**Why I chose Cruise control?**

Chosen Java virtual machine JamaicaVM was used more in automobile industry so we chose an application based on that.

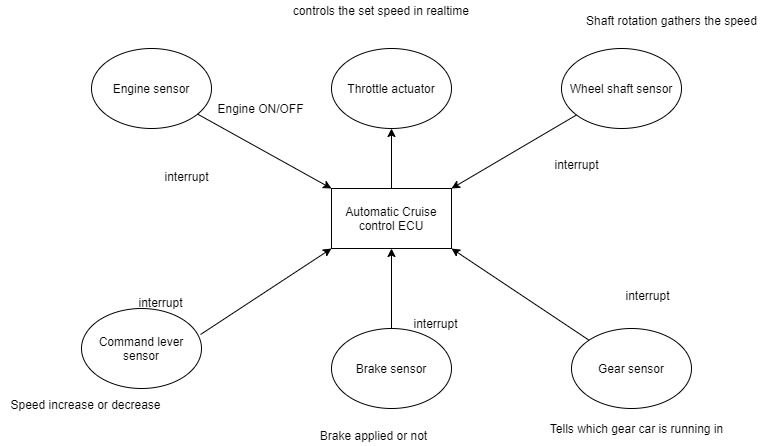
**Things which I found:**

* Automotive domain people test and develop in Matlab->Simulink they do software testing and then they convert the code for specific thing like autosar, emdedded code, we can generate C/C++ code. Then they implement in board. But in out college they keep a board and do freertos code simply.
* Should I use the sensors used in this also or random variable number generation will do.
* If you go to matlab and addons->> Hardware support tools we get everything like Arduino, beagle bone, raspberry pi and all.
* Using HIL(Hardware in loop) they can simulate and test the hardware
* To see the Simulink block this is the path to matlab example files C->Users->Prashaanth->Documents->Matlab->Examples->mpc\_featured. This Simulink aims at correcting the safe distance gap and telling the host car.
* They are doing it in Real time Simulink.

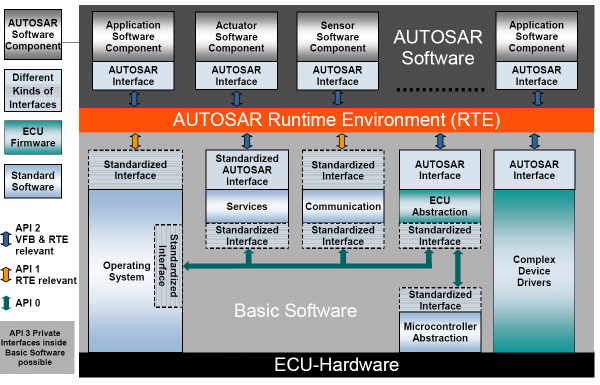
**Functionalities:**

* There will be a button for adjusting the necessary speed.
* The system will work only in top gear and above 70 KM/hr.
* If the brake is pressed the system will be deactivated.
* If the speed is set and activate button is pressed the cruise control system will be turned ON. I the system is ON the set speed will be maintained the driver has to lower or higher the speed manually based on the need and vehicle coming.
* Only when the engine is ON cruise control system works.
* If the driver tends to lower the gear the cruise control will be deactivated.

**Block diagram:**



**About Autosar:**

****

Here developing things in normal OS is not possible (like Windows CE, Vx works and all) Automotive they have specific OS. [OSEK OS](https://www.irisa.fr/alf/downloads/puaut/TPNXT/images/os223.pdf)

OSEK defines three processing levels:

* Interrupt level
* Logical level for scheduler
* Task level

**Free open source OSEK OS [Erika enterprise]:**

Erika Enterprise is an **open-source OSEK/VDX Hard Real Time Operating System (RTOS).**

<https://www.youtube.com/watch?v=flcyGWf60hM&t=193s> = Shows implementation in Ubuntu.

**Getting freertos on raspberry pi:**

<https://github.com/jameswalmsley/RaspberryPi-FreeRTOS>

<https://github.com/Forty-Tw0/RaspberryPi-FreeRTOS>

<https://www.eecs.umich.edu/courses/eecs473/Labs/Lab3F17.pdf>

<https://www.raspberrypi.org/forums/viewtopic.php?t=201447> = Gives 3 github links that tells how to do it.

<https://www.freertos.org/FreeRTOS_Support_Forum_Archive/July_2017/freertos_FreeRtos_porting_to_Raspberry_Pi_3_c67682caj.html> = Main forum telling freertos not available for Raspberry Pi

**Difference between full fledged RTOS and bare bone RTOS:**

An RTOS is an OS which is intended to serve real-time application requests. It must be able to process data as it comes in, typically without significant delays1 . RTOSes come in a wide variety of forms. Some are full-fledged OSes with an emphasis on real-time responses. Those may have memory management, disk protection, support for multiple simultaneous users, etc. Others are more “bare-bones” and can be viewed as just of a collection of libraries to do specific tasks. In this lab we’ll be using an RTOS that leans toward the “bare-bones” side of things—FreeRTOS.

**References:**

1. [https://in.mathworks.com/matlabcentral/answers/339523-compile-simulink-generated-code-in-arduino-ide](https://www.google.com/url?q=https://in.mathworks.com/matlabcentral/answers/339523-compile-simulink-generated-code-in-arduino-ide&sa=D&source=hangouts&ust=1590125485758000&usg=AFQjCNEu1kU7YaLXUCYuDctbBjAVEJmKnw)
2. <https://www.youtube.com/watch?v=9ttIZoUXCKY>  = This video shows how you can include different boards to dump the matlab code.
3. <https://en.wikipedia.org/wiki/Hardware-in-the-loop_simulation>
4. <https://www.youtube.com/watch?v=lOJLlc48VH0> =video shows the cruise control Simulink matlab
5. <https://www.youtube.com/watch?v=Rb3Aefnu0NY> = this shows HIL testing with matlab
6. <https://www.youtube.com/watch?v=iouY7iYJoeA> = dspace hil simulator explained
7. <https://www.youtube.com/watch?v=07IecbWS9Wk> = dspace tutorial
8. <https://in.mathworks.com/matlabcentral/answers/489925-how-to-perform-real-time-simulation-with-matlab-script> = this tells that you should use a real time OS after you extract code from code generator.
9. <https://www.irisa.fr/alf/downloads/puaut/TPNXT/images/os223.pdf> = Whole OSEK information is available here.
10. <https://www.autosar.org/fileadmin/user_upload/standards/classic/3-0/AUTOSAR_SRS_OS.pdf> = Autosar documentation for RTOS
11. <https://www.autosar.org/fileadmin/user_upload/standards/classic/4-2/AUTOSAR_SWS_OS.pdf> = autosar full documentation
12. <http://erika.tuxfamily.org/drupal/> = This is where you can download Erika enterprise

t1. get.priority() tells you the priority.

T1. Getname() tells you the thread number it’s available in Eclipse platform.

<https://www.youtube.com/watch?v=BeV8eh84tEM> = Check this link if you want to know priority or getname.