Osnove mikroprocesorske elektronike

Priprava 10: LCD zaslon

Pri tej vaji bomo oživili LCD zaslon. *Uporabno funkcionalnost* LCD zaslona bomo dosegli tako, da bomo uspešno oživili sledeče pod-module:

1) *modul za osvetljevanje* LCD zaslona z zadnje strani (angl. LCD backlight),
2) modul za izris grafike na LCD zaslon *na osnovnem nivoju* preko vmesnika ILI9341,
3) *grafično knjižnico* `uGUI`, ki bistveno poenostavi izris grafike na zaslon,
4) *modul za detekcijo lokacije pritiska* na LCD zaslon (angl. LCD touch sreen functionality) na podlagi čipa XPT2046.

Vse te module bomo povezali znotraj modula `lcd.c`.

Tekom priprave boste predvsem poskrbeli, da se za vsakega od pod-modulov *pripravi nizko-nivojska inicializacija strojne opreme* s pomočjo orodja CubeMX, da v projekt *vključite vse potrebne datoteke* ter da *poiščete določene informacije*, ki vam bodo prišle prav pri oživljanju pod-modulov.

Priprava novega projekta

1. ***S pomočjo orodja*** `STM Cube Project Copy` ***ustvarite nov projekt z imenom***

   `VAJA_10-LCD`

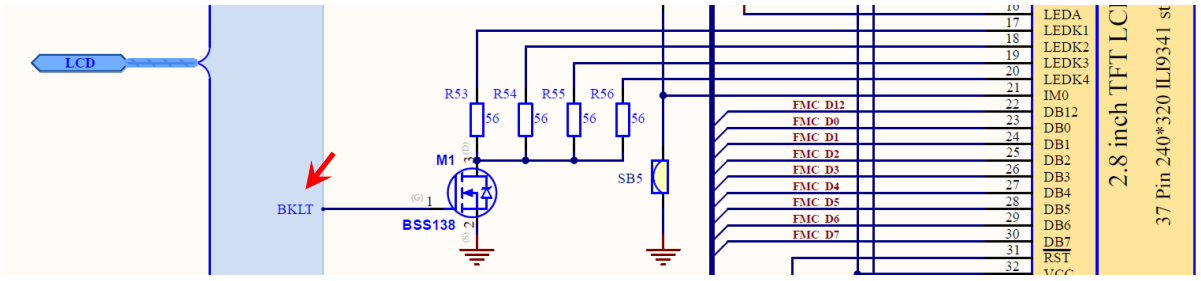
   na podlagi projekta, kjer ste rešili prejšnjo vajo. Projekt nato uvozite v "STM Cube IDE".

Priprava – osvetlitev LCD zaslona

1. ***Novemu projektu dodajte modula*** `lcd_backlight.c` ***in*** `lcd_backlight.h`, znotraj katerega bomo v sklopu vaje implementirali funkcije za krmiljenje osvetlitve. Modula najdete v mapi "predloge\backlight". Dodajte ju v projektno mapo znotraj podmape "System" na ustrezno mesto.

2. ***Preučite, kateri pin mikrokrmilnika je uporabljen za krmiljenje osvetlitve.***

   Osvetlitev LCD zaslona uravnavamo s pomočjo *pulznega krmiljenja* MOSFET tranzistorja, ki vklopi napajanje za LED osvetlitev LCD zaslona. S spreminjanjem širine pulza za vklop osvetlitve lahko dosežemo različne jakosti osvetlitve LCD zaslona.



   Ugotovite torej, kateri pin mikrokrmilnika krmili signal `BKLT`. Pomagajte si z izseki iz sheme, ki jih najdete na spodnjih slikah.

3. ***V orodju CubeMX nastavite funkcijo tega pina tako, da bo <u>povezan na kanal časovnika</u>*** `TIM4`.

4. ***Preučite in dopolnite sledeči razmislek glede generiranja PWM signala za krmiljenje osvetlitve:***

   kakšna naj bo frekvenca PWM krmilnega signala?

   Frekvenca mora biti dovolj visoka, da oko ne zazna utripanja svetlobe. Ni pa potrebno in tudi ni smiselno, da bi bila zelo visoka. Smiselna frekvenca PWM signala se zdi nekako v področju 100-200 Hz.

   To nam je sedaj izhodišče za nastavitev delilnika časovnika ("prescaler") in modula časovnika ("period").

   Zaradi nazornosti lahko nastavimo časovnik tako, da bo frekvenca PWM signala 100 Hz,

   $$f_{\mathrm{PWM}} = 100\ \mathrm{Hz}\ .$$

   To pomeni, da je perioda PWM signala

   $$T_{\mathrm{PWM}} = \frac{1}{f_{\mathrm{PWM}}} = \qquad \mathrm{ms}\ .$$

   <u>Odločitev:</u> *svetilnost bi želeli nastavljati na procent natančno*. Zato se zdi smiselno, da osnovno periodo PWM signala razdelimo na 100 delov, torej bo *osnovna enota* časa števca,

   $$\Delta t = \frac{T_{\mathrm{PWM}}}{100} = \qquad \mathrm{us}\ .$$

   In hkrati je potem smiselno, da modul števca nastavimo tako, da bo preštel 100 takih enot, torej bo

   $$\mathrm{modul} = \qquad .$$

   Le še premislimo, kako je potrebno nastaviti pred-delilnik, da bo naš časovnik štel z osnovno enoto $\Delta t = 100$ us. Upoštevamo, da je interna ura, ki poganja števec, enaka $f_{\mathrm{APB1}} = 144$ MHz. Tako sledi

   $$\mathrm{prescaler} = \frac{\Delta t}{T_{\mathrm{APB1}}} = \frac{\Delta t}{\frac{1}{f_{\mathrm{APB1}}}} = \Delta t \cdot f_{\mathrm{APB1}} =$$

   Ugotovitve tega razmisleka uporabite za nastavitev pred-delilnika in periode časovnika v naslednji točki.

5. ***Poskrbite za sledeče nastavitve časovnika*** `TIM4`:
   a) za *vir ure* časovnika izberite notranjo uro (angl. internal clock),
   b) na ustreznemu *kanalu časovnika* (angl. timer channel) izberite funkcijo generiranja PWM signala ("PWM generation CHx"),

      (namig: ustrezno številko kanala lahko ugotovite iz poimenovanja izbrane funkcije pina mikrokrmilnika),
   c) smiselno nastavite periodo časovnika,
   d) smiselno nastavite pred-delilnik ("prescaler").

6. ***Poskrbite, da se bo za*** `TIM4` ***modul uporabljala LL knjižnica*** (angl. low-level) in ne HAL knjižnica.

7. ***Shranite nastavitve v orodju CubeMX in potrdite avtomatsko generiranje nove kode za inicializacijo časovnika*** `TIM4`**.**

   Podobno kot pri prejšnjih vajah, ste s tem korakom poskrbeli, da se bo dodala nizko-nivojska programska koda za inicializacijo časovnika TIM4, ki ga bomo uporabili za implementacijo krmilnika osvetlitve na sistemskem nivoju.

```
/* Initialize all configured peripherals */
MX_GPIO_Init();
MX_USART3_UART_Init();
MX_TIM6_Init();
MX_TIM4_Init();
/* USER CODE BEGIN 2 */
```

8. ***Preučite, katere nizko-nivojske funkcije bomo potrebovali za delo s časovnikom.***

   Potrebovali bomo sledečo nizko-nivojsko funkcionalnost LL knjižnice:
   a) omogočanje časovnika,
   b) omogočanje kanala za *izhodno primerjavo* (angl. output compare),
   c) nastavitev registra za primerjavo,
   d) branje registra za primerjavo.

   Tokrat so te funkcije *že navedene v seznamu zglavne datoteke* `lcd_backlight.h`. Preučite, kaj te funkcije počnejo ter kako se jih uporabi.

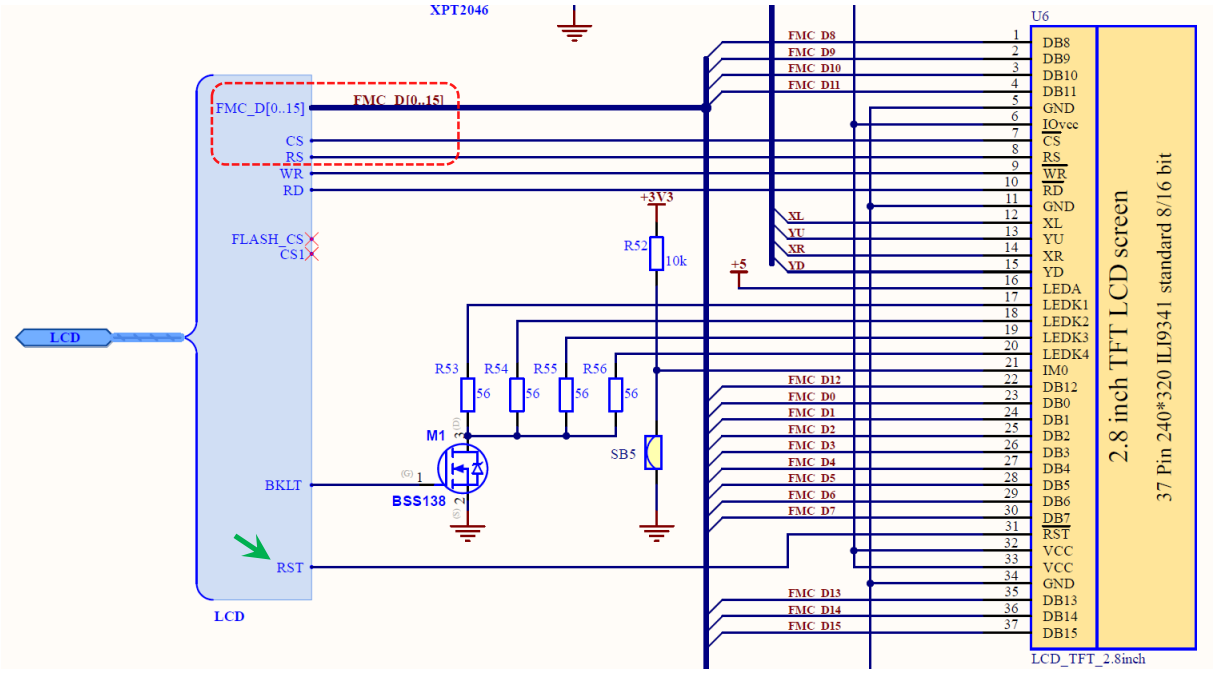
```
// Pri implementaciji sistemskih funkcij za upravljanje časovnika bomo
// potrebovali sledeče nizko-nivojske funkcije:
//      - void LL_TIM_EnableCounter (TIM_TypeDef * TIMx)
//      - void LL_TIM_CC_EnableChannel (TIM_TypeDef * TIMx, uint32_t Channels)
//      - void LL_TIM_OC_SetCompareCH1 (TIM_TypeDef * TIMx, uint32_t CompareValue)
//      - uint32_t LL_TIM_OC_GetCompareCH1 (TIM_TypeDef * TIMx)
```

Priprava – uporaba LCD zaslona z ILI9341

Na najnižjem osnovnem nivoju je izris grafike na LCD zaslon izveden s pomočjo čipa ILI9341, ki je integriran v modul LCD zaslona in opravlja vlogo krmilnika zaslona. Komunikacija med mikrokrmilnikom STM32G4 in LCD zaslonom (tj. čipom ILI9341) je izvedena s pomočjo posebnega vmesnika, ki se s tujko imenuje *"Flexible Memory Controller"* (FMC). *Vmesnik FMC* je primarno namenjen priklopu zunanjega pomnilnika (SRAM, flash, PSRAM, FRAM ipd.), vendar ga je mogoče uporabiti tudi pri prenosu podatkov na LCD zaslon (če pomislite: v LCD zaslonu se pravzaprav nahaja pomnilnik, ki hrani informacijo o barvah pikslov na zaslonu).

FMC vmesnik omogoča *paralelen prenos podatkov* z mikrokrmilnika na LCD zaslon. *Ključni signali FMC vmesnika* so v našem primeru sledeči (rdeči del na sliki spodaj):

- podatkovni signali `FMC_D[0..15]`,
- signal za izbiro naprave `CS` (angl. Chip Select, pogosto označen tudi z `NE`, "Negated Enable") , s katerim se zagotovi, da komunikacija poteka z le eno napravo na deljenem podatkovnem vodilu. FMC vmesnik namreč *omogoča priklop štirih naprav na isto podatkovno vodilo*.
- Signal za izbiro registra `RS` (angl. Register Select), s katerim sporočimo, ali želimo delati s podatkovnim registrom ali pa s kontrolnim registrom čipa ILI9341



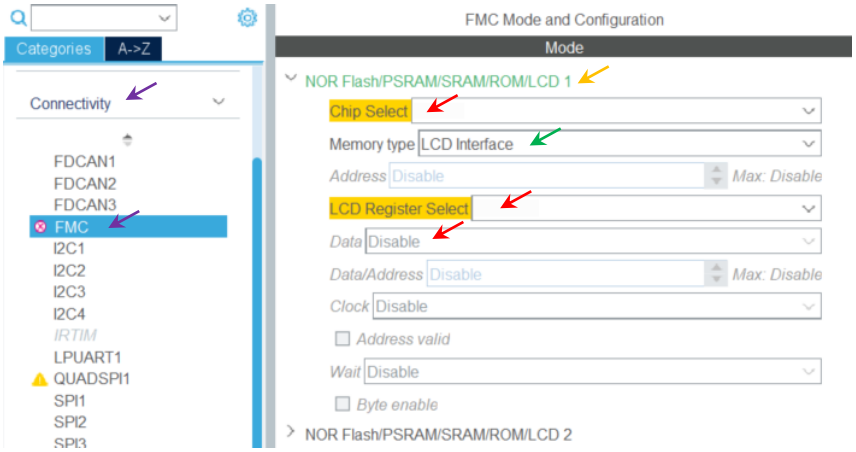
Za delo z LCD zaslonom pa FMC vmesnik potrebuje *še dodaten signal, ki ga implementiramo s pomočjo običajnega GPIO digitalnega izhoda* (zeleni del na sliki zgoraj):

- signal za resetiranje LCD zaslona `RST`.

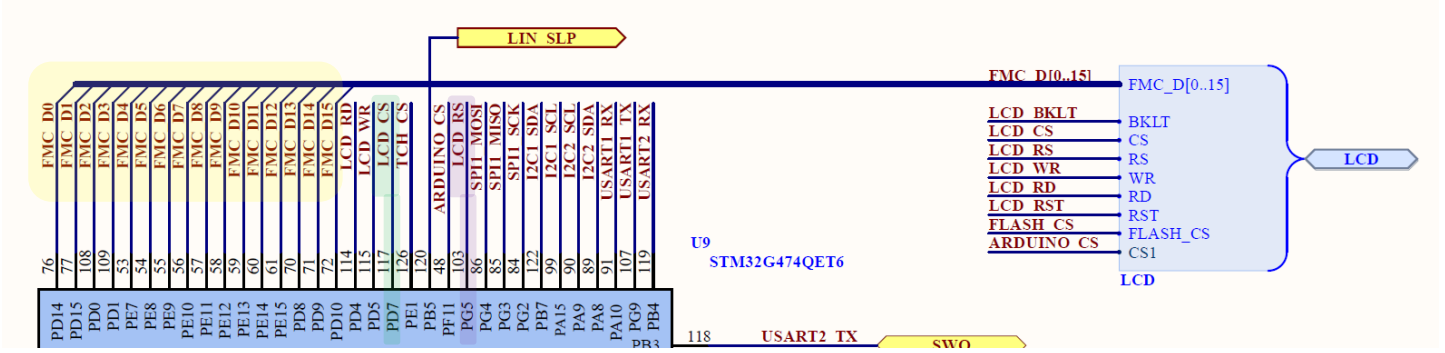
Vaša naloga bo, da v sklopu priprave poskrbite za ustrezno inicializacijo FMC vmesnika ter GPIO digitalnega izhoda.

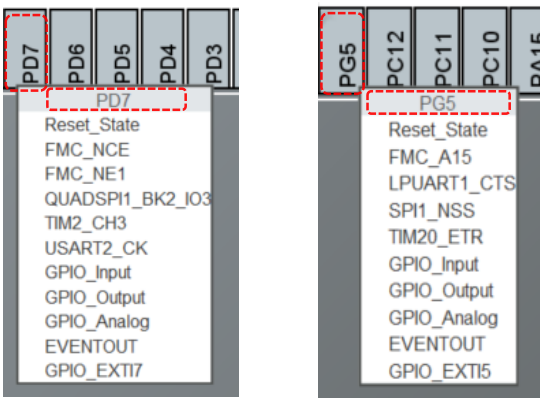
9. ***Vklopite FMC vmesnik za <u>1. napravo</u> in ustrezno nastavite*** "chip select" signal, "register select" signal ter širino podatkovnega vodila (rdeči del na sliki spodaj).

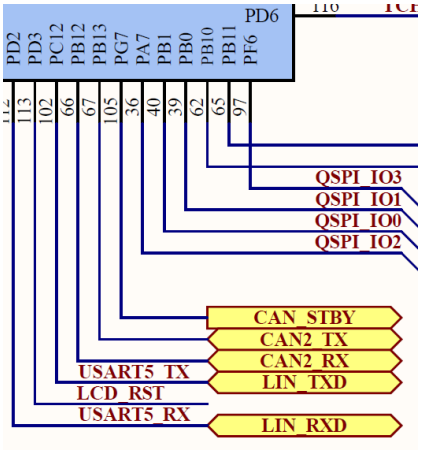
   FMC vmesnik vklopite tako, da specificirate tip pomnilnika ("memory type"), ki je v našem primeru kar "LCD interface" (zelena pušcica spodaj).



Vse potrebne informacije o signalih FMC vmesnika najdete na slikah spodaj.

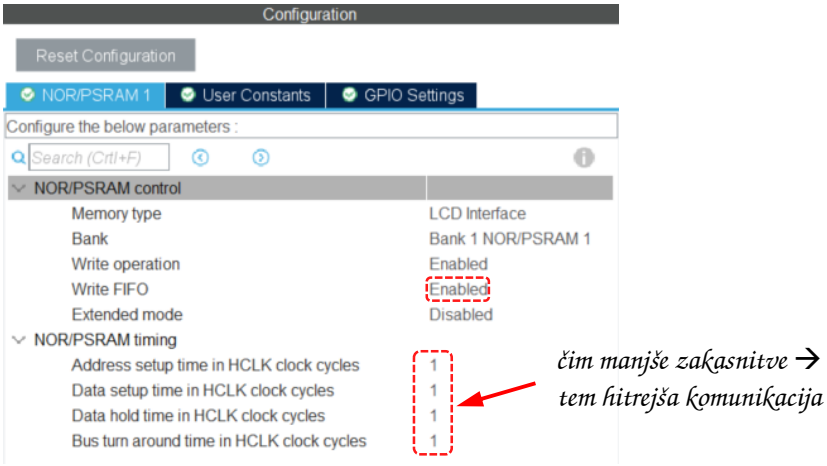






Slika 1 - vse možne funkcije pinov mikrokrmilnika lahko hitro preverimo v orodju CubeMX

10. Nastavite parametre FMC vmesnika za 1. napravo tako, kot prikazuje slika spodaj.



Ko boste shranili nastavitve v CubeMX, boste lahko sprožili avtomatsko generiranje kode za inicializacijo FMC vmesnika, kar pa bo povzročilo sledečo spremembo:

```
/* Initialize all configured peripherals */
MX_GPIO_Init();
MX_USART3_UART_Init();
MX_TIM6_Init();
MX_TIM4_Init();
MX_FMC_Init();
/* USER CODE BEGIN 2 */
```

11. Poskrbite, da bo ustrezen pin mikrokrmilnika nastavljen kot GPIO digitalni izhod, s katerim bomo implementirali "LCD reset" signal.

Izhod naj bo simetričnega tipa ("push-pull").

Lokacijo "LCD reset" pina lahko najdete na izseku iz sheme vezja na sliki z zgoraj.

12. Projektu dodajte vse potrebne datoteke za delo z LCD zaslonom.

Datoteke najdete v mapi "predloge\LCD". Dodajte ju v projektno mapo znotraj podmape "System" na ustrezno mesto.

Priprava – grafična knjižnica uGUI

Preletite izsek z uradne strani knjižnice, da dobite občutek, kaj sploh je knjižnica `uGUI` (izgovarjamo "micro GUI", kjer je GUI okrajšava za "Graphical User Interface").

What is µGUI

µGUI is a free and open source graphic library for embedded systems. It is platform-independent and can be easily ported to almost any microcontroller system. As long as the display is capable of showing graphics, µGUI is not restricted to a certain display technology. Therefore, display technologies such as LCD, TFT, E-Paper, LED or OLED are supported. The whole module consists of two files: **ugui.c** and **ugui.h**.

µGUI Features

- µGUI supports any color, grayscale or monochrome display
- µGUI supports any display resolution
- µGUI supports multiple different displays
- µGUI supports any touch screen technology (e.g. AR, PCAP)
- µGUI supports windows and objects (e.g. button, textbox)
- 16 different fonts available
- integrated and free scalable system console
- basic geometric functions (e.g. line, circle, frame etc.)
- can be easily ported to almost any microcontroller system
- no risky dynamic memory allocation required

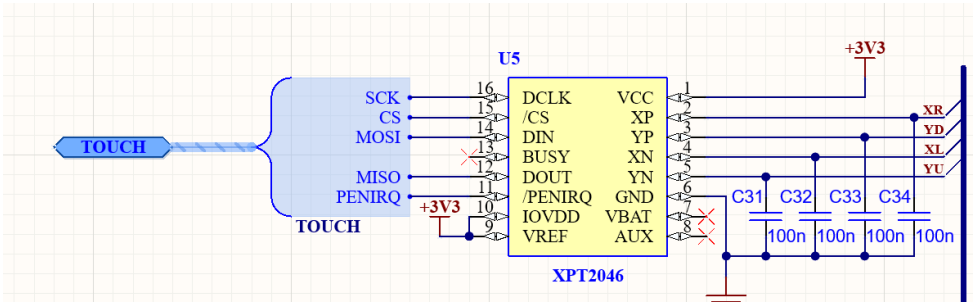
13. ***Projektu dodajte vse potrebne datoteke za s knjižnico*** `uGUI`**.**

    Datoteke najdete v mapi "predloge\uGUI". Dodajte ju v projektno mapo znotraj podmape "System" na ustrezno mesto.

Priprava – modul za detekcijo pritiska na zaslon

Čez LCD zaslon je povezanjena rezistivni folija, s pomočjo katere lahko zaznamo lokacijo pritiska na zaslon. Detekcija lokacije pritiska je olajšana s pomočjo čipa XPT2046, ki analogne signale s folije (tj. `XR`, `YD`, `XL`, `YU` na spodnji sliki) preračuna v lokacijo pritiska in jo nato mikrokrmilnik lahko prebere preko serijskega vodila SPI (`MISO`,`MOSI`, `SCK` na spodnji sliki). Čip za delovanje potrebuje še dva digitalna signala:

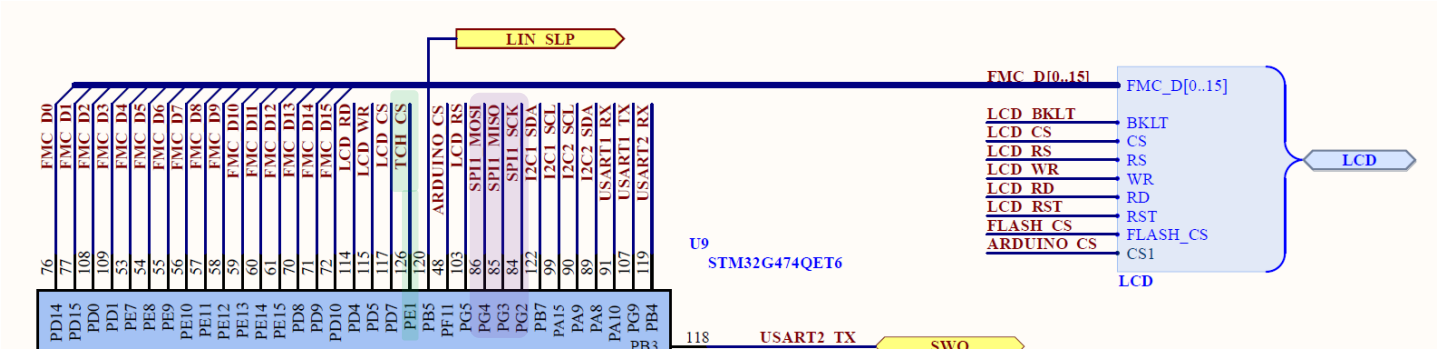
- "chip select" signal `CS`, s katerim mikrokrmilnik izbere napravo za komunikacijo na SPI vodilu,
- "pending interrupt" `PENIRQ`, s katerim čip sporoča mikrokrmilniku, da je zaznal pritisk na zaslon.



Vaša naloga v sklopu priprave bo, da poskrbite za ustrezno nizko-nivojsko inicializacijo SPI vmesnika ter GPIO digitalni vhod in izhod.

14. ***V orodju CubeMX nastavite ustrezne pine mikrokrmilnika tako, da bodo opravljali funkcijo SPI vmesnika***, ki je uporabljen za komunikacijo s čipom XPT2046.

    Vse potrebne informacije o signalnih povezavah in pinih lahko najdete na spodnjih slikah.



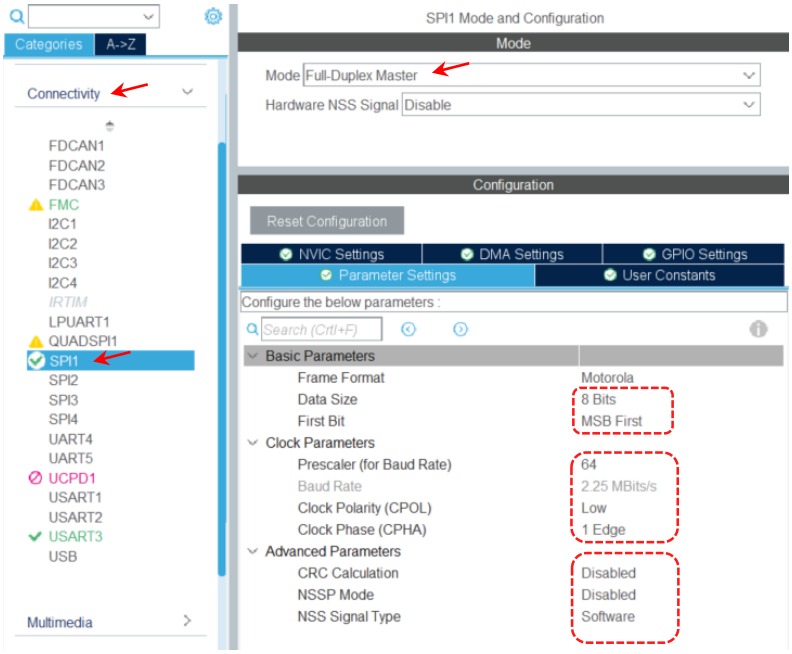
15. ***V orodju CubeMX poskrbite, da bosta ustrezna pin mikrokrmilnika nastavljena kot GPIO digitalni izhod oziroma vhod***, s katerima bomo implementirali "chip select" signal in pa "interrupt pending" signal.

    Izhod naj bo simetričnega tipa ("push-pull").

    Lokacijo obeh pinov lahko najdete na izseku iz sheme vezja na slikah z zgoraj.

16. ***V orodju CubeMX omogočite in pravilno nastavite serijski vmesnik*** `SPI1`**.**

    Pomagajte si s sliko spodaj.



    Zagotoviti moramo, da je hitrost komunikacije s čipom XPT2046 nižja od 2,5 Mbit/s.

    Ko se bo zgenerirala avtomatska koda, bo poskrbljeno za inicializacijo `SPI1` vmesnika.

```
/* Initialize all configured peripherals */
MX_GPIO_Init();
MX_USART3_UART_Init();
MX_TIM6_Init();
MX_TIM4_Init();
MX_FMC_Init();
MX_SPI1_Init();
/* USER CODE BEGIN 2 */
```

17. ***Projektu dodajte vse potrebne datoteke za delo z modulom za detekcijo pritiska na zaslon.***

    Datoteke najdete v mapi "predloge\touch". Dodajte ju v projektno mapo znotraj podmape "System" na ustrezno mesto.