

Gymnázium, Praha 6, Arabská 14

Programování



ROČNÍKOÝ PROJEKT

Kontejnerizační systém pro školní server

Vypracovali:

Havránek Kryštof, Vávra Vladimír, Litoš Josef 3.E

Vedoucí práce:

Ing. Daniel Kahoun

Duben 2021

Prohlašuji, že jsme jedinými autory tohoto projektu, všechny citace jsou řádně označené a všechna použitá literatura a další zdroje jsou v práci uvedené. Tímto dle zákona 121/2000 Sb. (tzv. Autorský zákon) ve znění pozdějších předpisů uděluji bezúplatně škole Gymnázium, Praha 6, Arabská 14 oprávnění k výkonu práva na rozmnožování díla (§ 13) a práva na sdělování díla veřejnosti (§ 18) na dobu časově neomezenou a bez omezení územního rozsahu.

V dne

Podpisy autorů

Poděkování

Název práce: Kontejnerizační systém pro školní server

Autoři: Havránek Kryštof, Vávra Vladimír, Litoš Josef 3.E

Abstrakt: Cílem práce je vytvořit kontejnerizační systém pro operační systém GNU/Linux. Ten by měl v jednoduchém a přehledném uživatelském prostředí umožnit žákům vytvořit si vlastní linuxový server. Na které si poté mohou hostovat svoje aplikace, webové stránky, či videohry. Systém by se měl postarat o to, aby zdroje serveru byly spravedlivě rozdělené mezi jednotlivé uživatele.

Klíčová slova: linux, lxd, lxc, kontejnery, nodejs, REST API, react js

Title: Containerization system for school server

Authors: Havránek Kryštof, Vávra Vladimír, Litoš Josef 3.E

Abstract: Goal of the work is to create containerization system for GNU/Linux operating system. With its help students should be able to create their own private server in a easy to use webUI. On such server they will be able to host their applications, websites or video games. System should aslo guarantee fair redistribution of server resources.

Key words: linux, lxd, lxc, containers, nodejs, REST API, react js

Titlel: Containerisierungssystem für Schulserver

Autoren: Havránek Kryštof, Vávra Vladimír, Litoš Josef 3.E

Abstrakt: Ich spreche Deutsch nicht sehr gut.

Obsah

Úvod	v
1 Základní informace	1
1.1 Hardware	1
1.2 Použité knihovny	2
1.3 Použité programy	3
1.4 Požité technologie	3
1.5 Struktura souborů práce	4
2 Vybrané části programu	5
2.1 ESP-NOW	5
2.2 Telnet	6
2.3 Čas	6
2.4 Nextion display	7
2.5 Paměť	8
2.6 Počasi	8
3 Budoucí části programu	9
3.1 Senzor u pneumatik	9
3.2 Vytápění podlahy a stav baterie	9
3.2.1 Elektřina	10
3.2.2 Vytápění podlahy	10
Závěr	vi
Seznam použité literatury	vii
Seznam obrázků	viii

Úvod

1. Základní informace

Tato kapitola spěšně projde hardware, na kterém je program vybudován, knihovny, které byly použity. Dále projde technologie, či protokoly, a na závěr čtenáře seznámí se strukturou souborů.

1.1 Hardware

Projekt je postaven na modulární platformě M5Stack, který je postaven na základě platformy ESP32 (velice podobné Arduino, ale má zabudovaný wifi modul). Tyto jednotky obsahuje projekt 5 – voda, GPS, vytápění podlahy/stav baterie, kola a na závěr samotná centrála. Další hardwarové prvky jsou rozepsány níže ve seznamu, a jsou přiřazeny k jednotlivým jednotkám. Nákrety s obvody nejsou součástí práce, neb připojení je zřetelné z kódu (jsou zapsané GPIO piny).

- Centrální jednotka
 - NEXTION display intelligent series model NX8048P070-011C, kód kompatibilní i s Enchanted sérií (stačí vypnout videa pro počasí)
 - LAN M5Stack modul
 - Switch od firmy Mikrotik
- Jednotka pro vodu
 - 2 * tepelný senzor DS18B20 – jeden umístění zvenčí, druhý ponořený do vody. (Jsou zapojeny na busu paralelně)
 - průtokoměr – umístění u odtoku z nádrže s vodou
 - 2 * relé – jedno ovládá tepelný okruh v nádrži, druhé ventil pro napouštění nádrže
 - analogový podvodní senzor tlaku – umístěn v přípojce na vodu, zjišťuje zda je umístěna přípojka
 - 2 * vodotěsný spínač – umístěn dole a nahoře v nádrži, určuje zda voda klesla pod kritickou úroveň a zda je nádrž plná

- Jednotka pro GPS – M5Stack GPS module
- Jednotka pro vytápění podlahy, stav baterie (Není realizována)
 - 8 * tepelný senzor DS18B20 – v každém okruhu dva senzory
 - 4 solid state relé – ovládání okruhů
 - eventuálně Bezdrátový stejnosměrný wattmetr VAC1030A, není ani zmíněn v kódu
 - hall sensory

1.2 Použité knihovny

- M5Stack – ovládání displaye na M5Stack – Dostupné z URL: <https://github.com/m5stack/M5Stack>
- WiFi – WiFi, nutné pro protokol ESP-NOW – Dostupné z URL: https://dl.espressif.com/dl/package_esp32_index.json
- esp_now – ESP-NOW – Dostupné z URL: https://dl.espressif.com/dl/package_esp32_index.json
- EEPROM – zprovoznění EEPROM – Dostupné z URL: https://dl.espressif.com/dl/package_esp32_index.json
- TinyGPS++ – zpracování dat z GPS – Dostupné z URL: <http://arduinoiana.org/libraries/tinygpsplus/>
- Time – trackování času – součást standardní knihovny
- Timezone – práci s časovými pásmy, použita jen pro středoevropské – Dostupné z URL: <https://github.com/JChristensen/Timezone>
- NTPClient – zajištění protokolu NTP – Dostupné z URL: <https://github.com/taranais/NTPClient/>
- Ethernet2 – LAN modul k M5Stack – Dostupné z URL: <https://github.com/m5stack/M5Stack/blob/master/examples/Modules/W5500/Ethernet2.zip>

- Ethernet2Udp – zajištění UDP protokolu – součást Ethernet2
- WebServer – web server – součást Ethernet2
- Nextion – práci s displayem Nextion – Dostupné z URL: https://github.com/itead/ITEADLIB_Arduino_Nextion
- espsoftwareserial – potřebný k knihovně Nextion – Dostupné z URL: <https://github.com/plerup/espsoftwareserial>
- OneWire – knihovna potřebná k DallasTemperature
- DallasTemperature – tepelné čidla – Dostupné z URL: <https://github.com/milesburton/Arduino-Temperature-Control-Library>
- PCA9554 a ClosedCube_TCA9548A – pro IO extender na jednotce vytápění
- ArduinoJson – parsování jsonu – Dostupné z URL: <https://github.com/bblanchon/ArduinoJson>

1.3 Použité programy

Celý kód byl napsán v textovém editoru Vim a v Arduino IDE. Gui interface na Nextion display byl vytvořen v Nextion editoru.

Vývoj nejvíce komplikovalo právě Arduino IDE. Prostředí nemá automatické doplnění, ani žádné možnosti rychle přecházet mezi metodami. Samozřejmě nejsou k dispozici ani debugovací nástroje, takže přehledné výpisy na sériovou linku jsou nutností. Tyto debug zprávy jsou v kódu ponechané a většina z nich má následující formát: ZKRATKA NÁZVU TŘÍDY | název metody | samotná správa.

1.4 Požité technologie

Bezdrátovou komunikaci mezi jednotlivými jednotkami zprostředkovává proprietární protokol ESP-NOW, který je vlastní čipu ESP32. K získání aktuálního počasí centrála používá standardní http protokol, jedná se však o pouhý jeden GET request. Komunikace se switchem je zprostředkována za pomoci telnetu. Čas je aktualizován

prostřednictvím NTP, v případě potřeby je však možnost použít data z GPS. Ke zbylé komunikaci slouží sériová linka a GPIO piny.

Většina práce je napsaná v programovacím jazyce C++, přesněji jeho verzi pro Arduino. Některý kód je napsán v programovacím jazyce pro Nextion, ten je uložen v .HMI souboru. Jedná však o několik málo primitivních funkcí, jako třeba přičtení dne.

1.5 Struktura souborů práce

Karavan

- └ CentralUnit - složka obsahující kód centrální jednotky
 - └ CentralUnit.ino - hlavní třída
 - └ Connection.h - zajišťuje ovládání switchu MikroTik pomocí telnetu
 - └ PowerAndHeating.h - reprezentuje jednotku zajišťující vytápění podlahy a kontrolu stavu baterie
 - └ Security.h - reprezentuje jednotku zajišťující GPS
 - └ UnitAbstract.h - abstraktní třída, slouží jako šablona pro všechny třídy, které reprezentují fyzickou jednotku
 - └ Water.h - reprezentuje jednotku u vody
 - └ Weather - spravuje informace o počasí
 - └ Wheels - reprezentuje možnou jednotku u kol, nebyla vůbec provedena instalace do čidel do kol karavanu
- └ ESPNowSlaveTemplate - složka obsahující šablonu pro jednotku
 - └ ESPNowSlaveTemplate.ino - šablona s implementací ESP-NOW
- └ HeatingUnit - složka obsahující kód jednotky pro vytápění
 - └ HeatingUnit.ino - kód pro jednotku na vytápění
- └ WaterUnit - složka obsahující kód jednotky pro vodu
 - └ WaterUnit.ino - kód pro jednotku pro vodu
- └ WheelsUnit - složka obsahující kód jednotky pro kola
 - └ WheelsUnit.ino - kód pro jednotku pro kola
- └ NextionFiles - soubory pro display Nextion
 - └ Backgrounds - složka s pozadími
 - └ Fonts - složka s fonty
 - └ Icons - složka s ikonkami pro Nextion
 - └ UI - ikonky pro UI, vlastní tvorba
 - └ Weather icons - ikonky pro počasí, jak obrázky, tak videa¹
 - └ Caravan.HMI - soubor s rozložením displaye
 - └ Caraven.tft - vyexportované rozložení displaye

¹Obrázky pochází z <https://dribbble.com/shots/1669480-Weather-Icons/attachments/263247>, ikona tornáda z <https://www.flaticon.com/free-icons/tornado>, obrázky, které na těchto stránkách, nenajdete vznikli editací. Zakoupeny z <https://videohive.net/item/animated-weather-icons/19199689>

2. Vybrané části programu

Vzhledem k délce programu a jeho různorodosti, není možné s limitem délky této dokumentace projít všechny funkce programu. Proto se tato dokumentace zaměří jen na několik vybraných částí.

2.1 ESP-NOW

Jak už bylo zmíněno, bezdrátový přenos dat mezi jednotkami je zprostředkována pomocí protokolu ESP-NOW. Tento protokol je relativně jednoduché používat, stačí přidat jednotku mezi peery a poté lze jednoduše posílat na danou mac adresu. Data jsou posílána ve formě struktury, která je stejná jak na centrále, tak na jednotce. Po přijmutí dat se jednoduše pomocí memcpy překopírují do instance struktury.

Samozřejmě je nutné při spuštění vytvořit síť s centrálou ve středu a k ní připojenými jednotlivými jednotkami. Centrála musí vědět od jaké jednotky data přichází a musí být schopna poslat data, či konfiguraci (Zatím není použito, bude nutné u podlahy.).

Původně systém fungoval následovně. Všechny jednotky měli viditelné SSID centrála je postupně všechny kontaktovala. Jakmile našla jednotku bylo jí zasláno číslo 92 (Není v tom žádný význam) a jednotka odpověděla svým kódem – 101 pro GPS, 102 pro vodu. Tento systém měl však několik nevýhod. Zaprvé nahrál li se nový kód do centrály, tak se museli zresetovat všechny jednotky, protože centrála se v nich se smaže až po 5 minutách nečinnosti. A zadruhé SSID jednotek byly pořád viditelné a systém neumožňoval je schovat.

Proto byl nakonec tento systém obrácen. Místo toho aby centrála vyhledávala jednotky nyní jednotky vyhledávají centrálu. Té poté pošlou svůj kód a centrála opět odpoví číslem 92. Zároveň si jednotky uloží mac centrály do své EEPROM (stejně tak činní i centrála, kdy ukládá mac jednotek do své EEPROM). Při dalším spuštění nemusí tak mít centrála viditelné SSID a síť se stejně postaví. Viditelnost SSID se ovládá pomocí tlačítka na M5Stacku, s tím že na rozdíl od předchozí implementace pouze centrála má své SSID viditelné.

Centrála pravidelně posílá všem jednotkám ping, aby si jednotky centrálu nevzali

jako neaktivní a nesmazali jí. Stejně tak centrála si pamatuje poslední čas, kdy přišla zpráva od jednotky a v případě nečinnosti ji smaže.

Jediný problém při vývoji, který stojí za zmínku, je chybová hláška, již vyvolávala knihovni funkce `esp_now_add_peer()`. Ta z neznámých důvodů vyhazovala hlášku `invalid peer`, ovšem někdy se stejným vstupním parametrem funkce proběhla v pořádku. Tento problém vyřešilo až přesunutí instance struktury `toAdd` na globální proměnou. Naštěstí tento problém byl vyřešen již na fóru¹.

2.2 Telnet

Jednou z funkcí projektu je možnost přepínat zdroj internetu – mezi WiFi access pointem a LTE modemem. Tuto funkci zajišťuje switch, ke kterému je centrála připojena přes ethernetový kabel.

Původně bylo v plánu napsat celou implementaci protokolu telnet. Bohužel logika z neznámých důvodů nefungovala. Pomocí programu `wire shark` byly proto odposlechnuty jaké příkazy si mezi sebou client a server posílají během párování. Ty poté byly napevno dané do programu. Program tak jen čeká na určitou posloupnost a jí odpoví určitou posloupností.

Samotné posílání příkazů již nečiní žádné problémy a celý systém funguje.

2.3 Čas

Uživatel má celkem 3 možnosti jak nastavit čas.

Standardně je čas řešen pomocí protokolu NTP pomocí knihovny `NTPClient`. Tento způsob samozřejmě vyžaduje internetové připojení, ale to je téměř vždy dostupné. Navíc po jednom dotazu si čas centrála udržuje.

Případě nedostupnosti sítě lze použít GPS, samozřejmě pouze za předpokladu, že je připojená GPS jednotka. Čas z GPS není tak tak přesný jako u NTP, protože po přečtení se musí poslat přes EPS-NOW (Třeba na několikátý pokus.), na centrále se musí zpracovat callback (`ESP32` má jen jedno vlákno, takže než se zpracuje uběhne nějaký čas.). Každopádně se jedná o možnou variantu.

¹EYCK, L. (2020). *ESPNow: Peer Interface is invalid*. Random Nerd Tutorials Lab. Dostupné z <https://rntlab.com/question/espnow-peer-interface-is-invalid/>. [cit 2020-28-4]

V poslední řadě lze čas nastavit manuálně. Uživatel na stránce s nastavením času zmáčkne tlačítko nastavit, dále nastaví čas a poté zmáčkne tlačítko znovu. V callbacku, který tato akce na centrále vyvolá se zkontroluje stav tlačítek NTP a GPS a jsou li obě vypnuté čas se nastaví ze zadaného. V jiném případě bude uživatelem zadaný čas ignorován. Na Nextionu jsou samozřejmě přidány různé omezení, aby uživatel nemohl zadat třeba 13. měsíc. Lze však i tak zadat neplatné datum a to tím, že člověk změní měsíc po nastavení dne. Bohužel programovací jazyk Nextionu je velmi primitivní a nepodporuje ani základní věci jako je tvorba proměnných (Všechny proměnné jsou součástí objektu na display). Už jen kód na přidání dne má 57 řádků a není možné třeba při změně měsíce udělat aby vyskočila uživateli chybová hláška. Nebo minimálně by nebyl takový kód zrovna elegantní.

Střídání času je funguje pouze na střeoevropské pásmo, jinak si uživatel musí měnit střídání času a časové zóny sám. Logika všech pravidel na úpravu času je relativně složitá a jsou na ní potřeba stovky souborů. Ty sice lze mít uložené na SD kartě (Kterou M5Stack podporuje.), ale nejedná se o příliš efektivní řešení, protože parsování souborů je relativně složitý proces na něčem nevykoném jako je ESP32. Uživatel si proto na druhé stránce Nextionu může zvolit manuální nastavení posunu času, to je dě pomocí hodinových inkrementů.

2.4 Nextion display

Ovládacím prvkem celého systému je display od Nextionu. Na něm jsou aktuálně tři stránky – hlavní, nastavení času a nastavení vytápění. Stránka na vytápění však aktuálně neplní žádnou funkci.

Posílání dat display je relativně jednoduché. Pro čtení dat a používání callbacků je nutná knihovna Nextion. Pomocí ní vytvoříme instance objektů, které odpovídají těm na obrazovce. U callbacků je ještě nutné přidat instanci do pole `nex_listen_list` a manuálně přiřadit jaká metoda bude zavolána.

Nutno upozornit, že stejně všechny akce na centrále obstarává jen jedno vlákno. Takže ne vždy uživatelské akce na obrazovce vyvolá okamžitou reakci v programu. Někdy se může jedna až o sekundy, než bude callback zpracován. Z tohoto důvodu se malé akce jako přičtení čísla řeší přímo na Nextionu.

Syntaxe jazyku na Nextionu je velmi podobná ostatním mainstreamovým jazykům. Ale, jak už bylo zmíněno, nedovoluje vytvářet vlastní proměnné. Z tohoto důvodu jsou například na stránce s nastavení času tři čísla, které mají barvu textu stejnou jako barvu pozadí. Jedná se o krkolomné řešení, ale jiné neexistuje. Také jazyk nepodporuje základní věci jako složené podmínky, či možnost provádět operace v podmínkách.

Jediným problémem je, že přes sériovou linku nelze poslat charaktery s diakritikou. Z tohoto důvodu není v aktuální době sladěný jazyk, kde ovládací prvky na Nextionu jsou v češtině, zatímco data přicházející z centrály v angličtině.

2.5 Paměť

Některé informace si systém přenáší do svého dalšího spuštění. K tomu slouží integrovaná EEPROM v ESP-32. Jedná se o velmi primitivní typ paměti a konstrukce jako soubory nejsou možné. Všechna data se dokonce musí zapisovat ručně po bytech. To činní nutným mít paměť rozkouskovanou a vědět jaké bloky patří jaké proměnné. Zároveň má EEPROM relativně malý počet možných zapsání. Z tohoto důvodu se při zápisu čte paměť a data se porovnávají, aby nedocházelo k zbytečným zápisům.

2.6 Počasí

Počasí je získáno pomocí API z open weather map. Odpovědí na http request je json, který se zparsuje pomocí knihovny ArduinoJson. Zobrazení počasí může být uděláno jak formou videa, tak formou obrázku. Video je však možné jen pro Enchanted Series display. Mezi styly je možné vybírat si pouze v kódu, a změna vyžaduje flashnutí centrály novým kódem. Jaké ikona se zobrazí se řídí ID daného počasí, kompletní seznam je dostupný na stránkách open weather map².

² *Weather Conditions*. Open Weather Map. Dostupné z <https://openweathermap.org/weather-conditions>. [cit 2020-28-4]

3. Budoucí části programu

Bohužel nebylo možné kompletně práci dokončit. Nebyl realizován systém pro kola, z důvodu chybějící instalace, a nutnosti ji prakticky otestovat. Také nebyla realizována část podlahy a stavu baterie. Technicky jsou však tyto části již vymyšlené a na GitHubu, kde je práce publikována, budou časem dostupné. Tato kapitola projde alespoň jejich algoritmickou stránku, i když samotný kód ještě nebyl naspán.

3.1 Senzoru u pneumatik

Je plánované do kol karavanu vložit senzory tlaku a teploty. Podobný systém je již instalovaný v multivanu, které karavan tahá. Informace ze senzorů se budou stejně jako další zobrazovat na Nextionu, ale zároveň budou posílány i do jednotky v autě, aby byly dostupné pro řidiče i za jízdy. Bude se jednat o senzory vybavené technologií bluetooth, kterou M5Stick podporuje.

Prakticky je implementace relativně jednoduchá, s tím, že valnou většinu práce řeší knihovna. Jedinou problémovou stránku je identifikace jednotlivých kol. Zatím je kód napsaný s napevno danými ID senzorů a kola musí být dána vždy ve stejném pořadí. Na vysvětlenou mám zjištěný ID předního levého kola a to kolo musí být vždy levé přední, nelze ho dát jinak a pokud by bylo potřeba kolo vyměnit za nové, tak by se musel předělávat kód. Toto je nadmíru nepraktické. Teoreticky by nová implementace měla pracovat se silou signálu. M5Stick by se nenacházel ve středu auta, ale byl by výrazně blíž jedné pneumatice (Jeho pozice by byla napevno daná.). Identifikace čidel by poté probíhala pomocí síly signálu. Bohužel implementace vyžaduje zkušenosti a ruční doladování, neboť vzdálenost čidel se mění (Jsou v kolech, která se otáčí.), navíc některé části auta mohou bránit signálu. To je potřeba řešit přímo u karavanu, z tohoto důvodu nebyla udělána implementace.

3.2 Vytápění podlahy a stav baterie

Implementace podlahy a stavu baterie nebyla zprovozněna ze stejných důvodů jako pneumatiky – psaní naslepo bez možnosti některé části testovat je velmi komplikované

a prakticky zbytečné.

3.2.1 Elektřina

Implementace elektřiny je relativně jednoduchá. Původně měla probíhat komunikace s wattmetrem pomocí bezdrátové technologie, bohužel protokol není otevřený veřejnosti a data jsou zašifrovaná. Komunikace aktuálně funguje pomocí sériové linky přes USB. Bohužel se také nepodařilo sehnat chtěné hall senzory, z tohoto důvodu je dokončení této části pozastaveno na neurčito. Nutno podotknout, že jak wattmetrem, tak hall senzory se zabýval hlavně strýc. Sám jsem se v této části projektu příliš neangažoval. Z tohoto důvodu nejsou v kódu jednotky vytápění žádné reference na elektřinu a je zmíněna pouze na Nextionu a centrále (Nemá smysl schovávat).

3.2.2 Vytápění podlahy

U vytápění podlahy se řeší podobný problém jako u kol – je potřeba identifikovat tepelné senzory, které jsou při každém zapnutí indexované v jiném pořadí. Ze senzorů lze získat 8 bytové unikátní číslo, které bylo senzoru přiřazeno při výrobě. Identifikace bude probíhat podobně jako probíhá u vody, kde byl takový systém v mnohem menším měřítku již implementován a odzkoušen. Vytápěcích okruhů bude 4 až 5 a na každý případnou právě dvě čidla DS.

Při identifikaci, kterou člověk spustí z Nextionu, se postupně budou zapínat okruhy a bude se měřit Δt , vždy které senzory přesáhnou 3 (asi 3) stupně, tak se přidají k danému okruhu. Na rozdíl od vody, je však potřeba více testování, protože podlaha se bude zahřívat i tam, kde se zrovna netopí a mohly by tak vzniknout třeba 3 senzory pro jeden okruh atd. Také je nutné řešit, že některé čidla přestanou fungovat a jejich rychlá výměna je nemožná. Program vyžaduje nalezení celé řady konstant, navíc velmi záleží na rozmístění čidel, a proto jsem implementaci řešil pouze algoritmicky.

Na vysvětlenou čidla teploty podlahy jsou nutné, aby se podlaha nezahřála nad limitní teplotu. Nehledě na to, že je praktické mít možnost nastavit si teplotu podlahy a ne pouze teplotu vzduchu.

Závěr

Seznam použité literatury

EYCK, L. (2020). *ESPNow: Peer Interface is invalid*. Random Nerd Tutorials Lab.

Dostupné z <https://rntlab.com/question/espnow-peer-interface-is-invalid/>. [cit 2020-28-4].

Weather Conditions. Open Weather Map. Dostupné z <https://openweathermap.org/weather-conditions>. [cit 2020-28-4].

Seznam obrázků