# 9\_AutoSAR基础认知-9-AUTOSAR工具链讲解

面试要点：Autosar内部模块的工作原理，比如存储、通信、看门狗，以及模块的分层。

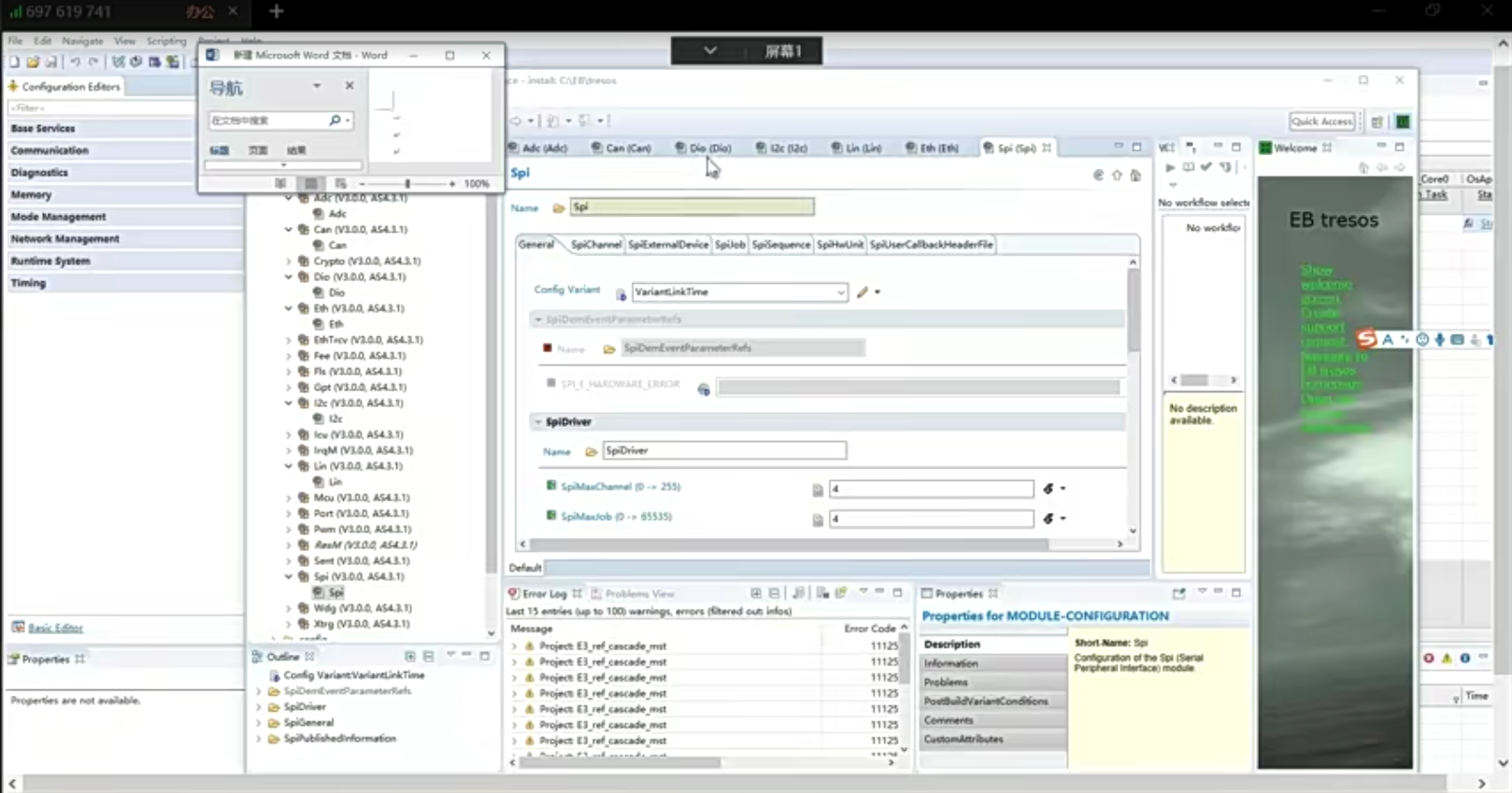
工具链的作用是配置代码，使用工具链的前提也是理解Autosar各个模块的工作原理。

首先Autosar的开发分了应用层软件和底层软件。应用层软件在最上面，各种软件组件swc的开发，可以理解成各种APP的开发。应用层软件开发主要是基于matlab来开发。工业软件Matlab有很强大的数学库。可以使用matlab来建模，比如PID算法都可以直接通过模型拖过来之后直接生成PID的算法，所以应用层软件使用matlab来开发显得非常方便。比如算法中的一些滤波、数学运算等。Matlab建模之后生成应用层软件的代码，然后和底层软件的代码进行集成，合成最终的代码，然后编译生成最终的代码去放到开发板中去运行。

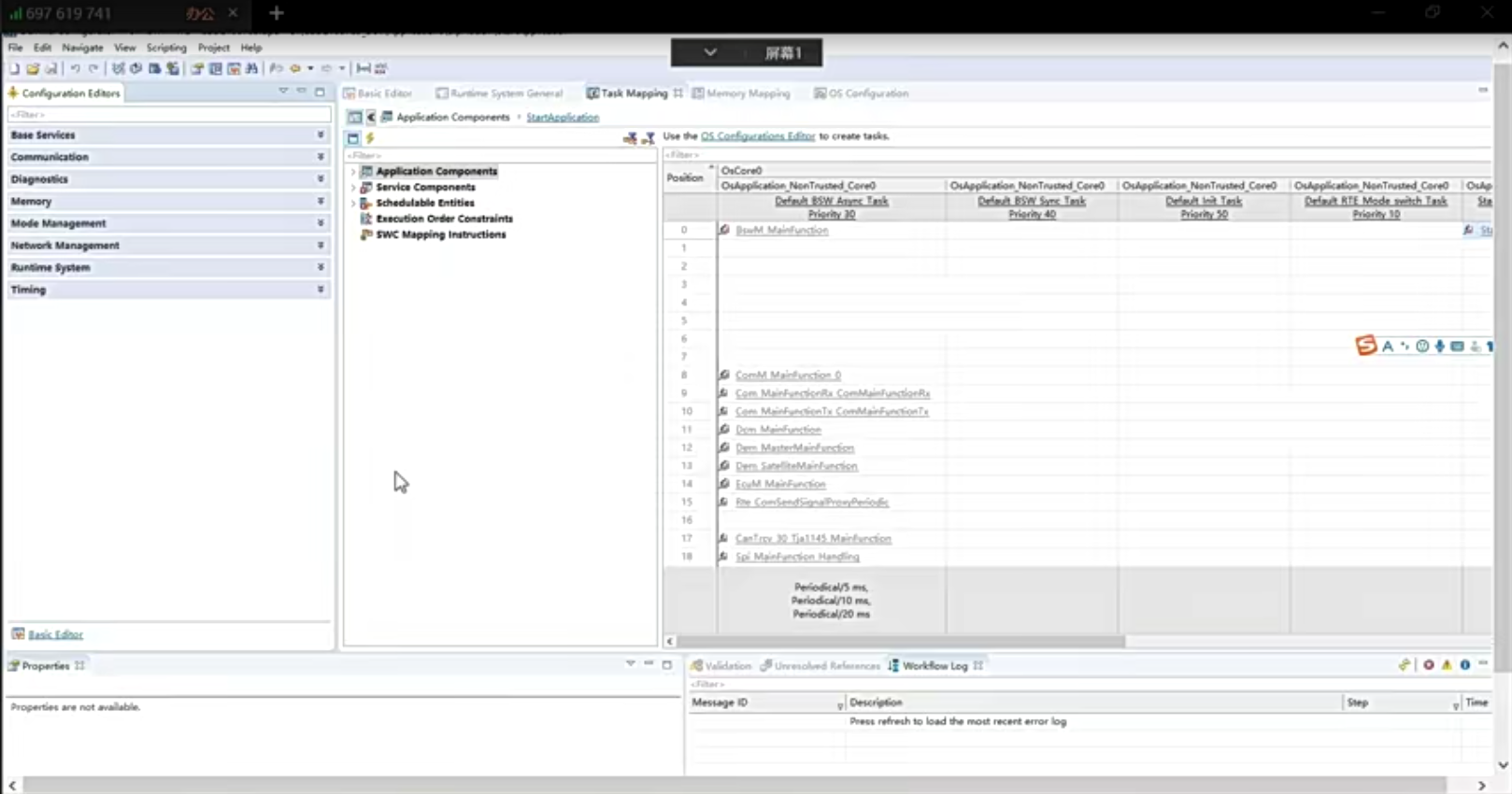


先介绍底层软件mcal驱动的配置，驱动主要使用的是EB工具链，tresos工具链。eb这家公司是大陆汽车电子的子公司，专门做汽车电子的商业软件的，以在autosar领域开发驱动而闻名。比如卖397芯片的英飞凌公司可以找eb工具链公司开发满足autosar规范的驱动，包括恩智浦的s32G。恩智浦公司也会找eb公司开发mcal驱动。芯片公司会提供一些上位机软件去配置一些GPIO、AD、PWM、SPI，配置完之后点生成代码，就可以生成驱动代码了。现在很多芯片公司都支持生成代码了，只不过Autosar不仅仅是生成驱动的配置代码。还得生成满足autosar规范的接口，这些驱动不是简单的配置寄存器就够了，还需要向协议栈层提供标准的函数接口。这些函数接口时不会变的，因为EB只配置了驱动，一定要保证向上层提供的函数接口时唯一的，是不变的，这样工具链才能分离。

然后协议栈层和系统服务层，还有操作系统，用的是vector公司的达芬奇工具。达芬奇工具可以配置除mcal驱动以外的基础软件的这些模块。这些协议栈，系统服务，操作系统都可以通过达芬奇配置出来。达芬奇app有2大功能，衍生出2个小工具，一个是达芬奇developer，这是用来配置RTE的，以及配置一个中间文件给应用层使用，应用层才知道底层会提供哪些接口，哪些CAN信号，应用层就会很方便的拿到文件去使用。达芬奇的另外一个衍生工具Configurator，Configurator是用来配置底层软件的除mcal驱动之外的基础服务的，如操作系统，系统服务，协议栈。

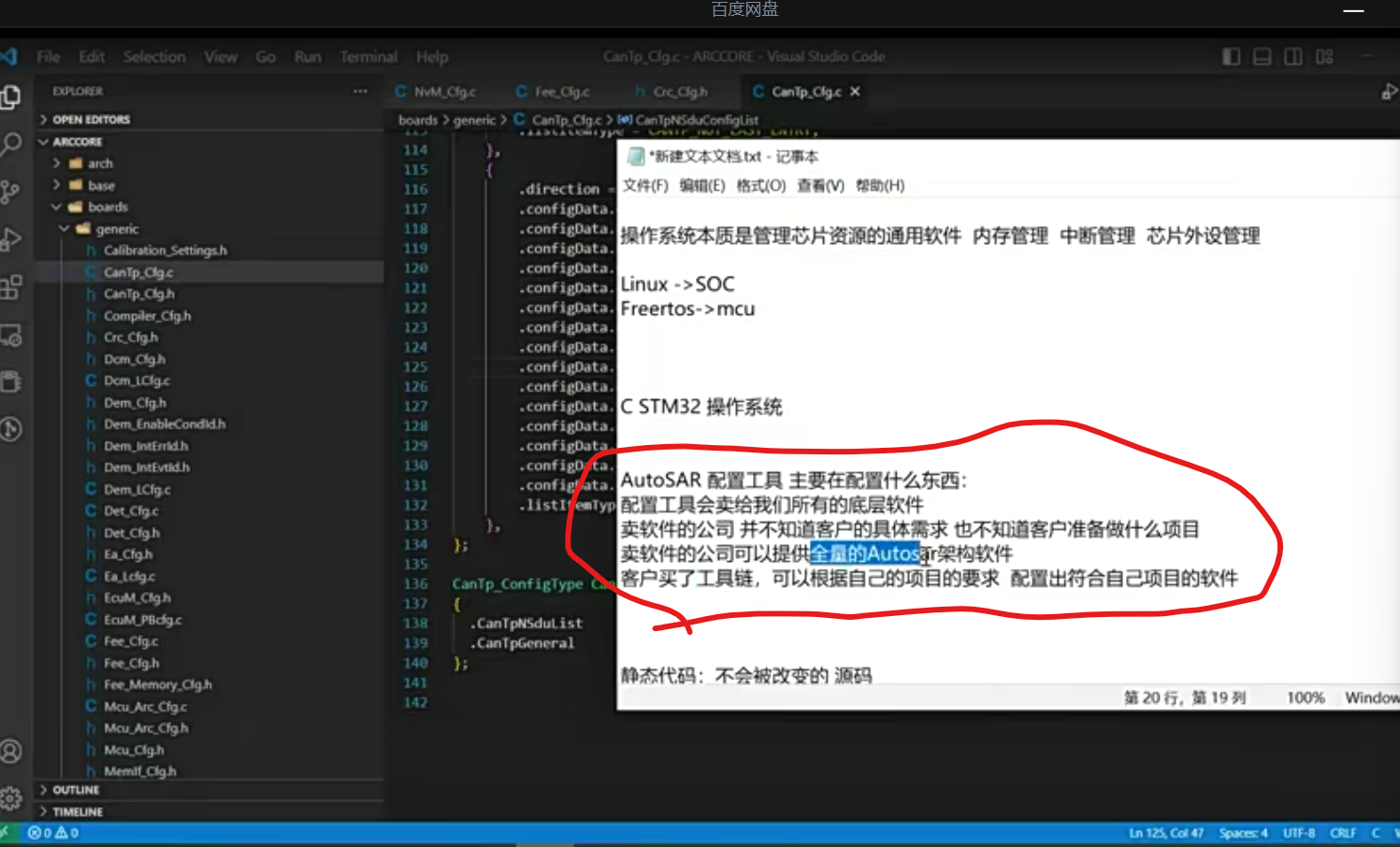


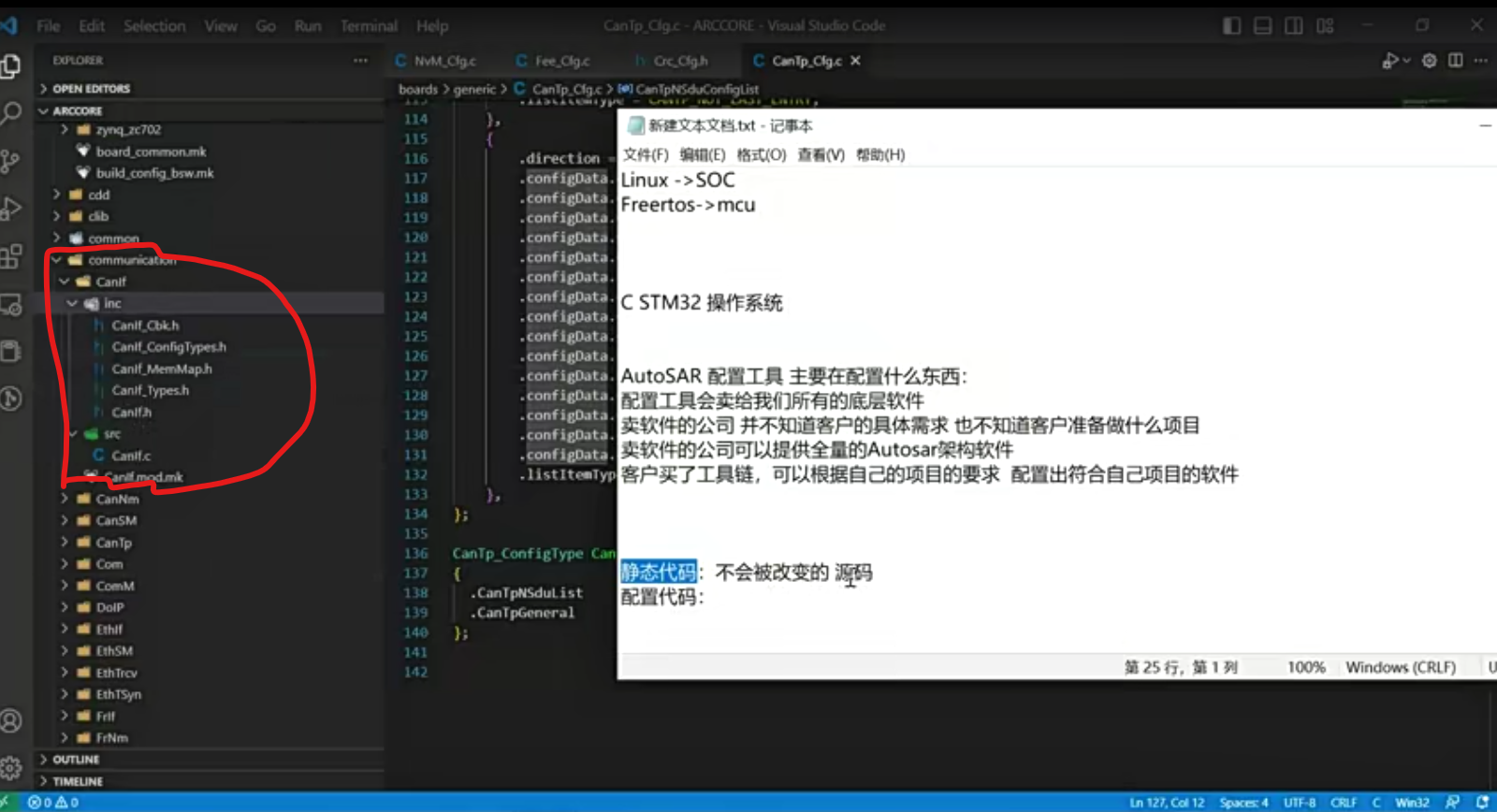
首先是MCAL的EB工具，左边是各种外设，AD采样和CAN总线，DIO、以太网、I2C、SPI、LIN总线等，port口，GPIO口，以及看门狗等外设驱动。在Autosar里面则是配置这些外设驱动相关的参数，精度，IO口等。EB工具主要用来配置寄存器、驱动等。除了配置简单的驱动以外，生成的代码还是符合Autosar规范的。



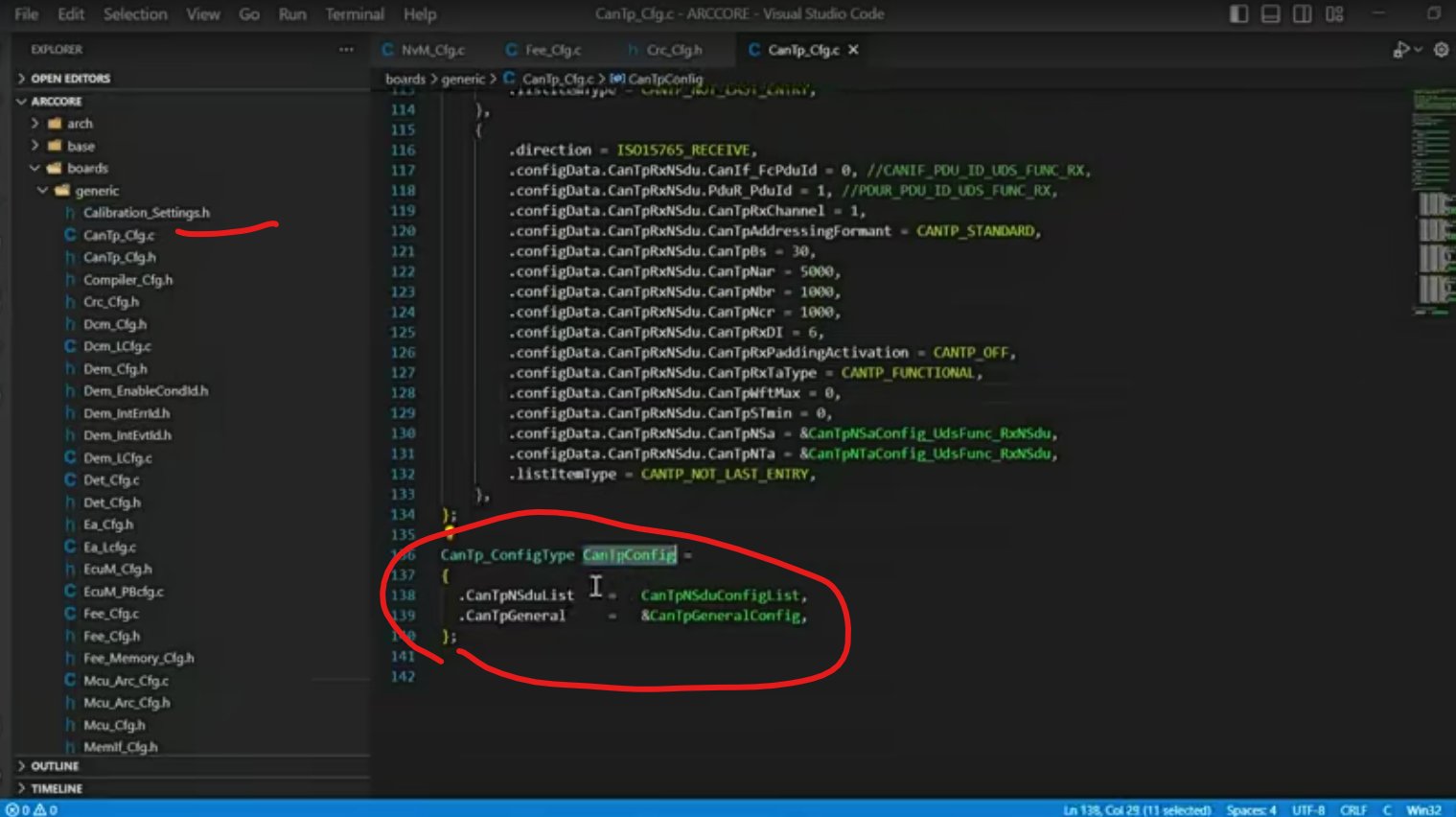
还有一个工具是达芬奇的developer，这个工具是用来配置除了驱动以外所有的底层软件的，需要配置协议栈、操作系统。左边可以看到几大配置项，一个是基础服务，通信(CAN/LIN/ETH)，诊断，存储、模式管理、网络管理、运行时。

比如任务配置，是映射到几毫秒的任务周期里的，以及任务优先级，再比如Com模块，是否使能某模式等，参数设置为多少等。





上图这块通信模块的CAN总线代码，被称为不会改变的静态代码。也就是说在使用配置工具的时候，这部分代码是不会被改变的。配置工具配置出来的代码是针对客户的要求而言的，所以每个产品都不一样，这种代码一般命名为xx\_cfg.c xx\_cfg.h。比如下图就是配置出来的CANtp模块的代码，里面有一个结构体，里面所有的参数根据配置脚本而改变。



配置代码中的结构体中的值会通过配置工具而实时变化。初始化的时候，传递的就是配置的参数，从这里来看是指针。也就是说配置出来的结构体会被静态代码使用，这样就实现了模块的解耦。