EAXVA04

- Data transmission interfaces:
  - 3 Automotive Ethernet
  - 2 Standard Ethernet
  - 8 FPD-Link III Video Input
  - 1 PPS_IN, 4 PPS_OUT
  - 2 CAN, 6 CAN FD
  - 4 RS-232, 1 RS-485
  - 4 LIN
  - 1 USB
  - 1 HDMI
- Operating voltage: DC 9-36V
- Operation memory: 16GB
- Main chip: NVIDIA Xavier, Infineon TC297
- Storage memory: 32GB
- Calculation capability: 32TOPS (INT8)
- Dimensions: 3555
- Operating Temperature: -25 to 85 °C
- Humidity: 0 - 95%, no condensation
- Storage temperature: -40 to 125 °C
- Weight: less than 3,700g

Revision History

Date	Version	Detail	Reviser
Jun. 19, 2019	V1.0	Initial version	Yibo Wang
Jun. 24, 2020	V1.1	'2.2 Connector' information update	Peter Zhu
August 25, 2020	V1.2	'4.1 Specifications' information update	Chris Wu
March 4, 2022	V1.3	Update GMSL camera information	Jason Du

Contact us

Web: www.ecotron.ai

Email: info@ecotron.ai

support@ecotron.ai

Address: 13115 Barton Road, Ste H,

Whittier, CA, 90605, USA

Telephone: +1 562-758-3039

Fax: +1 562-352-0552

Contents

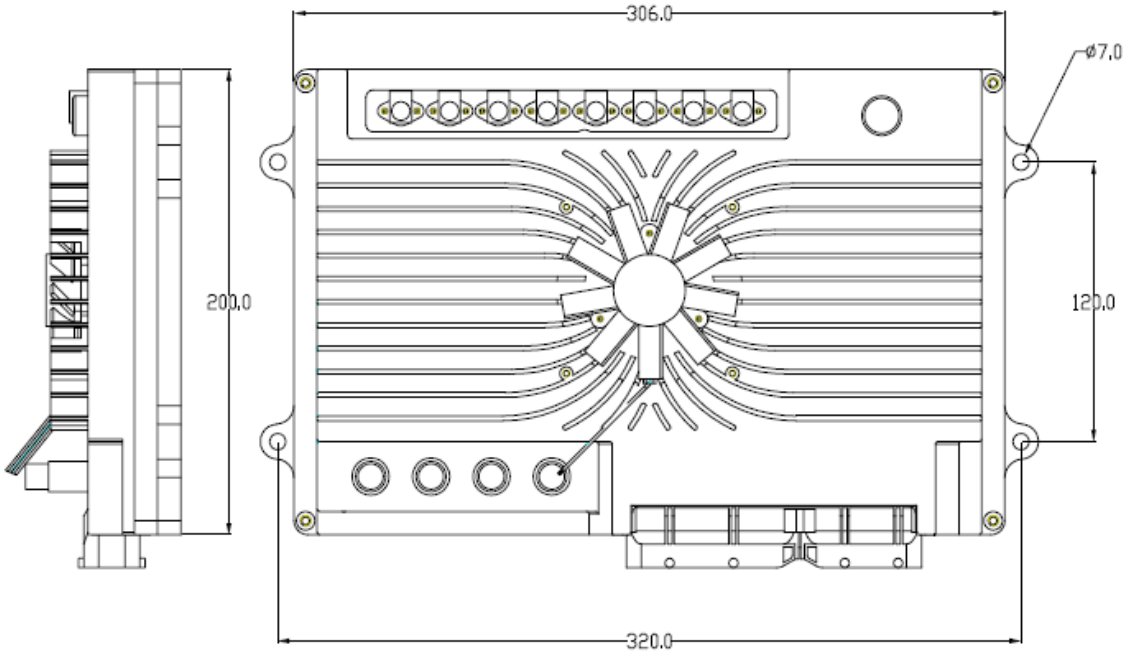
Chapter 1 Summary

EAXVA04 is an intelligent computing platform developed by Ecotron, for autonomous driving systems using NVIDIA Jetson AGX Xavier and Infineon TC297. Using the supporting basic software and development tools, developers can build an L4 level low-speed autonomous driving system in a safe, convenient and efficient manner.

NVIDIA Jetson AGX Xavier is designed for embedded intelligent systems including autonomous driving systems which can be used to implement autonomous driving function such as sensor fusion, environment perception, and path planning, etc. Infineon TC297 is based on TriCore™ architecture with a 300MHz operating frequency and an ECC (Error Correction Code) protected RAM with 728KB + 2MB capacity. It is designed based on the ISO26262 standard and supports ASIL-D safety level requirements. Developers can develop and deploy vehicle control and functional safety related strategies based on the MCU.

Chapter 2 Mechanical

2.1 Dimensions



2.2 Connector

The connectors used by EAXVA04 are qualified automotive rated products. The connector models are as follows:

#	Connector	Name	Type	Supplier	Link
1	121P	PCB needle	1241434-1	TE	--
2		81P sheath	1473244-1	TE	http://www.digikey.com/products/en?keywords=1473244-1
3		40P sheath	1473252-1	TE	http://www.digikey.com/products/en?keywords=1473252-1
4		Terminal	964282-2	TE	http://www.digikey.com/products/en?keywords=964282-2%20

5		Terminal	968220-1	TE	http://www.digikey.com/products/en?keywords=968220-1
6		81P back	1473247-1	TE	http://www.digikey.com/products/en?keywords=1473247-1
7		40P back	1473255-1	TE	http://www.digikey.com/products/en?keywords=1473255-1
8		81P retainer	368382-1	TE	http://www.digikey.com/products/en?keywords=368382-1
9		40P retainer	368388-1	TE	http://www.digikey.com/products/en?keywords=368388-1
10	FAKRA	FAKRA needle Z Type	smbf-fkm1-3gt30g-50	Amphenol	
11	Aviation plug	Board-side	EEG.1K.308.CLN	JX	
12		Harness-side	FGG.1K.308.CLAC	JX	
13		Board-side	EEG.1K.306.CLN	JX	
14		Harness-side	FGG.1K.306.CLAC	JX	

Chapter 3 Quick Start

3.1 Prepare in Advance

Before using this device, please prepare the following items:

- Stable power supply, 12V DC/ 5A min (At least a 50W power supply is recommended)
- USB to RS-232 adapter
- Laptop

3.2 Basic Knowledge

If you are a Linux beginner, it is helpful to learn how to use Linux command line tools. Here is a good Linux tutorial: tutorial.

3.3 Get Started

1. Connection

   Connect the positive and negative of the ACU to a DC power source, and then connect the RS232-1 of the device to the computer through the USB to RS-232 adapter. Please make sure that the computer can use the serial port normally.

2. Configuration

   Configure serial port:

   Baud rate: 115200

   8 data bits

   No parity checks

   1 stop bit

   Please use Putty or Minicom to open the serial port.

3. Start ACU

Turn on the ACU KeyOn switch first and turn on the device power. The device first starts U-Boot, then runs the Linux system.

If you see the following information shown below in the serial terminal window, it indicates that the system starts normally.

User name: *nvidia* **Password: *nvidia***

Chapter 4 Hardware

The hardware circuit of the computing platform is designed according to the application requirements of the autonomous driving system. The electrical parameters meet the requirements of the automotive industry regulations and has a variety of data transmission interfaces to meet the needs of multi-sensor fusion of the autonomous driving system. The main chip contains a variety of high-performance computing units to adapt to the computation-intense characteristics of autonomous driving, including sequential and parallel computing.

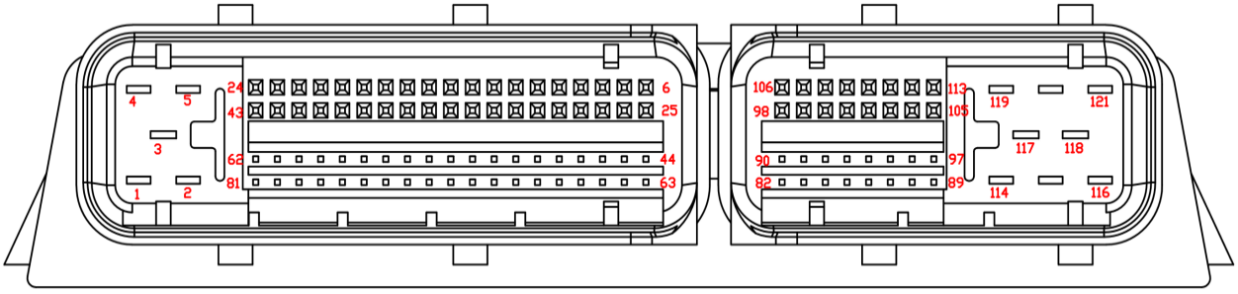
4.1 Specifications

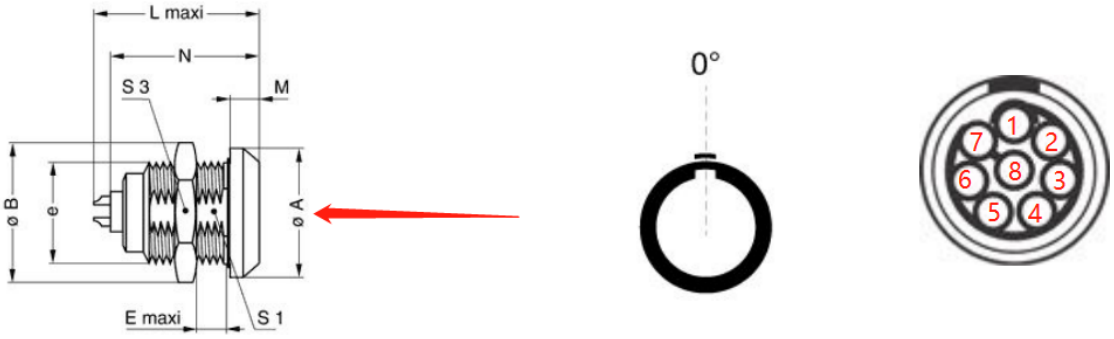
Item	Parameter
Operating voltage	DC 9-36V
Operation memory	32GB
Storage memory	32GB
Operating temperature	-25 to 85 °C
Operating humidity	0 - 95%, no condensation
Storage temperature	-40 to 125 °C
Dimensions	335mm*214mm*60mm
Weight	⩽3700g
Protection level	IP67

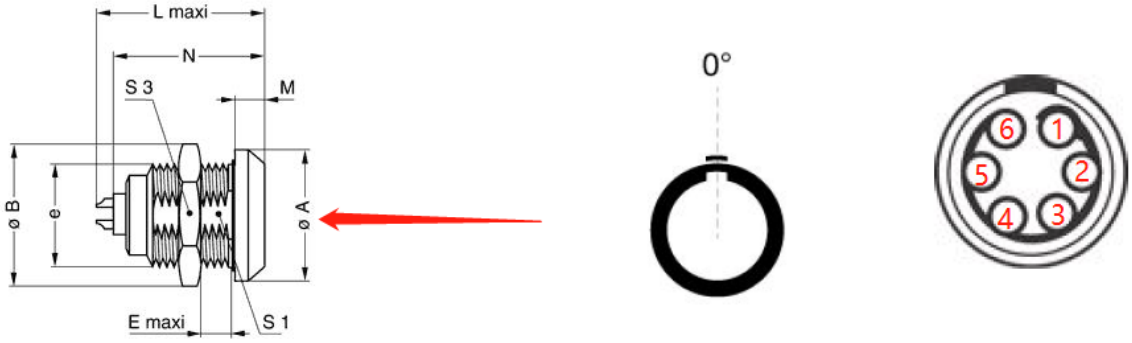
4.2 Device Ports

4.2.1 Port Placement

The distribution of input and output ports of the ACU is shown in the figure below. All the figures are front view.







4.2.2 Pinout

Signal Name	PIN	Description	Note
Automotive Ethernet			
ENet0_N	EEG1-3	Automotive Ethernet 0	1000Base-T1 can be configured to 100Base-T1
ENet0_P	EEG1-4		
ENet1_N	EEG1-5	Automotive Ethernet 1	

ENet1_P	EEG1-6		1000Base-T1 can be configured to 100Base-T1
ENet2_N	EEG1-1	Automotive Ethernet 2	1000Base-T1 can be configured to 100Base-T1
ENet2_P	EEG1-2		
Standard Ethernet			
NPort3_BI_DD+	EEG2-1	Standard Ethernet 3	1000Base-T1 can be configured to 100Base-T1
NPort3_BI_DD-	EEG2-2		
NPort3_BI_DC+	EEG2-3		
NPort3_BI_DC-	EEG2-4		
NPort3_BI_DB+	EEG2-5		
NPort3_BI_DB-	EEG2-6		
NPort3_BI_DA+	EEG2-7		
NPort3_BI_DA-	EEG2-8		
NPort4_BI_DD+	EEG3-1	Standard Ethernet 4	1000Base-T1 can be configured to 100Base-T1
NPort4_BI_DD-	EEG3-2		
NPort4_BI_DC+	EEG3-3		
NPort4_BI_DC-	EEG3-4		
NPort4_BI_DB+	EEG3-5		
NPort4_BI_DB-	EEG3-6		
NPort4_BI_DA+	EEG3-7		
NPort4_BI_DA-	EEG3-8		
Camera			
Camera-1	FAKRA-1	GMSL2 serial camera interface 1	Z Type
Camera-2	FAKRA-2	GMSL2 serial camera interface 2	Z Type
Camera-3	FAKRA-3	GMSL2 serial camera interface 3	Z Type
Camera-4	FAKRA-4	GMSL2 serial camera interface 4	Z Type
Camera-5	FAKRA-5	GMSL2 serial camera interface 5	Z Type
Camera-6	FAKRA-6	GMSL2 serial camera interface 6	Z Type
Camera-7	FAKRA-7	GMSL2 serial camera interface 7	Z Type

Camera-8	FAKRA-8	GMSL2 serial camera interface 8	Z Type
Display Interface			
HDMI	HDMI	HDMI Interface	
USB Interface			
USB	USB	USB Interface	Support USB2.0, USB3.0, USB3.1
Power			
BATT	121P-1 121P-3 121P-116 121P-118 121P-121	Power Positive	
Power Ground			
PGND	121P-2 121P-4 121P-5 121P-117 121P-119 121P-120	Power Ground	
Signal Ground			
GND	121P-36 121P-45 121P-63 121P-65	Signal ground	Ground for 5V sensor power supply

	121P-87		
Sensor Power Supply (5V)			
5V-1	121P-83	5V-1 Sensor Power Supply	Max current: 100mA
5V-2	121P-86	5V-2 Sensor Power Supply	Max current: 100mA
Wakeup Signal			
KEYON1	121P-44	KEYON1	High effective, control Xavier power on
KEYON_2	121P-56	KEYON2	High effective, control TC297 power on
Analog Input			
AI01	121P-42	Analog Input 0~5V (Voltage type)	12-bit resolution
AI02	121P-60	Analog Input 0~5V (Voltage type)	12-bit resolution
AI03	121P-43	Analog Input (Resistance type)	12-bit resolution
AI04	121P-24	Analog Input (Resistance type)	12-bit resolution
AI13	121P-62	Analog Input 0~36V (Voltage type)	12-bit resolution
AI14	121P-40	Analog Input 0~36V (Voltage type)	12-bit resolution
Digital Input			
DI01	121P-20	Digital Input 0~BATT	High effective
DI02	121P-58	Digital Input 0~BATT	High effective
DI03	121P-77	Digital Input 0~BATT	Low effective
DI04	121P-38	Digital Input 0~BATT	Low effective
DI07	121P-15	Digital Input 0~BATT	High effective
DI19	121P-35	Digital Input 0~BATT	Low effective
DI21	121P-74	Digital Input 0~BATT	High effective
DI22	121P-16	Digital Input 0~BATT	High effective
Output Signal			
HSO01	121P-88	Continuous 0.5A，Maximum 1A	
HSO02	121P-89	Continuous 0.5A，Maximum 1A	
HSO03	121P-97	Continuous 1A，Maximum 1.5A	
HSO04	121P-96	Continuous 1A，Maximum 1.5A	

LSO01	121P-101	Continuous 250mA	
LSO02	121P-94	Continuous 250mA	
LSO03	121P-90	Continuous 250mA	
LSO04	121P-92	Continuous 250mA	
LSO05	121P-110	Continuous 250mA	
LSO06	121P-103	Continuous 250mA	
LSO07	121P-109	Continuous 250mA	
LSO08	121P-107	Continuous 250mA	
Communication Port			
CAN_0_H	121P-31	With 120 Ω Terminal Resistor	Support CANFD, optional terminal resistor, corresponding to the CANA in EcoCoder
CAN_0_L	121P-32		
CAN_1_H	121P-11	With 120 Ω Terminal Resistor	Support CANFD, optional terminal resistor, corresponding to the CANB in EcoCoder
CAN_1_L	121P-12		
CAN_2_H	121P-29	With 120 Ω Terminal Resistor	Support CANFD, optional terminal resistor, corresponding to the CANC in EcoCoder
CAN_2_L	121P-30		
CAN_3_H	121P-13	With 120 Ω Terminal Resistor	Support CANFD, optional terminal resistor, corresponding to the CAND in EcoCoder
CAN_3_L	121P-14		
CAN_R0_H	121P-27	Without 120 Ω Terminal Resistor	

CAN_R0_L	121P-28		Support wakeup by user-defined message ID. Optional terminal resistor, corresponding to the CANE in EcoCoder
CAN_R1_H	121P-9	Without 120 Ω Terminal Resistor	Support wakeup by user-defined message ID. Optional terminal resistor, corresponding to the CANF in EcoCoder
CAN_R1_L	121P-10		
CAN_X0_H	121P-47	With 120 Ω Terminal Resistor	Optional termination resistor, corresponding to CAN0 in Xavier
CAN_X0_L	121P-66		
CAN_X1_H	121P-48	With 120 Ω Terminal Resistor	Optional termination resistor, corresponding to CAN1 in Xavier
CAN_X1_L	121P-67		
CAN_SHILD-1	121P-46	CAN Shield	
CAN_SHILD-2	121P-8	CAN Shield	
LIN0	121P-6	LIN bus 0	
LIN1	121P-26	LIN bus 1	
LIN2	121P-7	LIN bus 2	
LIN3	121P-25	LIN bus 3	
RS232_1_TXD	121P-52	RS-232 interface 1	Xavier *ttyTHS0*
RS232_1_RXD	121P-71		
RS232_2_TXD	121P-69	RS-232 interface 2	Xavier *ttyTHS1*
RS232_2_RXD	121P-50		
RS232_3_TXD	121P-51	RS-232 interface 3	Xavier *ttyTHS2,* used by default for Debug
RS232_3_RXD	121P-70		
RS232_4_TXD	121P-68	RS-232 interface 4	Xavier *ttyTHS6*
RS232_4_RXD	121P-49		
RS485_A	121P-34	RS485	Xavier *ttyTHS4*

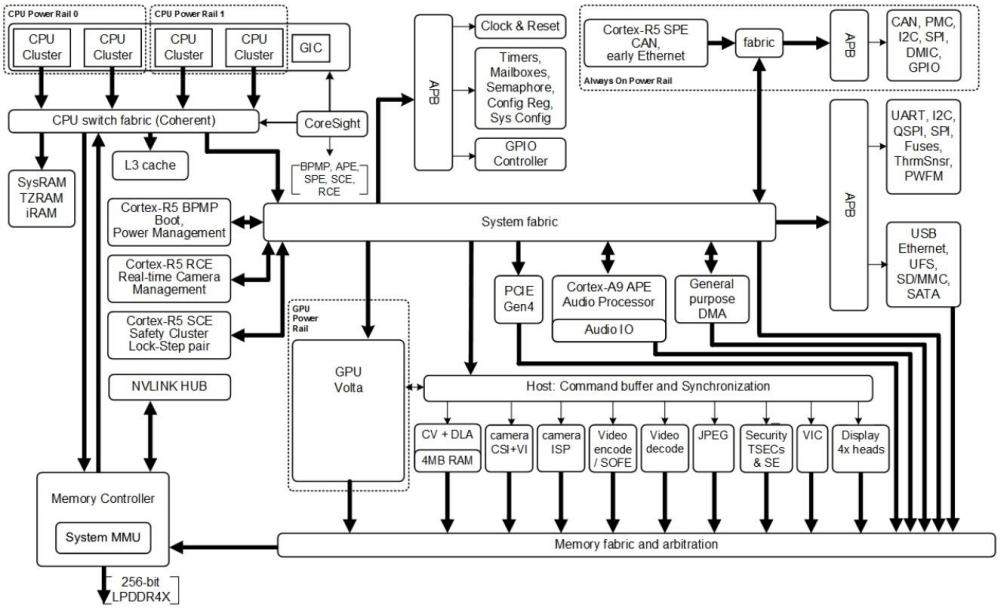
RS485_B	121P-33		
Others			
PPS_IN	121P-23	Second pulse synchronization input signal	Xavier, support 3.3V-16V, hardware configuration
PPS_OUT1	121P-81	Second pulse synchronization output signal	Xavier, 12V output
PPS_OUT2	121P-80	Second pulse synchronization output signal	Xavier, 12V output
PPS_OUT3	121P-79	Second pulse synchronization output signal	Xavier, 3.3V or 5V output
PPS_OUT4	121P-78	Second pulse synchronization output signal	Xavier, 3.3V or 5V output

4.3 System Main Chip

The main chip of EAXVA04 is NVIDIA Jetson AGX Xavier which is designed for embedded autonomous driving control systems. The computing performance of different internal processors is listed below.

- 8-Core CPU: 8-Core Carmel CPU based on ARMv8 ISA
- Deep Learning Accelerator (DLA): 5 TFLOPS (FP16) | 10 TOPS (INT8)
- Volta GPU：512 CUDA cores | 20 TOPS (INT8) | 1.3 TFLOPS (FP32)
- Vision Processor: 1.6 TOPS
- Stereo and Optical Flow Engine (SOFE): 6 TOPS
- Image Signal Processor (ISP): 1.5 Giga Pixels/s
- Video Encoder: 1.2 GPix/s
- Video Decoder: 1.8 GPix/s

The internal structure of the chip is shown below:

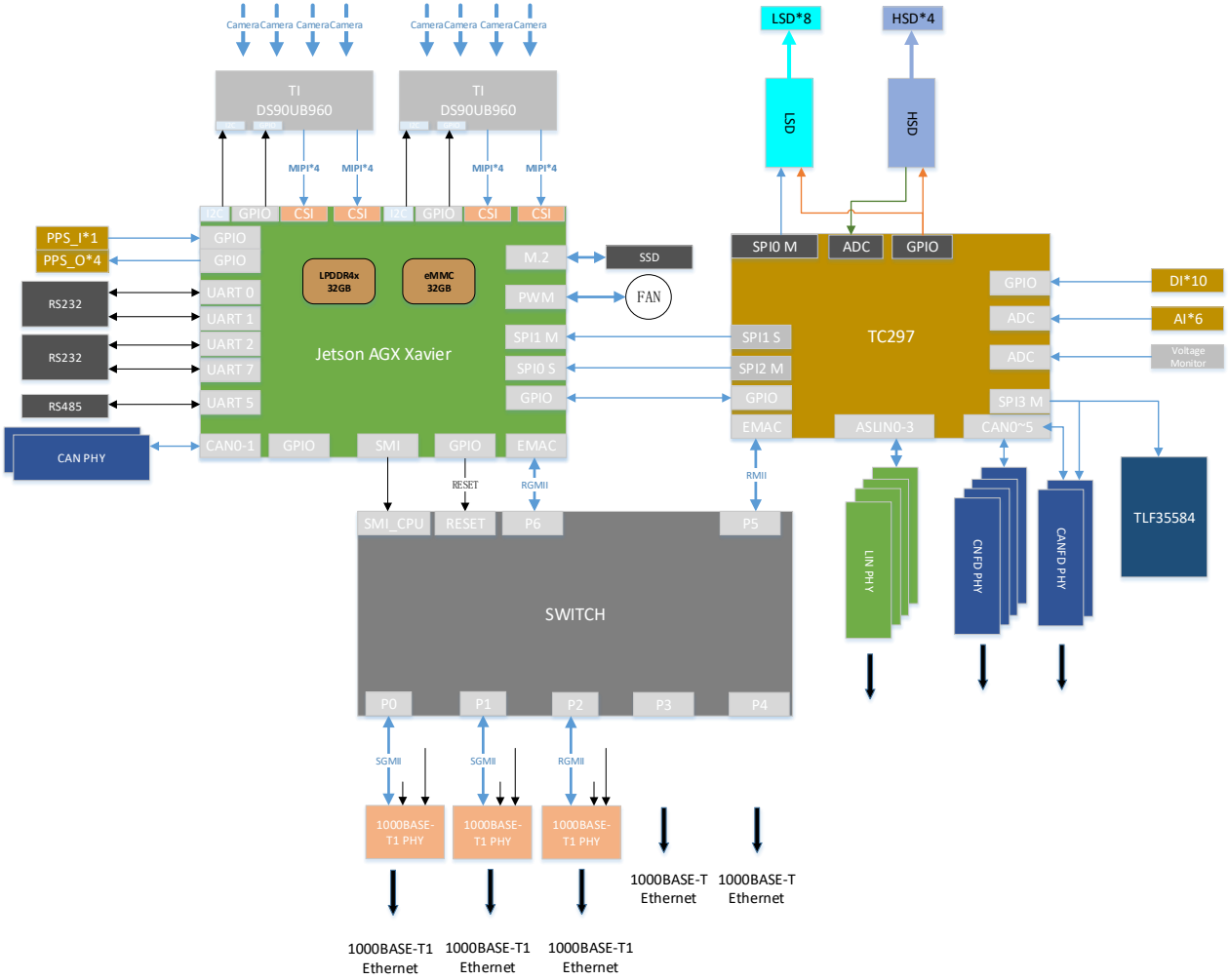


The microcontroller used in EAXVA04 is Infineon TC297 which has a TriCore™ architecture, working at 300MHz and has an ECC (Error Correction Code) protected RAM with a capacity of up to 728KB + 2MB, designed based on ISO26262, supporting up to ASIL-D. By working with a system basic chip (SBC), a hardware core security architecture design is realized. The resources of the chip are as follows:

Feature	**Detail**
Micro Control Core	32-bit Infineon TC297TP
Maximum Frequency	300MHz
Flash	8M
SRAM	728K
EEPROM	128K
Float Point Capability	Yes
SBC	TLF35584

4.4 Circuit Structure

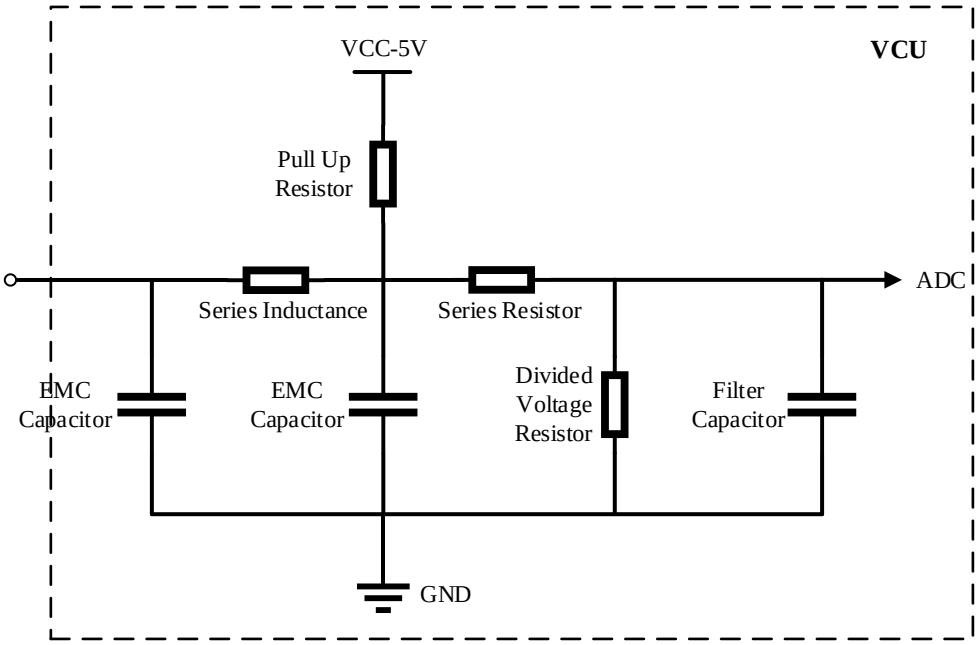
The internal circuit structure of EAXVA04 is shown below:



4.5 Circuit Description

4.5.1 Analog Input

The analog input channel circuit has the same structure, the circuit schematic and circuit details are shown below:



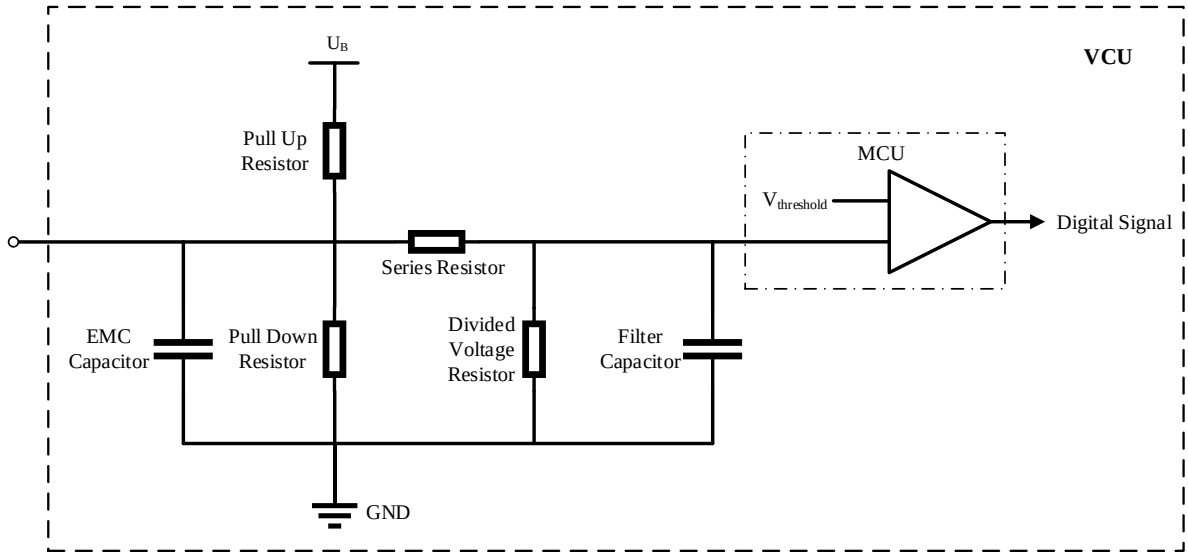
Note: 1. "--" means not soldered; 2. UB means power supply BATT voltage; 3. AI28 collects BATT voltage signal.

Pin #	AI	EMC Capacitor (F)	Pull Up Resistor to 5V (Ohm)	Series Resistor (Ohm)	Divided Voltage Resistor (Ohm)	Filter Capacitor (F)	Input Range Min	Input Range Max	Conditions / Remarks
42	AI01	100n	--	22k	--	1n	0V	5V	
60	AI02	100n	--	22k	--	1n	0V	5V	
43	AI03	100n	10k	22k	--	1n	--	--	Resistance type

24	AI04	100n	10k	22k	--	1n	--	--	Resistance type
62	AI13	100n	--	22k	3.48k	1n	0V	32V	
40	AI14	100n	--	22k	3.48k	1n	0V	32V	
--	AI28	100n	--	22k	3.48k	1n	0V	32V	BATT

4.5.2 Digital Input

The digital input channel circuit has the same structure, the circuit schematic and circuit details are shown below:



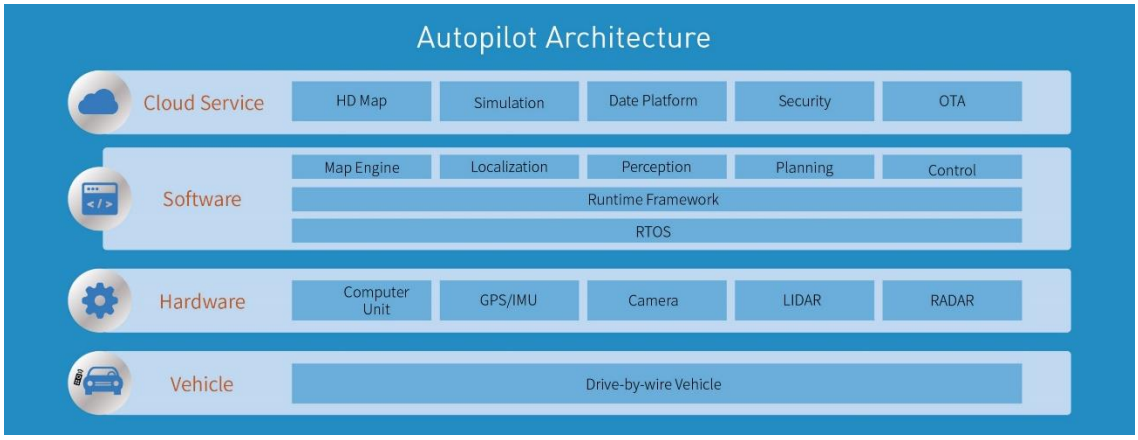
Note: 1. "--" means not welded. 2. UB represents the power supply BATT voltage. 3. KEYON and DC_WAKE only make hard-wire wake-up signal. 4. Digital input DI21, DI22 can be configured as frequency input SPEED1, SPEED2.

Pin #	DI	Pull Up Resistor	Pull Down Resistor	Operation Threshold for Input Signal		Input Range		Conditions/ Remarks
		to U_B(Ohm)	(Ohm)	V_{low}	V_{high}	Min	Max	
20	DI01	--	10k	3V	8.5V	0V	U_B	High effective
58	DI02	--	10k	3V	8.5V	0V	U_B	High effective
77	DI03	10k	--	3V	8.5V	0V	U_B	Low effective
38	DI04	10k	--	3V	8.5V	0V	U_B	Low effective
74	DI21	--	10k	3V	8.5V	0V	U_B	High effective
16	DI22	--	10k	3V	8.5V	0V	U_B	High effective
44	KEYON1	--	10k	--	--	0V	U_B	Wakeup Signal
56	KEYON2	--	10k	--	--	0V	U_B	Wakeup Signal

Chapter 5 SoC Basic Software

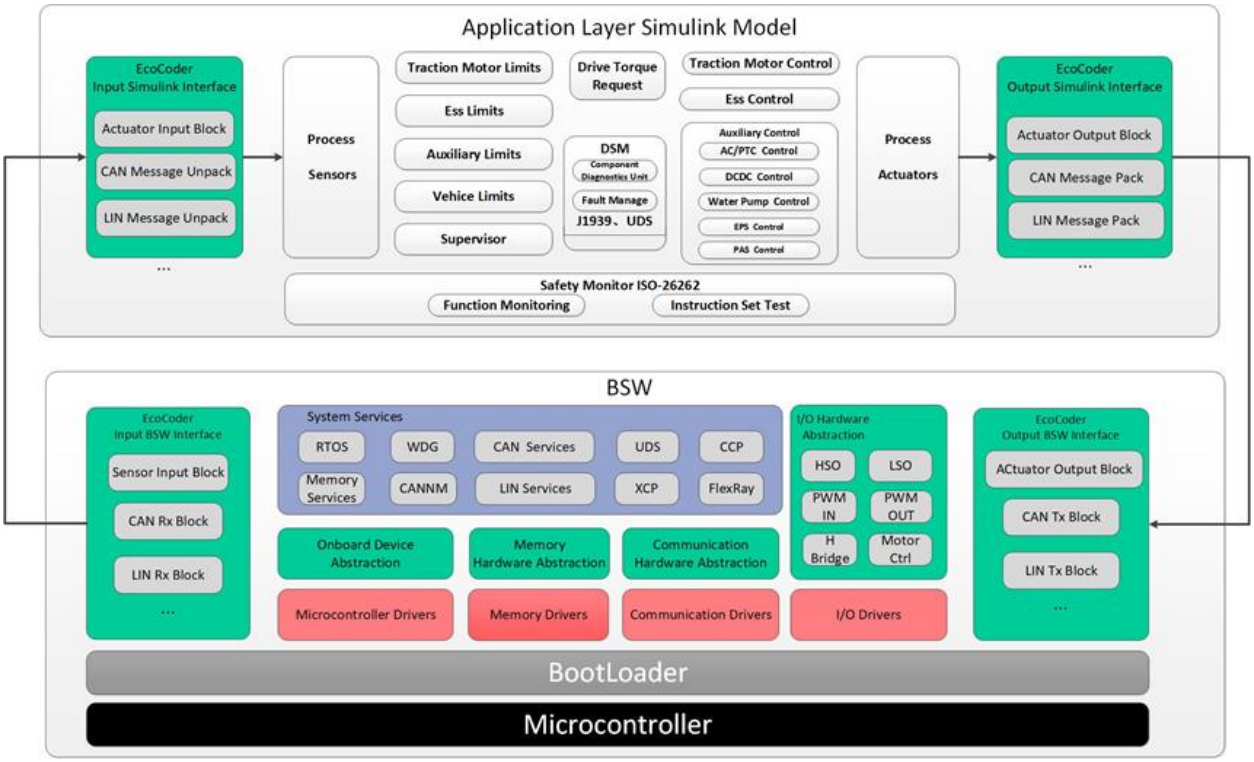
The SOC software system of the computing platform is customized for the autonomous driving system. A typical framework of an autonomous driving system is shown below. The SoC software system of EAXVA04 consists of RTOS and Runtime Framework. RTOS is a Linux operating system. Runtime Framework is ROS (Robot Operating System) Melodic.

Linux is a bridge connecting the hardware and the users, providing functions such as Process Management, Memory Management, File System, Network, Security, User Interface, and Device Drivers. Users can enter commands through the user interface of the operating system. The operating system interprets the commands, drives the hardware devices, and implements user requirements. ROS provides some standard operating system services, such as Hardware Abstraction, Low-Level Device Control, Inter-Process Messaging, and Message Packet Management. ROS is built on a graph architecture, various nodes can publish, subscribe and aggregate all kinds of information, e.g. sensing, control, status, planning.



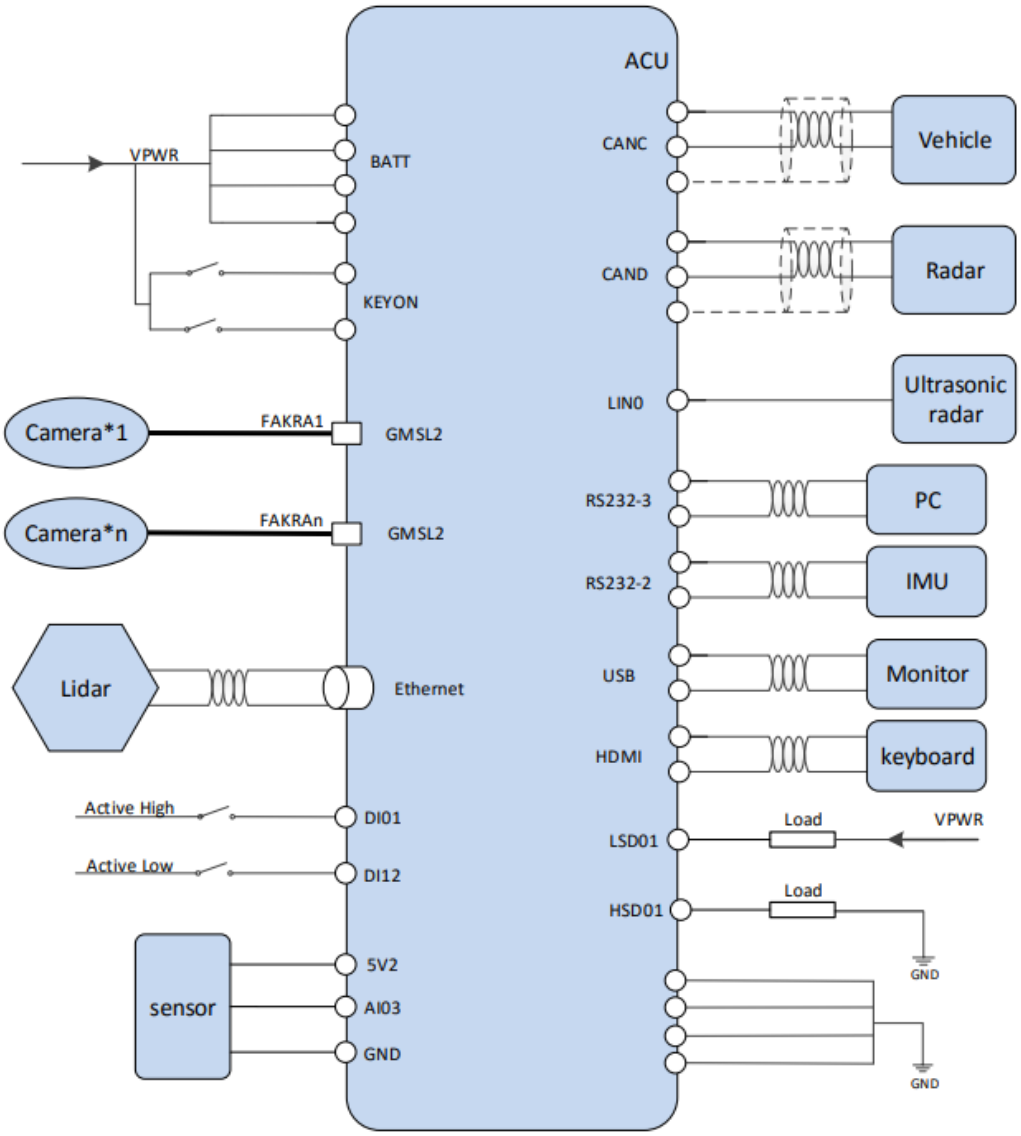
The software architecture of the MCU inside EAXVA04 is designed according to AUTOSAR, which is divided into Application Software Layer and Basic Software Layer. Basic Software Layer consists of a microcontroller abstraction layer, an ECU abstraction layer, a service layer, and a complex driver. Application software and basic software are connected and integrated through EcoCoder. EcoCoder encapsulates the low-level software interfaces into the Simulink library via s-functions. Application developers can use Simulink to build the model and generate executable program files for TC297 via Simulink by just one click.

The basic software interfaces that EcoCoder encapsulates can read digital and analog input signals, control high and low side outputs, support .dbc file interpretation, implement CCP and UDS protocols, and define the measurement, calibration and NVM variables. MCU application development is implemented with the calibration software EcoCAL and the flashing software EcoFlash.



Chapter 6 Demo Application

A demo for an autonomous driving hardware platform is shown below, which consists of EAXVA04 and sensors.



Chapter 7 Development Tool

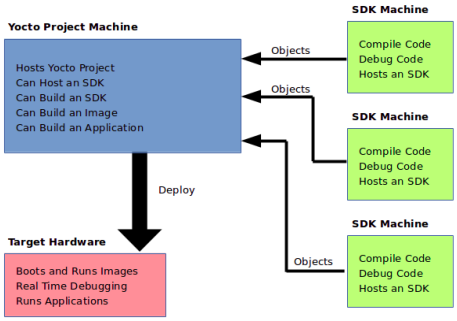
A combination of hardware, operating system stacks, and runtime environments are not capable enough to realize autonomous driving, therefore, users need to develop software that can perform specific functionality and deploy them to EAXVA04. For autonomous driving processor Xavier, two development tools are provided: EcoSDK-XV, EcoCoder-AI. For MCU Infineon TC297, three development tools are provided: EcoCoder, EcoCAL, and EcoFlash. Developers can select the tools they need.

7.1 EcoSDK-XV

EcoSDK-XV provides users with a complete application development environment, including:

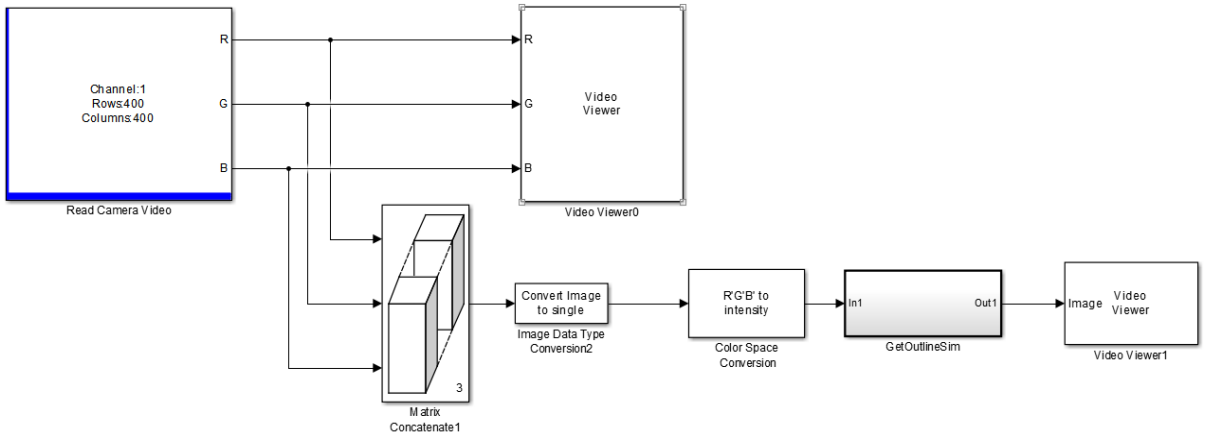
- **Cross-development toolchain:** consists of a cross-compiler, cross-connector, cross-debugger, and a set of other tools for application development.
- **System root:** EcoSDK-XV contains 2 system roots. One is for the development host, which contains the cross-development toolchain and other tools; the other is a complete root file system for the target, also contains development kits including header files and libraries.
- **Environment configuration:** The script provided by the EcoSDK-XV package allows users to configure an environment for cross-development on the development host.
- **Analysis tools:** userspace tools for analyzing applications on the target system.

EcoSDK-XV gives application developers all the tools necessary to write applications based on Linux, ROS, and Apollo Cyber RT. For details, please refer to EcoSDK-XV Manual.



7.2 EcoCoder-AI

EcoCoder-AI is a powerful automatic code generation library based on Matlab / Simulink that links directly to the target controller. EcoCoder-AI integrates code generation, compilation and one-click generation of executable files. In addition, the control model based on Simulink can be directly converted into an ROS-based executable program suitable for the target controller and downloaded to the target controller. For details, please refer to EcoCoder-AI Manual.



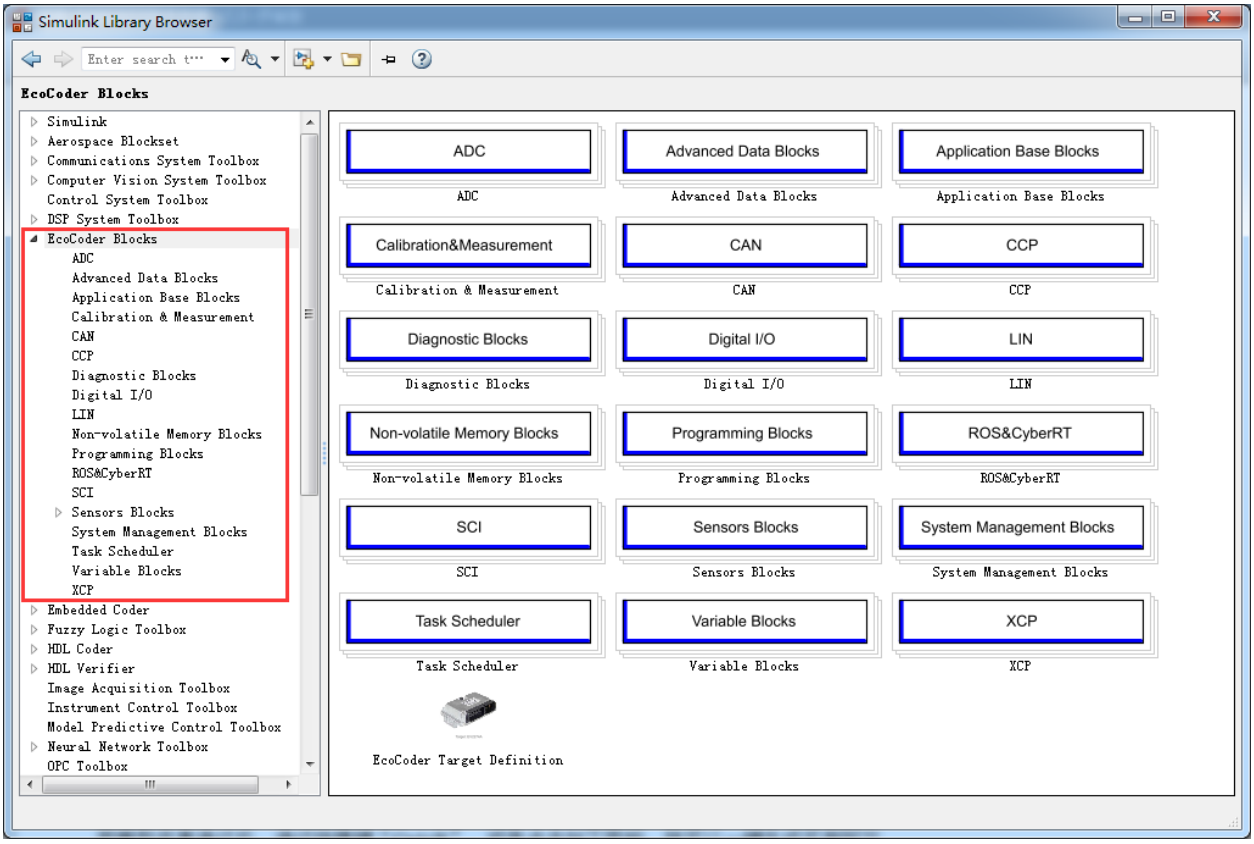
7.3 EcoCoder

EcoCoder is an application development tool for the control system, which makes it easier for users to develop embedded application software in the Simulink environment. It expands the resources of Simulink and Real-Time Workshop embedded encoders to generate the necessary code module and automatically configures and optimizes code generation. By encapsulating the low-level software library to s-functions, EcoCoder allows developers to use low-level software interfaces by model-based-design method and configure basic parameters. It can generate executable files and data description files with one click and provide .a2l file address update tool.

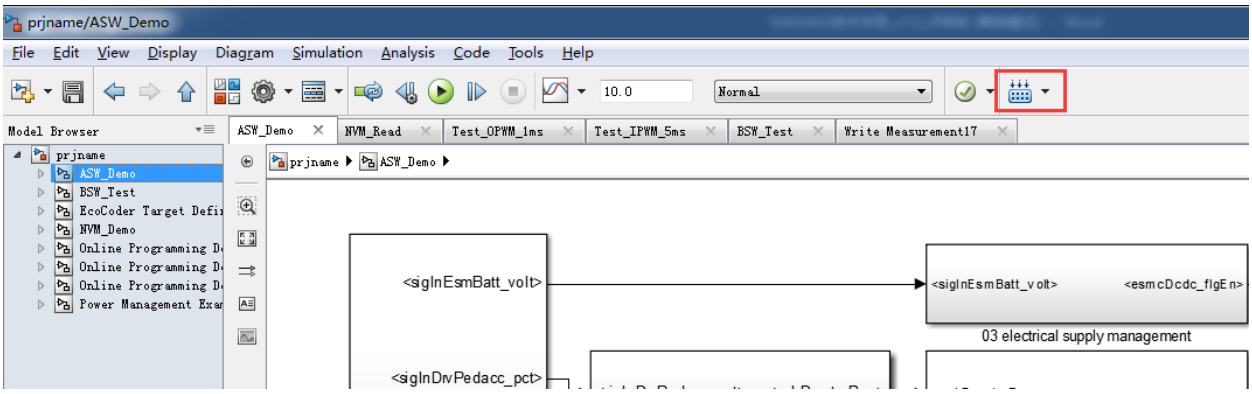
Features:

- Users develop embedded application software in the Simulink environment.
- Application developers can focus on control strategy development without knowing all the information about hardware.

- By encapsulating the low-level software library to s-functions, EcoCoder enables developers to use the low-level software interfaces and configure parameters using the model-based-design method.
- Executable file and data description file can be generated by one click, and a .a2l file address update tool is provided. During the generation, the code generated by the model is integrated with the low-level software automatically in the background, then makefile is used to call the compiler to generate executables.



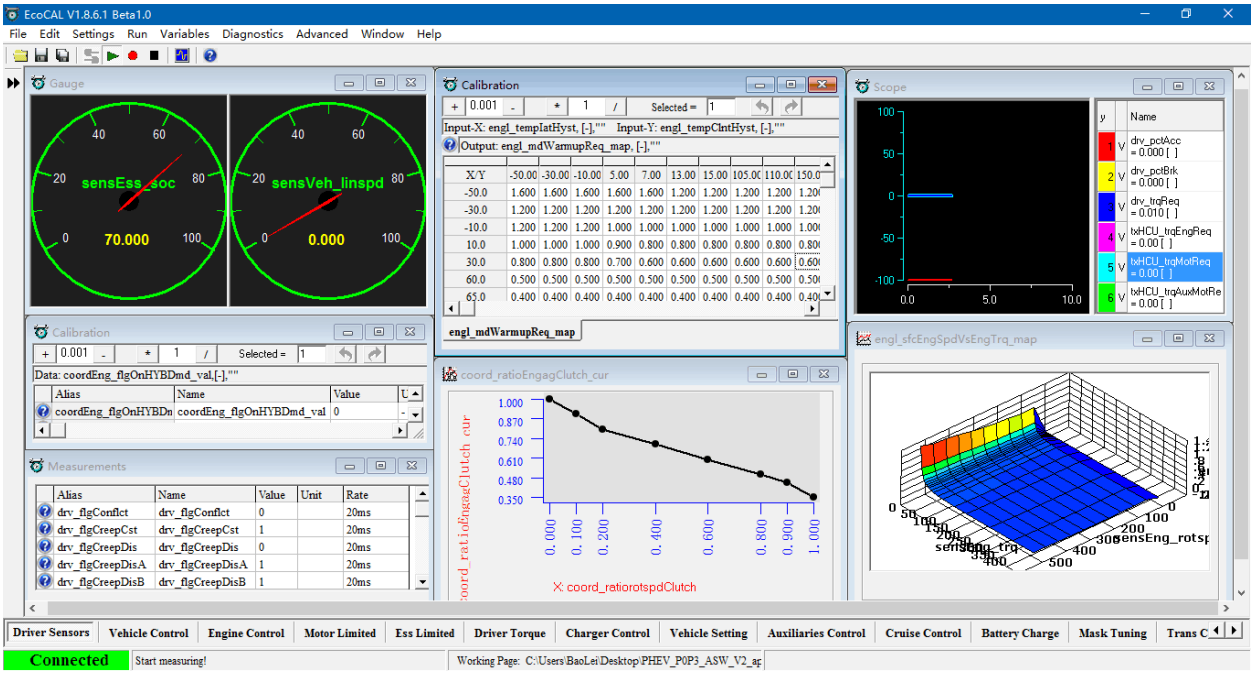
After compilation of the model, use the shortcut “Ctrl + B” or click the button shown below, the files ready to be flashed will be generated.



Developers can use EcoCoder to develop application software for MCU in EAXVA04. Please refer to EcoCoder User Manual.

7.4 EcoCAL

EcoCAL is a PC-side calibration software based on the CCP protocol. By loading .a2l and .hex files, real-time observation of the measurable variables and on-the-fly calibration can be realized. It can assist control strategy development engineers to debug and calibrate application software. Please refer to EcoCAL User Manual for more details.



7.5 EcoFlash

EcoFlash is PC-side software working with BootLoader to flash target program files. The CAN communication uses CCP/UDS protocol, and .s19, .mot and .hex files are supported.