

04.10.2014 12:59 GECOM INTERNET

Система автоматизації трьохфазного світлофорного регулювання

**Керівник проекту:
доц. Проценко Д.П.**

**Виконала: ст. гр. ЕПА-14м
Кметь Н.Д.**

Мета і завдання роботи

Мета і завдання дослідження. Метою магістерської роботи є поліпшення пропускну́ї спроможності та надійності роботи елементів систем керування світлофорних перехресть за рахунок розробки системи автоматизації трьохфазного світлофорного регулювання та схеми керування на базі сучасних МП пристроїв.

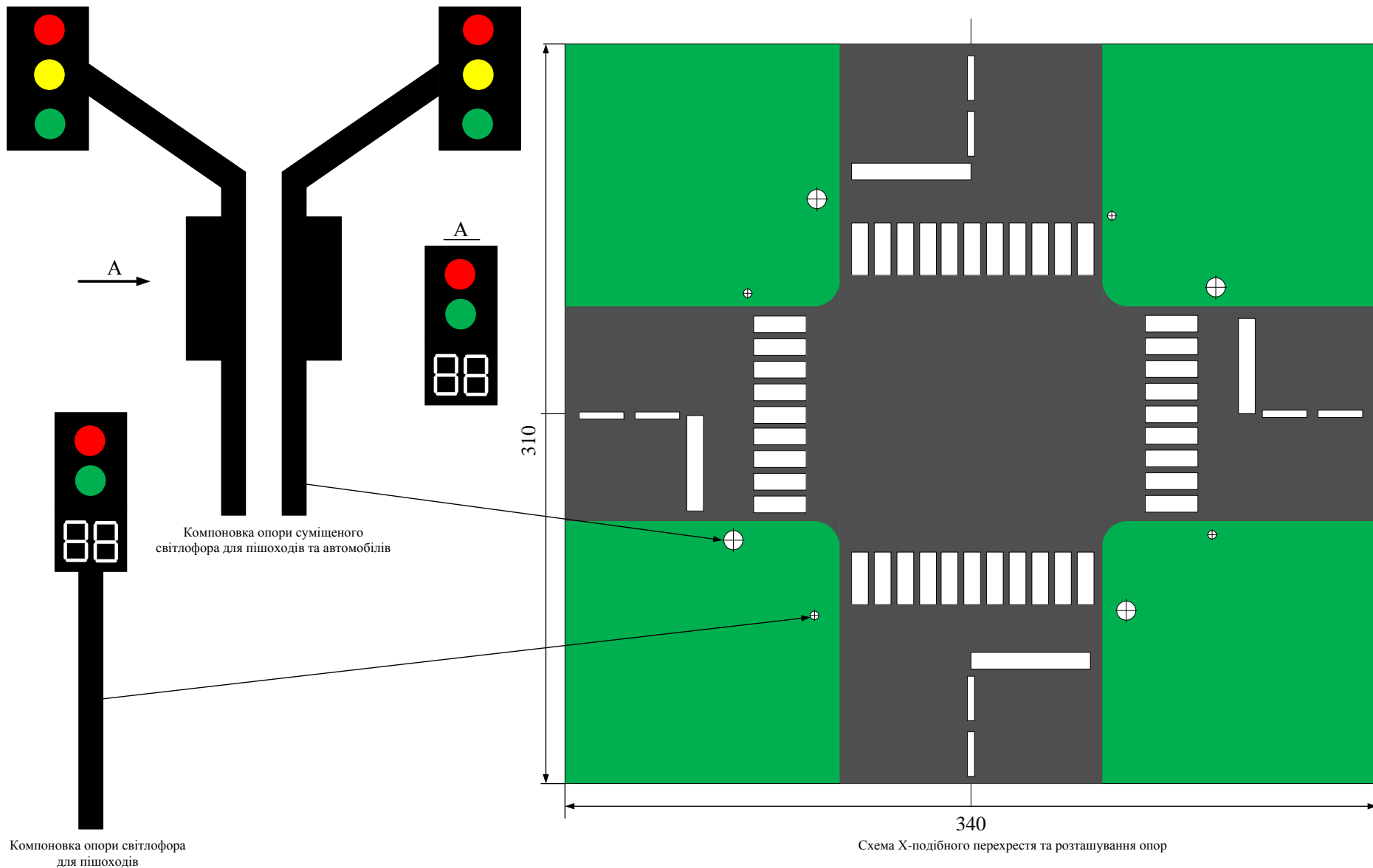
Для досягнення поставленої мети необхідно розв'язати такі завдання:

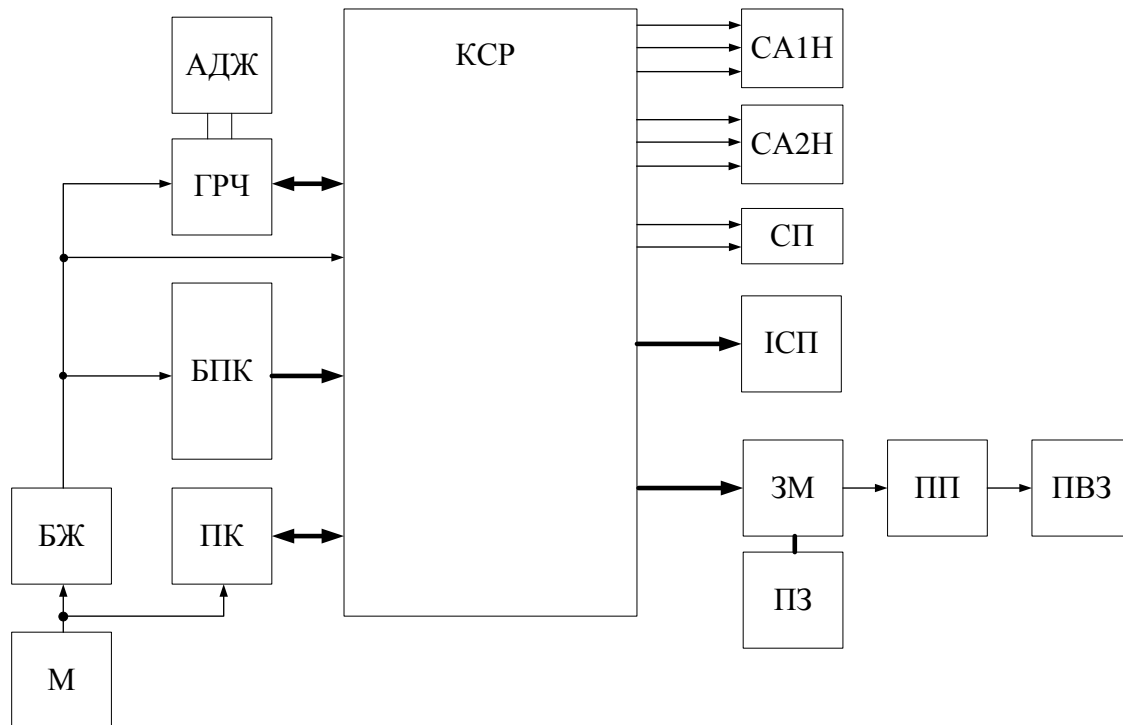
- привести аналіз алгоритмів світлофорного регулювання, та вимог до побудови систем автоматизації регулювання;
- здійснити техніко-економічне обґрунтування базового модуля керування;
- здійснити розрахунок та вибір елементної бази для побудови системи.
- розробити принципову, функціональну та структурну схему системи автоматизації трифазного світлофорного регулювання;
- розробити комплекс алгоритмічного та програмного забезпечення для функціонування апаратного комплексу;
- здійснити оптимізацію роботи світлофорного перехрестя в залежності від інтенсивності руху транспорту;
- здійснити практичну реалізацію у вигляді макета.

ТЕХНІКО-ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ ВИБОРУ ВАРІАНТУ РЕАЛІЗАЦІЇ СИСТЕМИ АВТОМАТИЗАЦІЇ ТРЬОХФАЗНОГО СВІТЛОФОРНОГО РЕГУЛЮВАННЯ

Показники	Система електричного привода			
	На базі МК EPF8282ALC84-4	На базі МК AT90S4433- 8AU	На базі МК PIC18F452	На базі ArduinoMega 2560
Мікропроцесорний пристрій МП, грн	700	630	600	1200
Інформаційно-вимірювальні прилади ІВП, грн	600	600	600	0
Комутаційна апаратура КА, грн	100	100	100	0
Загальна вартість системи керування СК, грн	1400	1330	1300	1200
Капітальні вкладення К, грн	2400	2330	2300	2200
Річні капітальні витрати $K_{річні}$, грн/рік	408	396,1	391	374
Амортизаційні відрахування C_A , грн/рік	240	233	230	220
Відрахування на ремонт C_p , грн/рік	48	46,6	46	44
Відрахування на обслуговування C_o , грн/рік	14,4	13,98	13,8	13,2
Загальні відрахування С, грн/рік	302,4	293,58	289,8	277,2
Приведені витрати З, грн/рік	710,4	689,68	680,8	651,2

Схема перехрестя та конструкція опор світлофора на макеті

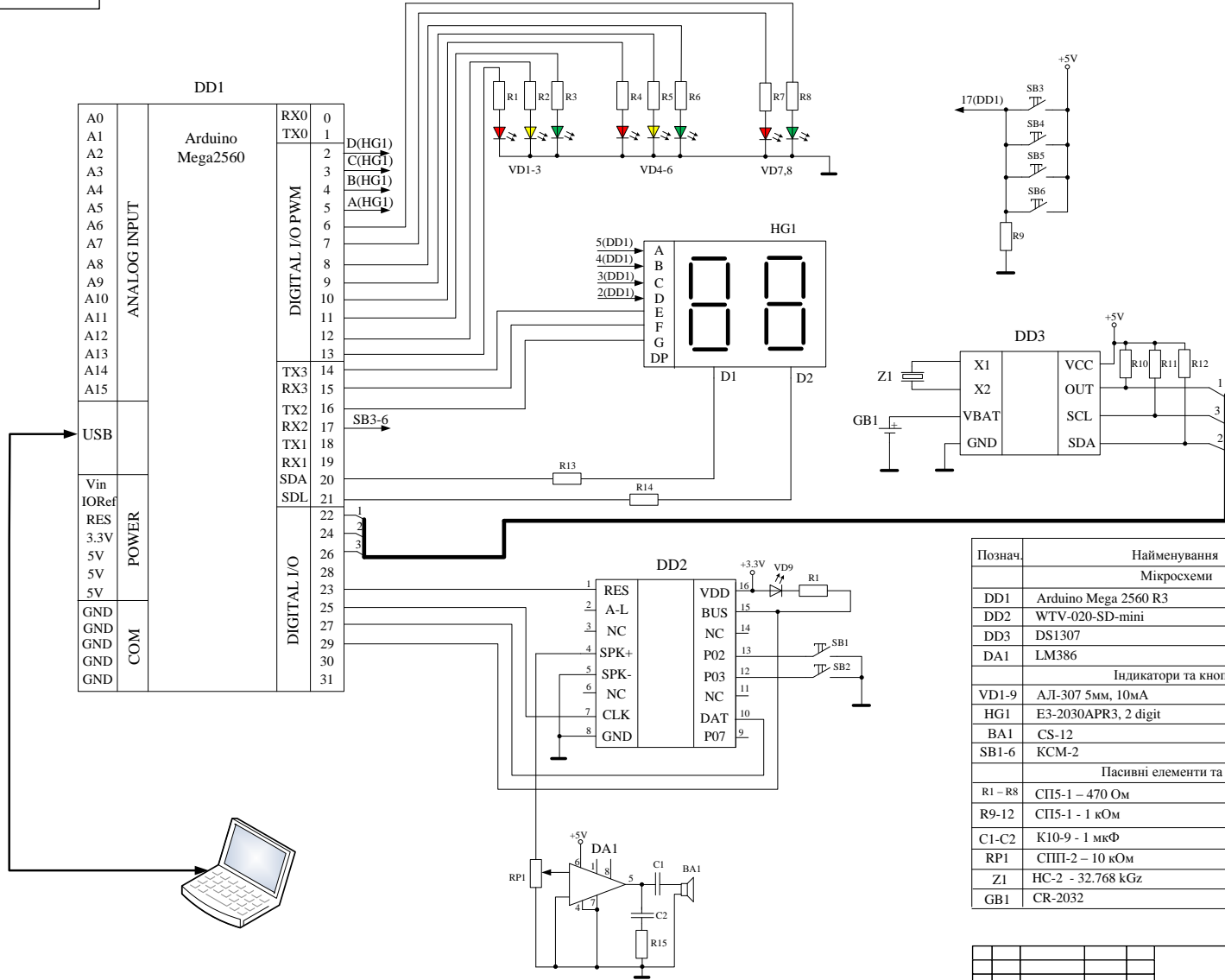




Познач.	Найменування	Кіл.	Примітка
КСР	Контролер світлофорного регулювання	1	
СА1Н	Світлофор автомобільний 1 напрям	1	
СА2Н	Світлофор автомобільний 2 напрям	1	
СП	Світлофор пішоходний	1	
ІСП	Індикатор світлофора для пішоходів	1	
ЗМ	Звуковий модуль	1	
ПЗ	Пирстрий запам'ятовування	1	
ПП	Підсилювач потужності	1	
ПВЗ	Пристрій виведення звуку	1	
АДЖ	Автономне джерело живлення	1	
ГРЧ	Годинник реального часу	1	
БПК	Блок кнопок пішоходів	1	
М	Мережа живлення	1	
ПК	Персональний компютер	1	
БЖ	Блок живлення	1	

					08-19.МКР.003.00.000 Е1			
Зм	Лист	№ Докум	Плп.	Дата	Система автоматизації трьохфазного світлофорного регулювання. Схема електрична структура	Літ.	Маса	Масш.
Розробив	Кметь Н.Д.							
Перевіряв	Пронько Д.П.					Аркуш 4		Аркушів
Н.контр.	Курочка В.П.					ВНТУ, гр.ЕПА-14м		
Затв.	Кутів В.М.							

№ аркуша
№ документа
№ аркуша
№ документа
№ аркуша
№ документа



Познач.	Найменування	Кіл.	Примітка
Мікросхеми			
DD1	Arduino Mega 2560 R3	1	Базовий модуль
DD2	WTV-020-SD-mini	1	MP3 модуль
DD3	DS1307	1	МС годинника
DA1	LM386	3	Підсилювач звуку
Індикатори та кнопки			
VD1-9	АЛ-307 5мм, 10мА	9	
HG1	Е3-2030APR3, 2 digit	1	
BA1	CS-12	1	Гучномовець
SB1-6	KCM-2	6	
Пасивні елементи та інші			
R1 - R8	СПІ5-1 - 470 Ом	8	
R9-12	СПІ5-1 - 1 кОм	4	
C1-C2	K10-9 - 1 мкФ	3	
RP1	СПП-2 - 10 кОм	1	Потенціометр
Z1	HC-2 - 32.768 kGz		
GB1	CR-2032		Гальв. елем

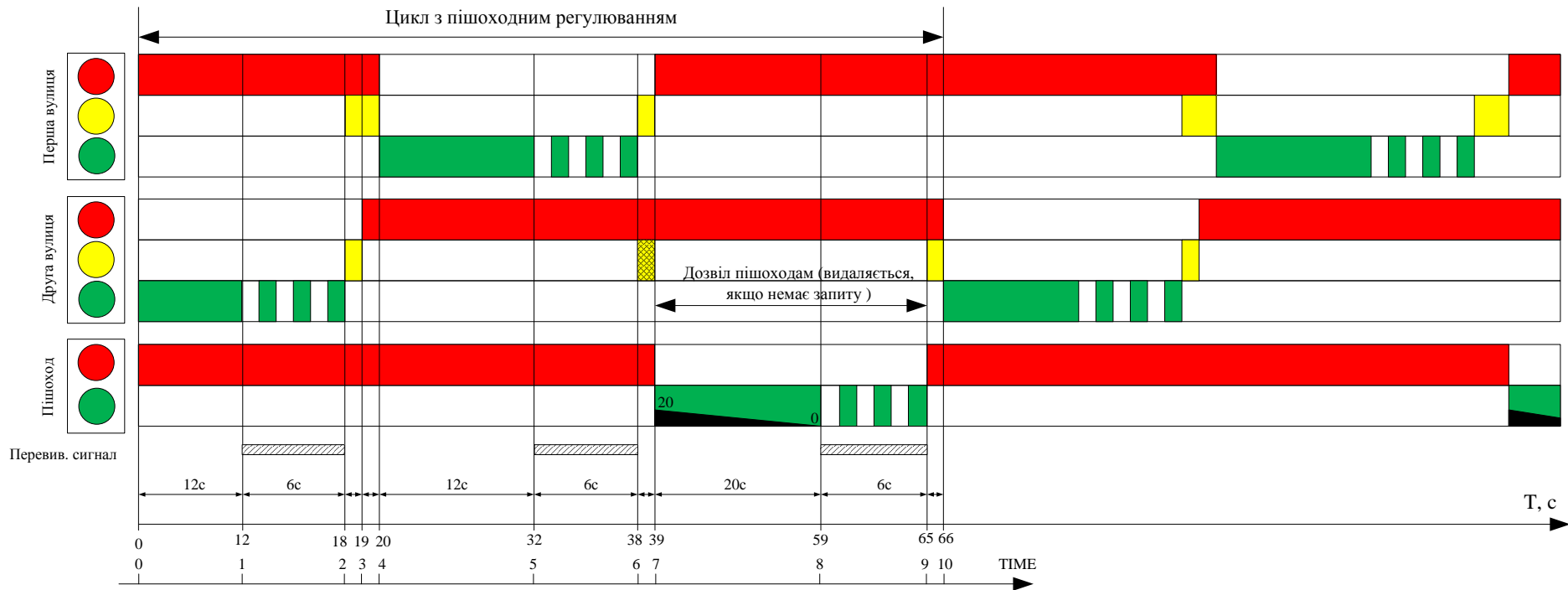
08-19.МКР.003.00.000 ЕЗ			
Зм.	Лист	№ Докум.	Лит.
Розробив	Кметь Н.Д.	Підп.	Маса
Перевірив	Проценка Д.П.	Дата	Масш.
Н.контр.	Курочка Н.П.		Аркуш
Зав.	Кутні В.М.		Аркуш

Система автоматизації трьохфазного світлофорного регулювання
Схема електрична принципова

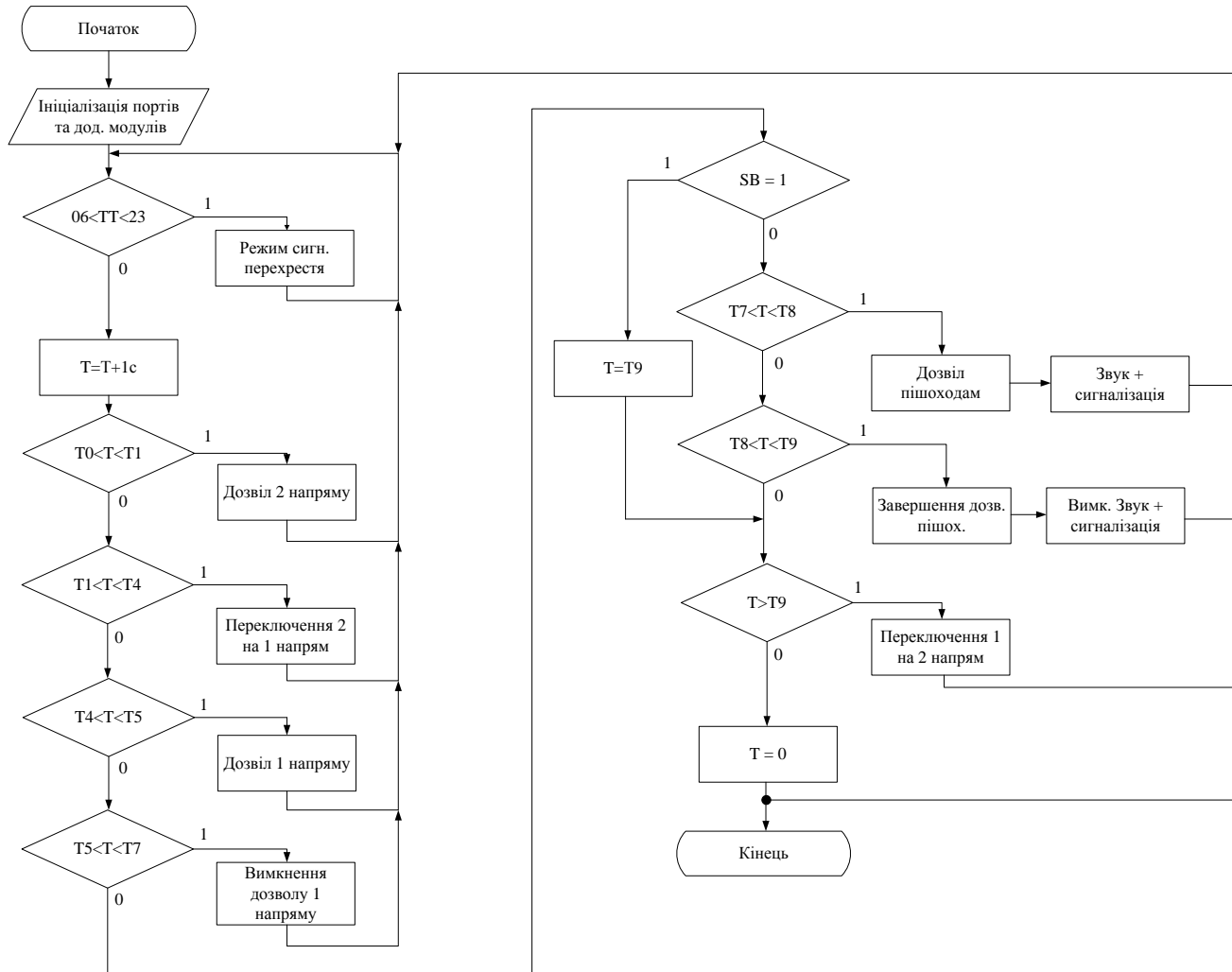
ВНТУ, гр.ЕПА-14м

Планув. та зарп. №
Інв. №
Інв. №
Планув. та зарп. №
Інв. №

Часова діаграма роботи трьохфазного світлофорного регулювання



Алгоритм програми трьохфазного світлофорного регулювання



Особливості реалізації програмних алгоритмів

Ініціалізація підключення звукового модуля та вклення в проєкт відповідної бібліотеки.
#include <Wtv020sd16p.h>
int resetPin = 23; // The pin number of the reset pin.
int clockPin = 25; // The pin number of the clock pin.
int dataPin = 27; // The pin number of the data pin.
int busyPin = 29; // The pin number of the busy pin.
Wtv020sd16p
Wtv020sd16p(resetPin,clockPin,dataPin,busyPin);

Ініціалізація підключення семисегментного індикатора зворотного відліку та вклення в проєкт відповідної бібліотеки.

```
#include <SevenSeg.h>;  
SevenSeg disp (5,4,3,2,14,15,16);  
const int numOfDigits = 2;  
int digitPins[numOfDigits]={  
20,21};
```

Ініціалізація змінних тривалості часу для семисегментного індикатора, додаткових змінних переходів, та кнопки дозволу.

```
int TimeGreen=20+1, fiksator=0, kk=0, k=0, n=0;  
const int buttonPin = 17; // the number of the  
pushbutton pin  
int buttonState = 0; // variable for reading the  
pushbutton status
```

Ініціалізація підключення мікросхеми годинника реального часу та вклення в проєкт відповідної бібліотеки.

```
#include <DS1302.h>  
// Init the DS1302  
DS1302 rtc(22, 24, 26);  
// DS1302: CE pin -> Arduino Digital 22  
// I/O pin -> Arduino Digital 24  
// SCLK pin -> Arduino Digital 26  
// Init a Time-data structure  
Time t;
```

Ініціалізація цифрових виходів для підключення світлодіодів відповідних сигналів світлофора.

```
int RedLed_1 = 13; // Порт 13, черв світлодіод  
int YelLed_1 = 12; // Порт 12, жовт світлодіод  
int GrnLed_1 = 11; // Порт 11, зелений світлодіод
```

```
int RedLed_2 = 10; // Порт 13, черв світлодіод  
int YelLed_2 = 9; // Порт 12, жовт світлодіод  
int GrnLed_2 = 8; // Порт 11, зелений світлодіод  
int RedMan = 7; // Порт 7, черв світлодіод пію  
int GrnMan = 6; // Порт 6, зелений пію світлодіод  
boolean Blink = false;  
long Previous = 0; // встановлюєм початкове  
значення Mills()  
int val = 0; // змінна для перемикання
```

Ініціалізація характерних точок комутації сигналів світлофора згідно часової діаграми роботи.

```
int Time_0 = 0; // початковий стан  
int Time_1 = 12; //  
int Time_2 = 18;
```

```
int Time_3 = 19;  
int Time_4 = 20;  
int Time_5 = 32;  
int Time_6 = 38;  
int Time_7 = 39;  
int Time_8 = 59;  
int Time_9 = 65;
```

Основний цикл ініціалізація модулів та встановлення початкового стану згідно діаграми.

```
void setup()  
{  
  wtv020sd16p.reset();  
  rtc.halt(false);  
  rtc.writeProtect(false);  
  // Setup Serial connection  
  Serial.begin(9600);  
  disp.setDigitPins (numOfDigits,digitPins);  
  disp.setCommonAnode();  
  pinMode(RedLed_1, OUTPUT);  
  pinMode(YelLed_1, OUTPUT);  
  pinMode(GrnLed_1, OUTPUT);  
  pinMode(RedLed_2, OUTPUT);  
  pinMode(YelLed_2, OUTPUT);  
  pinMode(GrnLed_2, OUTPUT);  
  pinMode(RedMan, OUTPUT);  
  pinMode(GrnMan, OUTPUT);  
  pinMode(buttonPin, INPUT);
```

Налаштування послідовної передачі та годинника реального часу.

```
// Setup Serial connection  
Serial.begin(9600);  
  
// The following lines can be commented out to use  
the values already stored in the DS1302  
// rtc.setDOW(MONDAY); // Set Day-of-Week to  
FRIDAY  
// rtc.setTime(12, 0, 0); // Set the time to 12:00:00  
(24hr format)  
// rtc.setDate(1, 6, 2014); // Set the date to August  
6th, 2010
```

Початок основної програми зчитування поточного часу та виведення значень в послідовний порт.

```
void MyTimeGet()  
{  
  t = rtc.getTime();  
  // Send date over serial connection  
  Serial.print("Today is the ");  
  Serial.print(t.date, DEC);  
  Serial.print(", day of ");  
  Serial.print(rtc.getMonthStr());  
  Serial.print(" in the year ");  
  Serial.print(t.year, DEC);  
  Serial.println(" ");  
  // Send Day-of-Week and time  
  Serial.print("It is the ");  
  Serial.print(t.dow, DEC);  
  Serial.print(", day of the week (counting monday as  
the 1th), and it has passed ");  
  Serial.print(t.hour, DEC);  
  Serial.print(" hour(s). ");  
  Serial.print(t.min, DEC);
```

```
Serial.print(" minute(s) and ");  
Serial.print(t.sec, DEC);  
Serial.println(" second(s) since midnight.");  
// Send a divider for readability  
Serial.println(" -----  
- - -");
```

```
}  
void loop()  
{  
  MyTimeGet();  
  while((t.hour>6)&&(t.hour<22)){  
    if(k==1){  
      disp.write(TimeGreen);  
    }  
    buttonState = digitalRead(buttonPin);  
    if ((buttonState == HIGH)&&kk==0) {  
      fiksator=1;  
    }  
    if (mills() - Previous > 1000)  
    {  
      Previous = mills(); // встановлюємо нове  
значення previous  
      val ++;
```

Блок програми, що відповідає за зміну стану сигналів перевірка здійснюється кожену секунду.

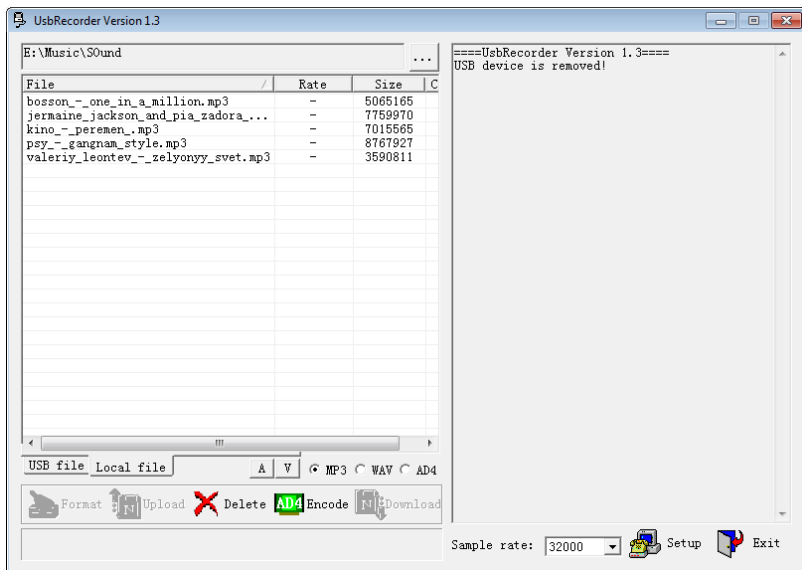
```
if (val<Time_1)  
{  
  digitalWrite(RedLed_1, HIGH);  
  digitalWrite(YelLed_1, LOW);  
  digitalWrite(GrnLed_1, LOW);  
  
  digitalWrite(RedLed_2, LOW);  
  digitalWrite(YelLed_2, LOW);  
  digitalWrite(GrnLed_2, HIGH);  
  
  digitalWrite(RedMan, HIGH);  
  digitalWrite(GrnMan, LOW);  
  Blink=!Blink;  
}  
if ((val>=Time_1)&&(val<Time_2))  
{  
  digitalWrite(RedLed_1, HIGH);  
  digitalWrite(YelLed_1, LOW);  
  digitalWrite(GrnLed_1, LOW);  
  digitalWrite(RedLed_2, LOW);  
  digitalWrite(YelLed_2, LOW);  
  digitalWrite(GrnLed_2, Blink);  
  Blink=!Blink;  
  digitalWrite(RedMan, HIGH);  
  digitalWrite(GrnMan, LOW);  
}  
if (val == Time_2)  
{  
  MyTimeGet();  
  digitalWrite(RedLed_1, HIGH);  
  digitalWrite(YelLed_1, HIGH);  
  digitalWrite(GrnLed_1, LOW);  
  
  digitalWrite(RedLed_2, LOW);  
  digitalWrite(YelLed_2, HIGH);  
  digitalWrite(GrnLed_2, HIGH);  
  
  digitalWrite(RedMan, HIGH);  
  digitalWrite(GrnMan, LOW); }  
}
```

```
if (val == Time_3)  
{  
  digitalWrite(RedLed_1, HIGH);  
  digitalWrite(YelLed_1, HIGH);  
  digitalWrite(GrnLed_1, LOW);  
  
  digitalWrite(RedLed_2, HIGH);  
  digitalWrite(YelLed_2, LOW);  
  digitalWrite(GrnLed_2, LOW);  
  
  digitalWrite(RedMan, HIGH);  
  digitalWrite(GrnMan, LOW);  
}  
if (val == Time_4)  
{  
  digitalWrite(RedLed_1, LOW);  
  digitalWrite(YelLed_1, LOW);  
  digitalWrite(GrnLed_1, HIGH);  
  
  digitalWrite(RedLed_2, HIGH);  
  digitalWrite(YelLed_2, LOW);  
  digitalWrite(GrnLed_2, LOW);  
  
  digitalWrite(RedMan, HIGH);  
  digitalWrite(GrnMan, LOW);  
}  
if ((val>=Time_5)&&(val<Time_6))  
{  
  digitalWrite(RedLed_1, LOW);  
  digitalWrite(YelLed_1, LOW);  
  digitalWrite(GrnLed_1, Blink);  
  Blink=!Blink;  
  
  digitalWrite(RedLed_2, HIGH);  
  digitalWrite(YelLed_2, LOW);  
  digitalWrite(GrnLed_2, LOW);  
  
  digitalWrite(RedMan, HIGH);  
  digitalWrite(GrnMan, LOW);  
}  
if (val == Time_6)  
{  
  digitalWrite(RedLed_1, LOW);  
  digitalWrite(YelLed_1, HIGH);  
  digitalWrite(GrnLed_1, LOW);  
  digitalWrite(RedLed_2, HIGH);  
  if(fiksator==1) digitalWrite(YelLed_2, LOW);  
  if(fiksator==0) digitalWrite(YelLed_2, HIGH);  
  kk=1;  
  digitalWrite(GrnLed_2, LOW);  
  digitalWrite(RedMan, HIGH);  
  digitalWrite(GrnMan, LOW);  
}  
if (val == Time_7)&&(val<Time_8)&&  
fiksator==1)  
  //Plays synchronously an audio file. Busy pin is  
used for this method.  
  if(val==Time_7) {  
    wtv020sd16p.asyncPlayVoice(n);
```

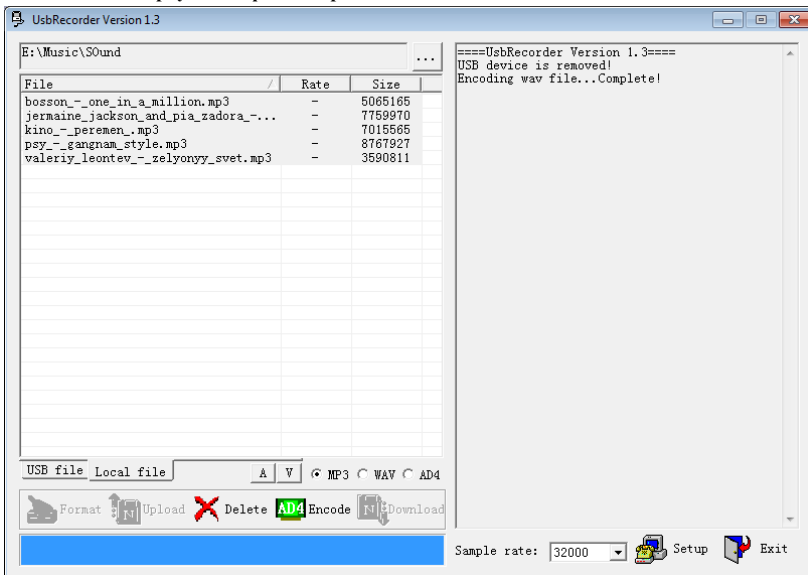
```
n++;  
if(n==20){  
  n=0;  
}  
}  
digitalWrite(RedLed_1, HIGH);  
digitalWrite(YelLed_1, LOW);  
digitalWrite(GrmLed_1, LOW);  
digitalWrite(YelLed_2, HIGH);  
digitalWrite(YelLed_2, LOW);  
digitalWrite(GrmLed_2, LOW);  
digitalWrite(YelLed_2, HIGH);  
digitalWrite(GrmLed_2, LOW);  
digitalWrite(GrmMan, HIGH);  
k=1;  
TimeGreen--;  
}  
else if ((val>=Time_7)&&(val<Time_8)&&  
fiksator==0) {  
  val = Time_9;  
}  
if ((val>=Time_8)&&(val<Time_9)&&  
fiksator==1)  
{  
  wtv020sd16p.stopVoice();  
  k=0;  
  TimeGreen=21;  
  digitalWrite(RedLed_1, HIGH);  
  digitalWrite(YelLed_1, LOW);  
  digitalWrite(GrmLed_1, LOW);  
  digitalWrite(RedLed_2, HIGH);  
  digitalWrite(YelLed_2, HIGH);  
  digitalWrite(GrmLed_2, LOW);  
  digitalWrite(GrmMan, Blink);  
  Blink=!Blink;  
}  
if (val == Time_9)  
{  
  kk=0;  
  fiksator=0;  
  digitalWrite(RedLed_1, HIGH);  
  digitalWrite(YelLed_1, LOW);  
  digitalWrite(GrmLed_1, LOW);  
  digitalWrite(RedLed_2, HIGH);  
  digitalWrite(YelLed_2, HIGH);  
  digitalWrite(GrmLed_2, LOW);  
  
  digitalWrite(RedMan, HIGH);  
  digitalWrite(GrmMan, LOW);  
  val = 0;  
}  
}  
digitalWrite(YelLed_1, LOW);  
digitalWrite(YelLed_2, LOW);  
delay(1000);  
digitalWrite(YelLed_1, HIGH);  
digitalWrite(YelLed_2, HIGH);  
delay(1000);
```

Конвертування файлів для звукового модуля

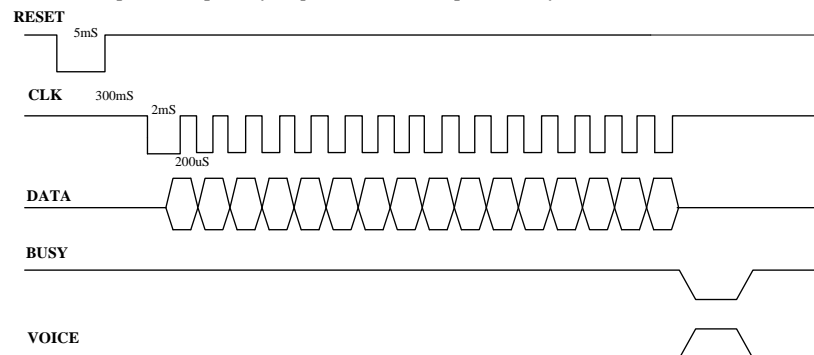
Інтерфейс програми UsbRecorder Version 1.3



Конвертування файлів mp3 в UsbRecorder Version 1.3



Передача адреси аудіофайла для відтворення модулем WTV-020-SD-mini



Відповідність адрес та назв файлів

ADDRESSES	TRIGGER STATE	FILE NAME (.ad4)	TRIGGER DATA (BINARY)
ADDR 1	PLAY 1 th GROUP VOICE	0000	0000000000000000
ADDR 2	2 th	0001	0000000000000001
ADDR 3	3 th	0002	0000000000000010
ADDR 4	4 th	0003	0000000000000011
.....
ADDR 509	509 th	0508	0000000111111100
ADDR 510	510 th	0509	0000000111111101
ADDR 511	511 th	0510	0000000111111110
ADDR 512	512 th	0511	0000000111111111

Результат конвертування файлів

Имя	Тип	Размер
[.]	<Папка>	
psy_-_gangnam_style	mp3	8 767 927
bosson_-_one_in_a_million	mp3	5 065 165
valeriy_leontev_-_zelyonyy_svet	mp3	3 590 811
kino_-_peremen_	mp3	7 015 565
jermaine_jackson_and_pia_zadora_-_when_the_rain_begins_to_fall..	mp3	7 759 970
bosson_-_one_in_a_million	ad4	3 376 697
jermaine_jackson_and_pia_zadora_-_when_the_rain_begins_to_fall..	ad4	5 177 683
kino_-_peremen_	ad4	4 683 237
psy_-_gangnam_style	ad4	3 507 100
valeriy_leontev_-_zelyonyy_svet	ad4	3 590 692

Имя	Тип	Размер
[.]	<Папка>	
0000	ad4	3 376 697
0001	ad4	5 177 683
0002	ad4	4 683 237
0003	ad4	3 507 100
0004	ad4	3 590 692

Адаптивний алгоритм керування дорожнім рухом

Тривалість циклу регулювання:

$$T_c = t_{п1} + t_{о1} + t_{п2} + t_{о2}$$

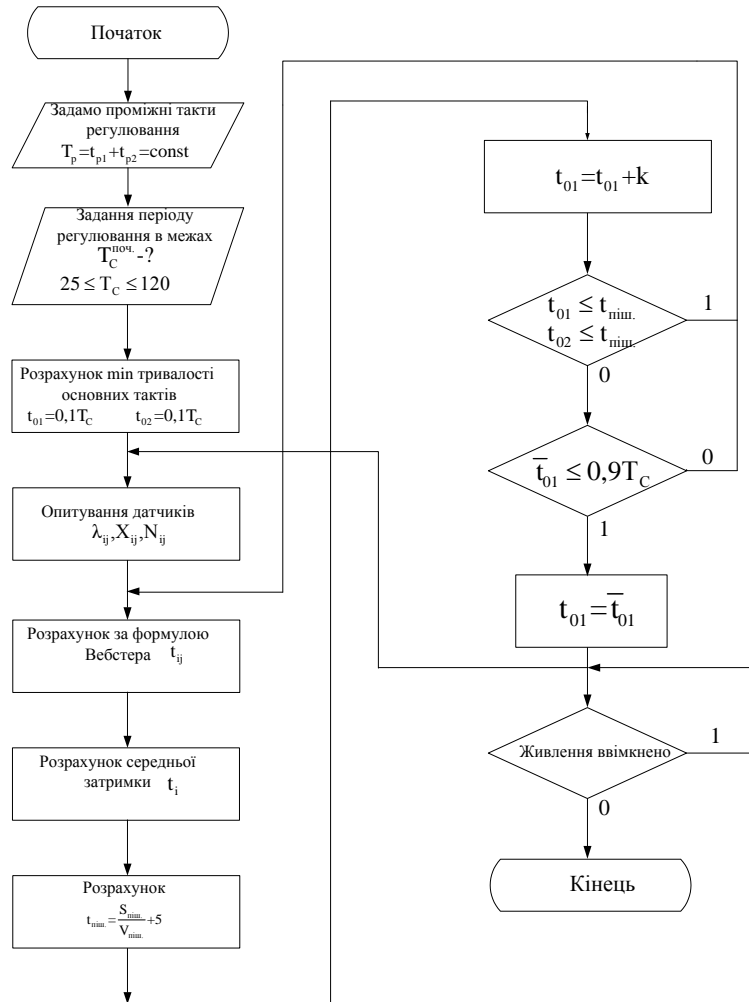
В якості критерію оптимізації вибрано критерій мінімальної затримки, яка розраховується за відомою формулою Вебстера:

$$t_{ij} = \frac{T_c \cdot (1 - N_{ij})^2}{2(1 - N_{ij} x_{ij})} + \frac{x_{ij}^2}{2\lambda_{ij}(1 - x_{ij})} - 0,65 \left(\frac{T_c}{\lambda_{ij}^2} \right)^{1/5} \times x_{ij}^{2+3x_{ij}}$$

λ_{ij} – інтенсивність руху за напрямом i на перехресті j ; N_{ij} – відношення тривалості циклу дозволяючого сигналу до тривалості циклу; x_{ij} – ступінь насиченості конкретного напрямку руху; T_c – тривалість циклу регулювання.

$$\bar{t}_i = \frac{\sum_j t_{ij} \cdot \lambda_j}{\sum_j \lambda_j} \rightarrow \min$$

Алгоритм мікропроцесорного пристрою, що реалізує адаптивне керування сигналами світлофора



Висновки

Удосконалено підхід до дослідження адаптивної системи керування дорожнім рухом за критерієм мінімальної транспортної затримки, який дозволяє проводити лабораторні дослідження систем автоматизації трьохфазного світлофорного регулювання та оперативно визначати необхідний час вмикання дозволяючи сигналів для заданих умов руху.

Практичне значення одержаних у роботі результатів полягає в тому, що на основі теоретичних положень розроблено програмні і апаратні засоби, зокрема:

- розроблені структурна, функціональна та принципова електричні схеми стенда системи автоматизації трьохфазного світлофорного регулювання;
- розроблено алгоритмічне та програмне забезпечення для роботи системи автоматизації трьохфазного світлофорного регулювання;
- розглянуто особливості реалізації програмних алгоритмів та оптимізацію світлофорного регулювання дорожнього руху;
- здійснено експериментальні дослідження.

Результати роботи опубліковані на сайті III Міжнародної науково-технічної конференції «Оптимальне керування електроустановками (ОКЕУ-2015)» професорсько-викладацького складу, співробітників та студентів університету.

Дякую за увагу!