

5 laboratorinio darbo aprašymas

32 BITŲ MIKROVALDIKLIŲ SPI PROGRAMAVIMAS

1. Darbo tikslas
2. Pasiruošimas darbui
3. Darbo eiga
 - 3.1. Darbas su dėstytoju
 - 3.1.1. STM32CubeIDE
 - 3.1.2. Kodo šablonas
 - 3.1.3. Svarbiausios naudojamos funkcijos
 - 3.2. Darbas pagal individualią užduotį
 - 3.3. Papildoma literatūra

1. Darbo tikslas

Susipažinti su SPI nuosekliaja duomenų perdavimo sąsaja. Formuoti gebėjimus analizuoti mikrovaldiklių specifikacijas.

2. Pasiruošimas darbui

Šio laboratorinio darbo metu pradėsime dirbti su SPI nuosekliaja sąsaja. Susipažinę su pagrindais studentai savarankiškai bandys atlikti keletą užduočių.

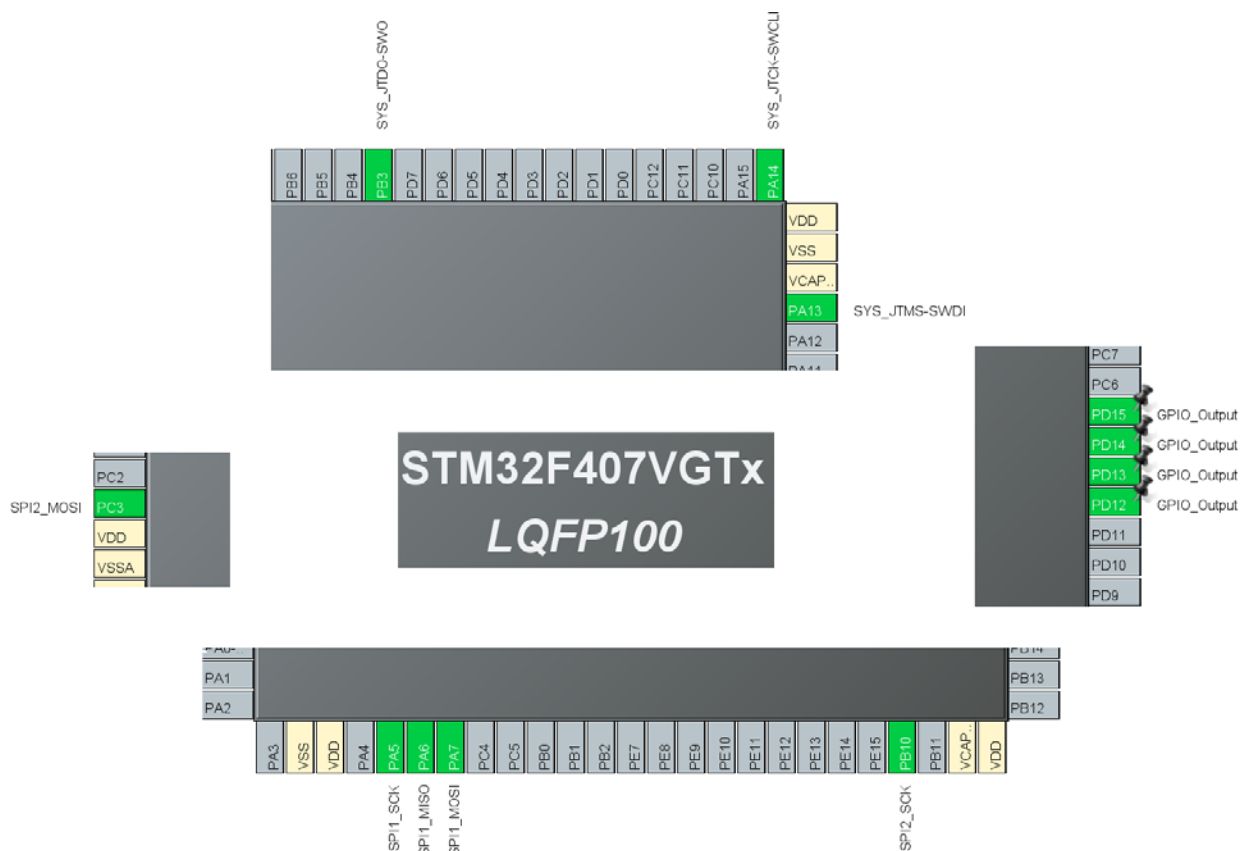
3. Darbo eiga

3.1. Darbas su dėstytoju

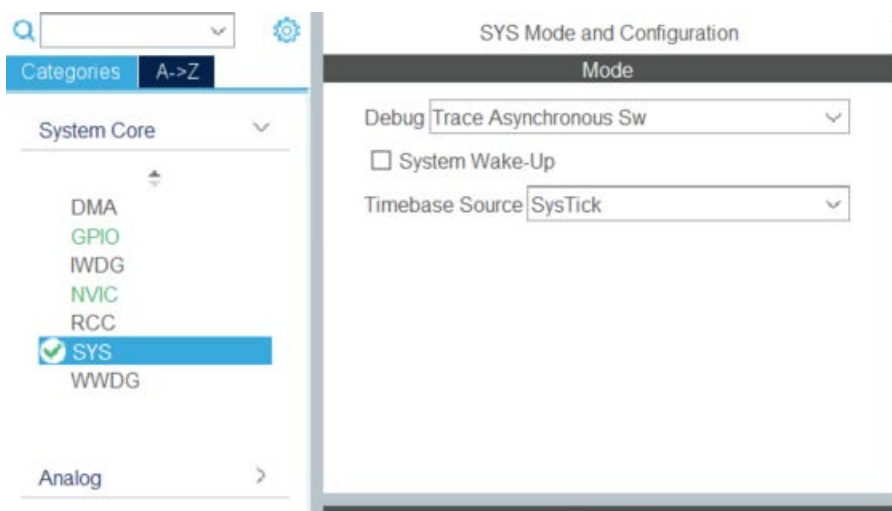
Laboratorinio darbo metu visų pirma dėstytojas pademonstruoja maketus, paaiškina jų sudedamąsias dalis bei programavimo ypatumus.

3.1.1. STM32CubeIDE

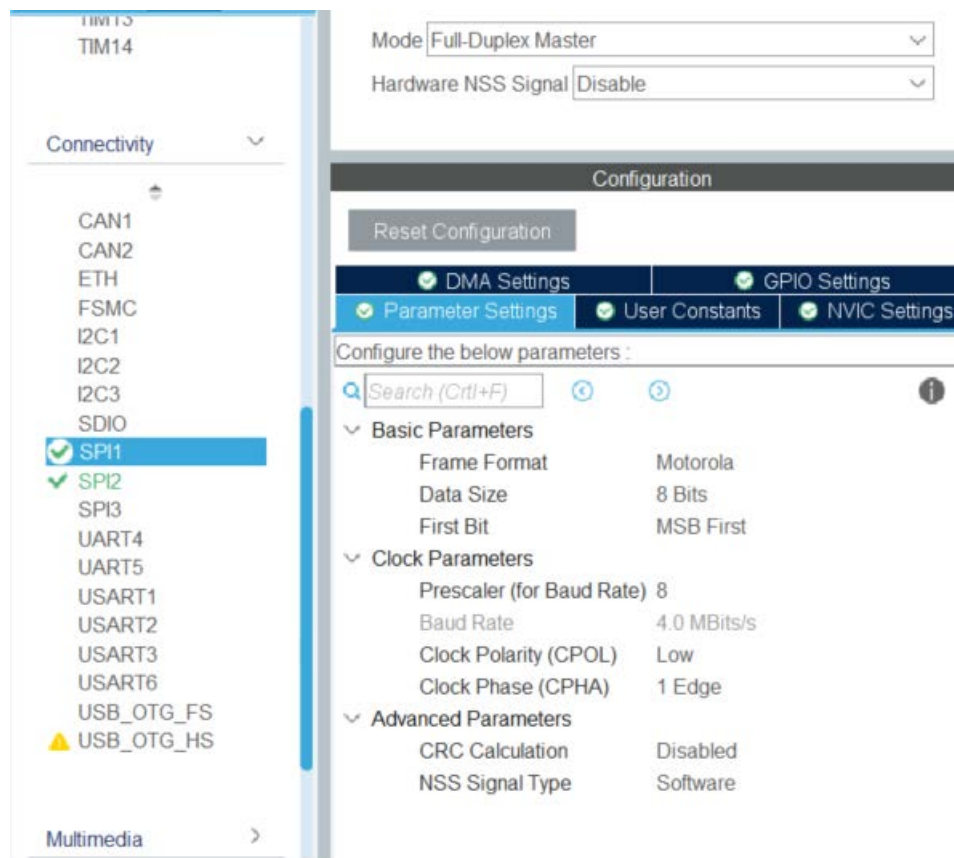
Detalus projekto konfigūravimas jau buvo aptartas pirmųjų trijų laboratorinių darbų aprašyme, todėl čia pateikiu tik esminius 5 laboratoriniam darbui atlikti reikalingus grafinio konfigūravimo lango printscreen'us. Pateikiu paveikslus jų nedetalizuodamas. Tai yra skirtinguose variantuose jums reikės skirtingos konfigūracijos. Šių paveikslų tikslas pasufleruoti, kur reikalingą konfigūraciją galima surasti.



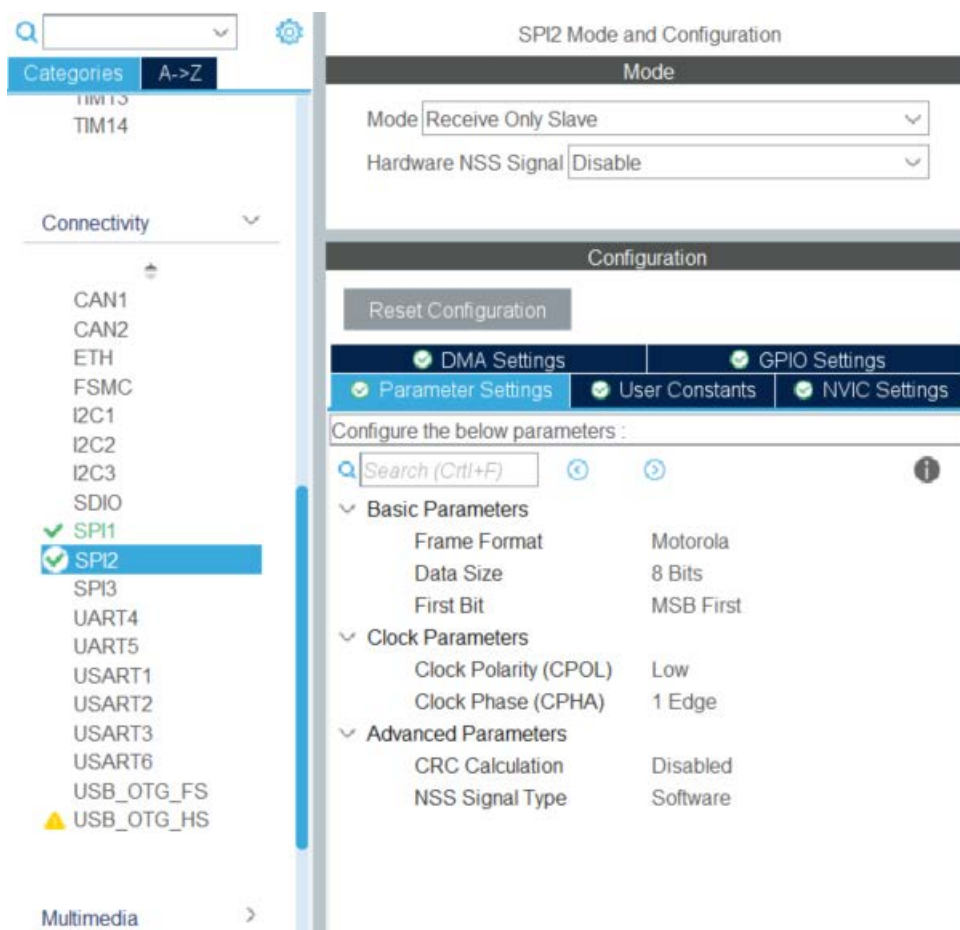
1 pav. SPI1, SPI2 ir PD12, PD13, PD14, PD15 konfigūracija. Taip pat Trace Asynchronous debug konfigūracija. Kai kuriuose variantuose taip pat gali reikėti SPI3.



2 pav. Konfigūracinio „Debug“ režimo įjungimas



3 pav. SPI1 konfigūracija, kai SPI1 turi dirbti *Master Full Duplex* režime.



4 pav. SPI2 konfigūracinis langas, kai SPI2 turi dirbti *Slave Read Only* režime.

Primenu, kad tai nėra visų galinčių skirtinguose variantuose pasitaikyti konfigūracijų pavyzdžiai. Tai yra apibendrinti pavyzdžiai. Savo variante jums reiks prisitaikyti prie gautos sąlygos.

3.1.2. Kodo šablonas

Kodo šablono pavyzdžiai yra pateikti ankstesnių laboratorinių darbų aprašymuose. Primenu, kad jums, priklausomai nuo jūsų varianto, reikės dirbti su *main.c* ir *stm32f4xx_it.c* failiukais.

3.1.3. Svarbiausios naudojamos funkcijos V-ajame laboratoriniame darbe

- HAL_SPI_TransmitReceive(...) – Blocking mode: SPI transmit and receive.
- HAL_TIM_Base_Start_IT(...) – Non-Blocking mode: TIMx start.
- HAL_SPI_Receive_IT(...) – wait for SPI receive event in non-blocking mode.
- HAL_SPI_Transmit(...) – Transmit data using SPI.
- TIMx_IRQHandler(...) – Function handles TIMx global interrupt.
- SPIx_IRQHandler (...) – Function handles SPIx global interrupt.

Duomenų struktūrų pavyzdžiai

- uint8_t masyvas[2] = { 1, 2 }; – dviejų celių masyvas
- int a = 10; a – bus grąžinama reikšmė 10; &a – bus grąžinamas adresas, kuriuo yra išsaugota reikšmė 10.

3.2. Darbas pagal individualią užduotį

Studentai savarankiškai atlieka dėstytojo pateiktas individualias darbo užduotis. Studentai gali naudotis bet kokia papildoma literatūra, bet negali tartis tarpusavyje. Susidūrę su sunkumais ir nerandantys sprendimo studentai visada gali užduoti konkrečius klausimus dėstytojui.

3.3. Papildoma literatūra

1. Paskaitų skaidrės. Galite rasti moodle sistemoje.
2. STM32F407. Galite rasti moodle sistemoje.
3. STM32F407 Discovery. Galite rasti moodle sistemoje.
4. UM1467 Getting started with software and firmware environments for the STM32F4DISCOVERY Kit. Galite rasti moodle sistemoje.
5. UM1472 Discovery kit for STM32F4. Galite rasti moodle sistemoje.
6. DM00031020 stm32f4x reference manual registers level. Galite rasti moodle sistemoje.
7. Pavyzdžių paieška google paieškos sistemoje.