Stredná priemyselná škola informačných technológií Ignáca Gessaya

Technické lýceum

Albertov odkaz

Tvrdošín 2022 Matej Hušla

Stredná priemyselná škola informačných technológií Ignáca Gessaya

Technické lýceum

Albertov odkaz

Študijný odbor: 3918 M technické lýceum

Trieda: IV. C

Konzultant práce: Juraj Kasan

Tvrdošín 2022 Matej Hušla

Čestné vyhlásenie

Vyhlasujem, že som zadanú prácu vypracoval samostatne, pod odborným vedením vedúceho práce Juraja Kasana a používal som len literatúru uvedenú v práci.

Súhlasím s tým, že práca bude uložená v školskej knižnici a môže byť zapožičaná záujemcom o jej preštudovanie.

Tvrdošín, 21. 3. 2022

podpis: .....

Obsah

Zoznam skratiek

Skratka	Anglický význam	Slovenský význam
MEMS	Micro-electro-mechanical systems	Mikro-elektro-mechanické systémy
UART	Universal asynchronous Receiver-Transmitter	Univerzálny asynchrónny príjimač-odosielač
GSM	Global System for Mobile communication	Globálny systém mobilných komunikácií
GPRS	General Packet Radio Service	Univerzálna paketová rádiová služba
USB	Universal Serial Bus	Univerzálna sériová zbernica
GND	Ground	zem/katóda
LED	Light Emitting Diode	Luminiscenčná dióda
Li-po	Lithium polymer	Lítiovo polymérová
IoT	Internet of Things	Internet vecí
AMS	Advance Monolithic System	Pokročilý monolitcký systém
MPU	Micro Processor Unit	Mikroprocesorová jednotka

Zoznam symbolov

Symbol	Jednotka	Význam symbolu
V	[Volt]	Elektrické napätie
A	[Ampér]	Elektrický prúd
°C	[Celzius]	Teplota
s	[Sekunda]	Čas
Hz	[Hertz]	Frekvencia
B	[Byte]	Binárna informácia
C	[-]	Relatívny násobok nominálneho prúdu akumulátora

Úvod

Z názvu možno vydedukovať hlavnú motiváciu pre náš projekt. Konkrétne je to odkaz Alberta Eisteina, ktorý údajne povedal, že ak zahynú všetky včely tak s nimi zahynie aj ľudstvo. Nie je to úplne také jednoduché, ale podstata ostáva. Našim projektom chceme pomôcť včelárom pri monitorovaní základných veličín ako teplota a vlhkosť vzduchu. Tieto veličiny sa budú sledovať pre každý úľ osobitne. Jednotlivé úľové modely budú prepojené pomocou decentralizovanej siete, pričom údaje sa budú ukladať do databázy a prehľad údajov bude možné sledovať prostredníctvom web stránky. Jednotlivé modely taktiež budú obsahovať gyroskop/akcelerometer, ktorého úlohou je signalizovať včelára o prípadnom prevrátení úľa. Napájanie je realizované prostredníctvom nabíjateľného akumulátora pričom samotné nabíjanie akumulátora je sprostredkované pomocou fotovoltaického článku.

Naše riešene nie je úplne prvý prípad kedy by včelárstvo využívalo moderné technológie. Ale aj tak môžeme byť prínosom. V dávnejšej minulosti úplne absentovala takáto možnosť a včelár nemal dostatok údajov, vďaka ktorým by vedel lepšie optimalizovať chov včiel. Taktiež v prípade narušenia úľu by sa to dozvedel až vtedy ak by fyzicky prišiel na miesto. Pričom dnes by včelár vedel s pomocou moderných IoT zariadený vedieť monitorovať stav svojich úľov. Čiže v prípade prevrátenia úľa by sa to vedel dozvedieť rýchlejšie a tým pádom by mohol aj rýchlejšie konať. Moderné technológie formujú dnes už každé odvetvie. A robia to aj plným právom, pretože oproti minulosti máme prístup k omnoho väčším dátam za rýchlejší čas. Automatizáciou si vieme zefektívniť fungovanie prevádzok. Preto si vďaka týmto benefitom myslíme, že môžeme spolu pokračovať aj s inými iniciatívami, aj pri modernizácií včelárskeho odvetvia.

Našou úlohou bolo skonštruovať moduly, ktoré budú obsahovať teplomer, vlhkomer a gyroskop. Následne jednotlivé moduly majú byť prepojené prostredníctvom mesh siete. Jeden modul má obsahovať GSM celok, ktorý bude odosielať dáta do databázy prostredníctvom Node red. Následne majú byť dáta zobrazené na web stránke, ktorá ich bude zobrazovať aj grafickou formou. My sme k úlohe pridali realizovanie nabíjania akumulátora prostredníctvom fotovoltaického článku. Našou úlohou bolo taktiež vybrať si podľa vlastného uváženia konkrétne komponenty k splneniu nášho cieľa.

1 Súhrn vybraných komponentov

Komponenty sme vyberali na základe požiadavky konzultanta, ale aj na základe vlastného uváženia. Našou úlohou bolo vybrať konkrétne teplomer, vlhkometer, gyroskop a GSM modul. Našim vlastným rozhodnutím nasledovne bolo, že modul bude napájaný prostredníctvom fotovoltaického článku. K tomu aby náš modul mohol fungovať, bolo potrebné vybrať riadiacu jednotku, ktorá by spracúvala a posielala dáta zo senzorov. V našom prípade sme zvolili bežne dostupný mikrokontroler. V tejto kapitole sa budeme venovať osobitne každému komponentu, ktorý bol použitý pri skladaní modulu. Jednou z tém bude aj zdôvodnenie výberu daných komponentov.

1. 1 Riadiaca časť modulu

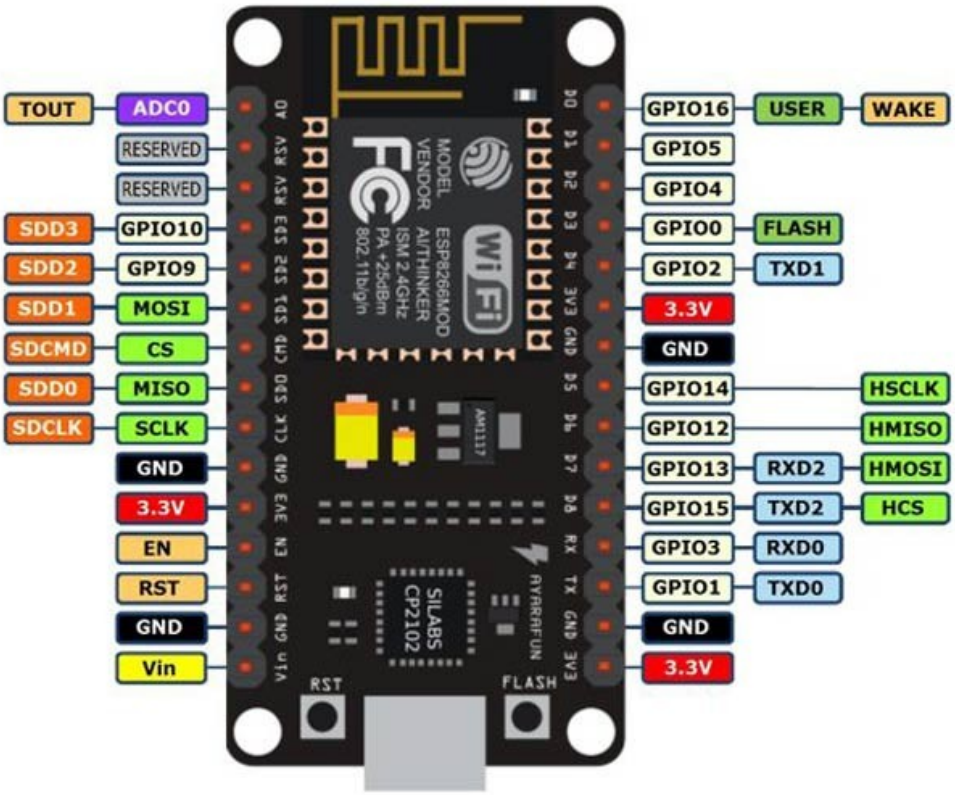
V tejto podkapitole sa budeme venovať kategorizovaniu a zdôvodneniu výberu komponentov, ktoré boli použité pri skladaní riadiacej časti nášho modulu. Pod pojmom riadiaca časť modulu je myslená taká časť modulu, ktorá sa priamo podieľa pri spracúvaní, meraní alebo odosielaní monitorovaných veličín.

1. 1. 1 Riadiaca jednotka

Našou úlohou je spracúvanie a posielanie dát, ktoré avšak nie sú vo veľkom množstve, preto sme zvolili menej výkonný variant mikrokontroleru. Konkrétne sa jedná o prevedenie mikrokontroleru IoT ESP8266 Lua NodeMCU V3.

Vzhľadom na to, že náš finálny modul je napájaný prostredníctvom batérie, je potrebné voliť efektívnu elektroniku. Čip ESP8266 pracuje s 3.3V logikou, čo pre nás znamená nižšiu spotrebu elektrickej energie, čiže je to pre nás výhoda oproti napr. modulom Arduino, ktoré pracujú na 5V logike. Oproti konkurencii taktiež vyšiel modul na báze ESP8266 v porovnaní ceny a kvality ako lepší variant. Výhodou riadiacej jednotky je ešte aj fakt, že disponuje integrovaným WiFi modulom, ktorý je pre nás kľúčovým na vytvorenie siete a odosielanie dát medzi jednotlivými modulmi.

Rýchlosť mikroprocesoru je 80 MHz. Ďalej modul disponuje flash pamäťou o kapacite 4MB. Obsahuje 16 vstupno-výstupných digitálnych a jeden analógový pin. Program sa do riadiacej jednotky nahráva prostredníctvom sériového rozhrania, ktoré je vytvorené USB káblom. Modul Lua NodeMCU V3 obsahuje práve jeden mikro USB port. Tento port môže slúžiť aj na napájanie, lenže v našom prípade volíme napájanie prostredníctvom pinov GND a 3V3.



Obr. 1. 1. 1: pinout modelu Lua NodeMCU V3
(Zdroj:https://www.root.cz/clanky/nodemcu-a-jeho-verzie-doska-s-wi-fi-cipom-esp8266/)

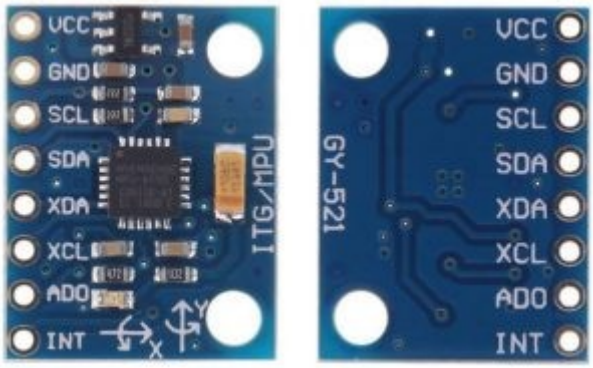
1. 1. 2 Teplomer a senzor vlhkosti

Z bežne dostupných teplomerov existuje sada DHT od spoločnosti Adafruit, ktorá kombinuje meranie teploty a zároveň vlhkosti vzduchu. Táto skutočnosť je celkom prijateľná, pretože nemusíme riešiť nákup a výber dvoch osobitných senzorov. Sada DHT je vhodná do projektu aj v rámci nenáročnosti zapojenia, pričom existuje aj podpora rôznych knižníc, ktoré sú priamo určené pre dané typy senzorov. My sme zvolili teplomer a vlhkomer DHT22, ktorý oproti verzii DHT11 disponuje presnejším meraním teploty, ale aj vlhkosti vzduchu, a to konkrétne odchýlkou ±2% pri meraní vlhkosti a priemerne ±0.5°C (najhoršie ±1°C) pri meraní teploty. Dátová logika je v rozmedzí 3.3V až 5V. Odporúčaný minimálny interval medzi jednotlivými meraniami sú 2s. Senzor ponúka digitálny výstup cez SDA a obsahuje celkovo 4 piny.

Obr. 1.1.2: DHT22, predný pohľad
(Zdroj:https://www.laskakit.cz/user/related_files/am2302_datasheet.pdf)

1. 1. 3 Gyroskop a akcelerometer

Náš modul disponuje monitorovaním vonkajšieho stavu úľa. Pod pojmom vonkajší stav úľa sa rozumie či sa úľ nachádza v korektnej orientácii, čiže sleduje horizontálnu alebo vertikálnu polohu úľa. K tomu nám bude slúžiť akcelerometer, ktorý meria vektor gravitačného zrýchlenia. V rámci šetrenia priestorom volíme 3-osí gyroskop a akcelerometer GY-521, MPU6050, ktorý funguje na báze MEMS. Komunikácia prebieha prostredníctvom sériového rozhrania na báze I^2C. Obsahuje jednu červenú LED diódu, ktorá signalizuje aktívny stav senzora.



Obr. 1.1.3: MPU6050, predný a zadný pohľad
(Zdroj:https://www.indiamart.com/proddetail/gy-521-mpu6050-module-3-accelerometer-for-arduino-22309698791.html)

1. 1. 4 GSM modul

Keďže mobilná sieť je v dnešnej dobe pomerne dostupná, odosielanie všetkých údajov do databázy realizujeme prostredníctvom nej. Prístup k mobilnej sieti nám sprostredkováva GPRS/GSM arduino shield SIM900, ktorý je kompatibilný s 2G sieťami. Shield obsahuje jeden

integrovaný port pre bežnú SIM kartu. V sleep mode má nízku spotrebu a to iba 1.5mA. Energeticky najnáročnejšia pasáž je počas odosielania dát. Vtedy na určitú dobu modul berie prúd okolo 80mA. GPRS/GSM shield je konfigurovaný a kontrolovaný prostredníctvom UART rozhrania pomocou AT príkazov. Napájanie je realizované prostredníctvom priameho zapojenia výstupných pinov TP4056 na piny 5V a GND shieldu. Komunikácia ESP8266 a SIM900 je sprostredkovaná pomocou sériového rozhrania.

Obr. 1.1.4: Arduino GSM/GPRS shield SIM900
(Zdroj:https://www.laskakit.cz/arduino-gprs-gsm-shield-sim900/)

1. 2 Napájacia časť modulu

V tejto podkapitole sa budeme venovať klasifikácii a zdôvodnení výberu súčiastok, ktoré boli potrebné pre napájanie modulu. Ako bolo vyššie spomínané, náš modul bude pracovať v exteriéry bez prístupu napájania z konvenčnej elektrickej siete. Z toho vyplýva, že sme volili napájanie prostredníctvom nabíjateľného elektrického akumulátora. Samotné nabíjanie akumulátora bude realizované prostredníctvom fotovoltaického článku. Náš modul bude taktiež vedieť prečítať aktuálnu percentuálnu úroveň nabitia akumulátora.

1. 2. 1 Akumulátor

Napájanie je realizované prostredníctvom Li-Po akumulátora o kapacite 500mAh, výnimku bude tvoriť modul, ktorý obsahuje GSM. Ten bude napájaný Li-Po akumulátorom o kapacite 1500mAh, a to z dôvodu vyššej spotreby elektrickej energie. Napätie bežnej batérie sa pohybuje na úrovni 3.7V. Batéria disponuje ochranou proti úplnému vybitiu, čomu zodpovedá hodnota napätia 3V, ale taktiež proti úplnému preťaženiu, čo predstavuje hodnotu napätia 4.26V. Nabíjanie akumulátora sa realizuje pomocou napätia 4.2V, pričom je možné zvoliť dva režimy nabíjania. Prvý je prostredníctvom konštantného prúdu 0,2C a napätia do 4,25V alebo pomocou konštantného napätia 4,2V a prúdu do 0,05C.

Obr. 1.2.1.1: Akumulátor bežného modulu (Zdroj:https://www.laskakit.cz/baterie-li-po-3-7v-500mah-lipo/)

Obr. 1.2.1.2: Akumulátor modulu obsahujúci GSM shield (Zdroj:https://www.laskakit.cz/baterie-li-po-3-7v-1500mah-lipo/)

1. 2. 2 Fotovoltaický článok

Samotný, vyššie spomínaný akumulátor, bude primárne nabíjaný prostredníctvom slnečného svetla, ktoré zachytí fotovoltaický článok pričom prekonvertuje svetelnú energiu na energiu elektrickú. Nami zvolený solárny článok je postavený na polykryštalickej báze. Výstupná úroveň napätia fotovoltaického článku sa pohybuje na úrovni 6V, pričom dokáže poskytovať nabíjací výkon približne okolo 1W. Výstupný prúd je do 167mA. Skratový prúd je 186mA. Rozmeri solárneho článku sú kompaktné a to konkrétne 110x60x3mm. Reálny výstupný výkon samozrejme záleží od vonkajších podmienok. Najideálnejšie podmienky sú počas letných mesiacov za slnečného počasia a najhoršie počas zimných mesiacov alebo počas zamračeného počasia.

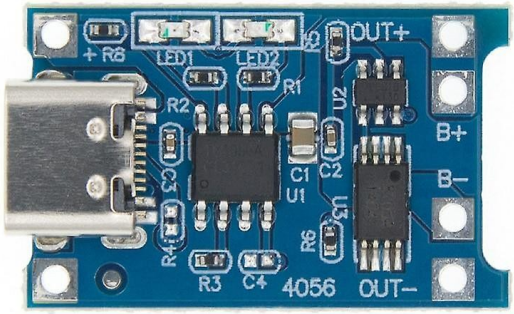
Výnimku bude tvoriť modul s GSM, ktorý bude disponovať väčším a výkonnejším solárnym článkom. Je to z dôvodu väčšej kapacity akumulátora a vyššej spotreby elektrickej energie vďaka spomínanému GSM modulu. Tento variant dodáva výstupný prúd do 333mA. Skratový prúd je 372mA. Elektrické napätie naprázdno sa pohybuje na úrovni 7.3V. Výstupný výkon je okolo 2W a rozmeri sú 110x136x3mm.

Obr. 1. 2. 2. 1: Solárny panel modulu, ktorý obsahuje GSM modul (Zdroj:https://www.laskakit.cz/solarni-panel-6v-2w/)

Obr. 1. 2. 2. 2: Solárny panel bežného modulu (Zdroj:https://www.laskakit.cz/solarni-panel-6v-1w/)

1. 2. 3 Nabíjacia stanica

K úspešnému nabíjaniu nášho akumulátora pomocou solárneho článku budeme potrebovať nabíjaciu stanicu, ktorá obsahuje ochranné a stabilizačné obvody. Tieto obvody sú potrebné pre bezpečné nabíjanie nášho Li-Po akumulátora, keďže výstupné napätie a prúd sú v prípade fotovoltaického článku premenlivé. My však potrebujeme stabilné hodnoty. V našom prípade sme si zvolili nabíjačku TP4056. Výstupné napätie je 4.2V. Nabíjacia stanica obsahuje celkovo 6 pinov, plus k tomu micro USB vstup. Nabíjačka obsahuje dve indikačné LED diódy. Modrá dióda signalizuje plné nabitie batérie a červená LED dióda signalizuje nabíjanie akumulátora, pričom mení svoju svetelnú intenzitu podľa výstupného výkonu fotovoltaickej jednotky.



Obr. 1. 2. 2: Fyzické zobrazenie TP4056 (Zdroj:https://www.laskakit.cz/nabijecka-li-ion-clanku-tp4056-s-ochranou-microusb/)

1. 2. 4 Ostatné pomocné elektronické súčiastky

Vychádzame zo skutočnosti, že nabíjacia stanica TP4056 má výstupné napätia 4.2V. Avšak napájacie napätie mikrokontroleru na báze čipu ESP8266 je iba 3.3V. Z toho dôvodu sme potrebovali usmerniť hodnotu výstupného napätia nabíjačky TP4056 na 3.3V. K tomu sme potrebovali lineárny regulátor napätia. My sme zvolili AMS1117. Pre správne plynulé fungovanie regulovania napätia sme potrebovali ešte dva kondenzátory. Jeden kondenzátor bol elektrolytický o kapacite 100µF a maximálnom napätí 35V pri ±20% odchýlke. Druhý kondenzátor bol keramický o kapacite 100nF a maximálnom napätí 50V pri ±20% odchýlke.

Pri zisťovaní percentuálneho nabitia akumulátora sme potrebovali dva rezistory, ktoré taktiež znižovali výstupné napätie batérie pomocou napäťového deliča. Je to z toho dôvodu, že analógový pin Node MCU V3 toleruje maximálne vstupné napätie do 3.3V. Jeden rezistor bol o hodnote 100kΩ. Druhý rezisotr bol o hodnote 33kΩ. Ďalší nesúvisiaci rezistor, ktorý bol o hodnote 4.8kΩ, bol použitý pri zapojení senzoru DHT22 k Node MCU V3, kvôli stabilnému prenosu údajov.

2. Fyzická architektúra modulu

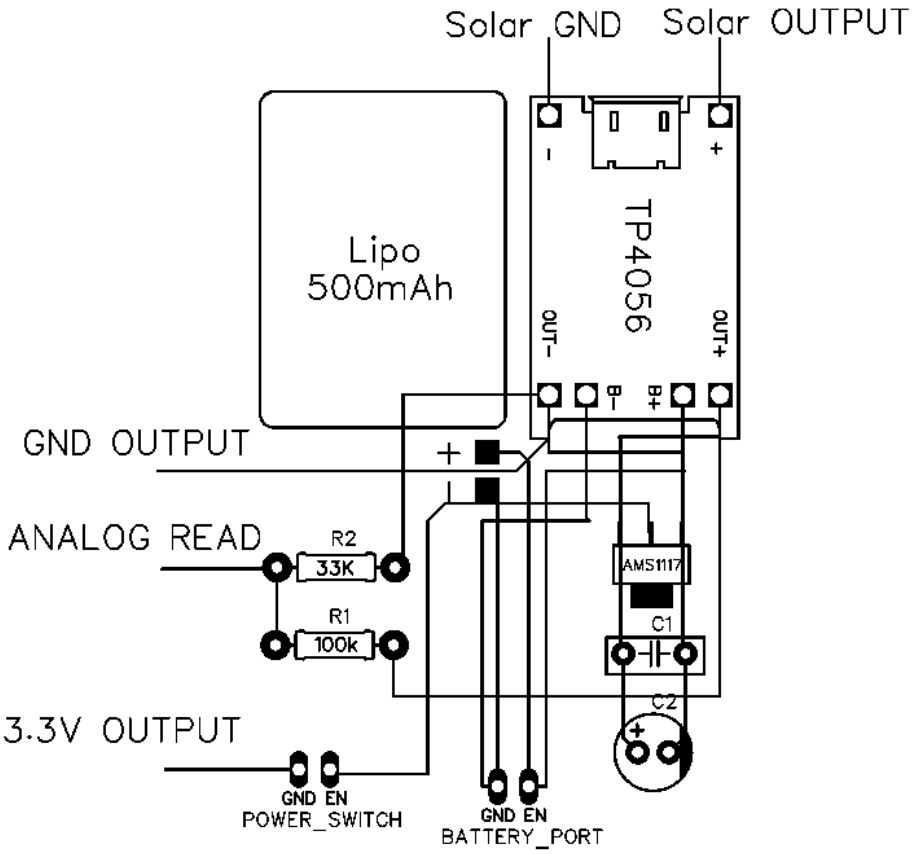
V tejto kapitole sa budeme venovať samotnému obvodu modulu. Konkrétne tomu, ako bol skonštruovaný, a do akých celkov je rozdelený. Náš modul môžeme rozdeliť na dva typy a to aj napriek tomu že jednotlivé moduly sú prepojené pomocou mesh network. Je to z dôvodu, že aj keď formálne žiadny modul nezastupuje funkcie servera, tak neformálne predsa existuje centrálny modul, ktorý cez mobilnú sieť odosiela všetky údaje, od všetkých modulov v sieti, do databázy. Samotná odlišnosť neformálne centrálneho modulu spočíva len v pridaní GPRS/GSM shieldu plus väčšej kapacite batérie a výkonnejšiemu fotovoltaickému článku.

2. 1 Architektúra bežného modulu

Pôvodný modul mal zodpovedať kompaktnejšiemu rozmeru z hľadiska vonkajšej plochy. Inak povedané na výšku mal byť väčší. Avšak z praktických dôvodov sme sa rozhodli pre rozsiahlejšiu no zároveň nižšiu variantu. Hlavný praktický dôvod je, že modul bude uložený priamo vo vnútri včelieho úľa, kde je potrebné prispôsobiť sa hrúbke priestoru jednotlivých stien. Čiže náš finálny projekt bude kvádrového tvaru podobnému pevnému disku. Ako už bolo spomínané v prvej kapitole, náš modul je rozdelený na napájací a riadiaci celok. V tejto podkapitole prejdeme každý z nich, pričom v závere uvedieme prepojenie dvoch celkov do finálneho modulu.

2. 1. 1 Napájací elektronický obvod

Samotný napájací celok vieme rozdeliť do troch elektronických obvodov. Prvý obvod pripája akumulátor do TP4056 článku. Druhý elektronický obvod usmerňuje výstupné elektrické napätia TP4056 modulu na hodnotu 3.3V. Zároveň ponúka výstup pre napájanie Node MCU V3. Posledný elektronický obvod znižuje výstupné elektrické napätie TP4056 článku pomocou napäťového deliča, pričom poskytuje jeden výstup pre analógový pin riadiacej jednotky na báze ESP8266. Pomocou tohto usmernenia vieme bezpečne čítať elektrické napätie akumulátora, z čoho následne vieme vyjadriť percentuálne nabitie batérie. Nasledujúca schéma zobrazuje celú napájaciu zložku.



Obr. 2. 1. 1: Schéma napájacieho elektroníckeho obvodu
(Zdroj: vlastný)

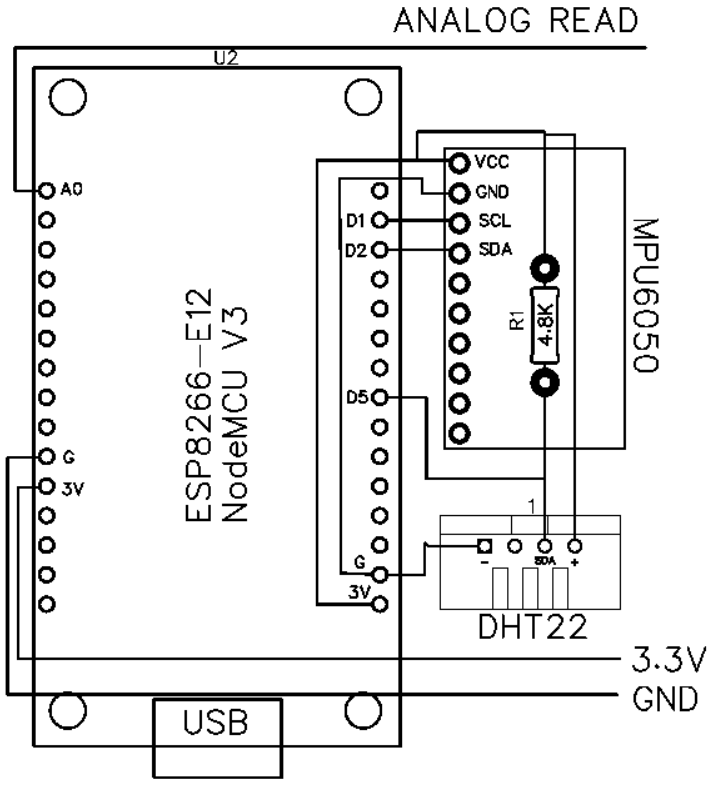
Výstupné elektrické napätie z fotovoltaického článku je pripojené na piny - a + pričom nemôžeme ignorovať korektnú polaritu zapojenia. Výstup akumulátora bol konektor typu samec. Preto sme sa rozhodli že na plošnú dosku pripevníme port typu samica, aby sme jednoducho vedeli pripojiť a odpojiť akumulátor v prípade potreby. Následne z battery portu pripájame vodivým spojom akumulátor do pinov B+ a B- TP4056, pričom znova zachováme správnu polaritu.

K regulovaniu výstupného napätia nám slúži hlavne lineárny regulátor AMS1117, ku ktorému sú avšak paralelne za sebou pripojené dva kondenzátory o kapacite 100µF a 100nF. Výstup elektrického napätia z modulu TP4056 je pripojený ku krajným nožičkám AMS1117, pričom je zachovaná správna polarita. Stredná nožička slúži ako kladný výstup.

Pre bezpečné čítanie hodnoty napätia z akumulátora v analógovom pine ESP8266, sme použili napäťový delič. K 33kΩ rezistoru sme k pravému koncu pripojili pin OUT- TP4056 celku. A naopak pin OUT+ TP4056 sme pripojili k pravému koncu 100kΩ rezistora. Druhé konce rezistorov sú navzájom prepojené. Samotné prepojenie ponúka jeden výstup z ľavej strany rezistorov, ktorý je možné zapojiť do analógového pinu ESP8266.

2. 1. 2 Riadiaci elektronický obvod

Riadiaci obvod sa skladá z riadiacej jednotky na báze ESP8266 a príslušných senzorov. Architektúra tohto celku je jednoduchšia. Všetky komponenty elektronického obvodu, okrem pomocného rezistora, sú vsunuté do naspájkovaných dutinkových líšt. Napájanie mikrokontrolera ESP8266 Node MCU V3 je realizované prostredníctvom pinov 3V a G na ľavej strane (viď obrázok 2. 1. 2). Riadiaca jednotka, ako bolo spomínané, monitoruje percentuálny stav nabitia batérie prostredníctvom čítania elektrického napätia z batérie skrz analógový pin A0. Nasledujúci obrázok zobrazuje schému elektronického obvodu riadiacej zložky.



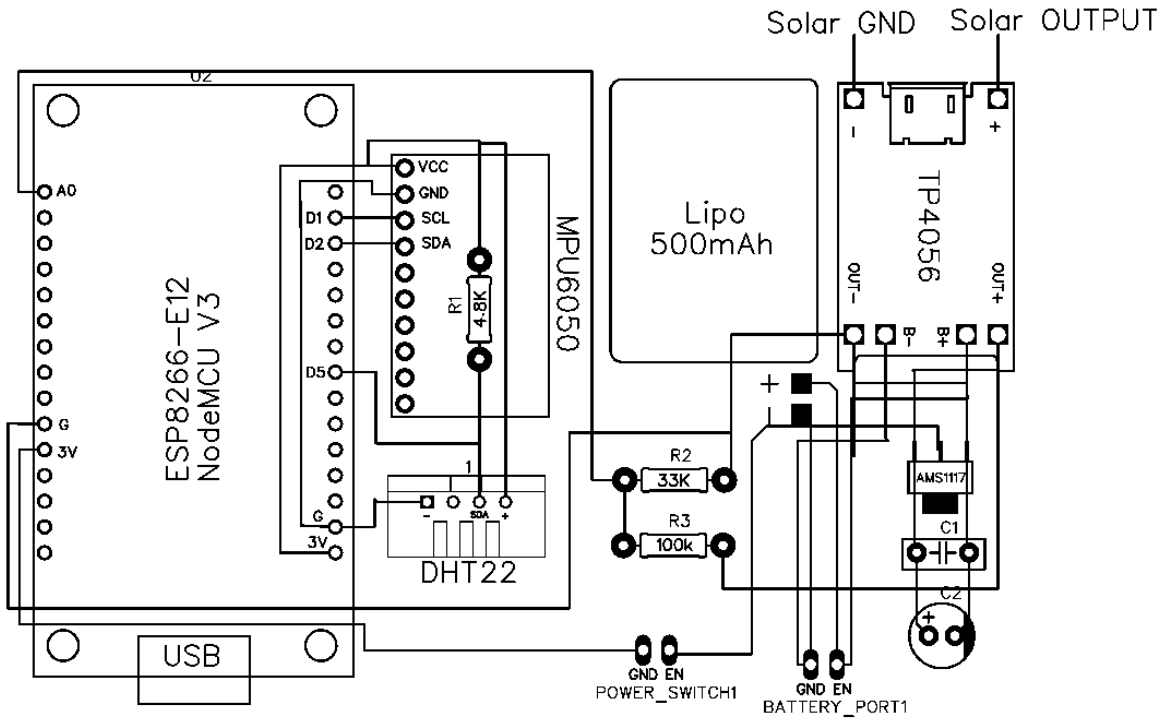
Obr. 2. 1. 2: Schéma riadiaceho elektronickeho obvodu
(Zdroj: vlastný)

Teplomer a vlhkomer DHT22 je napájaný prostredníctvom pinov 3V a GND na pravej strane mikrokontroleru na báze ESP8266, pričom je dodržaná korektná polarita zapojenia. Prenos dát je sprostredkovaný prepojením SDA výstupu DHT22 zapojeného do pinu D5 resp. GPIO14 riadiacej jednotky na báze ESP8266. Pre stabilizačné potreby sme SDA výstup prepojili s 3V pinom skrz 4.8kΩ rezistor.

Gyroskop a akcelerometer MPU6050 je taktiež napájaný prostredníctvom rovnakých pinov 3V a GND ako senzor DHT22. Polarita zapojenia je nutná. Dáta sú prenášané skrz dva piny. Na strane MPU6050 sú to piny SCL a SDA. Na strane Node MCU V3 sú to piny D1, D2 resp. GPIO5, GPIO4. Piny sú zapojené postupne SCL do GPIO5 a SDA do GPIO4.

2. 1. 3 Celý elektronický obvod

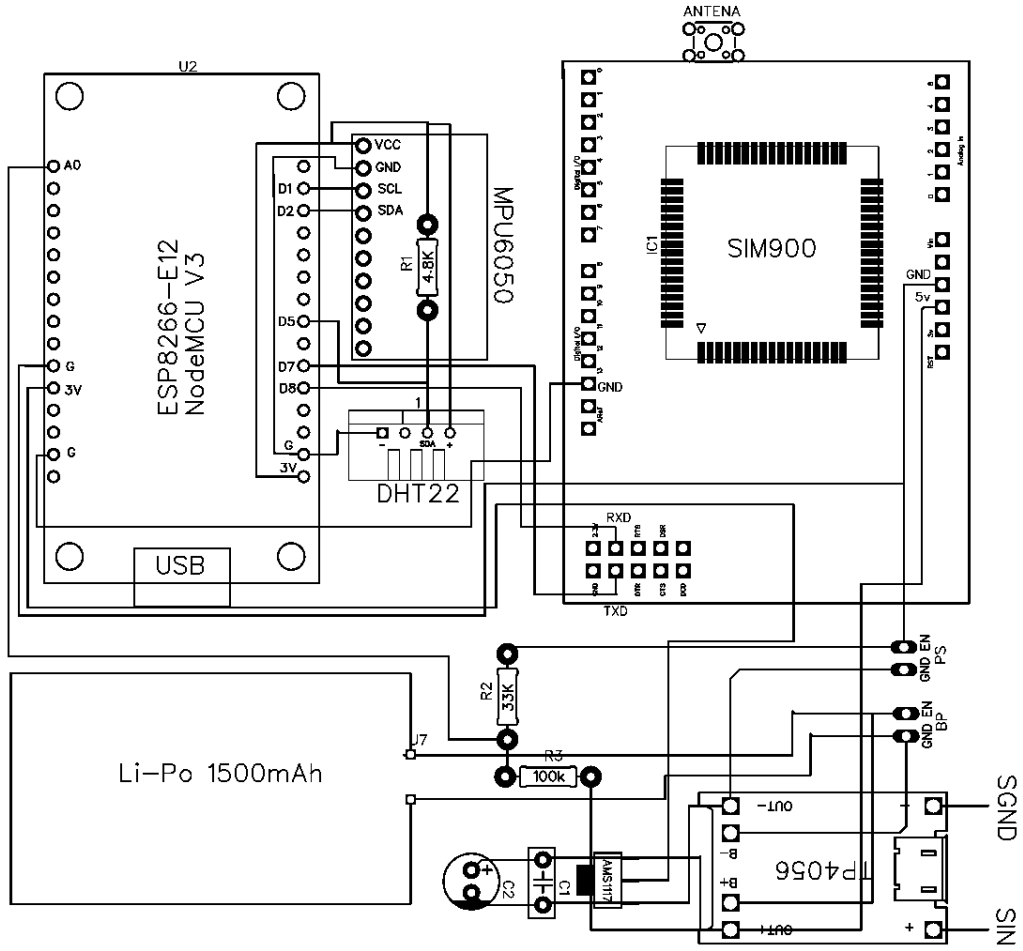
Celky modulu sú prepojené prostredníctvom troch vodivých spojov. Dva vodivé spoje prislúchajú napájaniu a jeden slúži ku čítaniu elektrického napätia batérie prostredníctvom analógového pinu. Celý modul bude vložený do plastovej krabičky, ktorá bude vytlačená prostredníctvom 3D tlačiarne. Nasledujúci obrázok ukazuje schému zapojenia celého modulu.



Obr. 2. 1. 3: Schéma bežného modulu
(Zdroj: vlastný)

2. 2 Architektúra neformálne centrálneho modulu

Tento elektronický obvod je podobný s elektronickým obvodom bežného modulu. Odlišnosti v tomto prípade tvorí pridanie GPRS/GPS shieldu SIM900 a zväčšenie kapacity akumulátora na hodnotu 1500mAh. Je taktiež použitý výkonnejší fotovoltaický článok, aby sme uspokojili väčší energetický dopyt. Nasledujúci obrázok zobrazuje schému celého neformálne centrálneho modulu.



Obr. 2. 2: Schéma neformálne centrálneho modulu
(Zdroj: vlastný)

Vďaka spomínaním úpravám sa upravila aj celková rozloha modulu k väčšiemu obsahu. Samotný modul pripomína teraz viac štvorcový tvar. Kvôli väčšiemu akumulátoru sme zvolili osobitné pripevnenie akumulátora na rozdiel od integrovaného pripevnenia bežného akumulátora. Výstup nabíjačky TP4056 v tomto prípade ide najprv na plošnú dosku pripevnenú na SIM900 shielde, až následne napájame mikrokontroler ESP8266. GRPS/GSM shield je napájaný prostredníctvom priameho výstupu, bez regulácie elektrického napätia a nabíjacej stanice TP4056. Samotný GSM shield je stavaný na vyššie napätie, preto nám to bolo umožnené. Originálne je prispôsobený na elektrické napätie 5V. Je však možné napájanie aj prostredníctvom nižších

elektrických napätí. Práve takými hodnotami ktorými disponuje výstup s nabíjačky TP4026.

Prepojenie riadiacej jednotky na báze EPS8266 a GPRS/GSM shieldu bolo realizované prostredníctvom sériového rozhrania, pričom sme v prípade Node MCU V3 použili piny D7 a D8 ako RX a TX piny. Táto možnosť bola nutná pretože priame piny RX a TX neboli z hardwerových dôvodov možné. Avšak už samotná sekundárna úloha pinov D7 a D8 je totožná s RX a TX pinmi. Piny sériovej komunikácie sme prepojili krížom. D8(RX) sme prepojili s TXD pinom GSM shieldu a D7(TX) sme prepojili s RXD pinom shieldu. K úspešnému fungovaniu sme prepojili GND pin GPRS/GSM shieldu s G pinom Node MCU V3 mikrokontrolera. Ostatné zapojenie je v podstate totožné so zapojením bežného modulu.

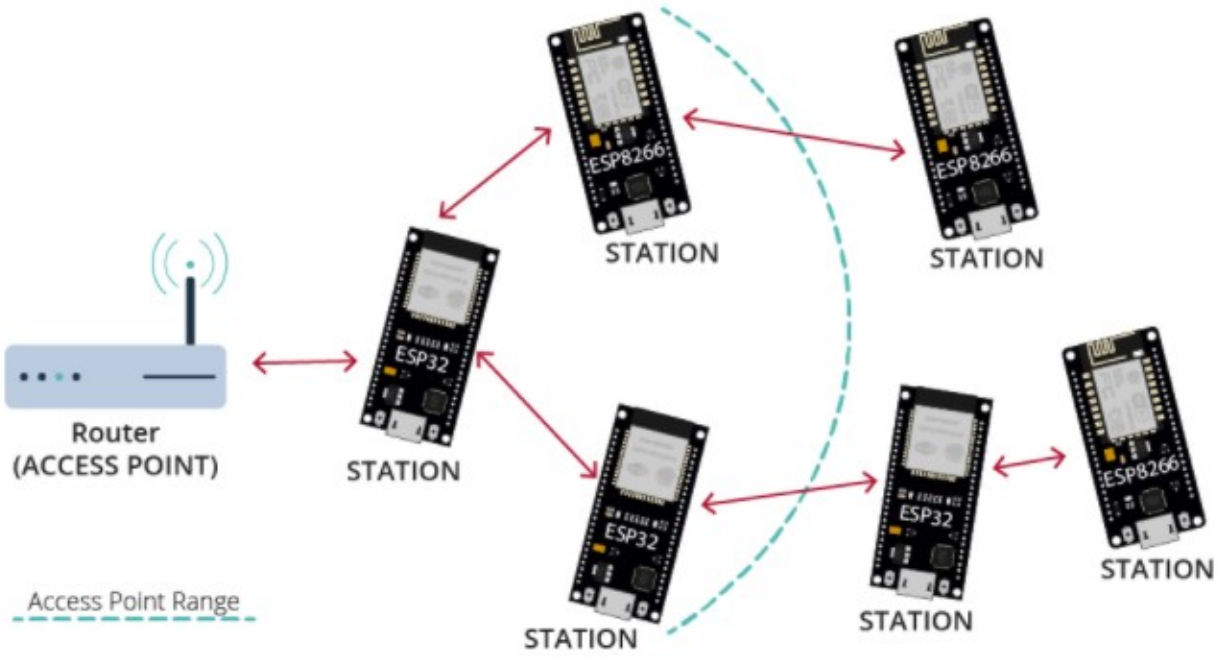
3 Softwerová časť projektu

V tejto kapitole si rozoberieme sotwarovú stránku nášho projektu. Konkrétne sa pozrieme na serverovú stranu, ktorá obsahuje Node-red a MySql databázu. Potom sa pozrieme na interný kód nahraný vo FLASH pamäti modulu. Taktiež sa pozrieme na všetky knižnice, ktoré sme využívali. Celý aktualizovaný zdrojový kód si môžete pozrieť na github účte humatej v repozitáry Alb_msg_project2022.

3. 1 Esp mesh sieť

Esp mesh siet sa definuje ako sieť, ktorá je decentralizovaná a v našom prípade aj asynchrónna. Pod pojmom decentralizovaná sieť chápeme takú sieť, ktorá nemá centrálne zariadenie, ktoré má úlohu servera. Čiže v mesh sieti nemáme stranu servera a stranu klienta. Všetky zariadenia mesh siete fungujú zároveň ako server aj ako klient naraz. Výhodou tejto siete je fakt, že je samoregulovateľná, a že v nej prebieha dynamická komunikácia medzi modulmi. To znamená, že je zabezpečené doručenie paketu do neformálne hlavného modulu aj za prípadu ak by sa daný modul odpojil od siete. Aj v samotnom prípade, ak by sa modul odpojil od siete, vie sa naspäť automaticky pripojiť. Ďalší benefit siete spočíva v ľahkej škálovateľnosti, čiže sa hodí na rozsiahle prepojene zariadení aj na väčšie vzdialenosti, pokiaľ to umožňuje hardware mikrokrotrolerov.

Pod pojmom asynchrónnosť siete rozumieme už spomínaný fakt dynamickej komunikácie medzi modulmi. Vo všeobecnosti však môžeme povedať, že asynchrónnosť je synonymum pre slovo paralelnosť.



Obr. 3. 1: Schéma Esp mesh siete
(Zdroj:https://randomnerdtutorials.com/esp-mesh-esp32-esp8266-painlessmesh/)

3. 2 Klasifikácia použitých knižníc

V tejto podkapitole sa budeme stručne venovať klasifikácii vybraných knižníc. Knižnice môžeme rozdeliť zhruba nasledovne. Prvá skupina zahŕňa knižnice, ktoré boli použité pri interpretácii meraní zo senzorov. Druhá skupina zahŕňa knižnice, ktoré slúžili vo všeobecnosti na prenos dát, či už medzi jednotlivými modulmi, alebo aj na samotné posielanie dát do databázy.

Pri interpretácii údajov zo senzora DHT22, ktorý meria teplotu a vlhkosť vzduchu, sme použili knižnicu DHT senzor library od spoločnosti Adafruit. Daná knižnica bola zvolená kvôli faktu, že samotný senzor DHT22 je vyrobený spoločnosťou Adafruit, čiže disponujeme originálne navrhnutou knižnicou pred daný účel. Vďaka metódam zahrnutých v knižnici vieme jednoducho získať hodnotu teploty a vlhkosti vzduchu. K správnemu fungovaniu knižnice bolo potrebné zahrnúť závislú knižnicu Adafruit Unified Senzor Driver, tiež od spoločnosti Adafruit.

Údaje z gyroskopu a akcelerometra MPU6050 nám sprostredkovala knižnica Adafruit MPU6050 library od spoločnosti Adafruit. Dôvod výberu bol totožný ako dôvod výberu knižnice pre DHT22. Práca s knižnicou bola trochu náročnejšia, pretože senzor MPU6050 sa musí zo začiatku nakalibrovať. Vďaka metódam knižnice máme prístup k údajom o jednotlivých zrýchleniach podľa osi x, y a z. Podľa jednotlivých osí tiež vieme získať údaje z gyroskopu.

Na tvorbu mesh siete medzi jednotlivými modulmi sme použili knižnicu PainlessMesh. Táto knižnica ponúka možnosť asynchrónneho a relatívne jednoduchého vytvorenia decentralizovanej

mesh siete. Knižnica disponuje širokou variabilitou metód, ktoré môžu byť použité. Základná periodicita posielania údajov sa nastavuje cez triedu Task.

Sériovú komunikáciu GPRS/GSM shieldu a Node MCU V3 sme zabezpečili pomocou knižnice EspSoftwareSerila od autora Dirk Kaar. Knižnica obsahuje triedu SoftwareSerial, vďaka ktorej môžeme vytvárať objekt, ktorý je viazaný na piny sériovej komunikácie. Pomocou daného objektu vieme pomocou bežných arduino metód, ako napr. *print()*, sériovo komunikovať medzi zariadeniami.

Prístup do databázy a MySql príkazy zadávané v Javascripte nám zabezpečuje knižnica Node-red-node-mysql. Ktorú avšak využívame v samotnom Node Red flow pričom daná knižnica nie je súčasťou kódu modulu.

3. 3 Software bežného modulu

Celý kód môžeme rozdeliť do nasledujúcich dvoch celkov. Prvý celok sprístupňuje a pripravuje dáta na poslanie a druhý celok spravuje a udržuje sieťové spojenie medzi zariadeniami, pričom taktiež zabezpečuje preposielanie dát. Nasledujúci kód sa nachádza v alpha verzii, čiže v budúcnosti je tu priestor na refaktorizáciu a optimalizáciu.

3. 3. 1 Incializačná hlavička kódu

```
#include "painlessMesh.h"
#include <Adafruit_MPU6050.h>
#include <Adafruit_Sensor.h>
#include <DHT.h>
#include <Wire.h>
// konštanty pre mesh siet
#define   MESH_PREFIX     "albMsg"
#define   MESH_PASSWORD   [redacted]
#define   MESH_PORT       5555

Scheduler userScheduler; // kontrola personalizovaním taskov
painlessMesh  mesh;

#define ANALOG_PIN A0

// definovanie základných konštát pre DHT22
#define DHTPIN 14
#define DHTTYPE DHT22
// Adafruit senzori
Adafruit_MPU6050 mpu;
DHT dht(DHTPIN, DHTTYPE);
const String modul_num = "M1";
bool mpuError = false;
```

Obr. 3. 3. 1: ukážka incializačnej hlavičky
(Zdroj: vlastný)

Daný obrázok reprezentuje hornú hlavičku kódu, v ktorej postupne inicializujeme potrebné knižnice a pomocou *#define* definujeme základné potrebné konštanty. Taktiež deklarujeme

senzorové objekty pomocou tried na to určených. Každý modul má dvojznakový názov, ktorý je uložený do premennej modul_num. MESH_PREFIX reprezentuje názov mesh siete. My sme sa rozhodli pre názov albMsg, ktorý odkazuje na názov projektu Albertov odkaz podľa anglického slovného spojenia Albert Message. MESH_PASSWORD zastupuje heslo mesh siete, ktoré je v našom prípade 13-miestne, pričom kombinujeme veľké písmená, malé písmena a čísla. Komunikácia medzi zariadeniami prebieha na porte číslo 5555.

3. 3. 2 Deklarácia metód

Nasledujúca podkapitola popisuje deklaráciu všetkých metód či už vlastných alebo závislých na knižnici PainlessMesh. V našom kóde sme využili tri vlastné metódy. Prvá naša metóda *isFallen()* typu *bool* zisťuje pomocou akcelerometra MPU6050 či sa hlavné gravitačné zrýchlenie neprenieslo svojimi zložkami na iné osi od osi základnej. V našom prípade je základná os **Z.** Pod pojmom základná os sa myslí taká os, ktorá je kolmá na rovinu. Výstupné hodnoty funkcie sú typu *bool.* Inak povedané pod hodnotou 0 sa chápe taký stav úľa, ktorý nie je prevrátený, resp. nedošlo k zmene orientácie. Hodnota 1 znamená, že úľ zmenil svoju orientáciu.

```
100  bool isFallen(){
101      sensors_event_t a, g, temp;
102      mpu.getEvent(&a, &g, &temp);
103      if(absolute(a.acceleration.x) > 5 || absolute(a.acceleration.y) > 5){
104        return 1;
105      }
106      return 0;
107  }
```

Obr. 3. 3. 2. 1: ukážka metódy isFallen()
(Zdroj: vlastný)

Druhá naša metóda *batPer()* typu *float* prepočítava stav nabitia batérie z hodnoty elektrického napätia na percentuálnu hodnotu, pričom používame na mieru nami vytvorené prevodové konštanty, ktoré vychádzajú aj z minimálneho a maximálneho elektrického napätia poskytnutého pomocou napäťového deliča. Výstupná hodnota je typu *float*, pričom udáva percentuálnu hodnotu nabitia akumulátora

Posledná naša metóda *absolute()* dátového typu *double* vracia absolutnú hodnotu čísla. Je to jediná metóda, ktorá požaduje vstup, a to dátového typu *float*. Motiváciou k vytvoreniu danej metódy bol fakt, že sme nechceli zbytočne deklarovať celú matematickú knižnicu. Nasledujúce metódy vychádzajú priamo z knižnice Painless mesh.

Metóda *sendMessage()* dátového typu *void* vykonáva primárne posielanie údajov pomocou reťazca. V našom prípade sme metódu upravili tak, aby zároveň vykonávala čítanie údajov

z jednotlivých senzorov a čítanie percentuálneho nabitia batérie. Dáta skladáme postupne do jedného reťazca, pričom to robíme spôsobom zápisu pomocou jednoznakového kľuča a príslušnej najčastejšie desatinnej hodnoty. Oddelenie kľúča a hodnoty realizujeme skrz znamienkom =. Tento typ zápisu dát v reťazci sme volili preto, lebo dáta do databázy sú posielané cez url adresu. Dáta sú broadcastom preposielané všetkým modulom pomocou metódy *endBroadcast()* objektu mesh. Následne ešte deklarujeme objekt taskSendMessage typu *Task*, ktorý slúži aj na nastavenie periodicity odosielania údajov broadcastom medzi zariadeniami. Ostatné metódy sú pomocné, čiže ich nebudeme všetky menovať.

```
33  Task taskSendMessage( TASK_SECOND * 60 , TASK_FOREVER, &sendMessage );
34
35  void sendMessage() {
36    String msg = "";
37    msg += "f=" + String(isFallen());
38    if (!isnan(dht.readTemperature()) && !isnan(dht.readHumidity())){
39      String t = String(dht.readTemperature()), h= String(dht.readHumidity());
40      msg += "&t=" + t.substring(0,4);
41      msg += "&h=" + h.substring(0,4);
42    }
43    msg += "&b="+ String(batPer());
44    msg += "&m=" + modul_num;
45    mesh.sendBroadcast( msg );
46    taskSendMessage.setInterval( random( TASK_SECOND * 59, TASK_SECOND * 61 ));
47  }
```

Obr. 3. 2. 2. 2: ukážka metódy sendMessage()
(Zdroj: vlastný)

3. 3. 3 Zloženie funkcie setup() a loop()

V bežnej funkcii *setup()* spúštame metódami funkcionality mesh siete ale aj samotného merania senzorov. A vo funkcii *loop()* zadávame metódu objektu mesh *update()*, ktorá je zodpovedná za udržiavanie mesh siete. Zároveň táto metóda spravuje a spúšťa ostatné deklarované metódy PainlessMesh knižnice.

3. 4 Software neformálne centrálneho modulu

Software neformálne centrálneho modulu je takmer totožný so softwarom bežného modulu. Odlišnosť spočíva v pridaní knižnice EspSoftwareSerial, ktorá sprostredkúva sériovú komunikáciu medzi Node MCU V3 a GPRS/GSM shieldom SIM900. Ďalej sme pridali vlastnú metódu *sendData()*, ktorej sa povenujeme bližšie.

Metóda *sendData()*, ktorá je dátového typu *void*, posiela dátový reťazec pomocou AT príkazov cez sériovú komunikáciu sprostredkúvanú objektom typu *SoftwareSerial*. Konštrukcia zadávania príkazov je nasledovná. Na začiatku sme si deklarovali objekt gsmSerial typu *SoftwareSerial.* V konštruktore sme zadali číselné hodnoty GPIO pinov, ktoré sú zapojené na RXD

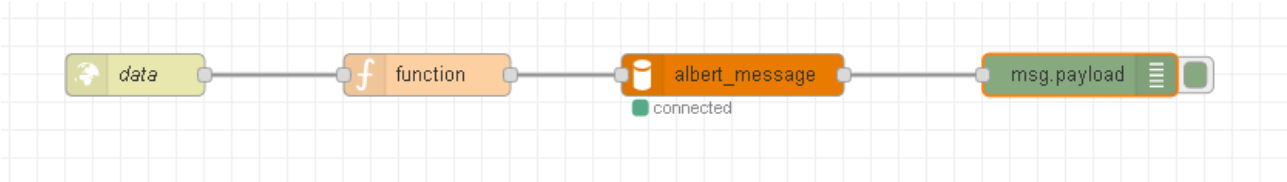
TXD piny GSM shieldu. V Našom prípade to boli spomínané GPIO piny čísla 13 a 15. Po deklarácii objektu môžeme využívať metódy zápisu ako *print(), println()* alebo *write()*. My využívame metódy *print()* a *println()*. Jediný rozdiel v týchto metódach je, že funkcia *print()* posiela reťazec primárne bez ukončovacieho znaku enter. Dátový reťazec je posielaný cez url adresu. Danú metódu voláme v prípade, že sme prijali dátový reťazec od iného modulu prostredníctvom mesh siete. V budúcnosti však plánujeme refaktorizovať kód tak, že jednotlivé dátové reťazce budeme ukladať do poľa typu *string*. Následne po určitej časovej dobe, napr. 5. minútach, budeme postupne volať metódu *sendData()*, pričom aj medzi jednotlivými volaniami funkcie bude časové oneskorenie. Je to z dôvodu, že AT príkazy nemôžu byť zadávané asynchrónne, ale lineárne.

```
134  void sendData(String data){
135    gsmSerial.println(F("AT+SAPBR=3,1,\"CONTYPE\",\"GPRS\""));
136    updateSerial();
137    delay(2000);
138    gsmSerial.println(F("AT+SAPBR=3,1,\"APN\",\"internet\""));
139    updateSerial();
140    delay(2000);
141    gsmSerial.println(F("AT+SAPBR=1,1"));
142    updateSerial();
143    delay(2000);
144    gsmSerial.println(F("AT+SAPBR=2,1"));
145    updateSerial();
146    delay(2000);
147    gsmSerial.println(F("AT+HTTPINIT"));
148    updateSerial();
149    delay(2000);
150    gsmSerial.println(F("AT+HTTPPARA=\"CID\",1"));
151    updateSerial();
152    delay(3000);
153    gsmSerial.print(F("AT+HTTPPARA=\"URL\",\"www.sstv-prax.tk:1880/send?"));
154    gsmSerial.print(data);
155    gsmSerial.print(F("\r\n"));
156    updateSerial();
157    delay(2000);
158    gsmSerial.print(F("AT+HTTPPARA=\"CONTENT\",\"application/json\"\r\n"));
159    updateSerial();
160    delay(5000);
161    gsmSerial.println(F("AT+HTTPACTION=0"));
162    updateSerial();
163    delay(6000);
164    gsmSerial.println(F("AT+HTTPTERM"));
165    updateSerial();
166    delay(2000);
167  }
```

Obr. 3. 3: ukážka metódy sendData()
(Zdroj: vlastný)

3. 5 Serverová stránka projektu

Keďže všetky údaje sú posielané do databázy, potrebovali sme zriadiť server. Server je založený na Rasbery Pie, pričom pomocou Node Red zabezpečujeme prístup k MySql databáze. Všetky dáta sú posielané cez url adresu, pričom v Node Red čítame údaje z GET požiadavky.



Obr. 3. 5: štruktúra Node Red flow
(Zdroj: vlastný)

Nasledujúca schéma zobrazuje jednoduchý Flow vytvorený v Node Red. Blok dáta zachytáva údaje z GET požiadavky a posiela ich ďalej v JSON formáte. Blok funkcia následne skontroluje či sú zadané všetky hodnoty po jednotlivých stĺpcov tabuľky MySql databázy. Konkrétne kontrolujeme iba prípadnú absenciu dát zo senzora DHT22. Následne sú dáta do databázy posielané pomocou MySql príkazu INSERT. Posledný blok slúži na debugovacie účely, nie je nutne potrebný.

3. 5. 1 Architektúra MySql databázi projektu

#	Názov	Typ	Zotriedenie	Atribúty	Nulový	Predvolené	Komentáre	Extra
1	**id**	int(8)		UNSIGNED	Nie	*Žiadny*		AUTO_INCREMENT
2	**modul**	varchar(2)	utf8mb4_general_ci		Nie	*Žiadny*		
3	**temperature**	float			Áno	*NULL*		
4	**humidity**	float			Áno	*NULL*		
5	**battery**	float			Nie	*Žiadny*		
6	**isFallen**	tinyint(1)			Nie	*Žiadny*		
7	**time**	timestamp			Nie	current_timestamp()		

Obr. 3. 5. 1: štruktúra MySql databázi
(Zdroj: vlastný)

Nasledujúca štruktúra ukazuje, ako sme zvolili ukladanie dát v našej MySql databáze. Rozpoznávacia tabuľka, ktorá môže slúžiť ako kľúč, je tabuľka modul typu *varchar* veľkosti 2B, ktorá udáva názov modulu. Následne ostatné tabuľky uchovávajú nami merané veličiny. Pri tabuľke temperature a huminidity nie je potrebné zadávať hodnotu. Je to z dôvodu možného zlyhania senzora DHT22. Tabuľka tiež automaticky generuje dátum a čas, kedy došlo k zápisu dát. Následne budú dáta zobrazené pomocou web stránky, pričom údaje budú brané práve z našej databázy.

Záver

V závere môžeme zhodnotiť, že primárne zadané úlohy sme zvládli splniť nasledovne. Výber hardweru bol postačujúci, ale zároveň nebol úplne zoptimalizovaný. Počas práce sme prehodnotili podmienku gyroskopou. Zistili sme, že ak by sme chceli pomocou gyroskopu zistiť stav zmeny orientácie včelieho úľa, bolo by to v rámci našich možností energeticky náročné. Museli by sme neustále čítať hodnoty gyroskopu. My avšak budeme realizovať čítanie hodnôt len za určitú časovú periódu, aby sme šetrili elektrickou energiou, keďže nedisponujeme veľkou kapacitou batérie. Preto sme sa rozhodli, že na zmenu orientácie použijeme akcelerometer, vďaka ktorému by sme vedeli dosiahnuť náš cieľ. Keďže senzor MPU6050 bol aj zároveň akcelerometer, nemuseli sme nič dokupovať. Počas tvorby elektronického obvodu sme však predsa dokúpili určité menšie súčiastky ako rezistor pre DHT22 alebo port typu samica pre zapojenie batérie, taktiež vypínač a dutinkové lišty. Tvorbou modulov sme postupne refaktovalizovali architektúru modulu až na dnešnú podobu. Logický prevodník nakoniec nebol potrebný keďže sme zistili že GPRS/GSM shield obsahuje aj piny s 3.3V logikou

Odosielanie dát do databázy iba z jedného modulu pomocou GSM siete bolo úspešné. Problém nastal až pri pridaní mesh siete. Esp mesh sieť medzi modulmi zatiaľ funguje, ale už odosielanie z neformálneho hlavného modulu do databázy nie je korektné. Chyba nastáva pri zadávaní konkrétnych AT príkazov. Túto chybu sa budeme v budúcnosti snažiť odstrániť. Ako sme tiež spomínali, náš kód je v alfa verzii, čiže je tu priestor na refaktorizáciu a optimalizáciu.

Prácou sme sa naučili novým zručnostiam. Dozvedeli sme sa aj nové informácie v oblasti včelárstva, keďže sme sa zúčastnili jednej prednášky. Kde sme po krátkej konzultácií so včelárom prehodnotili vonkajší dizajn modulu. Bolo tomu preto lebo sme sa dozvedeli, že samotný modul bude umiestnený vnútri úľa, pričom na začiatku sme si mysleli, že modul sa bude nachádzať na úli. Dúfame taktiež, že naša samotná maturitná práca môže slúžiť ako inšpirácia pre podobné projekty.

Zoznam použitej literatúry

[1] am2302_datasheet.pdf (laskakit.cz) [cit. 2022-03-02]

[2] mpu-6050_datasheet_v3_4.pdf (laskakit.cz) [cit. 2022-03-02]

[3] Solární panel 6V 1W | laskakit.cz [cit. 2022-03-09]

[4] 复件 tp4056_42_English_空页脚.doc (laskakit.cz) [cit. 2022-03-09]

[5] Solární panel 6V 2W | laskakit.cz [cit. 2022-03-09]

[6] Microsoft Word - DS1117 (laskakit.cz) [cit. 2022-03-11]

[7] Dersonic CC1H104ZA1FD3F5P10MF 100nF -20%~+80% 50V kondenzátor keramický | laskakit.cz [cit. 2022-03-11]

[8] aishi-aihua-group-ers.pdf (laskakit.cz) [cit. 2022-03-11]

[9] Arduino GPRS-GSM Shield SIM900 | laskakit.cz [cit. 2022-03-17]

[10] In-Depth: Send Receive SMS & Call with SIM900 GSM Shield & Arduino (lastminuteengineers.com) [cit. 2022-03-17]

[11] GPRS/GSM Shield v1.0 - Elecrow [cit. 2022-03-17]

[12] GitHub - adafruit/DHT-sensor-library: Arduino library for DHT11, DHT22, etc Temperature & Humidity Sensors [cit. 2022-03-20]

[13] GitHub - plerup/espsoftwareserial: Implementation of the Arduino software serial for ESP8266 [cit. 2022-03-20]

[14] ESP-MESH with ESP32 and ESP8266: Getting Started | Random Nerd Tutorials [cit. 2022-03-20]

[15] Power ESP32/ESP8266 with Solar Panels and Battery | Random Nerd Tutorials [cit. 2022-03-20]

[16] ESP8266 NodeMCU MPU-6050 Accelerometer and Gyroscope (Arduino) | Random Nerd Tutorials [cit. 2022-03-20]

[17] What Is A Battery C Rating & How Do I Calculate C Rate - Power Sonic (power-sonic.com) [cit. 2022-03-20]

[18] SIM900/800 HTTP Post Request in JSON Format with Arduino (how2electronics.com) [cit. 2022-03-20]

[19] (317) Tutorial NodeRed : Receive values from http request and save to File - YouTube [cit. 2022-03-20]

[20] node-red-node-mysql (node) - Node-RED (nodered.org) [cit. 2022-03-20]