	สาขาวิชา	อิเล็กทรอนิกส์	**ใบงานการทดลองที่ 15**
	ชื่อวิชา	ไมโครคอนโทรลเลอร์	
	รหัสวิชา	20105-2105	หน้าที่ 119
	ชื่องาน	งานโปรแกรมใช้งานอินเตอร์รัพท์	

คำชี้แจง ให้ผู้เรียนทุกคนทำการทดลองตามใบงานการทดลองที่ 15 เรื่องโปรแกรมใช้งานอินเตอร์รัพท์ ตามขั้นตอนการปฏิบัติงาน

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. มีกิจนิสัยในการแสวงหาความรู้เพิ่มเติม 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 หรือชีลต่างๆ บนโต๊ะโลหะหรือที่วางที่เป็นโลหะ เพราะอาจเกิดการลัดวงจรของภาคจ่ายไฟได้
3. ไม่ควรต่อสายต่อวงจรในบอร์ด Arduino Uno ทิ้งไว้ ควรถอดสายต่อวงจรออกให้หมด เพราะผล การทดลองอาจเกิดการผิดพลาดไม่เป็นไปตามทฤษฎีได้
4. ไม่ควรถอดสายสายโหลด USB เข้าออกตลอดเวลา เพราะอาจทำให้ภาคจ่ายไฟของบอร์ด Arduino Uno เสียหายได้
5. ควรระวังเครื่องมือและอุปกรณ์เสียหายจากการปฏิบัติงานไม่ถูกต้องตามขั้นตอนและไม่ปลอดภัย

ทฤษฎี

การขัดจังหวะการทำงานหรือเรียกทับศัพท์ว่าการอินเตอร์รัพต์ (Interrupt) เป็นการขัดจังหวะการทำงานปกติ (ประมวลผลในโปรแกรมหลัก) ของไมโครคอนโทรลเลอร์โดยจะกระโดดไปทำงานใน โปรแกรมตอบสนองการอินเตอร์รัพท์ในตำแหน่งที่ตอบสนองการอินเตอร์รัพท์ (Interrupt Vector) ชนิด นั้น ๆ เมื่อทำงานในโปรแกรมตอบสนองการอินเตอร์รัพท์เสร็จสิ้นซีพียูจะกระโดดกลับมาทำงานใน ตำแหน่งเดิมของโปรแกรมหลักต่อไป ไมโครคอนโทรลเลอร์ในทุกตระกูลจะมีอินเตอร์รัพท์ที่ไม่ สามารถปฏิเสธได้ 1 ชนิดได้แก่ Reset กล่าวคือเมื่อซีพียูได้รับสัญญาณอินเตอร์รัพท์ชนิดนี้ไม่ว่าจะทำงานในคำสั่งใดอยู่ก็ตามจะต้องกลับไปทำงานในตำแหน่ง 0x0000 ซึ่งเป็นตำแหน่งแรกของโปรแกรม บอร์ด Arduino ในรุ่นที่ใช้ไมโครคอนโทรลเลอร์ตระกูล AVR เบอร์ Atmega328 สามารถอินเตอร์รัพท์ได้จาก หลายแหล่งโดยแต่ละแหล่งมีตำแหน่งตอบสนองการอินเตอร์รัพท์ที่แตกต่างกันไปดังตารางที่ 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

ใบงานนี้เป็นการเรียนรู้การอินเตอร์รัพท์ที่รับการกระตุ้นจากสัญญาณภายนอกซึ่งสามารถรับการ อินเตอร์รัพท์ได้ 2 แหล่งคือ INT0 (ขา D2) และ INT1 (ขา D3) เนื่องจากใบงานใช้ Arduino รุ่น UNO ใน การทดลองแต่สำหรับบอร์ด Arduino ในรุ่นอื่น ๆ สามารถรับสัญญาณอินเตอร์รัพท์จากขาที่แตกต่างกันดังตารางที่ 15.2

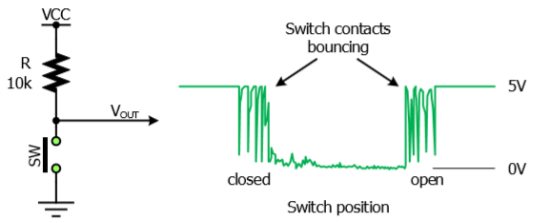
ตารางที่ 15.2 แสดงขาดิจิทัลที่พร้อมใช้งานอินเตอร์รัพท์ของบอร์ด Arduino ในรุ่นต่าง ๆ

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)

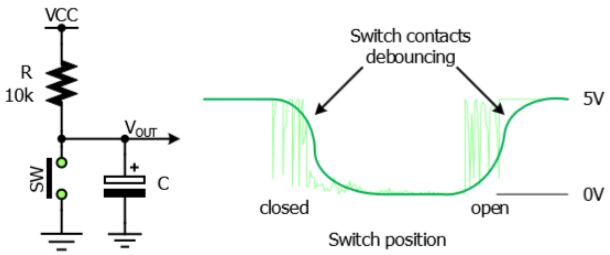
ตารางที่ 15.3 แสดงชนิดของอินเตอร์รัพท์ของบอร์ด Arduino ในรุ่นต่าง ๆ

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 สัญญาณรบกวนนี้เรียกว่าสัญญาณกระเด้งกระดอน หรือ เรียกทับศัพท์ว่าสัญญาณเบาส์ (Bouncing signal) ซึ่งการกดเพียงหนึ่งครั้งจะเกิดสัญญาณรบกวนขึ้นทำให้ไมโครคอนโทรลเลอร์ที่ทำงานด้วยความเร็วสูงเข้าใจว่ากดหลายครั้งส่งผลทำให้เกิดการประมวลผลที่ คลาดเคลื่อนไป ดังนั้นเมื่อใช้งานจริงสามารถใช้วิธีการแก้ได้ 2 แบบคือ



รูปที่ 15.1 แสดงวงจรสวิตช์ทั่วไป



รูปที่ 15.2 แสดงวงจรสวิตช์ที่ได้รับการแก้ไข

ฟังก์ชั่น Arduino ที่ใช้งานในใบงานการทดลอง

1. **ฟังก์ชั่นกำหนดโหมดการทำงานให้กับขาพอร์ต** โดยสามารถกำหนดได้ทั้งขาดิจิทัลโดยใส่เพียงตวัเลขของขา (0, 1, 2,...13) และขาแอนาลอกที่ต้องการให้ทำงานในโหมดดิจิทัลแต่ การใส่ขาต้องใส่ A นำ หน้าซึ่งใช้ได้เฉพาะ A0, A1,...A5 ส่วนขา A6 และ A7 ไม่สามารถใช้งานในโหมดดิจิทัลได้ รูปแบบของฟังก์ชั่นเป็นดังนี้

   pinMode(pin,mode);

   pin : หมายเลขขาที่ต้องการเซตโหมด,mode : INPUT, OUTPUT, INPUT_PULLUP

2. **ฟังก์ชั่นส่งค่าลอจิกดิจิทัลไปยังขาพอร์ต** ค่า HIGH เป็นการส่งลอจิก 1 และค่า LOW เป็นการ ส่งลอจิก 0 ออกไปยังขาพอร์ต ฟังก์ชั่นนี้จะทำงานได้ต้องมีการใช้ฟังก์ชั่น pinMode ก่อน รูปแบบของฟังก์ชั่นเป็นดังนี้

   digitalWrite(pin,value);

   pin : หมายเลขขาที่ต้องการเขียนลอจิกออกพอร์ต ,value : HIGH หรือ LOW

3. **ฟังก์ชั่นอ่านค่าลอจิกดิจิทัลที่ขาพอร์ต** เป็นการอ่านค่าเข้ามาซึ่งอาจนำมาเก็บไว้ในตัวแปรไว้ตรวจสอบลอจิกทีหลังหรือจะตรวจสอบลอจิกแบบทันทีก็ได้ ฟังก์ชั่นนี้จะทำงานได้ต้องมี การใช้ฟังก์ชั่น pinMode ก่อน รูปแบบของฟังก์ชั่นเป็นดังนี้

   digitalRead(PIN); pin : หมายเลขขาพอร์ตที่ต้องการอ่านลอจิก

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

4. **ฟังก์ชั่นหน่วงเวลาหรือฟังก์ชั่นหยุดค้าง** การใช้งานสามารถกำหนดตัวเลขของเวลาที่ต้องการหยุดค้าง ตัวเลขที่ใส่เป็นตัวเลขของเวลาหน่วยเป็นมิลลิวินาที ตัวเลขของเวลาที่ใส่ ได้สูงสุดคือ 4,294,967,295 ซึ่งเป็นขนาดของตวัแปร unsigned long รูปแบบของฟังก์ชั่นเป็นดังนี้

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

5. **ฟังก์ชั่นกำหนดความเร็วในการสื่อสารทางพอร์ตอนุกรม** รูปแบบของฟังก์ชั่นเป็นดังนี้

   Serial.begin(speed); speed: ตัวเลขของอัตราเร็วในการสื่อสารผ่านพอร์ตอนุกรม

6. **ฟังก์ชั่นส่งข้อมูลออกพอร์ต** เป็นฟังก์ชั่นที่ใช้ในการส่งข้อมูลออกทางพอร์ตอนุกรมหรือพิมพ์ข้อมูลออกทางพอร์ตเพื่อแสดงผลที่จอคอมพิวเตอร์ เมื่อพิมพ์เสร็จตัวเคอร์เซอร์จะรออยู่ที่ท้ายสิ่งที่พิมพ์นั้น ๆ รูปแบบของฟังก์ชั่นเป็นดังนี้

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

7. **ฟังก์ชั่นส่งข้อมูลออกพอร์ต** คล้ายกับฟังก์ชั่น Serial.print ต่างกันตรงที่เมื่อพิมพ์เสร็จตัวเคอร์เซอร์จะขึ้นมารอยังบรรทัดใหม่ ดังนั้นเมื่อสั่งพิมพ์ครั้งถัดไปข้อมูลที่ปรากฏจะอยู่ที่บรรทัดใหม่แทนที่จะต่อท้ายเหมือนกับฟังก์ชั่น Serial.print รูปแบบของฟังก์ชั่นเป็นดังนี้

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

ฟังก์ชั่นใช้งานอนิเตอร์รัพท์

Arduino เตรียมฟังก์ชั่นเกี่ยวกับอินเตอร์รัพท์ให้ใช้งานทั้งหมด 4 ฟังก์ชั่นด้วยกันคือ

- attachInterrupt() เปิดการใช้งานอินเตอร์รัพทจ์ากขาอินเตอร์รัพท์ภายนอก
- detachInterrupt() ปิดการใช้งานอินเตอร์รัพทจ์ากขาอินเตอร์รัพท์ภายนอก
- interrupts() เปิดการใช้งานอินเตอร์รัพท์อีกครั้ง
- noInterrupts() ปิดการใช้งานอินเตอร์รัพท์ทั้งหมด

1. **ฟังก์ชั่นเปิดการใช้งานอินเตอร์รัพท์จากขาอินเตอร์รัพท์ภายนอก** ค่าเริ่มต้นของ Arduino ไม่ได้เปิดการใช้งานส่วนนี้ไว้โดยขาใช้งานได้นำไปใช้งานเป็นขาดิจิทัลปกติ ในทางปฏิบัติ โปรแกรมตอบสนองการอินเตอร์รัพท์จะต้องสั้นเพื่อให้ซีพียูได้ทำงานเสร็จสิ้นด้วยความ รวดเร็ว เนื่องจากเมื่อกำลังทำโปรแกรมตอบสนองการอินเตอร์รัพท์อยู่นั้นฟังชั่นอื่นที่มีการใช้งานอินเตอร์รัพท์จะไม่สามารถใช้งานได้เช่น delay(), millis() และหากมีการใช้งานตัวแปรที่เป็นตัวแปรโกลบอลจะต้องประกาศด้านหน้าว่า volatile เพื่อให้ค่าที่นำไปใช้งานอินเตอร์รัพท์ได้รับค่าหรือกำหนดค่าเพื่อส่งกลับเข้าโปรแกรมหลักได้อย่างถูกต้อง รูปแบบ ของฟังก์ชั่นเปิดการใช้งานนี้มี 2 แบบด้วยกันคือ

- แบบที่กำหนดชื่อขาดิจิทัล

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

  pin: หมายเลขขาดิจิทัลที่สามารถใช้งานได้เช่น D2 ใส่เฉพาะเลข 2

  ISR: ชื่อฟังก์ชั่นรองที่ใช้ตอบสนองการอินเตอร์รัพท์

  mode: เป็นการกำหนดลักษณะของสัญญาณที่ใช้กระตุ้นการอินเตอร์รัพท์

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

- แบบที่กำหนดชนิดของอนิเตอร์รัพท์

  attachInterrupt(interrupt, ISR, mode);

  interrupt: หมายเลขขาอินเตอร์รัพท์เช่น INT0(D2) ใส่เฉพาะเลข 0

  ISR: ชื่อฟังก์ชั่นรองที่ใช้ตอบสนองการอินเตอร์รัพท์

  mode: เป็นการกำหนดลักษณะของสัญญาณที่ใช้กระตุ้นการอินเตอร์รัพท์

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

2. **ฟังก์ชั่นปิดการใช้งานอินเตอร์รัพท์จากขาอินเตอร์รัพท์ภายนอก** เป็นฟังก์ชั่นที่ปิดการใช้งาน การอินเตอร์รัพท์ในขานั้น ๆ โดยขาที่ถูกปิดจะกลับไปเป็นขาดิจิทลัดังเดิมรูปแบบดังนี้

   detachInterrupt(digitalPinToInterrupt(pin));

   detachInterrupt(interrupt);

   pin: หมายเลขขาดิจิทัลที่สามารถใช้งานได้เช่น D2 ใส่เฉพาะเลข 2

   interrupt: หมายเลขขาอินเตอร์รัพท์เช่น INT0(D2) ใส่เฉพาะเลข 0

3. **ฟังก์ชั่นเปิดการใช้งานอินเตอร์รัพท์อีกครั้ง** เป็นฟังก์ชั่นที่ใช้เมื่อต้องการเปิดให้มีการอินเตอร์รัพท์ได้อีกครั้งหลังจากการถูกสั่งปิดการอินเตอร์รัพท์จากฟังก์ชั่น noInterrupts(); interrupts();
4. **ฟังก์ชั่นปิดการใช้งานอินเตอร์รัพท์** เป็นฟังก์ชั่นที่ใช้ปิดการอินเตอร์รัพท์ทุกชนิด ดังนั้นเมื่อใช้งานฟังก์ชั่นนี้แล้วฟังก์ชั่นอื่น ๆ ที่มีการใช้งานอินเตอร์รัพท์จะใช้งานไม่ได้เช่น delay(); noInterrupts();

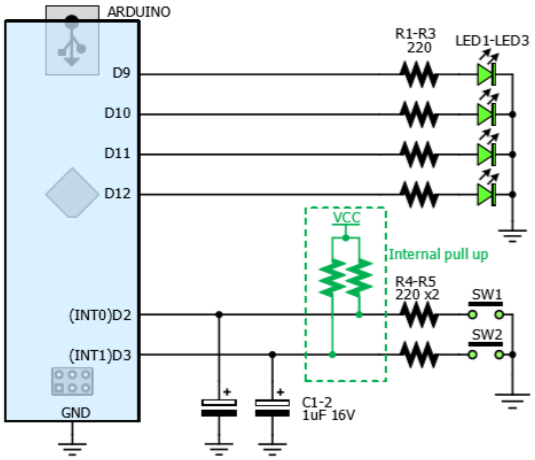
[ที่มา:ครูประภาส สุวรรณเพชร,เอกสารประกอบการอบรม เรียนรู้และลองเล่น Arduino เบื้องต้น **(ฉบับปรับปรุงครั้งที่ 1)** ,หน้าที่ 202-207]

ลำดับขั้นการทดลอง

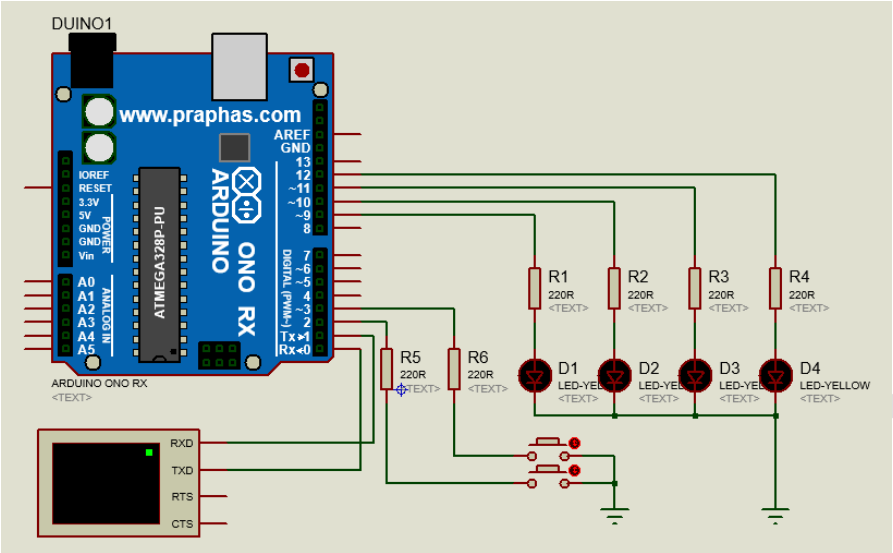
ตอนที่ 1 เขียนโปรแกรมทดสอบฟังก์ชั่น noInterrupts(); และ interrupts();

แนวคิดการเรียนรู้ คือ เขียนโปรแกรมทดสอบฟังก์ชั่น noInterrupts(); และ interrupts(); โดยการสั่งให้ LED ติดและดับโดยใช้ฟังก์ชั่นหน่วงเวลาพร้อมแสดงการนับเลขสังเกตผลที่เกิดขึ้นโดยมีขั้นตอนดังนี้

1. ประกอบวงจรการทดสอบฟังก์ชั่น noInterrupts(); และ interrupts(); ใช้บอรด Arduino UNO ดังรูปที่ 15.3



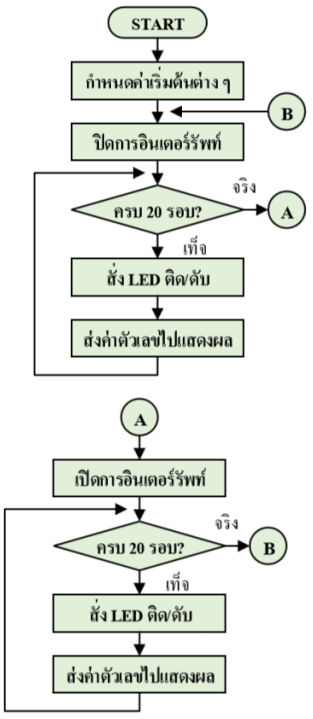
(ก) วงจรไมโครคอนโทรลเลอร์ที่ใช้บอร์ด Arduino Uno



(ข) การต่อวงจรทดลองในโปรแกรมจำลองการทำงาน

รูปที่ 15.3 แสดงการต่อวงจรการใช้งานอินเตอร์รัพท์

2. เปิดโปรแกรม Arduino IDE จากนั้นพิมพ์โค้ดโปรแกรมทดสอบฟังก์ชั่น noInterrupts(); และ interrupts(); โดยใช้บอรด Arduino UNO ตามรูปที่ 15.4 ดังต่อไปนี้




```
#define LED 12
byte state = LOW;
void setup() {
  pinMode(LED, OUTPUT);
  Serial.begin(9600);
}
void loop() {
  noInterrupts();
  Serial.print("Disable Interrupt :");
  for(int i=1;i<=20;i++)
  {
    Serial.print("-");Serial.print(i);delay(250);
    state = !state;
    digitalWrite(LED, state);
   }
  Serial.println("");
  interrupts();
  Serial.print("Enable Interrupt :");
  for(int i=1;i<=20;i++)
  {
    Serial.print(".");Serial.print(i);delay(250);
    state = !state;
    digitalWrite(LED, state);
   }
  Serial.println("");
}
```

(ก) ผังงาน (ข) โค้ดโปรแกรม

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

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

.....

.....

.....

.....

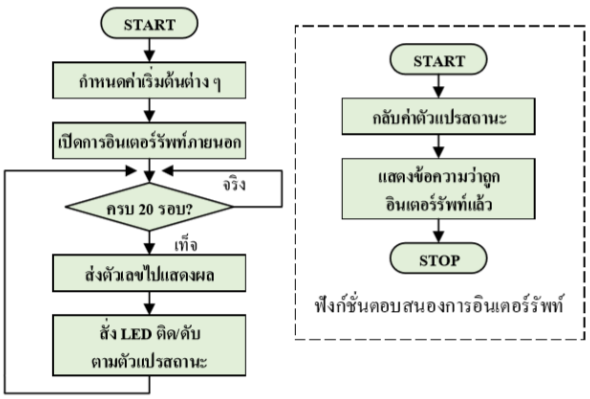
.....

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 ด้วยสวิตช์โดยวิธีอินเตอร์รัพท์ในขณะที่ยัง ไม่มีการกดสวิตช์ให้แสดงผลการนับรอบเพื่อให้รู้ว่ากำลังวนทำงานอยู่ในส่วนใด โดยมีขั้นตอนดังนี้

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



(ก) ผังงาน

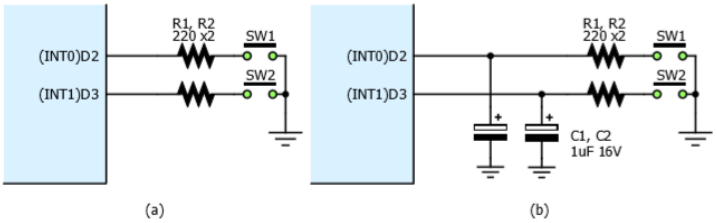
```
#define LED 12
#define interruptPin 2
volatile byte state = LOW;

void setup()
{
  pinMode(LED, OUTPUT);
  pinMode(interruptPin, INPUT_PULLUP);
  attachInterrupt(digitalPinToInterrupt(interruptPin), blink, CHANGE);
  Serial.begin(9600);
}
void loop() {
//  noInterrupts();
  Serial.print("Test :");
  for(int i=1;i<=20;i++)
  {
    Serial.print("-");
    Serial.print(i);
    delay(250);
    digitalWrite(LED, state);
   }
  Serial.println("");
}
void blink()
{
  state = !state;
  Serial.println("");
  Serial.println("Interrupted");
}
```

(ข) โค้ดโปรแกรม

รูปที่ 15.5 แสดงโปรแกรมควบคุมการติดดับของ LED ด้วยสวิตซ์โดยวิธีอินเตอร์รัพท์

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



รูปที่ 15.6 แสดงวงจรแก้สัญญาณรบกวน

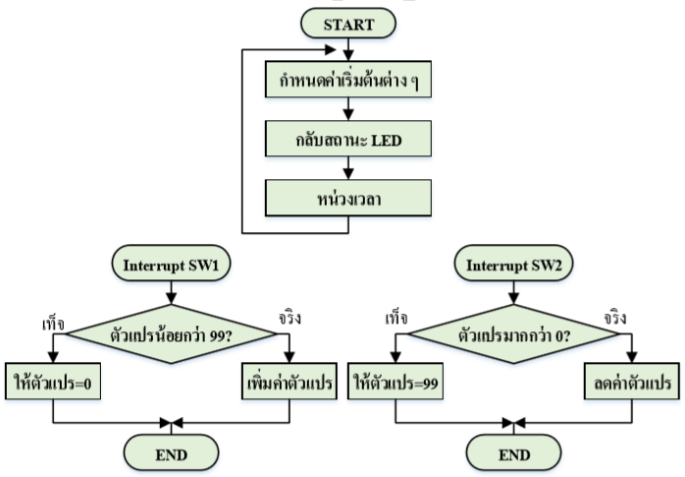
18. บันทึกผลการทดลอง.....
.....

19. คำถามท้ายการทดลองตอนที่ 2 จากโค้ดโปรแกรม Lab14-2 จงตอบคำถามต่อไปนี้
    19.1. บรรทัดที่ 1 ทำหน้าที่.....
    19.2. บรรทัดที่ 6 ทำหน้าที่.....
    19.3. บรรทัดที่ 7 ทำหน้าที่.....
    19.4. บรรทัดที่ 8 ทำหน้าที่.....
    19.5. บรรทัดที่ 9 ทำหน้าที่.....
    19.6. บรรทัดที่ 14-20 ทำหน้าที่.....
    19.7. บรรทัดที่ 24-29 ทำหน้าที่.....
    .....

ตอนที่ 3 เขียนโปรแกรมรับสวิตช์ 2 ตัวสำหรับเพิ่มลดตัวเลข

แนวคิดการเรียนรู้ คือ เขียนโปรแกรมรับสวิตช์ 2 ตัวสำหรับเพิ่มลดตัวเลขพร้อมแสดงค่าที่จอคอมพิวเตอร์ตลอดเวลาที่โปรแกรมทำงานให้ LED สว่างติดดับสลับกัน โดยมีขั้นตอนดังนี้

20. เปิดโปรแกรม Arduino IDE จากนั้นพิมพ์โค้ดโปรแกรมรับสวิตช์ 2 ตัวสำหรับเพิ่มลดตัวเลขพร้อมแสดงค่าที่จอคอมพิวเตอร์โดยใช้บอรด Arduino UNO R3 ตามรูปที่ 15.7 ดังต่อไปนี้



(ก) ผังงาน

```
#define LED 12
#define interruptPin1 2
#define interruptPin2 3
volatile int num =0;
void setup(){
  pinMode(LED, OUTPUT);
  pinMode(interruptPin1, INPUT_PULLUP);
  pinMode(interruptPin2, INPUT_PULLUP);
  attachInterrupt(digitalPinToInterrupt(interruptPin1), increment, FALLING);
  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 }
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 แสดงโปรแกรมรับสวิตช์ 2 ตัวสำหรับเพิ่มลดตัวเลข

21. บันทึกไฟล์โค้ด ชื่อ Lab15-3
22. ทำการ Compile โค้ด Lab15-3
23. เชื่อมต่อสาย USB กับ บอร์ด Arduino Uno
24. Upload โปรแกรม Lab15-3 ลงบอรด Arduino UNO
25. ทดลองกดสวิตช์แล้วสังเกตผลที่เกิดขึ้นบันทึกผลการทดลอง.....
    .....
    .....
26. ทดลองเปลี่ยนรูปแบบของลักษณะสัญญาณอินเตอร์รัพท์เป็น LOW, CHANGE, RISING, FALLING บันทึกผลการทดลอง.....
    .....
    .....
27. คำถามท้ายการทดลองตอนที่ 3 จากโค้ดโปรแกรม Lab15-3 จงตอบคำถามต่อไปนี้
    27.1. บรรทัดที่ 1-3 ทำหน้าที่.....
    27.2. บรรทัดที่ 2 ทำหน้าที่.....
    27.3. บรรทัดที่ 9-10 ทำหน้าที่.....
    .....
    27.4. บรรทัดที่ 13-17 ทำหน้าที่.....
    27.5. บรรทัดที่ 18-20 ทำหน้าที่.....
    27.6. บรรทัดที่ 21-23 ทำหน้าที่.....

ตอนที่ 4 งานที่มอบหมาย

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

28. จงเขียนผังงานจากงานที่มอบหมาย

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

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

.....

.....

.....

.....

.....

35. สรุปผลการทดลอง

.....

.....

.....

.....

.....

.....

.....