

## Application Notes for using EVE modules with Gameduino2 library

**Tested Platform Version:** Gameduino2 Library 1.3.3

The purpose of this document is to explain the steps required to use **Version 1.3.3** of the Gameduino2 library with any of our EVE TFT modules. The Gameduino2 library has been written by James Bowman for the FT80x/FT81x graphic controller IC series and contains support for various microcontrollers, along with many example sketches ranging from a simple “Hello World” program to more complicated programs using various widgets, accessing PNG files from a microSD card, and generating board game simulations. In this App Notes document, we will be using an Arduino Uno to drive our NHD-4.3-480272FT-CSXP-CTP EVE module and list the changes necessary in order to compile the basic ‘Hello World’ sketch.

### Applicable Displays

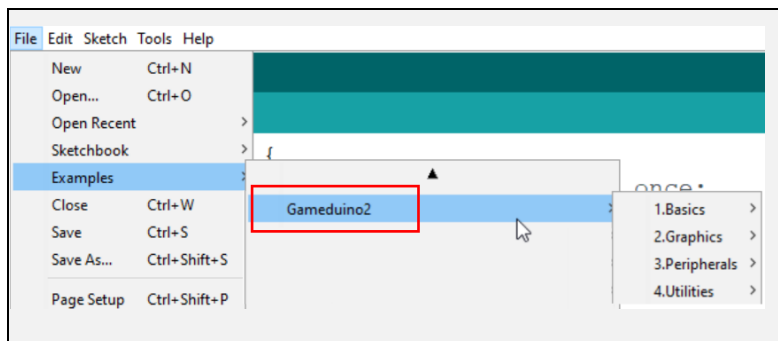
3.5" TFT	4.3" TFT	5.0" TFT	7.0" TFT
NHD-3.5-320240FT-CSXP-T	NHD-4.3-800480FT-CSXP-CTP NHD-4.3-480272FT-CSXP-CTP NHD-4.3-480272FT-CSXP-T	NHD-5.0-800480FT-CSXP-CTP NHD-5.0-800480FT-CTXL-CTP NHD-5.0-800480FT-CTXL-T	NHD-7.0-800480FT-CSXP-CTP NHD-7.0-800480FT-CSXV-CTP NHD-7.0-800480FT-CSXP-T NHD-7.0-800480FT-CSXV-T

### Requirements:

- Newhaven Display [NHD-4.3-480272FT-CSXP-CTP](#)
- Newhaven Display [20 POS FFC](#)
- Newhaven Display [NHD-FT81x-SHIELD](#)
- Arduino Uno
- USB 2.0 Cable Type A/B
- Arduino IDE tool installed
- 9V Power Supply
- Gameduino2 Library
- [Eve Asset Builder](#) (For images)

Once the library is [downloaded](#), add the library to your **Arduino IDE**. For more details, especially for first-time library installers, see this tutorial at: <https://www.arduino.cc/en/guide/libraries>.

After installing the library, it should show in **File-> Examples->Gameduino2**



## Hardware Setup

Make sure the NHD-SHIELD has J4-J6 shorted and J1-J3 Open for Arduino Uno Connection.

Connect the 9V power supply to the Arduino Uno.



## Modifying Gameduino2 Library

Before running any of the example sketches, GD2.cpp file needs to be updated by commenting a line of code in order to make the Gameduino2 library compatible with our EVE displays:

Navigate to the Documents-> Arduino-> libraries-> Gameduino2-> GD2.cpp

Open this file with your preferred text editor.

Before	Comment The Line of Code	After
<pre>void GDClass::tune(void) {   uint32_t f;   for (byte i = 0; (i &lt; 31) &amp;&amp; ((f = measure_freq()) &lt; LOW_FREQ_BOUND); i++) {     GDTR.wr(REG_TRIM, i);   }   GDTR.wr32(REG_FREQUENCY, f); }  void GDClass::begin(uint8_t options, int cs, int sdcs) {   #if defined(ARDUINO)    defined(ESP8266)    defined(ESP32)    defined(SPIDRIVER)   GDTR.begin0(cs);   if (STORAGE &amp;&amp; (options &amp; GD_STORAGE))     SD.begin(sdcs);   #endif }</pre>	<p>Comment The Line of Code</p>	<pre>void GDClass::tune(void) {   uint32_t f;   for (byte i = 0; (i &lt; 31) &amp;&amp; ((f = measure_freq()) &lt; LOW_FREQ_BOUND); i++) {     GDTR.wr(REG_TRIM, i);   }   GDTR.wr32(REG_FREQUENCY, f); }  void GDClass::begin(uint8_t options, int cs, int sdcs) {   #if defined(ARDUINO)    defined(ESP8266)    defined(ESP32)    defined(SPIDRIVER)   GDTR.begin0(cs);   //if (STORAGE &amp;&amp; (options &amp; GD_STORAGE))   SD.begin(sdcs);   #endif }</pre>

Save the changes and closed the modified file.

## EVE TFT Registers

The required registers values for each respective display can be copied and pasted from the table below

3.5" TFT 320 x 240	4.3" TFT 480 x 272	4.3" TFT 800 x 480	5.0" TFT 800 x 480	7.0" TFT 800 x 480
GD.wr16(REG_HSIZE, 320);	GD.wr16(REG_HSIZE, 480);	GD.wr16(REG_HSIZE, 800);	GD.wr16(REG_HSIZE, 800);	GD.wr16(REG_HSIZE, 800);
GD.wr16(REG_HCYCLE, 408);	GD.wr16(REG_HCYCLE, 548);	GD.wr16(REG_HCYCLE, 928);	GD.wr16(REG_HCYCLE, 928);	GD.wr16(REG_HCYCLE, 928);
GD.wr16(REG_HOFFSET, 70);	GD.wr16(REG_HOFFSET, 43);	GD.wr16(REG_HOFFSET, 88);	GD.wr16(REG_HOFFSET, 88);	GD.wr16(REG_HOFFSET, 88);
GD.wr16(REG_HSYNCO, 0);	GD.wr16(REG_HSYNCO, 0);	GD.wr16(REG_HSYNCO, 0);	GD.wr16(REG_HSYNCO, 0);	GD.wr16(REG_HSYNCO, 0);
GD.wr16(REG_HSYNC1, 10);	GD.wr16(REG_HSYNC1, 41);	GD.wr16(REG_HSYNC1, 48);	GD.wr16(REG_HSYNC1, 48);	GD.wr16(REG_HSYNC1, 48);
GD.wr16(REG_VSIZE, 240);	GD.wr16(REG_VSIZE, 272);	GD.wr16(REG_VSIZE, 480);	GD.wr16(REG_VSIZE, 480);	GD.wr16(REG_VSIZE, 480);
GD.wr16(REG_VCYCLE, 263);	GD.wr16(REG_VCYCLE, 292);	GD.wr16(REG_VCYCLE, 525);	GD.wr16(REG_VCYCLE, 525);	GD.wr16(REG_VCYCLE, 525);
GD.wr16(REG_VOFFSET, 13);	GD.wr16(REG_VOFFSET, 12);	GD.wr16(REG_VOFFSET, 32);	GD.wr16(REG_VOFFSET, 32);	GD.wr16(REG_VOFFSET, 32);
GD.wr16(REG_VSYNCO, 0);	GD.wr16(REG_VSYNCO, 0);	GD.wr16(REG_VSYNCO, 0);	GD.wr16(REG_VSYNCO, 0);	GD.wr16(REG_VSYNCO, 0);
GD.wr16(REG_VSYNC1, 2);	GD.wr16(REG_VSYNC1, 10);	GD.wr16(REG_VSYNC1, 3);	GD.wr16(REG_VSYNC1, 3);	GD.wr16(REG_VSYNC1, 3);
GD.wr16(REG_PCLK, 8);	GD.wr16(REG_PCLK, 5);	GD.wr16(REG_PCLK, 2);	GD.wr16(REG_PCLK, 2);	GD.wr16(REG_PCLK, 2);
GD.wr16(REG_SWIZZLE, 0);	GD.wr16(REG_SWIZZLE, 0);	GD.wr16(REG_SWIZZLE, 0);	GD.wr16(REG_SWIZZLE, 0);	GD.wr16(REG_SWIZZLE, 0);
GD.wr16(REG_PCLK_POL, 0);	GD.wr16(REG_PCLK_POL, 1);	GD.wr16(REG_PCLK_POL, 1);	GD.wr16(REG_PCLK_POL, 0);	GD.wr16(REG_PCLK_POL, 1);
GD.wr16(REG_CSPREAD, 1);	GD.wr16(REG_CSPREAD, 1);	GD.wr16(REG_CSPREAD, 0);	GD.wr16(REG_CSPREAD, 0);	GD.wr16(REG_CSPREAD, 0);
GD.wr16(REG_DITHER, 1);	GD.wr16(REG_DITHER, 1);	GD.wr16(REG_DITHER, 1);	GD.wr16(REG_DITHER, 1);	GD.wr16(REG_DITHER, 1);
GD.wr16(REG_ROTATE, 0);	GD.wr16(REG_ROTATE, 0);	GD.wr16(REG_ROTATE, 0);	GD.wr16(REG_ROTATE, 0);	GD.wr16(REG_ROTATE, 0);



## Example: Hello World

Once the above modifications have been made to the respective file, proceed to test the 'Hello World' sketch:

**Note:** The following changes will reference our 4.3" EVE TFT timings / registers as an example guide.

Go to **File->Examples->Gameduino2->Basics->Hello World**

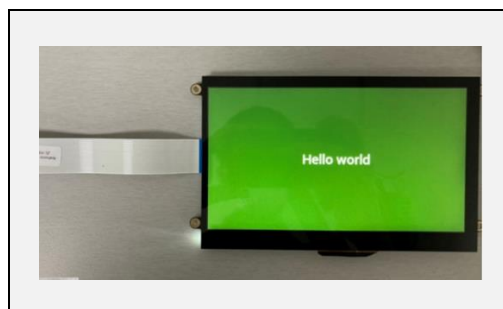
Before	After
<pre> helloworld 1 #include &lt;EEPROM.h&gt; 2 #include &lt;SPI.h&gt; 3 #include &lt;GD2.h&gt; 4 5 void setup() 6 { 7   GD.begin(0); 8 } 9 10 void loop() 11 { 12   GD.ClearColorRGB(0x103000); 13   GD.Clear(); 14   GD.cmd_text(GD.w / 2, GD.h / 2, 31, OPT_CENTER, "Hello world"); 15   GD.swap(); 16 }           </pre>	<p style="color: green; font-weight: bold;">Replace GD.begin(0); with GD.begin(0,10,5);</p> <pre> helloworld \$ 1 #include &lt;EEPROM.h&gt; 2 #include &lt;SPI.h&gt; 3 #include &lt;GD2.h&gt; 4 5 void setup() 6 { 7   GD.begin(0,10,5); 8 } 9 10 void loop() 11 { 12   GD.ClearColorRGB(0x103000); 13   GD.Clear(); 14   GD.cmd_text(GD.w / 2, GD.h / 2, 31, OPT_CENTER, "Hello world"); 15   GD.swap(); 16 }           </pre>

Obtain the values from the [EVE TFT Registers](#) table for your display size.

Add the function below and call it in the setup function:	
<pre> void TFT_4_3(){ GD.wr16(REG_HSIZE, 480); GD.wr16(REG_HCYCLE, 548); GD.wr16(REG_HOFFSET, 43); GD.wr16(REG_HSYNCO, 0); GD.wr16(REG_HSYNC1, 41); GD.wr16(REG_VSIZE, 272); GD.wr16(REG_VCYCLE, 292); GD.wr16(REG_VOFFSET, 12); GD.wr16(REG_VSYNCO, 0); GD.wr16(REG_VSYNC1, 10); GD.wr16(REG_PCLK, 5); GD.wr16(REG_SWIZZLE, 0); GD.wr16(REG_PCLK_POL, 1); GD.wr16(REG_CSPREAD, 1); GD.wr16(REG_DITHER, 1); GD.wr16(REG_ROTATE, 0); }           </pre>	<p style="color: green; font-weight: bold;">Final program should look like below before upload:</p> <pre> 1 #include &lt;EEPROM.h&gt; 2 #include &lt;SPI.h&gt; 3 #include &lt;GD2.h&gt; 4 5 void TFT_4_3() 6 { 7   GD.wr16(REG_HSIZE, 480); 8   GD.wr16(REG_HCYCLE, 548); 9   GD.wr16(REG_HOFFSET, 43); 10  GD.wr16(REG_HSYNCO, 0); 11  GD.wr16(REG_HSYNC1, 41); 12  GD.wr16(REG_VSIZE, 272); 13  GD.wr16(REG_VCYCLE, 292); 14  GD.wr16(REG_VOFFSET, 12); 15  GD.wr16(REG_VSYNCO, 0); 16  GD.wr16(REG_VSYNC1, 10); 17  GD.wr16(REG_PCLK, 5); 18  GD.wr16(REG_SWIZZLE, 0); 19  GD.wr16(REG_PCLK_POL, 1); 20  GD.wr16(REG_CSPREAD, 1); 21  GD.wr16(REG_DITHER, 1); 22  GD.wr16(REG_ROTATE, 0); 23  GD.swap(); 24 } 25 26 void setup() 27 { 28   GD.begin(0,10,5); 29   TFT_4_3(); 30 } 31 32 void loop() 33 { 34   GD.ClearColorRGB(0x103000); 35   GD.Clear(); 36   GD.cmd_text(GD.w / 2, GD.h / 2, 31, OPT_CENTER, "Hello world"); 37   GD.swap(); 38 }           </pre>

### Upload to Arduino Uno

If the modifications were made successfully, you should expect to see the following on the EVE TFT Display:



## Software Support

Please note all software related to the Gameduino2 library was designed by James Bowman, and are not created nor maintained by Newhaven Display International Inc. For any additional software support relating to the Gameduino2 library, please contact James Bowman.

## Reference Links

- [Gameduino2 Homepage](#)
- [Gameduino2 Github Page](#)
- [Gameduino2 Cookbook](#)
- [FTDI EVE Example Files](#)

---

**For additional support on using our TFT EVE series displays with the Gameduino2 library or questions about our other display products, please contact us through any of our technical support channels listed below:**

Email: [techsupport@newhavendisplay.com](mailto:techsupport@newhavendisplay.com)

Phone: (847) 844-8795

Forum: [https://www.newhavendisplay.com/NHD\\_forum](https://www.newhavendisplay.com/NHD_forum)

