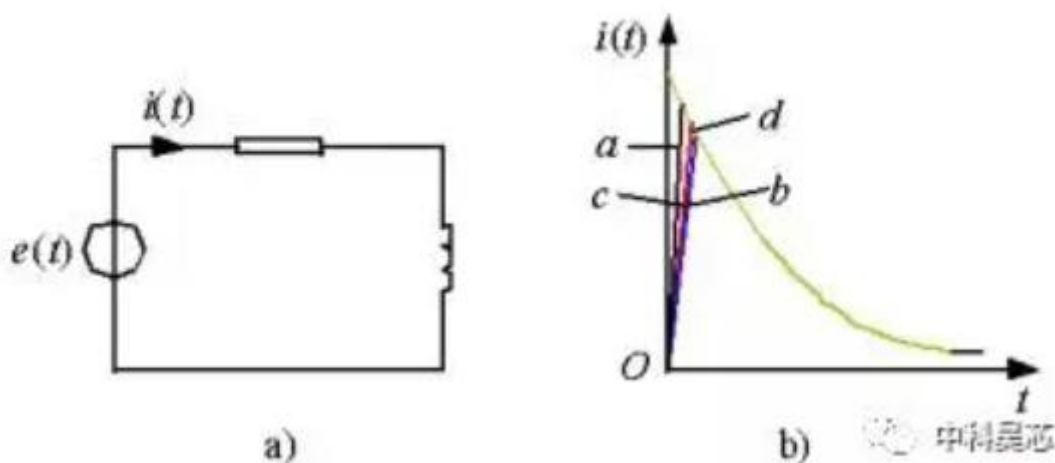
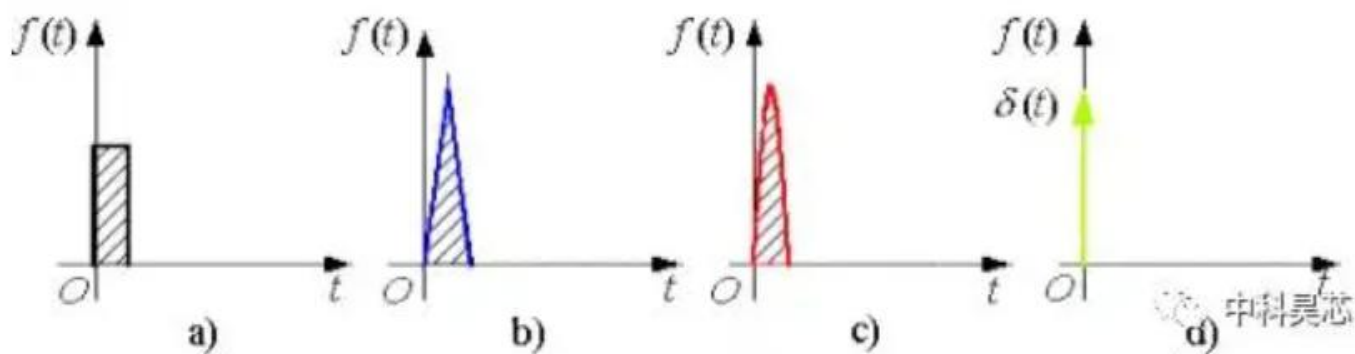


全球智能环保优先，PWM 因其控制逻辑简单，在电动车、电动汽车、轨道交通、新能源发电、工业机器人等电机与电源驱动领域得到重要应用，如下图所示：

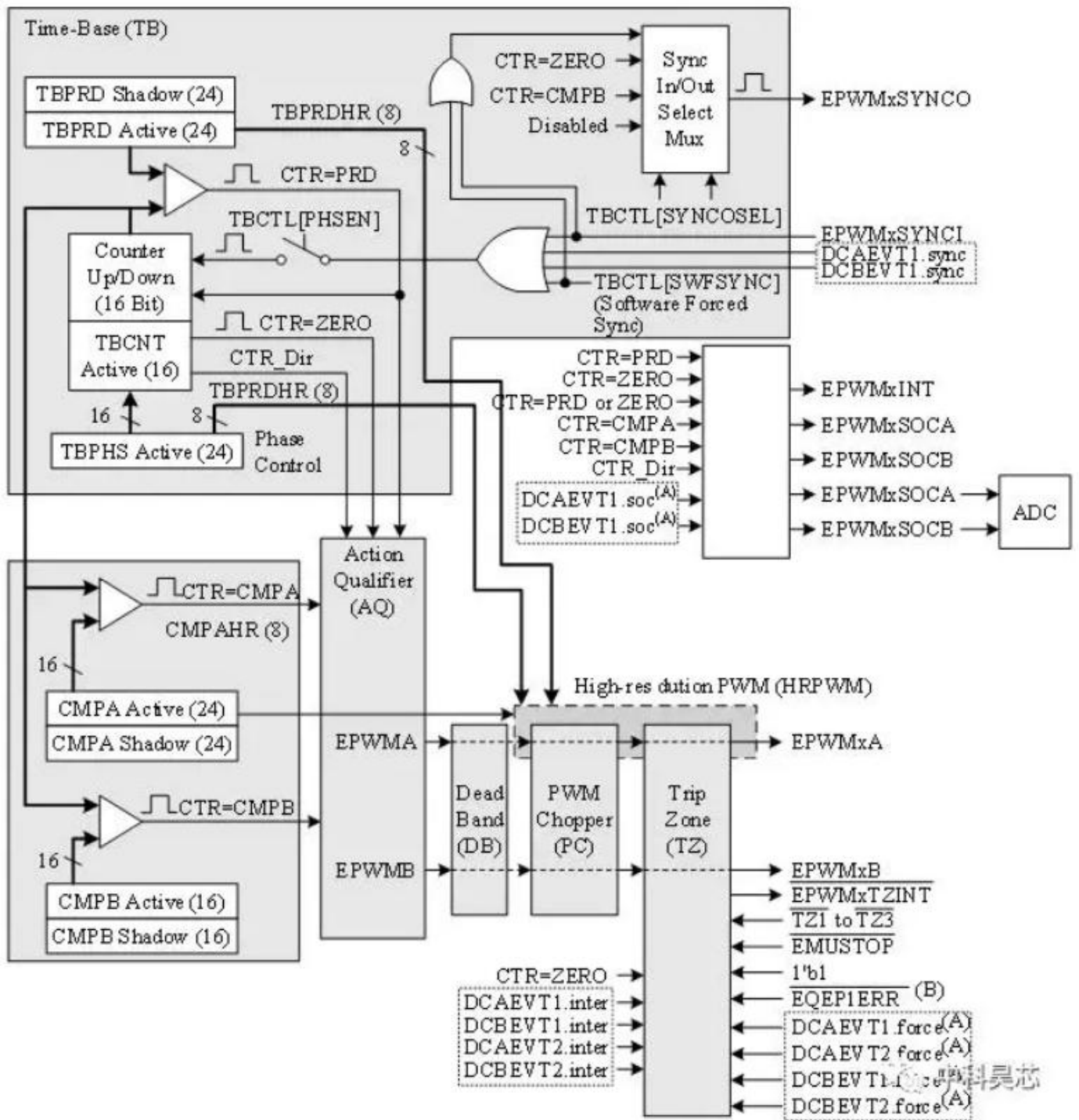


在平头哥半导体有限公司的剑池集成开发环境（简称“CDK”）V2.10.1 版本开始支持中科昊芯 HX2000 系列芯片开发与调试后，本文以 HXS320F28027 的 PWM 输出实例对 PWM 电机调速原理及程序开发展开介绍。

PWM 脉宽调制控制技术，即通过对一系列脉冲的宽度进行调制，来等效地获得所需要的波形，其原理来源于面积等效原理：如下图所示的矩形、三角形与正弦半波、单位窄脉冲波形，其脉冲面积即冲量都等于 1，若将它们分别加在具有惯性的同一环节上时，电路输入为  $e(t)$ ，输出为  $i(t)$ ，其输出响应基本相同，即在低频段特性非常接近，高频段略有差异。



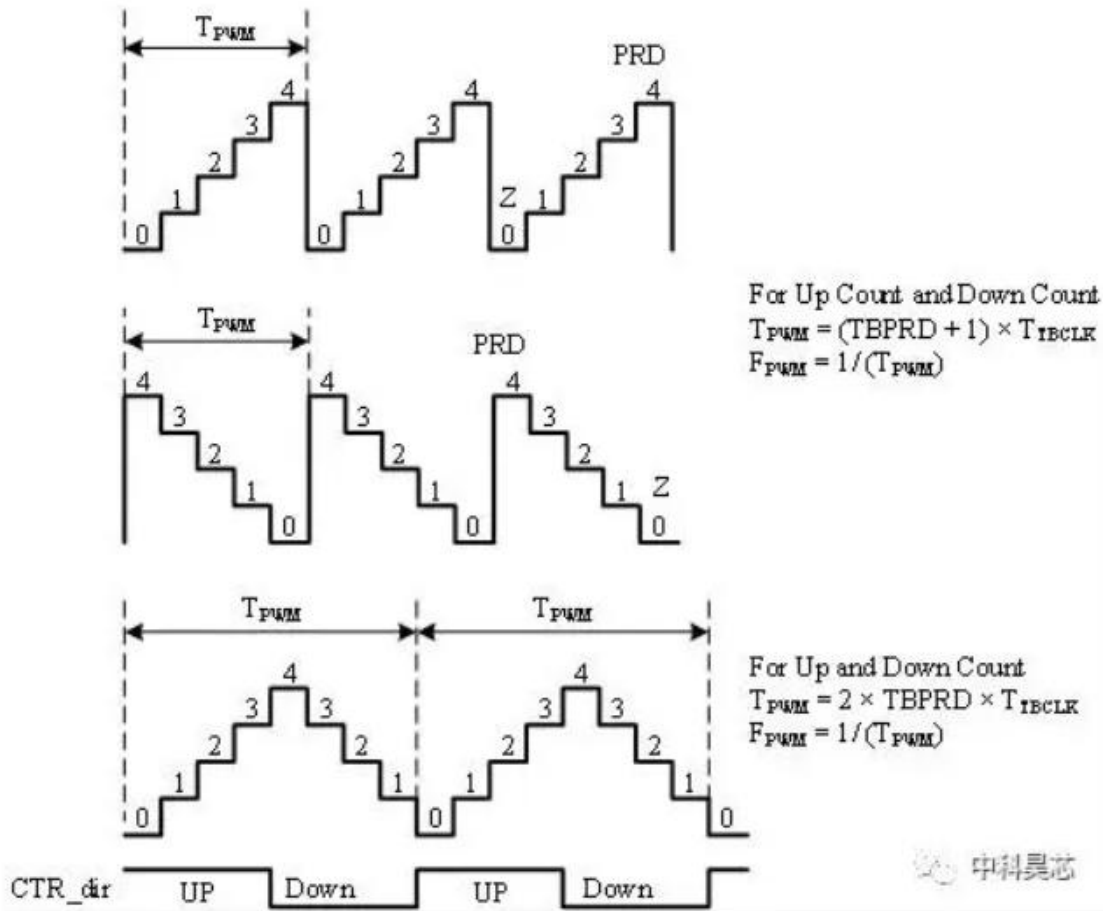
HX2000 系列 PWM 模块如下图：



主要由时基计数器 TB，计数比较 CC、动作 AQ、事件 ET、死区 DB、斩波 PC、错误联防 TZ 与时间比较 DC 模块组成，其中 TB、CC、AQ、ET 是电机调速控制中最重要的四个模块，其使用配置原理为：

### 1) 时基计数器模块：

通过 TBCTL[CTRMODE]配置计数器工作于向上（递增）计数、向下（递减）计数、向上-下（先递增后递减）计数模式；通过 TBCTL[CLKDIV]与 TBCTL[HSPCLKDIV]合理分频系统时钟，使计数器可以工作在任意要求的频率下，计算方法如下图：



分频计算公式为： $TBCLK = SYSCLKOUT / (HSPCLKDIV * CLKDIV)$  配置 EPWMxSYNC 管理 ePWM 模块间的同步；配置 TBPHS 调节两路 PWM 输出的相位差；产生 CTR=PRD 与 CTR=ZERO 事件。

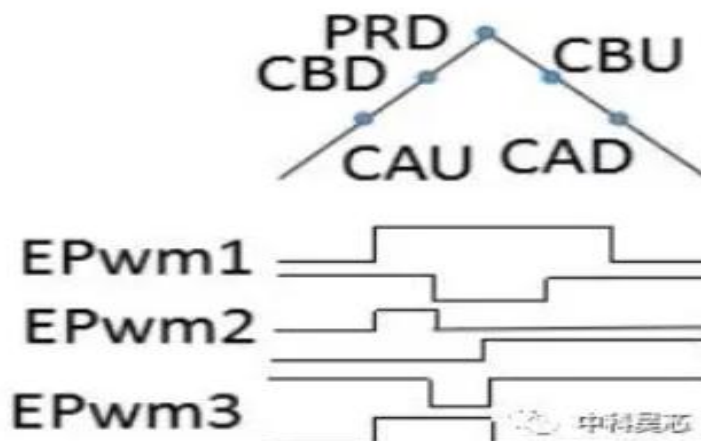
### 2) 计数器比较与动作模块：

以时基计数器的值作为输入，与比较寄存器 CMPA 与 CMPB 不断进行比较，产生比较事件 CTR=CMPA 或 CTR=CMPB，通过动作模块配置在比较事件产生时在不同路 PWM 上产生置高、拉低或翻转动作，从而调节 PWM 控制的脉冲宽度，同时可采用影子寄存器来更新比较值，以有效减少故障与毛刺现象。

### 3) 事件触发模块：

主要接收时基、计数比较与时间比较模块产生相关事件输入，利用时基模块计数器的方向信息、事件模块预定标判断逻辑判断，在相关事件产生时触发，或通过软件配置触发，产生每周一次、两次、三次的相应中断事件或 ADC 启动事件。

本例程主要完成输出如下图周期在 2.2ms，高电平在 55us 到 2145us 变化的 PWM 波，脉冲计数为 0-10 循环。

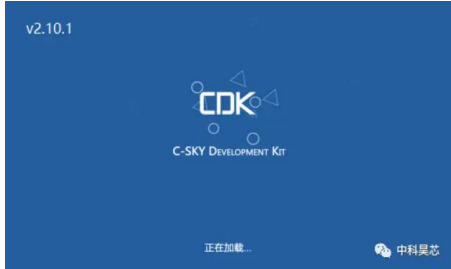
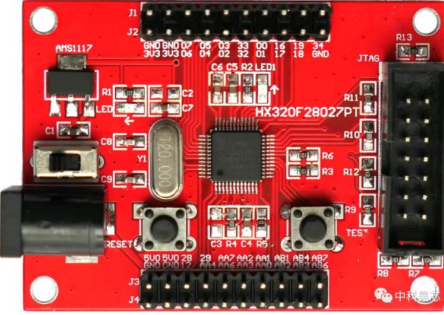



详细介绍参见公众号 B 站视频讲解，二维码如下图：



中科昊芯28027  
 .bilibili.  
 PWM TB时基 AQ动作 CC时间  
 比较

工欲善其事必先利其器，程序开发前准备阶段如下表：

开发环境	开发板	仿真器
剑池集成开发环境 V2.10.1	Core_DSC28027 核心板	HX100V2
		
下载地址： <a href="https://occ.t-head.cn/community/download?id=575997419775328256">https://occ.t-head.cn/community/download?id=575997419775328256</a>	资料地址： <a href="http://haawking.cn/core28027">http://haawking.cn/core28027</a>	申请地址： <a href="http://haawking.cn/DSP-EMULATOR">http://haawking.cn/DSP-EMULATOR</a>

下载界面如下：



准备好开发工具后就可以开始程序开发。详细的“CDK”安装及创建工程方法请看第一篇推送《芯教程 | 平头哥 CDK 助力中科昊芯 HX2000 系列芯片系统开发》。

CDK 上开发 EPWM 波生成，代码包括：

- ①EPWM 的外设引脚、时基 TB、比较 CC、动作 AQ 与事件 ET 模块的初始化配置；
- ②占空比 CMPA 增大与减少的 EPWM 中断服务程序；
- ③主程序调用执行。

```
int main(void)
{
    /*系统初始化控制*/
    InitSysCtrl();
    /*EPWM 外设的 GPIO 配置*/
    InitEPwm1Gpio();
    InitEPwm2Gpio();
    InitEPwm3Gpio();
    /*LED 的 GPIO 配置, 用于判断 PWM 波周期的增减变化区间*/
    InitLED();
    /*关中断, 清中断*/
    InitPieCtrl();
    /*压栈入栈*/
    IER_DISABLE(0xffff);
    IFR_DISABLE(0xffff);
    /*初始化中断向量表*/
    InitPieVectTable();
    /*EPWM_INT 中断向量表地址指向执行相应的 EPWM 中断服务程序*/
    EALLOW;
    PieVectTable.EPWM1_INT=&epmw1_isr;
    PieVectTable.EPWM2_INT=&epmw2_isr;
    PieVectTable.EPWM3_INT=&epmw3_isr;
    EDIS;
    /*禁止 EPWM 的时基使能, 此时允许进行 EPWM 初始化配置*/
    EALLOW;
    SysCtrlRegs.PCLKCR0.bit.TBCLKSYNC=0;
    EDIS;
    /*EPWM 模块初始化配置*/
    InitEPwm1Example();
    InitEPwm2Example();
    InitEPwm3Example();
    /*EPWM 的时基使能, 此时 EPWM 的配置功能将开始起作用*/
```

```

EALLOW;

SysCtrlRegs.PCLKCR0.bit.TBCLKSYNC=1;

EDIS;

/*PIEIER 第 3 组中断向量使能*/

IER_ENABLE(M_INT3);

/*PIEIER 第 3 组第 1 到 3 个中断向量使能*/

PieCtrlRegs.PIEIER3.bit.INTx1=1;

PieCtrlRegs.PIEIER3.bit.INTx2=1;

PieCtrlRegs.PIEIER3.bit.INTx3=1;

/*打开全局中断*/

EINT;

/*关闭全局中断*/

ERTM;

while(1){
}

return 0;
}

```

在 CDK 上开发 PWM 波生成，并对程序进行编译，其结果如下图：

```

int main(void)
{
    InitSysCtrl();

    InitEPwm1Gpio();
    InitEPwm2Gpio();
    InitEPwm3Gpio();

    InitLED();

    InitPieCtrl();

    IER_DISABLE(0xffff);
    IFR_DISABLE(0xffff);

    InitPieVectTable();

    EALLOW;
    PieVectTable.EPwm1_INT=&epwv1_isr;
    PieVectTable.EPwm2_INT=&epwv2_isr;
    PieVectTable.EPwm3_INT=&epwv3_isr;
    EDIS;

    EALLOW;
    SysCtrlRegs.PCLKCR0.bit.TBCLKSYNC=0;
    EDIS;

    InitEPwmExample();
    InitEPwm2Example();
    InitEPwm3Example();

    EALLOW;
}

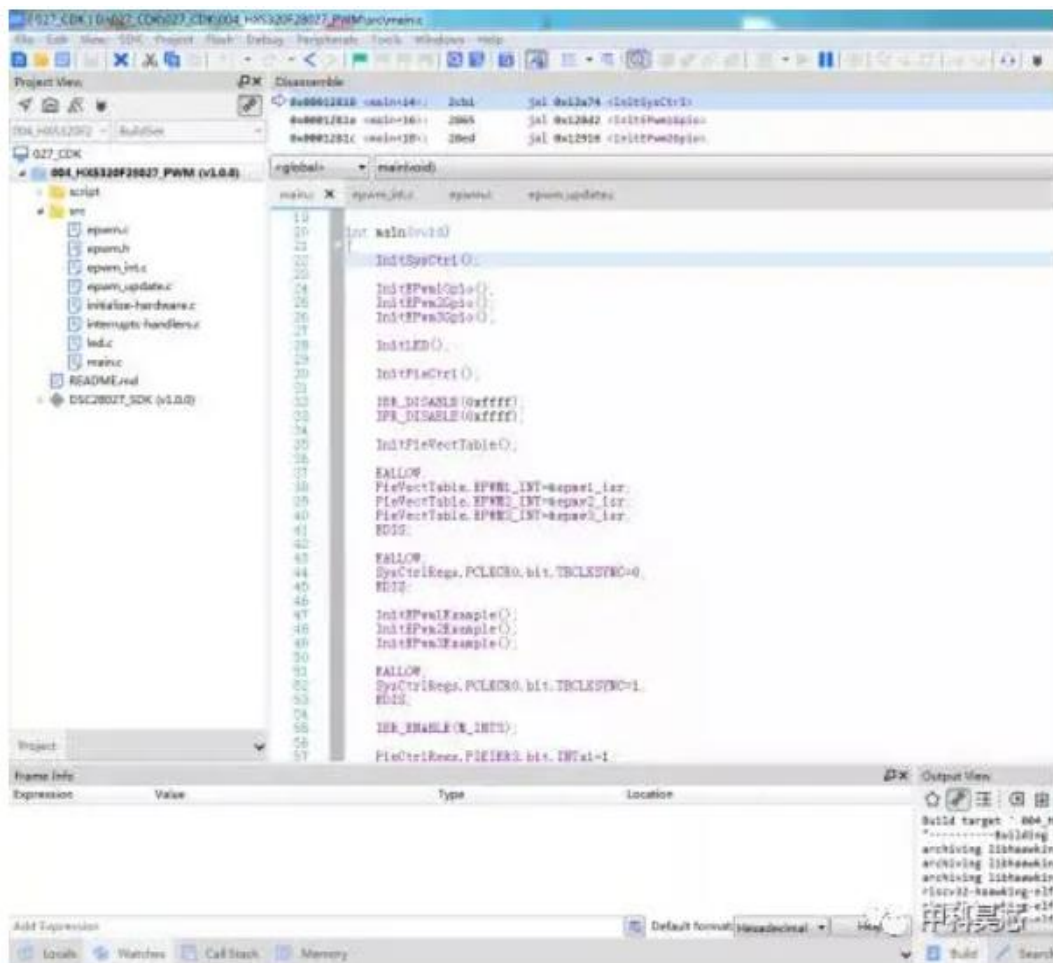
```

```

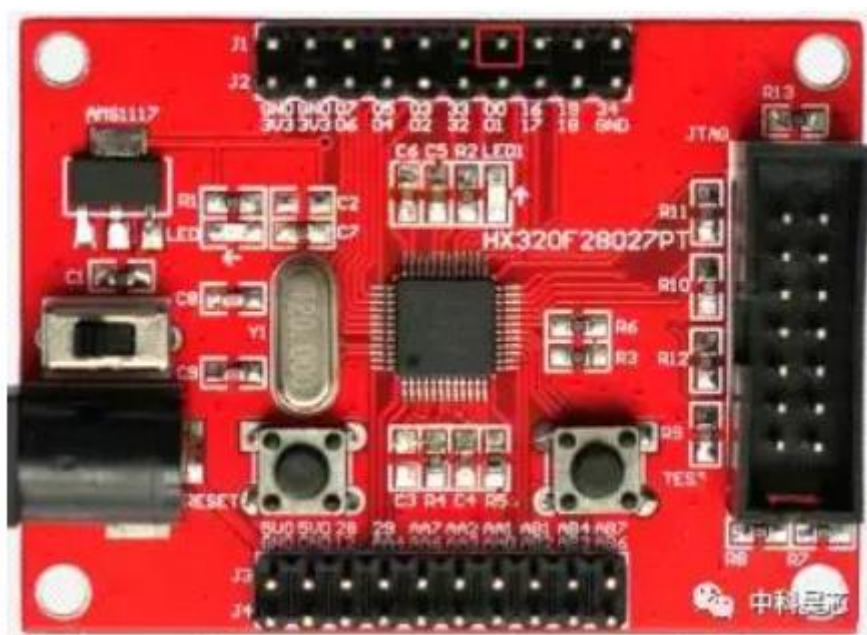
1 warning generated.
archiving libhawkwing_dsc28027_board.a ...
riscv32-haawking-elf-gcc --target=riscv32-unknown-elf --sysroot=../../C-Sky/CDK/CSKY/MinGW/riscv-tc-gcc/riscv64-unknown-elf --gcc-toolchain=...
riscv32-haawking-elf-gcc --target=riscv32-unknown-elf --sysroot=../../C-Sky/CDK/CSKY/MinGW/riscv-tc-gcc/riscv64-unknown-elf --gcc-toolchain=...
riscv32-haawking-elf-gcc --target=riscv32-unknown-elf --sysroot=../../C-Sky/CDK/CSKY/MinGW/riscv-tc-gcc/riscv64-unknown-elf --gcc-toolchain=...
riscv32-haawking-elf-gcc --target=riscv32-unknown-elf --sysroot=../../C-Sky/CDK/CSKY/MinGW/riscv-tc-gcc/riscv64-unknown-elf --gcc-toolchain=...
riscv32-haawking-elf-gcc --target=riscv32-unknown-elf --sysroot=../../C-Sky/CDK/CSKY/MinGW/riscv-tc-gcc/riscv64-unknown-elf --gcc-toolchain=...
riscv32-haawking-elf-objcopy -O ihex ./Obj/004_HXS320F28027_PwM.elf ./Obj/004_HXS320F28027_PwM.ihex
riscv32-haawking-elf-objdump -D ./Obj/004_HXS320F28027_PwM.elf > ./Lst/004_HXS320F28027_PwM.asm
size of target:
text data bss dec hex filename
4620 0 1512 6132 17f4 ./Obj/004_HXS320F28027_PwM.elf
checksum value of target:0xED6EEF75 (182,880)
Obj/004_HXS320F28027_PwM.elf is modified at: 2021/11/17 14:33:03
====0 error(s), 1 warning(s), total time : 2s392ms====

```

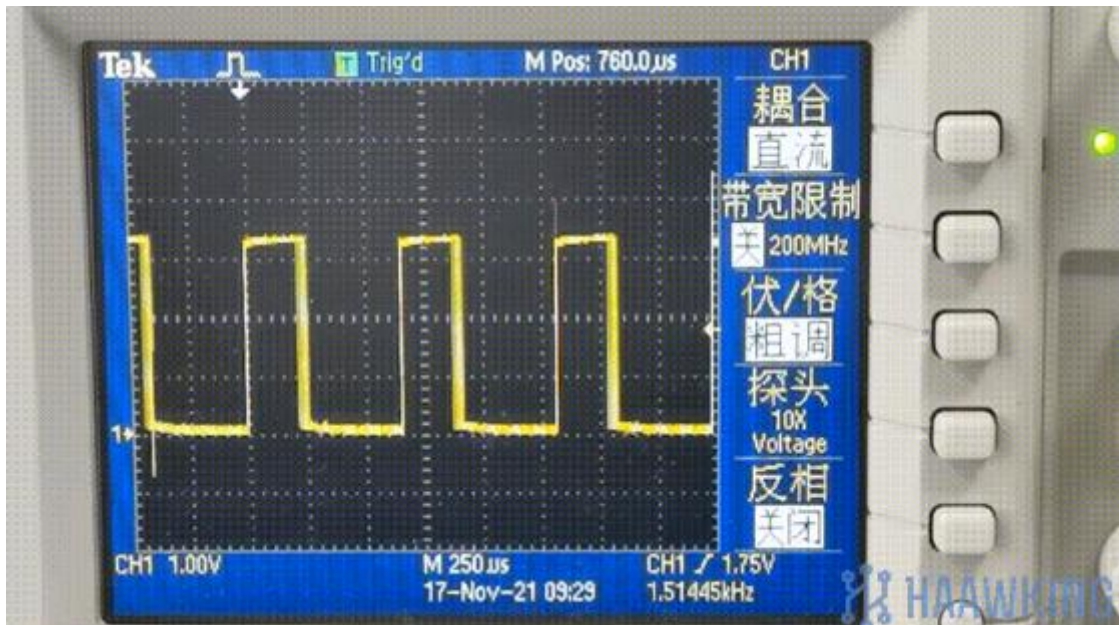
编译成功后就可进入“Debug”模式调试了，其界面如下：



PWM 波通过 GP100 引脚输出，对应的核心板为“00”的插针，具体位置如下图：



PWM 波输出的正确性及其占空比可调性效果如下：



## 关于中科昊芯

“智由芯生 创享未来”，中科昊芯是数字信号处理器专业供应商。作为中国科学院科技成果转化企业，瞄准国际前沿芯片设计技术，依托多年积累的雄厚技术实力及对产业链的理解，以开放积极的心态，基于开源指令集架构 RISC-V，打造多个系列数字信号处理器产品，并构建完善的处理器产品生态系统。产品具有广阔的市场前景，可广泛应用于数字信号处理、工业控制及电机驱动、数字电源、消费电子、白色家电等领域。