โครงสร้างของการเขียนโปรแกรม

การเขียนโปรแกรมเพื่อสั่งงาน Arduino นั้นจะใช้ภาษาซี โดยโครงสร้างของการเขียนโปรแกรมจะคล้ายๆกับการเขียนโปรแกรมบนไมโครคอนโทรลเลอร์ทั่วๆไป แต่จะมีความง่ายกว่าเพราะ Arduino ได้มีการรวมคำสั่งและ Library ต่างๆไว้ให้ผู้ใช้สามารถเรียกใช้งานได้เลย ถ้าจะกล่าวถึงการเขียนโปรแกรมด้วยภาษาซีนั้น ฟังชั้นหลักที่จะขาดไม่ได้เลยนั้นคือฟังก์ชัน Main แต่การเขียนโปรแกรมกับ Arduino จะเป็นฟังก์ชัน setup และฟังก์ชัน loop ตามรูป

```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

รูปโครงสร้างการเขียนโปรแกรมขั้นต่ำของ Arduino

การทำงานของฟังก์ชันทั้งสองมีความหมายดังนี้

การทำงานของ void setup() เป็นฟังก์ชันที่ใช้สำหรับกำหนดค่าเริ่มต้นต่างๆให้กับบอร์ด เมื่อเริ่มต้นการทำงาน Arduino จะทำตามคำสั่งต่างๆที่อยู่ใน void setup() ก่อน 1 รอบ หลังจากนั้นจะเข้าสู่ void loop() ต่อไป การกำหนดรูปแบบโหมดการทำงานของสัญญาณในแต่ละขาที่จะใช้งานของไมโครคอนโทรลเลอร์ว่ามีหน้าที่อะไรนั้นแบ่งออกเป็นสองรูปแบบคือ สัญญาณขาเข้าหรือเรียกว่าสัญญาณอินพุท (input) และสัญญาณขาออกหรือเรียกว่าสัญญาณเอาท์พุท (output) ซึ่งเราต้องกำหนดโหมดการทำงานของมันเสียก่อนด้วยคำสั่ง pinMode(pin, mode) โดย pin เป็นหมายเลขขาของบอร์ดไมโครคอนโทรลเลอร์ และในส่วนของ mode เป็นการเลือกโหมดการทำงานของขานั้นโดยมีสามรูปแบบคือ

INPUT	กำหนดให้ขานั้นทำหน้าที่เป็น input
OUTPUT	กำหนดให้ขานั้นทำหน้าที่เป็น output
INPUT_PULLUP	กำหนดให้ขานั้นทำหน้าที่เป็น input ที่มีการต่อ internal resistor แบบ pull-up

การทำงานของ void loop() เป็นฟังก์ชันสำหรับสั่งให้ Arduino ทำตามคำสั่งต่างๆที่เราเขียนไว้วนรอบซ้ำกันไป โดยจะเริ่มต้นทำงานเมื่อผ่านจาก void setup() มาแล้ว

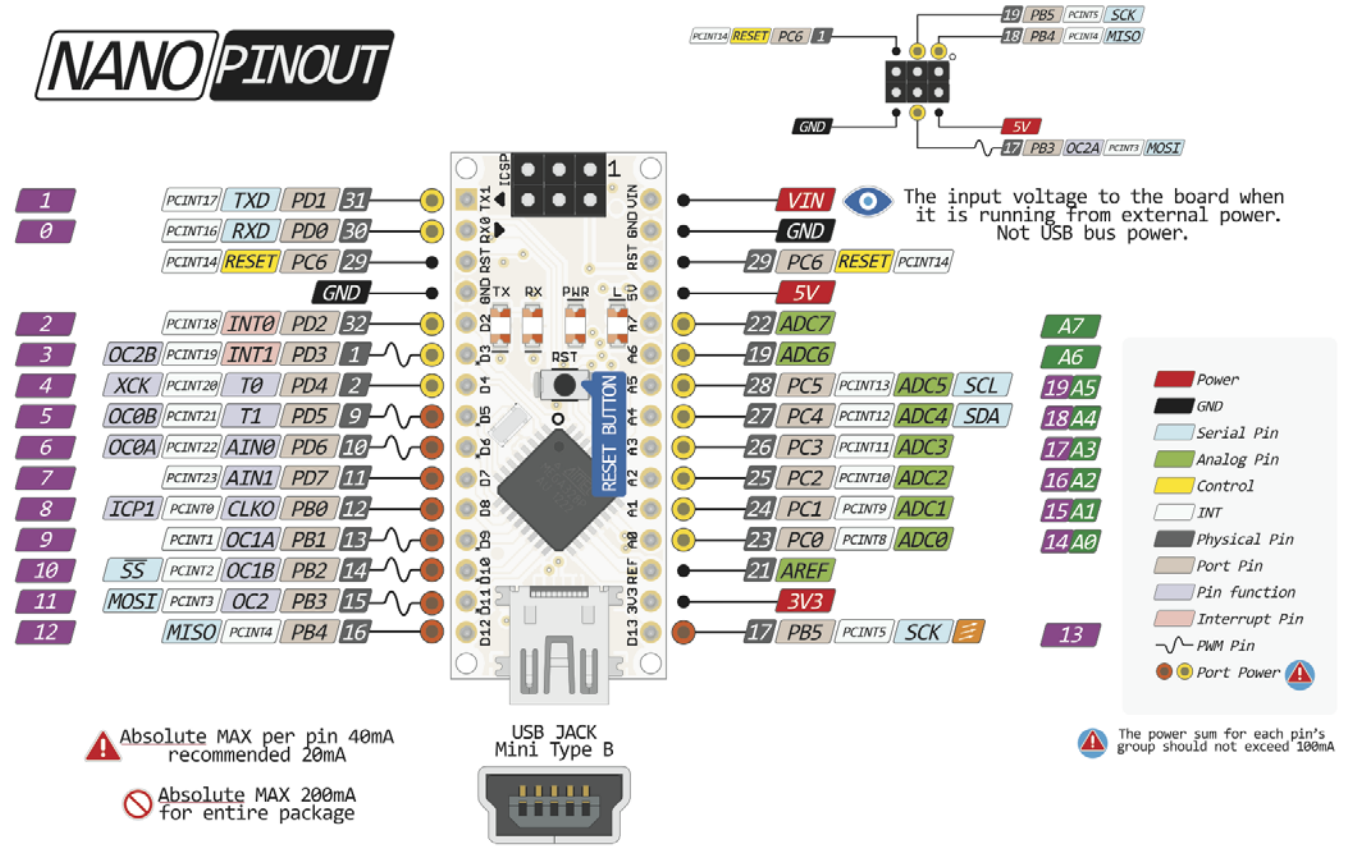
การทดลองจะเขียนโปรแกรมควบคุมโดยใช้ Arduino ต่อกับ Protoboard แล้วทำการเชื่อมต่ออุปกรณ์ที่ใช้ในการทดลองลงบนโปรโตบอร์ดดังรูป และให้ขาดิจิตอลที่ D2 ของ Arduino ส่งค่าสัญญาณเอาท์พุทแบบดิจิตอลออกมา ด้วยคำสั่ง digitalWrite(pin,value) โดยที่ค่า value ของสัญญาณดิจิตอลที่ได้มีอยู่ 2 รูปแบบคือ สัญญาณ HIGH และ LOW

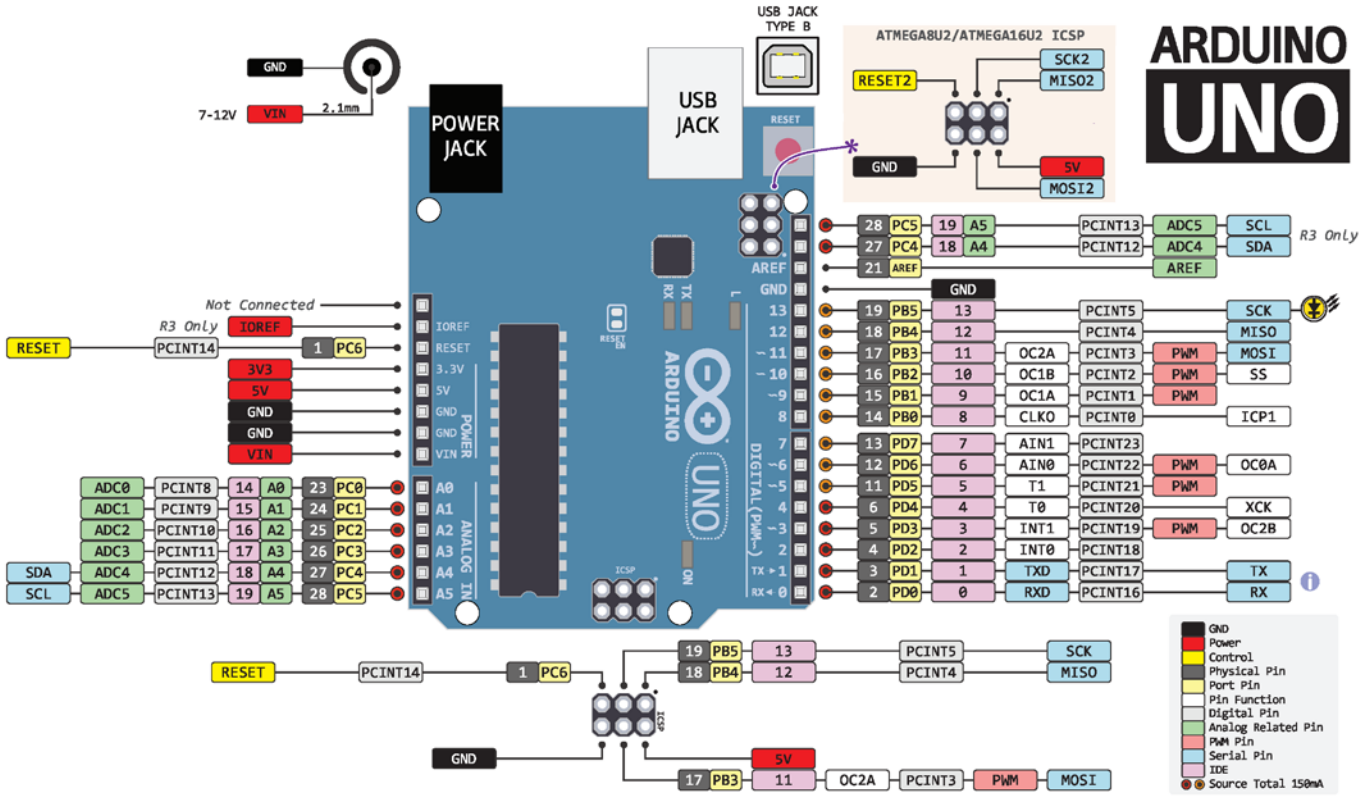
เมื่ออยู่ในสถานะ “HIGH” ขาของ Arduino Nano จะส่งแรงดันไฟฟ้าขนาด 5 โวลต์ออกมา

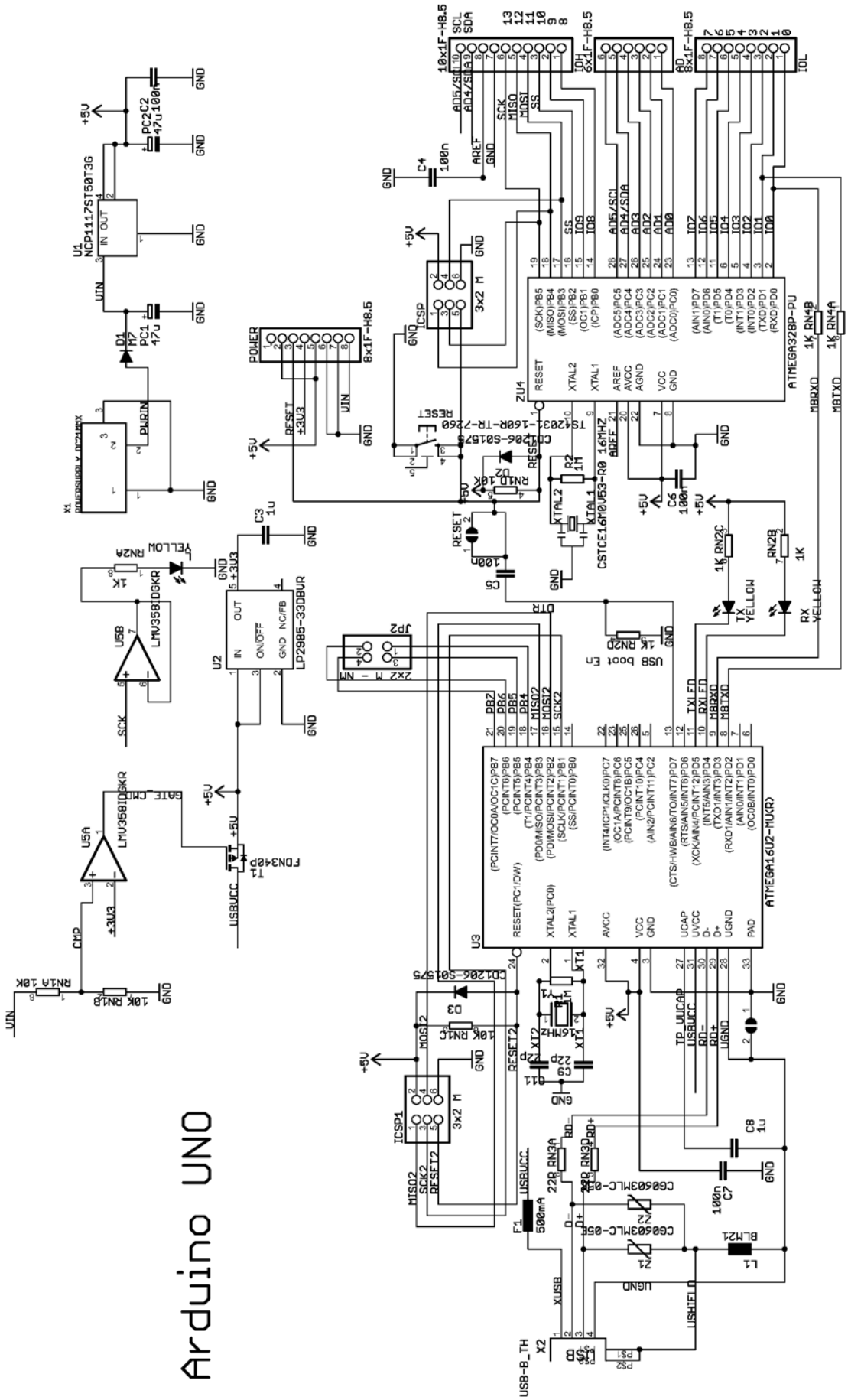
เมื่ออยู่ในสถานะ “LOW” ขาของ Arduino Nano จะเชื่อมต่อกับ Ground (GND)

แรงดันไฟฟ้าที่ไมโครคอนโทรลเลอร์ส่งออกมา เมื่อเราสั่ง HIGH นั้นขึ้นอยู่กับรุ่นของบอร์ดไมโครคอนโทรลเลอร์ที่เราใช้งาน หากเป็นบอร์ดไมโครคอนโทรลเลอร์ที่ใช้แหล่งจ่ายไฟ 3.3 โวลต์ การสั่ง HIGH จะเป็นการสร้างสัญญาณ 3.3 โวลต์ออกมาที่ขานั้น และหากเป็นไมโครคอนโทรลเลอร์ที่ใช้ไฟ 5 โวลต์ สัญญาณ HIGH ที่ออกมาก็จะเป็น 5 โวลต์

วงจรและตำแหน่งขาต่างๆของบอร์ด Arduino ที่ใช้ในการทดลองเป็นดังนี้

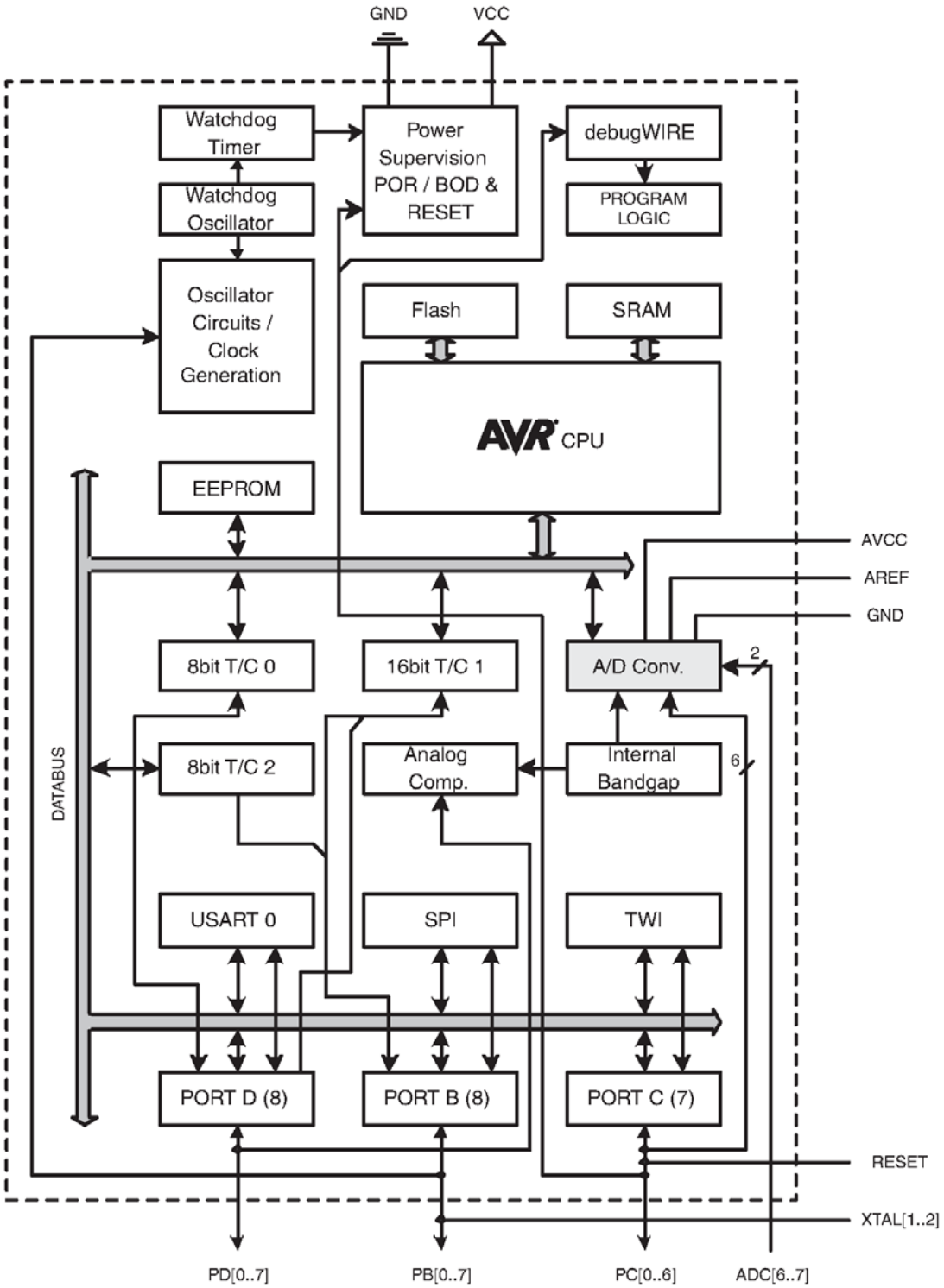

NANO PINOUT
The input voltage to the board when it is running from external power. Not USB bus power.
RESET BUTTON
USB JACK Mini Type B
Absolute MAX per pin 40mA recommended 20mA
Absolute MAX 200mA for entire package
The power sum for each pin's group should not exceed 100mA
Power
GND
Serial Pin
Analog Pin
Control
INT
Physical Pin
Port Pin
Pin function
Interrupt Pin
PWM Pin
Port Power


ARDUINO UNO
USB JACK TYPE B
POWER JACK
USB JACK
ATMEGA8U2/ATMEGA16U2 ICSP
7-12V
2.1mm
R3 Only
Not Connected
GND
Power
Control
Physical Pin
Port Pin
Pin Function
Digital Pin
Analog Related Pin
PWM Pin
Serial Pin
IDE
Source Total 150mA


Arduino UNO

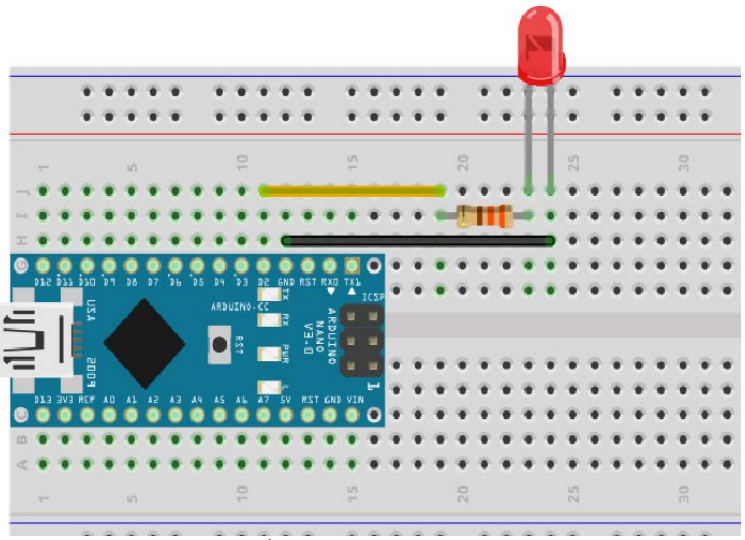
ภายในบอร์ด Arduino ที่ใช้ในการทดลองจะใช้ตัวประมวลผลเป็น Microcontroller AVR ขนาด 8 bit เบอร์ ATmega328P ใช้สถาปัตยกรรม (Architecture) แบบ Reduced Instruction Set Computer (RISC) ที่มีชุดคำสั่งที่สั้นและมีจำนวนคำสั่งไม่มากนัก สามารถกระทำการตามคำสั่งได้อย่างรวดเร็ว ตรงข้ามกับ Complex Instruction Set Computer (CISC) รูปด้านล่างจะเป็น Block Diagram ภายในของไมโครคอนโทรลเลอร์นี้ รายละเอียดต่างๆเพิ่มเติมให้เปิดดูได้จากเอกสาร ATMEGA328 DataSheet

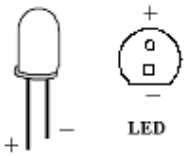
Block Diagram ของ ATmega328P



การใช้คำสั่ง digitalWrite กับขาที่เลือกโหมดเป็น input จะเป็นการเปิด-ปิด การต่อ pull-up ภายในวงจรของไมโครคอนโทรลเลอร์ โดย HIGH เป็นการเปิดโหมดการต่อ Pull-up และ LOW เป็นการปิดโหมด pull-up

ในกรณีการใช้คำสั่ง digitalWrite เพื่อสั่งให้ LED สว่าง โดยที่ไม่ได้กำหนดโหมดการทำงานของขาด้วยคำสั่ง pinMode จะส่งผลให้ LED ที่ต่ออยู่กับขานั้นไม่สว่างเท่าที่ควร เพราะการไม่ใช้คำสั่ง pinMode จะเปิดการทำงานโหมด pull-up ซึ่งจะทำให้กระแสไฟบางส่วนไหลผ่านตัวต้านทานภายในโดยไม่ผ่าน LED





รูปการเชื่อมต่ออุปกรณ์การทดลองลงบนโปรโตบอร์ด

1. ให้เชื่อมต่อสาย USB ของบอร์ดกับคอมพิวเตอร์ เปิดโปรแกรม Arduino ที่ได้ติดตั้งในคอมพิวเตอร์ จากนั้นทำการเขียนโปรแกรมที่ทำหน้าที่สั่งงานให้ LED ที่อยู่บนบอร์ดไมโครคอนโทรลเลอร์กระพริบทุก 1 วินาที จากนั้นทำการคอมไพล์แล้วทำการ Upload โปรแกรมที่ได้ลงบนบอร์ด Arduino

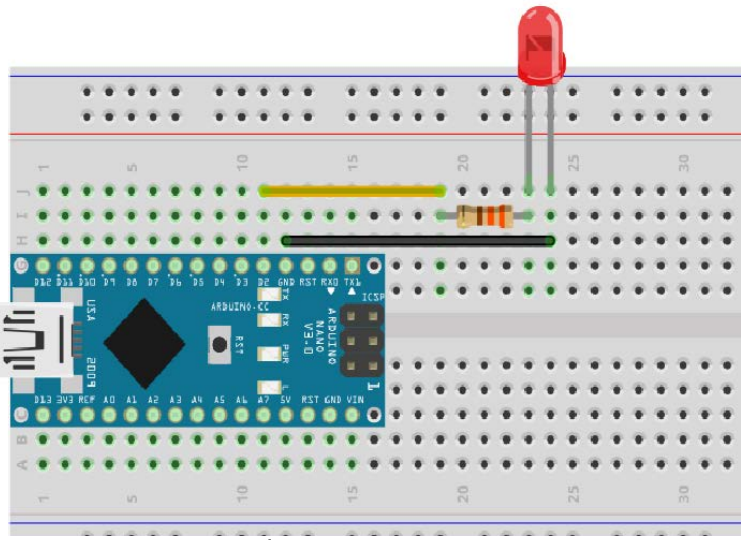
```
int led = 13;                    // LED connected to digital pin 13

void setup()
{
  pinMode(led, OUTPUT);          // initialize the digital pin as an output
}

void loop()
{
  digitalWrite(led, HIGH);       // turn the LED on (HIGH is the voltage level)
  delay(1000);                   // wait for a second (1000 milliseconds)
  digitalWrite(led, LOW);        // turn the LED off by making the voltage LOW
  delay(1000);                   // wait for a second (1000 milliseconds)
}
```

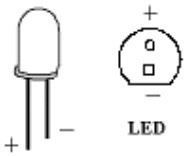
คำสั่งที่ใช้มีความหมายดังนี้

int led = 13;	ใช้ในการกำหนดขาที่ต่อ LED ภายในบอร์ด Arduino ว่าต่ออยู่ที่ขา 13
pinMode(led, OUTPUT);	กำหนดให้ขาที่ต่อ LED เป็นขาเอาท์พุท
digitalWrite(led, HIGH);	ให้ส่งค่าลอจิก 1 ออกไปขาที่ต่อกับ LED
digitalWrite(led, LOW);	ให้ส่งค่าลอจิก 0 ออกไปขาที่ต่อกับ LED
delay(1000);	ให้ทำการหน่วงเวลา 1000 ms (Milliseconds)

การใช้คำสั่ง digitalWrite กับขาที่เลือกโหมดเป็น input จะเป็นการเปิด-ปิด การต่อ pull-up ภายในวงจรของไมโครคอนโทรลเลอร์ โดย HIGH เป็นการเปิดโหมดการต่อ Pull-up และ LOW เป็นการปิดโหมด pull-up

ในกรณีการใช้คำสั่ง digitalWrite เพื่อสั่งให้ LED สว่าง โดยที่ไม่ได้กำหนดโหมดการทำงานของขาด้วยคำสั่ง pinMode จะส่งผลให้ LED ที่ต่ออยู่กับขานั้นไม่สว่างเท่าที่ควร เพราะการไม่ใช้คำสั่ง pinMode จะเปิดการทำงานโหมด pull-up ซึ่งจะทำให้กระแสไฟบางส่วนไหลผ่านตัวต้านทานภายในโดยไม่ผ่าน LED





รูปการเชื่อมต่ออุปกรณ์การทดลองลงบนโปรโตบอร์ด

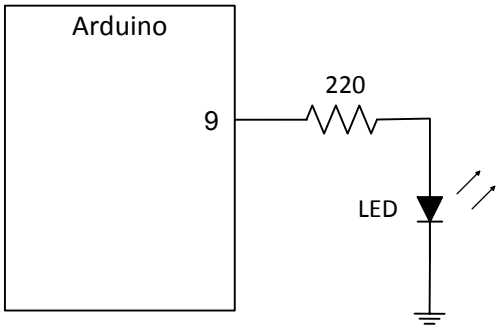
1. ให้เชื่อมต่อสาย USB ของบอร์ดกับคอมพิวเตอร์ เปิดโปรแกรม Arduino ที่ได้ติดตั้งในคอมพิวเตอร์ จากนั้นทำการเขียนโปรแกรมที่ทำหน้าที่สั่งงานให้ LED ที่อยู่บนบอร์ดไมโครคอนโทรลเลอร์กระพริบทุก 1 วินาที จากนั้นทำการคอมไพล์แล้วทำการ Upload โปรแกรมที่ได้ลงบนบอร์ด Arduino

```
int led = 13;                    // LED connected to digital pin 13

void setup()
{
  pinMode(led, OUTPUT);          // initialize the digital pin as an output
}

void loop()
{
  digitalWrite(led, HIGH);       // turn the LED on (HIGH is the voltage level)
  delay(1000);                   // wait for a second (1000 milliseconds)
  digitalWrite(led, LOW);        // turn the LED off by making the voltage LOW
  delay(1000);                   // wait for a second (1000 milliseconds)
}
```

คำสั่งที่ใช้มีความหมายดังนี้

int led = 13;	ใช้ในการกำหนดขาที่ต่อ LED ภายในบอร์ด Arduino ว่าต่ออยู่ที่ขา 13
pinMode(led, OUTPUT);	กำหนดให้ขาที่ต่อ LED เป็นขาเอาท์พุท
digitalWrite(led, HIGH);	ให้ส่งค่าลอจิก 1 ออกไปขาที่ต่อกับ LED
digitalWrite(led, LOW);	ให้ส่งค่าลอจิก 0 ออกไปขาที่ต่อกับ LED
delay(1000);	ให้ทำการหน่วงเวลา 1000 ms (Milliseconds)

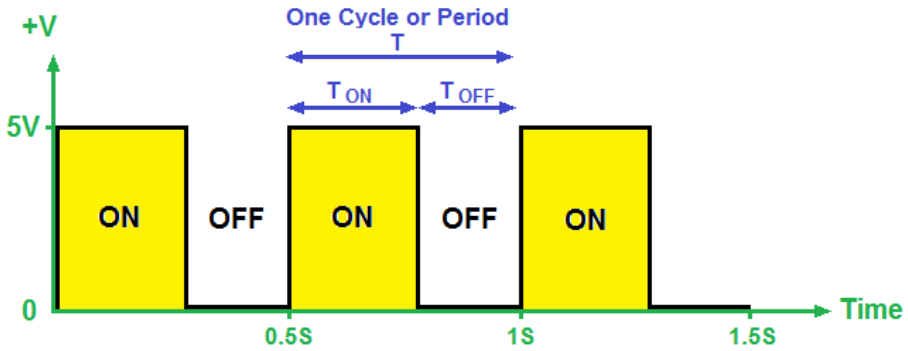
2. ให้ทำการย้ายขา LED ของโปรแกรมที่ ต่ออยู่ขาที่ 13 ไปเป็นขาที่ 9 และให้ต่อ LED อนุกรมกับตัวความต้านทาน 220 Ω เข้ากับขาที่ 9 แล้วลงกราวด์



3. จากข้อ 1 ให้แก้ไขโปรแกรมให้ LED กระพริบเป็นความถี่ 10 Hz

โดยที่ความถี่ (Frequency) เป็นจำนวนรอบที่แสดงว่าคลื่นเคลื่อนที่ไปได้กี่รอบในหนึ่งวินาที (Second) มีหน่วยเป็น รอบต่อวินาทีหรือเฮิรตซ์ (Hz) ใช้แทนสัญลักษณ์ด้วย f

คาบเวลา (Period) คือ เวลาที่ใช้ในการเคลื่อนที่ครบ 1 รอบ (One Cycle) มีหน่วยเป็นวินาที (Second) ใช้แทนสัญลักษณ์ด้วย T



จากรูปเมื่อเวลาผ่านไปหนึ่งวินาที คลื่นเคลื่อนที่ได้สองลูก แสดงว่าคลื่นนี้มีความถี่ 2 Hz

หรือถ้าพิจารณาจากคาบเวลาจะเห็นว่าใน 1 รอบจะใช้เวลา T = TON + TOFF = 0.5 Sec

ดังนั้นจะได้ความสัมพันธ์ระหว่างความถี่ (f) และคาบ (T) ตามสมการ

$$f = \frac{1}{T}$$

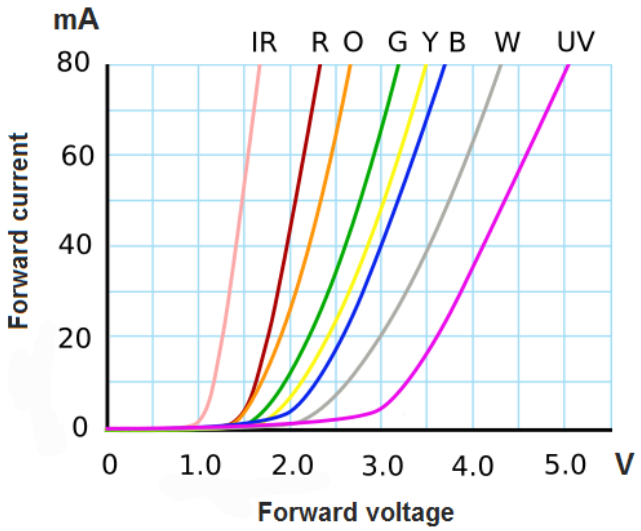
$$f = \frac{1}{0.5}$$

$$f = 2 \text{ Hz}$$

4. ให้วัดแรงดันไฟฟ้าตกคร่อมตัว LED ในช่วงที่ LED กำลังทำงาน (LED ON) โดยที่ LED จะต้องต่ออนุกรมกับตัวความต้านทาน 220 Ω

LED	Forward Voltage	$5-V_F$
infrared (IR)	1.26	3.74
Red	1.85	3.15
Yellow	2.02	2.98
Green	2.03	2.97
Blue	2.88	2.12
White	2.93	2.07
Ultraviolet (UV)	3.13	1.87

ค่าแรงดันไฟฟ้าที่ตกคร่อม LED ขณะป้อนแรงดันไฟฟ้าแบบตรงตามขั้ว (Forward) จะแปรผันตามกระแสที่ไหลผ่าน และจะขึ้นอยู่กับค่าความยาวคลื่นของแสงที่ส่องสว่างออกมาจาก LED ด้วย ดังตัวอย่างจะเป็นกราฟ แสดงค่า Characteristic ของ LED แต่ละสี



5. ให้คำนวณหากระแสที่ไหลผ่าน LED ที่ใช้ทดลองมา 7 ตัวในช่วงขณะที่ LED กำลังทำงาน (LED ON) โดยใช้กฎของโอห์ม

$V_{IR} = \frac{3.74}{220} = 17$ mA.

$V_{Red} = \frac{3.15}{220} = 14$ mA.

$V_{Yellow} = \frac{2.98}{220} = 14$ mA.

$V_{Green} = \frac{2.97}{220} = 14$ mA.

$I_{Blue} = \frac{2.12}{220} = 10$ mA.

$I_{White} = \frac{2.07}{220} = 9$ mA.

$I_{UV} = \frac{1.87}{220} = 9$ mA.

กฎของโอห์ม (Ohm's Law) กล่าวไว้ว่ากระแสไฟฟ้าที่ไหลในตัวนำไฟฟ้าจะแปรผันตามแรงดันที่ตกคร่อมตัวนำนั้น และจะแปรผกผันกับค่าความต้านทานของตัวนำนั้น ดังสมการ

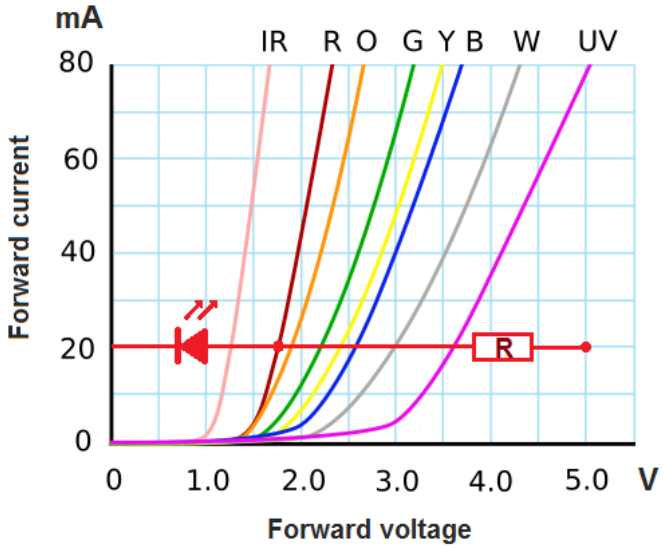
$$I = V / R$$

เมื่อ I = กระแสไฟฟ้ามีหน่วยเป็นแอมป์แปร์ (A)

V = แรงดันไฟฟ้ามีหน่วยเป็นโวลต์ (V)

R = ความต้านทานมีหน่วยเป็นโอห์ม (Ω)

โดยที่แรงดันตกคร่อมตัวความต้านทานได้จากแรงดันของแหล่งจ่ายไฟลบด้วยแรงดันตกคร่อม LED มีความสัมพันธ์ดังรูป



6. ให้ทดลองทำการเปลี่ยนค่าความต้านทานจาก 220 Ω ไปเป็น 1K Ω แล้วให้อธิบายผลที่ได้เป็นอย่างไร

ความสว่างของหลอดไฟลดลง

7. ให้แสดงวิธีการคำนวณหาค่าความต้านทานที่เหมาะสม เมื่อกำหนดให้ Forward Current ของ LED เท่ากับ 20 mA

$R = \frac{V}{I} = \frac{5 - 1.26}{20 \times 10^{-3}} = \frac{3.74}{2} \times 10^2$

$= 1.87 \times 10^2\ \Omega$

8. ให้ทำการแก้ไขโปรแกรมโดยการเปลี่ยนค่า delay( ) เพื่อให้ LED ติด สว่าง 0.5 วินาที และดับ 1.5 วินาที จากนั้นให้ LED กระพริบเร็วขึ้นเรื่อยๆ ตามลำดับจนกว่าเราจะไม่เห็นการกระพริบ โดยใช้คำสั่ง for (.....)

9. จากข้อ 8 ค่าความถี่ในขณะที่เริ่มจะไม่เห็น LED กระพริบคือความถี่เท่าไร

$T_1 = T_{ON} + T_{OFF} = 20+60 = 80$ ms. $f = \frac{1}{T_1} = \frac{1}{80\times10^{-3}} = 12.5$ Hz

10. จากข้อ 8 ค่าความสว่างของ LED ในขณะที่เริ่มจะไม่เห็น LED กระพริบ ความสว่างนั้นเท่าเดิมหรือน้อยลง และให้เหตุผลว่าทำไมจึงเป็นเช่นนั้น

ความสว่าง น้อยลง เพราะ เมื่อสัมผัสกระพริบเร็วขึ้น เรื่อยๆ กระแสไฟ (แรงดันไฟ) ไปน้อย

11. ให้ต่อ LED หลอดที่ 2 อนุกรมกับตัวความต้านทาน 220 Ω เข้ากับขาที่ 10 แล้วลงกราวด์

12. ให้เขียนโปรแกรมให้ LED ขาที่ 9 กระพริบเป็นความถี่ 1 Hz และให้ LED ขาที่ 10 กระพริบเป็นความถี่ 2 Hz

13. ให้ต่อ LED อนุกรมกับตัวความต้านทาน 220 Ω เพิ่มอีกเป็นจำนวน 5 หลอด แล้วให้เขียนโปรแกรมควบคุมให้หลอดไฟ LED กระพริบไล่จากขวาไปซ้าย แล้วกระพริบไล่จากซ้ายสุดและขวาสลับกันไปมา โดยใช้คำสั่ง for (.....)

การสื่อสารกับอุปกรณ์ภายนอกของบอร์ด Arduino จะใช้พอร์ตที่เรียกว่าพอร์ตอนุกรม (Serial Port) ในการเชื่อมต่อกับอุปกรณ์อื่น หรือสื่อสารระหว่างไมโครคอนโทรลเลอร์กับคอมพิวเตอร์ การสื่อสารนี้เรียกว่า UART โดยจะใช้ขาหมายเลข 0 (RX) ในการรับค่า และขาหมายเลข 1 (TX) ในการส่งค่า คำสั่งต่างๆที่จำเป็นมีดังนี้

void serial.begin(rate) เป็นการกำหนดอัตราของการรับส่งข้อมูล หน่วยเป็น bits per second (baud rate)

int serial.available() ใช้ตรวจสอบว่าบัฟเฟอร์รับข้อมูลไว้จำนวนกี่ไบต์

int serial.read() อ่านค่าข้อมูลที่ถูกส่งเข้ามายังพอร์ตอนุกรมของไมโครคอนโทรลเลอร์

void Serial.flush() เคลียร์บัฟเฟอร์ของพอร์ตอนุกรมให้ว่าง

void Setial.print() พิมพ์ข้อมูล ออกทางพอร์ตอนุกรม

void Setial.println() พิมพ์ข้อมูล ออกทางพอร์ตอนุกรมและขึ้นบรรทัดใหม่

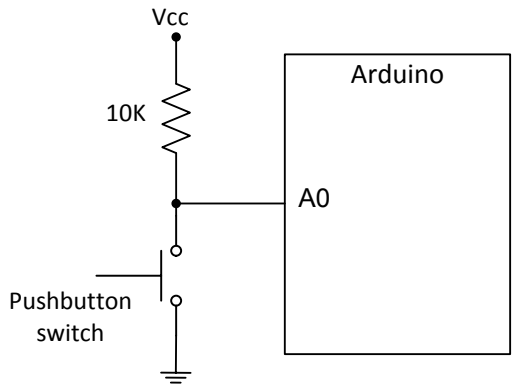
14. ให้เพิ่มคำสั่ง `Serial.begin(9600);` *// initialize serial communication at 9600 bits per second*
ลงใน void setup() เพื่อใช้กำหนดอัตราความเร็วในการรับส่งข้อมูลผ่าน Serial Monitor มีค่าเท่ากับ 9600 bps

15. ให้เพิ่มคำสั่ง `int Temp = analogRead(A0);` *// read the input on analog pin 0*
ลงใน void loop() เพื่อใช้รับค่าสัญญาณอนาล็อกจากขา A0 ของบอร์ด Arduino และแปลงค่าที่ได้ไปเป็นสัญญาณดิจิตอลขนาด 10 บิท แล้วเก็บไว้ที่ตัวแปร Temp ซึ่งค่าที่ได้จะอยู่ระหว่าง 0 ถึง 1023 (คำนวณได้จาก 2^{10})

16. ให้เพิ่มคำสั่ง `Serial.println(Temp);` *// print out the value*
ต่อจากคำสั่งในข้อ 15 เพื่อให้พิมพ์ผลลัพธ์ค่าข้อมูลตัวแปร Temp ส่งออกไปทาง Serial Monitor

การทดลองเขียนโปรแกรมเพื่อสั่งงานให้ไมโครคอนโทรลเลอร์อ่านค่าสัญญาณของขาที่ทำการเชื่อมต่ออยู่กับวงจรที่เป็นอุปกรณ์ภายนอก เมื่อมีการกำหนดให้ขาใดขาหนึ่งของไมโครคอนโทรลเลอร์ทำหน้าที่เป็น Input ด้วยคำสั่ง pinMode ก็สามารถใช้คำสั่ง digitalRead เพื่อสั่งให้ไมโครคอนโทรลเลอร์อ่านค่าสัญญาณที่เป็นแบบดิจิตอลเข้ามาจากอุปกรณ์ที่เชื่อมต่ออยู่กับขานั้นๆได้ ด้วยการใช้คำสั่ง digitalRead(pin) โดยที่ pin เป็นค่าของหมายเลขขาดิจิตอลที่ต้องการอ่านค่าว่าเป็นสัญญาณ HIGH หรือ LOW ในบอร์ด Arduino จะมีขาที่มีวงจรที่ทำหน้าที่แปลงสัญญาณสัญญาณอนาล็อกไปเป็นสัญญาณดิจิตอล หรือ Analog to Digital Converter (ADC) ขนาด 10 บิท ซึ่งจะใช้ในการอ่านค่าของสัญญาณที่เป็นแบบอนาล็อกเข้ามาจากวงจรภายนอกหรือเซนเซอร์ต่างๆที่เป็นแบบอนาล็อกที่เชื่อมต่ออยู่ ซึ่งจะต้องใช้เป็นคำสั่ง analogRead(pin) โดยที่ pin จะเป็นหมายเลขขาอินพุทที่เป็นสัญญาณอนาล็อกซึ่งจะขึ้นต้นด้วย A ใน Arduino จะมีขาที่เป็นอนาล็อกอยู่ทั้งหมด 8 ขา ซึ่งค่าของสัญญาณอนาล็อกที่อ่านได้นี้จะต้องถูกแปลงค่าจากสัญญาณอนาล็อกไปเป็นสัญญาณดิจิตอลขนาด 10 บิท ทำให้ได้ค่าที่อ่านออกมาทั้งหมดเท่ากับ 2^{10} ซึ่งค่าที่ได้จะอยู่ในช่วง 0 ถึง 1023 นอกจากนี้แล้วยังมีขา Analog Reference ใช้สำหรับอ้างอิงค่า Analog ในการเปรียบเทียบแรงดันแบบ Analog

17. ให้ต่อตัวความต้านทาน 10 K Ω อนุกรมกับสวิทช์ เข้ากับขา Vcc ของบอร์ด Arduino แล้วลงกราวด์ โดยให้ขา A0 ที่ทำหน้าทีเป็น Analog to Digital Converter ต่อเข้ากับจุดต่อร่วมระหว่างตัวความต้านทานกับสวิทช์



18. ให้เขียนโปรแกรมทดสอบการกดสวิทช์ โดยอ่านจากขา A0 แล้วให้บันทึกค่าที่ได้

เมื่อกดสวิทช์ Temp มีค่าเท่ากับ ..... 0 .....

เมื่อปล่อยสวิทช์ Temp มีค่าเท่ากับ ..... ~1023 .....

19. ให้แก้ไขโปรแกรมในข้อ 18 โดยกำหนดให้เมื่อกดสวิทช์ให้ LED ขาที่ 9 จะสว่าง และเมื่อปล่อยสวิทช์ ให้ LED ขาที่ 10 สว่าง โดยใช้คำสั่ง if (.....) else

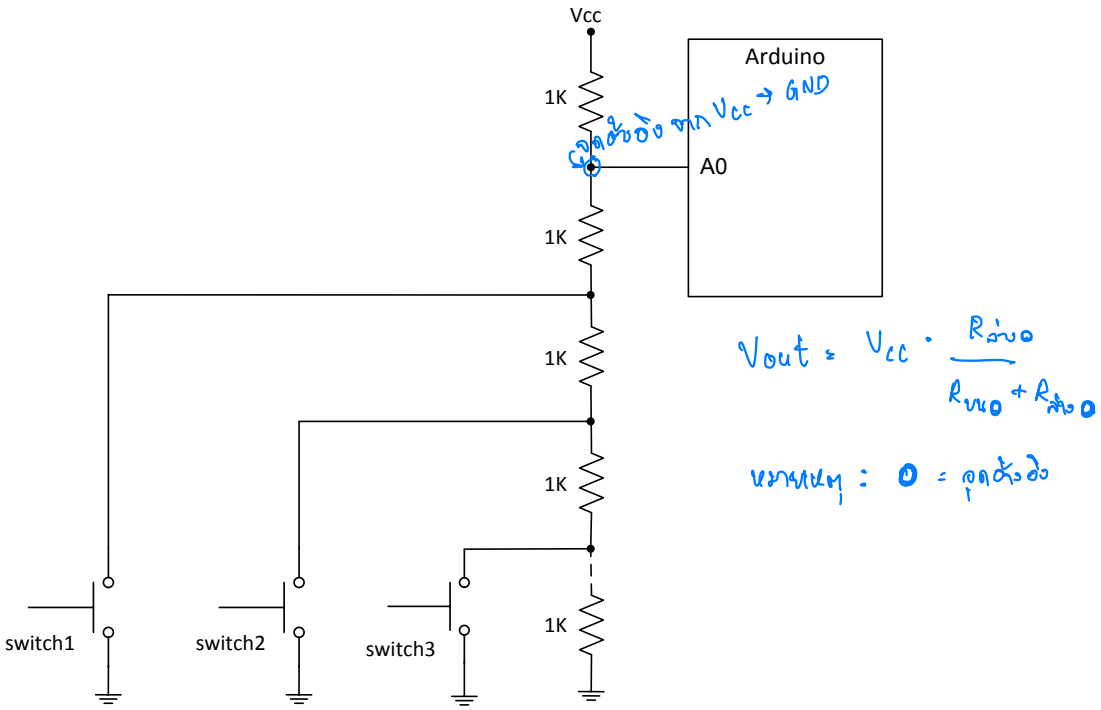
20. ให้ทดลองต่อตัวความต้านทาน 1 KΩ จำนวน 5 ตัวอนุกรมกันแล้วต่อเข้ากับขา Vcc ของบอร์ด Arduino เพื่อทำวงจรแบ่งแรงดันไฟฟ้า (Voltage Divider) และให้ขา A0 ต่อเข้ากับจุดต่อร่วมระหว่างตัวความต้านทานจุดแรก แล้วให้ใช้สวิทช์ 3 ตัวต่อเข้ากับจุดต่อร่วมของตัวความต้านทานที่เหลือแล้วลงกราวด์ โดยกำหนดให้ค่าที่อ่านออกมาได้ไม่ให้ซ้ำกัน แล้วบันทึกผลที่ได้

เมื่อไม่กดสวิทช์ Temp มีค่าเท่ากับ ..... 819, 820 .....

เมื่อกดสวิทช์ตัวที่ 1 Temp มีค่าเท่ากับ ..... 513, 514 .....

เมื่อกดสวิทช์ตัวที่ 2 Temp มีค่าเท่ากับ ..... 687, 688 .....

เมื่อกดสวิทช์ตัวที่ 3 Temp มีค่าเท่ากับ ..... 768, 769 .....



21. จากข้อ 20 ให้แสดงวิธีคำนวณหาค่า A0 ที่ได้จากวงจรแบ่งแรงดันไฟฟ้า (Voltage Divider) เมื่อกำหนดเงื่อนไขไว้ดังนี้

เมื่อไม่กดสวิทช์ A0 มีค่าแรงดันไฟฟ้าเท่าไร ?

Vout = 5 × 4/5 = 4 V.

เมื่อกดสวิทช์ตัวที่ 1 A0 มีค่าแรงดันไฟฟ้าเท่าไร ?

Vout = 5 × 1/2 = 2.5 V.

เมื่อกดสวิทช์ตัวที่ 2 A0 มีค่าแรงดันไฟฟ้าเท่าไร ?

Vout = 5 × 2/3 = 3.33 V.

เมื่อกดสวิทช์ตัวที่ 3 A0 มีค่าแรงดันไฟฟ้าเท่าไร ?

$V_{out} = 5 \times \frac{3}{4} = 3.75$ V.

22. ให้อธิบายว่าค่า Temp ในข้อ 20 กับค่า A0 ในข้อ 21 ว่ามีความสัมพันธ์กันอย่างไร และถ้ากำหนดให้ A0 ที่ได้จากวงจรแบ่งแรงดันไฟฟ้า (Voltage Divider) มีค่าเท่ากับ 2 V จงคำนวณหาค่าตัวแปร Temp ที่ได้จากขา Analog to Digital Converter ของ Arduino ว่าจะอ่านเข้ามามีค่าเท่ากับเท่าไร

$Temp = \frac{V_{A0}}{V_{CC}} \times 1023$

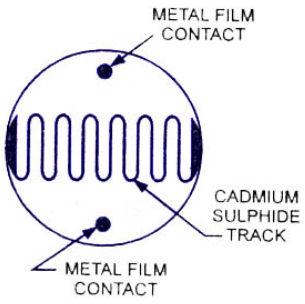
$= \frac{2}{5} \times 1023$

$= 2 \times 204.6$

$= 409.2$

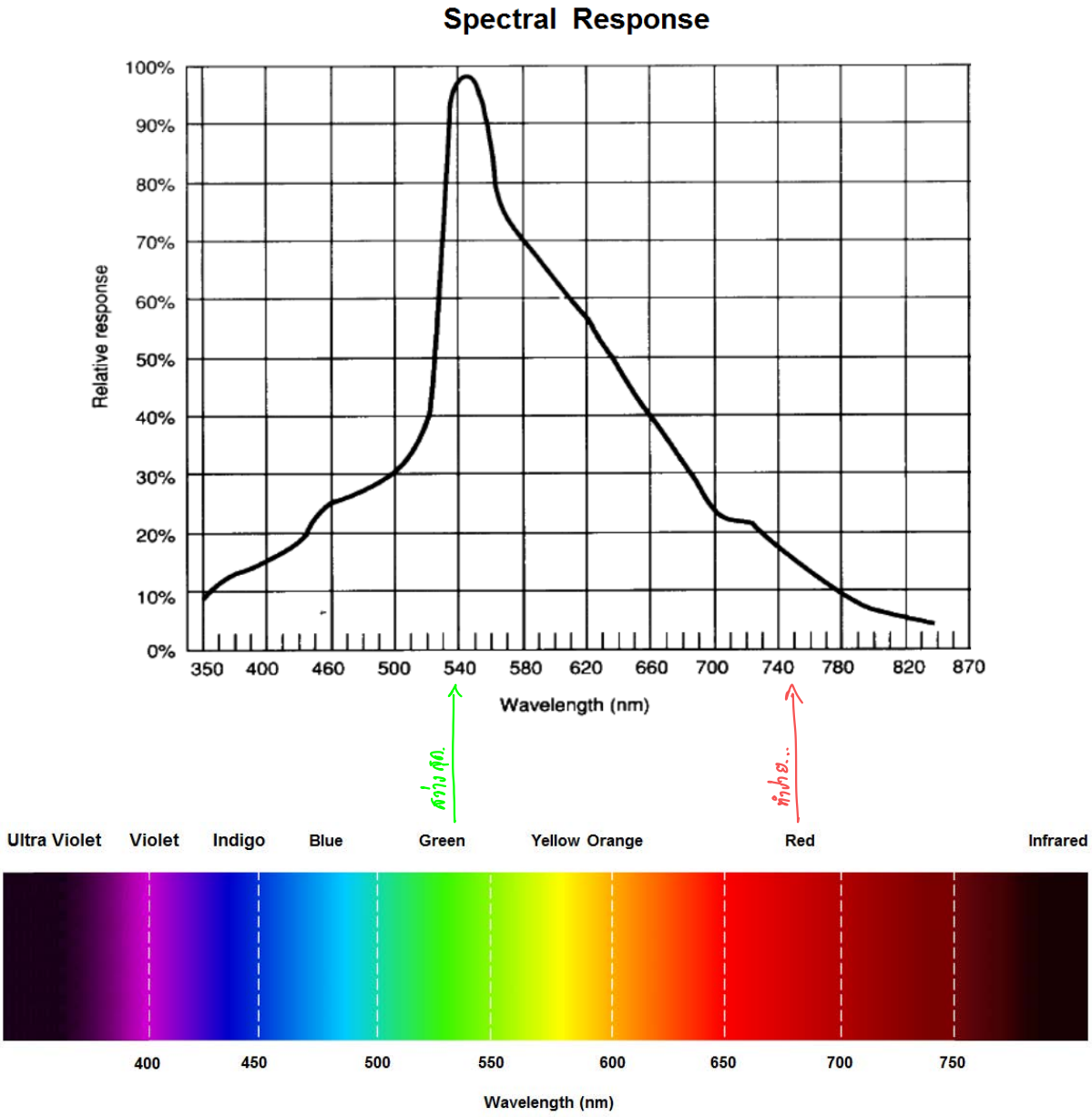
23. จากข้อ 20 ให้เขียนโปรแกรมที่มีข้อกำหนดคือ เมื่อกดสวิทช์ตัวที่ 1 ให้ LED ขาที่ 9 ติดสว่าง ถ้ากดสวิทช์ตัวที่ 2 ให้ LED ขาที่ 10 ติดสว่าง และถ้ากดสวิทช์ตัวที่ 3 ให้ LED ขาที่ 11 ติดสว่าง
โดยให้ใช้คำสั่ง switch (.....) case หรือ if (.....) else ก็ได้

LDR (Light Dependent Resistor) เป็นตัวต้านทานที่เปลี่ยนค่าความนำไฟฟ้าได้เมื่อมีแสงมากระทบ หรือเรียกว่าโฟโตรีซีสเตอร์ (Photo Resistor) ทำมาจากสารกึ่งตัวนำ (Semiconductor) ประเภทแคดเมี่ยมซัลไฟด์ (Cadmium Sulfide) หรือแคดเมี่ยมซิลินายส์ (Cadmium Selenide) ซึ่งเป็นสารประเภทกึ่งตัวนำที่เอามาฉาบลงบนแผ่นเซรามิกที่ใช้เป็นฐานรองแล้วต่อขาจากสารที่ฉาบนั้นออกมา เมื่อมีแสงตกกระทบลงบนสารกึ่งตัวนำที่ฉาบอยู่ นี้จะถ่ายทอดพลังงานทำให้เกิดโฮลกับอิเล็กตรอนอิสระวิ่งพล่านกันมากเป็นผลให้ค่าความต้านทานลดลง

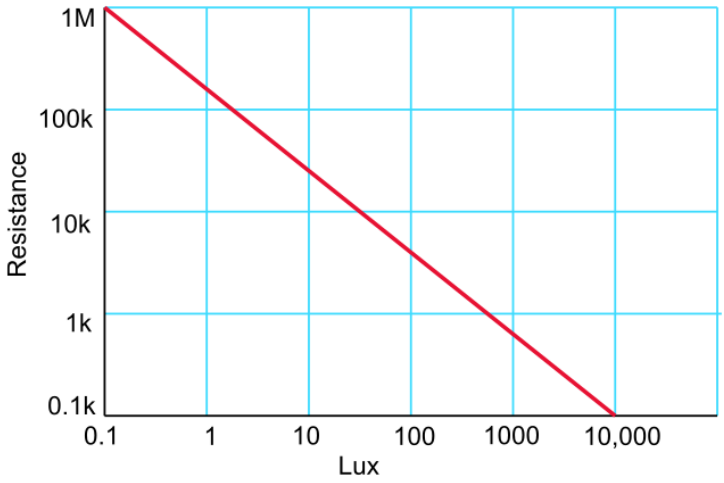


LDR Basic Structure

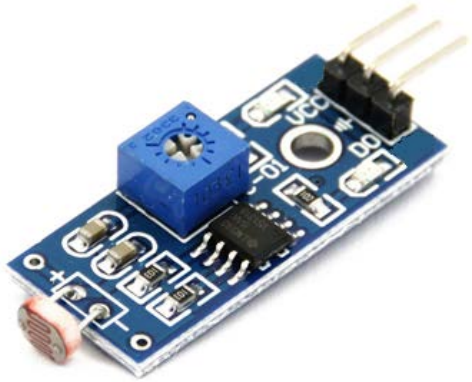
LDR ที่ทำจากแคดเมียมซัลไฟด์ จะมีตอบสนองทางสเปกตรัมต่อแสงได้ดีที่ช่วงความยาวคลื่น (Wavelength) ประมาณห้าร้อยห้าสิบนาโนเมตร (nm) ซึ่งจะเป็นช่วงที่อยู่ระหว่างแสงสีเขียวกับสีเหลือง และในช่วงแสงสีแดงผลการตอบสนองจะลดลงเหลือเพียง 30% เมื่อเทียบกับแสงสีเขียว ดังแสดงในกราฟด้านล่าง



เมื่อไม่มีแสงมาตกกระทบในสภาวะมืดจะทำให้ค่าความต้านทานระหว่างขั้วจะสูงถึง 1 MΩ หรือมากกว่านั้น ความต้านทานจะลดลงตามระดับแสงที่เพิ่มขึ้นและจะลดลงเหลือไม่กี่ร้อยโอห์มที่ความสว่างสูงดังแสดงในรูปกราฟ แต่อุปกรณ์ชนิดนี้ยังมีผลตอบสนองทางเวลา (Response time) ที่ไม่ดีนักคือจะช้ากว่าอุปกรณ์พวกโฟโต้ทรานซิสเตอร์มาก โดยจะมีค่าอยู่ในช่วงประมาณ 2 ถึง 50 mSec รายละเอียดต่างๆเพิ่มเติมให้เปิดดูได้จากเอกสาร LDR 3190 DataSheet

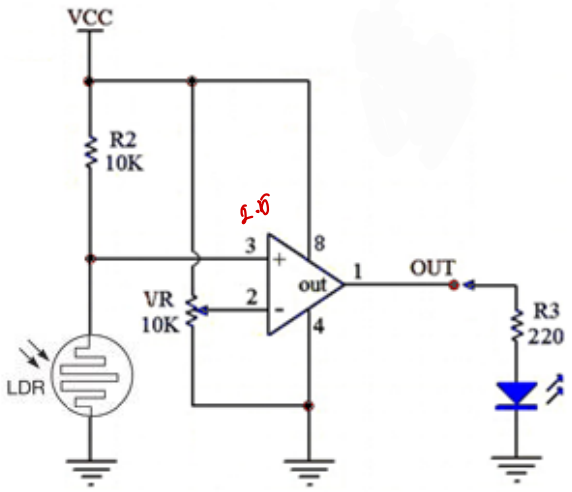


LDR มักนิยมใช้ในวงจรสวิทช์ที่เปิด-ปิดไฟด้วยแสง ตัวอย่างตามรูป วงจรภายในมักจะใช้ Op-Amp ทำเป็นวงจร Comparator โดยมีตัวต้านทานปรับค่าได้ ทำหน้าที่ปรับระดับแสงที่ต้องการให้เปิดปิดไฟ



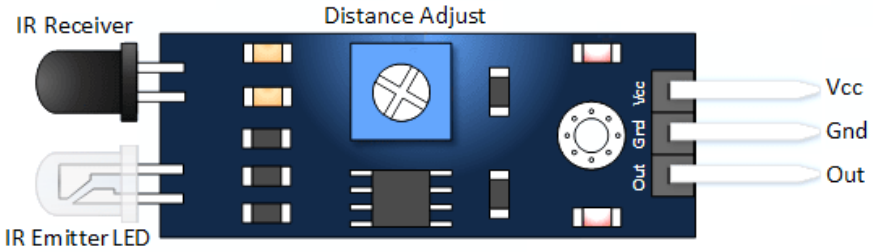
LDR Photoresistor Light Detection Sensor Module

24. การทดลองจะไม่ใช้ Op-Amp แต่จะใช้การเขียนโปรแกรมเพื่อเปรียบเทียบบน Arduino แทน โดยนำสัญญาณที่ต่อเข้าขา 3 ของ Op-Amp ไปป้อนเข้า Analog to Digital Converter (ADC) ขาอนาล็อกอินพุท A0 บนบอร์ด Arduino แทน ซึ่งจะทำให้ได้ค่าเป็นสัญญาณดิจิตอลขนาด 10 บิท ในช่วงระหว่าง 0 ถึง 1023 หรือก็คือได้ค่าทั้งหมดเท่ากับ 2^{10}

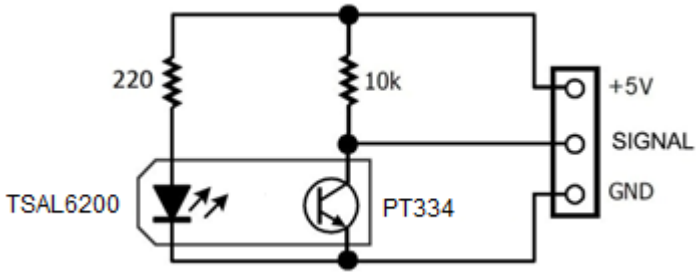


25. ให้นักศึกษาทำการแก้ไขโปรแกรมในข้อ 23 เพื่อทำเป็นเครื่องวัดความเข้มของแสงสว่าง โดยใช้ LDR เป็นเซ็นเซอร์วัดแสง และให้แสดงผลออกมาเป็นสีต่างๆ 7 สี ด้วย LED ที่เป็นแม่สี 3 สีคือ RGB โดย LED ที่ใช้ในการทดลองนี้จะต้องให้ขาของ LED ที่เป็นแม่สีทั้ง 3 ขาต่อกับตัวความต้านทาน 220 Ω และต่อเข้ากับขา 9, 10, 11 ของ Arduino ตามลำดับ แล้วให้ขา Common Cathode ที่เป็นจุดร่วมซึ่งจะเป็นขาที่ยาวที่สุดให้ต่อลงกราวด์ (GND)

การทดลองนี้จะเหมือนวงจรเซ็นเซอร์ตรวจจับแสงที่ใช้ Op-Amp ทำเป็นวงจร Comparator มาแก้ไข โดยใช้ Infrared Emitting Diode TSAL6200 ที่ทำหน้าที่ส่งแสงความยาวคลื่น 940 นาโนเมตรออกไป และใช้ Phototransistor ชนิด NPN silicon เป็นอินฟาเรดเซ็นเซอร์ ช่วง 840-1200 nm ทำหน้าที่รับแสงที่สะท้อนเข้ามา เพื่อใช้ในการวัดระยะห่าง รายละเอียดต่างๆของอุปกรณ์ทั้งสองชนิดนี้ให้เปิดดูได้จากเอกสาร Infrared Emitting Diode TSAL6200 DataSheet และ Phototransistor PT334 DataSheet



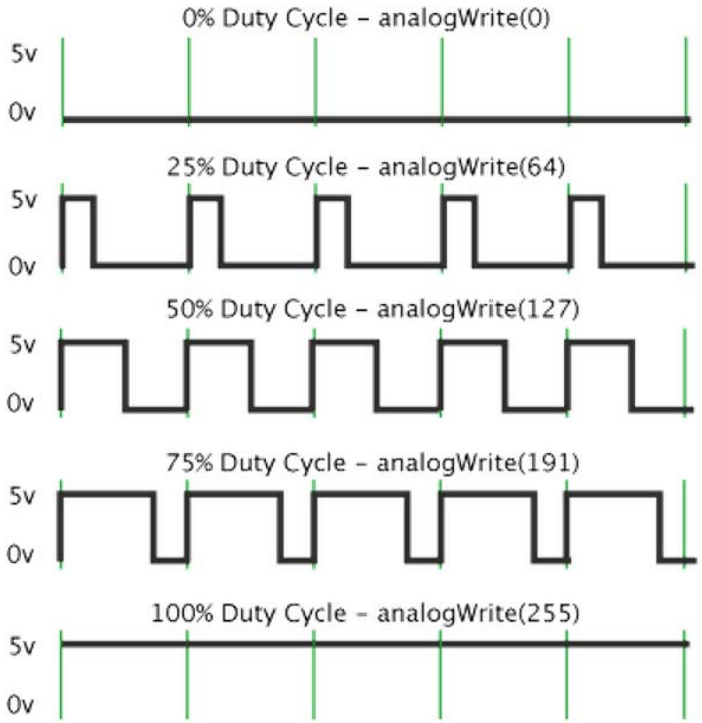
26. ให้ต่อวงจรเซ็นเซอร์ที่มีหน้าที่ตรวจจับแสงอินฟาเรดที่สะท้อนเพื่อใช้ในการวัดระยะห่างตามรูปด้านล่าง แล้วนำสัญญาณ SIGNAL ที่ได้ป้อนเข้า ADC ขาอนาล็อกอินพุท A1 ของบอร์ด Arduino



27. ให้นักศึกษาทำการแก้ไขโปรแกรมในข้อ 23 เพื่อทำเป็นเครื่องตรวจธนบัตรปลอมแบบอัตโนมัติ โดยเพิ่มวงจรใช้ Ultra Violet (UV) LED ที่มีช่วงความยาวคลื่น 390 นาโนเมตร ต่ออนุกรมกันตัวความต้านทาน 100 โอห์ม เมื่อนำธนบัตรมาเข้าใกล้ให้สั่งเปิด LED ที่เป็นแสง UV ให้ติดสว่าง

Pulse Width Modulation Pin

Arduino สามารถส่งสัญญาณที่เป็น Pulse width modulation หรือเรียกย่อว่า PWM ซึ่งเป็นเทคนิคในการสร้างสัญญาณอนาล็อกด้วยค่าเฉลี่ยของสัญญาณดิจิตอล ออกมาผ่านทางขา PWM ได้ ในบอร์ด Arduino Nano มีขา PWM ให้ใช้งานทั้งหมด 6 ขา แต่ละขาจะเป็นขนาด 8 bit โดยผู้ใช้สามารถสร้างความถี่ที่เป็นสัญญาณดิจิตอลรูปคลื่นสี่เหลี่ยม (square wave) พร้อมกับควบคุม Percent of Duty Cycle ได้ด้วยคำสั่ง analogWrite การกำหนดเพื่อปรับค่าดิวตี้ไซเคิล จะเป็นการควบคุมคาบเวลาของสัญญาณที่เป็นลอจิก 1 เทียบกับคาบเวลาที่เป็นลอจิก 0 ซึ่งจะทำให้ค่าแรงดันเฉลี่ยของสัญญาณที่จำลองเป็นค่าอนาล็อกต่างกันออกไป โดยค่าของดิวตี้ไซเคิลจะเรียกเป็นเปอร์เซ็น ตัวอย่างของคำสั่งตามรูป



รูปตัวอย่างสัญญาณ PWM

การทดลองจะเขียนโปรแกรมควบคุมให้ LED ที่อยู่บนบอร์ดไมโครคอนโทรลเลอร์ หรือ LED ที่เรานำมาต่อวงจรเพิ่มเข้าไป สามารถปรับความสว่างได้ ด้วยคำสั่งที่ใช้ในการสร้างสัญญาณ PWM คือ analogWrite คำสั่งนี้ขาที่กำหนดจะสร้างสัญญาณคลื่นสี่เหลี่ยมด้วย Duty Cycle ค่าหนึ่งตามที่กำหนด และจะไม่เปลี่ยนค่าสัญญาณจนกว่าจะมีการเรียกคำสั่ง analogWrite ในครั้งต่อไป การใช้คำสั่ง analogWrite ไม่จำเป็นจะต้องมีการกำหนดขาด้วยคำสั่ง pinMode ก่อน รูปแบบของคำสั่งคือ analogWrite(pin, value) โดยที่ pin คือขาที่ต้องการให้สร้างสัญญาณ PWM และ value จะเป็น เปอร์เซ็นต์ของ duty cycle ที่เราต้องการ โดย 0 หมายถึง 0 เปอร์เซ็นต์ และ 255 หมายถึง 100 เปอร์เซ็นต์

28. ให้ต่อวงจรใช้ตัวต้านทาน 220 ohm และ LED เข้าที่ขา D3 โดยขาอีกด้านหนึ่งให้ต่อลงกราวด์ และป้อนโปรแกรมดังตัวอย่างที่จะใช้คำสั่ง analogWrite ในการควบคุมปรับความสว่างของ LED

```
int led = 3;                                    // LED connected to digital pin 3
int fade = 5;                                   // how many points to fade the LED

void setup()
{
}
```

```
void loop()
{
 for(int brightness = 0;  brightness <= 255;  brightness +=fade)      // fade in from min to max
 {
   analogWrite(led, brightness);                                      // sets the brightness
   delay(30);
 }
 for(int brightness = 255;  brightness >= 0;  brightness -=fade)      // fade out from max to min
 {
   analogWrite(led, brightness);                                      // sets the brightness
   delay(30);
  }
}
```

29. ถ้าต้องการจะหรี่หลอดไฟ LED มีวิธีอะไรบ้าง

.....

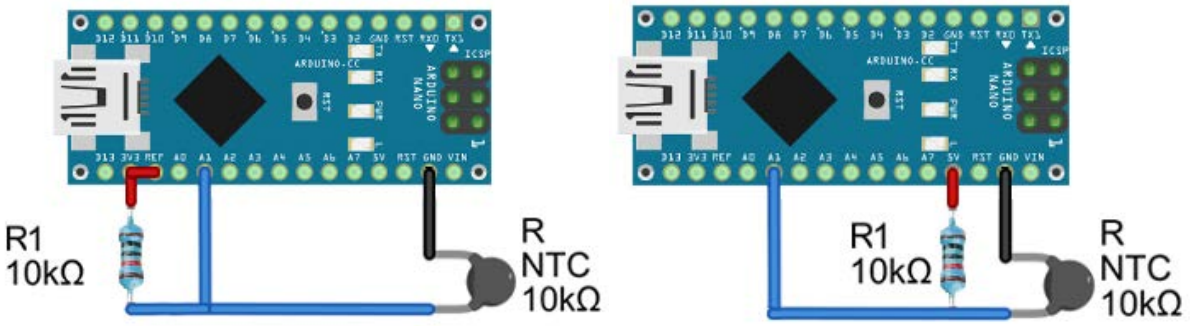
.....

.....

30. ให้ต่อ LED เพิ่มเป็นจำนวน 6 หลอด โดยใช้ขา PWM แล้วให้เขียนโปรแกรมควบคุมให้หลอดไฟแอลอีดีวิ่งแล้วหรี่ลักษณะเหมือนฝนดาวตก หรือมีรูปแบบอื่นๆ ตามที่ต้องการมาหลายๆแบบ

Temperature Sensor

การทดลองการวัดอุณหภูมิ จะใช้เซ็นเซอร์สำหรับวัดอุณหภูมิเป็นตัวเทอร์มิสเตอร์ ซึ่งจะเป็นตัวต้านทานที่มีค่าของความต้านทานเปลี่ยนแปลงตามอุณหภูมิ การทดลองจะต้องวัดค่าความต้านทานของเทอร์มิสเตอร์นั้นและเปลี่ยนค่าความต้านทานที่ได้ไปเป็นระดับแรงดันไฟฟ้า โดยต่อค่าแรงดันไฟฟ้าที่วัดได้ผ่านเข้าทางขาอนาล็อกของ Arduino และใช้การคำนวณค่าอุณหภูมิโดยสมการ Steinhart-Hart ซึ่งอธิบายค่า thermistor resistance – temperature curve โดยวงจรที่จะใช้ทดลองเป็นวงจรแบ่งแรงดันไฟฟ้าแสดงได้ตามรูป



การวัดแรงดันไฟฟ้า เราจะต้องเชื่อมต่อตัวเทอร์มิสเตอร์เข้ากับตัวต้านทาน R1 ขนาด 10KΩ 1% และถูกต่อเข้ากับแรงดันไฟฟ้า Vcc ของวงจร เพื่อทำวงจรแบ่งแรงดันไฟฟ้า ตัวความต้านทานของเทอร์มิสเตอร์ R จะใช้ NTC Thermistor เบอร์MF52-3435 มีความต้านทาน 10KΩ ที่ 25°C และมีความคลาดเคลื่อน 1% รายละเอียดต่างๆเพิ่มเติมให้เปิดดูได้จากเอกสาร Thermistor MF52_3435 DataSheet

กำหนดให้แรงดันขาเอาท์พุทเป็น Vo , แหล่งจ่ายไฟเป็น Vcc , ความต้านทานของตัวแปรเทอร์มิสเตอร์เป็น R และตัวความต้านทานคงที่เป็น R1 จะได้แรงดันขาเอาท์พุทคือ

$$Vo = Vcc \frac{R}{R + R1}$$

แรงดันขาเอาท์พุทเชื่อมต่อเข้ากับขาแบบอนาล็อก A1 ของ Arduino Micro เป็นวงจร ADC ทำหน้าที่แปลงสัญญาณอนาล็อกเป็นดิจิตอล (Analog to Digital Converter) ขนาด 10 บิต ซึ่งจะทำให้แรงดันไฟฟ้าถูกแปลงเป็นตัวเลขระหว่าง 0 ถึง 1023 ค่า ADC ที่วัดจาก Arduino Micro จะได้แรงดันขาเอาท์พุทดังนี้

$$Vo = Vcc \frac{A1}{1023}$$

โดยการแทนค่า Vo ทั้งสองสมการเข้าด้วยกันเป็น

$$Vcc \frac{R}{R + R1} = Vcc \frac{A1}{1023}$$

จะได้

$$\frac{R}{R + R1} = \frac{A1}{1023}$$

การวัดอุณหภูมิได้ตัวแปรความต้านทานเทอร์มิสเตอร์ R คือ

$$R = R1 \frac{A1}{1023 - A1} \quad \ldots(1)$$

เพื่อให้การวัดความต้านทานของเทอร์มิสเตอร์มีเสถียรภาพมากขึ้น ป้องกันไม่ให้ค่าที่วัดได้เปลี่ยนแปลงไปตามแรงดันไฟฟ้าของแหล่งจ่ายไฟที่มาจาก USB ของคอมพิวเตอร์ ซึ่งใช้เป็นแหล่งจ่ายพลังงานกับบอร์ดและวงจรต่างๆ ดังนั้นอาจจะมีสัญญาณรบกวนได้ จึงอาจจะใช้การเชื่อมต่อ Vcc กับขา Arduino 3V แทนขา 5V เพราะมันจะผ่านมาจากวงจรควบคุมแรงดันอีกครั้งและความถูกต้องของอุณหภูมิจะขึ้นอยู่กับแรงดันไฟฟ้าที่ต่ำกว่า

การทำวงจรจะต้องใช้อุปกรณ์ชิ้นส่วนอิเล็กทรอนิกส์แบบพาสซีฟทุกชิ้นที่มีค่าความคลาดเคลื่อนน้อยที่สุด เพราะจะมีความสัมพันธ์กับค่าที่อ่านออกมาเกิดความผิดพลาด ดังนั้นจึงให้เลือกใช้เทอร์มิสเตอร์ 10 KΩ ที่มีความคลาดเคลื่อน 1% ซึ่งจะมีผลให้ค่าความต้านทานเกิดความผิดพลาดได้สูงสุด 100 โอห์มที่อุณหภูมิ 25 องศาเซลเซียส โดยที่อุณหภูมิ 25 องศาเซลเซียสความแตกต่างของค่าความต้านทาน 450 โอห์มจะได้อุณหภูมิต่างกันประมาณ 1 องศาเซลเซียส ดังนั้นค่าความต้านทานที่มีความผิดพลาด 1% จะได้ความผิดพลาดของอุณหภูมิประมาณ 0.2 องศาเซลเซียส การแปลงค่าความต้านทานไปเป็นการวัดอุณหภูมิของเทอร์มิสเตอร์มีความสัมพันธ์ที่ค่อนข้างซับซ้อนระหว่างความต้านทานและอุณหภูมิ โดยทั่วไปแล้วสามารถใช้ตารางการค้นหาความต้านทานต่อการเปลี่ยนแปลงอุณหภูมิได้ตาม datasheet ของอุปกรณ์ได้ แต่ในที่นี้จะใช้ สมการ Steinhart-Hart (สมการพารามิเตอร์ B) ซึ่งเป็นการคำนวณค่าของความต้านทานเทอร์มิสเตอร์ที่มีความสัมพันธ์กับอุณหภูมิ จะได้

$$\frac{1}{T} = \frac{1}{T0} + \frac{1}{B} * ln \frac{R}{R0} \quad \ldots(2)$$

โดยที่ R เป็นความต้านทานของเทอร์มิสเตอร์ที่อุณหภูมิ T ในขณะนั้น

R0 คือความต้านทานที่ T0 = 25 ° C

B เป็นค่าคงที่ขึ้นอยู่กับเทอร์มิสเตอร์ ค่า B มักอยู่ระหว่าง 3000-4,000

สมการขึ้นอยู่กับพารามิเตอร์ (R0, T0 และ B) ซึ่งหาได้จาก datasheet ของ thermistor ที่ใช้

1. การทดลองจะใช้ตัวเซ็นเซอร์วัดอุณหภูมิด้วย Thermistor ขนาด 10K ohm เบอร์ NTC-MF52-103/3435 คลาดเคลื่อน 1% ต่อระหว่างขา A1 กับกราวด์ และใช้ตัวความต้านทาน R1 ขนาด 10K ohm 1% ต่อระหว่างขา A1 กับไฟบวก 5V เพื่อทำเครื่องวัดอุณหภูมิระบบดิจิตอล

2. ให้ทดลองป้อนโปรแกรมโดยกำหนดค่าต่างของอุปกรณ์ต่างๆ ซึ่งได้จาก datasheet ของ thermistor และอ่านค่าที่ได้จาก ADC ขา A1 แสดงผลออกไปทาง Serial Monitor ดังนี้

```
#define  THERMISTOR  A1                  // thermistor pin
#define  R0  10000                       // Ω resistance at 25 Celsius
#define  B   3435                        // B: 3435 K the beta coefficient of the thermistor
#define  R1  10000                       // 10KΩ the value of the series resistor

float T0 = 25;                           // °C reference temp.

void setup()
{
 T0 = T0 + 273.15;                       // conversion from Celsius to kelvin
 Serial.begin(9600);
 }

void loop() {
 int samples;

 samples = analogRead(THERMISTOR);       // read the input on analog pin 0
 Serial.print("Analog reading : ");      // print out the value
 Serial.println(samples);

 delay(1000);                            // Wait for next sample
}
```

3. ให้ทำการเพิ่มโปรแกรมการคำนวณเปลี่ยนค่าที่อ่านได้ จากขา A1 ตัวแปร sample ไปเป็นค่าความต้านทานของตัว Thermistor โดยใช้สมการที่ 1 และกำหนดให้ตัวแปรความต้านทานเทอร์มิสเตอร์ R เป็นชนิด float แล้วให้พิมพ์ผลที่ได้ออก Serial.print(R);

4. ให้เพิ่มโปรแกรมการคำนวณค่าของความต้านทานเทอร์มิสเตอร์ที่มีความสัมพันธ์กับอุณหภูมิ โดยใช้สมการที่ 2 ซึ่งจะได้ค่าอุณหภูมิของเทอร์มิสเตอร์ออกมาเป็นตัวแปร T โดยค่าอุณหภูมิที่ได้นี้จะเป็น kelvin ให้แปลงค่าเป็น Celsius และพิมพ์ผลที่ได้ออกไปทาง Serial Monitor

5. จากการทดลองจะเห็นได้ว่าค่าที่อ่านออกมาอาจจะกระโดดไปมาไม่นิ่ง ให้แก้ไขโปรแกรมเพิ่มการคำนวณหาค่าเฉลี่ยการวัดอุณหภูมิแสดงผลออกมาทุกครึ่งวินาที