**Урок №5. Работа с портом ввода/вывода**

**Вводный блок (10 минут)**

**Цели и задачи урока:**

1. Повторение функций порта ввода-вывода
2. Знакомство с функцией считывания данных с порта ввода-вывода
3. Создание алгоритма изменения уровня на светодиодах с помощью порта ввода-вывода

Вспомним прошлые занятия

Что такое переменная?

Как создаются условия и зачем они нужны?

Какие компоненты мы изучили на предыдущих занятиях? Как они подключаются?

**Познание нового** (**30 минут**)

Зачастую случаются ситуации, когда приходит сигнал на Arduino для выполнения определенных действий. Например, Arduino через Bluetooth подключена к мобильному телефону и оттуда управляется. Это может быть отслеживание данных о температуре или влажности в определенном месте.

Для этого используется *Serial.avaliable*

Функция получает количество байт (символов) доступных для чтения из последовательного интерфейса связи. Это те байты, которые уже поступили и записаны в буфер последовательного порта. Буфер может хранить до 64 байт.

void loop() {

if (Serial.available() > 0) {

byte data = Serial.read();

if (data == '1') {

Serial.println("YES 1");

}

}

}

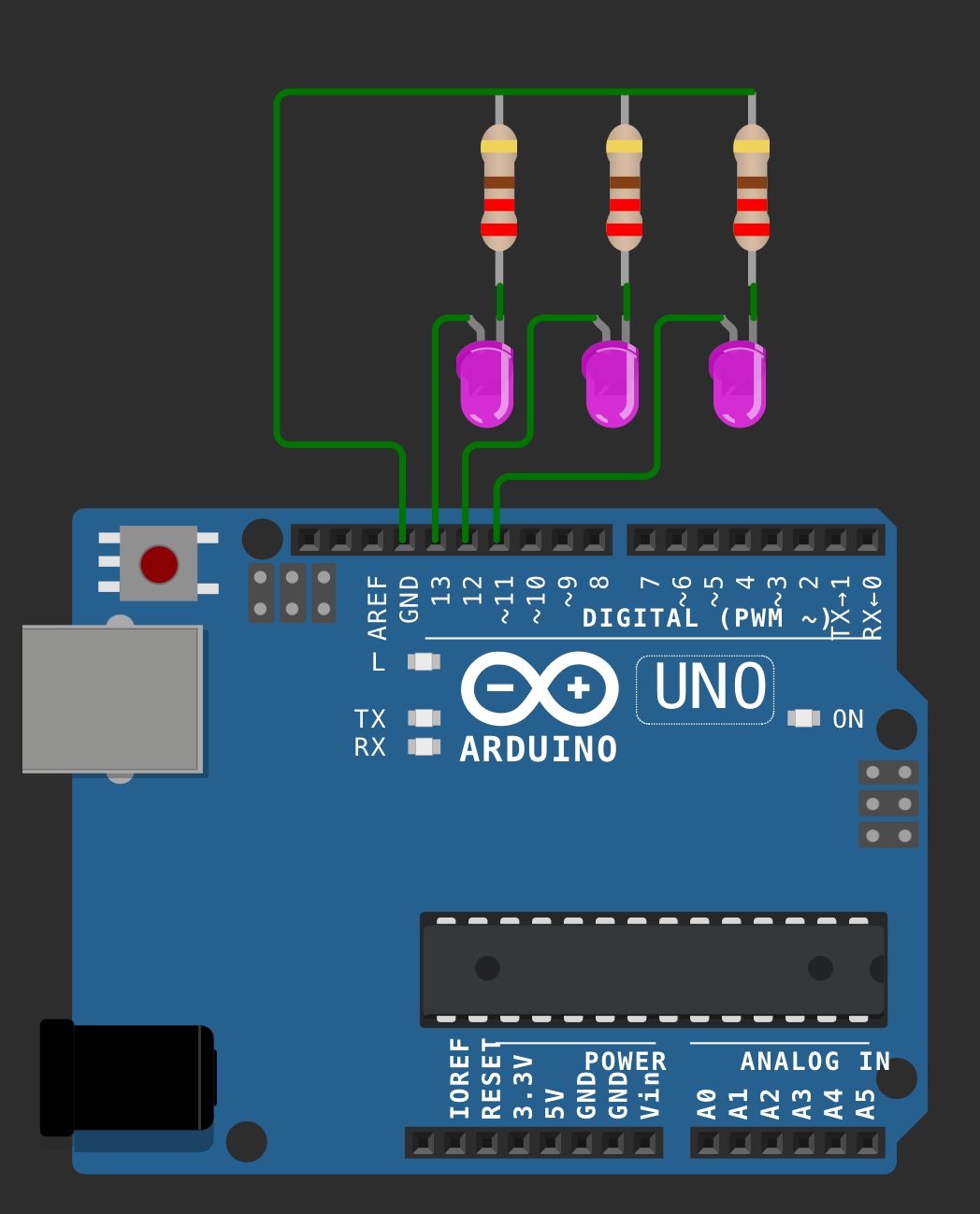
Например, С помощью функции *Serial.avaliable* мы проверяем, есть ли данные и если они есть, то начинается обработка данных.

В данном примере создается переменная типа byte, в которую считывается пришедший байт, причем считывание происходит последовательно (сначала первый символ, потом второй и так далее).

Если пришедший байт равен 1, то выводится сообщение, сигнализирующее о том, что пришел нужный байт.

Управление светодиодами с помощью порта ввода-вывода

Создадим схему, состоящую из 3 светодиодов, подключенным к портам 11 12 и 13



Подключим все контакты и настроим порт ввода-вывода на скорость 9600. Помимо этого, создадим три переменные типа bool (правда или ложь) для каждого светодиода.

bool led\_1 = LOW;

bool led\_2 = LOW;

bool led\_3 = LOW;

void setup() {

Serial.begin(9600);

pinMode(11, OUTPUT);

pinMode(12, OUTPUT);

pinMode(13, OUTPUT);

}

В функции loop ожидаем данные:

void loop() {

if (Serial.available() > 0) {

byte data = Serial.read();

}

}

Далее, запишем условия изменения уровня, тем самым включая или выключая светодиод:

void loop() {

if (Serial.available() > 0) {

byte data = Serial.read();

**if (data == '1') {**

**led\_1 = !led\_1;**

**}**

**if (data == '2') {**

**led\_2 = !led\_2;**

**}**

**if (data == '3') {**

**led\_3 = !led\_3;**

**}**

}

digitalWrite(11, led\_1);

digitalWrite(12, led\_2);

digitalWrite(13, led\_3);

}

*Пояснение к оператору «!»*

Оператор ! в данном случае выступает как логическое «Не», то есть, если в переменной хранится значение LOW (низкий уровень), то НЕ LOW это HIGH. Таким образом, и происходит изменение уровня на

[**https://wokwi.com/projects/404573119198760961**](https://wokwi.com/projects/404573119198760961)

**Некомпьютерная активность** (**20 минут**)

*Шифр Цезаря*

Шифр Цезаря - один из старейших и простейших методов шифрования, при котором каждая буква заменяется другой буквой, находящейся на определенное число позиций вперед в алфавите.

*Материалы*:

- Бумага и ручки/карандаши для записей

- Алфавит (можно использовать как латинский, так и кириллический)

1. Объясните ученикам принцип шифра Цезаря:

Каждая буква заменяется на букву, находящуюся на определенное число позиций вперед в алфавите.

Например, если сдвиг равен 3, то A -> D, B -> E, и так далее.

2. Создайте таблицу-"ключ" для шифрования, где показано, какая буква соответствует какой при разных сдвигах.

*Ход игры:*

1. Разделите учеников на пары или небольшие группы.

2. Каждая группа выбирает свой "ключ" шифрования - величину сдвига (например, 5 позиций).

3. Ученики пишут короткие сообщения, используя шифр Цезаря с выбранным ключом, и обмениваются ими с другими группами.

4. Группы-получатели пытаются расшифровать полученные сообщения, используя таблицу-ключ.

5. После расшифровки ученики сравнивают оригинальное и расшифрованное сообщения.

**Работа над проектом** (**20 минут**)

Тема: Создание нового проекта и написание программы, которая управляет светодиодами с помощью принятия данных с порта ввода-вывода

*Реализовать алгоритм, позволяющий при вводе определенного числа использовать один из представленных вариантов:*

При вводе 0 светодиоды не светятся

При вводе 1 светится только 1 светодиод

При вводе 2 светятся 2 светодиода

При вводе 3 светятся все светодиоды

Изначально никакой из светодиодов не светится

[**https://wokwi.com/projects/404573644298389505**](https://wokwi.com/projects/404573644298389505)

**Рефлексия (10 минут):**

"Что нового вы узнали сегодня?"

"Как работает функция *Serial.available*?"

"Как работает функция *Serial.read*?"