

ZÁVĚREČNÁ STUDIJNÍ PRÁCE

dokumentace

GPS sledovač pro psy

Michal Kurečka



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

Třída: IT4

Školní rok: 2021/2022

Poděkování

Jako první bych chtěl poděkovat svému pejskovi, který mě dovedl k tomuto projektu, dále bych chtěl poděkovat panu Ing. Petru Grusmannovi za pomoc s komunikační a hardwarovou částí projektu, panu Ing. Marceli Godovskému za vypůjčení hardwaru a také panu Mgr. Marku Lučnému, za rady v oblasti webu a webových stránek.

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 31.12.2021

podpis autora práce

ANOTACE

Účelem tohoto GPS sledovače je pomocí zařízení připevněného na obojku získávat polohu psa a poslat ji na zařízení příjemce, které tato data zpracuje a následně zobrazí na webu.

Projekt se skládá ze tří částí. Hardware obsahuje GPS modul a dvě Esp32 zařízení: jedno je u uživatele (Receiver), druhé na obojku psa (Sender). Softwarová část zajišťuje zpracovávání a posílání informací. Poslední část je webová aplikace, na které se informace zobrazují v přehledném uživatelském rozhraní.

Klíčová slova: uživatel, webová aplikace, GPS modul, uživatelské rozhraní, Esp32

OBSAH

ÚVOD.....	5
1. VYUŽITÉ TECHNOLOGIE.....	6
1.1 Heltec Esp32 LoRa V2.....	6
1.2 SPIFFS	6
1.3 Async Web Server.....	6
1.4 LoRa	6
1.5 PlatformIO.....	6
1.6 Vývojové prostředí	7
1.7 JQuery	7
1.8 Tinkercad	7
1.9 GPS	7
2. ZPŮSOBY ŘEŠENÍ A POUŽITÉ POSTUPY.....	8
2.1 Zapojení hardware	8
2.2 Řešení LoRa komunikace.....	8
2.2.1 Seznámení s LoRa.....	8
2.2.2 Problém s pásmem.....	9
2.3 Získávání GPS souřadnic	9
2.3.1 Princip kódu.....	9
2.3.2 Přesná GPS data.....	10
2.4 Webový server	10
2.4.1 Jak server funguje.....	10
2.4.2 Komunikace serveru s ESP.....	11
2.5 Klientská webová aplikace	12
2.5.1 Design aplikace.....	12
2.5.2 Zobrazení mapy.....	12
2.5.3 Funkčnost stránky.....	13
3. VÝSLEDKY ŘEŠENÍ A TESTY.....	14
3.1 Shrnutí funkcí zařízení	14
3.2 Testy funkcí a dosahu zařízení	14
3.3 Porovnání s konkurencí.....	14
ZÁVĚR.....	15
SEZNAM INFORMAČNÍCH ZDROJŮ.....	16

ÚVOD

Nápad na tento konkrétní projekt vzniknul, když jsem byl jeden den na procházce s mým psem, který se zběhl za srnkou a já ho pak musel půl hodiny hledat. Jelikož s ním chodím na procházky často, a výskyt srnek je v mojí oblasti docela hojný, řekl jsem si, že mu udělám nějaký GPS sledovač, který by mi posílal jeho polohu a já tak nemusel mít strach, že se zase ztratí.

První návrhem bylo využití Arduina s GPS modulem, který by data posílal pomocí SMS zpráv přímo do telefonu, avšak to se po pár konzultacích změnilo. Mým konečným cílem bylo vytvořit dvě zařízení: první Esp32 *Sender* má za úkol získávat pozici z GPS modulu (toto zařízení je na psím obojku a sleduje, kde se pes nachází), kterou pomocí Lo-Ra sítě ve formě packetů s informacemi posílá na druhé Esp32 *Reciever*, který tyto packety přijímá, parsuje a následovně vypisuje na displej a na web.

Ve své prezentaci se nejprve zmiňuji o použitých technologiích, poté o práci na samotném projektu. Nakonec popíšu, jak projekt dohromady funguje, jak obstál v porovnání s konkurencí a jaké byly výsledky testování.

-

1 VYUŽITÉ TECHNOLOGIE

1.1 Heltec Esp32 LoRa V2

Vývojová deska Esp32 je série mikrokontrolerů, pracujících s napětím 3-7V, které společně se zabudovaným Wifi a Bluetooth modulem fungují jako malý počítač. Zvolil jsem tuto vývojovou desku, protože má v sobě společně s Wifi a Bluetooth zabudovaný také LoRa modul a 0,96-palcový OLED display.

1.2 SPIFFS

Je souborový systém, dokáže ukládat soubory do paměti Flash. SPIFFS používám k ukládání dat (především souborů jako index.html nebo style.css), které se poté posílají na web.

1.3 Async Web Server

Asynchronní webový server jsem použil, protože oproti normálnímu webovému serveru se dokáže dynamicky měnit, bez nutné interakce uživatele. Toto je výhoda, při updatování času, mapy a informací o poloze.

1.4 LoRa

LoRa je řešení pro bezdrátový přenos dat, kdy je hlavním cílem co nejnižší spotřeba energie při malých pořizovacích nákladech a postačí malý datový tok. Vzhledem k vysílacímu výkonu v řádu jednotek až desítek miliwattů a občasnému vysílání třeba jen několikrát denně může být životnost baterií zařízení mnoho let. Díky využití tzv. bezlicenčních pásem v rozsahu metrových, resp. decimetrových vln může být v příznivém terénu dosah přes 10 km. LoRa síť jsem zvolil z jednoduchého důvodu, potřebuju poslat malé množství dat co nejefektivněji a v tom LoRa exceluje.

1.5 PlatformIO

Rozšíření ve Visual Studiu, funguje jako spouštěč arduino kódu, je to open source uživatelské rozhraní a debugger pro Arduino a Esp zařízení. PlatformIO jsem použil, kvůli výhodě zobrazení všech knihoven na jednom místě a jednoduchému ovládání.

1.6 Vývojové prostředí

Jako první věc, kterou jsem při děláni mého projektu, musel splnit, bylo vybrat si správné vývojové prostředí, ve kterém bych mohl pracovat na svém projektu. Poprvé jsem chtěl využít aplikaci Arduino IDE, kterou jsem používal již nějakou dobu, avšak pro tento složitější projekt jsem nakonec zvolil Visual Studio Code s rozšířením PlatformIO, které jsem nainstaloval skrz možnost extensions v samotném Visual Studiu.

1.7 JQuery

JQuery je open source javascriptová knihovna s širokou podporou prohlížečů, která klade důraz na interakci mezi JavaScriptem a HTML. JQuery jsem využil na automatické obnovování stránky, byla to nejefektivnější metoda.

1.8 Tinkercad

Tinkercad je 3D modelovací program, který je dostupný ve webovém prohlížeči. Je to často používaný nástroj, pro vytváření modelů pro 3D tisk. Tento program jsem využil proto, protože jsem potřeboval vymodelovat nějakou “krabičku”, do které bych mohl dát své Esp zařízení. V této aplikaci modeluji už nějakou dobu, takže se v ní vyznám a umím ji používat

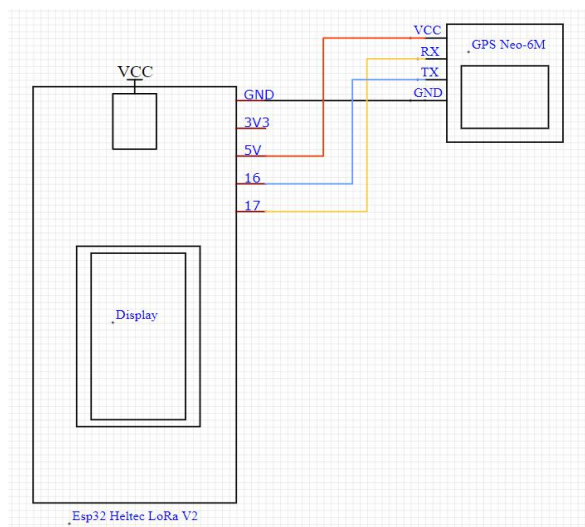
1.9 GPS

GPS (anglicky Global Positioning System) je globální družicový polohový systém. GPS umožňuje pomocí elektronického přijímače určit přesnou polohu na povrchu země. GPS jsem zvolil, protože je zdarma, je (skoro) všude, a hlavně poskytuje velké množství informací, nejen polohu. Na komunikaci s GPS jsem zvolil modul GPS-NEO-6M, což je přídatný modul pro Esp zařízení.

2 ZPŮSOBY ŘEŠENÍ A POUŽITÉ POSTUPY

2.1 Zapojení hardware

Pro můj projekt jsem použil 3 hardware zařízení: dvě “Heltec Esp32 LoRa V2” (Sender a Reciever) a jeden “GPS-NEO-6M” modul. První věc, kterou jsem musel udělat bylo zajistit správnou funkci těchto zařízení. Pro *Sender* jsem dle informací na internetu zapojil modul GPS (zapojení ve schématu). A obě zařízení poté pomocí micro USB kabelu připojil k počítači.



Obrázek 1: Schéma zapojení GPS do Esp32

Odkaz na použité zařízení k datu 28.12.2021:

- Heltec Esp 32 LoRa V2: <https://www.ebay.com/itm/173624285281>

- GPS-NEO-6M: <https://www.ebay.com/itm/174930619300>

2.2 Řešení LoRa komunikace

2.2.1 Seznámení s LoRou

Jako další přišlo na řadu zjistit, jak funguje LoRa a jak posílat určité informace z jednoho, na druhé zařízení. Celý princip komunikace mezi zařízeními je takový, že *Sender*, neboli odesílač začne tvořit packet, do kterého uloží informace, ukončí tvoření packetu a dále pořád do kola vysílá dané packety v LoRa pásmu.

Ukázka kódu ze souboru “/Sender/src/main.cpp” (vytváření packetu pro odeslání LoRa):

```
LoRa.beginPacket();  
//začíná tvořit jeden packet, který pak bude odeslán pomocí LoRa  
LoRa.print(gps.altitude.meters());  
//uloží do packetu informaci o nadmořské výšce  
LoRa.endPacket(); //ukončí a odešle packet
```


Reciever neboli přijímač má za úkol čekat a skenovat dané pásmo, dokud nezachytí packet vyslaný Senderem. Pokud se tak stane, tento packet uloží a je možné s ním provádět jiné operace.

Ukázka kódu ze souboru “/Reciever/src/main.cpp” (přijímání packetu LoRa):

```
if (velikostPaketu)
{ //pokud se nějaký packet přijme spustí se funkce
  while (LoRa.available())
  { //prevedeni textu do jednoho stringu
    VypisovanyText += (VyslednyText + strlen(VyslednyText), "%c",
(char)LoRa.read());
  }
}
```

2.2.2 Problém s pásmem

Problém, se kterým jsem se v této oblasti setkal, byl ten, že Esp zařízení komunikovaly pouze, pokud se jejich antény dotýkaly, což se nakonec vyřešilo změněním LoRa pásma, které bylo pro jiný model Esp.

Ukázka kódu ze sozboru “/Reciever/src/main.cpp” (vybrání pásma pro LoRa):

```
#define PASM0 434E6 //LoRa pasmo pro Evropu
void setup()
{
  Heltec.begin(false, true, true, true, PASM0 /*pasma pro Evropu*/);
}
```

2.3 Získávání GPS souřadnic

2.3.1 Princip kódu

Po vyřešení komunikace mezi zařízeními, přišlo na řadu zajistit informace, které by se odesílaly. Pro jednodušší programování, jsem vybral knihovnu TinyGps++.h, která jak spolehlivá, tak jednoduchá na používání. Celý GPS kód funguje tak, že zařízení skenuje prostor, ze kterého má GPS signál přijít a pokud je signál přijat, přeloží ho a dále je možné s ním pracovat. GPS signál dovoluje získávat zařízení, nejen polohu a nadmořskou výšku, ale také čas, datum a počet satelitů, čehož jsem v mém projektu také využil.

2.3.2 Přesná GPS data

Problém v této fázi projektu byl ten, že pozice, kterou jsem dostával, měla často odchylku až 1 km. Toto jsem však vyřešil posláním delší informace o poloze (z 50.09 se stalo 50.099999) což silně ovlivnilo určování přesné polohy.

Ukázka kódu ze souboru “/Sender/src/main.cpp” (získávání GPS souřadnic):

```
while (SerialGPS.available() > 0)
//pokud je GPS signál dostupný
{
    gps.encode(SerialGPS.read());
//přelož GPS signál
}
}
```

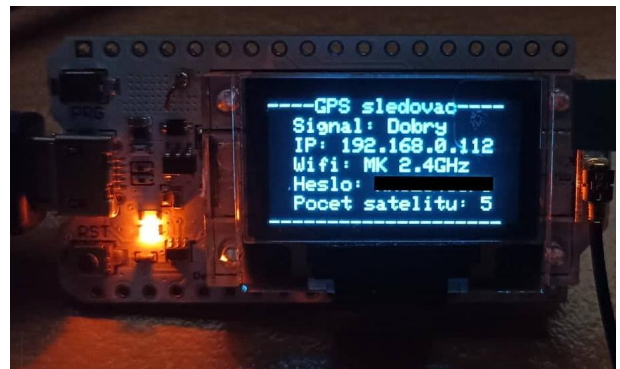
```
Pocet Satelitu: 0
Latitude: 4956.27N
Lognitude: 1754.43E
Den: 12
Hodina: 8
Pocet Satelitu: 0
Latitude: 4956.27N
Lognitude: 1754.43E
Den: 12
Hodina: 8
Pocet Satelitu: 7
Latitude: 4956.27N
Lognitude: 1754.43E
Den: 12
Hodina: 8
Pocet Satelitu: 7
Latitude: 4956.27N
Lognitude: 1754.43E
█
```

Obrázek 2: Informace získané z GPS

2.4 Webový server

2.4.1 Jak server funguje

Abych mohl zobrazovat data na webové stránce, bylo nutné vytvořit webový server, na kterém by stránka běžela. Jako první byla verze, která na samotném Esp zařízení vytvořila vlastní wifi, uživatel se na ní připojil a do vyhledávače zadal IP adresu, která ho přesměrovala na webovou stránku. Problém ale nastal, když jsem potřeboval využít služby jako mapy nebo Bootstrap, ke kterým byl nejjednodušší přístup přes internet, který Esp nemá. Proto jsem vybral druhou variantu, která zahrnuje připojení se na již vytvořenou wifi a vytvoření serveru na ní.



Obrázek 3: Výpis informací na display od Esp32

Ukázka kódu ze souboru “/Reciever/src/main.cpp” (zapínání webserveru a připojení na wifi):

```
AsyncWebServer server(80); //vytvoření serveru
void setup()
{
    WiFi.begin(ssid, password); //zapnutí wifi (připojení se na wifi pomocí údajů)
    while (WiFi.status() != WL_CONNECTED) //dokud se wifi nepřipojí
    {
        Serial.println("Connecting to WiFi.."); //vypiš connecting...
    }
}
```

2.4.2 Komunikace serveru s ESP

Při vytváření stránky se server zeptá na zdroje, které jsou na stránce použité, Esp mu pošle cestu uloženou v souborovém systému SPIFFS.

Ukázka kódu ze souboru “/Reciever/src/main.cpp” (odesílání dat na webserver):

```
server.on("/style.css", HTTP_GET, [] (AsyncWebServerRequest *request) {
    //cesta pro styly v .css souboru
    request->send(SPIFFS, "/style.css", "text/css");
});
```

Pokud se však web chce zeptat například na informaci o poloze, zeptá se na ukazatel GPS_LON (zeměpisná délka), funkce tento dotaz zpracuje a vydá mu správnou informaci.

Ukázka kódu ze souboru “/Reciever/src/main.cpp” (odesílání dat na webserver):

```
String processor(const String &var)
//slouží k odesílání informací na webový server
{
    if (var == "GPS_LON")
    { //pokud se webserver zeptá na informaci o GPS_LON, bude mu poslán
      VypisovanyText
      return (Parser(VypisovanyText2, 1));
    }
    else if (var == "GPS_LAT")
    {
        return (Parser(VypisovanyText2, 2));
    }
}
```

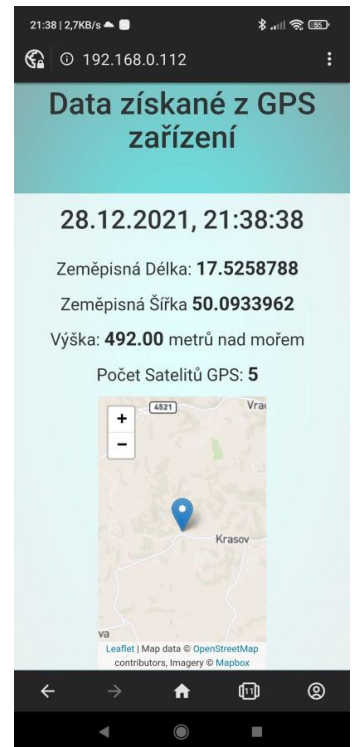
2.5 Klientská webová aplikace

2.5.1 Design aplikace

Poslední část projektu, byl design webu tak, aby byl jednoduchý na pochopení a využití. Základem je *index.html*, který společně se *style.css* tvoří to, co uživatel uvidí. Na začátek jsem vytvořil samotnou stránku, se získávanými informacemi a popisky, pomocí Bootstrap jsem zajistil responzivitu stránky a v souboru *style.css* využil třídy pro úpravu vzhledu stránky.

2.5.2 Zobrazení mapy

Dále jsem musel přijaté informace zobrazit, tak aby je nemusel uživatel manuálně zadávat do map (např. Mapy Google). Zaprvé jsem implementoval zdroje pro knihovnu *leaflet*, která dokázala vložit na stránku mapu a ukázat na ní polohu. Za druhé jsem z webu *mapbox.com*, využil pokládání “dlaždic” (mapa je rozdělena do jednotlivých dlaždic, které se pak podle potřeby a velikosti zvolené oblasti na mapu pokládají) a tím zobrazil mapu na stránce.



Obrázek 4: Ukázka webové stránky

Ukázka kódu ze souboru “/Receiver/data/script.js” (funkce pro vytvoření mapy a ukazatele polohy)

```
//získá souřadnice, které jsou již na stránce vypsané
let lon = document.getElementById('lon').innerText;
let lat = document.getElementById('lat').innerText;
//nastaví pohled mapy na zadané souřadnice
let map = L.map('map').setView([lat, lon], 12);
L.tileLayer('https://api.mapbox.com/styles/v1/{id}/tiles/{z}/{x}/{y}?', {
  maxZoom: 18, //nastaví zoom na 18
  id: 'mapbox/streets-v11',
  tileSize: 512, //velikost dlaždice
  zoomOffset: -1,
}).addTo(map); //přidá na mapu
let marker = L.marker([lat, lon]).addTo(map);
// vytvoří nový ukazatel podle získaných informací
```

2.5.3 Funkčnost stránky

Pro správné fungování stránky jsem pomocí JQuery vyřešil obnovování informací jak na mapě, tak v textu. Tyto informace se obnovují po uběhlém čase pomocí funkce *setInterval()*.

Ukázka kódu ze souboru “/Reciever/data/script.js” (obnovování textu a mapy):

```
function updateDiv() { // funkce pro obnovování textu a mapy
    //mapa
    lon = document.getElementById('lon').innerText; //získá souřadnice, které
    jsou již na stránce vypsáné
    lat = document.getElementById('lat').innerText; //získá souřadnice, které
    jsou již na stránce vypsáné
    map.removeLayer(marker); //odstraní ukazatel na mapě
    marker = L.marker([lat, lon]).addTo(map); // vytvoří nový ukazatek podle
    získaných informací
    //text
    $("#vypisovani").load(window.location.href + " #vypisovani"); //update textu
    s id vypisovani
}
setInterval(updateDiv(), 1000); //funkce se spustí každých 1000 milisekund
```

3 VÝSLEDKY ŘEŠENÍ A TESTY

3.1 Shrnutí funkcí zařízení

První zařízení *Sender*, dokáže získat informace z GPS satelitů a pomocí sítě LoRa je sdílí v předem určeném pásmu jako proud dat. Druhé zařízení na tyto data čeká, a pokud je přijme, rozloží je na jednotlivé části (poloha, čas, počet satelitů, ...). Když se na ně webový server zeptá, tak mu je pošle a zobrazí je na webové stránce. Podle pozice, kterou server dostal, vytvoří interaktivní mapu, po které se může uživatel volně posouvat, přibližovat a oddalovat. Stránka se podle intervalu obnoví a tím se obnoví i dostupné informace a mapu.

3.2 Testy funkcí a dosahu zařízení

- I. První test se konal v lese, terén byl převážně zarostlý křovím a stromy. LoRa zařízení se nedokázaly připojit, test byl **neúspěšný**. Odhadovaná vzdálenost byla asi 1500 metrů.
- II. Druhý test, se konal na okraji lesu, terén byl převážně prostupný, občasné stromy. Příjímač byl doma ve vesnici, musíme tedy brát v potaz různé typy elektromagnetického rušení a budovy. Test byl **úspěšný** s dosahem asi 900 metrů.
- III. Třetí test, se konal v obývané oblasti, v okolí byly budovy a ostatní zařízení, terén byl smíšený. Přesto byl test **úspěšný** s dosahem asi 1300 metrů.
- IV. Poslední test byl proveden na otevřeném poli, podmínky pro dosah byly nejlepší, co mohly být. Dosah byl 4000 metrů, se stále přijatelným rušením (zařízení by dokázalo komunikovat ještě na vyšší vzdálenost). Test byl **úspěšný**.

3.3 Porovnání s konkurencí

- a) Test dosahu - konkurenční zařízení byly lepší, s průměrným dosahem 15 km, mé zařízení bylo schopné dosáhnout okolo 4000 metrů.
- b) Test zobrazení - konkurenční zařízení mají stejně jako já možnost obrazit pozici psa na mapě. Některé zařízení mají navíc i kompas, který ukazuje směrem k pozici.
- c) Cena - mé zařízení bych ohodnotil (bez započítání práce) na asi 1 000 Kč. U konkurenčních zařízení cena začíná na 9 000 Kč a roste.

ZÁVĚR

Projekt v tuto chvíli obsahuje tři části. Hardware, který se skládá z GPS modulu a dvou Esp32 zařízení: jedno je u uživatele (Receiver), druhé na obojku psa (Sender). Software, který zajišťuje zpracovávání a posílání informací. Webová aplikace, na které se informace zobrazují. Zařízení umí získat polohu zařízení spolu s dalšími použitelnými informacemi (čas, počet satelitů, nadmořská výška...) a tyto informace poslat k uživateli. Na webu vypíše informace a vytvoří mapu, na které zobrazí polohu.

V průběhu práce mě napadly některé vylepšení, které by se v budoucnu daly uplatnit, aby se i mé zařízení mohlo rovnat konkurenci na trhu, například:

- kompas ukazující směr k zařízení,
- lepší zobrazení stránky,
- použít vlastní zdroj napájení,
- vylepšení hardwaru, pro lepší dosah signálu.

Využití mého zařízení bych viděl jak v oblasti sledování mazlíčků (včetně dětí), tak například při sledování vozidel či zavazadel.

Odkaz na Github s projektem:

https://github.com/NeViMjAkSeJmEnOvAt/rocnikovy_projekt_V2

SEZNAM INFORMAČNÍCH ZDROJŮ

- [1] NMEA 2000. Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001-, 3. October 2021 [cit. 2021-12-28]. Dostupné z: https://en.wikipedia.org/wiki/NMEA_2000
- [2] SPIFFS. Random Nerd Tutorials [online]. RandomNerdTutorials.com, 2021 [cit. 2021-12-28]. Dostupné z: <https://randomnerdtutorials.com/esp32-vs-code-platformio-spiffs/>
- [3] Webserver + SPIFFS. Random Nerd Tutorials [online]. RandomNerdTutorials.com, 2021 [cit. 2021-12-28]. Dostupné z: <https://randomnerdtutorials.com/esp32-web-server-spiffs-spi-flash-file-system/>
- [4] LoRa. Random Nerd Tutorials [online]. RandomNerdTutorials.com, 2021 [cit. 2021-12-28]. Dostupné z: <https://randomnerdtutorials.com/esp32-lora-rfm95-transceiver-arduino-ide/>
- [5] ESP32 access point. Random Nerd Tutorials [online]. RandomNerdTutorials.com, 2021 [cit. 2021-12-28]. Dostupné z: <https://randomnerdtutorials.com/esp32-access-point-ap-web-server/>
- [6] Heltec Library. Github: Where the world builds software [online]. Github.com, 2021 [cit. 2021-12-28]. Dostupné z: https://github.com/HelTecAutomation/Heltec_ESP32/tree/master/examples/LoRa
- [7] LoRa příklady. Github: Where the world builds software [online]. Github.com, 2021 [cit. 2021-12-28]. Dostupné z: <https://github.com/sandeepmistry/arduino-LoRa/blob/master/API.md>
- [8] GPS. Github: Where the world builds software [online]. Github.com, 2018 [cit. 2021-12-28]. Dostupné z: https://github.com/DzikuVx/esp32_gps_thingy/blob/master/gps_logger.ino
- [9] Wifi. Github: Where the world builds software [online]. Github.com, 2021 [cit. 2021-12-28]. Dostupné z: <https://gist.github.com/futechiot/ee0223dd269cbe7d8605ce97d120d7d2>

- [10] Webová stránka s mapou. Github: Where the world builds software [online]. Github.com, 2021 [cit. 2021-12-28]. Dostupné z: <https://github.com/superucitelka/SmartWatch-REST>
- [11] Partitions pro SPIFFS. Github: Where the world builds software [online]. Github.com, 2021 [cit. 2021-12-28]. Dostupné z: <https://github.com/espressif/arduino-esp32/tree/master/tools/partitions>
- [12] Obrázky. Pixabay [online]. Berliner Freiheit 2, 10785 Berlin: Pixabay.com [cit. 2021-12-28]. Dostupné z: <https://pixabay.com>
- [13] Partitions size. Stackoverflow [online]. Stackoverflow, 2016, 4.4.2016 [cit. 2021-12-28]. Dostupné z: <https://stackoverflow.com/questions/36413423/mapbox-offline-storage-size>
- [14] JQuery div update. Stackoverflow [online]. Stackoverflow, 2021, 19.11.2015 [cit. 2021-12-28]. Dostupné z: <https://stackoverflow.com/questions/33801650/how-do-i-refresh-a-div-content>
- [15] Map - delete markers. Stackoverflow [online]. Stackoverflow, 2021, 28.3.2012 [cit. 2021-12-28]. Dostupné z: <https://stackoverflow.com/questions/9912145/leaflet-how-to-find-existing-markers-and-delete-markers>
- [16] Mapbox. Mapbox: Maps and location for developers [online]. Mapbox.com, 2021 [cit. 2021-12-28]. Dostupné z: <https://www.mapbox.com>
- [17] Leaflet. Leaflet: <https://leafletjs.com/examples/quick-start/> [online]. Leaflet.com, 2021 [cit. 2021-12-28]. Dostupné z: <https://leafletjs.com/examples/quick-start/>
- [18] REST server. Renzo Mischianti: A blog of digital electronics and programming [online]. Renzo Mischianti, 2021, 16.4.2021 [cit. 2021-12-28]. Dostupné z: <https://www.mischianti.org/2020/05/16/how-to-create-a-rest-server-on-esp8266-and-esp32-startup-part-1/>
- [19] Adafruit Parsing. Adafruit [online]. Adafruit.com, 2021, 23. Aug 2012 [cit. 2021-12-28]. Dostupné z: <https://learn.adafruit.com/adafruit-ultimate-gps/parsed-data-output>