桥式行车防摇摆控制系统

Antisway Control System for Overhead Crane

李少彬

（瑞钢钢板中国有限公司 SSAB APAC，昆山）

摘要：行车在运行过程中总是不可避免地造成吊物的摇摆，通常需要非常熟练的行车操作工手动操作控制吊物的摇摆，这也是目前最为常用的做法。吊物的摇摆会加速机械磨损，增长吊物的转运时间，甚至造成安全事故。因此多种防摇摆控制策略已经开发多年，防摇摆控制可以自动消除吊物在运行过程中产生的摇摆，可以更快地完成吊物的转运，特别是带有定位功能的自动化行车，防摇摆系统可以使行车的操作变得更高效、更安全。

关键词：防摇摆、高效、安全

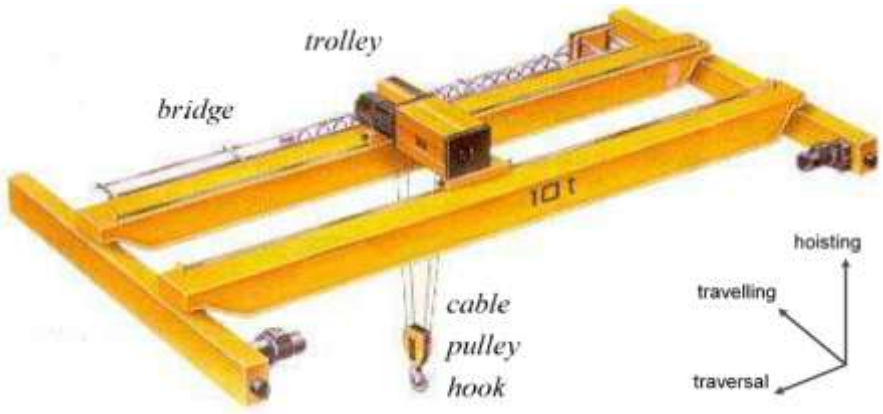
Abstract: When operating a crane, a state of sway is more natural than a state of equilibrium. A skilled crane operator will eliminate sway manually, and by far this is still the most common solution to the antisway problem. The sway will shorter the mechanical life of the crane and increase the transport time. Even cause safety accident. Antisway strategies of different kinds have been devised for many years. An antisway system may often do the job faster, especially if the antisway is combined with automatic positioning.

Key Words: Antisway、Efficient、Safety

一、防摇摆理论

1. 摇摆的产生

行车在加减速的过程中，吊物的运动总是落后行车的运动，从而在行车与吊物间形成夹角，在重力的作用下，吊物会形成来回摇摆的现象。



2. 摆动周期

著名物理学家伽利略在比萨大学读书时，对摆动规律的探究，是他第一个重要的科学发现。有一次他发现教堂上的吊灯因为风吹而不停地摆动。尽管吊灯的摆动幅度越来越小，但每一次摆动的时间似乎相等。

通过进一步的观察，伽利略发现：不论摆动的幅度大些还是小些，完成一次摆动的时间（即摆动周期）是一样的。这在物理学中叫做“摆的等时性原理”。各种机械摆钟都是根据这个原理制作的。

后来，伽利略又把不同质量的铁块系在绳端作摆锤进行实验。他发现，只要用同一条摆绳，摆动周期并不随摆锤质量的影响。随后，伽利略用相同的摆锤，用不同的绳长做实验，最后得出结论：摆绳越长，往复摆动一次的时间（即摆动周期）就越长。

公式 ：$T=2\pi\sqrt{(l/g)}$

.

3. 防摇摆方法

行车防摇摆方法主要有机械防摇摆、电气控制防摇摆。机械防摇摆需要增加机械设备造成行车自重增加且造价昂贵，因此很少采用。电气控制防摇摆分闭环控制和开环控制，闭环控制需要有摆角测量装置，配合控制算法实现。开环控制无需角度测量装置，通过控制算法即可实现对行车的防摇摆控制，因此也是目前使用最广泛的防摇摆方法。开环防摇摆的控制算法有好几种，本控制系统采用最为常用的一种控制方法，双脉冲前馈控制算法。

4. 双脉冲前馈防摇摆控制原理

吊物的摆动是由于行车的加速或减速而造成的，经专家研究当行车以一定加速度运行时，吊物会以一定摆动周期（T）摆动，在一半摆动周期（T/2）时施加一个等量等时的短脉冲后，吊物形成的摆动将会消除，因此该控制方法称为双脉冲前馈防摇摆控制（Double pulse control），也有称为 Posicast control 或 Cancellation control，最初该控制理论是由 O.J.M Smith 在 1958 年提出。

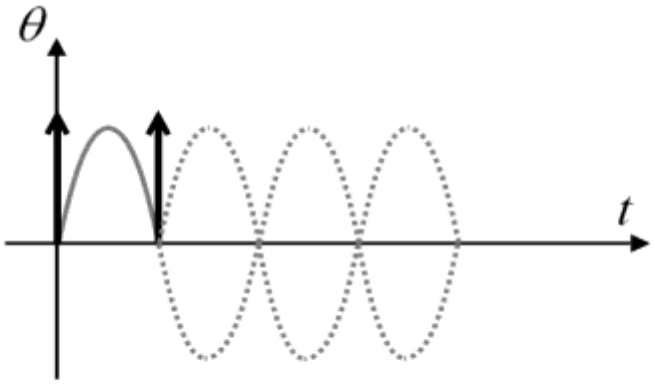
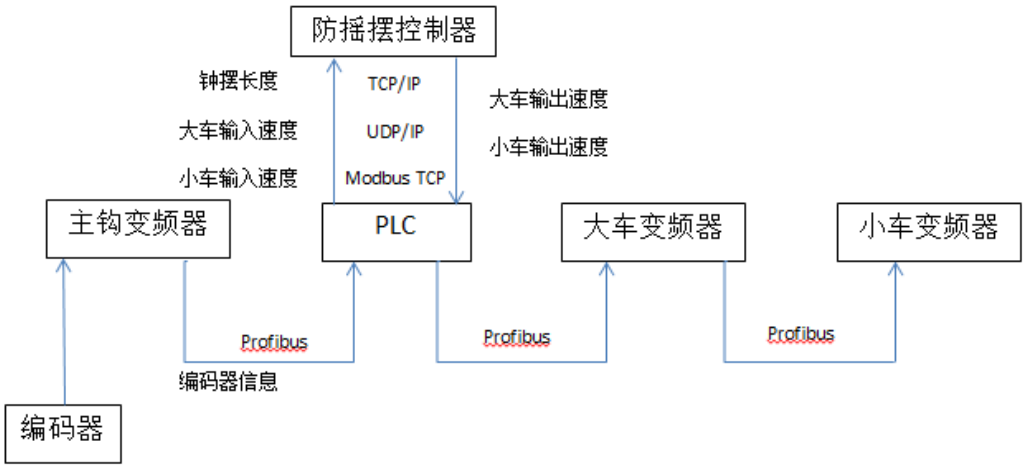


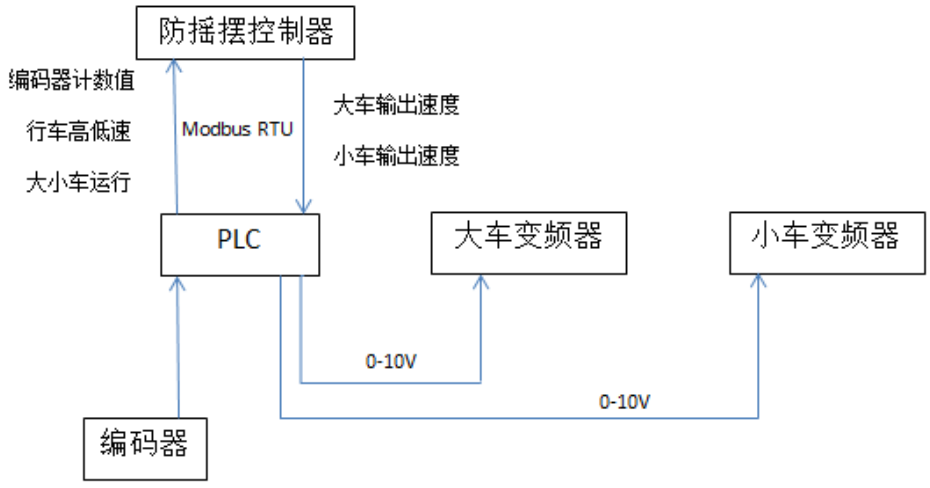
Figure 1.4: Double pulse sequence

二、防摇摆控制系统组成

防摇摆控制系统的核心为基于 X86 架构的工控机或 ARM 架构的其它嵌入式控制器，采用开源控制系统平台 Proview。Proview 支持高级语言编程（C/C++/JAVA），可将复杂的控制算法通过高级语言编程并连结到 Proview 系统平台，甚至可以将其直接写入 Proview 软件的源代码，成为软件标准功能。Proview 系统支持多种通信模式，可通过以太网使用 TCP/UDP 或 Modbus TCP/Profinet 等协议和其它控制系统通信，同时支持 Profibus/Modbus RTU 等现场总线协议。因此该防摇摆控制器可以用于支持网络的 PLC 控制器的自动化行车，也可以用于简易的手动行车改造。



带以太网口 PLC 的自动化行车防摇摆方案



简易手动行车防摇摆方案

1. 自动化行车防摇摆

行车主要配置：支持标准以太网或 Modbus TCP 的 PLC（西门子 CPU315-2 PN/DP），带 Profibus DP 通信口的变频器，X86 架构的工控机。行车的控制主要由 PLC 来完成，PLC 可通过主钩变频器获取吊物的高度信息，同时从操作者或上位机得到大车和小车的运行速度命令，PLC 将获取的摆长、大车速度和小车速度通过以太网（UDP/IP 协议）传送到防摇摆控制器（运行于 Linux 操作平台的 X86 工控机），防摇摆控制器获取双脉冲前馈防摇摆模型的参数后，经过运算将大车和小车的输出速度通过以太网回传至 PLC，PLC 通过 Profibus 将速度输出传输到大车变频器和小车变频器，实现防摇摆控制。

2. 简易手动行车防摇摆

行车主要配置：带串口支持 Modbus RTU，具备 0－10V 模拟量输出的小型 PLC(西门子 200PLC 或其它小型 PLC)，支持模拟量输入的变频器，微型控制器（树莓派 B+）。PLC 主要用于采集主钩增量型编码器的脉冲数，大车、小车运行等指令，通过 Modbus RTU 将采集到的信息传输到防摇摆控制器（树莓派），防摇摆控制器将运算的大小车速度输出通过模拟量输出 0－10V 到变频器，实现防摇摆控制。

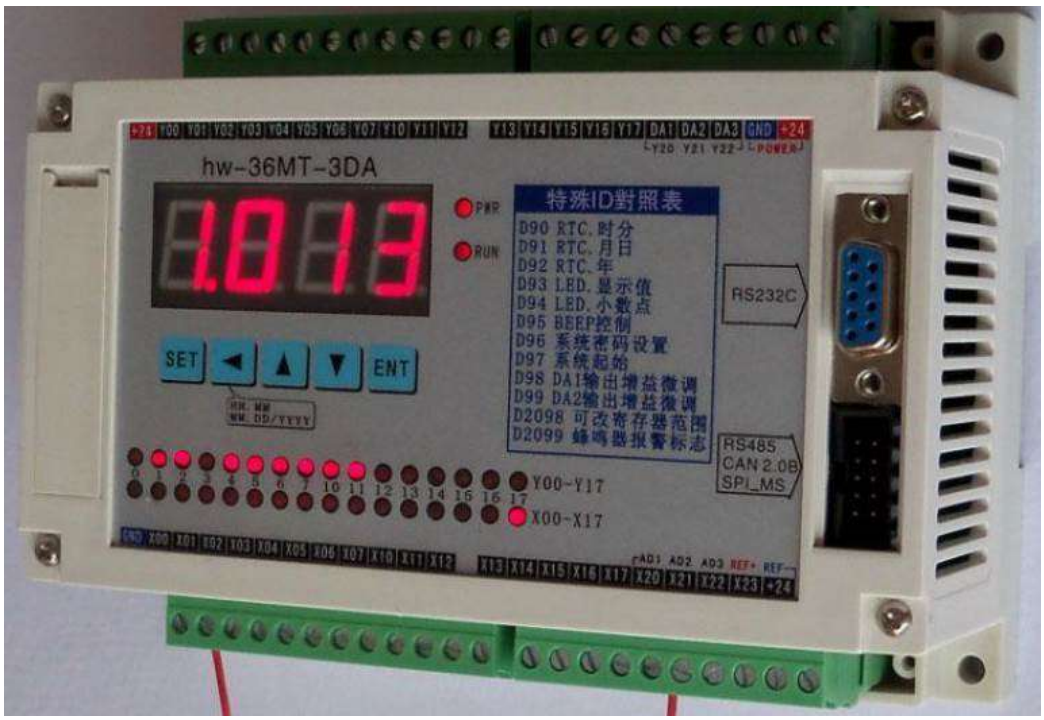
2.1 使用硬件

2.1.1 防摇摆控制器－－树莓派 B+

Raspberry Pi 型号 B+ 是信用卡大小的计算机板，当添加了键盘、鼠标、显示屏、电源和已安装 OS 的 MicroSD 卡时，可以启动和运行。 它是基于微型 ARM 的 PC，可运行许多通常需要台式 PC 的应用，如电子表格、文字处理和游戏。 它还可播放高清视频。与较早型号相比，除其他改进之外，型号 B+ 均具有低功耗、更佳的音频性能和 40 引脚 GPIO 连接器。 Raspberry Pi 是开源产品，设计为可得到基于互联网的用户论坛支持。 有关入门信息，请参见官方网站：http://www.raspberrypi.org/。

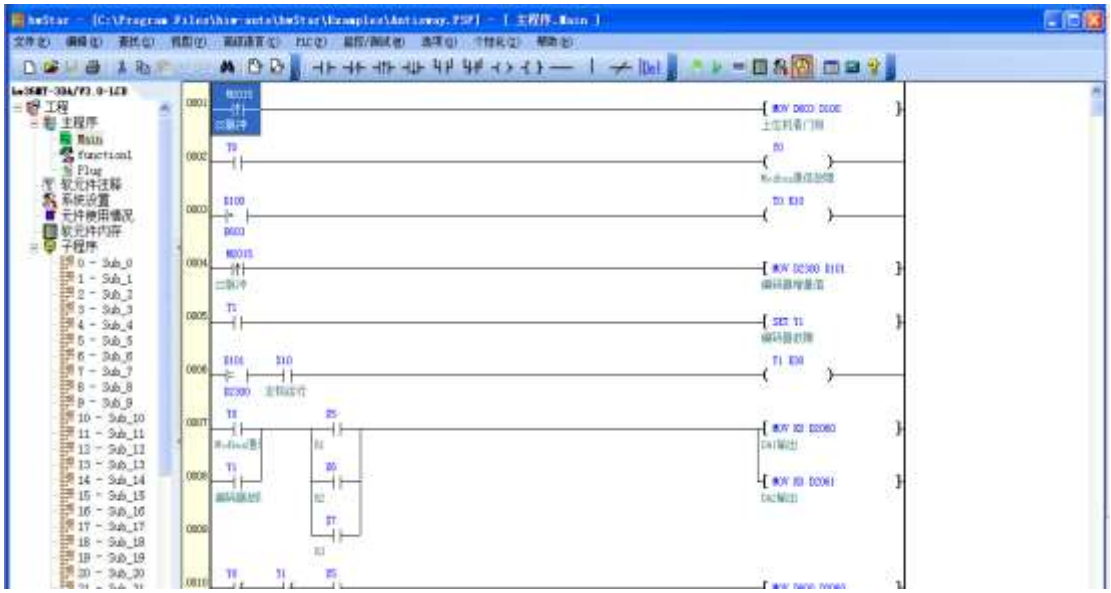
2.1.2PLC

普中科技 PLC，型号 PZ-36MT-3DA。20 点输入，16 点输出，3 路模拟量输出，内置 2 组 AB 相正交编码器计数，标配 8.5KHz。通信接口 RS232C、RS485、SPI_MS，支持 Modbus RTU 通信。

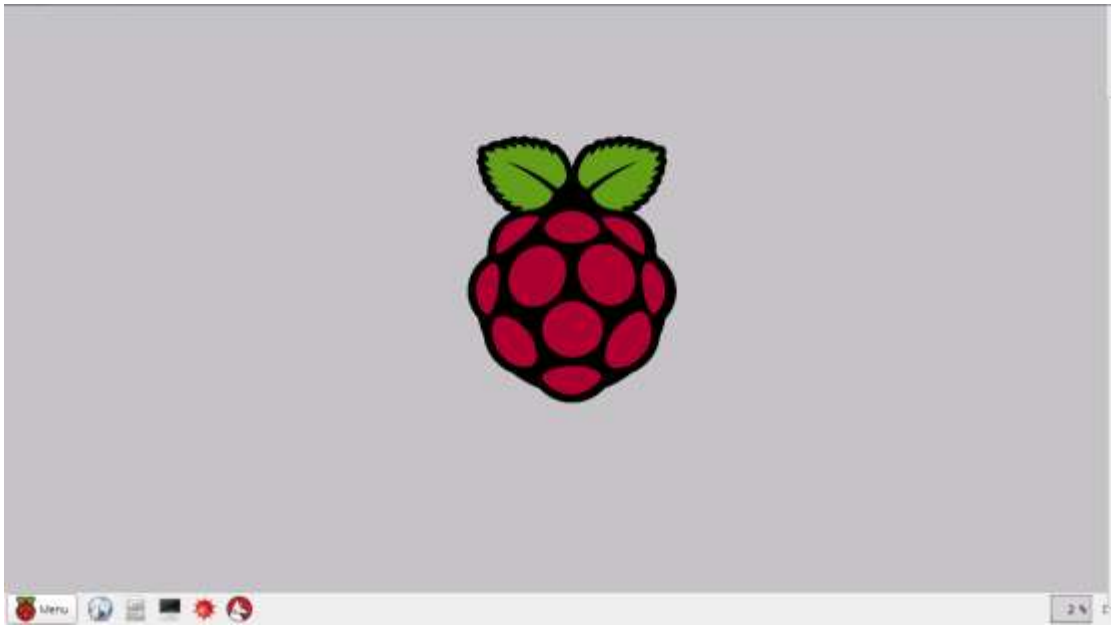


2.2 软件环境

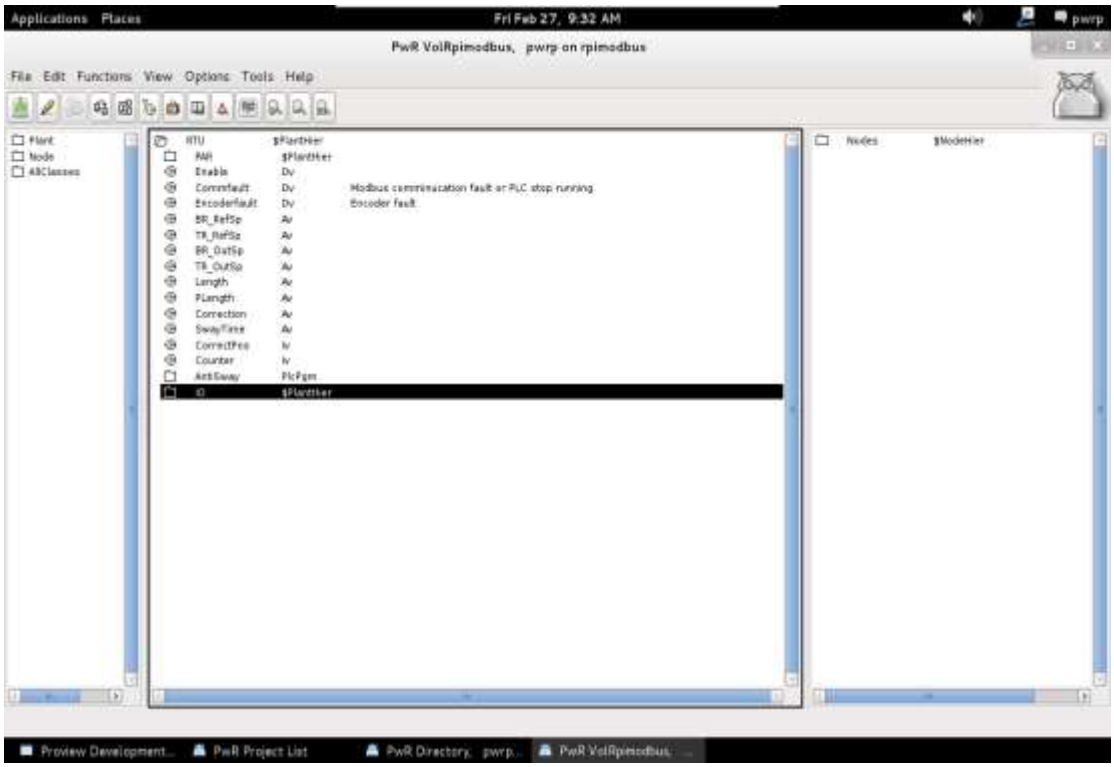
2.2.1 普中 PLC 编程软件 hwStart，可至普中科技官网下载 www.prechin.com



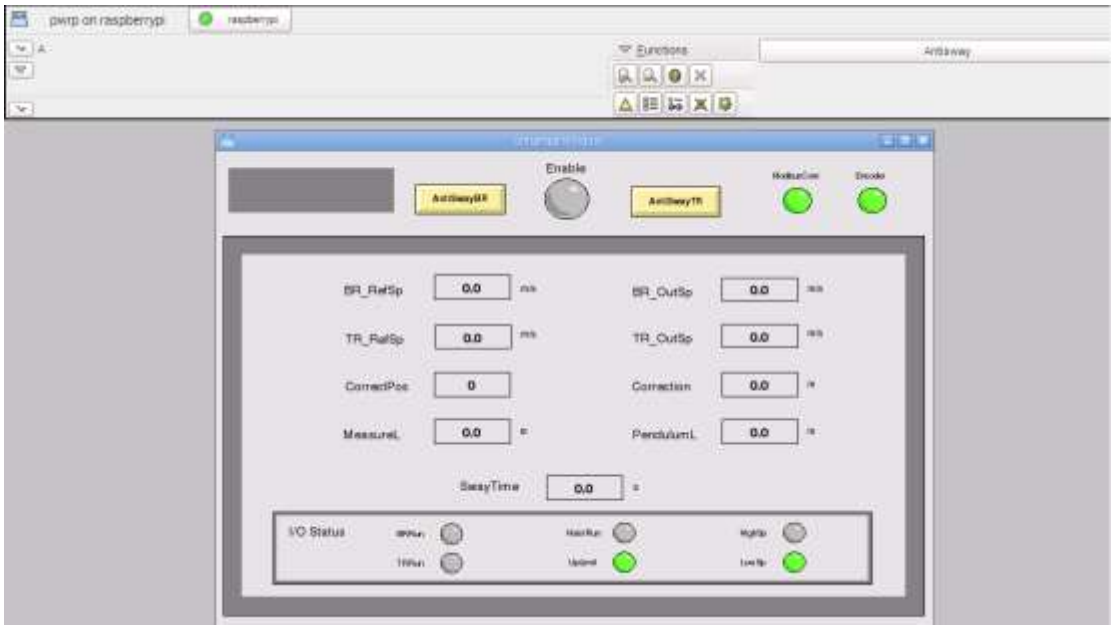
2.2.2 树莓派 B+操作系统 RASPBIAN，可至树莓派官网下载 http://www.raspberrypi.org/



2.2.3 开源过程控制软件 Proview，可至 Proview 网址下载 www.proview.se，软件包括开发包 pwr52_5.2.1-1_i386.deb)，树莓派开发包 pwrrpi52_5.2.1-1l386.deb，这两个开发包需要安装在一台基于 Linux 系统的 PC 上，目前支持 Debian 和 Ubuntu 发行版，其它发行版需要 pwrsrc 包，使用 Proview 源码安装。



树莓派实时运行包 pwrrt_5.2.1-1_armhf.deb，该包直接安装在树莓派上，用于运行 Proview 的实时环境。



2.3 基于简易行车防摇摆控制方案的实施

2.3.1 PLC 侧

（1）I/O 配置

数值量输入端：X00（主钩上限位）,X01（低速选择未使用）,X02（高速选择）,X03（大车运行）,X04（小车运行）,X05/X06/X07（摆长补偿长度），X10（主钩运行），X20/X21（主钩编码器 AB 相）

数字量输出端：Y00（Modbus 通信错误），Y01（编码器错误）

模拟量输出端：DA1（大车输出速度），DA2（小车输出速度）

（2）MODBUS RTU 通信地址对照表

寄存器名	通道范围	读写属性	数据类型	变量类型
0x	1 ~ 5200	读写	BIT	M000-M5119
1x	1 ~ 378	只读	BIT	X000-X377
3x	1 ~ 400	只读	Ushort	T000-T399
4x	1 ~ 5200	读写	Ushort/Long	D000-D5119

（3）PLC 程序控制逻辑

PLC 主要作为防摇摆控制的执行端，采集防摇摆控制器树莓派所需的输入参数，并获取树莓派防摇摆模型运算的输出速度，通过模拟量输出端口 DA1 和 DA2 输出 0-10VDC 电压分别

送至行车大车和小车的变频器模拟量输入，从而控制行车的大小车的运行速度，达到防摇摆的目的。

防摇摆控制模型主要输入参数摆长 L,通过 PLC 的 X20,X21 端口收集主钩编码器计数值，树莓派将通过 MODBUS RTU 获取该计数值，换算成主钩摆长。X00 主钩上限位用来校准编码器计数值，到达上限位，编码器计数值将清零，同时 X00 还用于复位编码器错误，如果编码器产生错误，计算的主钩摆长将会是错误的值，错误的摆长将会使防摇摆模型运算出错误的速度输出，造成吊物摇摆无法控制，因此，当编码器产生错误时，行车只能在无防摇摆模式下人工操作控制摇摆，只有编码器故障解除后，通过主钩上限位 X00 校准后才允许运行在防摇摆模式下。当主钩上升或下降 X10 激活，超过 2 秒后 PLC 读取的编码器计数值未增加或者减少，可判断编码器读数出错，PLC 禁止运行在防摇摆模式，同时 Y02 编码器错误输出，提醒操作人员，编码器错误，操作人员只能切换到无防摇摆模式才能继续操作行车。

行车高低速运行，X01 低速运行（未使用），X02 高速运行，X02 激活时，行车以 100%高速运行，X02 未激活时，行车以 50%低速运行。

大小车运行指令 X03/X04，用于启动大小车速度的输出，X03 未激活时，大车速度输出 DA1 输出电压为 0VDC,X03 激活时,根据行车高低速状态输出 10VDC 或 5VDC(无防摇摆模式)，或根据防摇摆控制器运算的速度输出相应的电压（防摇摆模式）。

摆长补偿 X05/X06/X07，选择开关用于激活防摇摆模式和摆长补偿。

8-step rotary switch

Switch position	Binary inputs			Pendulum length correction
	RSI3	RSI2	RSI1	
0	0	0	0	Sway control OFF
1	0	0	1	0.0 m
2	0	1	0	0.5 m
3	0	1	1	1.0 m
4	1	0	0	1.5 m
5	1	0	1	2.0 m
6	1	1	0	3.0 m
7	1	1	1	4.0 m

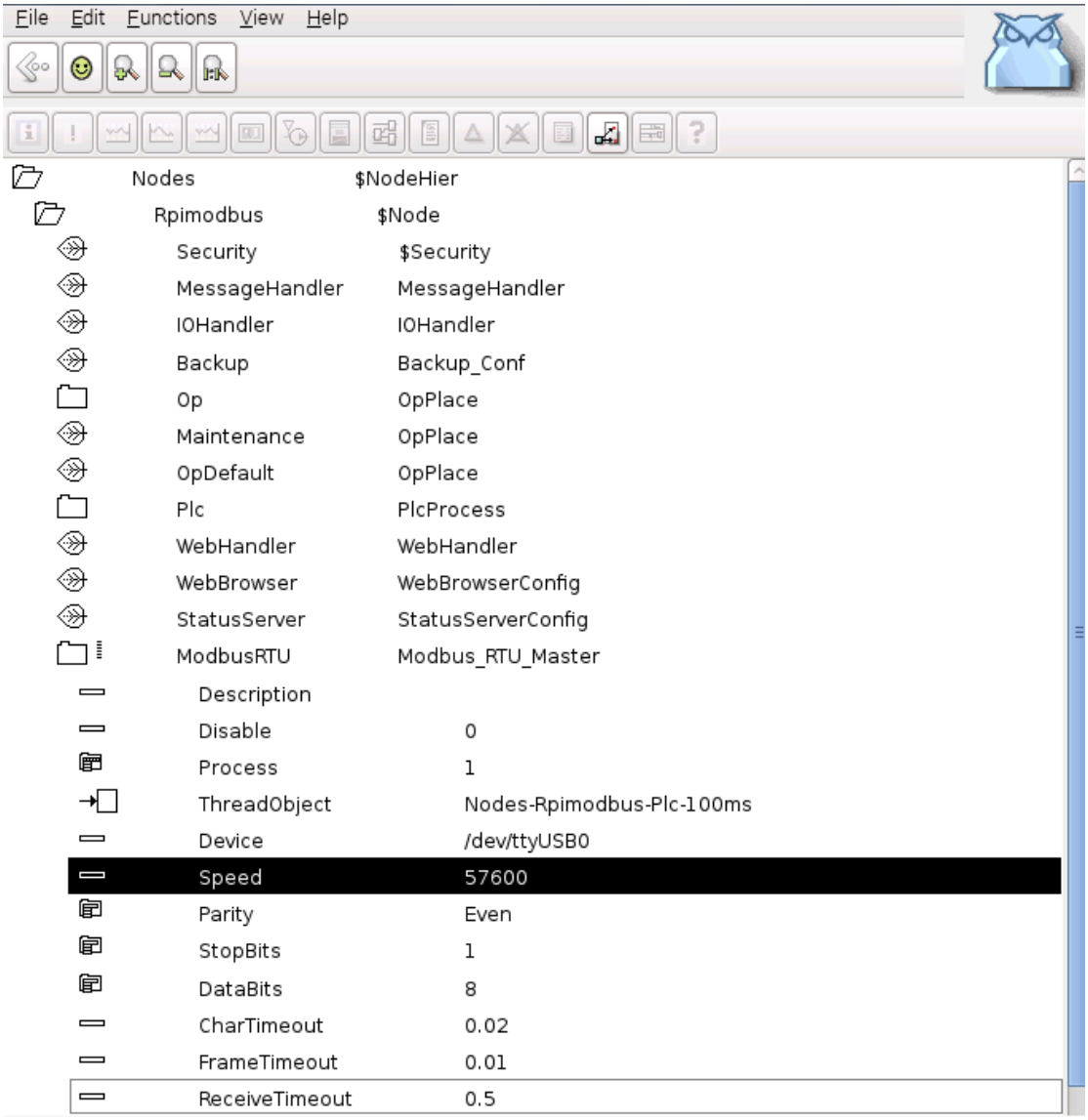
Modbus 通信故障报警 Y00，PLC 与防摇摆控制器通信发生中断后，PLC 无法正确获取防摇摆控制器运算出的速度输出，造成输出速度错误，无法控制吊物摇摆，因此当通信故障时，停止行车的运行并发出通信故障报警，提醒操作者，此时只能切换到无防摇摆模式下运行。

2.3.2 树莓派侧

树莓派作为防摇摆控制系统的核心，采集 PLC 的信号，经过防摇摆模型运算后传送至 PLC，通过 PLC 实现行车的防摇摆控制。树莓派的操作系统为基于 Debian Wheezy 的专用系统，控制软件为开源的 Proview，使用 Modbus RTU 协议采集 PLC 信息，树莓派作为客户端，PLC 作为服务器，树莓派通过 USB 转 RS232C 接口连接 PLC 的通信口，实现基于 RS232C 的 Modbus RTU 通信。

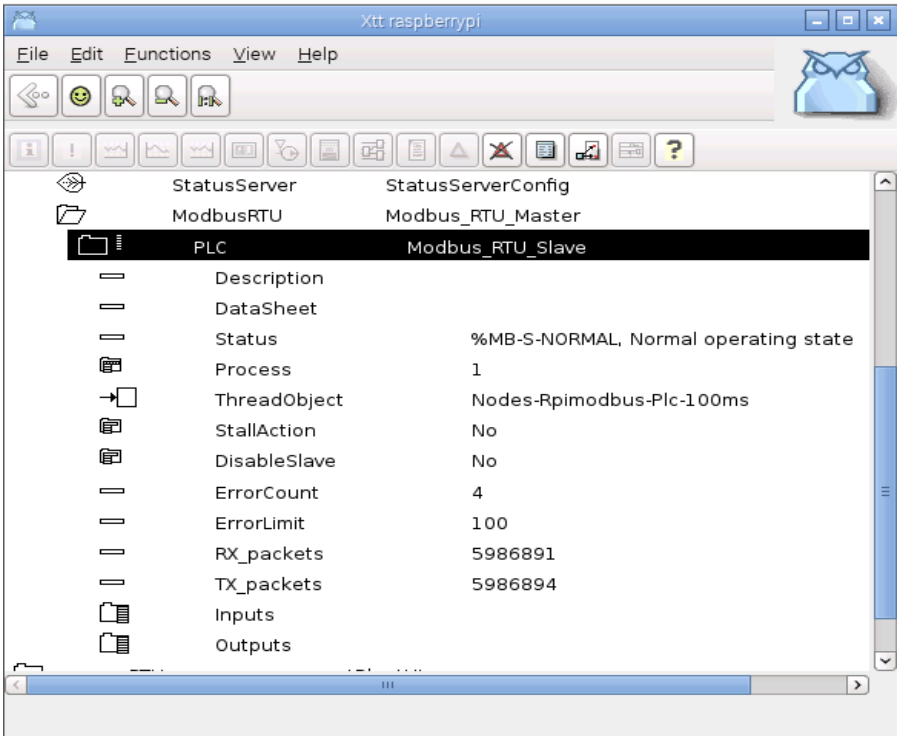
（1）Proview 与 PLC 的 Modbus RTU 通信的实现

在 Nodes 下插入 Modbus_RTU_Master 对象，根据 PLC 的端口设置参数设置相应的端口参数。

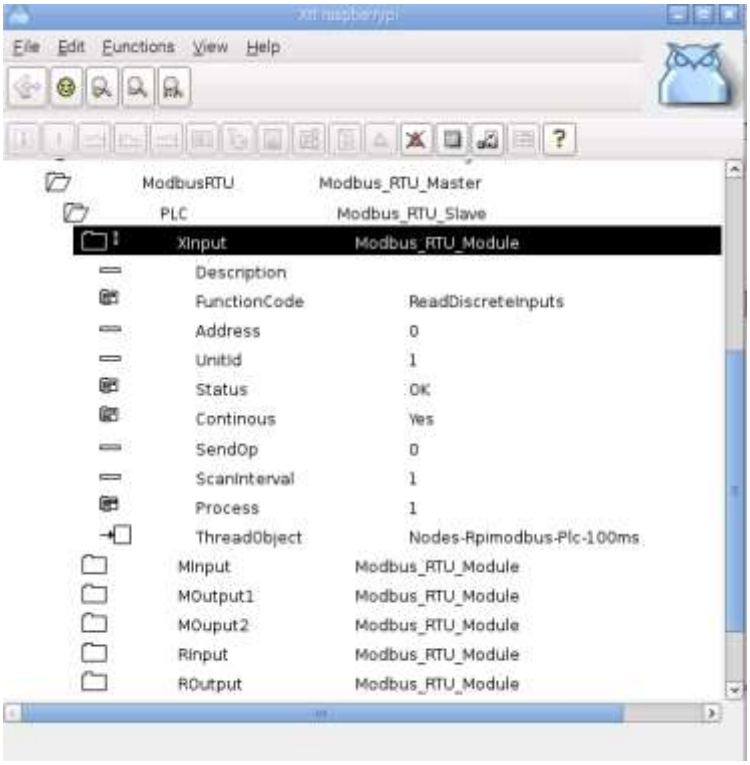


对应的 PLC 端口参数设置如下：

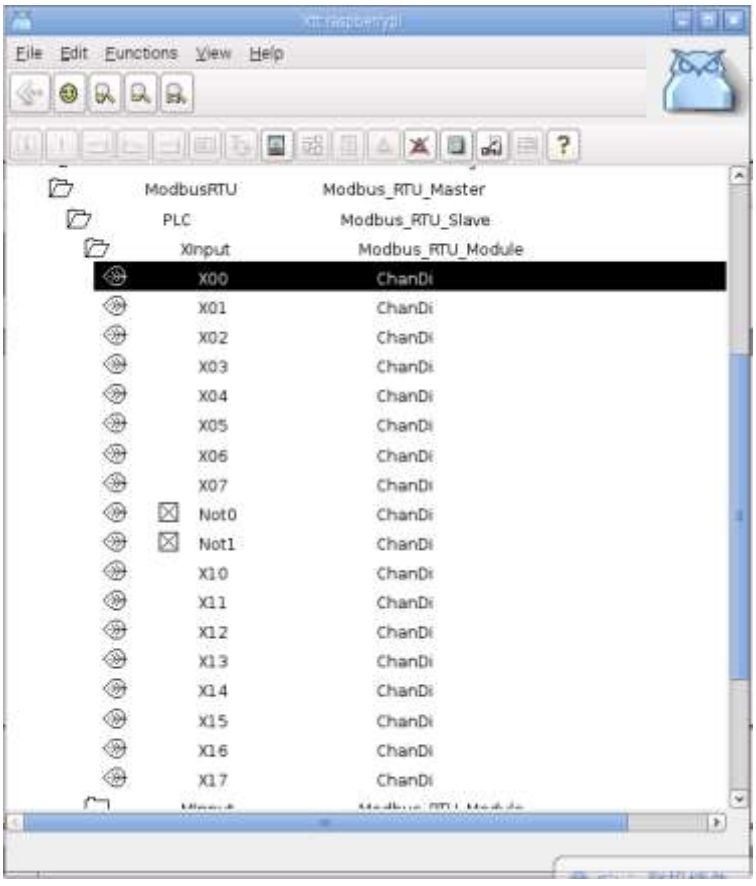
在 Modbus_RTU_Master 下插入 Modbus_RTU_Slave 对象。



在 Modbus_RTU_Slave 下插入需要的 Modbus_RTU_Module，使用 Modbus 相应的功能码读取 PLC 的寄存器值。

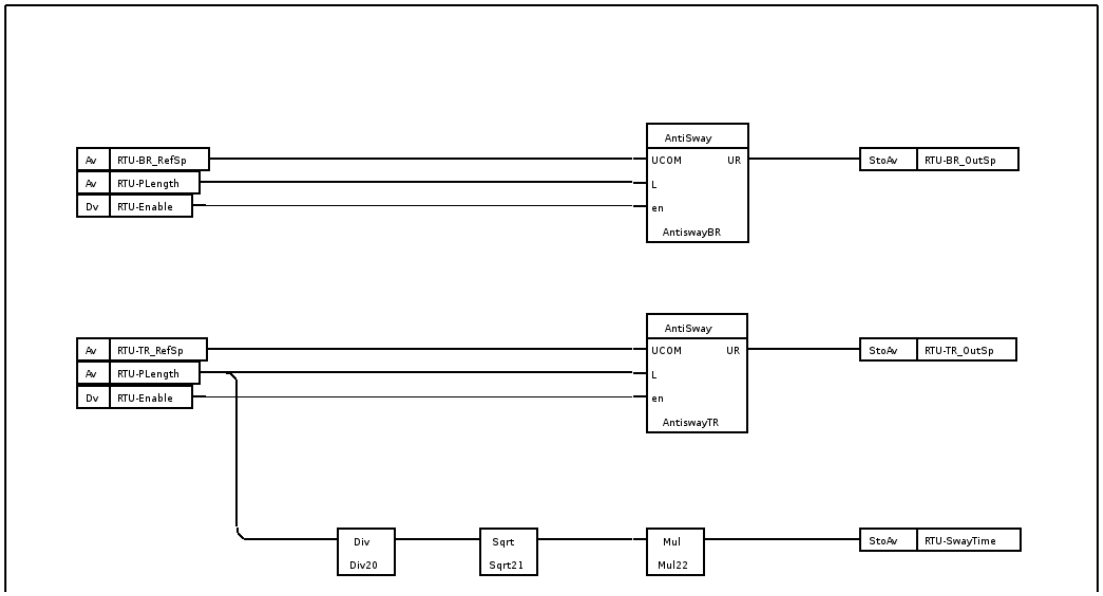


在 Modbus_RTU_Module 插入相应的 Channels 对象读取 PLC 相应的寄存器值。

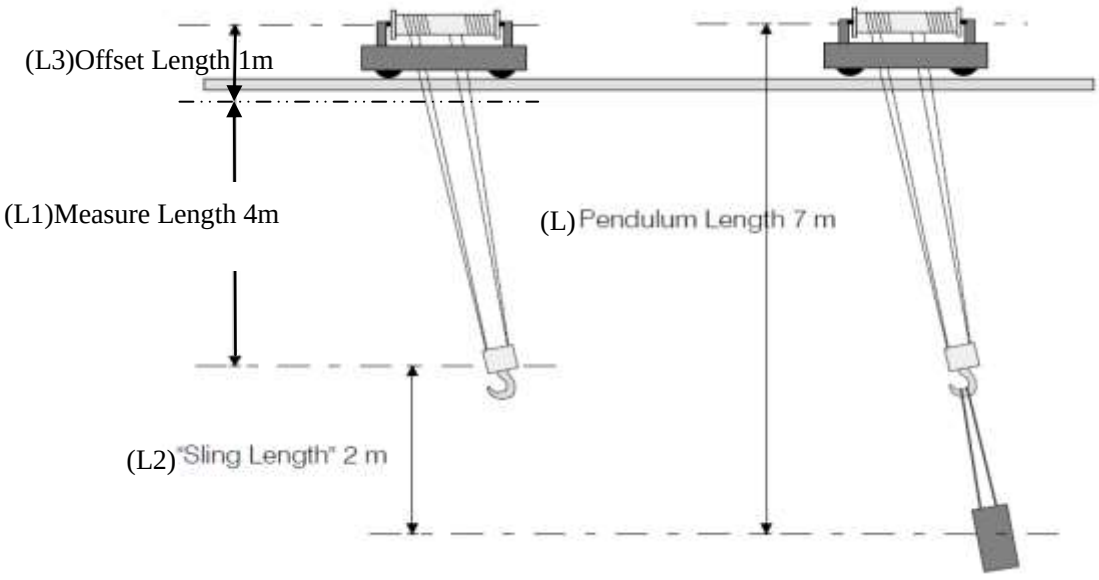


（2）Proview 程序控制逻辑

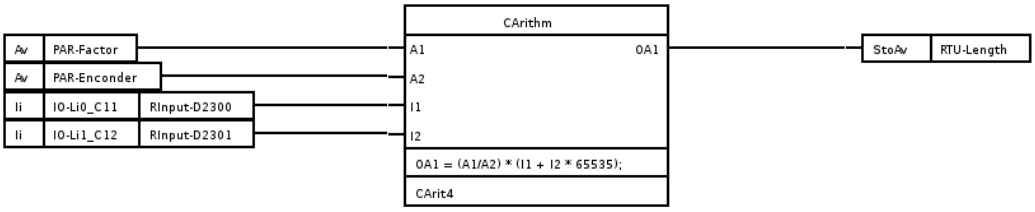
Proview 侧主要是使用防摇摆控制功能运算出相应的速度输出通过 Modbus RTU 传送至 PLC。Proview 系统自带防摇摆控制功能 Ssab_AntiSway，该功能块实现双脉冲前馈防摇摆控制功能。



吊物摆长 L 的计算：L=L1+L2+L3（单位 M）,L1 为编码器获取的数值运算的长度，L2 为从选择开关选择的补偿长度，L3 为预设长度。L 对应 Proview 的变量为 RTU-PLength，L1 对应为变量 RTU-Length，L2 对应变量为 RTU-Correction，L3 对应变量为 RTU-PAR-Offset。在现场实际调试计算摆长长度时，有 3 个参数可以在实时系统里直接调整，计算出正确的摆长长度。主钩上升至上限位后，编码器数值清零，L1 为 0M，实际上吊物仍然有一定的摆长，需要用 Offset 参数 L3 去调整。

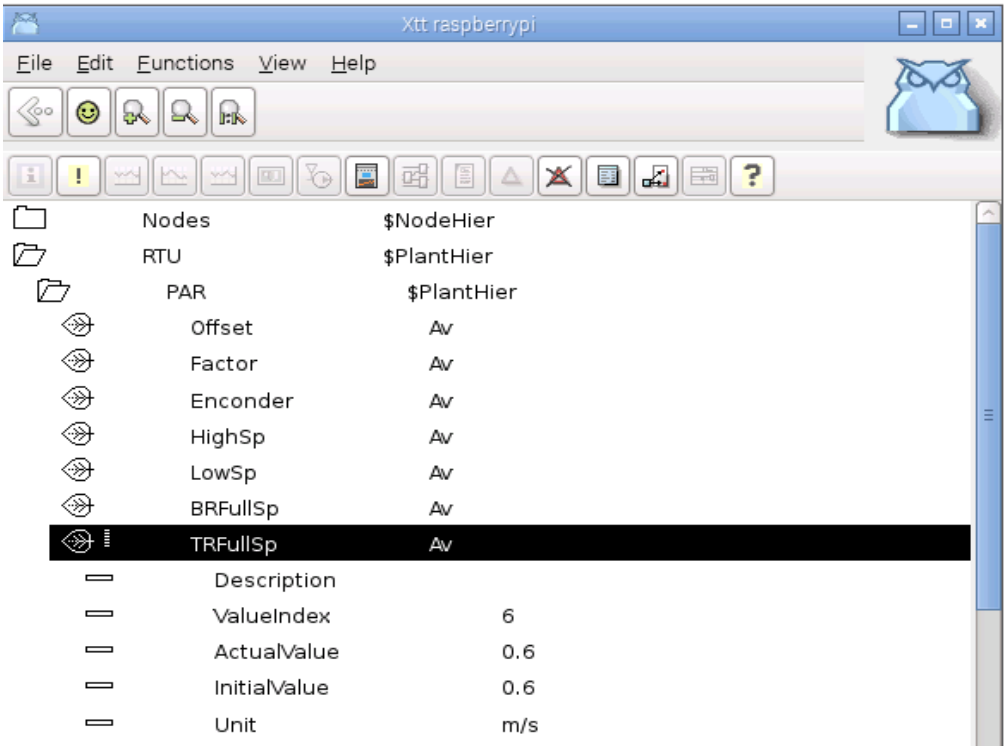


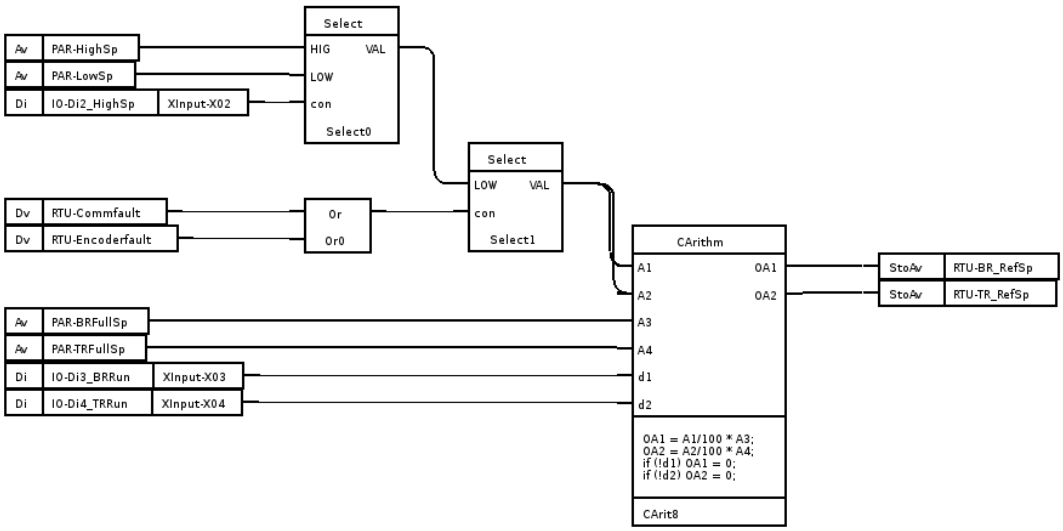
测量长度 L2 的计算需要用到 2 个参数 RTU-PAR-Factor（主钩旋转一圈吊物上升或下降的高度，单位为 m，预置为 2m/圈，根据实际情况调整），RTU-PAR-Encoder（主钩旋转编码器的脉冲数，预置为 1024/圈，根据实际情况调整）。参数设置后，程序则可算出编码器测量长度。



钟摆长度的确定，在无吊物的情况下使用无防摇摆模式开动行车，吊钩会形成来回摆动，使用计时器记录 10 次来回摇摆的时间，求平均值。比如 10 次的总时间为 40 秒，则摆动周期为 4 秒，根据 $T=2\pi\sqrt{(l/g)}$ 可求得摆长 L＝3.98 米。假如编码器测得的长度为 2.5 米，那么偏移长度参数 Offset 为 1.48 米（L3=L-L1-L2=3.98-2.5-0=1.48）。

除了钟摆长度外，防摇摆模型还需要给定速度，目前使用高低速两种速度，低速为全速的 50%，高速为 100%，高低速参数设定为 RTU-PAR-HighSp 和 RTU-PAR-LowSp，由于防摇摆模型速度设定单位为 m/s,所以高低速需要转换成 m/s，因此需要设定大车和小车全速的速度，这两个参数分别为 RTU-PAR-BRFullSp 和 RTU-PAR-TRFullSp，根据行车的实际运行速度设定，程序将自动将百分比的速度转换为实际运行速度。



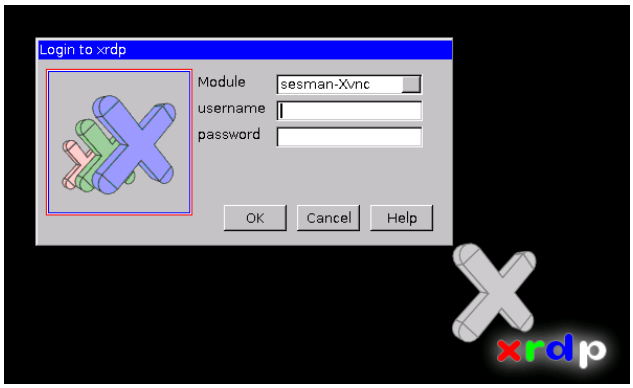


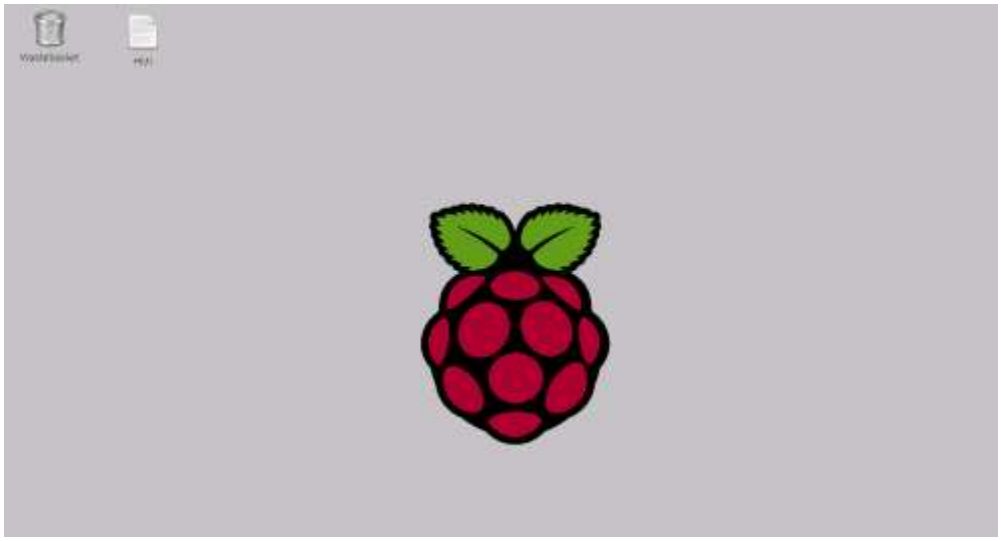
2.4 防摇摆控制系统的使用

2.4.1 防摇摆控制器的连接及 HMI 使用

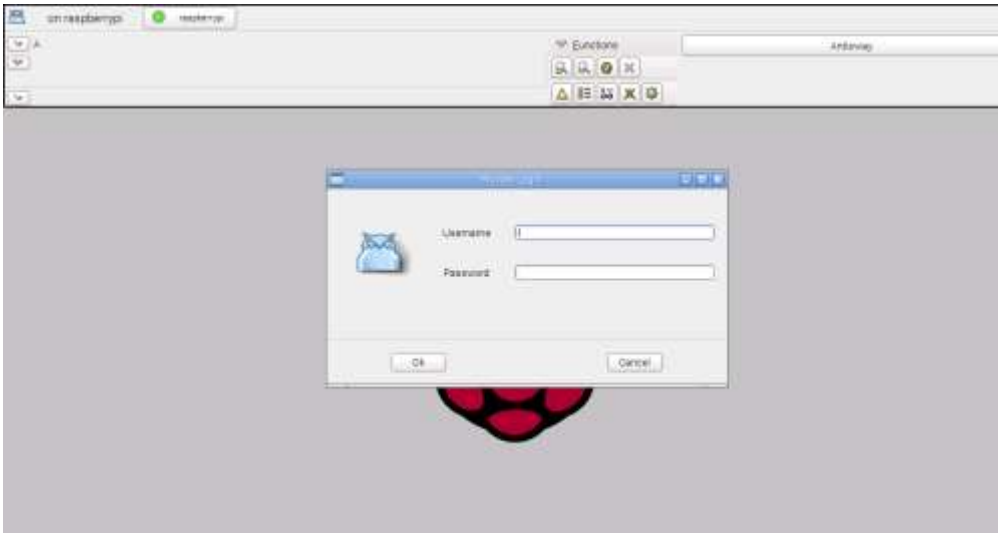
防摇摆控制器树莓派为微型 PC，可外接鼠标键盘及显示器，打开 HMI 画面，对防摇摆系统进行监控及调整。另外也可以通过远程桌面连接树莓派，进行监控调整，无需鼠标键盘及显示器，对于现场调试更为方便。以下介绍通过远程桌面连接的方法：可以使用网线直接连接笔记本和树莓派，双方都设置为固定 IP 地址，同一网段，在现在进行监控及调试，也可以通过网络进行远程调试。

点击 Connect 按钮连接，连接树莓派，输入用户名和密码登录，进入桌面环境。

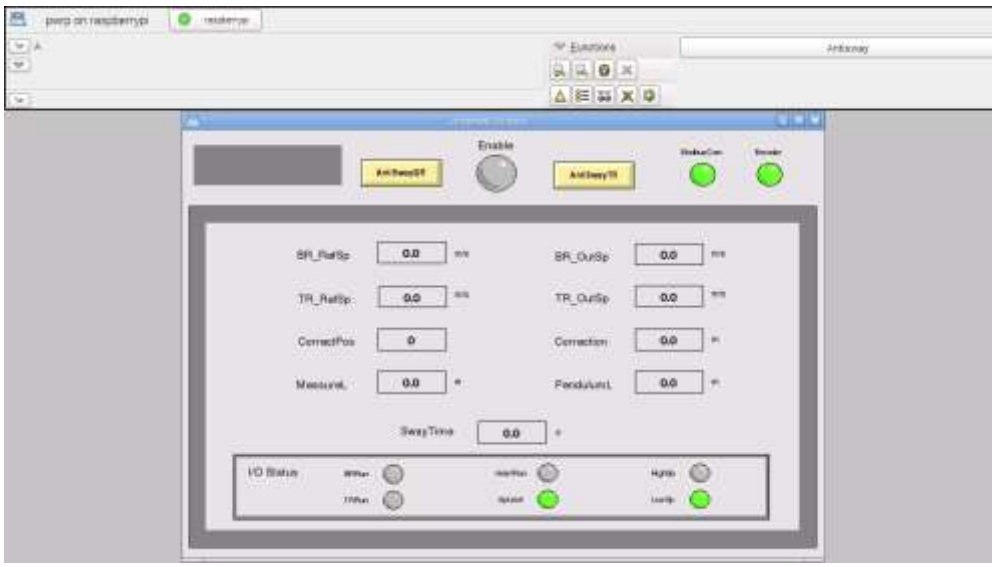




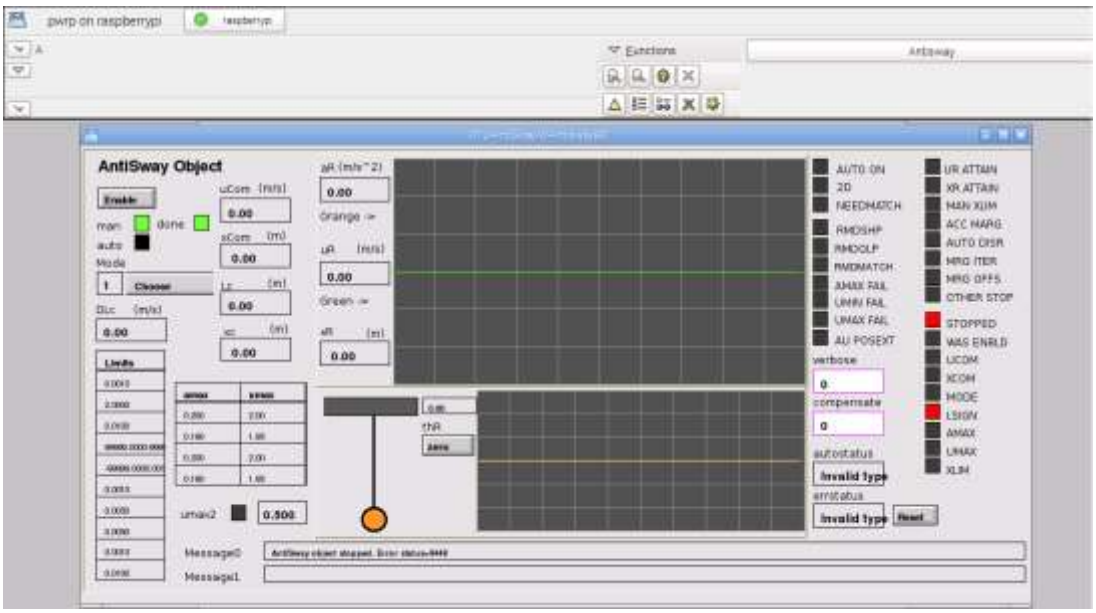
双击 HMI 图标即可打开 Proview 操作窗口，输入用户名和密码登录后即可对 HMI 进行操作。



点击操作窗口右上方的 Antisway 按钮即可打开防摇摆系统的 HMI 画面。

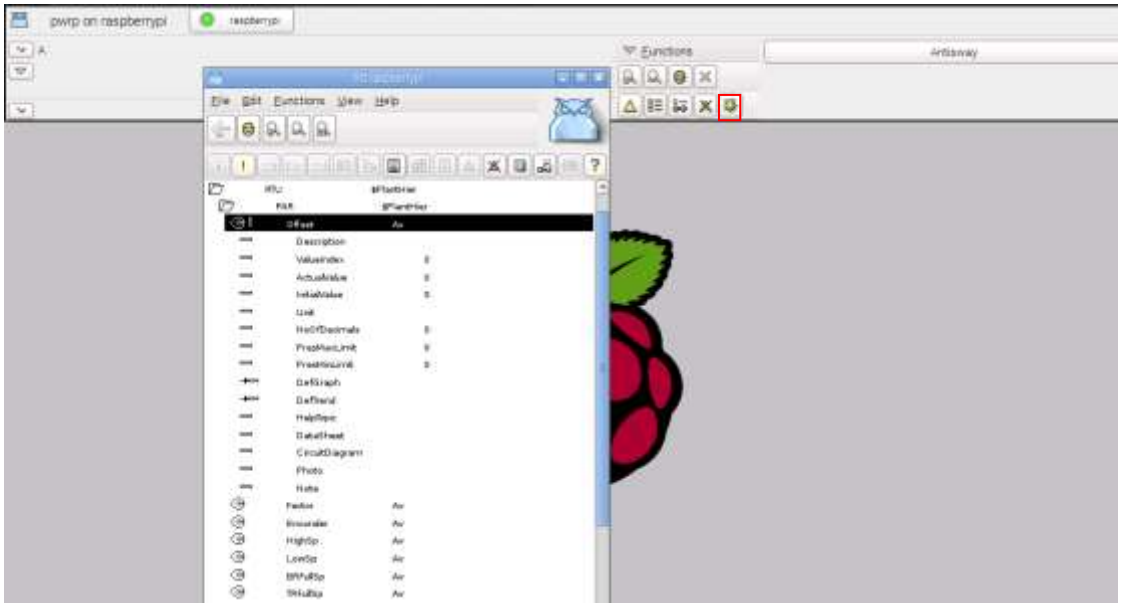


防摇摆 HMI 主窗口包含了行车防摇摆的主要信息：大小车、主钩是否运行，行车选择高速或低速运行，大小车速度输入设定，大小车防摇摆速度输出，钟摆长度，编码器测量长度，吊物补偿长度，摆动周期，防摇摆控制器是否激活信息。防摇摆控制模型信息可点击 AntiSwayBR 按钮查看大车防摇摆状态，AntiSwayTR 按钮查看小车防摇摆状态。



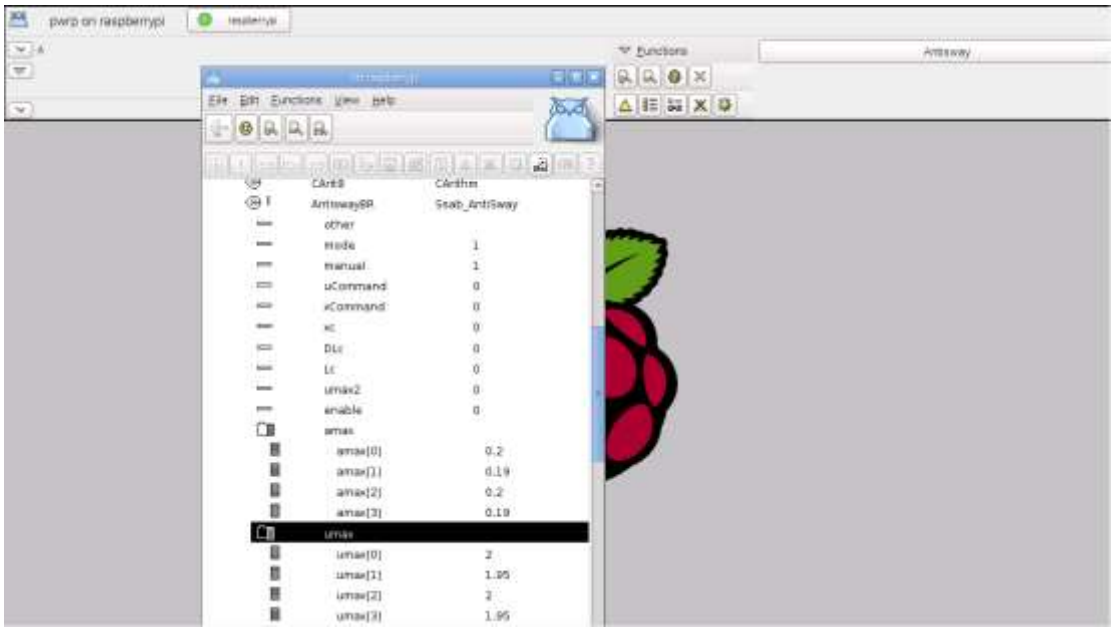
2.4.2 Proview 导航器 Xtt 的使用

点击操作窗口工具栏的图标打开 Xtt 导航器，可实时对 RTU-PAR 里面的变量进行设定，调试行车的防摇摆系统。



防摇摆模型的调整主要为根据行车的实际情况调节模型的速度输入上限及加速度参数，模型给定速度输入上限只要高于行车实际给定输入即可，最好略高约实际最高速度给定，对应参数为 umax[0]和 umax[1],路径为 RTU-AntiSway-W-AntiswayBR-umax-umax[0]/umax[1]。

对防摇摆效果影响最主要的参数为加速度，双脉冲防摇摆原理即通过两个短脉冲的加速度相互抵消造成的摇摆现象，如果给定速度一定，加速度过高或过低，都会造成防摇摆效果不理想，因此调整合适的加速度对防摇摆控制模型尤为重要。模型加速度参数为 amax[0]和 amax[1],路径为 RTU-AntiSway-W-AntiswayBR-umax-amax[0]/amax[1]。小车的调整与大车调整方法一样，调整 AntiswayTR 中的 umax[0]/[1]和 amax[0]/[1]。



行车防摇摆控制系统调试至满意效果后，根据实时系统调整的参数，在开发站中修改调整的参数与实时系统一致，重新编译并发布至实时系统，在实时系统重新安装更新的运行包，重新启动软件或重新启动树莓派后，新的运行包生效，防摇摆控制系统运行在调整后的效果。如果不更新运行包，防摇摆控制系统一旦重启，修改调整的参数将恢复成原始参数，实时系统的参数调整只是用于临时调整，需要永久调整必须到开发站修改并生成新的运行包，实时系统安装重启后生效。

三、项目运行

四、应用体会

五、参考文献