



**«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 №3

по курсу «Основы микропроцессорной техники»

«Организация обмена данными через UART»

СОДЕРЖАНИЕ

1 Подключение по UART к логическому анализатору.....	3
1.1 Схема подключения	3
1.2 Необходимые файлы	4
1.3 Структура сообщения UART.....	4
1.4 Задание	6
2 Использование UART для организации связи МК.....	6
2.1 Схема подключения.....	6
2.2 Необходимые файлы.....	7
2.3 Задание	10
2.4 Самостоятельная работа	11
2.4.1 Анализ диаграммы сигналов по индивидуальному варианту.....	11
2.4.2 Реализация "светофора" на двух микроконтроллерах с помощью UART.....	11
3 Отчет к лабораторной работе.....	11
4 Контрольные вопросы	12

1 Подключение по UART к логическому анализатору

Цель: освоить подключение и работу с протоколом передачи данных UART.

Понять особенности протокола.

Задачи:

1. Изучить структуру сообщений UART.
2. Подключиться по UART к логическому анализатору и расшифровать получаемые сообщения.
3. Организовать связь и передачу сообщений между двумя микроконтроллерами с помощью UART.
4. Реализовать светофор на основе кода предыдущих заданий.

1.1 Схема подключения

Создайте проект на Arduino Uno на сайте <https://wokwi.com>. На макете необходимо разместить, в соответствии со схемой ниже, логический анализатор (logic analyzer).

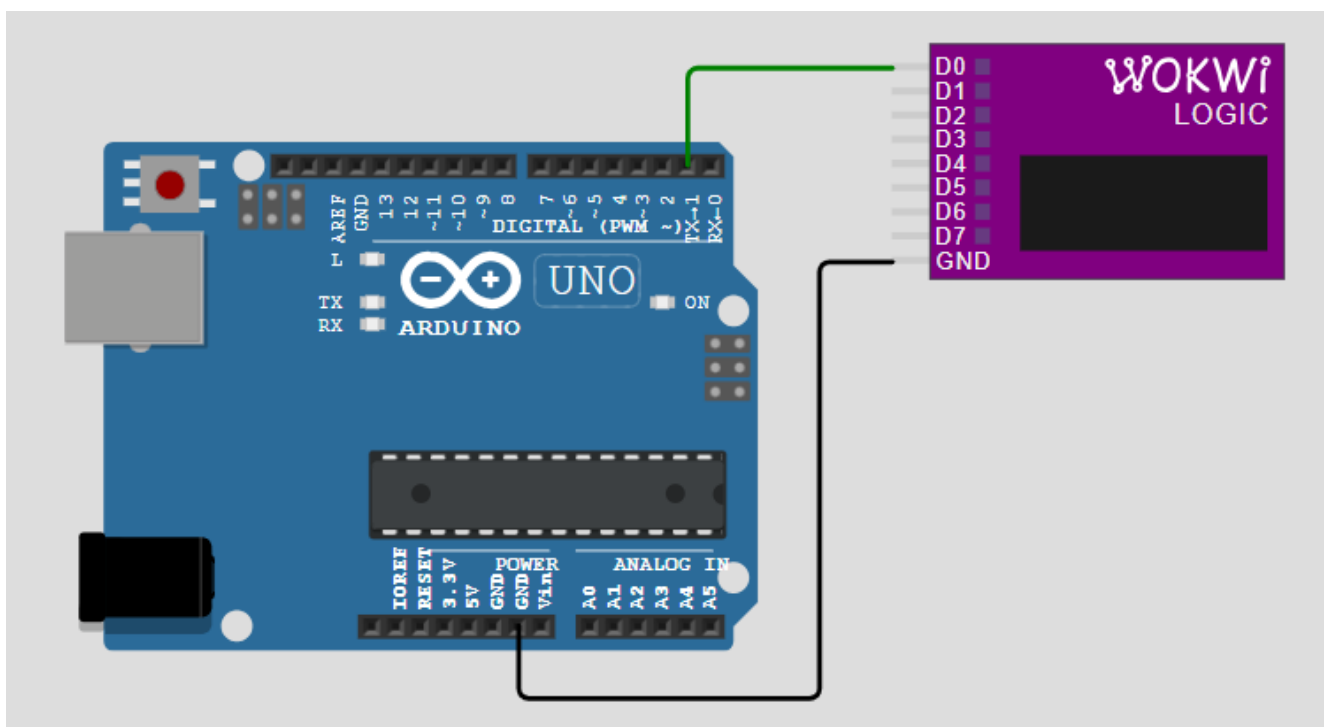


Рисунок 1 – Схема подключения

Далее необходимо повторить 2 и 3 пункты из раздела 1.2.4 Лабораторной работы №1.

1.2 Необходимые файлы

Добавьте в проект следующий файл и код:

main.cpp

```
#include <Arduino.h>

void setup() {
  Serial.begin(115200);
}

void loop() {
  Serial.println("I love UART");
  delay(1000);
}
```

После успешного запуска симуляции работы макета и ее завершения, необходимо сохранить файл с расширением .vcd, необходимо сохранить его в корне проекта (либо в другом удобном месте).

Далее двойным щелчком требуется открыть этот файл и перейти в раздел Vaporview (его установка была описана в методических указаниях к Лабораторной работе №1). В боковом меню будут отображаться Netlist и Displayed signals, разверните Netlist и выберете D0 (требуется поставить галочку). После этого на экране появится записанный сигнал D0, пролистните в самое начало записи, к постоянной 1.

1.3 Структура сообщения UART

Каждое сообщение UART включает:

- стартовый бит — сигнализирует приёмнику о начале посылки. всегда имеет значение логического 0;
- биты данных — пользовательские данные, идут сразу после стартового бита. может быть от 5 до 9 битов, чаще всего — 7 или 8 битов;
- необязательный бит чётности — вставляется между последним битом данных и стоповым битом;

- стоповый бит — указывает на окончание пользовательских данных. может быть 1, 1,5 или 2 стоп-бита, чаще всего — 1.

В данном примере происходит отправка сообщения "I love UART". Ниже представлен фрагмент диаграммы, считанной с логического анализатора. Изображены первые два символа сообщения - буква "I" и пробел.



Рисунок 2 – Пример расшифровки сообщения

Пользовательские данные шифруются кодами таблицы ASCII. Ниже приведена таблица.

Таблица 1. Стандартная часть кода ANSI (ASCII)

32		00100000	56	8	00111000	80	P	01010000	104	h	01101000
33	!	00100001	57	9	00111001	81	Q	01010001	105	i	01101001
34	"	00100010	58	:	00111010	82	R	01010010	106	j	01101010
35	#	00100011	59	;	00111011	83	S	01010011	107	k	01101011
36	\$	00100100	60	<	00111100	84	T	01010100	108	l	01101100
37	%	00100101	61	=	00111101	85	U	01010101	109	m	01101101
38	&	00100110	62	>	00111110	86	V	01010110	110	n	01101110
39	'	00100111	63	?	00111111	87	W	01010111	111	o	01101111
40	(	00101000	64	@	01000000	88	X	01011000	112	p	01110000
41	)	00101001	65	A	01000001	89	Y	01011001	113	q	01110001
42	*	00101010	66	B	01000010	90	Z	01011010	114	r	01110010
43	+	00101011	67	C	01000011	91	[	01011011	115	s	01110011
44	,	00101100	68	D	01000100	92	\	01011100	116	t	01110100
45	-	00101101	69	E	01000101	93	]	01011101	117	u	01110101
46	.	00101110	70	F	01000110	94	^	01011110	118	v	01110110
47	/	00101111	71	G	01000111	95	_	01011111	119	w	01110111
48	0	00110000	72	H	01001000	96	`	01100000	120	x	01111000
49	1	00110001	73	I	01001001	97	a	01100001	121	y	01111001
50	2	00110010	74	J	01001010	98	b	01100010	122	z	01111010
51	3	00110011	75	K	01001011	99	c	01100011	123	{	01111011
52	4	00110100	76	L	01001100	100	d	01100100	124		01111100
53	5	00110101	77	M	01001101	101	e	01100101	125	}	01111101
54	6	00110110	78	N	01001110	102	f	01100110	126	~	01111110
55	7	00110111	79	O	01001111	103	g	01100111	127	□	01111111

Рисунок 3 – Таблица ASCII

Как можно заметить, коды символов в сообщении представлены в обратном порядке, то есть передача битов данных происходит начиная с наименее значимого бита. Поэтому при расшифровке сообщений через диаграммы логического анализатора следует расшифровывать символы справа налево.

Один символ передается за 8500 нс при скорости передачи 115200 бод (скорость передачи символов линии в секунду). Существует перечень стандартных скоростей UART для Arduino, 115200 бод - основное значение для высокоскоростной передачи в современном стандарте.

1.4 Задание

Проанализируйте диаграмму сообщения и разберитесь в принципе формирования сигналов.

2 Использование UART для организации связи МК

2.1 Схема подключения

Для выполнения данного пункта необходимо запрограммировать два микроконтроллера, поэтому создайте пару новых проектов на Arduino Uno на сайте <https://wokwi.com>. На макете для первого МК необходимо разместить кнопку, в соответствии со схемой ниже.

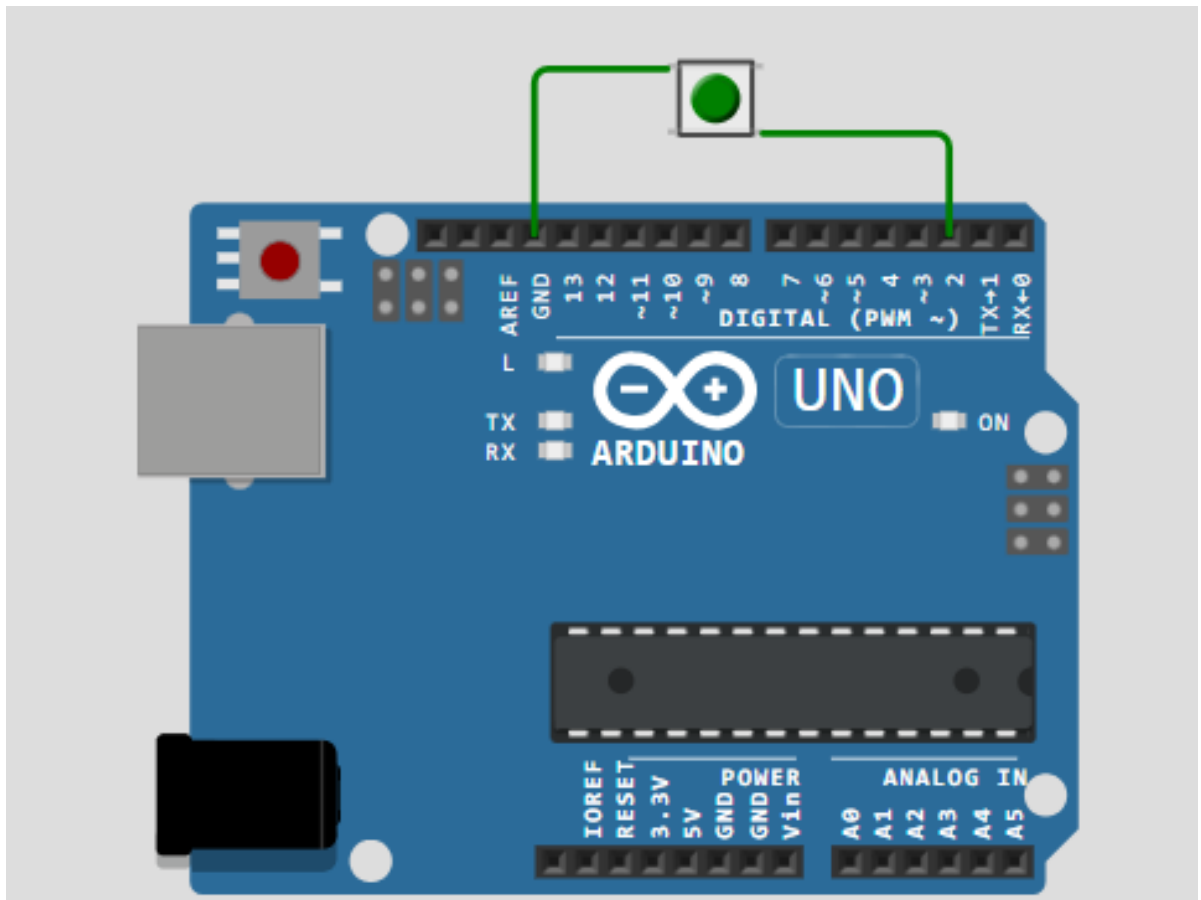


Рисунок 4 – Макет первого МК

Макет второго МК выглядит следующим образом:

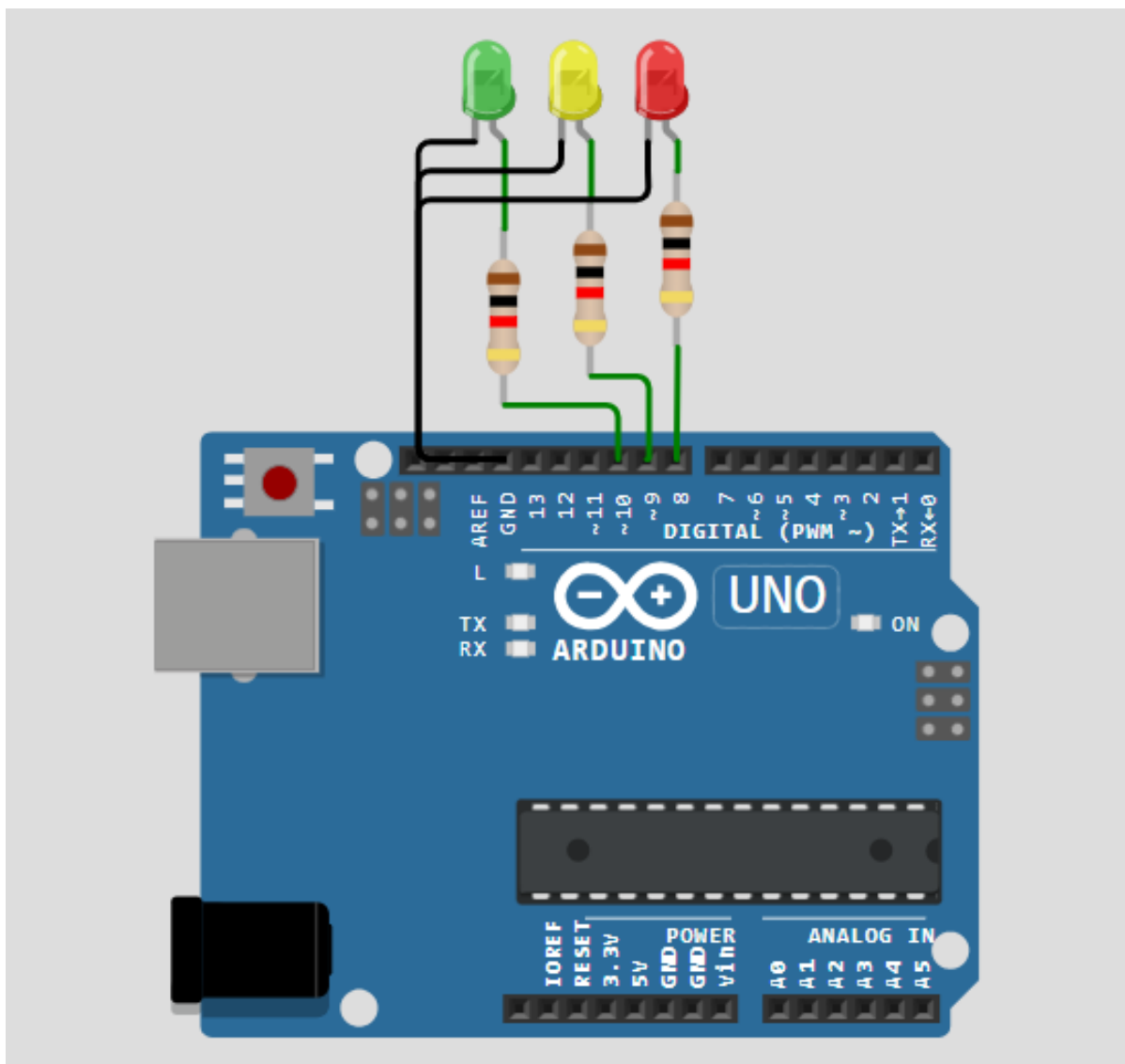


Рисунок 5 – Макет второго МК

Далее в обоих проектах необходимо повторить 2 и 3 пункты из раздела 1.2.4 Лабораторной работы №1. Однако в файле **platformio.ini** необходимо добавить 2 новые строки, отвечающие за скорость обмена данными в Serial Monitor и скорость загрузки прошивки:

```
monitor_speed = 115200
```

```
upload_speed = 115200
```

2.2 Необходимые файлы

Добавьте в проект первого микроконтроллера следующий файл и код:

main.cpp

```

#include <Arduino.h>

const int buttonPin = 2;
const int ledPin = LED_BUILTIN;

int buttonState = 0;
int lastButtonState = 0;
bool ledState = false;
unsigned long previousMillis = 0;
const long blinkInterval = 500; // Интервал 500 мс

bool buttonActive = false;
unsigned long pressTime = 0;
const long pressDelay = 1000; // 1 секунда

unsigned long messageLedTime = 0;
bool messageReceived = false;
const long messageDisplayTime = 1000;

void checkIncomingData() {
    if (Serial.available() > 0) {
        String message = Serial.readStringUntil('\n');
        message.trim();

        if (message == "DATA_FROM_MK2") {
            messageReceived = true;
            messageLedTime = millis();

            digitalWrite(ledPin, HIGH);
        }
    }
}

void setup() {
    pinMode(buttonPin, INPUT_PULLUP);
    pinMode(ledPin, OUTPUT);
    Serial.begin(115200);

    Serial.println("First Arduino is ready");
}

void loop() {
    unsigned long currentMillis = millis();

    checkIncomingData();

    if (!messageReceived && currentMillis - previousMillis >= blinkInterval) {
        previousMillis = currentMillis;
        ledState = !ledState;
        digitalWrite(ledPin, ledState);
    }
}

```



```

if (messageReceived && currentMillis - messageLedTime >= messageDisplayTime) {
    messageReceived = false;
    digitalWrite(ledPin, LOW);
}

if (digitalRead(buttonPin) == LOW) {
    if (!buttonActive && (currentMillis - pressTime > pressDelay)) {
        buttonActive = true;
        // Отправка данных
        Serial.println("BUTTON_PRESSED");
        pressTime = currentMillis;
    }
} else {
    buttonActive = false;
}
}

```

Во второй проект для другого МК требуется добавить следующий файл и код:

main.cpp

```

#include <Arduino.h>

const int redLedPin = 8;
const int greenLedPin = 10;
const int yellowLedPin = 9;

unsigned long previousSendTime = 0;
const long sendInterval = 10000; // 10 секунд

unsigned long previousYellowBlink = 0;
const long yellowBlinkInterval = 15000; // 15 секунд
bool yellowLedState = false;

void setup() {
    pinMode(redLedPin, OUTPUT);
    pinMode(greenLedPin, OUTPUT);
    pinMode(yellowLedPin, OUTPUT);

    Serial.begin(115200);
    Serial.println("Second Arduino is ready");

    previousSendTime = millis();
    previousYellowBlink = millis();
}

void loop() {
    unsigned long currentMillis = millis();

```

```

    if (currentMillis - previousSendTime >= sendInterval) {
        previousSendTime = currentMillis;
        sendData();
    }

    if (currentMillis - previousYellowBlink >= yellowBlinkInterval) {
        previousYellowBlink = currentMillis;
        blinkYellow();
    }

    checkIncomingData();
}

void sendData() {
    // Отправка данных
    Serial.println("DATA_FROM_MK2");

    digitalWrite(greenLedPin, HIGH);
    delay(500);
    digitalWrite(greenLedPin, LOW);
}

void blinkYellow() {
    digitalWrite(yellowLedPin, HIGH);
    delay(500);
    digitalWrite(yellowLedPin, LOW);
}

void checkIncomingData() {
    if (Serial.available() > 0) {
        String message = Serial.readStringUntil('\n');
        message.trim();

        if (message == "BUTTON_PRESSED") {
            digitalWrite(redLedPin, HIGH);
            delay(2000);
            digitalWrite(redLedPin, LOW);
        }
    }
}

```

2.3 Задание

Студенту необходимо создать и собрать два проекта, разобраться в логике работы каждого из них. Далее требуется прошить физическое устройство и проверить корректность работы системы.

2.4 Самостоятельная работа

2.4.1 Анализ диаграммы сигналов по индивидуальному варианту

Требуется модернизировать первую часть лабораторной работы таким образом, чтобы в Serial Monitor выводилась строка с фамилией студента. После запуска симуляции необходимо произвести расшифровку диаграммы сигналов, так как это было объяснено в примере (см. Рисунок 2).

2.4.2 Реализация "светофора" на двух микроконтроллерах с помощью UART

Реализуйте имитацию работы светофора модернизировав программу из второй задания данной лабораторной работы. При нажатии кнопки на первом микроконтроллере происходит отправка команды на второй МК, который в ответ запускает цикл работы светофора. В исходном состоянии горит красный диод, который при получении команды сменяется желтым, затем тот гаснет и далее включается зеленый диод на 5 секунд. После этого он начинает мигать три раза и гаснет, затем оранжевый мигает также три раза и отключается, и в завершение загорается постоянный красный сигнал. Когда весь цикл светофора завершен, на первый микроконтроллер поступает сообщение об успешной активации системы.

3 Отчет к лабораторной работе

Составьте отчет к лабораторной работе.

Требования:

1. Описание лабораторной работы и основных этапов её выполнения, с соответствующими скриншотами.
2. Ответы на вопросы.
3. Выполнение самостоятельного задания со скриншотами и результатами измерений.
4. Выводы к лабораторной работе.

4 Контрольные вопросы

1. Почему при использовании UART соединяется GND приёмника и передатчика?
2. Почему при использовании UART необходимо задавать общую скорость приема и передачи?
3. Какими характеристиками обладает UART как асинхронный приемопередатчик?
4. Для чего используется контрольная сумма в пересылаемом по UART пакете данных?
5. Что такое дребезг кнопки? Обработан ли он в лабораторной работе? Если да, то где?
6. Каким образом соединяются два микроконтроллера с помощью UART? Как соединяются контакты?
7. Из какого максимального и минимального количества бит может состоять сообщение UART?