PHARO IoT

Present and Future

Allex Oliveira
allex.oliveira@msn.com
twitter.com/allex_oliveira

Marcus Denker
marcus.denker@inria.fr
twitter.com/marcusdenker

Nobert Hartl
norbert@2denker.de
twitter.com/NobertHartl

Summary

1 - Overview

- Created by **Rmod Team**, a research team from **INRIA** (France)
- Written by Denis Kudriashov in 2016/17
  dionisiydk@gmail.com
- In 2018, Allex Oliveira joined the Rmod Team to continue the project

What is Pharo IoT?

- A **Pharo image** running on IoT device (ARM VM)
  - A Pharo library to control GPIOs (PharoThings)
- A **Remote IDE**
  - Remote Playground, Browser, Inspectors
  - An advanced board inspector for **Raspberry PI**

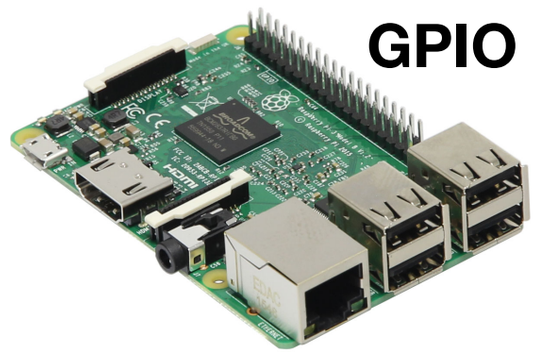
- Other IoT Projects:
  - A Pharo library to control **Arduino** Devices (Firmata)

What is Pharo IoT?

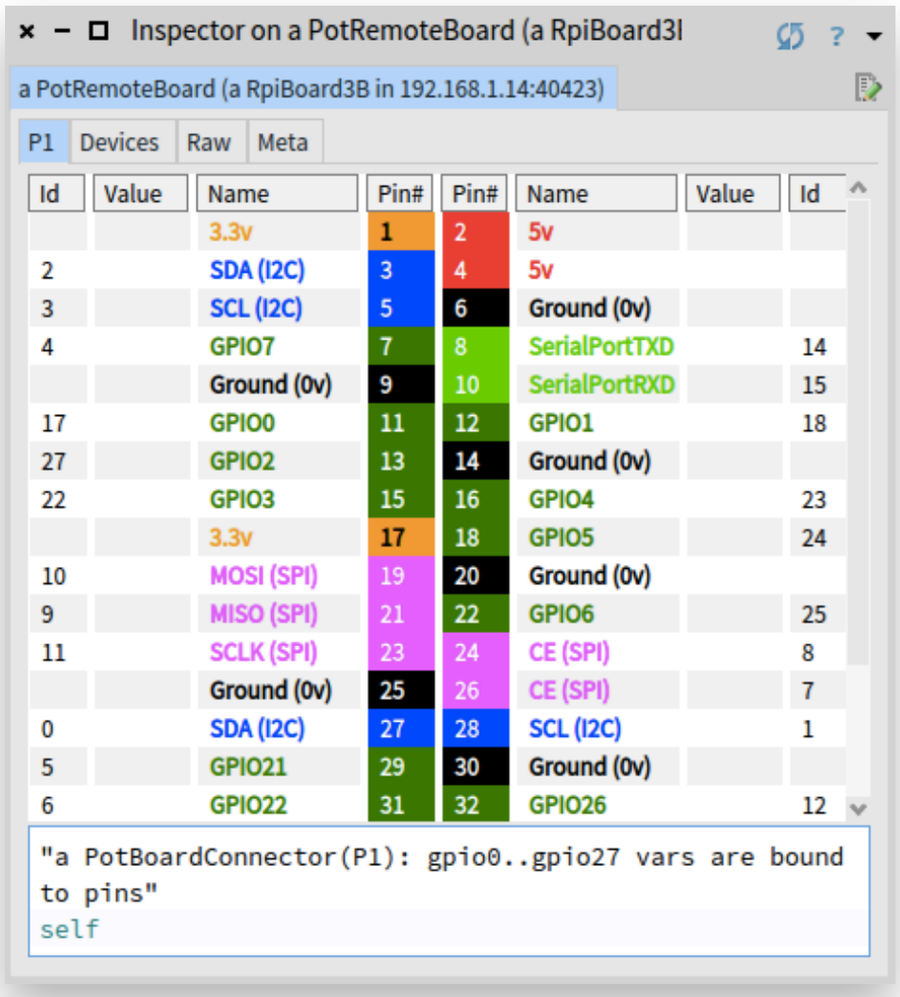
Pharo Things IDE

TelePharo
Remote
Communication

Pharo Things

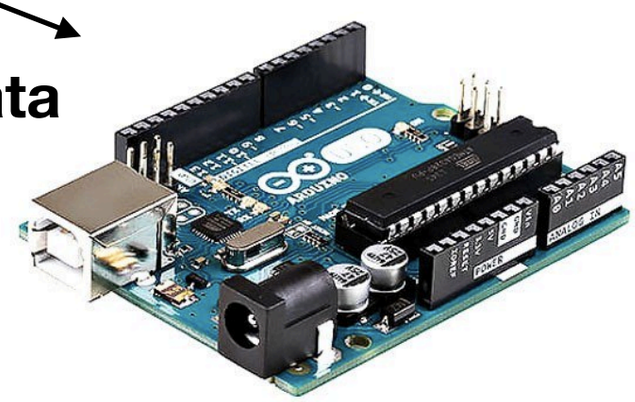


GPIO



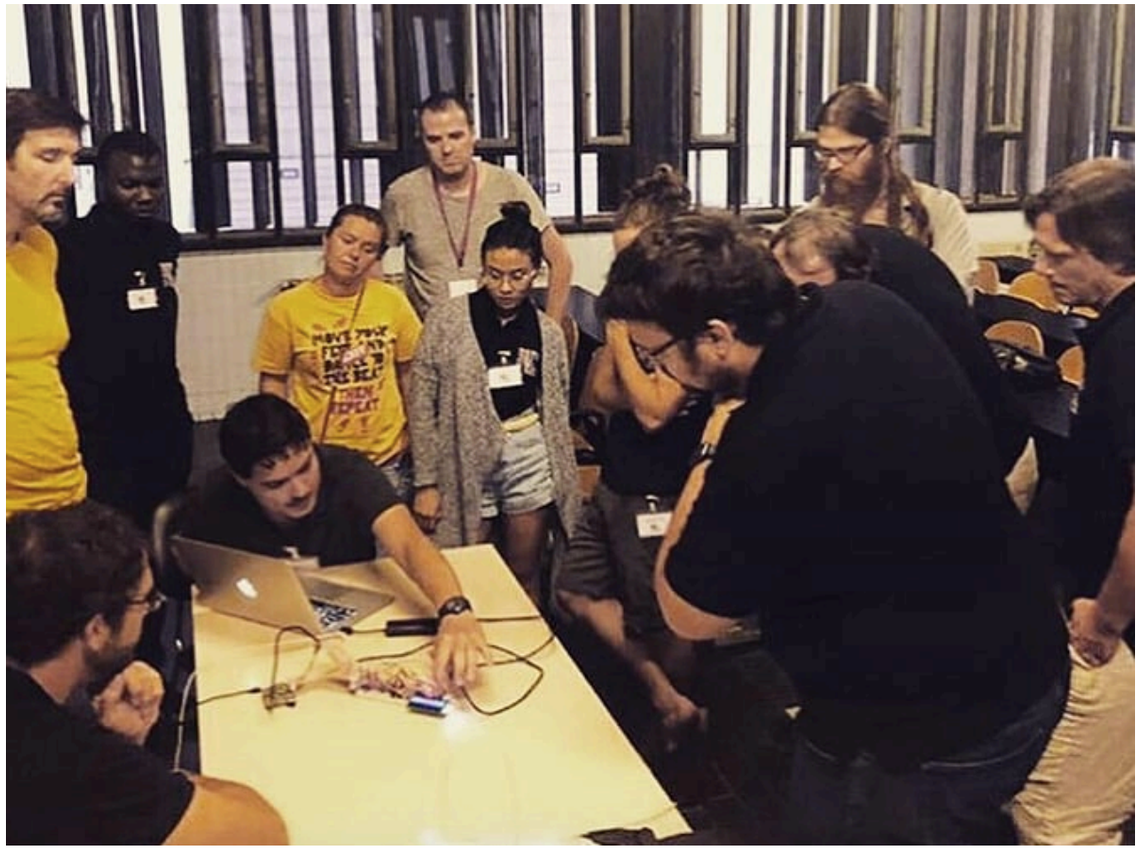
Id	Value	Name	Pin#	Pin#	Name	Value	Id
		3.3v	1	2	5v		
2		SDA (I2C)	3	4	5v		
3		SCL (I2C)	5	6	Ground (0v)		
4		GPIO7	7	8	SerialPortTXD		14
		Ground (0v)	9	10	SerialPortRXD		15
17		GPIO0	11	12	GPIO1		18
27		GPIO2	13	14	Ground (0v)		
22		GPIO3	15	16	GPIO4		23
		3.3v	17	18	GPIO5		24
10		MOSI (SPI)	19	20	Ground (0v)		
9		MISO (SPI)	21	22	GPIO6		25
11		SCLK (SPI)	23	24	CE (SPI)		8
		Ground (0v)	25	26	CE (SPI)		7
0		SDA (I2C)	27	28	SCL (I2C)		1
5		GPIO21	29	30	Ground (0v)		
6		GPIO22	31	32	GPIO26		12

Firmata



Pharo IoT in the world

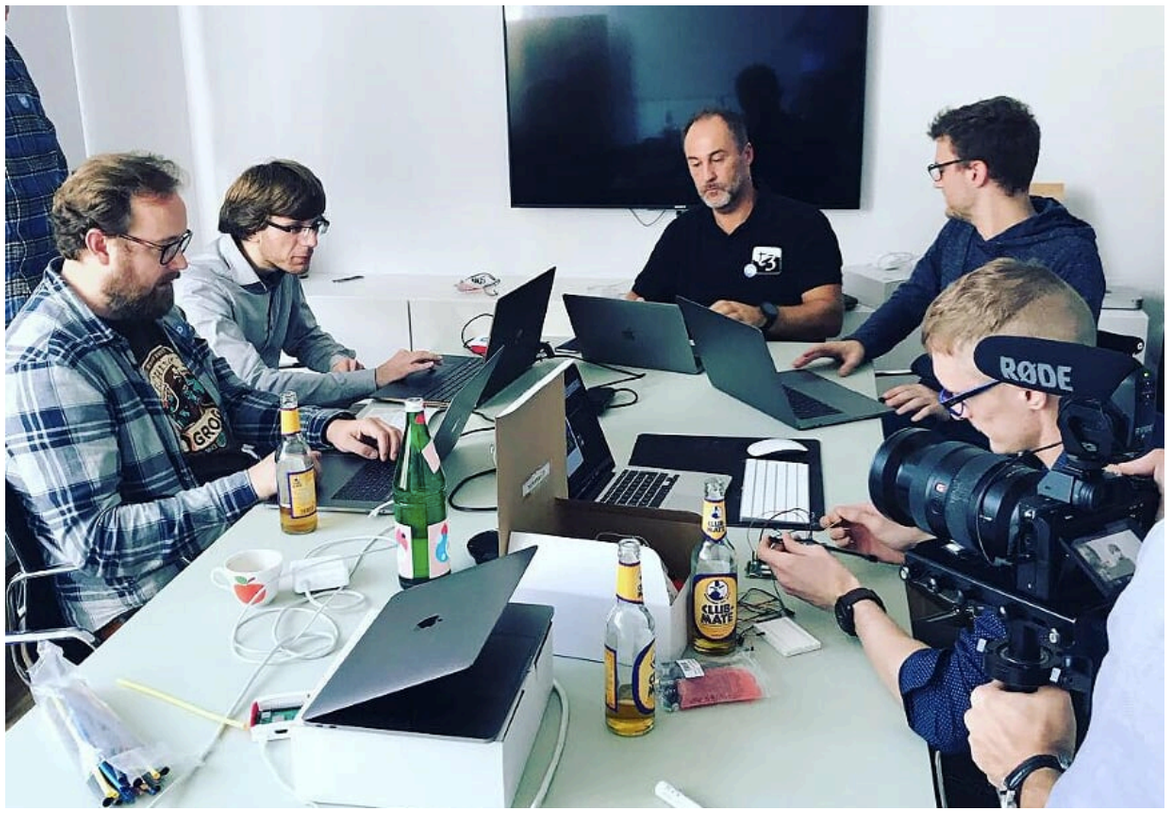
🇮🇹



IoT Workshop, ESUG
September 2018 - Cagliari, Italy

Pharo IoT in the world

🇩🇪



IoT Hackaton, Zweidenker GmbH
October 2018 - Cologne, Germany

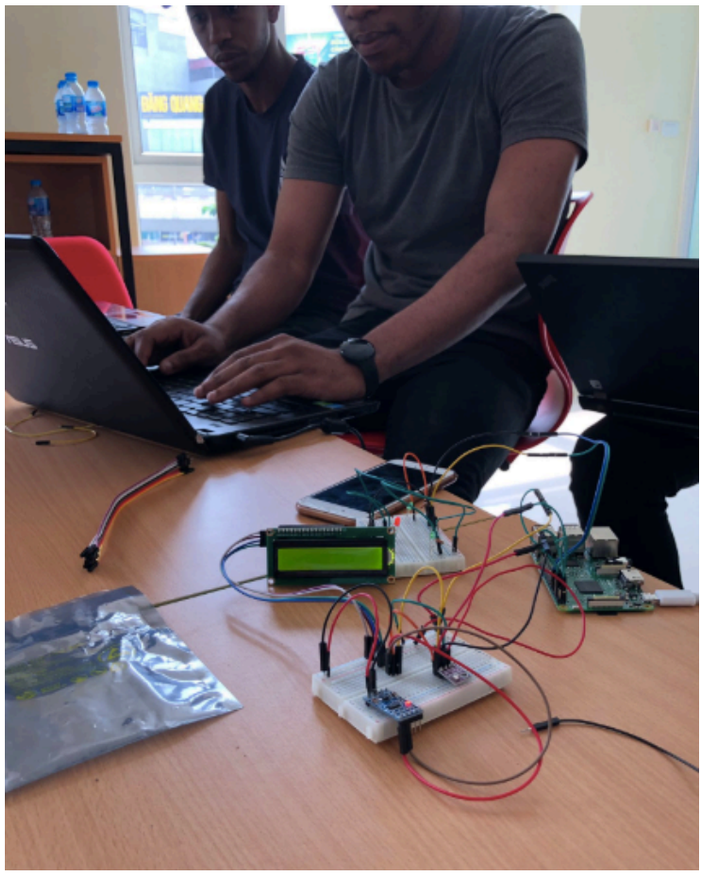
Pharo IoT in the world

🇻🇳

Live Programming IoT devices with PharoThings
January 2019 - Can Tho University, Vietnam

Pharo IoT in the world

🇻🇳



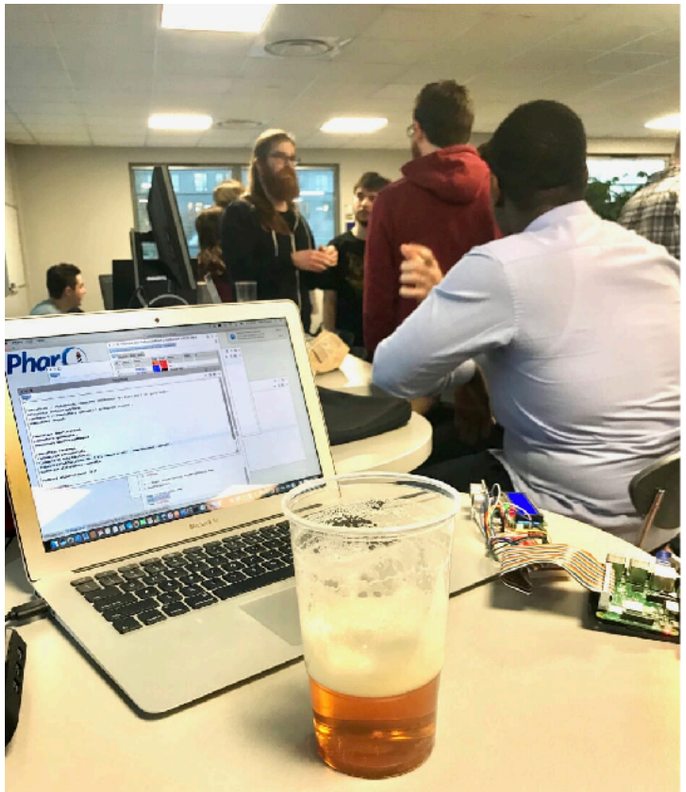
Live Programming IoT devices with PharoThings
May 2019 - International Francophone Institute, Vietnam

Pharo IoT in the world

ESUG Conference

European Smalltalk User Group

Sep 2018 Cagliari, Italy



INRIA

Pharo 10 Years

Nov 2018 Lille, France

USTH

University of Science and Technology of Hanoi

Jan 2019 Hanoi, Vietnam

Improvements

1. Easy installation (**zero-conf** scripts) less than 1 min
2. Zero-conf pages hosted in Github
3. Everything packed (VMs, 32/64 images, 1 click-run files)
4. Installing **from scratch** with Raspbian in less than 10 min
5. Code improvements and support to **new sensors**
6. **Pharo IoT Booklet** with many lessons
7. Welcome window with code examples
8. Pharo IoT website **pharoiot.org**
9. Using Continuous Integration - **CI** Travis

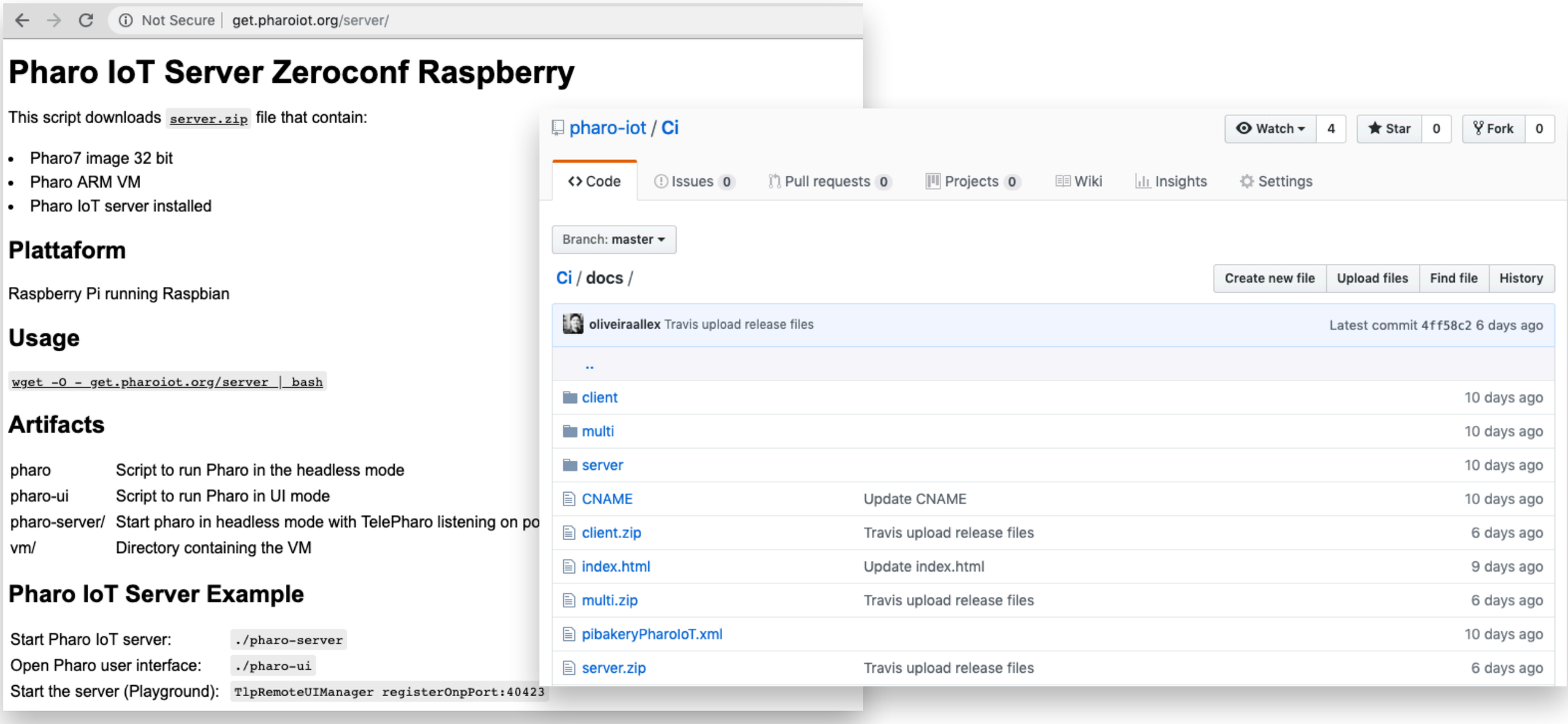
How to easy install (zero-conf)

1. Run the command to download and extract the files:
   - **wget -O - get.pharoiot.org/server | bash**
2. Run TelePharo server:
   - 1 click on pharo-server file or…
   - type in terminal: **./pharo-server**

Less than 1 minute!

get.pharoiot.org

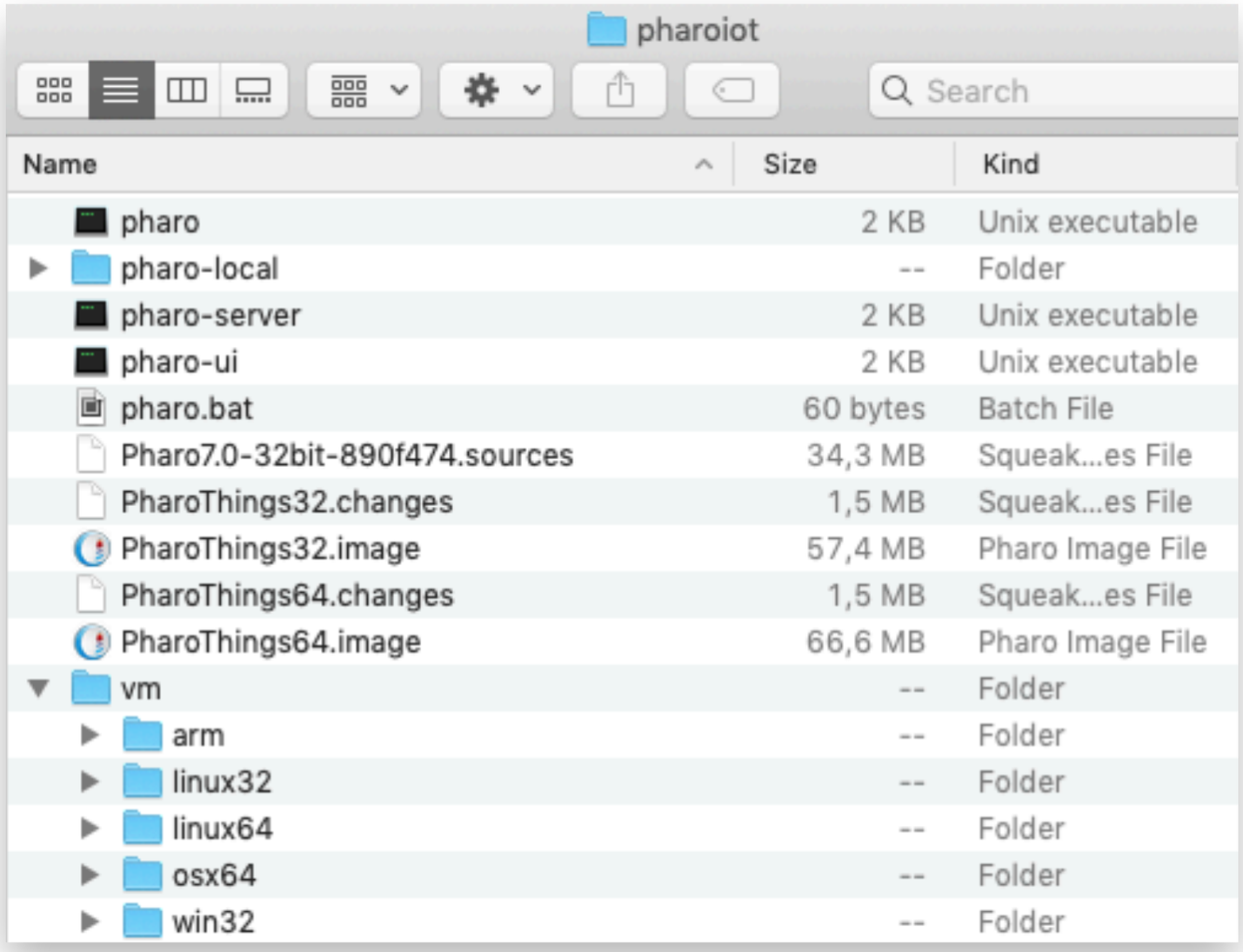
- We are using the Github Pages to host the zero-conf pages



https://github.com/pharo-iot/Ci/docs

Everything packed

+ Pharo Image 32/64
+ PharoThings loaded
+ ARM VM
+ Windows, Linux, Mac VMs



1-click run files

+ Pharo Image 32/64
+ PharoThings loaded
+ ARM VM
+ Windows, Linux, Mac VMs
+ **1 click run files**
  - pharo-ui
  - pharo-server
  - pharo
  - pharo.bat

pharo-server

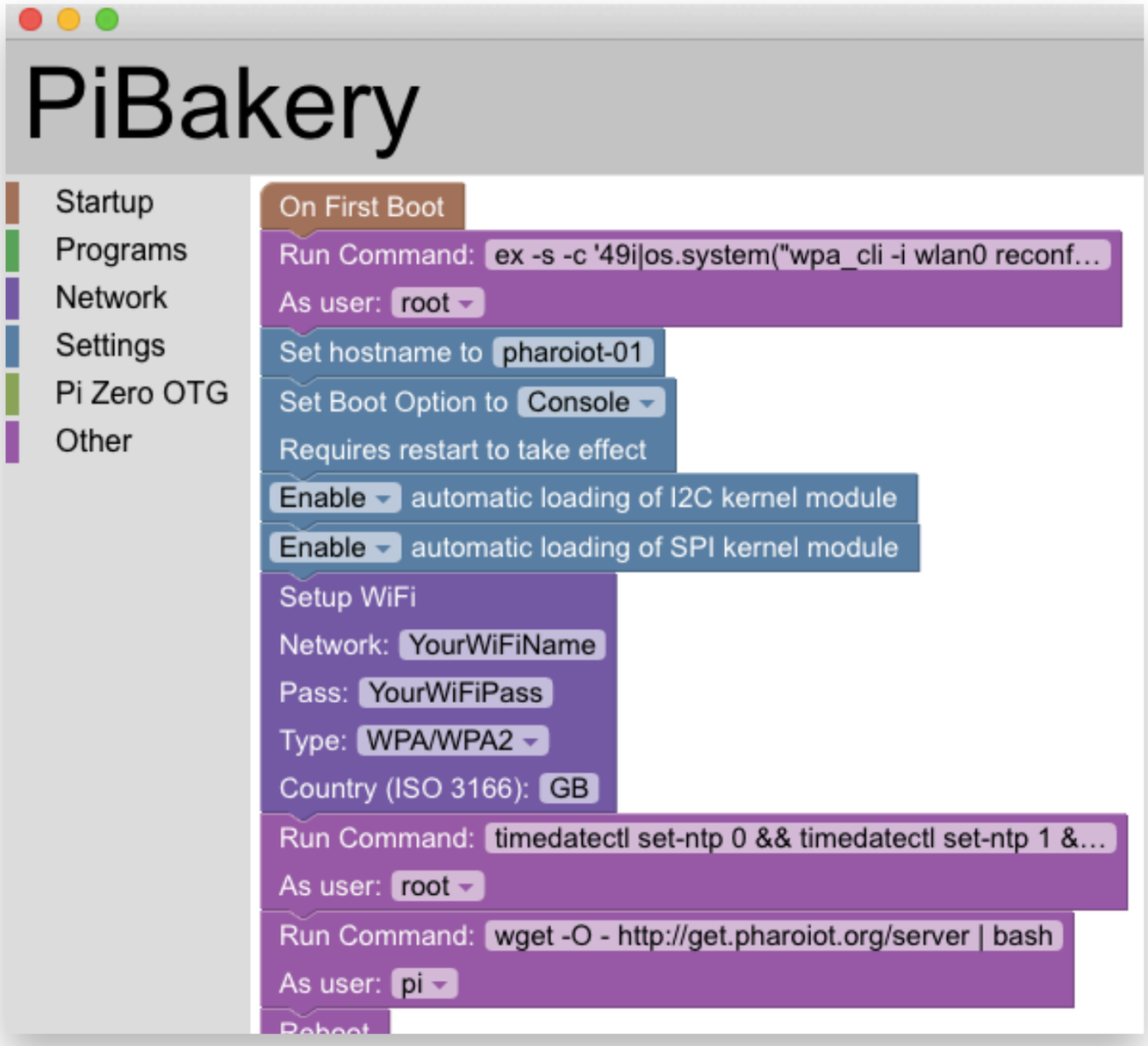
```
    OS="win";
elif [[ "{$TMP_OS}" = *mingw* ]]; then
    OS="win";
elif [[ "{$TMP_OS}" = *msys* ]]; then
    OS="win";
else
    echo "OS not identified. Try to run the correct VM on vm folder";
    exit 1;
fi
# 1.2 Getting the architecture
if [[ "{$TMP_ARCH}" = *arm* ]]; then
    ARCH="arm";
elif [[ "{$TMP_ARCH}" = *64* ]]; then
    ARCH="64";
else
    ARCH="32";
fi
# 1.3 Setting the correcty VM folder
if [[ "{$ARCH}" = *arm* ]]; then
    VM="arm";
elif [[ "{$OS}" = *windows* ]]; then
    VM="win32";
elif [[ "{$OS}" = *mac* ]]; then
    VM="osx32";
elif [[ "{$OS}" = *linux* ]] && [[ "{$ARCH}" = *64* ]]; then
    VM="linux64";
elif [[ "{$OS}" = *linux* ]] && [[ "{$ARCH}" = *32* ]]; then
    VM="linux32";
else
    echo "VM not identified. Try to run the correct VM on vm folder";
    exit 1;
fi

# Step 2 - Running the correcty VM and Pharo image
if [[ "{$VM}" = *arm* ]]; then
vm/$VM/pharo --headless PharoThings32.image remotePharo --startServerOnPort=40423
elif [[ "{$VM}" = *osx32* ]]; then
    # some magic to find out the real location of this script dealing with symlink
    DIR=`readlink "$0"` || DIR="$0";
    DIR=`dirname "$DIR"`;
    cd "$DIR"
    DIR=`pwd`
    cd - > /dev/null
    # disable parameter expansion to forward all arguments unprocessed to the VM
    set -f
    # run the VM and pass along all arguments as is
    "$DIR"/"vm/$VM/Pharo.app/Contents/MacOS/Pharo" --headless "$DIR"/PharoThings32
elif [[ "{$VM}" = *linux64* ]]; then
    vm/$VM/pharo --headless PharoThings64.image remotePharo --startServerOnPort=40
elif [[ "{$VM}" = *linux32* ]]; then
    vm/$VM/pharo --headless PharoThings32.image remotePharo --startServerOnPort=40
```

Installing from scratch

+ Installing Raspbian
+ Download Pharo IoT
+ Set Hostname
+ Enable I2C and SPI
+ Connect on WiFi
+ Start server every boot

Keyboard,mouse or
monitor not required

Less than 10 minutes!



Board modelling improvements

Inspector on a RpiBoard3B

a RpiBoard3B

P1 | Devices | Raw | Meta

Value	Function	Name	Pin#	Pin#	Name	Function	Value
		3.3v	1	2	5v		
out	SDA (I2C)	gpio2	3	4	5v		
	SCL (I2C)	gpio3	5	6	Ground		
	Clock	gpio4	7	8	gpio14	TXD (Serial	out
		Ground	9	10	gpio15	RXD (Serial	
		gpio17	11	12	gpio18	PWM	
		gpio27	13	14	Ground		
		gpio22	15	16	gpio23		
		3.3v	17	18	gpio24		
in	MOSI (SPI)	gpio10	19	20	Ground		
	MISO (SPI)	gpio9	21	22	gpio25		
	SCLK (SPI)	gpio11	23	24	gpio8	CE (SPI)	
		Ground	25	26	gpio7	CE (SPI)	
	SDA (I2C)	gpio0	27	28	gpio1	SCL (I2C)	
		gpio5	29	30	Ground		
		gpio6	31	32	gpio12	PWM	
	PWM	gpio13	33	34	Ground		
	MISO (SPI)	gpio19	35	36			
		gpio26	37	38			
		Ground	39	40			

```
led1 := gpio2.
led1 beDigitalOutput.
led1 value:1.

led2 := gpio14.
led2 beDigitalOutput.
led2 value:1.

button1 := gpio10.
button1 beDigitalInput.
button1 value
```

PotLCDHD44780Gpio>>configurePeripherals

BaselineOfPharoThi
PharoThings-Device
PharoThings-Device
PharoThings-Device
thin

PotLCDHD44780
PotLCDHD44780Gpio
PotLCD1602Device !
PotLCDHD44780I2C
Filter...

instance side
accessing
commands
controlling
initialization

configurePeripherals
connect
initializeRegisters
isConfigured
isConnected

All Packages | Scoped View | Flat | Hier. | Inst. side | Class side | Methods | Vars | Class ref

Comment | PotLCDHD44780G | configurePeripher | Inst. side method

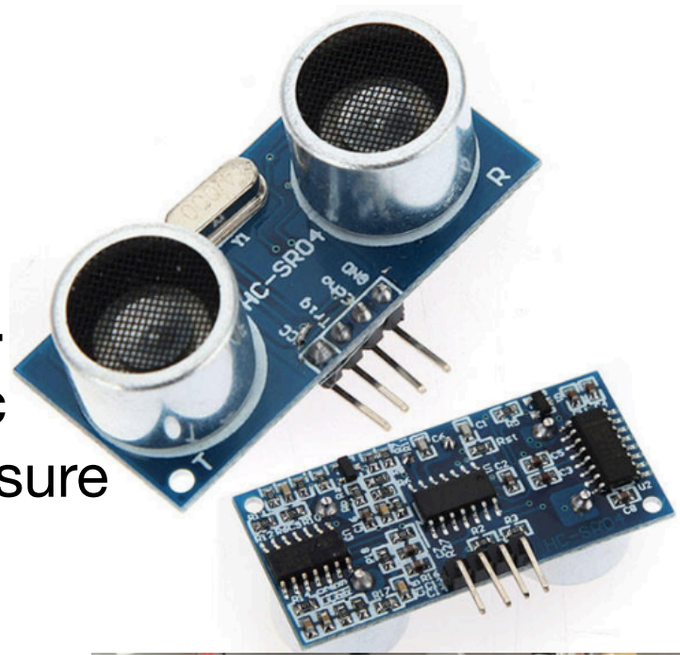
```
configurePeripherals
   "Pin mapping:
   ------------------------------------------------------------
   VSS|VDD|V0 |RS |RW | E |D0 |D1 |D2 |D3 |D4 |D5 |D6 |D7 | A | K
      |   |   |27 |   |22 |   |   |   |   |25 |24 |23 |18 |   |12
                                [8 BIT]           4 BIT
   ------------------------------------------------------------"
   modePin := 13 gpioHeader.   "RS 1 character or 0 lcd commands"
   clockPin := 15 gpioHeader.  "EN clock enable"
   dataPins := #(12 16 18 22) collect: [ :id | id gpioHeader ]. "D7 D6 D5 D4 [D3
bit 4bit or [8bit]"
   backlightPin := 32 gpioHeader  "BL backlight PWM"
```

Board modelling improvements

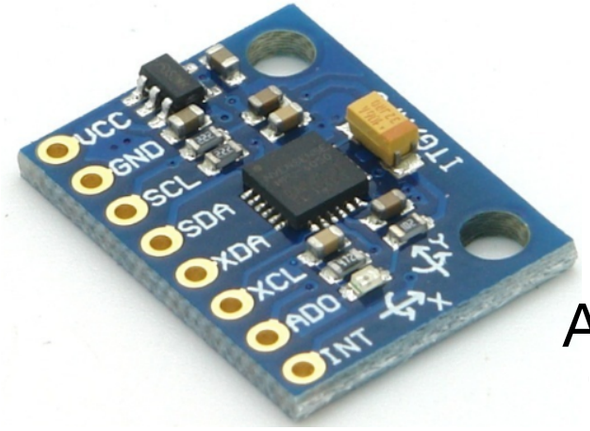
1. Removed WiringPi numbers reference and start adopt BCM
2. Enable basic GPIO behaviour to all GPIOs
   - before were 14, now we can use 28 gpios
3. Add configure peripherals methods
4. Create the GPIO instance using *header number* or *GPIO number*

Support to new sensors

HD44780 I2C
LCD display

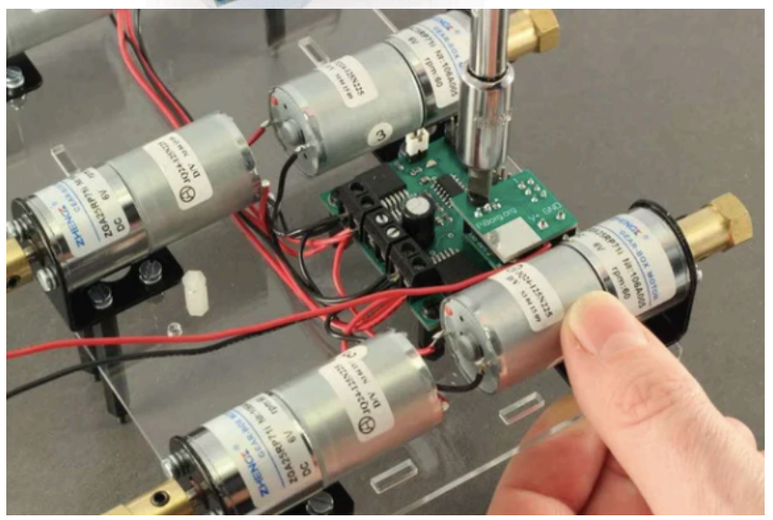


HC-SR04
Ultrasonic
distance measure



MPU6050
Gyroscope
Accelerometer
Temperature

PicoBorg
I2C motors



https://github.com/oliveiraallex/PotHD44780Controller (integrated in oficial repository)
https://github.com/oliveiraallex/PotHCSR04 (integrated in oficial repository)
https://github.com/oliveiraallex/PicoBorgReverseMotors
https://github.com/oliveiraallex/PotMPU6050Device

PharoThings Booklet

https://github.com/SquareBracketAssociates/Booklet-APharoThingsTutorial

PharoThings Booklet

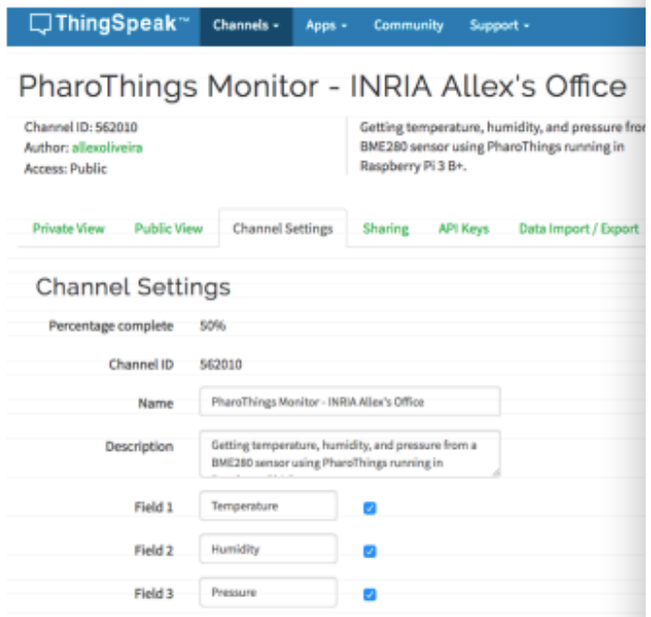


Figure 12-1 ThingSpeak Channel Configuration.

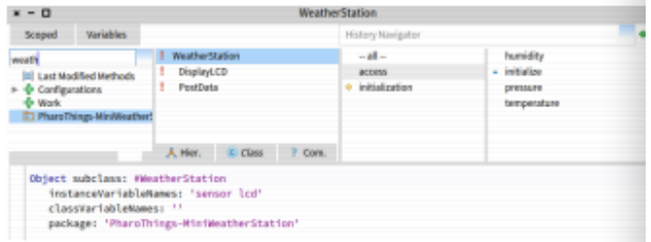


Figure 12-2 Mini Weather Station code.

play this information on the LCD and the second will send the data to cloud. Your final code will seem like the Picture 12-2.

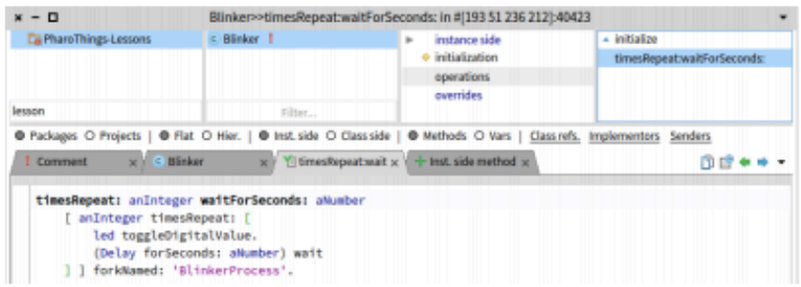


Figure 4-7 Creating an operation method.

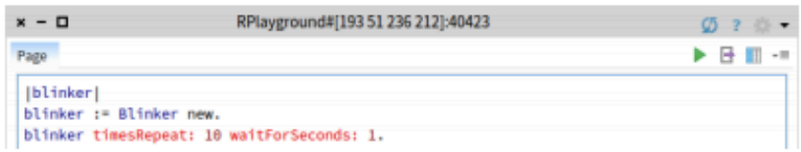


Figure 4-8 Remote playground.

```
|blinker|
blinker := Blinker new.
blinker timesRepeat: 10 waitForSeconds: 1.
```

Run this code, as shown in Figure 4-8 and... `cool`! Now your LED is blinking! And the better, you did this using object-oriented programming!

You do not need to change your code every time you wanna change these parameters. Just change the messages you send to the object and it will behave as you want.

4.9 Save your work

Don't forget to save your work remotely. To do this, run this command on your local playground:

```
remotePharo saveImage.
```

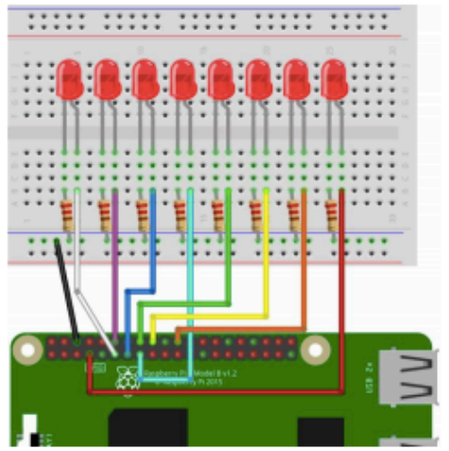



Figure 5-1 Schema connection 8 LEDs.

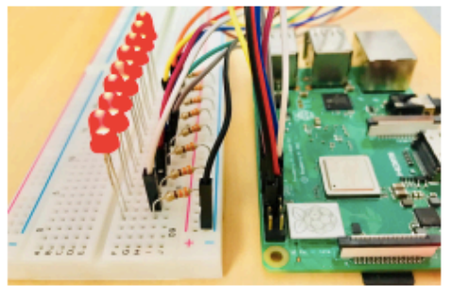
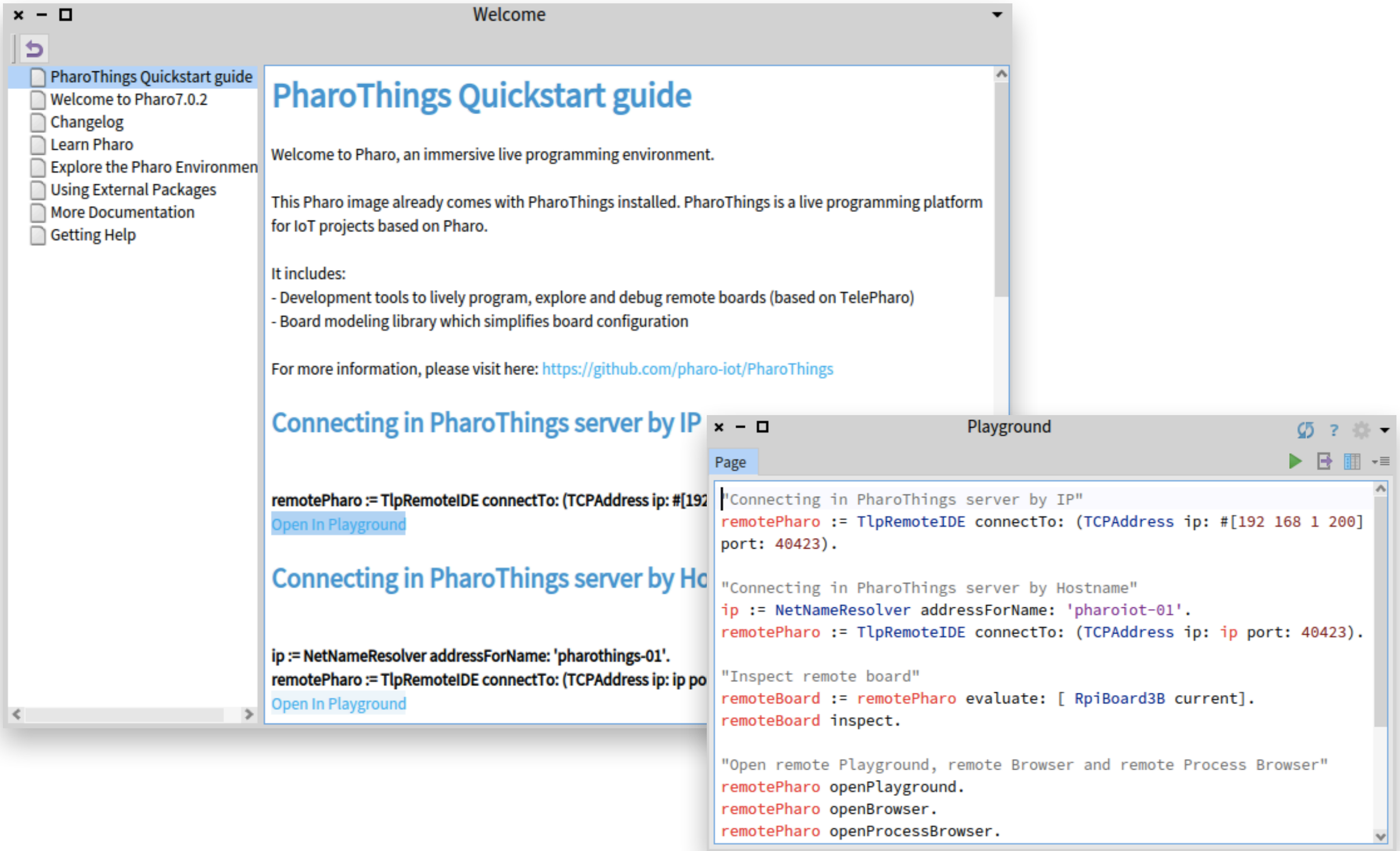


Figure 5-2 Physical connection 8 LEDs.

- Connect the Ground PIN from Raspberry in the breadboard blue rail (-).
- Then connect the 8 resistors from the blue rail (-) to a column on the breadboard, as shown below;
- Now push the LED legs into the breadboard, with the long leg (with the kink) on the right;
- And insert the jumper wires connecting the right column of each LED to GPIO from 0 to 7, as shown in the Picture 5-1.

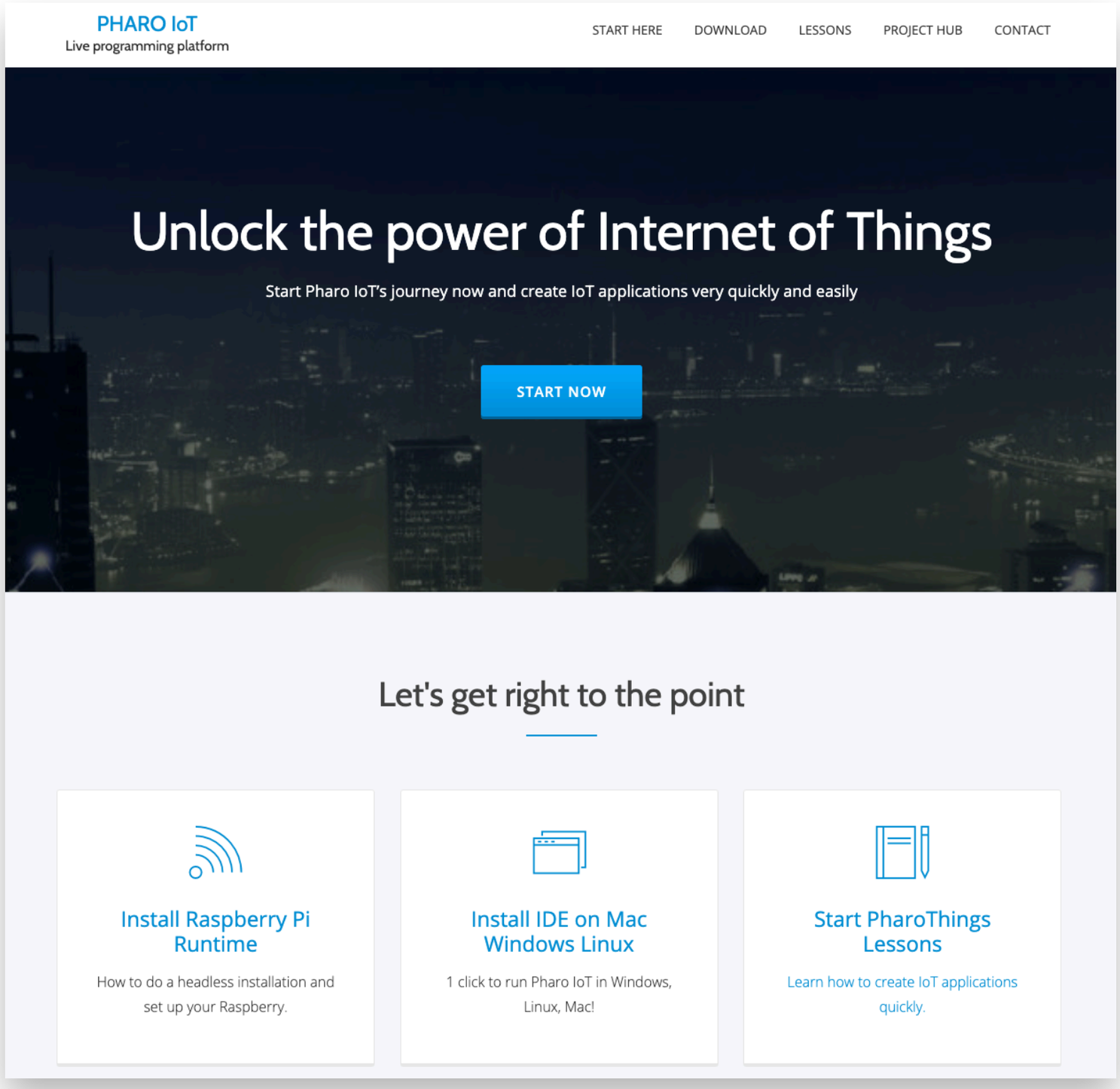
https://github.com/SquareBracketAssociates/Booklet-APharoThingsTutorial

Welcome Window PharoThings



pharoiot.org

Everything in the same place, to facilitate the journey of the new user.



pharoiot.org

Location

TO SAVE EXPORT TO SHARE TO EDIT INSIGHTS

All users
100.00% Users

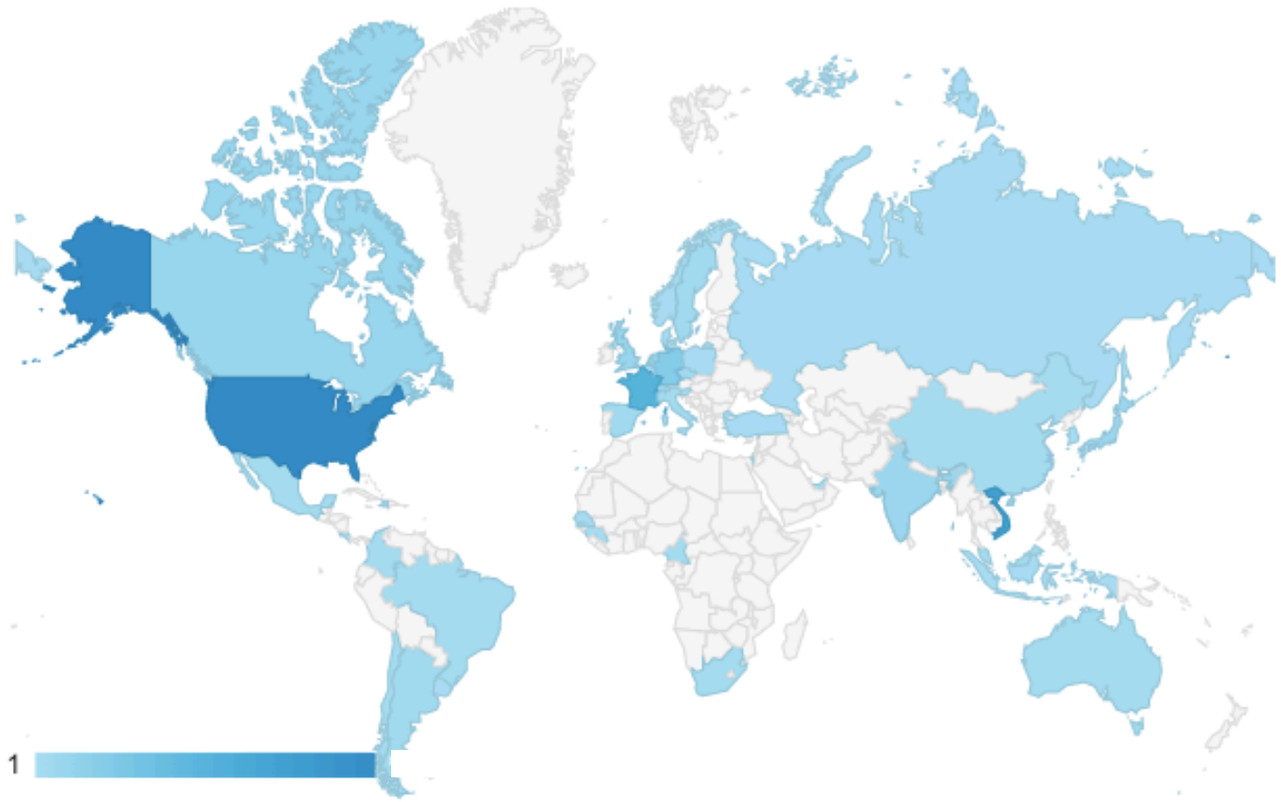
+ Add Segment

Mar 13, 2019 - Aug 20, 2019

Regional Explorer Coverage

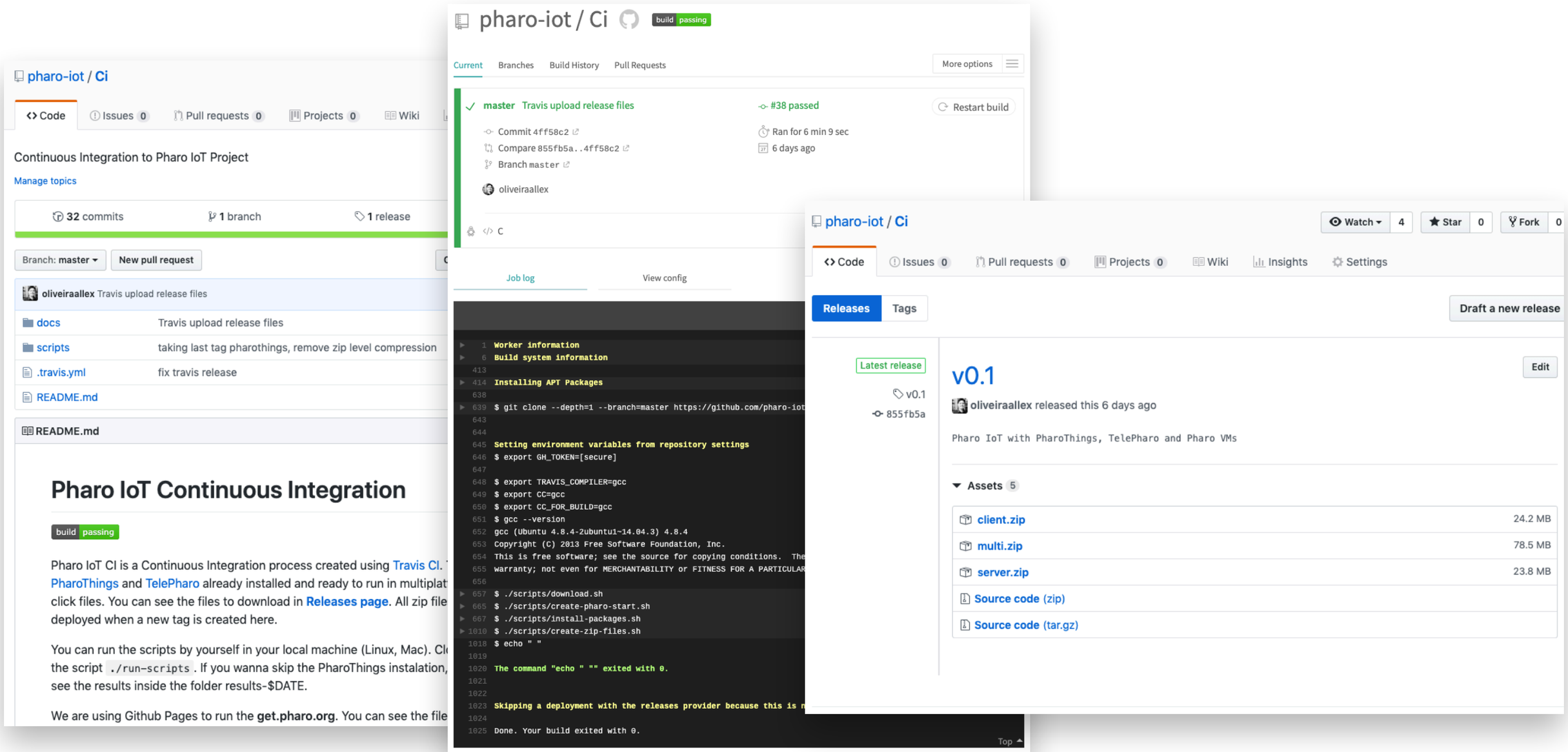
Summary Data Using the Website Ecommerce

Users



Continuous Integration

Travis CI on Pharo IoT



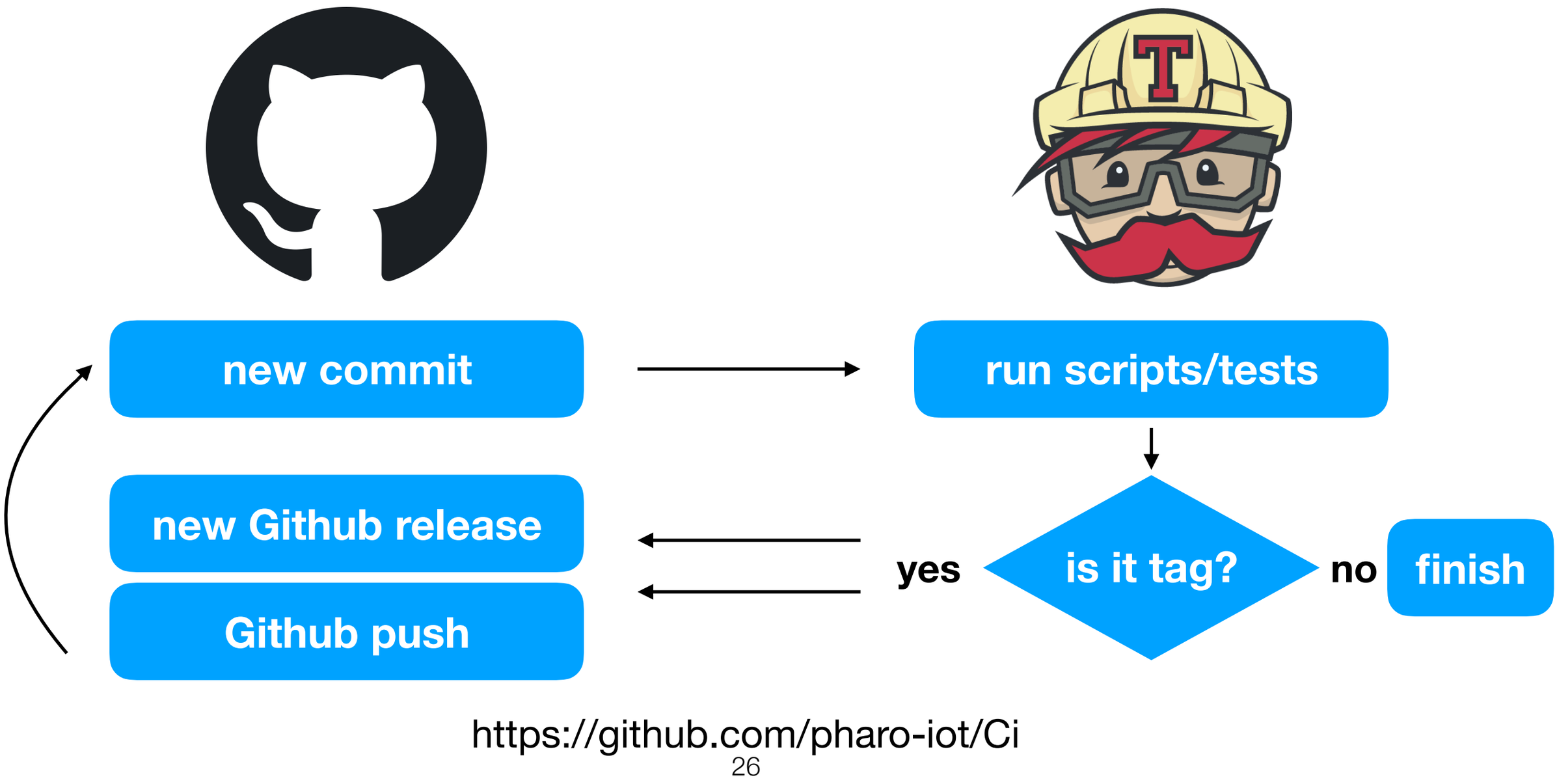
https://github.com/pharo-iot/Ci

Continuous Integration

Travis CI on Pharo IoT

github.com/pharo-iot/Ci/**.travis.yml**

travis-ci.org/**pharo-iot/Ci**



https://github.com/pharo-iot/Ci

Continuous Integration

https://github.com/pharo-iot/Ci

- Load PharoThings on image 32/64
- Download last VMs
- Create the 1 click-run files
- Zip everything
- Deploy
- Create PDF Booklet

2 - Collaborative work

- Denis Kudriashov (PharoThings improvements)

  https://github.com/dionisiydk

- ZweiDenker (Minimal PharoThings image)

  https://github.com/noha/pharo-minimal

- Bela IO, Jack Armitage (Pharo IoT on musical context)

  https://bela.io/

- Serge Stinkwich (Pharo IoT lessons, Pharo IoT Booklet)

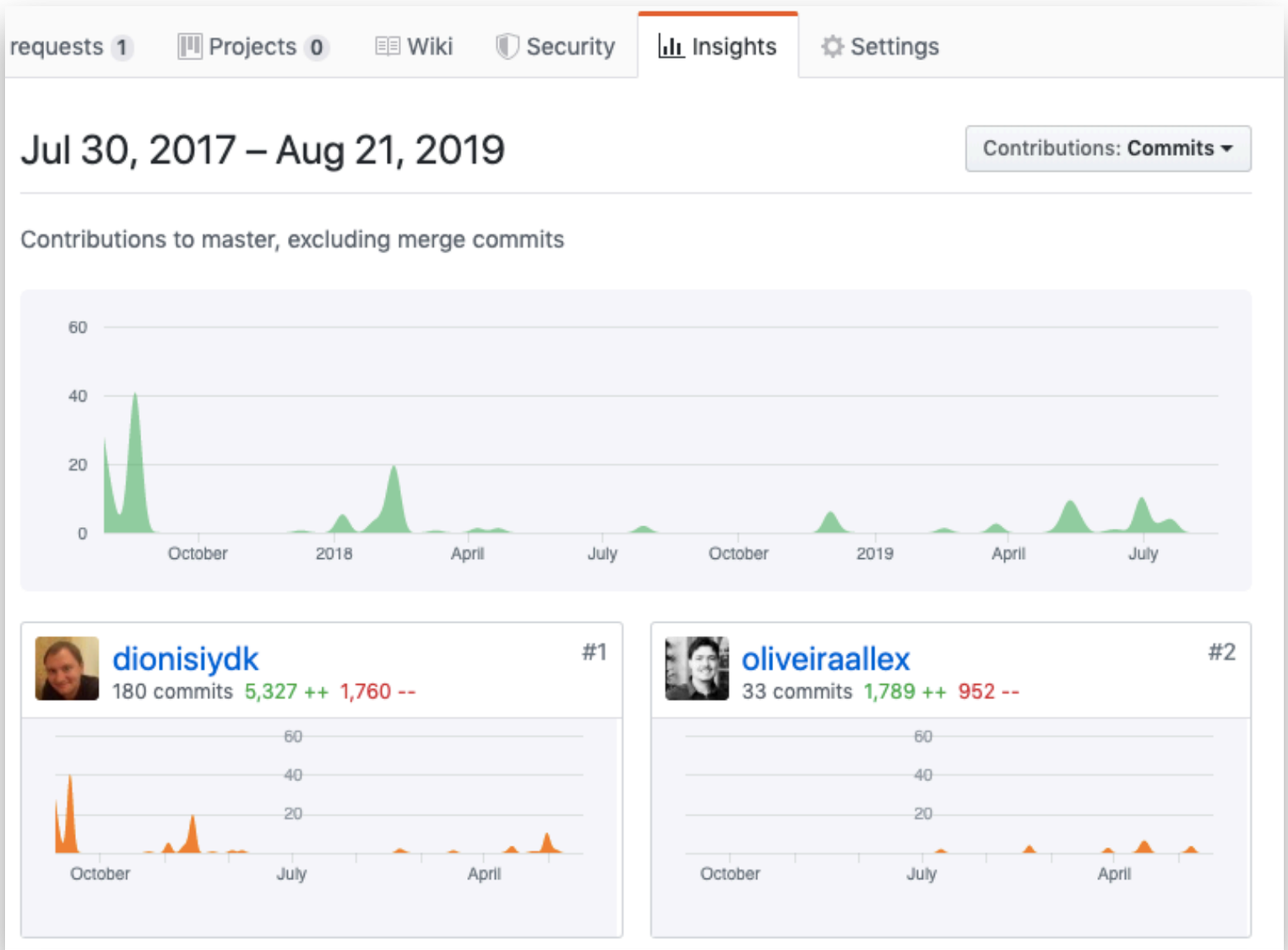
  https://github.com/SergeStinckwich

- Do Hoang, Vietnam (Pharo IoT Booklet - Arduino Chapter)

  https://github.com/huyhoang8398

PharoThings improvements

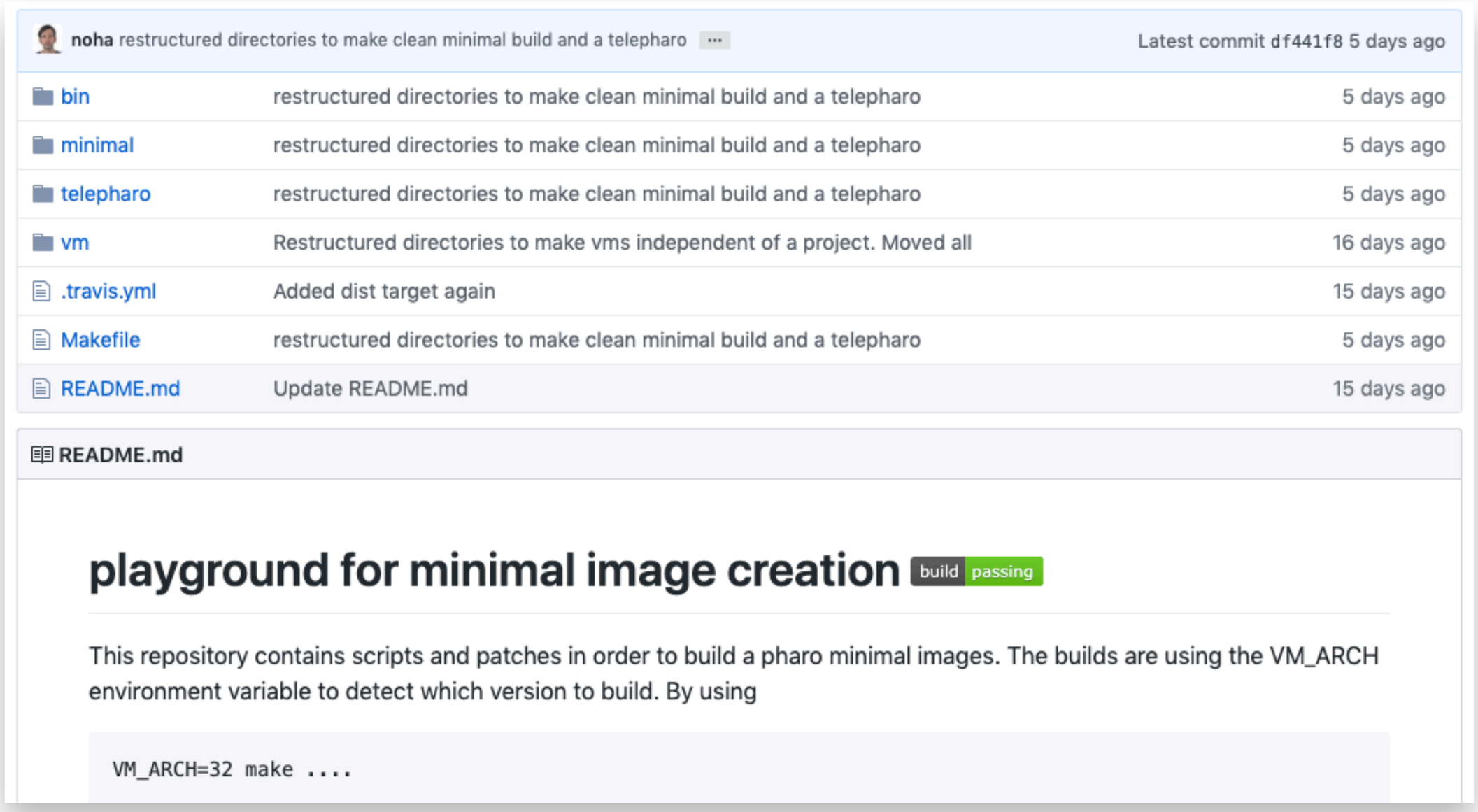
- Denis Kudriashov (Russia)



https://github.com/dionisiydk

Minimal PharoThings image

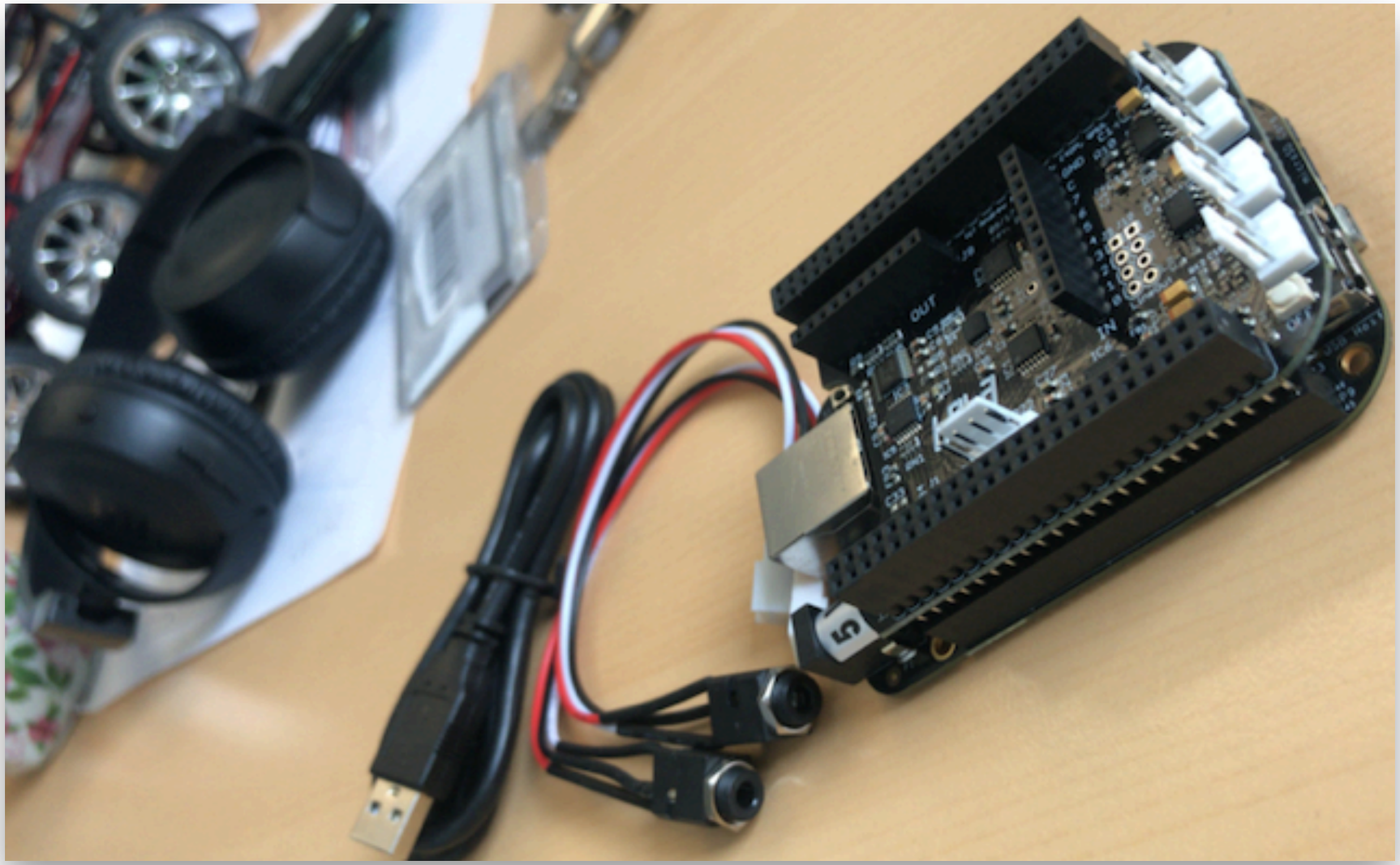
- Norbert Hartl (Germany)



https://github.com/noha/pharo-minimal

Bela IO - Beaglebone

- Jack Armitage (UK)



https://bela.io/

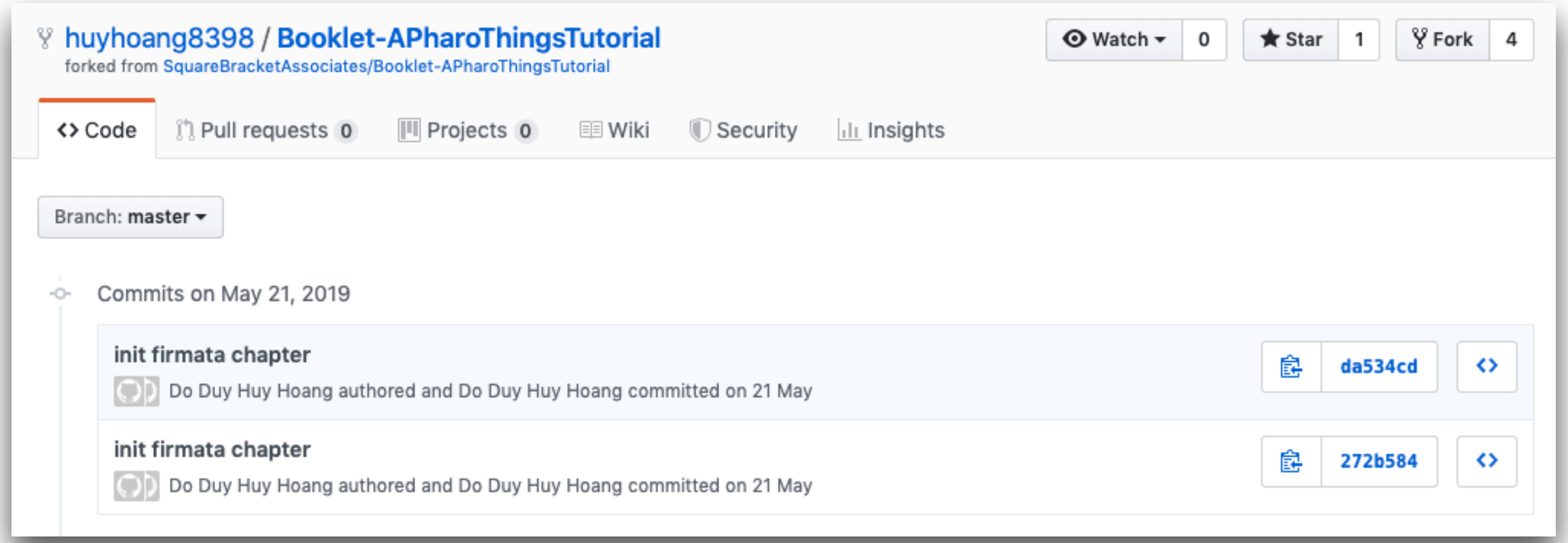
Pharo IoT workshops and booklet collaborations

- Serge Stinckwich (France)

https://twitter.com/sergestinckwich

Pharo IoT Booklet
Arduino Chapter

- Do Hoang (Vietnam)



https://github.com/huyhoang8398/Booklet-APharoThingsTutorial

3 - Projects using Pharo IoT

- Coffee Machine IoT
- Autonomous Robot
- Door opener

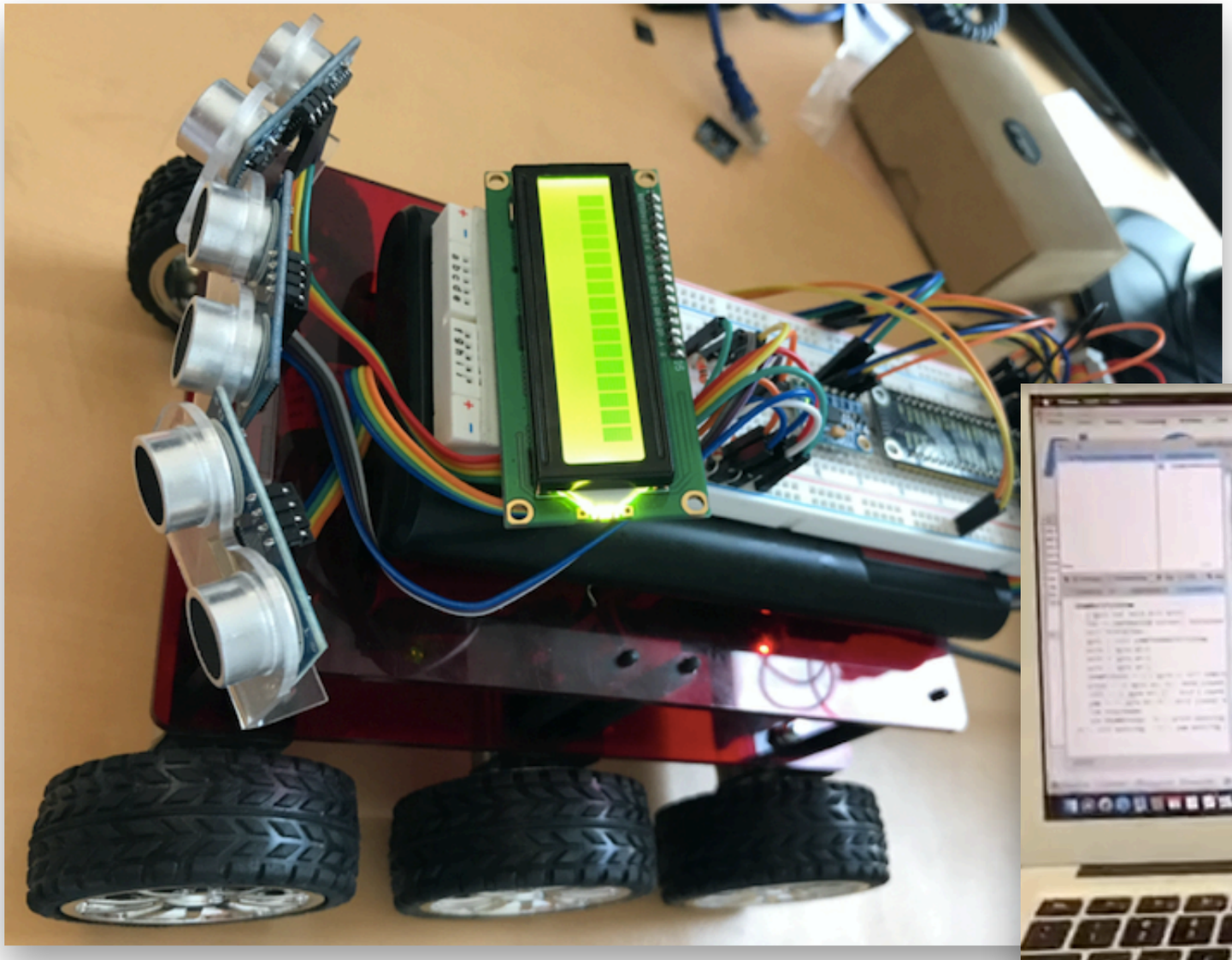
Coffee Machine IoT

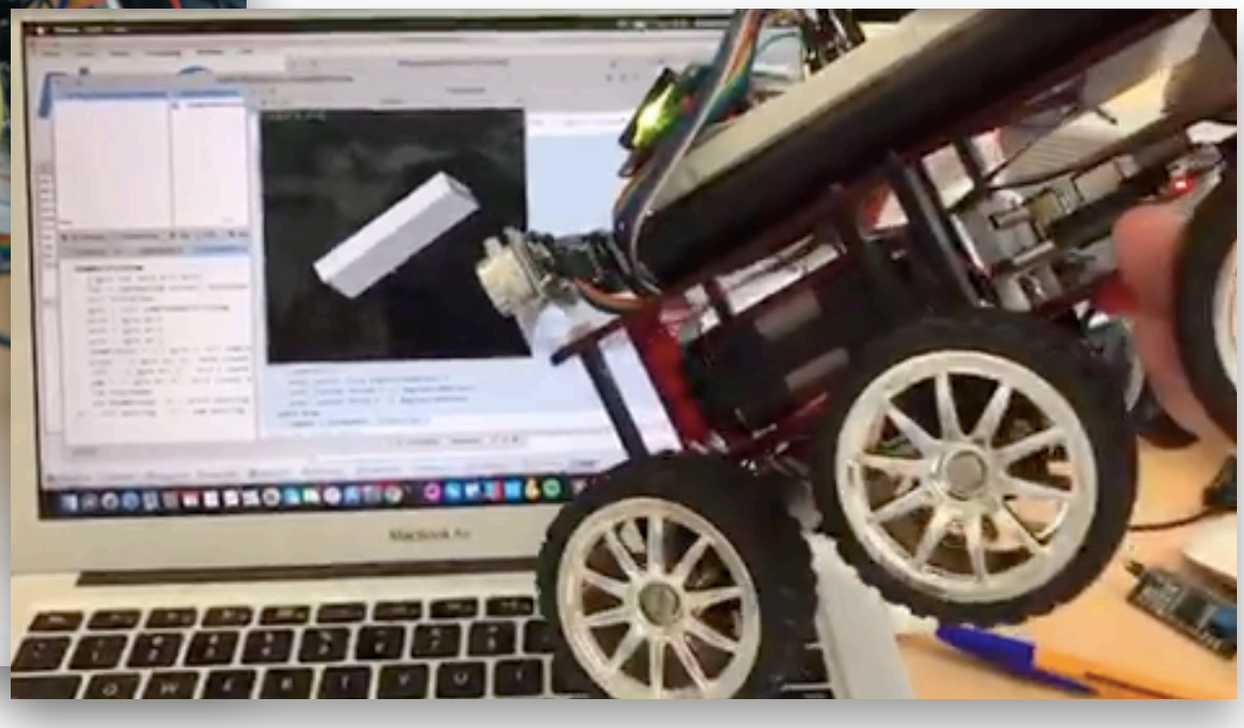
- Allex Oliveira

https://github.com/oliveiraallex/CoffeeMachine

Autonomous robot

- Allex Oliveira and Steven Costiou

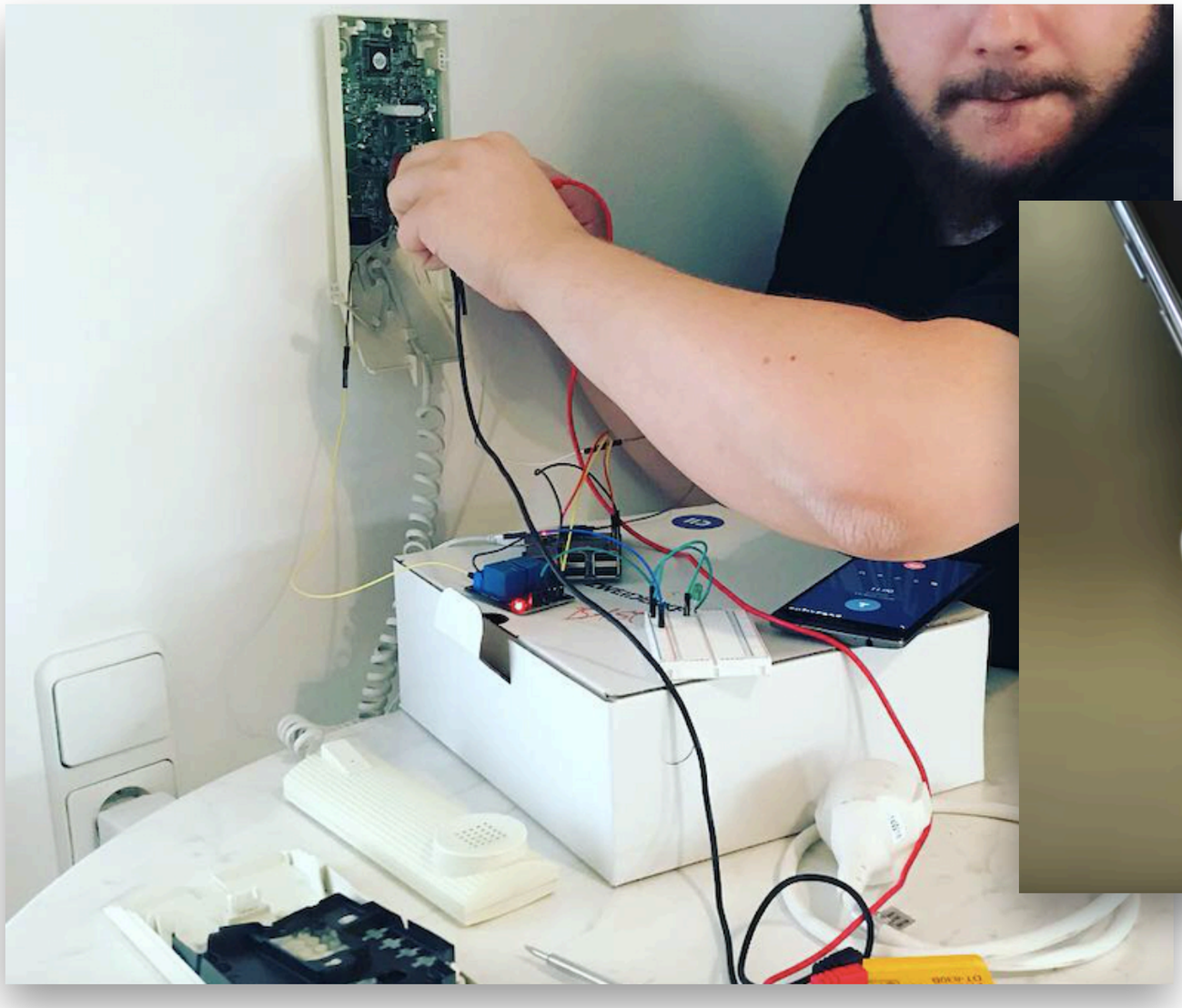


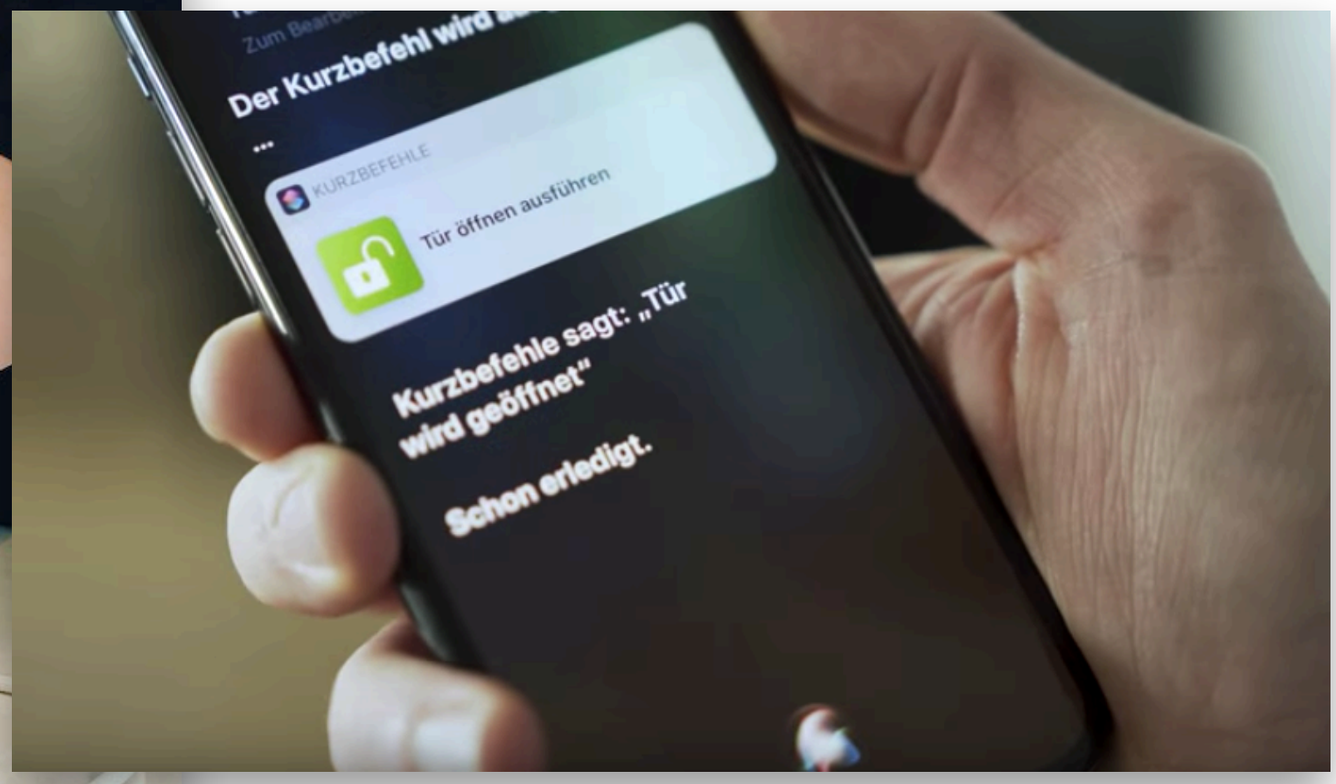


https://www.youtube.com/watch?v=1j5WSCkIiKk

Door opener

- ZweiDenker workshop

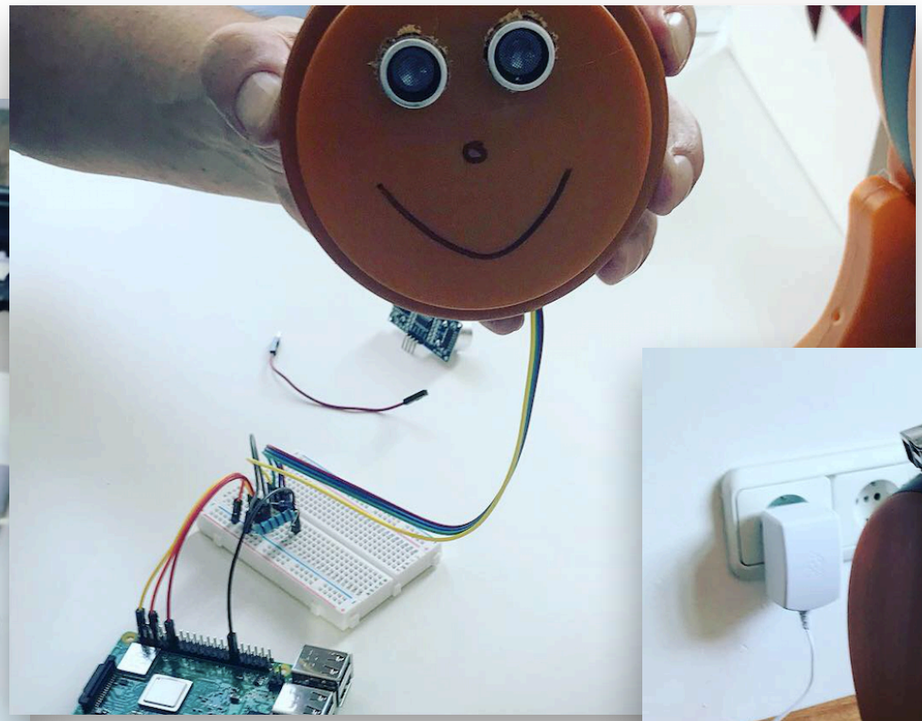




https://www.youtube.com/watch?v=dIl9FAatKyw

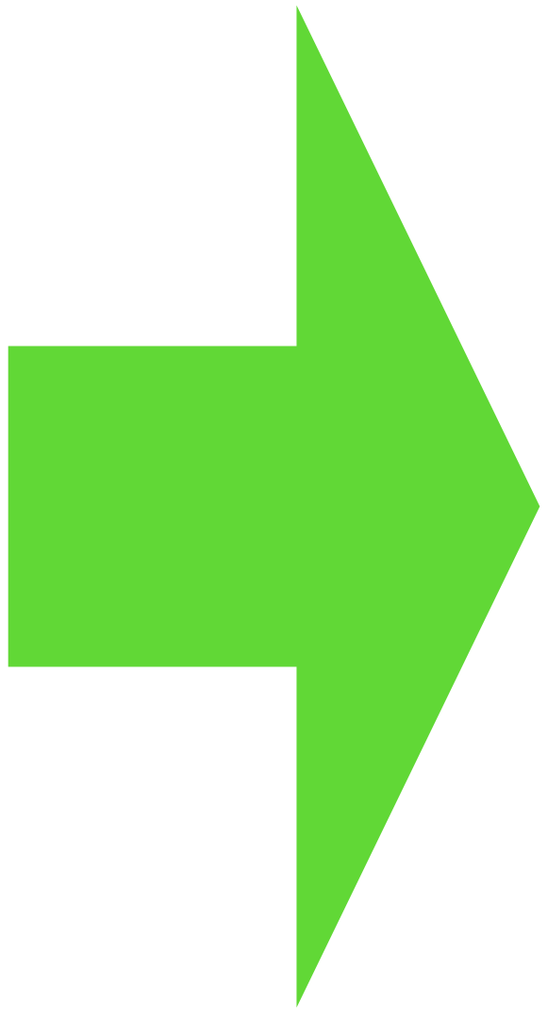
Wine measurer

- ZweiDenker workshop



https://www.youtube.com/watch?v=dIl9FAatKyw

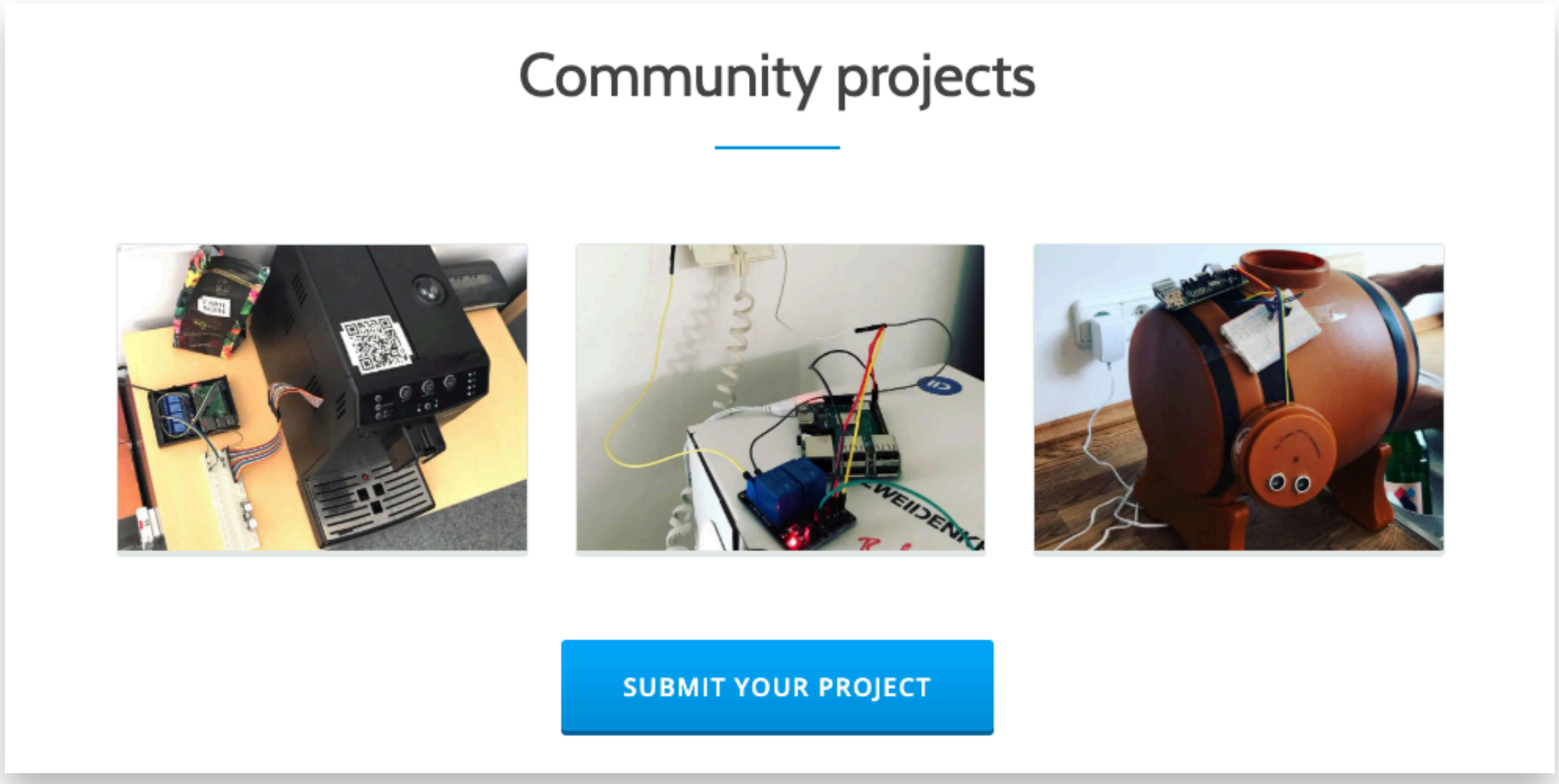
4 - Future



Booklet
Creating drivers and application with Pharo IoT

pharoiot.org

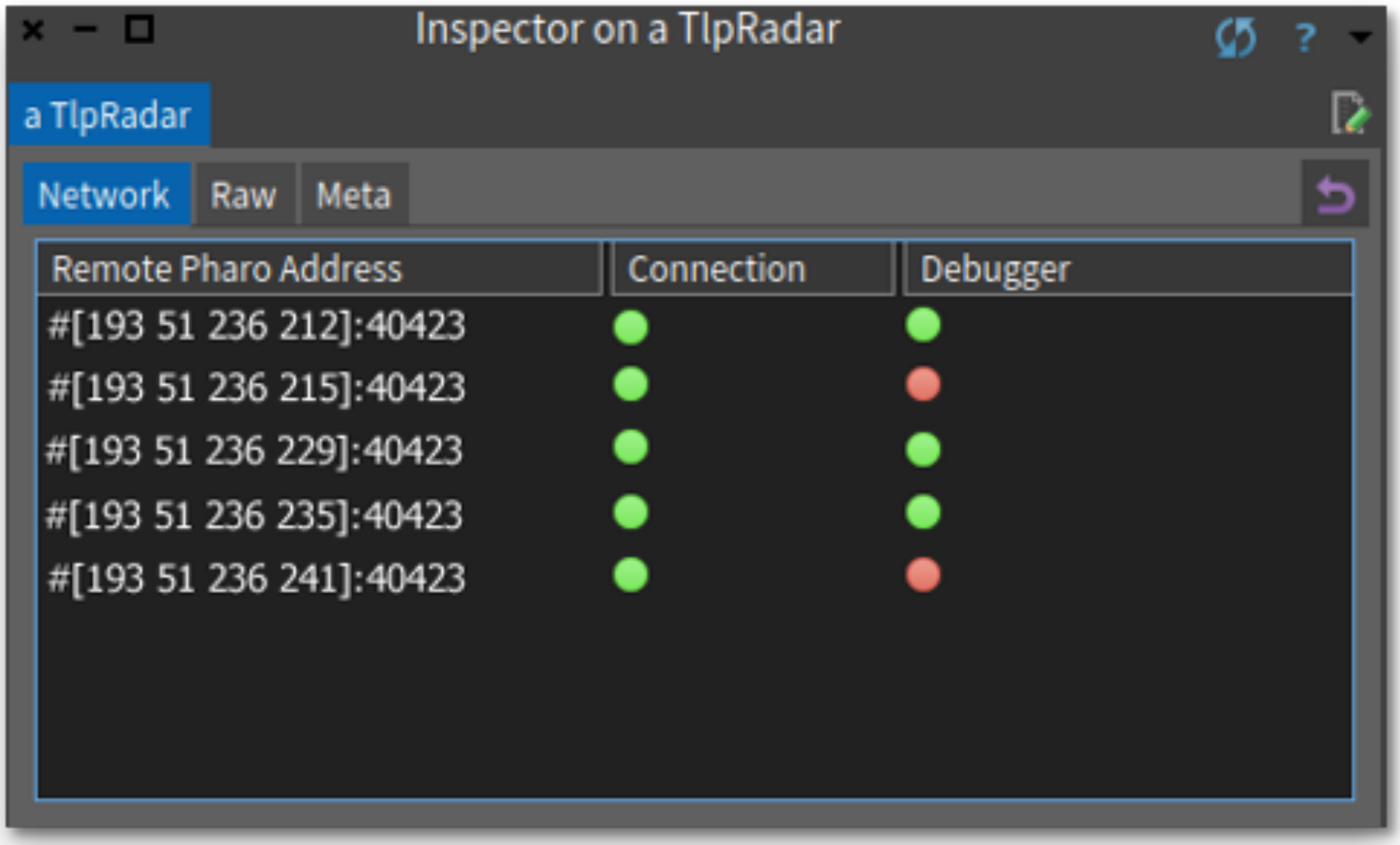
- Share your IoT projects using Pharo with the community!



Tele Radar

Automatic detection of running images in network

(TeleRadar using SSDP protocol)



PharoThings Booklet inside Pharo

Welcome

- PharoThings Booklet
- Welcome to Pharo7.0.2
- Changelog
- Learn Pharo
- Explore the Pharo Environment
- Using External Packages
- More Documentation
- Getting Help

PharoThings Booklet

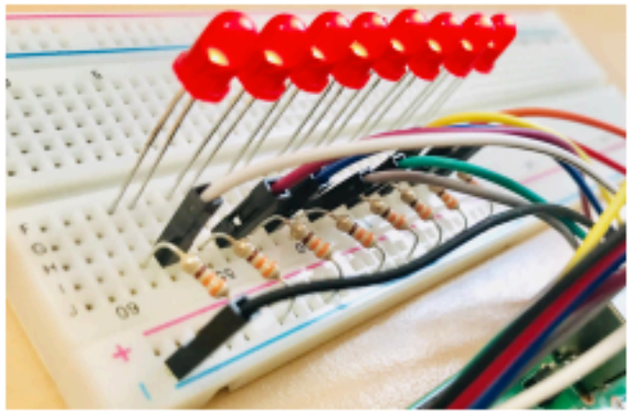


Figure 5-4 LEDs turn On.

5.4 Adding features

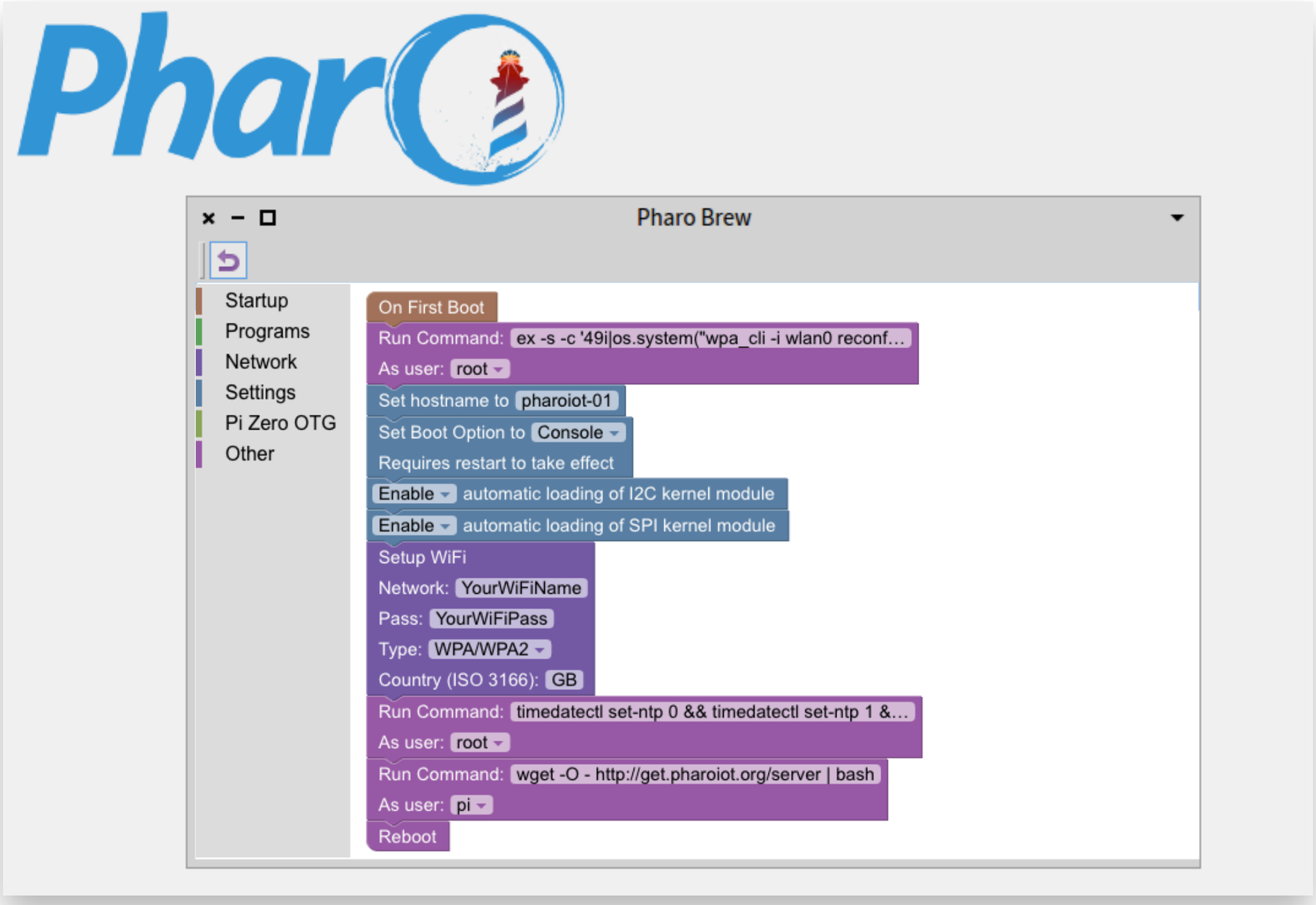
Every time you run this code, the LEDs toggles the state, from Off to On or vice versa. Let's reduce the delay time and add the `timesRepeat:` method, as we did in the last lesson, to repeat the alternation as many times as we want:

```
[ 2 timesRepeat: [
    gpioArray do: [ :item | item toggleDigitalValue. (Delay
    forSeconds: 0.1) wait ].
] ] forkNamed: 'FlowingProcess'.
```

Execute this code and... cool! Now your LEDs are flowing On and Off!

Tool to “brew” SD Cards

- “brew" a new SD Card to inside Pharo (like PiBakery)



With Pharo IoT you can

- Dynamically update your running board
- Interact remotely with pins and boards
- Modify the system while it is running (create new board, change code)
- Make your changes persistent

get.pharoiot.org

THANKS!

Any questions?

allex.oliveira@msn.com

Presentation Information

This slides was presented at ESUG 2019, Cologne, Germany
https://esug.github.io/2019-Conference/conf2019.html

- Title: Pharo IoT - Present and Future
- Presenters:
  Marcus Denker - marcus.denker@inria.fr
  Norbert Hartl - norbert@2denker.de
  Allex Oliveira - linkedin.com/in/allex-oliveira

INRIA
https://www.inria.fr/
RMOD TEAM
https://rmod.inria.fr/web
PHARO PROJECT
https://github.com/pharo-project/pharo
PHAROTHINGS PROJECT
https://github.com/pharo-iot/PharoThings
PHARO IoT
http://get.pharoiot.org