

Vyšší odborná škola
a Střední průmyslová škola elektrotechnická
Plzeň, Koterovská 85

DLOUHODOBÁ MATURITNÍ PRÁCE S OBHAJOBOU

Téma: ESP Craft – propojení herního světa s realitou

Autor práce: Jiří Švihla
Třída: 4.P
Obor studia: 26-41-M/01 Elektrotechnika (zaměření: Internet věcí)
Vedoucí práce: Pavel Koenig
Dne: 16. 4. 2021

Hodnocení:



Vyšší odborná škola
a Střední průmyslová škola elektrotechnická
Plzeň, Koterovská 85

Zadání dlouhodobé maturitní práce

Žák: Jiří Švihla
Třída: 4. P
Studijní obor: 26-41-M/01 Elektrotechnika
Zaměření: Internet věcí (IoT)
Školní rok: 2020 - 2021

Téma práce: ESP Craft – propojení herního světa s realitou

Pokyny k obsahu a rozsahu práce:

1. Seznamte se s vývojovou platformou (ESP8266/ESP32)
2. Seznamte se softwarem KiCAD pro návrh PCB
3. Seznamte se s problematikou přenosu informací mezi vývojovou platformou a externím serverem přes internet
4. Seznamte se s problematikou přenosu informací mezi programem Minecraft® a externím serverem přes internet
5. Navrhněte a vyrobte vývojovou desku založenou na zvolené vývojové platformě
6. Vytvořte plugin pro samotnou hru Minecraft®, který bude zprostředkovávat komunikaci mezi hrou a externím serverem
7. Naprogramujte firmware pro vyrobenou vývojovou desku zprostředkovávající komunikaci mezi deskou a externím serverem

Plán konzultací:

20.10.2020	Zprovoznění externích serverů – Minecraft® a komunikační server
17.11.2020	Prototyp PCB – základní návrh, vývoj rozšíření pro Minecraft®
5.1.2021	Prototyp PCB – koncový návrh, výroba, vývoj FW pro desku
2.2.2021	Zprovoznění výrobku, dokumentace

Požadavek na počet vyhotovení maturitní práce: 2 výtisky

Termín odevzdání: 16. dubna 2021

Čas obhajoby: 15 minut

Vedoucí práce: Pavel Koenig

Projednáno v **katedře ODP** a schváleno ředitelkou školy.

V Plzni dne: 30. října 2020

Ing. Naděžda Mauleová, MBA, v.r.

Anotace

Cílem této dlouhodobé maturitní práce je za pomoci profesionálních vývojových programů navrhnout a vyrobit vývojovou desku určenou pro začátečníky i pokročilé s možností připojení k bezdrátové síti. Dalším úkolem je vyvinout potřebné rozšíření pro samotnou hru a potřebný firmware pro samotnou desku. Koncovým úkolem je vše zprovoznit a zhodnotit přínosy a možnosti tohoto projektu, včetně vypracování potřebné dokumentace.

„Prohlašuji, že jsem tuto práci vypracoval samostatně a použil literárních pramenů a informací, které cituji a uvádím v seznamu použité literatury a zdrojů informací.“

V Plzni dne: Podpis:

Poděkování

Tímto bych chtěl poděkovat vedoucímu nviás z.s. Pavlu Koenigovi za financování tohoto projektu, Jiřímu Siterovi ml. za pomoc při vývoji rozšíření pro hru, třídní učitelce Ing. Šárce Blechové Ph. D. za podporu při zpracování a rodině a přátelům za psychologickou podporu.

Obsah

Anotace	2
Poděkování	3
Obsah	4
Úvod	5
1. Cíle a požadavky	6
1.1. IoT node (vývojová deska)	6
1.2. Rozšíření pro hru	6
1.3. Komunikace	6
2. Návrhový software	7
2.1. KiCAD EDA.....	7
2.2. Visual Studio Code a Arduino.....	10
3. Vybrané komponenty a SW	11
3.1. Vybrané komponenty pro HW	11
3.2. Komunikace mezi hrou a deskou (SW)	12
4. Funkce (HW část)	13
4.1. MCU	13
4.2. Power Delivery controller	14
4.3. USB-C USB 2.0 interface	15
4.4. Napájecí kaskáda – 3,3V @ 3A.....	16
4.5. Rozhraní a periferie na desce.....	17
5. Funkce (SW část)	19
6. Plugin (rozšíření) pro hru a komunikace	20
7. Výrobek a závěr	22
8. Použité zdroje	25
9. Seznam příloh	26

Úvod

Jako programátor a zároveň hráč videoher vnímám jakoukoliv možnost rozšíření hry jako nabádání vývojářů stylem „Tady máte virtuální pískoviště a tady hmotu na bábovičky. Hrajte si.“

Takové hry existují již od dob prvních domácích počítačů a stále se vyvíjejí, ačkoliv nezáří na polích mezi příběhovými RPG jako je např. série Assassin's Creed, česká Mafia či Cyberpunk 2077. Občas jejich hvězdy přece jen vystoupí nad horizont a jasně zazáří. Zmínil bych hlavně dvě české perly – Space engineers a Factorio. Na rozdíl od podpory modifikací v příběhových hrách staví primárně na kreativě jejích hráčů a jejich koncepce je hluboce provázána s modifikovatelností téměř čehokoliv.

Mezi těmito poli ovšem stojí jeden legendární titul stavějící na obou základnách, dovolující vám změnit cokoliv. Virtuální kostičkový svět bez hranic, od kterého se všichni učili více než deset let – Minecraft.

Minecraft je hra, která přinesla nevídané možnosti svým hráčům: Cestu hrou si budete sami. Samotné dohrání příběhové části záleží čistě na vás. Záleží jen na vaší kreativě. V podstatě dostanete svět z virtuálního Lega. Postavte si dům, hrad, létající pevnost. Záleží jen na vás.

Hra ovšem obsahuje i základní elektroniku, tzv. „Redstone“ (v čj „Rudit“), která dovoluje zapojení jednoduchých, primárně digitálních obvodů, které lze kombinovat do složitějších bloků. Ve hře lze postavit dokonce i počítač (ano, CPU, ROM, RAM atd...), pokud jste ochotni jej stavět úplně z diskrétních komponent.

Ve hře je ale schován i jeden složitější blok dovolující přímé programování a napojení na tento elektronický systém. Příkazový blok, který v podstatě emuluje hráče. Má podobné možnosti jako hráč, který napíše do příkazového řádku příkaz ke změně herního módu, nalezení nejbližší vesnice atd... Díky svému napojení na herní elektronický systém ovšem dovoluje velkou spoustu věcí automatizovat. Společně s módy (které lze jednoduše tvořit v jazyce Java) tvoří programovatelnou páteř hry.

Kdokoliv, kdo pronikne trochu hlouběji do této hry rychle zjistí, že virtuální svět s takovými možnostmi lze použít téměř k čemukoliv. Proč ji tedy nevyužít k výuce?

Již zde byly pokusy o využití Minecraftu jako výukové pomůcky (dokonce existuje verze Minecraft: Education edition přidávající základy fyziky, chemie apod. do hry) a v této práci jej využiji k výuce základů elektroniky, programování a automatizace (alias „úvod do Internetu věcí“).

Tato práce je tvořena ve spolupráci s [nvias z.s.](#)



1. Cíle a požadavky

1.1. IoT node (vývojová deska)

Jedním z cílů projektu je navržení a výroba nové/známé vývojové desky dle níže uvedených požadavků:

- Zařízení by mělo mít bezdrátovou konektivitu (nejlépe **WiFi/Bluetooth**)
- Zařízení by mělo mít **kompaktní rozměry** (cca do 10x10x5cm)
- Zařízení by mělo být jednoduše připojitelné k počítači a přímo programovatelné (bez převodníku/programátoru, nejlépe připojitelné přes USB Micro B / **USB C**, jelikož jsou nejrozšířenější a kompaktní)
- Bylo by praktické, kdyby zařízení umělo ovládat **Power Delivery** (možnost kvalitních nabíječek a powerbank zvýšit výstupní napětí nad 5V pro přenesení vyššího výkonu). Lze jej tím pádem použít jako částečně regulovatelný zdroj.
- Zařízení by mělo zvládat základní funkce nodu v IoT (**měření teploty, senzor světla, ...**)
- K zařízení by měly být **jednoduše připojitelné periferie** rozšiřující možnosti základní desky.
- Zařízení by mělo svým designem zapadat do **stylu hry Minecraft®** pro kterou je navrženo

1.2. Rozšíření pro hru

Rozšíření pro hru by mělo být jednoduše ovladatelné a schopné přenášet signály mezi deskou a vnitřní hrou (virtuálním kostičkovým světem). Zároveň by jej mělo být možno jednoduše nastavit na připojení k libovolné desce.

Hra Minecraft® používá pro práci s elektronikou („Redstonem“) pouze napěťové informace, přenášený proud (jako v elektronice) se nebere v potaz. Napětí redstonového vodiče může nabývat hodnot 0-15 a každý blok vodiče se snižuje o 1, tj. první blok drátu má napětí 15, druhý 14, třetí 13 atd...

Rozšíření pro hru by tedy mělo být schopno přenášet těchto 16 hodnot napětí mezi hrou a deskou.

1.3. Komunikace

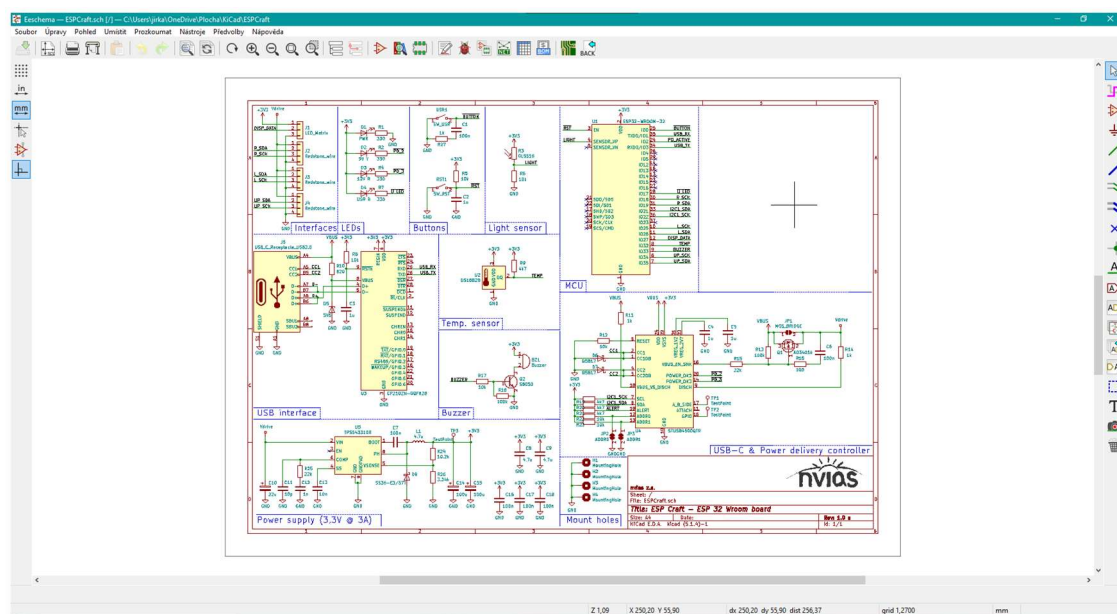
Komunikace by měla být realizována buď přímo mezi deskou a serverem, na kterém je hostována hra s rozšířením, nebo přes komunikační server. Komunikační protokol by **mělo být možné** v případě potřeby **zabezpečit alespoň jednoduchým šifrováním**, ačkoliv jej pro používání není třeba. Zároveň by bylo vhodné použít dobře zdokumentovaný, nejlépe průmyslový komunikační protokol.

2. Návrhový software

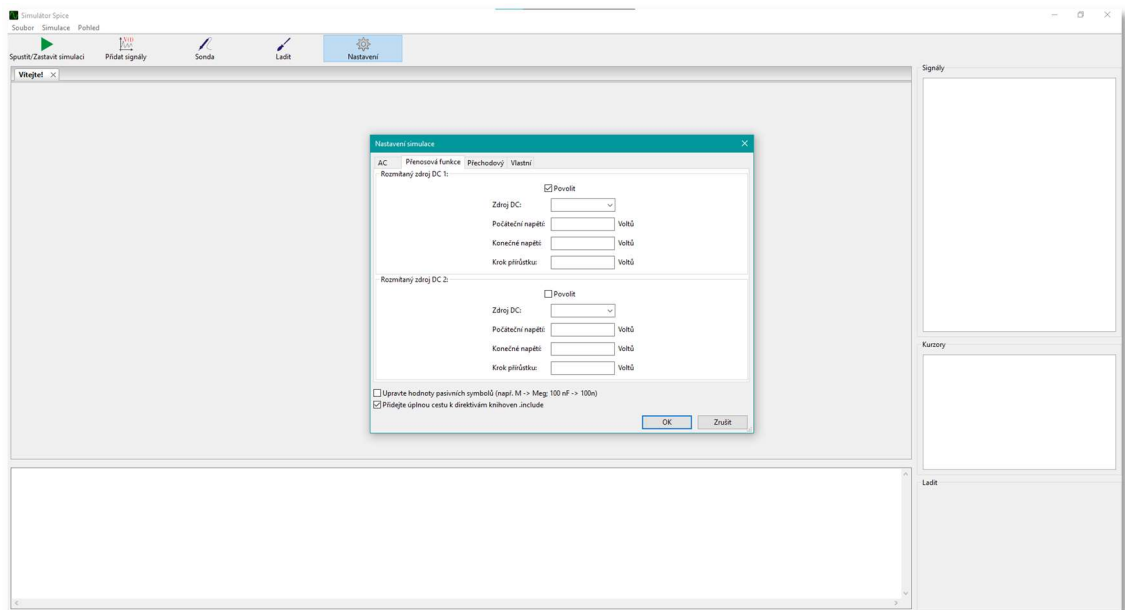
2.1. KiCAD EDA

KiCAD EDA je software pro návrh a vývoj elektronických zařízení. Samotná aplikace je složena ze tří hlavních částí – Eeschema, Pcbnew a GerbView.

Eeschema je program pro navrhování elektrotechnických schémat a generování netlistů. Obsahuje i rozhraní pro simulační program NGSPICE, takže jej lze použít i pro simulace jednoduchých obvodů. Eeschema má velmi rozsáhlou knihovnu součástek (symbolů) – přes 12 tis. již po instalaci a lze vytvářet další / měnit či dodávat další od komunity). Ke každé součástce je pak třeba přiřadit tzv. „footprint“, tedy rozložení vývodů z databáze, či také vytvořit / upravit / importovat vlastní. Pro simulace jsou ovšem dodány pouze základní součástky – pasivní komponenty, zdroje signálu apod., takže je třeba si ke složitějším schémátům dohledat SPICE modely. To se ukázalo jako zásadní nevýhoda (oproti např. NI Multisim), avšak oproti jiným simulačním programům je zdarma a lze v něm zařízení kompletně vytvořit od návrhu přes simulace až po samotný výsledek. Také v něm nelze spustit simulaci v reálném čase (na rozdíl od Multisimu, který je na tom založen), avšak lze nastavit přesnost simulace i čas, ve kterém chceme vidět aktuální výsledky. Navíc je i přesnější než Multisim a kompletně zdarma.

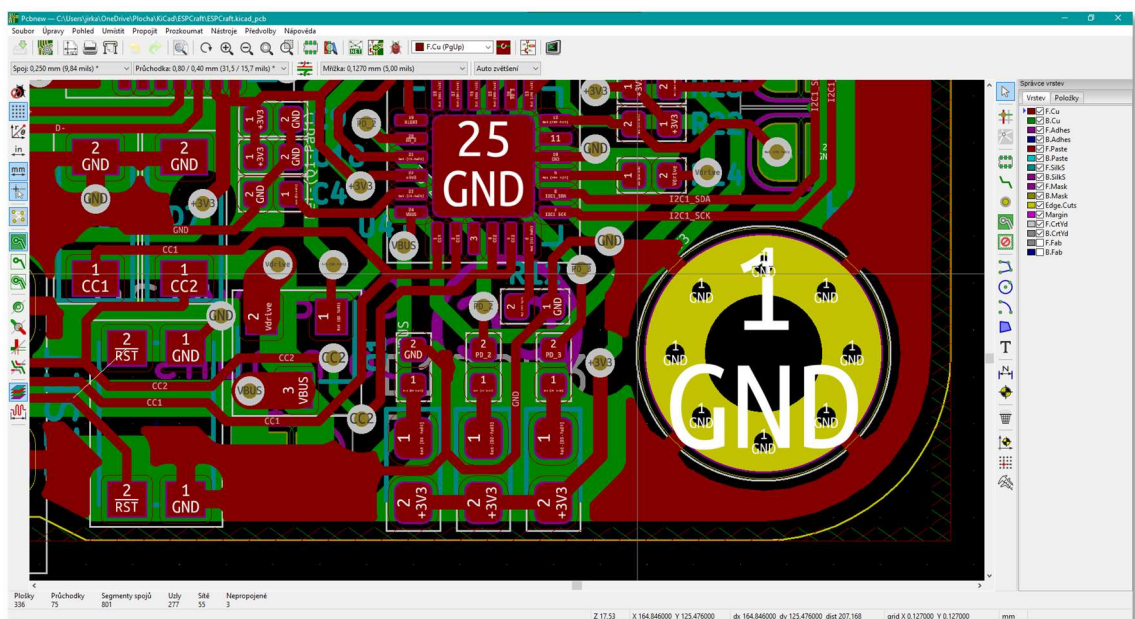


Obr. 2.1.1 – GUI programu Eeschema [Zdroj: vlastní]

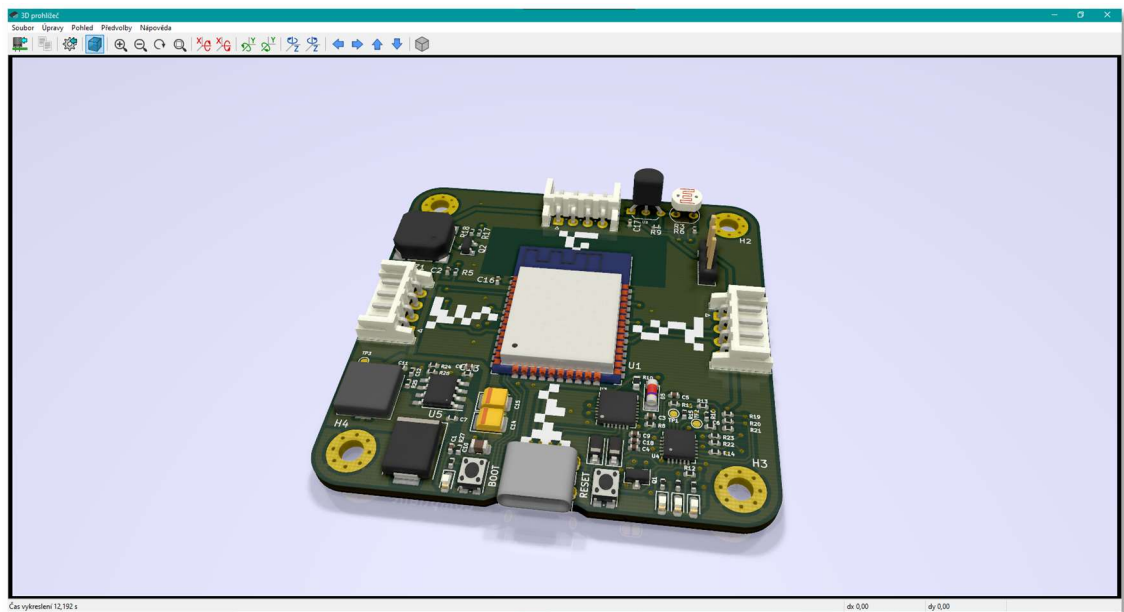


Obr. 2.1.2 – Nastavení simulátoru [Zdroj: vlastní]

Pcbnew je druhá část samotného balíčku, ve které lze navrhovat desku plošných spojů a rozložení součástek. Lze do něj importovat z Eeschemy netlist (seznam propojů a součástek). Pcbnew pak automaticky naimportuje potřebné součástky a nastaví propoje. Autorouter pak ani nedovolí zkrat více sítí. I kdyby k takovéto situaci došlo (např. přemístěním vývodů součástky na již položenou cestu), lze tyto (a další – typu malá izolační mezera, příliš slabá cestička apod.) chyby odhalit pomocí kontroly DRC. Nám jen zbývá propojit předznačené spoje cestičkami na desce (lze nastavit počet vrstev dle potřeby) a umístit součástky. Poté se můžeme podívat na výsledek ve 3D prohlížeči.

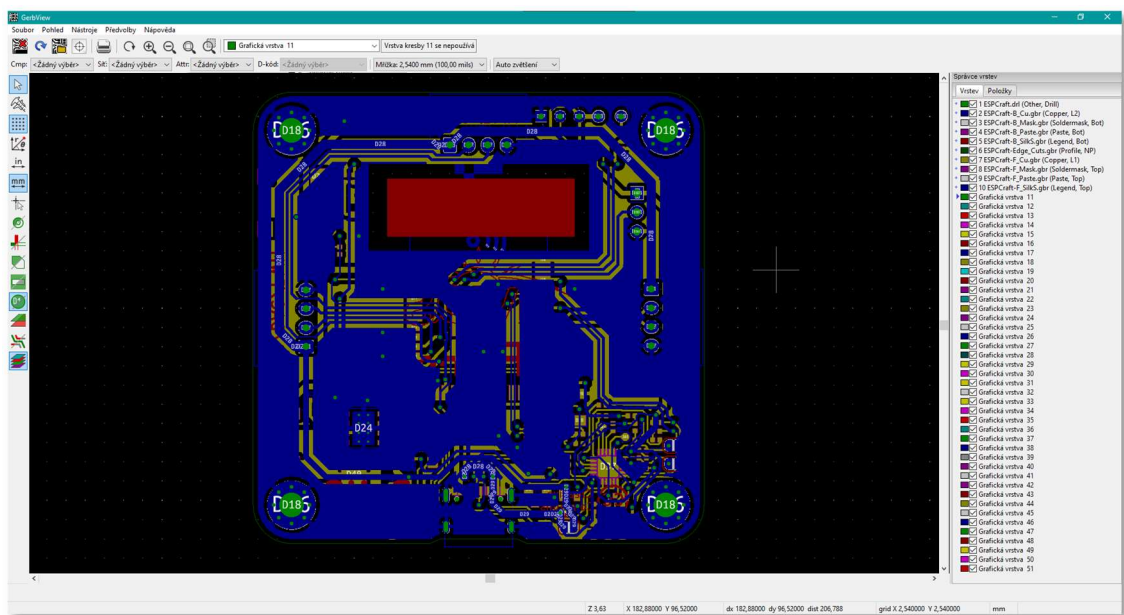


Obr. 2.1.3 – Rozhraní Pcbnew [Zdroj: vlastní]



Obr. 2.1.4 – 3D vizualizace desky (zapnuté RTX) [Zdroj: vlastní]

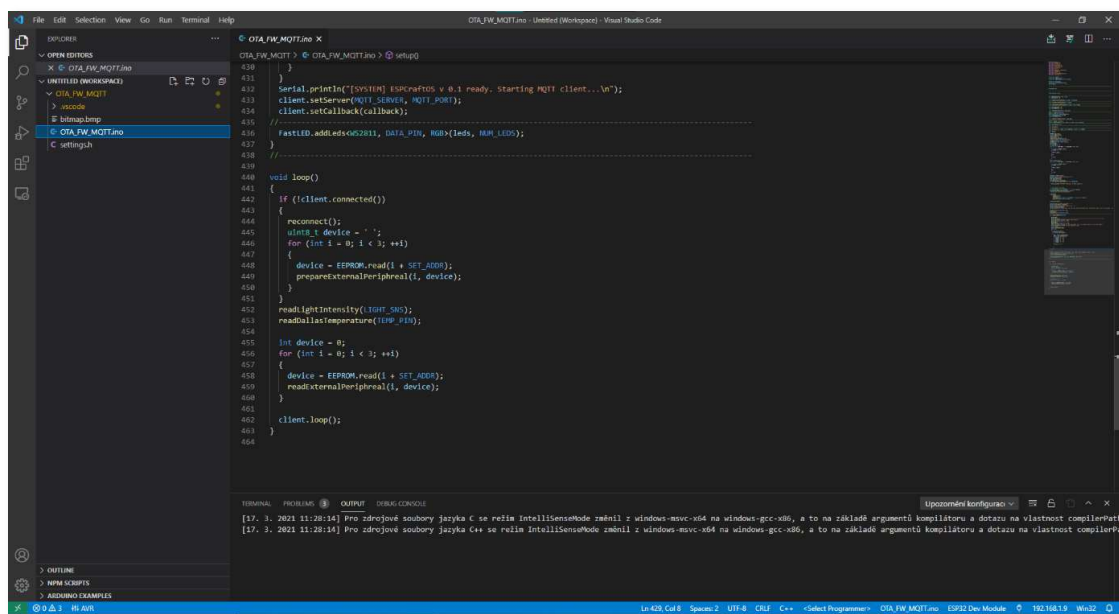
GerbView je tím, co od názvu lze očekávat – prohlížečem Gerber dat. Částečně z nich dokáže zpětně rekonstruovat návrh desky upravitelný v Pcbnew.



Obr. 2.1.5 – Rozhraní GerbView [Zdroj: vlastní]

2.2. Visual Studio Code a Arduino...

Visual Studio Code je nenáročný a otevřený IDE vyvíjený firmou Microsoft. Vzniklo v reakci na popularitu programů jako PyCharm, Atom a podobných. Obsahuje vlastní integrovaný „marketplace“ pro zjednodušení instalace rozšíření, rozhraní pro debugger či možnost integrace Gitu (lze tak pracovat s repositáři přímo v IDE). Po instalaci samotné IDE sotva umí podbarvovat kód a nějaké to automatické dokončování (IntelliSense je dodáván již v základu, číslování řádků a code folding je implicitně zapnut). A to nejdůležitější – prostředí je v temném kabátku!



Obr. 2.2.1 – GUI programu Visual Studio Code (otevřena část zdrojového kódu MP) [Zdroj: vlastní]

Osobně ve Visual Studio Code píšu kódy v HTML, CSS, PHP, PYTHON 3, C++, C#, Wiring (Arduino C++), SC/MP ASM, Z80 ASM a JavaScript. Knihovna podporovaných jazyků je samozřejmě mnohonásobně větší a stále roste, stejně jako možnosti i počet jednotlivých rozšíření. Možnost použití jednoho intuitivního, multiplatformního a jednoduchého (a otevřeného) IDE, které si programátor sám sestaví dle vlastních představ je skvělou volbou pro každého programátora.

3. Vybrané komponenty a SW

3.1. Vybrané komponenty pro HW

Řešení je navrženo s ohledem na všechny body zadání (včetně doporučení). Deska díky použitelnému řídicímu modulu ESP32-WROOM-32 disponuje bezdrátovou konektivitou WiFi 2,4GHz (b,g,n) a Bluetooth. ESP32 disponuje dvoujádrovým 32-bit RISC procesorem Xtensa(R) LX6, vysokým výkonem a možností programovat ve Wiringu (Arduino IDE), C++ (originální IDE od Espressif) či MicroPython. Zároveň disponuje (pouze) softwarovým PWM na všech pinech, dvěma 8-kanálovými 12-bit AD převodníky (! POUZE ADC_CH1 lze využívat společně se zapnutou WiFi) a dvěma 8-bit DA převodníky.

Pro komunikaci mezi ESP32 a PC je použit USB-UART bridge ovladač Silabs CP2102 s přenosovou rychlostí až 1Mbaud, který je také dobře znám bastlířské komunitě.

Napájení a komunikace s PC je realizována konektorem USB-C (ačkoliv data lze stáhnout do desky i přes síť – OTA programming). O ovládání proudového omezení zdroje a napájecího napětí pro desku se stará čip STUSB4500 od ST Microelectronics, který podporuje plynulé řízení napětí zdroje i nastavení proudové pojistky. Ovládání je řešeno přes HW I2C (TWI) port ESP32.

Palubní napětí 3,3V zařizuje napájecí kaskáda s TPS54331DR (nebo DDAR pro lepší chlazení) s maximálním výstupním proudem 3A a vysokou účinností.

Deska také disponuje měřením teploty (v případě použití staršího typu ESP32 i v samotném modulu) pomocí čidla DS18B20 od firmy Dallas semi. a možností měřit intenzitu světla pomocí napěťového děliče s fotorezistorem (nejedná se o nic závratně přesného, ale pro základní funkcionality postačí). Zároveň lze produkovat jednoduché melodie pomocí vestavěného reproduktoru (bzučáku)

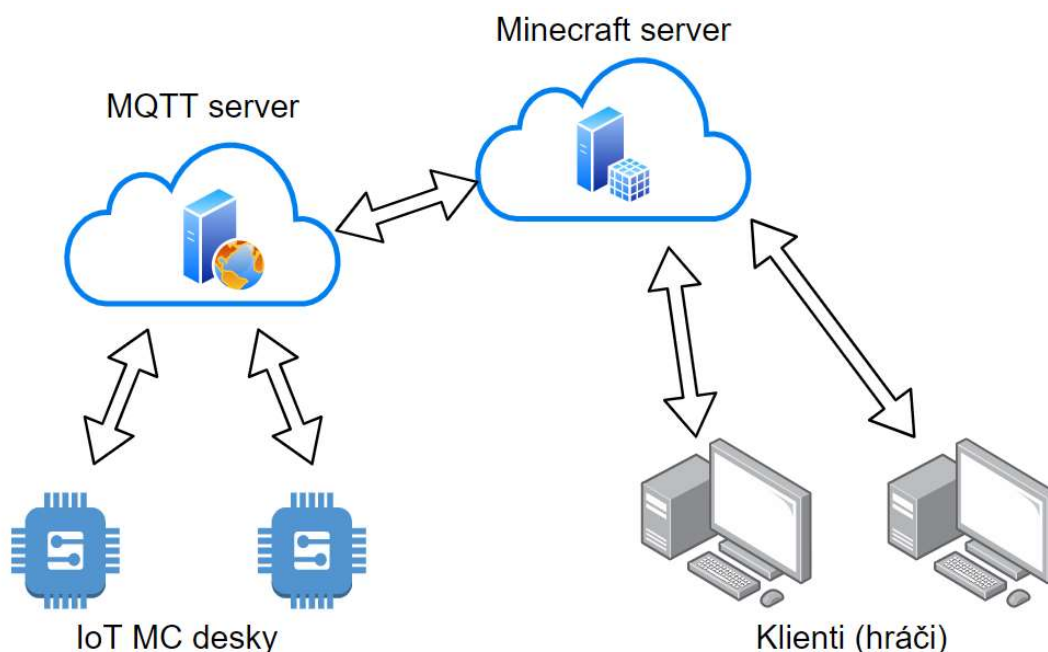
Na pravé straně od USB-C konektoru je umístěno resetovací tlačítko a stavové LED diody (napájení a dvě LED pro indikaci Power Delivery). Na levé straně je umístěno uživatelské (programovatelné) tlačítko a LED dioda.

Komunikace a napájení periférií lze uskutečnit připojením do tří portů na levé, pravé a horní hraně desky. Lze připojit buď desky s koncovými perifériemi, či rozšiřující moduly s více možnostmi.

3.2. Komunikace mezi hrou a deskou (SW)

Komunikace mezi serverem se hrou a samotnou deskou je řešena pomocí externího komunikačního serveru protokolem MQTT. Jedná se o jednoduchý, standardizovaný a dobře známý protokol vhodný pro komunikaci IoT nodů i s možnostmi zabezpečení komunikace.

O propojení hry a komunikačního serveru se stará plugin (rozšíření) pro samotnou hru běžící na straně serveru (-> netřeba jej instalovat do hry klienta/hráče)

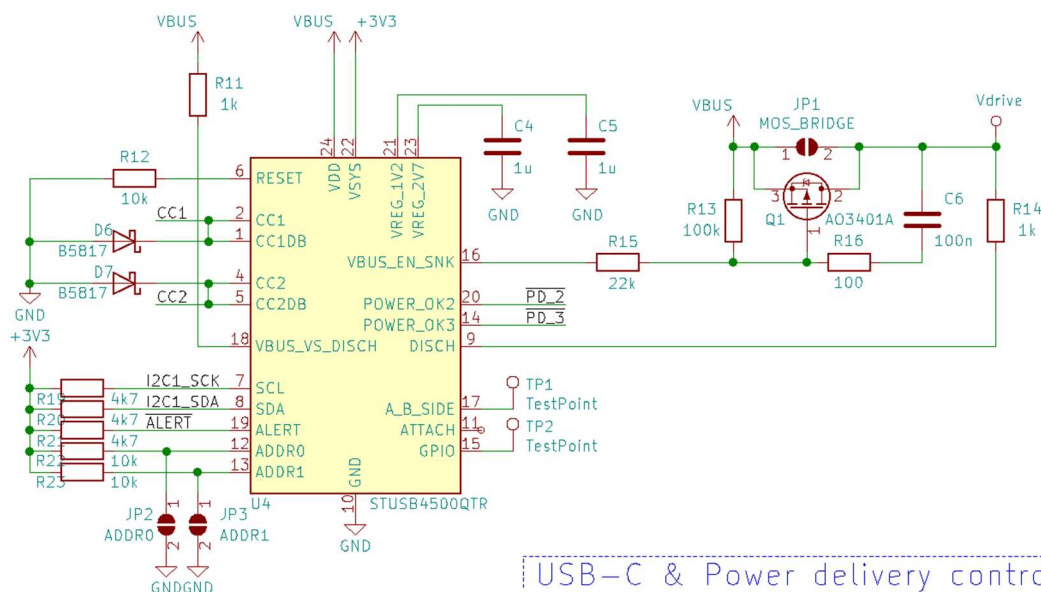


Obr. 3.2.1 – obecné schéma komunikace [Zdroj: vlastní, vytvořeno pomocí draw.io]

Celý projekt (všechny desky) má na MQTT serveru vlastní hlavní topic, tam jsou děleny na subtopic (dle jejich výrobních/registračních čísel napsaných na zadní straně desky), kde jsou rozděleny subtopic pro jednotlivé periferie.

4.2. Power Delivery controller

IO STUSB4500 je napájecím srdcem celé vývojové desky. Ovládá napájení pro vše – komunikuje se zdrojem a nastavuje omezení výstupního proudu či velikost výstupního napětí.



USB-C & Power delivery controller

Obr. 4.2.1 – zapojení Power delivery® ovladače [Zdroj: vlastní]

Ovládací signály mezi MCU a STUSB jsou přenášeny po interní I2C sběrnici. Adresu řadiče lze nastavit zkratováním/rozpojením plošek zkratovacích propojů ADDR0 a ADDR1 na zadní straně desky na adresu 0x28/0x29/0x2A/0x2B. Základně je nastavena na 0x2B (oba piny jsou na log.1).

Interní I2C sběrnice má také zapojeny 4,7kΩ pull-up rezistory. U ESP32 lze nastavit na jakémkoliv IO pinu lze nastavit pull-up, avšak zde je možná programová chyba ošetřena jejich reálným umístěním na desku. V případě objednání (výroby i osazení) např. v Číně vyjde jejich osazení cca na 5,-

Po odpojení napájení nechá tento obvod na pinech CC1 a CC2 (a jejich opačné var. ozn. **DB) odpor 5,1kΩ proti zemi.

Po připojení USB-C kabelu jsou CC piny připojeny na zem přes interní rezistor 5,1kΩ, což zdroji indikuje připojení zátěže a zdroj začne dodávat základní napětí 5V. Z něj je napájen samotný ovladač STUSB (poradí si se vstupním napětím 4-20V). Na piny interních regulátorů 1,2V a 2,7V jsou připojeny filtrační kondenzátory. Po naběhnutí samotného driveru je propuštěno napájecí napětí dále do desky. Na CC piny (tolerující pouze 22V) jsou připojeny ochranné Schottkyho diody pro odfiltrování statické elektřiny.

Řízením obvodu po I2C sběrnici lze nastavit výstupní napětí zdroje (pokud podporuje PD) na 9V, 12V...20V. Toto napětí je použito pro napájení step-down konvertoru napájecího samotnou desku a také je vyvedeno na konektorech pro periferie. Lze si tedy programem nastavit výstupní napětí z desky a vyrobit si jednoduchý „regulovatelný“ zdroj. Lze nastavit i proudové omezení.

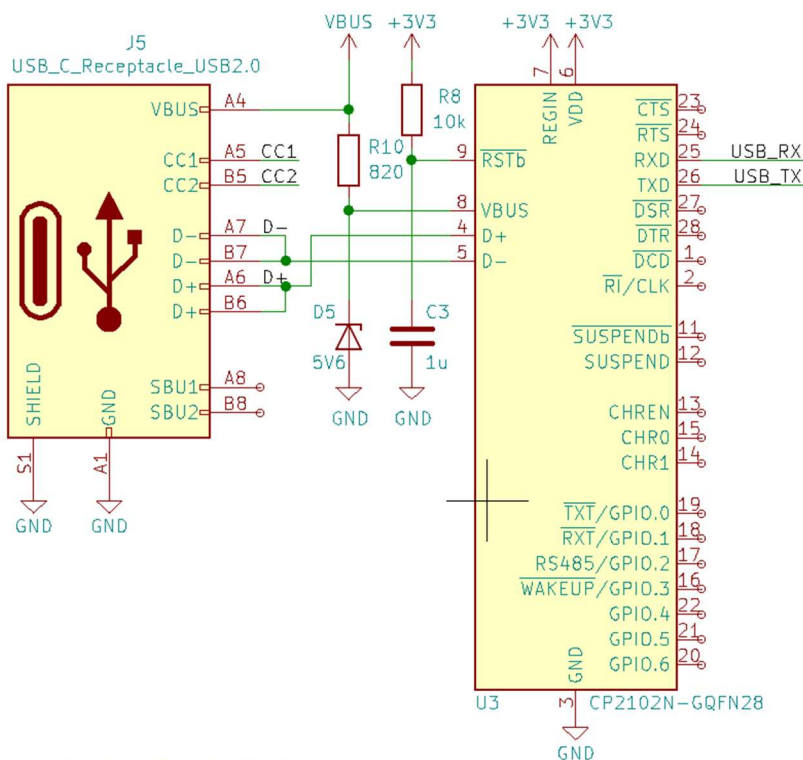
V případě potřeby lze STUSB vyřadit proletováním propojky JP1 na zadní straně desky. Tím dojde k vyřazení ovládacího MOSFET tranzistoru a napájení je přímo vedeno do obvodů desky, ovšem

v případě poškození CC IO pinů samotného řadiče zdroj nedodá napájecí napětí a desku tím pádem není možné přes USB napájet.

4.3. USB-C USB 2.0 interface

Deska je vybavena konektorem USB-C jakožto novým standardem propojení zařízení. USB-C postupně vytlačuje starší USB Micro B často používané pro mobilní telefony. Lze jej připojit jakoukoliv polaritou, přenese až 120W a s maximální přenosovou rychlostí 40Gbit/s mnohonásobně převyšuje kapacitu USB Micro B. Ačkoliv jeho maximální rychlost deska není schopna využít, alespoň přináší jednoduchost potřeby pouze jednoho propojovacího kabelu ke všemu.

USB-UART TTL převodník CP2102 vytvoří virtuální komunikační (COM) port, přes který systém komunikuje s deskou. Použitý převodník CP2102-GMR podporuje přenosovou rychlost až 1Mbit/s, čímž dovoluje rychlé stahování nových programů do desky, jelikož programy pro ESP8266 a ESP32 jsou často mnohem obsáhlejší než např. v Arduino (ATMega328).



USB interface

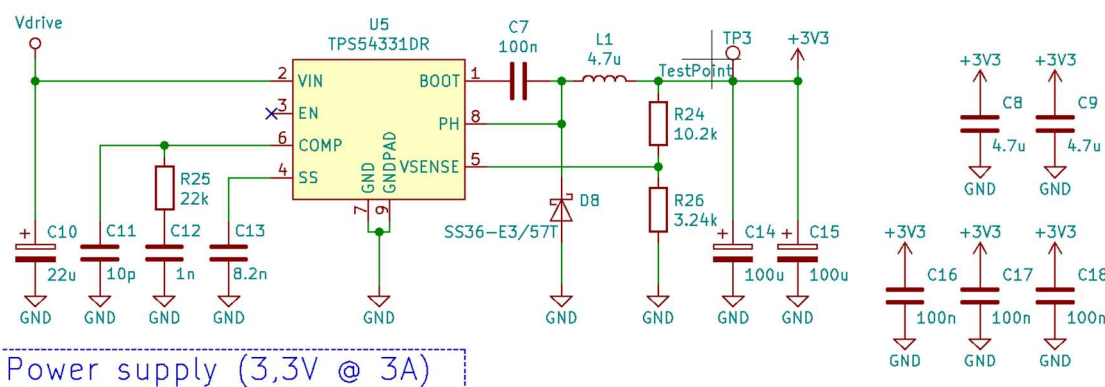
Obr. 4.3.1 – zapojení USB konektoru a USB-UART bridge
[Zdroj: vlastní]

Převodník zjišťuje připojení k PC přítomností log. 1 na pinu VBUS. Tento pin má napěťovou toleranci pouze 6V (dlouhodobě), avšak přes VBUS piny je přiváděno napětí v rozmezí 5-20V.

Napěťová ochrana převodníku je řešena jednoduchým parametrickým stabilizátorem s odporovým děličem a Zenerovou diodou. Velikost rezistoru R10 na omezení proudu diodou D5 je určena z dokumentace výrobce diody. Pro její správnou stabilizaci by jí měl protékat proud minimálně 4mA, aby bylo zajištěno správné otevření. Pokud nastavíme zdroj na nejnižší hodnotu Power Delivery® (9V), napětí musí být rozděleno na 5V (5,6V) úbytek na diodě a 4V (3,4V) úbytek na rezistoru, přičemž proud děličem by se měl pohybovat kolem 4mA. Z tohoto vztahu (ideálního) lze Ohmovým zákonem vypočítat hodnotu rezistoru 850Ω, což je zaokrouhleno a zařazeno do řady E12, 820Ω, 1/16W.

4.4. Napájecí kaskáda – 3,3V @ 3A

Napájecí kaskáda s IO TPS54331DR (nebo DDAR v případě potřeby výkonnějšího chlazení) je schopna udržet desku v provozu již od 3,5V napájecího napětí až do 28V (dle dat výrobce). Tento čip od Texas Instruments je schopen bez pomocného spínacího tranzistoru dodávat až 3A proudu při maximálním zatížení (dlouhodobě). Jedná se o buck (step-down) konvertor se spínací frekvencí 500 kHz.



Obr. 4.4.1 – zapojení step-down konvertoru [Zdroj: vlastní]

Zapojení se skládá z filtračního kondenzátoru na větvi se vstupním napětím ze zdroje, který vyhlazuje propady napětí při prudkých změnách proudového odběru konvertoru. Výstupní napětí odebírané z pinu PH (výkonový výstup) je filtrováno LC dvojbranem tvořeným cívkou L1 (4,7 uH @ 3A) a paralelním zapojením kondenzátorů C14 - C18. Zpětnovazební napětí je do obvodu přiváděno přes dělič napětí tvořený rezistory R24 a R26. Zde opravdu záleží na přesných komponentách. Za použití rezistorů s tolerancí 1% lze klidně dosáhnout podobné kvality filtrace, jako u lineárního regulátoru. Simulace při maximální zátěži ukazuje zvlnění kolem 15 mV ($V_{in} = 5\text{ V}$, $f = 500\text{ kHz}$). Dioda D8 uzemňuje napěťové špičky („přestřely“) z výkonového výstupu. Kvůli vysoké spínací frekvenci a velkému proudovému odběru je použita Schottkyho dioda SS36-E3/57T se závěrným proudem až 3 A.

V případě potřeby výkonnějšího chlazení lze použít tento obvod v DDAR pouzdře. To obsahuje chladicí plochu ve spodní části, ovšem za normálního běhu desky neklesá účinnost obvodu pod 90% a i s animací na LED displeji se po půl hodině neohřál na více jak 30 °C (stálý odběr > 1,2A). Při maximálním zatížení (3 A) se pohybuje účinnost kolem 85% a ztrátový výkon na čipu nepřesahuje 2 W.

Zapojení bylo navrženo za pomoci programu TI Workbench®, který firma dovoluje zdarma použít pro návrhy zapojení se svými IO. Program zároveň provádí simulace koncového zařízení a dovoluje i vygenerování zprávy o navrženém zařízení (viz příloha⁵).

4.5. Rozhraní a periferie na desce

Deska je vybavena 5-ti konektory:

1x USB-C USB 2.0

3x IO konektor (2x 3,3V GPIO + nastavitelné výstupní napětí z PD®)

1x Výstupní datová sběrnice pro adresovatelný LED display

Periferní konektory jsou každý vybaven dvěma GPIO piny z řídicího modulu a přímým výstupem napájecího napětí. Lze tedy řízením Power Delivery® nastavovat výstupní napětí. Proudové omezení je také ponecháno na napájecím zdroji. Proudová pojistka zdroje je většinou základně nastavena na:

- 3,0 A při 5 V (15 W)
- 2,0 A při 9 V (18 W)
- 1,5 A při 12 V (18 W)

(pozn. – platí pro 18 W zdroje)

Deska je ovšem schopna zvládnout

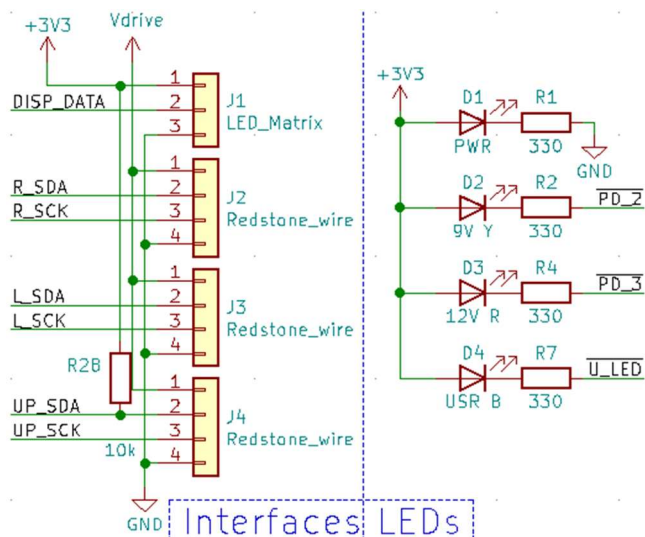
i napětí 20 V (aktuální maximální napětí podporované PD), a udrží proudy až 4,0 A (celkově dlouhodobě).

Indikace aktuálního stavu je realizována čtveřicí LED diod. Indikovány jsou stavy:

- (PWR LED) Napájení připojeno a interní regulátor dodává 3,3 V pro elektroniku
- (PD_2) Napájecí zdroj podporuje PD (QC 2.0)
- (PD_3) Napájecí zdroj podporuje PD 2.0 (QC 3.0)
- (U_LED) Hlášení systému dle programu

Systémové hlášení LED diodou je rozděleno do několika kódů:

- Dlouhé svícení (< 5 s) – připojeno k WiFi
- Dlouhé svícení (> 5 s) – problém s připojením
- Blikání každých 5 s – pokoušení o připojení k MQTT serveru (chyba spojení)
- Bliknutí po připojení k WiFi – programovací mód



Obr. 4.5.1 – zapojení IO portů a LED diod na desce

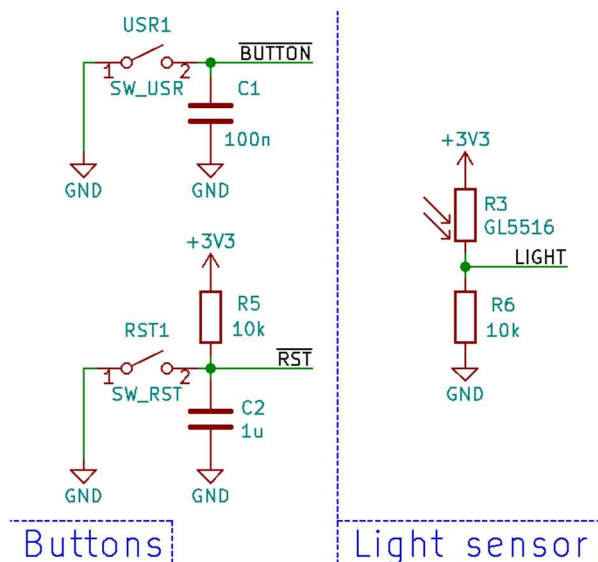
[Zdroj: vlastní]

Desku lze ovládat dvěma vestavěnými tlačítky – RESET (RST1) a BOOT (USR1). Resetovací tlačítko resetuje modul s ESP32 (pouze jej!)

Druhé tlačítko je programovatelné a jeho funkce se mění dle programu. U uživatelského tlačítka je také jako PULL-UP použit interní rezistor v ESP32 (softwarově odpojitelný). Lze tedy tlačítko softwarově vyřadit v případě potřeby.

Funkce:

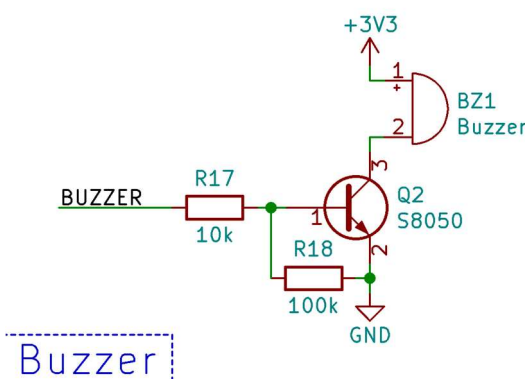
- Držení při resetu desky vyvolá programovací mód přes USB.
- Držení po resetu desky (při připojování k WiFi) vyvolá programovací mód přes WiFi a nastavení po sériové lince.



Obr. 4.5.2 – zapojení tlačítek a senzoru světla
[Zdroj: vlastní]

Deska je také vybavena senzorem světla – fotorezistorem GL5516 zapojeném jako světelně-závislým děličem napětí. Měření intenzity světla není určeno pro přesná měření, zapojení funguje pouze jako možnost analogového vstupu. Nejen že křivka závislosti odporu není lineární, ale ani vestavěný A/D převodník není nikterak kvalitní (sice s 12-bit rozlišením, avšak nikterak s dobrou linearitou*). K osvětlení základních principů však stačí.

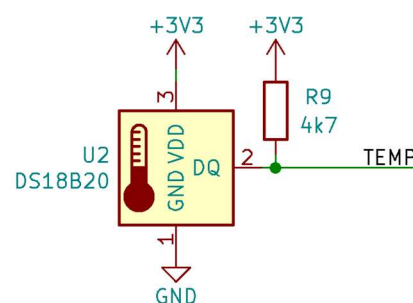
Jako výstupní vestavěná periferie slouží SMD bzučák. Lze jej použít jako sirénu, či i tónový



Obr. 4.5.3 – zapojení bzučáku [Zdroj: vlastní]

generátor. Dle informací z MQTT serveru jsou přiřazovány frekvence jednotlivých tónů. Možno je přehrát 12 tónů v 8 oktávách. Výstupní signál je ovšem pouze 1-bitový. Interní 8-bitový DAC je pouze pro výstupní periferie.

Pro měření teploty je deska vybavena interním senzorem v řídicím modulu a externím čidlem DALLAS DS18B20+ s rozlišením 0,5 °C připojeným přes sériovou sběrnici OneWire



Temp. sensor

Obr. 4.5.4 – zapojení čidla teploty [Zdroj: vlastní]

* - Zdroj: <https://randomnerdtutorials.com/esp32-adc-analog-read-arduino-ide/>

5. Funkce (SW část)

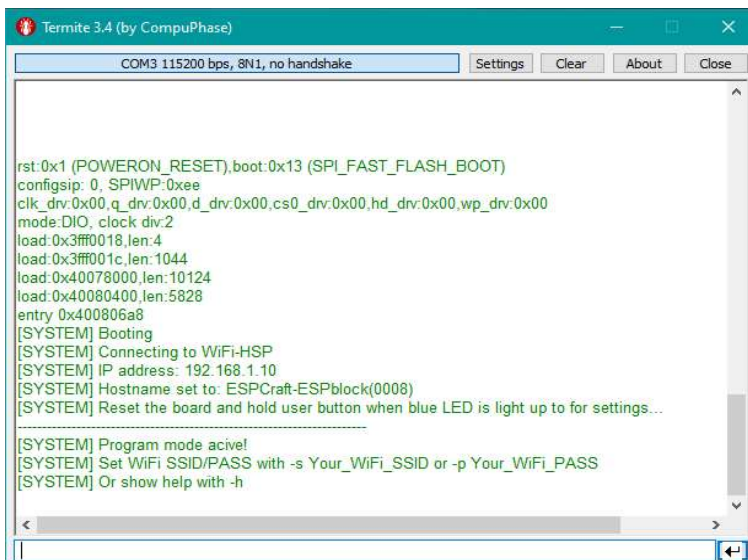
Firmware desky je psán v C++ (konkrétněji ve Wiringu). Tento jazyk byl vybrán z důvodu jednoduchosti a velké popularity mezi bastlíři a amatérskými programátory. Zroveň (jelikož je deska složena z široce používaných komponent) dovoluje použití již používaných knihoven, urychlit celkový vývoj a zjednodušit programování.

RTOS (Real Time Operating System) nahraný v samotném čipu ESP32 po zapnutí spustí boot z SPI Flash v modulu, ve které je nahrán samotný firmware, stejně tak jako paměť alokovaná (volná) pro OTA (Over The Air) programování. Také je přítomný malý (720 kB) SPIFFS (Seriál Periphreal Device Fast File System) např. pro webové rozhraní, ačkoliv to nebylo implementováno. Ovšem desku je možno ovládat přes sériovou linku z počítače.

Řízení přes webové rozhraní by bylo pěkné, ovšem v tomto případě je potřeba, aby desku dokázalo ovládat dítě, zatím co je připojeno na nějakou internetovou komunikační platformu. Prvotní nastavení WiFi by ovšem vyžadovalo připojení nějakého zařízení na WiFi signál desky, což není bez odpojení z komunikační platformy možné. Internetové GUI bylo tedy odloženo, aby byla co nejdříve zajištěna základní funkčnost.

Ovládání desky je momentálně realizováno přes sériový terminál.

Ke komunikaci lze použít jakýkoliv emulátor sériového terminálu. Pro správnou komunikaci je třeba nastavit rychlost 115,2 kBaud, 8 data bitů, 1 stop bit, žádná parita, bez přidání CR/LF (návrat vozíku/konec řádky).



Deska podporuje následující příkazy:

- | | |
|-----------------------------------|--|
| -h | Zobrazí nápovědu |
| -d | Obnoví tovární nastavení |
| -s [jméno_WiFi] | Nastaví jméno WiFi pro připojení na zadaný text [jméno_WiFi] |
| -p [heslo_WiFi] | Nastaví heslo WiFi pro připojení na zadaný text [heslo_WiFi] |
| -a [push/pot/leds] [L/R/U] | Připojí periferní desku (tlačítko – zkr. „push“, potenciometr – zkr. „pot“, LED diody – zkr. „leds“) na zvolený port (levý – L, pravý – R, horní – U). |

Deska také (po připojení k serveru) vypisuje přijaté a odeslané informace.

Obr. 5.1 – Terminálové spojení s deskou [Zdroj: vlastní]

6. Plugin (rozšíření) pro hru a komunikace

Rozšíření pro Minecraft® se dělí do tří kategorií – „plugin“ (rozšíření funkcí hry ze strany serveru), „mód“ (rozšíření ze strany klienta i serveru) či „datapack“ (rozšíření ze strany klienta či serveru).

- Pluginy jsou jednoduchá rozšíření funkcí – primárně příkazy, správa světů, ochrana herních oblastí, řízení přístupu k serveru apod.
- Datapack je rozšíření neupravující základní funkce hry – třeba přidání nových zvířat, některých funkcionalit apod.
- Mód je rozšíření zasahující do samotného strojového kódu hry dovolující cokoliv, co si programátor zamane – přidání zvířat, bloků, zbraní, mobů (zvířat/obyvatel), či celých nových světů. Jsou ovšem nejsložitější na instalaci.

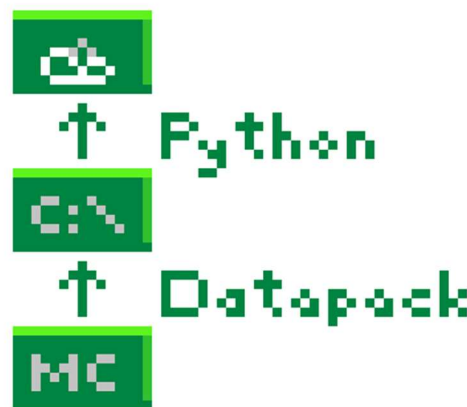
Rozšíření pro komunikaci s ESP Craftem se skládá ze dvou částí – datapacku pro hru a MQTT mostu v podobě skriptu psaného v jazyce Python (verze 3+).

Rozšíření v podobě datapacku virtuálně přidá nový blok do hry – ESP blok, který reaguje na vnější podmínky (změny intenzity vstupního signálu) a vypisuje je do „chatu“. Hra je zaznamenává do tzv. logu, ve kterém je vyhledává skript psaný v Pythonu. Zprávy z ESP Bloku pak dekóduje a posílá na MQTT Server.

Zpětná komunikace je také obstarávána tímto skriptem. Zde se skript zaregistruje jako „subscriber“ k určitým topicům. V případě zjištění změny požádá hru o nalezení souřadnic přiřazeného ESP Bloku. Pokud hra vrátí nalezenou pozici, je tento blok nastaven na příchozí hodnotu. Pokud nebude nalezen, hra nijak nereaguje. Tato část ovšem zatím není implementována.

Jelikož se o samotnou komunikaci stará skript mimo hru, je třeba mít přístup přímo k terminálu serveru, na kterém hra běží. Nestačí pouze přístup ke konzoli samotné hry. Zároveň také hostování na serveru s OS Linux (např. Ubuntu), ve kterém je Python většinou integrován již při instalaci a je potřeba akorát doinstalovat knihovny pro zajištění komunikace (obsaženo v instalačním skriptu). Dále je potřeba zapnout RCON (Remote Console), aby celý systém fungoval. Veškeré programové vybavení ke zprovoznění serveru je na datovém médiu na zadních deskách práce.

Volba typu rozšíření padla na datapack z důvodu široké kompatibility mezi verzemi hry a jednoduchosti instalace. Každý herní „svět“ lze rozšířit o tyto jednoduché modifikace pouze přidáním rozšíření do složky světa. To je výhoda oproti pluginům, které musí běžet na serverové verzi hry, či modům, které potřebují i API implementované přímo ve hře (Forge/Fabric/...).



Obr. 6.1.1 – grafické znázornění cesty dat ze hry [Zdroj: vlastní]



Obr. 6.1.2 – aktuální verze rozšíření ve hře [Zdroj: vlastní]

Ačkoliv datapack lze použít i v lokální hře, nelze ze hry navázat spojení s MQTT serverem, což dělá celé rozšíření (z hlediska IoT) takřka nepoužitelné. To je ovšem dáno potřebou komunikace přes RCON, který v lokálním světě nelze aktivovat. Nefungoval by tedy ani mód, ani plugin (který by vůbec nebylo možné aktivovat).

Další možností je místo uvolnění datapacku uvolnit verzi celého serveru, což ale není možné z právního hlediska (není možné distribuovat instalovaný MC server – [MC EULA](#), sekce „**ONE MAJOR RULE**“*). Server je tak nutné kompletně vytvořit na klientském PC a ručně do něj přidat veškeré dodatky. Využití datapacku zároveň umožňuje kompatibilitu nejen mezi verzemi, ale i mezi edicemi. Lze jej tedy provozovat na serveru s klasickou Java edicí, s Bedrock (Windows 10 + Xbox), či Education edicí, která je určena přímo pro školy.

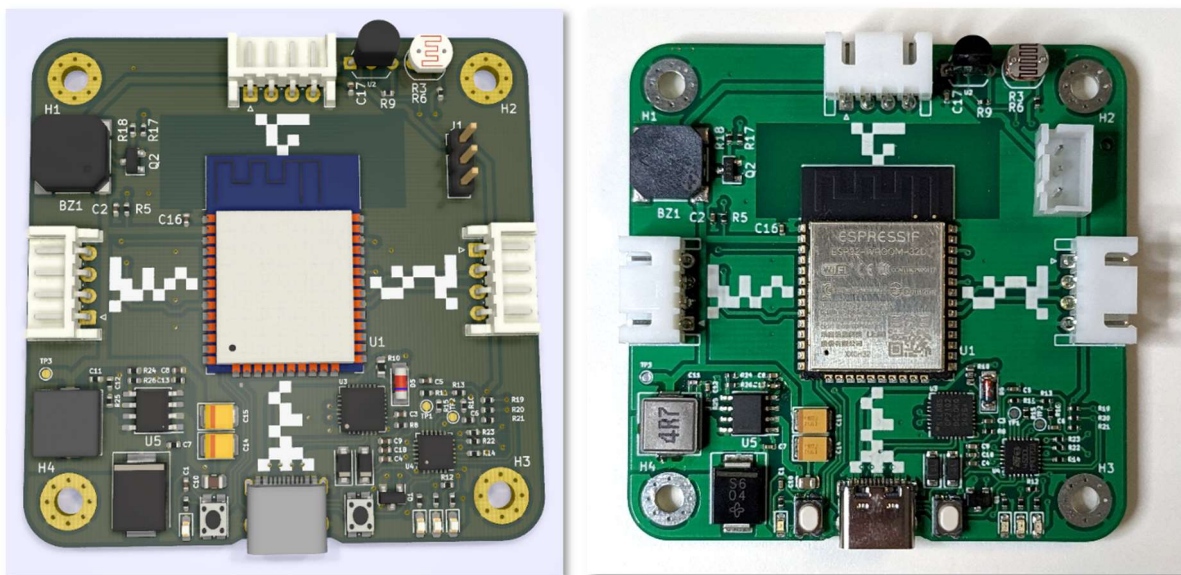
Vývoj tohoto rozšíření aktuálně pouze vedu a pracuji na zpětné komunikaci. Zásluhy za vývoj datapacku pro hru, skriptu i samotného nápadu komunikace přes RCON patří Jiřímu Siterovi.

* - Zdroj: <https://www.minecraft.net/en-us/eula/>

7. Výrobek a závěr

Výrobek (e.g. samotná deska) je vyroben čínskou společností J@LC, velkovýrobcem plošných spojů, který nedávno přidal i možnost osazení SMD komponent. Veškeré komponenty jsou tedy vybrány s ohledem na jejich skladové zásoby, cenu za ks a primárně SMD, což celý výrobní proces urychluje a zjednodušuje. Pro dokončení je již třeba pouze doletovat 4 konektory a dvě čidla (fotorezistor a teplotní čidlo DS18B20). Jelikož se nejedná o klíčové komponenty k zajištění funkce, ale pouze o periferie, lze desku zprovoznit a otestovat ihned po doručení, což urychluje nalezení případných výrobních vad, kterým se ani tento projekt nevyhnul.

Samotná deska si prošla dlouhým vývojem v několika verzích. Zde popisovaná, v.3, je první verzí plně odladěnou a připravenou pro prodej.



Obr. 7.1.1 a 7.7.2 – vizualizace v.3 (vlevo) a reálný výrobek (vpravo) [Zdroj: vlastní]

Na obrázcích výše si lze všimnout mírných rozdílů, jako např. jiné barvy reálného PCB, nepoužití zlaté povrchové úpravy a jiných modelů konektorů a tlačítek, ovšem díky pouze rozdílům v detailech je vizualizace velmi přínosná pro samotný vývoj zařízení a pro design na desce již při vývoji.

PCB je pouze 2-vrstvé, což zesložituje návrh a klade větší nároky na tloušťku cest po povrchu desky, avšak snižuje výrobní náklady (35,60\$ @ 30ks u JLC). Zároveň také nepřítomnost vnitřních napájecích ploch zvyšuje elektromagnetické emise desky harmonických složek datových signálů.

U projektu lze tedy zvolit při výrobě několik výrobních možností:

- 2 -> 4 vrstvy PCB -> +26,313Kč / ks, nižší elm. Emise, jednodušší návrh
- 4M -> 16M paměti v ESP32 modulu -> +26,61Kč / ks, více prostoru pro SW výbavu.

Dlouhý vývoj

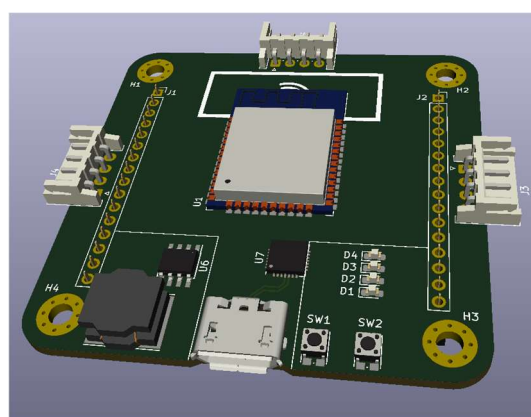
Vývoj desky pro ESP Craft sahá až k projektu z minulého roku, který jsem vyvíjel společně s Danielem Bekem a Milanem Maržencevem pro školu jako výukovou pomůcku. Ta si sama o sobě prošla třemi verzemi, a právě ta se stala základem pro samotný kit pro ESP Craft.

Po dohodnutí na složení a cílech, které by kit měl splňovat a umět byl zahájen vývoj, který se datuje na listopad roku 2020.



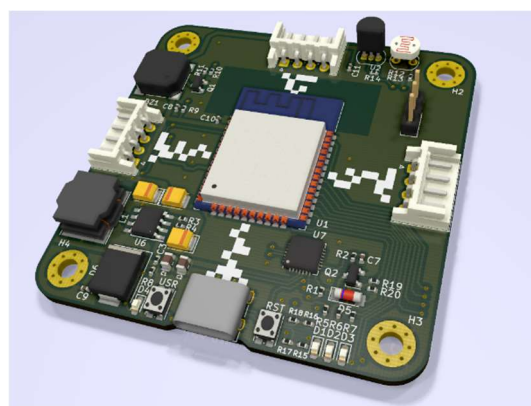
Obr. 7.1.3 – deska explore IoT [Zdroj: e-iot.cz]

Nedlouho na to byla sestavena první verze, která již vzdáleně připomíná aktuální verzi (design bloku, rozdělení na 9 sekcí - 4 komunikační, 4 s výbavou a samotné MCU). Design byl upraven pro osazení po obou stranách na 2 až 4-vrstvé PCB s nasazovací RGB LED maticí. Tento design se neosvědčil a byl nahrazen verzí bez řadiče pro RGB LED display.



Obr. 7.1.4 – ESP Craft v.0.1 [Zdroj: vlastní]

Místo něj bylo usouzeno použít adresovatelnou LED matici s diodami WS2812B. Řešení, které mělo původně celý návrh zjednodušit, ale prodražit, nakonec snížilo výrobní náklady téměř na polovinu, což umožnilo přidání dalších periférií a potisků na desku. Verze 1.0 vznikla jakožto první určená k výrobě a měla stihnout i únorové vyhlášení vítězů soutěže Future Factory, kde měla být hlavní cenou. Várka 30ks desek s USB-C, která stavěla na mých poznatcích s USB Micro B, neměla nejmenší šanci fungovat. Chybný výběr napájecího obvodu způsobil funkčnost až od 5,4 V napájecího napětí. Řízení dodávání energie nebylo vůbec implementováno. Datové piny USB-C nebyly správně zapojeny. Naštěstí seznam součástek se primárně jen rozšířil, takže lze použít součástky z této várky na opravu nových verzí, což se i skutečně stalo.



Obr. 7.1.5 – ESP Craft v.1 [Zdroj: vlastní]

Verze 2 a 3 a designově téměř neliší a změny nejsou na první pohled příliš patrné. Verzi 2 ovšem postihla výrobní vada („výrobní překlep“), a to zaměnění inventárních čísel pro sklad SMD komponent J@LC. Kondenzátor C13 v napájecí kaskádě (8,2 nF, str. 16) byl zaměněn za cívku 8,2

nH. Tato chyba způsobila vychýlení spínací frekvence step-downu z 0,5 MHz na nestabilních 5 Hz. Várku zachránily komponenty z první série, kde je použit kondenzátor 10 nF ve stejné velikosti. Ten sice lehce nesplňuje návrhové tolerance ($6,7 \text{ nF} < C13 < 9,4 \text{ nF}$), avšak i tak s ním deska spolehlivě funguje i v plném zatížení. U v.4 je již tato chyba opravena, stejně tak jako připojení periferních pinů na 2. kanál interního A/D převodníku, který nelze využít při běžícím WiFi připojení. Zároveň také uživatelské tlačítko bylo přepojeno na GPIO pin 0, jelikož jeho uzemněním při bootu ESP32 vstupuje do programovacího módu. Tlačítko tedy sekunduje i jako boot. Taková funkce mu byla přidělena i v mém firmwaru, jakožto aktivace OTA programování a nastavení.

8. Použité zdroje

ESP32 Pinout Reference: Which GPIO pins should you use? | Random Nerd Tutorials. *Random Nerd Tutorials | Learn ESP32, ESP8266, Arduino, and Raspberry Pi* [online]. Copyright © 2013 [cit. 29.03.2021]. Dostupné z: <https://randomnerdtutorials.com/esp32-pinout-reference-gpios/>

ESP32 PWM with Arduino IDE (Analog Output) | Random Nerd Tutorials. *Random Nerd Tutorials | Learn ESP32, ESP8266, Arduino, and Raspberry Pi* [online]. Copyright © 2013 [cit. 29.03.2021]. Dostupné z: <https://randomnerdtutorials.com/esp32-pwm-arduino-ide/>

ESP32 Analog Input with Arduino IDE | Random Nerd Tutorials. *Random Nerd Tutorials | Learn ESP32, ESP8266, Arduino, and Raspberry Pi* [online]. Copyright © 2013 [cit. 29.03.2021]. Dostupné z: <https://randomnerdtutorials.com/esp32-adc-analog-read-arduino-ide/>

ESP32 I2C Communication: Set Pins, Multiple Bus Interfaces and Peripherals | Random Nerd Tutorials. *Random Nerd Tutorials | Learn ESP32, ESP8266, Arduino, and Raspberry Pi* [online]. Copyright © 2013 [cit. 29.03.2021]. Dostupné z: <https://randomnerdtutorials.com/esp32-i2c-communication-arduino-ide/>

Download Our Free Electronics eBooks and Resources | Random Nerd Tutorials. *Random Nerd Tutorials | Learn ESP32, ESP8266, Arduino, and Raspberry Pi* [online]. Copyright © 2013 [cit. 29.03.2021]. Dostupné z: <https://randomnerdtutorials.com/download/>

ESP32 and WS2812b RGB led example | ESP32 Learning. *esp32 learning* [online]. Dostupné z: <http://www.esp32learning.com/code/esp32-and-ws2812b-rgb-led-example.php>

Hacking Qualcomm (Quick Charge) QC 2.0/3.0 With ATtiny85 - Hackster.io. *Hackster.io - The community dedicated to learning hardware.* [online]. Copyright © [cit. 29.03.2021]. Dostupné z: https://www.hackster.io/PSoC_Rocks/hacking-qualcomm-quick-charge-qc-2-0-3-0-with-attiny85-b7627d

SparkFun Power Delivery Board - USB-C (Qwiic) - DEV-15801 - SparkFun Electronics. *SparkFun Electronics* [online]. Dostupné z: <https://www.sparkfun.com/products/15801>

A Beginner's Guide to the ESP8266. *Pieter's Pages* [online]. Dostupné z: <https://tttapa.github.io/ESP8266/Chap01%20-%20ESP8266.html>

Get Started - ESP32 - — ESP-IDF Programming Guide latest documentation. *302 Found* [online]. Copyright © Copyright 2016 [cit. 29.03.2021]. Dostupné z: <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/get-started/>

Objevte Internet věcí s námi – NEUČTE SE | HRAJTE SI. *Objevte Internet věcí s námi – NEUČTE SE | HRAJTE SI* [online]. Dostupné z: <https://www.e-iot.cz/>

USB Charger Fooled Into Variable Voltage Source | Hackaday. *Hackaday | Fresh Hacks Every Day* [online]. Copyright © 2021 [cit. 29.03.2021]. Dostupné z: <https://hackaday.com/2017/06/02/usb-charger-persuaded-to-variable-voltage-source/>

Installation | Spigot high performace Minecraft [online]. Dostupné z: <https://www.spigotmc.org/wiki/spigot-installation/>

Tutorials/Creating a data pack | Official Minecraft Wiki. [online]. Dostupné z: https://minecraft.fandom.com/wiki/Tutorials/Creating_a_data_pack

9. Seznam příloh

Příloha 1: AN1953 | Microchip – Introduction to USB Type-C™ | Autor: Andrew Rogers, © 2015
Microchip Technology Inc. (elektronicky)

Příloha 2: TPS54331 3-A, 28-V Input, Step-down DC-DC Converter with Eco-mode™ | Texas
Instruments (elektronicky)

Příloha 3: STUSB4500L Standalone USB Type-C™ sink port controller | ST Microelectronics
(elektronicky)

Příloha 4: CP2102/9 Single-Chip USB to UART Bridge | Silicon Labs (elektronicky)

Příloha 5: TI Webench® Design report | Texas Instruments (elektronicky)

Příloha 6: ESP32-WROOM-32 Datasheet | Espressif systems (elektronicky)

Příloha 7: Projektová data (elektronicky)

Všechny elektronické přílohy lze najít na přiloženém datovém CD

