

# Переменные

---

Важной составляющей синтаксиса языков программирования являются переменные. Переменные — это поименованная память, в которую можно вносить данные и из которой можно брать данные. До начала работы с переменными их нужно объявить. В C++ переменные можно объявлять практически в любых местах программы, но важно знать о зоне видимости переменных, — они видны лишь внутри того блока, в котором объявлены.

---

## Листинг 4.2. Объявление переменных. Зоны видимости. Сообщение об ошибке

---

```
//переменные будут видны везде в программе
int ex1;
float ex2;
void setup() // Настройка
{
    // переменная будет видна только внутри функции setup.
    int ex3;
    ex3=8;
    //Устанавливаем скорость порта связи Arduino - ПК.
    Serial.begin(9600);
    // значение ex1 изменится на 8*2.
    ex1=ex3*2;
}
// Основная программа
void loop()
{
    // Следующая строка содержит ошибку,
    //переменная ex3 не может быть видна за пределами функции setup.
    ex3=ex1+5;

    delay(ex1*100);           // Ждет 8*2*100 миллисекунд.
    // передает на компьютер время работы в миллисекундах.
    Serial.println(millis());
    delay(1000);             // Ждет 1 сек.
}
```

Листинг 4.2 в демонстрационных целях содержит специально добавленную в него ошибку объявления переменной, и при попытке его скомпилировать об этой ошибке будет выведено сообщение: **'ex3' was not declared in this scope.**

Переменные могут отличаться по типу данных, для хранения которых созданы (табл. 4.1).

Таблица 4.1. Типы данных

Обозначение в программе	Название	Принимаемые значения	Размер в памяти
boolean	Логический	True/false, 1/0	1 байт
char	Символьный	-128/+127	1 байт
byte	Короткое беззнаковое целое	0-255	1 байт
int	Целое число	-32768/32767	2 байта
long	Длинное целое число	-2147483648/2147483647	4 байта
float	Число с плавающей точкой	$-3,4028235 \cdot 10^{38} / 3,4028235 \cdot 10^{38}$	4 байта

При работе с переменными следует понимать, что они не являются идеальным хранилищем информации — так, например, целочисленные переменные могут переполняться. Это происходит в тех случаях, когда значение, которое следует записать в переменную, больше максимально возможного для этого типа данных. Переменные с плавающей точкой подвержены другой проблеме — они округляют свои значения при сложении большого числа с малым. Например, сложение  $1,1 \cdot 10^{25}$  и 10 даст  $1,1 \cdot 10^{25}$ , а число 10 просто потеряется. Также Arduino IDE не всегда корректно работает с преобразованием типов данных.

Рассмотрим небольшую программу, осуществляющую получение данных от компьютера через порт ввода/вывода (листинг 4.3).

---

#### Листинг 4.3. Получение данных от компьютера через порт ввода/вывода

---

```
void setup() // Настройка
{
  //Устанавливаем скорость порта связи Arduino - ПК.
  Serial.begin(9600);
}

// Основная программа
void loop()
{
  char char1;
  // если поступили данные.
  if (Serial.available() > 0)
  {
    // считываем символ.
    char1 = Serial.read();
    // отсылаем то, что получили, обратно на ПК.
    Serial.println(char1);
  }
}
```

Скомпилируйте и загрузите эту программу. Введите в верхней части окна **Монитор порта** слово ПРИВЕТ! и нажмите кнопку **Отправить**. Программа отправит это слово обратно на ПК посимвольно. Результат будет виден в нижней части окна:

Arduino Leonardo



sketch\_jan3a.ino



```
1
2 // Настройка
3 void setup(){
4
5     //Устанавливаем скорость порта связи Arduino - ПК
6     Serial.begin(9600);
7 }
8
9 // Основная программа
10 void loop(){
11     char ch;
12     // Если поступили данные
13     if (Serial.available()>0) {
14         //Считываем символ
15         ch = Serial.read();
16         // Отсылаем обратно на ПК
17         Serial.println(ch);
18     }
19 }
```

Вывод Монитор порта ×



Message (Enter to send message to 'Arduino Leonardo' on 'COM4')

Новая строка ▾

9600 baud ▾

```
j
g
h
f
j
d
g
l
```

Код программы:

```
// Настройка
void setup(){

  //Устанавливаем скорость порта связи Arduino - ПК
  Serial.begin(9600);
}

// Основная программа
void loop(){
  char ch;
  // Если поступили данные
  if (Serial.available()>0) {
    //Считываем символ
    ch = Serial.read();
    // Отсылаем обратно на ПК
    Serial.println(ch);
  }
}
```