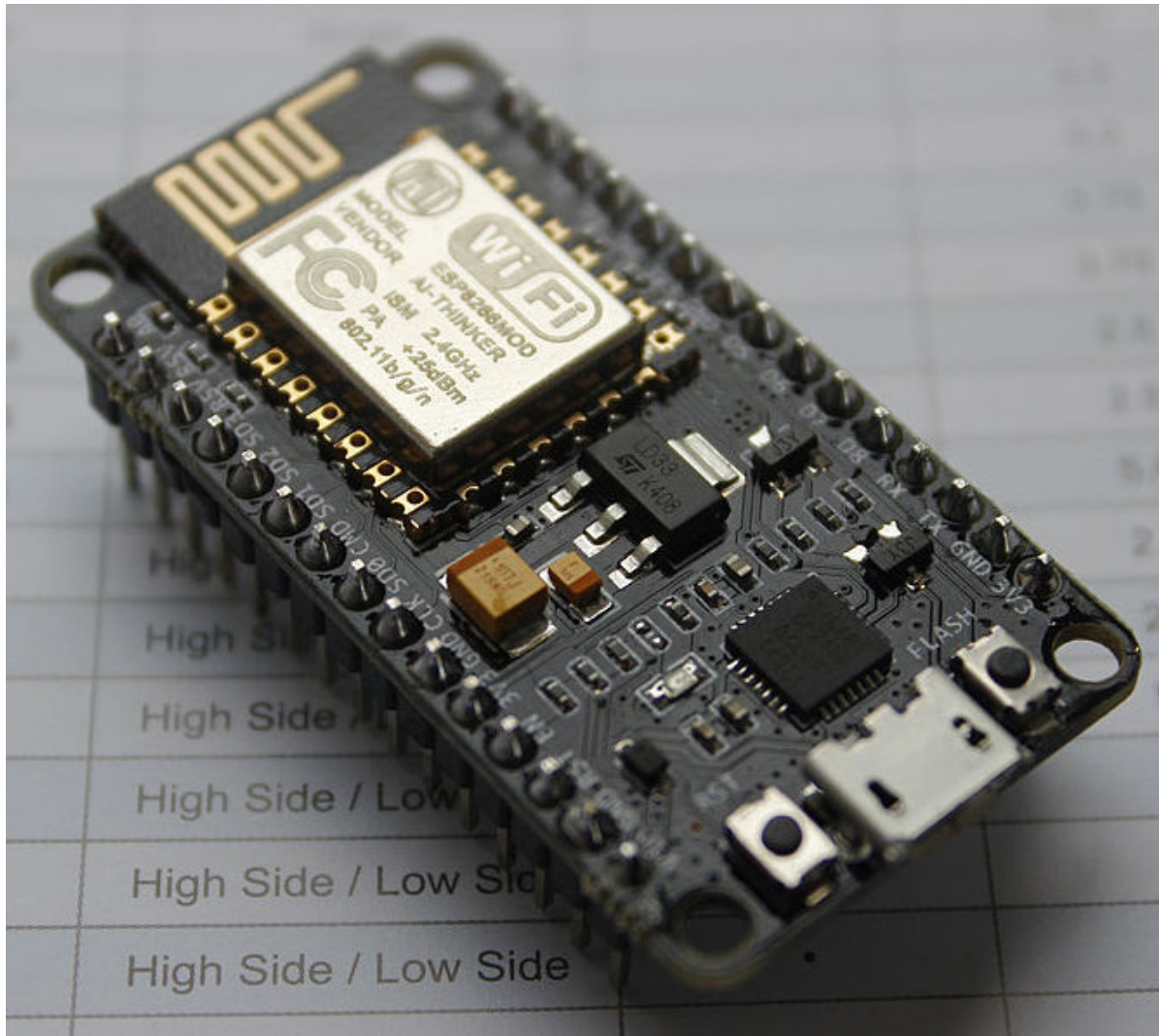


สอนการใช้งาน Esp8266 Nodemcu

[สอนการใช้งาน Esp8266 Nodemcu](#)





บทที่ 1 การติดตั้งโปรแกรม Arduino 1.6.5

ทำการดาวน์โหลดติดตั้ง โปรแกรม IDE ของ Arduino โดยเข้าไปที่ <http://arduino.cc> เข้าไปที่ Download คลิกที่ช่อง Previous Releases เข้าไปที่ Previous version of current release เลือกดาวน์โหลด เวอร์ชัน 1.6.5 สำหรับ Windows ให้ใช้ Windows Installer แล้วทำการติดตั้งโปรแกรม arduino เป็นโปรแกรม portable คือสามารถทำการย้ายโปรแกรมทั้งหมดไปอยู่ที่โฟลเดอร์ใดก็ได้โดยที่โปรแกรมจะยังคงทำงานได้เหมือนเดิม

เมื่อเริ่มเข้าโปรแกรมให้เข้าไปที่ File Preferences และเปลี่ยน Additional Boards Manager เป็น URLs: http://arduino.esp8266.com/package_esp8266com_index.json

เข้าไปที่ Tools เข้า Boards Manager ค้นหาคำว่า Esp8266 คลิกที่บาร์ esp8266 by ESP8266 Community จะปรากฏแถบเวอร์ชันให้เลือก เลือกเวอร์ชันล่าสุด กด Install ขั้นตอนอาจใช้เวลานาน ขึ้นอยู่กับความเร็วอินเทอร์เน็ต

เข้าไปที่ Sketch เข้าไปที่ Include Library เข้าไปที่ Manage Libraries ค้นหาคำว่า ESP8266 เลือก install ที่จะใช้ ให้เลือก Adafruit ESP8266 เพื่อติดตั้ง Code ตัวอย่าง

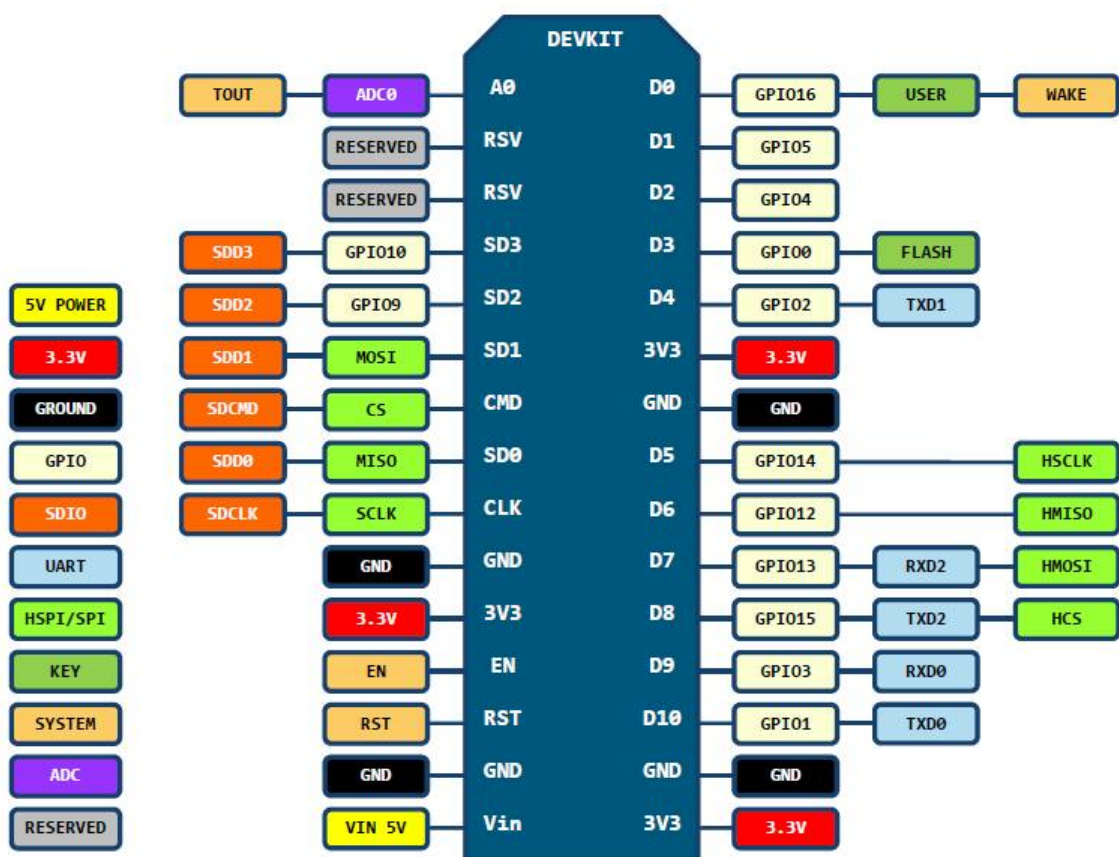
ทำการทดสอบบอร์ดโดยต่อบอร์ดเข้ากับ สาย microusb สำหรับ ESP บางรุ่นอาจจำเป็นต้องดาวน์โหลด driver เฉพาะ ขึ้นกับชิป usb ที่นำมาใช้

เข้าไปที่ Tools เลือก Boards "NodeMCU 1.0 (ESP-12E Module)" เลือก Programmer เป็น Parallel Programmer

เข้าไปที่ File เลือก Examples เลือก ESP8266 เลือก Blink จะขึ้นหน้าต่างใหม่พร้อม code ตัวอย่าง Blink ในที่นี้ คือ Blink LED ที่อยู่บนบอร์ดของ NodeMCU อยู่แล้ว ไม่ต้องต่อ LED ก่อน กดถูก เพื่อ Compile กดลูกศรขวาเพื่อ Upload ได้เลย

LED บนบอร์ดนี้จะตรงกับ GPIO16 หรือขา D0

PIN DEFINITION



D0(GPIO16) can only be used as gpio read/write, no interrupt supported, no pwm/i2c/ow supported.

บทที่ 2 การต่อไฟกระพริบ ให้ต่อหลอด LED เข้ากับ breadboard ตามรูป โดยให้ขายาวของหลอด LED หันไปทางด้านที่ออกจากขา D เช่น D0,D1,D2 ต่อตัวต้านทาน 220 โอห์ม ด้านใดก็ได้ ไม่มีขั้ว

โปรแกรม(สามารถดึงจาก Basics Example ของ Arduino ได้)

// the setup function runs once when you press reset or power the board

```

void setup() {
  // initialize digital pin D0 as an output.
  pinMode(D0, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(D0, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(D0, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}

```

ทั้งนี้ต้องเข้าไปที่ Tools เลือก Board NodeMCU 1.0

เลือก Programmer Parallel Programmer

สามารถดึง Example Program จาก File Examples

เลือก 01. Basics Blink แก่ Pin 13 เป็น D0

=====

กดเครื่องหมาย ถูก เพื่อ Verify

กดเครื่องหมาย ลูกศรขวา เพื่อ Upload

การบ้าน

=====

ให้นักเรียนทดลองเขียนโปรแกรมขับ LED D1 และ D2 ที่เหลือ โดยวิเคราะห์จากตัวอย่างโค้ดข้างต้น อาจให้สลับกันติดหรือติดพร้อมกันได้ตามต้องการ

บทที่ 3 การติดตั้ง NodeMCU เป็น Server เบื้องต้น

ในโปรแกรม Arduino 1.6.5 เลือก Tools เลือก Board เป็น NodeMCU 1.0 (ESP 12E Module) กลับไปที่ Files เลือก Examples เลือก ESP8266 WebServer เลือก HelloServer จะได้ code ตัวอย่างของการติดตั้ง Server เบื้องต้นบน NodeMCU

ปรับเปลี่ยนบรรทัด

```
const char* ssid = ".....";
```

```
const char* password = ".....";
```

ให้ตรงกับค่าของ wireless router หรือ AP ภายในบ้าน

ทำการคอมไพล์ และ อัปโหลด

หลังจากนั้น ให้ไปที่ Tools เลือก Serial Monitor ปรับความเร็วไปที่ 115200 ในหน้าต่าง

Serial Monitor จะแสดงค่า ip ของ NodeMCU ที่ได้จาก DNS ของ AP เข้าไปที่โทรศัพท์เรียก ip ที่ได้จาก Serial Monitor จะได้ข้อความ

hello from esp8266!

บนเว็บเบราว์เซอร์ของโทรศัพท์ การบ้านให้ปรับเปลี่ยนข้อความ โดยใช้ html ปรับแต่งให้มีความสวยงามตามต้องการ

บทที่ 4 การติดตั้งจอภาพ LCD ให้ไปดาวน์โหลดไลบรารีที่

<https://github.com/fdebra.../Arduino-LiquidCrystal-I2C-library> แล้วทำการวางไลบรารีในโฟลเดอร์ไลบรารีของ arduino ide จากนั้นเข้าไปในโปรแกรม Arduino เลือก Files ,Examples ,Arduino-LiquidCrystal-I2C-library-master เลือก Hello World บนบอร์ดให้อาสาสายที่แถมให้ ต่อ SCL เข้ากับ D1 ต่อ SDA เข้ากับ D2 ต่อ VCC เข้ากับ + 5 V DC บน breadboard ต่อ GND เข้ากับ - ในโปรแกรม HelloWorld ให้เปลี่ยนเป็น LiquidCrystal_I2C lcd(0x3F, 16, 2) ทำการคอมไพล์แล้วอัปโหลด ให้ปรับ contrast ที่หลังจอไปขวาสุด

บทที่ 5 การต่อเชื่อมขา input และ output หนึ่งในขาที่พิเศษที่สุดของ NodeMCU V2 คือขา A0 มีหน้าที่ไว้รับอินพุตที่เป็นอนาล็อก (มีเพียงขาเดียว) โดยปกติ ESP12E ขา ADC จะรับอินพุตที่เป็นอนาล็อกได้เพียง 0 ถึง 1V แต่บน V2 ได้ปรับการรับอินพุตได้ถึง 5V ปกติมักใช้กับ Soil Sensor ที่เป็น Analog สำหรับขาอินพุต เอาท์พุต ที่เป็นดิจิตอล ที่ใช้ได้คือขา D0 ถึง D8 ทั้งหมด 9 ขา สามารถทำเอาท์พุตเป็น Analog ได้ คือส่งหรือไฟได้ D1 D2 ใช้สำหรับ I2C D5 ถึง D8 ใช้กับ SPI เช่นการติดต่อกับ SD card สำหรับ D9 กับ D10 สงวนไว้ใช้ กับ Serial ส่วน GPIO 5 6 7 8 9 10 และ 11 นั้นที่ใช้ไม่ได้เพราะตัว ESP แย่งเอาไปใช้ภายในแล้ว คือ อาจต้องนำไปใช้เพื่อจะได้สื่อสารทาง Wi-Fi ได้

การบ้าน ให้ทดลองต่อหลอด LED ทุกขาตั้งแต่ D0 ถึง D8

การทำงานของ ไอสแควร์ซี โปรโตคอล <https://www.youtube.com/watch?v=6IAkYpmA1DQ>

เชิงอรรถ ของบทที่ 5

การ scan address ของอุปกรณ์ที่อยู่ใน ไอสแควร์ซีบัส บางทีถ้าเราซื้ออุปกรณ์ ไอสแควร์ซี เช่น จอภาพ LCD แบบ I2C มาแล้วแต่ไม่รู้ว่า address ของมันคืออะไร ให้ใช้โปรแกรม i2c_scanner ทำการสแกน address ของอุปกรณ์นั้น โดยเปิดดูที่ Serial monitor ที่ความเร็ว 9600

```
// -----  
// i2c_scanner  
//  
// Version 1  
// This program (or code that looks like it)  
// can be found in many places.  
// For example on the Arduino.cc forum.
```

```

// The original author is not know.
// Version 2, Juni 2012, Using Arduino 1.0.1
// Adapted to be as simple as possible by Arduino.cc user Krodal
// Version 3, Feb 26 2013
// V3 by louarnold
// Version 4, March 3, 2013, Using Arduino 1.0.3
// by Arduino.cc user Krodal.
// Changes by louarnold removed.
// Scanning addresses changed from 0...127 to 1...119,
// according to the i2c scanner by Nick Gammon
// http://www.gammon.com.au/forum/?id=10896
// Version 5, March 28, 2013
// As version 4, but address scans now to 127.
// A sensor seems to use address 120.
// Version 6, November 27, 2015.
// Added waiting for the Leonardo serial communication.
//
//
// This sketch tests the standard 7-bit addresses
// Devices with higher bit address might not be seen properly.
//
#include <Wire.h>
void setup()
{
  Wire.begin();
  Serial.begin(9600);
  while (!Serial); // Leonardo: wait for serial monitor
  Serial.println("\nI2C Scanner");
}
void loop()
{
  byte error, address;
  int nDevices;
  Serial.println("Scanning...");
  nDevices = 0;
  for(address = 1; address < 127; address++ )
  {
    // The i2c_scanner uses the return value of

```

```

// the Wire.endTransmission to see if
// a device did acknowledge to the address.
Wire.beginTransmission(address);
error = Wire.endTransmission();
if (error == 0)
{
  Serial.print("I2C device found at address 0x");
  if (address < 16)
  Serial.print("0");
  Serial.print(address, HEX);
  Serial.println(" !");
  nDevices++;
}
else if (error == 4)
{
  Serial.print("Unknow error at address 0x");
  if (address < 16)
  Serial.print("0");
  Serial.println(address, HEX);
}
}
if (nDevices == 0)
  Serial.println("No I2C devices found\n");
else
  Serial.println("done\n");

delay(5000); // wait 5 seconds for next scan
}

```

บทที่ 6 การใช้งาน RFID SPI module ร่วมกับ LCD I2C module
การต่อสาย RFID module

```

// IRQ = none
// NSS = GPIO10 SD3
// RST = GPIO16 D0
// MOSI = GPIO13 D7
// MISO = GPIO12 D6
// SCK = GPIO14 D5
// GND = GND

```

```
// 3.3V = 3.3V
```

สังเกตว่าเลือกใช้ SS ที่ GPIO10 และ RST ที่ D0

สำหรับการต่อ LCD I2C ต่อ SCL เข้ากับ D1 และ SDA เข้า D2

ที่มา <http://www.instructables.com/id/WiFi-RFID-Reader/>

แต่ได้ปรับปรุงจากของ instructables เพราะพบว่าถ้าทำตาม

วิธีของ instructables จะไม่สามารถ upload ได้ครับ

ไบนารีของ RFID

<https://github.com/miguelbalboa/rfid>

โค้ดของโปรแกรม

```
#include <Wire.h>
```

```
#include <LiquidCrystal_I2C.h>
```

```
#include "MFRC522.h"
```

```
#define RST_PIN 16 // RST-PIN for RC522 - RFID - SPI - Modul GPIO16
```

```
#define SS_PIN 10 // SDA-PIN for RC522 - RFID - SPI - Modul GPIO10
```

```
// Set the LCD address to 0x27 or 0x3F for a 16 chars and 2 line display
```

```
LiquidCrystal_I2C lcd(0x3F, 16, 2);
```

```
MFRC522 mfrc522(SS_PIN, RST_PIN); // Create MFRC522 instance
```

```
void setup() {
```

```
  // initialize the LCD
```

```
  lcd.begin();
```

```
  // Turn on the backlight and print a message.
```

```
  lcd.backlight();
```

```
  lcd.print("RFID Reader");
```

```
  Serial.begin(9600); // Initialize serial communications
```

```
  SPI.begin(); // Init SPI bus
```

```
  mfrc522.PCD_Init(); // Init MFRC522
```

```
}
```

```
void loop() {
```

```
  // Look for new cards
```

```
  if ( ! mfrc522.PICC_IsNewCardPresent() ) {
```

```
    delay(50);
```

```
    return;
```

```
  }
```

```
  // Select one of the cards
```

```
  if ( ! mfrc522.PICC_ReadCardSerial() ) {
```

```
    delay(50);
```

```
    return;
```

```
  }
```



```

lcd.clear();
// Show some details of the PICC (that is: the tag/card)
lcd.print("Card UID:");
lcd.setCursor(0, 1);
Serial.print(F("Card UID:"));
dump_byte_array(mfrc522.uid.uidByte, mfrc522.uid.size);
Serial.println();
}

// Helper routine to dump a byte array as hex values to Serial
void dump_byte_array(byte *buffer, byte bufferSize) {
  for (byte i = 0; i < bufferSize; i++) {
    lcd.print(buffer[i] < 0x10 ? " 0" : " ");
    lcd.print(buffer[i], HEX);
    Serial.print(buffer[i] < 0x10 ? " 0" : " ");
    Serial.print(buffer[i], HEX);
  }
}

```

บทที่ 7 การใช้งานดินฟ้าอากาศ หลังจากผ่านมาแล้ว 6 บท เข้าใจว่านักเรียนสามารถแกะการทำงานจากโค้ดของโปรแกรมได้แล้ว บทนี้ใช้ จอภาพแบบ OLED เพื่อแสดงอุณหภูมิ ความชื้นในอากาศ และ ความชื้นในดิน ที่ได้จากการอ่านค่า Analog A0 โดยตรงมีหน่วยเป็น โวลต์ 0 ถึง 5 V DC การต่อขาต่างๆ

OLED SCL --> D5

OLED SDA --> D3

DHT สายเหลือง --> D1

Moisture OUT --> A0

สำหรับ DHT ในโครงงานนี้ใช้ AM2301

แต่สามารถใช้ ไลบรารีร่วมกับ DHT ทั่วไปได้

สามารถติดตั้งได้จาก Library Manager

Adafruit DHT และ ใช้ของ ESP8266 OLED Driver

for SSD1306 by Daniel Eichhorn

โค้ดทั้งหมด

[#include](#) <SPI.h>

#include <Wire.h>

#include "SSD1306.h"

#include <stdint.h>

#include "DHT.h"

```

#define DHTPIN D1 // modify to the pin we connected
#define DHTTYPE DHT21 // AM2301
SSD1306 display(0x3c, D3, D5);
DHT dht(DHTPIN, DHTTYPE);
void setup()
{
  pinMode(A0, INPUT);
  dht.begin();
  display.init();
  display.flipScreenVertically();
  display.setFont(ArialMT_Plain_10);
}

void loop()
{
  // Clear the buffer.
  display.clear();

  // text display tests
  display.setColor(WHITE);
  float hum = dht.readHumidity();
  float temp = dht.readTemperature();
  // check if returns are valid, if they are NaN (not a number) then something went
  wrong!
  if (isnan(temp) || isnan(hum))
  {
    display.drawString(0, 0, "Failed to read from DHT");
  }
  else
  {
    display.drawString(0, 0, "Temperature: "+String(temp)+" c");
    display.drawString(0, 20, "Humidity: "+String(hum));
  }
  // read the input on analog pin 0:
  int sensorValue = analogRead(A0);
  // Convert the analog reading (which goes from 0 - 1023) to a voltage (0 - 5V):
  float voltage = sensorValue * (5.0 / 1023.0);
  // print out the value you read:

```

```

display.drawString(0, 40, "Soil Moisture: "+String(voltage)+" V");
display.display();
delay(2000);
}

```

บอร์ด 8 NodeMCU Motor Driver shield เป็น shield ชนิดเดียวที่มีขายโดยทั่วไปสำหรับ V2 หรือ 1.0 เดิมทำสำเร็จรูปมาให้ใช้เป็นรถบังคับ แต่ก็สามารถนำมาคุมมอเตอร์ hi-torque ได้ดี ควบคุมมอเตอร์ได้ 2 ตัว สั่งงานหมุนไปกลับและควบคุม speed ได้ด้วยชุดคำสั่ง การควบคุมนี้ใช้ชิป L293D ที่อยู่บนตัว shield วิธีใช้ให้ต่อมอเตอร์เข้าขา A+ A- และ B+ B- ต่อไฟเลี้ยงมอเตอร์ อาจเป็น 12 V DC หรือ 6 V DC เข้าที่ขา VM และ GND ต่อไฟเลี้ยง shield 5 V DC เข้าที่ VCC และ GND แยกจากกัน ใน code ตัวอย่างนี้จะมีการใช้ LCD 2 บรรทัดเข้ามาด้วย โค้ดตัวอย่าง

```

=====
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,16,2);
#define DIRA 0
#define PWMA 5
void setup() {
  // put your setup code here, to run once:
  Serial.begin(115200);
  Serial.println();
  Serial.println("Starting...");
  Serial.println("Initialising LCD...");
  lcd.init(); // initialize the lcd
  // Print a message to the LCD.
  lcd.backlight();
  lcd.setCursor(0,0);
  lcd.print("Motor test");
  lcd.setCursor(0,1);
  lcd.print("Speed: 0");
  // Serial.println("Flashing internal LED");
  // pinMode(BUILTIN_LED, OUTPUT);
  // digitalWrite(BUILTIN_LED, LOW);
  // delay(100);
  // digitalWrite(BUILTIN_LED, HIGH);
  // delay(300);
}

```

```
Serial.println("Preparing motor...");
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Preparing motor");
pinMode(DIRA, OUTPUT);
pinMode(PWMA, OUTPUT);
analogWrite(PWMA,0);
digitalWrite(DIRA,1);
delay(5000);
lcd.clear();
Serial.println("Starting motor...");
lcd.setCursor(0,0);
lcd.print("Operating motor");
lcd.setCursor(0,1);
lcd.print("Speed: 5");
analogWrite(PWMA,5);
delay(5000);
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Operating motor");
lcd.setCursor(0,1);
lcd.print("Speed: 50");
analogWrite(PWMA,50);
delay(5000);
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Operating motor");
lcd.setCursor(0,1);
lcd.print("Speed: 100");
analogWrite(PWMA,100);
delay(5000);
lcd.setCursor(0,1);
lcd.print("Speed: 300");
analogWrite(PWMA,300);
delay(5000);
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Operating motor");
```

```

lcd.setCursor(0,1);
lcd.print("Speed: 1000");
analogWrite(PWMA,1000);
delay(3000);
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Slowing down");
analogWrite(PWMA,0);
delay(3000);
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Reversing motor");
digitalWrite(DIRA,0);
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Operating motor (reverse)");
lcd.setCursor(0,1);
lcd.print("Speed: 50");
analogWrite(PWMA,250);
delay(3000);
}
void loop() {
  // put your main code here, to run repeatedly:
}

```

Motor Shield NodeMCU DevKit GPIO Purpose

D1 PWMA (Motor A) D1 5 Speed
D3 DIRA (Motor A) D3 0 Direction
D2 PWMA (Motor B) D2 4 Speed
D4 DIRB (Motor B) D4 2 Direction