

Qualification of MicroPython for use in space applications

David Sanchez de la Llana, Manuel Fernandez (ESA)
Presenter: Manrico Fedi Casas (ESA)

06/06/2018

Contents

- What is MicroPython?
- Advantages
- Qualification targets
- Current status
- Applications
- Features
- Challenges
- Target missions
- License and Availability



What is MicroPython?



- “MicroPython is a lean and fast implementation of the **Python 3** programming language that is optimized to run on a microcontroller. ”
 - Kickstarted project by Damien George (George Robotics LTD).
- ESA asked for a micropython port to LEON on top of RTEMS.



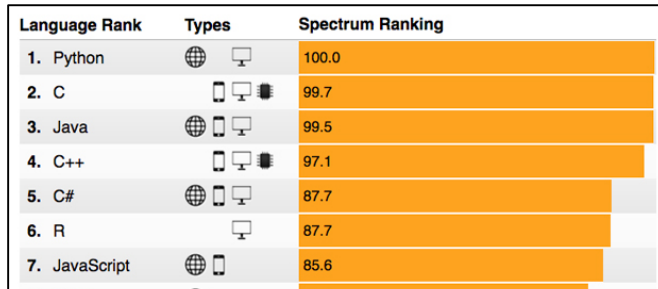
- Next step was to go for qualification as:
 - Software **Criticality Category B**.
 - Following **ESA standards** (ECSS).



MicroPython Advantages

MicroPython is “Python”

- High level programming language:
 - Widely used by industry.



Programming Language	2018	2013	2008	2003
Java	1	2	1	1
C	2	1	2	2
C++	3	4	3	3
Python	4	7	6	12

Source: IEEE Spectrum and TIOBE.com

- Standard (and powerful) SDEs exists as PyCharm, PyDev.
 - Auto completion, debugging, documentation, unit testing... are for free.
- The code (unchanged) is uploaded and tested in target.

MicroPython Qualification targets



- MicroPython for LEON shall run on top of
 - LEON-2, LEON-3, LEON-4 (monocore)
 - RTEMS 4.8, 4.10, 4.11



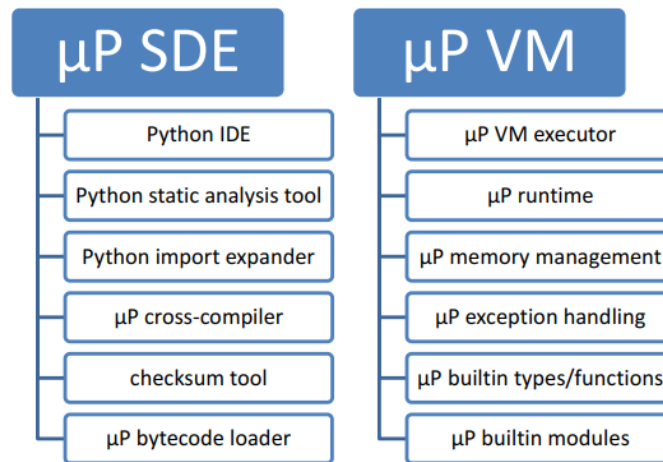
- Only RTEMS “Edisoft” 4.8 is **qualified**
 - Micropython must be qualified on top of a qualified RTEMS



Current status



- Prototype finished 2016
- Contract awarded to Spacebel (Belgium, SPB), near finalization
 - With George Robotics LTD (original MicroPython creator) as “service provider”
- Two products will be delivered :
 - Qualified (Category B)
MicroPython Virtual Machine
 - Free use inside ESA
 - Qualified (Category B) MicroPython
OBCP Engine (based in the VM)
 - Remains SPB proprietary
 - (see license schema later)

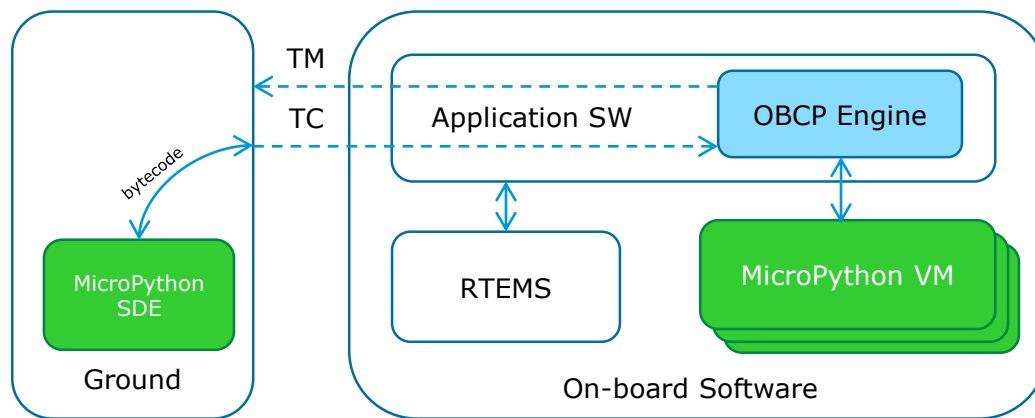


Applications: use as OBCP engine



OBCP (On Board Control Procedure)

- An OBCP script is a small program that can be uploaded by Ground control without requiring OBSW compilation (aka **Spacecraft Scripting**).
 - Commonly used for diagnosis, repetitive tasks, AIT...



Reproduced with Spacebel permission

Applications: use for FDIR, Mode Manager, ..



Failure Detection Isolation and Recovery, Mode Manager, Thermal Control SW.

- In general, as any “application SW” that requires low frequency (e.g. 1 Hz), no hard real time requirements, and may have complex logic.
 - Rich Python language features, exception management allows for
 - RAD (Rapid Application Development).
- But **NOT** as number crunching. (Performance x100 slower than native C code).



- Surprisingly wide set of Python 3 language features supported:
 - Almost all python syntax, objects and exception management.
 - **Interface with RTEMS** (queues, semaphores...).
 - Task scheduling delegated to RTEMS (1 VM = 1 Task).
 - **Easy interface** with C-code.
- Plus added:
 - **Determinism** if heap is locked (after initialization).
 - Separation and control of C-Stack and Python-stack.
 - **Small footprint (≈ 230 KB).**

MicroPython qualification: Criticality challenges



- No Mission Criticality Analysis: Different mission profiles
 - Set it as Software **Criticality Category B**.
 - ECSS-Q-ST-30C/40C Rev.1.
 - Severity Critical => Level 2 => **Mission Critical (Loss of Mission)**.
- Compliance to **ESA Standards**:
 - For Software and Software Product Assurance:
 - Missing requirements and design, but **good testing** (from python).
 - Independent Software Verification and Validation required.
 - => ESA providing additional verification services.

MicroPython qualification: Code challenges

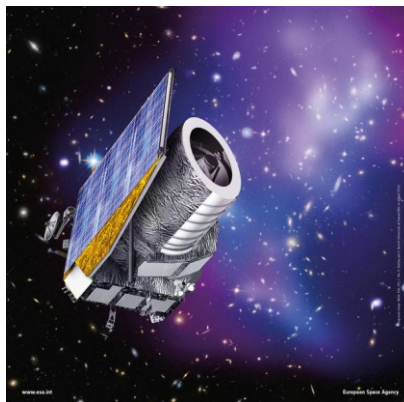


- The MicroPython VM is pre-existing:
 - Has a big amount of selectable “options”, “modules”, etc...
 - Big effort to select the ones that are really needed and shall be qualified.
 - Nice to have, less used features, excluded at pre-compilation level
 - Only a well-defined set is qualified.
 - Improvement of the code removing “not reachable branches”.
 - Effort to approach the behavior of MicroPython to that of C-Python.
 - Extra stack checks to guarantee that stack exhaustion is properly managed.

MicroPython qualification: Testing challenges



- MicroPython has a pre-existing “test bench”, similar to the C-Python test bench
 - Unit testing=>Major refactoring of codebase and huge effort... hardly feasible.
 - Use of:
 - Functional (+Expanded) testing covering features of the VM.
 - Combined tools to measure coverage for branches and statements (~100%!).
 - Adapt C-python tests to MicroPython to **guarantee compatibility**.
- Validation tested on:
 - Software Simulator.
 - Flight computer (Euclid).



MicroPython qualification: Continuous integration

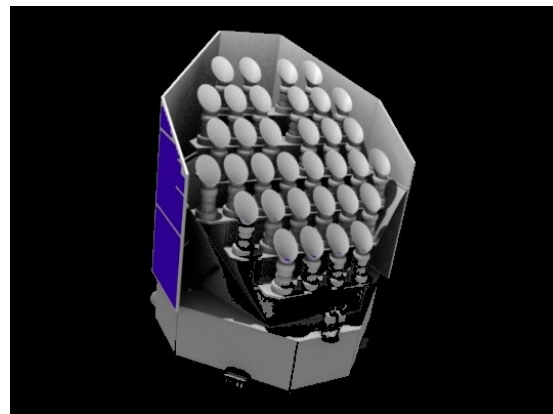


- As part of ESA contribution replacing the ISVV...
 - A Git server is set-up by industry, accessible by ESA
 - Mathworks Polyspace Bug Finder® is run each night on the last commit
 - Additional full static analysis (in depth) was made to filter out false positives on results from Polyspace Bug Finder®, Clang Static Analyzer, Infer.
 - Additional formal verification with Polyspace Code Prover® is on-going.
- Benefits:
 - The number of critical warnings reported by the tools has been reduced to zero.
 - The remaining warnings have all been reviewed and verified to be false positives.

MicroPython target missions (ESA)



- As part of the GSTP contract, SPB develops an engine for the Euclid mission
 - Euclid is a space telescope for measuring of dark matter and dark energy.
 - See <http://sci.esa.int/euclid>
 - Additional interest from the PLATO project (search for extrasolar planetary systems).
 - See <http://sci.esa.int/plato/>

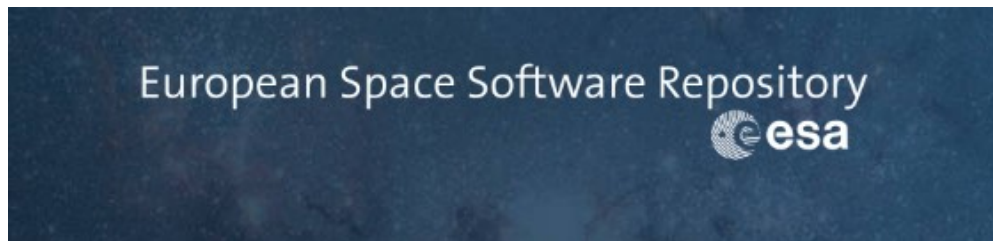


- 2 different License schemes apply:
 - MicroPython VM is MIT.
 - IPR (Intellectual Property Rights) is George Robotics LTD.
 - MicroPython adaption for LEON is ESA Community License Type 3
 - (Similar to MIT inside "ESA members states").
 - IPR is George Robotics LTD.
 - Export to third countries requires an ESA Authorization Transfer Board.
- Note: The OBCPE engine developed by SpaceBel remains SpaceBel proprietary.

Availability



- MicroPython is available at <https://essr.esa.int/>



→ MICROPYTHON FOR LEON

MicroPython is a lean and efficient implementation of the Python 3 programming language that includes a small subset of the Python standard library and is optimised...

Licenses: [ESA Software Community License – Type 3 - v1.1](#)

READ MORE



Updated on: 09/01/2017 Created on: 27/07/2016

Owner: George Robotics LTD

Links:

1. Please login to see 3 links.

Tags: [Python LEON](#)

- Under ESA License

(* Requires free registration. Only available for access inside "ESA space")

- Thanks for you attention!
- Questions?

