

STREDNÁ PRIEMYSELNÁ ŠKOLA ELEKTROTECHNICKÁ
PREŠOV

IV.C - 13

ŠK. R. 2017 – 2018

PČOZ – DATALOGGER PRE OUTDOOROVÉ AKTIVITY

Čestné vyhlásenie

Vyhlasujem, že celú prácu na tému „Datalogger pre outdoorové aktivity“ som vypracoval samostatne, s použitím uvedenej literatúry.

Prešov, 18. február 2018

.....

vlastnoručný podpis

Pod'akovanie

Týmto by som sa chcel poďakovať konzultantovi, pánovi Ing. Martinovi Ambrozemu za odbornú pomoc počas celej práce.

Obsah

Úvod	4
1 Cieľ práce	5
2 Metodika práce (Materiál a metodika práce)	6
2.1 Hardware	6
2.2 Software.....	8
3 Praktická časť.....	Chyba! Záložka nie je definovaná.
3.1 Návrh DPS a krabičky	10
3.2 Zdrojový kód	14
4 Závery práce	20
Anotácia v slovenskom jazyku	Chyba! Záložka nie je definovaná.
Anotácia v anglickom jazyku	Chyba! Záložka nie je definovaná.
Zoznam použitej literatúry	22

Úvod

V dnešnej modernej dobe sa často stretávame so snahou ľudí mať zmapovaný každý svoj krok pomocou najnovšej technológie, presne a aktuálne. K tomuto samozrejme takmer vždy postačuje skvelé zariadenie, ktoré má vo vrecku už snád' každý – smartfón. No čo ak existujú činnosti, pri ktorých je použitie takéhoto zariadenia nepraktické, hrozí jeho poškodenie alebo činnosť, ktorú chceme s týmto zariadením vykonávať znemožňuje slabý signál?

S takýmto problémom som sa stretol, keď som chcel pomocou mobilnej aplikácie zaznamenávať svoju cyklotúru po lese. Jednou z podmienok bezporuchového prijímania GPS signálu, ako vieme, je umiestnenie prijímača pod holé nebo. V takomto prípade fungovala aplikácia v mobile s miernou nepresnosťou, avšak pri zaznamenávaní trasy v lese signál vypadával a nepresnosť sa zvyšovala.

To ma motivovalo k vytvoreniu vlastného GPS lokátora s vyššou presnosťou ako rôzne aplikácie pre smartfóny a za oveľa nižšiu cenu, než za akú sú predávané profesionálne zariadenia zaznamenávajúce vašu polohu, napríklad pri jazde na bicykli.

Ďalšou výhodou môjho zariadenia je veľká výdrž zabezpečená batériou s kapacitou porovnateľnou s kapacitou batérií v mobilných telefónoch. Môj datalogger disponuje aj zobrazením aktuálnych informácií na displeji. Podobné zariadenia, ktoré sa cenovo najviac približujú môjmu zariadeniu žiaden displej nemajú a často potrebujú externý zdroj elektrickej energie, čo samozrejme zvyšuje náklady a taktiež rozmery.

V kapitole ciele práce je načrtnuté, aké funkcie by malo mať dané zariadenie. V kapitole 2 je popísaná konštrukcia a funkcie kódu, ktorý zabezpečuje fungovanie prístroja. Kapitola 3 sa venuje praktickým častiam projektu. Sú v nej všetky schémy a návrhy DPS, ktoré sú použité v danom výrobku.

1 Cieľ práce

Cieľom našej práce je navrhnuť univerzálne zariadenie na zaznamenávanie pohybu človeka vo vonkajšom prostredí, napríklad v prírode, pomocou GPS modulu s použitím vývojovej dosky Arduino, ktoré bude cenovo dostupnejšie oproti podobným konkurenčným zariadeniam. Zariadenie je určené pre športovcov, primárne pre cyklistov ako aj pre turistov, na zber dát o svojej vlastnej činnosti v prírode.

Na základe lokalizácie človeka má datalogger zaznamenávať GPS dáta a zapisovať ich na SD kartu, ktorú možno jednoducho zo zariadenia vybrať a s údajmi pracovať na počítači alebo mobilnom telefóne.

Následne je možné získane dáta vizualizovať pomocou počítačového programu Google Earth.

Aktuálne informácie o polohe človeka, čase a rýchlosti dokážeme zariadením zobrazovať na praktický LCD displej s rozmermi 16x2.

Čiastkovým cieľom tohto projektu bolo vyriešiť zabezpečenie stabilného napájania dataloggera aj počas dlhšieho pobytu človeka vo vonkajšom prostredí bez externého zdroja elektrickej energie.

Kritériom pre výrobu zariadenia bola vyššia presnosť oproti rôznym mobilným aplikáciám pracujúcim na obdobnom princípe.

2 Materiál a metodika práce

Prvotným krokom pre zostrojenie zariadenia – dataloggera bol výber jednotlivých častí, modulov pre vývojovú dosku Arduino na základe nami požadovaných funkcií zariadenia. Druhým krokom bol návrh krabičky, v ktorej bude DPS umiestnená. Posledný a dovoľm si tvrdiť najpodstatnejší krok bolo vytvorenie funkčného programu

2.1 Hardware

Obal zariadenia sme navrhli v programe Autodesk Inventor a následne vytlačili na 3D tlačiarňu. Toto riešenie je najvýhodnejšie z dôvodu nízkych nákladov na jeho výrobu, jednoduchosti, nízkej hmotnosti obalu a minimalizácie odpadového materiálu.

Rozmery zariadenia sú 145x75x25 mm. Pre porovnanie s mobilným telefónom je zariadenie hrubšie iba približne o 15 mm, avšak je oveľa presnejšie. Vďaka týmto rozmerom je zariadenie praktické, ľahké. Vzhľadom k jeho umiestneniu na riadidlách bicykla je odčítavanie údajov z displeja pre človeka jednoduché aj počas jazdy na bicykli. Odčítavanie za zhoršených svetelných podmienok je zabezpečené podsvietením displeja. Odosielanie údajov na displej sa zjednodušilo použitím prevodníka, ktorý využíva na prijímanie a odosielanie dát I2C zbernicu.

Keďže v prírode človek nemá k dispozícii zdroj elektrickej energie, na napájanie sme boli nútení použiť batériu. Vzhľadom k tomu, že sme nepotrebovali odoberať vysoký elektrický prúd, volili sme cenovo dostupnú jednočlánkovú Li-Pol batériu s kapacitou 2200 mAh. Tá poskytuje nášmu zariadeniu vysokú výdrž. Na ochranu pred podbitím batérie pod 2,4 V, ochranu pred prebitím nad 4,2 V, ochranu pred rýchlym vybíjaním a skratom (pri vybíjacom prúde väčšom ako 3 A) sme použili nabíjačku s obvodom TP4056 a ochrannou elektronikou DW01A.

Na zaznamenávanie polohy sme potrebovali GPS modul s dostatočne veľkou anténou, aby dokázala zachytiť signál zo satelitov aj v lese, preto sme zvolili modul GY-NEO6MV2 s externou anténou. Odosielanie údajov do Arduina je zabezpečená sériovou komunikáciou cez piny RX a TX.

Použitý adaptér na zápis údajov na microSD kartu, ktorý má názov Catalex v1.0 je umiestnený tak, aby sa pamäťová karta dala z dataloggera odstrániť jednoducho a bez nutnosti rozoberať obal celého zariadenia. S Arduinom komunikuje pomocou SPI pinov.

Najdôležitejšou časťou celého prístroja je riadiaca časť, zabezpečená už spomínanou doskou Arduino. Pri výbere dosky sme sa riadili hlavne množstvom vstupov, možnosťou integrovaného programátora, veľkosťou pamäte.

Najvýhodnejším modelom pre tento projekt bolo Arduino Micro pro s 32 kB pamäťou, 16 MHz procesorom, vstupom micro USB-B pre napájanie a/alebo programovanie a 12 digitálnymi vstupmi/výstupmi. S rozmermi 36x18 mm a vyššie vymenovanými parametrami akurát vyhovoval pre aplikáciu v tomto zariadení.

2.2 Software

Program bol písaný v počítačovej aplikácii Arduino IDE. Pre zabezpečenie spolupráce modulov a riadiacej časti sme hneď na začiatku programu pridali knižnice potrebné na fungovanie celého kódu

V tejto časti sme narazili na problém, pretože príklady použité z knižníc boli robené tak, aby boli získané hodnoty z GPS modulu vypisované pomocou softvéru Arduino IDE v Serial Monitor-e.

Problém spočíval v nedostatku pamäte Arduina, a preto boli knižnice následne upravované tak, aby kód zaberal čo najmenej miesta v pamäti tejto vývojovej dosky.

Vyriešením problému sme ďalej pokračovali v editácii kódu. Po pridaní knižníc sme deklarovali premenné, vo funkcii setup nastavili komunikáciu modulov a následne sa začal nepretržite vykonávať kód vo funkcii loop.

Funkcia loop sa vykonáva s periódou 1 sekunda. Je to prispôbené GPS modulu, aby načítaval novú polohu za danú jednotku času (1 s). Aby sme dosiahli požadovaný čas, museli sme do kódu vložiť takzvaný smart delay (inteligentné oneskorenie). Princíp fungovania rovnomennej funkcie smartdelay() spočíva v tom, že počíta čas trvania funkcie v milisekundách. V prípade, že čas od spustenia funkcie je menší ako 1000 ms, načíta nové informácie z GPS modulu. Keď čas dosiahne hodnotu 1000 ms, funkcia sa ukončí a program pokračuje ďalej.

Po spustení programu môžu nastať viaceré situácie: GPS modul získava signál z menej ako štyroch satelitov, GPS modul má dobrý signál (4 a viac satelitov), pamäťová karta je vložená a pracuje správne alebo pamäťová karta vložená nie je, prípadne pracuje nesprávne.

Ak má GPS modul slabý signál, na displeji sa zobrazí chybová hláška „WEAK GPS SIGNAL, SATELITES: (počet satelitov, s ktorými komunikuje)“ a zariadenie nezapisuje na SD kartu žiadne údaje.

Ak je signál dostatočný na určenie polohy, dátum, čas a GPS súradnice sa automaticky ukladajú na SD kartu vo formáte .csv, čo je súbor programu Microsoft Excel, z ktorého sa následne dajú získané dáta o polohe vyexportovať do aplikácie Google Earth, v ktorej ich jednoducho vizualizujeme. Namerané dáta sa ukladajú do súboru s názvom rovnakým, ako je dátum zápisu. Je to výhoda pre používateľa, pretože vie rýchlo zistiť, v ktorý deň dané dáta zozbieral.

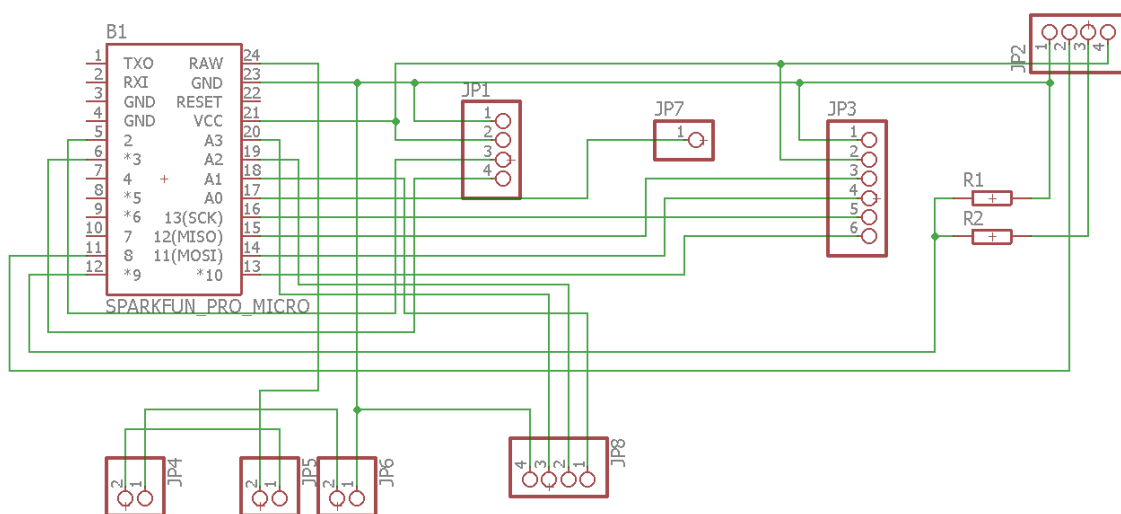
V prípade, že informácie o polohe získame, ale SD karta chýba alebo nepracuje správne, na displeji sa striedavo zobrazuje chybová hláška „CARD FAILURE“ a aktuálny čas.

Rôzne možnosti môžeme dosiahnuť aj pomocou troch tlačidiel. Tieto tlačidlá slúžia používateľovi na zapnutie alebo vypnutie podsvietenia displeja, spustenie alebo pozastavenie nahrávania prejdenej vzdialenosti a zápisu na SD kartu a posledné slúži na zresetovanie prejdenej vzdialenosti a úplné vypnutie nahrávania. Tlačidlá je nutné držať stlačené po dobu aspoň 1s.

3 Praktická časť

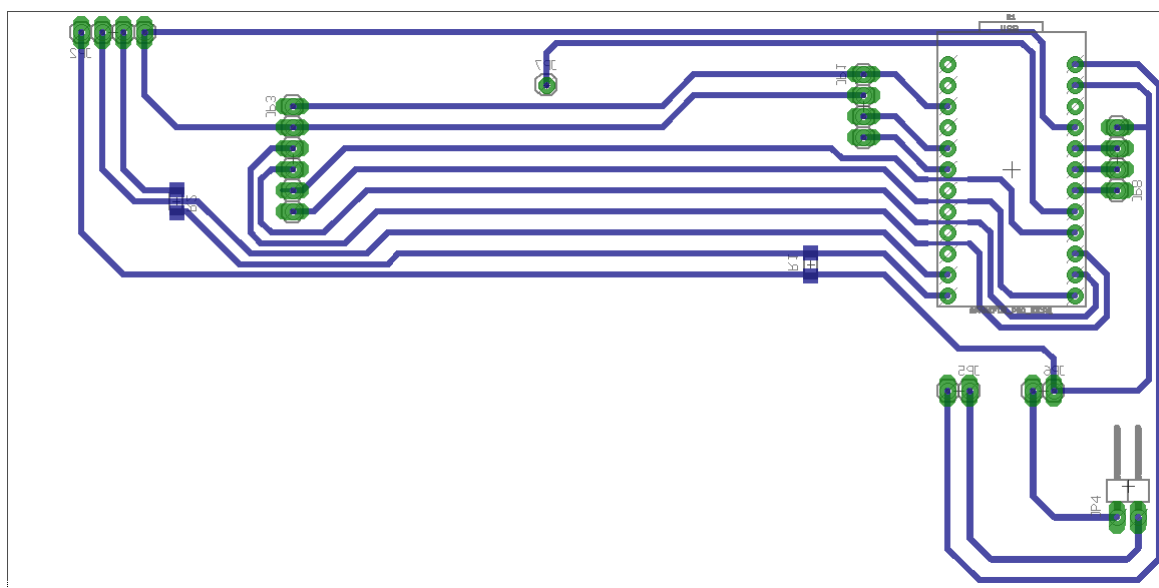
V tejto kapitole sú všetky časti zariadenia, schémy, návrhy plošných spojov, návrhy častí krabičky a zdrojový kód s komentármi k jednotlivým častiam kódu.

3.1 Návrhy DPS a krabičky

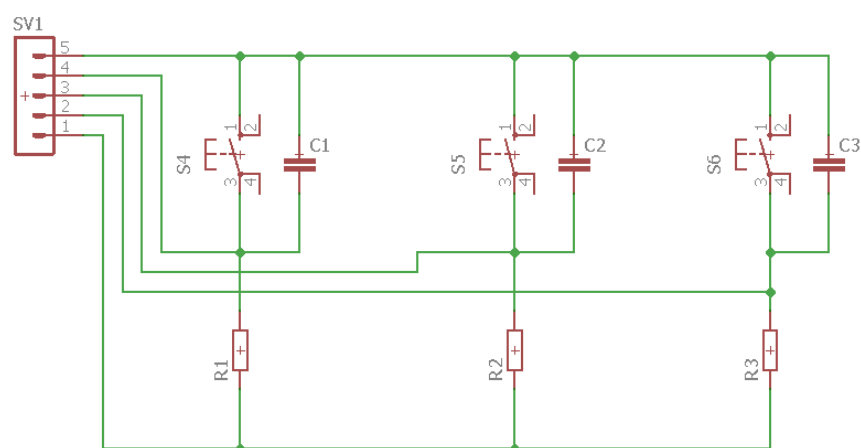


Obr. 1 Schéma zapojenia základnej DPS

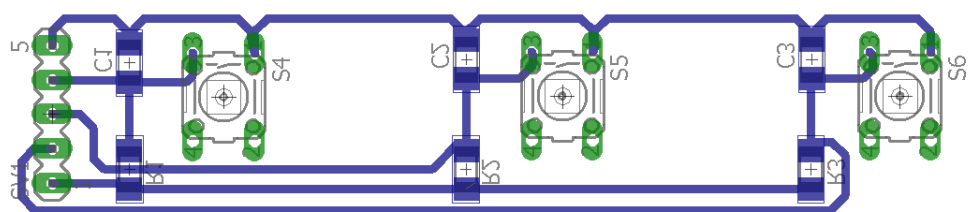
Step up menic



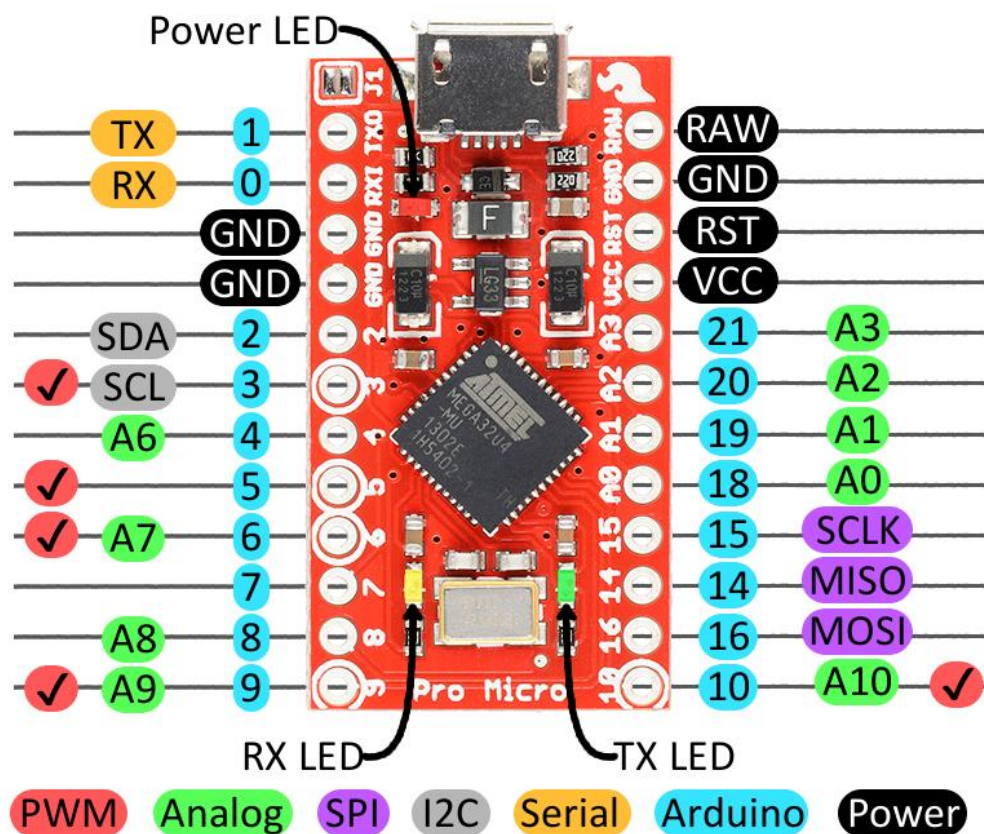
Obr. 2 Návrh základnej DPS



Obr. 3 Schéma zapojenia pre DPS s tlačidlami

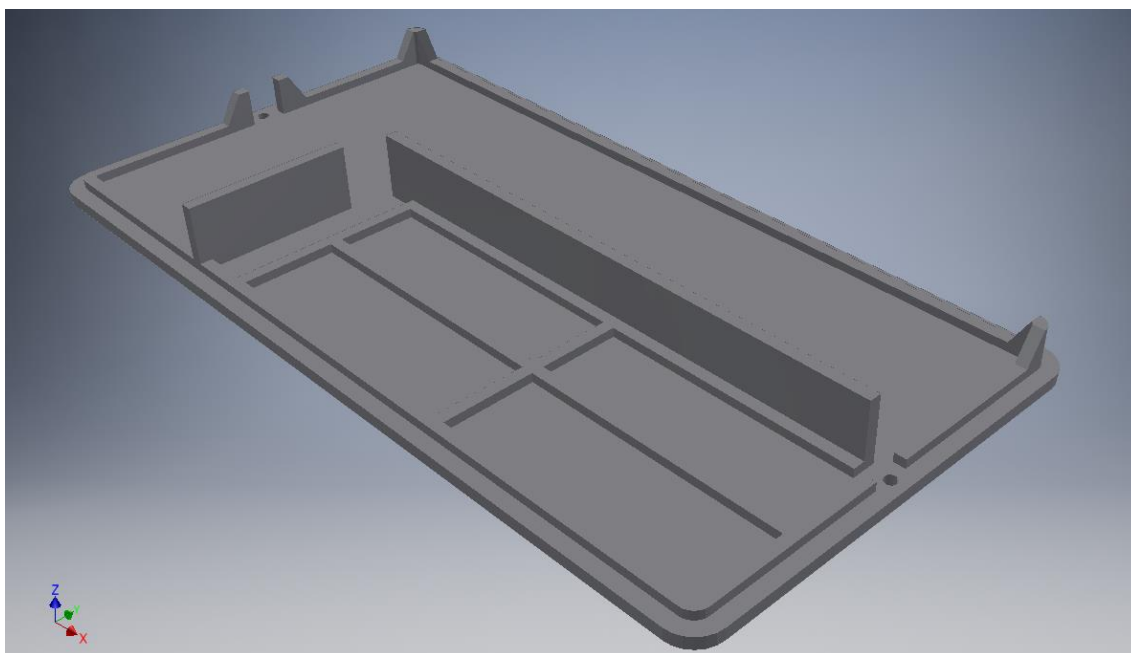


Obr. 4 Návrh pre DPS s tlačidlami

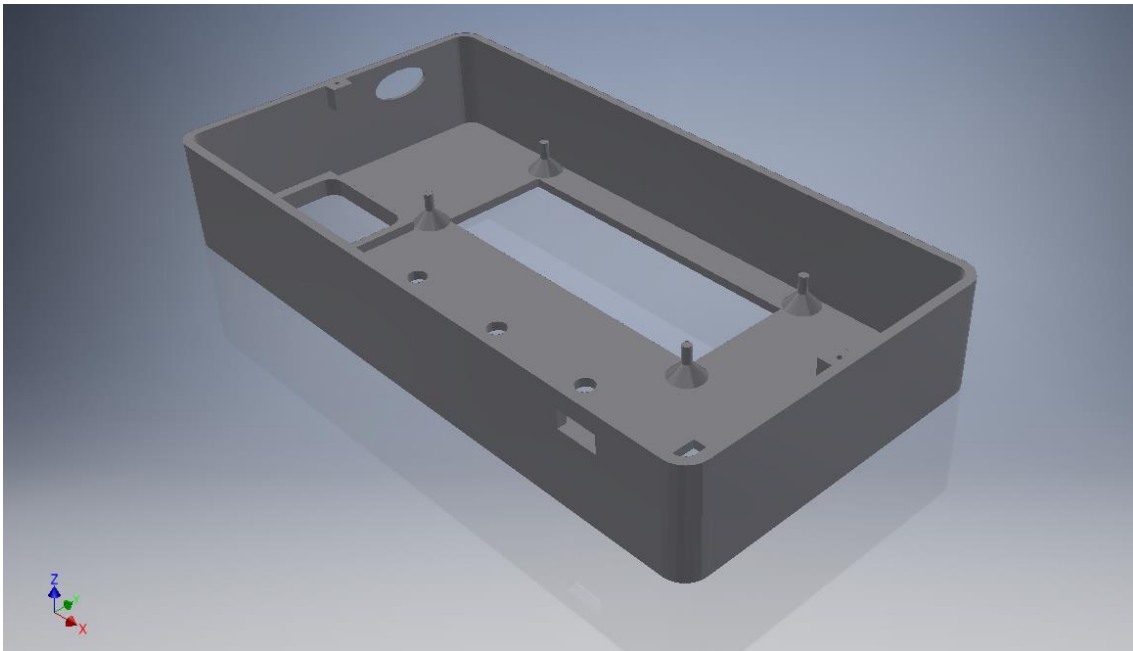


Obr. 5 Rozloženie pinov Arduina Micro Pro

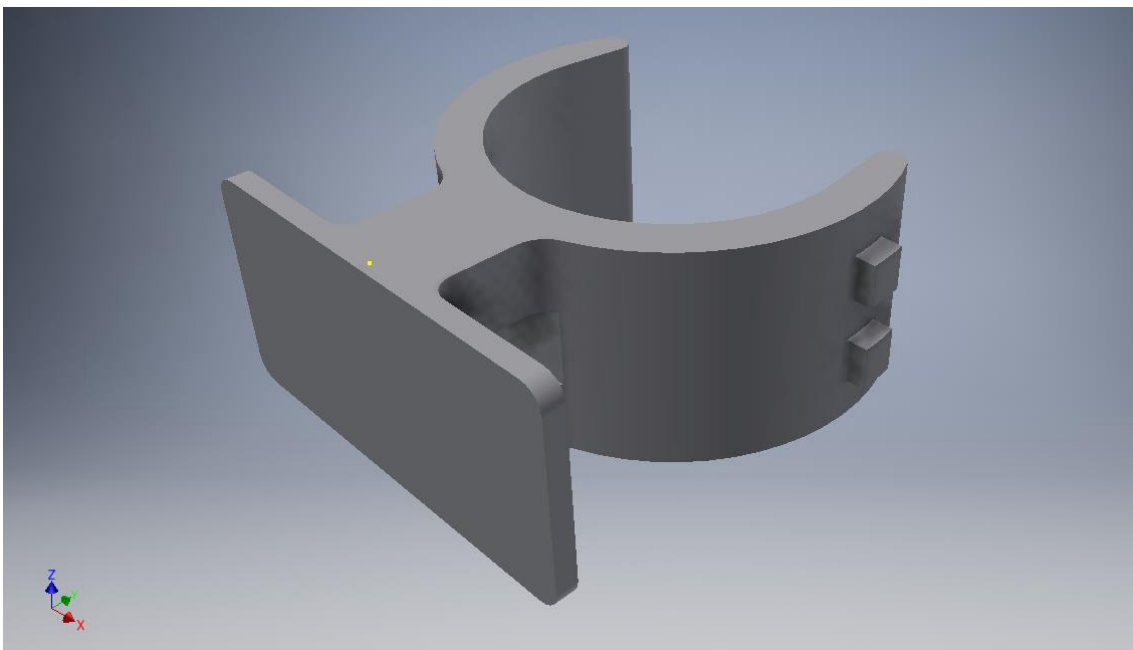
(Zdroj: <https://learn.sparkfun.com/tutorials/pro-micro--fio-v3-hookup-guide/hardware-overview-pro-micro>)



Obr. 6 Krabička – spodný diel



Obr. 7 Krabíčka – vrchný diel



Obr. 8 Držiak na krabíčku



Obr. 9 Fotografia hotovej krabičky

3.2 Zdrojový kód

```
#include <LiquidCrystal_I2C.h> // F Malpartida's NewLiquidCrystal library
#define I2C_ADDR 0x3F // Define I2C Address for controller
#define En_pin 2
#define Rw_pin 1
#define Rs_pin 0
#define D4_pin 4
#define D5_pin 5
#define D6_pin 6
#define D7_pin 7
#define BACKLIGHT 3
LiquidCrystal_I2C lcd(I2C_ADDR, En_pin, Rw_pin, Rs_pin, D4_pin, D5_pin, D6_pin,
D7_pin);
#include <Wire.h>

#include <SoftwareSerial.h>
#include <TinyGPS.h>
TinyGPS gps;
SoftwareSerial ss(8, 9);
static void smartdelay(unsigned long ms);

#include <SPI.h>
#include <SD.h>
#include <stdlib.h>
int CS = 10;
String dataString = "invalid";
String SD_lat = "invalid";
String SD_lon = "invalid";

int inc = 0;

int year, dist, prevdist;
```

```

byte month, day, hour, minute, second, hundredths;
float flat, flon, trackd = 0, prevlat = 0, prevlon = 0;
unsigned long age, date, time;
static const double LONDON_LAT = 51.508131, LONDON_LON = 0.000000;
char sats[32];
char realtime[16];
char filename[16];
char SD_date_time[16];

```

```

File file;

```

```

const int button1 = 21;
const int button2 = 20;
const int button3 = 19;
const int ledPin = 18;

```

```

int backLight = LOW;
int start_stop = 0;
int reset = 0;
int buttonState1 = LOW;
int buttonState2 = LOW;
int buttonState3 = LOW;
int lastButtonState1 = LOW;
int lastButtonState2 = LOW;
int lastButtonState3 = LOW;

```

```

//-----
-----

```

```

void setup()
{
  pinMode(CS, OUTPUT);

  ss.begin(9600); //setting communication with GPS

```

```

Serial.begin(9600);
SD.begin(10); //setting communication with SD card

lcd.begin (16, 2); //setting communication with LCD
lcd.setBacklightPin(BACKLIGHT, POSITIVE); //setting backlight pin as positive
lcd.setBacklight(HIGH);

pinMode(button1, INPUT);
pinMode(button2, INPUT);
pinMode(button3, INPUT);
pinMode(ledPin, OUTPUT);

digitalWrite(ledPin, backLight); //turning the backlight off on the start
}

//-----
-----

void loop()
{
  int reading1 = digitalRead(button1); //reading from buttons
  int reading2 = digitalRead(button2);
  int reading3 = digitalRead(button3);

  if (reading1 != buttonState1) {
    buttonState1 = reading1;

    if (buttonState1 == HIGH) {
      backLight = !backLight;
    }
  }
  if (reading2 != buttonState2) {
    buttonState2 = reading2;

```

```

    if (buttonState2 == HIGH) {
        start_stop = !start_stop;
    }
}

if (reading3 == HIGH) {
    start_stop = LOW;
    trackd = 0;
    prevdist = 0;
}

digitalWrite(ledPin, backLight);

gps.crack_datetime(&year, &month, &day, &hour, &minute, &second, &hundredths,
&age); //getting time and position from GPS
gps.f_get_position(&flat, &flon, &age);

if (gps.satellites() == TinyGPS::GPS_INVALID_SATELLITES || gps.satellites() < 4) {
//finding how many active satellites
    lcd.clear();
    lcd.home();
    lcd.setCursor(0, 0);
    lcd.print("WEAK GPS SIGNAL");
    lcd.setCursor(0, 1);
    lcd.print("SATELLITES: ");
    if (gps.satellites() == TinyGPS::GPS_INVALID_SATELLITES) {
        lcd.print("0");
    }
    else if ((0 < gps.satellites() < 4) && (gps.satellites() !=
TinyGPS::GPS_INVALID_SATELLITES)) {
        lcd.print(gps.satellites());
    }
}
else {
    lcd.clear(); //writin on the LCD actual time, speed, traveled distance

```

```
lcd.home();  
lcd.setCursor(0, 0);  
sprintf(realtime, "%02d:%02d:%02d ", hour + 1, minute, second);  
lcd.print(realtime);
```

```
lcd.setCursor(0, 1);  
lcd.print(gps.f_speed_kmph(), 2);
```

```
lcd.setCursor(5, 1);  
lcd.print("km/h");
```

```
if (start_stop == 1) { //starting or stopping recording  
    lcd.setCursor(10, 1);  
    if (0 <= trackd < 9) {  
        lcd.print(trackd);  
        lcd.setCursor(15, 1);  
        lcd.print("m");  
    }  
    else if (10 <= trackd < 99) {  
        lcd.print(trackd);  
        lcd.setCursor(15, 1);  
        lcd.print("m");  
    }  
    else if (100 <= trackd < 999) {  
        lcd.print(trackd);  
        lcd.setCursor(15, 1);  
        lcd.print("m");  
    }  
    else if (1000 <= trackd < 9999) {  
        lcd.print(trackd / 1000);  
        lcd.setCursor(15, 1);  
        lcd.print("km");  
    }  
    else if (10000 <= trackd < 99999) {
```

```

    lcd.print(trackd / 1000);
    lcd.setCursor(15, 1);
    lcd.print("km");
}

file.close();

sprintf(filename, "%02d%02d%02d.csv", day, month, year); //giving a file the same
name like an actual date

file = SD.open(filename, FILE_WRITE); //creating the file

if (file) {
    sprintf(SD_date_time, "%02d:%02d:%02d", hour + 1, minute, second); //writing the
actual time and position on the card
    file.print(SD_date_time); file.print(",");
    file.print(flat); file.print(",");
    file.println(flon);
}
else {
    if(inc%2==0){
        lcd.clear();
        lcd.home();
        lcd.setCursor(0, 0);
        lcd.print("CARD FAILURE");
    }
}
file.close();
}

dist = TinyGPS::distance_between(flat, flon, prevlat, prevlon);

if (inc <= 1) {
    trackd = trackd - trackd; //reseting the traveled distance at the beginning
    prevdist = 0; //reseting the previous distance at the beginning
}

```

```

    if (inc > 1 && dist) {
        trackd = prevdist + dist; //actual value of trav. dist. is increasing every repeating of loop
        prevdist = trackd; //setting the value of actual traveled distance as previous distance
    }

    prevlat = flat;
    prevlon = flon;

    inc++;
}
reading1 = 0;
reading2 = 0;
reading3 = 0;
smartdelay(1000); //function smartdelay, setting the time of one loop cycle on 1 second
}

static void smartdelay(unsigned long ms) //function smardelay used from TinyGPS
library's example "test_with_gps_device"
{
    unsigned long start = millis();
    do
    {
        while (ss.available())
            gps.encode(ss.read());
    } while (millis() - start < ms);
}

```

4 Závěry práce

Vzhľadom k tomu, že veľa ľudí v dnešnej dobe obľubuje aktivity v prírode a chcú mať o nich aj prehľad, datalogger je preto veľmi užitočným zariadením. Datalogger získava informácie nepretržite, sú aktuálne.

Za ciele našej práce sme si stanovili vysokú presnosť, ľahké upevnenie, nízku hmotnosť, malé rozmery, vysokú výdrž batérie a čo je pre ľudí jedným z najdôležitejších parametrov – nízku cenu. Všetky tieto ciele sa nám podarilo úspešne splniť.

Pri zhotovovaní výrobku sme sa stretli s niekoľkými problémami, najčastejšie to boli komplikácie v oblasti softvéru, ktoré sme vyriešili pomerne jednoducho úpravou zdrojového kódu alebo použitím iných knižníc pre vývojovú dosku Arduino. Komplikácie v oblasti hardvéru vznikli pri dizajnovaní krabičky. Požiadavkou bolo, aby nemala príliš veľké rozmery, ale zas aby sa do nej vošli všetky moduly potrebné pre kompletnú činnosť zariadenia. Zároveň sme potrebovali umiestniť GPS modul s anténou vo vrchnej časti pre lepší príjem signálu. Riešením bolo navrhnuť výrez vo vrchnej časti krabičky a umiestniť doňho GPS anténu. Ďalším hardvérovým problémom bolo výrazné obmedzenie veľkosti zdrojového kódu veľkosťou pamäti Arduina. Súčasný kód zaberá 92% pamäte Arduina. Veľkosť zdrojového kódu sme zredukovali použitím alternatívnych knižníc s rovnakou funkcionalitou.

Na ovládanie zariadenia slúži jeden prepínač na odpojenie napájania a tri tlačidlá s funkciami na ovládanie podsvietenia displeja a nahrávania trasy.

Možnosť vylepšenia do budúcnosti vidím v navrhnutí a vytvorení vlastnej mobilnej aplikácie pre vizualizáciu nameraných dát a zámenu LCD za grafický OLED displej.

Anotácia v slovenskom jazyku

V práci „Datalogger pre outdoorové aktivity“ bolo primárnym cieľom získať presné, aktuálne informácie o lokalizácii osoby v prírode a zaznamenávať ich na microSD kartu. GPS prijímač dosahoval oveľa lepšie výsledky, ako mobilný telefón, bol spoľahlivejší, presnejší a rýchlejší. V úvode sú popísané dôvody, prečo sme uvažovali nad vytvorením práve takejto práce, kapitola ciele popisuje očakávané výsledky, v časti 2 sa nachádzajú hardvérové špecifikácie. Táto kapitola taktiež rozoberá softvérové funkcie. Čitateľ sa môže dozvedieť o presnom označení modulov, ktoré boli použité, o ich zapojení, funkcii v kompletnom výrobku. Následne v kapitole 3 Praktická časť sú pre čitateľa prístupné schémy, návrhy DPS, 3D modely krabičky a celý zdrojový kód s komentármi v anglickom jazyku.

Anotácia v anglickom jazyku

In this project „Datalogger for outdoor activities“ was the primary goal to get accurate, actual information of the location of the person in nature and record them on the microSD card. The GPS receiver had much better results than the mobile phone, it was more reliable, more accurate and faster. In the Introduction are described the reasons why we have thought about the creation of this work, the chapter Objectives describes the expected results, the part 2 contains the hardware specifications. This chapter also discusses about software features. The reader can learn about the precise designation of modules that have been used, their connection, function in the complete product. Then, in Chapter 3, practical part has the reader access to the schemes, PCB design, 3D models of case and full source code with comments in English.

Zoznam použitej literatúry

Knižnica pre GPS: <https://github.com/mikalhart/TinyGPS/releases/tag/v13>

Informácie o Aduine Micro Pro: <https://learn.sparkfun.com/tutorials/pro-micro--fio-v3-hookup-guide/hardware-overview-pro-micro>

Vizualizácia: [Tutorial 15 for Arduino: GPS Tracking | JeremyBlum.com](https://www.jeremyblum.com/tutorial/15-arduino-gps-tracking/)

Knižnica Arduina pre CAD program Eagle: <https://github.com/sparkfun/SparkFun-Eagle-Libraries>

Zapojenie modulu Catalex: <https://forum.arduino.cc/index.php?topic=240815.0>