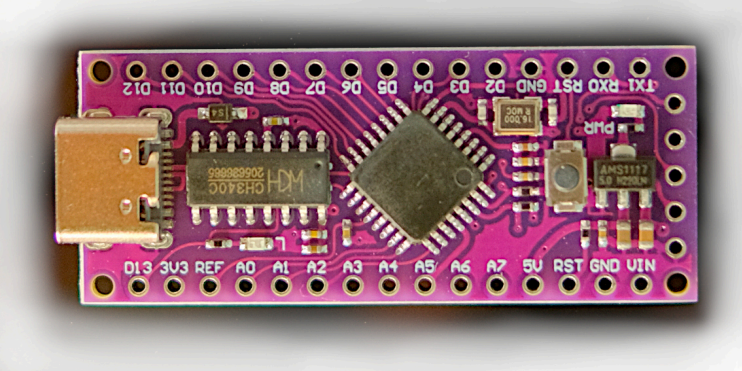
คู่มือการใช้
NANO F328-C



ภาพที่ 1 บอร์ด NANO F328-C

บอร์ด NANO F328-C (ภาพที่ 1) เป็นบอร์ดไมโครคอนโทรลเลอร์ของบริษัท Logic Green Technology รุ่น LGT8F328P ที่ถูกออกแบบให้เป็นสถาปัตยกรรมคอมพิวเตอร์ประเภท RISC (Reduce Instruction Set Computer) ที่ส่วนใหญ่จะทำงานจบใน 1 วงรอบการทำงาน (Clock Cycle)

นอกจากนี้ ชุดคำสั่งและโครงสร้างของสถาปัตยกรรมภายในชิพมีความเข้ากันได้กับไมโครคอนโทรลเลอร์ Atmel AVR atmega328p และตัวบอร์ดถูกออกแบบเพื่อใช้แทนบอร์ด Arduino Nano จึงจัดวางตำแหน่งขาต่าง ๆ ตรงกัน ทำให้สามารถนำบอร์ดไปใช้แทนบอร์ดเดิมได้โดยไม่ต้องออกแบบแผงวงจรใหม่

คุณสมบัติของบอร์ด NANO F328-C เป็นดังนี้

- เป็นไมโครคอนโทรลเลอร์ 8 บิต ที่ใช้สถาปัตยกรรมแบบ RISC
- ทำงานที่ความถี่สัญญาณนาฬิกา 32MHz
- มีชุดคำสั่งภาษาแอสเซมบลีจำนวน 131 ชุดคำสั่ง
- มีเรจิสเตอร์ขนาด 8 บิตจำนวน 32 ตัว (R0 ถึง R31)
- หน่วยความจำรอมแบบแฟลชขนาด 32KB
- หน่วยความจำแรมแบบ SRAM ขนาด 2KB
- มีโมดูล Timer/Counter แบบ 8 บิตจำนวน 2 ชุด ที่สามารถทำงานแยกกันได้ในโหมด Prescaler หรือ Compare
- มีโมดูล Timer/Counter แบบ 16 บิตจำนวน 2 ชุดที่ทำงานแยกกันในโหมด Prescaler, Compare หรือ Capture

- มีขาทำงาน PWM (Pulse-Width-Modulate) จำนวน 9 ขา
- มีโมดูล ADC (Analog to Digital Converter) ความละเอียด 12 บิต หรือแปลงเป็นค่าดิจิทัลเป็น 0 ถึง 4095
- มีโมดูล DAC (Digital to Analog Converter) จำนวน 1 โมดูลที่ต่อเชื่อมเข้ากับขา 4 หรือ DAC0 ทำงานที่ความละเอียด 8 บิต
- มีภาค WDT (Watch Dog Timer) ที่สามารถโปรแกรมการทำงานได้
- รองรับการสื่อสารแบบ USART, SPI, I2C และ TWI
- ใช้ CH340Cเป็นชิพแปลงภาคเชื่อมต่อกับพอร์ต USB และรองรับการทำงานกับระบบปฏิบัติการ Windows, Linux และ macOS
- รองรับการนำหน่วยความจำรอมแบบ Flash ROM กันเป็นรอมสำหรับเก็บข้อมูลที่เรียกว่า EZPROM โดยมีสัดส่วนของการแบ่งเป็นดังนี้

ขนาดของ Flash ROM	ขนาดของ EZPROM
32KB	0KB
30KB	1KB
32KB	2KB
28KB	4KB
16KB	8KB

ความแตกต่างของไมโครคอนโทรลเลอร์ LGT8F328P กับ atmega328p ได้แก่

- มีโมดูล DAC ความละเอียด 8 บิต
- โมดูล ADC มีความละเอียด 12บิต แต่ atmeta328p มีความละเอียด 10บิต
- โมดูลเปรียบเทียบแอนาล็อกจำนวน 2 ตัว แต่ atmeta328p มีเพียง 1 ตัว
- มี UniqueID ของแต่ละชิพที่ไม่เหมือนกัน
- ขานำออก (output) จำนวน 27 ขา แต่ atmeta328p มี 20ขา
- ขานำเข้า(input) จำนวน 30 ขา แต่ atmeta328p มี 23 ขา
- ชุดคำสั่งที่มีความเร็วในการทำงานที่แตกต่างกันเป็นดังนี้

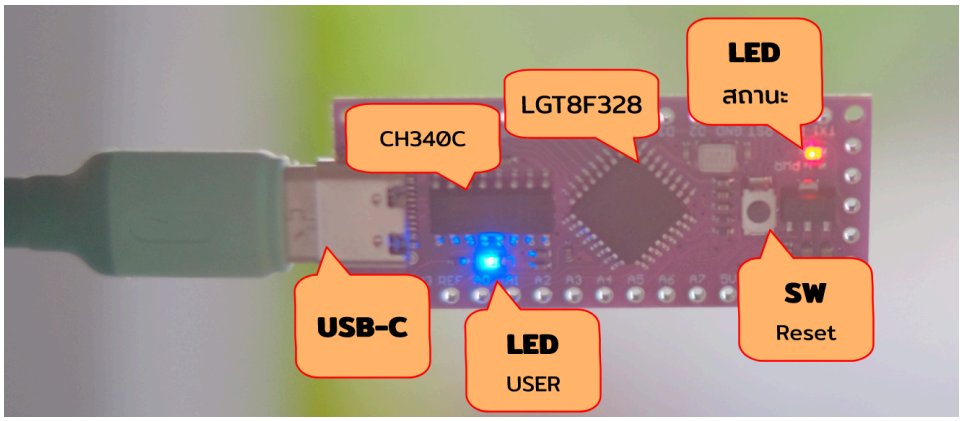
ชุดคำสั่ง	จำนวนสัญญาณนาฬิกา	
	ATMEGA328P	LGT8F328P
ADIW Add immediate to word	2	1
SBIW Subtract immediate to word	2	1
MUL / S / SU 8bit multiply	2	1
FMUL / S / SU Fractional multiply	2	1
RJMP / RCALL Relative jump / call	2-3	1
IJMP / ICALL Indirect jump / call	2-3	1
RET / IRET Return	4	2
CPSE Compare, skip if equal	1-3	1-2
SBIS / SBRS Skip if set	1-3	1-2
SBIC / SBRC Skip if cleared	1-3	1-2
LD / LDD Load indirect	2	1
ST / STD Store indirect	2	1
LPM Load program memory	3	2
PUSH / POP Stack access	2	1

นอกจากนี้ในชุดคำสั่งภาษาแอสเซมบลีของ LGT8F328P มีความแตกต่างกับ atmega328p ในเรื่องของความเร็วในนการทำงานที่ใช้จำนวนสัญญาณนาฬิกา (clock) น้อยลง ซึ่งผู้นำไปใช้จะต้องทำความเข้าใจเพื่อให้ระบบทำงานด้วยความแม่นยำ

การใช้งานบอร์ด

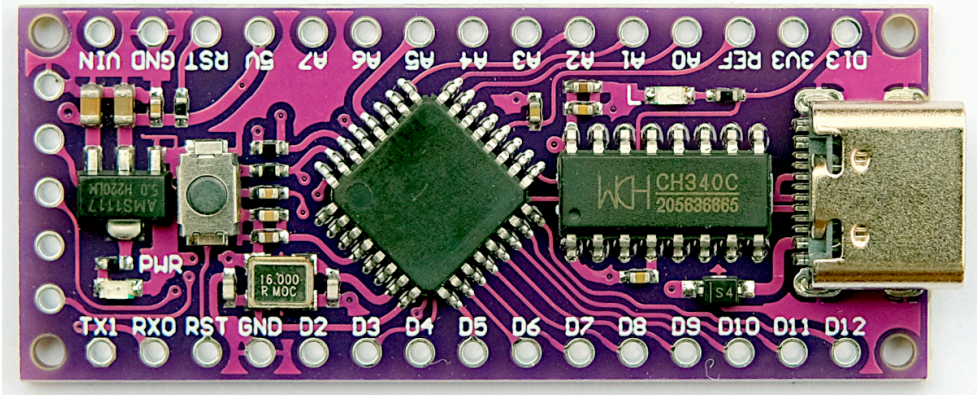
บอร์ด NANO F328-C มีส่วนประกอบหลัก ๆ ของบอร์ดเป็นดังภาพที่ 2 จะพบว่าต้องใช้หัวเชื่อมต่อประเภท USB-C โดยมีชิพ CH340C ทำหน้าที่แปลงสัญญาณสื่อสารผ่านพอร์ต USB บนบอร์ด ดังนั้น ผู้ใช้สามารถจ่ายไฟผ่านทาง USB-C หรือขา Vin ของบอร์ดเพื่อแปลงแรงดันไฟฟ้ากระแสตรง 5VDC เป็น 3V3 บนบอร์ด และเมื่อมีแรงดันเข้าบอร์ดจะทำให้หลอดแอลอีดีแสดงสถานะส่องสว่าง

กรณีที่ต้องการรีเซ็ตระบบสามารถกดที่สวิตช์ Reset บนบอร์ด หรือต่อวงจรรีเซ็ตต่อภายนอกเองได้โดยเชื่อมกับขา RST ที่แสดงในภาพที่ 5

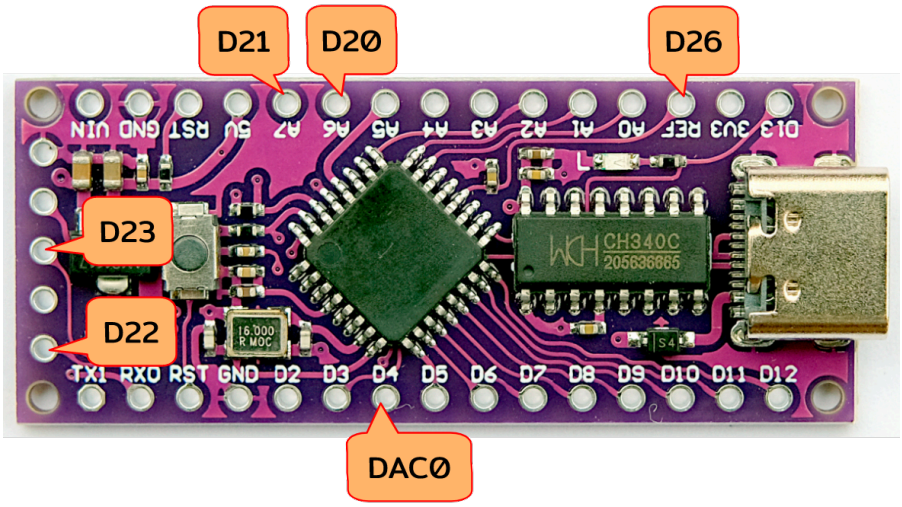


ภาพที่ 2 องค์ประกอบบนบอร์ด NANO F328-C

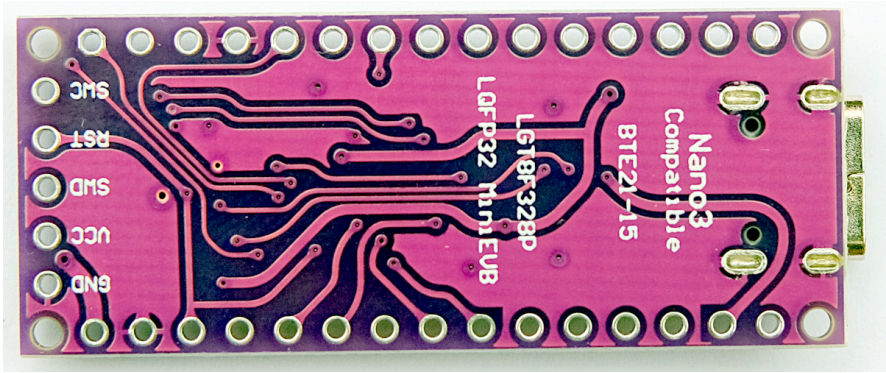
การจัดวางขาของบอร์ด NANO เป็นดังภาพที่ 3, 4 และด้านหลังของบอร์ดเป็นดังภาพที่ 5 ซึ่งจะพบว่าขาด้านข้างของบอร์ดนั้นเรียงตามบอร์ด Arduino Nano



ภาพที่ 3 ภาพด้านหน้าและชื่อขาของบอร์ด NANO F328-C



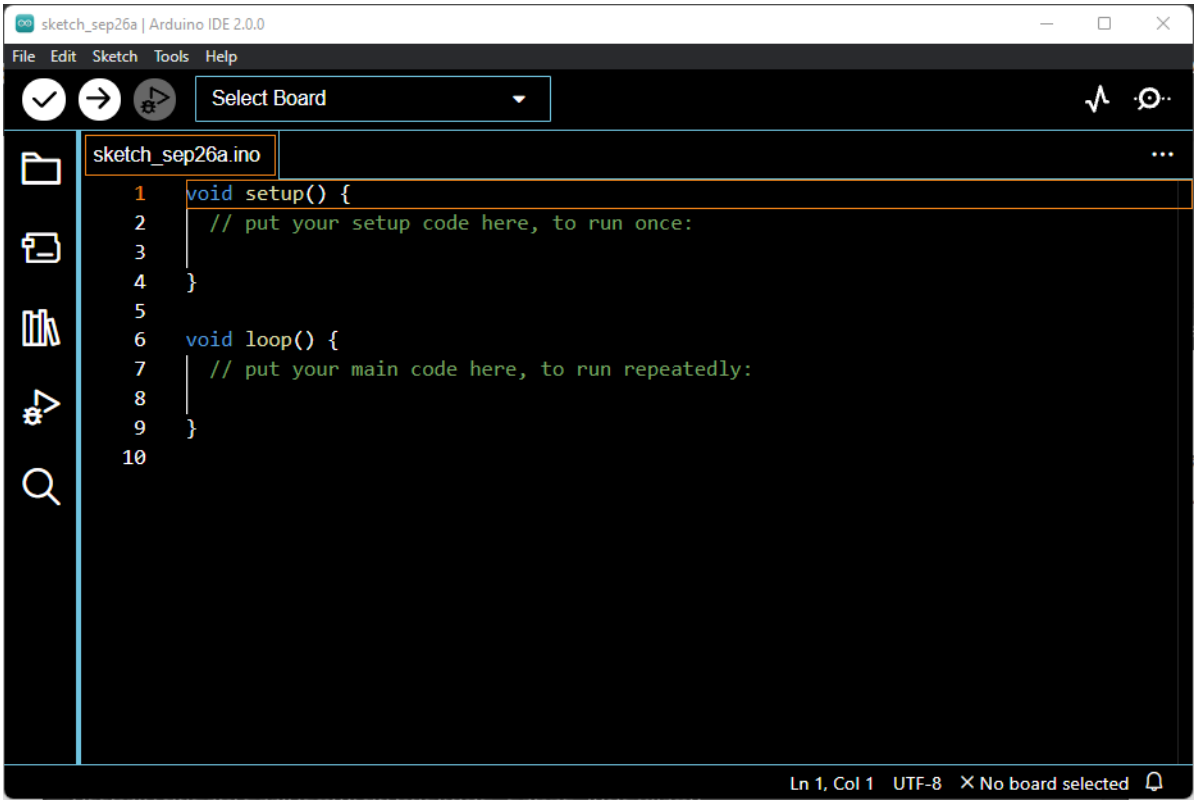
ภาพที่ 4 ภาพแสดงขา D20, D21, D22, D26 และ DAC0



ภาพที่ 5 ภาพด้านหลังของบอร์ด NANO F328-C

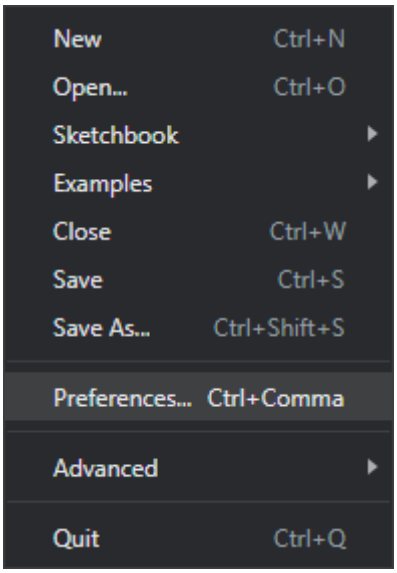
การพัฒนาซอฟต์แวร์

การติดตั้งชุดพัฒนาซอฟต์แวร์ของบอร์ด NANO F328-C เพื่อใช้กับ Arduino IDE รุ่น 2.0 (สามารถใช้อ้างอิงกับรุ่น 1.9.18 ได้) ในภาพที่ 6

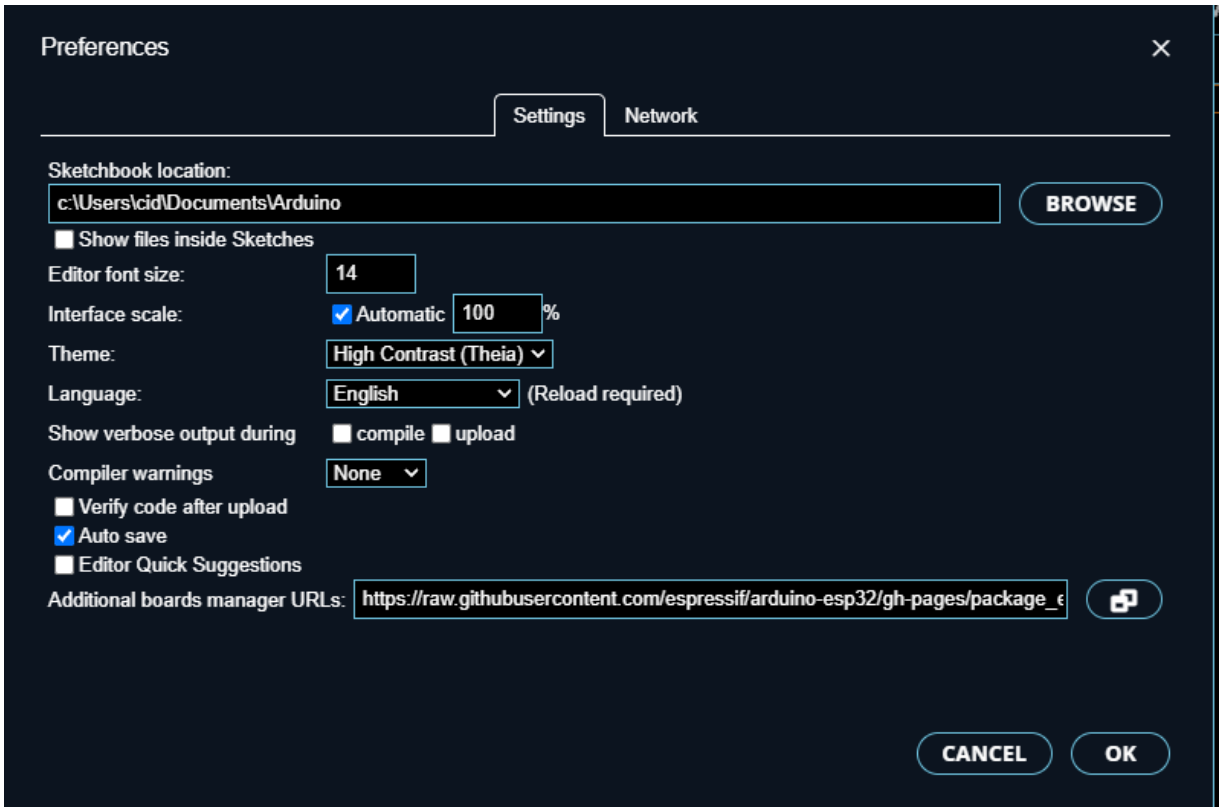


ภาพที่ 6 หน้าจอหลักของโปรแกรม Arduino IDE

จากหน้าจอหลักให้เลือกเมนู File เพื่อเลือกรายการ Preferences... ดังภาพที่ 7 หรือกดแป้น Ctrl ค้างไว้ตามด้วยกดที่แป้น comma หรือ , เพื่อเปิดหน้าต่างตั้งค่าดังภาพที่ 8



ภาพที่ 7 เมนูย่อยของ File

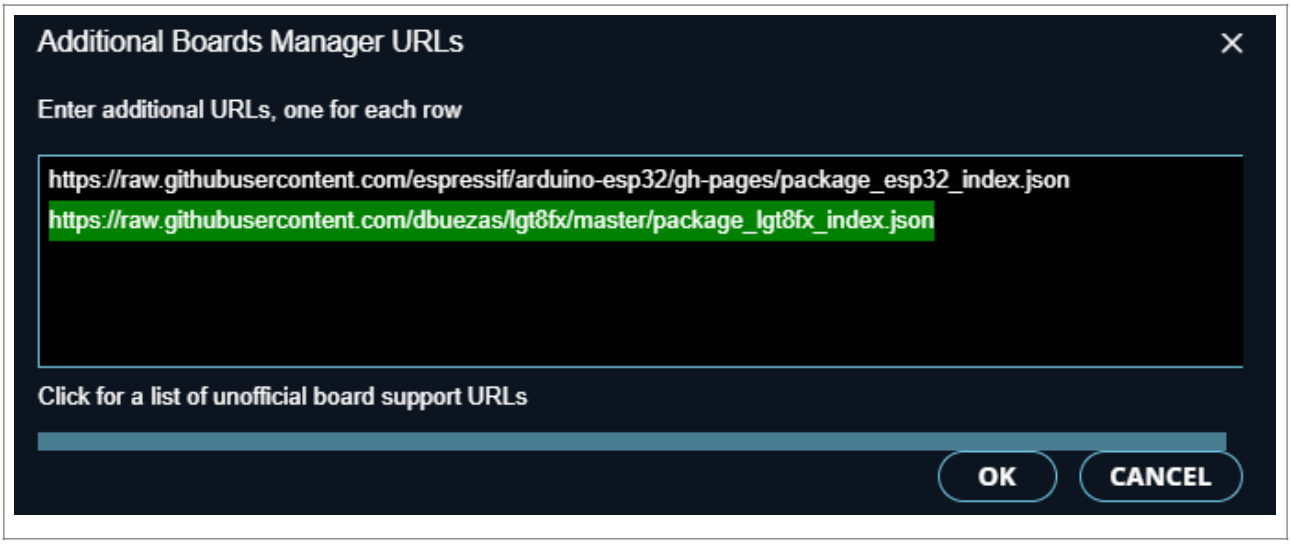


ภาพที่ 8 หน้าต่าง Preferences...

จากภาพที่ 8 ให้คลิกที่ไอคอนด้านหลัง Additional boards manager URLs ที่ปรากฏอยู่ด้านบนของปุ่ม OK เมื่อคลิกเสร็จจะมีหน้าต่างดังภาพที่ 9 ให้ผู้ใช้งานใส่แหล่งที่มาของข้อมูลบอร์ด

จากหน้าต่างนี้ให้เพิ่ม URL ของแหล่งที่มาสำหรับบอร์ด NANO F328-C ดังนี้

https://raw.githubusercontent.com/dbuezas/lgt8fx/master/package_lgt8fx_index.json



ภาพที่ 9 หน้าต่างใส่ URL ของบอร์ด

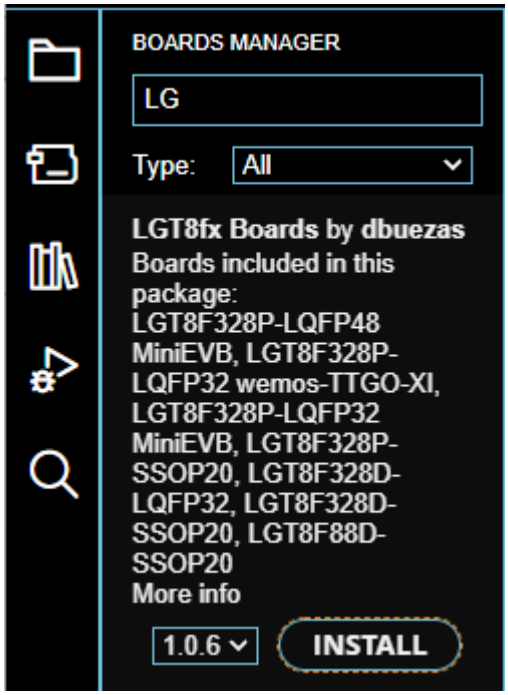
เมื่อกรอกเพิ่มเสร็จให้คลิกที่ปุ่ม OK เพื่อบันทึก และปิดหน้าต่างแล้วกลับไปยังหน้าต่างของภาพที่ 8 ให้คลิก OK อีกครั้งเพื่อกลับไปยังหน้าต่างโปรแกรม Arduino IDE ในภาพที่ 6

ภาพภาพที่ 6 ให้เข้าหน้าต่าง Boards Manager ด้วยการไปที่เมนู Tools/Board แล้วเลือก Boards Manager หรือคลิกที่ไอค่อน Board Manager ทางด้านซ้ายของหน้าต่าง Arduino IDE จะแสดงรายการของบอร์ดดังภาพที่ 10

จากภาพที่ 10 ให้พิมพ์คำค้นเป็น LG หรือถ้าเป็น Arduino IDE รุ่น 1.x จะแสดงหน้าต่างขึ้นมาแล้วให้ค้นหา LG จะแสดงรายการบอร์ด LGT8fx Boards by dbuezas ให้คลิกที่ปุ่ม INSTALL หลังจากนั้นจะเข้าสู่ขั้นตอนของการดาวน์โหลดเพื่อติดตั้งเครื่องมือพัฒนา ดังตัวอย่างรายงานการดาวน์โหลดในภาพที่ 11 ในขั้นตอนนี้ให้รอจนกว่าจะมีรายงานผลของการดาวน์โหลดเสร็จสิ้นดังตัวอย่างในภาพที่ 12 และตรงปุ่ม INSTALL จะเปลี่ยนเป็น UNINSTALL (หรือ UPGRADE เมื่อพบว่ามีรุ่นใหม่กว่า) แล้วในด้านหลังของชื่อบอร์ดจะแสดงคำว่า INSTALLED ในกรอบ เพื่อแสดงให้ผู้ใช้ทราบว่าบอร์ดนี้ถูกติดตั้งในเครื่องแล้ว ดังภาพที่ 13 เป็นอันเสร็จสิ้นขั้นตอนของการติดตั้งไฟล์ต่าง ๆ สำหรับบอร์ด NANO F328-C

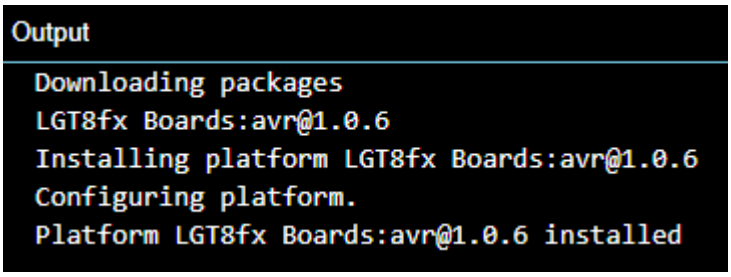
หมายเหตุ

ในขั้นตอนการติดตั้งบอร์ด NANO F328-C จะต้องเชื่อมต่ออินเทอร์เน็ตเพื่อดาวน์โหลดไฟล์จากแหล่ง URL ที่ได้ระบุ

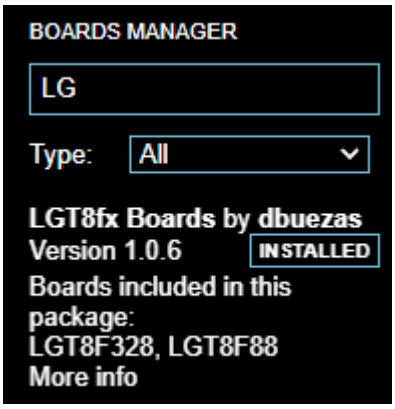


ภาพที่ 10 รายการบอร์ด LGT8fx Boards by dbuezaz

ภาพที่ 11 ขั้นตอนดาวน์โหลดไฟล์สำหรับบอร์ด NANO F328-C

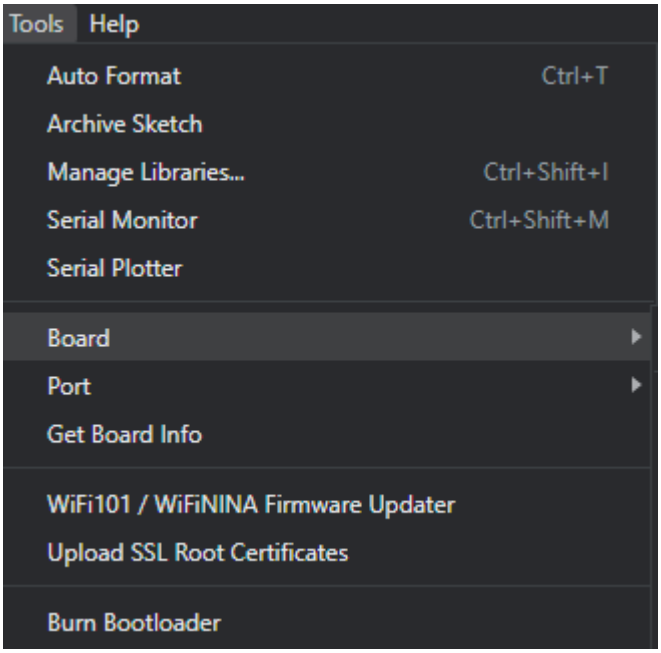


ภาพที่ 12 รายงานผลการติดตั้งไฟล์ของบอร์ด NANO F328-C ในหน้าต่าง Output



ภาพที่ 13 ตัวอย่างผลลัพธ์เมื่อติดตั้งเสร็จแล้ว

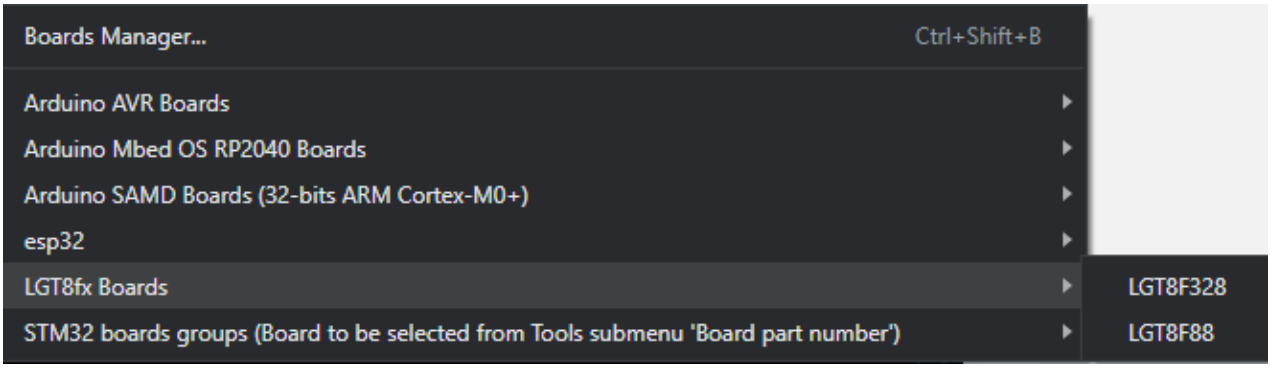
หลังจากติดตั้งไฟล์เครื่องมือและไฟล์ต่าง ๆ เพื่อใช้กับบอร์ด NANO F328-C เสร็จสิ้น ให้ผู้ใช้เลือกใช้งานบอร์ดด้วยการเข้าไปที่เมนู Tools แล้วเข้ารายการ Board ดังภาพที่ 14



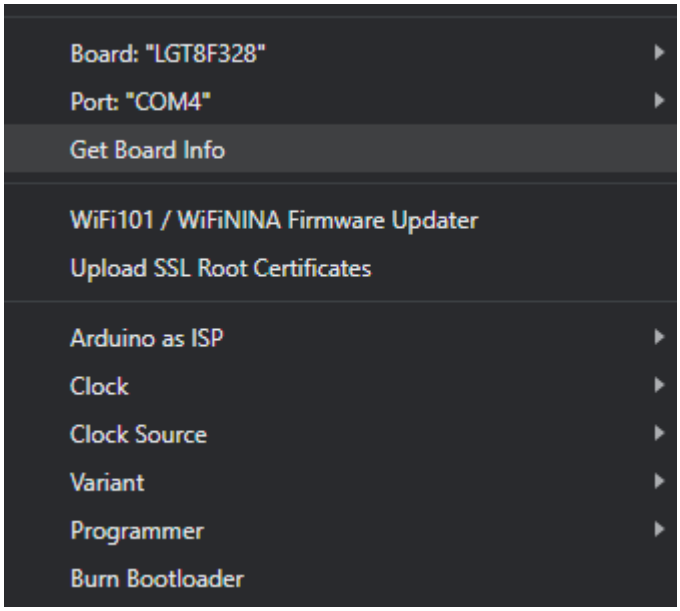
ภาพที่ 14 รายการเมนูย่อยของเมนู Tools

เมื่อเลือกเข้า Board ตามภาพที่ 14 จะแสดงรายการบอร์ดต่าง ๆ ที่ติดตั้งไว้ในเครื่อง ให้เลือกรายการ LGT8fx Board เพื่อเลือกรายการ LGT8F328 ดังภาพที่ 15 หลังจากนั้นรายการย่อยของเมนู Tools จะเปลี่ยนเป็นดังตัวอย่างภาพที่ 16

ในขั้นตอนนี้ต้องแน่ใจว่าได้เชื่อมต่อบอร์ด NANO F328-C เข้ากับพอร์ต USB ของเครื่องคอมพิวเตอร์แล้ว ดังนั้น ให้ไปที่เมนู Port สำหรับระบุพอร์ตที่เชื่อมต่ออยู่กับบอร์ด ดังตัวอย่างในภาพที่ 17 เพื่อใช้เป็นข้อมูลสำหรับการอัพโหลดโปรแกรมเข้าบอร์ด



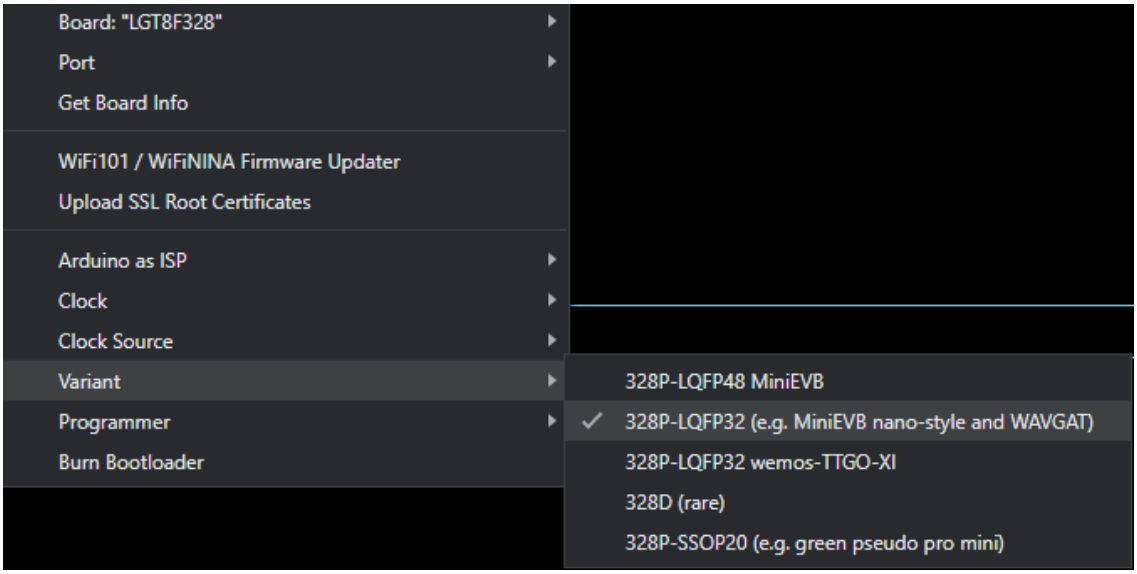
ภาพที่ 15 รายการเมนูของบอร์ด LGT8fx



ภาพที่ 16 รายการเมนูในเมนู Tools หลังจากเลือกบอร์ด LGT8F328

ภาพที่ 17 รายการพอร์ตสื่อสารที่เชื่อมต่อกับบอร์ด NANO F328-C

ขั้นตอนสุดท้ายของการติดตั้งค่าสำหรับ Arduino IDE คือเลือกประเภทของบอร์ดใช้งานด้วยการเข้าไปที่รายการย่อย Variant ของเมนู Tools จะแสดงรายการประเภทของบอร์ดให้เลือก ทั้งนี้บอร์ด NANO F328-C เป็นบอร์ดที่ออกแบบมาตามลักษณะของ Arduino Nano จึงให้เลือกประเภทของบอร์ดเป็น 328P LQFP32 (e.g. MiniEVB nano-style and WAVGAT) ดังตัวอย่างภาพที่ 18



ภาพที่ 18 รายการประเภทของบอร์ด

ตัวอย่างโปรแกรมกระพริบหลอดแอลอีดี

หลังจากเลือกประเภทของบอร์ดและเชื่อมต่อกับบอร์ดเป็นที่เรียบร้อยให้ทดลองใช้งานบอร์ดด้วยตัวอย่างโปรแกรมเบื้องต้นคือกระพริบหลอดแอลอีดีบนบอร์ด NANO F328-C ดังนี้

```
uint8_t status;

void setup() {
  status = HIGH;
  pinMode(LED_BUILTIN, OUTPUT);
}

void loop() {
  status = (HIGH - status);
  digitalWrite(LED_BUILTIN, status);
  delay(250);
}
```

เมื่อพิมพ์โค้ดเสร็จให้คลิกปุ่ม Compile เพื่อตรวจสอบความถูกต้องของโปรแกรม และเมื่อทุกอย่างถูกต้องไม่มีข้อผิดพลาดให้คลิก Upload เพื่ออัพโหลดโปรแกรมเข้าชิพ LGT8F328P และดูผลลัพธ์ของการทำงาน

กรณีที่ใช้ตัวตั้งเวลา (Timer) เป็นตัวกระตุ้นการสั่งเปิดหรือปิดการเปล่งแสงเขียนได้ดังนี้

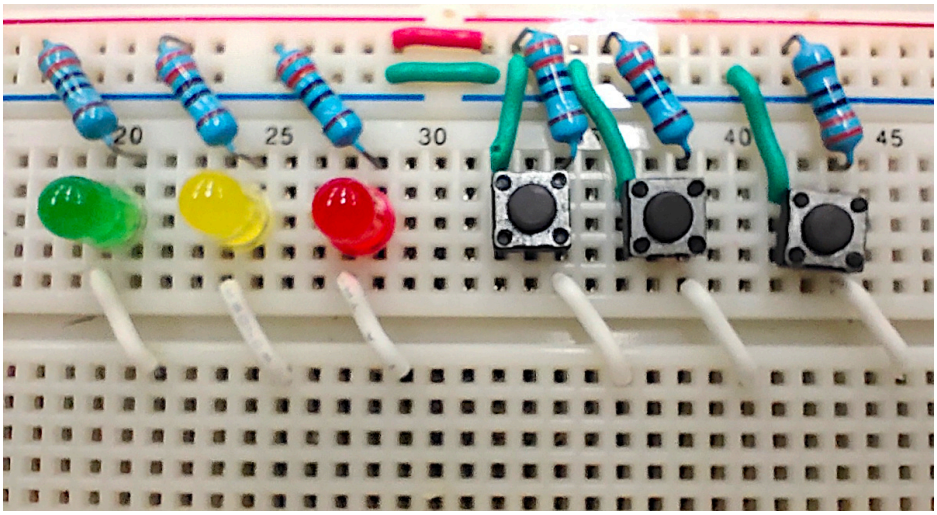
```
int timer=0;
uint8_t state=HIGH;
void setup() {
  pinMode(2,OUTPUT);
// ตั้งค่าตัวนับและตัวตั้งเวลา
  TCCR0A=(1<<WGM01);
  OCR0A=0xF9; //Value for ORC0A for 1ms
// ร้องของการทำงาน และเปิดให้เกิดการขัดจังหวะ (enable interrupt)
  TIMSK0|=(1<<OCIE0A);
  sei(); //Enable interrupt
// ตั้งค่าพรีสเกล
  TCCR0B|=(1<<CS01);
  TCCR0B|=(1<<CS00);
}
void loop() {
  if(timer>=1000){
    state=HIGH-state;
    timer=0;
  }
  digitalWrite(2,state);
}
ISR(TIMER0_COMPA_vect){
  timer++;
}
```

ตัวอย่างจำลองไฟจราจร

ตัวอย่าง.โปรแกรมสำหรับจำลองไฟจราจรโดยกำหนดให้สามารถสั่งงานด้วยสวิตช์เพื่อกำหนดโหมดทำงานเป็นช่วงว่าง (CMD_IDLE) และการทำงานปกติ (CMD_START) เพื่อให้กระพริบเฉพาะหลอดสีเหลืองกับการทำงานปกติที่กำหนดให้มีการทำงานดังนี้

รายการอุปกรณ์สำหรับทดลองตามภาพที่ 19 ประกอบด้วย

- หลอดแอลอีดีสีแดง ต่อเข้ากับขา 2
- หลอดแอลอีดีสีเหลือง ต่อเข้ากับขา 3
- หลอดแอลอีดีสีเขียว ต่อเข้ากับขา 5
- สวิตช์กดติดปล่อยดับสำหรับปุ่ม CMD_IDLE ต่อเข้ากับขา 6
- สวิตช์กดติดปล่อยดับสำหรับปุ่ม CMD_START ต่อเข้ากับขา 7



ภาพที่ 19 ตัวอย่างวงจรสำหรับทดลองเรื่องไฟจราจร

โค้ดของโปรแกรมเป็นดังนี้

```
#define LED_RED 2
#define LED_YELLOW  3
#define LED_GREEN 5
#define CMD_IDLE 6
#define CMD_START 7

uint8_t status;
uint8_t led_status;
uint8_t light_status;

void setup() {
  status = 1;
  led_status = LOW;
  light_status = 0;
  pinMode( LED_RED, OUTPUT );
  pinMode( LED_YELLOW, OUTPUT );
  pinMode( LED_GREEN, OUTPUT );
  pinMode( CMD_IDLE, INPUT_PULLUP );
  pinMode( CMD_START, INPUT_PULLUP );
}

void loop() {
  if (digitalRead( CMD_IDLE )==0) {
    status = 0;
  }
  else if (digitalRead( CMD_START )==0) {
    status = 1;
    light_status = 0;
  }
  if (status) {
    if (light_status < 20)  {
      digitalWrite(LED_RED, HIGH);
      digitalWrite(LED_GREEN, LOW);
      digitalWrite(LED_YELLOW, HIGH);
      light_status++;
    } else if (light_status < 30) {
      digitalWrite(LED_RED, HIGH);
      digitalWrite(LED_GREEN, HIGH);
      digitalWrite(LED_YELLOW, LOW);
      light_status++;
    } else if (light_status < 50) {
      digitalWrite(LED_RED, LOW);
      digitalWrite(LED_GREEN, HIGH);
      digitalWrite(LED_YELLOW, HIGH);
      light_status++;
    } else if (light_status == 50) {
      light_status=0;
    }
  } else {
    digitalWrite( LED_RED, HIGH); // off
    digitalWrite( LED_GREEN, HIGH); // off
    digitalWrite( LED_YELLOW, led_status);
    led_status = HIGH-led_status;
  }
```

ตัวอย่างการหรี่หลอดแอลอีดีด้วย PWM

ตัวอย่างโปรแกรมการหรี่หลอดแอลอีดีที่เชื่อมต่อกับขา 5 (PIN_LED) ของบอร์ด NANO F328-C ด้วยการใช้ PWM (Pulse-Width-Modulate) ด้วยการส่งค่าดิวตี้ (duty) ที่มีค่าในช่วง 0 ถึง 255 แทนค่าระดับแรงดันที่ส่งไปยังขา 5 ด้วยคำสั่ง

analogWrite(หมายเลขขา, ค่าดิวตี้)

โค้ดโปรแกรมเป็นดังนี้

```
#define PIN_LED 5
int pwmValue = 0;\
void setup() {
  pinMode( PIN_LED, OUTPUT );
}

void loop() {
    analogWrite( PIN_LED, pwmValue);
    pwmValue++;
    if (pwmValue > 255) {
       pwmValue = 0;
    }
    delay(10);
}
```

ตัวอย่างการทำลอจิกโพรบ

ตัวอย่างการทำลอจิกโพรบเพื่อบอกค่าความเป็นสัญญาณดิจิทัลด้วยการอ่านค่าจากขาที่เชื่อมต่อกับโมดูล ADC (Analog to Digital Converter) ของไมโครคอนโทรลเลอร์ LGT8G328P ที่มีความละเอียดในระดับ 12 บิต หรือสามารถแปลงค่าแรงดันจาก 0 ถึง 5VDC เป็นตัวเลขในช่วง 0 ถึง 4095 ด้วยคำสั่งสำหรับอ่านค่า ADC ของชิพ LGT8F328P ใช้คำสั่ง analogRead( ) พร้อมระบุหมายเลขขาสำหรับอ่านค่า โดยขาที่อ่านได้คือ A0, A1, A2, A3, 4, A5, A6 และ A7

ค่าที่อ่านได้ = analogRead(หมายเลขขา)

การปรับความละเอียดของการแปลง ADC มีรูปแบบการสั่งงานดังนี้

analogReadResolution(ค่าความละเอียด)

หลังจากนั้นแปลงค่าดิจิทัลกลับไปเป็นค่แรงดันด้วยสมการต่อไปนี้

ค่าแรงดัน = ค่าแรงดันสูงสุด * (ค่าดิจิทัล / 4095)

สุดท้ายทำการแบ่งช่วงค่าแรงดันที่แปลงได้เป็น 3 ช่วงตามค่าแรงดันแบบ TTL ที่มี 3 ช่วงค่า คือ

- 1 ตั้งแต่ 2.0VDC และให้เปิดหลอดแอลอีดีสีเขียว
- 0 น้อยกว่า 0.9VDC และให้เปิดหลอดแอลอีดีสีแดง
- ระบุไม่ได้ อยู่ในช่วง 0.9VDC ถึง 1.9VDC และให้เปิดหลอดแอลอีดีสีเหลือง

โค้ดโปรแกรมเป็นดังนี้

```
#define pinADC A0
#define LED_RED 2
#define LED_YELLOW  3
#define LED_GREEN 5
#define ADC_BITS 12

void setup() {
  Serial.begin(9600);
  pinMode( LED_RED, OUTPUT );
  pinMode( LED_YELLOW, OUTPUT );
  pinMode( LED_GREEN, OUTPUT );
  pinMode( pinADC, INPUT );
  analogReadResolution(ADC_BITS);
  analogReference(DEFAULT);  // 5v
}

void loop() {
  int aValue = analogRead( pinADC );
  float approxVDC = 5.0*(aValue/4095.0);
  Serial.print(approxVDC);
  Serial.print(",");
  Serial.println(aValue);
  if (approxVDC < 0.9) {
    digitalWrite(LED_RED, LOW);
    digitalWrite(LED_YELLOW, HIGH);
    digitalWrite(LED_GREEN, HIGH);
  } else if (approxVDC < 2.0) {
    digitalWrite(LED_RED, HIGH);
    digitalWrite(LED_YELLOW, LOW);
    digitalWrite(LED_GREEN, HIGH);
  } else {
    digitalWrite(LED_RED, HIGH);
    digitalWrite(LED_YELLOW, HIGH);
    digitalWrite(LED_GREEN, LOW);
  }
}
```

ตัวอย่างการสร้างกราฟจาก DAC

การใช้งาน DAC จะต้องใช้ขา D4 หรือขา 4 บนบอร์ด NANO F328-C โดยกำหนดการทำงานของขาเป็น ANALOG ด้วยคำสั่งต่อไปนี้

```
pinMode( 4, ANALOG );
```

นอกจากนี้ต้องกำหนดแหล่งของแรงดันอ้างอิงสำหรับการทำงานของภาคแอนาล็อกสามารถระบุแหล่งอ้างอิงได้จากคำสั่ง analogReference() โดยกำหนดแหล่งอ้างอิง x ดังรูปแบบต่อไปนี้

```
analogReference( x )
```

จากคำสั่ง ค่า x เป็นดังนี้

- DEFAULT ใช้แรงดันอ้างอิง 5V
- EXTERNAL ใช้ขา REF เป็นขานำเข้าแหล่งแรงดันอ้างอิง
- INTERNAL4V096 ใช้แรงดันอ้างอิง 4.096v จากภายในบอร์ด
- INTERNAL2V048 ใช้แรงดันอ้างอิง 2.048v จากภายในบอร์ด
- INTERNAL1V048 ใช้แรงดันอ้างอิง 1.048v จากภายในบอร์ด

ตัวอย่างโปรแกรมสร้างกราฟรูปโคไซน์ (cosine)

```
#define PI 3.14159265358979323846264338327 95

float rad, CosValue;
byte dacValue;

void setup() {
  pinMode(4, ANALOG);  // เปิดให่้ขา 4 ทำงานเป็น DAC
  analogReference(INTERNAL4V096); //ใช้ค่าแรงดันอ้างอิงเป็น 4.096V
}

void loop()
{
  for(float i=0.0;i<=2.0;i=i+0.01) {
      rad=PI*i;
      CosValue=cos(rad);
      dacValue=(byte)(127+(CosValue*127)); // สเกลค่าเป็น 127+(-127..128)
      analogWrite(4, dacValue);   //DAC output
  }
}
```

ตัวอย่างโปรแกรมสร้างกราฟรูปสามเหลี่ยมโดยใช้วิธีการสร้างระดับแรงดันที่ขา DAC0 หรือขา 4 แล้วต่อขาเข้ากับขา A1 เพื่ออ่านค่าที่สร้างด้วยโมดูล DAC มาแสดงพอร์ตอนุกรมทำให้สามารถใช้โปรแกรมแสดงกราฟของ Arduino IDE ทำหน้าที่พล็อตกราฟจากค่าที่อ่านได้โดยไม่ต้องใช้ออสซิโลสโคปดูค่าจากขา DAC

```
#define MAX_VALUE 255

int adcValue;
int i;

void setup() {
  Serial.begin(9600);
  analogReference(DEFAULT);  // 5v
  pinMode(DAC0, ANALOG);
}

void loop() {
  for (i = 0; i <= MAX_VALUE; i ++) {
    analogWrite( DAC0, i);
    adcValue = analogRead(A1);
    Serial.println(adcValue);
  }
  for (i = MAX_VALUE-1; i > 0; i--) {
    analogWrite( DAC0, i);
    adcValue = analogRead(A1);
    Serial.println(adcValue);
  }
}
```