

# **ZÁVĚREČNÁ STUDIJNÍ PRÁCE**

## **dokumentace**

### **Dávkovač krmiva**

Ondřej Marek

**Obor:** 18-20-M/01 INFORMAČNÍ TECHNOLOGIE  
se zaměřením na počítačové sítě a programování

**Třída:** IT4

**Školní rok:** 2020/2021

### ***Poděkování***

*Děkuji panu učiteli Ing. Petru Grussmannovi za cenné rady. Také bych rád poděkoval i mým spolužákům za pomoc při řešení problémů.*

Prohlašuji, že jsem závěrečnou práci vypracoval samostatně a uvedl veškeré použité informační zdroje.

Souhlasím, aby tato studijní práce byla použita k výukovým účelům na Střední průmyslové a umělecké škole v Opavě, Praskova 399/8.

V Opavě      31. 12. 2016

---

*podpis autora práce*

## **ANOTACE**

Cílem tohoto projektu bylo vytvoření dávkovače krmiva pro moji andulku. Uživatel by měl být schopen pomocí webové aplikace nastavit čas krmení.

Toto zařízení je řízeno pomocí procesoru ESP8266, který se automaticky připojuje na WiFi a poté vytváří samostatný web server. Na web serveru je uživatel schopen nastavovat čas krmení. Dávkovač se poté spustí v uživatelem nastaveném čase. Softwarový program je napsán v jazyce Wiring, který byl přímo vytvořen k programování integrovaných obvodů.

# **OBSAH**

<b>ÚVOD.....</b>	<b>5</b>
<b>1 VÝROBA DÁVKOVAČE KRMIVA .....</b>	<b>6</b>
<b>2 VYUŽITÉ TECHNOLOGIE .....</b>	<b>7</b>
2.1 HARDWARE .....	7
2.1.1 ESP8266 WeMos D1 mini .....	7
2.1.2 Servo motor SG90.....	7
2.2 SOFTWARE .....	7
2.2.1 Jazyk Wiring .....	7
2.2.2 Visual Studio Code .....	8
2.2.3 PlatformIO.....	8
<b>3 ZPŮSOBY ŘEŠENÍ A POUŽITÉ POSTUPY.....</b>	<b>9</b>
3.1 SCHÉMATICKÉ ZAPOJENÍ .....	9
3.2 POPIS FUNGOVÁNÍ DÁVKOVAČE .....	9
3.3 PŘIPOJENÍ K WiFi.....	9
3.4 KOMUNIKACE SE SERVEREM .....	10
3.1 ZÍSKÁVÁNÍ ČASU .....	11
<b>4 VÝSLEDKY ŘEŠENÍ.....</b>	<b>12</b>
4.1 PODOBA DÁVKOVAČE.....	12
4.2 WEBOVÁ APLIKACE.....	12
<b>ZÁVĚR .....</b>	<b>13</b>
<b>SEZNAM POUŽITÝCH INFORMAČNÍCH ZDROJŮ .....</b>	<b>14</b>
<b>SEZNAM PŘÍLOH.....</b>	<b>15</b>

## ÚVOD

Rozhodl jsem se vytvořit dávkovač krmiva se vzdáleným ovládáním prostřednictvím webové aplikace, protože jsem si chtěl v této problematice vyzkoušet pracovat a něco nového se naučit.

Cílem tohoto projektu bylo vytvoření dávkovače krmiva pro moji andulku, u kterého budu schopen měnit čas krmení pomocí webové aplikace.

Ve své dokumentaci zmiňuji použité technologie a rozebírám, jak funguje nejen samotný dávkovač krmiva, ale i webová aplikace potřebná k jeho vzdálenému ovládání. V další části vysvětluji, na jakých principech funguje jejich vzájemná komunikace, a popisuji jednotlivé úkony obou částí systému.

## 1 VÝROBA DÁVKOVAČE KRMIVA

Na začátku projektu byl důležitý výběr správných součástek a technologií. Pro můj projekt mi pan učitel Grussman doporučil použít vývojovou desku WeMos D1 mini, založená na ESP8266EX. K ní jsem dokoupil malý servo motor.

S nakoupenými součástkami jsem nikdy nepracoval, jejich funkčnost jsem testoval pomocí jednoduchých příkladů. Stejným způsobem jsem testoval i použité knihovny jako např. NTPClient nebo ESPAsyncWiFiManager.

Po zapojení a odzkoušení všech součástek jsem přešel k programování samotného dávkovače. Přirozeně jsem zvolil programovací jazyk vytvořený přímo pro Arduino, což je kombinace jazyků C a C++. Začal jsem tím, že jsem připojil ESP8266 k WiFi a pomocí něj jsem vytvořil web server, na kterém běží webová aplikace, pomocí níž se měli nastavovat časy spuštění motoru. Poté jsem napsal funkci pro spuštění krmení.

Po dokončení kódu bylo potřeba sestavit jednoduchou konstrukci dávkovače, vyzkoušet ho v praxi a také doladit rychlost otáčení motorku, aby se dávkovalo správné množství krmiva.

## 2 VYUŽITÉ TECHNOLOGIE

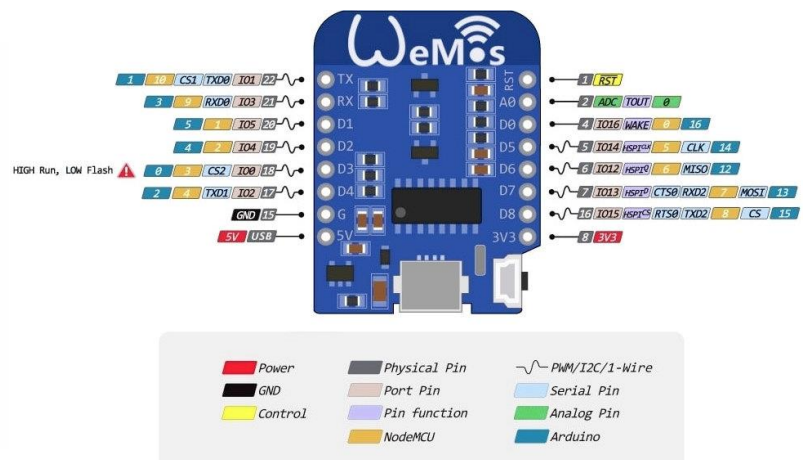
### 2.1 Hardware

#### 2.1.1 ESP8266 WeMos D1 mini

WeMos D1 mini je deska postavená na Wi-Fi modulu ESP8266 od firmy Espressif. Obsahuje šestnáct pinů z toho devět digitálních a jeden analogový a další jako napájení, výstup 5 V a 3,3 V.



Obrázek č. 2 ESP8266 WeMos D1 mini



Obrázek č. 1 Piny ESP8266 WeMos D1 mini

#### 2.1.2 Servo motor SG90

Servo motor SG90 je malý jednoduchý motor, který se dokáže otáčet o 180° a funguje jako standartní servo. Váží pouhých 9 gramů, ale pro mé účely je dostačující.

### 2.2 Software

#### 2.2.1 Jazyk Wiring

Wiring je programovací jazyk vytvořený pro programování mikrokontrolerů bez specifických znalostí hardwaru. Až na drobné úpravy, velmi podobný jazyku C nebo C++. Jazyk Wiring byl přímo vytvořen k programování integrovaných obvodů.

### **2.2.2 Visual Studio Code**

Visual Studio Code je editor zdrojového kódu vyvíjený společností Microsoft pro operační systémy Windows, Linux a macOS. Obsahuje podporu pro Git, zvýraznění syntaxe, kontextový našeptávač a podporu pro ladění a refaktorizaci.

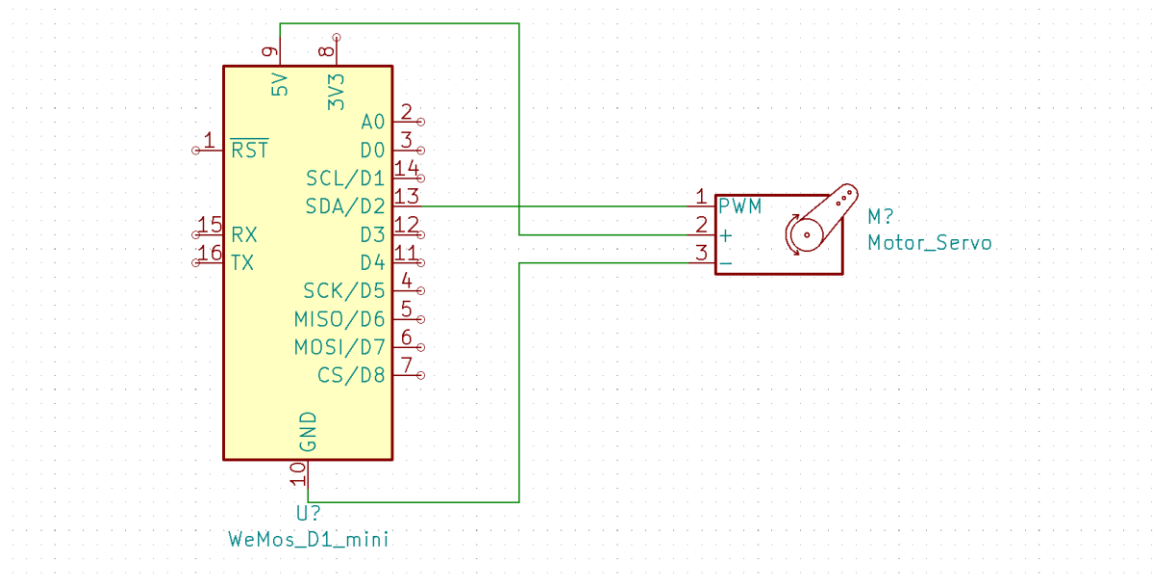
### **2.2.3 PlatformIO**

PlatformIO je vývojový ekosystém kombinující IDE, tedy nástroj pro zápis zdrojového kódu, správce knihoven a nástroje pro překlad a nahrání výsledného projektu do MCU.



### 3 ZPŮSOBY ŘEŠENÍ A POUŽITÉ POSTUPY

#### 3.1 Schématické zapojení



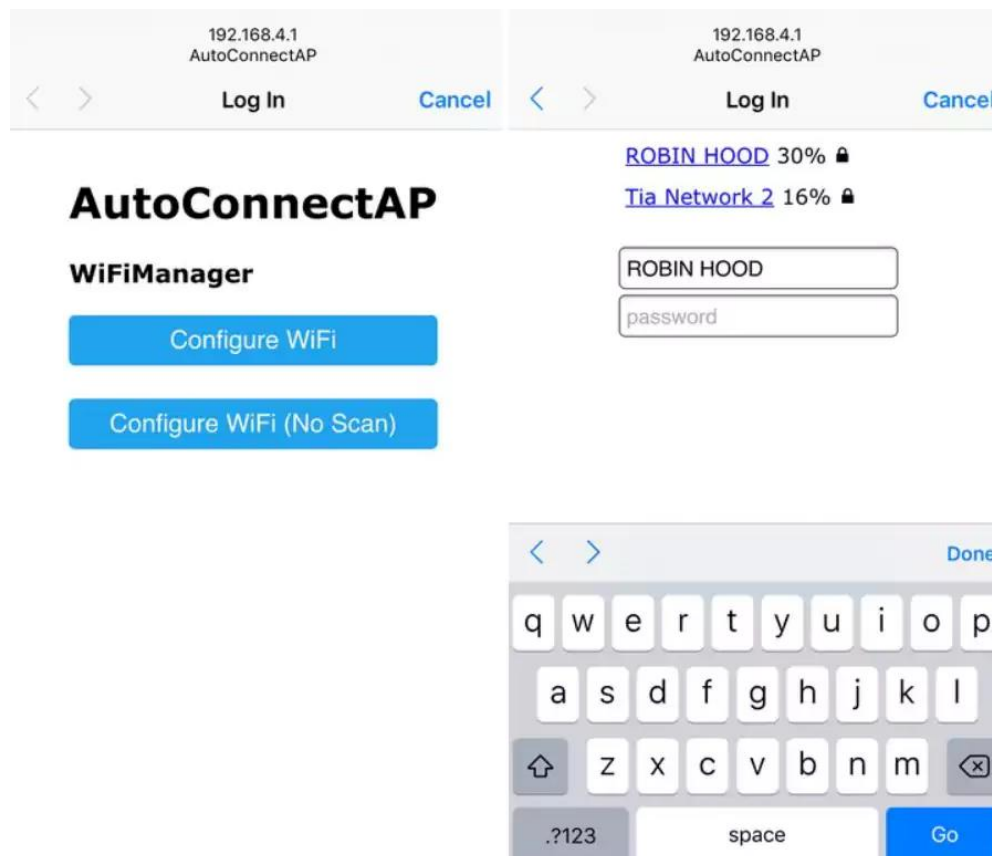
Obrázek č. 3 Schéma zapojení

#### 3.2 Popis fungování dávkovače

Po zapnutí ESP8266 se automaticky spustí funkce `setup()`, ve které se spustí funkce pro připojení k WiFi pomocí knihovny `WiFiManager`, poté se spustí funkce, pomocí níž zařízení komunikuje se serverem. Pak následuje funkce `loop()`, ve které se spustí funkce `feed()`, která zjišťuje, jestli se čas krmení, který je zadán uživatelem shoduje s momentálním časem, pokud tomu tak je spouští servo motor, který zajišťuje krmení.

#### 3.3 Připojení k WiFi

Pro připojení k WiFi jsem využil knihovnu `ESPAsyncWiFiManager`. Zařízení se při zapnutí automaticky pokusí připojit k WiFi síti bez hesla nebo již nakonfigurované WiFi síti. Pokud se to nepodaří, automaticky vytvoří konfigurační portál, na kterém uživatel zadá heslo ke své WiFi a ESP8266 se k ní pak bude automaticky připojovat.



Obrázek č. 4 WifiManager

### 3.4 Komunikace se serverem

Po tom, co jsem napsal jednoduchou html stránku, na níž je uživatel schopen zadat čas krmění, kterou budu využívat jako vzdálené ovládání dávkovače. Jsem vytvořil web serveru pomocí knihovny ESPAsyncWebServer. Komunikaci se serverem zajišťují HTTP GET a POST požadavky z ESP8266 na server.

```
server.on("/", HTTP_GET, [](AsyncWebServerRequest *request) {
    request->send(SPIFFS, "/index.html", String(), false, processor);
});
server.on("/style.css", HTTP_GET, [](AsyncWebServerRequest *request) {
    request->send(SPIFFS, "/style.css", "text/css");
});
server.on("/main.js", HTTP_GET, [](AsyncWebServerRequest *request) {
    request->send(SPIFFS, "/main.js", String());
});
```

### **3.1 Získávání času**

Čas získávám pomocí knihovny NTPClient. Pro fungování je nutné, aby bylo Arduino připojeno na Wifi, poté se připojí na NTP server, ze kterého získává čas. Získaný čas poté ve funkci porovnávám s časem získaným z webové aplikace, pokud se časy budou rovnat spustí se v programu funkce pro krmení.

## 4 VÝSLEDKY ŘEŠENÍ

### 4.1 Podoba dávkovače

Dávkovač krmiva je v současné době plně funkční a funguje bez problému. Podobu budu, ale v nejbližší budoucnosti předělávat. Zařízení je totiž momentálně hodně křehké a vizuálně nevypadá dobře. Momentálně je motorek připojený k PET lahvi, ve které je umístěno krmivo a nepájivé pole leží nemotorně vedle zařízení.

### 4.2 Webová aplikace

Webová aplikace funguje tak jak jsem si představoval. Uživatel dokáže čas krmení nastavit a kdykoliv ho změnit. Aplikace je také responzivní, lze ji proto používat i na mobilních telefonech nebo tabletech. V budoucnu bych chtěl aplikaci vylepšit nastavením více času krmení a spolu s určitými dny krmení.



Obrázek č. 5 Vzhled webové aplikace

## Závěr

Cílem projektu bylo vytvořit dávkovač krmiva se vzdáleným ovládáním prostřednictvím webové aplikace. Určené cíle byli splněny. Do budoucna bych určitě vylepšil vizuální podobu dávkovače krmiva. Při vývoji mě také napadlo několik možností změn, které by zařízení vylepšily, jako například nastavením více časů krmení, které by se ukládali do JSON souboru, spolu s určitými dny krmení. Těmto vylepšením bych se chtěl v nejbližší budoucnosti věnovat.

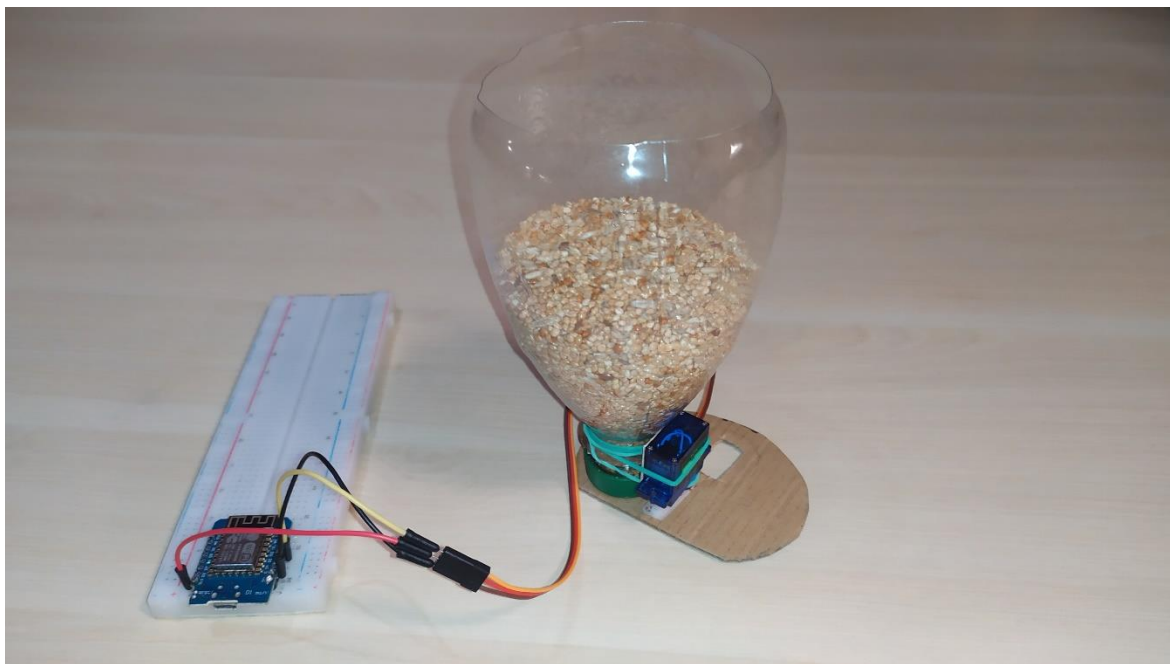
## SEZNAM POUŽITÝCH INFORMAČNÍCH ZDROJŮ

- [1] STEREMBERG, Alan. ESPAsyncWiFiManager (github) [online].  
[cit. 2020-12-28].  
<<https://github.com/alanswx/ESPAsyncWiFiManager>>.
- [2] NTPClient (github) [online].  
[cit. 2020-12-28].  
<<https://github.com/taranais/NTPClient>>.
- [3] ESPAsyncWebServer (github) [online].  
[cit. 2020-12-28].  
<<https://github.com/me-no-dev/ESPAsyncWebServer>>.
- [4] <<https://www.arduino.cc/en/Tutorial/LibraryExamples/Sweep>>.

## **SEZNAM PŘÍLOH**

č. 1 Fotodokumentace

## Příloha č. 1: Fotodokumentace



Vzhled dávkovače krmiva