Univerzitet u Nišu
Prirodno-matematički fakultet
Departman za računarske nauke

Predmet:
Internet pametnih uređaja

Praktična vežba:
Primeri programa

Cilj vežbe:

Opis vežbe:

U okviru ove vežbe student je u obavezi da realizuje sledeće aktivnosti:

- 1.0 Razvoj i testiranje programa za kontrolu LED
- 2.0 Razvoj i testiranje programa za povezivanje na WiFi
- 3.0 Razvoj i testiranje programa za ocitavanje adrese DS18S20

1.0 Razvoj i testiranje programa za kontrolu LED

Varijanta programa za razvojne ploce sa klasicnom LED:

```
#define LED 2 // ovde treba staviti broj esp32 pin-a na koji je vezana dioda

void setup(void) {
      pinMode(LED,OUTPUT); // Set pin mode

      Serial.begin(115200);
      Serial.printf ("Setup finised.\n");
}


void loop(void) {
      delay(2000);
      digitalWrite(LED, HIGH);

      Serial.printf ("Pali/gasi \n");

      delay(300);
      digitalWrite(LED, LOW);
}
```

Varijanta programa za razvojne ploce sa WS2812 LED. U ovom slucaju treba najpre u projekat dodati bibilioteku FastLED nakon cega ce se u fajlu Platformio.ini pojaviti:

```
lib_deps = fastled/FastLED@^3.10.3
```

a sam program bi izgledao ovako:

```
#include "FastLED.h"

#define NUM_LEDS 1
#define DATA_PIN 21 //  broj esp32 pin-a na koji je vezana WS281 dioda

CRGB leds[NUM_LEDS];

void setup()
{
      FastLED.addLeds<WS2812B, DATA_PIN, GRB>(leds, NUM_LEDS); // GRB ordering
is typical
      FastLED.setBrightness(50);
}

void loop()
{
      leds[0] = CRGB::Red;
      FastLED.show();
      delay(500);

      leds[0] = CRGB::Green;
      FastLED.show();
      delay(500);

      leds[0] = CRGB::Blue;
      FastLED.show();
      delay(500);

      leds[0] = CRGB::White;
      FastLED.show();
      delay(500);

      leds[0] = CRGB::Black;
      FastLED.show();
      delay(500);
}
```

Druga varijanta je da se koristi Adafruit_NeoPixel biblioteka za kontrolu WS2812 LED.

```
#include <Arduino.h>
#include <Adafruit_NeoPixel.h>
```

```
#define PIN_LED_RGB   21 // RGB LED is connected to the GPIO 21

// #define PORT_SERIAL USBSerial // Use USB for communication
#define PORT_SERIAL Serial // Use USB for communication
// #define PORT_SERIAL Serial0 // Use UART0 for communication
// #define PORT_SERIAL Serial1 // Use UART1 for communication

Adafruit_NeoPixel led (1, PIN_LED_RGB, NEO_GRB + NEO_KHZ800);

void setup() {
      PORT_SERIAL.begin (115200);
      delay (500);

      PORT_SERIAL.println ("Blink with ESP32-S3-Zero-S3FH4R2");
      led.begin();
}

void loop() {
      led.clear();

      // Cycle through the three colors.
      led.setPixelColor (0, led.Color (150, 0, 0));
      led.show();
      PORT_SERIAL.println ("RED");
      delay (500);

      led.setPixelColor (0, led.Color (0, 150, 0));
      led.show();
      PORT_SERIAL.println ("GREEN");
      delay (500);

      led.setPixelColor (0, led.Color (0, 0, 150));
      led.show();
      PORT_SERIAL.println ("BLUE");
      delay (500);
}
```

2.0 Razvoj i testiranje programa za povezivanje na WiFi

```
#include <Arduino.h>

#include <WiFi.h>

#define WIFI_SSID "????S"
#define WIFI_PASSWORD "??????"

// MAC addres
uint8_t baseMac[6];
```

```
  char apName[] = "ada-xxxxxxxxxxxx";
  char mac_adr [13];

  void setup(void) {
        Serial.begin(115200);

        esp_read_mac(baseMac, ESP_MAC_WIFI_STA); // Get MAC address for WiFi
station

        sprintf (apName,  "ada-%02X%02X%02X%02X%02X%02X", baseMac[0], baseMac[1],
baseMac[2], baseMac[3], baseMac[4], baseMac[5]);
        sprintf (mac_adr, "%02X%02X%02X%02X%02X%02X",  baseMac[0], baseMac[1],
baseMac[2], baseMac[3], baseMac[4], baseMac[5]);

        WiFi.setHostname(apName);

        WiFi.mode(WIFI_STA);

        Serial.println("Connecting to Wi-Fi...");
        WiFi.begin(WIFI_SSID, WIFI_PASSWORD);

        while (WiFi.status() != WL_CONNECTED) {
              Serial.print('.');
              delay(1000);
        }

        Serial.println ();
        Serial.print ("IP adresa: ");
        Serial.println (WiFi.localIP());

        Serial.printf ("\nSetup zavrsen.\n");
  }

  void loop (void) {

  }
```

3.0 Razvoj i testiranje programa za ocitavanje adrese DS18S20

```
  #include <Arduino.h>
  #include <OneWire.h>

  #define SENSOR_PIN 23

  OneWire ds(SENSOR_PIN);

  void setup(void) {
        Serial.begin(115200);
        Serial.printf ("Setup zavrsen.\n");
```

```
}

void loop(void) {

      byte i;
      byte present = 0;
      byte data[12];
      byte addr[8];

      ds.reset_search();

      if ( !ds.search(addr)) {
            Serial.print("Nema vise mogucih adresa.\n");
            ds.reset_search();
            return;
      }

      Serial.print("R=");

      for( i = 0; i < 8; i++) {
            Serial.print(addr[i], HEX);
            Serial.print(" ");
      }


      if ( OneWire::crc8( addr, 7) != addr[7]) {
            Serial.print("CRC nije validan!\n");
            return;
      }

      if ( addr[0] == 0x10) {
            Serial.print("Ovaj senzor pripada DS18S20 familiji.\n");
      }
      else {
            if ( addr[0] == 0x28) {
                  Serial.print("Ovaj senzor pripada DS18B20 familiji.\n");
            }
            else {
                  Serial.print("Ne moze se prepoznati familija senzora.: 0x");
                  Serial.println(addr[0],HEX);
                  return;
            }
      }

      ds.reset();
      ds.select(addr);
      ds.write(0x44,1); // start A/D konverzije

      delay(1000);      // 750ms je dovoljno

      present = ds.reset();
```

```
        ds.select(addr);

        ds.write(0xBE);   // citamo rezultat A/D konv

        Serial.print("P=");
        Serial.print(present,HEX);
        Serial.print(" ");

        for ( i = 0; i < 9; i++) {    // nama treba 9 bajtova
              data[i] = ds.read();
              Serial.print(data[i], HEX);
              Serial.print(" ");
        }

        Serial.print(" CRC=");
        Serial.print (OneWire::crc8( data, 8), HEX);
        Serial.println();
  }
```