# PWM使用sysfs进行控制

和gpio设备一样，pwm也可以通过sysfs进行控制，同样进入/sys/class/pwm目录下。（/sys/class下有着大量的sysfs框架的设备）

使用ls查看当前路径下的文件，发现只有一个pwmchip0的文件夹，进入该文件夹。

里面有这几个文件。

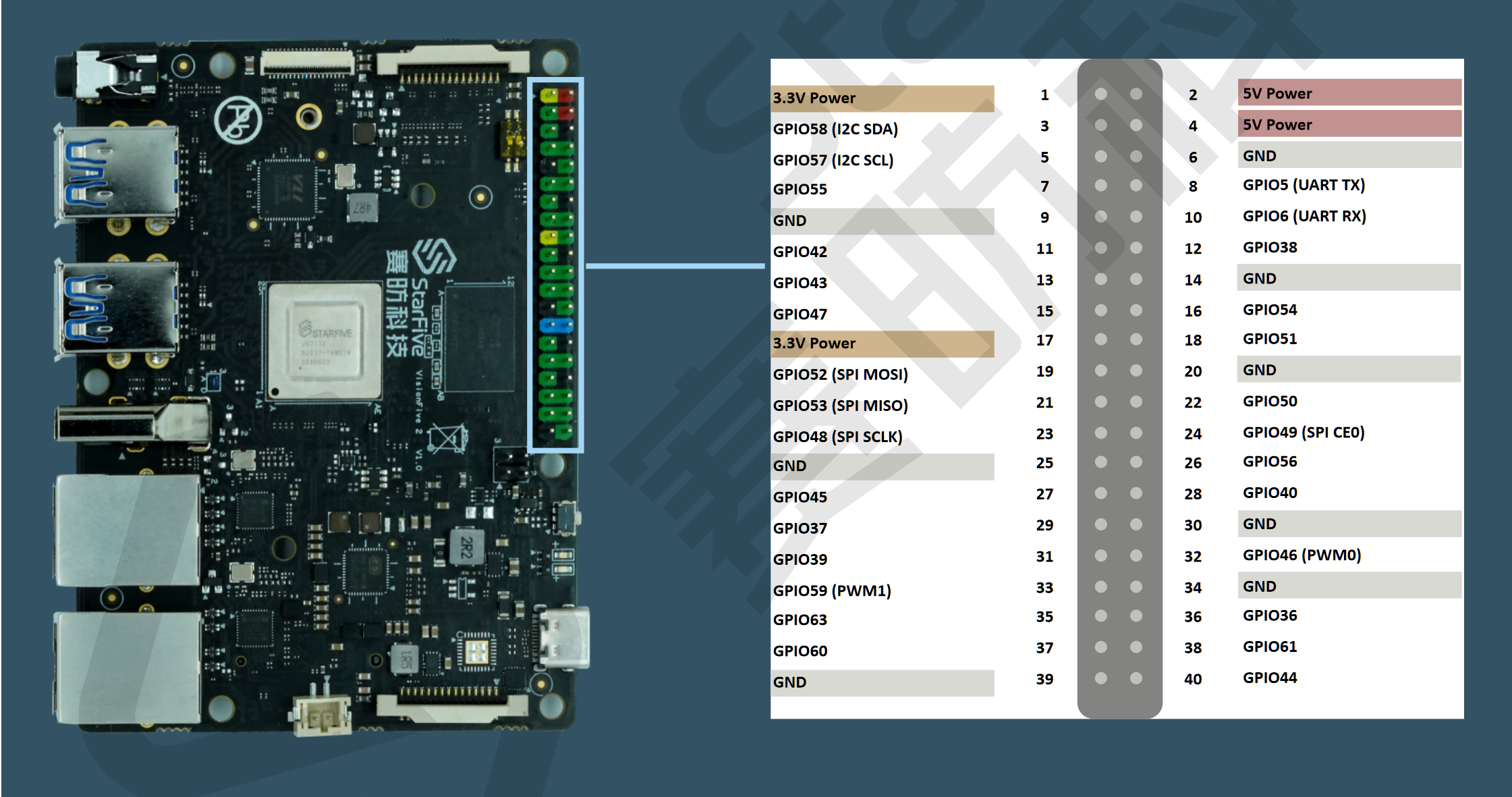
device export npwm power subsystem uevent unexport

我们一般只用关注export npwm unexport三个文件。

npwm文件只读，使用cat npwm，读出数字8，表明，在pwmchi0下有8个pwm。

可以导出控制，export、unexport为将需要使用的pwm导出或取消导出到用户空间，给用户使用（注意在管理员权限下操作）。

现在我们使用pwm0（GPIO46）进行pwm输出使用，这里贴出VF2的引脚定义图方便查看。



我们可以像GPIO一样，将pwm0导出用户空间给用户使用。输出 echo 0 > export,在当前路径下会生成一个pwm0的文件夹，进入pwm0文件夹后也有几个文件：

capture duty\_cycle enable period polarity power uevent

capture：输入捕获，（VF2未实现该功能，应该是硬件不支持）

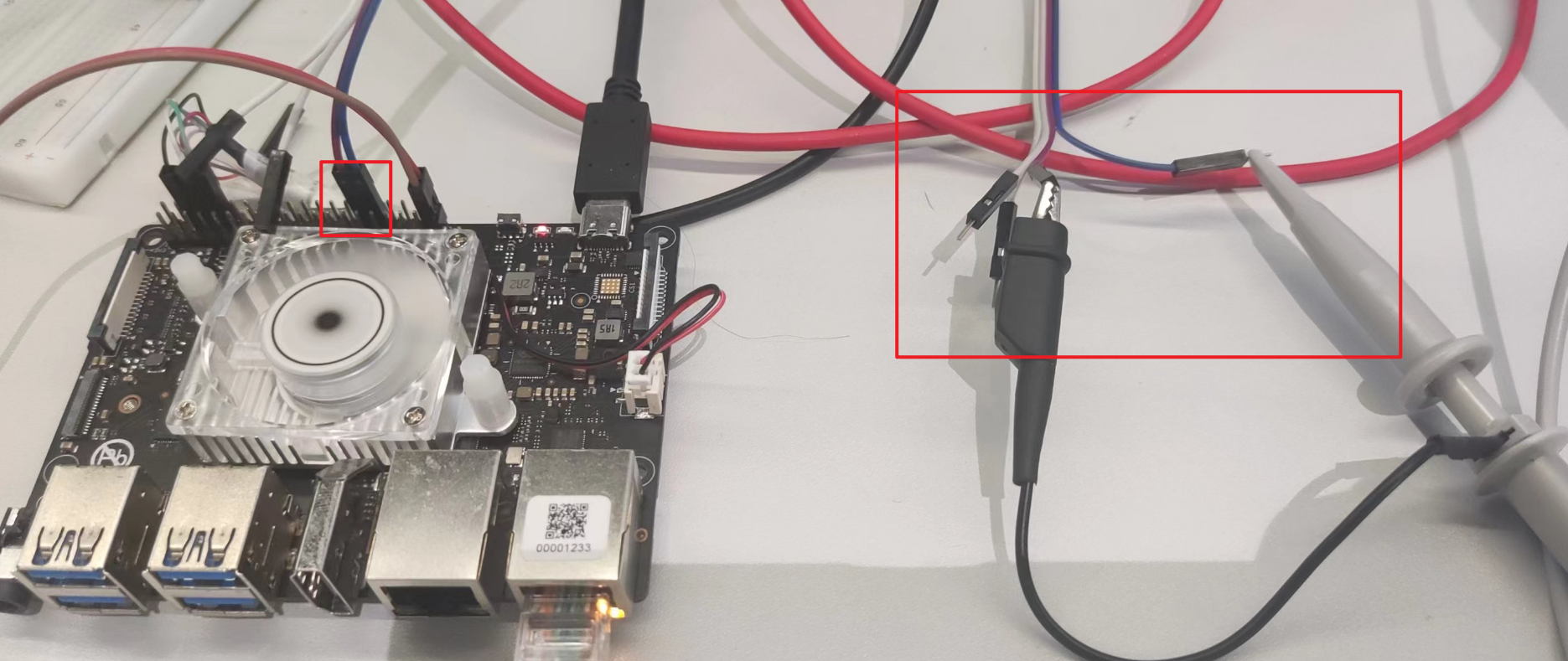
duty\_cycle:配置pwm占空比 n\*10ns

period:配置 pwm周期 n\*10ns

polarity：极性 （normal 正常、inversed 反转）。

enable：使能pwm （0 关闭，1 打开）。

按这样接线：



输入：

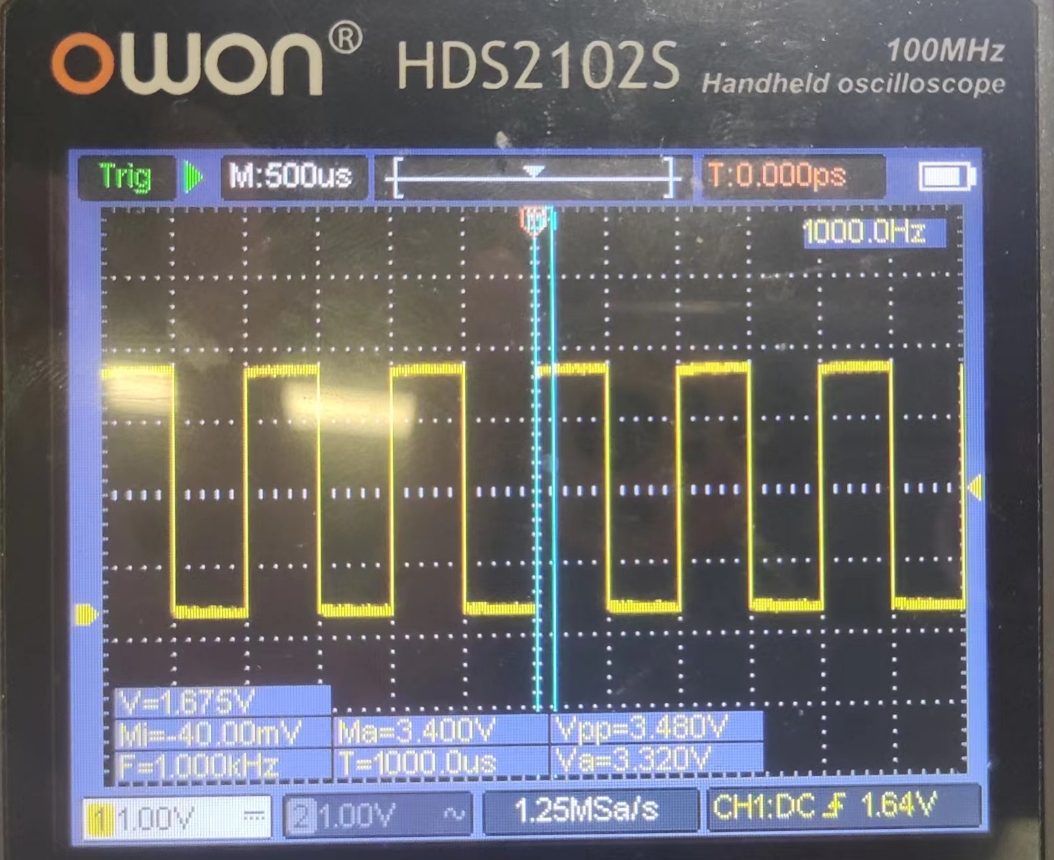
echo 0 > enable #打开pwm0

echo normal > polarity #设置正常极性

echo 1000000 > period #设置1ms周期

echo 500000 > duty\_cycle #设置0.5mszhankb

就会获得一个1ms，占空比为50%的pwm波：



总结：从gpio和pwm来看，VF2的sysfs是非常完善的，控制也比较准确、迅速、无误。可以仿照之前c语言编程实现一个pwm控制简单应用。