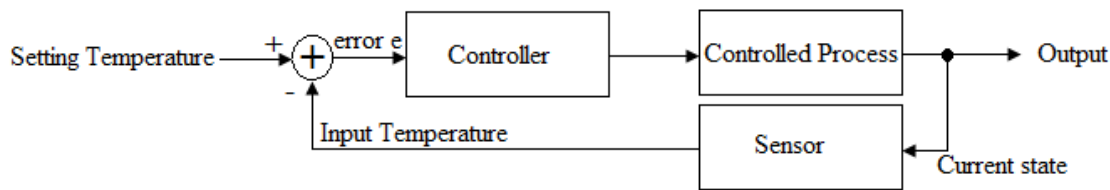


# การควบคุมอุณหภูมิ

## 1. คำนำ

ระบบควบคุมหมายถึงอุปกรณ์หรือเครื่องมือที่ใช้ควบคุมการทำงานให้เป็นไปตามที่ตั้งไว้ เช่นการควบคุมอัตราการหมุนของมอเตอร์ให้อยู่ที่รอบต่อนาที การควบคุมนั้นมียุหลายวิธี สำหรับในตัวอย่างนี้จะแสดงการควบคุมอุณหภูมิ ด้วย Arduino โดยมีไดอะแกรมของระบบตามรูปที่ 1



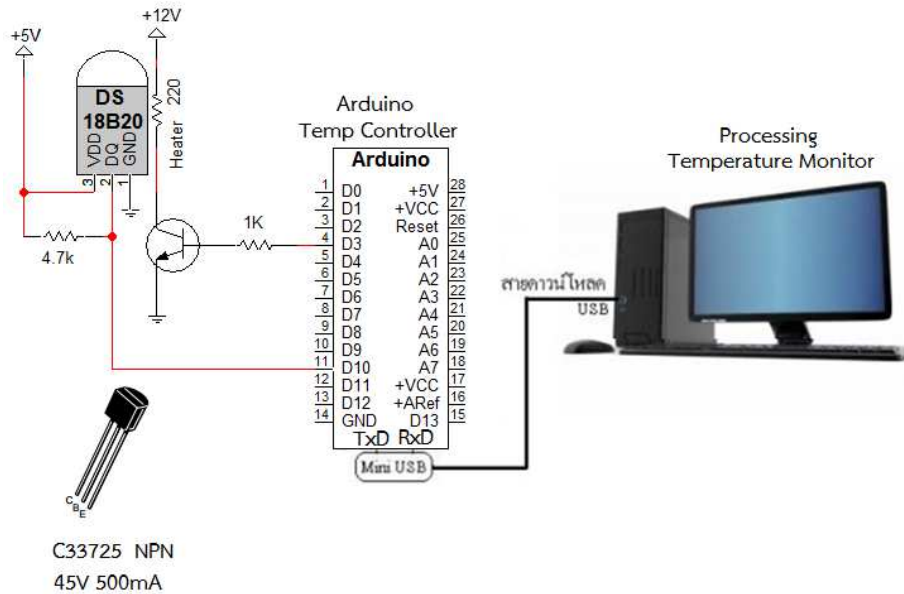
รูปที่ 1 ระบบควบคุมโดยใช้ Arduino

Setting Temperature	เป็นค่าอุณหภูมิที่ตั้ง
Controller	คือ Arduino และฮาร์ดแวร์ที่เกี่ยวข้อง
Controlled Process	คือวิธีการควบคุม ที่เขียนเป็นโปรแกรมไว้ Arduino ทำงาน มีอยู่ 3 แบบ 3 วิธี คือ การควบคุมแบบเปิด-ปิด การควบคุมแบบพีไอดี และการควบคุมแบบพีซีซี
Output	คือค่าสัญญาณที่ส่งไปควบคุมอุปกรณ์ทำความร้อน
Current state	คือค่าอุณหภูมิเวลาปัจจุบัน
Sensor	เป็นอุปกรณ์แปลงค่าอุณหภูมิ ให้เป็นสัญญาณดิจิทัล
Input Temperature	คือค่าอุณหภูมิที่แปลงเป็นสัญญาณดิจิทัลแล้ว

หมายเหตุ เพื่อความปลอดภัย

- เปลี่ยนจาก Heater ที่ใช้กับไฟฟ้ากระแสสลับ มาเป็น ความต้านทานขนาด 100 โอห์ม
- เปลี่ยนจากการใช้ไฟฟ้า 220 v AC มาเป็นไฟ DC 12 V
- Sensor ใช้ DS18B20
- Setting Temperature ตั้งค่าด้วยโปรแกรม Processing บน PC ผ่านทางพอร์ตอนุกรม
- Input Temperature ดูจากโปรแกรม Processing บน PC เช่นเดียวกัน

วงจรทดลองใช้ตามรูปที่ 2



รูปที่ 2 วงจรทดลองการควบคุมอุณหภูมิด้วย Arduino

## 2. การควบคุมแบบเปิด-ปิด

เป็นระบบที่ทำงานในลักษณะ 2 สถานะ คือ เปิด และ ปิด ดังนั้นสัญญาณ เอาท์พุทของการควบคุมจะมีเพียง 2 สถานะเท่านั้น คือเปิด (ให้เป็นเลข 1) และ ปิด (ให้เป็นเลข 0) การทำงานนี้ถูกกำหนดด้วยค่าความผิดพลาดของกระบวนการ ตามรูปสมการคือ

$$e = \text{TempSet} - \text{TempIn} \quad (1)$$

$$\text{Output} = \begin{cases} 0 & \text{เมื่อ } e \leq 0 \\ 1 & \text{เมื่อ } e > 0 \end{cases} \quad (2)$$

- เมื่อ
- e คือ ค่าความผิดพลาด
  - TempSet คือค่าอุณหภูมิที่ตั้ง
  - TempIn คือค่าอุณหภูมิที่อ่านได้
  - 1 คือ สัญญาณที่ส่งให้ระบบทำความร้อนทำงาน
  - 0 คือ สัญญาณที่ส่งให้ระบบทำความร้อนหยุดทำงาน

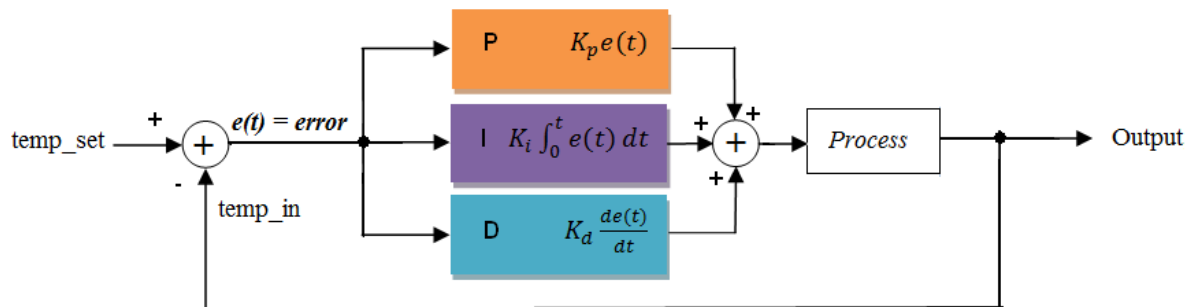
จากสมการที่ (1) และ (2) เขียนเป็นคำสั่ง ได้ดังนี้

```
if(temp_in > temp_set){
  heaterOFF;
}
else{
  heaterON;
}
```

## 2.2 การควบคุมแบบพีไอดี (PID)

การควบคุมแบบพีไอดี เป็นการควบคุมแบบระบบวงปิดหรือระบบควบคุมแบบป้อนกลับ (closed-loop control systems, feedback control systems) ค่าที่นำไปใช้ในการคำนวณเป็นค่าความผิดพลาดที่หามาจากความแตกต่างระหว่าง ค่าที่ตั้งกับค่าเอาต์พุตปัจจุบัน แล้วใช้สมการที่ (3) คำนวณเพื่อลดความผิดพลาดให้เหลือน้อยที่สุด การควบคุมแบบพีไอดี จะประกอบไปด้วยส่วนการควบคุมที่สำคัญ 3 ส่วน คือเทอม Proportional term  $K_p e(t)$  Integral term  $K_i \int_0^t e(t) dt$  และ Derivative term  $K_d \frac{de(t)}{dt}$  ตามไดอะแกรมในรูปที่ 3

$$Y(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt} \quad (3)$$



รูปที่ 3 บล็อกไดอะแกรมของตัวควบคุมแบบพีไอดี

ค่า  $K_p$   $K_i$  และ  $K_d$  เป็นค่าคงที่ ขึ้นอยู่กับระบบที่ควบคุม การปรับเปลี่ยนค่านี้มีหลายวิธี ตัวอย่างการปรับด้วยมือดูได้จาก [https://en.wikipedia.org/wiki/PID\\_controller#Alternative\\_nomenclature\\_and\\_PID\\_forms](https://en.wikipedia.org/wiki/PID_controller#Alternative_nomenclature_and_PID_forms) จากสมการนำมาเขียนเป็นโปรแกรมได้ดังนี้

```
void Compute()
{
    /*How long since we last calculated*/
    unsigned long now = millis();
    double timeChange = (double)(now - lastTime);

    /*Compute all the working error variables*/
    double error = temp_set - temp_in;
    errSum += (error * timeChange);
    double dErr = (error - lastErr) / timeChange;

    /*Compute PID Output*/
    Output = kp * error + ki * errSum + kd * dErr;

    /*Remember some variables for next time*/
    lastErr = error;
    lastTime = now;
}
```

จาก <http://brettbeauregard.com/blog/2011/04/improving-the-beginners-pid-introduction/>

เพื่อความสะดวกในการคำนวณ สามารถใช้ฟังก์ชันในไลบรารี PID ของ Arduino ก็ได้  
สามารถดูได้จาก <http://playground.arduino.cc/Code/PIDLibrary>