



**«Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)»**

---

**ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ**

**КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ**

---

### **Задание к лабораторной работе №6**

по курсу «Основы микропроцессорной техники»

«Сравнительный анализ архитектур микроконтроллеров на примере ATmega328,  
STM32 и ESP32»

# 1 Тест производительности

## 1.1 ATmega328

Создайте проект на Arduino Uno на сайте <https://wokwi.com>, выбрав плату Arduino Uno.

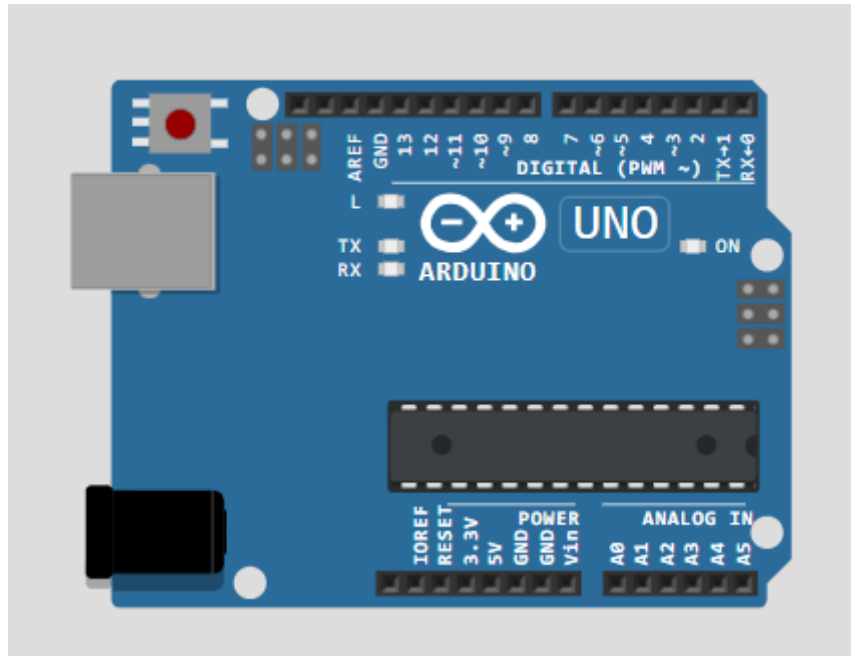


Рисунок 1 – Arduino Uno

Далее необходимо повторить 2 и 3 пункты из раздела 1.2.4 Лабораторной работы №1. Также добавьте в проект следующий файл и код:

### **main.cpp**

```
#include <Arduino.h>
#include <math.h>

void testPerformance() {
    unsigned long startTime = micros();
    volatile long result = 0;
    for (volatile long i = 0; i < 10000L; i++) {
        result += i * i - i / 3 + sqrt(i);
    }

    unsigned long endTime = micros();
    Serial.print("Execution time (us): ");
    Serial.println(endTime - startTime);
    Serial.print("Result: "); Serial.println(result);
}

void setup() {
```

```

Serial.begin(115200);
Serial.println("=== ATmega328 Performance Test ===");
Serial.println("Starting performance test...");
}

void loop() {
    testPerformance();
    Serial.println("Test completed. Waiting 5 seconds...");
    Serial.println();
    delay(5000);
}

```

## 1.2 ESP32

Создайте проект на Arduino Uno на сайте <https://wokwi.com>, выбрав плату ESP32.



Рисунок 2 – ESP32

Далее необходимо повторить 2 и 3 пункты из раздела 1.2.4 Лабораторной работы №1. Добавьте в проект следующий файл и код:

```

main.cpp

#include <Arduino.h>
#include <math.h>

void testPerformance() {
    unsigned long startTime = micros();
    volatile long result = 0;

    for (volatile long i = 0; i < 10000L; i++) {

```

```

    result += i * i - i / 3 + sqrt(i);
}

unsigned long endTime = micros();
Serial.print("Execution time (us): ");
Serial.println(endTime - startTime);
Serial.print("Result: "); Serial.println(result);
}

void setup() {
    Serial.begin(115200);
    Serial.println("=== ESP32 Performance Test ===");
    Serial.println("XTAL Frequency: 40MHz");
    Serial.println("CPU Frequency: 240MHz");
    Serial.println("Dual Core: Yes");
    Serial.println("Starting performance test...");
    Serial.println();
}

void loop() {
    testPerformance();
    Serial.println("Test completed. Waiting 5 seconds...");
    Serial.println();
    delay(5000);
}

```

### 1.3 STM32

Создайте проект на Arduino Uno на сайте <https://wokwi.com>, выбрав плату STM32.

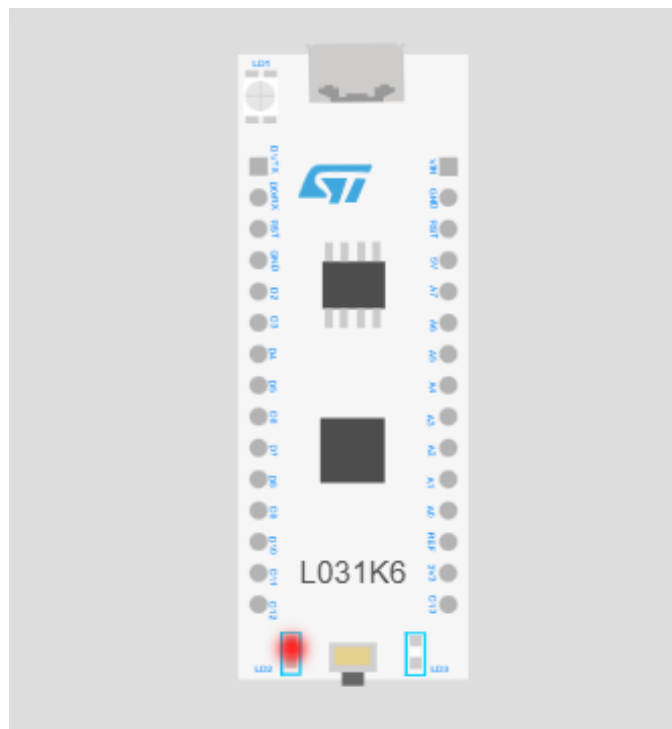


Рисунок 3 – STM32

Далее необходимо повторить 2 и 3 пункты из раздела 1.2.4 Лабораторной работы №1. Добавьте в проект следующий файл и код:

### **main.cpp**

```
#include <Arduino.h>
#include <math.h>

void testPerformance() {
    unsigned long startTime = micros();
    volatile long result = 0;
    for (volatile long i = 0; i < 10000L; i++) {
        result += i * i - i / 3 + sqrt(i);
    }

    unsigned long endTime = micros();
    Serial.print("Execution time (us): ");
    Serial.println(endTime - startTime);
    Serial.print("Result: "); Serial.println(result);
}

void setup() {
    Serial.begin(115200);
    Serial.println("=== STM32 Performance Test ===");
    Serial.println("CPU: ARM Cortex-M3");
    Serial.println("Frequency: 72MHz");
    Serial.println("FPU: No (Software Floating Point)");
    Serial.println("Starting performance test...");
    Serial.println();
}

void loop() {
    testPerformance();
    Serial.println("Test completed. Waiting 5 seconds...");
    Serial.println();
    delay(5000);
}
```

## **1.4 Задание**

Студенту необходимо создать и собрать проект, разобраться в логике работы системы, проверить корректность работы на симуляции, после чего ответить на вопросы преподавателя по данной части лабораторной работы.

Полученные числовые результаты проанализировать и сравнить (определить лучший и худший результаты и причину таких значений). Сравнение представить в отчете.

## 2 Тест скорости переключения GPIO

### 2.1 ATmega328

Используйте проект из первой части методических указаний для Arduino Uno. Добавьте в проект следующий файл и код:

#### **main.cpp**

```
#include <Arduino.h>

const int PINS[] = {2, 3, 4, 5, 6, 7, 8, 9};
const int NUM_PINS = 8;

void testGPISpeed() {
    Serial.println("=== Тест скорости GPIO ===");

    // Настройка пинов как выходов
    for (int i = 0; i < NUM_PINS; i++) {
        pinMode(PINS[i], OUTPUT);
    }

    unsigned long startTime = micros();
    unsigned long cycles = 100000;

    for (unsigned long i = 0; i < cycles; i++) {
        for (int pin = 0; pin < NUM_PINS; pin++) {
            digitalWrite(PINS[pin], HIGH);
        }
        for (int pin = 0; pin < NUM_PINS; pin++) {
            digitalWrite(PINS[pin], LOW);
        }
    }

    unsigned long endTime = micros();
    unsigned long duration = endTime - startTime;
    unsigned long totalToggles = cycles * NUM_PINS * 2;

    Serial.println("=== РЕЗУЛЬТАТЫ ===");
    Serial.print("Циклов: "); Serial.println(cycles);
    Serial.print("Время: "); Serial.print(duration); Serial.println(" мкс");
    Serial.print("Всего переключений: "); Serial.println(totalToggles);
    Serial.print("Скорость: "); Serial.print(totalToggles / (duration / 1000000.0));
    Serial.println(" переключений/сек");
    Serial.print("Время одного переключения: ");
    Serial.print((double)duration / totalToggles, 3); Serial.println(" мкс");
}

void setup() {
    Serial.begin(115200);
    delay(1000);
}
```

```

Serial.println("Starting GPIO Speed Test...");

testGPISpeed();
Serial.println("Тест завершен!");
}

void loop() {
    digitalWrite(13, HIGH);
    delay(500);
    digitalWrite(13, LOW);
    delay(500);
}

```

Выполнение может занять некоторое время из-за большого количества итераций. Дождитесь вывода результатов и зафиксируйте их.

## 2.2 Задание

Студенту необходимо создать и собрать проект, разобраться в логике работы системы, проверить корректность работы на симуляции. Далее аналогично части первой методических указаний необходимо реализовать программу для определения скорости переключения GPIO для проектов с использованием ESP32 и STM32. Результаты зафиксировать и сравнить.

Основываясь на технических характеристиках, ожидаются следующие замеры скорости переключения GPIO:

- ESP32: ~0.01-0.05 мкс (самый быстрый),
- STM32: ~0.05-0.2 мкс,
- Arduino UNO: ~0.1-0.5 мкс (самый медленный).

Сравнить эталонные значения и полученные экспериментально на симуляции. При наличии расхождений выяснить причину.

## 3 Отчет к лабораторной работе

Составьте отчет к лабораторной работе.

Требования:

1. Описание лабораторной работы и основных этапов её выполнения, с соответствующими скриншотами.

2. Выполнение самостоятельного задания со скриншотами и результатами измерений.
3. Ответы на вопросы.
4. Выводы к лабораторной работе.

#### **4 Контрольные вопросы**

1. Что означает volatile?
2. Какие важные характеристики системы демонстрируются измерениями скорости вычислений, GPIO?
3. Почему скорость GPIO разная у разных мк? Какие факторы влияют на скорость?
4. За что отвечает функция micros()?