

Справочник языка Ардуино

Язык можно разделить на четыре раздела: операторы, данные, функции и библиотеки.

Язык Arduino	Пример	Описание
Операторы		
setup()	<pre>void setup() { pinMode(3, INPUT); }</pre>	Функция используется для инициализации переменных, определения режимов работы выводов и т.д. Функция запускается только один раз, после каждой подачи питания на микроконтроллер.
loop()	<pre>void loop() { digitalWrite(3, HIGH); delay(1000); digitalWrite(3, LOW); delay(1000); }</pre>	Функция loop крутится в цикле, позволяя программе совершать вычисления и реагировать на них. Функции setup() и loop() должны присутствовать в каждом скетче, даже если эти функции не используются.
Управляющие операторы		
if	<pre>... if (x > 100) digitalWrite(3, HIGH); if (x < 100) digitalWrite(3, LOW); ...</pre>	Оператор if используется в сочетании с операторами сравнения (==, !=, <, >) и проверяет, достигнута ли истинность условия. Например, если значение переменной x больше 100, то включается светодиод на выходе 13, если меньше — выключается.
if..else	<pre>... if (x > 100) digitalWrite(3, HIGH); else digitalWrite(3, LOW); ...</pre>	Оператор else позволяет сделать проверку отличную от указанной в if, чтобы осуществлять несколько взаимоисключающих проверок. Если ни одна из проверок не получила результат ИСТИНА, то выполняется блок операторов в else.
switch...case	<pre>... switch (x) { case 1: digitalWrite(3, HIGH); case 2: digitalWrite(3, LOW); case 3: break; default: digitalWrite(4, HIGH); } ...</pre>	Подобно if, оператор switch управляет программой, позволяя задавать действия, которые будут выполняться при разных условиях. Break является командой выхода из оператора, default выполняется, если не выбрана ни одна альтернатива.

for	<pre>void setup() { pinMode(3, OUTPUT); } void loop() { for (int i=0; i <= 255; i++){ analogWrite(3, i); delay(10); } }</pre>	<p>Конструкция for используется для повторения операторов, заключенных в фигурные скобки. Например, плавное затемнение светодиода. Заголовок цикла for состоит из трех частей: for (initialization; condition; increment) — initialization выполняется один раз, далее проверяется условие condition, если условие верно, то выполняется приращение increment и цикл повторяется.</p>
while	<pre>void loop() { while (x < 10) { x = x + 1; Serial.println(x); delay(200); } }</pre>	<p>Оператор while используется, как цикл, который будет выполняться, пока условие в круглых скобках является истиной. В примере оператор цикла while будет повторять код в скобках бесконечно до тех пор, пока x будет меньше 10.</p>
do...while	<pre>void loop() { do { x = x + 1; delay(100); Serial.println(x); } while (x < 10); delay(900); }</pre>	<p>Оператор цикла do...while работает так же, как и цикл while. Однако, при истинности выражения в круглых скобках происходит продолжение работы цикла, а не выход из цикла. В приведенном примере, при x больше 10 операция сложения будет продолжаться, но с большей паузой (1000 мс).</p>
break continue	<pre>switch (x) { case 1: digitalWrite(3, HIGH); case 2: digitalWrite(3, LOW); case 3: break; case 4: continue; default: digitalWrite(4, HIGH); }</pre>	<p>Break используется для принудительного выхода из циклов switch, do, for и while, не дожидаясь завершения цикла. Оператор continue пропускает оставшиеся операторы в текущем шаге цикла.</p>
Синтаксис		
; (точка с запятой)	<pre>... digitalWrite(3, HIGH); ...</pre>	<p>Точка с запятой используется для обозначения конца оператора. Забытая в конце строки точка с запятой приводит к ошибке.</p>

{ (фигурные скобки)	<pre>void setup() { pinMode(3, INPUT); }</pre>	Открывающая скобка “{” должна сопровождаться закрывающей скобкой “}”. Непарные скобки могут приводить к скрытым и непонятным ошибкам при компиляции скетча.
// (комментарий)	<pre>x = 5; // комментарий</pre>	Комментарии используются для напоминания, как работает программа. Они игнорируются компилятором и не экспортируются в процессор, не занимая место в памяти микроконтроллера Ардуино.
#define	<pre>#define ledPin 3</pre>	Директива #define позволяет дать имя константе. Директива служит исключительно для удобства и улучшения читаемости программы.
#include	<pre>// библиотека для серво #include <Servo.h></pre>	Директива #include используется для включения сторонних библиотек в скетч. Обратите внимание, что директивы #include и #define, не требует точки запятой в конце строки.
Константы		

Программирование Arduino намного проще и понятнее, чем язык C++.

Например.

На языке программирования Arduino включить в скетче последовательный порт на скорости 9600 бит в секунду можно всего лишь одной строчкой:

```
Serial.begin(9600);
```

При использовании C/C++ нам бы пришлось долго разбираться с документацией на микроконтроллер и написать в скетче нечто подобное:

```
UBRR0H = ((F_CPU / 16 + 9600 / 2) / 9600 - 1) >> 8;
UBRR0L = ((F_CPU / 16 + 9600 / 2) / 9600 - 1);
sbi(UCSR0B, RXEN0);
sbi(UCSR0B, TXEN0);
sbi(UCSR0B, RXCIE0);
```