PHYS 319 Programming and Debugging with the MSP430 for Windows

This is a summary of how to get started writing assembly and C programs for the MSP430 Launchpad that we will be using in Phys 319.

The new Texas Code Composer Studio works well with Windows 10.
To install it go to:
https://software-dl.ti.com/ccs/esd/documents/ccs_downloads.html
and download Version 10. The off line installer works best. After downloading install it with custom installation using the parts mensioning MSP430.
Open CCS and cheese new project, set the family to MSP430G and processor to MSP430G.
You have a choice of three options to proceed:

1) Set up the assembler tools as described later in option 1. Use Windows 10 with the CCS for labs 3-6.

2) You can use the lab computers (not in 2021!). The lab computers have Linux installed and are known to work with

the Launchpad boards. The downsides of this option are that the lab computers are a) old, b) slow, and c) not available when you're not in the lab.
3) We have prepared a Linux operating system image that you can write to a USB flash drive. You can then boot your computer from this USB flash drive and use the same software as we have installed on the lab computers. This will not harm the software on your computer. When you remove the flash drive and reboot, your computer will be just as it was before. With a modern flash drive, this option is actually very attractive, and is recommended.

Instructions for setting up Option 3 follow, with instructions for Option 1 afterwards. Instructions for Option 2 are in a separate document.

Setting up a USB flash drive [Option 3]

1) Buy a decent USB flash drive. Really, spend ~$20 and get a USB 3.0 drive with a read speed greater than 100 MB/s. Write speed is less important. It doesn't need to be huge (though larger drives tend to be faster). 8 GB is plenty big, but you probably want a 16 or 32 GB drive for the increased speed. If the packaging doesn't specify the read speed, keep looking. London Drugs has a pretty good selection. A decent flash drive on a USB 3 port runs nearly as fast as an installed OS. A cheap USB 2 drive will feel like molasses.

2) Download our OS image from:
http://www.phas.ubc.ca/~kotlicki/Physics_319/ubuntu-mate-p319.iso

3) Download and install unetbootin from https://unetbootin.github.io/

4) plug in your USB stick. Ensure there is nothing on it that you want to keep, as we'll overwrite the entire stick.

5) Start up unetbootin. Select Diskimage, ISO, and navigate to the iso image your downloaded in step 2).
IMPORTANT: enter 4000 in the field: "Space used to preserve files across reboots)." Without this, nothing you do will ever get saved.
Then, select the USB stick as the target drive (make sure you select the correct drive!), and hit OK.

6) Wait till its done. It might take a while.

7) Test the USB stick: Reboot your computer. There is a slim chance that it will boot from the USB stick immediately. Probably it won't, and you will probably need to tell the BIOS setup program to boot from the USB stick. Reboot again, and press F1 or F2 or F8 or F10 or F12 (which one varies from computer to computer) to get into the BIOS setup program. You'll need to dig around the options to find something like "Boot Order." Set the USB stick so it is first. Then exit the BIOS setup, making sure to Save Changes. Now the computer should boot from the USB stick into Linux. The username and password are both phys319.

On some computers, the BIOS will reset the boot order after the USB stick is removed. If your computer doesn't boot from the USB stick the next time you insert it, you may need to repeat the above steps to put the USB stick as the highest boot priority each time the USB stick is used.

Once booted from the USB stick, no software installation should be necessary and you can follow the "Using the Lab Computers" document (everything except the Logging In section).

The os will probably prompt you at some point to install updates. Don't do it! It will eat up the persistent storage on the flash drive.

Installing Linux on your own computer [Not recommended]

It is possible to install Linux onto your computer from the USB flash drive, but this is outside the scope of these instructions and carries some risk of loss of data. If you do attempt this, following instructions from the web on resizing your windows partition, one quirk with the image we created is that the installer will fail unless it is started with root permissions. So to do an install from the flash drive you created, boot it with the 'live session' option, then from a terminal window run: 'sudo ubiquity' from a terminal window to start the installer.

Setting up for MSP430 on Windows [Option 1]

We will need several components: an assembler, a program to load our assembled programs onto the MSP430, a C compiler, and a debugger. There are a number of options available for several of these components. For the first couple of weeks we only need the assembler and loader, so we'll deal with those first. For the third and fourth weeks we need a C compiler and debugger, and for the following two weeks, we'll use some other software on the host computer to talk to the Launchpad via its serial port.

We strongly recommend using a browser other than a Microsoft browser for these steps. Use Firefox or Chrome.

Installing and using an Assembler (Needed for Labs 1+2)

There are a number of assemblers available for the MSP430 line. To keep things simple and the same for everyone, we're going to use a small, bare-bones assembler.

1) Download the .zip file with the assembler from:
http://www.mikekohn.net/micro/naken430asm_msp430_assembler.php
For now: Download the package and unzip it.

Grab:https://www.phas.ubc.ca/~kotlicki/Physics_319/blink.asm
and ensure that you can assemble it without errors.

To use the assembler:
Move your program to the directory where the assembler was unzipped, then get a command window,

```
naken430asm.exe -I .\include\ -o <name>.hex <name>.asm
```

(where <name> is the name of your program)

2) To load programs into the MSP430 on Windows, we'll use a utility provided by TI, MSP430Flasher. Download it from:
http://processors.wiki.ti.com/index.php/MSP430_Flasher_-_Command_Line_Programmer

TI will force you to register at their site and make you promise that you won't export their code to Cuba before allowing the download. Execute the .exe installer. By default the flasher will install into

```
\ti\MSP430Flasher_1.3.7
```

The installer will prompt you to get USB FET drivers. You'll need them, but don't follow the link for now. It takes you to a page that is somewhat confusing.

Copy two files: MSP430FLASHER.EXE and MSP430.dll from the install directory into your assembler directory so you can do all your assembly programming and flashing from that one directory.

You will need a driver to talk to the serial port on the Launchpad. Download and install EZ430-UART.zip from:
https://github.com/energia/Energia/raw/gh-pages/files/EZ430-UART.zip
Unzip it, and run DPInst.exe or DPInst64.exe (depending on whether your windows version is 32 or 64 bits – probably 64 for a recent computer).

You won't be able to test downloading programs without a Launchpad, but to use it, from the command line type:

```
MSP430Flasher.exe -n MSP430g2xx3 -w <name>.hex -v -g -z [VCC]
```

You can look at the MSP430_Flasher web page to see what all the options do.

Sometimes the EZ430-UART driver installation doesn't 'take.' After you plug in the Launchpad, it should appear as a serial port in the windows device manager. If the port appears to have a problem, open the Properties, and Update Driver. Then point to the .inf file that came in the EZ430-UART.zip download.

Installing a CCS For Labs 3-6

TI provides a feature-rich integrated development environment (IDE) called Code-Composer Studio (CCS). .

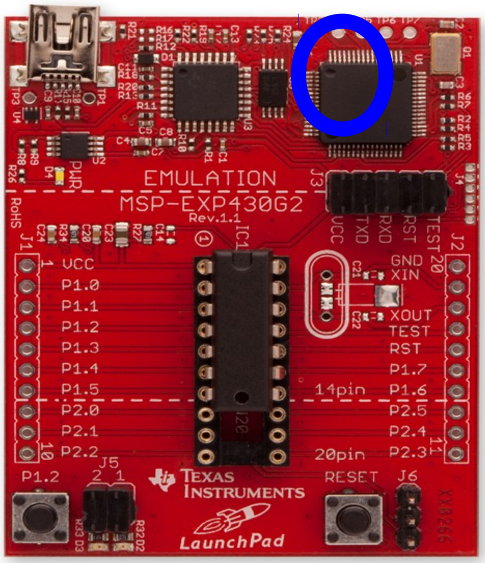
The new Texas Code Composer Studio works well with Windows 10.
To install it go to:
https://software-dl.ti.com/ccs/esd/documents/ccs_downloads.html
and download Version 10. The off line installer works best. After downloading install it with custom installation using the parts mensioning MSP430.
Open CCS and cheese new project, set the family to MSP430G and processor to MSP430G.
Name your project and choose empty project with main.

Our next example is to set up a program that talks to the MSP430 while it is running. The Launchpad board presents a USB-Serial interface to the host computer, so that we can talk to the MSP430 as though it were connected by an old-fashioned serial port.

We will do this with a program in python. The python program uses a graphical user interface called gtk to draw windows on the screen, and a plotting library called matplotlib to make graphs. We will need to install these components.

Components needed are: Anaconda 3 with Python 3.7 which you most likely have from other courses

https://phas.ubc.ca/~kotlicki/Physics_319/temperature_demo.zip
Ensure that the two left-most jumpers (circled in the photo) are oriented horizontally (opposite to all the other jumpers).



Build the program in main.c and flash into the Launchpad.

You will need to edit the python program to contain the correct serial port name.

On Windows this might be COM3, COM5 or COM7 But you should be able to confirm which port it is in the ports item of the device manager.

Then you can start the python program load it into Spyder environment and press RUN

start it with: `python python-serial-          .py`

After it has started, push button s2 on the Launchpad, and temperature measurements will be delivered from the Launchpad to the python program and printed.

If you don't see data coming onto the plot immediately after pushing the button, check that the jumpers circled in the image are oriented correctly.

After you see the temperature data being printed in the console switch to:
python-serial-plot-new.py
After it has started, push button s2 on the Launchpad, and temperature measurements will be delivered from the Launchpad to the python program and plotted.