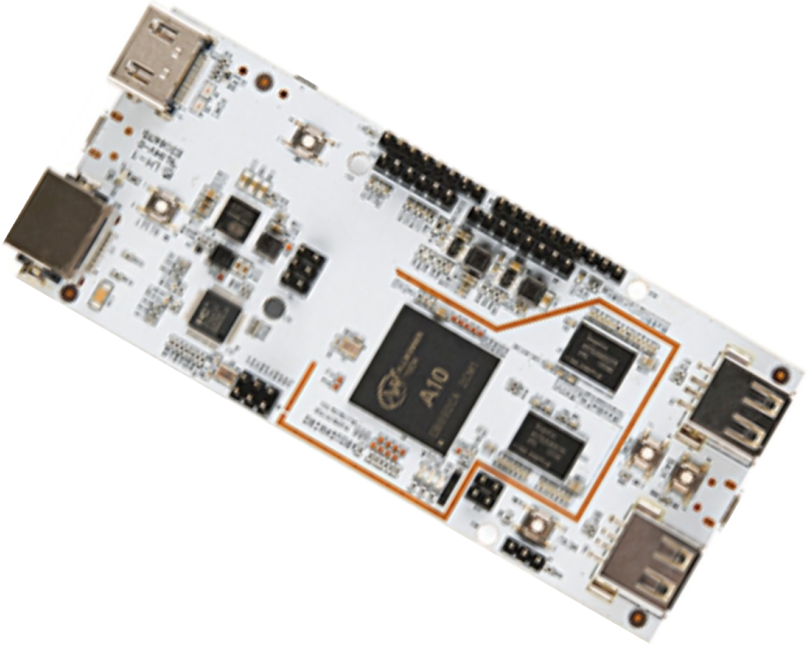


PiN GO

APi "UNIVERSAL" PARA HARDWARE HACKING

Mini Bio

- Luciano Ramalho, programador-repentista
- Desenvolvedor Web desde 1994
  - IDG Now, Brasil Online, UOL/BOL, AOL Brasil...
- Professor
  - para profissionais: Python.pro.br
  - para crianças e adolescentes: Oficina Turing
- Palestrante: FISL, PyCon US, OSCON, TDC...
- Co-fundador do Garoa Hacker Clube

PiNGO: FEiTO NO GAROA

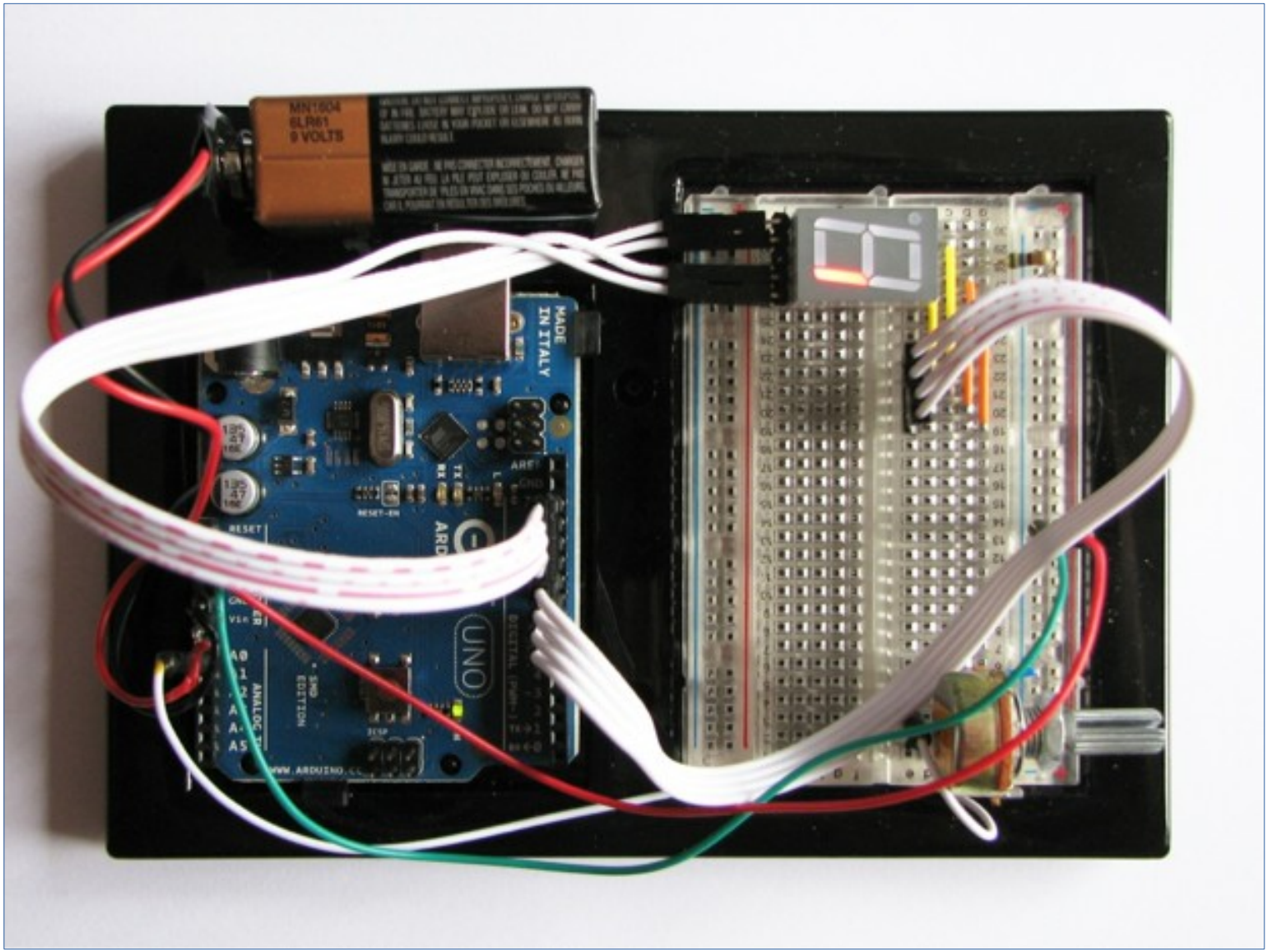
- Garoa Hacker Clube (garoa.net.br)
- Um **hackerspace**
  == laboratório comunitário **independente**
- Em São Paulo (próximo do metrô **Pinheiros**)
- Mantido por uma associação sem fins lucrativos operada pelos **associados**
  - hoje somos 46 associados pagando mensalidades de R$ 60 ou R$ 100
  - isso paga o aluguel, a manutenção da casa e permite a compra de equipamentos

O “PROBLEMA”

Pinos GPIO

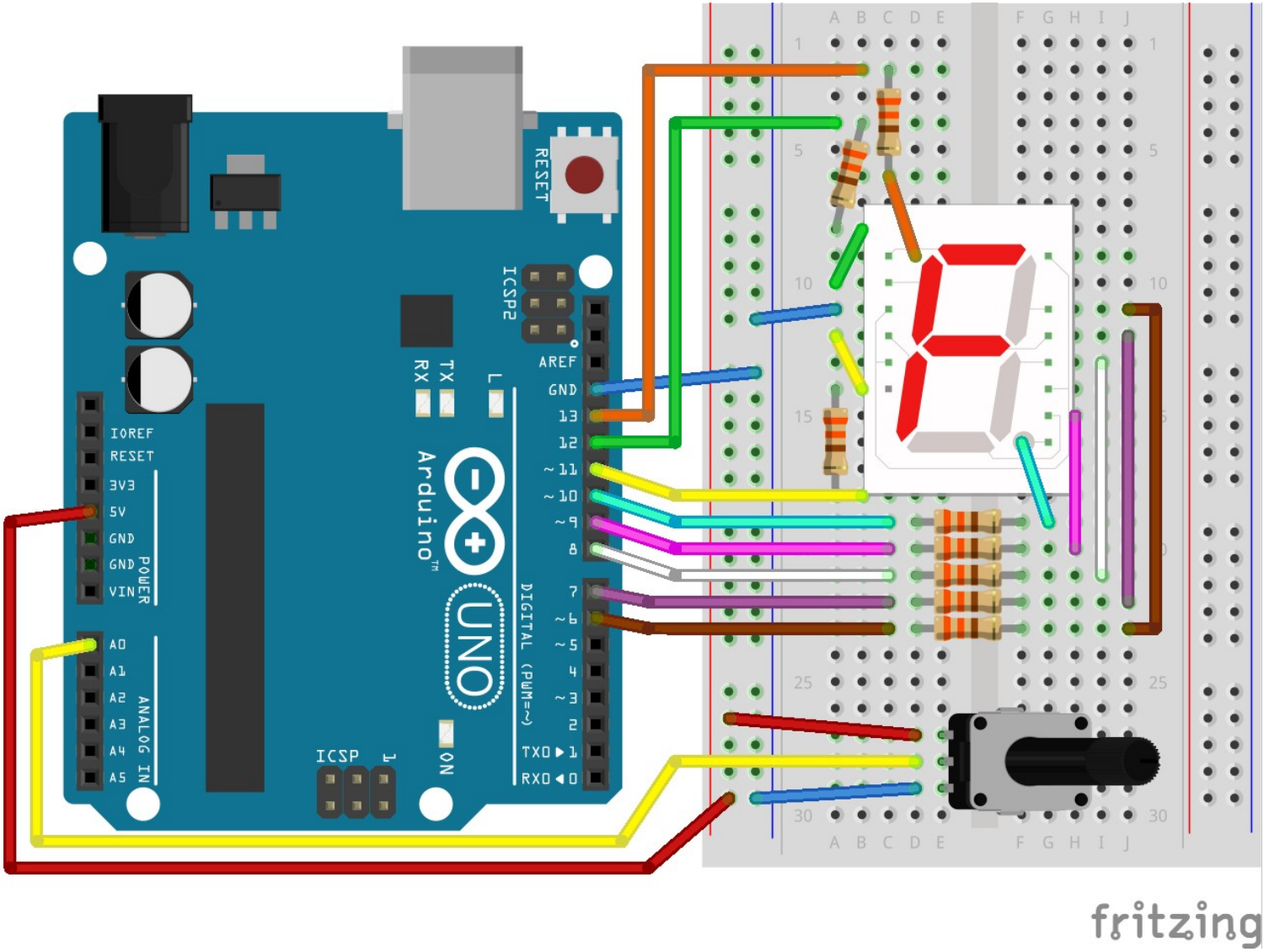
- Todas as placas interessantes para computação física possuem pinos **GPIO**
  - General Purpose I/O
- Pinos **digitais** I/O
  - 0/1 == HIGH/LOW
- Pinos **analógicos**
  - conversor A/D
- Pinos digitais com suporte a **PWM**
  - substituto saída analógica sem conversor D/A

DOJO COM ARDUINO



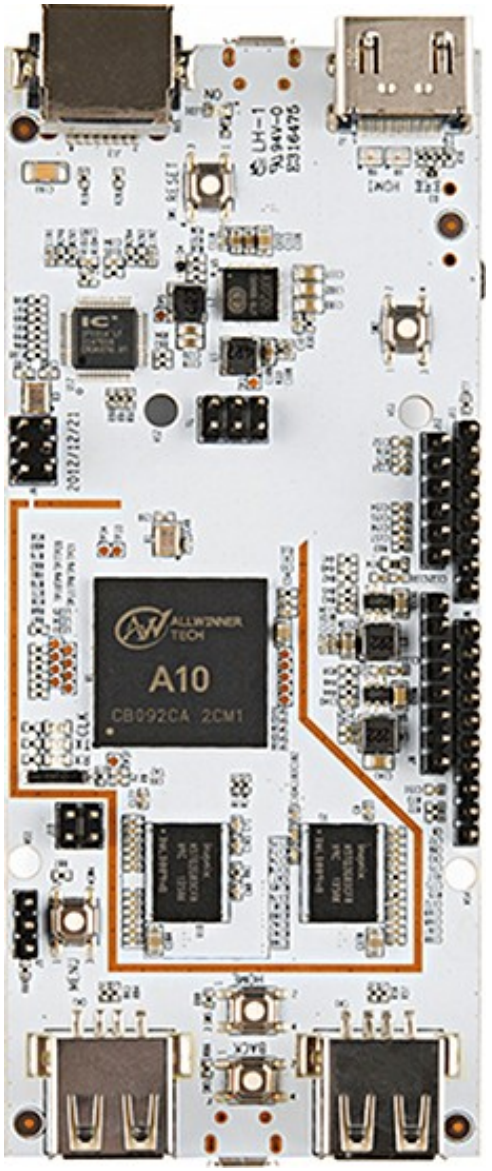
GAROA HACKER CLUBE

DOJO COM ARDUINO



GAROA HACKER CLUBE

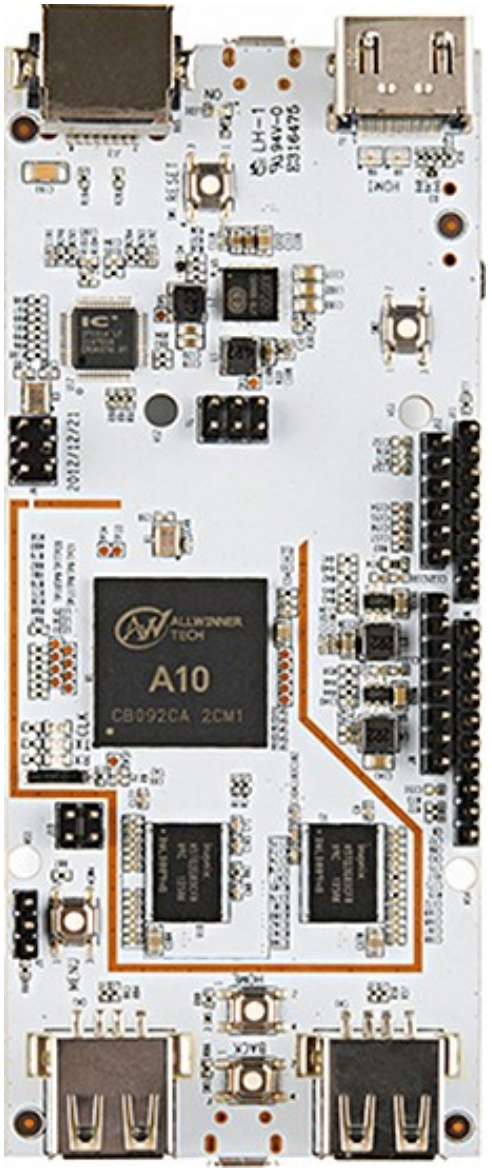
PCDUINO

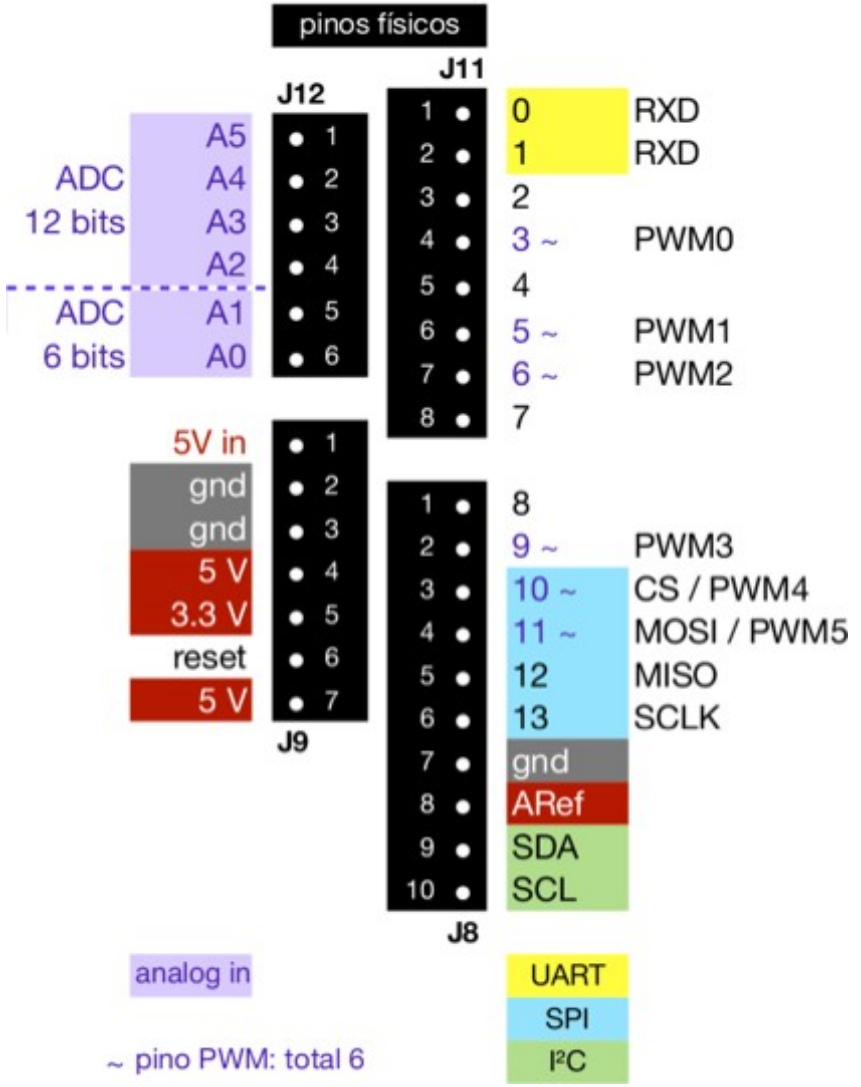


	pcDuino
SoC	Allwinner A10
CPU	ARM Cortex A8 (ARMv7 + NEON SIMD)
GPU	Mali 400
clock	1 GHz
SO	GNU/Linux (vários) Android
RAM	1 GB
Flash onboard	2 GB
SD card	micro-SD
GPIO	14
PWM	6
ADC	6
USB client	1
USB host	1
Ethernet	10/100
Wifi	✘
HDMI	HDMI
video composto	✘
audio	via HDMI
preço USD	59
Arduinos	1.7
open hardware	?

GAROA HACKER CLUBE

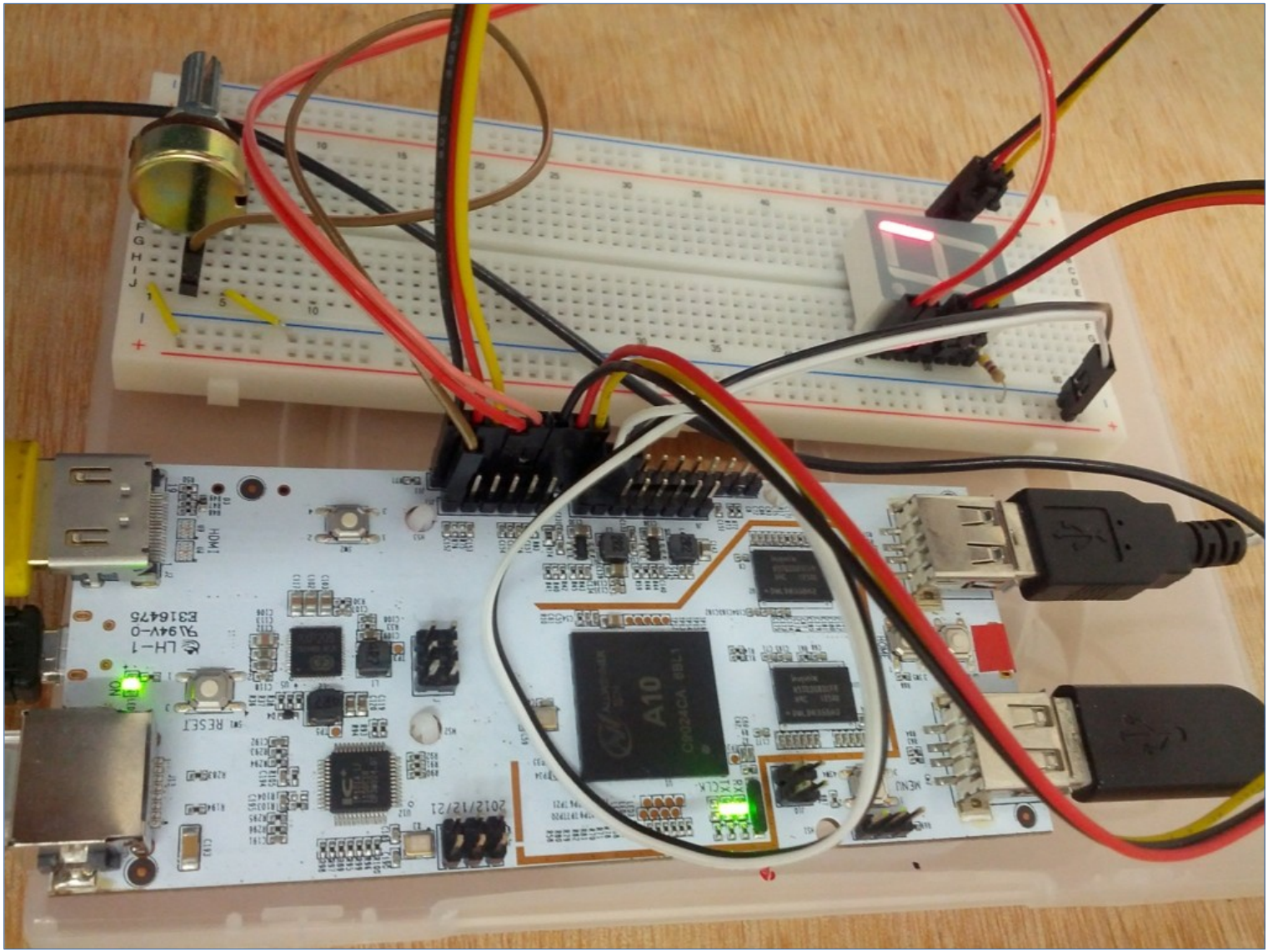
PCDUINO



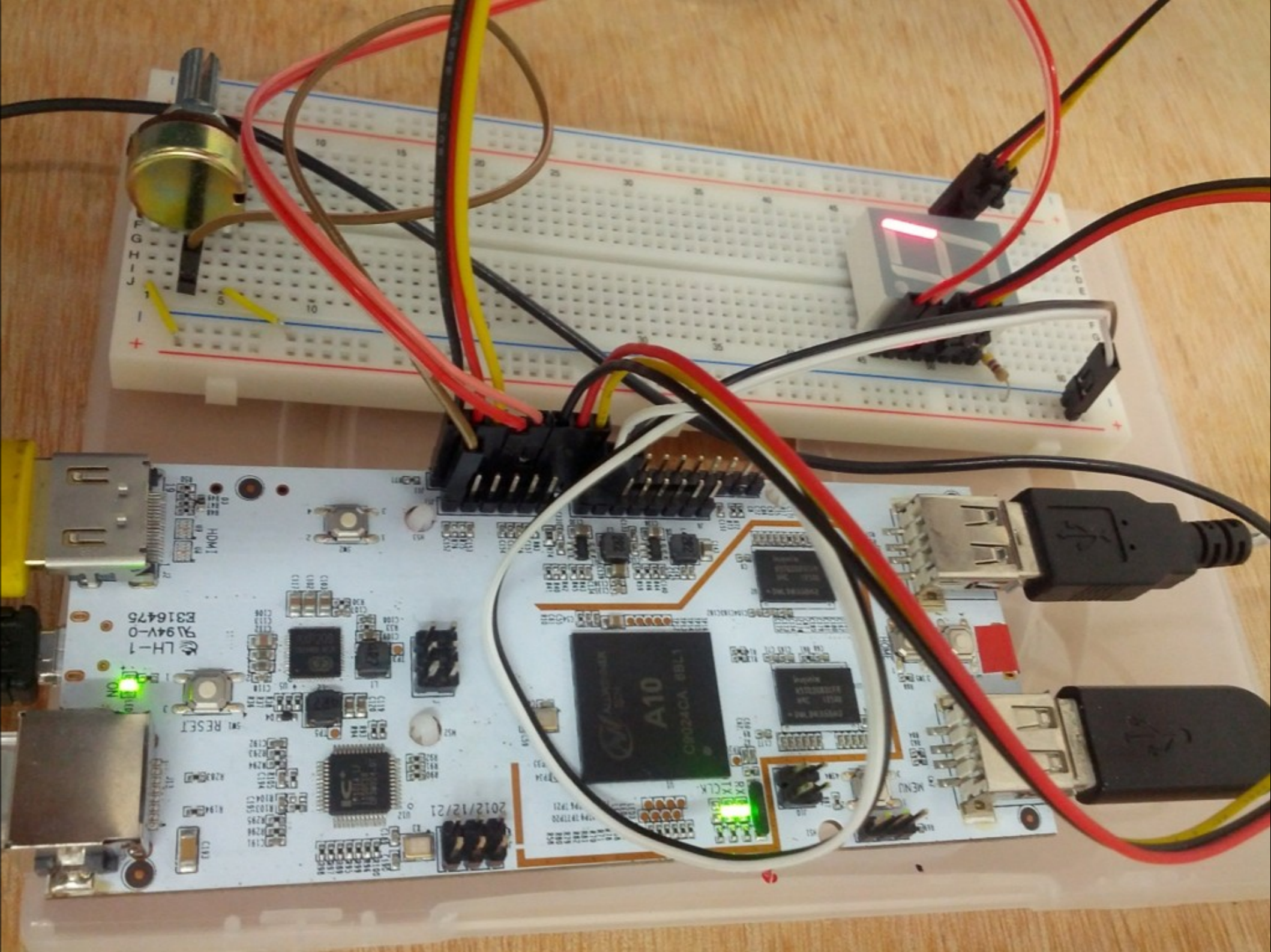


GAROA HACKER CLUBE

DOJO COM PCDUINO



GAROA HACKER CLUBE


A10
HDMI
RESET
2012/12/21

RASPBERRY Pi

	Raspberry Pi mod. B
SoC	Broadcom BCM2835
CPU	ARM11 (ARMv6)
GPU	VideoCore IV
clock	700 MHz
SO	GNU/Linux (vários)
RAM	512 MB
Flash onboard	✘
SD card	SD
GPIO	17 (+4 no P5)
PWM	1
ADC	✘
USB client	✘
USB host	2
Ethernet	10/100
Wifi	✘
HDMI	HDMI
video composto	RCA
audio	HDMI + plug 3.5mm
preço USD	35
Arduinos	1.0
open hardware	✘

GAROA HACKER CLUBE

RASPBERRY Pi

P1

P1

		pinos físicos			
	3.3 V	1	2	5 V	
SDA	* 0→2	3	4	5 V	
SCL	* 1→3	5	6	gnd	
GPCLK0	4	7	8	14	TXD
	gnd	9	10	15	RXD
	17	11	12	18 ~	PCM_CLK
	* 21→27	13	14	gnd	
	22	15	16	23	
	3.3 V	17	18	24	
MOSI	10	19	20	gnd	
MISO	9	21	22	25	
SCLK	11	23	24	8	CE0
	gnd	25	26	7	CE1

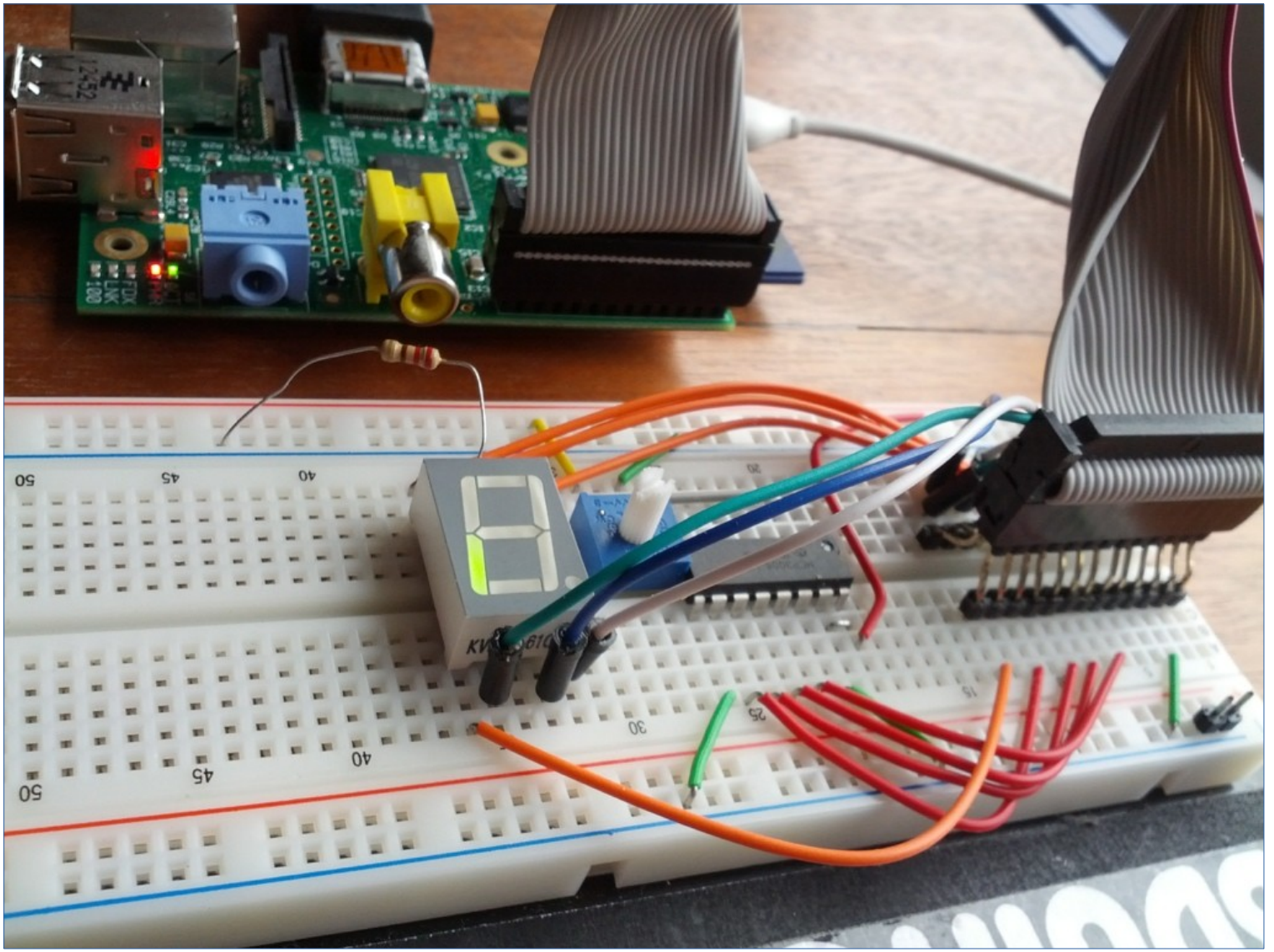
~ pino PWM: 12 (GPIO_18)

* pinos GPIO renumerados rev.1→rev.2

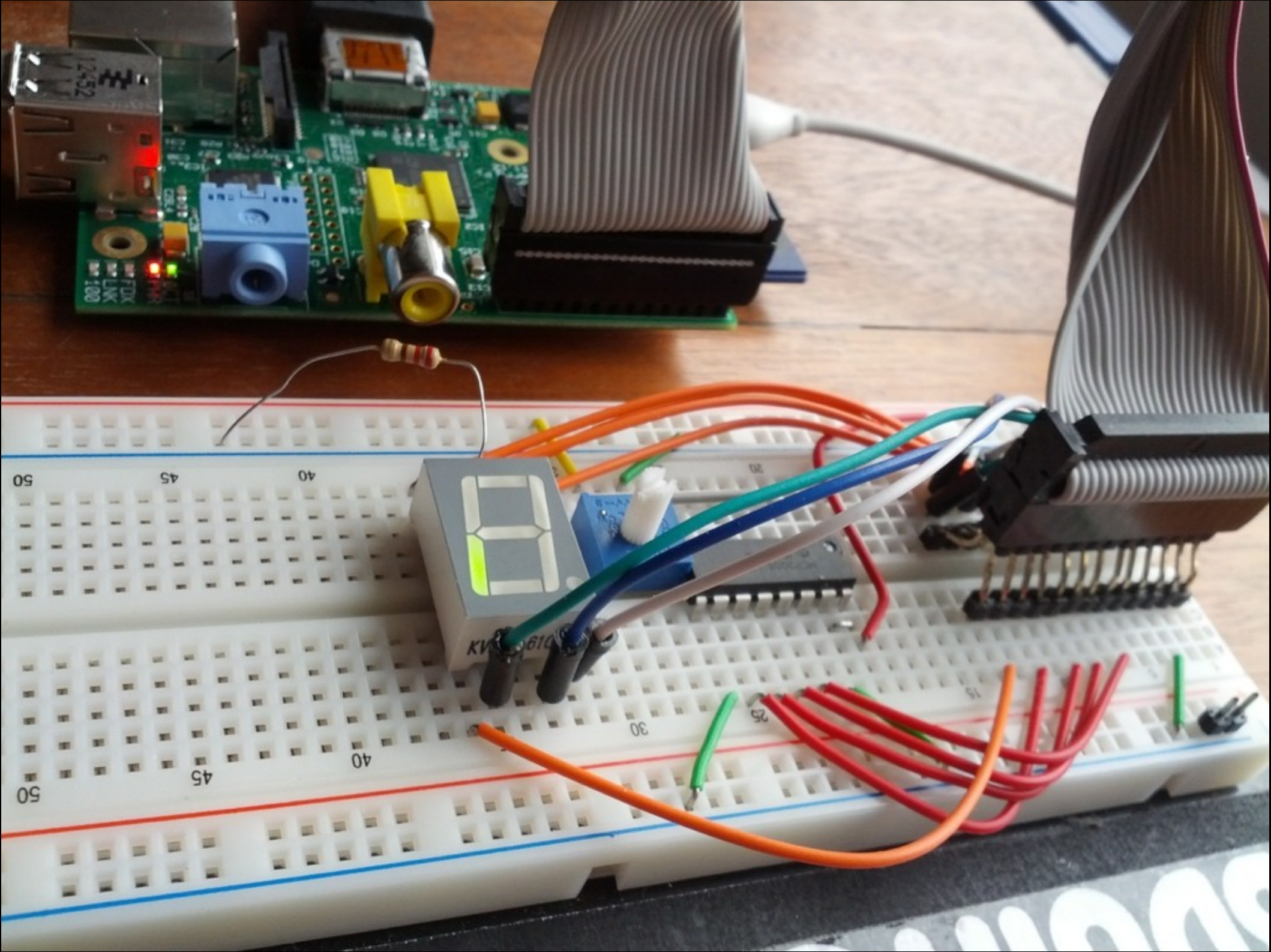
I²C
UART
SPI

GAROA HACKER CLUBE

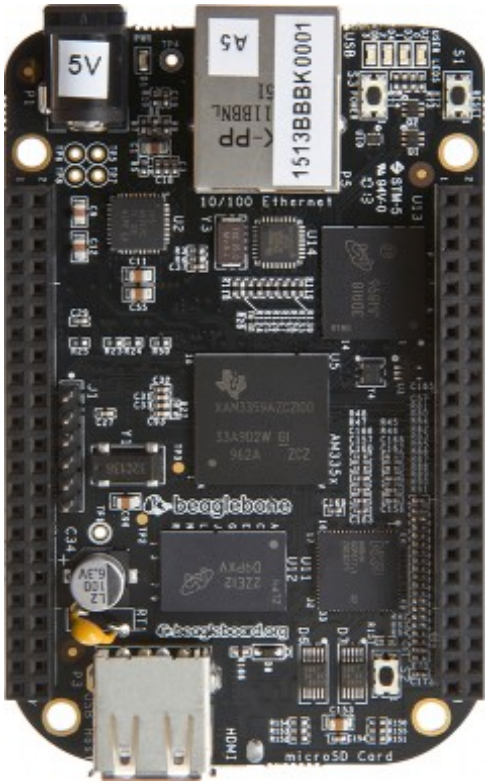
DOJO COM RASPBERRY Pi



GAROA
HACKER CLUBE



BEAGLEBONE BLACK



	Beaglebone Black
SoC	Texas Instruments Sitara AM335x
CPU	**ARM Cortex A8 (ARMv7 + NEON SIMD)**
GPU	PowerVR SGX 530
clock	**1 GHz**
SO	GNU/Linux (vários) Android
RAM	512 MB
Flash onboard	**2 GB**
SD card	**micro-SD**
GPIO	**66**
PWM	**8**
ADC	**7**
USB client	**1**
USB host	1
Ethernet	10/100
Wifi	✘
HDMI	micro-HDMI
video composto	✘
audio	via HDMI
preço USD	45
Arduinos	1.3
open hardware	✔

GAROA HACKER CLUBE

BEAGLEBONE BLACK

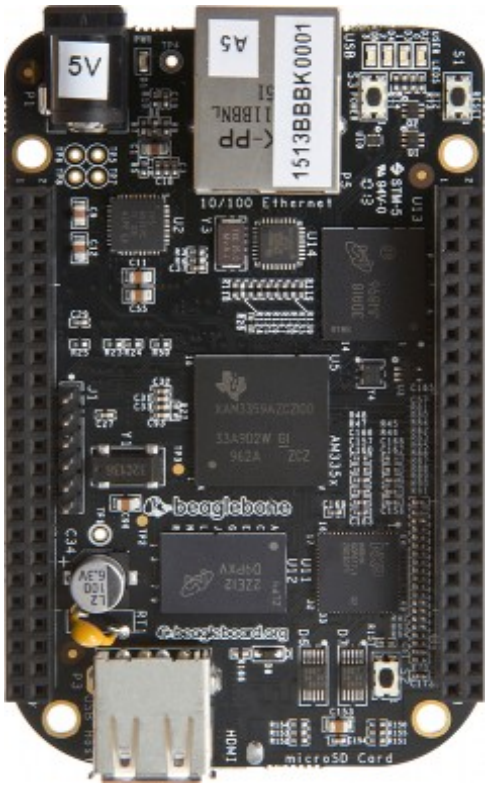
P9

	pinos físicos	
gnd	1 2	gnd
3.3 V	3 4	3.3 V
VDD 5 V	5 6	5 V VDD
SYS 5 V	7 8	5 V SYS
PWR_BUT	9 10	SYS_RESETn
UART4_RXD	11 12	1-28
UART4_TXD	13 14	EHRPWM1A ~
1-16	15 16	EHRPWM1B ~
I2C1_SCL	17 18	I2C1_SDA
I2C2_SCL	19 20	I2C2_SDA
UART2_TXD	21 22	UART2_RXD
1-17	23 24	UART1_TXD
* 3-21	25 26	UART1_RXD
3-19	27 28	SPI1_CS0 ~
SPI1_D0	29 30	SPI1_D1
SPI1_SCLK	31 32	VADC
AIN4	33 34	AGND
AIN6	35 36	AIN5
AIN2	37 38	AIN3
AIN0	39 40	AIN1
[2] CLKOUT2, 3-20	41 42	0-7 ~, 3-18 [2]
gnd	43 44	gnd
gnd	45 46	gnd

x-yy pino GPIOX_YY
~ pino PWM: 14, 16, 28, 42[2] (0-7)
* em uso: oscilador audio HDMI
[2] pino físico ligado a dois pinos lógicos

analog input
I²C
UART
SPI
misc.

P9 P8



P8

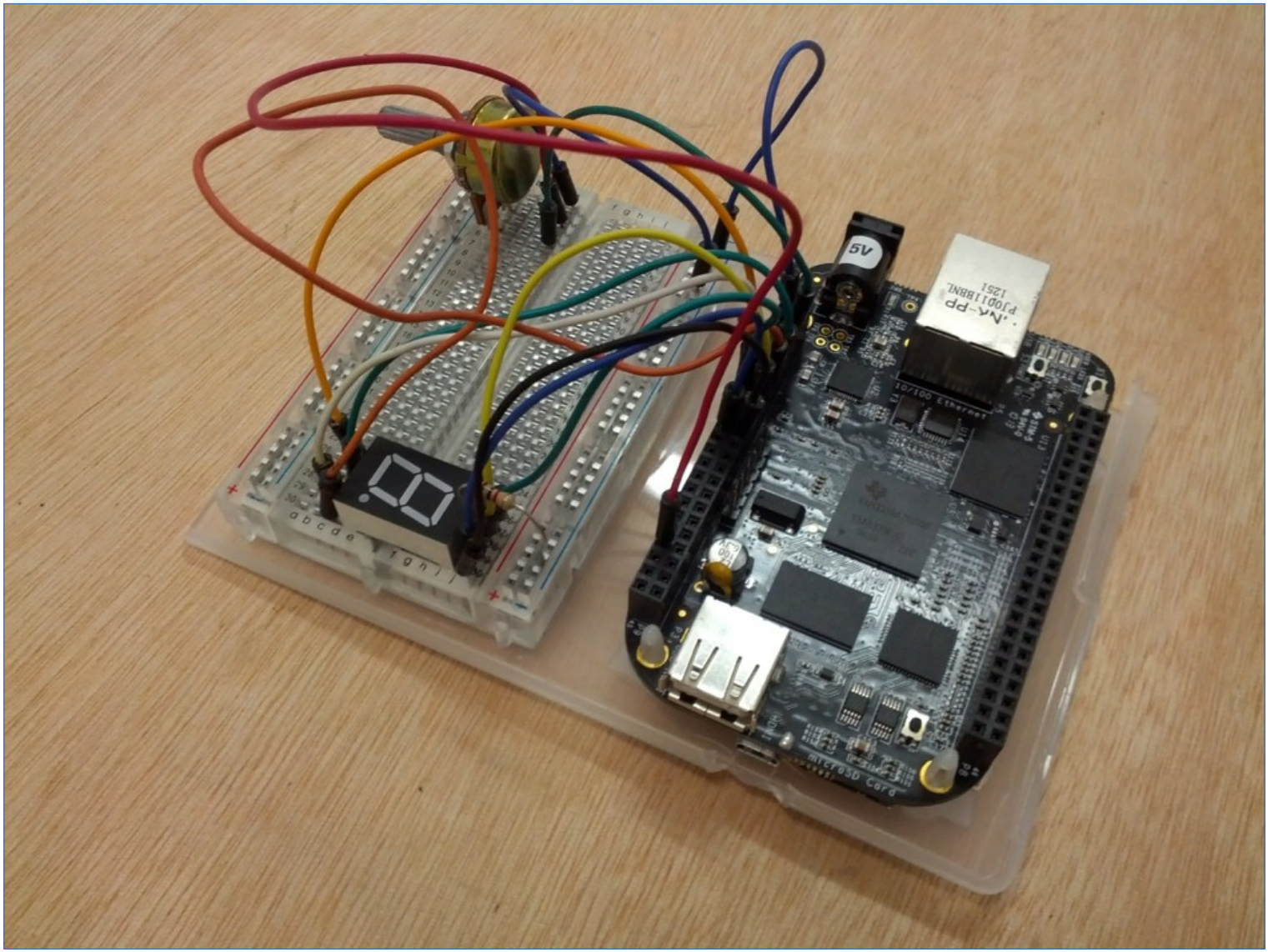
	pinos físicos	
gnd	1 2	gnd
1-6	3 4	1-7
1-2	5 6	1-3
TIMER4	7 8	TIMER7
TIMER5	9 10	TIMER6
1-13	11 12	1-12
~ EHRPWM2B	13 14	0-26
1-15	15 16	1-14
0-27	17 18	2-1
~ EHRPWM2A	19 20	1-31
1-30	21 22	1-5
1-4	23 24	1-1
1-0	25 26	1-29
2-22	27 28	2-24
2-23	29 30	2-25
UART5_CTSN	31 32	UART5_RTSN
UART4_RTSN	33 34	UART3_RTSN
UART4_CTSN	35 36	UART3_CTSN
UART5_TXD	37 38	UART5_RXD
2-12	39 40	2-13
2-10	41 42	2-11
2-8	43 44	2-9
2-6	45 46	2-27

x-yy pino GPIOX_YY
~ pino PWM: 13, 19

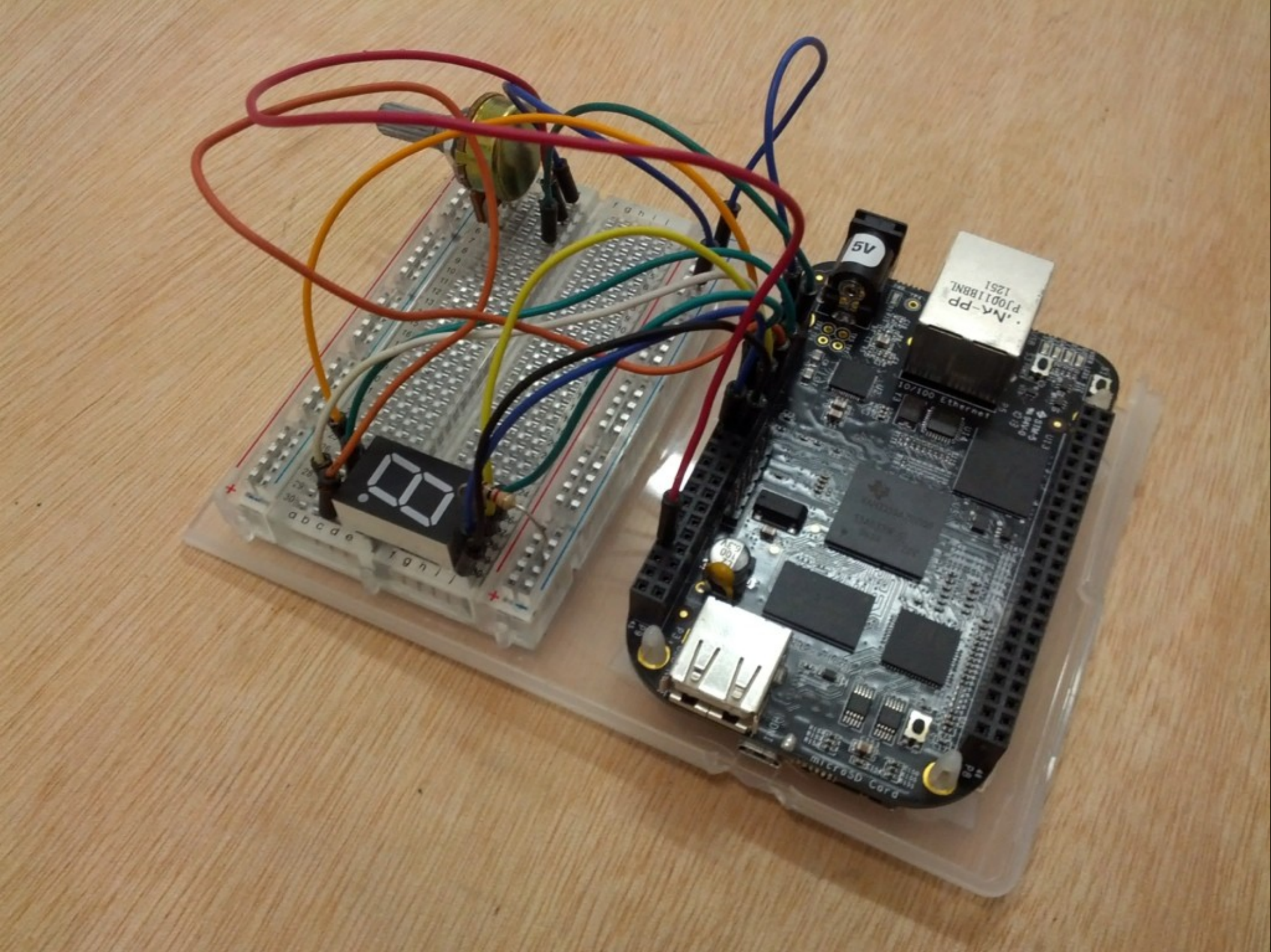
UART
timers

GAROA HACKER CLUBE

DOJO COM BBB



GAROA
HACKER CLUBE


5V
10/100 Ethernet
MICROSD Card

PROGRAMANDO PINOS

- Quase todas essas placas são programáveis em **Python**
- Python embarcado, naquelas que rodam **GNU/Linux** embarcado
- Python em **outro** computador, controlando remotamente
  - ex: **Arduino** via serial com protocolo **Firmata** e biblioteca **pyFirmata**

```
import time, os

GPIO_PATH = os.path.normpath('/sys/devices/virtual/misc/gpio/')
ADC_PATH = os.path.normpath('/proc/')

INPUT = LOW = "0"
OUTPUT = HIGH = "1"

def pin_mode(pin, mode):
    with open(GPIO_PATH+'mode/gpio%s' % pin, 'w') as f:
        f.write(mode)

def digital_write(pin, value):
    with open(GPIO_PATH+'pin/gpio%s' % pin, 'w') as f:
        f.write(str(value))

def analog_read(pin):
    with open(ADC_PATH+'adc%d' % pin) as f:
        f.seek(0)
        return int(f.read(16).split(':')[1])

def setup():
    for i in range(18):
        pin_mode(i, OUTPUT)
        digital_write(i, LOW)

setup()
while True:
    for i in [0, 1, 7, 5, 4, 2]:
        digital_write(i, 1)
        delay = analog_read(5)/4096.0
        time.sleep(delay)
        digital_write(i, 0)
```

SysFS: pinos GPIO mapeados em arquivos

Nenhuma biblioteca especial é usada neste script!

GAROA HACKER CLUBE

PCDUINO

```
import atexit
import time
import RPi.GPIO as GPIO
import spi

# executar cleanup ao sair
atexit.register(GPIO.cleanup)

# usar numeração lógica dos pinos
GPIO.setmode(GPIO.BCM)

DISPLAY = [17, 4, 9, 11, 7, 27, 22, 10]

SPI_CLK = 18
SPI_MISO = 23
SPI_MOSI = 24
SPI_CS = 25
conversor_ad = spi.Mcp3008(SPI_CLK, SPI_MISO, SPI_MOSI, SPI_CS)

CANAL_POTENCIOMETRO = 1

for led in DISPLAY[:6]:
    GPIO.setup(led, GPIO.OUT)
    GPIO.output(led, 0)

while True:
    for led in DISPLAY[:6]:
        GPIO.output(led, 1)
        atraso = conversor_ad.read(CANAL_POTENCIOMETRO)/1000.0
        time.sleep(atraso)
        GPIO.output(led, 0)
```

Duas bibliotecas específicas: RPi.GPIO e spi (feita em casa)

RASPBERRY Pi

GAROA HACKER CLUBE

```
import Adafruit_BBIO.GPIO as GPIO
import Adafruit_BBIO.ADC as ADC

ADC.setup()

from time import sleep
pinos = [16, 21, 22, 13, 12, 11]

for pino in pinos:
    GPIO.setup("P9_" + str(pino), GPIO.OUT)

while True:
    for pino in pinos:
        GPIO.output("P9_" + str(pino), GPIO.HIGH)
        tempo = ADC.read('P9_39')
        print tempo
        sleep(tempo)
        GPIO.output("P9_" + str(pino), GPIO.LOW)
```

Biblioteca específica: Adafruit_BBIO

GAROA HACKER CLUBE

BEAGLEBONE BLACK

NOSSA SOLUÇÃO

DEMO

BLINK: MODO INTERATIVO

```
>>> from pingo import *
>>> ard = detect.MyBoard()
>>> ard
<ArduinoFirmata '/dev/tty.usbmodemfa1341'>
>>> ard.pins
{0: <DigitalPin @0>, 1: <DigitalPin @1>, 2: <DigitalPin @2>, 3: <DigitalPin
@3>, 4: <DigitalPin @4>, 5: <DigitalPin @5>, 6: <DigitalPin @6>, 7: <Digita
lPin @7>, 8: <DigitalPin @8>, 9: <DigitalPin @9>, 10: <DigitalPin @10>, 11:
<DigitalPin @11>, 12: <DigitalPin @12>, 13: <DigitalPin @13>, 'A1': <Analog
Pin @A1>, 'A0': <AnalogPin @A0>, 'A3': <AnalogPin @A3>, 'A2': <AnalogPin @A
2>, 'A5': <AnalogPin @A5>, 'A4': <AnalogPin @A4>}
>>> p13 = ard.pins[13]
>>> p13.mode = OUT
>>> p13.hi()
>>> p13.lo()
>>> from time import sleep
>>> p13.toggle()
>>> p13.toggle()
>>> p13.toggle()
>>> while True:
...     p13.toggle()
...     sleep(.1)
...
```

GAROA HACKER CLUBE

BLINK: SCRIPT 1

```
import time
import pingo

ard =
pingo.arduino.ArduinoFirmata('/dev/tty.usbmodemfa1341')
led = ard.pins[13]
led.mode = pingo.OUT

while True:
    led.toggle()
    time.sleep(.1)
```

GAROA HACKER CLUBE

BLINK: SCRIPT 2

```
import time
import pingo

ard = pingo.arduino.get_arduino()
led = ard.pins[13]
led.mode = pingo.OUT

while True:
    led.toggle()
    time.sleep(.1)
```

Detecta Arduino em porta serial/USB

GAROA HACKER CLUBE

BLINK: SCRIPT 3

```
import time
import pingo

board = pingo.detect.MyBoard()
led = board.pins[13]
led.mode = pingo.OUT

while True:
    led.toggle()
    time.sleep(.1)
```

Detecta qualquer placa suportada

GAROA HACKER CLUBE

DOJO SCRIPT

```
import time
import pingo

board = pingo.detect.MyBoard()
print('board: %s' % board)
pot = board.pins['A0']
leds = board.digital_pins[6:13]

for led in leds:
    led.mode = pingo.OUT

while True:
    for led in leds:
        if led.location == 9:
            continue
        led.high()
        time.sleep(pot.ratio())
        led.low()
```

AnalogPin: 'A0'

DigitalPins: 6 a 12

Script compatível com qualquer placa com suporte a AnalogPin.

Para algumas placas, será preciso editar a localização dos pinos.

GAROA HACKER CLUBE

CONCEITOS

PRINCÍPIOS

- API orientada a objeto
  - exploração interativa
  - fácil de estender para novos dispositivos
- Usabilidade
  - default: pinos identificados por localização física
  - atributos de conveniência: digital_pins, toggle...
- Boas práticas
  - Python idiomático
  - testes automatizados (o maior desafio)

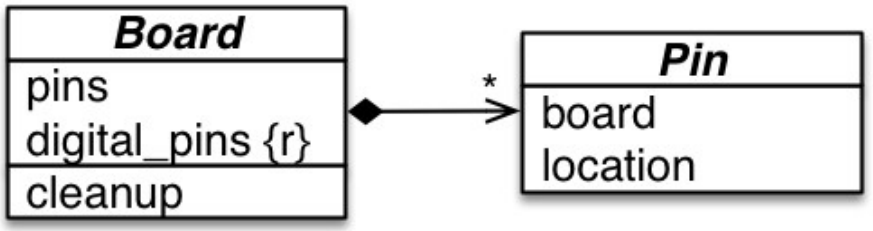
DRIVERS

- Implementam métodos de controle específicos de cada placa
- Disponíveis em maio de 2014

driver	operação	funcionalidade
ArduinoFirmata	remota	digital + analógica
PcDuino	na placa	digital + analógica
Raspberry Pi	na placa	digital
UDOO	na placa	digital

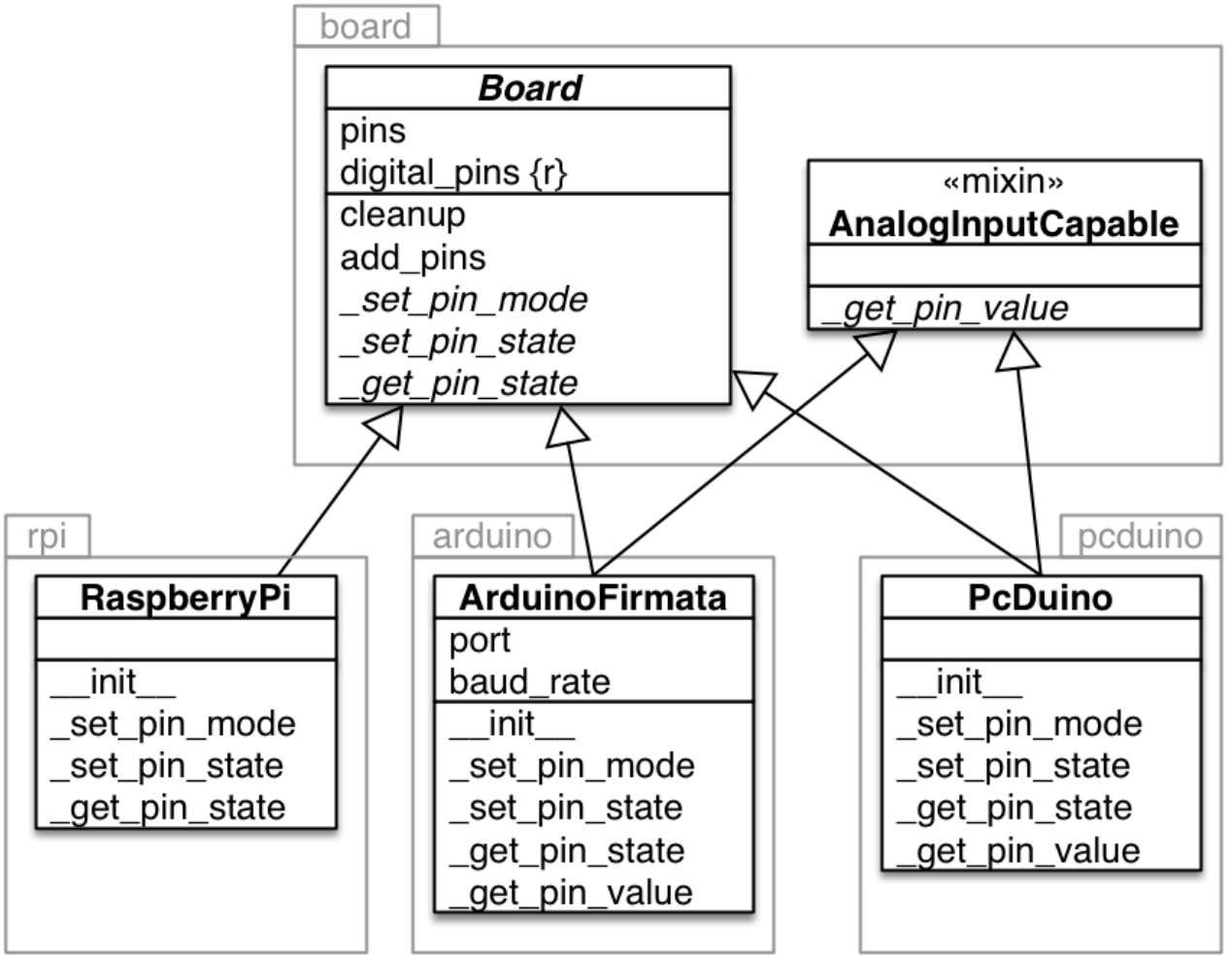
OBJETOS BÁSICOS

- pingo.Board (placa)
  - pingo.arduino.ArduinoFirmata
  - pingo.rpi.RaspberryPi
  - pingo.pcduino.PcDuino
  - ...
- pingo.Pin (pino)
  - pingo.DigitalPin
  - pingo.AnalogPin
    - ...



GAROA HACKER CLUBE

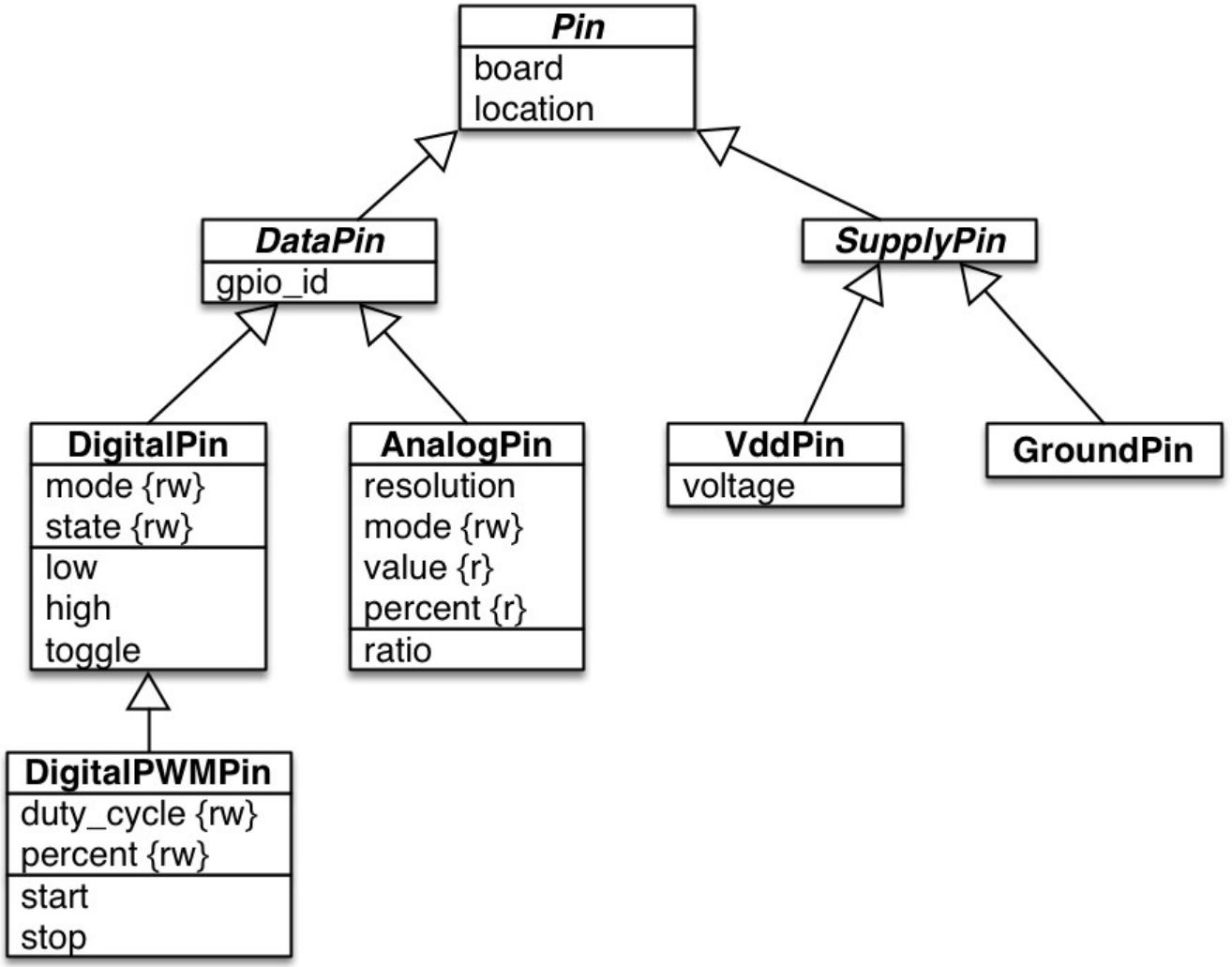
PLACAS



• • •

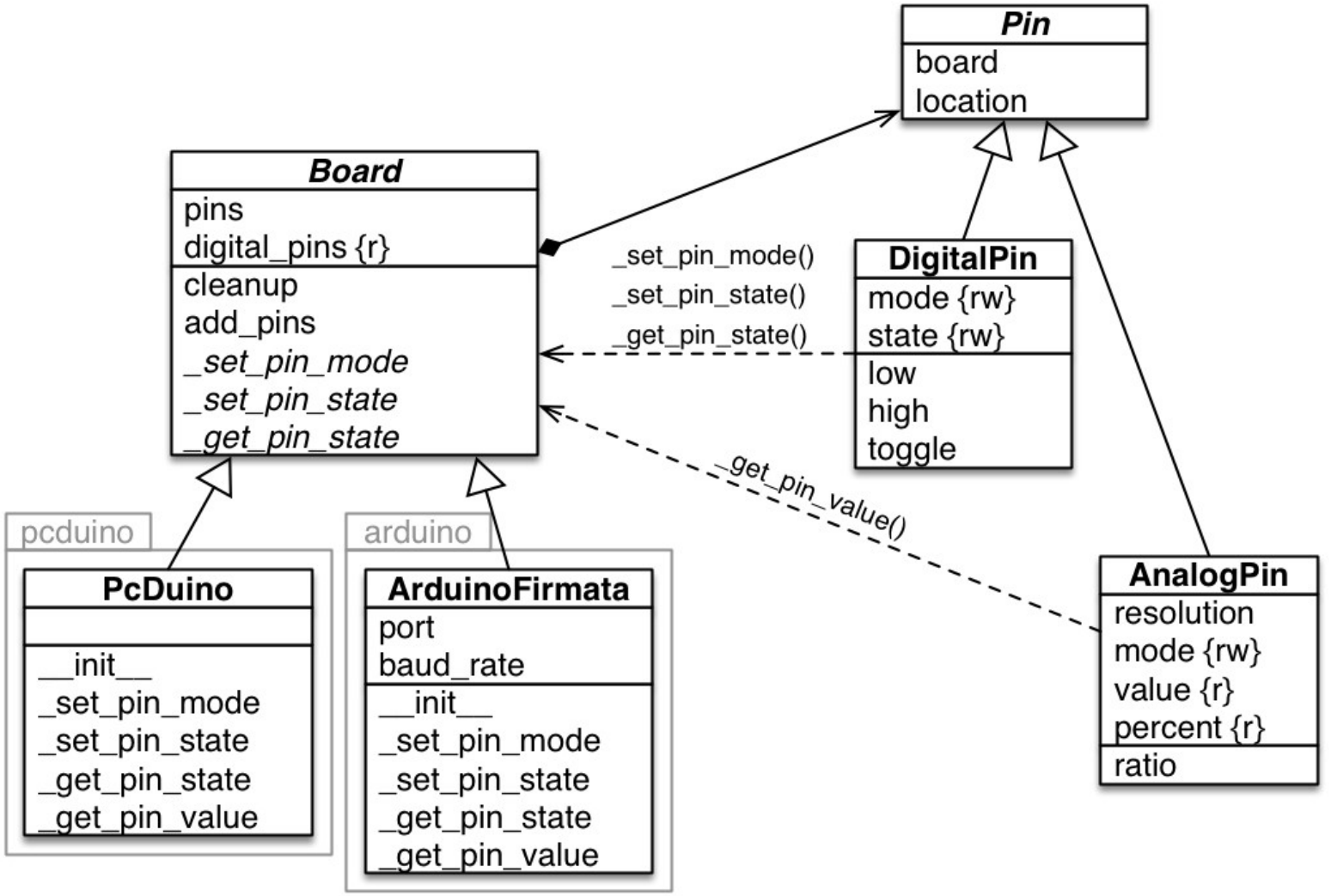
GAROA HACKER CLUBE

PiNOS



GAROA HACKER CLUBE

COLABORAÇÕES

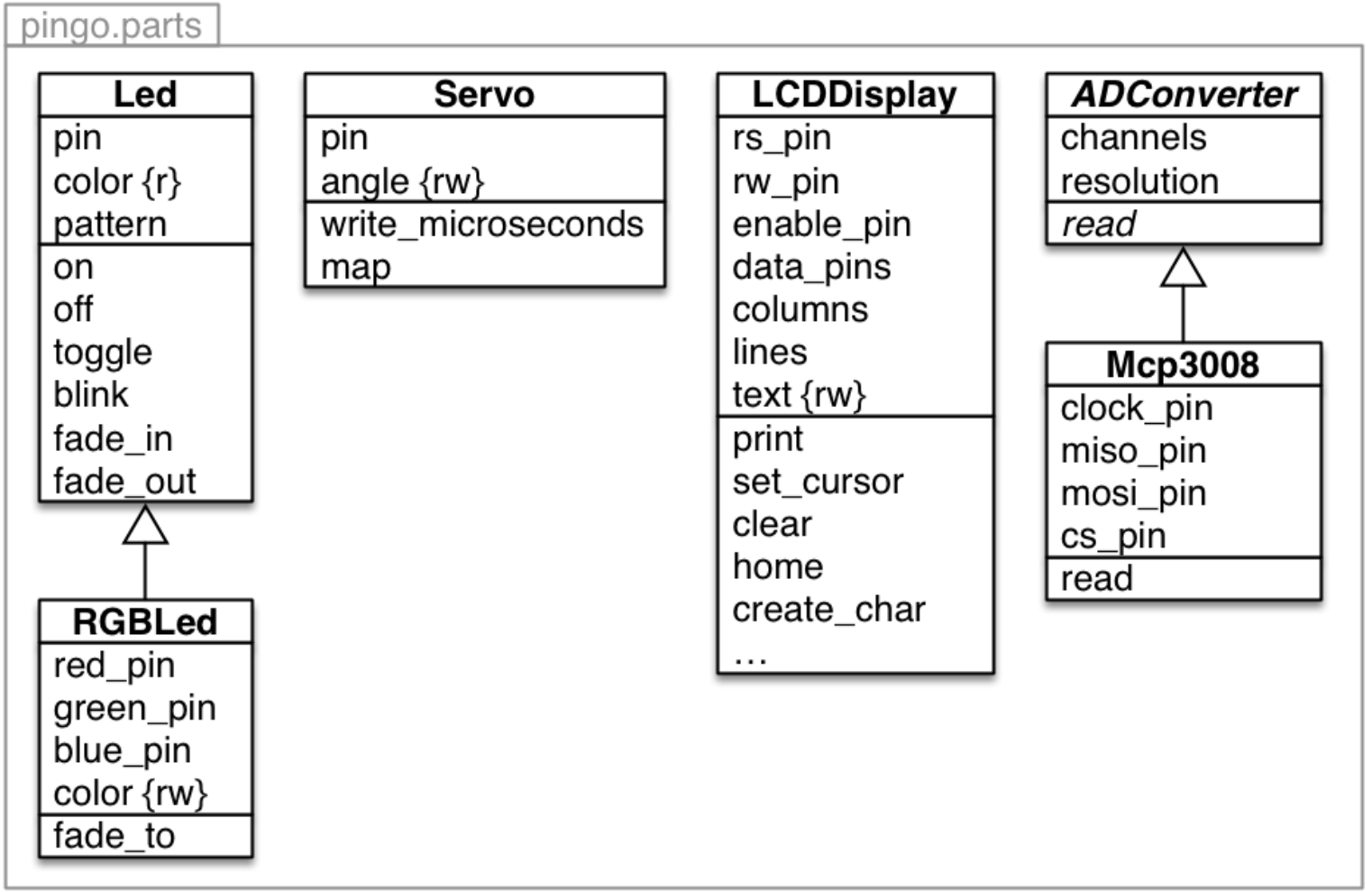


GAROA HACKER CLUBE

PLANOS

- Drivers para: Intel Galileo, Arduino Yún, BeagleBone Black, Arduino Tre
- Reestruturar sistema de testes
  - com mock e testes distribuídos em 'cluster' de placas físicas
- Implementar pinos especializados: PWM, DAC, multi-função digital/analógico...
- Implementar protocolos: SPI, I2C...
- Implementar componentes

COMPONENTES: PACOTE PINGO PARTS



GAROA HACKER CLUBE

CONTRIBUIÇÕES

- Lucas Vido, colaborador do Pingo, está contribuindo com o projeto PyFirmata, implementando o suporte a detecção automática de pinos do Firmata 2.2
- Precisamos:
  - usuários que experimentem, reportem falhas e sugiram melhorias
  - especialistas em testes automatizados distribuídos em Python

JUNTE-SE A NÓS!

- Projeto de software livre no estágio inicial
  - decisões de arquitetura interessantes
  - qualquer contribuição faz diferença
  - reuniões quinzenais: 2ª-feira, 19:30 no Garoa

- Site com documentação: http://pingo.io
- Repositório: http://github.com/garoa/pingo
- Google Groups: pingo-io
  http://groups.google.com/forum/#!forum/pingo-io

GAROA: #COMOFAZ

- Todas as atividades são **abertas** a **todos** os interessados
  - **não** precisa se associar para participar
  - mas é preciso **participar** para poder se associar!
- Aberto todos os dias úteis a partir de 19:30
  - às vezes também no fim de semana
- Veja a programação no site:

http://garoa.net.br