Contact: [[email protected]](http://www.cloudflare.com/email-protection)

### Tags: Platform

*Proof of concept that threads in a Real Time Operating System for the Raspberry Pi can emulate various Arduinos with help of a library that mimics the Arduino API. This will maximize the efficiency of the ArduSat platform by allowing multiple Arduinos to use dynamically asigned sensors. Please turn on the captions in the video for english subtitles.*

This project is solving the [**ArduSat**](https://2013.spaceappschallenge.org/challenge/ardusat) challenge.

**Description**

This project solves the Ardusat hardware level 3 challenge by using a Raspberry Pi running ChibiOS (a Real Time Operating System for embedded systems). Inside ChibiOS threads runs Arduino code with help of a library that mimics the Arduino platform. The RTOS can be configured to map pins and devices betwen the Raspberry Pi and the virtual Arduinos allowing the maximum use of the sensors in Ardustat. The code and configuration running on the Raspberry pi can be uploaded via serial console for remote management.

**Build & Run**

**1. Wire the prototype**

See [Arduinos on the Raspberry Pi schematics](https://raw.github.com/manuel-rabade/arduino-chibios-rpi/master/schematic/arduinos-on-the-raspberry-pi.png)

**2. Install the GNU ARM toolchain**

In MAC OS X you can use [Yagarto](http://www.yagarto.de/#downloadmac) and in Linux you can use [yol](https://github.com/phaenovum/yol).

**3. Get ChibiOS for the RPi**

**git clone https://github.com/steve-bate/ChibiOS-RPi**

**4. Get Arduinos on the Raspbery Pi**

**git clone https://github.com/manuel-rabade/arduino-chibios-rpi**

**5. Build**

**cd arduino-chibios-rpi && make**

**cd arduino-chibios-rpi/bootloader && make**

**6. Setup SD card**

Copy **arduino-chibios-rpi/bootloader/kernel.img** to the SD card. Download [bootcode.bin and start.elf](https://github.com/raspberrypi/firmware) and copy them to the SD card.

**7. Serial console**

Install minicom and configure it:

• Serial Device: /dev/ttyUSB0

• Bps/Par/Bits: 115200 8N1

• Hardware Flow Control: No

• Software Flow Control: No

**8. Boot**

Power your Raspberry Pi and upload **arduino-chibios-rpi/build/ch.bin** using XModem.

**Serial console**

Once ChibiOS has been uploaded the following commands are available in the serial shell:

• **status** get virtual Arduinos status

• **pause [arduino #]** pause operation of a virtual Arduino

• **resume [arduino #]** resume operation of a virtual Arduino

• **log [arduino #]** get the serial buffer of a virtual Arduino

• **mem** get ChibiOS memory usage

• **threads** get ChibiOS threads status

• **info** get ChibiOS build information

• **systime** get system time

• **reboot** reboot ChibiOS

**Pin configuration**

This project map the following Raspberry Pi pins to virtual Arduino pins:

Raspberry Pi Virtual Arduino Arduino pin  
GPIO22 arduino0 13  
GPIO7 arduino1 13  
GPIO25 arduino1 12  
GPIO17 arduino2 13  
GPIO24 arduino3 13

**Hack**

**arduino-chibios-rpi/main.cpp** it's the entry point that spawns the threads.

**arduino-chibios-rpi/arduino/arduino0.cpp** , **arduino-chibios-rpi/arduino/arduino1.cpp**, **arduino-chibios-rpi/arduino/arduino2.cpp** and **arduino-chibios-rpi/arduino/arduino3.cpp** are the code for each virtual Arduino.

**arduino-chibios-rpi/arduino-chibios-rpi/arduino/arduino0.h** , **arduino-chibios-rpi/arduino/arduino1.h**, **arduino-chibios-rpi/arduino/arduino2.h** and **arduino-chibios-rpi/arduino/arduino3.h** are the configuration for each Arduino.

**arduino-chibios-rpi/arduino/arduino.h** and **arduino-chibios-rpi/arduino/arduino.cpp** is the library that "emulates" the Arduino plataform.

**References**

• [Ardusat challenges](http://2013.spaceappschallenge.org/challenge/ardusat/)

• [ChibiOS-RPi port and demos by Steve Bate](https://github.com/steve-bate/ChibiOS-RPi)

• [Raspberry Pi bootloader by David Welch](https://github.com/dwelch67/raspberrypi/tree/master/bootloader05)

• [Arduino languaje reference and source code](http://arduino.cc/en/Reference/HomePage)

• [Raspberry Pi Low-level peripherals](http://elinux.org/RPi_Low-level_peripherals)

**Project Information**

* License: [BSD 2-Clause License](http://opensource.org/licenses/BSD-2-Clause)
* Source Code/Project URL: <https://github.com/manuel-rabade/arduino-chibios-rpi>

**Resources**

* Bootloader build and SD card setup - <http://www.youtube.com/watch?v=gKtXT4GJeRg>
* Upload, build and run - <http://www.youtube.com/watch?v=D3jWURlTU5I>
* Prototype - <https://raw.github.com/manuel-rabade/arduino-chibios-rpi/master/photo/arduinos-on-the-raspberry-pi.jpg>
* Schematic - <https://raw.github.com/manuel-rabade/arduino-chibios-rpi/master/schematic/arduinos-on-the-raspberry-pi.png>