

2020.04.28

Project OBDH system

Juan A. de la Puente
[<jpuente@dit.upm.es>](mailto:jpuente@dit.upm.es)

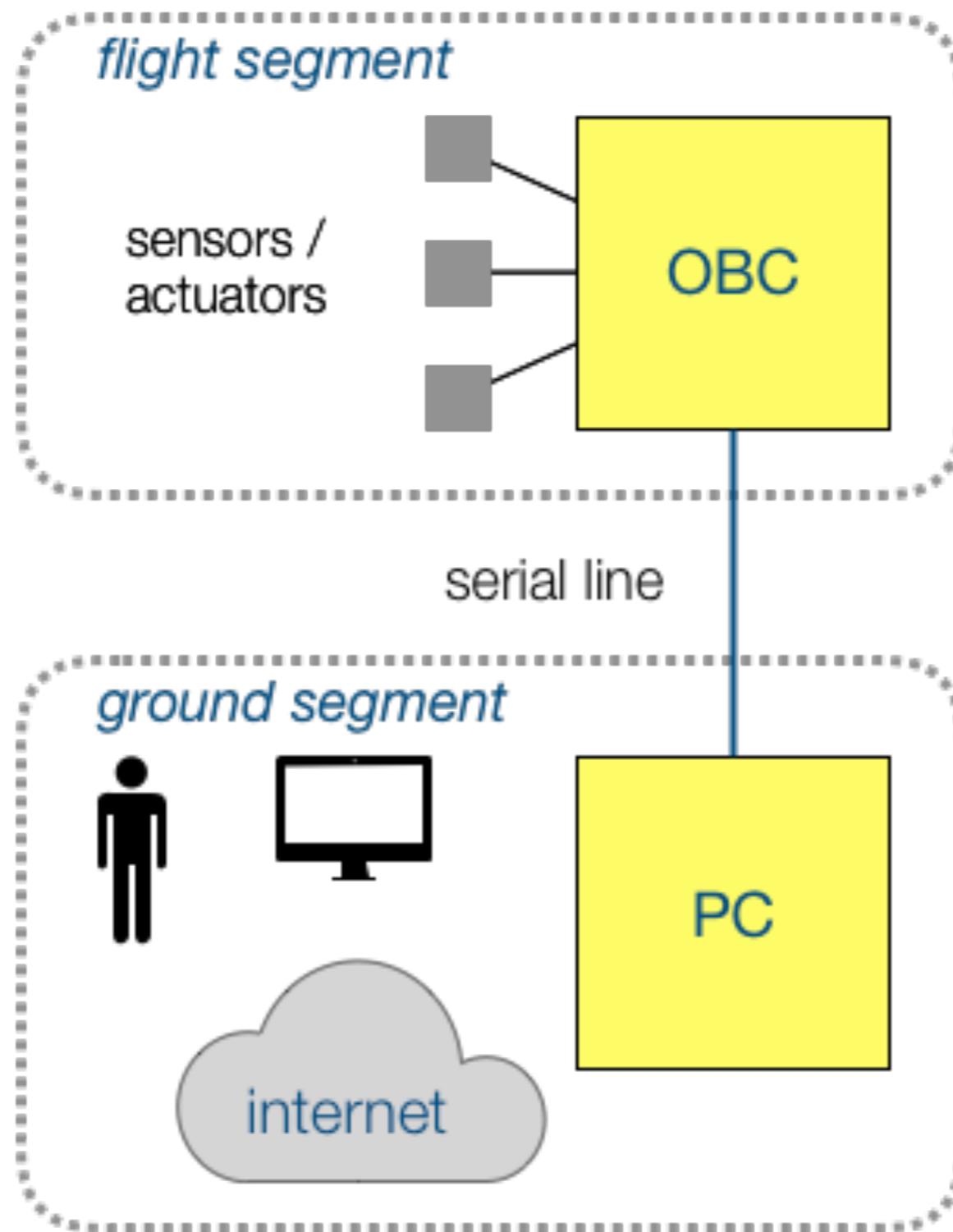


Algunos derechos reservados. Este documento se distribuye bajo licencia
[Creative Commons Reconocimiento-NoComercial-CompartirIgual 3.0 Unported.](http://creativecommons.org/licenses/by-nc-sa/3.0/deed.es)
<http://creativecommons.org/licenses/by-nc-sa/3.0/deed.es>

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
- The target platform is an STM32F407 discovery board
- The host platform is a PC workstation
 - ▶ Windows, MacOS, GNU Linux

Fight and ground segments



Functional requirements

- A set of sensors are periodically sampled with period T_s
- A basic TM message is sent periodically with period T_B . The message contains the last measured values from all sensors
- The system can receive a TC messages from the ground station, and replies with an appropriate TM message
- TM messages are stamped with the mission time when the message is sent
- Sensor measurements are stamped with mission time at which the measurements have been taken
- Time stamp values are given in seconds from the system start time, with a resolution of at least 1 ms

Temporal requirements

- Sensors 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

Operating modes

- **Idle**
 - ▶ The ground station has no visibility of the satellite
 - ▶ Only basic TM is transmitted
 - ▶ Only *open link* TC is accepted
- **Coverage**
 - ▶ The ground station has visibility of the satellite
 - ▶ Started upon reception of an *open link* TC from ground
 - ▶ TC can be sent to the satellite replied by TM
 - ▶ Switch back to idle mode upon reception of a *close link* TC
 - or automatically after a maximum visibility window time

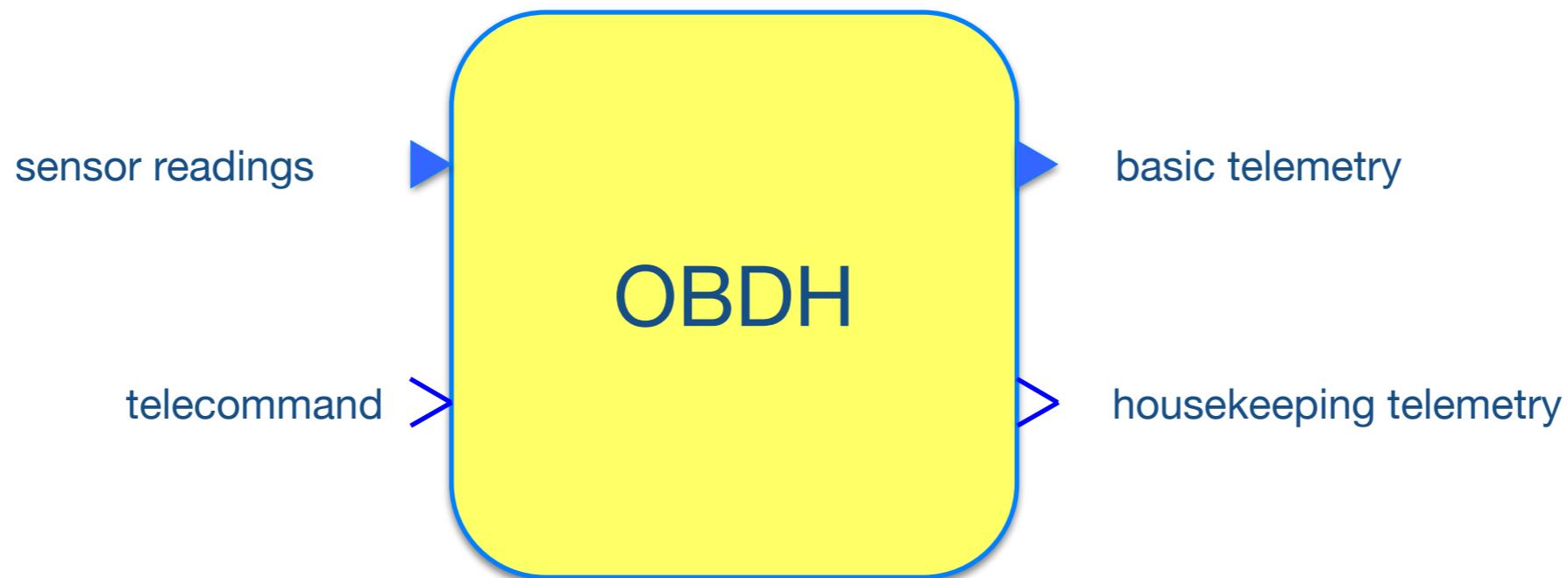
Telecommands

- OPEN : Open link
 - ▶ start coverage mode
- REQUEST
 - ▶ request housekeeping log
 - ▶ only in coverage mode
- CLOSE : Close link
 - ▶ end coverage mode

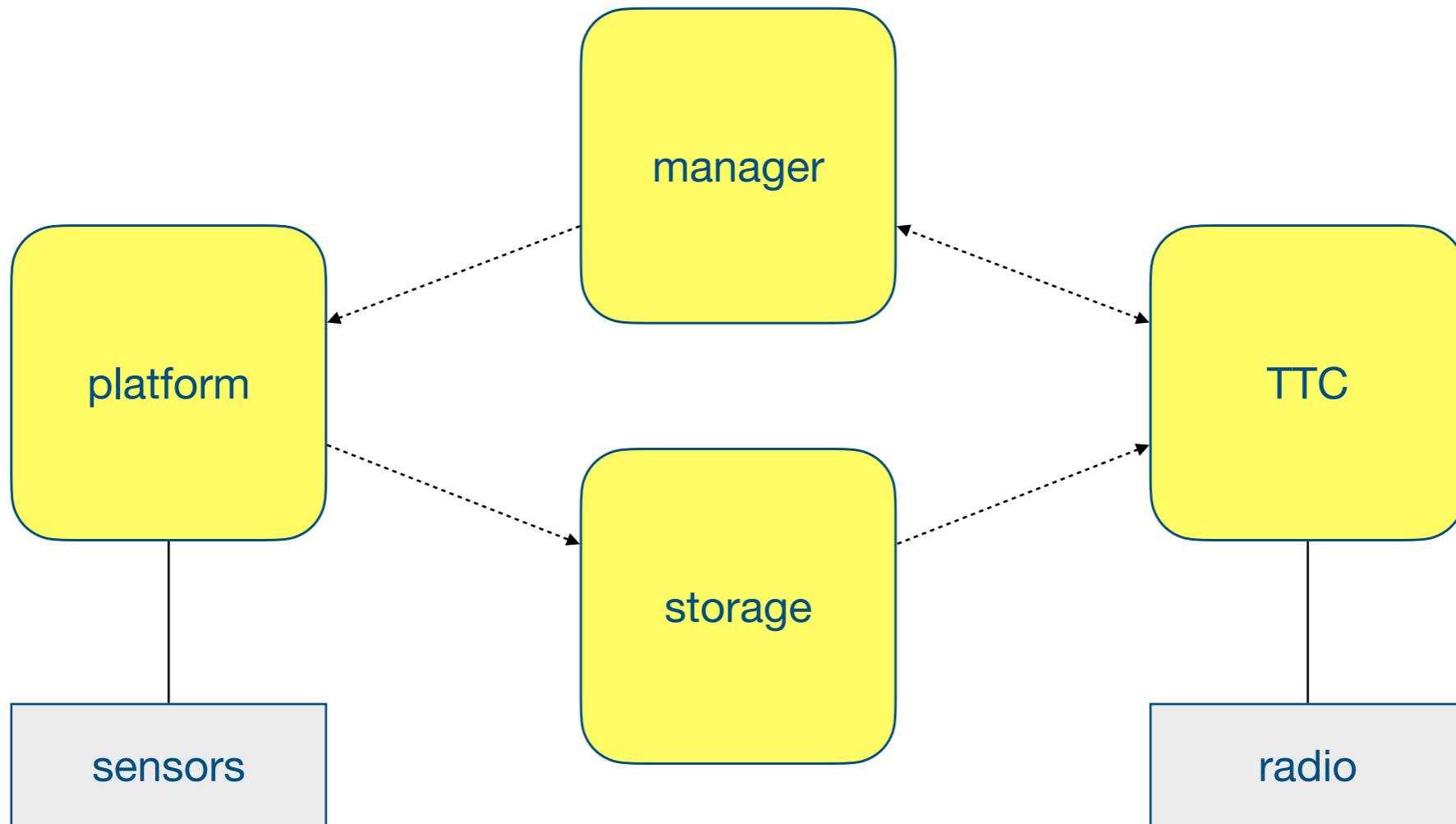
Telemetry messages

- **HELLO** : basic telemetry
 - ▶ last measured values from all the sensors.
 - ▶ periodically transmitted in idle mode
- **HK** : housekeeping
 - ▶ record with the last N measurements
 - ▶ transmitted in response to a telecommand in coverage mode
- **MODE**
 - ▶ current operating mode of the system
 - ▶ transmitted after a mode change in idle or coverage mode
- **ERROR**
 - ▶ abnormal conditions or erroneous TC received

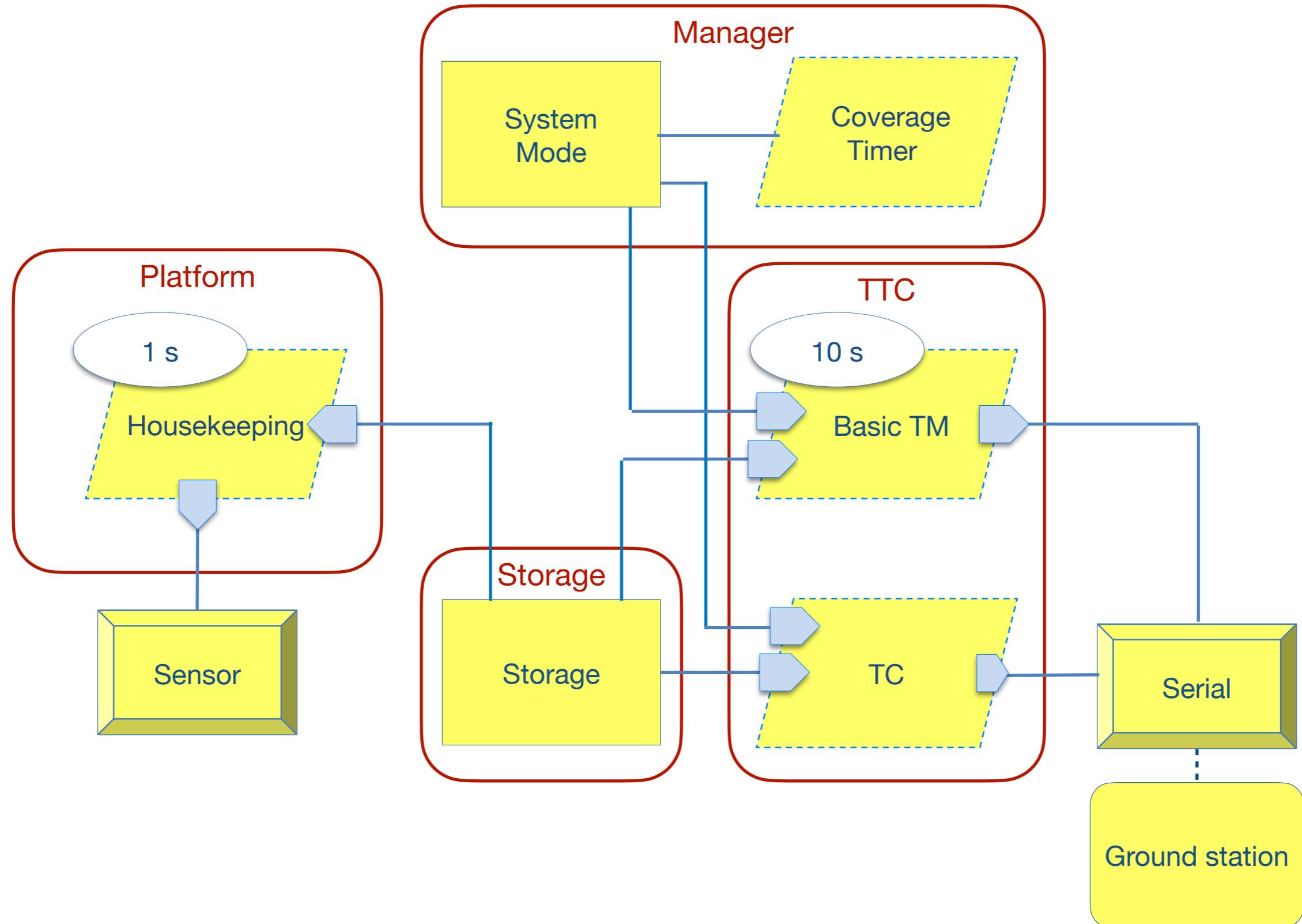
Context diagram



System architecture

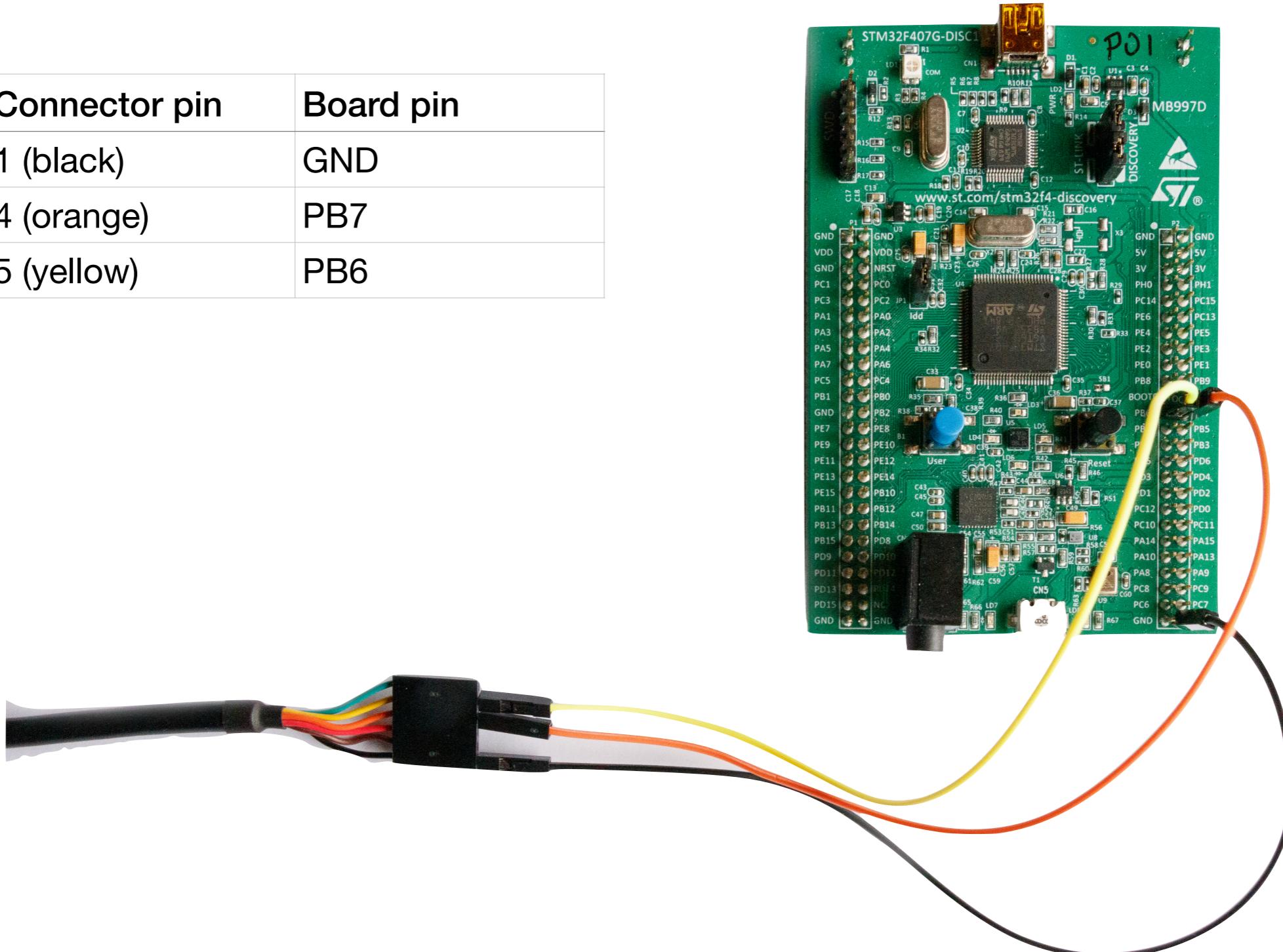


Architectural design (AADL)

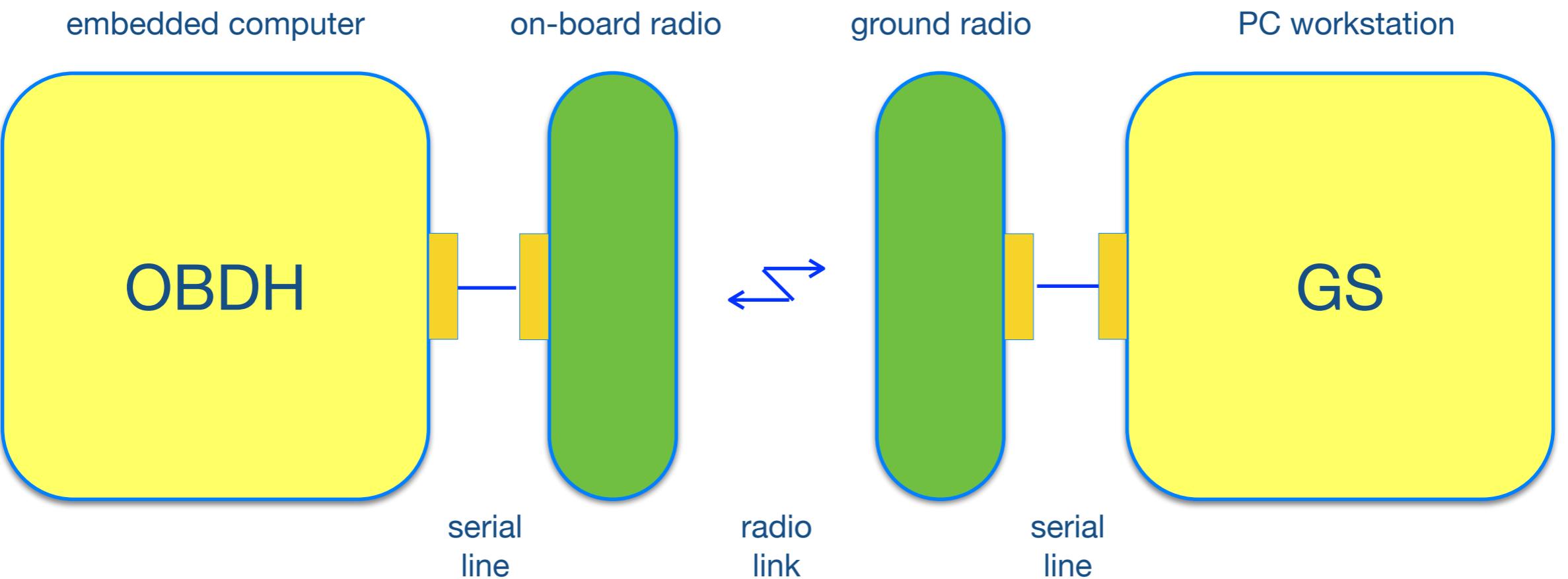


Hardware connections

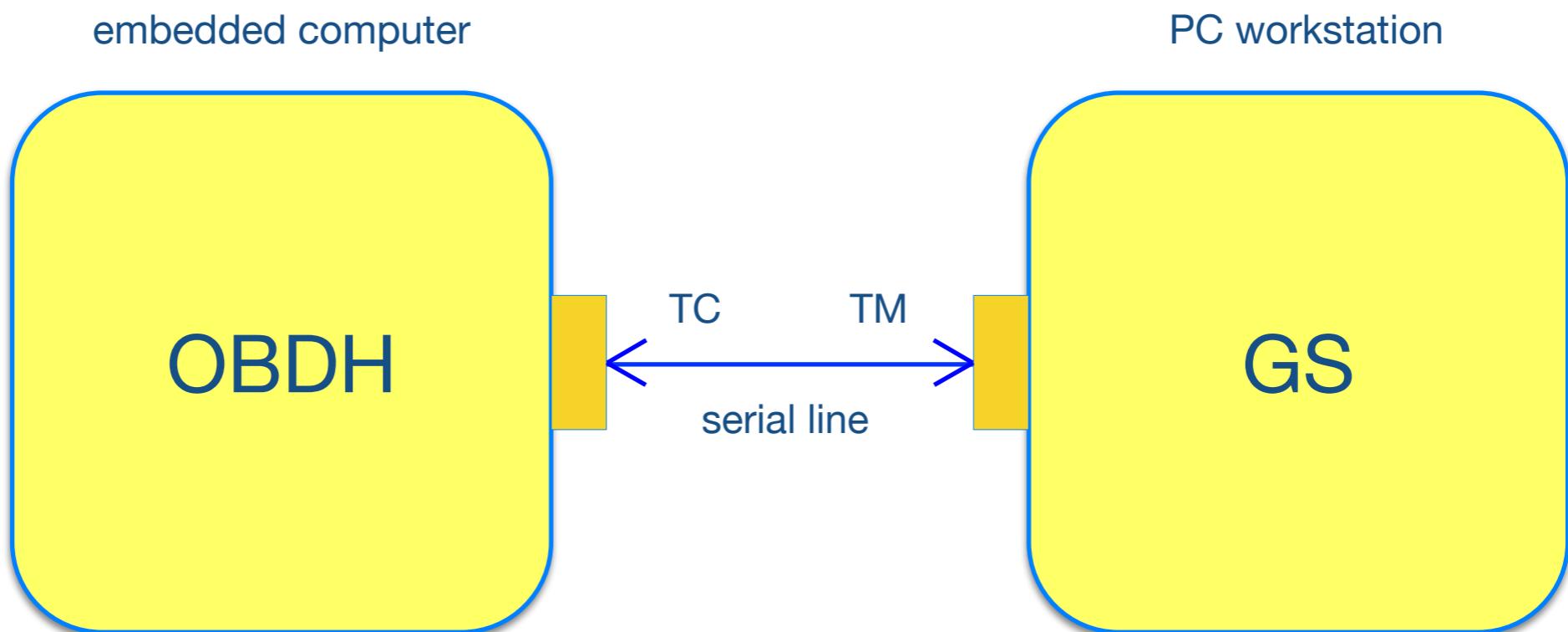
Connector pin	Board pin
1 (black)	GND
4 (orange)	PB7
5 (yellow)	PB6



Ground station



Ground station test arrangement



Ground station output

```
● ● ● jpuente — screen /dev/cu.usbserial-FTA5I24G 115200 — screen — screen /dev...
000000026 | HELLO 000000026:1063:2073
000000036 | HELLO 000000036:1063:2078
000000039 | MODE COVERAGE
000000045 | HK LOG
          000000040:1064:2080
          000000041:1066:2080
          000000042:1070:2077
          000000043:1063:2080
000000049 | MODE IDLE
000000056 | HELLO 000000056:1068:2080
000000066 | HELLO 000000066:1066:2079
|
```

Implementation

- **obdh:** compile on development platform
 - ▶ download code from the [STR-UPM OBDH LABS GitHub](#) repository
 - ▶ compile and build with the arm-elf compiler
 - ▶ upload to the board through usb connection

- **gs:** run on linux/Mac/Windows workstation
 - ▶ requires python3 and pyserial.py

RT Analysis

<i>i</i>	Task		P	T	C	B	R	D
1	HK	P	4	1,0	$13 \cdot 10^{-6}$	$4 \cdot 10^{-6}$	$17 \cdot 10^{-6}$	0,100
2	Timer	S	3	60,0	$5 \cdot 10^{-6}$	$2 \cdot 10^{-6}$	$20 \cdot 10^{-6}$	0,200
3	Basic_TM	P	2	10,0	$26 \cdot 10^{-6}$	$4 \cdot 10^{-6}$	$48 \cdot 10^{-6}$	0,500
4	TC	S	1	2,0	$20 \cdot 10^{-6}$	—	$64 \cdot 10^{-6}$	1,0
	PO							
	Storage		4		$4 \cdot 10^{-6}$			
	Mode		3		$2 \cdot 10^{-6}$			

Cálculo del tiempo de respuesta

- La ecuación del tiempo de respuesta es

$$R_i = C_i + B_i + \sum_{j \in \text{hp}(i)} \left\lceil \frac{R_i}{T_j} \right\rceil C_j$$

- ▶ La ecuación no es continua ni lineal
- ▶ No se puede resolver analíticamente

Iteración lineal

- La ecuación del tiempo de respuesta se puede resolver mediante la relación de recurrencia

$$w_i^{n+1} = C_i + B_i \sum_{j \in hp(i)} \left\lceil \frac{w_i^n}{T_j} \right\rceil \cdot C_k$$

- ▶ la sucesión $w_i^0, w_i^1, w_i^2, \dots$ es monótona no decreciente
- ▶ un valor inicial aceptable es $w_i^0 = C_i + B_i$
- ▶ se termina cuando
 - a) $w_i^{n+1} = w_i^n$ (y entonces $R_i = w_i^n$), o bien
 - b) $w_i^{n+1} > T_i$ (no se cumple el plazo)

Ejemplo

En el ejemplo anterior

$$R_1 = C_1 + B_1 = 17 \cdot 10^{-6}$$

$$w_2^0 = C_2 + B_2 = 7 \cdot 10^{-6}$$

$$w_2^1 = C_2 + B_2 + \left\lceil \frac{w_2^0}{T_1} \right\rceil \cdot C_1 = 20 \cdot 10^{-6}$$

$$w_2^2 = C_2 + B_2 + \left\lceil \frac{w_2^1}{T_1} \right\rceil \cdot C_1 = 20 \cdot 10^{-6}$$

$$R_2 = 20 \cdot 10^{-6}$$

etc.

