NeoPixel ring u ulozi analognog sata

Irvana Hrustić i Lejla Vinčević

Korištene komponente

-STM32F407 Disco Board

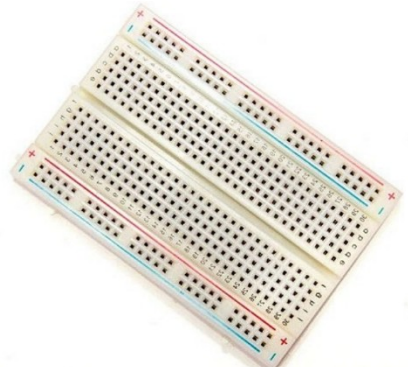
-Neopixel ring (WS2812)

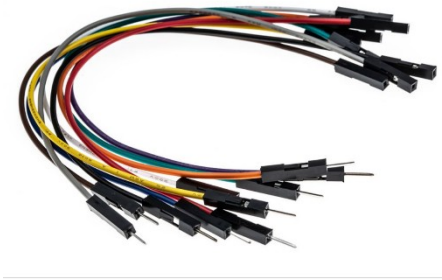
-Kondenzator

-6 Wire jumpers [MM,FF,MF]

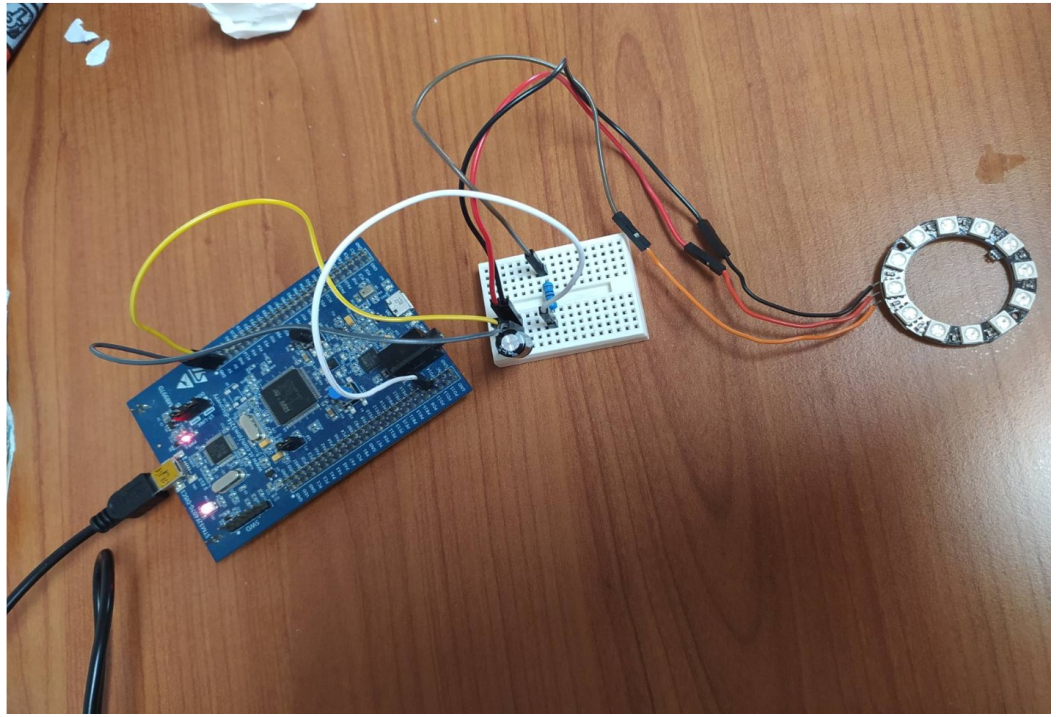
-Bredboard

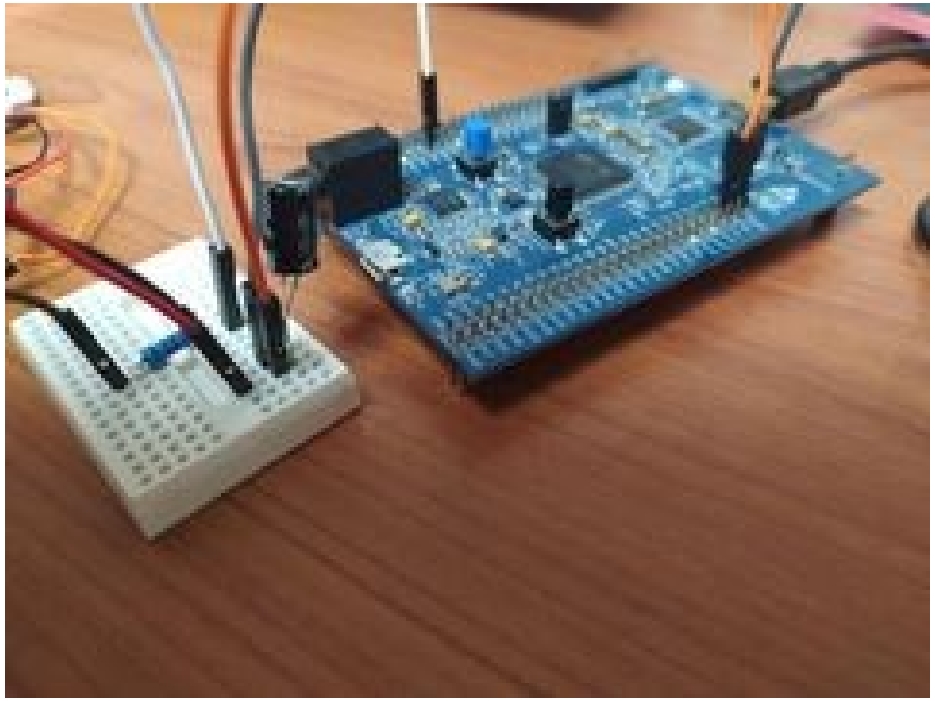
-Otpornik (300-500[Ohm])





Spoj Neopixel sa STM32F407





Kondenzator između vcc i ground, otpornik(470 Ohm) kao zaštita, produžen Ring sa 3 jumpera, a 3 jumpera korištena za spajanje na breadboard.

Neopixel ring

-adresni ili pametni RGB LED nazvan Neopixel

-upravljanje RGB LED diodama je upravljane levela crvene, plave i zelene,(Lightness i Temperature, Saturation) te time kreiranje željene boje (ovo upravljanje zasniva se na teoriji boja)

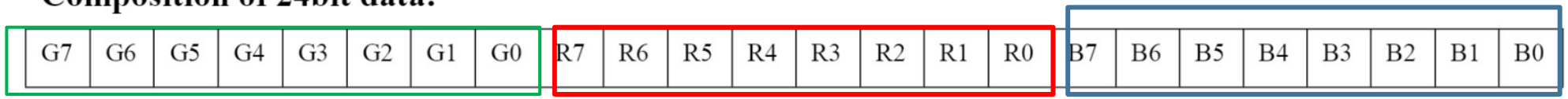
-Teorija boja- kombinacija boja kojom se dostiže specifican vizualni efekat

-Podaci unutar LEDA Ringa- svaka dioda ima 3 byte->ring od 12 LED ima 36byte

Neopixel ring

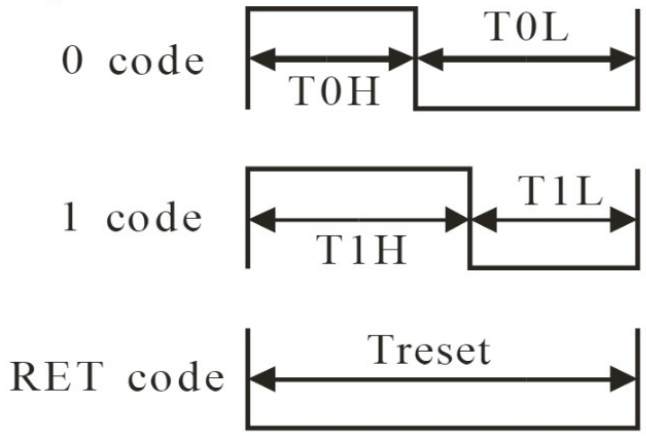
-Svaki led treba 24 byte (8b zelene, 8b crvene, 8b plave)

Composition of 24bit data:



-Svaki od tih bita je napisan u sljedećoj formi

Sequence chart:



Za 0-data line će da se popne za dužinu od TH0 i da se spusti za dužinu THL

Za 1- popne se za T1H, a spusti se T1L

Ako je bilo low duže vrijeme od reset vremena, tada LED posmatra prijenos kao završen, resetovaće za idući prijenos

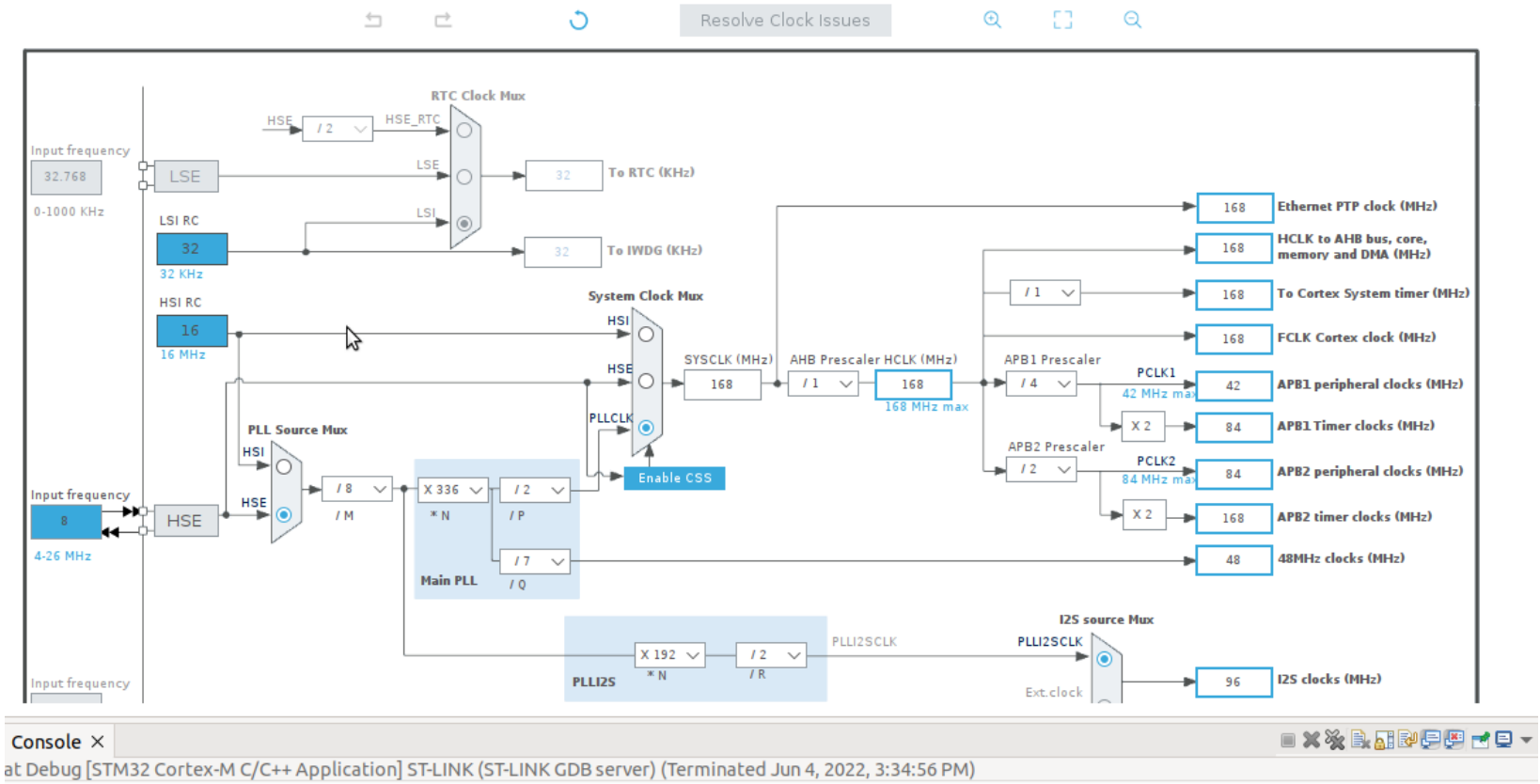
Neopixel ring

Data transfer time(TH+TL=1.25µs±600ns)

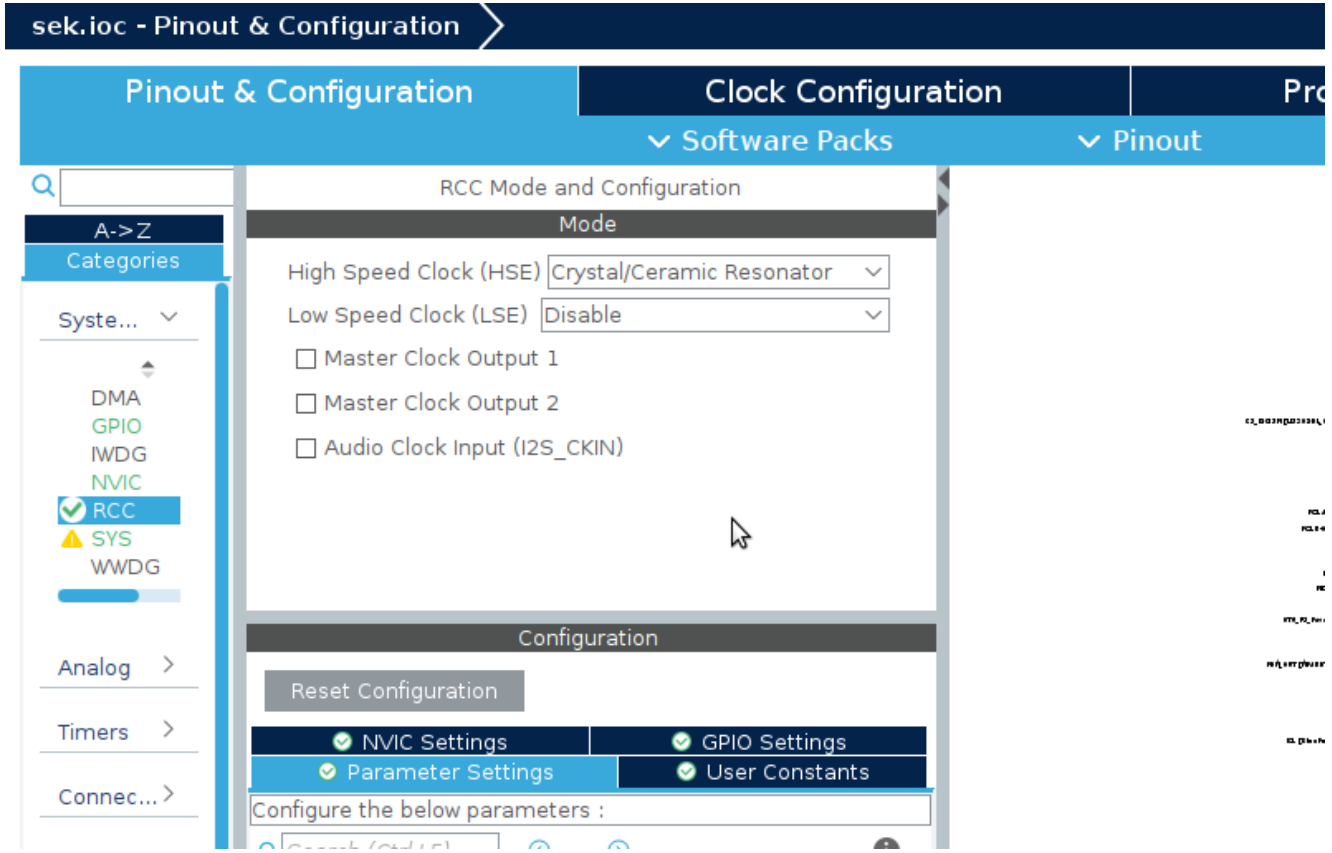
T0H	0 code ,high voltage time	0.35us	±150ns
T1H	1 code ,high voltage time	0.7us	±150ns
T0L	0 code , low voltage time	0.8us	±150ns
T1L	1 code ,low voltage time	0.6us	±150ns
RES	low voltage time	Above 50µs	

Tajminzi potrebni za prsten, da bi ispravno radio.

Clock Configuration



RCC Mode i Configuration



```
/* USER CODE END Header */
/* Includes ------------------------------------------------------------------
#include "main.h"
#include "stm32f4xx_hal.h"

#define timer_freq 84.0  //timer clock freq in MHz
#define T0H 0.35  //each different clone can have their own timings
#define T1H 0.7  //timing here are in us
#define T0L 0.8
#define T1L 0.6
#define Treset 50

uint8_t LED_data[36]; //I have a ring with 12 LEDs and need 3 bytes/LED
```

Tajminzi koji su nam potrebni za prsten. Slika sa kodom u kojem su isti definisani.

Prsten sadrži 12 dioda, svaka dioda po 3 byte, ukupno 36

```
uint16_t pos;
uint8_t mask = 0B10000000;
uint8_t lastbit;

long double period;
uint16_t low_CCR1, low_ARR, high_CCR1, high_ARR, treset_ARR;

void Neopixel_setup(void){

    //calculate all the timings.
    period = 1 / timer_freq;
    low_CCR1 = round(T0H / period);
    low_ARR = round((T0H + T0L) / period);
    high_CCR1 = round(T1H / period);
    high_ARR = round((T1H + T1L) / period);
    treset_ARR = ceil(Treset / period);
```

-Varijabla pomoću koje prolazimo kroz niz,
-Varijabla kojom biramo pojedinačni byte
-flag koji koristima da znamo da li su svi biti poslani ili ne

-Varijabla koja pohranjuje dužinu svakog cycle tajmera

```
void Neopixel_setup(void){

    //calculate all the timings.
    period = 1 / timer_freq;
    low_CCR1 = round(T0H / period);
    low_ARR = round((T0H + T0L) / period);
    high_CCR1 = round(T1H / period);
    high_ARR = round((T1H + T1L) / period);
    treset_ARR = ceil(Treset / period);

    RCC->AHB1ENR |= RCC_AHB1ENR_GPIODEN; //enable port D clock
    GPIOD->MODER |= GPIO_MODER_MODER12_1; //setup pin 12 on port d to AF mode
    GPIOD->AFR[1] = (GPIOD->AFR[1] & (0b1111<<(4*(12-8))) | 0b0010<<(4*(12-8))); //setup pin 12 on port d to AF timer 4

    RCC->APB1ENR |= RCC_APB1ENR_TIM4EN; //enable the timer4 clock
    TIM4->PSC = 0;   //set prescale to zero as timer has to go as fast as posible
    TIM4->CCMR1 = (TIM4->CCMR1 & ~(0b110<<4)) | (0b110<<4); //set PWM mode 110
    TIM4->CCR1 = 0; //set to zero so that the pin stay low until transmission
    TIM4->ARR = treset_ARR; //set to timing for reset LEDs
    TIM4->CCER |= TIM_CCER_CC1E; //enable output to pin.
    TIM4->CR1 |= TIM_CR1_CEN; //Disable channel 1. This bit is used to start and stop transmission.
    TIM4->CR1 |= TIM_CR1_ARPE; //buffer ARR
    TIM4->CCMR1 |= TIM_CCMR1_OC1PE; //buffer CCR1
    TIM4->DIER &= ~TIM_DIER_UIE; // ensure we are not enabling interrupt flag to be generated this bit is used to start/stop transmission
    TIM4->CR1 |= TIM_CR1_CEN; //enable channel 1.

    NVIC_EnableIRQ(TIM4_IRQn); // Enable interrupt(NVIC level)
}
```

Biće high za vrijednost u CCR1 registru
Bit će ostati low dok ne dobije vrijednost high

-enable timer
-prescale setujemo na 0
-timer disable, želimo operaciju nad pinom samo kad šaljemo podatke
-bit u ovom registru dozvoljava da se promijene u ostalim registrima izvrše odmah
Ukoliko ovaj bit nije setovan, dok ne završi ciklus, update se neće izvršiti

```
void show_neopixels(){
    pos = 0; //set the interupt to start at first byte
    lastbit = 0;
    mask = 0B10000000; //set the interupt to start at second bit

    TIM4->SR &= ~TIM_SR_UIF; // clear UIF flag
    TIM4->DIER |= TIM_DIER_UIE; //enable interupt flag to be generated to start transmission
}
```

-Funkcija za prenos podataka, ako je prvi bit high setujemo ga za high tajming, else setujemo ga za low tajming
-Pozicija je nula, jer želimo da intterupti startaju od te pozicije
-Postavljamo na nulu
-Mask pocinje od 2-bita jer smo iznad postavili prvi bit
-enable output pin i tajmer
-Tajmer započinje sa izvršenjem

Kraj prvog ciklusa-promjena ARR, izvršenje interrupta, tajmer nastavlja sa izvršenjem

```
void TIM4_IRQHandler(void){

    TIM4->SR &= ~TIM_SR_UIF; // clear UIF flag

        if(pos<sizeof(LED_data)){
            if(LED_data[pos] & mask){
                TIM4->CCR1 = high_CCR1;
                TIM4->ARR = high_ARR;
            }else{
                TIM4->CCR1 = low_CCR1;
                TIM4->ARR = low_ARR;
            }
            if(mask==1){
                mask = 0B10000000;
                pos+=1;
            }else mask = mask >> 1;
        }else{
            TIM4->CCR1 = 0; //set to zero so that pin stays low
            TIM4->ARR = treset_ARR; //set to timing for reset LEDs
            TIM4->DIER &= ~TIM_DIER_UIE; //disable interrupt flag to end transmission.
        }
}
```

-clear flag-zbog ponovnog izvrsenja
-Ako je poziciha manja od velicine LED-nismo proslu kroz sve LED podatke
Za bilo koji mask- ako je 1, tajming je 1, u suprotnom 0

```
  while (1){
//1000 je 1 sek, priblizno!!!
//5000 je 5 sek priblizno!!! (12 LED = 60 / 12 = 5)!


      // Green LEDS are for minute
      for (uint8_t i = 0; i < 36; i+=3){
          LED_data[i] = 25; // brightness is 25(255 is max)
          show_neopixels();  //transmit the data to the neopixel ring.
        //Red LEDS sre for seconds
          for(uint8_t j = 1; j < 36; j+=3){
                    LED_data[j] = 25;
                    show_neopixels();  //transmit the data to the neopixel ring.
                    HAL_Delay(5000);
                    LED_data[j] = 0; //brightness is zero!
      }
          LED_data[i] = 0;
      }

  }
}
```

Napravljene su dvije for petlje, jedna vanjska (za minute) i jedna unutrašnja (za sekunde). Jedna LED zelene boje(za minute) i jedna crvene boje(za sekunde)