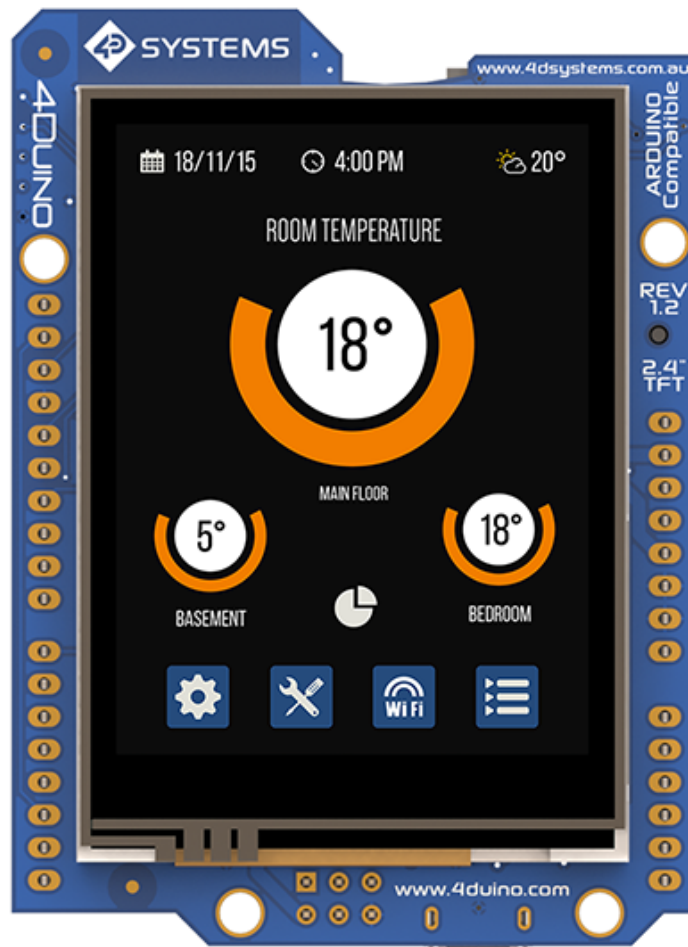


How to Integrate a 4D System's 4Duino LCD Display



Introduction

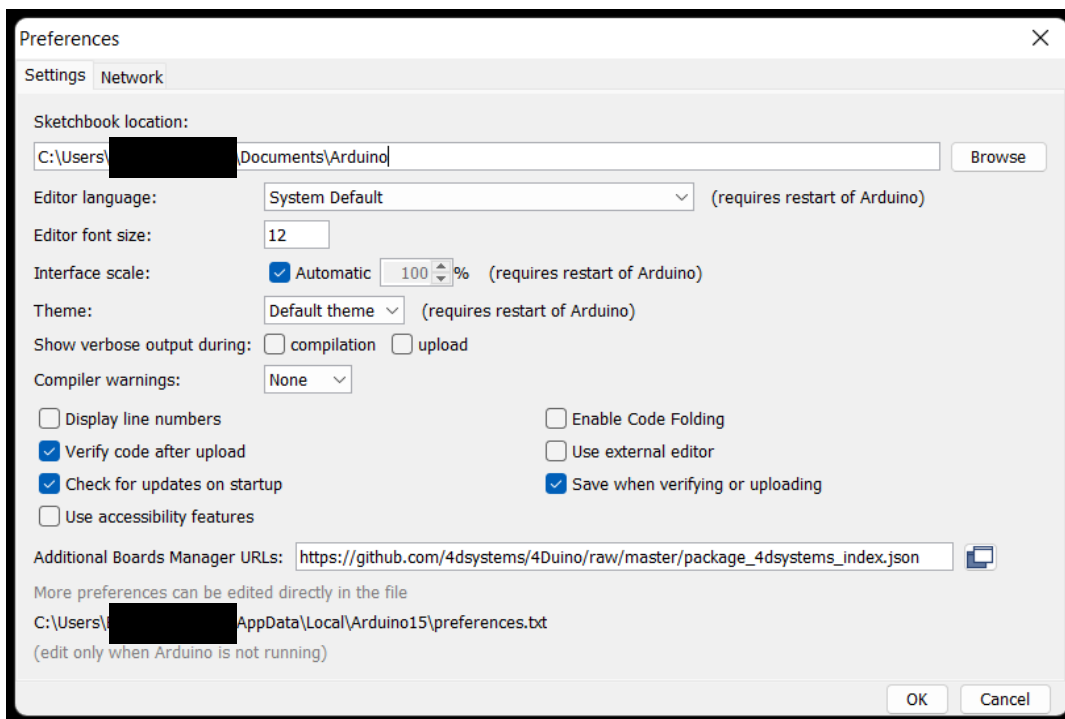
This guide will walk you through the steps for integrating a 4D System's 4Duino into your Beagle Bone based embedded system. This will include setting up your computer to be able to program a 4Duino as well as display a colored rectangle and establish I2C communications with the Beagle Bone. The 4Duino is equipped with a 2.4", 240x320 LCD display. Onboard, the 4Duino has an ATmega32u4 microprocessor making it similar, at heart, to Arduino's Leonardo board. It also has a PICASO GPU. The 4Duino can be programmed using either the 4D Workshop4 IDE or the Arduino IDE. In this guide we'll be using the 4D Workshop4 IDE.

Arduino IDE

Even though we are using the 4D Workshop4 IDE, the Arduino IDE is still needed in order to be able to program your 4Duino board. To start, download and install the Arduino IDE. The 4D Workshop4 IDE requires a 32 bit version of the Arduino IDE so be sure to download the legacy version (1.8.X and older).

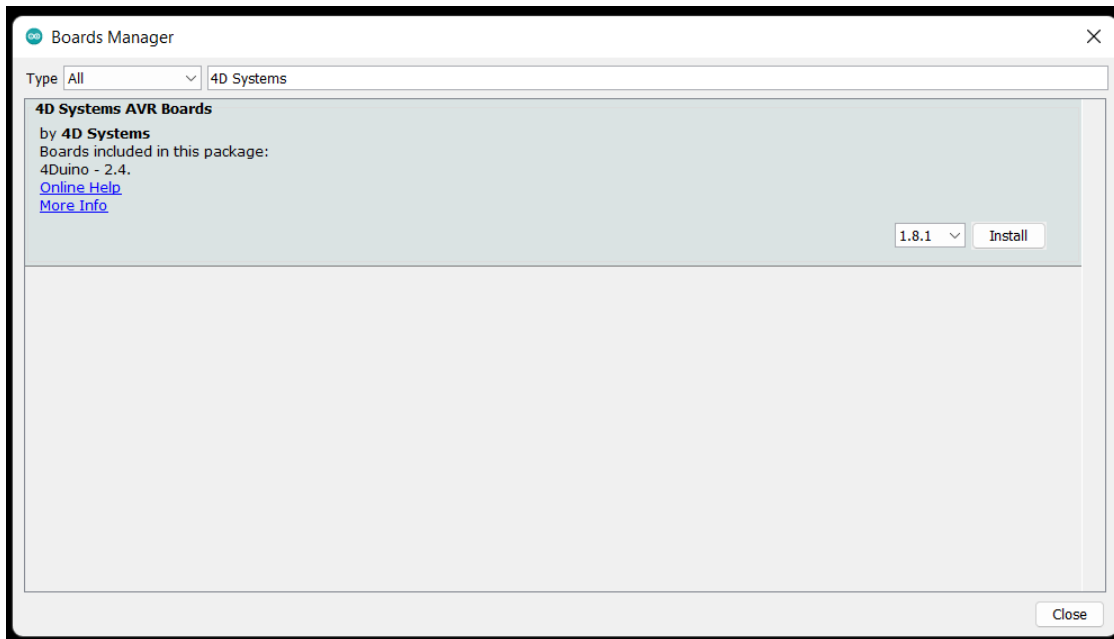
Once installed, open the arduino IDE and add the 4Duino board to the boards manager. To do so, navigate to "File > Preferences", and under the settings tab, add the following URL to the "Additional Boards Manager URLs".

https://github.com/4dsystems/4Duino/raw/master/package_4dsystems_index.json

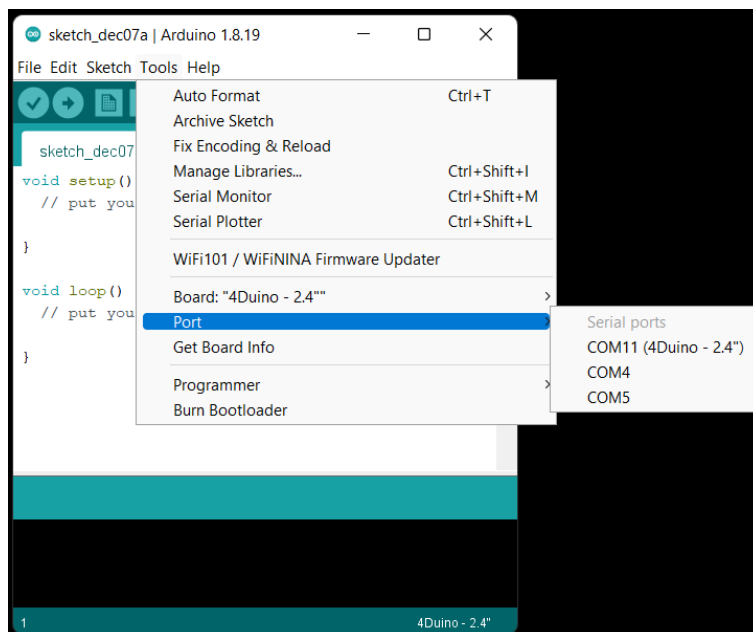


How to Integrate a 4D System's 4Duino LCD Display

Next, navigate to “Tools > Board > Board Manager”, and search “4D systems” in the search bar. Install the “4D Systems AVR Boards” package.

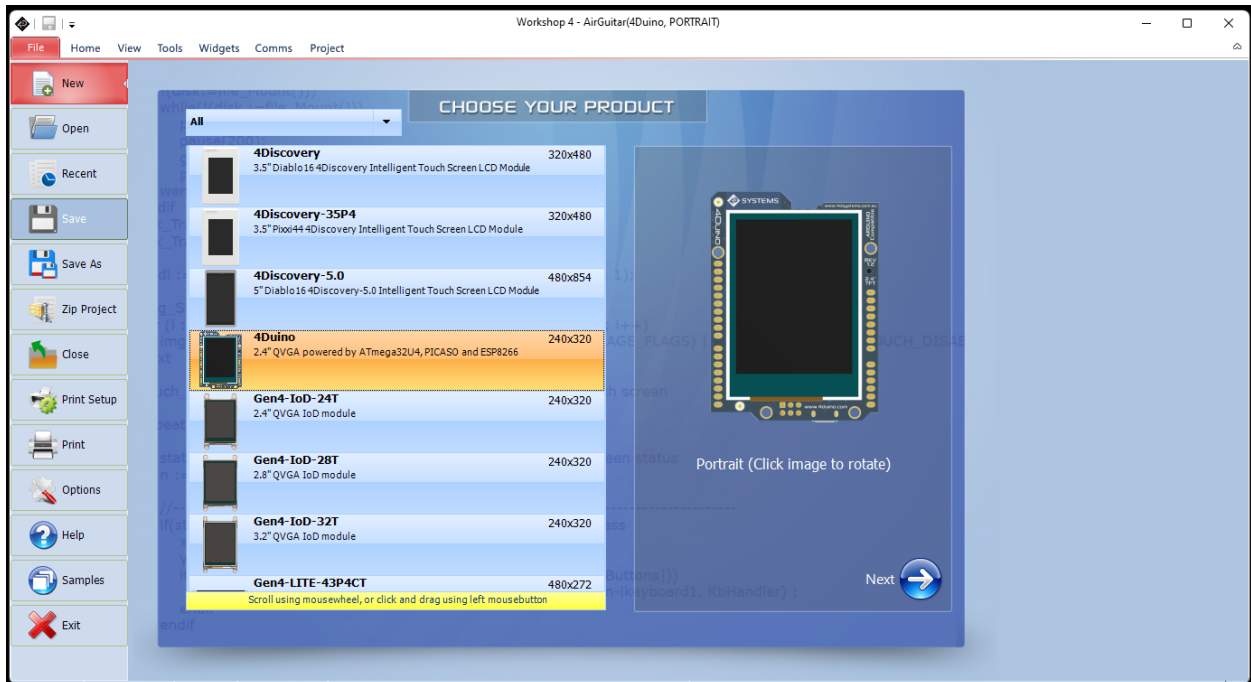


As a final step, determine what com port your 4Duino board is associated with. If you haven't yet, plug your 4Duino board into your computer. Then, navigate to “Tools > Port” and make note of which port is labeled with “4Duino - 2.4”. Close the Arduino IDE. If left open, the Arduino IDE may hold onto the 4Duino's com port, preventing the 4D Workshop4 IDE from being able to use it.



Display a colored rectangle

Create a new project in 4D Workshop4 IDE by clicking on “File > New > 4Duino”.



Click next and select the basic graphics option. You are now ready to begin coding. Add the line “Display.gfx_RectangleFilled(0, 0, 100, 100, RED);” to the automatically created loop function at the bottom of the project.

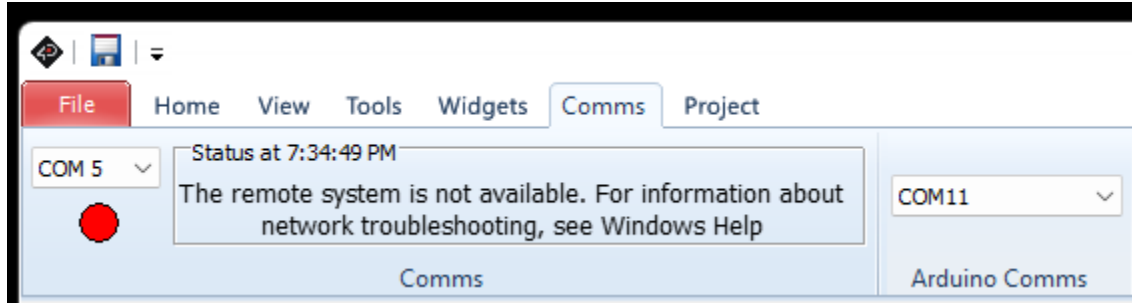
```
void loop() {  
  Display.gfx_RectangleFilled(0, 0, 100, 100, RED);  
}
```

This function displays a red rectangle in the upper left corner of the display. The parameters to the Display.gfx_RectangleFilled function are as follows:

- The x coordinate of the rectangle's upper left corner
- The y coordinate of the rectangle's upper left corner
- The x coordinate of the rectangle's lower right corner
- The y coordinate of the rectangle's lower right corner
- The color of the rectangle

Compiling and loading

In the menu at the top of the 4D Workshop4 IDE, select Comms. In the Arduino Comms section, select the com port from the Arduino IDE section above.



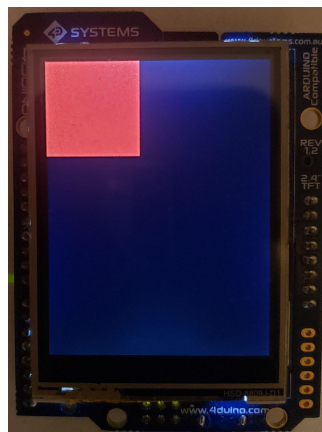
Finally, back in the home menu, click on "Comp 'nLoad"



After some time you should see the following message in the terminal as well as a red rectangle in the upper left corner of the display.

```
Loading configuration...
Initializing packages...
Preparing boards...
Verifying...
Sketch uses 7526 bytes (26%) of program storage space. Maximum is 28672 bytes.
Global variables use 518 bytes (20%) of dynamic memory, leaving 2042 bytes for local variables. Maximum is 2560 bytes.
Uploading...

Completed RC=0
```



I2C

You will need some way of communicating with the 4Duino from your Beagle Bone. Luckily, the 4Duino is equipped with I2C. Add the line “#include <Wire.h>” to your 4Duino code.

```
#include <Wire.h>
```

There will be an automatically generated function called “setup” in your 4Duino project. At the very end of this function, add the lines “Wire.begin(0x08);” and “Wire.onReceive(bitMaskRecieved);”.

```
Wire.begin(0x08);  
Wire.onReceive(bitMaskRecieved);  
} // end Setup **do not alter, remove or duplicate this line**
```

The parameter for Wire.begin is the I2C address of your 4Duino board and can be changed to whatever you want. The parameter for Wire.onReceive is a callback function pointer to a function that gets called once an I2C message is received by the 4Duino. You can also change this to whatever you want as long as the function is defined later in your code. Below is an example function to get you started. “Wire.available” checks whether the I2C bus is available for communications and “Wire.read” returns whatever was sent to the 4Duino.

```
444 void bitMaskRecieved(int numBytes) {  
445     while(Wire.available()) { // read all bytes received  
446         noteBitMask = Wire.read();  
447         // do something  
448     }  
449 }
```

Troubleshooting

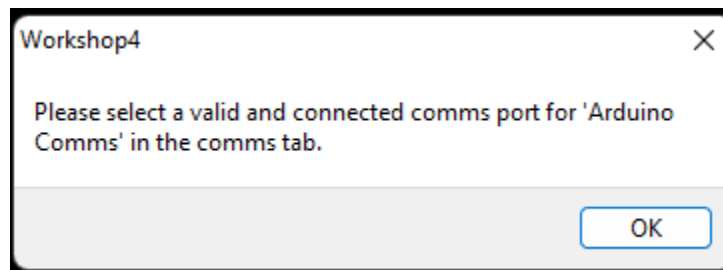
An error occurred while uploading the sketch

If you encounter the error message “An error occurred while uploading the sketch” in the 4D Workshop4 IDE after trying to compile and load your project, this means that connection to the board was lost during the upload process. During the process of compiling and uploading your program, the 4Duino disconnects and reconnects automatically a couple times. This is normal, but if you are using a virtual machine such as VMWare at the same time, when the 4Duino tries to reconnect it may get stopped by the virtual machine asking whether to connect it to the host or virtual machine. If you don't manually tell VMWare to connect the 4Duino to the host, 4D Workshop4 IDE will time out and assume connection to the 4Duino was lost. Also note that the VMWare connection message may pop up behind the 4D Workshop4 IDE window and can be easily missed.

```
Loading configuration...
Initializing packages...
Preparing boards...
Verifying...
Sketch uses 8100 bytes (28%) of program storage space. Maximum is 28672 bytes.
Global variables use 534 bytes (20%) of dynamic memory, leaving 2026 bytes for local variables. Maximum is 2560 bytes.
Uploading...
An error occurred while uploading the sketch
Completed RC=1
```

Select a valid comms port

If this error pops up on screen after you've tried to compile and load your program, this could mean that you don't have the correct com port selected in the comms menu, or it could mean that another program is using the 4Duino com port. If you believe that you have the wrong com port selected, open the arduino IDE, navigate to “Tools > Port” and make note of which port is labeled with “4Duino - 2.4”. Close the Arduino IDE and any other programs that may be using the 4Duino com port such as additional instances of 4D Workshop IDE.



References

[1] 4DUINO-24. 4D Systems. (n.d.). Retrieved December 7, 2022, from <https://4dsystems.com.au/catalog/product/view/id/804>

[2] 4D Systems. (2020, May 26). Arduino™ Compatible Display Module DATASHEET, from <https://4dsystems.com.au/mwdownloads/download/link/id/166/>