

Documentație Simulearea unei mașini [Parcare]

Betea Adrian Vlad
Alăman Alexandru Petrișor
CTI-RO, anul II, grupa 1.1

Cuprins

- A. Componente utilizate
 - a. Tabel
 - b. Descriere
 - c. Pini utilizați
- B. Arhitectură Hardware
- C. Arhitectură Software
 - a. Schemă logică
 - b. Diagramă de stare
- D. Concluzii
 - a. Dificultăți
 - b. Optimizare

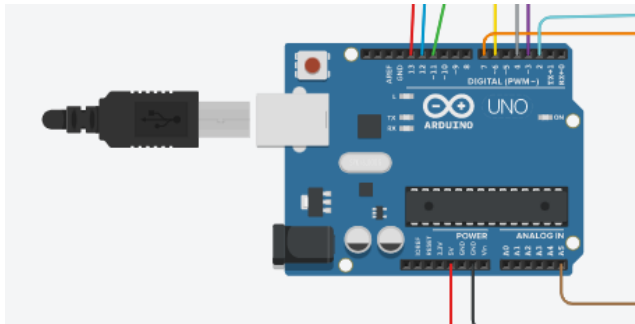
A. Componente utilizzate

 a. Tabel

Name	Quantity	Component
U1	1	Arduino Uno R3
D1	1	LED RGB
R1 R2 R3	3	1 kΩ Resistor
Rpot1	1	10 kΩ Potentiometer
D2	1	Red LED
R4	1	100 Ω Resistor
S1	1	Pushbutton
R5	1	10 kΩ Resistor
PIEZO1	1	Piezo (Buzzer)
PING1	1	Ultrasonic Distance Sensor
R6	1	200 Ω Resistor
LED1	1	NeoPixel Strip 8

b. Descriere

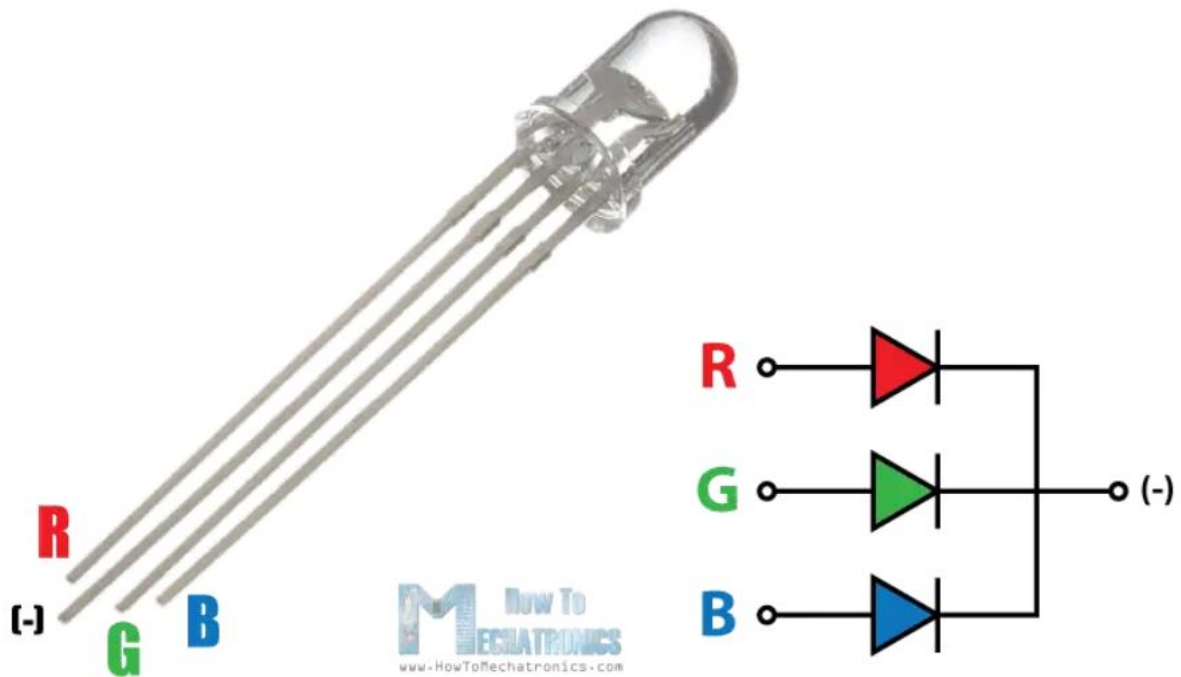
Arduino Uno 3



Board	Name	Arduino UNO R3
	SKU	A000066
Microcontroller	ATmega328P	
USB connector	USB-B	
Pins	Built-in LED Pin	13
	Digital I/O Pins	14
	Analog input pins	6
	PWM pins	6
Communication	UART	Yes
	I2C	Yes
	SPI	Yes
Power	I/O Voltage	5V
	Input voltage (nominal)	7-12V
	DC Current per I/O Pin	20 mA
	Power Supply Connector	Barrel Plug
Clock speed	Main Processor	ATmega328P 16 MHz
	USB-Serial Processor	ATmega16U2 16 MHz
Memory	ATmega328P	2KB SRAM, 32KB FLASH, 1KB EEPROM
Dimensions	Weight	25 g
	Width	53.4 mm
	Length	68.6 mm

LED RGB

- Folosit pentru a vedea starea mașini
 - o Verde – mașina e pornită
 - o Galben – mașina e în modul accesorii
 - o Roșu – mașina e oprită



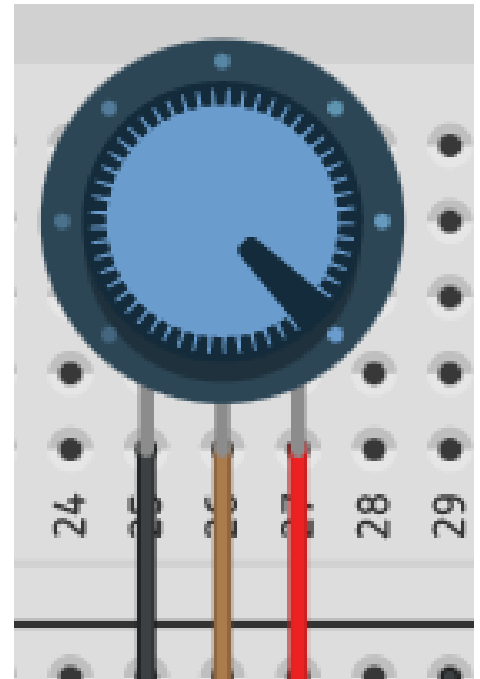
Rezistențe

- Folosite pentru a controla cât de multă electricitate trece prin LED-uri.



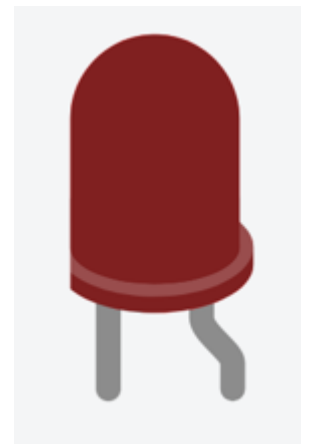
Potențiometru

- Folosit pentru a vedea starea motorului
 - 0-30% - oprit
 - 31-60% - accesorii
 - 61-100% - pornit



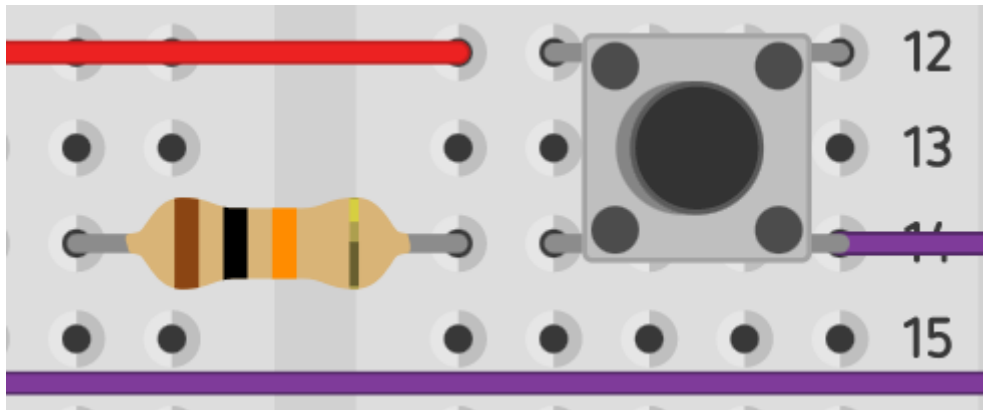
LED

- Folosit pentru a vedea starea funcției de parcare
 - Când e pornit funcția parcare e oprită
 - Când e oprit funcția de parcare e pornită

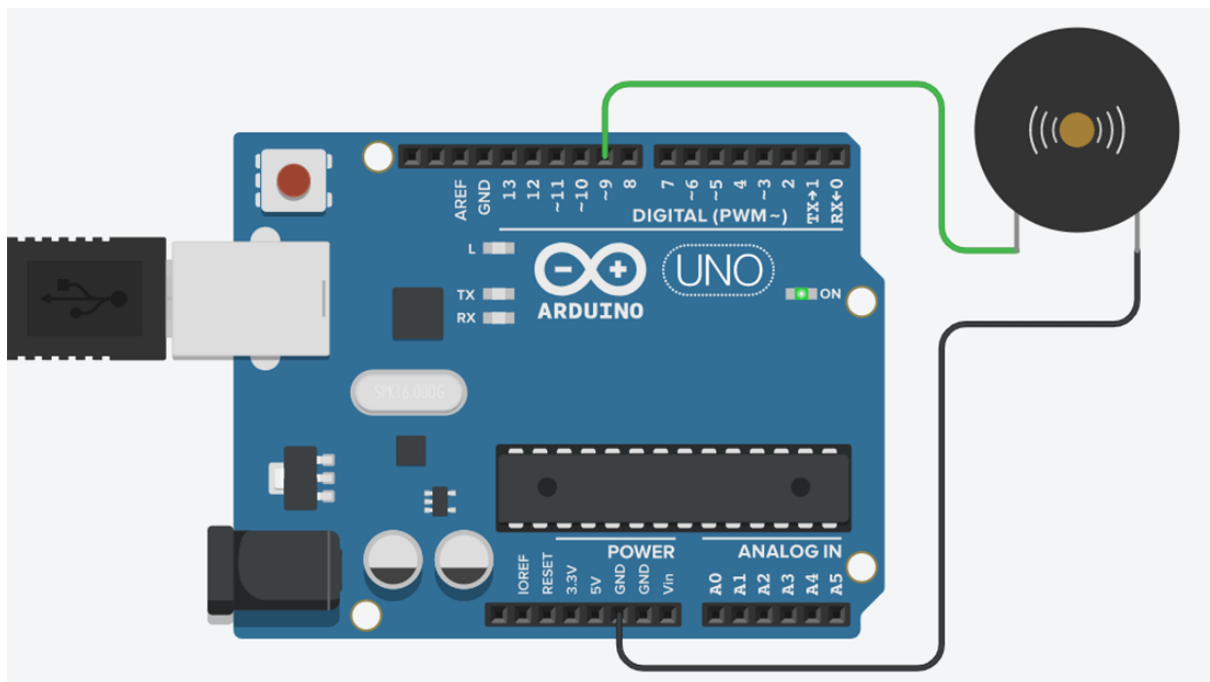


Buton

- Folosit pentru a porni/opri funcția de parcare

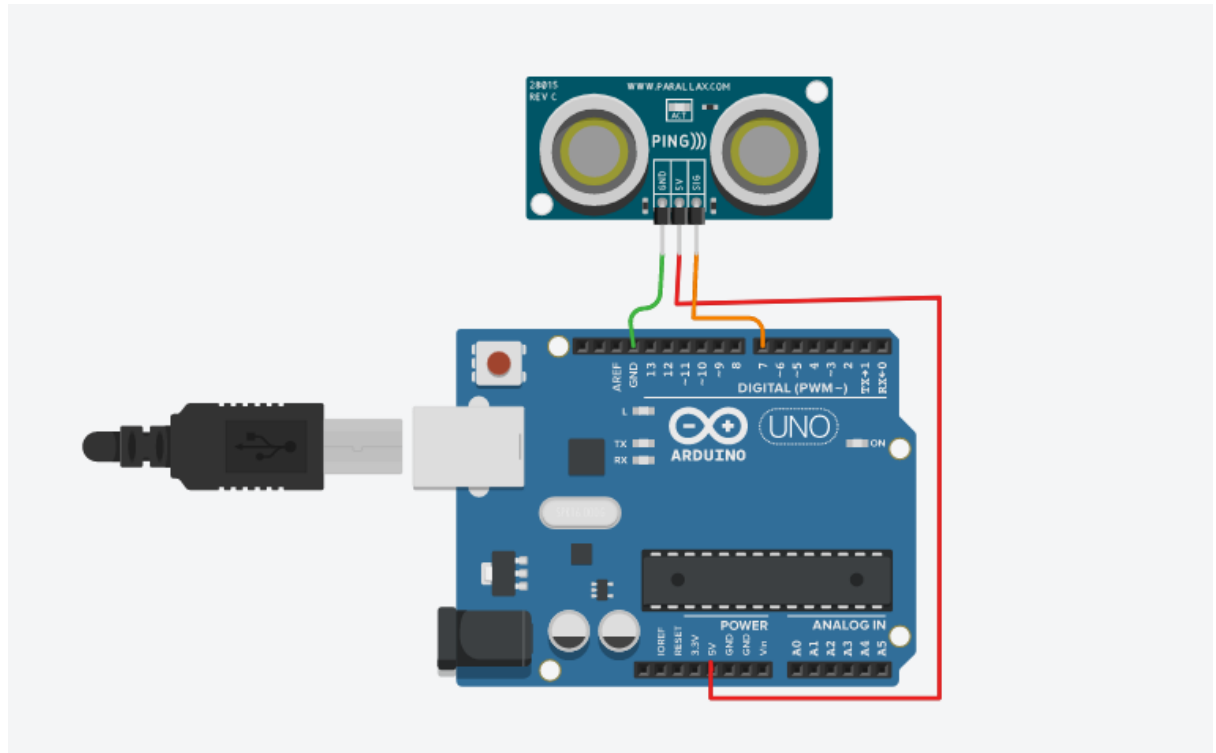


Piezo (buzzer)

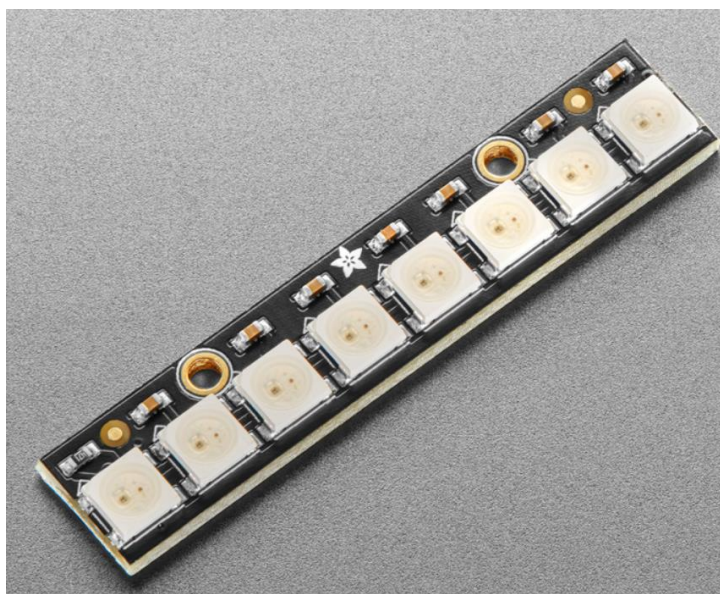


Senzor ultrasonic

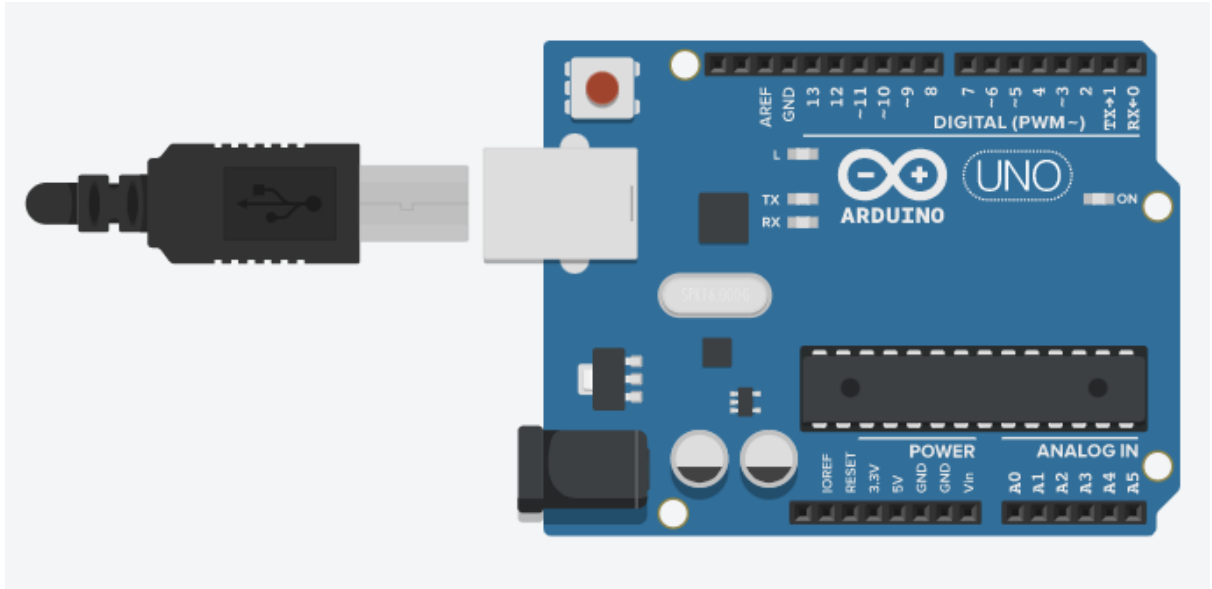
- Măsoară distanța dintre mașini



Neopixel strip



c. Pini utilizați



LED RGB (D1):

- PIN_RED = 13
- PIN_BLUE = 12
- PIN_GREEN = 11

POTENȚIOMETRU (Rpot1):

- POTENTIOMETER_PIN = A5 (are valori între 0-1023)

RED LED/Parking led (D2):

- PIN_PARKING_LED = 6

BUTTON (S1):

- PARKING_BUTTON = 3

PIEZO/BUZZER (PIEZO1):

- BUZZER_PIN = 4

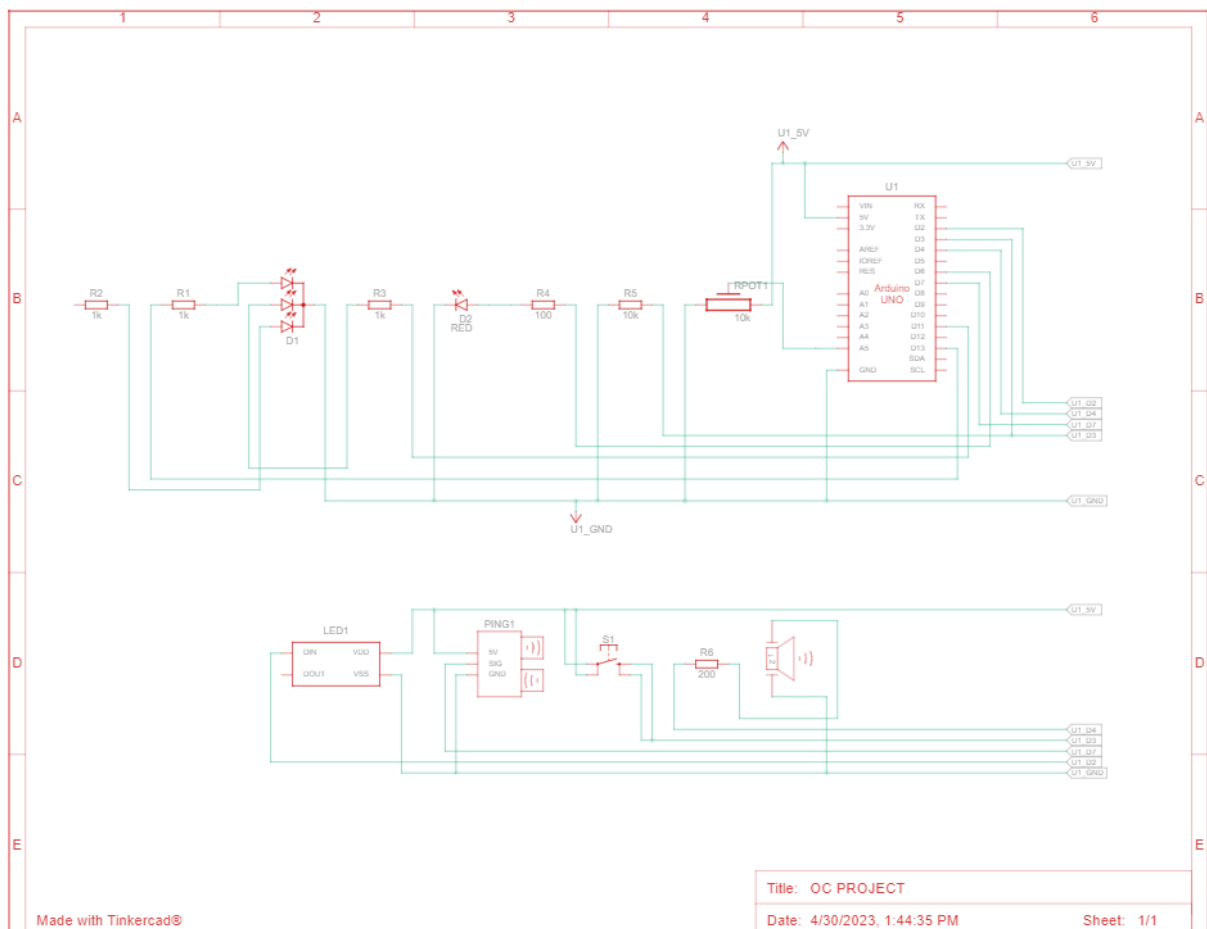
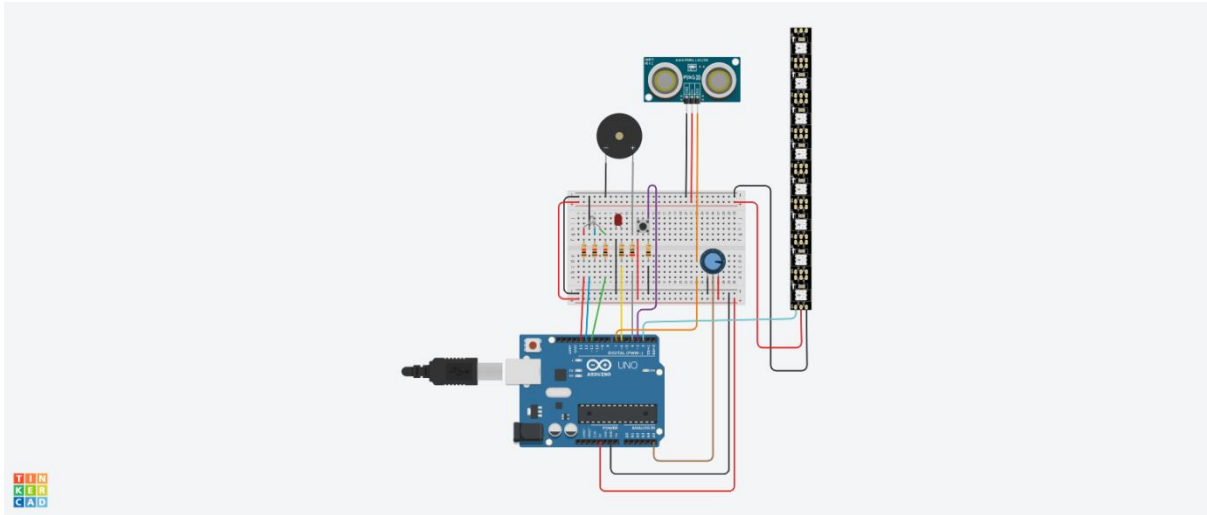
ULTRASONIC SENSOR (PING1):

- TriggerPin = 7

NEOPIXEL STRIP (LED1):

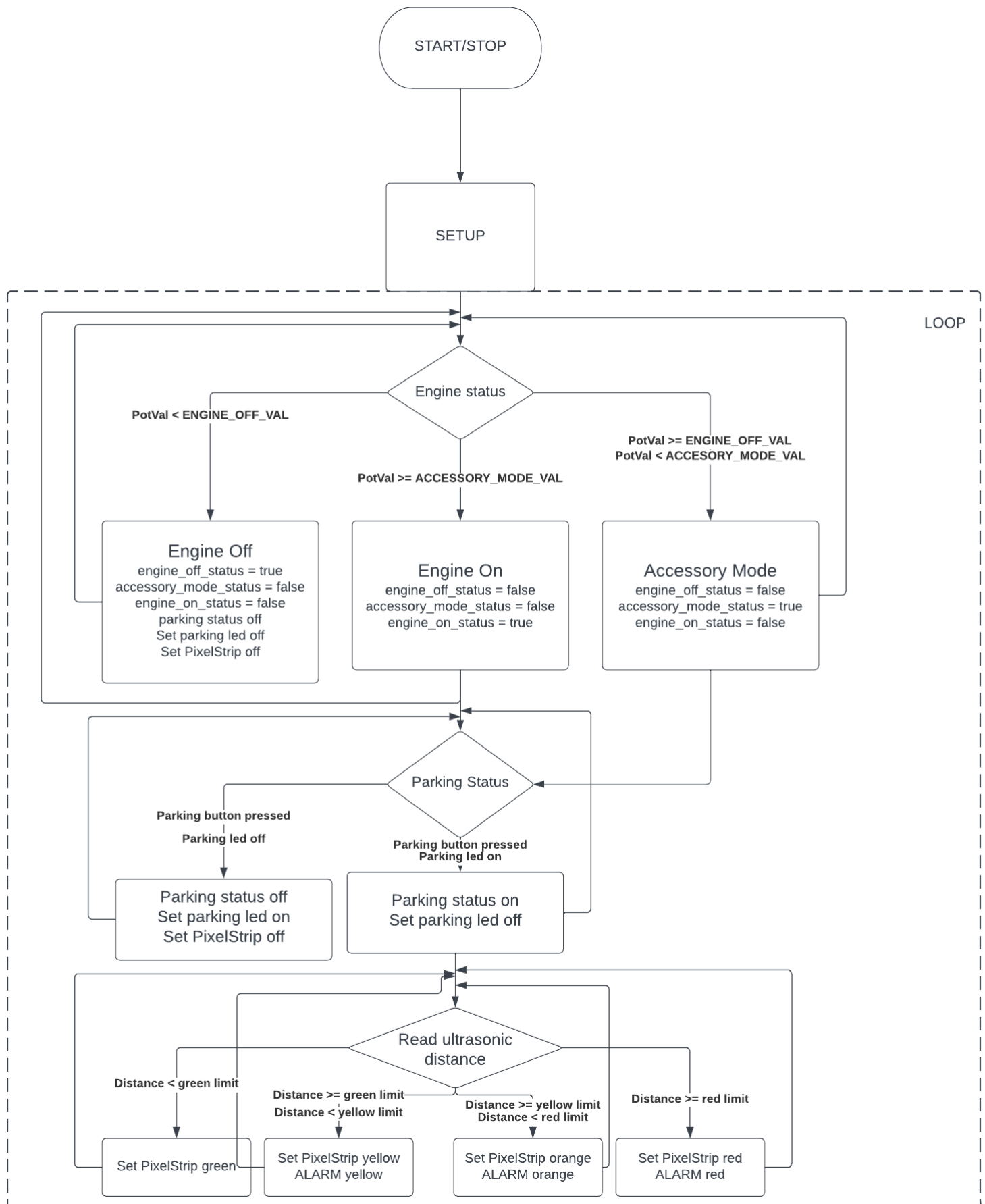
- PIXELS_PIN = 2

B. Arhitectura Hardware

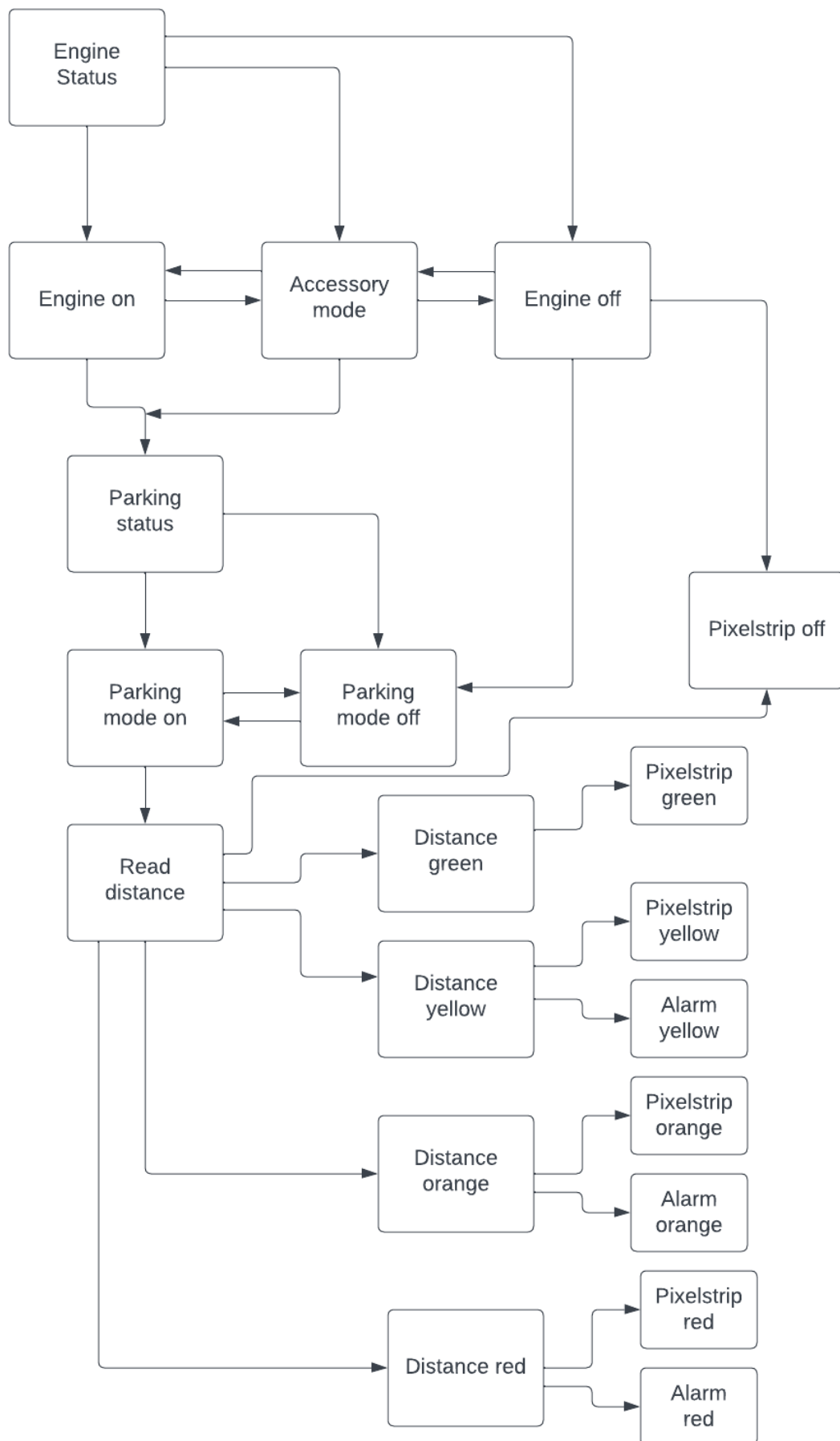


C. Arhitectura Software

a. Schemă logică



b. Diagramă de stare



D. Concluzii

a. Dificultăți

Am avut dificultăți la implementarea funcției alarmă, deoarece am vrut să evităm folosirea funcției `delay()` care opreste programul.

Pentru a nu folosi funcția `delay()` a trebuit sa folosim funcția `millis()`.

b. Optimizare

Pentru a optimiza codul am putea evita folosirea variabilelor de tip `int` (4 bytes). In locul acestora am putea folosi `uint8_t` (1 byte) sau `uint16_t` (2 bytes) pentru a ocupa mai putina memorie.

Funcții și librării folosite

- <Adafruit_NeoPixel.h>
 - Libraria aceasta este folosita de Neo Pixel Strip, pentru clasa Strip-ului si accesarea pin-ului si a colorilor.
 - Functia pixels acceseaza un anumit pixel/led din Strip
- Delay
 - Functie folosita pentru a intrerupe codul timp de n milisecunde
- analogWrite
 - functie folosita pentru a seta culorile led-urilor
- digitalWrite
 - functie folosita pentru a porni/opri led-urile
- pinMode
 - asociaza o variabila unui pin si ii seteaza folosinta(input/ouput)
- Serial.begin
 - Functie folosita pentru a putea printa valori in serial monitor
- Millis
 - Functie folosita pentru a evita folosirea functiei delay
 - Millis returneaza timpul curent in milisecunde

BIBLIOGRAFIE

Arduino UNO R3

<https://docs.arduino.cc/learn/programming/eeprom-guide>

LED RGB

<https://howtomechatronics.com/tutorials/arduino/how-to-use-a-rgb-led-with-arduino/>

RESISTOR

<https://learn.adafruit.com/adafruit-arduino-lesson-2-leds/resistors>

POTENTIOMETER

<https://docs.arduino.cc/learn/electronics/potentiometer-basics>

BUTTON

<https://docs.arduino.cc/built-in-examples/digital/Button>

BUZZER

<https://docs.arduino.cc/built-in-examples/digital/toneMelody>

ULTRASONIC SENSOR

<https://howtomechatronics.com/tutorials/arduino/ultrasonic-sensor-hc-sr04/>

NEOPIXEL STRIP

<https://arduinogetstarted.com/tutorials/arduino-neopixel-led-strip>