

PHARO IOT

Using Pharo to playing with GPIOs and sensors on IoT devices remotely

HELLO!

Marcus Denker

CR1 Inria (tenured researcher)
marcus.denker@inria.fr

Alex Oliveira

IoT Engineer
alex.oliveira@msn.com

SUMMARY

1 – OVERVIEW

2 – INSTALLATION

3 – PLAYING

4 – PERSONAL WEATHER STATION

5 – THE FUTURE

1 - OVERVIEW

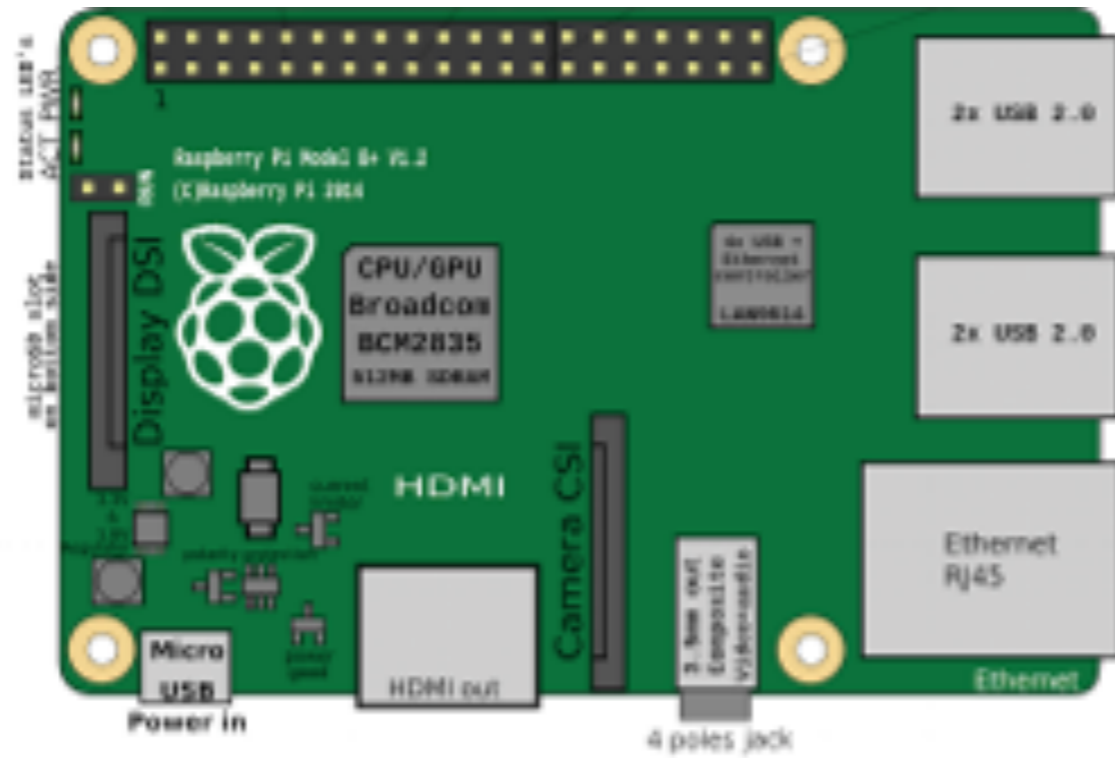
- Created by **Rmod Team**, a research team from **INRIA** (France)
- Written by Denis Kudriashov in 2016/17
dionisiydk@gmail.com
- In 2018, Alex Oliveira joined the Rmod Team to continue the project

1 - OVERVIEW

What is PharoThings?

- A Pharo image running on IoT device (ARM VM)
- A Pharo image controlling remote IoT device
- A Pharo library running on Raspberry Pi to control GPIOs through an object board model (using Wiring Pi)
- A Pharo library to control Arduino Devices (using Firmata)
- An advanced board model inspector
- Remote IDE (TelePharo)

1 - OVERVIEW



1 - OVERVIEW

- GPIO pins (general-purpose input/output), I2C, SPI
- Board modelling library which simplifies board configuration
- Raspberry Pi only (for now)

1 - OVERVIEW

PharoThings Library

The screenshot displays the Pharo IDE interface for the PotBoard class. The window title is "PotBoard". The interface is divided into several panes:

- Left Pane (pharothin):** A tree view showing the library structure. The "PharoThings-Hardware-Core" package is selected.
- Middle Pane:** A list of classes under the selected package, including PotBoard, PotBoardConnector, PotBoardDriver, PotBoardDriverDummy, PotBoardPin, PotBoardIOPin, PotGPIOPin, PotClockGPIOPin, PotPwmGPIOPin, PotI2CPin, and PotSCLPin. The "PotBoard" class is selected.
- Right Pane (History Navigator):** A list of methods for the selected class, including -- all --, accessing, controlling, initialization, io protocols, operations, *PharoThings-GTTools, *PharoThings-RemoteToolsSe, addNewConnector:with:, bindingsForPinsOf:, cacheRemoteState, connect, connectDriver, connectToI2CDevice:, connectWithExistingState, connectorNamed:, connectors, connectors:, createSeamlessReference, and devices.
- Bottom Pane:** A text area showing the object subclass and instance/class variable names for the selected class.

Object subclass: #PotBoard
instanceVariableNames: 'connectors driver devices'
classVariableNames: ''
package: 'PharoThings-Hardware-Core'

1 - OVERVIEW

Board Inspector

- Provides a scheme of pins similar to physical position
- A live tool which represents the current pins state
- Digital pins are shown with green/red icons which represent high/low (1/0) values

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	● out	GPIO2	13	14	Ground (0v)		
22		GPIO3	15	16	GPIO4	● out	23
		3.3v	17	18	GPIO5		24
10		MOSI (SPI)	19	20	Ground (0v)		
9		MISO (SPI)	21	22	GPIO6	● in	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	● out	GPIO22	31	32	GPIO26		12
13		GPIO23	33	34	Ground (0v)		
19		MISO (SPI)	35	36	GPIO27	● out	16
26	● in	GPIO25	37	38	MOSI (SPI)		20
		Ground (0v)	39	40	SCLK (SPI)		21

1 - OVERVIEW

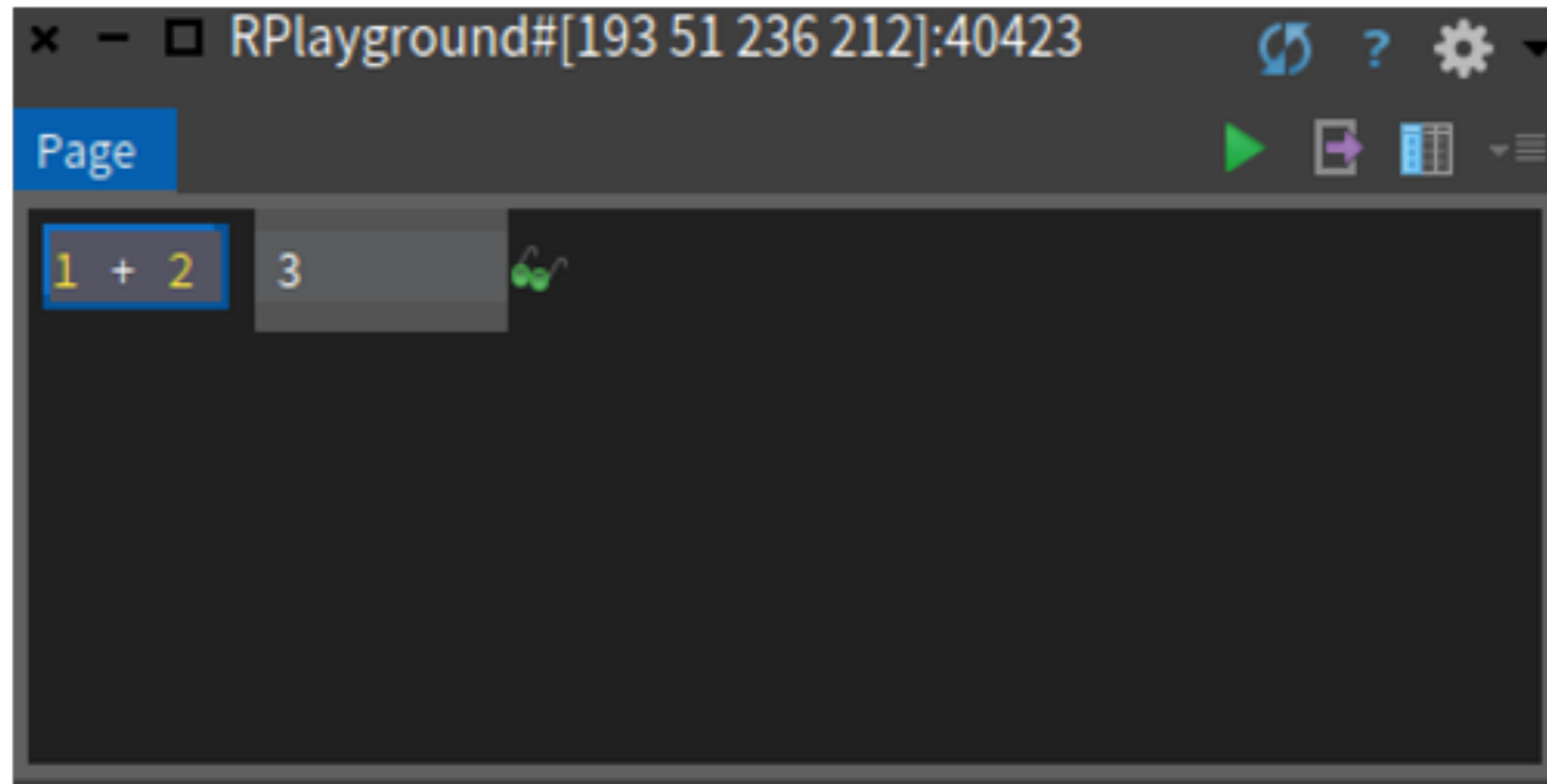
*With PharoThings you can to develop tools to lively program, explore and debug **remote** boards.*

- It is part of **TelePharo** project
- With remotePharo instance you can open:
remote playground
remote system browser or
remote process browser

```
remotePharo := TlpRemoteIDE connectTo: (TCPAddress ip: #[193 51 236 212] port: 40423).  
remotePharo openPlayground.  
remotePharo openBrowser.  
remotePharo openProcessBrowser.  
TlpRemoteIDE disconnectAll.
```

1 - OVERVIEW

Remote Playground



1 - OVERVIEW

Remote System Browser

The screenshot displays the Remote System Browser interface for the RpiBoard3B class. The top window title is "RpiBoard3B>>initialize in #[193 51 236 212]:40423". The left sidebar shows a tree view of packages, with "PharoThings-Hardware-RaspberryPi" selected. The middle pane lists several classes, including "RpiBoard3B", "RpiBoardB", "RpiBoardBRev1", "RpiBoardBRev2", "RpiWiringPiDriver", "PotDigitalInputPinMode", "PotDigitalOutputPinMode", and "PotNoPinResistor class". The right pane shows the "initialize" method, which is currently selected. Below the panes, there are tabs for "Comment", "RpiBoard3B", "initialize", and "Inst. side method". The bottom pane displays the source code for the "initialize" method:

```
initialize
  super initialize.
  self addNewConnector: 'P1' with: {
    PotPower3dot3VPin new. PotPower5VPin new.
    PotSDAPin id: 2. PotPower5VPin new.
    PotGNDPin id: 0. PotGroundPin new.
```

The bottom status bar shows "20/25 [65]" and "initialization extension F +L W".

1 - OVERVIEW

Remote Process Browser

The screenshot displays the Remote Process Browser interface. The main window is titled "ProcessBrowser in #[193 51 236 212]:40423". It shows a list of processes on the left and a detailed view of the selected process on the right.

Process	Method
Morphic UI Process: Delay>>wait	DelayWaitTimeout
Delay Scheduling Process: DelaySpinScheduler>>runTimerEvent	BlockClosure
Low Space Watcher: SmalltalkImage>>lowSpaceWatcher	DelayWaitTimeout
Idle Process: ProcessorScheduler class>>idleProcess	Semaphore
Input Event Fetcher Process: InputEventFetcher>>waitForInput	Socket
WeakArray Finalization Process: WeakArray class>>finalizationPr	Socket
a BasysServer(40423): DelayWaitTimeout>>wait	wait
a BasysConnection: DelayWaitTimeout>>wait	TCPPharoNetworkLibrary class
a SeamlessNetwork: ClyDefaultSystemEnvironmentPlugin>>prep	BasysServer
	BlockClosure
	BasysServer

Below the process list, there are buttons for "Update", "Debug", "Terminate", "Browse", "Implementors", and "Senders".

The detailed view for the selected process shows the following information:

```
wait
```

"Wait until either the semaphore is signaled or the delay times out"
[self schedule.
"It is critical that the following has no suspension point so that
the test and the wait primitive are atomic. In addition, if the delay
is no longer being waited on while entering the way we know that it

At the bottom, there is a status bar with "1/12 [1]" and "waiting extension F +L W".

2 - INSTALLATION

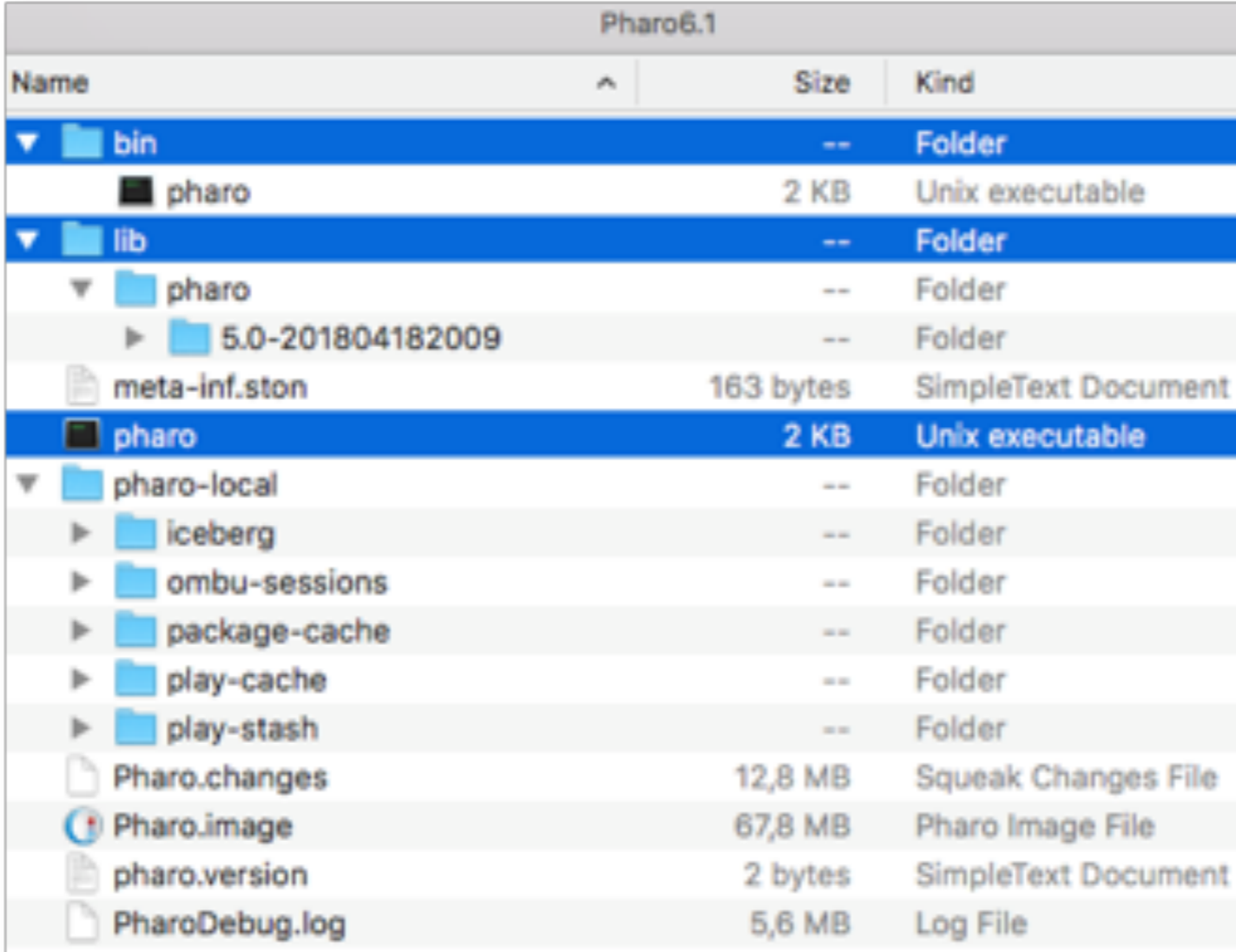
How to run Pharo on ARM architecture?

- Metacello new
baseline: 'PharoThings';
repository: 'github://pharo-iot/PharoThings/src';
load: #(RemoteDevServer Raspberry).
- Metacello new
baseline: 'PharoThings';
repository: 'github://pharo-iot/PharoThings/src';
load: 'RemoteDev'

2 - INSTALLATION

- ArmVM:

files.pharo.org/vm/pharo-spur32/linux/armv6/latest.zip



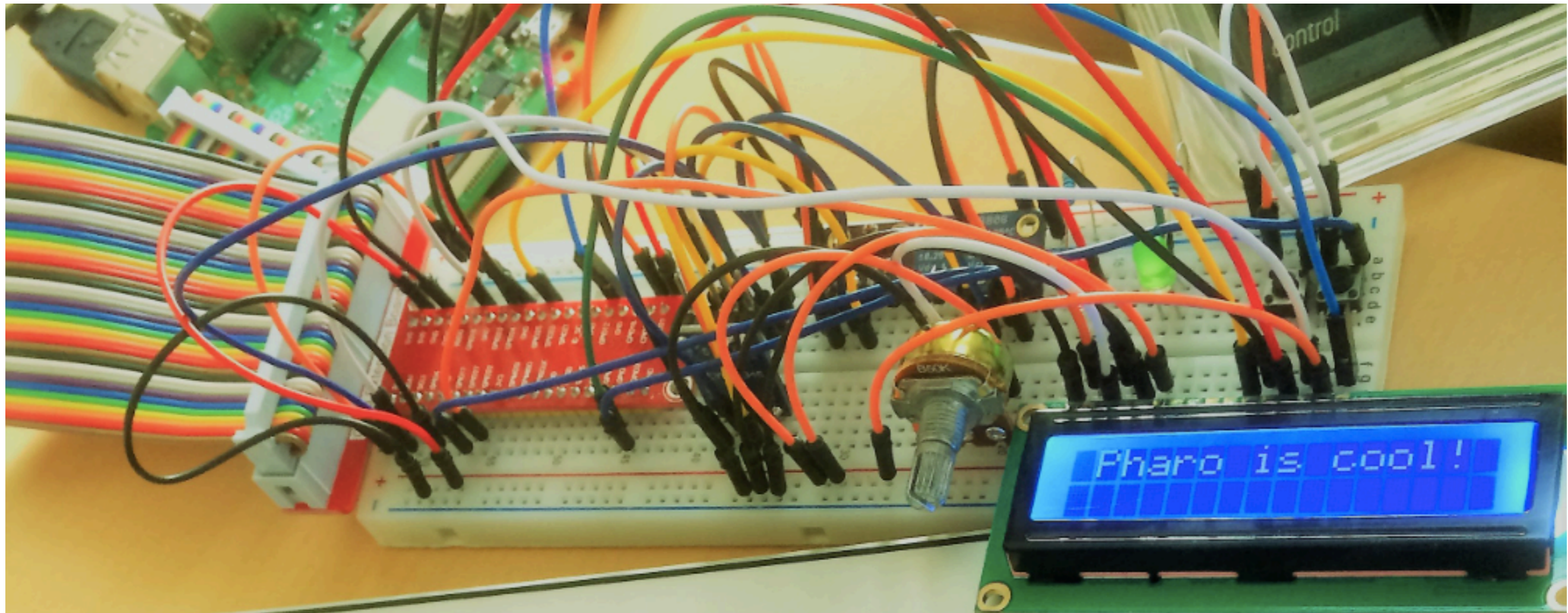
Name	Size	Kind
bin	--	Folder
pharo	2 KB	Unix executable
lib	--	Folder
pharo	--	Folder
5.0-201804182009	--	Folder
meta-inf.ston	163 bytes	SimpleText Document
pharo	2 KB	Unix executable
pharo-local	--	Folder
iceberg	--	Folder
ombu-sessions	--	Folder
package-cache	--	Folder
play-cache	--	Folder
play-stash	--	Folder
Pharo.changes	12,8 MB	Squeak Changes File
Pharo.image	67,8 MB	Pharo Image File
pharo.version	2 bytes	SimpleText Document
PharoDebug.log	5,6 MB	Log File

2 - INSTALLATION

- `#chmod +x pharo`
- `#chmod +x lib/pharo/5.0-201804182009/pharo`
- `#./pharo Pharo.image`
- `#./pharo --headless Pharo.image remotePharo
--startServerOnPort=40423`

3 - PLAYING

Controlling LEDs, sensors and LCD display remotely



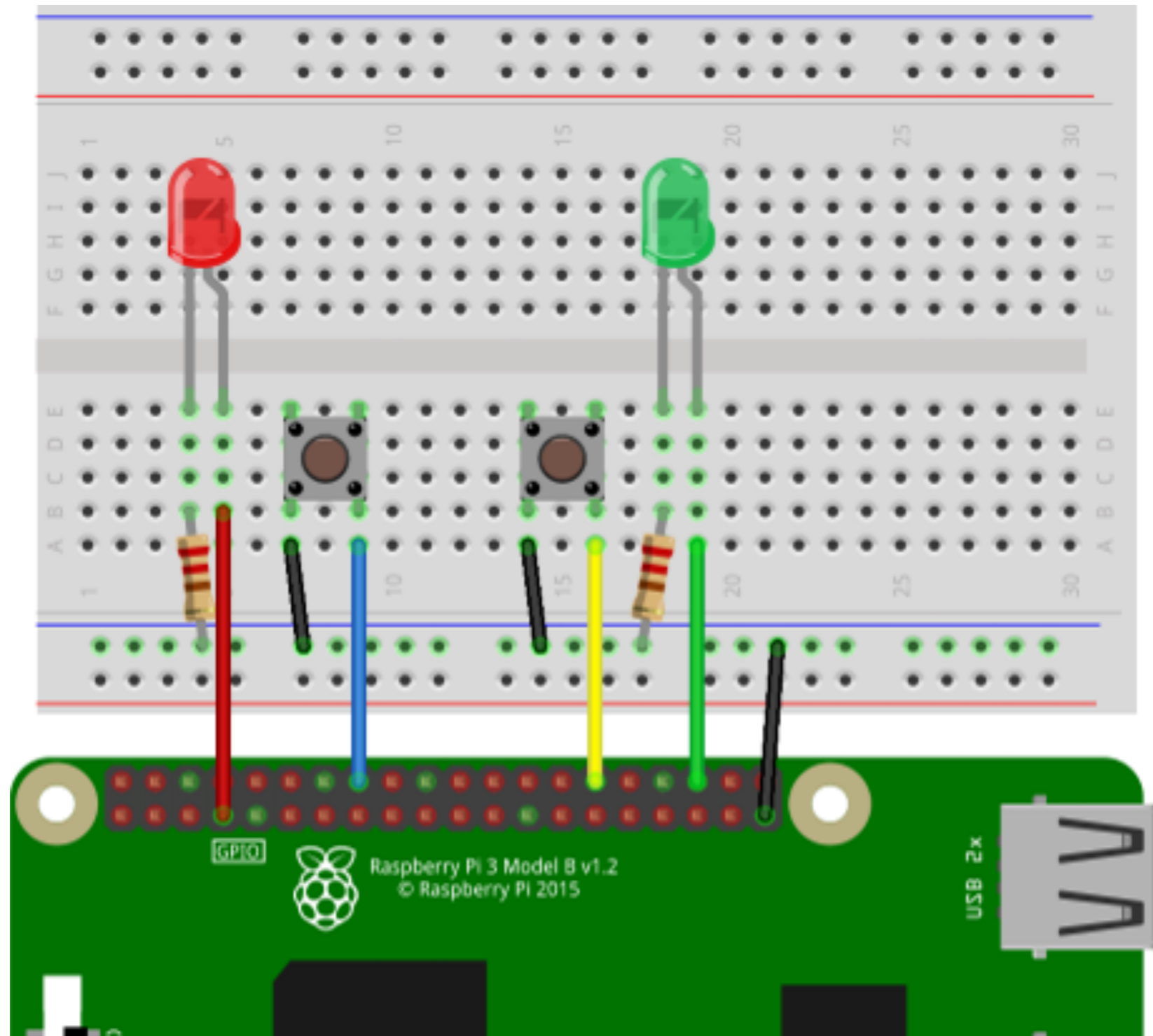
3 - PLAYING

Playing with LEDs and Button

- 1 Raspberry Pi (any model) connected to your network (wired or wireless)
- 1 Breadboard
- 2 LEDs
- 2 Buttons
- 2 Resistors (330ohms)
- Jumper wires

3 - PLAYING

Playing with LEDs and Button



3 - PLAYING

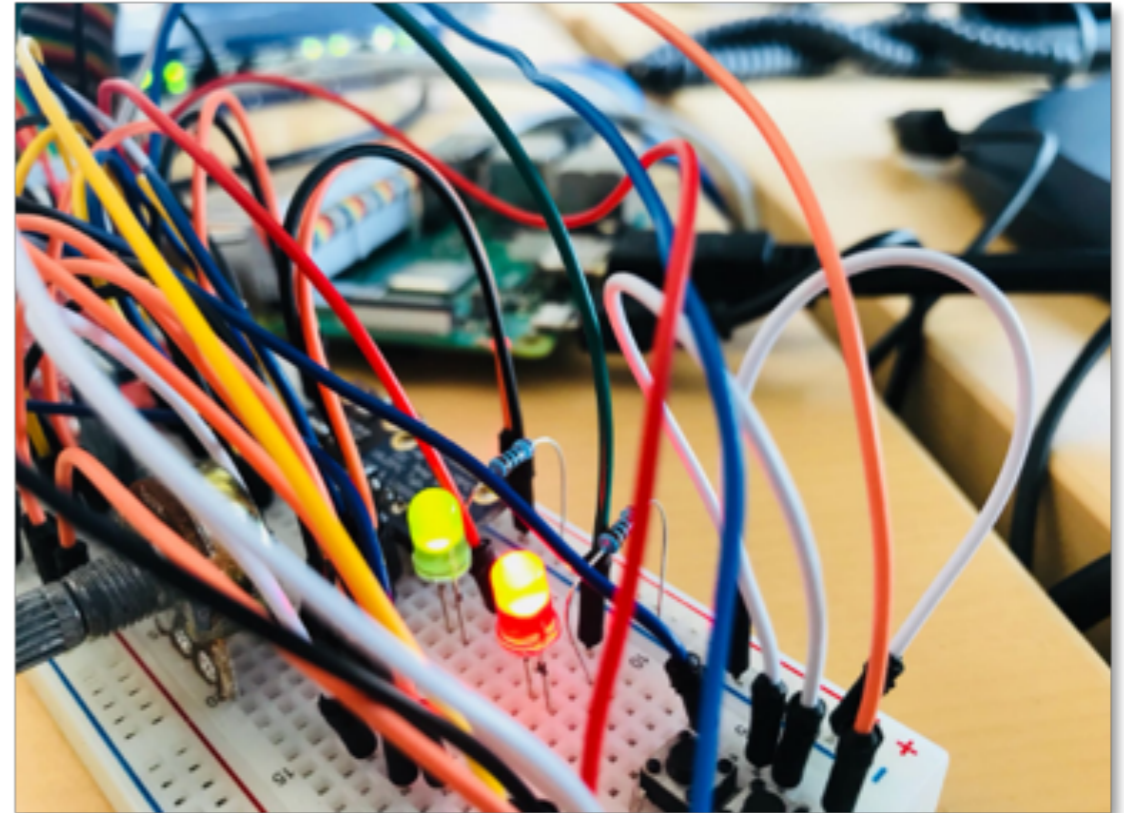
Playing with LEDs and Button

```
ledRed := gpio22.  
ledRed beDigitalOutput.  
ledRed value: 1.  
ledRed toggleDigitalValue.
```

```
buttonRed := gpio25.  
buttonRed beDigitalInput.  
buttonRed enablePullUpResister.
```

```
buttonProcess := [ [100 milliSeconds wait.  
    ledRed value: (buttonRed value=0) asBit  
] repeat  
] forkNamed: 'button process'.
```

```
buttonProcess terminate.
```



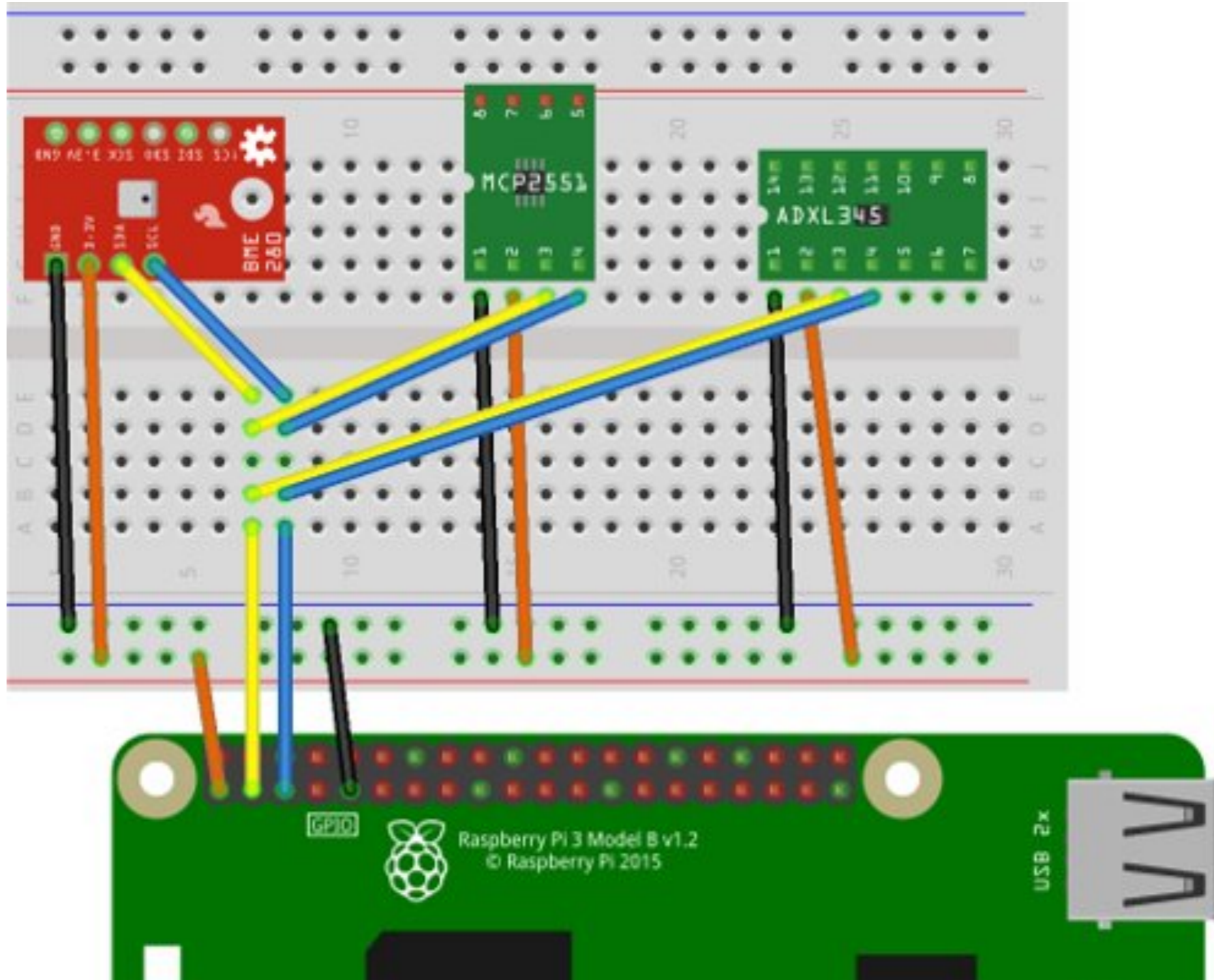
3 - PLAYING

Playing with Sensors (I2C)

- 1 Raspberry Pi (any model) connected to your network (wired or wireless)
- 1 Breadboard
- 1 BME280 sensor
(Temperature, pressure and humidity)
- 1 MCP9808 sensor (Temperature)
- 1 ADXL345 sensor (Accelerometer, axis X, Y and Z)
- Jumper wires

3 - PLAYING

Playing with Sensors (I2C)



3 - PLAYING

Playing with Sensors (I2C)

- Add the follow line in /boot/config.txt
dtparam=i2c1=on
- Add the 'pi' user to I2C group and restart the Raspberry
sudo adduser pi i2c
- Code:
 - a:= board installDevice: PotBME280Device new.
a readParameters.
 - b := board installDevice: PotMCP9808Device new.
b readTemperature.
 - c := board installDevice: PotADXL345Device new.
c readCoordinates.

3 - PLAYING

Playing with Sensors (I2C)

```
a := board installDevice: PotBME280Device new.
```

```
a readParameters. #(26.867964648851192 1011.8081171444288 38.00360479983378)
```

```
b := board installDevice: PotMCP9808Device new.
```

```
b readTemperature. 26.625
```

```
c := board installDevice: PotADXL345Device new.
```

```
c readCoordinates. #(2 65269 21)
```

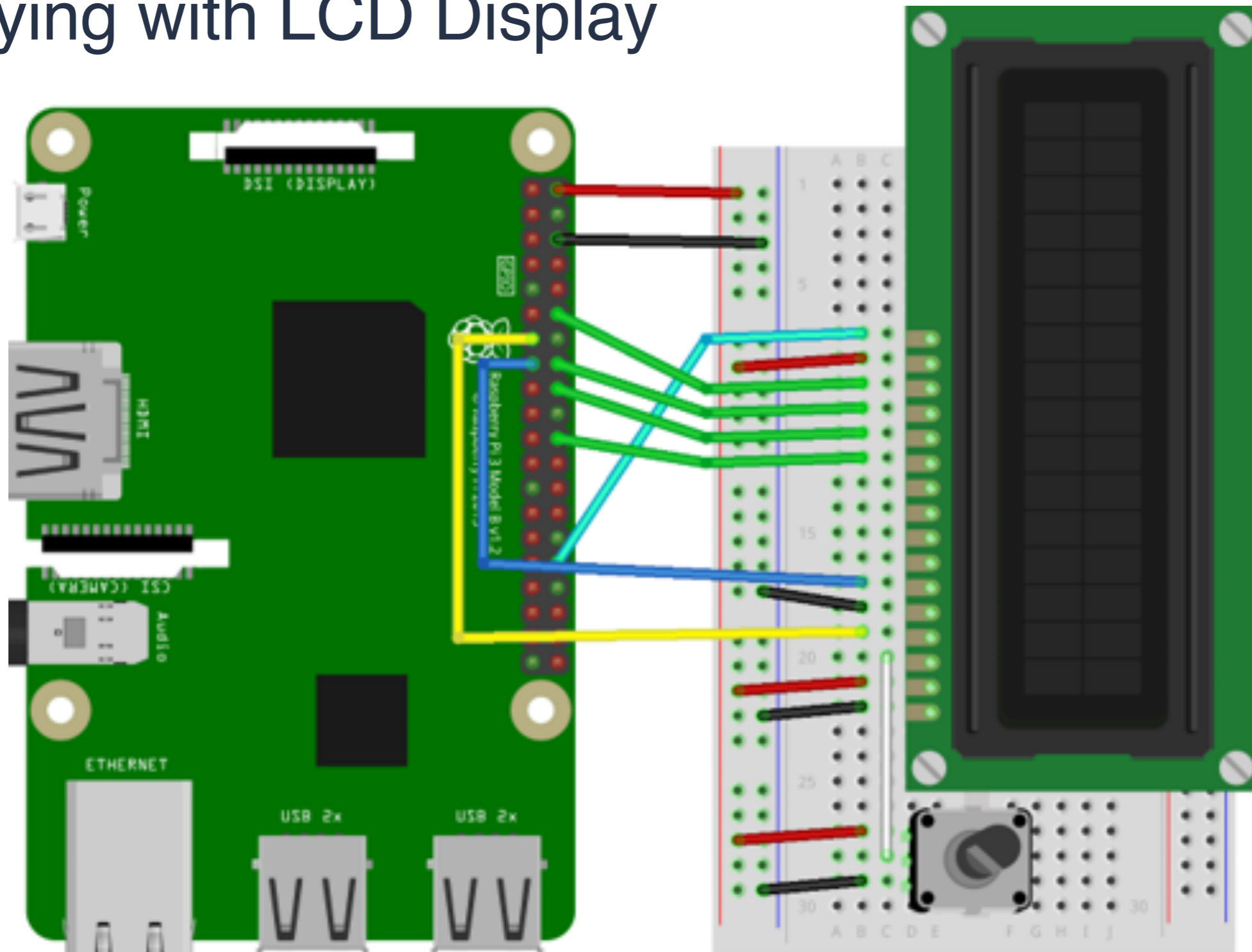

3 - PLAYING

Playing with LCD Display

- 1 Raspberry Pi (any model) connected to your network (wired or wireless)
- 1 Breadboard
- 1 LCD 1602
- Jumper wires

3 - PLAYING

Playing with LCD Display



3 - PLAYING

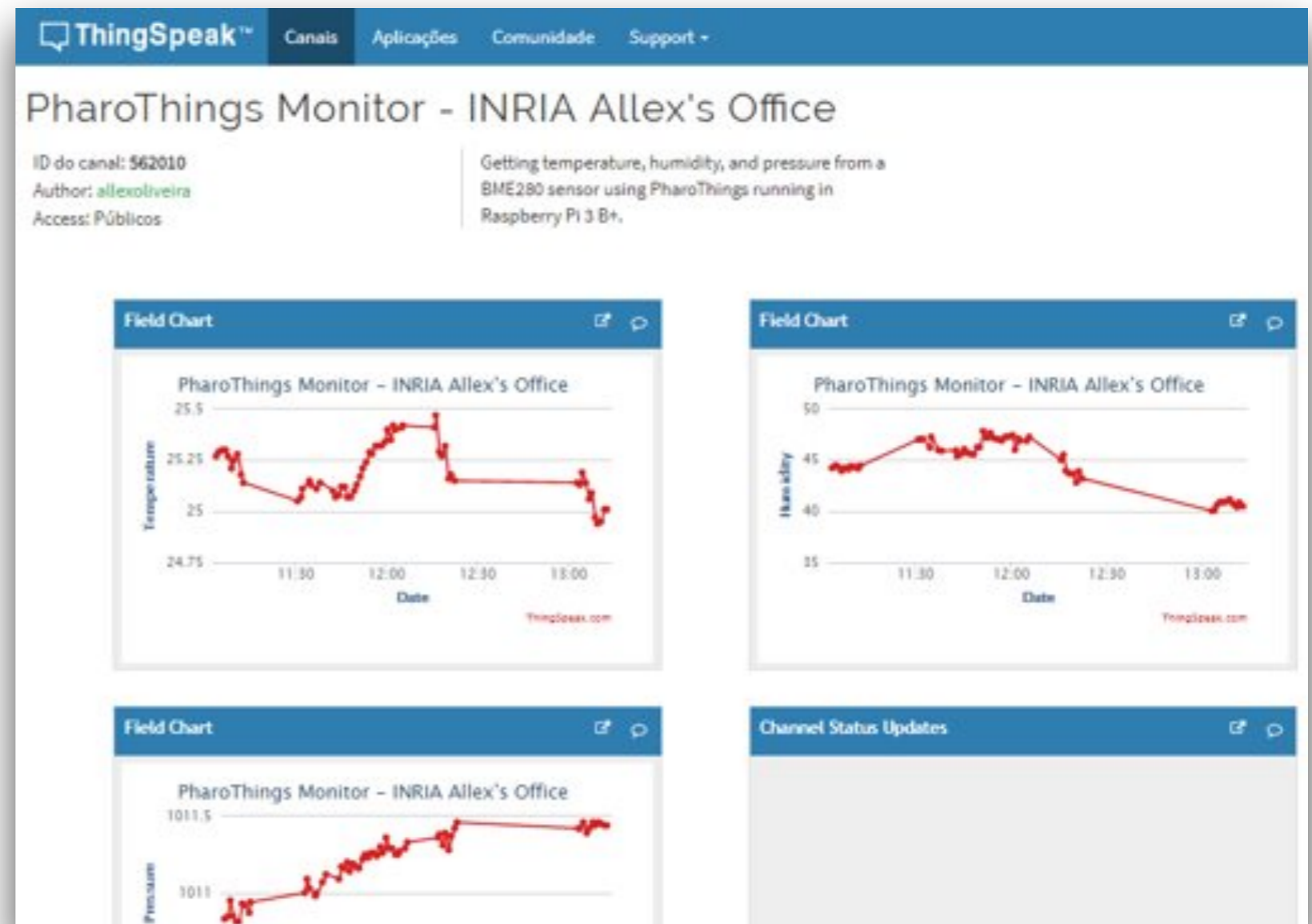
Playing with LCD Display

```
lcd := board installDevice: PotLCD1602Device new.  
lcd message: 'Hello everybody!\nPharo is cool!'.  
lcd clear.
```



4 - PERSONAL WEATHER STATION

Taking the sensor data with Pharo, showing in LCD display and sending to a remote server



4 - PERSONAL WEATHER STATION

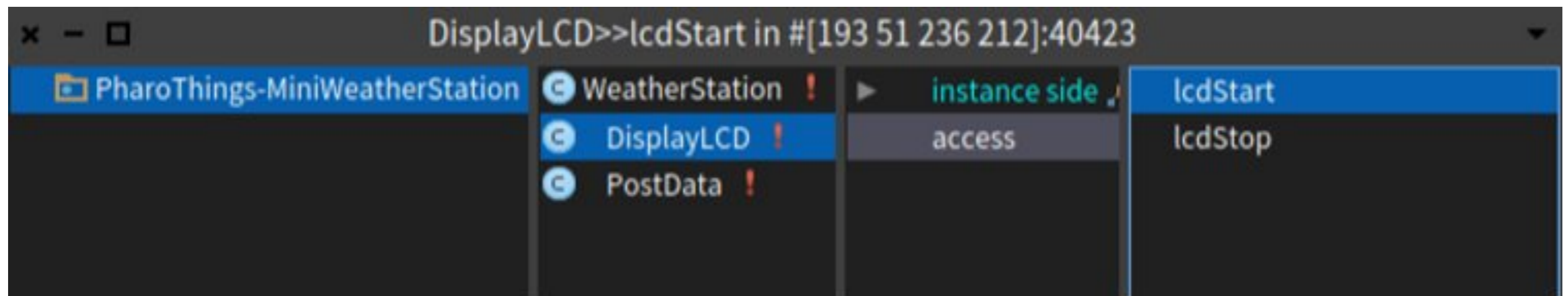
- We will run PharoThings in a Raspberry Pi
- To collect the sensor data (BME280)
temperature, humidity and pressure
- Show this data in a LCD
- Send the data to a remote server

4 - PERSONAL WEATHER STATION

- Created class to instantiate the LCD and Sensor
- Created subclass to create process:

print information on LCD each 1 second

send data to a webserver in cloud each 1 minute



4 - PERSONAL WEATHER STATION

- Started Pharo on Raspberry boot with arguments to start the 2 process (DisplayLCD and PostData)

/etc/init.d/pharo.sh

```
#PATH=/home/pi/pharo-iot/  
#NAME=pharo  
sudo /home/pi/pharo-iot/pharo --headless /home/pi/pharo-iot/Pharo6.1-IoT.image /home/pi/pharo-iot/start.st  
exit
```

/home/pi/pharo-iot/start.st

```
(DisplayLCD new) lcdStart.  
(PostData new) apiKey:'F1MKEG7PJ4JY30L8'; dataStart.  
TlpRemoteUIManager registerOnPort: 40423.
```

LCD Display



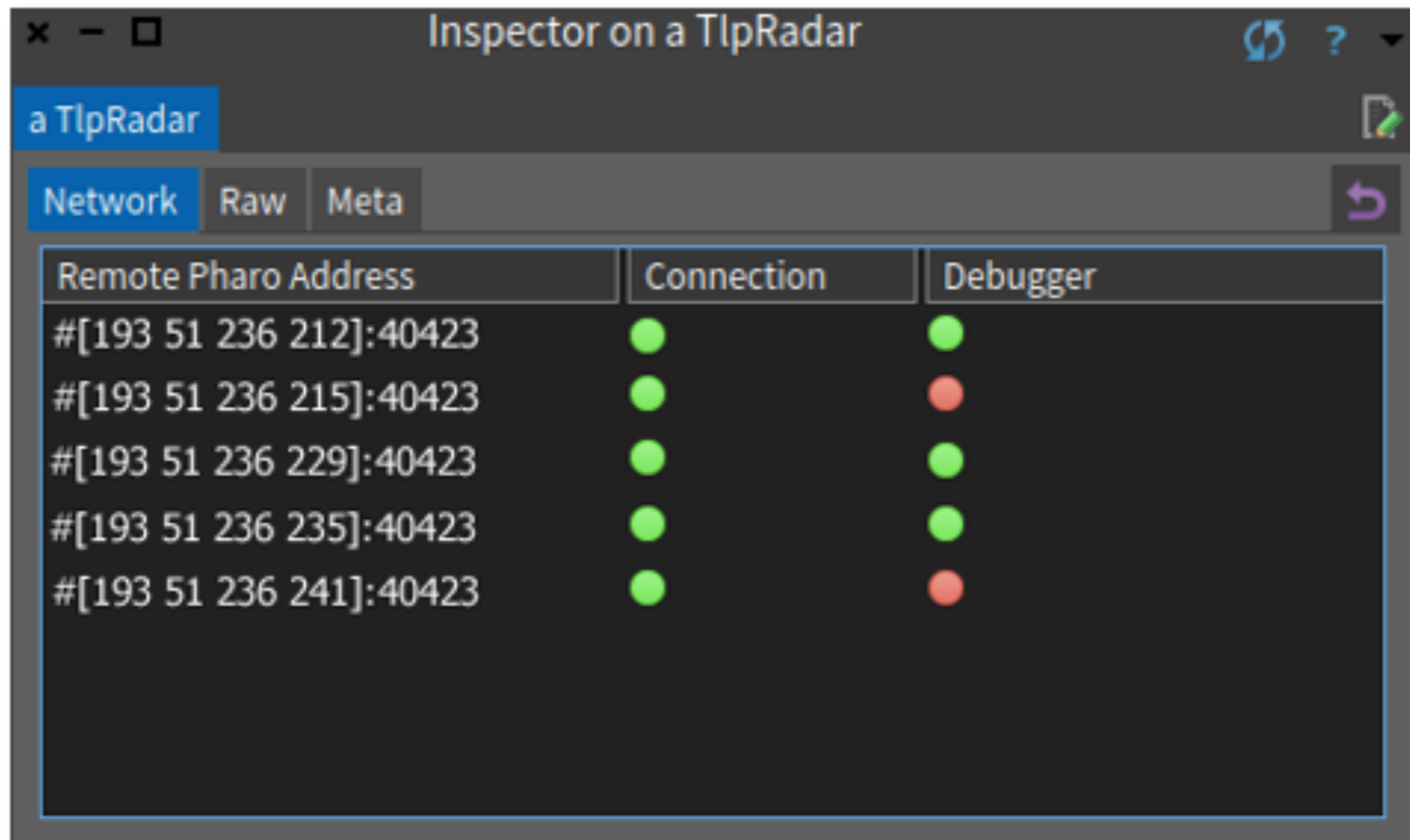
5 - THE FUTURE

Projects and news

- Beaglebone models
- Zeroconf for armVM + PharoThings
- Remote refactoring
- Security

5 - THE FUTURE

Automatic detection of running images in network
(TeleRadar using SSDP protocol)



The screenshot shows a window titled "Inspector on a TlpRadar" with a tab labeled "a TlpRadar". Below the tab are three sub-tabs: "Network" (selected), "Raw", and "Meta". A table displays the following data:

Remote Pharo Address	Connection	Debugger
#[193 51 236 212]:40423	●	●
#[193 51 236 215]:40423	●	●
#[193 51 236 229]:40423	●	●
#[193 51 236 235]:40423	●	●
#[193 51 236 241]:40423	●	●

5 - THE FUTURE

Middleware to manage the devices

The image displays the PharoThings Cloud web interface. The top navigation bar includes a search bar and a menu icon. The left sidebar contains navigation options: Dashboard, Statistics, Map, Devices, and Settings. The main content area is divided into two sections. The upper section, titled 'Dashboard / Devices', shows a table of device information. The lower section, titled 'Dashboard / Devices / Office-01', provides a detailed view of a specific device, including its model, uptime, last data received, and various status indicators. A line graph at the bottom right shows memory and CPU usage over time.

Device Name	IP Address	Status	Model	Version	Actions
Device-01	193.51.236.212	Connected	Raspberry 3 B+	1.4.3	Playground Browser Upgrade Restart
Device-02	193.51.236.213	Disconnected	Raspberry 3 B	1.2.1	Playground Browser Upgrade Restart
Device-03	193.51.236.214	Disconnected	Raspberry 2 B	1.4.3	Playground Browser Upgrade Restart
Device-04	193.51.236.215	Disconnected	Raspberry 1 B v1	1.4.3	Playground Browser Upgrade Restart
Device-05	193.51.236.216	Disconnected	Raspberry 1 B v2	1.4.3	Playground Browser Upgrade Restart
Device-06	193.51.236.217	Disconnected	Raspberry 3 B+	1.4.3	Playground Browser Upgrade Restart
Device-07	193.51.236.218	Disconnected	Raspberry 3 B+	1.4.3	Playground Browser Upgrade Restart
Device-08	193.51.236.219	Disconnected	Raspberry 3 B+	1.1.6	Playground Browser Upgrade Restart

Device Details:

- Model: Raspberry 3 B+
- Uptime: 3d 12h 13min
- Last data received: 3 min ago
- PharoThings version: 0.1
- TelePharo version: 0.1

Network Information:

- MAC Address: 78:24:90:45:a6:e7
- Internal IP: 10.0.23.212
- External IP: 193.51.236.212
- Router failed (24h): 1 times
- Internet failed (24h): 2 times

Usage Graph:

Hour	Memory (%)	CPU (%)
3	15	5
7	15	5
11	15	5
15	15	5
19	15	5
23	15	5

5 - THE FUTURE

Website official with many lessons and tutorials

Draft: <http://pharothings.allexoliveira.com.br>

The screenshot shows the PharoThings website interface. At the top, there is a dark header with the logo "PHAROTHINGS" and the tagline "control everything from anywhere". Below the header is a navigation menu with links for Home, About, Tutorials, Learn, Project Hub, and Contact. The "Learn" menu is open, showing a sub-menu for "Raspberry" with a right-pointing arrow. The sub-menu lists several lessons: Introduction, Lesson 1 - Turn on/off LED, Lesson 2 - Blinking LED, Lesson 3 - LED flowing Lights, Lesson 4 - Controlling LED by a button, Lesson 5 - LED breathing (PWM), and Lesson 6 - Temperature sensor. The main content area displays the "LESSON 1 - TURN ON/OFF LED" page, which includes an "Introduction" section. The text in the introduction reads: "One of the classic analogies in electronics to 'Hello World' is turn... In this first lesson, we will learn how to connect correctly an LED i... control this led by turning it on and off. Coding in Pharo is very simple, but it is very powerful and **you can**... If you didn't see how to install the PharoThings on your Raspberri... tutorial: [Installing PharoThings on your Raspberry Pi](#)".

WITH PHAROTHTINGS 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

Easy, powerful.

THANKS!



Any questions?

marcus.denker@inria.fr
allex.oliveira@msn.com

PRESENTATION INFORMATION

This slides was presented at ESUG 2018, in Cagliari, ITALY

- Title: Pharo IoT

- Presenters:

Marcus Denker - marcusdenker.de / zweidenker.de

Alex Oliveira - www.linkedin.com/in/alex-oliveira

ESUG conferences

<http://www.esug.org/wiki/pier/Conferences>

INRIA

<https://www.inria.fr/>

RMOD TEAM

<https://rmod.inria.fr/web>

PHARO PROJECT

<https://github.com/pharo-project/pharo>

PHAROTINGS PROJECT

<https://github.com/pharo-iot/PharoThings>