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

Allex Oliveira  
IoT Engineer  
allex.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, Allex Oliveira joined the Rmod Team to continue the project
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
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
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

```plaintext
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
Remote Process Browser

1 - OVERVIEW

```
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
```
2 - INSTALLATION

How to run Pharo on ARM architecture?

• Metacello new
  baseline: 'PharoThings';
  repository: 'github://pharo-iot/PharoThings/src';

• 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
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
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'.
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)

```plaintext
a := board installDevice: PotBME280Device new.
a readParameters. #(26.867964648851192 1011.8081171444288 38.00360479983378)
```

```plaintext
b := board installDevice: PotMCP9808Device new.
b readTemperature. 26.625
```

```plaintext
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)
5 - THE FUTURE

Middleware to manage the devices
5 - THE FUTURE

Website official with many lessons and tutorials

Draft: http://pharothings.allexoliveira.com.br
WITH PHARO THINGS 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
  Allex Oliveira - www.linkedin.com/in/allex-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
PHAROTHINGS PROJECT
https://github.com/pharo-iot/PharoThings