Платформа за безпилотен летателен апарат с четири ротора

Рафаел Калъчев

Технически Университет София



25 ФЕВРУАРИ 2022

Увод

Увод

Цели и задачи - Софтуер

Цели и задачи - Софтуер

- Изграждане на среда за разработка на софтуер с основа *Make* под *Linux*
- Подбор на хардуерни и софтуерни решения за интеграция с разработената среда
- Създаване на софтуерни модули и драйвъри за работа с вътрешния и външен хардуер
- Подбор на изходни портове, пинове и нужна периферия
- Инициализация и конфигурация на микроконтролера и периферията

Увод

Цели и задачи - Хардуер

Цели и задачи - Хардуер

- Разположение на хардуера и избор на конфигурация за системата
- Подбор на изходни портове, пинове и нужна периферия
- Свързване на компонентите и електроснабдяване
- Изграждане платформи за опитни постановки
- Сглобаване на системата

Увод

Цели и задачи - Идентификация, моделиране, наблюдение и управление

Цели и задачи - Идентификация, моделиране, наблюдение и управление

- Моделиране:
 - ▶ Ротор с витло
 - ▶ Платформа за управление на ъгъл на завъртане
 - ▶ Платформа за безпилотен летателен апарат с четири ротора
- Идентификация на параметри
- Компенсация на сензорни отмествания
- Калибриране на сензорите
- Синтез на наблюдател за оценка на ориентацията на платформата
- Синтез на управление

Софтуерна част

Софтуерна част

Среда за разработка на софтуер с основа *Make* под *Linux*

Описание и съставни части

Състав на средата за работа под *Linux*

- Система за насочено изграждане: GNU Make
- **Компилатор:**GCC ARM NON-EABI
- **■ Връзка с контролера:** *ST-LINK*
- **Текстов редактор:** *VIM* + *Ctags*
- Дебъгер: GDB (GNU Project Debuger)







Команден итерфейс

Подържани команди

make | make all Цялостно изграждане чрез компилиране и свързване на всчики *нужни* елементи.

make clean Изчистване на средата.

make flash Запис на изградения ¹ софтуер в паметта на микроконтролера.

1.1

¹В случай, че софтуерът има нужда от (пре)изграждане, системата автоматично го (пре)изгражда.

make debug Отваряне на порт за дебъг, стартиране и свързване на дебъгера 2

make usart Стартира интерактивен комуникационен интерфейс за връзка (IO) с контролера чрез USART.

make usart_read Създава файл, съдържащ получените данни през USART порта.

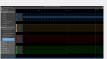
12

 $^{^2}$ Системата автоматино конфигурира дебъгера да използва дебъг символите в изградения софтуер.

Подбрани хардуерни и софтуерни решения за анализ

- Логически анализатор и генератор на логически сигнали
 - ► SQ50 logic analyzer (IKALOGIC S.A.S)
 - ► ScanaStudio 16.04 (for Linux)
- UART / USB Конвертор + picocom







Решени задачи по отнощение на средата

Конфигурация на компилатор

- Използване на хардуерния модул за числа с плаваща запетая
- Изключване на оптимизациите
- Генериране на символи за дебъг

Линкерен скрипт

- Очертаване на разположението на отделните секции в паметта
- Генериране на символи, използвани от буутлоудъра.
- Правилно подравняване на отделните секции.
- Остранване на символи от стнадартните библиотеки.

Конфигурация на дебъгер

- Автоматично отваряне на сесия и подвързване към дебъгерният порт.
- Предварително зареждане на дебъг символите от последния обектен файл.

Софтуерна част

Софтуерни модули за вътрешна и външна периферия

Описание на модулите за вътрешна периферия

Софтуерните модули са изградени на база описанието на регистирте за вътрешната периферия, посочени в документа за техническа справка на микроконтролери от семейство *STM32F4xxx* [2].

Основите за регистърните блокове на периферията в паметта са описани в дкоумента за съответния модел микроконтролер [3].

Изградени модули за вътрешна периферия

RCC (Reset and Clock Control)

Модул за управление на:

- Времеви бази (clock source: HSE, HSI, LSE)
- PLL честотен множител
- Делители на честота
- Часовници на периферните шини
- Часовници на периферията

NVIC (Nested Vectored Interrupt Controller)

Модул за управление на:

- Конфигурацията на NVIC (контролера).
- Разрешаване и забрана на отделни прекъсвания.
- Маскиране на прекъсвания.
- Приоритет на прекъсванията
- Локация на таблицата на прекъсванията.

GPIO (General Purpose Input/Output)

Модул за управление на:

- Отделните GPIO портове.
- Посока (изход/вход) за отделни пинове.
- Режим на работа на отделните пинове (Open-drain, Push-Pull, Analog, AF).
- Вътрешни (Pull-up/Pull-down/Floating) конфигурации.
- Източник за управление на състоянието на отделни пинове (Регистър/Алтернативна функция)
- Работна честота на модула.

TIM (Timers)

Модул за управление на прости таймери, таймери с общо предназначение и специализирани таймери:

- Управление на честотните делители.
- Настройка на период на таймера (ARR).
- Настройка на режим на броене.
- Настгойка на флагове, стартиращи прекъсвания.
- Настройка на режимите на използване на ССR регистрите.

I2C (Inter Integrated Circuit)

Модулът предоставя:

- Конфигуриране на отделните I2С периферни модули.
- Прости команди за отделни операции във връзка с I2C комуникацията.
- Абстрактни команди за управление на комуникацията изцяло през прекъсването.
- Динамична поправка на проблеми с комуникацията (при възможност).

USART

Модулът предоставя:

- Опростен интерфейс за работа с USART периферията.
- Настройка на честота на предаване/приемане
- Настройка на отделни параметри на комуникацията.

Софтуерна част

Описание на модулите за външна периферия

Описание на модулите за външна периферия

Софтуерните модули за външната периферия са изградени на база описанието и комуникационните интерфейси на отделното използвано устройство.

Строго проекто специфични, като се възползват от знанията за целите на проекта и са оптимизирани само и единствено за него.

Изградени модули за външна периферия

6-канален RC приемник

Модул за четене и обновяване на данните, постъпващи от 6-канален RC приемник.

- Сигнал тип PPM 50Hz. $(1000 \rightarrow 2000 \mu s)$
- Използват се TIM3 и TIM5 (резолюция 0.2μ s), режим Input-Capture.
- Сигналите се подновяват в струкрурата на суровите входове в момента на засичнае

BLDC 4 канала

Модул за управление на заданията на четирите ESC за BLDC.

- \blacksquare Сигнал тип PPM 50Hz. (1000 $o 2000 \mu$ s)
- Използва се TIM2 (резолюция 1μ s), режим Output-Compare.
- Сигналите се подновяват след рестартиране на брояча чрез писане върху ССR регистрите.
- Режим на изходните пинове Open-Drain No-Internal-Pull.

Жироскоп и акселерометър

Модул за конфигурация и четене на данните от жироскопа и акселерометъра:

- Сигнал тип I2C.
- Използва се I2C3, режим Master.
- Хадуерът е конфигуриран на 52Hz честота на дискретизация.
- Обхват: акселерометър: 8g, жироскоп: 1000DPS.

Магнитометър

Модул за конфигурация и четене на данните от магнитометъра:

- Сигнал тип I2C.
- Използва се I2C3, режим Master.
- Хадуерът е конфигуриран на 52Hz честота на дискретизация.
- Обхват: 8gaus.

Софтуерна част

Допълнителни библиотеки

Допълнителни библотеки

Библиотека за работа с матрици (стандарт за именоване CMSIS)

- Събиране и изваждане
- Умножение
- Транспониране
- Инверсия (Гаус-Жордан)
- Псевдоинверсия (Пенроз-Мур) (WIP)

Библиотека за управление

- Универсален ПИД регулатор
- Релеен регулатор
- Ограничители max-min

Софтуерна част

Инициализация на микроконтролера

Инициализация на микроконтролера

Стартиране

Започва от ResetHandler, където е имплементиран примитивен буутлоудър

Примитивен буутлоудър (ARM assembly)

- Инициализира стековия указател (SP) и програмния брояч (PC)
- Зарежда статичните променливи в паметта
- Инициализира .bss секцията с 0.
- Извиква системната инициализация и отдава управлението към main()
- Дефинира символите от таблицата на прекъсванията като .weak към DefaultHandler

Системна инициализация

- Конфигураця на системния часовник (168MHz)
 - ▶ Използване на HSE за времева основа
 - ► Конфигурация на PLL (PLL_M=8, PLL_N=336, PLL_P=2, PLL_Q=7)
- Конфигурация на АНВ и APB* (AHB_Prescaler=1, APB1_Prescaler=4, APB2_Prescaler=2).
- Разрешаване на хадруерното устройство за числа с плаваща запетая.

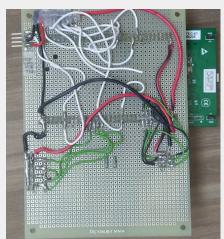
Хардуерна част

Хардуерна част

Свързване на системата

Свързваща платка





Предна страна

Задна страна

Подбрани портове, пинове и периферия

I2C

РА8 (I2C3 SCL) и РА9 (I2C3 SDA). Цялата периферия по шината оперира на 3.3V.

USART

РВ6(USART1 TX) и РВ7 (USART1 RX). Цялата периферия по шината оперира на 3.3V.

Управляващи сигнали ESC

PD12 - PD15. Конфигурация Open-drain, външни Pull-up резистори $10M\Omega$. Нужно е да се генерира 5V PPM сигнал.

Входове РРМ (приемник, сензор за разстояние)

РАО(ТІМ5 канал 1), РАЗ(ТІМ5 канал 4), РВО(ТІМ3 канал 3), РВ1(ТІМ3 канал 4), РВ4(ТІМ3 канал 2), РВ5(ТІМ3 канал 1). Периферията генерира 5V сигнали на входа. Входовете са 5V толерантни.

Балансиране на витла



Фигура: Небалансирано витло

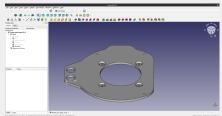


Фигура: Балансирано витло

Основа за монтиране на контролер и външна периферия



Фигура: Основа за монтиране, Ultimaker Cura



Фигура: Основа за монтиране FreeCad

Платформа за управление на ъгъл на завъртане



Платформа за безпилотен летателен апарат с четири ротора



Моделиране и идентификация

Моделиране и идентификация

Ротор с витло

Моделиране на ротор с витло

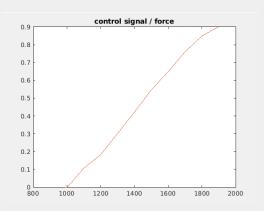
Опростен апроксимиран модел - нискочестотен филтър и коефициент на пропорционалност.



Фигура: Опростен модел на витло и мотор

$$G_{motor}(s) = \frac{k_{motor}}{s\tau_{motor} + 1}$$

Снемане на характеристики на ротор с витло



 $k_{motor} = 12.2 \emph{e} - 3$ при сигнал $1300 \to 1700$. Опорната точка при тяга $424 \emph{g} \approx 4.15 \emph{N}$

Моделиране и идентификация

Платформа за управление на ъгъл на завъртане

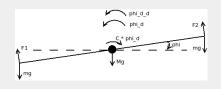
Моделиране на платформа за Управление на ъгъл на завъртане

Ако приемем:

$$F_2 = F_0 + \delta f, F_1 = F_0 - \delta f$$

И апроксимираме:

$$I = \frac{l^2}{12}(\mathbf{M} + 6\mathbf{m})$$



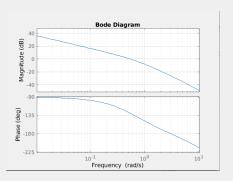
Получаваме:

$$\textit{G}_{\textit{sys}} = \textit{G}_{\textit{motor}}(\textit{s}) \textit{G}_{\textit{platform}}(\textit{s}) = \frac{\textit{k}_{\textit{motor}} \textit{l}}{\textit{s}(\textit{sl} + \textit{C})(\textit{s}\tau_{\textit{motor}} + 1)}$$

Идентификация на параметри на платформа за управление на ъгъл на завъртане

След снемане на преходен процес:

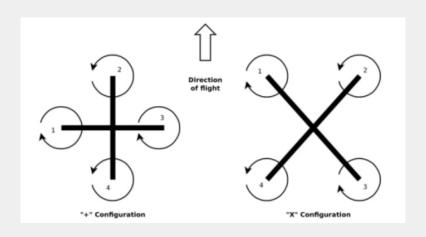
$$I = 0.025, C = 0.018, \tau_{motor} = 0.08$$



Моделиране и идентификация

Платформа за безпилотен летателен апарат с четири ротора

Конфигурация



Сили, моменти и отправни координатни системи



Моделиране

За да се носи платформата във въздуха, е нужно:

$$\sum_{i=1}^{4} T_i = - mg$$

$$\sum_{i=1}^{4} M_i = 0$$

$$T_{1,2,3,4} || g$$

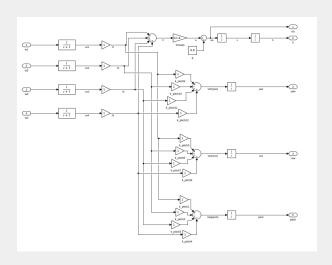
$$(\omega_1 + \omega_3) - (\omega_2 + \omega_4) = 0$$

от което и следва, че $\phi = 0, \theta = 0, \psi = 0$.

За промяна на ориентацията се налага да:

$$\begin{split} \dot{\psi} &= \mathbf{k}_{\psi}((\omega_{1} + \omega_{3}) - (\omega_{2} + \omega_{4})), \psi = \int \dot{\psi} d\mathbf{t} \\ \dot{\phi} &= \mathbf{k}_{\phi}((\omega_{1} + \omega_{4}) - (\omega_{2} + \omega_{3})), \phi = \int \dot{\phi} d\mathbf{t} \\ \dot{\theta} &= \mathbf{k}_{\theta}((\omega_{1} + \omega_{2}) - (\omega_{3} + \omega_{4})), \theta = \int \dot{\theta} d\mathbf{t} \end{split}$$

Моделиране на системата чрез блок схема

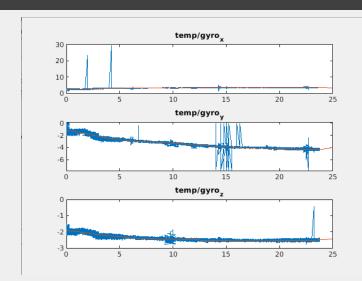


Калибриране и компенсация

Калибриране и компенсация

Жироскопен дрейф

Компенсация на жироскопен дрейф по температура



p_x =

 $-42.6978e-9 \text{ x}^7 + 3.7797e-6 \text{ x}^6 -131.8066e-006 \text{ x}^5 + 2.2717e-3 \text{ x}^4 -19.5918e-3 \text{ x}^3 + 67.9744e-3 \text{ x}^2 + 95.1384e-3 \text{ x} + 2.2440e+0$

 $p_y =$

 $87.1716e-9 \text{ x}^7 -7.6384e-6 \text{ x}^6 + 263.1019e-006 \text{ x}^5 -4.4660e-3 \text{ x}^4 37.8030e-3 \text{ x}^3 -129.2365e-3 \text{ x}^2 -191.6577e-3 \text{ x} -1.1695e+0$

 $p_z =$

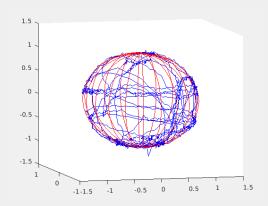
 $15.9591e-9 \text{ x}^7 -1.4613e-6 \text{ x}^6 + 53.0721e-006 \text{ x}^5 -961.5439e-6 \text{ x}^4 + 8.8365e-3 \text{ x}^3 -33.3197e-3 \text{ x}^2 -44.0584e-003 \text{ x} -1.9076e+0$

62

Калибриране и компенсация

Магнитни отмествания от околната среда

Компенсация на магнитни отмествания от околната среда



Използвана е техника на Мерайо[1] (трансформация отместен елипсоид - центрирана сфера).

21.4229e+000 837.9000e-003 3.0861e+000 0.0000e+000 21.6201e+000 2.0207e+000 0.0000e+000 0.0000e+000 27.2441e+000

c =

-1.6000e-003

19.9000e-003

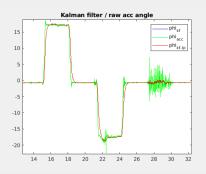
45.7000e-003

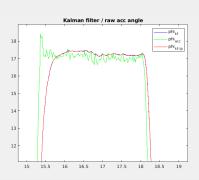
Синтез на наблюдател

Филтър на Калман за оценка на ъгъл на завъртане

Ковариантни матрици

Проверка на резултатите





Синтез на управление

Управление на ъгъл на завъртане

Опростена схема на управлението:

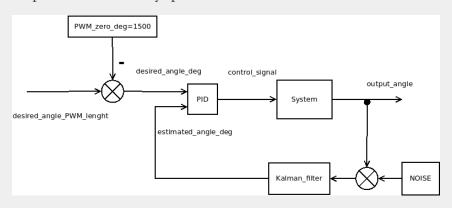
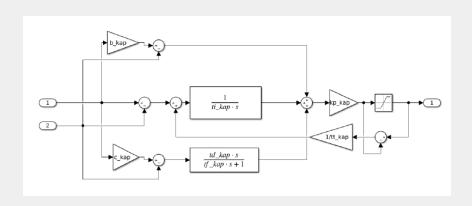
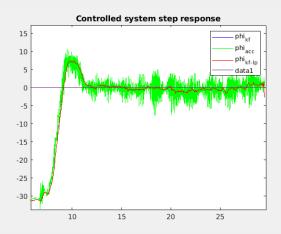


Схема на имплементирания универсален ПИД



Параметри на регулатор и преходен процес

$$\mathit{Kp} = 0.4, \mathit{K_i} = 0.003, \mathit{K_d} = 2, \tau_d = 0.26, \mathit{out_{limit}} = \pm 150$$



Литература



José MG Merayo, Peter Brauer, Fritz Primdahl, Jan Raagaard Petersen, and Otto V Nielsen.

Scalar calibration of vector magnetometers.

Measurement science and technology, 11(2):120, 2000.



ST MICROELECTRONICS.

RM0090 Reference manual STM32F405/415, STM32F407/417, STM32F427/437 and STM32F429/439 advanced ARM-based 32-bit MCUs.

ST Microelectronics, 2020.



ST MICROELECTRONICS.

STM32F427xx STM32F429xx 32b Arm Cortex-M4 MCU+FPU, 225DMIPS, up to 2MB Flash/256+4KB RAM, USB OTG HS/FS, Ethernet, 17 TIMs, 3 ADCs, 20 com. interfaces, camera and LCD-TFT.

ST Microelectronics, 2021.

Благодаря за вниманието!