

Платформа за безпилотен летателен апарат с четири ротора

РАФАЕЛ КАЛЪЧЕВ

ТЕХНИЧЕСКИ УНИВЕРСИТЕТ СОФИЯ

25 ФЕВРУАРИ 2022



Структура

1 Структура

2 Увод

- Цели и задачи - Софтуер
- Цели и задачи - Хардуер
- Цели и задачи - Идентификация, моделиране, наблюдение и управление

3 Софтуерна част

- Среда за разработка на софтуер с основа *Make* под *Linux*
- Софтуерни модули за вътрешната и външна периферия

4 Хардуерна част

5 Моделиране и Идентификация

- Ротор с витло
- Платформа за управление на ъгъл на завъртане

Увод

Увод

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

- Оглед на съществуващия хардуер и подбор на компоненти за преизползване.
- Подбор нов на 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
- Сглобяване 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

- Компилятор:

GCC ARM NON-EABI

- Връзка с контролера:

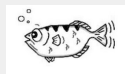
ST-LINK

- Текстов редактор:

VIM + Ctags

- Дебъгер:

GDB (GNU Project Debugger)



Подържани команди

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

make clean Изчистване на средата.

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

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

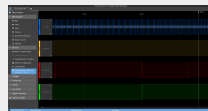
make usart Стартира интерактивен комуникационен интерфейс за връзка (IO) с контролера чрез USART.

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

²Системата автоматично конфигурира дебъгера да използва дебъг символите в изградения софтуер.

ПОДБРАНИ ХАРДУЕРНИ И СОФТУЕРНИ РЕШЕНИЯ

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



Софтуерна част

Софтуерни модули за вътрешната и външна периферия

ОПИСАНИЕ НА МОДУЛИТЕ ЗА ВЪТРЕШНА ПЕРИФЕРИЯ

Софтуерните модули са изградени на база описанието на регистрите за вътрешната периферия посочени в Документът за Техническа справка на микроконтролери от семейство *STM32F4xxx* [1].

А основите за регистърните блокове на периферията в паметта е както е описано в дкоумента за съответният модел микроконтролер [2].

RCC (Reset and Clock Control)

Модул за управление на:

- Времеви баци (clock source: HSE, HSI, LSE)
- PLL честотен множител
- Делители на честота
- Часовниците на периферните шини
- Часовниците 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
- Локацията 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. Модула предоставя следните възможности: AAAAAAAAAAAAAAAAAAAAAA-
AAA-
AAA

I2C (Inter Integrated Circuit)

Модул за управление на конфигурацията на отделните I2C периферни модули. Предоставя ниво на абстракция позволяваща лесно четене и писане по регистрите на външни устройства, като цялото управление на потока на комуникация е част от имплементираното прекъсване.

Хардуерна част

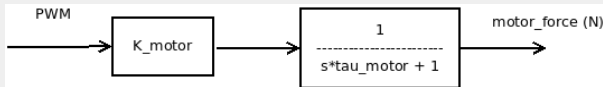
Моделиране и Идентификация

Моделиране и Идентификация

Ротор с витло

МОДЕЛИРАНЕ НА РОТОР С ВИТЛО

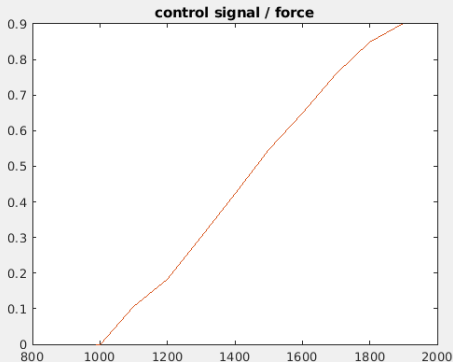
Опростен апроксимиран модел - нискочестотен филтър и коефициент на пропорционалност.



Фигура: Опростен модел на витло и мотор

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

СЧЕМАНЕ НА ХАРАКТЕРИСТИКИ НА РОТОР С ВИТ-ЛО



$k_{motor} = 12.2e - 3$ при сигнал $1300 \rightarrow 1700$.

Опорната точка при тяга $424g \approx 4.15N$

Моделиране и Идентификация

Платформа за управление на тъгъл на завъртане

МОДЕЛИРАНЕ НА ПЛАТФОРМА ЗА УПРАВЛЕНИЕ НА ЪГЪЛ НА ЗАВЪРТАНЕ

Ако приемем:

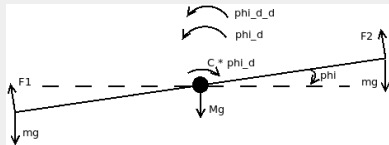
$$F_2 = F_0 + \delta f, F_1 = F_0 - \delta f$$

И апроксимираме:

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

Получаваме:

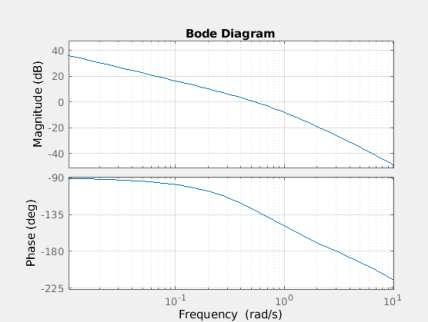
$$G_{sys} = G_{motor}(s)G_{platform}(s) = \frac{k_{motor}l}{s(sl + C)(s\tau_{motor} + 1)}$$



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

AA

Калибриране и компенсация

Синтез на наблюдател

Синтез на управление

ТАЗИ ПРЕЗЕНТАЦИЯ Е
СЪЗДАДЕНА С ПОМОЩТА НА
beamer L^AT_EX- *Focus*

СПЕЦИЯЛНИ БЛАГОДАРНОСТИ НА
Кристина Дойчева ЗА
РЕДАКЦИЯ И ОФРМЛЕНИЕ.



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.

BACKUP SLIDE

This is a backup slide, useful to include additional materials to answer questions from the audience.

The package `appendixnumberbeamer` is used to refrain from numbering appendix slides.