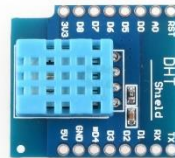


LASKKIT



LaskKit IoT MINI Starter Kit

Úvod	2
Příprava.....	2
Podpora modulu ESP8266 v Arduino IDE	2
Instalace knihoven pro čidlo DHT11	4
Lekce 1 - Test funkčnosti	4
Lekce 2 - Měření teploty a vlhkosti + výpis hodnot na sériový port	5
Lekce 3 – Začínáme s WiFi.....	6
Lekce 4 – Posíláme data na server	7
Lekce 5 – OLED displej.....	9

Úvod

Základní sada pro začátek s IoT, je postavena na modulu WeMos D1 mini, který obsahuje modul ESP8266 s WiFi.

Jako další je v kitu použit WeMos D1 mini DHT11 shield, pomocí kterého lze měřit teplotu a vlhkost okolního vzduchu.

Celý kit lze napájet pomocí mikro USB konektoru na desce WeMos D1, nebo pomocí pinů. Přes stejné USB se kit i programuje.

K této sadě lze dokoupit celá řada dalšího příslušenství. Najít ho lze v našem eshopu v sekci [Shield moduly - ESP a Wemos](#) označené jako **WeMos D1 mini xxx shield**.

Příprava

Kromě samotné desky **Wemos D1 mini** budeš potřebovat vývojové prostředí **Arduino IDE**, ve kterém budeš vyvíjet vlastní software pro desku a pomocí něho budeš software do desky i nahrávat.

Pokud ještě Arduino IDE nemáš, můžeš si ho nainstalovat podle našeho návodu - [Začínáme s Arduinem](#).

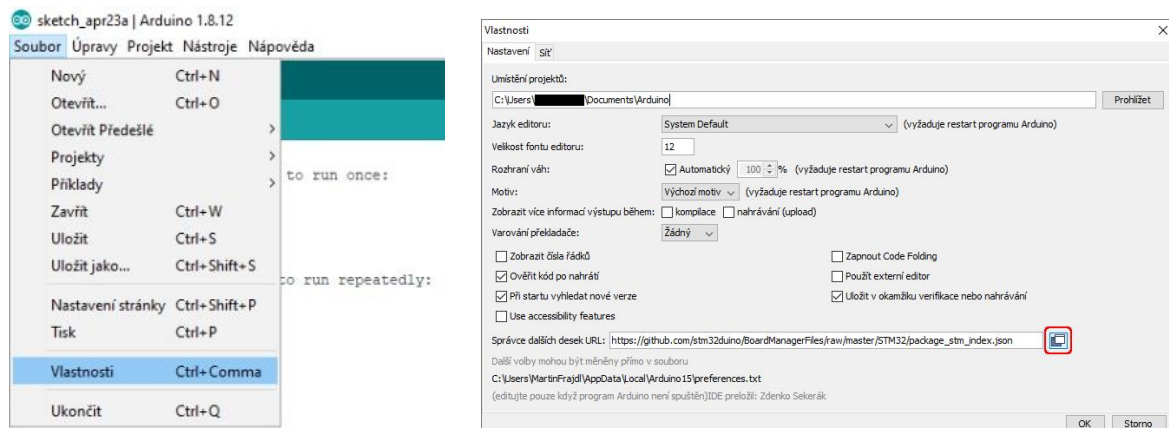
Jako další bude potřeba nainstalovat ovladače USB převodníku pro nahrávání softwaru. Převodník je založen na čipu CH340 a pro instalaci můžeš zase využít našeho návodu - [Instalace ovladače převodníku USB na UART CH340](#).

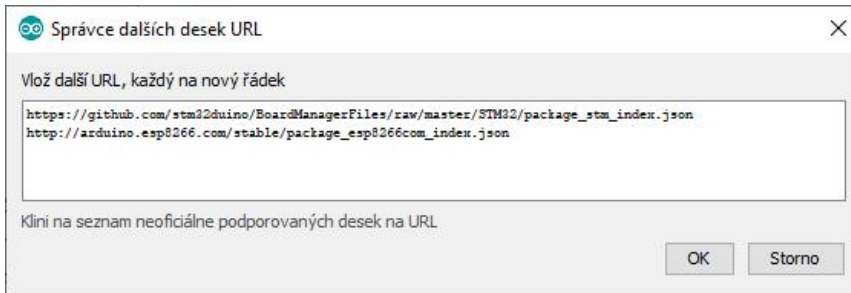
Jakmile budeš mít oboje nainstalováno, můžeš pokračovat zprovozněním podpory čipu ESP8266 v Arduino IDE.

Podpora modulu ESP8266 v Arduino IDE

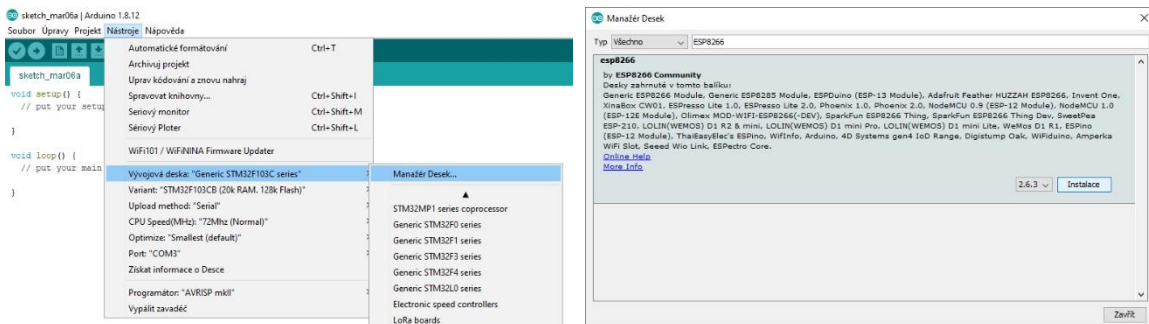
Spusť Arduino IDE a v menu **Soubor - Vlastnosti** do seznamu **Správce dalších desek URL** přidej nový řádek:

http://arduino.esp8266.com/stable/package_esp8266com_index.json

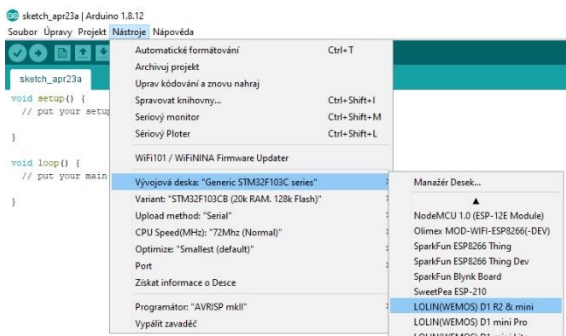




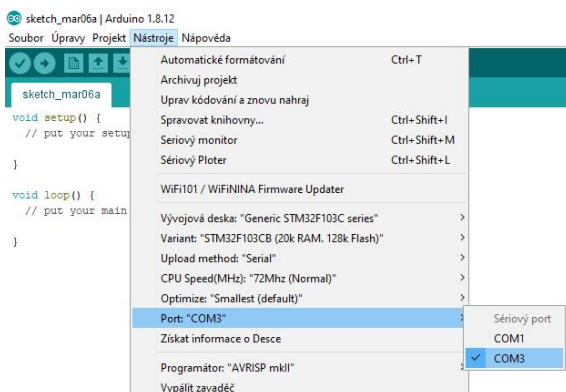
Potom v menu **Nástroje - Vývojová deska - Manažér desek** zadej do horního vyhledávacího políčka **ESP8266** a klepni na tlačítko **Instalace**.



Po nainstalování v menu **Nástroje - Vývojová deska** vyber desku **LOLIN(WEMOS) D1 R2 & mini**.



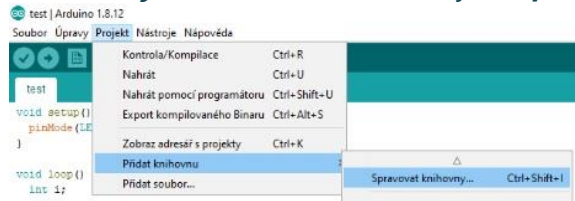
Jako poslední je potřeba zadat port, po kterém se bude deska programovat. V menu **Nástroje - Port** vyber COM port, na který se ti nainstaloval převodník v desce. V našem případě to je COM3.



Instalace knihoven pro čidlo DHT11

Abys mohl začít měřit teplotu a vlhkost tímto čidlem, je potřeba do Arduino IDE doinstalovat potřebné knihovny.

V menu **Projekt – Přidat knihovnu** vyber **Sprovozt knihovny**.



Zadej do horního vyhledávacího políčka **DHT11** a najdi knihovnu **DHT sensor library** by **Adafruit**.



Klepní na tlačítko **Instalace**.

Objeví se okno s upozorněním na další knihovny potřebné pro instalovanou knihovnu. Bez jejich instalací by knihovna nefungovala. Tobě však stačí jen klepnout na tlačítko **Install all** a vše potřebné se automaticky doinstaluje.




Jakmile instalace knihoven skončí, máš podporu čidla DHT11 v Arduino IDE hotovou.

Lekce 1 - Test funkčnosti

Příprava je hotová, nyní můžeš vyzkoušet, že vše funguje. K tomu účelu použij jednoduchý program, který rozbliká LED diodu na desce.

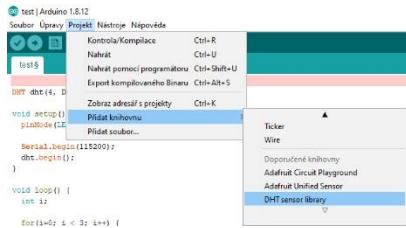
```
void setup() {  
  pinMode(LED_BUILTIN, OUTPUT);  
}  
  
void loop() {  
  int i;  
  
  for(i=0; i < 3; i++) {  
    digitalWrite(LED_BUILTIN, HIGH);  
    delay(100);  
  
    digitalWrite(LED_BUILTIN, LOW);  
    delay(100);  
  }  
  
  delay(500);  
  
  for(i=0; i < 2; i++) {  
    digitalWrite(LED_BUILTIN, HIGH);  
    delay(500);  
  
    digitalWrite(LED_BUILTIN, LOW);  
    delay(500);  
  }  
}
```

Nahraj program do desky tlačítkem **Nahrát** . Po úspěšném nahrání programu do desky se program automaticky spustí a LEDka začne blikat.

Tím máš vše připravené a můžeš se pustit do zprovoznění čidla teploty a vlhkosti DHT11, které je osazené na přiloženém shieldu.

Lekce 2 - Měření teploty a vlhkosti + výpis hodnot na sériový port

Jako první je potřeba do zdrojového kódu přidat potřebné knihovny. V menu **Projekt – Přidat knihovnu** klepni na řádek **DHT sensor library**.



Tím se na začátek zdrojového kódu vloží potřebné řádky pro import knihoven.

```
#include <DHT.h>
#include <DHT_U.h>
```

Následující vzorový kód načte hodnoty z čidla a vypíše je na sériový port.

```
// Vzorový kód pro čidlo teploty a vlhkosti okolního vzduchu DHT11, nebo DHT22
// Použijte knihovny: DHT sensor library by Adafruit
// laskarduino.cz obchod@laskarduino.cz

#include <DHT.h>
#include <DHT_U.h>

#define DHTPIN D4 // Pin, na kterém je připojena komunikace s čidlem
uint32_t delayMS = 2*1000; // Prodleva mezi měřeními - 2*1000ms = 2s
// (prodleva by neměla být menší než 2s, měření v čidle je velmi pomalé)

// Kód se dá použít jak pro čidlo DHT11, tak i pro DHT22.
// Podle typu použitého čidla se odkomentuje příslušný řádek a ten druhý zakomentuje
#define DHTTYPE DHT11 // Typ použitého čidla - DHT 11
// #define DHTTYPE DHT22 // Typ použitého čidla - DHT 22 (AM2302)

DHT dht(DHTPIN, DHTTYPE); // Inicializace čidla DHT11

void setup() {
  Serial.begin(115200); // Spuštění komunikace po seriovém portu - rychlost 115200baud
  dht.begin(); // Spuštění komunikace s čidlem
}

void loop() {
  Serial.println("Namerené údaje z čidla DHT" + String(DHTTYPE) + ":");

  // Doba načtení každé hodnoty je cca 250ms, takže načtení údajů z čidla potrvá cca 0,5s
  float temperature = dht.readTemperature(); // Načtení teploty z čidla
  int humidity = dht.readHumidity(); // Načtení vlhkosti z čidla

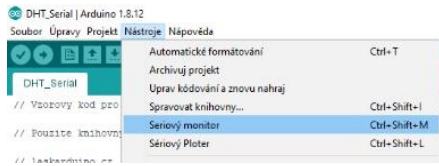
  // Kontrola, zda načtení údajů z čidla proběhlo v pořádku
  if (isnan(temperature) || isnan(humidity)) {
    Serial.println(F("Chyba čtení údajů z DHT čidla!")); // Pokud čtení údajů z čidla neproběhlo
    // v pořádku, vypíšeme chybové hlášení
    // na seriový port
    return;
  }

  // Na seriový port vypíšeme aktuální změřená data z čidla
  Serial.print(F("Teplota: "));
  Serial.print(String(temperature, 1)); // Zaokrouhlení teploty na jedno desetinné číslo
  Serial.println("°C");

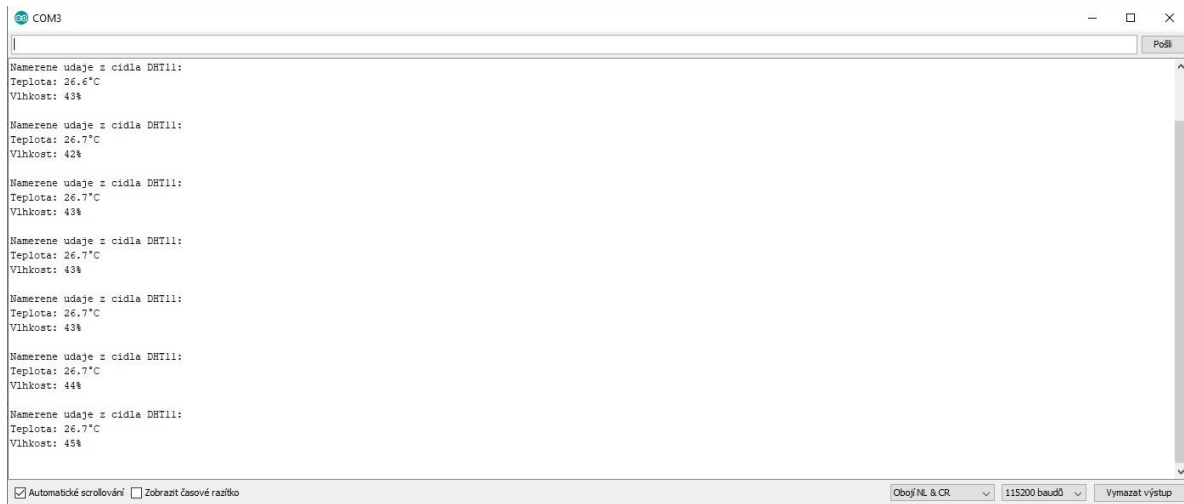
  Serial.print(F("Vlhkost: "));
  Serial.print(humidity);
  Serial.println("%");
  Serial.println("");

  delay(delayMS); // Prodleva mezi měřeními
}
```

Po nahrání programu do desky si spustí v menu **Nástroje – Seriový monitor**.



V okně monitoru uvidíš vypisovat naměřené hodnoty teploty a vlhkosti:



Lekce 3 – Začínáme s WiFi

Jako první se budeš připojovat k domácí WiFi síti. Pro práci s WiFi budeš potřebovat do projektu přidat knihovnu pro práci s WiFi v ESP8266.

```
#include <ESP8266WiFi.h>
```

Nadefinuj si dvě proměnné – v jedné bude název tvé domácí WiFi sítě a v druhé bude heslo, které o ni používáš.

```
const char* wifi_ssid = "navez_wifi_site"; // Nazev WiFi site
const char* wifi_heslo = "heslo_k_wifi_siti"; // Heslo k WiFi siti
```

Potom v proceduře `setup()` spustíš vlastní připojení k WiFi síti.

```
Serial.println(F("Pripojuji k WiFi siti"));
WiFi.begin(wifi_ssid, wifi_heslo); // Spusteni pripojeni k WiFi AP
// Cekani na pripojeni k AP, po dobu pripojovani vypisuje na seriový port tecku
while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.print(".");
}
Serial.println("");
Serial.print(F("Pripojeno k WiFi siti "));
Serial.println(wifi_ssid);
Serial.print(F("IP adresa: "));
Serial.println(WiFi.localIP()); // Vypise na seriovou konzoli pridelenou IP adresu
Serial.println("");
```

Tím je modul připojen přes WiFi k internetu a můžeš začít posílat data na nějaký server.

Lekce 4 – Posíláme data na server

Pro ukládání tvých dat z modulu se dá použít třeba server <https://thingspeak.com/>. Jak si založit účet a zprovoznit ho najdeš v našem článku [IoT Backend – ThingSpeak](#).

Podle návodu si vytvoř dva kanály **Teplota** a **Vlhkost**. Do nich budeš posílat naměřená data z čidla DHT11 z [Lekce 2](#).

V menu **Projekt – Přidat knihovnu – Spravovat knihovny** vyhledej a nainstaluj knihovnu **ThingSpeak** by **MathWorks**.

Přidej do projektu knihovnu:

```
#include "ThingSpeak.h"
```

Nadefinuj si proměnné a inicializuj klienta:

```
const char* api_klic = "api_klic"; // Tvuj API klic na ThingSpeak - Write API Key
unsigned long cislo_kanaluu = cislo_kanaluu; // Cislo tveho kanalu - Channel ID

WiFiClient client; // Inicializace klienta
```

V proceduře `setup()` spust' komunikaci se serverem ThingSpeak:

```
ThingSpeak.begin(client); // Spusteni komunikace se serverem ThingSpeak
```

A posílej data na server:

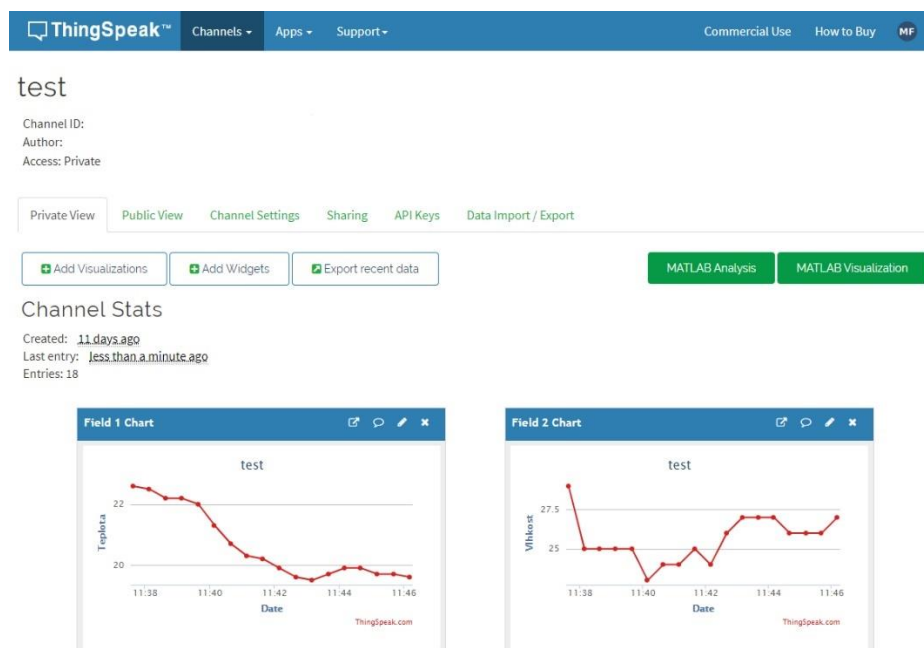
```
ThingSpeak.setField(1, temperature); // Priprava dat pro field1 - Teplota
ThingSpeak.setField(2, humidity); // Priprava dat pro field2 - Vlhkost

int httpCode = ThingSpeak.writeFields(cislo_kanaluu, api_klic); // Odeslani dat na server

// Kontrola, zda odeslani na server probehlo v poradku, popr. vypise chybovy kod
if (httpCode == 200) {
  Serial.println(F("Data uspesne odeslana na server."));
} else {
  Serial.println("Problem s odeslanim dat. Chyba: " + String(httpCode));
}

Serial.println("");
```

Jestli vše funguje, zjistiš na stránce kanálu v záložce **Private View**.



Zde máš výsledný kompletní kód:

```
// Vzorovy kod pro cidlo teploty a vlhkosti okolniho vzduchu DHT11, nebo DHT22

// Pouzite knihovny:
// DHT sensor library by Adafruit, ThingSpeak by MathWorks

// laskarduino.cz  obchod@laskarduino.cz

#include <DHT.h>
#include <DHT_U.h>
#include <ESP8266WiFi.h>
#include "ThingSpeak.h"

#define DHTPIN D4 // Pin, na kterem je pripojena komunikace s cidlem
uint32_t delayMS = 30*1000; // Prodleva mezi merenimi - 2*1000ms = 2s
// (prodleva by nemela byt mensi nez 2s, mereni v cidle je velmi pomale)
// pokud se budou odesilat data na server ThingSpeak je potreba nastavit prodlevu na >= 30s

// Kod se da pouzít jak pro cidlo DHT11, tak i pro DHT22.
// Podle typu pouzitého cidla se odkomentuje prislusny radek a ten druhy zakomentuje
#define DHTTYPE DHT11 // Typ pouzitého cidla - DHT 11
// #define DHTTYPE DHT22 // Typ pouzitého cidla - DHT 22 (AM2302)

const char* wifi_ssid = "nazev_wifi_siti"; // Nazev WiFi siti
const char* wifi_heslo = "heslo_k_wifi_siti"; // Heslo k WiFi siti

const char* api_klic = "api_klic"; // Tvuj API klic na ThingSpeak
unsigned long cislo_kanaluu = cislo_kanaluu; // Cislo tveho kanalu - Channel ID

WiFiClient client; // Inicializace klienta

DHT dht(DHTPIN, DHTTYPE); // Inicializace cidla DHT11

void setup() {
  Serial.begin(115200); // Spusteni komunikace po seriovem portu - rychlost 115200baud
  dht.begin(); // Spusteni komunikace s cidlem

  Serial.println("");
  Serial.println(F("Pripojuji k WiFi siti"));
  WiFi.begin(wifi_ssid, wifi_heslo); // Spusteni pripojeni k WiFi AP

  // Cekani na pripojeni k AP, po dobu pripojovani vypisuje na seriový port tecku
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  Serial.println("");
  Serial.print(F("Pripojeno k WiFi siti "));
  Serial.println(wifi_ssid);
  Serial.print(F("IP adresa: "));
  Serial.println(WiFi.localIP()); // Vypise na seriovou konzoli pridelenou IP adresu
  Serial.println("");

  ThingSpeak.begin(client); // Spusteni komunikace se serverem ThingSpeak
}

void loop() {
  Serial.println("Namerene udaje z cidla DHT" + String(DHTTYPE) + " :");

  // Doba nacteni kazde hodnoty je cca 250ms, takze nacteni udaju z cidla potrvá cca 0,5s
  float temperature = dht.readTemperature(); // Nacteni teploty z cidla
  float humidity = dht.readHumidity(); // Nacteni vlhkosti z cidla

  // Kontrola, zda nacteni udaju z cidla probehlo v poradku
  if (isnan(temperature) || isnan(humidity)) {
    Serial.println(F("Chyba cteni udaju z DHT cidla!")); // Pokud cteni udaju z cidla neproběhlo
// v poradku, vypiseme chybove hlášení
// na seriový port
    return;
  }

  // Na seriový port vypiseme aktualni zmerena data z cidla
  Serial.print(F("Teplota: "));
  Serial.print(String(temperature, 1)); // Zaokrouhleni teploty na jedno desetinne cislo
  Serial.println("°C");

  Serial.print(F("Vlhkost: "));
  Serial.print(String(humidity, 0)); // Zaokrouhleni vlhkosti na cele cislo
  Serial.println("%");

  ThingSpeak.setField(1, temperature); // Priprava dat pro field1 - Teplota
  ThingSpeak.setField(2, humidity); // Priprava dat pro field2 - Vlhkost

  int httpCode = ThingSpeak.writeFields(cislo_kanaluu, api_klic); // Odeslani dat na server

  // Kontrola, zda odeslani na server probehlo v poradku, popr. vypise chybovy kod
  if (httpCode == 200) {
    Serial.println(F("Data uspesne odeslana na server."));
  } else {
    Serial.println("Problem s odeslanim dat. Chyba: " + String(httpCode));
  }

  Serial.println("");
  delay(delayMS); // Prodleva mezi merenimi
}
```

Lekce 5 – OLED displej

Dalším užitečným modulem je modul s OLED displejem. Jedná se o monochromatický grafický 0,66“ displej s rozlišením 64x48 bodů a komunikačním rozhraním I2C. Pro jeho používání si nejprve v Arduino IDE, menu **Projekt – Přidat knihovnu – Spravovat knihovny**, nainstaluj knihovny **Adafruit SSD1306 Wemos Mini OLED** a **Adafruit GFX Library**. Pokud při instalaci knihoven bude chtít instalátor doinstalovat nějaké další knihovny, je potřeba nainstalovat i je.

Přidej knihovny do projektu:

```
#include <Adafruit_SSD1306.h>
#include <Adafruit_GFX.h>
```

Inicializuj OLED displej:

```
Adafruit_SSD1306 display(0); // 0 - reset pin, u I2C se nepoužívá, ale je vyžadován
```

V proceduře `setup()` spusť komunikaci s displejem:

```
display.begin(SSD1306_SWITCHCAPVCC, 0x3D); // komunikace s displejem na I2C adrese 0x3C
```

V následujícím vzorovém kódu se po spuštění napájení vykreslí na displeji naše logo a po 1s se začnou zobrazovat naměřené hodnoty z čidla DHT11:

```
#include <DHT.h>
#include <ESP8266WiFi.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define DHTPIN 2
#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);
Adafruit_SSD1306 display(0); // 0 - reset pin, u I2C se nepoužívá, ale je vyžadován

uint32_t delayMS = 2000;

// Nadefinování dat pro zobrazení bmp loga na displeji
static const unsigned char PROGMEM laskarduino_glcd_bmp[] = {
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x1c, 0x0e, 0x00, 0x00, 0x00,
  0x00, 0x3c, 0x1f, 0x00, 0x00, 0x00,
  0x00, 0x7e, 0x1f, 0x00, 0x00, 0x00,
  0x00, 0x7f, 0xff, 0x00, 0x00, 0x00,
  0x00, 0x7f, 0xff, 0x00, 0x00, 0x00,
  0x00, 0x3f, 0xff, 0x03, 0x00, 0x00,
  0x00, 0x3f, 0xff, 0x8f, 0x80, 0x00,
  0x0f, 0xff, 0xff, 0xff, 0x80, 0x00,
  0x1f, 0xff, 0xff, 0xff, 0x80, 0x00,
  0x1f, 0xff, 0x3f, 0xff, 0x00, 0x00,
  0x1f, 0xfc, 0x0f, 0xfe, 0x00, 0x00,
  0x07, 0xf8, 0x07, 0xfe, 0x00, 0x00,
  0x03, 0xf8, 0x01, 0xff, 0x00, 0x78,
  0x01, 0xf8, 0x00, 0xfe, 0x00, 0xfc,
  0x03, 0xf8, 0x00, 0x7c, 0x01, 0xce,
  0x07, 0xf8, 0x00, 0x39, 0xff, 0x86,
  0x1f, 0xf8, 0x00, 0x03, 0xff, 0xce,
  0x3f, 0xf8, 0x00, 0x07, 0x00, 0xfc,
  0x1f, 0xfc, 0x0f, 0xfe, 0x00, 0x78,
  0x1f, 0xfe, 0x0f, 0xfc, 0x3c, 0x00,
  0x0c, 0xfe, 0x00, 0x00, 0x7e, 0x00,
  0x00, 0x7f, 0x00, 0x1f, 0xe7, 0x00,
  0x00, 0x3f, 0x0f, 0xff, 0xc7, 0x00,
  0x00, 0x3f, 0x07, 0xff, 0xe6, 0x00,
  0x00, 0x7f, 0x00, 0x00, 0x7e, 0x00,
  0x0c, 0xfe, 0x0f, 0xfc, 0x18, 0x78,
  0x1f, 0xfe, 0x0f, 0xfe, 0x00, 0xfc,
  0x1f, 0xfc, 0x00, 0x07, 0x01, 0xce,
  0x3f, 0xf8, 0x00, 0x03, 0xff, 0x86,
  0x1f, 0xf8, 0x00, 0x19, 0xff, 0xce,
  0x07, 0xf8, 0x00, 0x3c, 0x00, 0xfc,
  0x03, 0xf0, 0x00, 0x7e, 0x00, 0x78,
  0x01, 0xf0, 0x00, 0xff, 0x00, 0x00,
  0x03, 0xf8, 0x01, 0xff, 0x00, 0x00,
  0x07, 0xf8, 0x07, 0xfe, 0x00, 0x00,
  0x1f, 0xfc, 0x0f, 0xfe, 0x00, 0x00,
  0x1f, 0xff, 0x1f, 0xff, 0x00, 0x00,
  0x1f, 0xff, 0xff, 0xff, 0x80, 0x00,
  0x0f, 0xff, 0xff, 0xff, 0x80, 0x00,
  0x00, 0x3f, 0xff, 0x8f, 0x80, 0x00,
  0x00, 0x3f, 0xff, 0x03, 0x00, 0x00,
  0x00, 0x7f, 0xff, 0x00, 0x00, 0x00,
  0x00, 0x7f, 0xff, 0x00, 0x00, 0x00,
  0x00, 0x7e, 0x1f, 0x00, 0x00, 0x00,
  0x00, 0x3c, 0x1f, 0x00, 0x00, 0x00,
  0x00, 0x18, 0x0e, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00
};

void setup() {
  Serial.begin(9600);
```

```
dht.begin();
Serial.println(F("DHTxx Unified Sensor Example"));

display.begin(SSD1306_SWITCHCAPVCC, 0x3D); // komunikace s displejem na I2C adrese 0x3C
display.clearDisplay();
display.setRotation(1);
display.drawBitmap(0, (64-48)/2, laskarduino_glcd_bmp, 48, 48, 1);
display.display();
delay (1000);
}

void loop() {
  Serial.println("DHT11 Sensor Example");

  float temperature = dht.readTemperature();
  int humidity = dht.readHumidity();

  if (isnan(temperature) || isnan(humidity)) {
    Serial.println(F("Failed to read from DHT sensor!"));
    return;
  }

  Serial.print("Temperature: ");
  Serial.print(String(temperature,1)); // Zaokrouhleni na xx,x
  Serial.println("°C");

  Serial.print("Humidity: ");
  Serial.print(humidity);
  Serial.print("%");
  Serial.println("");
  Serial.println("");

  // Vypis hodnot na displej
  display.clearDisplay();
  display.setTextSize(2);
  display.setRotation(0);
  display.setTextColor(WHITE);
  display.setCursor(0,2); // X ; Y start
  display.println(String(temperature,1) + "C");
  display.setCursor(22 ,30); // X ; Y start
  display.println(String(humidity) + String("%"));

  display.display();

  // Prodlava mezi merenimi
  delay(delayMS);
}
```

Výsledek by měl vypadat takto:

