Ардуино

Массивы, строки, циклы

И ФУНКЦИИ

Операторы

+ - * /	арифметика
%	остаток от деления
x++ ++x	прибавление 1 (1 - после выполнения текущего блока кода (statement или {внутри})); мгновенное прибавление 1
= += -= /= *=	присвоить; прибавить число и присвоить получившееся; вычесть число и присвоить; разделить на число и присвоить; умножить и присвоить
==	оператор сравнения

С подключённой библиотекой math.h	
pow(x, power)	возведение числа в степень
cos(x) sin(x) tan(x)	тригонометрия (в радианах!)
sqrt(x)	корень
exp(x) log(x)	e^x; ln(x)
floor(x) ceil(x)	округление вниз; вверх
fabs(x)	модуль (приводит к формату float)
Полный перечень функций и констант:	http://www.nongnu.org/avr-libc/user-manual/group__avr__math.html

Указатели

Объявление указателя:

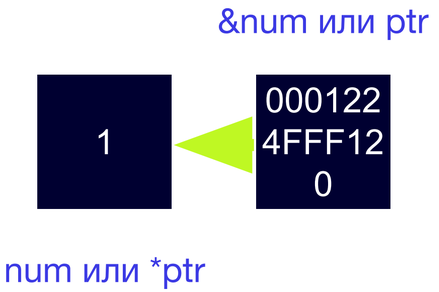
variable_type *variable_name;

Пример:

int num, *ptr;

ptr = #

*ptr = 1;



Массивы

ar и ar[0]

Объявление:

variable_type variable_name[length];

Пример:

int ar[10];

int ar[10] = {1,3,7,3};

Строки

str и str[0]

Объявление:

char variable_name[length];

#include<String.h>…

String str() = char_array;

Пример:

char str[10] = "HELLO", strc[] = {'H','E','L','L','O','\0'};

String str1() = "HELLO";

Спецсимволы

Символ	Описание
\r	возврат каретки в начало строки
\n	новая строка
\t	горизонтальная табуляция
\v	вертикальная табуляция
\»	двойные кавычки
\’	апостроф
****	обратный слеш
\0	нулевой символ

Циклы

```
for(init variable; condition;
increment/decrement operator) {
   statement
}
```

```
while (condition){
   statement
}
```

```
do {
   statement
} while (condition);
```

Примеры:

```
for(int i = 0; i <= 5; i++) {
   Serial.println(i);
}
```

```
int i = 5;
while (i--){
   Serial.println(i);
}
```

```
do {
   break;
} while (1);
```

Функции

Инициализация функций:

```
type_of_returning_value function_name(type1 param1, type2 param2, …) {
    statement
    return result;
}
```

Объявление функций (прототип):

```
type_of_returning_value function_name(type1, type2, …);
```

Примеры:

Инициализация функций:

```
int sum(int a, int b) {
    return a+b;
}
```

Объявление функций (прототип):

```
int sum(int, int);
```

Функции

- Прототип функции нужен если вы хотите создать функцию (сделать её видимой для программы), но инициализировать позже. Это атавизм языка C (там пользоваться любым объектом можно было только после его определения), в Arduino можно обойтись без прототипов.
- Если функция ничего не возвращает, её тип void.
- Чтобы менять переменные внутри функции (более одной переменной или объекты), нужно передавать в качестве параметров функции указатели на объекты (напр. не переменную b, а её указатель &b).
- При передачи одномерных массивов можно не указывать размер в скобках. При передаче многомерных массивов, размерности n нужно указывать любые n-1 размеров (напр. для двумерного 1 размер: int b[][5]).