

[Home](#)[articles](#)[Sitemap](#)[Facebook](#)[Github](#)[me contacter](#)  

FORTH on ARDUINO

article: 11 mai 2019 / mis à jour 07 nov. 2019

- [Advantages and disadvantages of the C language](#)
- [Qu'est-ce que FORTH?](#)
 - [FORTH is the simplest man-machine interface](#)
 - [The FORTH compiler](#)
- [FORTH for Arduino](#)
 - [Running FORTH for Arduino](#)

ARDUINO is also a development environment that favors the C language. But this environment requires constant compilations and uploads to the ARDUINO card in development. The development of complex applications can quickly become a real calvary that can discourage the most valiant.

Advantages and disadvantages of the C language

With Arduino, we must use a minimal code when creating a program in C language. code allows to divide the program that we will create in two big parts:

```
void setup() //fonction d'initialisation de la carte
{
    //contenu de l'initialisation
}
void loop() //fonction principale qui se répète (s'exécute) à l'infini
{
    //contenu du programme
}
```

A program for ARDUINO written in C language must be nested at loop() to test and exploit it.

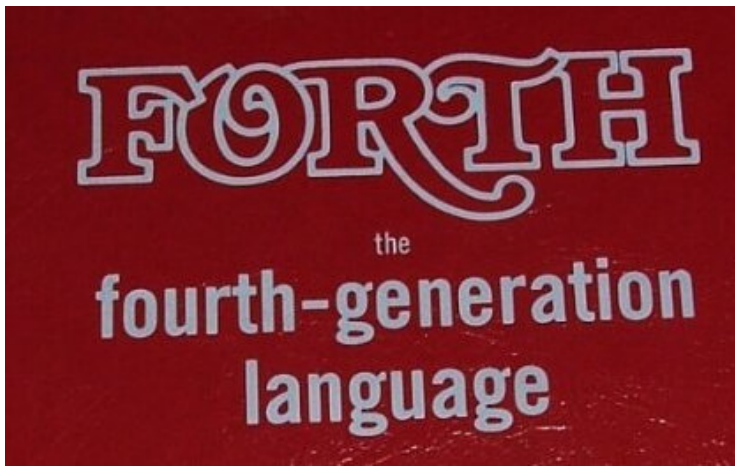
Among the advantages of the C language: it has an impressive bibliography, it is used by many developers, the application libraries are numerous ...

Among the disadvantages of the C language: it remains verbose, there is no mastery of the compiled code, it is difficult to develop subprogrammes independently from the general program ...

For us, the main disadvantage of programs written in C language for ARDUINO is the difficulty to diagnose the bugs and dysfunctions of the programs. This involves developing layer by layer and testing each function individually before nesting them in the general program. If a function in C language modifies a variable, the only way to know the value taken by this variable is to ask the function to display its contents.

This phase of design and development requires frequent compilations and uploads of the source code written in C language.

The ideal would be to have a real alternative programming language, preferably implanted directly on the ARDUINO board. There is one: **FORTH**. A very compact language, powerful, fast.



What is the FORTH language?

FORTH can be installed on an ARDUINO board, both for development and operation for all your applications, without loss of performance.

A program written in FORTH runs almost as fast as its language counterpart pure machine. And compiled FORTH programs are more compact than those written in C language.

- FORTH is an operating system
- FORTH is a programming language
- FORTH is the simplest man-machine interface

You type a list of FORTH commands. FORTH executes these commands in sequence and wait for the following list of commands. Example:

```
123 dup * . \ affiche 15129 ok
```

New orders are compiled to extend existing order lists.

```
. <nom> <liste de commandes> ;
```

Example:

```
. carre dup * ;
123 carre . \ affiche 15129
```

FORTH is the simplest man-machine interface

The most advanced controls can be tested and debugged interactively, thanks to the interpreter integrated into the FORTH language.

Bottom programming makes it possible to create and test reliable applications. All defined or pre-defined words can be tested via the interpreter except for a few words that can only be used in compilation (IF..ELSE..THEN for example)

Memory and I/O devices can be examined and programmed interactively, thanks to the interpreter. Access to any register is totally transparent: PORTA c @ retrieves via the interpreter the contents of the PORTA register of PORT A.

The unique features of Forth:

- a simple English syntax, easy to adapt to another language
- very compact and **better suited to microcontrollers**
- very friendly to interact with microcontrollers
- easy programming language and operating system for personal apps

The advantages of FORTH:

- one-pass code analysis

- text interpreter
- FORTH compiler with no equivalent
- double stack architecture

The text interpreter analyzes a list of commands in textual form. Examples:

```
13 DUP * _ .
WORDS
HEX
0 100 DUMP
```

Some FORTH commands:

```
+ - * / mod
and or xor not
key emit type dump
@ ! c@ c! cmove fill erase
dup drop swap over rot
If-else-then, begin-while-repeat,
for-next
```

The FORTH compiler

The FORTH compiler is the text interpreter running in compile mode.

The FORTH compiler creates new commands. These combine lists of existing orders.

Examples:

```
: CARRE DUP * ;
: SALUT CR ." BONJOUR LES AMIS!" ;
: FIBONACCI ( n --)
  0 1 ROT
  FOR SWAP OVER + NEXT
  DROP ;
```

FORTH for Arduino

Some versions of the FORTH language for Arduino:

- AmForth by Matthias Trute,
GNU Public License v2,
<http://amforth.sourceforge.net/>
- Flash Forth
<http://flashforth.com/>

Running FORTH for Arduino

example works with AmForth

Ignition and extinction LED:

```
HEX
20 24 C! \ positionne Pin 13 en sortie
20 23 C! \ bascule Pin 13
```

example works with Flash FORTH

Most examples are compatible between the **AmForth** version and **Flash Forth** , the latter being case-sensitive, it will sometimes be necessary to adapt the available source codes on the net.

Ignition and extinction LED connected to pin 13:

```
hex
20 24 C! \ positionne Pin 13 en sortie
```

```
20 23 c! \ bascule Pin 13
```

This example is executable directly in interfered mode from a terminal connected to the ARDUINO board, **without compilation or uploading** as it would be the case in C!

example works with AmForth

Ignition and extinction beeper:

```
40 2A c! \ positionne Pin 6 en sortie
42 44 c! \ positionne Timer0 en mode CTC
FF 47 c! \ comptage maximum pour Timer0
3 45 c! \ prescaler=3, démarrage beeper
0 45 c! \ prescaler=0, arrêt beeper
```

The example can be entered line by line from the terminal or by copying / pasting between this page and the terminal connected to the ARDUINO board (voir: [Communiquer avec Flash Forth](#))

FORTH is a super compact programming language. It offers an interesting alternative to the C language. FORTH has an interpreter, a compiler, an assembler. All this in a few tens of kilobytes!