**Different boards available:**

* Beagle bone
* ESP32
* ARM
* Arduino
* STM32
* Raspberry Pi

**Different open source RTOS platform:**

We know that we are going to do with 3 languages C/C++, Java and Python we want to know what RTOS to use and what hardware to use. The software and hardware platform must be same for all three.

|  |  |  |  |
| --- | --- | --- | --- |
| **Hardware** | **RTOS/platform** | **Language** | **Library /IDE/API’s needed** |
| Beagle Bone Black | Xenomai patch is already there just use it that’s all | C/C++, Python can also be done. |  |
| Beagle Bone Black with C | Sequential program | C program is used and sensor interfacing is done | [cross compiling with Eclipse IDE](•%09https:/www.youtube.com/watch?v=4pz5nO6TVAA) |
| Raspberry Pi with Embedded C | But RTOS can’t be ported | Just for C program interfacing with sensors | [Wiring Pi library can be used](•%09https:/www.youtube.com/watch?v=IhPRNucVhYM) |
| Raspberry Pi with Java | Sequential programming | Java coding | [Java ME](•%09https:/www.youtube.com/watch?v=EMKdq_lw04A) |
| ARDUINO (I have experience) with Embedded C++ | FREERTOS [Not a full- fledged OS] | C++ programming | Can be done using ARDUINO IDE either on Windows or Linux and the code can be dumped on ARDUINO board even sensors can be placed in this. |
| ARDUINO with Java | [Java SE is used](https://www.youtube.com/watch?v=MsWIVsXR8bk) | [Java coding](•%09https:/www.youtube.com/watch?v=5N30jHMhw9c) | [It is done connecting to the serial port.](https://www.youtube.com/watch?v=vSezI5tKIi8) |
| Arduino with Python | No | Python | [Using Firmata sketch](https://realpython.com/arduino-python/) sensors can be interfaced |
| [Arduino with Python](https://micropython.org/download/) | Freertos | [Python](https://github.com/thearn/Python-Arduino-Command-API) | [Installation of Micropython](https://create.arduino.cc/projecthub/AdiK/arduino-due-micropython-enabled-717dd9) |
| ESP32 (I have experience) with Embedded C++ | FREERTOS [Not a full- fledged OS] | C++ coding | Can be done using ARDUINO IDE either on Windows or Linux and the code can be dumped on ESP32 board even sensors can be placed in this. |
| Esp32 with python | Micropython | Python 3 | [Using Pycraft IDE](https://randomnerdtutorials.com/getting-started-micropython-esp32-esp8266/) |

**Easier view:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Hardware platform** | **C/C++** | **Java** | **Python** | **Recommended** | **Boards availability and price** |
| Raspberry Pi | Yes(NO RTOS) | Yes (No RTOS) | Yes(no RTOS) | No | Yes |
| Beagle Bone Black | Yes | Yes (No RTOS) | Yes(No RTOS) | No | No (Rs.6000) |
| ESP32 | Yes(RTOS possible) | [Yes (No RTOS)with ESP wrover board](https://developer.microej.com/get-started-app-development-on-espressif-esp32-wroverkit-v41/) | [Yes(Possible)](https://docs.micropython.org/en/latest/esp32/tutorial/intro.html) | No | Yes |
| Arduino | Yes(RTOS possible) | Yes (But with Jamaica I will check) | Yes | Yes(but using micropython sensor interfacing is not put anywhere) | Arduino Uno (Yes)  Arduino Due (No) Rs.2000 |
| STM32 | Yes(RTOS possible) |  | [Yes(Possible)](https://micropython.org/download/) | Yes | No (Rs.1600) |

**How are real time capabilities achieved in each language?**

C/C++- RTOS and POSIX

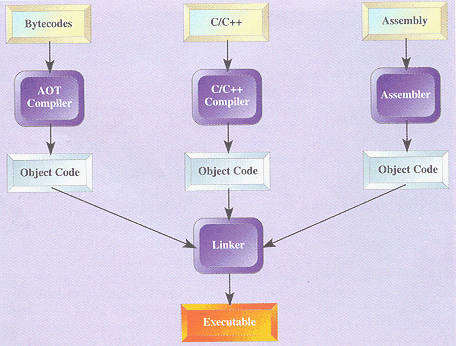
Java- Real time virtual machine(Jamaica VM)

Python- Micropython (Requirements- 256KB of ROM (flash memory) and 16KB of RAM)

**Why Java can’t be used for Real time systems:**

* [Main thing is memory constraint 8 bit boards don’t even have any space for taking up Java programs.](https://barrgroup.com/embedded-systems/how-to/embedded-java#:~:text=Developers%20of%20real%2Dtime%20systems,even%20have%20sufficient%20address%20space.)
* Just in time (JIT) compiler is an alternative for Java Virtual machine (JVM).
* For one thing, Java has no means of accessing memory or hardware registers directly (the downside of having no pointers).
* Lack of Libraries and API’s but Sun microsystems is developing it now.

Sun Microsystems has made Java processors which can be seen in the first groundwork document. This processors are really fast and can take java bytecode instead of object file.



**Real Time programming with python:**

Getting Python to work in real-time situations where execution deadlines are short can be tricky. Python’s philosophy when it comes to speed is basically “if it runs well enough, then it’s fine”. Normally this is a perfectly okay approach for a lot of applications, but if you want this to work in a real-time system with short task deadlines, you might end up with some problems, especially for tasks that are computation-heavy.

Usage of [pyboard hardware for micropython](https://micropython.org/download/)

Micropython can be done with ARM processor Arduino Due has ARM processor only have to try it with that.

**References:**

* <https://www.youtube.com/watch?v=af-1hDfoRcg> = Beagle bone black in visual studio
* <https://www.youtube.com/watch?v=thfxL7KAAyQ> = Beagle bone black startup
* <https://makezine.com/2014/02/25/how-to-choose-the-right-platform-raspberry-pi-or-beaglebone-black/> = Comparison of Raspberry Pi and beagle bone black
* <https://www.youtube.com/watch?v=fVSOZDOzTjk> = Running python program in beagle bone black
* <https://www.youtube.com/watch?v=4pz5nO6TVAA> = Doing cross compiling in Beagle bone black with C code.
* <https://www.youtube.com/watch?v=jY7KVruyGjs> = Video showing Raspberry Pi C programming.
* <https://www.youtube.com/watch?v=IhPRNucVhYM> = Raspberry Pi with Wiring Pi library for C.
* <http://www.diva-portal.org/smash/get/diva2:1251188/FULLTEXT01.pdf> = Xenomai installation on Raspberry pi and latency and stress test is done.
* <https://www.youtube.com/watch?v=grORuAxDQ2Q> = Java ME installing on Raspberry Pi(Java ME doesn’t have real time capabilities).
* <https://www.youtube.com/watch?v=EMKdq_lw04A> = Java ME led blinking using Raspberry Pi.
* <https://www.youtube.com/watch?v=5N30jHMhw9c> = Doing java in Arduino but no real-time capabilities
* <https://barrgroup.com/embedded-systems/how-to/embedded-java#:~:text=Developers%20of%20real%2Dtime%20systems,even%20have%20sufficient%20address%20space.> = Why java can’t be used for real time applications.
* <https://www.youtube.com/watch?v=MsWIVsXR8bk> = Netbeans IDE is used
* <https://www.youtube.com/watch?v=vSezI5tKIi8> = Potentiometer java code on Arduino board.
* <https://create.arduino.cc/projecthub/Menphis/multitasking-and-real-time-arduino-system-789d6f> = Arduino programming using Erika RTOS.
* <https://realpython.com/arduino-python/> = Arduino with pyFirmata
* <https://create.arduino.cc/projecthub/smart-tech/programming-arduino-using-python-f3d2c0> = Python using pyserial library on Windows.
* <https://www.edn.com/pythons-role-in-developing-real-time-embedded-systems/> = Things about micropython which adds real time capabilities to python.
* <https://learn.adafruit.com/micropython-basics-what-is-micropython/overview> = Has whole information about what board supports micropython and comparison of Arduino and micropython.
* <https://pumbaa.readthedocs.io/en/latest/getting-started.html> = Using pumba to achieve micropython in Python language
* <https://www.youtube.com/watch?v=5W3WvXAmDJc&list=PL6F17pWypPy_KSmpnR5CV8x78QhAiKBIl> = micropython with esp32 series.
* <https://micropython.org/unicorn/> = You can run micropython online also
* <https://jsolano.net/2018/01/28/real-time-software-with-micropython/#:~:text=MicroPython%20has%20evolved%20recently%20to,in%20a%20matter%20of%20minutes.> = Shows how to test the interrupt latency gpio toggling using oscilloscope.