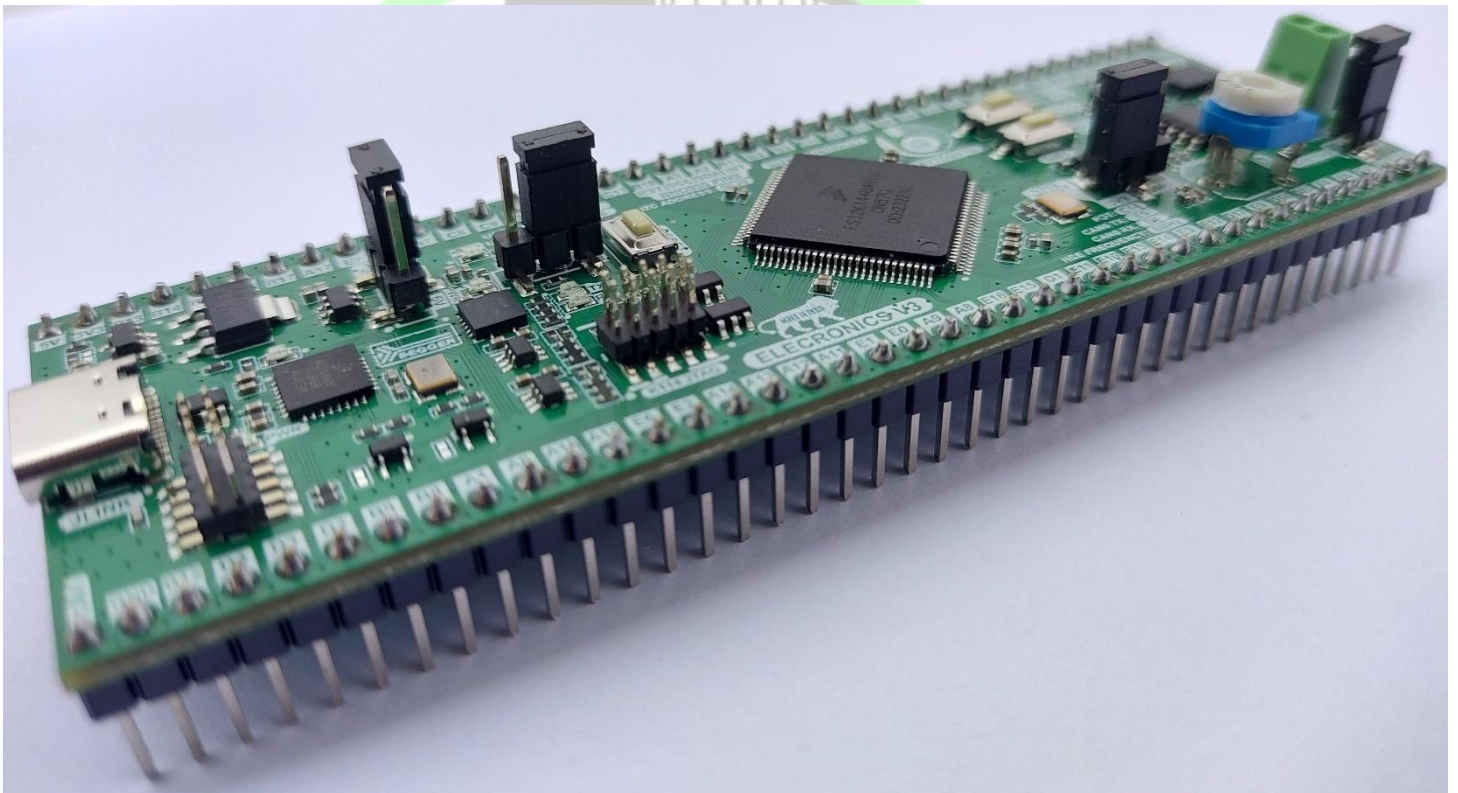


Getting Started Manual of ElectronicsV3 Board



~~ Arduino of Automotive World

Reinventing Tech With Bytes

Table of Contents

Contents

Getting Started Manual of ElectronicsV3 Board	1
Table of Contents	2
Getting to Know the Hardware	3
Getting to Know the Software	5
How To do the Debugging and Building in S32 Design Studio	7
Resources for ElectronicsV3	9
Reference Documents	11
Revisions	12



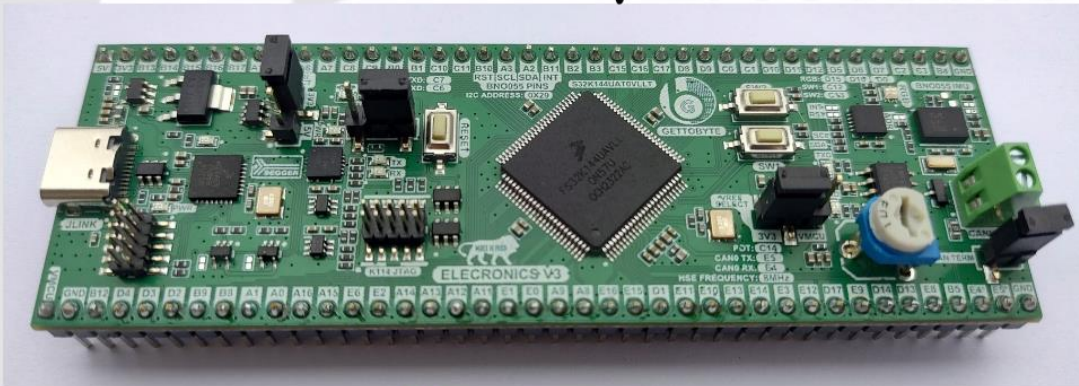
GettobYTE Technologies

Reinventing Tech With Bytes

Getting to Know the Hardware

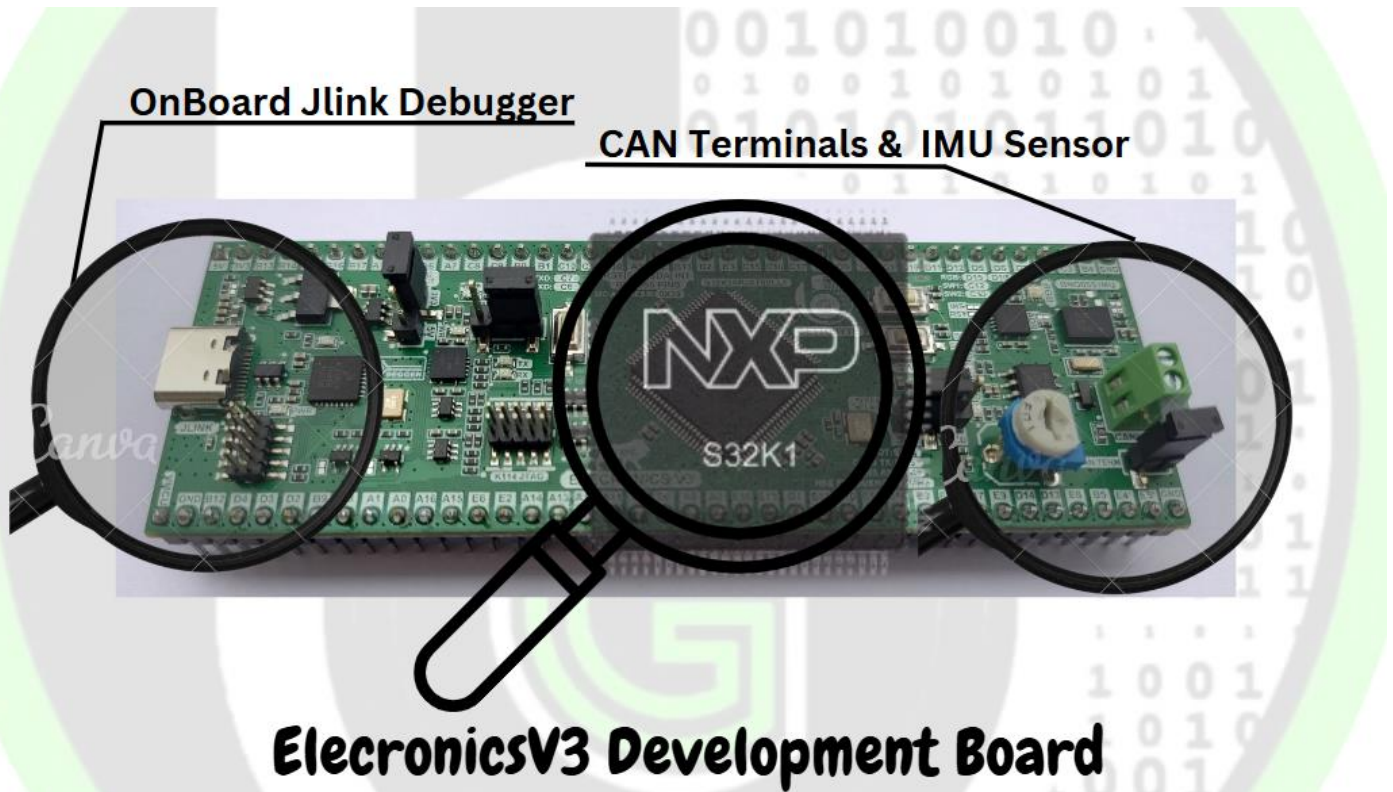
ElectronicsV3 is a development board based on **NXP Semiconductor S32K144 MCU** with a **100-pin package**. When the board is shipped to you, it will be shipped with the following content:

ElectronicsV3 Development Board



Arduino of Automotive World

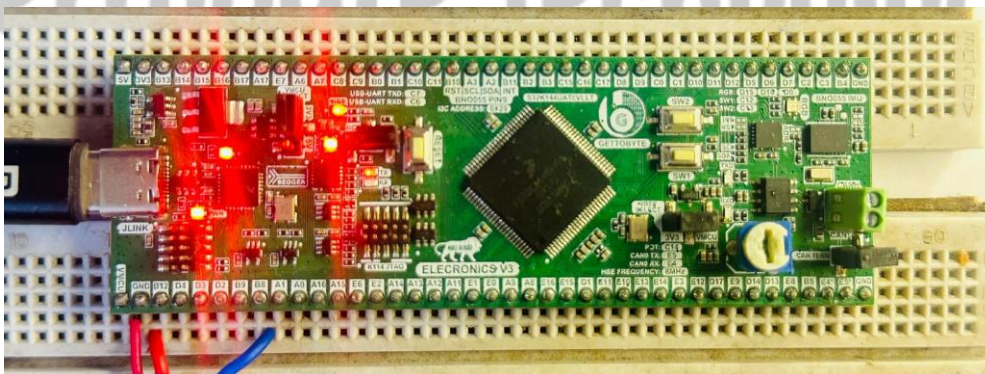
The Board has all the things which are needed to start with it, **we don't need any external debugger or connecting wires to use it**. It has on Board Jlink Debugger which can be accessed by Type C USB Cable, Virtual COM Port which can be accessed by same Type C Cable, CAN transceiver IC making ElectronicsV3 self sufficient CAN Node, Bosch BNO055 IMU Sensor and various other small features. To know **more in-depth about ElectronicsV3 board hardware refer to its User Manual (Link in Reference Documents Section)**.



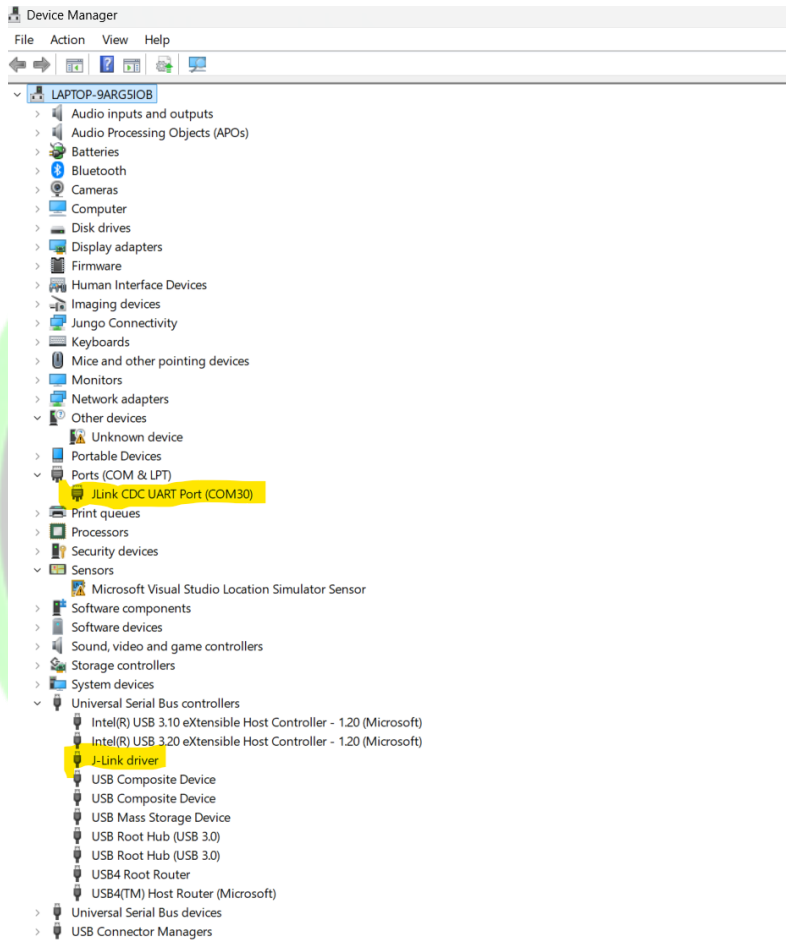
ElectronicsV3 Development Board

To use ElectronicsV3 Board, you **Just** need to arrange the USB Type C cable on your own and connect to its USB Type C Port. The ElectronicsV3 board would be powered, programmed, debugged and UART interface via same USB Port. So ElectronicsV3 Onboard USB Type C port provides **very easy and hassle-free setup to get started with the Automotive MCU in easy way.**

When you will connect the ElectronicsV3 Board with USB type C cable connected to host PC/Desktop, the board will be powered, and its onboard power and debugger LEDs will start glowing. Indicating board is powered successfully.



On your Host Desktop/PC, open the Device manager and you will see Jlink Driver and Virtual UART COM Port being detected over there. Indicating that necessary hardware drivers required to program and debug the ElectronicsV3 board via its Onboard Jlink Debugger and UART COM are identified by Host PC/Desktop.



That all what is required for getting started with ElectronicaV3 Board from hardware point of view. Now lets move to Software Applications that we would be needing to use ElectronicsV3 board and their installation and configurations.

Reinventing Tech With Bytes

Getting to Know the Software

In terms of Software, to do the development on electronicsV3, we need the following software:

- 1) **S32 Design Studio Integrated Development Environment**
- 2) **S32K1 SDK, which includes all peripheral drivers and NXP demo codes.**



Software tools needed to use ElectronicsV2

Both of these things users can get to know from this blog: [Getting Started with S32 Design Studio Part 1 | Gettobyte](#), in which I have explained step by step the installation of the above 2 mentioned things.

Table of Contents

1. Overview
2. Objective of the Blog
3. What is S32 Design Studio IDE
4. S32 Design Studio IDE and Software Package Installation
 - 4.1. S32 Design Studio IDE Installation:
 - 4.2. Microcontroller software development package and SDK Installation
 - 4.2.1. S32K1xx Microcontroller Development Package Installation
 - 4.2.2. S32K1xx Non Autosar Complaint RTD packages Software Installation
 - 4.2.3. S32K1xx Autosar Complaint RTD packages Software Installation

Blog to refer for S32 Design Studio and S32 SDK Installation Software

For ease of development on ElectronicsV3, there are some software tools available. These 2 tools ease the development and make application development fast:

1) S32 code configuration tool:

For the S32 **Configuration tool**, it comes with **S32 Design studio inbuilt**. Users can refer to this blog on how to use S32 Configuration tool: [Getting Started with S32 Design Studio Part 2 | Gettobyte](#)

2) Free master Debugger: Free master **debugger is a debugger tool**. How to use the Debugger tool and getting started with it would be there in future content.

How To do the Debugging and Building in S32 Design Studio

For Debugging and building the Embedded Software programs for ElectronicsV3 board, S32 Design Studio has inbuilt features.



For **Debugging the code**, S32 Design Studio has built in plug-ins and full support hardware and software debugging tools and features like:

- hardware debugger support: **PeMicro Debugger** and **J Link Debugger**.
- Software Debugger tool's: **GDB, OpenOCD**.
- Debugging features: **Step Into, Step Over, Watchpoints, Memory/Live Variable/Expression View, FreeRTOS debug plug in** and many others.

Users can refer this video to know how to do the debugging in S32 Design Studio:

https://youtu.be/ExAXWjBNP_4?si=7te68Rw-zEz4-NO-



Video for Debugging Project in S32 Design Studio

Also, **In terms of Building the project**, S32 Design Studio has comprehensive set of features and **GUI to do building**. Building in embedded is a process of generating the ELF file out of our source code(.c,.h files). And then this ELF file is flashed into our target MCU via debugging/programming tools. In S32 Design Studio we have built features like:

- **GNU GCC compiler and make file support**
- Build Configurations: **To make different subprojects with in one project**
- Build Configuration Explorer: **To Select which files/folder to include in a project for building.**
- Memory analyzer: **To analyze what memory and space of our Code after compiling.**
- Build Clean/All: **GUI based framework to clean the build and build the project.**

Users can refer this video to know how to do the debugging in S32 Design Studio:

https://youtu.be/e0aK_sCnbhA



Video for Building Project in S32 Design Studio

There is one more tool, which we will be using for the ElectronicsV2 board, that is: **J-Link software and documentation pack**. This software pack contains various **software tools which can be used with J Link Debugger**. As we would be using J Link Debugger for Programming and debugging electronicsV2 board, so these tools would be used from time to time to analyze things with J Link debugger.

Download the **J-Link software and documentation pack**, from here: [SEGGER - The Embedded Experts - Downloads - J-Link / J-Trace](#). Download the exe file and follow the traditional software installation process.

Resources for ElectronicsV3

Best thing about ElectronicsV3, is that for using the ElectronicsV3 you will get all step-by step tutorials and content on Gettobyte Website. **An ElectronicsV2 board has full Autosar MCAL Layer support, FreeRTOS Support and Stack of various External Sensor/Modules** that can be connected to it.

Users can refer to this link for:

1. **Getting started and peripheral driver's on ElectronicV2:** [S32K144 Peripherals Archives - Get To Byte](#)
2. **External Sensor/Module stack:** [Sensor-Modules-interfacing Archives - Get To Byte](#)
3. **DIY projects with ElectronicsV2:** [DIY Projects Archives - Get To Byte](#)
4. **Autosar MCAL Layer:** [Peripheral Drivers\(Autosar\) Archives - Get To Byte](#)
5. **Automotive EdTech Courses:** [Store \(gettobyte.com\)](#)

It is recommended to follow the chronological order, which is stated above, to learn things in best way. Users should explore the 2nd and 4th pointer to see what kind of application we can develop using ElectronicsV2.

Most important thing in resources is **GitHub repo of ElectronicsV3**, which is developed and maintained by Gettobyte: [gettobyte/GB ElectronicsV S32K144 Autosar MCAL: NXP Semiconductors S32K144 Microcontroller Autosar MCAL layer demo examples \(github.com\)](#)

You can clone the repo and open the Project in S32 Design Studio. To know how to open the project in S32 Design Studio, [refer this link:](#)

In the repo you will find various examples of **different peripherals of the S32K144 MCU and different sensor/module interfacing with S32K144 MCU.**



Gettobyte Technologies
Reinventing Tech With Bytes

Reference Documents

S. No	Document Name	Reference
1)	User Manual of ElectronicsV3	Link
2)	Reference Manual of S32K144 MCU	Link
3)	Data brief of S32K144 MCU	Link
4)	Tutorials and Resources for ElectronicsV3 Board	Link
5)	Automotive EdTech Courses using ElectronicsV3 Board	Link

Gettobyte Technologies
Reinventing Tech With Bytes

Revisions

Date	Revision Number	Description	Author
04/08/2024	1.0	Getting Started Manual of ElectronicsV3 Board	Kunal Gupta



Gettobyte Technologies

Reinventing Tech With Bytes