

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра САПР

ОТЧЕТ
по лабораторной работе №2
по дисциплине «Проектирование цифровых устройств»
Тема: Управление временем на Arduino

Студенты гр. 0302

Хаматов В.Р.

Блюдин А.И.

Преподаватель

Рыбин В.Г.

Санкт-Петербург

2023

1 Цель работы

Реализовать гирлянду, управление временем в которой производится (переключение режимов по кнопке) альтернативными способами: delay, millis, timer (прерывания).

2 Ход выполнения работы

2.1 Сборка схемы

Для выполнения данной лабораторной работы нам потребуется:

1. 3 светодиода (красный, желтый, зеленый).
2. 3 резистора на 220 Ом.
3. 1 резистор на 10 Ком.
4. Контактная кнопка.
5. 8 проводов.

Чтобы ограничить ток, подаваемый на светодиод, подключим последовательно к каждому светодиоду резистор на 220 Ом. Для подключения кнопки возьмем резистор на 10Ком, таким образом мы избавимся от дребезга и работа кнопки будет более стабильная. Соберем принципиальную схему в Tinkercard (Рис. 3.1.1)

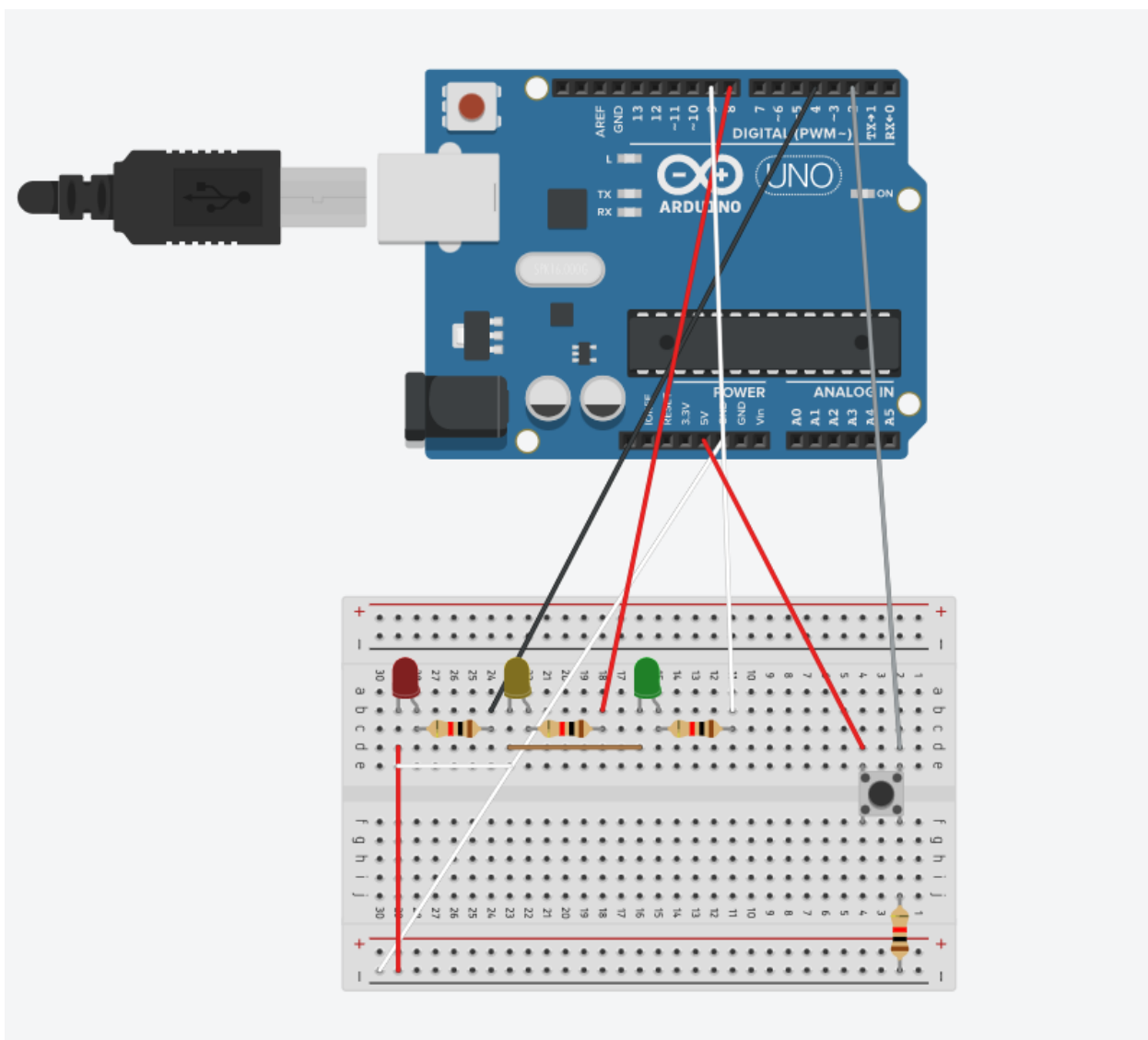


Рис. 3.1.1 — Принципиальная схема устройства

Далее по принципиально схеме устройства соберем физическую схему устройства (Рис. 3.1.2)

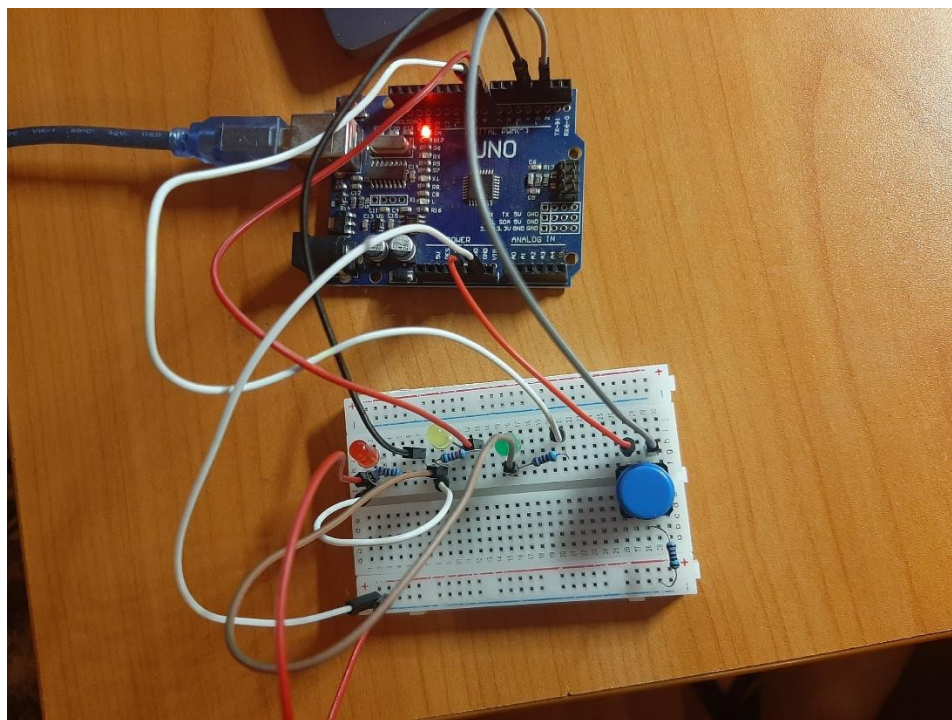


Рис. 3.1.2. — Физическая схема устройства.

2.2 Принцип работы цифрового устройства

Управление временем с помощью `delay()`

Сначала разберемся, что такое `delay` – это пауза, но не в работе скетча, а всей Arduino. `Delay()` практически парализуют ардуину останавливая все процессы. `Delay()` больше всего подходит для маленьких и тестовых скетчей.

Далее приступим к написанию кода, для начала сконфигурируем все необходимые нам пины:

```
1lab_sketch | Arduino 1.8.19
Файл Правка Скетч Инструменты Помощь
1lab_sketch §
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
  //attachInterrupt(2, stateMove, CHANGE);
  pinMode(4, OUTPUT);
  pinMode(8, OUTPUT);
  pinMode(9, OUTPUT);
  pinMode(2, INPUT);
}
```

Рис. 3.2.1 — Конфигурация пинов

Создадим глобальную функцию stateMove(), которая будет менять свое состояние по нажатию кнопки(Рис. 3.2.2).

```
int state = LOW;
int stateMove() {
    state = !state;
    return state;
}
```

Рис. 3.2.2 — Функция stateMove();

Напишем основной цикл loop (Рис. 3.2.3). В нем сначала происходит проверка нажатия на кнопку и по результатам проверки меняется состояние state. Далее в зависимости от состояния переменной state происходит перевод состояния пинов, к которым подключены светодиоды. В данной функции используется delay(200). Он означает что программа просто замрет на 200 миллисекунд. Это нужно для считывания параметров с кнопки.

```
// the loop function runs over and over again forever
void loop() {

    if(digitalRead(2)) {
        stateMove();
        delay(200);
    }
    if (state == LOW) {
        digitalWrite(4,HIGH);
        digitalWrite(8,HIGH);
        digitalWrite(9,HIGH);
    } else if(state == HIGH) {
        digitalWrite(4,LOW);
        digitalWrite(8,LOW);
        digitalWrite(9,LOW);
    }
}
```

Рис. 3.2.3 — Loop

Управление временем с помощью millis()

Функция возвращает количество миллисекунд с момента начала подачи питания на плату Ардуино. И не важно, что делает наша программа - миллисекунды отсчитываются постоянно и без перебоев. Из минусов – это

переполнение. При наступлении 49 дней работы, функция обнулиться и начнет отсчет заново.

Число 49 дней взялось из размерности переменной long int. Оно может хранить число не более 4,294,967,295. А в 49 сутках 4,233,600,000 миллисекунд. Это предел значения. Расчет приведенных данных изображен на (Рис. 3.2.4):

Конвертер величин			
Мера времени ▾			
сутки ▾	49	=	4 233 600 000 мс ▾
char	8		−127 to 127
unsigned char	8		0 to 255
signed char	8		−127 to 127
int	16		−32,767 to 32,767
unsigned int	16		0 to 65,535
signed int	16		−32,767 to 32,767
short int	16		−32,767 to 32,767
unsigned short int	16		0 to 65,535
signed short int	16		−32,767 to 32,767
long int	32		−2,147,483,647 to 2,147,483,647
long long int	64		−(2 ⁶³ − 1) to 2 ⁶³ − 1 (Added by C99)
signed long int	32		−2,147,483,647 to 2,147,483,647
unsigned long int	32		0 to 4,294,967,295

Рис. 3.2.4 — Расчет данных о переполнении функции millis

Приступим к написанию кода. Создадим переменную prev_ms, которая будет помогать вычислять разницу во времени между нажатиями кнопки. Так же создадим переменную On, которая будет содержать в себе значение в миллисекундах, которое необходимо ожидать. В функции setup сконфигурируем пины 4,8,9 в режиме output, которые будут отвечать за светодиоды, пин 2 для считывания нажатия кнопки (Рис. 3.2.5).

```

unsigned long prev_ms = 0;
const int On = 200;
void setup() {
    pinMode(4, OUTPUT);
    pinMode(8, OUTPUT);
    pinMode(9, OUTPUT);
    pinMode(2, INPUT);
}

```

Рис. 3.2.5 — Объявление переменных

В функции loop создадим переменную currentTime и будем записывать в нее текущее значение функции millis (количество секунд с включения микроконтроллера). Далее ставим условие на нажатие кнопки (digitalRead(2)), а так же, в отличие от программы с использованием delay() проверяем разницу между переменными currentTime и prev_ms и сравниваем ее со значением переменной On. Данное выражение позволяет нам заменить функцию delay(200) из прошлого скетча, но уже наша программа не будет обладать минусом функции delay() – остановкой микроконтроллера (Рис. 3.2.6).

```
// the loop function runs over and over again forever
void loop() {

    unsigned long currentTime = millis(); // текущее время в миллисекундах

    if((digitalRead(2)) && (currentTime - prev_ms >= On)){
        stateMove();
        prev_ms = currentTime;
    }
    if (state == LOW) {
        digitalWrite(4,HIGH);
        digitalWrite(8,HIGH);
        digitalWrite(9,HIGH);
    } else if(state == HIGH) {
        digitalWrite(4,LOW);
        digitalWrite(8,LOW);
        digitalWrite(9,LOW);
    }
}
```

Рис. 3.2.6 — Функция loop

3 Контрольные вопросы

1. В чем разница между предлагаемыми способами управления временем?

Охарактеризуйте каждый.

Delay – это пауза, но не в работе скетча, а всей Arduino. Delay() практически парализуют ардуину останавливая все процессы. Delay() больше всего подходит для маленьких и тестовых скетчей.

Функция millis() возвращает количество миллисекунд с момента начала подачи питания на плату Ардуино. И не важно, что делает наша программа - миллисекунды отсчитываются постоянно и без перебоев. Из минусов – это переполнение. При наступлении 49 дней работы, функция обнулится и начнет отсчет заново.

Число 49 дней взялось из размерности переменной long int. Оно может хранить число не более 4,294,967,295. А в 49 сутках 4,233,600,000 миллисекунд. Это предел значения.

4 Вывод

В результате выполнения данной лабораторной работы были изучены способы работы со временем на микроконтроллере Arduino, а именно с использованием функций delay() и millis().

5 Листинг

Скетч с использованием delay()

```
void setup() {
  pinMode(4, OUTPUT);
  pinMode(8, OUTPUT);
  pinMode(9, OUTPUT);
  pinMode(2, INPUT);
}

int state = LOW;
int stateMove(){
  state = !state;
  return state;
}

// the loop function runs over and over again forever
void loop() {

  if(digitalRead(2)){
    stateMove();
    delay(200);
  }
  if (state == LOW) {
    digitalWrite(4,HIGH);
    digitalWrite(8,HIGH);
    digitalWrite(9,HIGH);
  } else if(state == HIGH) {
    digitalWrite(4,LOW);
    digitalWrite(8,LOW);
    digitalWrite(9,LOW);
  }
}
```

Скетч с использованием millis():

```
unsigned long prev_ms = 0;
```



```

const int On = 200;
void setup() {
  pinMode(4, OUTPUT);
  pinMode(8, OUTPUT);
  pinMode(9, OUTPUT);
  pinMode(2, INPUT);
}

int state = LOW;
int stateMove(){
  state = !state;
  return state;
}

// the loop function runs over and over again forever
void loop() {

  unsigned long currentTime = millis(); // текущее время в миллисекундах

  if((digitalRead(2)) && (currentTime - prev_ms >= On)){
    stateMove();
    prev_ms = currentTime;
  }
  if (state == LOW) {
    digitalWrite(4,HIGH);
    digitalWrite(8,HIGH);
    digitalWrite(9,HIGH);
  } else if(state == HIGH) {
    digitalWrite(4,LOW);
    digitalWrite(8,LOW);
    digitalWrite(9,LOW);
  }

}

```