

mROSチュートリアル

的なサムシング

京都大学

森 智也

mori@lab3.kuis.kyoto-u.ac.jp



mROS のセッティングアップ

mROSの設定：Ubuntuの準備

- Ubuntuのほうでターミナルを4つ立ち上げます



で分割できるターミナルが立ち上がります

- Ctr-eで水平分割, Ctr-oで垂直分割できます
- こんな感じ↓

```
mori@mori-PC-GN246Y3G6:~$ cat /etc/network/interfaces
UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
RX packets:17393  errors:0  overruns:0  frame:0
TX packets:8759  errors:0  overruns:0  carrier:0
 collisions:0  TX queue length:1000
RX bytes:17822823 (17.8 MB)  TX bytes:1722382 (1.7 MB)

mori@mori-PC-GN246Y3G6:~$ sudo ifup eth2
[sudo] password for mori:
Cannot find device "eth2"
Failed to bring up eth2.
mori@mori-PC-GN246Y3G6:~$
```

```
mori@mori-PC-GN246Y3G6:~$ sudo gt
gtbl          gtk-launch
gttester      gtk-update-icon-cache
gttester-report  gtk-update-icon-cache-3.0
gttf          gtk-window-decorator
gtk-builder-convert  gtkterm
mori@mori-PC-GN246Y3G6:~$ sudo gtkterm
[sudo] password for mori:

(gtkterm:6462): IBUS-WARNING **: The owner of /home/mori/.config/ibus/bus is not root!
Control signals read: 入力/出力エラーです
Control signals read: 入力/出力エラーです
mori@mori-PC-GN246Y3G6:~$ sudo gtkterm

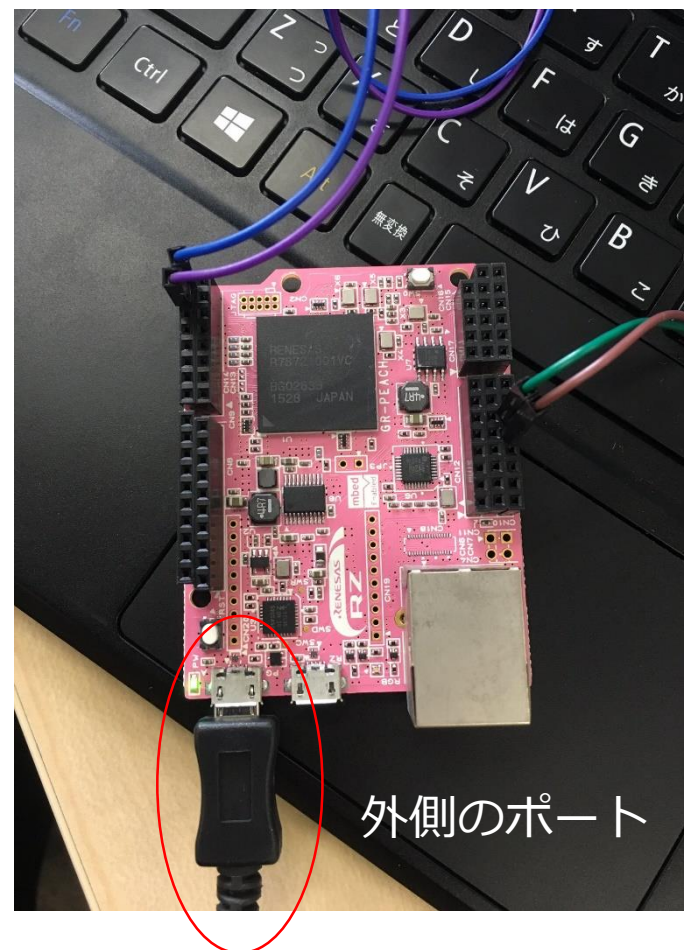
(gtkterm:6549): IBUS-WARNING **: The owner of /home/mori/.config/ibus/bus is not root!
mori@mori-PC-GN246Y3G6:~$ sudo gtkterm

(gtkterm:6559): IBUS-WARNING **: The owner of /home/mori/.config/ibus/bus is not root!
$
```

mROSの設定：機器接続&IP設定

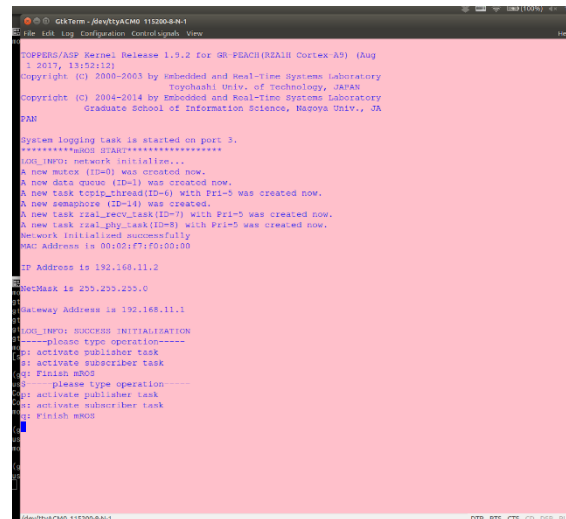
- もろもろの接続を行います
 1. ルータを起動
 2. gr-peachとルータを接続
 3. gr-peachとPCをUSBで接続
 4. PCとルータを接続
 5. PCのIPとgr-peachのIPを設定

ros_emb.cpp:19
xml_call.cpp:148



mROSの設定：実行コマンド

- mROSの実行方法
- 別ターミナルでそれぞれ以下のコマンドを入力
 - \$roscore ROSマスタ実行
 - \$roslaunch mros_test mros_talker パブリッシュノード起動
 - \$roslaunch mros_test mros_listner サブスクライブノード起動
 - \$sudo gtkterm シリアル通信コンソール起動
- ~~gtktermのConfiguration->Load configuration->mros
でピンクになります~~



```
GTKTerm: /dev/ttyCH06 115200-8-N-1
File Edit Log Configuration Control Window View Help

TOPPERS/ACP Kernel Release 1.9.2 For GE PEARCH (REAL Cortex-A5) (Aug
 3 2017, 13:52:12)
Copyright (C) 2000-2003 by Embedded and Real-time Systems Laboratory
  (Yokohashi Univ. of Technology, JAPAN)
Copyright (C) 2004-2014 by Embedded and Real-time Systems Laboratory
  Graduate School of Information Science, Nagoya Univ., JA
PAN

System logging task is started on port 3.
*****MROS START*****
[00] MROS network initialization...
A new mutex (ID=0) was created now.
A new data queue (ID=1) was created now.
A new task tcpip_thread(ID=6) with Pri=5 was created now.
A new semaphore (ID=14) was created.
A new task real_time_task(ID=7) with Pri=5 was created now.
A new task real_time_task(ID=8) with Pri=5 was created now.
Network initialized successfully
MAC Address is 08:02:7F:FD:08:08
IP Address is 192.168.11.2
NetMask is 255.255.255.0
Gateway Address is 192.168.11.1
[00] MROS SUCCESS INITIALIZATION
-----please type operation-----
p: activate publisher task
s: activate subscriber task
q: Finish MROS
? -----please type operation-----
p: activate publisher task
s: activate subscriber task
q: Finish MROS
```

mROSの設定 : mROS側

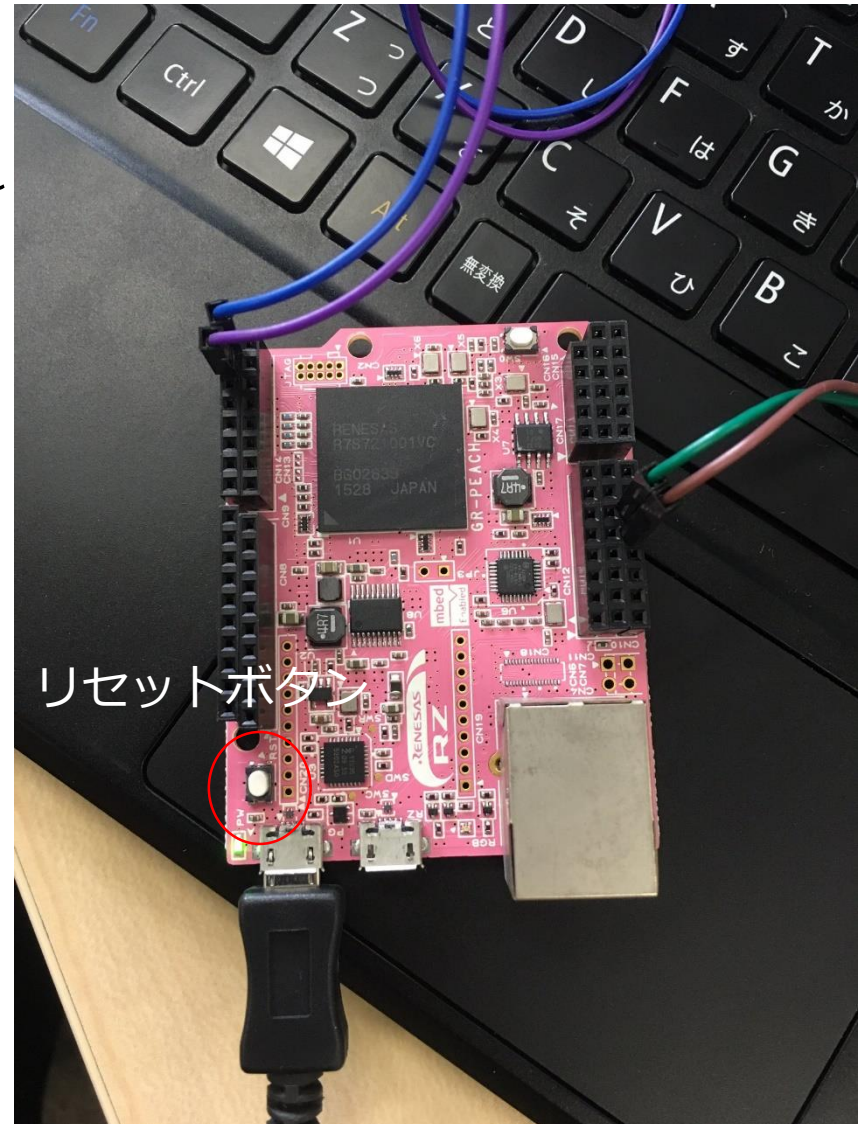
- 接続が完了している場合, gr-peachのリセットボタンを押すことでプログラムが実行されます
- シリアルコンソールにネットワーク初期化が走って, IPが表示されます (IPの確認はここで)

```
TOPPERS/ASP Kernel Release 1.9.2 for GR-PEACH(RZA1H Cortex-A9) (Aug
1 2017, 13:52:12)
Copyright (C) 2000-2003 by Embedded and Real-Time Systems Laboratory
Toyohashi Univ. of Technology, JAPAN
Copyright (C) 2004-2014 by Embedded and Real-Time Systems Laboratory
Graduate School of Information Science, Nagoya Univ., JA
PAN

System logging task is started on port 3.
*****mROS START*****
LOG_INFO: network initialize...
A new mutex (ID=0) was created now.
A new data queue (ID=1) was created now.
A new task tcpip_thread(ID=6) with Pri=5 was created now.
A new semaphore (ID=14) was created.
A new task rzal_recv_task(ID=7) with Pri=5 was created now.
A new task rzal_phy_task(ID=8) with Pri=5 was created now.
Network Initialized successfully
MAC Address is 00:02:f7:f0:00:00

IP Address is 192.168.11.2
NetMask is 255.255.255.0
Gateway Address is 192.168.11.1

LOG_INFO: SUCCESS INITIALIZATION
t-----please type operation-----
p: activate publisher task
s: activate subscriber task
q: Finish mROS
s$-----please type operation-----
p: activate publisher task
s: activate subscriber task
q: Finish mROS
```

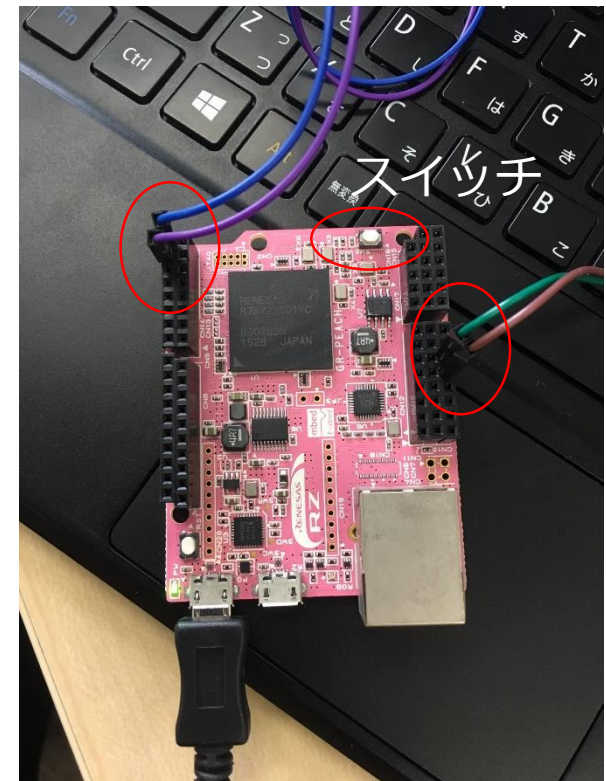


mROSの設定：mROS側

- シリアルコンソールの入力に応じてタスクが起動します
 - ~~p~~：パブリッシャータスクが起動
 - ~~s~~：サブスクライバタスクが起動
 - ~~q~~：mROSを終了します

※通信させるROSノードを先に起動させておいてください
変なことになるかもしれません

- パブリッシャータスクではセンサを使用します
- 右のようにピンを刺してください
 - 青：Echo-左上 紫：Trig-青の下
 - 緑：Gnd-2段目右上から下に3つめ
 - 茶：Vcc緑の下
- スイッチを押すとパブリッシュの実行/停止が可能です



mROSの設定：動作確認

- mROSサブスクライバの確認
 - mros_talkerで文字を入力, 入力に応じてLEDが光る
red,blue,greenに対応. 二度入力しても消えない
消すときはresetを入力
- mROSパブリッシャの確認
 - mros_listnerでセンサデータが表示される
- mROSを終了して再度立ち上げる際はROSマスタ, ROSノードも再起動するgtktermはそのまま大丈夫

以上, 動いてるものは自分の目で確かめてください

Let's mROS!

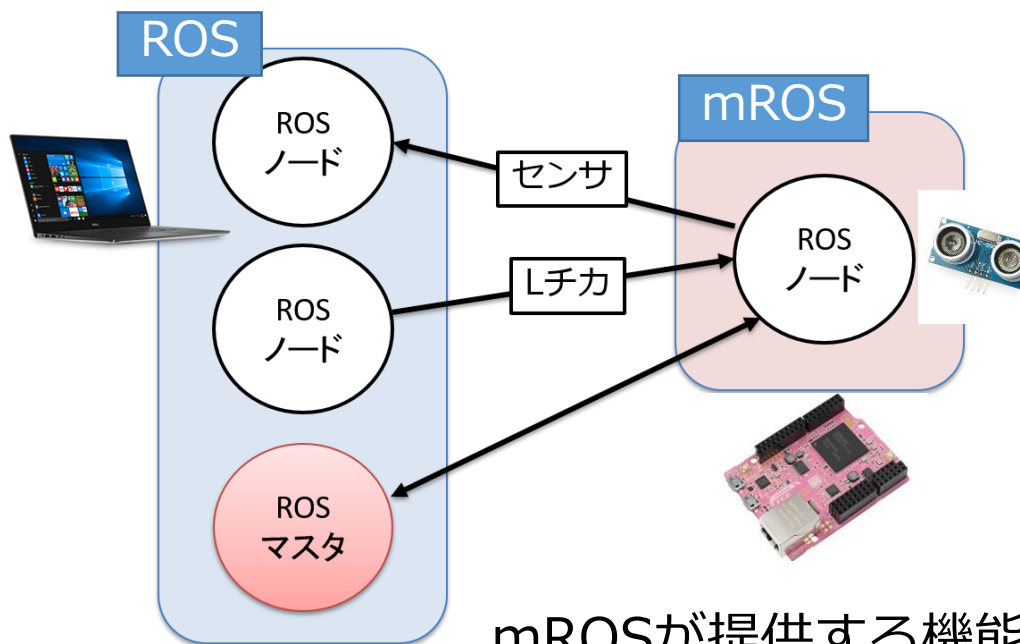
おまけ

mROSについての説明

概要

組み込みマイコンでROSノードが実行できる！

Linux上でしか動かなかったROSが組み込みマイコンで実行可能！
ROSの通信プロトコルをサポートすることで実現！



TOPPERS/ASPカーネル
+
mbedライブラリ (LWIP)
+
mROS通信ライブラリ
||
**ROSコンポーネント
軽量実行環境**

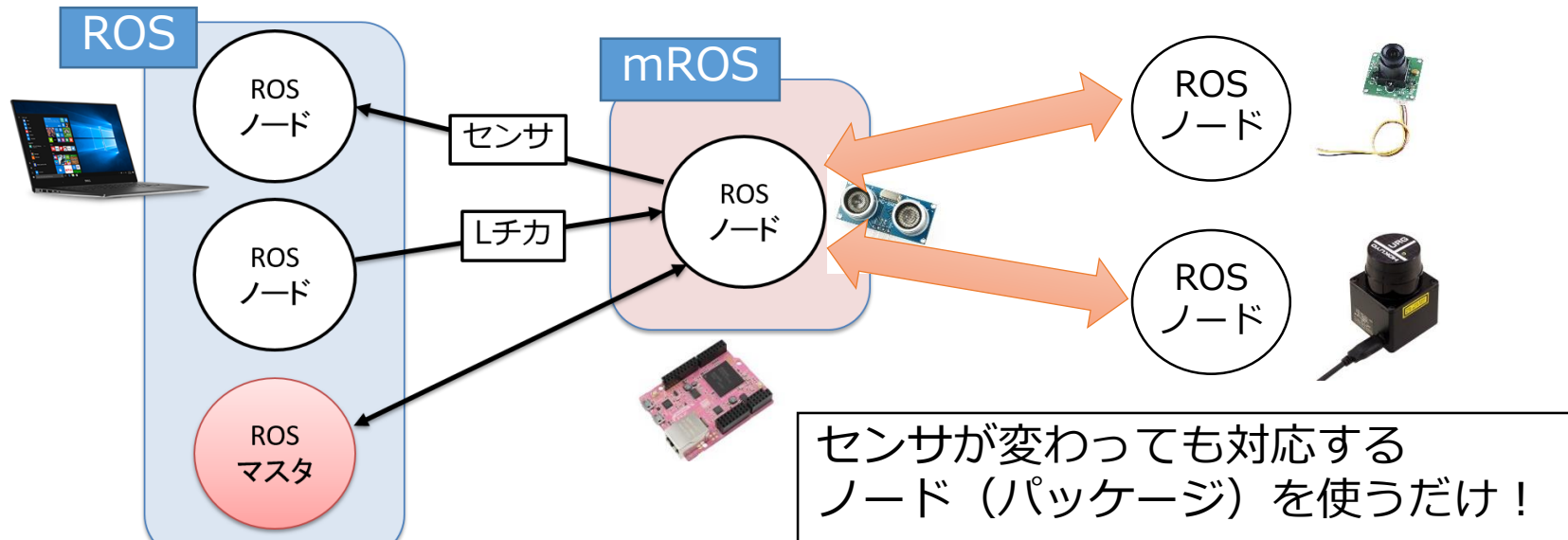
mROSが提供する機能
✓ ROSマスタとの通信
✓ ノード間でのpub/sub通信

ROSの関数を使って
プログラミング可能

概要

組み込みマイコンでROSノードが実行できる！

オープンソースで公開されている既存のパッケージが簡単に使える！

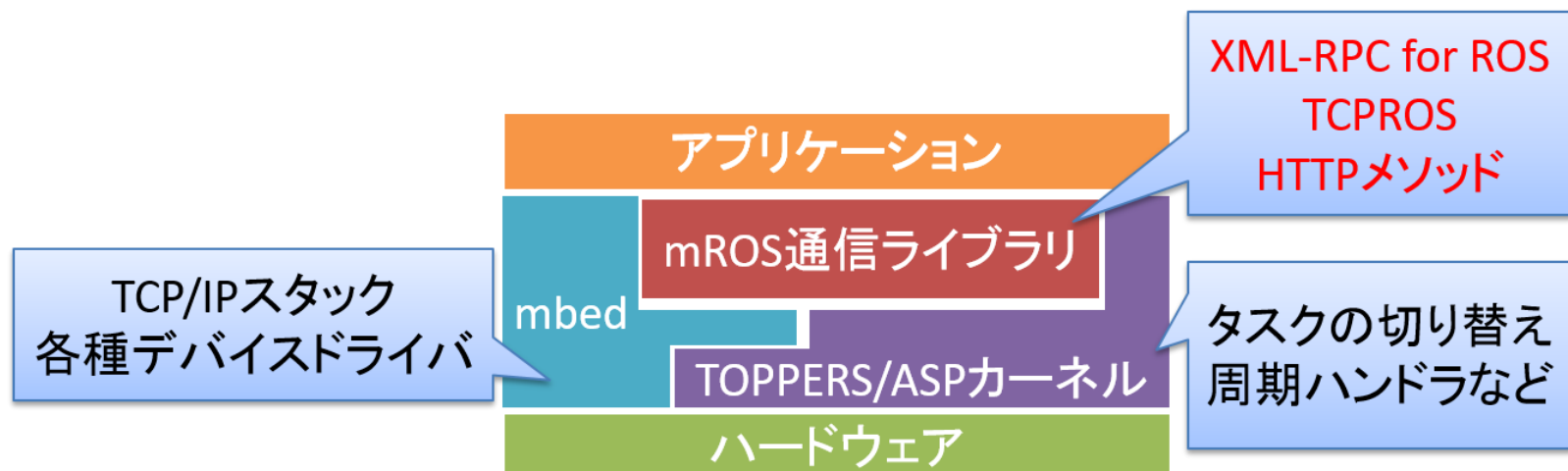


本日のデモ：

- ✓ PCのROSノードから出版されたLチカ命令をmROSが購読して組み込みデバイスのLEDを光らせる（サブスクライバー）
- ✓ mROSノードが超音波センサが測定した値を出版してROSノードが購読し、標準出力する（パブリッシャー）

mROSの構成

- mROS通信ライブラリ
 - アプリケーションに対してROSの通信をサポート
 - ROS仕様の関数を再定義
- mbed
 - LWIPスタックやLEDなどのデバイスドライバを含む
- TOPPERS/ASPカーネル
 - μ ITRON仕様のリアルタイムOS



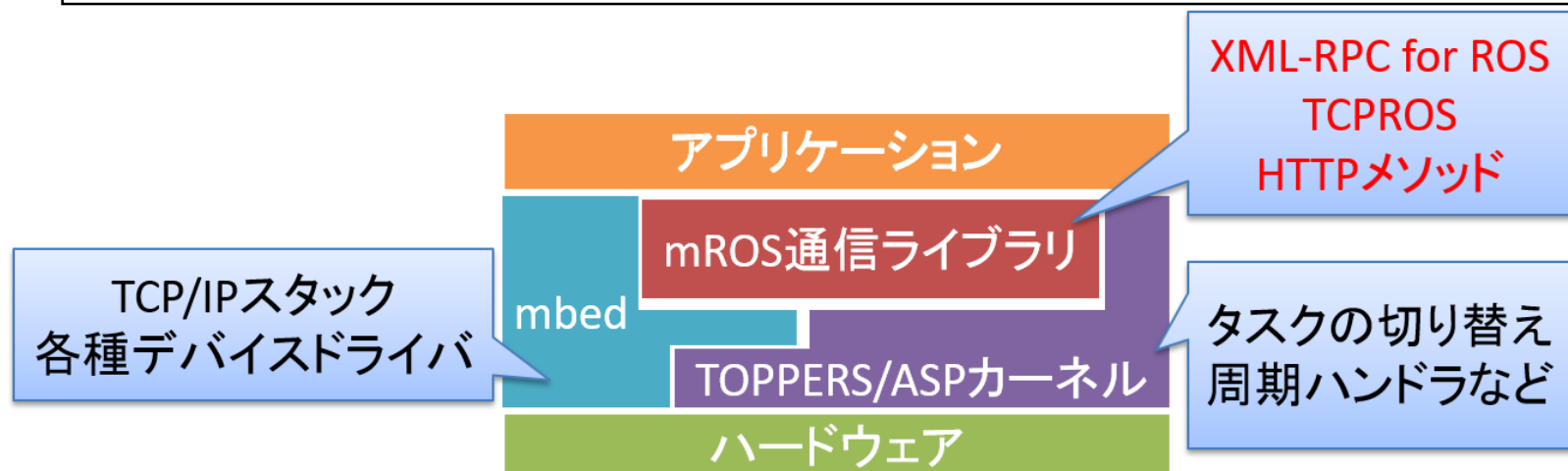
mROSの構成

- アプリケーション
 - ユーザが定義するプログラム
 - ROSプログラムで使用するROS特有の関数が使用可能

アプリケーションは組込み記述を気にせずにROS記述可能！

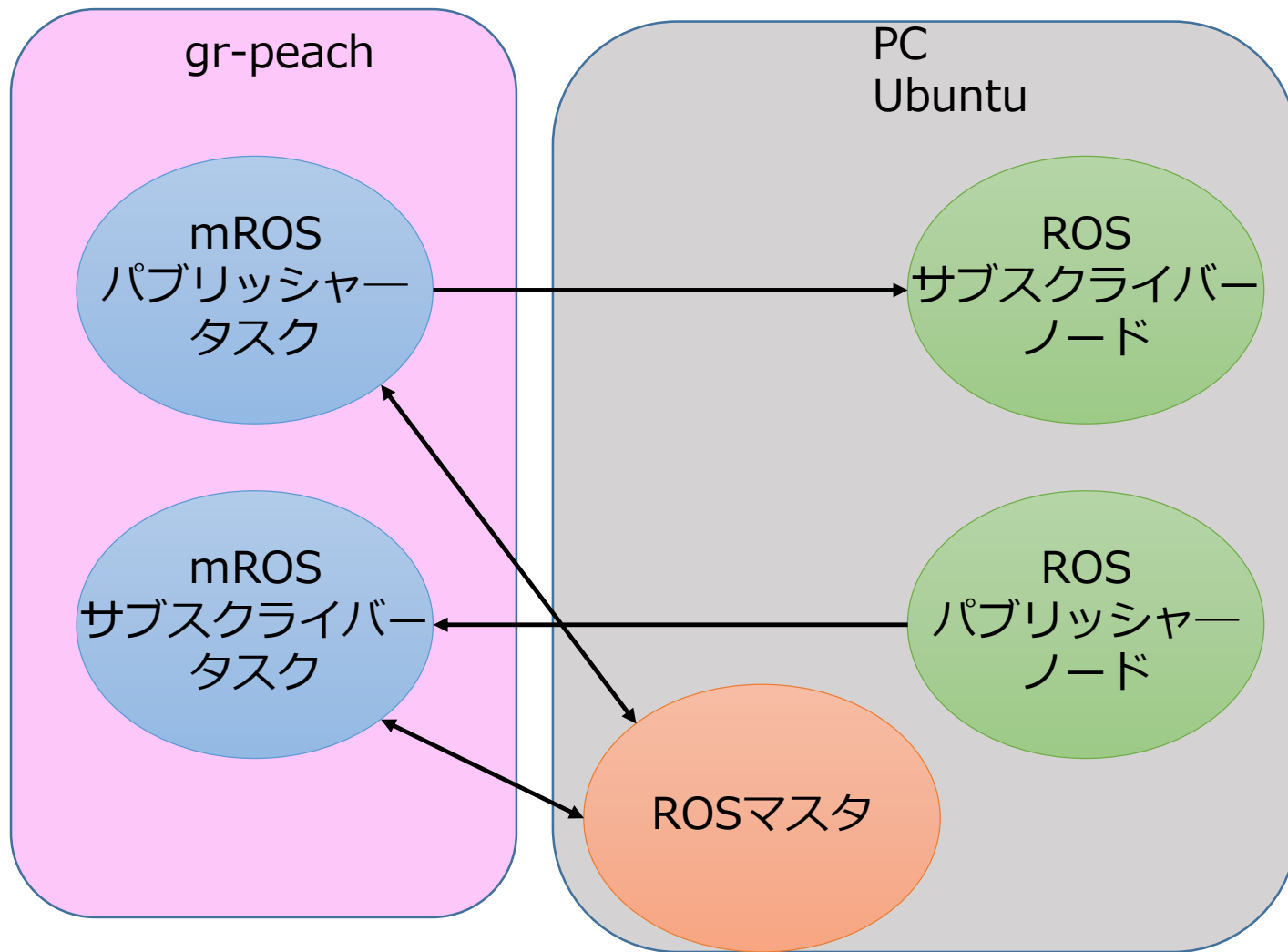
or

ITRON仕様の設計を行うことでリアルタイム性の確保も容易！



mROSの実行タスク

- 今回のmROSはノードをタスクとして実装
 - パブリッシャーノード, サブスクライバーノードがタスク化



タスク内の処理

- パブリッシャータスクではセンサからのデータを取得して出版
 - ROSにおけるadvertise(),publish()に相当
- サブスクライバータスクでは購読したトピックに応じて処理
 - ROSにおけるsubscribe()に相当
 - 内部でコールバック関数を定義して実行
- 共通
 - メッセージタイプは両者ともstd_msgs/Stringを使用
 - XML-RPCを使用して, マスタおよびノードと通信
 - TCPROSプロトコルのデータ処理を実装

