Типы данных в среде Arduino. Считывание значений. Монитор последовательного порта. Создание условия.

boolean

char

byte

int

long

short

float

digitalRead

digitalWrite

Типы данных

1. **bool, boolean** – может принимать одно из двух значений: **true** или **false** (**1 или 0**).

Пример: **boolean** running = false**;** //running несёт в себе отрицание, логическое «нет».

2. **char** - хранит символьное значение. Символы пишутся в одинарных кавычках, например: ‘A’ (совокупность символов — строки — пишутся в двойых кавычках: «ABC»).

Пример: **char** x = "ABCDEF"; //икс содержит в себе символ «A»

Типы данных

3. **byte , uint8_t –** хранит 8-битное беззнаковое число, от 0 до 255.

Пример: **byte** x = B10010; /* "B" - префикс двоичной системы счисления B10010 = 18 в десятичной системе*/

4. **int, short, int16_t –** основной тип данных для хранения целых чисел от -32768 до 32767.

Пример: **int** x = 10; //x присвоено значение 10

Типы данных

5. unsigned int, word, uint16_t – беззнаковое целое число, также как и тип int (знаковое) занимает в памяти 2 байта. Но в отличие от int, тип unsigned int может хранить только положительные целые числа в диапазоне от 0 до 65535

Типы данных

6. long , int32_t – может хранить число от -2 147 483 648 до 2 147 483 647.

Пример: **long** x = 186000; //большое число

7. unsigned long , uint32_t – может хранить число от 0 до 4294967295. //очень большое число

8. float , double – может хранить **очень большое** число с плавающей точкой (с точностью до 6-7 знаков)

Пример: **float** x = 9,81; //дробное число

Тип	Синоним	Байт	Диапазон
bool	boolean	1	false, true
byte	uint8_t	1	0...255
char		1	-128...127
unsigned char		1	0...255
int	short, int16_t	2	-32768...32767
unsigned int	word, uint16_t	2	0...65535
long	int32_t	4	-2147483648... 2147483647
unsigned long	uint32_t	4	0...4294967295
float	double	4	-3.4028235E+38… 3.4028235E+38

Функции digitalWrite() и digitalRead()

- Функция **digitalWrite()** отправляет на цифровой вывод значение на заданный вход - **HIGH** или **LOW**. Пример: digitalWrite(13, HIGH); /* подаём высокий уровень сигнала на 13-й пин Arduino*/
- Функция **digitalRead()** считывает значение с заданного входа - **HIGH** или **LOW**.

val = digitalRead(13); /* считываем значение **HIGH** или **LOW** с входа. Если на 13-м пине окажется напряжение выше 2,6 В, то val будет иметь значение **HIGH** */

Функции analogWrite() и analogRead()

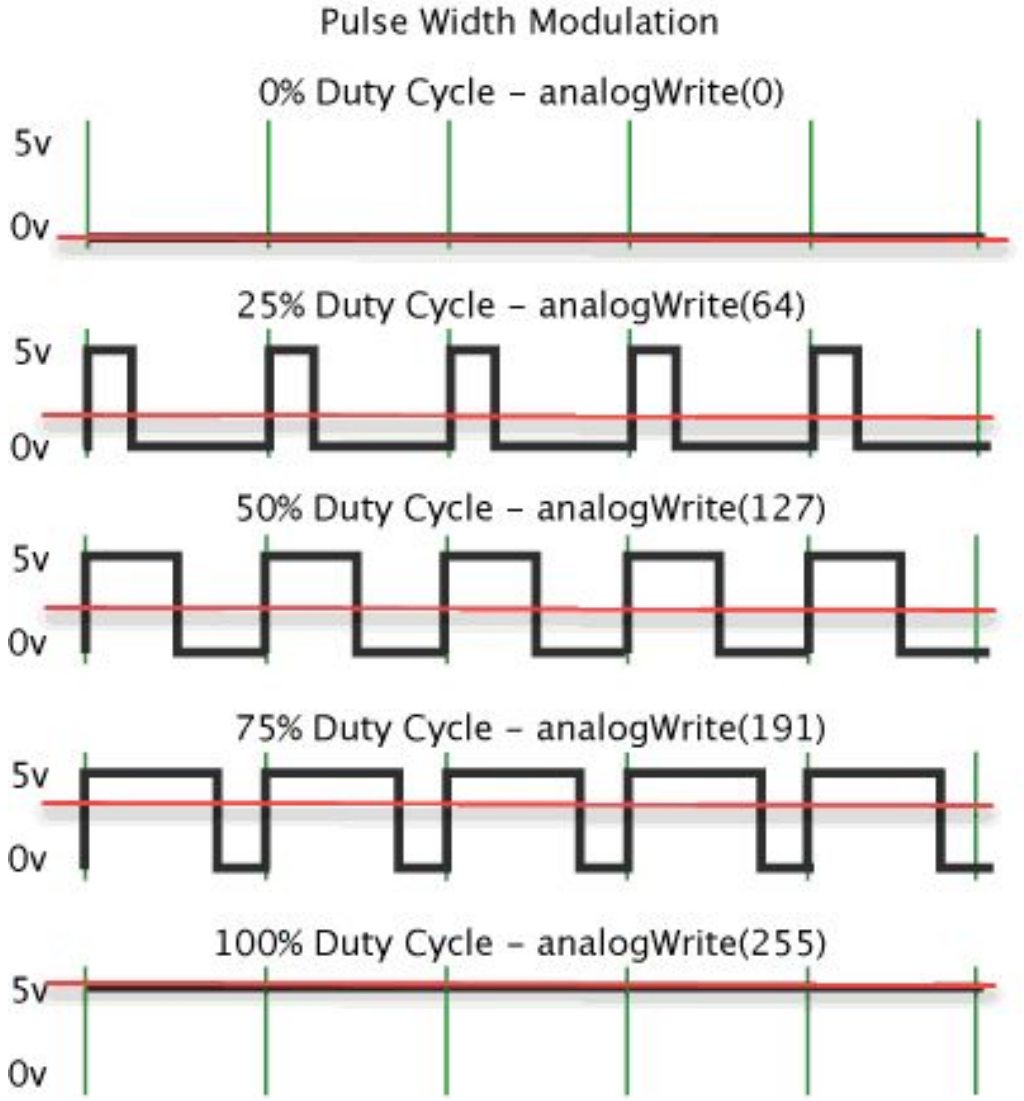
- Функция **analogWrite()** формирует заданное аналоговое напряжение от 0 до 5 вольт на выводе в виде ШИМ-сигнала. Может использоваться для варьирования яркости свечения светодиода или управления скоростью вращения двигателя.

Пример: **analogWrite(pin, value)**

pin: вывод, на котором будет формироваться напряжение.

value: коэффициент заполнения – лежит в пределах от

Пример значений функции «analogWrite» от 0 до 255



Функции analogWrite() и analogRead()

- Функция **analogRead()** считывает величину напряжения с указанного аналогового вывода. Разрешающая способность АЦП* составляет: 5 В / 1024 значения или 0.0049 В (4.9 мВ) на одно значение.

Пример: **value** = analogRead(**pin**);

pin: вывод, с которого будет считываться напряжение.

value: целое число int (от 0 до 1023)

Если аналоговый вход ни к чему не подключен, значение, возвращаемое функцией analogRead(), будет меняться под влиянием нескольких факторов (таких, как величина напряжения на других аналоговых входах, наводок от вашей руки вблизи платы и т.д.).

* АЦП (Аналого-цифровой преобразователь) — устройство, преобразующее входной аналоговый сигнал в дискретный код.

Функция **pulseIn()**

Считывает длину сигнала на заданном порту (**HIGH** или **LOW**).

Например, если задано считывание HIGH функцией pulseIn(), функция ожидает пока на заданном порту не появиться **HIGH**. Когда **HIGH** получен, включается **таймер**, который будет остановлен когда на порту вход/выхода будет **LOW**.

Функция **pulseIn()** возвращает длину сигнала в микросекундах. Функция возвращает 0, если в течение заданного времени (таймаута) не был зафиксирован сигнал на порту.

Функция pulseIn()

Синтаксис

pulseIn(pin, value)

pulseIn(pin, value, timeout)

Параметры

pin: номер порта вход/выхода, на котором будет ожидаться сигнал. (int)

value: тип ожидаемого сигнала — **HIGH** или **LOW**.

timeout (опционально): время ожидания сигнала (таймаут) в микросекундах; по умолчанию - одна секунда. (unsigned long)

Возвращаемое значение

Длина сигнала в микросекундах или 0, если сигнал не получен до истечения таймаута. (unsigned long)

Пример pulseIn()

```
int pin = 7;
unsigned long duration;

void setup()
{
  pinMode(pin, INPUT);
}

void loop()
{
  duration = pulseIn(pin, HIGH);
}
```

Serial

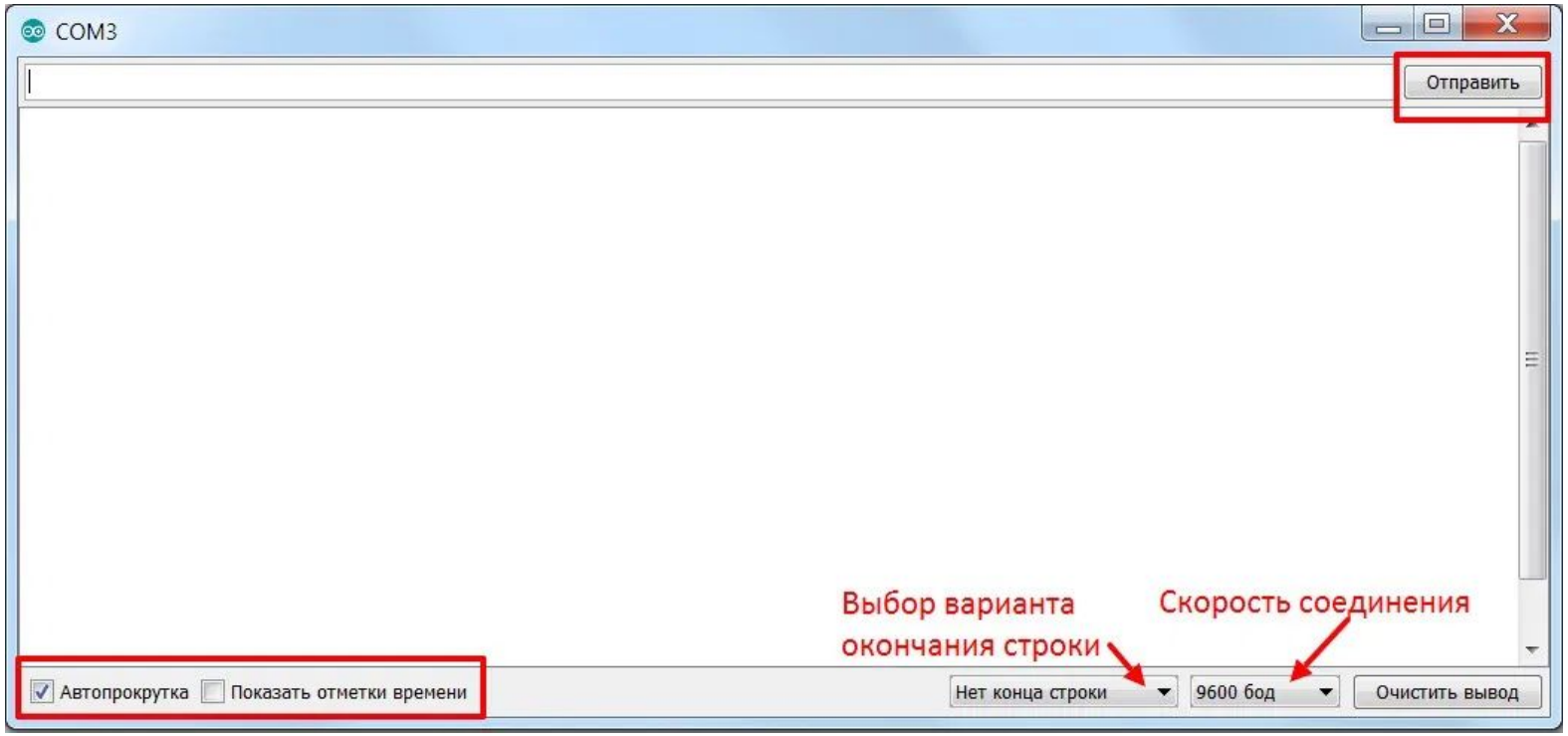
Класс Serial используется для связи платы Ардуино с компьютером или другими устройствами. Все платы Arduino имеют последовательный порт (UART).

Он связан с цифровыми выводами 0 (RX) и 1 (TX), а также используется для связи с компьютером через USB.

Во время использования последовательного порта, выводы 0 и 1 не могут использоваться в качестве цифровых входов или выходов.

Монитор порта в Arduino

Позволяет установить Arduino двунаправленую связь с компьютером через виртуальный последовательный порт.



Класс **Serial**

- **Serial.begin()** – объявление начала работы с последовательным портом указанием скорости в бодах* (baud). Прописывается в самом начале в void setup(){…}

```
void() setup{

Serial.begin(9600); /*начало работы с последовательным портом на скорости 9600 бод в секунду*/

}
```

Класс **Serial**

- **Serial.print ()** – вывод значений в монитор порта
- **Serial.println()** – вывод значений в монитор порта на новую строку

Serial.print(“Hello world”) – на экране монитора порта мы увидим надпись Hello world (без“”).

Serial.println(x) - на экране монитора порта мы увидим значение переменной «x» в том формате, в котором она представляется.

Операторы сравнения

- == равенство (a == b)
- != неравенство (a != b)
- >= больше или равно (a >= b)
- <= меньше или равно (a <= b)
- > больше (a > b)
- < меньше (a < b)

Оператор If (условие)

Для обозначения условий в Arduino используется такая конструкция:

```
if (условие)
{
  // В этом блоке список команд, выполняющихся, если условие истино или имеет значение, отличное от 0
}
else
{
  // В этом блоке список команд, выполняющихся, если условие ложно или имеет значение, равное 0
}
```

Можно обойтись и без блока **else**, если вы хотите делать что-то только при выполнении условия и не будете ничего делать, если условие не выполнилось.

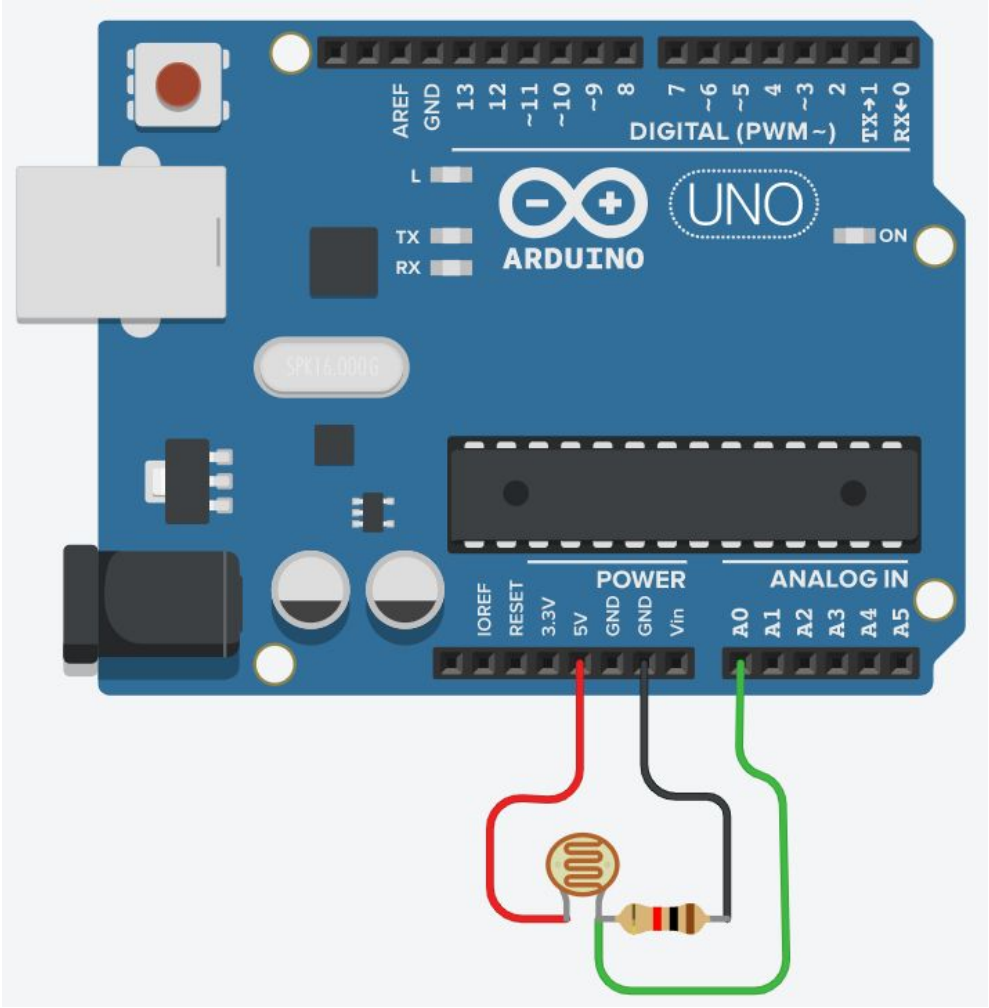
Оператор If (условие)

Конструкция **else if** может быть использована с или без заключительного else и наоборот. Допускается неограниченное число таких переходов **else if**.

```
if (x < 500)
{
  // выполнять действие A
}
else if (x >= 1000)
{
  // выполнять действие B
}
else
{
  // выполнять действие  C
}
```

Практическая работа

1) Собрать схему в Tinkercad с фоторезистором и резистором постоянного сопротивления на 1 кОм.



Практическая работа

2) Переписать от руки код без комментариев, переименновать переменную «x» на любую другую, сменить аналоговый пин «A0» на «A1».

Применить все исправления кода к схеме, затем проверить работоспособность схемы в эмуляторе: наблюдая в правой нижней части значения монитора порта, изменить влияние света на фоторезистор (нажать на него, затем поперемещать появившийся ползунок)

Код начальной программы

```
//объявляем переменную целочисленного типа (int) как "x"
int x;


void setup(){
//устанавливаем порт на приём сигнала
pinMode(A0, INPUT);
//подключаем монитор последовательного порта
Serial.begin(9600);

}

void loop(){

//считывание ЗНАЧЕНИЯ с пина и присвоение его переменной
x = analogRead(A0);
//вывод в монитор последовательного порта значения переменной "x"
Serial.print(x);
// возврат каретки
Serial.println("");
//задержка (чтобы успевать просматривать значения)
delay(500);
}
```

Контрольное задание

- Имея исходный код программы, необходимо разработать прибор, способный измерять интенсивность света. Для индикации уровня света необходимо использовать 5 светодиодов разного цвета. Разница отображаемой светодиодами интенсивности света между каждым светодиодом должна быть около 20%.

Полезные ссылки

- https://arduino-tex.ru/news/3/urok-2-peremennye--izuchaem-arduino-bez-arduino.html
- https://www.youtube.com/watch?v=0olrHvuccTY