

Informatika – maturitní práce

Sledování vytíženosti prostoru pomocí ultrazvukových měřičů

Mikoláš Fromm

Vedoucí práce: Emil Miler

Školní rok: 2020/21

Obsah

Úvod.....	3
Zadání.....	3
Návrh.....	3
<i>SR-HCO4/HC05</i>	3
<i>ESP32</i>	4
<i>OrangePi 3</i>	4
Zpracování	5
<i>Funkčnost projektu</i>	6
<i>Úskalí projektu</i>	7
Hodnocení / Evaluace	8

Úvod

Jako svůj maturitní projekt jsem si zvolil po vlastním výběru **navržení obecného počítadla vytíženosti specifických objektů**. Jako reálný fyzický objekt, u kterého chci svým projektem vyzjistit onu vytíženost, je pak pánská toaleta, resp. pánské školní pisoáry v 1. patře před ředitelnu.

Zadání

Pro návrh jsem si zadal několik vstupních podmínek, které bude celý projekt splňovat a podle kterých budu projekt vytvářet. Projekt musí:

- anonymně počítat osoby u jednotlivých pisoárů
- porovnávat vytíženost s ostatními pisoáry
- zobrazovat aktuální data přes WiFi síť na vlastním web-severu
- správně vyhodnocovat vstupní data poskytnutá senzory
- fungovat na napětí max 5V

Návrh

Pro realizaci projektu jsem si vybral 6 ultrazvukových měřičů vzdálenosti SR-HC04 a SR-HC05 pro svou vysokou přesnost a zároveň velký funkční rozsah, vývojový mikrokontrolér ESP32-DEVKIT1 pro správu všech senzorů, vyhodnocování výsledků měření a možnost data sdílet přes WiFi modul po bezdr. síti protokolu 802.11g a také mikropočítač OrangePi3 jakožto webserver a mqtt broker s grafickým zobrazením získaných dat.

SR-HC04/HC05

Ultrazvukové měřice vzdálenosti jsou dostupné a spolehlivé měřiče, které fungují na jednoduchém principu rychlé iniciace ultrazvuku a následném čekání na zpětný odraz zvuku od měřené překážky. Jako výstup vrací zařízení čas, za který zvuk zdolal vzdálenost k překážce a zpět.

ESP32

Mikrokontrolér ESP32 jsem si vybral hlavně díky své možnosti data sdílet po WiFi síti. Původní plán byl využít mikrokontrolér Arduino UNO, který však nedisponuje WiFi modulem, a tedy se nehodí do konceptu mého projektu.

ESP se v mém projektu stará o většinu výpočtů a úloh. Iniciuje měření u jednotlivých senzorů, stejně jako přepočítává vzdálenost, vyhodnocuje obsazenost a posílá ji na webserver pomocí MQTT nodů. Jeho nejdůležitější logická úloha pak spočívá ve správném *bodovém* ohodnocení vytíženosti. Mikrokontrolér musí pomocí senzorů poznat, že u pisoáru stojí stále stejná osoba, případně že se osoby změnily. Stejně tak musí mikrokontrolér ohodnotit každou osobu pouze jednou, aby data byla relevantní.

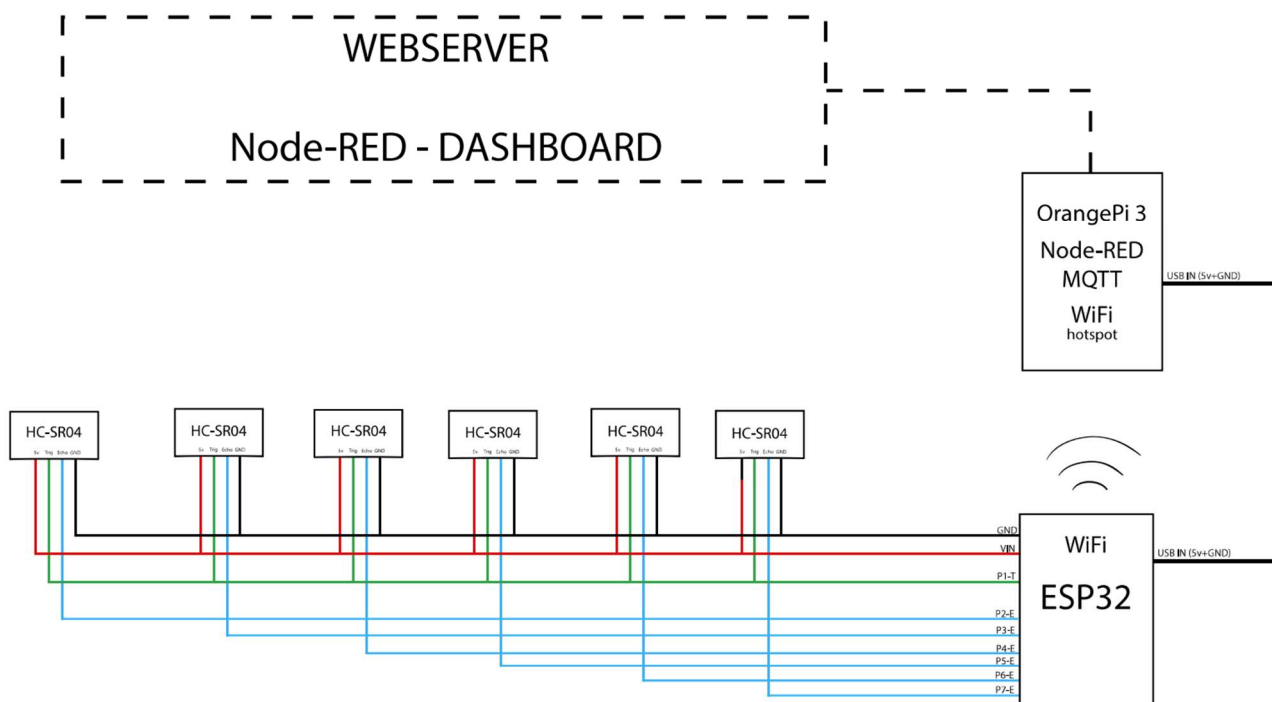
Má však jednu technickou nevýhodu, která se projeví zejména při zapojování senzorů. Tento mikroprocesor pracuje s napětím 3,3v, zatímco senzory jsou nepřesnější v 5v zapojení. To znamená, že do fyzického konceptu je třeba zapojit i „level-shifter“, tedy převodník napět'ových úrovní, který dovoluje komunikaci mezi 5v a 3,3v zařízeními, aniž by došlo k poškození kteréhokoliv zapojeného zařízení. Schéma zapojení a koncept DPS (plošného spoje) bude k vidění níže.

OrangePi 3

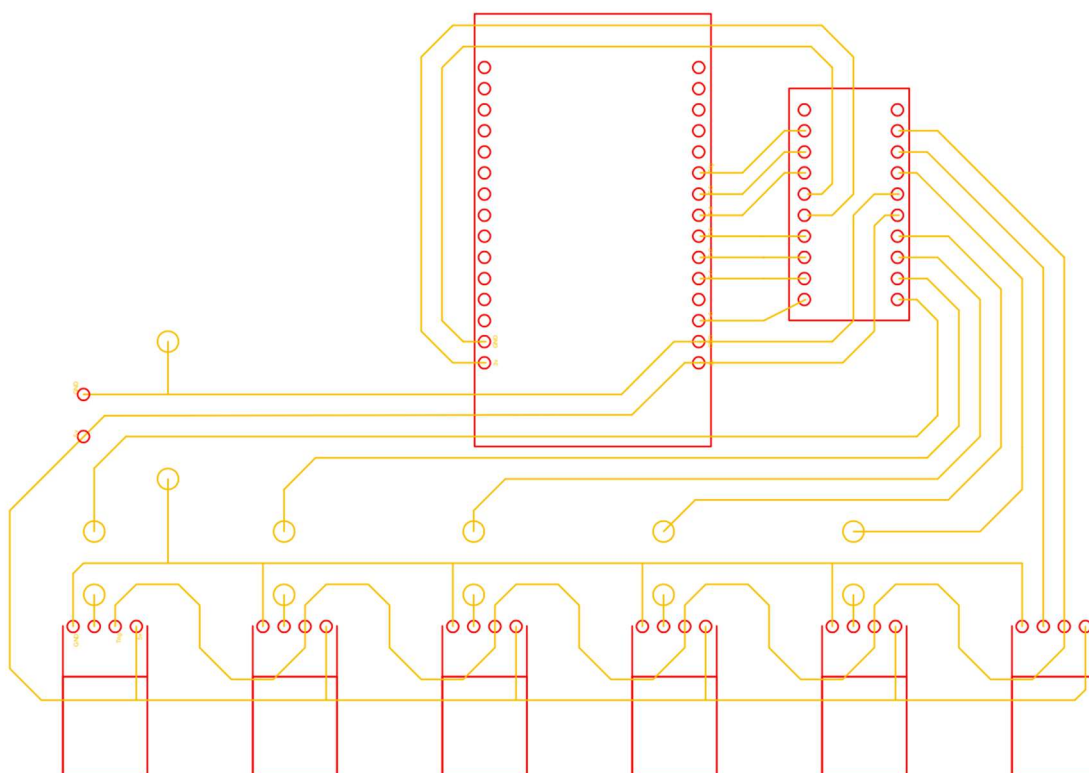
Mikropočítač OrangePi 3 jsem vybral hlavně díky svým dvěma vlastnostem: Má v sobě zabudovaný WiFi modul, který je schopný paralelně vytvářet Wifi-sít' a zároveň z ní čerpat data, a je schopný plynule a paralelně pracovat jako webserver a MQTT broker. Ke všemu to byl pro mě dostupný počítač, který byl pro moje použití vhodný.

Zpracování

V následujícím schématu je přesně popsané zapojení celého projektu:



OrangePi3 je zapojením velmi jednoduchý, avšak schéma zapojení ESP32 kontroléru a jednotlivých senzorů, zejména díky jiným napěťovým úrovním, je složitější a vyžadovalo individuální řešení pomocí DPS a to hlavně pro zapojení „převodníku“, což je obdélníková součástka vpravo na schématu vedle většího ESP.



Zároveň zpracování důležitého a hlavního programu je dostupné na GitHubu:

https://github.com/MeCoolGJK/GJK_WC_counter

Instalace Node-RED, stejně jako MQTT brokeru *Mosquitto* a Node-RED – dashboard na Ubuntu-server běžící na OrangePi3 proběhla bez problémů podle základních dokumentací a návodů, a proto mu nebudu věnovat více prostoru, mimo jiné proto, že OrangePi a podružený webserver není klíčovým prvkem projektu.

Funkčnost projektu

Program je díky celkové koncepci schopný rozeznat příchozí osobu k pisoáru. Jakmile se osoba, případně i jakákoliv jiná překážka, dostane do požadovaného vzdálenostního rozsahu, program začne v rámci sekund vyčkávat, zdali objekt setrvá v rozsahu nebo jde o jinou, neměřenou situaci. Jakmile osoba setrvá před pisoárem požadovaný časový úsek, pisoáru se přičte jeden bod v celkové statistice vytiženosti, což je okamžitě vidět v grafickém zobrazení na webserveru. Spolu s tím ztrácí osoba možnost znovu pisoáru inkasovat další bod za obsazenost, jelikož dokud bude osoba v požadované vzdálenosti sebedelší dobu po ohodnocení, program vyčkává a znovu neboduje. Tzn. že program čeká, než bude před senzorem opět volno, aby mohl začít celý proces znovu. Takto běží program paralelně na všech 6 senzorech / pisoárech. O všechny početní úkony, jako je přepočítání vzdálenosti z uraženého času, přičtení bodu při splnění podmínek a zaznamenání do celkové statistiky, se stará ESP, které následně tyto data ve formě nodů pošle bezdrátově pomocí Wifi na MQTT server, který tyto nody zpracovává. Pro zjednodušení situace je pak MQTT server fyzicky i MQTT client, který data přijímá a WebServer, který data zobrazuje. Je pravděpodobné, že přenos dat je pak rychlejší, ale hlavně je celé schéma díky tomuto sjednocení velmi jednoduché a praktické pro aplikaci „v terénu“.

Úskalí projektu

Přestože jsem se snažil vybrat nejvhodnější prostředky pro můj projekt, nejsou všechny naprosto ideální. Jmenovitě jde o samotný senzor a kvantitu mikrokontrolerů:

Senzor má sice velmi vysokou přesnost a velmi široký vzdálenostní rozsah, ale není soběstačný – neexistuje u něj pasivní mód, a tedy mikrokontrolér ho musí v každém cyklu spouštět, aby zjistil, jestli není objekt v rozsahu. To při použití 6 senzorů a jednoho mikrokontroleru může tvořit problémy, resp. dlouhou pomyslnou frontu, jelikož bude v jistých situacích dlouhá odezva na změnu vzdáleností; *zatímco si osoba stoupne před první pisoár v čase t_0 a program bude v čase t_0 u začátku cyklu druhého pisoáru, bude program chronologicky a postupně iniciovat a vyhodnocovat všechny zbylé senzory / pisoáry, než opět dojde metodicky k prvnímu a vyhodnotí, že před ním někdo stojí.* Na obhajobu projektu je však potřeba zmínit, že se předpokládá, že u pisoáru člověk přetrvává, nežli že by okolo něj jen rychle proběhl, což nepřímo vyřazuje tento problém jako příčinu zkreslení dat. Jako řešení by se nabízelo senzory oddělit od jednoho mikrokontroleru a provozovat například 3 jednotky ESP32 mikrokontroleru celkem – tedy 2 senzory na jeden mikrokontroler. Takové řešení je však příliš drahé vzhledem k účelu.

Tento problém jsem se snažil alespoň částečně vylepšit úpravou funkce programu, což prodlevu částečně zkrátilo, ale za cenu, že v grafickém zobrazení na webserveru zůstává poslední zaznamenaná časová hodnota, dokud do žádaného vzdálenostního rozsahu nevstoupí další osoba.

Hodnocení / Evaluate