



Институт за математику и информатику  
Природно-математички факултет  
Крагујевац

Предмет:  
Микропроцесорски системи

Пројектни задатак:  
Систем за бројање саобраћаја и повезивање  
са LoRaWan мрежом

Ментор:  
Др Александар Пеулић

Студент:  
Поповић Лазар 84-2017

Крагујевац, август 2023.

## Садржај

Увод.....	4
Опис пројекта.....	6
Сврха пројекта.....	6
LoRaWan.....	6
Архитектура LoRaWan-а.....	7
Реализација пројекта.....	8
Конфигурација пинова и STM32CubeMX пројекат.....	10
Анализа firmware-а.....	10
Функције за приказ на LCD екрану.....	10
Функције за визуелни приказ преко LED диода.....	13
Обрада изузетака.....	16
Главна петља програма.....	16
Шематски дијаграм у Proteus 8 програму.....	18
Упутство за употребу система.....	19

# Увод

Овај пројекат се фокусира на развој и имплементацију симулације система за бројање саобраћаја користећи микроконтролер STM32 и сродне компоненте. Главна намена овог пројекта је да пружи реално и интерактивно искуство саобраћајног бројања, помажући разумевању и анализирању динамике саобраћаја на одређеном подручју. Ово је од суштинске важности за унапређење безбедности, планирање инфраструктуре и оптимизацију токова саобраћаја у различитим окружењима.

# Опис пројекта

## Сврха пројекта

Користећи техничке вештине и знања, овај пројекат ће имплементирати систем који симулира пролазак возила кроз систем за бројање. Централна компонента је микроконтролер STM32, који прима информације од различитих сензора, укључујући потенциометар за контролисање брзине симулираног кретања возила и дугме за покретање и заустављање симулације. Резултати се приказују на LCD екрану, а LED диоде омогућавају визуелну индикацију броја пролазака возила.

Кроз овај пројекат, истражују се различите могућности коришћења микроконтролера, сензора и других компоненти за имплементацију реалних система и њихове симулације. Истовремено, овај пројекат има потенцијал да пружи дубоко разумевање саобраћајних потреба и могућности урбаних подручја, доприносећи наравно и областима као што су инжењеринг безбедности саобраћаја, градско планирање и иновације у подручју интелигентног транспорта.

## LoRaWan

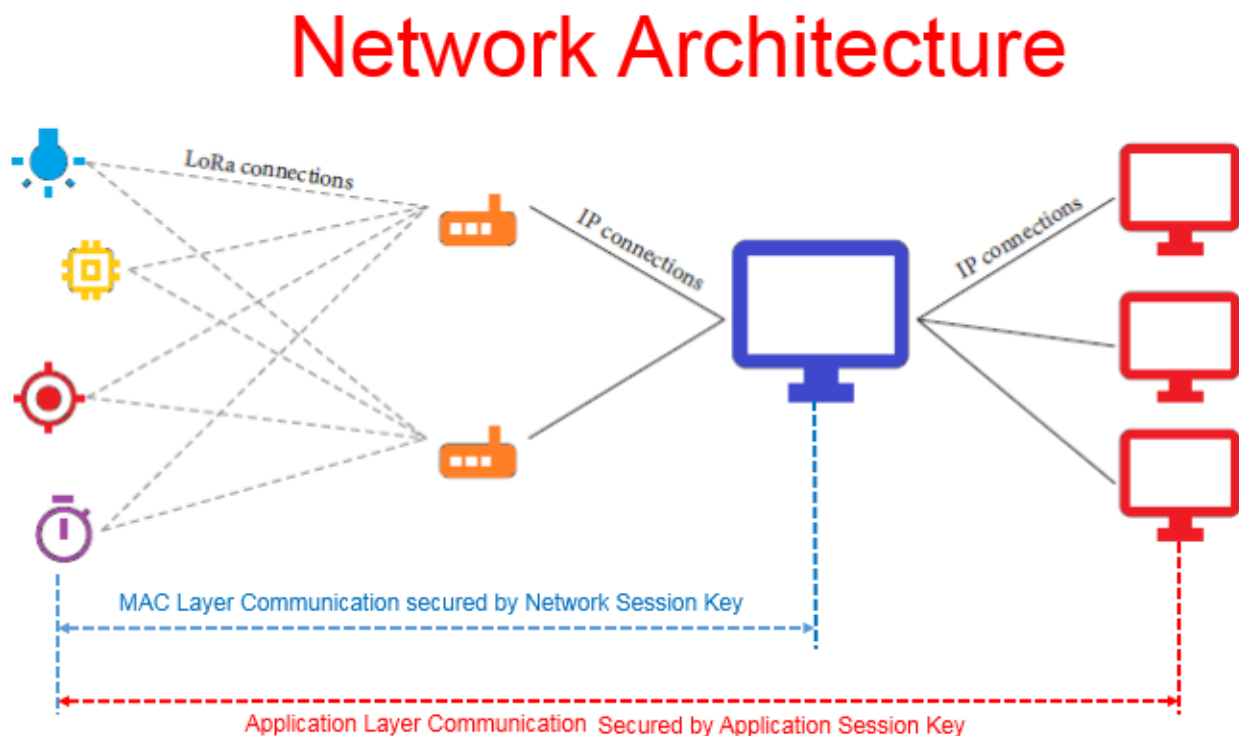
У данашњем времену, постоји изузетно велико интересовање за применом Интернет ствари (IoT) технологија, посебно у индустријским и саобраћајним системима. Кључни циљ ових технологија је смањење оперативних трошкова и постизање веће ефикасности. За овај поступак је неопходно извући податке из већ постојећих инфраструктурних и производних система који су у функцији већ годинама. Овај процес захтева не само бројне сензоре, већ и способност скупљања података у реалном времену или са минималним кашњењем, затим њихово обједињавање и анализу на нивоу целокупног система.

Да би се овај систем успешно реализовао, потребно је имати велики број сензора који се региструју на мрежи као јединствени извори стања система на месту где се сензор налази. Употреба LoRaWAN (Long Range Wide Area Network) мреже може бити решење за овакве изазове. LoRaWAN користи специфичну модулацију радиосигнала, познату као LoRa или

"Chirp" модулација, која је изузетно отпорна на интерференције. Ово је што га чини изузетно погодним за примену у креирању најпоузданијег бежичног система у својој категорији. LoRaWAN је основан на LoRa радио-модулацији и представља нискоенергетски мрежни протокол који омогућава безжично повезивање уређаја са Интернетом и управља комуникацијом између уређаја, крајњих чворова и мрежних гејтвеја. Мрежа се заснива на коришћењу нелецицираних ISM (индустријска, научна, медицинска) радио опсега за постављање комуникационе инфраструктуре. Овде се не ради само о радиоталасима, већ и о томе како они комуницирају са LoRaWAN гејтвејима како би обезбедили функције као што су шифровање и идентификација.

Мрежна архитектура LoRaWAN заснована је на звезданој топологији, где гејтвеји преносе поруке између крајњих уређаја и централних мрежних сервера. Протокол и мрежна архитектура имају велики утицај на трајање батерије уређаја, капацитет мреже, квалитет услуге, безбедност и разноврсност примена које мрежа може обављати.

## Архитектура LoRaWan-a



Архитектура LoRa (Long Range) система се састоји из следећих елемената:

### **1. Сензори и Уређаји:**

Ови уређаји, који се називају и крајњим уређајима, имају улогу да прикупљају различите врсте података из свог окружења. То могу бити температура, влажност, осветљеност, присуство, бројачи итд. Сензори користе нискоенергетски радио модул да би проследили податке.

### **2. Gateways (Гејтвеји):**

Гејтвеји су промежни уређаји између крајњих уређаја и мреже. Они прихватају податке које шаљу сензори путем радио таласа и преносе их до мрежног сервера. Гејтвеји такође обављају превођење фреквенције и модулације, што омогућава комуникацију између различитих технологија и уређаја.

### **3. Мрежни сервери:**

Ови сервери прихватају податке које су проследили гејтвеји. Подаци се сабирају и обрађују на централном мрежном серверу. Одатле се подаци могу проследити ка апликацијама или системима који користе те податке за анализу, визуализацију или донешење одлука.

### **4. LoRaWAN Мрежа:**

LoRaWAN (Long Range Wide Area Network) је спецификација мрежног протокола који омогућава паметну комуникацију између сензора, гејтвеја и мрежних сервера. Ова технологија користи "LoRa" модулацију која је специјално дизајнирана за дугачке дистанце и ниску потрошњу енергије.

Крајњи уређаји комуницирају са гејтвејима путем LoRa модулације, што омогућава широк опсег комуникације. Гејтвеји обрађују и прослеђују податке мрежном серверу који се налази на облаку или локалном серверу. Одатле се подаци анализирају, визуализују и користе у различитим апликацијама за побољшање оперативних процеса, уштеду енергије и донешење бољих одлука.

## **Реализација пројекта**

За овај пројекат, коришћена је STM32F103C6Tx микроконтролер плоча. Систем је дизајниран за бројање саобраћаја помоћу различитих компоненти, укључујући потенциометар, LED диоде и дугме.

Потенциометар се користи за симулирање брзине пролаза возила у симулацији.

Коришћењем потенциометра, корисник може да контролише брзину којом се симулирају пролази возила кроз систем.

Дугме се користи за покретање и заустављање симулације. Када се дугме притисне, систем се покреће и почиње са симулирањем пролаза возила. Поновним притиском на дугме, симулација се зауставља.

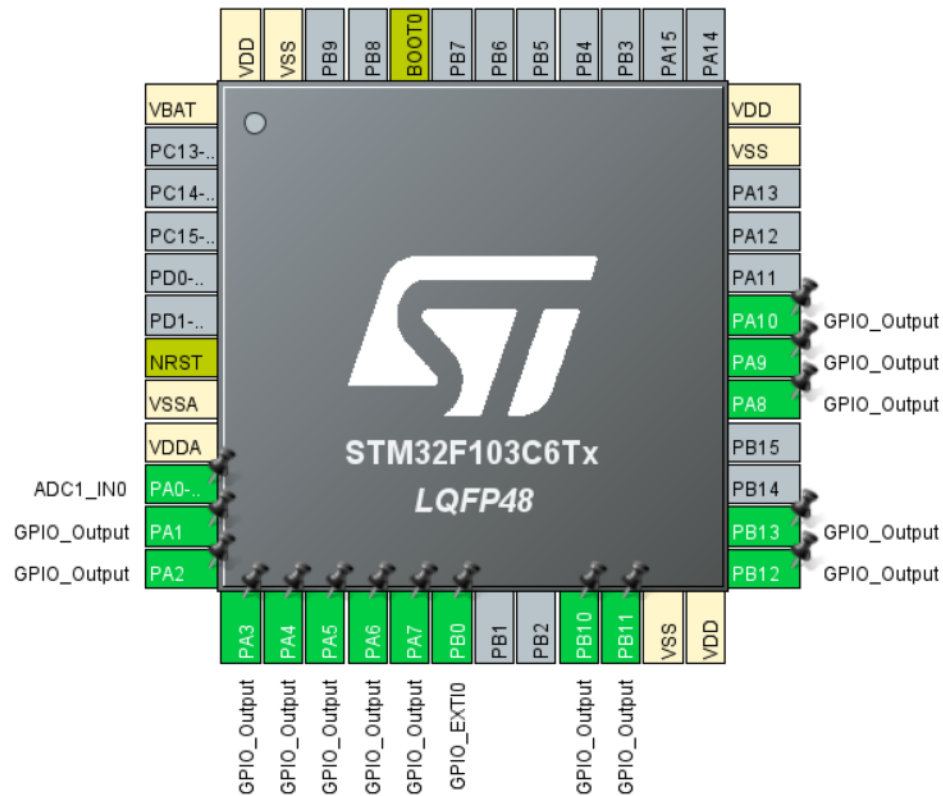
За приказ информација о броју возила која су прошла, коришћен је LCD дисплеј. На дисплеју се приказује текући број возила која су прошла кроз систем.

Коришћењем LED диода, индикован је текући број возила на путу. За сваки опсег броја возила, постављена је одговарајућа LED диода која се укључује када се достигне одговарајући број прошлих возила. На пример, зелена LED диода се укључује када је број возила мањи од 10, црвена када је број возила између 10 и 20, а жута када је број возила већи од 20.

Теоретски, уз помоћ LoRa модула, подаци о броју возила и стању симулације могу се безжично преносити на друге уређаје или сервере, омогућавајући праћење саобраћаја и анализу података у реалном времену. Ова технологија има потенцијал да се примени у различитим окружењима, укључујући градске инфраструктуре и индустријске системе, са циљем оптимизације управљања саобраћајем и ресурсима.

За реализацију firmware-а користе се алати STM32CubeMX I STM32CubeIDE, као и алат Proteus 8 који служи за дизајнирање електричних шема као и симулацију услед недостатка реалног хардвера.

## Конфигурација пинова и STM32CubeMX пројекта



Коришћени пинови:

РА0 – пин који је коришћен за аналогно-дигиталну конверзију

PA1-PA10 – пинови у output режиму, коришћени за приказ на LCD екрану

РВ0 – пин који користимо при обради прекида када се кликне дугме

PB10-PB13 – пинови у output режиму, коришћени за приказ преко диода

## Анализа firmware-a

## Функције за приказ на LCD екрану

### LCD() функција:



```

67 void LCD(uint8_t value, uint8_t command)
68 {
69     uint8_t data;
70
71     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_8, command);
72
73     data = value & 0x01;
74     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_10, data);
75
76     data = (value >> 1) & 0x01;
77     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_1, data);
78
79     data = (value >> 2) & 0x01;
80     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_2, data);
81
82     data = (value >> 3) & 0x01;
83     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_3, data);
84
85     data = (value >> 4) & 0x01;
86     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, data);
87
88     data = (value >> 5) & 0x01;
89     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, data);
90
91     data = (value >> 6) & 0x01;
92     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6, data);
93
94     data = (value >> 7) & 0x01;
95     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_7, data);
96
97     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_9, GPIO_PIN_SET);
98     HAL_Delay(50);
99     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_9, GPIO_PIN_RESET);
100 }

```

Функција LCD се користи за слање команди или података на LCD дисплеј користећи сет GPIO пинова на микроконтролеру. Параметар “command” одређује да ли су у питању команде или стварни подаци за приказ, а параметар “value” садржи податке или команду која ће бити послата. Функција се брине о постављању одговарајућих линија података на стање укључено и искључено, покретању LCD дисплеја да прочита податке и увођењу неопходних кашњења за правилну комуникацију са LCD дисплејом.

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_8, command):  
Ова линија поставља стање одређеног GPIO пина (GPIO\_PIN\_8) на порту GPIOA у зависности од параметра “command”. Ако је “command” 1, пин се поставља на HIGH; ако је “command” 0, пин се поставља на LOW. Ово се користи за контролисање да ли се шаље команда или стварни подаци за приказ.

Следи низ линија data = value & 0x01 до data = (value >> 7) & 0x01; које издвајају појединачне битове из бајта “value” и смештају их у променљиву “data”. Ови битови одговарају подацима који се шаљу на LCD дисплеј за сваку од линија података (D0 до D7).

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_9, GPIO\_PIN\_SET):

Ова линија поставља други GPIO пин (GPIO\_PIN\_9) на порту GPIOA на HIGH. Овај пин се често користи да би се покренуло читање података са линија података на LCD дисплеју.

HAL\_Delay(50):

Ова линија уводи паузу од 50 милсекунди, што омогућава LCD дисплеју да прочита податке са линија и обради команду или податке.

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_9, GPIO\_PIN\_RESET):

Ова линија поставља исти GPIO пин (GPIO\_PIN\_9) назад на LOW. Овај корак помаже у завршавању процеса преноса података.

### InitializeLCDScreen() функција:

```
103 void InitializeLCDScreen()  
104 {  
105     LCD(0x38,0);  
106     LCD(0x0C,0);  
107     LCD(0x06,0);  
108     LCD(0x01,0);  
109     LCD(0x80,0);  
110 }
```

Ова функција шаље низ команди на LCD екран у циљу конфигурације поставке приказа и да припреми екран за операције писања. Ове команде обезбеђују да се LCD постави за приказ у режиму са 2 линије, са скривеним курсором у сврху приказивања одређених података на дисплеју.

### UpdateVehicleCountLCD() функција:

```
150 void updateVehicleCountLCD()  
151 {  
152     InitializeLCDScreen();  
153     char buffer[16];  
154     snprintf(buffer, sizeof(buffer), "Vozila: %u", vehicleCount);  
155     LCD(0x01, 0);  
156     for (int i = 0; buffer[i]; i++) {  
157         LCD(buffer[i], 1);  
158     }  
159 }
```

Ова функција служи за ажурирање LCD екрана са информацијом о броју возила која пролазе:

1. `InitializeLCDScreen()`: Позива претходно дефинисану функцију `InitializeLCDScreen` која иницијализује ЛЦД екран постављајући почетне параметре за приказ.
2. `char buffer[16];`: Дефинише низ карактера (стринг) под називом `buffer` са капацитетом од 16 карактера. Овај низ служи за смештање информације о броју возила која ће се приказати на екрану.
3. `snprintf(buffer, sizeof(buffer), "Vozila: %u", vehicleCount);`: Користи функцију `snprintf` да форматира информацију о броју возила (`vehicleCount`) у облику "Vozila: X", где је X број возила. Резултат се смешта у низ `buffer`.
4. `LCD(0x01, 0);`: Брише приказ са LCD екрана и поставља курсор на почетну позицију.
5. `for (int i = 0; buffer[i]; i++) { LCD(buffer[i], 1); }`: Користи петљу да прође кроз све карактере у низу `buffer` и за сваки карактер позива функцију `LCD` са одговарајућим карактером и параметром 1 који означава да се ради о подацима који се приказују (не командама).

На крају ове функције, LCD екран ће бити ажуриран са новим информацијама о броју возила који се приказује на екрану.

## Функције за визуелни приказ преко LED диода

Систем је конципиран тако да, поред визуелног приказа броја возила која су прошла кроз систем на LCD дисплеју, имамо и визуелне ознаке броја возила која су прошла користећи LED диоде у различитим бојама. Ако је прошло мање од 10 возила, светли зелена диода, уколико је број возила између 10 и 20, светли црвена диода а уколико је тај број већи од 20, светли мешавина жуте и плаве диоде. На овај начин имамо сигнални приказ броја возила, који је из даљине видљив и врло интуитиван по питању представљања броја возила чији је пролазак регистрован. Ово је постигнуто следећим функцијама:

```
121 void turnOnGreenLED()  
122 {  
123     HAL_GPIO_WritePin(GPIOB, GPIO_PIN_10, GPIO_PIN_RESET);  
124     HAL_GPIO_WritePin(GPIOB, GPIO_PIN_11, GPIO_PIN_SET);  
125     HAL_GPIO_WritePin(GPIOB, GPIO_PIN_12, GPIO_PIN_SET);  
126 }  
127
```

Пали се зелена диода, док су црвена и плава угашене.

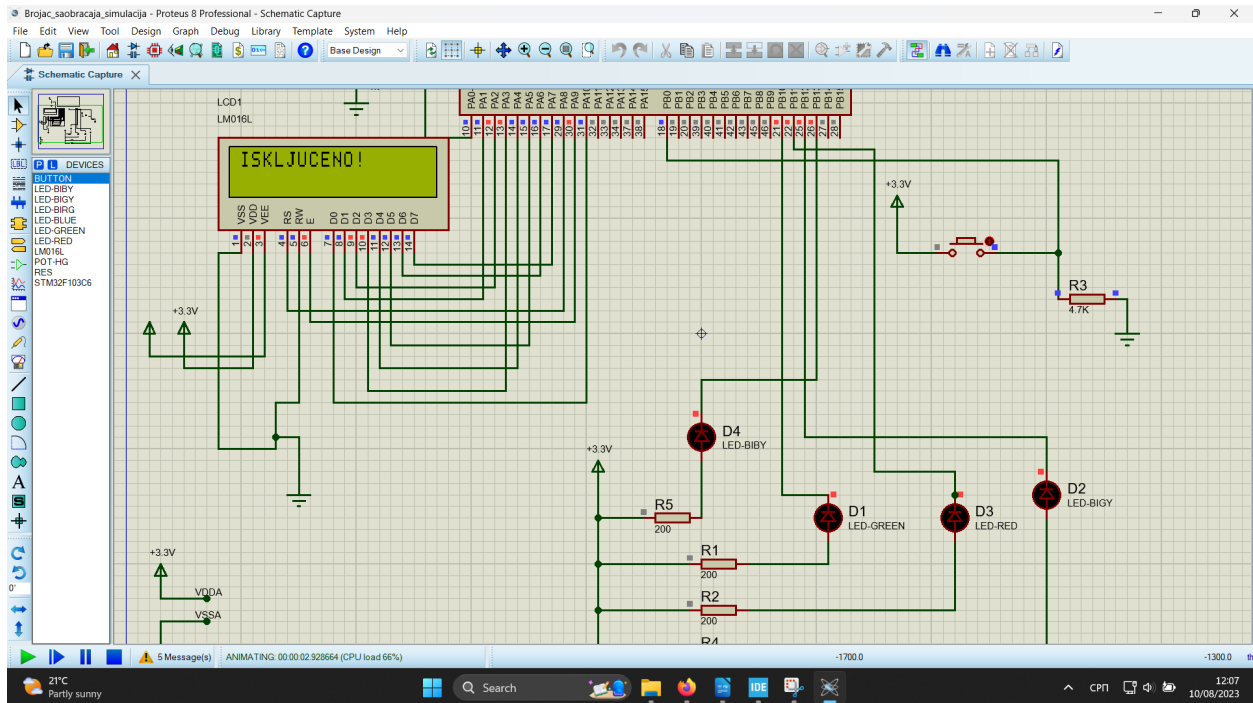
```
128 void turnOnRedLED()
129 {
130     HAL_GPIO_WritePin(GPIOB, GPIO_PIN_10, GPIO_PIN_SET);
131     HAL_GPIO_WritePin(GPIOB, GPIO_PIN_11, GPIO_PIN_RESET);
132     HAL_GPIO_WritePin(GPIOB, GPIO_PIN_12, GPIO_PIN_SET);
133 }
```

Пали се црвена диода, док су зелена и плава угашене.

```
135 void turnOnBlueLED()
136 {
137     HAL_GPIO_WritePin(GPIOB, GPIO_PIN_10, GPIO_PIN_SET);
138     HAL_GPIO_WritePin(GPIOB, GPIO_PIN_11, GPIO_PIN_SET);
139     HAL_GPIO_WritePin(GPIOB, GPIO_PIN_12, GPIO_PIN_RESET);
140 }
```

Пали се плава диода, док су црвена и зелена угашене.

Такође имамо и функцију за гашење свих диода у складу са тим да ли је систем покренут или не, уколико је систем угашен а симулација тече, на LCD дисплеју ће бити исписана порука искључено и све диоде ће бити угашене:



Искључивање диода постижемо следећом функцијом:

```

142 void turnAllLEDsOff()
143 {
144     HAL_GPIO_WritePin(GPIOB, GPIO_PIN_10, GPIO_PIN_SET);
145     HAL_GPIO_WritePin(GPIOB, GPIO_PIN_11, GPIO_PIN_SET);
146     HAL_GPIO_WritePin(GPIOB, GPIO_PIN_12, GPIO_PIN_SET);
147     HAL_GPIO_WritePin(GPIOB, GPIO_PIN_13, GPIO_PIN_SET);
148 }

```

Приликом рада система, додата је и диода која се пали и гаси приликом проласка аутомобила кроз сензор за бројање(у реалном сценарију употребе можемо претпоставити да би се користио инфрацрвени сензор). Сврха ове диоде је да да визелни приказ исправности рада система као и да заправо симулира пролазак возила кроз систем.Брзина трептања диоде зависи од одабраног напона на потенциометру. Потенциометар нам служи за контролу брзину и интензитета којом возила пристижу ка систему за бројање. Ово постижемо следећом функцијом:

```

161 // Ova funkcija simulira prolazak u skladu sa naponom na potencijometru
162 void simulatePassage(uint32_t delay_time) {
163
164     HAL_GPIO_WritePin(GPIOB, GPIO_PIN_13, GPIO_PIN_RESET);
165     HAL_Delay(delay_time);
166     HAL_GPIO_WritePin(GPIOB, GPIO_PIN_13, GPIO_PIN_SET);
167 }

```

## Обрада изузетака

Приликом покретања симулације, систем је угашен и на дисплеју је исписана порука „ИСКЉУЧЕНО“ и све диоде су угашене. Систем се покреће кликом на дугме, и том приликом почиње бројање возила, на дисплеју се исписује порука у укупном броју возила која су прошла и светли одговарајућа диода. Поновним кликом на дугме, систем се гаси и исписује се одговарајућа порука. Ово је постигнуто имплементацијом одговарајуће методе која обрађује прекид као и метода у главној петљи програма:

```
112 void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
113 {
114     if (GPIO_Pin == GPIO_PIN_0)
115     {
116         simulationRunning = !simulationRunning;
117     }
118 }
119 }
```

## Главна петља програма

```

201  turnAllLEDsOff();
202  InitializeLCDScreen();
203
204  /* USER CODE END 3 */
205  while (1) {
206      if (simulationRunning)
207      {
208          HAL_ADC_Start(&hadc1);
209          HAL_ADC_PollForConversion(&hadc1, HAL_MAX_DELAY);
210          uint16_t potentiometer_value = HAL_ADC_GetValue(&hadc1);
211          HAL_ADC_Stop(&hadc1);
212
213          uint32_t delay_time = (potentiometer_value * 1000) / 4095;
214
215          // Simuliramo prolazak vozila
216          if (vehicleCount <= 10)
217          {
218              // Manje od 11 vozila je proslo
219              turnOnGreenLED();
220
221              vehicleCount++;
222              updateVehicleCountLCD();
223
224              simulatePassage(delay_time);
225          }
226          else if (vehicleCount >= 11 && vehicleCount <= 20)
227          {
228              // 20 - 50 vozila je proslo
229              turnOnRedLED();
230
231              vehicleCount++;
232              updateVehicleCountLCD();
233
234              simulatePassage(delay_time);
235          }

```

Пре покретања главне петље вршимо почетне иницијализације дисплеја и диода а затим се проверава променљива „simulationRunning” која представља ознаку да ли је систем покренут или не а којом манипулишемо преко функције за обраду изузетака.

Уколико је систем покренут, врши се аналогно-дигитална конверзија једног канала и учитавање „raw“ вредности у променљиву “potentiometer\_value” а потом се врши калкулација брзине симулације брзине пристизања возила ка систему за бројање и то је представљено променљивом „delay\_time”.

Након тога, у зависности од броја возила која су прошла до сада, врши се паљење одговарајућих метода, инкрементирање броја возила која су прошла као приказивање тог броја на LCD дисплеју.

```

236         else
237         {
238             // Vise od 50 vozila je proslo
239             turnOnBlueLED();
240
241             vehicleCount++;
242             updateVehicleCountLCD();
243
244             simulatePassage(delay_time);
245         }
246     }
247     else
248     {
249         // Sistem je iskljucen
250         InitializeLCDScreen();
251         char buffer[16];
252         snprintf(buffer, sizeof(buffer), "ISKLJUCENO!");
253         LCD(0x01, 0);
254         for (int i = 0; buffer[i]; i++)
255         {
256             LCD(buffer[i], 1);
257         }
258         vehicleCount = 0;
259         turnAllLEDsOff();
260     }
261 }
262 }

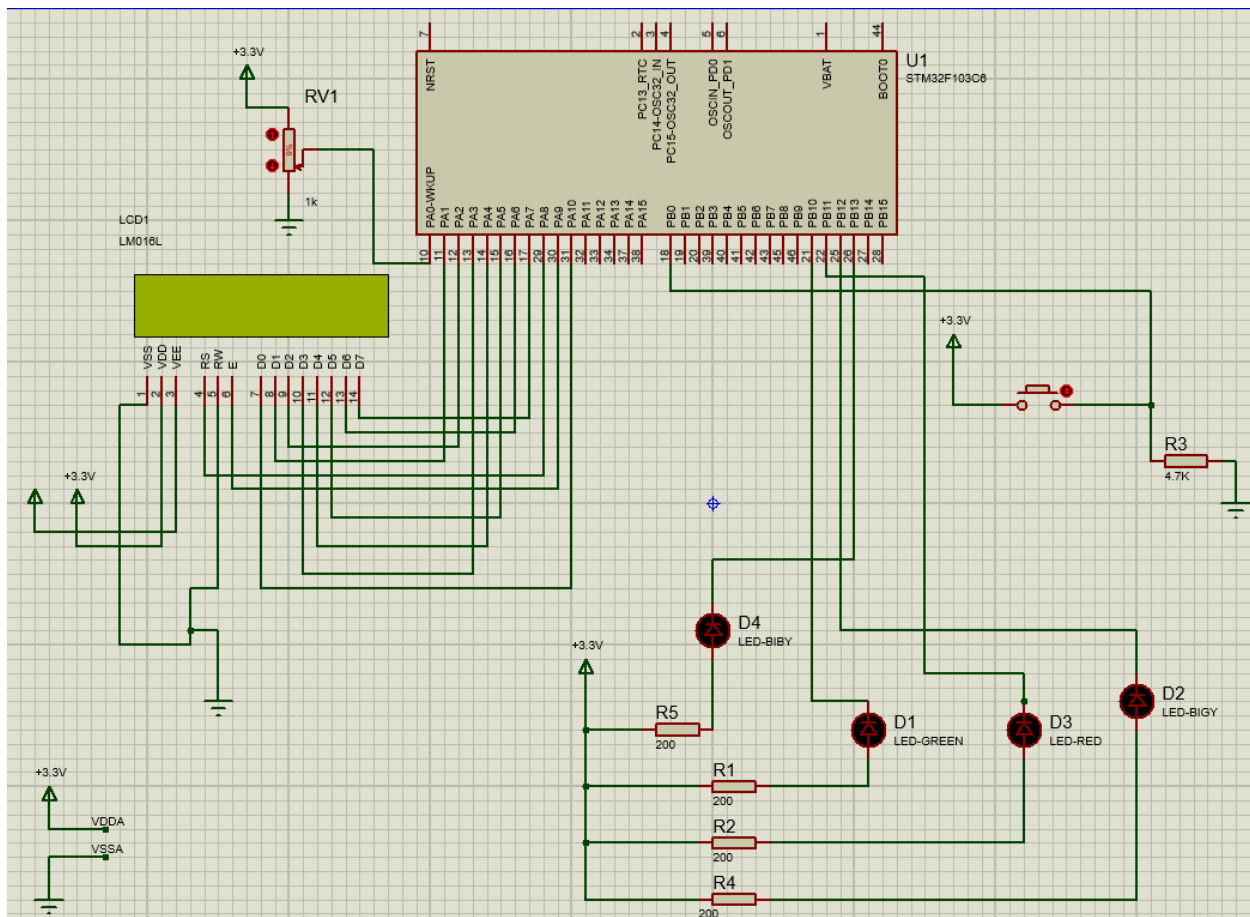
```

Уколико је кликнуто дугме за гашење система, на дисплеј се шаље порука „ИСКЉУЧЕНО“, ресетује се број возила који је прошао и гасе се све диоде.

## Шематски дијаграм у Proteus 8 програму

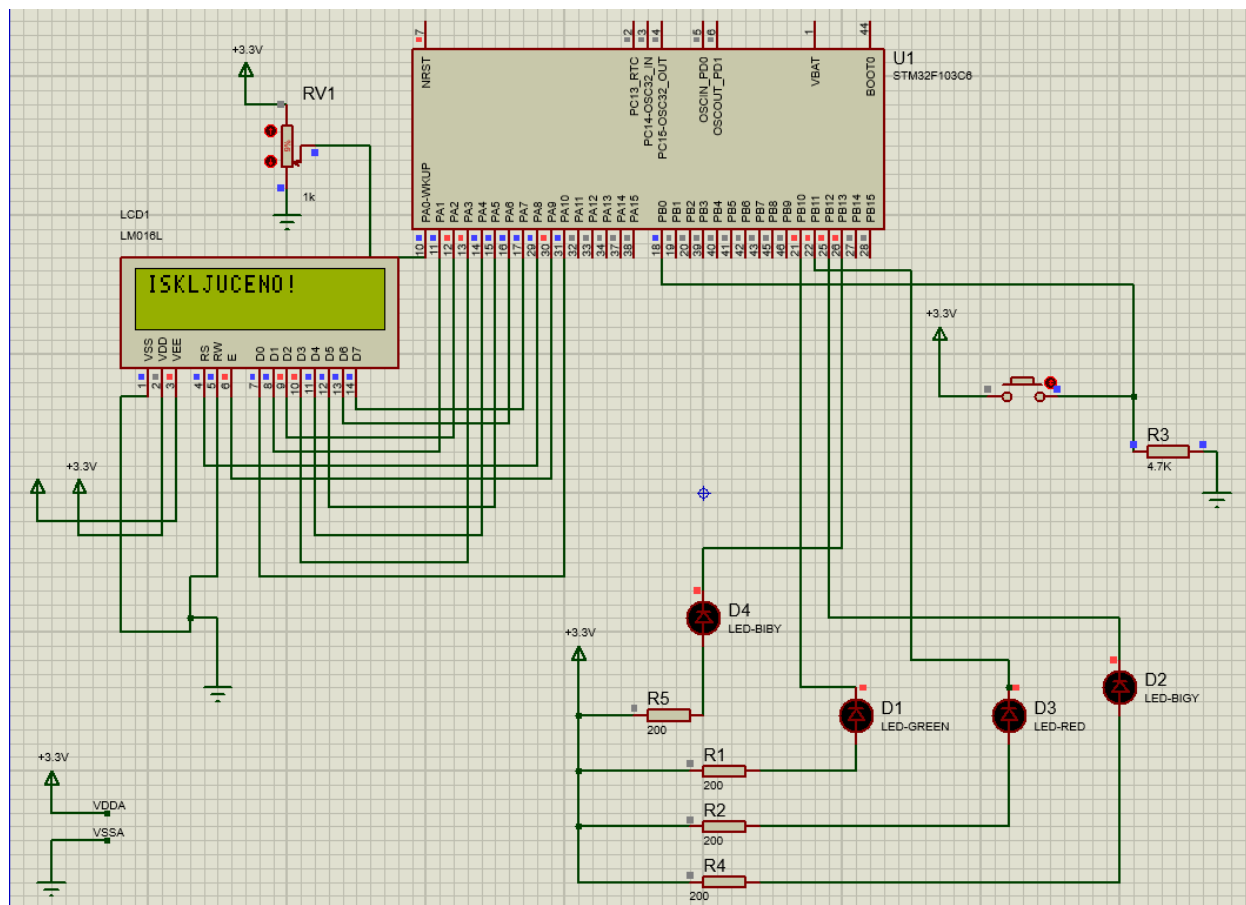
- Дугме – Дугме служи као прекид, све док дугме није притиснуто, ситем неће вршити бројање, притиском на дугме покрећемо бројање.
- Лед диоде – Зелена лед диода означава да мање од 10 возила прошло кроз систем, зелена означава да је прошло између 10 и 20, док плава означава број већи од 20.
- LCD – Приказује тренутан број возила која су прошла кроз ситем.
- Потенциометар – Симулира рад инфрацрвеног сензора са аналогним излазом за регистровање проласка возила као и за симулацију брзине и интензитета којом возила пристижу ка систему.



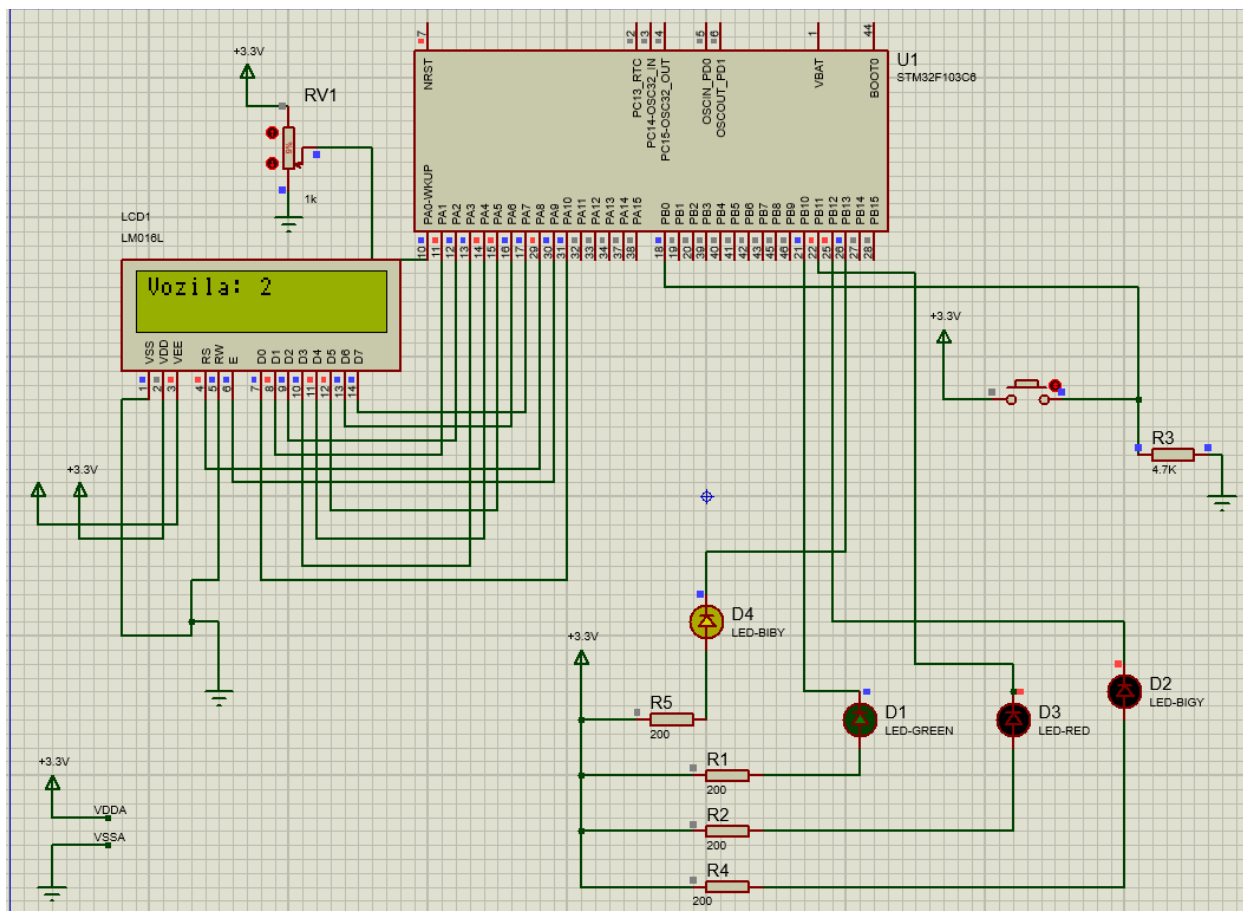


Упутсво за употребу система

Као што је раније описано, систем се покреће кликом на дугме, пре покретања система на дисплеју исписана порука „ИСКЉУЧЕНО“ и све диоде су угашене, тј. не светле.



Приликом клика на дугме, систем се покреће, на дисплеју се исписује број возила која су тренутно прошла и упаљена је одговарајућа диода:



Диода D1(зелена диода) је упаљена и сигнализира да је број возила који је прошао мањи од 10, а D4(амбер диода) се пали/гаси приликом проласка возила кроз систем и на тај начин сигнализира пролазак возила као и даје јасну визуелну идентификацију исправности ситема.Брзина „свиткања“ зависи од одабране вредности на потенциометру.