PRIMENA MIKROKONTROLERA-MS1PMK

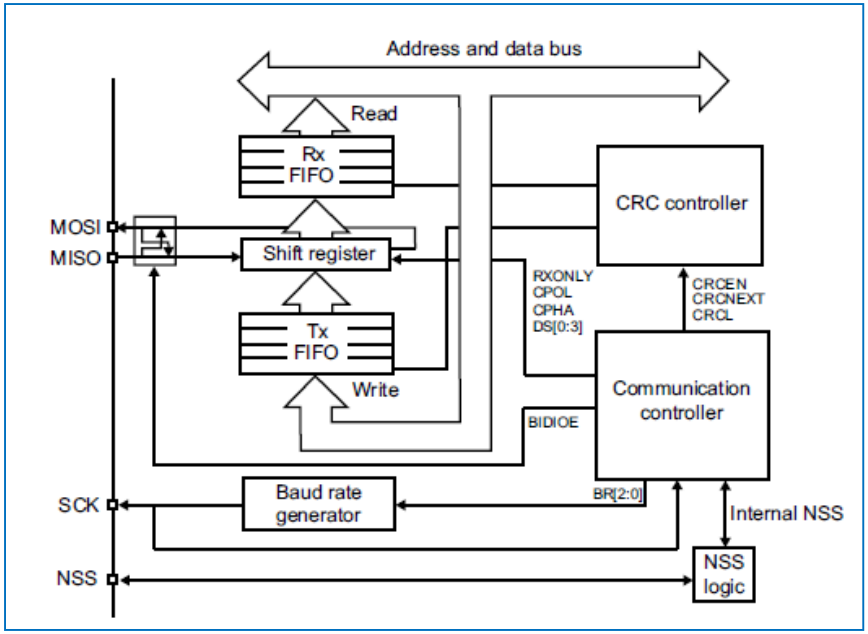
8. deo

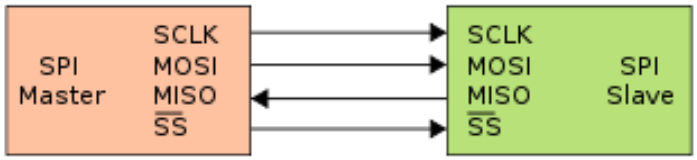
2018

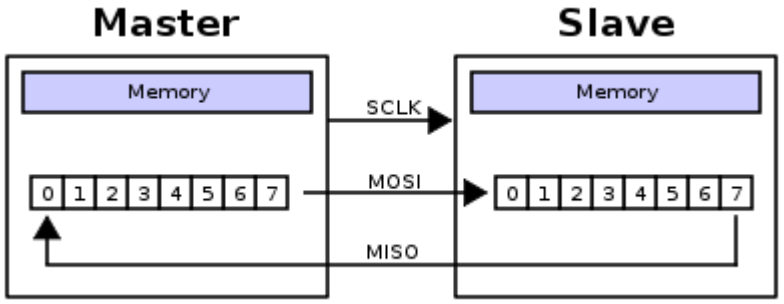
Nenad Jovičić

STM32L476 SPI modul

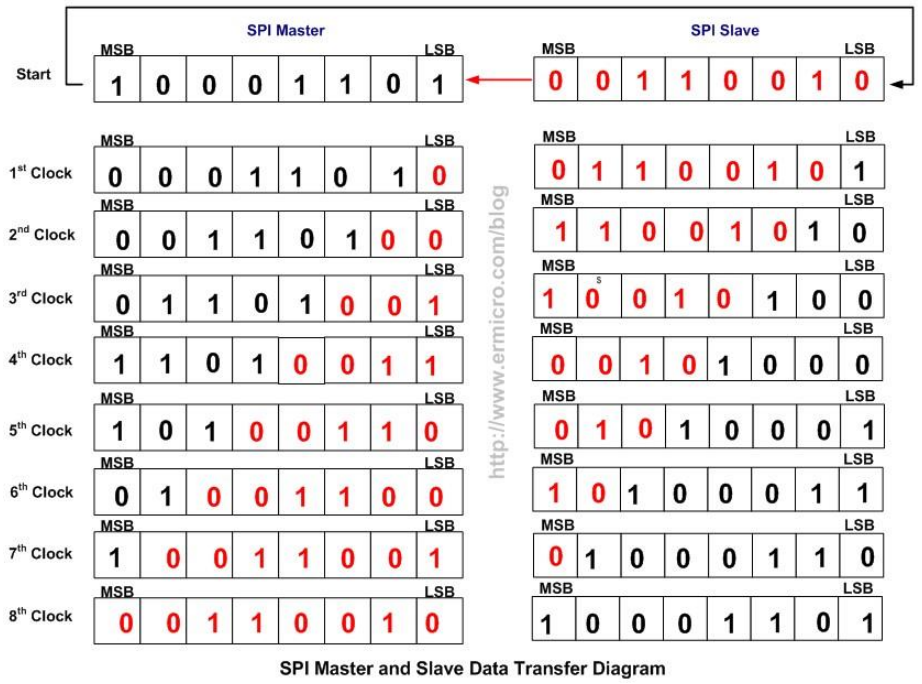
- **Osnovne karakteristike**
  - Kompatibilan sa SPI specifikacijom za pun dupleks i poludupleks
  - SPI master ili slave mod
  - 4-16 bita po prenosu
  - 32bitni baferi i za predaju i za prijem
  - Signaliziranje grešaka







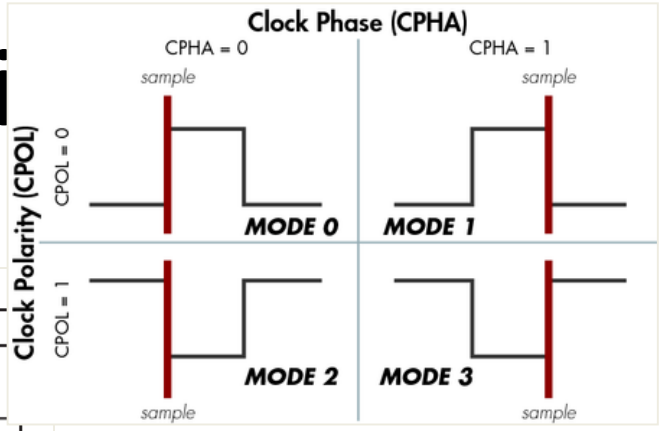
SPI opis rada

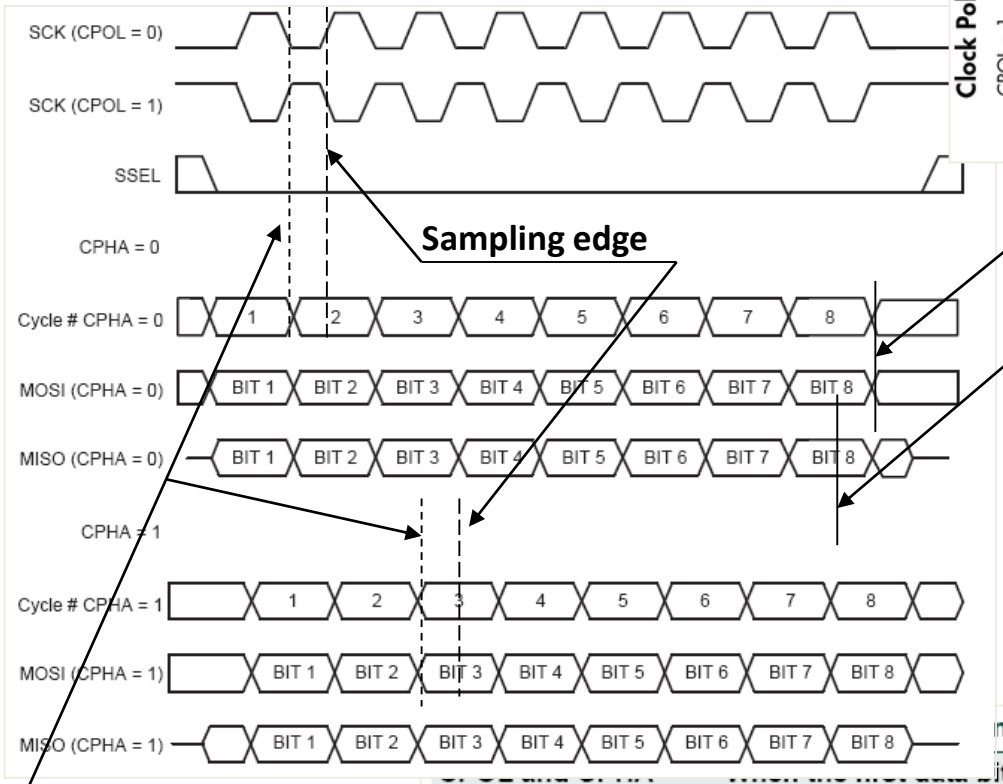


SPI Master and Slave Data Transfer Diagram

- **Master mod rada**
  - Inicijalizacija registara SPI modula
  - Upis podatka u SPI data registar što inicira slanje podatka
  - Čekanje na kraj slanja
  - Čitanje statusa i primljenog podatka u SPI data registru.
- **Slave mod rada**
  - Inicijalizacija registara SPI modula
  - Upis podatka u SPI data registar (opciono) što briše SPIF status bit
  - Čekanje na signal prijema podatka
  - Čitanje statusa i primljenog podatka

SPI signali





Sampling edge

Transfer complete (SPIF=1) master

Transfer complete
(prekidni fleg SPIF=1 na slave strani)

Driving edge

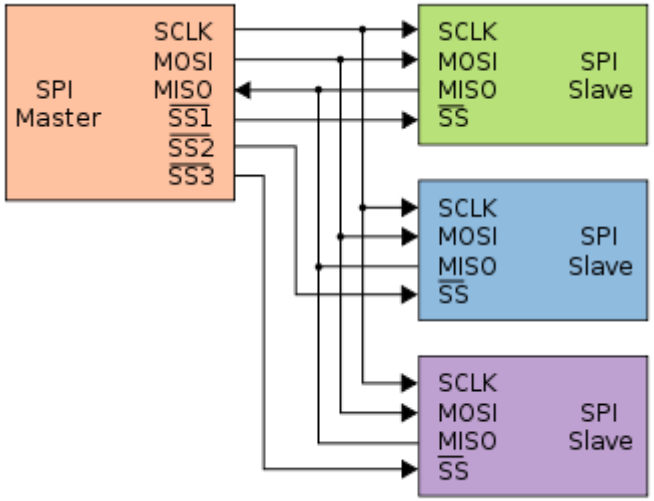
nship

settings	driven	When all other data bits are driven	When data is sampled
CPOL = 0, CPHA = 0	Prior to first SCK rising edge	SCK falling edge	SCK rising edge
CPOL = 0, CPHA = 1	First SCK rising edge	SCK rising edge	SCK falling edge
CPOL = 1, CPHA = 0	Prior to first SCK falling edge	SCK rising edge	SCK falling edge
CPOL = 1, CPHA = 1	First SCK falling edge	SCK falling edge	SCK rising edge

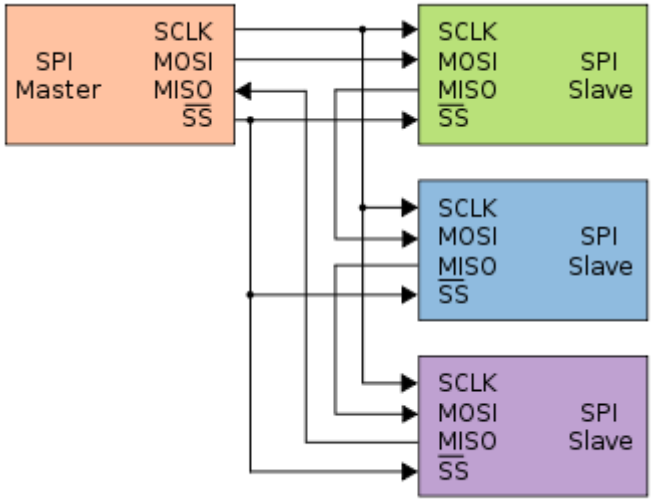
CPHA – kontrola faze, CPOL – kontrola polariteta taktnog signala

Povezivanje više Slave-ova

Greedy



Daisy Chain



MBED SPI Master

http://mbed.org/handbook/SPI

Public Member Functions

SPI (PinName mosi, PinName miso, PinName sclk, PinName ssel=NC)
Create a **SPI** master connected to the specified pins.

void format (int bits, int mode=0)
Configure the data transmission format.

void frequency (int hz=1000000)
Set the spi bus clock frequency.

virtual int write (int value)
Write to the **SPI** Slave and return the response.

template<typename Type >
int transfer (const Type *tx_buffer, int tx_length, Type *rx_buffer, int rx_lengt
Start non-blocking **SPI** transfer using 8bit buffers.

void abort_transfer ()
Abort the on-going **SPI** transfer, and continue with transfer's in the queue

void clear_transfer_buffer ()
Clear the transaction buffer.

void abort_all_transfers ()
Clear the transaction buffer and abort on-going transfer.

int set_dma_usage (DMAUsage usage)
Configure DMA usage suggestion for non-blocking transfers.

Protected Member Functions

void irq_handler_asynch (void)
SPI IRQ handler.

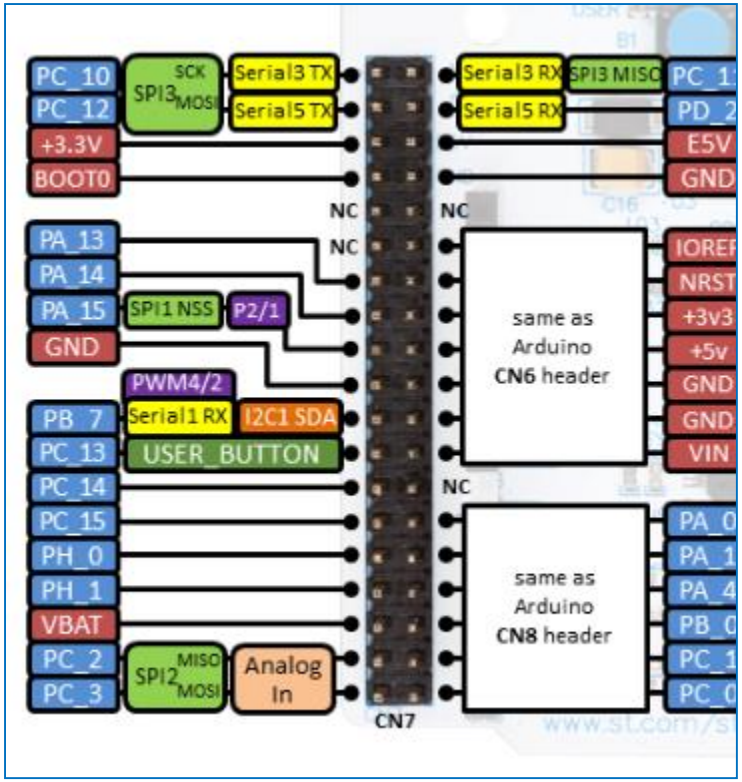
int transfer (const void *tx_buffer, int tx_length, void *rx_buffer, int rx_length
Common transfer method.

int queue_transfer (const void *tx_buffer, int tx_length, void *rx_buffer, int rx

void start_transfer (const void *tx_buffer, int tx_length, void *rx_buffer, int rx_
Configures a callback, spi peripheral and initiate a new transfer.

void start_transaction (transaction_t *data)
Start a new transaction.

void dequeue_transaction ()
Dequeue a transaction.



Klasa SPISlave

http://mbed.org/handbook/SPISlave

Public Member Functions

SPISlave (PinName mosi, PinName miso, PinName sclk, PinName ssel)
Create a SPI slave connected to the specified pins.

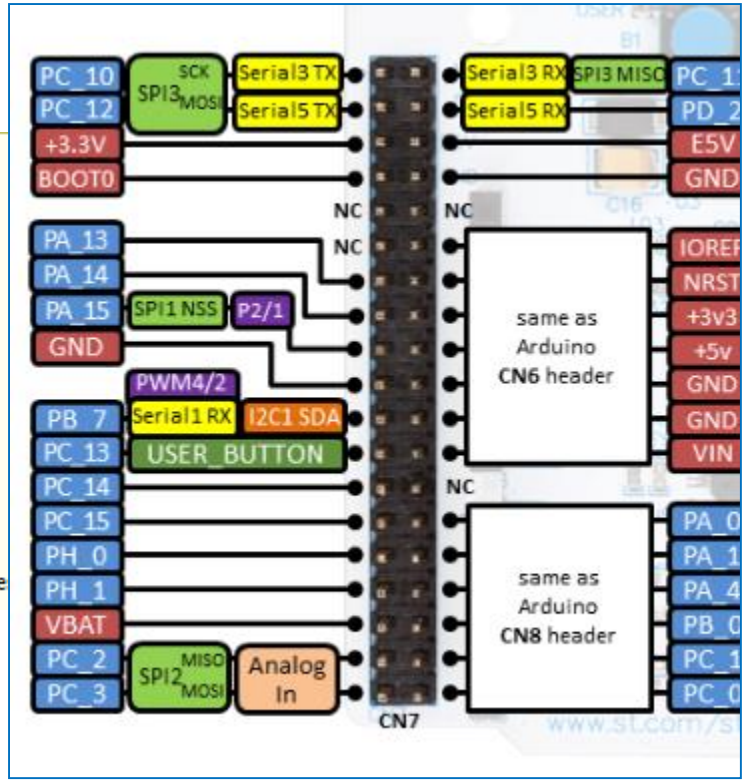
void format (int bits, int mode=0)
Configure the data transmission format.

void frequency (int hz=1000000)
Set the spi bus clock frequency.

int receive (void)
Polls the SPI to see if data has been received.

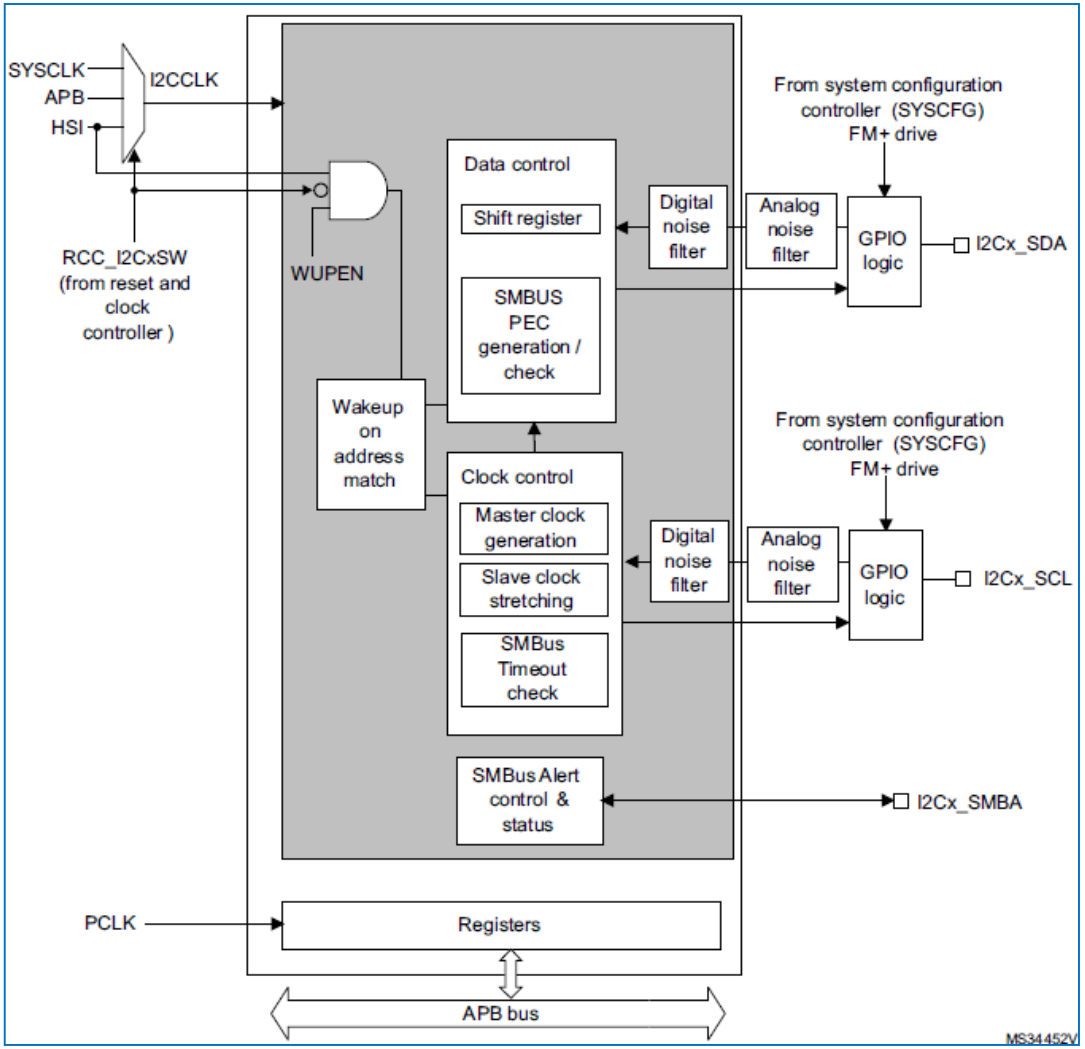
int read (void)
Retrieve data from receive buffer as slave.

void reply (int value)
Fill the transmission buffer with the value to be written out as slave on the next received me



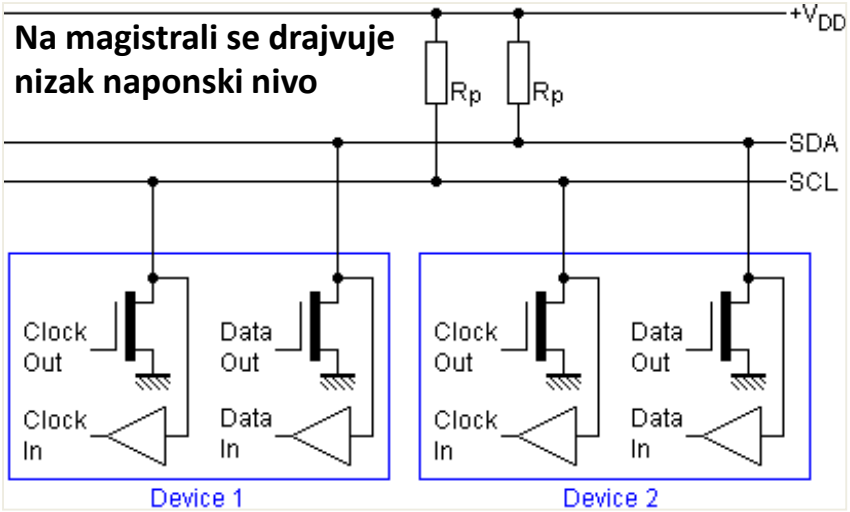
STM32L476 I²C modul

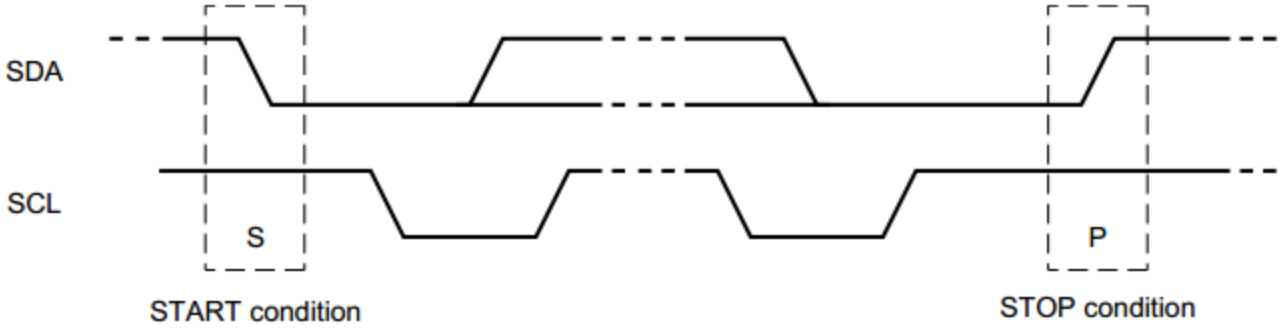
- Podržani master i slave
- Arbitracija pristupa u multimaster modu rada
- Programabilni takt (100 kHz, 400 kHz, 1 MHz)
- Podrška za prenos do 1Mb/s (fast plus mod)
- Bidirekcioni prenos podataka
- 7-bitno i 10-bitno adresiranje
- Podrška za 2 slave adrese



I2C protokol

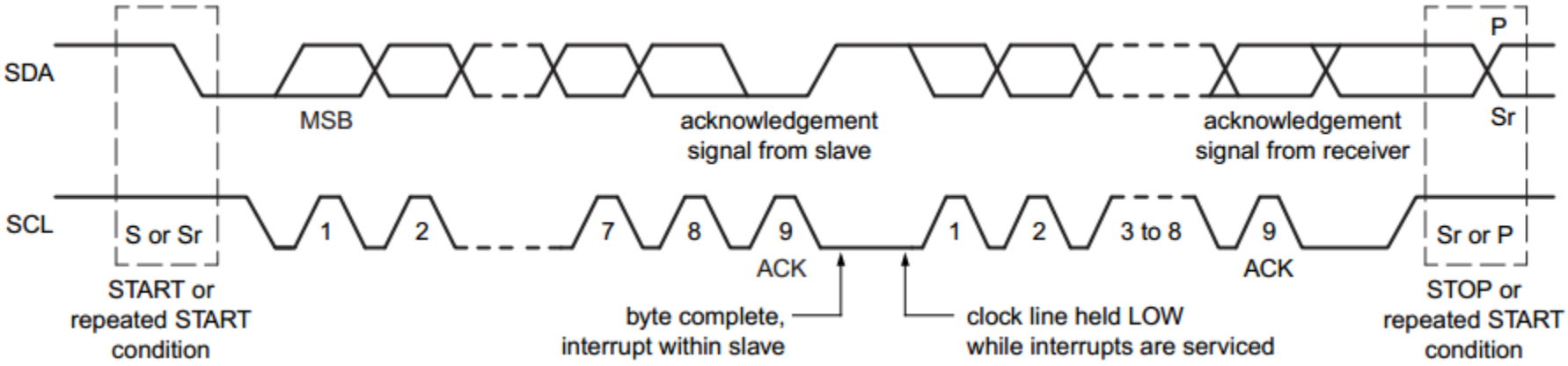
- Svaki transfer započinje i završava se takozvanim START i STOP zaglavljima.
- Ova zaglavlja definiše tj. drajvuje master.
- Master diktira takt na magistrali.
- Ukoliko Slave iz nekog razloga ne može da primi podatak, zadržavanjem linije takta na niskom nivou master se stavlja na čekanje.



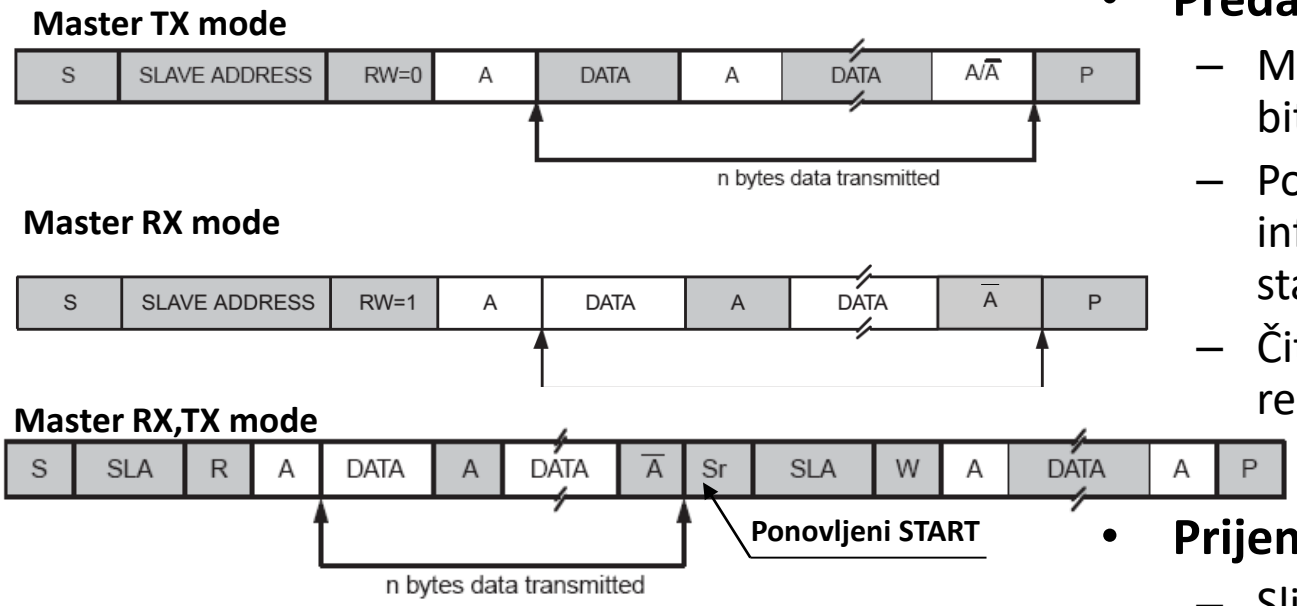


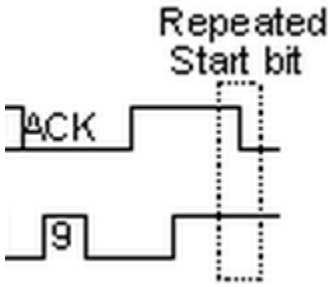
ACK i NACK

- Uvek se prenosi podatak koji ima 8-bita, i nakon svakog transfera podatka dodaje se još jedan bit za potvrdu koji može biti ACK i NACK.
- Prijemnik izdaje ACK posle svakog korektno primljenog podatka, time što prilikom devetog taktnog impulsa na SCL liniji SDA liniju drži nisko.
- NACK se generiše tako što se prilikom devetog SCL impulsa SDA linija ne drajvuje i može da znači da:
  - Ne postoji slave na liniji.
  - Slave nije mogao da primi podatke
  - Slave ne razume podatke
  - Slave ne može da nastavi sa primanje podataka.
  - Master u slučaju da je prijemnik obaveštava Slave da je završeno sa prijemom.



I2C frejm

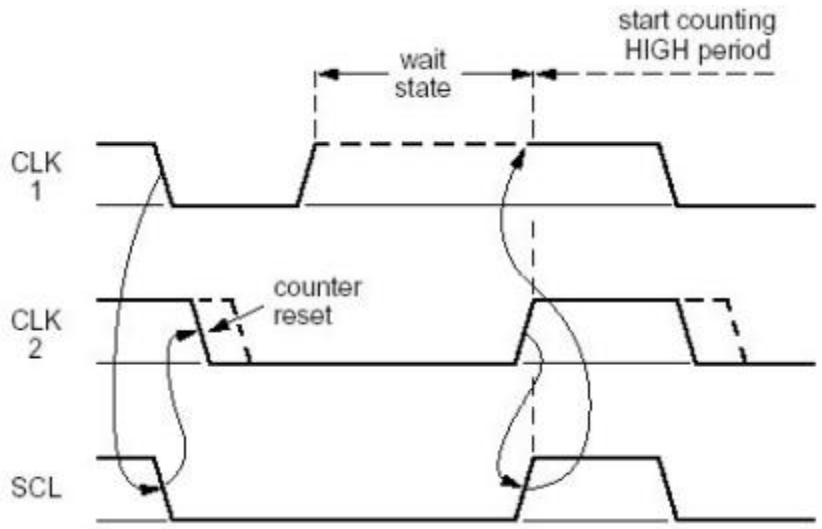




- **Predajni mod**
  - Master šalje slave adresu (7 bita) i R/W bit (0)
  - Po prijemu svake 8-bitne informacije slave generiše stanje potvrde prijema (A)
  - Čita se sadržaj statusnog registra I2CSTAT
- **Prijemni mod**
  - Slično kao predajni mod osim što je R/W bit (1)
- **Predajno/prijemni mod**
  - Koristi se stanje ponovljeni START da se inicira novi ciklus komunikacije

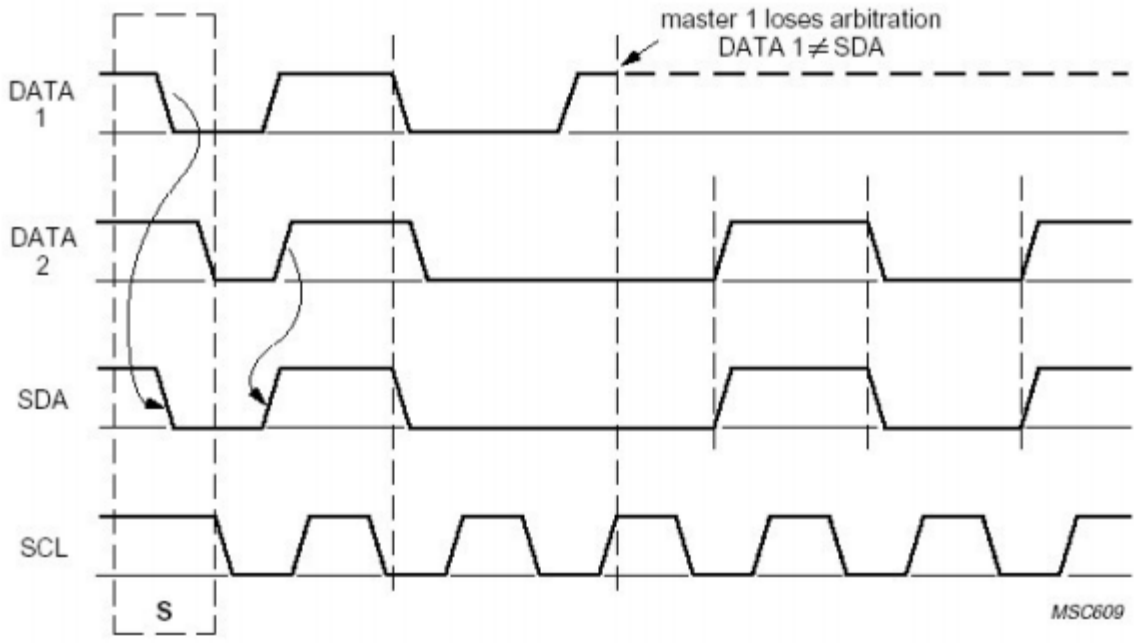
Multimaster – sinhronizacija takta

- Svi uređaji rade na istom taktu.
- Kada master 1 obori SCL liniju na nizak nivo, master 2 sinhronizuje svoj interni brojač.
- Trajanje niskog nivoa na SCL liniji jednako je maksimalnom trajanju niskon nivoa takta svih mastera.
- Trajanje visokog nivoa na SCL liniji jednako je minimalnom trajanju visokog nivoa takta svih mastera.



Arbitracija

- Nakon sinhornizacije takta i START zaglavlja svaki master emituje svoju poruku sa podacima.
- Na rastuću ivicu takva svaki master proveraava da li je stanje na SDA liniji jednako onom koje je postavio.
- U slučaju da nije taj master gubi arbitraciju.



MBED I2C Master

http://mbed.org/handbook/I2C

Public Member Functions

I2C (PinName sda, PinName scl)
Create an **I2C** Master interface, connected to the specified pins.

void frequency (int hz)
Set the frequency of the **I2C** interface.

int read (int address, char *data, int length, bool repeated=false)
Read from an **I2C** slave.

int read (int ack)
Read a single byte from the **I2C** bus.

int write (int address, const char *data, int length, bool repeated=false)
Write to an **I2C** slave.

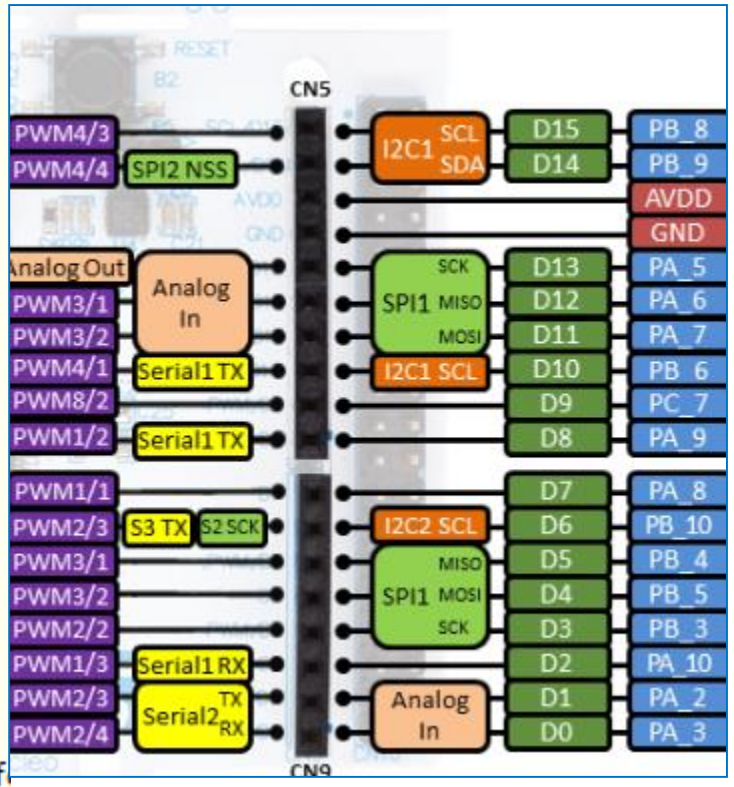
int write (int data)
Write single byte out on the **I2C** bus.

void start (void)
Creates a start condition on the **I2C** bus.

void stop (void)
Creates a stop condition on the **I2C** bus.

int transfer (int address, const char *tx_buffer, int tx_length, char *rx_buff
Start non-blocking **I2C** transfer.

void abort_transfer ()
Abort the on-going **I2C** transfer.



MBED I2C Slave

http://mbed.org/handbook/I2CSlave

Public Member Functions

I2CSlave (PinName sda, PinName scl)
Create an I2C Slave interface, connected to the specified pins.

void frequency (int hz)
Set the frequency of the I2C interface.

int receive (void)
Checks to see if this I2C Slave has been addressed.

int read (char *data, int length)
Read from an I2C master.

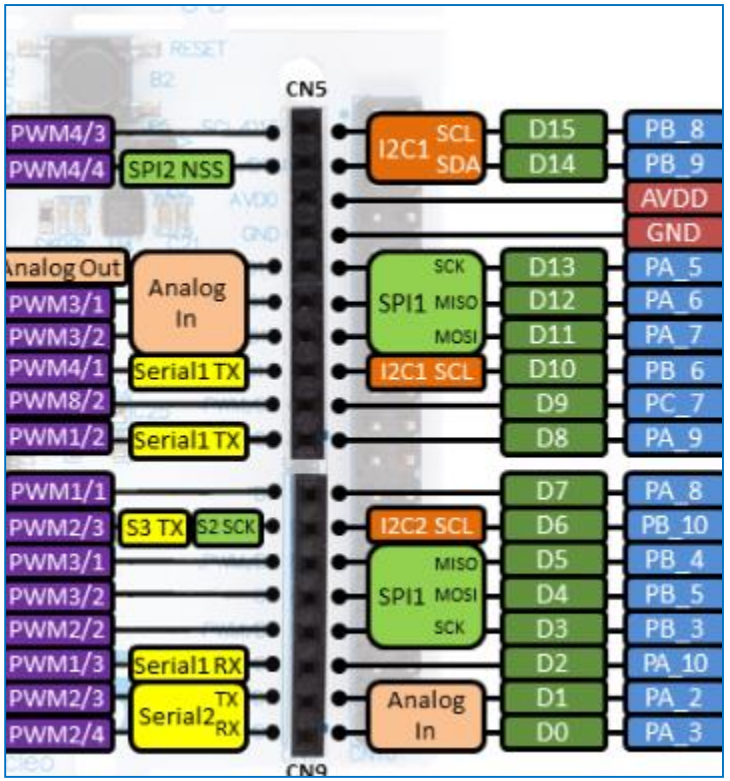
int read (void)
Read a single byte from an I2C master.

int write (const char *data, int length)
Write to an I2C master.

int write (int data)
Write a single byte to an I2C master.

void address (int address)
Sets the I2C slave address.

void stop (void)
Reset the I2C slave back into the known ready receiving state.



I2C SENZOR – Color click

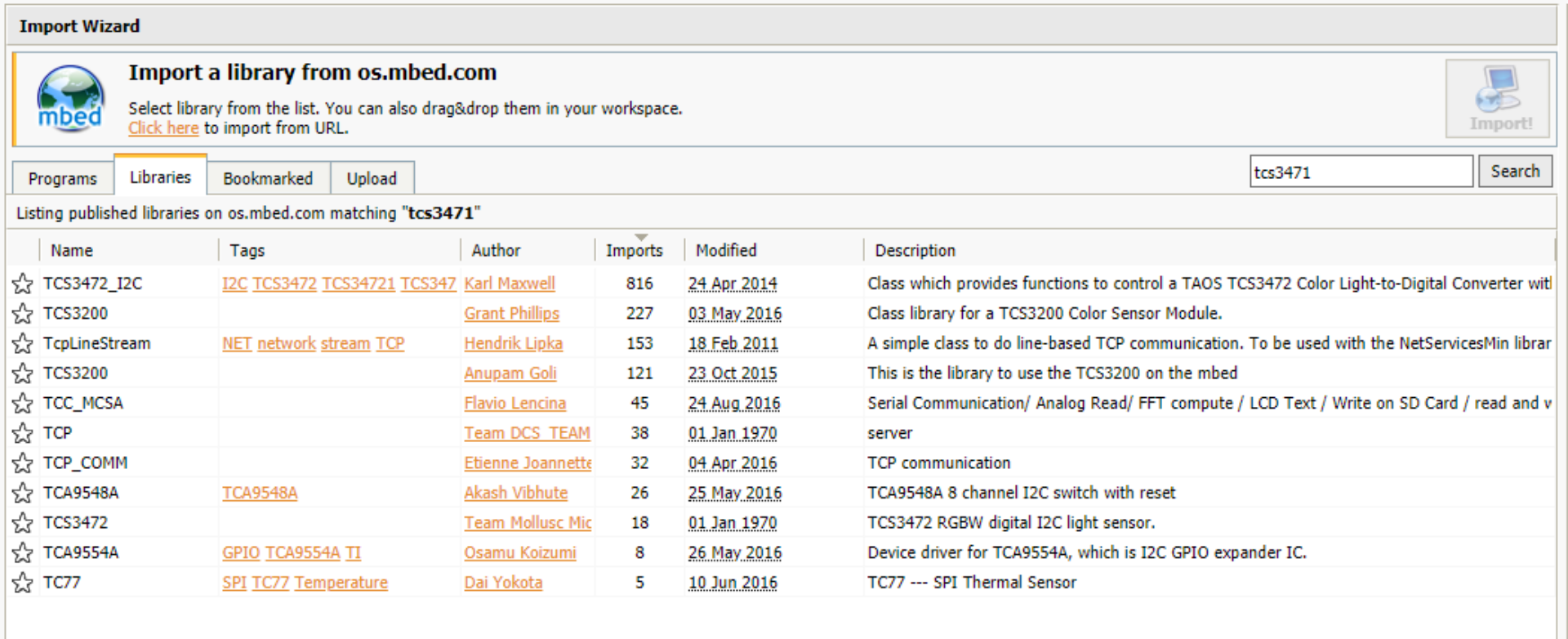
- Color Click sadrži TCS3471 senzor boja i osvetljaja
- Daje 16bitne podatke o CRGB
  - osvetljenosti (Clear)
  - crvenoj boji (Red)
  - zelenoj boji (Green)
  - plavoj boji (Blue)
- Programiranje čipa se radi upisom odgovarajućih komandi u registre čipa - datasheet

I2C SENZOR – Altitude click

- Altitude Click sadrži MPL3115A2 senzor nadmorske visine, temperature i vazdušnog pritiska
- Programiranje čipa se radi upisom odgovarajućih komandi u registre čipa - datasheet

MBED - Import

- Unos postojećih biblioteka za odgovarajuće čipove (senzore i akturatore)
- Odabrati odgovarajuću biblioteku i dodati je u projekat



Name	Tags	Author	Imports	Modified	Description
TCS3472_I2C	I2C TCS3472 TCS34721 TCS347	Karl Maxwell	816	24 Apr 2014	Class which provides functions to control a TAOS TCS3472 Color Light-to-Digital Converter witl
TCS3200		Grant Phillips	227	03 May 2016	Class library for a TCS3200 Color Sensor Module.
TcpLineStream	NET network stream TCP	Hendrik Lipka	153	18 Feb 2011	A simple class to do line-based TCP communication. To be used with the NetServicesMin librar
TCS3200		Anupam Goli	121	23 Oct 2015	This is the library to use the TCS3200 on the mbed
TCC_MCSA		Flavio Lencina	45	24 Aug 2016	Serial Communication/ Analog Read/ FFT compute / LCD Text / Write on SD Card / read and v
TCP		Team DCS_TEAM	38	01 Jan 1970	server
TCP_COMM		Etienne Joannette	32	04 Apr 2016	TCP communication
TCA9548A	TCA9548A	Akash Vibhute	26	25 May 2016	TCA9548A 8 channel I2C switch with reset
TCS3472		Team Mollusc Mic	18	01 Jan 1970	TCS3472 RGBW digital I2C light sensor.
TCA9554A	GPIO TCA9554A TI	Osamu Koizumi	8	26 May 2016	Device driver for TCA9554A, which is I2C GPIO expander IC.
TC77	SPI TC77 Temperature	Dai Yokota	5	10 Jun 2016	TC77 --- SPI Thermal Sensor

Zadatak Color Click

- Napisati aplikaciju (korišćenjem postojeće biblioteke za čip TCS3471) koji na svaku sekundu
  - Očitava vrednosti CRGB registara
  - Ispisuje vrednosti osvetljaja i boje na UART2, to jest hiperterminal
- Komunikacija se obavlja preko I2C
  - Naći koji su I2C pinovi na ploči L476RG
  - Povezati pinove ploče i color click-a
    - GND
    - 3.3V
    - SDA
    - SCL

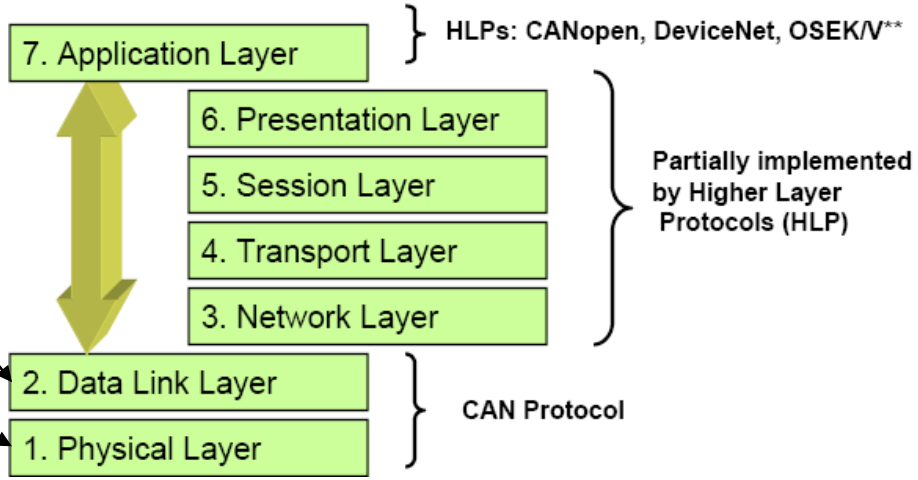
Zadatak Altitude Click

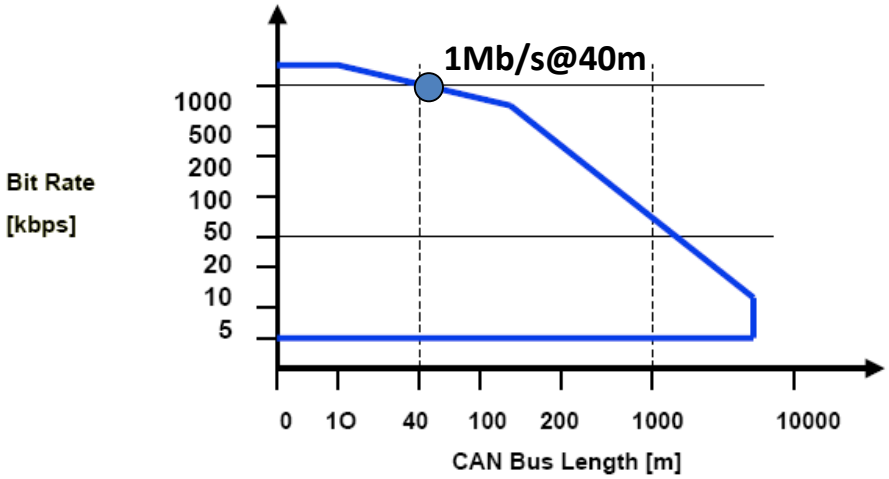
- Napisati aplikaciju (korišćenjem postojeće biblioteke za čip MPL3115A2) koji na svaku sekundu
  - Očitava vrednosti visine, pritiska i temperature
  - Ispisuje date vrednosti na UART2, to jest hiperterminal
- Komunikacija se obavlja preko I2C
  - Naći koji su I2C pinovi na ploči L476RG
  - Povezati pinove ploče i altitude click-a
    - GND
    - 3.3V
    - SDA
    - SCL

CAN protokol

Pristup magistrali, formati frejma, adresiranje

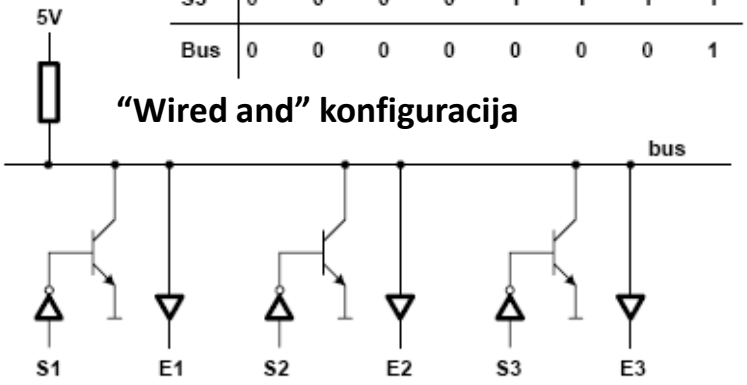
Definisanje prenosnog medijuma, povezivanja, brzine signaliziranja, linijskog kodovanja





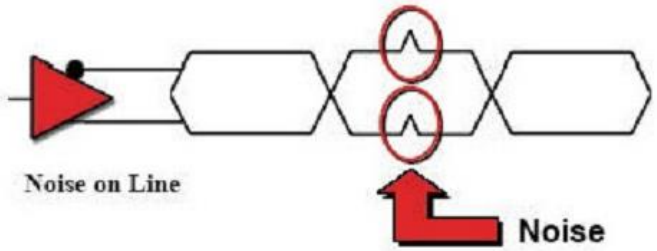
CAN protokol - PHY

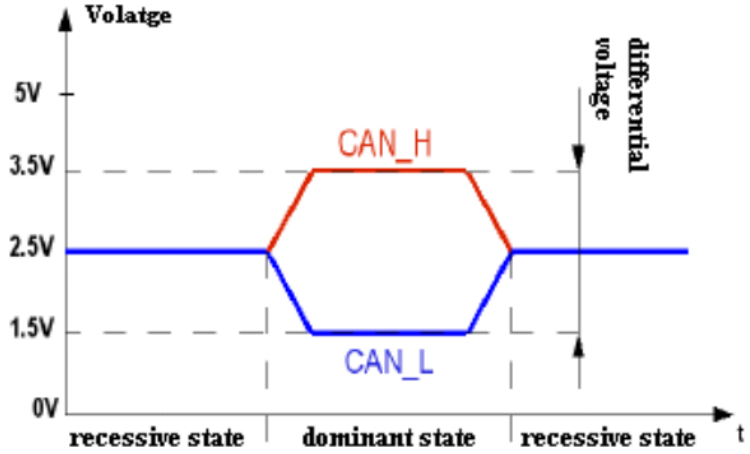
	level							
S1	0	1	0	1	0	1	0	1
S2	0	0	1	1	0	0	1	1
S3	0	0	0	0	1	1	1	1
Bus	0	0	0	0	0	0	0	1



“Wired and” konfiguracija

Differential Signal (Two Wires)

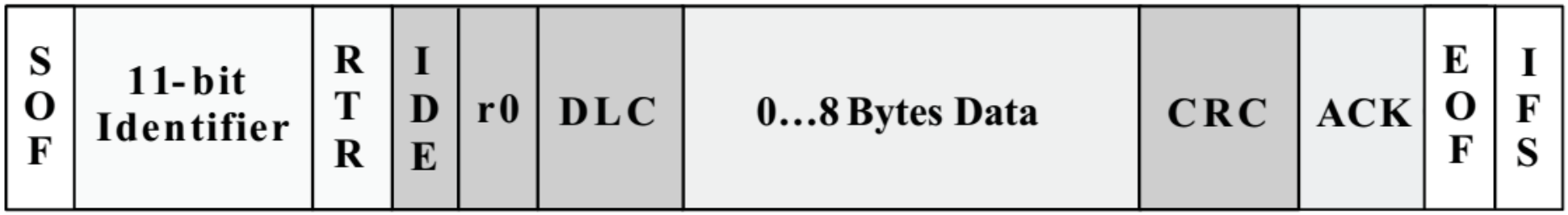




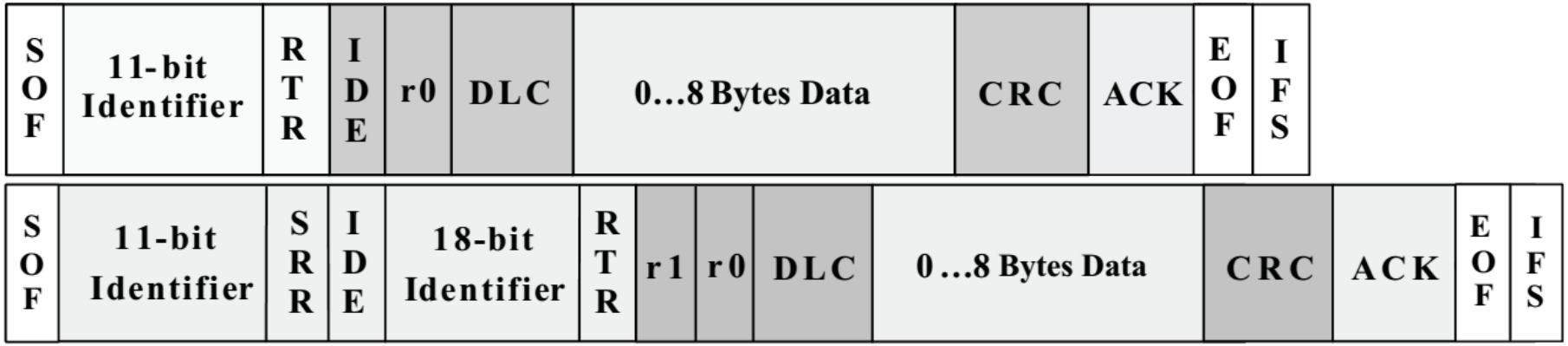
Logička stanja na magistrali

CAN format – 11 bita ID

- SOF–The single dominant start of frame
- RTR – The single remote transmission request –zahtev prozvanoj jedinici da pošalje podatke
- IDE– A dominant (0) single identifier extension – ostavljeno za proširenje
- Ro – ostavljeno za proširenja – po standardu dominantan (0)
- DLC–The 4-bit data length code – broj podataka koji se šalju
- CRC–The 16-bit cyclic redundancy check
- ACK– Node receiving an accurate message overwrites this recessive bit in the original message with a dominate bit (0), indicating an error-free message has been sent.
- EOF–This end-of-frame , 7-bit field marks the end of a CAN frame (message) – recesivni biti (1).
- IFS–This 7-bit interframe space ‘ recesivni biti (1)

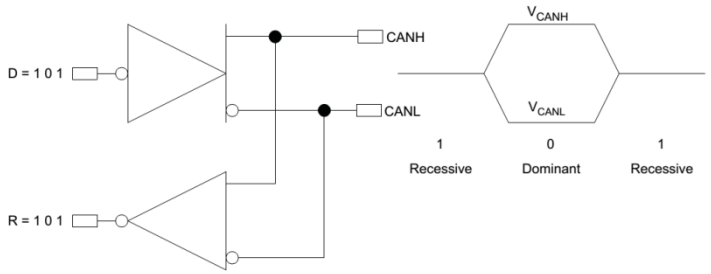


CAN format – 29 bita ID



- SRR–The substitute remote request – samo zamenjuje mesto RTR bitu u 11-bitnom formatu.
- IDE–A recessive bit in the identifier extension

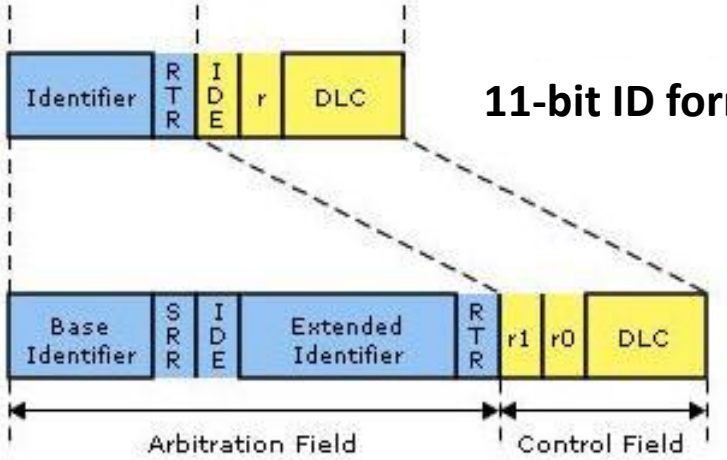
Arbitracija na magistrali



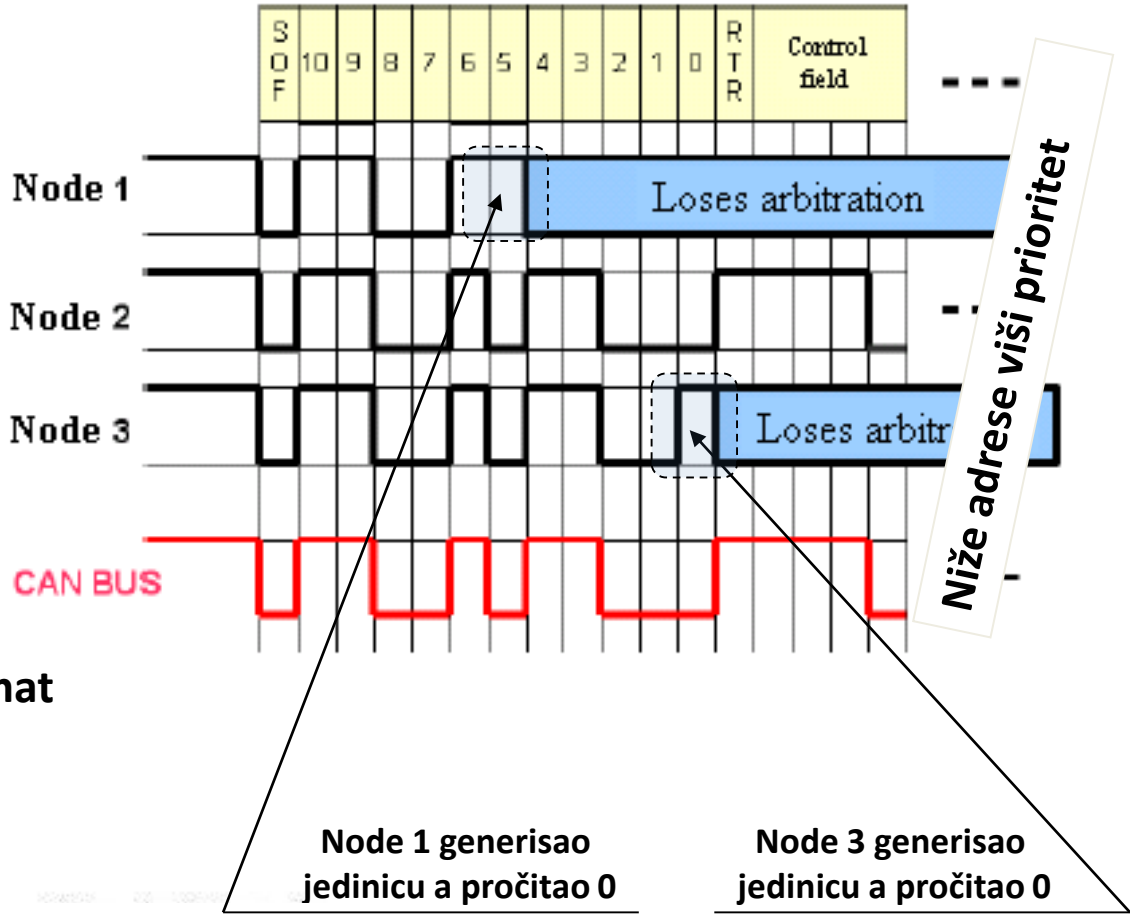
ID broj se dodeljuje:

- Prema prioritetu poruke (deadline, urg.)
- Prema tipu podatka

CAN je primenljiv za razmenu poruka kod sistema za rad u realnom vremenu



11-bit ID format

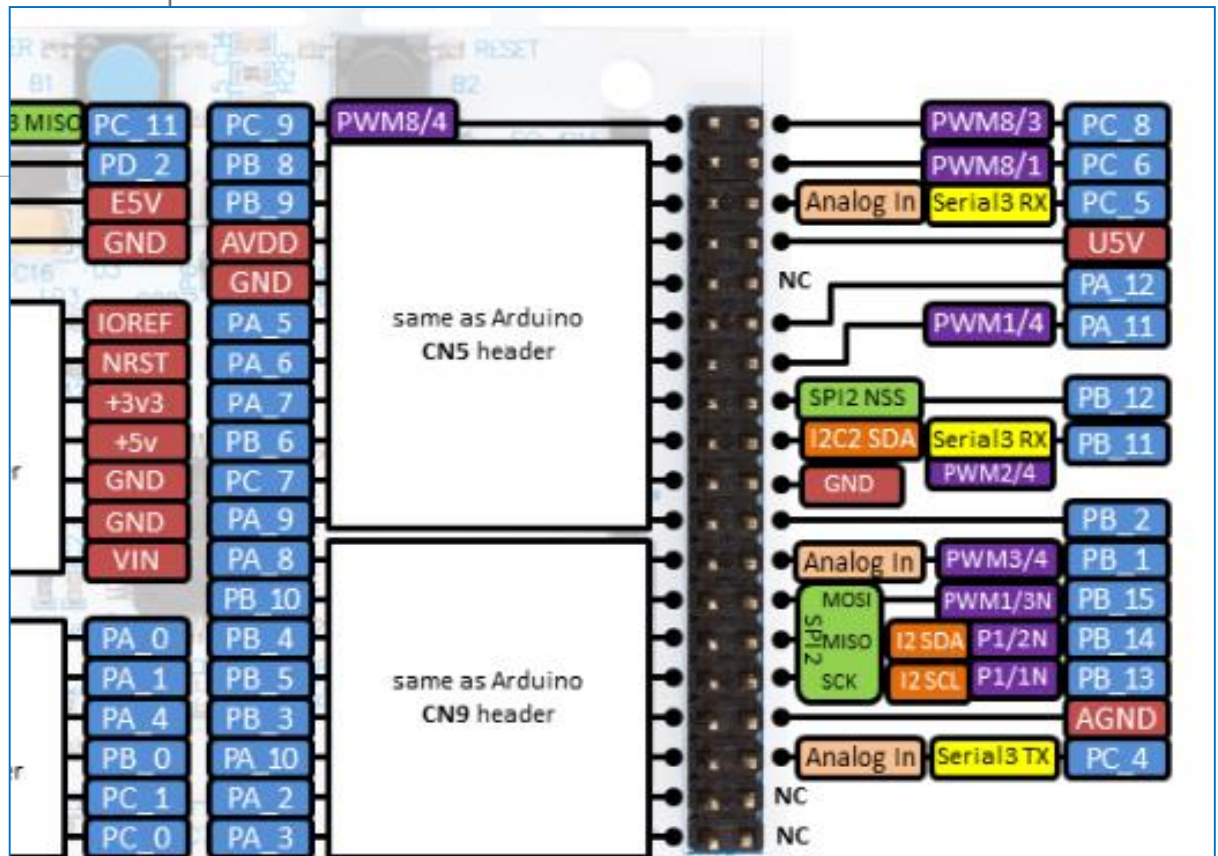


29-bit ID format

Gde je CAN na NULCEOu?

103	PA11	I/O	FT_u	-	TIM1_CH4, TIM1_BKIN2, USART1_CTS, CAN1_RX, OTG_FS_DM, TIM1_BKIN2_ COMP1, EVENTOUT	-
104	PA12	I/O	FT_u	-	TIM1_ETR, USART1_RTS_ DE, CAN1_TX, OTG_FS_DP, EVENTOUT	

- Datasheet – alternate pin functions



CAN-MBED

CAN Class Reference

`#include <CAN.h>`

Public Member Functions

CAN (PinName rd, PinName td)
Creates an **CAN** interface connected to specific pins.

int frequency (int hz)
Set the frequency of the **CAN** interface.

int write (CANMessage msg)
Write a CANMessage to the bus.

int read (CANMessage &msg, int handle=0)
Read a CANMessage from the bus.

void reset ()
Reset **CAN** interface.

void monitor (bool silent)
Puts or removes the **CAN** interface into silent monitoring mode.

int mode (Mode mode)
Change **CAN** operation to the specified mode.

int filter (unsigned int id, unsigned int mask, CANFormat format=CANAny, int handle=0)
Filter out incomming messages.

unsigned char rderror ()
Returns number of read errors to detect read overflow errors.

unsigned char tderror ()
Returns number of write errors to detect write overflow errors.

void attach (void(*fptr)(void), IrqType type=RxIrq)
Attach a function to call whenever a **CAN** frame received interrupt is generated.

template<typename T >
void attach (T *tptr, void(T::*mptr)(void), IrqType type=RxIrq)
Attach a member function to call whenever a **CAN** frame received interrupt is generated.

CANMessage Class Reference

`#include <CAN.h>`

Inherits CAN_Message.

Public Member Functions

CANMessage ()
Creates empty CAN message.

CANMessage (int _id, const char *_data, char _len=8, CANType _type=CANData, CANFormat _format=CANStandard)
Creates CAN message with specific content.

CANMessage (int _id, CANFormat _format=CANStandard)
Creates CAN remote message.

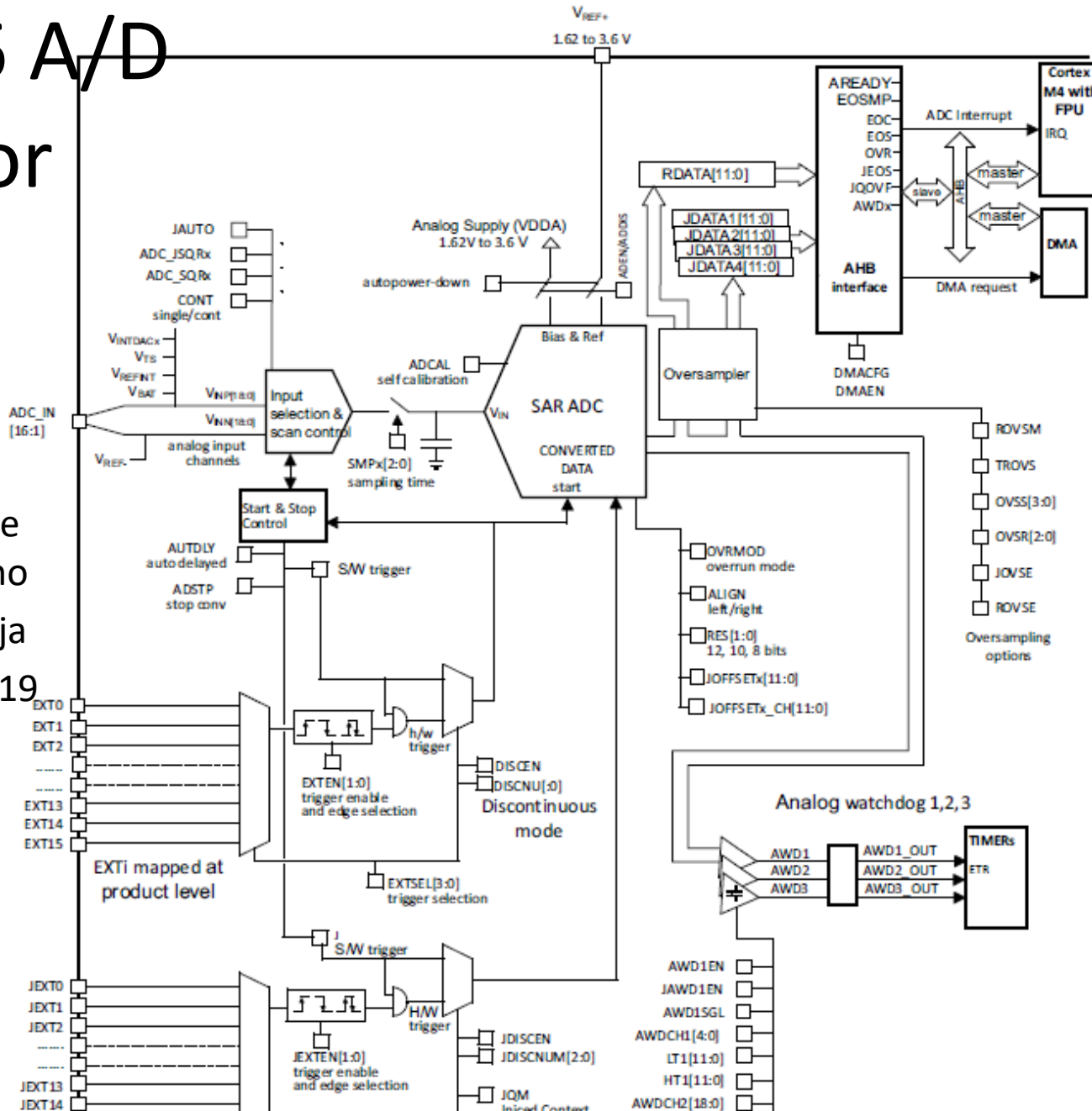
ADC

- 12-bitna rezolucija
- Nekoliko različitih izvora prekida:
  - ADC ready
  - End of Conversion
  - End of Sequence Conversion
  - Analog Watchdog 1, 2, 3
  - overrun events
- Jednostruki i kontinualni mod konverzije
- Automatska konverzija sekvence kanala
- Auto-kalibracija! Kompenzacija raznih parazitnih kapacitivnosti
- Vreme konverzije programabilno za svaki kanal ponaosob
- Eksterni trigeri
- Diskontinulani mod konverzije
- DMA prenos podržan za regularne kanale
- Veza sa integrisanim analognim temperaturnim senzorom

STM32L476 A/D konvertor

Osnovne karakteristike

- Trostruki 12-bitni A/D konvertor sa sukcesivnim aproksimacijama
- AD konvertori mogu da rade sinhronizovano i ispreplitano
- 12, 10, 8 ili 6-bitna rezolucija
- Svaki AD konvertor ima do 19 analognih ulaza
- Merni opseg 0-Vdd
- Učestanost uzorkovanja 5 Msample/s (Tconvmin =187.5ns)



Modovi konverzije

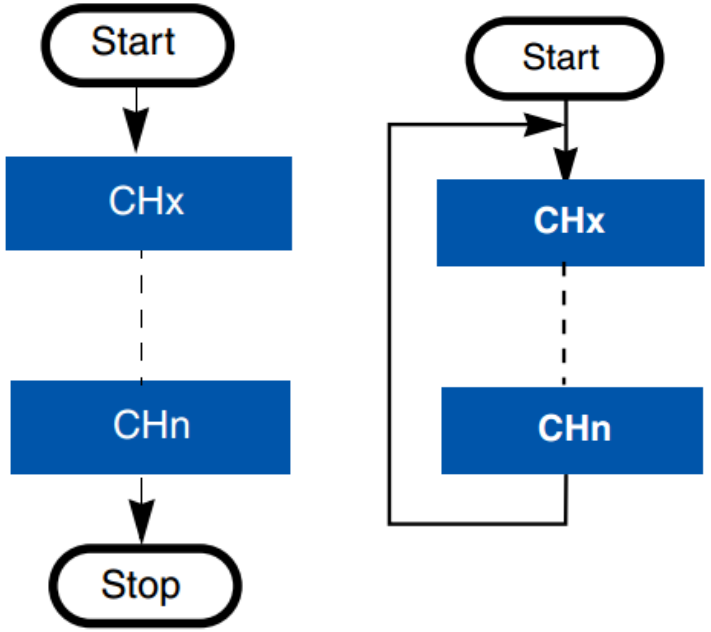
- Prema **načinu startovanja** konverzije modovi su:
  - Jednostruki mod (**single conversion**)
    - Posle svake konverzije ADC se zaustavlja uz generisanje dozvoljenih prekida ili DMA zahteva.
  - Kontinulani mod (**continuous conversion**)
    - Nakon konverzije generišu se prekidi ili DMA zahtevi ali se automatski startuje nova konverzija.
  - Diskontinualni mod (**discontinuous conversion**)
    - Može se zadati konačan broj konverzija (n<=8) nakon čijeg izvršavanja se ADC zaustavlja.
- Prema **broju kanala** koji se konvertuju modovi su:
  - **Single channel** – konvertuje se samo jedan kanal i to u jednostrukom ili kontinualnom modu
  - **Scan mode** – konvertuje se sekvenca kanala, i to u jednostrukom modu (single conversion), kontinualnom modu ili diskontinulanom modu. U ovom modu moguće je definisati do 16 proizvoljnih regularnih kanala ili do 4 injektovana kanala.

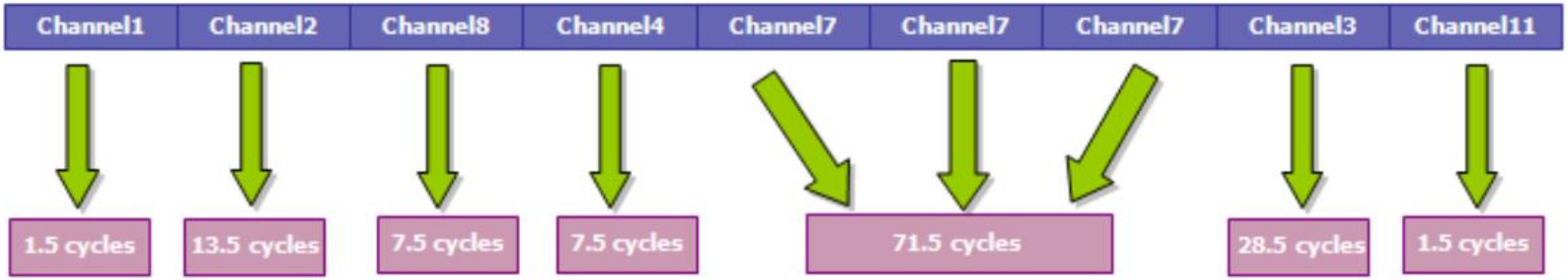
Startovanje konverzije

- Konverzija se može startovati:
  - **Softverski**: pozivom funkcije za start konverzije i za regularne i za injektovane kanale
  - **Hardverski**: eksternim ili internim hardverskim trigerima čiji se polaritet (uzlazna, silazna ili obe ivice) mogu definisati. Takođe podržano i za regularne i za injektovane kanale.
    - interni triger je na primer Tajmer koji periodično pokreće konverziju (TRGO), a
    - eksterni je triger signal koji se dovodi sa određenog GPIO pina.

Multichannel (scan) mode

- Sukcesivna konverzija više kanala, pri čemu se za svaku konverziju može da se definiše različito vreme semplovanja.
- Do 16 kanala za regularnu grupu i do 4 za injektovanu grupu.





Dozvoljena vremena semplovanja

Each channel can be sampled with a different sampling time which is programmable using the SMP[2:0] bits in the ADC_SMPR1 and ADC_SMPR2 registers. It is therefore possible to select among the following sampling time values:

- SMP = 000: 2.5 ADC clock cycles
- SMP = 001: 6.5 ADC clock cycles
- SMP = 010: 12.5 ADC clock cycles
- SMP = 011: 24.5 ADC clock cycles
- SMP = 100: 47.5 ADC clock cycles
- SMP = 101: 92.5 ADC clock cycles
- SMP = 110: 247.5 ADC clock cycles
- SMP = 111: 640.5 ADC clock cycles

The total conversion time is calculated as follows:

T_{CONV} = Sampling time + 12.5 ADC clock cycles

Example:

With F_{ADC_CLK} = 80 MHz and a sampling time of 2.5 ADC clock cycles:

T_{CONV} = (2.5 + 12.5) ADC clock cycles = 15 ADC clock cycles = 187.5 ns (for fast channels)

The ADC notifies the end of the sampling phase by setting the status bit EOSMP (only for regular conversion).

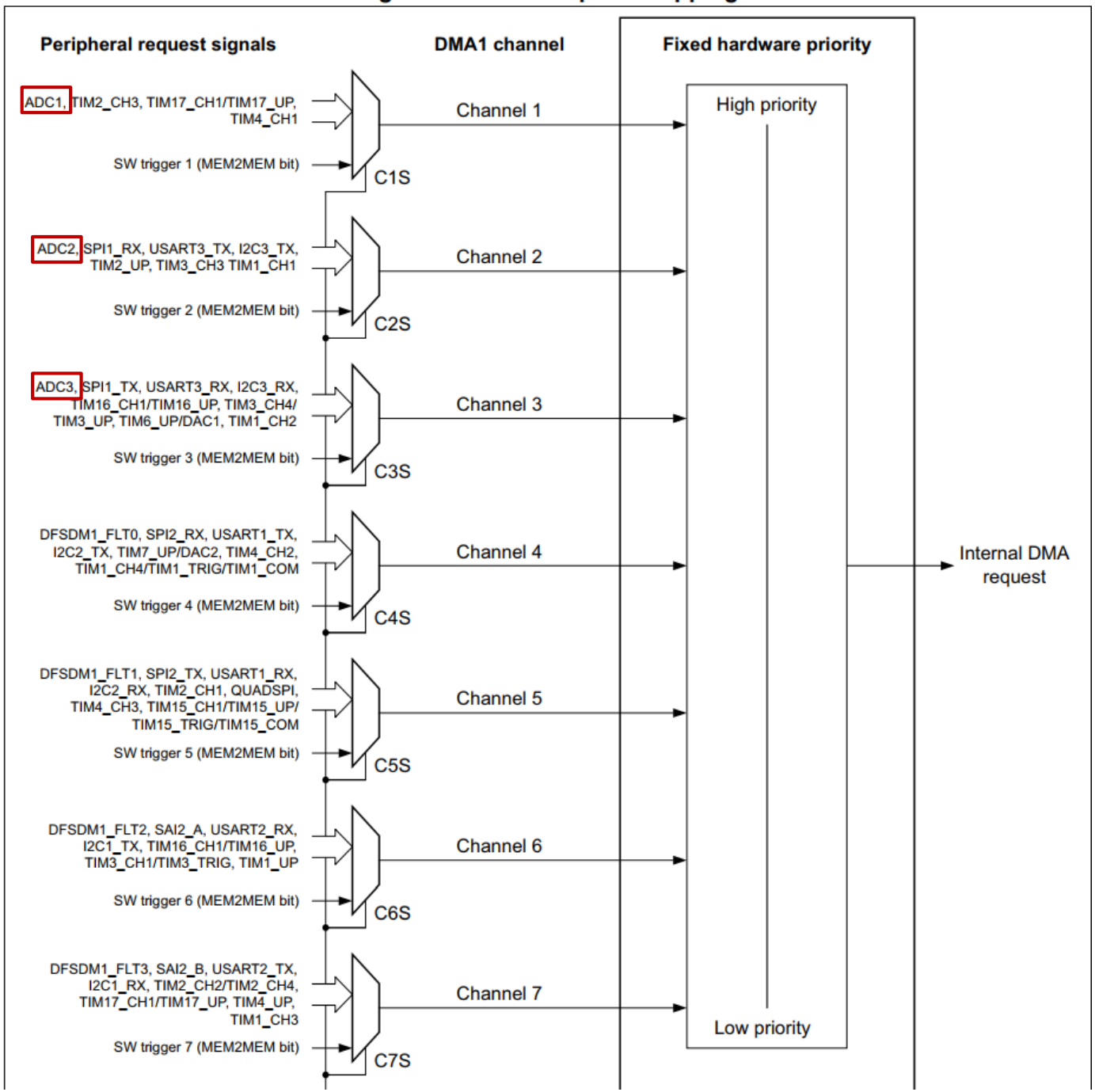
ADC DMA

- DMA transfer limited:
  - ADC conversions are transferred by DMA, in limited mode (one shot mode): DMA transfer requests are stopped when number of DMA data transfers (number of ADC conversions) is reached. This ADC mode is intended to be used with DMA mode non-circular.
- DMA transfer unlimited:
  - ADC conversions are transferred by DMA, in unlimited mode: DMA transfer requests are unlimited, whatever number of DMA data transferred (number of ADC conversions). This ADC mode is intended to be used with DMA mode circular.
- DMA transfer none

DMA1 kanali

- Reference manual

Figure 30. DMA1 request mapping



Analog Watchdog

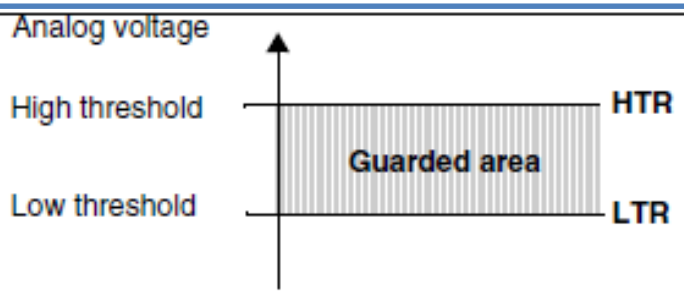


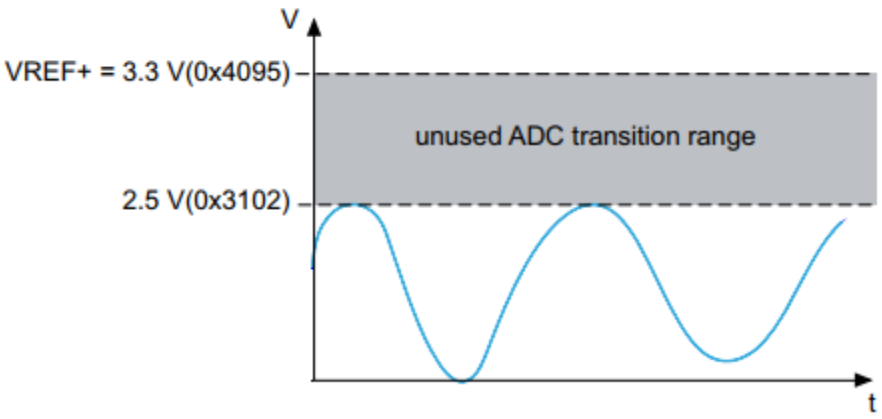
Table 58. Analog watchdog channel selection

Channels to be guarded by analog watchdog	ADC_CR1 register control bits (x = don't care)		
	AWDSGL bit	AWDEN bit	JAWDEN bit
None	x	0	0
All injected channels	0	0	1
All regular channels	0	1	0
All regular and injected channels	0	1	1
Single(1) injected channel	1	0	1
Single(1) regular channel	1	1	0
Single(1) regular or injected channel	1	1	1

Iskorišćenje ADC opsega

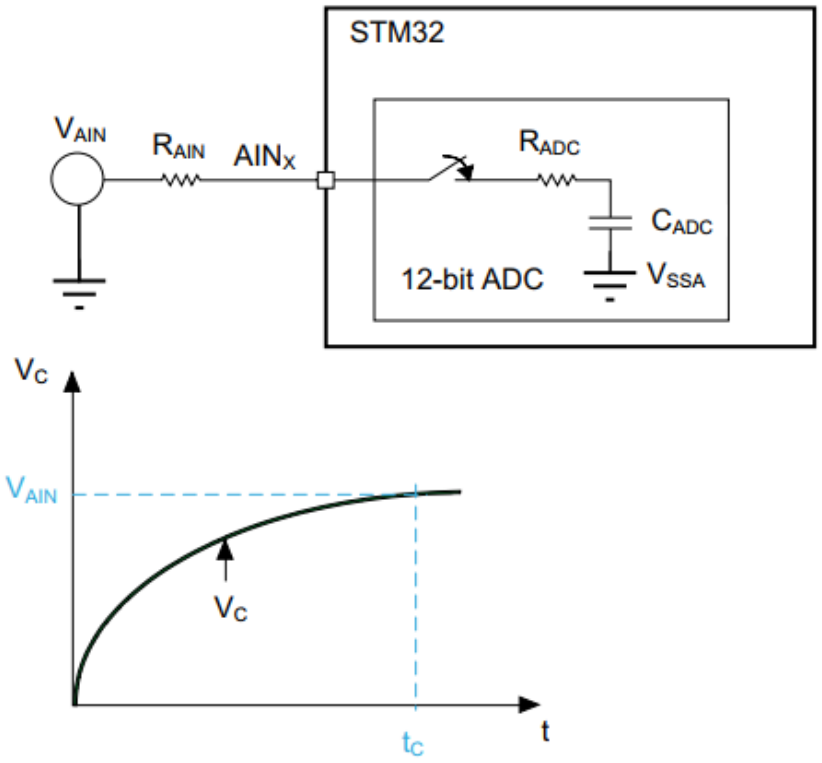
- Želimo da postignemo što bolju rezoluciju i iskoristimo ceo opseg ADC-a
  - Na Vref vežemo naš maksimalni napon
  - Koristimo sabirač
  - Pojačamo naš signla da bude u punom opsegu ADC-a

Figure 14. Input signal amplitude vs. ADC dynamic range



Izlazna otpornost analognog izvora

Figure 15. Analog signal source resistance effect

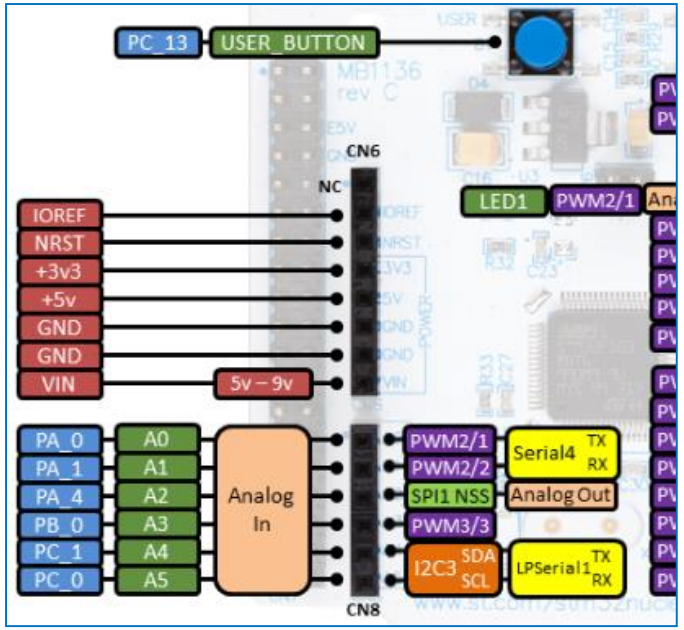


1. t_c is the time taken by the C_{ADC} capacitor to fully charge: $V_c = V_{AIN}$ (with max.1/2 LSB error)
   V_c: capacitor (C_{ADC}) voltage
   $t_c = (R_{ADC} + R_{AIN}) \times C_{ADC}$

MBED Analogni ulazi

http://mbed.org/handbook/AnalogIn

```
//Turn on a LED when AnalogIn goes
    above 0.3
#include "mbed.h"
AnalogIn ain(A0);
DigitalOut led(LED1);
int main() {
        while (1){
                if(ain > 0.3) {
                        led = 1;
                }
                else {
                        led = 0;
                }
        }
}
```



AnalogIn Class Reference

```
#include <AnalogIn.h>
```

Public Member Functions

AnalogIn (PinName pin)
Create an **AnalogIn**, connected to the specified pin.

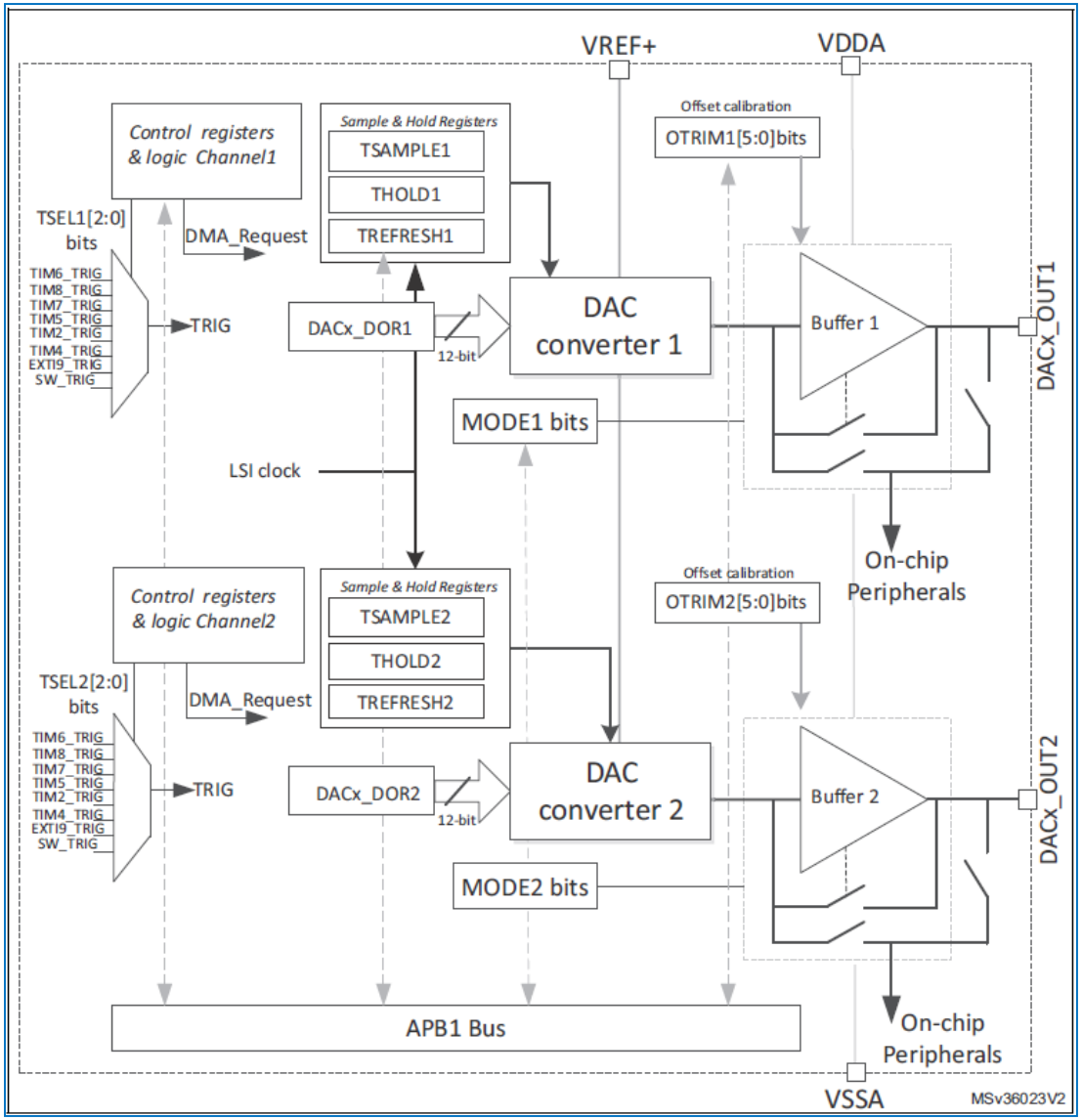
float read ()
Read the input voltage, represented as a float in the range [0.0, 1.0].

unsigned short read_u16 ()
Read the input voltage, represented as an unsigned short in the range [0x0, 0xFFFF].

operator float ()
An operator shorthand for read()

STM32L476 D/A konvertor

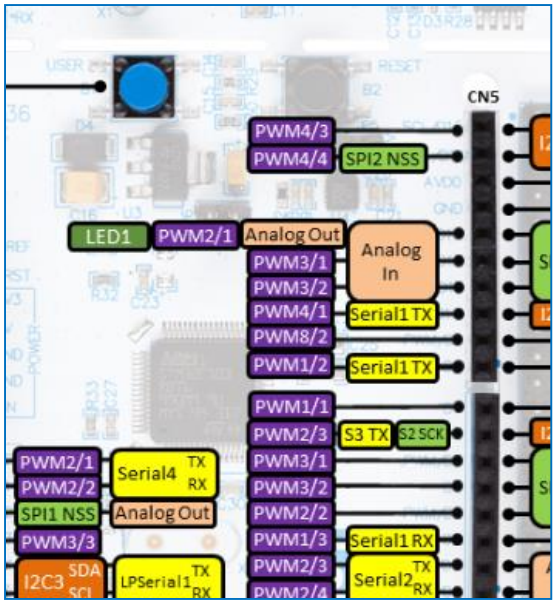
- Dvostruki 12-bitni D/A kovertor sa otpornim razdelinicima
- Eksterni trigeri
- Vreme smirivanja manje od 3us
- Podrška za DMA prenos
- Generator šuma
- Generator trougaonog talasnog oblika



MBED Analogni izlazi

http://mbed.org/handbook/AnalogIOut

```
#include "mbed.h"
AnalogOut signal(pA_5);
int main() {
    while(1) {
        for(float i=0.0; i<1.0; i+=0.1) {
            signal = i;
            wait(0.0001);
        }
    }
}
```



Public Member Functions

	AnalogOut (PinName pin) Create an **AnalogOut** connected to the specified pin.
void	write (float value) Set the output voltage, specified as a percentage (float)
void	write_u16 (unsigned short value) Set the output voltage, represented as an unsigned short in the range [0x0, 0xFFFF].
float	read () Return the current output voltage setting, measured as a percentage (float)
AnalogOut &	operator= (float percent) An operator shorthand for write() operator float () An operator shorthand for read()

ADC I DAC

- Unakrsno povezati AD i DA konvertore dva MBED-a.
- Na jednom MBED-u se izvršava program koji vrši DA konverziju podatka serijski poslatog sa računara iz hiper terminala.
- Na drugom MBED-u se izvršava program koji radi AD konverziju podatka i serijski ga šalje na računar u hiper terminal.