WiSH-2025

Let's Rock with MSPM0

Devansh Tanna
d-tanna@ti.com

Texas Instruments

Table of Contents

1.
Background

In this experiment, we're going to explore TI's latest low-power microcontroller family MSPM0, and more specifically `TI-MSPM0G3507` which is Arm® Cortex®-M0+ based microcontroller.

We'll not go very deep into how this microcontroller works in low-level but it would be very interesting if you can explore on your own, TI provides lot of documents including TRM(Technical Reference Manual) and many application notes which go through different cases in which this microcontroller can be useful

2.
Objective

In our daily routine, we have used IR remotes for many applications, like controlling TV, Fan, AC, and many more appliances. In this lab, we'll try to build a similar IR transmitter using MSPM0. MSPM0 is very cost-effective and powerful solution to this application compared to few other cheaper microcontrollers which lacks in proper documentation.

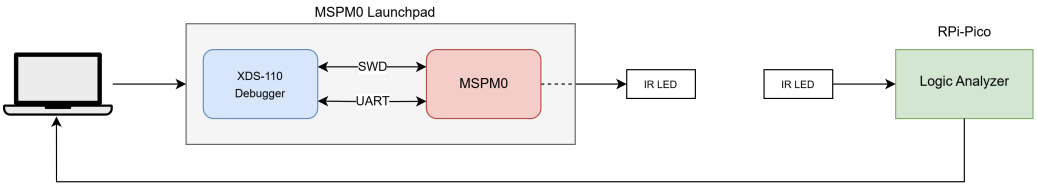
As mentioned, we'll be using LaunchPad for MSPM0G3507(LP-MSPM0G3507) which has already all the required component built into single board, upper part of board is used as debugger(XDS-110ET) which will be used to program the microcontroller, launchpad has male and female header pins and few jumper caps

Before building any prototype/product we should do system level design, directly jumping into connecting wire and writing code generally doesn't work if you don't understand system thoroughly. You can look at any TI provided application notes(See References) you will get to know basics of system design using this microcontroller

- First, we'll look what is our requirement and target prototype
- Then, we'll look how this can work at system level and what are the target specifications
- After that, we'll see if we can build this system using give components
- Note, generally in system design, components are finalized until last step, as per the given specification components are selected based on cost, form-factor and other attributes. But for our design, we're trying to showcase the MSPM0's capability so we don't need to go through the last step. We'll just see if we can somehow manage to build our target prototype using given components

3.
System Design

- We will build programmable packet transmitter using IR Led which is controlled by GPIO pin from MSPM0 Microcontroller.
- To test our system, we'll use Logic Analyzer, which can sample any digital signal at very high sample-rate, so we can debug our program.
- For this experiment, we'll send one 8-bits packet to IR receiver. We'll get this 8-bits from UART peripheral, because UART peripheral is already interfaced with our PC through the debugger.
- So, our final system will consist of PC interfaced with MSPM0 which drives IR LED, which transmits 8-bits data, which can be received by IR Receiver connected to Logic Analyzer.
- There are many things which might be new to you at this point, but don't worry, we'll go through each steps in following sections.



3.1 Appendix-A: Installing PulseView

3.1.1 Installation:

1. Download `PulseView` installer from this link: https://sigrok.org/download/binary/pulseview/pulseview-NIGHTLY-x86_64-release-installer.exe
2. Download the `sigrok-cli` installer from this link, in our case we might not use it but it's very useful tool: https://sigrok.org/download/binary/sigrok-cli/sigrok-cli-NIGHTLY-x86_64-release-installer.exe
3. Install both the tools using their installers. We'll use Raspberry Pi Pico as logic analyzer, so download the firmware from below link and keep it your machine, we'll see how to flash it in Raspberry Pi Pico(You are encouraged to explore Raspberry Pi Pico, it's very useful microcontroller like MSPM0 with faster clock but with less analog peripherals) : https://raw.githubusercontent.com/pico-coder/sigrok-pico/main/pico_sdk_sigrok/build/pico_sdk_sigrok.uf2

3.1.2 Workaround

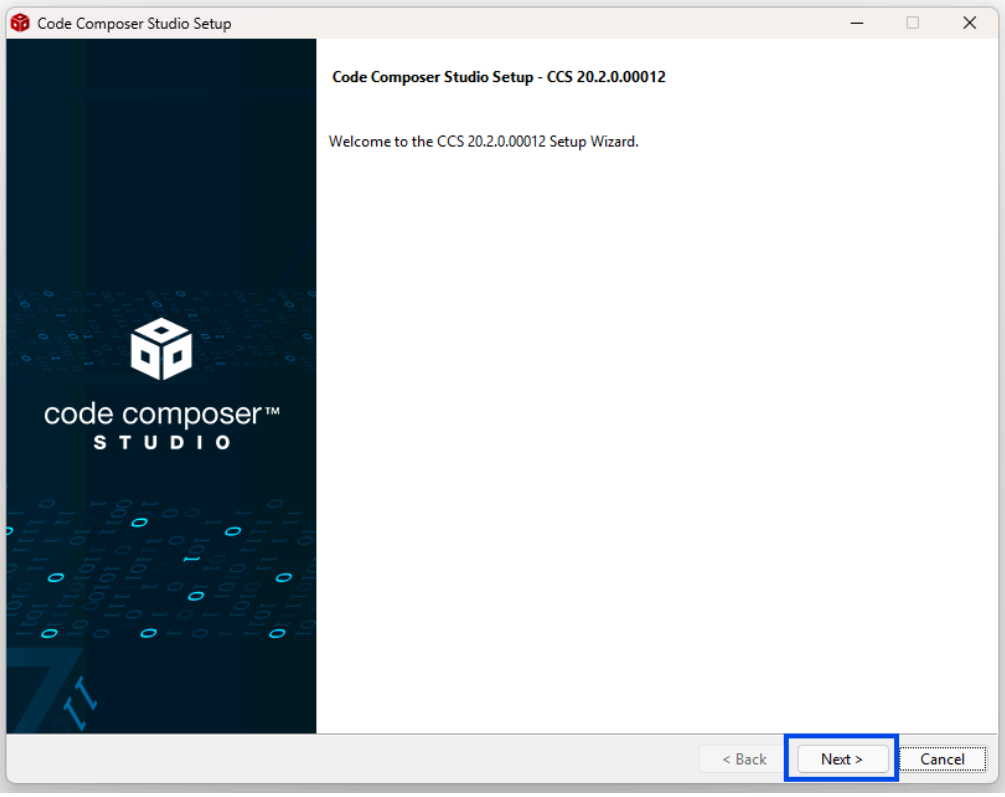
```
mode COM7 BAUD=115200 data=8 Parity=n DTR=on
```

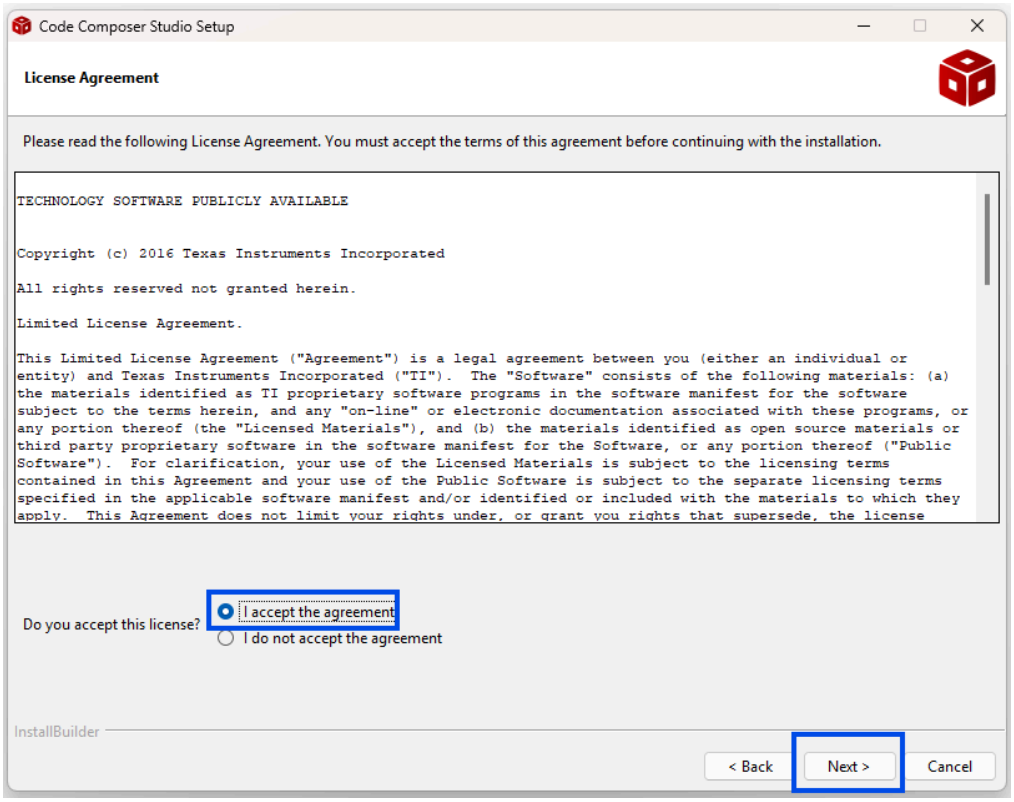
3.2 Appendix-B: Installing CC Studio

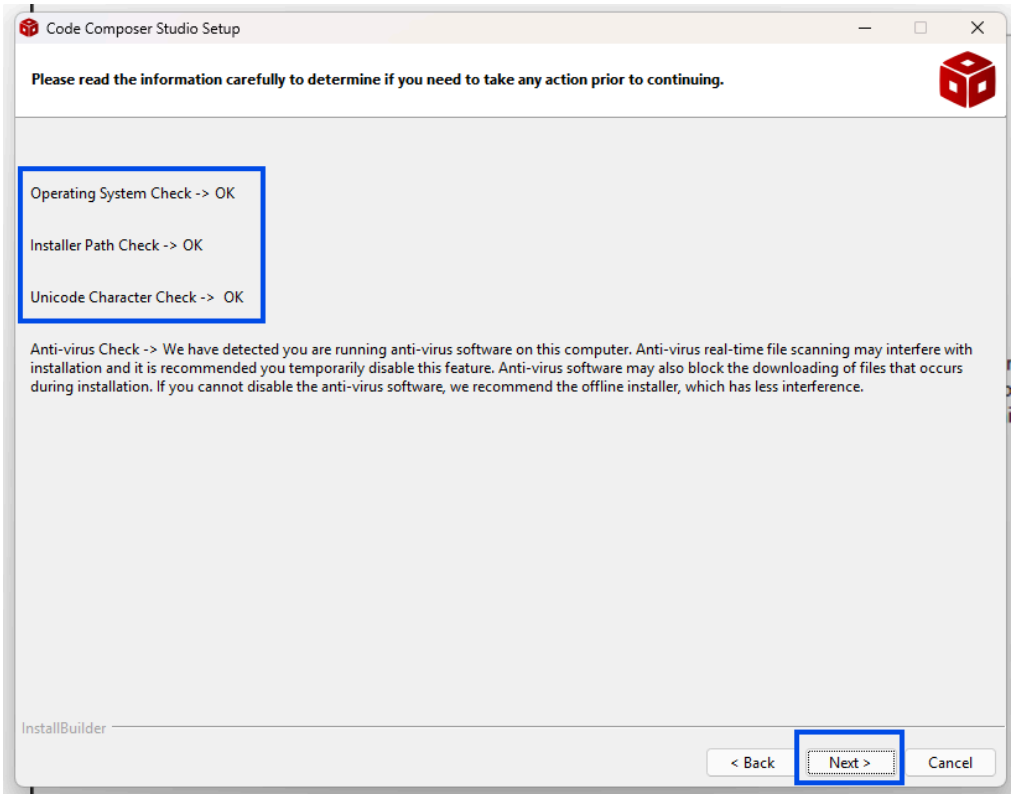
- Code Composer Studio is IDE(Integration Developement Environment) which provides tool-chains for TI-Devices.
- We'll be using this for developement on LP-MSPM0G3507 board

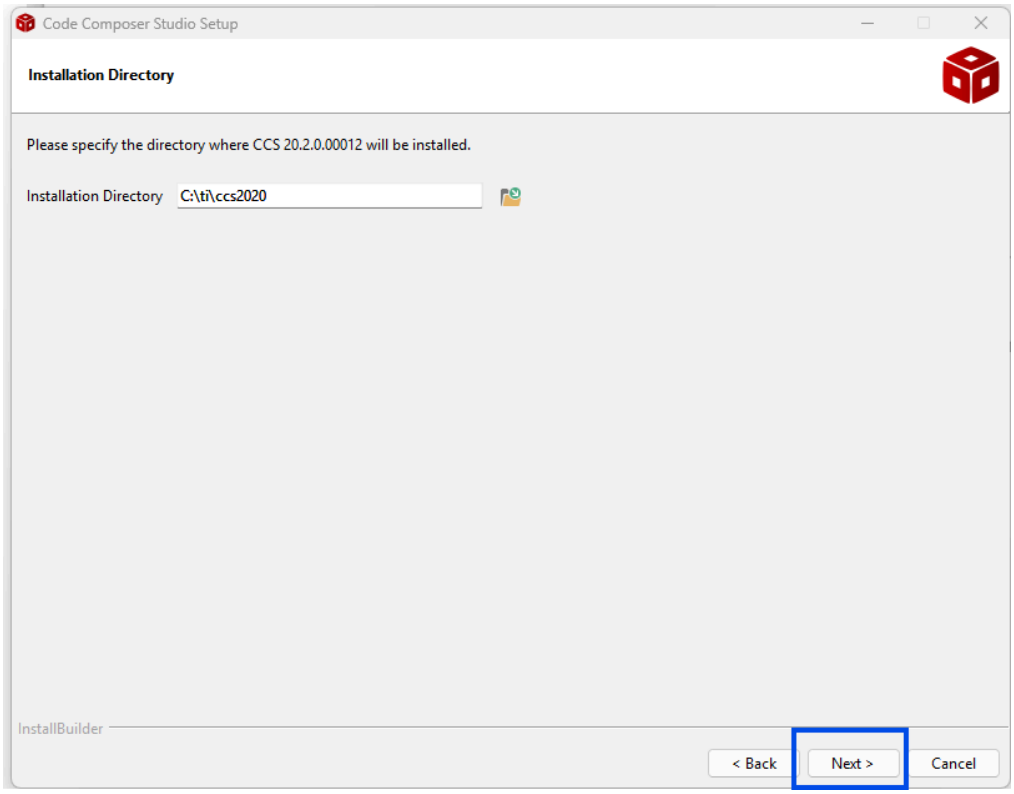
Follow these steps for installation:

1. Download CCS Installer (offline installer) for latest version from https://www.ti.com/tool/CCSTUDIO
2. Start the installer(Ignore Version mentioned in image)

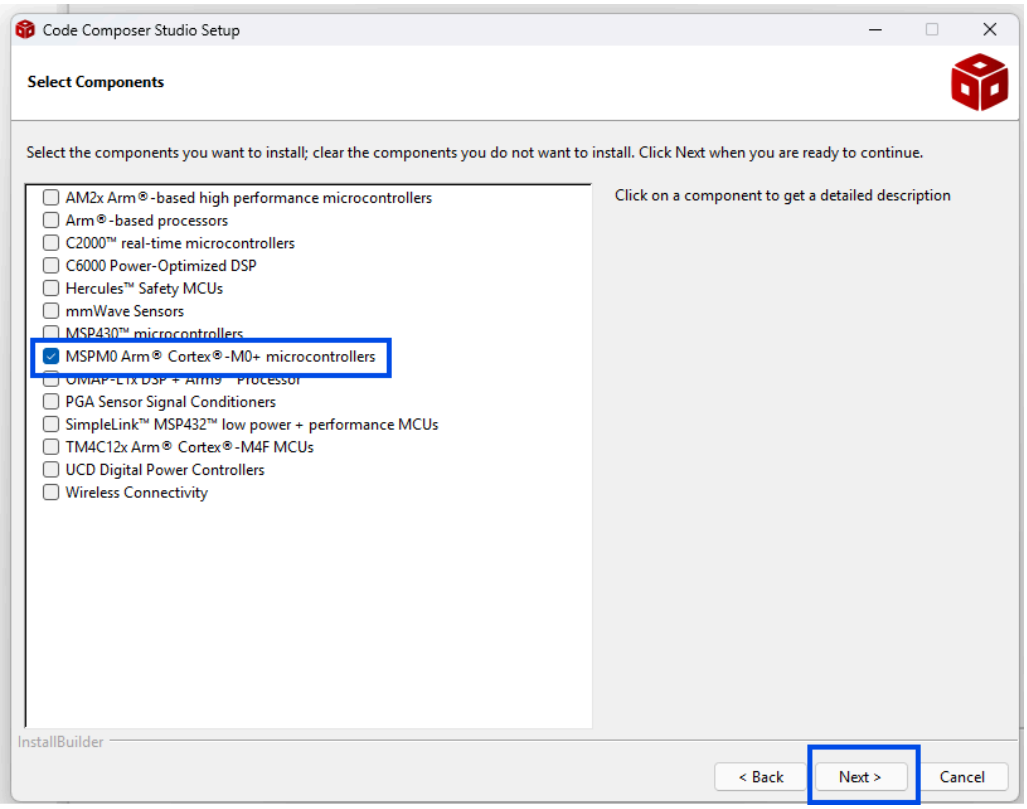



Code Composer Studio Setup
License Agreement
Please read the following License Agreement. You must accept the terms of this agreement before continuing with the installation.
TECHNOLOGY SOFTWARE PUBLICLY AVAILABLE
Copyright (c) 2016 Texas Instruments Incorporated
All rights reserved not granted herein.
Limited License Agreement.
This Limited License Agreement ("Agreement") is a legal agreement between you (either an individual or entity) and Texas Instruments Incorporated ("TI"). The "Software" consists of the following materials: (a) the materials identified as TI proprietary software programs in the software manifest for the software subject to the terms herein, and any "on-line" or electronic documentation associated with these programs, or any portion thereof (the "Licensed Materials"), and (b) the materials identified as open source materials or third party proprietary software in the software manifest for the Software, or any portion thereof ("Public Software"). For clarification, your use of the Licensed Materials is subject to the licensing terms contained in this Agreement and your use of the Public Software is subject to the separate licensing terms specified in the applicable software manifest and/or identified or included with the materials to which they apply. This Agreement does not limit your rights under, or grant you rights that supersede, the license
Do you accept this license?
I accept the agreement
I do not accept the agreement
InstallBuilder
< Back
Next >
Cancel

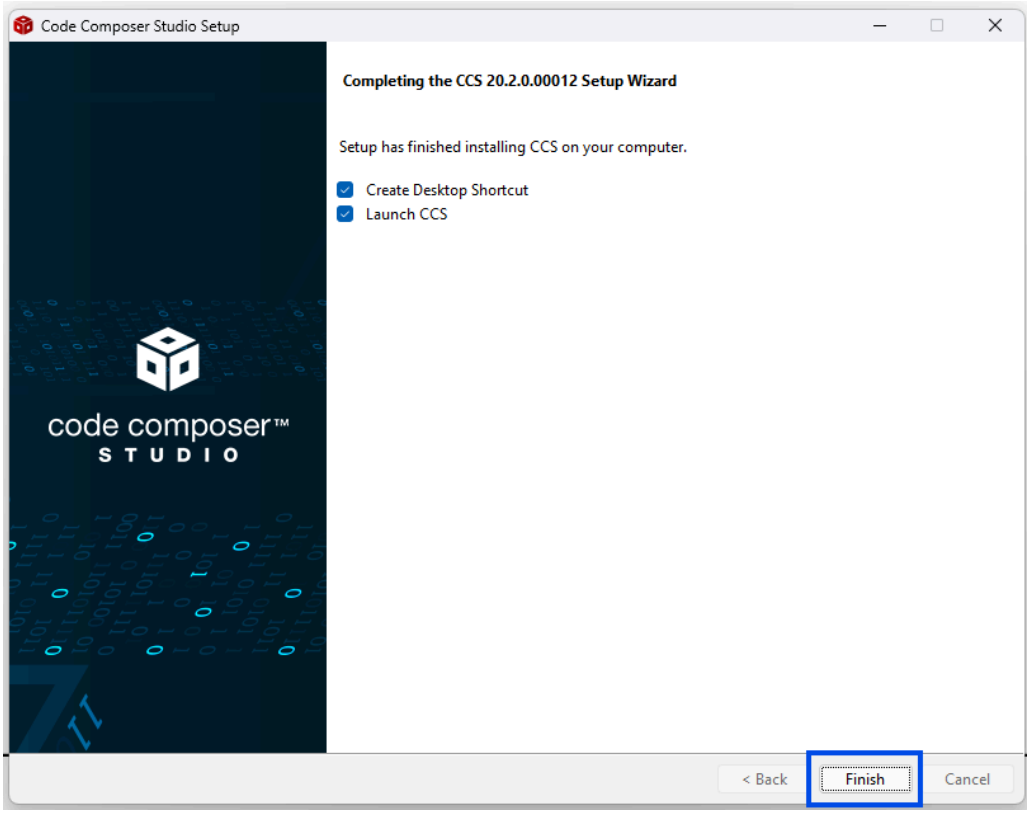

Code Composer Studio Setup
Please read the information carefully to determine if you need to take any action prior to continuing.
Operating System Check -> OK
Installer Path Check -> OK
Unicode Character Check -> OK
Anti-virus Check -> We have detected you are running anti-virus software on this computer. Anti-virus real-time file scanning may interfere with installation and it is recommended you temporarily disable this feature. Anti-virus software may also block the downloading of files that occurs during installation. If you cannot disable the anti-virus software, we recommend the offline installer, which has less interference.
InstallBuilder
< Back
Next >
Cancel


Code Composer Studio Setup
Installation Directory
Please specify the directory where CCS 20.2.0.00012 will be installed.
Installation Directory
C:\ti\ccs2020
InstallBuilder
< Back
Next >
Cancel

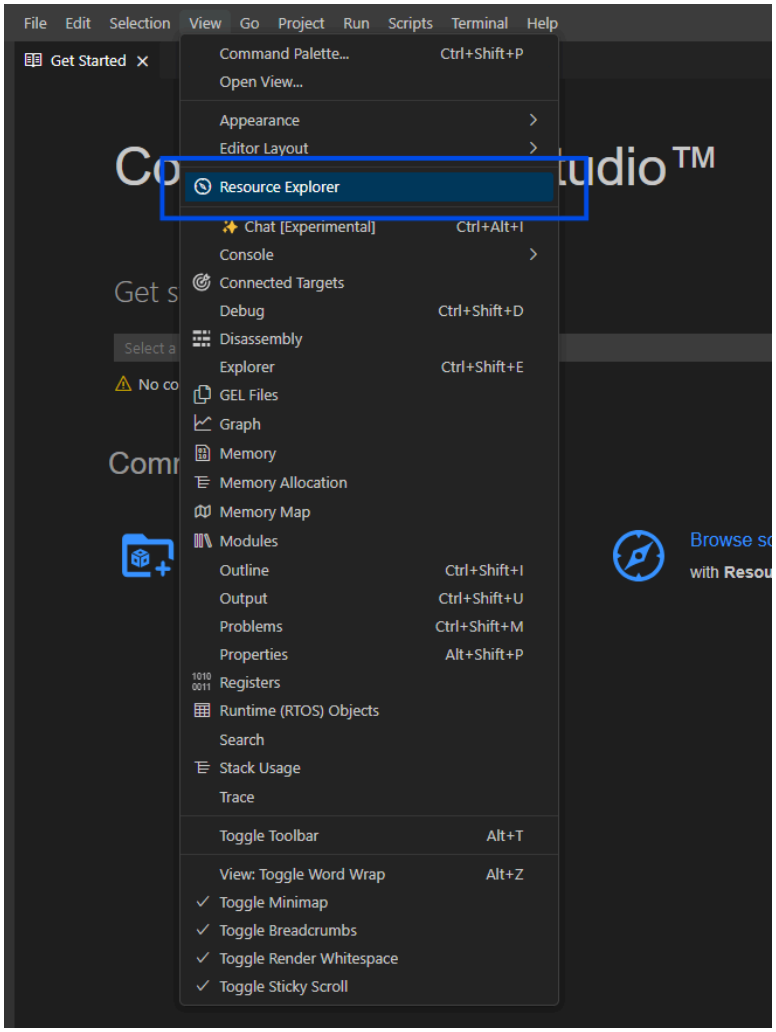
1. Select these devices on



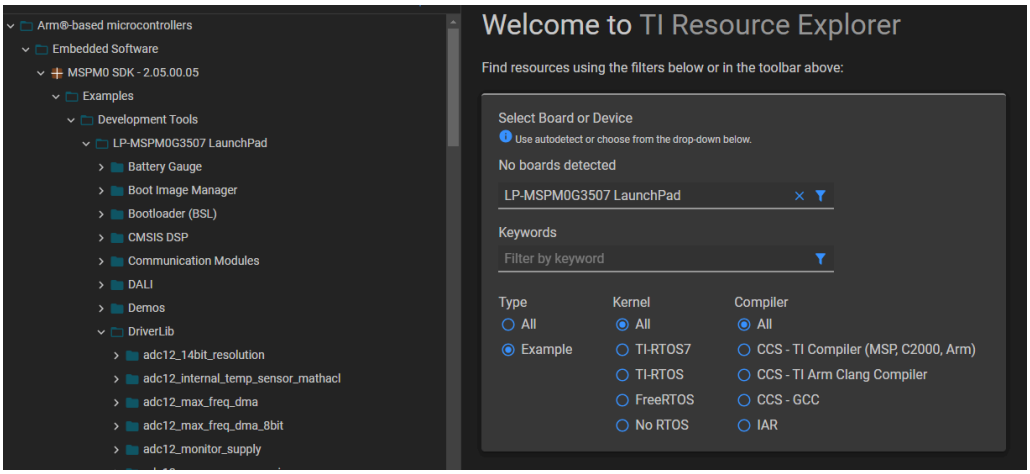
2. Wait for installation to complete



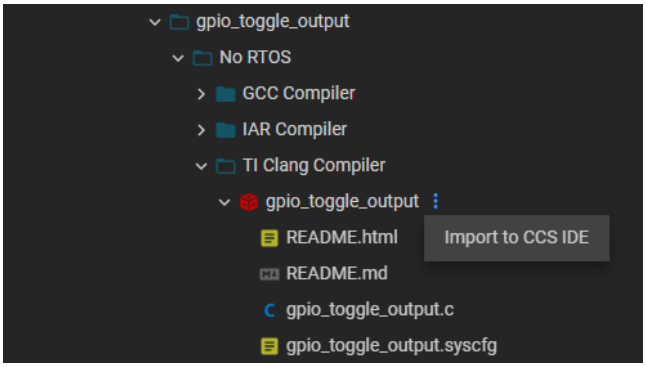
3. After completion, open the CC Studio and go to `View->Resource Explorer`



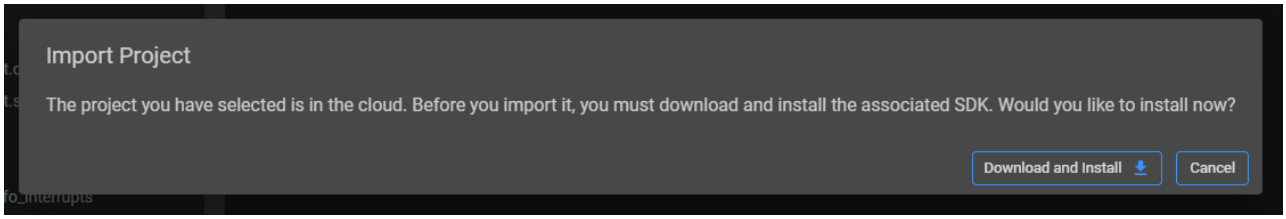
4. Enter the board name(`LP-MSPM0G3507`) and open `Example` in left-pane



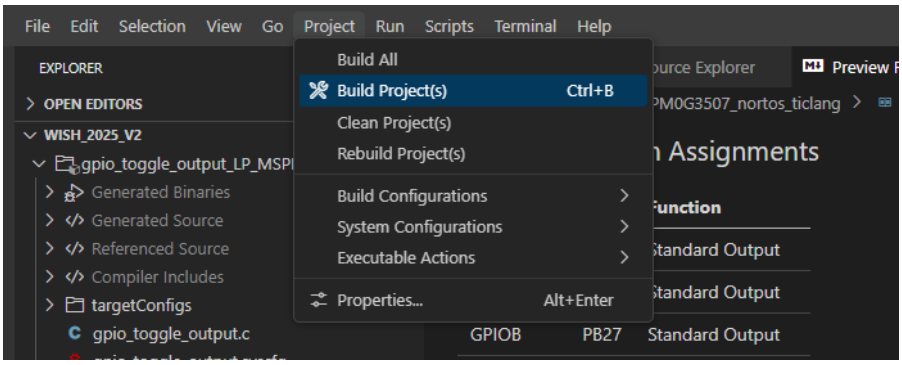
5. Go to `gpio_toggle_output` inside `DriverLib` folder. Click on `Import to CCS IDE`



6. You will get one prompt to install SDK, select `Download and Install` in that box



7. Wait for download to complete, and go to `Project->Build Project(s)`

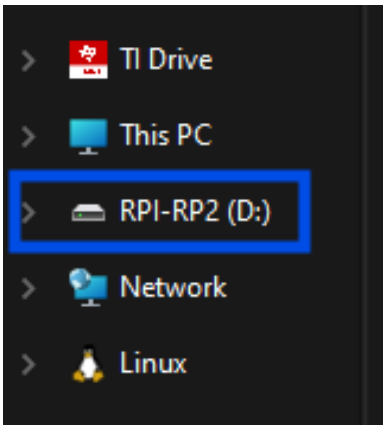


8. Now, make sure that you are able to build the project without any errors.

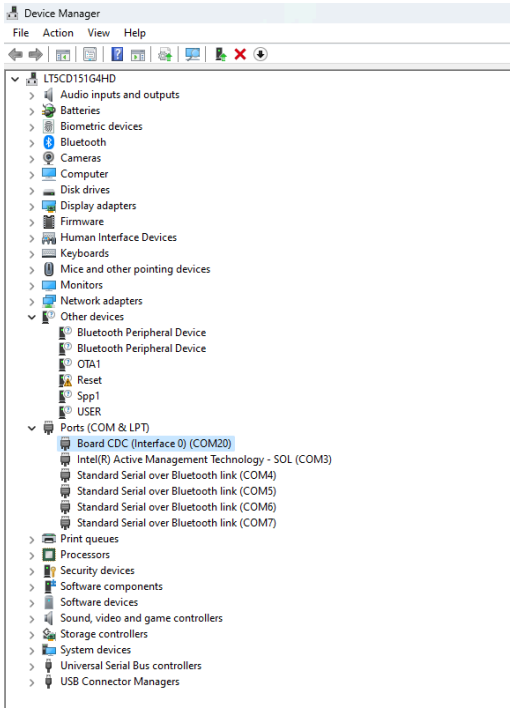
3.3 Appendix-C: Getting Started with Logic Analyzer

3.3.1 Set-Up

1. Download the firmware for the Raspberry Pi Pico from this link https://github.com/pico-coder/sigrok-pico/raw/main/pico_sdk_sigrok/build/pico_sdk_sigrok.uf2
2. Now, connect the Raspberry Pi Pico while pressing `BOOTSEL` button as shown in this guide https://projects.raspberrypi.org/en/projects/getting-started-with-the-pico/3
3. You will see new drive created in your file-manager like this:



4. Copy the firmware which we've downloaded in this folder
5. After this, you won't see this folder on your File System that means firmware is uploaded correctly.
6. Now disconnect and connect the Raspberry Pi Pico
7. Open Device Manager in Control Panel
8. Identify the `COM` Port on which Raspberry Pi Pico is connected, in my case it's `COM20`



9. If you are not able to identify the COM Port, you can disconnect and connect Raspberry Pi Pico and see which port appeared when you connected the board again

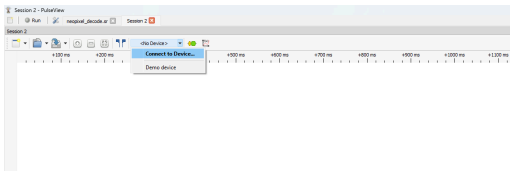
10. *Only for Windows users*: Now open the command prompt in your system, enter the following command, replace `<COM_PORT>` with your COM Port

    ```
    mode <COM_PORT> BAUD=115200 data=8 Parity=n DTR=on
    ```

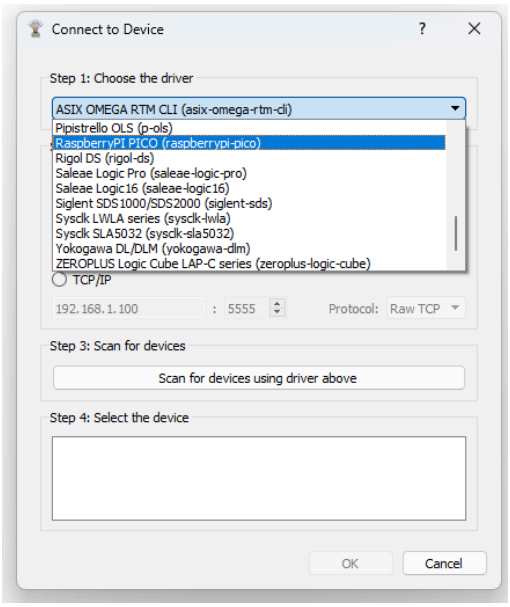
    For my case,

    ```
    mode COM20 BAUD=115200 data=8 Parity=n DTR=on
    ```

11. Now, open the PulseView software
12. You should see menu shown in below screenshot:

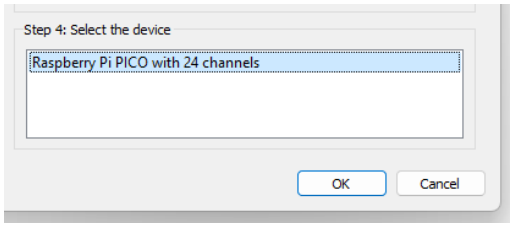


13. Click on the `Connect to Device`
14. In the `Step-1 : Choose the driver`, find the `Raspberry Pi Pico` in the menu and click on that:



15. In the `Step-2 : Choose the interface`, select `Serial Port`
16. In the same step, select your COM Port which you found in earlier step and also enter the baudrate as `921600`

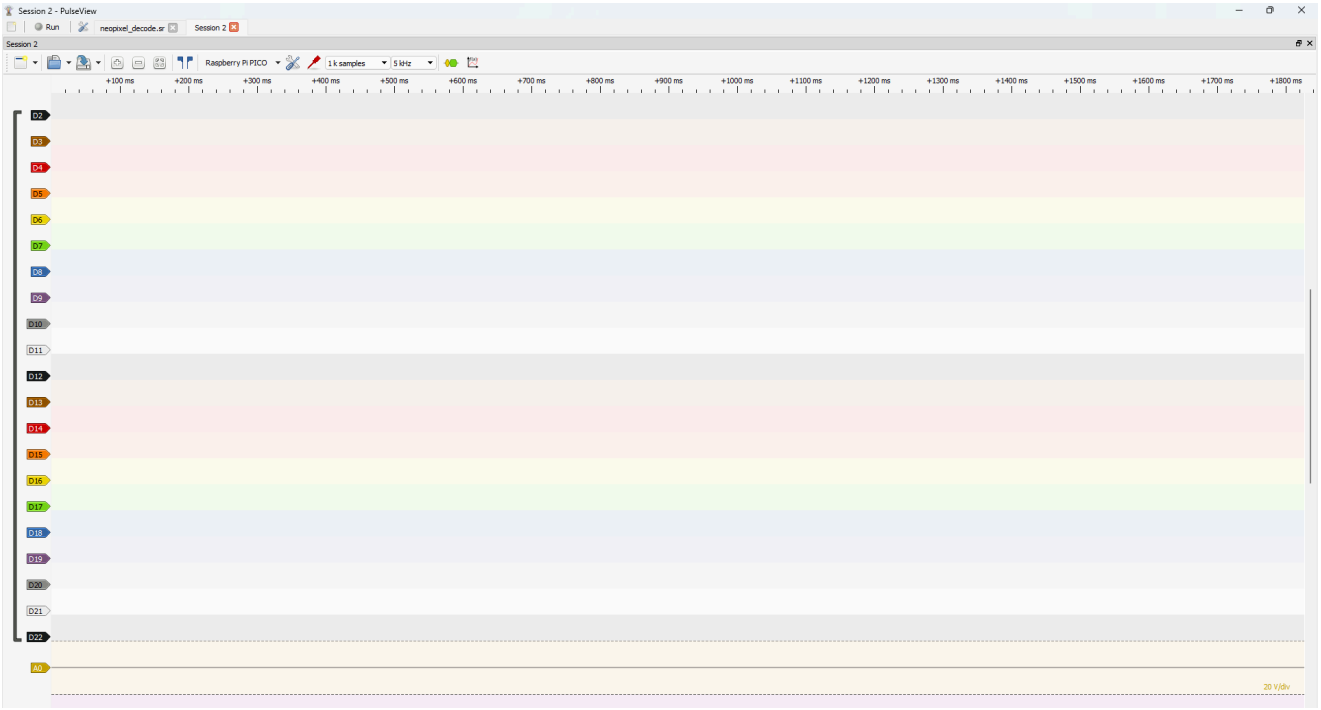
17. Now in the `Step-3: Scan for devices`, press the button named `Scan for the devices using driver above`
18. You should this option:



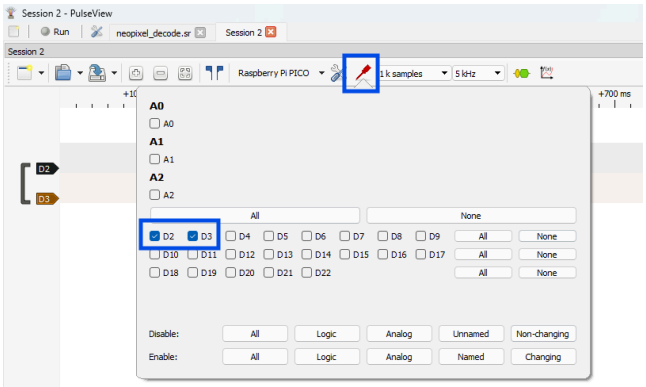
19. After this press `OK`, and you are done this initial setup

3.3.2 User Guide

After above setup you will be having your window like this:

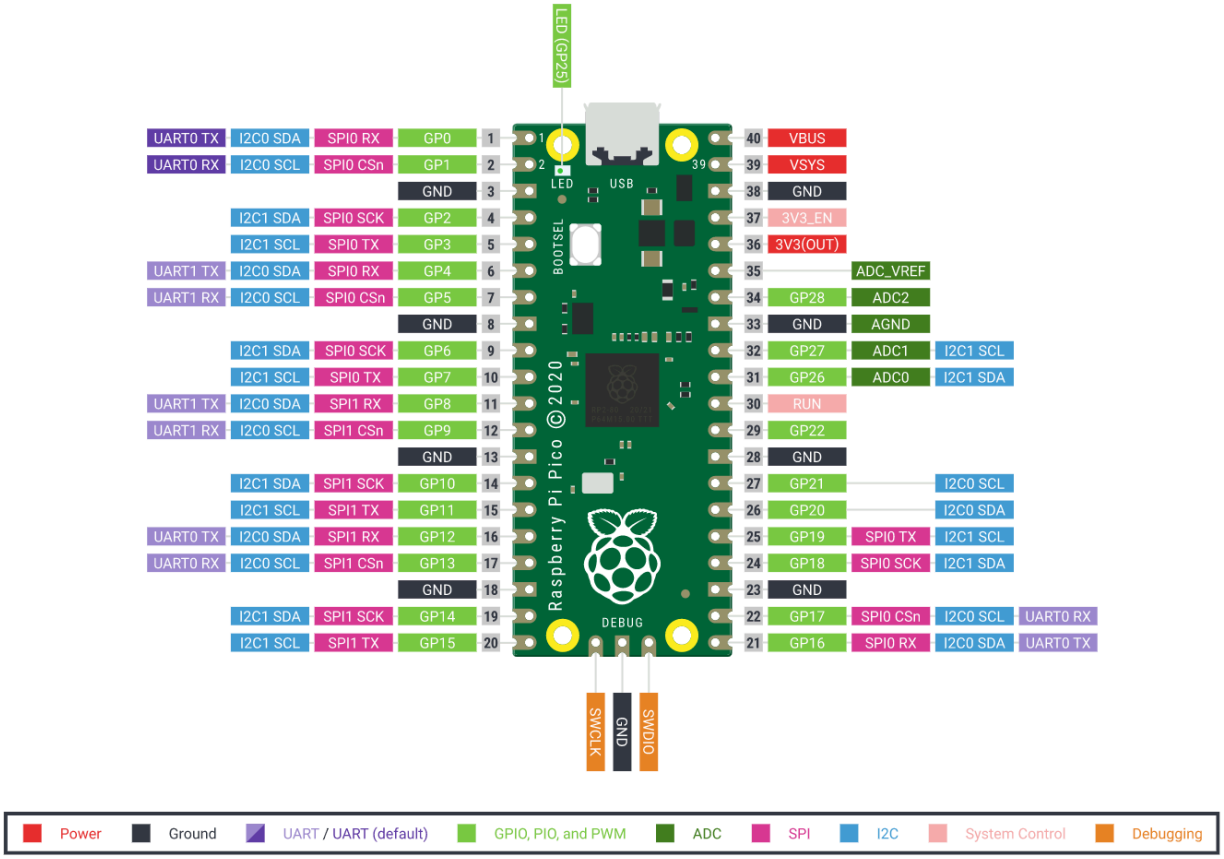


1. We only need 2-channels so you can disable other channels using this menu



2. You can change the sample-rate and number of channels using this menu:

3. You can start the capture using `Run` button in top-bar
4. `D2`, `D3` ... pins are `GP2`, `GP3` pins shown in this pin-out diagram
5. So connect your probe pins `GP2` and `GP3` to external hardware pins which you want to analyze and press the `Run` button


LED (GP25)
UART0 TX I2C0 SDA SPI0 RX GP0 1
UART0 RX I2C0 SCL SPI0 CSn GP1 2
GND 3
I2C1 SDA SPI0 SCK GP2 4
I2C1 SCL SPI0 TX GP3 5
UART1 TX I2C0 SDA SPI0 RX GP4 6
UART1 RX I2C0 SCL SPI0 CSn GP5 7
GND 8
I2C1 SDA SPI0 SCK GP6 9
I2C1 SCL SPI0 TX GP7 10
UART1 TX I2C0 SDA SPI1 RX GP8 11
UART1 RX I2C0 SCL SPI1 CSn GP9 12
GND 13
I2C1 SDA SPI1 SCK GP10 14
I2C1 SCL SPI1 TX GP11 15
UART0 TX I2C0 SDA SPI1 RX GP12 16
UART0 RX I2C0 SCL SPI1 CSn GP13 17
GND 18
I2C1 SDA SPI1 SCK GP14 19
I2C1 SCL SPI1 TX GP15 20
40 VBUS
39 VSYS
38 GND
37 3V3_EN
36 3V3(OUT)
35 ADC_VREF
34 GP28 ADC2
33 GND AGND
32 GP27 ADC1 I2C1 SCL
31 GP26 ADC0 I2C1 SDA
30 RUN
29 GP22
28 GND
27 GP21 I2C0 SCL
26 GP20 I2C0 SDA
25 GP19 SPI0 TX I2C1 SCL
24 GP18 SPI0 SCK I2C1 SDA
23 GND
22 GP17 SPI0 CSn I2C0 SCL UART0 RX
21 GP16 SPI0 RX I2C0 SDA UART0 TX
LED
USB
BOOTSEL
DEBUG
Raspberry Pi Pico © 2020
SWCLK
GND
SWDIO
Power
Ground
UART / UART (default)
GPIO, PIO, and PWM
ADC
SPI
I2C
System Control
Debugging

3.4 Reference

1. Universal Asynchronous Receiver Transmitter (UART) - TI C29x Academy
2. MSPM0G3507 Product Page - https://www.ti.com/product/MSPM0G3507
3. MSPM0G3507 Datasheet - https://www.ti.com/lit/gpn/mspm0g3507
4. MSPM0G3507 Technical Reference Manual - https://www.ti.com/lit/pdf/slau846
5. Video Series on MSPM0 Peripherals - https://www.ti.com/video/series/mspm0-peripherals.html
6. MSPM0G3x0x DriverLib GPIO Documentation - MSPM0G1X0X_G3X0X Driver Library
7. GPIO Example from SDK - `gpio_toggle_output`
8. MSPM0 Open Drain 5V Design - 5-V Interface
9. LP-MSPM0G3507 Product Page - https://www.ti.com/tool/LP-MSPM0G3507
10. LP-MSPM0G3507 EVM Userguide - https://www.ti.com/lit/pdf/slau873