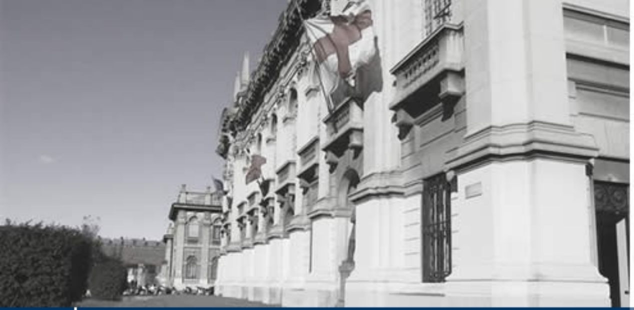


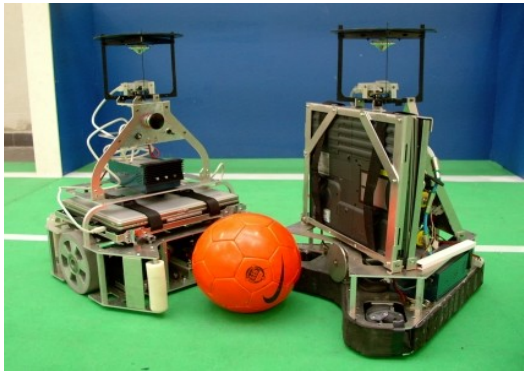
POLITECNICO DI MILANO

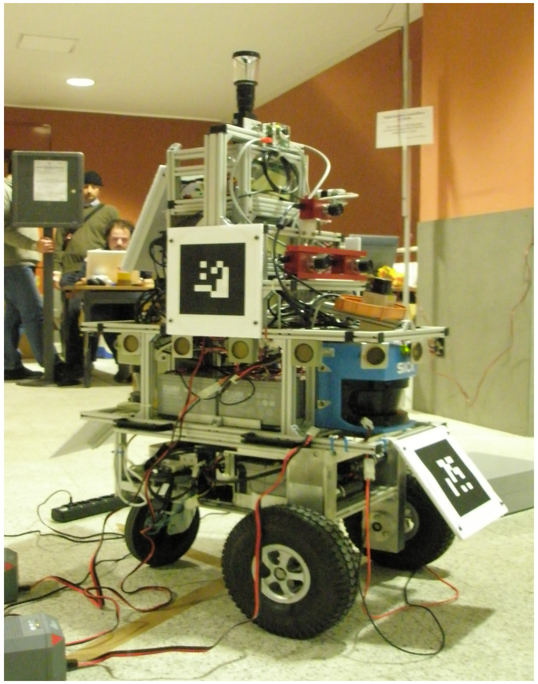
μROSnode – running ROS on microcontrollers

Martino Migliavacca, Andrea Zoppi, *Matteo Matteucci, Andrea Bonarini*
migliavacca@elet.polimi.it, andrea.zoppi@mail.polimi.it
AIRLab - Artificial Intelligence and Robotics Laboratory
Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Italy

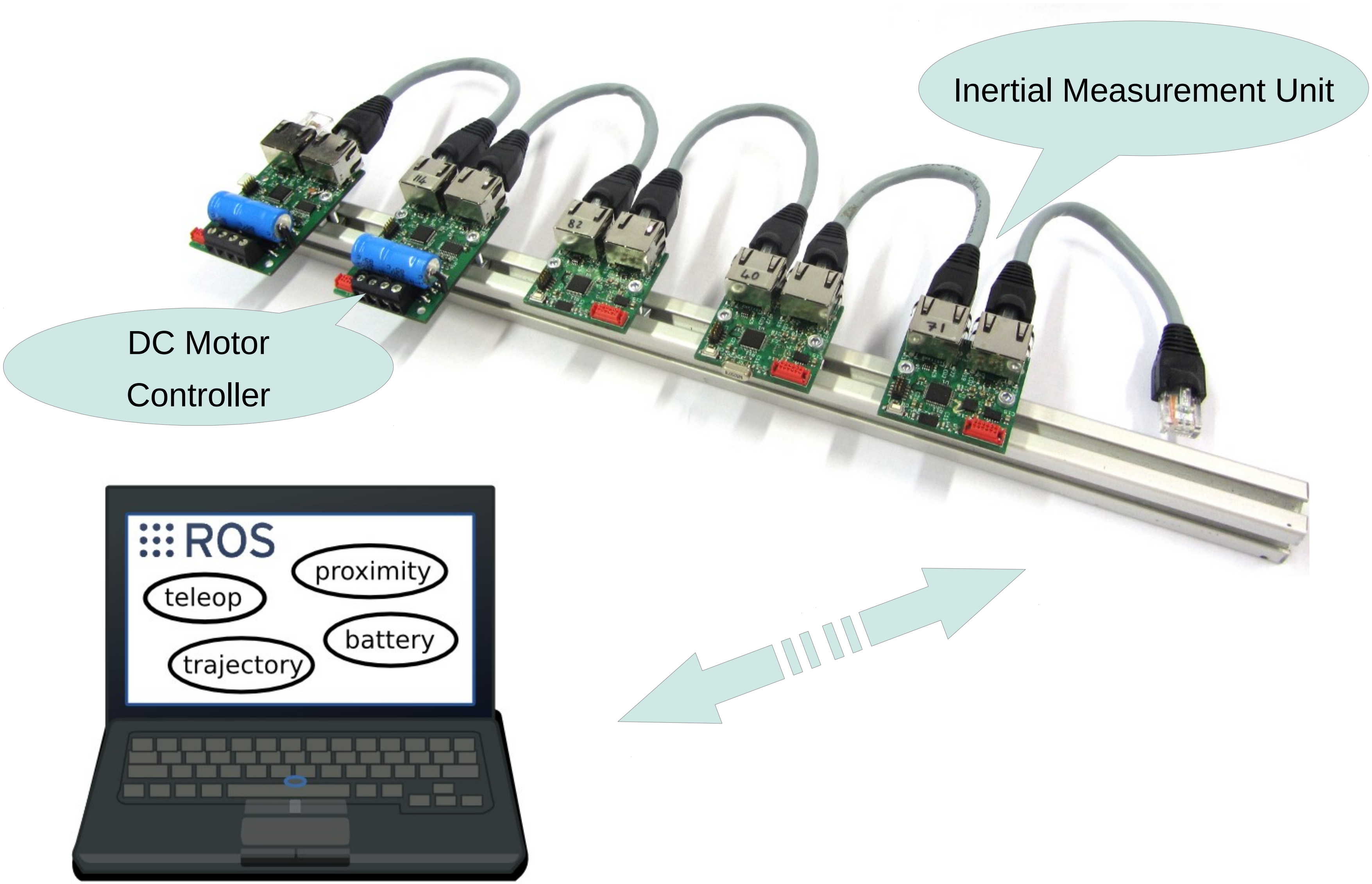
The AIRLab

- Aritifical Intelligence and Robotics laboratory active since 1973
- 11 senior researchers, 10 Phd Students, more than 60 Master theses/year
- Industrial, National and EU Projects
- Development of autonomous robots and unmanned vehicles
- Robot perception and multisensor fusion (laser, vision, inertial, GPS, etc.)

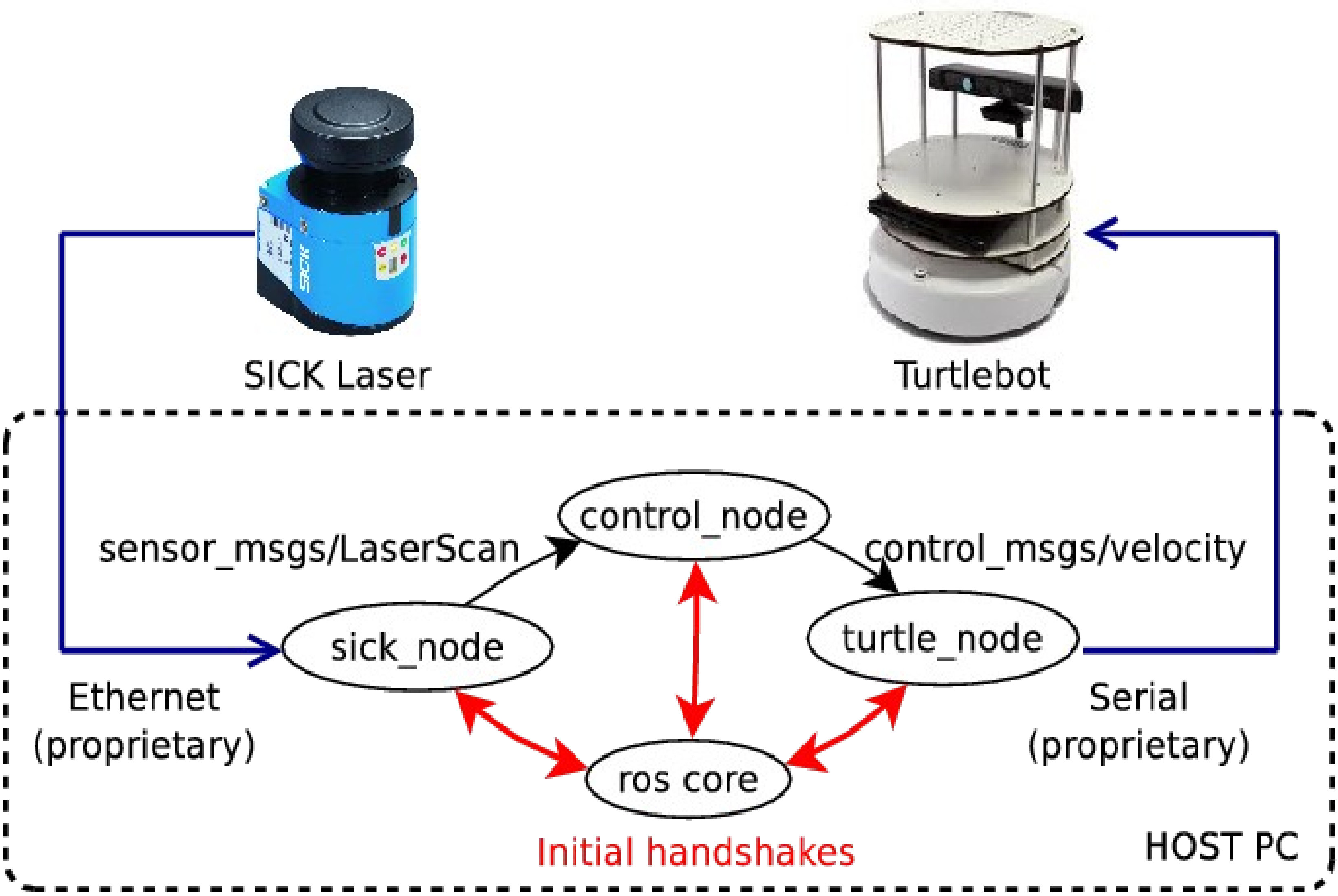




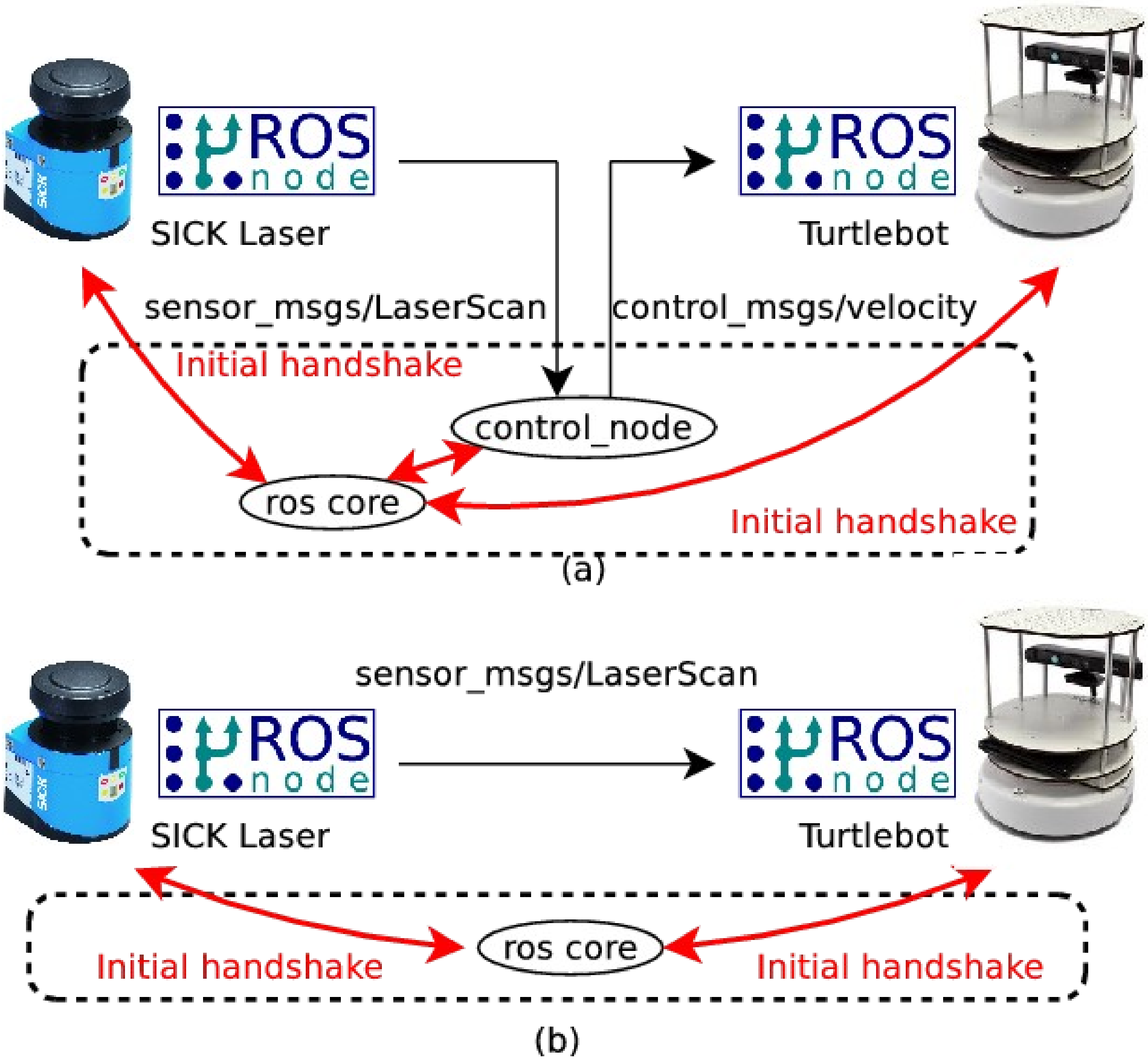
R2P: Rapid Robot Prototyping



Dedicated nodes

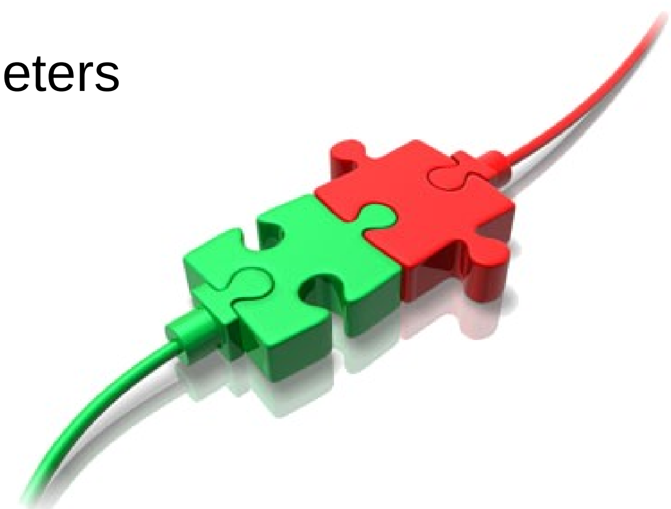


Native ROS (TCPROS) devices



In a nutshell

- Lightweight ROS client (not a cross-compiled ROS system!)
- Runs on modern 32-bit microcontrollers (ARM Cortex-M)
- ANSI C implementation
- Supported ROS features
  - XML-RPC graph management
  - TCPROS data protocol
  - Topics
  - Services
  - Global parameters



Framework architecture

- Modular, object-oriented style on top of ANSI C
- Multithreaded implementation
- ChibiOS/RT and Posix ports
- Abstraction layer to low-level libraries
- Simple integration with user firmware
  - Code generator tool
  - Topic/service handler functions

Master node communication - XML-RPC

Features

- Custom implementation
- No external dependencies
- Minimal ROS-oriented syntax
- Single-pass parsing
- On-the-fly processing

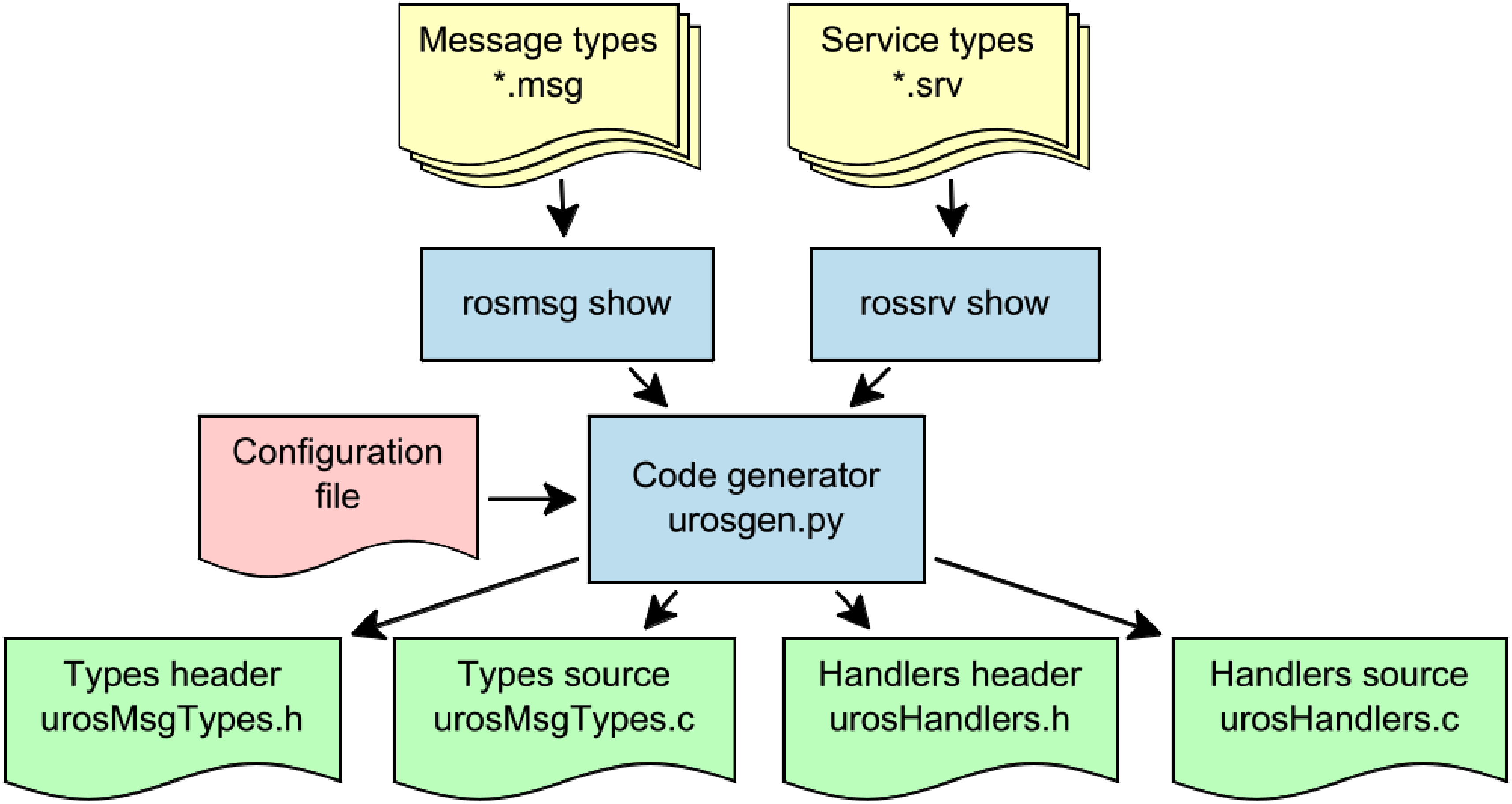
Limits

- No *gzip* support
- Size-limited unbuffered messages and strings

```
POST /RPC2 HTTP/1.1
Content-Type: text/xml
Content-Length: 178

<?xml version="1.0"?>
<methodCall>
  <methodName>getPid</methodName>
  <params>
    <param>
      <value><string>/rosnode</string></value>
    </param>
  </params>
</methodCall>
```

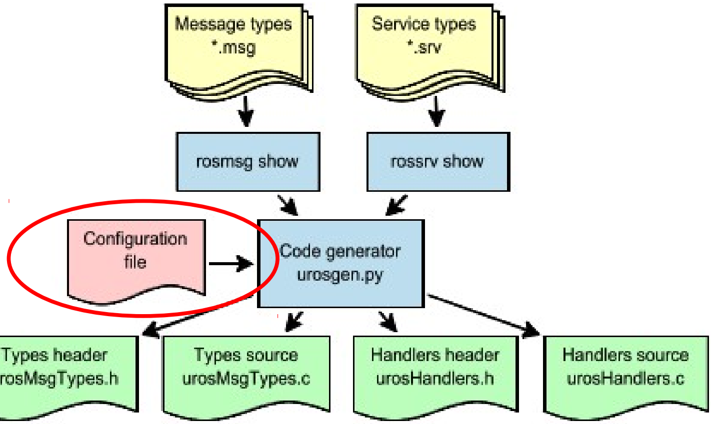
```
HTTP/1.1 200 OK
Content-Type: text/xml
Content-Length: 309

<?xml version="1.0"?>
<methodResponse>
  <params>
    <param>
      <value>
        <array>
          <data>
            <value><i4>1</i4></value>
            <value></value>
            <value><i4>6544</i4></value>
          </data>
        </array>
      </value>
    </param>
  </params>
</methodResponse>
```

Generation flow



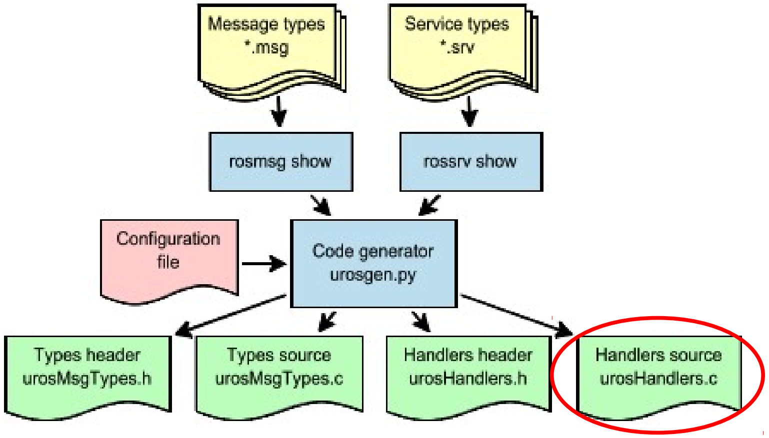
Configuration file


```
[Options]
nodeName                        = turtlesim

[PubTopics]
rosout                          = rosgraph_msgs/Log
turtleX/pose                    = turtlesim/Pose
turtleX/color_sensor            = turtlesim/Color

[SubTopics]
turtleX/command_velocity        = turtlesim/Velocity

[PubServices]
clear                           = std_srvs/Empty
kill                            = turtlesim/Kill
spawn                           = turtlesim/Spawn
turtleX/set_pen                 = turtlesim/SetPen
turtleX/teleport_absolute       = turtlesim/TeleportAbsolute
turtleX/teleport_relative       = turtlesim/TeleportRelative

[CallServices]
# none
```



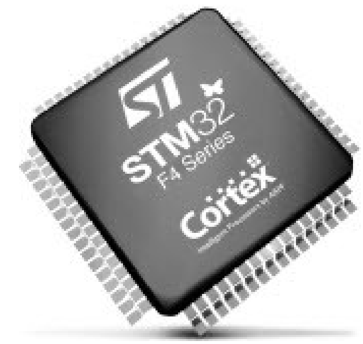
Output code

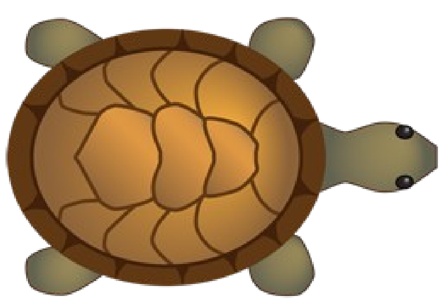

```
uros_err_t pub_tpc__turtleX__pose(UrosTcpRosStatus *tcpstp) {
  /* Message allocation and initialization.*/
  UROS_TPC_INIT_H(msg__turtlesim__Pose);

  /* Published messages loop.*/
  while (!urosTcpRosStatusCheckExit(tcpstp)) {
    /* Fill in the contents of the message.*/
    ... user code ...

    /* Send the message.*/
    UROS_MSG_SEND_LENGTH(msgp, msg__turtlesim__Pose);
    UROS_MSG_SEND_BODY(msgp, msg__turtlesim__Pose);
  }
  tcpstp->err = UROS_OK;

_finally: /* Message deinitialization and deallocation.*/
  UROS_TPC_UNINIT_H(msg__turtlesim__Pose);
  return tcpstp->err;
}
```

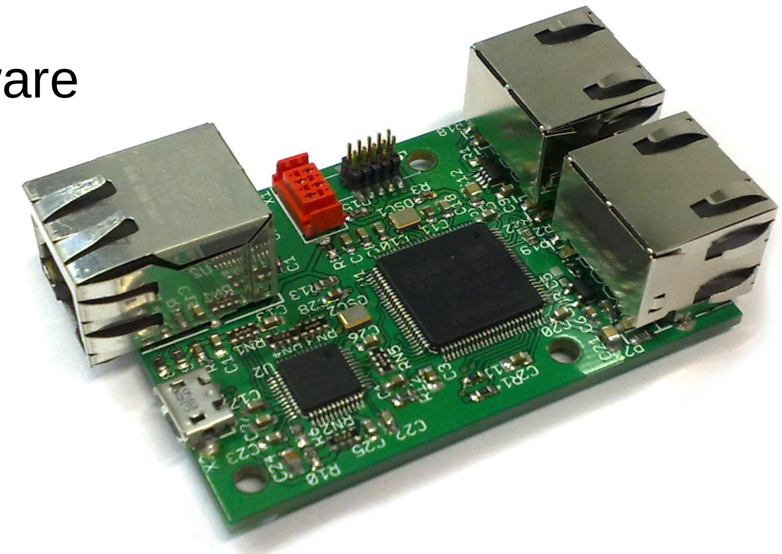


Test Platform: R2P Gateway module

- R2P—ROS Gateway module
- STM32F407 microcontroller
  - 32-bit ARM Cortex-M4 core @ 168 MHz
  - 1 MiB flash program memory
  - 192 KB RAM (112 KB main shared block)
- 100BASE-TX Ethernet port
- ChibiOS/RT 2.5.2 real-time OS
- LwIP 1.4.1 network stack
- *Turtlesim* clone as benchmark firmware





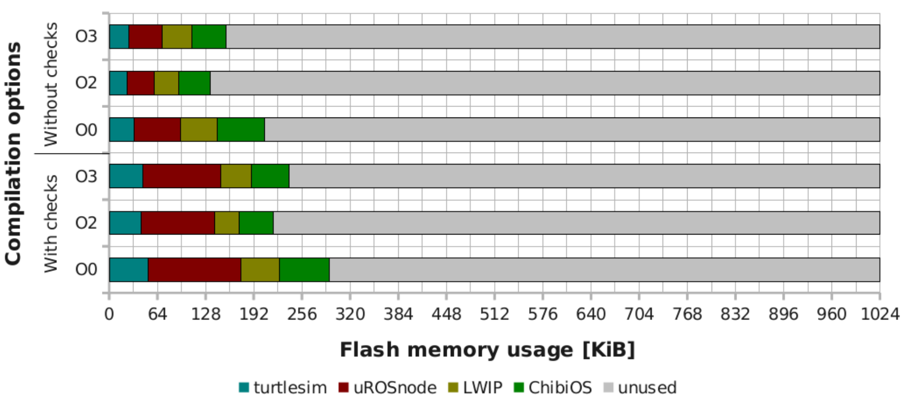


Code and memory footprint

Code footprint

- < 40 KB for µROSnode

Component	With checks [B]			Without checks [B]		
	O0	O2	O3	O0	O2	O3
turtlesim	52332	42020	45191	32895	23629	26582
µROSnode	123270	97691	103488	61285	36660	43285
LWIP	49898	33133	40820	49195	32414	40086
ChibiOS/RT	67219	45551	49613	63250	41473	44910
unused	731281	805605	784887	817375	889824	869137



Memory footprint

- < 1 KB for each topic instance

Thread/pool entry point functions	Maximum stack depth [B]	
	O0 +checks	O3 -checks
lwip_thread	952	740
main	264	776
pub_srv__clear	716	456
pub_srv__kill	1100	784
pub_srv__spawn	1244	904
pub_srv__turtleX__set_pen	740	472
pub_srv__turtleX__teleport_absolute	764	512
pub_srv__turtleX__teleport_relative	764	504
pub_tpc__rosout	708	432
pub_tpc__turtleX__color_sensor	660	400
pub_tpc__turtleX__pose	660	408
sub_tpc__turtleX__command_velocity	724	464
urosNodeThread	1188	888
urosRpcSlaveListenerThread	660	476
urosRpcSlaveServerThread	1004	616
urosTcpRosListenerThread	652	476
urosThreadPoolWorkerThread	240	148

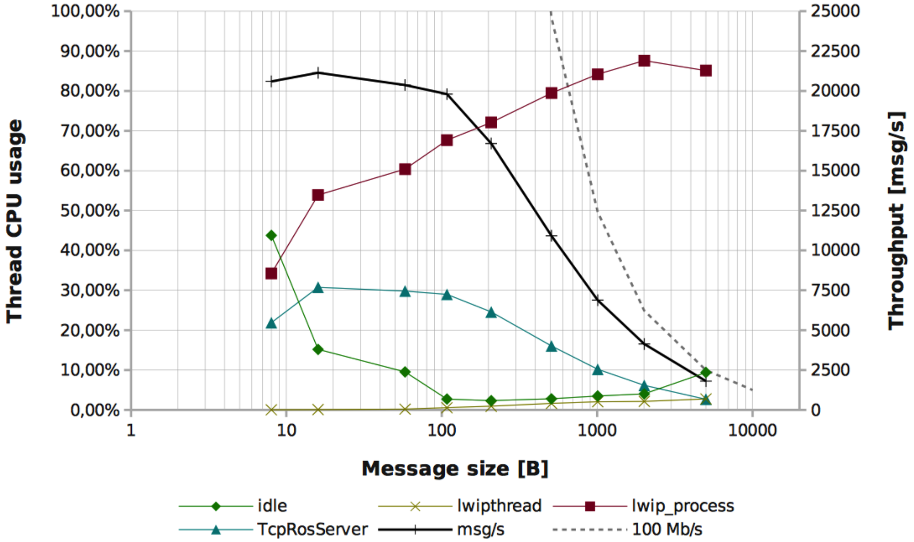
Communication Benchmark

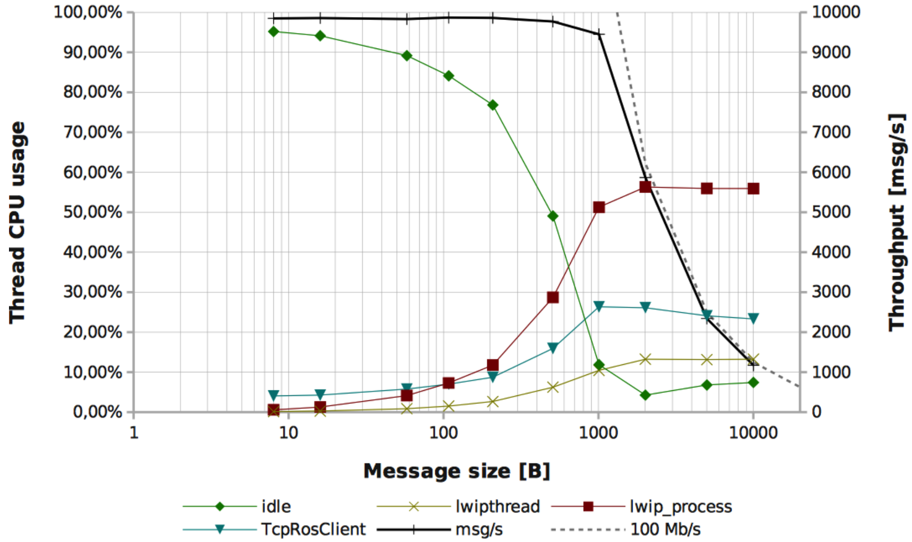
Transmission performance

- Up to 20.000 msg/s
- < 100 B limited by `rostopic hz`
- > 100 B limited by LwIP

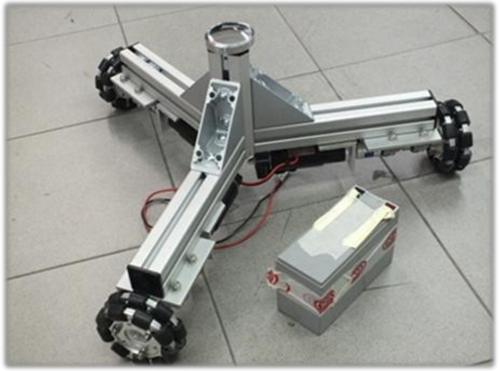
Reception performance

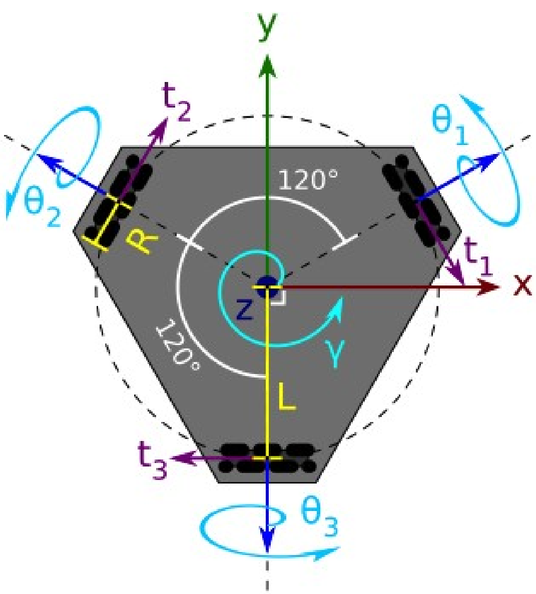
- Up to 10.000 msg/s
- < 1 KB limited by `rostopic echo`
- > 2 KB Ethernet saturation

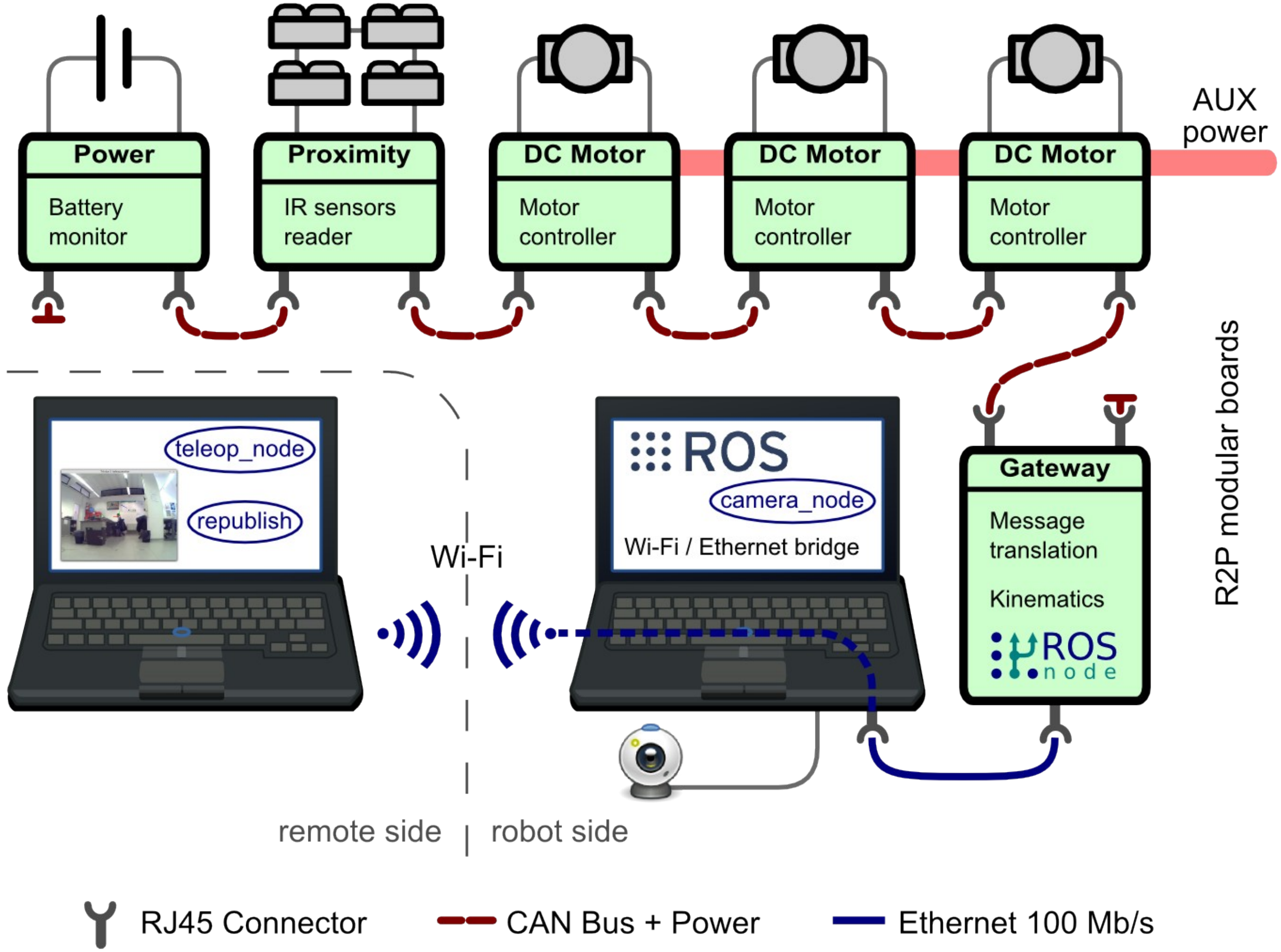




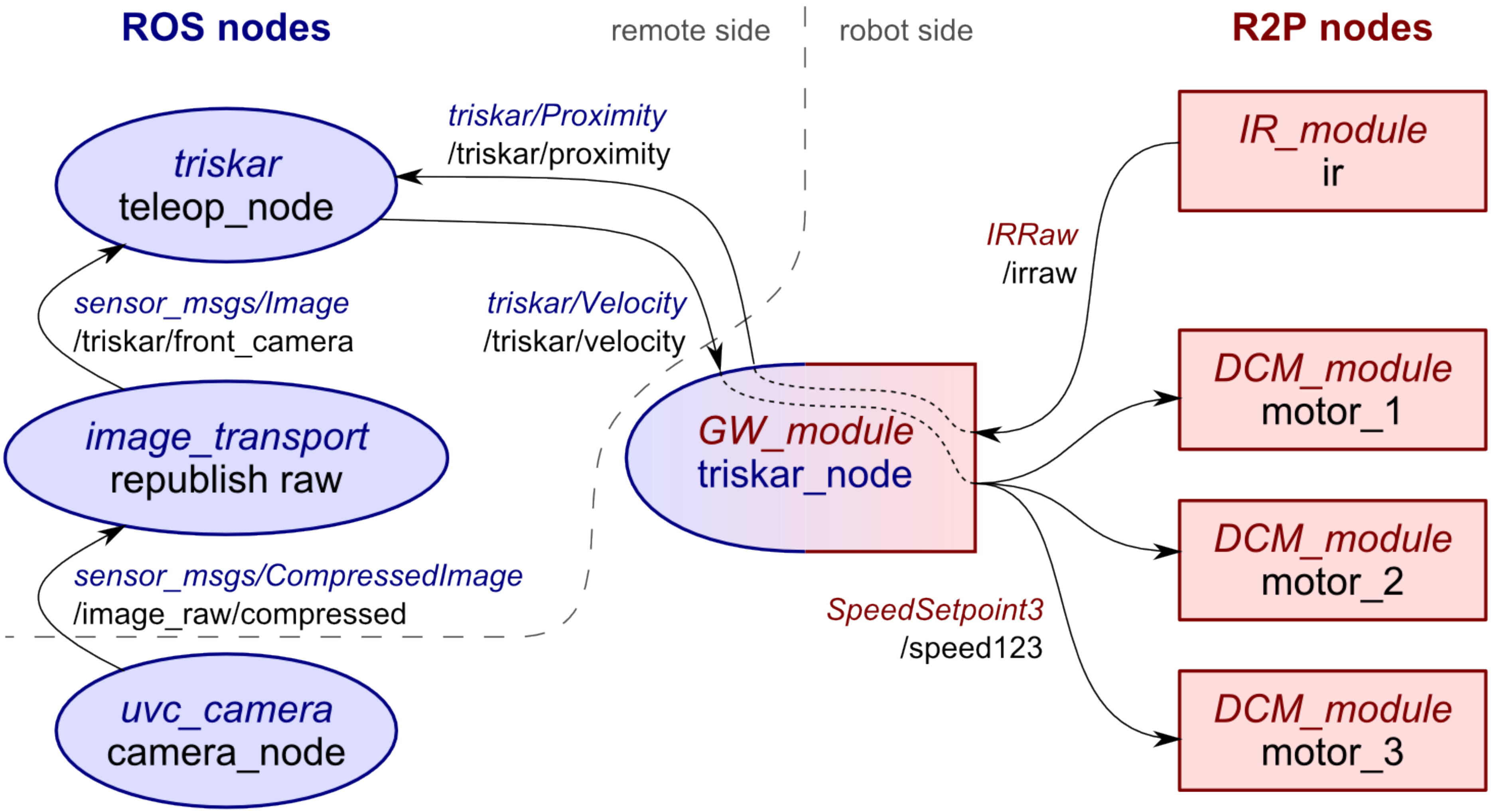
The Triskar2 Omnidirectional Robot



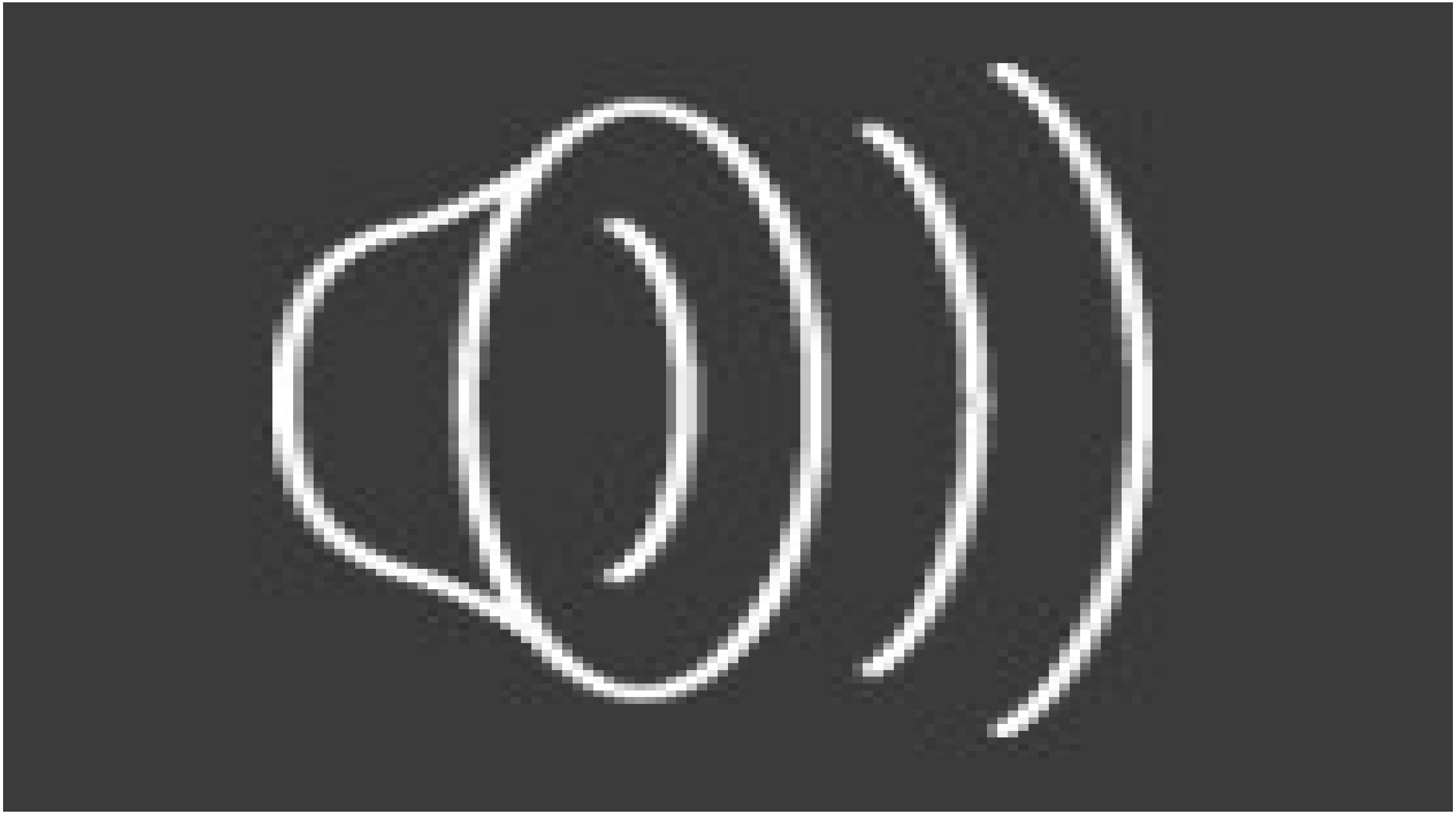




Software Architecture



Demo Time!



Conclusions

- µROSnode working implementation available
  - `https://github.com/openrobots-dev/uROSnode`
- Main ROS features supported
- Turtlesim clone demo included
- Doxygen documentation

Future work

- UDPROS support
- C++ wrappers
- Open to suggestions

Questions?

µROSnode – running ROS on microcontrollers

https://github.com/openrobots-dev/uROSnode

Martino Migliavacca, Andrea Zoppi, *Matteo Matteucci, Andrea Bonarini*
migliavacca@elet.polimi.it, andrea.zoppi@mail.polimi.it
AIRLab - Artificial Intelligence and Robotics Laboratory
Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Italy