

深圳维特智能科技有限公司

SDK API 接口说明书

产品名称	关于 SDK 的 API 接口
产品协议	Normal、Modbus、IIC、CAN
测试日期	2022-05-20

目录

	2
1. SDK 概述	- 1 -
2. API 接口	- 1 -
2.1 int32_t WitSerialWriteRegister(SerialWrite Write_func)	- 1 -
2.2 void WitSerialDataIn(uint8_t ucData)	- 1 -
2.3 int32_t WitI2cFuncRegister(WitI2cWrite,write_func,WitI2cRead read_func)	- 1 -
2.4 int32_t WitCanWriteRegister(CanWrite Write_func)	- 2 -
2.5 void WitCanDataIn(uint8_t ucData[8], uint8_t ucLen)	- 2 -
2.6 int32_t WitRegisterCallBack(RegUpdateCb update_func)	- 2 -
2.7 int32_t WitWriteReg(uint32_t uiReg, uint16_t usData)	- 2 -
2.8 int32_t WitReadReg(uint32_t uiReg, uint32_t uiReadNum)	- 3 -
2.9 int32_t WitInit(uint32_t uiProtocol, uint8_t ucAddr)	- 3 -
2.10 void WitDelnit(void)	- 3 -
2.11 int32_t WitDelayMsRegister(DelaymsCb delayms_func)	- 3 -
2.12 int32_t WitStartAccCali(void)	- 4 -
2.13 int32_t WitStopAccCali(void)	- 4 -
2.14 int32_t WitStartMagCali(void)	- 4 -
2.15 int32_t WitStopMagCali(void)	- 4 -
2.16 int32_t WitSetUartBaud(int32_t uiBaudIndex)	- 4 -
2.17 int32_t WitSetCanBaud(int32_t uiBaudIndex)	- 5 -
2.18 int32_t WitSetBandwidth(int32_t uiBaudWidth)	- 5 -
2.19 int32_t WitSetOutputRate(int32_t uiRate)	- 5 -
2.20 int32_t WitSetContent(int32_t uiRsw)	- 5 -
3. 联系我们	- 6 -

1. SDK 概述

SDK 的 API 接口函数是基于 C 语言开发完成的，示例程序中的协议有 4 种；

- ①Normal 协议：例如 JY901、JY901S 等支持 Normal 协议系列传感器；
- ②Modbus 协议：例如 HWT905-485、WT901C485 等支持 Modbus 协议系列传感器；
- ③IIC 协议：例如 JY901S、JY61P 等支持 IIC 协议系列传感器；
- ④CAN 协议：例如 HWT901B-CAN 等支持 CAN 协议系列传感器；

2. API 接口

2.1 int32_t WitSerialWriteRegister(SerialWrite Write_func)

函数名称	WitSerialWriteRegister
函数原型	int32_t WitSerialWriteRegister(SerialWrite Write_func)
函数功能	串行口写注册
输入参数	“Write_func” 指向函数的指针
返回值	WIT_HAL_INVALID：无效的参数，说明传入的函数指针为空 WIT_HAL_OK：正确

2.2 void WitSerialDataIn(uint8_t ucData)

函数名称	WitSerialDataIn
函数原型	void WitSerialDataIn(uint8_t ucData)
函数功能	串行数据输入
输入参数	“ucData”为传入的数据
返回值	无

2.3 int32_t WitI2cFuncRegister(WitI2cWrite,write_func,WitI2cRead read_func)

函数名称	WitI2cFuncRegister
函数原型	int32_t WitI2cFuncRegister(WitI2cWrite write_func, WitI2cRead read_func)
函数功能	IIC 协议函数的注册
输入参数	“write_func” 传入的数据是 IIC 写函数指针 “read_func” 传入的数据是 IIC 读函数指针
返回值	WIT_HAL_INVALID：无效的参数，传入的函数指针可能为空 WIT_HAL_OK：正确

2.4 int32_t WitCanWriteRegister(CanWrite Write_func)

函数名称	WitCanWriteRegister
函数原型	int32_t WitCanWriteRegister(CanWrite Write_func)
函数功能	Can 协议的写注册
输入参数	“write_func” 传入的数据是 Can 写函数指针
返回值	WIT_HAL_INVALID: 无效的参数, 传入的函数指针可能为空 WIT_HAL_OK: 正确

2.5 void WitCanDataIn(uint8_t ucData[8], uint8_t ucLen)

函数名称	WitCanDataIn
函数原型	void WitCanDataIn(uint8_t ucData[8], uint8_t ucLen)
函数功能	Can 协议的数据输入
输入参数	“ucData[8]” 传入数组用来存储数据 “ucLen” 数据的长度
返回值	无

2.6 int32_t WitRegisterCallBack(RegUpdateCb update_func)

函数名称	WitRegisterCallBack
函数原型	int32_t WitRegisterCallBack(RegUpdateCb update_func)
函数功能	注册返回的应答函数
输入参数	“update_func” 传入的数据是返回应答函数指针
返回值	WIT_HAL_INVALID: 无效的参数, 传入的函数指针可能为空 WIT_HAL_OK: 正确

2.7 int32_t WitWriteReg(uint32_t uiReg, uint16_t usData)

函数名称	WitWriteReg
函数原型	int32_t WitWriteReg(uint32_t uiReg, uint16_t usData)
函数功能	写寄存器
输入参数	“uiReg” 传入寄存器号 “usData” 写入寄存器的数据
返回值	WIT_HAL_INVALID: 无效的参数 WIT_HAL_OK: 正确

2.8 int32_t WitReadReg(uint32_t uiReg, uint32_t uiReadNum)

函数名称	WitReadReg
函数原型	int32_t WitReadReg(uint32_t uiReg, uint32_t uiReadNum)
函数功能	读寄存器
输入参数	“uiReg” 传入的寄存器号 “uiReadNum” 要写入寄存器的数据
返回值	WIT_HAL_INVALID: 无效的参数 WIT_HAL_OK: 正确

2.9 int32_t WitInit(uint32_t uiProtocol, uint8_t ucAddr)

函数名称	WitInit
函数原型	int32_t WitInit(uint32_t uiProtocol, uint8_t ucAddr)
函数功能	初始化函数
输入参数	“uiProtocol” 协议参数 “ucAddr” 地址参数，当协议参数为 normal 时，该参数无效
返回值	WIT_HAL_INVALID: 无效的参数 WIT_HAL_OK: 正确

2.10 void WitDeInit(void)

函数名称	WitDeInit
函数原型	void WitDeInit(void)
函数功能	去除初始化
输入参数	无
返回值	无

2.11 int32_t WitDelayMsRegister(DelaymsCb delaysms_func)

函数名称	WitDelayMsRegister
函数原型	int32_t WitDelayMsRegister(DelaymsCb delaysms_func)
函数功能	注册毫秒级延时函数
输入参数	“delaysms_func” 传入的数据是延时函数指针
返回值	WIT_HAL_INVALID: 无效的参数，传入的函数指针可能为空 WIT_HAL_OK: 正确

2.12 int32_t WitStartAccCali(void)

函数名称	WitStartAccCali
函数原型	int32_t WitStartAccCali(void)
函数功能	加速度开始校准
输入参数	无
返回值	WIT_HAL_ERROR: 有错误发生 WIT_HAL_OK: 正确

2.13 int32_t WitStopAccCali(void)

函数名称	WitStopAccCali
函数原型	int32_t WitStopAccCali(void)
函数功能	加速度停止校准
输入参数	无
返回值	WIT_HAL_ERROR: 有错误发生 WIT_HAL_OK: 正确

2.14 int32_t WitStartMagCali(void)

函数名称	WitStartMagCali
函数原型	int32_t WitStartMagCali(void)
函数功能	开始磁场校准
输入参数	无
返回值	WIT_HAL_ERROR: 有错误发生 WIT_HAL_OK: 正确

2.15 int32_t WitStopMagCali(void)

函数名称	WitStopMagCali
函数原型	int32_t WitStopMagCali(void)
函数功能	停止磁场校准
输入参数	无
返回值	WIT_HAL_ERROR: 有错误发生 WIT_HAL_OK: 正确

2.16 int32_t WitSetUartBaud(int32_t uiBaudIndex)

函数名称	WitSetUartBaud
函数原型	int32_t WitSetUartBaud(int32_t uiBaudIndex)
函数功能	设置串口波特率
输入参数	“uiBaudIndex” 需要设置波特率的值

返回值	WIT_HAL_INVALID: 无效的参数 WIT_HAL_ERROR: 有错误发生 WIT_HAL_OK: 正确
-----	--

2.17 int32_t WitSetCanBaud(int32_t uiBaudIndex)

函数名称	WitSetCanBaud
函数原型	int32_t WitSetCanBaud(int32_t uiBaudIndex)
函数功能	设置 Can 的波特率
输入参数	“uiBaudIndex”: 需要设置波特率的值
返回值	WIT_HAL_INVALID: 无效的参数 WIT_HAL_ERROR: 正确 WIT_HAL_OK: 没有错误

2.18 int32_t WitSetBandwidth(int32_t uiBaudWidth)

函数名称	WitSetBandwidth
函数原型	int32_t WitSetBandwidth(int32_t uiBaudWidth)
函数功能	设置带宽
输入参数	“uiBaudWidth”: 需要设置带宽的值
返回值	WIT_HAL_INVALID: 无效的参数 WIT_HAL_ERROR: 有错误发生 WIT_HAL_OK: 正确

2.19 int32_t WitSetOutputRate(int32_t uiRate)

函数名称	WitSetOutputRate
函数原型	int32_t WitSetOutputRate(int32_t uiRate)
函数功能	设置输出速率
输入参数	“uiRate”: 需要设置输出速率的值
返回值	WIT_HAL_INVALID: 无效的参数 WIT_HAL_ERROR: 有错误发生 WIT_HAL_OK: 正确

2.20 int32_t WitSetContent(int32_t uiRsw)

函数名称	WitSetContent
函数原型	int32_t WitSetContent(int32_t uiRsw)
函数功能	设置
输入参数	“uiRsw”: 需要设置的值
返回值	WIT_HAL_INVALID: 无效的参数 WIT_HAL_ERROR: 有错误发生 WIT_HAL_OK: 正确

3. 联系我们



深圳维特智能科技有限公司

WitMotion ShenZhen Co., Ltd

电话: 0755-33185882

邮箱: wit@wit-motion.com

网站: <http://www.wit-motion.cn>

店铺: <https://robotcontrol.taobao.com>

地址: 深圳市光明区西环大道 143 号光明云里智能园