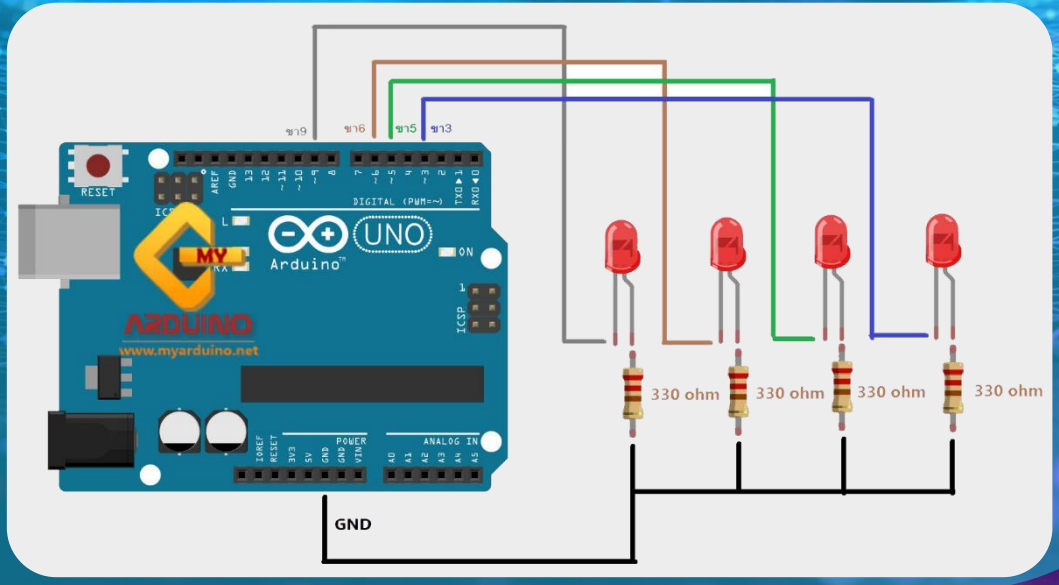
- Arduino UNO R3 พร้อม สายUSB
- LED ขนาด 5mm สีแดง
- บอร์ดทดลอง Breadboard 830 Point
- Resistor ตัวต้านทาน 330 Ohm 1/4W Metal film 1%
- สายไฟจัมเปอร์ ผู้-ผู้

2.6 ตัวอย่างโปรแกรม ควบคุมความสว่าง หรี่ไฟ หลอดไฟ LED

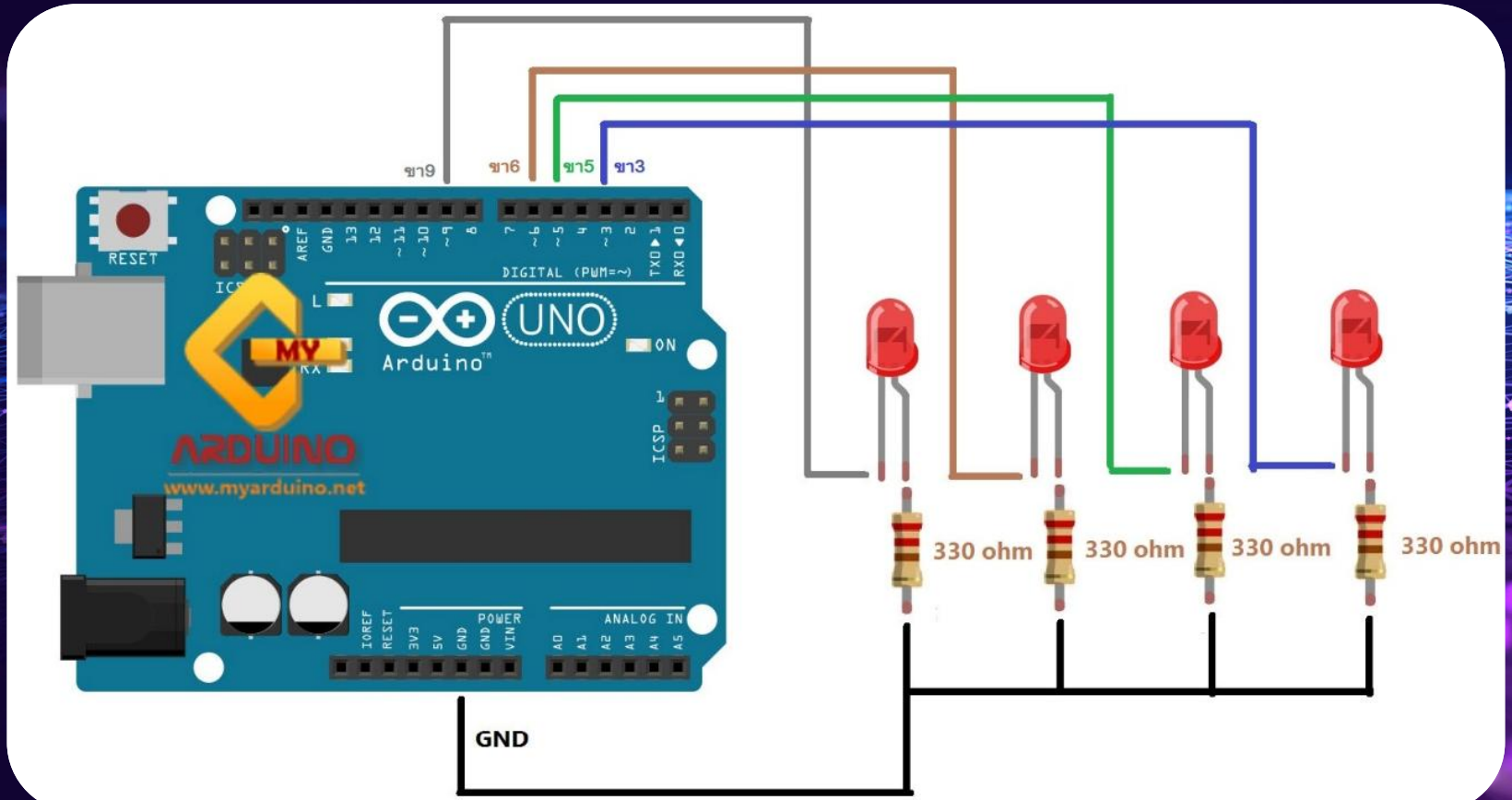
วิธีการต่ออุปกรณ์ บทความ สอนใช้งาน Arduino สัญญาณ Analog OutPut ควบคุมความสว่าง หรี่ไฟ หลอดไฟ LED

Arduino uno r3 -> หลอดไฟ LED

- ขา3 -> LED1
- ขา5 -> LED2
- ขา6 -> LED3
- ขา9 -> LED4



วิธีการต่ออุปกรณ์ บทความ สอนใช้งาน Arduino สัญญาณ Analog OutPut ควบคุมความสว่าง หรี่ไฟ หลอดไฟ LED



2.6 ตัวอย่างโปรแกรม ควบคุมความสว่าง หรี่ไฟ หลอดไฟ LED

วิธีการต่ออุปกรณ์ บทความ สอนใช้งาน Arduino สัญญาณ Analog OutPut ควบคุมความสว่าง หรี่ไฟ หลอดไฟ LED

1
```
int ledPin3 = 3;
int ledPin5 = 5;
int ledPin6 = 6;
int ledPin9 = 9;
void setup() {
  pinMode(ledPin3, OUTPUT);
  pinMode(ledPin5, OUTPUT);
  pinMode(ledPin6, OUTPUT);
  pinMode(ledPin9, OUTPUT);
  Serial.begin(9600);
}
```

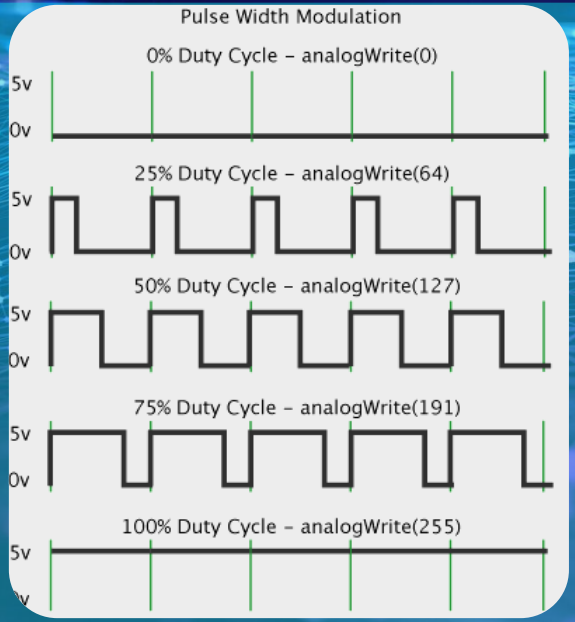
2
```
void loop() {
  for (int x = 0; x < 255; x++) {
    analogWrite(ledPin3, x);
    analogWrite(ledPin5, x);
    analogWrite(ledPin6, x);
    analogWrite(ledPin9, x);
    delay(10);
    Serial.println(x);
  }
```

3
```
delay(500);
  for (int x = 255; x > 0; x--) {
    analogWrite(ledPin3, x);
    analogWrite(ledPin5, x);
    analogWrite(ledPin6, x);
    analogWrite(ledPin9, x);
    delay(10);
    Serial.println(x);
  }
  delay(500);
}
```

2.6 ตัวอย่างโปรแกรม ควบคุมความสว่าง หรี่ไฟ หลอดไฟ LED

วิธีการต่ออุปกรณ์ บทความ สอนใช้งาน Arduino สัญญาณ Analog OutPut ควบคุมความสว่าง หรี่ไฟ หลอดไฟ LED

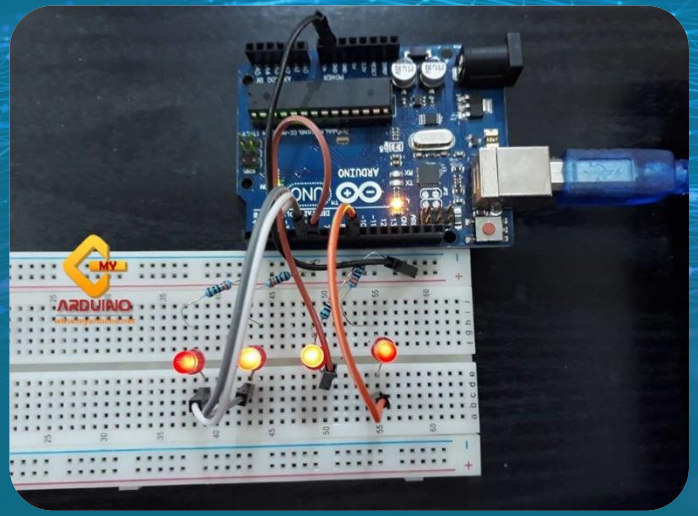
สัญญาณ PWM สามารถกำหนดได้ในช่วง 0-255 ค่า 0 จะเท่ากับ 0v ค่า 255 จะเท่ากับ 5v

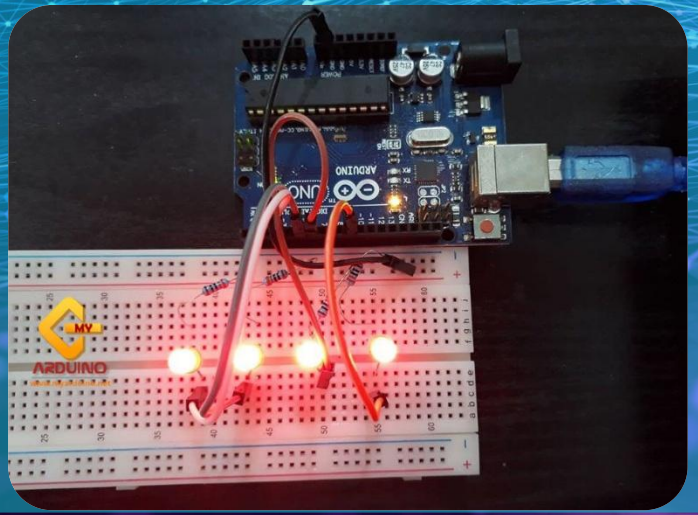


2.6 ตัวอย่างโปรแกรม ควบคุมความสว่าง หรี่ไฟ หลอดไฟ LED

วิธีการต่ออุปกรณ์ บทความ สอนใช้งาน Arduino สัญญาณ Analog OutPut ควบคุมความสว่าง หรี่ไฟ หลอดไฟ LED

- จากตัวอย่างโค้ดจะเห็นว่า ไฟ จะค่อย ๆ สว่างและค่อยๆหรี่จนดับลงตามรูป





2.6 ตัวอย่างโปรแกรม ควบคุมความสว่าง หรี่ไฟ หลอดไฟ LED

วิธีการต่ออุปกรณ์ บทความ สอนใช้งาน Arduino สัญญาณ Analog OutPut ควบคุมความสว่าง หรี่ไฟ หลอดไฟ LED

ในโค้ดตัวอย่างจะใช้ for วนลูปค่อย ๆ ปรับความสว่าง ค่าจะ เพิ่ม และ ลดลง ทีละ 1

```
for (int x = 0; x < 255; x++) { // ให้วนลูป ทั้งหมด 255 ครั้ง 0-254
        analogWrite(ledPin3, x); //นำค่า x มาสั่งงานให้ลอดไฟ LED ค่อยๆสว่าง
        analogWrite(ledPin5, x);
        analogWrite(ledPin6, x);
        analogWrite(ledPin9, x);
        delay(10);
        Serial.println(x);
}
```

อ้างอิง

- http://www.siam2dev.com
- https://www.myarduino.net/
- https://www.entech.co.th/cctv-internet-of-things/?lang=th
- https://www.uih.co.th/th/knowledge/nectec