ใบงานที่ 8

วิชา ไมโครคอนโทรลเลอร์เบื้องต้น **รหัสวิชา** 20104-2112

ชื่อหน่วย คำสั่งในการเขียนโปรแกรมควบคุมอุปกรณ์ภายนอกงานเชื่อมต่อบอร์ดคอนโทรลเลอร์กับอุปกรณ์อินพุต เอาต์พุต (เครื่องมือพัฒนาโปรแกรม)

1. จุดประสงค์การเรียนรู้

1.1 อธิบายการเขียนโปรแกรมด้วยภาษา C++ ได้ **(ด้านความรู้)**

1.2 สามารถเขียนโปรแกรมด้วยภาษา C++ เบื้องต้นได้ **(ด้านทักษะ)**

1.3 ใช้วัสดุและอุปกรณ์อย่างเหมาะสมคุ้มค่ามากที่สุด ดูแลรักษาเครื่องมือ ยืดอายุการใช้งานของเครื่องมือและอุปกรณ์ **(ด้านคุณธรรมจริยธรรม)**

2. จุดประสงค์เชิงพฤติกรรม/บูรณาการเศรษฐกิจพอเพียง

2.1. ใช้ความรู้นำไปประยุคใช้งานด้านอิเล็กทรอนิกส์ได้อย่างถูกต้องและคุ้มค่า**(ด้านความรู้)**

2.2. ปฏิบัติงานได้ถูกต้องและสำเร็จภายในเวลาที่กำหนดอย่างมีเหตุผล **(ด้านทักษะ)**

2.3. เตรียมความพร้อมด้านวัสดุอุปกรณ์สอดคล้องกับงานและใช้วัสดุอุปกรณ์อย่างคุ้มค่าประหยัด ตามหลักปรัชญาของเศรษฐกิจพอเพียง **(ด้านคุณธรรมจริยธรรม)**

เครื่องมือ/อุปกรณ์

1. บอร์ด Arduino Uno R3 พร้อมสาย Upload
2. LED จำนวน 1 ดวง
3. ความต้านทาน 330 Ω 1 ตัว
4. ความต้านทานแบบปรับค่าได้ 10 KΩ 1 ตัว
5. สายไฟ จัมเปอร์
6. คอมพิวเตอร์ PC หรือ Note Book 1 เครื่อง
7. มอเตอร์ DC 12V 1 ตัว

รายการสอน

เครื่องมือพัฒนาโปรแกรม

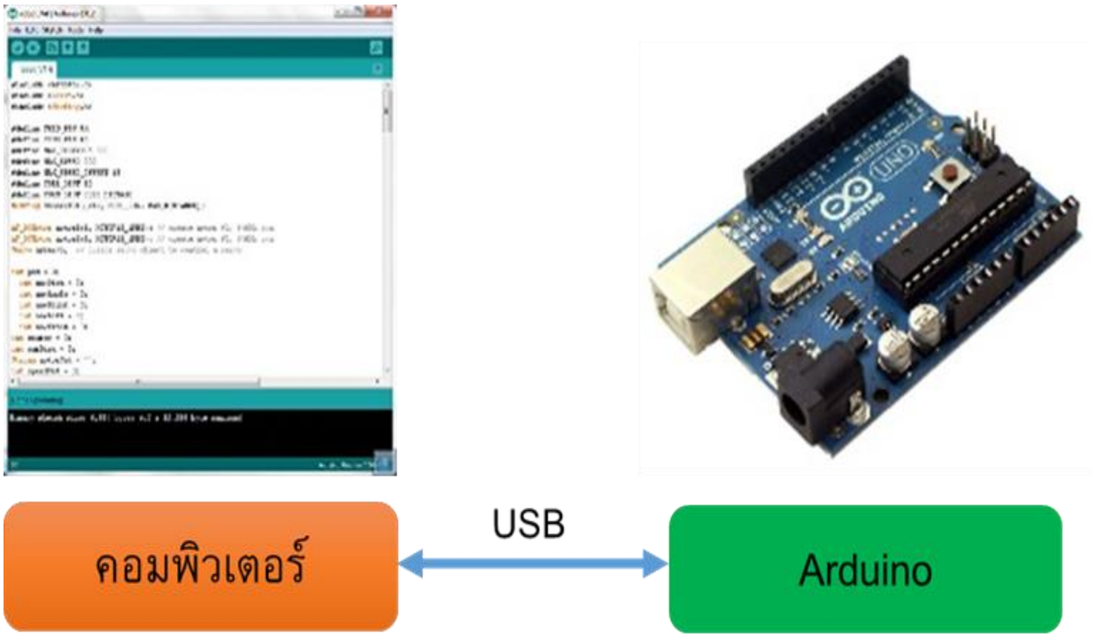
ทางที่ง่ายที่สุดที่คุณจะสามารถเขียนโปรแกรมในภาษา C++ ได้นั้นคือการใช้ IDE IDE เป็นการรวบรวมชุดโปรแกรมที่จำเป็นสำหรับในการพัฒนาโปรแกรม มันเป็นโปรแกรมที่อำนวยความสะดวกและให้เครื่องมือที่จำเป็นสำหรับในการพัฒนาโปรแกรม โดยปกติแล้ว IDE จะประกอบไปด้วยตัวที่ใช้แก้ไขและพิมพ์โค้ด ที่สร้างมากับเครื่องมืออัตโนมัติและตัวดีบักโปรแกรม

สำหรับในบทเรียนนี้ โปรแกรมที่เป็นที่นิยมที่สุดที่เราจะแนะนำคือ Code blocks มันสามารถใช้ได้บนแพลตฟอร์มต่างๆ เช่น Windows Linux และ MacOS ซึ่งมากับคอมไพเลอร์ GCC (MingW / GNU GCC) MSVC++ clang Digital Mars Borland C++ 5.5 Open Watcom และอื่นๆ Code blocks นั้นสนับสนุนการเขียนทั้งภาษา C++ และภาษา C

คุณสามารถดาวน์โหลด Code blocks ได้ที่เว็บไซต์ทางการของมันและเลือกที่ตรงกับแพลตฟอร์มของคุณ http://www.codeblocks.org/

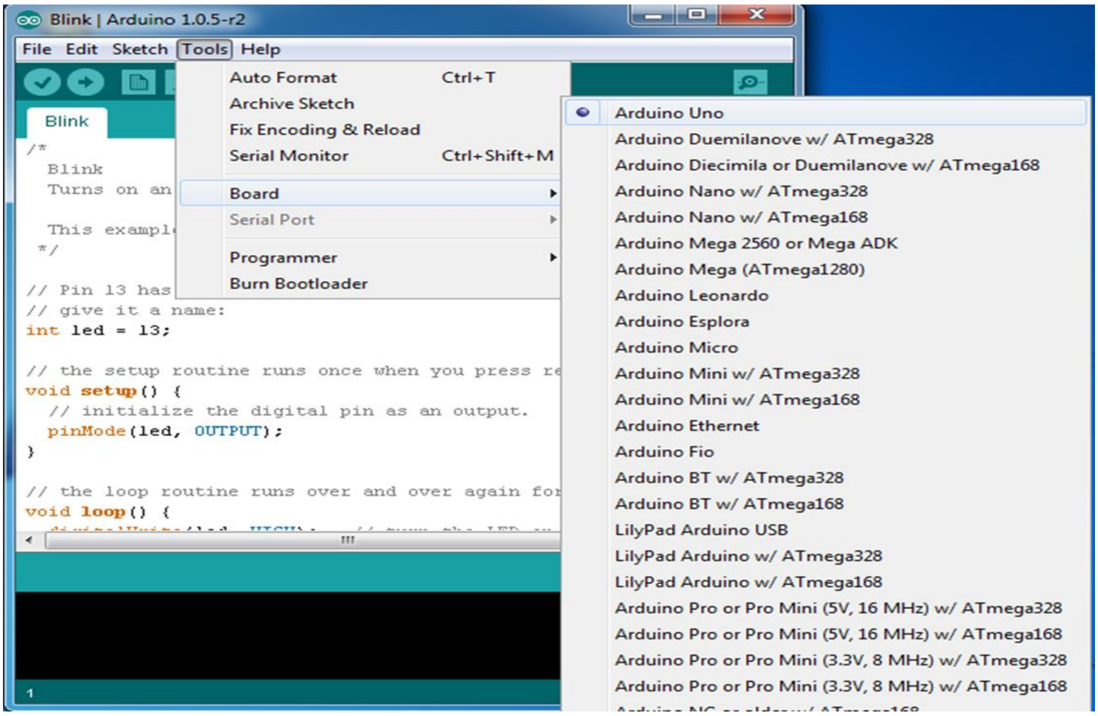
คุณยังสามารถใช้ IDE อื่นได้ถ้าหากคุณต้องการ เช่น Visual Studio C++ ที่พัฒนาโดย Microsoft

ในที่นี้ขอใช้ เป็นโปรแกรม Arduino IDE แทนเพื่อให้เหมาะสมกับการเขียนโปรแกรมแล้วนำไปควบคุมอุปกรณ์ภายนอกได้

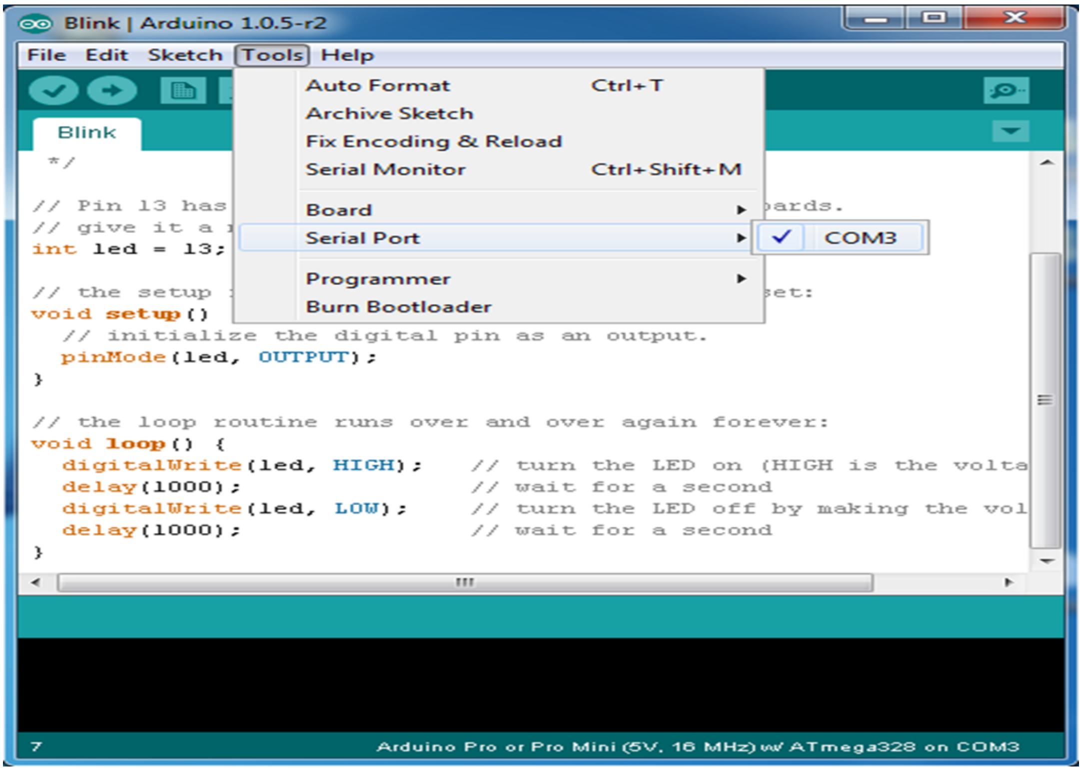


1. เขียนโปรแกรมบนคอมพิวเตอร์ ผ่านทางโปรแกรม ArduinoIDE ซึ่งสามารถดาวน์โหลดได้จาก Arduino.cc/en/main/software

2. หลังจากที่เขียนโค้ดโปรแกรมเรียบร้อยแล้ว ให้ผู้ใช้งานเลือกรุ่นบอร์ด Arduino ที่ใช้และหมายเลข Com port

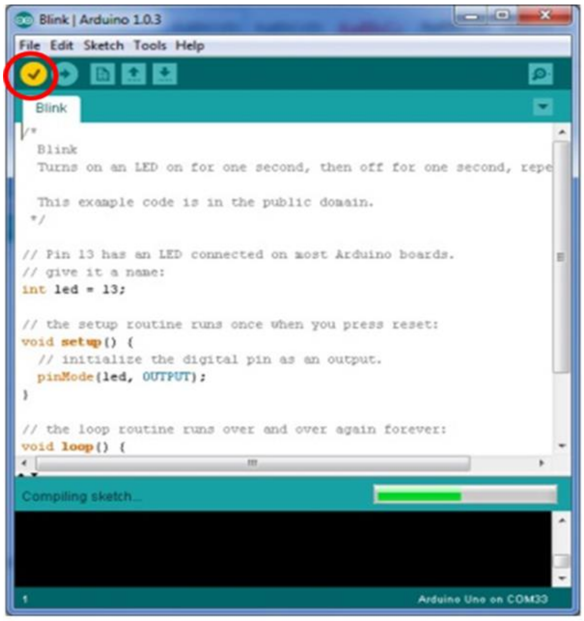


เลือกรุ่นบอร์ด Arduino ที่ต้องการ upload

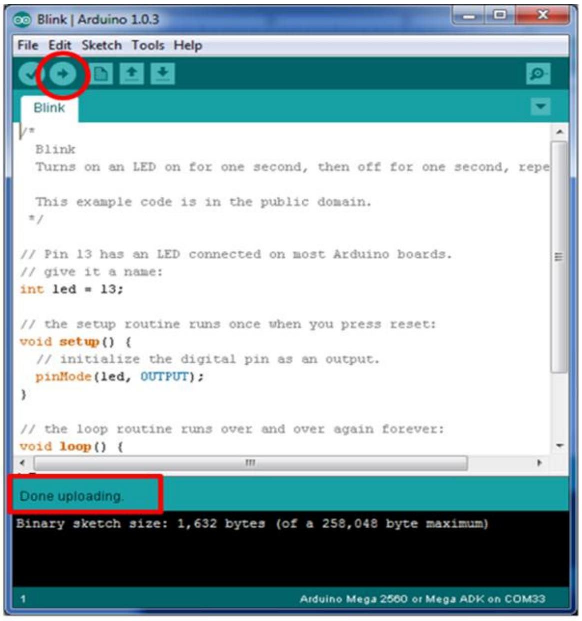


Comport ของบอร์ด

3. กดปุ่ม Verify เพื่อตรวจสอบความถูกต้องและ Compile โค้ดโปรแกรม จากนั้นกดปุ่ม Upload โค้ดโปรแกรมไปยังบอร์ด Arduinoผ่านทางสาย USB เมื่ออัปโหลดเรียบร้อยแล้ว จะแสดงข้อความแถบข้างล่าง “Done uploading” และบอร์ดจะเริ่มทำงานตามที่เขียนโปรแกรมไว้ได้ทันที



กดปุ่ม Verify เพื่อตรวจสอบความถูกต้อง
และ Compile โค้ดโปรแกรม



Upload โค้ดโปรแกรม

การติดต่อกับหน้าจอคอมและคีย์บอร์ด

```
void setup()
{
  Serial.begin(9600);     //เริ่มต้นการสื่อสารที่ความเร็ว 9600 bit/sec
}
void loop()
{
  Serial.print("Hello ! A-TECH");//ให้แสดงคำว่า Hello ! A-TECH ไปเรื่อยๆ ครั้งละ1วินาที โดยไม่มีการเว้นบรรทัด
  delay(1000);
```

การตั้ง Buad rate

Buad rate หรือ บอดเรท คือความเร็วในการสื่อสาร จำเป็นจะต้องตั้งให้ตรงกันระหว่างบอร์ดและคอมพิวเตอร์ โดยทั่วไปจะใช้ค่าตั้งแต่ 9600 - 115200 ถ้าความเร็วต่ำ จะใช้เวลามาก ในการรับส่งข้อมูล แต่ถ้าความเร็วสูงเกินไป อาจทำให้การอ่านผิดพลาดได้ ดังนั้นในตัวอย่างนี้ เราจะใช้ค่า 9600

จุดเน้นย้ำ

เมื่อต้องการทดสอบโปรแกรม จะต้องเชื่อมต่อบอร์ดเข้ากับคอมพิวเตอร์

และเมื่อมีการเปลี่ยนแปลงโค้ดโปรแกรมที่ใช้ทดสอบ จะต้องอัพโหลดโค้ดใหม่ทุกครั้ง

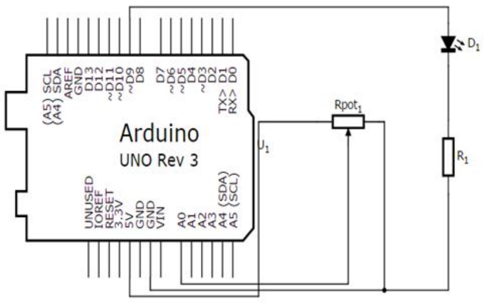
การรับอินพุทอนาลอก

รับได้จาก เซ็นเซอร์อนาลอก LDR   VR แรงดันไฟฟ้าที่ไม่เกิน 5 v. และอีกมากมาย ในกรณีที่ Load มีค่าไม่เกิน 1000 Ohm

```
ตัวอย่าง
int a=0;
void setup()
{
  pinMode(14,INPUT);  //ขาที่ 14 คือขา A0นั่นเอง
  Serial.begin(9600);
}
void loop()
{
  a=analogRead(14);    //ให้ a เก็บค่าที่อ่านได้จากขา 14
  Serial.println(a);        //ให้แสดงผลค่าที่อ่านได้บนจอคอม
  delay(1000);
}
```

ลำดับขั้นการทดลอง

1. เขียนโปรแกรม ตาม **ตัวอย่างที่ 1** ในโปรแกรม Arduino IDE และทำการ compiler ให้เรียบร้อย แล้วนำ ไฟล์ที่มีนามสกุล .hex ไฟล์ มา run ในโปรแกรม proteus 8.1 หลังจากที่ได้ต่อวงจรไว้สมบูรณ์แล้วทำการ Simulate ดูผลของการทดลอง
2. ให้ทดลองต่อวงจรจริงที่ เบรดบอร์ด แล้วทำการ copy โปรแกรมลงบนบอร์ด Arduino UNO R3 เพื่อทดลอง สังเกตผลการทดลองบันทึกผล



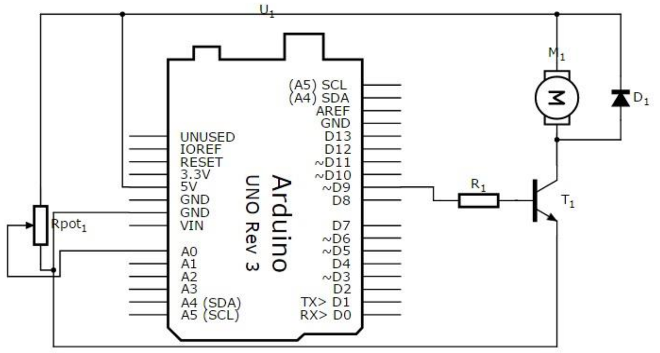
ตัวอย่างที่ 1

```
const int analogInPin = A0;  // Analog input pin that the potentiometer is attached to
const int analogOutPin = 9; // Analog output pin that the LED is attached to
int sensorValue = 0;        // value read from the pot
int outputValue = 0;        // value output to the PWM (analog out)
void setup() {
  // initialize serial communications at 9600 bps:
  Serial.begin(9600);
}
void loop() {
  // read the analog in value:
  sensorValue = analogRead(analogInPin);
  // map it to the range of the analog out:
  outputValue = map(sensorValue, 0, 1023, 0, 255);
  // change the analog out value:
  analogWrite(analogOutPin, outputValue);
  // print the results to the serial monitor:
  Serial.print("sensor = " );
  Serial.print(sensorValue);
  Serial.print("\t output = ");
  Serial.println(outputValue);
  // wait 2 milliseconds before the next loop
  // for the analog-to-digital converter to settle
  // after the last reading:
  delay(2);
}
```

3. เขียนโปรแกรมตาม **ตัวอย่างที่ 2** ในโปรแกรม Arduino IDE และทำการ compiler ให้เรียบร้อย แล้วนำ ไฟล์ที่มีนามสกุล .hex ไฟล์ มา run ในโปรแกรม proteus 8.1 หลังจากที่ได้ต่อวงจรไว้สมบูรณ์แล้วทำการ Simulate ดูผลของการทดลอง
4. ให้ทดลองต่อวงจรจริงที่ เบรดบอร์ด แล้วทำการ copy โปรแกรมลงบนบอร์ด Arduino UNO R3 เพื่อทดลอง สังเกตผลการทดลองบันทึกผล

ตัวอย่างที่ 2

```
const int analogInPin = A0;  // Analog input pin that the potentiometer is attached
to
const int analogOutPin = 9; // Analog output pin that the LED is attached to
int sensorValue = 0;        // value read from the pot
int outputValue = 0;        // value output to the PWM (analog out)
void setup() {
  // initialize serial communications at 9600 bps:
  Serial.begin(9600);
}
void loop() {
  // read the analog in value:
  sensorValue = analogRead(analogInPin);
  // map it to the range of the analog out:
  outputValue = map(sensorValue, 0, 1023, 0, 255);
  // change the analog out value:
  analogWrite(analogOutPin, outputValue);
  // print the results to the serial monitor:
  Serial.print("sensor = " );
  Serial.print(sensorValue);
  Serial.print("\t output = ");
  Serial.println(outputValue);
  // wait 2 milliseconds before the next loop
  // for the analog-to-digital converter to settle
  // after the last reading:
  delay(2);
}
```



ผลการทดลอง

....

....

....

สรุปผลการทดลอง

....

....

....

การประเมินผล.................................

เอกสารอ้างอิง Credit :.........................

ชื่อ-สกุล..ชั้น..............................เลขที่...............................