C.A.R.S.

Gianluca Moro

Control and Acquisition on RaspberryPI Systems

Fondazione Ingegneri Padova - 7 November 2018

gianluca.moro@unipd.it - www.giammy.com

Introduction

Control, data acquisition and remote monitoring is a common requirement in almost every experimental setup; the type of control and data to be acquired is different in each specific application, and can change in the same experiment according to new requests or new ongoing analysis.

Requirements

Project CARS, Control and Acquisition on RaspberryPi Systems, requires a flexible and modular solution:

- 12 bit ADC/DAC up to 100kHz on 1 channel, or 25kHz on 4 channels
- with self contained control software
- offering both local or remote interface
- can be included in pre-existing experimental infrastructure (tested with LabView and MDSPlus)

Some options

- Single board PC:
  - https://www.raspberrypi.org/
  - http://www.orangepi.org/
  - http://www.banana-pi.org/
- Open-source electronics platform
  - https://www.arduino.cc/
- Both:
  - https://www.udoo.org/

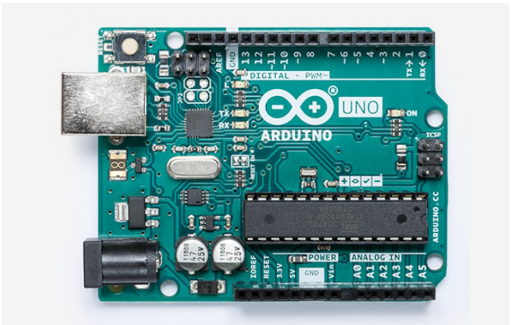
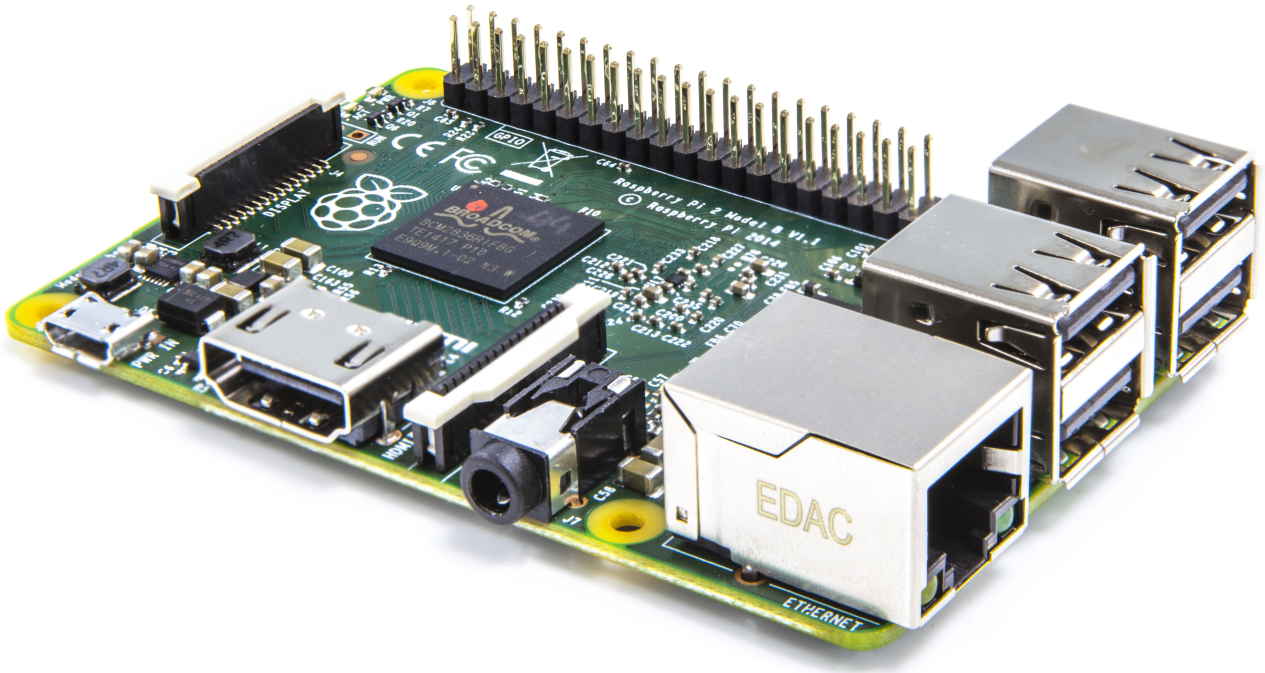


Figure: Arduino Uno

RaspberryPI 3 Model B 1/2



RaspberryPi 3 Model B 2/2

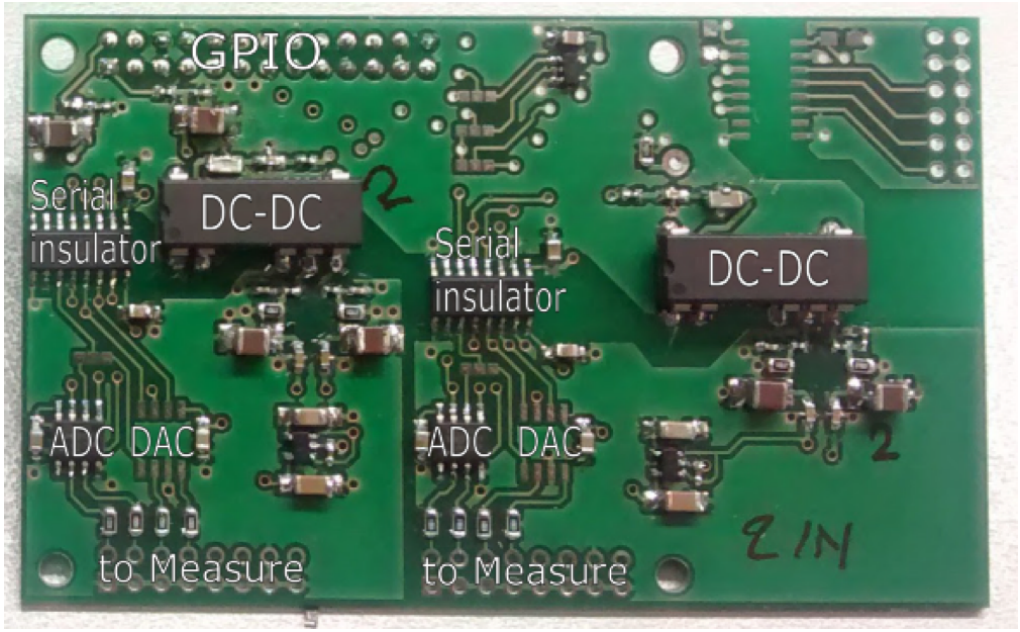
Quad Core 1.2GHz Broadcom BCM2837 64bit CPU
1GB RAM
BCM43438 wireless LAN and Bluetooth Low Energy (BLE) on board
100 Base Ethernet
40-pin extended GPIO
4 USB 2 ports
4 Pole stereo output and composite video port
Full size HDMI
CSI camera port for connecting a Raspberry Pi camera
DSI display port for connecting a Raspberry Pi touchscreen display
Micro SD port for loading your operating system and storing data
Upgraded switched Micro USB power source up to 2.5A

Project hardware

The framework has been developed on

- Raspberry Pi v.3 board, model B
- R-Adapt acquisition board with 2 slots for MCP3202 ADC and MCP4822 DAC
- standard Raspbian operating system with support for SPI to control the ADC/DAC.
- acquisition program, shell scripts, web server

Extension board (in house solution) 1/2



R-Adapt extension board

Extension board (in house solution) 2/2

- Connects to Raspberry PI extended GPIO
- Uses SPI, Serial Peripheral Interface (MOSI,MISO,SCLK,CS)
- Opto isolated
- Can host 2 chips at the same time, and can be configured via hardware dip switch to manage:
  - 4 12 bit ADC channels
  - 4 12 bit DAC channels
  - 2 12 bit ADC channels and 2 12 bit DAC channels

Control and Acquisition fundamentals

- Input
  - ADC
  - signal status
  - commands (UI, WEB, external control system)
- Output
  - DAC
  - logical signal
  - data monitor (web or other external tools)
- Storage
  - internal
  - external

  both database or dedicated solutions

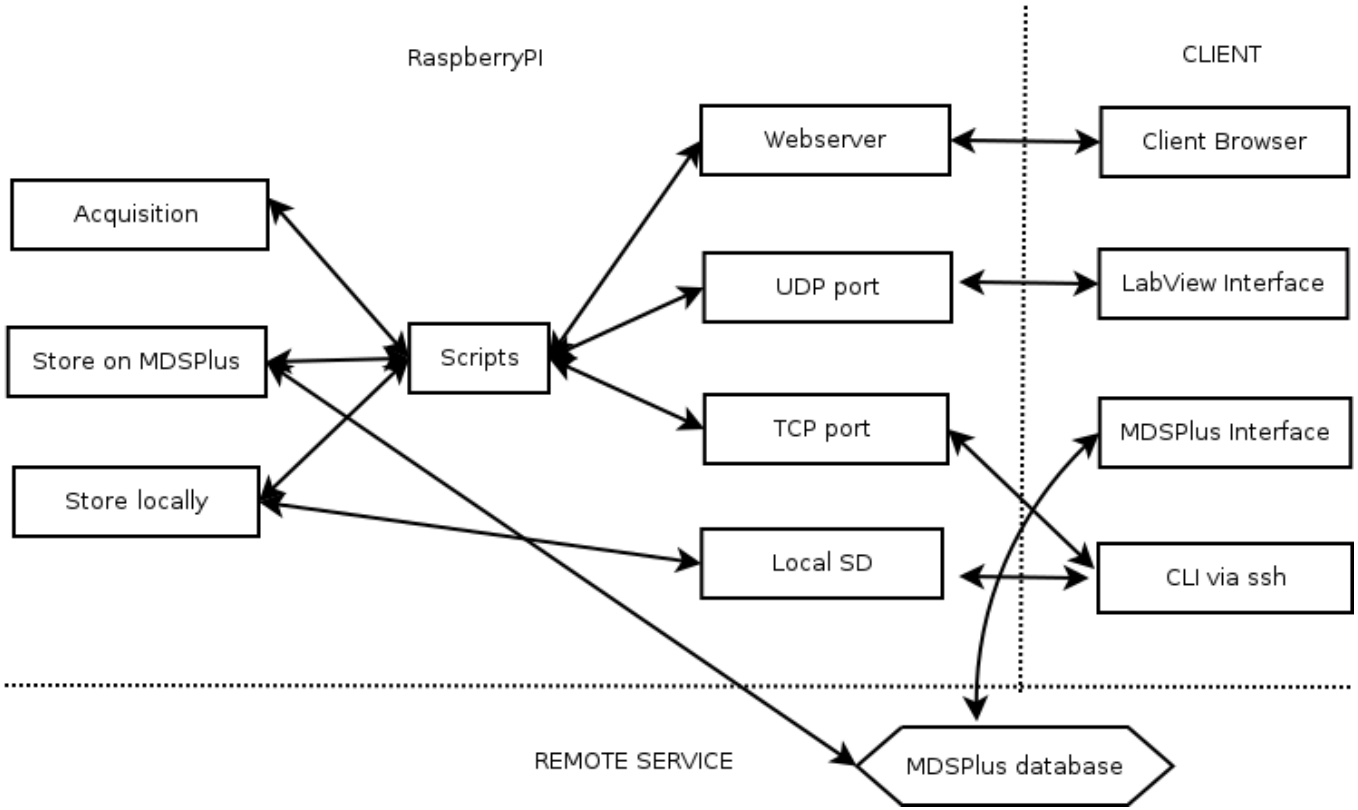
Environment 1/2

The solution needs to be **extremely modular** with

- base tasks(ADC/DAC control, MDSPlus data storage) implemented in separate programs (C)
- TCP/IP connections managed by standard Linux utilities (such as `socat`)

making the result extremely flexible, efficient and adaptable to different configurations, using a standard Linux platform.

Environment 2/2



Structure

Building blocks 1/3

Acquisition C program to control the ADC/DAC via SPI: a command line utility which can be configured by command line arguments

Interface various methods are present, and can be used at the same time:

- webserver (`websocketd`) giving both a remote user interface and a WebSocket channel for commands
- UDP/TCP ports, made available with `socat` linux utility

Building blocks 2/3

Storage can be done both locally on the SD, or remotely on a MDSPlus database

Integration the different tools are configurable using command line arguments: the integration between all these utilities is done using shell scripts so the configuration is easily adaptable to the ongoing requests.

Building blocks 3/3

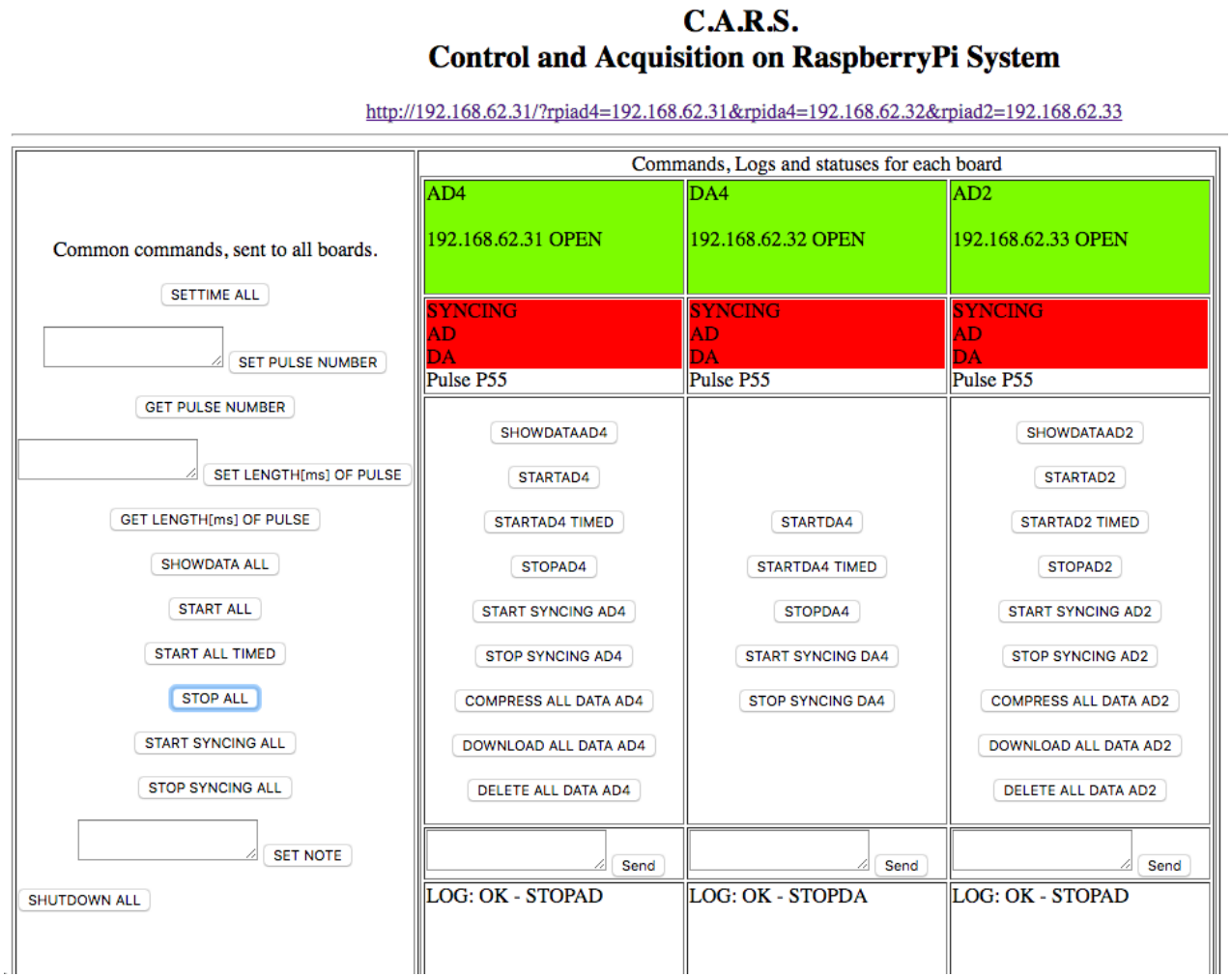
Modular composition with Unix pipes:

```
stdbuf -oL cars-acquisition | stdbuf -oL cars-store |
socat - UDP-DATAGRAM:192.168.1.1:9000,broadcast
```

A command-server:

```
socat TCP4-LISTEN:6000,reuseaddr,fork
EXEC:/usr/local/bin/server.sh,fdin=3,fdout=4
```

Web interface 1/2



Web interface

Web interface 2/2

Interface a local web server is available to allow a self-contained solution controlled by a web browser.
Data is exchanged via Websocket.
CLI is always available via `ssh`.
The framework can be integrated in experimental setups based on MDSPlus or Labview.

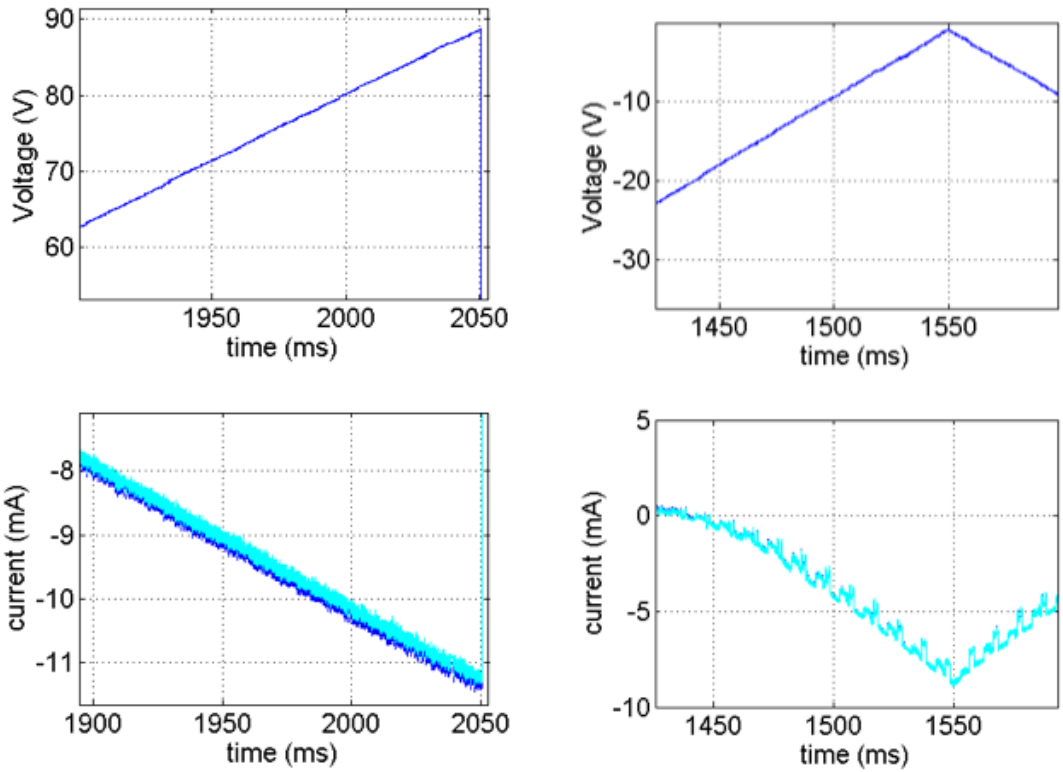
Results

The acquisition part has been implemented for slow measures: the tested implementation are all running on standard linux kernel (the kernel with real-time extension has been used, but it is not needed for these working frequencies). The requested setups runs from 100/1000Hz (using linux timers for timing) to a maximum of 100kHz (the hardware limit of the used ADC/DAC SPI, using the BCM library `bcm2835-1.50.tar.gz` available at `https://gist.github.com/annem/3183536`), which means, an acquisition frequency of up to 25kHz (acquiring from 4 channels).

Results

The framework provides acquisition of raw data, which can be elaborated externally: here we report some graphs of data acquired in a Langmuir probe setup on NIO1 ((Negative Ion Optimization phase 1) experiment, a moderate size negative ion source (60 kV, 9 beamlets of 15 mA H- each).
We show just the calibrated data (converted to Volt and Ampere) which is the starting point for the following analysis.

Results



Comparison of shot in vacuum (no plasma), on the left and with plasma on the right.

Results

We can note the (unexpected) noise having a frequency of about 100 Hz: in the post-acquisition analysis it resulted as a physical effect present in the plasma.

Conclusions

The framework has been used in these setups:

Current measures on NIO1: 1 RaspberryPi integrated in a Labview control system and MDSPlus storage/monitoring software

Standalone RFEA diagnostic: a cluster of 3 RaspberryPi, 1 master with a control web interface and 2 slaves, giving a total of 4 DA channels (1kHz) and 8 AD channels (25kHz)

Langmuir probe: on NIO1 experiment, 1 RaspberryPi in standalone mode, with 2 DAC and 2 ADC

The different setups have shown the flexibility of the framework, included a multiboard configuration.

References


Current sense board for plasma NIO1 with R_Adapt, B. Laterza, M. Zanini, G. Serianni, G. Moro, D. Ravarotto, RFX-NIO-TN-036.


Development of a system for conversion ADC and DAC voltage, with RaspberryPI, called R_Adapt, B. Laterza, R. Cavazzana, G. Moro, D. Ravarotto, G. Serianni, M. Recchia, RFX-NIO-TN-038.


System for voltage control and for data acquisition of Retarder Field Energy Analyzer (RFEA), B. Laterza, M. Brombin, E. Sartori, A. Pimazzoni, G. Moro,P. Veltri, RFX-NIO-TN-039.


NIO1 A versatile negative ion source, M. Cavenago et al., Proceedings of IPAC'10, Kyoto, Japan