Lua as a common language for the IoT
Dipl.-Ing. Andre Riesberg, Nogs GmbH - Lua Workshop Moskow September 2014

Hello, how are you?

fine, and you?
Agenda

1. Smart objects in the Internet of Things
2. Babylonian confusion in the IoT world
3. Are smart objects really smart today?
4. How nodes can get savvy by means of Lua
5. Nogs - a new IoT framework and communication ecosystem
Smart Objects connect Digital and Real World

- The digital revolution of the 21st century will be much, much larger than previous two digital revolutions of personal computers and the internet.

- We are now facing the digital revolution of the 21st century: Smart objects in the internet of things, that interconnect the digital world with the physical world.

- A smart object is a small microelectronic device that consists of a communication device, typically low power radio, a small microprocessor and a sensor and/or actuator.

Credits: Jean Philippe Vasseur Adam Dunkels
Internet of Things changing Automation Paradigm

- The "Internet of Things" (IoT) is describing billions of embedded devices that are communicating with each other through Internet technology without involving human beings directly.

- In the automation context, the IoT means the shift from centralized and hierarchical control towards cooperative, distributed networks and control structures.

- Formerly passive sensors (simple objects) become active players (smart objects) in networks and are enhanced with the capability for computation and decision making.
Enabling Technologies

- IPv6
- Low Power Wireless (6LoWPAN, ZigBee IP, BLE, Thread)
- New Protocols for constrained Embedded Devices
- Mobile Multiagent Systems
Wireless Sensor Networks

- Environmental Monitoring
- Ambient Assisted Living
- Context-Awareness
- Real-time City Information
- Structural Health
- Smart Energy
- Logistics

Credits: Matthias Kovatsch
Protocols for constrained Embedded Devices

100s - 1000s of bytes

- XML
- HTTP
- TCP
- IP

10s of bytes

- JSON
- CoAP
- UDP
- 6LoWPAN

Credits: Zach Shelby
Agenda

1. Smart objects in the internet of things
2. Babylonian confusion in the IoT world
3. Are smart objects really smart today?
4. How nodes can get savvy by means of Lua
5. Nogs - a new IoT framework and communication ecosystem
Clash of Technologies

Application Standards
e.g. DMX, DALI

Domain Standards
e.g. Bacnet, KNX

IT Standards
e.g. TCP IP
Magic Square for Embedded System Development

Integration & Usability

Scalability & Flexibility

Costs & Time to Market

Security & Safety

?
Agenda

1. Smart objects in the Internet of Things
2. Babylonian confusion in the IoT world
3. Are smart objects really smart today?
4. How nodes can get savvy by means of Lua
5. Nogs - a new IoT framework and communication ecosystem
Embedded today

- Firmware / software updates
- Only predictable content
- Difficult to handle multiple protocols
- No consistent development platform
- Platform independent visualisation?
Highly changeable world

- New requirements at any time
- New types of smart objects e.g. wearables
- New communication protocols e.g. Thread
- Innovation in front-end devices e.g. Tizen
- New nodes pop up.
Let’s start here ...

Imagine all nodes would use the same unified way of communication among each other ...

... and the way how they communicate is exchanging apps [executable save code] ...

... and even sensor nodes are able to run such apps.
And think further ...

Let’s assume everything could be an app: programm, data & communication ...

... even the simple on/off command could be an app which destroys itself after it’s executed ...

... and wouldn’t it be great if these apps could be executed on the fly?
Agenda

1. Smart objects in the internet of things
2. Babylonian confusion in the IoT world
3. Are smart objects really smart today?
4. How nodes can get savvy by means of Lua
5. Nogs - a new IoT framework and communication ecosystem
Basics

- Lua as a language for embedded
  5.2 -> 5.3
- Platform independent software development
- Open Source MIT
- Classification of nodes depending on Lua capabilities
  hardware constraints.
Type of Nodes

- **Permanent Smart Nodes**
- **Temporary Smart Nodes** both with GUI
- **Clever Nodes** with Lua VM
- **Primitive Nodes** single line interpreting
- **Alien Nodes** to connect with software gateway.
Typical Topology
How to do Apps for Embedded?

- Standardized software layers for different types of nodes
  hardware independent

- Unified communication between tasks & devices
  independent from protocols.
Nogs CN Platform

- Application for this node: Light management, heater control...
- Helper module for one or more Applications
- Hardware abstraction for this node: Self introducing
- Hardware depending functions
- Libraries for Lua

- C/C++
- Lua
- HAL
- Sys
- Libraries
  - [RTOS optional]

- IO
  - DALI, DMX, GIO, ...

- Transport
  - JSON, HTTP
  - UDP, CoAP, TCP
  - IPv4, IPv6
  - ZigBee, Eth, WiFi, ...

- Module
  - App

- Module
Dynamic coding as a communication principle

- Clever nodes as a new class of smart objects running a Lua VM
- Those nodes exchange objects where Lua represents data or functions
- These objects are executed on the fly.
Agenda

1. Smart objects in the Internet of Things
2. Babylonian confusion in the IoT world
3. Are smart objects really smart today?
4. How nodes can get savvy by means of Lua
5. Nogs - a new IoT framework and communication ecosystem
Communication Concept

- Distributed system
- Loose couplings
- Event driven with closers
- Security levels depending on requirements.
Security

- All hardware with encryption chip
  „Nogs inside“ requirement

- Authentication of apps

- Special mechanisms
  e.g. for man in the middle & overload attacks

- Bare metal supervisor
  e.g. for APT-infection

- Encryption option
  above protocol level.
Unified Communication

- Using JSON as a compact & human readable standard format
- Using a nesting mechanism to wrap & unwrap plain JSON data
- The wrap & unwarp mechanism is depending on the underlying data protocol and can be nested.
Simplified Commissioning

- Every node and/or subsystem is represented by an avatar
- An avatar is Lua code that describes and manipulates its owner in any kind of representation e.g. graphical
- By this means commissioning can be done with a simple tool, which doesn’t know the specifics of the nodes.
Visualisation with Live Coding

- Visualisation for hardware independent GUI
- Live coding for any type of device
- All objects are Lua coded
- Supporting common Lua based game engines
  e.g. Corona / Gideros / Marmelade.
Occupation of Alien Systems

- SN, CN as PN as applications running on various systems
- Building gateways to Alien systems on SN, CN, PN
- Tunneling through Alien nodes
- Hacking & reverse engineering of Alien nodes?
# Nogs Software & Hardware Matrix

<table>
<thead>
<tr>
<th>Hardware / Software</th>
<th>Windows based</th>
<th>Linux based</th>
<th>Android based</th>
<th>iOS based</th>
<th>Fieldbus Systems</th>
<th>Clever Nodes</th>
<th>Primitive Nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Debugger</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nogs Communicator</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nogs Configurator</td>
<td>Planned</td>
<td></td>
<td>Planned</td>
<td>Planned</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nogs Designer</td>
<td>Yes, Live Coding</td>
<td></td>
<td>Planned</td>
<td>Planned</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nogs SN</td>
<td>SN.EXE</td>
<td>SN.O</td>
<td>SN.APK</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nogs CN</td>
<td>CN.EXE</td>
<td>CN.O</td>
<td>CN.APK</td>
<td>Planned</td>
<td>CN.EXE / Runtime</td>
<td>CN Runtime</td>
<td></td>
</tr>
<tr>
<td>Nogs PN</td>
<td>PN*.EXE</td>
<td>PN*.O</td>
<td>PN*.APK</td>
<td>PN*.IPA</td>
<td>PN*.EXE</td>
<td></td>
<td>PN Runtime</td>
</tr>
<tr>
<td>Nogs MATLAB Simulink Block</td>
<td>Nogs CN Block</td>
<td>Planned</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Rapid Product Development

- Building reference hardware for quick start
- CPU boards for integration in series products
- Open hardware & software
- Turnkey.
Rapid Multiplatform Development

- ZeroBrane Studio
  - Debugger
  - Crossplatform Live Coding
- LuaRocks

- IPv6, IPv4
- TCP
- UDP
- CoAP
- HTTP Client & Server
- FTP
- SMTP

- Bare Metal
- eLua
- RTOS
- Linux
- Android
- iOS
- Windows
PiNogs - on PiNogs Backplane
Nogs Nucleo CN1 - Arduino format

Bare metal with Lua VM & Arduino programming
Nogs Stamp CN1 - for series products
<table>
<thead>
<tr>
<th>Name</th>
<th>Arduino Uno</th>
<th>Nogs Nucleo</th>
<th>Nogs Stamp</th>
<th>piNogs</th>
<th>Raspberry Pi</th>
<th>BeagleBone Black</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Model</strong></td>
<td>R3</td>
<td>CN1</td>
<td>CN1</td>
<td>CN1</td>
<td>Model B</td>
<td>REV A6</td>
</tr>
<tr>
<td><strong>Price</strong></td>
<td>25 €</td>
<td>35 €</td>
<td>50 €</td>
<td>180 €</td>
<td>35 €</td>
<td>50 €</td>
</tr>
<tr>
<td><strong>Size</strong></td>
<td>68.6 x 53.3 mm</td>
<td>68.6 x 53.3 mm</td>
<td>56 x 38 mm</td>
<td>31.4 x 31.4 mm</td>
<td>85.6 x 53.98 mm</td>
<td>86 x 53 mm</td>
</tr>
<tr>
<td><strong>Processor</strong></td>
<td>ATMega 328</td>
<td>ARM Cortex-M4</td>
<td>ARM Cortex-M4</td>
<td>ARM Cortex-M4</td>
<td>ARM 11</td>
<td>ARM Cortex-A8</td>
</tr>
<tr>
<td><strong>Clock Speed</strong></td>
<td>16 MHz</td>
<td>120 MHz</td>
<td>120 MHz</td>
<td>180 MHz</td>
<td>700 MHz</td>
<td>1 GHz</td>
</tr>
<tr>
<td><strong>RAM</strong></td>
<td>2 KB</td>
<td>160 KB</td>
<td>160 KB</td>
<td>256 KB</td>
<td>512 MB</td>
<td>2 GB</td>
</tr>
<tr>
<td><strong>Flash</strong></td>
<td>32 KB</td>
<td>1024 KB</td>
<td>1024 KB</td>
<td>2048 KB</td>
<td>2 GB</td>
<td></td>
</tr>
<tr>
<td><strong>Storage</strong></td>
<td>1 KB ERROM</td>
<td>Micro SD</td>
<td>Micro SD</td>
<td></td>
<td>SD Card</td>
<td>Micro SD</td>
</tr>
<tr>
<td><strong>Encryption</strong></td>
<td>RNG, 72bit serial, OTP</td>
<td>RNG, 72bit serial, OTP</td>
<td>RNG, 72bit serial, OTP</td>
<td>RNG, 72bit serial, OTP</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Input Voltage</strong></td>
<td>7-12 V</td>
<td>5 V Micro USB Host</td>
<td>5 V</td>
<td>5 V</td>
<td>5 V Micro USB Host</td>
<td>Jack</td>
</tr>
<tr>
<td><strong>Min Power</strong></td>
<td>42 mA (0.3W)</td>
<td>150 mA (0.75 W)</td>
<td>150 mA (0.75 W)</td>
<td>120 mA (0.75 W)</td>
<td>700 mA (3.5W)</td>
<td>170 mA (0.85W)</td>
</tr>
<tr>
<td><strong>Digital GPIO</strong></td>
<td>14</td>
<td>14</td>
<td>14</td>
<td></td>
<td>8</td>
<td>65</td>
</tr>
<tr>
<td><strong>Analog Input</strong></td>
<td>6 10-bit</td>
<td>6 10-bit</td>
<td>6 10-bit</td>
<td></td>
<td>N/A</td>
<td>7.10-bit</td>
</tr>
<tr>
<td><strong>PWM</strong></td>
<td>6</td>
<td>3</td>
<td>3</td>
<td></td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td><strong>TWI/I2C</strong></td>
<td>2</td>
<td>1</td>
<td>1</td>
<td></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td><strong>SPI</strong></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td><strong>UART</strong></td>
<td>1</td>
<td>3</td>
<td>4</td>
<td></td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td><strong>RTOS</strong></td>
<td>Arduino</td>
<td>Clever Node</td>
<td>Clever Node</td>
<td>Clever Node</td>
<td>Linux etc.</td>
<td>Android, Linux etc.</td>
</tr>
<tr>
<td><strong>DEV IDE</strong></td>
<td>Arduino Tool</td>
<td>Lua, ZeroBrane Studio Debugger &amp; Live Coding, Arduino Tool</td>
<td>Lua, ZeroBrane Studio Debugger &amp; Live Coding</td>
<td>Lua, ZeroBrane Studio Debugger &amp; Live Coding</td>
<td>IDLE, Scratch, Squeak/Linux</td>
<td>Python, Scratch, Squeak, Cloud9/Linux</td>
</tr>
<tr>
<td><strong>Ethernet</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10/100</td>
<td>10/100</td>
</tr>
<tr>
<td><strong>USB Master</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2 USB 2.0</td>
<td>10/100</td>
</tr>
<tr>
<td><strong>Video Output</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>HDMI, Composite</td>
<td>USB 2.0</td>
</tr>
<tr>
<td><strong>Audio Output</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>HDMI, Analog</td>
<td>Micro HDMI</td>
</tr>
<tr>
<td><strong>Interfaces</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CSI, LCD</td>
<td>N/A</td>
</tr>
<tr>
<td><strong>Expansions</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Expansion Boards</td>
<td>CAN, LCD</td>
</tr>
<tr>
<td><strong>Nogs Integration</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Arduino Shields</td>
<td>BeagleBoard Cape</td>
</tr>
<tr>
<td><strong>PN for Arduino</strong></td>
<td></td>
<td>CN Bare Metal</td>
<td>CN Bare Metal</td>
<td>CN Bare Metal</td>
<td>CN.O, PN*.O</td>
<td>SN.APk, CN.APk, PN*.APk, CN.0, PN*.O</td>
</tr>
</tbody>
</table>
Outlook

- **Economics**
  Apps for embedded as a market
  Sensor data as a service

- **Artificial Intelligence**
  Fuzzy Logic
  Semantic Networks (JSON-LD)
  Neuronal Networks
  Collaborative Intelligence

- **New Hardware**
  CN SoC
  Lua VM on FPGA
  Native Lua CPU?
Nogs Wrap Up

• Dynamic coding of smart objects on the fly

• Transferring objects instead of data as a new communication paradigm

• Unified communication by nesting & wrapping

• Simplified commissioning & visualisation by using avatars & live coding

• Rapid product development by open hardware & software.
Join the Club ...
Thank you - Questions?

Nogs GmbH

in October 2014:

www.nogs.info
info@nogs.info

andre.riesberg@nogs.info

Have a look at GitHub soon