2018.04.10

Toy OBDH system

Juan A. de la Puente
<jpuente@dit.upm.es>

Overview

- The aim of this project is to build a simple mockup of a satellite OBDH system performing basic housekeeping telemetry
  - periodic sensor sampling
  - periodic basic telemetry
  - on-request housekeeping telemetry with recent data

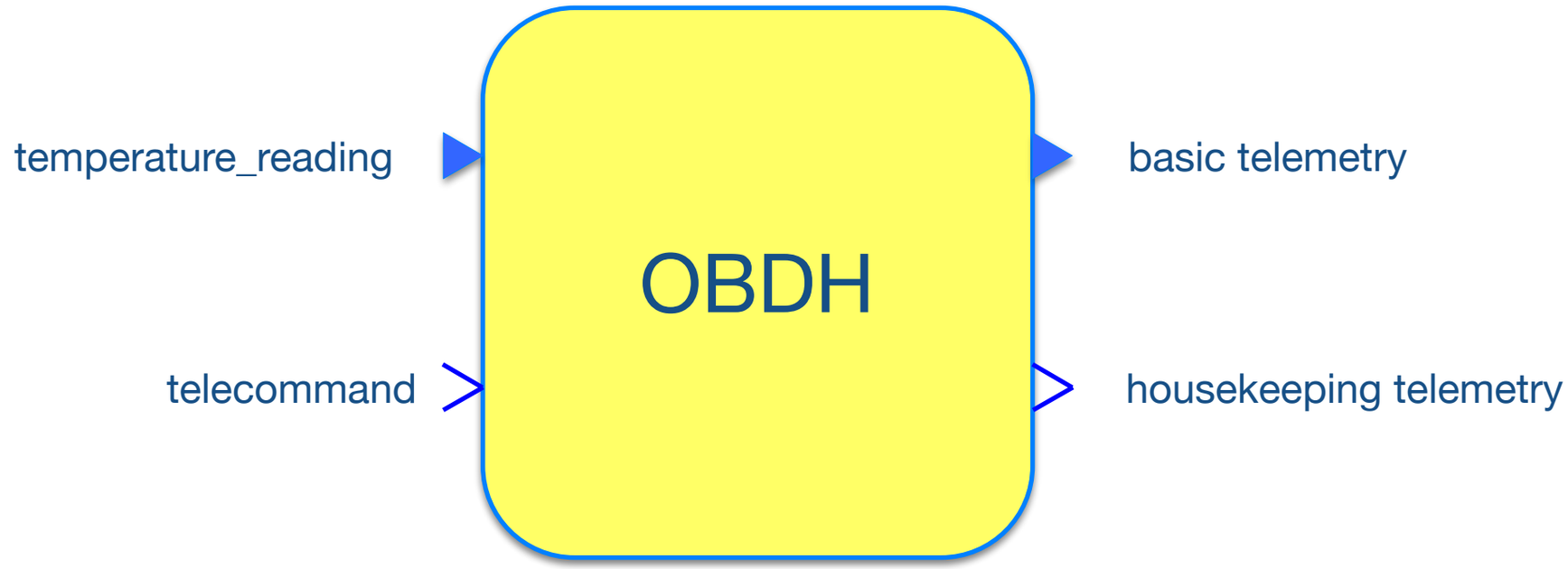
Functional requirements

- A temperature sensor is periodically sampled with period T_S
- A basic TM message is sent periodically with period T_B. The message contains the average value of the temperatures measured since the last basic TM
- The system can receive a TC from the ground station requesting a temperature report. It replies with a housekeeping TM message including the values of all temperatures stored since the last basic TM message and their respective reading times.
- TM messages are stamped with the current time.
- Time stamp values are given in seconds from the system start time, with a resolution of at least 1 ms.

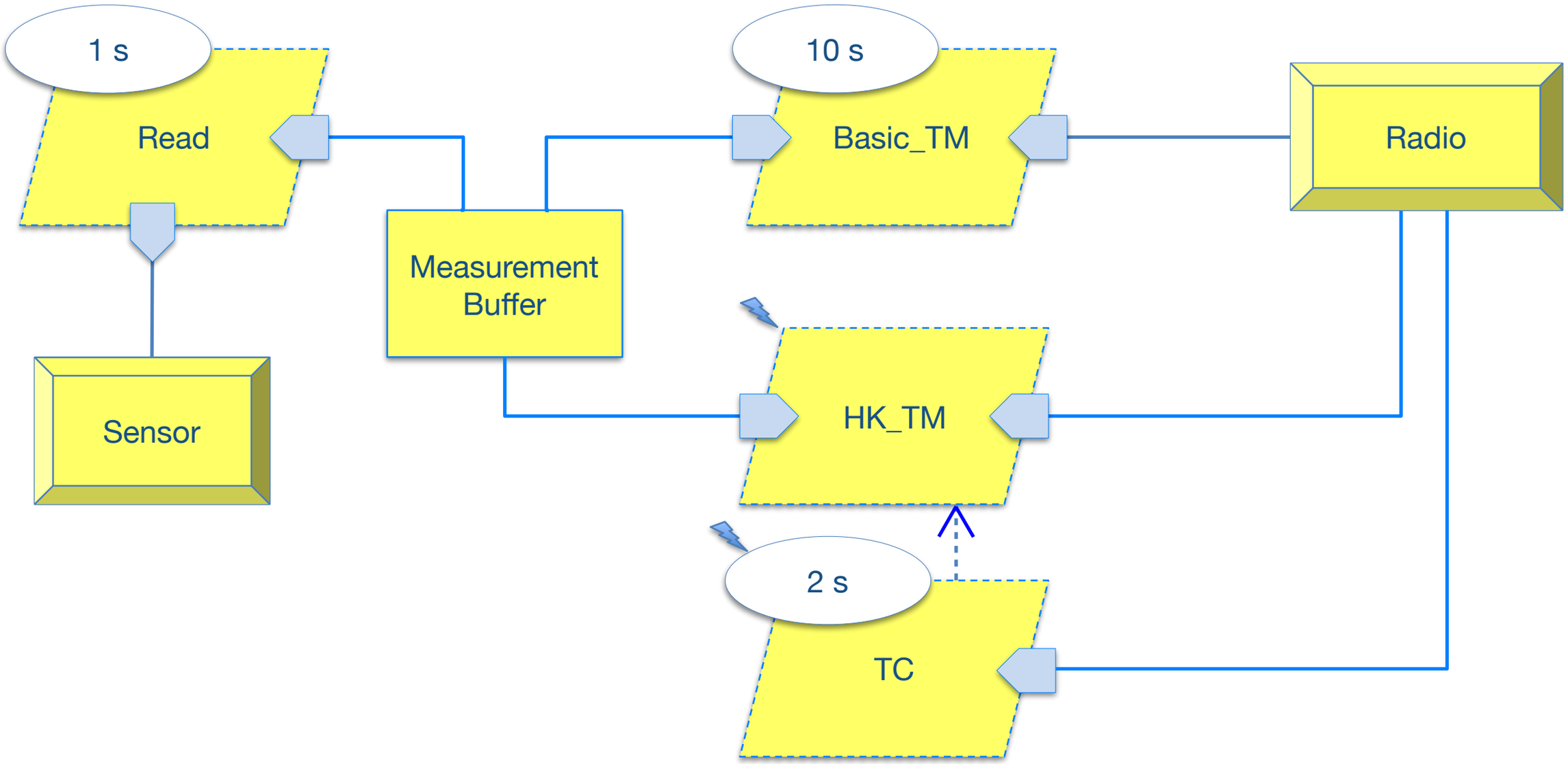
Temporal requirements

- Temperature must be sampled every T_S = 1 s
  - reading to be completed before D_S = 0.1 s
- Basic telemetry must be sent every T_B = 10 s
  - message must be sent before D_B = 0.5 s
- Telecommands are separated by at least T_C = 2 s
  - processing must be completed before D_C = 0.05 s
- Housekeeping telemetry messages are sent after reception of a TC
  - message must be sent before D_H = 0.2 s

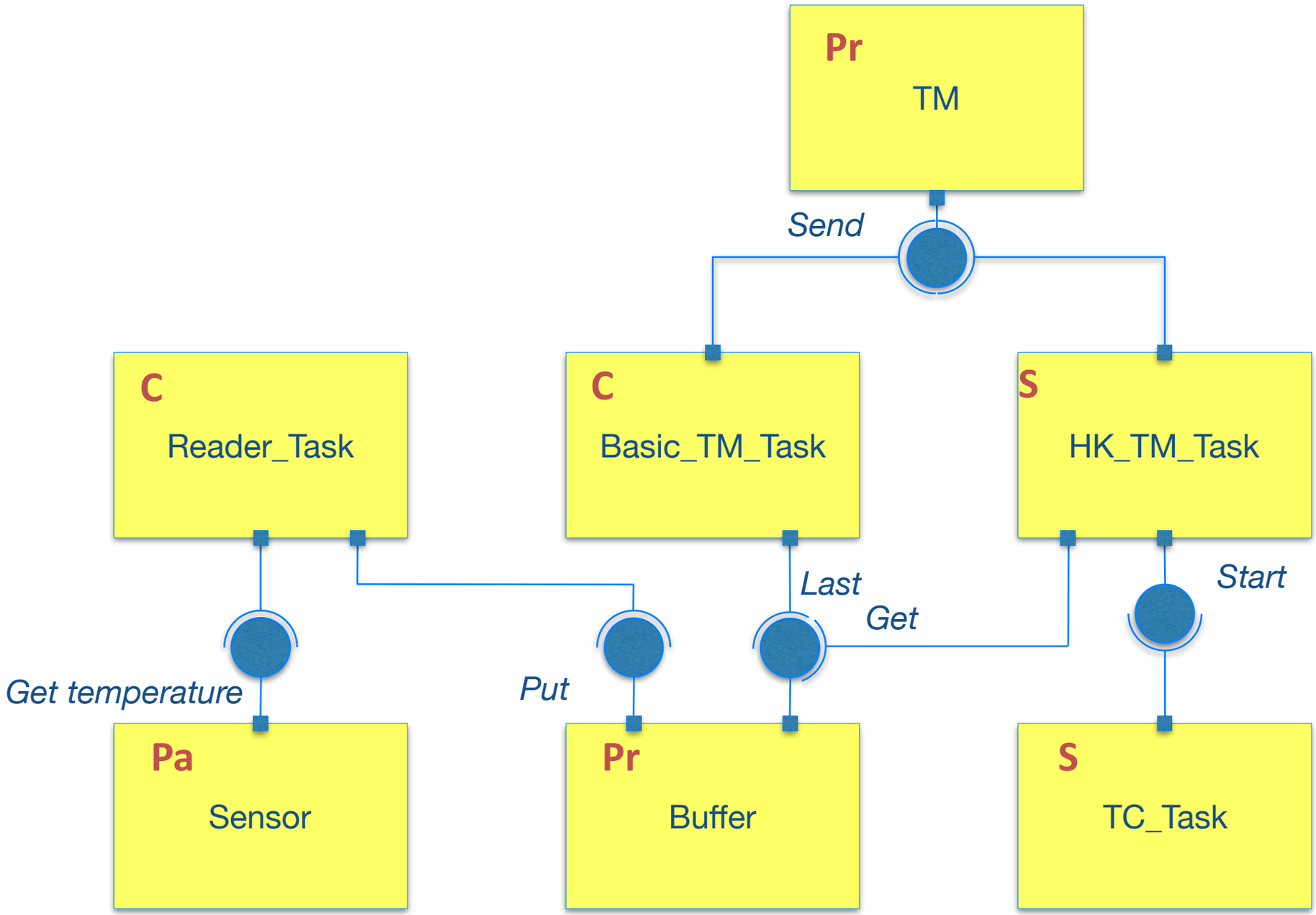
Context diagram



Architectural design (AADL)



Architectural design (UML)



Detailed design

Parameters

<<main procedure >>
OBDH

Measurements

Buffer

Reader_Task

Basic_TM_Task

HK_TM_Task

<<start>> + Send

Sensor

+ Get

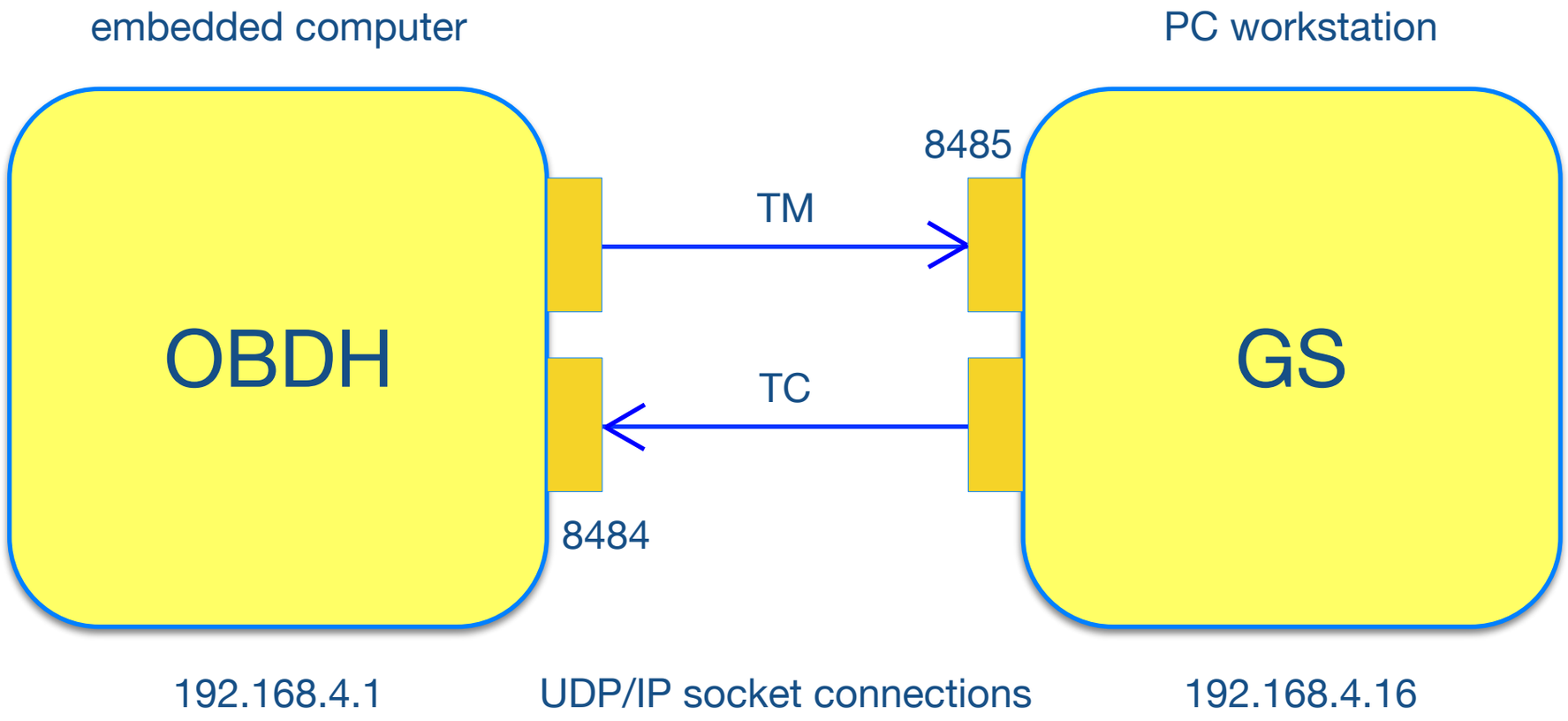
TM

TC_Task

RT Analysis

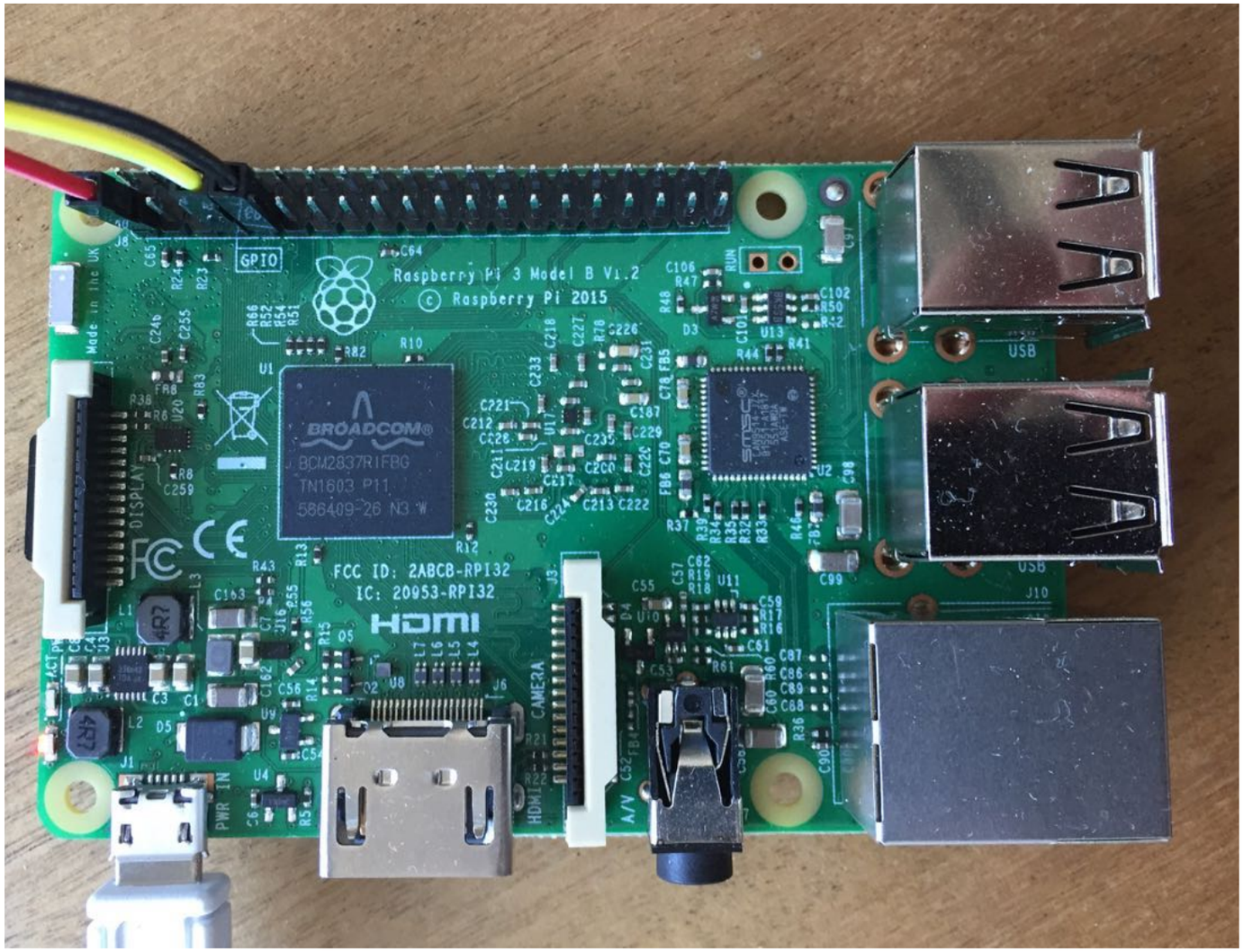
Task		P	T	C	B	R	D
TC	S	4	2,0	0,020	0,001	0,021	0,050
Reader	P	3	0,1	0,010	0,002	0,032	0,100
HK_TM	S	2	2,0	0,12	0,005	0,165	0,200
Basic_TM	P	1	10,0	0,050	0,000	0,210	0,500
PO							
HK event		4		0,001			
TC event		4		0,001			
Buffer		3		0,002			
TM		2		0,005			

Ground station

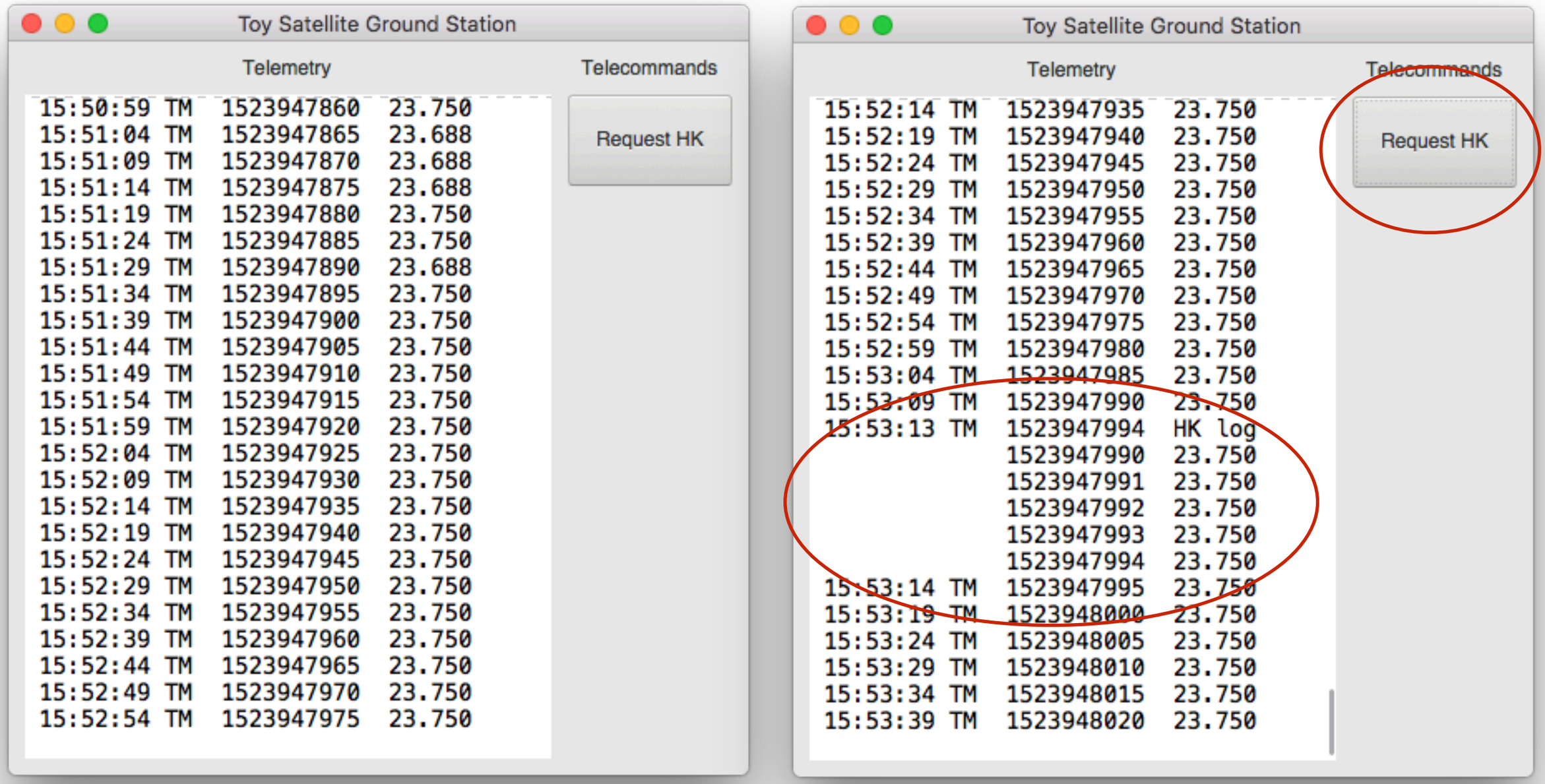


Embedded computer

RaspberryPi 3B



Ground station



runs on Ubuntu, MacOS, Windows

Implementation

- **obdh**: compile on development platform (linux)
  - download code from the STR-UPM ToyOBDH GitHub repository
  - compile and build with the raspberrypi-linux compiler
  - upload to the board through wifi connection

```
scp obdh pi@192.168.4.1:bin
pi@192.168.4.1's password:
obdh                                          100%  799KB 799.2KB/s   00:00
```

- **gs**: compile on linux/Mac/Windows workstation
  - requires a native GNAT compiler and the GtkAda library

References

- Github repositories
  - https://github.com/STR-UPM
    - ToyOBDH and ToyGS sources
- AdaCore Community site
  - https://www.adacore.com/community
    - native GNAT compilation system for the development platform
      - ✓ download also gtkada libraries
    - cross compilation system for Raspberry Pi with linux, hosted on linux