// ad7292 head file

// ad7292.h

/\* 数据类型列表宏

\* ad7292\_HandleTypeDef结构的enum型datatype变量存储值

\*/

/\*…………datatype micro define……………..\*/

#define ADDRESS\_DATA16 0

#define ADDRESS\_SUBADDRESS\_DATA8 1

#define ADDRESS\_SUBADDRESS\_DATA16 2

/\*

\* 数据类型定义主要功能：需要根据主地址号确定使用的三类datatype数据缓存区大小

\* 功能说明：

\* 如VENDOR\_ID（0x00）主地址号，使用主地址+16位地址数据组织结构

\* 而CONFIG\_B（0x05）主地址号下存在子存储地址，使用主地址+子地址+16位地址数据组\* 织结构

\* 不同结构的数据需要通过SPI总线连续地从主控制器发送至ad7292

\* 如果从主控制器发送至ad7292的操作类型为读取，那么还需要在上述传输后紧随16位的\* SPI读回时序

\*/

/\*…………………ad7292\_dataTypeDef……………………\*/

typedef union {

struct RW\_Address{ // 8 bits RW and Address

read: 1;

write: 1;

address: 6;

};

uint16\_t data16; // fig 36

struct subAddress\_data8 {uint8\_t subAddress; uint8\_t data8}; // fig 37

struct subAddress\_data16 {uint8\_t subAddress; uint16\_t data16}; // fig 38

} ad7292\_dataTypeDef;

/\* ad7292实体主地址寄存器类型定义

\* 一个总长度为0x34的连续内存作为主地址寄存器内存映射

\* 无定义偏移地址使用main\_null\_reg X进行占位

\*/

typedef struct

{

\_\_IO uint16\_t VENDOR\_ID; // main register + offset 0x00

\_\_IO uint16\_t ADC\_DATA; // main register + offset 0x01

\_\_IO uint16\_t main\_null\_reg0; // main register + offset 0x02, non-define

\_\_IO uint16\_t ADC\_SEQUENCE; // main register + offset 0x03

\_\_IO uint16\_t main\_null\_reg1; // main register + offset 0x04, non-define

\_\_IO uint16\_t CONFIG\_B; // main register + offset 0x05

\_\_IO uint16\_t ALERT\_LIMITS\_B; // main register + offset 0x06

\_\_IO uint16\_t ALERT\_FLAGS\_B; // main register + offset 0x07

\_\_IO uint16\_t MINI\_MAX\_B; // main register + offset 0x08

\_\_IO uint16\_t OFFSET\_B; // main register + offset 0x09

\_\_IO uint16\_t DAC\_BUFFER\_EN; // main register + offset 0x0A

\_\_IO uint16\_t GPIO; // main register + offset 0x0B

\_\_IO uint16\_t CONV\_COMMAND; // main register + offset 0x0C

\_\_IO uint16\_t ADC\_CONV\_RESULT0; // main register + offset 0x0D

\_\_IO uint16\_t ADC\_CONV\_RESULT1; // main register + offset 0x0E

\_\_IO uint16\_t ADC\_CONV\_RESULT2; // main register + offset 0x0F

\_\_IO uint16\_t ADC\_CONV\_RESULT3; // main register + offset 0x10

\_\_IO uint16\_t ADC\_CONV\_RESULT4; // main register + offset 0x11

\_\_IO uint16\_t ADC\_CONV\_RESULT5; // main register + offset 0x12

\_\_IO uint16\_t ADC\_CONV\_RESULT6; // main register + offset 0x13

\_\_IO uint16\_t ADC\_CONV\_RESULT7; // main register + offset 0x14

\_\_IO uint16\_t main\_null\_reg2[12]; // main register + offset 0x15-0x1F, non-define

\_\_IO uint16\_t Tsense\_CONV\_RESULT; // main register + offset 0x20

\_\_IO uint16\_t main\_null\_reg3[16]; // main register + offset 0x21-0x2F, non-define

\_\_IO uint16\_t DAC\_CHANNEL0; // main register + offset 0x30

\_\_IO uint16\_t DAC\_CHANNEL1; // main register + offset 0x31

\_\_IO uint16\_t DAC\_CHANNEL2; // main register + offset 0x32

\_\_IO uint16\_t DAC\_CHANNEL3; // main register + offset 0x33

} AD7292\_mainAddress\_TypeDef;

/\* ad7292实体子地址寄存器类型定义

\* 一个总长度为0x32的连续内存作为CONFIG bank (0x05)的子地址寄存器内存映射

\* 无定义偏移地址使用sub5\_null\_reg X进行占位

\*/

typedef struct

{

\_\_IO uint16\_t sub5\_null\_reg0; //sub register + offset 0x01, non-define

\_\_IO uint16\_t DIGITAL\_OUT\_DRIVER; //sub register of CONFIG\_B bank + offset 0x01

\_\_IO uint16\_t DIGITAL\_IO\_FUNC; //sub register of CONFIG\_B bank + offset 0x02

\_\_IO uint16\_t sub5\_null\_reg1[5]; //sub register + offset 0x03-07, non-define

\_\_IO uint16\_t GERAL; //sub register of CONFIG\_B bank + offset 0x08

\_\_IO uint16\_t sub5\_null\_reg2[7]; // sub register + offset 0x09-0F, non-define

\_\_IO uint16\_t VIN\_RANGE0; //sub register of CONFIG\_B bank + offset 0x10

\_\_IO uint16\_t VIN\_RANGE1; //sub register of CONFIG\_B bank + offset 0x11

\_\_IO uint16\_t ADC\_SAMPLE\_MODEL; //sub register of CONFIG\_B bank + offset 0x12

\_\_IO uint16\_t VIN\_ALERT0\_ROUTE; //sub register of CONFIG\_B bank + offset 0x13

\_\_IO uint16\_t VIN\_ALERT1\_ROUTE; //sub register of CONFIG\_B bank + offset 0x14

\_\_IO uint16\_t VIN\_FILTER; //sub register of CONFIG\_B bank + offset 0x15

\_\_IO uint16\_t CONV\_DELAY; //sub register of CONFIG\_B bank + offset 0x16

\_\_IO uint16\_t sub5\_null\_reg2[8]; // sub register + offset 0x17-1F, non-define

\_\_IO uint16\_t Tsense\_CONFIG; //sub register of CONFIG\_B bank + offset 0x20

\_\_IO uint16\_t Tsense\_ALERT\_ROUTE; //sub register of CONFIG\_B bank + offset 0x21

\_\_IO uint16\_t sub5\_null\_reg3[13]; // sub register + offset 0x22-2F, non-define

\_\_IO uint16\_t GPIO2\_DAC\_DISABLE0; //sub register of CONFIG\_B bank + offset 0x30

\_\_IO uint16\_t GPIO2\_DAC\_DISABLE1; //sub register of CONFIG\_B bank + offset 0x31

} subAddress\_CONFIG\_B\_TypeDef;

/\* ad7292实体子地址寄存器类型定义

\* 一个总长度为0x的连续内存作为ALERT\_LIMITS\_B (0x06)的子地址寄存器内存映射

\* 无定义偏移地址使用sub6\_null\_reg X进行占位

\*/

typedef struct

{

} subAddress\_ALERT\_LIMITS\_B\_TypeDef;

/\* ad7292实体子地址寄存器类型定义

\* 一个总长度为0x的连续内存作为ALERT\_FLAGS\_B (0x07)的子地址寄存器内存映射

\* 无定义偏移地址使用sub7\_null\_reg X进行占位

\*/

typedef struct

{

} subAddress\_ ALERT\_FLAGS\_B\_TypeDef;

/\* ad7292实体子地址寄存器类型定义

\* 一个总长度为0x的连续内存作为MINI\_MAX\_B (0x08)的子地址寄存器内存映射

\* 无定义偏移地址使用sub8\_null\_reg X进行占位

\*/

typedef struct

{

} subAddress\_ MINI\_MAX\_B\_TypeDef;

/\* ad7292实体子地址寄存器类型定义

\* 一个总长度为0x的连续内存作为OFFSET\_B (0x09)的子地址寄存器内存映射

\* 无定义偏移地址使用sub9\_null\_reg X进行占位

\*/

typedef struct

{

} subAddress\_OFFSET\_B \_TypeDef;

/\* ad7292实体地址寄存器类型定义

\* 将上述主地址和子地址组结构体整合在一起形成描述AD7292的地址（寄存器）内存映射

\* 因为每类地址（寄存器）已按序组织，每个组内特定寄存器可按名字宏偏移检索到

\* 如mainAddr组内偏移为0x05的CONFIG\_B可以通过ad7292\_TypeDef基地址+0x05

\* 检索到

\* ad7292\_TypeDef基地址通过 AD7292Config.h定义 #define AD7292 0x1000(RAM区内)

\* 0x05通过宏定义进行 #define CONFIG\_B 0x05 + AD7292

\* 子寄存器组基地址相对ad7292\_TypeDef基地址偏移为#define CONFIGBANK AD7292 +

\* sizeof(AD7292\_mainAddress\_TypeDef)

\* 如subAddrCONFIG组内偏移为0x10的VIN\_RANGE0可以通过子寄存器组基地址+

\* 0x10访问到

\* subAddrCONFIG组基地址为 #define

\* 0x10偏移通过宏定义进行 #define VIN\_RANGE0 0x10 + CONFIGBANK

\*

\*/

typedef struct

{

AD7292\_mainAddress\_TypeDef mainAddr;

subAddress\_CONFIG\_B\_TypeDef subAddrCONFIG;

subAddress\_ALERT\_LIMITS\_B\_TypeDef subAddrALTLIM;

subAddress\_ ALERT\_FLAGS\_B\_TypeDef subAddrALTFLG;

subAddress\_ MINI\_MAX\_B\_TypeDef subAddrMINMAX;

subAddress\_OFFSET\_B \_TypeDef subAddrOFST;

} ad7292\_TypeDef;

/\*……………..…ad7292\_HandleTypeDef………………..\*/

typedef struct \_ad7292\_ HandleTypeDef {

ad7292\_TypeDef \*Instance; // register memory map，通过用户指定地址

// 作为AD7292基地址，也是mainAddr基地

// 址

ad7292\_dataTypeDef data; // maxium 32bits data transaction with spi

enum datatype; // datatype indicator,

uint8\_t length; // data length

SPI\_HandleTypeDef \*hspi; // handle for stm32 spi

} ad7292\_HandleTypeDef;

/\*……………..…ad7292\_ HandleTypeDef………………..\*/

// ad7292 implement file

// ad7292.c

/\*……………main…Address……………………\*/

enum ADDRESS {

VENDOR\_ID = 0x00,

ADC\_DATA = 0x01,

ADC\_SEQUENCE = 0x03,

CONFIG\_B = 0x05,

ALERT\_LIMITS\_B ,

ALERT\_FLAGS\_B ,

MINI\_MAX\_B ,

OFFSET\_B ,

DAC\_BUFFER\_EN ,

GPIO ,

CONV\_COMMAND ,

ADC\_CONV\_RESULT0 ,

ADC\_CONV\_RESULT1 ,

ADC\_CONV\_RESULT2 ,

ADC\_CONV\_RESULT3 ,

ADC\_CONV\_RESULT4 ,

ADC\_CONV\_RESULT5 ,

ADC\_CONV\_RESULT6 ,

ADC\_CONV\_RESULT7 ,

Tsense\_CONV\_RESULT = 0x20,

DAC\_CHANNEL0 = 0x30,

DAC\_CHANNEL1 ,

DAC\_CHANNEL2 ,

DAC\_CHANNEL3 ,

};

/\*……………sub\_Address using by Address group 0x05 CONFIG\_B…………\*/

enum SUB\_ADDRESS\_ CONFIG\_B {

DIGITAL\_OUT\_DRIVER = 0x01,

DIGITAL\_IO\_FUNC = 0x02,

GERAL = 0x08,

VIN\_RANGE0 = 0x10,

VIN\_RANGE1 = 0x11,

ADC\_SAMPLE\_MODEL = 0x12,

VIN\_ALERT0\_ROUTE = 0x13,

VIN\_ALERT1\_ROUTE = 0x14,

VIN\_FILTER = 0x15,

CONV\_DELAY = 0x16,

Tsense\_CONFIG = 0x20,

Tsense\_ALERT\_ROUTE = 0x21,

GPIO2\_DAC\_DISABLE0 = 0x30,

GPIO2\_DAC\_DISABLE1 = 0x31,

};

/\*……………sub\_Address using by Address group 0x06 ALERT\_LIMITS\_B…………\*/

enum SUB\_ADDRESS\_ ALERT\_LIMITS\_B {

VIN0\_MAX, VIN0\_MIN, VIN0\_DELAY,

VIN1\_MAX, VIN1\_MIN, VIN1\_DELAY,

VIN2\_MAX, VIN2\_MIN, VIN2\_DELAY,

VIN3\_MAX, VIN3\_MIN, VIN3\_DELAY,

VIN4\_MAX, VIN4\_MIN, VIN4\_DELAY,

VIN5\_MAX, VIN5\_MIN, VIN5\_DELAY,

VIN6\_MAX, VIN6\_MIN, VIN6\_DELAY,

VIN7\_MAX, VIN7\_MIN, VIN7\_DELAY,

Tsense\_MAX = 0x30, Tsense\_MIN = 0x31, Tsense\_DELAY = 0x32

};

/\*……………sub\_Address using by Address group 0x07 ALERT\_FLAGS\_B…………\*/

enum SUB\_ADDRESS\_ ALERT\_ FLAGS\_B {

ADC\_ALERT\_FLAG, Tsense\_FLAG = 0x02

};

/\*……………sub\_Address using by Address group 0x08 MINI\_MAX\_B…………\*/

enum SUB\_ADDRESS\_ MINI\_MAX\_B {

VIN0\_MAX, VIN0\_MIN,

VIN1\_MAX, VIN1\_MIN,

VIN2\_MAX, VIN2\_MIN,

VIN3\_MAX, VIN3\_MIN,

VIN4\_MAX, VIN4\_MIN,

VIN5\_MAX, VIN5\_MIN,

VIN6\_MAX, VIN6\_MIN,

VIN7\_MAX, VIN7\_MIN,

Tsense\_MAX = 0x20, Tsense\_MIN = 0x21

};