PEGODA RD710/RD852
Reader Firmware
Software Reference Manual

v. 1.0

Contents

1 Main Page **1**

2 Module Index **3**

2.1 Modules 3

3 Data Structure Index **5**

3.1 Data Structures 5

4 File Index **7**

4.1 File List 7

5 Module Documentation **9**

5.1 Reader Functions 9

5.1.1 Detailed Description 9

5.1.2 Function Documentation 9

5.1.2.1 p2_fw_reader_read_config 9

5.1.2.2 p2_fw_reader_set_up_external_interface 9

5.1.2.3 p2_fw_reader_set_up_reader_chip 10

5.1.2.4 p2_fw_reader_setup_hardware 10

5.2 Tasks 10

5.2.1 Detailed Description 10

5.2.2 Function Documentation 11

5.2.2.1 p2_fw_task_demo_mode 11

5.2.2.2 p2_fw_task_err_and_nfo_loop 11

5.2.2.3 p2_fw_task_pcsc_execute 11

5.2.2.4 p2_fw_task_pcsc_poll_and_act_loop 11

5.3 BFL 4.8 11

5.3.1 Detailed Description 12
5.3.2 Function Documentation 12
5.3.2.1 p2_fw_blf_change_rc523_baud_rate 12
5.3.2.2 p2_fw_blf_init 12
5.3.2.3 p2_fw_blf_reset_reader 12
5.3.2.4 p2_fw_blf_set_com_speed 13
5.3.2.5 p2_fw_blf_set_timeout 13
5.3.2.6 p2_fw_blf_set_up_rc_type_a_reading 13
5.4 Slots managment functions 13
5.4.1 Detailed Description 14
5.4.2 Function Documentation 14
5.4.2.1 p2_fw_slots_add_new_l3_card 14
5.4.2.2 p2_fw_slots_add_new_l4_card 14
5.4.2.3 p2_fw_slots_add_new_sam_card 15
5.4.2.4 p2_fw_slots_clear_cid_list 15
5.4.2.5 p2_fw_slots_free_cid 15
5.4.2.6 p2_fw_slots_get_atr 15
5.4.2.7 p2_fw_slots_get_free_cid 16
5.4.2.8 p2_fw_slots_get_free_slot 16
5.4.2.9 p2_fw_slots_init 16
5.4.2.10 p2_fw_slots_is_known_l3_card 16
5.4.2.11 p2_fw_slots_remove_card 17
5.5 Utils 17
5.5.1 Detailed Description 17
5.5.2 Function Documentation 17
5.5.2.1 p2_fw_utils_blink 17
5.5.2.2 p2_fw_utils_field_off 17
5.5.2.3 p2_fw_utils_field_on 18
5.5.2.4 p2_fw_utils_get_dri 18
5.5.2.5 p2_fw_utils_get_dsi 18
5.5.2.6 p2_fw_utils_reg_read 18
5.5.2.7 p2_fw_utils_reg_write 19
5.6 CCID functions 19
5.6.1 Detailed Description 21

5.6.2 Define Documentation 21
5.6.2.1 P2_FW_CCID_BULK_HEADER 21
5.6.2.2 P2_FW_CCID_BULK_OUT_REQ_ESCAPE . . . 22
5.6.2.3 P2_FW_CCID_BULK_OUT_REQ_-GETPARAMETERS 22
5.6.2.4 P2_FW_CCID_BULK_OUT_REQ_-GETSLOTSTATUS 22
5.6.2.5 P2_FW_CCID_BULK_OUT_REQ_ICCPOWEROFF 22
5.6.2.6 P2_FW_CCID_BULK_OUT_REQ_ICCPOWERON 22
5.6.2.7 P2_FW_CCID_BULK_OUT_REQ_-SETPARAMETERS 22
5.6.2.8 P2_FW_CCID_BULK_OUT_REQ_XFRBLOCK . 22
5.6.2.9 P2_FW_CCID_ERROR_SLOT_BUSY 22
5.6.2.10 P2_FW_CCID_ERROR_SLOT_CMD_NOT_-SUPPORTED 23
5.6.2.11 P2_FW_CCID_ERROR_SLOT_HW_ERROR . . . 23
5.6.2.12 P2_FW_CCID_ERROR_SLOT_ICC_MUTE . . . 23
5.6.2.13 P2_FW_CCID_ERROR_SLOT_NOT_EXIST . . . 23
5.6.2.14 P2_FW_CCID_ERROR_SLOT_XFR_OVERRUN . 23
5.6.2.15 P2_FW_CCID_HEADER_LENGTH_BYTE_1 . . 23
5.6.2.16 P2_FW_CCID_HEADER_LENGTH_BYTE_2 . . 23
5.6.2.17 P2_FW_CCID_HEADER_LENGTH_BYTE_3 . . 23
5.6.2.18 P2_FW_CCID_HEADER_LENGTH_BYTE_4 . . 24
5.6.2.19 P2_FW_CCID_HEADER_MESSAGE_TYPE . . . 24
5.6.2.20 P2_FW_CCID_HEADER_MSG_BYTE_1 24
5.6.2.21 P2_FW_CCID_HEADER_MSG_BYTE_2 24
5.6.2.22 P2_FW_CCID_HEADER_MSG_BYTE_3 24
5.6.2.23 P2_FW_CCID_HEADER_SEQ 24
5.6.2.24 P2_FW_CCID_HEADER_SLOT 24
5.6.2.25 P2_FW_CCID_INT_IN_NOTIFY_SLOT_CHANGE 24
5.6.2.26 P2_FW_CCID_RDR_TO_PC_ESCAPE 25
5.6.2.27 P2_FW_CCID_RDR_TO_PC_PARAMETERS . . 25
5.6.2.28 P2_FW_CCID_RDR_TO_PC_SLOT_DATA_-BLOCK 25
5.6.2.29 P2_FW_CCID_RDR_TO_PC_SLOT_STATUS . . 25

5.6.2.30 P2_FW_CCID_STATUS_CMD_FAILED 25
5.6.3 Function Documentation 25
5.6.3.1 p2_fw_ccid_check_header 25
5.6.3.2 p2_fw_ccid_escape_bottom_half 26
5.6.3.3 p2_fw_ccid_escape_top_half 26
5.6.3.4 p2_fw_ccid_get_parameters_bottom_half 26
5.6.3.5 p2_fw_ccid_get_parameters_top_half 26
5.6.3.6 p2_fw_ccid_get_slot_status_bottom_half 27
5.6.3.7 p2_fw_ccid_get_slot_status_top_half 27
5.6.3.8 p2_fw_ccid_icc_power_off_bottom_half 27
5.6.3.9 p2_fw_ccid_icc_power_off_top_half 27
5.6.3.10 p2_fw_ccid_icc_power_on_bottom_half 28
5.6.3.11 p2_fw_ccid_icc_power_on_top_half 28
5.6.3.12 p2_fw_ccid_send_apdu 28
5.6.3.13 p2_fw_ccid_send_data 29
5.6.3.14 p2_fw_ccid_send_notify 29
5.6.3.15 p2_fw_ccid_set_parameters_bottom_half 29
5.6.3.16 p2_fw_ccid_set_parameters_top_half 30
5.6.3.17 p2_fw_ccid_top_half_dispatch 30
5.6.3.18 p2_fw_ccid_xfr_block_bottom_half 30
5.6.3.19 p2_fw_ccid_xfr_block_top_half 30
5.6.3.20 p2_fw_ccid_xfr_clear_busy 31
5.6.3.21 p2_fw_ccid_xfr_set_busy 31
5.7 Configuration definitions 31
5.7.1 Detailed Description 31
5.7.2 Define Documentation 31
5.7.2.1 P2_FW_CONFIG_COMM_SER_232_PORT . . . 31
5.7.2.2 P2_FW_CONFIG_COMM_SER_485_PORT . . . 32
5.7.2.3 P2_FW_CONFIG_DEBUG_PORT 32
5.7.2.4 P2_FW_CONFIG_RC523_UART_PORT 32
5.8 Direct Mode 32
5.8.1 Detailed Description 35
5.8.2 Define Documentation 35
5.8.2.1 P2_FW_DM_ALLOWED_CMDS_ALL 35

5.8.2.2 P2_FW_DM_ALLOWED_CMDS_CID 35
5.8.2.3 P2_FW_DM_ALLOWED_CMDS_CONTACT_-CARD 35
5.8.2.4 P2_FW_DM_ALLOWED_CMDS_HAL 36
5.8.2.5 P2_FW_DM_ALLOWED_CMDS_KSTOR 36
5.8.2.6 P2_FW_DM_ALLOWED_CMDS_L3 36
5.8.2.7 P2_FW_DM_ALLOWED_CMDS_L4 36
5.8.2.8 P2_FW_DM_ALLOWED_CMDS_L4A 36
5.8.2.9 P2_FW_DM_ALLOWED_CMDS_RO 36
5.8.2.10 P2_FW_DM_ALLOWED_CMDS_XCHG 36
5.8.2.11 P2_FW_DM_CID 36
5.8.2.12 P2_FW_DM_CID_FREE 37
5.8.2.13 P2_FW_DM_CID_GET_FREE 37
5.8.2.14 P2_FW_DM_CID_INIT 37
5.8.2.15 P2_FW_DM_CLASS_BYTE 37
5.8.2.16 P2_FW_DM_CONTACT_CARD 37
5.8.2.17 P2_FW_DM_CONTACTCARD_ACTIVATE_CARD 37
5.8.2.18 P2_FW_DM_CONTACTCARD_CLOCK_START 37
5.8.2.19 P2_FW_DM_CONTACTCARD_CLOCK_STOP . 37
5.8.2.20 P2_FW_DM_CONTACTCARD_COLD_RESET . 38
5.8.2.21 P2_FW_DM_CONTACTCARD_DEACTIVATE_-CARD 38
5.8.2.22 P2_FW_DM_CONTACTCARD_PPS 38
5.8.2.23 P2_FW_DM_CONTACTCARD_PRESENCE_-CHECK 38
5.8.2.24 P2_FW_DM_CONTACTCARD_TRANSMIT_DATA 38
5.8.2.25 P2_FW_DM_CONTACTCARD_WARM_RESET . 38
5.8.2.26 P2_FW_DM_FAILD 38
5.8.2.27 P2_FW_DM_HAL 38
5.8.2.28 P2_FW_DM_HAL_APP_PROT_SET 39
5.8.2.29 P2_FW_DM_HAL_EXEC_CMD 39
5.8.2.30 P2_FW_DM_HAL_GET_CFG 39
5.8.2.31 P2_FW_DM_HAL_INIT 39
5.8.2.32 P2_FW_DM_HAL_MFC_AUTH 39
5.8.2.33 P2_FW_DM_HAL_MFC_AUTH_KEY 39

5.8.2.34 P2_FW_DM_HAL_SET_CFG 39
5.8.2.35 P2_FW_DM_HAL_WAIT 39
5.8.2.36 P2_FW_DM_HAL_XCHG 40
5.8.2.37 P2_FW_DM_INSTR_BYTE 40
5.8.2.38 P2_FW_DM_KSTOR 40
5.8.2.39 P2_FW_DM_KSTOR_CHG_KUC 40
5.8.2.40 P2_FW_DM_KSTOR_FORMAT_KEY 40
5.8.2.41 P2_FW_DM_KSTOR_GET_CFG_STR 40
5.8.2.42 P2_FW_DM_KSTOR_GET_CONFIG 40
5.8.2.43 P2_FW_DM_KSTOR_GET_KEY 40
5.8.2.44 P2_FW_DM_KSTOR_GET_KEY_ENTRY 41
5.8.2.45 P2_FW_DM_KSTOR_GET_KUC 41
5.8.2.46 P2_FW_DM_KSTOR_INIT 41
5.8.2.47 P2_FW_DM_KSTOR_SET_CEK 41
5.8.2.48 P2_FW_DM_KSTOR_SET_CFG_STR 41
5.8.2.49 P2_FW_DM_KSTOR_SET_CONFIG 41
5.8.2.50 P2_FW_DM_KSTOR_SET_FULL_KEY 41
5.8.2.51 P2_FW_DM_KSTOR_SET_KEY 41
5.8.2.52 P2_FW_DM_KSTOR_SET_KEY_POS 42
5.8.2.53 P2_FW_DM_KSTOR_SET_KUC 42
5.8.2.54 P2_FW_DM_L3 42
5.8.2.55 P2_FW_DM_L3_ACT_CARD 42
5.8.2.56 P2_FW_DM_L3_ANTICOL 42
5.8.2.57 P2_FW_DM_L3_GET_SER 42
5.8.2.58 P2_FW_DM_L3_HLTA 42
5.8.2.59 P2_FW_DM_L3_INIT 42
5.8.2.60 P2_FW_DM_L3_REQA 43
5.8.2.61 P2_FW_DM_L3_SELECT 43
5.8.2.62 P2_FW_DM_L3_WKUA 43
5.8.2.63 P2_FW_DM_L3_XCHG 43
5.8.2.64 P2_FW_DM_L4 43
5.8.2.65 P2_FW_DM_L4_DESELECT 43
5.8.2.66 P2_FW_DM_L4_GET_CFG 43
5.8.2.67 P2_FW_DM_L4_INIT 43

5.8.2.68 P2_FW_DM_L4_PRES_CHECK 44
5.8.2.69 P2_FW_DM_L4_RESET_PROTO 44
5.8.2.70 P2_FW_DM_L4_SET_CFG 44
5.8.2.71 P2_FW_DM_L4_SET_PROTO 44
5.8.2.72 P2_FW_DM_L4_XCHG 44
5.8.2.73 P2_FW_DM_L4A 44
5.8.2.74 P2_FW_DM_L4A_ACT_CARD 44
5.8.2.75 P2_FW_DM_L4A_GET_PROTO_PARM 44
5.8.2.76 P2_FW_DM_L4A_INIT 45
5.8.2.77 P2_FW_DM_L4A_PPS 45
5.8.2.78 P2_FW_DM_L4A_RATS 45
5.8.2.79 P2_FW_DM_LENGTH_LSB_IN 45
5.8.2.80 P2_FW_DM_LENGTH_LSB_OUT 45
5.8.2.81 P2_FW_DM_LENGTH_MSB_IN 45
5.8.2.82 P2_FW_DM_LENGTH_MSB_OUT 45
5.8.2.83 P2_FW_DM_OFFSET_IN 46
5.8.2.84 P2_FW_DM_OFFSET_OUT 46
5.8.2.85 P2_FW_DM_OK 46
5.8.2.86 P2_FW_DM_RO 46
5.8.2.87 P2_FW_DM_RO_CONF_OVER 46
5.8.2.88 P2_FW_DM_RO_FIELD_OFF 46
5.8.2.89 P2_FW_DM_RO_FIELD_ON 46
5.8.2.90 P2_FW_DM_RO_FIELD_RESET 46
5.8.2.91 P2_FW_DM_RO_GET_CONF 47
5.8.2.92 P2_FW_DM_RO_GET_STATUS 47
5.8.2.93 P2_FW_DM_RO_LEDS_OFF 47
5.8.2.94 P2_FW_DM_RO_LEDS_ON 47
5.8.2.95 P2_FW_DM_RO_READ_REG 47
5.8.2.96 P2_FW_DM_RO_RESET 47
5.8.2.97 P2_FW_DM_RO_SET_CONF 47
5.8.2.98 P2_FW_DM_RO_SET_PCSC_MODE 47
5.8.2.99 P2_FW_DM_RO_TEST_MODE 48
5.8.2.100 P2_FW_DM_RO_WRITE_REG 48
5.8.2.101 P2_FW_DM_STATUS_LSB 48

5.8.2.102 P2_FW_DM_STATUS_MSB 48
5.8.2.103 P2_FW_DM_XCHG 48
5.8.2.104 P2_FW_DM_XCHG_INIT 48
5.8.2.105 P2_FW_DM_XCHG_L3 48
5.8.2.106 P2_FW_DM_XCHG_L4 48
5.8.2.107 P2_FW_DM_XCHG_MFC_AUTH 49
5.8.2.108 P2_FW_DM_XCHG_MFC_AUTH_KEY 49
5.8.2.109 P2_FW_DM_XCHG_PC 49
5.8.2.110 P2_FW_DM_XCHG_RAW 49
5.8.3 Function Documentation 49
5.8.3.1 p2_fw_dm 49
5.8.3.2 p2_fw_dm_hal_wait 49
5.8.3.3 p2_fw_dm_key_store 50
5.8.3.4 p2_fw_dm_key_store_init 50
5.8.3.5 p2_fw_dm_mfc_auth_hal_key_store 50
5.8.3.6 p2_fw_dm_xcgh_l4 51
5.8.3.7 p2_fw_key_store_get_key 51
5.9 Internal Direct Mode 52
5.9.1 Detailed Description 55
5.9.2 Define Documentation 55
5.9.2.1 P2_FW_DM_CHK_LEN 55
5.9.2.2 PH_EXCHANGE_BUFFER_CONT 55
5.9.2.3 PH_EXCHANGE_BUFFER_FIRST 55
5.9.2.4 PH_EXCHANGE_BUFFER_LAST 56
5.9.2.5 PH_RC523_BIT_106KBPS 56
5.9.2.6 PH_RC523_BIT_212KBPS 56
5.9.2.7 PH_RC523_BIT_424KBPS 56
5.9.2.8 PH_RC523_BIT_848KBPS 56
5.9.2.9 PH_RC523_MASK_RXALIGN 56
5.9.2.10 PH_RC523_MASK_RXWAIT 56
5.9.2.11 PH_RC523_MASK_TXBITS 56
5.9.2.12 PH_RC523_SERIALSPEED_115200 57
5.9.2.13 PH_RC523_SERIALSPEED_19200 57
5.9.2.14 PH_RC523_SERIALSPEED_230400 57

5.9.2.15 PH_RC523_SERIALSPEED_38400 57
5.9.2.16 PH_RC523_SERIALSPEED_460800 57
5.9.2.17 PH_RC523_SERIALSPEED_57600 57
5.9.2.18 PH_RC523_SERIALSPEED_9600 57
5.9.2.19 PHHAL_HW_CARDTYPE_CURRENT 57
5.9.2.20 PHHAL_HW_CARDTYPE_ISO14443A 58
5.9.2.21 PHHAL_HW_CARDTYPE_ISO14443B 58
5.9.2.22 PHHAL_HW_CONFIG_ADDITIONAL_INFO . . 58
5.9.2.23 PHHAL_HW_CONFIG_ASK100 58
5.9.2.24 PHHAL_HW_CONFIG_BAL_CONNECTION . . 58
5.9.2.25 PHHAL_HW_CONFIG_CLEARBITSAFTERCOLL 58
5.9.2.26 PHHAL_HW_CONFIG_DISABLE_MF_CRYPTO1 58
5.9.2.27 PHHAL_HW_CONFIG_FIELD_OFF_TIME . . . 58
5.9.2.28 PHHAL_HW_CONFIG_FIELD_RECOVERY_TIME 59
5.9.2.29 PHHAL_HW_CONFIG_MAX_PRECACHED_-BYTES 59
5.9.2.30 PHHAL_HW_CONFIG_MODINDEX 59
5.9.2.31 PHHAL_HW_CONFIG_PARITY 59
5.9.2.32 PHHAL_HW_CONFIG_RFRESET_ON_TIMEOUT 59
5.9.2.33 PHHAL_HW_CONFIG_RXALIGN 59
5.9.2.34 PHHAL_HW_CONFIG_RXBUFFER_BUFSIZE . 59
5.9.2.35 PHHAL_HW_CONFIG_RXBUFFER_STARTPOS 59
5.9.2.36 PHHAL_HW_CONFIG_RXCRC 60
5.9.2.37 PHHAL_HW_CONFIG_RXDATARATE 60
5.9.2.38 PHHAL_HW_CONFIG_RXDEAFBITS 60
5.9.2.39 PHHAL_HW_CONFIG_RXLASTBITS 60
5.9.2.40 PHHAL_HW_CONFIG_SERIAL_BITRATE . . . 60
5.9.2.41 PHHAL_HW_CONFIG_SUBCARRIER 60
5.9.2.42 PHHAL_HW_CONFIG_SYMBOL_END 60
5.9.2.43 PHHAL_HW_CONFIG_SYMBOL_START 60
5.9.2.44 PHHAL_HW_CONFIG_TIMEOUT_VALUE_MS 61
5.9.2.45 PHHAL_HW_CONFIG_TIMEOUT_VALUE_US . 61
5.9.2.46 PHHAL_HW_CONFIG_TIMING_MODE 61
5.9.2.47 PHHAL_HW_CONFIG_TIMING_MS 61

5.9.2.48 PHHAL_HW_CONFIG_TIMING_US 61
5.9.2.49 PHHAL_HW_CONFIG_TXBUFFER 61
5.9.2.50 PHHAL_HW_CONFIG_TXBUFFER_BUFSIZE . 61
5.9.2.51 PHHAL_HW_CONFIG_TXBUFFER_LENGTH . 61
5.9.2.52 PHHAL_HW_CONFIG_TXCRC 62
5.9.2.53 PHHAL_HW_CONFIG_TXDATARATE 62
5.9.2.54 PHHAL_HW_CONFIG_TXLASTBITS 62
5.9.2.55 PHHAL_HW_CONFIG_TXWAIT_US 62
5.9.2.56 PHHAL_HW_MFC_KEYA 62
5.9.2.57 PHHAL_HW_MFC_KEYB 62
5.9.2.58 PHHAL_HW_MFC_USE_KEYMODIFIER 62
5.9.2.59 PHHAL_HW_RF_DATARATE_106 62
5.9.2.60 PHHAL_HW_RF_DATARATE_212 63
5.9.2.61 PHHAL_HW_RF_DATARATE_424 63
5.9.2.62 PHHAL_HW_RF_DATARATE_848 63
5.9.2.63 PHHAL_HW_RS232_BITRATE_115200 63
5.9.2.64 PHHAL_HW_RS232_BITRATE_19200 63
5.9.2.65 PHHAL_HW_RS232_BITRATE_230400 63
5.9.2.66 PHHAL_HW_RS232_BITRATE_38400 63
5.9.2.67 PHHAL_HW_RS232_BITRATE_460800 63
5.9.2.68 PHHAL_HW_RS232_BITRATE_57600 64
5.9.2.69 PHHAL_HW_RS232_BITRATE_9600 64
5.9.2.70 PHHAL_HW_TIMING_MODE_COMM 64
5.9.2.71 PHHAL_HW_TIMING_MODE_FDT 64
5.9.2.72 PHHAL_HW_TIMING_MODE_OFF 64
5.9.2.73 PHPAL_I14443P4_PARAM_BLOCKNO 64
5.9.2.74 PHPAL_I14443P4_PARAM_CID 64
5.9.2.75 PHPAL_I14443P4_PARAM_FSI 64
5.9.2.76 PHPAL_I14443P4_PARAM_FWI 65
5.9.2.77 PHPAL_I14443P4_PARAM_MAXRETRYCOUNT 65
5.9.2.78 PHPAL_I14443P4_PARAM_NAD 65
5.9.2.79 SELECT_CASCADE_LEVEL_1 65
5.9.2.80 SELECT_CASCADE_LEVEL_2 65
5.9.2.81 SELECT_CASCADE_LEVEL_3 65

5.9.2.82 SINGLE_UID_LENGTH 65
5.9.3 Function Documentation 65
5.9.3.1 p2_fw_direct_mode_xchg 65
5.9.3.2 p2_fw_dm_check_if_vaild 66
5.9.3.3 p2_fw_dm_check_max_len 66
5.9.3.4 p2_fw_dm_cid 67
5.9.3.5 p2_fw_dm_contact_card 67
5.9.3.6 p2_fw_dm_hal 67
5.9.3.7 p2_fw_dm_hal_set_cfg_rxdatarate 68
5.9.3.8 p2_fw_dm_hal_set_cfg_timeout 68
5.9.3.9 p2_fw_dm_hal_set_cfg_txdatarate 68
5.9.3.10 p2_fw_dm_hal_switch_config 68
5.9.3.11 p2_fw_dm_l3 69
5.9.3.12 p2_fw_dm_l4 69
5.9.3.13 p2_fw_dm_l4a 69
5.9.3.14 p2_fw_dm_mfc_auth_hal 70
5.9.3.15 p2_fw_dm_ro 70
5.9.3.16 p2_fw_dm_send 70
5.9.3.17 p2_fw_dm_translate_error_code 71
5.9.3.18 p2_fw_dm_xchg_hal 71
5.10 External Interfaces 71
5.10.1 Detailed Description 72
5.10.2 Define Documentation 72
5.10.2.1 P2_FW_USB_BULK_IN_EP 72
5.10.2.2 P2_FW_USB_BULK_OUT_EP 72
5.10.2.3 P2_FW_USB_INT_IN_EP 72
5.10.3 Function Documentation 72
5.10.3.1 p2_fw_serial_init_serial 72
5.10.3.2 p2_fw_usb_init_usb 73
5.11 Pegoda 2 Pins 73
5.11.1 Detailed Description 73
5.11.2 Define Documentation 73
5.11.2.1 P2_FW_PINS_ANTENA_BLUE 73
5.11.2.2 P2_FW_PINS_ANTENA_GREEN 74

5.11.2.3 P2_FW_PINS_ANTENA_RED 74
5.11.2.4 P2_FW_PINS_BEEPER 74
5.11.2.5 P2_FW_PINS_CFG_1 74
5.11.2.6 P2_FW_PINS_CFG_2 74
5.11.2.7 P2_FW_PINS_CFG_3 74
5.11.2.8 P2_FW_PINS_CFG_4 74
5.11.2.9 P2_FW_PINS_DIP_1 74
5.11.2.10 P2_FW_PINS_DIP_2 75
5.11.2.11 P2_FW_PINS_DIP_3 75
5.11.2.12 P2_FW_PINS_DIP_4 75
5.11.2.13 P2_FW_PINS_DIP_5 75
5.11.2.14 P2_FW_PINS_DIP_6 75
5.11.2.15 P2_FW_PINS_DIP_7 75
5.11.2.16 P2_FW_PINS_DIP_8 75
5.11.2.17 P2_FW_PINS_LEDS_YELLOW_2 75
5.11.2.18 P2_FW_PINS_LEDS_YELLOW_3 76
5.11.2.19 P2_FW_PINS_LEDS_YELLOW_4 76
5.12 Timing 76
5.12.1 Detailed Description 76
5.12.2 Function Documentation 76
5.12.2.1 p2_fw_timing_init 76
5.12.2.2 p2_fw_timing_start 77
5.12.2.3 p2_fw_timing_stop_cless 77
5.12.2.4 p2_fw_timing_stop_contac 77

6 Data Structure Documentation **79**
6.1 _p2_fw_data Struct Reference 79
6.1.1 Detailed Description 82
6.1.2 Field Documentation 83
6.1.2.1 ant_p 83
6.1.2.2 auth_done 83
6.1.2.3 buff_in 83
6.1.2.4 buff_in_len 83
6.1.2.5 buff_out 83

6.1.2.6 buff_out_idx 83
6.1.2.7 buff_out_len 83
6.1.2.8 c_slot 83
6.1.2.9 chip_type 84
6.1.2.10 cid_list 84
6.1.2.11 com_p 84
6.1.2.12 comm_buff 84
6.1.2.13 config 84
6.1.2.14 cont_tim 84
6.1.2.15 current_slot_idx 84
6.1.2.16 data 84
6.1.2.17 dsl_p 84
6.1.2.18 exc_p 85
6.1.2.19 g_rp 85
6.1.2.20 gm_rp 85
6.1.2.21 hlt_p 85
6.1.2.22 in 85
6.1.2.23 in_ccid_len 85
6.1.2.24 in_idx 85
6.1.2.25 in_len 85
6.1.2.26 ipc 86
6.1.2.27 is_busy 86
6.1.2.28 iso14443_3 86
6.1.2.29 iso14443_3_data 86
6.1.2.30 iso14443_4 86
6.1.2.31 iso14443_4_data 86
6.1.2.32 iso14443_4A_data 86
6.1.2.33 key 86
6.1.2.34 last_notify_evt 86
6.1.2.35 len 86
6.1.2.36 m_rp 86
6.1.2.37 max_slots 87
6.1.2.38 mfrd 87
6.1.2.39 mifare 87

6.1.2.40 mutex_transmission . . . 87
6.1.2.41 num . . . 87
6.1.2.42 out . . . 87
6.1.2.43 out_len . . . 87
6.1.2.44 out_written . . . 87
6.1.2.45 pcsc_m . . . 88
6.1.2.46 pcsc_mode . . . 88
6.1.2.47 pps_p . . . 88
6.1.2.48 pro_p . . . 88
6.1.2.49 prs_p . . . 88
6.1.2.50 queue_pcsc_exec . . . 88
6.1.2.51 rc_com_bal . . . 88
6.1.2.52 rc_io . . . 88
6.1.2.53 rc_op_ctl . . . 88
6.1.2.54 rc_reg_ctl . . . 89
6.1.2.55 reader . . . 89
6.1.2.56 regs_ops . . . 89
6.1.2.57 regs_parms . . . 89
6.1.2.58 rst_p . . . 89
6.1.2.59 rx_xspeed . . . 89
6.1.2.60 s_rp . . . 89
6.1.2.61 scb_p . . . 89
6.1.2.62 sel_p . . . 89
6.1.2.63 send . . . 89
6.1.2.64 slot_data . . . 90
6.1.2.65 slots . . . 90
6.1.2.66 sm_rp . . . 90
6.1.2.67 spr_p . . . 90
6.2 _p2_fw_pcsc_exec Struct Reference . . . 90
6.2.1 Detailed Description . . . 90
6.2.2 Field Documentation . . . 90
6.2.2.1 bottom_half . . . 90
6.2.2.2 slot . . . 91
6.3 _p2_fw_slot Struct Reference . . . 91

6.3.1 Detailed Description 92
6.3.2 Field Documentation 92
6.3.2.1 atq 92
6.3.2.2 ats 92
6.3.2.3 b_block_pool 92
6.3.2.4 bRfResetAfterTo 92
6.3.2.5 c_type 92
6.3.2.6 card_pro_p 93
6.3.2.7 cardtype_hw 93
6.3.2.8 chain 93
6.3.2.9 data 93
6.3.2.10 direct_m 93
6.3.2.11 dri 93
6.3.2.12 dsi 93
6.3.2.13 dwTimingUs 93
6.3.2.14 iso14443_4 94
6.3.2.15 len 94
6.3.2.16 m_BlockNumber 94
6.3.2.17 m_Flags 94
6.3.2.18 m_HistoryFlag 94
6.3.2.19 m_State 94
6.3.2.20 max_retrys 94
6.3.2.21 nad 94
6.3.2.22 params 95
6.3.2.23 proto 95
6.3.2.24 s_regs 95
6.3.2.25 sak 95
6.3.2.26 type 95
6.3.2.27 uid 95
6.3.2.28 wAdditionalInfo 95
6.3.2.29 wFieldOffTime 95
6.3.2.30 wFieldRecoveryTime 95
6.3.2.31 wTimeoutUnit 96
6.3.2.32 wTimingMode 96

6.4 cpot_atr_frame Struct Reference 96
6.4.1 Detailed Description 96
6.4.2 Field Documentation 96
6.4.2.1 historicalC 96
6.4.2.2 state 96
6.4.2.3 TAi 96
6.4.2.4 TBi 97
6.4.2.5 TCi 97
6.4.2.6 TCK 97
6.4.2.7 TDi 97
6.5 p2_fw_SAM_ctrl_ Struct Reference 97
6.5.1 Detailed Description 98
6.5.2 Field Documentation 98
6.5.2.1 bitIndex 98
6.5.2.2 bwi 98
6.5.2.3 chipMode 98
6.5.2.4 chipType 98
6.5.2.5 conversion 98
6.5.2.6 data 98
6.5.2.7 data 98
6.5.2.8 dataIndex 98
6.5.2.9 etu 99
6.5.2.10 len 99
6.5.2.11 mode 99
6.5.2.12 parityCount 99
6.5.2.13 recExtraGuardTime 99
6.5.2.14 send_data 99
6.5.2.15 sendLen 99
6.5.2.16 sendSeqData 99
6.5.2.17 SessionATR 99
6.5.2.18 SessionATR_Size 99
6.5.2.19 t0 99
6.5.2.20 timingMode 100
6.5.2.21 tmpByteWait_time 100

6.6 sam_t1_param Struct Reference 100
6.6.1 Detailed Description 100
6.6.2 Field Documentation 100
6.6.2.1 BWI_CWI 100
6.6.2.2 ClockStop 100
6.6.2.3 FI_DI 100
6.6.2.4 GuardTime 101
6.6.2.5 IFSC 101

7 File Documentation **103**

7.1 include/p2_fw.h File Reference 103
7.1.1 Define Documentation 108
7.1.1.1 __float32_t_defined 108
7.1.1.2 P2_FW_APDU_CC_EXT_INS 108
7.1.1.3 P2_FW_APDU_CC_EXT_MANAGE_SESSION . 108
7.1.1.4 P2_FW_APDU_CC_EXT_SWITCH_PROTOCOL 108
7.1.1.5 P2_FW_APDU_CC_EXT_TRANS_EXCHANGE . 108
7.1.1.6 P2_FW_APDU_CLASS 108
7.1.1.7 P2_FW_APDU_G_AUTH_CMD_INS 108
7.1.1.8 P2_FW_APDU_GET_DATA_INS 108
7.1.1.9 P2_FW_APDU_INS 109
7.1.1.10 P2_FW_APDU_Lc 109
7.1.1.11 P2_FW_APDU_LOAD_KEY_INS 109
7.1.1.12 P2_FW_APDU_P1 109
7.1.1.13 P2_FW_APDU_P2 109
7.1.1.14 P2_FW_APDU_READ_BIN 109
7.1.1.15 P2_FW_APDU_UPDATE_BIN 109
7.1.1.16 P2_FW_CARD_IN_SLOT 109
7.1.1.17 P2_FW_CARD_OUT_SLOT 110
7.1.1.18 P2_FW_CFG_BOOTLOADER_ACTIVE 110
7.1.1.19 P2_FW_CFG_BOOTLOADER_VERSION 110
7.1.1.20 P2_FW_CFG_CONT_TIMING 110
7.1.1.21 P2_FW_CFG_GET_CONT_TIMING 110
7.1.1.22 P2_FW_CFG_MAX_SLOTS 110

7.1.1.23 P2_FW_CFG_SET_DIP_SWITCHES 110
7.1.1.24 P2_FW_DEFAULT_SLOTS 110
7.1.1.25 P2_FW_ERR_AND_NFO_LOOP_BFL_ERROR . 110
7.1.1.26 P2_FW_ERR_AND_NFO_LOOP_CHIP_NOT_-DETECTED 110
7.1.1.27 P2_FW_ERR_AND_NFO_LOOP_COM_ERROR . 111
7.1.1.28 P2_FW_ERR_AND_NFO_LOOP_DONE 111
7.1.1.29 P2_FW_ERR_AND_NFO_LOOP_ERASE_FAILD 111
7.1.1.30 P2_FW_ERR_AND_NFO_LOOP_FLASH_FAILD 111
7.1.1.31 P2_FW_ERR_AND_NFO_LOOP_HAL_CAN_-NOT_BE_SET 111
7.1.1.32 P2_FW_ERR_AND_NFO_LOOP_HW_ERROR . 111
7.1.1.33 P2_FW_ERR_AND_NFO_LOOP_OS_ERROR . . 111
7.1.1.34 P2_FW_ERR_AND_NFO_LOOP_UNKNOWN_-ERROR 112
7.1.1.35 P2_FW_EXT_COM_INTF_ETHERNET 112
7.1.1.36 P2_FW_EXT_COM_INTF_RS232 112
7.1.1.37 P2_FW_EXT_COM_INTF_RS485 112
7.1.1.38 P2_FW_EXT_COM_INTF_USB 112
7.1.1.39 P2_FW_FSI 112
7.1.1.40 P2_FW_INT_COM_INTF_I2C 112
7.1.1.41 P2_FW_INT_COM_INTF_SPI 112
7.1.1.42 P2_FW_INT_COM_INTF_UART 113
7.1.1.43 P2_FW_MAX_ATS 113
7.1.1.44 P2_FW_MAX_COMM_BUFF_SIZE 113
7.1.1.45 P2_FW_MAX_EXCH_BUFF_SIZE 113
7.1.1.46 P2_FW_MAX_PCSC_KEY_LEN 113
7.1.1.47 P2_FW_MAX_RETRY_ISO14443_4A 113
7.1.1.48 P2_FW_MAX_SLOT_USB_BUFF_SIZE 113
7.1.1.49 P2_FW_MAX_SLOTS 113
7.1.1.50 P2_FW_MAX_UID 114
7.1.1.51 P2_FW_MODE_DEMO 114
7.1.1.52 P2_FW_MODE_ENT_SEC_BOOT 114
7.1.1.53 P2_FW_MODE_OVER_USER_CFG 114
7.1.1.54 P2_FW_MODE_PCSC 114

7.1.1.55 P2_FW_PCSC_MODE_DIRECT_MODE 114
7.1.1.56 P2_FW_PCSC_MODE_STANDARD 114
7.1.1.57 P2_FW_PCSC_PROTO_PARAMS_LEN_T0 . . . 115
7.1.1.58 P2_FW_PCSC_PROTO_PARAMS_LEN_T1 . . . 115
7.1.1.59 P2_FW_PCSC_PROTO_RAW 115
7.1.1.60 P2_FW_PCSC_PROTO_T0 115
7.1.1.61 P2_FW_PCSC_PROTO_T1 115
7.1.1.62 P2_FW_PCSC_SHADOW_REGS 115
7.1.1.63 P2_FW_READER_CHIP_RC523 115
7.1.1.64 P2_FW_READER_CHIP_SAM 115
7.1.1.65 P2_FW_READER_MODE_NO_SAM 116
7.1.1.66 P2_FW_READER_MODE_SAM_IN_X 116
7.1.1.67 P2_FW_READER_MODE_SAM_NON_X 116
7.1.1.68 P2_FW_TMR_MS 116
7.1.1.69 P2_FW_TMR_START_NOW 116
7.1.1.70 P2_FW_TMR_US 116
7.1.2 Typedef Documentation 116
7.1.2.1 float32_t 116
7.1.2.2 p2_fw_data 116
7.1.2.3 p2_fw_pcsc_exec 117
7.1.2.4 p2_fw_slot 117
7.1.3 Enumeration Type Documentation 117
7.1.3.1 p2_fw_card_type 117
7.1.4 Function Documentation 117
7.1.4.1 p2_fw_ccid_xfr_clear_busy 117
7.1.4.2 p2_fw_ccid_xfr_set_busy 117
7.1.4.3 p2_fw_flash_erase_config 117
7.1.4.4 p2_fw_flash_get_config 118
7.1.4.5 p2_fw_flash_read_serial 118
7.1.4.6 p2_fw_flash_set_config 118
7.1.4.7 p2_fw_invoke_error_mode 118
7.1.4.8 p2_fw_pcsc_commands 119
7.1.4.9 p2_fw_pcsc_prepare_l3_card 119
7.1.4.10 p2_fw_pcsc_send_apdu 119

7.1.4.11 p2_fw_utils_dump_regs 119
7.1.4.12 phcsBflBal_Hw1SerCM3Init 119
7.1.4.13 UART3_IRQHandler 120
7.1.5 Variable Documentation 120
7.1.5.1 p2_fw 120
7.2 include/p2_fw_ccid.h File Reference 120
7.3 include/p2_fw_config.h File Reference 123
7.4 include/p2_fw_direct_mode.h File Reference 123
7.5 include/p2_fw_direct_mode_int.h File Reference 127
7.6 include/p2_fw_ext.h File Reference 130
7.7 include/p2_fw_peridot_pins.h File Reference 131
7.8 include/p2_fw_sam_t1.h File Reference 132
7.8.1 Define Documentation 133
7.8.1.1 P2_FW_SAM_FRAME_APDU 133
7.8.1.2 P2_FW_SAM_FRAME_T1 133
7.8.1.3 P2_FW_SAM_MODE_PPS 133
7.8.1.4 P2_FW_SAM_MODE_X 133
7.8.1.5 P2_FW_TIMING_MODE_COM 133
7.8.1.6 P2_FW_TIMING_MODE_FDT 133
7.8.1.7 P2_FW_TIMING_MODE_NONE 133
7.8.2 Function Documentation 133
7.8.2.1 EINT3_IRQHandler 133
7.8.2.2 p2_fw_sam_t1_deinit 133
7.8.2.3 p2_fw_sam_t1_get_atr 133
7.8.2.4 p2_fw_sam_t1_get_timing_mode 134
7.8.2.5 p2_fw_sam_t1_init 134
7.8.2.6 p2_fw_sam_t1_is_busy 134
7.8.2.7 p2_fw_sam_t1_is_power_off 134
7.8.2.8 p2_fw_sam_t1_is_sam_inserted 134
7.8.2.9 p2_fw_sam_t1_pps 134
7.8.2.10 p2_fw_sam_t1_prepare_pps 134
7.8.2.11 p2_fw_sam_t1_receive 134
7.8.2.12 p2_fw_sam_t1_send 134
7.8.2.13 p2_fw_sam_t1_set_bwi_cwi 134

7.8.2.14 p2_fw_sam_t1_set_etu . . . 135
7.8.2.15 p2_fw_sam_t1_set_my_debug . . . 135
7.8.2.16 p2_fw_sam_t1_set_rec_extraGuardTime . . . 135
7.8.2.17 p2_fw_sam_t1_set_timing_mode . . . 135
7.8.2.18 p2_fw_sam_t1_start . . . 135
7.8.2.19 p2_fw_sam_t1_warm_reset . . . 135
7.8.2.20 TIMER0_IRQHandler . . . 135
7.8.2.21 TIMER2_IRQHandler . . . 135
7.9 include/p2_fw_timing.h File Reference . . . 136
7.10 include/p2_fw_version.h File Reference . . . 137
7.10.1 Define Documentation . . . 137
7.10.1.1 P2_FW_VER_BUILD . . . 137
7.10.1.2 P2_FW_VER_MAJOR . . . 137
7.10.1.3 P2_FW_VER_MINOR . . . 137
7.11 src/bfl_support_files/phcsBflBal_Hw1SerCM3.c File Reference . . . 137
7.11.1 Detailed Description . . . 138
7.11.2 Define Documentation . . . 138
7.11.2.1 BUFFER_LENGTH . . . 138
7.11.2.2 THE_UART . . . 138
7.11.2.3 UART_PORT . . . 139
7.11.3 Function Documentation . . . 139
7.11.3.1 phcsBflBal_Hw1SerCM3Init . . . 139
7.12 src/bfl_support_files/phcsBflRegCtl_SpiHw1.c File Reference . . . 139
7.12.1 Detailed Description . . . 140
7.12.2 Define Documentation . . . 140
7.12.2.1 BUFFER_SIZE . . . 140
7.12.3 Function Documentation . . . 140
7.12.3.1 phcsBflRegCtl_SpiHw1GetMultiReg . . . 140
7.12.3.2 phcsBflRegCtl_SpiHw1GetReg . . . 140
7.12.3.3 phcsBflRegCtl_SpiHw1Init . . . 140
7.12.3.4 phcsBflRegCtl_SpiHw1ModReg . . . 141
7.12.3.5 phcsBflRegCtl_SpiHw1SetMultiReg . . . 141
7.12.3.6 phcsBflRegCtl_SpiHw1SetReg . . . 141
7.13 src/p2_fw_blf.c File Reference . . . 141

7.13.1 Define Documentation 142
7.13.1.1 PHHAL_HW_RC523_ETU_106 142
7.13.1.2 PHHAL_HW_RC523_TIMER_FREQ 142
7.13.1.3 PHHAL_HW_RC523_TIMER_SHIFT 142
7.14 src/p2_fw_ccid.c File Reference 142
7.15 src/p2_fw_ccid_xfer.c File Reference 144
7.15.1 Function Documentation 144
7.15.1.1 p2_fw_pcsc_prepare_l3_card 144
7.15.1.2 p2_fw_pcsc_send_apdu 145
7.16 src/p2_fw_demo_mode.c File Reference 145
7.17 src/p2_fw_direct_mode.c File Reference 145
7.18 src/p2_fw_direct_mode_cid.c File Reference 146
7.19 src/p2_fw_direct_mode_contact_card.c File Reference 147
7.20 src/p2_fw_direct_mode_hal.c File Reference 148
7.21 src/p2_fw_direct_mode_key_store.c File Reference 149
7.21.1 Define Documentation 150
7.21.1.1 KEY_COUNT 150
7.21.1.2 KEY_VERSIONS 150
7.22 src/p2_fw_direct_mode_l3.c File Reference 150
7.23 src/p2_fw_direct_mode_l4.c File Reference 151
7.23.1 Define Documentation 152
7.23.1.1 PHPAL_I14443P4_CID_MAX 152
7.23.1.2 PHPAL_I14443P4_FRAMESIZE_MAX 152
7.23.1.3 PHPAL_I14443P4_FWI_MAX 152
7.24 src/p2_fw_direct_mode_l4a.c File Reference 152
7.25 src/p2_fw_direct_mode_ro.c File Reference 153
7.26 src/p2_fw_direct_mode_xchg.c File Reference 154
7.27 src/p2_fw_err_and_nfo_mode.c File Reference 154
7.28 src/p2_fw_ext_intf_serial.c File Reference 155
7.28.1 Define Documentation 156
7.28.1.1 SERIAL_UART 156
7.28.2 Function Documentation 156
7.28.2.1 UART3_IRQHandler 156
7.29 src/p2_fw_ext_intf_usb.c File Reference 156

7.29.1 Define Documentation 157
7.29.1.1 P2_FW_MAX_SLOT_INDEX 157
7.29.1.2 P2_FW_NAME_USB_OFFSET 157
7.29.1.3 P2_FW_SERNUM_USB_OFFSET 157
7.29.1.4 P2_FW_USB_CONTROL_REQUEST_ABORT . . 157
7.29.1.5 P2_FW_USB_CONTROL_REQUEST_GET_-CLOCK_FREQUENCIES 157
7.29.1.6 P2_FW_USB_CONTROL_REQUEST_GET_-DATA_RATES 157
7.29.1.7 P2_FW_USB_LE_DWORD 157
7.29.1.8 P2_FW_USB_LE_WORD 157
7.29.1.9 P2_FW_USB_MAX_PACKET_SIZE 157
7.30 src/p2_fw_flash_utils.c File Reference 158
7.30.1 Define Documentation 159
7.30.1.1 IAP_BLANK_CHECK_SECTOR 159
7.30.1.2 IAP_COMPARE 159
7.30.1.3 IAP_COPY_RAM_TO_FLASH 159
7.30.1.4 IAP_ERASE_SECTOR 159
7.30.1.5 IAP_LOCATION 159
7.30.1.6 IAP_PREP_SECTORS 159
7.30.1.7 IAP_READ_BOOT_CODE_VER 159
7.30.1.8 IAP_READ_DEV_SER_NUM 159
7.30.1.9 IAP_READ_ID 159
7.30.1.10 IAP_REINVOKE 159
7.30.1.11 IAP_SEC_28 160
7.30.1.12 IAP_SEC_29 160
7.30.1.13 P2_FW_CFG_COUNT 160
7.30.1.14 P2_FW_CFG_ENTRY_OFFSET 160
7.30.1.15 P2_FW_CFG_ENTRY_SIZE 160
7.30.1.16 P2_FW_CFG_MAX 160
7.30.1.17 P2_FW_CFG_S1 160
7.30.1.18 P2_FW_CFG_S2 160
7.30.2 Typedef Documentation 160
7.30.2.1 IAP 160

7.30.3 Function Documentation 160
7.30.3.1 p2_fw_flash_erase_config 160
7.30.3.2 p2_fw_flash_get_config 161
7.30.3.3 p2_fw_flash_read_serial 161
7.30.3.4 p2_fw_flash_set_config 161
7.31 src/p2_fw_main.c File Reference 162
7.31.1 Function Documentation 162
7.31.1.1 main 162
7.31.1.2 p2_fw_invoke_error_mode 162
7.31.2 Variable Documentation 163
7.31.2.1 p2_fw 163
7.32 src/p2_fw_pcsc_ext.c File Reference 163
7.32.1 Function Documentation 163
7.32.1.1 p2_fw_pcsc_commands 163
7.33 src/p2_fw_pcsc_mode.c File Reference 164
7.34 src/p2_fw_reader.c File Reference 164
7.35 src/p2_fw_sam_t1.c File Reference 165
7.35.1 Define Documentation 167
7.35.1.1 BITBAND_PERI 167
7.35.1.2 BITBAND_PERI_BASE 167
7.35.1.3 BITBAND_PERI_REF 167
7.35.1.4 P2_FW_SAM_CLOCK_LEN_NS 167
7.35.1.5 P2_FW_SAM_CONVERSION_DIRECT 167
7.35.1.6 P2_FW_SAM_CONVERSION_INVERSE 167
7.35.1.7 P2_FW_SAM_CONVERSION_UNKNOWN . . . 167
7.35.1.8 P2_FW_SAM_DEBUG 167
7.35.1.9 P2_FW_SAM_DEFAULT_BWI 168
7.35.1.10 P2_FW_SAM_MAX_ATR_SIZE 168
7.35.1.11 P2_FW_SAM_MAX_REC_DATALEN 168
7.35.1.12 P2_FW_SAM_MODE_IDLE 168
7.35.1.13 P2_FW_SAM_MODE_POWER_OFF 168
7.35.1.14 P2_FW_SAM_MODE_RECEIVE 168
7.35.1.15 P2_FW_SAM_MODE_RESET 168
7.35.1.16 P2_FW_SAM_MODE_SEND 168

7.35.1.17 P2_FW_SAM_MODE_WAIT_START_BIT 168
7.35.1.18 P2_FW_SAM_PWM_CH0_MATCH_VALUE . . . 168
7.35.1.19 P2_FW_SAM_PWM_CH1_MATCH_VALUE . . . 169
7.35.1.20 P2_FW_SAM_T1_CLOCK_CONFIG 169
7.35.1.21 P2_FW_SAM_TIMER_PRESCALE_VALUE . . . 169
7.35.1.22 T1_BYTE_WAIT_TIME 169
7.35.1.23 TIMER0_IR 169
7.35.1.24 TIMER0_IR_MR0 169
7.35.1.25 TIMER0_TCR 169
7.35.1.26 TIMER0_TCR_ENABLE 169
7.35.1.27 TIMER0_TCR_RESET 169
7.35.2 Function Documentation 170
7.35.2.1 EINT3_IRQHandler 170
7.35.2.2 p2_fw_sam_t1_deinit 170
7.35.2.3 p2_fw_sam_t1_get_atr 170
7.35.2.4 p2_fw_sam_t1_get_timing_mode 170
7.35.2.5 p2_fw_sam_t1_init 170
7.35.2.6 p2_fw_sam_t1_is_busy 170
7.35.2.7 p2_fw_sam_t1_is_power_off 170
7.35.2.8 p2_fw_sam_t1_is_sam_inserted 170
7.35.2.9 p2_fw_sam_t1_pps 170
7.35.2.10 p2_fw_sam_t1_prepare_pps 170
7.35.2.11 p2_fw_sam_t1_receive 171
7.35.2.12 p2_fw_sam_t1_send 171
7.35.2.13 p2_fw_sam_t1_set_bwi_cwi 171
7.35.2.14 p2_fw_sam_t1_set_etu 171
7.35.2.15 p2_fw_sam_t1_set_my_debug 171
7.35.2.16 p2_fw_sam_t1_set_rec_extraGuardTime 171
7.35.2.17 p2_fw_sam_t1_set_timing_mode 171
7.35.2.18 p2_fw_sam_t1_start 171
7.35.2.19 p2_fw_sam_t1_warm_reset 171
7.35.2.20 TIMER0_IRQHandler 171
7.35.2.21 TIMER2_IRQHandler 172
7.36 src/p2_fw_slots.c File Reference 172

7.37 src/p2_fw_timing_com.c File Reference 172
7.37.1 Define Documentation 173
7.37.1.1 PHHAL_HW_RC523_DIGI_DELAY_US 173
7.37.1.2 PHHAL_HW_RC523_ETU_106 173
7.37.1.3 PHHAL_HW_RC523_TIMER_FREQ 174
7.37.1.4 PHHAL_HW_RC523_TIMER_SHIFT 174
7.37.1.5 TIMING_MODE_OPTION_AUTOCLEAR 174
7.37.1.6 TIMING_MODE_OPTION_MASK 174
7.38 src/p2_fw_utils.c File Reference 174
7.38.1 Function Documentation 175
7.38.1.1 p2_fw_utils_dump_regs 175

Chapter 1

Main Page

This reference manual documents the SW architecture of the Pegoda RD710/RD852 reader firmware.

STRICT PRODUCT

LIABILITY OR OTHERWISE, EVEN IF ENDUSER HAS BEEN ADVISED OF THE POSSIBILITY

OF SUCH DAMAGES.

Some states or jurisdictions do not allow the exclusion or limitation of incidental or consequential

damages, so the above limitation, exclusion or parts of it may not apply to you.

Revision History

VERSION	DATE	DESCRIPTION
1.0	10.11.2010	First Release

Chapter 2

Module Index

2.1 Modules

Here is a list of all modules:

Reader Functions 9
Tasks 10
BFL 4.8 11
Slots managment functions 13
Utils 17
CCID functions 19
Configuration definitions 31
Direct Mode 32
Internal Direct Mode 52
External Interfaces 71
Pegoda 2 Pins 73
Timing 76

Chapter 3

Data Structure Index

3.1 Data Structures

Here are the data structures with brief descriptions:

_p2_fw_data 79
_p2_fw_pcsc_exec (Job parameters for p2_fw_task_pcsc_execute task) . . . 90
_p2_fw_slot (Contains slot private data) 91
cpot_atr_frame 96
p2_fw_SAM_ctrl_ 97
sam_t1_param 100

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

include/p2_fw.h . . . 103
include/p2_fw_ccid.h . . . 120
include/p2_fw_config.h . . . 123
include/p2_fw_direct_mode.h . . . 123
include/p2_fw_direct_mode_int.h . . . 127
include/p2_fw_ext.h . . . 130
include/p2_fw_peridot_pins.h . . . 131
include/p2_fw_sam_t1.h . . . 132
include/p2_fw_timing.h . . . 136
include/p2_fw_version.h . . . 137
src/p2_fw_blf.c . . . 141
src/p2_fw_ccid.c . . . 142
src/p2_fw_ccid_xfer.c . . . 144
src/p2_fw_demo_mode.c . . . 145
src/p2_fw_direct_mode.c . . . 145
src/p2_fw_direct_mode_cid.c . . . 146
src/p2_fw_direct_mode_contact_card.c . . . 147
src/p2_fw_direct_mode_hal.c . . . 148
src/p2_fw_direct_mode_key_store.c . . . 149
src/p2_fw_direct_mode_l3.c . . . 150
src/p2_fw_direct_mode_l4.c . . . 151
src/p2_fw_direct_mode_l4a.c . . . 152
src/p2_fw_direct_mode_ro.c . . . 153
src/p2_fw_direct_mode_xchg.c . . . 154
src/p2_fw_err_and_nfo_mode.c . . . 154
src/p2_fw_ext_intf_serial.c . . . 155
src/p2_fw_ext_intf_usb.c . . . 156
src/p2_fw_flash_utils.c . . . 158
src/p2_fw_main.c . . . 162

src/p2_fw_pcsc_ext.c 163
src/p2_fw_pcsc_mode.c 164
src/p2_fw_reader.c 164
src/p2_fw_sam_t1.c 165
src/p2_fw_slots.c 172
src/p2_fw_timing_com.c 172
src/p2_fw_utils.c 174
src/bfl_support_files/phcsBflBal_Hw1SerCM3.c 137
src/bfl_support_files/phcsBflRegCtl_SpiHw1.c 139

Chapter 5

Module Documentation

5.1 Reader Functions

Functions

- void p2_fw_reader_setup_hardware (void)
- void p2_fw_reader_read_config (void)
- Bool p2_fw_reader_set_up_reader_chip (void)
- void p2_fw_reader_set_up_external_interface ()

5.1.1 Detailed Description

The Reader Functions perform configuration and initialization of the reader.

5.1.2 Function Documentation

5.1.2.1 void p2_fw_reader_read_config (void)

Checks reader type and DIP switches

Parameters

void

Definition at line 116 of file p2_fw_reader.c.

5.1.2.2 void p2_fw_reader_set_up_external_interface ( )

Depending on the reader chip and configuration sets external interface

Parameters

void

Returns

void

Definition at line 210 of file p2_fw_reader.c.

5.1.2.3 Bool p2_fw_reader_set_up_reader_chip (void)

Depending on the reader chip and configuration sets up reader chip and BFL.

Parameters

void

Returns

Bool - TRUE if success or FALSE if error

Definition at line 232 of file p2_fw_reader.c.

5.1.2.4 void p2_fw_reader_setup_hardware (void)

Sets up reader hardware: -calls initialization code from CMSIS -initialization of DIP switch and LEDs GPIOs

Parameters

void

Definition at line 49 of file p2_fw_reader.c.

5.2 Tasks

Functions

- void p2_fw_task_err_and_nfo_loop (void *param)
- void p2_fw_task_demo_mode (void *param)
- void p2_fw_task_pcsc_poll_and_act_loop (void *param)
- void p2_fw_task_pcsc_execute (void *param)

5.2.1 Detailed Description

FreeRTOS tasks of our system.

5.2.2 Function Documentation

5.2.2.1 void p2_fw_task_demo_mode (void * *param*)

The Demo Mode works by performing anticolision and select on one card and then blinking the number of times the blue LEDs that are found on Pegoda 2 antena according to cards SAK.

Parameters

param - not used (pass NULL)

Definition at line 30 of file p2_fw_demo_mode.c.

5.2.2.2 void p2_fw_task_err_and_nfo_loop (void * *param*)

Reports errors and information back to te user by means of multiple blue LEDs that are found on Pegoda 2 antena.

Parameters

param - integer error or information value cast to (void *)

Definition at line 26 of file p2_fw_err_and_nfo_mode.c.

5.2.2.3 void p2_fw_task_pcsc_execute (void * *param*)

Executes the scheduled bottom halfs

Parameters

param - not used (pass NULL)

Definition at line 33 of file p2_fw_pcsc_mode.c.

5.2.2.4 void p2_fw_task_pcsc_poll_and_act_loop (void * *param*)

Poll and Activation Loop is used to perform PC/SC part 2 figure 2-1.

Parameters

param - not used (pass NULL)

Definition at line 66 of file p2_fw_pcsc_mode.c.

5.3 BFL 4.8

Functions

- Bool p2_fw_blf_init (void)

- void p2_fw_blf_set_up_rc_type_a_reading (void)
- Bool p2_fw_blf_reset_reader (void)
- void p2_fw_blf_set_timeout (uint16_t qsec, uint8_t aFlags)
- void p2_fw_blf_set_com_speed (uint8_t dri, uint8_t dsi)
- void p2_fw_blf_change_rc523_baud_rate (uint32_t baudrate)

5.3.1 Detailed Description

Functions for initilization of BFL 4.8 and its operation.

5.3.2 Function Documentation

5.3.2.1 void p2_fw_blf_change_rc523_baud_rate (uint32_t *baudrate*)

Set reader UART speed

Parameters

baudrate - new baudrate

Definition at line 418 of file p2_fw_blf.c.

5.3.2.2 Bool p2_fw_blf_init (void)

Initializes BFL. Performs initialization of required BFL structures. We initialize the hardware interface, the required subsystems (IO and OpCtrl) and ISO14443 layer 3, 4A and 4 components.

Parameters

param - not used (pass NULL)

Definition at line 40 of file p2_fw_blf.c.

5.3.2.3 Bool p2_fw_blf_reset_reader (void)

Resets the reader chip using the softreset command

Parameters

param - not used (pass NULL)

Definition at line 224 of file p2_fw_blf.c.

5.3.2.4 void p2_fw_blf_set_com_speed (uint8_t *dri*, uint8_t *dsi*)

Set card communication speed

Parameters

dri - DSI parameter

dsi - DRI parameter

Definition at line 369 of file p2_fw_blf.c.

5.3.2.5 void p2_fw_blf_set_timeout (uint16_t *qsec*, uint8_t *aFlags*)

Sets timeout to the reader chip timer in milliseconds or microseconds depending on the flags setting. Please note that values larger then 39590 ms will cause overflow.

Parameters

qsec - timeout value

aFlags - type of timeout (P2_FW_TMR_MS, P2_FW_TMR_US) and should we force start (P2_FW_TMR_START_NOW)

Definition at line 238 of file p2_fw_blf.c.

5.3.2.6 void p2_fw_blf_set_up_rc_type_a_reading (void)

Sets up required parameters for reading type A cards

Parameters

param - not used (pass NULL)

Returns

Bool - TRUE if success or FALSE if error

Definition at line 135 of file p2_fw_blf.c.

5.4 Slots managment functions

Functions

- void p2_fw_slots_init (void)
- Bool p2_fw_slots_add_new_l4_card (phcsBflI3P4AAct_RatsParam_t *rat_p, uint8_t cid_index, uint8_t sak, uint8_t *atq, uint8_t *uid, uint8_t uid_len)
- Bool p2_fw_slots_add_new_l3_card (uint8_t sak, uint8_t *atq, uint8_t *uid, uint8_t uid_len)

- Bool p2_fw_slots_add_new_sam_card ()
- Bool p2_fw_slots_is_known_l3_card (uint8_t ∗uid, uint8_t uid_len)
- Bool p2_fw_slots_get_free_cid (uint8_t ∗cid)
- Bool p2_fw_slots_get_free_slot (uint8_t ∗slot)
- void p2_fw_slots_free_cid (uint8_t cid)
- void p2_fw_slots_remove_card (uint8_t slot_index)
- void p2_fw_slots_clear_cid_list (void)
- Bool p2_fw_slots_get_atr (uint8_t slot_index, uint8_t ∗buffer, uint8_t ∗max_-length)

5.4.1 Detailed Description

Functions for initilization and managment of slots which contain cards.

5.4.2 Function Documentation

5.4.2.1 Bool p2_fw_slots_add_new_l3_card (uint8_t *sak*, uint8_t ∗ *atq*, uint8_t ∗ *uid*, uint8_t *uid_len*)

Adds a new L3 card to a slot.

Parameters

sak - uint8_t - SAK of the card

atq - uint8_t[2] - pointer to buffer with ATQ of the card

uid - uint8_t[uid_len] - pointer to buffer with UID of the card

uid_len - uint8_t - length of uid

Returns

Bool - TRUE if success or FALSE if error

Definition at line 167 of file p2_fw_slots.c.

5.4.2.2 Bool p2_fw_slots_add_new_l4_card (phcsBflI3P4AAct_RatsParam_t ∗ *rat_p*, uint8_t *cid_index*, uint8_t *sak*, uint8_t ∗ *atq*, uint8_t ∗ *uid*, uint8_t *uid_len*)

Adds a new L4 card to a slot.

Parameters

cid_index - uint8_t - CID on which we talk to the card

sak - uint8_t - SAK of the card

atq - uint8_t[2] - pointer to buffer with ATQ of the card

uid - uint8_t[uid_len] - pointer to buffer with UID of the card

uid_len - uint8_t - length of uid

Returns

Bool - TRUE if success or FALSE if error

Definition at line 125 of file p2_fw_slots.c.

5.4.2.3 Bool p2_fw_slots_add_new_sam_card ( )

Adds a new SAM card to a slot.

Returns

Bool - TRUE if success or FALSE if error

Definition at line 202 of file p2_fw_slots.c.

5.4.2.4 void p2_fw_slots_clear_cid_list (void)

Clears all slots CID

5.4.2.5 void p2_fw_slots_free_cid (uint8_t *cid*)

Frees a CID.

Parameters

cid - CID id to free

Definition at line 47 of file p2_fw_slots.c.

5.4.2.6 Bool p2_fw_slots_get_atr (uint8_t *slot_index,* uint8_t * *buffer,* uint8_t * *max_length*)

Returns ATR for a card in a slot

Parameters

slot_index - index of a slot to return ATR

buffer - pointer to buffer to put the ATR in to

max_length - length of the buffer

Returns

Bool - TRUE if success or FALSE if error

Definition at line 257 of file p2_fw_slots.c.

5.4.2.7 Bool p2_fw_slots_get_free_cid (uint8_t * *cid*)

Returns a free CID for communication with ISO14443-4 card.

Parameters

cid - uint8_t * - pointer to uint8_t for cid

Returns

Bool - TRUE if success or FALSE if error

Definition at line 52 of file p2_fw_slots.c.

5.4.2.8 Bool p2_fw_slots_get_free_slot (uint8_t * *slot*)

Returns a free slot.

Parameters

void

Returns

Bool - TRUE if success or FALSE if error

Definition at line 93 of file p2_fw_slots.c.

5.4.2.9 void p2_fw_slots_init (void)

Initialize slots to default state

Definition at line 22 of file p2_fw_slots.c.

5.4.2.10 Bool p2_fw_slots_is_known_l3_card (uint8_t * *uid,* uint8_t *uid_len*)

Checks if the L3 card we found is the same as the one we are currently talking to.

Parameters

uid - uint8_t[uid_len] - pointer to buffer with UID of the card

uid_len - uint8_t - length of uid

Returns

Bool - TRUE if success or FALSE if error

Definition at line 234 of file p2_fw_slots.c.

5.4.2.11 void p2_fw_slots_remove_card (uint8_t *slot_index*)

Removes card from a slot.

Parameters

slot_index - uint8_t - slot number to nuke

Definition at line 435 of file p2_fw_slots.c.

5.5 Utils

Functions

- uint8_t p2_fw_utils_get_dri (uint8_t ta1)
- uint8_t p2_fw_utils_get_dsi (uint8_t ta1)
- void p2_fw_utils_blink (int count)
- void p2_fw_utils_field_off (void)
- void p2_fw_utils_field_on (uint16_t wFiledRecoveryTime)
- void p2_fw_utils_reg_read (uint8_t addr, uint8_t ∗val)
- void p2_fw_utils_reg_write (uint8_t addr, uint8_t val)

5.5.1 Detailed Description

Supporting function.

5.5.2 Function Documentation

5.5.2.1 void p2_fw_utils_blink (int *count*)

Blinks the Pegoda 2 antena blue LEDs for **count** times

Parameters

count - int - number of times to blink the antena LEDs

Definition at line 129 of file p2_fw_utils.c.

5.5.2.2 void p2_fw_utils_field_off (void)

Turns off RF

Definition at line 146 of file p2_fw_utils.c.

5.5.2.3 void p2_fw_utils_field_on (uint16_t *wFiledRecoveryTime*)

Turns on RF

Definition at line 155 of file p2_fw_utils.c.

5.5.2.4 uint8_t p2_fw_utils_get_dri (uint8_t *ta1*)

Parses TA1 from ATS and returns DRI.

Parameters

ta1 - uint8_t - the value of cards ATSs TA1

Returns

uint8_t - DRI

Definition at line 24 of file p2_fw_utils.c.

5.5.2.5 uint8_t p2_fw_utils_get_dsi (uint8_t *ta1*)

Parses TA1 from ATS and returns DSI.

Parameters

ta1 - uint8_t - the value of cards ATSs TA1

Returns

uint8_t - DSI

Definition at line 77 of file p2_fw_utils.c.

5.5.2.6 void p2_fw_utils_reg_read (uint8_t *addr,* uint8_t * *val*)

Read from a register

Parameters

addr - register address

val - value of the register

Returns

uint8_t - DSI

Definition at line 195 of file p2_fw_utils.c.

5.5.2.7 void p2_fw_utils_reg_write (uint8_t *addr*, uint8_t *val*)

Write to a register

Parameters

- register address
- value to write

Definition at line 184 of file p2_fw_utils.c.

5.6 CCID functions

Defines

- #define P2_FW_CCID_BULK_HEADER 0x0A

Functions

- void p2_fw_ccid_top_half_dispatch (void)

  Top level ISR Dispatcher.

- Bool p2_fw_ccid_check_header (uint8_t message_type)

  Checks if the CCID header is correct.

- Bool p2_fw_ccid_send_apdu (uint8_t *payload, uint8_t payload_len, uint8_t sw1, uint8_t sw2)

  Sends APDU with payload of payload_len and SW1 and SW2.

- void p2_fw_ccid_send_data (uint8_t message_type, uint8_t byte_1, uint8_t byte_2, uint8_t byte_3)

  Sends the CCID message.

- void p2_fw_ccid_xfr_block_top_half (void)

  Top half (ISR) function for transfer command.

- Bool p2_fw_ccid_xfr_block_bottom_half (uint8_t slot_idx)

  Bottom half function for transfer command.

- void p2_fw_ccid_get_slot_status_top_half (void)

  Top half (ISR) function for get status command.

- Bool p2_fw_ccid_get_slot_status_bottom_half (uint8_t slot_idx)

  Bottom half function for get status command.

- void p2_fw_ccid_icc_power_on_top_half (void)
  Top half (ISR) function for power on command.
- Bool p2_fw_ccid_icc_power_on_bottom_half (uint8_t slot_idx)
  Bottom half function for power on command.
- void p2_fw_ccid_icc_power_off_top_half (void)
  Top half (ISR) function for power off command.
- Bool p2_fw_ccid_icc_power_off_bottom_half (uint8_t slot_idx)
  Bottom half function for power off command.
- void p2_fw_ccid_get_parameters_top_half (void)
  Top half (ISR) function for get parameters command.
- Bool p2_fw_ccid_get_parameters_bottom_half (uint8_t slot_idx)
  Bottom half function for get parameters command.
- void p2_fw_ccid_set_parameters_top_half (void)
  Top half (ISR) function for set parameters command.
- Bool p2_fw_ccid_set_parameters_bottom_half (uint8_t slot_idx)
  Bottom half function for set parameters command.
- void p2_fw_ccid_escape_top_half (void)
  Top half (ISR) function for escape command.
- Bool p2_fw_ccid_escape_bottom_half (uint8_t slot_idx)
  Bottom half function for Escape command.
- Bool p2_fw_ccid_send_notify (uint8_t slot_idx)
  Sends slot change notify event.
- void p2_fw_ccid_xfr_set_busy (void)
  Set the slot to busy (receiving and processing data).
- void p2_fw_ccid_xfr_clear_busy (void)
  Set the slot to not busy (receiving and processing data).

Block CCID Commands

- #define P2_FW_CCID_BULK_OUT_REQ_ICCPOWERON 0x62
- #define P2_FW_CCID_BULK_OUT_REQ_ICCPOWEROFF 0x63
- #define P2_FW_CCID_BULK_OUT_REQ_GETSLOTSTATUS 0x65
- #define P2_FW_CCID_BULK_OUT_REQ_XFRBLOCK 0x6F

- #define P2_FW_CCID_BULK_OUT_REQ_GETPARAMETERS 0x6C
- #define P2_FW_CCID_BULK_OUT_REQ_SETPARAMETERS 0x61
- #define P2_FW_CCID_BULK_OUT_REQ_ESCAPE 0x6B
- #define P2_FW_CCID_INT_IN_NOTIFY_SLOT_CHANGE 0x50

Block CCID Replays

- #define P2_FW_CCID_RDR_TO_PC_SLOT_DATA_BLOCK 0x80
- #define P2_FW_CCID_RDR_TO_PC_SLOT_STATUS 0x81
- #define P2_FW_CCID_RDR_TO_PC_PARAMETERS 0x82
- #define P2_FW_CCID_RDR_TO_PC_ESCAPE 0x83

Block CCID Error Defines

- #define P2_FW_CCID_STATUS_CMD_FAILED 0x40
- #define P2_FW_CCID_ERROR_SLOT_BUSY 0xE0
- #define P2_FW_CCID_ERROR_SLOT_NOT_EXIST 0x05
- #define P2_FW_CCID_ERROR_SLOT_ICC_MUTE 0xFE
- #define P2_FW_CCID_ERROR_SLOT_XFR_OVERRUN 0xFC
- #define P2_FW_CCID_ERROR_SLOT_CMD_NOT_SUPPORTED 0x00
- #define P2_FW_CCID_ERROR_SLOT_HW_ERROR 0xFB

Block CCID Header Indexes

- #define P2_FW_CCID_HEADER_MESSAGE_TYPE 0x00
- #define P2_FW_CCID_HEADER_LENGTH_BYTE_1 0x01
- #define P2_FW_CCID_HEADER_LENGTH_BYTE_2 0x02
- #define P2_FW_CCID_HEADER_LENGTH_BYTE_3 0x03
- #define P2_FW_CCID_HEADER_LENGTH_BYTE_4 0x04
- #define P2_FW_CCID_HEADER_SLOT 0x05
- #define P2_FW_CCID_HEADER_SEQ 0x06
- #define P2_FW_CCID_HEADER_MSG_BYTE_1 0x07
- #define P2_FW_CCID_HEADER_MSG_BYTE_2 0x08
- #define P2_FW_CCID_HEADER_MSG_BYTE_3 0x09

5.6.1 Detailed Description

Supporting function.

5.6.2 Define Documentation

5.6.2.1 #define P2_FW_CCID_BULK_HEADER 0x0A

CCID Header Size

Definition at line 53 of file p2_fw_ccid.h.

5.6.2.2 #define P2_FW_CCID_BULK_OUT_REQ_ESCAPE 0x6B

ICC Power On Command

Definition at line 37 of file p2_fw_ccid.h.

5.6.2.3 #define P2_FW_CCID_BULK_OUT_REQ_GETPARAMETERS 0x6C

Get Parameters Command

Definition at line 35 of file p2_fw_ccid.h.

5.6.2.4 #define P2_FW_CCID_BULK_OUT_REQ_GETSLOTSTATUS 0x65

Get Slot Status Command

Definition at line 33 of file p2_fw_ccid.h.

5.6.2.5 #define P2_FW_CCID_BULK_OUT_REQ_ICCPOWEROFF 0x63

ICC Power Off Command

Definition at line 32 of file p2_fw_ccid.h.

5.6.2.6 #define P2_FW_CCID_BULK_OUT_REQ_ICCPOWERON 0x62

ICC Power On Command

Definition at line 31 of file p2_fw_ccid.h.

5.6.2.7 #define P2_FW_CCID_BULK_OUT_REQ_SETPARAMETERS 0x61

ICC Power On Command

Definition at line 36 of file p2_fw_ccid.h.

5.6.2.8 #define P2_FW_CCID_BULK_OUT_REQ_XFRBLOCK 0x6F

Transfer Block Command

Definition at line 34 of file p2_fw_ccid.h.

5.6.2.9 #define P2_FW_CCID_ERROR_SLOT_BUSY 0xE0

Error Code: Slot is busy

Definition at line 61 of file p2_fw_ccid.h.

5.6.2.10 #define P2_FW_CCID_ERROR_SLOT_CMD_NOT_SUPPORTED 0x00

Error Code: Command not supported

Definition at line 65 of file p2_fw_ccid.h.

5.6.2.11 #define P2_FW_CCID_ERROR_SLOT_HW_ERROR 0xFB

Error Code: Hardware error

Definition at line 66 of file p2_fw_ccid.h.

5.6.2.12 #define P2_FW_CCID_ERROR_SLOT_ICC_MUTE 0xFE

Error Code: ICC is mute

Definition at line 63 of file p2_fw_ccid.h.

5.6.2.13 #define P2_FW_CCID_ERROR_SLOT_NOT_EXIST 0x05

Error Code: Slot does not exit

Definition at line 62 of file p2_fw_ccid.h.

5.6.2.14 #define P2_FW_CCID_ERROR_SLOT_XFR_OVERRUN 0xFC

Error Code: Buffer overrun

Definition at line 64 of file p2_fw_ccid.h.

5.6.2.15 #define P2_FW_CCID_HEADER_LENGTH_BYTE_1 0x01

CCID Header - Byte 2 - Length 1 (LSB)

Definition at line 74 of file p2_fw_ccid.h.

5.6.2.16 #define P2_FW_CCID_HEADER_LENGTH_BYTE_2 0x02

CCID Header - Byte 3 - Length 2

Definition at line 75 of file p2_fw_ccid.h.

5.6.2.17 #define P2_FW_CCID_HEADER_LENGTH_BYTE_3 0x03

CCID Header - Byte 4 - Length 3

Definition at line 76 of file p2_fw_ccid.h.

5.6.2.18 #define P2_FW_CCID_HEADER_LENGTH_BYTE_4 0x04

CCID Header - Byte 5 - Length 4 (MSB)

Definition at line 77 of file p2_fw_ccid.h.

5.6.2.19 #define P2_FW_CCID_HEADER_MESSAGE_TYPE 0x00

CCID Header - Byte 1 - Message Type

Definition at line 73 of file p2_fw_ccid.h.

5.6.2.20 #define P2_FW_CCID_HEADER_MSG_BYTE_1 0x07

CCID Header - Byte 8 - Message Byte 1

Definition at line 80 of file p2_fw_ccid.h.

5.6.2.21 #define P2_FW_CCID_HEADER_MSG_BYTE_2 0x08

CCID Header - Byte 9 - Message Byte 2

Definition at line 81 of file p2_fw_ccid.h.

5.6.2.22 #define P2_FW_CCID_HEADER_MSG_BYTE_3 0x09

CCID Header - Byte 10 - Message Byte 3

Definition at line 82 of file p2_fw_ccid.h.

5.6.2.23 #define P2_FW_CCID_HEADER_SEQ 0x06

CCID Header - Byte 7 - Sequence

Definition at line 79 of file p2_fw_ccid.h.

5.6.2.24 #define P2_FW_CCID_HEADER_SLOT 0x05

CCID Header - Byte 6 - Slot Number

Definition at line 78 of file p2_fw_ccid.h.

5.6.2.25 #define P2_FW_CCID_INT_IN_NOTIFY_SLOT_CHANGE 0x50

Slot Change Notify Command

Definition at line 39 of file p2_fw_ccid.h.

5.6.2.26 #define P2_FW_CCID_RDR_TO_PC_ESCAPE 0x83

Escape Message Replay Block

Definition at line 50 of file p2_fw_ccid.h.

5.6.2.27 #define P2_FW_CCID_RDR_TO_PC_PARAMETERS 0x82

Slot Parameters Replay

Definition at line 49 of file p2_fw_ccid.h.

5.6.2.28 #define P2_FW_CCID_RDR_TO_PC_SLOT_DATA_BLOCK 0x80

Data Message Replay Block

Definition at line 47 of file p2_fw_ccid.h.

5.6.2.29 #define P2_FW_CCID_RDR_TO_PC_SLOT_STATUS 0x81

Slot Status Replay

Definition at line 48 of file p2_fw_ccid.h.

5.6.2.30 #define P2_FW_CCID_STATUS_CMD_FAILED 0x40

Command Faild

Definition at line 59 of file p2_fw_ccid.h.

5.6.3 Function Documentation

5.6.3.1 Bool p2_fw_ccid_check_header (uint8_t *message_type*)

Checks if the CCID header is correct.

Returns

Boolean

Return values

TRUE - header is correct

Parameters

message_type Type of CCID message recived

Definition at line 678 of file p2_fw_ccid.c.

5.6.3.2 Bool p2_fw_ccid_escape_bottom_half (uint8_t *slot_idx*)

Bottom half function for Escape command.

Returns

Boolean

Return values

TRUE - successful

Parameters

slot_idx slot index for which to perform escape command

Definition at line 562 of file p2_fw_ccid.c.

5.6.3.3 void p2_fw_ccid_escape_top_half (void)

Top half (ISR) function for escape command.

Returns

Nothing

Definition at line 540 of file p2_fw_ccid.c.

5.6.3.4 Bool p2_fw_ccid_get_parameters_bottom_half (uint8_t *slot_idx*)

Bottom half function for get parameters command.

Returns

Boolean

Return values

TRUE - successful

Parameters

slot_idx slot index for which to perform get parameters command

Definition at line 401 of file p2_fw_ccid.c.

5.6.3.5 void p2_fw_ccid_get_parameters_top_half (void)

Top half (ISR) function for get parameters command.

Returns

Nothing

Definition at line 379 of file p2_fw_ccid.c.

5.6.3.6 Bool p2_fw_ccid_get_slot_status_bottom_half (uint8_t *slot_idx*)

Bottom half function for get status command.

Returns

Boolean

Return values

TRUE - successful

Parameters

slot_idx slot index for which to perform status command

Definition at line 142 of file p2_fw_ccid.c.

5.6.3.7 void p2_fw_ccid_get_slot_status_top_half (void)

Top half (ISR) function for get status command.

Returns

Nothing

Definition at line 120 of file p2_fw_ccid.c.

5.6.3.8 Bool p2_fw_ccid_icc_power_off_bottom_half (uint8_t *slot_idx*)

Bottom half function for power off command.

Returns

Boolean

Return values

TRUE - successful

Parameters

slot_idx slot index for which to perform power off command

Definition at line 322 of file p2_fw_ccid.c.

5.6.3.9 void p2_fw_ccid_icc_power_off_top_half (void)

Top half (ISR) function for power off command.

Returns

Nothing

Definition at line 300 of file p2_fw_ccid.c.

5.6.3.10 Bool p2_fw_ccid_icc_power_on_bottom_half (uint8_t *slot_idx*)

Bottom half function for power on command.

Returns

Boolean

Return values

TRUE - successful

Parameters

slot_idx slot index for which to perform power on command

Definition at line 214 of file p2_fw_ccid.c.

5.6.3.11 void p2_fw_ccid_icc_power_on_top_half (void)

Top half (ISR) function for power on command.

Returns

Nothing

Definition at line 192 of file p2_fw_ccid.c.

5.6.3.12 Bool p2_fw_ccid_send_apdu (uint8_t * *payload*, uint8_t *payload_len*, uint8_t *sw1*, uint8_t *sw2*)

Sends APDU with payload of payload_len and SW1 and SW2.

Returns

Boolean

Return values

TRUE - successful send

Parameters

payload payload to be send

payload_len length of the payload

sw1 APDU SW1

sw2 APDU SW2

Definition at line 692 of file p2_fw_ccid.c.

5.6.3.13 void p2_fw_ccid_send_data (uint8_t *message_type*, uint8_t *byte_1*, uint8_t *byte_2*, uint8_t *byte_3*)

Sends the CCID message.

Returns

Nothing

Parameters

message_type Type of CCID message to send

byte_1 CCID Message Byte 1

byte_2 CCID Message Byte 2

byte_3 CCID Message Byte 3

Definition at line 714 of file p2_fw_ccid.c.

5.6.3.14 Bool p2_fw_ccid_send_notify (uint8_t *slot_idx*)

Sends slot change notify event.

Returns

Boolean

Return values

TRUE - successful

Parameters

slot_idx slot index (not used)

Definition at line 578 of file p2_fw_ccid.c.

5.6.3.15 Bool p2_fw_ccid_set_parameters_bottom_half (uint8_t *slot_idx*)

Bottom half function for set parameters command.

Returns

Boolean

Return values

TRUE - successful

Parameters

slot_idx slot index for which to perform set parameters command

Definition at line 482 of file p2_fw_ccid.c.

5.6.3.16 void p2_fw_ccid_set_parameters_top_half (void)

Top half (ISR) function for set parameters command.

Returns

Nothing

Definition at line 436 of file p2_fw_ccid.c.

5.6.3.17 void p2_fw_ccid_top_half_dispatch (void)

Top level ISR Dispatcher.

Returns

Nothing

Definition at line 26 of file p2_fw_ccid.c.

5.6.3.18 Bool p2_fw_ccid_xfr_block_bottom_half (uint8_t *slot_idx*)

Bottom half function for transfer command.

Returns

Boolean

Return values

TRUE - successful

Parameters

slot_idx slot index for which to perform transfer command

Definition at line 74 of file p2_fw_ccid_xfer.c.

5.6.3.19 void p2_fw_ccid_xfr_block_top_half (void)

Top half (ISR) function for transfer command.

Returns

Nothing

Definition at line 40 of file p2_fw_ccid_xfer.c.

5.6.3.20 void p2_fw_ccid_xfr_clear_busy (void)

Set the slot to not busy (receiving and processing data).

Returns

Nothing

Reader completed action

Definition at line 33 of file p2_fw_ccid_xfer.c.

5.6.3.21 void p2_fw_ccid_xfr_set_busy (void)

Set the slot to busy (receiving and processing data).

Returns

Nothing

Reader started to perform action

Definition at line 27 of file p2_fw_ccid_xfer.c.

5.7 Configuration definitions

Defines

- #define P2_FW_CONFIG_RC523_UART_PORT 1
- #define P2_FW_CONFIG_DEBUG_PORT 0
- #define P2_FW_CONFIG_COMM_SER_232_PORT 3
- #define P2_FW_CONFIG_COMM_SER_485_PORT 1

5.7.1 Detailed Description

Configuration definitions

5.7.2 Define Documentation

5.7.2.1 #define P2_FW_CONFIG_COMM_SER_232_PORT 3

Sets which LPC17XX UART port is used for communication over serial (232)

Definition at line 27 of file p2_fw_config.h.

5.7.2.2 #define P2_FW_CONFIG_COMM_SER_485_PORT 1

Sets which LPC17XX UART port is used for communication over serial (485)

Definition at line 28 of file p2_fw_config.h.

5.7.2.3 #define P2_FW_CONFIG_DEBUG_PORT 0

Sets which LPC17XX UART port is used for debug messages

Definition at line 26 of file p2_fw_config.h.

5.7.2.4 #define P2_FW_CONFIG_RC523_UART_PORT 1

Sets which LPC17XX UART port is used for communication with RC523

Definition at line 25 of file p2_fw_config.h.

5.8 Direct Mode

Defines

- #define P2_FW_DM_CLASS_BYTE P2_FW_CCID_BULK_HEADER
- #define P2_FW_DM_INSTR_BYTE P2_FW_CCID_BULK_HEADER + 0x01
- #define P2_FW_DM_STATUS_LSB P2_FW_CCID_BULK_HEADER + 0x02
- #define P2_FW_DM_STATUS_MSB P2_FW_CCID_BULK_HEADER + 0x03
- #define P2_FW_DM_LENGTH_LSB_IN P2_FW_CCID_BULK_HEADER + 0x02
- #define P2_FW_DM_LENGTH_MSB_IN P2_FW_CCID_BULK_HEADER + 0x03
- #define P2_FW_DM_LENGTH_LSB_OUT P2_FW_CCID_BULK_HEADER + 0x04
- #define P2_FW_DM_LENGTH_MSB_OUT P2_FW_CCID_BULK_HEADER + 0x05
- #define P2_FW_DM_OFFSET_IN P2_FW_CCID_BULK_HEADER + 0x04
- #define P2_FW_DM_OFFSET_OUT P2_FW_CCID_BULK_HEADER + 0x06
- #define P2_FW_DM_OK 0x80
- #define P2_FW_DM_FAILD 0xF0
- #define P2_FW_DM_ALLOWED_CMDS_ALL 0xFFFF
- #define P2_FW_DM_ALLOWED_CMDS_RO 0x0001
- #define P2_FW_DM_ALLOWED_CMDS_HAL 0x0002
- #define P2_FW_DM_ALLOWED_CMDS_L3 0x0008
- #define P2_FW_DM_ALLOWED_CMDS_L4A 0x0010
- #define P2_FW_DM_ALLOWED_CMDS_L4 0x0020
- #define P2_FW_DM_ALLOWED_CMDS_XCHG 0x0040
- #define P2_FW_DM_ALLOWED_CMDS_CID 0x0080
- #define P2_FW_DM_ALLOWED_CMDS_KSTOR 0x0100

- #define P2_FW_DM_ALLOWED_CMDS_CONTACT_CARD 0x0200
- #define P2_FW_DM_CID 0x90
- #define P2_FW_DM_RO 0xA0
- #define P2_FW_DM_HAL 0xB0
- #define P2_FW_DM_L3 0xC0
- #define P2_FW_DM_L4A 0xD0
- #define P2_FW_DM_L4 0xE0
- #define P2_FW_DM_XCHG 0xF0
- #define P2_FW_DM_KSTOR 0x70
- #define P2_FW_DM_CONTACT_CARD 0x80
- #define P2_FW_DM_RO_LEDS_OFF 0x01
- #define P2_FW_DM_RO_LEDS_ON 0x02
- #define P2_FW_DM_RO_RESET 0x03
- #define P2_FW_DM_RO_CONF_OVER 0x04
- #define P2_FW_DM_RO_SET_CONF 0x05
- #define P2_FW_DM_RO_GET_CONF 0x0C
- #define P2_FW_DM_RO_GET_STATUS 0x06
- #define P2_FW_DM_RO_READ_REG 0x07
- #define P2_FW_DM_RO_WRITE_REG 0x08
- #define P2_FW_DM_RO_FIELD_ON 0x09
- #define P2_FW_DM_RO_FIELD_OFF 0x0A
- #define P2_FW_DM_RO_FIELD_RESET 0x0B
- #define P2_FW_DM_RO_SET_PCSC_MODE 0x0D
- #define P2_FW_DM_RO_TEST_MODE 0x0E
- #define P2_FW_DM_HAL_INIT 0x01
- #define P2_FW_DM_HAL_XCHG 0x02
- #define P2_FW_DM_HAL_SET_CFG 0x03
- #define P2_FW_DM_HAL_GET_CFG 0x04
- #define P2_FW_DM_HAL_APP_PROT_SET 0x05
- #define P2_FW_DM_HAL_WAIT 0x06
- #define P2_FW_DM_HAL_MFC_AUTH 0x07
- #define P2_FW_DM_HAL_EXEC_CMD 0x08
- #define P2_FW_DM_HAL_MFC_AUTH_KEY 0x09
- #define P2_FW_DM_L3_INIT 0x01
- #define P2_FW_DM_L3_REQA 0x02
- #define P2_FW_DM_L3_WKUA 0x03
- #define P2_FW_DM_L3_HLTA 0x04
- #define P2_FW_DM_L3_ANTICOL 0x05
- #define P2_FW_DM_L3_SELECT 0x06
- #define P2_FW_DM_L3_ACT_CARD 0x07
- #define P2_FW_DM_L3_XCHG 0x08
- #define P2_FW_DM_L3_GET_SER 0x09
- #define P2_FW_DM_L4A_INIT 0x01
- #define P2_FW_DM_L4A_RATS 0x02
- #define P2_FW_DM_L4A_PPS 0x03
- #define P2_FW_DM_L4A_ACT_CARD 0x04

- #define P2_FW_DM_L4A_GET_PROTO_PARM 0x05
- #define P2_FW_DM_L4_INIT 0x01
- #define P2_FW_DM_L4_SET_PROTO 0x02
- #define P2_FW_DM_L4_RESET_PROTO 0x03
- #define P2_FW_DM_L4_DESELECT 0x04
- #define P2_FW_DM_L4_PRES_CHECK 0x05
- #define P2_FW_DM_L4_XCHG 0x06
- #define P2_FW_DM_L4_SET_CFG 0x07
- #define P2_FW_DM_L4_GET_CFG 0x08
- #define P2_FW_DM_XCHG_L3 0x01
- #define P2_FW_DM_XCHG_L4 0x02
- #define P2_FW_DM_XCHG_PC 0x03
- #define P2_FW_DM_XCHG_RAW 0x04
- #define P2_FW_DM_XCHG_MFC_AUTH 0x05
- #define P2_FW_DM_XCHG_MFC_AUTH_KEY 0x06
- #define P2_FW_DM_XCHG_INIT 0x07
- #define P2_FW_DM_CID_GET_FREE 0x01
- #define P2_FW_DM_CID_FREE 0x02
- #define P2_FW_DM_CID_INIT 0x03
- #define P2_FW_DM_KSTOR_INIT 0x01
- #define P2_FW_DM_KSTOR_FORMAT_KEY 0x02
- #define P2_FW_DM_KSTOR_SET_KEY 0x03
- #define P2_FW_DM_KSTOR_SET_KEY_POS 0x04
- #define P2_FW_DM_KSTOR_SET_KUC 0x05
- #define P2_FW_DM_KSTOR_SET_CEK 0x06
- #define P2_FW_DM_KSTOR_SET_FULL_KEY 0x07
- #define P2_FW_DM_KSTOR_GET_KEY_ENTRY 0x08
- #define P2_FW_DM_KSTOR_GET_KEY 0x09
- #define P2_FW_DM_KSTOR_SET_CONFIG 0x0A
- #define P2_FW_DM_KSTOR_GET_CONFIG 0x0B
- #define P2_FW_DM_KSTOR_CHG_KUC 0x0C
- #define P2_FW_DM_KSTOR_GET_KUC 0x0D
- #define P2_FW_DM_KSTOR_SET_CFG_STR 0x0E
- #define P2_FW_DM_KSTOR_GET_CFG_STR 0x0F
- #define P2_FW_DM_CONTACTCARD_ACTIVATE_CARD 0x01
- #define P2_FW_DM_CONTACTCARD_COLD_RESET 0x02
- #define P2_FW_DM_CONTACTCARD_WARM_RESET 0x03
- #define P2_FW_DM_CONTACTCARD_CLOCK_STOP 0x04
- #define P2_FW_DM_CONTACTCARD_CLOCK_START 0x05
- #define P2_FW_DM_CONTACTCARD_DEACTIVATE_CARD 0x06
- #define P2_FW_DM_CONTACTCARD_PRESENCE_CHECK 0x07
- #define P2_FW_DM_CONTACTCARD_TRANSMIT_DATA 0x08
- #define P2_FW_DM_CONTACTCARD_PPS 0x09

Functions

- Bool p2_fw_dm (uint8_t message_type, uint16_t allowed_cmds)
  Main dispatcher function for direct mode.
- Bool p2_fw_dm_xcgh_l4 (uint8_t message_type, uint8_t slot_idx)
  Performs the exchange function for L4.
- Bool p2_fw_dm_key_store (uint8_t message_type)
  Performs KeyStore related functions.
- Bool p2_fw_dm_key_store_init (void)
  Inits the key store.
- Bool p2_fw_dm_mfc_auth_hal_key_store (uint8_t message_type)
  Performs authentication by using key store.
- void p2_fw_dm_hal_wait (uint16_t timeout1, uint8_t flags)
  Performs wait.
- Bool p2_fw_key_store_get_key (uint16_t key_num, uint16_t key_ver, uint8_t *p_key, uint8_t key_len, uint16_t *key_type)
  Returns a key store key.

5.8.1 Detailed Description

Defines and declarations for direct mode

5.8.2 Define Documentation

5.8.2.1 #define P2_FW_DM_ALLOWED_CMDS_ALL 0xFFFF

Direct Mode Commands Set: All

Definition at line 40 of file p2_fw_direct_mode.h.

5.8.2.2 #define P2_FW_DM_ALLOWED_CMDS_CID 0x0080

Direct Mode Commands Set: Channel ID

Definition at line 48 of file p2_fw_direct_mode.h.

5.8.2.3 #define P2_FW_DM_ALLOWED_CMDS_CONTACT_CARD 0x0200

Direct Mode Commands Set: Contact Card

Definition at line 50 of file p2_fw_direct_mode.h.

5.8.2.4 #define P2_FW_DM_ALLOWED_CMDS_HAL 0x0002

Direct Mode Commands Set: HAL

Definition at line 43 of file p2_fw_direct_mode.h.

5.8.2.5 #define P2_FW_DM_ALLOWED_CMDS_KSTOR 0x0100

Direct Mode Commands Set: Key Store

Definition at line 49 of file p2_fw_direct_mode.h.

5.8.2.6 #define P2_FW_DM_ALLOWED_CMDS_L3 0x0008

Direct Mode Commands Set: ISO14443-3

Definition at line 44 of file p2_fw_direct_mode.h.

5.8.2.7 #define P2_FW_DM_ALLOWED_CMDS_L4 0x0020

Direct Mode Commands Set: ISO14443-4

Definition at line 46 of file p2_fw_direct_mode.h.

5.8.2.8 #define P2_FW_DM_ALLOWED_CMDS_L4A 0x0010

Direct Mode Commands Set: ISO14443-4A

Definition at line 45 of file p2_fw_direct_mode.h.

5.8.2.9 #define P2_FW_DM_ALLOWED_CMDS_RO 0x0001

Direct Mode Commands Set: Reader

Definition at line 42 of file p2_fw_direct_mode.h.

5.8.2.10 #define P2_FW_DM_ALLOWED_CMDS_XCHG 0x0040

Direct Mode Commands Set: Exchange

Definition at line 47 of file p2_fw_direct_mode.h.

5.8.2.11 #define P2_FW_DM_CID 0x90

Direct Mode Class: Channel ID

Definition at line 52 of file p2_fw_direct_mode.h.

5.8.2.12 #define P2_FW_DM_CID_FREE 0x02

Direct Mode Command: CID Free

Definition at line 121 of file p2_fw_direct_mode.h.

5.8.2.13 #define P2_FW_DM_CID_GET_FREE 0x01

Direct Mode Command: CID Get Free

Definition at line 120 of file p2_fw_direct_mode.h.

5.8.2.14 #define P2_FW_DM_CID_INIT 0x03

Direct Mode Command: CID Init

Definition at line 122 of file p2_fw_direct_mode.h.

5.8.2.15 #define P2_FW_DM_CLASS_BYTE P2_FW_CCID_BULK_HEADER

Direct Mode CCID Offset: Class Byte

Definition at line 25 of file p2_fw_direct_mode.h.

5.8.2.16 #define P2_FW_DM_CONTACT_CARD 0x80

Direct Mode Class: Contact Card

Definition at line 60 of file p2_fw_direct_mode.h.

5.8.2.17 #define P2_FW_DM_CONTACTCARD_ACTIVATE_CARD 0x01

Direct Mode Command: ContactCard Activate Card

Definition at line 140 of file p2_fw_direct_mode.h.

5.8.2.18 #define P2_FW_DM_CONTACTCARD_CLOCK_START 0x05

Direct Mode Command: ContactCard Clock Start

Definition at line 144 of file p2_fw_direct_mode.h.

5.8.2.19 #define P2_FW_DM_CONTACTCARD_CLOCK_STOP 0x04

Direct Mode Command: ContactCard Clock Stop

Definition at line 143 of file p2_fw_direct_mode.h.

5.8.2.20 #define P2_FW_DM_CONTACTCARD_COLD_RESET 0x02

Direct Mode Command: ContactCard Cold Reset

Definition at line 141 of file p2_fw_direct_mode.h.

5.8.2.21 #define P2_FW_DM_CONTACTCARD_DEACTIVATE_CARD 0x06

Direct Mode Command: ContactCard Deactivate Card

Definition at line 145 of file p2_fw_direct_mode.h.

5.8.2.22 #define P2_FW_DM_CONTACTCARD_PPS 0x09

Direct Mode Command: ContactCard Send PPS

Definition at line 148 of file p2_fw_direct_mode.h.

5.8.2.23 #define P2_FW_DM_CONTACTCARD_PRESENCE_CHECK 0x07

Direct Mode Command: ContactCard Presence Check

Definition at line 146 of file p2_fw_direct_mode.h.

5.8.2.24 #define P2_FW_DM_CONTACTCARD_TRANSMIT_DATA 0x08

Direct Mode Command: ContactCard Transmit Data

Definition at line 147 of file p2_fw_direct_mode.h.

5.8.2.25 #define P2_FW_DM_CONTACTCARD_WARM_RESET 0x03

Direct Mode Command: ContactCard Warm Reset

Definition at line 142 of file p2_fw_direct_mode.h.

5.8.2.26 #define P2_FW_DM_FAILD 0xF0

Direct Mode Status Code: FAILD

Definition at line 38 of file p2_fw_direct_mode.h.

5.8.2.27 #define P2_FW_DM_HAL 0xB0

Direct Mode Class: HAL

Definition at line 54 of file p2_fw_direct_mode.h.

5.8.2.28 #define P2_FW_DM_HAL_APP_PROT_SET 0x05

Direct Mode Command: HAL Set Protocol

Definition at line 81 of file p2_fw_direct_mode.h.

5.8.2.29 #define P2_FW_DM_HAL_EXEC_CMD 0x08

Direct Mode Command: HAL Execute Command

Definition at line 84 of file p2_fw_direct_mode.h.

5.8.2.30 #define P2_FW_DM_HAL_GET_CFG 0x04

Direct Mode Command: HAL Get Configuration

Definition at line 80 of file p2_fw_direct_mode.h.

5.8.2.31 #define P2_FW_DM_HAL_INIT 0x01

Direct Mode Command: HAL Init

Definition at line 77 of file p2_fw_direct_mode.h.

5.8.2.32 #define P2_FW_DM_HAL_MFC_AUTH 0x07

Direct Mode Command: HAL MIFARE Auth

Definition at line 83 of file p2_fw_direct_mode.h.

5.8.2.33 #define P2_FW_DM_HAL_MFC_AUTH_KEY 0x09

Direct Mode Command: HAL MIFARE Auth with KeyStore

Definition at line 85 of file p2_fw_direct_mode.h.

5.8.2.34 #define P2_FW_DM_HAL_SET_CFG 0x03

Direct Mode Command: HAL Set Configuration

Definition at line 79 of file p2_fw_direct_mode.h.

5.8.2.35 #define P2_FW_DM_HAL_WAIT 0x06

Direct Mode Command: HAL Wait

Definition at line 82 of file p2_fw_direct_mode.h.

5.8.2.36 #define P2_FW_DM_HAL_XCHG 0x02

Direct Mode Command: HAL Exchange

Definition at line 78 of file p2_fw_direct_mode.h.

5.8.2.37 #define P2_FW_DM_INSTR_BYTE P2_FW_CCID_BULK_HEADER + 0x01

Direct Mode CCID Offset: Command Byte

Definition at line 26 of file p2_fw_direct_mode.h.

5.8.2.38 #define P2_FW_DM_KSTOR 0x70

Direct Mode Class: Key Store

Definition at line 59 of file p2_fw_direct_mode.h.

5.8.2.39 #define P2_FW_DM_KSTOR_CHG_KUC 0x0C

Direct Mode Command: KeyStore Change KUC

Definition at line 135 of file p2_fw_direct_mode.h.

5.8.2.40 #define P2_FW_DM_KSTOR_FORMAT_KEY 0x02

Direct Mode Command: KeyStore Format Key

Definition at line 125 of file p2_fw_direct_mode.h.

5.8.2.41 #define P2_FW_DM_KSTOR_GET_CFG_STR 0x0F

Direct Mode Command: KeyStore Get Configuration String

Definition at line 138 of file p2_fw_direct_mode.h.

5.8.2.42 #define P2_FW_DM_KSTOR_GET_CONFIG 0x0B

Direct Mode Command: KeyStore Get Configuration

Definition at line 134 of file p2_fw_direct_mode.h.

5.8.2.43 #define P2_FW_DM_KSTOR_GET_KEY 0x09

Direct Mode Command: KeyStore Get Key

Definition at line 132 of file p2_fw_direct_mode.h.

5.8.2.44 #define P2_FW_DM_KSTOR_GET_KEY_ENTRY 0x08

Direct Mode Command: KeyStore Set Key Entry

Definition at line 131 of file p2_fw_direct_mode.h.

5.8.2.45 #define P2_FW_DM_KSTOR_GET_KUC 0x0D

Direct Mode Command: KeyStore Get KUC

Definition at line 136 of file p2_fw_direct_mode.h.

5.8.2.46 #define P2_FW_DM_KSTOR_INIT 0x01

Direct Mode Command: KeyStore Init

Definition at line 124 of file p2_fw_direct_mode.h.

5.8.2.47 #define P2_FW_DM_KSTOR_SET_CEK 0x06

Direct Mode Command: KeyStore Set CEK

Definition at line 129 of file p2_fw_direct_mode.h.

5.8.2.48 #define P2_FW_DM_KSTOR_SET_CFG_STR 0x0E

Direct Mode Command: KeyStore Set Configuration String

Definition at line 137 of file p2_fw_direct_mode.h.

5.8.2.49 #define P2_FW_DM_KSTOR_SET_CONFIG 0x0A

Direct Mode Command: KeyStore Set Configuration

Definition at line 133 of file p2_fw_direct_mode.h.

5.8.2.50 #define P2_FW_DM_KSTOR_SET_FULL_KEY 0x07

Direct Mode Command: KeyStore Set Full Key Entry

Definition at line 130 of file p2_fw_direct_mode.h.

5.8.2.51 #define P2_FW_DM_KSTOR_SET_KEY 0x03

Direct Mode Command: KeyStore Set Key

Definition at line 126 of file p2_fw_direct_mode.h.

5.8.2.52 #define P2_FW_DM_KSTOR_SET_KEY_POS 0x04

Direct Mode Command: KeyStore KeyPos

Definition at line 127 of file p2_fw_direct_mode.h.

5.8.2.53 #define P2_FW_DM_KSTOR_SET_KUC 0x05

Direct Mode Command: KeyStore Set KUC

Definition at line 128 of file p2_fw_direct_mode.h.

5.8.2.54 #define P2_FW_DM_L3 0xC0

Direct Mode Class: ISO14443-3

Definition at line 55 of file p2_fw_direct_mode.h.

5.8.2.55 #define P2_FW_DM_L3_ACT_CARD 0x07

Direct Mode Command: ISO14443-3 Activate Card

Definition at line 93 of file p2_fw_direct_mode.h.

5.8.2.56 #define P2_FW_DM_L3_ANTICOL 0x05

Direct Mode Command: ISO14443-3 Anticolision

Definition at line 91 of file p2_fw_direct_mode.h.

5.8.2.57 #define P2_FW_DM_L3_GET_SER 0x09

Direct Mode Command: ISO14443-3 Get Serial

Definition at line 95 of file p2_fw_direct_mode.h.

5.8.2.58 #define P2_FW_DM_L3_HLTA 0x04

Direct Mode Command: ISO14443-3 Halt

Definition at line 90 of file p2_fw_direct_mode.h.

5.8.2.59 #define P2_FW_DM_L3_INIT 0x01

Direct Mode Command: ISO14443-3 Init

Definition at line 87 of file p2_fw_direct_mode.h.

5.8.2.60 #define P2_FW_DM_L3_REQA 0x02

Direct Mode Command: ISO14443-3 ReqA

Definition at line 88 of file p2_fw_direct_mode.h.

5.8.2.61 #define P2_FW_DM_L3_SELECT 0x06

Direct Mode Command: ISO14443-3 Select

Definition at line 92 of file p2_fw_direct_mode.h.

5.8.2.62 #define P2_FW_DM_L3_WKUA 0x03

Direct Mode Command: ISO14443-3 WkuA

Definition at line 89 of file p2_fw_direct_mode.h.

5.8.2.63 #define P2_FW_DM_L3_XCHG 0x08

Direct Mode Command: ISO14443-3 Exchange

Definition at line 94 of file p2_fw_direct_mode.h.

5.8.2.64 #define P2_FW_DM_L4 0xE0

Direct Mode Class: ISO14443-4

Definition at line 57 of file p2_fw_direct_mode.h.

5.8.2.65 #define P2_FW_DM_L4_DESELECT 0x04

Direct Mode Command: ISO14443-4 Deselect

Definition at line 106 of file p2_fw_direct_mode.h.

5.8.2.66 #define P2_FW_DM_L4_GET_CFG 0x08

Direct Mode Command: ISO14443-4 Get Configuration

Definition at line 110 of file p2_fw_direct_mode.h.

5.8.2.67 #define P2_FW_DM_L4_INIT 0x01

Direct Mode Command: ISO14443-4 Init

Definition at line 103 of file p2_fw_direct_mode.h.

5.8.2.68 #define P2_FW_DM_L4_PRES_CHECK 0x05

Direct Mode Command: ISO14443-4 Present Check

Definition at line 107 of file p2_fw_direct_mode.h.

5.8.2.69 #define P2_FW_DM_L4_RESET_PROTO 0x03

Direct Mode Command: ISO14443-4 Reset Protocol

Definition at line 105 of file p2_fw_direct_mode.h.

5.8.2.70 #define P2_FW_DM_L4_SET_CFG 0x07

Direct Mode Command: ISO14443-4 Set Configuration

Definition at line 109 of file p2_fw_direct_mode.h.

5.8.2.71 #define P2_FW_DM_L4_SET_PROTO 0x02

Direct Mode Command: ISO14443-4 Set Protocol

Definition at line 104 of file p2_fw_direct_mode.h.

5.8.2.72 #define P2_FW_DM_L4_XCHG 0x06

Direct Mode Command: ISO14443-4 Exchange

Definition at line 108 of file p2_fw_direct_mode.h.

5.8.2.73 #define P2_FW_DM_L4A 0xD0

Direct Mode Class: ISO14443-4A

Definition at line 56 of file p2_fw_direct_mode.h.

5.8.2.74 #define P2_FW_DM_L4A_ACT_CARD 0x04

Direct Mode Command: ISO14443-4A Activate Card

Definition at line 100 of file p2_fw_direct_mode.h.

5.8.2.75 #define P2_FW_DM_L4A_GET_PROTO_PARM 0x05

Direct Mode Command: ISO14443-4A Get Protocol Parameters

Definition at line 101 of file p2_fw_direct_mode.h.

5.8.2.76 #define P2_FW_DM_L4A_INIT 0x01

Direct Mode Command: ISO14443-4A Init

Definition at line 97 of file p2_fw_direct_mode.h.

5.8.2.77 #define P2_FW_DM_L4A_PPS 0x03

Direct Mode Command: ISO14443-4A PPS

Definition at line 99 of file p2_fw_direct_mode.h.

5.8.2.78 #define P2_FW_DM_L4A_RATS 0x02

Direct Mode Command: ISO14443-4A RATS

Definition at line 98 of file p2_fw_direct_mode.h.

5.8.2.79 #define P2_FW_DM_LENGTH_LSB_IN P2_FW_CCID_BULK_-HEADER + 0x02

Direct Mode CCID in Offset: Length LSB

Definition at line 29 of file p2_fw_direct_mode.h.

5.8.2.80 #define P2_FW_DM_LENGTH_LSB_OUT P2_FW_CCID_BULK_-HEADER + 0x04

Direct Mode CCID out Offset: Length LSB

Definition at line 31 of file p2_fw_direct_mode.h.

5.8.2.81 #define P2_FW_DM_LENGTH_MSB_IN P2_FW_CCID_BULK_-HEADER + 0x03

Direct Mode CCID in Offset: Length MSB

Definition at line 30 of file p2_fw_direct_mode.h.

5.8.2.82 #define P2_FW_DM_LENGTH_MSB_OUT P2_FW_CCID_BULK_-HEADER + 0x05

Direct Mode CCID out Offset: Length MSB

Definition at line 32 of file p2_fw_direct_mode.h.

5.8.2.83 #define P2_FW_DM_OFFSET_IN P2_FW_CCID_BULK_HEADER + 0x04

Direct Mode CCID in

Definition at line 34 of file p2_fw_direct_mode.h.

5.8.2.84 #define P2_FW_DM_OFFSET_OUT P2_FW_CCID_BULK_- HEADER + 0x06

Direct Mode CCID out

Definition at line 35 of file p2_fw_direct_mode.h.

5.8.2.85 #define P2_FW_DM_OK 0x80

Direct Mode Status Code: OK

Definition at line 37 of file p2_fw_direct_mode.h.

5.8.2.86 #define P2_FW_DM_RO 0xA0

Direct Mode Class: Reader Operations

Definition at line 53 of file p2_fw_direct_mode.h.

5.8.2.87 #define P2_FW_DM_RO_CONF_OVER 0x04

Direct Mode Command: Configuration Overwrite

Definition at line 65 of file p2_fw_direct_mode.h.

5.8.2.88 #define P2_FW_DM_RO_FIELD_OFF 0x0A

Direct Mode Command: Field Off

Definition at line 72 of file p2_fw_direct_mode.h.

5.8.2.89 #define P2_FW_DM_RO_FIELD_ON 0x09

Direct Mode Command: Field On

Definition at line 71 of file p2_fw_direct_mode.h.

5.8.2.90 #define P2_FW_DM_RO_FIELD_RESET 0x0B

Direct Mode Command: Field Reset

Definition at line 73 of file p2_fw_direct_mode.h.

5.8.2.91 #define P2_FW_DM_RO_GET_CONF 0x0C

Direct Mode Command: Get Configuration

Definition at line 67 of file p2_fw_direct_mode.h.

5.8.2.92 #define P2_FW_DM_RO_GET_STATUS 0x06

Direct Mode Command: Get Status

Definition at line 68 of file p2_fw_direct_mode.h.

5.8.2.93 #define P2_FW_DM_RO_LEDS_OFF 0x01

Direct Mode Command: Led Off

Definition at line 62 of file p2_fw_direct_mode.h.

5.8.2.94 #define P2_FW_DM_RO_LEDS_ON 0x02

Direct Mode Command: Led On

Definition at line 63 of file p2_fw_direct_mode.h.

5.8.2.95 #define P2_FW_DM_RO_READ_REG 0x07

Direct Mode Command: Read Register

Definition at line 69 of file p2_fw_direct_mode.h.

5.8.2.96 #define P2_FW_DM_RO_RESET 0x03

Direct Mode Command: Reset

Definition at line 64 of file p2_fw_direct_mode.h.

5.8.2.97 #define P2_FW_DM_RO_SET_CONF 0x05

Direct Mode Command: Set Configuration

Definition at line 66 of file p2_fw_direct_mode.h.

5.8.2.98 #define P2_FW_DM_RO_SET_PCSC_MODE 0x0D

Direct Mode Command: Set PCSC Mode

Definition at line 74 of file p2_fw_direct_mode.h.

5.8.2.99 #define P2_FW_DM_RO_TEST_MODE 0x0E

Direct Mode Command: Test Mode

Definition at line 75 of file p2_fw_direct_mode.h.

5.8.2.100 #define P2_FW_DM_RO_WRITE_REG 0x08

Direct Mode Command: Write Register

Definition at line 70 of file p2_fw_direct_mode.h.

5.8.2.101 #define P2_FW_DM_STATUS_LSB P2_FW_CCID_BULK_-HEADER + 0x02

Direct Mode CCID Offset: Status Byte LSB

Definition at line 27 of file p2_fw_direct_mode.h.

5.8.2.102 #define P2_FW_DM_STATUS_MSB P2_FW_CCID_BULK_-HEADER + 0x03

Direct Mode CCID Offset: Status Byte MSB

Definition at line 28 of file p2_fw_direct_mode.h.

5.8.2.103 #define P2_FW_DM_XCHG 0xF0

Direct Mode Class: Exchange

Definition at line 58 of file p2_fw_direct_mode.h.

5.8.2.104 #define P2_FW_DM_XCHG_INIT 0x07

Direct Mode Command: MIFARE Init

Definition at line 118 of file p2_fw_direct_mode.h.

5.8.2.105 #define P2_FW_DM_XCHG_L3 0x01

Direct Mode Command: MIFARE Exchange L3

Definition at line 112 of file p2_fw_direct_mode.h.

5.8.2.106 #define P2_FW_DM_XCHG_L4 0x02

Direct Mode Command: MIFARE Exchange L4

Definition at line 113 of file p2_fw_direct_mode.h.

5.8.2.107 #define P2_FW_DM_XCHG_MFC_AUTH 0x05

Direct Mode Command: MIFARE Auth

Definition at line 116 of file p2_fw_direct_mode.h.

5.8.2.108 #define P2_FW_DM_XCHG_MFC_AUTH_KEY 0x06

Direct Mode Command: MIFARE Auth with KeyStore

Definition at line 117 of file p2_fw_direct_mode.h.

5.8.2.109 #define P2_FW_DM_XCHG_PC 0x03

Direct Mode Command: MIFARE Exchange PC

Definition at line 114 of file p2_fw_direct_mode.h.

5.8.2.110 #define P2_FW_DM_XCHG_RAW 0x04

Direct Mode Command: MIFARE Exchange RAW

Definition at line 115 of file p2_fw_direct_mode.h.

5.8.3 Function Documentation

5.8.3.1 Bool p2_fw_dm (uint8_t *message_type,* uint16_t *allowed_cmds*)

Main dispatcher function for direct mode.

Returns

Boolean

Return values

TRUE

Parameters

message_type Type of CCID message recived

allowed_cmds Commands which can be executed

Definition at line 36 of file p2_fw_direct_mode.c.

5.8.3.2 void p2_fw_dm_hal_wait (uint16_t *timeout1,* uint8_t *flags*)

Performs wait.

Returns

Boolean

Return values

TRUE

Parameters

timeout1 Time to wait

flags Time unit

Definition at line 421 of file p2_fw_direct_mode_hal.c.

5.8.3.3 Bool p2_fw_dm_key_store (uint8_t *message_type*)

Performs KeyStore related functions.

Returns

Boolean

Return values

TRUE

Parameters

message_type Type of CCID message recived

Definition at line 53 of file p2_fw_direct_mode_key_store.c.

5.8.3.4 Bool p2_fw_dm_key_store_init (void)

Inits the key store.

Returns

Boolean

Return values

TRUE

Definition at line 42 of file p2_fw_direct_mode_key_store.c.

5.8.3.5 Bool p2_fw_dm_mfc_auth_hal_key_store (uint8_t *message_type*)

Performs authentication by using key store.

Returns

Boolean

Return values

TRUE

Parameters

message_type Type of CCID message recived

Definition at line 533 of file p2_fw_direct_mode_hal.c.

5.8.3.6 Bool p2_fw_dm_xcgh_l4 (uint8_t *message_type*, uint8_t *slot_idx*)

Performs the exchange function for L4.

Returns

Boolean

Return values

TRUE

Parameters

message_type Type of CCID message recived
slot_idx slot to use

Definition at line 121 of file p2_fw_direct_mode.c.

5.8.3.7 Bool p2_fw_key_store_get_key (uint16_t *key_num*, uint16_t *key_ver*, uint8_t * *p_key*, uint8_t *key_len*, uint16_t * *key_type*)

Returns a key store key.

Returns

Boolean

Return values

TRUE

Parameters

key_num key index
key_ver key version
p_key the key
key_len key length
key_type returns the type of the key

Definition at line 351 of file p2_fw_direct_mode_key_store.c.

5.9 Internal Direct Mode

Defines

- #define PH_EXCHANGE_BUFFER_FIRST (PH_EXCHANGE_DEFAULT | PH_EXCHANGE_BUFFERED_BIT)
- #define PH_EXCHANGE_BUFFER_CONT (PH_EXCHANGE_DEFAULT | PH_EXCHANGE_BUFFERED_BIT | PH_EXCHANGE_LEAVE_BUFFER_BIT)
- #define PH_EXCHANGE_BUFFER_LAST (PH_EXCHANGE_DEFAULT | PH_EXCHANGE_LEAVE_BUFFER_BIT)
- #define PHHAL_HW_MFC_KEYA 0x0A
- #define PHHAL_HW_MFC_KEYB 0x0B
- #define PHHAL_HW_MFC_USE_KEYMODIFIER 0x80
- #define PHHAL_HW_CARDTYPE_CURRENT 0x0000
- #define PHHAL_HW_CARDTYPE_ISO14443A 0x0001
- #define PHHAL_HW_CARDTYPE_ISO14443B 0x0002
- #define PH_RC523_MASK_TXBITS 0x07
- #define PH_RC523_MASK_RXALIGN 0x70
- #define PH_RC523_MASK_RXWAIT 0x3F
- #define SELECT_CASCADE_LEVEL_1 0x93
- #define SELECT_CASCADE_LEVEL_2 0x95
- #define SELECT_CASCADE_LEVEL_3 0x97
- #define SINGLE_UID_LENGTH 0x20
- #define PH_RC523_BIT_106KBPS 0x00
- #define PH_RC523_BIT_212KBPS 0x10
- #define PH_RC523_BIT_424KBPS 0x20
- #define PH_RC523_BIT_848KBPS 0x30
- #define PHHAL_HW_TIMING_MODE_OFF 0x0000
- #define PHHAL_HW_TIMING_MODE_FDT 0x0001
- #define PHHAL_HW_TIMING_MODE_COMM 0x0002
- #define PHHAL_HW_RF_DATARATE_106 0x0000
- #define PHHAL_HW_RF_DATARATE_212 0x0001
- #define PHHAL_HW_RF_DATARATE_424 0x0002
- #define PHHAL_HW_RF_DATARATE_848 0x0003
- #define PH_RC523_SERIALSPEED_9600 0xEB
- #define PH_RC523_SERIALSPEED_19200 0xCB
- #define PH_RC523_SERIALSPEED_38400 0xAB
- #define PH_RC523_SERIALSPEED_57600 0x9A
- #define PH_RC523_SERIALSPEED_115200 0x7A
- #define PH_RC523_SERIALSPEED_230400 0x5A
- #define PH_RC523_SERIALSPEED_460800 0x3A
- #define PHHAL_HW_RS232_BITRATE_9600 0x0000
- #define PHHAL_HW_RS232_BITRATE_19200 0x0001
- #define PHHAL_HW_RS232_BITRATE_38400 0x0002
- #define PHHAL_HW_RS232_BITRATE_57600 0x0003

- #define PHHAL_HW_RS232_BITRATE_115200 0x0004
- #define PHHAL_HW_RS232_BITRATE_230400 0x0005
- #define PHHAL_HW_RS232_BITRATE_460800 0x0006
- #define PHHAL_HW_CONFIG_PARITY 0x0000U
- #define PHHAL_HW_CONFIG_TXCRC 0x0001U
- #define PHHAL_HW_CONFIG_RXCRC 0x0002U
- #define PHHAL_HW_CONFIG_TXLASTBITS 0x0003U
- #define PHHAL_HW_CONFIG_RXLASTBITS 0x0004U
- #define PHHAL_HW_CONFIG_RXALIGN 0x0005U
- #define PHHAL_HW_CONFIG_RXDEAFBITS 0x0006U
- #define PHHAL_HW_CONFIG_TXWAIT_US 0x0007U
- #define PHHAL_HW_CONFIG_CLEARBITSAFTERCOLL 0x0008U
- #define PHHAL_HW_CONFIG_TXDATARATE 0x0009U
- #define PHHAL_HW_CONFIG_RXDATARATE 0x000AU
- #define PHHAL_HW_CONFIG_MODINDEX 0x000BU
- #define PHHAL_HW_CONFIG_ASK100 0x000CU
- #define PHHAL_HW_CONFIG_TIMEOUT_VALUE_US 0x000DU
- #define PHHAL_HW_CONFIG_TIMEOUT_VALUE_MS 0x000EU
- #define PHHAL_HW_CONFIG_SUBCARRIER 0x000FU
- #define PHHAL_HW_CONFIG_TIMING_MODE 0x0010U
- #define PHHAL_HW_CONFIG_TIMING_US 0x0011U
- #define PHHAL_HW_CONFIG_TIMING_MS 0x0012U
- #define PHHAL_HW_CONFIG_FIELD_OFF_TIME 0x0013U
- #define PHHAL_HW_CONFIG_FIELD_RECOVERY_TIME 0x0014U
- #define PHHAL_HW_CONFIG_SYMBOL_START 0x0015U
- #define PHHAL_HW_CONFIG_SYMBOL_END 0x0016U
- #define PHHAL_HW_CONFIG_DISABLE_MF_CRYPTO1 0x002EU
- #define PHHAL_HW_CONFIG_ADDITIONAL_INFO 0x002FU
- #define PHHAL_HW_CONFIG_RXBUFFER_STARTPOS 0x0030U
- #define PHHAL_HW_CONFIG_RXBUFFER_BUFSIZE 0x0031U
- #define PHHAL_HW_CONFIG_TXBUFFER_BUFSIZE 0x0032U
- #define PHHAL_HW_CONFIG_TXBUFFER_LENGTH 0x0033U
- #define PHHAL_HW_CONFIG_TXBUFFER 0x0034U
- #define PHHAL_HW_CONFIG_MAX_PRECACHED_BYTES 0x0035U
- #define PHHAL_HW_CONFIG_BAL_CONNECTION 0x0040U
- #define PHHAL_HW_CONFIG_SERIAL_BITRATE 0x0041U
- #define PHHAL_HW_CONFIG_RFRESET_ON_TIMEOUT 0x0050U
- #define PHPAL_I14443P4_PARAM_BLOCKNO 0x0000
- #define PHPAL_I14443P4_PARAM_CID 0x0001
- #define PHPAL_I14443P4_PARAM_NAD 0x0002
- #define PHPAL_I14443P4_PARAM_FWI 0x0003
- #define PHPAL_I14443P4_PARAM_FSI 0x0004
- #define PHPAL_I14443P4_PARAM_MAXRETRYCOUNT 0x0005
- #define P2_FW_DM_CHK_LEN(len, class, ins)

Functions

- Bool p2_fw_direct_mode_xchg (uint8_t message_type)

  Performs exchange with a card in direct mode.

- void p2_fw_dm_hal_switch_config (uint8_t slot_idx)

  Switches reader chip to correct HAL configuration stack for a card.

- Bool p2_fw_dm_xchg_hal (uint8_t message_type, uint8_t slot_idx)

  Performs the exchange function for HAL.

- Bool p2_fw_dm_mfc_auth_hal (uint8_t message_type)

  Performs MIFARE Classic Authentication.

- Bool p2_fw_dm_hal (uint8_t message_type)

  Contains all functions relating to HAL implementation.

- Bool p2_fw_dm_ro (uint8_t message_type)

  Contains all functions relating to reader operations.

- Bool p2_fw_dm_l3 (uint8_t message_type)

  Contains all functions relating to ISO14443 level 3.

- Bool p2_fw_dm_cid (uint8_t message_type)

  Contains all functions relating to Channel ID Managment.

- Bool p2_fw_dm_l4a (uint8_t message_type)

  Contains all functions relating to ISO14443 level 4 Activation.

- Bool p2_fw_dm_l4 (uint8_t message_type)

  Contains all functions relating to ISO14443 level 4.

- Bool p2_fw_dm_contact_card (uint8_t message_type)

  Contains all functions relating to contact cards.

- void p2_fw_dm_send (uint8_t message_type, uint16_t status, uint8_t class, uint8_t cmd, uint16_t pay_len)

  Sends direct mode reply.

- Bool p2_fw_dm_check_if_vaild (uint8_t message_type)

  Performs checks on direct mode message.

- void p2_fw_dm_hal_set_cfg_txdatarate (uint8_t slot_idx)

  Sets the TX Data Rate.

- void p2_fw_dm_hal_set_cfg_rxdatarate (uint8_t slot_idx)

Sets the RX Data Rate.

- Bool p2_fw_dm_check_max_len (uint16_t len)

  Check if max length reached.

- phStatus_t p2_fw_dm_translate_error_code (phcsBfl_Status_t error)

  Translates the old BFL error code to the new BFL error code.

- void p2_fw_dm_hal_set_cfg_timeout (uint8_t slot_idx)

5.9.1 Detailed Description

Defines and declarations for direct mode - internal part - new bfl wrapper

5.9.2 Define Documentation

5.9.2.1 #define P2_FW_DM_CHK_LEN(*len, class, ins*)

Value:

```
if (p2_fw.comm_buff.in_len < (P2_FW_DM_OFFSET_IN + len))
                                                                {


    p2_fw_dm_send (message_type, PH_ERR_LENGTH_ERROR, class, ins, 0);       \
                                                                        r
    eturn FALSE;
                                                                }
```

Definition at line 121 of file p2_fw_direct_mode_int.h.

5.9.2.2 #define PH_EXCHANGE_BUFFER_CONT (PH_EXCHANGE_DEFAULT | PH_EXCHANGE_BUFFERED_BIT | PH_EXCHANGE_LEAVE_BUFFER_BIT)

Middle part of the message

Definition at line 28 of file p2_fw_direct_mode_int.h.

5.9.2.3 #define PH_EXCHANGE_BUFFER_FIRST (PH_EXCHANGE_DEFAULT | PH_EXCHANGE_BUFFERED_BIT)

First part of the message

Definition at line 27 of file p2_fw_direct_mode_int.h.

5.9.2.4 #define PH_EXCHANGE_BUFFER_LAST (PH_EXCHANGE_DEFAULT | PH_EXCHANGE_LEAVE_BUFFER_BIT)

Last part of the message

Definition at line 29 of file p2_fw_direct_mode_int.h.

5.9.2.5 #define PH_RC523_BIT_106KBPS 0x00

Set speed to 106kbps

Definition at line 48 of file p2_fw_direct_mode_int.h.

5.9.2.6 #define PH_RC523_BIT_212KBPS 0x10

Set speed to 212kbps

Definition at line 49 of file p2_fw_direct_mode_int.h.

5.9.2.7 #define PH_RC523_BIT_424KBPS 0x20

Set speed to 424kbps

Definition at line 50 of file p2_fw_direct_mode_int.h.

5.9.2.8 #define PH_RC523_BIT_848KBPS 0x30

Set speed to 848kbps

Definition at line 51 of file p2_fw_direct_mode_int.h.

5.9.2.9 #define PH_RC523_MASK_RXALIGN 0x70

Masks RX alignment

Definition at line 40 of file p2_fw_direct_mode_int.h.

5.9.2.10 #define PH_RC523_MASK_RXWAIT 0x3F

Masks RX Wait Time

Definition at line 41 of file p2_fw_direct_mode_int.h.

5.9.2.11 #define PH_RC523_MASK_TXBITS 0x07

Masks TX bits

Definition at line 39 of file p2_fw_direct_mode_int.h.

5.9.2.12 #define PH_RC523_SERIALSPEED_115200 0x7A

Set Serial Speed to 115200 - register value

Definition at line 66 of file p2_fw_direct_mode_int.h.

5.9.2.13 #define PH_RC523_SERIALSPEED_19200 0xCB

Set Serial Speed to 19200 - register value

Definition at line 63 of file p2_fw_direct_mode_int.h.

5.9.2.14 #define PH_RC523_SERIALSPEED_230400 0x5A

Set Serial Speed to 230400 - register value

Definition at line 67 of file p2_fw_direct_mode_int.h.

5.9.2.15 #define PH_RC523_SERIALSPEED_38400 0xAB

Set Serial Speed to 38400 - register value

Definition at line 64 of file p2_fw_direct_mode_int.h.

5.9.2.16 #define PH_RC523_SERIALSPEED_460800 0x3A

Set Serial Speed to 460800 - register value

Definition at line 68 of file p2_fw_direct_mode_int.h.

5.9.2.17 #define PH_RC523_SERIALSPEED_57600 0x9A

Set Serial Speed to 57600 - register value

Definition at line 65 of file p2_fw_direct_mode_int.h.

5.9.2.18 #define PH_RC523_SERIALSPEED_9600 0xEB

Set Serial Speed to 9600 - register value

Definition at line 62 of file p2_fw_direct_mode_int.h.

5.9.2.19 #define PHHAL_HW_CARDTYPE_CURRENT 0x0000

Unknown Card Type

Definition at line 35 of file p2_fw_direct_mode_int.h.

5.9.2.20 #define PHHAL_HW_CARDTYPE_ISO14443A 0x0001

ISO14443-A Card

Definition at line 36 of file p2_fw_direct_mode_int.h.

5.9.2.21 #define PHHAL_HW_CARDTYPE_ISO14443B 0x0002

ISO14443-B Card

Definition at line 37 of file p2_fw_direct_mode_int.h.

5.9.2.22 #define PHHAL_HW_CONFIG_ADDITIONAL_INFO 0x002FU

Set / Get additional information.

Definition at line 103 of file p2_fw_direct_mode_int.h.

5.9.2.23 #define PHHAL_HW_CONFIG_ASK100 0x000CU

Enable (PH_ON) or disable (PH_OFF) 100% modulation.

Definition at line 91 of file p2_fw_direct_mode_int.h.

5.9.2.24 #define PHHAL_HW_CONFIG_BAL_CONNECTION 0x0040U

Set the BAL connection type. The default value is always PHHAL_HW_BAL_-CONNECTION_RS232.

Definition at line 110 of file p2_fw_direct_mode_int.h.

5.9.2.25 #define PHHAL_HW_CONFIG_CLEARBITSAFTERCOLL 0x0008U

Enable or Disable clearing of bits after coll.

Definition at line 87 of file p2_fw_direct_mode_int.h.

5.9.2.26 #define PHHAL_HW_CONFIG_DISABLE_MF_CRYPTO1 0x002EU

Disable MIFARE(R) Classic Crypto1.

Definition at line 102 of file p2_fw_direct_mode_int.h.

5.9.2.27 #define PHHAL_HW_CONFIG_FIELD_OFF_TIME 0x0013U

Set the field off time for field-reset ([ms]).

Definition at line 98 of file p2_fw_direct_mode_int.h.

5.9.2.28 #define PHHAL_HW_CONFIG_FIELD_RECOVERY_-TIME 0x0014U

Set the field recovery time for field-reset ([ms]).

Definition at line 99 of file p2_fw_direct_mode_int.h.

5.9.2.29 #define PHHAL_HW_CONFIG_MAX_PRECACHED_-BYTES 0x0035U

Configures the max. number of bytes which are precached prior to command execution.

Definition at line 109 of file p2_fw_direct_mode_int.h.

5.9.2.30 #define PHHAL_HW_CONFIG_MODINDEX 0x000BU

Set modulation index (unit and value are hardware-dependent).

Definition at line 90 of file p2_fw_direct_mode_int.h.

5.9.2.31 #define PHHAL_HW_CONFIG_PARITY 0x0000U

Enable or Disable Parity.

Definition at line 79 of file p2_fw_direct_mode_int.h.

5.9.2.32 #define PHHAL_HW_CONFIG_RFRESET_ON_TIMEOUT 0x0050U

Perform an Rf-Reset in case of a timeout (only for phhalHw_Exchange).

Definition at line 112 of file p2_fw_direct_mode_int.h.

5.9.2.33 #define PHHAL_HW_CONFIG_RXALIGN 0x0005U

Set Rx-Aligned Bits.

Definition at line 84 of file p2_fw_direct_mode_int.h.

5.9.2.34 #define PHHAL_HW_CONFIG_RXBUFFER_BUFSIZE 0x0031U

Buffer size of RX buffer set at Init.

Definition at line 105 of file p2_fw_direct_mode_int.h.

5.9.2.35 #define PHHAL_HW_CONFIG_RXBUFFER_STARTPOS 0x0030U

Start position of RX buffer to be used (never changed by hal).

Definition at line 104 of file p2_fw_direct_mode_int.h.

5.9.2.36 #define PHHAL_HW_CONFIG_RXCRC 0x0002U

Enable or Disable RxCrc.

Definition at line 81 of file p2_fw_direct_mode_int.h.

5.9.2.37 #define PHHAL_HW_CONFIG_RXDATARATE 0x000AU

Configure Data-Rate for Reception.

Definition at line 89 of file p2_fw_direct_mode_int.h.

5.9.2.38 #define PHHAL_HW_CONFIG_RXDEAFBITS 0x0006U

Configure Receiver Deaf-Time in ETUs.

Definition at line 85 of file p2_fw_direct_mode_int.h.

5.9.2.39 #define PHHAL_HW_CONFIG_RXLASTBITS 0x0004U

Get number of valid bits of last Rx-Byte.

Definition at line 83 of file p2_fw_direct_mode_int.h.

5.9.2.40 #define PHHAL_HW_CONFIG_SERIAL_BITRATE 0x0041U

Bitrate for serial communication.

Definition at line 111 of file p2_fw_direct_mode_int.h.

5.9.2.41 #define PHHAL_HW_CONFIG_SUBCARRIER 0x000FU

Subcarrier setting for ISO 15693.

Definition at line 94 of file p2_fw_direct_mode_int.h.

5.9.2.42 #define PHHAL_HW_CONFIG_SYMBOL_END 0x0016U

Disable / Set the EOF symbol of a frame.

Definition at line 101 of file p2_fw_direct_mode_int.h.

5.9.2.43 #define PHHAL_HW_CONFIG_SYMBOL_START 0x0015U

Disable / Set the SOF symbol of a frame.

Definition at line 100 of file p2_fw_direct_mode_int.h.

5.9.2.44 #define PHHAL_HW_CONFIG_TIMEOUT_VALUE_MS 0x000EU

Set RC Timeout (in [ms]).

Definition at line 93 of file p2_fw_direct_mode_int.h.

5.9.2.45 #define PHHAL_HW_CONFIG_TIMEOUT_VALUE_US 0x000DU

Set RC Timeout (in [us]).

Definition at line 92 of file p2_fw_direct_mode_int.h.

5.9.2.46 #define PHHAL_HW_CONFIG_TIMING_MODE 0x0010U

Set the timing mode.

Definition at line 95 of file p2_fw_direct_mode_int.h.

5.9.2.47 #define PHHAL_HW_CONFIG_TIMING_MS 0x0012U

Retrieve elapsed time of RC timer ([ms]).

Definition at line 97 of file p2_fw_direct_mode_int.h.

5.9.2.48 #define PHHAL_HW_CONFIG_TIMING_US 0x0011U

Retrieve elapsed time of RC timer ([us]).

Definition at line 96 of file p2_fw_direct_mode_int.h.

5.9.2.49 #define PHHAL_HW_CONFIG_TXBUFFER 0x0034U

Access the TxBuffer at the address defined by PHHAL_HW_CONFIG_-ADDITIONAL_INFO.

Definition at line 108 of file p2_fw_direct_mode_int.h.

5.9.2.50 #define PHHAL_HW_CONFIG_TXBUFFER_BUFSIZE 0x0032U

Buffer size of TX buffer set at Init.

Definition at line 106 of file p2_fw_direct_mode_int.h.

5.9.2.51 #define PHHAL_HW_CONFIG_TXBUFFER_LENGTH 0x0033U

Amount of valid bytes in TX buffer.

Definition at line 107 of file p2_fw_direct_mode_int.h.

5.9.2.52 #define PHHAL_HW_CONFIG_TXCRC 0x0001U

Enable or Disable TxCrc.

Definition at line 80 of file p2_fw_direct_mode_int.h.

5.9.2.53 #define PHHAL_HW_CONFIG_TXDATARATE 0x0009U

Configure Data-Rate for Transmission.

Definition at line 88 of file p2_fw_direct_mode_int.h.

5.9.2.54 #define PHHAL_HW_CONFIG_TXLASTBITS 0x0003U

Set number of valid bits of last Tx-Byte.

Definition at line 82 of file p2_fw_direct_mode_int.h.

5.9.2.55 #define PHHAL_HW_CONFIG_TXWAIT_US 0x0007U

Set TxWait (= time between last RxIrq and Tx of succeeding frame) in microseconds.

Definition at line 86 of file p2_fw_direct_mode_int.h.

5.9.2.56 #define PHHAL_HW_MFC_KEYA 0x0A

MIFARE Classic Key A

Definition at line 31 of file p2_fw_direct_mode_int.h.

5.9.2.57 #define PHHAL_HW_MFC_KEYB 0x0B

MIFARE Classic Key B

Definition at line 32 of file p2_fw_direct_mode_int.h.

5.9.2.58 #define PHHAL_HW_MFC_USE_KEYMODIFIER 0x80

Use key Modifier

Definition at line 33 of file p2_fw_direct_mode_int.h.

5.9.2.59 #define PHHAL_HW_RF_DATARATE_106 0x0000

Set datarate to 106kbps

Definition at line 57 of file p2_fw_direct_mode_int.h.

5.9.2.60 #define PHHAL_HW_RF_DATARATE_212 0x0001

Set datarate to 212kbps

Definition at line 58 of file p2_fw_direct_mode_int.h.

5.9.2.61 #define PHHAL_HW_RF_DATARATE_424 0x0002

Set datarate to 424kbps

Definition at line 59 of file p2_fw_direct_mode_int.h.

5.9.2.62 #define PHHAL_HW_RF_DATARATE_848 0x0003

Set datarate to 848kbps

Definition at line 60 of file p2_fw_direct_mode_int.h.

5.9.2.63 #define PHHAL_HW_RS232_BITRATE_115200 0x0004

Set Serial Speed to 115200 - config value

Definition at line 74 of file p2_fw_direct_mode_int.h.

5.9.2.64 #define PHHAL_HW_RS232_BITRATE_19200 0x0001

Set Serial Speed to 19200 - config value

Definition at line 71 of file p2_fw_direct_mode_int.h.

5.9.2.65 #define PHHAL_HW_RS232_BITRATE_230400 0x0005

Set Serial Speed to 230400 - config value

Definition at line 75 of file p2_fw_direct_mode_int.h.

5.9.2.66 #define PHHAL_HW_RS232_BITRATE_38400 0x0002

Set Serial Speed to 38400 - config value

Definition at line 72 of file p2_fw_direct_mode_int.h.

5.9.2.67 #define PHHAL_HW_RS232_BITRATE_460800 0x0006

Set Serial Speed to 460800 - config value

Definition at line 76 of file p2_fw_direct_mode_int.h.

5.9.2.68 #define PHHAL_HW_RS232_BITRATE_57600 0x0003

Set Serial Speed to 57600 - config value

Definition at line 73 of file p2_fw_direct_mode_int.h.

5.9.2.69 #define PHHAL_HW_RS232_BITRATE_9600 0x0000

Set Serial Speed to 9600 - config value

Definition at line 70 of file p2_fw_direct_mode_int.h.

5.9.2.70 #define PHHAL_HW_TIMING_MODE_COMM 0x0002

Timing mode: Communication Time

Definition at line 55 of file p2_fw_direct_mode_int.h.

5.9.2.71 #define PHHAL_HW_TIMING_MODE_FDT 0x0001

Timing mode: Frame Delay Time

Definition at line 54 of file p2_fw_direct_mode_int.h.

5.9.2.72 #define PHHAL_HW_TIMING_MODE_OFF 0x0000

Timing mode: off

Definition at line 53 of file p2_fw_direct_mode_int.h.

5.9.2.73 #define PHPAL_I14443P4_PARAM_BLOCKNO 0x0000

ISO14443-4 Protocol Parameter: Block Number - config value

Definition at line 114 of file p2_fw_direct_mode_int.h.

5.9.2.74 #define PHPAL_I14443P4_PARAM_CID 0x0001

ISO14443-4 Protocol Parameter: CDI - config value

Definition at line 115 of file p2_fw_direct_mode_int.h.

5.9.2.75 #define PHPAL_I14443P4_PARAM_FSI 0x0004

ISO14443-4 Protocol Parameter: FSI - config value

Definition at line 118 of file p2_fw_direct_mode_int.h.

5.9.2.76 #define PHPAL_I14443P4_PARAM_FWI 0x0003

ISO14443-4 Protocol Parameter: FWI - config value

Definition at line 117 of file p2_fw_direct_mode_int.h.

5.9.2.77 #define PHPAL_I14443P4_PARAM_MAXRETRYCOUNT 0x0005

ISO14443-4 Protocol Parameter: Max Retry Count - config value

Definition at line 119 of file p2_fw_direct_mode_int.h.

5.9.2.78 #define PHPAL_I14443P4_PARAM_NAD 0x0002

ISO14443-4 Protocol Parameter: NAD - config value

Definition at line 116 of file p2_fw_direct_mode_int.h.

5.9.2.79 #define SELECT_CASCADE_LEVEL_1 0x93

Code for cascade level 1

Definition at line 43 of file p2_fw_direct_mode_int.h.

5.9.2.80 #define SELECT_CASCADE_LEVEL_2 0x95

Code for cascade level 2

Definition at line 44 of file p2_fw_direct_mode_int.h.

5.9.2.81 #define SELECT_CASCADE_LEVEL_3 0x97

Code for cascade level 2

Definition at line 45 of file p2_fw_direct_mode_int.h.

5.9.2.82 #define SINGLE_UID_LENGTH 0x20

Uid is of length 4

Definition at line 46 of file p2_fw_direct_mode_int.h.

5.9.3 Function Documentation

5.9.3.1 Bool p2_fw_direct_mode_xchg (uint8_t *message_type*)

Performs exchange with a card in direct mode.

Returns

Boolean

Return values

TRUE

Parameters

message_type Type of CCID message recived

Definition at line 38 of file p2_fw_direct_mode_xchg.c.

5.9.3.2 Bool p2_fw_dm_check_if_vaild (uint8_t *message_type*)

Performs checks on direct mode message.

Returns

Boolean

Return values

TRUE

Parameters

message_type Type of CCID message recived

Definition at line 101 of file p2_fw_direct_mode.c.

5.9.3.3 Bool p2_fw_dm_check_max_len (uint16_t *len*)

Check if max length reached.

Returns

Boolean

Return values

TRUE - length ok

Parameters

len length of the message

Definition at line 386 of file p2_fw_direct_mode.c.

5.9.3.4 Bool p2_fw_dm_cid (uint8_t *message_type*)

Contains all functions relating to Channel ID Managment.

Returns

Boolean

Return values

TRUE

Parameters

message_type Type of CCID message recived

Definition at line 33 of file p2_fw_direct_mode_cid.c.

5.9.3.5 Bool p2_fw_dm_contact_card (uint8_t *message_type*)

Contains all functions relating to contact cards.

Returns

Boolean

Return values

TRUE

Parameters

message_type Type of CCID message recived

Definition at line 33 of file p2_fw_direct_mode_contact_card.c.

5.9.3.6 Bool p2_fw_dm_hal (uint8_t *message_type*)

Contains all functions relating to HAL implementation.

Returns

Boolean

Return values

TRUE

Parameters

message_type Type of CCID message recived

Definition at line 48 of file p2_fw_direct_mode_hal.c.

5.9.3.7 void p2_fw_dm_hal_set_cfg_rxdatarate (uint8_t *slot_idx*)

Sets the RX Data Rate.

Returns

void

Parameters

slot_idx Slot (HAL stack) index

Definition at line 646 of file p2_fw_direct_mode_hal.c.

5.9.3.8 void p2_fw_dm_hal_set_cfg_timeout (uint8_t *slot_idx*)

Definition at line 1055 of file p2_fw_direct_mode_hal.c.

5.9.3.9 void p2_fw_dm_hal_set_cfg_txdatarate (uint8_t *slot_idx*)

Sets the TX Data Rate.

Returns

void

Parameters

slot_idx Slot (HAL stack) index

Definition at line 606 of file p2_fw_direct_mode_hal.c.

5.9.3.10 void p2_fw_dm_hal_switch_config (uint8_t *slot_idx*)

Switches reader chip to correct HAL configuration stack for a card.

Returns

void

Parameters

slot_idx Index of stack to use

Definition at line 387 of file p2_fw_direct_mode_hal.c.

5.9.3.11 Bool p2_fw_dm_l3 (uint8_t *message_type*)

Contains all functions relating to ISO14443 level 3.

Returns

Boolean

Return values

TRUE

Parameters

message_type Type of CCID message recived

Definition at line 35 of file p2_fw_direct_mode_l3.c.

5.9.3.12 Bool p2_fw_dm_l4 (uint8_t *message_type*)

Contains all functions relating to ISO14443 level 4.

Returns

Boolean

Return values

TRUE

Parameters

message_type Type of CCID message recived

Definition at line 44 of file p2_fw_direct_mode_l4.c.

5.9.3.13 Bool p2_fw_dm_l4a (uint8_t *message_type*)

Contains all functions relating to ISO14443 level 4 Activation.

Returns

Boolean

Return values

TRUE

Parameters

message_type Type of CCID message recived

Definition at line 38 of file p2_fw_direct_mode_l4a.c.

5.9.3.14 Bool p2_fw_dm_mfc_auth_hal (uint8_t *message_type*)

Performs MIFARE Classic Authentication.

Returns

Boolean

Return values

TRUE

Parameters

message_type Type of CCID message recived

Definition at line 476 of file p2_fw_direct_mode_hal.c.

5.9.3.15 Bool p2_fw_dm_ro (uint8_t *message_type*)

Contains all functions relating to reader operations.

Returns

Boolean

Return values

TRUE

Parameters

message_type Type of CCID message recived

Definition at line 37 of file p2_fw_direct_mode_ro.c.

5.9.3.16 void p2_fw_dm_send (uint8_t *message_type*, uint16_t *status*, uint8_t *class*, uint8_t *cmd*, uint16_t *pay_len*)

Sends direct mode reply.

Returns

void

Parameters

message_type Type of CCID message recived
status Status code of direct mode command
class Direct Mode Command Class
cmd Direct Mode Command Command
pay_len The reply payload length

Definition at line 372 of file p2_fw_direct_mode.c.

5.9.3.17 phStatus_t p2_fw_dm_translate_error_code (phcsBfl_Status_t *error*)

Translates the old BFL error code to the new BFL error code.

Returns

phStatus_t

Return values

return code

Parameters

error old error

Definition at line 404 of file p2_fw_direct_mode.c.

5.9.3.18 Bool p2_fw_dm_xchg_hal (uint8_t *message_type*, uint8_t *slot_idx*)

Performs the exchange function for HAL.

Returns

Boolean

Return values

TRUE

Parameters

message_type Type of CCID message recived

slot_idx Index of stack to use

Definition at line 210 of file p2_fw_direct_mode.c.

5.10 External Interfaces

Defines

- #define P2_FW_USB_INT_IN_EP 0x81
- #define P2_FW_USB_BULK_OUT_EP 0x05
- #define P2_FW_USB_BULK_IN_EP 0x82

Functions

- void p2_fw_usb_init_usb (void)

  Initializes USB Communication.

- void p2_fw_serial_init_serial (void)

  Initializes Serial (RS232, RS485) Communication.

5.10.1 Detailed Description

Defines and declarations for External Interfaces

5.10.2 Define Documentation

5.10.2.1 #define P2_FW_USB_BULK_IN_EP 0x82

USB Bulk-In Endpoint

Definition at line 27 of file p2_fw_ext.h.

5.10.2.2 #define P2_FW_USB_BULK_OUT_EP 0x05

USB Bulk-Out Endpoint

Definition at line 26 of file p2_fw_ext.h.

5.10.2.3 #define P2_FW_USB_INT_IN_EP 0x81

USB Int-In Endpoint

Definition at line 25 of file p2_fw_ext.h.

5.10.3 Function Documentation

5.10.3.1 void p2_fw_serial_init_serial (void)

Initializes Serial (RS232, RS485) Communication.

Returns

void

Definition at line 41 of file p2_fw_ext_intf_serial.c.

5.10.3.2 void p2_fw_usb_init_usb (void)

Initializes USB Communication.

Returns

void

Definition at line 173 of file p2_fw_ext_intf_usb.c.

5.11 Pegoda 2 Pins

Defines

- #define P2_FW_PINS_DIP_1 0x02000000
- #define P2_FW_PINS_DIP_2 0x04000000
- #define P2_FW_PINS_DIP_3 0x08000000
- #define P2_FW_PINS_DIP_4 0x10000000
- #define P2_FW_PINS_DIP_5 0x00020000
- #define P2_FW_PINS_DIP_6 0x00040000
- #define P2_FW_PINS_DIP_7 0x00080000
- #define P2_FW_PINS_DIP_8 0x00100000
- #define P2_FW_PINS_CFG_1 0x02000000
- #define P2_FW_PINS_CFG_2 0x04000000
- #define P2_FW_PINS_CFG_3 0x10000000
- #define P2_FW_PINS_CFG_4 0x20000000
- #define P2_FW_PINS_ANTENA_BLUE 0x00000080
- #define P2_FW_PINS_ANTENA_GREEN 0x00000040
- #define P2_FW_PINS_ANTENA_RED 0x00000020
- #define P2_FW_PINS_BEEPER 0x00000010
- #define P2_FW_PINS_LEDS_YELLOW_2 0x00000100
- #define P2_FW_PINS_LEDS_YELLOW_3 0x00000080
- #define P2_FW_PINS_LEDS_YELLOW_4 0x00000100

5.11.1 Detailed Description

Defines for the pins on peridot board (NXP Pegoda 2)

5.11.2 Define Documentation

5.11.2.1 #define P2_FW_PINS_ANTENA_BLUE 0x00000080

(P2) the LEDs on Pegoda 2 antena

Definition at line 40 of file p2_fw_peridot_pins.h.

5.11.2.2 #define P2_FW_PINS_ANTENA_GREEN 0x00000040

(P2) the LEDs on Pegoda 2 antena

Definition at line 41 of file p2_fw_peridot_pins.h.

5.11.2.3 #define P2_FW_PINS_ANTENA_RED 0x00000020

(P2) the LEDs on Pegoda 2 antena

Definition at line 42 of file p2_fw_peridot_pins.h.

5.11.2.4 #define P2_FW_PINS_BEEPER 0x00000010

(P2)

Definition at line 44 of file p2_fw_peridot_pins.h.

5.11.2.5 #define P2_FW_PINS_CFG_1 0x02000000

CFG PIN 1

Definition at line 35 of file p2_fw_peridot_pins.h.

5.11.2.6 #define P2_FW_PINS_CFG_2 0x04000000

CFG PIN 2

Definition at line 36 of file p2_fw_peridot_pins.h.

5.11.2.7 #define P2_FW_PINS_CFG_3 0x10000000

CFG PIN 3

Definition at line 37 of file p2_fw_peridot_pins.h.

5.11.2.8 #define P2_FW_PINS_CFG_4 0x20000000

CFG PIN 4

Definition at line 38 of file p2_fw_peridot_pins.h.

5.11.2.9 #define P2_FW_PINS_DIP_1 0x02000000

DIP switch P1.25

Definition at line 25 of file p2_fw_peridot_pins.h.

5.11.2.10 #define P2_FW_PINS_DIP_2 0x04000000

DIP switch P1.26

Definition at line 26 of file p2_fw_peridot_pins.h.

5.11.2.11 #define P2_FW_PINS_DIP_3 0x08000000

DIP switch P1.27

Definition at line 27 of file p2_fw_peridot_pins.h.

5.11.2.12 #define P2_FW_PINS_DIP_4 0x10000000

DIP switch P1.28

Definition at line 28 of file p2_fw_peridot_pins.h.

5.11.2.13 #define P2_FW_PINS_DIP_5 0x00020000

DIP switch P0.17

Definition at line 30 of file p2_fw_peridot_pins.h.

5.11.2.14 #define P2_FW_PINS_DIP_6 0x00040000

DIP switch P0.18

Definition at line 31 of file p2_fw_peridot_pins.h.

5.11.2.15 #define P2_FW_PINS_DIP_7 0x00080000

DIP switch P0.19

Definition at line 32 of file p2_fw_peridot_pins.h.

5.11.2.16 #define P2_FW_PINS_DIP_8 0x00100000

DIP switch P0.20

Definition at line 33 of file p2_fw_peridot_pins.h.

5.11.2.17 #define P2_FW_PINS_LEDS_YELLOW_2 0x00000100

(P2)

Definition at line 46 of file p2_fw_peridot_pins.h.

5.11.2.18 #define P2_FW_PINS_LEDS_YELLOW_3 0x00000080

(P0)

Definition at line 47 of file p2_fw_peridot_pins.h.

5.11.2.19 #define P2_FW_PINS_LEDS_YELLOW_4 0x00000100

(P0)

Definition at line 48 of file p2_fw_peridot_pins.h.

5.12 Timing

Functions

- void p2_fw_timing_init (void)
  Initializes Timing Mode.
- void p2_fw_timing_start (void)
  Starts Timer.
- void p2_fw_timing_stop_cless (int timeout)
  Stops Timer for Contact Less Cards.
- void p2_fw_timing_stop_contac ()
  Stops Timer for Contact Cards.

5.12.1 Detailed Description

Defines and declarations for Timing Mode

5.12.2 Function Documentation

5.12.2.1 void p2_fw_timing_init (void)

Initializes Timing Mode.

Returns

void

Definition at line 42 of file p2_fw_timing_com.c.

5.12.2.2 void p2_fw_timing_start (void)

Starts Timer.

Returns

void

Definition at line 54 of file p2_fw_timing_com.c.

5.12.2.3 void p2_fw_timing_stop_cless (int *timeout*)

Stops Timer for Contact Less Cards.

Returns

void

Definition at line 74 of file p2_fw_timing_com.c.

5.12.2.4 void p2_fw_timing_stop_contac ( )

Stops Timer for Contact Cards.

Returns

void

Definition at line 67 of file p2_fw_timing_com.c.

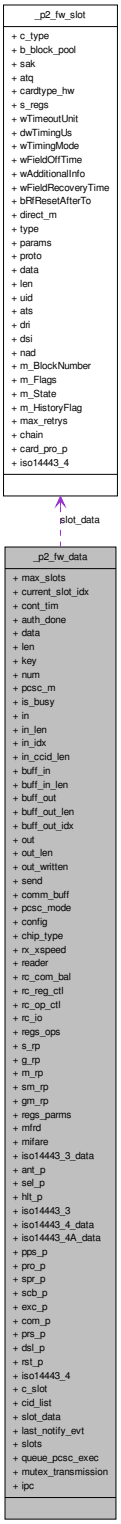
Chapter 6

Data Structure Documentation

6.1 _p2_fw_data Struct Reference

`#include <p2_fw.h>`

Collaboration diagram for _p2_fw_data:



Data Fields

- uint8_t max_slots
- uint8_t current_slot_idx
- uint32_t cont_tim
- struct {
  Bool auth_done
  struct {
  uint8_t data [P2_FW_MAX_PCSC_KEY_LEN]
  uint8_t len
  } key
  uint8_t num
  } pcsc_m

- struct {
  Bool is_busy
  uint8_t in [P2_FW_MAX_COMM_BUFF_SIZE]
  uint32_t in_len
  uint32_t in_idx
  uint32_t in_ccid_len
  uint8_t buff_in [P2_FW_MAX_EXCH_BUFF_SIZE]
  uint32_t buff_in_len
  uint8_t buff_out [P2_FW_MAX_EXCH_BUFF_SIZE]
  uint32_t buff_out_len
  uint32_t buff_out_idx
  uint8_t out [P2_FW_MAX_COMM_BUFF_SIZE]
  uint32_t out_len
  uint32_t out_written
  uint32_t_(∗ send)(uint8_t where, uint8_t ∗what, uint32_t how_much)
  } comm_buff

- uint8_t pcsc_mode
- struct {
  uint8_t config
  uint8_t chip_type
  uint8_t rx_xspeed
  } reader

- struct {
  phcsBflBal_t rc_com_bal
  phcsBflRegCtl_t rc_reg_ctl
  phcsBflOpCtl_t rc_op_ctl
  phcsBflIo_t rc_io
  } regs_ops

- struct {
  phcsBflRegCtl_SetRegParam_t s_rp
  phcsBflRegCtl_GetRegParam_t g_rp
  phcsBflRegCtl_ModRegParam_t m_rp

phcsBflRegCtl_SetMultiRegParam_t sm_rp
phcsBflRegCtl_GetMultiRegParam_t gm_rp
} regs_parms

- struct {
  phcsBflMfRd_t mfrd
  } mifare

- struct {
  phcsBflI3P3A_t iso14443_3_data
  phcsBflI3P3A_AnticollParam_t ant_p
  phcsBflI3P3A_SelectParam_t sel_p
  phcsBflI3P3A_HaltAParam_t hlt_p
  } iso14443_3

- struct {
  phcsBflI3P4_t iso14443_4_data
  phcsBflI3P4AAct_t iso14443_4A_data
  phcsBflI3P4AAct_PpsParam_t pps_p
  phcsBflI3P4_ProtParam_t pro_p
  phcsBflI3P4_SetProtParam_t spr_p
  phcsBflI3P4_SetCbParam_t scb_p
  phcsBflI3P4_ExchangeParam_t exc_p
  phcsBflI3P4_CommParam_t com_p
  phcsBflI3P4_PresCheckParam_t prs_p
  phcsBflI3P4_DeselectParam_t dsl_p
  phcsBflI3P4_ResetProtParam_t rst_p
  } iso14443_4

- struct {
  enum p2_fw_card_type c_slot
  uint8_t cid_list [P2_FW_MAX_SLOTS]
  p2_fw_slot slot_data [P2_FW_MAX_SLOTS+1]
  uint8_t last_notify_evt [P2_FW_MAX_SLOTS+1]
  } slots

- struct {
  xQueueHandle queue_pcsc_exec
  xSemaphoreHandle mutex_transmission
  } ipc

6.1.1 Detailed Description

Definition at line 258 of file p2_fw.h.

6.1.2 Field Documentation

6.1.2.1 phcsBflI3P3A_AnticollParam_t _p2_fw_data::ant_p

Internal structs required by BFL

Definition at line 338 of file p2_fw.h.

6.1.2.2 Bool _p2_fw_data::auth_done

Was authentication done (PCSC extensions)

Definition at line 268 of file p2_fw.h.

6.1.2.3 uint8_t _p2_fw_data::buff_in[P2_FW_MAX_EXCH_BUFF_SIZE]

Internal exchange buffer

Definition at line 288 of file p2_fw.h.

6.1.2.4 uint32_t _p2_fw_data::buff_in_len

Internal exchange buffer length

Definition at line 289 of file p2_fw.h.

6.1.2.5 uint8_t _p2_fw_data::buff_out[P2_FW_MAX_EXCH_BUFF_SIZE]

Internal exchange buffer

Definition at line 291 of file p2_fw.h.

6.1.2.6 uint32_t _p2_fw_data::buff_out_idx

Definition at line 293 of file p2_fw.h.

6.1.2.7 uint32_t _p2_fw_data::buff_out_len

Internal exchange buffer length

Definition at line 292 of file p2_fw.h.

6.1.2.8 enum p2_fw_card_type _p2_fw_data::c_slot

Kind of slots we have

Definition at line 362 of file p2_fw.h.

6.1.2.9 uint8_t _p2_fw_data::chip_type

Type of reader chip

Definition at line 307 of file p2_fw.h.

6.1.2.10 uint8_t _p2_fw_data::cid_list[P2_FW_MAX_SLOTS]

CID list for CID manager

Definition at line 364 of file p2_fw.h.

6.1.2.11 phcsBflI3P4_CommParam_t _p2_fw_data::com_p

Internal structs required by BFL

Definition at line 354 of file p2_fw.h.

6.1.2.12 struct { ... } _p2_fw_data::comm_buff

6.1.2.13 uint8_t _p2_fw_data::config

Reader configuration (DIP switches)

Definition at line 306 of file p2_fw.h.

6.1.2.14 uint32_t _p2_fw_data::cont_tim

Definition at line 264 of file p2_fw.h.

6.1.2.15 uint8_t _p2_fw_data::current_slot_idx

Definition at line 262 of file p2_fw.h.

6.1.2.16 uint8_t _p2_fw_data::data[P2_FW_MAX_PCSC_KEY_LEN]

Key for authentication (PCSC extensions)

Definition at line 272 of file p2_fw.h.

6.1.2.17 phcsBflI3P4_DeselectParam_t _p2_fw_data::dsl_p

Internal structs required by BFL

Definition at line 356 of file p2_fw.h.

6.1.2.18 phcsBflI3P4_ExchangeParam_t _p2_fw_data::exc_p

Internal structs required by BFL

Definition at line 353 of file p2_fw.h.

6.1.2.19 phcsBflRegCtl_GetRegParam_t _p2_fw_data::g_rp

Internal structs required by BFL

Definition at line 323 of file p2_fw.h.

6.1.2.20 phcsBflRegCtl_GetMultiRegParam_t _p2_fw_data::gm_rp

Internal structs required by BFL

Definition at line 326 of file p2_fw.h.

6.1.2.21 phcsBflI3P3A_HaltAParam_t _p2_fw_data::hlt_p

Internal structs required by BFL

Definition at line 340 of file p2_fw.h.

6.1.2.22 uint8_t _p2_fw_data::in[P2_FW_MAX_COMM_BUFF_SIZE]

In communication buffer

Definition at line 283 of file p2_fw.h.

6.1.2.23 uint32_t _p2_fw_data::in_ccid_len

Index for CCID writing to In communication buffer

Definition at line 286 of file p2_fw.h.

6.1.2.24 uint32_t _p2_fw_data::in_idx

Index for writing in to In communication buffer

Definition at line 285 of file p2_fw.h.

6.1.2.25 uint32_t _p2_fw_data::in_len

Length of In communication buffer

Definition at line 284 of file p2_fw.h.

6.1.2.26 struct { ... } _p2_fw_data::ipc

6.1.2.27 Bool _p2_fw_data::is_busy

Are we busy with an operation

Definition at line 281 of file p2_fw.h.

6.1.2.28 struct { ... } _p2_fw_data::iso14443_3

6.1.2.29 phcsBflI3P3A_t _p2_fw_data::iso14443_3_data

Internal structs required by BFL

Definition at line 336 of file p2_fw.h.

6.1.2.30 struct { ... } _p2_fw_data::iso14443_4

6.1.2.31 phcsBflI3P4_t _p2_fw_data::iso14443_4_data

Internal structs required by BFL

Definition at line 345 of file p2_fw.h.

6.1.2.32 phcsBflI3P4AAct_t _p2_fw_data::iso14443_4A_data

Internal structs required by BFL

Definition at line 346 of file p2_fw.h.

6.1.2.33 struct { ... } _p2_fw_data::key

6.1.2.34 uint8_t _p2_fw_data::last_notify_evt[P2_FW_MAX_SLOTS+1]

Last Notify Event for particular slot

Definition at line 368 of file p2_fw.h.

6.1.2.35 uint8_t _p2_fw_data::len

Length of the key for authentication (PCSC extensions)

Definition at line 273 of file p2_fw.h.

6.1.2.36 phcsBflRegCtl_ModRegParam_t _p2_fw_data::m_rp

Internal structs required by BFL

Definition at line 324 of file p2_fw.h.

6.1.2.37 uint8_t _p2_fw_data::max_slots

Number of slots that are in use

Definition at line 260 of file p2_fw.h.

6.1.2.38 phcsBflMfRd_t _p2_fw_data::mfrd

Internal structs required by BFL

Definition at line 331 of file p2_fw.h.

6.1.2.39 struct { ... } _p2_fw_data::mifare

6.1.2.40 xSemaphoreHandle _p2_fw_data::mutex_transmission

Lock for reader chip resources

Definition at line 375 of file p2_fw.h.

6.1.2.41 uint8_t _p2_fw_data::num

Sector to authenticate

Definition at line 276 of file p2_fw.h.

6.1.2.42 uint8_t _p2_fw_data::out[P2_FW_MAX_COMM_BUFF_SIZE]

Out communication buffer

Definition at line 295 of file p2_fw.h.

6.1.2.43 uint32_t _p2_fw_data::out_len

Length of Out communication buffer

Definition at line 296 of file p2_fw.h.

6.1.2.44 uint32_t _p2_fw_data::out_written

How much data was already written to the external interface

Definition at line 297 of file p2_fw.h.

6.1.2.45 struct { ... } _p2_fw_data::pcsc_m

6.1.2.46 uint8_t _p2_fw_data::pcsc_mode

Reader operating mode

Definition at line 302 of file p2_fw.h.

6.1.2.47 phcsBflI3P4AAct_PpsParam_t _p2_fw_data::pps_p

Internal structs required by BFL

Definition at line 348 of file p2_fw.h.

6.1.2.48 phcsBflI3P4_ProtParam_t _p2_fw_data::pro_p

Internal structs required by BFL

Definition at line 350 of file p2_fw.h.

6.1.2.49 phcsBflI3P4_PresCheckParam_t _p2_fw_data::prs_p

Internal structs required by BFL

Definition at line 355 of file p2_fw.h.

6.1.2.50 xQueueHandle _p2_fw_data::queue_pcsc_exec

Queue for communication to p2_fw_task_pcsc_execute task

Definition at line 373 of file p2_fw.h.

6.1.2.51 phcsBflBal_t _p2_fw_data::rc_com_bal

Internal structs required by BFL

Definition at line 314 of file p2_fw.h.

6.1.2.52 phcsBflIo_t _p2_fw_data::rc_io

Internal structs required by BFL

Definition at line 317 of file p2_fw.h.

6.1.2.53 phcsBflOpCtl_t _p2_fw_data::rc_op_ctl

Internal structs required by BFL

Definition at line 316 of file p2_fw.h.

6.1.2.54 phcsBflRegCtl_t _p2_fw_data::rc_reg_ctl

Internal structs required by BFL

Definition at line 315 of file p2_fw.h.

6.1.2.55 struct { ... } _p2_fw_data::reader

6.1.2.56 struct { ... } _p2_fw_data::regs_ops

6.1.2.57 struct { ... } _p2_fw_data::regs_parms

6.1.2.58 phcsBflI3P4_ResetProtParam_t _p2_fw_data::rst_p

Internal structs required by BFL

Definition at line 357 of file p2_fw.h.

6.1.2.59 uint8_t _p2_fw_data::rx_xspeed

RC523 Serial Communication Speed

Definition at line 309 of file p2_fw.h.

6.1.2.60 phcsBflRegCtl_SetRegParam_t _p2_fw_data::s_rp

Internal structs required by BFL

Definition at line 322 of file p2_fw.h.

6.1.2.61 phcsBflI3P4_SetCbParam_t _p2_fw_data::scb_p

Internal structs required by BFL

Definition at line 352 of file p2_fw.h.

6.1.2.62 phcsBflI3P3A_SelectParam_t _p2_fw_data::sel_p

Internal structs required by BFL

Definition at line 339 of file p2_fw.h.

6.1.2.63 uint32_t(* _p2_fw_data::send)(uint8_t where, uint8_t *what, uint32_t how_much)

Interface independent function for sending data

Definition at line 299 of file p2_fw.h.

6.1.2.64 p2_fw_slot _p2_fw_data::slot_data[P2_FW_MAX_SLOTS+1]

Slots

Definition at line 366 of file p2_fw.h.

6.1.2.65 struct { ... } _p2_fw_data::slots

6.1.2.66 phcsBflRegCtl_SetMultiRegParam_t _p2_fw_data::sm_rp

Internal structs required by BFL

Definition at line 325 of file p2_fw.h.

6.1.2.67 phcsBflI3P4_SetProtParam_t _p2_fw_data::spr_p

Internal structs required by BFL

Definition at line 351 of file p2_fw.h.

The documentation for this struct was generated from the following file:

- include/p2_fw.h

6.2 _p2_fw_pcsc_exec Struct Reference

Job parameters for p2_fw_task_pcsc_execute task.

`#include <p2_fw.h>`

Data Fields

- uint8_t slot
- Bool(∗ bottom_half)(uint8_t slot_idx)

6.2.1 Detailed Description

Job parameters for p2_fw_task_pcsc_execute task.

Definition at line 188 of file p2_fw.h.

6.2.2 Field Documentation

6.2.2.1 Bool(∗ _p2_fw_pcsc_exec::bottom_half)(uint8_t slot_idx)

Function to call as bottom half handler

Definition at line 192 of file p2_fw.h.

6.2.2.2 uint8_t _p2_fw_pcsc_exec::slot

Slot number to use in bottom_half handler

Definition at line 190 of file p2_fw.h.

The documentation for this struct was generated from the following file:

- include/p2_fw.h

6.3 _p2_fw_slot Struct Reference

Contains slot private data.

`#include <p2_fw.h>`

Data Fields

- enum p2_fw_card_type c_type
- Bool b_block_pool
- uint8_t sak
- uint8_t atq [2]
- struct {
  uint8_t cardtype_hw
  uint16_t s_regs [P2_FW_PCSC_SHADOW_REGS]
  uint16_t wTimeoutUnit
  uint32_t dwTimingUs
  uint16_t wTimingMode
  uint16_t wFieldOffTime
  uint16_t wAdditionalInfo
  uint16_t wFieldRecoveryTime
  uint16_t bRfResetAfterTo
  } direct_m

- struct {
  uint8_t type
  uint8_t params [P2_FW_PCSC_PROTO_PARAMS_LEN_T1]
  } proto

- struct {
  uint8_t data [P2_FW_MAX_UID]
  uint8_t len
  } uid

- struct {
  uint8_t data [P2_FW_MAX_ATS]
  uint8_t len
  } ats

- struct {
  uint8_t dri
  uint8_t dsi
  uint8_t nad
  uint8_t m_BlockNumber
  uint8_t m_Flags
  uint8_t m_State
  uint8_t m_HistoryFlag
  uint8_t max_retrys
  uint8_t chain
  phcsBflI3P4_ProtParam_t card_pro_p
  } iso14443_4

6.3.1 Detailed Description

Contains slot private data.

Definition at line 198 of file p2_fw.h.

6.3.2 Field Documentation

6.3.2.1 uint8_t _p2_fw_slot::atq[2]

Cards ATQ

Definition at line 205 of file p2_fw.h.

6.3.2.2 struct { ... } _p2_fw_slot::ats

6.3.2.3 Bool _p2_fw_slot::b_block_pool

Should we poll for this card

Definition at line 202 of file p2_fw.h.

6.3.2.4 uint16_t _p2_fw_slot::bRfResetAfterTo

Internal HAL variable

Definition at line 217 of file p2_fw.h.

6.3.2.5 enum p2_fw_card_type _p2_fw_slot::c_type

Type of card in the slot

Definition at line 200 of file p2_fw.h.

6.3.2.6 phcsBflI3P4_ProtParam_t _p2_fw_slot::card_pro_p

Internal BFL ISO14443-4 parameters

Definition at line 254 of file p2_fw.h.

6.3.2.7 uint8_t _p2_fw_slot::cardtype_hw

Internal HAL variable

Definition at line 209 of file p2_fw.h.

6.3.2.8 uint8_t _p2_fw_slot::chain

Are we doing chaining

Definition at line 252 of file p2_fw.h.

6.3.2.9 uint8_t _p2_fw_slot::data[P2_FW_MAX_ATS]

UID

ATS

Definition at line 228 of file p2_fw.h.

6.3.2.10 struct { ... } _p2_fw_slot::direct_m

6.3.2.11 uint8_t _p2_fw_slot::dri

ISO14443-4 DRI Protocol Setting

Definition at line 240 of file p2_fw.h.

6.3.2.12 uint8_t _p2_fw_slot::dsi

ISO14443-4 DSI Protocol Setting

Definition at line 241 of file p2_fw.h.

6.3.2.13 uint32_t _p2_fw_slot::dwTimingUs

Internal HAL variable

Definition at line 212 of file p2_fw.h.

6.3.2.14 struct { ... } _p2_fw_slot::iso14443_4

6.3.2.15 uint8_t _p2_fw_slot::len

UIDs Length

ATSs Length

Definition at line 229 of file p2_fw.h.

6.3.2.16 uint8_t _p2_fw_slot::m_BlockNumber

ISO14443-4 Block Number Protocol Setting

Definition at line 245 of file p2_fw.h.

6.3.2.17 uint8_t _p2_fw_slot::m_Flags

ISO14443-4 Flags Protocol Setting

Definition at line 246 of file p2_fw.h.

6.3.2.18 uint8_t _p2_fw_slot::m_HistoryFlag

ISO14443-4 History Flags Protocol Setting

Definition at line 248 of file p2_fw.h.

6.3.2.19 uint8_t _p2_fw_slot::m_State

ISO14443-4 State Protocol Setting

Definition at line 247 of file p2_fw.h.

6.3.2.20 uint8_t _p2_fw_slot::max_retrys

ISO14443-4 Protocol Number of retries

Definition at line 250 of file p2_fw.h.

6.3.2.21 uint8_t _p2_fw_slot::nad

ISO14443-4 NAD Protocol Setting

Definition at line 243 of file p2_fw.h.

6.3.2.22 uint8_t _p2_fw_slot::params[P2_FW_PCSC_PROTO_PARAMS_LEN_T1]

Protocol parameters

Definition at line 223 of file p2_fw.h.

6.3.2.23 struct { ... } _p2_fw_slot::proto

6.3.2.24 uint16_t _p2_fw_slot::s_regs[P2_FW_PCSC_SHADOW_REGS]

Internal HAL variable

Definition at line 210 of file p2_fw.h.

6.3.2.25 uint8_t _p2_fw_slot::sak

Cards SAK

Definition at line 204 of file p2_fw.h.

6.3.2.26 uint8_t _p2_fw_slot::type

Protocol type (T=0, T=1, T=RAW)

Definition at line 222 of file p2_fw.h.

6.3.2.27 struct { ... } _p2_fw_slot::uid

6.3.2.28 uint16_t _p2_fw_slot::wAdditionalInfo

Internal HAL variable

Definition at line 215 of file p2_fw.h.

6.3.2.29 uint16_t _p2_fw_slot::wFieldOffTime

Internal HAL variable

Definition at line 214 of file p2_fw.h.

6.3.2.30 uint16_t _p2_fw_slot::wFieldRecoveryTime

Internal HAL variable

Definition at line 216 of file p2_fw.h.

6.3.2.31 uint16_t _p2_fw_slot::wTimeoutUnit

Internal HAL variable

Definition at line 211 of file p2_fw.h.

6.3.2.32 uint16_t _p2_fw_slot::wTimingMode

Internal HAL variable

Definition at line 213 of file p2_fw.h.

The documentation for this struct was generated from the following file:

- include/p2_fw.h

6.4 cpot_atr_frame Struct Reference

Data Fields

- uint8_t state
- uint8_t historicalC
- uint8_t TAi
- uint8_t TBi
- uint8_t TCi
- uint8_t TDi
- uint8_t TCK

6.4.1 Detailed Description

Definition at line 887 of file p2_fw_sam_t1.c.

6.4.2 Field Documentation

6.4.2.1 uint8_t cpot_atr_frame::historicalC

Definition at line 890 of file p2_fw_sam_t1.c.

6.4.2.2 uint8_t cpot_atr_frame::state

Definition at line 889 of file p2_fw_sam_t1.c.

6.4.2.3 uint8_t cpot_atr_frame::TAi

Definition at line 891 of file p2_fw_sam_t1.c.

6.4.2.4 uint8_t cpot_atr_frame::TBi

Definition at line 892 of file p2_fw_sam_t1.c.

6.4.2.5 uint8_t cpot_atr_frame::TCi

Definition at line 893 of file p2_fw_sam_t1.c.

6.4.2.6 uint8_t cpot_atr_frame::TCK

Definition at line 895 of file p2_fw_sam_t1.c.

6.4.2.7 uint8_t cpot_atr_frame::TDi

Definition at line 894 of file p2_fw_sam_t1.c.

The documentation for this struct was generated from the following file:

- src/p2_fw_sam_t1.c

6.5 p2_fw_SAM_ctrl_ Struct Reference

Data Fields

- uint8_t mode
- uint8_t data [P2_FW_SAM_MAX_REC_DATALEN]
- uint32_t dataIndex
- uint8_t bitIndex
- uint32_t sendLen
- uint8_t chipType
- uint8_t chipMode
- uint32_t t0
- uint32_t etu
- uint8_t SessionATR [P2_FW_SAM_MAX_ATR_SIZE]
- uint8_t SessionATR_Size
- uint8_t conversion
- uint8_t sendSeqData
- struct {
  uint16_t data [500]
  uint32_t len
  uint32_t dataIndex
  uint8_t bitIndex
  uint8_t parityCount
  } send_data

- uint8_t recExtraGuardTime
- uint32_t bwi
- uint32_t tmpByteWait_time
- uint8_t timingMode

6.5.1 Detailed Description

Definition at line 97 of file p2_fw_sam_t1.c.

6.5.2 Field Documentation

6.5.2.1 uint8_t p2_fw_SAM_ctrl_::bitIndex

Definition at line 103 of file p2_fw_sam_t1.c.

6.5.2.2 uint32_t p2_fw_SAM_ctrl_::bwi

Definition at line 128 of file p2_fw_sam_t1.c.

6.5.2.3 uint8_t p2_fw_SAM_ctrl_::chipMode

Definition at line 108 of file p2_fw_sam_t1.c.

6.5.2.4 uint8_t p2_fw_SAM_ctrl_::chipType

Definition at line 107 of file p2_fw_sam_t1.c.

6.5.2.5 uint8_t p2_fw_SAM_ctrl_::conversion

Definition at line 115 of file p2_fw_sam_t1.c.

6.5.2.6 uint8_t p2_fw_SAM_ctrl_::data[P2_FW_SAM_MAX_REC_DATALEN]

Definition at line 101 of file p2_fw_sam_t1.c.

6.5.2.7 uint16_t p2_fw_SAM_ctrl_::data[500]

Definition at line 120 of file p2_fw_sam_t1.c.

6.5.2.8 uint32_t p2_fw_SAM_ctrl_::dataIndex

Definition at line 102 of file p2_fw_sam_t1.c.

6.5.2.9 uint32_t p2_fw_SAM_ctrl_::etu

Definition at line 110 of file p2_fw_sam_t1.c.

6.5.2.10 uint32_t p2_fw_SAM_ctrl_::len

Definition at line 121 of file p2_fw_sam_t1.c.

6.5.2.11 uint8_t p2_fw_SAM_ctrl_::mode

Definition at line 99 of file p2_fw_sam_t1.c.

6.5.2.12 uint8_t p2_fw_SAM_ctrl_::parityCount

Definition at line 124 of file p2_fw_sam_t1.c.

6.5.2.13 uint8_t p2_fw_SAM_ctrl_::recExtraGuardTime

Definition at line 127 of file p2_fw_sam_t1.c.

6.5.2.14 struct { ... } p2_fw_SAM_ctrl_::send_data

6.5.2.15 uint32_t p2_fw_SAM_ctrl_::sendLen

Definition at line 104 of file p2_fw_sam_t1.c.

6.5.2.16 uint8_t p2_fw_SAM_ctrl_::sendSeqData

Definition at line 116 of file p2_fw_sam_t1.c.

6.5.2.17 uint8_t p2_fw_SAM_ctrl_::SessionATR[P2_FW_SAM_MAX_ATR_SIZE]

Definition at line 112 of file p2_fw_sam_t1.c.

6.5.2.18 uint8_t p2_fw_SAM_ctrl_::SessionATR_Size

Definition at line 113 of file p2_fw_sam_t1.c.

6.5.2.19 uint32_t p2_fw_SAM_ctrl_::t0

Definition at line 109 of file p2_fw_sam_t1.c.

6.5.2.20 uint8_t p2_fw_SAM_ctrl_::timingMode

Definition at line 131 of file p2_fw_sam_t1.c.

6.5.2.21 uint32_t p2_fw_SAM_ctrl_::tmpByteWait_time

Definition at line 129 of file p2_fw_sam_t1.c.

The documentation for this struct was generated from the following file:

- src/p2_fw_sam_t1.c

6.6 sam_t1_param Struct Reference

`#include <p2_fw_sam_t1.h>`

Data Fields

- uint8_t FI_DI
- uint8_t GuardTime
- uint8_t BWI_CWI
- uint8_t ClockStop
- uint8_t IFSC

6.6.1 Detailed Description

Definition at line 31 of file p2_fw_sam_t1.h.

6.6.2 Field Documentation

6.6.2.1 uint8_t sam_t1_param::BWI_CWI

Definition at line 35 of file p2_fw_sam_t1.h.

6.6.2.2 uint8_t sam_t1_param::ClockStop

Definition at line 36 of file p2_fw_sam_t1.h.

6.6.2.3 uint8_t sam_t1_param::FI_DI

Definition at line 33 of file p2_fw_sam_t1.h.

6.6.2.4 uint8_t sam_t1_param::GuardTime

Definition at line 34 of file p2_fw_sam_t1.h.

6.6.2.5 uint8_t sam_t1_param::IFSC

Definition at line 37 of file p2_fw_sam_t1.h.

The documentation for this struct was generated from the following file:

- include/p2_fw_sam_t1.h

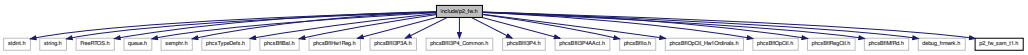
Chapter 7

File Documentation

7.1 include/p2_fw.h File Reference

```
#include <stdint.h>
#include <string.h>
#include <FreeRTOS.h>
#include <queue.h>
#include <semphr.h>
#include <phcsTypeDefs.h>
#include <phcsBflBal.h>
#include <phcsBflHw1Reg.h>
#include <phcsBflI3P3A.h>
#include <phcsBflI3P4_Common.h>
#include <phcsBflI3P4.h>
#include <phcsBflI3P4AAct.h>
#include <phcsBflIo.h>
#include <phcsBflOpCtl_Hw1Ordinals.h>
#include <phcsBflOpCtl.h>
#include <phcsBflRegCtl.h>
#include <phcsBflMfRd.h>
#include <debug_frmwrk.h>
#include <p2_fw_sam_t1.h>
```

Include dependency graph for p2_fw.h:



This graph shows which files directly or indirectly include this file:

Data Structures

- struct _p2_fw_pcsc_exec

  Job parameters for p2_fw_task_pcsc_execute task.

- struct _p2_fw_slot

  Contains slot private data.

- struct _p2_fw_data

Defines

- #define __float32_t_defined
- #define P2_FW_READER_CHIP_RC523 0x01

  Which chip is on the board.

- #define P2_FW_READER_CHIP_SAM 0x02
- #define P2_FW_READER_MODE_NO_SAM 0x00

  SAM operating modes.

- #define P2_FW_READER_MODE_SAM_NON_X 0x01
- #define P2_FW_READER_MODE_SAM_IN_X 0x02
- #define P2_FW_EXT_COM_INTF_USB 0x00

  External communication interface.

- #define P2_FW_EXT_COM_INTF_RS232 0x04
- #define P2_FW_EXT_COM_INTF_RS485 0x08
- #define P2_FW_EXT_COM_INTF_ETHERNET 0x0C
- #define P2_FW_INT_COM_INTF_SPI 0x00

  Internal communcation interface.

- #define P2_FW_INT_COM_INTF_I2C 0x10
- #define P2_FW_INT_COM_INTF_UART 0x20

- #define P2_FW_MODE_PCSC 0x00

  Firmware operating modes.

- #define P2_FW_MODE_DEMO 0x40
- #define P2_FW_MODE_OVER_USER_CFG 0x80
- #define P2_FW_MODE_ENT_SEC_BOOT 0xC0
- #define P2_FW_ERR_AND_NFO_LOOP_DONE 0x00000001

  Error and notifications codes for err_and_nfo_mode.

- #define P2_FW_ERR_AND_NFO_LOOP_UNKNOWN_ERROR 0x00000002
- #define P2_FW_ERR_AND_NFO_LOOP_CHIP_NOT_-DETECTED 0x00000003
- #define P2_FW_ERR_AND_NFO_LOOP_COM_ERROR 0x00000004
- #define P2_FW_ERR_AND_NFO_LOOP_HAL_CAN_NOT_BE_-SET 0x00000005
- #define P2_FW_ERR_AND_NFO_LOOP_HW_ERROR 0x00000006
- #define P2_FW_ERR_AND_NFO_LOOP_OS_ERROR 0x00000007
- #define P2_FW_ERR_AND_NFO_LOOP_ERASE_FAILD 0x00000008
- #define P2_FW_ERR_AND_NFO_LOOP_FLASH_FAILD 0x00000009
- #define P2_FW_ERR_AND_NFO_LOOP_BFL_ERROR 0x0000000A
- #define P2_FW_TMR_START_NOW 0x80
- #define P2_FW_TMR_US 0x00
- #define P2_FW_TMR_MS 0x01
- #define P2_FW_FSI 5
- #define P2_FW_MAX_ATS 32
- #define P2_FW_MAX_UID 10
- #define P2_FW_MAX_SLOTS 15
- #define P2_FW_DEFAULT_SLOTS 1
- #define P2_FW_MAX_EXCH_BUFF_SIZE 256
- #define P2_FW_MAX_COMM_BUFF_SIZE 271
- #define P2_FW_MAX_SLOT_USB_BUFF_SIZE 271
- #define P2_FW_CARD_IN_SLOT 0x01
- #define P2_FW_CARD_OUT_SLOT 0x00
- #define P2_FW_PCSC_PROTO_T0 0x00
- #define P2_FW_PCSC_PROTO_T1 0x01
- #define P2_FW_PCSC_PROTO_RAW 0x02
- #define P2_FW_PCSC_PROTO_PARAMS_LEN_T0 0x05
- #define P2_FW_PCSC_PROTO_PARAMS_LEN_T1 0x07
- #define P2_FW_PCSC_MODE_STANDARD 0x00
- #define P2_FW_PCSC_MODE_DIRECT_MODE 0x01
- #define P2_FW_PCSC_SHADOW_REGS 0x0F
- #define P2_FW_MAX_PCSC_KEY_LEN 32
- #define P2_FW_CFG_MAX_SLOTS 0xCA000001
- #define P2_FW_CFG_CONT_TIMING 0xCB000001
- #define P2_FW_CFG_GET_CONT_TIMING 0xCC000001
- #define P2_FW_CFG_SET_DIP_SWITCHES 0xCD000001

- #define P2_FW_CFG_BOOTLOADER_VERSION 0xEF000001
- #define P2_FW_CFG_BOOTLOADER_ACTIVE 0xEF000002
- #define P2_FW_MAX_RETRY_ISO14443_4A 5
- #define P2_FW_APDU_CLASS P2_FW_CCID_BULK_HEADER
- #define P2_FW_APDU_INS (P2_FW_APDU_CLASS + 1)
- #define P2_FW_APDU_P1 (P2_FW_APDU_INS + 1)
- #define P2_FW_APDU_P2 (P2_FW_APDU_P1 + 1)
- #define P2_FW_APDU_Lc (P2_FW_APDU_P2 + 1)
- #define P2_FW_APDU_CC_EXT_MANAGE_SESSION 0x00
- #define P2_FW_APDU_CC_EXT_TRANS_EXCHANGE 0x01
- #define P2_FW_APDU_CC_EXT_SWITCH_PROTOCOL 0x02
- #define P2_FW_APDU_CC_EXT_INS 0xC2
- #define P2_FW_APDU_GET_DATA_INS 0xCA
- #define P2_FW_APDU_LOAD_KEY_INS 0x82
- #define P2_FW_APDU_G_AUTH_CMD_INS 0x86
- #define P2_FW_APDU_READ_BIN 0xB0
- #define P2_FW_APDU_UPDATE_BIN 0xD6

Typedefs

- typedef float float32_t

  32 bit floating point

- typedef struct _p2_fw_slot p2_fw_slot

  Contains slot private data.

- typedef struct _p2_fw_data p2_fw_data

  Contains reader private data.

- typedef struct _p2_fw_pcsc_exec p2_fw_pcsc_exec

  Contains PCSC extensions private data.

Enumerations

- enum p2_fw_card_type { P2_FW_CARD_TYPE_NONE, P2_FW_CARD_-TYPE_SAM, P2_FW_CARD_TYPE_ISO14443_3A, P2_FW_CARD_TYPE_-ISO14443_4A }

  Possible types of card in a slot.

Functions

- void p2_fw_reader_setup_hardware (void)
- void p2_fw_reader_read_config (void)
- Bool p2_fw_reader_set_up_reader_chip (void)
- void p2_fw_reader_set_up_external_interface ()
- void p2_fw_task_err_and_nfo_loop (void *param)
- void p2_fw_task_demo_mode (void *param)
- void p2_fw_task_pcsc_poll_and_act_loop (void *param)
- void p2_fw_task_pcsc_execute (void *param)
- Bool p2_fw_blf_init (void)
- void p2_fw_blf_set_up_rc_type_a_reading (void)
- Bool p2_fw_blf_reset_reader (void)
- void p2_fw_blf_set_timeout (uint16_t qsec, uint8_t aFlags)
- void p2_fw_blf_set_com_speed (uint8_t dri, uint8_t dsi)
- void p2_fw_blf_change_rc523_baud_rate (uint32_t baudrate)
- void p2_fw_slots_init (void)
- Bool p2_fw_slots_add_new_l4_card (phcsBflI3P4AAct_RatsParam_t *rat_p, uint8_t cid_index, uint8_t sak, uint8_t *atq, uint8_t *uid, uint8_t uid_len)
- Bool p2_fw_slots_add_new_l3_card (uint8_t sak, uint8_t *atq, uint8_t *uid, uint8_t uid_len)
- Bool p2_fw_slots_add_new_sam_card ()
- Bool p2_fw_slots_is_known_l3_card (uint8_t *uid, uint8_t uid_len)
- Bool p2_fw_slots_get_free_cid (uint8_t *cid)
- Bool p2_fw_slots_get_free_slot (uint8_t *slot)
- void p2_fw_slots_free_cid (uint8_t cid)
- void p2_fw_slots_remove_card (uint8_t slot_index)
- void p2_fw_slots_clear_cid_list (void)
- Bool p2_fw_slots_get_atr (uint8_t slot_index, uint8_t *buffer, uint8_t *max_-length)
- uint8_t p2_fw_utils_get_dri (uint8_t ta1)
- uint8_t p2_fw_utils_get_dsi (uint8_t ta1)
- void p2_fw_utils_blink (int count)
- void p2_fw_utils_field_off (void)
- void p2_fw_utils_field_on (uint16_t wFiledRecoveryTime)
- void p2_fw_utils_reg_read (uint8_t addr, uint8_t *val)
- void p2_fw_utils_reg_write (uint8_t addr, uint8_t val)
- void phcsBflBal_Hw1SerCM3Init (phcsBflBal_t *cif, void *comm_params)
- void p2_fw_invoke_error_mode (uint32_t error_code)
- void p2_fw_ccid_xfr_set_busy (void)
- void p2_fw_ccid_xfr_clear_busy (void)
- Bool p2_fw_pcsc_commands (uint8_t slot_idx)
- void p2_fw_pcsc_send_apdu (uint8_t sw1, uint8_t sw2, uint16_t len)
- Bool p2_fw_pcsc_prepare_l3_card (uint8_t slot_idx)
- void p2_fw_flash_erase_config (void)
- Bool p2_fw_flash_read_serial (uint32_t *sernum)
- Bool p2_fw_flash_get_config (uint32_t cfg_id, uint8_t *buff)
- Bool p2_fw_flash_set_config (uint32_t cfg_id, uint8_t *buff)
- void p2_fw_utils_dump_regs (void)
- void UART3_IRQHandler (void)

Variables

- p2_fw_data p2_fw

7.1.1 Define Documentation

7.1.1.1 #define __float32_t_defined

Definition at line 43 of file p2_fw.h.

7.1.1.2 #define P2_FW_APDU_CC_EXT_INS 0xC2

Definition at line 167 of file p2_fw.h.

7.1.1.3 #define P2_FW_APDU_CC_EXT_MANAGE_SESSION 0x00

Definition at line 163 of file p2_fw.h.

7.1.1.4 #define P2_FW_APDU_CC_EXT_SWITCH_PROTOCOL 0x02

Definition at line 165 of file p2_fw.h.

7.1.1.5 #define P2_FW_APDU_CC_EXT_TRANS_EXCHANGE 0x01

Definition at line 164 of file p2_fw.h.

7.1.1.6 #define P2_FW_APDU_CLASS P2_FW_CCID_BULK_HEADER

APDU Class Offset

Definition at line 157 of file p2_fw.h.

7.1.1.7 #define P2_FW_APDU_G_AUTH_CMD_INS 0x86

PCSC Extension: Authenticate Command

Definition at line 170 of file p2_fw.h.

7.1.1.8 #define P2_FW_APDU_GET_DATA_INS 0xCA

PCSC Extension: GetData

Definition at line 168 of file p2_fw.h.

7.1.1.9 #define P2_FW_APDU_INS (P2_FW_APDU_CLASS + 1)

APDU Instruction Offset

Definition at line 158 of file p2_fw.h.

7.1.1.10 #define P2_FW_APDU_Lc (P2_FW_APDU_P2 + 1)

APDU Lc Offset

Definition at line 161 of file p2_fw.h.

7.1.1.11 #define P2_FW_APDU_LOAD_KEY_INS 0x82

PCSC Extension: LoadKey

Definition at line 169 of file p2_fw.h.

7.1.1.12 #define P2_FW_APDU_P1 (P2_FW_APDU_INS + 1)

APDU P1 Offset

Definition at line 159 of file p2_fw.h.

7.1.1.13 #define P2_FW_APDU_P2 (P2_FW_APDU_P1 + 1)

APDU P2 Offset

Definition at line 160 of file p2_fw.h.

7.1.1.14 #define P2_FW_APDU_READ_BIN 0xB0

PCSC Extension: Read Binary

Definition at line 171 of file p2_fw.h.

7.1.1.15 #define P2_FW_APDU_UPDATE_BIN 0xD6

PCSC Extension: Update Binary

Definition at line 172 of file p2_fw.h.

7.1.1.16 #define P2_FW_CARD_IN_SLOT 0x01

Slot is occupied

Definition at line 131 of file p2_fw.h.

7.1.1.17 #define P2_FW_CARD_OUT_SLOT 0x00

Slot is not occupied

Definition at line 132 of file p2_fw.h.

7.1.1.18 #define P2_FW_CFG_BOOTLOADER_ACTIVE 0xEF000002

Definition at line 153 of file p2_fw.h.

7.1.1.19 #define P2_FW_CFG_BOOTLOADER_VERSION 0xEF000001

Definition at line 152 of file p2_fw.h.

7.1.1.20 #define P2_FW_CFG_CONT_TIMING 0xCB000001

Definition at line 149 of file p2_fw.h.

7.1.1.21 #define P2_FW_CFG_GET_CONT_TIMING 0xCC000001

Definition at line 150 of file p2_fw.h.

7.1.1.22 #define P2_FW_CFG_MAX_SLOTS 0xCA000001

Definition at line 148 of file p2_fw.h.

7.1.1.23 #define P2_FW_CFG_SET_DIP_SWITCHES 0xCD000001

Definition at line 151 of file p2_fw.h.

7.1.1.24 #define P2_FW_DEFAULT_SLOTS 1

Definition at line 110 of file p2_fw.h.

7.1.1.25 #define P2_FW_ERR_AND_NFO_LOOP_BFL_ERROR 0x0000000A

BFL Error

Definition at line 98 of file p2_fw.h.

7.1.1.26 #define P2_FW_ERR_AND_NFO_LOOP_CHIP_NOT_-DETECTED 0x00000003

Reader chip not detected

Definition at line 91 of file p2_fw.h.

7.1.1.27 #define P2_FW_ERR_AND_NFO_LOOP_COM_ERROR 0x00000004

Communication error

Definition at line 92 of file p2_fw.h.

7.1.1.28 #define P2_FW_ERR_AND_NFO_LOOP_DONE 0x00000001

Error and notifications codes for err_and_nfo_mode.

Operation completed

Definition at line 89 of file p2_fw.h.

7.1.1.29 #define P2_FW_ERR_AND_NFO_LOOP_ERASE_FAILD 0x00000008

Erase failed

Definition at line 96 of file p2_fw.h.

7.1.1.30 #define P2_FW_ERR_AND_NFO_LOOP_FLASH_FAILD 0x00000009

Flash failed

Definition at line 97 of file p2_fw.h.

7.1.1.31 #define P2_FW_ERR_AND_NFO_LOOP_HAL_CAN_NOT_BE_SET 0x00000005

Error in HAL layer

Definition at line 93 of file p2_fw.h.

7.1.1.32 #define P2_FW_ERR_AND_NFO_LOOP_HW_ERROR 0x00000006

Hardware error

Definition at line 94 of file p2_fw.h.

7.1.1.33 #define P2_FW_ERR_AND_NFO_LOOP_OS_ERROR 0x00000007

Operating system error

Definition at line 95 of file p2_fw.h.

7.1.1.34 #define P2_FW_ERR_AND_NFO_LOOP_UNKNOWN_-ERROR 0x00000002

Unknown error

Definition at line 90 of file p2_fw.h.

7.1.1.35 #define P2_FW_EXT_COM_INTF_ETHERNET 0x0C

External interface is Ethernet

Definition at line 69 of file p2_fw.h.

7.1.1.36 #define P2_FW_EXT_COM_INTF_RS232 0x04

External interface is RS232

Definition at line 67 of file p2_fw.h.

7.1.1.37 #define P2_FW_EXT_COM_INTF_RS485 0x08

External interface is RS485

Definition at line 68 of file p2_fw.h.

7.1.1.38 #define P2_FW_EXT_COM_INTF_USB 0x00

External communication interface.

External interface is USB

Definition at line 66 of file p2_fw.h.

7.1.1.39 #define P2_FW_FSI 5

FSI for PPS

Definition at line 105 of file p2_fw.h.

7.1.1.40 #define P2_FW_INT_COM_INTF_I2C 0x10

Internal interface is I2C

Definition at line 75 of file p2_fw.h.

7.1.1.41 #define P2_FW_INT_COM_INTF_SPI 0x00

Internal communcation interface.

Internal interface is SPI

Definition at line 74 of file p2_fw.h.

7.1.1.42 #define P2_FW_INT_COM_INTF_UART 0x20

Internal interface is UART

Definition at line 76 of file p2_fw.h.

7.1.1.43 #define P2_FW_MAX_ATS 32

Max ATS Length

Definition at line 106 of file p2_fw.h.

7.1.1.44 #define P2_FW_MAX_COMM_BUFF_SIZE 271

Communication buffer size

Definition at line 113 of file p2_fw.h.

7.1.1.45 #define P2_FW_MAX_EXCH_BUFF_SIZE 256

Exchange buffer size

Definition at line 112 of file p2_fw.h.

7.1.1.46 #define P2_FW_MAX_PCSC_KEY_LEN 32

Definition at line 146 of file p2_fw.h.

7.1.1.47 #define P2_FW_MAX_RETRY_ISO14443_4A 5

Number or retries for ISO14443-4 communication

Definition at line 155 of file p2_fw.h.

7.1.1.48 #define P2_FW_MAX_SLOT_USB_BUFF_SIZE 271

Max USB package size

Definition at line 114 of file p2_fw.h.

7.1.1.49 #define P2_FW_MAX_SLOTS 15

Max number of slots

Definition at line 109 of file p2_fw.h.

7.1.1.50 #define P2_FW_MAX_UID 10

Max UID Length

Definition at line 108 of file p2_fw.h.

7.1.1.51 #define P2_FW_MODE_DEMO 0x40

Demo Mode

Definition at line 82 of file p2_fw.h.

7.1.1.52 #define P2_FW_MODE_ENT_SEC_BOOT 0xC0

Enter Secondary Boot Loader

Definition at line 84 of file p2_fw.h.

7.1.1.53 #define P2_FW_MODE_OVER_USER_CFG 0x80

Overwrite Configuration

Definition at line 83 of file p2_fw.h.

7.1.1.54 #define P2_FW_MODE_PCSC 0x00

Firmware operating modes.

PCSC Mode

Definition at line 81 of file p2_fw.h.

7.1.1.55 #define P2_FW_PCSC_MODE_DIRECT_MODE 0x01

Reader mode: Direct Mode

Definition at line 142 of file p2_fw.h.

7.1.1.56 #define P2_FW_PCSC_MODE_STANDARD 0x00

Reader mode: Standard

Definition at line 141 of file p2_fw.h.

7.1.1.57 #define P2_FW_PCSC_PROTO_PARAMS_LEN_T0 0x05

Length of parameters for T=0

Definition at line 138 of file p2_fw.h.

7.1.1.58 #define P2_FW_PCSC_PROTO_PARAMS_LEN_T1 0x07

Length of parameters for T=1

Definition at line 139 of file p2_fw.h.

7.1.1.59 #define P2_FW_PCSC_PROTO_RAW 0x02

Protocol is T=RAW

Definition at line 136 of file p2_fw.h.

7.1.1.60 #define P2_FW_PCSC_PROTO_T0 0x00

Protocol is T=0

Definition at line 134 of file p2_fw.h.

7.1.1.61 #define P2_FW_PCSC_PROTO_T1 0x01

Protocol is T=1

Definition at line 135 of file p2_fw.h.

7.1.1.62 #define P2_FW_PCSC_SHADOW_REGS 0x0F

Number of shadow registers

Definition at line 144 of file p2_fw.h.

7.1.1.63 #define P2_FW_READER_CHIP_RC523 0x01

Which chip is on the board.

RC523

Definition at line 53 of file p2_fw.h.

7.1.1.64 #define P2_FW_READER_CHIP_SAM 0x02

SAM or RX852

Definition at line 54 of file p2_fw.h.

7.1.1.65 #define P2_FW_READER_MODE_NO_SAM 0x00

SAM operating modes.

No SAM present

Definition at line 59 of file p2_fw.h.

7.1.1.66 #define P2_FW_READER_MODE_SAM_IN_X 0x02

SAM is working in X mode

Definition at line 61 of file p2_fw.h.

7.1.1.67 #define P2_FW_READER_MODE_SAM_NON_X 0x01

SAM is working in S mode

Definition at line 60 of file p2_fw.h.

7.1.1.68 #define P2_FW_TMR_MS 0x01

Definition at line 102 of file p2_fw.h.

7.1.1.69 #define P2_FW_TMR_START_NOW 0x80

Definition at line 100 of file p2_fw.h.

7.1.1.70 #define P2_FW_TMR_US 0x00

Definition at line 101 of file p2_fw.h.

7.1.2 Typedef Documentation

7.1.2.1 typedef float float32_t

32 bit floating point

Definition at line 47 of file p2_fw.h.

7.1.2.2 typedef struct _p2_fw_data p2_fw_data

Contains reader private data.

Definition at line 124 of file p2_fw.h.

7.1.2.3 typedef struct _p2_fw_pcsc_exec p2_fw_pcsc_exec

Contains PCSC extensions private data.

Definition at line 129 of file p2_fw.h.

7.1.2.4 typedef struct _p2_fw_slot p2_fw_slot

Contains slot private data.

Definition at line 119 of file p2_fw.h.

7.1.3 Enumeration Type Documentation

7.1.3.1 enum p2_fw_card_type

Possible types of card in a slot.

Enumerator:

P2_FW_CARD_TYPE_NONE No card

P2_FW_CARD_TYPE_SAM Card is SAM

P2_FW_CARD_TYPE_ISO14443_3A Card is ISO14443-3 A Type

P2_FW_CARD_TYPE_ISO14443_4A Card is ISO14443-4 A Type

Definition at line 177 of file p2_fw.h.

7.1.4 Function Documentation

7.1.4.1 void p2_fw_ccid_xfr_clear_busy (void)

Reader completed action

Definition at line 33 of file p2_fw_ccid_xfer.c.

7.1.4.2 void p2_fw_ccid_xfr_set_busy (void)

Reader started to perform action

Definition at line 27 of file p2_fw_ccid_xfer.c.

7.1.4.3 void p2_fw_flash_erase_config (void)

Erases the configuration

Definition at line 50 of file p2_fw_flash_utils.c.

7.1.4.4 Bool p2_fw_flash_get_config (uint32_t *cfg_id*, uint8_t * *buff*)

Reads configuration option cfg_id from flash

Parameters

cfg_id - id of configuration option
buff - buffer to read it in

Returns

Bool - TRUE if success or FALSE if error

Definition at line 77 of file p2_fw_flash_utils.c.

7.1.4.5 Bool p2_fw_flash_read_serial (uint32_t * *sernum*)

Reads the serial number of the reader chip

Parameters

sernum - pointer to uint32_t to store serial number

Returns

Bool - TRUE if success or FALSE if error

Definition at line 58 of file p2_fw_flash_utils.c.

7.1.4.6 Bool p2_fw_flash_set_config (uint32_t *cfg_id*, uint8_t * *buff*)

Writes configuration option to flash

Parameters

cfg_id - id of configuration option
buff - buffer to read it out

Returns

Bool - TRUE if success or FALSE if error

Definition at line 127 of file p2_fw_flash_utils.c.

7.1.4.7 void p2_fw_invoke_error_mode (uint32_t *error_code*)

Puts the reader in error mode

Parameters

error_code - Error code

Definition at line 249 of file p2_fw_main.c.

7.1.4.8 Bool p2_fw_pcsc_commands (uint8_t *slot_idx*)

Execute PCSC extended command on slot

Parameters

slot_index - index of a slot

Returns

Bool - TRUE if success or FALSE if error

Definition at line 29 of file p2_fw_pcsc_ext.c.

7.1.4.9 Bool p2_fw_pcsc_prepare_l3_card (uint8_t *slot_idx*)

Prepare L3 card to execute a PCSC extended command

Parameters

slot_index - index of a slot

Returns

Bool - TRUE if success or FALSE if error

Definition at line 362 of file p2_fw_ccid_xfer.c.

7.1.4.10 void p2_fw_pcsc_send_apdu (uint8_t *sw1*, uint8_t *sw2*, uint16_t *len*)

Send APDU to the client

Parameters

sw1 - SW 1 parameter

sw2 - SW 2 parameter

le - length of whole command

Definition at line 352 of file p2_fw_ccid_xfer.c.

7.1.4.11 void p2_fw_utils_dump_regs (void)

Definition at line 207 of file p2_fw_utils.c.

7.1.4.12 void phcsBflBal_Hw1SerCM3Init (phcsBflBal_t * *cif*, void * *comm_params*)

Definition at line 54 of file phcsBflBal_Hw1SerCM3.c.

7.1.4.13 void UART3_IRQHandler (void)

Definition at line 238 of file p2_fw_ext_intf_serial.c.

7.1.5 Variable Documentation

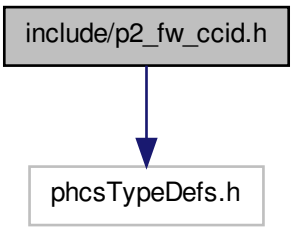
7.1.5.1 p2_fw_data p2_fw

Definition at line 35 of file p2_fw_main.c.

7.2 include/p2_fw_ccid.h File Reference

```
#include <phcsTypeDefs.h>
```

Include dependency graph for p2_fw_ccid.h:



This graph shows which files directly or indirectly include this file:

Defines

- #define P2_FW_CCID_BULK_HEADER 0x0A

Block CCID Commands

- #define P2_FW_CCID_BULK_OUT_REQ_ICCPOWERON 0x62
- #define P2_FW_CCID_BULK_OUT_REQ_ICCPOWEROFF 0x63

- #define P2_FW_CCID_BULK_OUT_REQ_GETSLOTSTATUS 0x65
- #define P2_FW_CCID_BULK_OUT_REQ_XFRBLOCK 0x6F
- #define P2_FW_CCID_BULK_OUT_REQ_GETPARAMETERS 0x6C
- #define P2_FW_CCID_BULK_OUT_REQ_SETPARAMETERS 0x61
- #define P2_FW_CCID_BULK_OUT_REQ_ESCAPE 0x6B
- #define P2_FW_CCID_INT_IN_NOTIFY_SLOT_CHANGE 0x50

Block CCID Replays

- #define P2_FW_CCID_RDR_TO_PC_SLOT_DATA_BLOCK 0x80
- #define P2_FW_CCID_RDR_TO_PC_SLOT_STATUS 0x81
- #define P2_FW_CCID_RDR_TO_PC_PARAMETERS 0x82
- #define P2_FW_CCID_RDR_TO_PC_ESCAPE 0x83

Block CCID Error Defines

- #define P2_FW_CCID_STATUS_CMD_FAILED 0x40
- #define P2_FW_CCID_ERROR_SLOT_BUSY 0xE0
- #define P2_FW_CCID_ERROR_SLOT_NOT_EXIST 0x05
- #define P2_FW_CCID_ERROR_SLOT_ICC_MUTE 0xFE
- #define P2_FW_CCID_ERROR_SLOT_XFR_OVERRUN 0xFC
- #define P2_FW_CCID_ERROR_SLOT_CMD_NOT_SUPPORTED 0x00
- #define P2_FW_CCID_ERROR_SLOT_HW_ERROR 0xFB

Block CCID Header Indexes

- #define P2_FW_CCID_HEADER_MESSAGE_TYPE 0x00
- #define P2_FW_CCID_HEADER_LENGTH_BYTE_1 0x01
- #define P2_FW_CCID_HEADER_LENGTH_BYTE_2 0x02
- #define P2_FW_CCID_HEADER_LENGTH_BYTE_3 0x03
- #define P2_FW_CCID_HEADER_LENGTH_BYTE_4 0x04
- #define P2_FW_CCID_HEADER_SLOT 0x05
- #define P2_FW_CCID_HEADER_SEQ 0x06
- #define P2_FW_CCID_HEADER_MSG_BYTE_1 0x07
- #define P2_FW_CCID_HEADER_MSG_BYTE_2 0x08
- #define P2_FW_CCID_HEADER_MSG_BYTE_3 0x09

Functions

- void p2_fw_ccid_top_half_dispatch (void)

  Top level ISR Dispatcher.

- Bool p2_fw_ccid_check_header (uint8_t message_type)

  Checks if the CCID header is correct.

- Bool p2_fw_ccid_send_apdu (uint8_t *payload, uint8_t payload_len, uint8_t sw1, uint8_t sw2)

  Sends APDU with payload of payload_len and SW1 and SW2.

- void p2_fw_ccid_send_data (uint8_t message_type, uint8_t byte_1, uint8_t byte_2, uint8_t byte_3)

Sends the CCID message.

- void p2_fw_ccid_xfr_block_top_half (void)
  Top half (ISR) function for transfer command.
- Bool p2_fw_ccid_xfr_block_bottom_half (uint8_t slot_idx)
  Bottom half function for transfer command.
- void p2_fw_ccid_get_slot_status_top_half (void)
  Top half (ISR) function for get status command.
- Bool p2_fw_ccid_get_slot_status_bottom_half (uint8_t slot_idx)
  Bottom half function for get status command.
- void p2_fw_ccid_icc_power_on_top_half (void)
  Top half (ISR) function for power on command.
- Bool p2_fw_ccid_icc_power_on_bottom_half (uint8_t slot_idx)
  Bottom half function for power on command.
- void p2_fw_ccid_icc_power_off_top_half (void)
  Top half (ISR) function for power off command.
- Bool p2_fw_ccid_icc_power_off_bottom_half (uint8_t slot_idx)
  Bottom half function for power off command.
- void p2_fw_ccid_get_parameters_top_half (void)
  Top half (ISR) function for get parameters command.
- Bool p2_fw_ccid_get_parameters_bottom_half (uint8_t slot_idx)
  Bottom half function for get parameters command.
- void p2_fw_ccid_set_parameters_top_half (void)
  Top half (ISR) function for set parameters command.
- Bool p2_fw_ccid_set_parameters_bottom_half (uint8_t slot_idx)
  Bottom half function for set parameters command.
- void p2_fw_ccid_escape_top_half (void)
  Top half (ISR) function for escape command.
- Bool p2_fw_ccid_escape_bottom_half (uint8_t slot_idx)
  Bottom half function for Escape command.
- Bool p2_fw_ccid_send_notify (uint8_t slot_idx)
  Sends slot change notify event.

- void p2_fw_ccid_xfr_set_busy (void)

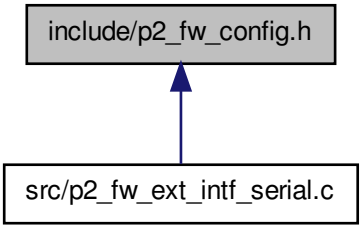
  Set the slot to busy (receiving and processing data).

- void p2_fw_ccid_xfr_clear_busy (void)

  Set the slot to not busy (receiving and processing data).

7.3 include/p2_fw_config.h File Reference

This graph shows which files directly or indirectly include this file:



Defines

- #define P2_FW_CONFIG_RC523_UART_PORT 1
- #define P2_FW_CONFIG_DEBUG_PORT 0
- #define P2_FW_CONFIG_COMM_SER_232_PORT 3
- #define P2_FW_CONFIG_COMM_SER_485_PORT 1

7.4 include/p2_fw_direct_mode.h File Reference

This graph shows which files directly or indirectly include this file:

Defines

- #define P2_FW_DM_CLASS_BYTE P2_FW_CCID_BULK_HEADER
- #define P2_FW_DM_INSTR_BYTE P2_FW_CCID_BULK_HEADER + 0x01
- #define P2_FW_DM_STATUS_LSB P2_FW_CCID_BULK_HEADER + 0x02
- #define P2_FW_DM_STATUS_MSB P2_FW_CCID_BULK_HEADER + 0x03
- #define P2_FW_DM_LENGTH_LSB_IN P2_FW_CCID_BULK_HEADER + 0x02
- #define P2_FW_DM_LENGTH_MSB_IN P2_FW_CCID_BULK_HEADER + 0x03
- #define P2_FW_DM_LENGTH_LSB_OUT P2_FW_CCID_BULK_HEADER + 0x04
- #define P2_FW_DM_LENGTH_MSB_OUT P2_FW_CCID_BULK_HEADER + 0x05
- #define P2_FW_DM_OFFSET_IN P2_FW_CCID_BULK_HEADER + 0x04
- #define P2_FW_DM_OFFSET_OUT P2_FW_CCID_BULK_HEADER + 0x06
- #define P2_FW_DM_OK 0x80
- #define P2_FW_DM_FAILD 0xF0
- #define P2_FW_DM_ALLOWED_CMDS_ALL 0xFFFF
- #define P2_FW_DM_ALLOWED_CMDS_RO 0x0001
- #define P2_FW_DM_ALLOWED_CMDS_HAL 0x0002
- #define P2_FW_DM_ALLOWED_CMDS_L3 0x0008
- #define P2_FW_DM_ALLOWED_CMDS_L4A 0x0010
- #define P2_FW_DM_ALLOWED_CMDS_L4 0x0020
- #define P2_FW_DM_ALLOWED_CMDS_XCHG 0x0040
- #define P2_FW_DM_ALLOWED_CMDS_CID 0x0080
- #define P2_FW_DM_ALLOWED_CMDS_KSTOR 0x0100
- #define P2_FW_DM_ALLOWED_CMDS_CONTACT_CARD 0x0200
- #define P2_FW_DM_CID 0x90
- #define P2_FW_DM_RO 0xA0
- #define P2_FW_DM_HAL 0xB0
- #define P2_FW_DM_L3 0xC0
- #define P2_FW_DM_L4A 0xD0
- #define P2_FW_DM_L4 0xE0
- #define P2_FW_DM_XCHG 0xF0
- #define P2_FW_DM_KSTOR 0x70
- #define P2_FW_DM_CONTACT_CARD 0x80
- #define P2_FW_DM_RO_LEDS_OFF 0x01
- #define P2_FW_DM_RO_LEDS_ON 0x02
- #define P2_FW_DM_RO_RESET 0x03
- #define P2_FW_DM_RO_CONF_OVER 0x04
- #define P2_FW_DM_RO_SET_CONF 0x05
- #define P2_FW_DM_RO_GET_CONF 0x0C
- #define P2_FW_DM_RO_GET_STATUS 0x06
- #define P2_FW_DM_RO_READ_REG 0x07
- #define P2_FW_DM_RO_WRITE_REG 0x08
- #define P2_FW_DM_RO_FIELD_ON 0x09

- #define P2_FW_DM_RO_FIELD_OFF 0x0A
- #define P2_FW_DM_RO_FIELD_RESET 0x0B
- #define P2_FW_DM_RO_SET_PCSC_MODE 0x0D
- #define P2_FW_DM_RO_TEST_MODE 0x0E
- #define P2_FW_DM_HAL_INIT 0x01
- #define P2_FW_DM_HAL_XCHG 0x02
- #define P2_FW_DM_HAL_SET_CFG 0x03
- #define P2_FW_DM_HAL_GET_CFG 0x04
- #define P2_FW_DM_HAL_APP_PROT_SET 0x05
- #define P2_FW_DM_HAL_WAIT 0x06
- #define P2_FW_DM_HAL_MFC_AUTH 0x07
- #define P2_FW_DM_HAL_EXEC_CMD 0x08
- #define P2_FW_DM_HAL_MFC_AUTH_KEY 0x09
- #define P2_FW_DM_L3_INIT 0x01
- #define P2_FW_DM_L3_REQA 0x02
- #define P2_FW_DM_L3_WKUA 0x03
- #define P2_FW_DM_L3_HLTA 0x04
- #define P2_FW_DM_L3_ANTICOL 0x05
- #define P2_FW_DM_L3_SELECT 0x06
- #define P2_FW_DM_L3_ACT_CARD 0x07
- #define P2_FW_DM_L3_XCHG 0x08
- #define P2_FW_DM_L3_GET_SER 0x09
- #define P2_FW_DM_L4A_INIT 0x01
- #define P2_FW_DM_L4A_RATS 0x02
- #define P2_FW_DM_L4A_PPS 0x03
- #define P2_FW_DM_L4A_ACT_CARD 0x04
- #define P2_FW_DM_L4A_GET_PROTO_PARM 0x05
- #define P2_FW_DM_L4_INIT 0x01
- #define P2_FW_DM_L4_SET_PROTO 0x02
- #define P2_FW_DM_L4_RESET_PROTO 0x03
- #define P2_FW_DM_L4_DESELECT 0x04
- #define P2_FW_DM_L4_PRES_CHECK 0x05
- #define P2_FW_DM_L4_XCHG 0x06
- #define P2_FW_DM_L4_SET_CFG 0x07
- #define P2_FW_DM_L4_GET_CFG 0x08
- #define P2_FW_DM_XCHG_L3 0x01
- #define P2_FW_DM_XCHG_L4 0x02
- #define P2_FW_DM_XCHG_PC 0x03
- #define P2_FW_DM_XCHG_RAW 0x04
- #define P2_FW_DM_XCHG_MFC_AUTH 0x05
- #define P2_FW_DM_XCHG_MFC_AUTH_KEY 0x06
- #define P2_FW_DM_XCHG_INIT 0x07
- #define P2_FW_DM_CID_GET_FREE 0x01
- #define P2_FW_DM_CID_FREE 0x02
- #define P2_FW_DM_CID_INIT 0x03
- #define P2_FW_DM_KSTOR_INIT 0x01

- #define P2_FW_DM_KSTOR_FORMAT_KEY 0x02
- #define P2_FW_DM_KSTOR_SET_KEY 0x03
- #define P2_FW_DM_KSTOR_SET_KEY_POS 0x04
- #define P2_FW_DM_KSTOR_SET_KUC 0x05
- #define P2_FW_DM_KSTOR_SET_CEK 0x06
- #define P2_FW_DM_KSTOR_SET_FULL_KEY 0x07
- #define P2_FW_DM_KSTOR_GET_KEY_ENTRY 0x08
- #define P2_FW_DM_KSTOR_GET_KEY 0x09
- #define P2_FW_DM_KSTOR_SET_CONFIG 0x0A
- #define P2_FW_DM_KSTOR_GET_CONFIG 0x0B
- #define P2_FW_DM_KSTOR_CHG_KUC 0x0C
- #define P2_FW_DM_KSTOR_GET_KUC 0x0D
- #define P2_FW_DM_KSTOR_SET_CFG_STR 0x0E
- #define P2_FW_DM_KSTOR_GET_CFG_STR 0x0F
- #define P2_FW_DM_CONTACTCARD_ACTIVATE_CARD 0x01
- #define P2_FW_DM_CONTACTCARD_COLD_RESET 0x02
- #define P2_FW_DM_CONTACTCARD_WARM_RESET 0x03
- #define P2_FW_DM_CONTACTCARD_CLOCK_STOP 0x04
- #define P2_FW_DM_CONTACTCARD_CLOCK_START 0x05
- #define P2_FW_DM_CONTACTCARD_DEACTIVATE_CARD 0x06
- #define P2_FW_DM_CONTACTCARD_PRESENCE_CHECK 0x07
- #define P2_FW_DM_CONTACTCARD_TRANSMIT_DATA 0x08
- #define P2_FW_DM_CONTACTCARD_PPS 0x09

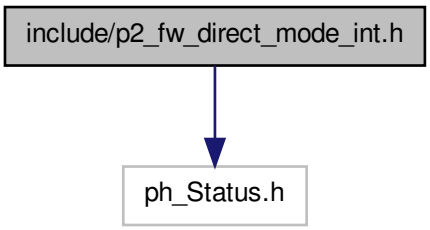
Functions

- Bool p2_fw_dm (uint8_t message_type, uint16_t allowed_cmds)
  Main dispatcher function for direct mode.
- Bool p2_fw_dm_xcgh_l4 (uint8_t message_type, uint8_t slot_idx)
  Performs the exchange function for L4.
- Bool p2_fw_dm_key_store (uint8_t message_type)
  Performs KeyStore related functions.
- Bool p2_fw_dm_key_store_init (void)
  Inits the key store.
- Bool p2_fw_dm_mfc_auth_hal_key_store (uint8_t message_type)
  Performs authentication by using key store.
- void p2_fw_dm_hal_wait (uint16_t timeout1, uint8_t flags)
  Performs wait.
- Bool p2_fw_key_store_get_key (uint16_t key_num, uint16_t key_ver, uint8_t *p_key, uint8_t key_len, uint16_t *key_type)
  Returns a key store key.

7.5 include/p2_fw_direct_mode_int.h File Reference

```
#include <ph_Status.h>
```

Include dependency graph for p2_fw_direct_mode_int.h:



This graph shows which files directly or indirectly include this file:

Defines

- #define PH_EXCHANGE_BUFFER_FIRST (PH_EXCHANGE_DEFAULT | PH_EXCHANGE_BUFFERED_BIT)
- #define PH_EXCHANGE_BUFFER_CONT (PH_EXCHANGE_DEFAULT | PH_EXCHANGE_BUFFERED_BIT | PH_EXCHANGE_LEAVE_BUFFER_-BIT)
- #define PH_EXCHANGE_BUFFER_LAST (PH_EXCHANGE_DEFAULT | PH_EXCHANGE_LEAVE_BUFFER_BIT)
- #define PHHAL_HW_MFC_KEYA 0x0A
- #define PHHAL_HW_MFC_KEYB 0x0B
- #define PHHAL_HW_MFC_USE_KEYMODIFIER 0x80
- #define PHHAL_HW_CARDTYPE_CURRENT 0x0000
- #define PHHAL_HW_CARDTYPE_ISO14443A 0x0001
- #define PHHAL_HW_CARDTYPE_ISO14443B 0x0002
- #define PH_RC523_MASK_TXBITS 0x07
- #define PH_RC523_MASK_RXALIGN 0x70
- #define PH_RC523_MASK_RXWAIT 0x3F
- #define SELECT_CASCADE_LEVEL_1 0x93

- #define SELECT_CASCADE_LEVEL_2 0x95
- #define SELECT_CASCADE_LEVEL_3 0x97
- #define SINGLE_UID_LENGTH 0x20
- #define PH_RC523_BIT_106KBPS 0x00
- #define PH_RC523_BIT_212KBPS 0x10
- #define PH_RC523_BIT_424KBPS 0x20
- #define PH_RC523_BIT_848KBPS 0x30
- #define PHHAL_HW_TIMING_MODE_OFF 0x0000
- #define PHHAL_HW_TIMING_MODE_FDT 0x0001
- #define PHHAL_HW_TIMING_MODE_COMM 0x0002
- #define PHHAL_HW_RF_DATARATE_106 0x0000
- #define PHHAL_HW_RF_DATARATE_212 0x0001
- #define PHHAL_HW_RF_DATARATE_424 0x0002
- #define PHHAL_HW_RF_DATARATE_848 0x0003
- #define PH_RC523_SERIALSPEED_9600 0xEB
- #define PH_RC523_SERIALSPEED_19200 0xCB
- #define PH_RC523_SERIALSPEED_38400 0xAB
- #define PH_RC523_SERIALSPEED_57600 0x9A
- #define PH_RC523_SERIALSPEED_115200 0x7A
- #define PH_RC523_SERIALSPEED_230400 0x5A
- #define PH_RC523_SERIALSPEED_460800 0x3A
- #define PHHAL_HW_RS232_BITRATE_9600 0x0000
- #define PHHAL_HW_RS232_BITRATE_19200 0x0001
- #define PHHAL_HW_RS232_BITRATE_38400 0x0002
- #define PHHAL_HW_RS232_BITRATE_57600 0x0003
- #define PHHAL_HW_RS232_BITRATE_115200 0x0004
- #define PHHAL_HW_RS232_BITRATE_230400 0x0005
- #define PHHAL_HW_RS232_BITRATE_460800 0x0006
- #define PHHAL_HW_CONFIG_PARITY 0x0000U
- #define PHHAL_HW_CONFIG_TXCRC 0x0001U
- #define PHHAL_HW_CONFIG_RXCRC 0x0002U
- #define PHHAL_HW_CONFIG_TXLASTBITS 0x0003U
- #define PHHAL_HW_CONFIG_RXLASTBITS 0x0004U
- #define PHHAL_HW_CONFIG_RXALIGN 0x0005U
- #define PHHAL_HW_CONFIG_RXDEAFBITS 0x0006U
- #define PHHAL_HW_CONFIG_TXWAIT_US 0x0007U
- #define PHHAL_HW_CONFIG_CLEARBITSAFTERCOLL 0x0008U
- #define PHHAL_HW_CONFIG_TXDATARATE 0x0009U
- #define PHHAL_HW_CONFIG_RXDATARATE 0x000AU
- #define PHHAL_HW_CONFIG_MODINDEX 0x000BU
- #define PHHAL_HW_CONFIG_ASK100 0x000CU
- #define PHHAL_HW_CONFIG_TIMEOUT_VALUE_US 0x000DU
- #define PHHAL_HW_CONFIG_TIMEOUT_VALUE_MS 0x000EU
- #define PHHAL_HW_CONFIG_SUBCARRIER 0x000FU
- #define PHHAL_HW_CONFIG_TIMING_MODE 0x0010U
- #define PHHAL_HW_CONFIG_TIMING_US 0x0011U

- #define PHHAL_HW_CONFIG_TIMING_MS 0x0012U
- #define PHHAL_HW_CONFIG_FIELD_OFF_TIME 0x0013U
- #define PHHAL_HW_CONFIG_FIELD_RECOVERY_TIME 0x0014U
- #define PHHAL_HW_CONFIG_SYMBOL_START 0x0015U
- #define PHHAL_HW_CONFIG_SYMBOL_END 0x0016U
- #define PHHAL_HW_CONFIG_DISABLE_MF_CRYPTO1 0x002EU
- #define PHHAL_HW_CONFIG_ADDITIONAL_INFO 0x002FU
- #define PHHAL_HW_CONFIG_RXBUFFER_STARTPOS 0x0030U
- #define PHHAL_HW_CONFIG_RXBUFFER_BUFSIZE 0x0031U
- #define PHHAL_HW_CONFIG_TXBUFFER_BUFSIZE 0x0032U
- #define PHHAL_HW_CONFIG_TXBUFFER_LENGTH 0x0033U
- #define PHHAL_HW_CONFIG_TXBUFFER 0x0034U
- #define PHHAL_HW_CONFIG_MAX_PRECACHED_BYTES 0x0035U
- #define PHHAL_HW_CONFIG_BAL_CONNECTION 0x0040U
- #define PHHAL_HW_CONFIG_SERIAL_BITRATE 0x0041U
- #define PHHAL_HW_CONFIG_RFRESET_ON_TIMEOUT 0x0050U
- #define PHPAL_I14443P4_PARAM_BLOCKNO 0x0000
- #define PHPAL_I14443P4_PARAM_CID 0x0001
- #define PHPAL_I14443P4_PARAM_NAD 0x0002
- #define PHPAL_I14443P4_PARAM_FWI 0x0003
- #define PHPAL_I14443P4_PARAM_FSI 0x0004
- #define PHPAL_I14443P4_PARAM_MAXRETRYCOUNT 0x0005
- #define P2_FW_DM_CHK_LEN(len, class, ins)

Functions

- Bool p2_fw_direct_mode_xchg (uint8_t message_type)

  Performs exchange with a card in direct mode.

- void p2_fw_dm_hal_switch_config (uint8_t slot_idx)

  Switches reader chip to correct HAL configuration stack for a card.

- Bool p2_fw_dm_xchg_hal (uint8_t message_type, uint8_t slot_idx)

  Performs the exchange function for HAL.

- Bool p2_fw_dm_mfc_auth_hal (uint8_t message_type)

  Performs MIFARE Classic Authentication.

- Bool p2_fw_dm_hal (uint8_t message_type)

  Contains all functions relating to HAL implementation.

- Bool p2_fw_dm_ro (uint8_t message_type)

  Contains all functions relating to reader operations.

- Bool p2_fw_dm_l3 (uint8_t message_type)

  Contains all functions relating to ISO14443 level 3.

- Bool p2_fw_dm_cid (uint8_t message_type)

  Contains all functions relating to Channel ID Managment.

- Bool p2_fw_dm_l4a (uint8_t message_type)

  Contains all functions relating to ISO14443 level 4 Activation.

- Bool p2_fw_dm_l4 (uint8_t message_type)

  Contains all functions relating to ISO14443 level 4.

- Bool p2_fw_dm_contact_card (uint8_t message_type)

  Contains all functions relating to contact cards.

- void p2_fw_dm_send (uint8_t message_type, uint16_t status, uint8_t class, uint8_t cmd, uint16_t pay_len)

  Sends direct mode reply.

- Bool p2_fw_dm_check_if_vaild (uint8_t message_type)

  Performs checks on direct mode message.

- void p2_fw_dm_hal_set_cfg_txdatarate (uint8_t slot_idx)

  Sets the TX Data Rate.

- void p2_fw_dm_hal_set_cfg_rxdatarate (uint8_t slot_idx)

  Sets the RX Data Rate.

- Bool p2_fw_dm_check_max_len (uint16_t len)

  Check if max length reached.

- phStatus_t p2_fw_dm_translate_error_code (phcsBfl_Status_t error)

  Translates the old BFL error code to the new BFL error code.

- void p2_fw_dm_hal_set_cfg_timeout (uint8_t slot_idx)

7.6 include/p2_fw_ext.h File Reference

This graph shows which files directly or indirectly include this file:

Defines

- #define P2_FW_USB_INT_IN_EP 0x81
- #define P2_FW_USB_BULK_OUT_EP 0x05
- #define P2_FW_USB_BULK_IN_EP 0x82

Functions

- void p2_fw_usb_init_usb (void)

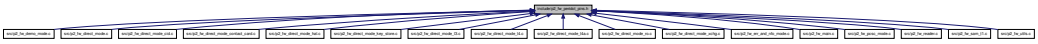
  Initializes USB Communication.

- void p2_fw_serial_init_serial (void)

  Initializes Serial (RS232, RS485) Communication.

7.7 include/p2_fw_peridot_pins.h File Reference

This graph shows which files directly or indirectly include this file:



Defines

- #define P2_FW_PINS_DIP_1 0x02000000
- #define P2_FW_PINS_DIP_2 0x04000000
- #define P2_FW_PINS_DIP_3 0x08000000
- #define P2_FW_PINS_DIP_4 0x10000000
- #define P2_FW_PINS_DIP_5 0x00020000
- #define P2_FW_PINS_DIP_6 0x00040000
- #define P2_FW_PINS_DIP_7 0x00080000
- #define P2_FW_PINS_DIP_8 0x00100000
- #define P2_FW_PINS_CFG_1 0x02000000
- #define P2_FW_PINS_CFG_2 0x04000000
- #define P2_FW_PINS_CFG_3 0x10000000
- #define P2_FW_PINS_CFG_4 0x20000000
- #define P2_FW_PINS_ANTENA_BLUE 0x00000080
- #define P2_FW_PINS_ANTENA_GREEN 0x00000040
- #define P2_FW_PINS_ANTENA_RED 0x00000020
- #define P2_FW_PINS_BEEPER 0x00000010
- #define P2_FW_PINS_LEDS_YELLOW_2 0x00000100
- #define P2_FW_PINS_LEDS_YELLOW_3 0x00000080
- #define P2_FW_PINS_LEDS_YELLOW_4 0x00000100

7.8 include/p2_fw_sam_t1.h File Reference

This graph shows which files directly or indirectly include this file:

Data Structures

- struct sam_t1_param

Defines

- #define P2_FW_SAM_MODE_PPS 0
- #define P2_FW_SAM_MODE_X 1
- #define P2_FW_SAM_FRAME_T1 0
- #define P2_FW_SAM_FRAME_APDU 1
- #define P2_FW_TIMING_MODE_NONE 0
- #define P2_FW_TIMING_MODE_COM 1
- #define P2_FW_TIMING_MODE_FDT 2

Functions

- void p2_fw_sam_t1_init (uint8_t chip, uint8_t mode)
- void p2_fw_sam_t1_deinit (void)
- void p2_fw_sam_t1_start (void)
- void p2_fw_sam_t1_send (uint8_t *data, uint32_t len, uint8_t frame_type)
- uint32_t p2_fw_sam_t1_receive (uint8_t *data, uint8_t frame_type)
- void p2_fw_sam_t1_get_atr (uint8_t *buffer, uint8_t *max_length)
- void p2_fw_sam_t1_warm_reset (void)
- Bool p2_fw_sam_t1_is_busy (void)
- Bool p2_fw_sam_t1_is_power_off (void)
- Bool p2_fw_sam_t1_is_sam_inserted (void)
- void p2_fw_sam_t1_set_rec_extraGuardTime (uint8_t guardTime)
- void p2_fw_sam_t1_set_bwi_cwi (uint8_t bwi, uint8_t cwi)
- Bool p2_fw_sam_t1_prepare_pps (uint8_t *ppsData, uint8_t *ppsLen)
- void p2_fw_sam_t1_pps (void)
- void p2_fw_sam_t1_set_etu (uint8_t fi_di)
- void p2_fw_sam_t1_set_my_debug (uint32_t x)
- void p2_fw_sam_t1_set_timing_mode (uint8_t mode)
- uint8_t p2_fw_sam_t1_get_timing_mode (void)
- void EINT3_IRQHandler (void)
- void TIMER0_IRQHandler (void)
- void TIMER2_IRQHandler (void)

7.8.1 Define Documentation

7.8.1.1 #define P2_FW_SAM_FRAME_APDU 1

Definition at line 24 of file p2_fw_sam_t1.h.

7.8.1.2 #define P2_FW_SAM_FRAME_T1 0

Definition at line 23 of file p2_fw_sam_t1.h.

7.8.1.3 #define P2_FW_SAM_MODE_PPS 0

Definition at line 20 of file p2_fw_sam_t1.h.

7.8.1.4 #define P2_FW_SAM_MODE_X 1

Definition at line 21 of file p2_fw_sam_t1.h.

7.8.1.5 #define P2_FW_TIMING_MODE_COM 1

Definition at line 27 of file p2_fw_sam_t1.h.

7.8.1.6 #define P2_FW_TIMING_MODE_FDT 2

Definition at line 28 of file p2_fw_sam_t1.h.

7.8.1.7 #define P2_FW_TIMING_MODE_NONE 0

Definition at line 26 of file p2_fw_sam_t1.h.

7.8.2 Function Documentation

7.8.2.1 void EINT3_IRQHandler (void)

Definition at line 1010 of file p2_fw_sam_t1.c.

7.8.2.2 void p2_fw_sam_t1_deinit (void)

Definition at line 393 of file p2_fw_sam_t1.c.

7.8.2.3 void p2_fw_sam_t1_get_atr (uint8_t * *buffer*, uint8_t * *max_length*)

Definition at line 725 of file p2_fw_sam_t1.c.

7.8.2.4 uint8_t p2_fw_sam_t1_get_timing_mode (void)

Definition at line 150 of file p2_fw_sam_t1.c.

7.8.2.5 void p2_fw_sam_t1_init (uint8_t *chip*, uint8_t *mode*)

Definition at line 207 of file p2_fw_sam_t1.c.

7.8.2.6 Bool p2_fw_sam_t1_is_busy (void)

Definition at line 796 of file p2_fw_sam_t1.c.

7.8.2.7 Bool p2_fw_sam_t1_is_power_off (void)

Definition at line 976 of file p2_fw_sam_t1.c.

7.8.2.8 Bool p2_fw_sam_t1_is_sam_inserted (void)

Definition at line 986 of file p2_fw_sam_t1.c.

7.8.2.9 void p2_fw_sam_t1_pps (void)

Definition at line 613 of file p2_fw_sam_t1.c.

7.8.2.10 Bool p2_fw_sam_t1_prepare_pps (uint8_t * *ppsData*, uint8_t * *ppsLen*)

Definition at line 172 of file p2_fw_sam_t1.c.

7.8.2.11 uint32_t p2_fw_sam_t1_receive (uint8_t * *data*, uint8_t *frame_type*)

Definition at line 670 of file p2_fw_sam_t1.c.

7.8.2.12 void p2_fw_sam_t1_send (uint8_t * *data*, uint32_t *len*, uint8_t *frame_type*)

Definition at line 492 of file p2_fw_sam_t1.c.

7.8.2.13 void p2_fw_sam_t1_set_bwi_cwi (uint8_t *bwi*, uint8_t *cwi*)

Definition at line 168 of file p2_fw_sam_t1.c.

7.8.2.14 void p2_fw_sam_t1_set_etu (uint8_t *fi_di*)

Definition at line 630 of file p2_fw_sam_t1.c.

7.8.2.15 void p2_fw_sam_t1_set_my_debug (uint32_t *x*)

Definition at line 140 of file p2_fw_sam_t1.c.

7.8.2.16 void p2_fw_sam_t1_set_rec_extraGuardTime (uint8_t *guardTime*)

Definition at line 155 of file p2_fw_sam_t1.c.

7.8.2.17 void p2_fw_sam_t1_set_timing_mode (uint8_t *mode*)

Definition at line 145 of file p2_fw_sam_t1.c.

7.8.2.18 void p2_fw_sam_t1_start (void)

Definition at line 433 of file p2_fw_sam_t1.c.

7.8.2.19 void p2_fw_sam_t1_warm_reset (void)

Definition at line 757 of file p2_fw_sam_t1.c.

7.8.2.20 void TIMER0_IRQHandler (void)

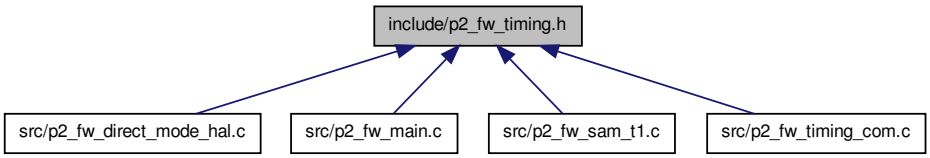
Definition at line 1090 of file p2_fw_sam_t1.c.

7.8.2.21 void TIMER2_IRQHandler (void)

Definition at line 1040 of file p2_fw_sam_t1.c.

7.9 include/p2_fw_timing.h File Reference

This graph shows which files directly or indirectly include this file:



Functions

- void p2_fw_timing_init (void)

  Initializes Timing Mode.

- void p2_fw_timing_start (void)

  Starts Timer.

- void p2_fw_timing_stop_cless (int timeout)

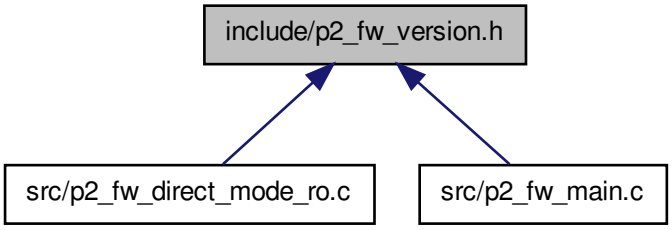
  Stops Timer for Contact Less Cards.

- void p2_fw_timing_stop_contac ()

  Stops Timer for Contact Cards.

7.10 include/p2_fw_version.h File Reference

This graph shows which files directly or indirectly include this file:



Defines

- #define P2_FW_VER_MAJOR 0x01
- #define P2_FW_VER_MINOR 0x10
- #define P2_FW_VER_BUILD 0x06

7.10.1 Define Documentation

7.10.1.1 #define P2_FW_VER_BUILD 0x06

Definition at line 21 of file p2_fw_version.h.

7.10.1.2 #define P2_FW_VER_MAJOR 0x01

Definition at line 19 of file p2_fw_version.h.

7.10.1.3 #define P2_FW_VER_MINOR 0x10

Definition at line 20 of file p2_fw_version.h.

7.11 src/bfl_support_files/phcsBflBal_Hw1SerCM3.c File Reference

```
#include <p2_fw.h>
```

```
#include <phcsBflBal.h>
#include <lpc17xx_uart.h>
#include <lpc17xx_libcfg.h>
#include <lpc17xx_pinsel.h>
#include <lpc17xx_gpio.h>
```

Include dependency graph for phcsBflBal_Hw1SerCM3.c:

Defines

- #define UART_PORT 2
- #define BUFFER_LENGTH 1
- #define THE_UART UART2

Functions

- void phcsBflBal_Hw1SerCM3Init (phcsBflBal_t *cif, void *comm_params)

7.11.1 Detailed Description

Projekt: Object Oriented Reader Library Framework RegCtl component.

Source: phcsBflBal_Hw1SerCM3.c

Comment: UART communication between LPC17xx and RC523

History: SP: Adoppted from sample for LPC17xx (Pegoda 2 Project) 22. October 2009

Definition in file phcsBflBal_Hw1SerCM3.c.

7.11.2 Define Documentation

7.11.2.1 #define BUFFER_LENGTH 1

Definition at line 41 of file phcsBflBal_Hw1SerCM3.c.

7.11.2.2 #define THE_UART UART2

Definition at line 44 of file phcsBflBal_Hw1SerCM3.c.

7.11.2.3 #define UART_PORT 2

Definition at line 39 of file phcsBflBal_Hw1SerCM3.c.

7.11.3 Function Documentation

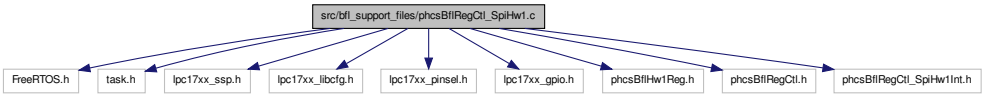
7.11.3.1 void phcsBflBal_Hw1SerCM3Init (phcsBflBal_t * *cif*, void * *comm_params*)

Definition at line 54 of file phcsBflBal_Hw1SerCM3.c.

7.12 src/bfl_support_files/phcsBflRegCtl_SpiHw1.c File Reference

```
#include <FreeRTOS.h>
#include <task.h>
#include <lpc17xx_ssp.h>
#include <lpc17xx_libcfg.h>
#include <lpc17xx_pinsel.h>
#include <lpc17xx_gpio.h>
#include <phcsBflHw1Reg.h>
#include <phcsBflRegCtl.h>
#include <phcsBflRegCtl_SpiHw1Int.h>
```

Include dependency graph for phcsBflRegCtl_SpiHw1.c:



Defines

- #define BUFFER_SIZE 65

Functions

- void phcsBflRegCtl_SpiHw1Init (phcsBflRegCtl_t *cif, void *p_params, phcsBflBal_t *p_lower)

- phcsBfl_Status_t phcsBflRegCtl_SpiHw1ModReg (phcsBflRegCtl_ModRegParam_t *modify_param)
- phcsBfl_Status_t phcsBflRegCtl_SpiHw1GetReg (phcsBflRegCtl_GetRegParam_t *getreg_param)
- phcsBfl_Status_t phcsBflRegCtl_SpiHw1SetReg (phcsBflRegCtl_SetRegParam_t *setreg_param)
- phcsBfl_Status_t phcsBflRegCtl_SpiHw1SetMultiReg (phcsBflRegCtl_SetMultiRegParam_t *setmultireg_param)
- phcsBfl_Status_t phcsBflRegCtl_SpiHw1GetMultiReg (phcsBflRegCtl_GetMultiRegParam_t *getmultireg_param)

7.12.1 Detailed Description

Projekt: Object Oriented Reader Library Framework RegCtl component.

Source: phcsBflRegCtl_SpiHw1.c

Comment: SPI communication between LPC17xx and RC523

History: SP: Adoppted from sample for LPC17xx (Pegoda 2 Project) 22. October 2009

Definition in file phcsBflRegCtl_SpiHw1.c.

7.12.2 Define Documentation

7.12.2.1 #define BUFFER_SIZE 65

Definition at line 48 of file phcsBflRegCtl_SpiHw1.c.

7.12.3 Function Documentation

7.12.3.1 phcsBfl_Status_t phcsBflRegCtl_SpiHw1GetMultiReg (phcsBflRegCtl_GetMultiRegParam_t * *getmultireg_param*)

Definition at line 243 of file phcsBflRegCtl_SpiHw1.c.

7.12.3.2 phcsBfl_Status_t phcsBflRegCtl_SpiHw1GetReg (phcsBflRegCtl_GetRegParam_t * *getreg_param*)

Definition at line 179 of file phcsBflRegCtl_SpiHw1.c.

7.12.3.3 void phcsBflRegCtl_SpiHw1Init (phcsBflRegCtl_t * *cif*, void * *p_params*, phcsBflBal_t * *p_lower*)

Definition at line 71 of file phcsBflRegCtl_SpiHw1.c.

7.12.3.4 phcsBfl_Status_t phcsBflRegCtl_SpiHw1ModReg (phcsBflRegCtl_ModRegParam_t * *modify_param*)

Definition at line 139 of file phcsBflRegCtl_SpiHw1.c.

7.12.3.5 phcsBfl_Status_t phcsBflRegCtl_SpiHw1SetMultiReg (phcsBflRegCtl_SetMultiRegParam_t * *setmultireg_param*)

Definition at line 213 of file phcsBflRegCtl_SpiHw1.c.

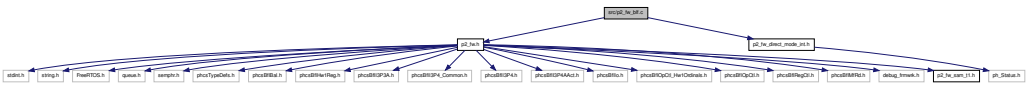
7.12.3.6 phcsBfl_Status_t phcsBflRegCtl_SpiHw1SetReg (phcsBflRegCtl_SetRegParam_t * *setreg_param*)

Definition at line 197 of file phcsBflRegCtl_SpiHw1.c.

7.13 src/p2_fw_blf.c File Reference

```
#include <p2_fw.h>
#include <p2_fw_direct_mode_int.h>
```

Include dependency graph for p2_fw_blf.c:



Defines

- #define PHHAL_HW_RC523_TIMER_FREQ 13.56f
- #define PHHAL_HW_RC523_TIMER_SHIFT 4.8f
- #define PHHAL_HW_RC523_ETU_106 9.44f

Functions

- Bool p2_fw_blf_init (void)
- void p2_fw_blf_set_up_rc_type_a_reading (void)
- Bool p2_fw_blf_reset_reader (void)
- void p2_fw_blf_set_timeout (uint16_t qsec, uint8_t aFlags)
- void p2_fw_blf_set_com_speed (uint8_t dri, uint8_t dsi)
- void p2_fw_blf_change_rc523_baud_rate (uint32_t baudrate)

7.13.1 Define Documentation

7.13.1.1 #define PHHAL_HW_RC523_ETU_106 9.44f

Definition at line 22 of file p2_fw_blf.c.

7.13.1.2 #define PHHAL_HW_RC523_TIMER_FREQ 13.56f

Definition at line 20 of file p2_fw_blf.c.

7.13.1.3 #define PHHAL_HW_RC523_TIMER_SHIFT 4.8f

Definition at line 21 of file p2_fw_blf.c.

7.14 src/p2_fw_ccid.c File Reference

```
#include <FreeRTOS.h>
#include <task.h>
#include "p2_fw.h"
#include "p2_fw_ext.h"
#include "p2_fw_ccid.h"
#include "p2_fw_direct_mode.h"
```

Include dependency graph for p2_fw_ccid.c:

Functions

- void p2_fw_ccid_top_half_dispatch (void)

  Top level ISR Dispatcher.

- void p2_fw_ccid_get_slot_status_top_half (void)

  Top half (ISR) function for get status command.

- Bool p2_fw_ccid_get_slot_status_bottom_half (uint8_t slot_idx)

  Bottom half function for get status command.

- void p2_fw_ccid_icc_power_on_top_half (void)

Top half (ISR) function for power on command.

- Bool p2_fw_ccid_icc_power_on_bottom_half (uint8_t slot_idx)

  Bottom half function for power on command.

- void p2_fw_ccid_icc_power_off_top_half (void)

  Top half (ISR) function for power off command.

- Bool p2_fw_ccid_icc_power_off_bottom_half (uint8_t slot_idx)

  Bottom half function for power off command.

- void p2_fw_ccid_get_parameters_top_half (void)

  Top half (ISR) function for get parameters command.

- Bool p2_fw_ccid_get_parameters_bottom_half (uint8_t slot_idx)

  Bottom half function for get parameters command.

- void p2_fw_ccid_set_parameters_top_half (void)

  Top half (ISR) function for set parameters command.

- Bool p2_fw_ccid_set_parameters_bottom_half (uint8_t slot_idx)

  Bottom half function for set parameters command.

- void p2_fw_ccid_escape_top_half (void)

  Top half (ISR) function for escape command.

- Bool p2_fw_ccid_escape_bottom_half (uint8_t slot_idx)

  Bottom half function for Escape command.

- Bool p2_fw_ccid_send_notify (uint8_t slot_idx)

  Sends slot change notify event.

- Bool p2_fw_ccid_check_header (uint8_t message_type)

  Checks if the CCID header is correct.

- Bool p2_fw_ccid_send_apdu (uint8_t *payload, uint8_t payload_len, uint8_t sw1, uint8_t sw2)

  Sends APDU with payload of payload_len and SW1 and SW2.

- void p2_fw_ccid_send_data (uint8_t message_type, uint8_t byte_1, uint8_t byte_2, uint8_t byte_3)

  Sends the CCID message.

7.15 src/p2_fw_ccid_xfer.c File Reference

```
#include <FreeRTOS.h>
#include <task.h>
#include "p2_fw.h"
#include "p2_fw_ccid.h"
#include "p2_fw_direct_mode.h"
```

Include dependency graph for p2_fw_ccid_xfer.c:

Functions

- void p2_fw_ccid_xfr_set_busy (void)
  Set the slot to busy (receiving and processing data).
- void p2_fw_ccid_xfr_clear_busy (void)
  Set the slot to not busy (receiving and processing data).
- void p2_fw_ccid_xfr_block_top_half (void)
  Top half (ISR) function for transfer command.
- Bool p2_fw_ccid_xfr_block_bottom_half (uint8_t slot_idx)
  Bottom half function for transfer command.
- void p2_fw_pcsc_send_apdu (uint8_t sw1, uint8_t sw2, uint16_t len)
- Bool p2_fw_pcsc_prepare_l3_card (uint8_t slot_idx)

7.15.1 Function Documentation

7.15.1.1 Bool p2_fw_pcsc_prepare_l3_card (uint8_t *slot_idx*)

Prepare L3 card to execute a PCSC extended command

Parameters

slot_index - index of a slot

Returns

Bool - TRUE if success or FALSE if error

Definition at line 362 of file p2_fw_ccid_xfer.c.

7.15.1.2 void p2_fw_pcsc_send_apdu (uint8_t *sw1*, uint8_t *sw2*, uint16_t *len*)

Send APDU to the client

Parameters

sw1 - SW 1 parameter

sw2 - SW 2 parameter

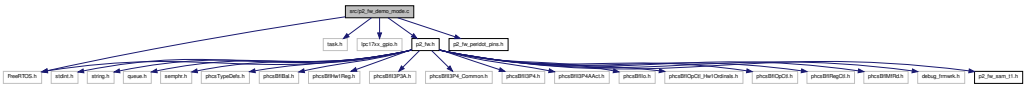
le - length of whole command

Definition at line 352 of file p2_fw_ccid_xfer.c.

7.16 src/p2_fw_demo_mode.c File Reference

```
#include <FreeRTOS.h>
#include <task.h>
#include <lpc17xx_gpio.h>
#include <p2_fw.h>
#include <p2_fw_peridot_pins.h>
```

Include dependency graph for p2_fw_demo_mode.c:



Functions

- void p2_fw_task_demo_mode (void ∗param)

7.17 src/p2_fw_direct_mode.c File Reference

```
#include <FreeRTOS.h>
#include <task.h>
#include <core_cm3.h>
#include <lpc17xx_gpio.h>
#include "p2_fw.h"
#include "p2_fw_ext.h"
#include "p2_fw_ccid.h"
```

```
#include "p2_fw_direct_mode.h"
#include "p2_fw_peridot_pins.h"
#include "p2_fw_direct_mode_int.h"
#include <debug_frmwrk.h>
```

Include dependency graph for p2_fw_direct_mode.c:

Functions

- Bool p2_fw_dm (uint8_t message_type, uint16_t allowed_cmds)
  Main dispatcher function for direct mode.
- Bool p2_fw_dm_check_if_vaild (uint8_t message_type)
  Performs checks on direct mode message.
- Bool p2_fw_dm_xcgh_l4 (uint8_t message_type, uint8_t slot_idx)
  Performs the exchange function for L4.
- Bool p2_fw_dm_xchg_hal (uint8_t message_type, uint8_t slot_idx)
  Performs the exchange function for HAL.
- void p2_fw_dm_send (uint8_t message_type, uint16_t status, uint8_t class, uint8_t cmd, uint16_t pay_len)
  Sends direct mode reply.
- Bool p2_fw_dm_check_max_len (uint16_t len)
  Check if max length reached.
- phStatus_t p2_fw_dm_translate_error_code (phcsBfl_Status_t error)
  Translates the old BFL error code to the new BFL error code.

7.18 src/p2_fw_direct_mode_cid.c File Reference

```
#include <FreeRTOS.h>
#include <task.h>
#include <core_cm3.h>
#include <lpc17xx_gpio.h>
```

```
#include "p2_fw.h"
#include "p2_fw_ext.h"
#include "p2_fw_ccid.h"
#include "p2_fw_direct_mode.h"
#include "p2_fw_peridot_pins.h"
#include "p2_fw_direct_mode_int.h"
#include <debug_frmwrk.h>
```

Include dependency graph for p2_fw_direct_mode_cid.c:

Functions

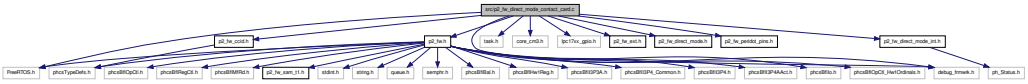
- Bool p2_fw_dm_cid (uint8_t message_type)

  Contains all functions relating to Channel ID Managment.

7.19 src/p2_fw_direct_mode_contact_card.c File Reference

```
#include <FreeRTOS.h>
#include <task.h>
#include <core_cm3.h>
#include <lpc17xx_gpio.h>
#include "p2_fw.h"
#include "p2_fw_ext.h"
#include "p2_fw_ccid.h"
#include "p2_fw_direct_mode.h"
#include "p2_fw_peridot_pins.h"
#include "p2_fw_direct_mode_int.h"
#include <debug_frmwrk.h>
```

Include dependency graph for p2_fw_direct_mode_contact_card.c:



Functions

- Bool p2_fw_dm_contact_card (uint8_t message_type)

  Contains all functions relating to contact cards.

7.20 src/p2_fw_direct_mode_hal.c File Reference

```
#include <FreeRTOS.h>
#include <task.h>
#include <core_cm3.h>
#include <lpc17xx_gpio.h>
#include "p2_fw.h"
#include "p2_fw_ext.h"
#include "p2_fw_ccid.h"
#include "p2_fw_direct_mode.h"
#include "p2_fw_peridot_pins.h"
#include "p2_fw_timing.h"
#include "p2_fw_direct_mode_int.h"
#include <debug_frmwrk.h>
```

Include dependency graph for p2_fw_direct_mode_hal.c:

Functions

- Bool p2_fw_dm_hal (uint8_t message_type)

  Contains all functions relating to HAL implementation.

- void p2_fw_dm_hal_switch_config (uint8_t slot_idx)

  Switches reader chip to correct HAL configuration stack for a card.

- void p2_fw_dm_hal_wait (uint16_t timeout1, uint8_t flags)

  Performs wait.

- Bool p2_fw_dm_mfc_auth_hal (uint8_t message_type)

  Performs MIFARE Classic Authentication.

- Bool p2_fw_dm_mfc_auth_hal_key_store (uint8_t message_type)

  Performs authentication by using key store.

- void p2_fw_dm_hal_set_cfg_txdatarate (uint8_t slot_idx)

  Sets the TX Data Rate.

- void p2_fw_dm_hal_set_cfg_rxdatarate (uint8_t slot_idx)

  Sets the RX Data Rate.

- void p2_fw_dm_hal_set_cfg_timeout (uint8_t slot_idx)

7.21 src/p2_fw_direct_mode_key_store.c File Reference

```
#include <FreeRTOS.h>
#include <task.h>
#include <core_cm3.h>
#include <lpc17xx_gpio.h>
#include "p2_fw.h"
#include "p2_fw_ext.h"
#include "p2_fw_ccid.h"
#include "p2_fw_direct_mode.h"
#include "p2_fw_peridot_pins.h"
#include "p2_fw_direct_mode_int.h"
#include "phKeyStore_Sw.h"
```

Include dependency graph for p2_fw_direct_mode_key_store.c:

Defines

- #define KEY_COUNT 2
- #define KEY_VERSIONS 3

Functions

- Bool p2_fw_dm_key_store_init (void)

  Inits the key store.

- Bool p2_fw_dm_key_store (uint8_t message_type)

  Performs KeyStore related functions.

- Bool p2_fw_key_store_get_key (uint16_t key_num, uint16_t key_ver, uint8_t *p_key, uint8_t key_len, uint16_t *key_type)

  Returns a key store key.

7.21.1 Define Documentation

7.21.1.1 #define KEY_COUNT 2

Definition at line 33 of file p2_fw_direct_mode_key_store.c.

7.21.1.2 #define KEY_VERSIONS 3

Definition at line 34 of file p2_fw_direct_mode_key_store.c.

7.22 src/p2_fw_direct_mode_l3.c File Reference

```
#include <FreeRTOS.h>
#include <task.h>
#include <core_cm3.h>
#include <lpc17xx_gpio.h>
#include "p2_fw.h"
#include "p2_fw_ext.h"
#include "p2_fw_ccid.h"
#include "p2_fw_direct_mode.h"
#include "p2_fw_peridot_pins.h"
#include "phcsBflI3P3A_Hw1Int.h"
```

```
#include "p2_fw_direct_mode_int.h"
#include <debug_frmwrk.h>
```

Include dependency graph for p2_fw_direct_mode_l3.c:

Functions

- Bool p2_fw_dm_l3 (uint8_t message_type)
  Contains all functions relating to ISO14443 level 3.

7.23 src/p2_fw_direct_mode_l4.c File Reference

```
#include <FreeRTOS.h>
#include <task.h>
#include <core_cm3.h>
#include <lpc17xx_gpio.h>
#include "p2_fw.h"
#include "p2_fw_ext.h"
#include "p2_fw_ccid.h"
#include "p2_fw_direct_mode.h"
#include "p2_fw_peridot_pins.h"
#include "phcsBflI3P3A_Hw1Int.h"
#include "p2_fw_direct_mode_int.h"
#include <debug_frmwrk.h>
```

Include dependency graph for p2_fw_direct_mode_l4.c:

Defines

- #define PHPAL_I14443P4_CID_MAX 14

- #define PHPAL_I14443P4_FWI_MAX 14
- #define PHPAL_I14443P4_FRAMESIZE_MAX 8

Functions

- Bool p2_fw_dm_l4 (uint8_t message_type)

  Contains all functions relating to ISO14443 level 4.

7.23.1 Define Documentation

7.23.1.1 #define PHPAL_I14443P4_CID_MAX 14

Definition at line 40 of file p2_fw_direct_mode_l4.c.

7.23.1.2 #define PHPAL_I14443P4_FRAMESIZE_MAX 8

Definition at line 42 of file p2_fw_direct_mode_l4.c.

7.23.1.3 #define PHPAL_I14443P4_FWI_MAX 14

Definition at line 41 of file p2_fw_direct_mode_l4.c.

7.24 src/p2_fw_direct_mode_l4a.c File Reference

```
#include <FreeRTOS.h>
#include <task.h>
#include <core_cm3.h>
#include <lpc17xx_gpio.h>
#include "p2_fw.h"
#include "p2_fw_ext.h"
#include "p2_fw_ccid.h"
#include "p2_fw_direct_mode.h"
#include "p2_fw_peridot_pins.h"
#include "phcsBflI3P3A_Hw1Int.h"
#include "p2_fw_direct_mode_int.h"
#include <debug_frmwrk.h>
```

Include dependency graph for p2_fw_direct_mode_l4a.c:

Functions

- Bool p2_fw_dm_l4a (uint8_t message_type)

  Contains all functions relating to ISO14443 level 4 Activation.

7.25 src/p2_fw_direct_mode_ro.c File Reference

```
#include <FreeRTOS.h>
#include <task.h>
#include <core_cm3.h>
#include <lpc17xx_gpio.h>
#include "p2_fw.h"
#include "p2_fw_ext.h"
#include "p2_fw_ccid.h"
#include "p2_fw_direct_mode.h"
#include "p2_fw_peridot_pins.h"
#include "p2_fw_version.h"
#include "p2_fw_direct_mode_int.h"
#include <debug_frmwrk.h>
```

Include dependency graph for p2_fw_direct_mode_ro.c:

Functions

- Bool p2_fw_dm_ro (uint8_t message_type)

  Contains all functions relating to reader operations.

7.26 src/p2_fw_direct_mode_xchg.c File Reference

```
#include <FreeRTOS.h>
#include <task.h>
#include <core_cm3.h>
#include <lpc17xx_gpio.h>
#include "p2_fw.h"
#include "p2_fw_ext.h"
#include "p2_fw_ccid.h"
#include "p2_fw_direct_mode.h"
#include "p2_fw_peridot_pins.h"
#include "p2_fw_direct_mode_int.h"
#include <phcsBflMfRd_Int.h>
#include <debug_frmwrk.h>
```

Include dependency graph for p2_fw_direct_mode_xchg.c:

Functions

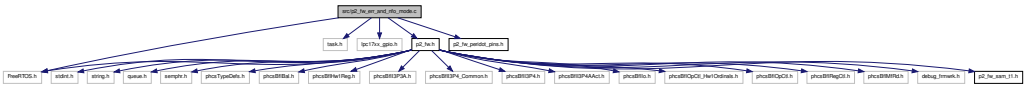
- Bool p2_fw_direct_mode_xchg (uint8_t message_type)

  Performs exchange with a card in direct mode.

7.27 src/p2_fw_err_and_nfo_mode.c File Reference

```
#include <FreeRTOS.h>
#include <task.h>
#include <lpc17xx_gpio.h>
#include <p2_fw.h>
#include <p2_fw_peridot_pins.h>
```

Include dependency graph for p2_fw_err_and_nfo_mode.c:



Functions

- void p2_fw_task_err_and_nfo_loop (void *param)

7.28 src/p2_fw_ext_intf_serial.c File Reference

```
#include "usbapi.h"
#include "p2_fw.h"
#include "p2_fw_ext.h"
#include "p2_fw_ccid.h"
#include <p2_fw_config.h>
#include "lpc17xx_pinsel.h"
```

Include dependency graph for p2_fw_ext_intf_serial.c:

Defines

- #define SERIAL_UART UART0

Functions

- void p2_fw_serial_init_serial (void)

  Initializes Serial (RS232, RS485) Communication.

- void UART3_IRQHandler (void)

7.28.1 Define Documentation

7.28.1.1 #define SERIAL_UART UART0

Definition at line 27 of file p2_fw_ext_intf_serial.c.

7.28.2 Function Documentation

7.28.2.1 void UART3_IRQHandler (void)

Definition at line 238 of file p2_fw_ext_intf_serial.c.

7.29 src/p2_fw_ext_intf_usb.c File Reference

```
#include "usbapi.h"
#include "p2_fw.h"
#include "p2_fw_ext.h"
#include "p2_fw_ccid.h"
```

Include dependency graph for p2_fw_ext_intf_usb.c:

Defines

- #define P2_FW_USB_LE_WORD(x) ((x)&0xFF),((x)>>8)
- #define P2_FW_USB_LE_DWORD(x) ((x)&0xFF),(((x)>>8)&0xFF),(((x)>>16)&0xFF),((x)>>24)
- #define P2_FW_USB_MAX_PACKET_SIZE 64
- #define P2_FW_USB_CONTROL_REQUEST_ABORT 0x01
- #define P2_FW_USB_CONTROL_REQUEST_GET_CLOCK_-FREQUENCIES 0x02
- #define P2_FW_USB_CONTROL_REQUEST_GET_DATA_RATES 0x03
- #define P2_FW_SERNUM_USB_OFFSET 147
- #define P2_FW_NAME_USB_OFFSET 143
- #define P2_FW_MAX_SLOT_INDEX 40

Functions

- void p2_fw_usb_init_usb (void)

  Initializes USB Communication.

7.29.1 Define Documentation

7.29.1.1 #define P2_FW_MAX_SLOT_INDEX 40

Definition at line 33 of file p2_fw_ext_intf_usb.c.

7.29.1.2 #define P2_FW_NAME_USB_OFFSET 143

Definition at line 32 of file p2_fw_ext_intf_usb.c.

7.29.1.3 #define P2_FW_SERNUM_USB_OFFSET 147

Definition at line 31 of file p2_fw_ext_intf_usb.c.

7.29.1.4 #define P2_FW_USB_CONTROL_REQUEST_ABORT 0x01

Definition at line 27 of file p2_fw_ext_intf_usb.c.

7.29.1.5 #define P2_FW_USB_CONTROL_REQUEST_GET_CLOCK_-FREQUENCIES 0x02

Definition at line 28 of file p2_fw_ext_intf_usb.c.

7.29.1.6 #define P2_FW_USB_CONTROL_REQUEST_GET_DATA_-RATES 0x03

Definition at line 29 of file p2_fw_ext_intf_usb.c.

7.29.1.7 #define P2_FW_USB_LE_DWORD(*x*) ((x)&0xFF),(((x)>>8)&0xFF),(((x)>>16)&0xFF),((x)>>24)

Definition at line 23 of file p2_fw_ext_intf_usb.c.

7.29.1.8 #define P2_FW_USB_LE_WORD(*x*) ((x)&0xFF),((x)>>8)

Definition at line 22 of file p2_fw_ext_intf_usb.c.

7.29.1.9 #define P2_FW_USB_MAX_PACKET_SIZE 64

Definition at line 25 of file p2_fw_ext_intf_usb.c.

7.30 src/p2_fw_flash_utils.c File Reference

```
#include <LPC17xx.h>
#include <core_cm3.h>
#include <system_LPC17xx.h>
#include <p2_fw.h>
```

Include dependency graph for p2_fw_flash_utils.c:

Defines

- #define IAP_LOCATION 0x1FFF1FF1
- #define IAP_SEC_28 0x00070000
- #define IAP_SEC_29 0x00078000
- #define IAP_PREP_SECTORS 50
- #define IAP_COPY_RAM_TO_FLASH 51
- #define IAP_ERASE_SECTOR 52
- #define IAP_BLANK_CHECK_SECTOR 53
- #define IAP_READ_ID 54
- #define IAP_READ_BOOT_CODE_VER 55
- #define IAP_READ_DEV_SER_NUM 58
- #define IAP_COMPARE 56
- #define IAP_REINVOKE 57
- #define P2_FW_CFG_S1 0x00
- #define P2_FW_CFG_S2 0x01
- #define P2_FW_CFG_COUNT 0x02
- #define P2_FW_CFG_ENTRY_OFFSET 0x04
- #define P2_FW_CFG_ENTRY_SIZE 0x24
- #define P2_FW_CFG_MAX 14

Typedefs

- typedef void(∗ IAP)(long[ ], long[ ])

Functions

- void p2_fw_flash_erase_config (void)
- Bool p2_fw_flash_read_serial (uint32_t ∗sernum)
- Bool p2_fw_flash_get_config (uint32_t cfg_id, uint8_t ∗buff)
- Bool p2_fw_flash_set_config (uint32_t cfg_id, uint8_t ∗buff)

7.30.1 Define Documentation

7.30.1.1 #define IAP_BLANK_CHECK_SECTOR 53

Definition at line 30 of file p2_fw_flash_utils.c.

7.30.1.2 #define IAP_COMPARE 56

Definition at line 34 of file p2_fw_flash_utils.c.

7.30.1.3 #define IAP_COPY_RAM_TO_FLASH 51

Definition at line 28 of file p2_fw_flash_utils.c.

7.30.1.4 #define IAP_ERASE_SECTOR 52

Definition at line 29 of file p2_fw_flash_utils.c.

7.30.1.5 #define IAP_LOCATION 0x1FFF1FF1

Definition at line 22 of file p2_fw_flash_utils.c.

7.30.1.6 #define IAP_PREP_SECTORS 50

Definition at line 27 of file p2_fw_flash_utils.c.

7.30.1.7 #define IAP_READ_BOOT_CODE_VER 55

Definition at line 32 of file p2_fw_flash_utils.c.

7.30.1.8 #define IAP_READ_DEV_SER_NUM 58

Definition at line 33 of file p2_fw_flash_utils.c.

7.30.1.9 #define IAP_READ_ID 54

Definition at line 31 of file p2_fw_flash_utils.c.

7.30.1.10 #define IAP_REINVOKE 57

Definition at line 35 of file p2_fw_flash_utils.c.

7.30.1.11 #define IAP_SEC_28 0x00070000

Definition at line 24 of file p2_fw_flash_utils.c.

7.30.1.12 #define IAP_SEC_29 0x00078000

Definition at line 25 of file p2_fw_flash_utils.c.

7.30.1.13 #define P2_FW_CFG_COUNT 0x02

Definition at line 39 of file p2_fw_flash_utils.c.

7.30.1.14 #define P2_FW_CFG_ENTRY_OFFSET 0x04

Definition at line 40 of file p2_fw_flash_utils.c.

7.30.1.15 #define P2_FW_CFG_ENTRY_SIZE 0x24

Definition at line 41 of file p2_fw_flash_utils.c.

7.30.1.16 #define P2_FW_CFG_MAX 14

Definition at line 43 of file p2_fw_flash_utils.c.

7.30.1.17 #define P2_FW_CFG_S1 0x00

Definition at line 37 of file p2_fw_flash_utils.c.

7.30.1.18 #define P2_FW_CFG_S2 0x01

Definition at line 38 of file p2_fw_flash_utils.c.

7.30.2 Typedef Documentation

7.30.2.1 typedef void(∗ IAP)(long[ ], long[ ])

Definition at line 45 of file p2_fw_flash_utils.c.

7.30.3 Function Documentation

7.30.3.1 void p2_fw_flash_erase_config (void)

Erases the configuration

Definition at line 50 of file p2_fw_flash_utils.c.

7.30.3.2 Bool p2_fw_flash_get_config (uint32_t *cfg_id,* uint8_t * *buff*)

Reads configuration option cfg_id from flash

Parameters

cfg_id - id of configuration option

buff - buffer to read it in

Returns

Bool - TRUE if success or FALSE if error

Definition at line 77 of file p2_fw_flash_utils.c.

7.30.3.3 Bool p2_fw_flash_read_serial (uint32_t * *sernum*)

Reads the serial number of the reader chip

Parameters

sernum - pointer to uint32_t to store serial number

Returns

Bool - TRUE if success or FALSE if error

Definition at line 58 of file p2_fw_flash_utils.c.

7.30.3.4 Bool p2_fw_flash_set_config (uint32_t *cfg_id,* uint8_t * *buff*)

Writes configuration option to flash

Parameters

cfg_id - id of configuration option

buff - buffer to read it out

Returns

Bool - TRUE if success or FALSE if error

Definition at line 127 of file p2_fw_flash_utils.c.

7.31 src/p2_fw_main.c File Reference

```
#include <LPC17xx.h>
#include <lpc17xx_nvic.h>
#include <lpc17xx_gpio.h>
#include <lpc17xx_pinsel.h>
#include <FreeRTOS.h>
#include <task.h>
#include <queue.h>
#include <semphr.h>
#include <p2_fw.h>
#include <p2_fw_version.h>
#include <p2_fw_timing.h>
#include <p2_fw_peridot_pins.h>
```

Include dependency graph for p2_fw_main.c:

Functions

- int main ()
- void p2_fw_invoke_error_mode (uint32_t error_code)

Variables

- p2_fw_data p2_fw

7.31.1 Function Documentation

7.31.1.1 int main ( )

Definition at line 89 of file p2_fw_main.c.

7.31.1.2 void p2_fw_invoke_error_mode (uint32_t *error_code*)

Puts the reader in error mode

Parameters

error_code - Error code

Definition at line 249 of file p2_fw_main.c.

7.31.2 Variable Documentation

7.31.2.1 p2_fw_data p2_fw

Definition at line 35 of file p2_fw_main.c.

7.32 src/p2_fw_pcsc_ext.c File Reference

```
#include <FreeRTOS.h>
#include <task.h>
#include "p2_fw.h"
#include "p2_fw_ccid.h"
#include "p2_fw_direct_mode.h"
```

Include dependency graph for p2_fw_pcsc_ext.c:

Functions

- Bool p2_fw_pcsc_commands (uint8_t slot_idx)

7.32.1 Function Documentation

7.32.1.1 Bool p2_fw_pcsc_commands (uint8_t *slot_idx*)

Execute PCSC extended command on slot

Parameters

slot_index - index of a slot

Returns

Bool - TRUE if success or FALSE if error

Definition at line 29 of file p2_fw_pcsc_ext.c.

7.33 src/p2_fw_pcsc_mode.c File Reference

```
#include <FreeRTOS.h>
#include <task.h>
#include <semphr.h>
#include <lpc17xx_gpio.h>
#include <p2_fw.h>
#include <p2_fw_peridot_pins.h>
#include <p2_fw_direct_mode.h>
```

Include dependency graph for p2_fw_pcsc_mode.c:

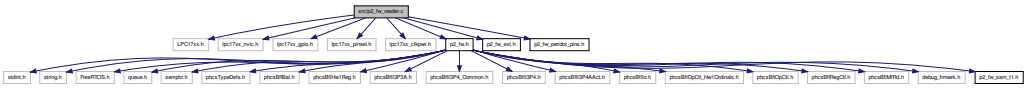
Functions

- void p2_fw_task_pcsc_execute (void *param)
- void p2_fw_task_pcsc_poll_and_act_loop (void *param)

7.34 src/p2_fw_reader.c File Reference

```
#include <LPC17xx.h>
#include <lpc17xx_nvic.h>
#include <lpc17xx_gpio.h>
#include <lpc17xx_pinsel.h>
#include "lpc17xx_clkpwr.h"
#include <p2_fw.h>
#include <p2_fw_ext.h>
#include <p2_fw_peridot_pins.h>
```

Include dependency graph for p2_fw_reader.c:



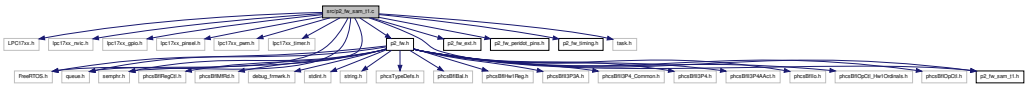
Functions

- void p2_fw_reader_setup_hardware (void)
- void p2_fw_reader_read_config (void)
- void p2_fw_reader_set_up_external_interface (void)
- Bool p2_fw_reader_set_up_reader_chip (void)

7.35 src/p2_fw_sam_t1.c File Reference

```
#include <LPC17xx.h>
#include <lpc17xx_nvic.h>
#include <lpc17xx_gpio.h>
#include <lpc17xx_pinsel.h>
#include <lpc17xx_pwm.h>
#include <lpc17xx_timer.h>
#include <p2_fw.h>
#include <p2_fw_ext.h>
#include <p2_fw_peridot_pins.h>
#include <p2_fw_sam_t1.h>
#include <p2_fw_timing.h>
#include <FreeRTOS.h>
#include <task.h>
#include <queue.h>
#include <semphr.h>
```

Include dependency graph for p2_fw_sam_t1.c:



Data Structures

- struct p2_fw_SAM_ctrl_
- struct cpot_atr_frame

Defines

- #define P2_FW_SAM_DEBUG 0

- #define P2_FW_SAM_T1_CLOCK_CONFIG 1
- #define T1_BYTE_WAIT_TIME 5
- #define P2_FW_SAM_PWM_CH0_MATCH_VALUE 4
- #define P2_FW_SAM_PWM_CH1_MATCH_VALUE 2
- #define P2_FW_SAM_TIMER_PRESCALE_VALUE 4
- #define P2_FW_SAM_CLOCK_LEN_NS 222
- #define P2_FW_SAM_MODE_RECEIVE 0
- #define P2_FW_SAM_MODE_SEND 1
- #define P2_FW_SAM_MODE_RESET 2
- #define P2_FW_SAM_MODE_IDLE 4
- #define P2_FW_SAM_MODE_WAIT_START_BIT 6
- #define P2_FW_SAM_MODE_POWER_OFF 7
- #define BITBAND_PERI_REF 0x40000000
- #define BITBAND_PERI_BASE 0x42000000
- #define BITBAND_PERI(a, b) ((BITBAND_PERI_BASE + ((a)-BITBAND_-PERI_REF)∗32 + ((b)∗4)))
- #define TIMER0_IR 0x40004000
- #define TIMER0_TCR 0x40004004
- #define TIMER0_IR_MR0 ∗((volatile unsigned char ∗) (BITBAND_-PERI(TIMER0_IR,0)))
- #define TIMER0_TCR_ENABLE ∗((volatile unsigned char ∗) (BITBAND_-PERI(TIMER0_TCR,0)))
- #define TIMER0_TCR_RESET ∗((volatile unsigned char ∗) (BITBAND_-PERI(TIMER0_TCR,1)))
- #define P2_FW_SAM_CONVERSION_UNKNOWN 0
- #define P2_FW_SAM_CONVERSION_DIRECT 1
- #define P2_FW_SAM_CONVERSION_INVERSE 2
- #define P2_FW_SAM_MAX_ATR_SIZE 33
- #define P2_FW_SAM_MAX_REC_DATALEN 600
- #define P2_FW_SAM_DEFAULT_BWI 8405405

Functions

- void p2_fw_sam_t1_set_my_debug (uint32_t x)
- void p2_fw_sam_t1_set_timing_mode (uint8_t mode)
- uint8_t p2_fw_sam_t1_get_timing_mode (void)
- void p2_fw_sam_t1_set_rec_extraGuardTime (uint8_t guardTime)
- void p2_fw_sam_t1_set_bwi_cwi (uint8_t bwi, uint8_t cwi)
- Bool p2_fw_sam_t1_prepare_pps (uint8_t ∗ppsData, uint8_t ∗ppsLen)
- void p2_fw_sam_t1_init (uint8_t chip, uint8_t mode)
- void p2_fw_sam_t1_deinit (void)
- void p2_fw_sam_t1_start (void)
- void p2_fw_sam_t1_send (uint8_t ∗data, uint32_t len, uint8_t frame_type)
- void p2_fw_sam_t1_pps (void)
- void p2_fw_sam_t1_set_etu (uint8_t fi_di)
- uint32_t p2_fw_sam_t1_receive (uint8_t ∗data, uint8_t frame_type)

- void p2_fw_sam_t1_get_atr (uint8_t *buffer, uint8_t *max_length)
- void p2_fw_sam_t1_warm_reset (void)
- Bool p2_fw_sam_t1_is_busy (void)
- Bool p2_fw_sam_t1_is_power_off (void)
- Bool p2_fw_sam_t1_is_sam_inserted (void)
- void EINT3_IRQHandler (void)
- void TIMER2_IRQHandler (void)
- void TIMER0_IRQHandler (void)

7.35.1 Define Documentation

7.35.1.1 #define BITBAND_PERI(*a*, *b*) ((BITBAND_PERI_BASE + ((a)-BITBAND_PERI_REF)*32 + ((b)*4)))

Definition at line 78 of file p2_fw_sam_t1.c.

7.35.1.2 #define BITBAND_PERI_BASE 0x42000000

Definition at line 77 of file p2_fw_sam_t1.c.

7.35.1.3 #define BITBAND_PERI_REF 0x40000000

Definition at line 76 of file p2_fw_sam_t1.c.

7.35.1.4 #define P2_FW_SAM_CLOCK_LEN_NS 222

Definition at line 49 of file p2_fw_sam_t1.c.

7.35.1.5 #define P2_FW_SAM_CONVERSION_DIRECT 1

Definition at line 89 of file p2_fw_sam_t1.c.

7.35.1.6 #define P2_FW_SAM_CONVERSION_INVERSE 2

Definition at line 90 of file p2_fw_sam_t1.c.

7.35.1.7 #define P2_FW_SAM_CONVERSION_UNKNOWN 0

Definition at line 88 of file p2_fw_sam_t1.c.

7.35.1.8 #define P2_FW_SAM_DEBUG 0

Definition at line 35 of file p2_fw_sam_t1.c.

7.35.1.9 #define P2_FW_SAM_DEFAULT_BWI 8405405

Definition at line 95 of file p2_fw_sam_t1.c.

7.35.1.10 #define P2_FW_SAM_MAX_ATR_SIZE 33

Maximum ATR size

Definition at line 92 of file p2_fw_sam_t1.c.

7.35.1.11 #define P2_FW_SAM_MAX_REC_DATALEN 600

Definition at line 93 of file p2_fw_sam_t1.c.

7.35.1.12 #define P2_FW_SAM_MODE_IDLE 4

Definition at line 72 of file p2_fw_sam_t1.c.

7.35.1.13 #define P2_FW_SAM_MODE_POWER_OFF 7

Definition at line 74 of file p2_fw_sam_t1.c.

7.35.1.14 #define P2_FW_SAM_MODE_RECEIVE 0

Definition at line 69 of file p2_fw_sam_t1.c.

7.35.1.15 #define P2_FW_SAM_MODE_RESET 2

Definition at line 71 of file p2_fw_sam_t1.c.

7.35.1.16 #define P2_FW_SAM_MODE_SEND 1

Definition at line 70 of file p2_fw_sam_t1.c.

7.35.1.17 #define P2_FW_SAM_MODE_WAIT_START_BIT 6

Definition at line 73 of file p2_fw_sam_t1.c.

7.35.1.18 #define P2_FW_SAM_PWM_CH0_MATCH_VALUE 4

Definition at line 46 of file p2_fw_sam_t1.c.

7.35.1.19 #define P2_FW_SAM_PWM_CH1_MATCH_VALUE 2

Definition at line 47 of file p2_fw_sam_t1.c.

7.35.1.20 #define P2_FW_SAM_T1_CLOCK_CONFIG 1

Definition at line 36 of file p2_fw_sam_t1.c.

7.35.1.21 #define P2_FW_SAM_TIMER_PRESCALE_VALUE 4

Definition at line 48 of file p2_fw_sam_t1.c.

7.35.1.22 #define T1_BYTE_WAIT_TIME 5

Definition at line 39 of file p2_fw_sam_t1.c.

7.35.1.23 #define TIMER0_IR 0x40004000

Definition at line 80 of file p2_fw_sam_t1.c.

7.35.1.24 #define TIMER0_IR_MR0 *((volatile unsigned char *) (BITBAND_PERI(TIMER0_IR,0)))

Definition at line 83 of file p2_fw_sam_t1.c.

7.35.1.25 #define TIMER0_TCR 0x40004004

Definition at line 81 of file p2_fw_sam_t1.c.

7.35.1.26 #define TIMER0_TCR_ENABLE *((volatile unsigned char *) (BITBAND_PERI(TIMER0_TCR,0)))

Definition at line 84 of file p2_fw_sam_t1.c.

7.35.1.27 #define TIMER0_TCR_RESET *((volatile unsigned char *) (BITBAND_PERI(TIMER0_TCR,1)))

Definition at line 85 of file p2_fw_sam_t1.c.

7.35.2 Function Documentation

7.35.2.1 void EINT3_IRQHandler (void)

Definition at line 1010 of file p2_fw_sam_t1.c.

7.35.2.2 void p2_fw_sam_t1_deinit (void)

Definition at line 393 of file p2_fw_sam_t1.c.

7.35.2.3 void p2_fw_sam_t1_get_atr (uint8_t * *buffer,* uint8_t * *max_length*)

Definition at line 725 of file p2_fw_sam_t1.c.

7.35.2.4 uint8_t p2_fw_sam_t1_get_timing_mode (void)

Definition at line 150 of file p2_fw_sam_t1.c.

7.35.2.5 void p2_fw_sam_t1_init (uint8_t *chip,* uint8_t *mode*)

Definition at line 207 of file p2_fw_sam_t1.c.

7.35.2.6 Bool p2_fw_sam_t1_is_busy (void)

Definition at line 796 of file p2_fw_sam_t1.c.

7.35.2.7 Bool p2_fw_sam_t1_is_power_off (void)

Definition at line 976 of file p2_fw_sam_t1.c.

7.35.2.8 Bool p2_fw_sam_t1_is_sam_inserted (void)

Definition at line 986 of file p2_fw_sam_t1.c.

7.35.2.9 void p2_fw_sam_t1_pps (void)

Definition at line 613 of file p2_fw_sam_t1.c.

7.35.2.10 Bool p2_fw_sam_t1_prepare_pps (uint8_t * *ppsData,* uint8_t * *ppsLen*)

Definition at line 172 of file p2_fw_sam_t1.c.

7.35.2.11 uint32_t p2_fw_sam_t1_receive (uint8_t * *data*, uint8_t *frame_type*)

Definition at line 670 of file p2_fw_sam_t1.c.

7.35.2.12 void p2_fw_sam_t1_send (uint8_t * *data*, uint32_t *len*, uint8_t *frame_type*)

Definition at line 492 of file p2_fw_sam_t1.c.

7.35.2.13 void p2_fw_sam_t1_set_bwi_cwi (uint8_t *bwi*, uint8_t *cwi*)

Definition at line 168 of file p2_fw_sam_t1.c.

7.35.2.14 void p2_fw_sam_t1_set_etu (uint8_t *fi_di*)

Definition at line 630 of file p2_fw_sam_t1.c.

7.35.2.15 void p2_fw_sam_t1_set_my_debug (uint32_t *x*)

Definition at line 140 of file p2_fw_sam_t1.c.

7.35.2.16 void p2_fw_sam_t1_set_rec_extraGuardTime (uint8_t *guardTime*)

Definition at line 155 of file p2_fw_sam_t1.c.

7.35.2.17 void p2_fw_sam_t1_set_timing_mode (uint8_t *mode*)

Definition at line 145 of file p2_fw_sam_t1.c.

7.35.2.18 void p2_fw_sam_t1_start (void)

Definition at line 433 of file p2_fw_sam_t1.c.

7.35.2.19 void p2_fw_sam_t1_warm_reset (void)

Definition at line 757 of file p2_fw_sam_t1.c.

7.35.2.20 void TIMER0_IRQHandler (void)

Definition at line 1090 of file p2_fw_sam_t1.c.

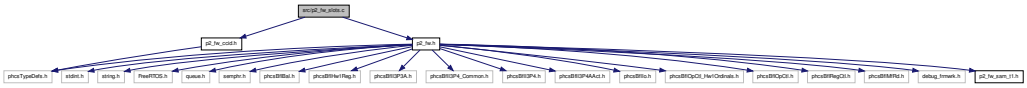
7.35.2.21 void TIMER2_IRQHandler (void)

Definition at line 1040 of file p2_fw_sam_t1.c.

7.36 src/p2_fw_slots.c File Reference

```
#include <p2_fw.h>
#include <p2_fw_ccid.h>
```

Include dependency graph for p2_fw_slots.c:



Functions

- void p2_fw_slots_init (void)
- void p2_fw_slots_free_cid (uint8_t cid)
- Bool p2_fw_slots_get_free_cid (uint8_t *cid)
- Bool p2_fw_slots_get_free_slot (uint8_t *slot)
- Bool p2_fw_slots_add_new_l4_card (phcsBflI3P4AAct_RatsParam_t *rat_p, uint8_t cid_index, uint8_t sak, uint8_t *atq, uint8_t *uid, uint8_t uid_len)
- Bool p2_fw_slots_add_new_l3_card (uint8_t sak, uint8_t *atq, uint8_t *uid, uint8_t uid_len)
- Bool p2_fw_slots_add_new_sam_card (void)
- Bool p2_fw_slots_is_known_l3_card (uint8_t *uid, uint8_t uid_len)
- Bool p2_fw_slots_get_atr (uint8_t slot_index, uint8_t *buffer, uint8_t *max_-length)
- void p2_fw_slots_remove_card (uint8_t slot_index)

7.37 src/p2_fw_timing_com.c File Reference

```
#include <FreeRTOS.h>
#include <task.h>
#include <core_cm3.h>
#include <lpc17xx_timer.h>
#include <lpc17xx_pinsel.h>
#include "p2_fw.h"
#include <debug_frmwrk.h>
```

```
#include <p2_fw_timing.h>
#include "p2_fw_direct_mode_int.h"
```

Include dependency graph for p2_fw_timing_com.c:

Defines

- #define TIMING_MODE_OPTION_MASK 0xFF00
- #define TIMING_MODE_OPTION_AUTOCLEAR 0x0100
- #define PHHAL_HW_RC523_DIGI_DELAY_US 59U
- #define PHHAL_HW_RC523_TIMER_FREQ 13.56f
- #define PHHAL_HW_RC523_TIMER_SHIFT 4.8f
- #define PHHAL_HW_RC523_ETU_106 9.434f

Functions

- void p2_fw_timing_init (void)

  Initializes Timing Mode.

- void p2_fw_timing_start (void)

  Starts Timer.

- void p2_fw_timing_stop_contac ()

  Stops Timer for Contact Cards.

- void p2_fw_timing_stop_cless (int timeout)

  Stops Timer for Contact Less Cards.

7.37.1 Define Documentation

7.37.1.1 #define PHHAL_HW_RC523_DIGI_DELAY_US 59U

Definition at line 35 of file p2_fw_timing_com.c.

7.37.1.2 #define PHHAL_HW_RC523_ETU_106 9.434f

Duration of one ETU at 106 kBit/s in [us].

Definition at line 38 of file p2_fw_timing_com.c.

7.37.1.3 #define PHHAL_HW_RC523_TIMER_FREQ 13.56f

RC internal timer frequency.

Definition at line 36 of file p2_fw_timing_com.c.

7.37.1.4 #define PHHAL_HW_RC523_TIMER_SHIFT 4.8f

Shift of the internal RC timer in ETUs.

Definition at line 37 of file p2_fw_timing_com.c.

7.37.1.5 #define TIMING_MODE_OPTION_AUTOCLEAR 0x0100

Definition at line 33 of file p2_fw_timing_com.c.

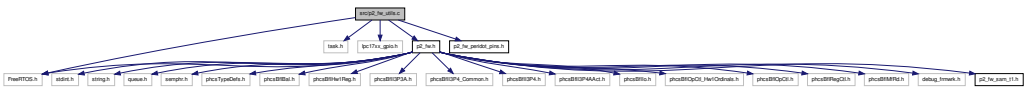
7.37.1.6 #define TIMING_MODE_OPTION_MASK 0xFF00

Definition at line 32 of file p2_fw_timing_com.c.

7.38 src/p2_fw_utils.c File Reference

```
#include <FreeRTOS.h>
#include <task.h>
#include <lpc17xx_gpio.h>
#include <p2_fw.h>
#include <p2_fw_peridot_pins.h>
```

Include dependency graph for p2_fw_utils.c:



Functions

- uint8_t p2_fw_utils_get_dri (uint8_t ta1)
- uint8_t p2_fw_utils_get_dsi (uint8_t ta1)
- void p2_fw_utils_blink (int count)
- void p2_fw_utils_field_off (void)
- void p2_fw_utils_field_on (uint16_t wFiledRecoveryTime)
- void p2_fw_utils_reg_write (uint8_t addr, uint8_t val)

- void p2_fw_utils_reg_read (uint8_t addr, uint8_t *val)
- void p2_fw_utils_dump_regs (void)

7.38.1 Function Documentation

7.38.1.1 void p2_fw_utils_dump_regs (void)

Definition at line 207 of file p2_fw_utils.c.

Index

__float32_t_defined
    p2_fw.h, 108
_p2_fw_data, 79
    ant_p, 83
    auth_done, 83
    buff_in, 83
    buff_in_len, 83
    buff_out, 83
    buff_out_idx, 83
    buff_out_len, 83
    c_slot, 83
    chip_type, 83
    cid_list, 84
    com_p, 84
    comm_buff, 84
    config, 84
    cont_tim, 84
    current_slot_idx, 84
    data, 84
    dsl_p, 84
    exc_p, 84
    g_rp, 85
    gm_rp, 85
    hlt_p, 85
    in, 85
    in_ccid_len, 85
    in_idx, 85
    in_len, 85
    ipc, 85
    is_busy, 86
    iso14443_3, 86
    iso14443_3_data, 86
    iso14443_4, 86
    iso14443_4_data, 86
    iso14443_4A_data, 86
    key, 86
    last_notify_evt, 86
    len, 86
    m_rp, 86
    max_slots, 86
    mfrd, 87
    mifare, 87
    mutex_transmission, 87
    num, 87
    out, 87
    out_len, 87
    out_written, 87
    pcsc_m, 87
    pcsc_mode, 88
    pps_p, 88
    pro_p, 88
    prs_p, 88
    queue_pcsc_exec, 88
    rc_com_bal, 88
    rc_io, 88
    rc_op_ctl, 88
    rc_reg_ctl, 88
    reader, 89
    regs_ops, 89
    regs_parms, 89
    rst_p, 89
    rx_xspeed, 89
    s_rp, 89
    scb_p, 89
    sel_p, 89
    send, 89
    slot_data, 89
    slots, 90
    sm_rp, 90
    spr_p, 90
_p2_fw_pcsc_exec, 90
    bottom_half, 90
    slot, 90
_p2_fw_slot, 91
    atq, 92
    ats, 92
    b_block_pool, 92
    bRfResetAfterTo, 92
    c_type, 92
    card_pro_p, 92
    cardtype_hw, 93
    chain, 93

data, 93
direct_m, 93
dri, 93
dsi, 93
dwTimingUs, 93
iso14443_4, 93
len, 94
m_BlockNumber, 94
m_Flags, 94
m_HistoryFlag, 94
m_State, 94
max_retrys, 94
nad, 94
params, 94
proto, 95
s_regs, 95
sak, 95
type, 95
uid, 95
wAdditionalInfo, 95
wFieldOffTime, 95
wFieldRecoveryTime, 95
wTimeoutUnit, 95
wTimingMode, 96

ant_p
_p2_fw_data, 83
atq
_p2_fw_slot, 92
ats
_p2_fw_slot, 92
auth_done
_p2_fw_data, 83

b_block_pool
_p2_fw_slot, 92
BFL 4.8, 11
bfl_4_8
p2_fw_blf_change_rc523_baud_-rate, 12
p2_fw_blf_init, 12
p2_fw_blf_reset_reader, 12
p2_fw_blf_set_com_speed, 12
p2_fw_blf_set_timeout, 13
p2_fw_blf_set_up_rc_type_a_-reading, 13
BITBAND_PERI
p2_fw_sam_t1.c, 167
BITBAND_PERI_BASE
p2_fw_sam_t1.c, 167
BITBAND_PERI_REF
p2_fw_sam_t1.c, 167
bitIndex
p2_fw_SAM_ctrl_, 98
bottom_half
_p2_fw_pcsc_exec, 90
bRfResetAfterTo
_p2_fw_slot, 92
buff_in
_p2_fw_data, 83
buff_in_len
_p2_fw_data, 83
buff_out
_p2_fw_data, 83
buff_out_idx
_p2_fw_data, 83
buff_out_len
_p2_fw_data, 83
BUFFER_LENGTH
phcsBflBal_Hw1SerCM3.c, 138
BUFFER_SIZE
phcsBflRegCtl_SpiHw1.c, 140
bwi
p2_fw_SAM_ctrl_, 98
BWI_CWI
sam_t1_param, 100

c_slot
_p2_fw_data, 83
c_type
_p2_fw_slot, 92
card_pro_p
_p2_fw_slot, 92
cardtype_hw
_p2_fw_slot, 93
ccid
P2_FW_CCID_BULK_HEADER, 21
P2_FW_CCID_BULK_OUT_-REQ_ESCAPE, 21
P2_FW_CCID_BULK_OUT_-REQ_GETPARAMETERS, 22
P2_FW_CCID_BULK_OUT_-REQ_GETSLOTSTATUS, 22
P2_FW_CCID_BULK_OUT_-REQ_ICCPOWEROFF, 22
P2_FW_CCID_BULK_OUT_-REQ_ICCPOWERON, 22

P2_FW_CCID_BULK_OUT_-
REQ_SETPARAMETERS,
22
P2_FW_CCID_BULK_OUT_-
REQ_XFRBLOCK, 22
p2_fw_ccid_check_header, 25
P2_FW_CCID_ERROR_SLOT_-
BUSY, 22
P2_FW_CCID_ERROR_SLOT_-
CMD_NOT_SUPPORTED,
22
P2_FW_CCID_ERROR_SLOT_-
HW_ERROR, 23
P2_FW_CCID_ERROR_SLOT_-
ICC_MUTE, 23
P2_FW_CCID_ERROR_SLOT_-
NOT_EXIST, 23
P2_FW_CCID_ERROR_SLOT_-
XFR_OVERRUN, 23
p2_fw_ccid_escape_bottom_half, 25
p2_fw_ccid_escape_top_half, 26
p2_fw_ccid_get_parameters_-
bottom_half, 26
p2_fw_ccid_get_parameters_top_-
half, 26
p2_fw_ccid_get_slot_status_-
bottom_half, 26
p2_fw_ccid_get_slot_status_top_-
half, 27
P2_FW_CCID_HEADER_-
LENGTH_BYTE_1, 23
P2_FW_CCID_HEADER_-
LENGTH_BYTE_2, 23
P2_FW_CCID_HEADER_-
LENGTH_BYTE_3, 23
P2_FW_CCID_HEADER_-
LENGTH_BYTE_4, 23
P2_FW_CCID_HEADER_-
MESSAGE_TYPE, 24
P2_FW_CCID_HEADER_MSG_-
BYTE_1, 24
P2_FW_CCID_HEADER_MSG_-
BYTE_2, 24
P2_FW_CCID_HEADER_MSG_-
BYTE_3, 24
P2_FW_CCID_HEADER_SEQ, 24
P2_FW_CCID_HEADER_SLOT,
24
p2_fw_ccid_icc_power_off_-
bottom_half, 27
p2_fw_ccid_icc_power_off_top_-
half, 27
p2_fw_ccid_icc_power_on_-
bottom_half, 27
p2_fw_ccid_icc_power_on_top_-
half, 28
P2_FW_CCID_INT_IN_NOTIFY_-
SLOT_CHANGE, 24
P2_FW_CCID_RDR_TO_PC_-
ESCAPE, 24
P2_FW_CCID_RDR_TO_PC_-
PARAMETERS, 25
P2_FW_CCID_RDR_TO_PC_-
SLOT_DATA_BLOCK, 25
P2_FW_CCID_RDR_TO_PC_-
SLOT_STATUS, 25
p2_fw_ccid_send_apdu, 28
p2_fw_ccid_send_data, 28
p2_fw_ccid_send_notify, 29
p2_fw_ccid_set_parameters_-
bottom_half, 29
p2_fw_ccid_set_parameters_top_-
half, 29
P2_FW_CCID_STATUS_CMD_-
FAILED, 25
p2_fw_ccid_top_half_dispatch, 30
p2_fw_ccid_xfr_block_bottom_half,
30
p2_fw_ccid_xfr_block_top_half, 30
p2_fw_ccid_xfr_clear_busy, 30
p2_fw_ccid_xfr_set_busy, 31
CCID functions, 19
chain
_p2_fw_slot, 93
chip_type
_p2_fw_data, 83
chipMode
p2_fw_SAM_ctrl_, 98
chipType
p2_fw_SAM_ctrl_, 98
cid_list
_p2_fw_data, 84
ClockStop
sam_t1_param, 100
com_p
_p2_fw_data, 84
comm_buff
_p2_fw_data, 84
config
_p2_fw_data, 84

P2_FW_CONFIG_COMM_SER_-232_PORT, 31
P2_FW_CONFIG_COMM_SER_-485_PORT, 31
P2_FW_CONFIG_DEBUG_PORT, 32
P2_FW_CONFIG_RC523_UART_-PORT, 32
Configuration definitions, 31
cont_tim
_p2_fw_data, 84
conversion
p2_fw_SAM_ctrl_, 98
cpot_atr_frame, 96
historicalC, 96
state, 96
TAi, 96
TBi, 96
TCi, 97
TCK, 97
TDi, 97
current_slot_idx
_p2_fw_data, 84

data
_p2_fw_data, 84
_p2_fw_slot, 93
p2_fw_SAM_ctrl_, 98
dataIndex
p2_fw_SAM_ctrl_, 98
Direct Mode, 32
direct_m
_p2_fw_slot, 93
dm
p2_fw_dm, 49
P2_FW_DM_ALLOWED_CMDS_-ALL, 35
P2_FW_DM_ALLOWED_CMDS_-CID, 35
P2_FW_DM_ALLOWED_CMDS_-CONTACT_CARD, 35
P2_FW_DM_ALLOWED_CMDS_-HAL, 35
P2_FW_DM_ALLOWED_CMDS_-KSTOR, 36
P2_FW_DM_ALLOWED_CMDS_-L3, 36
P2_FW_DM_ALLOWED_CMDS_-L4, 36
P2_FW_DM_ALLOWED_CMDS_-L4A, 36
P2_FW_DM_ALLOWED_CMDS_-RO, 36
P2_FW_DM_ALLOWED_CMDS_-XCHG, 36
P2_FW_DM_CID, 36
P2_FW_DM_CID_FREE, 36
P2_FW_DM_CID_GET_FREE, 37
P2_FW_DM_CID_INIT, 37
P2_FW_DM_CLASS_BYTE, 37
P2_FW_DM_CONTACT_CARD, 37
P2_FW_DM_CONTACTCARD_-ACTIVATE_CARD, 37
P2_FW_DM_CONTACTCARD_-CLOCK_START, 37
P2_FW_DM_CONTACTCARD_-CLOCK_STOP, 37
P2_FW_DM_CONTACTCARD_-COLD_RESET, 37
P2_FW_DM_CONTACTCARD_-DEACTIVATE_CARD, 38
P2_FW_DM_CONTACTCARD_-PPS, 38
P2_FW_DM_CONTACTCARD_-PRESENCE_CHECK, 38
P2_FW_DM_CONTACTCARD_-TRANSMIT_DATA, 38
P2_FW_DM_CONTACTCARD_-WARM_RESET, 38
P2_FW_DM_FAILD, 38
P2_FW_DM_HAL, 38
P2_FW_DM_HAL_APP_PROT_-SET, 38
P2_FW_DM_HAL_EXEC_CMD, 39
P2_FW_DM_HAL_GET_CFG, 39
P2_FW_DM_HAL_INIT, 39
P2_FW_DM_HAL_MFC_AUTH, 39
P2_FW_DM_HAL_MFC_AUTH_-KEY, 39
P2_FW_DM_HAL_SET_CFG, 39
P2_FW_DM_HAL_WAIT, 39
p2_fw_dm_hal_wait, 49
P2_FW_DM_HAL_XCHG, 39
P2_FW_DM_INSTR_BYTE, 40
p2_fw_dm_key_store, 50
p2_fw_dm_key_store_init, 50

P2_FW_DM_KSTOR, 40
P2_FW_DM_KSTOR_CHG_KUC, 40
P2_FW_DM_KSTOR_FORMAT_KEY, 40
P2_FW_DM_KSTOR_GET_CFG_STR, 40
P2_FW_DM_KSTOR_GET_CONFIG, 40
P2_FW_DM_KSTOR_GET_KEY, 40
P2_FW_DM_KSTOR_GET_KEY_ENTRY, 40
P2_FW_DM_KSTOR_GET_KUC, 41
P2_FW_DM_KSTOR_INIT, 41
P2_FW_DM_KSTOR_SET_CEK, 41
P2_FW_DM_KSTOR_SET_CFG_STR, 41
P2_FW_DM_KSTOR_SET_CONFIG, 41
P2_FW_DM_KSTOR_SET_FULL_KEY, 41
P2_FW_DM_KSTOR_SET_KEY, 41
P2_FW_DM_KSTOR_SET_KEY_POS, 41
P2_FW_DM_KSTOR_SET_KUC, 42
P2_FW_DM_L3, 42
P2_FW_DM_L3_ACT_CARD, 42
P2_FW_DM_L3_ANTICOL, 42
P2_FW_DM_L3_GET_SER, 42
P2_FW_DM_L3_HLTA, 42
P2_FW_DM_L3_INIT, 42
P2_FW_DM_L3_REQA, 42
P2_FW_DM_L3_SELECT, 43
P2_FW_DM_L3_WKUA, 43
P2_FW_DM_L3_XCHG, 43
P2_FW_DM_L4, 43
P2_FW_DM_L4_DESELECT, 43
P2_FW_DM_L4_GET_CFG, 43
P2_FW_DM_L4_INIT, 43
P2_FW_DM_L4_PRES_CHECK, 43
P2_FW_DM_L4_RESET_PROTO, 44
P2_FW_DM_L4_SET_CFG, 44
P2_FW_DM_L4_SET_PROTO, 44
P2_FW_DM_L4_XCHG, 44
P2_FW_DM_L4A, 44
P2_FW_DM_L4A_ACT_CARD, 44
P2_FW_DM_L4A_GET_PROTO_PARM, 44
P2_FW_DM_L4A_INIT, 44
P2_FW_DM_L4A_PPS, 45
P2_FW_DM_L4A_RATS, 45
P2_FW_DM_LENGTH_LSB_IN, 45
P2_FW_DM_LENGTH_LSB_OUT, 45
P2_FW_DM_LENGTH_MSB_IN, 45
P2_FW_DM_LENGTH_MSB_OUT, 45
p2_fw_dm_mfc_auth_hal_key_store, 50
P2_FW_DM_OFFSET_IN, 45
P2_FW_DM_OFFSET_OUT, 46
P2_FW_DM_OK, 46
P2_FW_DM_RO, 46
P2_FW_DM_RO_CONF_OVER, 46
P2_FW_DM_RO_FIELD_OFF, 46
P2_FW_DM_RO_FIELD_ON, 46
P2_FW_DM_RO_FIELD_RESET, 46
P2_FW_DM_RO_GET_CONF, 46
P2_FW_DM_RO_GET_STATUS, 47
P2_FW_DM_RO_LEDS_OFF, 47
P2_FW_DM_RO_LEDS_ON, 47
P2_FW_DM_RO_READ_REG, 47
P2_FW_DM_RO_RESET, 47
P2_FW_DM_RO_SET_CONF, 47
P2_FW_DM_RO_SET_PCSC_MODE, 47
P2_FW_DM_RO_TEST_MODE, 47
P2_FW_DM_RO_WRITE_REG, 48
P2_FW_DM_STATUS_LSB, 48
P2_FW_DM_STATUS_MSB, 48
p2_fw_dm_xcgh_l4, 51
P2_FW_DM_XCHG, 48
P2_FW_DM_XCHG_INIT, 48
P2_FW_DM_XCHG_L3, 48
P2_FW_DM_XCHG_L4, 48
P2_FW_DM_XCHG_MFC_AUTH, 48

P2_FW_DM_XCHG_MFC_-
AUTH_KEY, 49
P2_FW_DM_XCHG_PC, 49
P2_FW_DM_XCHG_RAW, 49
p2_fw_key_store_get_key, 51
dm_int
p2_fw_direct_mode_xchg, 65
p2_fw_dm_check_if_vaild, 66
p2_fw_dm_check_max_len, 66
P2_FW_DM_CHK_LEN, 55
p2_fw_dm_cid, 66
p2_fw_dm_contact_card, 67
p2_fw_dm_hal, 67
p2_fw_dm_hal_set_cfg_rxdatarate,
67
p2_fw_dm_hal_set_cfg_timeout, 68
p2_fw_dm_hal_set_cfg_txdatarate,
68
p2_fw_dm_hal_switch_config, 68
p2_fw_dm_l3, 68
p2_fw_dm_l4, 69
p2_fw_dm_l4a, 69
p2_fw_dm_mfc_auth_hal, 69
p2_fw_dm_ro, 70
p2_fw_dm_send, 70
p2_fw_dm_translate_error_code, 70
p2_fw_dm_xchg_hal, 71
PH_EXCHANGE_BUFFER_-
CONT, 55
PH_EXCHANGE_BUFFER_-
FIRST, 55
PH_EXCHANGE_BUFFER_-
LAST, 55
PH_RC523_BIT_106KBPS, 56
PH_RC523_BIT_212KBPS, 56
PH_RC523_BIT_424KBPS, 56
PH_RC523_BIT_848KBPS, 56
PH_RC523_MASK_RXALIGN, 56
PH_RC523_MASK_RXWAIT, 56
PH_RC523_MASK_TXBITS, 56
PH_RC523_SERIALSPEED_-
115200, 56
PH_RC523_SERIALSPEED_-
19200, 57
PH_RC523_SERIALSPEED_-
230400, 57
PH_RC523_SERIALSPEED_-
38400, 57
PH_RC523_SERIALSPEED_-
460800, 57
PH_RC523_SERIALSPEED_-
57600, 57
PH_RC523_SERIALSPEED_9600,
57
PHHAL_HW_CARDTYPE_-
CURRENT, 57
PHHAL_HW_CARDTYPE_-
ISO14443A, 57
PHHAL_HW_CARDTYPE_-
ISO14443B, 58
PHHAL_HW_CONFIG_-
ADDITIONAL_INFO, 58
PHHAL_HW_CONFIG_ASK100,
58
PHHAL_HW_CONFIG_BAL_-
CONNECTION, 58
PHHAL_HW_CONFIG_-
CLEARBITSAFTERCOLL,
58
PHHAL_HW_CONFIG_-
DISABLE_MF_CRYPTO1,
58
PHHAL_HW_CONFIG_FIELD_-
OFF_TIME, 58
PHHAL_HW_CONFIG_FIELD_-
RECOVERY_TIME, 58
PHHAL_HW_CONFIG_MAX_-
PRECACHED_BYTES, 59
PHHAL_HW_CONFIG_-
MODINDEX, 59
PHHAL_HW_CONFIG_PARITY,
59
PHHAL_HW_CONFIG_-
RFRESET_ON_TIMEOUT,
59
PHHAL_HW_CONFIG_-
RXALIGN, 59
PHHAL_HW_CONFIG_-
RXBUFFER_BUFSIZE,
59
PHHAL_HW_CONFIG_-
RXBUFFER_STARTPOS,
59
PHHAL_HW_CONFIG_RXCRC,
59
PHHAL_HW_CONFIG_-
RXDATARATE, 60
PHHAL_HW_CONFIG_-
RXDEAFBITS, 60
PHHAL_HW_CONFIG_-

RXLASTBITS, 60
PHHAL_HW_CONFIG_SERIAL_-BITRATE, 60
PHHAL_HW_CONFIG_-SUBCARRIER, 60
PHHAL_HW_CONFIG_-SYMBOL_END, 60
PHHAL_HW_CONFIG_-SYMBOL_START, 60
PHHAL_HW_CONFIG_-TIMEOUT_VALUE_MS, 60
PHHAL_HW_CONFIG_-TIMEOUT_VALUE_US, 61
PHHAL_HW_CONFIG_TIMING_-MODE, 61
PHHAL_HW_CONFIG_TIMING_-MS, 61
PHHAL_HW_CONFIG_TIMING_-US, 61
PHHAL_HW_CONFIG_-TXBUFFER, 61
PHHAL_HW_CONFIG_-TXBUFFER_BUFSIZE, 61
PHHAL_HW_CONFIG_-TXBUFFER_LENGTH, 61
PHHAL_HW_CONFIG_TXCRC, 61
PHHAL_HW_CONFIG_-TXDATARATE, 62
PHHAL_HW_CONFIG_-TXLASTBITS, 62
PHHAL_HW_CONFIG_-TXWAIT_US, 62
PHHAL_HW_MFC_KEYA, 62
PHHAL_HW_MFC_KEYB, 62
PHHAL_HW_MFC_USE_-KEYMODIFIER, 62
PHHAL_HW_RF_DATARATE_-106, 62
PHHAL_HW_RF_DATARATE_-212, 62
PHHAL_HW_RF_DATARATE_-424, 63
PHHAL_HW_RF_DATARATE_-848, 63
PHHAL_HW_RS232_BITRATE_-115200, 63
PHHAL_HW_RS232_BITRATE_-19200, 63
PHHAL_HW_RS232_BITRATE_-230400, 63
PHHAL_HW_RS232_BITRATE_-38400, 63
PHHAL_HW_RS232_BITRATE_-460800, 63
PHHAL_HW_RS232_BITRATE_-57600, 63
PHHAL_HW_RS232_BITRATE_-9600, 64
PHHAL_HW_TIMING_MODE_-COMM, 64
PHHAL_HW_TIMING_MODE_-FDT, 64
PHHAL_HW_TIMING_MODE_-OFF, 64
PHPAL_I14443P4_PARAM_-BLOCKNO, 64
PHPAL_I14443P4_PARAM_CID, 64
PHPAL_I14443P4_PARAM_FSI, 64
PHPAL_I14443P4_PARAM_FWI, 64
PHPAL_I14443P4_PARAM_-MAXRETRYCOUNT, 65
PHPAL_I14443P4_PARAM_NAD, 65
SELECT_CASCADE_LEVEL_1, 65
SELECT_CASCADE_LEVEL_2, 65
SELECT_CASCADE_LEVEL_3, 65
SINGLE_UID_LENGTH, 65

dri
_p2_fw_slot, 93

dsi
_p2_fw_slot, 93

dsl_p
_p2_fw_data, 84

dwTimingUs
_p2_fw_slot, 93

EINT3_IRQHandler
p2_fw_sam_t1.c, 170
p2_fw_sam_t1.h, 133

etu
p2_fw_SAM_ctrl_, 98
exc_p
_p2_fw_data, 84
ext_int
p2_fw_serial_init_serial, 72
P2_FW_USB_BULK_IN_EP, 72
P2_FW_USB_BULK_OUT_EP, 72
p2_fw_usb_init_usb, 72
P2_FW_USB_INT_IN_EP, 72
External Interfaces, 71

FI_DI
sam_t1_param, 100
float32_t
p2_fw.h, 116

g_rp
_p2_fw_data, 85
gm_rp
_p2_fw_data, 85
GuardTime
sam_t1_param, 100

historicalC
cpot_atr_frame, 96
hlt_p
_p2_fw_data, 85

IAP
p2_fw_flash_utils.c, 160
IAP_BLANK_CHECK_SECTOR
p2_fw_flash_utils.c, 159
IAP_COMPARE
p2_fw_flash_utils.c, 159
IAP_COPY_RAM_TO_FLASH
p2_fw_flash_utils.c, 159
IAP_ERASE_SECTOR
p2_fw_flash_utils.c, 159
IAP_LOCATION
p2_fw_flash_utils.c, 159
IAP_PREP_SECTORS
p2_fw_flash_utils.c, 159
IAP_READ_BOOT_CODE_VER
p2_fw_flash_utils.c, 159
IAP_READ_DEV_SER_NUM
p2_fw_flash_utils.c, 159
IAP_READ_ID
p2_fw_flash_utils.c, 159
IAP_REINVOKE
p2_fw_flash_utils.c, 159
IAP_SEC_28
p2_fw_flash_utils.c, 159
IAP_SEC_29
p2_fw_flash_utils.c, 160
IFSC
sam_t1_param, 101
in
_p2_fw_data, 85
in_ccid_len
_p2_fw_data, 85
in_idx
_p2_fw_data, 85
in_len
_p2_fw_data, 85
include/p2_fw.h, 103
include/p2_fw_ccid.h, 120
include/p2_fw_config.h, 123
include/p2_fw_direct_mode.h, 123
include/p2_fw_direct_mode_int.h, 127
include/p2_fw_ext.h, 130
include/p2_fw_peridot_pins.h, 131
include/p2_fw_sam_t1.h, 132
include/p2_fw_timing.h, 136
include/p2_fw_version.h, 137
Internal Direct Mode, 52
ipc
_p2_fw_data, 85
is_busy
_p2_fw_data, 86
iso14443_3
_p2_fw_data, 86
iso14443_3_data
_p2_fw_data, 86
iso14443_4
_p2_fw_data, 86
_p2_fw_slot, 93
iso14443_4_data
_p2_fw_data, 86
iso14443_4A_data
_p2_fw_data, 86

key
_p2_fw_data, 86
KEY_COUNT
p2_fw_direct_mode_key_store.c, 150
KEY_VERSIONS
p2_fw_direct_mode_key_store.c, 150

last_notify_evt
_p2_fw_data, 86
len
_p2_fw_data, 86
_p2_fw_slot, 94
p2_fw_SAM_ctrl_, 99

m_BlockNumber
_p2_fw_slot, 94
m_Flags
_p2_fw_slot, 94
m_HistoryFlag
_p2_fw_slot, 94
m_rp
_p2_fw_data, 86
m_State
_p2_fw_slot, 94
main
p2_fw_main.c, 162
max_retrys
_p2_fw_slot, 94
max_slots
_p2_fw_data, 86
mfrd
_p2_fw_data, 87
mifare
_p2_fw_data, 87
mode
p2_fw_SAM_ctrl_, 99
mutex_transmission
_p2_fw_data, 87

nad
_p2_fw_slot, 94
num
_p2_fw_data, 87

out
_p2_fw_data, 87
out_len
_p2_fw_data, 87
out_written
_p2_fw_data, 87

p2_fw.h
P2_FW_CARD_TYPE_-
ISO14443_3A, 117
P2_FW_CARD_TYPE_-
ISO14443_4A, 117
P2_FW_CARD_TYPE_NONE, 117
P2_FW_CARD_TYPE_SAM, 117
P2_FW_CARD_TYPE_ISO14443_3A
p2_fw.h, 117
P2_FW_CARD_TYPE_ISO14443_4A
p2_fw.h, 117
P2_FW_CARD_TYPE_NONE
p2_fw.h, 117
P2_FW_CARD_TYPE_SAM
p2_fw.h, 117
p2_fw
p2_fw.h, 120
p2_fw_main.c, 163
p2_fw.h
__float32_t_defined, 108
float32_t, 116
p2_fw, 120
P2_FW_APDU_CC_EXT_INS, 108
P2_FW_APDU_CC_EXT_-
MANAGE_SESSION, 108
P2_FW_APDU_CC_EXT_-
SWITCH_PROTOCOL,
108
P2_FW_APDU_CC_EXT_-
TRANS_EXCHANGE, 108
P2_FW_APDU_CLASS, 108
P2_FW_APDU_G_AUTH_CMD_-
INS, 108
P2_FW_APDU_GET_DATA_INS,
108
P2_FW_APDU_INS, 108
P2_FW_APDU_Lc, 109
P2_FW_APDU_LOAD_KEY_INS,
109
P2_FW_APDU_P1, 109
P2_FW_APDU_P2, 109
P2_FW_APDU_READ_BIN, 109
P2_FW_APDU_UPDATE_BIN, 109
P2_FW_CARD_IN_SLOT, 109
P2_FW_CARD_OUT_SLOT, 109
p2_fw_card_type, 117
p2_fw_ccid_xfr_clear_busy, 117
p2_fw_ccid_xfr_set_busy, 117
P2_FW_CFG_BOOTLOADER_-
ACTIVE, 110
P2_FW_CFG_BOOTLOADER_-
VERSION, 110
P2_FW_CFG_CONT_TIMING,
110
P2_FW_CFG_GET_CONT_-
TIMING, 110

P2_FW_CFG_MAX_SLOTS, 110
P2_FW_CFG_SET_DIP_-
SWITCHES, 110
p2_fw_data, 116
P2_FW_DEFAULT_SLOTS, 110
P2_FW_ERR_AND_NFO_LOOP_-
BFL_ERROR, 110
P2_FW_ERR_AND_NFO_LOOP_-
CHIP_NOT_DETECTED,
110
P2_FW_ERR_AND_NFO_LOOP_-
COM_ERROR, 111
P2_FW_ERR_AND_NFO_LOOP_-
DONE, 111
P2_FW_ERR_AND_NFO_LOOP_-
ERASE_FAILD, 111
P2_FW_ERR_AND_NFO_LOOP_-
FLASH_FAILD, 111
P2_FW_ERR_AND_NFO_LOOP_-
HAL_CAN_NOT_BE_SET,
111
P2_FW_ERR_AND_NFO_LOOP_-
HW_ERROR, 111
P2_FW_ERR_AND_NFO_LOOP_-
OS_ERROR, 111
P2_FW_ERR_AND_NFO_LOOP_-
UNKNOWN_ERROR, 111
P2_FW_EXT_COM_INTF_-
ETHERNET, 112
P2_FW_EXT_COM_INTF_RS232,
112
P2_FW_EXT_COM_INTF_RS485,
112
P2_FW_EXT_COM_INTF_USB,
112
p2_fw_flash_erase_config, 117
p2_fw_flash_get_config, 117
p2_fw_flash_read_serial, 118
p2_fw_flash_set_config, 118
P2_FW_FSI, 112
P2_FW_INT_COM_INTF_I2C, 112
P2_FW_INT_COM_INTF_SPI, 112
P2_FW_INT_COM_INTF_UART,
113
p2_fw_invoke_error_mode, 118
P2_FW_MAX_ATS, 113
P2_FW_MAX_COMM_BUFF_-
SIZE, 113
P2_FW_MAX_EXCH_BUFF_-
SIZE, 113
P2_FW_MAX_PCSC_KEY_LEN,
113
P2_FW_MAX_RETRY_-
ISO14443_4A, 113
P2_FW_MAX_SLOT_USB_-
BUFF_SIZE, 113
P2_FW_MAX_SLOTS, 113
P2_FW_MAX_UID, 114
P2_FW_MODE_DEMO, 114
P2_FW_MODE_ENT_SEC_BOOT,
114
P2_FW_MODE_OVER_USER_-
CFG, 114
P2_FW_MODE_PCSC, 114
p2_fw_pcsc_commands, 118
p2_fw_pcsc_exec, 116
P2_FW_PCSC_MODE_DIRECT_-
MODE, 114
P2_FW_PCSC_MODE_-
STANDARD, 114
p2_fw_pcsc_prepare_l3_card, 119
P2_FW_PCSC_PROTO_-
PARAMS_LEN_T0, 114
P2_FW_PCSC_PROTO_-
PARAMS_LEN_T1, 115
P2_FW_PCSC_PROTO_RAW, 115
P2_FW_PCSC_PROTO_T0, 115
P2_FW_PCSC_PROTO_T1, 115
p2_fw_pcsc_send_apdu, 119
P2_FW_PCSC_SHADOW_REGS,
115
P2_FW_READER_CHIP_RC523,
115
P2_FW_READER_CHIP_SAM,
115
P2_FW_READER_MODE_NO_-
SAM, 115
P2_FW_READER_MODE_SAM_-
IN_X, 116
P2_FW_READER_MODE_SAM_-
NON_X, 116
p2_fw_slot, 117
P2_FW_TMR_MS, 116
P2_FW_TMR_START_NOW, 116
P2_FW_TMR_US, 116
p2_fw_utils_dump_regs, 119
phcsBflBal_Hw1SerCM3Init, 119
UART3_IRQHandler, 119
P2_FW_APDU_CC_EXT_INS
p2_fw.h, 108

P2_FW_APDU_CC_EXT_MANAGE_-SESSION
p2_fw.h, 108
P2_FW_APDU_CC_EXT_SWITCH_-PROTOCOL
p2_fw.h, 108
P2_FW_APDU_CC_EXT_TRANS_-EXCHANGE
p2_fw.h, 108
P2_FW_APDU_CLASS
p2_fw.h, 108
P2_FW_APDU_G_AUTH_CMD_INS
p2_fw.h, 108
P2_FW_APDU_GET_DATA_INS
p2_fw.h, 108
P2_FW_APDU_INS
p2_fw.h, 108
P2_FW_APDU_Lc
p2_fw.h, 109
P2_FW_APDU_LOAD_KEY_INS
p2_fw.h, 109
P2_FW_APDU_P1
p2_fw.h, 109
P2_FW_APDU_P2
p2_fw.h, 109
P2_FW_APDU_READ_BIN
p2_fw.h, 109
P2_FW_APDU_UPDATE_BIN
p2_fw.h, 109
p2_fw_blf.c
PHHAL_HW_RC523_ETU_106, 142
PHHAL_HW_RC523_TIMER_-FREQ, 142
PHHAL_HW_RC523_TIMER_-SHIFT, 142
p2_fw_blf_change_rc523_baud_rate
bfl_4_8, 12
p2_fw_blf_init
bfl_4_8, 12
p2_fw_blf_reset_reader
bfl_4_8, 12
p2_fw_blf_set_com_speed
bfl_4_8, 12
p2_fw_blf_set_timeout
bfl_4_8, 13
p2_fw_blf_set_up_rc_type_a_reading
bfl_4_8, 13
P2_FW_CARD_IN_SLOT
p2_fw.h, 109
P2_FW_CARD_OUT_SLOT
p2_fw.h, 109
p2_fw_card_type
p2_fw.h, 117
P2_FW_CCID_BULK_HEADER
ccid, 21
P2_FW_CCID_BULK_OUT_REQ_-ESCAPE
ccid, 21
P2_FW_CCID_BULK_OUT_REQ_-GETPARAMETERS
ccid, 22
P2_FW_CCID_BULK_OUT_REQ_-GETSLOTSTATUS
ccid, 22
P2_FW_CCID_BULK_OUT_REQ_-ICCPOWEROFF
ccid, 22
P2_FW_CCID_BULK_OUT_REQ_-ICCPOWERON
ccid, 22
P2_FW_CCID_BULK_OUT_REQ_-SETPARAMETERS
ccid, 22
P2_FW_CCID_BULK_OUT_REQ_-XFRBLOCK
ccid, 22
p2_fw_ccid_check_header
ccid, 25
P2_FW_CCID_ERROR_SLOT_BUSY
ccid, 22
P2_FW_CCID_ERROR_SLOT_CMD_-NOT_SUPPORTED
ccid, 22
P2_FW_CCID_ERROR_SLOT_HW_-ERROR
ccid, 23
P2_FW_CCID_ERROR_SLOT_ICC_-MUTE
ccid, 23
P2_FW_CCID_ERROR_SLOT_NOT_-EXIST
ccid, 23
P2_FW_CCID_ERROR_SLOT_XFR_-OVERRUN
ccid, 23
p2_fw_ccid_escape_bottom_half
ccid, 25
p2_fw_ccid_escape_top_half
ccid, 26

p2_fw_ccid_get_parameters_bottom_half
ccid, 26
p2_fw_ccid_get_parameters_top_half
ccid, 26
p2_fw_ccid_get_slot_status_bottom_half
ccid, 26
p2_fw_ccid_get_slot_status_top_half
ccid, 27
P2_FW_CCID_HEADER_LENGTH_-
BYTE_1
ccid, 23
P2_FW_CCID_HEADER_LENGTH_-
BYTE_2
ccid, 23
P2_FW_CCID_HEADER_LENGTH_-
BYTE_3
ccid, 23
P2_FW_CCID_HEADER_LENGTH_-
BYTE_4
ccid, 23
P2_FW_CCID_HEADER_MESSAGE_-
TYPE
ccid, 24
P2_FW_CCID_HEADER_MSG_-
BYTE_1
ccid, 24
P2_FW_CCID_HEADER_MSG_-
BYTE_2
ccid, 24
P2_FW_CCID_HEADER_MSG_-
BYTE_3
ccid, 24
P2_FW_CCID_HEADER_SEQ
ccid, 24
P2_FW_CCID_HEADER_SLOT
ccid, 24
p2_fw_ccid_icc_power_off_bottom_half
ccid, 27
p2_fw_ccid_icc_power_off_top_half
ccid, 27
p2_fw_ccid_icc_power_on_bottom_half
ccid, 27
p2_fw_ccid_icc_power_on_top_half
ccid, 28
P2_FW_CCID_INT_IN_NOTIFY_-
SLOT_CHANGE
ccid, 24
P2_FW_CCID_RDR_TO_PC_ESCAPE
ccid, 24
P2_FW_CCID_RDR_TO_PC_-
PARAMETERS
ccid, 25
P2_FW_CCID_RDR_TO_PC_SLOT_-
DATA_BLOCK
ccid, 25
P2_FW_CCID_RDR_TO_PC_SLOT_-
STATUS
ccid, 25
p2_fw_ccid_send_apdu
ccid, 28
p2_fw_ccid_send_data
ccid, 28
p2_fw_ccid_send_notify
ccid, 29
p2_fw_ccid_set_parameters_bottom_half
ccid, 29
p2_fw_ccid_set_parameters_top_half
ccid, 29
P2_FW_CCID_STATUS_CMD_-
FAILED
ccid, 25
p2_fw_ccid_top_half_dispatch
ccid, 30
p2_fw_ccid_xfer.c
p2_fw_pcsc_prepare_l3_card, 144
p2_fw_pcsc_send_apdu, 144
p2_fw_ccid_xfr_block_bottom_half
ccid, 30
p2_fw_ccid_xfr_block_top_half
ccid, 30
p2_fw_ccid_xfr_clear_busy
ccid, 30
p2_fw.h, 117
p2_fw_ccid_xfr_set_busy
ccid, 31
p2_fw.h, 117
P2_FW_CFG_BOOTLOADER_-
ACTIVE
p2_fw.h, 110
P2_FW_CFG_BOOTLOADER_-
VERSION
p2_fw.h, 110
P2_FW_CFG_CONT_TIMING
p2_fw.h, 110
P2_FW_CFG_COUNT
p2_fw_flash_utils.c, 160
P2_FW_CFG_ENTRY_OFFSET
p2_fw_flash_utils.c, 160
P2_FW_CFG_ENTRY_SIZE

p2_fw_flash_utils.c, 160
P2_FW_CFG_GET_CONT_TIMING
p2_fw.h, 110
P2_FW_CFG_MAX
p2_fw_flash_utils.c, 160
P2_FW_CFG_MAX_SLOTS
p2_fw.h, 110
P2_FW_CFG_S1
p2_fw_flash_utils.c, 160
P2_FW_CFG_S2
p2_fw_flash_utils.c, 160
P2_FW_CFG_SET_DIP_SWITCHES
p2_fw.h, 110
P2_FW_CONFIG_COMM_SER_232_-PORT
config, 31
P2_FW_CONFIG_COMM_SER_485_-PORT
config, 31
P2_FW_CONFIG_DEBUG_PORT
config, 32
P2_FW_CONFIG_RC523_UART_PORT
config, 32
p2_fw_data
p2_fw.h, 116
P2_FW_DEFAULT_SLOTS
p2_fw.h, 110
p2_fw_direct_mode_key_store.c
KEY_COUNT, 150
KEY_VERSIONS, 150
p2_fw_direct_mode_l4.c
PHPAL_I14443P4_CID_MAX, 152
PHPAL_I14443P4_FRAMESIZE_-MAX, 152
PHPAL_I14443P4_FWI_MAX, 152
p2_fw_direct_mode_xchg
dm_int, 65
p2_fw_dm
dm, 49
P2_FW_DM_ALLOWED_CMDS_ALL
dm, 35
P2_FW_DM_ALLOWED_CMDS_CID
dm, 35
P2_FW_DM_ALLOWED_CMDS_-CONTACT_CARD
dm, 35
P2_FW_DM_ALLOWED_CMDS_HAL
dm, 35
P2_FW_DM_ALLOWED_CMDS_-KSTOR
dm, 36
P2_FW_DM_ALLOWED_CMDS_L3
dm, 36
P2_FW_DM_ALLOWED_CMDS_L4
dm, 36
P2_FW_DM_ALLOWED_CMDS_L4A
dm, 36
P2_FW_DM_ALLOWED_CMDS_RO
dm, 36
P2_FW_DM_ALLOWED_CMDS_-XCHG
dm, 36
p2_fw_dm_check_if_vaild
dm_int, 66
p2_fw_dm_check_max_len
dm_int, 66
P2_FW_DM_CHK_LEN
dm_int, 55
P2_FW_DM_CID
dm, 36
p2_fw_dm_cid
dm_int, 66
P2_FW_DM_CID_FREE
dm, 36
P2_FW_DM_CID_GET_FREE
dm, 37
P2_FW_DM_CID_INIT
dm, 37
P2_FW_DM_CLASS_BYTE
dm, 37
P2_FW_DM_CONTACT_CARD
dm, 37
p2_fw_dm_contact_card
dm_int, 67
P2_FW_DM_CONTACTCARD_-ACTIVATE_CARD
dm, 37
P2_FW_DM_CONTACTCARD_-CLOCK_START
dm, 37
P2_FW_DM_CONTACTCARD_-CLOCK_STOP
dm, 37
P2_FW_DM_CONTACTCARD_-COLD_RESET
dm, 37
P2_FW_DM_CONTACTCARD_-DEACTIVATE_CARD
dm, 38
P2_FW_DM_CONTACTCARD_PPS

dm, 38
P2_FW_DM_CONTACTCARD_-PRESENCE_CHECK
dm, 38
P2_FW_DM_CONTACTCARD_-TRANSMIT_DATA
dm, 38
P2_FW_DM_CONTACTCARD_-WARM_RESET
dm, 38
P2_FW_DM_FAILD
dm, 38
P2_FW_DM_HAL
dm, 38
p2_fw_dm_hal
dm_int, 67
P2_FW_DM_HAL_APP_PROT_SET
dm, 38
P2_FW_DM_HAL_EXEC_CMD
dm, 39
P2_FW_DM_HAL_GET_CFG
dm, 39
P2_FW_DM_HAL_INIT
dm, 39
P2_FW_DM_HAL_MFC_AUTH
dm, 39
P2_FW_DM_HAL_MFC_AUTH_KEY
dm, 39
P2_FW_DM_HAL_SET_CFG
dm, 39
p2_fw_dm_hal_set_cfg_rxdatarate
dm_int, 67
p2_fw_dm_hal_set_cfg_timeout
dm_int, 68
p2_fw_dm_hal_set_cfg_txdatarate
dm_int, 68
p2_fw_dm_hal_switch_config
dm_int, 68
P2_FW_DM_HAL_WAIT
dm, 39
p2_fw_dm_hal_wait
dm, 49
P2_FW_DM_HAL_XCHG
dm, 39
P2_FW_DM_INSTR_BYTE
dm, 40
p2_fw_dm_key_store
dm, 50
p2_fw_dm_key_store_init
dm, 50
P2_FW_DM_KSTOR
dm, 40
P2_FW_DM_KSTOR_CHG_KUC
dm, 40
P2_FW_DM_KSTOR_FORMAT_KEY
dm, 40
P2_FW_DM_KSTOR_GET_CFG_STR
dm, 40
P2_FW_DM_KSTOR_GET_CONFIG
dm, 40
P2_FW_DM_KSTOR_GET_KEY
dm, 40
P2_FW_DM_KSTOR_GET_KEY_-ENTRY
dm, 40
P2_FW_DM_KSTOR_GET_KUC
dm, 41
P2_FW_DM_KSTOR_INIT
dm, 41
P2_FW_DM_KSTOR_SET_CEK
dm, 41
P2_FW_DM_KSTOR_SET_CFG_STR
dm, 41
P2_FW_DM_KSTOR_SET_CONFIG
dm, 41
P2_FW_DM_KSTOR_SET_FULL_-KEY
dm, 41
P2_FW_DM_KSTOR_SET_KEY
dm, 41
P2_FW_DM_KSTOR_SET_KEY_POS
dm, 41
P2_FW_DM_KSTOR_SET_KUC
dm, 42
P2_FW_DM_L3
dm, 42
p2_fw_dm_l3
dm_int, 68
P2_FW_DM_L3_ACT_CARD
dm, 42
P2_FW_DM_L3_ANTICOL
dm, 42
P2_FW_DM_L3_GET_SER
dm, 42
P2_FW_DM_L3_HLTA
dm, 42
P2_FW_DM_L3_INIT
dm, 42
P2_FW_DM_L3_REQA
dm, 42

P2_FW_DM_L3_SELECT
dm, 43
P2_FW_DM_L3_WKUA
dm, 43
P2_FW_DM_L3_XCHG
dm, 43
P2_FW_DM_L4
dm, 43
p2_fw_dm_l4
dm_int, 69
P2_FW_DM_L4_DESELECT
dm, 43
P2_FW_DM_L4_GET_CFG
dm, 43
P2_FW_DM_L4_INIT
dm, 43
P2_FW_DM_L4_PRES_CHECK
dm, 43
P2_FW_DM_L4_RESET_PROTO
dm, 44
P2_FW_DM_L4_SET_CFG
dm, 44
P2_FW_DM_L4_SET_PROTO
dm, 44
P2_FW_DM_L4_XCHG
dm, 44
P2_FW_DM_L4A
dm, 44
p2_fw_dm_l4a
dm_int, 69
P2_FW_DM_L4A_ACT_CARD
dm, 44
P2_FW_DM_L4A_GET_PROTO_-
PARM
dm, 44
P2_FW_DM_L4A_INIT
dm, 44
P2_FW_DM_L4A_PPS
dm, 45
P2_FW_DM_L4A_RATS
dm, 45
P2_FW_DM_LENGTH_LSB_IN
dm, 45
P2_FW_DM_LENGTH_LSB_OUT
dm, 45
P2_FW_DM_LENGTH_MSB_IN
dm, 45
P2_FW_DM_LENGTH_MSB_OUT
dm, 45
p2_fw_dm_mfc_auth_hal
dm_int, 69
p2_fw_dm_mfc_auth_hal_key_store
dm, 50
P2_FW_DM_OFFSET_IN
dm, 45
P2_FW_DM_OFFSET_OUT
dm, 46
P2_FW_DM_OK
dm, 46
P2_FW_DM_RO
dm, 46
p2_fw_dm_ro
dm_int, 70
P2_FW_DM_RO_CONF_OVER
dm, 46
P2_FW_DM_RO_FIELD_OFF
dm, 46
P2_FW_DM_RO_FIELD_ON
dm, 46
P2_FW_DM_RO_FIELD_RESET
dm, 46
P2_FW_DM_RO_GET_CONF
dm, 46
P2_FW_DM_RO_GET_STATUS
dm, 47
P2_FW_DM_RO_LEDS_OFF
dm, 47
P2_FW_DM_RO_LEDS_ON
dm, 47
P2_FW_DM_RO_READ_REG
dm, 47
P2_FW_DM_RO_RESET
dm, 47
P2_FW_DM_RO_SET_CONF
dm, 47
P2_FW_DM_RO_SET_PCSC_MODE
dm, 47
P2_FW_DM_RO_TEST_MODE
dm, 47
P2_FW_DM_RO_WRITE_REG
dm, 48
p2_fw_dm_send
dm_int, 70
P2_FW_DM_STATUS_LSB
dm, 48
P2_FW_DM_STATUS_MSB
dm, 48
p2_fw_dm_translate_error_code
dm_int, 70
p2_fw_dm_xcgh_l4

dm, 51
P2_FW_DM_XCHG
dm, 48
p2_fw_dm_xchg_hal
dm_int, 71
P2_FW_DM_XCHG_INIT
dm, 48
P2_FW_DM_XCHG_L3
dm, 48
P2_FW_DM_XCHG_L4
dm, 48
P2_FW_DM_XCHG_MFC_AUTH
dm, 48
P2_FW_DM_XCHG_MFC_AUTH_-KEY
dm, 49
P2_FW_DM_XCHG_PC
dm, 49
P2_FW_DM_XCHG_RAW
dm, 49
P2_FW_ERR_AND_NFO_LOOP_-BFL_ERROR
p2_fw.h, 110
P2_FW_ERR_AND_NFO_LOOP_-CHIP_NOT_DETECTED
p2_fw.h, 110
P2_FW_ERR_AND_NFO_LOOP_-COM_ERROR
p2_fw.h, 111
P2_FW_ERR_AND_NFO_LOOP_-DONE
p2_fw.h, 111
P2_FW_ERR_AND_NFO_LOOP_-ERASE_FAILD
p2_fw.h, 111
P2_FW_ERR_AND_NFO_LOOP_-FLASH_FAILD
p2_fw.h, 111
P2_FW_ERR_AND_NFO_LOOP_-HAL_CAN_NOT_BE_SET
p2_fw.h, 111
P2_FW_ERR_AND_NFO_LOOP_-HW_ERROR
p2_fw.h, 111
P2_FW_ERR_AND_NFO_LOOP_OS_-ERROR
p2_fw.h, 111
P2_FW_ERR_AND_NFO_LOOP_-UNKNOWN_ERROR
p2_fw.h, 111
P2_FW_EXT_COM_INTF_ETHERNET
p2_fw.h, 112
P2_FW_EXT_COM_INTF_RS232
p2_fw.h, 112
P2_FW_EXT_COM_INTF_RS485
p2_fw.h, 112
P2_FW_EXT_COM_INTF_USB
p2_fw.h, 112
p2_fw_ext_intf_serial.c
SERIAL_UART, 156
UART3_IRQHandler, 156
p2_fw_ext_intf_usb.c
P2_FW_MAX_SLOT_INDEX, 157
P2_FW_NAME_USB_OFFSET, 157
P2_FW_SERNUM_USB_OFFSET, 157
P2_FW_USB_CONTROL_-REQUEST_ABORT, 157
P2_FW_USB_CONTROL_-REQUEST_GET_CLOCK_-FREQUENCIES, 157
P2_FW_USB_CONTROL_-REQUEST_GET_DATA_-RATES, 157
P2_FW_USB_LE_DWORD, 157
P2_FW_USB_LE_WORD, 157
P2_FW_USB_MAX_PACKET_-SIZE, 157
p2_fw_flash_erase_config
p2_fw.h, 117
p2_fw_flash_utils.c, 160
p2_fw_flash_get_config
p2_fw.h, 117
p2_fw_flash_utils.c, 161
p2_fw_flash_read_serial
p2_fw.h, 118
p2_fw_flash_utils.c, 161
p2_fw_flash_set_config
p2_fw.h, 118
p2_fw_flash_utils.c, 161
p2_fw_flash_utils.c
IAP, 160
IAP_BLANK_CHECK_SECTOR, 159
IAP_COMPARE, 159
IAP_COPY_RAM_TO_FLASH, 159
IAP_ERASE_SECTOR, 159
IAP_LOCATION, 159

IAP_PREP_SECTORS, 159
IAP_READ_BOOT_CODE_VER, 159
IAP_READ_DEV_SER_NUM, 159
IAP_READ_ID, 159
IAP_REINVOKE, 159
IAP_SEC_28, 159
IAP_SEC_29, 160
P2_FW_CFG_COUNT, 160
P2_FW_CFG_ENTRY_OFFSET, 160
P2_FW_CFG_ENTRY_SIZE, 160
P2_FW_CFG_MAX, 160
P2_FW_CFG_S1, 160
P2_FW_CFG_S2, 160
p2_fw_flash_erase_config, 160
p2_fw_flash_get_config, 161
p2_fw_flash_read_serial, 161
p2_fw_flash_set_config, 161
P2_FW_FSI
p2_fw.h, 112
P2_FW_INT_COM_INTF_I2C
p2_fw.h, 112
P2_FW_INT_COM_INTF_SPI
p2_fw.h, 112
P2_FW_INT_COM_INTF_UART
p2_fw.h, 113
p2_fw_invoke_error_mode
p2_fw.h, 118
p2_fw_main.c, 162
p2_fw_key_store_get_key
dm, 51
p2_fw_main.c
main, 162
p2_fw, 163
p2_fw_invoke_error_mode, 162
P2_FW_MAX_ATS
p2_fw.h, 113
P2_FW_MAX_COMM_BUFF_SIZE
p2_fw.h, 113
P2_FW_MAX_EXCH_BUFF_SIZE
p2_fw.h, 113
P2_FW_MAX_PCSC_KEY_LEN
p2_fw.h, 113
P2_FW_MAX_RETRY_ISO14443_4A
p2_fw.h, 113
P2_FW_MAX_SLOT_INDEX
p2_fw_ext_intf_usb.c, 157
P2_FW_MAX_SLOT_USB_BUFF_SIZE
p2_fw.h, 113
P2_FW_MAX_SLOTS
p2_fw.h, 113
P2_FW_MAX_UID
p2_fw.h, 114
P2_FW_MODE_DEMO
p2_fw.h, 114
P2_FW_MODE_ENT_SEC_BOOT
p2_fw.h, 114
P2_FW_MODE_OVER_USER_CFG
p2_fw.h, 114
P2_FW_MODE_PCSC
p2_fw.h, 114
P2_FW_NAME_USB_OFFSET
p2_fw_ext_intf_usb.c, 157
p2_fw_pcsc_commands
p2_fw.h, 118
p2_fw_pcsc_ext.c, 163
p2_fw_pcsc_exec
p2_fw.h, 116
p2_fw_pcsc_ext.c
p2_fw_pcsc_commands, 163
P2_FW_PCSC_MODE_DIRECT_MODE
p2_fw.h, 114
P2_FW_PCSC_MODE_STANDARD
p2_fw.h, 114
p2_fw_pcsc_prepare_l3_card
p2_fw.h, 119
p2_fw_ccid_xfer.c, 144
P2_FW_PCSC_PROTO_PARAMS_LEN_T0
p2_fw.h, 114
P2_FW_PCSC_PROTO_PARAMS_LEN_T1
p2_fw.h, 115
P2_FW_PCSC_PROTO_RAW
p2_fw.h, 115
P2_FW_PCSC_PROTO_T0
p2_fw.h, 115
P2_FW_PCSC_PROTO_T1
p2_fw.h, 115
p2_fw_pcsc_send_apdu
p2_fw.h, 119
p2_fw_ccid_xfer.c, 144
P2_FW_PCSC_SHADOW_REGS
p2_fw.h, 115
P2_FW_PINS_ANTENA_BLUE
pins, 73
P2_FW_PINS_ANTENA_GREEN

pins, 73
P2_FW_PINS_ANTENA_RED
pins, 74
P2_FW_PINS_BEEPER
pins, 74
P2_FW_PINS_CFG_1
pins, 74
P2_FW_PINS_CFG_2
pins, 74
P2_FW_PINS_CFG_3
pins, 74
P2_FW_PINS_CFG_4
pins, 74
P2_FW_PINS_DIP_1
pins, 74
P2_FW_PINS_DIP_2
pins, 74
P2_FW_PINS_DIP_3
pins, 75
P2_FW_PINS_DIP_4
pins, 75
P2_FW_PINS_DIP_5
pins, 75
P2_FW_PINS_DIP_6
pins, 75
P2_FW_PINS_DIP_7
pins, 75
P2_FW_PINS_DIP_8
pins, 75
P2_FW_PINS_LEDS_YELLOW_2
pins, 75
P2_FW_PINS_LEDS_YELLOW_3
pins, 75
P2_FW_PINS_LEDS_YELLOW_4
pins, 76
P2_FW_READER_CHIP_RC523
p2_fw.h, 115
P2_FW_READER_CHIP_SAM
p2_fw.h, 115
P2_FW_READER_MODE_NO_SAM
p2_fw.h, 115
P2_FW_READER_MODE_SAM_IN_X
p2_fw.h, 116
P2_FW_READER_MODE_SAM_-
NON_X
p2_fw.h, 116
p2_fw_reader_read_config
reader, 9
p2_fw_reader_set_up_external_interface
reader, 9
p2_fw_reader_set_up_reader_chip
reader, 10
p2_fw_reader_setup_hardware
reader, 10
P2_FW_SAM_CLOCK_LEN_NS
p2_fw_sam_t1.c, 167
P2_FW_SAM_CONVERSION_-
DIRECT
p2_fw_sam_t1.c, 167
P2_FW_SAM_CONVERSION_-
INVERSE
p2_fw_sam_t1.c, 167
P2_FW_SAM_CONVERSION_-
UNKNOWN
p2_fw_sam_t1.c, 167
p2_fw_SAM_ctrl_, 97
bitIndex, 98
bwi, 98
chipMode, 98
chipType, 98
conversion, 98
data, 98
dataIndex, 98
etu, 98
len, 99
mode, 99
parityCount, 99
recExtraGuardTime, 99
send_data, 99
sendLen, 99
sendSeqData, 99
SessionATR, 99
SessionATR_Size, 99
t0, 99
timingMode, 99
tmpByteWait_time, 100
P2_FW_SAM_DEBUG
p2_fw_sam_t1.c, 167
P2_FW_SAM_DEFAULT_BWI
p2_fw_sam_t1.c, 167
P2_FW_SAM_FRAME_APDU
p2_fw_sam_t1.h, 133
P2_FW_SAM_FRAME_T1
p2_fw_sam_t1.h, 133
P2_FW_SAM_MAX_ATR_SIZE
p2_fw_sam_t1.c, 168
P2_FW_SAM_MAX_REC_DATALEN
p2_fw_sam_t1.c, 168
P2_FW_SAM_MODE_IDLE
p2_fw_sam_t1.c, 168

P2_FW_SAM_MODE_POWER_OFF
p2_fw_sam_t1.c, 168
P2_FW_SAM_MODE_PPS
p2_fw_sam_t1.h, 133
P2_FW_SAM_MODE_RECEIVE
p2_fw_sam_t1.c, 168
P2_FW_SAM_MODE_RESET
p2_fw_sam_t1.c, 168
P2_FW_SAM_MODE_SEND
p2_fw_sam_t1.c, 168
P2_FW_SAM_MODE_WAIT_START_-
BIT
p2_fw_sam_t1.c, 168
P2_FW_SAM_MODE_X
p2_fw_sam_t1.h, 133
P2_FW_SAM_PWM_CH0_MATCH_-
VALUE
p2_fw_sam_t1.c, 168
P2_FW_SAM_PWM_CH1_MATCH_-
VALUE
p2_fw_sam_t1.c, 168
p2_fw_sam_t1.c
BITBAND_PERI, 167
BITBAND_PERI_BASE, 167
BITBAND_PERI_REF, 167
EINT3_IRQHandler, 170
P2_FW_SAM_CLOCK_LEN_NS,
167
P2_FW_SAM_CONVERSION_-
DIRECT, 167
P2_FW_SAM_CONVERSION_-
INVERSE, 167
P2_FW_SAM_CONVERSION_-
UNKNOWN, 167
P2_FW_SAM_DEBUG, 167
P2_FW_SAM_DEFAULT_BWI,
167
P2_FW_SAM_MAX_ATR_SIZE,
168
P2_FW_SAM_MAX_REC_-
DATALEN, 168
P2_FW_SAM_MODE_IDLE, 168
P2_FW_SAM_MODE_POWER_-
OFF, 168
P2_FW_SAM_MODE_RECEIVE,
168
P2_FW_SAM_MODE_RESET, 168
P2_FW_SAM_MODE_SEND, 168
P2_FW_SAM_MODE_WAIT_-
START_BIT, 168
P2_FW_SAM_PWM_CH0_-
MATCH_VALUE, 168
P2_FW_SAM_PWM_CH1_-
MATCH_VALUE, 168
P2_FW_SAM_T1_CLOCK_-
CONFIG, 169
p2_fw_sam_t1_deinit, 170
p2_fw_sam_t1_get_atr, 170
p2_fw_sam_t1_get_timing_mode,
170
p2_fw_sam_t1_init, 170
p2_fw_sam_t1_is_busy, 170
p2_fw_sam_t1_is_power_off, 170
p2_fw_sam_t1_is_sam_inserted,
170
p2_fw_sam_t1_pps, 170
p2_fw_sam_t1_prepare_pps, 170
p2_fw_sam_t1_receive, 170
p2_fw_sam_t1_send, 171
p2_fw_sam_t1_set_bwi_cwi, 171
p2_fw_sam_t1_set_etu, 171
p2_fw_sam_t1_set_my_debug, 171
p2_fw_sam_t1_set_rec_-
extraGuardTime, 171
p2_fw_sam_t1_set_timing_mode,
171
p2_fw_sam_t1_start, 171
p2_fw_sam_t1_warm_reset, 171
P2_FW_SAM_TIMER_-
PRESCALE_VALUE, 169
T1_BYTE_WAIT_TIME, 169
TIMER0_IR, 169
TIMER0_IR_MR0, 169
TIMER0_IRQHandler, 171
TIMER0_TCR, 169
TIMER0_TCR_ENABLE, 169
TIMER0_TCR_RESET, 169
TIMER2_IRQHandler, 171
p2_fw_sam_t1.h
EINT3_IRQHandler, 133
P2_FW_SAM_FRAME_APDU,
133
P2_FW_SAM_FRAME_T1, 133
P2_FW_SAM_MODE_PPS, 133
P2_FW_SAM_MODE_X, 133
p2_fw_sam_t1_deinit, 133
p2_fw_sam_t1_get_atr, 133
p2_fw_sam_t1_get_timing_mode,
133
p2_fw_sam_t1_init, 134

p2_fw_sam_t1_is_busy, 134
p2_fw_sam_t1_is_power_off, 134
p2_fw_sam_t1_is_sam_inserted, 134
p2_fw_sam_t1_pps, 134
p2_fw_sam_t1_prepare_pps, 134
p2_fw_sam_t1_receive, 134
p2_fw_sam_t1_send, 134
p2_fw_sam_t1_set_bwi_cwi, 134
p2_fw_sam_t1_set_etu, 134
p2_fw_sam_t1_set_my_debug, 135
p2_fw_sam_t1_set_rec_extraGuardTime, 135
p2_fw_sam_t1_set_timing_mode, 135
p2_fw_sam_t1_start, 135
p2_fw_sam_t1_warm_reset, 135
P2_FW_TIMING_MODE_COM, 133
P2_FW_TIMING_MODE_FDT, 133
P2_FW_TIMING_MODE_NONE, 133
TIMER0_IRQHandler, 135
TIMER2_IRQHandler, 135
P2_FW_SAM_T1_CLOCK_CONFIG
p2_fw_sam_t1.c, 169
p2_fw_sam_t1_deinit
p2_fw_sam_t1.c, 170
p2_fw_sam_t1.h, 133
p2_fw_sam_t1_get_atr
p2_fw_sam_t1.c, 170
p2_fw_sam_t1.h, 133
p2_fw_sam_t1_get_timing_mode
p2_fw_sam_t1.c, 170
p2_fw_sam_t1.h, 133
p2_fw_sam_t1_init
p2_fw_sam_t1.c, 170
p2_fw_sam_t1.h, 134
p2_fw_sam_t1_is_busy
p2_fw_sam_t1.c, 170
p2_fw_sam_t1.h, 134
p2_fw_sam_t1_is_power_off
p2_fw_sam_t1.c, 170
p2_fw_sam_t1.h, 134
p2_fw_sam_t1_is_sam_inserted
p2_fw_sam_t1.c, 170
p2_fw_sam_t1.h, 134
p2_fw_sam_t1_pps
p2_fw_sam_t1.c, 170
p2_fw_sam_t1.h, 134
p2_fw_sam_t1_prepare_pps
p2_fw_sam_t1.c, 170
p2_fw_sam_t1.h, 134
p2_fw_sam_t1_receive
p2_fw_sam_t1.c, 170
p2_fw_sam_t1.h, 134
p2_fw_sam_t1_send
p2_fw_sam_t1.c, 171
p2_fw_sam_t1.h, 134
p2_fw_sam_t1_set_bwi_cwi
p2_fw_sam_t1.c, 171
p2_fw_sam_t1.h, 134
p2_fw_sam_t1_set_etu
p2_fw_sam_t1.c, 171
p2_fw_sam_t1.h, 134
p2_fw_sam_t1_set_my_debug
p2_fw_sam_t1.c, 171
p2_fw_sam_t1.h, 135
p2_fw_sam_t1_set_rec_extraGuardTime
p2_fw_sam_t1.c, 171
p2_fw_sam_t1.h, 135
p2_fw_sam_t1_set_timing_mode
p2_fw_sam_t1.c, 171
p2_fw_sam_t1.h, 135
p2_fw_sam_t1_start
p2_fw_sam_t1.c, 171
p2_fw_sam_t1.h, 135
p2_fw_sam_t1_warm_reset
p2_fw_sam_t1.c, 171
p2_fw_sam_t1.h, 135
P2_FW_SAM_TIMER_PRESCALE_VALUE
p2_fw_sam_t1.c, 169
p2_fw_serial_init_serial
ext_int, 72
P2_FW_SERNUM_USB_OFFSET
p2_fw_ext_intf_usb.c, 157
p2_fw_slot
p2_fw.h, 117
p2_fw_slots_add_new_l3_card
slots, 14
p2_fw_slots_add_new_l4_card
slots, 14
p2_fw_slots_add_new_sam_card
slots, 15
p2_fw_slots_clear_cid_list
slots, 15
p2_fw_slots_free_cid
slots, 15

p2_fw_slots_get_atr
slots, 15
p2_fw_slots_get_free_cid
slots, 15
p2_fw_slots_get_free_slot
slots, 16
p2_fw_slots_init
slots, 16
p2_fw_slots_is_known_l3_card
slots, 16
p2_fw_slots_remove_card
slots, 16
p2_fw_task_demo_mode
tasks, 11
p2_fw_task_err_and_nfo_loop
tasks, 11
p2_fw_task_pcsc_execute
tasks, 11
p2_fw_task_pcsc_poll_and_act_loop
tasks, 11
p2_fw_timing_com.c
PHHAL_HW_RC523_DIGI_-
DELAY_US, 173
PHHAL_HW_RC523_ETU_106,
173
PHHAL_HW_RC523_TIMER_-
FREQ, 173
PHHAL_HW_RC523_TIMER_-
SHIFT, 174
TIMING_MODE_OPTION_-
AUTOCLEAR, 174
TIMING_MODE_OPTION_-
MASK, 174
p2_fw_timing_init
timing, 76
P2_FW_TIMING_MODE_COM
p2_fw_sam_t1.h, 133
P2_FW_TIMING_MODE_FDT
p2_fw_sam_t1.h, 133
P2_FW_TIMING_MODE_NONE
p2_fw_sam_t1.h, 133
p2_fw_timing_start
timing, 76
p2_fw_timing_stop_cless
timing, 77
p2_fw_timing_stop_contac
timing, 77
P2_FW_TMR_MS
p2_fw.h, 116
P2_FW_TMR_START_NOW
p2_fw.h, 116
P2_FW_TMR_US
p2_fw.h, 116
P2_FW_USB_BULK_IN_EP
ext_int, 72
P2_FW_USB_BULK_OUT_EP
ext_int, 72
P2_FW_USB_CONTROL_REQUEST_-
ABORT
p2_fw_ext_intf_usb.c, 157
P2_FW_USB_CONTROL_-
REQUEST_GET_CLOCK_-
FREQUENCIES
p2_fw_ext_intf_usb.c, 157
P2_FW_USB_CONTROL_REQUEST_-
GET_DATA_RATES
p2_fw_ext_intf_usb.c, 157
p2_fw_usb_init_usb
ext_int, 72
P2_FW_USB_INT_IN_EP
ext_int, 72
P2_FW_USB_LE_DWORD
p2_fw_ext_intf_usb.c, 157
P2_FW_USB_LE_WORD
p2_fw_ext_intf_usb.c, 157
P2_FW_USB_MAX_PACKET_SIZE
p2_fw_ext_intf_usb.c, 157
p2_fw_utils.c
p2_fw_utils_dump_regs, 175
p2_fw_utils_blink
utils, 17
p2_fw_utils_dump_regs
p2_fw.h, 119
p2_fw_utils.c, 175
p2_fw_utils_field_off
utils, 17
p2_fw_utils_field_on
utils, 17
p2_fw_utils_get_dri
utils, 18
p2_fw_utils_get_dsi
utils, 18
p2_fw_utils_reg_read
utils, 18
p2_fw_utils_reg_write
utils, 18
P2_FW_VER_BUILD
p2_fw_version.h, 137
P2_FW_VER_MAJOR
p2_fw_version.h, 137

P2_FW_VER_MINOR
p2_fw_version.h, 137
p2_fw_version.h
P2_FW_VER_BUILD, 137
P2_FW_VER_MAJOR, 137
P2_FW_VER_MINOR, 137
params
_p2_fw_slot, 94
parityCount
p2_fw_SAM_ctrl_, 99
pcsc_m
_p2_fw_data, 87
pcsc_mode
_p2_fw_data, 88
Pegoda 2 Pins, 73
PH_EXCHANGE_BUFFER_CONT
dm_int, 55
PH_EXCHANGE_BUFFER_FIRST
dm_int, 55
PH_EXCHANGE_BUFFER_LAST
dm_int, 55
PH_RC523_BIT_106KBPS
dm_int, 56
PH_RC523_BIT_212KBPS
dm_int, 56
PH_RC523_BIT_424KBPS
dm_int, 56
PH_RC523_BIT_848KBPS
dm_int, 56
PH_RC523_MASK_RXALIGN
dm_int, 56
PH_RC523_MASK_RXWAIT
dm_int, 56
PH_RC523_MASK_TXBITS
dm_int, 56
PH_RC523_SERIALSPEED_115200
dm_int, 56
PH_RC523_SERIALSPEED_19200
dm_int, 57
PH_RC523_SERIALSPEED_230400
dm_int, 57
PH_RC523_SERIALSPEED_38400
dm_int, 57
PH_RC523_SERIALSPEED_460800
dm_int, 57
PH_RC523_SERIALSPEED_57600
dm_int, 57
PH_RC523_SERIALSPEED_9600
dm_int, 57
phcsBflBal_Hw1SerCM3.c
BUFFER_LENGTH, 138
phcsBflBal_Hw1SerCM3Init, 139
THE_UART, 138
UART_PORT, 138
phcsBflBal_Hw1SerCM3Init
p2_fw.h, 119
phcsBflBal_Hw1SerCM3.c, 139
phcsBflRegCtl_SpiHw1.c
BUFFER_SIZE, 140
phcsBflRegCtl_-
SpiHw1GetMultiReg, 140
phcsBflRegCtl_SpiHw1GetReg, 140
phcsBflRegCtl_SpiHw1Init, 140
phcsBflRegCtl_SpiHw1ModReg,
140
phcsBflRegCtl_-
SpiHw1SetMultiReg, 141
phcsBflRegCtl_SpiHw1SetReg, 141
phcsBflRegCtl_SpiHw1GetMultiReg
phcsBflRegCtl_SpiHw1.c, 140
phcsBflRegCtl_SpiHw1GetReg
phcsBflRegCtl_SpiHw1.c, 140
phcsBflRegCtl_SpiHw1Init
phcsBflRegCtl_SpiHw1.c, 140
phcsBflRegCtl_SpiHw1ModReg
phcsBflRegCtl_SpiHw1.c, 140
phcsBflRegCtl_SpiHw1SetMultiReg
phcsBflRegCtl_SpiHw1.c, 141
phcsBflRegCtl_SpiHw1SetReg
phcsBflRegCtl_SpiHw1.c, 141
PHHAL_HW_CARDTYPE_CURRENT
dm_int, 57
PHHAL_HW_CARDTYPE_ISO14443A
dm_int, 57
PHHAL_HW_CARDTYPE_ISO14443B
dm_int, 58
PHHAL_HW_CONFIG_-
ADDITIONAL_INFO
dm_int, 58
PHHAL_HW_CONFIG_ASK100
dm_int, 58
PHHAL_HW_CONFIG_BAL_-
CONNECTION
dm_int, 58
PHHAL_HW_CONFIG_-
CLEARBITSAFTERCOLL
dm_int, 58
PHHAL_HW_CONFIG_DISABLE_-
MF_CRYPTO1
dm_int, 58

PHHAL_HW_CONFIG_FIELD_OFF_-
TIME
dm_int, 58
PHHAL_HW_CONFIG_FIELD_-
RECOVERY_TIME
dm_int, 58
PHHAL_HW_CONFIG_MAX_-
PRECACHED_BYTES
dm_int, 59
PHHAL_HW_CONFIG_MODINDEX
dm_int, 59
PHHAL_HW_CONFIG_PARITY
dm_int, 59
PHHAL_HW_CONFIG_RFRESET_-
ON_TIMEOUT
dm_int, 59
PHHAL_HW_CONFIG_RXALIGN
dm_int, 59
PHHAL_HW_CONFIG_RXBUFFER_-
BUFSIZE
dm_int, 59
PHHAL_HW_CONFIG_RXBUFFER_-
STARTPOS
dm_int, 59
PHHAL_HW_CONFIG_RXCRC
dm_int, 59
PHHAL_HW_CONFIG_RXDATARATE
dm_int, 60
PHHAL_HW_CONFIG_RXDEAFBITS
dm_int, 60
PHHAL_HW_CONFIG_RXLASTBITS
dm_int, 60
PHHAL_HW_CONFIG_SERIAL_-
BITRATE
dm_int, 60
PHHAL_HW_CONFIG_SUBCARRIER
dm_int, 60
PHHAL_HW_CONFIG_SYMBOL_-
END
dm_int, 60
PHHAL_HW_CONFIG_SYMBOL_-
START
dm_int, 60
PHHAL_HW_CONFIG_TIMEOUT_-
VALUE_MS
dm_int, 60
PHHAL_HW_CONFIG_TIMEOUT_-
VALUE_US
dm_int, 61
PHHAL_HW_CONFIG_TIMING_-
MODE
dm_int, 61
PHHAL_HW_CONFIG_TIMING_MS
dm_int, 61
PHHAL_HW_CONFIG_TIMING_US
dm_int, 61
PHHAL_HW_CONFIG_TXBUFFER
dm_int, 61
PHHAL_HW_CONFIG_TXBUFFER_-
BUFSIZE
dm_int, 61
PHHAL_HW_CONFIG_TXBUFFER_-
LENGTH
dm_int, 61
PHHAL_HW_CONFIG_TXCRC
dm_int, 61
PHHAL_HW_CONFIG_TXDATARATE
dm_int, 62
PHHAL_HW_CONFIG_TXLASTBITS
dm_int, 62
PHHAL_HW_CONFIG_TXWAIT_US
dm_int, 62
PHHAL_HW_MFC_KEYA
dm_int, 62
PHHAL_HW_MFC_KEYB
dm_int, 62
PHHAL_HW_MFC_USE_-
KEYMODIFIER
dm_int, 62
PHHAL_HW_RC523_DIGI_DELAY_-
US
p2_fw_timing_com.c, 173
PHHAL_HW_RC523_ETU_106
p2_fw_blf.c, 142
p2_fw_timing_com.c, 173
PHHAL_HW_RC523_TIMER_FREQ
p2_fw_blf.c, 142
p2_fw_timing_com.c, 173
PHHAL_HW_RC523_TIMER_SHIFT
p2_fw_blf.c, 142
p2_fw_timing_com.c, 174
PHHAL_HW_RF_DATARATE_106
dm_int, 62
PHHAL_HW_RF_DATARATE_212
dm_int, 62
PHHAL_HW_RF_DATARATE_424
dm_int, 63
PHHAL_HW_RF_DATARATE_848
dm_int, 63

PHHAL_HW_RS232_BITRATE_-
115200
dm_int, 63
PHHAL_HW_RS232_BITRATE_19200
dm_int, 63
PHHAL_HW_RS232_BITRATE_-
230400
dm_int, 63
PHHAL_HW_RS232_BITRATE_38400
dm_int, 63
PHHAL_HW_RS232_BITRATE_-
460800
dm_int, 63
PHHAL_HW_RS232_BITRATE_57600
dm_int, 63
PHHAL_HW_RS232_BITRATE_9600
dm_int, 64
PHHAL_HW_TIMING_MODE_COMM
dm_int, 64
PHHAL_HW_TIMING_MODE_FDT
dm_int, 64
PHHAL_HW_TIMING_MODE_OFF
dm_int, 64
PHPAL_I14443P4_CID_MAX
p2_fw_direct_mode_l4.c, 152
PHPAL_I14443P4_FRAMESIZE_MAX
p2_fw_direct_mode_l4.c, 152
PHPAL_I14443P4_FWI_MAX
p2_fw_direct_mode_l4.c, 152
PHPAL_I14443P4_PARAM_-
BLOCKNO
dm_int, 64
PHPAL_I14443P4_PARAM_CID
dm_int, 64
PHPAL_I14443P4_PARAM_FSI
dm_int, 64
PHPAL_I14443P4_PARAM_FWI
dm_int, 64
PHPAL_I14443P4_PARAM_-
MAXRETRYCOUNT
dm_int, 65
PHPAL_I14443P4_PARAM_NAD
dm_int, 65
pins
P2_FW_PINS_ANTENA_BLUE,
73
P2_FW_PINS_ANTENA_GREEN,
73
P2_FW_PINS_ANTENA_RED, 74
P2_FW_PINS_BEEPER, 74
P2_FW_PINS_CFG_1, 74
P2_FW_PINS_CFG_2, 74
P2_FW_PINS_CFG_3, 74
P2_FW_PINS_CFG_4, 74
P2_FW_PINS_DIP_1, 74
P2_FW_PINS_DIP_2, 74
P2_FW_PINS_DIP_3, 75
P2_FW_PINS_DIP_4, 75
P2_FW_PINS_DIP_5, 75
P2_FW_PINS_DIP_6, 75
P2_FW_PINS_DIP_7, 75
P2_FW_PINS_DIP_8, 75
P2_FW_PINS_LEDS_YELLOW_2,
75
P2_FW_PINS_LEDS_YELLOW_3,
75
P2_FW_PINS_LEDS_YELLOW_4,
76
pps_p
_p2_fw_data, 88
pro_p
_p2_fw_data, 88
proto
_p2_fw_slot, 95
prs_p
_p2_fw_data, 88

queue_pcsc_exec
_p2_fw_data, 88

rc_com_bal
_p2_fw_data, 88
rc_io
_p2_fw_data, 88
rc_op_ctl
_p2_fw_data, 88
rc_reg_ctl
_p2_fw_data, 88
reader
_p2_fw_data, 89
p2_fw_reader_read_config, 9
p2_fw_reader_set_up_external_-
interface, 9
p2_fw_reader_set_up_reader_chip,
10
p2_fw_reader_setup_hardware, 10
Reader Functions, 9
recExtraGuardTime
p2_fw_SAM_ctrl_, 99
regs_ops

_p2_fw_data, 89
regs_parms
_p2_fw_data, 89
rst_p
_p2_fw_data, 89
rx_xspeed
_p2_fw_data, 89

s_regs
_p2_fw_slot, 95
s_rp
_p2_fw_data, 89
sak
_p2_fw_slot, 95
sam_t1_param, 100
BWI_CWI, 100
ClockStop, 100
FI_DI, 100
GuardTime, 100
IFSC, 101
scb_p
_p2_fw_data, 89
sel_p
_p2_fw_data, 89
SELECT_CASCADE_LEVEL_1
dm_int, 65
SELECT_CASCADE_LEVEL_2
dm_int, 65
SELECT_CASCADE_LEVEL_3
dm_int, 65
send
_p2_fw_data, 89
send_data
p2_fw_SAM_ctrl_, 99
sendLen
p2_fw_SAM_ctrl_, 99
sendSeqData
p2_fw_SAM_ctrl_, 99
SERIAL_UART
p2_fw_ext_intf_serial.c, 156
SessionATR
p2_fw_SAM_ctrl_, 99
SessionATR_Size
p2_fw_SAM_ctrl_, 99
SINGLE_UID_LENGTH
dm_int, 65
slot
_p2_fw_pcsc_exec, 90
slot_data
_p2_fw_data, 89
slots
_p2_fw_data, 90
p2_fw_slots_add_new_l3_card, 14
p2_fw_slots_add_new_l4_card, 14
p2_fw_slots_add_new_sam_card, 15
p2_fw_slots_clear_cid_list, 15
p2_fw_slots_free_cid, 15
p2_fw_slots_get_atr, 15
p2_fw_slots_get_free_cid, 15
p2_fw_slots_get_free_slot, 16
p2_fw_slots_init, 16
p2_fw_slots_is_known_l3_card, 16
p2_fw_slots_remove_card, 16
Slots managment functions, 13
sm_rp
_p2_fw_data, 90
spr_p
_p2_fw_data, 90
src/bfl_support_files/phcsBflBal_Hw1SerCM3.c, 137
src/bfl_support_files/phcsBflRegCtl_SpiHw1.c, 139
src/p2_fw_blf.c, 141
src/p2_fw_ccid.c, 142
src/p2_fw_ccid_xfer.c, 144
src/p2_fw_demo_mode.c, 145
src/p2_fw_direct_mode.c, 145
src/p2_fw_direct_mode_cid.c, 146
src/p2_fw_direct_mode_contact_card.c, 147
src/p2_fw_direct_mode_hal.c, 148
src/p2_fw_direct_mode_key_store.c, 149
src/p2_fw_direct_mode_l3.c, 150
src/p2_fw_direct_mode_l4.c, 151
src/p2_fw_direct_mode_l4a.c, 152
src/p2_fw_direct_mode_ro.c, 153
src/p2_fw_direct_mode_xchg.c, 154
src/p2_fw_err_and_nfo_mode.c, 154
src/p2_fw_ext_intf_serial.c, 155
src/p2_fw_ext_intf_usb.c, 156
src/p2_fw_flash_utils.c, 158
src/p2_fw_main.c, 162
src/p2_fw_pcsc_ext.c, 163
src/p2_fw_pcsc_mode.c, 164
src/p2_fw_reader.c, 164
src/p2_fw_sam_t1.c, 165
src/p2_fw_slots.c, 172
src/p2_fw_timing_com.c, 172
src/p2_fw_utils.c, 174

state
cpot_atr_frame, 96

t0
p2_fw_SAM_ctrl_, 99
T1_BYTE_WAIT_TIME
p2_fw_sam_t1.c, 169
TAi
cpot_atr_frame, 96
Tasks, 10
tasks
p2_fw_task_demo_mode, 11
p2_fw_task_err_and_nfo_loop, 11
p2_fw_task_pcsc_execute, 11
p2_fw_task_pcsc_poll_and_act_-loop, 11
TBi
cpot_atr_frame, 96
TCi
cpot_atr_frame, 97
TCK
cpot_atr_frame, 97
TDi
cpot_atr_frame, 97
THE_UART
phcsBflBal_Hw1SerCM3.c, 138
TIMER0_IR
p2_fw_sam_t1.c, 169
TIMER0_IR_MR0
p2_fw_sam_t1.c, 169
TIMER0_IRQHandler
p2_fw_sam_t1.c, 171
p2_fw_sam_t1.h, 135
TIMER0_TCR
p2_fw_sam_t1.c, 169
TIMER0_TCR_ENABLE
p2_fw_sam_t1.c, 169
TIMER0_TCR_RESET
p2_fw_sam_t1.c, 169
TIMER2_IRQHandler
p2_fw_sam_t1.c, 171
p2_fw_sam_t1.h, 135
Timing, 76
timing
p2_fw_timing_init, 76
p2_fw_timing_start, 76
p2_fw_timing_stop_cless, 77
p2_fw_timing_stop_contac, 77
TIMING_MODE_OPTION_-AUTOCLEAR
p2_fw_timing_com.c, 174
TIMING_MODE_OPTION_MASK
p2_fw_timing_com.c, 174
timingMode
p2_fw_SAM_ctrl_, 99
tmpByteWait_time
p2_fw_SAM_ctrl_, 100
type
_p2_fw_slot, 95

UART3_IRQHandler
p2_fw.h, 119
p2_fw_ext_intf_serial.c, 156
UART_PORT
phcsBflBal_Hw1SerCM3.c, 138
uid
_p2_fw_slot, 95
Utils, 17
utils
p2_fw_utils_blink, 17
p2_fw_utils_field_off, 17
p2_fw_utils_field_on, 17
p2_fw_utils_get_dri, 18
p2_fw_utils_get_dsi, 18
p2_fw_utils_reg_read, 18
p2_fw_utils_reg_write, 18

wAdditionalInfo
_p2_fw_slot, 95
wFieldOffTime
_p2_fw_slot, 95
wFieldRecoveryTime
_p2_fw_slot, 95
wTimeoutUnit
_p2_fw_slot, 95
wTimingMode
_p2_fw_slot, 96