Lab 3 Report – Ryan King

Lab 3 Part 2:

For part 2 I got help from ChatGPT in converting from Arduino.

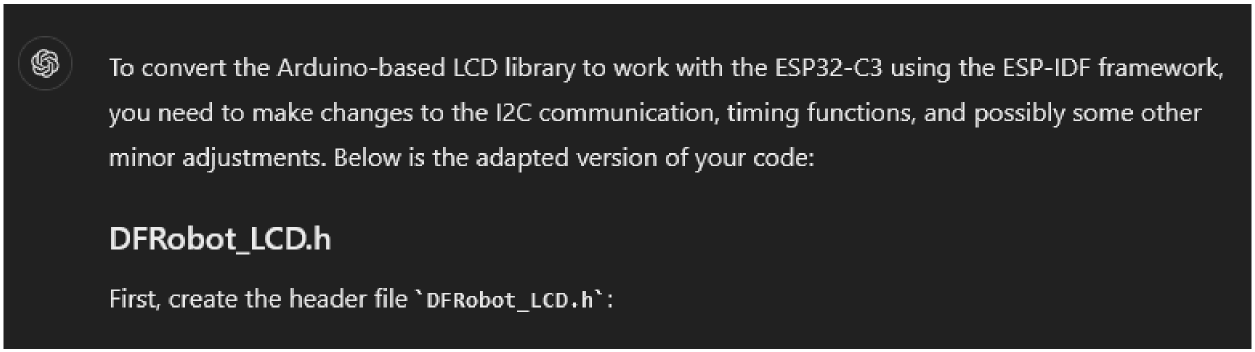
https://chatgpt.com/share/19da33b7-af4b-44de-bae8-1370603b8a3f

I also got help from this forum post.
https://esp32.com/viewtopic.php?t=21030

Lab 3 Part 3:

Since this was just integrating code already written, no web assistance was needed.

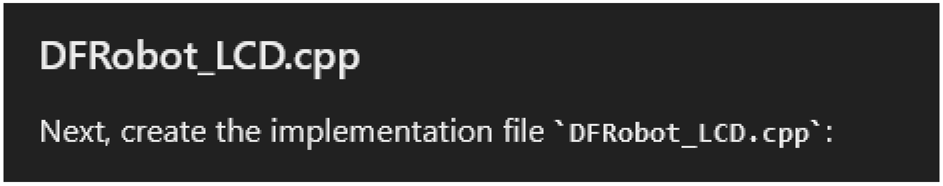
Code is not included because it would take forever to screenshot everything. It is included at the share link. Let me know if this report is not substantial enough.



To convert the Arduino-based LCD library to work with the ESP32-C3 using the ESP-IDF framework, you need to make changes to the I2C communication, timing functions, and possibly some other minor adjustments. Below is the adapted version of your code:

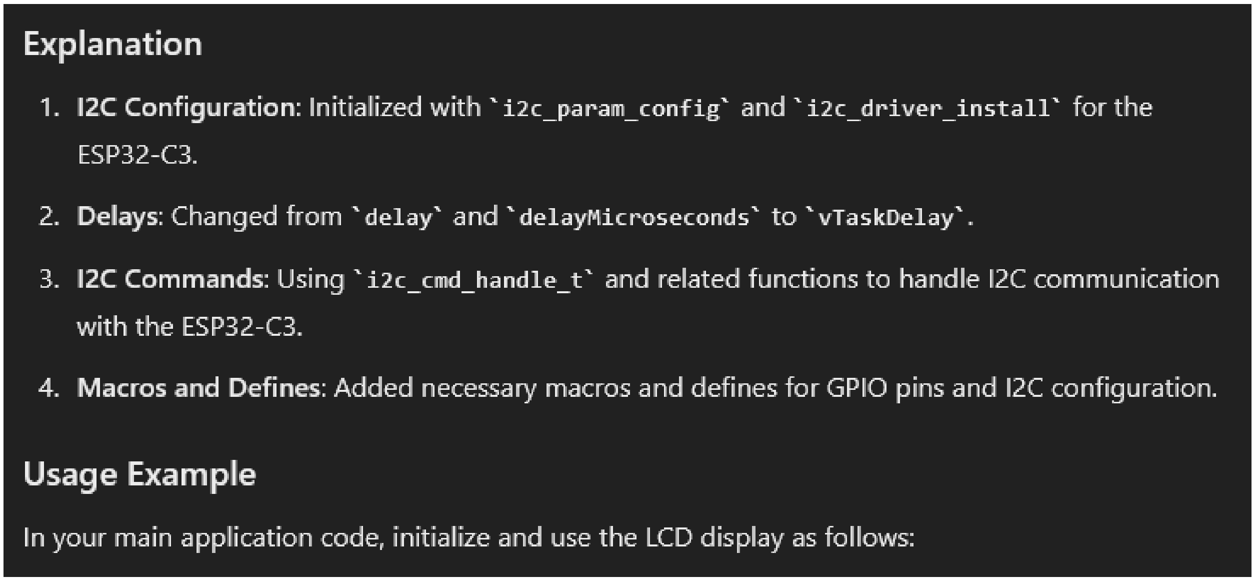
DFRobot_LCD.h

First, create the header file `DFRobot_LCD.h`:



DFRobot_LCD.cpp

Next, create the implementation file `DFRobot_LCD.cpp`:



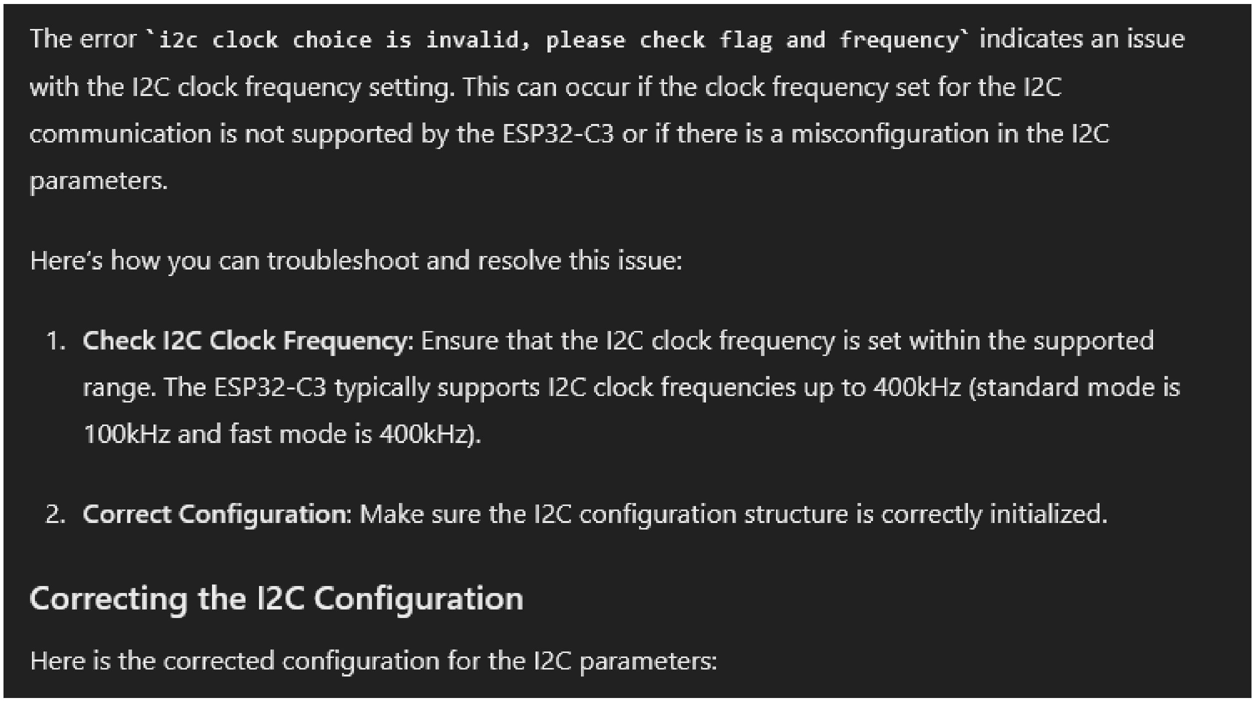
Explanation

1. **I2C Configuration**: Initialized with `i2c_param_config` and `i2c_driver_install` for the ESP32-C3.
2. **Delays**: Changed from `delay` and `delayMicroseconds` to `vTaskDelay`.
3. **I2C Commands**: Using `i2c_cmd_handle_t` and related functions to handle I2C communication with the ESP32-C3.
4. **Macros and Defines**: Added necessary macros and defines for GPIO pins and I2C configuration.

Usage Example

In your main application code, initialize and use the LCD display as follows:

“what could cause this error? E (291) i2c: i2c_param_config(793): i2c clock choice is invalid, please check flag and frequency”



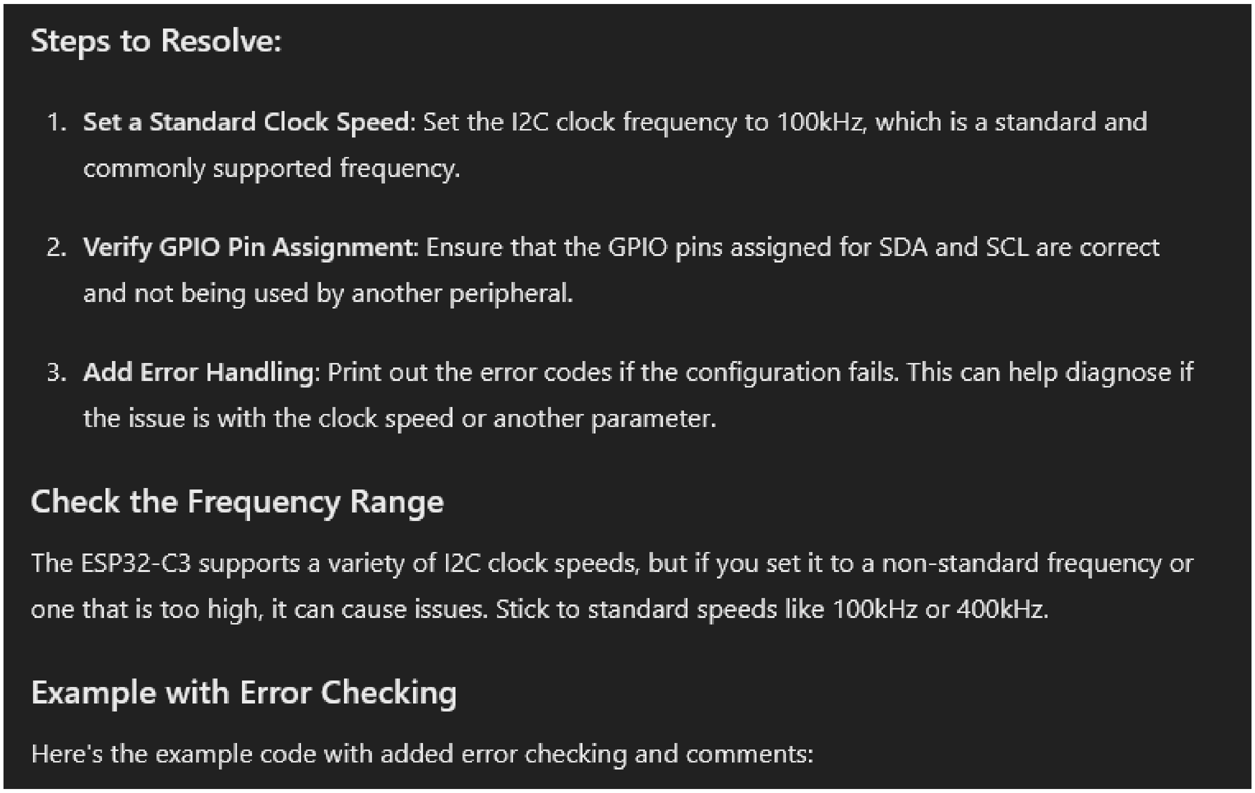
The error `i2c clock choice is invalid, please check flag and frequency` indicates an issue with the I2C clock frequency setting. This can occur if the clock frequency set for the I2C communication is not supported by the ESP32-C3 or if there is a misconfiguration in the I2C parameters.

Here's how you can troubleshoot and resolve this issue:

1. **Check I2C Clock Frequency:** Ensure that the I2C clock frequency is set within the supported range. The ESP32-C3 typically supports I2C clock frequencies up to 400kHz (standard mode is 100kHz and fast mode is 400kHz).
2. **Correct Configuration**: Make sure the I2C configuration structure is correctly initialized.

Correcting the I2C Configuration

Here is the corrected configuration for the I2C parameters:



Steps to Resolve:

1. **Set a Standard Clock Speed**: Set the I2C clock frequency to 100kHz, which is a standard and commonly supported frequency.
2. **Verify GPIO Pin Assignment**: Ensure that the GPIO pins assigned for SDA and SCL are correct and not being used by another peripheral.
3. **Add Error Handling**: Print out the error codes if the configuration fails. This can help diagnose if the issue is with the clock speed or another parameter.

Check the Frequency Range

The ESP32-C3 supports a variety of I2C clock speeds, but if you set it to a non-standard frequency or one that is too high, it can cause issues. Stick to standard speeds like 100kHz or 400kHz.

Example with Error Checking

Here's the example code with added error checking and comments: