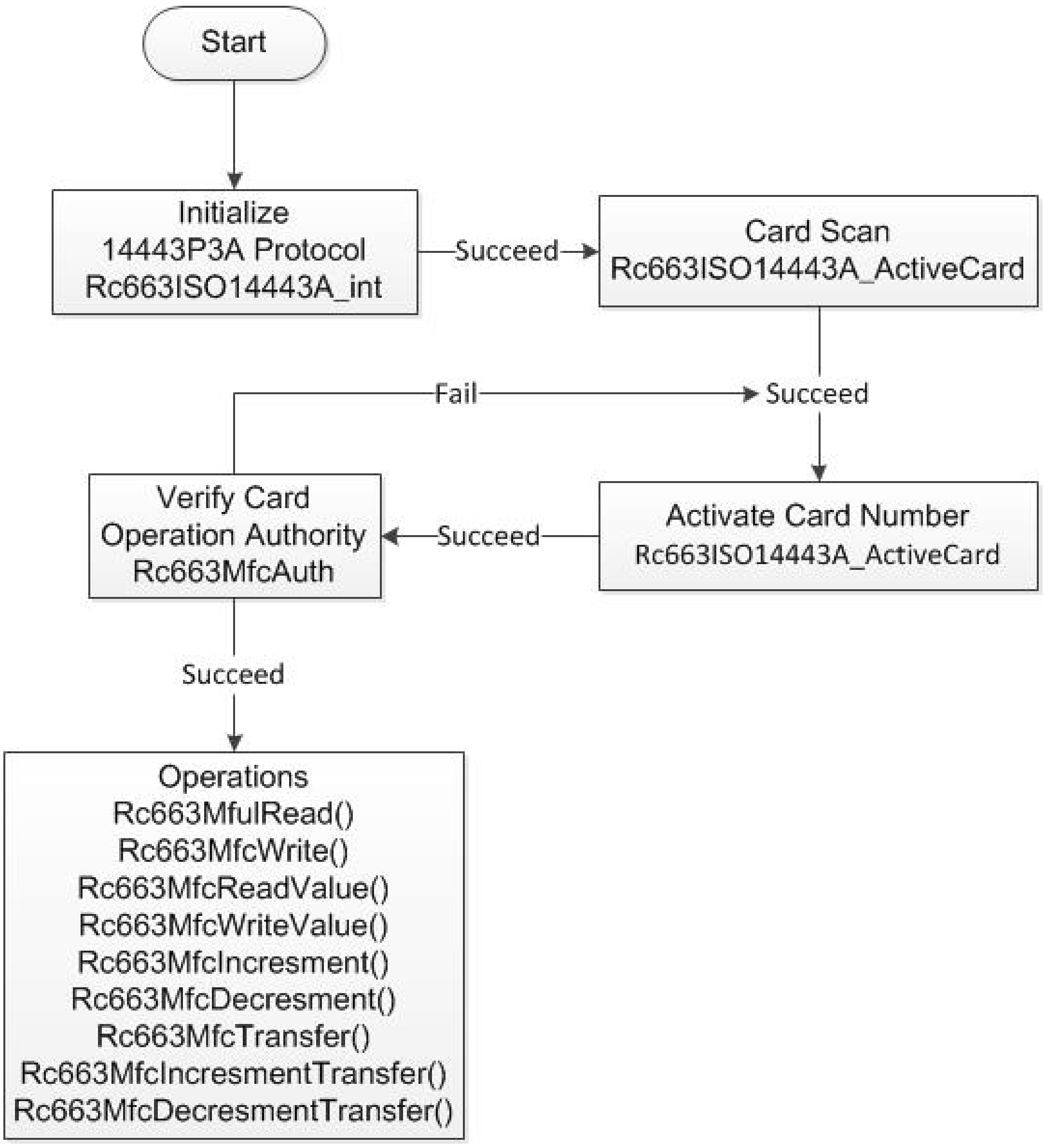
Document reversion V1.0 2014-10-20

TT43 Data Acquisition Terminal High Frequency RFID Function Interface User Manual

SPEEDATA 思必拓
北京卓享智和科技有限公司
您的移动物联网伙伴
Your Mobile Computing Partne

Yuhao Tian
2014/10/20

14443P3A Protocol Function Interface Introduction



BOOLRc663ISO14443P3A_Init();

Function Name	Rc663ISO14443P3A_Init
Function	Initialize the ISO14443A card hardware, software and protocol
Parameters	None
Return Values	TRUE(1) operation successful, FALSE(0) operation failed

*intRc663ISO14443P3A_ActiveCard(BYTE*Uidin,uint8_tUidinLen,BYTE*Uid,UINT8**

*UidLen,BYTE*SAK,BYTE*MoreCardsAvialable);*

Function Name	*Rc663ISO14443P3A_ActiveCard*
Function	Scan and activate current ISO14443A card
Parameters	*[in]Uidin* ID of the card needs to be activated. Can be null when no card selected. *[in]UidinLen* Length of ID of the card needs to be activated. *[out]Uid* ID of the scanned card. *[out]UidLen* The length of ID of the scanned card. *[out]SAK* Type of scanned card. *[out]MoreCardsAvialable* More card or not.(>0 more card, <=0 no more card)
Return Values	*0* when operation successful, error code when operation failed

intRc663MfulAuth(UINT16KeyNumber,UINT16KeyVersion);

Function Name	Rc663MfulAuth
Function	Ensure operation authority of MIFARE(R)Ultralight-C card
Parameters	*[in]KeyNumber* Verify the password *[in]KeyVersion* Verify the version number
Return Values	*0x00* or *0x07* when operation successful, error code when operation failed

*intRc663MfulRead(BYTEAdd,BYTE*pData);*

Function Name	Rc663MfulRead
Function	Reading command of MIFARE(R)Ultralight-C card
Parameters	*[in]Add* Read address *[out]pData* The read data
Return Values	*0* when operation successful, error code when operation failed

*intRc663MfcAuth(BYTEBlock,BYTEKeyType,BYTE*Key,BYTE*Uid,BYTEUidLen);*

Function Name	Rc663MfcAuth
Function	Verification command of MIFARE(R) card
Parameters	*[in]Block* Verify address *[in]KeyType* Verify card type. A card: KeyType= 0x0A; B card: KeyType= 0x0B. *[in]Key* Verify pass code *[in]Uid* Verify UID of the card *[in]UidLen* Verify the length of card UID
Return Values	*0* when operation successful, error code when operation failed

*intRc663MfulCompatibilityWrite(BYTEAdd,BYTE*pData);*

Function Name	Rc663MfulCompatibilityWrite
Function	Compatible writing command of MIFARE(R)Ultralight-C card
Parameters	*[in]Add* Write address *[in]pData* The written data
Return Values	*0* when operation successful, error code when operation failed

*intRc663MfulWrite(BYTEAdd,BYTE*pData);*

Function Name	Rc663MfulWrite
Function	Writing command of MIFARE(R)Ultralight-C card
Parameters	*[in]Add* Write address *[in]pData* The written data
Return Values	*0* when operation successful, error code when operation failed

*intRc663MfcRead(BYTEBlock,BYTE*pBlockData);*

Function Name	Rc663MfcRead
Function	Reading command of MIFARE(R) card
Parameters	*[in]Block* Read address *[out]pBlockData* The read data
Return Values	*0* when operation successful, error code when operation failed

*intRc663MfcWrite(BYTEBlock,BYTE*pBlockData);*

Function Name	Rc663MfcWrite
Function	Writing command of MIFARE(R) card
Parameters	*[in]Block* Write address *[in]pBlockData* The written data
Return Values	*0* when operation successful, error code when operation failed

*intRc663MfcReadValue(BYTEBlock,BYTE*pValue,BYTE*pAddrData);*

Function Name	Rc663MfcReadValue
Function	E-wallet reading command of MIFARE(R) card
Parameters	*[in]Block* Read address *[out]pValue* The read value *[out]pAddrData* Return address of the read data
Return Values	*0* when operation successful, error code when operation failed

*intRc663MfcWriteValue(BYTEBlock,BYTE*pValue,BYTEpAddrData);*

Function Name	Rc663MfcWriteValue
Function	E-wallet writing command of MIFARE(R) card
Parameters	*[in]Block* Write address *[out]pValue* The written value *[out]pAddrData* Return address of the written data
Return Values	*0* when operation successful, error code when operation failed

*intRc663MfcIncresment(BYTEBlock,BYTE*pValue);*

Function Name	Rc663MfcIncresment
Function	E-wallet increase command to add pValue to target address value of MIFARE(R) card.
Parameters	*[in]Block* Operation address *[in]pValue* Increase value
Return Values	*0* when operation successful, error code when operation failed

*intRc663MfcDecresment(BYTEBlock,BYTE*pValue);*

Function Name	*Rc663MfcDecresment*
Function	E-wallet decrease command to minus pValue from target address value of MIFARE(R) card.
Parameters	*[in]Block* Operation address *[in]pValue* Decrease value
Return Values	*0* when operation successful, error code when operation failed

Note: In order to save the operation results of Rc663MfcIncresment and Rc663MfcDecresment, the results must be written into specified addresses using Rc663MfcTransfer command, otherwise the saving will be invalid.

intRc663MfcTransfer(BYTEBlock);

Function Name	Rc663MfcTransfer
Function	E-wallet transfer command of MIFARE(R) card Note: This function is the only way to write the operation results of Rc663MfcIncresment and Rc663MfcDecresment function into the Block parameter specified address.
Parameters	*[in]Block* Operation address
Return Values	*0* when operation successful, error code when operation failed

*intRc663MfcIncresmentTransfer(BYTEpSrcBlock,BYTEpDstBlock,BYTE*pValue);*

Function Name	Rc663MfcIncresmentTransfer

Function	E-wallet increase and transfer command of MIFARE(R) card Note: The command can be executed by verify any of the two address
Parameters	*[in]pSrcBlock* Value address *[in]pDstBlock* Target address *[in]pValue* Increase value
Return Values	*0* when operation successful, error code when operation failed

intRc663MfcDecresmentTransfer(BYTEpSrcBlock,BYTEpDstBlock,BYTE*pValue);

Function Name	*Rc663MfcDecresmentTransfer*
Function	E-wallet decrease and transfer command of MIFARE(R) card Note: The command can be executed by verify any of the two address
Parameters	*[in]pSrcBlock* Value address *[in]pDstBlock* Target address *[in]pValue* Decrease value
Return Values	*0* when operation successful, error code when operation failed

intRc663MfcRestore(BYTEBlock);

Function Name	Rc663MfcRestore
Function	E-wallet recovery command of MIFARE(R) card Note: When Rc663MfcTransfer recovery value is not being used.
Parameters	*[in]Block* Recovery address
Return Values	*0* when operation successful, error code when operation failed

intRc663MfcRestoreTransfer(BYTEbSrcBlockNo,BYTEbDstBlockNo);

Function Name	Rc663MfcRestore
Function	E-wallet recovery command of MIFARE(R) card Note: When Rc663MfcTransfer recovery value is not being used.
Parameters	*[in]SrcBlockNo* Source address *[in]bDstBlockNo* Target address
Return Values	*0* when operation successful, error code when operation failed

intRc663MfcPersonalizeUid(BYTEbUIDType);

Function Name	Rc663MfcPersonalizeUid
Function	Personalized UID command

Parameters	*[in]bUIDType* UID type
	PHAL_MFC_UID_TYPE_UIDF0 = 0x00
	PHAL_MFC_UID_TYPE_UIDF1 = 0x40
	PHAL_MFC_UID_TYPE_UIDF2 = 0x20
	PHAL_MFC_UID_TYPE_UIDF3 = 0x60

Return Values	*0* when operation successful, error code when operation failed

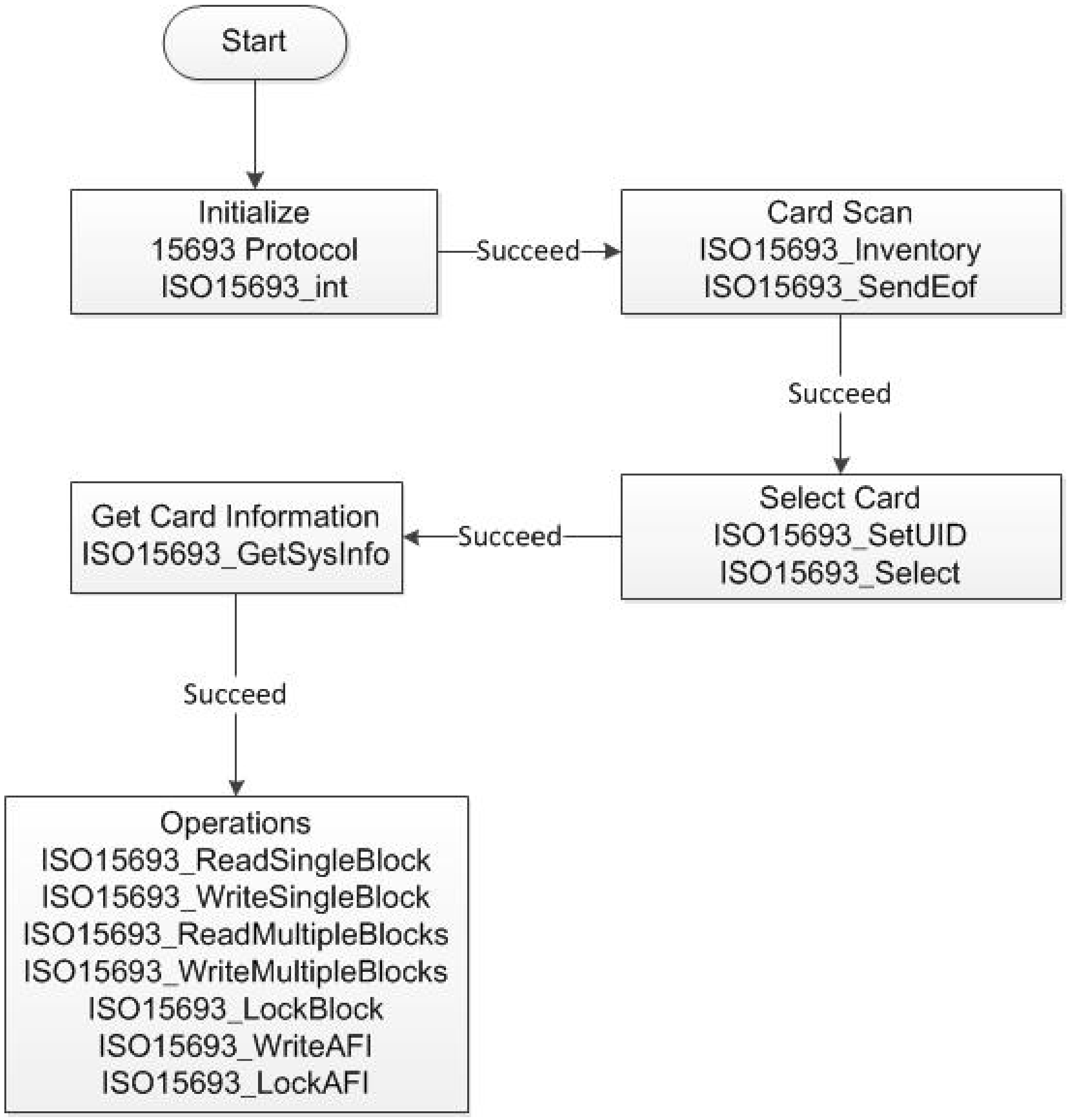
voidRc66314443P3A_Reset();

Function Name	Rc66314443P3A_Reset
Function	Device reset command
Parameters	None
Return Values	None

voidRc66314443P3A_Release();

Function Name	*Rc66314443P3A_Release*
Function	Device close command
Parameters	None
Return Values	None

ISO15693 Protocol Function Interface Introduction



BOOLISO15693_Init();

Function Name	SO15693_Init
Function	Initialize the ISO15693 card hardware, software and protocol
Parameters	None
Return Values	TRUE(1) operation successful, FALSE(0) operation failed

*intISO15693_GetSysInfo(byte*pRxBuffer,ushort*pRxLength);;*

Function Name	*ISO15693_GetSysInfo*
Function	Get selected card system information

Parameters	*[out]pRxBuffer* Scan and get selected card information *[out]pRxLength* Scan and get the length of selected card information
Return Values	*0* when operation successful, error code when operation failed

*intISO15693_Inventory(bytebFlags,bytebAfi,byte*pMask,bytebMaskBitLength,*

*byte*pDsfid,byte*pUid);*

Function Name	ISO15693_Inventory
Function	Set the scan card configuration information
Parameters	*[in]bFlags* Flag bit of card Default：0x20 (Use a time slot to scan) *[in]bAfi* Flag bit of application field Default: 0 (All field) *[in]pMask* Shielding value use to prevent collision *[in]bMaskBit Length* Length of shielding value *[out]pDsfid* Read Dsfid of the card *[out]pUid* Returned UID
Return Values	*0x00* when operation successful, error code when operation failed

*intISO15693_SendEof(bytebOption,byte*pDsfid,byte*pUid,byte*pUidLength,*
*byte*pData,ushort*pDataLength);*

Function	ISO15693_SendEof
Function	EOF flag sending command of ISO15693 card
Parameters	*[in]bOption* Operating setup code Default 0 *[out]pDsfid* The read Dsfid of the card *[out]pUid* The read UID of the card *[out]pUidLength* Length of the read UID *[out]pData* The read data *[out]pDataLength* Length of the read data
Return	*0* when operation successful, error code when operation failed

*intISO15693_SetUID(byte*pUid);*

Function Name	ISO15693_SetUID
Function	Set selected card UID
Parameters	*[in]pUid* UID of the card
Return Values	*0* when operation successful, error code when operation failed

intISO15693_Select();

Function Name	ISO15693_Select
Function	Select the set UID card
Parameters	None
Return Values	*0* when operation successful, error code when operation failed

*intISO15693_ReadSingleBlock(bytebOption,bytebBlockNo,byte*ppRxBuffer,*

*ushort*pRxLength);*

Function Name	*ISO15693_ReadSingleBlock*
Function	Read single block of ISO15963 card
Parameters	*[in]bOption* Operation code Default: 0 *[in]bBlockNo* The read block address *[out]ppRxBuffer* The read data *[out]pRxLength* Length of read data
Return Values	*0* when operation successful, error code when operation failed

*intISO15693_WriteSingleBlock(bytebOption,bytebBlockNo,byte*pTxBuffer,*

ushortwTxLength);

Function Name	ISO15693_WriteSingleBlock
Function	Write single block of ISO15963 card
Parameters	*[in]bOption* Operation code Default: 0 *[in]bBlockNo* The written block address *[out]ppTxBuffer* The written data *[out]pTxLength* Length of written data
Return Values	*0* when operation successful, error code when operation failed

intISO15693_ReadMultipleBlocks*(uint8_tbOption,uint8_tbBlockNo,uint16_t wNumBlocks,uint8_t* pRxBuffer,uint16_t* pRxLength);*

Function Name	ISO15693_ReadMultipleBlocks
Function	Read multiple blocks of IOS15693 card
Parameters	*[in]bOption* Operation code Default: 0 *[in]bBlockNo* Initial address of the read block *[in]wNumBlocks* Number of read blocks *[out]ppRxBuffer* The read data *[out]pRxLength* Length of read data
Return Values	*0* when operation successful, error code when operation failed

int*ISO15693_WriteMultipleBlocks(uint8_tbOption,uint8_tbBlockNo,uint16_t*

wNumBlocks,uint8_t pTxBuffer,uint16_twTxLength);*

Function Name	ISO15693_WriteMultipleBlocks
Function	Write multiple blocks of IOS15693 card
Parameters	*[in]bOption* Operation code Default: 0 *[in] bBlockNo* Initial address of the written block *[in]wNumBlocks* Number of written blocks *[out]ppTxBuffer* The written data *[out]pTxLength* Length of written data
Return Values	*0* when operation successful, error code when operation failed

int*ISO15693_LockBlock(uint8_tbOption,uint8_tbBlockNo);*

Function Name	ISO15693_LockBlock
Function	Lock data block operation
Parameters	*[in]bOption* Operation code *[in]bBlockNo* Lock block address
Return Values	*0* when operation successful, error code when operation failed

int*ISO15693_WriteAFI(uint8_tbOption,uint8_tbAfi);*

Function Name	ISO15693_WriteAFI
Function	Write application type of the card

Parameters	*[in]bOption* Operation code *[in]bAfi* Type value of application
Return Values	*0* when operation successful, error code when operation failed

intISO15693_LockAFI(uint8_tbOption);

Function Name	ISO15693_LockAFI
Function	AFI value of locked card
Parameters	*[in]bOption* Operation code
Return Values	*0* when operation successful, error code when operation failed