


| | | | |
|---|----------|-------------------------------|---------------------|
|  | สาขาวิชา | อิเล็กทรอนิกส์ | ใบงานการทดลองที่ 15 |
| | ชื่อวิชา | ไมโครคอนโทรลเลอร์ | |
| | รหัสวิชา | 2105-2105 | หน้าที่ |
| | ชื่องาน | งานโปรแกรมใช้งานอินเตอร์รัพท์ | 140 |

คำชี้แจง ให้ผู้เรียนทุกคนทำการทดลองตามใบงานการทดลองที่ 15 เรื่องโปรแกรมใช้งานอินเตอร์รัพท์ตามขั้นตอนการปฏิบัติงาน

จุดประสงค์ทั่วไป

เพื่อให้มีทักษะการปฏิบัติงานโปรแกรมใช้งานอินเตอร์รัพท์

จุดประสงค์การเรียนรู้เชิงพฤติกรรม (เพื่อให้ผู้เรียน.....)


1. สามารถใช้โปรแกรม Arduino IDE ในการเขียนโปรแกรมภาษา C เบื้องต้นได้อย่างถูกต้อง
2. สามารถใช้งานไมโครคอนโทรลเลอร์ บอร์ด Arduino UNO เบื้องต้นได้อย่างถูกต้อง
3. สามารถประกอบและทดสอบวงจรการใช้งานอินเตอร์รัพท์ได้อย่างถูกต้อง
4. สามารถเขียนโปรแกรมใช้งานอินเตอร์รัพท์ได้อย่างถูกต้อง
5. สามารถประยุกต์ใช้งานไมโครคอนโทรลเลอร์บอร์ด Arduino UNO เบื้องต้นได้อย่างถูกต้อง
6. มีทัศนคติในการแสวงหาความรู้เพิ่มเติม การทำงานด้วยความประณีต รอบคอบและปลอดภัย

เครื่องมือและอุปกรณ์

| | | |
|--|---|---------|
| 1. โปรแกรม Arduino IDE 1.8.4 หรือสูงกว่า | 1 | โปรแกรม |
| 2. สาย USB สำหรับ Arduino Uno | 1 | เส้น |
| 3. ชุดทดลอง Arduino Uno พร้อมสายต่อวงจร | 1 | ชุด |
| 4. เครื่องคอมพิวเตอร์แบบพกพา | 1 | เครื่อง |
| 5. แผงต่อวงจร | 1 | ตัว |
| 6. มัลติมิเตอร์ | 1 | ตัว |
| 7. เครื่องมือประจำตัว | 1 | ชุด |

ข้อห้ามและข้อควรระวัง

1. ไม่เล่นและหยอกล้อกันในเวลาปฏิบัติงาน
2. ควรระวังไม่วางบอร์ด Arduino Uno หรือซีลต่างๆ บนโต๊ะโลหะหรือที่วางที่เป็นโลหะเพราะอาจเกิดการลัดวงจรของภาคจ่ายไฟได้
3. ไม่ควรต่อสายต่อวงจรในบอร์ด Arduino Uno ทิ้งไว้ ควรถอดสายต่อวงจรออกให้หมด เพราะผลการทดลองอาจเกิดการผิดพลาดไม่เป็นไปตามทฤษฎีได้
4. ไม่ควรถอดสายสายโหนด USB เข้าออกตลอดเวลา เพราะอาจทำให้ภาคจ่ายไฟของบอร์ด Arduino Uno เสียหายได้
5. ควรระวังเครื่องมือและอุปกรณ์เสียหายจากการปฏิบัติงานไม่ถูกต้องตามขั้นตอนและไม่ปลอดภัย


| | | | | |
|---|----------|-------------------------------|---------------------|--|
|  | สาขาวิชา | อิเล็กทรอนิกส์ | ใบงานการทดลองที่ 15 | |
| | ชื่อวิชา | ไมโครคอนโทรลเลอร์ | | |
| | รหัสวิชา | 2105-2105 | หน้าที่ | |
| | ชื่องาน | งานโปรแกรมใช้งานอินเตอร์รัพท์ | 141 | |

ทฤษฎี

การขัดจังหวะการทำงานหรือเรียกทับศัพท์ว่าการอินเทอร์รัพท์ (Interrupt) เป็นการขัดจังหวะการทำงานปกติ (ประมวลผลในโปรแกรมหลัก) ของไมโครคอนโทรลเลอร์โดยจะกระโดดไปทำงานในโปรแกรมตอบสนองการอินเทอร์รัพท์ในตำแหน่งที่ตอบสนองการอินเทอร์รัพท์ (Interrupt Vector) ชนิดนั้น ๆ เมื่อทำงานในโปรแกรมตอบสนองการอินเทอร์รัพท์เสร็จสิ้นซีพียูจะกระโดดกลับมาทำงานในตำแหน่งเดิมของโปรแกรมหลักต่อไป ไมโครคอนโทรลเลอร์ในทุกตระกูลจะมีอินเทอร์รัพท์ที่ไม่สามารถปฏิเสธได้ 1 ชนิดได้แก่ Reset กล่าวคือเมื่อซีพียูได้รับสัญญาณอินเทอร์รัพท์ชนิดนี้ไม่ว่าจะทำงานในคำสั่งใดอยู่ก็ตามจะต้องกลับไปทำงานในตำแหน่ง 0x0000 ซึ่งเป็นตำแหน่งแรกของโปรแกรม บอร์ด Arduino ในรุ่นที่ใช้ไมโครคอนโทรลเลอร์ตระกูล AVR เบอร์ Atmega328 สามารถอินเทอร์รัพท์ได้จาก หลายแหล่ง โดยแต่ละแหล่งมีตำแหน่งตอบสนองการอินเทอร์รัพท์ที่แตกต่างกันไปดังตารางที่ 15.1

ตารางที่ 15.1 แสดง Reset and Interrupt Vectors in ATmega328P

| Vector No. | Program Address | Source | Interrupt Definition |
|------------|-----------------|--------------|---|
| 1 | 0x0000 | RESET | External pin, power-on reset, brown-out reset and watchdog system reset |
| 2 | 0x0002 | INT0 | External interrupt request0 |
| 3 | 0x0004 | INT1 | External interrupt request1 |
| 4 | 0x0006 | PCINT0 | Pin change Interrupt request 0 |
| 5 | 0x0008 | PCINT1 | Pin change interrupt request 1 |
| 6 | 0x000A | PCINT2 | Pin change interrupt request 2 |
| 7 | 0x000C | WDT | Watchdog time-out interrupt |
| 8 | 0x000E | TIMER2 COMPA | Timer/Counter2 compare match A |
| 9 | 0x0010 | TIMER2 COMPB | Timer/Counter2 compare match B |
| 10 | 0x0012 | TIMER2 OVF | Timer/Counter2 overflow |
| 11 | 0x0014 | TIMER1 CAPT | Timer/Counter1 capture event |
| 12 | 0x0016 | TIMER1 COMPA | Timer/Counter1 compare match A |
| 13 | 0x0018 | TIMER1 COMPB | Timer/Counter1 compare match B |
| 14 | 0x001A | TIMER1 OVF | Timer/Counter1 overflow |
| 15 | 0x001C | TIMER0 COMPA | Timer/Counter0 compare match A |
| 16 | 0x001E | TIMER0 COMPB | Timer/Counter0 compare match B |
| 17 | 0x0020 | TIMER0 OVF | Timer/Counter0 overflow |
| 18 | 0x0022 | SPI, STC | SPI serial transfer complete |
| 19 | 0x0024 | USART, RX | USART Rx complete |
| 20 | 0x0026 | USART, UDRE | USART, data register empty |
| 21 | 0x0028 | USART, TX | USART, Tx complete |
| 22 | 0x002A | ADC | ADC conversion complete |
| 23 | 0x002C | EE READY | EEPROM ready |
| 24 | 0x002E | ANALOG COMP | Analog comparator |
| 25 | 0x0030 | TWI | 2-wire serial interface |
| 26 | 0x0032 | SPM READY | Store program memory ready |

| | | | | |
|---|----------|-------------------------------|---------------------|--|
|  | สาขาวิชา | อิเล็กทรอนิกส์ | ใบงานการทดลองที่ 15 | |
| | ชื่อวิชา | ไมโครคอนโทรลเลอร์ | | |
| | รหัสวิชา | 2105-2105 | หน้าที่ | |
| | ชื่องาน | งานโปรแกรมใช้งานอินเตอร์รัพท์ | 142 | |

ใบงานนี้เป็นการเรียนรู้การอินเทอร์รัพท์ที่รับการกระตุ้นจากสัญญาณภายนอกซึ่งสามารถรับการอินเทอร์รัพท์ได้ 2 แหล่งคือ INTO (ขา D2) และ INT1 (ขา D3) เนื่องจากใบงานใช้ Arduino รุ่น UNO ในการทดลองแต่สำหรับบอร์ด Arduino ในรุ่นอื่น ๆ สามารถรับสัญญาณอินเทอร์รัพท์จากขาที่แตกต่างกัน ดังตารางที่ 15.2


ตารางที่ 15.2 ขาดิจิทัลที่พร้อมใช้งานอินเทอร์รัพท์ของบอร์ด Arduino ในรุ่นต่าง ๆ

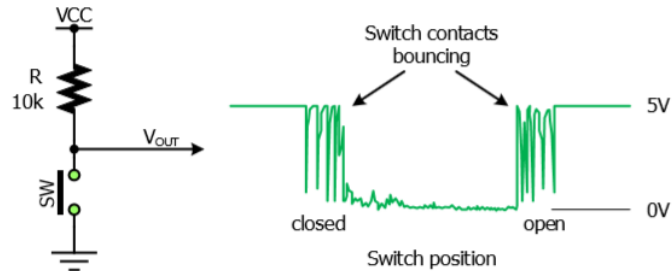
| Board | Digital Pins Usable for Interrupts |
|-----------------------------------|--|
| Uno, Nano, Mini, other 328-based | 2, 3 |
| Mega, Mega2560, MegaADK | 2, 3, 18, 19, 20, 21 |
| Micro, Leonardo, other 32u4-based | 0, 1, 2, 3, 7 |
| Zero | all digital pins, except 4 |
| MKR1000 Rev.1 | 0, 1, 4, 5, 6, 7, 8, 9, A1, A2 |
| Due | all digital pins |
| 101 | all digital pins (Only pins 2, 5, 7, 8, 10, 11, 12, 13 works with CHANGE) |

ตารางที่ 15.3 ชนิดของอินเทอร์รัพท์ของบอร์ด Arduino ในรุ่นต่าง ๆ

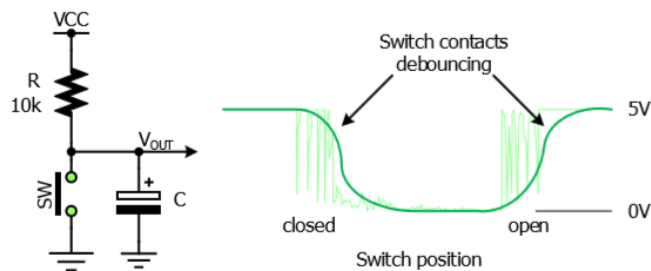
| Board | Int.0 | Int.1 | Int.2 | Int.3 | Int.4 | Int.5 |
|----------------------------------|-------------------------------|-------|-------|-------|-------|-------|
| Uno, Nano, Mini, Ethernet | 2 | 3 | | | | |
| Mega2560 | 2 | 3 | 21 | 20 | 19 | 18 |
| 32u4 based (e.g Leonardo, Micro) | 3 | 2 | 0 | 1 | 7 | |
| Due, Zero, MKR1000, 101 | interrupt number = pin number | | | | | |

Arduino เป็นไมโครคอนโทรลเลอร์ที่ทำงานด้วยความเร็วสูงดังนั้นเมื่อใช้งานวงจรสวิตช์ทั่วไป ดังรูป 15.1 จะทำให้เกิดสัญญาณรบกวนได้ สัญญาณรบกวนนี้เรียกว่าสัญญาณกระดิ่งกระดอนหรือเรียกทับศัพท์ว่าสัญญาณเบาส์ (Bouncing signal) ซึ่งการกดเพียงหนึ่งครั้งจะเกิดสัญญาณรบกวนขึ้นทำให้ไมโครคอนโทรลเลอร์ที่ทำงานด้วยความเร็วสูงเข้าใจว่ากดหลายครั้งส่งผลทำให้เกิดการประมวลผลที่คลาดเคลื่อนไป ดังนั้นเมื่อใช้งานจริงสามารถใช้วิธีการแก้ได้ 2 แบบคือ

| | | | |
|---|----------|-------------------------------|---------------------|
|  | สาขาวิชา | อิเล็กทรอนิกส์ | ใบงานการทดลองที่ 15 |
| | ชื่อวิชา | ไมโครคอนโทรลเลอร์ | |
| | รหัสวิชา | 2105-2105 | หน้าที่ |
| | ชื่องาน | งานโปรแกรมใช้งานอินเตอร์รัพท์ | 143 |



รูปที่ 15.1 วงจรสวิตช์ทั่วไป



รูปที่ 15.2 วงจรสวิตช์ที่ได้รับการแก้ไข

ฟังก์ชัน Arduino ที่ใช้งานในใบงานการทดลอง

1. ฟังก์ชันกำหนดโหมดการทำงานให้กับขาพอร์ต โดยสามารถกำหนดได้ทั้งขาดิจิตอลโดยใส่เพียงตัวเลขของขา (0, 1, 2,...13) และขาแอนาล็อกที่ต้องการให้ทำงานในโหมดดิจิตอลแต่ การใส่ขาต้องใส่ A นำ หน้าซึ่งใช้ได้เฉพาะ A0, A1,...A5 ส่วนขา A6 และ A7 ไม่สามารถใช้งานในโหมดดิจิตอลได้ รูปแบบของฟังก์ชันเป็นดังนี้


`pinMode(pin,mode);`

pin : หมายเลขขาที่ต้องการเซตโหมด, mode : INPUT, OUTPUT, INPUT_PULLUP

2. ฟังก์ชันส่งค่าลอจิกดิจิตอลไปยังขาพอร์ต ค่า HIGH เป็นการส่งลอจิก 1 และค่า LOW เป็นการส่งลอจิก 0 ออกไปยังขาพอร์ต ฟังก์ชันนี้จะทำงานได้ต้องมีการใช้ฟังก์ชัน `pinMode` ก่อน รูปแบบของฟังก์ชันเป็นดังนี้

`digitalWrite(pin,value);`

pin : หมายเลขขาที่ต้องการเขียนลอจิกออกพอร์ต ,value : HIGH หรือ LOW

| | | | |
|---|----------|-------------------------------|---------------------|
|  | สาขาวิชา | อิเล็กทรอนิกส์ | ใบงานการทดลองที่ 15 |
| | ชื่อวิชา | ไมโครคอนโทรลเลอร์ | |
| | รหัสวิชา | 2105-2105 | หน้าที่ |
| | ชื่องาน | งานโปรแกรมใช้งานอินเตอร์รัพท์ | 144 |

3. ฟังก์ชันอ่านค่าลอจิกดิจิทัลที่ขาพอร์ต เป็นการอ่านค่าเข้ามาซึ่งอาจนำมาเก็บไว้ในตัวแปรไว้ตรวจสอบลอจิกที่หลังหรือจะตรวจสอบลอจิกแบบทันทีก็ได้ ฟังก์ชันนี้จะทำงานได้ต้องมี การใช้ฟังก์ชัน pinMode ก่อน รูปแบบของฟังก์ชันเป็นดังนี้

digitalRead(PIN); pin : หมายเลขขาพอร์ตที่ต้องการอ่านลอจิก

ตัวอย่างเช่น value=digitalRead(2); หมายถึง อ่านค่าลอจิกที่ขา D2 มาเก็บไว้ในตัวแปร value if(digitalRead(2)==LOW) หมายถึง ตรวจสอบขา D2 ว่าเป็นลอจิก 0 หรือไม่

4. ฟังก์ชันหน่วงเวลาหรือฟังก์ชันหยุดค้าง การใช้งานสามารถกำหนดตัวเลขของเวลาที่ ต้องการหยุดค้าง ตัวเลขที่ใส่เป็นตัวเลขของเวลาหน่วยเป็นมิลลิวินาที ตัวเลขของเวลาที่ใส่ ได้สูงสุดคือ 4,294,967,295 ซึ่งเป็นขนาดของตัวแปร unsigned long รูปแบบของฟังก์ชันเป็นดังนี้

Delay(ms); ms : ตัวเลขที่หยุดค้างของเวลาหน่วยมิลลิวินาที (unsigned long)

5. ฟังก์ชันกำหนดความเร็วในการสื่อสารทางพอร์ตอนุกรม รูปแบบของฟังก์ชันเป็นดังนี้

Serial.begin(speed); speed: ตัวเลขของอัตราเร็วในการสื่อสารผ่านพอร์ตอนุกรม

6. ฟังก์ชันส่งข้อมูลออกพอร์ต เป็นฟังก์ชันที่ใช้ในการส่งข้อมูลออกทางพอร์ตอนุกรมหรือพิมพ์ข้อมูลออกทางพอร์ตเพื่อแสดงผลที่จอคอมพิวเตอร์ เมื่อพิมพ์เสร็จตัวเคอร์เซอร์จะรออยู่ที่ท้ายสิ่งที่พิมพ์นั้น ๆ รูปแบบของฟังก์ชันเป็นดังนี้

Serial.print(val); Serial.print(val, format);


7. ฟังก์ชันส่งข้อมูลออกพอร์ต คล้ายกับฟังก์ชัน Serial.print ต่างกันตรงที่เมื่อพิมพ์เสร็จตัวเคอร์เซอร์จะขึ้นมารอที่ยบรรทัดใหม่ ดังนั้นเมื่อส่งพิมพ์ครั้งถัดไปข้อมูลที่จะปรากฏจะอยู่ที่บรรทัดใหม่ แทนที่จะต่อท้ายเหมือนกับฟังก์ชัน Serial.print รูปแบบของฟังก์ชันเป็นดังนี้

Serial.println(val); Serial.println(val, format);

ฟังก์ชันใช้งานอินเตอร์รัพท์

Arduino เตรียมฟังก์ชันเกี่ยวกับอินเตอร์รัพท์ให้ใช้งานทั้งหมด 4 ฟังก์ชันด้วยกันคือ

- attachInterrupt() เปิดการใช้งานอินเตอร์รัพท์จากขาอินเตอร์รัพท์ภายนอก
- detachInterrupt() ปิดการใช้งานอินเตอร์รัพท์จากขาอินเตอร์รัพท์ภายนอก
- interrupts() เปิดการใช้งานอินเตอร์รัพท์อีกครั้ง
- noInterrupts() ปิดการใช้งานอินเตอร์รัพท์ทั้งหมด

| | | | |
|---|----------|-------------------------------|---------------------|
|  | สาขาวิชา | อิเล็กทรอนิกส์ | ใบงานการทดลองที่ 15 |
| | ชื่อวิชา | ไมโครคอนโทรลเลอร์ | |
| | รหัสวิชา | 2105-2105 | หน้าที่ |
| | ชื่องาน | งานโปรแกรมใช้งานอินเตอร์รัพท์ | 145 |

1. ฟังก์ชันเปิดการใช้งานอินเตอร์รัพท์จากขาอินเตอร์รัพท์ภายนอก ค่าเริ่มต้นของ Arduino ไม่ได้เปิดการใช้งานส่วนนี้ไว้โดยหาใช้งานได้นำไปใช้งานเป็นขาดิจิทัลปกติ ในทางปฏิบัติ โปรแกรมตอบสนองการอินเตอร์รัพท์จะต้องสั้นเพื่อให้ซีพียูได้ทำงานเสร็จสิ้นด้วยความรวดเร็ว เนื่องจากเมื่อกำลังทำโปรแกรมตอบสนองการอินเตอร์รัพท์อยู่นั้นฟังก์ชันอื่นที่มีการใช้งานอินเตอร์รัพท์จะไม่สามารถใช้งานได้ เช่น delay(), millis() และหากมีการใช้งานตัวแปรที่เป็นตัวแปรโกลบอลจะต้องประกาศด้านหน้าว่า volatile เพื่อให้ค่าที่นำไปใช้งานอินเตอร์รัพท์ได้รับค่าหรือกำหนดค่าเพื่อส่งกลับเข้าโปรแกรมหลักได้อย่างถูกต้อง รูปแบบของฟังก์ชันเปิดการใช้งานนี้มี 2 แบบด้วยกันคือ

- แบบที่กำหนดชื่อขาดิจิทัล

attachInterrupt(digitalPinToInterrupt(pin), ISR, mode);

pin: หมายเลขขาดิจิทัลที่สามารถใช้งานได้เช่น D2 ใส่เฉพาะเลข 2

ISR: ชื่อฟังก์ชันรองที่ใช้ตอบสนองการอินเตอร์รัพท์

mode: เป็นการกำหนดลักษณะของสัญญาณที่ใช้กระตุ้นการอินเตอร์รัพท์

| | |
|---------|---|
| LOW | เมื่อขาเป็นลอจิกศูนย์ |
| CHANGE | เมื่อขามีการเปลี่ยนระดับลอจิก 1->0, 0->1 |
| RISING | เมื่อขามีการเปลี่ยนระดับลอจิกจาก 0 ไปเป็น 1 |
| FALLING | เมื่อขามีการเปลี่ยนระดับลอจิกจาก 1 ไปเป็น 0 |

- แบบที่กำหนดชนิดของอินเตอร์รัพท์


attachInterrupt(interrupt, ISR, mode);

interrupt: หมายเลขขาอินเตอร์รัพท์เช่น INT0(D2) ใส่เฉพาะเลข 0

ISR: ชื่อฟังก์ชันรองที่ใช้ตอบสนองการอินเตอร์รัพท์

mode: เป็นการกำหนดลักษณะของสัญญาณที่ใช้กระตุ้นการอินเตอร์รัพท์

| | |
|---------|---|
| LOW | เมื่อขาเป็นลอจิกศูนย์ |
| CHANGE | เมื่อขามีการเปลี่ยนระดับลอจิก 1->0, 0->1 |
| RISING | เมื่อขามีการเปลี่ยนระดับลอจิกจาก 0 ไปเป็น 1 |
| FALLING | เมื่อขามีการเปลี่ยนระดับลอจิกจาก 1 ไปเป็น 0 |

| | | | | |
|---|----------|-------------------------------|---------------------|--|
|  | สาขาวิชา | อิเล็กทรอนิกส์ | ใบงานการทดลองที่ 15 | |
| | ชื่อวิชา | ไมโครคอนโทรลเลอร์ | | |
| | รหัสวิชา | 2105-2105 | หน้าที่ | |
| | ชื่องาน | งานโปรแกรมใช้งานอินเตอร์รัพท์ | 146 | |

2. ฟังก์ชันปิดการใช้งานอินเตอร์รัพท์จากขาอินเตอร์รัพท์ภายนอก เป็นฟังก์ชันที่ปิดการใช้งานการอินเตอร์รัพท์ในขาอื่น ๆ โดยขาที่ถูกปิดจะกลับไปเป็นขาดีจิทัลดั้งเดิมรูปแบบดังนี้

```
detachInterrupt(digitalPinToInterrupt(pin));
```

```
detachInterrupt(interrupt);
```


pin: หมายเลขขาดีจิทัลที่สามารถใช้งานได้เช่น D2 ใส่เฉพาะเลข 2

interrupt: หมายเลขขาอินเตอร์รัพท์เช่น INT0(D2) ใส่เฉพาะเลข 0

3. ฟังก์ชันเปิดการใช้งานอินเตอร์รัพท์อีกครั้ง เป็นฟังก์ชันที่ใช้เมื่อต้องการเปิดให้มีการอินเตอร์รัพท์ได้อีกครั้งหลังจากการถูกสั่งปิดการอินเตอร์รัพท์จากฟังก์ชัน noInterrupts();
interrupts();

4. ฟังก์ชันปิดการใช้งานอินเตอร์รัพท์ เป็นฟังก์ชันที่ใช้ปิดการอินเตอร์รัพท์ทุกชนิด ดังนั้นเมื่อใช้งานฟังก์ชันนี้แล้วฟังก์ชันอื่น ๆ ที่มีการใช้งานอินเตอร์รัพท์จะใช้งานไม่ได้เช่น delay();
noInterrupts();

[ที่มา:ครูประภาส สุวรรณเพชร,เอกสารประกอบการอบรม เรียนรู้และลองเล่น Arduino เบื้องต้น (ฉบับปรับปรุงครั้งที่ 1) ,หน้าที่ 202-207]

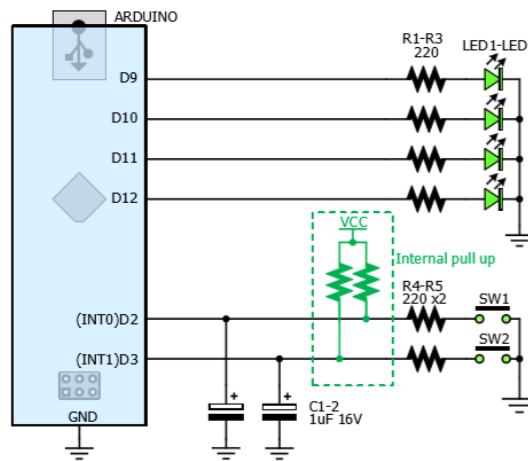
| | | | |
|---|----------|-------------------------------|---------------------|
|  | สาขาวิชา | อิเล็กทรอนิกส์ | ใบงานการทดลองที่ 15 |
| | ชื่อวิชา | ไมโครคอนโทรลเลอร์ | |
| | รหัสวิชา | 2105-2105 | หน้าที่ |
| | ชื่องาน | งานโปรแกรมใช้งานอินเตอร์รัพท์ | 147 |

ลำดับขั้นการทดลอง

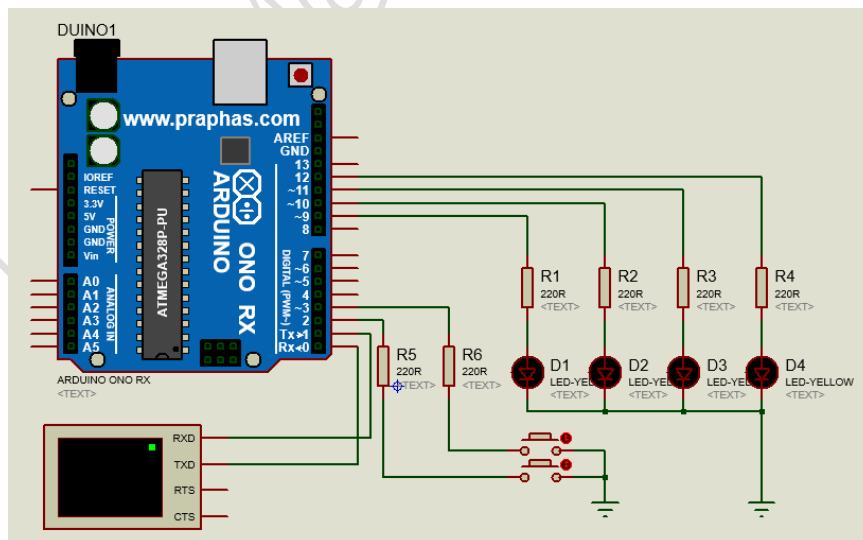
ตอนที่ 1 เขียนโปรแกรมทดสอบฟังก์ชัน noInterrupts(); และ interrupts();

แนวคิดการเรียนรู้ คือ เขียนโปรแกรมทดสอบฟังก์ชัน noInterrupts(); และ interrupts(); โดยการสั่งให้ LED ติดและดับโดยใช้ฟังก์ชันหน่วงเวลาพร้อมแสดงการนับเลขสังเกตผลที่เกิดขึ้นโดยมีขั้นตอนดังนี้


1. ประกอบวงจรการทดสอบฟังก์ชัน noInterrupts(); และ interrupts(); ใช้บอร์ด Arduino UNO ดังรูปที่ 15.3



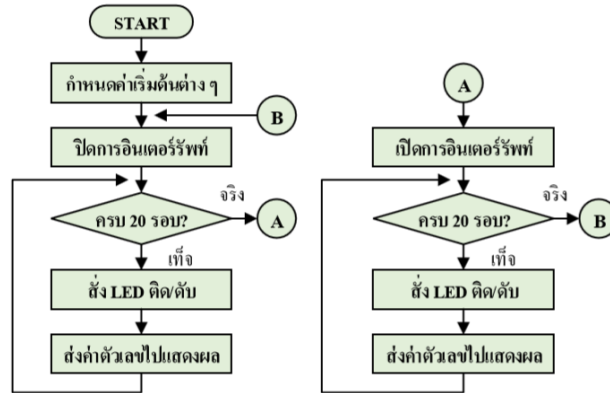
(ก) วงจรไมโครคอนโทรลเลอร์ที่ใช้บอร์ด Arduino Uno



(ข) การต่อวงจรทดลองในโปรแกรมจำลองการทำงานรูปที่ 15.3 แสดงการต่อวงจรการใช้งานอินเตอร์รัพท์

| | | | |
|---|----------|-------------------------------|---------------------|
|  | สาขาวิชา | อิเล็กทรอนิกส์ | ใบงานการทดลองที่ 15 |
| | ชื่อวิชา | ไมโครคอนโทรลเลอร์ | |
| | รหัสวิชา | 2105-2105 | หน้าที่ |
| | ชื่องาน | งานโปรแกรมใช้งานอินเตอร์รัพท์ | 148 |

2. เปิดโปรแกรม Arduino IDE จากนั้นพิมพ์โค้ดโปรแกรมทดสอบฟังก์ชัน noInterrupts(); และ interrupts(); โดยใช้บอร์ด Arduino UNO ตามรูปที่ 15.4 ดังต่อไปนี้



(ก) ฝั่งงาน


```

1 #define LED 12
2 byte state = LOW;
3 void setup() {
4   pinMode(LED, OUTPUT);
5   Serial.begin(9600);
6 }
7 void loop() {
8   noInterrupts();
9   Serial.print("Disable Interrupt :");
10  for(int i=1;i<=20;i++)
11  {
12    Serial.print("-");Serial.print(i);delay(250);
13    state = !state;
14    digitalWrite(LED, state);
15  }
16  Serial.println("");
17  interrupts();
18  Serial.print("Enable Interrupt :");
19  for(int i=1;i<=20;i++)
20  {
21    Serial.print(".");Serial.print(i);delay(250);
22    state = !state;
23    digitalWrite(LED, state);
24  }
25  Serial.println("");
26 }

```

(ข) โค้ดโปรแกรม

รูปที่ 15.4 แสดงโปรแกรมทดสอบฟังก์ชัน noInterrupts(); และ interrupts();

| | | | |
|---|----------|-------------------------------|---------------------|
|  | สาขาวิชา | อิเล็กทรอนิกส์ | ใบงานการทดลองที่ 15 |
| | ชื่อวิชา | ไมโครคอนโทรลเลอร์ | |
| | รหัสวิชา | 2105-2105 | หน้าที่ |
| | ชื่องาน | งานโปรแกรมใช้งานอินเตอร์รัพท์ | 149 |

3. บันทึกไฟล์โค้ด ชื่อ Lab15-1
4. ทำการ Compile โค้ด Lab15-1
5. เชื่อมต่อสาย USB กับ บอร์ด Arduino Uno
6. Upload โปรแกรม Lab15-1 ลงบอร์ด Arduino UNO
7. สังเกตวงจรการทำงานและบันทึกผลการทดลอง

.....

.....

.....

.....

.....


8. คำถามท้ายการทดลองตอนที่ 1 จากโค้ดโปรแกรม Lab15-1 จงตอบคำถามต่อไปนี้

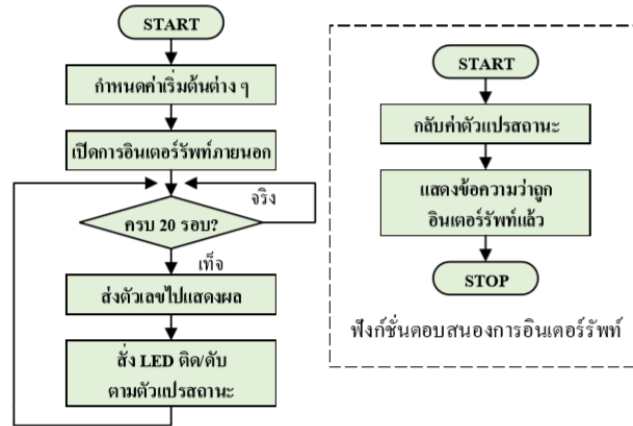
- 8.1. บรรทัดที่ 1,2 ทำหน้าที่.....
- 8.2. บรรทัดที่ 5 ทำหน้าที่.....
- 8.3. บรรทัดที่ 6,7 ทำหน้าที่.....
- 8.4. บรรทัดที่ 11 ทำหน้าที่.....
- 8.5. บรรทัดที่ 12-16 ทำหน้าที่.....
- 8.6. บรรทัดที่ 17 ทำหน้าที่.....
- 8.7. บรรทัดที่ 18 ทำหน้าที่.....
- 8.8. บรรทัดที่ 19 ทำหน้าที่.....

ตอนที่ 2 เขียนโปรแกรมควบคุมการติดดับของ LED ด้วยสวิตช์โดยวิธีอินเตอร์รัพท์

แนวคิดการเรียนรู้ คือ เขียนโปรแกรมควบคุมการติดดับของ LED ด้วยสวิตช์โดยวิธีอินเตอร์รัพท์ ในขณะที่ยัง ไม่มีการกดสวิตช์ให้แสดงผลการนับรอบเพื่อให้รู้ว่ากำลังวนทำงานอยู่ในส่วนใด โดยมีขั้นตอน ดังนี้

9. เปิดโปรแกรม Arduino IDE จากนั้นพิมพ์โค้ดโปรแกรมควบคุมการติดดับของ LED ด้วยสวิตช์โดยวิธีอินเตอร์รัพท์โดยใช้บอร์ด Arduino UNO ตามรูปที่ 15.5 ดังต่อไปนี้

| | | | |
|---|----------|-------------------------------|---------------------|
|  | สาขาวิชา | อิเล็กทรอนิกส์ | ใบงานการทดลองที่ 15 |
| | ชื่อวิชา | ไมโครคอนโทรลเลอร์ | |
| | รหัสวิชา | 2105-2105 | หน้าที่ |
| | ชื่องาน | งานโปรแกรมใช้งานอินเตอร์รัพท์ | 150 |



(ก) ผังงาน


```

1 #define LED 12
2 #define interruptPin 2
3 volatile byte state = LOW;
4
5 void setup()
6 {
7   pinMode(LED, OUTPUT);
8   pinMode(interruptPin, INPUT_PULLUP);
9   attachInterrupt(digitalPinToInterrupt(interruptPin), blink, CHANGE);
10  Serial.begin(9600);
11 }
12 void loop() {
13 // noInterrupts();
14  Serial.print("Test :");
15  for(int i=1;i<=20;i++)
16  {
17    Serial.print("-");
18    Serial.print(i);
19    delay(250);
20    digitalWrite(LED, state);
21  }
22  Serial.println("");
23 }
24 void blink()
25 {
26  state = !state;
27  Serial.println("");
28  Serial.println("Interrupted");
29 }

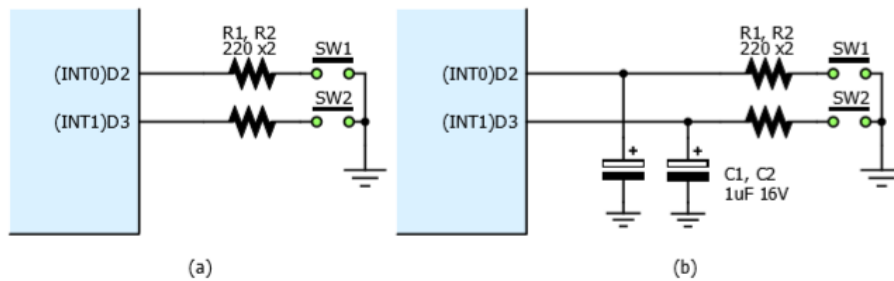
```

(ข) โค้ดโปรแกรม

รูปที่ 15.5 แสดงโปรแกรมควบคุมการติดดับของ LED ด้วยสวิตช์โดยวิธีอินเตอร์รัพท์

| | | | |
|---|----------|-------------------------------|---------------------|
|  | สาขาวิชา | อิเล็กทรอนิกส์ | ใบงานการทดลองที่ 15 |
| | ชื่อวิชา | ไมโครคอนโทรลเลอร์ | |
| | รหัสวิชา | 2105-2105 | หน้าที่ |
| | ชื่องาน | งานโปรแกรมใช้งานอินเตอร์รัพท์ | 151 |


10. บันทึกไฟล์โค้ด ชื่อ Lab15-2
11. ทำการ Compile โค้ด Lab15-2
12. เชื่อมต่อสาย USB กับ บอร์ด Arduino Uno
13. Upload โปรแกรม Lab15-2 ลงบอร์ด Arduino UNO
14. ทดลองกดสวิตช์แล้วสังเกตผลที่เกิดขึ้นบันทึกผลการทดลอง.....
.....
15. ทดลองเปลี่ยนรูปแบบของลักษณะสัญญาณอินเตอร์รัพท์เป็น LOW, CHANGE, RISING, FALLING บันทึกผลการทดลอง.....
.....
16. ทดลองใช้และไม่ใช้วงจรแก้สัญญาณรบกวนดังรูปที่ 15.6 (a) และ (b) สังเกตผลที่เกิดขึ้น



รูปที่ 15.6 แสดงวงจรแก้สัญญาณรบกวน

บันทึกผลการทดลอง.....
.....

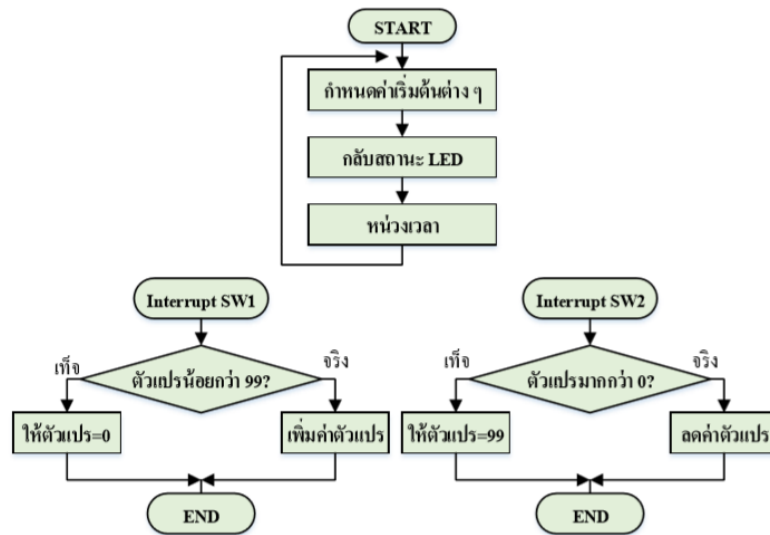
17. คำถามท้ายการทดลองตอนที่ 2 จากโค้ดโปรแกรม Lab14-2 จงตอบคำถามต่อไปนี้
 - 17.1. บรรทัดที่ 1 ทำหน้าที่.....
 - 17.2. บรรทัดที่ 6 ทำหน้าที่.....
 - 17.3. บรรทัดที่ 7 ทำหน้าที่.....
 - 17.4. บรรทัดที่ 8 ทำหน้าที่.....
 - 17.5. บรรทัดที่ 9 ทำหน้าที่.....
 - 17.6. บรรทัดที่ 14-20 ทำหน้าที่.....
 - 17.7. บรรทัดที่ 24-29 ทำหน้าที่.....
.....

| | | | |
|---|----------|-------------------------------|---------------------|
|  | สาขาวิชา | อิเล็กทรอนิกส์ | ใบงานการทดลองที่ 15 |
| | ชื่อวิชา | ไมโครคอนโทรลเลอร์ | |
| | รหัสวิชา | 2105-2105 | หน้าที่ |
| | ชื่องาน | งานโปรแกรมใช้งานอินเตอร์รัพท์ | 152 |

ตอนที่ 3 เขียนโปรแกรมรับสวิตช์ 2 ตัวสำหรับเพิ่มลดตัวเลข

แนวความคิดการเรียนรู้ คือ เขียนโปรแกรมรับสวิตช์ 2 ตัวสำหรับเพิ่มลดตัวเลขพร้อมแสดงค่าที่ จอคอมพิวเตอร์ตลอดเวลาที่โปรแกรมทำงานให้ LED สว่างติดดับสลับกัน โดยมีขั้นตอนดังนี้

18. เปิดโปรแกรม Arduino IDE จากนั้นพิมพ์โค้ดโปรแกรมรับสวิตช์ 2 ตัวสำหรับเพิ่มลดตัวเลขพร้อม แสดงค่าที่จอคอมพิวเตอร์โดยใช้บอร์ด Arduino UNO R3 ตามรูปที่ 15.7 ดังต่อไปนี้




(ก) ผังงาน

```

1 #define LED 12
2 #define interruptPin1 2
3 #define interruptPin2 3
4 volatile int num =0;
5 void setup() {
6   pinMode(LED, OUTPUT);
7   pinMode(interruptPin1, INPUT_PULLUP);
8   pinMode(interruptPin2, INPUT_PULLUP);
9   attachInterrupt(digitalPinToInterrupt(interruptPin1), increment, FALLING);
10  attachInterrupt(digitalPinToInterrupt(interruptPin2), decrement, FALLING);
11  Serial.begin(9600);
12 }
13 void loop() {
14  Serial.println(num);
15  digitalWrite(LED, !digitalRead(LED));
16  delay(250);
17 }

```

| | | | |
|---|----------|-------------------------------|---------------------|
|  | สาขาวิชา | อิเล็กทรอนิกส์ | ใบงานการทดลองที่ 15 |
| | ชื่อวิชา | ไมโครคอนโทรลเลอร์ | |
| | รหัสวิชา | 2105-2105 | หน้าที่ |
| | ชื่องาน | งานโปรแกรมใช้งานอินเตอร์รัพท์ | 153 |

```


18 void increment () {
19     (num<99) ? num++ : num=0;
20 }
21 void decrement () {
22     (num>0) ? num-- : num=99;
23 }

```

(ข) โค้ดโปรแกรม

รูปที่ 15.7 แสดงโปรแกรมรับสวิตซ์ 2 ตัวสำหรับเพิ่มลดตัวเลข

19. บันทึกไฟล์โค้ด ชื่อ Lab15-3
20. ทำการ Compile โค้ด Lab15-3
21. เชื่อมต่อสาย USB กับ บอร์ด Arduino Uno
22. Upload โปรแกรม Lab15-3 ลงบอร์ด Arduino UNO
23. ทดลองกดสวิตซ์แล้วสังเกตผลที่เกิดขึ้นบันทึกผลการทดลอง.....
.....
.....
24. ทดลองเปลี่ยนรูปแบบของลักษณะสัญญาณอินเทอร์รัพท์เป็น LOW, CHANGE, RISING, FALLING บันทึกผลการทดลอง.....
.....
.....
25. คำถามท้ายการทดลองตอนที่ 3 จากโค้ดโปรแกรม Lab15-3 จงตอบคำถามต่อไปนี้
 - 25.1. บรรทัดที่ 1-3 ทำหน้าที่.....
 - 25.2. บรรทัดที่ 2 ทำหน้าที่.....
 - 25.3. บรรทัดที่ 9-10 ทำหน้าที่.....
.....
 - 25.4. บรรทัดที่ 13-17 ทำหน้าที่.....
 - 25.5. บรรทัดที่ 18-20 ทำหน้าที่.....
 - 25.6. บรรทัดที่ 21-23 ทำหน้าที่.....

| | | | |
|---|----------|-------------------------------|---------------------|
|  | สาขาวิชา | อิเล็กทรอนิกส์ | ใบงานการทดลองที่ 15 |
| | ชื่อวิชา | ไมโครคอนโทรลเลอร์ | |
| | รหัสวิชา | 2105-2105 | หน้าที่ |
| | ชื่องาน | งานโปรแกรมใช้งานอินเตอร์รัพท์ | 154 |

ตอนที่ 4 งานที่มอบหมาย

เขียนโปรแกรมไฟวิ่ง LED 4 ตัวโดยกำหนดรูปแบบการติด/ดับตามต้องการพร้อมให้ สามารถรับการอินเตอร์รัพท์ ได้โดยเมื่อกดสวิตช์ SW1 ให้ LED ทุกตัวดับหมดและเมื่อกด สวิตช์ SW2 ให้ LED ทุกตัวติดสว่างทั้งหมดวงจรที่ใช้ทดลองดังรูปที่ 15.3

26. จงเขียนผังงานจากงานที่มอบหมาย

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

- 27. พิมพ์โค้ดโปรแกรมตามผังงานในข้อที่ 26
- 28. บันทึกไฟล์โค้ด ชื่อ Lab15-4
- 29. ทำการ Compile โค้ด Lab15-4
- 30. เชื่อมต่อสาย USB กับ บอร์ด Arduino Uno
- 31. Upload โปรแกรม Lab15-4 ลงบอร์ด Arduino UNO
- 32. สังเกตวงจรการทำงานและบันทึกผลการทดลอง


.....

.....

.....

.....

.....

| | | | |
|---|----------|-------------------------------|---------------------|
|  | สาขาวิชา | อิเล็กทรอนิกส์ | ใบงานการทดลองที่ 15 |
| | ชื่อวิชา | ไมโครคอนโทรลเลอร์ | |
| | รหัสวิชา | 2105-2105 | หน้าที่ |
| | ชื่องาน | งานโปรแกรมใช้งานอินเตอร์รัพท์ | 155 |

33. สรุปผลการทดลอง

.....

.....

.....

.....

.....

.....

.....

.....