



C++ Code Editor Tutorial

6 steps to success

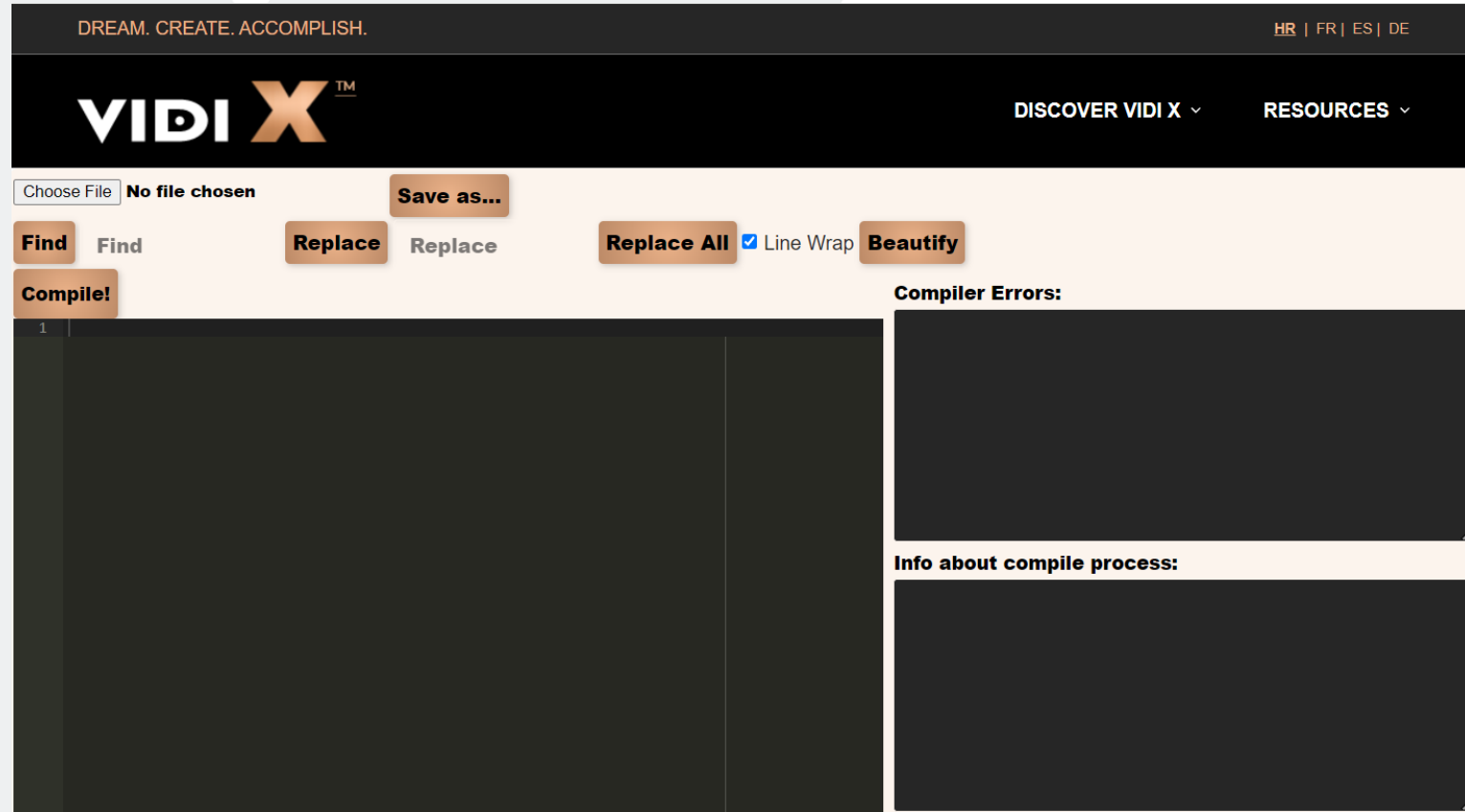
1. Open Code Editor
2. Connect VIDIX to computer
3. Input code
4. Compile & Upload
- 5. Reboot VIDIX**
6. Initialize code

1

Open Code Editor

Open this URL in Chrome or Edge:

code.vidi-x.org

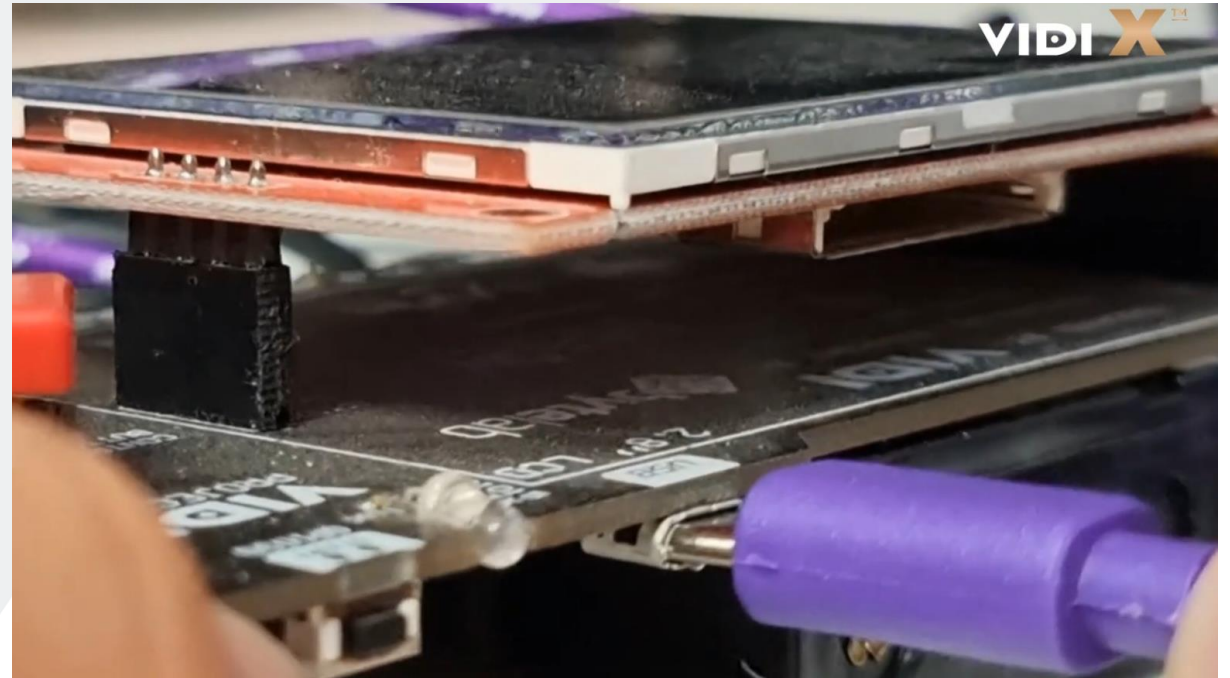


2

Connect VIDIX to USB

Connect the VIDIX to your computer with a **USB cable**.

Your operating system might signal that a new USB device has been connected.



3

Input the code

Upload an .ino file from your computer, by using „Choose File” button

Input the code:

Type the code or paste it from VIDI X EDU

The screenshot displays the VIDI X IDE interface. At the top, the logo "VIDI X™" is visible. Below the logo, there is a toolbar with several buttons: "Choose File" (with "No file chosen" text next to it), "Save as...", "Find" (with "Find" text next to it), "Replace" (with "Replace" text next to it), "Replace All", "Line Wrap" (checked), and "Beautify". Below the toolbar, there is a "Compile!" button. The main area shows a code editor with the following code:

```
1 int light = 2;
2 void setup() {
3     pinMode(light, OUTPUT);
4     digitalWrite(light, HIGH);
5     delay(1000);
6     digitalWrite(light, LOW);
7     delay(1000);
8 }
9 void loop() {
10 }
```

To the right of the code editor, there are two panels: "Compiler Errors:" and "Info about compile process:". Two orange arrows point to the "Choose File" button and the code editor area.

4.1

Compile C++ Code

Press the button **Compile!**



In the title tab of the Internet browser, a rotating circular clip appears, indicating the process has started.

The screenshot shows an IDE interface with a toolbar at the top containing buttons for 'Choose File', 'No file chosen', 'Save as...', 'Find', 'Find', 'Replace', 'Replace', 'Replace All', 'Line Wrap', and 'Beautify'. Below the toolbar is a 'Compile!' button. The main area displays C++ code:

```
1 int light = 2;
2
3 void setup() {
4   pinMode(light, OUTPUT);
5 }
6
7 void loop() {
8   digitalWrite(light, HIGH);
9   delay(1000);
10  digitalWrite(light, LOW);
11  delay(1000);
12 }
```

On the right side of the IDE, there is a circular progress indicator consisting of a white ring with a green segment at the bottom, indicating that the compilation process has started.

4.2

Code is ready for upload

After the compile process was completed, we can see the **UPLOAD** button with **red arrow**.

We also received information about compile process.

If we received an error message, we need to correct the code.

The screenshot shows the Arduino IDE interface. At the top, there are buttons for 'Choose File', 'No file chosen', 'Save as...', 'Find', 'Find', 'Replace', 'Replace', 'Replace All', 'Line Wrap', and 'Beautify'. Below these is the 'Upload!' button, which has a red arrow pointing to it. The code editor shows the following code:

```
1 int PinTemp = 26; //input pin for sensor
2 int temp; //variable for storing sensor value
3 void setup() { //beginning of setup function
4   pinMode(PinTemp, INPUT); //putting selected GPIO to input state
5   Serial.begin(9600); //starting serial communication
6 } //end of setup function
7 void loop() { //beginning of loop function
8   temp = analogRead(PinTemp); //storing sensor value into variable temp
9   Serial.print(temp); //printing out the sensor value on serial port
10  delay(1000); //delay of one second
11 } //end of loop functio
```

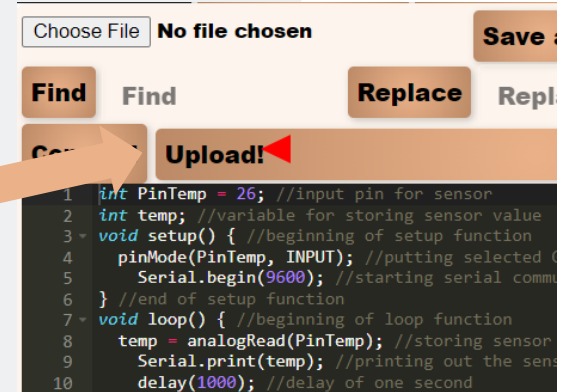
On the right side, there are two panels. The top panel is titled 'Compiler Errors:' and contains the text 'No Errors!'. The bottom panel is titled 'Info about compile process:' and contains the following information:

```
1:-> Sketch uses 283313 bytes (21%) of program storage
space. Maximum is 1310720 bytes.
2:-> Global variables use 21600 bytes (6%) of dynamic
memory, leaving 306080 bytes for local variables. Maximum is
327680 bytes.
3:->
4:-> Used platform
5:-> esp32:esp32 2.0.11
```

4.3

Click Upload!

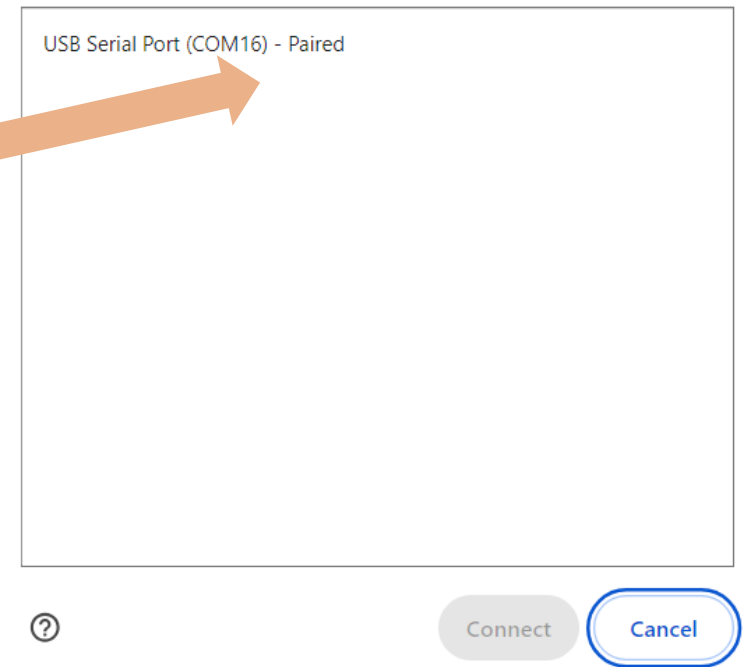
Click "**Upload!**" on code editor webpage.



A message will appear in a new window, that code.vidi-x.org wants to connect to a serial port.

The message window looks different on Mac or Chromebook – you might need to scroll down to find the appropriate port.

code.vidi-x.org wants to connect to a serial port

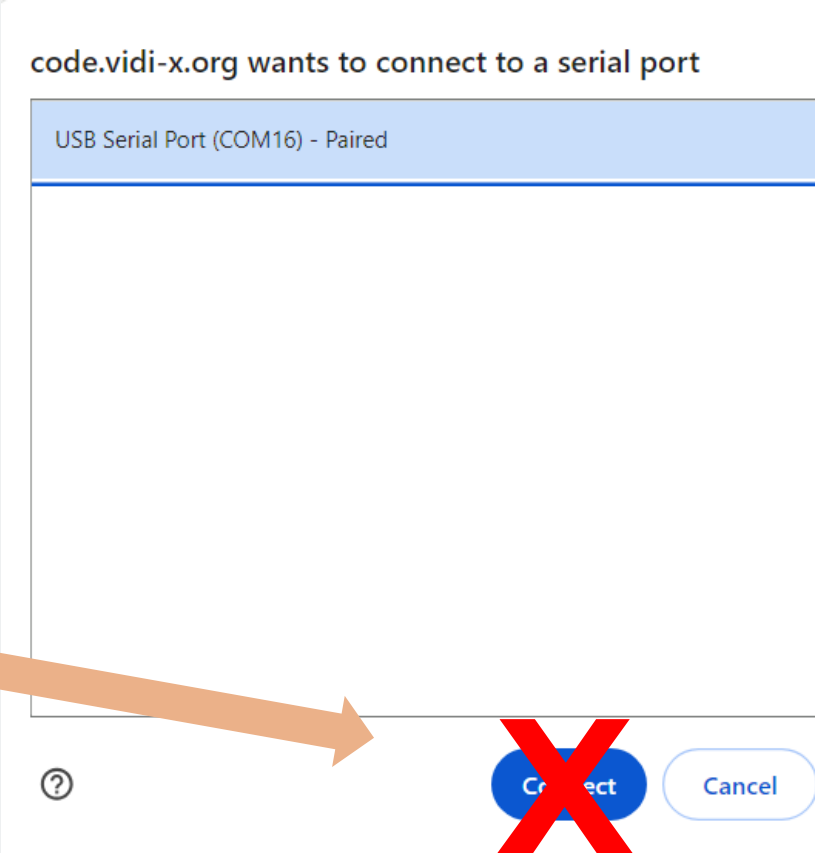


4.4

DO NOT Connect

Select the USB serial port that you used to connect VIDIX with your mouse.

Do NOT click Connect on the computer screen at this time.



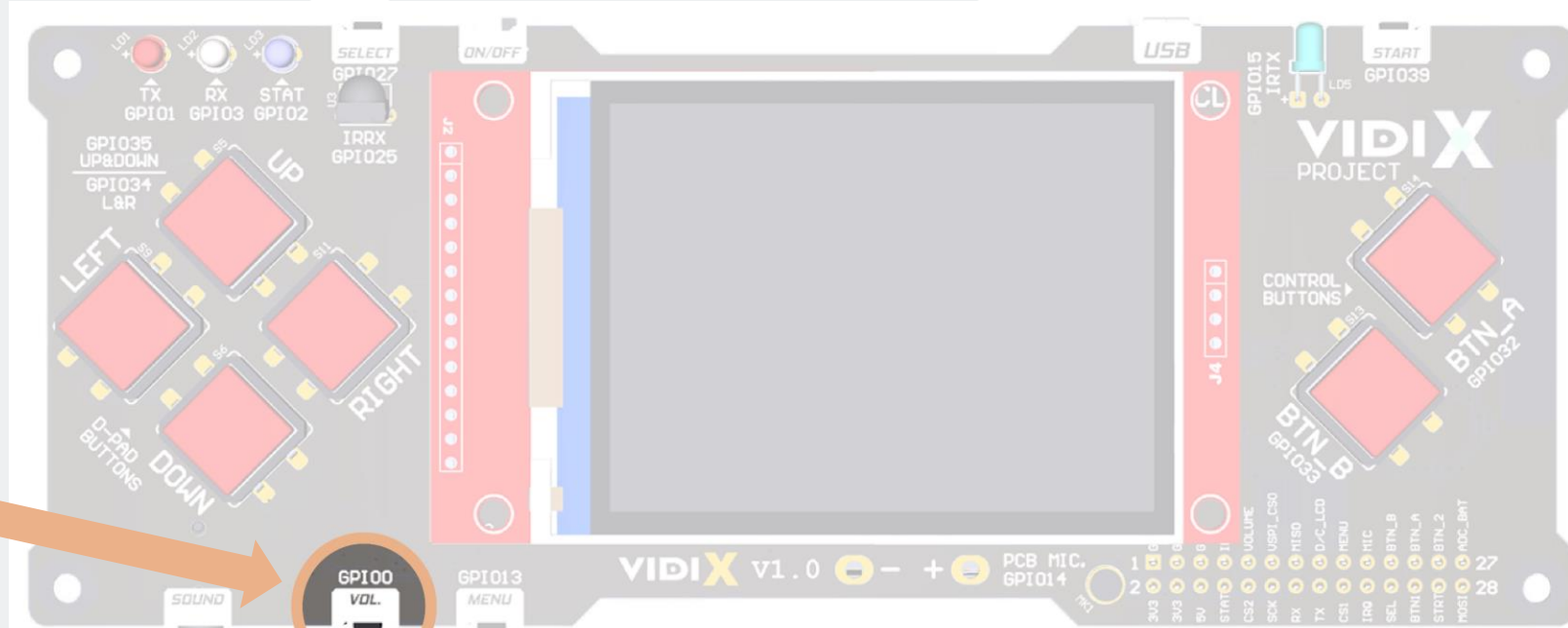
5.1

REBOOT: Press and hold VOL

Press and keep holding the "Vol." button.

Volume button, also marked as GPIO0.

Keep pressing the VOLUME button through entire step 5



VOL./BOOT Button
GPIO0

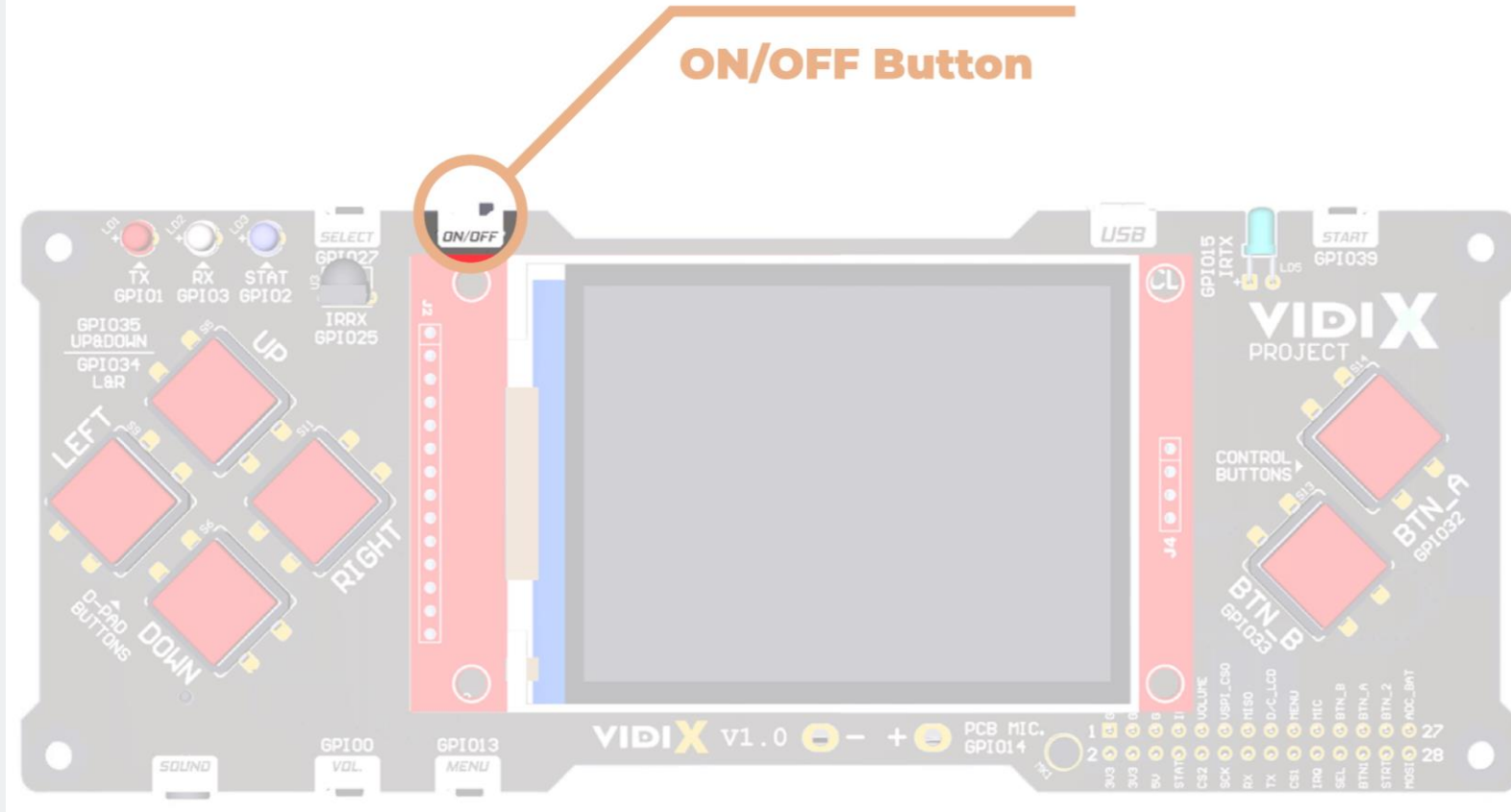
(frequent mistake – do not press the MENU button)

5.2

REBOOT: VIDI X OFF

Turn **VIDI X is OFF**, using the **on/off** button.

(keep holding the volume button)

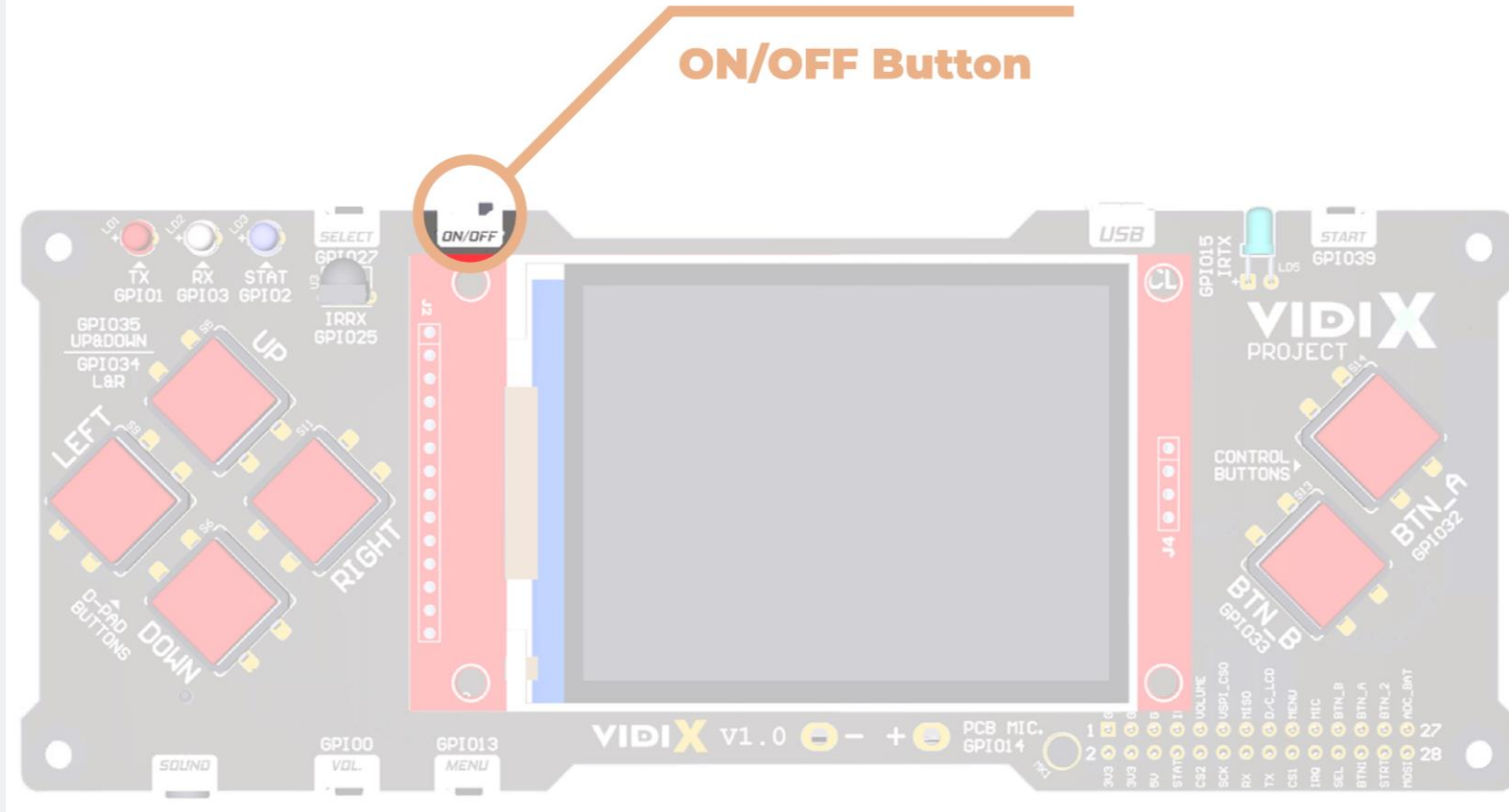


5.3

REBOOT: VIDI X ON

Turn the VIDI X back ON

(keep holding the volume button)

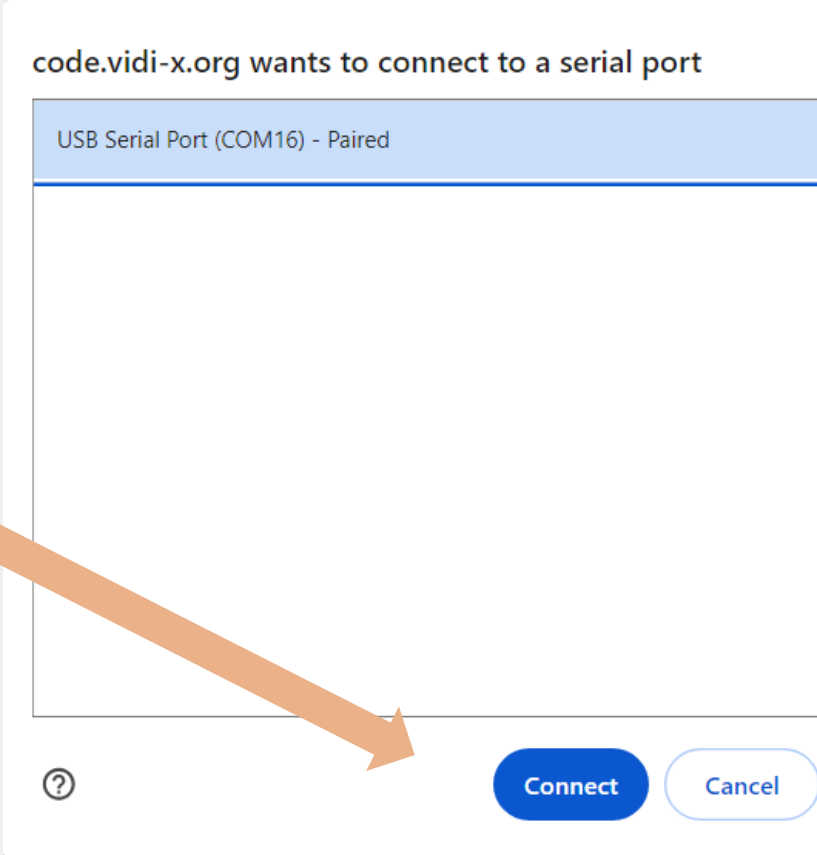


5.4

NOW click Connect

Click Connect while still holding the VOLUME button.

Remember the sequence of **step 5:**
VOL
OFF
ON
Connect



6

Initializing code on VIDIX

After the **Initializing...** message has changed and you start seeing writing progress percentages,

you can now **release the volume button**.

The screenshot shows the VIDIX Web Upload Tool interface. At the top, there is a file selection area with a 'Choose File' button and 'No file chosen' text. To the right is a 'Save as...' button. Below this are buttons for 'Find', 'Replace', 'Replace All', 'Line Wrap' (checked), and 'Beautify'. A large orange arrow points to the 'Upload!' button, which is highlighted in orange. Below the 'Upload!' button is a progress bar showing 'Writing progress: 20%'. Underneath the progress bar, it says 'VIDIX Web Upload Tool: ESP32'. At the bottom, there is a code editor with the following code:

```
1 #include <Adafruit_GFX.h>
2 #include <SPI.h>
3 #include <Wire.h>
4 #include <Adafruit_ILI9341.h>
5 #include <XPT2046_Touchscreen.h>
6
7 // This is calibration data for the raw touch data to the screen coordinates
8 #define TS_MINX 150
9 #define TS_MINY 130
10 #define TS_MAXX 3800
11 #define TS_MAXY 4000
12
13 // define touch screen CS pin
14 #define TS_CS 4
15 XPT2046_Touchscreen ts(TS_CS);
16
17 // display define display pins
18 #define TFT_CS 5
19 #define TFT_DC 21
```

Success!

When the code was successfully transferred, we received a message confirming it.

Your VIDIX microcomputer automatically runs the C++ code you compiled!

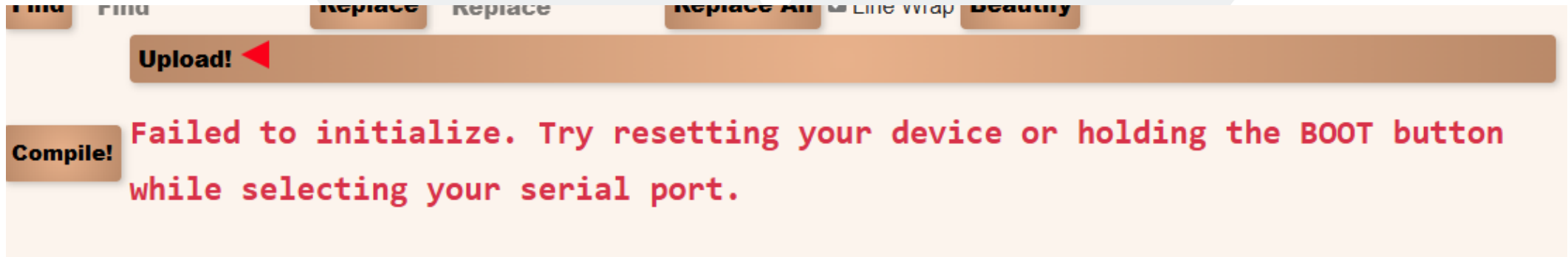
The screenshot displays the VIDIX Web Upload Tool interface for an ESP32. At the top, there are buttons for 'Choose File', 'No file chosen', 'Save as...', 'Find', 'Replace', 'Replace All', 'Line Wrap', and 'Beautify'. A prominent 'Upload!' button with a red arrow is highlighted. Below it, a 'Compile!' button is shown with the text 'All done!' in green. A status bar indicates 'VIDIX X Web Upload Tool: ESP32'. The main area contains a code editor with the following C++ code:

```
1 int PinTemp = 26; //input pin for sensor
2 int temp; //variable for storing sensor value
3 void setup() { //beginning of setup function
4   pinMode(PinTemp, INPUT); //putting selected GPIO to input state
5   Serial.begin(9600); //starting serial communication
6 } //end of setup function
7 void loop() { //beginning of loop function
8   temp = analogRead(PinTemp); //storing sensor value into variable temp
9   Serial.print(temp); //printing out the sensor value on serial port
10  delay(1000); //delay of one second
11 } //end of loop functio
```

On the right side, the 'Compiler Errors' section shows 'No Errors!'. Below that, the 'Info about compile process:' section provides the following details:

- 1:-> Sketch uses 283313 bytes (21%) of program storage space. Maximum is 1310720 bytes.
- 2:-> Global variables use 21600 bytes (6%) of dynamic memory, leaving 306080 bytes for local variables. Maximum is 327680 bytes.
- 3:->
- 4:-> Used platform
- 5:-> esp32:esp32 2.0.11

In case of an error message...



If you receive an **error message** in Code Editor: you will need to **change to code**, so it works.

If there is **problem with Initialization** process, just go **back to STEP 5** and reboot VIDI X.