

Overview

PHPoC is a programming language that Sollae Systems has developed. All of our PHPoC products have PHPoC interpreter in firmware. PHPoC is based on a wide use script language PHP. As PHPoC is simple, whoever has seen the programming language once can use it easily.

There are a few limitations due to embedded systems' characteristic even if PHPoC is basically compatible with PHP. We recommend you to read this page before using the product.

Basic Grammar

Comment

Not support for #, single line comment.

Types

Booleans

String "0" will be not FALSE but TRUE when being converted to Boolean.

Integers

the maximum value of an integer is about 9.2×10^{18} in PHPoC.

Classes

Not support for classes.

Arrays

Features are as follows:

- Initial value should be assigned in their declaration.
- Only number is acceptable to KEY value.
- Not support for array generation or functional call with a square bracket.
- Examples

```
<?php
// Example 1
$var0 = array(3, 7, "array", "test");
$int0 = $var0[0] + $var0[1];
$str0 = $var0[2] . " " . $var0[3];
?>
```

```
<?php
// Example 2
$int0 = 3;
$int1 = 7;
$str0 = "array";
$str1 = "test";
$var1 = array($int0, $int1, $str0, $str1);
?>
```

```
<?php
// Example 3
$str0 = "array test";
$str0[6] = "T";
$str0[7] = "E";
$str0[8] = "S";
```

```
$str0[9] = "T";
echo $str0; // Output : array TEST
?>
```

Null

Not support for Null.

Type Juggling

- Arithmetic Operator: addition(+), subtraction(-), multiplication(*), division(/)

Types	Boolean	Integer	Floating Point	String
Boolean	X	X	X	X
Integer	X	O	O	X
Floating Point	X	O	O	X
String	X	X	X	X

- Arithmetic Operator: the rest(%)

Types	Boolean	Integer	Floating Point	String
Boolean	X	X	X	X
Integer	X	O	X	X
Floating Point	X	X	X	X
String	X	X	X	X

- Bitwise Operator: AND(&), OR(|), XOR(^), left shift(<<), right shift(>>)

Types	Boolean	Integer	Floating Point	String
Boolean	X	X	X	X
Integer	X	O	X	X
Floating Point	X	X	X	X
String	X	X	X	X

- Bitwise Operator: compliment(~)

Boolean	Integer	Floating Point	String
X	O	X	X

- Comparison Operator: less than(<), greater than(>), less or equal(<=), greater or equal(>=)

Types	Boolean	Integer	Floating Point	String
Boolean	X	X	X	X
Integer	X	O	O	X
Floating Point	X	O	O	X
String	X	X	X	O

- Comparison Operator: equal(==), not equal(!=), not equal(<>)

Types	Boolean	Integer	Floating Point	String
Boolean	O	X	X	X
Integer	X	O	X	X
Floating Point	X	X	O	X
String	X	X	X	O

- Increment/Decrement Operator: increment(++), decrement(--)

Boolean	Integer	Floating Point	String
X	O	X	X

- Logical Operator: AND(&&), OR(||)

Types	Boolean	Integer	Floating Point	String
Boolean	O	O	X	O
Integer	O	O	X	O
Floating Point	X	X	O	X
String	O	O	X	O

- Logical Operator: NOT(!)

Boolean	Integer	Floating Point	String
O	O	X	O

- Sign Operator: positive(+), negative(-)

Boolean	Integer	Floating Point	String
X	O	O	X

- Expression in Control Structure: if, for, (do) while

Boolean	Integer	Floating Point	String
O	O	X	O

- printf function output format

Types	Boolean	Integer	Floating Point	String
%b, %o, %x	X	O	X	X
%d, %u	X	O	X	X
%c	X	O	X	X
%e, %f, %g	X	X	O	X
%s	X	X	X	O

Variables

Variables Definition

Initial value should be assigned when you declare variables. You cannot declare more than 2 variables in single line.

Max size of variable's name is 32 bytes. If it exceeds the max length, the rest are ignored.

Variable Variables

Not support for variable variables.

Constants

Constants Definition

Not support for declaring constants with 'const' keyword.

Predefined Constants

PHPoC provides several predefined constants. Refer to the appendix for more details.

Execution Operators

Not support for execution operators(` `).

Logical Operators

Logical operators only accept mark ('!', '&&', '||' and '^') but not accepts letter (NOT, AND, OR, and XOR).

Array Operators

Not support for array operators (Union, Equality, Identity and etc.).

Control Structures

Control Structures

Not support for control structure as follows.

- foreach, declare, require, require_once, goto and label

Alternative Syntax for Control Structures

Not support for any alternative syntax for control structure (':', endif, endwhile and etc.).

Switch

It is not acceptable to use ';' (semicolon) instead of ':' (colon) after case expression of switch statement.

Functions

Function Definition

In PHPoC, function has to be declared before being called. Meanwhile, PHP can call function before declaring.

Function Arguments

It is not allowed to use the duplicated argument names in a user-defined function.

Returning Values

If there is no returning value, user-defined function returns zero (0) instead of NULL. Not support for returning by reference.

Variable Functions

Not support for variable functions.

Anonymous functions

Not support for anonymous functions

Internal Functions

PHPoC supports for various internal functions. Refer to a document named [PHPoC Internal Functions](#) to know how to use those functions.

Namespaces

User-defined Namespaces

Not support for User-defined Namespaces.

Function Arguments

Keywords, functions and constants share namespaces. Be careful not to make the same name of those when you create them.

Predefined Constants

PHPoC provides predefined constants as following below.

Name	Description
COUNT_NORMAL	Normal Counting for one-dimension array
COUNT_RECURSIVE	Recursive Counting for multi-dimension array
EPIPE	Broken Pipe, Returning as connection is broken while sending TCP data
EBUSY	Device or Resource Busy
ENOENT	No file entry
FALSE	False
M_PI	Pi (≈ 3.1415926535898)
M_E	Euler's Constant (≈ 2.718281828459)
MAX_STRING_LEN	Maximum length of a string
O_NODIE	avoid to terminate script execution: file open
PHP_VERSION_ID	PHPoC Version
SEEK_CUR	File pointer position: current position of file
SEEK_END	File pointer position: at the end of file
SEEK_SET	File pointer position: at the beginning of file
SSL_CLOSED	SSL status: not connected
SSL_CONNECTED	SSL status: connected
SSL_LISTEN	SSL status: wait for connection
TRUE	True
TCP_CLOSED	TCP status: not connected
TCP_CONNECTED	TCP status: connected
TCP_LISTEN	TCP status: wait for connection

Keyword

This table below shows predefined keywords in PHPoC. As keywords, functions and constants share namespace, try not to use duplicated names with those keywords when creating functions or constants.

Alphabet	Keywords
a	array
b	bool boolean break
c	case const continue
d	default define die do
e	echo else elseif exit
f	float for
g	global goto
i	if int integer include include_once
p	print
r	return
s	static string switch
w	while

Restriction about Memory

Number of Variables

PHPoC is limited memory size for using variables. Thus, both the number of variables and the length of string variables are limited. The max number for variable creation is in inverse proportion to the size of variables.

Error Messages

PHPoC prints various error messages out for debugging by console.

Error Messages
address already in use
argument count mismatch
cannot break/continue N level(s)
'case' or 'default' expected
device or resource busy
divided by zero
duplicated name
expression syntax error
file name too long
file not found
function not implemented
integer number too large
invalid argument
invalid constant name
invalid device or address
maximum execution time exceeded
missing operator
missing terminating character ''' or ''''
modifiable value required
only variable can be passed by reference
operation not permitted
out of memory
string too long
syntax error
syntax error, unexpected array [, expecting 'token']
syntax error, unexpected character
syntax error, unexpected 'character' [, expecting 'character']
syntax error, unexpected end of file
syntax error, unexpected 'name' [, expecting 'character']
syntax error, unexpected number [, expecting 'character']
syntax error, unexpected 'operator' [, expecting 'token']
syntax error, unexpected string [, expecting 'character']
syntax error, unexpected 'token' [, expecting 'token']
syntax error, unexpected variable [, expecting 'character']
too many open files
undefined name
undefined offset
unsupported argument type
unsupported operand type
unsupported operator
unsupported pid
unsupported type juggling
'while' expected