

## การทดลองที่ 3HB08

### Arduino “UART, TIMER, and LCD”

#### 1) บทนำ

MCU ตระกูล ATMEGA ซึ่งเป็นตระกูลที่พัฒนาต่อมาจาก 89c51 ที่ได้รับความนิยมเป็นอย่างสูงจากในอดีตโดยได้เพิ่มเติมความสามารถให้กับ MCU จาก 89c51 ไปหลายส่วน เช่น ขนาดของ ROM RAM และ EEPROM เพิ่มเติมนำมาใช้ในการอ่านค่าจากสัญญาณอนาล็อกจากภายนอก “Analog to Digital Converter” และฟังก์ชันในการสร้าง Pulse width modulation และอื่น ๆ อีกหลายส่วน ซึ่งทำให้ MCU ตระกูล ATMEGA ถือได้ว่าเป็น MCU ที่ครบครันที่จะเลือกสรรในการใช้งาน

บอร์ด Arduino กำลังได้รับความนิยมในการพัฒนาการใช้งานประยุกต์ของ Microcontroller เนื่องจากสามารถเรียนรู้ได้ง่าย มีราคาไม่แพง โดย Board Arduino นั้นได้ใช้ MCU ของ ATMEGA เป็น main controller ในการพัฒนาบอร์ดสำเร็จรูปสำหรับการทำงานพร้อมทั้งยังได้มีส่วนสนับสนุนทางด้านการติดต่อสื่อสารระหว่าง software และ Board ผ่าน Serial port (RS-232) และฟังก์ชันพื้นฐานที่สามารถเรียกใช้งานได้อย่างสะดวกสบายและอาจจะเรียกได้ว่าเป็น 3G programming ซึ่งส่งผลให้การพัฒนาความสามารถของ MCU ใช้งานจริงมีความสะดวกรวดเร็วยิ่งขึ้น

#### 2) วัตถุประสงค์

- แนะนำให้ผู้รู้จักการใช้งาน Board Arduino ในการออกแบบพัฒนางจรรวม
- สามารถประยุกต์ใช้งาน Board Arduino ในการแสดงผลการนับสัญญาณนาฬิกา
- สามารถประยุกต์ใช้งาน Board Arduino ในการติดต่อสื่อสารผ่าน UART

#### 3) อุปกรณ์

- LCD ขนาด 16x2 (16 characters 2 lines)
- Arduino board รุ่น Atmega 32
- Variable Resistor (VR) 20K

#### 4) ตัวอย่างการใช้งานฟังก์ชันพื้นฐานก่อนการทดลอง

##### a) ทำการต่อบอร์ดการทดลองเหมือน 3HB03 (Ardunio + LCD)

LCD	Ardunio
RS	digital pin 12
Enable	digital pin 11
D4	digital pin 5
D5	digital pin 4
D6	digital pin 3
D7	digital pin 2

i) เชื่อมโยงขา VDD และ VSS เข้ากับ Power Supply ขนาด 5 volt

ii) เชื่อมโยงขา R/W กับขา Ground

##### b) ตัวอย่างโค้ดในการใช้ Library “LiquidCrystal.h”

```
#include <LiquidCrystal.h>
```

```
LiquidCrystal lcd(12, 11, 5, 4, 3, 2); //pin RS ENABLE D4 D5 D6 D7
```

```
int tmp = 5;
void setup() {
  lcd.begin(16, 2); // initial 16x2
  lcd.print("Reborn");
  lcd.setCursor(0,1); // col 0, row 1
  lcd.print(tmp);
}
```

##### c) ตัวอย่างโค้ดในการติดต่อระบบ BUS UART

```
int incomingByte = 0;
```

```
void setup() {
  Serial.begin(9600); // initialize serial communications at 9600, N, 8, 1, P
}
```

```
void loop(){
  Serial.print(10); // sent 10 to UART
  If(Serial.available() > 0) // ตรวจสอบจำนวน Byte ที่อยู่ใน Receive Buffer ของ UART
  { lcd.print(Serial.Read());
  }
}
```

##### d) ตัวอย่างโค้ดการใช้งาน Timer เพื่อกำหนดฐานเวลา

```
unsigned long time;
```

```
void loop(){
  lcd.print("Time: ");
  time = millis();
  lcd.println(time);
  lcd.setCursor(0,0); // col 0, row 0
}
```

## 5) ทำการทดลอง

- a) ทำการเชื่อมต่อ **Arduinio Board** และทำการแสดงรหัสนักศึกษา บรรทัดที่ 1 เพื่อตรวจสอบความถูกต้องของการติดต่อสื่อสาร

Check Point 1: ดูผล Output บนจอ LCD

- b) **Arduinio Board** และทำการแสดงรหัสผ่าน UART บน PC โดยใช้ Function Serial

Monitor ของ Arduinio Software  เพื่อตรวจสอบความถูกต้องของการติดต่อสื่อสาร

Check Point 2: ดูผล Output บน PC

- c) ทำการเขียนโปรแกรมแสดง เวลา บนหน้าจอ LCD โดยแสดงผลเป็น HH:MM:SS โดยใช้ Function Timer “`millis();`” เป็นฐานเวลา โดยแสดงผลแบบ Real-Time

- i) HH แสดงผลเป็นชั่วโมงแบบ 24 ชั่วโมง
- ii) MM แสดงผลของนาฬิกา
- iii) SS แสดงผลของวินาที

Check Point 3: ดูผล Output บน LCD

- d) ทำการเขียนโปรแกรมติดต่อ UART เพื่อสอบถามเวลาผ่าน โปรแกรม Hyper Terminal ของ Windows โดยมีข้อกำหนดต่อไปนี้
- i) เมื่อผู้ใช้ทำการพิมพ์ “h” ลงบน Hyper Terminal จะเป็นการสอบถามเวลาแบบชั่วโมง ไปยัง Board Arduinio ซึ่งจะทำให้การแสดงผลเวลาของ HH แสดงออกทาง Serial “Hours:HH”
  - ii) เมื่อผู้ใช้ทำการพิมพ์ “m” ลงบน Hyper Terminal จะเป็นการสอบถามเวลาแบบชั่วโมง ไปยัง Board Arduinio ซึ่งจะทำให้การแสดงผลเวลาของ MM แสดงออกทาง Serial “Mintues:MM”
  - iii) เมื่อผู้ใช้ทำการพิมพ์ “s” ลงบน Hyper Terminal จะเป็นการสอบถามเวลาแบบชั่วโมง ไปยัง Board Arduinio ซึ่งจะทำให้การแสดงผลเวลาของ SS แสดงออกทาง Serial “Second:SS”

Check Point 4: ดูผลการทำงาน