

PLC从入门到精通 (上)

杨亮
恩智浦半导体 边缘处理事业部
资深客户应用方案与支持工程师

SEP 2022



SECURE CONNECTIONS
FOR A SMARTER WORLD

PUBLIC

NXP, THE NXP LOGO AND NXP SECURE CONNECTIONS FOR A SMARTER WORLD ARE TRADEMARKS OF NXP B.V.
ALL OTHER PRODUCT OR SERVICE NAMES ARE THE PROPERTY OF THEIR RESPECTIVE OWNERS. © 2022 NXP B.V.





内容摘要

什么是PLC

为什么用PLC

PLC硬件组成

PLC编译器

PLC CPU软件组成

PLC通讯组成

PLC的趋势与发展

什么是PLC



SECURE CONNECTIONS
FOR A SMARTER WORLD

PUBLIC

NXP, THE NXP LOGO AND NXP SECURE CONNECTIONS FOR A SMARTER WORLD ARE TRADEMARKS OF NXP B.V.
ALL OTHER PRODUCT OR SERVICE NAMES ARE THE PROPERTY OF THEIR RESPECTIVE OWNERS. © 2022 NXP B.V.



什么是PLC

PLC也称可编程逻辑控制器，是一个系统工程，总的来说是由硬件，软件和编译器三部分组成。

根据系统的大小，一般还会分为大中小型PLC，可能涉及MCU和MPU等嵌入式产品，



PLC系统组成

为什么用PLC



SECURE CONNECTIONS
FOR A SMARTER WORLD

PUBLIC

NXP, THE NXP LOGO AND NXP SECURE CONNECTIONS FOR A SMARTER WORLD ARE TRADEMARKS OF NXP B.V.
ALL OTHER PRODUCT OR SERVICE NAMES ARE THE PROPERTY OF THEIR RESPECTIVE OWNERS. © 2022 NXP B.V.



为什么用PLC

随着MCU/CPU技术的不断发展，不少人会拿它和PLC做对比，PLC从成本和性能上都并不占优势，但为什么工业现场还更热衷于使用PLC？

1. 从产品的设计角度讲，PLC的设计厂商有多年的行业积累，在设计上有高可靠性的保障，同时还会加入相对完善的测试，比如硬件上有板级的时序测试(DDR, USB, Flexbus)，环境/EMC测试，软件上有黑白盒测试，使之在可靠性，抗干扰方面有一定优势
2. 从产品易用性角度讲，MCU/CPU的开发还是相对复杂，不仅仅是对语言的掌握，还有对开发环境的要求。梯形图/FBD这种编程方式要比C语言要更容易上手，而且自动化从业人员更为熟悉
3. 从产品灵活性角度讲，PLC可以非常灵活的进行组合，针对不同的应用场景可以很容易搭建起原形，并且有些PLC编译器支持将PLC工程导出为C文件，从而制作专用控制器



PLC硬件组成



SECURE CONNECTIONS
FOR A SMARTER WORLD

PUBLIC

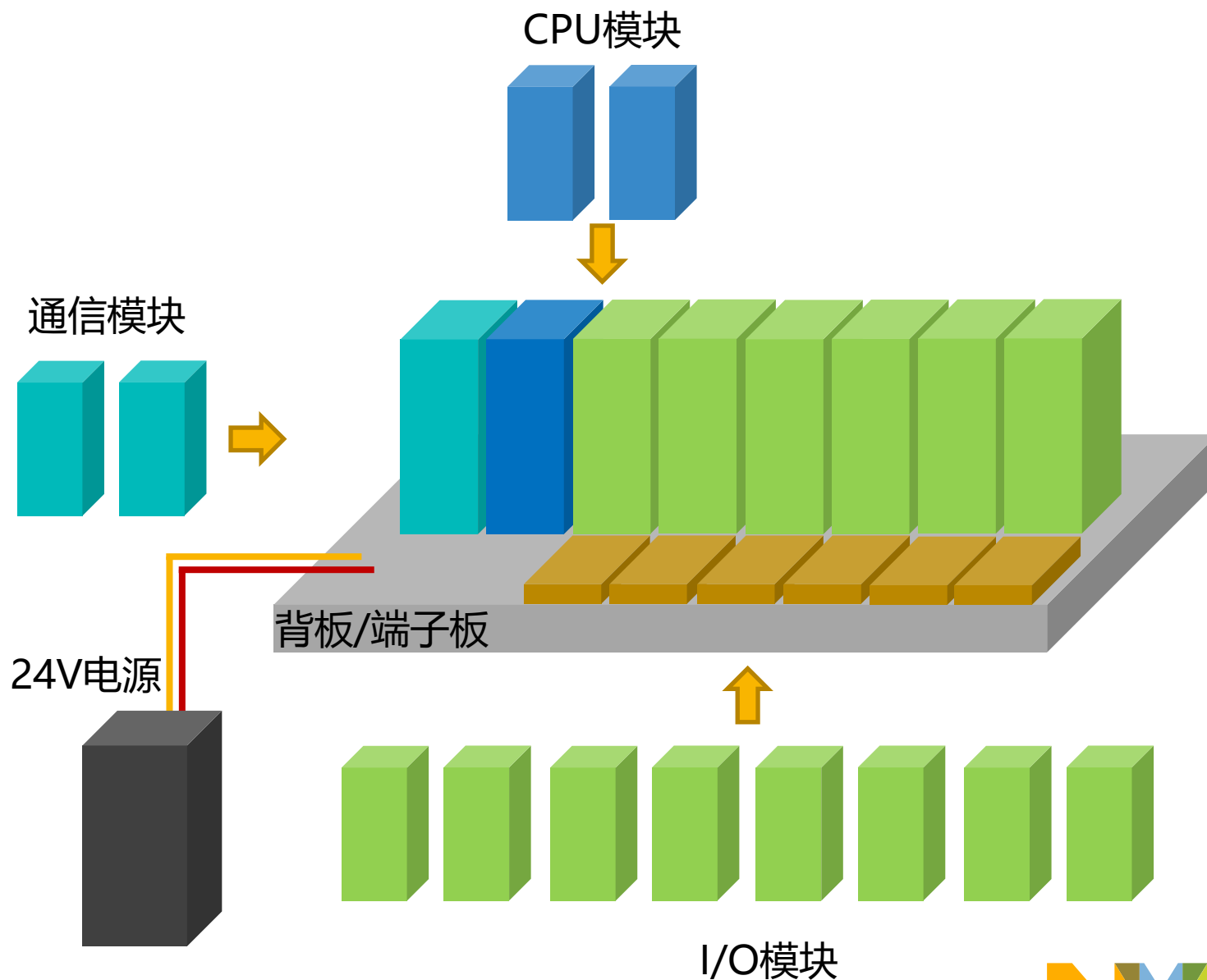
NXP, THE NXP LOGO AND NXP SECURE CONNECTIONS FOR A SMARTER WORLD ARE TRADEMARKS OF NXP B.V.
ALL OTHER PRODUCT OR SERVICE NAMES ARE THE PROPERTY OF THEIR RESPECTIVE OWNERS. © 2022 NXP B.V.



PLC硬件组成

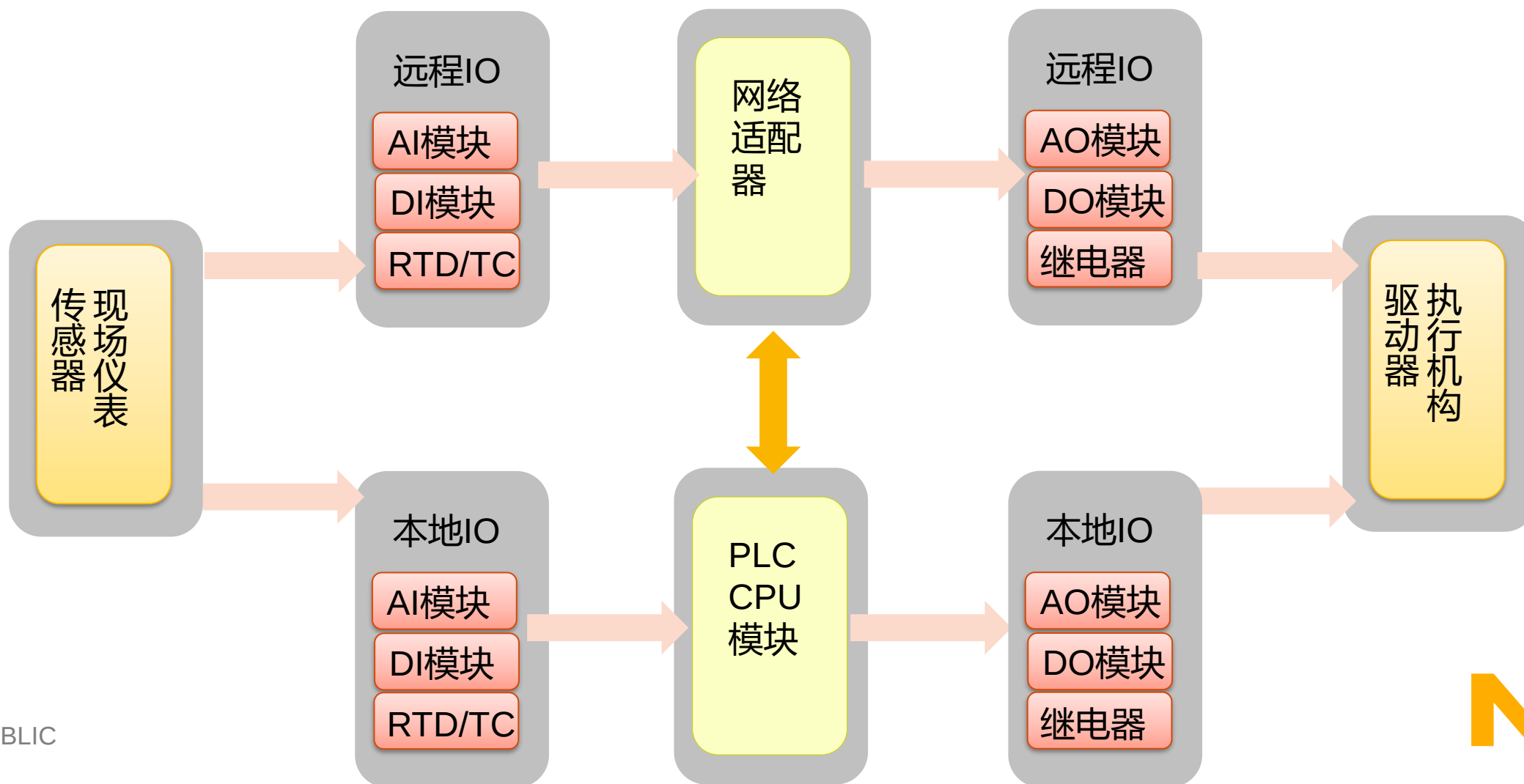
PLC硬件一般由以下几部分组成：

- 24VDC开关电源
- CPU模块
- I/O模块
- 通信模块/适配器
- 背板/端子板



PLC硬件组成-总体结构

PLC一般都是通过背板通讯的方式从输入模块获取现场的信号，经过用户程序运算后将需要输出的结果通过背板总线发送给输出模块，从而驱动执行机构做出相应的动作



PLC硬件组成-I/O模块

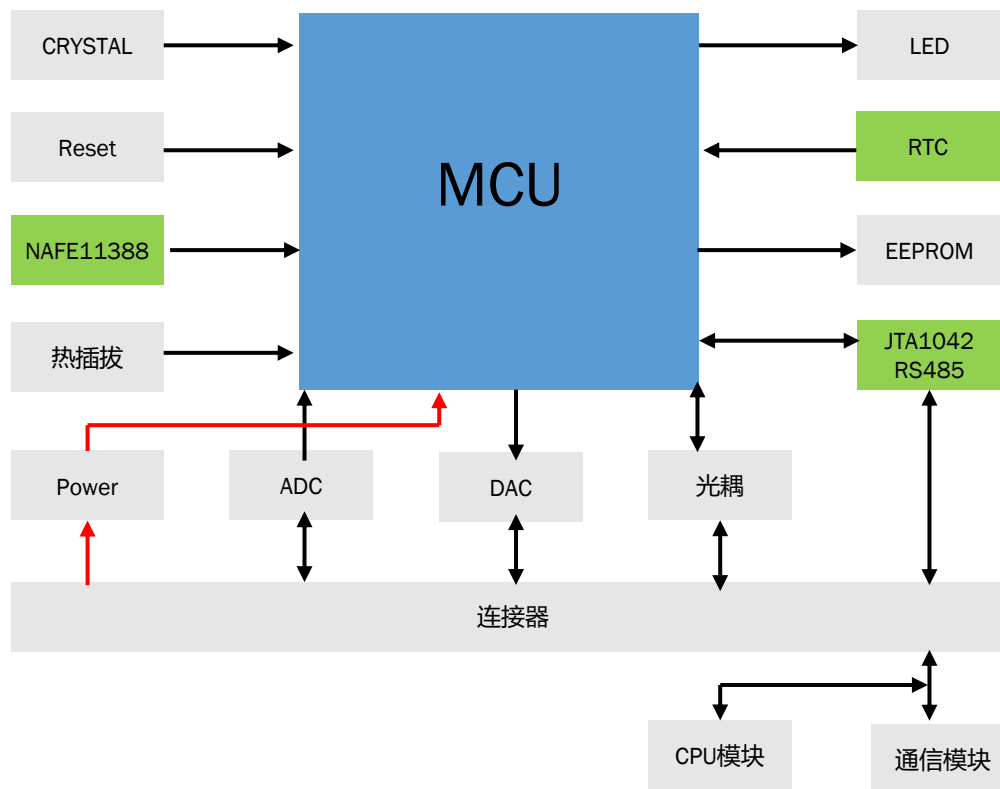
I/O模块通过端子板接线的方式与现场传感器/执行器相连，并通过背板总线与PLC/适配器相连，其内部常见的通信方式有RS485、CAN等，下面列出了常见的IO模块及常见的参数

模块类型	参数示例
数字量输入	漏型：24VDC；源型：0VDC
模拟量输入	电压 / 电流型， $\pm 10\text{V}/0 \sim 5\text{V}/0 \sim 10\text{V}/0 \sim 20\text{mA}/4 \sim 20\text{mA}$
数字量输出	晶体管：10 ~ 30VDC，容量 0.5A；继电器：最大切换电压 250VAC/220VDC
模拟量输出	电压型， $-10.25\text{V} \sim 10.25\text{V} / 0 \sim 10.25\text{V} / 0 \sim 5.125\text{V}$
热电阻	PT100/200/500/1000，Ni100/120/200/500，Cu10/50
热电偶	B/C/E/J/K/N/R/S/T 型热电偶， $-12\text{mV} \sim 32\text{mV}(78\text{mV})$ ，带冷端补偿



PLC硬件组成-I/O模块

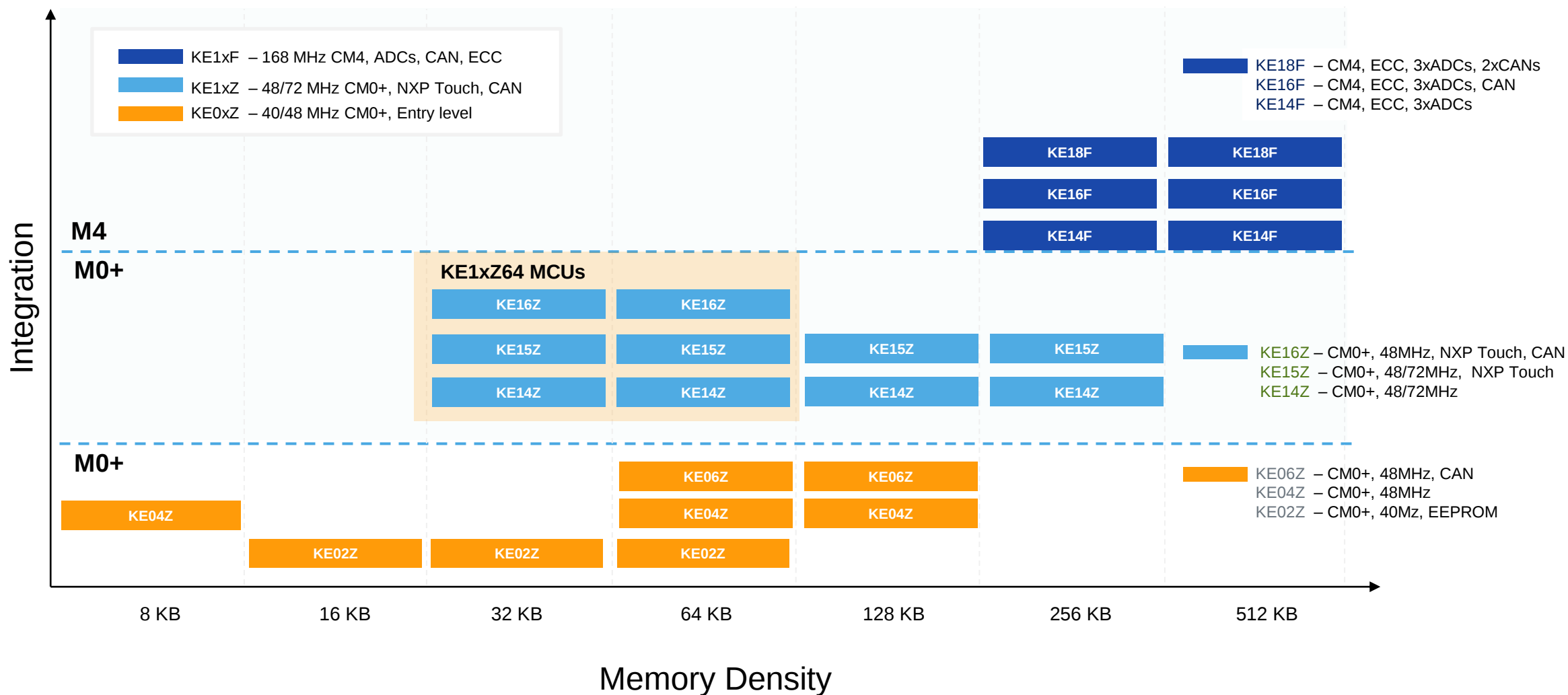
I/O模块所需要的MCU功能相对简单，通常需要通过LED显示当前模块状态，非易失性存储器保存校准值或故障信息，背板接口来进行通讯



PLC硬件组成-I/O模块-NXP型号推荐

MCU,	MKE14Z (48或72Mhz CM0+, 5V供电, FlexIO灵活接口)
	MKV10/11 (75Mhz CM0+, 2*16位ADC, 硬件除法器)
	LPC550x (96Mhz CM33, 16位ADC, 接口丰富, 安全增强型)
16位/24位 ADC,	NAFE13388,
Pre-driver,	MC33GD3000
Low power RTC,	PCF85063A
Full feature RTC,	PCF2131
GPIO expander,	8bit PCAL6408A/ 16bit PCA9555A
CAN transceiver,	TJA1057/ 1043/ 1052
I/OLINK-Master,	MC34CM3120EP,
Level shifter,	PCA9306/ PCA9548A/ NTS0304E
Load Switches,	NX20P3483
High side switch,	MC33XS2410
Low side switch,	MC33999
PMIC, IMX6/ IMX7/ IMX8,	MC34PF1510, PCA9450

KE 系列MCU产品



KV1X 系列MCU产品

Core/System

- 75MHz Cortex-M0+ with Hardware Divide & Square Root
- 4ch DMA

Memory

- 16/32/64/128KB Flash**
- 8/16KB SRAM**
- Option with FAC

Communications

- Multiple serial ports + **1 FlexCAN (仅限KV11)**

Analog

- 2 x 8ch 16-bit ADC
 - 1.2MSPS in 12-bit mode (835ns)
- 1 x12-bit DAC
- 2 x ACMP with 6-bit DAC

Timers

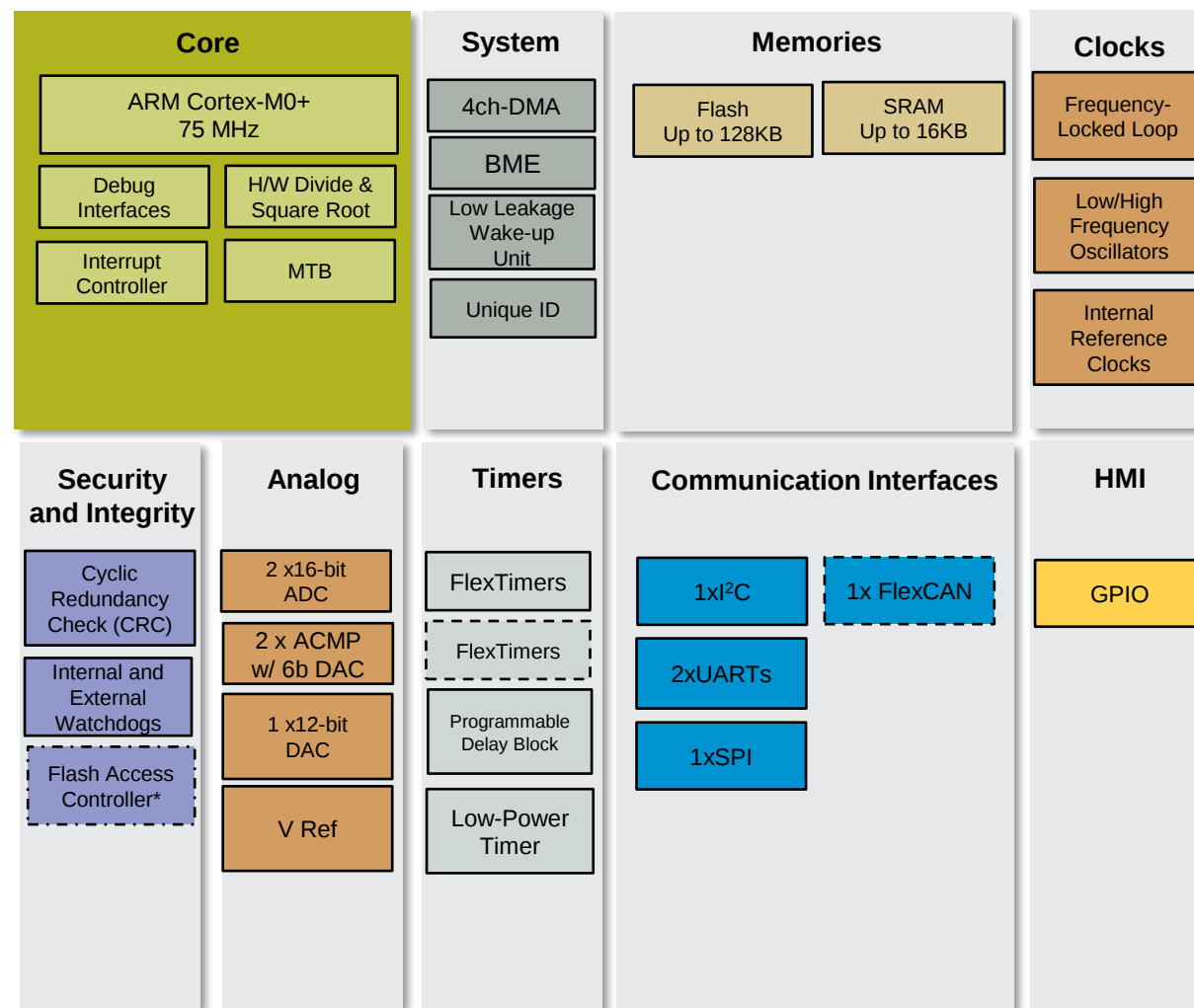
- Up to 2x6ch FlexTimer (PWM) ***
- Up to 4x2ch FlexTimer (PWM/Quad Dec.)
- Low Power Timer

Other

- 32-bit CRC
- Up to 54 GPIO
- 1.71V-3.6V; -40 to 105C

Packages

- 32QFN, *32LQFP, 48LQFP, 64LQFP

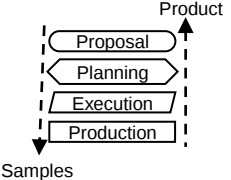


KV1X 系列特点与优势

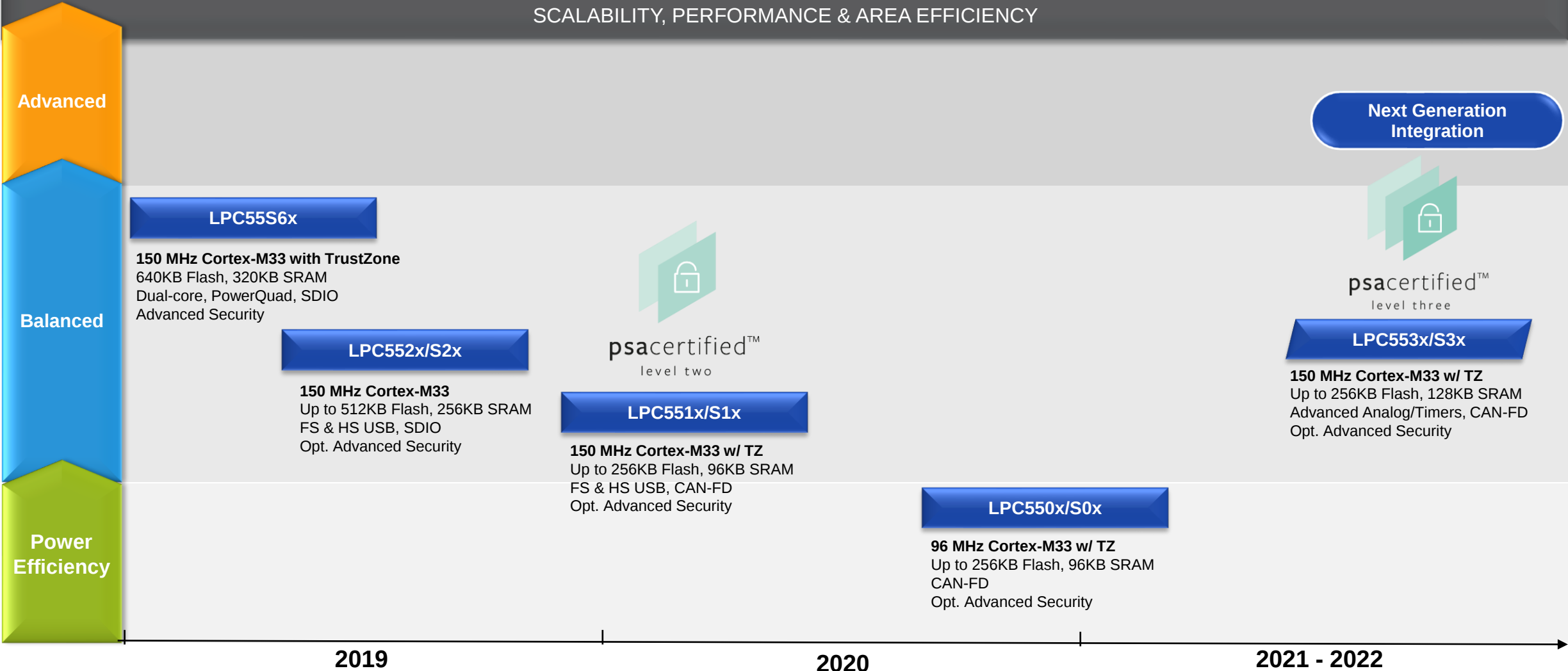
Features	Benefits
Cortex M0+ @75MHz	Fastest Cortex M0+ in the market enables PMSM motor control with a M0+ solution
Hardware Square Root & Divide Hardblock	More than 25% performance improvement running math intensive applications such as Sensorless PMSM FOC algorithms
Dual ADC Blocks sampling @ 1.2MS/s in 12b mode	Capture current & voltage simultaneously for the most accurate result
4ch DMA	Further improvements in performance realized through increased CPU bandwidth-
6ch FlexTimer + 2x2ch FlexTimer	Motor control PWM generation with integrated PFC, or integrated speed sensor decoder (incremental decoder / hall sensor)
Optional Additional 6ch FlexTimer	Capability to operate a 2 nd motor from a single MCU
Integrated 6b DAC & CMP	Reduce BOM costs with integrated components for over current over voltage fault detection
Peripheral Interconnection	ADC and CMP interconnected with PWM and PDB for real time hardware control.
Light weight peripheral and memory configuration	Enough performance for the majority of Motor Control applications, with the right amount of memory to fit complex motor control algorithms, FlexCAN available for extended communications
Dual Watchdog	IEC60730 Compliant solution

LPC5500 MCU SERIES ROADMAP

Arm CM7 Arm CM33 Arm CM4



COMMON PLATFORM ARCHITECTURE SCALABILITY, PERFORMANCE & AREA EFFICIENCY



Next Generation Integration



psacertified™
level three

LPC553x/S3x

150 MHz Cortex-M33 w/ TZ
Up to 256KB Flash, 128KB SRAM
Advanced Analog/Timers, CAN-FD
Opt. Advanced Security

2019

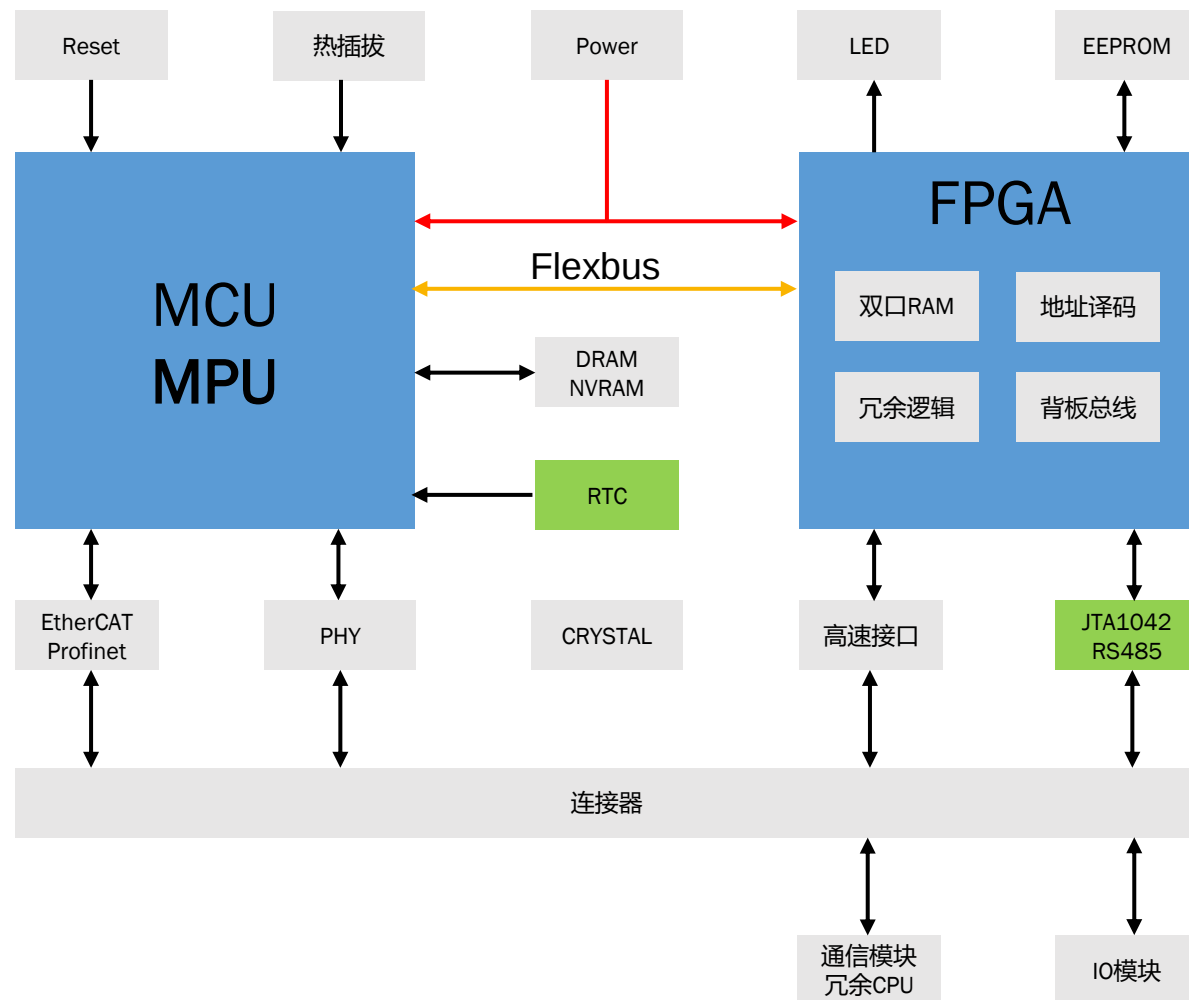
2020

2021 - 2022

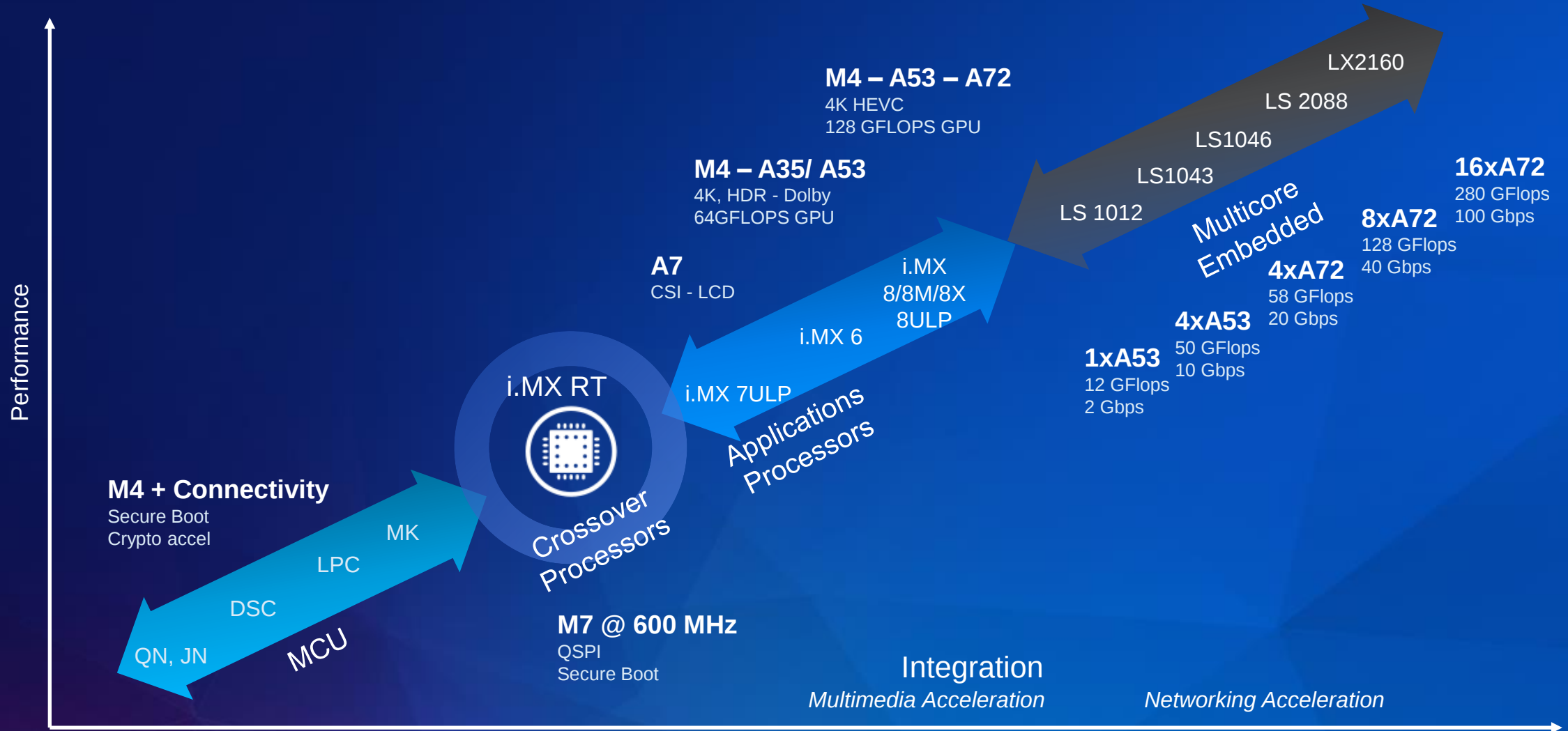
PLC硬件组成-CPU模块

PLC根据应用规模可以分为中大型PLC和小型PLC，中大型PLC通常选择MPU作为核心处理并搭配FPGA协同处理，小型PLC一般单独选择MCU作为核心处理。其中MCU/MPU主要负责用户程序的运算，FPGA主要起以下作用：

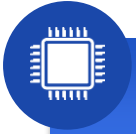
- 对所有外部总线设备进行译码
- 驱动并实现高速总线，实现同步数据的交互
- 内建双口RAM，实现通信模块与CPU之间进行交互
- 运动控制算法，例如插补，编码器解析
- IP扩展，比如高速UART（12Mbps以上），SPI等
- 控制外部ADC进行同步采样，并进行简单的处理



NXP EDGE PROCESSING PRODUCTS



小型PLC的CPU模块适用的NXP型号-ARM核



Performance & Integration

i.MX RT Crossover MCUs for Ultimate Real-time Performance & Integration



Performance Efficiency

Kinetis K, LPC54000 MCUs & **LPC5500** for Performance Efficiency



Power Efficient

Kinetis L & **K32 L MCUs** for Cost-, Power-Sensitive Applications, **LPC55**



Value

S08 (8-bit) and **LPC800 (32-bit)** MCUs for Entry-Level Applications



Robust Industrial MCU

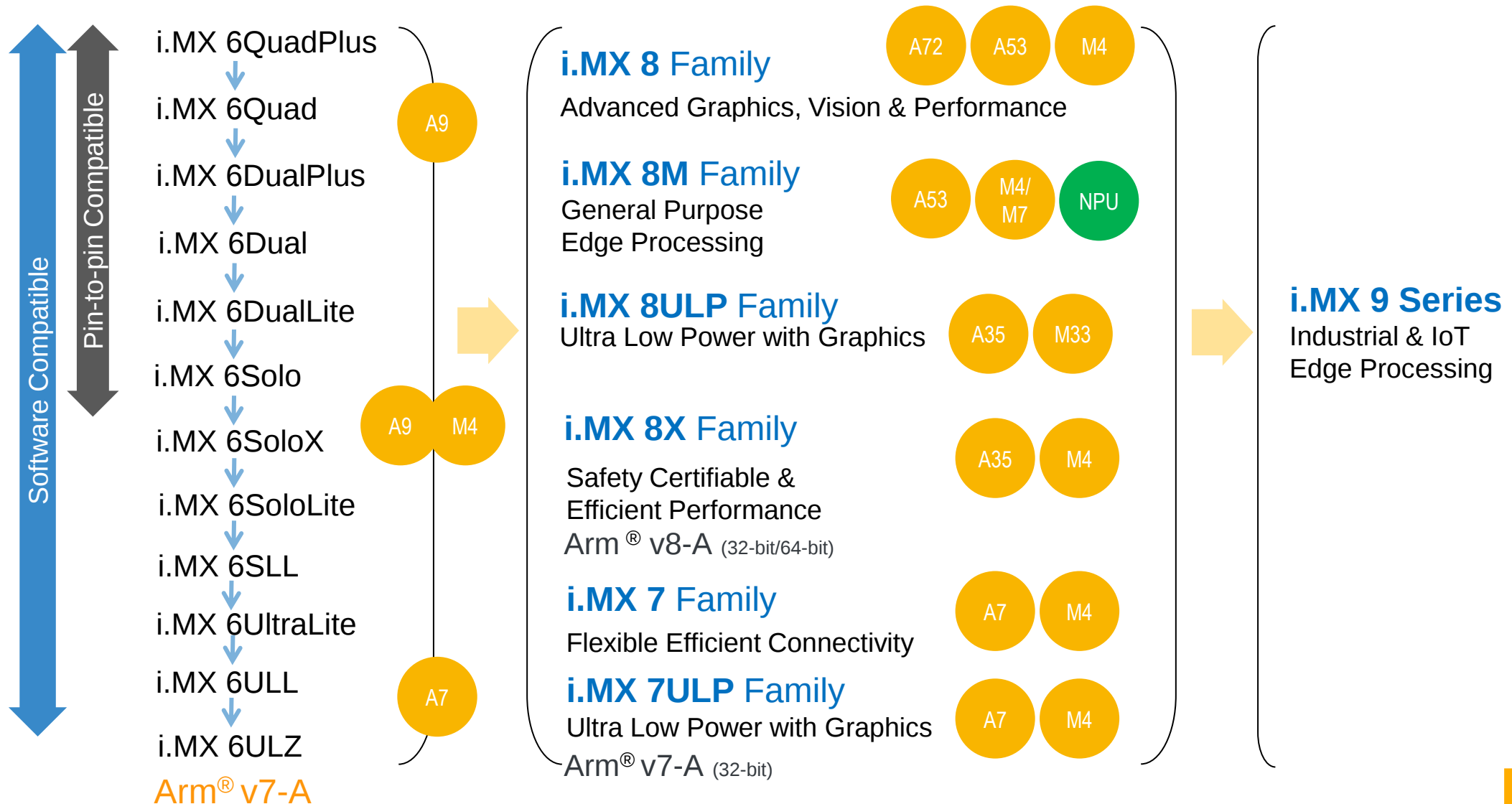
KE (Industrial) & **S32K (Automotive)** for Safe and Reliable 5V MCU Applications
DSC/ KV for Motor Control & Digital Power; & **KM** for Metering Applications

小型PLC的CPU模块适用的NXP型号-RT1000

800+MHz Premium					i.MX RT1160 Cortex-M7, 600Mhz Cortex-M4, 240Mhz 2MB SRAM 8/16/32-bit EMI (SDRAM/SRAM) MIPI CSI/DSI / 2D acceleration Automotive, Advanced Security 289BGA	i.MX RT1170 Cortex-M7, 32K/32K L1 Cortex-M4, 16K/16K L1 2MB SRAM 8/16/32-bit EMI (SDRAM/SRAM) MIPI CSI/DSI / 2D acceleration Automotive, Advanced Security 289BGA	i.MX RT1180 Cortex-M7, 32K/32K L1 Cortex-M4, 16K/16K L1 Advanced timers Advanced Security ECAT slave + TSN Switch 289BGA
	i.MX RT1050 Cortex-M7, 32K/32K L1 512KB SRAM 8/16-bit EMI (SDRAM/SRAM) LCD / CSI / 2D acceleration Standard Security 196BGA	i.MX RT1060 Cortex-M7, 32K/32K L1 1MB SRAM 8/16-bit EMI (SDRAM/SRAM) LCD / CSI / 2D acceleration Standard Security 196BGA 225BGA	i.MX RT1064 RT1060 + 4MB QSPI Flash 196BGA	i.MX RT1040 Cortex-M7, 32K/32K L1 512KB SRAM, 125C 8/16-bit EMI (SDRAM/SRAM) LCD / 2D acceleration Standard Security 169 BGA			
	i.MX RT1020 Cortex-M7, 16K/16K L1 256KB SRAM 16-bit EMI (SDRAM/SRAM) Standard Security ENET, CAN, USB 144LQFP 100LQFP	i.MX RT1015 Cortex-M7, 16K/16K L1 128KB SRAM Flex SPI Standard Security USB 100LQFP	i.MX RT1010 Cortex-M7, 16K/8K L1 128KB SRAM Flex SPI Standard Security USB 80LQFP	i.MX RT1024 RT1020 + 4MB QSPI Flash 144LQFP			

2021

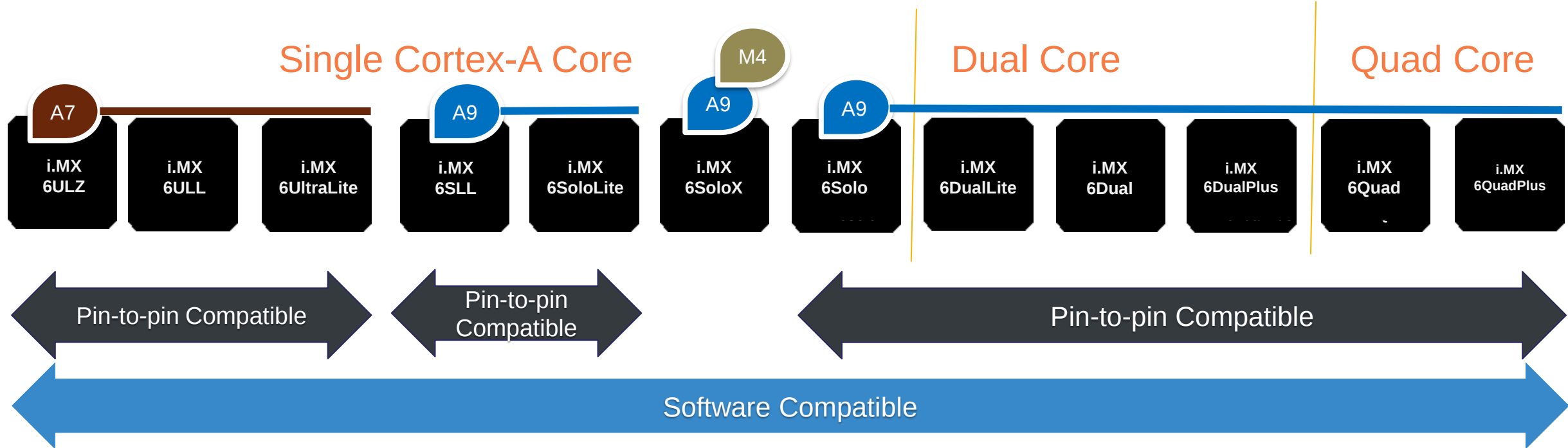
中大型PLC的CPU模块/显示模块适用的NXP型号 – IMX产品



i.MX 6 Series: Supreme Scalability and Flexibility

Leverage One Design into Diverse Product Portfolio

Scalable series of **Twelve** Arm-based SoC Families



Expanded series for performance, power efficiency and lower BOM

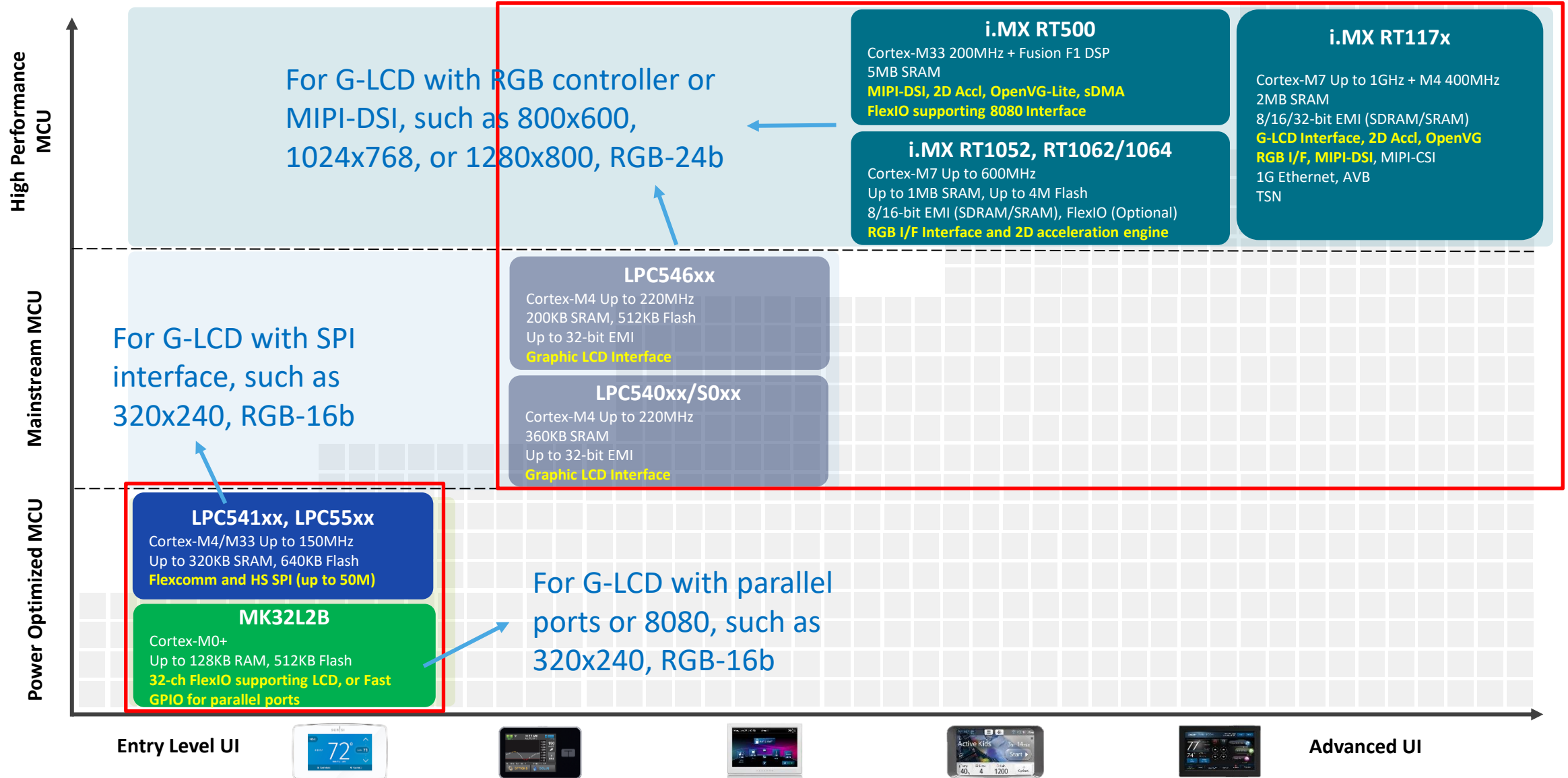
i.MX 8M Family for Embedded C&I Applications

Product	i.MX 8M Quad / QuadLite	i.MX 8M Mini / Mini Lite	i.MX 8M Nano / Nano Lite	i.MX 8M Nano UltraLite	i.MX 8M Plus
Sample / Production	Production Website – www.nxp.com/imx8m	Production Website – www.nxp.com/imx8mini	Production Website – www.nxp.com/imx8mnano	Production www.nxp.com/imx8mnano	Production Website – www.nxp.com/imx8mplus
Main CPU	2x or 4x A53 1.5GHz, 1MB L2	1x, 2x or 4x A53 1.8GHz , 512KB L2	1x, 2x or 4x A53 1.5GHz, 512KB L2	1x, 2x or 4x A53 1.4GHz, 512KB L2	2x or 4x A53 1.8-2GHz, 512KB L2
MCU/DSP	M4 266MHz	M4 400MHz	M7 up to 750MHz	M7 up to 750MHz	M7 800MHz, HiFi4 800 MHz
DDR	x16 or x32 LPDDR4/DDR4/DDR3L	x16 or x32 LPDDR4/DDR4/DDR3L	x16 LPDDR4/DDR4/DDR3L	x16 LPDDR4/DDR4/DDR3L	x16 or x32 LPDDR4/DDR4/DDR3L Inline ECC
GPU	3D – GC7000Lite (4 shaders) (OpenGL® ES 2.1/3.0/3.1, OpenCL™ 1.2, Vulkan)	2D – GC320 3D – GC NanoULTRA (OpenGL® ES 2.1)	GC7000UL (OpenGL® ES 2.1/3.0/3.1, OpenCL™ 1.2, Vulkan)	-	2D - GC520L 3D – GC7000UltraLite (OpenGL® ES 2.1/3.0/3.1, OpenCL™ 1.2, Vulkan)
Security	CAAM, RDC, TrustZone	CAAM, RDC, TrustZone	CAAM, RDC, TrustZone	CAAM, RDC, TrustZone	CAAM, RDC, TrustZone
AI/ML	A53, GPU (OpenCL)	A53	A53, GPU (OpenCL)	A53	ML Accel 2+ TOPS
SRAM	128KiB + 32KiB	256KiB + 32KiB	512KiB + 32KiB	512KiB + 32KiB	768KiB + 32KiB
Camera	2x MIPI CSI (4-lane)	1x MIPI CSI (4-lane)	1x MIPI CSI (4-lane)	1x MIPI CSI (4-lane)	2x MIPI CSI (4-lane), ISP 2 camera
Display	HDMI 2.0a Tx, MIPI DSI (4-lane), eDP	1x MIPI DSI (4-lane)	1x MIPI DSI (4-lane)	-	HDMI 2.0a Tx (eARC), MIPI DSI (4-lane), 1x LVDS (8-lane)
OSD Overlay	4Kp60	1080p60	1080p60	-	1080p60
HDR	HDR10, HLG, Dolby Vision	None	None	None	None
Video Decode	4Kp60 HEVC , VP9, 4Kp30 H.264, legacy codecs	1080p60 HEVC, H.264, VP9, VP8	None	None	1080p60 H.265, H.264, VP9, VP8
Video Encode	No HW acceleration	1080p60 H.264, VP8	No HW acceleration	No HW acceleration	1080p60 H.265 , H.264
Connectivity	PCIe, SDIO, USB	PCIe, SDIO, USB	SDIO, USB	SDIO, USB	PCIe, SDIO, USB
Audio	20x I2S TDM (32b @384KHz), S/PDIF Tx+Rx	20x I2S TDM (32b @384KHz), 8ch PDM DMIC input), S/PDIF Tx+Rx	12x I2S TDM (32b @384KHz), ASRC , 8ch PDM DMIC input), S/PDIF Tx+Rx	12x I2S TDM (32b @384KHz), ASRC , 8ch PDM DMIC input), S/PDIF Tx+Rx	18x I2S TDM (32b @384KHz), ASRC , 8ch PDM DMIC input), S/PDIF Tx+Rx
Expansion I/O	2x USB3.0, 2x PCIe Gen 2	2x USB2.0, 1x PCIe Gen 2	1x USB2.0	1x USB2.0	2x USB 3.0 Type C, 1x PCIe Gen 3
Network, Storage	1x Enet, 2x SD/eMMC, MLC/SLC NAND	1x Enet, 3x SD/eMMC, MLC/SLC NAND	1x Enet, 3x SD/eMMC, MLC/SLC NAND	1x Enet, 3x SD/eMMC, MLC/SLC NAND	1x Enet, 1x TSN Enet , 2x CAN-FD, 3x SD/eMMC, MLC/SLC NAND
Process	28nm	14nm FinFET	14nm FinFET	14nm FinFET	14nm FinFET
Package	17x17mm, 0.65p (no microvias)	14x14mm, 0.5p (no microvias)	14x14mm, 0.5p (no microvias)	11x11mm, 0.5p (no microvias)	15x15mm, 0.5p (no microvias)

In Production

In Progress

RECOMMENDED MCUS FOR GRAPHICS



* Select NXP devices also support segment LCDs

† Recommended platforms for graphics applications – does not include all device families that are enabled for graphics applications

图形显示推荐的MCU家族

		Core	Frequency	Memory	Graphics Acceleration	Display Interface / Controller	Resolutions	
Power Optimized		LPC541xx, MK32L2B, etc	Cortex-M0+/M4	Up to 100 MHz	Up to 192KB RAM, 512KB Flash	—	SPI or 8080 parallel with FlexIO or Fast GPIO of M0+	QVGA 320x240
		LPC55xx	Cortex-M33	150 MHz	320 KB RAM, 640 KB Flash	—	SPI	Up to FWVGA 854x480
		i.MX RT101x/102x	Cortex-M7	Up to 500MHz	Up to 256KB RAM	—	SPI or 8080 parallel with FlexIO	QVGA 320x240
Mainstream		LPC54S/540xx	Cortex-M4	180 MHz	360 KB RAM Up to 32-bit EMI	—	LCDIF	Up to XGA 1024x768
		LPC546xx	Cortex-M4	220 MHz	512 KB Flash, 200 KB RAM Up to 32-bit EMI	—	LCDIF	Up to XGA 1024x768
		i.MX RT105x	Cortex-M7	600 MHz	512 KB SRAM 8/16-bit EMI	PXP	eLCDIF	Up to WXGA 1366x768
High Performance		i.MX RT106x	Cortex-M7	600 MHz	512 KB SRAM 8/16-bit EMI	PXP	eLCDIF	Up to WXGA 1366x768
		i.MX RT500	Cortex-M33 and Fusion F1 DSP	200 MHz / 200 MHz	5 MB SRAM 2x Quad/Octal Mem Interface	GPU	MIPI-DSI	Up to XGA 1024x768
		i.MX RT1170	Cortex-M7 and Cortex-M4	1 GHz / 400 MHz	2 MB SRAM 2x Quad/Octal Mem Interface 8/16/32-bit EMI	PXP GPU	eLCDIF LCDIFv2 MIPI-DSI	Up to WXGA 1366x768

Note: LCDIF, eLCDIF, and LCDIFv2 include parallel RGB display interface

对GUI工具的支持

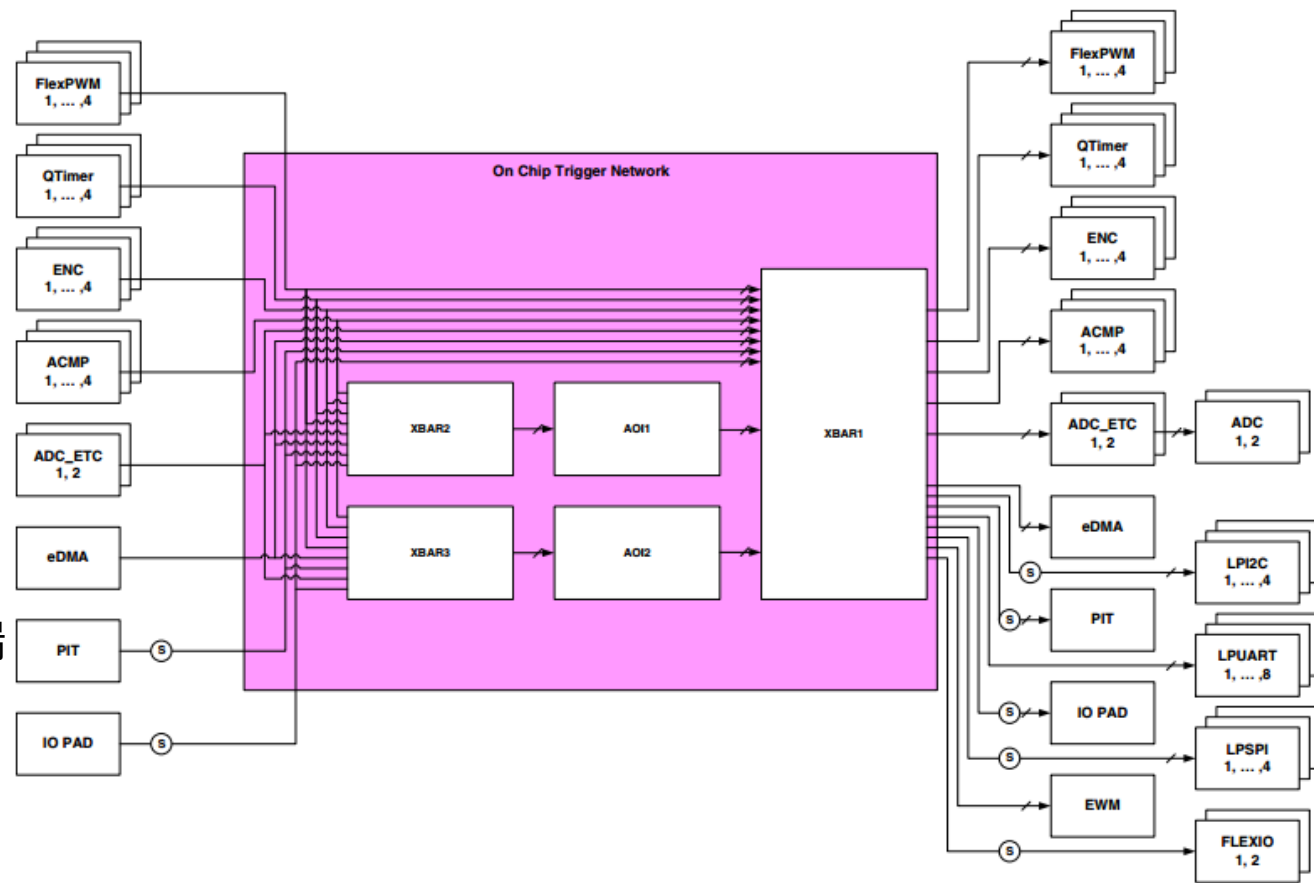
Provider / Product	Language	GUI builder tools	Business model	RTOS required?	Support LPC54xxx/55xx, RT101x/102x	Support RT105x/106x, RT117x, RT500
AWTK	C	AWTK Designer	Open source to end customer	Optional (RTOS)	✓	✓
Crank/Story Board*	C/C++	Crank Software	Developer seats, volume based product line license	Optional (any)		✓
LVGL*	C/C++	Visual Studio, Eclipse, NXP Builder	Open source, free	Optional (any)	✓	✓
MicroEJ	C/C++/Java	NA	Developer seat licenses, volume based licenses	Yes (MicroEJ)		✓
QT for MCU	C++	QT Design Studio, QT Creator	Three types of license: GPL (open source & free), LGPL (free), and Commercial	Optional (any)		✓
RT-Thread/Persimmon UI	C/C++	Persimmon UI	Developer seat licenses, volume based licenses	Yes (RT-Thread)	✓	✓
SEGGER / emWIN*	C	GUI Builder, AppWizard	Free/no royalty object (via NXP), per product source license available from SEGGER	Optional (any)	✓	✓
TARA / Embedded Wizard*	C/Javascript	Embedded Wizard Studio	Developer seats, volume based product line license	Optional (any)	✓	✓

* Demos are available in NXP MCUXpresso SDK.

PLC硬件组成-CPU模块-PTO/HCI

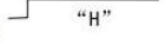
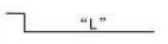
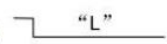
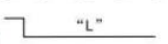
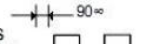
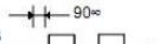
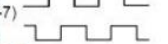
对于支持运动控制的PLC而言，脉冲输出和高速脉冲计数是比较重要的两个功能。早期这两个功能都会方在FPGA上去处理，由于i.MX RT 系列MCU内置有非常灵活的定时器，使其可以完美支持该功能，相关的模块如下：

- i.MX RT Quadrature Encoder/Decoder (ENC)
- i.MX RT Quad Timer (TMR)
 - 计数外部输入信号的宽度
 - 上升沿，下降沿及双边沿
 - 解码正交编码信号AB相
 - 方向加脉冲的输入信号解码
 - 固定数量脉冲的输出
- i.MX RT eFlexPWM
 - 输出指定频率或者90度相位差的多路PWM时钟信号
- i.MX RT CorssBar(XBAR)
 - 通过XBAR能够将指定的PWM信号输入至TMR的input端



PLC硬件组成-CPU模块-HCI

类型	实现方式
符号加脉冲	可以使用TMR的Signed-Count Mode
CW+CCW脉冲	可以使用TMR模块完成
正交解码	1. 可以使用专用的ENC模块 2. 可以使用TMR模块的Quadrature-Count Mode

用户常数 Pn200.0	指令脉冲形态	输入倍增	原理	电机正转指令	电机反转指令
0	符号 + 脉冲列	-	正原理	PULS (CN1-7)  SIGN (CN1-11)  "H"	PULS (CN1-7)  SIGN (CN1-11)  "L"
1	CW脉冲+CCW脉冲	-		PULS (CN1-7)  "L" SIGN (CN1-11) 	PULS (CN1-7)  SIGN (CN1-11)  "L"
2	90° 位相差 2 相脉冲	×1		 PULS (CN1-7)  SIGN (CN1-11) 	 PULS (CN1-7)  SIGN (CN1-11) 
3		×2			
4		×4			

Functional mode:

- Quadrature-Count Mode with Index Input:
Count encoder signal (A/B/Z)
- Count Mode: for speed measurement.
- Signed-Count Mode:
Count external pulse/direction signal.
- Pulse-Output
output a pulse stream of pulses that has the same frequency of the selected clock source

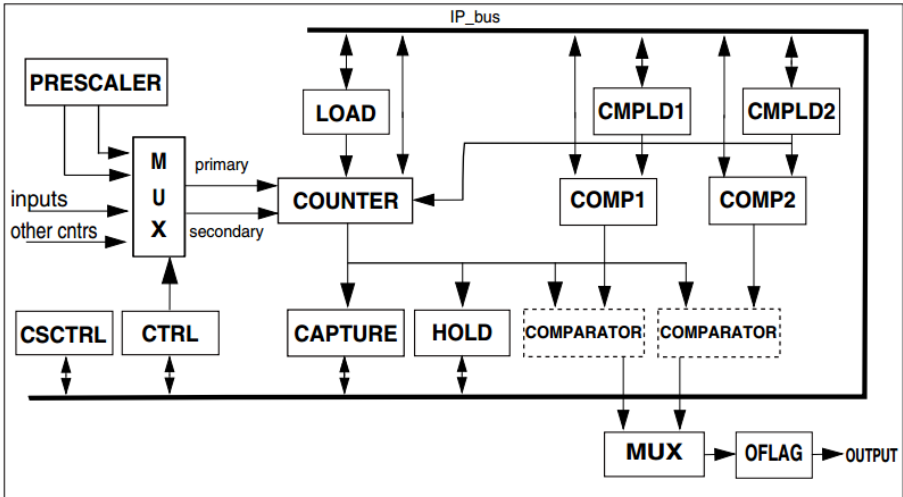
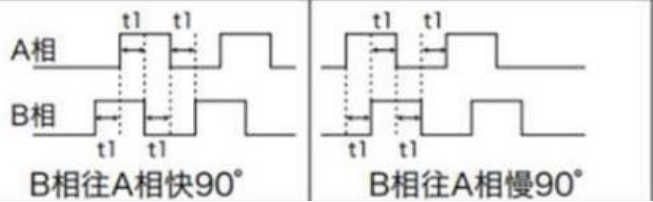
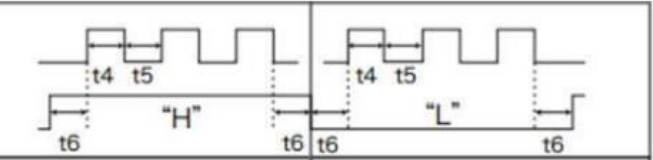
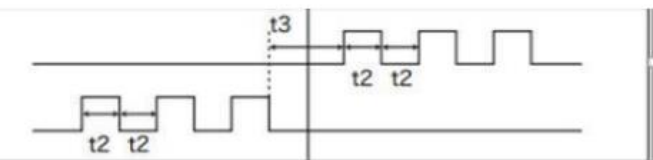


Figure 31-1. Quad Timer Block Diagram

PLC硬件组成-CPU模块-PTO

类型	实现方式			
正交解码	可以使用FlexPWM生成90°相差的时钟，并通过XBAR将其接入TMR时钟输入，TMR使用Pulse-Output功能即可	90° 位相差 2相脉冲 (A相+B相)	PULS SIGN	
脉冲串+符号	可以使用GPIO+TMR模块的Pulse-Output功能	脉冲串 + 符号	PULS SIGN	
CW+CCW脉冲	可以使用TMR模块的Pulse-Output功能	正方向脉冲列 + 反方向脉冲列	PULS SIGN	
PTO所需外设资源		个数	说明	
FlexPWM submodule		2	用于生成90°相差的QTMR时钟，其配置决定PTO的频率	
QTMR channel		2	根据FlexPWM生成时钟计数并发送脉冲，其配置决定PTO的脉冲个数	

PLC硬件组成-CPU模块-PTO

```
typedef struct _PTO_port
{
    uint8_t OutputFlag;           //输出状态
    uint32_t Pwm_SrcClock_Hz;     //PWM时钟输出源
    uint32_t PwmFreq_Hz;         //当前输出频率
    qtmr_channel_selection_t qtmr_chA;
    qtmr_channel_selection_t qtmr_chB;
    pwm_submodule_t subModule0;
    pwm_submodule_t subModule1;
    uint8_t dir_watch;
    TMR_Type *Tmr_Module;
    PWM_Type *Pwm_Module;
    void (*PTO_PWM_Init)(uint8_t port); //通过PWM模块产生两路90度差的时钟源
    void (*PTO_PWM_Update_Frequency)(uint8_t port, uint8_t dir, uint32_t pwmFreq_Hz);
    void (*PTO_QTMR_Init)(uint8_t port);
    void (*PTO_QTMR_Output)(uint8_t port, uint8_t dir, uint32_t count, uint32_t pwmFreq_Hz);
    void (*PTO_PWM_Disable)(uint8_t port);
}
PTO_port;
```

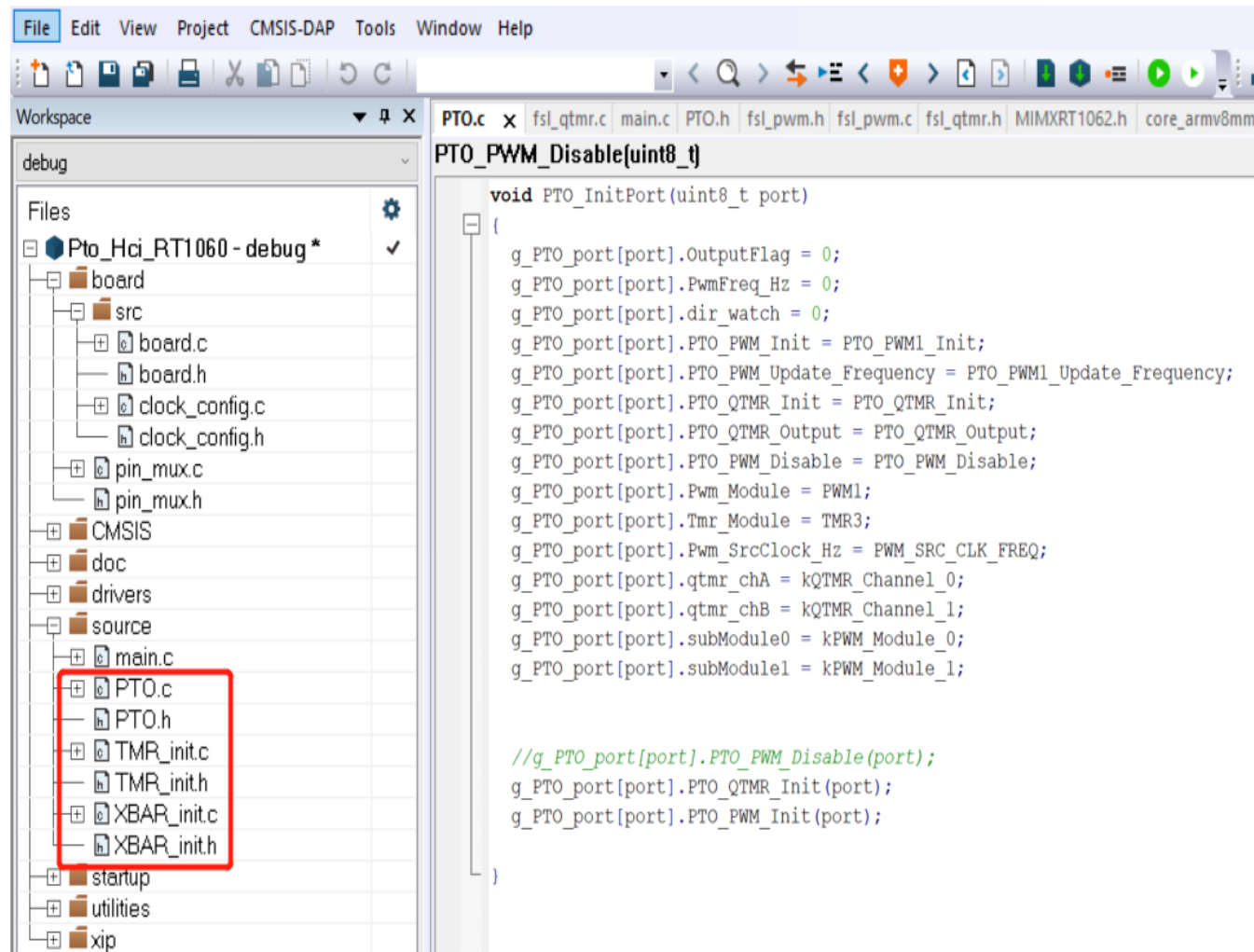
```
int main(void)
{
    uint8_t pulse_number = 10;
    /* Board pin, clock, debug console init */
    BOARD_ConfigMPU();
    BOARD_InitPins();
    BOARD_BootClockRUN();
    BOARD_InitDebugConsole();

    CLOCK_SetDiv(kCLOCK_AhbDiv, 0x2); /* Set AHB PODF to 2, divide by 3 */
    CLOCK_SetDiv(kCLOCK_IpgDiv, 0x3); /* Set IPG PODF to 3, divide by 4 */

    PTO_InitPort(0);

    g_PTO_port[0].PTO_QTMR_Output(0, 1, pulse_number, 1000);
}
```

Pto_Hci_RT1060 - IAR Embedded Workbench IDE - Arm 9.10.2



PLC硬件组成-通信模块

通信模块通常有两种功能：

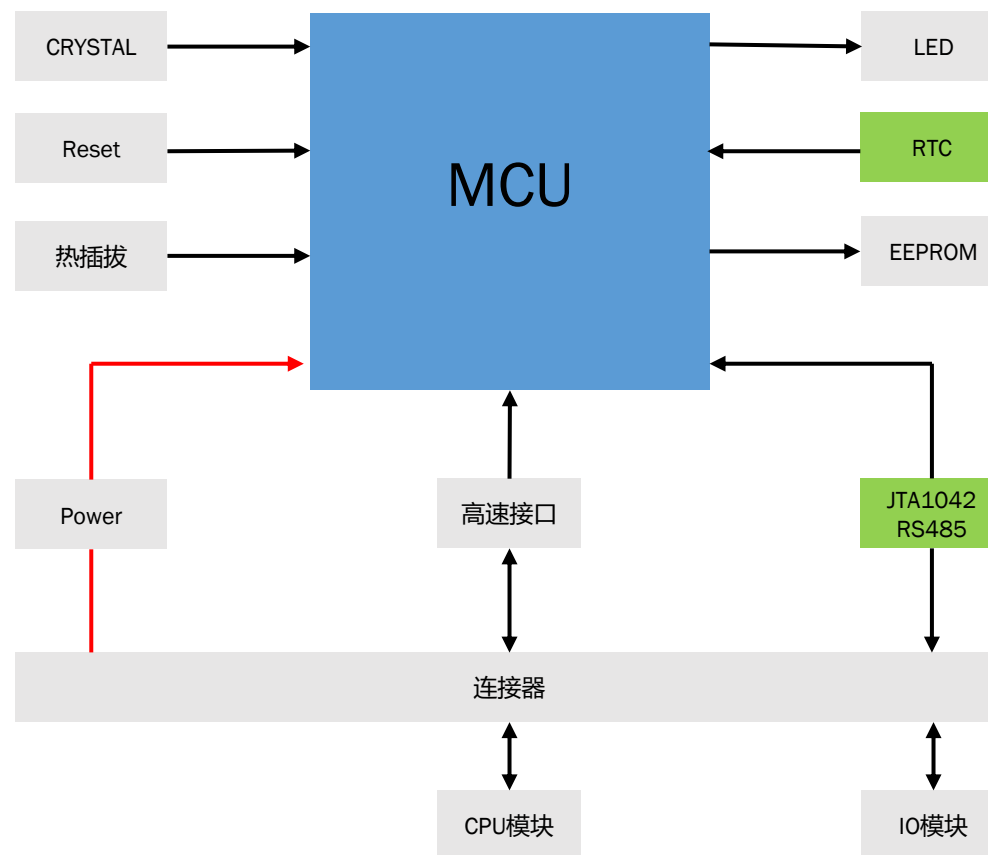
1. 获取本地IO模块的数据
2. 通过网络进行远程数据的交换

本地IO通常使用背板总线，物理层大多是RS485或者CAN，常见的协议有：

- Modbus RTU,
- Profibus-DP,
- CANopen,
- DeviceNet

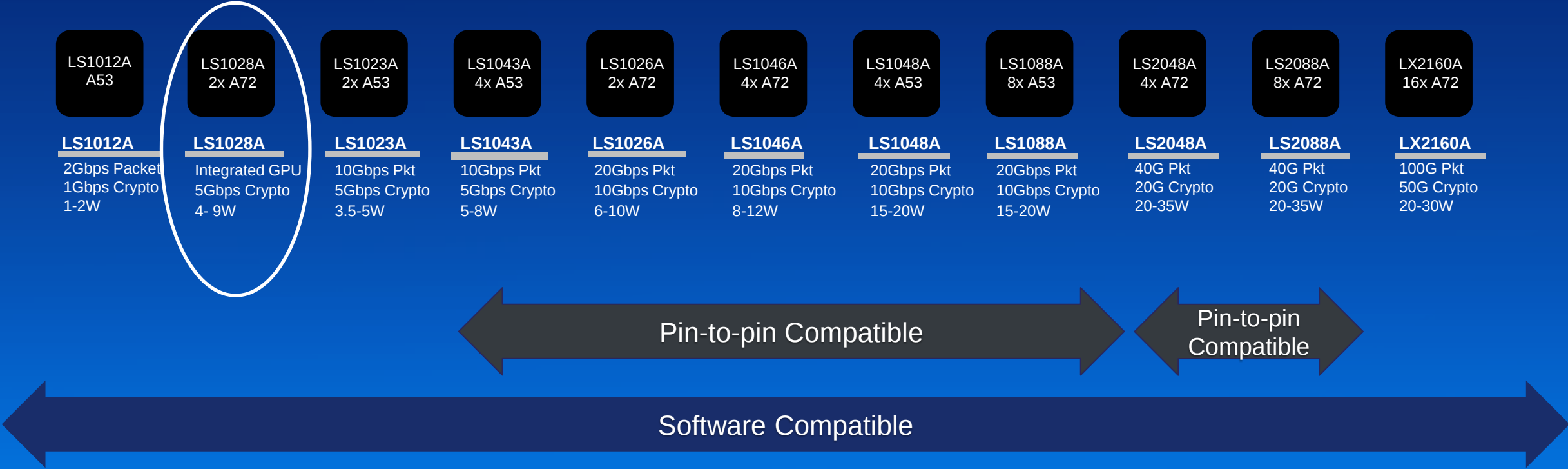
远程IO通常使用工业以太网来进行数据的交换，目前比较常见的有以下几种

- Profinet
- Ethercat
- Powerlink
- EtherNet/IP



中大型PLC的CPU及通信模块 – LAYERSCAPE产品

Scalable series of **64-bit** ARM-based SoC Families



Expanded series for performance, power efficiency and lower BOM



PLC编译器



SECURE CONNECTIONS
FOR A SMARTER WORLD

PUBLIC

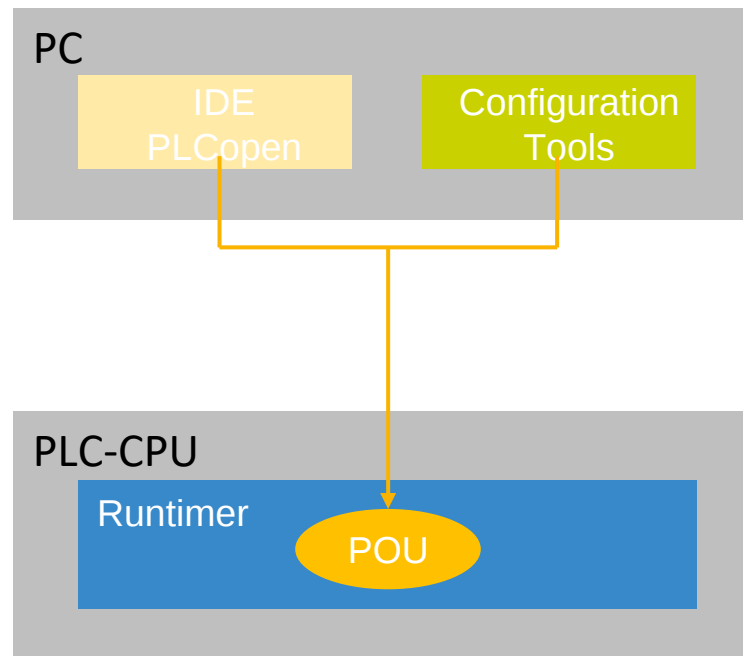
NXP, THE NXP LOGO AND NXP SECURE CONNECTIONS FOR A SMARTER WORLD ARE TRADEMARKS OF NXP B.V.
ALL OTHER PRODUCT OR SERVICE NAMES ARE THE PROPERTY OF THEIR RESPECTIVE OWNERS. © 2022 NXP B.V.



PLC编译器

PLC软件系统通常由以下几部分组成：

- PC端IDE编译器：用于提供用户进行编程，类似于嵌入式工程师常用的MDK/IAR
- CPU模块运行的Runtime system(RTS)：PLC运行的核心代码，用于解析/执行PC端编译出的用户逻辑代码
- 配置工具（可选）：PC端编译器与RTS之间的桥梁



PLC编译器-分类

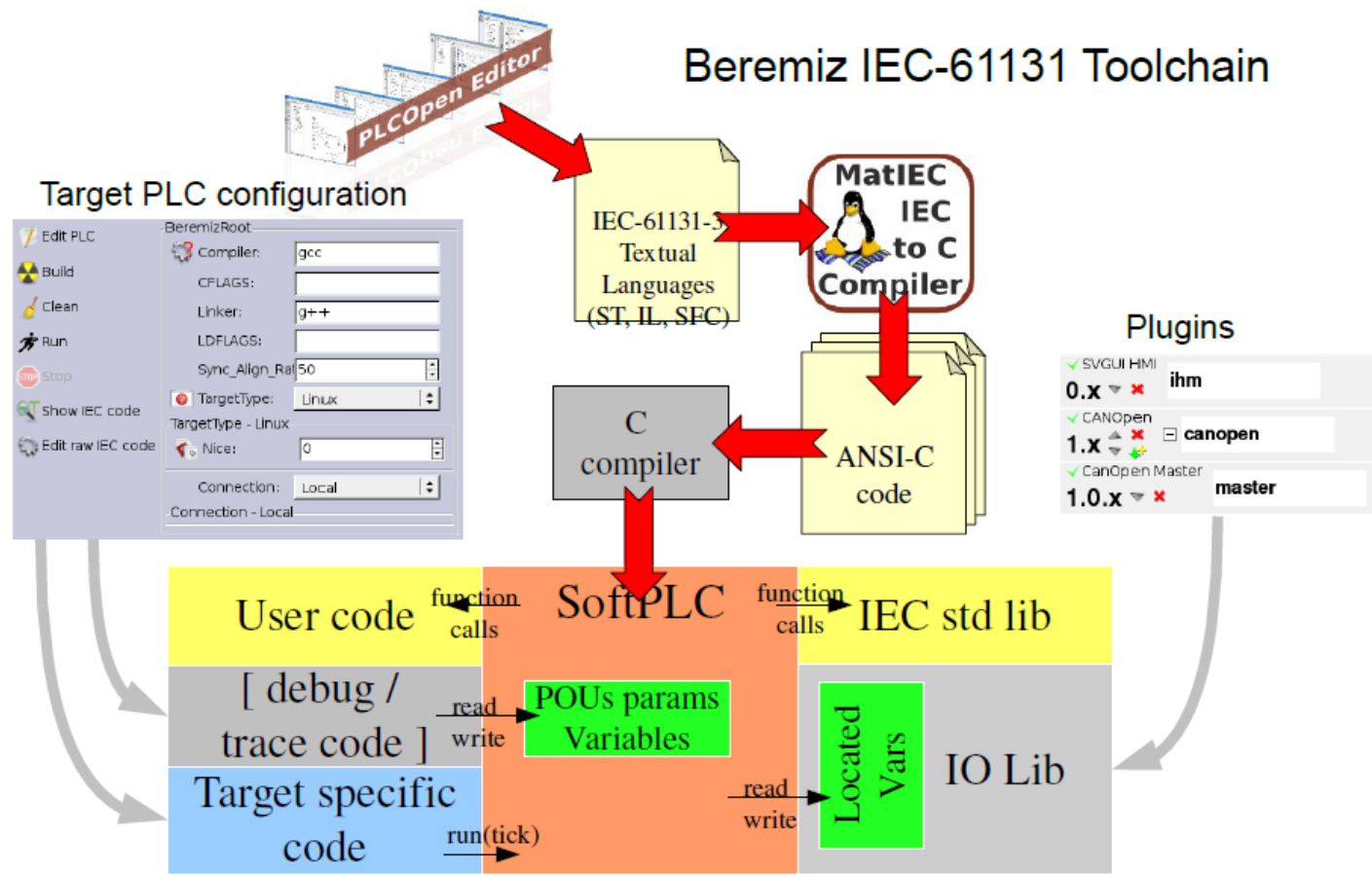
从实现方式上讲，PLC控制器分编译型和解释型两种。这两条技术路线还是有比较大的区别，工程人员在PC端IDE编写梯形图或者ST代码，如果IDE生成的文件是PLC可以执行的二进制代码，这就是编译型的，而如果生成的文件仅仅是中间文件，其需要PLC的程序去解释执行，文件主要包含命令码，操作码和配置信息等，PLC在获取中间文件后会根据预先定义好的命令码来执行相应的操作，这种就是解释型的。

区别	编译性	解释型
执行效率	高	低
开发难度	难	易
跨平台运行	难	易
反编译源文件	难	一般
无扰下装	难	易
防克隆	较好	较差



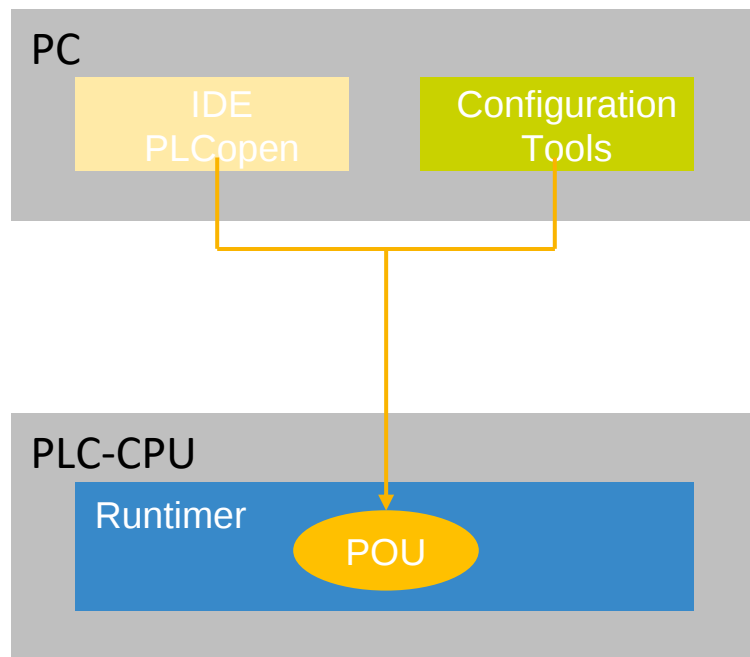
PLC编译器-编译型PLC

编译型PLC本质上就是PC端IDE生成的固件或者二进制文件是可以直接在PLC设备端运行的机器码，这就要求PC端IDE要集成CPU模块的编译器。为了更容易说明这个问题，以开源PLC编译器Beremiz为例：



PLC编译器-解释型PLC

解释型PLC的实现过程相对比较容易，PC端的IDE预先定义好了一套指令集，例如加，减，乘，除对应不同的码值，IDE将用户的逻辑生成相应的POU，当该POU在PLC运行时，Runtime会根据预先定义好的指令集进行解析执行。



程序流程指令

1. CJ 跳转指令
2. CALL SRET FEND 调用子程序
3. IRET DI EI 中断处理
4. WDT 看门狗定时器
5. FOR NEXT 循环指令

运算逻辑指令

1. ADD 加法指令
2. SUB 减法指令
3. MUL 二进制乘法
4. DIV 二进制除法
5. INC 增加
6. DEC 减少
7. WAND 逻辑与
8. WOR 逻辑或
9. WXOR 逻辑异或

数据传输，格式转换指令

1. MOV 字/字节传送
2. XCHP 交换字节
3. SMOV 数据位移指令
4. BMOV 批量传输指令
5. BCD 转码指令
6. BIN 转码指令

旋转，位移操作指令

1. RORP, ROLP 左，右旋转
2. RCRP, RCLP 包含进位标志的左，右旋转
3. SF.TRP, SF.TLP 位数据左，右移
4. WSFRP, WSFLP 字符数据左，右移
5. SFWRP, WFRDP 字符数据位移写入、读出

PLC CPU软件组成



SECURE CONNECTIONS
FOR A SMARTER WORLD

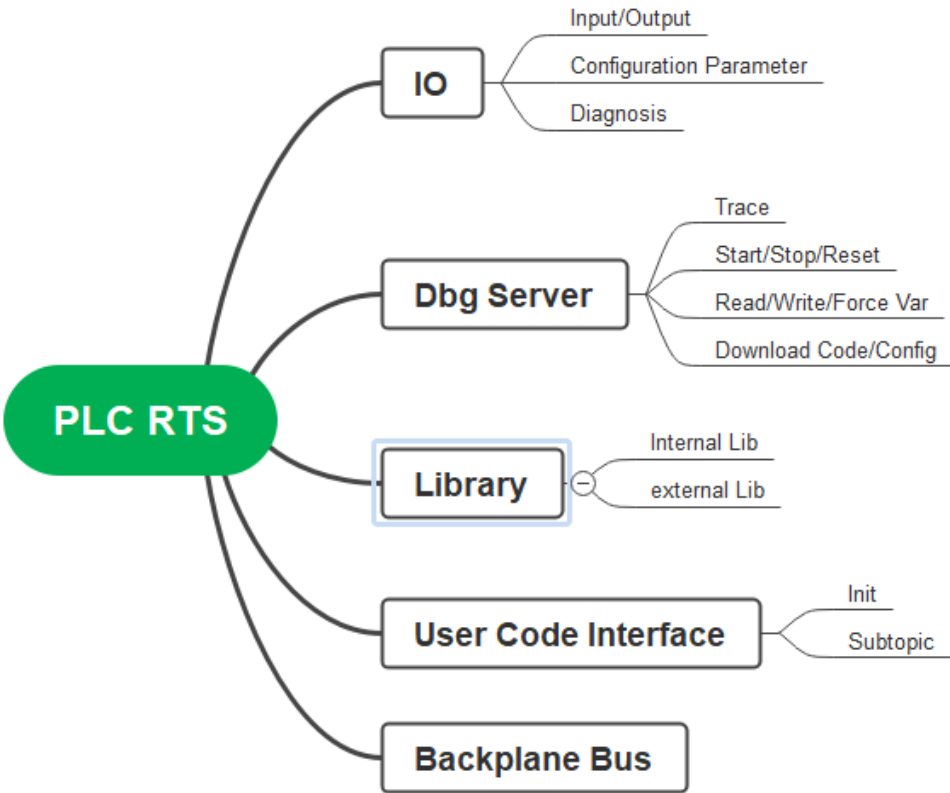
PUBLIC

NXP, THE NXP LOGO AND NXP SECURE CONNECTIONS FOR A SMARTER WORLD ARE TRADEMARKS OF NXP B.V.
ALL OTHER PRODUCT OR SERVICE NAMES ARE THE PROPERTY OF THEIR RESPECTIVE OWNERS. © 2022 NXP B.V.



PLC CPU软件组成

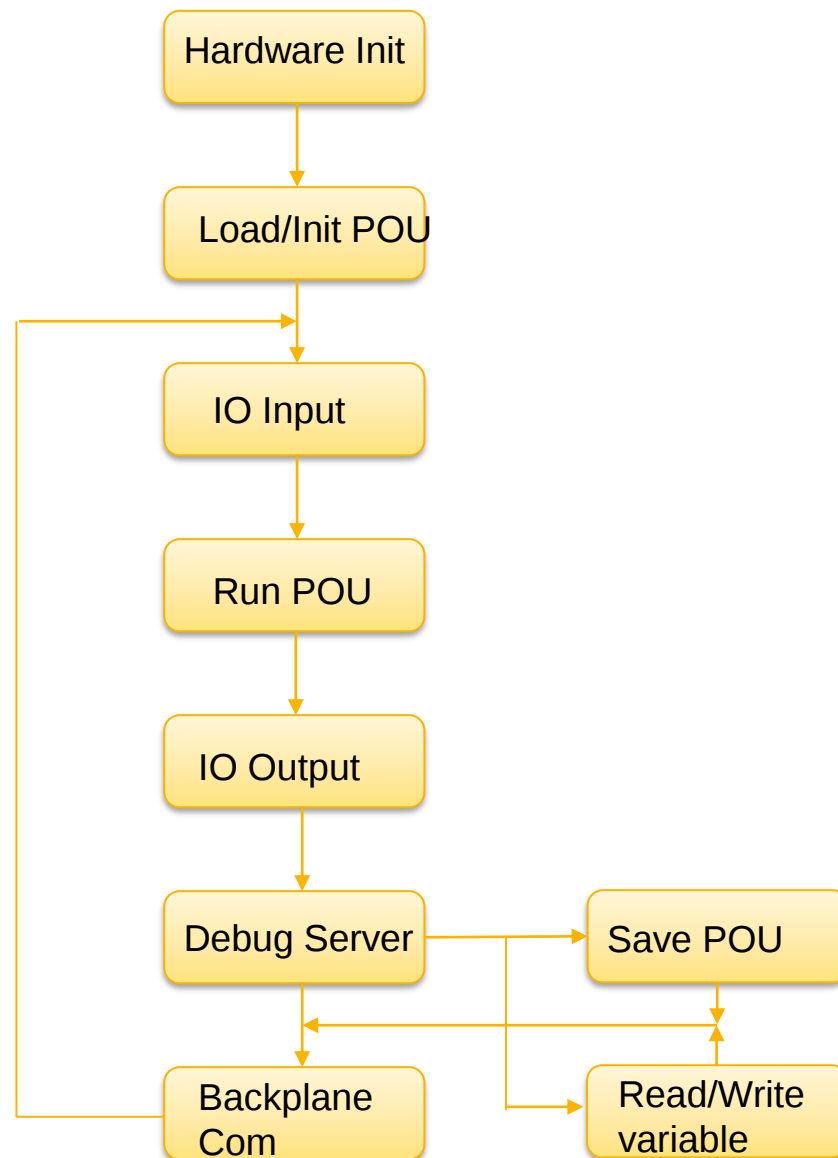
RTS：PLC设备端软件核心，主要有以下五大功能，其中最核心的功能是Dbg和User Code Interface，只要有这两部分功能就可以实现最基本的可编程控制器，I/O组件往往针对本体有I/O功能的设备；Library功能可以不断积累现场应用，简化现场部署难度；对于有扩展功能需求的PLC，背板总线是必不可少的，新一代的PLC在实现背板总线扩展本地I/O的同时，还可以使用支持工业以太网的通讯模块挂接远程I/O



PLC RTS	功能
IO	主要指CPU本体所带的IO通道，常见的有DI，DO，AI，PTO，HCI等
Debug Server	主要用于和IDE进行通信，获取下载用户程序，登录/注销调试模式，读写变量等服务
Library	库分两种，内部库是用户编写的，外部库是PLC在RTS中编写的
POU Interface	RTS的主要功能，配合PC端来运行用户程序
Backplane Bus	背板总线主要与IO模块进行通信，获取IO通道数据

PLC CPU软件组成

- 右图为CPU模块RTS软件的主要流程，对于PLC而言，循环周期是一个很重要的时间参数（即从IO input到下次IO input所需要的时间）
- 每个受控对象都有控制周期的要求，这就要求PLC的循环周期不能超过控制周期的1/2，因为IO的变化可能在本周期IO Input的末尾，这样本周期是不会得到期望的结果，只有在下周期才会去控制输出
- PLC通常要求循环周期不能超过10ms
- PLC用户有时会用输入到输出时间来测试该参数



PLC CPU软件组成-Memory Maps

```
typedef struct
{
    uint32_t * sstart;
    app_fp_t entry;
    //App startup interface
    uint32_t * data_loadaddr;
    uint32_t * data_start;
    uint32_t * data_end;
    uint32_t * bss_end;
    app_fp_t * pa_start;
    app_fp_t * pa_end;
    app_fp_t * ia_start;
    app_fp_t * ia_end;
    //Hardware ID
    uint32_t hw_id;
    //IO manager data
    plc_loc_tbl_t * l_tab; //Location table
    uint32_t * w_tab; //Weigh table
    uint16_t l_sz; //Location table size
    //RTE
    int (*start)(int ,char **);
    int (*stop)(void);
    void (*run)(void);
}
plc_app_abi_t;
```

PLC内存映射

POU

POU Header

RTS

Bootloader

PLC
编译器生成

嵌入式
编译器生成

PLC CPU软件组成-POU加载/运行

```
void plc_app_cstratup(void)
{
    volatile uint32_t *src, *dst, *end;
    app_fp_t *func, *func_end;
    //Init .data
    dst = plc_curr_app->data_start;
    end = plc_curr_app->data_end;
    src = plc_curr_app->data_loadaddr;
    while (dst < end)
    {
        *dst++ = *src++;
    }
    //Init .bss
    end = plc_curr_app->bss_end;
    while (dst < end)
    {
        *dst++ = 0;
    }
    // Constructors
    // .preinit_array
    func = plc_curr_app->pa_start;
    func_end = plc_curr_app->pa_end;
    while (func < func_end)
    {
        (*func)();
        func++;
    }
    // .init_array
    func = plc_curr_app->ia_start;
    func_end = plc_curr_app->ia_end;
    while (func < func_end)
    {
        (*func)();
        func++;
    }
}
```

```
typedef struct
{
    uint32_t * sstart;
    app_fp_t entry;
    //App startup interface
    uint32_t * data_loadaddr;
    uint32_t * data_start;
    uint32_t * data_end;
    uint32_t * bss_end;
    app_fp_t * pa_start;
    app_fp_t * pa_end;
    app_fp_t * ia_start;
    app_fp_t * ia_end;
    //Hardware ID
    uint32_t hw_id;
    //IO manager data
    plc_loc_tbl_t * l_tab; //Location table
    uint32_t * w_tab; //Weight table
    uint16_t l_sz; //Location table size
    //RTE
    int (*start)(int ,char **);
    int (*stop)(void);
    void (*run)(void);
}
plc_app_abi_t;
```

```
static inline void plc_app_start(void)
{
    plc_curr_app->start(0, 0);
    plc_iom_start();
    plc_state = PLC_STATE_STARTED;
}

static inline void plc_app_stop(void)
{
    plc_state = PLC_STATE_STOPPED;
    plc_iom_stop();
    plc_curr_app->stop();
}
```

PLC通讯组成



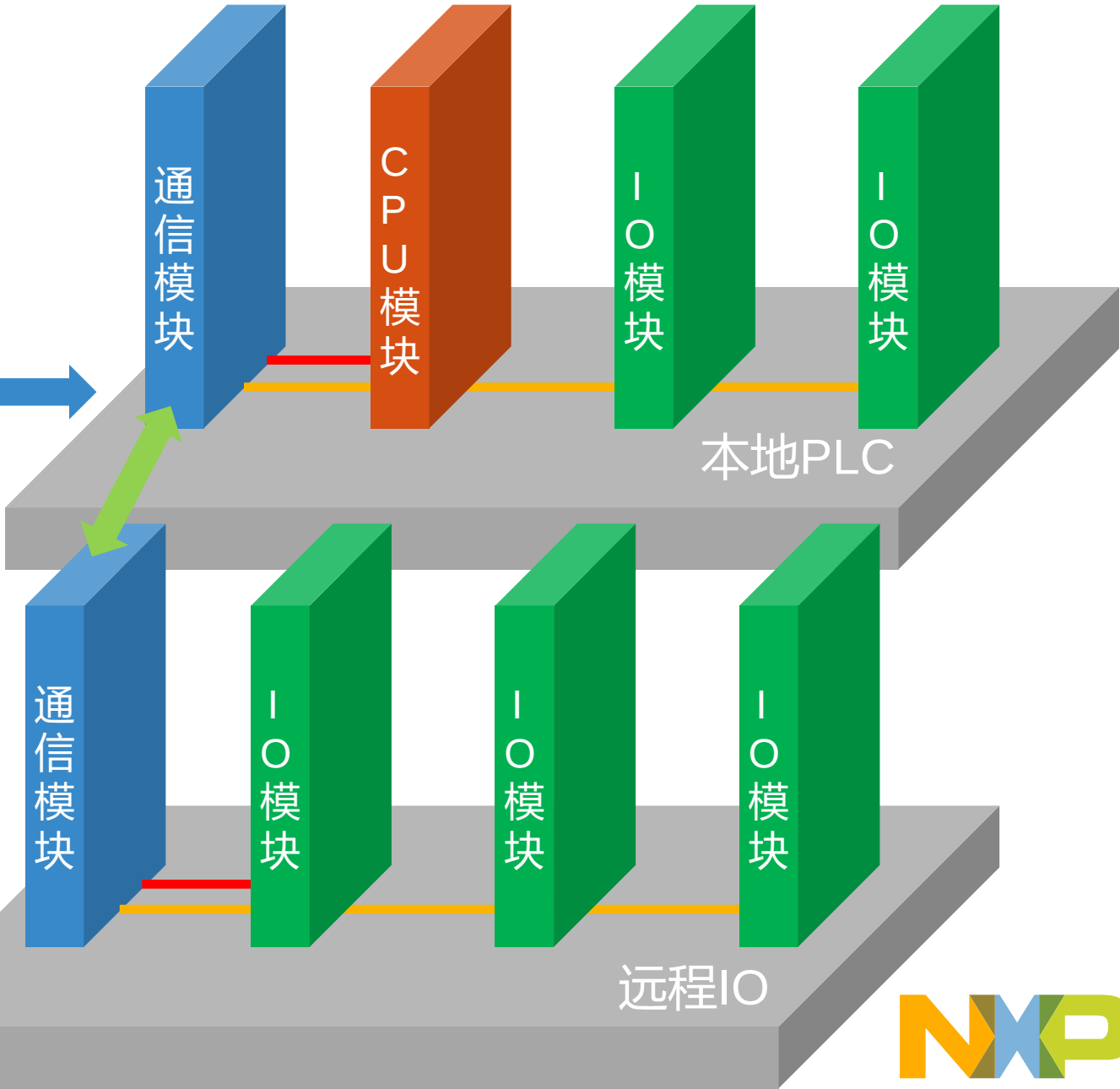
SECURE CONNECTIONS
FOR A SMARTER WORLD

PUBLIC

NXP, THE NXP LOGO AND NXP SECURE CONNECTIONS FOR A SMARTER WORLD ARE TRADEMARKS OF NXP B.V.
ALL OTHER PRODUCT OR SERVICE NAMES ARE THE PROPERTY OF THEIR RESPECTIVE OWNERS. © 2022 NXP B.V.



PLC通信组成

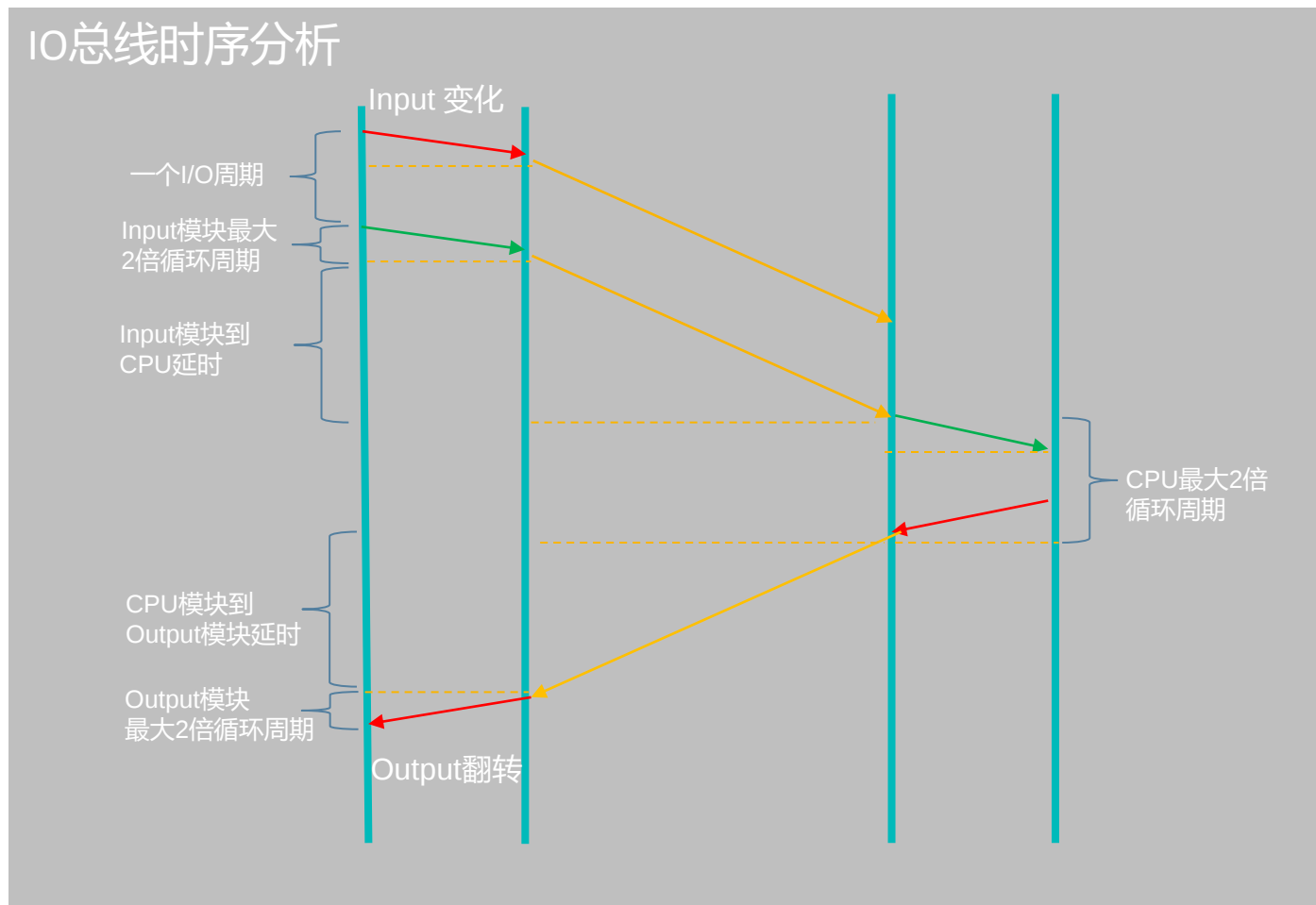


通信组件	
PC与CPU模块	主要用于下载用户代码及在线调试功能，内部协议不开放
CPU，通信模块与IO模块	通常采用RS485/CAN作为物理层，上层运行Profibus-DP，CANopen，Modbus协议
本地与远程	通常采用工业以太网：EtherCAT，Profinet，Powerlink，光纤等
CPU与通信模块	物理层可以采用PCI-E，LVDS，以太网，内部协议不开放



PLC通信组成-IO总线

之前为了引入控制周期的概念，简化了系统结构，以CPU本体I/O作为度量，而实际应用过程中，I/O总线周期对控制周期的影响往往更为关键，因为从输入的跳变到输出的翻转都需要经过I/O总线的延时，所以I/O总线对实时性有着比较高的要求。绝大多数的总线协议都是主/从结构，有多主的设备可以通过令牌的方式切换



PLC通信组成-工业以太网

	Ehternet/IP	POWERLINK	EtherCAT	Sercos III	PROFINET
厂商	Rockwell	B&R Automation	Beckhoff	Sercos	Siemens
网络类型	标准网络+标准MAC	开源协议+标准MAC	开放协议+定制MAC	开放协议+定制MAC	开放协议+定制MAC
应用层	DeviceNet	CANopen	CANopen	Sercos III	PROFIBUS
集中控制	✓	✓	✓	✓	✓
分散控制	✓	✓	-	✓	✓
网络拓扑					
树状网络	✓	✓	-	-	✓
星型网络	✓	✓	-	-	✓
环型网络	✓	✓	✓	✓	✓
菊花链	✓	✓	✓	✓	✓

PLC的趋势和发展



SECURE CONNECTIONS
FOR A SMARTER WORLD

PUBLIC

NXP, THE NXP LOGO AND NXP SECURE CONNECTIONS FOR A SMARTER WORLD ARE TRADEMARKS OF NXP B.V.
ALL OTHER PRODUCT OR SERVICE NAMES ARE THE PROPERTY OF THEIR RESPECTIVE OWNERS. © 2022 NXP B.V.



PLC的趋势和发展-信息安全

- 传统的PLC在控制域内的通信几乎是不加密的，不管是Profibus，Modbus，CANOpen等协议，本身都没有加入安全相关的技术，随着技术的不断发展，控制域的通信安全被更多的厂商所关注。有一些厂商开始试图在原先的通信协议中添加TLS。新型PLC在选型过程中已经开始有硬件加密单元的要求。各个协议组织也在不断的加入安全协议规范，Modbus组织已经发布了基于TLS的MODBUS/TCP
- IEC 62443对工业信息安全提供了结构性方法，国内已经有厂商在其控制器产品中通过了ISASecure(基于IEC 62443) Level 1认证，安全可信PLC即将将成为亮点



PLC的趋势和发展-功能安全

- 传统的PLC从简单的单机运行，到主/从机热备运行，再到三重冗余系统，功能安全的要求不断提高。对于有些应用领域，需要通过相应SIL等级认证才能被行业认可，比如国内的安全仪表系统(SIS)，大都采用了先进的三重化设计（2oo3D） TMR架构
- 各个工业总线组织也积极扩展Safety的特性，EtherCAT有基于功能安全传输的FSoE协议，openSAFETY可以运行在EtherNet/IP， Modbus-TCP， POWERLINK等多个协议之上

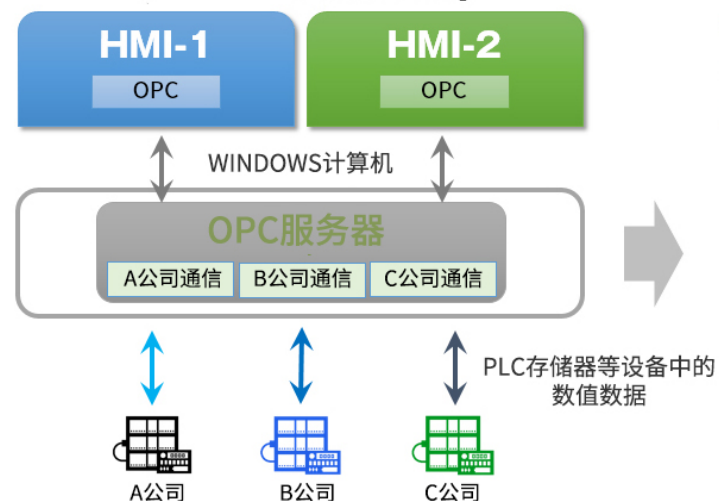


PLC的趋势和发展-OPC UA

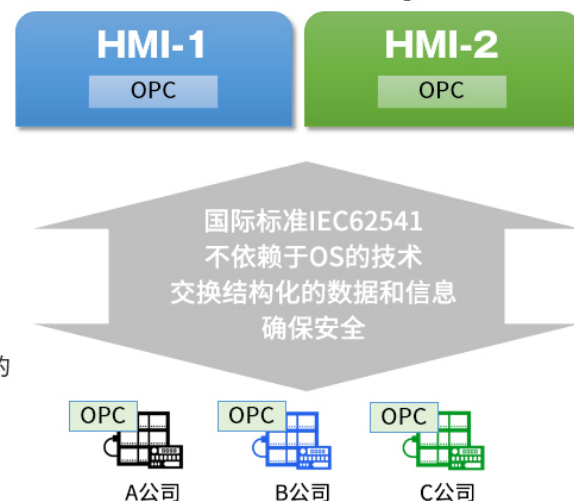
- OPC UA是一种标准的框架，其目的是为了统一不同厂商的网络层数据交互，在先前的工业现场，为了解决数据互通的问题，针对DCS/HMI/SCADA推出了OPC服务，它基于Windows的OLE技术，在IT端将数据整合，而OPC UA从“统一”这个命题出发，提出了更高的要求：独立于平台，可以在任何系统/硬件上运行；支持向下兼容；更合理的配置与维护；更高的可靠性；支持安全协议（例如TLS）等



• OPC 技术 第1代 (OPC Classic)

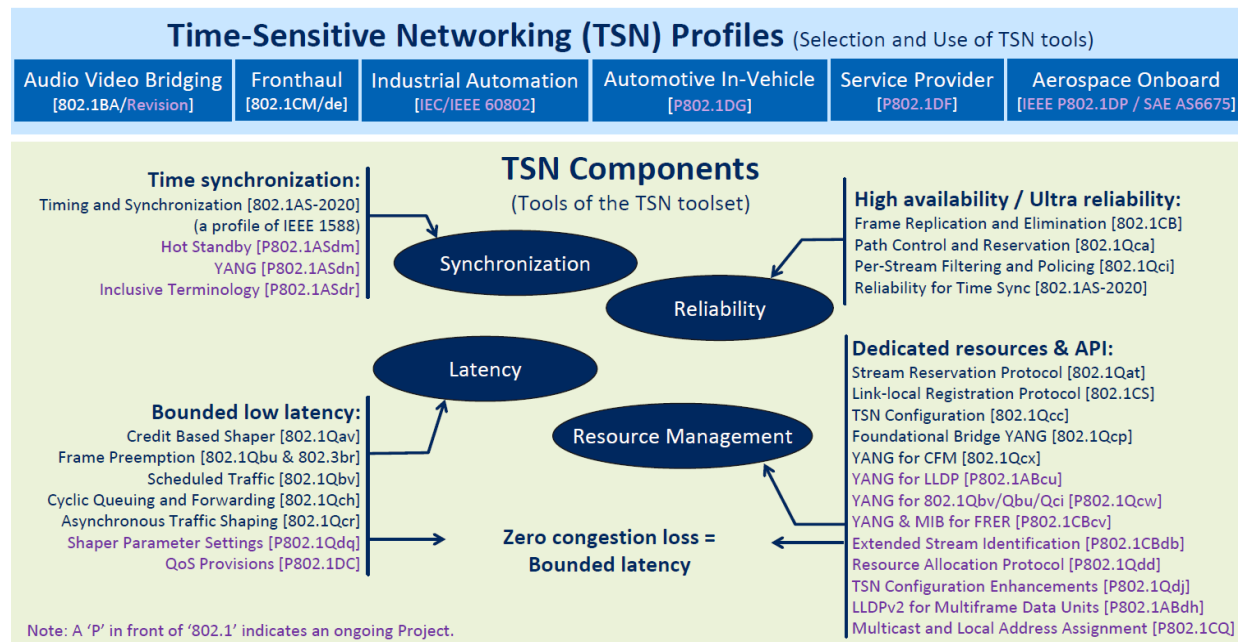


• OPC 技术 第2代 (OPC UA)



PLC的趋势和发展-TSN

- 时间敏感网络(TSN)是一种新兴的工业通信技术，它由一系列的标准所组成，IEEE 802.1于2005年成立了802.1 AVB(Audio Video Bridging)工作组，致力于定制基于以太网的音频/视频传输协议，从而解决在兼容传统以太网的情况下数据的实时性，低延时等问题，在发展过程中，引起了汽车和工业领域更多的关注，于是在2012年更名为TSN以适用更多的应用领域。
- TSN主要包括4部分的内容
 - 1. 时间同步
 - 2. 高可靠性
 - 3. 低延时
 - 4. 资源管理
- IEEE组织正在定义针对工业自动化的TSN Profile IEC60802
- OPC UA+TSN实现IT/OT技术的融合



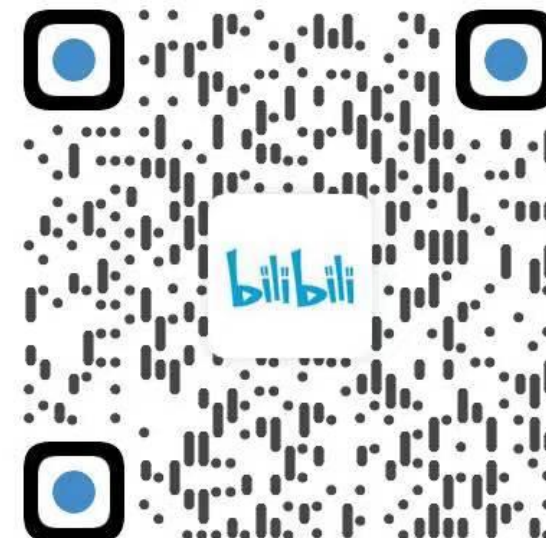
WELCOME TO FOLLOW nxp at social PLATFORMS



欢迎您关注「恩智浦微招聘」公众号
及时获取恩智浦“芯”职位及员工
活动相关资讯



关注NXP客栈公众号，查看恩
智浦最新官方资讯及技术材料



关注恩智浦B站官方账号，观
看恩智浦最新技术视频

Q&A





SECURE CONNECTIONS
FOR A SMARTER WORLD



[SHOWROOM.NXP.COM](https://showroom.nxp.com)