



西安交通大学
XI'AN JIAOTONG UNIVERSITY

课程：计算机控制实验

实验项目：实验一 基础前进和转弯控制

学生：郝世龙 学号：2171311028 班级：自动化 71

学生：田瑞峰 学号：2171311105 班级：自动化 71

学生：任泽华 学号：2171411498 班级：自动化 71

指导老师：景洲 联系方式：13720424008

实验地点：西一楼 A304

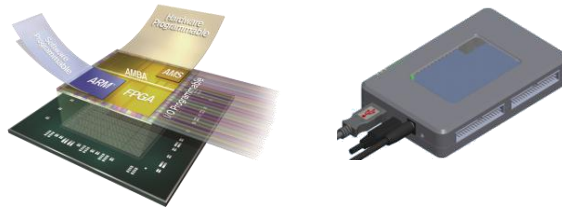
实验时间：2020/7/8

报告提交时间：2020/7/19

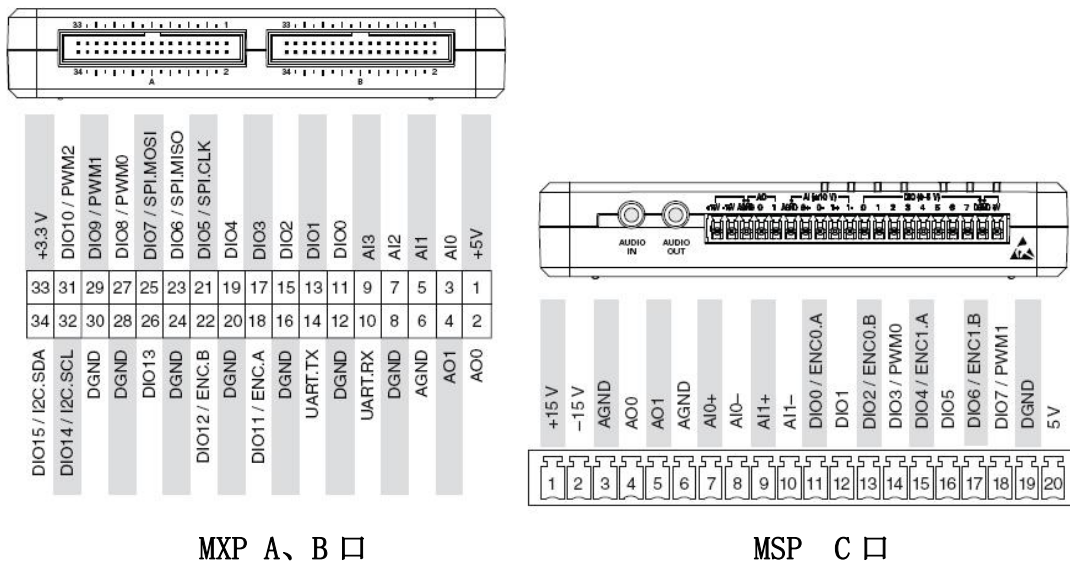
一、硬件框架及原理

1.NI myRIO-1900

NI myRIO-1900 的核心芯片是 Xilinx Zynq-7010，该芯片集成了 667 MHz 双核 ARM Cortex-A9 处理器以及包含 28K 逻辑单元、80 个 DSP slices、16 个 DMA 通道的 FPGA。



其中主要需要注意的是 myRIO 的端口配置：



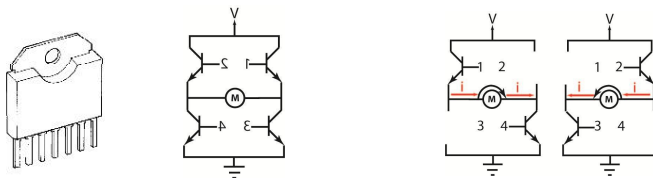
MXP A、B 口

MSP C 口

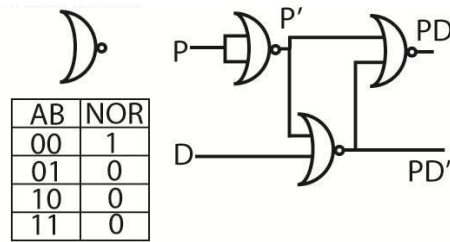
所有的外设都需要通过 MXP 口与 myRIO 相连，以共同完成一个任务。

2.电机驱动电路

1) H-桥 (TA8428K)

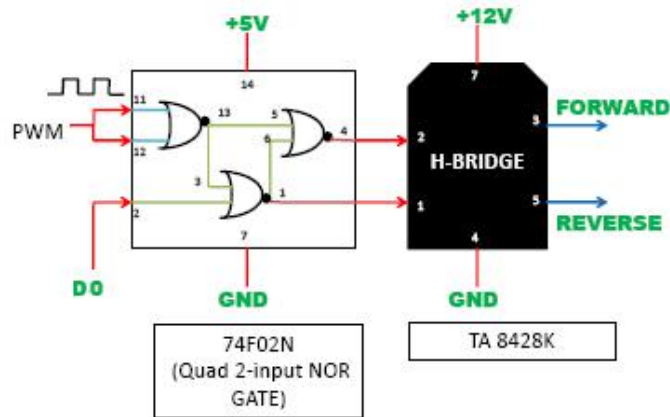


2)逻辑电路（或非门 74F02N）



这个电路的输入 P 为 PWM 信号，通过占空比的大小来控制小车两个轮的速度。D 为 D0 信号，来控制小车的前进和后退。通过真值表所示的组合或非逻辑判断，将信号输入到 H 桥进一步处理。

3)电机驱动电路

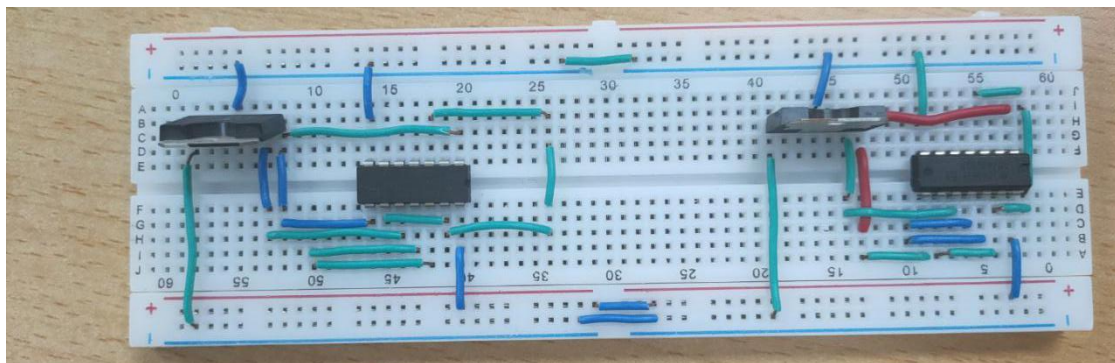


H 桥在接收前面或非逻辑组合提供的信号后，将其转换为控制电机的信号，FORWARD 为转动速度，REVERSE 为电机转动的方向。

3.连线图与说明

我们组的连线与其他组稍有不同，我们使用的是上面的两个与非门和下面左边的与非门。在连线时需要稍加注意，不影响使用。

在调试小车时我们的控制信号端口是随时定义的，按照小车情况和粘贴时端口的位置合理分配端口，这样加强了设计的灵活性，每次重新组装小车时只需修改 LabVIEW 端口参数即可。



二、LabVIEW 程序设计过程

1.原理和设计分析：

小车直线前进和后退需要保证左右两主动轮速度的一致，而转向则需要两轮之间出现差速，即左转需要右轮转速大于左轮转速，右转则需要左轮转速大于右轮转速。所以，直线前进或后退时，我们令小车两轮转动方向、转速完全相同；而左转、右转时，我们仍保持两轮转动方向相同，但令两轮之间转速出现差异，进而完成转向。

2.控制逻辑：

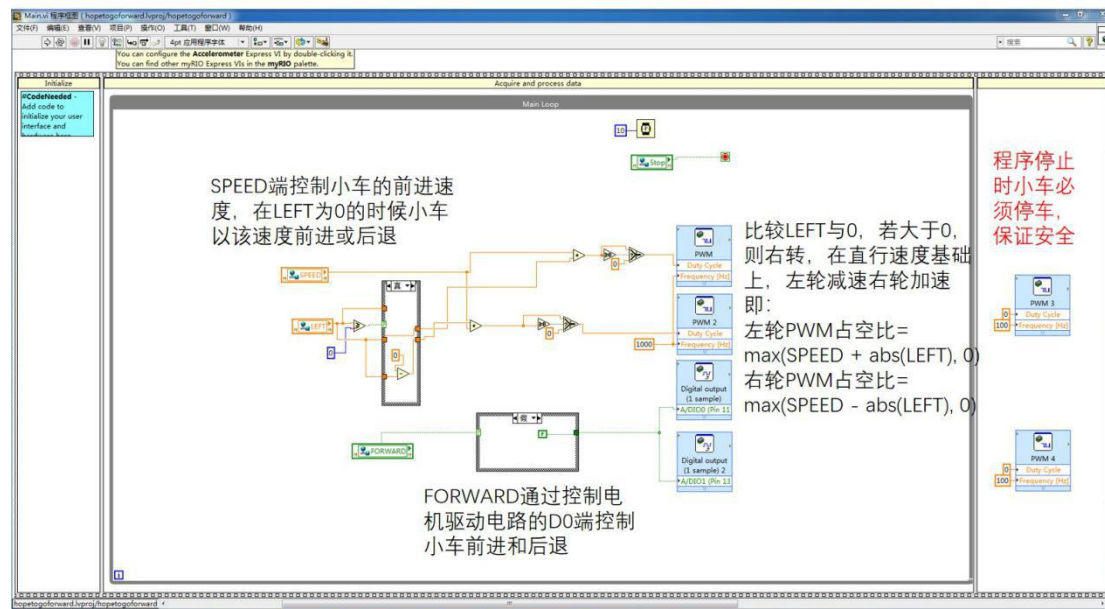
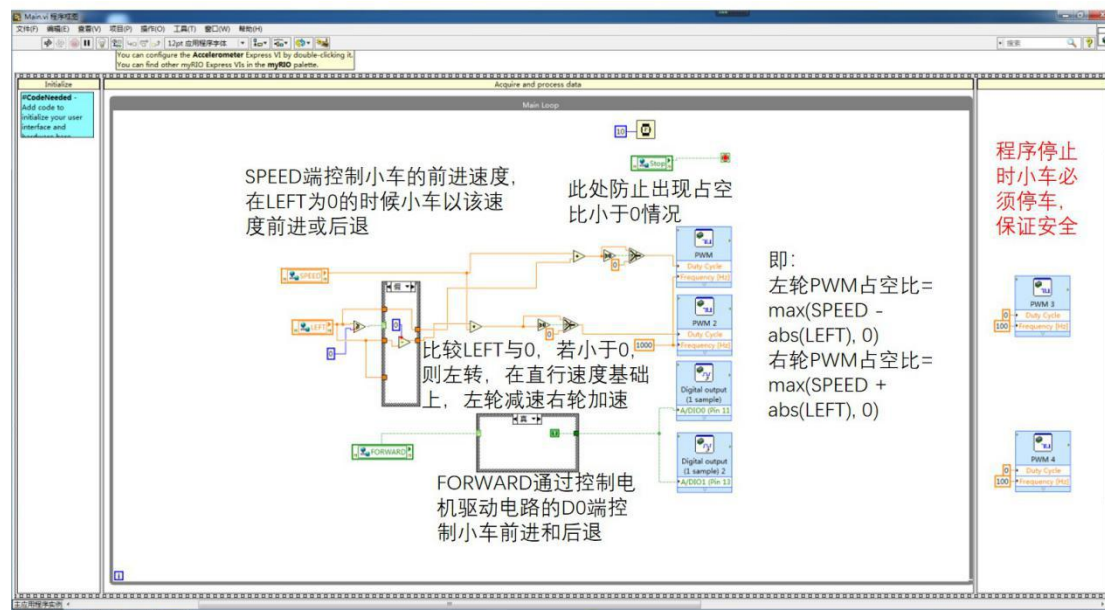
（为了方便解释，我们将给定的 PWM 大小直接等价于速度）

我们考虑使用 forward 按键给定 1/0 代表前进/后退，在条件判断后，同时向左右两轮 DO 端口给定 T/F 输入，进而同步控制小车的前进与后退，保证两轮转向始终相同（同时向前和向后）。同时，使用共享变量 speed 给定小车左右两轮相同的基本速度 v_0 ，这也就是小车直行状态下的前进速度，且同时给定保证了两轮转速始终相同。

此后添加转向部分功能，针对左转右转需要速度差这一条件，我们考虑使用共享变量 left 进行控制，其数值代表转向速度。不妨设 left 转向速度为 v_1 ，则当 v_1 小于 0 时，转向部分的条件判断为假，为左轮减去对应数值、右轮加上对应数值，同时在输入 PWM 端口前，再进行条件判断保证输入给定的转速不小于 0，保证两轮转向相同。所以这个情况下左轮速度为 $\max(0, v_0 - |v_1|)$ ，右轮速度为 $v_0 + |v_1|$ ，显然右轮更快，可以完成左转。同理，left 大于 0 时，转向部分条件判断为真，右轮速度为 $\max(0, v_0 - |v_1|)$ ，左轮速度为 $v_0 + |v_1|$ ，左轮更快，可以完成右转。

此外，按照我们的控制逻辑， v_0 ， v_1 为相互独立的变量，可分别给定。那么 $v_0 - |v_1|$ 和 $v_0 + |v_1|$ 也就是相互独立的变量。所以，我们还可以选择改变 v_0 、 v_1 的大小令小车有不同的转向姿态。 $v_0 < v_1$ 时，小车原地自转； $v_0 \geq v_1$ 时，小车在前进状态中进行转向。

3.程序框图以及说明



三、Data dashboard 设计过程

下为我们设计的 Data dashboard 测试电机以及手动控制的控制面板以及对面板的部分说明：



这里有两个控制面板，左边的是调试电机时的测试面板，主要控制两个电机，检查硬件是否有问题，测试软件编写有没有 bug。其中 motor 控制转动方向，duty 控制转速，在检验完成可以运行时，进入第二步。

右边的是小车的手动控制面板，为了方便控制转弯，我们将左转和右转集中到一个地方，之前在调试时发现两个分开会很难控制，而将左转定为负，右转定为正，在逻辑实现上也比较简单。我们也加入了急停下电的按钮，但是测试中发现如果不小心误触，会直接退出程序，还需要再次连接电脑下载，比较麻烦，所以在实际应用中将其去掉了。

四、控制参数的选取过程

控制参数主要是直行的速度以及转弯的速度两个参数。由于本次实现的是手动控制小车，所以我们需要选择一个较为合适的速度进行从而易于控制速度。

在实验的过程中，我们发现在小车速度较慢时才能走出较好的轨迹，所以速度我们尽量使得其走的略微慢一点才能够更好的进行操作来走 8 字或者走 U 形。所以在试验后，我们使用直行 PWM 波占空比 SPEED 为 0.3，而转弯时分为 3 档，LEFT=0.08, 0.16, 0.24，通过调节 LEFT 来控制速度，LEFT<0 为左转，LEFT>0 为右转。通过实际操作实验，我们可以得到一个较好的操作结果，能够在人手动操作的时候走出较为漂亮的 8 字形和 U 字形。

五、遇到的问题与解决方法

1. 触发 STOP 时电机失控

在前进过程中，我们在停下程序的时候，如果忘记将 SPEED 与 LEFT 在面板上调节为 0，在不小心中触碰了 STOP 按钮之后便会电机失控运行。所以我们在程序的平铺式顺序结构的 close 处将两个接到电机上的 PWM 波的占空比归零，这样我们在程序停止运行的时候使得小车停止从而避免电机失控。

2. 操作的时候转弯响应较慢，导致走 U 形的时候没法走出两条平行的线

这个操作主要因为手机端与 myRIO 之间的连接是借助 WiFi 来进行的，所以有一定延时，所以我们在操作的时候一定要进行预判，看到快转弯的时候一定要提前就将滑杆划到位但不放手，即将就位的时候放手，让电机进行变速，这样就能响应很好，划出较为标准的轨迹。

3. 小车转弯时候不能转大弯

原来程序的设计上，转弯时的设计是一个轮子维持不转而另一个轮子运动来完成转弯，但是实际上这样的转弯无法转大弯。所以必须采用差速的方式，就是在直行速度基础上给两个电机制造一个差速来使得其转弯。这样的转弯方式适合手动控制转弯大小。但这种方法实际上是不适合后面的循迹和避障的，只是比较适合手动转大弯，所以只在这里有所使用。

六、实验总结与建议

这次实验是计算机控制的第一次实验，我们主要对于 myRIO 有了初步的认识，对于 LabVIEW 图形化编程有了进一步的了解。对于移动机器人这门课中提到的一些控制方法进行了验证，实现了理论与实践的结合。在操控小车的过程中我们发现，经常打游戏的同学控制起来比较得心应手，还可以手动操控小车实现巡线任务，而我们在经过不断地摸索和适应后才可以走出比较完美的 U 型和 8 字曲线。