



Středoškolská technika 2022

Setkání a prezentace prací středoškolských studentů na ČVUT

PrintControl

Jan Mráz, Ondřej Matějka

Vyšší odborná škola a Střední průmyslová škola elektrotechnická Plzeň

Koterovská 828/85, Plzeň

Poděkování

Tímto bychom chtěli poděkovat především panu Josefu Fořtovi a panu Janu Drexlerovi za pomoc s vývojem projektu a za poskytnutí prostředků pro jeho tvorbu projektu. Poskytli nám zázemí a cenné zkušenosti, které jsme potřebovali pro realizaci projektu.

Anotace

Tato práce se zabývá návrhem a realizací boxu pro 3D tiskárny s možností rozšíření až o devět samostatných modulů pro zlepšení prostředí při tisku a celkové práce s 3D tiskárnami. PrintControl najde využití u každého kdo požaduje snížení nežádoucích problémů s 3D tiskem a usnadnění práce s údržbou 3D tiskáren. Celý box komunikuje s uživatelem za pomoci aplikace, odkud jsou ovládány některé funkce boxu. Díky přívětivému uživatelskému rozhraní i poloautonomnímu fungování celého systému je PrintControl vhodný, pro jednotlivce, kteří mají tiskárnu jen pro radost z tisku, ale také pro společnosti zabývající se zakázkovým 3D tiskem s větším počtem tiskáren.

Klíčová slova

3D tiskárna; Internet věcí; box

Annotation

This work deals with the design and implementation of a box for 3D printers with the possibility of expanding up to 9 separate modules to improve the printing environment and the overall work with 3D printers. PrintControl can be used by anyone who offers to reduce unwanted problems with 3D printing and simplify work and maintenance around 3D printers. The whole box communicates with the user with the help of an application from where some functions of the box are controlled. Thanks to the friendly user interface and semi-autonomous operation of the entire system, PrintControl is both for individuals who have a suitable accessory just for the pleasure of printing, and for a company dealing with custom 3D printing with many printers.

Key words

3D printer; Internet of Things; box

Obsah

1	Úvod.....	6
2	Technická část projektu.....	8
2.1	Program Autodesk Fusion 360	8
2.2	Blokové schéma PrintControlu.....	8
2.3	Konstrukce boxu.....	9
2.3.1	Nosná konstrukce	9
2.3.2	Základna boxu	10
2.3.3	Umístění modulu ventilátoru, senzoru a modulu pro LED podsvícení.....	11
2.3.4	Stěny boxu.....	11
2.3.5	Celkové rozměry boxu	12
2.4	Modul pro měření teploty a vlhkosti	13
2.4.1	Průmyslová sběrnice CAN bus	13
2.4.2	Připojení napájení a sériové sběrnice CAN.....	14
2.4.3	Stabilizátor	15
2.4.4	Mikrokontroler	16
2.4.5	CAN bus rozhraní.....	17
2.4.6	Sériová konzole	18
2.4.7	Uživatelské rozhraní.....	18
2.4.8	Senzor teploty a vlhkosti	19
2.4.9	Vizualizace modulu teploty a vlhkosti	20
2.5	Modul pro regulaci prostředí – ventilátor.....	22
2.5.1	Schéma zapojení pro ovládání ventilátoru	22
2.5.2	Vizualizace modulu pro regulaci prostředí – ventilátor	23
2.6	Modul pro regulaci prostředí – topné těleso	25
2.7	Modul pro komunikaci – gateway	26
2.7.1	Vývojová deska SAM-IoT-WG od společnosti Microchip.....	26
2.7.2	Vizualizace modulu gateway	27
3	Programovací část projektu.....	29
3.1	Programovací prostředí MPLAB X IDE	29
3.2	MPLAB Harmony 3	29
3.2.1	Jak začít s Harmony 3	30
3.3	Založení projektu v MPLAB X IDE.....	31

3.3.1	Harmony 3 – Project Graph	34
3.3.2	Harmony 3 – periferie	35
3.3.3	Nastavení pinů mikrokontroleru.....	36
3.4	MPLAB Snap In-Circuit Debugger	37
3.5	Kód pro modul teploty a vlhkosti	39
3.5.1	Definice globálních proměnných	39
3.5.2	Funkce pro čtení a zápis na CAN bus sběrnici.....	40
3.5.3	Funkce pro SERCOM 2, TC2 a tlačítko SW0	40
3.5.4	Funkce pro získání dat ze senzoru a výpočet teploty a vlhkosti	41
3.5.5	Funkce main	42
3.5.6	Funkce while	43
4	Aplikační část projektu.....	44
4.1	Popis jednotlivých nodů v NodeREDu.....	45
4.1.1	MQTT IN node.....	46
4.1.2	JSON node.....	46
4.1.3	Change node	47
4.1.4	Dashboard UI	48
5	Webové stránky.....	50
6	Budoucnost projektu	52
7	Závěr.....	53
8	Literatura	54
9	Seznam obrázků	55

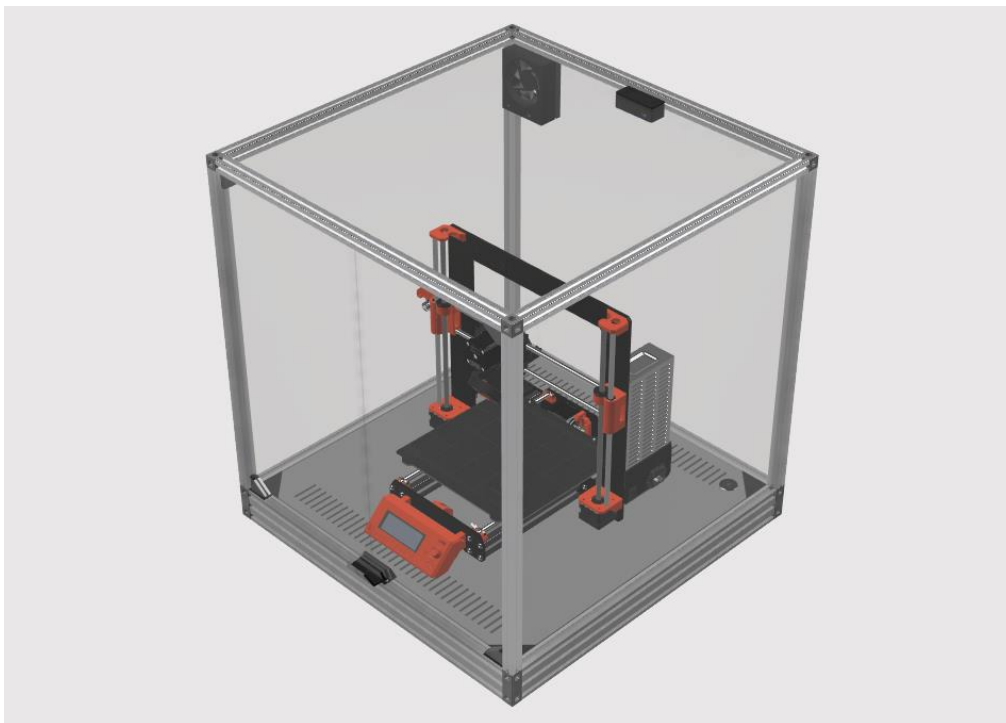
1 Úvod

Nápad pro vznik tohoto projektu jsme dostali ve spolupráci se zástupci firmy TData s.r.o., která se vývojem IoT zařízení zabývá již několik let. Formou mentoringu nám ukázali, že své teoretické znalosti můžeme využít v praxi, a ještě se při tom učit novým věcem. S jejich pomocí jsme zjistili, že se nemusíme bát větších projektů a realizace našich nápadů.

Projekt PrintControl se zaměřuje na řešení problematiky ohledně 3D tiskáren, zlepšení jejich ovládání a podmínek při tisku. Naše řešení je určeno komukoliv, kdo požaduje lepší podmínky pro tisk a snížení nežádoucích problémů s tiskárnami. Systém PrintControl je velmi užitečný pro malé a středně velké firmy pohybující se v byznysu se 3D tiskárnami a poskytující služby jako je zakázkový tisk. Těmto firmám často schází nějaký chytrý řídicí systém pro efektivní chod a řízení tiskáren, popřípadě celé tiskařské farmy, které mohou obsahovat až několik desítek 3D tiskáren. Dále je taky zařízení velmi užitečné pro střední a vysoké školy a má využití i v případech menšího počtu tiskáren jako například v kancelářích nebo domácnostech.

Kryt v podobě boxu pro každou 3D tiskárnu brání usazování prachu, snižuje hluk, potlačuje šíření výparů při tisku a pomocí jednotlivých IoT zařízení vytváří stabilní prostředí pro tisk ze všech materiálů. Díky možnosti rozšíření až o devět samostatných modulů vzniká jednoduchý modulární systém. Uživatel si tak může vybrat jaké moduly, a tedy i funkce vyhovují jeho potřebám a které ne. Ke třem hlavním modulům patří zařízení pro regulaci vzduchu tvořící dva ventilátory, modul s topným tělesem pro zvyšování teploty a v neposlední řadě modul s měřením teploty a vlhkosti v celém boxu. Moduly s ventilátory a modul s topným tělesem pracují na základě shromažďování dat ze senzoru teploty a vlhkosti a podle aktuálních podmínek v boxu se nastaví potřebná rychlost ventilátorů a výkon topného tělesa. Mezi další samostatná zařízení patří modul se zdrojem pro napájení celého boxu včetně tiskárny a zařízení pro kontrolu její spotřeby, nebo možného odpojení od napájení v případě hrozícího nebezpečí. Dále systém obsahuje modul pro ovládání LED diod, a nakonec zařízení jménem gateway, díky kterému se veškerá nashromážděná data ze všech zařízení odesílají pomocí technologie Wi-Fi do webové či mobilní aplikace, kde jsou následně vyhodnocena a zpracována k dalšímu použití. Uživatel tak formou vizualizace vidí detailní informace o 3D tiskárně pohodlně ve vlastní aplikaci a dokáže s každou tiskárnou komunikovat například v případě manuálního ovládání LED osvětlení.

Mezi hlavní konstrukční výhody boxu a celého zařízení patří snadné upevnění několik boxů na sebe. Jsou navrženy pro bezpečnou montáž a celý systém je konstruován tak, aby byl co nejvíce efektivní a užitečný co se týče místa a práce s 3D tiskárnami. Každý box zároveň nabízí možnost použití jako úložný prostor s ideálním prostředím pro skladování filamentů. V takovém případě jsou odebrány některé funkce pro tiskárny. Vzniká tak velmi praktická možnost pro tiskařské farmy v podobě sloupce ze tří boxů, kde se ve dvou boxech nacházejí 3D tiskárny a ve třetím nejvyšším boxu jsou uloženy filamenty. Teplo ze spodních tiskáren tak lze snadno využít pro vytápění nejvyššího boxu a dokážeme tak zajistit alespoň částečnou úsporu energie.



Obrázek 1 PrintControl – box pro 3D tiskárnu

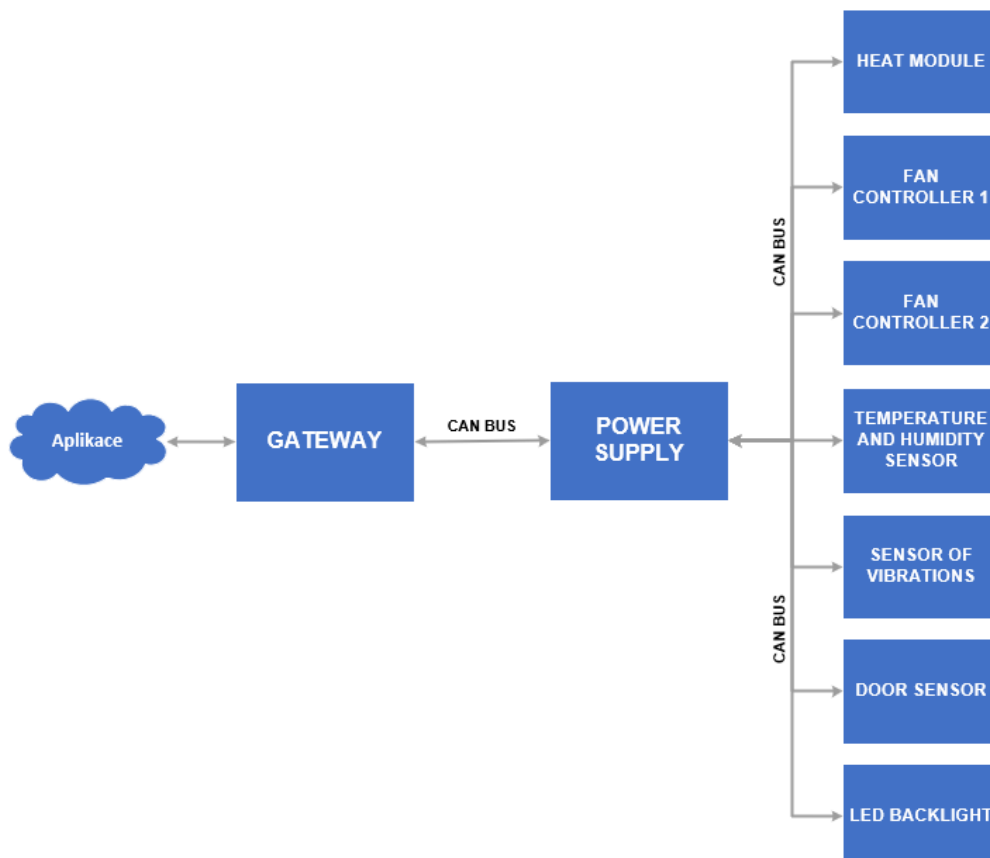
2 Technická část projektu

2.1 Program Autodesk Fusion 360

Fusion 360 je cloudový software pro návrh a tvorbu 3D modelů a konstrukcí. Program uživatelé také nabízí možnost tvorby plošných spojů. Oproti programům jako například Eagle (vytváření plošných spojů) nebo Inventor (3D modelování) nabízí Fusion 360 výhodu v jednoduchém propojení návrhu plošných spojů společně s 3D modely. Díky těmto vlastnostem a dalšímu vývoji programu se stává populárnější mezi vývojáři.

2.2 Blokové schéma PrintControlu

Jednotlivé moduly mezi sebou komunikují na průmyslové sériové sběrnici CAN. Všechna data se přenášejí pomocí gateway modulu do cloudové aplikace, kde jsou pak dále zpracována a vizualizována.

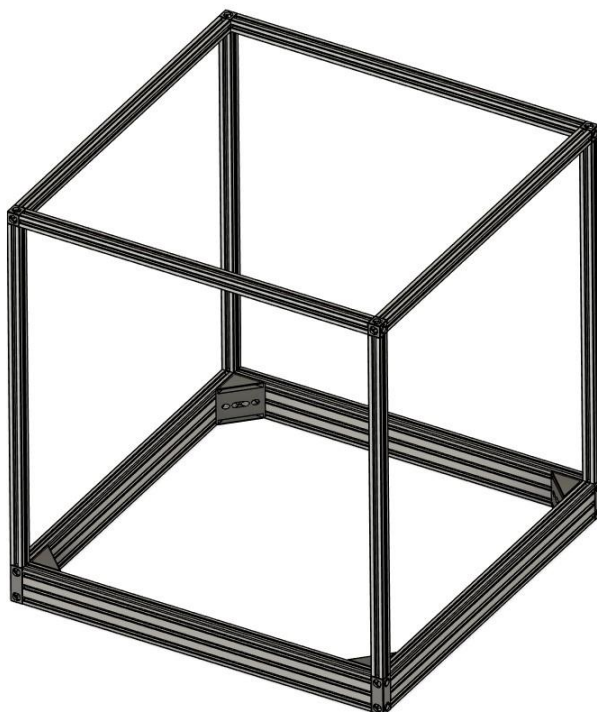


Obrázek 2 Blokové schéma systému PrintControl

2.3 Konstrukce boxu

2.3.1 Nosná konstrukce

Box je tvořen osmi hliníkovými profily velikosti 20x20x560 mm a čtyřmi hliníkovými profily o rozměrech 20x60x560 mm. Hliníkové profily jsme zvolili za účelem snížení hmotnosti a zároveň zvýšení pevnosti celé konstrukce. Jednotlivé profily jsme spojili pomocí speciálně navržených spojů v každém rohu šrouby velikosti M5x40 mm. Pro zajištění pevnosti jsme do každého rohu ještě umístili rohovou spojnicí připevněnou celkem čtyřmi šrouby M5. Všechny tištěné díly jsme vytiskli na 3D tiskárně, protože se jedná o nejlevnější variantu. Jako materiál pro tištěné díly jsme zvolili ABS z důvodu velké váhové zatížitelnosti a nutnosti odolávat poměrně vysokým teplotám v boxu. Tato sestava tak tvoří hlavní nosnou konstrukci celého boxu, ke které jsou připevněné další komponenty.



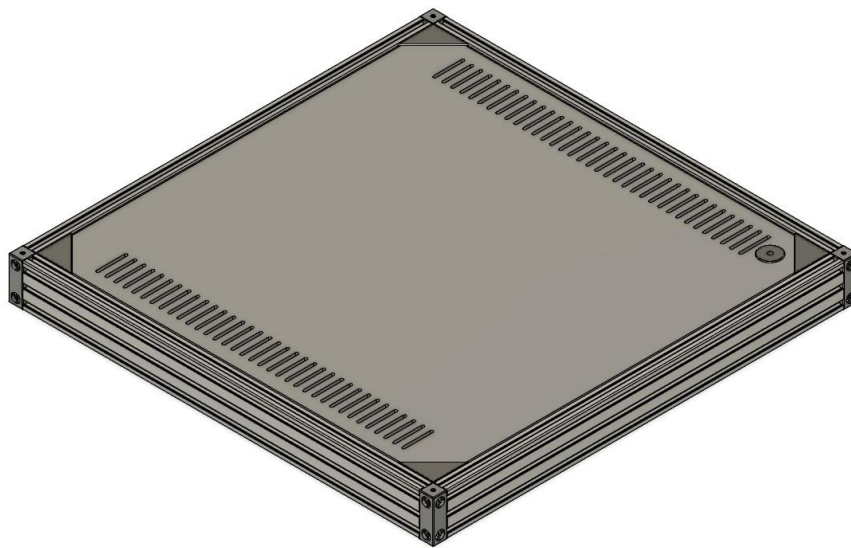
Obrázek 3 Nosná konstrukce boxu, 3D vizualizace

Spodní část celého boxu je nejdůležitější částí vzhledem k funkci celého systému. Jedná se o prostor, kde jsou umístěny dva hlavní moduly pro regulaci prostředí a zdroj napájení celého zařízení včetně napájení 3D tiskárny uvnitř boxu. Zvolili jsme tedy větší velikost spodních hliníkových profilů, abychom získali dostatečně velký prostor pro umístění modulů a jejich následnou montáž.

2.3.2 Základna boxu

Jak již bylo zmíněno, ve spodní části boxu jsou umístěny hlavní moduly. Tato část je z obou stran zakrytována překližkou o tloušťce 6 mm. Na horní překližce se nachází samotná 3D tiskárna a na spodní jsou umístěny moduly. V horní překližce jsme vyfrézovali drážky pro zajištění správné cirkulace vzduchu v boxu a také otvor pro napájecí kabel ke 3D tiskárně. Překližku jsme zvolili kvůli její pevnosti a snadnému opracování frézou.

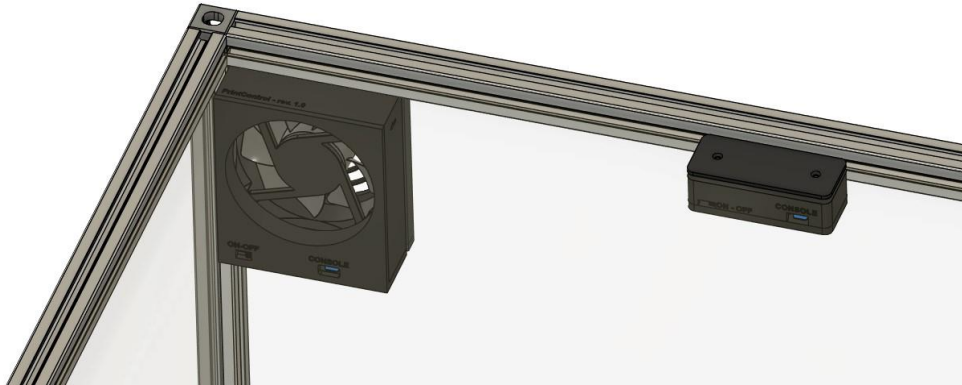
Uvnitř spodní části boxu se nacházejí dvě kazety s moduly. V jedné této kazetě se nachází modul s ventilátorem a modul s topným tělesem. Zadními drážkami v nosné překližce ventilátor nasává vzduch z boxu, který proudí šachtou přímo přes žebrování topného tělesa a tím dochází k jeho ohřívání. Drážkami vpředu se teplý vzduch opět dostává do celkové cirkulace vzduchu v boxu. V druhé kazetě je napájecí zdroj s řídicí jednotkou pro měření spotřeby 3D tiskárny a pro její případné odpojení od napájení. Oba tyto bloky jsme navrhli tak, aby bylo možné v zadní části boxu vysunout ven v případě poruchy či servisní kontroly. Obě kazety jsou stejně jako spojovací díly tištěny na 3D tiskárně z materiálu ABS.



Obrázek 4 Základna boxu, 3D vizualizace

2.3.3 Umístění modulu ventilátoru, senzoru a modulu pro LED podsvícení

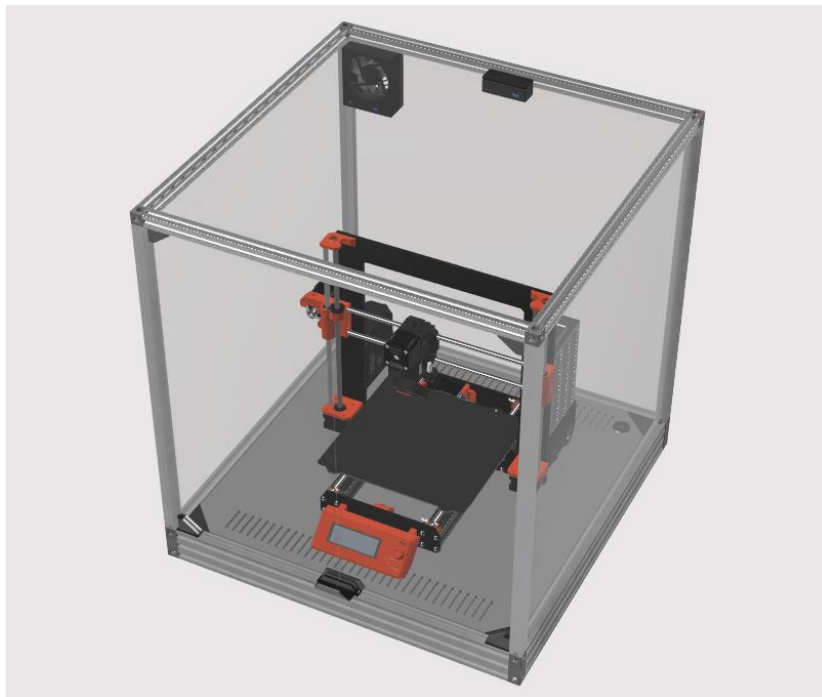
Modul s ventilátorem pro nasávání venkovního vzduchu, modul pro měření teploty a vlhkosti a modul pro ovládání LED diod jsme umístili do vrchní části boxu. Moduly jsou buď přišroubovány přímo do plexiskla nebo jako v případě ventilátoru zasunuty do plexiskla a upevněny pomocí drážek v krabičce.



Obrázek 5 Umístění modulů v boxu, 3D vizualizace

2.3.4 Stěny boxu

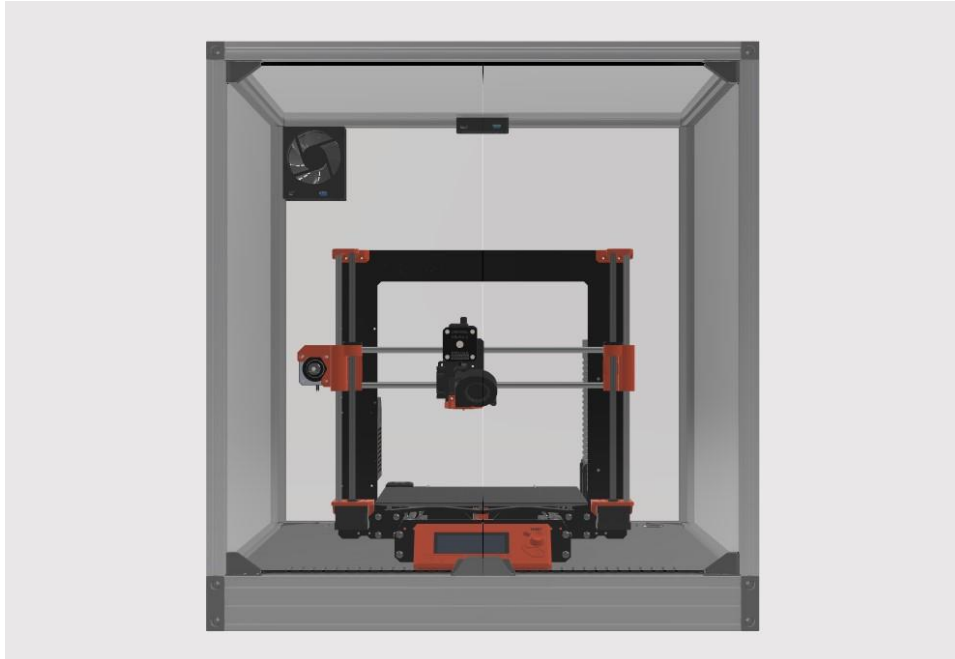
Jako materiál pro stěny boxu lze použít plexisklo nebo překližku. Záleží na každém uživateli, co preferuje, nebo co potřebuje. My jsme pro stěny boxu jsme zvolili plexisklo, z důvodu průhlednosti a možnosti kontroly 3D tiskárny při tisku (bez nutnosti otevírání celého boxu). Zároveň lze plexisklo snadno opracovat pomocí frézy přesně podle našich potřeb.



Obrázek 6 Celá sestava PrintControl, 3D vizualizace

2.3.5 Celkové rozměry boxu

Maximální rozměry pro tiskárnu jsou 560x560x560 mm. Do boxu se tedy vejde většina 3D tiskáren, které jsou dnes dostupné na trhu. Při této velikosti zbývá uvnitř boxu ještě dostatek prostoru například pro umístění filamentu. Vnější rozměry celého boxu jsou 600x600x640 mm.



Obrázek 7 PrintControl pohled zepředu, 3D vizualizace

2.4 Modul pro měření teploty a vlhkosti

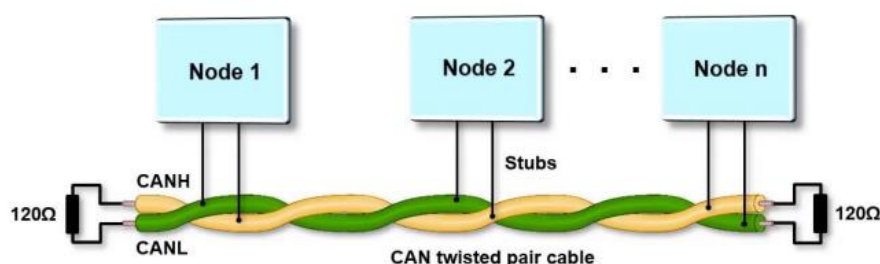
Všechny desky plošných spojů, které jsou v projektu PrintControl obsaženy jsou navrženy tak, aby hlavní obvody a komponenty byly zachovány ve stejné podobě na všech modulech. Výsledkem je snížení náročnosti programování všech modulů, návrhu dalších modulů i finálního zapojení v boxu a nižší cena součástek.

2.4.1 Průmyslová sběrnice CAN bus

Sběrnici CAN bus jsme si vybrali na základě naší potřeby používat zařízení v různých prostředích a také zajištění kvalitního přenosu dat na dlouhé vzdálenosti, a to až v řádech několika metrů.

Průmyslová sériová sběrnice CAN (Controller Area Network) byla vytvořena společností BOSCH roku 1986, kdy vznikla jeho první definice jako ISO 11898-1. Jedná se o sběrnici, která ke komunikaci nepoužívá hodinový signál, kvůli čemuž ji řadíme mezi sběrnice asynchronní. Jako přenosové médium pro CAN sběrnici se používá kroucená dvojlinka (jeden pár). Mezi nejčastěji používané standardy se řadí CAN 2.0 a CAN FD, které se navzájem liší maximální dosažitelnou rychlostí a velikostí přenášených dat. CAN 2.0 používá datový rámec o velikosti 8 bytů a přenosovou rychlost do 1 Mbps, zatímco CAN FD používá datový rámec o velikosti 64 bytů a přenosovou rychlost má v rozsahu od 2 do 5 Mbps. Sběrnice CAN je tvořena jednotlivými uzly (tzv. nody), které reprezentují samostatná komunikační zařízení připojená na sběrnici. Pro komunikaci se používá diferenční signalizace, kdy jeden vodič se označuje jako CAN H a druhý jako CAN L. Každá sběrnice CAN musí obsahovat zakončovací prvek (tzv. terminátor) pro zabránění odrazům signálu a snížení nežádoucího vyzařování. Jako terminátor se používají dva rezistory. Impedance vedení je 120 Ω .

V současnosti se průmyslová sběrnice CAN používá ve všech typech dopravních prostředků, automatizaci a dalších místech kde je potřeba odolné sériové sběrnice se spolehlivým a rychlým přenosem dat na dlouhé vzdálenosti.

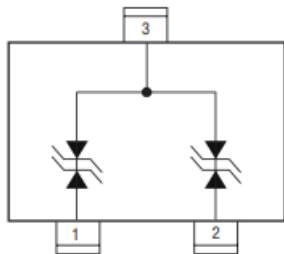


Obrázek 8 Příklad zapojení sběrnice CAN bus

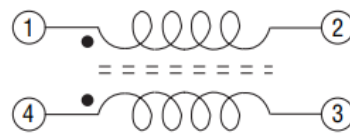
2.4.2 Připojení napájení a sériové sběrnice CAN

Ke každému modulu je přivedeno napájení a průmyslová sběrnice CAN společně jedním FCI konektorem se čtyřmi piny. Tento typ konektoru jsme zvolili, protože je navržen tak, aby se konektor při zapojování modulu nedal otočit. Každý modul jsme na sériovou sběrnici připojili paralelně jako samostatné zařízení neboli tzn. node. V celém boxu jsme se rozhodli vést 12 V stejnosměrného napětí z důvodu napájení ventilátorů, topného tělesa a LED diod, pro které je tato hodnota napětí stanovena datasheetem. Na vstupu napájení se zatím nachází pouze nulový odpor, protože pojistka pro náš požadovaný výkon se nevyrobí ve velikosti 0805, kterou jsme požadovali. V příštích verzích bude pravděpodobně nahrazen pojistkou větší velikosti.

Na sériové sběrnici jsme umístili filtr pro odstranění přechodových jevů a dalších nežádoucích vlivů. Filtr se skládá z dvoucestné obousměrné TVS diody, která slouží k potlačení napětí z přechodových jevů a ochraně před elektrostatickými výboji. Druhou součástí obsaženou ve filtru je tlumivka souhlasného napětí, která pomáhá k vyrovnávání průběhu proudu při velkých změnách napájecího napětí. Tlumivka se zároveň chová jako dolní propust.

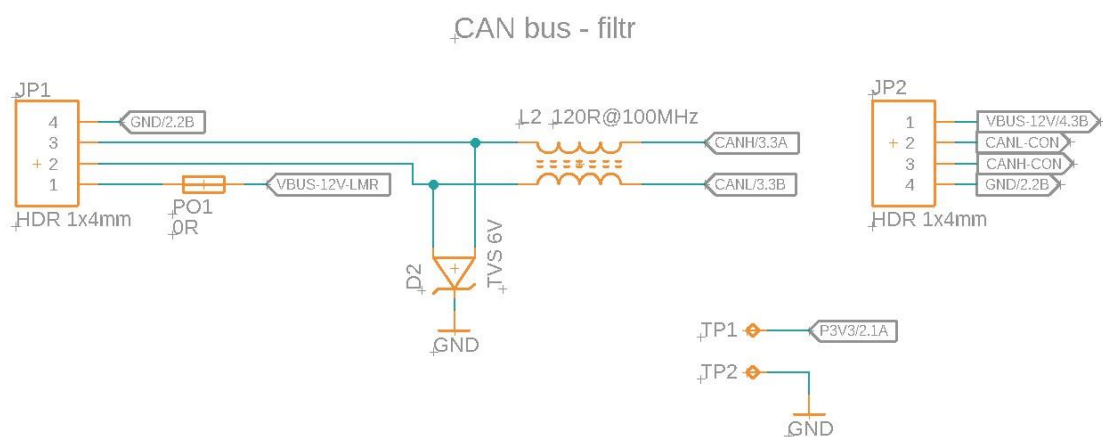


Obrázek 9 Symbol dvoucestné TVS diody



Obrázek 10 Symbol tlumivky souhlasného napětí

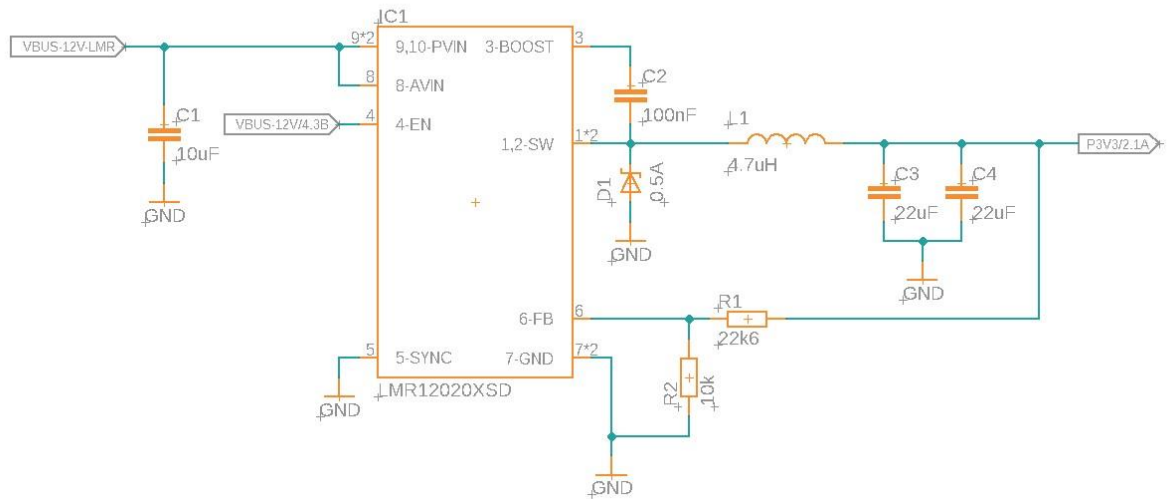
U výstupního konektoru jsme použili pro připojení také FCI konektor.



Obrázek 11 Schéma zapojení sběrnice CAN bus a napájení desky

2.4.3 Stabilizátor

Obvod stabilizátoru - napájecí cesta



Obrázek 12 Schéma zapojení stabilizátoru

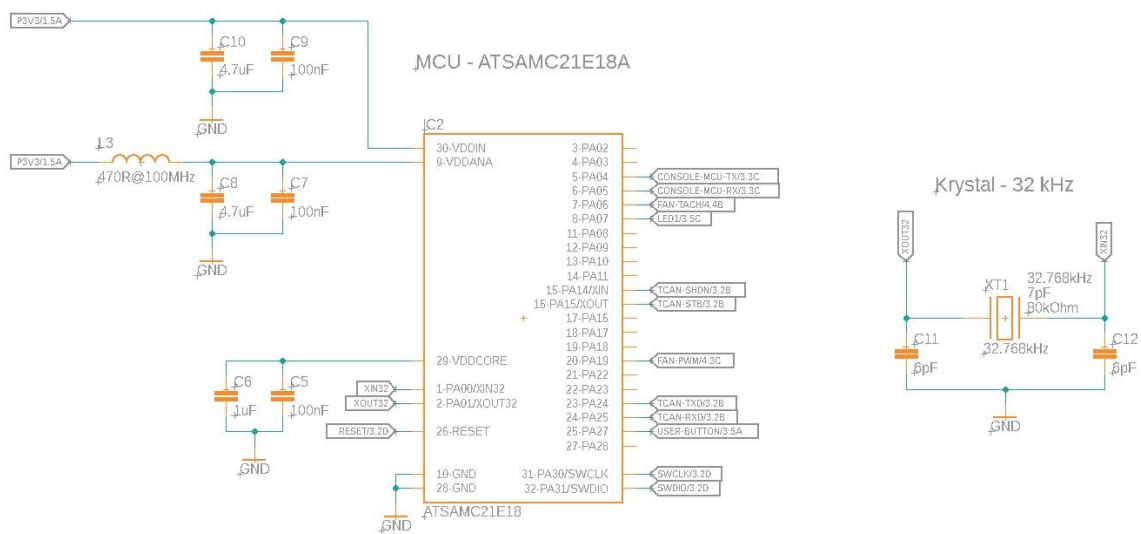
Jako zdroj napětí pro napájení každého modulu 3,3 V jsme zvolili spínaný DC/DC stabilizátor typu buck, který dokáže snížit napětí z původních 12 V na požadovaných 3,3 V pro napájení ostatních součástek na desce.

Obvod stabilizátoru obsahuje vstupní kondenzátor pro vyhlazení případného zvlněného napětí. Dále pak výkonovou cívku, která se snaží udržet stejnou velikost a směr proudu a po sepnutí integrovaného tranzistoru uvnitř stabilizátoru napětí na cívce obrací svoji polaritu. Také je zde obsažen napěťový dělič skládající se ze dvou rezistorů, který slouží k nastavení výstupního napětí. Obvod je zakončen výstupními kondenzátory pro odfiltrování nežádoucích jevů vytvořených spínáním stabilizátoru.

2.4.4 Mikrokontroler

Zvolili jsme 32bitový mikrokontroler ATSAMC21, který je založen na architektuře ARM Cortex M0+. Tuto architekturu jsme si vybrali, protože je u ní kladen důraz na minimální velikost mikrokontroleru a nízkou spotřebu. Tento mikrokontroler je velmi vhodný pro naše použití v průmyslovém sektoru, protože podporuje periférii CAN bus a několik dalších sériových sběrnic jako například I2C, UART nebo SPI. Dalším důvodem pro volbu tohoto mikrokontroleru je jeho schopnost pracovat i ve vyšších teplotách. Mikrokontroler je vybaven technologiemi správy napájení a režimu spánku, čímž lze dosáhnout velmi nízké spotřeby.

Mezi základní vlastnosti patří velikost programovací paměti flash 256 kB a frekvenci až 48 MHz. Další užitečnou funkcí může být bezpečnostní časovač Watchdog.



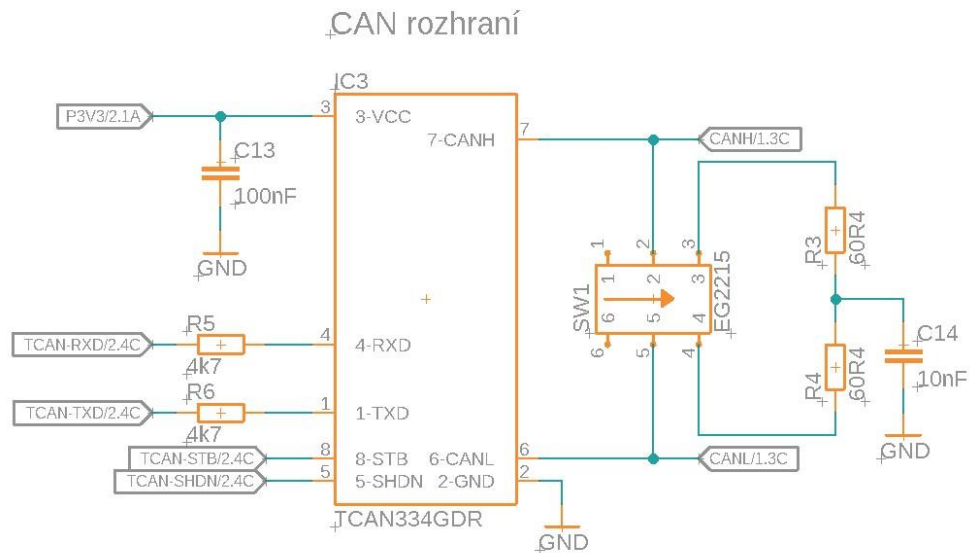
Obrázek 13 Schéma zapojení MCU a hodinového krystalu

K mikrokontroleru jsme zvolili 32 kHz krystal pro správnou funkci a práci s hodinovým signálem.

2.4.5 CAN bus rozhraní

Ačkoli mikrokontroler ATSAMC21 podporuje periférii CAN bus, jedná se pouze o tzv. CAN controller. Ke správnému fungování a komunikaci jednotlivých modulů mezi sebou je nutné, aby každý modul obsahoval CAN převodník. Jako převodník jsme zvolili součástku TCAN334GDR, protože podporuje naši 3.3 V logiku, kterou používáme i pro přenos pro sběrnici CAN. Díky této součástce jsme schopni převést data ze sběrnice CAN na UART, který již mikrokontroler zvládne rozpoznat.

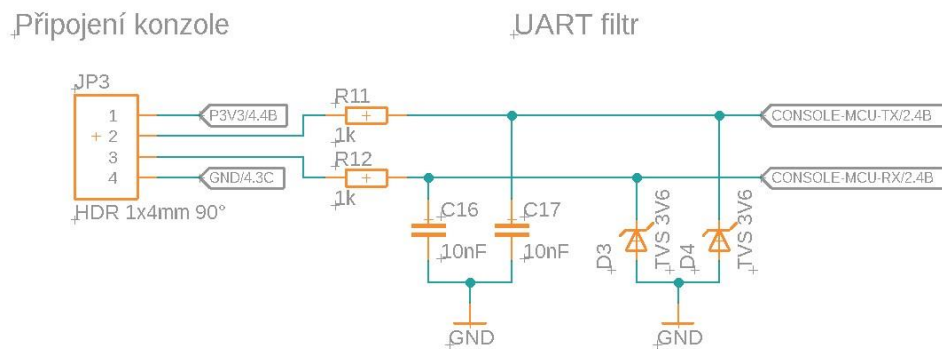
Na začátku i na konci sběrnice musí být umístěn takzvaný terminátor. Jelikož náš systém obsahuje více modulů a uživatel má možnost si vybrat jen ty, které chce, či potřebuje, tak nevíme, který modul bude na sběrnici poslední. Z toho důvodu se musí na každém modulu nacházet přepínač pro aktivaci nebo deaktivaci terminátoru. Uživatel tak musí dbát na to, aby vždy na posledním modulu byl terminátor aktivován, a tímto zajistit správnost komunikace mezi moduly.



Obrázek 14 Schéma zapojení CAN převodníku

2.4.6 Sériová konzole

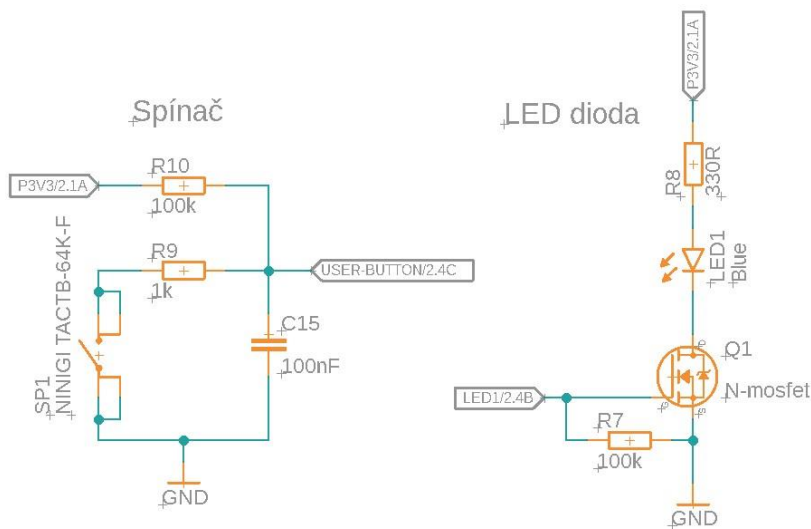
Z důvodu zamezení nežádoucích vlivů jsme za konektor pro sériovou konzolu, která slouží ke snadné komunikaci s deskou, umístili filtr přesně spočítaný na rychlost 9 600 baudů pro sběrnice UART. Za filtrem se nacházejí dvě TVS diody pro zabránění případnému přepětí.



Obrázek 15 Schéma zapojení sériové konzole

2.4.7 Uživatelské rozhraní

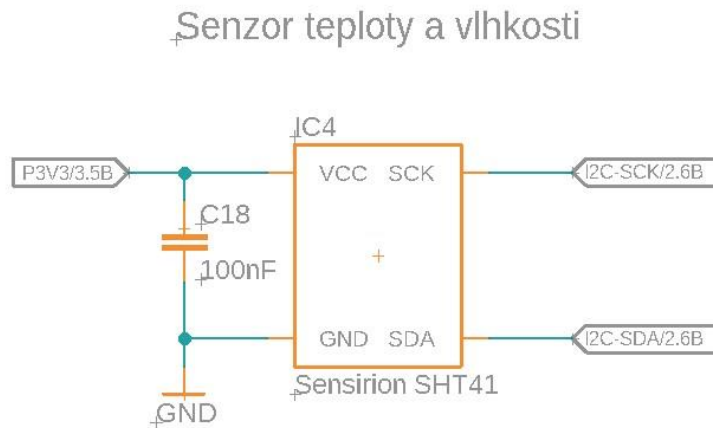
Každý modul obsahuje jedno tlačítko pro nutnost externí interakce s modulem. Dále každý modul disponuje indikační LED diodou pro jednoduchou a názornou odezvu desky.



Obrázek 16 Schéma zapojení spínače a LED diody

2.4.8 Senzor teploty a vlhkosti

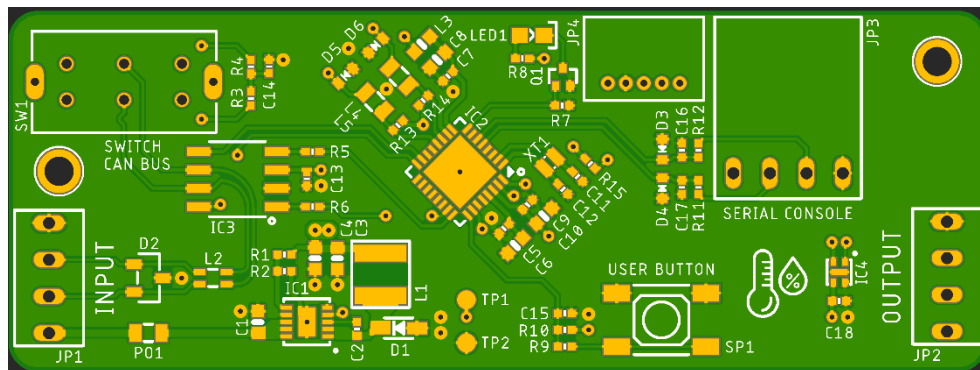
Pro měření hodnot teploty a vlhkosti jsme zvolili senzor SHT41 od společnosti Sensirion. Hlavními důvody jsou vysoká přesnost senzoru, a to až na 0,1 °C, nízká spotřeba a komunikace se senzorem pomocí sběrnice I2C.



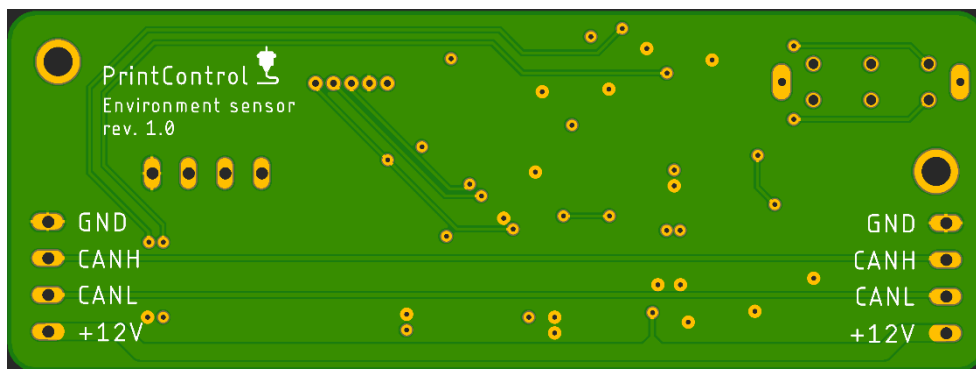
Obrázek 17 Schéma zapojení senzoru SHT41

2.4.9 Vizualizace modulu teploty a vlhkosti

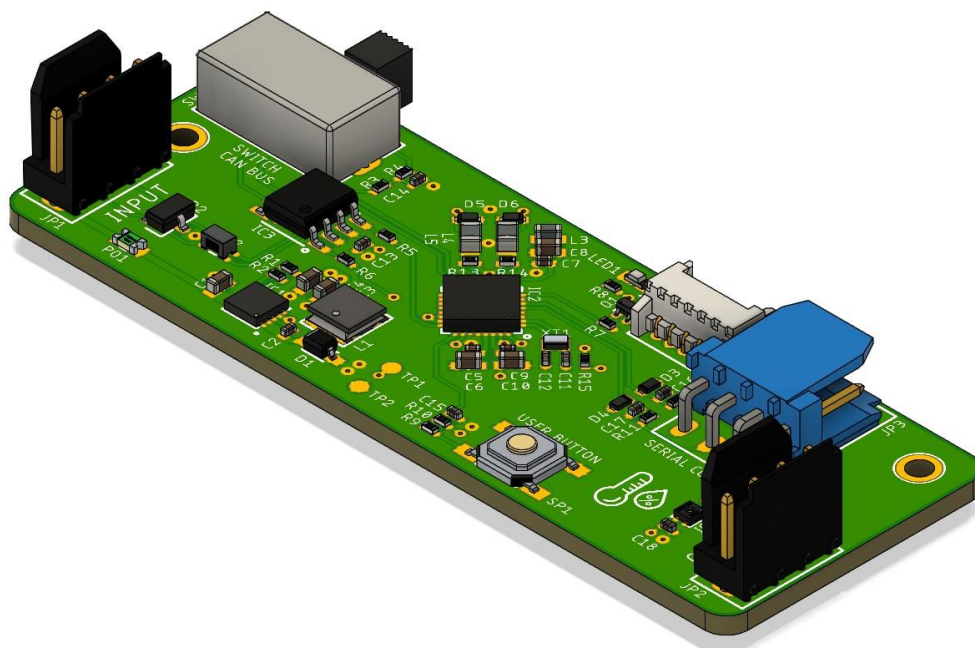
Na obrázcích 18 – 22 je vizualizace návrhu desky plošných spojů a 3D model krabičky pro modul teploty a vlhkosti.



Obrázek 18 PCB Environment sensor – pohled shora

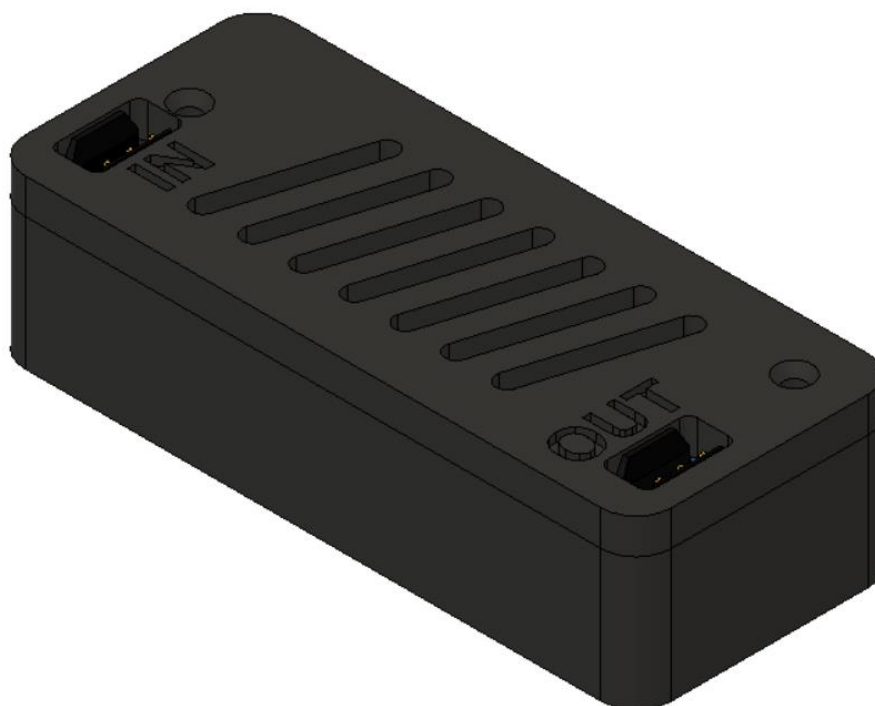


Obrázek 19 PCB Environment sensor – pohled zdola

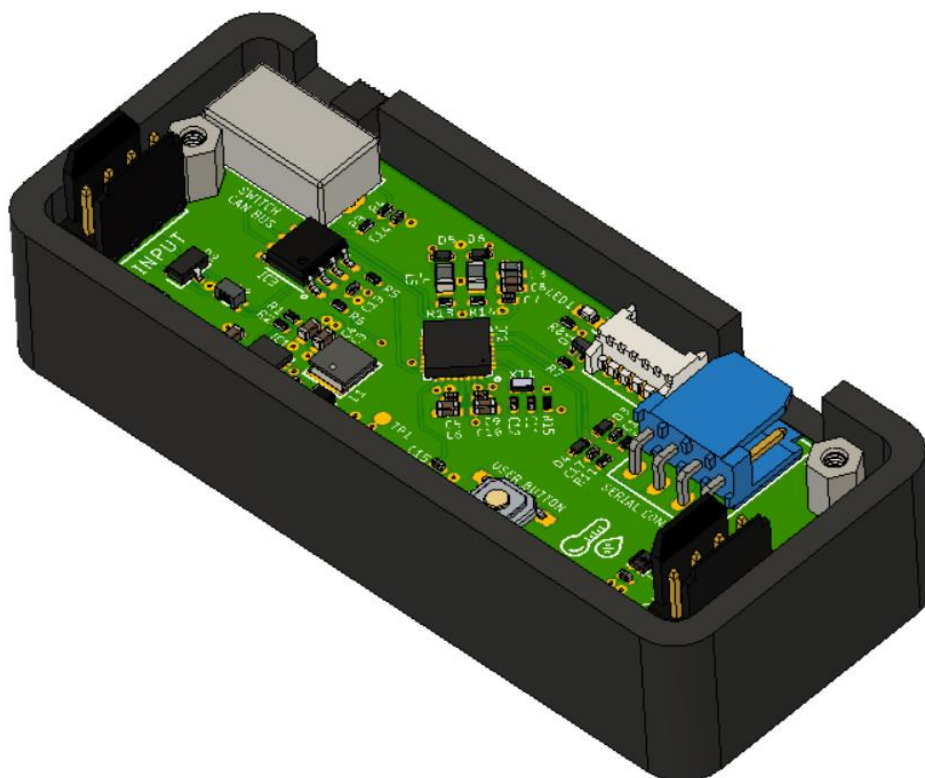


Obrázek 20 PCB Environment sensor – 3D vizualizace

Takto vypadá 3D vizualizace sestaveného modulu pro měření teploty a vlhkosti v boxu.



Obrázek 21 Environment sensor – uzavřená krabička



Obrázek 22 Environment sensor – otevřená krabička

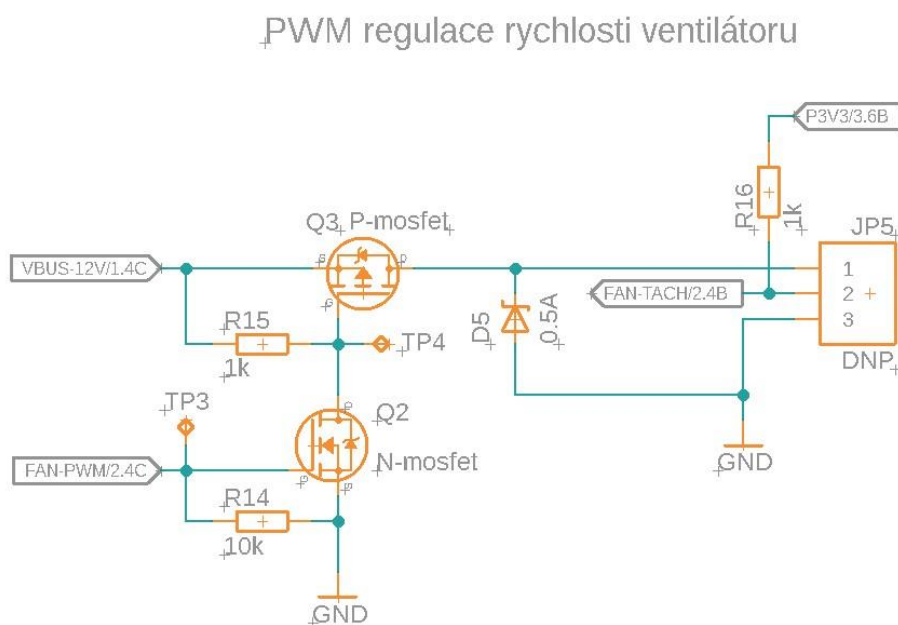
2.5 Modul pro regulaci prostředí – ventilátor

Jako již bylo zmíněno u prvního modulu, tak všechny desky plošných spojů, které jsou v projektu PrintControl obsaženy jsou navrženy tak, aby byli hlavní obvody a komponenty zůstaly ve stejné podobě na všech modulech. Díky tomu jsme schopni snížit náročnost programování všech modulů, návrhu dalších modulů i finálního zapojení v boxu.

2.5.1 Schéma zapojení pro ovládání ventilátoru

Pro regulaci rychlosti otáček ventilátoru jsme zvolili technologii PWM neboli pulzní šířkovou modulaci. Díky možnosti nastavení různé délky pulzu v čase a ideální frekvenci jsme schopni nastavit optimální rychlost ventilátoru.

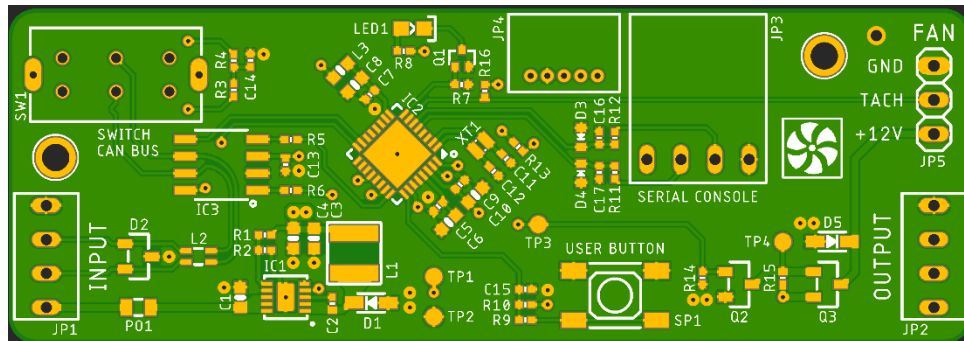
Obvod je tvořen dvěma tranzistory typu MOSFET. Jeden MOSFET je typu N, spínaný kladně nabitým nábojem a druhý typu P, spínaný záporně nabitým nábojem. MOSFET typu N jsme zvolili jako takzvaný low logic level MOSFET, abychom jej mohli ovládat přímo z mikrokontroleru. Na základě jeho sepnutí vůči zemi se sepne i MOSFET typu P, přes který následně protéká napětí 12 V pro napájení ventilátoru. Jelikož má ventilátor uvnitř motoru vinutí a chová se jako cívka, bylo zapotřebí před něj umístit Schottkyho diodu, abychom ochránili zbylé součástky v obvodu od zpětného napětí z cívky, které může být až několikanásobně vyšší, než je napájecí napětí ventilátoru.



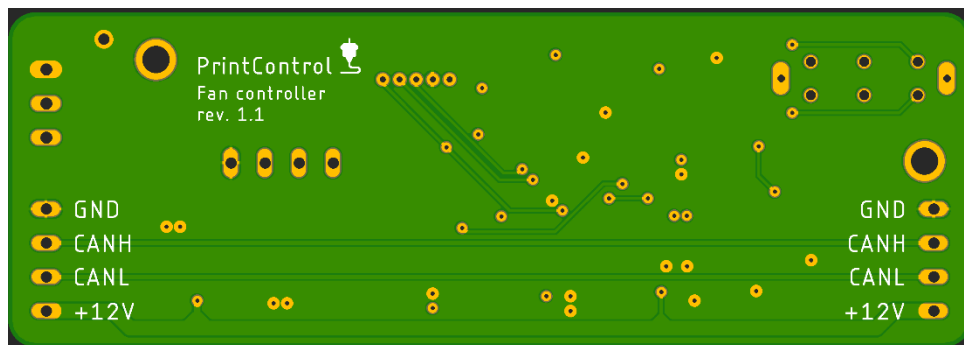
Obrázek 23 Schéma zapojení pro PWM regulaci ventilátoru

2.5.2 Vizualizace modulu pro regulaci prostředí – ventilátor

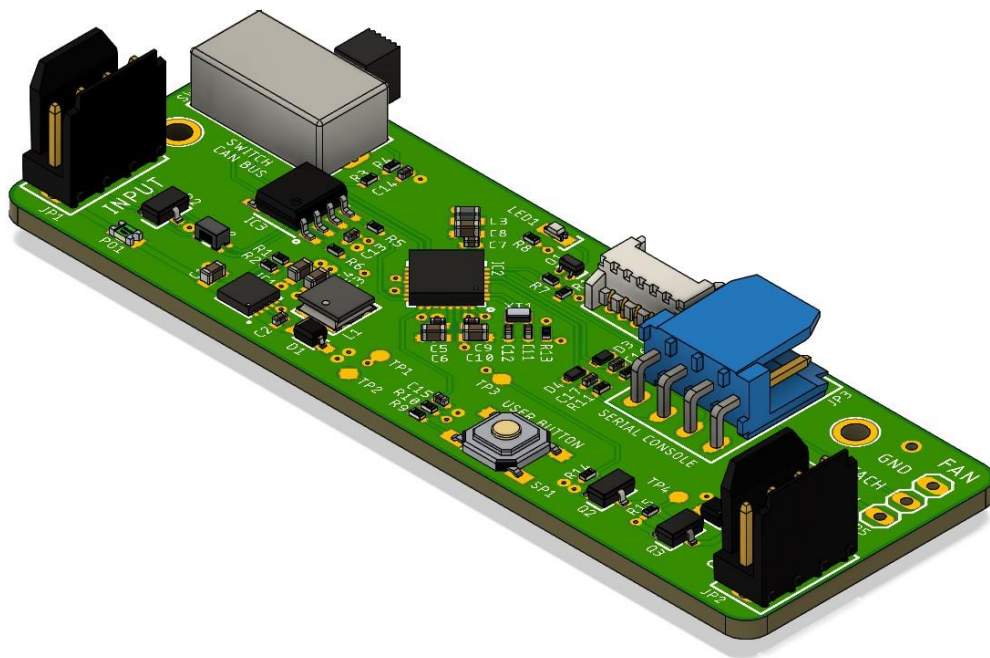
Na obrázcích 24 – 28 je vizualizace návrhu desky plošných spojů a 3D model krabičky pro modul s ventilátorem.



Obrázek 24 PCB Fan controller – pohled shora



Obrázek 25 PCB Fan controller – pohled zdola

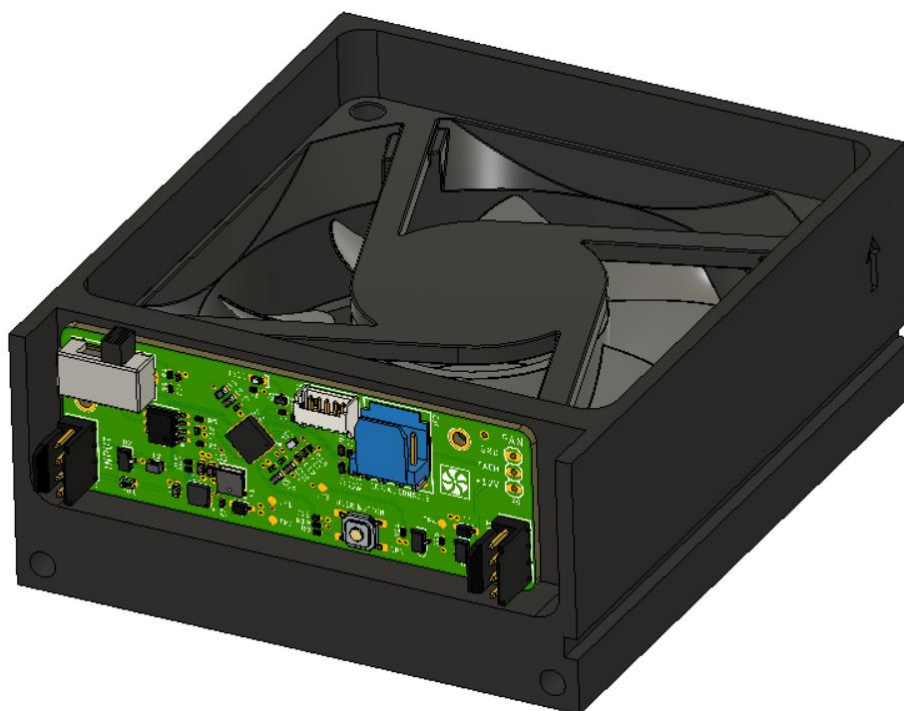


Obrázek 26 PCB Fan controller – 3D vizualizace

Takto vypadá 3D vizualizace sestaveného modulu s ventilátorem a deskou pro regulaci otáček ventilátoru.



Obrázek 27 Fan controller – uzavřená krabička

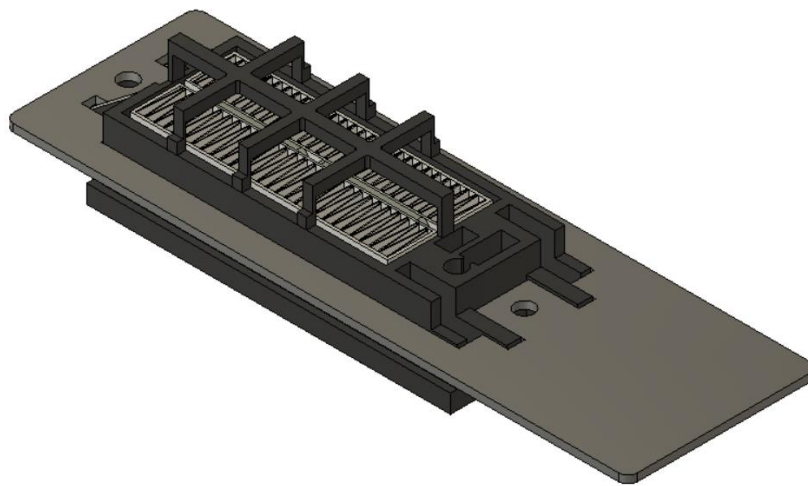


Obrázek 28 Fan controller – otevřená krabička

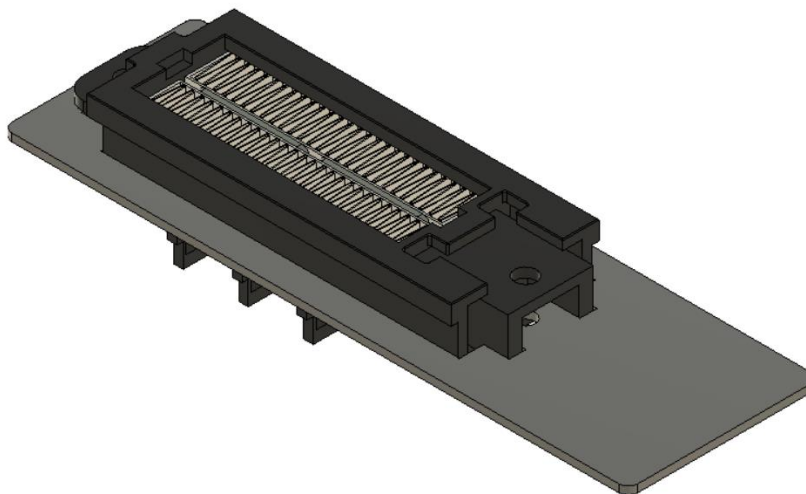
2.6 Modul pro regulaci prostředí – topné těleso

Tento modul je zatím pouze ve fázi návrhu a vývoje. Topné těleso bude tvořit PTC rezistivní zátěž a díky spínání výkonovým tranzistorem typu MOSFET budeme společně s PID regulací nastavovat výkon, a tudíž teplo jdoucí do boxu nastaví na požadovanou hodnotu teploty pro 3D tiskárnu. Ke správnému vytvoření PID regulace bude také potřeba připojit dva teplotní senzory, jeden pro měření vstupní a druhý pro měření výstupní teploty.

Takto vypadá hrubý návrh PCB s upevněním na topné těleso:



Obrázek 29 Heater module – pohled shora



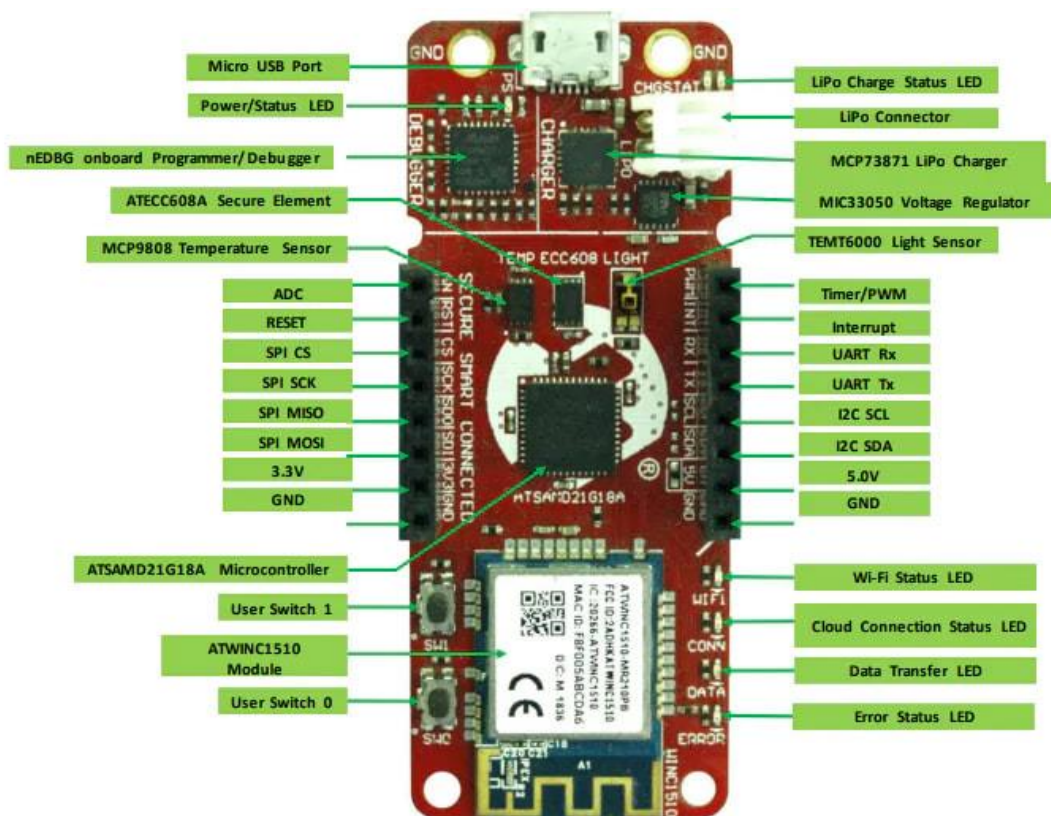
Obrázek 30 Heater module – pohled zdola

2.7 Modul pro komunikaci – gateway

Tento modul má sloužit pro přenos dat mezi boxem a aplikací pomocí Wi-Fi technologie. V rámci prototypu jsme tento gateway modul vyřešili tak, že jsme stejné zapojení jako u ostatních modulů navrhli speciálně jako mikroBUS pro vývojovou desku SAM-IoT-WG, která má potřebný Wi-Fi modul obsahuje. MikroBUS modul se skládá ze stejných součástek jako ostatní moduly a komunikuje také na sběrnici CAN bus. Potřebná data pak přenáší po sériové sběrnici UART na desku SAM-IoT-WG, ze které jsou data dále pomocí Wi-Fi modulu odesílána do aplikace.

2.7.1 Vývojová deska SAM-IoT-WG od společnosti Microchip

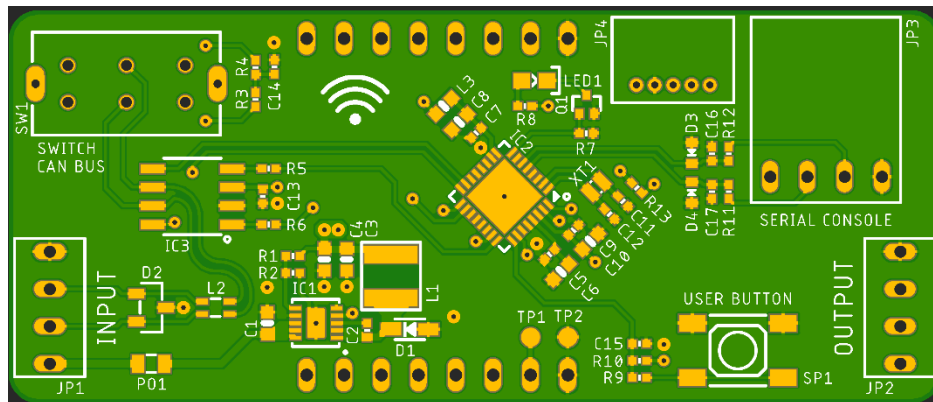
Na obrázku je zobrazen popis jednotlivých částí vývojové desky SAM-IoT-WG.



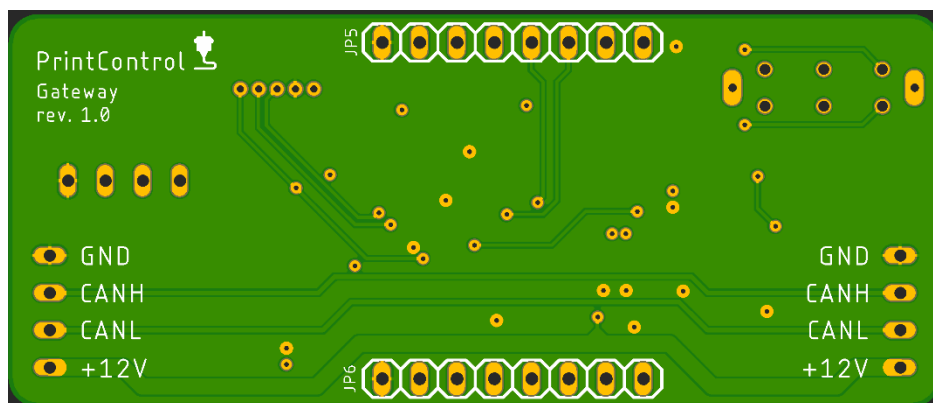
Obrázek 31 Vývojová deska SAM-IoT-WG

2.7.2 Vizualizace modulu gateway

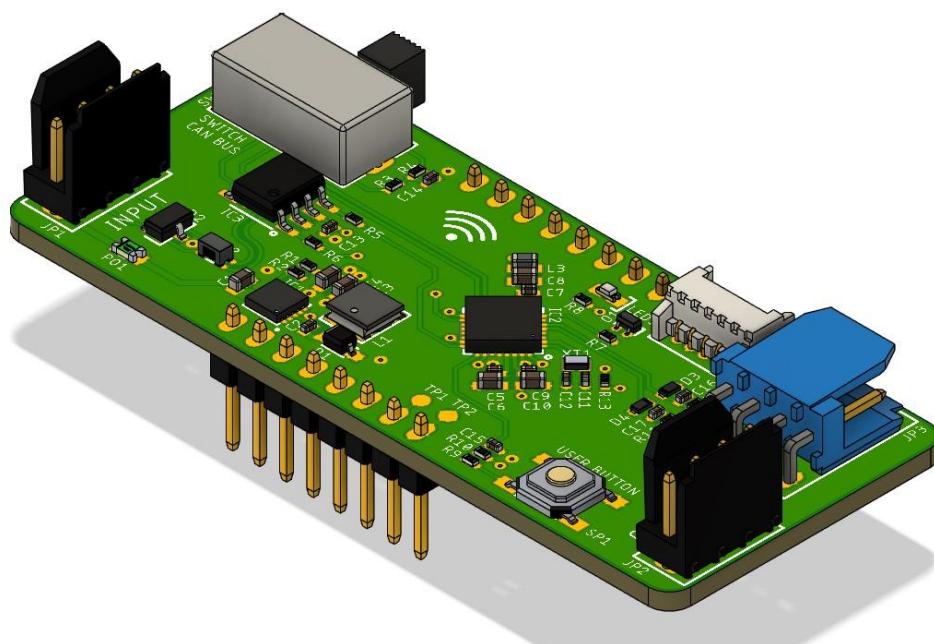
Níže se nachází vizualizace návrhu desky plošných spojů mikroBUS modulu pro vývojovou desku SAM-IoT-WG.



Obrázek 32 PCB Gateway – pohled shora



Obrázek 33 PCB Gateway – pohled zdola

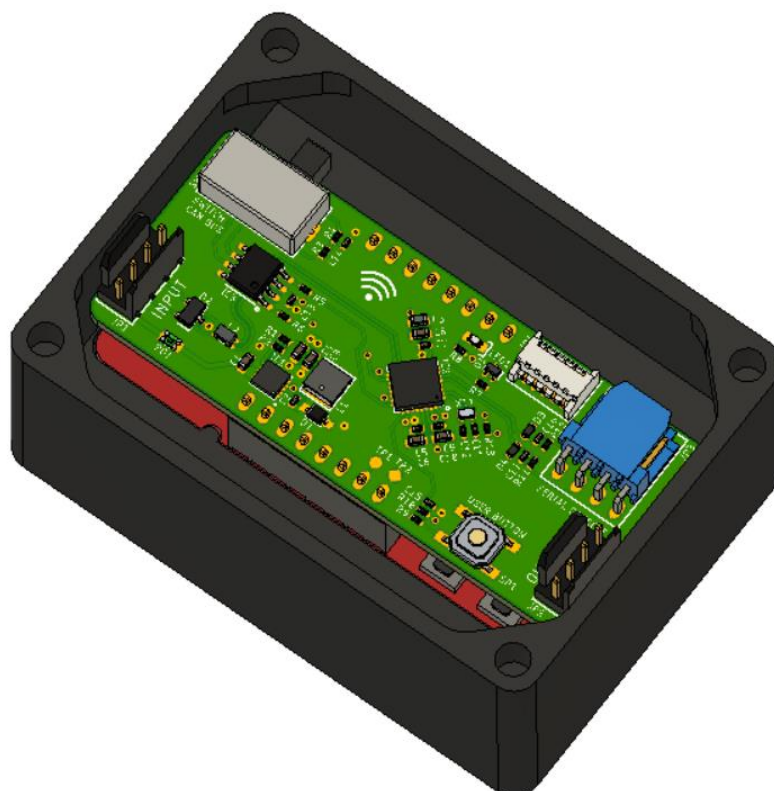


Obrázek 34 PCB Gateway – 3D vizualizace

Takto vypadá 3D vizualizace sestaveného gateway modulu.



Obrázek 35 Gateway – uzavřená krabička



Obrázek 36 Gateway – otevřená krabička

3 Programovací část projektu

3.1 Programovací prostředí MPLAB X IDE

K programování všech námi vytvořených modulů jsme použili program MPLAB X IDE s Harmony 3 pluginem. Nyní popíšu, co MPLAB X IDE a Harmony 3 umí a jak založit projekt.

MPLAB X IDE je program, který slouží k vývoji firmwaru embedded zařízení. Je určen hlavně pro 8,16 a 32bitové mikrokontrolery, které vyrábí a vyvíjí společnost Microchip. V programu MPLAB X IDE se jako programovací jazyk používá C.

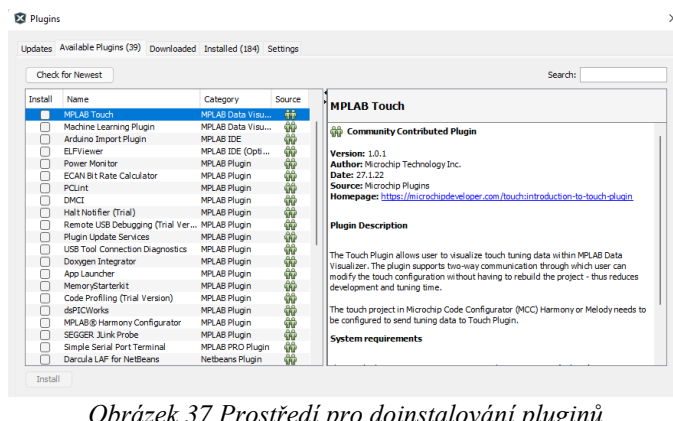
Výhodou programu je také to, že umožňuje vytvořit vlastní kód a následně jej spustit v simulačním režimu. Znamená to, že k počítači není třeba mít fyzicky připojenou vývojovou desku, nebo jiné zařízení, které chceme programovat, ale můžeme simulaci funkčnosti pouze nasimulovat. Jeho nevýhodou je jeho omezená funkčnost, která nám neumožňuje použít všechny možnosti zvoleného mikrokontroleru

Jestliže je k počítači fyzicky připojená vývojová deska nebo jiné zařízení, tak je možné kód do mikrokontroleru odeslat a zjistit, zda nám funguje. Vývojové desky většinou mají integrovaný obvod, kde pomocí USB konektoru lze kód do mikrokontroleru odeslat. V případě jiného zařízení, které nemá obvod, přes který odešleme kód do mikrokontroleru, musíme použít SWD rozhraní pro programování a použít k tomu nějaký debugger, který k SWD rozhraní připojíme.

Do programu MPLAB X je možné si doinstalovat další pluginy, které nám umožní rozšíření funkcí programu. Je možné si doinstalovat například MPLAB Harmony 3 Configurator nebo MPLAB Code Configurator.

3.2 MPLAB Harmony 3

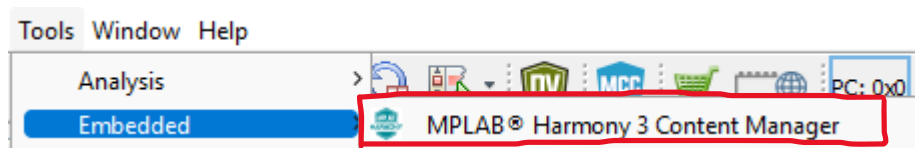
Prostředí MPLAB Harmony 3 je plugin, který vyvinula firma Microchip a integrovala jej do vývojového prostředí MPLAB X IDE. Podporuje pouze 32bitové mikrokontrolery od společnosti Microchip.



Obrázek 37 Prostředí pro doinstalování pluginů

3.2.1 Jak začít s Harmony 3

Poté, co nainstalujeme plugin Harmony 3, ještě pro jeho funkčnost musíme doinstalovat několik modulů v Content Manageru.

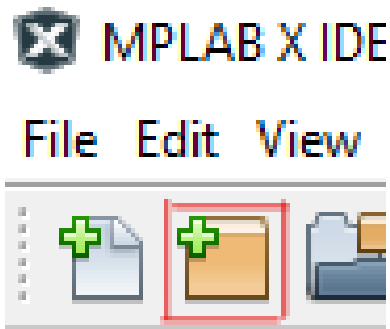


Obrázek 38 Postup, jak otevřít Content Manager

Pro základní funkčnost musíme doinstalovat modul MHC, BSP a CSP. Je možné si z nabídky doinstalovat i další moduly. Například modul pro Wi-Fi komunikaci, řízení motorů nebo pro dotyková tlačítka. Je zde také možnost stáhnout vzorové příklady projektů pro podporované mikrokontrolery.

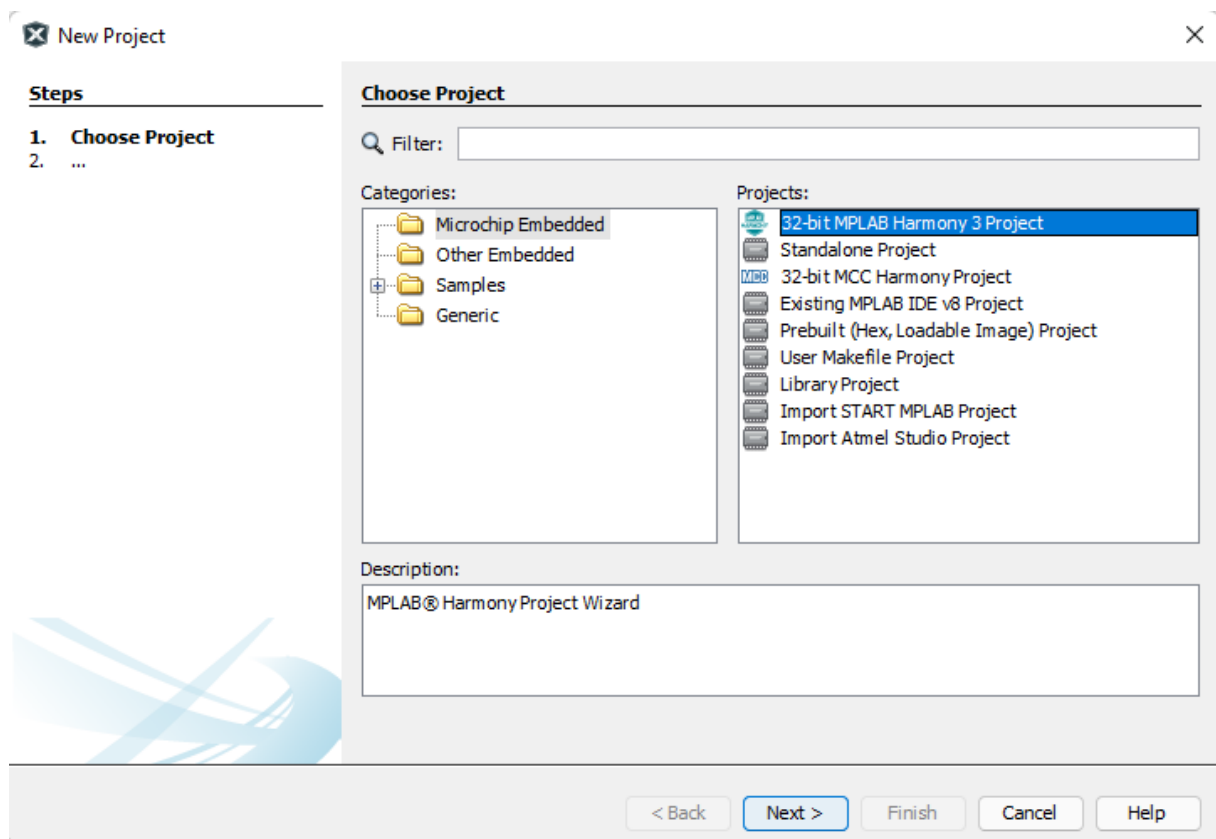
3.3 Založení projektu v MPLAB X IDE

Pokud chceme založit nový projekt, nejdříve musíme kliknout na zvýrazněné tlačítko.



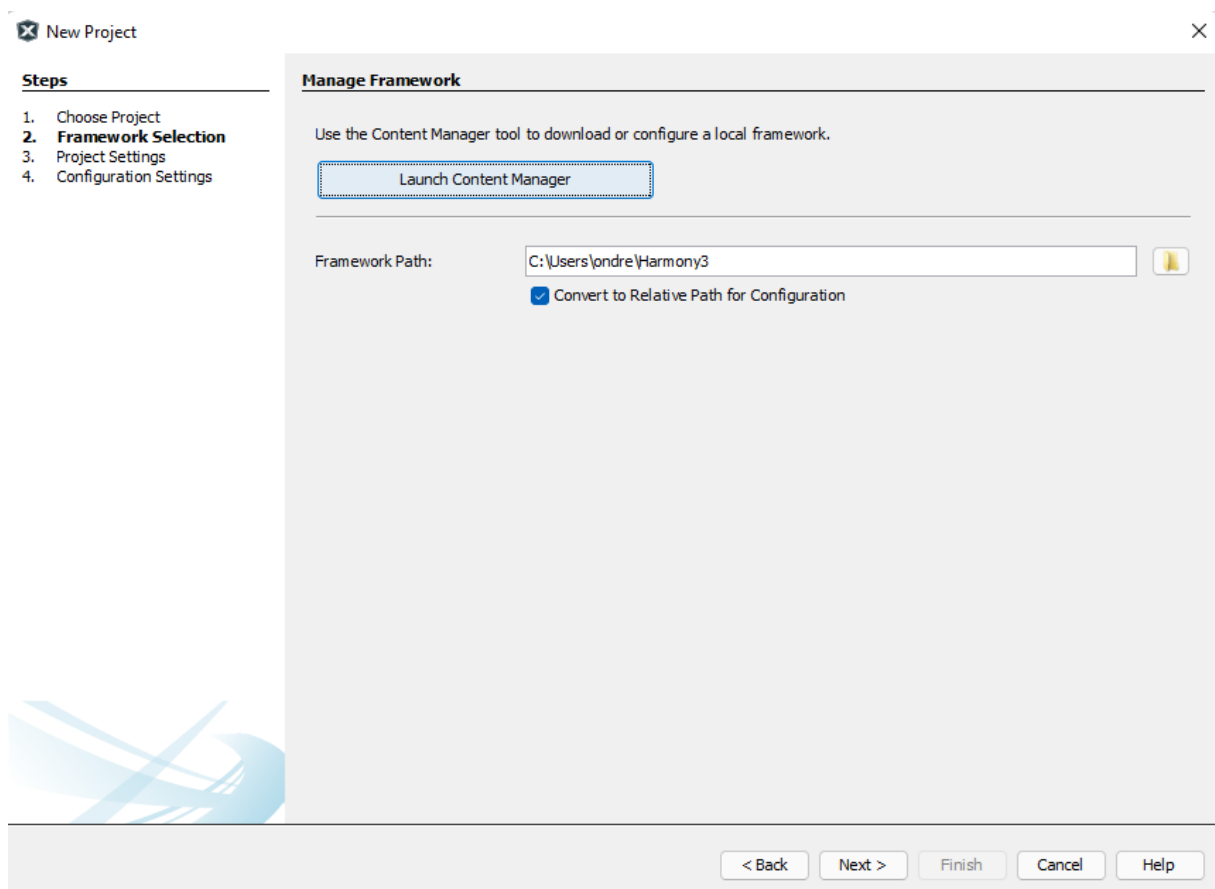
Obrázek 39 Tlačítko pro založení nového projektu

Následně se nám zobrazí tato tabulka, ve které kliknutím vybereme 32-bit MPLAB Harmony 3 Project



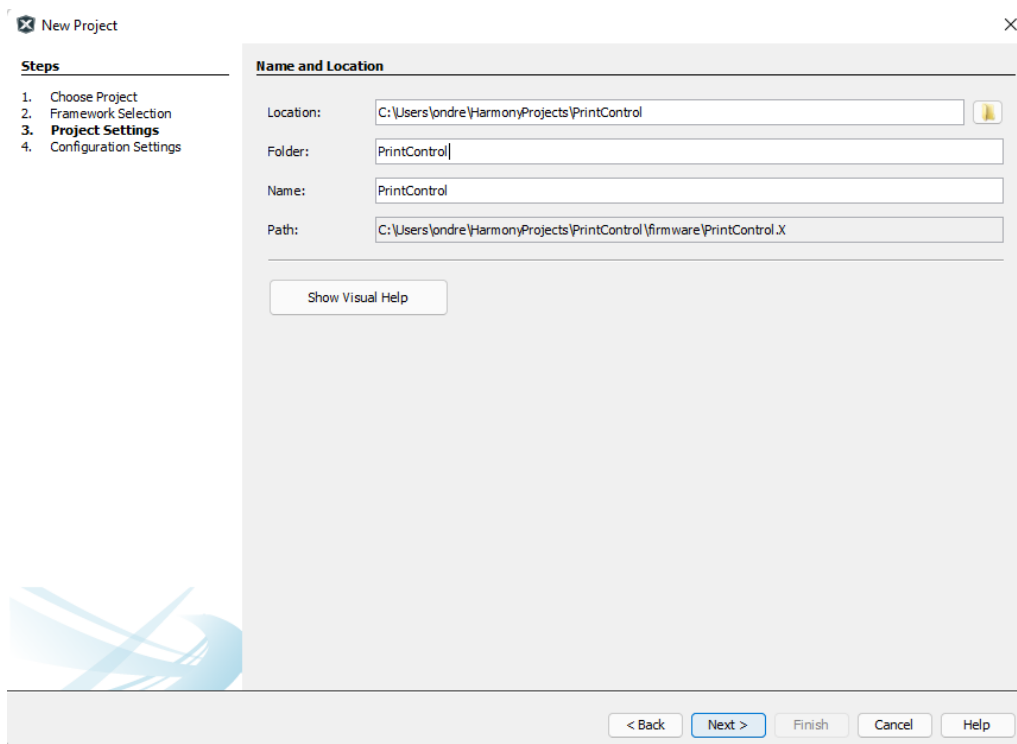
Obrázek 40 Návod na založení projektu – 1

Poté, co klikneme na tlačítko Next, objeví se tato tabulka. Opět klikneme na tlačítko Next.



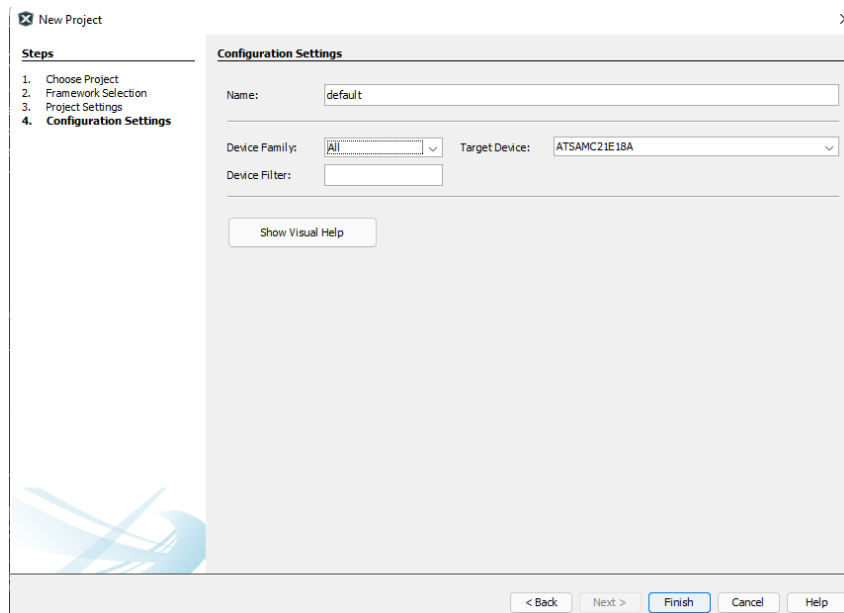
Obrázek 41 Návod na založení projektu – 2

Následně musíme projekt pojmenovat a uložit do našeho adresáře.



Obrázek 42 Návod na založení projektu – 3

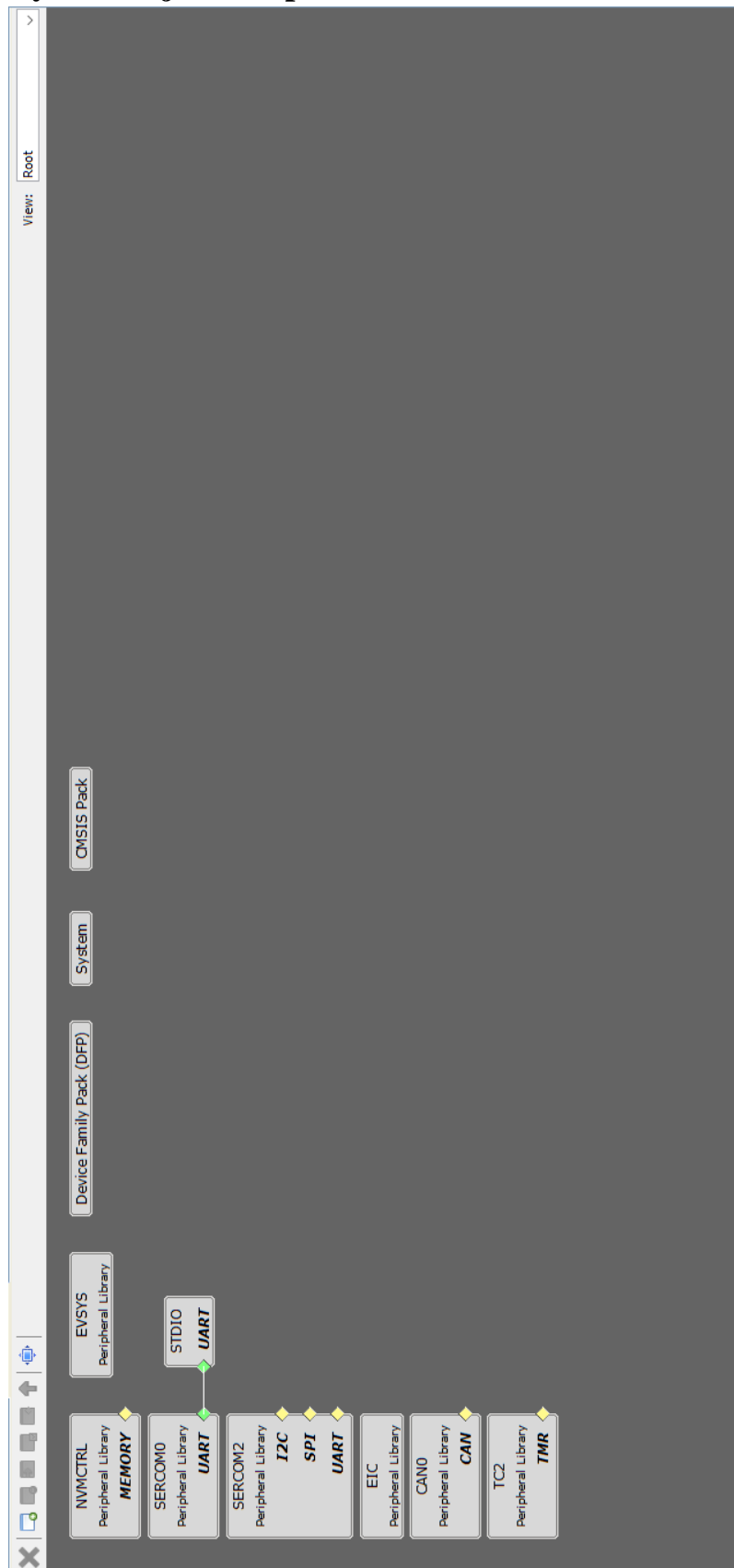
A nakonec si musíme zvolit mikrokontroler, který chceme použít. Po vybrání mikrokontroleru klikneme na tlačítko Finish.



Obrázek 43 Návod na založení projektu – 4

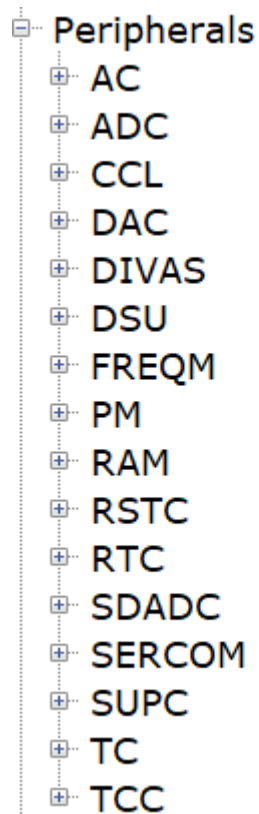
Tímto máme založený program. Poté se spustí Harmony 3 Configurator.

3.3.1 Harmony 3 – Project Graph



Obrázek 44 Harmony 3 – Project Graph

3.3.2 Harmony 3 – periferie



Obrázek 45 Seznam periferií MCU SAMC21E18A

Když máme představu o tom, co v projektu budeme dělat, musíme si přidat jednotlivé periferie.

V tomto případě chceme číst data ze senzoru SHT41 od společnosti Sensirion, data se dále vypíší do konzole a na CAN bus sběrnici. Také tam musíme přidat timer TC2, pro nastavení frekvence odesílání. Ještě je třeba přidat periferie EIC, které sleduje stav tlačítka.

3.3.3 Nastavení pinů mikrokontroleru

V Harmony 3 jsme schopni si nastavit funkci jednotlivých pinů mikrokontroleru. Funkce jednotlivých pinů jsme schopni zjistit z datasheetu mikrokontroleru. Nastavení pinů lze otevřít přes MHC → Tools → Pin Configuration.

Pin Number	Pin ID	Custom Name	Function	Mode	Direction	Latch	Pull Up	Pull Down	Drive Strength
1	PA00	XIN32	OSC32KCTRL_XIN32	Digital	High Impedance	n/a	<input type="checkbox"/>	<input type="checkbox"/>	NORMAL
2	PA01	XOUT32	OSC32KCTRL_XOUT32	Digital	High Impedance	n/a	<input type="checkbox"/>	<input type="checkbox"/>	NORMAL
3	PA02		Available	Digital	High Impedance	Low	<input type="checkbox"/>	<input type="checkbox"/>	NORMAL
4	PA03		Available	Digital	High Impedance	Low	<input type="checkbox"/>	<input type="checkbox"/>	NORMAL
5	PA04	CONSOLE_TX	SERCOM0_PAD0	Digital	High Impedance	n/a	<input type="checkbox"/>	<input type="checkbox"/>	NORMAL
6	PA05	CONSOLE_RX	SERCOM0_PAD1	Digital	High Impedance	n/a	<input type="checkbox"/>	<input type="checkbox"/>	NORMAL
7	PA06		Available	Digital	High Impedance	Low	<input type="checkbox"/>	<input type="checkbox"/>	NORMAL
8	PA07	LED1	GPIO	Digital	Out	Low	<input type="checkbox"/>	<input type="checkbox"/>	NORMAL
9	VDDANA			Digital	High Impedance	Low	<input type="checkbox"/>	<input type="checkbox"/>	NORMAL
10	GNDIO			Digital	High Impedance	Low	<input type="checkbox"/>	<input type="checkbox"/>	NORMAL
11	PA08	I2C_SDA	SERCOM2_PAD0	Digital	High Impedance	n/a	<input type="checkbox"/>	<input type="checkbox"/>	NORMAL
12	PA09	I2C_SCL	SERCOM2_PAD1	Digital	High Impedance	n/a	<input type="checkbox"/>	<input type="checkbox"/>	NORMAL
13	PA10		Available	Digital	High Impedance	Low	<input type="checkbox"/>	<input type="checkbox"/>	NORMAL
14	PA11		Available	Digital	High Impedance	Low	<input type="checkbox"/>	<input type="checkbox"/>	NORMAL
15	PA14	CAN_SHDN	GPIO	Digital	Out	Low	<input type="checkbox"/>	<input type="checkbox"/>	NORMAL
16	PA15	CAN_STB	GPIO	Digital	Out	Low	<input type="checkbox"/>	<input type="checkbox"/>	NORMAL
17	PA16		Available	Digital	High Impedance	Low	<input type="checkbox"/>	<input type="checkbox"/>	NORMAL
18	PA17		Available	Digital	High Impedance	Low	<input type="checkbox"/>	<input type="checkbox"/>	NORMAL
19	PA18		Available	Digital	High Impedance	Low	<input type="checkbox"/>	<input type="checkbox"/>	NORMAL
20	PA19		Available	Digital	High Impedance	Low	<input type="checkbox"/>	<input type="checkbox"/>	NORMAL
21	PA22		Available	Digital	High Impedance	Low	<input type="checkbox"/>	<input type="checkbox"/>	NORMAL
22	PA23		Available	Digital	High Impedance	Low	<input type="checkbox"/>	<input type="checkbox"/>	NORMAL
23	PA24	CAN_TX	CAN0_TX	Digital	High Impedance	n/a	<input type="checkbox"/>	<input type="checkbox"/>	NORMAL
24	PA25	CAN_RX	CAN0_RX	Digital	High Impedance	n/a	<input type="checkbox"/>	<input type="checkbox"/>	NORMAL
25	PA27	SW0	EIC_EXTINT15	Digital	High Impedance	n/a	<input type="checkbox"/>	<input type="checkbox"/>	NORMAL
26	RESET_N			Digital	High Impedance	Low	<input type="checkbox"/>	<input type="checkbox"/>	NORMAL
27	PA28		Available	Digital	High Impedance	Low	<input type="checkbox"/>	<input type="checkbox"/>	NORMAL
28	GNDIO			Digital	High Impedance	Low	<input type="checkbox"/>	<input type="checkbox"/>	NORMAL
29	VDDCORE			Digital	High Impedance	Low	<input type="checkbox"/>	<input type="checkbox"/>	NORMAL
30	VDDIN			Digital	High Impedance	Low	<input type="checkbox"/>	<input type="checkbox"/>	NORMAL
31	PA30		Available	Digital	High Impedance	Low	<input type="checkbox"/>	<input type="checkbox"/>	NORMAL
32	PA31		Available	Digital	High Impedance	Low	<input type="checkbox"/>	<input type="checkbox"/>	NORMAL

Obrázek 46 Harmony 3 – nastavení pinů

3.4 MPLAB Snap In-Circuit Debugger

Pro programování našich desek potřebujeme nějaký debugger. Na výběr jsme měli ze 2 možností. V našem případě jsme zvolili MPLAB Snap In-Circuit Debugger. Zvolili jsme jej z důvodu přijatelné ceny – 30 €. Na rozdíl od podobného MPLAB PICkit 4 In-Circuit Debugger, který stojí téměř 70 € a nabízí podobné funkce jako MPLAB Snap In-Circuit Debugger.

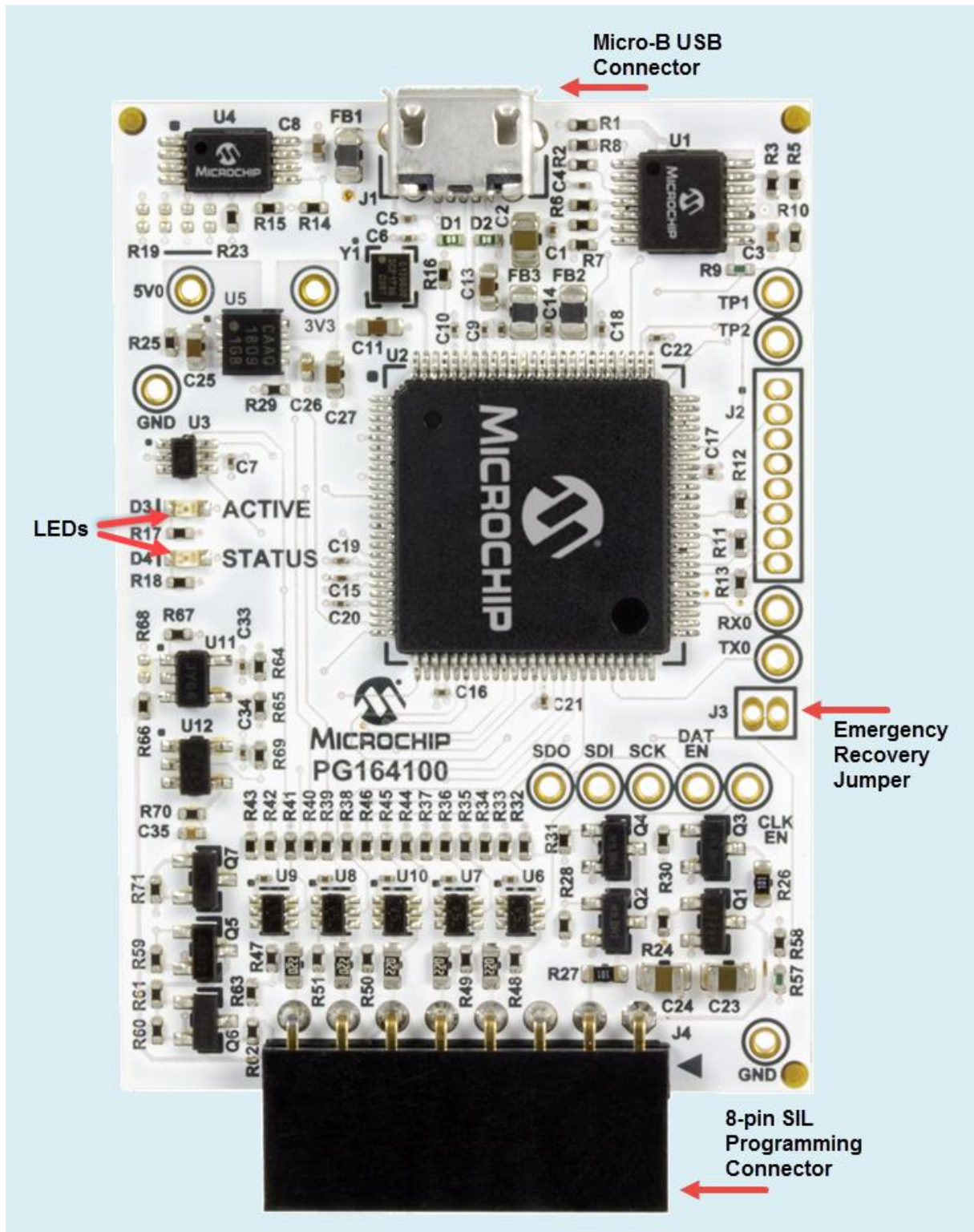
Jeho výhodou je kompatibilita s mikrokontrolery řad PIC, PIC32, SAM, AVR a dsPIC.

Jestliže chceme naprogramovat mikrokontroler, do počítače si připojíme USB kabel a druhý konec, na kterém musí být microUSB konektor, připojíme do MPLAB Snap In-Circuit Debuggeru. Z mikrokontroleru, který chceme programovat, tak musíme mít vyvedeno celkem 5 vodičů. Potřebujeme propojit piny SWD, SWCLK a RESET. Dále voltage target pin a ground pin pro uzemnění. K 8 pinovému konektoru připojíme pouze ty piny, které potřebujeme. V našem případě musíme použít Cortex SWD, protože SAMC21E18A používá ARM Cortex M0+ architekturu.

Table 10-4. Pinouts for Debug Interfaces

MPLAB Snap			DEBUG								
Connector	Pin #	Pin Name	ICSP (MCHP)	MIPS EJTAG	CORTEX® SWD	AVR® JTAG	AVR ISP(&DW)	UPDI	PDI	debugWIRE	TPI
	1	TVPP	MCLR	MCLR	MCLR						
	2	TVDD	VDD	VIO_REF	VTG	VTG	VTG	VTG	VTG	VTG	VTG
	3	GND	GND	GND	GND	GND	GND	GND	GND	GND	GND
	4	PGD	DAT	TDO	SWO	TDO	MISO	DAT	DAT		DAT
	5	PGC	CLK	TCK	SWCLK	TCK	SCK				CLK
	6	TAUX	AUX			RESET	RESET		CLK	dW	RST
	7	TTDI		TDI		TDI	MOSI				
	8	TTMS		TMS	SWDIO	TMS					

Obrázek 47 Tabulka „Debug Interfaces“



Obrázek 48 MPLAB Snap In – Circuit Debugger

3.5 Kód pro modul teploty a vlhkosti

Následující kód popisuje program pro environment modul. Další kódy k Fan modulu a Gateway modulu jsou obdobné.

3.5.1 Definice globálních proměnných

```
19 #include <stddef.h> // Defines NULL
20 #include <stdbool.h> // Defines true
21 #include <stdlib.h> // Defines EXIT_FAILURE
22 #include "definitions.h" // SYS function prototypes
23
24 /*Proměnné pro I2C*/
25 #define SHT4x_I2C_ADDRESS 0x44
26 #define SHT4x_TH_MEASURE 0xFD
27
28 uint8_t i2c_data_read[8];
29 uint8_t i2c_data_write[8];
30
31 volatile bool i2c_data_ready = false;
32 volatile bool i2c_data_error = false;
33
34 /*Proměnné pro telemetrii*/
35 uint16_t ID = 0x0106;
36 uint16_t FW_VERSION = 100;
37 uint16_t HOURS;
38 uint16_t SECONDS;
39
40 /*Proměnné pro výpočet teploty a vlhkosti*/
41 float temperature;
42 float humidity;
43 float t;
44 float h;
45 uint16_t checksum_t;
46 uint16_t checksum_h;
47
48 /*Proměnné pro callback funkce*/
49 volatile bool SW0 = false;
50 volatile bool timer = false;
```

Obrázek 49 Nastavení globálních proměnných

Globální proměnné je nutné nastavit na začátku kódu před funkcí main. Později je využíváme dále v kódu.

3.5.2 Funkce pro čtení a zápis na CAN bus sběrnici

```
57 void CAN_WRITE(void)
58 {
59     uint8_t txFiFo[CANO_TX_FIFO_BUFFER_SIZE];
60     CAN_TX_BUFFER *txBuffer = (CAN_TX_BUFFER *)txFiFo;
61     memset(txFiFo, 0x00, CANO_TX_FIFO_BUFFER_ELEMENT_SIZE);
62     txBuffer->id = 0x06;
63     txBuffer->dlc = 8; //zjistit se, jak funguje dlc code
64     txBuffer->fdcf = 1; //zjistit se, jak nastavit RX filtry
65     txBuffer->brs = 1; //jak funguje TX FIFO settings v harmony 3 - No. of elements & element size
66     txBuffer->data[0] = temperature;
67     txBuffer->data[1] = humidity;
68
69     // Transmit message from Tx FIFO
70     CANO_MessageTransmitFifo(2, txBuffer);
71     printf("DATA [0]: %d, DATA [1]: %d\r\n", txBuffer->data[0], txBuffer->data[1]);
72 }
73
74 void CAN_READ(void)
75 {
76     uint8_t rxFiFo0[CANO_RX_FIFO0_SIZE];
77     memset(rxFiFo0, 0x00, CANO_RX_FIFO0_ELEMENT_SIZE);
78
79     // Read message from Rx FIFO 0
80     CANO_MessageReceiveFifo(CANO_RX_FIFO_0, 2, (CAN_RX_BUFFER *)rxFiFo0);
81     printf("RX DATA [0]: %d; DATA [1]: %d\r\n", rxFiFo0[0], rxFiFo0[1]);
82 }
```

Obrázek 50 Funkce pro čtení a zápis na CAN sběrnici

Tyto dvě funkce nám umožňují číst a zapisovat na CAN bus sběrnici. Obě funkce lze dohledat ze souboru CSP, který se stáhne společně s Harmony 3 pluginem.

3.5.3 Funkce pro SERCOM 2, TC2 a tlačítko SW0

```
93 void SERCOM2_I2C_Callback (uintptr_t context)
94 {
95     if(SERCOM2_I2C_ErrorGet() == SERCOM_I2C_ERROR_NONE)
96     {
97         //Transfer is completed successfully
98         i2c_data_ready = true;
99     }
100     else
101     {
102         //Error occurred during transfer
103         i2c_data_error = true;
104     }
105 }
106
107 void TC2_Callback(TC_TIMER_STATUS status, uintptr_t context)
108 {
109     timer = true;
110 }
111
112 void SW0_Callback (uintptr_t context)
113 {
114     SW0 = true;
115 }
```

Obrázek 51 Callback funkce

3.5.4 Funkce pro získání dat ze senzoru a výpočet teploty a vlhkosti

```
117 void SHT4x_MEASUREMENT (void)
118 {
119     i2c_data_write[0] = SHT4x_TH_MEASURE;
120     SERCOM2_I2C_Write(SHT4x_I2C_ADDRESS,i2c_data_write,1);
121     while(!i2c_data_ready){}
122     i2c_data_ready = false;
123     SERCOM2_I2C_Read(SHT4x_I2C_ADDRESS,i2c_data_read,6);
124     while(!i2c_data_ready){}
125     i2c_data_ready = false;
126
127     t = i2c_data_read[0] * 256 + i2c_data_read[1];
128     checksum_t = i2c_data_read[2];
129     h = i2c_data_read[3] * 256 + i2c_data_read[4];
130     checksum_h = i2c_data_read[5];
131
132 }
133
134 float SHT4x_TEMPERATURE_CALCULATE(void)
135 {
136     temperature = -45 + 175 * t/65535;
137     return(temperature);
138 }
139
140 float SHT4x_HUMIDITY_CALCULATE(void)
141 {
142     humidity = -6 + 125 * h/65535;
143     return(humidity);
144 }
```

Obrázek 52 Čtení a zápis na I2C, výpočet teploty a vlhkosti

Ve funkcích jsme použili vzorový kód a vzorce pro výpočty proměnných *temperature*, *humidity*, *t* a *h* z datasheetu senzoru SHT41.

3.5.5 Funkce main

```
167 int main ( void )
168 {
169     SYS_Initialize (NULL);
170
171     /*SERCOM2 - registrace callbacku*/
172     SERCOM2_I2C_CallbackRegister (SERCOM2_I2C_Callback, (uintptr_t) NULL);
173
174     /*SW0 - registrace callbacku*/
175     EIC_CallbackRegister (EIC_PIN_15, SW0_Callback, 0);
176
177     /*TC2 - registrace callbacku*/
178     TC2_TimerCallbackRegister (TC2_Callback, 0);
179
180     /*TC2 - spusteni timeru*/
181     TC2_TimerStart ();
182
```

Obrázek 53 Funkce main

Ve funkci main si nyní zaregistrujeme veškeré Callback funkce, které jsme vytvořili dříve a spustíme timery, které v projektu používáme. Funkce main se provede pouze při spuštění modulu – tzn. pouze jedenkrát.

3.5.6 Funkce while

```
185     while ( true )
186     {
187         SYS_Tasks ( );
188
189         /*Counting hours from start*/
190         if(SECONDS == 3600)
191         {
192             HOURS++;
193             SECONDS = 0;
194         }
195
196         /*Measurement & calculating T & H, reading and writing data to CAN bus every 1 s*/
197         if(timer)
198         {
199             timer = false;
200             SHT4x_MEASUREMENT();
201             SHT4x_TEMPERATURE_CALCULATE();
202             SHT4x_HUMIDITY_CALCULATE();
203             CAN_READ();
204             CAN_WRITE();
205             SECONDS++;
206         }
207
208         /*Kdy? je stisknuto SW0, prob?hne v?pis do konzole*/
209         if(SW0)
210         {
211             SW0 = false;
212             LED1_Toggle();
213             printf("ID: %d\r\n",ID);
214             printf("Hours: %d\r\n",HOURS);
215             printf("FW Version: %d\r\n",FW_VERSION);
216             printf("Temperature: %.2f\r\n",temperature);
217             printf("Humidity: %.2f\r\n",humidity);
218         }
219
220     }
221     return (EXIT_FAILURE);
222 }
```

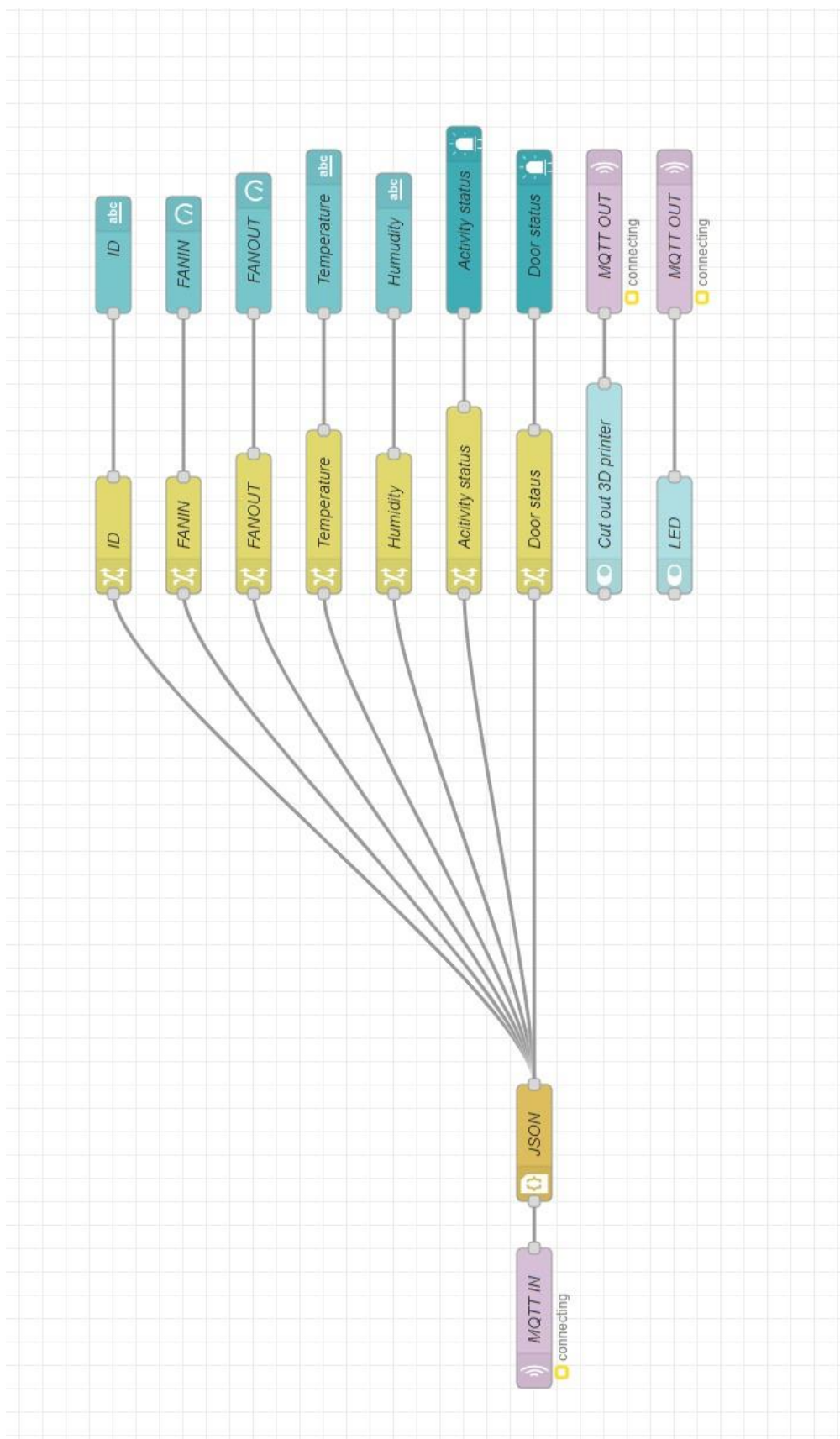
Obrázek 54 Funkce while

Ve funkci while počítáme čas od spuštění mikrokontroleru. Dále každou 1 s probíhá měření a výpočet teploty, zápis a čtení CAN bus sběrnice a také se přičítá 1 s pro výpočet času od spuštění modulu. Funkce while se provádí od spuštění modulu až do jeho vypnutí – odpojení od napájení.

4 Aplikační část projektu

Aplikace je prozatím ve fázi vývoje a také zatím nemáme vizualizační část. Prozatím jsme proto museli udělat dočasnou vizualizaci v programovacím prostředí NodeRED.

4.1 Popis jednotlivých nodů v NodeREDu



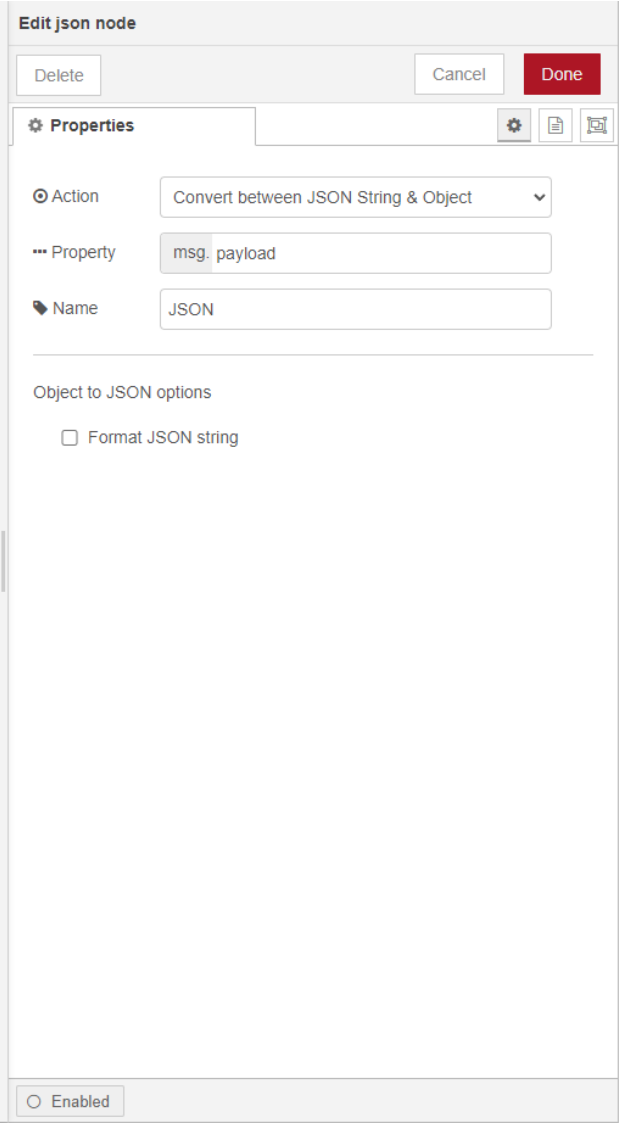
Obrázek 55 Schéma v aplikaci NodeRED

4.1.1 MQTT IN node

Za pomoci tohoto nodu se přijímají informace z MQTT serveru. V nastavení tohoto nodu je nutné nastavit správný subscribe topicu. Pokud by byl subscribe špatně nastaven, data by se nezobrazila.

4.1.2 JSON node

JSON node data převede do formátu JSON. Data se převádí proto, aby další nody dokázaly data rozpoznat a zobrazit v dashboardu.



The image shows a configuration window titled "Edit json node". At the top, there are three buttons: "Delete", "Cancel", and "Done". Below the buttons is a "Properties" section with a gear icon and three sub-panels: "Action", "Property", and "Name". The "Action" panel has a dropdown menu set to "Convert between JSON String & Object". The "Property" panel has a text input field containing "msg. payload". The "Name" panel has a text input field containing "JSON". Below these panels is a section titled "Object to JSON options" with a checkbox labeled "Format JSON string" which is currently unchecked. At the bottom of the window, there is a status bar with a radio button labeled "Enabled" which is selected.

Obrázek 56 Nastavení JSON nodu

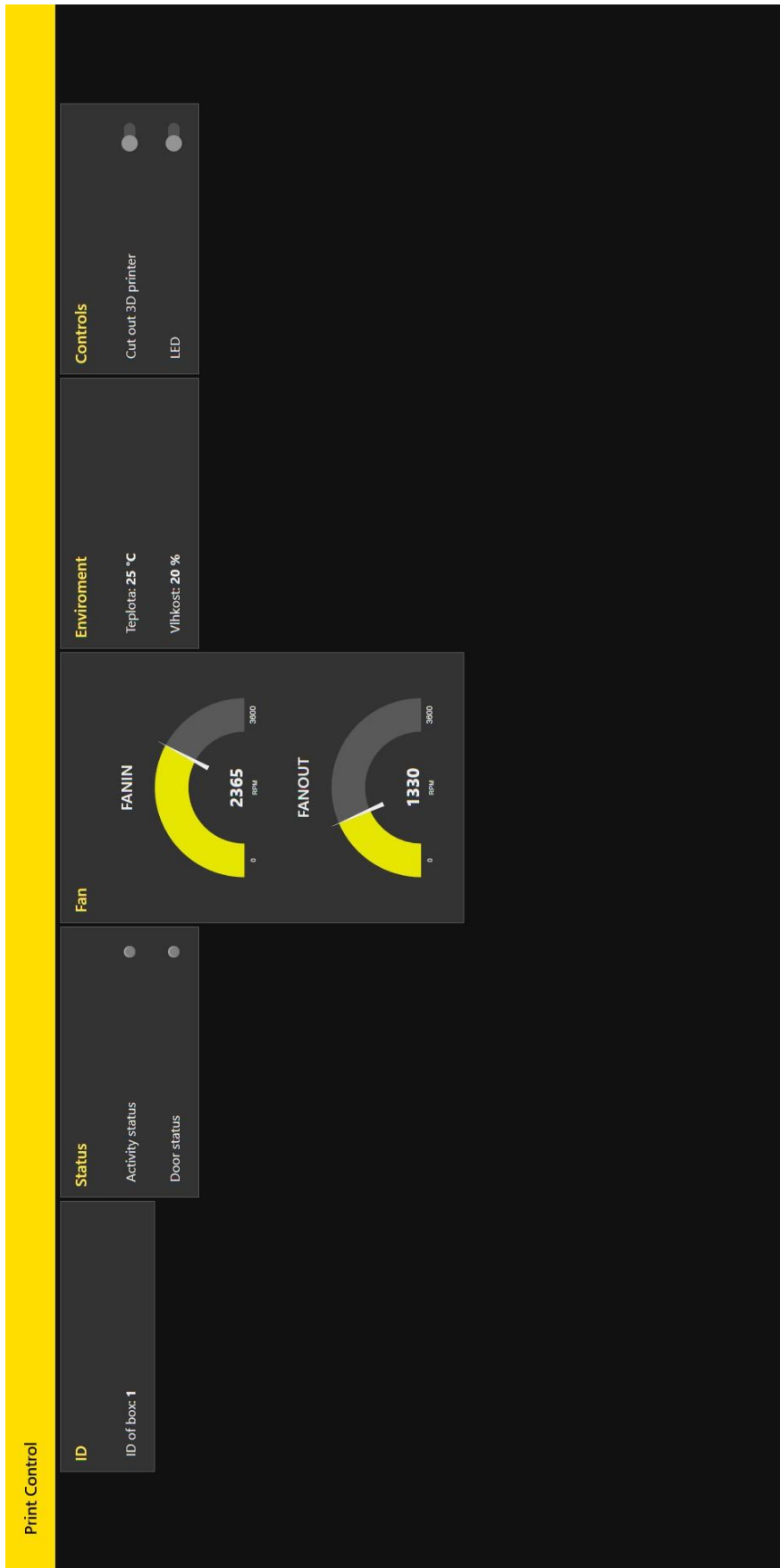
4.1.3 Change node

Change node změní tvar zprávy tak, aby Dashboard UI nody mohly z naměřených hodnot mohly přečíst neměřené hodnoty a zapsat je do Dashboard UI nodů.

The screenshot shows the 'Edit change node' configuration window. At the top, there are three buttons: 'Delete', 'Cancel', and 'Done'. Below the buttons is a 'Properties' section with a gear icon, a document icon, and a refresh icon. The 'Name' field contains 'ID'. The 'Rules' section shows a 'Move' action. The source is 'msg.payload.ID' and the target is 'msg.payload'. There is an '+ add' button at the bottom left and an 'Enabled' checkbox at the bottom right.

Obrázek 57 Nastavení change nodu

4.1.4 Dashboard UI



Obrázek 58 Rozložení Dashboard UI

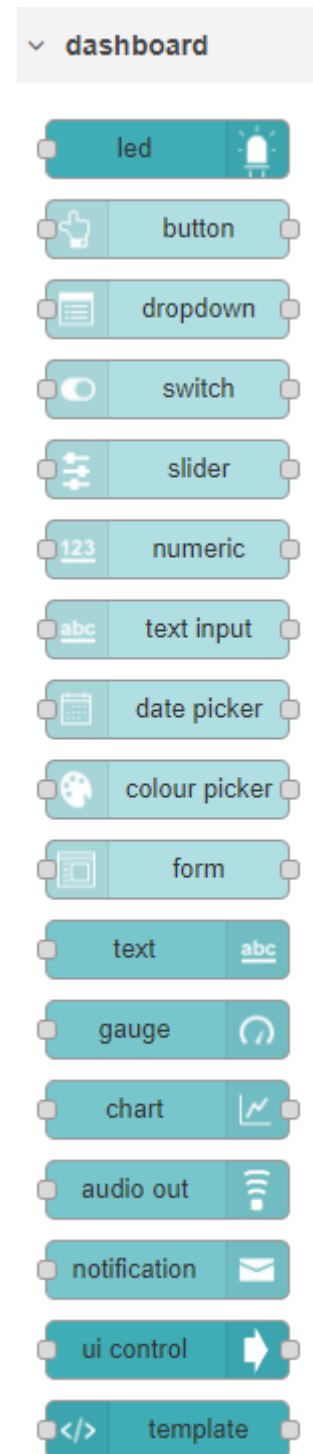
Dashboard UI je soubor několika nodů, který obsahuje grafické nody pro vizualizaci přijatých dat. V našem případě používáme node led, gauge, text input a switch.

LED node používáme pro zobrazení statusu aktivity a otevření dveří.

Gauge node je použit na zobrazení rychlosti ventilátorů, které jsou umístěni uvnitř boxu.

Text input používáme pro zobrazení ID boxu, teploty a vlhkosti uvnitř boxu.

Switch node slouží k ovládání LED diody a k odpojení 3D tiskárny od napájení.



Obrázek 59 Nabídka nodů z Dashboard UI

5 Webové stránky

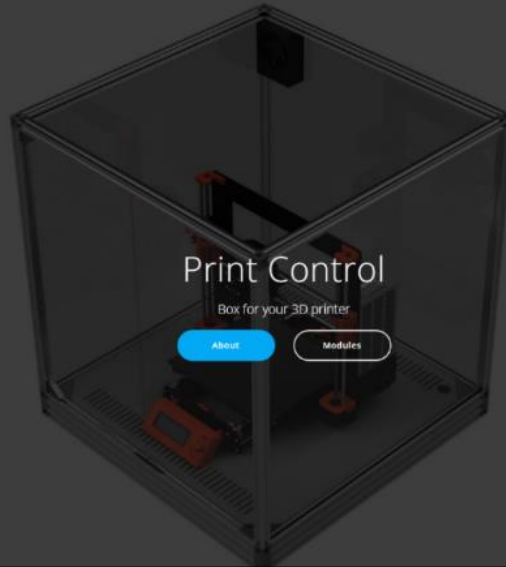
Na základě našich poznatků a znalostí z předmětu *Vývoj webových stránek* jsme vytvořili webové stránky k našemu projektu.

Naše webové stránky obsahují několik sekcí. Po kliknutí na tlačítka na úvodní stránce je možné se dostat na sekci *About* a sekci *Modules*. V sekci *About* je základní popis projektu a jeho možného využití. V sekci *Modules* jsou popsány dva moduly a box pro 3D tiskárnu. A na konci je kontaktní formulář, pro kontakt, kdybyste se nás potřebovali na něco zeptat.

Pro zobrazení webové stránky naskenujte QR kód.



Obrázek 60 QR kód pro webovou stránku



Print Control

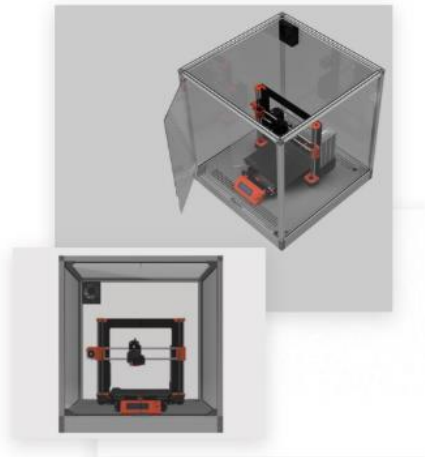
Box for your 3D printer

- About
- Modules

WHAT DO WE DO.

Box for 3D printer

We focus on solving problems related to 3D printers, improving their control and printing conditions. Our solution is for anyone who wants better printing conditions and reduces unwanted printer problems. The PrintControl system is really useful for small and medium-sized companies operating in with the 3D printers business, providing services such as custom printing, which need some intelligent control system for efficient operation and management of printers or entire printing farm. Furthermore, the device is also very useful for high schools and universities and has uses in cases of a smaller amount of printers as well, for instance in offices or households.



- About
- Modules

OUR WORK

Main modules

The most important modules of Print Control



Box for 3D printer
Module for measuring temperature and humidity. We can inform you about temperature and humidity through our web app.



Temperature and Humidity sensor
Module for measuring temperature and humidity. We can inform you about temperature and humidity through our web app.



Automatic fan controller
Module for automatic regulating fan speed. We can inform you about speed of fan through our web app.

View Hardware

- Modules

Obrázek 61 Webová stránka

6 Budoucnost projektu

V nejbližší budoucnosti bychom chtěli dokončit zbylé moduly potřebné k plné funkčnosti boxu: modul topného tělesa a LED diod. Dále bychom chtěli pracovat na odstranění konstrukčních chyb v celé konstrukci boxu. Již vzniklé moduly chceme dále vylepšovat a optimalizovat.

V další revizi plánujeme přidat více modulů, aby se systém PrintControl mohl stát ještě více autonomním. Mezi ty patří například modul se senzorem vibrací, modul pro detekci otevřených dveří nebo třeba LCD display umístěný v dolním rohu boxu. Užitečná by určitě byla i webkamera pro sledování a kontrolu 3D tiskáren na dálku z pohodlí aplikace. Dále bychom chtěli více optimalizovat firmware celého zařízení, abychom zjednodušili práci se systémem PrintControl na co nejjednodušší.

Rozhodně bychom chtěli vylepšit i aplikaci, která je zatím pouze v prototypové fázi. Plánujeme vývoj naší vlastní webové a mobilní aplikace, která bude pro uživatele přehledná a snadná na ovládání. Webovou aplikaci bychom chtěli mít propojenou společně s naší webovou stránkou, aby hledání informací o nových projektech či zakoupení nových periférií bylo pro uživatele co nejdostupnější.

7 Závěr

Cílem našeho projektu je vytvořit užitečný a uživatelsky přívětivý box pro 3D tiskárny, který svými funkcemi pomůže mnoha uživatelům 3D tiskáren. Momentálně se stále ještě nacházíme ve fázi vývoje celého zařízení. Přestože jsme si již řadu problémů s projektem zažili, jsme připraveni řešit i další.

Práce na projektu PrintControl nás baví a chceme celý projekt dotáhnout do finální funkční podoby. I nadále pak chceme posouvat projekt dál a nezastavovat vývoj, protože si myslíme, že PrintControl má budoucnost.

8 Literatura

[1] Microchip technology [online]

Dostupné z: <https://www.microchip.com/en-us/development-tool/EV75S95A>

[2] Autodesk Fusion 360 [online]

Dostupné z: <https://www.autodesk.cz/products/fusion-360/overview>

[3] Microchip technology – MPLAB X IDE [online]

Dostupné z: <https://www.microchip.com/en-us/tools-resources/develop/mplab-x-ide>

[4] Záhlava, V.: Návrh a konstrukce desek plošných spojů, BEN, Praha 2011

9 Seznam obrázků

Obrázek 1 PrintControl – box pro 3D tiskárnu	7
Obrázek 2 Blokové schéma systému PrintControl.....	8
Obrázek 3 Nosná konstrukce boxu, 3D vizualizace.....	9
Obrázek 4 Základna boxu, 3D vizualizace.....	10
Obrázek 5 Umístění modulů v boxu, 3D vizualizace.....	11
Obrázek 6 Celá sestava PrintControl, 3D vizualizace.....	11
Obrázek 7 PrintControl pohled zepředu, 3D vizualizace	12
Obrázek 8 Příklad zapojení sběrnice CAN bus	13
Obrázek 9 Symbol dvoucestné TVS diody	14
Obrázek 10 Symbol tlumivky souhlasného napětí	14
Obrázek 11 Schéma zapojení sběrnice CAN bus a napájení desky	14
Obrázek 12 Schéma zapojení stabilizátoru	15
Obrázek 13 Schéma zapojení MCU a hodinového krystalu	16
Obrázek 14 Schéma zapojení CAN převodníku	17
Obrázek 15 Schéma zapojení sériové konzole	18
Obrázek 16 Schéma zapojení spínače a LED diody	18
Obrázek 17 Schéma zapojení senzoru SHT41	19
Obrázek 18 PCB Environment sensor – pohled shora	20
Obrázek 19 PCB Environment sensor – pohled zdola	20
Obrázek 20 PCB Environment sensor – 3D vizualizace.....	20
Obrázek 21 Environment sensor – uzavřená krabička	21
Obrázek 22 Environment sensor – otevřená krabička.....	21
Obrázek 23 Schéma zapojení pro PWM regulaci ventilátoru	22
Obrázek 24 PCB Fan controller – pohled shora.....	23
Obrázek 25 PCB Fan controller – pohled zdola.....	23
Obrázek 26 PCB Fan controller – 3D vizualizace	23
Obrázek 27 Fan controller – uzavřená krabička.....	24
Obrázek 28 Fan controller – otevřená krabička	24
Obrázek 29 Heater module – pohled shora	25
Obrázek 30 Heater module – pohled zdola	25
Obrázek 31 Vývojová deska SAM-IoT-WG.....	26
Obrázek 32 PCB Gateway – pohled shora	27
Obrázek 33 PCB Gateway – pohled zdola	27
Obrázek 34 PCB Gateway – 3D vizualizace.....	27
Obrázek 35 Gateway – uzavřená krabička	28
Obrázek 36 Gateway – otevřená krabička.....	28
Obrázek 37 Prostředí pro doinstalování pluginů.....	29
Obrázek 38 Postup, jak otevřít Content Manager	30
Obrázek 39 Tlačítko pro založení nového projektu	31
Obrázek 40 Návod na založení projektu – 1	31
Obrázek 41 Návod na založení projektu – 2	32

Obrázek 42 Návod na založení projektu – 3	33
Obrázek 43 Návod na založení projektu – 4	33
Obrázek 44 Harmony 3 – Project Graph	34
Obrázek 45 Seznam periférií MCU SAMC21E18A	35
Obrázek 46 Harmony 3 – nastavení pinů	36
Obrázek 47 Tabulka „Debug Interfaces“	37
Obrázek 48 MPLAB Snap In – Circuit Debugger	38
Obrázek 49 Nastavení globálních proměnných	39
Obrázek 50 Funkce pro čtení a zápis na CAN sběrnici.....	40
Obrázek 51 Callback funkce	40
Obrázek 52 Čtení a zápis na I2C, výpočet teploty a vlhkosti	41
Obrázek 53 Funkce main.....	42
Obrázek 54 Funkce while.....	43
Obrázek 55 Schéma v aplikaci NodeRED	45
Obrázek 56 Nastavení JSON nodu.....	46
Obrázek 57 Nastavení change nodu	47
Obrázek 58 Rozložení Dashboard UI.....	48
Obrázek 59 Nabídka nodů z Dashboard UI	49
Obrázek 60 QR kód pro webovou stránku	50
Obrázek 61 Webová stránka	51