

question

[Like](#)

30 views

question on step 10 in section 12.7 (week C12) on robot car construction

Below is a cut and paste of section 10 of C12.7 / robot car construction. (My questions follow the cut and paste of a blurb from step 10).

10) Battery - 8.4V NiMH or 11.1V Lilon. I bought the 8.4V NiMH batteries you see in the video as surplus a long time ago. I teach a real-time OS class where students write an OS then deploy it on a robot.....

Question:

Above it says that students in your real time OS class write an OS and then deploy it on a robot. Do they use the Keil IDE to do that (deploy their own written OS on their robot?)

The reason I ask is because I noticed in the Keil tools we select which OS we download, and in this class's case, it is the TExaS OS. But if your students in another class download

their own OS, I was wondering if they just place it in the same directory as the TExaS OS, and download their own OS instead of TExaS OS. It would have to work with ARM though

for example, and any OS that was written for the Intel X86 for example would have to instead be written for the ARM, correct? So I was wondering when I saw your comment above if

this was how your students achieved the download of their own written OS to their own robot (just curious.)

Another Question:

Are there any plans to have the Keil IDE work on other desk top OS's such as Linux, or will it always run on Windows based computers? The reason I ask is because currently

I have to borrow my 11 year old son's lap top to do the labs, since my loaned desk top computer while unemployed has Linux on it! ;-). He does not take it with him to school, so

given I'm not working at the moment, I'm easily able to borrow it, but I was just wondering this as well.

c12

21 hours ago by Karen West

the students' answer, *where students collectively construct a single answer*

[Click to start off the wiki answer](#)

the instructors' answer, *where instructors collectively construct a single answer*

For the first question, the Texas OS just creates an option to call Texas DLLs in windows. This OS has nothing to do with embedded or microcontrollers. In Texas OS we write simulators for windows GUI. Ofcourse they interact with Keil but it is not something that runs on a launchpad. I took the OS course last year under him. We wrote the code in Keil just like you do. We had 7 labs, and by Lab 7 we had a Keil project which had entire OS C code (similar to one of your labs here) and we just downloaded the code and pressed reset to execute it.

As for question 2, I am not sure if Keil for linux will be available any time soon. I suggest either install virtual machine/ vine / or try with code composer studio

16 hours ago by Chinmaya Dattathri

followup discussions *for lingering questions and comments*

Resolved



Unresolved

**Karen West** Just now

I think I'm just going to borrow my son's lap top. I once tried installing a virtual machine on my Linux machine that was provided by one of the MOOCs and it was

running as a VM a different type of Linux that had all their tools installed on it, and any associated course code, and there was some sort of version issue I could never

work out, and I was unable to take part 2 of that MOOC. I had taken part 1 of that MOOC on a Windows machine (that crashed before part 2 began) and although the virtual

machine software ran fine on my old Windows machine, and I completed that entire MOOC part 1 running Linux as a VM on my Windows machine, when I tried to install

that same VM on my Linux machine--I could not get it to work, and wasted an incredible amount of time trying to get all this done, and then was unable to take part 2 of

that MOOC class. Another time I tried installing some VM software on Linux that caused me to have to reinstall Linux afterward. So if the online

MOOC does not have

someone who can provide guidance to a novice using the VM software, I avoid it, so for this class, I opted to borrow my son's Windows lap top he just received at Christmas

time since the machine I have right now runs Linux. I have had some success using the VM's to write and debug code, but there have been other occasions where I had so

much trouble getting the tools to work that I do not do it unless there is some TA that can provide guidance in using it for a MOOC.

Also, for the class you took from Prof. Valvano / Prof. Yerraballi (?) last year, where you wrote an OS that you downloaded to a robot you made using the Keil tools,

it sounds like you must have written an OS to download in C that would be ARM based. My husband wrote an embedded OS for X86 and I have never written my own OS,

and have never seen an online class that teaches that, but the thought had crossed my mind that if it was not too difficult, perhaps I could download my husband's OS for the

X86 converted to ARM to try it on the LaunchPad--but I'm guessing that would be beyond my expertise anyway, and my husband is the overworked professor type anyway,

who would have no time to help if I got into trouble! Where he teaches his OS and systems and physical systems classes are not online anyway, so you're on your own if

you want to attempt anything his classes do.

So thanks for letting me know how that class worked for you too, since I was curious about the ability to download different embedded OS's to the LaunchPad using the

Keil tools. As for what you described with Texas OS creating an option to call Texas DLLs in Windows--all beyond me--I know very little about Windows internals--but

get the gist of it, that it has nothing to do with the embedded microcontroller. What I wonder though is what is running on the embedded microcontroller--it has to be

running some sort of control program or embedded OS of some kind, in order for our code to run on the board instead of the simulator.

So sorry for the confusion and thanks for the advice / help! ;-)