

วิธีการแจ้งเตือนผ่าน Line

เป็นการประยุกต์ใช้ **LINE Notify** สำหรับ IoT โดยบอร์ด **ESP32** มีความสามารถเชื่อมต่อสายฟ้าและส่งข้อความแจ้งเตือนไปยัง **LINE Notify** ได้ สามารถนำไปประยุกต์ใช้งานได้หลากหลาย

<https://notify-bot.line.me/th/>

Token = YZofotJjQu2cEFjKiU68zkhGaJMcFpAXxJBINcLpgrU

Website Encode ภาษาไทย

<http://meyerweb.com/eric/tools/dencoder/>

ติดตั้ง Line Notify

Add Line Notify เป็นเพื่อนก่อน



ติดตั้ง Line Notify

เข้าไปที่หน้าเว็บ <https://notify-bot.line.me/my/>
เพื่อรับ Token

กดออก Token คือการสร้าง Token



The image shows the LINE registration page. At the top, the word "LINE" is written in large green letters. Below it, there are two input fields: "ชื่อผู้ใช้งาน" (Username) and "รหัสผ่าน" (Password). A purple button labeled "ลื้อคอิน" (Log In) is positioned below the password field. At the bottom of the page, there is a small footer that reads "เกี่ยวกับ LINE | © LINE Corporation".



The image shows the "ออก Access Token (สำหรับผู้พัฒนา)" (Generate Access Token (for developers)) page. Below the title, there is a line of text: "เมื่อใช้ Access Token แบบบุคคล จะสามารถตั้งค่าการแจ้งเตือนได้โดยไม่ต้องลงทะเบียน" (When using a personal Access Token, you can set up notifications without registration). There are two buttons: a purple button labeled "ออก Token" (Generate Token) and a white button with a purple border labeled "ดาวน์โหลดเอกสารการพัฒนา" (Download development document).

ติดตั้ง Line Notify

ใส่ชื่ออุปกรณ์ที่จะส่งข้อความ เลือกกลุ่มที่จะแจ้งเตือน

ออก Token

โปรดใส่ชื่อ Token (จะแสดงเมื่อมีการแจ้งเตือน)

โปรดเลือกห้องแชทที่ต้องการส่งข้อความแจ้งเตือน

🔍 Search by group name

- 👤 รับการแจ้งเตือนแบบตัวต่อตัวจาก LINE Notify
- 🟢 [Blurred]
- 🟠 [Blurred]
- 🌐 [Blurred]

Copy Token เก็บไว้ใน Text เพราะออกได้ครั้ง

Token ที่ออก

laQPmDB3eFUJAQtLPAr97dpkLafBr5g4H0ItWdi

ถ้าออกจากหน้านี้ ระบบจะไม่แสดง Token ที่ออกใหม่อีกต่อไป โปรดคัดลอก Token ก่อนออกจากหน้านี้

คัดลอก **ปิด**

Workshop คำสั่ง Line Notify

```
#include <WiFi.h>
#include <WiFiClientSecure.h>
//##### CONFIGURATION : MODIFY HERE #####
char ssid[] = "---"; //ชื่อสายพาย SSID
char pass[] = "---"; //รหัสผ่าน WiFi
String txt = ""; //ข้อความที่จะแสดงใน Line
//รหัสเชื่อมต่อ Line TOKEN
#define TokenLine "YZofotJjQu2cEFjKiU68zkhGaJMcFpAXxJBINcLpgrU"

boolean sent = false;
int state = 0;
void setup() {
  Serial.begin(115200);
  //เริ่มการเชื่อมต่อกับสายพายแม่ข่าย
  WiFi.begin(ssid, pass);
  //แสดง "WiFi Connecting" ในคอนโซล
  Serial.print("WiFi Connecting");
  //ตรวจสอบสถานะการเชื่อมต่อสายพาย
  //ถ้าไม่สำเร็จให้แสดง "." ในคอนโซลจนกว่าจะเชื่อมต่อได้
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }
  //แสดงสถานะการรายพายเชื่อมต่อแล้ว และแสดงหมายเลขไอพีที่ได้รับในคอนโซล
  Serial.println();
  Serial.print("connected: ");
  Serial.println(WiFi.localIP());
  pinMode(22, OUTPUT); // กำหนดขาในการควบคุม
}
```

Workshop คำสั่ง Line Notify

```
void loop() {
  int LDR = analogRead(A0); // pin ที่ต่อ LDR
  int Light = LDR;
  Serial.println("Light=" + String(Light));
  Light = map(Light, 1, 4095, 0, 100); //ทำการ Map %
  Serial.println("MapLight=" + String(Light) + " %"); // แสดงผลเป็น %
  delay(1000);
  // เงื่อนไขของการส่ง และสถานะแสง
  if (Light < 30 && state != 1 ) { //ถ้าแสงน้อยกว่า 30% ให้ไฟติดและส่งค่า
    txt = "Dark";
    digitalWrite (22,HIGH); // ให้ขาที่ 22 ไฟติด
    Serial.println(txt);
    state = 1;
    sent = 0;
  }
  else if (Light > 30 && state != 2 ) { //ถ้าแสงมากกว่า 30% ให้ไฟดับและส่งค่า
    txt = "Day";
    digitalWrite (22,LOW); // ให้ขาที่ 22 ไฟดับ
    Serial.println(txt);
    state = 2;
    sent = 0;
  }
  // เงื่อนไขของการส่งสถานะ
  if (sent == 0 ) { // sent 0 คือยังไม่ได้ส่ง
    Serial.println(txt);
    NotifyLine(txt);
    sent = 1; // ให้บอกว่าส่งแล้ว
  }
}
```

Workshop คำสั่ง Line Notify

```
void NotifyLine(String t) {
    WiFiClientSecure client;
    if (!client.connect("notify-api.line.me", 443)) {
        Serial.println("Connection failed");
        return;
    }
    String req = "";
    req += "POST /api/notify HTTP/1.1\r\n";
    req += "Host: notify-api.line.me\r\n";
    req += "Authorization: Bearer " + String(TokenLine) + "\r\n";
    req += "Cache-Control: no-cache\r\n";
    req += "User-Agent: ESP8266\r\n";
    req += "Content-Type: application/x-www-form-urlencoded\r\n";
    req += "Content-Length: " + String(String("message=" + t).length()) + "\r\n";
    req += "\r\n";
    req += "message=" + t;
    Serial.println(req);
    client.print(req);
    delay(20);
    Serial.println("-----");
    while (client.connected()) {
        String line = client.readStringUntil('\n');
        if (line == "\r") {
            break;
        }
    } //while
}
```