

TITULNÍ LIST

Namísto této stránky vložte **titulní list** (s logem) vygenerovaný v IS VUT.

ZADÁNÍ

Namísto této stránky vložte stránku **zadání FEKT** vygenerovanou v IS VUT.

ABSTRAKT

Abstrakt práce v originálním jazyce

KLÍČOVÁ SLOVA

Klíčová slova v originálním jazyce

ABSTRACT

Překlad abstraktu (v angličtině, pokud je originálním jazykem čeština či slovenština; v češtině či slovenštině, pokud je originálním jazykem angličtina)

KEYWORDS

Překlad klíčových slov (v angličtině, pokud je originálním jazykem čeština či slovenština; v češtině či slovenštině, pokud je originálním jazykem angličtina)

ROZŠÍŘENÝ ABSTRAKT

Výtah ze směrnice rektora 72/2017:

Bakalářská a diplomová práce předložená v angličtině musí obsahovat rozšířený abstrakt v češtině nebo slovenštině (čl. 15). To se netýká studentů, kteří studují studijní program akreditovaný v angličtině. (čl. 3, par. 7)

Nebude-li vnitřní normou stanoveno jinak, doporučuje se rozšířený abstrakt o rozsahu přibližně 3 normostrany, který bude obsahovat úvod, popis řešení a shrnutí a zhodnocení výsledků. (čl. 15, par. 5)

VAVRINEC, Tomáš. *Název studentské práce*. Semestrální práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2030. Vedoucí práce: prof. Ing. Křestní Příjmení, CSc.

Prohlášení autora o původnosti díla

Jméno a příjmení autora: Tomáš Vavrinec
VUT ID autora: 42
Typ práce: Semestrální práce
Akademický rok: 2029/30
Téma závěrečné práce: Název studentské práce

Prohlašuji, že svou závěrečnou práci jsem vypracoval samostatně pod vedením vedoucí/ho závěrečné práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno
.....
podpis autora*

*Autor podepisuje pouze v tištěné verzi.

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu bakalářské/diplomové/disertační práce panu Ing. XXX
YYY, Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Obsah

1	Teoretická část	25
1.1	Potřeby herních zařízení	25
2	Návrh zařízení	27
2.1	Struktura elektroniky základní jednotky	28
2.1.1	LedDesku	28
2.1.2	MiniUI	29
2.1.3	HlavníDeska	29
2.1.4	Propojení HlavníDesky a LedDesky	30
2.1.5	Modulový konektor	31
2.1.6	Konektor programátor	31
2.1.7	USB-C	32
2.1.8	Konektor komunikačního modulu	32
	Závěr	33
2.2	Použité zdroje	34
	Seznam symbolů a zkratk	39
	Seznam příloh	41
A	Některé příkazy balíčku thesis	43
A.1	Příkazy pro sazbu veličin a jednotek	43
A.2	Příkazy pro sazbu symbolů	43
B	Druhá příloha	45
C	Příklad sazby zdrojových kódů	47
C.1	Balíček listings	47
D	Obsah elektronické přílohy	51

Seznam obrázků

B.1 Alenčino zrcadlo	45
--------------------------------	----

Seznam tabulek

A.1 Přehled příkazů	43
-------------------------------	----

Seznam výpisů

C.1	Ukázka sazby zkratk	47
C.2	Příklad Schur-Cohnova testu stability v prostředí Matlab.	48
C.3	Příklad implementace první kanonické formy v jazyce C.	49

Úvod

Outdoorové hry bývají často složeny ze stanoviště na kterých mají hráči plnit různé úkoly. Aby bylo možné tyto úkoly zadat a vyhodnotit jejich výsledek, je většinou nutné aby na stanovišti byl nějaký organizátor a stanoviště obsluhoval. Tyto úkoly jsou ale často poměrně prosté a není tak problém je automatizovat což může organizátory uvolnit k jiné činnosti. Outdoorové hry by navíc znatelně oživila aktivní komunikace mezi stanovišti, která by tak mohla i vytvořit prostor pro nové herní mechaniky.

Řada outdoorové her využívá různé podomácku vyrobené zařízení, které nějaký z organizátorů postavil za účelem konkrétní hry. Taková zařízení ale autora stojí velké množství času protože jej musí celé od základu navrhnout, vyrobit a navíc je jej pak schopen obsluhovat jen on. Navíc, je pak takové zařízení typicky použito jen na jedné nebo dvou her, po kterých jej autor buď, rozebere, nebo bezpečně uloží nekam, kde si jej náhodou všimne o deset let později při uklidu. V neposlední řadě bývají jakýmsi zlatým hřebem celé akce např. týdenního tábora a jejich kouzlo je především v odlišnosti od zbytku akce.

Z těchto důvodů padlo rozhodnutí na vývoj univerzálního automatického herního stanoviště, které by se dalo opakovaně použít na různé hry i ve větší počtu. Podstatnou součástí je pochopitelně i pokud možno co nejintuitivnější ovládání aby uživatelé nezdržovalo od zábavy.

1 Teoretická část

1.1 Potřeby herních zařízení

Outdoorové hry mají většinou nějaký příběh, který se dá vyprávět konkrétními úkoly na stanovištích. Na některých stanovištích proto musí být lidská obsluha, na jiných ale může být lidská obsluha s příběhového pohledu nežádoucí. Když má hráč například vyřadit automatický bezpečnostní systém je lidské obsluha stanoviště prostě poslední možnost. Podobná stanoviště proto bývají realizovány pomocí různých papírků a provázků. To určitě má své kouzlo ale i tak je u podobného stanoviště vodné mít obsluhu. Elektronické řešení podobných stanovišť by ale mohlo otevřít úplně nové možnosti.

Položme si tedy otázku jak by takové zařízení mohlo vypadat. Podstatný fakt je že prakticky všichni u sebe dnes mají chytrý telefon, čehož můžeme využít. Nemá proto velký význam aby toto zařízení suplovalo funkce telefonu. Např. grafický výstup typu display proto v podobném zařízení není potřeb, a v tomto směru už odvádí telefon naprosto dostatečnou práci. Pokud by tedy v rámci hry bylo potřeba například předat hráči nějaký text nebo obrázek, může jej zařízení poslat uživateli na telefon. Možnost propojení s telefonem je také velmi významná při nastavování hry. Díky telefonu totiž zařízení nepotřebuje uživatelské rozhraní přizpůsobené nastavování.

Mohlo by se zdát že herní stanoviště vlastně ani není potřeba a stačila by mobilní aplikace. Ale přestože je mobil ve hrách velmi dobře využitelný jsou aplikace na které jednoduše vhodný není. Pokud má hráč například ze stanoviště získat nějaký fyzický objekt, mobil neposlouží. Pro hráče a ani organizátory také nemusí zrovna komfortní před hrou zařizovat aby všichni kdo to potřebují měli nainstalovaný správný software natož svůj telefon nechávat někde na stanovišti. Navíc telefon, nebude dejme tomu na stanovišti v lese dobře viditelný atd. V neposlední řadě jde i o jistý cool efekt, který běžné zařízení jako mobil nebo třeba tablet neposkytne.

Zařízení určené primárně pro outdoorové hry bychom podle mě měli rozdělit na dvě skupiny, statické a dynamické, protože jsou na tyto skupiny kladeny výrazně jiné požadavky. Dynamická zařízení jsou ta která může uživatel pohodlně nosit s sebou. Do dynamických zařízení by se tak dal zařadit právě i telefon, ten však může být z různých důvodů nevhodný a proto i tato zařízení dává smysl navrhnout specificky pro hry. Statické zařízení je naopak zařízení u kterého se nepředpokládá že jej bude hráč nosit s sebou. Přesto by mělo být jednoduše přenositelné, není účelem postavit nemobilní hřiště, prostě jen není plánováno aby se se zařízením např. běhalo. Jde tedy o stanoviště které se jednoduše donese na své místo a při hře se s ním nehýbe. Takové zařízení by tedy mělo být dobře viditelné a splnovat požadavky které na něj

hra klade. Požadavky různých her mohou být ale dost rozdílně, častým požadavkem je něco uchovávat a např. po zadání hesla hráči vydat. Hra ale taky může vyžadovat aby bylo zařízení sto přehrát nějakou audio nahrávku nebo ji naopak nahrát. Z těchto důvodů považujeme za vhodnější zařízení rozdělit na základní řídicí jednotku která je samostatně funkční a použitelná při hře, ale ke které se dají jednoduše připojit moduly pro konkrétní herní mechaniky.

Z toho plyne otázka jaká funkcionalita je potřebná v základním zařízení? Asi žádný systém se kterým hráč přímo interaguje není nutný v každé hře. Jde tedy o to vybrat takové systémy, které svými nároky nepřevýší užitečnost při hrách. Ze zkušeností považujeme za nejzákladnější systém nějaký světelný výstup, ten dokáže většinu her velmi příjemně ozvláštnit. Pochopitelně je také většinou nezbytný nějaký uživatelský vstup, na což většinou stačí obyčejná tlačítka. Problém je ale určit jaké a kolik jich bude potřeba. Některé hry vyžadují třeba jen jedno ale takové aby se do něj dalo co nejpohodlněji praštit v běhu, protože je zrovna cílem ke stanovišti co nejrychleji doběhnout. Jiná hra ale může vyžadovat tlačítek víc, ale už není potřeba aby byli tak velké protože hráč při jejich používání nebude tak akční, ale bude třeba zadávat výsledek nějakého logického úkolu. Univerzálnější je tedy nepoužívat tlačítka ale nějaký systém který se dá softwarově přizpůsobit. Příkladem může být dotyková plocha která se dá softwarově rozdělit na různé oblasti sloužící jako tlačítka a i během hry se tak dá počet tlačítek měnit.

2 Návrh zařízení

Uživatelským požadavkem je statické zařízení sloužící jako herní stanoviště. Vyžaduje tedy mobilitu, ale jen v rámci transportu na místo hry a zpět, nikoliv v rámci samotné hry. Z toho plynou požadavky na velikost výsledného zařízení.

Zařízení bude mít dva světelné kruhy složené z šedesáti inteligentních RGB LED. Číslo šedesát jsme zvolili, protože se jedná o dostatečně jemné dělení, aby se dali dělat plynulé efekty. Zároveň jde o číslo, které koresponduje s hodinovým ciferníkem nebo stupnicí na kompasu. Jeden z kruhů bude radiální a druhý axiální. Axiální je na horní straně zařízení a slouží primárně jako odezva pro hráče na malou vzdálenost, např. při zadávání hesla. Radiální kruh je pak také v horní části zařízení a slouží naopak pro signalizaci na delší vzdálenost, takový maják.

Uvnitř axiálního světelného kruhu se bude nacházet tzv. tlaková plocha. Jedná se o ovládací prvek podobný dotykové ploše, s tím rozdílem, že je schopen měřit i sílu, která na něj působí.

Aby bylo stanoviště reálně použitelné při hře, musí celou hru vydržet na baterii. Není ojedinělé, aby měla outdoorová hra čtyři až pět hodin bez přestávky. Plus je nutná nějaká rezerva a čas na nastavování. Pochopitelně je čas, který zařízení zvládne běžet z baterie, silně závisí na činnosti, ale nebylo by zrovna ideální, kdyby byla baterka, nějak výrazně omezující. Výdrž na jedno nabití bych tedy chtěl směřovat alespoň na pět hodin.

Vzhledem k plánu připojovat moduly je nutné vyřešit, jak se to bude dělat. Bylo by ideální, kdyby si mohl uživatel říct co bude hrát za hru a podle toho si sám připojí moduly, které potřebuje. Tomuto určitě nechceme bránit, ale přímo to podporovat nese řadu problémů, jak ze strany konektoru a mechaniky, tak ze strany softwaru. Konektor by totiž musel být ideálně beznástrojově rozpojitelný a opětovně spojitelný a přitom dostatečně pevný, aby se zařízení mechanicky chovalo jako jeden celek. Takový konektor je ale poměrně složité udělat, tak aby byl spolehlivý, a tak jde v tuto chvíli spíš o hudbu budoucnosti. Ze softwarového pohledu jde pak o problém jak detekovat konkrétní modul a hlavně o otázku jak se chovat k modulům, které jsou potenciálně záměnné. Dejme tomu, že máme modul klávesnici a modul dvířka. Dvířka jsou původně navrženy primárně jako úložný prostor, díky detekci zavření je lze ale použít i jako velmi pohodlná tlačítka a v některých hrách se proto používají jen jako tlačítka. Potenciální modul klávesnice je ovšem jen suma tlačítek. Při vytváření konkrétní hry na míru modulům, které herní návrhář má zrovna k dispozici, je tento problém nepodstatný, protože sám návrhář rozhodne co má jak být. Ale ve chvíli, kdy jde o hru navrženou pro jinou kombinaci modulů, nastává problém jak rozhodnout, zda se dají dvířka použít místo klávesnice nebo ne. Abychom se všem těmto problémům alespoň prozatím vyhnuli, rozhodli jsme, že doplnění či výměna

modulu, půjde jen při servisním zásahu. Problém záměny modulů pak budeme řešit tím, že každá hra bude napsaná jen pro konkrétní sadu modulů.

Některé hry vyžadují tak velké herní území, že by na komunikaci mezi stanovišti už nestačila WiFi ani Bluetooth, které AHS jinak využívá ke komunikaci. Proto bude mít AHS možnost připojení k mobilní síti a tedy připojení k internetu. Tím se za prvé rozšíří dosah AHS, všude kde je mobilní pokrytí a také přibude další metoda jak se stanovištěm komunikovat. V rámci tohoto komunikačního modulu bude možné používat navíc i GNSS ¹. Vzhledem k faktu, že se přece jen nejedná o systém, který by využila většina her a zároveň je poměrně drahý, došli jsme k rozhodnutí mít jej jen jako doplnitelný modul. Protože se jedná o modul, který zprostředkovává komunikaci se světem, je pravděpodobné, že bude potřebovat převádět výrazně větší množství dat než běžný modul. Primárně z tohoto důvodu je tento modul připojen na samostatném konektoru. Potřebné antény budou už v základním zařízení, ale samotný modul spadá do doplňkové výbavy.

V řadě případu je užitečné mít možnost zvukové zpětné vazby. Ideální by bylo moci přehrávat libovolnou nahrávku, většinou ale stačí jednoduchý tón řekněme jako potvrzení zadaného hesla. Možnost přehrávat plnohodnotnou nahrávku proto odsouváme jako možný doplňkový modul a v základní jednotce pro jednoduchost postačí piezoměnič.

2.1 Struktura elektroniky základní jednotky

Elektronika je rozdělena na tři samostatné PCB. Jde o HlavníDesku, na které je většina elektroniky, o desku s hlavním uživatelským rozhraním (LedDeska) a o obslužnou desku s minimalistickým uživatelským rozhraním pro neherní obsluhu (MiniUI).

2.1.1 LedDesku

Jak název napovídá, na LedDesce se nacházejí oba světelné kruhy, mimo to je zde i elektronika pro snímání tlakové plochy, tedy LDC1x14 [1] a jeho snímací cívky. Právě snímání tlakové plochy je jeden z podstatných důvodů oddělení této desky od zbytku, zabere totiž docela dost prostoru.

- Axiální LED kruh z 60ti RGB LED WS2812
- Zadní radiální LED kruh z 60ti RGB LED WS2812
- LDC1614 nebo LDC1314 se čtyřmi snímacími cívkami pro snímání tlakové plochy
- konektor na propojení s HlavníDeskou

¹Global Navigation Satellite System

2.1.2 MiniUI

Kvůli aktuální představě mechanické konstrukce není úplně dobře možné mít toto minimalistické uživatelské rozhraní na HlavníDesce. Proto jsme se rozhodli jej vyseparovat na samostatnou destičku, na které bude jen pár tlačítek a dvě signalizační LED.

- RESET tlačítka
- BOOT tlačítka
- zapínací tlačítko
- dvě uživatelská tlačítka
- dvě uživatelské ledky

2.1.3 HlavníDeska

Na HlavníDesce je většina systému základního zařízení.

Protože máme dlouholeté zkušenosti s mikrokontrolery ESP32, rozhodovali jsme jen mezi konkrétními čipy z této rodiny. Aktuální plán na tvorbu vysoko-úrovňového návrhu konkrétních her je používat skriptovací jazyk JavaScript, považujeme za nevhodné provádět tento návrh na úrovni jazyka C/C++. Abychom tohoto mohli dosáhnout, musíme provozovat nějaký interpreter JavaScriptu a z aktuálně dostupných možností, považujeme za nejideálnější Jaculus [2]. Jaculus dělá řadu věcí paralelně a je tím pádem pro jeho chod ideální mít více jader procesoru, což aktuálně znamená tři možnosti ESP32, ESP32S2 a ESP32S3. ESP32S2 nemá Bluetooth, který bychom rádi měli k dispozici. ESP32S3 je téměř jen vylepšená verze staršího ESP32, oproti tomu mu sice chybí pár možností, ty ale v AHS nepotřebujeme využívat a naopak se nám hodí vyšší výpočetní výkon, rychlejší paměť a více piny. AHS tedy ovládá mikrokontroler ESP32S3.

Zdroj AHS je tvořen dvěma LiIon články 18650 v paralelním uspořádání. Paralelní uspořádání jsme zvolili, aby nebylo nutné řešit balancování článků.

Aby nebylo možné softwarově baterii podvybít, má AHS systém, který celé zařízení vypne v případě kdy dojde k vybití baterie pod 2.8V. Pochopitelně software by měl vybitou baterii zaznamenat mnohem dříve a chovat se podle toho, např. neumožnit spustit hru s baterií na napětím 3.0V.

Baterii je pochopitelně nutno nabíjet a bylo by velmi nepohodlné kvůli tomu muset vytahovat články ze zařízení. Proto je na HlavníDesce i nabíjecí elektronika. Navíc, aby se minimalizoval čas nabíjení zařízení, podporuje standard PowerDelivery a to až do napětí 21V.

Protože různé periferie vyžadují různá napájecí a komunikační napětí jsou na HlavníDesce hned čtyři napájecí větve.

- VCC, napětí baterie sloužící jako zdroj pro ostatní napájecí větve a pro napájení komunikačního modulu.
- Napětí 3.3[V] na napájení logické části celého základního zařízení.
- Napětí 5.0[V] pro LedDesku a externí moduly
- Napětí 1.8[V] pro napájení napěťových převodníků sloužících na komunikaci s komunikačním modulem

Napětí jednotlivých větví to u 3.3[V] a 1.8[V] tvořeno pomocí LDO. Na vytvoření pěti voltové větve je ale potřeba spínaný zdroj a to primárně ze dvou důvodů. Za prvé protože napětí baterie ze které se tato větev napájí má nižší napětí a je jej tedy třeba vyspínat na napětí vyšší. Za druhé tento zdroj poskytuje do systému mnohem větší proudy než druhé dvě větve a bylo by tedy vhodné, ho použít i v případě použití sériového řazení článků.

Na hlavníDesce je také řada konektorů sloužící pro připojení ostatních systémů. Jde o konektory na:

- propojení s LedDeskou
- připojení MiniUI
- komunikační modul (M2 Konektor umožňuje použít různé moduly)
- externí moduly
- USB-C (nabíjení a programování AHS)
- programátor

Do konektorů by se asi dal zařadit i držák na dva LiIon články 18650.

Za zmínku také stojí přítomnost piezoměniče pro jednoduchou zvukovou odezvu.

2.1.4 Propojení HlavníDesky a LedDesky

Mezi HlavníDeskou a LedDeskou je třeba převést napájení a několik signálů. Led-Deska vyžaduje na konektoru přítomnost dvou napájecích větví 5[V] pro světelné kruhy a 3.3[V] pro snímání tlakové plochy. Protože do LEDek může téct proud až 5A a být zároveň i docela rychle spínaný, považujeme za rozumné oddělit napájecím větvím zem. Oddělení je tedy provedeno už na konektoru HlavníDesky.

Na samotné propojení jsme se rozhodli použít FFC kabel s roztečí 0.5[mm]. Jedním vodičem takového kabelu zle vést proud maximálně 0.4[A] [3]. Protože ale potřebujeme dodat proud až 5A použijeme 13 vodičů vedle sebe jakožto nejmenší počet který přenesení požadovaný proud v rámci daných mezí.

Mimo napájení je tímto propojením veden i signál s daty pro světelné kruhy a I2C sběrnice s interruptem pro připojení čipu LDC1614 [1].

Vzhledem k počtu potřebných vodičů (konkrétně 32) jsme se rozhodli použít velmi běžný FFC konektor se 40 kontakty, s tím že zbylé kontakty se mohou hodit v budoucnu.

2.1.5 Modulový konektor

Modulovým konektorem je vedeno 5[V] jako napájení pro moduly a UART s interruptem pro komunikaci. Nad volbou komunikační sběrnice jsme strávili poměrně dost času přemýšlením. Původně jsme uvažovali o využití RS485 jakožto odolné sběrnice u které by v případě potřeby nemusel být problém ani delší kabel. RS485 má ale nevýhodu v tom že potřebuje dodatečný hardware, kterému bychom se hlavně na modulech rádi vyhnuli. Stejný problém nastel u CANu i USB které by navíc mělo výhodu kompatibility se velkým množstvím hotových zařízení.

V první uvažce o UARTu jsme jej nejprve zavrhlí kvůli potenciální náročnosti na přeposílání dat mezi moduly. Při standardním použití, bychom totiž moduly řadili za sebe. Prvnímu modulu by tak chodili data pro všechny ostatní moduly a musel by je přeposílat dál, což by stálo nezanedbatelné množství procesorového času. V jisté chvíli jsme ale narazili na nestandardní komunikaci pomocí UARTu implementované v projektu Servio [4]. Tato implementace používá UART jako sběrnici. Namísto standardního použití pro komunikace jeden s jedním tak může komunikovat jeden s více. Na tomto řešení je výhodné že nevyžaduje žádný dodatečný hardware a prakticky každý dnešní mikrokontroler je možné k této sběrnici velmi snadno připojit. Ve srovnání s RS485 je sice mnohem méně odolná proti rušení, ale uvnitř zařízení nebude linka vedena na víc jak malé desítky centimetru. Komunikace na delším kabelu je pak jednoduše nahraditelné bezdrátovou komunikací a není tak potřebné aby to nativně umožňovala tato sběrnice. Každopádně v případě potřeby delšího kabelu je možné navrhnout externí modul, který s této sběrnice velmi jednoduše udělá RS485. Alternativně by se pro komunikaci na delším kabelu dalo použít USB které je společně s nabíjením přivedeno na USB-C.

Všechny moduly jsou tedy připojeny na jeden RX pin AHS. Proto musí firmware AHS zajistit aby dva moduly nevysílali současně. Aby se zabránilo možným zkratům, jako ochranu má každý modul své piny UARTu připojeny přes rezistor 180- Ω . Interrupt pin modulů se naopak chová jako open-collector a na straně hlavního zařízení je na něj tedy připojen pull-up rezistor. Abychom alespoň trochu zvýšili odolnost linky proti rušení přidáme na přijímací stranu pull-up rezistor. Cílem je zvýšení komunikačního proudu aby se případný proud vyvolaný rušením dal jednodušeji zanedbat. V neposlední řadě jsou mají všechny piny na konektoru ESD ochranu.

2.1.6 Konektor programátor

Zařízení se dá jednoduše programovat přes USB-C, tento kanál se ale dá softwarově narušit a pro takové případy je tu konektor na programátor. Jde o šest plošek na které se programátor připojuje pomocí pogo-pinů. Programátor sice obsahuje jen

jednoduchou elektroniku která by mohla být i přímo v elektronice AHS ale ve většině případu by byla zbytečná. Ve chvíli kdy by byla potřeba tak je stejně nutná odborná obsluha a pro tu není problém použít programátor.

2.1.7 USB-C

Jako napájecí a programovací konektor je použito USB-C. Díky němu je možné podporovat standard PowerDelivery využitý pro zrychlení nabíjení. Konektor je ale použit i na pohodlnější programování zařízení, pez potřeby programátoru.

2.1.8 Konektor komunikačního modulu

Pro připojení komunikačního modulu jsme zvolili konektor M2 typ-B jakožto standard pro tyto moduly. Díky tomuto konektoru můžeme jednoduše připojit různé LTE a GNSS moduly.

Závěr

Shrnutí studentské práce.

2.2 Použité zdroje

Online články a dokumenty

11. KUBICA, Petr. *Jaculus - JavaScript runtime for embedded devices*. 2023. Dostupné také z: <https://jaculus.org>.
12. KONIARIK, Jan. *Servio*. 2023. Dostupné také z: <https://github.com/emsro/servio/tree/main>.
13. CORPORATION, Panasonic. *OVERCHARGE/OVERDISCHARGE/OVER-CURRENT SAFETY CIRCUITS*. 2007. Dostupné také z: https://web.archive.org/web/20101122230718/https://www.panasonic.com/industrial/includes/pdf/Panasonic_LiIon_Charging.pdf.

Katalogové listy důležitých součástek

1. *Úprava, odevzdávání a zveřejňování vysokoškolských kvalifikačních prací na VUT v Brně*. Brno, 2009. Dostupné také z: <https://www.vutbr.cz/uredni-deska/vnitri-predpisy-a-dokumenty/smernice-rektora-f34920/>. Směrnice rektora č. 2/2009.
7. *TMS320C6416T DSK: Technical Reference*. 2004. Dostupné také z: http://c6000.spectrumdigital.com/dsk6416/V3/docs/dsk6416_TechRef.pdf.
8. *TMS320C6000 DSP Enhanced Direct Memory Access (EDMA) Controller Reference Guide*. 2006. spru234c.
14. INCORPORATED, Texas Instruments. *BQ24179 I2C Controlled, 1- to 4-Cell, 5-A Buck-Boost Battery Charger with Dual-Input Selector*. 2022. Dostupné také z: https://www.ti.com/lit/ds/symlink/bq24179.pdf?ts=1698854511988&ref_url=https%253A%252F%252Fwww.ti.com%252Fpower-management%252Fbattery-management%252Fcharger-ics%252Fproducts.html.
15. INCORPORATED, Texas Instruments. *BQ298xyz Voltage, Current, Temperature Protectors with an Integrated High-Side NFET Driver for Fast/Flash Charging Single-Cell Li-Ion and Li-Polymer Batteries*. 2022. Dostupné také z: https://www.ti.com/lit/ds/symlink/bq2980.pdf?ts=1701078971409&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FBQ2980.
16. INCORPORATED, Texas Instruments. *TPS61088 10-A Fully-Integrated Synchronous Boost Converter*. 2021. Dostupné také z: https://www.ti.com/lit/ds/symlink/tps61088.pdf?ts=1698858563376&ref_url=https%253A%252F%252Fwww.google.com%252F.
17. INCORPORATED, Diodes. *HIGH-PERFORMANCE USB PD SINK CONTROLLER*. 2023. Dostupné také z: <https://www.diodes.com/assets/manuals/AP33772.pdf>.
18. INCORPORATED, Texas Instruments. *LDC1612, LDC1614 Multi-Channel 28-Bit Inductance to Digital Converter (LDC) for Inductive Sensing*. 2018. Dostupné také z: <https://1url.cz/rudLI>.
19. COPYRIGHT, DEsspressif Systems. *ESP32-S3-WROOM-1*. 2023. Dostupné také z: https://www.espressif.com/sites/default/files/documentation/esp32-s3-wroom-1_wroom-1u_manual_en.pdf.
20. JUSHUC. *2304231848 JUSHUO-AFC07-S32FCC-00 C11061*. 2004. Dostupné také z: https://manual.lcsc.com/lcsc/2304231848_JUSHUO-AFC07-S32FCC-00_C11061.pdf. FFC konektor.

21. SOLUTIONSLIMITED, SIMCom Wireless. *SIM7600X-M2HardwareDesign*. 2020.
Dostupné také z: https://files.waveshare.com/upload/0/02/SIM7600X-M2_Hardware_Design_V1.01.pdf.

Seznam symbolů a zkratek

AHS	Automatické Herní Stanoviště
LDO	Low-dropout regulator - regulátor napětí s nízkým úbytkem
f_{vz}	Flexible flat cable - plochý ohební kabel
GNSS	Global Navigation Satellite System - ne jen GPS (Global Positioning System)
LTE	3GPP Long Term Evolution - dlouhodobý vývoj 3GPP
PCB	- deska plošných spojů

Seznam příloh

A	Některé příkazy balíčku <code>thesis</code>	43
A.1	Příkazy pro sazbu veličin a jednotek	43
A.2	Příkazy pro sazbu symbolů	43
B	Druhá příloha	45
C	Příklad sazby zdrojových kódů	47
C.1	Balíček <code>listings</code>	47
D	Obsah elektronické přílohy	51

A Některé příkazy balíčku thesis

A.1 Příkazy pro sazbu veličin a jednotek

Tab. A.1: Přehled příkazů pro matematické prostředí

Příkaz	Příklad	Zdroj příkladu	Význam
<code>\textind{...}</code>	β_{\max}	<code>\$_\beta_\textind{max}\$</code>	textový index
<code>\const{...}</code>	U_{in}	<code>\$_\textconst{U}_\textind{in}\$</code>	konstantní veličina
<code>\var{...}</code>	u_{in}	<code>\$_\textvar{u}_\textind{in}\$</code>	proměnná veličina
<code>\complex{...}</code>	\textit{u}_{in}	<code>\$_\textcomplex{u}_\textind{in}\$</code>	komplexní veličina
<code>\vect{...}</code>	\mathbf{y}	<code>\$_\textvect{y}\$</code>	vektor
<code>\mat{...}</code>	\mathbf{Z}	<code>\$_\textmat{Z}\$</code>	matice
<code>\unit{...}</code>	kV	<code>\$_\textunit{kV}\$</code> či <code>\unit{kV}</code>	jednotka

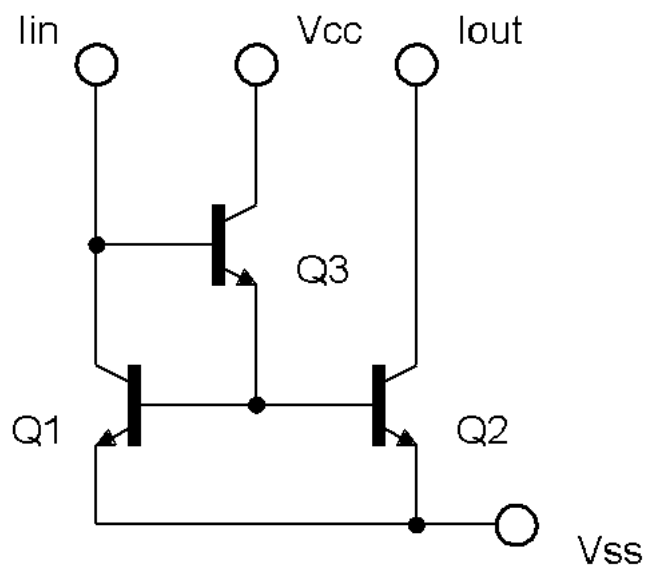
A.2 Příkazy pro sazbu symbolů

- `\E`, `\eul` – sazba Eulerova čísla: e ,
- `\J`, `\jmag`, `\I`, `\imag` – sazba imaginární jednotky: j , i ,
- `\dif` – sazba diferenciálu: d ,
- `\sinc` – sazba funkce: sinc ,
- `\mikro` – sazba symbolu mikro stojatým písmem¹: μ ,
- `\uppi` – sazba symbolu π (stojaté řecké pí, na rozdíl od `\pi`, což sází π).

Všechny symboly jsou určeny pro matematický mód, vyjma `\mikro`, jenž je použitelný rovněž v textovém módu.

¹znak pochází z balíčku `textcomp`

B Druhá příloha



Obr. B.1: Zlepšené Wilsonovo proudové zrcadlo.

Pro sazbu vektorových obrázků přímo v \LaTeX je možné doporučit balíček `TikZ`. Příklady sazby je možné najít na `TikZexample`. Pro vyzkoušení je možné použít programy `QTikz` nebo `TikzEdt`.

C Příklad sazby zdrojových kódů

C.1 Balíček listings

Pro vysázení zdrojových souborů je možné použít balíček `listings`. Balíček zavádí nové prostředí `lstlisting` pro sazbu zdrojových kódů, jako například:

```
\section{Balíček lstlistings}
Pro vysázení zdrojových souborů je možné použít
  balíček \href{https://www.ctan.org/pkg/listings}%
  {\texttt{listings}}.
Balíček zavádí nové prostředí \texttt{lstlisting} pro
  sazbu zdrojových kódů.
```

Podporuje množství programovacích jazyků. Kód k vysázení může být načítán přímo ze zdrojových souborů. Umožňuje vkládat čísla řádků nebo vypisovat jen vybrané úseky kódu. Např.:

Zkratky jsou sázeny v prostředí `acronym`:

```
6 \begin{acronym}[KolikMista]
```

Šířka textu volitelného parametru `KolikMista` udává šířku prvního sloupce se zkratkami. Proto by měla být zadávána nejdelší zkratka nebo symbol. Příklad definice zkratky `symfvz!` je na výpisu C.1.

Výpis C.1: Ukázka sazby zkratek

Ukončení seznamu je provedeno ukončením prostředí:

26

```
27 \end{acronym}
```

Poznámka k výpisům s použitím volby jazyka `czech` nebo `slovak`:

Pokud Váš zdrojový kód obsahuje znak spojovníku `-`, pak překlad může skončit chybou. Ta je způsobená tím, že znak `-` je v českém nebo slovenském nastavení balíčku `babel` tzv. aktivním znakem. Přepněte znak `-` na neaktivní příkazem `\shorthandoff{-}` těsně před výpisem a hned za ním jej vraťte na aktivní příkazem `\shorthandon{-}`. Podobně jako to je ukázáno ve zdrojovém kódu šablony.

Na výpisu C.2 naleznete příklad kódu pro Matlab, na výpisu C.3 zase pro jazyk C.

Výpis C.2: Příklad Schur-Cohnova testu stability v prostředí Matlab.

```
1 %% Příklad testování stability filtru
2
3 % koeficienty polynomu ve jmenovateli
4 a = [ 5, 11.2, 5.44, -0.384, -2.3552, -1.2288];
5 disp('Polynom:'); disp(poly2str(a, 'z'))
6
7 disp('Kontrola pomocí kořenů polynomu:');
8 zx = roots(a);
9 if( all( abs( zx) < 1))
10     disp('System je stabilní')
11 else
12     disp('System je nestabilní nebo na mezí stability');
13 end
14
15 disp(' '); disp('Kontrola pomocí Schur-Cohn:');
16 ma = zeros( length(a)-1,length(a));
17 ma(1,:) = a/a(1);
18 for( k = 1:length(a)-2)
19     aa = ma(k,1:end-k+1);
20     bb = fliplr( aa);
21     ma(k+1,1:end-k+1) = (aa-aa(end)*bb)/(1-aa(end)^2);
22 end
23
24 if( all( abs( diag( ma.'))))
25     disp('System je stabilní')
26 else
27     disp('System je nestabilní nebo na mezí stability');
28 end
```


Výpis C.3: Příklad implementace první kanonické formy v jazyce C.

<i>// první kanonická forma</i>	1
short fxdf2t(short coef[][5], short sample)	2
{	3
static int v1[SECTIONS] = {0,0}, v2[SECTIONS] = {0,0};	4
int x, y, accu;	5
short k;	6
	7
x = sample;	8
for (k = 0; k < SECTIONS; k++){	9
accu = v1[k] >> 1;	10
y = _sadd(accu, _smpy(coef[k][0], x));	11
y = _sshl(y, 1) >> 16;	12
	13
accu = v2[k] >> 1;	14
accu = _sadd(accu, _smpy(coef[k][1], x));	15
accu = _sadd(accu, _smpy(coef[k][2], y));	16
v1[k] = _sshl(accu, 1);	17
	18
accu = _smpy(coef[k][3], x);	19
accu = _sadd(accu, _smpy(coef[k][4], y));	20
v2[k] = _sshl(accu, 1);	21
	22
x = y;	23
}	24
return (y);	25
}	26

D Obsah elektronické přílohy

Elektronická příloha je často nedílnou součástí semestrální nebo závěrečné práce. Vkládá se do informačního systému VUT v Brně ve vhodném formátu (ZIP, PDF ...).

Nezapomeňte uvést, co čtenář v této příloze najde. Je vhodné okomentovat obsah každého adresáře, specifikovat, který soubor obsahuje důležitá nastavení, který soubor je určen ke spuštění, uvést nastavení kompilátoru atd. Také je dobře napsat, v jaké verzi software byl kód testován (např. Matlab 2018b). Pokud bylo cílem práce vytvořit hardwarové zařízení, musí elektronická příloha obsahovat veškeré podklady pro výrobu (např. soubory s návrhem DPS v Eagle).

Pokud je souborů hodně a jsou organizovány ve více složkách, je možné pro výpis adresářové struktury použít balíček `dirtree`.

```
/ .....kořenový adresář přiloženého archivu
├── logo .....loga školy a fakulty
│   ├── BUT_abbreviation_color_PANTONE_EN.pdf
│   ├── BUT_color_PANTONE_EN.pdf
│   ├── FEEC_abbreviation_color_PANTONE_EN.pdf
│   ├── FEKT_zkratka_barevne_PANTONE_CZ.pdf
│   ├── UTKO_color_PANTONE_CZ.pdf
│   ├── UTKO_color_PANTONE_EN.pdf
│   ├── VUT_barevne_PANTONE_CZ.pdf
│   ├── VUT_symbol_barevne_PANTONE_CZ.pdf
│   └── VUT_zkratka_barevne_PANTONE_CZ.pdf
├── obrazky .....ostatní obrázky
│   ├── soucastky.png
│   ├── spoje.png
│   ├── ZlepseneWilsonovoZrcadloNPN.png
│   └── ZlepseneWilsonovoZrcadloPNP.png
├── pdf .....pdf stránky generované informačním systémem
│   ├── student-desky.pdf
│   ├── student-titulka.pdf
│   └── student-zadani.pdf
├── text .....zdrojové textové soubory
│   ├── literatura.tex
│   ├── prilohy.tex
│   ├── reseni.tex
│   ├── uvod.tex
│   ├── vysledky.tex
│   ├── zaver.tex
│   └── zkratky.tex
├── sablona-obhaj.tex .....hlavní soubor pro sazbu prezentace k obhajobě
├── sablona-prace.tex .....hlavní soubor pro sazbu kvalifikační práce
└── thesis.sty .....balíček pro sazbu kvalifikačních prací
```