HPM5300

HPM5300 系列MCU RDC硬解码测试文档

适用于上海先楫半导体HPM5300系列高性能微控制器。

目录

版本信息

表1. 版本信息

日期	描述
2025-5-8	初版

1. 例程简介

HPM5300_RDC例程基于HPM5300实现了用旋变测试电机theta角度的功能，并可通过SEI接口与绝对值编码器通信。HPM5300_RDC硬件电路板具有的功能有：两路ADC、一路UART、一路SPI、两路DAC、QEO输出、LED显示、JLINK连接、SEI接口、USB TYPE-C连接等。

图1 HPM5300_RDC电路板实物图

RDC相关的接插件J3和J4定义表见表1。

表1 RDC板接线

功能	旋变板位置	说明
Exc+	J4[5]	旋变信号
Exc-	J4[6]	
Cos+	J4[1]	
Cos-	J4[2]	
Sin+	J4[3]	
Sin-	J4[4]	
DAT+	J3[8]	绝对值编码器信号
DAT-	J3[7]	
power	J3[5]	24V,150mA
GND	J3[4]	
RX	J3[1]	串口10M 输出角度
TX	J3[6]	

测试基于汇川SV670P伺服驱动器和汇川MS1H1-75B30CB-A331R电机，电机带有23位的绝对值编码器，编码器型号EA38H8B23M16RH5N3。测试的工装治具是通过定制加工的方法将旋变和电机固定在一起，测试工装和伺服驱动如图2所示。

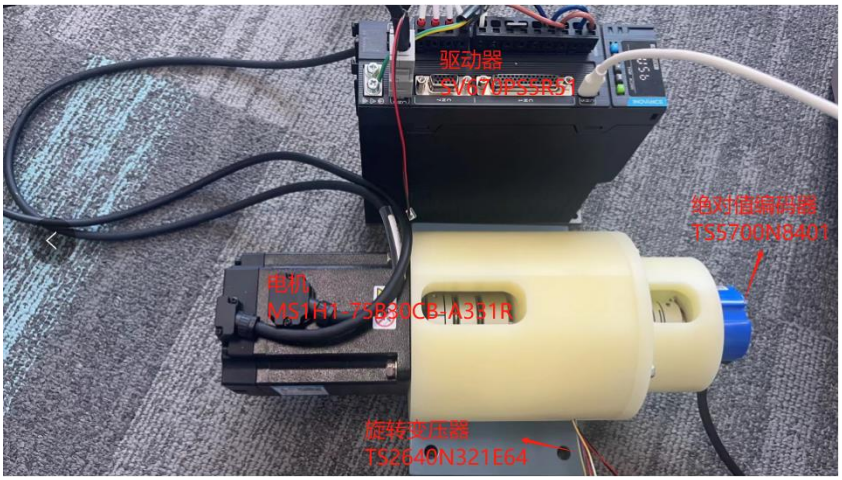

图2 测试工装治具和伺服驱动器连接图

2. 测试方法

本章以HPM5300_RDC电路板、汇川SV670P伺服驱动器和汇川MS1H1-75B30CB-A331R电机为例说明测试方法。

2.1. 测试设备

1、带有串口助手的电脑；

2、示波器、逻辑分析仪；

3、软件安装，需要安装汇川电机控制软件InoDriverShop和OZONE；

4、DAP-LINK调试器和J-LINK调试器；

5、杜邦线若干；

6、HPM5300EVK；

7、数字电源。

3. 硬件测试

3.1. 静态测试

静态测试RDC的信号，包括EXC_P、EXC_N、OSIN和OCOS信号。HPM5300_RDC电路板上有待测信号的测试点。EXC_P、EXC_N为PWM调制后正弦差分信号。

测试方法：24V供电，限流150mA（其他测试项的电压、电流均按此设置）。使用JLINK连接HPM5300_RDC电路板，运行RDC方案的工程示波器观测EXC_P EXC_N信号。图3为RDC静态下EXC_P和EXC_N信号，频率10KHz，幅值2.6V左右。

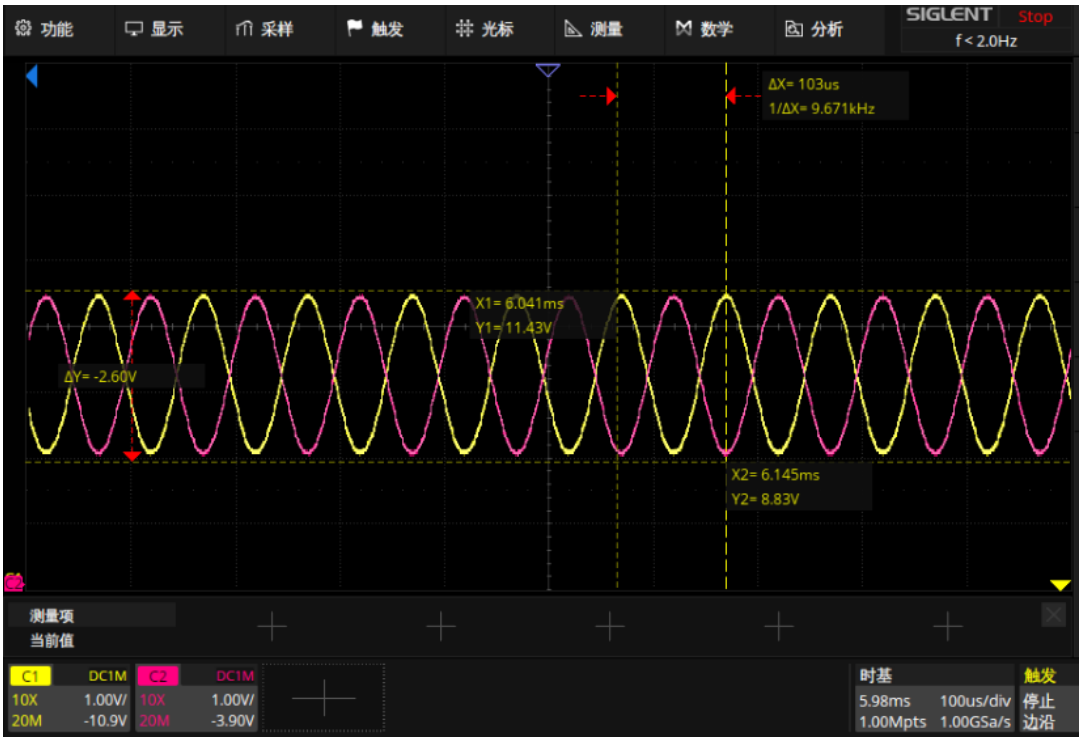

图3 静态下EXC_P（黄） EXC_N（红）信号

OSIN、OCOS信号为旋变正弦、余弦绕组信号。静态下OSIN和OCOS信号和编码器当前的状态有关，应该同为10KHz的正弦（余弦）信号。图4为静态时OSIN和OCOS信号。

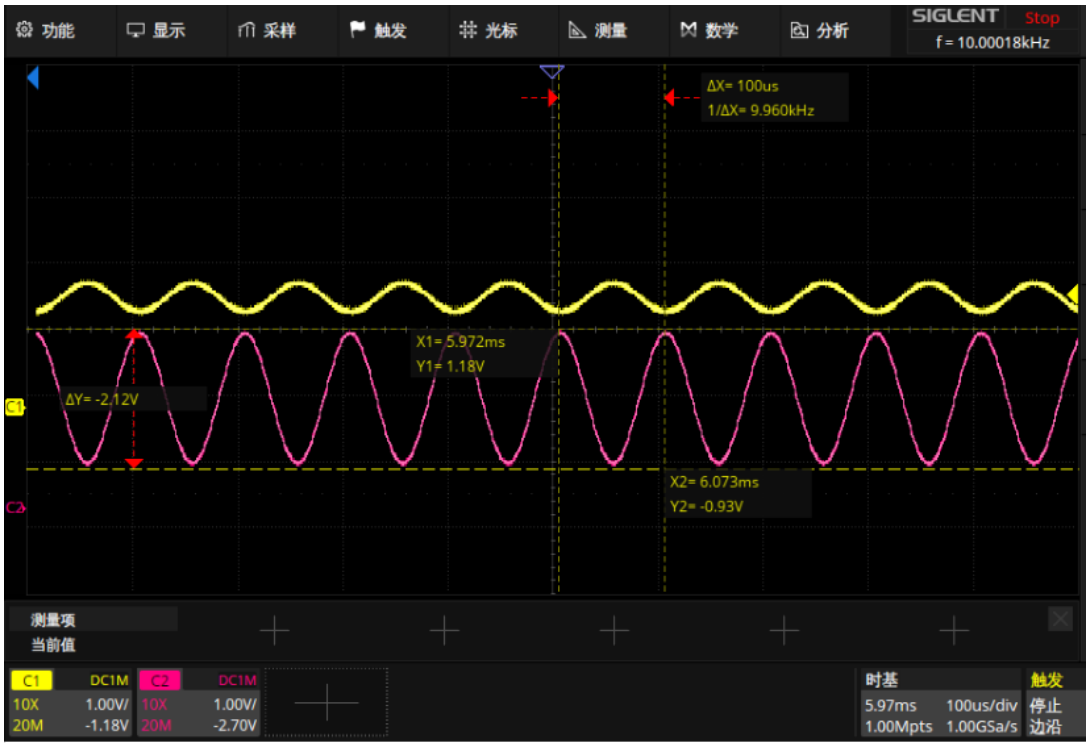

图4 静态下OSIN（黄） OCOS（红）信号

EXC+和 EXC-是EXC_P和EXC_N经过放大后的信号，频率10KHz，幅值9.6V左右。

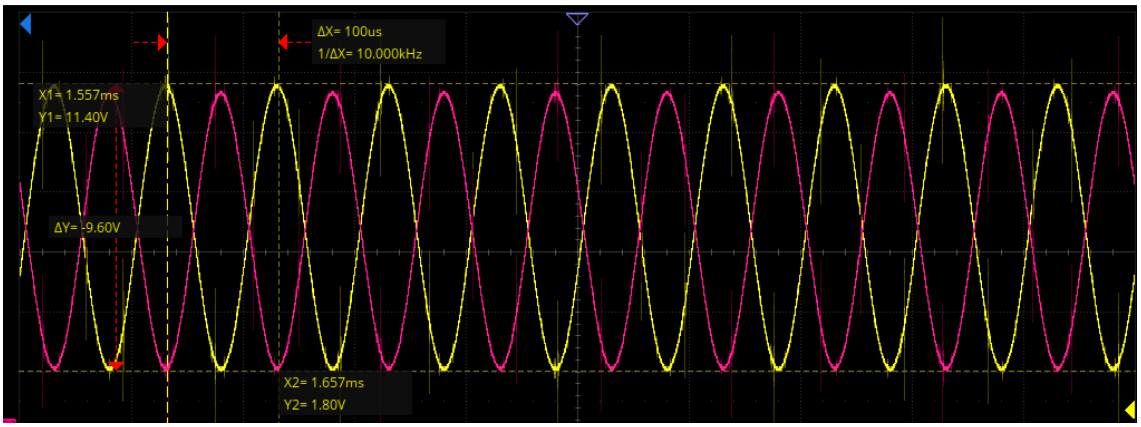

图5 静态下EXC+ EXC-信号

3.2. 动态测试

动态性能测试。电机旋转时OSIN、OCOS信号，与静态测试不同的是需要电机旋转时测试。

测试使用的是汇川SV670P伺服驱动器和汇川MS1H1-75B30CB-A331R电机。需要安装上位机软件并将例程中提供的工程压缩包解压后放到Inovance\InoDriverShop\Servo目录下。例程中提供了上位机安装包和测试工程文件压缩包。

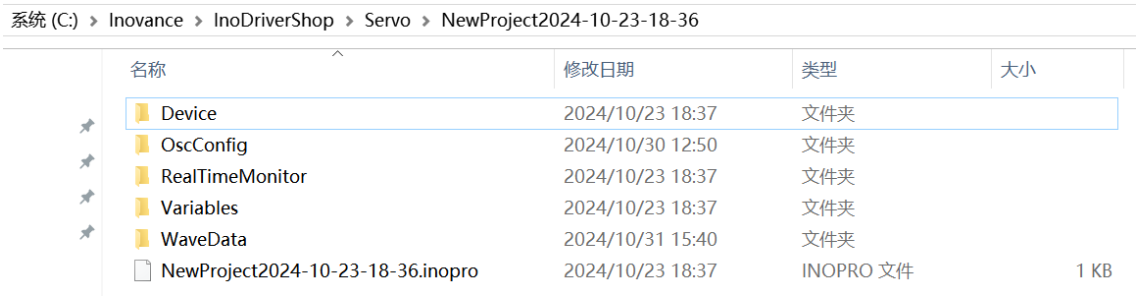

图6 测试工程文件夹存放目录

IDS软件的使用方法如下图，首先导入测试工程。点击左上角的打开工程按钮，选择测试工程文件夹并选择工程文件后点击打开。

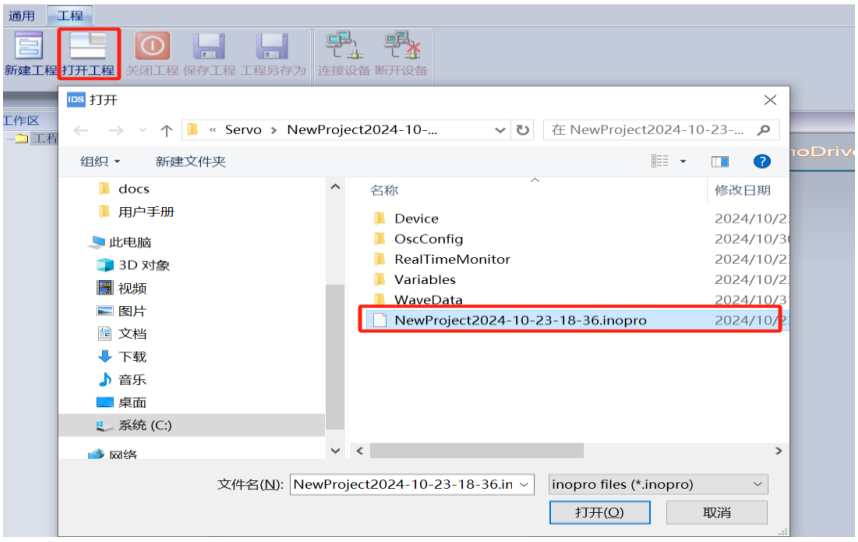

图7 导入测试工程

其中转速、加速时间和减速时间用户可以自行设定，最大转速3000rpm。运行次数和间隔时间也可自行设定，示意图仅供参考。点击开启使能后再点击正极限位置设定，此时多次点击正向按钮，正极限位置的值会变化。为了使电机单次转动的时间较长，建议多点击一段时间的正向按钮。最后点击运行电机即可转动。IDS软件配置及操作方法如图8所示。

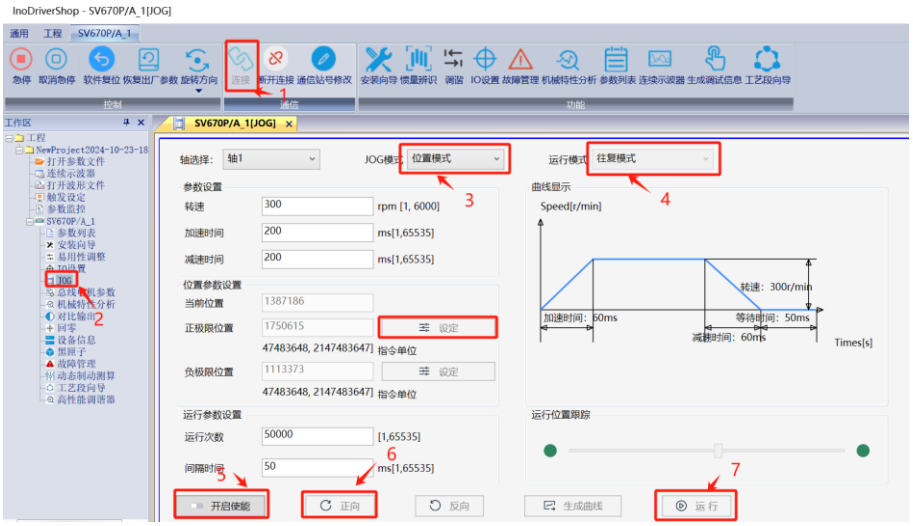

图8 IDS软件配置及操作方法

图9为300rpm时OSIN和 OCOS曲线，频率10KHz。

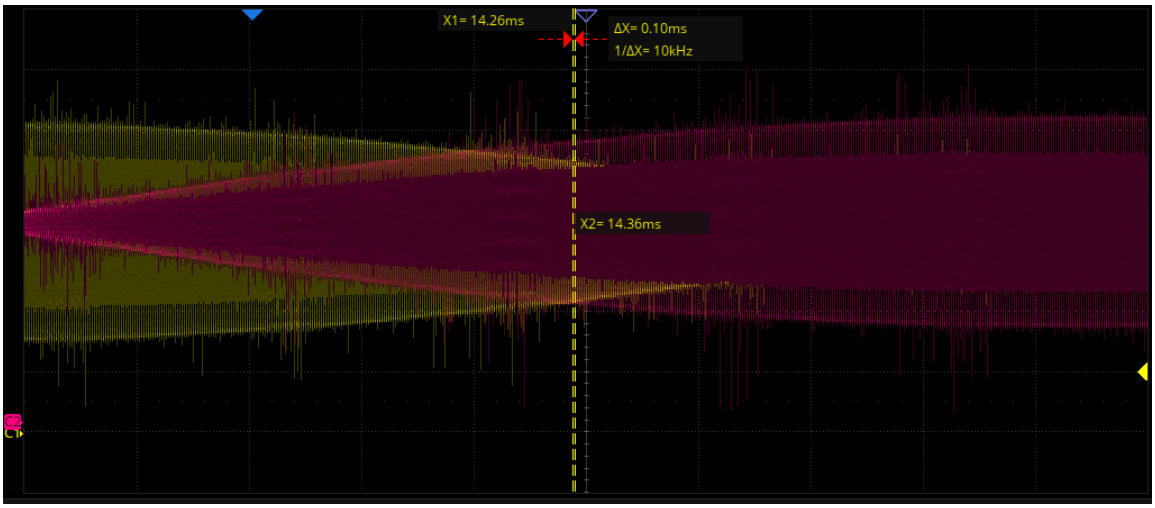

图9 动态（300rpm）时OSIN（黄） OCOS（红）曲线

图10为300rpm时OSIN和 OCOS的细节，频率10KHz。

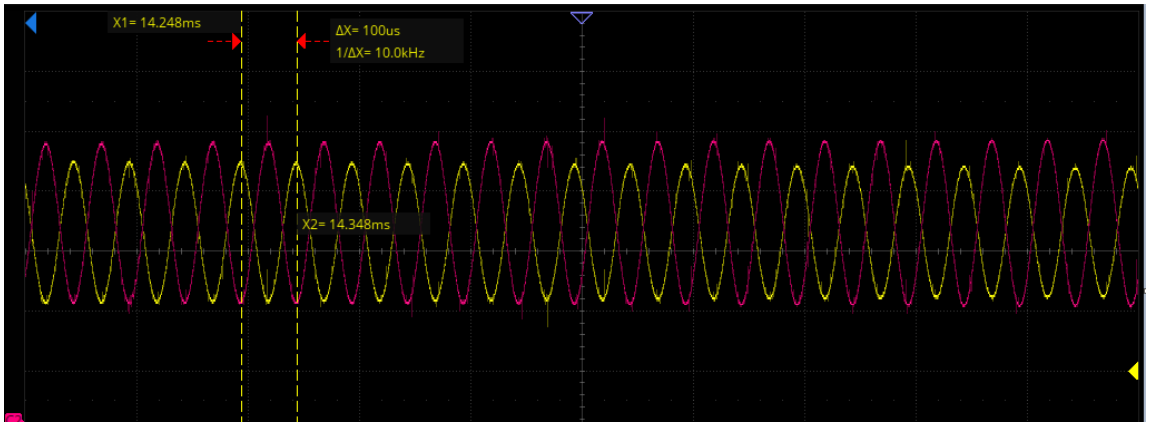

图10 动态（300rpm）时OSIN（黄） OCOS（红）细节曲线

4. 数据输出

4.1. 串口

4.1.1. 测试步骤

（1）准备一块HPM5300EVK使用UART与HPM5300_RDC板通信接线，用杜邦线将两块板子的UART连接起来，注意共地。

表2 UART测试接线

HPM5300_RDC电路板	HPM5300EVK
J3[1](UART_RX)	P1[8]/PB08(UART_TX)
J3[6](UART_TX)	P1[10]/PB09(UART_RX)

（2）将HPM5300_RDC例程中的UART_DEBUG_CONTROL宏定义置1。

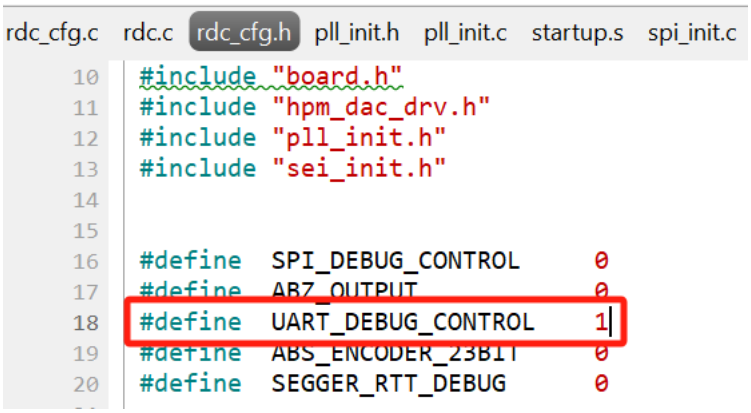

图11 UART_DEBUG_CONTROL宏定义置1

（3）运行HPM5300_RDC程序。

（4）设置电机转速为1200rpm并运行电机；

（5）测试工程设置成UART测试模式，将要导出的变量添加到watch，在电机匀速运行时运行测试程序，并在图12所示处打断点。

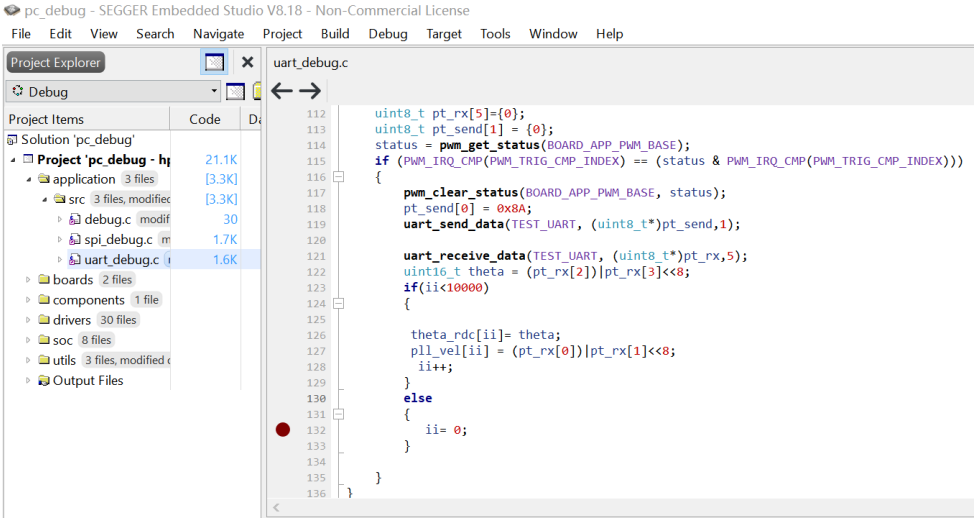

图12 UART测试程序断点位置

（6）导出数据并处理。用户可以用EXCEL处理数据，theta和velocity均除以100。

4.1.2. 测试结果

图13为1200rpm时的角度和速度，速度的单位为r/s，图中的速度为20左右，与理论计算值（1200/60）相符合。

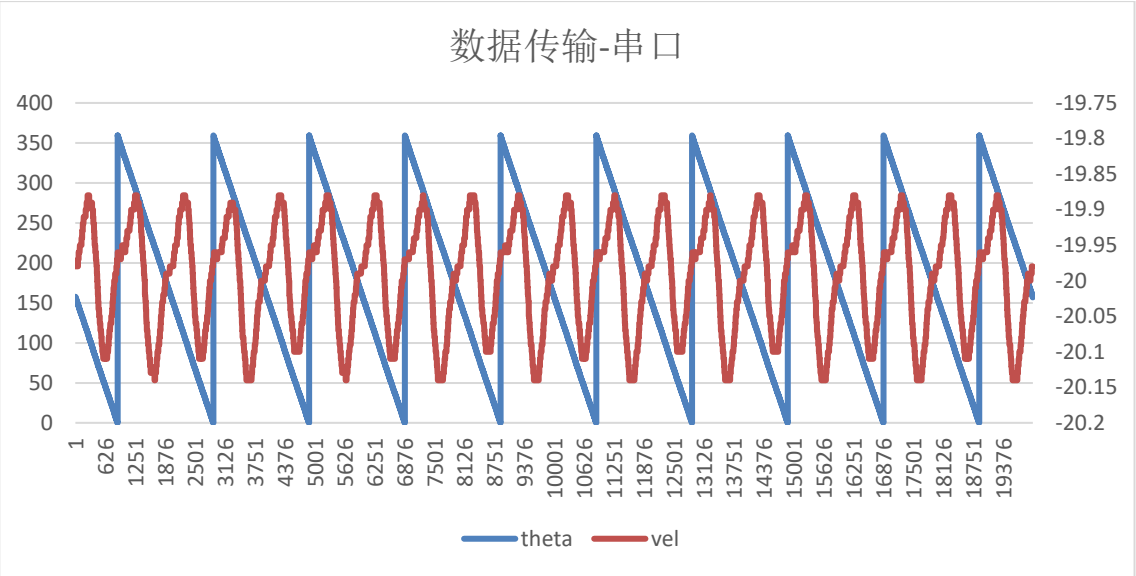

图13 1200rpm时的角度和速度（UART）

4.2. SPI

4.2.1. 测试步骤

（1）准备一块HPM5300EVK使用UART与HPM5300_RDC板通信接线，用杜邦线将两块板子的UART连接起来，注意共地。

表3 SPI测试接线

HPM5300_RDC电路板	HPM5300EVK
J3[1](SPI_MISO)	P1[21]/PPA28(SPI_MOSI)
J3[6](SPI_MOSI)	P1[19]/PA29(SPI_MISO)
J3[2](SPI_CLK)	P1[23]/PA27(SPI_CLK)
J3[3](SPI_CS)	P1[21]/PA28(SPI_CS)

（2）将HPM5300_RDC例程中的SPI_DEBUG_CONTROL宏定义置1。

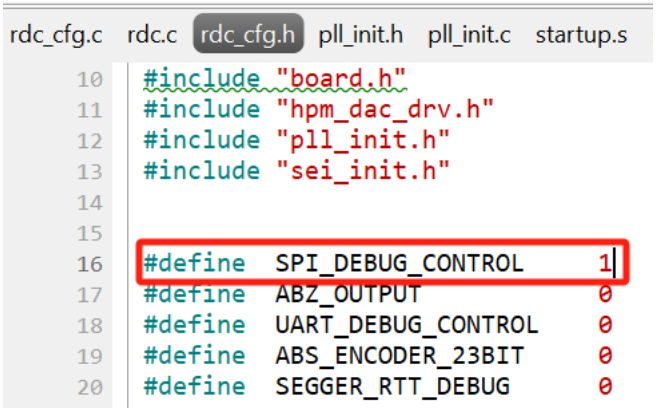

图14 SPI_DEBUG_CONTROL宏定义置1

（3）运行HPM5300_RDC程序。

（4）设置电机转速并运行电机；

（5）测试工程设置成SPI测试模式，将要导出的变量添加到watch，在电机匀速运行时运行测试程序。打断点和导出数据的方法与UART测试相同；

（6）导出数据并处理。theta和velocity均除以100。

4.2.2. 测试结果

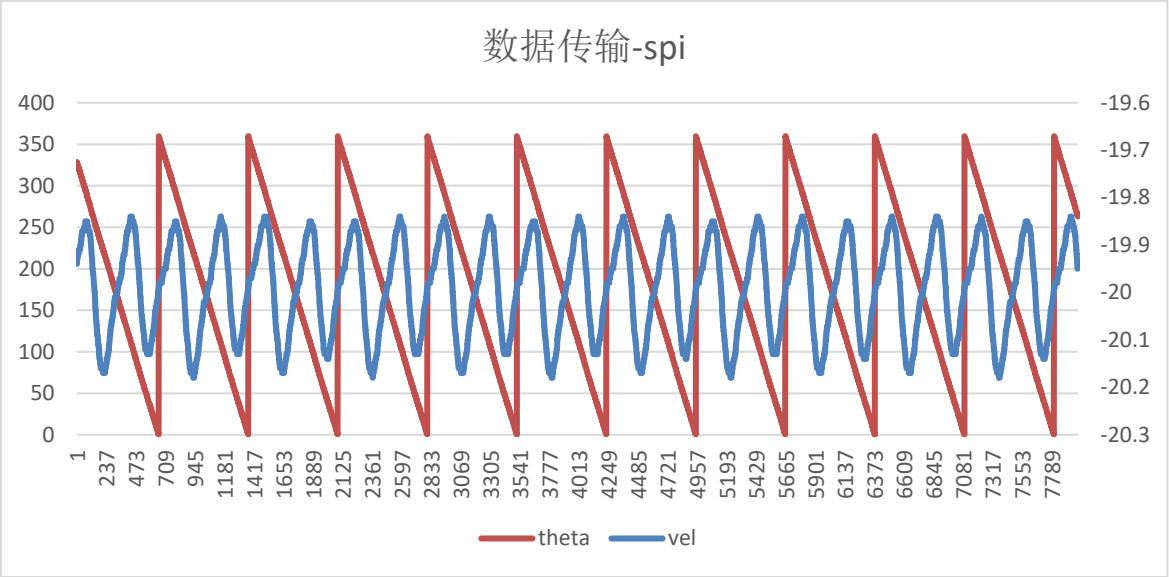

图15 1200rpm时的角度和速度（SPI）

4.3. QEO

4.3.1. 测试步骤

（1）准备逻辑分析仪观测QEO输出信号，并接线，分别将QEO的A/B/Z信号接到逻辑分析仪的channel0~2，注意共地。

（2）HPM5300_RDC程序中ABZ_OUTPUT置1。

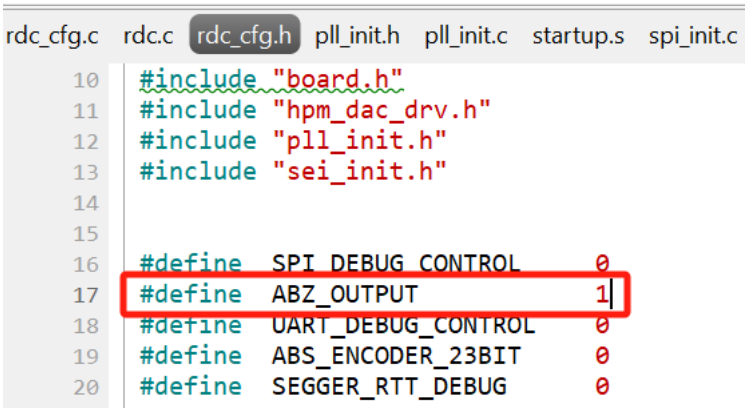

图16 ABZ_OUTPUT宏定义置1

（3）配置IDS软件，运行模式设置成距离模式，转速可以在[1,3000]范围内设置。运行距离设置为10000或-10000，对应正转一圈和反转一圈。配置方法如图17和

18所示。

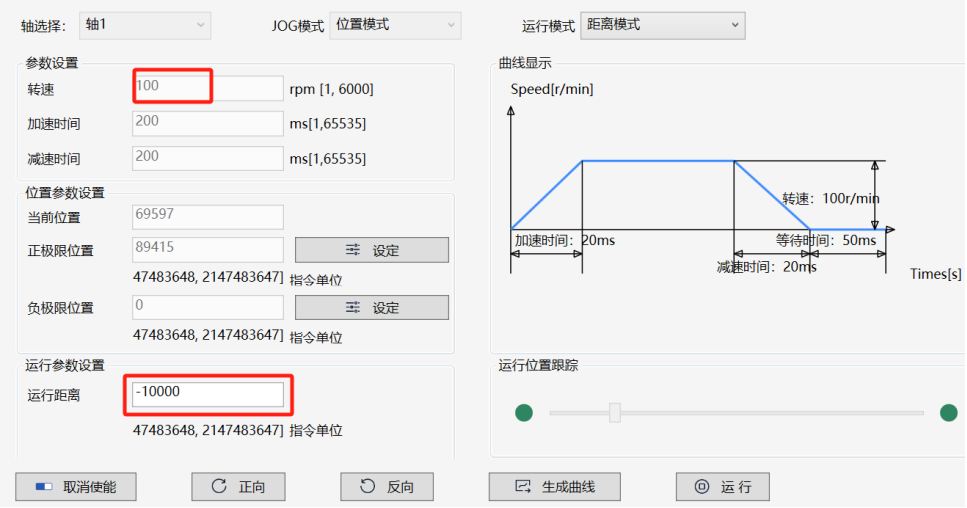

图17 电机反转一圈设置方法

图18 电机正转一圈设置方法

（4）下载HPM5300_RDC程序并运行，将逻辑分析仪的channel0(A相信号)设置为上升沿检测模式。

（5）运行电机，此时逻辑分析仪会抓取到QEO信号。

4.3.2. 测试结果

（1）电机正转一圈，A相超前于B相90度，1024线脉冲输出

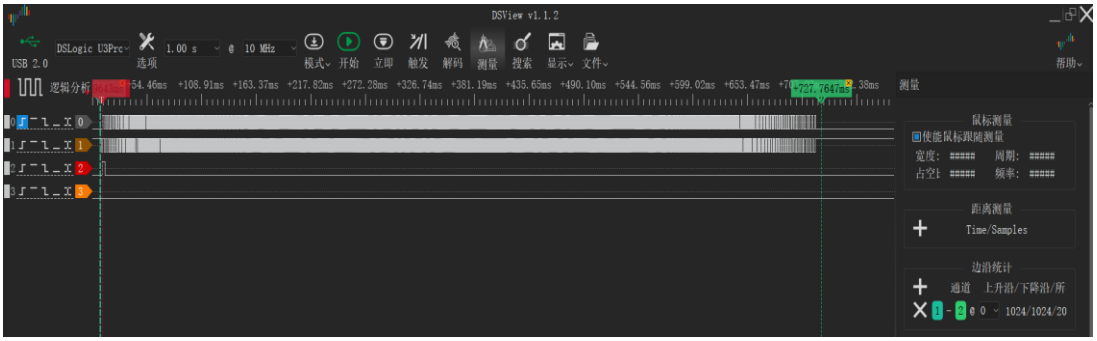

图18 正转一圈【整体】

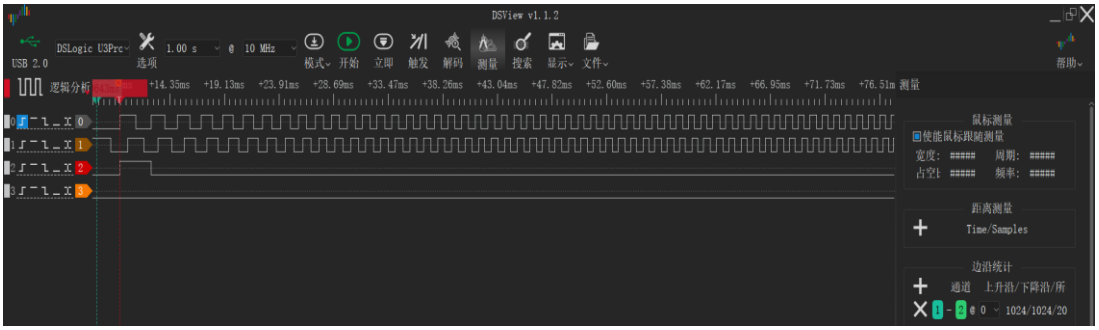

图19电机加速段【细节】

（2）电机反转一圈，B相超前A相90度，1024线脉冲输出

图20 反转一圈【整体】

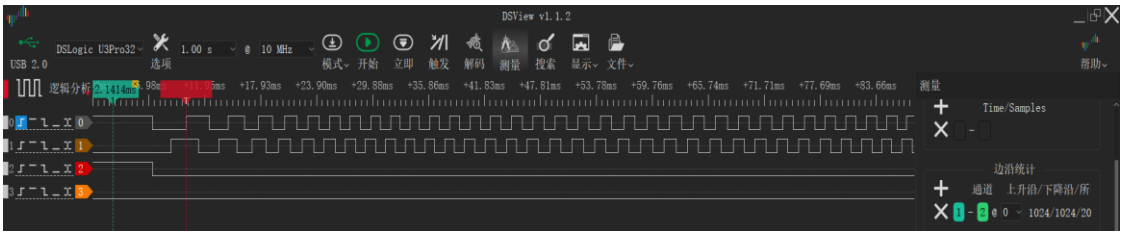

图21 电机加速段【细节】

4.4. Bissc从模式

SEI模块配置为从机模式，作为编码器，将旋变解码位置信息发送出去，有两种方式：

- 硬件tirg pos

  rdc->qei->sei

  0对应0°，0x100000000对应360°（电角度）

- 软件写入 pos

4.4.1. 测试步骤

（1）HPM5300RDC板数据接口做bissc通信协议输出时，需要将U10的5脚6脚剪掉。

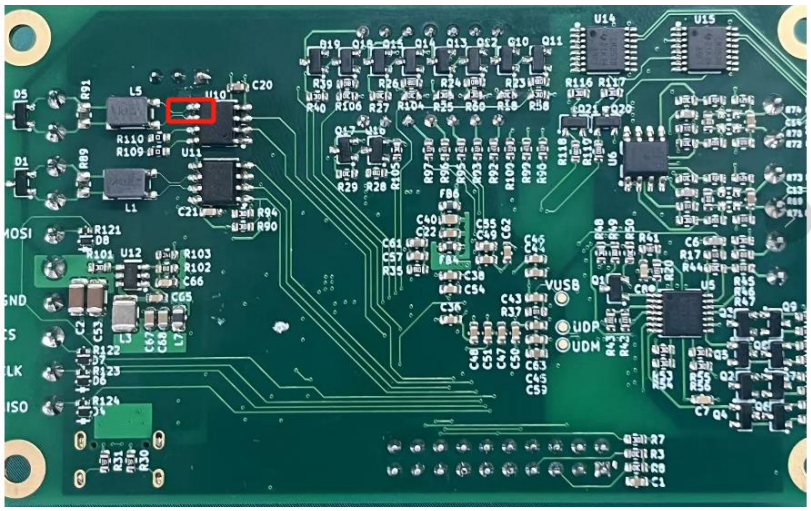

图22 引脚修改示意

（2）Master代码路径：samples/drivers/sei/master/bissc，并做如下修改：

```
/* data register 2: recv Multi Turn Value */
data_format_config.mode = sei_data_mode;
data_format_config.signed_flag = false;
data_format_config.bit_order = sei_bit_msb_first;
data_format_config.word_order = sei_word_nonreverse;
data_format_config.word_len = 32;//12;
data_format_config.last_bit = 0;
data_format_config.first_bit = 31;//11;
data_format_config.max_bit = 31;//11;
data_format_config.min_bit = 0;
sei_cmd_data_format_config_init(BOARD_SEI, SEI_SELECT_DATA, SEI_DAT_2, &data_format_config);
/* data register 3: recv Single Turn Value */
data_format_config.mode = sei_data_mode;
data_format_config.signed_flag = false;
data_format_config.bit_order = sei_bit_msb_first;
data_format_config.word_order = sei_word_nonreverse;
data_format_config.word_len = 32;//12;
data_format_config.last_bit = 0;//20;
data_format_config.first_bit = 31;
data_format_config.max_bit = 31;
data_format_config.min_bit = 0;//20;
sei_cmd_data_format_config_init(BOARD_SEI, SEI_SELECT_DATA, SEI_DAT_3, &data_format_config);
/* data register 4: error and warn Value */
data_format_config.mode = sei_data_mode;
data_format_config.signed_flag = false;
data_format_config.bit_order = sei_bit_msb_first;
data_format_config.word_order = sei_word_nonreverse;
data_format_config.word_len = 2;
data_format_config.last_bit = 0;
data_format_config.first_bit = 1;
data_format_config.max_bit = 1;
data_format_config.min_bit = 0;
sei_cmd_data_format_config_init(BOARD_SEI, SEI_SELECT_DATA, SEI_DAT_4, &data_format_config);
/* data register 5: check crc */
data_format_config.mode = sei_crc_mode;
data_format_config.signed_flag = false;
data_format_config.bit_order = sei_bit_msb_first;
data_format_config.word_order = sei_word_nonreverse;
data_format_config.word_len = 6;
data_format_config.crc_invert = true;
data_format_config.crc_shift_mode = false;
data_format_config.crc_len = 6;
data_format_config.last_bit = 0;
data_format_config.first_bit = 5;
data_format_config.max_bit = 5;
data_format_config.min_bit = 0;
data_format_config.crc_init_value = 0;
data_format_config.crc_poly = 3;
sei_cmd_data_format_config_init(BOARD_SEI, SEI_SELECT_DATA, SEI_DAT_5, &data_format_config);

/* [3] sei instructions */
instr_idx = 0;
sei_set_instr(BOARD_SEI, instr_idx++, SEI_INSTR_OP_WAIT, SEI_INSTR_M_CK_FALL_RISE, SEI_DAT_0, SEI_DATA_CONST_0, 1);  /* ACK:0 */
sei_set_instr(BOARD_SEI, instr_idx++, SEI_INSTR_OP_WAIT, SEI_INSTR_M_CK_FALL_RISE, SEI_DAT_0, SEI_DATA_CONST_1, 1);  /* START:1 */
sei_set_instr(BOARD_SEI, instr_idx++, SEI_INSTR_OP_RECV, SEI_INSTR_M_CK_FALL_RISE, SEI_DAT_0, SEI_DAT_0, 1);   /* CDS */
//sei_set_instr(BOARD_SEI, instr_idx++, SEI_INSTR_OP_RECV, SEI_INSTR_M_CK_FALL_RISE, SEI_DAT_5, SEI_DAT_2, 12);  /* MT */
//sei_set_instr(BOARD_SEI, instr_idx++, SEI_INSTR_OP_RECV, SEI_INSTR_M_CK_FALL_RISE, SEI_DAT_5, SEI_DAT_3, 12);  /* ST */
sei_set_instr(BOARD_SEI, instr_idx++, SEI_INSTR_OP_RECV, SEI_INSTR_M_CK_FALL_RISE, SEI_DAT_5, SEI_DAT_2, 32);  /* MT */
sei_set_instr(BOARD_SEI, instr_idx++, SEI_INSTR_OP_RECV, SEI_INSTR_M_CK_FALL_RISE, SEI_DAT_5, SEI_DAT_3, 32);  /* ST */
sei_set_instr(BOARD_SEI, instr_idx++, SEI_INSTR_OP_RECV, SEI_INSTR_M_CK_FALL_RISE, SEI_DAT_5, SEI_DAT_4, 2);   /* Error warn*/
sei_set_instr(BOARD_SEI, instr_idx++, SEI_INSTR_OP_RECV, SEI_INSTR_M_CK_FALL_RISE, SEI_DAT_0, SEI_DAT_5, 6);   /* CRC */
sei_set_instr(BOARD_SEI, instr_idx++, SEI_INSTR_OP_RECV, SEI_INSTR_M_CK_HIGH, SEI_DAT_0, SEI_DATA_CONST_0, 0);       /* timeout */
sei_set_instr(BOARD_SEI, instr_idx++, SEI_INSTR_OP_WAIT, SEI_INSTR_M_CK_HIGH, SEI_DAT_0, SEI_DATA_CONST_1, 0);
```

图23 例程配置

（3）硬件触发

a. 设置 rdc_cfg.h 中相关宏定义

```
#define  BISSC_SLAVE                        1
#define  BISSC_SLAVE_POS_HARDWARE_INJECT   1
```

b. 准备一个 HPM5300evk 板作为主机。

将Master的SEI_CLK跨针跨至Master侧

将Master的SEI接口信号DATA_P/DATA_N与Slave的SEI接口信号DATA_P/DATA_N相连接。

将Master的SEI接口信号CLKO_P/CLKO_N与Slave的SEI接口信号CLKI_P/CLKI_N相连接。

将Master的GND与Slave的GND相连接。

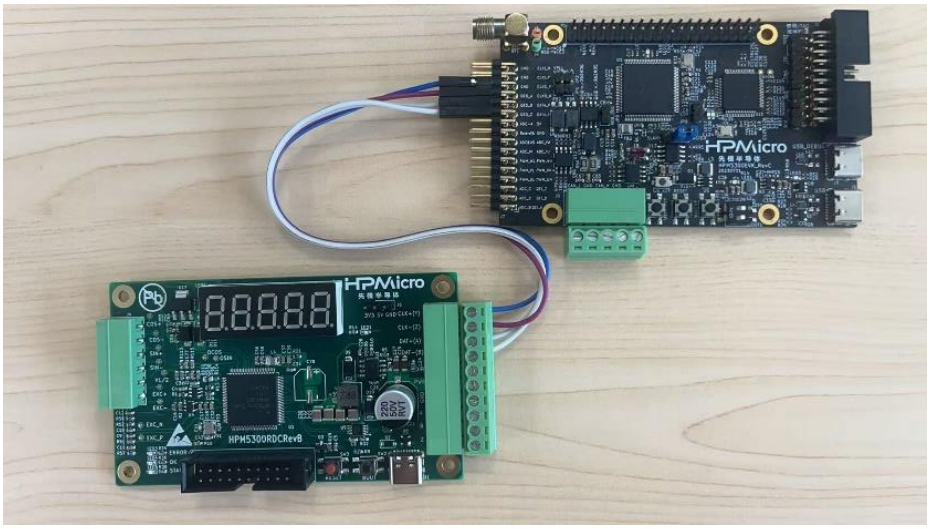

图23 接线示意

c. 将master程序下载至HPM5300EVK板，将rdc程序下载至旋变板

d. 通过串口终端查看各板输出的log信息

（4）软件写入

a. 设置相关宏定义

```
#define  BISSC_SLAVE                        1
#define  BISSC_SLAVE_POS_HARDWARE_INJECT   0
```

b. c. d步骤同（硬件注入）

4.4.2. 测试结果

（1）硬件触发

UVW_POS_CFG_UVW_POS4_CFG	0x00000000	0xf0304
UVW_POS_CFG_UVW_POS5_CFG	0x00000000	0xf0304
PHASE_CNT	0x00000000	0xf0304
PHASE_UPDATE	0x00000000	0xf0304
POSITION	0xac5a75fe	0xf0304
POSITION	0xac5a75fe	[31:0]
POSITION_UPDATE	0x00000000	0xf0304
INC	0 False	[31:31]
DEC	0 False	[30:30]
VALUE	0x00000000	[29:0]
ANGLE	0xac5a7600	0xf0304
POS_TIMEOUT	0x7fffffff	0xf0304

qei1

图24 qei1-postion寄存器

CTRL_1_POS_UPD_CFG	0x00000000	0xf032c
CTRL_1_POS_UPD_DAT	0x00000000	0xf032c
CTRL_1_POS_UPD_TIME	0x00000000	0xf032c
CTRL_1_POS_UPD_POS	0x00000000	0xf032c
CTRL_1_POS_UPD_REV	0x00000000	0xf032c
CTRL_1_POS_UPD_SPD	0x00000000	0xf032c
CTRL_1_POS_UPD_ACC	0x00000000	0xf032c
CTRL_1_POS_SMP_VAL	0x80808080	0xf032c
CTRL_1_POS_SMP_STS	0x00000000	0xf032c
CTRL_1_POS_TIME_IN	0xa85db97a	0xf032c
CTRL_1_POS_POS_IN	0xac5045fe	0xf032c
POS	0xac5045fe	[31:0]
CTRL_1_POS_REV_IN	0x00000000	0xf032c
CTRL_1_POS_SPD_IN	0x00000000	0xf032c
CTRL_1_POS_ACC_IN	0x00000000	0xf032c
CTRL_1_POS_UPD_STS	0x00000000	0xf032c
CTRL_1_IRQ_INT_EN	0x00031000	0xf032c
CTRL_1_IRQ_INT_FLAG	0x00132551	0xf032c
CTRL_1_IRQ_INT_STS	0x00000000	0xf032c

sei

图25 sei_pos寄存器

DAT_2_SET	0x00000000	0xf032f
DAT_2_CLR	0x00000000	0xf032f
DAT_2_INV	0x00000000	0xf032f
DAT_2_IN	0x00000000	0xf032f
DAT_2_OUT	0x00000000	0xf032f
DAT_2_STS	0x001e001f	0xf032f
DAT_3_MODE	0x001f0400	0xf032f
DAT_3_IDX	0x001f1f00	0xf032f
DAT_3_GOLD	0x00000000	0xf032f
DAT_3_CRCINIT	0x00000000	0xf032f
DAT_3_CRCPOLY	0x00000000	0xf032f
DAT_3_DATA	0xac5045fe	0xf032f
DATA	0xac5045fe	[31:0]
DAT_3_SET	0xac5045fe	0xf032f
DAT_3_CLR	0xac5045fe	0xf032f
DAT_3_INV	0xac5045fe	0xf032f
DAT_3_IN	0xac5045fe	0xf032f

sei

图26 sei_DATA寄存器

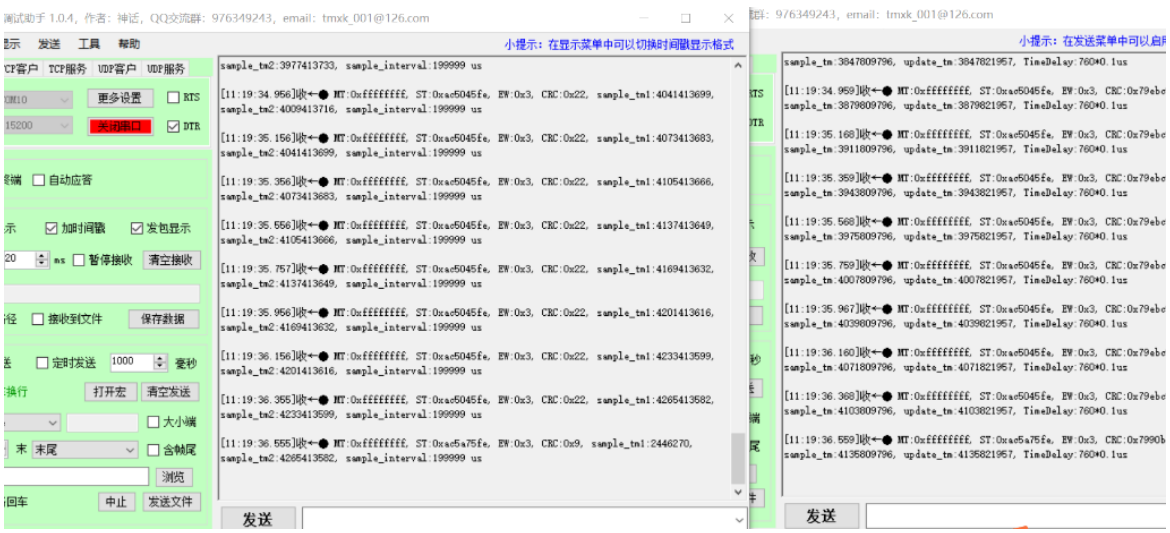

图27 硬件触发模式下slave与master log信息

此时对应的电角度：

tehta = sei_get_data_value(BOARD_SEI,SEI_DAT_3)*360.0/0x100000000 = 242（对应16进制0xf2）

（2）软件注入

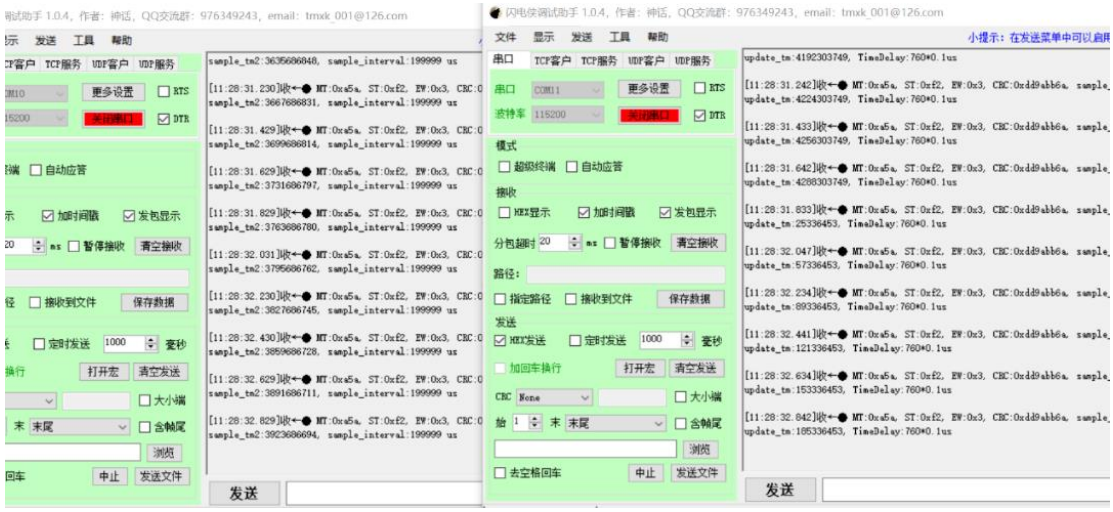

图28 软件注入模式下slave与master log信息

4.5. 多摩川从模式

SEI模块配置为从机模式，作为编码器，将旋变解码的位置信息发送出去，有两种方式：

- 硬件tirg pos

rdc->qei->sei

0对应0°，0x100000000对应360°（电角度）

- 软件写入 pos

4.5.1. 测试步骤

（1）硬件触发

a. 设置rdc_cfg.h中相关宏定义

#define TAMAGAWA_SLAVE_POS_HARDWARE_INJECT 1

#define TAMAGAWA_SLAVE 1

b. 准备一个USB转485模块，将SEI接口信号DATA_P/DATA_N与USB转485的A/B信号相连接

c. 将程序下载至开发板并运行。

d. 通过串口调试助手发送Hex数据：`1A` 或 `02` 或 `8A` 或 `92`，开发板模拟的编码器将会进行响应。同时，可通过串口终端查看开发板输出的log信息

(2) 软件写入

a. 设置相关宏定义

#define TAMAGAWA_SLAVE_POS_HARDWARE_INJECT 0

#define TAMAGAWA_SLAVE 1

b. 准备一个USB转485模块，将SEI接口信号DATA_P/DATA_N与USB转485的A/B信号相连接

c. 将程序下载至开发板并运行。

d. 通过串口调试助手发送Hex数据：`1A` 或 `02` 或 `8A` 或 `92`，开发板模拟的编码器将会进行响应。同时，可通过串口终端查看开发板输出的log信息

4.5.2. 测试结果

（1） 硬件触发

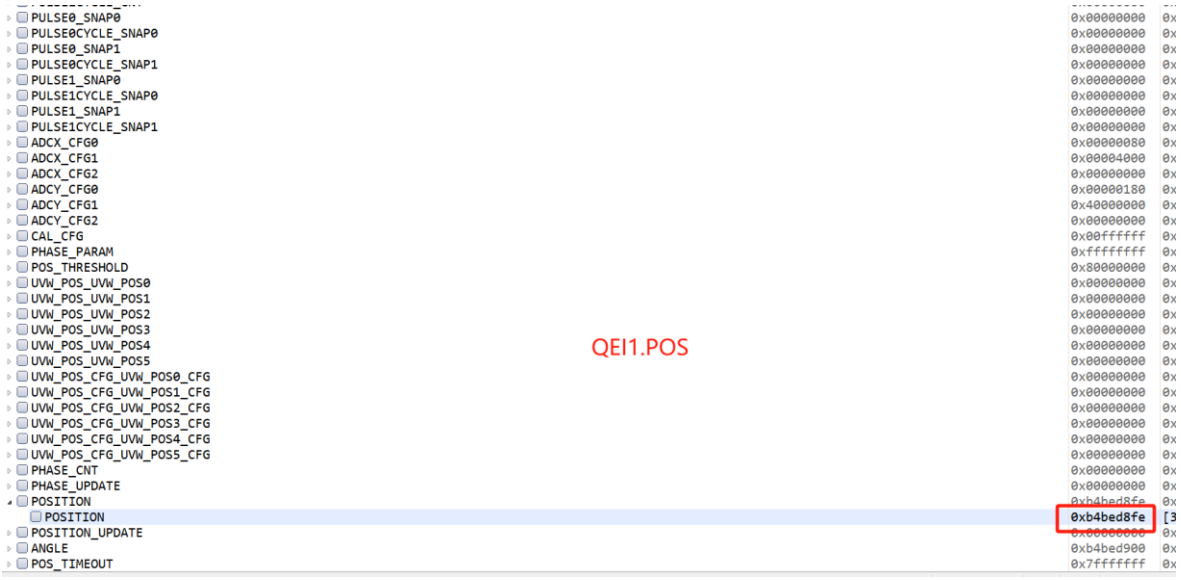

图29 qei1_position寄存器

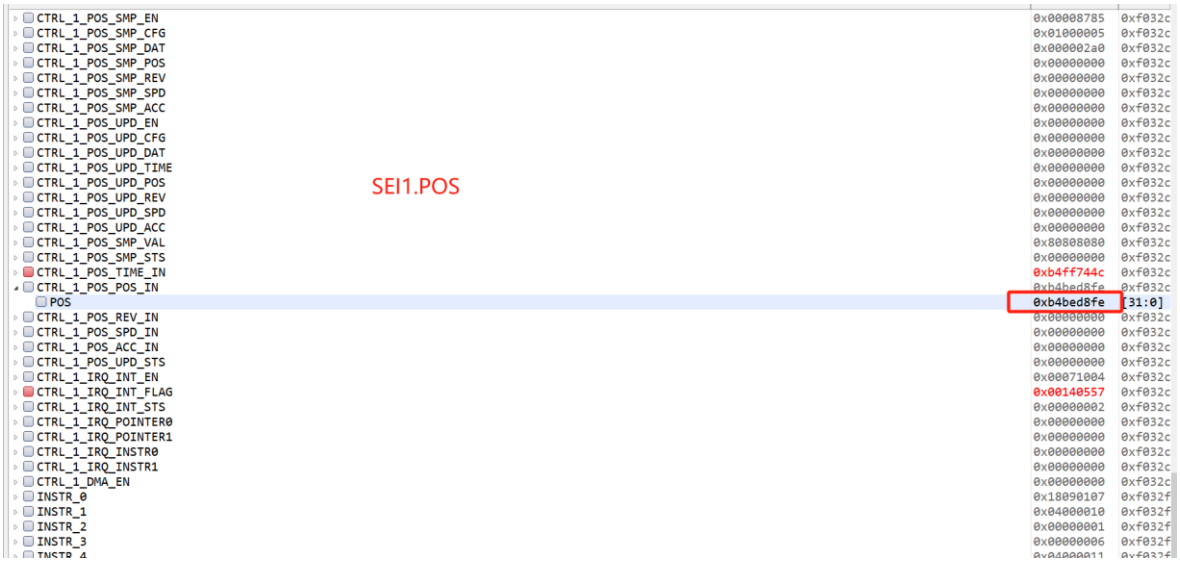

图30 sei_pos寄存器

图31 sei_DATA寄存器

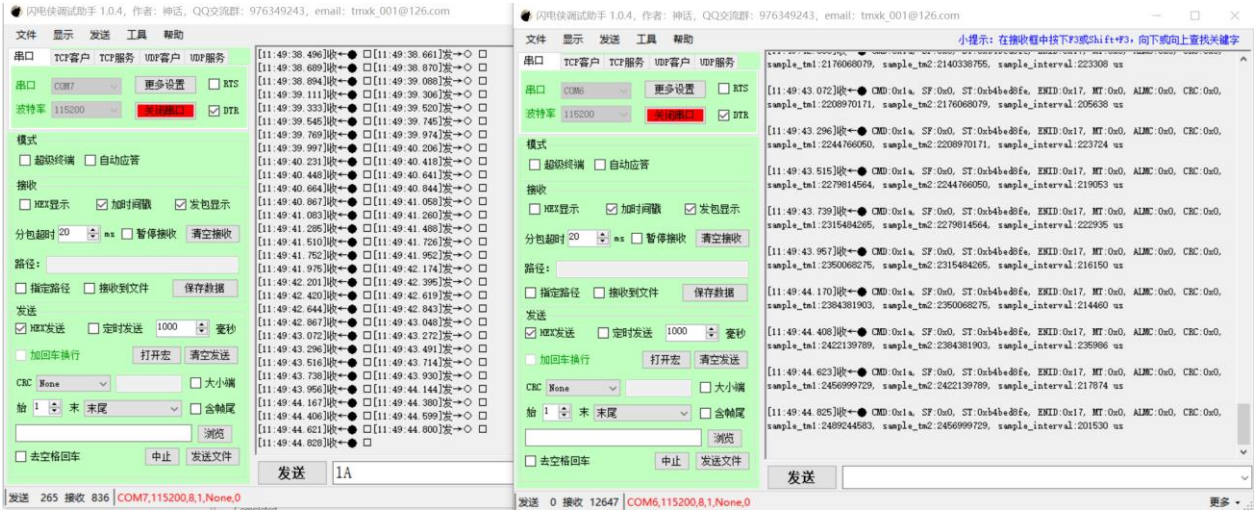

图32 硬件触发模式下slave与master log信息

此时对应的电角度：

theta = sei_get_data_value(BOARD_SEI, SEI_DAT_5)*360.0/0x100000000 =

254（对应十六进制0xfe）

（2）软件注入

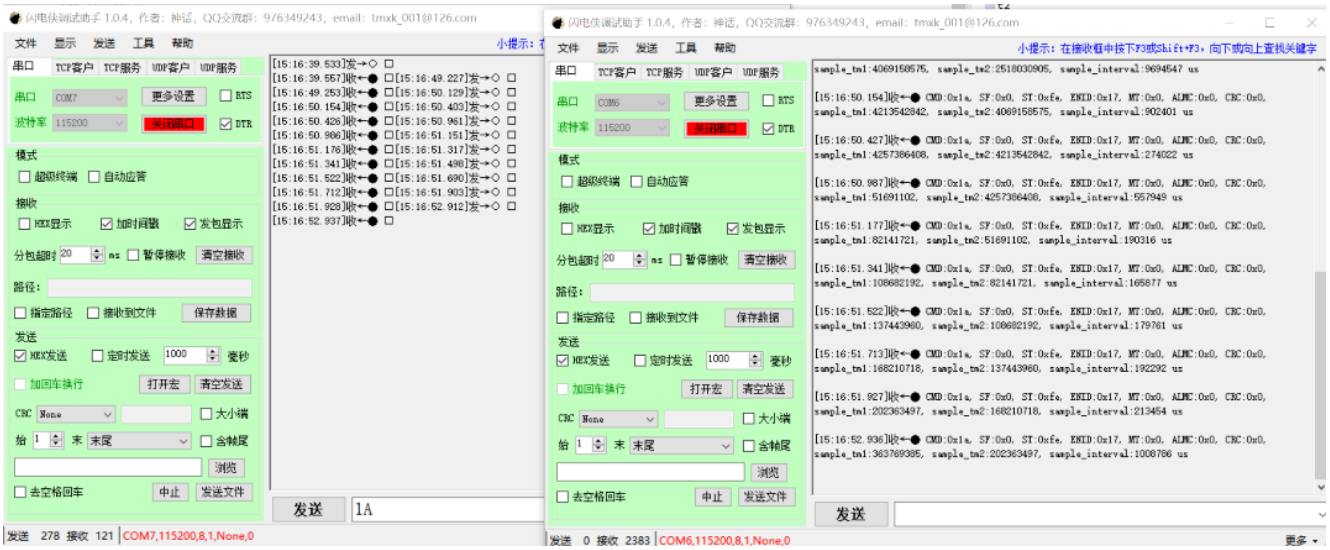

图33 sei从模式下log信息

注：由于测试使用的USB转485模块的最大波特率小于2.5Mbps，所以修改程序中的通讯波特率为115200，实际使用时，修改回2.5Mbps。

5. 误差测试

旋变解码板中角度计算有三种方式：反正切，pll，pll_ii。基于上述电机驱动平台，将旋变解码板的角度与绝对值编码器角度进行对比，获得角度误差。

同样的，测速基于角度也有三种方式。

需要安装OZONE和JLINK驱动。下载地址：https://www.segger.com/。本例程测试使用的JLINK版本为V8.12，OZONE版本为V3.38C。

5.1. 测试步骤

（1）接线。将HPM5300_RDC电路板的SEI与绝对值编码器连接。

表4 绝对值编码器与HPM5300_RDC电路板接线

HPM5300_RDC电路板	绝对值编码器
DATA+_	蓝色线
DATA-	蓝黑色线
5V	红色线

（2）将HPM5300_RDC程序中的ABS_ENCODER_23BIT和SEGGER_RTT_DEBUG宏定义置1。

rdc_cfg.c rdc.c rdc_cfg.h pll_init.h pll_init.c startup.s spi_init.c

```
10  #include "board.h"
11  #include "hpm_dac_drv.h"
12  #include "pll_init.h"
13  #include "sei_init.h"
14
15
16  #define  SPI_DEBUG_CONTROL      0
17  #define  ABZ_OUTPUT             0
18  #define  UART_DEBUG_CONTROL     0
19  #define  ABS_ENCODER_23BIT      1
20  #define  SEGGER_RTT_DEBUG       1
```

图34 ABS_ENCODER_23BIT和SEGGER_RTT_DEBUG宏定义置1

（3）编译HPM5300_RDC程序并用OZONE打开编译后生成的elf文件，并下载程序。

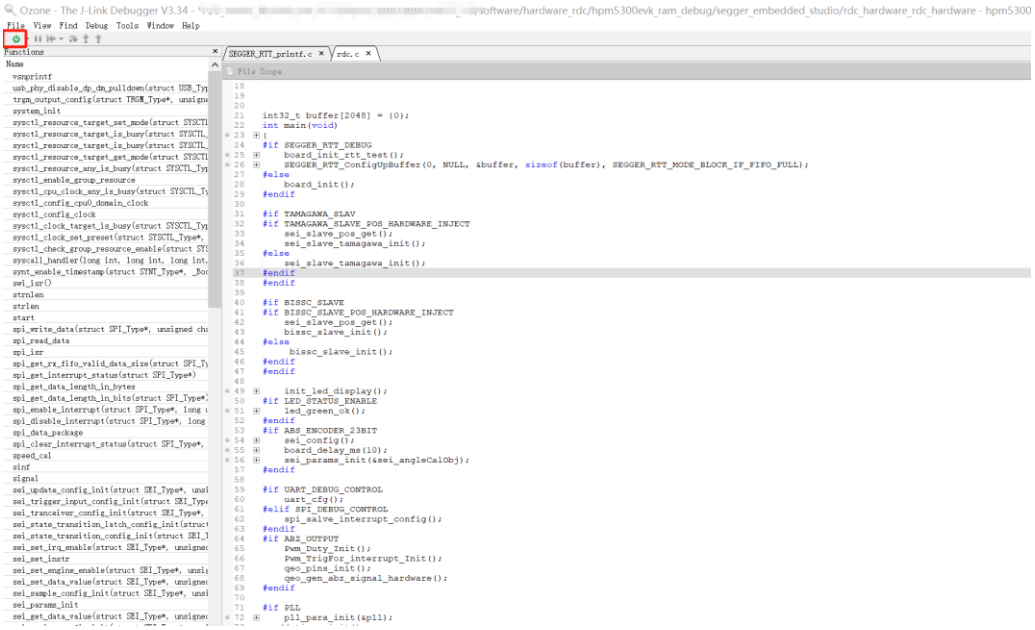

图35 OZONE下载程序

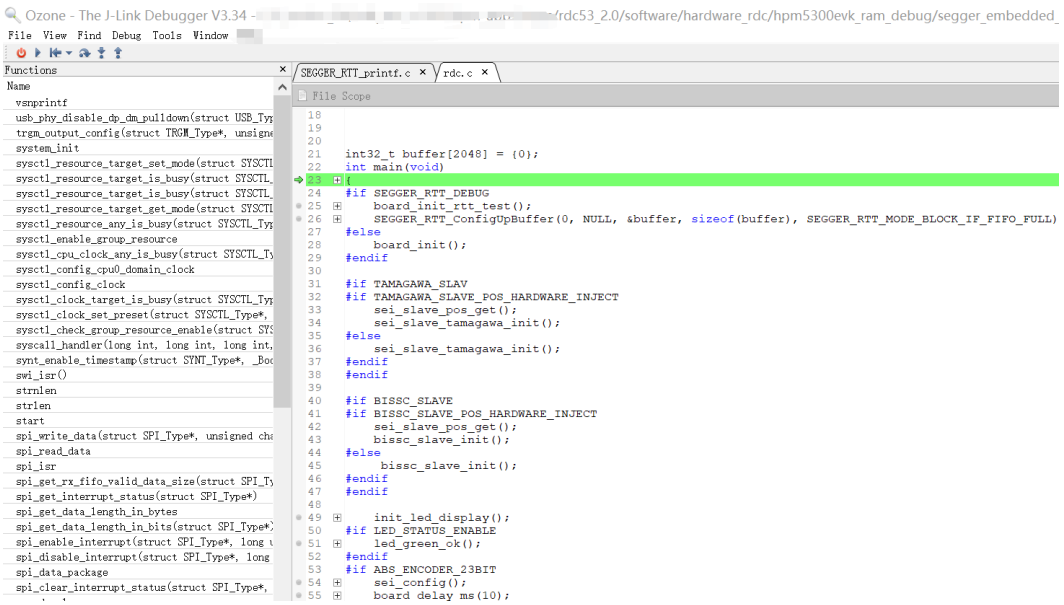

图36 程序等待运行中

（3）打开Jlink-RTT-Logger程序，本文测试使用的是V8.12版本。Target interface设置为JTAG；interface speed设置为9600；RTT control address设置为0x0008242c（在demo.map文件中找到_SEGGER_RTT的地址），RTT channel:0具体设置如图37所示。

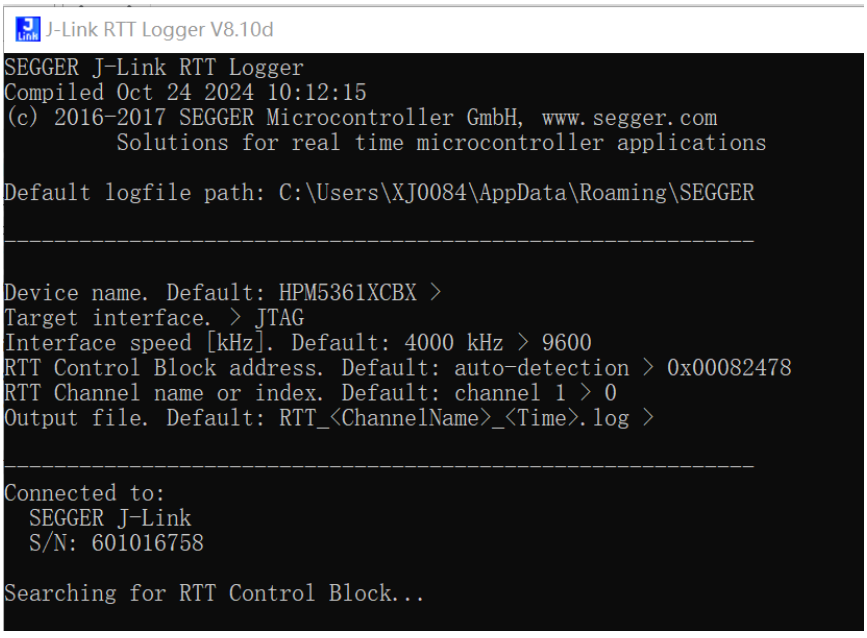

图37 Jlink设置

（4）使用OZONE运行HPM5300_RDC程序，此时Jlink-RTT-Logger开始采集数据；

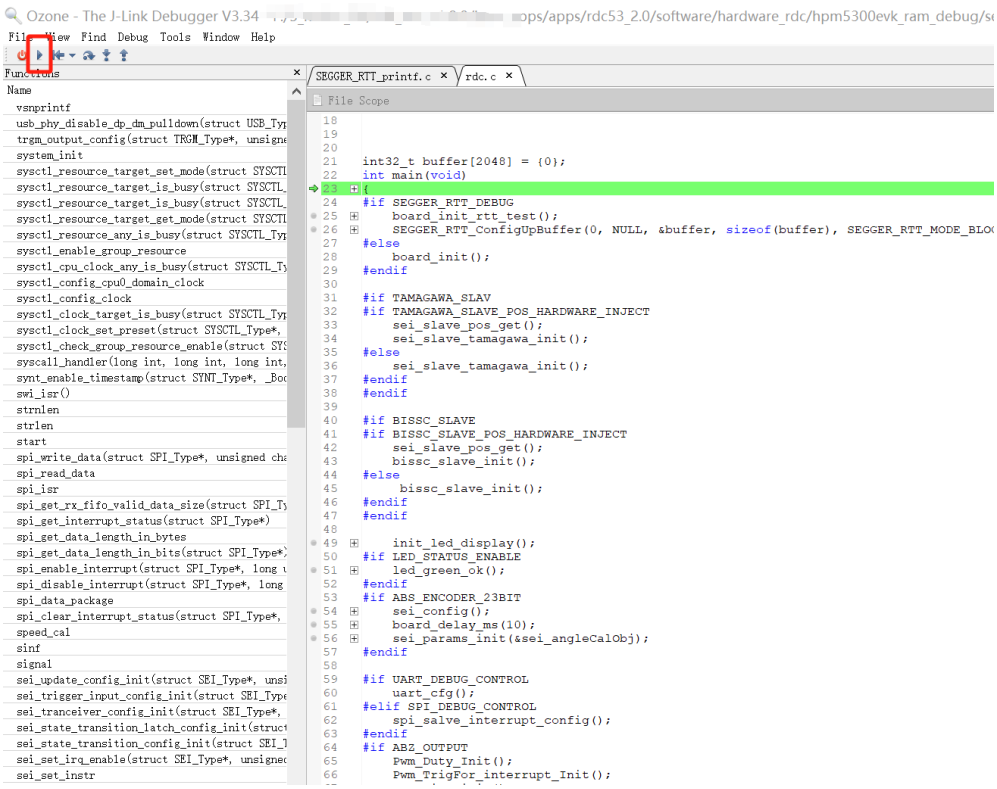

图38 程序运行

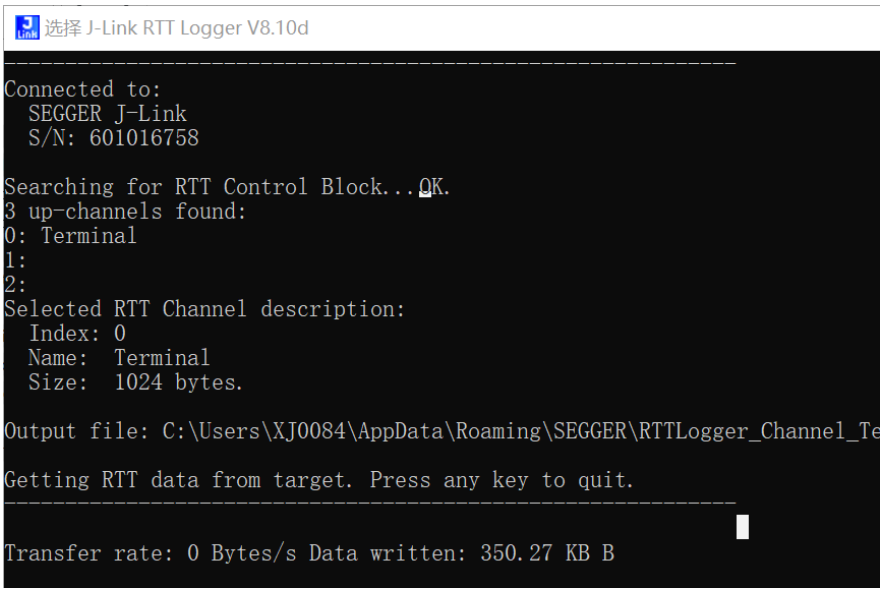

图39 log打印

（5）设置电机转速并运行电机，采集一定量数据后，停止程序运行。

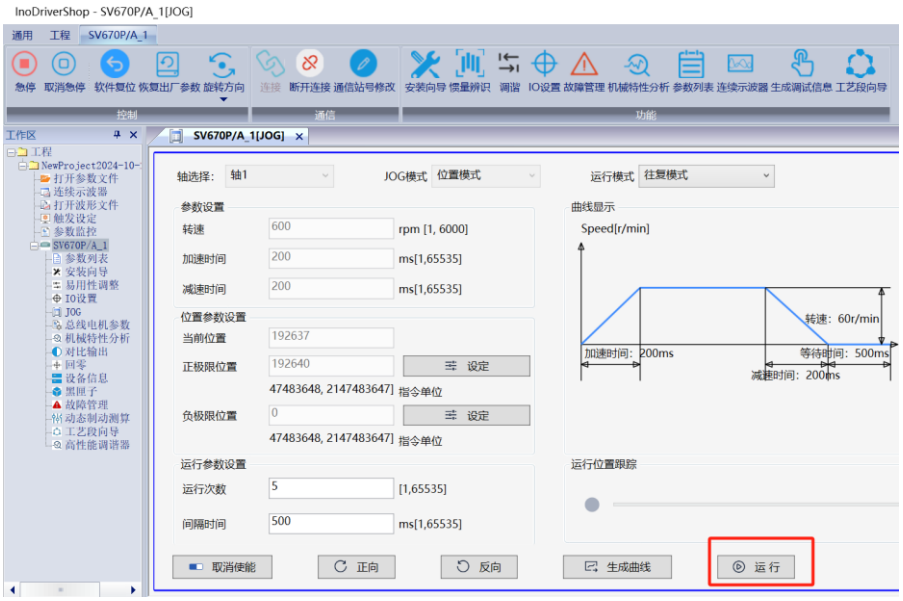

图40 电机运行/停止按钮

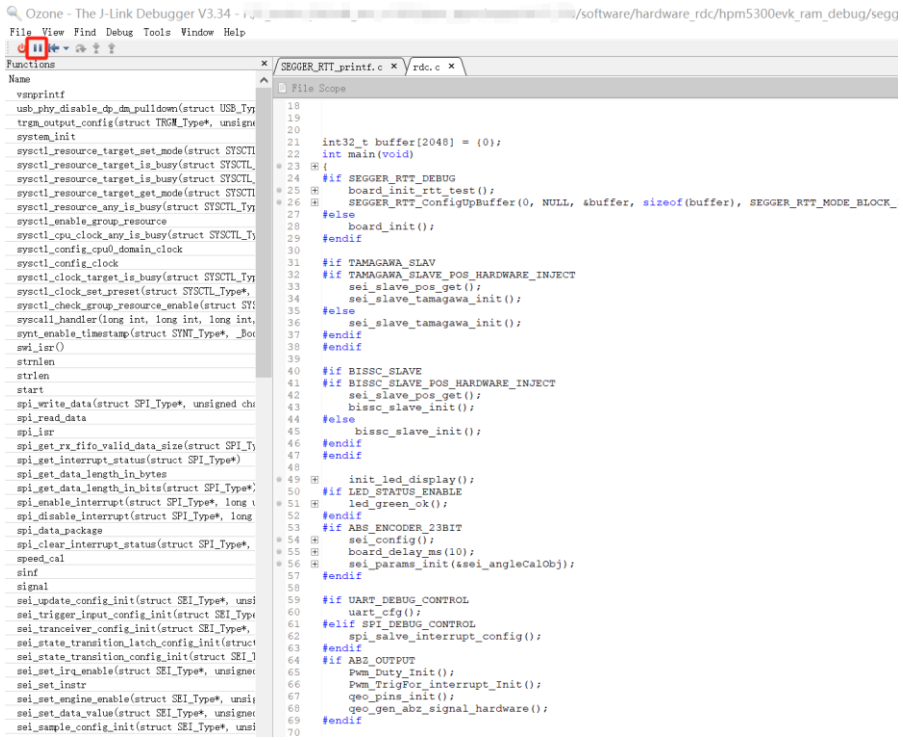

图41 程序停止按钮

（6）在C:\Users\XJXXX\AppData\Roaming\SEGGER

中找到RTTLogger_Channel_Terminal.log 文件。文件中一共五列数据，前三列分别表示SEI与RDC、PLL和PLL_II的误差值，后两列表示PLL与PLL_II计算的速度。各列数据放大了100倍，实际应用中需要除以100处理。用户可以将数据copy到EXCEL表格中处理，处理时将误差中±360度附近的数据（特殊点）去掉。

5.2. 测试结果

5.2.1. 角度误差

表5角度误差

vel_ref(r/s)	sei_rdc_error	sei_pll_error	Sei_pll_ii_error	如图
-10	(-0.51, 1.13)	(-0.2, 1.57)	(-0.04, 1.45)	42
-20	(-1, 0.92)	(-0.91, 2.3)	(-0.4, 1.58)	43
-30	(-1.5, 0.2)	(-0.44, 1.26)	(-0.54, 1.36)	44
-40	(-1.89, 0.26)	(-0.3, 1.59)	(-0.7, 1.93)	45
-50	(-2.42, 0)	(0.047, 1.136)	(-0.79, 1.86)	46

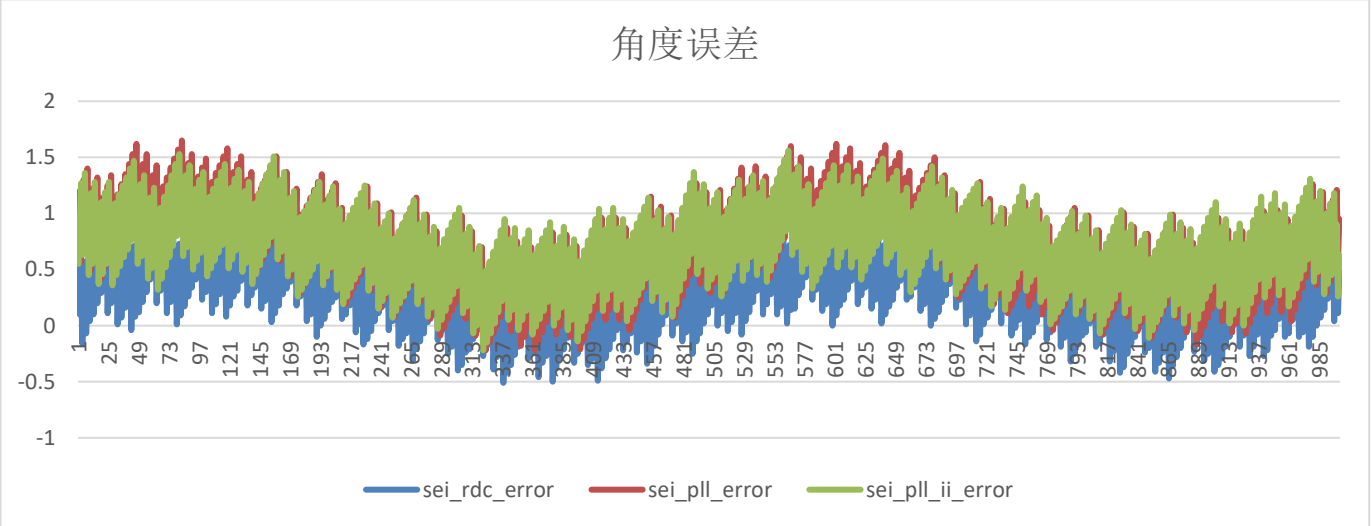

图42 -10r/s角度误差

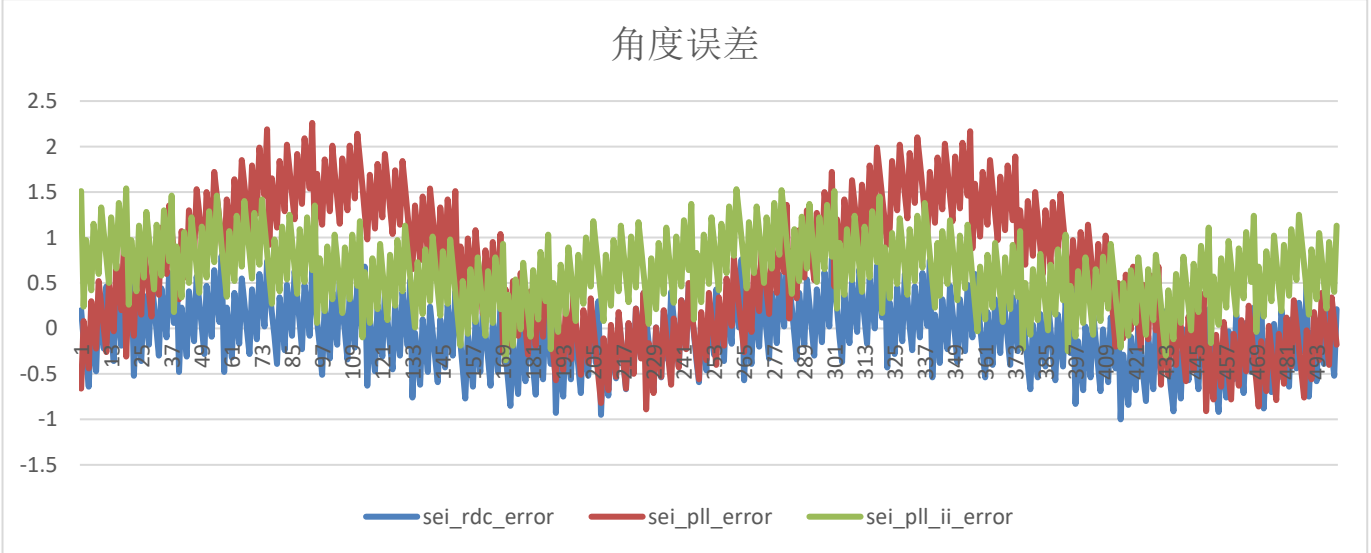

图43 -20r/s角度误差

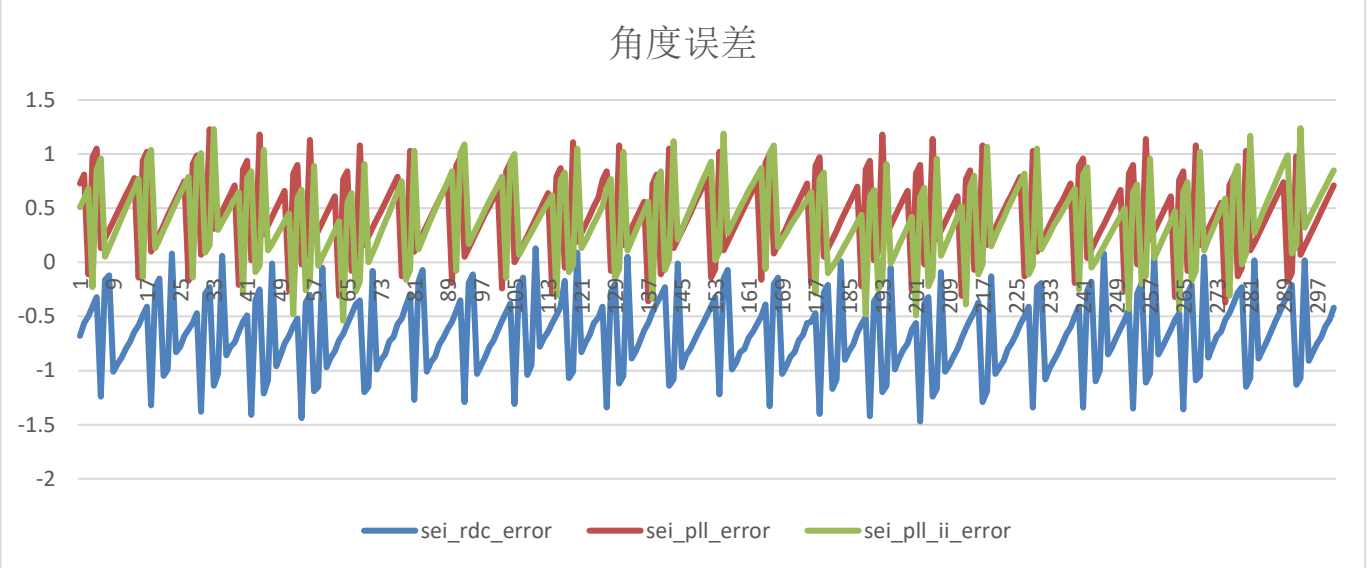

图44 -30r/s角度误差

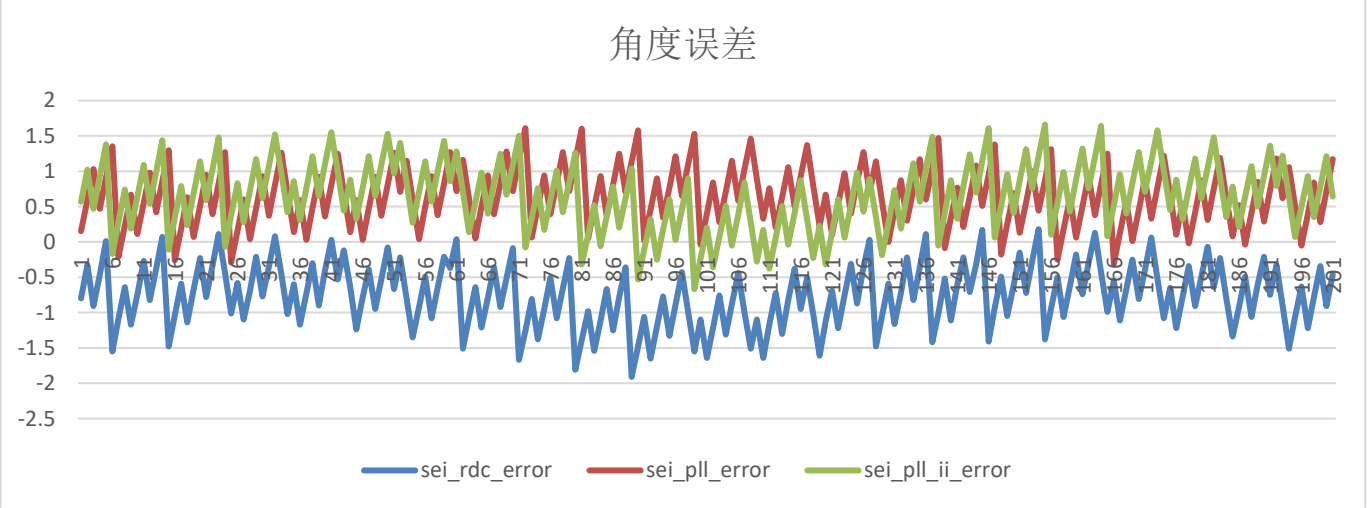

图45 -40r/s角度误差

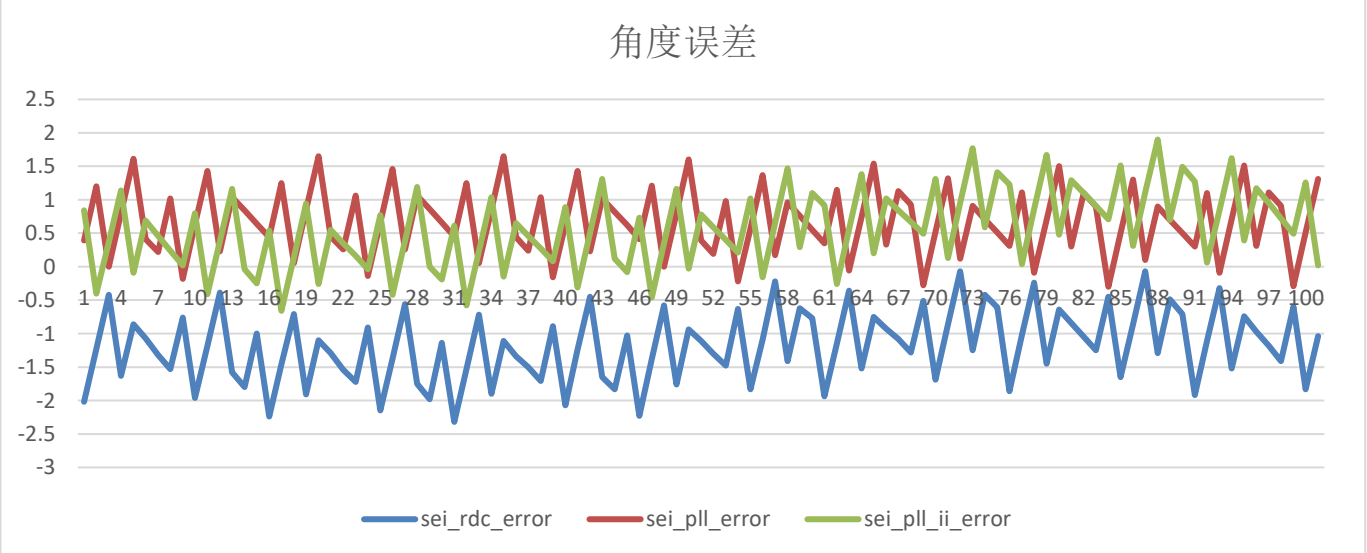

图46 -50r/s角度误差

5.2.2. 速度波动

表6 速度波动

Vel_ref(r/s)	pll_vel	pll_ii_vel	如图
-10	(-9.86, -10.17)	(-9.93, -10.11)	47
-20	(-19.5, -20.7)	(-19.85, -20.14)	48
-30	(-29.9, -30.13)	(-29.9, -30.11)	49
-40	(-39.82, -40.2)	(-39.83, -40.22)	50
-50	(-49.9, -50.14)	(-49.8, -50.22)	51

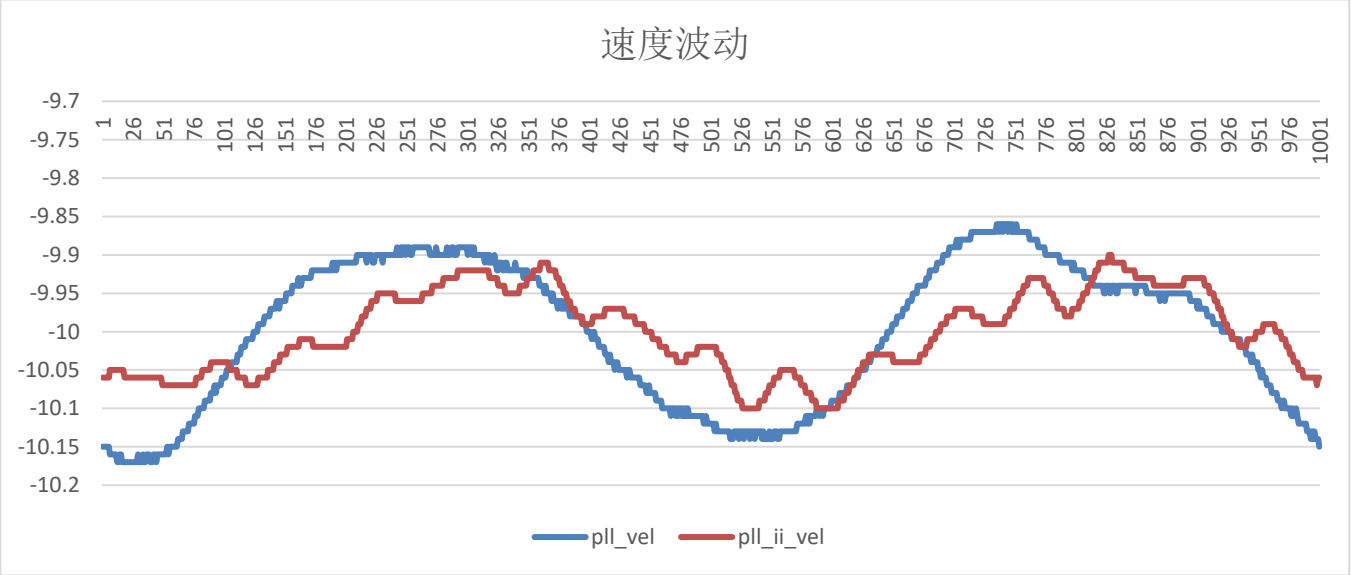

图47 -10r/s速度波动

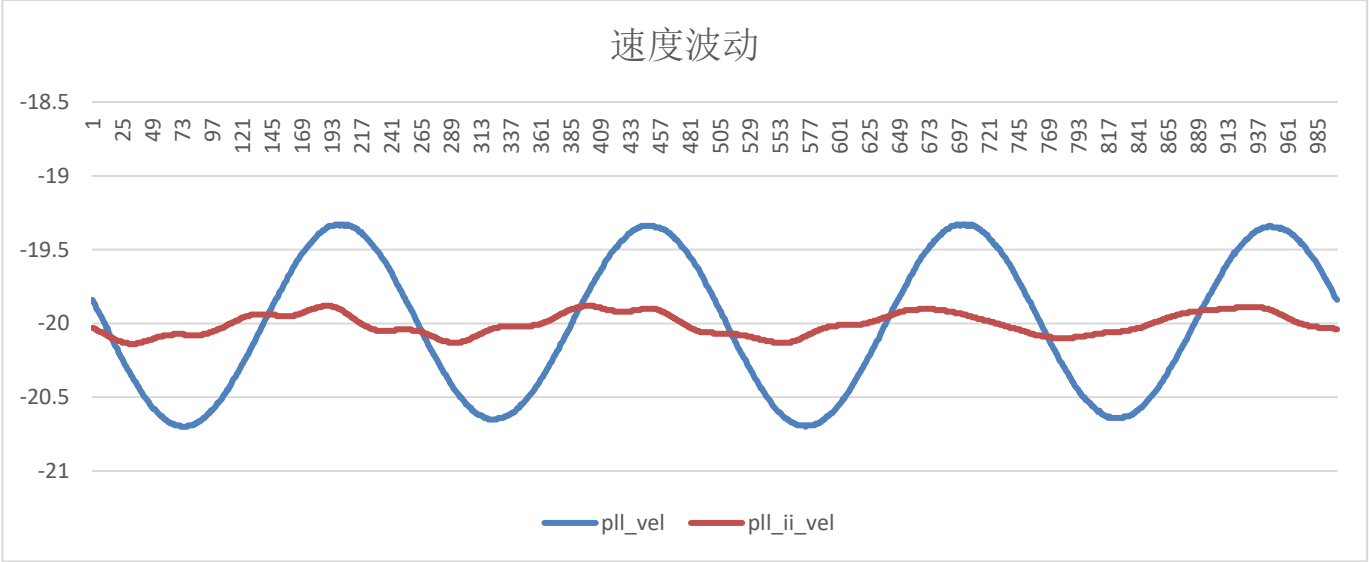

图48 -20r/s速度波动

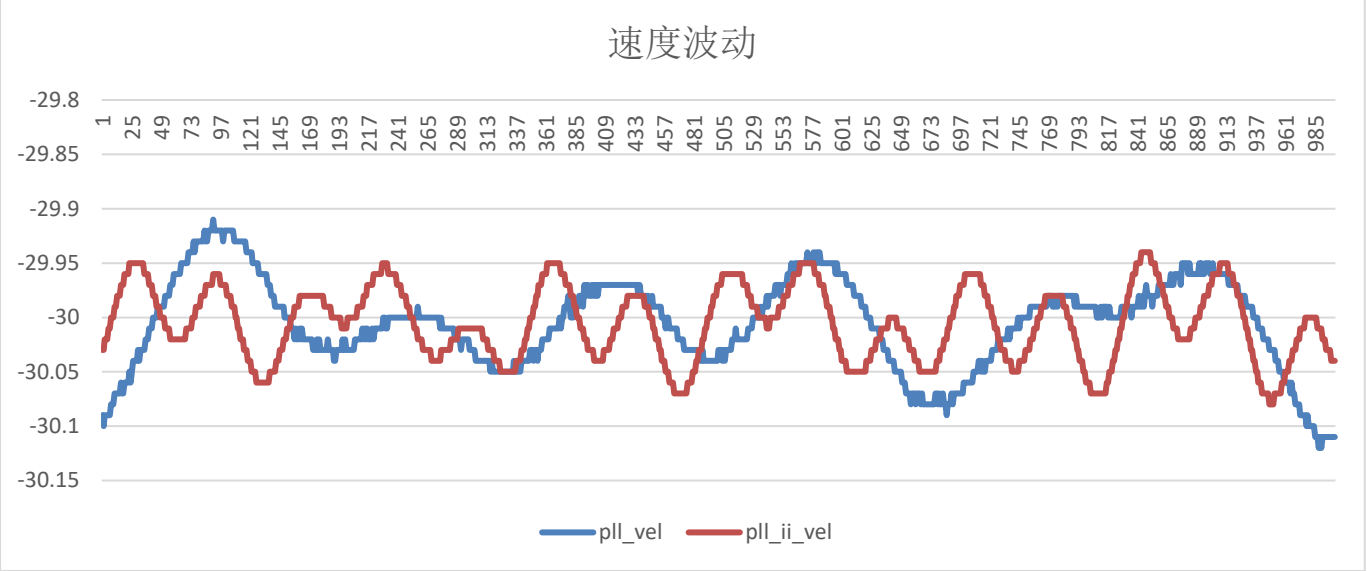

图49 -30r/s速度波动

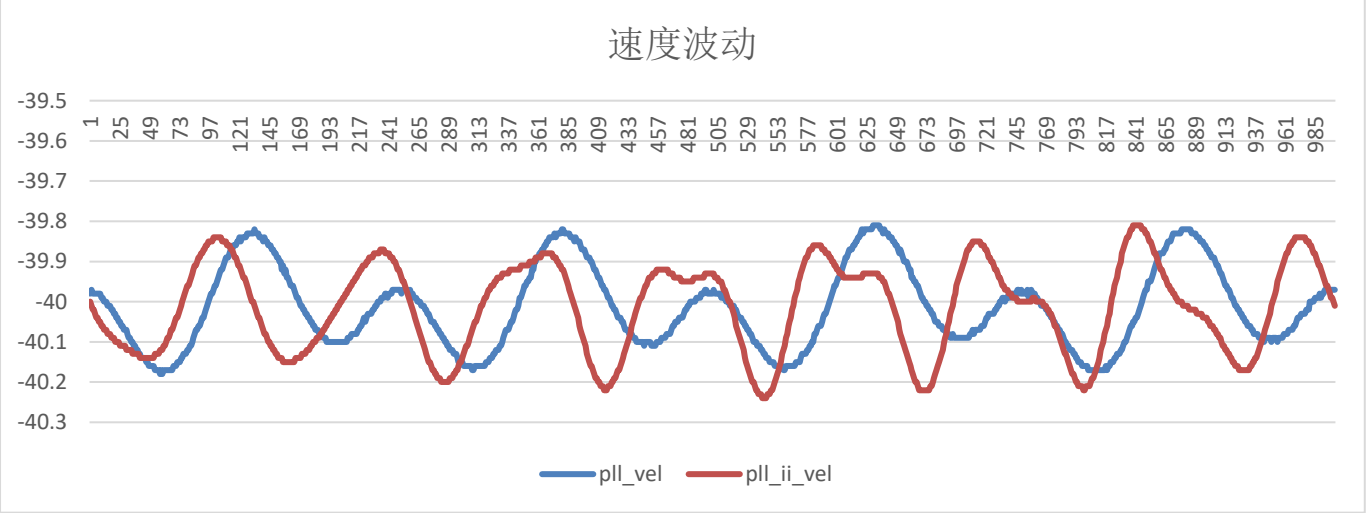

图50 -40r/s速度波动

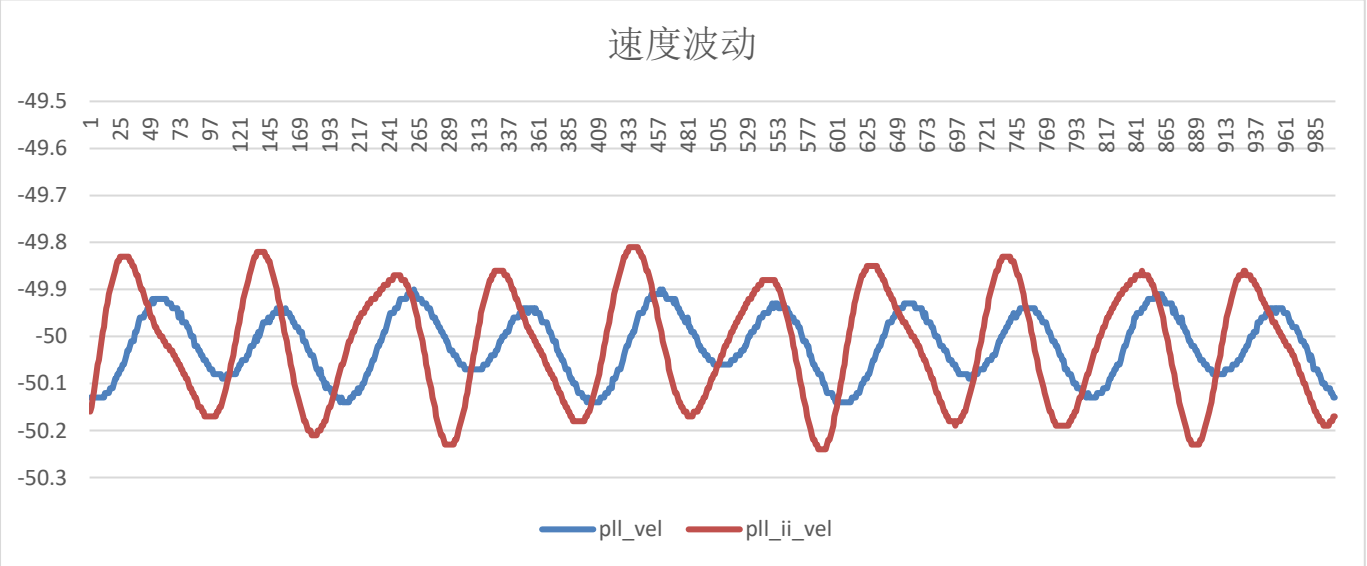

图51 -50r/s速度波动

6. 测试说明

1、例程中使用的是汇川SV670P伺服驱动器和汇川MS1H1-75B30CB-A331R电机，需要安装上位机软件并通过TYPE-C线连接，例程的tools文件夹内有相应的控制软件；

2、通信测试需要使用HPM5300EVK作为测试板，并在程序中选择UART或SPI进行通信；

3、需要注意供电电源的电压和电流，输入电压设置为24V，输入电流需要150mA；

7. 总结

本文介绍了HPM5300_RDC的测试方法，包括静态测试、动态测试、通信功能测试、QEO输出测试、角度误差测试和速度误差测试。给出了每种测试的操作步骤及测试结果。

测试结果：

1、静态测试和动态测试的EXC_P、EXC_N、OSIN和OCOS信号符合要求；

2、UART和SPI通信功能正常，读取的数据与设置的电机参数一致；

3、bissc从模式、多摩川从模式正确输出角度、位置信息；

4、QEO的A/B/Z相输出正常，输出波形与设置的电机参数相符合；

5、在10r/s、20r/s、30r/s、40r/s和50r/s转速下分别测试了角度误差数据，与绝对值编码器相比，RDC反正切角度误差最大为±2.5电角度，PLL观测器角度误差最大为±2.3电角度，PLL_II角度误差最大为±2电角度，均满足±3的spec。

6、中低高各速度下，PLL与PLL_II速度波动范围为±0.3r/s。