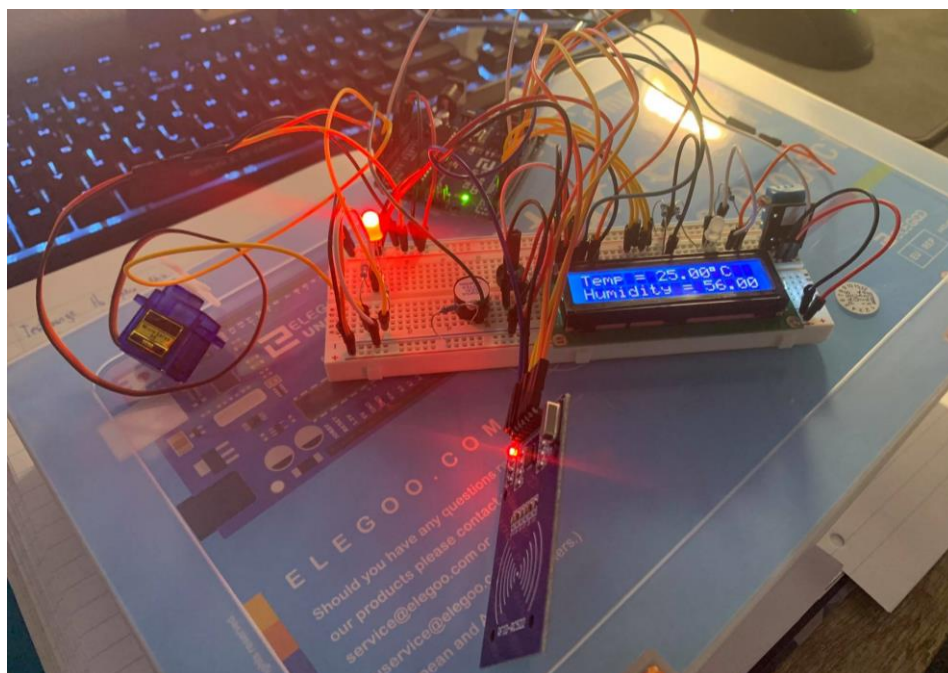


PROJEKTA NALOGA VIN – Koncept pametne sobe

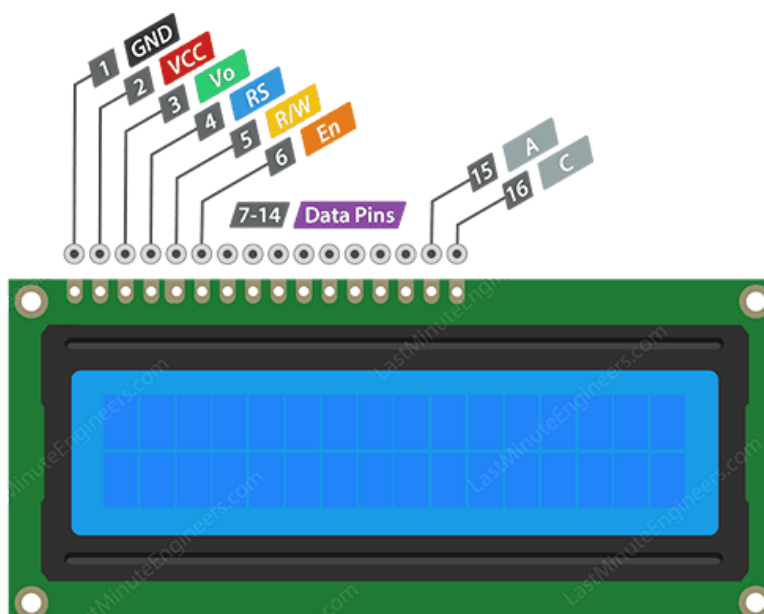
Opis projekta

Za projekt pri predmetu vzhodno-izhodne naprave sem si izbral nek »koncept« pametne sobe.

LCD zaslon nam prikazuje trenutno temperaturo ter vlažnost, z ustrezno kartico oz. čipom pa sežemo do RFID bralnika, ta pa sproži rotacijo servo motorja. V primeru, da je v sobi tema, se nam ob uspešnem skeniranju prižge tudi luč. Ob drugem skeniranju kartice se servo motor premakne na prvotno pozicijo.



Priklop LCD zaslona



Prve dva pina (GND, VCC) služita za napajanje LCD zaslona. Ustrezno jih povežemo na 5V in GND. Tretji pin (označen z Vo) povežemo na izhod 10kΩ potenciometra. Z prilagajanjem upornosti na tem potenciometru bomo spreminjali kontrast in svetlost zaslona.

Tretji pin (označen z RS – register select) služi za ločevanje podatkov od ukazov npr. v primeru, da zaslonu pošljamo podatke/značke, je RS pin postavljen na visoko stanje. V obratnem primeru zaslonu pošljamo ukaze – brisanje zaslona, premikanje kurzorja itd. Pin povežemo na poljuben digitalni priključek na arduinotu.

Na četrtem pinu (označen z R/W – branje/pisanje) nastavimo delovanje zaslona. Če bomo iz zaslona brali informacije, pin nastavimo na visoko stanje, v obratnem primeru pa na nizko. V našem primeru bomo na zaslon samo pisali, zato pin povežemo kar na GND.

Peti pin (označen z En – enable) omogoča kontroliranje samega zaslona. Ko je pin postavljen na nizko stanje, LCD ne registrira sprememb na R/W, RS ali podatkovnih pinih. V obratnem primeru registrira vse ukaze. Pin povežemo na poljuben digitalni priključek na arduinotu.

Pini od 7-14 so namenjeni prenosu podatkov. Zaslon deluje na dva načina – 8bit ali 4bit. V prvi konfiguraciji hkrati pošljamo po en bajt, v drugem primeru pa se en bajt prenese v dveh korakih. V našem primeru bomo uporabili kar 4bitno komunikacijo, saj si s tem prihranimo 4 dodatne proste pine na arduinotu. Pine povežemo na poljubne digitalne priključke na arduinotu.

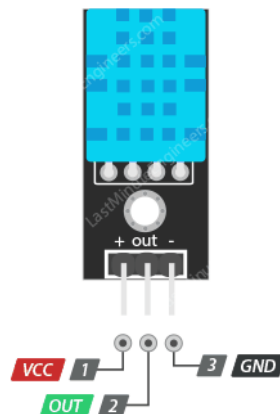
Zadnje dva pina sta A – anode, C – catode, služita pa za napajanje backlighta LCD-ja.

```
#include <LiquidCrystal.h>

// Za liquid display uporabimo 4bit komunikacijo (2, 3, 4, 5), register
// select pin (8), enable pin (7),
const int rs = 8, en = 7, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

void setup() {
  // Nastavimo LCD display na 16 stolpcev (char), ter 2 vrstice
  lcd.begin(16, 2);
  // Izpišemo na LCD
  lcd.print("Hello, World!");
}
```

Priklop senzorja za temperaturo ter vlago



Pin za napajanje-VCC in ground-GND ustrezno povežemo. Pin označen z OUT, ki nam bo služil za komunikacijo, pa povežemo na poljuben digitalni priključek na arduinotu.

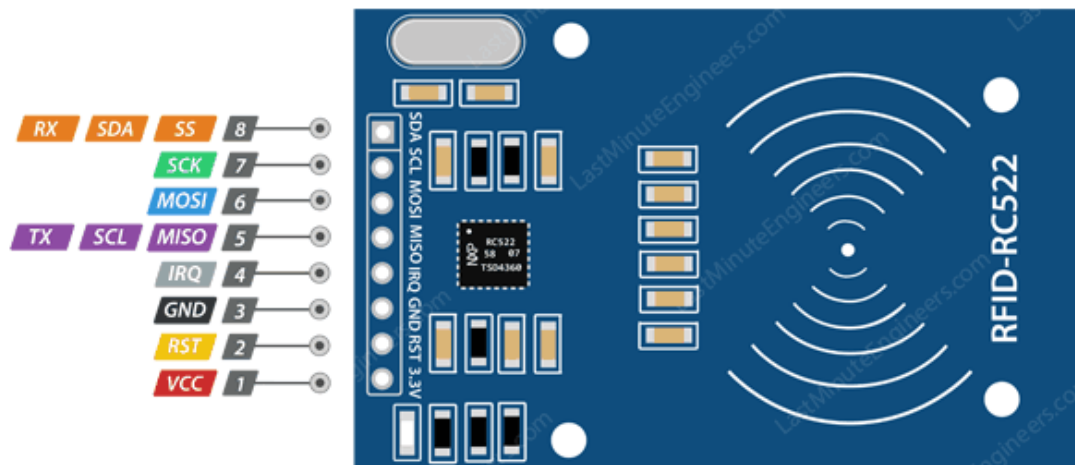
```

#include <dht.h>
// Definiramo pin za branje podatkov
#define DHT11_PIN      6
dht DHT;
void setup() {
    Serial.begin(9600);
}

void loop() {
    delay(2000);
    // Preberemo vrednost s tipala
    int chk = DHT.read11(DHT11_PIN);
    // Izpis vlage ter temperature
    Serial.print(DHT.temperature);
    Serial.print(DHT.humidity)
}

```

Priklop RFID bralnika – SPI komunikacija



Pine VCC in GND ustrezno povežemo.

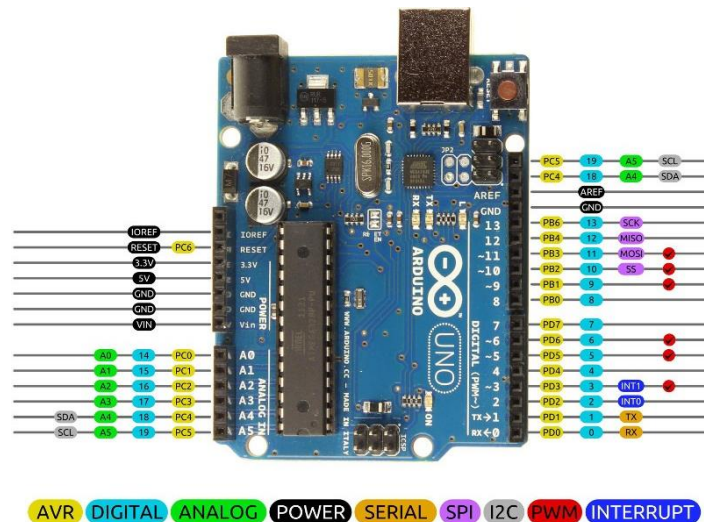
Drugi pin (označen z RST-reset) je namenjen za reset in izključitev samega bralnika. Ko na pin spustimo nizko stanje se bralnik izključi. Povežemo ga na poljubni digitalni priključek na arduinotu.

Četrti pin (označen z IRQ-interrupt) ustvari prekinitve na mikrokontrolerju ko se bralniku približa RFID tag.

S MISO in MOSI pinom, vzpostavimo komunikacijo slave naprave (RC522 modul) z master napravo (Arduino) in obratno. Peti pin (označen z MISO – master in slave out) služi kot SPI izhod RC522 modulu. Šesti pin (označen z MOSI – master out slave in) služi kot SPI vhod RC522 modulu.

Sedmi pin (označen z SCK – serial clock) sprejme urin signal, ki ga generira master – Arduino.

MISO, MOSI ter SCK priključek povežemo na digitalne priključke na arduinotu, ki služijo SPI komunikaciji.



V našem primeru (Arduino UNO), se SCK nahaja na pinu 13, MISO na pinu 12 ter MOSI na pinu 11. Prikazano tudi na zgornji sliki.

Osmi pin (označen z SS – signal input) – služi kot CS – chip select pri SPI komunikaciji. Pin povežemo na poljubni digitalni input.

```
#include <SPI.h>
#include <MFRC522.h>
// Reset pin za RFID scanner ter RFID signal input
#define RST_PIN          9
#define SS_PIN           10

// ID kartice, katero RFID scanner spusti naprej
String card = "87 66 71 62";
MFRC522 mfrc522(SS_PIN, RST_PIN);

void setup() {
    Serial.begin(9600);
    SPI.begin();
    // Inicializiramo RFID scanner
    mfrc522.PCD_Init();
}

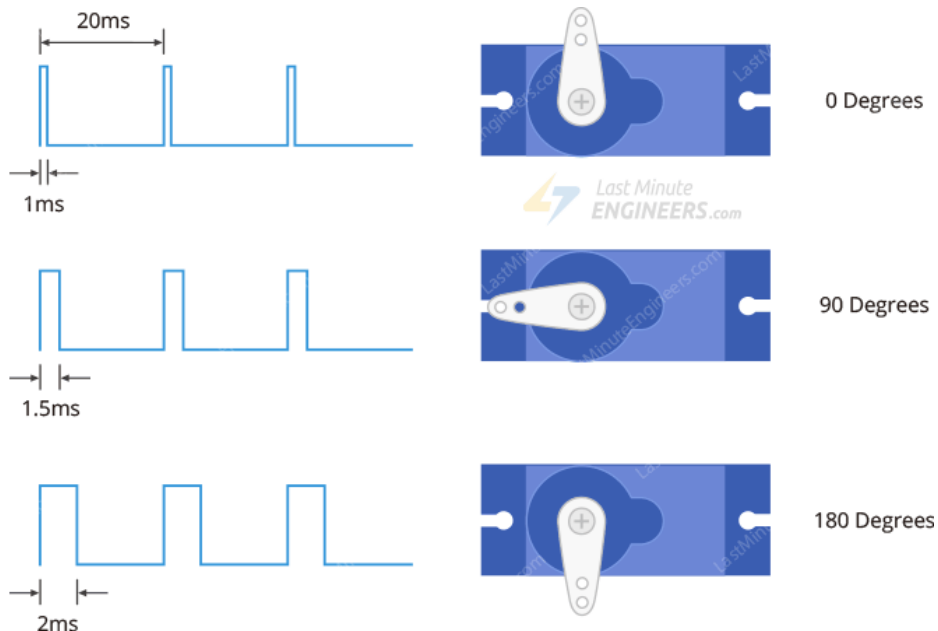
void loop() {
    // Preverimo če RFID zazna kartice in če lahko prebere ID kartice
    if (mfrc522.PICC_IsNewCardPresent()) {
        if (mfrc522.PICC_ReadCardSerial()) {
            //Poisčemo in zapišemo ID kartice v content
            String content= "";
            byte letter;
            for (byte i = 0; i < mfrc522.uid.size; i++)
            {
                content.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " "));
                content.concat(String(mfrc522.uid.uidByte[i], HEX));
            }
            // Postavimo ID v uppercase in preverimo če se ujema z String card
            content.toUpperCase();
            if (content.substring(1) == card)
            {
                Serial.print("CORRECT KEY!");
            }
            else {
                lcd.print("WRONG KEY!");
            }
        }
    }
}
```

Priklop servomotorja



Pine GND in 5V ustrezno povežemo.

Tretji pin služi kot input za kontroliranje sistema. Pin priporočljivo povežemo na digitalni input, ki omogoča pulzno-širinsko modulacijo.

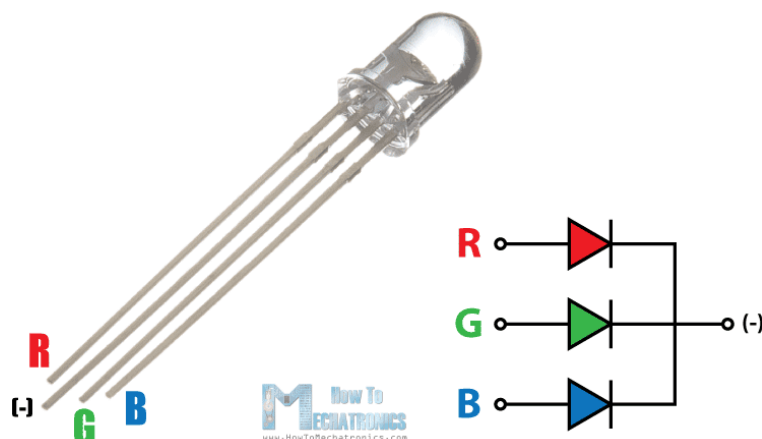


Pozicijo servo motor kontroliramo s serijo pulzov, ki jih pošiljamo na signalno linijo. Dolžina pulza nato določa pozicijo servo motorja. Če je pulz v visokem stanju 1ms, je servo motor obrnjen na 0 stopinj itd.

```
#include <Servo.h>
// Pin za komunikacijo s servo motorjem
#define servo 9
Servo Servom;

void setup() {
    // Inicializiramo servo motor in ga premaknemo na 0 stopinj
    Servom.attach(servo);
    Servom.write(0);
}
```

Priklop brenčača, LED diod ter foto upora



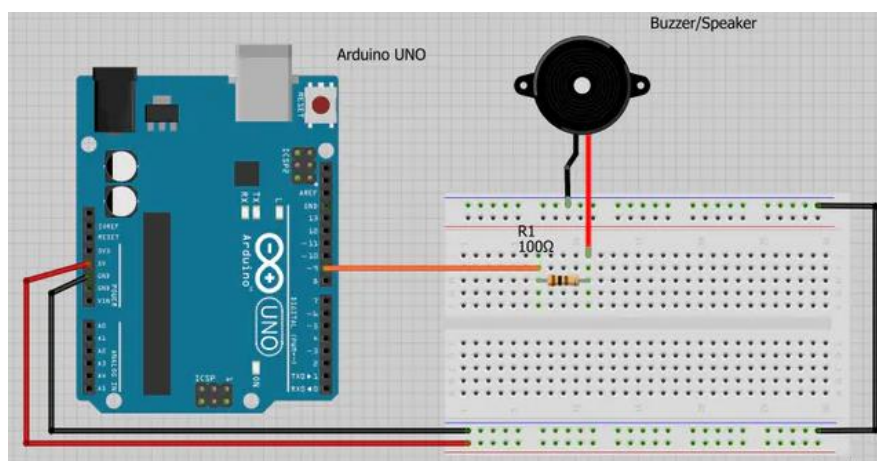
V celotno shemo povežemo še RGB LED diodo, ki bo svetila rdeče, če je motor obrnjen v zaklenjeno pozicijo (0 stopinj) in zeleno, če je obrnjen v odklenjeno pozicijo (90 stopinj). Potrebovali bomo le R in G del diode, oba pina pa povežemo na analogni input.

```
// Analogni pini za rdečo in zeleno barvo
#define red_light_pin  A4
#define green_light_pin A3

void setup() {
  //postavimo pinmode na output
  pinMode(red_light_pin, OUTPUT);
  pinMode(green_light_pin, OUTPUT);
}

void loop() {
  RGB_color(255, 0); // Red
  delay(1000);
  RGB_color(0, 255); // Green
  delay(1000);
}

//funkcija, ki nam olajša spreminjanje barve na diodi
void RGB_color(int red_light_value, int green_light_value)
{
  analogWrite(red_light_pin, red_light_value);
  analogWrite(green_light_pin, green_light_value);
}
```



Dodamo še brenčač, ki bo predvajal določen zvok, če bo kartica zavrnjena ali sprejeta. Plus pin brenčača povežemo na digitalni pin ki omogoča pulzno-širinsko modulacijo.

```
// Pin za aktivacijo brenčača
#define approve      9

void setup() {
  // Nastavimo pinmode na output
  pinMode(buzzer, OUTPUT);
}

void loop() {
  tone(buzzer, 1000); // Pošlji 1KHz zvočni signal...
  delay(1000);
  noTone(buzzer);
  delay(1000);
}
```

Priklopimo še navadno belo LED diodo ter foto upor. V primeru, da ima fotoupor dovolj veliko upornost (dovolj tema), vklopimo to LED diodo. Pri povezovanju pazimo, da foto upor povežemo na analogni priključek na arduinotu.

```
#define LEDOUTPUT      A1          // Pin output za led diodo
#define LIGHTINPUT     A0          // Pin input za fotosenzor

void setup() {
  // Nastavimo pin za foto tipalo kot input in LED diodo kot output
  pinMode(LIGHTINPUT, INPUT);
  pinMode(LEDOUTPUT, OUTPUT);
}

void loop() {
  // Preberemo vrednost foto tipala in ga mapiramo
  int LightVal = analogRead(LIGHTINPUT);
  int Light = map(LightVal, 10, 600, 0, 1400);
  Serial.println(Light);
  // Če je vrednost dovolj nizka oz. je v sobi tema, postavimo led na HIGH
  if(Light < 10) {
    digitalWrite(LEDOUTPUT, HIGH);
  }
}
```

Viri

<https://lastminuteengineers.com/>

<https://www.arduino.cc/>