My Project

Generated by Doxygen 1.9.7

Chapter 1

Module Index

1.1 Modules

Here is a list of all modules:

Chapter 2

File Index

2.1 File List

Here is a list of all documented files with brief descriptions:

Chapter 3

Module Documentation

3.1 MFRC522 Register Addresses

Macros

- #define **MFRC522_REG_RESERVED00** 0x00 << 1
- #define **MFRC522_REG_COMMAND** 0x01 << 1
- #define **MFRC522_REG_COMIEN** 0x02 << 1
- #define **MFRC522_REG_DIVIEN** 0x03 << 1
- #define **MFRC522_REG_COMIRQ** 0x04 << 1
- #define **MFRC522_REG_DIVIRQ** 0x05 << 1
- #define **MFRC522_REG_ERROR** 0x06 << 1
- #define **MFRC522_REG_STATUS1** 0x07 << 1
- #define **MFRC522_REG_STATUS2** 0x08 << 1
- #define **MFRC522_REG_FIFO_DATA** 0x09 << 1
- #define **MFRC522_REG_FIFO_LEVEL** 0x0A << 1
- #define **MFRC522_REG_WATER_LEVEL** 0x0B << 1
- #define **MFRC522_REG_CONTROL** 0x0C << 1
- #define **MFRC522_REG_BIT_FRAMING** 0x0D << 1
- #define **MFRC522_REG_COLL** 0x0E << 1
- #define **MFRC522_REG_RESERVED01** 0x0F << 1
- #define **MFRC522_REG_RESERVED10** 0x10 << 1
- #define **MFRC522_REG_MODE** 0x11 << 1
- #define **MFRC522_REG_TX_MODE** 0x12 << 1
- #define **MFRC522_REG_RX_MODE** 0x13 << 1
- #define **MFRC522_REG_TX_CONTROL** 0x14 << 1
- #define **MFRC522_REG_TX_AUTO** 0x15 << 1
- #define **MFRC522_REG_TX_SEL** 0x16 << 1
- #define **MFRC522_REG_RX_SEL** 0x17 << 1
- #define **MFRC522_REG_RX_THRESHOLD** 0x18 << 1
- #define **MFRC522_REG_DEMOD** 0x19 << 1
- #define **MFRC522_REG_RESERVED11** 0x1A << 1
- #define **MFRC522_REG_RESERVED12** 0x1B << 1
- #define **MFRC522_REG_MIFARE** 0x1C << 1
- #define **MFRC522_REG_RESERVED13** 0x1D << 1
- #define **MFRC522_REG_RESERVED14** 0x1E << 1
- #define **MFRC522_REG_SERIALSPEED** 0x1F << 1
- #define **MFRC522_REG_RESERVED20** 0x20 << 1

- #define **MFRC522_REG_CRC_RESULT_M** 0x21 << 1
- #define **MFRC522_REG_CRC_RESULT_L** 0x22 << 1
- #define **MFRC522_REG_RESERVED21** 0x23 << 1
- #define **MFRC522_REG_MOD_WIDTH** 0x24 << 1
- #define **MFRC522_REG_RESERVED22** 0x25 << 1
- #define **MFRC522_REG_RF_CFG** 0x26 << 1
- #define **MFRC522_REG_GS_N** 0x27 << 1
- #define **MFRC522_REG_CWGS_PREG** 0x28 << 1
- #define **MFRC522_REG_MOD_GS_PREG** 0x29 << 1
- #define **MFRC522_REG_T_MODE** 0x2A << 1
- #define **MFRC522_REG_T_PRESCALER** 0x2B << 1
- #define **MFRC522_REG_T_RELOAD_H** 0x2C << 1
- #define **MFRC522_REG_T_RELOAD_L** 0x2D << 1
- #define **MFRC522_REG_T_COUNTER_VALUE_H** 0x2E << 1
- #define **MFRC522_REG_T_COUNTER_VALUE_L** 0x2F << 1
- #define **MFRC522_REG_RESERVED30** 0x30 << 1
- #define **MFRC522_REG_TEST_SEL1** 0x31 << 1
- #define **MFRC522_REG_TEST_SEL2** 0x32 << 1
- #define **MFRC522_REG_TEST_PIN_EN** 0x33 << 1
- #define **MFRC522_REG_TEST_PIN_VALUE** 0x34 << 1
- #define **MFRC522_REG_TEST_BUS** 0x35 << 1
- #define **MFRC522_REG_AUTO_TEST** 0x36 << 1
- #define **MFRC522_REG_VERSION** 0x37 << 1
- #define **MFRC522_REG_ANALOG_TEST** 0x38 << 1
- #define **MFRC522_REG_TEST_DAC1** 0x39 << 1
- #define **MFRC522_REG_TEST_DAC2** 0x3A << 1
- #define **MFRC522_REG_TEST_ADC** 0x3B << 1
- #define **MFRC522_REG_RESERVED31** 0x3C << 1
- #define **MFRC522_REG_RESERVED32** 0x3D << 1
- #define **MFRC522_REG_RESERVED33** 0x3E << 1
- #define **MFRC522_REG_RESERVED34** 0x3F << 1

3.1.1 Detailed Description

3.2 MFRC522 Commands

Macros

- #define **PICC_CMD_REQA** 0x26
- #define **PICC_CMD_WUPA** 0x52
- #define **PICC_CMD_CT** 0x88
- #define **PICC_CMD_SEL_CL1** 0x93
- #define **PICC_CMD_SEL_CL2** 0x95
- #define **PICC_CMD_SEL_CL3** 0x97
- #define **PICC_CMD_HLTA** 0x50
- #define **PICC_CMD_RATS** 0xE0
- #define **PICC_CMD_MF_AUTH_KEY_A** 0x60
- #define **PICC_CMD_MF_AUTH_KEY_B** 0x61
- #define **PICC_CMD_MF_READ** 0x30
- #define **PICC_CMD_MF_WRITE** 0xA0
- #define **PICC_CMD_MF_DECREMENT** 0xC0
- #define **PICC_CMD_MF_INCREMENT** 0xC1

- #define **PICC_CMD_MF_RESTORE** 0xC2
- #define **PICC_CMD_MF_TRANSFER** 0xB0
- #define **PICC_CMD_UL_WRITE** 0xA2
- #define **PCD_CMD_IDLE** 0x00
- #define **PCD_CMD_MEM** 0x01
- #define **PCD_CMD_GEN_RANDOM_ID** 0x02
- #define **PCD_CMD_CALC_CRC** 0x03
- #define **PCD_CMD_TRANSMIT** 0x04
- #define **PCD_CMD_NO_CMD_CHANGE** 0x07
- #define **PCD_CMD_RECEIVE** 0x08
- #define **PCD_CMD_TRANSCEIVE** 0x0C
- #define **PCD_CMD_MF_AUTHENT** 0x0E
- #define **PCD_CMD_SOFT_RESET** 0x0F

3.2.1 Detailed Description

Chapter 4

File Documentation

4.1 BLE_Controller.h

```
00001 /*****************************************************************************
00002 * File Name: ESP 32 BLE Driver
00003 * Author: Jaime Malone
00004 * File Description:
00005 *
00006 *
00007 *
00008 *****************************************************************************/
00009
00010 #include <stdio.h>
00011 #include <stdlib.h>
00012 #include <string.h>
00013 #include <inttypes.h>
00014 #include "freertos/FreeRTOS.h"
00015 #include "freertos/task.h"
00016 #include "freertos/event_groups.h"
00017 #include "esp_system.h"
00018 #include "esp_log.h"
00019 #include "nvs_flash.h"
00020 #include "esp_bt.h"
00021
00022 #include "esp_gap_ble_api.h"
00023 #include "esp_gatts_api.h"
00024 #include "esp_bt_defs.h"
00025 #include "esp_bt_main.h"
00026 #include "esp_gatt_common_api.h"
00027
00028 #include "sdkconfig.h"
00029
00040 void example_write_event_env(esp_gatt_if_t gatts_if, prepare_type_env_t *prepare_write_env,
      esp_ble_gatts_cb_param_t *param);
00041
00051 void example_exec_write_event_env(prepare_type_env_t *prepare_write_env, esp_ble_gatts_cb_param_t
      *param);
00052
00061 static void gap_event_handler(esp_gap_ble_cb_event_t event, esp_ble_gap_cb_param_t *param);
00062
00073 void example_write_event_env(esp_gatt_if_t gatts_if, prepare_type_env_t *prepare_write_env,
      esp_ble_gatts_cb_param_t *param);
00074
00084 void example_exec_write_event_env(prepare_type_env_t *prepare_write_env, esp_ble_gatts_cb_param_t
      *param);
00085
00096 static void gatts_profile_a_event_handler(esp_gatts_cb_event_t event, esp_gatt_if_t gatts_if,
      esp_ble_gatts_cb_param_t *param);
00097
00108 static void gatts_event_handler(esp_gatts_cb_event_t event, esp_gatt_if_t gatts_if,
      esp_ble_gatts_cb_param_t *param);
00109
00115 void BLE_init(void);
```

4.2 Sources/drivers/MFRC522.h File Reference

```
#include ¨driver/spi_master.h¨
```

```
#include ¨esp_timer.h¨
```

Macros

- #define **MFRC522_REG_RESERVED00** 0x00 << 1
- #define **MFRC522_REG_COMMAND** 0x01 << 1
- #define **MFRC522_REG_COMIEN** 0x02 << 1
- #define **MFRC522_REG_DIVIEN** 0x03 << 1
- #define **MFRC522_REG_COMIRQ** 0x04 << 1
- #define **MFRC522_REG_DIVIRQ** 0x05 << 1
- #define **MFRC522_REG_ERROR** 0x06 << 1
- #define **MFRC522_REG_STATUS1** 0x07 << 1
- #define **MFRC522_REG_STATUS2** 0x08 << 1
- #define **MFRC522_REG_FIFO_DATA** 0x09 << 1
- #define **MFRC522_REG_FIFO_LEVEL** 0x0A << 1
- #define **MFRC522_REG_WATER_LEVEL** 0x0B << 1
- #define **MFRC522_REG_CONTROL** 0x0C << 1
- #define **MFRC522_REG_BIT_FRAMING** 0x0D << 1
- #define **MFRC522_REG_COLL** 0x0E << 1
- #define **MFRC522_REG_RESERVED01** 0x0F << 1
- #define **MFRC522_REG_RESERVED10** 0x10 << 1
- #define **MFRC522_REG_MODE** 0x11 << 1
- #define **MFRC522_REG_TX_MODE** 0x12 << 1
- #define **MFRC522_REG_RX_MODE** 0x13 << 1
- #define **MFRC522_REG_TX_CONTROL** 0x14 << 1
- #define **MFRC522_REG_TX_AUTO** 0x15 << 1
- #define **MFRC522_REG_TX_SEL** 0x16 << 1
- #define **MFRC522_REG_RX_SEL** 0x17 << 1
- #define **MFRC522_REG_RX_THRESHOLD** 0x18 << 1
- #define **MFRC522_REG_DEMOD** 0x19 << 1
- #define **MFRC522_REG_RESERVED11** 0x1A << 1
- #define **MFRC522_REG_RESERVED12** 0x1B << 1
- #define **MFRC522_REG_MIFARE** 0x1C << 1
- #define **MFRC522_REG_RESERVED13** 0x1D << 1
- #define **MFRC522_REG_RESERVED14** 0x1E << 1
- #define **MFRC522_REG_SERIALSPEED** 0x1F << 1
- #define **MFRC522_REG_RESERVED20** 0x20 << 1
- #define **MFRC522_REG_CRC_RESULT_M** 0x21 << 1
- #define **MFRC522_REG_CRC_RESULT_L** 0x22 << 1
- #define **MFRC522_REG_RESERVED21** 0x23 << 1
- #define **MFRC522_REG_MOD_WIDTH** 0x24 << 1
- #define **MFRC522_REG_RESERVED22** 0x25 << 1
- #define **MFRC522_REG_RF_CFG** 0x26 << 1
- #define **MFRC522_REG_GS_N** 0x27 << 1
- #define **MFRC522_REG_CWGS_PREG** 0x28 << 1
- #define **MFRC522_REG_MOD_GS_PREG** 0x29 << 1
- #define **MFRC522_REG_T_MODE** 0x2A << 1
- #define **MFRC522_REG_T_PRESCALER** 0x2B << 1
- #define **MFRC522_REG_T_RELOAD_H** 0x2C << 1
- #define **MFRC522_REG_T_RELOAD_L** 0x2D << 1
- #define **MFRC522_REG_T_COUNTER_VALUE_H** 0x2E << 1
- #define **MFRC522_REG_T_COUNTER_VALUE_L** 0x2F << 1
- #define **MFRC522_REG_RESERVED30** 0x30 << 1

- #define **MFRC522_REG_TEST_SEL1** 0x31 << 1
- #define **MFRC522_REG_TEST_SEL2** 0x32 << 1
- #define **MFRC522_REG_TEST_PIN_EN** 0x33 << 1
- #define **MFRC522_REG_TEST_PIN_VALUE** 0x34 << 1
- #define **MFRC522_REG_TEST_BUS** 0x35 << 1
- #define **MFRC522_REG_AUTO_TEST** 0x36 << 1
- #define **MFRC522_REG_VERSION** 0x37 << 1
- #define **MFRC522_REG_ANALOG_TEST** 0x38 << 1
- #define **MFRC522_REG_TEST_DAC1** 0x39 << 1
- #define **MFRC522_REG_TEST_DAC2** 0x3A << 1
- #define **MFRC522_REG_TEST_ADC** 0x3B << 1
- #define **MFRC522_REG_RESERVED31** 0x3C << 1
- #define **MFRC522_REG_RESERVED32** 0x3D << 1
- #define **MFRC522_REG_RESERVED33** 0x3E << 1
- #define **MFRC522_REG_RESERVED34** 0x3F << 1
- #define **PICC_CMD_REQA** 0x26
- #define **PICC_CMD_WUPA** 0x52
- #define **PICC_CMD_CT** 0x88
- #define **PICC_CMD_SEL_CL1** 0x93
- #define **PICC_CMD_SEL_CL2** 0x95
- #define **PICC_CMD_SEL_CL3** 0x97
- #define **PICC_CMD_HLTA** 0x50
- #define **PICC_CMD_RATS** 0xE0
- #define **PICC_CMD_MF_AUTH_KEY_A** 0x60
- #define **PICC_CMD_MF_AUTH_KEY_B** 0x61
- #define **PICC_CMD_MF_READ** 0x30
- #define **PICC_CMD_MF_WRITE** 0xA0
- #define **PICC_CMD_MF_DECREMENT** 0xC0
- #define **PICC_CMD_MF_INCREMENT** 0xC1
- #define **PICC_CMD_MF_RESTORE** 0xC2
- #define **PICC_CMD_MF_TRANSFER** 0xB0
- #define **PICC_CMD_UL_WRITE** 0xA2
- #define **PCD_CMD_IDLE** 0x00
- #define **PCD_CMD_MEM** 0x01
- #define **PCD_CMD_GEN_RANDOM_ID** 0x02
- #define **PCD_CMD_CALC_CRC** 0x03
- #define **PCD_CMD_TRANSMIT** 0x04
- #define **PCD_CMD_NO_CMD_CHANGE** 0x07
- #define **PCD_CMD_RECEIVE** 0x08
- #define **PCD_CMD_TRANSCEIVE** 0x0C
- #define **PCD_CMD_MF_AUTHENT** 0x0E
- #define **PCD_CMD_SOFT_RESET** 0x0F

Functions

- esp_err_t MFRC522_WriteRegister (spi_device_handle_t *spiHandle, uint8_t registerAddress, uint8_t value)

  Writes a value to the specified register in the MFRC522.
- esp_err_t MFRC522_ReadRegister (spi_device_handle_t *spiHandle, uint8_t registerAddress, uint8_t *data)

  Reads the value from the specified register in the MFRC522.
- esp_err_t MFRC522_ReadRegisterArr (spi_device_handle_t *spiHandle, uint8_t registerAddress, uint8_t *dataArr, uint8_t dataSize)

  Reads multiple consecutive registers in the MFRC522.

- esp_err_t MFRC522_WriteRegisterArr (spi_device_handle_t *spiHandle, uint8_t registerAddress, uint8_t *dataArr, uint8_t dataSize)

  Writes multiple consecutive registers in the MFRC522.
- esp_err_t MFRC522_Init (spi_device_handle_t *spiHandle)

  Initializes the MFRC522 RFID module.
- esp_err_t MFRC522_ClrRegBitMask (spi_device_handle_t *spiHandle, uint8_t registerAdress, uint8_t mask)

  Clears the specified bits in the register of the MFRC522.
- bool MFRC522_IsCardPresent (spi_device_handle_t *spiHandle)

  Checks if a card is present.
- esp_err_t MFRC522_AntennaOn (spi_device_handle_t *spiHandle)

  Turns on the antenna of the MFRC522.
- esp_err_t MFRC522_SelfTest (spi_device_handle_t *spiHandle)

  Performs a self-test of the MFRC522.
- esp_err_t MFRC522_Reset (spi_device_handle_t *spiHandle)

  Resets the MFRC522.
- esp_err_t MFRC522_SendPICCcmdTranscieve (spi_device_handle_t *spiHandle, uint8_t piccCmd)

  Sends a PICC command and receives the response from the MFRC522.

4.2.1 Function Documentation

4.2.1.1 MFRC522_AntennaOn()

```
esp_err_t MFRC522_AntennaOn (
            spi_device_handle_t * spiHandle )
```

Turns on the antenna of the MFRC522.

Parameters

spiHandle	Pointer to the SPI device handle.

Returns

ESP_OK if successful, otherwise an error code.

4.2.1.2 MFRC522_ClrRegBitMask()

```
esp_err_t MFRC522_ClrRegBitMask (
            spi_device_handle_t * spiHandle,
            uint8_t registerAdress,
            uint8_t mask )
```

Clears the specified bits in the register of the MFRC522.

Parameters

spiHandle	Pointer to the SPI device handle.
registerAdress	The address of the register to modify.
mask	The bitmask of the bits to clear.

Returns

ESP_OK if successful, otherwise an error code.

4.2.1.3 MFRC522_Init()

```
esp_err_t MFRC522_Init (
            spi_device_handle_t * spiHandle )
```

Initializes the MFRC522 RFID module.

Parameters

spiHandle	Pointer to the SPI device handle.

Returns

ESP_OK if successful, otherwise an error code.

4.2.1.4 MFRC522_IsCardPresent()

```
bool MFRC522_IsCardPresent (
            spi_device_handle_t * spiHandle )
```

Checks if a card is present.

Parameters

spiHandle	Pointer to the SPI device handle.

Returns

True if a card is present, false otherwise.

4.2.1.5 MFRC522_ReadRegister()

```
esp_err_t MFRC522_ReadRegister (
            spi_device_handle_t * spiHandle,
            uint8_t registerAddress,
            uint8_t * data )
```

Reads the value from the specified register in the MFRC522.

Parameters

spiHandle	Pointer to the SPI device handle.
registerAddress	The address of the register to read.
data	Pointer to store the read value.

Returns

ESP_OK if successful, otherwise an error code.

4.2.1.6 MFRC522_ReadRegisterArr()

```
esp_err_t MFRC522_ReadRegisterArr (
            spi_device_handle_t * spiHandle,
            uint8_t registerAddress,
            uint8_t * dataArr,
            uint8_t dataSize )
```

Reads multiple consecutive registers in the MFRC522.

Parameters

spiHandle	Pointer to the SPI device handle.
registerAddress	The address of the first register to read.
dataArr	Pointer to the array to store the read values.
dataSize	The number of registers to read.

Returns

ESP_OK if successful, otherwise an error code.

4.2.1.7 MFRC522_Reset()

```
esp_err_t MFRC522_Reset (
            spi_device_handle_t * spiHandle )
```

Resets the MFRC522.

Parameters

spiHandle	Pointer to the SPI device handle.

Returns

ESP_OK if successful, otherwise an error code.

4.2.1.8 MFRC522_SelfTest()

```
esp_err_t MFRC522_SelfTest (
            spi_device_handle_t * spiHandle )
```

Performs a self-test of the MFRC522.

Parameters

spiHandle	Pointer to the SPI device handle.

Returns

ESP_OK if successful, otherwise an error code.

4.2.1.9 MFRC522_SendPICCcmdTranscieve()

```
esp_err_t MFRC522_SendPICCcmdTranscieve (
            spi_device_handle_t * spiHandle,
            uint8_t piccCmd )
```

Sends a PICC command and receives the response from the MFRC522.

Parameters

spiHandle	Pointer to the SPI device handle.
piccCmd	The PICC command to send.

Returns

ESP_OK if successful, otherwise an error code.

4.2.1.10 MFRC522_WriteRegister()

```
esp_err_t MFRC522_WriteRegister (
            spi_device_handle_t * spiHandle,
            uint8_t registerAddress,
            uint8_t value )
```

Writes a value to the specified register in the MFRC522.

Parameters

spiHandle	Pointer to the SPI device handle.
registerAddress	The address of the register to write.
value	The value to write to the register.

Returns

ESP_OK if successful, otherwise an error code.

4.2.1.11 MFRC522_WriteRegisterArr()

```
esp_err_t MFRC522_WriteRegisterArr (
            spi_device_handle_t * spiHandle,
```

```
            uint8_t registerAddress,
            uint8_t * dataArr,
            uint8_t dataSize )
```

Writes multiple consecutive registers in the MFRC522.

Parameters

spiHandle	Pointer to the SPI device handle.
registerAddress	The address of the first register to write.
dataArr	Pointer to the array of values to write.
dataSize	The number of registers to write.

Returns

ESP_OK if successful, otherwise an error code.

4.3 MFRC522.h

Go to the documentation of this file.

```
00001
00005 #ifndef _MFRC522_H_
00006 #define _MFRC522_H_
00007
00008 #include "driver/spi_master.h"
00009 #include "esp_timer.h"
00010
00015 #define MFRC522_REG_RESERVED00          0x00 << 1
00016 #define MFRC522_REG_COMMAND             0x01 << 1
00017 #define MFRC522_REG_COMIEN              0x02 << 1
00018 #define MFRC522_REG_DIVIEN              0x03 << 1
00019 #define MFRC522_REG_COMIRQ              0x04 << 1
00020 #define MFRC522_REG_DIVIRQ              0x05 << 1
00021 #define MFRC522_REG_ERROR               0x06 << 1
00022 #define MFRC522_REG_STATUS1             0x07 << 1
00023 #define MFRC522_REG_STATUS2             0x08 << 1
00024 #define MFRC522_REG_FIFO_DATA           0x09 << 1
00025 #define MFRC522_REG_FIFO_LEVEL          0x0A << 1
00026 #define MFRC522_REG_WATER_LEVEL         0x0B << 1
00027 #define MFRC522_REG_CONTROL             0x0C << 1
00028 #define MFRC522_REG_BIT_FRAMING         0x0D << 1
00029 #define MFRC522_REG_COLL                0x0E << 1
00030 #define MFRC522_REG_RESERVED01          0x0F << 1
00031
00032 // Page 1: Command
00033 #define MFRC522_REG_RESERVED10          0x10 << 1
00034 #define MFRC522_REG_MODE                0x11 << 1
00035 #define MFRC522_REG_TX_MODE             0x12 << 1
00036 #define MFRC522_REG_RX_MODE             0x13 << 1
00037 #define MFRC522_REG_TX_CONTROL          0x14 << 1
00038 #define MFRC522_REG_TX_AUTO             0x15 << 1
00039 #define MFRC522_REG_TX_SEL              0x16 << 1
00040 #define MFRC522_REG_RX_SEL              0x17 << 1
00041 #define MFRC522_REG_RX_THRESHOLD        0x18 << 1
00042 #define MFRC522_REG_DEMOD               0x19 << 1
00043 #define MFRC522_REG_RESERVED11          0x1A << 1
00044 #define MFRC522_REG_RESERVED12          0x1B << 1
00045 #define MFRC522_REG_MIFARE              0x1C << 1
00046 #define MFRC522_REG_RESERVED13          0x1D << 1
00047 #define MFRC522_REG_RESERVED14          0x1E << 1
00048 #define MFRC522_REG_SERIALSPEED         0x1F << 1
00049
00050 // Page 2: Configuration
00051 #define MFRC522_REG_RESERVED20          0x20 << 1
00052 #define MFRC522_REG_CRC_RESULT_M        0x21 << 1
00053 #define MFRC522_REG_CRC_RESULT_L        0x22 << 1
00054 #define MFRC522_REG_RESERVED21          0x23 << 1
00055 #define MFRC522_REG_MOD_WIDTH           0x24 << 1
00056 #define MFRC522_REG_RESERVED22          0x25 << 1
00057 #define MFRC522_REG_RF_CFG              0x26 << 1
00058 #define MFRC522_REG_GS_N                0x27 << 1
```

```
00059 #define MFRC522_REG_CWGS_PREG            0x28 << 1
00060 #define MFRC522_REG_MOD_GS_PREG          0x29 << 1
00061 #define MFRC522_REG_T_MODE               0x2A << 1
00062 #define MFRC522_REG_T_PRESCALER          0x2B << 1
00063 #define MFRC522_REG_T_RELOAD_H           0x2C << 1
00064 #define MFRC522_REG_T_RELOAD_L           0x2D << 1
00065 #define MFRC522_REG_T_COUNTER_VALUE_H    0x2E << 1
00066 #define MFRC522_REG_T_COUNTER_VALUE_L    0x2F << 1
00067
00068 // Page 3: Test
00069 #define MFRC522_REG_RESERVED30           0x30 << 1
00070 #define MFRC522_REG_TEST_SEL1            0x31 << 1
00071 #define MFRC522_REG_TEST_SEL2            0x32 << 1
00072 #define MFRC522_REG_TEST_PIN_EN          0x33 << 1
00073 #define MFRC522_REG_TEST_PIN_VALUE       0x34 << 1
00074 #define MFRC522_REG_TEST_BUS             0x35 << 1
00075 #define MFRC522_REG_AUTO_TEST            0x36 << 1
00076 #define MFRC522_REG_VERSION              0x37 << 1
00077 #define MFRC522_REG_ANALOG_TEST          0x38 << 1
00078 #define MFRC522_REG_TEST_DAC1            0x39 << 1
00079 #define MFRC522_REG_TEST_DAC2            0x3A << 1
00080 #define MFRC522_REG_TEST_ADC             0x3B << 1
00081 #define MFRC522_REG_RESERVED31           0x3C << 1
00082 #define MFRC522_REG_RESERVED32           0x3D << 1
00083 #define MFRC522_REG_RESERVED33           0x3E << 1
00084 #define MFRC522_REG_RESERVED34           0x3F << 1
00091 // Commands sent to the PICC.
00092 #define PICC_CMD_REQA         0x26
00093 #define PICC_CMD_WUPA         0x52
00094 #define PICC_CMD_CT           0x88
00095 #define PICC_CMD_SEL_CL1      0x93
00096 #define PICC_CMD_SEL_CL2      0x95
00097 #define PICC_CMD_SEL_CL3      0x97
00098 #define PICC_CMD_HLTA         0x50
00099 #define PICC_CMD_RATS         0xE0
00100 #define PICC_CMD_MF_AUTH_KEY_A 0x60
00101 #define PICC_CMD_MF_AUTH_KEY_B 0x61
00102 #define PICC_CMD_MF_READ      0x30
00103 #define PICC_CMD_MF_WRITE     0xA0
00104 #define PICC_CMD_MF_DECREMENT 0xC0
00105 #define PICC_CMD_MF_INCREMENT 0xC1
00106 #define PICC_CMD_MF_RESTORE   0xC2
00107 #define PICC_CMD_MF_TRANSFER  0xB0
00108 #define PICC_CMD_UL_WRITE     0xA2
00109
00110 // MFRC522's commands for the PCD.
00111 #define PCD_CMD_IDLE                0x00  // NO action; cancels current command execution
00112 #define PCD_CMD_MEM                 0x01  // Stores 25 bytes into the internal buffer
00113 #define PCD_CMD_GEN_RANDOM_ID       0x02  // Generates a 10-byte random ID number
00114 #define PCD_CMD_CALC_CRC            0x03  // Activates the CRC coprocessor or performs a self-test
00115 #define PCD_CMD_TRANSMIT            0x04  // Transmits data from the FIFO buffer
00116 #define PCD_CMD_NO_CMD_CHANGE       0x07  // Can be used to modify the CommandReg register bits
      without affecting the command, if any, currently being executed
00117 #define PCD_CMD_RECEIVE             0x08  // Activates the receiver circuits
00118 #define PCD_CMD_TRANSCEIVE          0x0C  // Transmits data from FIFO buffer to antenna and
      automatically activates the receiver after transmission
00119 #define PCD_CMD_MF_AUTHENT          0x0E  // Performs the MIFARE standard authentication as a reader
00120 #define PCD_CMD_SOFT_RESET          0x0F  // Resets the MFRC522
00131 esp_err_t MFRC522_WriteRegister(spi_device_handle_t *spiHandle, uint8_t registerAddress, uint8_t
      value);
00132
00141 esp_err_t MFRC522_ReadRegister(spi_device_handle_t *spiHandle, uint8_t registerAddress, uint8_t
      *data);
00142
00152 esp_err_t MFRC522_ReadRegisterArr(spi_device_handle_t *spiHandle, uint8_t registerAddress, uint8_t
      *dataArr, uint8_t dataSize);
00153
00163 esp_err_t MFRC522_WriteRegisterArr(spi_device_handle_t *spiHandle, uint8_t registerAddress, uint8_t
      *dataArr, uint8_t dataSize);
00164
00171 esp_err_t MFRC522_Init(spi_device_handle_t *spiHandle);
00172
00181 esp_err_t MFRC522_ClrRegBitMask(spi_device_handle_t *spiHandle, uint8_t registerAdress, uint8_t mask);
00182
00189 bool MFRC522_IsCardPresent(spi_device_handle_t *spiHandle);
00190
00197 esp_err_t MFRC522_AntennaOn(spi_device_handle_t *spiHandle);
00198
00205 esp_err_t MFRC522_SelfTest(spi_device_handle_t *spiHandle);
00206
00213 esp_err_t MFRC522_Reset(spi_device_handle_t *spiHandle);
00214
00222 esp_err_t MFRC522_SendPICCcmdTranscieve(spi_device_handle_t *spiHandle, uint8_t piccCmd);
00223
00224 #endif // _MFRC522_H_
```

Index