

# ODROID

Year Two  
Issue #19  
Jul 2015

## *Popcorn* **Time**

**Magazine**

Watch  
movies  
and TV  
shows  
instantly  
with your  
ODROID



**Diet Pi**

ODROID's lightest  
distribution ever

- ODROID-C1 Music Stand
- GPIO Pins C1 Control

# INDEX



**AMSTRAD - 6**



**ANDROID GAMING: NUBS' ADVENTURE- 9**



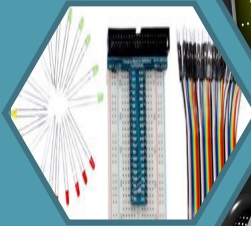
**LINUX GAMING: MILLENIUM - 10**



**ANDROID GAMING: KUNG FURY - 15**



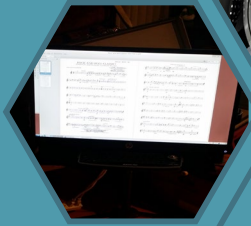
**ANDROID DEVELOPMENT - 16**



**JAVA GPIO - 19**



**WHITE NOISE GENERATOR - 20**



**ODROID MUSIC - 21**



**DIET PI - 25**



**POPCORN TIME - 30**



**MEET AN ODROIDIAN - 32**

# MAKING VIDEOGAMES FOR AMSTRAD CPC

HAVE FUN WITH THIS BLAST FROM THE PAST

by Jose Cerrejon



The AMSTRAD CPC is one of those beloved 8-bit relics we will always have room on our hearts

For those who do not know, the AMSTRAD CPC was an 8-bit computer that was popular between 1984 and 1990. CPCtelera is an engine that has been released recently in its first stable version, which facilitates the creation of games for the Amstrad computer using C or assembly code.

## Introducing CPCtelera

CPCtelera is an integrated development framework for creating Amstrad CPC games and content which includes:

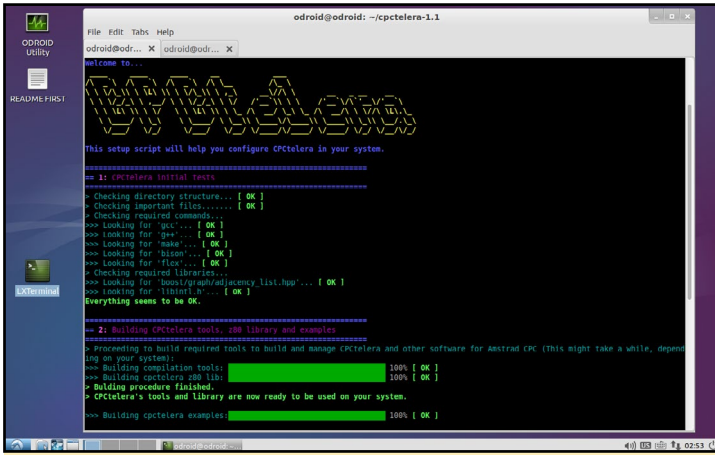
- A low-level library with support for: graphics, audio, keyboard, firmware, strings, video hardware manipulation and memory management.
- An API for developing games and software in C and Assembler
- Tools for content authoring (audio, graphics and level editing)
- Multi-platform: It works on Windows, Ubuntu, Debian, Arch, and Manjaro operating systems

The primary developer is Francisco Gallego (@frangallegobr), which is an Informatics engineer, video game developer and professor at the University of Alicante in Spain. For more features of this framework, please refer to the links at the end of this article.

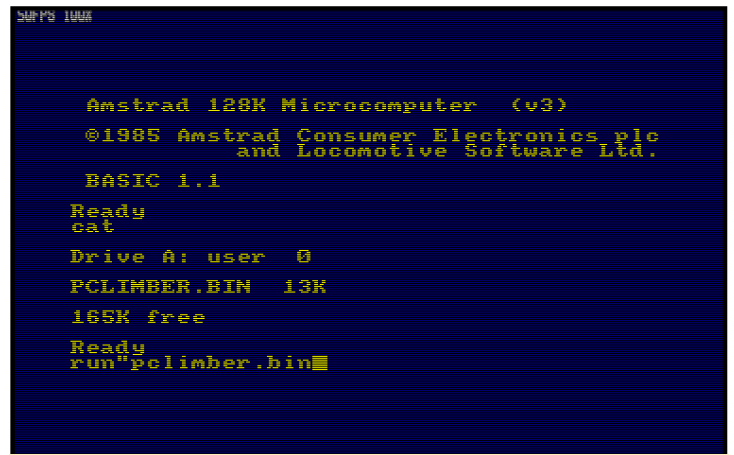
## Installing CPCtelera

First, you need to download the source code in order to compile it on your board. To do this, download the stable version like I did, or if you are brave, you can use the latest version from GitHub:

```
$ wget http://bit.ly/1MMdUMA && \
  unzip -nq $(basename $_) && \
  rm $(basename $_) && \
  cd cpctelera-1.1/
# or use the last commit:
$ git clone http://bit.ly/1IPxMOF && \
  cd $(basename $_)
```



Running ./setup.sh after resolving dependencies



Amstrad BASIC, time to dust out a bunch of essential books!

Next, we need to install any missing dependencies, some of which may be already installed:

```
$ sudo apt-get install -y build-essential libboost-dev flex bison
```

Then, invoke the installation script by running the setup file:

```
$ sudo ./setup.sh.
```

It will warn you about the necessary packages and prepare your system to run the engine. On an ODRROID-C1, it will take about 20 minutes to compile.

## Starting the Engine

If we browse through the directories, we can see some interesting folders such as docs/, which contains the reference manual, or tools/, which is used to make sprites, compose soundtracks, format converters, and more. Some of these tools are only available for Windows.

To create a new project from Bash, type the following:

```
$ cpct_mkproject [folder_project]
```

Navigate into the directory and you will see two subdirectories:

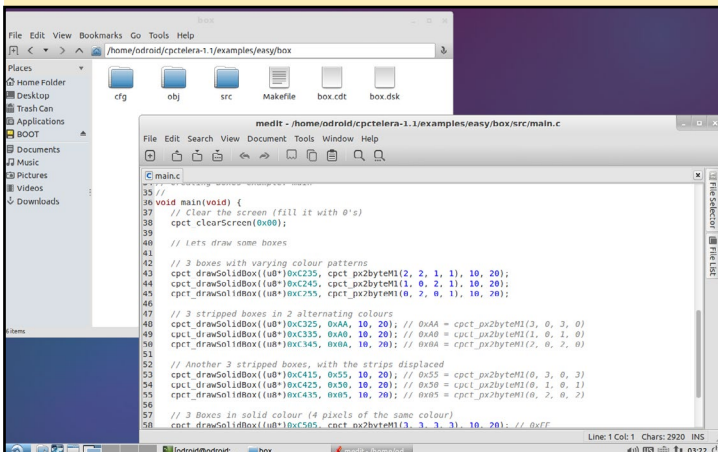
- **src/** with the source code (the first time we just have a main.c). You can create files and directories of your game here.
- **cfg/** contain settings to compile the game through the build\_config.mk file.

There are also examples that you can study. Just navigate to the directory, run make, and automatically create .CDT and .DSK files. So cool, isn't it?!

## Platform Climber

There is a complete game example included with CPCteler called Platform Climber. First, you'll need to get an AMSTRAD CPC emulator. I did not find any that were pre-compiled for ODRROID, so I had to download and compile it by myself. Don't worry, It's easy:

### You have a lot of examples to learn



### Scanlines alongside the best nostalgia, get ready to climb!





```
$ wget http://bit.ly/1U2RrjZ
$ unzip caprice*
$ make -f makefile.unix
RELEASE=TRUE
```

In a minute, you will get a binary called cap32. The use is very simple by obtaining any .BAS or .DSK file. To load the game Platform Climber, run the emulator followed with the path of the .DSK file:

```
$ ./cap32 ../cpctelera-1.1/
examples/games/platformClimber/
pclimber.dsk
```

Now you are inside the AMSTRAD! Do you feel the magic? Type cat to see the files inside the previously mounted disk, then run the game:

```
cat
run"game_name.extension
```

Notice that there is no end quote. For example, to run Platform Climber, type the following:

```
run"pclimber.bin
```

## Conclusion

Now it's your turn. I recommend you to start studying the examples in `examples/easy/src`, then modify and compile them. Below there is a link to the reference manual with all the info you need to know about functions and methods available. If you know something about SDL, everything will be a little easier.

Happy coding!

## More References:

<http://bit.ly/1IPxMOF>  
<http://bit.ly/1IpjQvV>  
<http://bit.ly/1FMAkrQ>  
<http://bit.ly/1IILeAT>

# LINUX KERNEL BUILD SCRIPTS

## TWEAK YOUR SYSTEM

by Rob Roy

To make Linux kernel compilation easier, I wrote a set of BASH scripts that can be used to download the latest version of any kernel stored in the Hardkernel GitHub branch, compile the source code, and package the completed kernel into a redistributable package. The scripts keep the original kernel on the compilation machine, so that the kernel for any ODROID device may be compiled on a different ODROID device.

The resulting package includes a single-click installation script, and may be shared with others, installed on any compatible ODROID, or added to a repository for distribution via apt-get. The scripts will give prompts for the next step of the process, making it ideal for use by users who are not yet experienced in kernel compilation.

## Overview

The build scripts are located at <http://bit.ly/1U6kQcU>, and may be downloaded using `wget` from the command line or any web browser. Unzip the package after navigating to the download folder:

```
$ wget \
http://bit.ly/1U6kQcU
$ tar -xvzf build.tgz
```

The resulting “build” folder contains the following scripts:

`download.sh` is used to download a particular branch from the Hardkernel repository  
`build.sh` is used to

launch the kernel compilation process  
`install.sh` is copied to the resulting kernel installation package in order to install the kernel on any ODROID device.

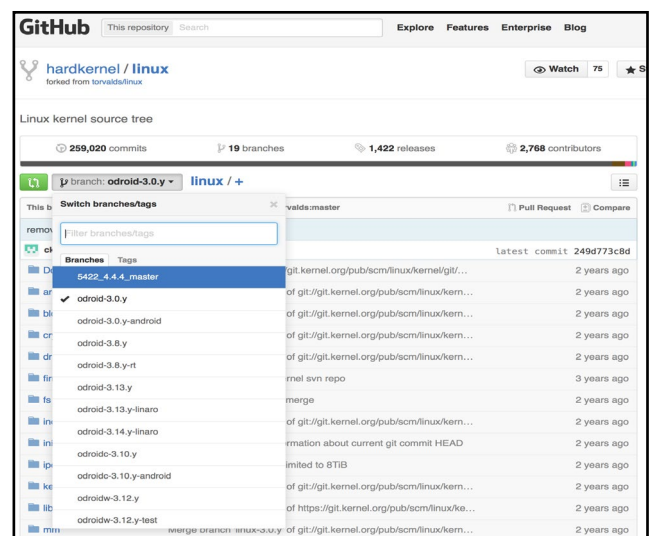
## Download script

The script “download.sh” may be used to download any available branch from the Hardkernel GitHub repository by specifying it as the first argument. For instance, to download the most recent version of the `odroidc-3.10.y` branch for the ODROID-C1, navigate to the “build” folder and type the following:

```
$ sh ./download.sh odroid-3.13.y
```

A list of branches may be obtained by visiting <http://bit.ly/1NvVQa1> and inspecting the “branch” dropdown selection menu, as shown below.

After the download script completes, the “install.sh” and “download.sh” files will automatically be copied to the downloaded branch directory in preparation for the next step.



The place to get the most up-to-date kernel is at the branch dropdown menu on Hardkernel's GitHub repository page