



УНИВЕРЗИТЕТ У НОВОМ САДУ  
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА У НОВОМ  
САДУ



---

# ПРОЈЕКАТ

из предмета примена електронике у индустрији

## ТЕМА ПРОЈЕКТА:

Пример SPI комуникационог протокола за микроконтролер PIC24FJ26GA106

Студент:

Јован Славујевић

Ментор:

проф. др Рајс Владимир

Нови Сад, 2020. год.

## Садржај

1 <i>SPI</i> протокол.....	3
1.1 Настанак и примена.....	3
1.2 Карактеристике и пинови.....	3
1.3 Поларитет, фаза и модови.....	4
2 <i>SPI</i> за <i>PIC24</i> .....	5
3 Референце .....	6

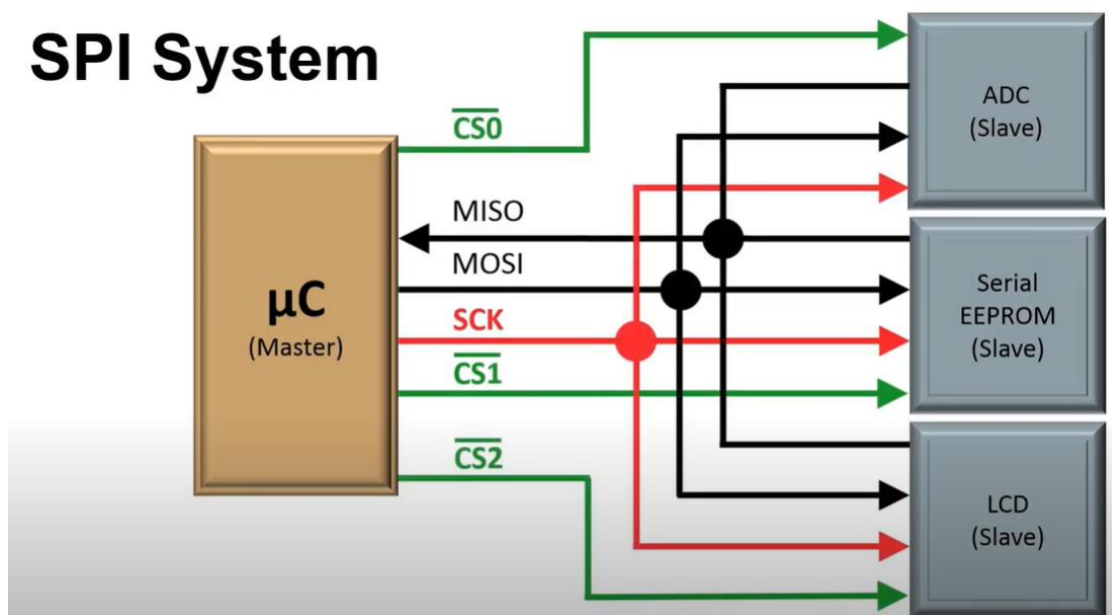
# 1 SPI протокол

## 1.1 Настанак и примена

*SPI* (енг. *Serial Peripheral Interface*) је акроним за Серијски Периферни Интерфејс и представља један од основних синхроних комуникационих протокола. Настао је 80-их година прошлог века, првобитно развијен од компаније *Motorola* као потреба за брзим комуникационим протоколом. Овај протокол има широку примену у самосталним меморијским уређајима као што су *NOR Flash (Flash EEPROM)*, серијски *EEPROM* (енг. *Electrically Erasable Programmable Read-Only Memory*), *EERAM* (енг. *Electrically Erasable Random Access Memory*), серијски *EERAM (SRAM)*, меморијским картицама, *DAC* и *ADC* конвертерима, *LCD* (енг. *Liquid Crystal Display*) екранима итд.

## 1.2 Карактеристике и пинови

За разлику од тада развијеног *I<sup>2</sup>C* (енг. *Inter-Integrated Circuit*) протокола који је имао брзину такта реда неколико *kHz* (данас је досегао и до *1 MHz*) *SPI* одликује знатно већа брзина – у нормалним условима има *20 MHz*, али додатним унапређењем (*Dual/Quad SPI*) може да досегне и до *144 MHz*. Овако важна одлика овог протокола је базирана је на томе да размена података иде на два комуникациона канала, *MISO* (енг. *Master Input Slave Output*) и *MOSI* (енг. *Master Output Slave Input*) за разлику од *I<sup>2</sup>C* који размењује податке преко једног пина – *SDA* (енг. *Serial Data Access*). Осим ова два пина, *SPI* као и сваки други синхорни серијски протокол захтева генерисање такт сигнала те постоји и трећи пин – *SCK* (енг. *Serial Clock*), што је пандан *SCL* (енг. *Serial Clock Line*) *I<sup>2</sup>C* пину. *SPI* је протокол где је комуникација не мора стриктно да буде „1 на 1“, те је омогућено једном господару (енг. *Master*) да комуницира са више подређених (енг. *Slave*) истовремено (енг. *Multi-Slave*), али не постоји више господара (енг. *Multi-Master*). Препознавање и одабир са којим подређеним уређајем или периферијом господар говори омогућено је уз помоћ четвртог пина – *CS* (енг. *Chip Select*). То значи да је у ситуацији „1 на 1“ неопходно 4 пина за комуникацију, а за комуникацију господара са *n* уређаја неопходно је *n* *CS* пинова за сваки од њих.



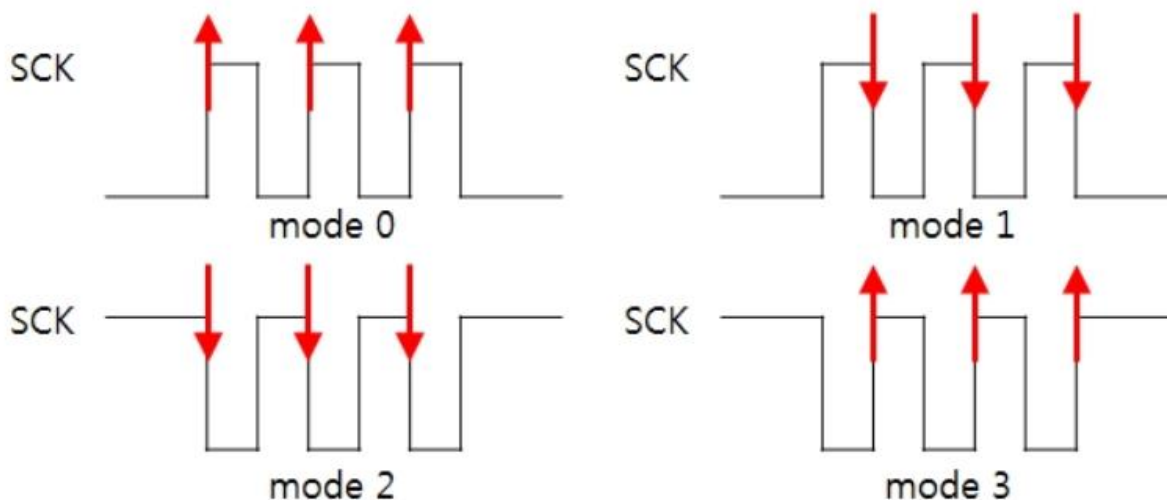
Слика 1. Пример *SPI* комуникације једног господара са 3 уређаја (подређена)

Дакле, *SPI* одликује знатно већа брзина (минимално 20 пута) преко већег броја ожичења што је само по себи негативна ставка, а та одлика га чини и скупљим за израду. Одликује га *Multi-Slave*, али не и *Multi-Master* могућност комуницирања. С' обзиром на то да ради на већим учестаностима овај протокол је осетљивији на шуме, али црпи мање енергије у односу на *I<sup>2</sup>C*. Такође, једна од значајних негативних ставки не постоји механизам потврде о пријему података (енг. *Acknowledgment*).

### 1.3 Поларитет, фаза и модови

Управљачка *SPI* логика господара поседује два параметра – поларитет клона (*CPOL*) и поларитет фазе (*CPHA*), а помоћу њих се дефинишу четири мода рада (*Mode 0*, *Mode 1*, *Mode 2*, *Mode 3*). На тај начин, омогућено је да господар постави поларитет и фазу такта, како би се омогућила велика флексибилност за комуникацију између господара и подређеног. Уз помоћ ова два параметра господар одређује који подаци морају бити стабилни, када треба да се мењају у складу са такт сигналом (линијом клона).

Поларитет клона (*CPOL*) додељује ниво клона када клон није активан. За случај када је поларитет фазе вредности нула (*CPOL* = 0) такт сигнал (*SCK*) је неинвертован, а за вредност један такт сигнал биће инвертован и тада се реагује на опадајућу ивицу такт сигнала. Поларитет фазе (*CPHA*) се користи за померање фазе хватања. За његову вредност нула (*CPHA* = 0) подаци се снимају на првој ивици клона, без обзира да ли је у питању растућа или опадајућа ивица. За вредност јединице (*CPHA* = 1) подаци се снимају на другој ивици клона.



Слика 2. Могући модови у *SPI* протоколу

## 2 SPI за PIC24

У овом поглављу приказан је начин имплементације *SPI* комуникације за конкретан микроконтролер. У питању је *PIC24FJ26GA106* који се налази на *EB800*.

```
1 void SPI_Init(void)
2 {
3     SPI1STATbits.SPIEN=1;           //ENABLE
4     SPI1STATbits.SPISDL=0;         //CONTINUES MODE IN IDLE MODE
5     SPI1STATbits.SPIROV=0;         //NO OVERFLOW OCCURED
6     SPI1STATbits.SPITBUF=0;        //TRANSMIT STARTED,SPI1TXB IS EMPTY
7     SPI1STATbits.SPIRBUF=0;        //RECEIVE IS NOT COMPLETED,SPI1RXB IS EMPTY
8
9     SPI1CON1bits.DISSCK=0;          //INTERNAL SPI CLOCK ENABLEOVAN
10    SPI1CON1bits.DISSDO=0;          //SDO KONTROLISE MODUL
11    SPI1CON1bits.MODE16=0;          //KOMUNIKACIJA BAJTNA
12    SPI1CON1bits.SMP=0;             //SAMPLE NA SREDINI DATA OUTPUT TIME-A
13    SPI1CON1bits.CKE=0;             //PROMENA PODATAKA SA IDLE KLOK STATE NA ACTIVE KLOK STATE
14    SPI1CON1bits.CKP=1;             //POLARITET KLOKA 1 JE IDLE STATE
15    SPI1CON1bits.MSTEN=1;           //MASTER MODE
16    SPI1CON1bits.SPRE=7;            //SEKUNDARNI PRESKALER 1
17    SPI1CON1bits.PPRE=3;            //PRIMARNI PRESKALER 1
18
19    __builtin_write_OSCCONL(OSCCON & 0xBF);
20
21    RPOR10bits.RP20R=7;             //MOSI---->OUTPUT--->RP20
22    RPINR20bits.SDI1R=20;           //MISO---->INPUT---->RP25
23    RPOR11bits.RP22R=8;             //CLK----->OUTPUT--->RP22
24    RPOR11bits.RP23R=9;             //SS(CS)--->OUTPUT--->RP23
25
26    __builtin_write_OSCCONL(OSCCON | 0x40);
27
28    TRISDbits.TRISD5=0; //MOSI
29    TRISDbits.TRISD4=1; //MISO
30    TRISDbits.TRISD3=0; //SCK
31    TRISDbits.TRISD2=0; //SS
32
33    SPI1BUF=0;
34 }
```

Слика 3. Пример иницијализације микроконтролера као господара

```
unsigned char SPI_READ_WRITE(unsigned char data)
{
    SPI1BUF = data;
    while(!SPI1STATbits.SPIRBUF);
    return SPI1BUF;
}
```

Слика 4. Пример уписа и пријема података

Наиме, у овом случају микроконтролер игра улогу господара, за случај да он буде подређени неком другом господару првенствено би *SPI1CON1bits.MSTEN* био постављен на нулу (енг. *Slave Mode*) а *CS* би био проглашен за улазни уместо излазног пина.

### 3 Референце

- [1] SPI Komunikacija.pptx – Презентација са предавања са предмета „Примена Електронике у Индустији“, проф. Владимир Рајс, приступљено: април 2020.
- [2] [https://en.wikipedia.org/wiki/Serial\\_Peripheral\\_Interface](https://en.wikipedia.org/wiki/Serial_Peripheral_Interface) - “*Serial Peripheral Interface*”, *Wikipedia* чланак, приступљено: април 2020.
- [3] <https://www.youtube.com/watch?v=NyxQkGXbG6I&t=182s> – “*What is...SPI?*”, *Microchip Technology* званични *YouTube* канал, приступљено: април 2020.
- [4] “*PIC24FJ256GA110 Family Data Sheet DS30009905F*”, *Microchip Technology* техничка документација за микроконтролер *PIC24FJ256GA110*, приступљено: април 2020