

**Федеральное государственное бюджетное образовательное
учреждение высшего образования "Белгородский государственный
технологический университет им. В.Г. Шухова"**

Кафедра программного обеспечения вычислительной техники и
автоматизированных систем.


Лабораторная работа работа № 3

Изучение среды разработки arduino IDE и стандартных библиотек для работы
с портами и интерфейсами микроконтроллера ATMEGA 328p.

Вариант 13

Выполнил:

Студент группы КБ-211



Коренев Д.Н.

Принял:

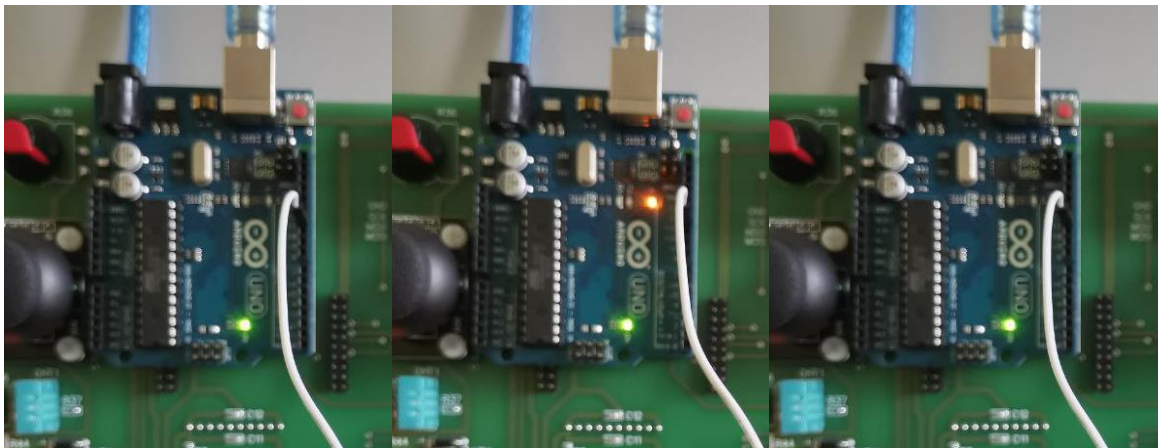
Шамраев А.А.

Цель работы: изучить функциональные возможности среды разработки Arduino IDE, научиться компилировать программу и загружать ее в микроконтроллер. Изучить стандартные библиотеки среды Arduino IDE для работы с портами ввода вывода, интерфейсами микроконтроллера Atmega 328p и научиться применять их на практике.

Задание

1. Получить навыки работы с GPIO портами. Зажечь светодиод, подключенный к порту D13, помогать им.

```
1 void setup() {  
2     pinMode(13, OUTPUT);  
3 }  
4  
5 void loop() {  
6     digitalWrite(13, HIGH);  
7     delay(1000);  
8     digitalWrite(13, HIGH);  
9     delay(1000);  
10 }
```



2. Воспользоваться стандартными функциями для генерации ШИМ:

В соответствии со своим номером в списке, сделать задания:

Номер меньше 8:

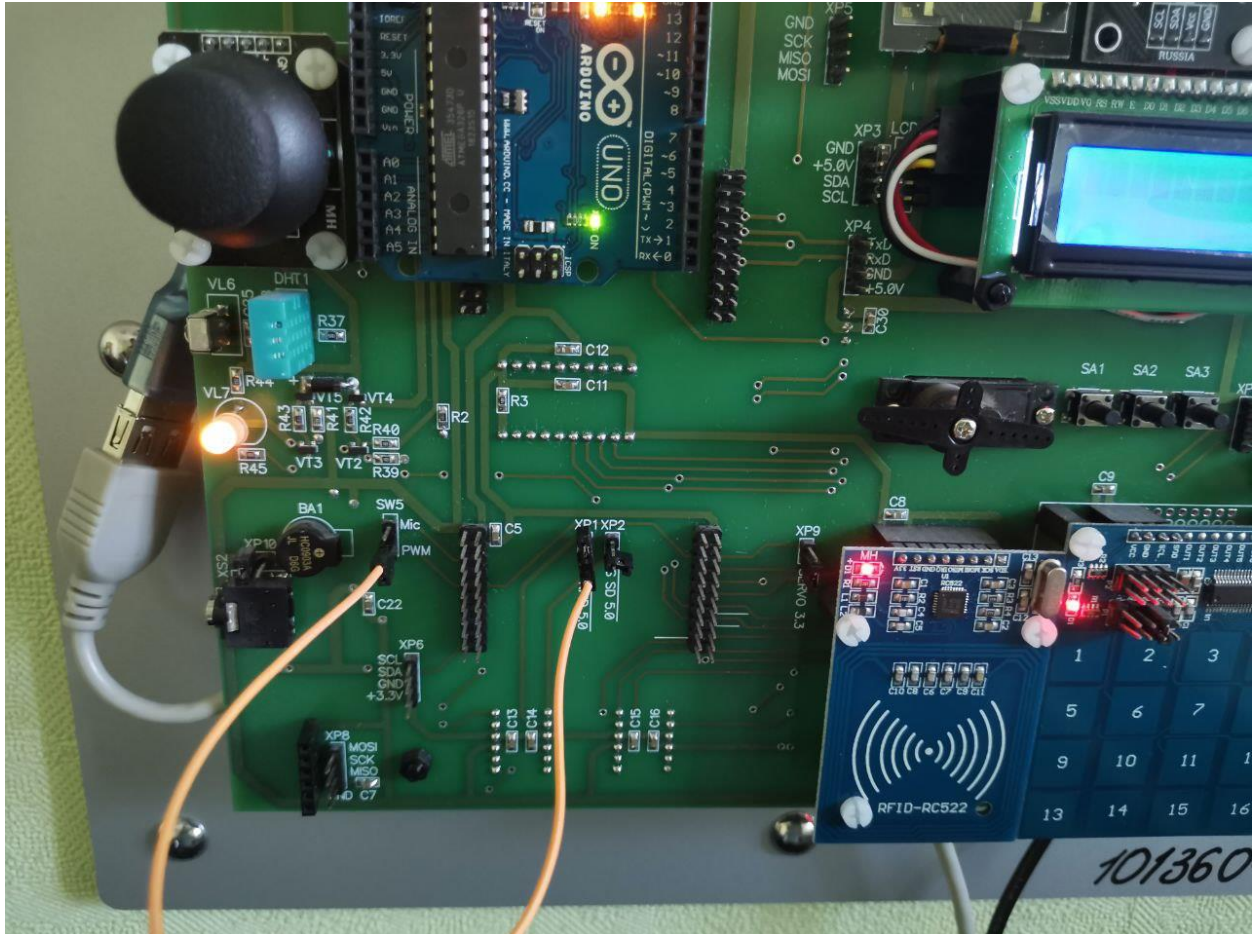
Сгенерировать ШИМ на 13 выводе Arduino. Плавно зажигать и гасить светодиод с некоторым периодом, яркостью в пределах от 0 до номера в журнале, умноженного на 30 с шагом, равным вашему номеру в журнале.

Управление по ШИМ недоступно на 13 выводе Arduino.

Номер больше или равен 8:

Установить перемычку для работы со спикером. Сделать сирену, изменяя скважность ШИМ от 30 до 150.

Период выбрать (номер в журнале)*50 мс.



```
1 void setup() {
2   pinMode(10, OUTPUT);
3 }
4
5 void loop() {
6   sise_t i;
7   for (i = 0; i <= 150; i += 10){
8     analogWrite(10, i);
9     delay(650);
10  }
11  while (i > 30) {
12    i -= 10;
13    analogWrite(10, i);
14    delay(650);
15  }
16 }
```

3. Получить напряжение на центральном выводе потенциометра при помощи АЦП и вывести его в COM порт. Вращая ручку, убедиться, что напряжение изменяется. С помощью функции map изменить диапазон получаемого значения в диапазон от 0 до номера в журнале. Если диапазон получился слишком маленьким, умножить максимальное значение на 10.

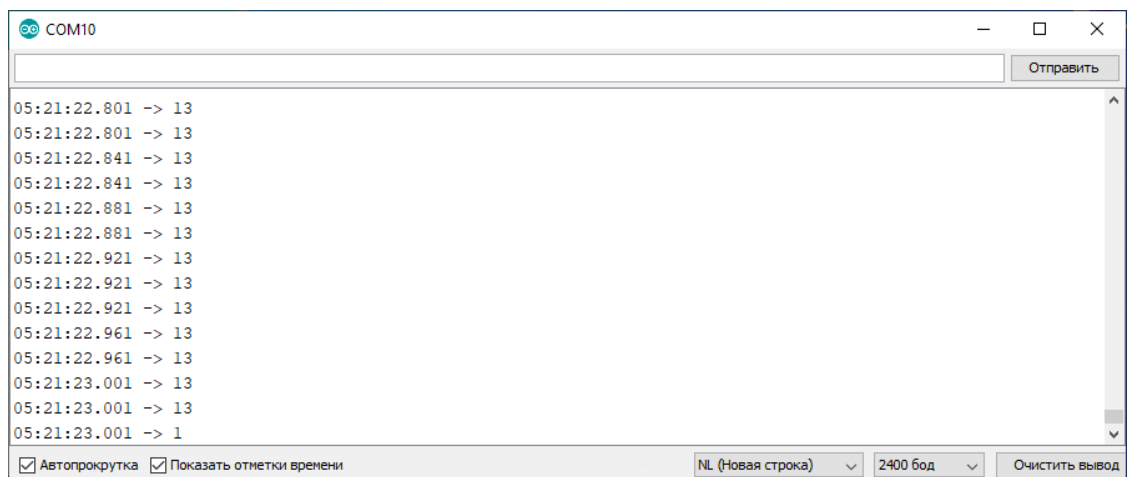
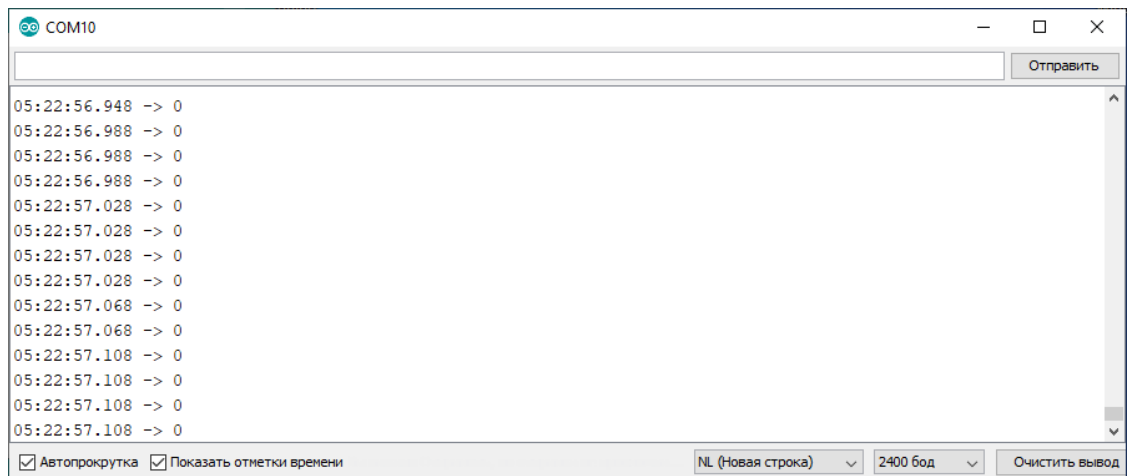
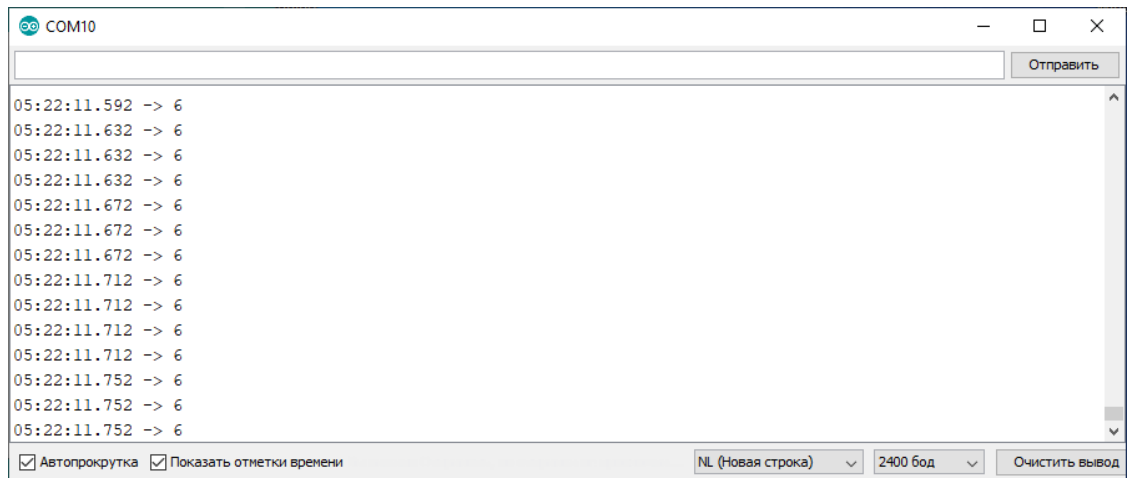
```
1 void setup() {
2   pinMode(A0, INPUT);
3   Serial.begin(2400);
4 }
5
```

```

6 void loop() {
7   int input = analogRead(A0);
8
9   int value = map(input, 0, 1023, 0, 14);
10  Serial.println(value);
11 }

```

Вращаем ручку:



4. С помощью интерфейса I2C изменить состояние расширителя портов. Зажечь несколько светодиодов на световой шкале, зажечь RGB светодиоды.

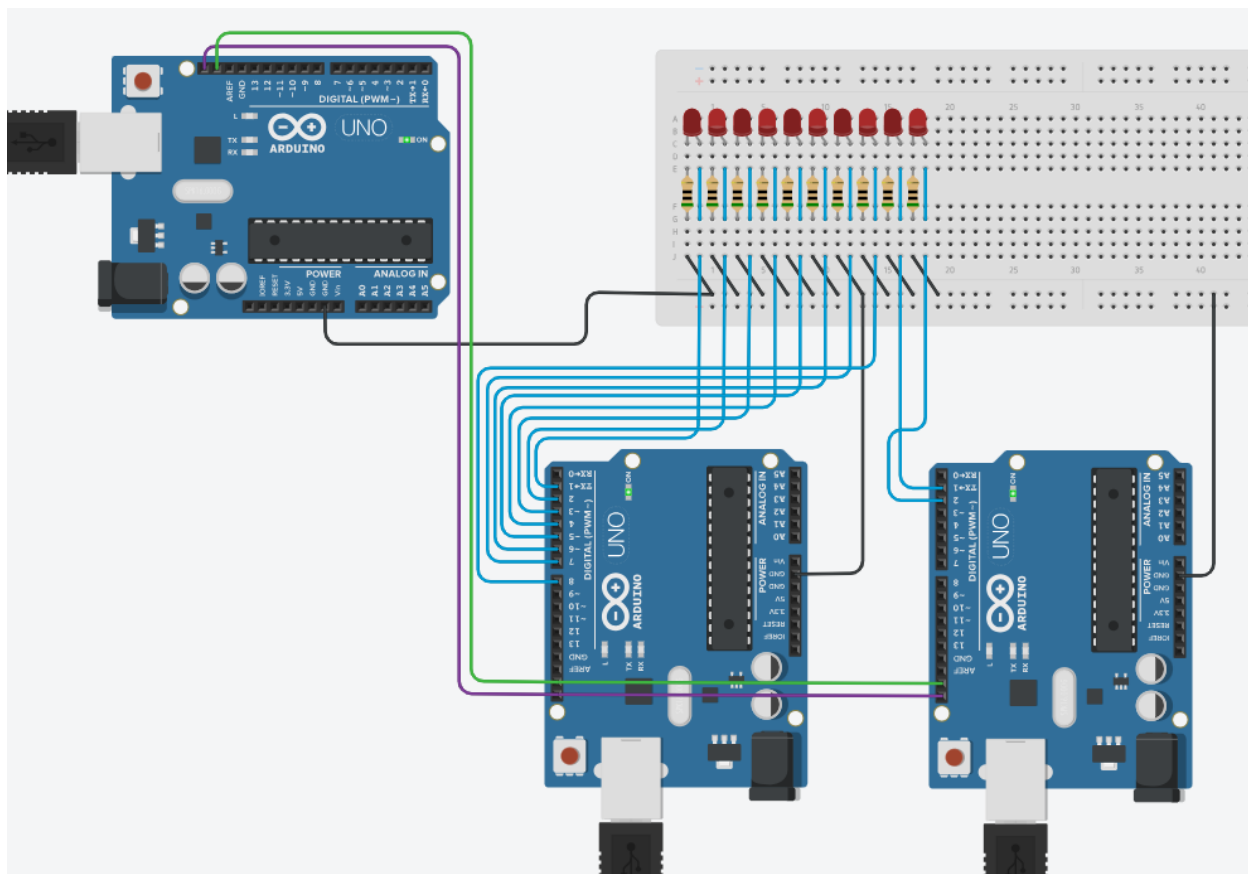
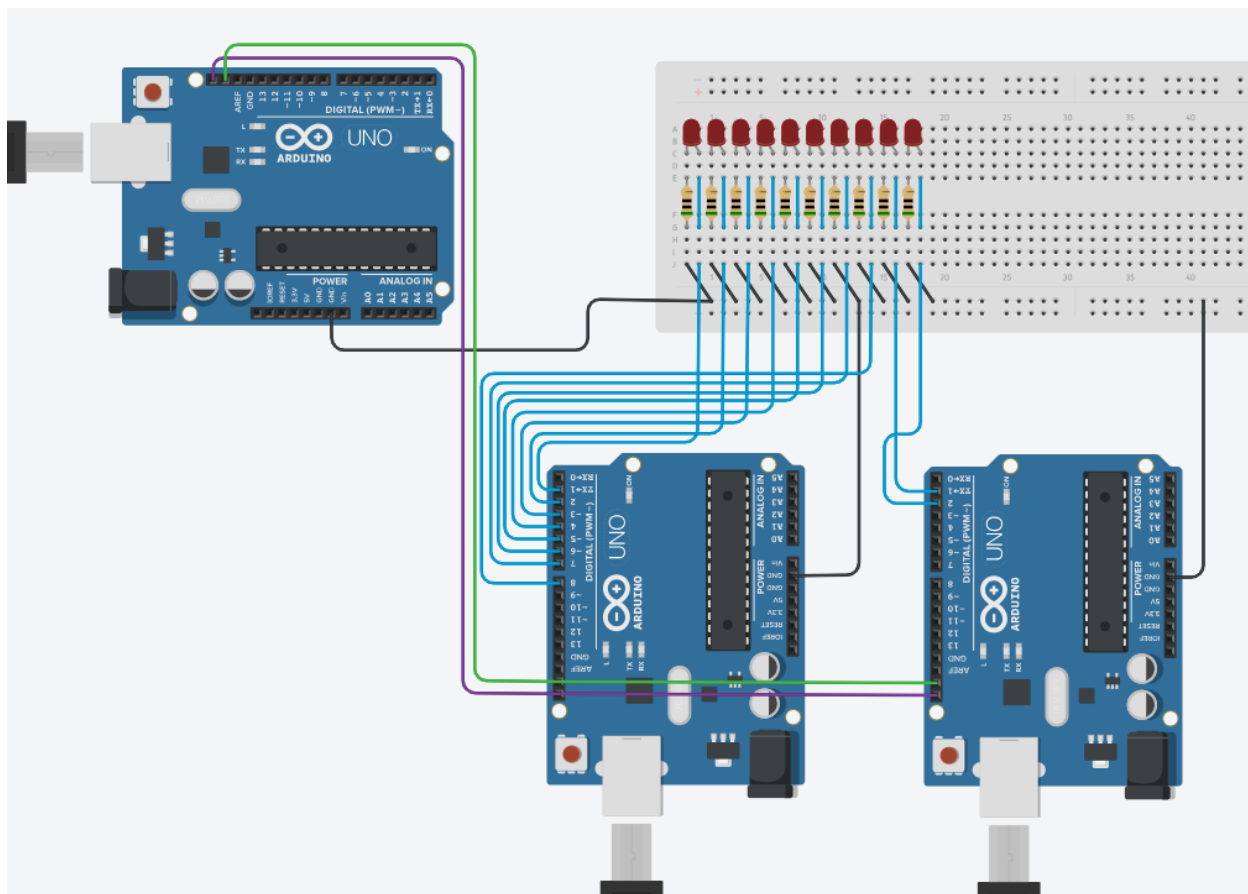
Полоски на шкале загораются при низком уровне на выводе (схема с общим анодом), а RGB светодиоды при высоком.

Сделать задание соответствующее своему варианту (номер в журнале % 6 + 1)

1. Зажечь полоски световой шкалы через одну, начиная с первой
2. **Зажечь полоски световой шкалы через одну, начиная со второй**
3. Зажечь оба RGB светодиода синим цветом
4. Зажечь половину световой шкалы и один из rgb светодиодов желтым цветом
5. Зажечь нижний RGB светодиод зеленым, а верхний красным
6. Зажечь оба RGB светодиода всеми цветами (смесь R, G, B), полоски на световой шкале сериями по 2 через 1.

```
1  #include <Wire.h>
2
3  void setup()
4  {
5      Wire.begin(0);
6  }
7
8  byte lampCode0 = 0b10101010; // 0x73
9  byte lampCode1 = 0b00000010; // 0x70
10
11 void loop()
12 {
13     Wire.beginTransmission(0x73);
14     Wire.write(lampCode0);
15     Wire.endTransmission();
16     Wire.beginTransmission(0x70);
17     Wire.write(lampCode1);
18     Wire.endTransmission();
19 }
```

Для задания в среде Tinkercad собрал схему (нижние платы Arduino замещают собой две PCA9538, верхняя работает в режиме “master”):



Код для плат Arduino в режиме “slave”:

```
1 #include <Wire.h>
```

```

2
3 #define WIRE_PORT 0x73
4 #define ZERO_PIN 1
5
6 void setup()
7 {
8     Wire.begin(WIRE_PORT);
9     Wire.onReceive(receiveEvent);
10 }
11
12 void loop()
13 {
14     delay(100);
15 }
16
17 void receiveEvent(int howMany)
18 {
19     byte c;
20     byte mask = 0b00000001;
21     c = Wire.read();
22     for (char i=0; i<8; i++)
23     {
24         if (c & mask){
25             digitalWrite(i+ZERO_PIN, HIGH);
26         } else {
27             digitalWrite(i+ZERO_PIN, LOW);
28         }
29         c >>= 1;
30     }
31 }

```

5. Написать микро-библиотеку (несколько функций), реализующую взаимодействие с расширителем портов PCA9538. Должна быть возможность устанавливать режим любого порта на расширителе, считывать уровень сигнала на нем, устанавливать выходной уровень, включать инверсию по входу.

Формировать нужные значения регистров можно при помощи маски/битового поля.

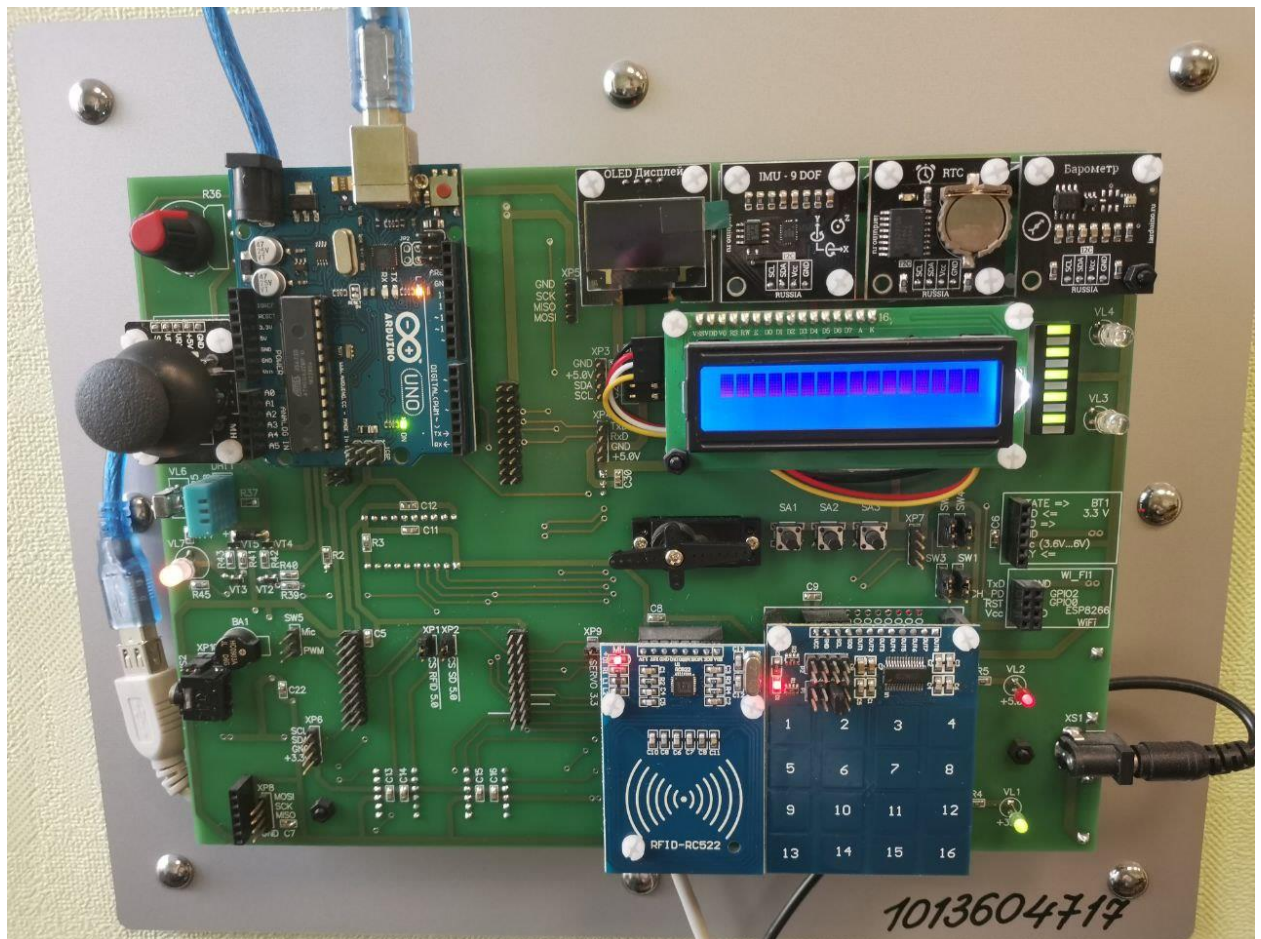
Номер в журнале меньше 8:

Используя полученную библиотеку, реализовать отображение числа при помощи световой шкалы. Задействована должна быть вся шкала, включая 2 нижних полоски. Сначала должно произойти тестирование шкалы – отображение чисел от 1 до 10, с шагом 1 за 1 секунду. Затем отображать остаток от деления значения, полученного с помощью `analogRead` на выводе A0 (U_R, центральный вывод потенциометра), на номер в журнале. При вращении ручки потенциометра, отображаемое число должно меняться.

Номер в журнале больше или равен 8:

Используя сирену из задания 2, сделать световой сигнал для сирены с помощью RGB светодиодов. Состояние расширителей портов менять при помощи реализованной функции. С помощью потенциометра должен

регулироваться период сирены. Начальное положение (при 0), должно соответствовать периоду в задании 2.



Файл "pca95xx.h":

```

1  #ifndef PCA95XX_H_
2  #define PCA95XX_H_
3
4  #include <Arduino.h>
5  #include <Wire.h>
6
7  #define PCA95XX_IN_REG 0x00
8  #define PCA95XX_OUT_REG 0x01
9  #define PCA95XX_POL_REG 0x02
10 #define PCA95XX_CONF_REG 0x03
11
12 #define PCA95XX_IN_ALL 0xFF
13 #define PCA95XX_OUT_ALL 0x00
14 #define PCA95XX_POL_DEFAULT_ALL 0x00
15 #define PCA95XX_POL_INVERTED_ALL 0xFF
16
17 #define PCA95XX_PIN_RESET_MODE OUTPUT
18
19 // Отправляет сигнал RESET для PCA9538.
20 // Parameters
21 // -----
22 //   resetPin(uint8_t): Вывод сброса
23 // Returns
24 // -----
25 //   NONE
26 void pca95xx_reset(int8_t resetPin);
27

```



```

28 // Настраивает все выводы в режим OUT по умолчанию.
29 // Parameters
30 // -----
31 //   addr(uint8_t): Адрес микросхемы PCA95XX на I2C
32 //   polInv(uint8_t): Инверсия выводов (байт настройки)
33 //   io(uint8_t): Режим ввод/вывод (байт настройки)
34 // Returns
35 // -----
36 //   NONE
37 void pca95xx_configure(uint8_t addr, uint8_t polInv = PCA95XX_POL_DEFAULT_ALL,
uint8_t io = PCA95XX_OUT_ALL);
38
39 // Получает данные из PCA95XX.
40 // Parameters
41 // -----
42 //   addr(uint8_t): Адрес микросхемы PCA95XX на I2C
43 // Returns
44 // -----
45 //   uint8_t: Данные, полученные от PCA95XX
46 uint8_t pca95xx_in(uint8_t addr);
47
48 // Отправляет данные в PCA95XX.
49 // Parameters
50 // -----
51 //   addr(uint8_t): Адрес микросхемы PCA95XX на I2C
52 //   data(uint8_t): Байт данных
53 // Returns
54 // -----
55 //   NONE
56 void pca95xx_out(uint8_t addr, uint8_t data);
57
58 #endif // PCA95XX_H_

```

Файл “pca95xx.cpp”:

```

1  #include "pca95xx.h"
2
3  // Внешний буфер для результатов работы функций
4  uint8_t pca95xx_i2c_buf = 0u;
5
6  // Отправляет байт data с командой cmd в PCA95XX
7  // с адресом addr.
8  // Parameters
9  // -----
10 //   addr(uint8_t): Адрес микросхемы PCA95XX на I2C
11 //   cmd(uint8_t): Код операции
12 //   data(uint8_t): Байт данных
13 // Returns
14 // -----
15 //   bool: Возвращает 1 в случае ошибки,
16 //         0 при удачном выполнении
17 bool pca95xx_i2c_sendData(uint8_t addr, uint8_t cmd,
uint8_t data = 0u)
18 {
19     Wire.beginTransmission(addr);
20     Wire.write(cmd);
21     if (cmd != PCA95XX_IN_REG)
22         Wire.write(data);
23     if (Wire.endTransmission())
24         return 1;
25     return 0;
26 }
27
28
29 // Получает байт данных из PCA95XX
30 // с адресом addr.
31 // Записывает результат в буфер.
32 // Parameters
33 // -----

```

```

34 //   addr(uint8_t): Адрес микросхемы PCA95XX на I2C
35 // Returns
36 // -----
37 //   bool: Возвращает 1 в случае ошибки,
38 //         0 при удачном выполнении
39 bool pca95xx_i2c_receiveData(uint8_t addr)
40 {
41     if (pca95xx_i2c_sendData(addr, PCA95XX_IN_REG)){
42         return 1;
43     }
44     delay(15);
45     Wire.requestFrom(addr, 1u, 1u);
46     while (Wire.available() < 1);
47     pca95xx_i2c_buf = Wire.read();
48     return 0;
49 }
50
51 // Отправляет сигнал RESET для PCA9538.
52 // Parameters
53 // -----
54 //   resetPin(uint8_t): Вывод сброса
55 // Returns
56 // -----
57 //   NONE
58 void pca95xx_reset(int8_t resetPin)
59 {
60     if (resetPin < 0)
61         return;
62     pinMode(resetPin, PCA95XX_PIN_RESET_MODE);
63     digitalWrite(resetPin, LOW);
64     delay(1);
65     digitalWrite(resetPin, HIGH);
66     delay(1);
67 }
68
69 // Настраивает все выходы в режим OUT по умолчанию.
70 // Parameters
71 // -----
72 //   addr(uint8_t): Адрес микросхемы PCA95XX на I2C
73 //   polInv(uint8_t): Инверсия выводов (байт настройки)
74 //   io(uint8_t): Режим ввод/вывод (байт настройки)
75 // Returns
76 // -----
77 //   NONE
78 void pca95xx_configure(uint8_t addr, uint8_t polInv, uint8_t io)
79 {
80     pca95xx_i2c_sendData(addr, PCA95XX_POL_REG, polInv);
81     pca95xx_i2c_sendData(addr, PCA95XX_CONF_REG, io);
82     delay(15);
83 }
84
85 // Получает данные из PCA95XX.
86 // Parameters
87 // -----
88 //   addr(uint8_t): Адрес микросхемы PCA95XX на I2C
89 // Returns
90 // -----
91 //   uint8_t: Данные, полученные от PCA95XX
92 uint8_t pca95xx_in(uint8_t addr)
93 {
94     if (!pca95xx_i2c_receiveData(addr)) return 0u;
95     return pca95xx_i2c_buf;
96 }
97
98 // Отправляет данные в PCA95XX.
99 // Parameters
100 // -----
101 //   addr(uint8_t): Адрес микросхемы PCA95XX на I2C
102 //   data(uint8_t): Байт данных

```

```

103 // Returns
104 // -----
105 //     NONE
106 void pca95xx_out(uint8_t addr, uint8_t data)
107 {
108     pca95xx_i2c_sendData(addr, PCA95XX_OUT_REG, data);
109 }

```

Файл “main.ino”:

```

1  #include <Wire.h>
2
3  #include "pca95xx.h"
4
5  void setup()
6  {
7      Wire.begin(0);
8  }
9
10 byte lampCode0 = 0b10101010; // 0x73 // BRG
11 byte lampCode1 = 0b00000010; // 0x70
12
13 #define COM_BYTE 0x03
14 #define COM_BYTE_OUT 0x01
15
16 void loop()
17 {
18     pca95xx_configure(0x73);
19     pca95xx_out(0x73, lampCode0);
20     pca95xx_configure(0x70);
21     pca95xx_out(0x70, lampCode1);
22     delay(1000);
23     pca95xx_configure(0x73);
24     pca95xx_out(0x73, lampCode0 ^ 0xff);
25     pca95xx_configure(0x70);
26     pca95xx_out(0x70, lampCode1 ^ 0b00000011);
27     delay(1000);
28 }

```

Вывод: в ходе лабораторной работы мы изучили функциональные возможности среды разработки Arduino IDE, научились компилировать программу и загружать ее в микроконтроллер. Изучили стандартные библиотеки среды Arduino IDE для работы с портами ввода вывода, интерфейсами микроконтроллера Atmega 328p и научились применять их на практике.