Particle Directions

Particle and Visual Studio

1. Download Particle Workbench at https://www.particle.io/workbench/ and create an account as a student using your email
2. Download visual studio (the one with the blue icon, not purple) at https://code.visualstudio.com/
3. Log into Particle on Visual Studio, if it doesn't automatically (this is important because it won't let you install necessary libraries without being logged in)

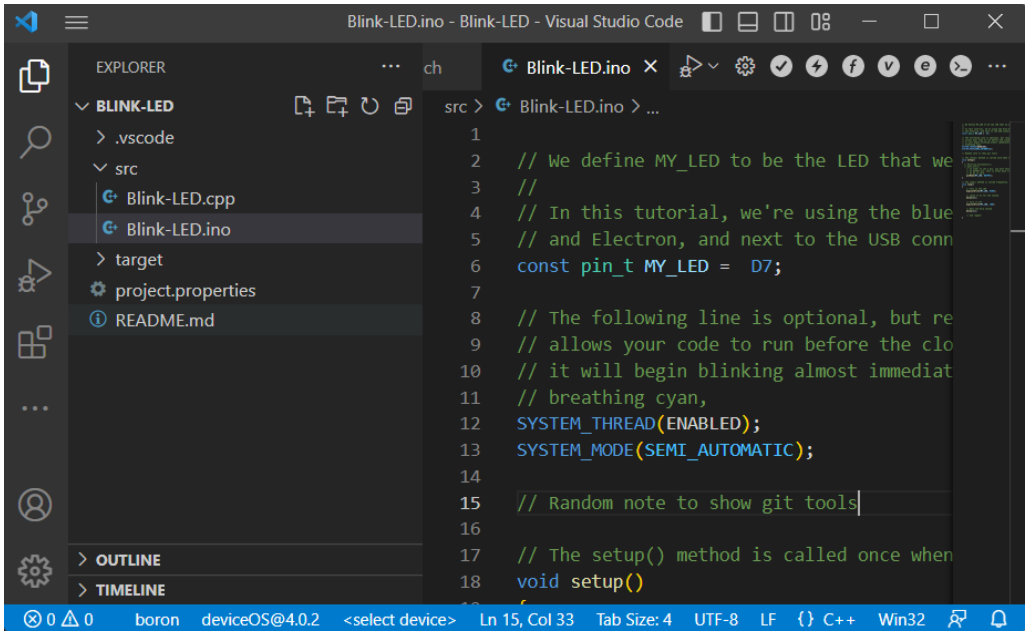
Useful commands

1. Open Particle command palette: Ctrl+Shift+P on Windows (or select View tab -> command palette, or click "Launch command palette" button on the Welcome tab)
2. In the top right corner, there are some important symbols in white circles
    - Compile: checkmark
    - Flash: lightning bolt

Practice code with Boron

1. Plug in Boron to your computer with correct chord
2. Create a folder on your computer to store Particle related items (no spaces in title)
3. Create a new project by opening the Particle command palette and selecting "Particle: Create New Project." You can also click the blue text that says, "Create new project" on the welcome page under Development workflow -> Code
4. Choose the folder you just made on your computer for this class when it asks for the parent folder
5. Once you have the directory in which you want to store your Particle projects, create a project name (no spaces). Something like Blink_LED.
6. Copy the code from the file named Blink_LED.ino (not the one you just made but from the example at https://github.com/SUPScientist/Smart-Coasts/blob/main/Class-01-Intro/Blink_LED.ino)
7. Navigate to the .ino file that is generated for your new project under the src tab on the left-hand side of screen (under explorer). Delete all the text there and past the coding you just copied
8. Open the command palette. Type "particle" and select "Particle Launch CLI"
9. Put the Boron into DFU (device firmware update) mode by pressing and holding the Mode button on the Boron while pressing and releasing the Reset button. Continue holding Mode until the RGB (Red Green Blue) LED starts flashing yellow.

10. Type “particle update” and press enter in the CLI (command line interface) that pops up. This should update your Device OS (mine updated to 4.0.2)
11. Check your Device OS by putting the Boron in listening mode (press and hold Mode until the LED blinks blue) and typing the following command into the CLI: particle identify (4.0.2 for me)
12. Configure Workbench to work with the Device OS that you confirmed you're using in the previous step. Your settings should be in the blue line across the bottom of the page. Make sure the first setting says “boron” and the second says “deviceOS@[4.0.2 or your specific number]” by clicking and changing them if need be



13. Open the command palette and run Particle: flash application (local) to compile and flash your script (your .ino file) to your device. If successful, this will result in the blinking of a blue LED close to the micro-USB socket on the Boron

Practice code with Boron and Adalogger

1. Plug in the boron and make sure the settings say “boron” again
2. Create a new project (I named mine RTCtest)
3. Open command palette and type “Particle: Install Library”
4. Type in SdFat and press enter to install SdFat library
5. A new tab titled “lib” should now be under the explorer tab for this project on the left, open it
6. Click on examples -> ReadWrite -> ReadWrite.ino
7. Copy all the coding on this page
8. Now click the tab that says “src” on the left, then “[your project title].ino” (not the .cpp)

a. The .ino file will always be the file you will want to use to flash coding

9. Highlight and delete everything in the .ino file, replace it with the new coding you just copied from the ReadWrite.ino file
10. In line 25, change “#define SD_CS_PIN SS” to “#define SD_CS_PIN D5” (SS -> D5)
11. After “File myFile;” create a new line and type “SYSTEM_MODE(MANUAL);”
12. After “void setup(){,“ create a new line and type “Cellular.off();” and in the next line type “delay(2500);”
13. Where it says “if (!SD.begin(SD_CS_PIN,)) {“ type “if (!SD.begin(SD_CS_PIN,SPI_FULL_SPEED)) {“

```
25  #define SD_CS_PIN D5
26  File myFile;
27  SYSTEM_MODE(MANUAL);
28
29  void setup() {
30
31    Cellular.off();
32    delay(2500);
33
34    // Open serial communications and wait for port to open:
35    Serial.begin(9600);
36    while (!Serial) {
37      ; // wait for serial port to connect. Needed for native USB port only
38    }
39
40
41    Serial.print("Initializing SD card...");
42
43    if (!SD.begin(SD_CS_PIN,SPI_FULL_SPEED)) {
44      Serial.println("initialization failed!");
45      return;
```

14. Click the checkmark to compile the project (will take a minute)
15. Once it compiles, click the lightning bolt button to flash the code to the Boron
    a. Boron should flash yellow/green when being put into DFU mode, then should breathe a white/blue light once it’s flashed
    b. Should also say it flashed successfully, unplugging and re-plugging chord into Boron can sometimes help if it doesn’t flash
16. Open the serial monitor: command palette -> “Particle: Serial monitor”

17. If the adalogger does not work, serial monitor may read something like the first chunk of text below, if it does work, it should read like the second chunk of text

```
Serial connection closed.  Attempting to reconnect...
Serial monitor opened successfully:
econnect...
Serial monitor opened successfully:
Initializing SD card...initialization failed!


Serial connection closed.  Attempting to reconnect...
Serial monitor opened successfully:
Initializing SD card...initialization done.
Writing to test.txt...done.
test.txt:
testing 1, 2, 3.
```

Practice water level sensor full code (Boron and Adalogger)

1. On the GitHub page, go to Firmware -> SLR_Boron_Maxbotix_MB7092_cm -> src -> SLR_Boron_Maxbotix_MB7092_cm.ino (or visit https://github.com/SUPScientist/Seaport_Tide-SLR/blob/main/Firmware/SLR_Boron_Maxbotix_MB7092_cm/src/SLR_Boron_Maxbotix_MB7092_cm.ino)
2. Copy all the coding on this page
3. In particle workbench, create a new project
4. Go to the .ino file and delete the coding and paste the new coding you copied from GitHub there
5. In line 4, where it says “const int SD_CHIP_SELECT =SS” change the SS to D5
6. In lines 32 and 33, change the comments so that 32 is uncommented (delete the double slashes in front) and 33 is commented out (add double slashes in front)

```
31  //------------------Turn off cellular for prelim testing; turn on for deployment
32  //SYSTEM_MODE(MANUAL); // uncomment for prelim testing
33  SYSTEM_MODE(SEMI_AUTOMATIC); // uncomment for deployment
34  SYSTEM_THREAD(ENABLED);
```

7. In line 45, where it says “const unsigned long MAX_TIME_TO_PUBLISH_MS = 60000” change the 60000 to 20000
8. In lines 49 and 50, swap the comments so that 49 is commented out and 50 is uncommented

```
49      Particle.connect();
50      //Cellular.off(); // turn off cellular for prelim testing (uncomment)
```

9. Comment out lines 148–153. You can select all the text in those lines and use the shortcut Ctrl+/

```
148     //connect particle to the cloud
149     if (Particle.connected() == false)
150     {
151       Particle.connect();
152       Serial.print("Trying to connect");
153     }
```

10. In line 208, where it says ".duration(54min)" change the 54 to 1
11. After line 138 that says "myFile.print(range_cm);" create a new line and make sure it's in line with the previous line and type "myFile.print(";");" This will make the data easier to read
12. Compile and flash code to boron
13. Open the serial monitor
14. Give it a second, but the terminal should look something like the picture below
    a. If it still doesn't work, try pressing the reset button on the Boron, or unplug and re-plug chord in
15. It should publish the time and distance once before repeating the phrase "not max time, try again to publish"

```
PS C:\Users\hillc\OneDrive\Documents\Particle\Project> particle serial monitor
Polling for available serial device...
Opening serial monitor for com port: "COM7"
Serial monitor opened successfully:
Maxbotix Test
Time: 946684815, Distance(cm): 76.55Not max time, try again to publish
Not max time, try again to publish
Not max time, try again to publish
Not max time, try again to publish
```

16. After 20 seconds (because of the value you changed from 60000 to 20000), the serial monitor should say "serial connection closed. Attempting to reconnect…"
17. After 1 minute (because of the value you changed from 54min to 1min), the serial monitor should say "serial monitor opened successfully"
18. Unplug the boron to stop the code from running
19. To check all the collected values, take out the SD card from the adalogger and put it into a SD card reader to then plug into your computer
20. Navigate to file explorer -> this PC -> USB drive -> distance. You should see values like below

```
946684815,15330,78.56;946684906,106280,78.52;946684997,197230,77.71;
```

21. Unix time values are listed first, then another value, and then the distance(cm) values followed by a semi-colon