PML® ESC0830 Development Board EEDB

User Manual

All Rights Reserved

Purple Mountain Laboratories
No. 9 Mozhou East Road, Jiangning District, Nanjing City
Postal Code：211111
Main: 025-57917093
Fax: 025-57917093
https://www.pmlabs.com.cn/

Table of Contents

List of Figures

List of Tables

ESC0830 Development Board EEDB Overview

The EEDB development board is an advanced functional development platform based on the ESC0830 microcontroller. This board is equipped with the ESC0830 microcontroller, independently developed by Purple Mountain Laboratories, featuring a main frequency of 80MHz and a rich array of peripheral interfaces, along with outstanding endogenous security characteristics. The EEDB development board not only fully supports all functionalities of this microcontroller but also integrates a variety of commonly used peripheral resources, providing developers with a comprehensive system-level development environment.

Features

The EEDB Development Board possesses the following features:

• Easy Setup — A USB cable can simultaneously provide debugging, communication, and power supply functions.

• Flexible Development Platform with rich peripheral support.

• Color TFT Display Interface

– TFT LCD interface

– Resistive touch interface

• LCD Display Interface

• 80 MHz ESC0830 Microcontroller, integrating 256KB Flash, 96KB SRAM, SPI, RS485, and CAN communication modules.

• 1MB Serial Flash Memory

• Controller Area Network (CAN) Interface

• 10/100 BaseT Ethernet Interface

• RS232 DB9 Interface

• RS485 Interface

• Wireless Module Interfaces such as WiFi, RFID, and Bluetooth modules.

• User LED Indicators and Buttons

• Rotary Potentiometer (can be used for menu navigation)

• MicroSD Card Slot

• Supports Multiple Debugging Options

– Supports JTAG protocol

– Standard ARM 10-pin JTAG debugging interface

– Supports USB debugging interface, onboard UART conversion.

• USB Virtual Serial Port

• Jumper Shorting Blocks for convenient reassignment of I/O resources.

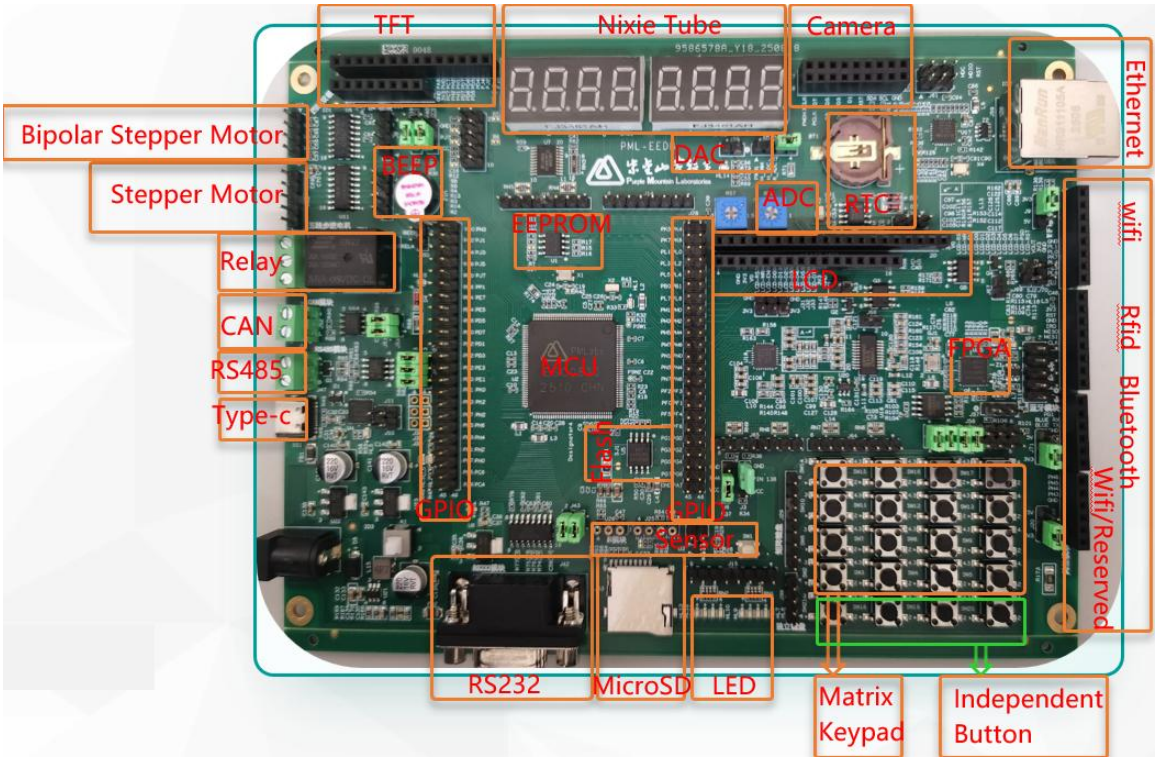


Figure 1: EEDB ESC0830 Development Board.

Development Kit Contents

The development kit includes a comprehensive set of components required for developing and running various applications with the ESC0830 microcontroller:

- ESC0830 development board
- USB Type-C cable for debugging
- 20-pin ribbon cable
- DC power cable
- Jumper caps
- Cloud storage containing evaluation boards for the following tools:

Microcontroller Development Toolkit

Block Diagram

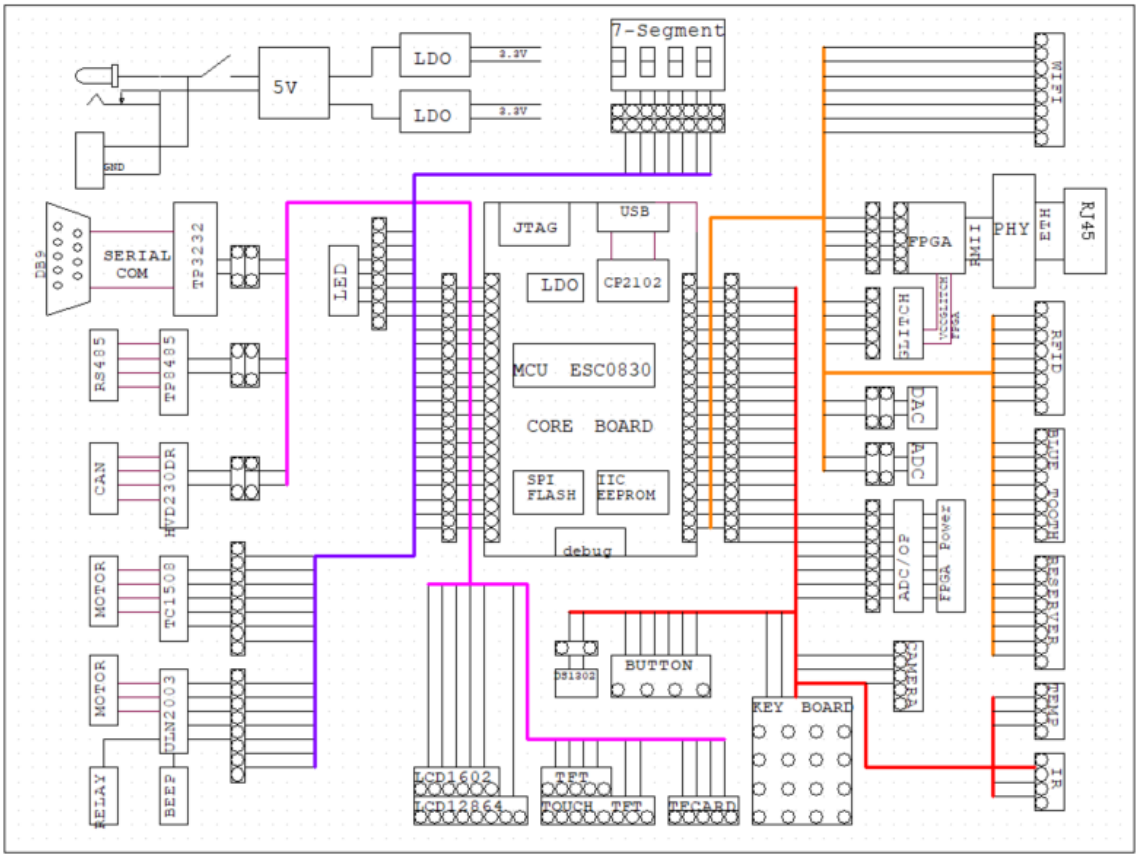


Figure 2. EEDB Development Board Functional Block Diagram

Development Board Specifications

Power Supply

• Supply Voltage: 9–17 V DC

• Power Interfaces:

– USB Type-C (connected to PC)

– DC Power Jack (external power adapter, specification 2.5×5.5 mm)

Physical Dimensions

• Length × Width × Height: 180 mm × 125 mm × 1.60 mm

Electrical Characteristics

• Reference Voltage: 3.3 V ±5%、5 V ±5%

• Compliance Standard: RoHS

Important Notes

• When the development board operates in USB host mode, it can supply power to connected USB devices through the host interface. In this case, jumper caps must be used to short the J10 and J11 pin headers.

• **Note:** It is prohibited to use both the power adapter and the USB interface for power supply simultaneously.

• If an external 9V power adapter with a rated current of ≥600 mA is not provided, the available supply current will be limited to 200 mA.

Hardware Description of the EEDB Development Board

This development board establishes a System-on-Chip (SoC) centered around the ESC0830 microcontroller, with extended key peripherals and debugging support. This chapter will focus on the hardware architecture of the system, emphasizing the control principles of various peripherals, their coupling methods with the core, and the usage specifications of standard debugging interfaces.

Overview of the ESC0830 Microcontroller

The ESC0830 is the world's first COTS-level MCU chip based on endogenous security theory and mimicry defense architecture. This chip integrates three heterogeneous instruction set microcontroller cores—ARM, MIPS, and RISC-V—along with a low-latency mimicry scheduler capable of nanosecond-level scheduling delays. It features 256KB of flash memory, an 80MHz operating frequency, and integrated SPI, UART, and IIC interfaces, encompassing a rich array of peripheral modules. For complete technical specifications, please refer to the "ESC0830 Microcontroller Datasheet."

Jumpers and GPIO Allocation

All peripherals on the development board are connected to the ESC0830 microcontroller via 2.54mm pitch jumper caps or pin headers. The factory default configuration is shown in Figure 2-1 (page 14). To ensure normal operation of the quick-start demonstration program, make sure all jumper caps are in their default positions.

Due to limitations in the number of microcontroller pins and their multiplexed functions, some peripherals cannot be used simultaneously. For example, the RS485 interface conflicts with GPIO pins PG4, PG5, and PG6. The RS485 jumpers are factory-configured to ensure basic functionality.

Table 1-1 lists peripherals that are not connected by default. To enable them, you must first disconnect any conflicting function jumpers. For more configuration options, please refer to the configuration table for each functional interface, which provides recommended configurations compatible with the example programs.

Table 1-1 Functions and Peripherals Connected in the Factory Default Configuration

Peripheral	Jumpers

RS485	J46, J48
Controller Area Network (CAN)	J51, J54
RS232	J41, J43
Analog 3.0V Reference	J37
PIN138	J3

Please refer to the appendix "EEDB Development Board Microcontroller GPIO Allocation Table" (page 38) for the complete GPIO allocation list. This table lists all MCU GPIO pins, with most GPIOs implementing default and alternative allocation schemes via 2.54 mm pitch pin headers and PCB routing. Additionally, the ESC0830 microcontroller features rich internal multiplexing functions, supporting further configuration options (such configurations share GPIO pins).

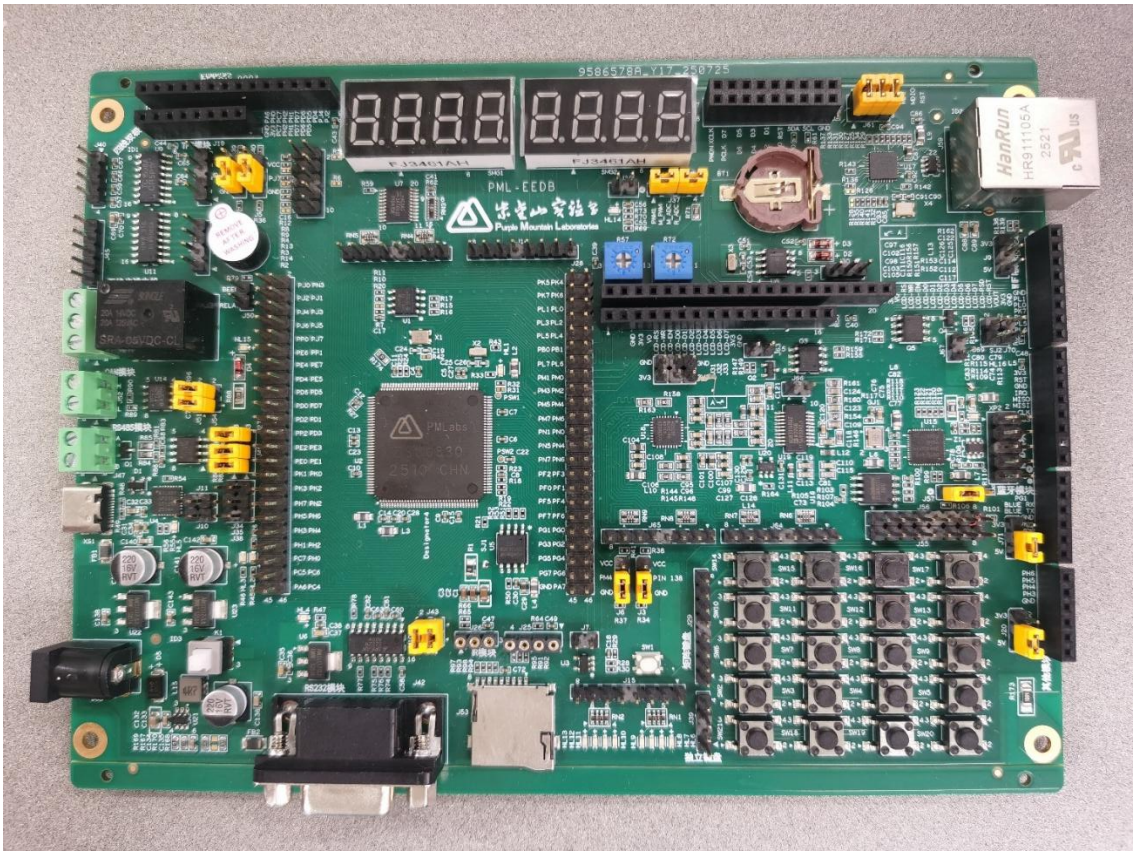


Figure 3: Factory Default Jumper Settings

Clock System

The clock system of this development board consists of the following three crystal oscillators:

• **MCU Main Clock (Y2)**: A 16.0 MHz crystal that provides the reference clock for the ESC0820 microcontroller. This clock can be multiplied by the internal PLL (Phase-Locked Loop) to generate the high-frequency clock required for system operation.

• **Ethernet Clock (Y1)**: A 25.0 MHz crystal dedicated to providing the reference clock for the Ethernet PHY (Physical Layer) chip.

• **FPGA Clock (Y3)**: A 25.0 MHz high-precision crystal that independently supplies the working clock for the FPGA chip.

Reset

The RESETn signal of the ESC0830 microcontroller is connected to the reset switch (SW2). A valid low-level external reset signal will be generated under any of the following conditions:

- Power-on reset (filtered through an RC network)
- Pressing the reset button SW2

Display modules, PHY, and other components have specific reset timing requirements, which must be managed via dedicated control lines from the microcontroller. The wiring of the reset signals will be explained in subsequent module chapters.

Power Supply

Power Supply Solution

The development board supports wide-voltage (4.5–17V) DC or standard 5V power input, selectable via jumpers (JP34–JP36), with the DC jack set as the factory default. The design strictly prohibits the coexistence of multiple input sources to avoid reverse current surges. The recommended DC power supply voltage range is 7–17V.

Power Distribution

After management and conversion, the input power forms four independent power rails to provide clean and stable voltages for different modules:

- VCC_FPGA: Core power supply for the FPGA.
- VCC3V3_M: Dedicated power supply for the microcontroller (ESC0830).
- VCC3V3: General peripheral power supply (e.g., memory, interface chips).
- VCC5V: Power supply for external expansion interfaces.

Among these, the +5V power is converted to various +3.3V power rails via low-dropout linear regulators (AMS1117-3.3), providing stable low-voltage power for the system.

Debugging

The ESC0830 microcontroller supports the standard JTAG protocol for program programming and online debugging. This protocol uses four signal lines—TCK (clock), TMS (mode select), TDI (data input), and TDO (data output)—to complete communication.

Program loading is accomplished via a dedicated USB interface labeled XS1. By configuring the logic levels of the relevant pins, either program loading (Download) or Run mode can be selected. Program downloading to the MCU is performed using Purple Mountain Laboratories' dedicated flashing tool, with specific operational methods detailed in the relevant documentation (ESC0830 Example Program Guide Manual).

Debugging Modes

The development board supports multiple hardware debugging configuration schemes. The specific configuration methods are detailed in Table 1-2.

Table 1-2 Mode Introduction

Mode	Debug Function	Use	Selected by...
1	Download Mode	Download to the onboard ESC0830 microcontroller via the debug USB interface	Default Mode
2	JTAG Debug Mode	Suitable for users who need to connect the development board via an external debugging interface (ULINK, JLINK, etc.)	Connect an external debugger to the JTAG interface

Table 1-3 Mode Settings

Microcontroller Pin	Board Function	Run	Download
PJ6	Configration	/	3.3V
PJ7	Configration	/	3.3V
Pin 138	Configration	GND	3.3V
PM4	Configration	/	3.3V

Important Note:
When the MCU is configured in download mode, please ensure:
1.Do NOT pull the PD3 pin to a high logic level.
2.Do NOT insert a MicroSD card.

Debugging USB Interface Overview

This development board utilizes the Silicon Labs CP2102 chip to implement USB-to-serial communication, creating a virtual COM port (VCP). This allows for a high-speed, stable communication link between the host and the target device using just a single USB cable. This serial port supports both program downloading and debugging functions, enabling users to complete firmware flashing via dedicated programming tools. For program downloading, a baud rate of 230400 is recommended, while a baud rate of 105200 is

recommended for serial port debugging.

RS485 Interface

The EEDB development board provides a standard RS485 communication interface. This interface achieves level conversion via the onboard TP8485 chip and is connected to the ESC0830 microcontroller's UART2 serial port. The RS485 interface uses a 3.85mm pitch terminal block for external connections. The UART2 function pins on the microcontroller side are multiplexed pins, allowing users to flexibly configure their signal mapping via jumper caps. For specific pin correspondence, please refer to the interface definition table in the appendix.

Table 1-4 RS485-related signals

Microcontroller Pin	Board Function	MCU UART	Jumper Name	Jumper Designator
PG5	UART TX	U2Tx	RS485_TX	J46、J48
PG4	UART RX	U2Rx	RS485_RX	
PG6	UART_RE		UART_RE	

a.Shared GPIO line (jumper cap installed by default).

b.This bus uses the MCU UART2.

CAN Interface

The EEDB development board integrates a CAN bus communication interface. This interface converts CAN signals to TTL levels via the onboard SN65HVD230DR transceiver and is connected to the CAN1 controller of the ESC0830 microcontroller. The CAN interface uses a robust 3.85mm pitch terminal block for external connections. The CAN1 function pins on the microcontroller are multiplexed, allowing users to flexibly configure their connections via jumper caps. For specific signal definitions and corresponding relationships, please refer to the table below.

Table 1-5 CAN Interface Related Signals

Microcontroller Pin	Board Function	MCU CAN	Jumper Name	Jumper Designator
PE7	CAN_TX	CAN1Tx	M_CAN_TX	J51、J54
PE6	CAN_RX	CAN1Rx	M_CAN_RX	

a.Shared GPIO line (jumper cap installed by default).
b.This bus uses the MCU CAN1.

RS232 Interface

The EEDB development board provides a standard RS232 interface. This interface

converts TTL levels to RS232 levels via the onboard TP3232N-SR chip and is connected to either UART1 or UART4 of the ESC0830 microcontroller. The interface uses a DB9 female connector externally, allowing direct connection to standard serial devices. The microcontroller pins corresponding to UART1 or UART4 are multiplexed function pins. Users can flexibly configure the connection method using jumper caps. Specific signal definitions and pin correspondences can be found in the table below.

Table 1-6 RS232 related signals

Microcontroller Pin	Board Function	MCU CAN	Jumper Name	Jumper Designator
PC4	RXD	U1Rx /U4Rx	RXD_R_M	J41、J43
PC5	TXD	U1Tx /U4Tx	TXD_R_M	

a.Shared GPIO line (jumper cap installed by default).

b.This bus uses the MCU UART1/UART4.

System Operation and Test Indicator Lights

The EEDB development board provides two user-programmable LED indicators, which are factory-configured by default to indicate system operating status and test status. The LED states are directly driven by the ESC0830 microcontroller. The corresponding pin definitions and control methods are detailed in the table below.

Table 1-7 Indicator light network table

Microcontroller Pin	Board Function	LED designator	Netlist Name
PJ7	System run	HL2	SYSTEM_RUN
PF3	test	HL3	test

Stepper Motor Interface

The EEDB development board provides a four-phase, five-wire stepper motor drive interface. This interface uses the onboard ULN2003 chip for current amplification and driving and is connected to the microcontroller's general-purpose I/O pins. The motor interface externally uses a standard 5-pin （J45） terminal block. Specific signal definitions and their correspondences with MCU pins can be found in the appendix table. Any GPIO of the MCU can be connected to the stepper motor control signals via DuPont wires.

Table 1-8 Stepper motor interface related signals

Microcontroller Pin	Board Function	Jumper Name	Related Designators

Any GPIO of the MCU (J27/J28)	CONTROL	MOTOR_IN1	J44.1
Any GPIO of the MCU (J27/J28)	CONTROL	MOTOR_IN2	J44.2
Any GPIO of the MCU (J27/J28)	CONTROL	MOTOR_IN3	J44.3
Any GPIO of the MCU (J27/J28)	CONTROL	MOTOR_IN4	J44.4

MicroSD Card Interface

The EEDB development board provides a standard MicroSD card interface. This interface utilizes the microcontroller's SSI (Synchronous Serial Interface) protocol to enable read/write access to the SD card, and externally uses a universal microSD card socket connector. For specific signal connection methods and their correspondences with MCU pins, please refer to the appendix table.

Table 1-9MicroSD card related signals

Microcontroller Pin	Board Function	Jumper Name	MCU Function	Socket designator
PD1	TF CARD	SD_CS	SSI1Fss	J53
PD0	TF CARD	SPI_MOSI	SSI1Clk	
PD2	TF CARD	SPI_SCK	SSI1Rx	
PD3	TF CARD	SPI_MISO	SSI1Tx	

a.Shared GPIO lines (default lines are already connected to the MicroSD card socket).

b.This bus uses the MCU SSI1.

WiFi Module Interface

The EEDB development board provides a universal WiFi module expansion interface, supporting connection to mainstream wireless communication modules such as the ATK-ESP8266, enabling flexible implementation of IoT and remote data transmission functions. This interface communicates with the module via GPIO-emulated UART protocol from the microcontroller. The hardware utilizes a standard 8-pin (J21) or 6-pin (J24) single-row pin header connector. Specific signal definitions and pin correspondences are detailed in the interface definition table. Additionally, if GPIO-emulated UART protocol is not desired, the MCU's multiplexed UART function pins can be used instead; however, in this case, the connectors designated as J21 and J24 cannot be utilized.

Table 1-10 WiFi module related signals

Microcontroller Pin	Board Function	Jumper Name	MCU Function	Connector designator
PL1	Wifi module TX	WIFI_CE	GPIO emulating UART RX	J21.3

PL0	Wifi module RX	WIFI_CS	GPIO emulating UART TX	J21.4
PK7	Wifi module RST	WIFI_SCK	GPIO	J21.5
PL4	Wifi module IO	WIFI_MOSI	GPIO	J21.6
PL5	/	WIFI_MISO	GPIO	J21.7
PL6	/	WIFI_IRQ	GPIO	J21.8
/	Wifi POWER	VCC	/	J21.1
/	Wifi POWER	GND	/	J21.2

a.Shared GPIO lines (default GPIO lines are already connected to the connector) .

b.GPIO Emulated UART Interface.

Microcontroller Pin	Board Function	Jumper Name	MCU Function	Connector designator
PH3	Wifi module TX	RSV_TX	GPIO emulating UART RX	J24.3
PH4	Wifi module RX	WIFI_CS	GPIO emulating UART TX	J24.4
PH5	Wifi module RST	WIFI_SCK	GPIO	J24.5
PH6	Wifi module IO	WIFI_MOSI	GPIO	J24.6
/	Wifi POWER	VCC	/	J241
/	Wifi POWER	GND	/	J24.2

Please note that if other models of WiFi modules are used, special attention must be paid to the signal definitions and electrical parameter matching of the interface header to avoid communication abnormalities or hardware damage.

RFID Module Interface

The EEDB development board provides a universal RFID module expansion interface, supporting connection to commonly used radio frequency identification modules such as the MFRC-522, enabling contactless data identification and collection functions. This interface communicates with the RFID module via GPIO-emulated SPI protocol from the microcontroller. The hardware uses a standard 8-pin single-row pin header connector. Specific signal definitions and pin correspondences are detailed in the interface definition table. If other models of RFID modules are used, be sure to verify the interface signal definitions and electrical parameters to ensure compatibility.

Table 1-21 RFID module related signals

Microcontroller Pin	Board Function	Jumper Name	MCU Function	Connector designator
PM1	RFID signals	MFR_NSS	GPIO emulating SPI	J22.1
PF6	RFID signals	MFR_CLK	GPIO emulating SPI	J22.2
PN6	RFID signals	MFR_MOSI	GPIO emulating SPI	J22.3

PN7	RFID signals	MFR_MISO	GPIO emulating SPI	J22.4
PM0	RFID signals	MFR_IRQ	GPIO	J22.5
/	Rfid GND	GND	/	J22.6
PL7	RFID control signals	MFR_RST	GPIO	J22.7
/	Rfid POWER	VCC	/	J22.8

a.Shared GPIO lines (default GPIO lines are already connected to the connector).

b.GPIO Emulated SPI Bus.

Bluetooth Module Interface

The EEDB development board provides a Bluetooth module expansion interface, supporting the connection of Bluetooth 4.0 BLE modules such as the AT-09 and BT05, enabling low-power wireless data transmission and device control functions. This interface communicates with the Bluetooth module via a serial protocol emulated by the microcontroller's GPIO. The hardware uses a 6-pin single-row pin header connector. Specific signal definitions and pin correspondences are detailed in the interface definition table. If other models of Bluetooth modules are used, be sure to verify the interface signal definitions, communication protocols, and electrical parameters to ensure compatibility.

Table 1-12 Bluetooth module related signals

Microcontroller Pin	Board Function	Jumper Name	MCU Function	Connector designator
PG1	Bluetooth	START	GPIO	J23.1
PG0	Bluetooth	BLUE_RX	GPIO	J23.2
PG3	Bluetooth	BLUE_TX	GPIO	J23.3
/	Bluetooth GND	GND	/	J23.4
/	Bluetooth POWER	POWER	/	J23.5
PG2	Bluetooth	PWRC	GPIO	J23.6

a.Shared GPIO lines (default GPIO lines are already connected to the connector).

b.GPIO port emulates Bluetooth module timing.

Temperature and Humidity Detection Module

The EEDB development board provides a universal sensor interface, supporting the connection of single-bus temperature sensors such as DS18B20 and humidity sensors such as DHT11, enabling environmental parameter detection and data collection functions. This interface communicates with the sensors via a single-bus protocol implemented through the microcontroller's GPIO. The hardware uses a standard 4-pin

single-row pin header connector. Specific signal definitions and pin correspondences are detailed in the interface definition table. If other models of sensor modules are used, be sure to verify the interface signal definitions, communication protocols, and electrical parameters to ensure compatibility.

Table 1-33 Temperature and humidity sensor module related signals

Microcontroller Pin	Board Function	Jumper Name	MCU Function	Connector designator
PE5	Detection Module	DS18B20	GPIO	J25.2
/	Detection Module	GND	/	J25.3
/	Detection Module	GND	/	J25.4
/	Detection Module	POWER	/	J25.1

a.Shared GPIO lines (default GPIO lines are already connected to the connector).

b.GPIO port emulates temperature sensor timing.

Infrared Sensor Module

The EEDB development board provides an infrared sensor interface, supporting the connection of infrared receiving modules such as the IRM-H638T/TR2, enabling the reception and decoding of infrared remote control signals. This interface communicates with the sensor via a single-bus protocol implemented through the microcontroller's GPIO. The hardware uses a standard 3-pin single-row pin header connector. Specific signal definitions and pin correspondences are detailed in the interface definition table. If other models of infrared sensor modules are used, be sure to verify the interface signal definitions, communication protocols, and electrical parameters to ensure compatibility.

Table 1-44 Infrared sensor module related signals

Microcontroller Pin	Board Function	Jumper Name	MCU Function	Connector designator
PG7	Detection Module	REMOTE_IN	GPIO	J26.1
/	Detection Module	GND	/	J26.2
/	Detection Module	VCC	/	J26.3

a.Shared GPIO lines (default GPIO lines are already connected to the connector).

b.GPIO port emulates infrared sensor timing.

Clock Module

The EEDB development board integrates a DS1302ZM/TR real-time clock module, providing precise timing and calendar functions. This module is connected to the mainboard via a standard 3-pin pin header and can communicate with any general-purpose GPIO pins of the microcontroller. Specific signal definitions and connection

methods are detailed in the relevant interface definition table.

Table 1-55 Clock module related signals

Microcontroller Pin	Board Function	Jumper Name	MCU Function	Related Designator
Any GPIO of the MCU (J27/J28)	Clock	DS_CLK	GPIO	J30.1
Any GPIO of the MCU (J27/J28)	Clock	DS_IO	GPIO	J30.2
Any GPIO of the MCU (J27/J28)	Clock	DS_RST	GPIO	J30.3

a.Connect to the clock module via ribbon cable using any GPIO.

b.GPIO port emulates DS1302 clock module timing.

Note: The board provides a coin cell battery socket suitable for CR1220 type coin cell batteries.

Nixie Tube Module

The EEDB development board provides two sets of nixie tube display modules, driven by the 74HC245 bus driver. These modules can be connected to the mainboard via two sets of 8-pin pin headers. The modules support connection to any general-purpose GPIO pins of the microcontroller, offering flexible hardware configuration capabilities. Specific signal definitions and connection methods are detailed in the relevant interface definition table.

Table 1-66 Nixie tube module related signals

Microcontroller Pin	Board Function	Control Name	MCU Function	Related Designators
Any GPIO of the MCU (J27/J28)	Nixie Tube Display Driver	a-g	GPIO	J16.1
Any GPIO of the MCU (J27/J28)	Nixie Tube Display Driver	b	GPIO	J16.2
Any GPIO of the MCU (J27/J28)	Nixie Tube Display Driver	c	GPIO	J16.3
Any GPIO of the MCU (J27/J28)	Nixie Tube Display Driver	d	GPIO	J16.4
Any GPIO of the MCU (J27/J28)	Nixie Tube Display Driver	e	GPIO	J16.5
Any GPIO of the MCU (J27/J28)	Nixie Tube Display Driver	f	GPIO	J16.6
Any GPIO of the MCU (J27/J28)	Nixie Tube Display Driver	g	GPIO	J16.7
Any GPIO of the MCU (J27/J28)	Nixie Tube Display Driver	dp	GPIO	J16.8
Any GPIO of the MCU (J27/J28)	Nixie tube cathode	G1-G4	GPIO	J14

a.Shared GPIO lines (connected to MCU GPIO ports via ribbon cable).

TFT LCD Interface

The EEDB development board provides a TFT LCD expansion interface, which connects

to the LCD module via 8-pin and 14-pin female headers. This interface is compatible with both touchscreen and non-touchscreen display types. Both types of screens share the microcontroller's control and data signals and are connected in a daisy-chain topology. The interface utilizes the MCU's multiplexed function GPIO pins to communicate with the TFT LCD. Specific signal definitions and pin correspondences can be found in the interface definition table.

Table 1-77 TFT LCD related signals

Microcontroller Pin	Board Function	MCU Function	Jumper Name	Connector designator
PC7	TFT_SCL	GPIO emulating SPI bus	T_CPT_DIN	J19.3
PH2	TFT_SDA	GPIO emulating SPI bus	T_CPT_CS	J19.4
PH1	TFT_RES	GPIO emulating SPI bus	T_CPT_CLK	J19.5
PD7	TFT_DC	GPIO emulating SPI bus	T_TFT_MISO	J19.6
PD6	TFT_CS	GPIO	T_TFT_LED	J19.7
PD5	TFT_BLK	GPIO	T_TFT_SCK	J19.8
/	TFT GND	/	GND	J19.1
/	TFT VCC	/	VCC	J19.2

a.Touchscreen TFT and standard TFT share GPIO lines (MCU GPIO is already connected to the TFT connector by default).

b.GPIO emulating SPI bus

Table 1-18 Touchscreen TFT LCD related signals

Microcontroller Pin	Board Function	MCU Function	Jumper Name	Connector designator
PA6	Touchscreen IRQ	GPIO	T_CPT_IRQ	J17.14
PH0	Touchscreen DO	GPIO	T_CPT_DO	J17.13
PC7	Touchscreen DIN	GPIO	T_CPT_DIN	J17.12
PH2	Touchscreen CS	GPIO	T_CPT_CS	J17.11
PH1	Touchscreen CLK	GPIO	T_CPT_CLK	J17.10
PD7	Touchscreen SPI	GPIO	T_TFT_MISO	J17.9
PD6	Touchscreen dimming	GPIO	T_TFT_LED	J17.8
PD5	Touchscreen SPI	GPIO	T_TFT_SCK	J17.7
PD4	Touchscreen SPI	GPIO	T_TFT_MOSI	J17.6
PJ4	Touchscreen read/write	GPIO	T_TFT_RS	J17.5
PJ2	Touchscreen reset	GPIO	T_TFT_RST	J17.4
PJ0	Touchscreen CS	GPIO	T_TFT_CS	J17.3

/	Touchscreen GND	/	GND	J17.2
/	Touchscreen VCC	/	VCC	J17.1

DAC Module

The EEDB development board provides DAC analog voltage output functionality, converting the microcontroller's PWM signal into a stable analog voltage through onboard filtering and conditioning circuitry. Users can enable the DAC output channel via jumper caps and observe changes in the DAC output voltage through the brightness feedback of the onboard LED (HL14). Specific signal connection methods and configuration instructions can be found in the relevant function configuration table.

Table 1-89 DAC module related signals

Microcontroller Pin	Board Function	MCU Function	Jumper Name	Jumper designator
PE4	DAC	M0PWM4/M1PWM2	M_PWM	J37.2
/	DAC	/	PWM1	J37.1
/	DAC out	/	DAC1	J36

a.To select the DAC function, use a jumper cap to connect pin1 and pin2 of J37.
b.The multiplexed functions PWM2 or PWM4 of MCU PE4 can be used.

ADC Module

The EEDB development board provides ADC analog signal acquisition functionality. It can generate analog voltage signals via the onboard adjustable potentiometer (R72) and feed them into the microcontroller's ADC interface for high-precision sampling. Users can enable this function via jumper caps. Specific signal connection methods and configuration instructions are detailed in the relevant function configuration table.

Table 1-20 ADC related signals

Microcontroller Pin	Board Function	MCU Function	Jumper Name	Jumper designator
PE1	ADC	AIN2	M_PWM	J37.3

a.To select the DAC function, connect pin3 and pin4 of J37 using a jumper cap.
b.The multiplexed function AIN2 of MCU GPIO can be utilized.

Camera Module

The EEDB development board provides a camera module expansion interface, utilizing an 18-pin dual-row female header connector. It is compatible with mainstream camera

modules such as the ATK-OV5640-AF, making it suitable for image acquisition and machine vision applications. Specific signal definitions and connection methods can be found in the interface definition table.

Table 1-21 Camera module related signals

Microcontroller Pin	Board Function	MCU Function	Jumper Name	Jumper designator
PF4	Camera IIC	GPIO	DCMI_SCL	J18.4
PF5	Camera IIC	GPIO	DCMI_SDA	J18.6
PH7	Camera parallel data	GPIO	DCMI_D0	J18.8
PK0	Camera parallel data	GPIO	DCMI_D1	J18.9
PK3	Camera parallel data	GPIO	DCMI_D2	J18.10
PE0	Camera parallel data	GPIO	DCMI_D3	J18.11
PK1	Camera parallel data	GPIO	DCMI_D4	J18.12
PE2	Camera parallel data	GPIO	DCMI_D5	J18.13
PE3	Camera parallel data	GPIO	DCMI_D6	J18.14
PJ5	Camera parallel data	GPIO	DCMI_D7	J18.15
PK4	Camera control signal	GPIO	DCMI_VSYNC	J18.3
PK5	Camera control signal	GPIO	DCMI_HREF	J18.5
PK2	Camera control signal	GPIO	DCMI_RESET	J18.7
PP0	Camera control signal	GPIO	DCMI_PCLK	J18.16
PJ3	Camera control signal	GPIO	DCMI_PWDN	J18.18
PJ1	Camera control signal	GPIO	DCMI_XCLK	J18.17
/	Camera power supply	/	VCC	J18.1
/	Camera GND	/	GND	J18.2

LED Module

The EEDB development board integrates 8 common-cathode LED lights, suitable for basic GPIO control routines such as running lights and status indication. The LED array is connected to the microcontroller via an 8-pin pin header and can be flexibly configured to any general-purpose GPIO port. Specific signal definitions and correspondences can be found in the relevant interface definition table.

Table 1-22 LED light related signals

Microcontroller Pin	Board Function	MCU Function	Jumper Name	Related Designator

Any GPIO of the MCU (J27/J28)	LED（HL6）	GPIO	/	J15.1
Any GPIO of the MCU (J27/J28)	LED（HL7）	GPIO	/	J15.2
Any GPIO of the MCU (J27/J28)	LED（HL8）	GPIO	/	J15.3
Any GPIO of the MCU (J27/J28)	LED（HL9）	GPIO	/	J15.4
Any GPIO of the MCU (J27/J28)	LED（HL10）	GPIO	/	J15.5
Any GPIO of the MCU (J27/J28)	LED（HL11）	GPIO	/	J15.6
Any GPIO of the MCU (J27/J28)	LED（HL12）	GPIO	/	J15.7
Any GPIO of the MCU (J27/J28)	LED（HL13）	GPIO	/	J15.8

Independent Button Module

The EEDB development board integrates 4 independent mechanical buttons, designed with active-low logic, supporting interrupt triggering and status detection functions. The buttons are connected to the microcontroller via a 4-pin pin header and can be flexibly configured to any general-purpose GPIO port. Specific signal definitions and correspondences can be found in the relevant interface definition table.

Table 1-23 Independent button related signals

Microcontroller Pin	Board Function	MCU Function	Jumper Name	Related Designator
Any GPIO of the MCU (J27/J28)	Button 1	GPIO	KEY1	J39.1
Any GPIO of the MCU (J27/J28)	Button 2	GPIO	KEY2	J39.2
Any GPIO of the MCU (J27/J28)	Button 3	GPIO	KEY3	J39.3
Any GPIO of the MCU (J27/J28)	Button 4	GPIO	KEY4	J39.4

Matrix Keypad Module

The EEDB development board integrates a matrix keyboard composed of 16 independent mechanical buttons, connected to the microcontroller via an 8-pin pin header. This keyboard can be flexibly configured to any general-purpose GPIO port, facilitating system expansion and custom function design. Specific signal definitions and pin correspondences can be found in the interface definition table.

Table 1-24 Matrix keypad related signals

Microcontroller Pin	Board Function	MCU Function	Jumper Name	Related Designator
Any GPIO of the MCU (J27/J28)	First row of matrix keypad	GPIO	/	J29.4

Any GPIO of the MCU (J27/J28)	Second row of matrix keypad	GPIO	/	J29.3
Any GPIO of the MCU (J27/J28)	Third row of matrix keypad	GPIO	/	J29.2
Any GPIO of the MCU (J27/J28)	Fourth row of matrix keypad	GPIO	/	J29.1
Any GPIO of the MCU (J27/J28)	First column of matrix keypad	GPIO	/	J29.5
Any GPIO of the MCU (J27/J28)	Second column of matrix keypad	GPIO	/	J29.6
Any GPIO of the MCU (J27/J28)	Third column of matrix keypad	GPIO	/	J29.7
Any GPIO of the MCU (J27/J28)	Fourth column of matrix keypad	GPIO	/	J29.8

LCD Screen Interface

The EEDB development board provides an LCD display expansion interface. Through 16-pin and 20-pin single-row pin header connectors, it is compatible with common liquid crystal display modules such as LCD1602 and LCD12864. Both types of displays share the control and data signals from the microcontroller and are connected using a daisy-chain topology. Communication is achieved by multiplexing the MCU's GPIO pins. Specific signal definitions and connection methods can be found in the interface definition table.

Table 1-25 LCD1602 screen related signals

Microcontroller Pin	Board Function	MCU Function	Jumper Name	Related Designator
PM3	LCD1602 RS	GPIO	LCD_RS	J12.4
PM5	LCD1602 WR	GPIO	LCD_WR	J12.5
PM7	LCD1602 EN	GPIO	LCD_EN	J12.6
PM6	LCD1602 data	GPIO	LCD_D0	J12.7
PN1	LCD1602 data	GPIO	LCD_D1	J12.8
PN0	LCD1602 data	GPIO	LCD_D2	J12.9
PN5	LCD1602 data	GPIO	LCD_D3	J12.10
PN4	LCD1602 data	GPIO	LCD_D4	J12.11
PN2	LCD1602 data	GPIO	LCD_D5	J12.12
PP2	LCD1602 data	GPIO	LCD_D6	J12.13
PN3	LCD1602 data	GPIO	LCD_D7	J12.14
/	LCD1602 GND	/	/	J12.1
/	LCD1602 VO	/	VO	J12.3

/	LCD1602 VCC	/	/	J12.2
/	LCD1602 VCC	/	/	J12.15
/	LCD1602 GND	/	/	J12.16

Table 1-26LCD16842 screen related signals

Microcontroller Pin	Board Function	MCU Function	Jumper Name	Related Designator
PM3	LCD16842 RS	GPIO	LCD_RS	J13.4
PM5	LCD16842 WR	GPIO	LCD_WR	J13.5
PM7	LCD16842 EN	GPIO	LCD_EN	J13.6
PM6	LCD16842 DATA	GPIO	LCD_D0	J13.7
PN1	LCD16842 DATA	GPIO	LCD_D1	J13.8
PN0	LCD16842 DATA	GPIO	LCD_D2	J13.9
PN5	LCD16842 DATA	GPIO	LCD_D3	J13.10
PN4	LCD16842 DATA	GPIO	LCD_D4	J13.11
PN2	LCD16842 DATA	GPIO	LCD_D5	J13.12
PP2	LCD16842 DATA	GPIO	LCD_D6	J13.13
PN3	LCD16842 DATA	GPIO	LCD_D7	J13.14
PL3	LCD16842 DATA	GPIO	LCD_PSD	J13.15
PL2	LCD16842 DATA	GPIO	LCD_RST	J13.17
/	LCD16842 GND	/	/	J13.1
/	LCD16842 VO	/	VO	J13.3
/	LCD16842 VCC	/	/	J13.2
/	LCD16842 VCC	/	/	J13.19
/	LCD16842 GND	/	/	J13.20

Note: The power and ground orientations of the LCD1602 and LCD12864 are different. Pay attention to the installation direction when connecting these two modules.

Relay Module

The EEDB development board integrates a relay module, which is driven by a ULN2003 Darlington transistor array and activates the relay when the input is set to a high level. This module provides a set of normally open (NO) and normally closed (NC) contact interfaces externally, using a 3.81mm pitch terminal block. The control end is connected

to the microcontroller GPIO via a 2-pin pin header. Specific signal definitions and connection methods can be found in the interface definition table.

Table 1-27 Relay module related signals

Microcontroller Pin	Board Function	MCU Function	Jumper Name	Jumper designator
Any GPIO of the MCU (J27/J28)	Relay control	GPIO	RELAY	J50.1

a.Shares a jumper pin with the buzzer module.

b.Use DuPont wires to connect the MCU pins to the pin header related to relay control.

Buzzer Module

The EEDB development board integrates a buzzer module, which is driven by a ULN2003 Darlington transistor array and triggers the buzzer to sound when the input is set to a high level. The control end is connected to the microcontroller GPIO via a 2-pin pin header. Specific signal definitions and connection methods can be found in the interface definition table.

Table 1-28 Buzzer related signals

Microcontroller Pin	Board Function	MCU Function	Jumper Name	Jumper designator
Any GPIO of the MCU (J27/J28)	Buzzer control	GPIO	BEEP	J50.2

a.Shares a jumper pin header with the relay module.

b.Use DuPont wires to connect the MCU pins to the pin header related to buzzer control.

Bipolar Stepper Motor Module

The EEDB development board integrates a bipolar stepper motor driver module, which utilizes the TC1508A dedicated driver chip to achieve precise control and efficient driving of the stepper motor. This module outputs motor drive signals externally via a 4-pin (J40) pin header, while the control end is connected to the microcontroller GPIO through a 4-pin (J8) pin header. Specific signal definitions and connection methods can be found in the interface definition table.

Table 1-29 Bipolar stepper motor related signals

Microcontroller Pin	Board Function	MCU Function	Jumper Name	Related Designator
Any GPIO of the MCU (J27/J28)	Bipolar motor control	GPIO	INA	J8.1
Any GPIO of the MCU (J27/J28)	Bipolar motor contro	GPIO	INB	J8.2

Any GPIO of the MCU (J27/J28)	Bipolar motor contro	GPIO	INC	J8.3
Any GPIO of the MCU (J27/J28)	Bipolar motor contro	GPIO	IND	J8.4

FPGA Module

The EEDB development board integrates a Gowin Semiconductor model GW1N-UV4QN48 FPGA. This FPGA is connected to the microcontroller via an SPI interface for communication and control management. Precise MOSFET control signals can be generated through the FPGA to implement glitch injection into the MCU's power supply. It can also connect to an Ethernet PHY chip via the RMII interface to expand Ethernet communication capabilities for the system. FPGA-related expansion signals are brought out through an 8-pin pin header. Specific signal definitions and correspondences can be found in the interface definition table.

Table 1-30 FPGA related signals

Microcontroller Pin	Board Function	MCU Function	Jumper Name	Jumper designator
PF0	SPI interface communicates with the MCU	SSI1Rx	M_FPGA_MISO	J55.1、J56.1
PF1	SPI interface communicates with the MCU	SSI1Tx	M_FPGA_MOSI	J55.2、J56.2
PF3	SPI interface communicates with the MCU	SSI1Fss	M_FPGA_CS	J55.3、J56.3
PF2	SPI interface communicates with the MCU	SSI1Clk	M_FPGA_SCK	J55.4、J56.4
PB1	IIC interface communicates with the MCU	GPIO	M_FPGA_SDA	J55.5、J56.5
PB0	IIC interface communicates with the MCU	GPIO	M_FPGA_SCA	J55.6、J56.6
/	Glitch injection	/	M_GLITCH_CLK	J55.7、J56.7
PP1	Reset FPGA	GPIO	M_FPGA_RST	J55.8、J56.8

a.The FPGA is connected to the MCU via 7 signal lines, enabling SPI and I2C protocol communication. These lines can also be adapted for other functions as needed.
b.The multiplexed function SSI1 of MCU GPIO can be used.

Glitch Injection Module

The EEDB development board supports voltage glitch injection functionality for both the MCU and FPGA, enabling both high-level and low-level glitch injection modes. Specifically, glitches in the MCU power supply are implemented by the FPGA through a MOSFET control circuit, while glitches in the FPGA power supply are implemented

by the MCU via a MOSFET control circuit. The two control signals are brought out through separate 2-pin pin headers. Specific signal definitions and correspondences can be found in the interface definition table. When using this feature, note that the current-limiting resistors (R1, R124) at the MCU and FPGA power inputs must be adjusted according to the actual glitch injection requirements。

Table 1-31 Glitch injection related signals

Microcontroller Pin	Board Function	MCU Function	Jumper Name	Jumper designator
PM2	FPGA low-level glitch control	GPIO	FPGA_GLITCHP	J69.1、J70.1
PM1	FPGA high-level glitch	GPIO	FPGA_GLITCH	J69.2、J70.2
/	FPGA glitch level selection	/	VCCGLITCH_FPGA	J67
/	MCU low-level glitch	/	GLITCHOUTLP	/
/	MCU high-level glitch	/	GLITCHOUT	/
/	MCU glitch level selection	/	VCCGLITCH	J63

a.When using the FPGA glitch voltage injection function, adjust resistor R124 according to the actual conditions.
b.When using the MCU glitch voltage injection function, adjust resistor R1 according to the actual conditions.

Ethernet Module

The EEDB development board provides one 10/100M adaptive Ethernet interface. This interface implements data communication between the FPGA and the YT8512C PHY chip based on the RMII protocol. The management interface (MDC/MDIO) of the YT8512C is directly controlled by the microcontroller, which uses GPIO to emulate the SMI bus for configuring and managing the PHY chip's status. Specific signal connection methods and pin definitions can be found in the relevant interface definition table.

Table 1-32 Ethernet related signals

Microcontroller Pin	Board Function	MCU Function	Jumper Name	Jumper designator
PJ7	SMI Management Por	GPIO	M_PHY_MDC	J55.1、J56.1
PJ6	SMI Management Por	GPIO	M_PHY_MDIO	J55.2、J56.2
PK6	SMI Management Por	GPIO	M_PHY_RST	J55.3、J56.3

a.MCU GPIO ports emulate MDC/MDIO signals to configure the PHY.

EEDB Development Board Schematic

This section includes the schematic diagram of the ESC0830 Development Board EEDB.

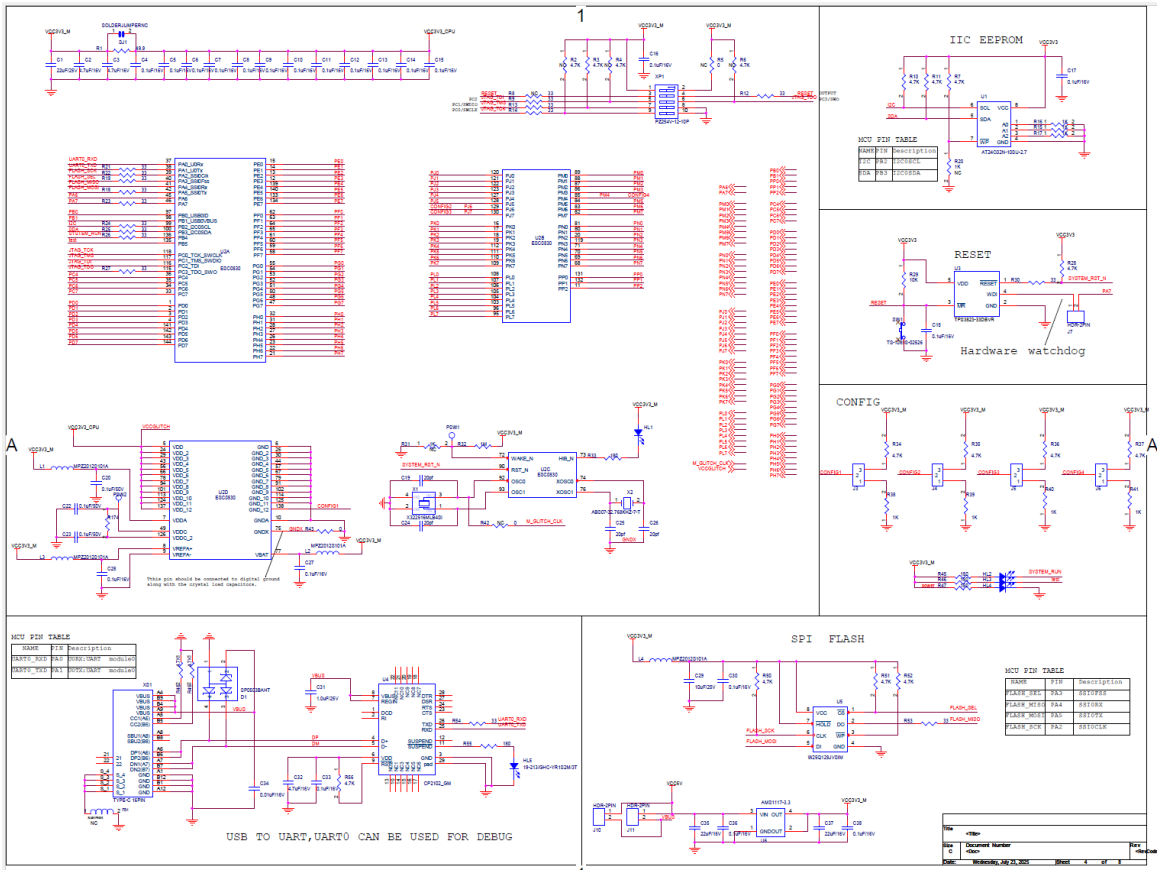


Figure 4: Development Board Schematic 1

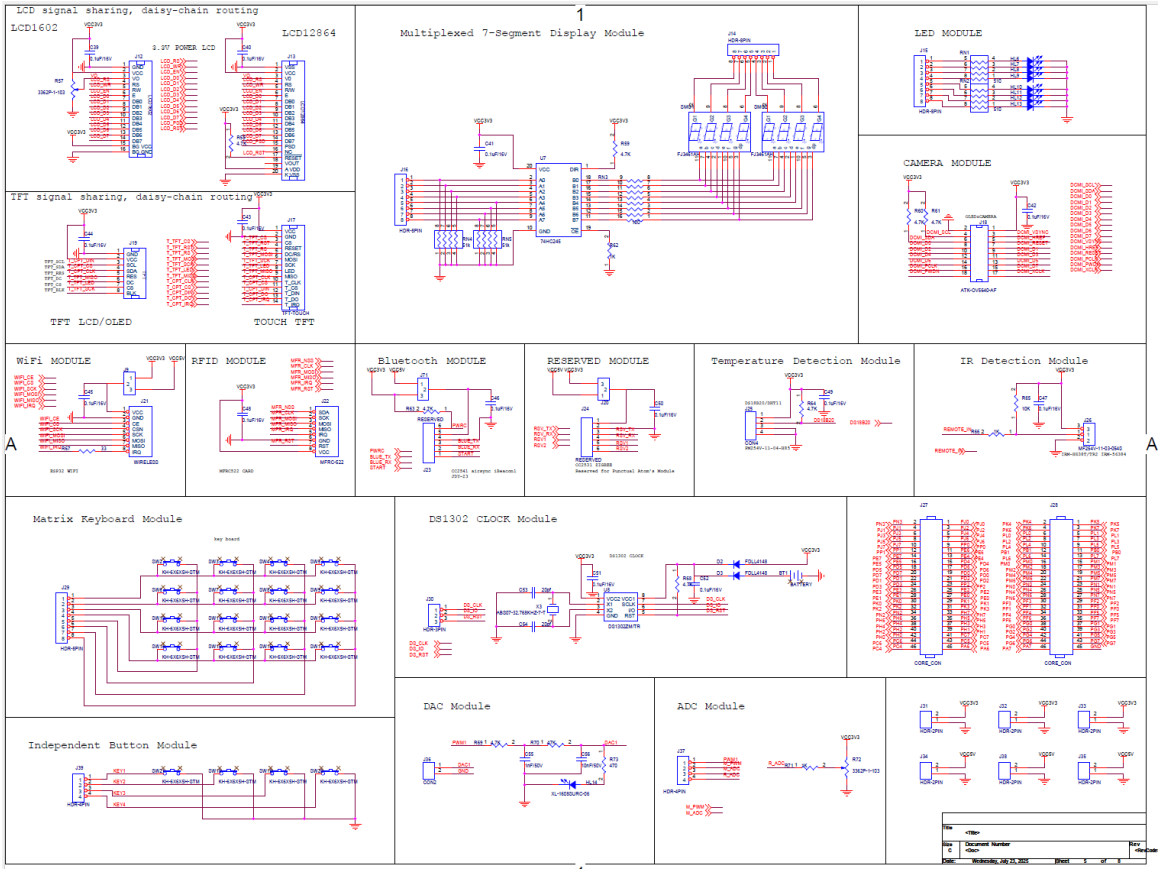


Figure 5: Development Board Schematic 2

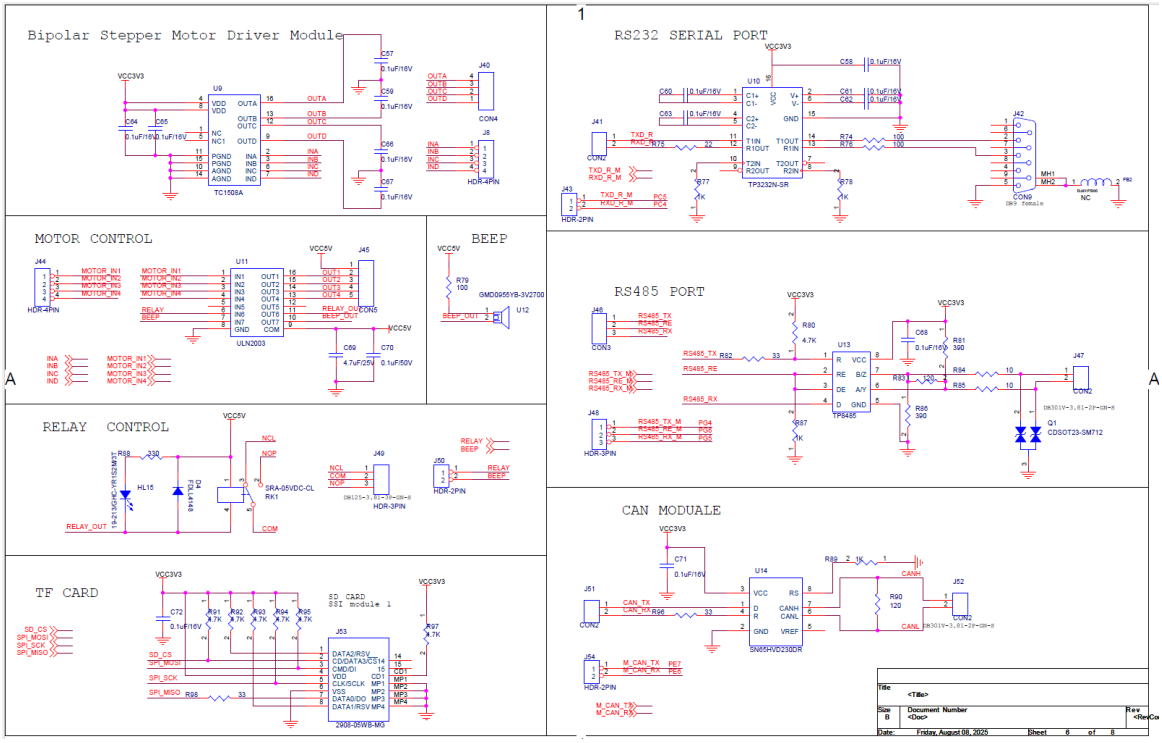


Figure 6: Development Board Schematic 3

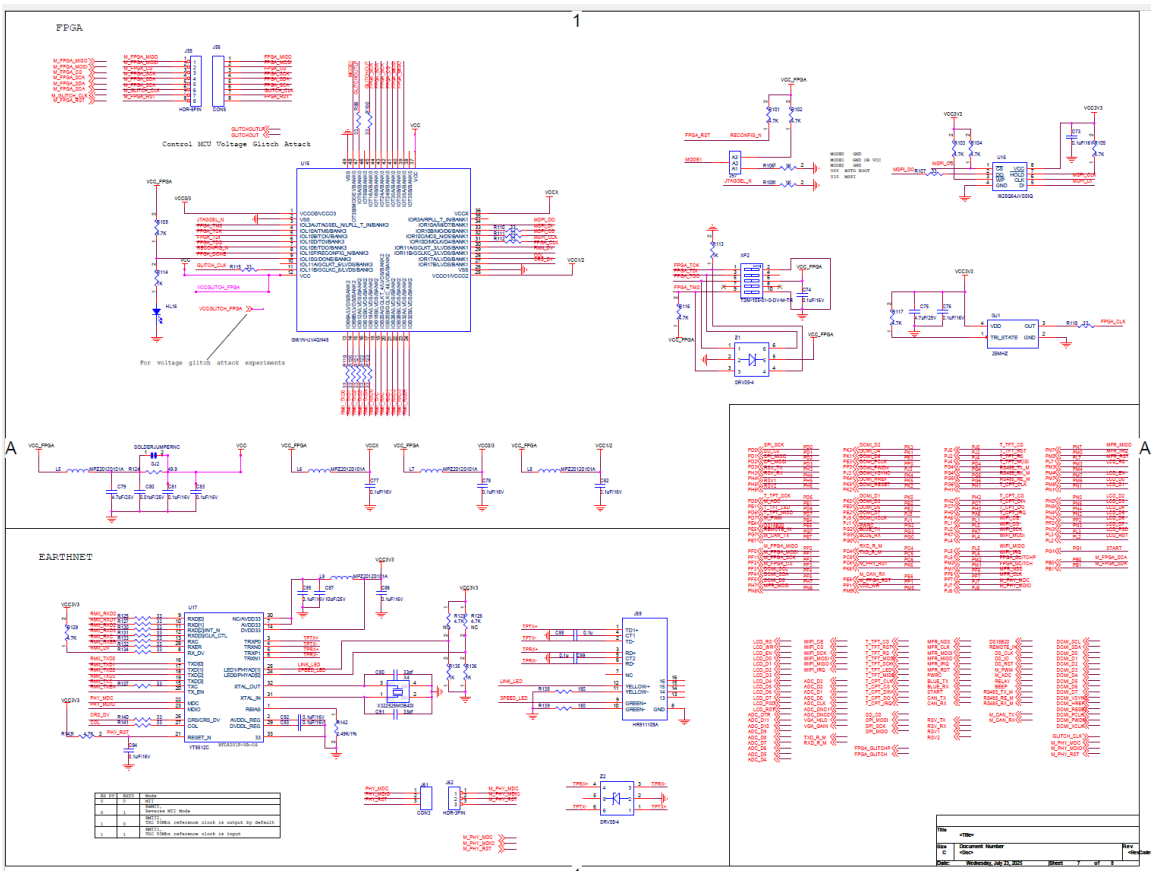


Figure 7: Development Board Schematic 4

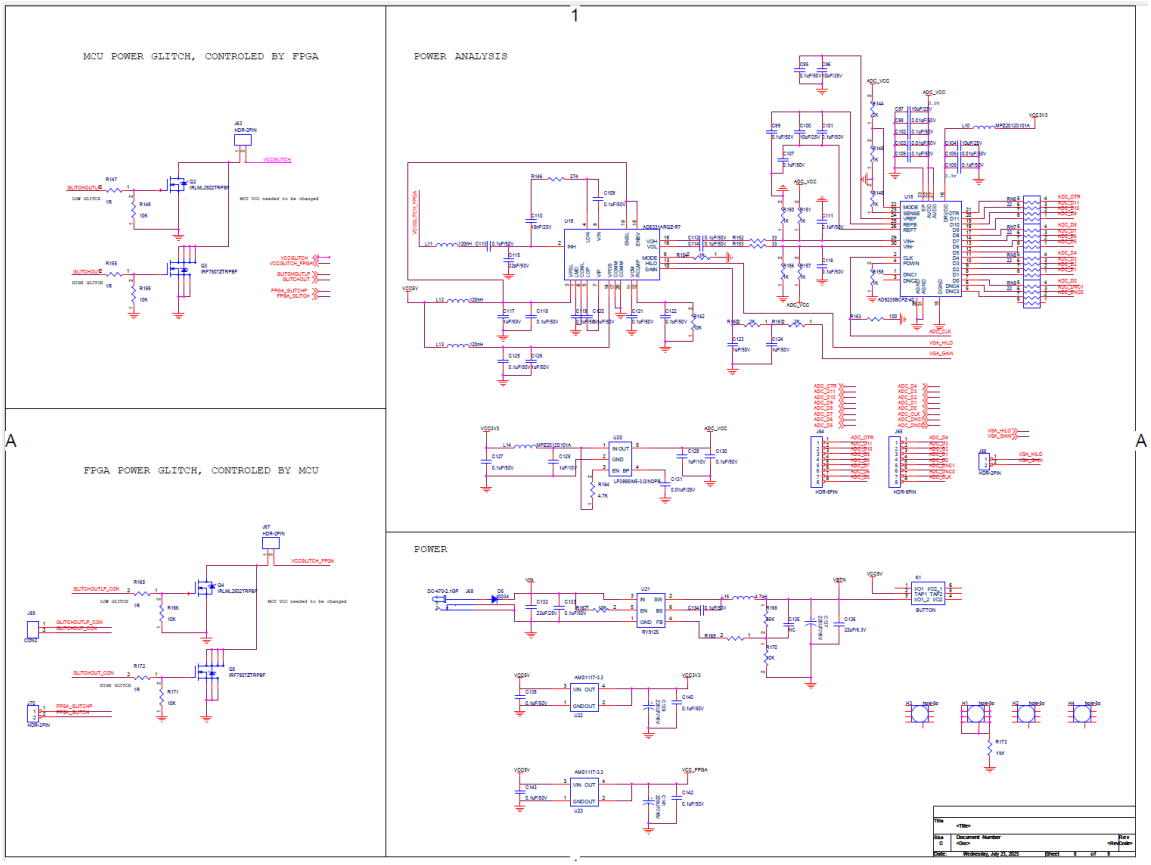


Figure 8: Development Board Schematic 5

EEDB Development Board Layout

The EEDB development board uses a single-sided layout, with all components placed on the front side. The layout is rational, concise, and aesthetically pleasing. The component arrangement is illustrated in the diagram below：

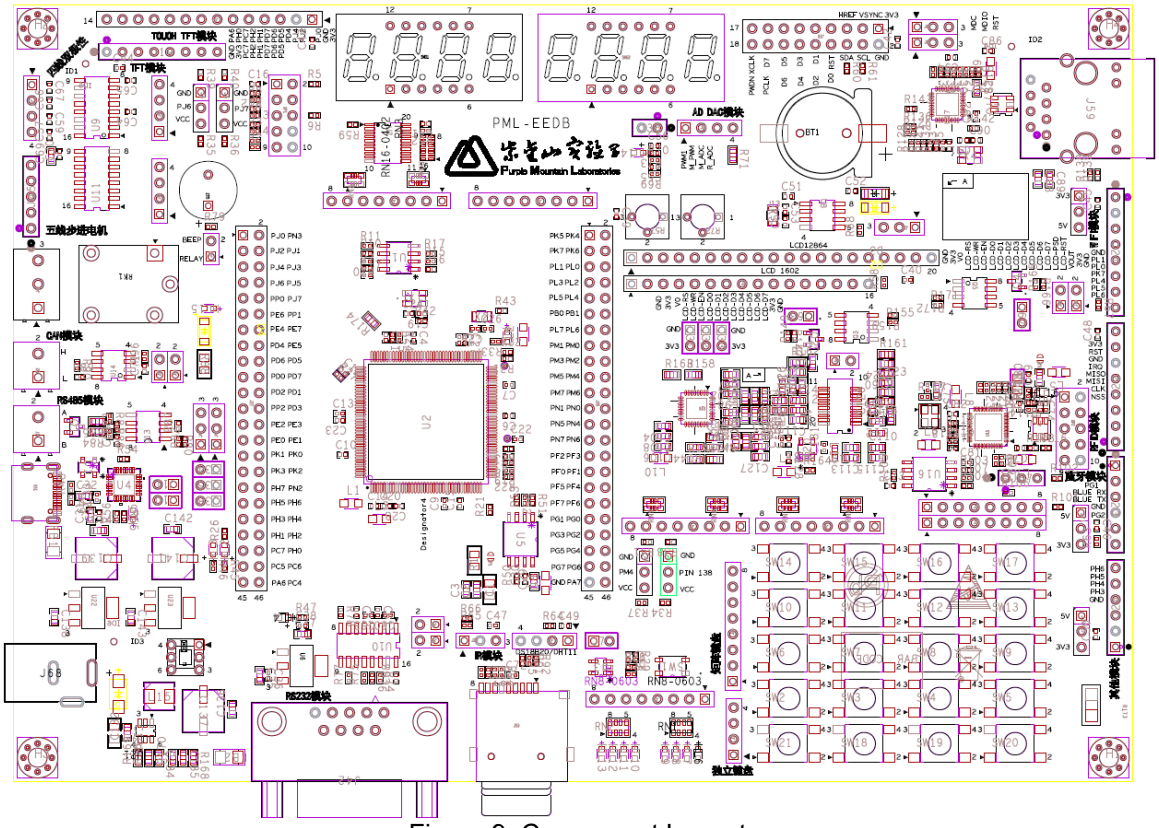


Figure 9: Component Layout

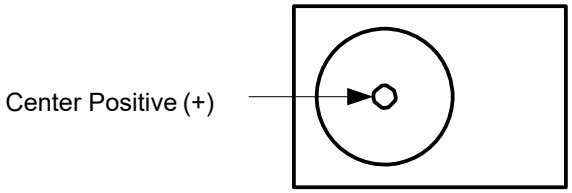
Detailed Explanation of Development Board Connections

This appendix includes the following sections:

- DC Power Interface
- Type-C Debugging Port

DC Power Interface

The development board is equipped with a DC power interface (designator: J68), which supports connection to an external regulated power supply of +7V to +17V (voltage tolerance: ±5%), providing stable and reliable power input to the system.



This interface adopts a standard DC socket specification of 5.5 mm (outer diameter) × 2.1 mm (inner pin) .

USB Interface

The development board connects to a PC via a Type-C interface (designator: XS1) to enable functions such as program downloading, online debugging, and serial communication.

Appendix1

EEDB Development Board Microcontroller ESC0830 GPIO Allocation Table.

Table 2-1 shows the pin assignments for the ESC0830microcontroller

Pin	Serial Number	Analog Function	Digital Function (GPIOPCTL)										
			1	2	3	4	5	6	7	8	9	10	11
PA0	37	-	U0Rx	-	-	-	-	-	-	CAN1Rx	-	-	-
PA1	38	-	U0Tx	-	-	-	-	-	-	CAN1Tx	-	-	-
PA2	39	-		SSI0CLK	-		-	-	-	-		-	-
PA3	40	- -		SSI0Fss SSI0Fss	-		-	-	-	-		-	-
PA4	41	-		SSI0Rx	-			-	-	-		-	-
PA5	42	-		SSI0Tx	-			-	-	-		-	-
PA6	45	-			I2C1SCL		M1PWM2					-	-
PA7	46	-			I2C1SDA		M1PWM3					-	-
PB0	97	USB0ID	U1Rx		-			-	T2CCP0	-	-	-	-
PB1	98	USB0VBUS	U1Tx		-			-	T2CCP1	-	-	-	-
PB2	99	-	I2C0SCL					-	-		-	-	-
PB3	100	-	I2C0SDA					-	-		-	-	-
PB4	136	AIN10 C0-		SSI2Clk		M0PWM2			T1CCP0	CAN0Rx	-	-	-
PB5	135	AIN11 C1-		SSI2Fss		M0PWM3			T1CCP1	CAN0Tx	-	-	-
PC0	118	-	TCK SWCLK	-		-	-	-	T4CCP0	-	-	-	-
PC1	117	-	TMS SWDIO	-		-	-	-	T4CCP1	-	-	-	-
PC2	116	-	TDI	-		-	-	-	T5CCP0	-	-	-	-
PC3	115	-	TDO SWO	-		-	-	-	T5CCP1	-	-	-	-
PC4	36	C1-	U4Rx	U1Rx		M0PWM6		IDX1	WT0CCP0	U1RTS		-	-
PC5	35	C1+	U4Tx	U1Tx		M0PWM7		PhA1	WT0CCP1	U1CTS			
PC6	34	C0+	U3Rx					PhB1	WT1CCP0	USB0EPEN ///			
PC7	33	C0-	U3Tx						WT1CCP1	USB0PFLT			
PD0	1	AIN15	SSI3ClK	SSI1Clk	I2C3SCL	M0PWM6	M1PWM0		WT2CCP0				

PD1	2	AIN14	SSI3Fss	SSI1Fss	I2C3SDA	M0PWM7	M1PWM1		WT2CCP1				
PD2	3	AIN13	SSI3Rx	SSI1Rx		M0FAULT0			WT3CCP0	USB0EPEN			
PD3	4	AIN12	SSI3Tx	SSI1Tx				IDX0	WT3CCP1	USB0PFLT			-
PD4	141	AIN7	U6Rx						WT4CCP0				-
PD5	142	AIN6	U6Tx						WT4CCP1				-
PD6	143	AIN5	U2Rx			M0FAULT0		PhA0	WT5CCP0				-
PD7	144	AIN4	U2Tx			M0FAULT1		PhB0	WT5CCP1	NMI			-
PE0	15	AIN3	U7Rx									-	-
PE1	14	AIN2	U7Tx									-	-
PE2	13	AIN1										-	-
PE3	12	AIN0										-	-
PE4	139	AIN9	U5Rx		I2C2SCL	M0PWM4	M1PWM2			CAN0Rx		-	-
PE5	140	AIN8	U5Tx		I2C2SDA	M0PWM5	M1PWM3			CAN0Tx		-	-
PE6	133	AIN21								CAN1Rx		-	-
PE7	134	AIN20	U1RI							CAN1Tx		-	-
PF0	62		U1RTS	SSI1Rx	CAN0Rx		M1PWM4	PhA0	T0CCP0	NMI	C0o	TRD2	-
PF1	63		U1CTS	SSI1Tx			M1PWM5	PhB0	T0CCP1		C1o	TRD1	-
PF2	64		U1DCD	SSI1Clk		M0FAULT0	M1PWM6		T1CCP0		C2o	TRD0	-
PF3	65		U1DSR	SSI1Fss	CAN0Tx	M0FAULT1	M1PWM7		T1CCP1			TRCLK	-
PF4	61		U1DTR			M0FAULT2	M1FAULT0	IDX0	T2CCP0	USB0EPEN		TRD3	-
PF5	60					M0FAULT3			T2CCP1	USB0PFLT		-	-
PF6	59				I2C2SCL				T3CCP0			-	-
PF7	58				I2C2SDA		M1FAULT0		T3CCP1			-	-
PG0	55				I2C3SCL		M1FAULT1	PhA1	T4CCP0			-	-
PG1	54				I2C3SDA		M1FAULT2	PhB1	T4CCP1				
PG2	53				I2C4SCL	M0FAULT1	M1PWM0		T5CCP0				
PG3	52				I2C4SDA	M0FAULT2	M1PWM1	PhA1	T5CCP1				
PG4	51		U2Rx		I2C1SCL	M0PWM4	M1PWM2	PhB1	WT0CCP0	USB0EPEN			
PG5	50		U2Tx		I2C1SDA	M0PWM5	M1PWM3	IDX1	WT0CCP1	USB0PFLT			
PG6	48				I2C5SCL	M0PWM6			WT1CCP0				

PG7	47				I2C5SDA	M0PWM7	IDX1		WT1CCP1				
PH0	32			SSI3Clk		M0PWM0		M0FAULT0	WT2CCP0				
PH1	31			SSI3Fss		M0PWM1	IDX0	M0FAULT1	WT2CCP1				
PH2	28			SSI3Rx		M0PWM2		M0FAULT2	WT5CCP0				
PH3	27			SSI3Tx		M0PWM3		M0FAULT3	WT5CCP1				
PH4	26	-	-	SSI2Clk	-	- M0PWM4	PhA0	-	WT3CCP0				-
PH5	23	-	-	SSI2Fss	-	M0PWM5	PhB0	-	WT3CCP1				-
PH6	22	-	-	SSI2Rx	-	M0PWM6	-	-	WT4CCP0				-
PH7	21	-	-	SSI2Tx	-	M0PWM7	-	-	WT4CCP1				-
PJ0	120		U4Rx						T1CCP0				
PJ1	121		U4Tx						T1CCP1				
PJ2	122		U5Rx				IDX0		T2CCP0				
PJ3	123		U5Tx						T2CCP1				
PJ4	127	C2+	U6Rx						T3CCP0				
PJ5	128	C2-	U6Tx						T3CCP1				
PJ6	129												
PJ7	130												
PK0	16	AIN16		SSI3Clk				M1FAULT0					
PK1	17	AIN17		SSI3Fss				M1FAULT1					
PK2	18	AIN18		SSI3Rx				M1FAULT2					
PK3	19	AIN19		SSI3Tx				M1FAULT3					
PK4	112		U7Rx					M0FAULT0	RTCCLK	C0o			
PK5	111		U7Tx					M0FAULT1		C1o			
PK6	110							M0FAULT2	WT1CCP0	C2o			
PK7	109							M0FAULT3	WT1CCP1				
PL0	108								T0CCP0	WT0CCP0			
PL1	107								T0CCP1	WT0CCP1			
PL2	106								T1CCP0	WT1CCP0			
PL3	105								T1CCP1	WT1CCP1			
PL4	104								T2CCP0	WT2CCP0			

PL5	103								T2CCP1	WT2CCP1			
PL6	96	USB0DP							T3CCP0	WT3CCP0			
PL7	95	USB0DM							T3CCP1	WT3CCP1			
PM0	89								T4CCP0	WT4CCP0			
PM1	88								T4CCP1	WT4CCP1			
PM2	87								T5CCP0	WT5CCP0			
PM3	86								T5CCP1	WT5CCP1			
PM4	85												
PM5	84												
PM6	83			M0PWM4					WT0CCP0				
PM7	82			M0PWM5					WT0CCP1				
PN0	81		CAN0Rx										
PN1	80		CAN0Tx										
PN2	20			M0PWM6					WT2CCP0				
PN3	119			M0PWM7					WT2CCP1				
PN4	71			M1PWM4					WT3CCP0				
PN5	70			M1PWM5					WT3CCP1				
PN6	69			M1PWM6					WT4CCP0				
PN7	68			M1PWM7					WT4CCP1				
PP0	131	AIN23	M0PWM0						T4CCP0				
PP1	132	AIN22	M0PWM1						T4CCP1				
PP2	11		M0PWM2						T5CCP0				

Appendix2

References

In addition to this document, the following reference materials can be downloaded from https://pan.baidu.com/s/1BkJC1DaaMMh6WZi0xlYREg?pwd=p7vq
using the access code: p7vq

- ESC0830 Embedded Low-Power Endogenous Security Controller Chip Datasheet
- Development Platform Software
- Driver Library

Other reference materials also include:

- LCD1602/LCD12864 documentation.
- Gowin FPGA GW1N-UV4QN48 Datasheet
- TP8485 Datasheet
- SN65HVD230DR Datasheet