SILICON LABS

works with
BY SILICON LABS

MAT-301

Designing for Matter Over Wi-Fi

Alfredo Pérez Grovas | August 2023

Agenda

- Introduction
- Hardware and Software Prerequisites
- Building and Running a Matter over Wi-Fi Demo
- Setting Up Google Ecosystem Devices for Matter over Wi-Fi demo
- Setting Up Raspberry Pi for Matter over Wi-Fi demo
- Commissioning and Control of Demo from Google Pixel
- Commissioning and Control of Demo from Raspberry Pi
- Using Multi-Admin Capabilities
- Video of Demo Execution
- Silicon Labs Matter over Wi-Fi Portfolio

works with
BY SILICON LABS

Introduction

Matter Overview

What is Matter?

- Matter is a Unified Approach to IoT Device Development.
- Matter is an open-source connectivity standard for smart home and Internet of things devices, which aims to improve their compatibility and security.
- It is not an entirely *new* protocol stack. What it provides is a common application layer that can commonly exist and interoperate over existing protocol stacks such as Wi-Fi and Thread.
- Matter drives the convergence between the major IoT ecosystems to create one easy, reliable, and secure wireless protocol to connect all IoT devices and networks
- Matter works over Wi-Fi, Ethernet, and Thread. Matter controllers take care of Wi-Fi and ethernet devices, but if you have Thread devices, you need a Thread border router to talk to the Matter controller

Matter Overview (Continued)

Consumers

- Extremely hard to mix and match the product they want with their preferred ecosystem
- Impossible to change once selected

Developers

- Developers are forced to pick what ecosystem integrations they support and often need to ship multiple SKUs for all connectivity standards
- Need to learn different IoT technologies and ecosystems

Retailers

- Too difficult to provide expert advice to answer consumer questions
- High return rates due to interoperability issues

WORKS WITH

amazon alexa

ring

xfinity home

SmartThings

Google Assistant

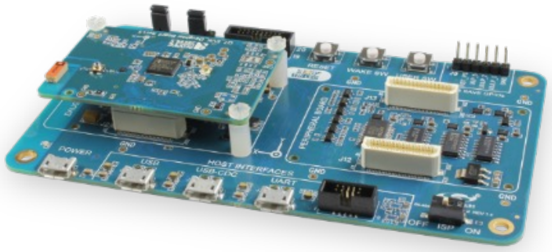
Friends of hue

personal wireless lighting

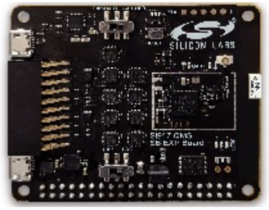
Getting Started with Silicon Labs Matter over Wi-Fi

- **WF200/WFM200S**
  - Raspberry Pi header
  - EXP header
  - WF200S Wi-Fi transceiver
  - On-board antennas
  - µ.FL connector
- **Kit contents**
  - SLEXP8022A

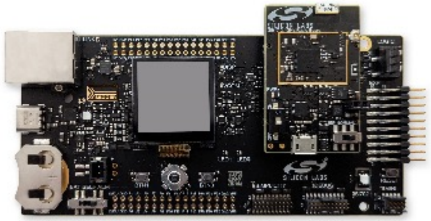
- **RS9116 Kits**
  - Modular development platform
  - Advanced development
  - RF measurements
  - Current measurements
  - Serial interface to host
- **Kit contents**
  - Baseboard
  - Wireless daughter card



- **SiWx917 Kits**[1]
  - EXP and Radio board options
- **Kit contents**
  - Radio Board Kit (SoC Mode)
    - 1 x Radio board
    - 1 x Main Board
  - EXP Kit (NCP Mode)
    - 1 x EXP board



SiWx917 EXP board



SiWx917 Dev kit

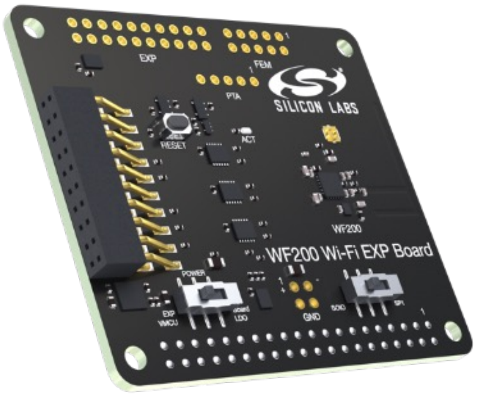
[1] SW with Matter support in Q4 2022

Selection Guidelines for Matter over Wi-Fi Ecosystems

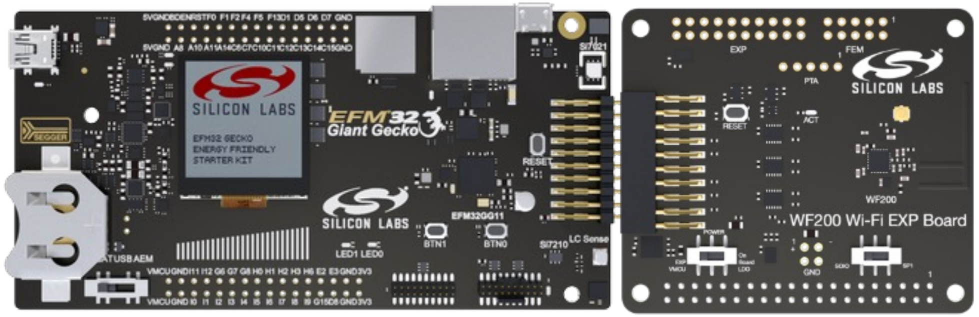
Use Case	Protocols	Mode	RS9116W + EFR32MG24[1]	WF200 + EFR32MG24[1]	SiWx917[2]	SiWx917 + EFR32MG24[1]
Matter Wi-Fi End Device	Wi-Fi 4	RCP		✓		
Matter Wi-Fi End Device	Wi-Fi 4, Bluetooth LE	NCP	✓			
Matter Wi-Fi End Device	Wi-Fi 6, Bluetooth LE	SoC			✓	
Matter Wi-Fi End Device	Wi-Fi 6, Bluetooth LE	NCP				✓

[1] In addition to the EFR32MG24, other host MCUs can also be used by porting host software

Development Kits – WF200/WFM200S



SLEXP8022A
WF200 Wi-Fi Expansion Kit

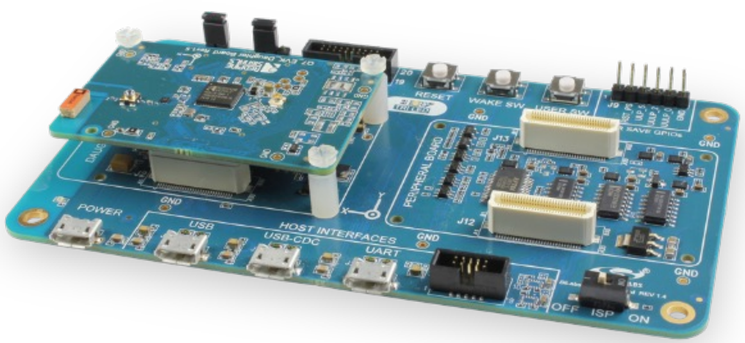


WF200 Wi-Fi Expansion board with Giant Gecko Starter Kit

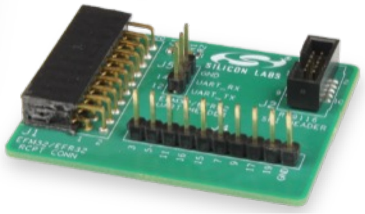
- Hardware Features
  - Raspberry Pi header
    - Supports Raspberry Pi (2, 2B & 3)
  - EXP header - support for Silicon Labs' MCUs (GG11) and Wireless MCUs (MG12) starter kits
  - WF200S Wi-Fi transceiver
  - On-board antennas
  - **µ**.FL connector for conducted measurements and adding in an external antenna

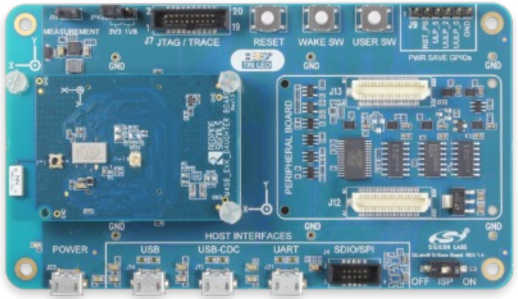
- Software tools and support
  - Open-Source drivers for Linux and RTOS
  - Example demo applications

Development Kits – RS9116

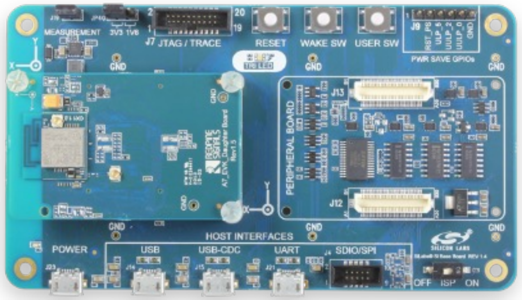


RS9116X-SB-EVK1
Single Band (QMS)





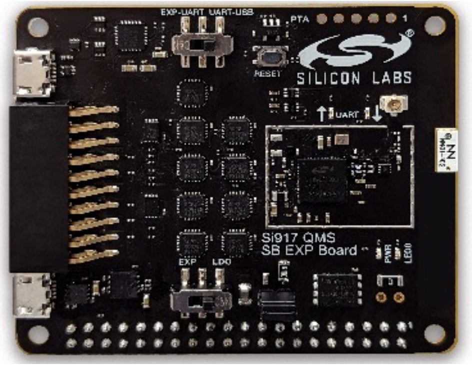
RS9116X-SB-EVK2
Single Band (B00)



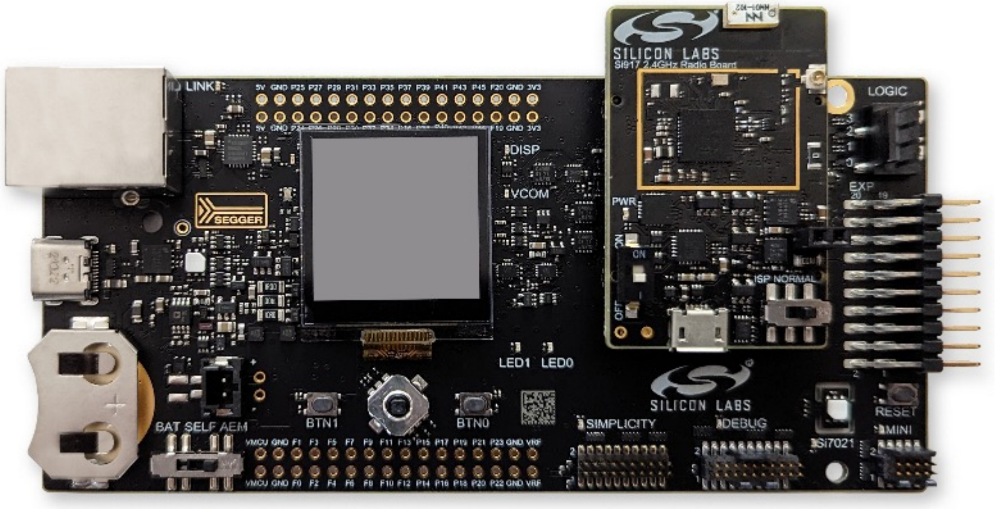
RS9116X-DB-EVK1
Dual Band (CC1)

- Same EVK for Transceiver and Full NCP
- All accessories and software included
  - Sample examples for reference
- Interface card for EFR & EFM boards

Development Kits – SiWx917 (Upcoming)



SiWx917 Expansion Board

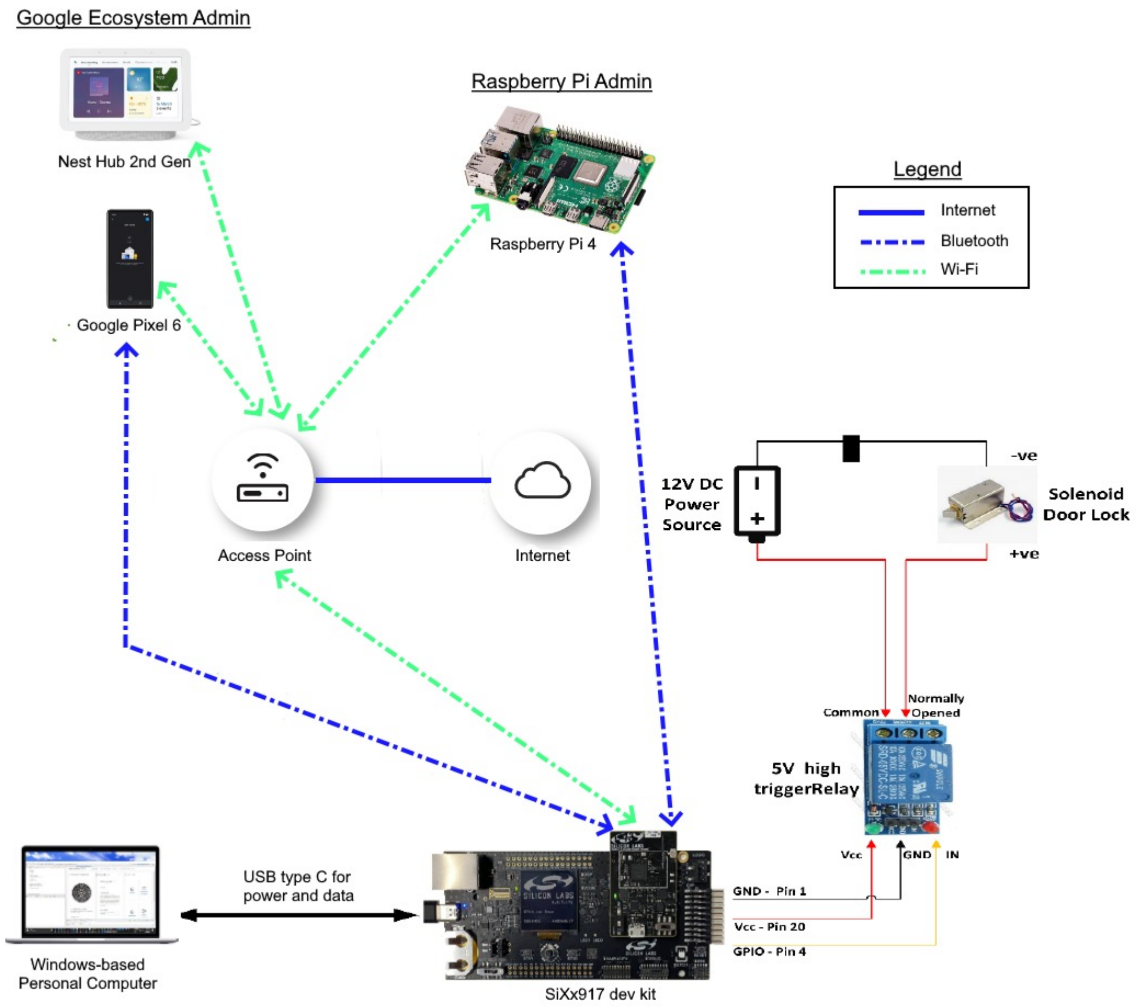


SiWx917 Development Kit

- SiWx917 EXP board for use with Silicon Labs MCU development board for NCP and RCP modes

- SiWx917 development board kit for use in SoC mode

What will we show you today?



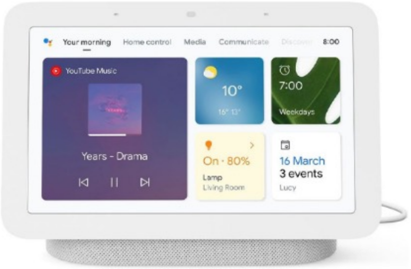
- We’ll show you how to use one of the matter over Wi-Fi code examples available in our latest release
- The example shown will be the smartlock matter over Wi-Fi example
- This code example will allow you to commission and control an IoT smartlock device
- It will demonstrate the support of multi-admin capabilities by operating with:
  - Google Ecosystem
  - Raspberry Pi

Hardware and Software Prerequisites

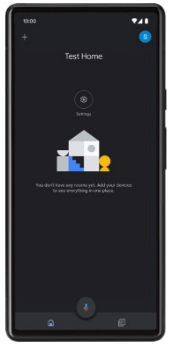
works with

BY SILICON LABS

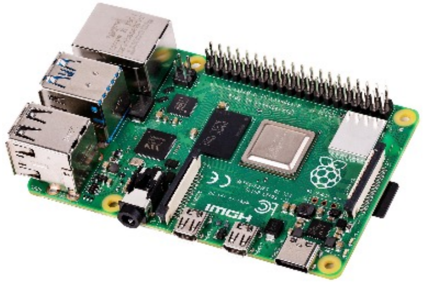
Hardware Prerequisites



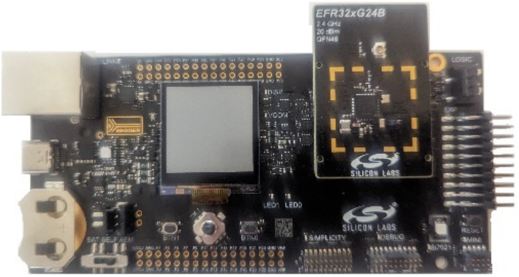
Nest Hub 2nd Gen



Google Pixel 6



Raspberry Pi 4 with 64 GB SD Card



BRD4001A dev kit baseboard with BRD4325B SiWx917 radio board

Solenoid Door Lock

5V DC Relay

To execute the SoC smartlock matter over Wi-Fi code example on an SiWx917, you will need the following hardware:

- Google Pixel 6 with Android 13 and March 5, 2023 security update or later.

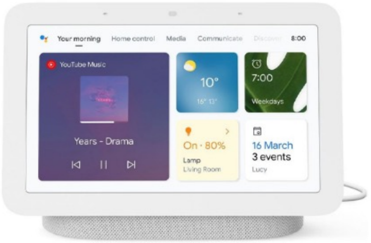
  https://www.amazon.com/Google-Pixel-Unlocked-Smartphone-Ultrawide/dp/B09HJZPFDD?th=1

- Google Nest Hub gen 2

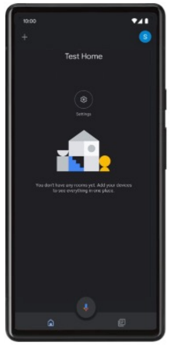
  https://store.google.com/product/nest_hub_2nd_gen?hl=en-US

- Raspberry Pi 4 with 64 GB SD Card
- Silicon Labs BRD4001A dev kit baseboard
- Silicon Labs BRD4325B SiWx917 radio board
- Windows-based PC with available USB port
- USB Type-C cable
- 5 V DC Relay – high triggered
- Solenoid door lock
- Jumper cables (female to male, male to male as needed for 5V DC relay and solenoid door lock)
- Dual-band Wi-Fi access point

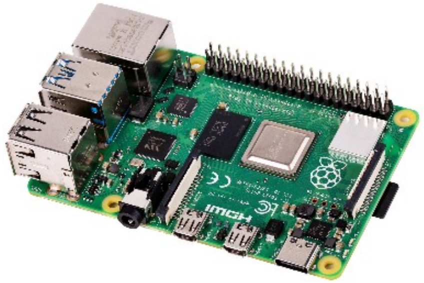
Software Prerequisites



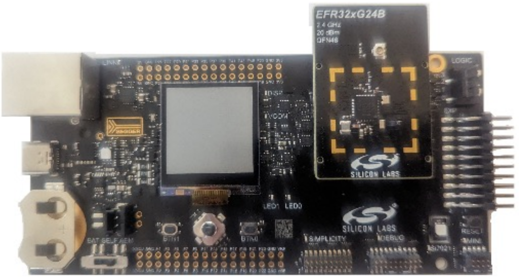
Nest Hub 2nd Gen



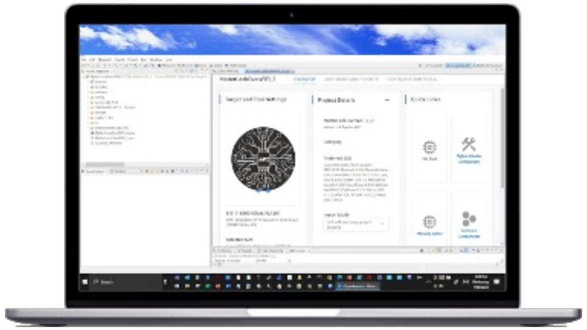
Google Pixel 6



Raspberry Pi 4 with 64 GB SD Card



BRD4001A dev kit baseboard with BRD4325B SiWx917 radio board



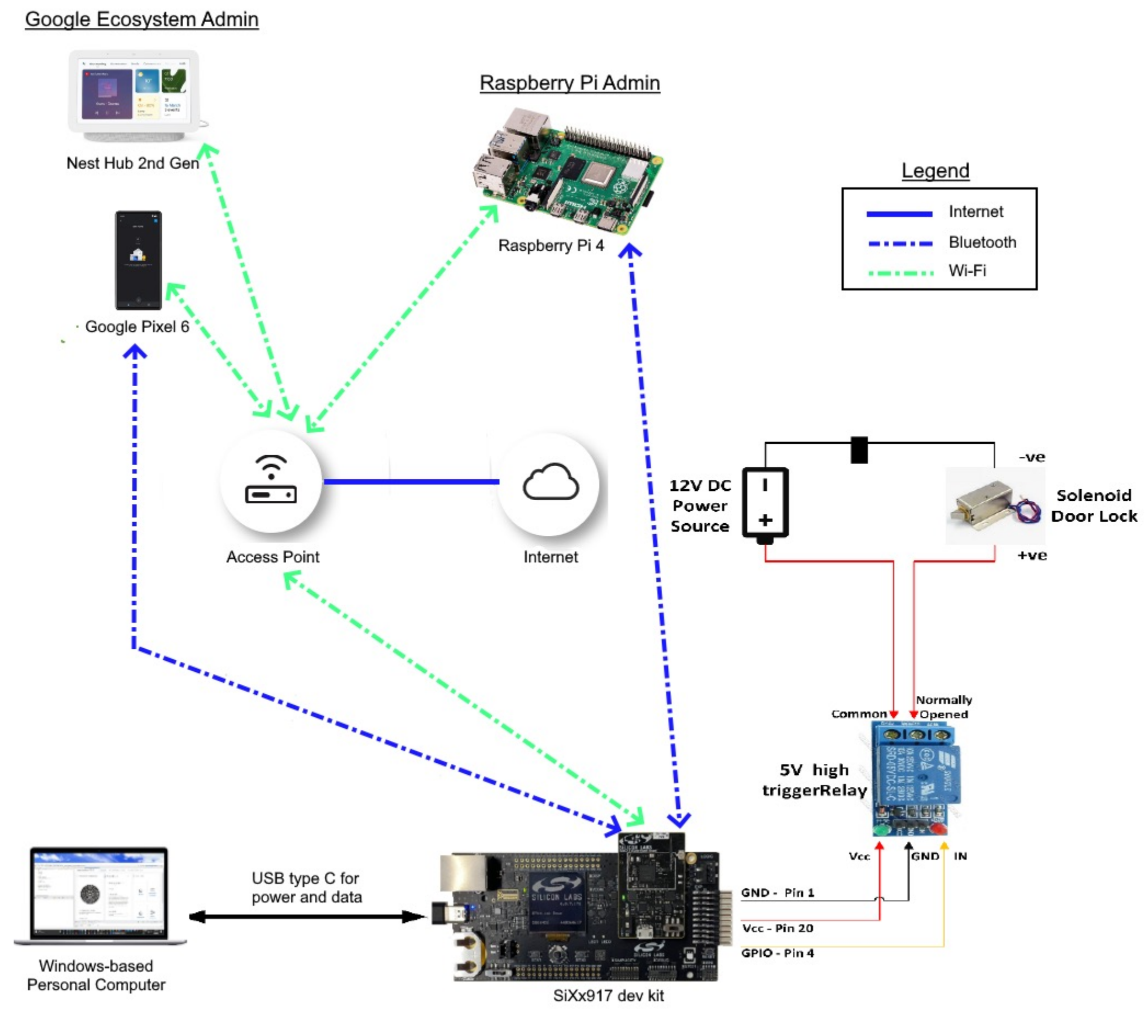
Windows-based PC

In order to execute matter over Wi-Fi code examples on an SiWx917 you will need the following software:

- Android 13 with latest security update (March 5, 2023 or later) running on Pixel 6
- Latest version of Google Home app installed on Pixel 6
- Latest version of firmware running on Nest Hub Gen 2
- Simplicity Commander version 1v14p6 or later
- Raspberry Pi Imager (Download latest version here)
- Raspberry Pi Silabs Matter Pre-Built Image (Click here to download)
- Ozone J-Link debugger (Download latest version here)
- Latest version of Simplicity Studio 5 (Download here here)
- Gecko SDK v4.3.0 or later in Simplicity Studio
- SSH Terminal for Windows. Putty or MobaXterm

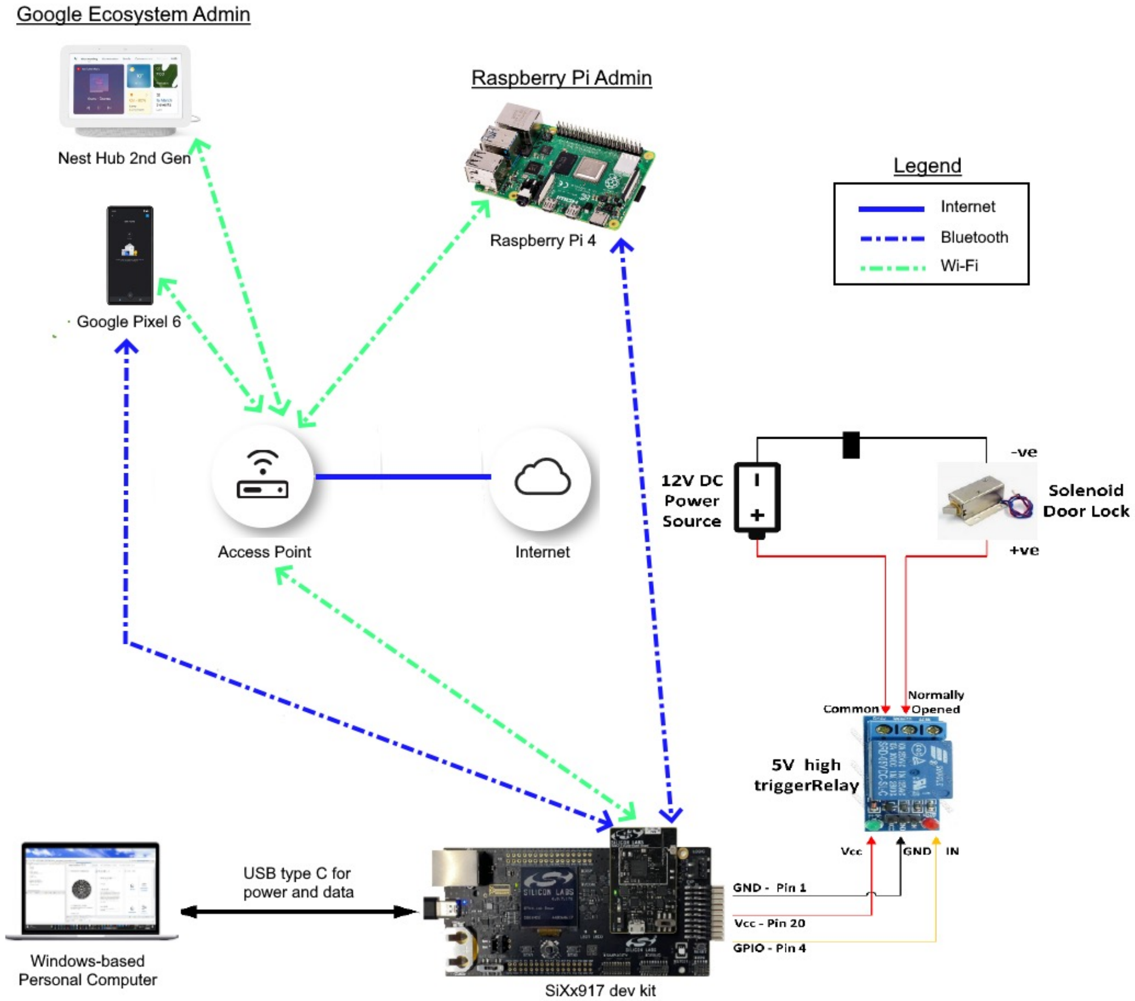
Building and Running a Matter over Wi-Fi Demo

Steps to set up and run the Matter over Wi-Fi Smartlock demo



- In today's presentation we will show you how to build and run the matter over Wi-Fi smartlock demo
- This is one of the matter over Wi-Fi demos available to you within Simplicity Studio, some others are:
  - Window Cover over Wi-Fi
  - On/Off Plug over Wi-Fi
  - Light Switch over Wi-Fi
- The smartlock matter over Wi-Fi demo will perform the following functions
  - Commissioning over BLE from Google Home app in Pixel 6
  - Associating to Wi-Fi network configured through commissioning
  - Control of solenoid door lock through Matter over Wi-Fi
- In this demo, we will show you our code's multi-admin capability support by allowing the above actions to be performed from the following controllers:
  - A Google Pixel 6
  - A Raspberry Pi 4
- We will describe its setup in the next slides

Steps to set up and run the Matter over Wi-Fi Smartlock demo



The following steps will be followed to set up the Matter over Wi-Fi Smartlock demo:

- Step 1: Setting up SDK and its extensions
- Step 2: Flash firmware onto SiWx917
- Step 3: Set up door lock circuit
- Step 4: Build smartlock binary
- Step 5: Flash smartlock binary

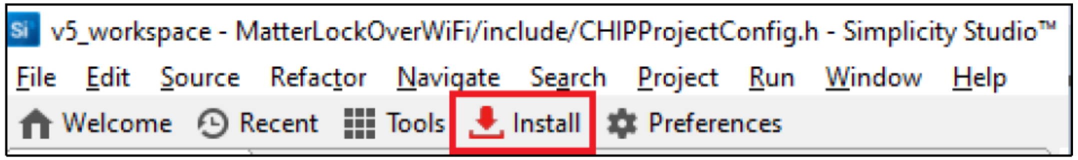
The following sections will show how to set the remaining parts of the demo, including the following:

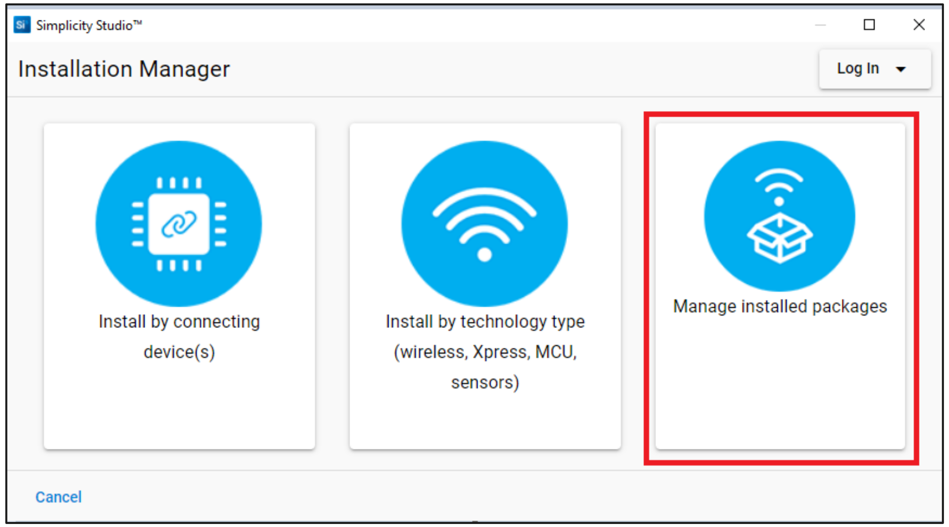
- Setting up the Google ecosystem devices (Pixel 6 and Google Smart Hub)
- Setting up the Raspberry Pi

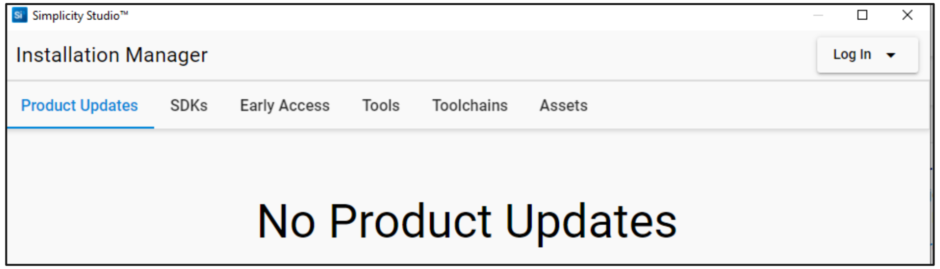
Finally, we will show how to run the demo through sections showing the following:

- How to commission and control the smartlock from Google Pixel 6
- How to commission and control the smartlock from Raspberry Pi 4

Step 1: Set up your SDK and its extensions (Part 1)



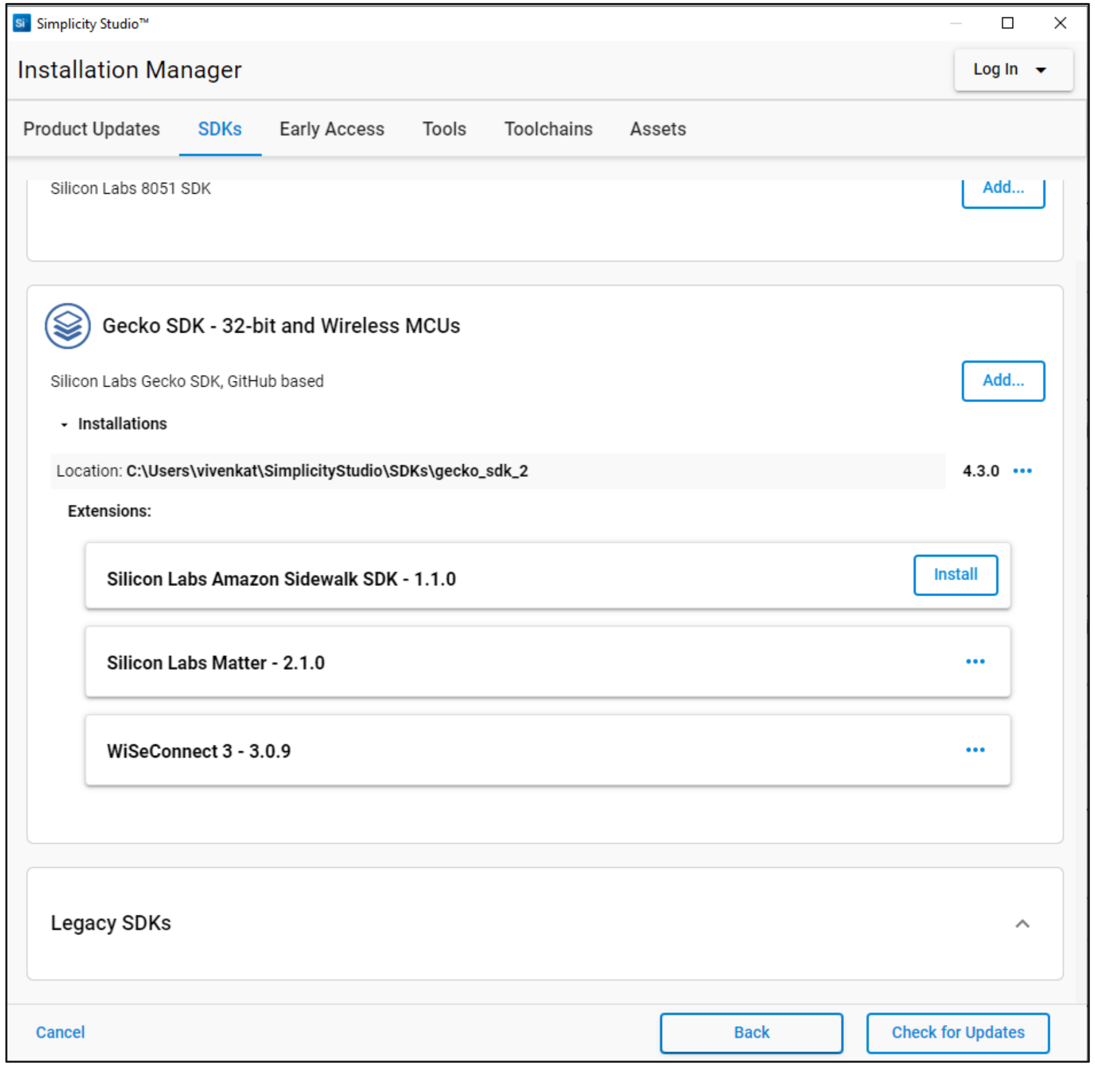

Simplicity Studio™
Installation Manager
Log In
Install by connecting device(s)
Install by technology type (wireless, Xpress, MCU, sensors)
Manage installed packages
Cancel



No Product Updates

- **Install Simplicity Studio v5 and the required Gecko SDK v4.3.1**
- **Open Simplicity Studio and make sure that all product updates are installed in Simplicity Studio by clicking on "Install" > "Manage installed packages"**

Step 1: Set up your SDK and its extensions (Part 2)



Silicon Labs 8051 SDK

Gecko SDK - 32-bit and Wireless MCUs

Silicon Labs Gecko SDK, GitHub based

Installations

Location: C:\Users\vivenkat\SimplicityStudio\SDKs\gecko_sdk_2 4.3.0

Extensions:

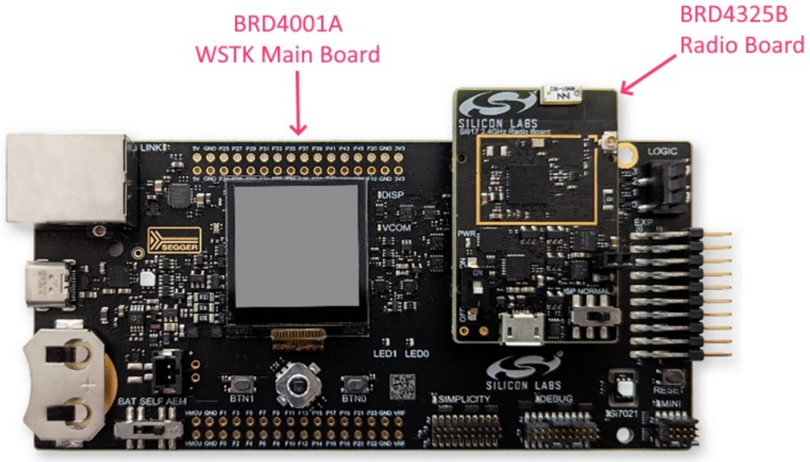
Silicon Labs Amazon Sidewalk SDK - 1.1.0

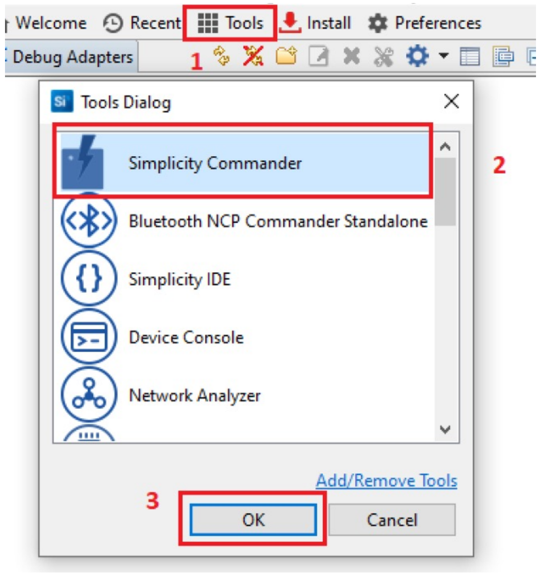
Silicon Labs Matter - 2.1.0

WiSeConnect 3 - 3.0.9

- **Navigate to "SDKs" tab, and in your latest Gecko SDK v4.3.1, add the latest "Silicon Labs Matter – 2.1.0" extension.**
- **Add the latest "WiseConnect 3 – 3.0.10" extension**
- **Note: Version numbers mentioned might be outdated. Install the latest packages available with Simplicity Studio.**

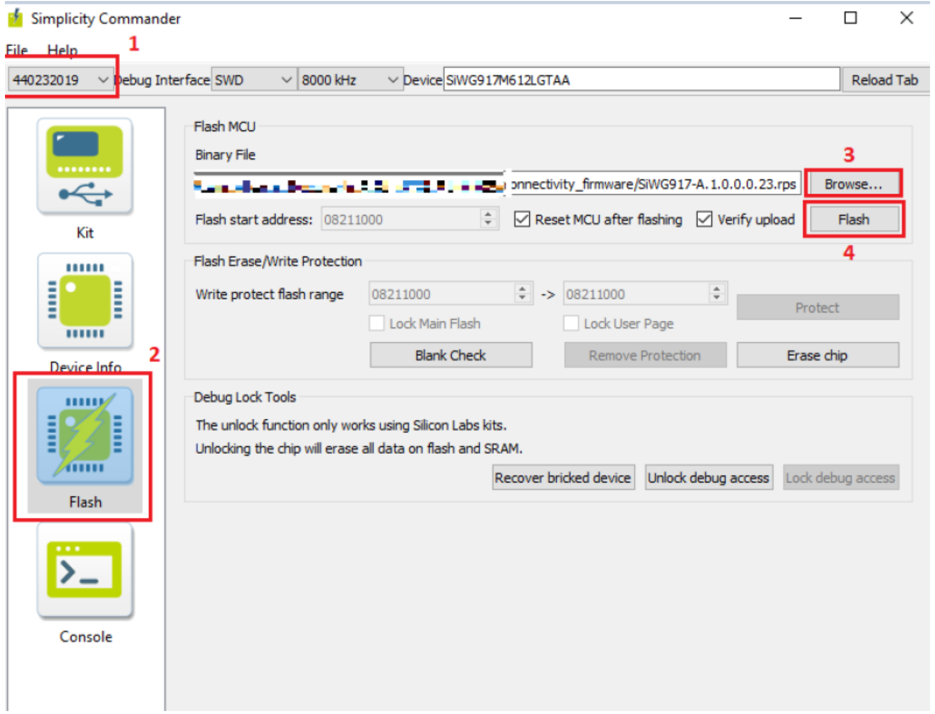
Step 2: Flash firmware onto SiWx917 (Part 1)





- **Mount the SiWx917 SoC radio board (BRD4325B) onto WSTK main board (BRD4001A) so that your system looks as shown here**
- **Connect USB Type-C cable from your PC to the WSTK main board**
- **Open Simplicity Commander by going to “Tools” in Simplicity Studio, selecting Simplicity Commander and clicking “OK”**

Step 2: Flash firmware onto SiWx917 (Part 2)



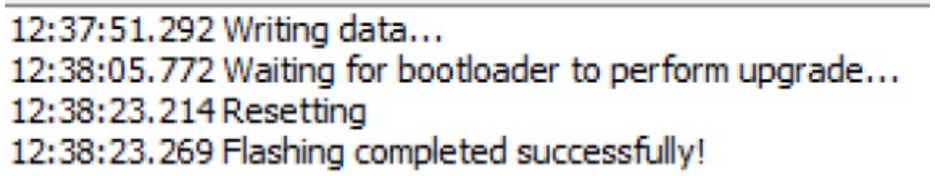
440232019 · Debug Interface: SWD · 8000 kHz · Device: SiWG917M612LGTAA

Binary File: [redacted]onnectivity_firmware/SiWG917-A.1.0.0.0.23.rps

Flash start address: 08211000 ☑ Reset MCU after flashing ☑ Verify upload

Write protect flash range: 08211000 -> 08211000

The unlock function only works using Silicon Labs kits.
Unlocking the chip will erase all data on flash and SRAM.



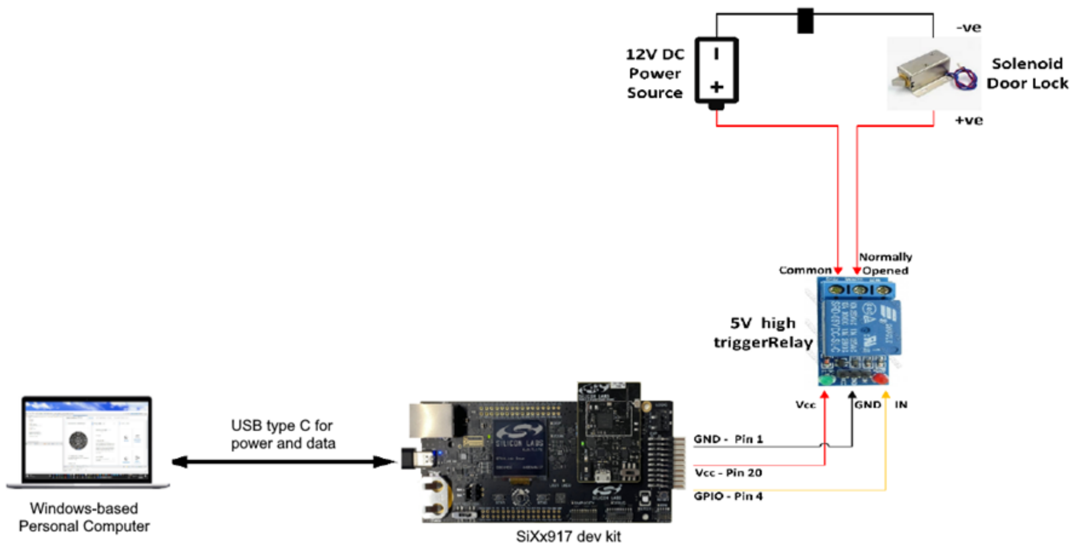
```
12:37:51.292 Writing data...
12:38:05.772 Waiting for bootloader to perform upgrade...
12:38:23.214 Resetting
12:38:23.269 Flashing completed successfully!
```

- **Select the connected kit in Simplicity Commander, then select “Flash” option.**

- **Select the required SiWG917A-X.X.x.x.x.rps file from the following directory:**
  - <SDK>\extension\wifi-sdk\connectivity_firmware

- **Click on “Flash” and wait for message confirming successful SiWx917 firmware upgrade**

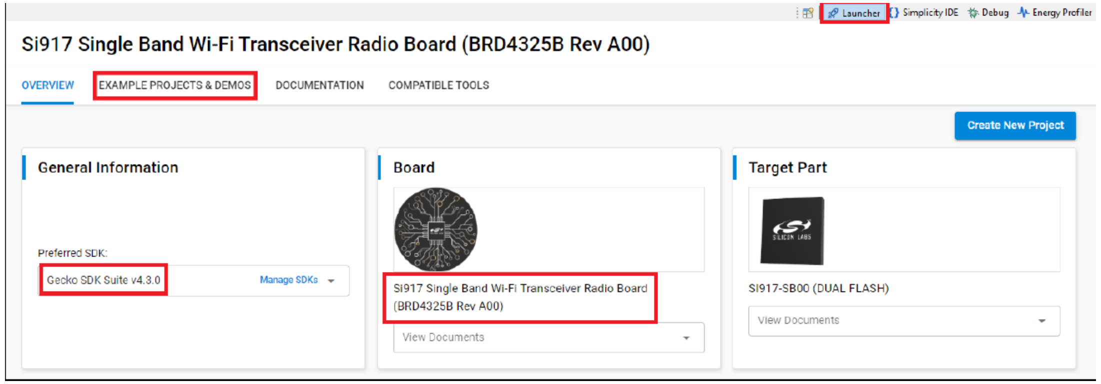
Step 3: Setup Door Lock Circuit



Please note that presented circuitry is for demo purposes only and does not constitute a final product.

- Connect GND pin on 5V high level triggered relay with pin 1 in EXP header of WSTK BRD4001A board
- Connect VCC pin on 5V high level triggered relay with pin 20 in EXP header of WSTK BRD4001A board
- Connect IN pin in 5V high level triggered relay with pin 4 in EXP header of WSTK BRD4001A board
- Connect positive of 12 V DC power source with common pin on relay
- Connect positive of Solenoid Door lock with “Normally Open” pin in relay
- Connect negative of 12 V DC power source with negative of Solenoid Door Lock
- Finally, connect USB Type-C cable from PC to WSTK BRD4001A board

Step 4: Build Smartlock Binary (Part 1)



Si917 Single Band Wi-Fi Transceiver Radio Board (BRD4325B Rev A00)

General Information

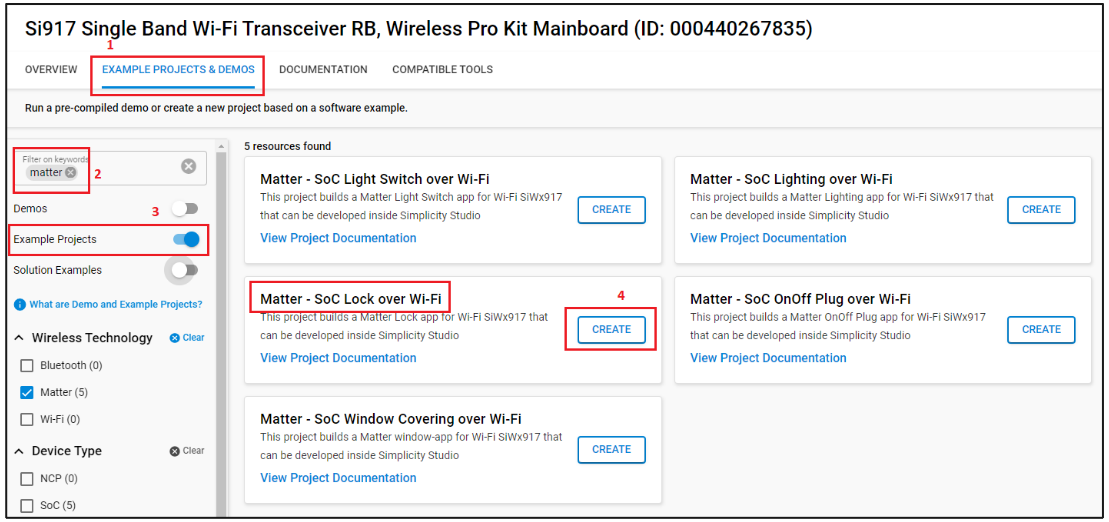
Preferred SDK: Gecko SDK Suite v4.3.0

Board

Si917 Single Band Wi-Fi Transceiver Radio Board (BRD4325B Rev A00)

Target Part

SI917-SB00 (DUAL FLASH)


1
OVERVIEW
EXAMPLE PROJECTS & DEMOS
DOCUMENTATION
COMPATIBLE TOOLS
2
3
4
CREATE
CREATE
CREATE
CREATE
CREATE
View Project Documentation
View Project Documentation
View Project Documentation
View Project Documentation
View Project Documentation
What are Demo and Example Projects?
Clear
Clear

Si917 Single Band Wi-Fi Transceiver RB, Wireless Pro Kit Mainboard (ID: 000440267835)

Run a pre-compiled demo or create a new project based on a software example.

Filter on keywords: matter

Demos ○
Example Projects ●
Solution Examples ○

Wireless Technology

- ☐ Bluetooth (0)
- ☑ Matter (5)
- ☐ Wi-Fi (0)

Device Type

- ☐ NCP (0)
- ☐ SoC (5)

5 resources found

Matter - SoC Light Switch over Wi-Fi
This project builds a Matter Light Switch app for Wi-Fi SiWx917 that can be developed inside Simplicity Studio

Matter - SoC Lighting over Wi-Fi
This project builds a Matter Lighting app for Wi-Fi SiWx917 that can be developed inside Simplicity Studio

Matter - SoC Lock over Wi-Fi
This project builds a Matter Lock app for Wi-Fi SiWx917 that can be developed inside Simplicity Studio

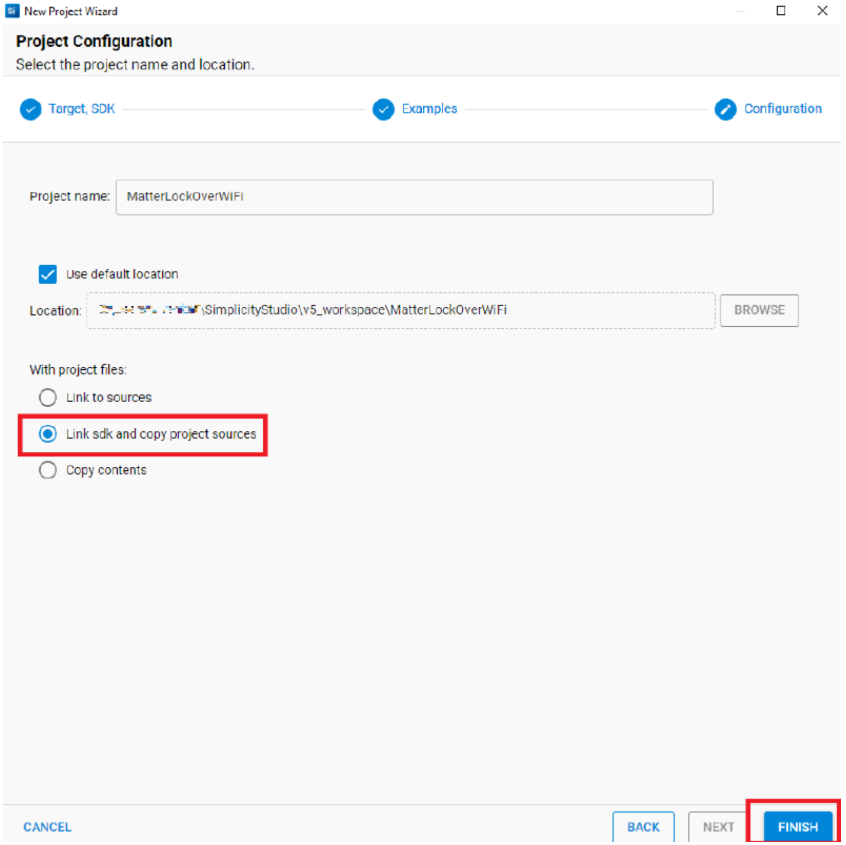
Matter - SoC OnOff Plug over Wi-Fi
This project builds a Matter OnOff Plug app for Wi-Fi SiWx917 that can be developed inside Simplicity Studio

Matter - SoC Window Covering over Wi-Fi
This project builds a Matter window-app for Wi-Fi SiWx917 that can be developed inside Simplicity Studio

- **In Simplicity Studio, navigate to "Example Projects & Demos" tab in "Launcher" view**
  - Make sure that "Preferred SDK" field is set to "Gecko SDK Suite v4.3.1

- **Filter by "Matter" in the "Wireless Technology" category, and "Example Projects", then select the "Matter – 917 SoC Lock over Wi-Fi" example and click on "Create"**

Step 4: Build Smartlock Binary (Part 2)



Project Configuration

Select the project name and location.

Project name: MatterLockOverWiFi

☑ Use default location

Location: [redacted]\SimplicityStudio\v5_workspace\MatterLockOverWiFi BROWSE

With project files:

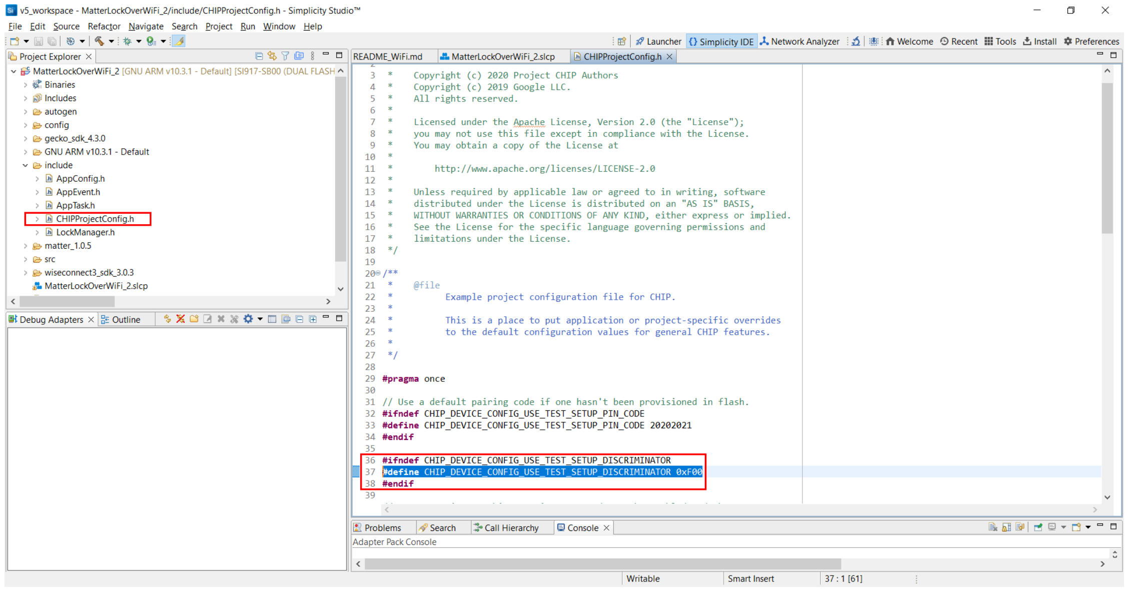
○ Link to sources

● Link sdk and copy project sources

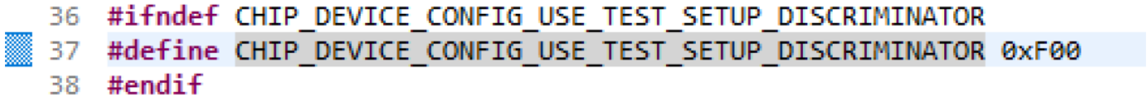
○ Copy contents

- Click on “Link sdk and copy project sources” option and then click on “Finish”

Step 4: Build Smartlock Binary (Part 3)



```
 3  *    Copyright (c) 2020 Project CHIP Authors
 4  *    Copyright (c) 2019 Google LLC.
 5  *    All rights reserved.
 6  *
 7  *    Licensed under the Apache License, Version 2.0 (the "License");
 8  *    you may not use this file except in compliance with the License.
 9  *    You may obtain a copy of the License at
10  *
11  *        http://www.apache.org/licenses/LICENSE-2.0
12  *
13  *    Unless required by applicable law or agreed to in writing, software
14  *    distributed under the License is distributed on an "AS IS" BASIS,
15  *    WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
16  *    See the License for the specific language governing permissions and
17  *    limitations under the License.
18  */
19
20 /**
21  *    @file
22  *          Example project configuration file for CHIP.
23  *
24  *          This is a place to put application or project-specific overrides
25  *          to the default configuration values for general CHIP features.
26  *
27  */
28
29 #pragma once
30
31 // Use a default pairing code if one hasn't been provisioned in flash.
32 #ifndef CHIP_DEVICE_CONFIG_USE_TEST_SETUP_PIN_CODE
33 #define CHIP_DEVICE_CONFIG_USE_TEST_SETUP_PIN_CODE 20202021
34 #endif
35
36 #ifndef CHIP_DEVICE_CONFIG_USE_TEST_SETUP_DISCRIMINATOR
37 #define CHIP_DEVICE_CONFIG_USE_TEST_SETUP_DISCRIMINATOR 0xF00
38 #endif
39
```



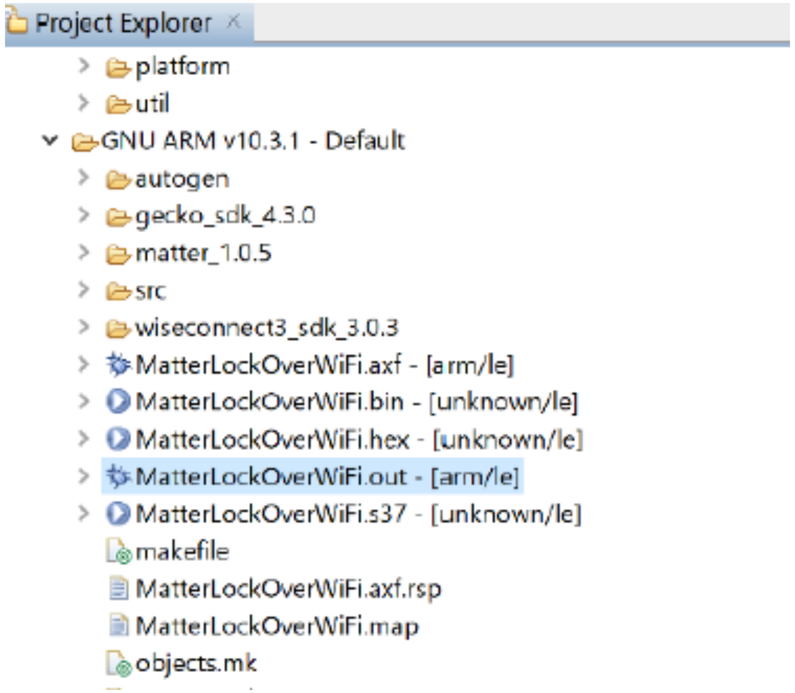
```
36 #ifndef CHIP_DEVICE_CONFIG_USE_TEST_SETUP_DISCRIMINATOR
37 #define CHIP_DEVICE_CONFIG_USE_TEST_SETUP_DISCRIMINATOR 0xF00
38 #endif
```

- **If you will have multiple smartlock devices in your environment, you will need to configure each one of them with a unique discriminator value so that they can be identified from each other**

- **To set your device’s discriminator value, open the following file:**
  - MatterLockOverWiFi\include\CHIPProjectConfig.h

- **Modify the discriminator value in line 37 of this file as shown here**

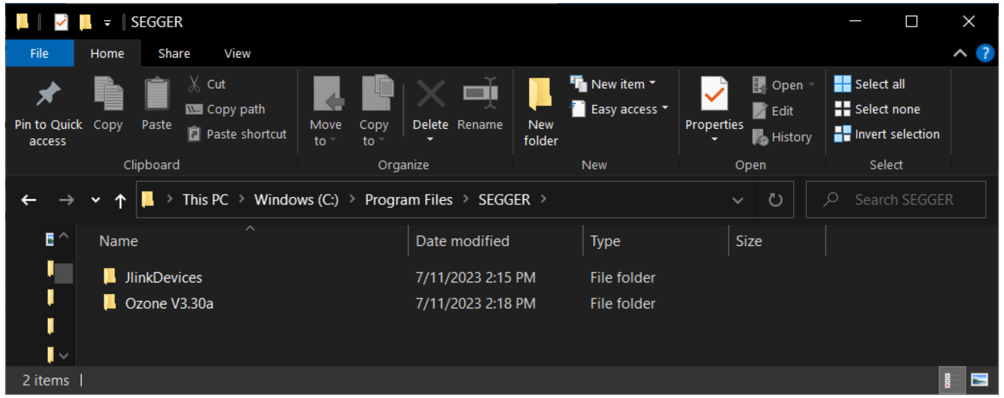
Step 4: Build Smartlock Binary (Part 4)



```
Project Explorer ✕
  > platform
  > util
v GNU ARM v10.3.1 - Default
  > autogen
  > gecko_sdk_4.3.0
  > matter_1.0.5
  > src
  > wiseconnect3_sdk_3.0.3
  > MatterLockOverWiFi.axf - [arm/le]
  > MatterLockOverWiFi.bin - [unknown/le]
  > MatterLockOverWiFi.hex - [unknown/le]
  > MatterLockOverWiFi.out - [arm/le]
  > MatterLockOverWiFi.s37 - [unknown/le]
    makefile
    MatterLockOverWiFi.axf.rsp
    MatterLockOverWiFi.map
    objects.mk
```

- **Build the example by right clicking on “MatterLockOverWiFi” project in “Project Explorer” tab and clicking on “Build Project”**
- **Once the project is built successfully, a “MatterLockOverWiFi.out” file will be seen in the project explorer tab under GNU ARM v10.3.1.**
- **This is the binary file that we will flash onto the SiWx917.**

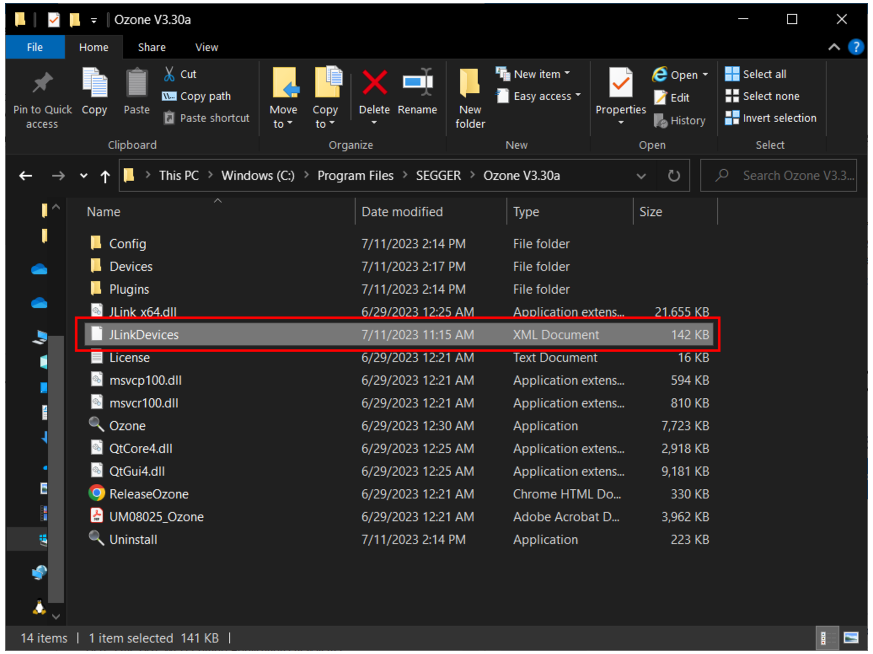
Step 5: Flash Smartlock Binary (Part 1)



This PC › Windows (C:) › Program Files › SEGGER ›

Name	Date modified	Type	Size
JlinkDevices	7/11/2023 2:15 PM	File folder	
Ozone V3.30a	7/11/2023 2:18 PM	File folder	

2 items



This PC › Windows (C:) › Program Files › SEGGER › Ozone V3.30a

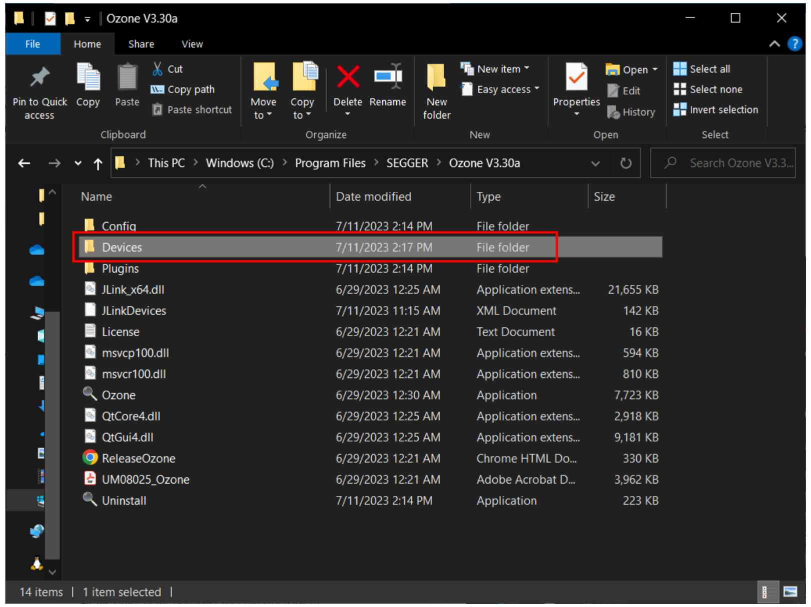
Name	Date modified	Type	Size
Config	7/11/2023 2:14 PM	File folder	
Devices	7/11/2023 2:17 PM	File folder	
Plugins	7/11/2023 2:14 PM	File folder	
JLink_x64.dll	6/29/2023 12:25 AM	Application extens...	21,655 KB
JLinkDevices	7/11/2023 11:15 AM	XML Document	142 KB
License	6/29/2023 12:21 AM	Text Document	16 KB
msvcp100.dll	6/29/2023 12:21 AM	Application extens...	594 KB
msvcr100.dll	6/29/2023 12:21 AM	Application extens...	810 KB
Ozone	6/29/2023 12:30 AM	Application	7,723 KB
QtCore4.dll	6/29/2023 12:25 AM	Application extens...	2,918 KB
QtGui4.dll	6/29/2023 12:25 AM	Application extens...	9,181 KB
ReleaseOzone	6/29/2023 12:21 AM	Chrome HTML Do...	330 KB
UM08025_Ozone	6/29/2023 12:21 AM	Adobe Acrobat D...	3,962 KB
Uninstall	7/11/2023 2:14 PM	Application	223 KB

14 items | 1 item selected 141 KB |

- **Open the path where you installed Ozone**
  - Typically (C:\Program Files\Segger)

- **Copy the "JlinkDevices.xml" file that can be downloaded from this link to the "SEGGER\Ozone Vx.x" directory:**

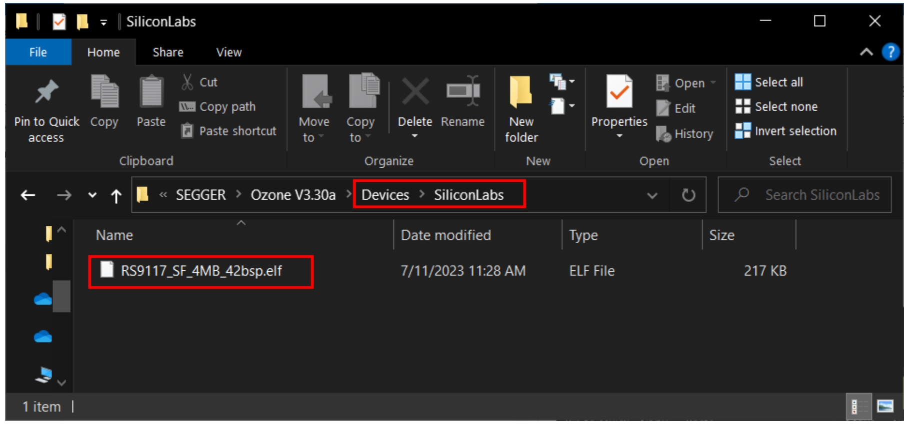
Step 5: Flash Smartlock Binary (Part 2)



This PC > Windows (C:) > Program Files > SEGGER > Ozone V3.30a

Name	Date modified	Type	Size
Config	7/11/2023 2:14 PM	File folder	
Devices	7/11/2023 2:17 PM	File folder	
Plugins	7/11/2023 2:14 PM	File folder	
JLink_x64.dll	6/29/2023 12:25 AM	Application extens...	21,655 KB
JLinkDevices	7/11/2023 11:15 AM	XML Document	142 KB
License	6/29/2023 12:21 AM	Text Document	16 KB
msvcp100.dll	6/29/2023 12:21 AM	Application extens...	594 KB
msvcr100.dll	6/29/2023 12:21 AM	Application extens...	810 KB
Ozone	6/29/2023 12:30 AM	Application	7,723 KB
QtCore4.dll	6/29/2023 12:25 AM	Application extens...	2,918 KB
QtGui4.dll	6/29/2023 12:25 AM	Application extens...	9,181 KB
ReleaseOzone	6/29/2023 12:21 AM	Chrome HTML Do...	330 KB
UM08025_Ozone	6/29/2023 12:21 AM	Adobe Acrobat D...	3,962 KB
Uninstall	7/11/2023 2:14 PM	Application	223 KB

14 items | 1 item selected |



« SEGGER > Ozone V3.30a > Devices > SiliconLabs

Name	Date modified	Type	Size
RS9117_SF_4MB_42bsp.elf	7/11/2023 11:28 AM	ELF File	217 KB

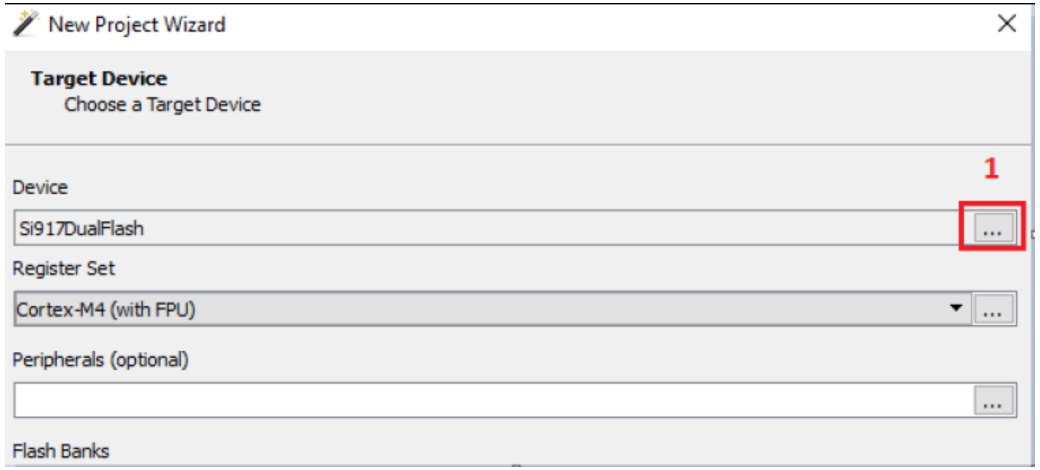
1 item |

- If you don’t already have a “Devices” directory under the SEGGER\Ozone Vx.x directory, create it as shown here

- Within the “Devices” directory, create a “SiliconLabs”subdirectory

- Donwload the “RS917_SF_4MB_42bsp.elf” file available in this link and copy it to the “SiliconLabs” subdirectory

Step 5: Flash Smartlock Binary (Part 3)



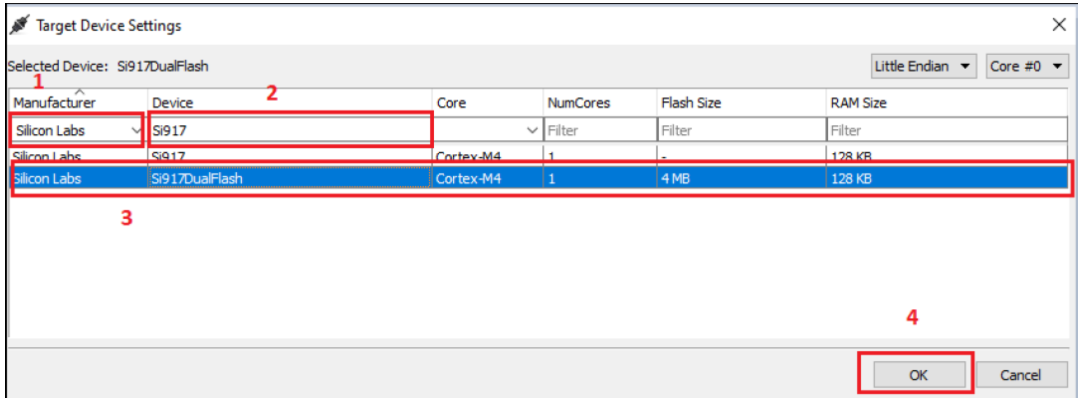
Target Device
Choose a Target Device

Device: Si917DualFlash

Register Set: Cortex-M4 (with FPU)

Peripherals (optional):

Flash Banks



Selected Device: Si917DualFlash

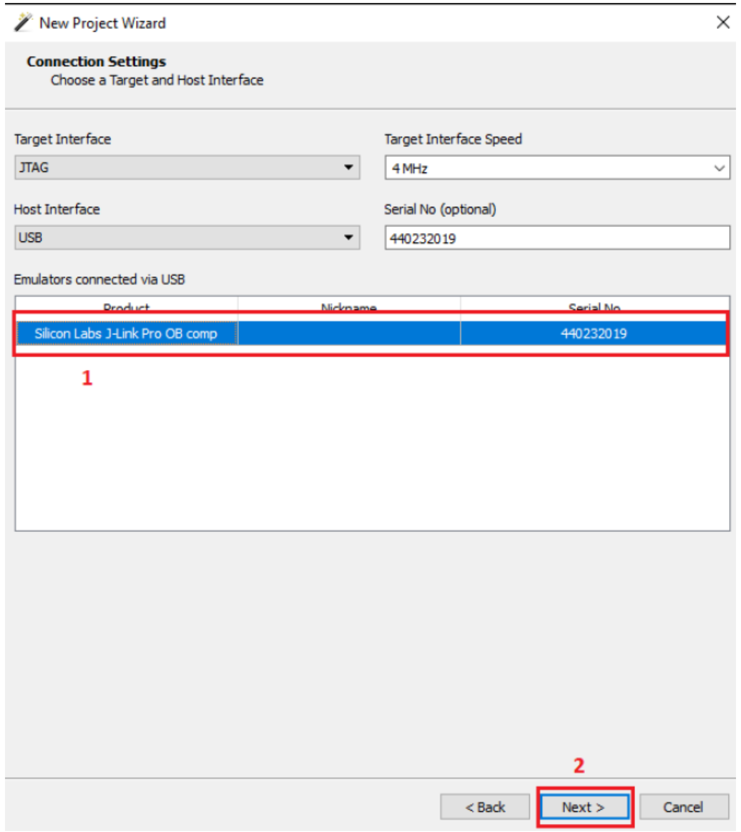
Little Endian

Core #0

Manufacturer	Device	Core	NumCores	Flash Size	RAM Size
Silicon Labs	Si917		Filter	Filter	Filter
Silicon Labs	Si917	Cortex-M4	1	-	128 KB
Silicon Labs	Si917DualFlash	Cortex-M4	1	4 MB	128 KB

- **Now plug your SiWx917 setup to your PC and launch the Ozone debugger. In the New Project Wizard click on the three dots (…) and select the following in the following screen:**
  - Manufacturer: Silicon Labs
  - Device: Si917DualFlash
- **Once you have done those selections, click on "OK"**

Step 5: Flash Smartlock Binary (Part 4)



New Project Wizard

Connection Settings

Choose a Target and Host Interface

Target Interface: JTAG

Target Interface Speed: 4 MHz

Host Interface: USB

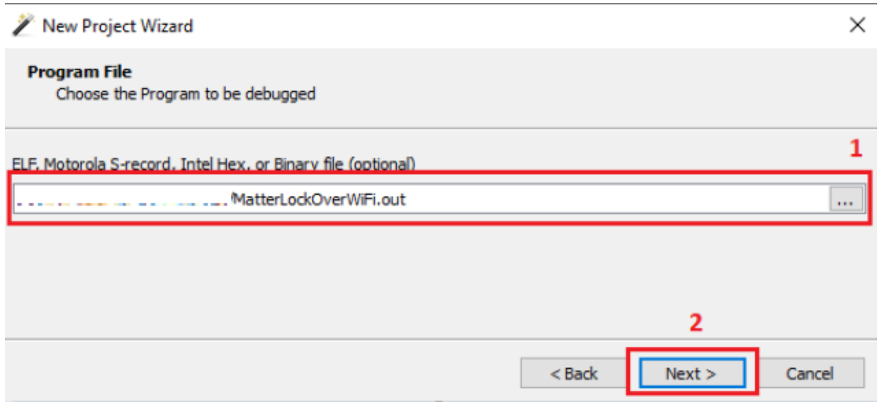
Serial No (optional): 440232019

Emulators connected via USB

Product	Nickname	Serial No
Silicon Labs J-Link Pro OB comp		440232019

- **In the next screen, select your Silicon Labs product under “Emulators connected via USB” and then click on “Next”**
  - Manufacturer: Silicon Labs
  - Device: Si917DualFlash
- **Once you have done those selections, click on “OK”**

Step 5: Flash Smartlock Binary (Part 5)



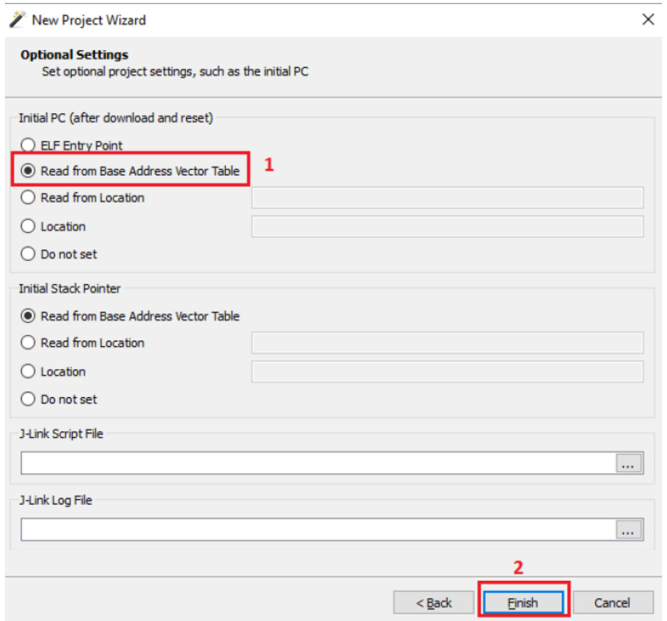
New Project Wizard

Program File

Choose the Program to be debugged

ELF, Motorola S-record, Intel Hex, or Binary file (optional)

[illegible]/MatterLockOverWiFi.out



New Project Wizard

Optional Settings

Set optional project settings, such as the initial PC

Initial PC (after download and reset)

- ○ ELF Entry Point
- ● Read from Base Address Vector Table
- ○ Read from Location
- ○ Location
- ○ Do not set

Initial Stack Pointer

- ● Read from Base Address Vector Table
- ○ Read from Location
- ○ Location
- ○ Do not set

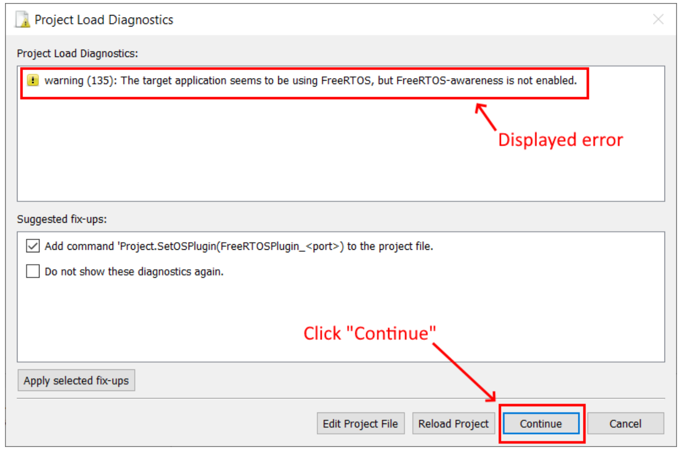
J-Link Script File

J-Link Log File

- In the next screen, browse to the "MatterLockOverWiFi.out" binary file that you generated with your build on Simplicity Studio and click on "Next"

- In "Optional Settings" make sure that "Initial PC" is set to "Read from Base Address Vector Table" and then click on "Finish"

Step 5: Flash Smartlock Binary (Part 6)


Project Load Diagnostics

Project Load Diagnostics:

warning (135): The target application seems to be using FreeRTOS, but FreeRTOS-awareness is not enabled.

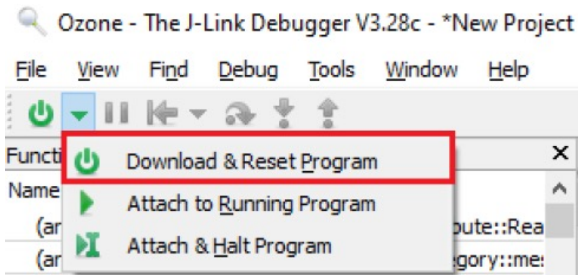
Displayed error

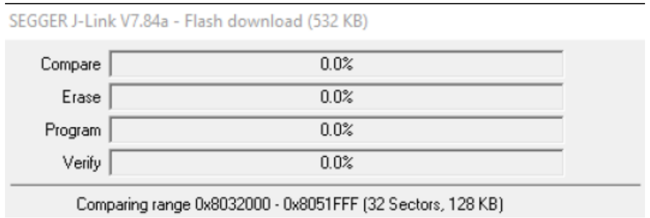
Suggested fix-ups:

☑ Add command 'Project.SetOSPlugin(FreeRTOSPlugin_<port>) to the project file.

□ Do not show these diagnostics again.

Click "Continue"


Ozone - The J-Link Debugger V3.28c - *New Project


```
SEGGER J-Link V7.84a - Flash download (532 KB)

 Compare  0.0%
   Erase  0.0%
 Program  0.0%
  Verify  0.0%

Comparing range 0x8032000 - 0x8051FFF (32 Sectors, 128 KB)
```

- **A diagnostics warning may be displayed regarding FreeRTOS being detected. Simply disregard it and click “Continue”**

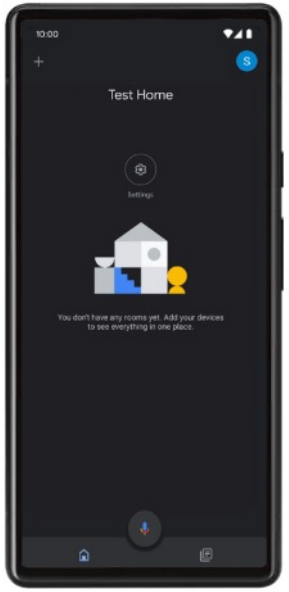
- **After the binary is downloaded, click on “Download & Reset Program” in the dropdown next to the Power button on Ozone’s top right**

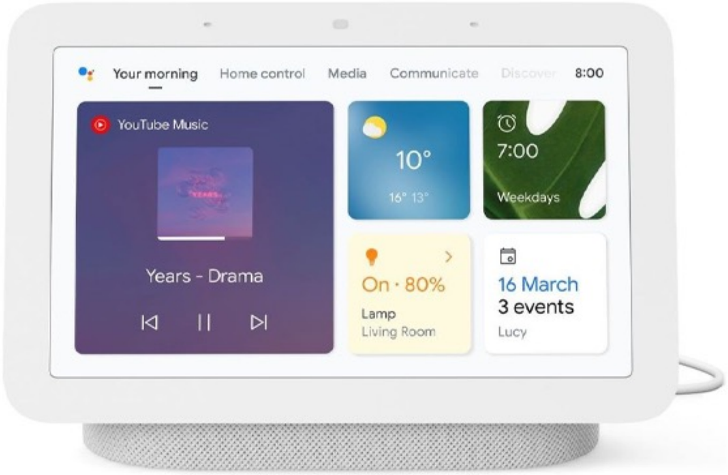
- **This will erase and program the SiWx917**

Setting Up Google Ecosystem Devices for Matter over Wi-Fi demo

works with
BY SILICON LABS

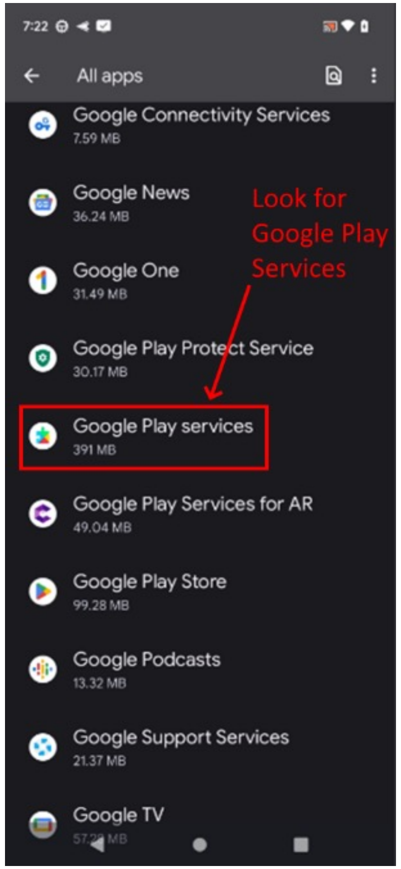
Setup Pixel 6 and Nest Hub setup (Part 1)


10:00
Test Home
Settings
You don't have any rooms yet. Add your devices to see everything in one place.



- We will need to set up several things in order to get our Pixel 6 and Nest Hub Setup
  - Set up a Google Home
  - Create a Matter Project and add a Matter Integration to it for Smartlock Matter devices
  - Open a webpage with the matter smartlock device QR code
- Let’s show this in the next few pages

Setup Pixel 6 and Nest Hub setup (Part 2)



7:22

All apps

Google Connectivity Services
7.59 MB

Google News
36.24 MB

Google One
31.49 MB

Google Play Protect Service
30.17 MB

Google Play services
391 MB

Google Play Services for AR
49.04 MB

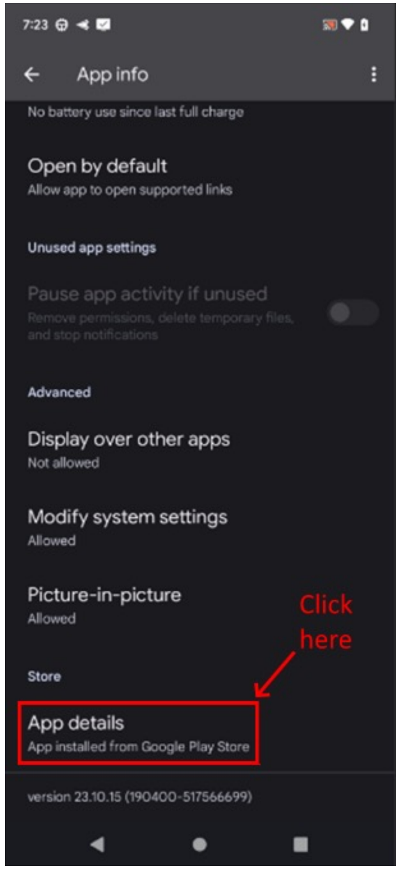
Google Play Store
99.28 MB

Google Podcasts
13.32 MB

Google Support Services
21.37 MB

Google TV
57.72 MB

Look for Google Play Services



7:23

App info

No battery use since last full charge

Open by default
Allow app to open supported links

Unused app settings

Pause app activity if unused
Remove permissions, delete temporary files, and stop notifications

Advanced

Display over other apps
Not allowed

Modify system settings
Allowed

Picture-in-picture
Allowed

Store

App details
App installed from Google Play Store

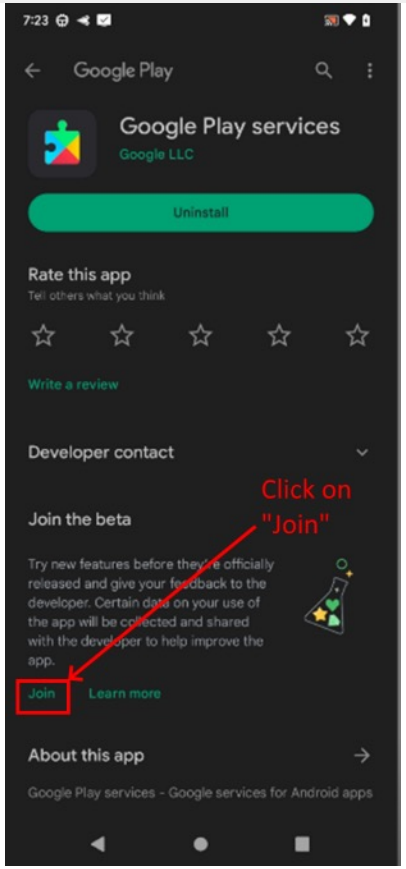
version 23.10.15 (190400-517566699)

Click here

- Before doing anything, we will need to make sure that we are running the latest beta version of Google Play Services. To do so, in your phone in settings -> apps, look for Google Play Services as shown here.

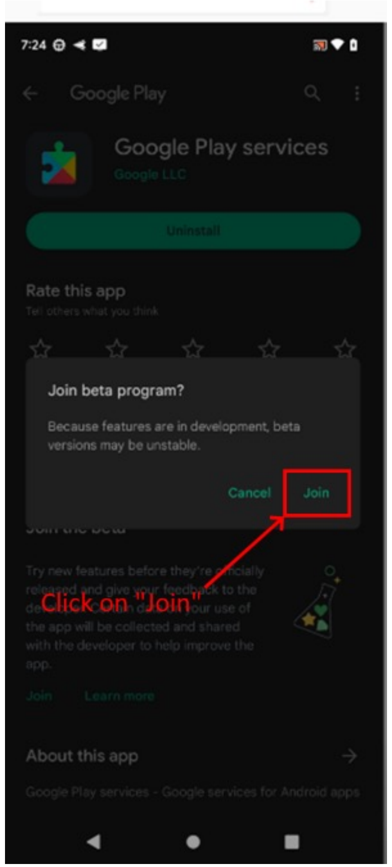
- Once the page for the app is open, go to the bottom of it and click on “App details”

Setup Pixel 6 and Nest Hub setup (Part 3)



Try new features before they're officially released and give your feedback to the developer. Certain data on your use of the app will be collected and shared with the developer to help improve the app.

Click on "Join"



Join beta program?

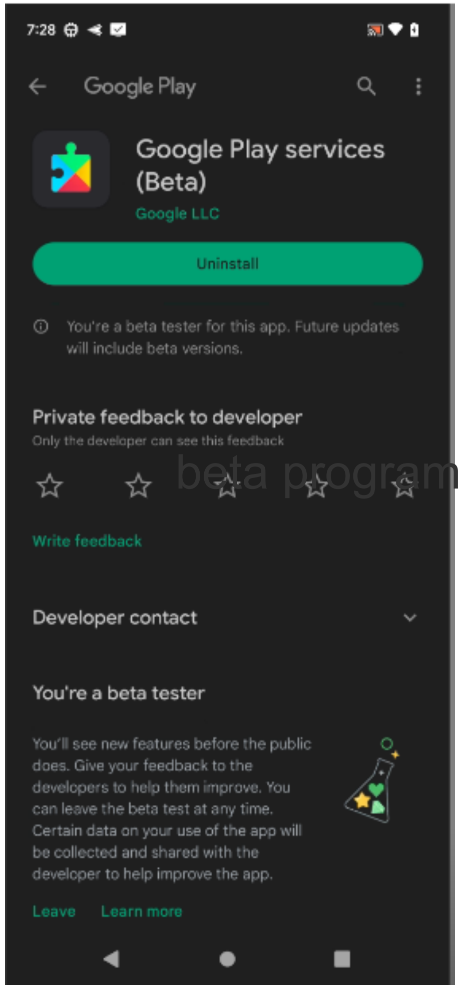
Because features are in development, beta versions may be unstable.

Try new features before they're officially released and give your feedback to the developer. Certain data on your use of the app will be collected and shared with the developer to help improve the app.

Click on "Join"

- **In the next page, click on "Join" to Join the beta program.**
- **Click on "Join" on the popup to accept joining the beta program**

Setup Pixel 6 and Nest Hub setup (Part 4)



7:28

Google Play

Google Play services (Beta)

Google LLC

Uninstall

You're a beta tester for this app. Future updates will include beta versions.

Private feedback to developer

Only the developer can see this feedback

beta program

☆ ☆ ☆ ☆ ☆

Write feedback

Developer contact

You're a beta tester

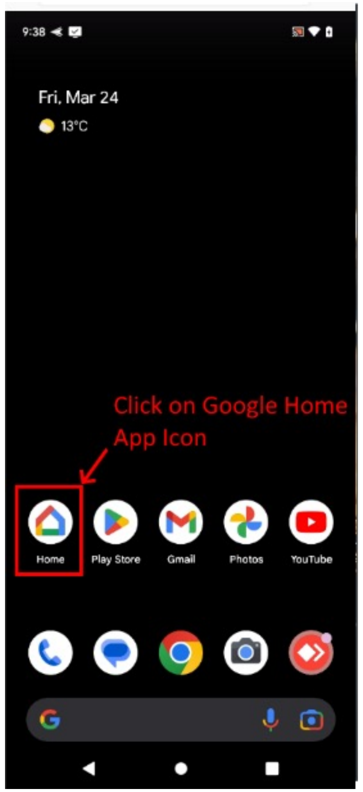
You'll see new features before the public does. Give your feedback to the developers to help them improve. You can leave the beta test at any time. Certain data on your use of the app will be collected and shared with the developer to help improve the app.

Leave Learn more

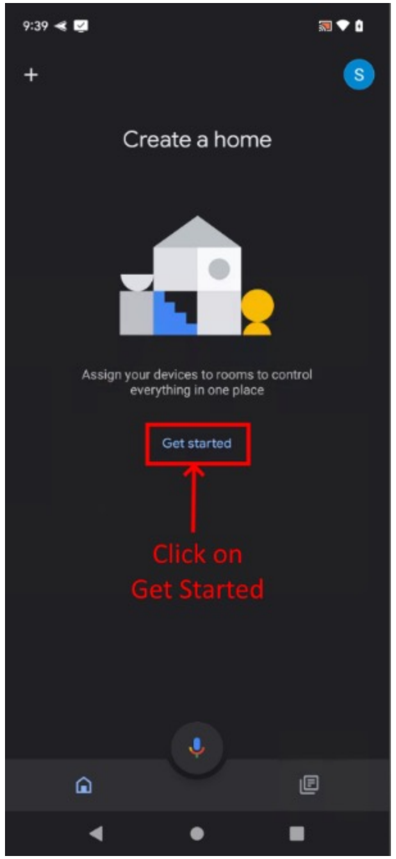
- Wait a few minutes to let the latest beta version of Google Play services to be installed and then close and reopen Google Play Services on settings->apps to check that it is now in beta version as shown in this screenshot.

- After verifying this, make sure to update the Google home app on your phone to the latest version

Setup Pixel 6 and Nest Hub setup (Part 5)



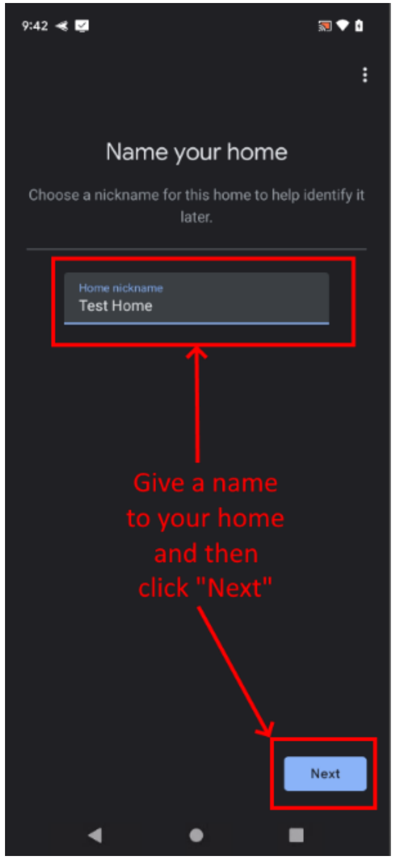
Click on Google Home App Icon


9:39
+
S
Create a home
Assign your devices to rooms to control everything in one place
Get started

Click on Get Started

- Now that we have the latest Google Play services beta version and latest Google home app in your phone, lets create a new home in your Pixel 6
- To do so, first open the Google Home app by clicking on its icon on your phone
- After opening the app, click on “Get started”

Setup Pixel 6 and Nest Hub setup (Part 6)



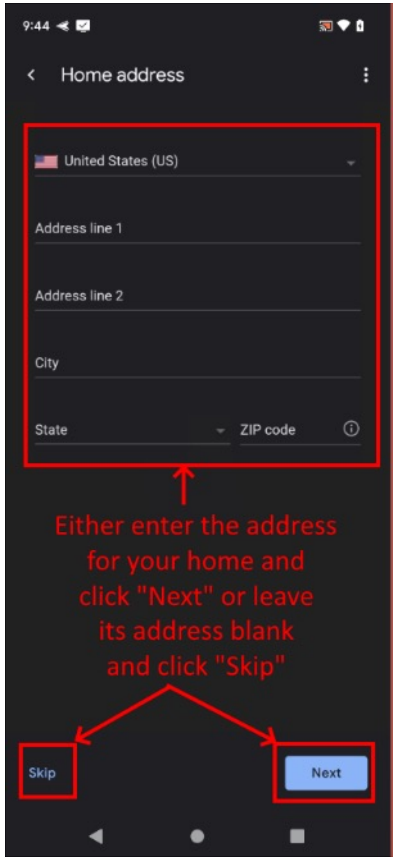
9:42

Name your home

Choose a nickname for this home to help identify it later.

Home nickname: Test Home

Give a name to your home and then click "Next"



9:44

Home address

United States (US)

Address line 1

Address line 2

City

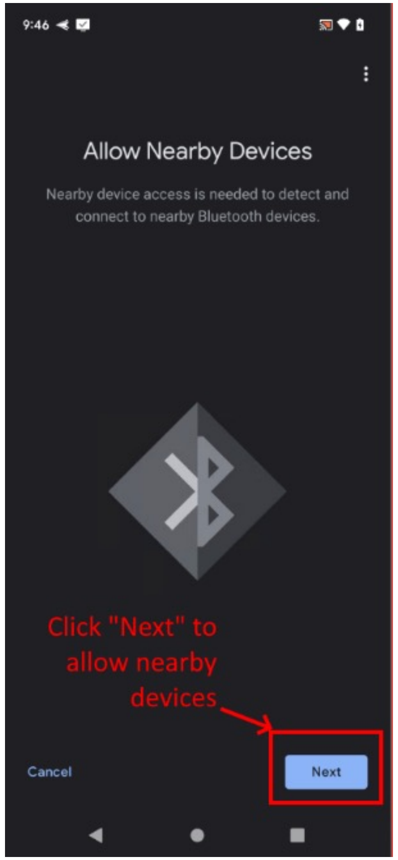
State

ZIP code

Either enter the address for your home and click "Next" or leave its address blank and click "Skip"

- Now give a name to your home and click on “Next”
- Either enter the address for your home and click “Next” or leave its address blank and click “Skip”

Setup Pixel 6 and Nest Hub setup (Part 7)

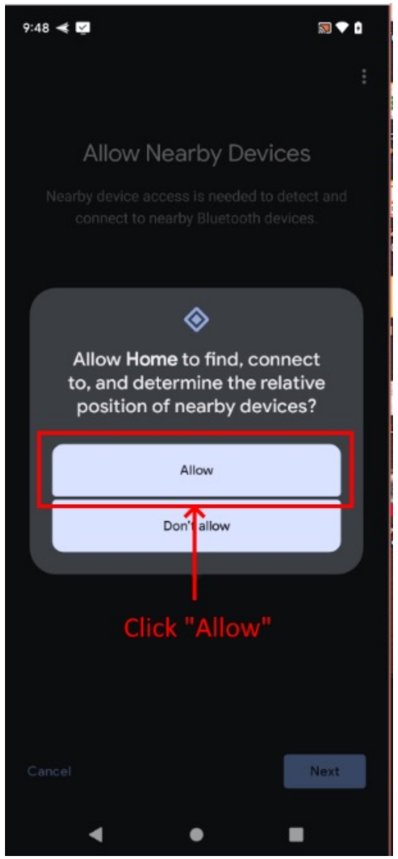


9:46

Allow Nearby Devices

Nearby device access is needed to detect and connect to nearby Bluetooth devices.

Click "Next" to allow nearby devices



9:48

Allow Nearby Devices

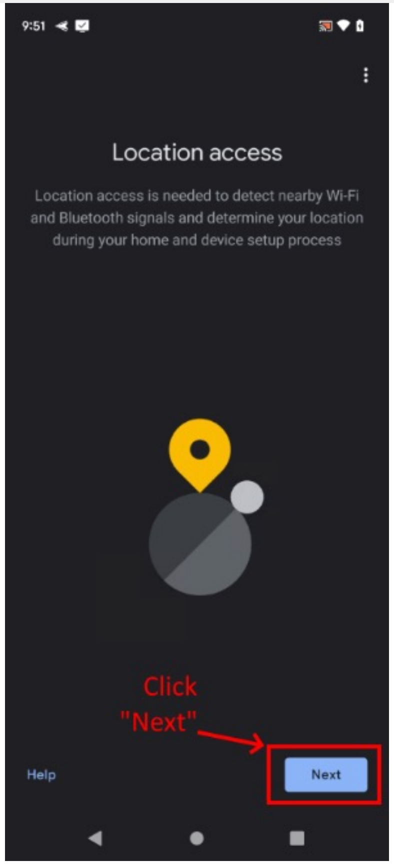
Nearby device access is needed to detect and connect to nearby Bluetooth devices.

Allow **Home** to find, connect to, and determine the relative position of nearby devices?

Click "Allow"

- Click “Next” to allow nearby devices
- Click “Allow” on the pop-up screen

Setup Pixel 6 and Nest Hub setup (Part 8)

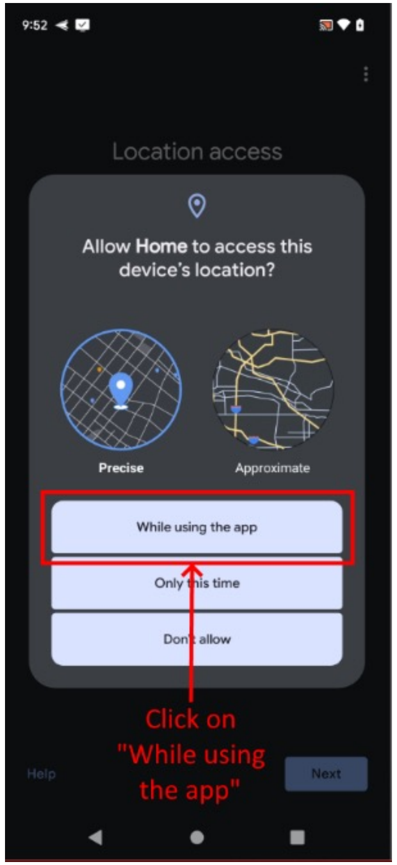


9:51

Location access

Location access is needed to detect nearby Wi-Fi and Bluetooth signals and determine your location during your home and device setup process

Click "Next"



9:52

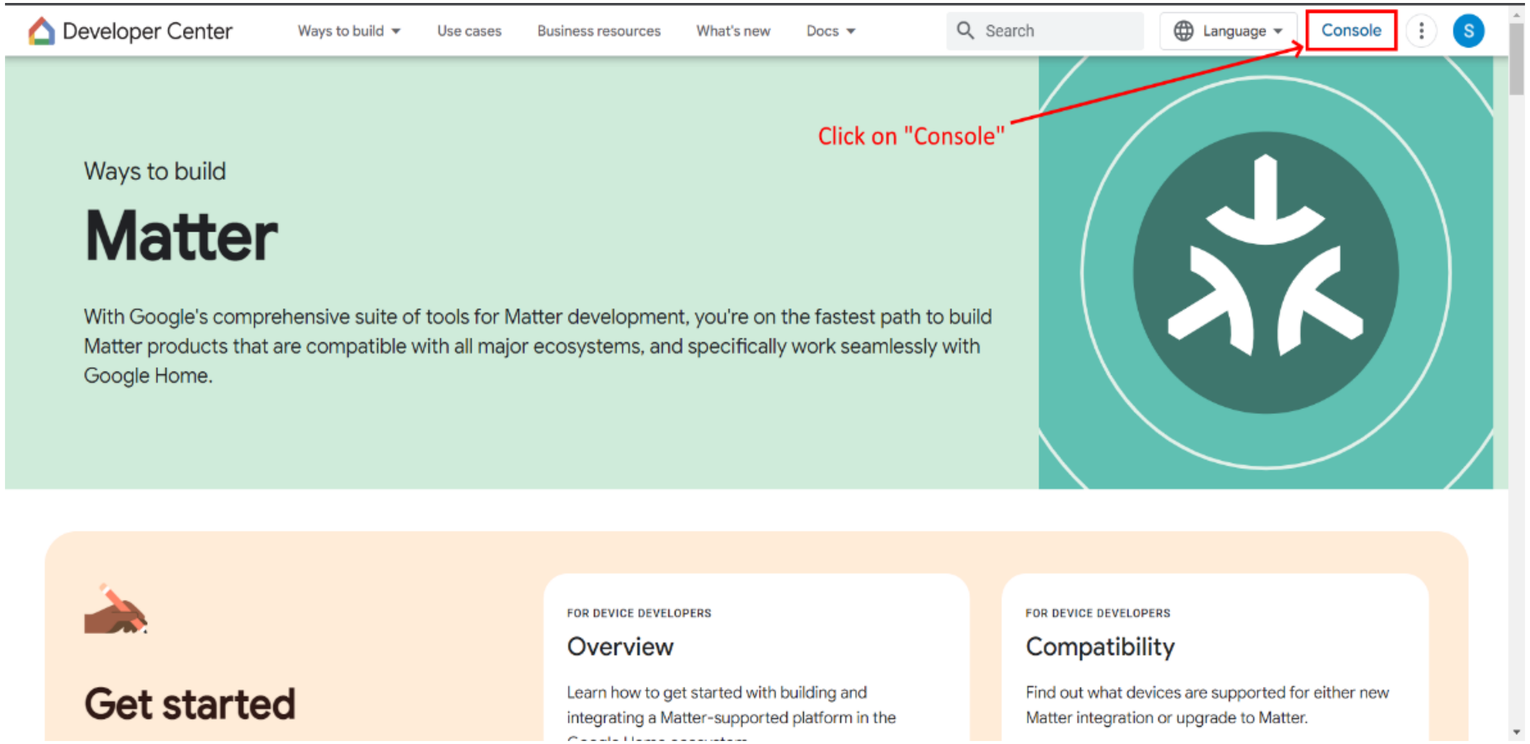
Location access

Allow **Home** to access this device's location?

Click on "While using the app"

- Click "Next" to allow location access
- Click on "While using the app"

Setup Pixel 6 and Nest Hub setup (Part 9)



Ways to build

Matter

With Google's comprehensive suite of tools for Matter development, you're on the fastest path to build Matter products that are compatible with all major ecosystems, and specifically work seamlessly with Google Home.

Get started

FOR DEVICE DEVELOPERS

Overview

Learn how to get started with building and integrating a Matter-supported platform in the

FOR DEVICE DEVELOPERS

Compatibility

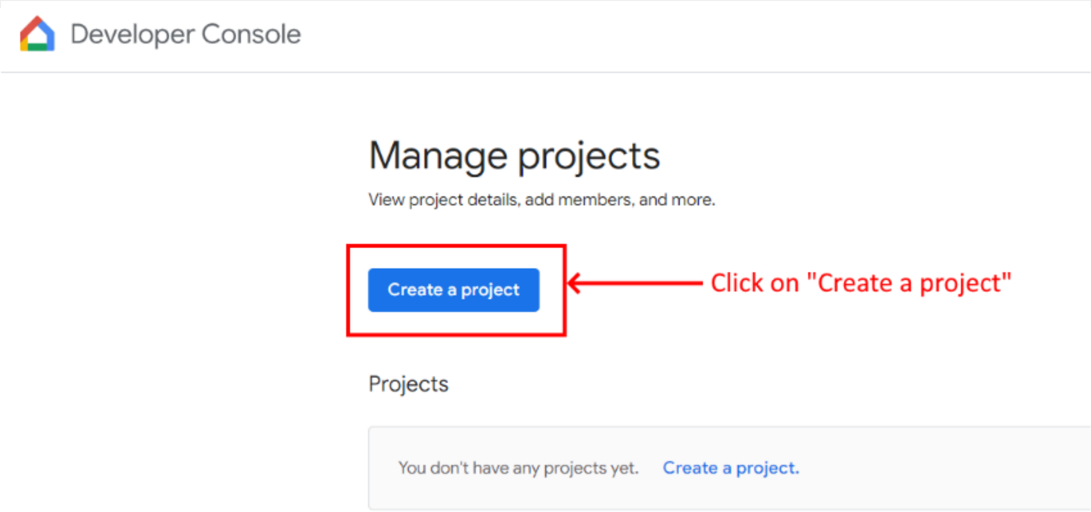
Find out what devices are supported for either new Matter integration or upgrade to Matter.

- Now that we have created a home in our Pixel phone, add your Nest Hub to that home
- After this, on a browser on your PC go to the following webpage to create a project:

  https://developers.home.google.com/matter

- Then click on “Console” at the top of the page as shown here.

Setup Pixel 6 and Nest Hub setup (Part 10)



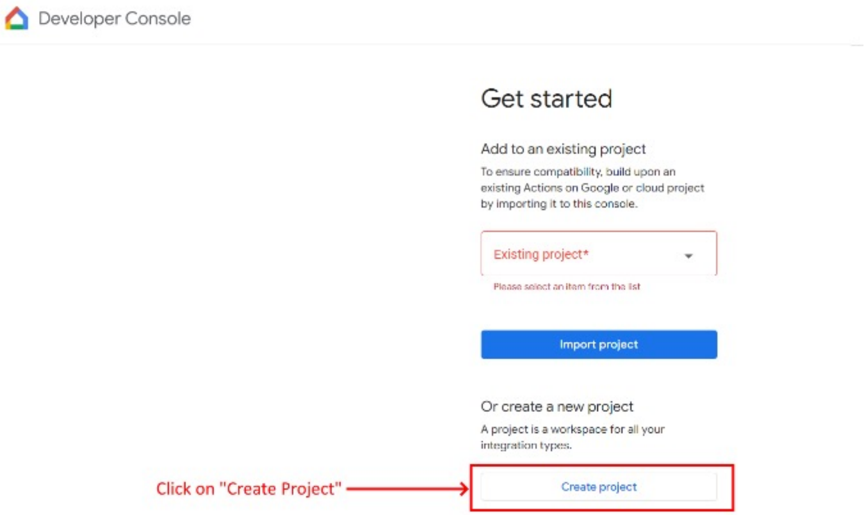
Manage projects

View project details, add members, and more.

Click on "Create a project"

Projects

You don't have any projects yet. Create a project.



Get started

Add to an existing project

To ensure compatibility, build upon an existing Actions on Google or cloud project by importing it to this console.

Please select an item from the list

Or create a new project

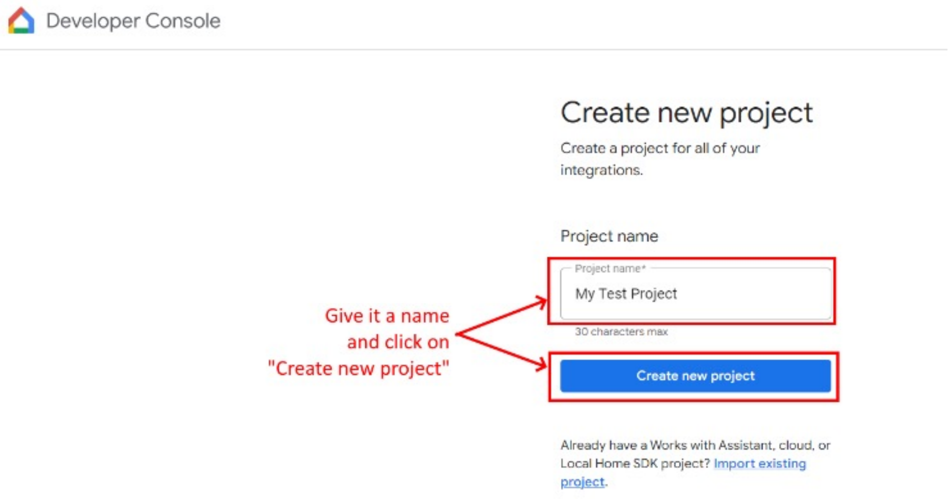
A project is a workspace for all your integration types.

Click on "Create Project"

- On the next page, click on “Create a Project”

- On the following screen, click on “Create project”

Setup Pixel 6 and Nest Hub setup (Part 11)



Developer Console

Create new project

Create a project for all of your integrations.

Project name

Project name*: My Test Project

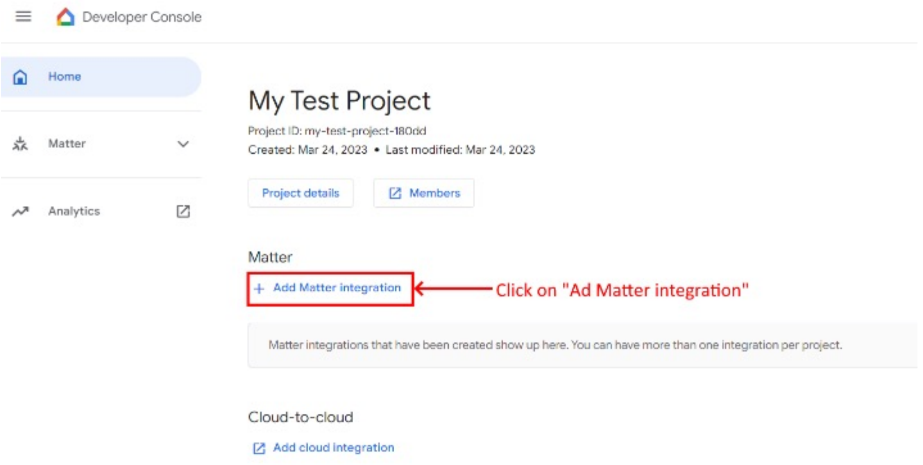
30 characters max

Create new project

Already have a Works with Assistant, cloud, or Local Home SDK project? Import existing project.

Give it a name and click on "Create new project"

- Give your project a name and click on "Create a new project"



My Test Project

Project ID: my-test-project-180dd

Created: Mar 24, 2023 • Last modified: Mar 24, 2023

Project details

Members

Matter

+ Add Matter integration

Click on "Ad Matter integration"

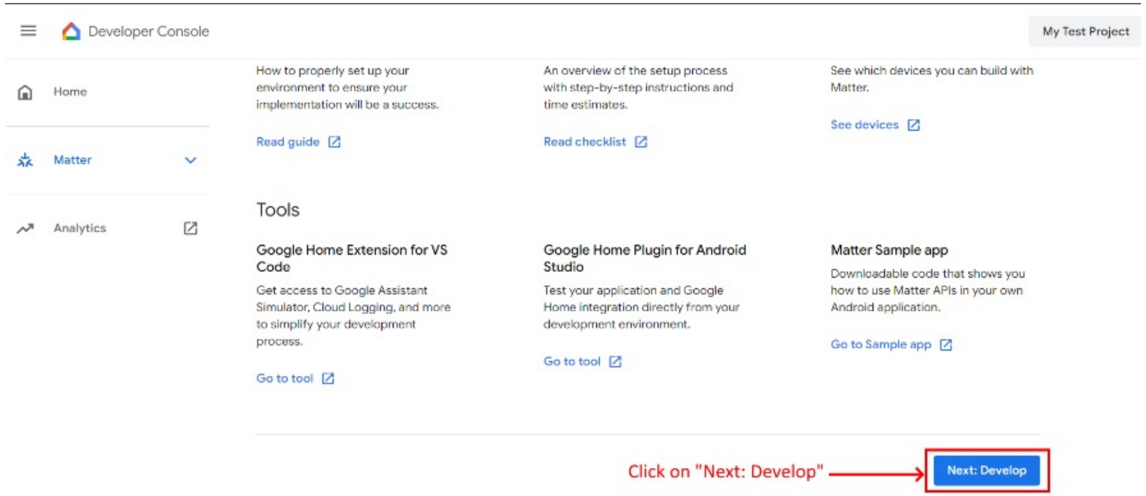
Matter integrations that have been created show up here. You can have more than one integration per project.

Cloud-to-cloud

Add cloud integration

- Now that you have created your project, on the next page click on "Add Matter integration"

Setup Pixel 6 and Nest Hub setup (Part 12)



Developer Console

My Test Project

Home

Matter

Analytics

How to properly set up your environment to ensure your implementation will be a success.

Read guide

An overview of the setup process with step-by-step instructions and time estimates.

Read checklist

See which devices you can build with Matter.

See devices

Tools

Google Home Extension for VS Code

Get access to Google Assistant Simulator, Cloud Logging, and more to simplify your development process.

Go to tool

Google Home Plugin for Android Studio

Test your application and Google Home integration directly from your development environment.

Go to tool

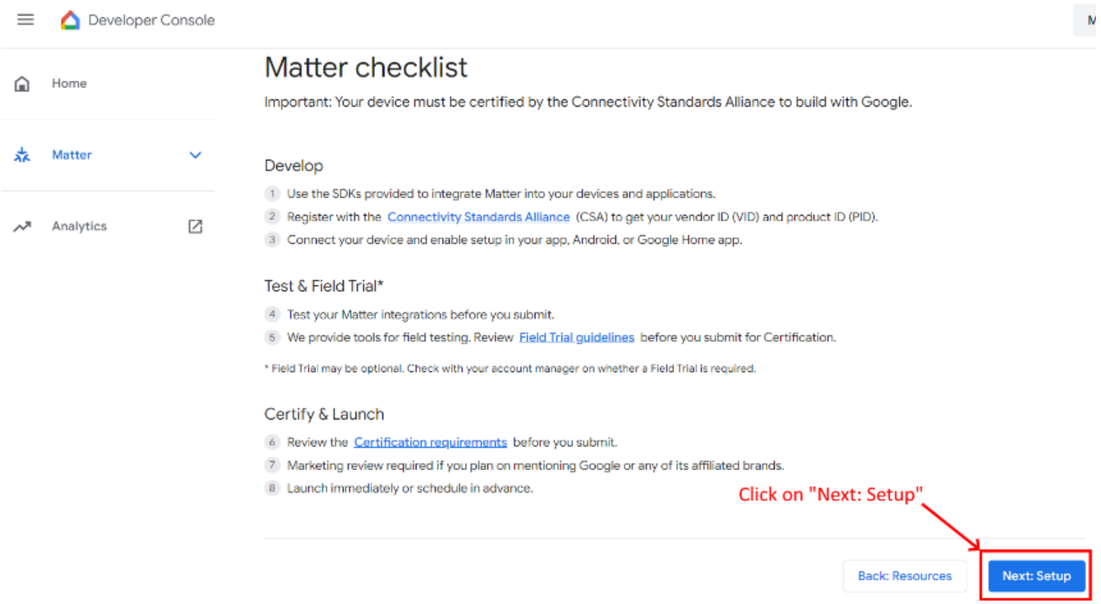
Matter Sample app

Downloadable code that shows you how to use Matter APIs in your own Android application.

Go to Sample app

Click on "Next: Develop"

Next: Develop



Developer Console

M

Home

Matter

Analytics

Matter checklist

Important: Your device must be certified by the Connectivity Standards Alliance to build with Google.

Develop

1. Use the SDKs provided to integrate Matter into your devices and applications.
2. Register with the Connectivity Standards Alliance (CSA) to get your vendor ID (VID) and product ID (PID).
3. Connect your device and enable setup in your app, Android, or Google Home app.

Test & Field Trial*

4. Test your Matter integrations before you submit.
5. We provide tools for field testing. Review Field Trial guidelines before you submit for Certification.

* Field Trial may be optional. Check with your account manager on whether a Field Trial is required.

Certify & Launch

6. Review the Certification requirements before you submit.
7. Marketing review required if you plan on mentioning Google or any of its affiliated brands.
8. Launch immediately or schedule in advance.

Click on "Next: Setup"

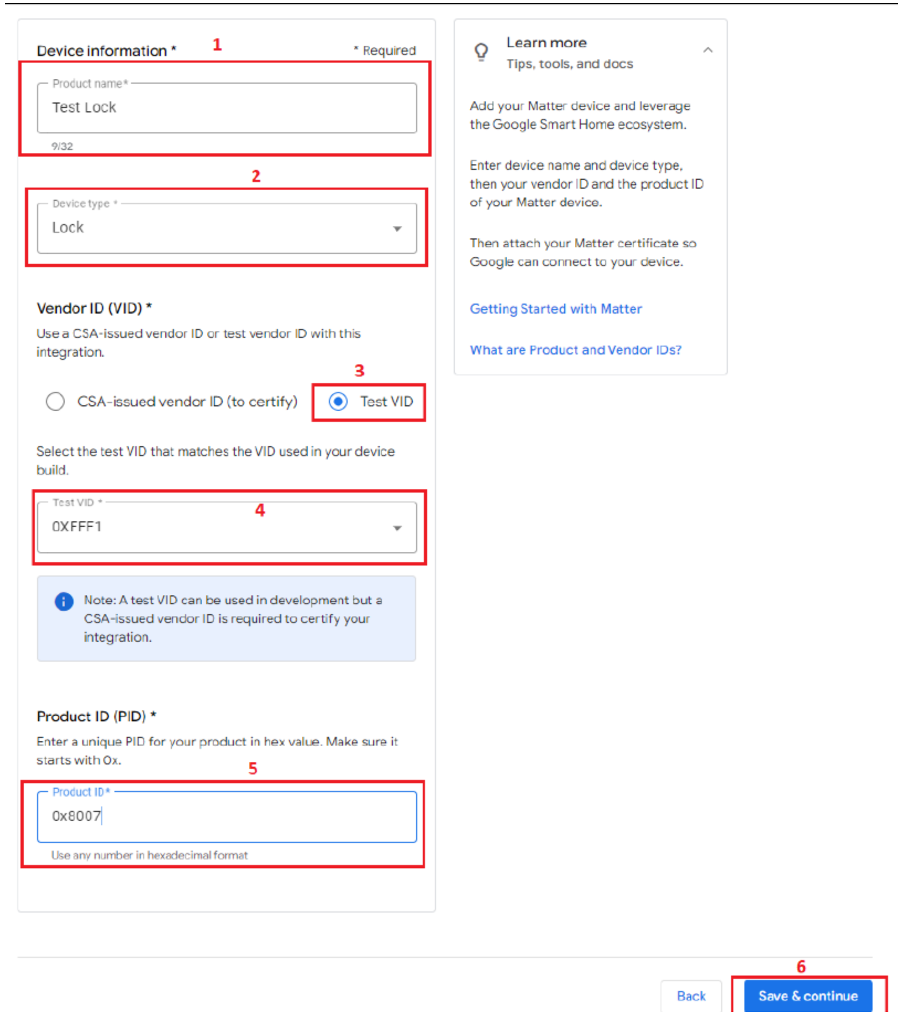
Back: Resources

Next: Setup

- **On the next page click on “Next: Develop”**

- **Now click on “Now click on “Next: Setup”**

Setup Pixel 6 and Nest Hub setup (Part 13)



Device information * 1 * Required

Product name* Test Lock 9/32

2

Device type * Lock

Vendor ID (VID) *

Use a CSA-issued vendor ID or test vendor ID with this integration.

3

○ CSA-issued vendor ID (to certify) ● Test VID

Select the test VID that matches the VID used in your device build.

Test VID * 4 0XFFF1

Note: A test VID can be used in development but a CSA-issued vendor ID is required to certify your integration.

Product ID (PID) *

Enter a unique PID for your product in hex value. Make sure it starts with 0x.

5

Product ID* 0x8007

Use any number in hexadecimal format

Learn more

Tips, tools, and docs

Add your Matter device and leverage the Google Smart Home ecosystem.

Enter device name and device type, then your vendor ID and the product ID of your Matter device.

Then attach your Matter certificate so Google can connect to your device.

Getting Started with Matter

What are Product and Vendor IDs?

6

- **Setup the fields on this page as shown below:**
  - Product name: Test Lock
  - Device type: Lock
  - Vendor ID (VID) Test VID
  - Test VID: 0xFFF1
  - Product ID (PID) 0x8007

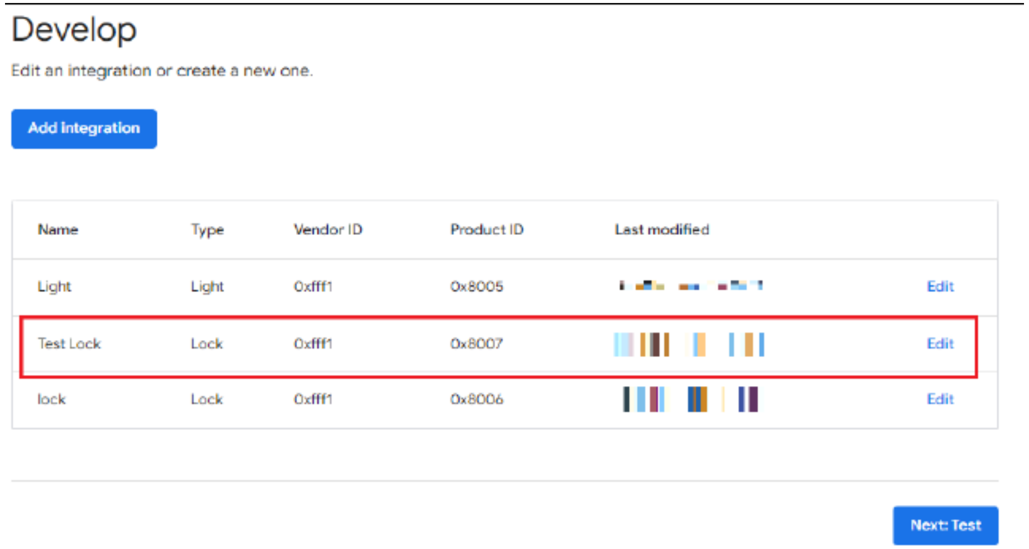
  As shown in these screens

  Product ID options for matter devices are as follows:

  Light-Switch: 0x8004
  Light: 0x8005
  Lock: 0x8007
  Thermostat: 0x800E
  Window Covering: 0x8010

- **After this, click on “Save & Continue”**

Setup Pixel 6 and Nest Hub setup (Part 14)



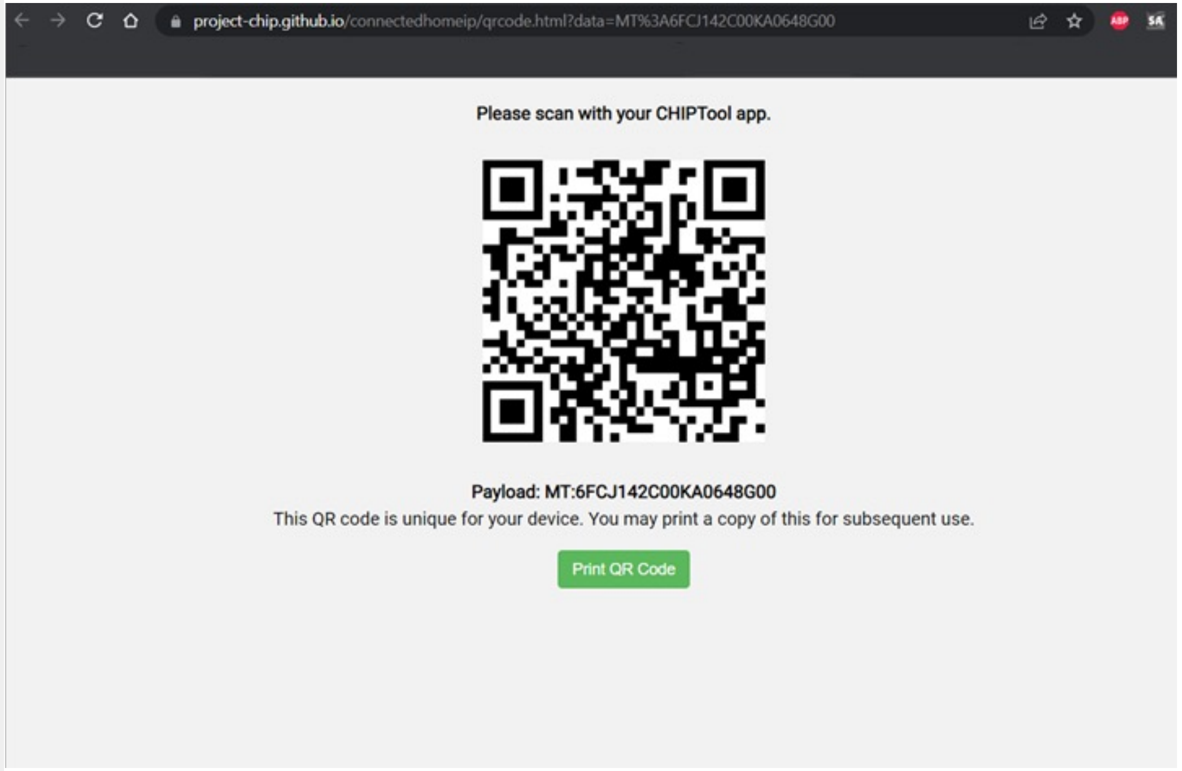
Develop

Edit an integration or create a new one.

Name	Type	Vendor ID	Product ID	Last modified	
Light	Light	0xfff1	0x8005	[illegible]	Edit
Test Lock	Lock	0xfff1	0x8007	[illegible]	Edit
lock	Lock	0xfff1	0x8006	[illegible]	Edit

- **You shall now see a Matter integration for device type lock in your console as shown here**

Setup Pixel 6 and Nest Hub setup (Part 15)



project-chip.github.io/connectedhomeip/qrcode.html?data=MT%3A6FCJ142C00KA0648G00

Please scan with your CHIPTool app.

Payload: MT:6FCJ142C00KA0648G00

This QR code is unique for your device. You may print a copy of this for subsequent use.

- **Now that you have completed setting up the following:**
  - Your home in the Google home app in your Pixel 6
  - A project in your Google developer console
  - A matter integration for the lock device type
- **Having finished the above, the only step left to have your setup ready is to open a QR code webpage for the smartlock device type in your PC**
- **This QR code webpage will be used to commission your smartlock device**
- **To open it, simply open the following URL:**

  https://project-chip.github.io/connectedhomeip/qrcode.html?data=MT%3A6FCJ142C00KA0648G00

(QR Code URL can also be obtained from device logs)

works with
BY SILICON LABS

Setting Up Raspberry Pi for Matter over Wi-Fi demo

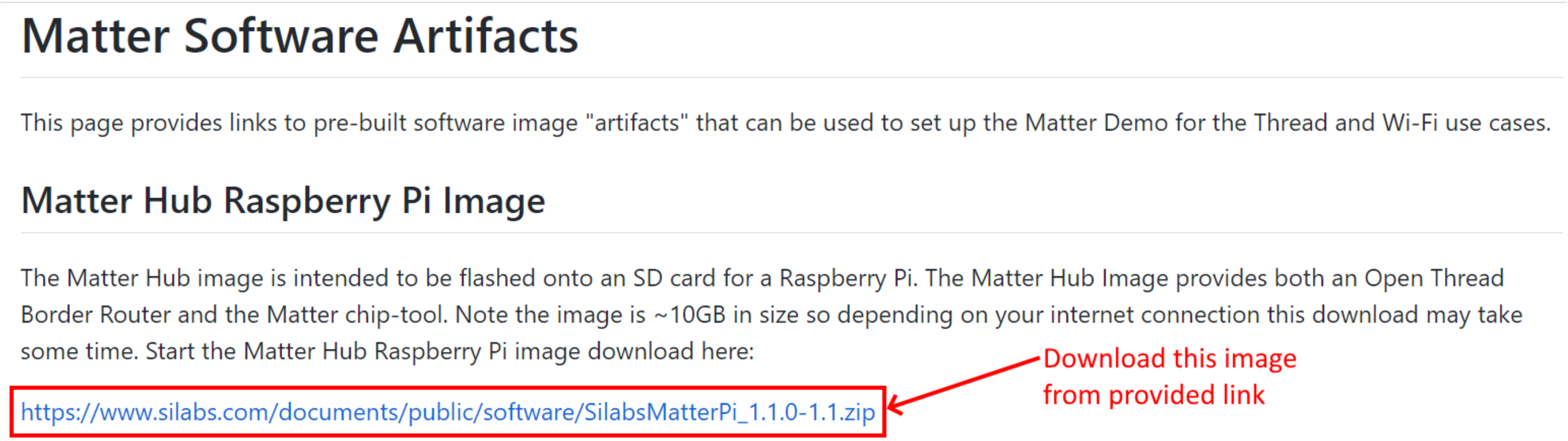
Set up Raspberry Pi 4 to run Chip-Tool in it (Part 1)

- The Raspberry Pi 4 will be used to run the Chip-Tool which will allow the following to be done:
  - Commissioning of the SiWx917 to the Wi-Fi network to be used
  - Controlling via matter over Wi-Fi the lock controlled by the SiWx917 development board

- In order to set up your Raspberry Pi 4, you will need to download the Raspberry Pi Image available in the following link:

  https://www.silabs.com/documents/public/software/SilabsMatterPi_2.0.0-1.1.zip

  - A screenshot showing the file to be downloaded is shown below:



Matter Software Artifacts

This page provides links to pre-built software image "artifacts" that can be used to set up the Matter Demo for the Thread and Wi-Fi use cases.

Matter Hub Raspberry Pi Image

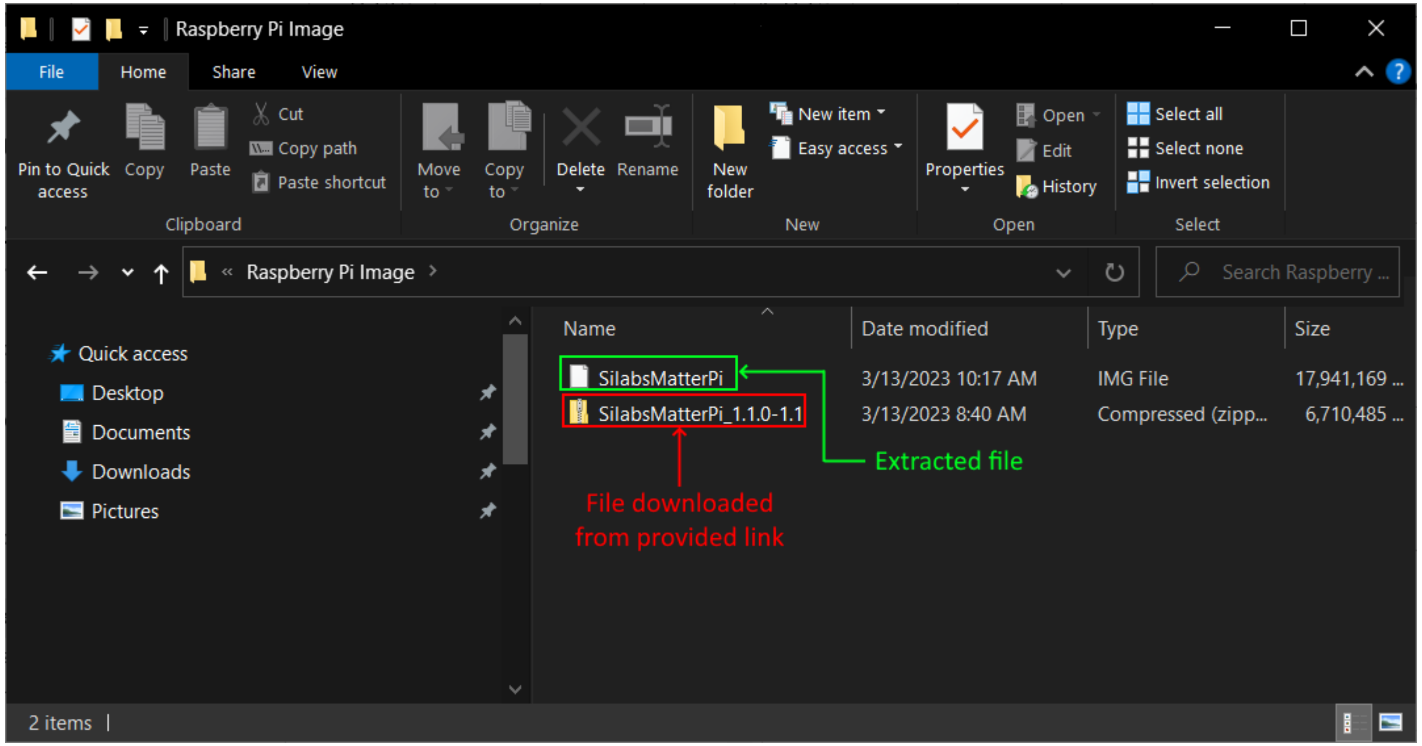
The Matter Hub image is intended to be flashed onto an SD card for a Raspberry Pi. The Matter Hub Image provides both an Open Thread Border Router and the Matter chip-tool. Note the image is ~10GB in size so depending on your internet connection this download may take some time. Start the Matter Hub Raspberry Pi image download here:

https://www.silabs.com/documents/public/software/SilabsMatterPi_1.1.0-1.1.zip

Download this image
from provided link

Set up Raspberry Pi 4 to run Chip-Tool in it (Part 2)

- Once you have downloaded the Raspberry Pi image from the link provided by the previous slide, uncompress it in your PC as shown below:
  - File in Red is the one downloaded from site
  - File highlighted in green is the uncompressed
  - file obtained from it. This is the Raspberry Pi
  - image we will use.



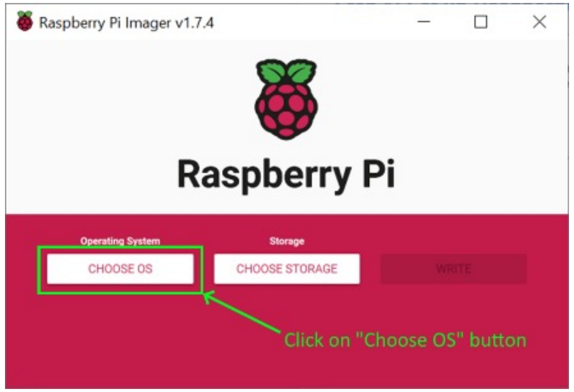
Name	Date modified	Type	Size
SilabsMatterPi	3/13/2023 10:17 AM	IMG File	17,941,169 ...
SilabsMatterPi_1.1.0-1.1	3/13/2023 8:40 AM	Compressed (zipp...	6,710,485 ...

Extracted file

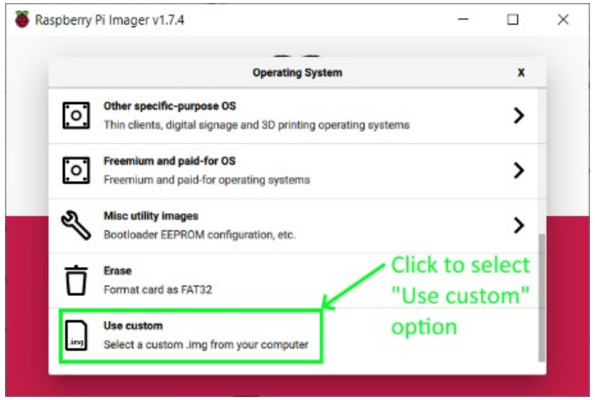
File downloaded from provided link

Set up Raspberry Pi 4 to run Chip-Tool in it (Part 3)

- Run the Raspberry Pi Imaging Tool, and, on it click on the “Choose OS” button and then select the “Use custom” option, as shown below:

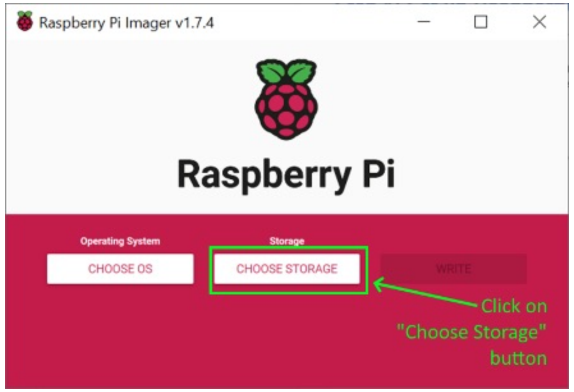


Click on "Choose OS" button

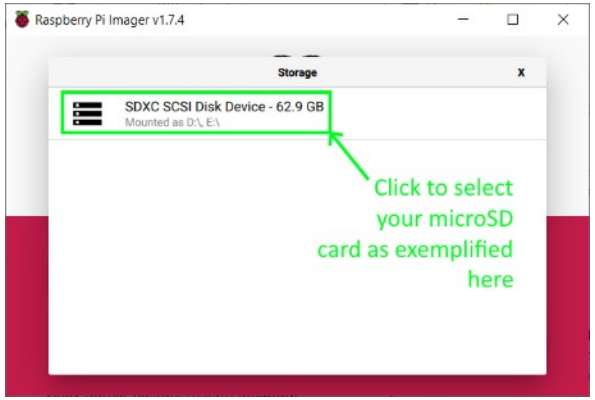

Raspberry Pi Imager v1.7.4
Operating System
x
Other specific-purpose OS
Thin clients, digital signage and 3D printing operating systems
Freemium and paid-for OS
Freemium and paid-for operating systems
Misc utility images
Bootloader EEPROM configuration, etc.
Erase
Format card as FAT32
Use custom
Select a custom .img from your computer

Click to select "Use custom" option

- **After selecting “Use custom", click on “Choose Storage” and then pick your microSD card, as exemplified below**



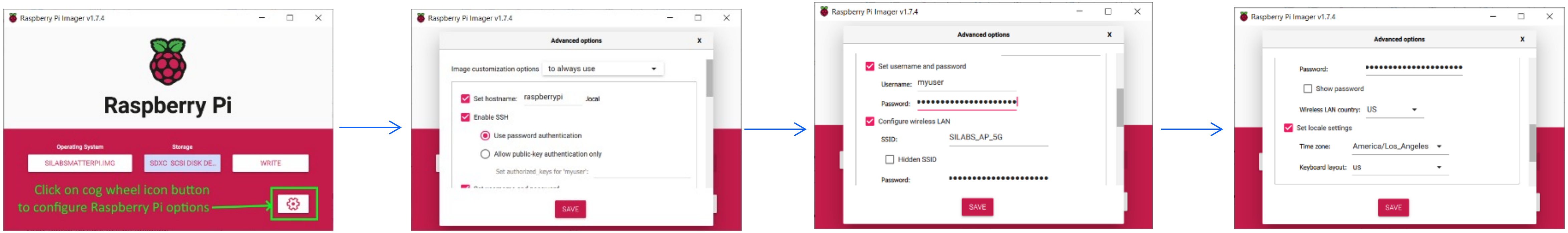
Click on "Choose Storage" button



Click to select your microSD card as exemplified here

Set up Raspberry Pi 4 to run Chip-Tool in it (Part 4)

- Click on cog wheel icon to select desired Raspberry Pi options, including username, password and SSID and password of AP that you want your Raspberry Pi to associate to:



Raspberry Pi Imager v1.7.4

Raspberry Pi

Operating System: SILABSMATTERPI.IMG

Storage: SDXC SCSI DISK DE...

WRITE

Click on cog wheel icon button to configure Raspberry Pi options

Advanced options

Image customization options: to always use

☑ Set hostname: raspberrypi .local

☑ Enable SSH

● Use password authentication

○ Allow public-key authentication only

Set authorized_keys for 'myuser':

SAVE

Advanced options

☑ Set username and password

Username: myuser

Password: ••••••••••••••••••••

☑ Configure wireless LAN

SSID: SILABS_AP_5G

☐ Hidden SSID

Password: ••••••••••••••••••••

SAVE

Advanced options

Password: ••••••••••••••••••••

☐ Show password

Wireless LAN country: US

☑ Set locale settings

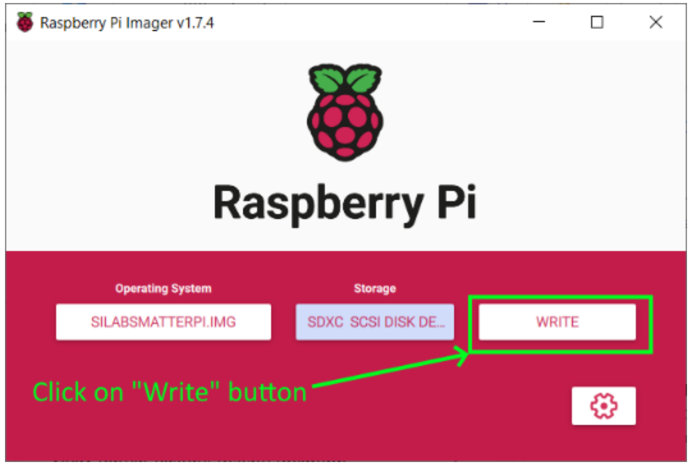
Time zone: America/Los_Angeles

Keyboard layout: us

SAVE

The settings used for this example were as follows:

- hostname = “raspberrypi.local”; Enable SSH; Use password authentication
- Username = “myuser”; Password = “1234567890”
- SSID = “SILABS_AP_5G”; Password = “1234567890”;
- Wireless LAN country = “US”
- **Finally, click on “Write” as shown here. This will write and verify the Raspberry Pi image with your selected settings onto your microSD card**



Raspberry Pi Imager v1.7.4

Raspberry Pi

Operating System: SILABSMATTERPI.IMG

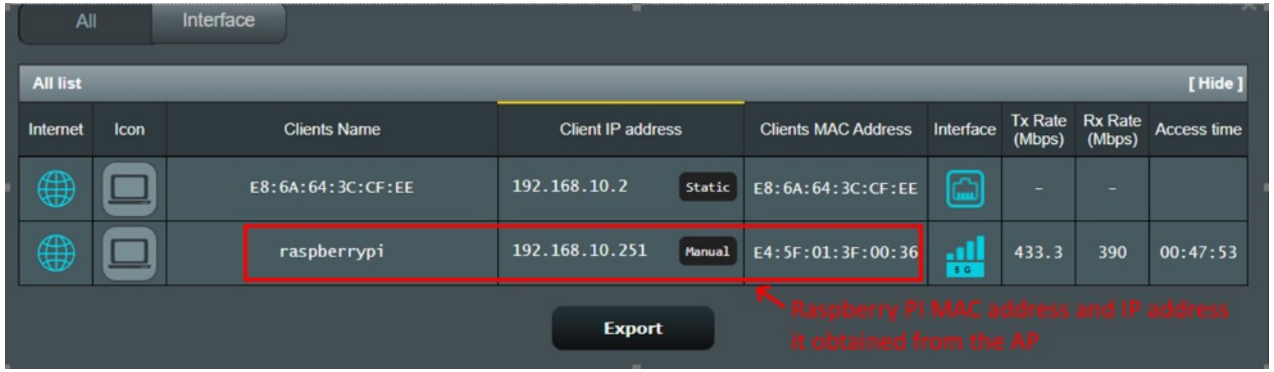
Storage: SDXC SCSI DISK DE...

WRITE

Click on "Write" button

Set up Raspberry Pi 4 to run Chip-Tool in it (Part 5)

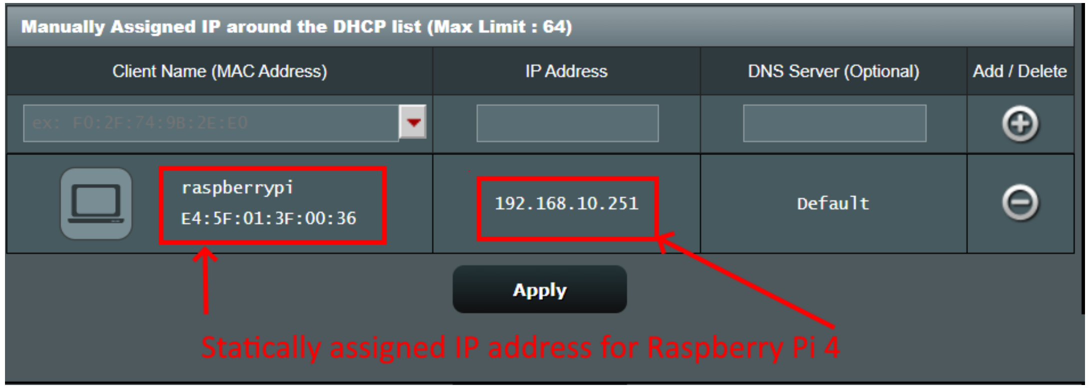
- Now plug the microSD card into your Raspberry Pi 4 board and power on your Raspberry Pi.
- You should see it associate to your access point.
- Verify MAC address and IP address it obtained from your access point and note it down.
- In our example, they were the following:
  - MAC address: E4:5F:01:35:00:36
  - IP Address: 192.168.10.251



Internet	Icon	Clients Name	Client IP address	Clients MAC Address	Interface	Tx Rate (Mbps)	Rx Rate (Mbps)	Access time
		E8:6A:64:3C:CF:EE	192.168.10.2 Static	E8:6A:64:3C:CF:EE		-	-	
		raspberrypi	192.168.10.251 Manual	E4:5F:01:3F:00:36	5 G	433.3	390	00:47:53

Raspberry PI MAC address and IP address it obtained from the AP

- It might be a good idea to set a static DHCP list on your AP in order for your Raspberry Pi to preserve this IP address as shown in the screenshot below:



Manually Assigned IP around the DHCP list (Max Limit : 64)

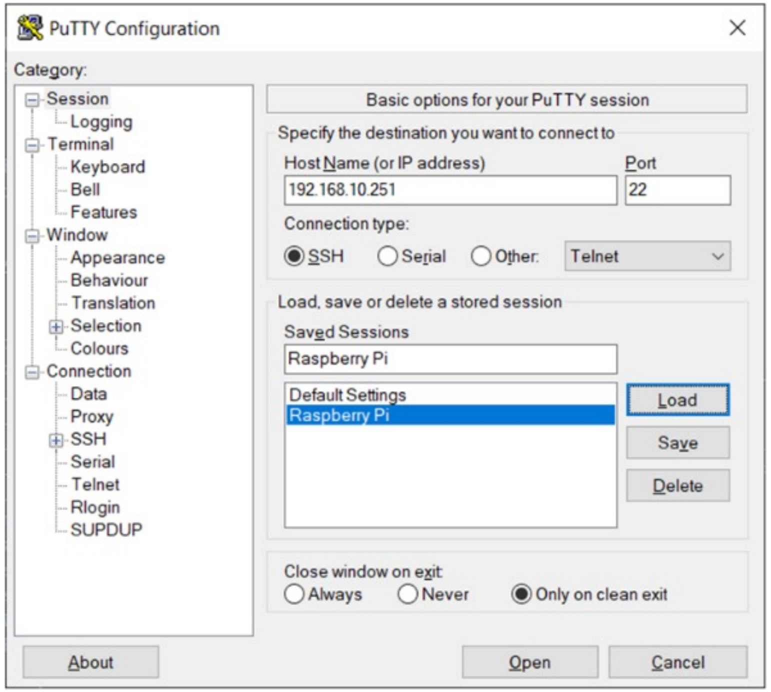
Client Name (MAC Address)	IP Address	DNS Server (Optional)	Add / Delete
ex: F0:2F:74:9B:2E:E0			⊕
raspberrypi E4:5F:01:3F:00:36	192.168.10.251	Default	⊖

Statically assigned IP address for Raspberry Pi 4

Set up Raspberry Pi 4 to run Chip-Tool in it (Part 6)

- On your PC, open putty and use the following settings to connect to the Raspberry Pi:
- Host name (or IP address): IP address of your Raspberry Pi (192.168.10.251 in our example)
- Port: 22
- Connection Type: SSH

- After your settings look similar to this screenshot, click the “Open” button:


PuTTY Configuration

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): 192.168.10.251

Port: 22

Connection type: ● SSH ○ Serial ○ Other: Telnet

Load, save or delete a stored session

Saved Sessions: Raspberry Pi

Default Settings
Raspberry Pi

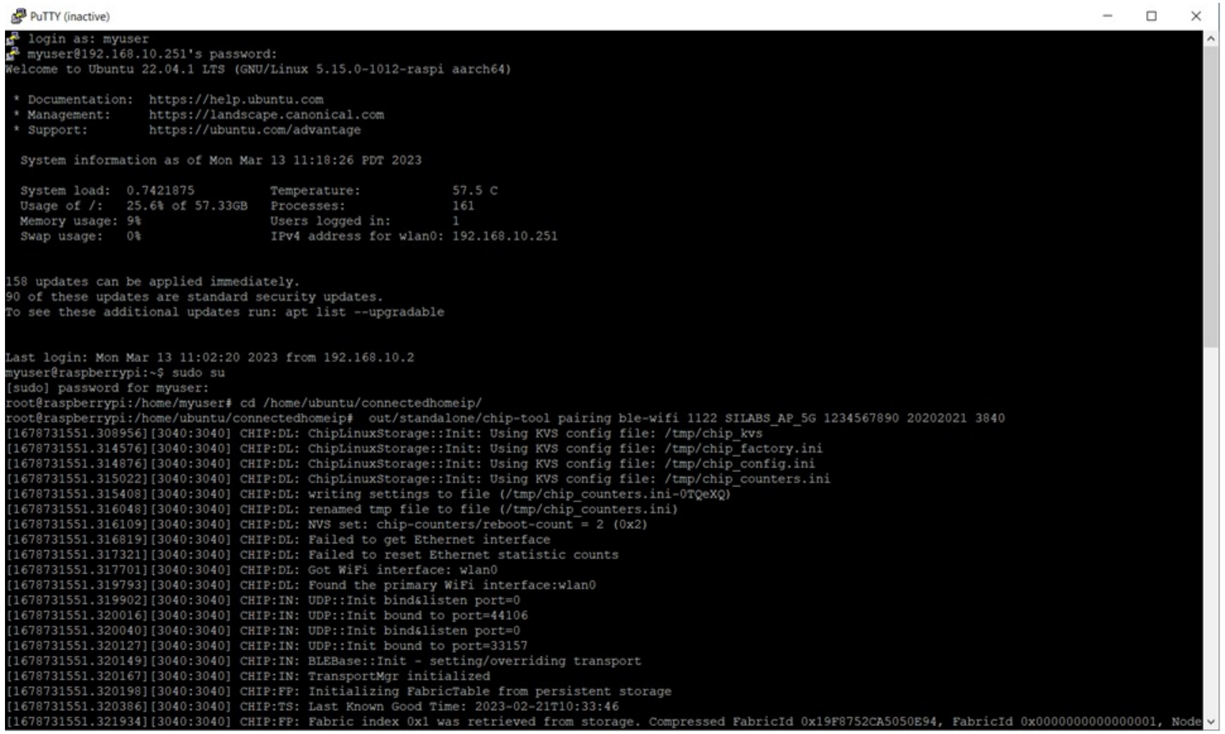
Close window on exit: ○ Always ○ Never ● Only on clean exit

Set up Raspberry Pi 4 to run Chip-Tool in it (Part 7)

- After establishing the connection to your Raspberry Pi, issue the following commands to verify that the Chip-Tool is installed in it and running properly:
- sudo su
- cd /home/ubuntu/connectedhomeip
- out/standalone/chip-tool pairing ble-wifi 1122 SILABS_AP_2G 1234567890 20202021 3840

- You should see something

  similar to the following screenshot:

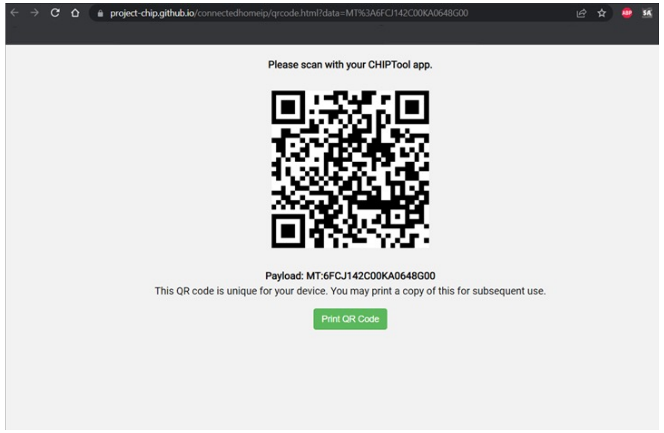


```
 login as: myuser
 myuser@192.168.10.251's password:
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-1012-raspi aarch64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Mon Mar 13 11:18:26 PDT 2023

  System load:  0.7421875         Temperature:           57.5 C
  Usage of /:   25.6% of 57.33GB   Processes:             161
  Memory usage: 9%                Users logged in:       1
  Swap usage:   0%                IPv4 address for wlan0: 192.168.10.251


158 updates can be applied immediately.
90 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable


Last login: Mon Mar 13 11:02:20 2023 from 192.168.10.2
myuser@raspberrypi:~$ sudo su
[sudo] password for myuser:
root@raspberrypi:/home/myuser# cd /home/ubuntu/connectedhomeip/
root@raspberrypi:/home/ubuntu/connectedhomeip#  out/standalone/chip-tool pairing ble-wifi 1122 SILABS_AP_5G 1234567890 20202021 3840
[1678731551.308956][3040:3040] CHIP:DL: ChipLinuxStorage::Init: Using KVS config file: /tmp/chip_kvs
[1678731551.314576][3040:3040] CHIP:DL: ChipLinuxStorage::Init: Using KVS config file: /tmp/chip_factory.ini
[1678731551.314876][3040:3040] CHIP:DL: ChipLinuxStorage::Init: Using KVS config file: /tmp/chip_config.ini
[1678731551.315022][3040:3040] CHIP:DL: ChipLinuxStorage::Init: Using KVS config file: /tmp/chip_counters.ini
[1678731551.315408][3040:3040] CHIP:DL: writing settings to file (/tmp/chip_counters.ini-0TQeXQ)
[1678731551.316048][3040:3040] CHIP:DL: renamed tmp file to file (/tmp/chip_counters.ini)
[1678731551.316109][3040:3040] CHIP:DL: NVS set: chip-counters/reboot-count = 2 (0x2)
[1678731551.316819][3040:3040] CHIP:DL: Failed to get Ethernet interface
[1678731551.317321][3040:3040] CHIP:DL: Failed to reset Ethernet statistic counts
[1678731551.317701][3040:3040] CHIP:DL: Got WiFi interface: wlan0
[1678731551.319793][3040:3040] CHIP:DL: Found the primary WiFi interface:wlan0
[1678731551.319902][3040:3040] CHIP:IN: UDP::Init bind&listen port=0
[1678731551.320016][3040:3040] CHIP:IN: UDP::Init bound to port=44106
[1678731551.320040][3040:3040] CHIP:IN: UDP::Init bind&listen port=0
[1678731551.320127][3040:3040] CHIP:IN: UDP::Init bound to port=33157
[1678731551.320149][3040:3040] CHIP:IN: BLEBase::Init - setting/overriding transport
[1678731551.320167][3040:3040] CHIP:IN: TransportMgr initialized
[1678731551.320198][3040:3040] CHIP:FP: Initializing FabricTable from persistent storage
[1678731551.320386][3040:3040] CHIP:TS: Last Known Good Time: 2023-02-21T10:33:46
[1678731551.321934][3040:3040] CHIP:FP: Fabric index 0x1 was retrieved from storage. Compressed FabricId 0x19F8752CA5050E94, FabricId 0x0000000000000001, Node
```

works with
BY SILICON LABS

Commissioning and Control of Demo from Google Pixel

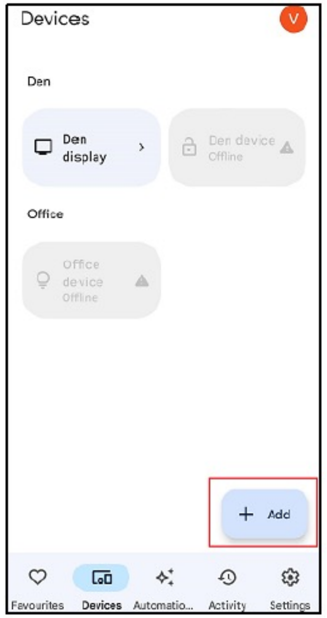
Commission and control SiWx917 matter smartlock with Google Home (Part 1)


project-chip.github.io/connectedhomeip/qrcode.html?data=MT%3A6FCJ142C00KA0648G00
Print QR Code

Please scan with your CHIPTool app.

Payload: MT:6FCJ142C00KA0648G00

This QR code is unique for your device. You may print a copy of this for subsequent use.

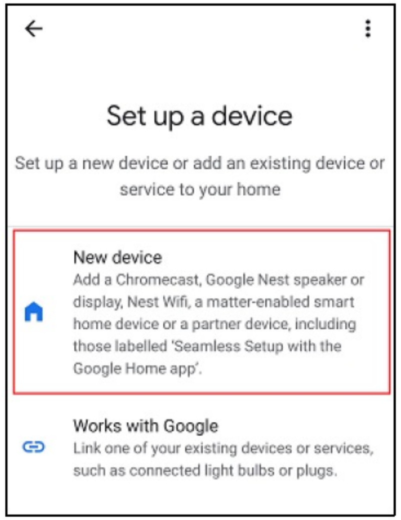


- As said on the previous section. Open the following URL on your PC to open the QR code to commission your SiWx917 smartlock device:

  https://project-chip.github.io/connectedhomeip/qrcode.html?data=MT%3A6FCJ142C00KA0648G00

- Open the Google Home app on your phone and click on “Add” in its devices section.

Commission and control SiWx917 matter smartlock with Google Home (Part 2)

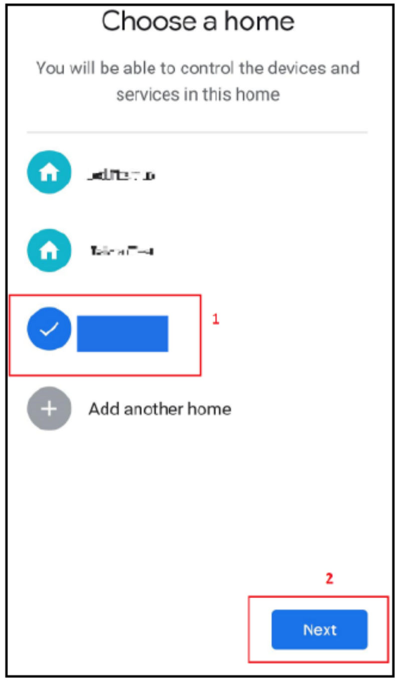


Set up a device

Set up a new device or add an existing device or service to your home

New device
Add a Chromecast, Google Nest speaker or display, Nest Wifi, a matter-enabled smart home device or a partner device, including those labelled 'Seamless Setup with the Google Home app'.

Works with Google
Link one of your existing devices or services, such as connected light bulbs or plugs.



Choose a home

You will be able to control the devices and services in this home

[illegible]

[illegible]

[redacted]

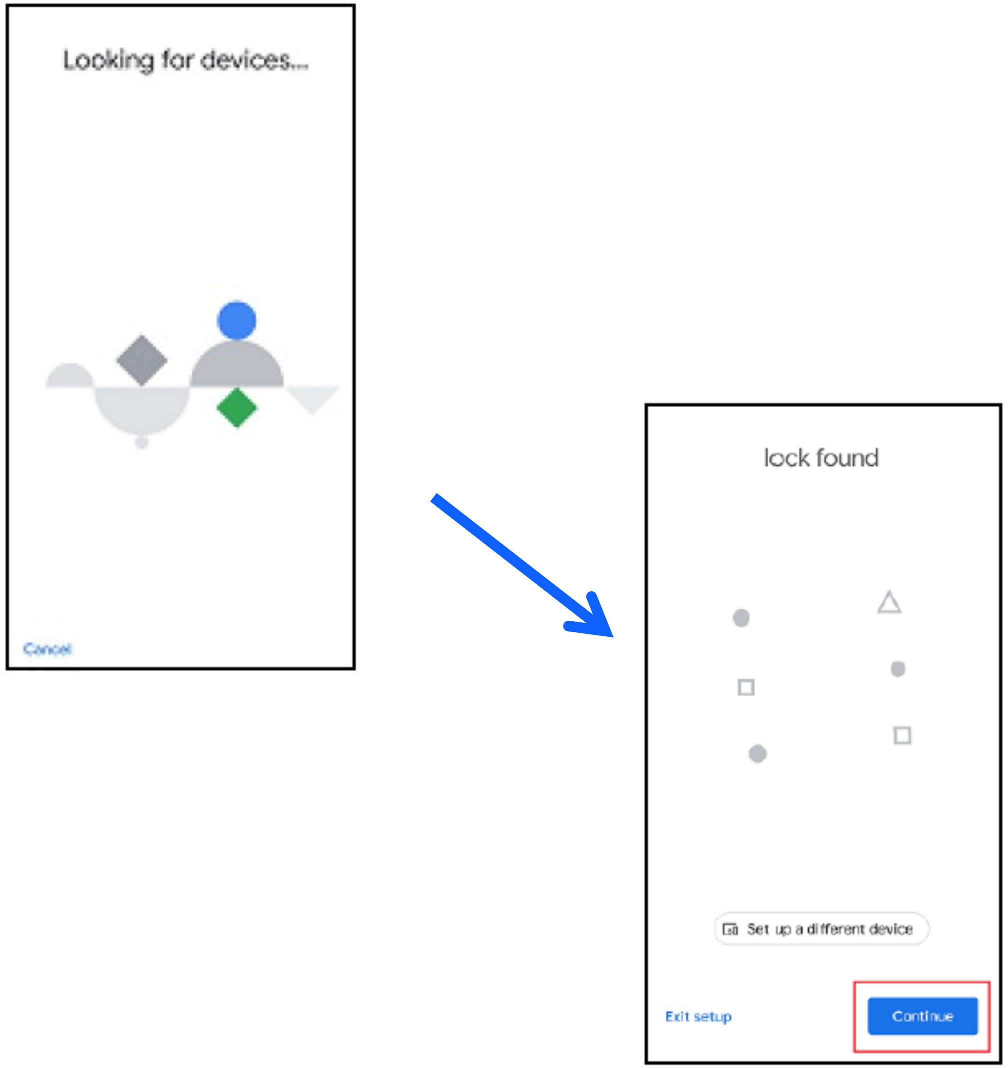
Add another home

Next

- Select “New Device”

- Select the home to which you configured your Nest Hub and click on “Next”

Commission and control SiWx917 matter smartlock with Google Home (Part 3)

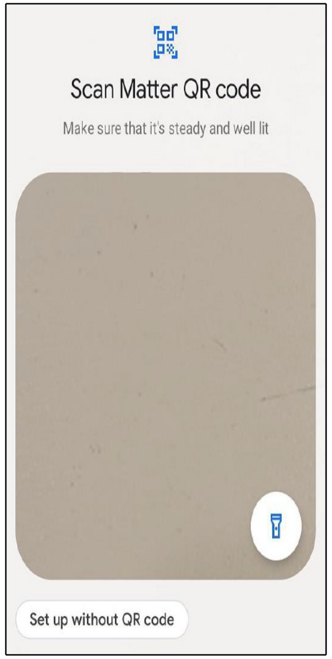


Looking for devices...

lock found

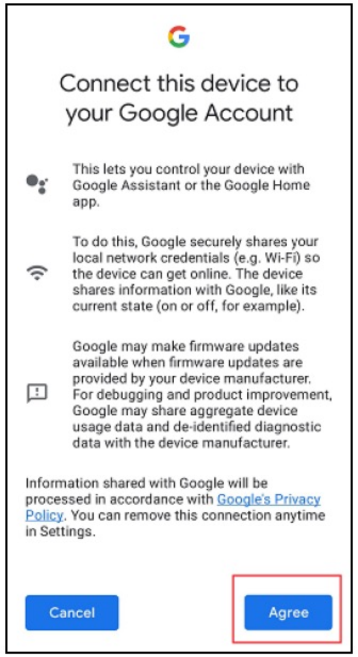
- **The Google Home app will now scan for any available new devices**
- **Once the Google Home app has found the lock it will display a screen with a message stating this. Click on "Continue"**

Commission and control SiWx917 matter smartlock with Google Home (Part 4)



Scan Matter QR code

Make sure that it's steady and well lit



Connect this device to your Google Account

This lets you control your device with Google Assistant or the Google Home app.

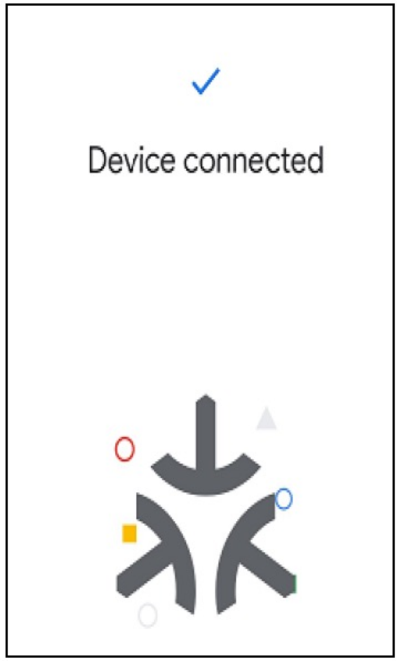
To do this, Google securely shares your local network credentials (e.g. Wi-Fi) so the device can get online. The device shares information with Google, like its current state (on or off, for example).

Google may make firmware updates available when firmware updates are provided by your device manufacturer. For debugging and product improvement, Google may share aggregate device usage data and de-identified diagnostic data with the device manufacturer.

Information shared with Google will be processed in accordance with Google's Privacy Policy. You can remove this connection anytime in Settings.

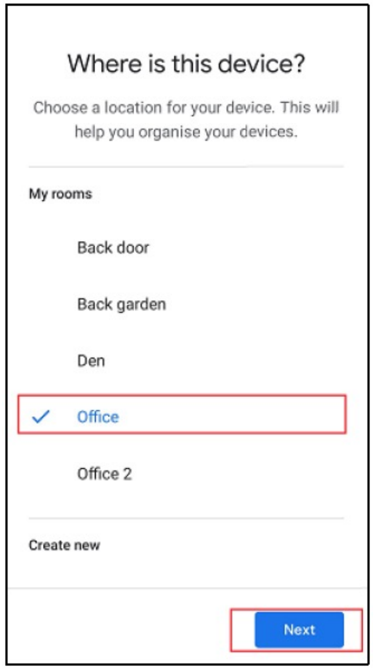
- **Scan the QR code that you opened in your PC's browser**
- **The Google Home app will ask for your permission to connect the device to your account. Click on "Agree" to accept this.**

Commission and control SiWx917 matter smartlock with Google Home (Part 5)



✓

Device connected



Where is this device?

Choose a location for your device. This will help you organise your devices.

My rooms

Back door

Back garden

Den

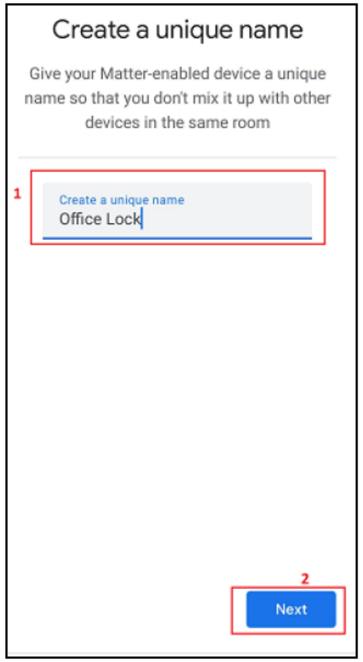
✓ Office

Office 2

Create new

- A message will now be shown stating that your device is now connected.
- Select what room your device is located at and click on “Next”

Commission and control SiWx917 matter smartlock with Google Home (Part 6)



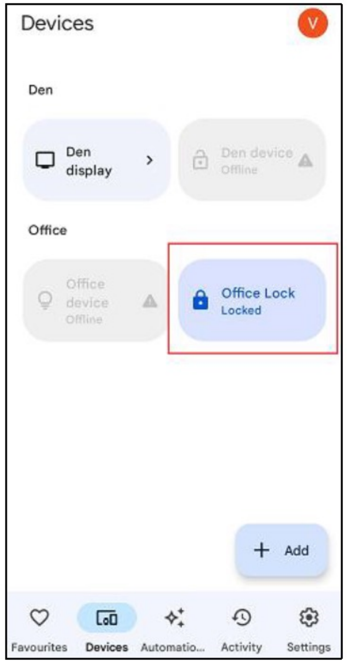
Create a unique name

Give your Matter-enabled device a unique name so that you don't mix it up with other devices in the same room

1

Create a unique name: Office Lock

2



Devices

Den

Den display

Den device Offline

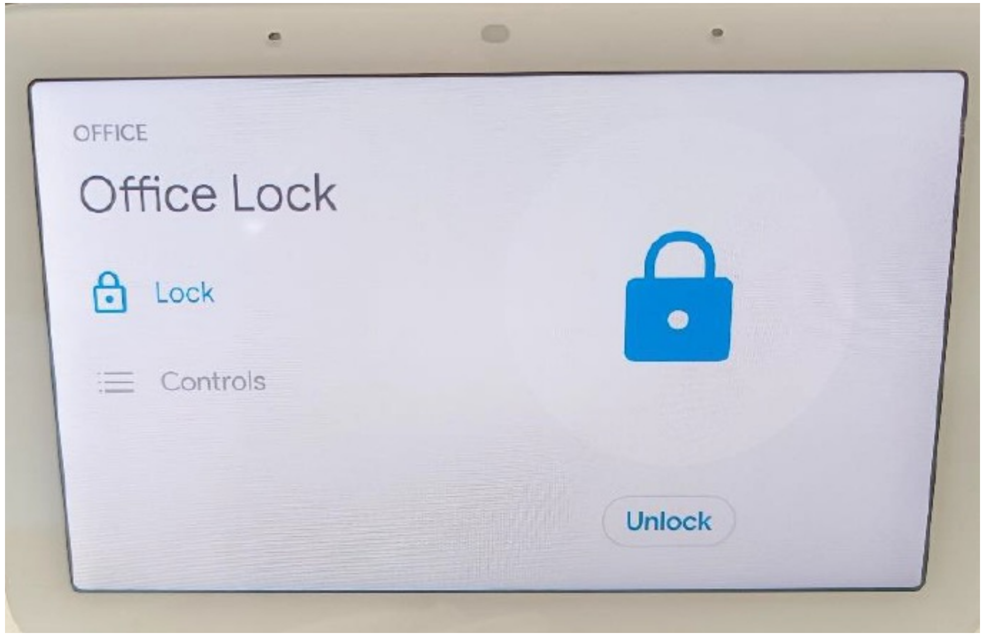
Office

Office device Offline

Office Lock Locked

- **Give your device a unique name and click on "Next"**
- **Your device can now be seen in the Google Home devices tab, where it can now be locked/unlocked**

Commission and control SiWx917 matter smartlock with Google Home (Part 7)

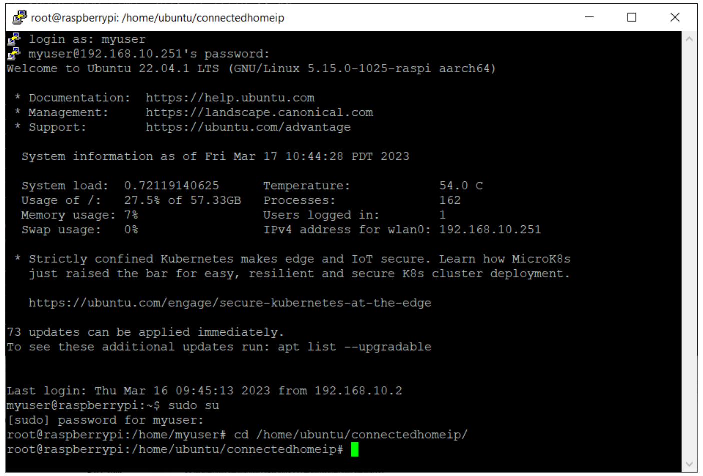


- **It can also be seen in the Google Nest Hub, which will allow it to be controllable through voice commands, for example:**
  - “Hey Google, lock my office lock”
  - “OK Google, unlock my office lock”
  - “OK Google, what is the status of my office lock?”
- **The lock will also be controllable through touch input on the Google Nest Hub.**

works with
BY SILICON LABS

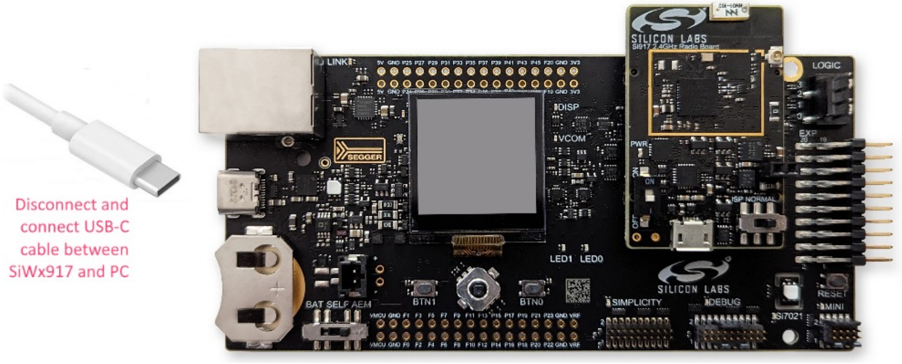
Commissioning and Control of Demo from Raspberry Pi

Commission and control SiWx917 matter smartlock with Raspberry Pi (Part 1)



root@raspberrypi: /home/ubuntu/connectedhomeip

```
 login as: myuser
 myuser@192.168.10.251's password:
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-1025-raspi aarch64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Fri Mar 17 10:44:28 PDT 2023

  System load:  0.72119140625      Temperature:            54.0 C
  Usage of /:   27.5% of 57.33GB   Processes:              162
  Memory usage: 7%                 Users logged in:        1
  Swap usage:   0%                 IPv4 address for wlan0: 192.168.10.251

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

73 updates can be applied immediately.
To see these additional updates run: apt list --upgradable


Last login: Thu Mar 16 09:45:13 2023 from 192.168.10.2
myuser@raspberrypi:~$ sudo su
[sudo] password for myuser:
root@raspberrypi:/home/myuser# cd /home/ubuntu/connectedhomeip/
root@raspberrypi:/home/ubuntu/connectedhomeip#
```

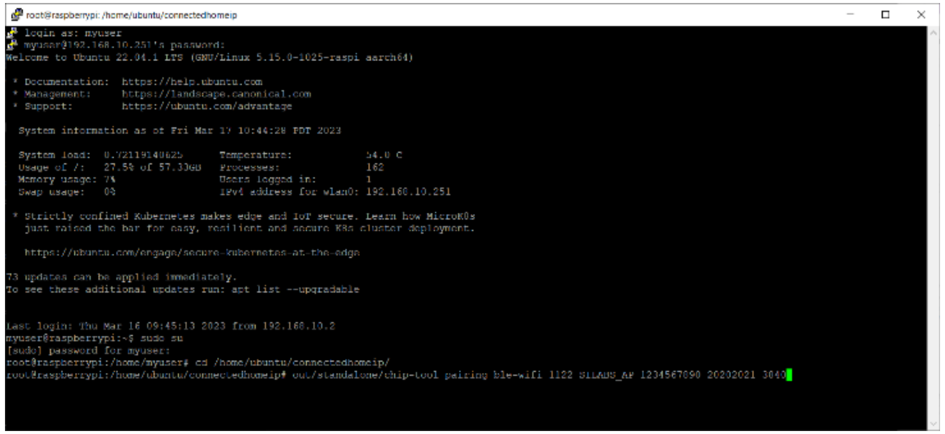


Disconnect and connect USB-C cable between SiWx917 and PC

- **Connect to your Raspberry Pi and issue the following commands**
- sudo su
- cd /home/ubuntu/connectedhomeip/

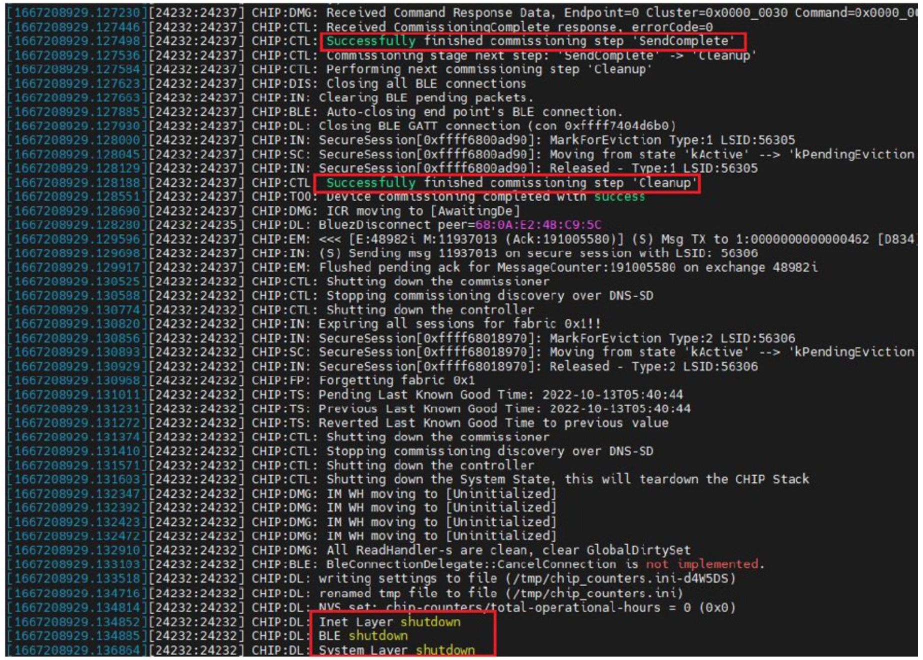
- **Disconnect your SiWx917 setup from your PC and connect it again.**

Commission and control SiWx917 matter smartlock with Raspberry Pi (Part 2)



root@raspberrypi: /home/ubuntu/connectedhomeip

```
login as: myuser
myuser@192.168.10.251's password:
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-1025-raspi aarch64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Fri Mar 17 10:44:28 PDT 2023

  System load:  0.72119140625      Temperature:           54.0 C
  Usage of /:   27.5% of 57.33GB   Processes:             162
  Memory usage: 7%                 Users logged in:       1
  Swap usage:   0%                 IPv4 address for wlan0: 192.168.10.251

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

73 updates can be applied immediately.
To see these additional updates run: apt list --upgradable


Last login: Thu Mar 16 09:45:13 2023 from 192.168.10.2
myuser@raspberrypi:~$ sudo su
[sudo] password for myuser:
root@raspberrypi:/home/myuser# cd /home/ubuntu/connectedhomeip/
root@raspberrypi:/home/ubuntu/connectedhomeip# out/standalone/chip-tool pairing ble-wifi 1122 SILABS_AP 1234567890 20202021 3840
```



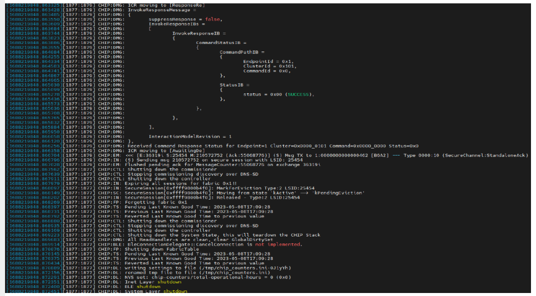
```
[1667208929.127230][24232:24237] CHIP:DMG: Received Command Response Data, Endpoint=0 Cluster=0x0000_0030 Command=0x0000_0
[1667208929.127446][24232:24237] CHIP:CTL: Received CommissioningComplete response, errorCode=0
[1667208929.127498][24232:24237] CHIP:CTL: Successfully finished commissioning step 'SendComplete'
[1667208929.127536][24232:24237] CHIP:CTL: Commissioning stage next step: 'SendComplete' -> 'Cleanup'
[1667208929.127584][24232:24237] CHIP:CTL: Performing next commissioning step 'Cleanup'
[1667208929.127623][24232:24237] CHIP:DIS: Closing all BLE connections
[1667208929.127663][24232:24237] CHIP:IN: Clearing BLE pending packets.
[1667208929.127885][24232:24237] CHIP:BLE: Auto-closing end point's BLE connection.
[1667208929.127930][24232:24237] CHIP:DL: Closing BLE GATT connection (con 0xffff7404d6b0)
[1667208929.128000][24232:24237] CHIP:IN: SecureSession[0xffff6800ad90]: MarkForEviction Type:1 LSID:56305
[1667208929.128045][24232:24237] CHIP:SC: SecureSession[0xffff6800ad90]: Moving from state 'kActive' --> 'kPendingEviction
[1667208929.128129][24232:24237] CHIP:IN: SecureSession[0xffff6800ad90]: Released - Type:1 LSID:56305
[1667208929.128188][24232:24237] CHIP:CTL  Successfully finished commissioning step 'Cleanup'
[1667208929.128551][24232:24237] CHIP:TOO: Device commissioning completed with success
[1667208929.128690][24232:24237] CHIP:DMG: ICR moving to [AwaitingDe]
[1667208929.128280][24232:24235] CHIP:DL: BluezDisconnect peer=68:0A:E2:48:C9:5C
[1667208929.129596][24232:24237] CHIP:EM: <<< [E:48982i M:11937013 (Ack:191005580)] (S) Msg TX to 1:0000000000000462 [D834
[1667208929.129698][24232:24237] CHIP:IN: (S) Sending msg 11937013 on secure session with LSID: 56306
[1667208929.129917][24232:24237] CHIP:EM: Flushed pending ack for MessageCounter:191005580 on exchange 48982i
[1667208929.130525][24232:24232] CHIP:CTL: Shutting down the commissioner
[1667208929.130588][24232:24232] CHIP:CTL: Stopping commissioning discovery over DNS-SD
[1667208929.130774][24232:24232] CHIP:CTL: Shutting down the controller
[1667208929.130820][24232:24232] CHIP:IN: Expiring all sessions for fabric 0x1!!
[1667208929.130856][24232:24232] CHIP:IN: SecureSession[0xffff68018970]: MarkForEviction Type:2 LSID:56306
[1667208929.130893][24232:24232] CHIP:SC: SecureSession[0xffff68018970]: Moving from state 'kActive' --> 'kPendingEviction
[1667208929.130929][24232:24232] CHIP:IN: SecureSession[0xffff68018970]: Released - Type:2 LSID:56306
[1667208929.130968][24232:24232] CHIP:FP: Forgetting fabric 0x1
[1667208929.131011][24232:24232] CHIP:TS: Pending Last Known Good Time: 2022-10-13T05:40:44
[1667208929.131231][24232:24232] CHIP:TS: Previous Last Known Good Time: 2022-10-13T05:40:44
[1667208929.131272][24232:24232] CHIP:TS: Reverted Last Known Good Time to previous value
[1667208929.131374][24232:24232] CHIP:CTL: Shutting down the commissioner
[1667208929.131410][24232:24232] CHIP:CTL: Stopping commissioning discovery over DNS-SD
[1667208929.131571][24232:24232] CHIP:CTL: Shutting down the controller
[1667208929.131603][24232:24232] CHIP:CTL: Shutting down the System State, this will teardown the CHIP Stack
[1667208929.132347][24232:24232] CHIP:DMG: IM WH moving to [Uninitialized]
[1667208929.132392][24232:24232] CHIP:DMG: IM WH moving to [Uninitialized]
[1667208929.132423][24232:24232] CHIP:DMG: IM WH moving to [Uninitialized]
[1667208929.132472][24232:24232] CHIP:DMG: IM WH moving to [Uninitialized]
[1667208929.132910][24232:24232] CHIP:DMG: All ReadHandler-s are clean, clear GlobalDirtySet
[1667208929.133103][24232:24232] CHIP:BLE: BleConnectionDelegate::CancelConnection is not implemented.
[1667208929.133518][24232:24232] CHIP:DL: writing settings to file (/tmp/chip_counters.ini-d4W5DS)
[1667208929.134716][24232:24232] CHIP:DL: renamed tmp file to file (/tmp/chip_counters.ini)
[1667208929.134814][24232:24232] CHIP:DL: NVS set: chip-counters/total-operational-hours = 0 (0x0)
[1667208929.134852][24232:24232] CHIP:DL: Inet Layer shutdown
[1667208929.134885][24232:24232] CHIP:DL: BLE shutdown
[1667208929.136864][24232:24232] CHIP:DL: System Layer shutdown
```

- **Issue the following command to the Raspberry Pi:**
- out/standalone/chip-tool pairing ble-wifi 1122 $SSID $PWD 20202021 3840
- Where:
  - $SSID is the SSID of the AP
  - $PWD is the password of the AP
- In our setup, for example, our AP's 2.4 GHz SSID and password are:
  - SSID = SILABS_AP
  - Password = 1234567890
- So our command is as follows:

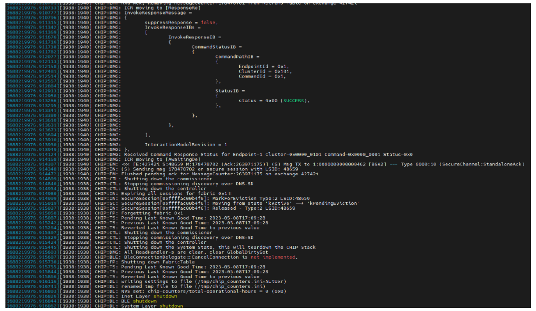
out/standalone/chip-tool pairing ble-wifi 1122 SILABS_AP 1234567890 20202021 3840

- **After issuing the above command, you shall see your SiWx917 setup be commissioned via BLE by the Chip tool to obtain the AP's credentials and then connect to the AP via Wi-Fi. This can be seen on the Raspberry Pi 4 through prints as shown here**

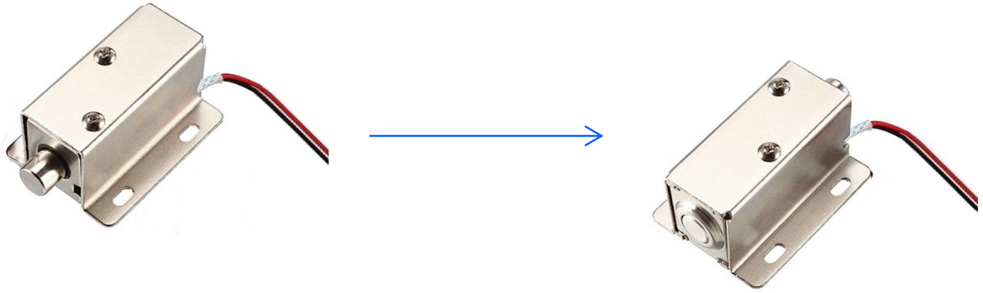
Commission and control SiWx917 matter smartlock with Raspberry Pi (Part 3)



```
[1688219848.863325][1877:1879] CHIP:DMG: ICR moving to [ResponseRe]
[1688219848.863428][1877:1879] CHIP:DMG: InvokeResponseMessage =
[1688219848.863485][1877:1879] CHIP:DMG: {
[1688219848.863550][1877:1879] CHIP:DMG:        suppressResponse = false,
[1688219848.863609][1877:1879] CHIP:DMG:        InvokeResponseIBs =
[1688219848.863684][1877:1879] CHIP:DMG:        [
[1688219848.863744][1877:1879] CHIP:DMG:                InvokeResponseIB =
[1688219848.863823][1877:1879] CHIP:DMG:                {
[1688219848.863886][1877:1879] CHIP:DMG:                        CommandStatusIB =
[1688219848.863955][1877:1879] CHIP:DMG:                        {
[1688219848.864084][1877:1879] CHIP:DMG:                                CommandPathIB =
[1688219848.864255][1877:1879] CHIP:DMG:                                {
[1688219848.864334][1877:1879] CHIP:DMG:                                        EndpointId = 0x1,
[1688219848.864583][1877:1879] CHIP:DMG:                                        ClusterId = 0x101,
[1688219848.864741][1877:1879] CHIP:DMG:                                        CommandId = 0x0,
[1688219848.864867][1877:1879] CHIP:DMG:                                },
[1688219848.864965][1877:1879] CHIP:DMG:
[1688219848.865030][1877:1879] CHIP:DMG:                                StatusIB =
[1688219848.865099][1877:1879] CHIP:DMG:                                {
[1688219848.865278][1877:1879] CHIP:DMG:                                        status = 0x00 (SUCCESS),
[1688219848.865436][1877:1879] CHIP:DMG:                                },
[1688219848.865573][1877:1879] CHIP:DMG:
[1688219848.865636][1877:1879] CHIP:DMG:                        },
[1688219848.865708][1877:1879] CHIP:DMG:
[1688219848.865765][1877:1879] CHIP:DMG:                },
[1688219848.865832][1877:1879] CHIP:DMG:
[1688219848.865884][1877:1879] CHIP:DMG:        ],
[1688219848.865950][1877:1879] CHIP:DMG:
[1688219848.866058][1877:1879] CHIP:DMG:        InteractionModelRevision = 1
[1688219848.866120][1877:1879] CHIP:DMG: },
[1688219848.866256][1877:1879] CHIP:DMG: Received Command Response Status for Endpoint=1 Cluster=0x0000_0101 Command=0x0000_0000 Status=0x0
[1688219848.866358][1877:1879] CHIP:DMG: ICR moving to [AwaitingDe]
[1688219848.866704][1877:1879] CHIP:EM: <<< [E:36319i S:25454 M:210572752 (Ack:55068776)] (S) Msg TX to 1:0000000000000462 [B6A2] --- Type 0000:10 (SecureChannel:StandaloneAck)
[1688219848.866796][1877:1879] CHIP:IN: (S) Sending msg 210572752 on secure session with LSID: 25454
[1688219848.867028][1877:1879] CHIP:EM: Flushed pending ack for MessageCounter:55068776 on exchange 36319i
[1688219848.867562][1877:1877] CHIP:CTL: Shutting down the commissioner
[1688219848.867639][1877:1877] CHIP:CTL: Stopping commissioning discovery over DNS-SD
[1688219848.867911][1877:1877] CHIP:CTL: Shutting down the controller
[1688219848.867979][1877:1877] CHIP:IN: Expiring all sessions for fabric 0x1!!
[1688219848.868097][1877:1877] CHIP:IN: SecureSession[0xffff9800b4f0]: MarkForEviction Type:2 LSID:25454
[1688219848.868149][1877:1877] CHIP:SC: SecureSession[0xffff9800b4f0]: Moving from state 'kActive' --> 'kPendingEviction'
[1688219848.868202][1877:1877] CHIP:IN: SecureSession[0xffff9800b4f0]: Released - Type:2 LSID:25454
[1688219848.868269][1877:1877] CHIP:FP: Forgetting fabric 0x1
[1688219848.868397][1877:1877] CHIP:TS: Pending Last Known Good Time: 2023-05-08T17:09:28
[1688219848.868731][1877:1877] CHIP:TS: Previous Last Known Good Time: 2023-05-08T17:09:28
[1688219848.868792][1877:1877] CHIP:TS: Reverted Last Known Good Time to previous value
[1688219848.868880][1877:1877] CHIP:CTL: Shutting down the commissioner
[1688219848.868935][1877:1877] CHIP:CTL: Stopping commissioning discovery over DNS-SD
[1688219848.869169][1877:1877] CHIP:CTL: Shutting down the controller
[1688219848.869223][1877:1877] CHIP:CTL: Shutting down the System State, this will teardown the CHIP Stack
[1688219848.869683][1877:1877] CHIP:DMG: All ReadHandler-s are clean, clear GlobalDirtySet
[1688219848.869914][1877:1877] CHIP:BLE: BleConnectionDelegate::CancelConnection is not implemented.
[1688219848.870076][1877:1877] CHIP:FP: Shutting down FabricTable
[1688219848.870145][1877:1877] CHIP:TS: Pending Last Known Good Time: 2023-05-08T17:09:28
[1688219848.870375][1877:1877] CHIP:TS: Previous Last Known Good Time: 2023-05-08T17:09:28
[1688219848.870434][1877:1877] CHIP:TS: Reverted Last Known Good Time to previous value
[1688219848.870889][1877:1877] CHIP:DL: writing settings to file (/tmp/chip_counters.ini-0J1yYh)
[1688219848.872156][1877:1877] CHIP:DL: renamed tmp file to file (/tmp/chip_counters.ini)
[1688219848.872291][1877:1877] CHIP:DL: NVS set: chip-counters/total-operational-hours = 0 (0x0)
[1688219848.872351][1877:1877] CHIP:DL: Inet Layer shutdown
[1688219848.872400][1877:1877] CHIP:DL: BLE shutdown
[1688219848.872451][1877:1877] CHIP:DL: System Layer shutdown
```



```
1688219976.910733][1938:1940] CHIP:DMG: ICR moving to [ResponseRe]
1688219976.910777][1938:1940] CHIP:DMG: InvokeResponseMessage =
1688219976.910796][1938:1940] CHIP:DMG: {
1688219976.911315][1938:1940] CHIP:DMG:        suppressResponse = false,
1688219976.911342][1938:1940] CHIP:DMG:        InvokeResponseIBs =
1688219976.911369][1938:1940] CHIP:DMG:        [
1688219976.911676][1938:1940] CHIP:DMG:                InvokeResponseIB =
1688219976.911716][1938:1940] CHIP:DMG:                {
1688219976.911738][1938:1940] CHIP:DMG:                        CommandStatusIB =
1688219976.911782][1938:1940] CHIP:DMG:                        {
1688219976.912077][1938:1940] CHIP:DMG:                                CommandPathIB =
1688219976.912113][1938:1940] CHIP:DMG:                                {
1688219976.912158][1938:1940] CHIP:DMG:                                        EndpointId = 0x1,
1688219976.912481][1938:1940] CHIP:DMG:                                        ClusterId = 0x101,
1688219976.912514][1938:1940] CHIP:DMG:                                        CommandId = 0x1,
1688219976.912557][1938:1940] CHIP:DMG:                                },
1688219976.912884][1938:1940] CHIP:DMG:
1688219976.912913][1938:1940] CHIP:DMG:                                StatusIB =
1688219976.912958][1938:1940] CHIP:DMG:                                {
1688219976.913266][1938:1940] CHIP:DMG:                                        status = 0x00 (SUCCESS),
1688219976.913295][1938:1940] CHIP:DMG:                                },
1688219976.913341][1938:1940] CHIP:DMG:
1688219976.913380][1938:1940] CHIP:DMG:                        },
1688219976.913610][1938:1940] CHIP:DMG:
1688219976.913631][1938:1940] CHIP:DMG:                },
1688219976.913673][1938:1940] CHIP:DMG:
1688219976.913694][1938:1940] CHIP:DMG:        ],
1688219976.913910][1938:1940] CHIP:DMG:
1688219976.913930][1938:1940] CHIP:DMG:        InteractionModelRevision = 1
1688219976.913949][1938:1940] CHIP:DMG: },
1688219976.914124][1938:1940] CHIP:DMG: Received Command Response Status for Endpoint=1 Cluster=0x0000_0101 Command=0x0000_0001 Status=0x0
1688219976.914168][1938:1940] CHIP:DMG: ICR moving to [AwaitingDe]
1688219976.914307][1938:1940] CHIP:EM: <<< [E:42742i S:48659 M:178478702 (Ack:263971175)] (S) Msg TX to 1:0000000000000462 [B6A2] --- Type 0000:10 (SecureChannel:StandaloneAck)
1688219976.914348][1938:1940] CHIP:IN: (S) Sending msg 178478702 on secure session with LSID: 48659
1688219976.914472][1938:1940] CHIP:EM: Flushed pending ack for MessageCounter:263971175 on exchange 42742i
1688219976.914809][1938:1938] CHIP:CTL: Shutting down the commissioner
1688219976.914840][1938:1938] CHIP:CTL: Stopping commissioning discovery over DNS-SD
1688219976.914954][1938:1938] CHIP:CTL: Shutting down the controller
1688219976.914980][1938:1938] CHIP:IN: Expiring all sessions for fabric 0x1!!
1688219976.914999][1938:1938] CHIP:IN: SecureSession[0xffffac00b4f0]: MarkForEviction Type:2 LSID:48659
1688219976.915019][1938:1938] CHIP:SC: SecureSession[0xffffac00b4f0]: Moving from state 'kActive' --> 'kPendingEviction'
1688219976.915037][1938:1938] CHIP:IN: SecureSession[0xffffac00b4f0]: Released - Type:2 LSID:48659
1688219976.915058][1938:1938] CHIP:FP: Forgetting fabric 0x1
1688219976.915087][1938:1938] CHIP:TS: Pending Last Known Good Time: 2023-05-08T17:09:28
1688219976.915242][1938:1938] CHIP:TS: Previous Last Known Good Time: 2023-05-08T17:09:28
1688219976.915264][1938:1938] CHIP:TS: Reverted Last Known Good Time to previous value
1688219976.915307][1938:1938] CHIP:CTL: Shutting down the commissioner
1688219976.915329][1938:1938] CHIP:CTL: Stopping commissioning discovery over DNS-SD
1688219976.915424][1938:1938] CHIP:CTL: Shutting down the controller
1688219976.915445][1938:1938] CHIP:CTL: Shutting down the System State, this will teardown the CHIP Stack
1688219976.915603][1938:1938] CHIP:DMG: All ReadHandler-s are clean, clear GlobalDirtySet
1688219976.915687][1938:1938] CHIP:BLE: BleConnectionDelegate::CancelConnection is not implemented.
1688219976.915730][1938:1938] CHIP:FP: Shutting down FabricTable
1688219976.915755][1938:1938] CHIP:TS: Pending Last Known Good Time: 2023-05-08T17:09:28
1688219976.915844][1938:1938] CHIP:TS: Previous Last Known Good Time: 2023-05-08T17:09:28
1688219976.915866][1938:1938] CHIP:TS: Reverted Last Known Good Time to previous value
1688219976.916116][1938:1938] CHIP:DL: writing settings to file (/tmp/chip_counters.ini-ALtUxr)
1688219976.916741][1938:1938] CHIP:DL: renamed tmp file to file (/tmp/chip_counters.ini)
1688219976.916803][1938:1938] CHIP:DL: NVS set: chip-counters/total-operational-hours = 0 (0x0)
1688219976.916826][1938:1938] CHIP:DL: Inet Layer shutdown
1688219976.916844][1938:1938] CHIP:DL: BLE shutdown
1688219976.916862][1938:1938] CHIP:DL: System Layer shutdown
```

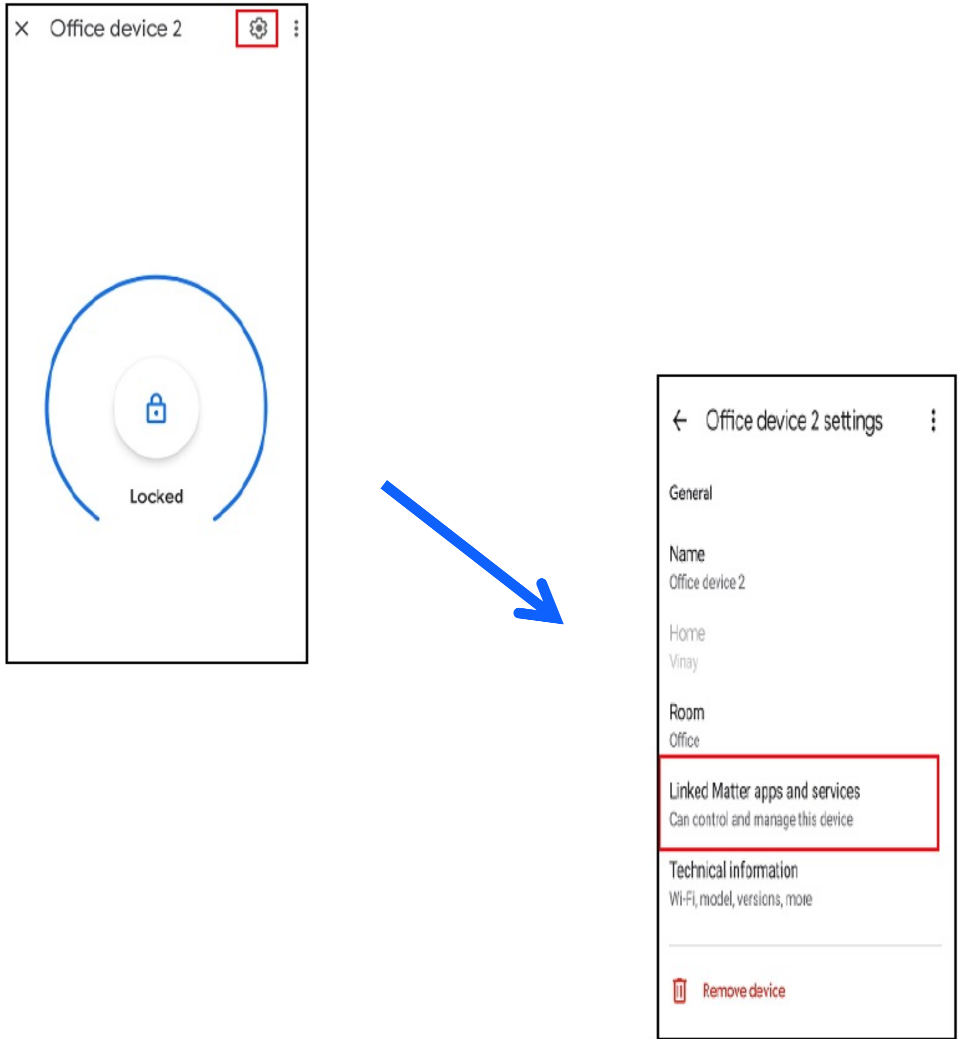


- **The following two commands can be issued to lock and unlock the SiWx917 smartlock**
- To lock:
  - out/standalone/chip-tool doorlock lock-door 1122 1 -timedInteractionTimeoutMs 1000
- To unlock
  - out/standalone/chip-tool doorlock unlock 1122 1
- **You will see responses in the Raspberry Pi 4 terminal as shown to the left.**
- **You will see the doorlock lock and unlock as shown below this.**

works with
BY SILICON LABS

Using Multi-Admin Capabilities

Using Multi-Admin Capabilities (Part 1)



Office device 2

Locked

Office device 2 settings

General

Name
Office device 2

Home
Vinay

Room
Office

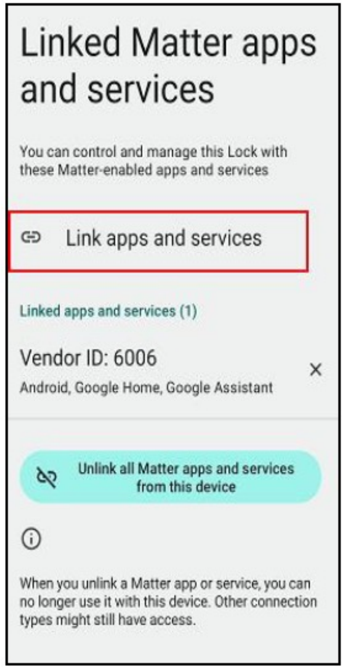
Linked Matter apps and services
Can control and manage this device

Technical information
Wi-Fi, model, versions, more

Remove device

- **As stated earlier, our demo will showcase how to use Matter's Multi-Admin capabilities**
- **We'll show you know how to use this feature with the Google Home and Raspberry Pi 4**
- **First, we'll need to configure the multi-admin setup.**
- **The first step will be to commission the SiWx917 lock to the Google ecosystem as shown previously**
- **After this, acquire the pairing code from the Google Home App to which the device was commissioned.**
- **To do so. Long press on the device name in the Google Home app and select settings (gear icon)**
- **In the next screen, click on "Linked Matter apps and services"**

Using Multi-Admin Capabilities (Part 2)



Linked Matter apps and services

You can control and manage this Lock with these Matter-enabled apps and services

Link apps and services

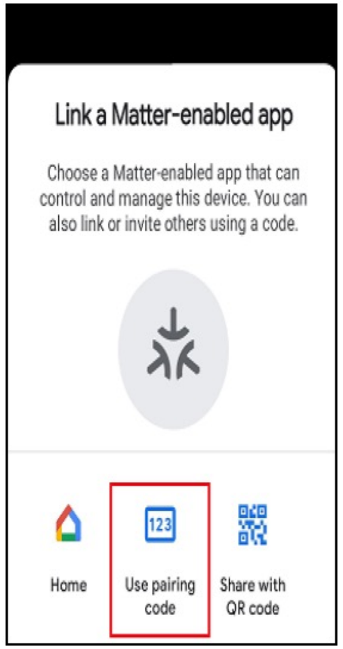
Linked apps and services (1)

Vendor ID: 6006 ×

Android, Google Home, Google Assistant

Unlink all Matter apps and services from this device

When you unlink a Matter app or service, you can no longer use it with this device. Other connection types might still have access.



Link a Matter-enabled app

Choose a Matter-enabled app that can control and manage this device. You can also link or invite others using a code.

Home

Use pairing code

Share with QR code

- **In the next screen, select “Link apps and services”**
- **And follow this by clicking on “Use pairing code”**
- **Finally, in the Raspberry Pi 4 issue the following command:**
  - “out/standalone/ chip-tool pairing code 1122 $CODE”
- **Where $CODE is the 11 digit pairing code acquired from the Google Home app**
- **Once commissioning with Raspberry Pi 4 is successful, lock and unlock commands can be issued from both Google and Raspberry Pi devices.**
- **Multi-admin is also possible in the opposite direction. First commissioning with Raspberry Pi and then with Google Home. We will show you a video of this now.**

works with
BY SILICON LABS

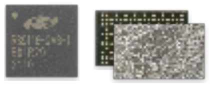
Video of Demo Execution

works with
BY SILICON LABS

Matter over Wi-Fi Smartlock Commissioning, Control and Multi-Admin Demo

Silicon Labs Matter over Wi-Fi Portfolio

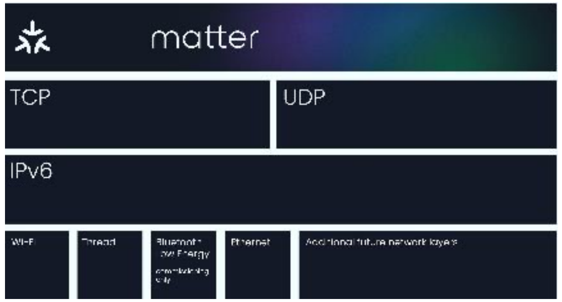
Silicon Labs’ Wi-Fi SoC Portfolio Summary

Features	**WF200** 	**RS9116** 	**SiWx917**
Wi-Fi (2.4 GHz)	**Wi-Fi 4**	**Wi-Fi 4**	**Wi-Fi 6**
BT Low Energy (LE)		✓	✓
BT Classic (Audio)		✓	
Low Power Modes	**PS-POLL**	**PS-POLL, Listen Interval**	**PS-POLL, Listen Interval, TWT**
Wi-Fi Features	**OFDM**	**ODFM**	**OFDM, OFDMA, MU-MIMO**
Wi-Fi WPA3 Security	✓	✓	✓
ARM® Apps MCU (SoC Mode)			✓
ML Accelerator, PSRAM Interface, MCU Security (PSA-L2)			✓
Ultra Low Power		✓	✓
Matter	✓	✓	✓

Silicon Labs Matter Solutions – More Than Just Silicon

HARDWARE

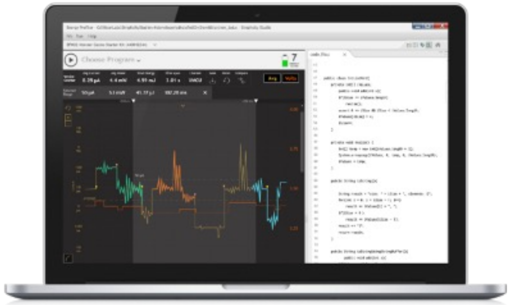
- Field-proven SoCs and modules for Thread and Wi-Fi with Bluetooth
- Robust and reliable wireless foundation for Matter devices



CERTIFICATION

- Support for Wi-Fi and 802.15.4 end product certification
- Participation in all CSA Matter test events
- Matter certification

Simplicity Studio 5



TOOLS

- Advanced development hardware, reference designs, and tools
- Simplifies development and speeds time-to-market

SOFTWARE

- Support for all Matter devices including border routers, and bridges
- The largest semiconductor contributor to Matter GitHub

works with
BY SILICON LABS

Q&A

works with
BY SILICON LABS

Thank you!