

# Posudek diplomové práce

Matematicko-fyzikální fakulta Univerzity Karlovy

**Autor práce** Bc. Michal Raška  
**Název práce** Framework for Customizable Autopilot Solutions  
**Rok odevzdání** 2017  
**Studijní program** Informatika      **Studijní obor** Softwarové systémy

**Autor posudku** RNDr. Martin Kruliš, Ph.D.      **Role** Oponent  
**Pracoviště** Katedra softwarového inženýrství

## Text posudku:

Práce navrhuje nový framework pro bezpilotní modely letadel, který využívá tři propojených platforem – Arduino, Raspberry Pi a Android. Arduino zajišťuje obsluhu hardware a základní stabilizaci letounu. Raspberry Pi je spojujícím článkem a volitelně by mohlo provádět složitější výpočty. Mobilní telefon s Androidem poskytuje komunikaci s pozemní stanicí (druhý Android telefon) a případně další data z vlastních senzorů (GPS, akcelerometr). Celý systém je vysoce modulární a umožňuje řadu operačních módů.

Po stránce softwarového návrhu a implementace je práce na velmi vysoké úrovni. Autor demonstroval, že je schopen navrhnout modulární systém, integrovat různá hardwarová zařízení s použitím různých prostředí a různých programovacích jazyků. Výsledek této práce byl navíc použit v reálném provozu a byl schopen stabilizovat let bezmotorového letounu.

Text práce je nadprůměrný a dobře stylizovaný. Jeho patrně největším neduhem je, že často kolísá úroveň detailu – některé části jsou vysvětleny až zbytečně podrobně, jiné si musí čtenář domyslet. Dále je v textu několik konkrétních nedostatků:

- V analýze postrádám diskusi nad vybranými jazyky a technologie pro jednotlivé platformy (především Raspberry Pi a Android). Později se sice v kapitole o implementaci objeví zmínka o výběru ze tří možností pro Raspberry Pi (Java, C++ a Python), ale samotné zdůvodnění je poněkud zavádějící. V požadavcích je uvedeno, že je potřeba vybrat úsporný jazyk, ale následně si autor vybere Python místo C++. Potenciální nevhodnost volby ještě podtrhuje fakt, že C++ už je v projektu použito (Arduino) a celý framework má „panický mód“, který má za cíl odlehčit přetíženému Raspberry v kritických situacích.
- Vzoreček v popisu PID kontroléru (2.3.3) není dostatečně vysvětlen a není ani jasné použité značení.
- V textu (především v kapitole 3) se často střídají popisy pro různé platformy, což ztěžuje udržení pozornosti a porozumění textu.
- V sekci 3.3.3 (Discovery Provider) je uvedeno, že IP adresa Raspberry Pi je dynamická, a proto je potřeba implementovat mechanismus nalezení zařízení. Není však vysvětleno, proč nemůžeme použít statickou IP, která by se např. zadala do konfigurace Android aplikace, aby se zařízení nemusela vzájemně hledat.
- V sekci 4.4.1 (technické detaily platformy Arduino) je popsán problém se sdílenými globálními proměnnými, které jsou zároveň čteny hlavní smyčkou programu a zároveň modifikovány v obsluze přerušení. Autor správně navrhuje implementaci kritické sekce pomocí zákazu přerušení, ale zároveň uvádí, že přerušení, která nastanou v době zákazu přerušení, jsou ztracena. Podle informací, které jsem našel v dokumentaci, se přerušení pouze pozdrží, dokud není obsluha znovu povolena.

Na úplný závěr bych si dovolil marginální poznámku, která nemá vliv na hodnocení. Je chvályhodné, že autor v poděkování věnuje slovo také kapitánovi Kirkovi a Picardovi, nicméně absence kapitána Siska v tomto kontextu naznačuje, že další studium dané oblasti by mohlo být velmi užitečné.

**Práci doporučuji k obhajobě.**

**Práci nenavrhuji na zvláštní ocenění.**

*Pokud práci navrhuje na zvláštní ocenění (cena děkana apod.), prosím uveďte zde stručné zdůvodnění (vzniklé publikace, významnost tématu, inovativnost práce apod.).*

**Datum** 1. června 2017

**Podpis**