

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

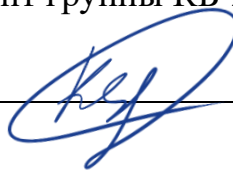
Кафедра программного обеспечения вычислительной техники и
автоматизированных систем.

Лабораторная работа работа № 5

Изучение принципа работы аналого-цифрового преобразователя.
Вариант 13

Выполнил:

Студент группы КБ-211



Коренев Д.Н.

Принял:

Шамраев А.А.

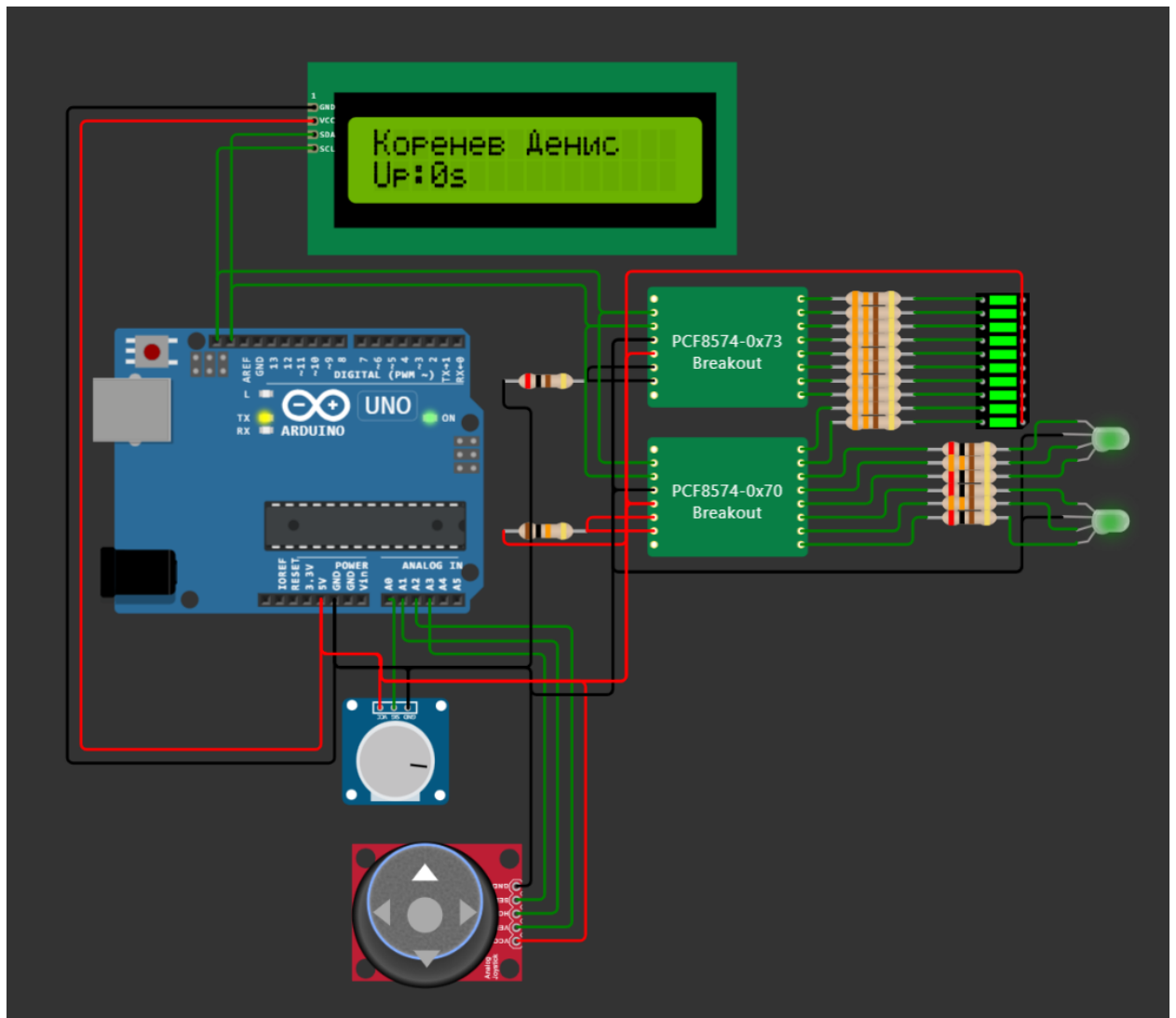
Цель работы: изучить принцип работы аналого-цифрового преобразователя (АЦП). Разработать алгоритм и программу соответствующие выбранному варианту.

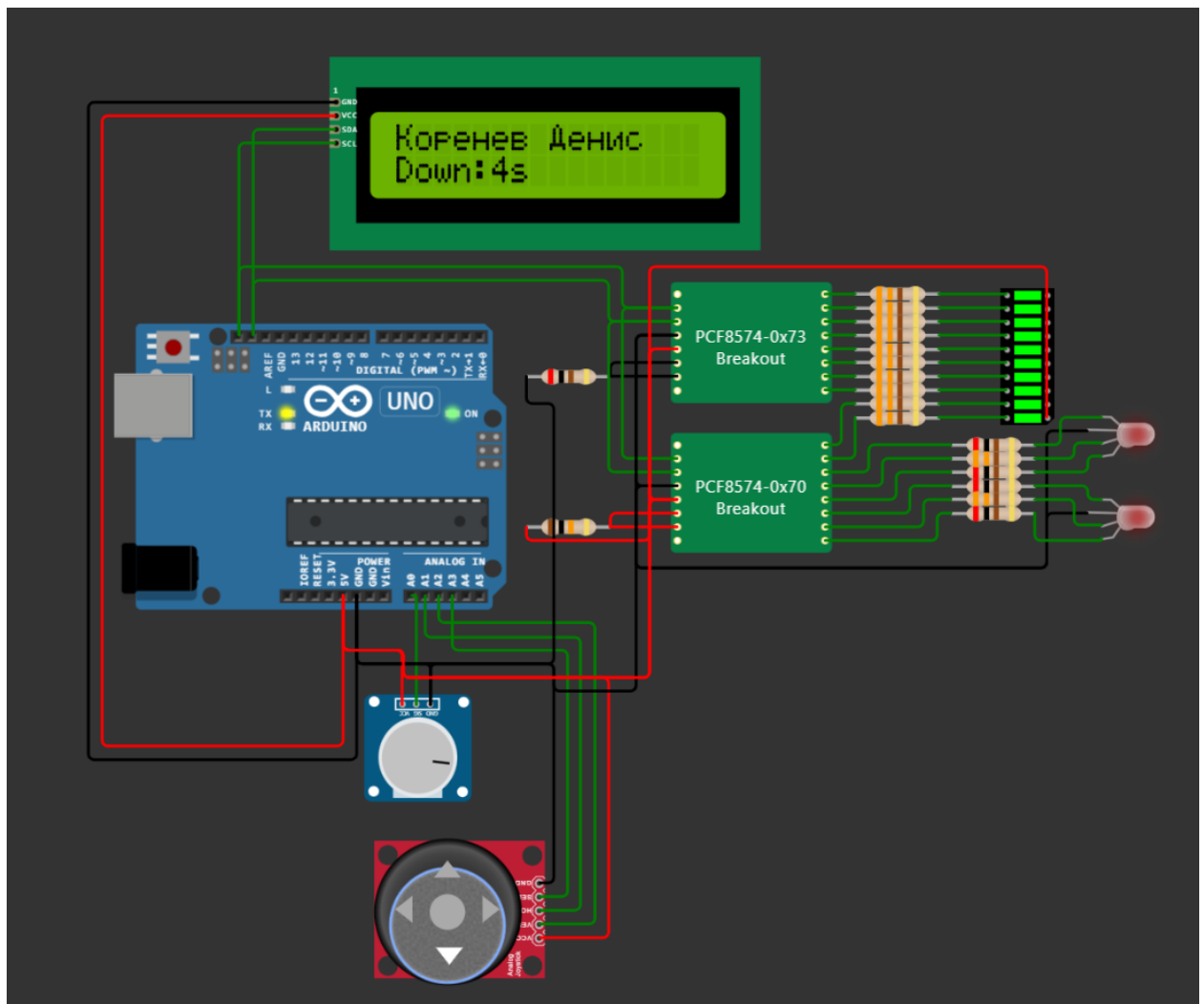
Задание

1. Используя стандартные функции для работы с АЦП, реализовать программу в соответствии с вариантом задания (номер в журнале % 2 + 1):

1) На первой строке ЖК-экрана выводить свое имя, фамилию, на второй строке положение ручки потенциометра в процентах.

2) При отклонении джойстика вниз зажигать красный светодиод, при отклонении вверх – зеленый, а при отклонении влево/вправо – мигать верхним/нижним желтым. На экране вывести свое имя, фамилию и считать время нахождения джойстика в каждом положении в секундах.





```

1  int xValue = analogRead(joyX);
2  int yValue = analogRead(joyY);
3
4  buttonState = digitalRead(button);
5
6  lcd.clear();
7  lcd.setCursor(0, 0);
8  lcd.write((uint8_t)0);
9  lcd.print("ope");
10 lcd.write((uint8_t)2);
11 lcd.print("e");
12 lcd.write((uint8_t)1);
13 lcd.print(" ");
14 lcd.write((uint8_t)4);
15 lcd.print("e");
16 lcd.write((uint8_t)2);
17 lcd.write((uint8_t)5);
18 lcd.print("c");
19
20 lcd.setCursor(0, 1);
21
22 unsigned long timeStart = millis();
23 delay(250);
24 if (yValue < 470)
25 {
26     // right
27     lcd.print("Right:");
28     lcd.print(String(timerRIGHT / 1000));
29     lcd.print("s");
30     timerRIGHT += 2.1 * (millis() - timeStart);

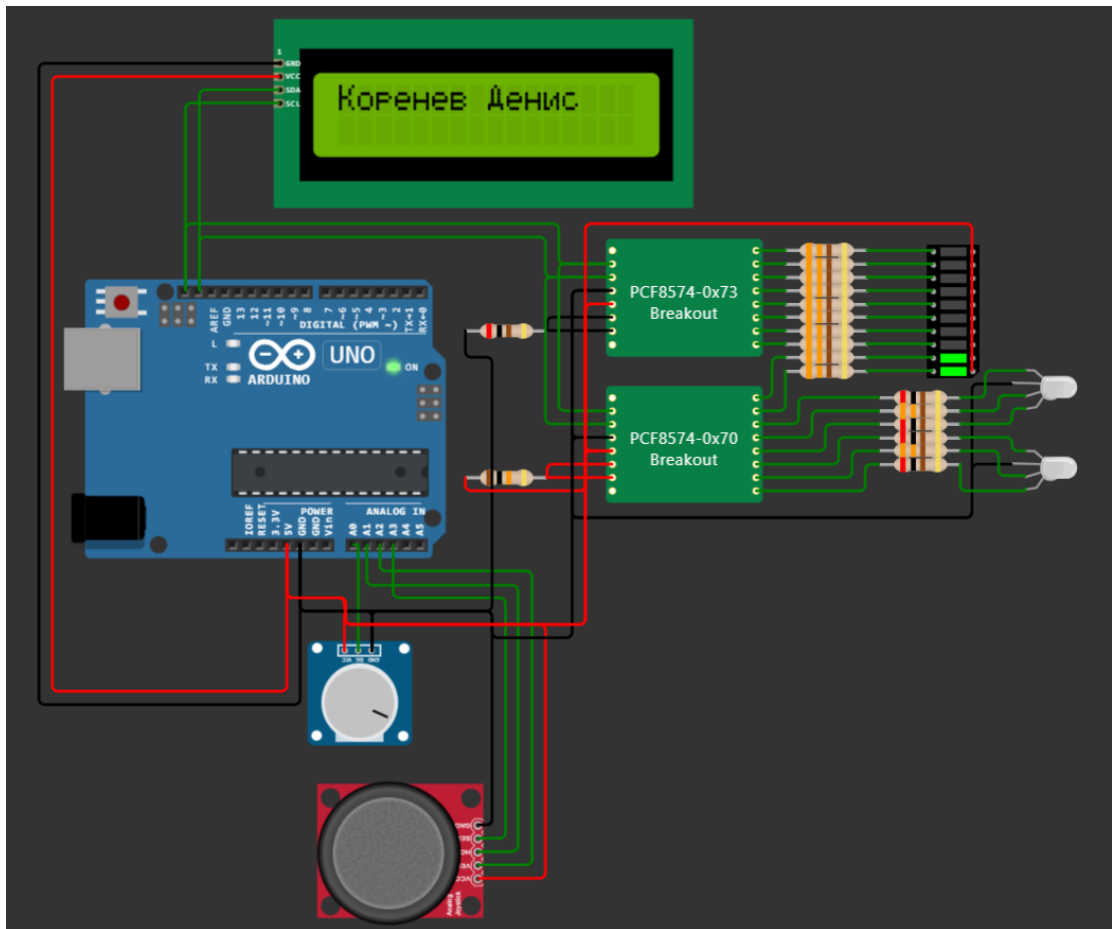
```

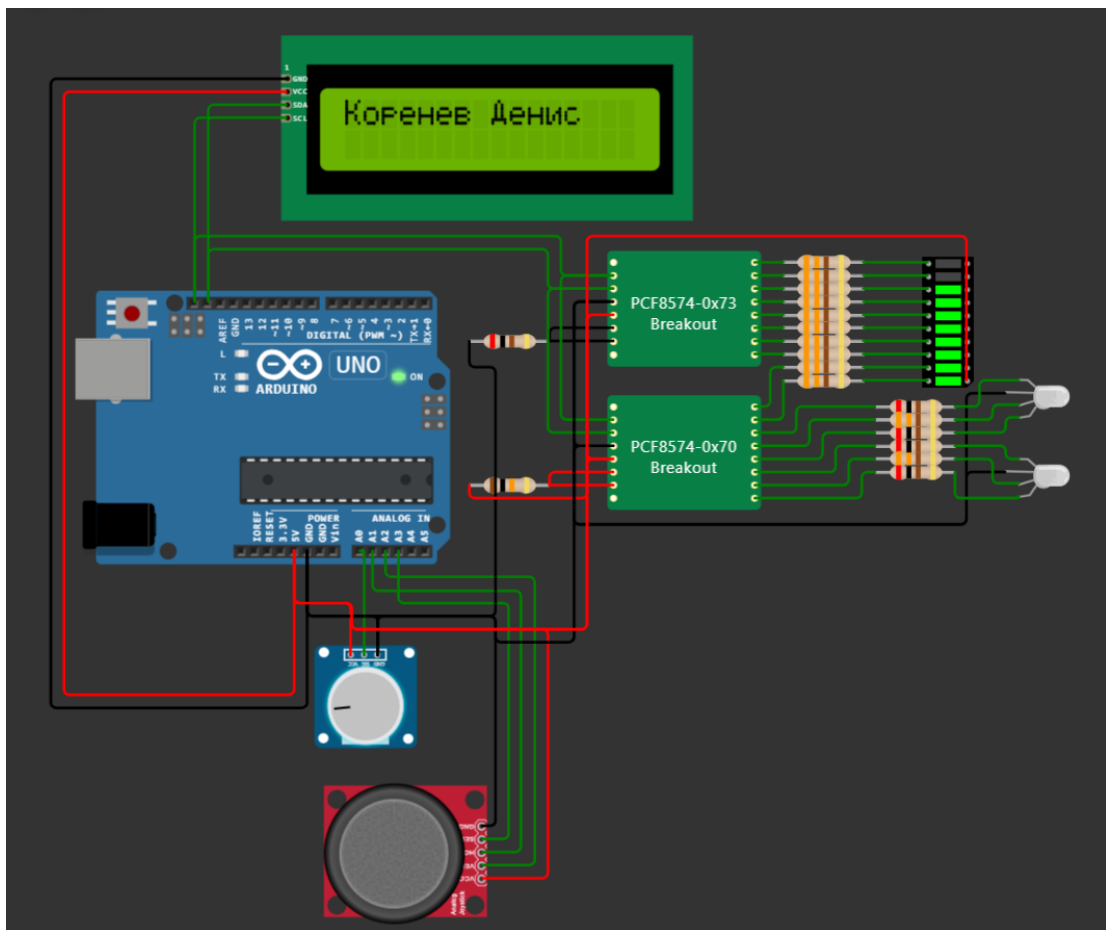
```

31     }
32     else if (yValue > 550)
33     {
34         // left
35         // int timeStart = millis();
36         lcd.print("Left:");
37         lcd.print(String(timeLEFT / 1000));
38         lcd.print("s");
39         timeLEFT += 2.1 * (millis() - timeStart);
40     }
41     else if (xValue < 470)
42     {
43         // up
44         lcd.print("Up:");
45         lcd.print(String(timeUP / 1000));
46         lcd.print("s");
47         timeUP += 2.1 * (millis() - timeStart);
48     }
49     else if (xValue > 550)
50     {
51         // down
52         lcd.print("Down:");
53         lcd.print(String(timeDOWN / 1000));
54         lcd.print("s");
55         timeDOWN += 2.1 * (millis() - timeStart);
56     }
57     else
58     {
59         // unpressed
60         lcd.print("                ");
61         timeUP = 0;
62         timeDOWN = 0;
63         timeLEFT = 0;
64         timeRIGHT = 0;
65     }
66
67     // При отклонении джойстика вниз зажигать красный светодиод, при отклонении
68     // вверх - зеленый, а
69     // при отклонении влево/вправо - мигать верхним/нижним желтым. На экране
70     // вывести свое имя,
71     // фамилию и считать время нахождения джойстика в каждом положении в
72     // секундах.
73     if (yValue < 470 || yValue > 550)
74     {
75         pca95xx_out(0x70, pca95xx_in(0x70) & RGB_ALL_REMOVE);
76         pca95xx_out(0x70, pca95xx_in(0x70) | RGB_RED_UP_ADD | RGB_GREEN_UP_ADD);
77         delay(250);
78         pca95xx_out(0x70, pca95xx_in(0x70) & RGB_ALL_REMOVE);
79         pca95xx_out(0x70, pca95xx_in(0x70) | RGB_RED_BOTTOM_ADD |
80         RGB_GREEN_BOTTOM_ADD);
81     }
82     else if (xValue < 470)
83     {
84         pca95xx_out(0x70, pca95xx_in(0x70) & RGB_ALL_REMOVE);
85         pca95xx_out(0x70, pca95xx_in(0x70) | RGB_GREEN_UP_ADD |
86         RGB_GREEN_BOTTOM_ADD);
87     }
88     else if (xValue > 550)
89     {
90         pca95xx_out(0x70, pca95xx_in(0x70) | RGB_RED_UP_ADD |
91         RGB_RED_BOTTOM_ADD);
92     }
93     else
94     {
95         pca95xx_out(0x70, pca95xx_in(0x70) & RGB_ALL_REMOVE);
96     }
97     delay(100);

```

б) Перемещать курсор на ЖК-экране с помощью джойстика влево/вправо. По нажатию на кнопку в джойстике выводить на текущем положении курсора первую букву своего имени.





```

1  int potentia = adc.read_its();
2  int potentia_mapped = map(potentia, 0, 1023, 0, 10);
3  displayLEDcolumn(potentia_mapped);
4  delay(150);
5  }
6
7  void displayLEDcolumn(int value)
8  {
9      uint16_t mask = 0b1111110000000000;
10     pca95xx_out(0x73, pca95xx_in(0x73) & (uint8_t) mask);
11     pca95xx_out(0x70, pca95xx_in(0x70) & ((uint8_t) (mask >> 8)));
12     uint16_t toOutput = 0b0;
13     for (int i = value; i > 0; i--)
14     {
15         toOutput |= (1 << (10 - i));
16     }
17     toOutput ^= 0b11111111;
18     pca95xx_out(0x73, pca95xx_in(0x73) | (uint8_t) toOutput);
19     pca95xx_out(0x70, pca95xx_in(0x70) | ((uint8_t) (toOutput >> 8)));
20 }

```

3. Взаимодействовать с АЦП напрямую с помощью регистров. Выполнить вариант задания из 2).

Проект Wokwi: <https://wokwi.com/projects/379824026508643329>

```

1  #ifndef ACD_REG_H__
2  #define ACD_REG_H__
3
4  class adc_atmega328p {
5  public:
6      void begin(){

```

```

7   DDRC = B000000; // назначает выводы с 14 по 19 входными
8   PORTC = B000000; // устанавливает HIGH на выводах с 14 по 19 (на 19 LOW)
9   ADCSRA |= (1 << ADEN); // Включаем АЦП
10  | (1 << ADPS2)|(1 << ADPS1)|(1 << ADPS0); // устанавливаем предделитель
    преобразователя на 128
11  ADMUX |= (0 << REFS1)|(1 << REFS0) // выставяем опорное напряжение Vcc
12  | (1 << MUX0)|(1 << MUX1)|(0 << MUX2)|(0 << MUX3); // снимать сигнал будем с
    входа AC3
13  analogReference(DEFAULT);
14  }
15
16  int read_its(){
17      auto pin = 0;
18      ADMUX = (pin & 0x07);
19      ADCSRA |= (0 << ADIF);
20      ADCSRA |= (1 << ADSC);
21      while ((ADCSRA & (1 << ADIF)) == 0); //ожидаем окончания преобразования
22      return (ADCL|ADCH << 8);
23  }
24
25  float read_voltage(){
26      auto pin = 0;
27      ADMUX = (pin & 0x07);
28      ADCSRA |= (0 << ADIF);
29      ADCSRA |= (1 << ADSC);
30      while ((ADCSRA & (1 << ADIF)) == 0); //ожидаем окончания преобразования
31      int data = (ADCL|ADCH << 8); // Считываем полученное значение
32      float V = (float) data * 0.0047031; // Переводим в вольты
33      /* 0.0047031 = Vcc / 1024 */
34      return V;
35  }
36  };
37
38  #endif // ACD_REG_H__

```

```

1   int potentia = adc.read_its();
2   int potentia_mapped = map(potentia, 0, 1023, 0, 10);
3   displayLEDcolumn(potentia_mapped);
4   delay(150);

```

Вывод: в ходе лабораторной работы мы изучили принцип работы аналого-цифрового преобразователя (АЦП), разработали алгоритм и программу соответствующие выбранному варианту.