PHARO IoT

Installation Improvements and Continuous Integration

Allex Oliveira
IoT Engineer @ RMoD Team, INRIA
allex.oliveira@msn.com
Summary

1 – Getting experience
2 – Making the process easy
3 – Continuous Integration
4 – 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, 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

Firmata

Inspector on a PotRemoteBoard (a RpiBoard3!)
Motivation

• It's not very easy to get started with PharoThings for anyone who is not a Pharo/Smalltalk user.
• How can we automate the process to make the developer’s life easier?

Before we automate something, we should ask: How can we improve it?
How to know what to improve?

Real world experience

IoT Workshop, ESUG
September 2018 - Cagliari, Italy
How to know what to improve?

Real world experience

IoT Hackaton, Zweidenker GmbH
October 2018 - Cologne, Germany
How to know what to improve?

Real world experience

Live Programming IoT devices with PharoThings
January 2019 - Can Tho University, Vietnam
How to know what to improve?

Real world experience

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
2 - Difficulties

1. Is not so easy to start (many steps needed to run it);
2. Takes a long time to start (get everything);
3. Download VMs to different OS (ARM, Linux, Win, OSX);
4. Need the same version on server/client;
5. Needs a keyboard/screen on Raspberry to install from scratch;
6. No many application examples;
7. Search example code: copy and paste (pdf bad formatting);
2 - Difficulties

1. Is not so easy to start (many steps needed to run it);

2. Takes a long time to start (get everything);
How to install (manually)

1. Install PharoLauncher and create a new image
2. Load PharoThings project from Github
3. Copy ARM VM and Pharo Image to Raspberry Pi
4. Set the run permissions "chmod +x"
5. Run TelePharo server in a big command line
6. Extra configurations: set hostname, WiFi, autorun
How to install (manually)

1. Install PharoLauncher and create a new image
How to install (manually)

1.  
2. Load PharoThings project from Github

Playground

```metacello
Metacello new
baseline: 'PharoThings';
repository: 'github://pharo-iot/PharoThings/src';
load: 'RemoteDev'.
```

```metacello
Metacello new
baseline: 'PharoThings';
repository: 'github://pharo-iot/PharoThings/src';
```
How to install (manually)

3. Copy ARM VM and Pharo Image to Raspberry Pi
How to install (manually)

4. Set the run permissions "chmod +x"
How to install (manually)

5. Run TelePharo server in a big command line

```
pi@pharoiot-01:~/.pharothings > ssh pi@pharoiot-01 76×24

pi@pharoiot-01:~/.pharothings $ ./pharo --headless Pharo32.image remotePharo --startServerOnPort=40423

'a TlpRemoteUIManager is registered on port 40423'
```
How to install (manually)

6. Extra configurations

- Auto connect on WiFi network
- Set hostname
- Autostart TelePharo server every boot
- Set boot to console (disable the Linux UI)
- Enable I2C and SPI kernel modules
How to 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!
How to install (zero-conf)

6. Extra configurations
2 - Difficulties

1. Is not so easy to start (many steps needed to run it);
2. Takes a long time to start (get everything);
3. Download VMs to different OS (ARM, Linux, Win, OSX);
4. Need the same version on server/client;
Packing everything

+ 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
2 - Difficulties

1. Is not so easy to start (many steps needed to run it);
2. Takes a long time to start (got everything);
3. Download VMs to different OS (ARM, Linux, Win, OSX);
4. Need the same version on server/client;
5. Needs a keyboard/screen on Raspberry to install from scratch;
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!
2 - Difficulties

1. Is not so easy to start (many steps needed to run it);
2. Takes a long time to start (got everything);
3. Download VMs to different OS (ARM, Linux, Win, OSX);
4. Need the same version on server/client;
5. Needs a keyboard/screen on Raspberry to install from scratch;
6. No many application examples;
Chapter 12.5

Creating the application

PharoThings Monitor - INRIA Allex's Office

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
the cloud. Your final code will seem like the Picture 12-2.

Figure 4.9: Creating an operation method.

Remote playground.

Figure 4.8

<table>
<thead>
<tr>
<th>blinker</th>
</tr>
</thead>
<tbody>
<tr>
<td>blinker := Blinker new,</td>
</tr>
<tr>
<td>blinker timesRepeat: 10 waitForSeconds: 1.</td>
</tr>
</tbody>
</table>

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:

\[
\text{remotePharo saveImage.}
\]

4.9 Save your work

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

1. Is not so easy to start (many steps needed to run it);
2. Takes a long time to start (get everything);
3. Download VMs to different OS (ARM, Linux, Win, OSX);
4. Need the same version on server/client;
5. Needs a keyboard/screen on Raspberry to install from scratch;
6. No many application examples;
7. Search example code: copy and paste (pdf bad formatting);
Welcome Window PharoThings

PharoThings Quickstart guide

Welcome to Pharo, an immersive live programming environment.

This Pharo image already comes with PharoThings installed. PharoThings is a live programming platform for IoT projects based on Pharo.

It includes:
- Development tools to lively program, explore and debug remote boards (based on TelePharo)
- Board modeling library which simplifies board configuration

For more information, please visit here: https://github.com/pharo-iot/PharoThings

Connecting in PharoThings server by IP

remotePharo := TlpRemoteIDE connectTo: (TCPAddress ip: #[192 168 1 209] port: 40423).

"Connecting in PharoThings server by Hostname"
ip := NetNameResolver addressForName: 'pharothings-01'.
remotePharo := TlpRemoteIDE connectTo: (TCPAddress ip: ip port: 40423).

"Inspect remote board"
remoteBoard := remotePharo evaluate: [ RpiBoard3B current ].
remoteBoard inspect.

"Open remote Playground, remote Browser and remote Process Browser"
remotePharo openPlayground.
remotePharo openBrowser.
remotePharo openProcessBrowser.
2 - Difficulties

1. Is not so easy to start (many steps needed to run it);
2. Takes a long time to start (get everything);
3. Download VMs to different OS (ARM, Linux, Win, OSX);
4. Need the same version on server/client;
5. Needs a keyboard/screen on Raspberry to install from scratch;
6. No many application examples;
7. Search example code: copy and paste (pdf bad formatting);
3 - Continuous Integration

After these improvements, now we can automate the process:

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

How to automate?
3 - Continuous Integration

Travis CI on PharoThings Booklet

• With each commit in Github, a new instance of a Travis virtual machine is created and the scripts executed
• The booklet is created using Pillar, a Pharo tool-suite to generate documentation, books, websites and slides
• Deploy in Github releases is done in each changes on the booklet
• The process is transparent, anyone can see and contribute to improvements
3 - Continuous Integration

Travis CI on PharoThings Booklet

github.com/SquareBracketAssociates/Booklet-APharoThingsTutorial/.travis.yml

travis-ci.org/SquareBracketAssociates/Booklet-APharoThingsTutorial

new commit ➔ run scripts/tests

new Github release ➔

https://github.com/SquareBracketAssociates/Booklet-APharoThingsTutorial
3 - Continuous Integration

Travis CI on PharoThings Booklet

https://github.com/SquareBracketAssociates/Booklet-APharoThingsTutorial
3 - Continuous Integration

Travis CI on Pharo IoT

• With each commit in Github, a new instance of a Travis virtual machine is created and the scripts executed
• Deploy is done only when a new Github Tag is created
• PharoThings is loaded in 32/64 images
• Zip files with everything (VMs, Images, 1 click-run) are up in Github releases. We put it inside the repo also.
• The process is transparent, anyone can see and contribute to improvements
3 - Continuous Integration

Travis CI on Pharo IoT

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

travis-ci.org/pharo-iot/Ci

new commit

run scripts/tests

is it tag?

yes

no

new Github release

Github push

https://github.com/pharo-iot/Ci
3 - Continuous Integration

Travis CI on Pharo IoT

Pharo IoT Continuous Integration

Pharo IoT CI is a Continuous Integration process created using Travis CI. PharoThings and TelePharo already installed and ready to run in multiple click files. You can see the files to download in Releases page. All zip files displayed when a new tag is created here.

You can run the scripts by yourself in your local machine (Linux, Mac). CI runs the script .run-scripts. If you wanna skip the PharoThings installation, see the results inside the folder results-$DATE.

We are using Github Pages to run the get.pharo.org. You can see the files as well.

https://github.com/pharo-iot/Ci
• We are using the Github Pages to host the zero-conf pages

get.pharoiot.org

Pharo IoT Server Zeroconf Raspberry

This script downloads `server.zip` file that contain:

- Pharo7 image 32 bit
- Pharo ARM VM
- Pharo IoT server installed

Plattaform

Raspberry Pi running Raspbian

Usage

`wget -O - get.pharoiot.org/server | bash`

Artifacts

pharo Script to run Pharo in the headless mode
pharo-ui Script to run Pharo in UI mode
pharo-server/ Start pharo in headless mode with TelePharo listening on port
vm/ Directory containing the VM

Pharo IoT Server Example

Start Pharo IoT server: `./pharo-server`
Open Pharo user interface: `./pharo-ui`
Start the server (Playground): `TlpRemoteUNManager registerCampPort:40423`

https://github.com/pharo-iot/Ci/docs
We are starting using Github Projects to work in a more transparent and dynamic way.

You can take some tasks!

https://github.com/orgs/pharo-iot/projects/1
Now you can

- **Raspbian IoT** < 10 minutes
- **Pharo IoT** < 1 minute
- **Pharo IoT IDE** < 1 minute

- **Booklet**
- **Quickstart Guide**
- **Be part of the project!**
4 - Future
pharoiot.org

Everything in the same place, to facilitate the journey of the new user.
Share your IoT projects using Pharo with the community!

Community projects

Submit your project
Tele Radar

Automatic detection of running images in network

(TeleRadar using SSDP protocol)
PharoThings Booklet inside Pharo

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:

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

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

• Norbert Hartl

---

playground for minimal image creation

This repository contains scripts and patches in order to build a pharo minimal images. The builds are using the VM_ARCH environment variable to detect which version to build. By using

```
VM_ARCH=32 make .....
```

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

NOW IN LESS THAN 10 MINUTES!

THANKS!
Any questions?
alex.oliveira@msn.com
Presentation Information

This slides was presented at Pharo Days 2019, Lille, France
https://pharo.org/2019PharoDays

• Title: Pharo IoT - Installation Improvements and Continuous Integration
• Presenters:
  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