目录

[1 BOOT及操作系统 1](#_Toc502612222)

[2 人机交互界面 1](#_Toc502612223)

[2.1 信息显示页面 1](#_Toc502612224)

[2.2 波控调试页面 2](#_Toc502612225)

[2.3 版本管理页面 3](#_Toc502612226)

[2.3.1 BOOT版本 3](#_Toc502612227)

[2.3.2 LINUX版本 3](#_Toc502612228)

[2.3.3 PL版本 3](#_Toc502612229)

[2.3.4 FPGA版本 3](#_Toc502612230)

[2.4 卫星指向角度计算 5](#_Toc502612231)

[2.5 在线调试功能 5](#_Toc502612232)

[3 FPGA寄存器说明 6](#_Toc502612233)

[3.1.1 FPGA版本与在位测试 6](#_Toc502612234)

[3.1.2 中断寄存器 7](#_Toc502612235)

[3.1.3 调试模式寄存器 7](#_Toc502612236)

[3.1.4 单通道调试 7](#_Toc502612237)

[3.1.5 RS422串口监控与调试 7](#_Toc502612238)

[3.1.6 校准值配置 8](#_Toc502612239)

本设计软件应包括的内容有：

1. BOOT及操作系统
2. 人机交互界面
3. 卫星指向角度计算
4. 在线调试功能

下面对各部分功能做出说明。

# BOOT及操作系统

BOOT及UBOOT存放在QSPI中，原则上出厂后不允许改动。

操作系统存放在EMMC中，可以在线进行升级。如果升级成功则使用新的系统，如升级不成功则继续使用原系统，并通过网页日志中记录并在状态中说明（记录内容包括日期，升级版本，如不成功记录失败原因）。

操作系统加载后，加载两片FPGA程序（加载方式见2.3.4节）

注：调试阶段可以都放在SD卡中，由TF启动。

# 人机交互界面

本项目人机交互采用网页方式完成，调试网页分为三部分：

1. 信息显示与监控页面。
2. 相控阵调试页面。
3. 版本管理页面
4. 在线调试功能

下面对各页面内容进行详细说明

## 信息显示页面

|  |  |  |  |
| --- | --- | --- | --- |
| 功能 | 内容 | 实现 | 备注 |
| 用户登录 | 用户密码口令输入界面 | 可以参考家用路由器界面 | 用户名：admin  默认口令：admin  口令可在界面修改，该口令应存于QSPI FLASH，不会因版本升级而失效 |
| 版本显示 | 1. BOOT 版本号 2. 软件版本号 3. FPGA版本号 | * BOOT 版本号软件记录软件版本号 * FPGA版本号（FPGA0和FPGA1） * Linux版本号 * APP版本号 | 含版本号，生成日期  PS和FPGA版本由寄存器读出，具体寄存器地址见第4.1.1 |
| 电压温度显示 | 1. ZYNQ内部温度显示 2. ZYNQ电压显示 | * ZYNQ内部温度 | ZYNQ内部温度，由PS通过总线读出，操作方法见手册。 |
| 工作状态显示 | 将工作状态从寄存器读出，并解析含义，显示出来 | 读取PL寄存内容并解析 | 分调试模式和工作模式，调试模式下由参数网页控制，正常工作下为设备正常工作。 |
| 维护日志 | 将操作日志保存成文件，写入FLASH，维护时可以从网页读出 | 日志文件存储到FLASH，每月的日志生成一个文件 | 日志文件可从网口提取到上位机，也可在线删除 |

## 波控调试页面



波控调试页面参考原上图。

包括如下内容：

|  |  |  |  |
| --- | --- | --- | --- |
| 功能 | 内容 | 实现 | 备注 |
| FLASH操作 | 根据界面上的操作对FLASH校表区域进行擦除与读写操作。 |  |  |
| 角度配置 | 根据输入的卫星信息处出theta,phy,然后由FPGA算出相位置入天线组件。 |  |  |
| 单通道调试 | 将输入的通道号，相位值与衰减值写入到对应的FPGA寄存器中。 |  |  |

## 版本管理页面



### BOOT版本

BOOT及UBOOT：支持ZYNQ启动，原则上出厂后不允许更新，但预留相应功能以便调试。

### LINUX版本

Linux 及BSP允许更新，但不覆盖原版本，更新后自动重起，如发现更新后的版本不可用，回退到以前版本，并通过网页通知上位机更新失败原因。确认软件可用后，将默认加载设置为新版本。

### PL版本

在版本更新的情况下，用户通过网页通知ZYNQ更新的版本号与下载文件，ZYNQ将下载的PL烧写文件写入FLASH中的临时版本区域，重启系统，如新的PL版本可以正常加载，则将该版本写入对应版本区域。

### FPGA加载

FPGA是基于SRAM编程的，编程信息在系统掉电时会丢失，每次上电时，都需要从器件外部的FLASH或EEPROM中存储的编程数据重现写入内部的SRAM中。FPGA在线加载需要有PS的帮助，并且在加载前PS已经启动并工作。

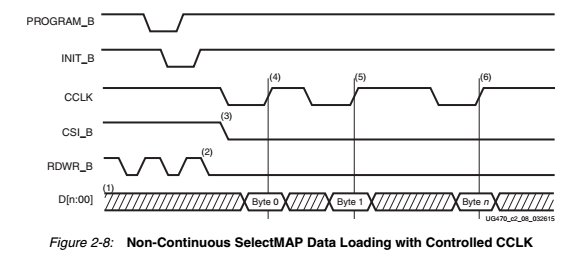
FPGA采用被动加载模式：

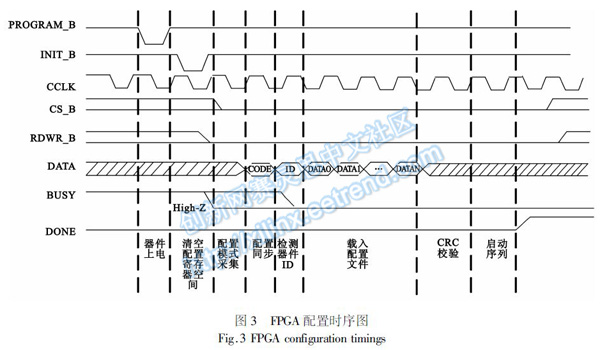
PS模式(Passive Serial Configuration Mode)，即被动串行加载模式。PS模式适合于逻辑规模小，对加载速度要求不高的FPGA加载场合。在此模式下，加载所需的配置时钟信号CCLK由FPGA外部时钟源或外部控制信号提供。另外，PS加载模式需要外部微控制器ZYNQ的支持。

PP模式(Passive Parallel Configuration Mode)，即被动并行加载模式。此模式适合于逻辑规模较大，对加载速度要求较高的FPGA加载场合。PP模式下，外部设备通过8bit并行数据线对FPGA进行逻辑加载，CCLK信号由ZYNQ提供。

在本项目中我们采用PS模式与PP模式兼容模式。

PP模式的FPGA加载方法如下：

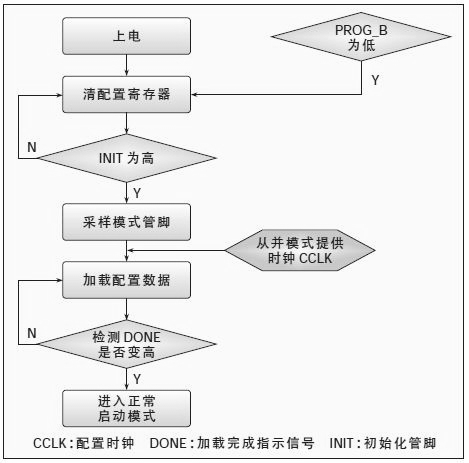




管脚定义如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 管脚 | 方向 | 功能 | 备注 |
| CCLK | Zynq🡪FPGA | I/O信号，配置模式下的时钟输入,上升沿有效 | 上沿有效 |
| PROG\_B | Zynq🡪FPGA | 输入信号（低电平有效），低电平时，异步复位芯片，为接收加载数据作准备。 | 低有效 |
| INTI\_B | FPGA🡪Zynq | 初始化完成信号，由低变高表示FPGA初始化完成。如果为低电平表示加载错误。 | OD输出，使用时需要上拉到VCC |
| DONE | FPGA🡪Zynq | 加载完成指示信号，I/O信号。 | OD输出，使用时需要上拉到VCC |
| CS\_B | Zynq🡪FPGA | 片选信号，低有效，表示启动从并口加载的总线 |  |
| RDWR\_B | Zynq🡪FPGA | 写信号，低有效，启动从并口的写使能 | 低有效 |
| DATA | Zynq🡪FPGA | 输入信号，加载数据输入，与CCLK信号的上升沿同步 | 8位或16位 |
| M | 上下拉电阻 | PP模式为110，PS为111 | 不需要占用zynq管脚 |

加载过程如下：



加载函数流程



|  |
| --- |
|  |

#### 版本更新

在版本更新的情况下，用户通过网页通知ZYNQ更新的版本号与下载文件，ZYNQ将下载的PL烧写文件写入FLASH中的临时版本区域，重启系统，如新的FPGA版本可以正常加载，则将该版本写入对应版本区域。否则通网页显示加载失败。

## 卫星指向角度计算

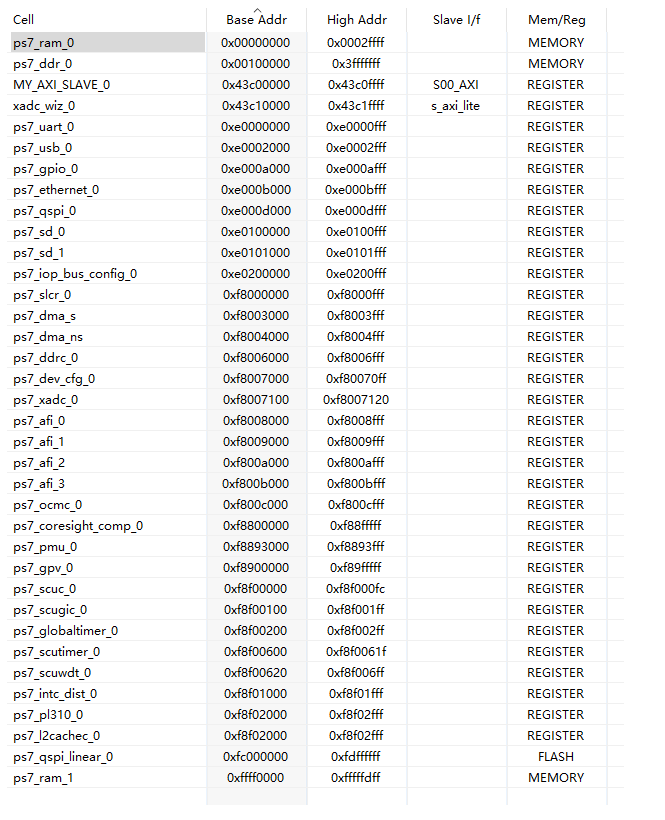
具体计算公式见MATLAB程序，要求使用浮点计算。请根据输入参数自定义控制界面。

## 在线调试功能

软件可从总线上挂载的RAM中读取指定长度的数据，并存成文件，通过网口上传到上位机进行分析。具体内容待定。

|  |  |
| --- | --- |
| 计算模块调试模式 | 从FPGA寄存器回读出计算出来的相位值，并与PS计算出来的相位值进行比较，将值打印出来 |
| RS422发送调试 | 由网页设置发送数据，由串口TXD发送. |
| RS422收发环回 | 串口RXD的串行数据直接返回串行的TXD以测试硬件是否正常 |
| RS422数据环回 | 串口RXD的数据串转并后直接返回TXD进行并串转换以测试数据收发是否正常 |

# 寄存器空间



# FPGA寄存器说明

本项目中含有一片ZYNQ，两片FPGA。PS需要访问的地址有PL,FPGA0,FPGA1.三块空间。

对PL的访问直接采用AXI总线，对FPGA0的访问通过PL与FPGA间的接口转接到AXI上。

PS与PL/FPGA的通信采用AXI接口实现，ARM通过对FPGA寄存器的读写实现对其它外扩器件的控制。

R/W：读写

RO： 只读

对于两片FPGA采用同样的逻辑。所以寄存器内容也保持一致。只是采用不同的控制寄存器加以区分。

### FPGA版本与在位测试

记录FPGA版本信息及读写测试寄存器。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 地址类型 | 寄存器名 | 寄存器内容 | 寄存器偏移地址 | 位宽 | 读写 | 含义 |
| PL版本号 | PS\_VER\_REG | FPGA版本寄存器 | 0x000 | 16 | RO | 版本号，如0x12\_34解析为12.34版本 |
| PL版本日期 | PS\_VER\_DATE | FPGA版本寄存器 | 0x001 | 32 | RO | 版本生成年月日如：0x20180101 |
| PL读写测试寄存器 | FPGA0\_TEST\_REG | 读写测试寄存器 | 0x002 | 8 | W/R | 通过该寄存器的读写确认PL读写是否正常 |
| FPGA读写流程请参见3.1.6 | | | | | | |
| FPGA0 ID号 | FPGA0\_ID\_REG | FPGA标识寄存器 | 0x000 | 8 | RO | 如读到0xA0则表示 FPGA0在位 |
| FPGA0在位测试寄存器 | FPGA0\_TEST\_REG | 读写测试寄存器 | 0x001 | 8 | W/R | 通过该寄存器的读写确认FPGA0读写是否正常 |
| FPGA0版本号 | FPGA0\_VER\_REG | FPGA版本寄存器 | 0x002-0x003 | 8 | RO | 版本号，0x102为主版本号  0x103为子版本号  如0x12\_34解析为12.34版本 |
| FPGA0版本日期 | FPGA0\_DATE\_REG | FPGA版本寄存器 | 0x004-0x007 | 8 | RO | 版本生成年月日如：0x20180101 |
| FPGA读写流程请参见3.1.6 | | | | | | |
| FPGA1 ID号 | FPGA1\_ID\_REG | FPGA标识寄存器 | 0x000 | 8 | RO | 如读到0xa1则表示 FPGA1在位 |
| FPGA1在位测试寄存器 | FPGA1\_TEST\_REG | 读写测试寄存器 | 0x001 | 8 | W/R | 通过该寄存器的读写确认FPGA1读写是否正常 |
| FPGA1版本号 | FPGA1\_VER\_REG | FPGA版本寄存器 | 0x002-0x003 | 8 | RO | 版本号，0x202为主版本号  0x203为子版本号  如0x12\_34解析为12.34版本 |
| FPGA0版本日期 | FPGA0\_DATE\_REG | FPGA版本寄存器 | 0x004-0x007 | 8 | RO | 版本生成年月日如：0x20180101 |

### 中断寄存器

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 地址类型 | 寄存器名 | 寄存器内容 | 寄存器地址 | 位宽 | 读写 | 含义 |
| 中断寄存器 | Int\_reg | 中断标志 | 0x010 | 1 | RO | 1：有中断，0：无中断 |
| 中断清除 | Int\_clr\_reg | 中断清除 | 0x011 | 1 | W/R | 写0🡪1🡪0,请除中断标志 |
| 中断类型寄存器 | Int\_typ\_reg | 中断类型 | 0x012 | 32 | RO | Bit0:RS422接收  Bit1:接收校准值写flash  Bit2:FLASH擦除 |

### 调试模式寄存器

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 地址类型 | 寄存器名 | 寄存器内容 | 寄存器地址 | 位宽 | 读写 | 含义 |
| 模式选择寄存器 | Mode\_sel | 中断标志 | 0x020 | 1 | W/R | 1：调试模式，0：正常模式 |
| 轴向角设置 | theta\_reg | 中断清除 | 0x021 | 16 | W/R | theta角度 |
| 入射角设置 | Phy\_reg | 中断类型 | 0x022 | 16 | W/R | phy角度 |
| 工作频率 | Fmhz\_reg | 工作频率 | 0x023 | 16 | W/R | 频率，单位MHZ |
| 整体开关 | all\_power\_sw | 整体通道开关 | 0x024 | 1 | W/R | 1：全开，0：全关 |
| 角度有效 | Angle\_dv | 角度有效标志 | 0x025 | 1 | W/R | 写0🡪1🡪0,角度置入启动计算 |
| 计算值通道号 | Calc\_ch\_no\_reg | 回读通道号 | 0x026 | 16 | W/R | 写入通道号，范围为0-0x17 |
| 计算值回读寄存器 | Calc\_value | 计算值寄存器 | 0x027 | 16 | RO | 通道对应的计算值：  Bit7-bit0:相位值  Bit15-bit8:衰减值 |
| 计算值回读启动 | phase\_rb | 回读计算寄存器 | 0x028 | 16 | W/R | 写0🡪1🡪0,启动回读，更新回读值 |

### 单通道调试

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 地址类型 | 寄存器名 | 寄存器内容 | 寄存器地址 | 位宽 | 读写 | 含义 |
| 单通道设置 | single\_ch\_data | 单通道设置的通道号与设置参数 | 0x030 | 32 | R/W | BIT[31:16] 通道号  BIT[15:0] 通道设置数据  其中：  bit[5:0]为衰减设置值  bit[12:8]为相位设置值  bit[15]为通道开关（1开，0关） |
| single\_set\_valid | 发送FIFO控制寄存器 | 0x031 | 1 | R/W | BIT0: 写0 —> 1 🡪 0  写入一次（仅在调试模式下有效即Mode\_sel=1） |

### RS422串口监控与调试

RS-485串行通讯标准，硬件连接成半双工方式。传输波特率为115200 bit/s，启动位一位，数据位八位，停止位一位，无奇偶校验，每个数据字节总位数为10位，字节(10位)传输速率为：11520 byte/s。

ARM的422接口操作通过读写FPGA寄存器完成。485接口底层驱动采用FPGA实现，并由FPGA实现信息的解析，FPGA收到命令后通知，由PS 通过寄存器来读取数据并显示和记录。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 地址类型 | 寄存器名 | 寄存器内容 | 寄存器地址 | 位宽 | 读写 | 含义 |
| 接收命令帧长寄存器 | RX\_Frame\_length | 接收命令帧长寄存器 | 0x040 | 16 | RO | RS422读取的数据长度 |
| 接收命令合法性标志 | RX\_\_frame \_Error | 接收命令合法性标志 | 0x041 | 1 | RO | 0：正确帧  1：错误帧 |
| 接收命令读取控制 | monitor\_fifo\_ctrl | 收发FIFO控制 | 0x042 | 1 | W/R | RS422监控FIFO控制  BIT0: RS422接收FIFO清空  BIT1:写0🡪1🡪0，启动FIFO 读数据 |
| 接收命令读取数据 | RX\_data | 接收命令读取数据 | 0x043 | 8 | RO | 读取的RS422接收数据 |
| 发送数据写控制 | Tx\_frame\_wr | PS 通过RS422发送标志 | 0x044 | 1 | W/R | 写0🡪1🡪0写入1个数据（仅在调试模式下有效即Mode\_sel=1，且RS422\_LOOP=0x03有效） |
| 接收命令读取数据 | Tx\_frame\_data | PS 通过RS422发送的数据 | 0x045 | 8 | W/R | PS 通过RS422发送的数据 |
| RS422收发环回 | RS422\_LOOP | RS422收发环回 | 0x046 | 8 | W/R | 仅在调试模式下有效（Mode\_sel=1）  0x00:正常工作模式，PS可通过FIFO监控收发数据  0x01:串口RXD的串行数据接收并返回串行的TXD以测试硬件是否正常  0x02:串口RXD的数据串转并后接收并返回TXD进行并串转换以测试数据收发功能是否正常  0x03: PS发送RS422 数据 |
| 收发监控 | I\_ps\_fifo\_rs422\_tx\_data | RS422发送的数据 | 0x047 | 8 | RO | RS422发送数据监控 |
| I\_ps\_fifo\_rs422\_rx\_status | 接收FIFO状态 | 0x048 | 16 | RO | RS422接收监控FIFO状态  Bit15:满标志  Bit14:空标志  Bit13:将满标志（数据量大于1024\*3）  Bit12:将空标志 （数据量小于1024）  Bit11-bit0:FIFO中的数据量 |
| I\_ps\_fifo\_rs422\_tx\_status | 发送FIFO状态 | 0x049 | 16 | RO | RS422发送监控FIFO状态  Bit15:满标志  Bit14:空标志  Bit13:将满标志（数据量大于1024\*3）  Bit12:将空标志 （数据量小于1024）  Bit11-bit0:FIFO中的数据量 |

### FPGA加载及读写

加载流程见2.3.4节。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 地址类型 | 寄存器名 | 寄存器内容 | 寄存器地址 | 位宽 | 读写 | 含义 |
| FPGA加载控制 | FPGA\_config | FPGA加载标志 | 0x050 | 1 | W/R | Bit0控制FPGA0  0: nomal 1: config  Bit1控制FPGA1:  0: nomal 1: config |
| FPGA\_cfg\_ctl | FPGA加载控制信号 | 0x051 | 4 | W/R | Bit0:FPGA0\_cclk Bit1:FPGA0\_rw\_n  Bit2:FPGA0\_csi\_n  Bit3:FPGA0\_prg\_n |
| FPGA\_cfg\_state | FPGA加载状态 | 0x052 | 2 | RO | Bit0: FPGA0\_init\_n  Bit1: FPGA0\_done |
| FPGA加载数据 | FPGA\_config\_dat | FPGA配置数据 | 0x053 | 8 | W/R | 两片FPGA加载与读写共用寄存器 |
| FPGA寄存器读写 | FPGA\_rw\_ctrl | FPGA读写控制 | 0x054 | 2 | W/R | Bit0: 发起读写  Bit2-bit1:  11: 写所有FPGA  10: 读FPGA0  01: 读FPGA1 |
| FPGA\_rw\_addr | FPGA读写地址 | 0x055 | 8 | W/R | 读写地址 |
| FPGA\_wr\_data | FPGA写入数据 | 0x056 | 8 | W/R | 写数据 |
| FPGA\_rd\_data | FPGA读出数据 | 0x057 | 8 | RO | 读数据 |

写FPGA寄存器流程：



读FPGA寄存器流程：



### FLASH控制及校准值读写

#### 校准出厂写表

校准值在出厂时通过RS422命令写入，PS将数据由FPGA读出，然后写入到FLASH。



#### 校准上电初始化

每次系统上电后，PS需要从FLASH中读取校准数据，写到FPGA的寄存器中，流程如下：



#### 校准值寄存器说明

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 地址类型 | 寄存器名 | 寄存器内容 | 寄存器地址 | 位宽 | 读写 | 含义 |
| 操作结果反馈 | flash\_op\_feedback | FLASH操作结果反馈给FPGA | 0x060 | 3 | W/R | 在收到FLASH读写或擦除中断后，进行相应操作，然后将操作结果反馈给FPGA  BIT2-BIT0:  100: flash write not ok;  101: flash write ok;  110: flash erasure not ok;  111: flash erasure ok; |
| 出厂写表 | calibration\_ch\_no | 数据读写校准值地址 | 0x062 | 16 | R/W | 数据读取通道号 |
| calibration\_rd\_data | 校准值初始数据 | 0x061 | 32 | RO | BIT[31:16] 校准通道号  BIT[15:0] 校准通道数据，其中：  bit[4:0]为衰减  bit[13:8]为相位校准值  由于计算需要这些校准值需要写入到寄存器中，16个通道为一组，组内通道以BIT[19:16]区分，通道共有5X4=20组，组号以BIT[24:20]区分 |
| 上电初始化 | calibrat\_initial\_wr | 数据写标志 | 0x063 | 1 | R/W | 系统上电并加载FPGA后，需要从FLASH将校准值读出，并初始化到FPGA中。  BIT0: 写0 —> 1 🡪 0更新一个通道值到FPGA缓存中， |
| calibration\_wr\_data | 校准值初始数据 | 0x061 | 32 | W/R | BIT[31:16] 校准通道号  BIT[15:0] 校准通道数据，其中：  bit[4:0]为衰减  bit[13:8]为相位校准值  由于计算需要这些校准值需要写入到寄存器中，16个通道为一组，组内通道以BIT[19:16]区分，通道共有5X4=20组，组号以BIT[24:20]区分 |
| CALIBRATION\_renew | 校验数据更新 | 0x064 | 1 | R/W | BIT0: 写0 —> 1 🡪 0将校准值从缓存更新到所有通道 |

### 延迟线控制

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 地址类型 | 寄存器名 | 寄存器内容 | 寄存器地址 | 位宽 | 读写 | 含义 |
| 组件延时 | SUBA01\_DLY | 对应组件3位延时位 | 0x070 | 3 | W/R | Bit2:bit0  对应组件3位延时位 |
| 组件延时 | SUBA02\_DLY | 对应组件3位延时位 | 0x071 | 3 | W/R | Bit2:bit0  对应组件3位延时位 |
| 组件延时 | SUBA03\_DLY | 对应组件3位延时位 | 0x072 | 3 | W/R | Bit2:bit0  对应组件3位延时位 |
| 组件延时 | SUBA04\_DLY | 对应组件3位延时位 | 0x073 | 3 | W/R | Bit2:bit0  对应组件3位延时位 |
| 组件延时 | SUBA05\_DLY | 对应组件3位延时位 | 0x074 | 3 | W/R | Bit2:bit0  对应组件3位延时位 |
| 组件延时 | SUBA06\_DLY | 对应组件3位延时位 | 0x075 | 3 | W/R | Bit2:bit0  对应组件3位延时位 |
| 组件延时 | SUBA07\_DLY | 对应组件3位延时位 | 0x076 | 3 | W/R | Bit2:bit0  对应组件3位延时位 |
| 组件延时 | SUBA08\_DLY | 对应组件3位延时位 | 0x077 | 3 | W/R | Bit2:bit0  对应组件3位延时位 |
| 组件延时 | SUBA09\_DLY | 对应组件3位延时位 | 0x078 | 3 | W/R | Bit2:bit0  对应组件3位延时位 |
| 组件延时 | SUBA10\_DLY | 对应组件3位延时位 | 0x079 | 3 | W/R | Bit2:bit0  对应组件3位延时位 |
| 组件延时 | SUBA11\_DLY | 对应组件3位延时位 | 0x07A | 3 | W/R | Bit2:bit0  对应组件3位延时位 |
| 组件延时 | SUBA12\_DLY | 对应组件3位延时位 | 0x07B | 3 | W/R | Bit2:bit0  对应组件3位延时位 |