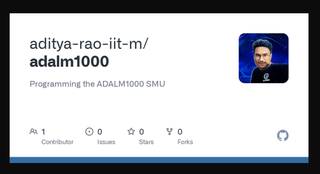
Python Programming for the ADALM1000 - Github link

AdityaRao 13h

Detailed instructions and source code to control the ADALM1000 using Python, as mentioned by Janakiraman Sir in the EE1103L Electronics Lab live session, couple of days ago on Monday : **https://www.youtube.com/live/8zaR6yib5eI** .



GitHub - aditya-rao-iit-m/adalm1000: Programming the ADALM1000 SMU

Programming the ADALM1000 SMU. Contribute to aditya-rao-iit-m/adalm1000 development by creating an account on GitHub.

Let me know if it worked for you. Please ensure you follow the Python 3.10.10 installation instructions carefully. This step has been a bit tricky for many students, but gets resolved in 10 minutes if done correctly.

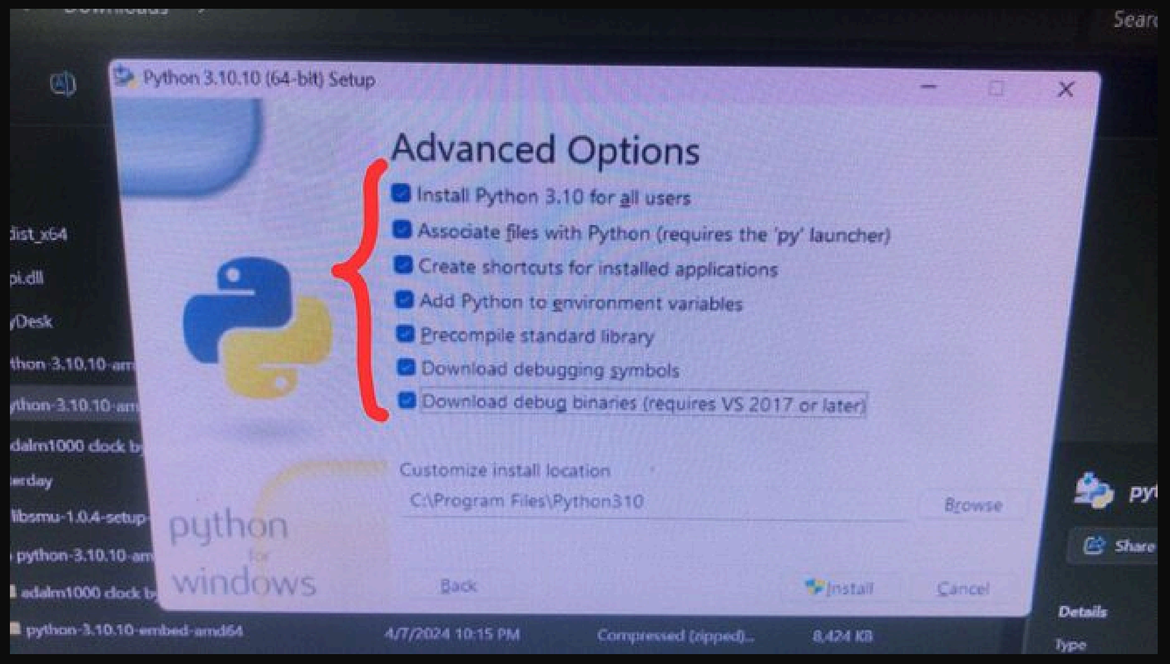
AdityaRao 1h

Here are the details and screenshots of each step in the Python 3.10.10 installation that is critical to get this setup properly. Each step mentioned here is super important, without which this code for ADALM1000 will not work properly. Python has to be setup this way only - no other choice :

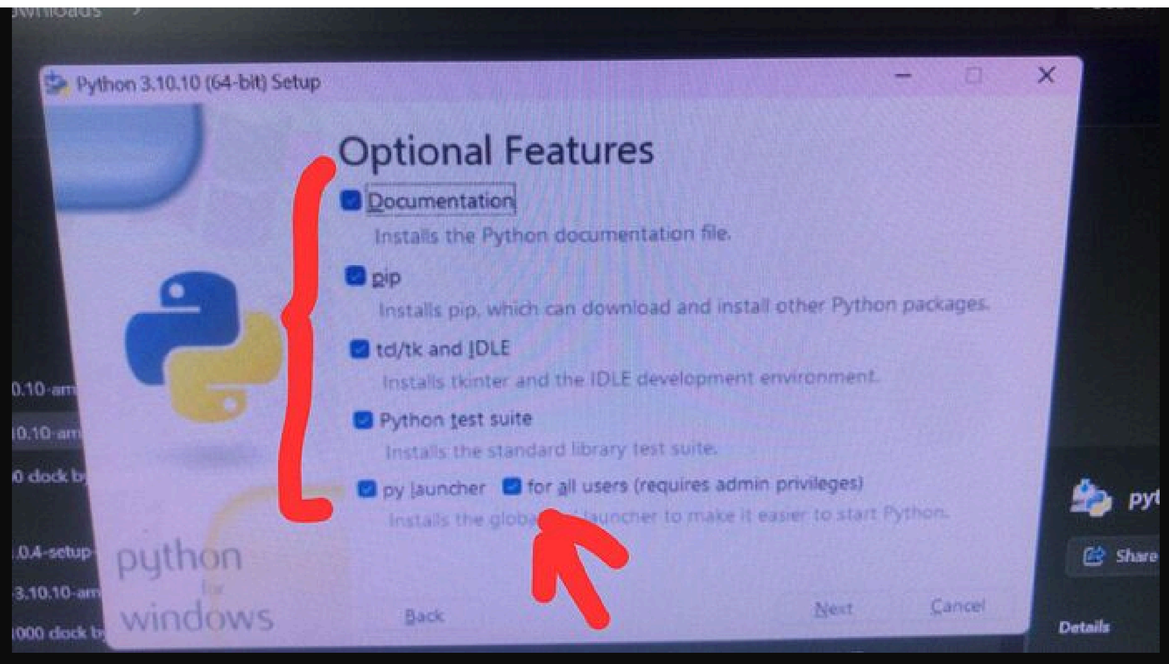
1. First uninstall any versions of Python that is installed in your computer. Go to Control Panel / Installed Apps and uninstall. (Important step to prevent previous erroneous settings from interfering with this install.)

2. Check Box “Add python.exe to PATH” is very impportant. Check all boxes here and select “Customize Installation”

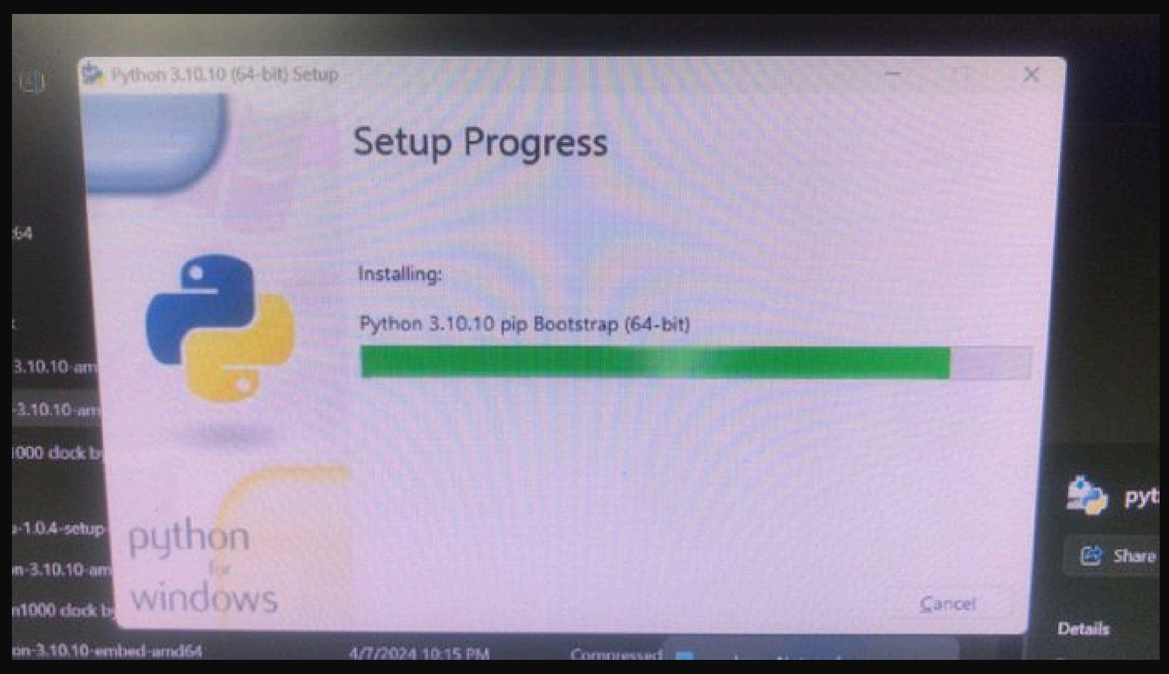
3. Check all boxes in the “Advanced Options” step



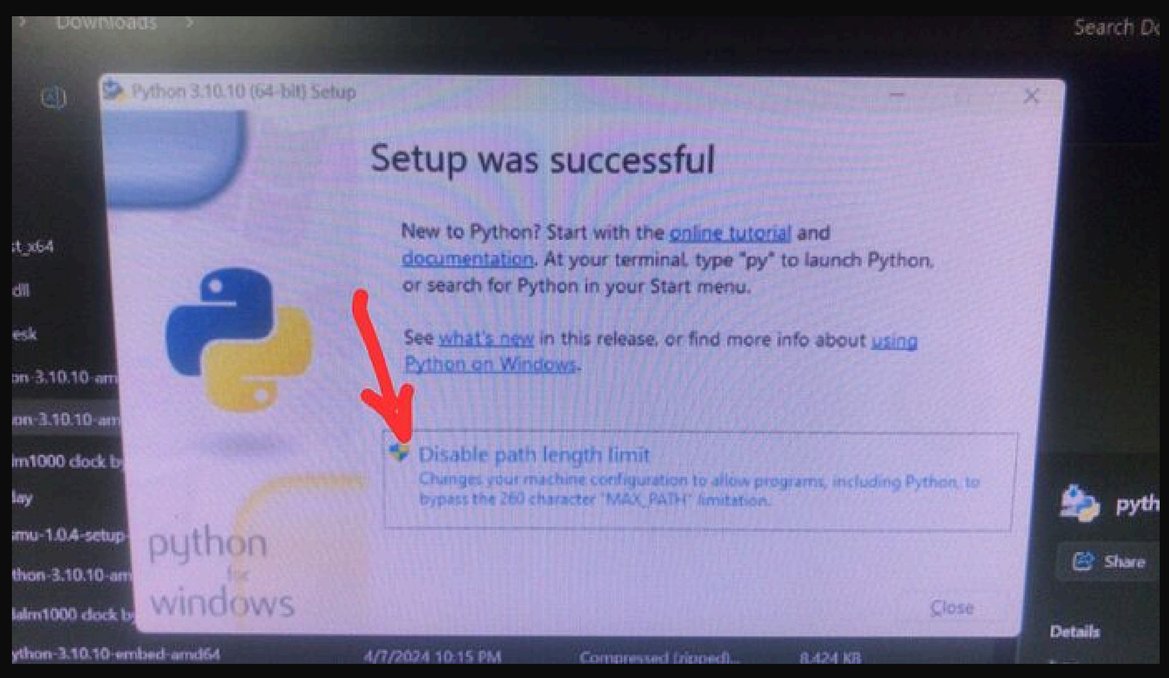
4. Check all boxes in the “Optional Features” step



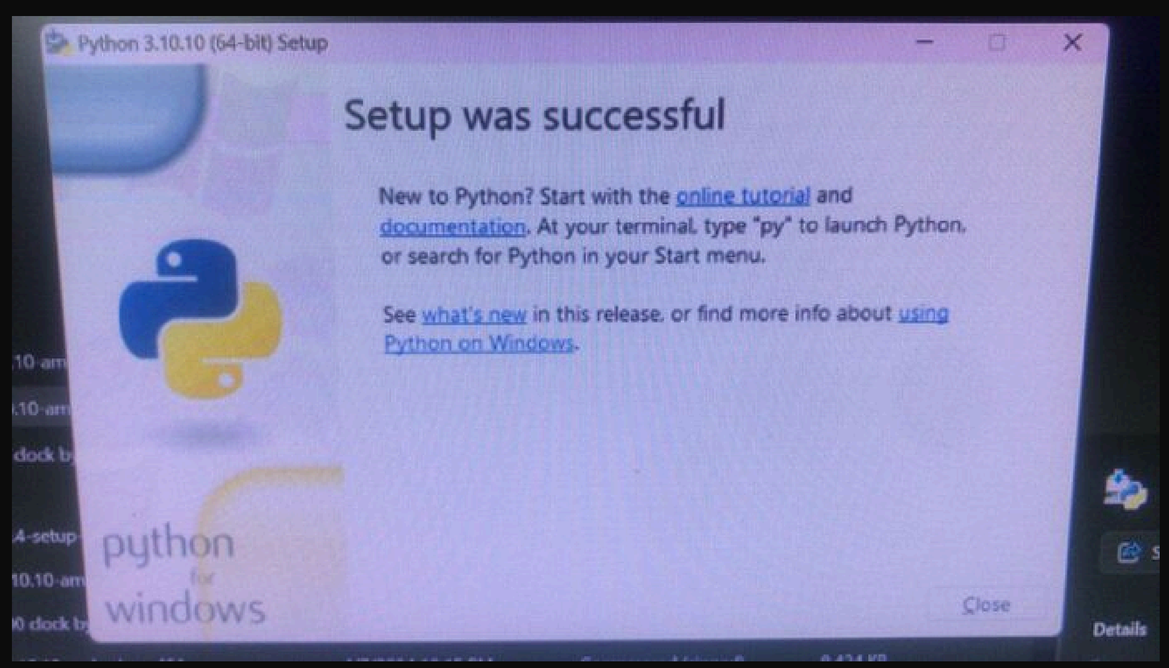
5. Setup will now proceed to install as per choices selected, in the next step



6. Then it will notify that “Setup was successful” and offer to “Disable path length limit”. Please click on that.



7. Then it will declare that “Setup was successful” again.



8. REBOOT the computer for all the PATH settings to get registered, then proceed with the remaining part of the installation of **libsmu** and **pysmu** and **ar_iitm whl file** as detailed in the README.md in the github repo.

Each one of these steps has to be done… if even one of these steps is missed or skipped, the python programming for ADALM1000 might not work as expected.

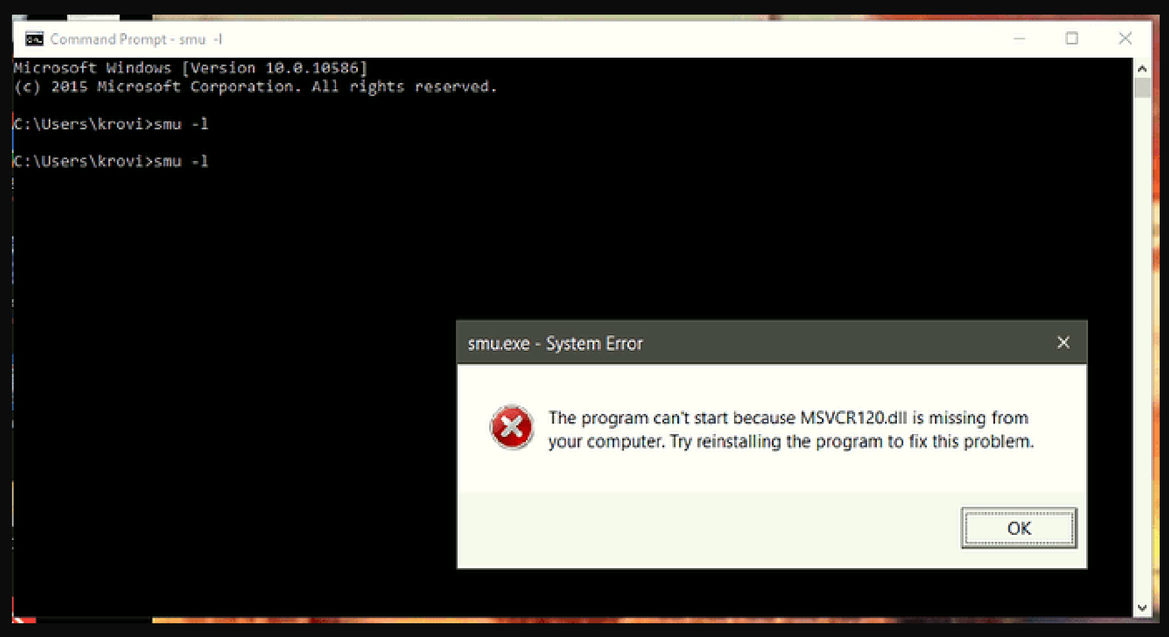
You may have to go through step 1 to step 8 again to get it working properly. All this takes just 10 to 15 minutes to be done properly.

Hope this helps someone who might be struggling at this stage.

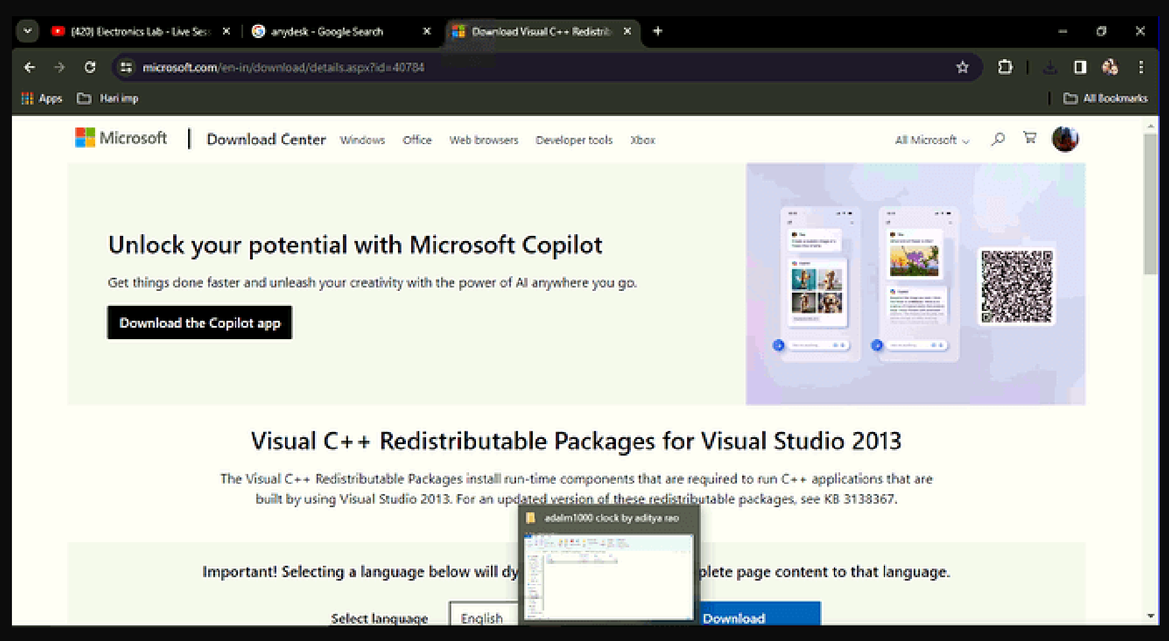
AdityaRao 36m

While helping a few of my course mates set this up properly, over an AnyDesk session, in some systems, I found that the VC++ redistributable is not installed.
Here are the steps to solve that.

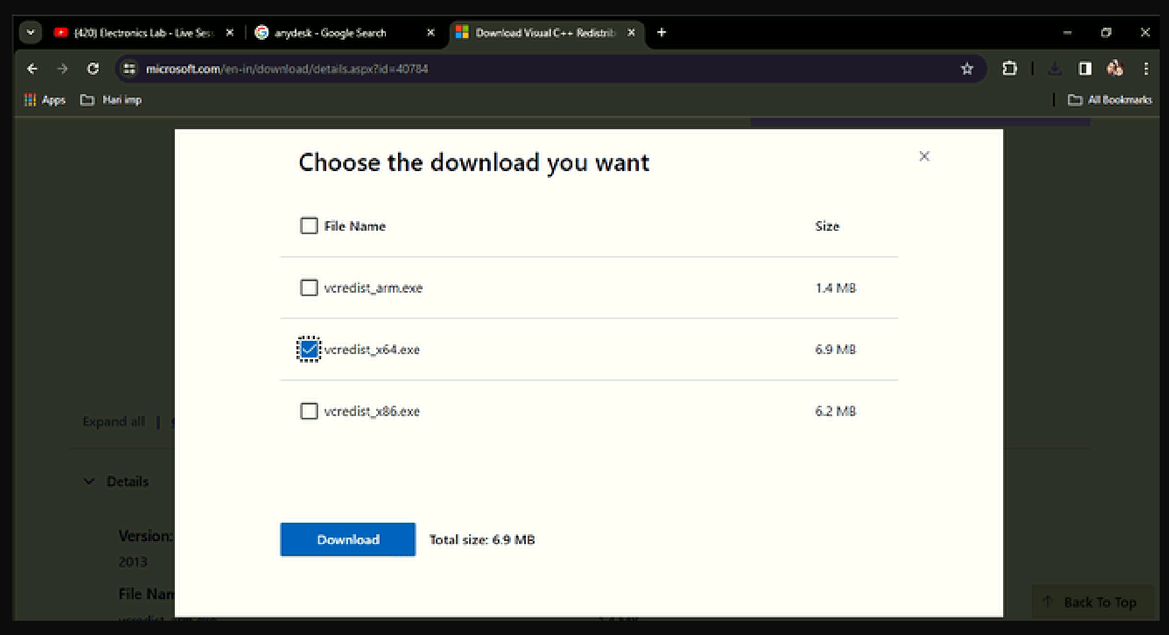
1. If the command “smu -l” gives an error like this : MSVCR120.dll is missing, follow these steps to solve the issue.



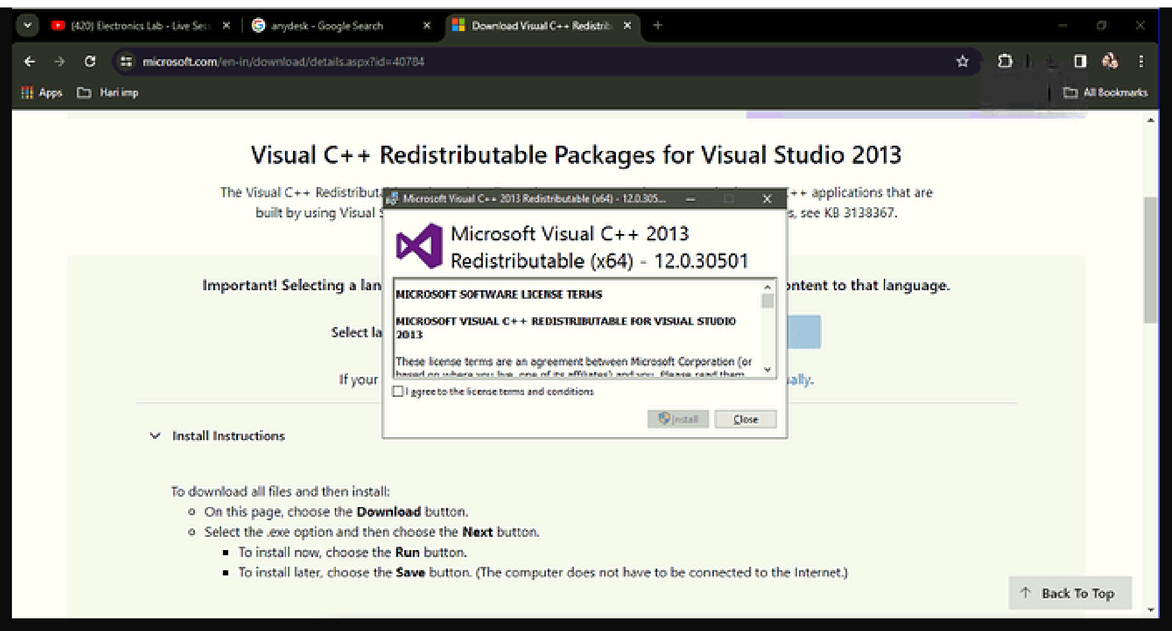
2. Visit : https://microsoft.com/en-in/download/details.aspx?id=40784 and click the blue “Download” button to get the necessary setup file.



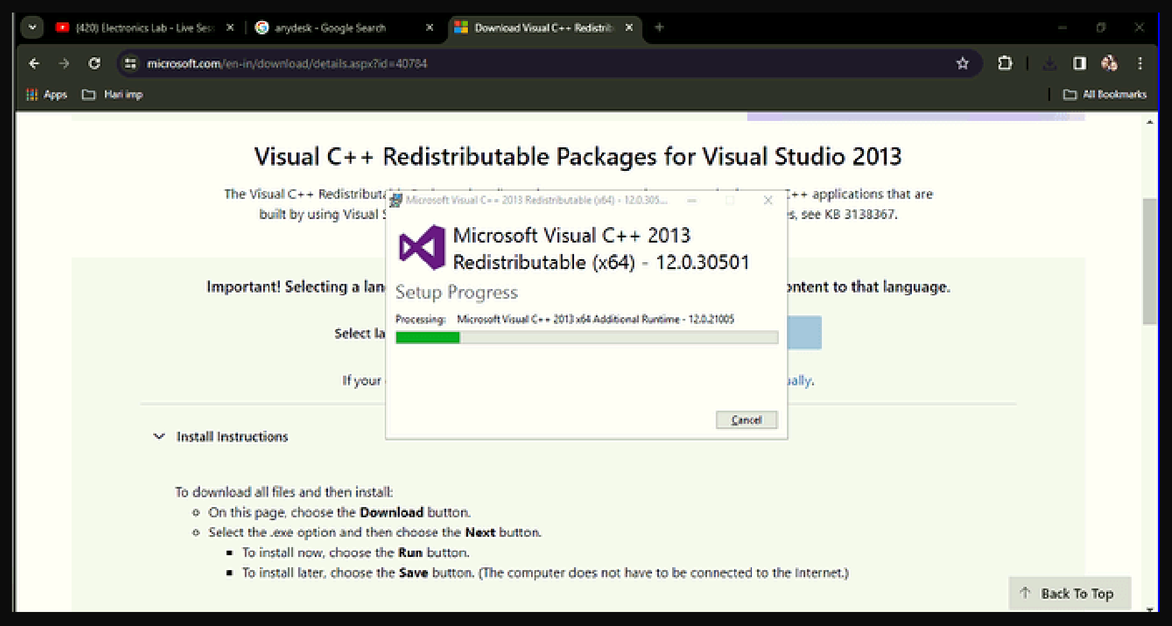
3. Choose the vcredist_x64.exe if your laptop has a 64bit intel or amd cpu. If its a tablet, you might need the arm variant (less likely though).



4. Run the downloaded installer. Click on the check box to “I agree to the license terms and conditions” and install.



5. Setup will proceed and you will not get the MSVCR120.dll message again



6. REBOOT to be sure that all the installations have registered properly in the Windows registry.

This issue can be resolved in this way.
Then proceed with the steps detailed in the Github README.md, and your python programming for the ADALM1000 can start.

AdityaRao 25m

Download link for all the files from the Github Repo (zipped : 14th April 2024 version 1.0)

20240414 adalm1000-main.zip

Google Drive file.

Includes the sample code : rgb_clock_test.py

Can be modified and customized to program the 3 R,G,B notification leds to blink with any delay (in seconds) and also program the 4 PIO pins high or low for any delay in seconds, at the digital output side of the ADALM1000 smu kit.

Try generating clocks of various frequencies, create a 4 bit counter by connecting LEDS to the PIO pins directly, drive a bcd counter, drive a seven segment display using the 74LS47 chip directly from the ADALM1000.

Please share pics of your ADALM1000 setup and python coding experiences and all digital experiments, performed using this code at github here in this thread... can't wait to see what you achieve using this library.

Related Topics

Topic	Replies	Views	Activity
My first experiments with the ADALM1000 — extra-resources, how-to, es-program	5	65	Nov 2023
ADALM100 kit for IITM Electronics Lab — clarification, operational, miscellaneous, discourse-forum	0	39	Dec 2023

Skip to main content

Topic	Replies	Views	Activity
Best price - Authorized reseller - ADALM1000 kit - Element14 - Rs.7930/- all inclusive (Nov 2023) operational extra-resources	53	620	Feb 1
Lab 4 : 74LS163 clock using the 555 chip error general	3	163	5d
Uses of thisADALM 1000 Analog kit after the lab of IITM clarification	1	78	Nov 2023