FIRST EDITION – Chapter 1

Kevin Thomas

Forward

I remember when I started learning programming to which my first language was 6502 Assembler. It was to program a Commodore 64 and right from the beginning I learned the lowest level development possible.

Literally every piece of the Commodore 64 was understood as it was a simple machine. There was absolutely no abstraction layer of any kind.

Everything we did we had an absolute mastery of however it was a very simple architecture.

Microcontrollers are small systems without an operating system and are also very simple in their design. They are literally everywhere from your toaster to your fridge to your TV and billions of other electronics that you never think about.

Most microcontrollers are developed in the C programming language with has its roots to the 1970’s however dominates the landscape.

Perhaps if Rust ever gains ground in microcontrollers there will be a supplemental book however today in 2023 it is very much a C landscape.

We will take our time and learn the basics of C within an STM32F401CC6 microcontroller.

Below are items you will need for this book.

STM32F401CCU6
https://www.amazon.com/SongHe-STM32F401-Development-STM32F401CCU6-%20Learning/dp/B07XBWGF9M

ST-Link V2 Emulator Downloader Programmer
https://www.amazon.com/HiLetgo-Emulator-Downloader-ProgrammerSTM32F103C8T6/dp/B07SQV6VLZ

DSD TECH HM-11 Bluetooth 4.0 BLE Module
https://www.amazon.com/DSD-TECH-Bluetooth-Compatible-Devices/dp/B07CHNJ1QN

Electronics Soldering Iron Kit
https://www.amazon.com/Electronics-Adjustable-Temperature-ControlledThermostatic/dp/B0B28JQ95M?th=1

Premium Breadboard Jumper Wires
https://www.amazon.com/Keszoox-Premium-Breadboard-Jumper-Raspberry/%20dp/B09F6X3N79

Breadboard Kit
https://www.amazon.com/Breadboards-Solderless-BreadboardDistribution-Connecting/dp/B07DL13RZH

SSD1306 Display
https://www.amazon.com/Hosyond-Display-Self-Luminous-Compatible-Raspberry/dp/B09T6SJBV5

8x8 WS2182 NeoPixel Array
https://www.amazon.com/Tiabiaya-LED-Panel-CJMCU-8X8-Module/dp/B0BLGN88NH

6x6x5mm Momentary Tactile Tact Push Button Switches
https://www.amazon.com/DAOKI-Miniature-Momentary-Tactile-Quality/dp/B01CGMP9GY

NOTE: The item links may NOT be available but the descriptions allow you to shop on any online or physical store of your choosing.

Let’s begin...

Table Of Contents

Chapter 1: hello, world

We begin our journey building the traditional *hello, world* example in Embedded C.

We will then reverse engineer the binary in GDB.

We are going to use PlatformIO and VSCode.

Let’s download the GNU Arm Embedded Toolchain for Windows, MAC or Linux.

https://developer.arm.com/downloads/-/gnu-rm

Let’s download OpenOCD.

https://gnutoolchains.com/arm-eabi/openocd

Let’s download Visual Studio Code which we will use as our integrated development environment.

https://code.visualstudio.com/

Once installed, let’s add the Platform IO IDE extension within VS Code.

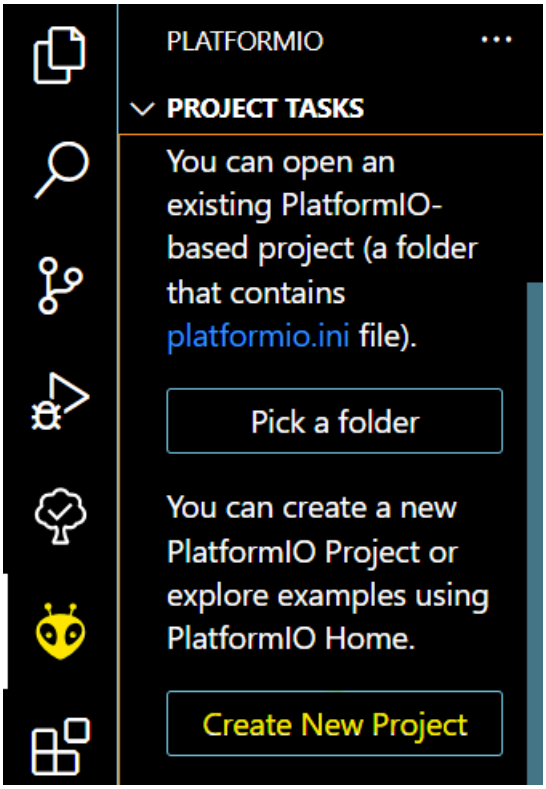
https://marketplace.visualstudio.com/items?itemName=platformio.platformio-ide

Let’s create a new project and get started by following the below steps.

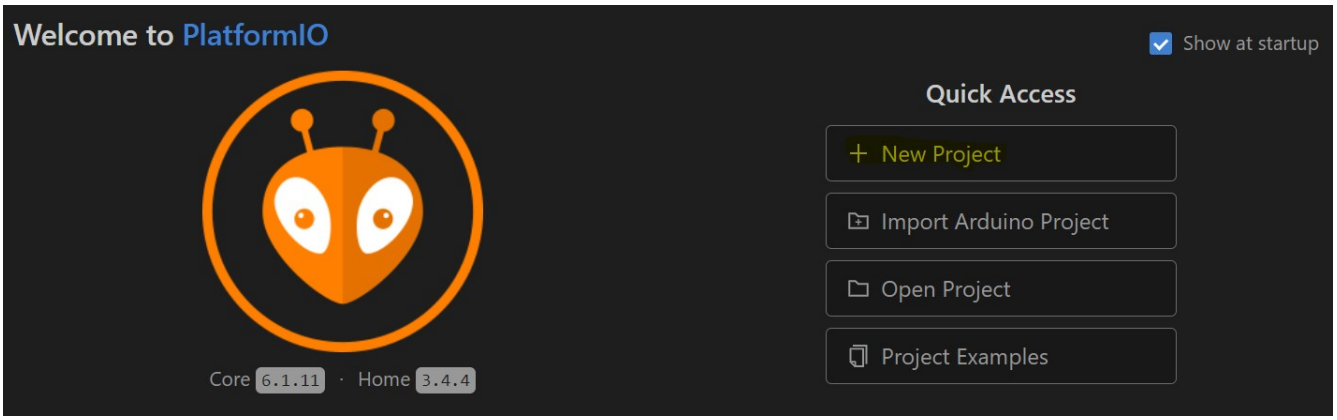
You will see the PlatformIO icon on the left side-bar click on it and it will pop up a menu as below. You will see the items in yellow to help identify what to click on through the process.

Let’s begin!

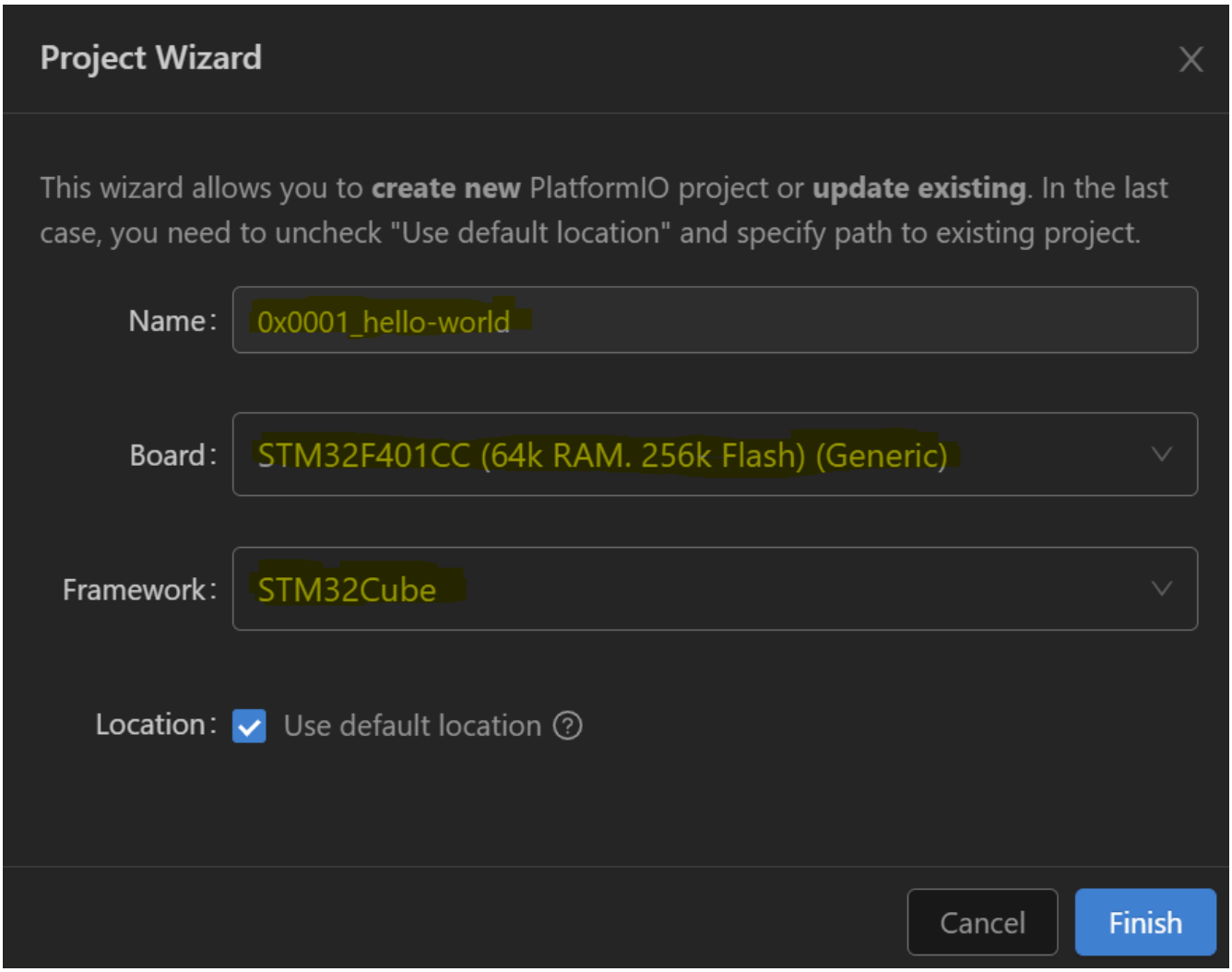
click **Create New Project**



click **New Project**



Name: **0x0001_hello-world**
Board: STM32F401CC
Framework: STM32Cube
click Finish



These are the steps to create a new project however I have created such a folder in the GitHub repo here which has all of the custom libraries needed to work with our example.

If you do not have Git installed, here is a link to install git on Windows, MAC and Linux.

https://git-scm.com/book/en/v2/Getting-Started-Installing-Git

Clone the repo to whatever folder you prefer.

git clone https://github.com/mytechnotalent/Embedded-Hacking.git

Open VS Code and click **File** then **Open Folder …** then click on the **Embedded-Hacking** folder and then select **0x0001_hello-world**.

Give it a few minutes to initialize.

If there are any issues I would point you to the below document as it is related to Windows. We are not using the Arduino framework however look for the heading, **Important steps for Windows users, before installing**, to handle the long paths issues in Git for Windows.

https://arduino-pico.readthedocs.io/en/latest/platformio.html

If you have any additional issues here is the docs for the STM32 Platform IO port.

https://docs.platformio.org/en/stable/platforms/ststm32.html#tutorials

Let’s first examine the toolbar on the bottom of VS Code.

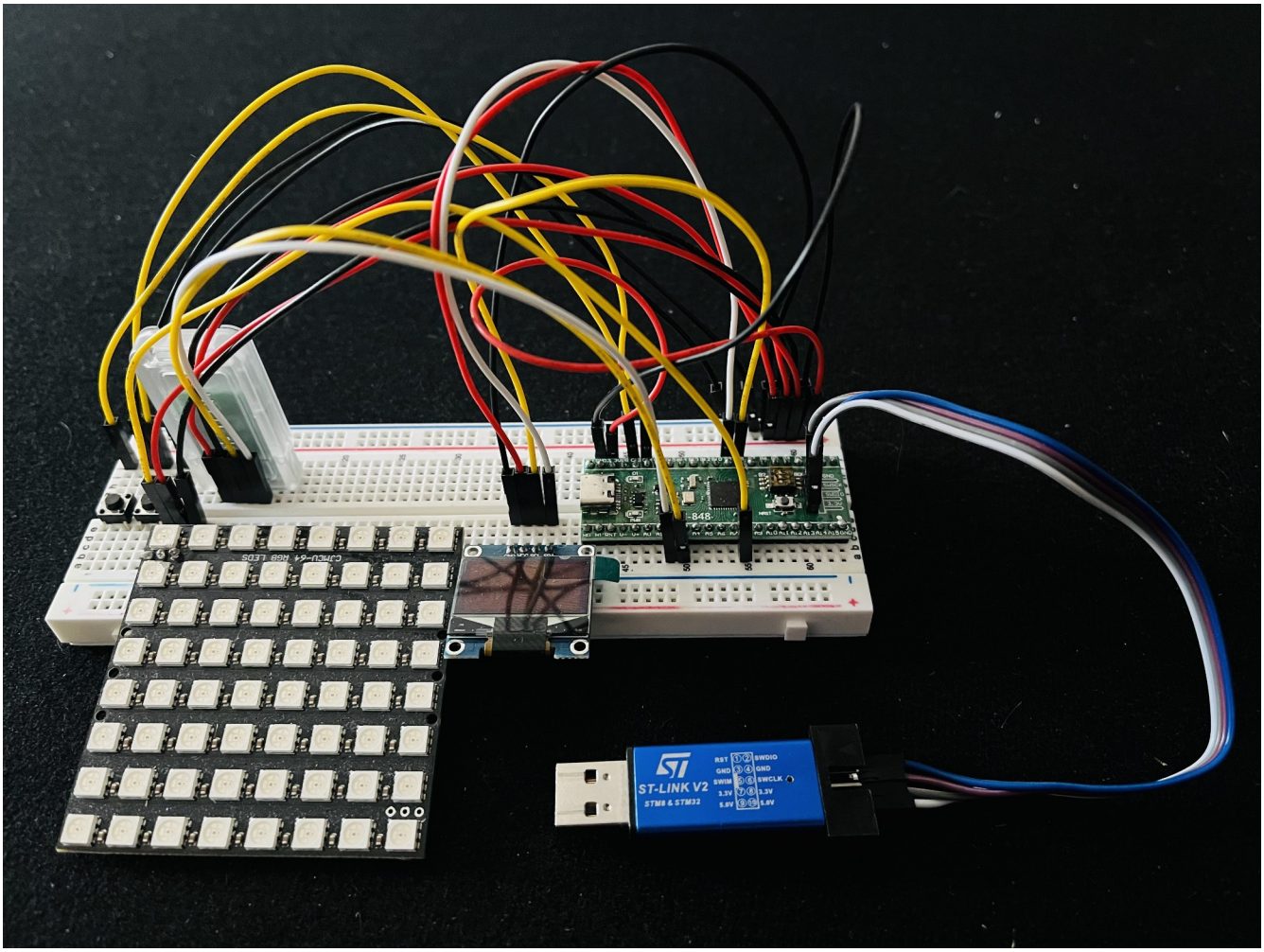
We can compile and upload this code in one button click by clicking on the → button above.

If everything is successful you will see SUCCESS in green within the terminal.

```
[SUCCESS]
```

Before we go any further we need to first soldier up our STM32F401CCU and attach it into the breadboard. We then need to wire up our various external devices.

Let’s start with an overall picture of our setup and I will walk through every connection one-by-one.



NOTE: The item links may NOT be available but the descriptions allow you to shop on any online or physical store of your choosing.

STM32F401CCU6
https://www.amazon.com/SongHe-STM32F401-Development-STM32F401CCU6-%20Learning/dp/B07XBWGF9M

ST-Link V2 Emulator Downloader Programmer
https://www.amazon.com/HiLetgo-Emulator-Downloader-ProgrammerSTM32F103C8T6/dp/B07SQV6VLZ

After soldering up the STM32F401CCU6, place it into the breadboard with the orientation showing above where the USB-C is to the left and not near the right edge. The debug pins are the 4 pins on the right and side of the board.

```
connect the MCU 3.3V DEBUG PIN to the ST-LINK 3.3V PIN
connect the MCU GND DEBUG PIN to the ST-LINK GND PIN
connect the MCU SWDIO DEBUG PIN to the ST-LINK SWDIO PIN
connect the MCU SWSCK DEBUG PIN to the ST-LINk SWSCK PIN
connect the MCU 3.3V to the 3.3V rail of the breadboard PIN
connect the MCU GND to the GND rail of the breadboard PIN
```

DSD TECH HM-11 Bluetooth 4.0 BLE Module
https://www.amazon.com/DSD-TECH-Bluetooth-Compatible-Devices/dp/B07CHNJ1QN

```
connect the MCU 3.3V RAIL to the DSD TECH VCC PIN
connect the MCU GND RAIL to the DSD TECH GND PIN
connect the MCU A3 PIN to the DSD TECH TXD PIN
connect the MCU A2 PIN to the DSD TECH RXD PIN
```

SSD1306 Display
https://www.amazon.com/Hosyond-Display-Self-Luminous-Compatible-Raspberry/dp/B09T6SJBV5

```
connect the MCU 3.3V RAIL to the SSD1306 VCC PIN
connect the MCU GND RAIL to the SSD1306 GND PIN
connect the MCU B8 PIN to the SSD1306 SCL PIN
connect the MCU B9 PIN to the SSD1306 SDA PIN
```

8x8 WS2182 NeoPixel Array
https://www.amazon.com/Tiabiaya-LED-Panel-CJMCU-8X8-Module/dp/B0BLGN88NH

```
connect the MCU 3.3V RAIL to the WS2812 +5V PIN
connect the MCU GND RAIL to the WS2812 GND PIN
connect the MCU A8 PIN to the WS2812 DIN PIN
```

6x6x5mm Momentary Tactile Tact Push Button Switches
https://www.amazon.com/DAOKI-Miniature-Momentary-Tactile-Quality/dp/B01CGMP9GY

```
connect the MCU GND RAIL to the LEFT BUTTON RIGHT PIN
connect the MCU C13 PIN to the LEFT BUTTON LEFT PIN
connect the MCU GND RAIL to the RIGHT BUTTON RIGHT PIN
connect the MCU C14 PIN to the RIGHT BUTTON LEFT PIN
```

Our last step is to download Bluetooth software to connect with our MCU (micro controller or STM32F401CCU).

Android
Serial Bluetooth Terminal
https://play.google.com/store/apps/details?id=de.kai_morich.serial_bluetooth_terminal&hl=en_US&gl=US

iPhone
BLE Serial tiny
https://apps.apple.com/us/app/ble-serial-tiny/id1607862132

Either option when connecting you will connect to the DSD TECH device.

Now let's review our **main.c** file as this is located in the **src** folder.

```
#include <stdio.h>

#include "usart.h"

int main(void)
{
    usart2_init();

    while (1)
    {
        printf("hello, world\r\n");
    }
}
```

When we connect to our MCU through either app explained above we will see, *hello, world*, displayed over and over. Congrats you just setup and built your first embedded C application!

In our next lesson we will debug hello, world using the ARM embedded GDB with OpenOCD to which we will actually connect LIVE to our running MCU!