

RP report

František Slepíčka

28.2.2025

1 Problém

(*Sorry za VELKÝ blok textu, pod nim je souhrn co v tom říkám.*)

Při testu zapojení jsem udělal chybu, která mi triggerovala WDT. V tu chvíli jsem ještě nevěděl, že je to chyba zapojení, a myslel si, že je chyba v kódu, tudíž jsem odložil používání UWB, až budu mít plně rozmyšlené a vytvořené ovládání ESP. Potom, co jsem toto ovládání měl hotové, jsem se vrhnul na UWB znovu. Tentokrát jsem již měl i druhé nepájivé pole. Na tom jsem však zjistil, že mi WDT netriggeruju a ESP běží tak, jak má (to úplně není pravda, ale dostanu se k tomu). Po dlouhém zkoumání, kde by mohla být chyba, jsem usoudil, že při zapojování první desky jsem nejspíš udělal chybu a že to zapojím znovu. To vskutku pomohlo, takže jsem měl dvě "fungující" UWB a mohl jsem zjistit, že spolu vůbec nekomunikují. Po dalším dlouhém zkoumání, kde je chyba jsem se rozhodl jít úplně od začátku a chybu jsem našel, ÚPLNĚ NA ZAČÁTKU. Zjistil jsem že když jsem dělal vlastní symbol a pouzdro pro UWB, tak jsem nakreslil obráceně. Z toho důvodu jsem musel opravit symbol, pouzdro, schéma, PCB a obě zapojení na nepájivém poli. Bohužel stále nekomunikují, zahřívají se a z registrů mi přichází *podivná* data (jsou o jeden bit posunuta).

:(Uvidím jestli to bude fungovat, ale moc šancí tomu nedávám. I kdyby ten chip přežil, tak nejspíš rádio bude dead a budu muset koupit nový.

Souhrn

- Chyba v zapojení mi triggerovala WDT, ale myslel jsem, že je problém v kódu.
- Po dokončení ovládání ESP jsem se k UWB vrátil a zjistil chybu v původním zapojení.
- UWB stále nekomunikovalo, nakonec jsem našel chybu v návrhu symbolu a pouzdra – bylo obráceně.
- Opravil jsem schéma, PCB i zapojení, ale moduly stále nekomunikují, zahřívají se a registr vrací posunutá data.

- Moc šancí tomu nedávám, rádio bude nejspíš mrtvé a budu muset koupit nový čip.

2 PCB

Jen malé změny.

- header navíc pro encodery pro motory
- hodnoty rezistorů pro LED upraveny na 150 Ω

3 Autíčko

Součástky, které už jsem vybral jsou napsány v seznamu na spodu dokumentu.

3.1 konstrukce

Rozhodli jsme se, že "autíčko" (od teď to budu nazývat "tank") bude mít pásy, protože potom se dokáže otáčet na místě, což značně ulehčí matiku při zjišťování, kde se zrovna v prostoru nachází. Pro určení otočení použijeme magnetometr. Ten budeme používat pouze venku a uvnitř budeme doufat, že se mi podaří vytvořit dost dobrou matiku pro zjištění rotace pouze z pohybu pásů.

Šasí budeme prototypovat z lega, kvůli rychlosti stvoření prototypu a taky protože již zmíněné pásy budou lego pásy. Interface mezi motůrky a legem vytiskem na 3D tiskárně. Později můžeme celé šasí mít 3D čištěné.

3.2 napájení

Vzhledem k tomu, že jsem vybral 12V motůrky, tak jsem zvolil, že autíčko bude mít napětí 3S (11.1 V). Na to použiji 3x 18650. Ty bude možné nabíjet když budou uvnitř tanku. K tomu je potřeba balancer a nabíječka s BMS.

3.3 interface s navigací

Pro snadné připojení a odpojení jsme vybrali pogopiny. V momentální konfiguraci přes ně pujde:

- I²C s napájením a GND
- 4x PWM pro motory s GND
- 4x signal feedback z encoderů s napájením
- 5V a GND pro nabíjení navigace z auta

Vím že je tam víc GND než potřeba, ale je to dimenzovaný, aby každý jeden header dokázal plnit svou funkci bez nutnosti připojovat nějaký další.

Jak už jsem zmínil tak mě napadlo, že by bylo fajn, aby se navigace nabíjele, když bude připojená k tanku. Pro tento účel jsem do tanku vybral stepdown na 5 V, který se skrze pogopiny zapojí do nabíječky navigace.

4 kód

Stále je ještě všechn kód na test [branchy](#) (jmenuje se WiFi, pořád jsem to nepřejmenoval :))

4.1 wifi-scan

[link](#)

Program pro skenování WiFi signálu a ukládání do vhodného formátu pro pozdější použití v navigaci. K tomuhle jsem napsal README takže nemusím komplet vysvětlovat jak to funguje. Plánuju přidat nějaký UI kde by uživatel mohl jednoduše kliknout na mapku SPŠE a do informací o scanu by se vypsaly i jeho "souřadnice", které by mohli být později užitečnější než pouze si zapisovat jméno místa.

4.2 main

[link](#)

Tady ještě Readme není, takže to popíšu nějak zde.

- [/data](#) obsahuje .html stránky které se používají k ovládání ESP
 - index.html - main menu
 - test.html - test WebControlu, který jsem stvořil pro ovládání (více níže).
 - wifi.connection.html - slouží k manuálnímu připojení k WiFi
- [/src](#) - vlastní kód
 - /LED_test - slouží k testování WebControlu - jsem schopen skrze WEB zapnout/vypnout ledky, fetchnout status ledky, oscanovat WiFi síť a poslat string na ESP.
 - /utils - zatím obsahuje pouze funkce k WiFi, okomentoval jsem to celkem pěkně. Eventuálně doufám, že takhle bude okomentovaný i zbytek souborů.
 - /wifi.connection - slouží k ovládání wifi.connection.html
 - /wifi.location - otestovaný kód k určování polohy podle síly wifi signálu. Mam definovat struct do kterého funkce 'findMatchingLocation' zapíše jméno a souřadice místa kde se nacházíte.

- main.cpp - momentálně tam nejspíš bude poměrně starý kód, protože v tuhle chvíli slouží pouze jako testovací soubor, pro testování ostatních funkcí.

4.3 WebControl

Pro kontrolování ESP jsem se rozhodl jít cestou bez čudlíků. Používám Web-server který běží na ESP a handluje requesty od clientů. Na základě těchto requestů pošle .html stránku, data, nebo spustí nějakou funkci na ESP (zap/vyp LED).

5 co ještě chybí

Samozřejmě UWB, veškerá logika k jeho používání a naskenovat WiFi školy.

6 součástky

Zde jsou součástky které jsem vybral. Ještě se neobjednávali a tudíž se místo nákupu nebo její model ještě může změnit.

- motůrky - 12V 600RMP 2x - [link](#)
- stepdown - 5V - [link](#)
- balancer - [link](#)
- nabíječka - 3S - [link](#)
- pogopin - 2x 5, 1x 4, 1x 2 - [link](#)