Fujian MoreFun Electronic Technology Co.,Ltd.

MFSDK API

V1.6

Fujian MoreFun Electronic Technology Co.,Ltd.

Contents

[Document history version](#_ne0blralbk47) [1](#_ne0blralbk47)

1. [Global](#_sgmgsd970abh) [2](#_sgmgsd970abh)
   1. [Module description](#_2gzi9qxam21x) [2](#_2gzi9qxam21x)
   2. [Module structure declaration](#_whr9bp3od0aj) [2](#_whr9bp3od0aj)
   3. [Constant declarations](#_dgeq0ywchmkw) [2](#_dgeq0ywchmkw)
2. [Version module](#_twq4s5k2oxlv) [2](#_twq4s5k2oxlv)
   1. [Module description](#_wodjg85y4lux) [2](#_wodjg85y4lux)
   2. [Module structure declaration](#_c47o3jmpxsb2) [3](#_c47o3jmpxsb2)
   3. [Constant declarations](#_jqeybnjigd0a) [3](#_jqeybnjigd0a)
   4. [MfSdkVerGetDataVersion](#_mp3n1ep504ew) [3](#_mp3n1ep504ew)
   5. [MfSdkVerMfOsVersion](#_roxlbqyob73k) [3](#_roxlbqyob73k)
   6. [MfSdkVerGetAppVer](#_9i3y7g6vpcxj) [4](#_9i3y7g6vpcxj)
   7. [MfSdkVerGetBootVer](#_fmel5ztsc63u) [4](#_fmel5ztsc63u)
   8. [MfSdkVerGetDriverVer](#_h1qo7t61o4f3) [4](#_h1qo7t61o4f3)
   9. [MfSdkVerGetSysVer](#_b6z8imq1n4f6) [5](#_b6z8imq1n4f6)
   10. [MfSdkVerGetSpVer](#_40lqfy3sb4ix) [5](#_40lqfy3sb4ix)
   11. [MfSdkVerSetAppVersion](#_1erk978l9jzn) [5](#_1erk978l9jzn)
   12. [MfSdkVerGetSecHwVer](#_82x3zikrqid) [6](#_82x3zikrqid)
   13. [MfSdkVerGetSecFwVer](#_3vfrdwg6vcj5) [6](#_3vfrdwg6vcj5)
3. [Util module](#_c6ctxapub1yj) [7](#_c6ctxapub1yj)
   1. [Module description](#_y1ow1lanboz5) [7](#_y1ow1lanboz5)
   2. [Module structure declaration](#_imx77o4o3qmx) [7](#_imx77o4o3qmx)
   3. [Constant declarations](#_chsh0jct71ol) [7](#_chsh0jct71ol)
   4. [MfSdkUtilGetModuleVer](#_fu7lpjob4440) [7](#_fu7lpjob4440)
   5. [MfSdkUtilAsc2Bcd](#_k3lx6wnkmbrq) [8](#_k3lx6wnkmbrq)
   6. [MfSdkUtilBcd2Asc](#_9timru9t1lk7) [8](#_9timru9t1lk7)
   7. [MfSdkUtilHex2Asc](#_43mppeecche4) [9](#_43mppeecche4)
   8. [MfSdkUtilInt2Bcd](#_4a8nqhvwp5s1) [10](#_4a8nqhvwp5s1)
   9. [MfSdkUtilBcd2Int](#_zh6ramnf6z73) [10](#_zh6ramnf6z73)
   10. [MfSdkUtilGenLrc](#_5skbcyak47p7) [11](#_5skbcyak47p7)
   11. [MfSdkUtilDes](#_p7bqkdwgif23) [11](#_p7bqkdwgif23)
   12. [MfSdkUtilBeep](#_5w7ryjsounae) [12](#_5w7ryjsounae)
   13. [MfSdkUtilBuzzerSound](#_hnuwbdr87y37) [12](#_hnuwbdr87y37)
   14. [MfSdkUtilGetRand](#_f9908p3hrlnv) [13](#_f9908p3hrlnv)
   15. [MfSdkUtilSHA1](#_lzu2l0htxz0d) [13](#_lzu2l0htxz0d)
   16. [MfSdkUtilGeneCodePic](#_kn3zb6sgam4m) [14](#_kn3zb6sgam4m)
   17. [MfSdkUtilLed](#_k8h2f4f9bml9) [14](#_k8h2f4f9bml9)
   18. [MfSdkUtilStr2Longlong](#_skb4dz54y5bu) [15](#_skb4dz54y5bu)
   19. [MfSdkUtilMd5File](#_imywm5yc2r7i) [15](#_imywm5yc2r7i)
   20. [MfSdkUtilUtf8str2Astr](#_d74wo3p58nek) [16](#_d74wo3p58nek)
   21. [MfSdkUtilUnicodeToUtf8](#_hh9fkqbyo0sq) [16](#_hh9fkqbyo0sq)
   22. [MfSdkUtilUtf8ToUnicode](#_l64z2tq5mjzo) [17](#_l64z2tq5mjzo)
   23. [MfSdkUtilTlvGetDataByTag](#_w3d085ckriuh) [17](#_w3d085ckriuh)
   24. [MfSdkUtilCtrlBeep](#_jy5hf6o3q84s) [18](#_jy5hf6o3q84s)
   25. [MfSdkUtilGenRsa](#_mo0s4qpwx062) [18](#_mo0s4qpwx062)
   26. [MfSdkUtilGenRsaKey](#_8tta65wrrnmj) [18](#_8tta65wrrnmj)
   27. [MfSdkUtilRsaSign](#_pdve2xlip78m) [19](#_pdve2xlip78m)
   28. [MfSdkUtilRsaGetPublicKey](#_wi5o7czb586f) [19](#_wi5o7czb586f)
   29. [MfSdkUtilGenRsaDelKey](#_mp1htkjcgo9b) [20](#_mp1htkjcgo9b)
   30. [MfSdkUtilGenRsaDelAllKey](#_u81uwz5yk5q2) [20](#_u81uwz5yk5q2)
   31. [MfSdkUtilRsaRand](#_nldv7qqroc9m) [21](#_nldv7qqroc9m)
   32. [MfSdkUtilCompress](#_umpkja9t6gtq) [21](#_umpkja9t6gtq)
   33. [MfSdkUtilCompressBound](#_dxe7u7k3908g) [22](#_dxe7u7k3908g)
   34. [MfSdkUtilSave2Zip](#_3safrf8xf7kq) [22](#_3safrf8xf7kq)
4. [RFID module](#_jo5535swl30e) [22](#_jo5535swl30e)
   1. [Module description](#_4rrq5astrhrg) [22](#_4rrq5astrhrg)
   2. [Module structure declaration](#_uib0h0kw37b1) [23](#_uib0h0kw37b1)
   3. [Constant declarations](#_95i27be2n7uu) [23](#_95i27be2n7uu)
   4. [MfSdkNfcInit](#_7gwrjy6wtjz2) [23](#_7gwrjy6wtjz2)
   5. [MfSdkNfcApdu](#_lfsq7g92q23s) [24](#_lfsq7g92q23s)
   6. [MfSdkNfcClose](#_qqfh9fc5pdib) [25](#_qqfh9fc5pdib)
   7. [MfSdkNfcCtlsComm](#_gmg9z5snm1a3) [25](#_gmg9z5snm1a3)
   8. [MfSdkNfcCtlsPowerDown](#_hp0t3n7wybpl) [25](#_hp0t3n7wybpl)
   9. [MfSdkNfcCtlsPowerUpAndSeek](#_tldtwkstejy7) [26](#_tldtwkstejy7)
   10. [MfSdkNfcDetect](#_i3irlxbo6n7) [26](#_i3irlxbo6n7)
   11. [MfSdkNfcApduStream](#_5r4v8amihx6q) [27](#_5r4v8amihx6q)
   12. [MfSdkNfcGetUid](#_wgd6lwibwxhc) [27](#_wgd6lwibwxhc)
   13. [MfSdkNfcIsProbe](#_tj536xp6y6o5) [28](#_tj536xp6y6o5)
   14. [MfSdkNfcM1Atqa](#_4vlonyk4qpmu) [28](#_4vlonyk4qpmu)
   15. [MfSdkNfcM1Auth](#_t5frvhxb5uu9) [28](#_t5frvhxb5uu9)
   16. [MfSdkNfcM1Close](#_99ws5nq2ezoc) [29](#_99ws5nq2ezoc)
   17. [MfSdkNfcM1Decrement](#_i9ok08czzjm0) [29](#_i9ok08czzjm0)
   18. [MfSdkNfcM1Increment](#_birabmf07qpk) [30](#_birabmf07qpk)
   19. [MfSdkNfcM1Open](#_q4z2anpaiho) [30](#_q4z2anpaiho)
   20. [MfSdkNfcM1Read](#_wc2g1dx1i25p) [30](#_wc2g1dx1i25p)
   21. [MfSdkNfcM1Restore](#_dbes4vjtg91w) [31](#_dbes4vjtg91w)
   22. [MfSdkNfcM1SetKey](#_grlgvz9ou3g6) [31](#_grlgvz9ou3g6)
   23. [MfSdkNfcM1Transfer](#_7sterd2nuldj) [32](#_7sterd2nuldj)
   24. [MfSdkNfcM1Uid](#_4wpbg5s6h8tq) [32](#_4wpbg5s6h8tq)
   25. [MfSdkNfcM1Write](#_uj6i0bwet2uo) [32](#_uj6i0bwet2uo)
   26. [MfSdkNfcOpen](#_hfcvropew41j) [33](#_hfcvropew41j)
   27. [MfSdkNfcReset](#_reynft9z6f6u) [33](#_reynft9z6f6u)
   28. [MfSdkNfcTagEmulateInitSetData](#_1rbfumc4rlh9) [34](#_1rbfumc4rlh9)
   29. [MfSdkNfcTagEmulateInitSetUrl](#_xt3jf8pw4b6h) [34](#_xt3jf8pw4b6h)
   30. [MfSdkNfcTagEmulateProcess](#_g5a4x4cxk7er) [35](#_g5a4x4cxk7er)
   31. [MfSdkNfcTagEmulateDeinit](#_p1whaoqcod0t) [35](#_p1whaoqcod0t)
   32. [MfSdkNfcLed](#_konm5e1zfwtl) [35](#_konm5e1zfwtl)
   33. [MfSdkNfcIsAroundFrontLcd](#_gez23mtxzsyw) [36](#_gez23mtxzsyw)
5. [Magstripe module](#_31cyh1cak2ch) [36](#_31cyh1cak2ch)
   1. [Module description](#_pstn4ukgq57l) [36](#_pstn4ukgq57l)
   2. [Module structure declaration](#_3cd7v3ens28a) [36](#_3cd7v3ens28a)
   3. [Constant declarations](#_iocz7wx70yzv) [36](#_iocz7wx70yzv)
   4. [MfSdkMagStripeDetect](#_2dg4r581x4a) [37](#_2dg4r581x4a)
   5. [MfSdkMagtekClose](#_yl54x9kozqfj) [37](#_yl54x9kozqfj)
   6. [MfSdkMagTekFlush](#_11ruk62nb41e) [37](#_11ruk62nb41e)
   7. [MfSdkMagtekOpen](#_socjsejyu79v) [38](#_socjsejyu79v)
6. [ICC module](#_oj844r87jpae) [38](#_oj844r87jpae)
   1. [Module description](#_i7p7452llvl) [38](#_i7p7452llvl)
   2. [Module structure declaration](#_638gk8l4c5ka) [38](#_638gk8l4c5ka)
   3. [Constant declarations](#_wea9umqvdqjq) [38](#_wea9umqvdqjq)
   4. [MfSdkIccClose](#_bzed6hqnt880) [39](#_bzed6hqnt880)
   5. [MfSdkIccComm](#_agppizt1a3lq) [40](#_agppizt1a3lq)
   6. [MfSdkIccGetCardATR](#_ykaf5sch77yt) [40](#_ykaf5sch77yt)
   7. [MfSdkIccGetCardStatus](#_hxd0tkn3fl74) [40](#_hxd0tkn3fl74)
   8. [MfSdkIccGetModuleVer](#_a4rlabvfab9o) [41](#_a4rlabvfab9o)
   9. [MfSdkIccInsertDetect](#_jluy8f2ksclm) [41](#_jluy8f2ksclm)
   10. [MfSdkIccApdu](#_gsm5s1n7ckgc) [42](#_gsm5s1n7ckgc)
   11. [MfSdkIccOpen](#_mjgfcqlnyh6p) [42](#_mjgfcqlnyh6p)
   12. [MfSdkIccPowerOff](#_clih4xv9cuj3) [42](#_clih4xv9cuj3)
   13. [MfSdkIccPowerUp](#_o5mz8o2nxauo) [43](#_o5mz8o2nxauo)
   14. [MfSdkIccPowerOn](#_aavsjbsbrfsf) [43](#_aavsjbsbrfsf)
7. [GUI module](#_rb6dhovabd8h) [44](#_rb6dhovabd8h)
   1. [Module description](#_vcdb8uhcydys) [44](#_vcdb8uhcydys)
   2. [Module structure declaration](#_jmrj2nedjhlk) [44](#_jmrj2nedjhlk)
   3. [Constant declarations](#_793635syrhbp) [44](#_793635syrhbp)
   4. [MfSdkGuiLedAmount](#_14tvd12saja3) [44](#_14tvd12saja3)
   5. [MfSdkGuiLedCounter](#_iss9bozihhnw) [45](#_iss9bozihhnw)
   6. [MfSdkGuiLedTime](#_atj7u2bs9wgh) [45](#_atj7u2bs9wgh)
   7. [MfSdkGuiLedDigitShow](#_sy7vsv9ny1fv) [45](#_sy7vsv9ny1fv)
   8. [MfSdkGuiBeginBatchPaint](#_5dx63c75ox5x) [46](#_5dx63c75ox5x)
   9. [MfSdkGuiEndBatchPaint](#_euyi56o4mwxn) [46](#_euyi56o4mwxn)
   10. [MfSdkGuiSetColor](#_k7qgmp1j6faq) [47](#_k7qgmp1j6faq)
   11. [MfSdkGuiGetColor](#_rwqv1q7uv4w) [47](#_rwqv1q7uv4w)
   12. [MfSdkGuiSetBgColor](#_lswx8bkv34f) [47](#_lswx8bkv34f)
   13. [MfSdkGuiGetBgColor](#_3snxps3v1lsf) [48](#_3snxps3v1lsf)
   14. [MfSdkGuiSetFullScreen](#_658kticoyzw0) [48](#_658kticoyzw0)
   15. [MfSdkGuiPixel](#_5xbuyjs535nu) [48](#_5xbuyjs535nu)
   16. [MfSdkGuiLineTo](#_ikarj1ft7ivz) [49](#_ikarj1ft7ivz)
   17. [MfSdkGuiBarRc](#_vkhrasg49r3r) [49](#_vkhrasg49r3r)
   18. [MfSdkGuiSetBarColor](#_rzjas02zb4pa) [50](#_rzjas02zb4pa)
   19. [MfSdkGuiGetBarColor](#_argl4qtbfa8e) [50](#_argl4qtbfa8e)
   20. [MfSdkGuiSetFont](#_nqd46jvo9im3) [51](#_nqd46jvo9im3)
   21. [MfSdkGuiGetFont](#_x1vlp96fglvw) [51](#_x1vlp96fglvw)
   22. [MfSdkGuiSetTextColor](#_1ey7bjsjhghy) [51](#_1ey7bjsjhghy)
   23. [MfSdkGuiGetTextColor](#_t7bi1qnijl08) [52](#_t7bi1qnijl08)
   24. [MfSdkGuiSetTextBgColor](#_apm3dmg7t10s) [52](#_apm3dmg7t10s)
   25. [MfSdkGuiGetTextBgColor](#_q77o2m1zmxo2) [52](#_q77o2m1zmxo2)
   26. [MfSdkGuiClearDc](#_kdstswuqvui0) [53](#_kdstswuqvui0)
   27. [MfSdkGuiSetTextZoom](#_7bgfd186cu2q) [53](#_7bgfd186cu2q)
   28. [MfSdkGuiGetTextZoom](#_457t7tcendzr) [54](#_457t7tcendzr)
   29. [MfSdkGuiSetPixel](#_g4rdc7si4n4s) [54](#_g4rdc7si4n4s)
   30. [MfSdkGuiGetPixel](#_kjchw9ajaksp) [54](#_kjchw9ajaksp)
   31. [MfSdkGuiTextOut](#_fpl0ry97yjxp) [55](#_fpl0ry97yjxp)
   32. [MfSdkGuiGetTextWidth](#_xcnt3rsozuwe) [55](#_xcnt3rsozuwe)
   33. [MfSdkGuiGetTextHeight](#_hjqtbiuw99kh) [56](#_hjqtbiuw99kh)
   34. [MfSdkGuiCline](#_9cotu4jv7633) [56](#_9cotu4jv7633)
   35. [MfSdkGuiGetWidth](#_98untf89cr81) [57](#_98untf89cr81)
   36. [MfSdkGuiGetHeight](#_m2znmjeeo8h7) [57](#_m2znmjeeo8h7)
   37. [MfSdkGuiPageOpPaint](#_5kxtv2so60a) [57](#_5kxtv2so60a)
   38. [MfSdkGuiImeSetMode](#_sdk9nqeeppjf) [58](#_sdk9nqeeppjf)
   39. [MfSdkGuiImeStartInput](#_fvoea1idezql) [58](#_fvoea1idezql)
   40. [MfSdkGuiMainMenuFuncAdd](#_n112hb6wikv7) [59](#_n112hb6wikv7)
   41. [MfSdkGuiMainMenuFuncDel](#_ngejy3hidmqu) [59](#_ngejy3hidmqu)
   42. [MfSdkGuiMainMenuShow](#_7eiroehw6h3h) [59](#_7eiroehw6h3h)
   43. [MfSdkGuiPostMessage](#_qiqeqv4cvtyu) [60](#_qiqeqv4cvtyu)
   44. [MfSdkGuiGetMessage](#_xhmxbv8t4vms) [60](#_xhmxbv8t4vms)
   45. [MfSdkGuiProcDefaultMsg](#_af9t7hbx65fq) [61](#_af9t7hbx65fq)
   46. [MfSdkGuiMessageBoxShow](#_pjl0tu7dd021) [61](#_pjl0tu7dd021)
   47. [MfSdkGuiLoadBmpEx](#_tv1b9uf3u2uy) [62](#_tv1b9uf3u2uy)
   48. [MfSdkGuiLoadBmp](#_dqpoe35uaz22) [62](#_dqpoe35uaz22)
   49. [MfSdkGuiBmpFree](#_u6otwqf5i0oc) [63](#_u6otwqf5i0oc)
   50. [MfSdkGuiOutBits](#_2j1hp2wlve7n) [63](#_2j1hp2wlve7n)
   51. [MfSdkGuiOutBitsEx](#_wdn6j6qurf22) [64](#_wdn6j6qurf22)
   52. [MfSdkGuiOutBitsZoom](#_jal31wd8e3ns) [65](#_jal31wd8e3ns)
   53. [MfSdkGuiTextWidthEx](#_6hglexks1s8m) [65](#_6hglexks1s8m)
   54. [MfSdkGuiTextOutEx](#_a02542j47p13) [66](#_a02542j47p13)
   55. [MfSdkGuiSetTextStyle](#_a7497shuk8t7) [66](#_a7497shuk8t7)
   56. [MfSdkGuiSelectPageEx](#_33sxkhekm8sx) [67](#_33sxkhekm8sx)
   57. [MfSdkGuiSelectPageExT](#_66z16uaqtub7) [67](#_66z16uaqtub7)
   58. [MfSdkGuiSelectPageCb](#_d1fzctloaxu6) [68](#_d1fzctloaxu6)
   59. [MfSdkGuiTitleColorBackground](#_npdigg7hlrre) [68](#_npdigg7hlrre)
   60. [MfSdkGuiTitleColorForeground](#_ftra2bg6jrbm) [69](#_ftra2bg6jrbm)
   61. [MfSdkGuiTitleFont](#_omxghsnpufmu) [69](#_omxghsnpufmu)
   62. [MfSdkGuiMenuHightlineColor](#_ikv4y3dvyhe9) [69](#_ikv4y3dvyhe9)
   63. [MfSdkGuiTextOutLineCenter](#_ffepz8d27mpw) [70](#_ffepz8d27mpw)
   64. [MfSdkGuiTextOutLineRight](#_4yln5n2viuxt) [70](#_4yln5n2viuxt)
   65. [MfSdkGuiTextOutLineLeft](#_vbhxc2cstmwf) [71](#_vbhxc2cstmwf)
   66. [MfSdkGuiTextOutWinCenter](#_97t0g8t9r5hx) [71](#_97t0g8t9r5hx)
   67. [MfSdkGuiClearRect](#_j9ajxmnuqef) [72](#_j9ajxmnuqef)
   68. [MfSdkGuiMessageBoxShowEx](#_c1zpv5jqj0cv) [72](#_c1zpv5jqj0cv)
   69. [MfSdkGuiSetTitle](#_9opc5mtwdyj8) [73](#_9opc5mtwdyj8)
   70. [MfSdkGuiTextOutHighlight](#_zb2sdsfzciyo) [73](#_zb2sdsfzciyo)
   71. [MfSdkGuiRectHighlight](#_hfi5lt4rgok6) [73](#_hfi5lt4rgok6)
   72. [MfSdkGuiDefaultMsgFuncAdd](#_4bp1vo2oeo8c) [74](#_4bp1vo2oeo8c)
   73. [MfSdkGuiMainMenuItemAdd](#_8oc9upsvps8n) [74](#_8oc9upsvps8n)
   74. [MfSdkGuiMainMenuItemDel](#_lexy5gf98gcb) [75](#_lexy5gf98gcb)
   75. [MfSdkGuiKeyGetEvent](#_v9qmxrihy7is) [75](#_v9qmxrihy7is)
   76. [MfSdkGuiWifiSetPage](#_x9w7dr7fq5q5) [75](#_x9w7dr7fq5q5)
   77. [MfSdkGuiSetProcs](#_aqcfgg7xszhb) [76](#_aqcfgg7xszhb)
   78. [MfSdkGuiSignExist](#_f69dlf5ww0hw) [76](#_f69dlf5ww0hw)
   79. [MfSdkGuiSignProc](#_1ra4h481qq4i) [77](#_1ra4h481qq4i)
   80. [MfSdkGuiSignPrint](#_rrcux1hacirs) [77](#_rrcux1hacirs)
   81. [MfSdkGuiSignEncode](#_op0yboh3semk) [77](#_op0yboh3semk)
   82. [MfSdkGuiSignEncodeFree](#_8g0cv8akm5l2) [78](#_8g0cv8akm5l2)
   83. [MfSdkGuiSignClean](#_aeito88qfl57) [78](#_aeito88qfl57)
   84. [MfSdkGuiSignSavetofile](#_kkzi9hmmioov) [79](#_kkzi9hmmioov)
   85. [MfSdkGuiSetPowerfullColor](#_uobesqeb26bz) [79](#_uobesqeb26bz)
   86. [MfSdkGuiStateFuncAdd](#_nb20hz814u3f) [79](#_nb20hz814u3f)
   87. [MfSdkGuiCleartState](#_5tlfdzorqdc1) [80](#_5tlfdzorqdc1)
   88. [MfSdkGuiOutStateBitsColor](#_9j82uzktai38) [80](#_9j82uzktai38)
   89. [MfSdkGuiGetDefineColor](#_7qnjb9z0pdd7) [81](#_7qnjb9z0pdd7)
   90. [MfSdkGuiSetCurlSocketIcon](#_iy6ujh3o6tm0) [81](#_iy6ujh3o6tm0)
   91. [MfSdkGuiRefreshMod](#_codmlbkg38ys) [81](#_codmlbkg38ys)
8. [Printer module](#_3cxyi5g61yld) [82](#_3cxyi5g61yld)
   1. [Module description](#_qo7s8lfzxn27) [82](#_qo7s8lfzxn27)
   2. [Module structure declaration](#_rvuor5ars7n7) [82](#_rvuor5ars7n7)
   3. [Constant declarations](#_nwe4enxmo3cd) [82](#_nwe4enxmo3cd)
   4. [MfSdkPrtInit](#_lt76ru2c655z) [83](#_lt76ru2c655z)
   5. [MfSdkPrtStr](#_yvoczb16b1ly) [83](#_yvoczb16b1ly)
   6. [MfSdkPrtBitMap](#_6ign8k9c9ksh) [84](#_6ign8k9c9ksh)
   7. [MfSdkPrtStart](#_k4igg82nea7d) [84](#_k4igg82nea7d)
   8. [MfSdkPrtStrBold](#_w0kv8tqys1m4) [85](#_w0kv8tqys1m4)
   9. [MfSdkPrtFeed](#_q0bavubrphqg) [85](#_q0bavubrphqg)
   10. [MfSdkPrtMatrixCode](#_lk3ksr3cw1pw) [86](#_lk3ksr3cw1pw)
   11. [MfSdkPrtSetFontEN](#_rk4mf17qzh0g) [86](#_rk4mf17qzh0g)
   12. [MfSdkPrtSetFontCN](#_gi6vvldeujfo) [87](#_gi6vvldeujfo)
   13. [MfSdkPrtSetDensity](#_89b4cs3o9wia) [87](#_89b4cs3o9wia)
   14. [MfSdkPrtSetAlign](#_8skfwcj8omhd) [88](#_8skfwcj8omhd)
   15. [MfSdkPrtSetLineAlign](#_ixo9uayiat79) [88](#_ixo9uayiat79)
   16. [MfSdkPrtStrLine](#_d2suy2s8x66c) [88](#_d2suy2s8x66c)
   17. [MfSdkPrtSetLineSpace](#_28u7red5infu) [89](#_28u7red5infu)
   18. [MfSdkPrtSetFontMode](#_9p19aq8ej6rc) [89](#_9p19aq8ej6rc)
   19. [MfSdkPrtCheckNopaper](#_uhducjwkhylm) [89](#_uhducjwkhylm)
   20. [MfSdkPrtSetArabicNumOrder](#_276762j1l6ll) [90](#_276762j1l6ll)
9. [Pinpad module](#_f54zoo5n24vo) [90](#_f54zoo5n24vo)
   1. [Module description](#_fwm0apkbez1u) [90](#_fwm0apkbez1u)
   2. [Module structure declaration](#_ffr6v52sqawv) [90](#_ffr6v52sqawv)
   3. [Constant declarations](#_eufkjl7kr11b) [91](#_eufkjl7kr11b)
   4. [MfSdkPedSetKeySize](#_8uspqabqba2l) [93](#_8uspqabqba2l)
   5. [MfSdkPedGetKeySize](#_7i1wsaj9pfry) [93](#_7i1wsaj9pfry)
   6. [MfSdkPedDukptLoadKey](#_k3w8f9rx8exm) [94](#_k3w8f9rx8exm)
   7. [MfSdkPedDukpt3desRun](#_gujfab7kzegd) [94](#_gujfab7kzegd)
   8. [MfSdkPedDukptGetKsn](#_m1t80py2hnhq) [95](#_m1t80py2hnhq)
   9. [MfSdkPedMkSkSavePlaintextKey](#_p92yhzmrhfld) [96](#_p92yhzmrhfld)
   10. [MfSdkPedMkSkSaveEncryptedKey](#_pgh1zqjb6fmn) [96](#_pgh1zqjb6fmn)
   11. [MfSdkPedMkSkGetKcv](#_dorxao8mb9s4) [97](#_dorxao8mb9s4)
   12. [MfSdkPedMkSk3desRun](#_23f4rqytgrsx) [98](#_23f4rqytgrsx)
   13. [MfSdkPedMacProc](#_ai5ilr77rnbu) [98](#_ai5ilr77rnbu)
   14. [MfSdkPedEncryptPinProc](#_4ovwq7kwhymd) [99](#_4ovwq7kwhymd)
   15. [MfSdkPedSetPinModeCfgInit](#_2tfzyzp71mvm) [100](#_2tfzyzp71mvm)
   16. [MfSdkPedSetPinModeCfg](#_3y131wvq53mg) [100](#_3y131wvq53mg)
   17. [MfSdkPedEncryptPinProcEx](#_y7eg6wp1ydvq) [101](#_y7eg6wp1ydvq)
   18. [MfSdkPedGetPinModeCfg](#_1ikzce74b8bi) [102](#_1ikzce74b8bi)
   19. [MfSdkPedGetPinModeStatus](#_4ddofskwtm2f) [102](#_4ddofskwtm2f)
   20. [MfSdkPedSaveRsaPriKey](#_rwtfe18jzy29) [103](#_rwtfe18jzy29)
   21. [MfSdkPedSaveRsaPukKey](#_9fl310cejg2m) [103](#_9fl310cejg2m)
   22. [MfSdkPedRsaBlock](#_xqhqzojk8ffi) [104](#_xqhqzojk8ffi)
   23. [MfSdkPedDeleteKey](#_afrm7uo7zjur) [104](#_afrm7uo7zjur)
   24. [MfSdkPedTr31GetData](#_ba2rdtfnl3g6) [105](#_ba2rdtfnl3g6)
   25. [MfSdkPedRkiDukptLoad](#_65jvi8a7b992) [105](#_65jvi8a7b992)
   26. [MfSdkPedRkiCheckAppKey](#_him32qrd5249) [105](#_him32qrd5249)
   27. [MfSdkPedSetPinpad](#_cprw99ewbr6y) [106](#_cprw99ewbr6y)
   28. [MfSdkPedGetPinpad](#_r25djp2zk4wv) [106](#_r25djp2zk4wv)
   29. [MfSdkPedSetPedUi](#_dz4vncck4oqn) [107](#_dz4vncck4oqn)
   30. [MfSdkPedCheckCardOnce](#_e7hcp22nukqt) [107](#_e7hcp22nukqt)
   31. [MfSdkPedIsWithPinpad](#_jrm2a9bqbdi) [108](#_jrm2a9bqbdi)
   32. [MfSdkPedExPinpadUpdate](#_wtn2srq87pjp) [108](#_wtn2srq87pjp)
   33. [MfSdkPedGetExPinpadVersion](#_n04p8ajbjisu) [109](#_n04p8ajbjisu)
   34. [MfSdkPedGetExPinpadInfo](#_m0v1gfnwl35) [109](#_m0v1gfnwl35)
   35. [MfSdkPedSetPedPinMod](#_al6cn88yu592) [110](#_al6cn88yu592)
   36. [MfSdkPedGetPin](#_4901gho14uap) [110](#_4901gho14uap)
   37. [MfSdkPedTR31LoadKBPK](#_edxm2rbx6npy) [111](#_edxm2rbx6npy)
   38. [MfSdkPedDukptLoadEncryptedKeyByTR31](#_wuedcfsebnh9) [111](#_wuedcfsebnh9)
   39. [MfSdkPedDukptLoadEncryptedKeyByTR31A](#_juc8i9wx6avn) [112](#_juc8i9wx6avn)
   40. [MfSdkPedMkskSaveEncryptedKeyByTR31](#_9fict25zr3ty) [112](#_9fict25zr3ty)
   41. [MfSdkPedSetTransKey](#_kdcyr2u9hkk5) [113](#_kdcyr2u9hkk5)
   42. [MfSdkPedGetTransKeyKvc](#_4obztqso1ruz) [114](#_4obztqso1ruz)
   43. [MfSdkPedSaveEncryptedMKByTransKey](#_hcm2ggk9vizn) [114](#_hcm2ggk9vizn)
   44. [MfSdkPedDeleteTransKey](#_slm3ym7sh9yf) [114](#_slm3ym7sh9yf)
   45. [MfSdkPedRebootCmd](#_n00ou3xa3crq) [115](#_n00ou3xa3crq)
   46. [MfSdkPedGotoMainPage](#_cjsc276uftf4) [115](#_cjsc276uftf4)
   47. [MfSdkPedAesSavePlaintextKey](#_ps44h54f94eb) [115](#_ps44h54f94eb)
   48. [MfSdkPedTr31LoadAesKey](#_y9oen4cntmja) [116](#_y9oen4cntmja)
   49. [MfSdkPedMkSkAesRun](#_svj6xcjtmqmu) [116](#_svj6xcjtmqmu)
10. [Communication module](#_uynsq72phtjp) [117](#_uynsq72phtjp)
    1. [Module description](#_wlwqfb9z3cvn) [117](#_wlwqfb9z3cvn)
    2. [Module structure declaration](#_tvf2zwk2f0c) [117](#_tvf2zwk2f0c)
    3. [Constant declarations](#_i2vdfntk5406) [117](#_i2vdfntk5406)
    4. [MfSdkCommLinkState](#_gy1zciwo4a29) [121](#_gy1zciwo4a29)
    5. [MfSdkCommGetNetMode](#_xpzqnkoro88k) [121](#_xpzqnkoro88k)
    6. [MfSdkCommAtcCell](#_t6r4ej9vkmtq) [121](#_t6r4ej9vkmtq)
    7. [MfSdkCommAtcCellInfor](#_xmyzmvyflvk6) [122](#_xmyzmvyflvk6)
    8. [MfSdkCommAtcCpin](#_z7otgq84cjsq) [122](#_z7otgq84cjsq)
    9. [MfSdkCommAtcGetLocalIp](#_dcv5cc748uws) [123](#_dcv5cc748uws)
    10. [MfSdkCommAtcIccid](#_trvsbnh2ajve) [123](#_trvsbnh2ajve)
    11. [MfSdkCommAtcImei](#_3391vzsimnmn) [124](#_3391vzsimnmn)
    12. [MfSdkCommAtcImsi](#_v2n66l5szac8) [124](#_v2n66l5szac8)
    13. [MfSdkCommAtcLac](#_k672xev9l33y) [124](#_k672xev9l33y)
    14. [MfSdkCommAtcGetMcc](#_xqgj9ohfyt7) [125](#_xqgj9ohfyt7)
    15. [MfSdkCommAtcGetMnc](#_k8esx0qr8izt) [125](#_k8esx0qr8izt)
    16. [MfSdkCommAtcLacInfor](#_eodhp2i0s9mv) [125](#_eodhp2i0s9mv)
    17. [MfSdkCommAtcSignal](#_22as8y2tjlc7) [126](#_22as8y2tjlc7)
    18. [MfSdkCommGetAtcGeneration](#_qzfmjt7o8eh) [126](#_qzfmjt7o8eh)
    19. [MfSdkCommGetAtcPower](#_3tkgb5ge08ou) [126](#_3tkgb5ge08ou)
    20. [MfSdkCommAtcSendCmd](#_z3a1cnk1b46q) [127](#_z3a1cnk1b46q)
    21. [MfSdkCommGetNetSelect](#_l955om5ev0n6) [127](#_l955om5ev0n6)
    22. [MfSdkCommGetOperateId](#_5tzi9v5fccr0) [128](#_5tzi9v5fccr0)
    23. [MfSdkCommGetWifiPower](#_tj0o6u7owj7m) [128](#_tj0o6u7owj7m)
    24. [MfSdkCommGsmGetSignal](#_wz6qvcuxv9ip) [128](#_wz6qvcuxv9ip)
    25. [MfSdkCommHttpDownload](#_f0u65ibixlm) [129](#_f0u65ibixlm)
    26. [MfSdkCommMbedtlsInit](#_x48yap50azka) [129](#_x48yap50azka)
    27. [MfSdkCommNetLink](#_3crsu4owchah) [130](#_3crsu4owchah)
    28. [MfSdkCommNetLinkWithUserInfo](#_ftr2x23v3nbp) [130](#_ftr2x23v3nbp)
    29. [MfSdkCommNetUnlink](#_7qrv0zuft8g) [131](#_7qrv0zuft8g)
    30. [MfSdkCommSetApMode](#_m03q4cl4zzj0) [131](#_m03q4cl4zzj0)
    31. [MfSdkCommSetApnList](#_8aptriejdwgp) [131](#_8aptriejdwgp)
    32. [MfSdkCommSetApnListMcc](#_593fopwgt4by) [132](#_593fopwgt4by)
    33. [MfSdkCommSetAtcPower](#_amgbtifgjmrq) [132](#_amgbtifgjmrq)
    34. [MfSdkCommSetInitApn](#_b10cvu1imq3u) [133](#_b10cvu1imq3u)
    35. [MfSdkCommSetNetMode](#_rz43oqx41reh) [133](#_rz43oqx41reh)
    36. [MfSdkCommSetNetSelect](#_mpwb5g609sy7) [134](#_mpwb5g609sy7)
    37. [MfSdkCommSetWifiName](#_hlf13kyqsuza) [134](#_hlf13kyqsuza)
    38. [MfSdkCommSocketClose](#_orocaf5pth61) [134](#_orocaf5pth61)
    39. [MfSdkCommSocketConnect](#_nvpkpz80t0qn) [135](#_nvpkpz80t0qn)
    40. [MfSdkCommSocketConnectPriority](#_owj68pcq0i06) [135](#_owj68pcq0i06)
    41. [MfSdkCommSocketCreate](#_8jvbtlcv5rjb) [136](#_8jvbtlcv5rjb)
    42. [MfSdkCommSocketFifoResize](#_z2n4sgtl7erp) [136](#_z2n4sgtl7erp)
    43. [MfSdkCommSocketRecv](#_sv79r9a5u5gx) [136](#_sv79r9a5u5gx)
    44. [MfSdkCommSocketSend](#_jjv7gwa16sv) [137](#_jjv7gwa16sv)
    45. [MfSdkCommSslAuthMode](#_d7ub091oi6jr) [137](#_d7ub091oi6jr)
    46. [MfSdkCommSslClose](#_54hyfsul4yil) [138](#_54hyfsul4yil)
    47. [MfSdkCommSslConnect](#_q0etw89grz8w) [139](#_q0etw89grz8w)
    48. [MfSdkCommSslInit](#_offnt4ml9blb) [139](#_offnt4ml9blb)
    49. [MfSdkCommSslSetHostname](#_4s9oz9q7f27a) [140](#_4s9oz9q7f27a)
    50. [MfSdkCommSslMbedtls](#_y30ez1x6brod) [140](#_y30ez1x6brod)
    51. [MfSdkCommSslRecv](#_lalg4fx05gsw) [141](#_lalg4fx05gsw)
    52. [MfSdkCommSslSend](#_t6hxu8bmmpu9) [141](#_t6hxu8bmmpu9)
    53. [MfSdkCommSslSend2](#_vre7nnecidlz) [142](#_vre7nnecidlz)
    54. [MfSdkCommUartClear](#_gazgzjkl4s56) [142](#_gazgzjkl4s56)
    55. [MfSdkCommUartClose](#_874y26xtedle) [143](#_874y26xtedle)
    56. [MfSdkCommUartGetRxBufLength](#_ddtgy6yhml3h) [143](#_ddtgy6yhml3h)
    57. [MfSdkCommUartOpen](#_20avt3z5l89h) [144](#_20avt3z5l89h)
    58. [MfSdkCommUartRecv](#_4whovda3f4ss) [144](#_4whovda3f4ss)
    59. [MfSdkCommUartSend](#_jkp5xrxvqgiu) [145](#_jkp5xrxvqgiu)
    60. [MfSdkCommUartSetupComm](#_30vcb9erbck3) [145](#_30vcb9erbck3)
    61. [MfSdkCommWifiClearListApNum](#_omrzzwycsdad) [146](#_omrzzwycsdad)
    62. [MfSdkCommWifiGetApMac](#_iupt1mb1iw51) [146](#_iupt1mb1iw51)
    63. [MfSdkCommWifiGetChannel](#_enfl7wvb3g2) [146](#_enfl7wvb3g2)
    64. [MfSdkCommWifiGetLinkState](#_5kzkfgqlchgc) [147](#_5kzkfgqlchgc)
    65. [MfSdkCommWifiGetLocalip](#_de6rjbak82e6) [147](#_de6rjbak82e6)
    66. [MfSdkCommWifiGetLocalMac](#_lpu4ztcca6ms) [147](#_lpu4ztcca6ms)
    67. [MfSdkCommWifiGetRssi](#_gfujp1tv2c30) [148](#_gfujp1tv2c30)
    68. [MfSdkCommWifiGetSignal](#_j2i5qlc5p3y) [148](#_j2i5qlc5p3y)
    69. [MfSdkCommWifiGetSsid](#_pjgqhfxoiayu) [149](#_pjgqhfxoiayu)
    70. [MfSdkCommWifiLinkAp](#_ip2uhn9djp3a) [149](#_ip2uhn9djp3a)
    71. [MfSdkCommWifiListAp](#_z3sn5e7ulr98) [149](#_z3sn5e7ulr98)
    72. [MfSdkCommWifiListApQuit](#_q7hws87154ej) [150](#_q7hws87154ej)
    73. [MfSDkCommWifiListNetWork](#_48c13shnsd01) [150](#_48c13shnsd01)
    74. [MfSdkCommWifiRemoveNetWorkAp](#_5nsso0pu8f08) [151](#_5nsso0pu8f08)
    75. [MfSdkCommWifiSetPower](#_6sljiinefssq) [151](#_6sljiinefssq)
    76. [MfSdkCommWifiSetScan](#_r1yy1ua0v0lc) [151](#_r1yy1ua0v0lc)
    77. [MfSdkCommWifiUnlinkAp](#_7rr5s8ny05nx) [152](#_7rr5s8ny05nx)
    78. [MfSdkCommSet4gMode](#_1078cqw0diw2) [152](#_1078cqw0diw2)
    79. [MfSdkCommWifiCheckState](#_fn3drq6mwyeo) [152](#_fn3drq6mwyeo)
    80. [MfSdkCommWifiStartConfig](#_l92z0vpw8iw6) [153](#_l92z0vpw8iw6)
    81. [MfSdkCommUartInit](#_k0drebxfkjgm) [153](#_k0drebxfkjgm)
    82. [MfSdkCommLanCableCheck](#_i64368cngcv) [154](#_i64368cngcv)
    83. [MfSdkCommLanSetDHCP](#_kcn5wzz1l8ku) [154](#_kcn5wzz1l8ku)
    84. [MfSdkCommLanIsDHCP](#_sptacxt61aa7) [154](#_sptacxt61aa7)
    85. [MfSdkCommLanCfg](#_ygwrr5zbdumn) [155](#_ygwrr5zbdumn)
    86. [MfSdkCommGetSwitchSimProcStatus](#_ezr9gc5lk02f) [155](#_ezr9gc5lk02f)
    87. [MfSdkCommSetOperateIdTask](#_jonv3pd9jzxc) [156](#_jonv3pd9jzxc)
    88. [MfSdkCommWifiPageInitMode](#_og4gi6qysph9) [156](#_og4gi6qysph9)
    89. [MfSdkCommWifiPageInit](#_3qvr0qlrlg6c) [156](#_3qvr0qlrlg6c)
    90. [MfSdkCommAtcPageInit](#_v1s6ctojelse) [157](#_v1s6ctojelse)
    91. [MfSdkCommAtcPageInitMode](#_l35ml9o6nrkc) [157](#_l35ml9o6nrkc)
    92. [MfSdkCommLanEnable](#_ib7uixowzztn) [158](#_ib7uixowzztn)
    93. [MfSdkCommLanDisenable](#_r4tlrey6kcbe) [158](#_r4tlrey6kcbe)
    94. [MfSdkCommAtcGetIp](#_izim79vs50n7) [158](#_izim79vs50n7)
    95. [MfSdkCommLanGetEthernetPower](#_7sb6zbsmr1b7) [159](#_7sb6zbsmr1b7)
    96. [MfSdkCommLanSetEthernetPower](#_dfqs5y3em43g) [159](#_dfqs5y3em43g)
    97. [MfSdkCommLanChipExist](#_cuq53bks43qd) [159](#_cuq53bks43qd)
    98. [MfSdkCommWifiChipExist](#_5wpz9vny1pgt) [160](#_5wpz9vny1pgt)
    99. [MfSdkCommAtcSetNetMode](#_so3nm1tdt41y) [160](#_so3nm1tdt41y)
    100. [MfSdkCommAtcSendAtCmd](#_31fh68fjy5qh) [161](#_31fh68fjy5qh)
    101. [MfSdkCommLanGetIpAndMask](#_4ojdvisooy82) [161](#_4ojdvisooy82)
    102. [MfSdkCommLanGetGateway](#_ihzaviq6pdg8) [161](#_ihzaviq6pdg8)
    103. [MfSdkCommLanGetDns](#_2ii856mrr00t) [162](#_2ii856mrr00t)
    104. [MfSdkCommLanGetMac](#_nffqxi4v9jy6) [162](#_nffqxi4v9jy6)
    105. [MfSdkCommConfigReset](#_ipjq9l6gprkm) [163](#_ipjq9l6gprkm)
    106. [MfSdkCommWifiStopConfig](#_ju9zvnsr0jey) [163](#_ju9zvnsr0jey)
    107. [MfSdkCommWifiRestart](#_5alym1fx4cz) [163](#_5alym1fx4cz)
    108. [MfSdkCommSetPingAddr](#_wctbgxryzk04) [164](#_wctbgxryzk04)
    109. [MFSdkCommSocketGetModelType](#_brffv8649rtt) [164](#_brffv8649rtt)
11. [EMV module](#_3bfu0pmz1ue) [164](#_3bfu0pmz1ue)
    1. [Module description](#_rv389wi74nar) [164](#_rv389wi74nar)
    2. [Module structure declaration](#_l4g1ffmgc552) [165](#_l4g1ffmgc552)
    3. [Constant declarations](#_yqrj2esme7se) [166](#_yqrj2esme7se)
    4. [MfSdkEmvKernelInit](#_poj9ijaz6gfl) [169](#_poj9ijaz6gfl)
    5. [MfSdkEmvTerminalConfigInit](#_4njtx6gwr8r5) [170](#_4njtx6gwr8r5)
    6. [MfSdkEmvCardLoop](#_n27rn7rtnj5s) [170](#_n27rn7rtnj5s)
    7. [MfSdkEmvCardProc](#_kffl14rdgkod) [171](#_kffl14rdgkod)
    8. [MfSdkEmvGetCard](#_g66669568x5p) [171](#_g66669568x5p)
    9. [MfSdkEmvGoOnChip](#_m12jeegntiuj) [172](#_m12jeegntiuj)
    10. [MfSdkEmvSetAid](#_m3vtb1tlhdv) [172](#_m3vtb1tlhdv)
    11. [MfSdkEmvDeleteOneAid](#_4sd5mu972e2u) [173](#_4sd5mu972e2u)
    12. [MfSdkEmvClearAllAid](#_cddv1nrbdo07) [173](#_cddv1nrbdo07)
    13. [MfSdkEmvGetAidNum](#_gbcp5zov7x0y) [174](#_gbcp5zov7x0y)
    14. [MfSdkEmvSetCapk](#_jgmri1jb6cxt) [174](#_jgmri1jb6cxt)
    15. [MfSdkEmvDeleteAllCapk](#_i9q5extv7u2s) [174](#_i9q5extv7u2s)
    16. [MfSdkEmvGetCapkNum](#_u27donqf8rhl) [175](#_u27donqf8rhl)
    17. [MfSdkEmvGetCapkByIndex](#_7nye5x6k3z70) [175](#_7nye5x6k3z70)
    18. [MfSdkEmvGetDataByTag](#_h4qo6tqvim6a) [176](#_h4qo6tqvim6a)
    19. [MfSdkEmvGetKernelData](#_jycgcq1vf285) [176](#_jycgcq1vf285)
    20. [MfSdkEmvGetDataByTag](#_eae9g2kv9d48) [177](#_eae9g2kv9d48)
    21. [MfSdkEmvPackTLVData](#_5lwytha8k9s) [177](#_5lwytha8k9s)
    22. [MfSdkEmvSetKernelData](#_paq3o5rnqt7) [178](#_paq3o5rnqt7)
    23. [MfSdkEmvSetDRL](#_ydwzcxnacy5y) [178](#_ydwzcxnacy5y)
    24. [MfSdkEmvClearDRLFile](#_ik6lw0u3d1ld) [179](#_ik6lw0u3d1ld)
    25. [MfSdkEmvMatchErrCode](#_metlv3u8b404) [179](#_metlv3u8b404)
    26. [MfSdkEmvOnlineRespPack](#_gktercfg03jn) [180](#_gktercfg03jn)
    27. [MfSdkEmvCardFree](#_bnfq4hvaa7z5) [180](#_bnfq4hvaa7z5)
    28. [MfSdkEmvAddCardBlackList](#_mwnlday029bz) [181](#_mwnlday029bz)
    29. [MfSdkEmvDelCardBlackList](#_6w1sdhj05dar) [181](#_6w1sdhj05dar)
    30. [MfSdkEmvGetVersion](#_gmer1hnyav2x) [181](#_gmer1hnyav2x)
    31. [MfSdkEmvGetEntryVersion](#_w6nb51cyimq6) [182](#_w6nb51cyimq6)
    32. [MfSdkEmvGetContactlessVersion](#_cckxev5gsusk) [182](#_cckxev5gsusk)
    33. [MfSdkEmvSetCallBackFunction](#_msh7d99qnt1q) [183](#_msh7d99qnt1q)
    34. [MfSdkEmvReadCardPage](#_t3cgoud0ofvs) [183](#_t3cgoud0ofvs)
    35. [MfSdkEmvSetSelectAppCallback](#_lqdek2w83cmf) [184](#_lqdek2w83cmf)
    36. [MfSdkEmvOfflinePinCallback](#_dvx0chmhhwe4) [184](#_dvx0chmhhwe4)
    37. [MfSdkEmvOnlinePinCallback](#_rzdc32vw1g6m) [184](#_rzdc32vw1g6m)
    38. [MfSdkEmvSetReadPageCallback](#_cuhkmrhzd3e4) [185](#_cuhkmrhzd3e4)
    39. [MfSdkEmvSetRuPay2ndTapCallback](#_o56swqy3e8ru) [185](#_o56swqy3e8ru)
    40. [MfSdkEmvSetDpas2ndTapCallback](#_37lppn5nqqwa) [186](#_37lppn5nqqwa)
    41. [MfSdkEmvGetPageWin](#_a2l8dodg8bnn) [186](#_a2l8dodg8bnn)
    42. [MfSdkEmvSetPreprocessOtherDataCallback](#_gcc10ekcd2d4) [186](#_gcc10ekcd2d4)
    43. [MfSdkEmvGetPageWinTip](#_jtkdo2dkifzn) [187](#_jtkdo2dkifzn)
    44. [MfSdkEmvGetProcInfo](#_ijy2wrfekrkh) [187](#_ijy2wrfekrkh)
    45. [MfSdkEmvCallbackEventSetAmtBeforeGpo](#_1n8i707ih4fk) [187](#_1n8i707ih4fk)
    46. [MfSdkEmvCallbackEventGetAmtBeforeGpo](#_jwxscnxzvbo1) [188](#_jwxscnxzvbo1)
    47. [MfSdkEmvCallbackEventClear](#_hvvhyfebez62) [188](#_hvvhyfebez62)
    48. [MfSdkEmvCallbackEventClear](#_272irrv5r7dr) [188](#_272irrv5r7dr)
    49. [MfSdkEmvGetEmvL2KernelName](#_my8gyp62b2eo) [189](#_my8gyp62b2eo)
    50. [MfSdkEmvGetAidsInit](#_btbtszl5shar) [189](#_btbtszl5shar)
    51. [MfSdkEmvGetAid](#_dmvjpmdzb7ns) [189](#_dmvjpmdzb7ns)
    52. [MfSdkEmvGetAidsFree](#_qoq24fp2bfnp) [190](#_qoq24fp2bfnp)
    53. [MfSdkEmvSetTlv2Kernel](#_5apjfpoguo7i) [190](#_5apjfpoguo7i)
12. [System module](#_vn8ux0qyigkn) [191](#_vn8ux0qyigkn)
    1. [Module description](#_6mqr8at004n5) [191](#_6mqr8at004n5)
    2. [Module structure declaration](#_49wdevvhxmk) [191](#_49wdevvhxmk)
    3. [Constant declarations](#_wkf2f1iblff9) [191](#_wkf2f1iblff9)
    4. [MfSdkSysGetDevModel](#_ldpml8j1m5yf) [194](#_ldpml8j1m5yf)
    5. [MfSdkSysGetDevModelName](#_q4s95u219eg7) [194](#_q4s95u219eg7)
    6. [MfSdkSysDevIs](#_4py8aalgfgpg) [195](#_4py8aalgfgpg)
    7. [MfSdkSysGetHardwareVer](#_upn0mckczifu) [195](#_upn0mckczifu)
    8. [MfSdkSysGetTime](#_yg185gp0xyve) [196](#_yg185gp0xyve)
    9. [MfSdkSysAppIsLock](#_42d8xlfw0dh2) [196](#_42d8xlfw0dh2)
    10. [MfSdkSysAuxlcdGetBrightness](#_7yl9exvlq37d) [197](#_7yl9exvlq37d)
    11. [MfSdkSysAuxlcdSetBrightness](#_nz0b372dtrot) [197](#_nz0b372dtrot)
    12. [MfSdkSysSubAuxlcdGetBrightness](#_3rqt1jcy5ky4) [197](#_3rqt1jcy5ky4)
    13. [MfSdkSysSubAuxlcdSetBrightness](#_ai3p9wt8z088) [198](#_ai3p9wt8z088)
    14. [MfSdkSysBatterCharge](#_tdvczts84avj) [198](#_tdvczts84avj)
    15. [MfSdkSysGetBatterStatus](#_gnl35dhh0was) [198](#_gnl35dhh0was)
    16. [MfSdkSysBuzzerSound](#_9z7eduinxqza) [199](#_9z7eduinxqza)
    17. [MfSdkSysCheckKey](#_h17wl4socwgu) [199](#_h17wl4socwgu)
    18. [MfSdkSysCheckTick](#_z4lpijgo3cky) [200](#_z4lpijgo3cky)
    19. [MfSdkSysClrKey](#_fg7cyo91x27x) [200](#_fg7cyo91x27x)
    20. [MfSdkSysConfig](#_1zh1zjmz8pmx) [200](#_1zh1zjmz8pmx)
    21. [MfSdkSysCurlInit](#_mn7jd11gdxsx) [201](#_mn7jd11gdxsx)
    22. [MfSdkSysDelay](#_411tgs8x3gq6) [201](#_411tgs8x3gq6)
    23. [MfSdkSysEnergySetTime](#_wglu1rvxyxkk) [202](#_wglu1rvxyxkk)
    24. [MfSdkSysEnergyTime](#_hqu3tux2kbc3) [202](#_hqu3tux2kbc3)
    25. [MfSdkSysEraseSecureArea](#_r4nv6h8ob29r) [202](#_r4nv6h8ob29r)
    26. [MfSdkSysFileSetPath](#_5jrszds2rz9t) [203](#_5jrszds2rz9t)
    27. [MfSdkSysGetBatterLevel](#_foixxngenxjg) [203](#_foixxngenxjg)
    28. [MfSdkSysGetDateTime](#_yzdvf2479o5l) [203](#_yzdvf2479o5l)
    29. [MfSdkSysGetIsLcd](#_but5yllffakf) [204](#_but5yllffakf)
    30. [MfSdkSysGetLcdType](#_n3t6rbmqkjka) [204](#_n3t6rbmqkjka)
    31. [MfSdkSysGetPsn](#_8sga2jkg8dc7) [205](#_8sga2jkg8dc7)
    32. [MfSdkSysGetTerminalInfo](#_2f98ql4cfvhe) [205](#_2f98ql4cfvhe)
    33. [MfSdkSysGetTermSn](#_go5gpfxybasg) [205](#_go5gpfxybasg)
    34. [MfSdkSysGetTerminalSn](#_2aqlppskpdkv) [206](#_2aqlppskpdkv)
    35. [MfSdkSysGetTick](#_pb01o0k1rfoo) [206](#_pb01o0k1rfoo)
    36. [MfSdkSysGetTickDiff](#_xr94518scfx2) [206](#_xr94518scfx2)
    37. [MfSdkSysGetTimeStamp](#_b2gcovmwv6lp) [207](#_b2gcovmwv6lp)
    38. [MfSdkSysGuiGroupGetObj](#_z86csnteydby) [207](#_z86csnteydby)
    39. [MfSdkSysHttpDownload](#_9zqt0ijkg5cz) [208](#_9zqt0ijkg5cz)
    40. [MfSdkSysHttpDownloadUseAgent](#_qw4301ifkby) [208](#_qw4301ifkby)
    41. [MfSdkSysInit](#_3k27s15g2eta) [208](#_3k27s15g2eta)
    42. [MfSdkSysLcdCalibration](#_nvhau3aqzvdx) [209](#_nvhau3aqzvdx)
    43. [MfSdkSysLogoInit](#_bufmnz285sch) [209](#_bufmnz285sch)
    44. [MfSdkSysLogoInitA](#_ywiqatuvivdl) [210](#_ywiqatuvivdl)
    45. [MfSdkSysModelType](#_gbolniahtbc8) [210](#_gbolniahtbc8)
    46. [MfSdkSysNetSetDatacallType](#_cb8z9oewj2eo) [211](#_cb8z9oewj2eo)
    47. [MfSdkSysPrintAdd](#_i507n5k2xuzv) [211](#_i507n5k2xuzv)
    48. [MfSdkSysReadFlashData](#_246f69t5puq5) [211](#_246f69t5puq5)
    49. [MfSdkSysReadSecureArea](#_dz1rqzf0qbcl) [212](#_dz1rqzf0qbcl)
    50. [MfSdkSysReboot](#_xhk44ob9hzgp) [212](#_xhk44ob9hzgp)
    51. [MfSdkSysRfidEmulateConfig](#_1zeg87bj3nn) [213](#_1zeg87bj3nn)
    52. [MfSdkSysRfidEmulateDeinit](#_rwr8f2ktzahf) [213](#_rwr8f2ktzahf)
    53. [MfSdkSysRfidEmulateInit](#_h71n6dd0es5m) [214](#_h71n6dd0es5m)
    54. [MfSdkSysRfidEmulateProcess](#_l2jvu1rwh0xv) [214](#_l2jvu1rwh0xv)
    55. [MfSdkSysRun](#_a1rz9fd8yx0l) [214](#_a1rz9fd8yx0l)
    56. [MfSdkSysSetDateTime](#_v6gezl6lnsk9) [215](#_v6gezl6lnsk9)
    57. [MfSdkSysSetLanguage](#_sghtp0zia9ag) [215](#_sghtp0zia9ag)
    58. [MfSdkSysSetLogData](#_h80626le1y2a) [216](#_h80626le1y2a)
    59. [MfSdkSysSetScrBackLight](#_t35i7vtmoh08) [216](#_t35i7vtmoh08)
    60. [MfSdkSysSleep](#_of3cgdwjl9vl) [216](#_of3cgdwjl9vl)
    61. [MfSdkSysStart](#_k6we5zbhf86) [217](#_k6we5zbhf86)
    62. [MfSdkSysTaskAppSet](#_j8k8pj7px8ld) [217](#_j8k8pj7px8ld)
    63. [MfSdkSysTaskCreate](#_7otj1yk6gwxb) [218](#_7otj1yk6gwxb)
    64. [MfSdkSysTimerCheck](#_q75w6nf8hgd4) [218](#_q75w6nf8hgd4)
    65. [MfSdkSysTimerClose](#_7vp4673lj0i5) [218](#_7vp4673lj0i5)
    66. [MfSdkSysTimerCreate](#_fkacrjlvs535) [219](#_fkacrjlvs535)
    67. [MfSdkSysTimerEnable](#_bzr4j57w9a13) [219](#_bzr4j57w9a13)
    68. [MfSdkSysTimerDelete](#_ttmkdrg0di4k) [220](#_ttmkdrg0di4k)
    69. [MfSdkSysTimerOpen](#_yaq0h7jawgwe) [220](#_yaq0h7jawgwe)
    70. [MfSdkSysTtsSystemSetFunc](#_fug1ana11tjs) [221](#_fug1ana11tjs)
    71. [MfSdkSysUnzipFileFunc](#_uwaehms5hbj9) [221](#_uwaehms5hbj9)
    72. [MfSdkSysVersion](#_af4exjly8j0q) [221](#_af4exjly8j0q)
    73. [MfSdkSysWriteFlashData](#_yudhmqb6idln) [222](#_yudhmqb6idln)
    74. [MfSdkSysWriteSecureArea](#_lqp3fdmd01r) [222](#_lqp3fdmd01r)
    75. [MfSdkSysZipUpdate](#_ua6fdpnu2i5l) [223](#_ua6fdpnu2i5l)
    76. [MfSdkSysDriverLibInit](#_nqgqslms25nb) [223](#_nqgqslms25nb)
    77. [MfSdkSysTaskAppInit](#_4p443h45brai) [223](#_4p443h45brai)
    78. [MfSdkSysPubDriverInit](#_ra3d9e5bq3nh) [224](#_ra3d9e5bq3nh)
    79. [MfSdkSysFontInit](#_ngzeudngvv8m) [224](#_ngzeudngvv8m)
    80. [MfSdkSysConsoleSwitch](#_npkcfrjor8sa) [225](#_npkcfrjor8sa)
    81. [MfSdkSysThreadMutexInit](#_29kmoc6dce8) [225](#_29kmoc6dce8)
    82. [MfSdkSysThreadMutexLock](#_v0l80u1hk0qg) [225](#_v0l80u1hk0qg)
    83. [MfSdkSysThreadMutexUnlock](#_xlv1ewubizov) [226](#_xlv1ewubizov)
    84. [MfSdkSysThreadMutexDestroy](#_kdipu6t296wu) [226](#_kdipu6t296wu)
    85. [MfSdkSysGetSegmentLcdDisplayMaxLength](#_21akqh2ixofv) [226](#_21akqh2ixofv)
    86. [MfSdkSysIsExternalFlash](#_6p8qurv1dv04) [227](#_6p8qurv1dv04)
    87. [MfSdkSysSetForceSleepTime](#_18kzcgwscz4i) [227](#_18kzcgwscz4i)
    88. [MfSdkSysSetSleepToPoweroff](#_rk8vd5dlwu97) [227](#_rk8vd5dlwu97)
    89. [MfSdkSysGetHeapInformation](#_l5cc0kdi5xze) [228](#_l5cc0kdi5xze)
    90. [MfSdkSysSetRestartEnabledOnce](#_rt5t2rd3a4kw) [228](#_rt5t2rd3a4kw)
13. [File System module](#_a0lwktai51vo) [229](#_a0lwktai51vo)
    1. [Module description](#_va9vf0bh7kyt) [229](#_va9vf0bh7kyt)
    2. [Module structure declaration](#_4pq8pcy6wvvb) [229](#_4pq8pcy6wvvb)
    3. [Constant declarations](#_uy4k6zkrjvux) [229](#_uy4k6zkrjvux)
    4. [MfSdkFsSetPath](#_veqc0twlrrq6) [230](#_veqc0twlrrq6)
    5. [MfSdkFsCheckPath](#_w2j6q6yv3o8a) [231](#_w2j6q6yv3o8a)
    6. [MfSdkFsClean](#_kd962oimxzko) [231](#_kd962oimxzko)
    7. [MfSdkFsClose](#_h8f0vch9p1i4) [231](#_h8f0vch9p1i4)
    8. [MfSdkFsDelete](#_n1c0p3as21a0) [232](#_n1c0p3as21a0)
    9. [MfSdkFsGetFreeSpace](#_1emomjdoij96) [232](#_1emomjdoij96)
    10. [MfSdkFsGetTotalSpace](#_xze0a13ac5w6) [233](#_xze0a13ac5w6)
    11. [MfSdkFsGetModuleVer](#_ymj87q6crv8k) [233](#_ymj87q6crv8k)
    12. [MfSdkFsLseek](#_7qfjc6mn5srg) [233](#_7qfjc6mn5srg)
    13. [MfSdkFsMkdir](#_iosiuzgllb8m) [234](#_iosiuzgllb8m)
    14. [MfSdkFsOpen](#_u54nro9uxglz) [234](#_u54nro9uxglz)
    15. [MfSdkFsPathClean](#_tz7zpn94hs1f) [236](#_tz7zpn94hs1f)
    16. [MfSdkFsRead](#_90jzs3dwd2pc) [237](#_90jzs3dwd2pc)
    17. [MfSdkFsReadLine](#_2nagfot74blx) [237](#_2nagfot74blx)
    18. [MfSdkFsReadProfileInt](#_xew6wkrojcxt) [237](#_xew6wkrojcxt)
    19. [MfSdkFsReadProfileString](#_erqok88fddds) [238](#_erqok88fddds)
    20. [MfSdkFsUnlink](#_2ryzt8za8kh1) [238](#_2ryzt8za8kh1)
    21. [MfSdkFsWrite](#_dxnatofnhp3y) [239](#_dxnatofnhp3y)
    22. [MfSdkFsWriteSync](#_co64a4xerrwv) [239](#_co64a4xerrwv)
    23. [MfSdkFsWriteBlockByName](#_tjbjml887pvh) [240](#_tjbjml887pvh)
    24. [MfSdkFsWriteProfileInt](#_9j6y0iswqhlz) [240](#_9j6y0iswqhlz)
    25. [MfSdkFsWriteProfileString](#_a53qkod0t7z2) [241](#_a53qkod0t7z2)
    26. [MfSdkFsRenamePath](#_5v73vs3ajf2t) [241](#_5v73vs3ajf2t)
    27. [MfSdkFsClearFile](#_vaqpuvon0wo4) [242](#_vaqpuvon0wo4)
    28. [MfSdkFsDelDirFiles](#_pb1gsddladjg) [242](#_pb1gsddladjg)
    29. [MfSdkFsRenameA](#_twn5wn8yho0y) [242](#_twn5wn8yho0y)
    30. [MfSdkFsGetFileLength](#_laqqewb6l4g5) [243](#_laqqewb6l4g5)
    31. [MfSdkFsRmdir](#_5vs59vldgkb7) [243](#_5vs59vldgkb7)
    32. [MfSdkFsOpenDir](#_m0w3h0l4p362) [244](#_m0w3h0l4p362)
    33. [MfSdkFsReadDir](#_711ztu6uh44r) [244](#_711ztu6uh44r)
    34. [MfSdkFsCloseDir](#_h9oj5rqrebvz) [245](#_h9oj5rqrebvz)
14. [FIFO module](#_38s68zrkt2ax) [246](#_38s68zrkt2ax)
    1. [Module description](#_z6dapf2trd0o) [246](#_z6dapf2trd0o)
    2. [Module structure declaration](#_dok1fk5iiblz) [246](#_dok1fk5iiblz)
    3. [Constant declarations](#_de8e0ze8pcqn) [246](#_de8e0ze8pcqn)
    4. [MfSdkFifoCreate](#_6zyqw1b6vlad) [247](#_6zyqw1b6vlad)
    5. [MfSdkFifoGet](#_1c7hvd6svjh6) [247](#_1c7hvd6svjh6)
    6. [MfSdkFifoInit](#_r2s8eaq287cl) [248](#_r2s8eaq287cl)
    7. [MfSdkFifoIsEmpty](#_vp41e36ypn4j) [248](#_vp41e36ypn4j)
    8. [MfSdkFifoIsFull](#_b3bi3z7b1no5) [249](#_b3bi3z7b1no5)
    9. [MfSdkFifoPut](#_2lkojn4jerdo) [249](#_2lkojn4jerdo)
    10. [MfSdkFifoResize](#_ioocw7z29wct) [250](#_ioocw7z29wct)
    11. [MfSdkGetFifoNum](#_ra3rrahkuwxg) [250](#_ra3rrahkuwxg)
    12. [MfSdkGetGetFifoSize](#_1pcix3giu4pv) [251](#_1pcix3giu4pv)
15. [Audio module](#_c8nuihlgd56t) [251](#_c8nuihlgd56t)
    1. [Module description](#_slyhkuhdtp4x) [251](#_slyhkuhdtp4x)
    2. [Module structure declaration](#_3y4jfkbay13w) [251](#_3y4jfkbay13w)
    3. [Constant declarations](#_485vyfku0aam) [252](#_485vyfku0aam)
    4. [MfSdkAudPlayVoice](#_bt6yc3cecwql) [253](#_bt6yc3cecwql)
    5. [MfSdkAudPlayAmt](#_ts604dvbn68n) [253](#_ts604dvbn68n)
    6. [MfSdkAudPlayBatteryLevel](#_cv170cc656ct) [254](#_cv170cc656ct)
    7. [MfSdkAudPlayFile](#_zhfufef1xvee) [254](#_zhfufef1xvee)
    8. [MfSdkAudPlay](#_4b1mw8x4gzw) [254](#_4b1mw8x4gzw)
    9. [MfSdkAudPlayNum](#_px6iosjd4obf) [255](#_px6iosjd4obf)
    10. [MfSdkAudPlayNumber](#_ghwurdi9fxs8) [255](#_ghwurdi9fxs8)
    11. [MfSdkAudPlayPayResult](#_eh1quxi7bnwt) [255](#_eh1quxi7bnwt)
    12. [MfSdkAudPlayPayType](#_igsm52oj2qt5) [256](#_igsm52oj2qt5)
    13. [MfSdkAudPlayUnit](#_acefzr3j4tgr) [256](#_acefzr3j4tgr)
    14. [MfSdkAudTtsState](#_qb8q03qinhat) [256](#_qb8q03qinhat)
    15. [MfSdkAudBatchBegin](#_4j7x2vbje1qa) [257](#_4j7x2vbje1qa)
    16. [MfSdkAudBathcEnd](#_wa6309bi8ake) [257](#_wa6309bi8ake)
    17. [MfSdkAudClear](#_gh7caf6smz77) [258](#_gh7caf6smz77)
    18. [MfSdkAudTtsPlay](#_yoezcgt4iitx) [258](#_yoezcgt4iitx)
    19. [MfSdkAudSetVolume](#_nd9yun4fthjr) [258](#_nd9yun4fthjr)
    20. [MfSdkAudSetVolumeRunning](#_6kuo9gp08skw) [259](#_6kuo9gp08skw)
    21. [MfSdkAudGetVolume](#_t81x3zfcktlm) [259](#_t81x3zfcktlm)
    22. [MfSdkAudSetSpeed](#_7fsykhopk1xi) [259](#_7fsykhopk1xi)
16. [KeyBoard module](#_6x3x52qcu8j4) [260](#_6x3x52qcu8j4)
    1. [Module description](#_fdzjhmq6zlvs) [260](#_fdzjhmq6zlvs)
    2. [Module structure declaration](#_njfumds29cub) [260](#_njfumds29cub)
    3. [Constant declarations](#_pvqnv8eum9jn) [260](#_pvqnv8eum9jn)
    4. [MfSdkKbKeySetParam](#_fvsans3j55wr) [261](#_fvsans3j55wr)
    5. [MfSdkKbGetKeySound](#_ca0viq3h1us8) [261](#_ca0viq3h1us8)
    6. [MfSdkKbSetKeySound](#_5xpbzagsh0jt) [262](#_5xpbzagsh0jt)
    7. [MfSdkKbWaitKey](#_rmt7jsaxl2di) [262](#_rmt7jsaxl2di)
17. [Lcd module](#_ru0l2n6nq43o) [262](#_ru0l2n6nq43o)
    1. [Module description](#_pj1kb58jzoro) [262](#_pj1kb58jzoro)
    2. [Module structure declaration](#_5lwca6m0khb2) [262](#_5lwca6m0khb2)
    3. [Constant declarations](#_gm2vw56zkj2) [263](#_gm2vw56zkj2)
    4. [MfSdkLcdBackLight](#_7iaoqjr3g028) [263](#_7iaoqjr3g028)
    5. [MfSdkLcdSegmentBackLight](#_eduu0erqfz0f) [264](#_eduu0erqfz0f)
    6. [MfSdkLcdGetSubProbe](#_wyzlawa1p694) [264](#_wyzlawa1p694)
    7. [MfSdkLcdSetIndex](#_dyu7va4bb4ef) [265](#_dyu7va4bb4ef)
    8. [MfSdkLcdGetPowerDownTime](#_oe53vmi9imn7) [265](#_oe53vmi9imn7)
    9. [MfSdkLcdSetPowerDownTime](#_7wmc6vz92p5c) [265](#_7wmc6vz92p5c)
    10. [MfSdkLcdGetBackLightTime](#_ncn83niydz7z) [266](#_ncn83niydz7z)
    11. [MfSdkLcdSetBackLightTime](#_gd2akjb965j0) [266](#_gd2akjb965j0)
    12. [MfSdkLcdGetFrontBackLightTime](#_96zmgmkkjw3m) [266](#_96zmgmkkjw3m)
    13. [MfSdkLcdSetFrontBackLightTime](#_bfwb076t90ys) [267](#_bfwb076t90ys)
    14. [MfSdkLcdGetRearBackLightTime](#_w6tfdop7buq6) [267](#_w6tfdop7buq6)
    15. [MfSdkLcdSetRearBackLightTime](#_ewecu45o2d0x) [267](#_ewecu45o2d0x)
    16. [MfSdkLcdBrightnessLevelSettings](#_jbf28za6k2s) [268](#_jbf28za6k2s)
    17. [MfSdkLcdAutoFlush](#_i4fzvgrkj068) [268](#_i4fzvgrkj068)
    18. [MfSdkLcdSetNormalDirection](#_oqsgf4jxrdso) [269](#_oqsgf4jxrdso)
    19. [MfSdkLcdArrowDisplay](#_ldunf2se736i) [269](#_ldunf2se736i)
    20. [MfSdkLcdBacklightIsBright](#_15tfr47qwpyn) [269](#_15tfr47qwpyn)
18. [Log module](#_12sdzpp4x8fp) [270](#_12sdzpp4x8fp)
    1. [Module description](#_nuop4n3o5uha) [270](#_nuop4n3o5uha)
    2. [Module structure declaration](#_d5sfgglnz17b) [270](#_d5sfgglnz17b)
    3. [Constant declarations](#_pwt6gnqddej4) [270](#_pwt6gnqddej4)
    4. [MfSdkLogSoundSet](#_anj61nw57wk2) [270](#_anj61nw57wk2)
    5. [MfSdkLog](#_tr89i2yc6qn0) [271](#_tr89i2yc6qn0)
    6. [MfSdkLogTip](#_rcvbh4wohtxg) [271](#_rcvbh4wohtxg)
    7. [MfSdkLogHexBuff](#_m54po711lgzl) [272](#_m54po711lgzl)
    8. [MfSdkLogLevel](#_3rjr78hh4kve) [272](#_3rjr78hh4kve)
    9. [MfSdkLogOutputSwitch](#_dvie0wx0zfwo) [273](#_dvie0wx0zfwo)
19. [Power module](#_vn1qdbn2x8ii) [273](#_vn1qdbn2x8ii)
    1. [Module description](#_hft5ubh5pib7) [273](#_hft5ubh5pib7)
    2. [Module structure declaration](#_qusu5bj8yqr4) [273](#_qusu5bj8yqr4)
    3. [Constant declarations](#_4a0n3io59h00) [273](#_4a0n3io59h00)
    4. [MfSdkPowerResumeProc](#_fempj2pa79ws) [274](#_fempj2pa79ws)
    5. [MfSdkPowerManagerSetFunc](#_iw1igk7grywm) [275](#_iw1igk7grywm)
    6. [MfSdkPowerLockApp](#_ofdu8sgdwc7z) [275](#_ofdu8sgdwc7z)
    7. [MfSdkPowerUnlockApp](#_w0bqhcadnmw5) [275](#_w0bqhcadnmw5)
    8. [MfSdkPowerTaskInit](#_z26xiq12rzj9) [276](#_z26xiq12rzj9)
    9. [MfSdkPowerTaskSuspend](#_bx8lf6p1oojv) [276](#_bx8lf6p1oojv)
    10. [MfSdkPowerReset](#_176wvx5sx94c) [276](#_176wvx5sx94c)
    11. [MfSdkPowerPageCb](#_r1u2kasdio8u) [277](#_r1u2kasdio8u)
    12. [MfSdkPowerPageInit](#_i1asv01cjoq6) [277](#_i1asv01cjoq6)
    13. [MfSdkPowerSetTime](#_o7fuw55ha0zx) [278](#_o7fuw55ha0zx)
    14. [MfSdkPowerOff](#_mx50xa10a1kp) [278](#_mx50xa10a1kp)
    15. [MfSdkPowerKeySetLight](#_1bu7b9pxa5tr) [278](#_1bu7b9pxa5tr)
    16. [MfSdkPowerGetBatteryPercentage](#_i3w3m7c5oidz) [279](#_i3w3m7c5oidz)
    17. [MfSdkPowerSetBacklightTime](#_nrgqs9mv2vdg) [279](#_nrgqs9mv2vdg)
    18. [MfSdkPowerSupertimeReset](#_2k44i9ip4v5u) [279](#_2k44i9ip4v5u)
    19. [MfSdkPowerSwitchResetTick](#_3ruvycmqhoow) [280](#_3ruvycmqhoow)
    20. [MfSdkPowerSleepSwitch](#_gd4iy6f4y0t4) [280](#_gd4iy6f4y0t4)
20. [QR module](#_o7mymv9bqxn) [281](#_o7mymv9bqxn)
    1. [Module description](#_zglyhy78m1v0) [281](#_zglyhy78m1v0)
    2. [Module structure declaration](#_79qsmfuhc0pz) [281](#_79qsmfuhc0pz)
    3. [Constant declarations](#_venpht6gdhv1) [281](#_venpht6gdhv1)
    4. [MfSdkQrDecode](#_o89d4r1dpkry) [281](#_o89d4r1dpkry)
    5. [MfSdkQrScannerClose](#_moaktnim0ie6) [282](#_moaktnim0ie6)
    6. [MfSdkQrScannerGetImg](#_tda5fk5xcqit) [282](#_tda5fk5xcqit)
    7. [MfSdkQrScannerOpen](#_ntaantvw8rj9) [282](#_ntaantvw8rj9)
    8. [MfSdkQrSetScanBoxPosition](#_c2fm0q5fw7uz) [283](#_c2fm0q5fw7uz)
    9. [MfSdkQrScannerStart](#_kt1n2k4yzzlp) [283](#_kt1n2k4yzzlp)
    10. [MfSdkQrScannerStop](#_7arwuso0v8gl) [283](#_7arwuso0v8gl)
    11. [MfSdkQrScannerSetPreview](#_3acftxch3ho2) [284](#_3acftxch3ho2)
21. [TMS module](#_hdylpr23z5h) [284](#_hdylpr23z5h)
    1. [Module description](#_m45i2pv4f284) [284](#_m45i2pv4f284)
    2. [Module structure declaration](#_dasy1yghnxe7) [284](#_dasy1yghnxe7)
    3. [Constant declarations](#_13leinkiusxf) [284](#_13leinkiusxf)
    4. [MfSdkTmsSetProgressCallback](#_f29j5p278no0) [286](#_f29j5p278no0)
    5. [MfSdkTmsSetResultCallback](#_yyas3rq24qsm) [287](#_yyas3rq24qsm)
    6. [MfSdkTmsHeartBeat](#_b5ml21rg66cr) [287](#_b5ml21rg66cr)
    7. [MfSdkTmsUpdate](#_w4ie5fo90d6r) [288](#_w4ie5fo90d6r)
    8. [MfSdkTmsUpdateFile](#_icltghhm9ion) [288](#_icltghhm9ion)
    9. [MfSdkTmsAppBusy](#_kme1c31j3dme) [288](#_kme1c31j3dme)
    10. [MfSdkTmsCheckTimeDisable](#_o2v8jrexa2uu) [289](#_o2v8jrexa2uu)
    11. [MfSdkTmsGetMsg](#_spqmalkexa36) [289](#_spqmalkexa36)
    12. [MfSdkTmsGetResult](#_o1bnzakyt4ly) [289](#_o1bnzakyt4ly)
    13. [MfSdkTmsSetSig](#_o1hujg4i8z63) [290](#_o1hujg4i8z63)
    14. [MfSdkTmsUpdateOta](#_ldmihbfd4ado) [290](#_ldmihbfd4ado)
    15. [MfSdkTmsSetConnectRetryCnt](#_crur0rxp5fnn) [291](#_crur0rxp5fnn)
    16. [MfSdkTmsSetErrorRetryCnt](#_4w5dvcl7adkv) [291](#_4w5dvcl7adkv)
    17. [MfSdkTmsEnable](#_irevursvnrsi) [291](#_irevursvnrsi)
    18. [MfSdkTmsSetActionCallback](#_3dj8lyczcy5h) [292](#_3dj8lyczcy5h)
    19. [MfSdkTmsEnableSyncTime](#_qkxfxnimz7q6) [292](#_qkxfxnimz7q6)
22. [LVGL module](#_2pfj07uhzlnz) [293](#_2pfj07uhzlnz)
    1. [Module description](#_gtgfrm6yyarc) [293](#_gtgfrm6yyarc)
    2. [Module structure declaration](#_i7l5ersqhpab) [293](#_i7l5ersqhpab)
    3. [Constant declarations](#_hoz1htq8p1y8) [293](#_hoz1htq8p1y8)
    4. [MfSdkLvglInit](#_k0jly6q7zrf3) [295](#_k0jly6q7zrf3)
    5. [MfSdkLvglGetPageBody](#_z0321lyq22ho) [295](#_z0321lyq22ho)
    6. [MfSdkLvglExit](#_lqc7ccqoamo1) [295](#_lqc7ccqoamo1)
    7. [MfSdkLvglCls](#_fbirl6psmzld) [296](#_fbirl6psmzld)
    8. [MfSdkLvglClsEnableStatusBar](#_cf7msbd6fht) [296](#_cf7msbd6fht)
    9. [MfSdkLvglEnableDefaultStatusBarIcons](#_6bre4gn75jni) [297](#_6bre4gn75jni)
    10. [MfSdkLvglClearLine](#_offu68mmndd6) [297](#_offu68mmndd6)
    11. [MfSdkLvglGetStatusBarHeight](#_c66vuvnoqico) [297](#_c66vuvnoqico)
    12. [MfSdkLvglCreateButtons](#_yrkgvg2f8yjc) [298](#_yrkgvg2f8yjc)
    13. [MfSdkLvglClearAll](#_9lvid3r7v95b) [298](#_9lvid3r7v95b)
    14. [MfSdkLvglWaitKey](#_affsvy6msuak) [298](#_affsvy6msuak)
    15. [MfSdkLvglWaitKeyMs](#_v8q9x0uw35l6) [299](#_v8q9x0uw35l6)
    16. [MfSdkLvglClrKeyFlag](#_q5wp3sb91j8a) [299](#_q5wp3sb91j8a)
    17. [MfSdkLvglCheckKey](#_i282jl755qx1) [300](#_i282jl755qx1)
    18. [MfSdkLvglGetKey](#_1gothoxz2fdx) [300](#_1gothoxz2fdx)
    19. [MfSdkLvglDispTextCoord](#_4lp2l3grf3bm) [300](#_4lp2l3grf3bm)
    20. [MfSdkLvglDispTextLine](#_x1mifbh2mdam) [301](#_x1mifbh2mdam)
    21. [MfSdkLvglDispMenuText](#_1j40n6jdu9vc) [301](#_1j40n6jdu9vc)
    22. [MfSdkLvglDispMenuIcon](#_yyqxdk3s4f70) [302](#_yyqxdk3s4f70)
    23. [MfSdkLvglDispButton](#_epetr06atvf5) [302](#_epetr06atvf5)
    24. [MfSdkLvglMessageBox](#_q182jtifq4) [303](#_q182jtifq4)
    25. [MfSdkLvglDispList](#_9f8hgzuq6u3t) [303](#_9f8hgzuq6u3t)
    26. [MfSdkLvglDispRollpage](#_30ffz3x981rl) [304](#_30ffz3x981rl)
    27. [MfSdkLvglInputText](#_s2jiirh7fkbl) [304](#_s2jiirh7fkbl)
    28. [MfSdkLvglInputTextEx](#_y92pwd2d03mx) [305](#_y92pwd2d03mx)
    29. [MfSdkLvglDispBar](#_3iqqojiivffp) [305](#_3iqqojiivffp)
    30. [MfSdkLvglUpdateBar](#_95t26ymbbd3v) [306](#_95t26ymbbd3v)
    31. [MfSdkLvglDispLed](#_qzi2vebjd5fw) [306](#_qzi2vebjd5fw)
    32. [MfSdkLvglUpdateLed](#_okx0905a4nwt) [307](#_okx0905a4nwt)
    33. [MfSdkLvglShowQrcode](#_qnmmvxxriuor) [307](#_qnmmvxxriuor)
    34. [MfSdkLvglShowQrcodeEx](#_hvz4y7lqjak4) [308](#_hvz4y7lqjak4)
    35. [MfSdkLvglShowImage](#_o9m4g9yuu312) [308](#_o9m4g9yuu312)
    36. [MfSdkLvglShowImageEx](#_i9ja7jxab5t3) [309](#_i9ja7jxab5t3)
    37. [MfSdkLvglShowImagebuff](#_epmzunx5x5dd) [309](#_epmzunx5x5dd)
    38. [MfSdkLvglShowScreenCanves](#_lwr47e60eegy) [310](#_lwr47e60eegy)
    39. [MfSdkLvglDrawLine](#_ebus5rmdd4yu) [310](#_ebus5rmdd4yu)
    40. [MfSdkLvglDrawBox](#_nib5154b05f) [311](#_nib5154b05f)
    41. [MfSdkLvglEsign](#_fimkvrdcgc6k) [311](#_fimkvrdcgc6k)
    42. [MfSdkLvglScan](#_yuhjp7bh2jmt) [312](#_yuhjp7bh2jmt)
    43. [MfSdkLvglDispDialog](#_keowpixjvpsu) [312](#_keowpixjvpsu)
    44. [MfSdkLvglUpdateDialog](#_1d1i9u1eb0q4) [313](#_1d1i9u1eb0q4)
    45. [MfSdkLvglCloseDialog](#_fjuybvah6rgm) [313](#_fjuybvah6rgm)
    46. [MfSdkLvglDispRollpageEx](#_xiwspqby473e) [313](#_xiwspqby473e)
    47. [MfSdkLvglDispMultipleChoice](#_72edwzrtuqae) [314](#_72edwzrtuqae)
    48. [MfSdkLvglDispSetSlider](#_t8bo0l7q1nhq) [314](#_t8bo0l7q1nhq)
    49. [MfSdkLvglDispSetResult](#_vhjt8n6tm0ha) [315](#_vhjt8n6tm0ha)
    50. [MfSdkLvglInputPin](#_9qjlfnuml2nd) [315](#_9qjlfnuml2nd)
    51. [MfSdkLvglDispAnimation](#_uwov5oknjyfs) [316](#_uwov5oknjyfs)
    52. [MfSdkLvglCloseAnimation](#_eh6mqceokncc) [316](#_eh6mqceokncc)
    53. [MfSdkLvglCreateStatusBarTask](#_mmoe822a6vix) [317](#_mmoe822a6vix)
    54. [MfSdkLvglSetStatusBarIcon](#_75pa73ne0j10) [317](#_75pa73ne0j10)
    55. [MfSdkLvglQuickSetStatusBarIcon](#_7yxfe72ah46s) [318](#_7yxfe72ah46s)
    56. [MfSdkLvglSetStatusBar](#_5tfqmxgkgsru) [318](#_5tfqmxgkgsru)
    57. [MfSdkLvglGetUiBuff](#_wohbcpuijwhw) [318](#_wohbcpuijwhw)
    58. [MfSdkLvglLoadPng](#_j45p6thohjo1) [319](#_j45p6thohjo1)
    59. [MfSdkLvglFreePng](#_3g3wfjh4sbs) [319](#_3g3wfjh4sbs)
    60. [MfSdkLvglGuiInit](#_jarlls8p95dm) [320](#_jarlls8p95dm)
    61. [MfSdkLvglGroupSet](#_ac0d895n9xbz) [320](#_ac0d895n9xbz)
23. [Memory module](#_dkmwiryv374b) [320](#_dkmwiryv374b)
    1. [Module description](#_vjr0pit5hbps) [320](#_vjr0pit5hbps)



# Document history version

| **Date** | **Version** | **Remark** | **Author** |
| --- | --- | --- | --- |
| **2025-04-02** | **V1.6** | 1. **update to MFSDK V1.7.4** | **MoreFun SDK group** |
| **2024-05-31** | **V1.5** | 1. **Added MfSdkFsRmdir** 2. **Added MfSdkFsOpenDir** 3. **Added MfSdkFsReadDir** 4. **Added MfSdkFsCloseDir** | **MoreFun SDK group** |
| **2024-05-31** | **V1.4** | 1. **Add APIs description** | **MoreFun SDK group** |
| **2024-02-20** | **V1.3** | 1. **Add Pinpad-related apis** | **MoreFun SDK group** |
| **2024-02-20** | **V1.2** | 1. **Added API MfSdkFsRenamePath and MfSdkFsGetFileLength** 2. **API MfSdkFsOpenadd descriptive information;** 3. **MfSdkLcdSetPowerDownTime input parameter unit second;** 4. **MfSdkCommSocketConnect input parameter timeout unit ms;** | **MoreFun SDK group** |
| **2024-02-17** | **V1.1** | 1. **MfSdkCommAtcCpin add descriptive information;** 2. **MfSdkQrDecode modify the return value type;** 3. **Added MfSdkTmsSetSig API whether to enable the signature verification function;** 4. **Added MfSdkVerGetSecHwVer/MfSdkVerGetSecFwVer;** 5. **MfSdkPrtSetLineSpace;** | **MoreFun SDK group** |
| **2023-12-31** | **V1.0** | **First version** | **MoreFun SDK**  **group** |

# Global

## Module description

The definition of the global description.

## Module structure declaration

None. 

## Constant declarations

#define MFSDK\_UNUSED(x) ((void)(x))

#define MFSDK\_COND\_RET( cond , retValue ) do{ if( cond ) \

{ return retValue ; } }while(0)

#define MFSDK\_FALSE (0)

#define MFSDK\_TRUE (1)

# Version module

## Module description

This module mainly includes APIs for getting versions.



## Module structure declaration

None.

## Constant declarations

None.

## MfSdkVerGetDataVersion

| **Prototype** | | **LIB\_EXPORT s8\* MfSdkVerGetDataVersion()** |
| --- | --- | --- |
| **Function** | | **Get data lib version** |
| **Params** | **in** | **Nothing** |
| **out** | **Nothing** |
| **return** | | **version string** |
| **remark** | |  |
| **demo** | | char\* data\_version = NULL;  data\_version = MfSdkVerGetDataVersion(); |

## MfSdkVerMfOsVersion

| **Prototype** | | **LIB\_EXPORT s8\* MfSdkVerMfOsVersion()** |
| --- | --- | --- |
| **Function** | | **Get MoreFun OS SDK version** |
| **Params** | **in** | **Nothing** |
| **out** | **Nothing** |
| **return** | | **version string** |
| **remark** | |  |
| **demo** | | char\* osVersion = NULL;  osVersion = MfSdkVerMfOsVersion(); |



## MfSdkVerGetAppVer

| **Prototype** | | **LIB\_EXPORT const s8 \*MfSdkVerGetAppVer()** |
| --- | --- | --- |
| **Function** | | **Get App version** |
| **Params** | **in** | **Nothing** |
| **out** | **Nothing** |
| **return** | | **version string** |
| **remark** | |  |
| **demo** | | char\* appVersion = NULL;  appVersion = MfSdkVerGetAppVer(); |

## MfSdkVerGetBootVer

| **Prototype** | | **LIB\_EXPORT const s8 \*MfSdkVerGetBootVer()** |
| --- | --- | --- |
| **Function** | | **Get boot version** |
| **Params** | **in** | **Nothing** |
| **out** | **Nothing** |
| **return** | | **version string** |
| **remark** | |  |
| **demo** | | char\* bootVersion = NULL;  bootVersion = MfSdkVerGetBootVer(); |

## MfSdkVerGetDriverVer

| **Prototype** | | **LIB\_EXPORT const s8 \*MfSdkVerGetDriverVer()** |
| --- | --- | --- |
| **Function** | | **Get driver version** |
| **Params** | **in** | **Nothing** |
| **out** | **Nothing** |
| **return** | | **version string** |



| **remark** |  |
| --- | --- |
| **demo** | char\* driverVersion = NULL;  driverVersion = MfSdkVerGetDriverVer(); |

## MfSdkVerGetSysVer

| **Prototype** | | **LIB\_EXPORT const s8 \*MfSdkVerGetSysVer()** |
| --- | --- | --- |
| **Function** | | **Get system version** |
| **Params** | **in** | **Nothing** |
| **out** | **Nothing** |
| **return** | | **version string** |
| **remark** | |  |
| **demo** | | char\* sysVersion = NULL;  sysVersion = MfSdkVerGetSysVer(); |

## MfSdkVerGetSpVer

| **Prototype** | | **LIB\_EXPORT const s8 \*MfSdkVerGetSpVer()** |
| --- | --- | --- |
| **Function** | | **Get sp version** |
| **Params** | **in** | **Nothing** |
| **out** | **Nothing** |
| **return** | | **version string** |
| **remark** | |  |
| **demo** | | char\* spVersion = NULL;  spVersion = MfSdkVerGetSpVer(); |

## MfSdkVerSetAppVersion

**Prototype LIB\_EXPORT s32 MfSdkVerSetAppVersion(s8 \*pszVer)**

****

| **Function** | | **Set application version.** |
| --- | --- | --- |
| **Params** | **in** | **pszVer** app version string |
| **out** | **Nothing** |
| **return** | | **0 Success** |
| **remark** | |  |
| **demo** | | char\* appVer = “V1.0.0”;  MfSdkVerSetAppVersion(appVer); |

## MfSdkVerGetSecHwVer

| **Prototype** | | **LIB\_EXPORT s8 \*MfSdkVerGetSecHwVer()** |
| --- | --- | --- |
| **Function** | | **get pci hardware version** |
| **Params** | **in** | **Nothing** |
| **out** | **Nothing** |
| **return** | | **version string** |
| **remark** | |  |
| **demo** | | char \*ver = (char\*)MfSdkVerGetSecHwVer(); |

## MfSdkVerGetSecFwVer

| **Prototype** | | **LIB\_EXPORT s8 \*MfSdkVerGetSecFwVer()** |
| --- | --- | --- |
| **Function** | | **Get pci firmware version** |
| **Params** | **in** | **Nothing** |
| **out** | **Nothing** |
| **return** | | **version string** |
| **remark** | |  |
| **demo** | | char \*ver = (char\*)MfSdkVerGetSecFwVer(); |

# Util module

## Module description

This module mainly includes encoding type conversion and other util APIs.

## Module structure declaration

//QR code parameters typedef struct

{

s32 nVersion; //< version number:1~40

s32 nLevel; //< Error correction level: 0-low, 1-in, 2-high, 3-maximum s32 moudleWidth; ///< Module width (unit: pixel)

} MfSdkUtilQrInfo\_T;

## Constant declarations



## MfSdkUtilGetModuleVer

| **Prototype** | | **LIB\_EXPORT s32 MfSdkUtilGetModuleVer(s8 \*pszVer)** |
| --- | --- | --- |
| **Function** | | **Get util module version** |
| **Params** | **in** | **Nothing** |
| **out** | **pszVer** Module version |
| **return** | | **Others Failed, Ref. MfSdkUtilRet\_E** |



|  | **MFSDK\_UTIL\_RET\_**  **Success**  **OK** |
| --- | --- |
| **remark** |  |
| **demo** | char utilVersion[20] = {0}; int ret = -1;  ret = MfSdkUtilGetModuleVer(utilVersion); |

## MfSdkUtilAsc2Bcd

| **Prototype** | | **LIB\_EXPORT s32 MfSdkUtilAsc2Bcd(s8 \*AscBuf, s8 \*BcdBuf, s32 AscLen)** |
| --- | --- | --- |
| **Function** | | **ASCII code to BCD code** |
| **Params** | **in** | **AscLen** Incoming ASCII code data length |
| **AscBuf** ASCII code data to be converted |
| **out** | **BcdBuf** Conversion output BCD code data |
| **return** | | **Others Failed, Ref. MfSdkUtilRet\_E** |
| **MFSDK\_UTIL\_RET\_**  **Success**  **OK** |
| **remark** | |  |
| **demo** | | char \*AscBuf = "1234"; char BcdBuf[3] = {0}; int ret = -1;  ret = MfSdkUtilAsc2Bcd(AscBuf, BcdBuf, strlen(AscBuf)); |

## MfSdkUtilBcd2Asc

| **Prototype** | | **LIB\_EXPORT s32 MfSdkUtilBcd2Asc(s8 \*BcdBuf, s8 \*AscBuf, s32 AscLen)** |
| --- | --- | --- |
| **Function** | | **BCD code to ASCII code** |
| **Params** | **in** | The length of ASCII code data, that is the double  **AscLen**  length of BCD code data |
| **BcdBuf** BCD code data to be converted |
| **out** | **AscBuf** Conversion output ASCII code data |
| **return** | | **Others Failed, Ref. MfSdkUtilRet\_E** |
| **MFSDK\_UTIL\_RET\_ Success** |



|  | **OK** |
| --- | --- |
| **remark** |  |
| **demo** | char AscBuf[9] = {0};  char \*BcdBuf = "\x12\x23"; int ret = -1;  ret = MfSdkUtilBcd2Asc(BcdBuf, AscBuf, 2); APP\_TRACE("MfSdkUtilBcd2Asc:%s\r\n",AscBuf);  Result: MfSdkUtilBcd2Asc:12 |

## MfSdkUtilHex2Asc

| **Prototype** | | **LIB\_EXPORT s32 MfSdkUtilHex2Asc(u8\* pszBcdBuf, s32 AscLen, s8 cAlignment, u8\* pszAsciiBuf)** |
| --- | --- | --- |
| **Function** | | **Hex code to ASCII code** |
| **Params** | **in** | The length of ASCII code data, that is the double  **AscLen**  length of Bcd code data |
| **cAlignment** Alignment |
| **pszBcdBuf** Bcd code data to be converted |
| **out** | **AscBuf** Conversion output ASCII code data |
| **return** | | **Others Failed** |
| **MFSDK\_UTIL\_RET\_**  **Success**  **OK** |
| **remark** | |  |
| **demo** | | char AscBuf[9] = {0};  char \*BcdBuf = "\x12\x23"; int ret = -1;  memset(AscBuf,0,sizeof(AscBuf)); MfSdkUtilHex2Asc(BcdBuf, 4, 1, AscBuf); APP\_TRACE("MfSdkUtilHex2Asc1:%s\r\n",AscBuf); memset(AscBuf,0,sizeof(AscBuf)); MfSdkUtilHex2Asc(BcdBuf, 4, 0, AscBuf); APP\_TRACE("MfSdkUtilHex2Asc2:%s\r\n",AscBuf); memset(AscBuf,0,sizeof(AscBuf)); MfSdkUtilHex2Asc(BcdBuf, 3, 1, AscBuf); APP\_TRACE("MfSdkUtilHex2Asc3:%s\r\n",AscBuf); memset(AscBuf,0,sizeof(AscBuf)); MfSdkUtilHex2Asc(BcdBuf, 3, 0, AscBuf); APP\_TRACE("MfSdkUtilHex2Asc4:%s\r\n",AscBuf);  Result: |



MfSdkUtilHex2Asc1:1223 MfSdkUtilHex2Asc2:1223 MfSdkUtilHex2Asc3:223 MfSdkUtilHex2Asc4:122

## MfSdkUtilInt2Bcd

| **Prototype** | | **LIB\_EXPORT s32 MfSdkUtilInt2Bcd(u32 IntData, s8 \*BcdBuf, s32 BcdLen)** |
| --- | --- | --- |
| **Function** | | **Int data transfer to BCD code** |
| **Params** | **in** | **BcdLen** Length of BCD code data after conversion |
| **IntData** Int data to be converted |
| **out** | **BcdBuf** BCD data after conversion |
| **return** | | **Others Failed, Ref. MfSdkUtilRet\_E** |
| **MFSDK\_UTIL\_RET\_**  **Success**  **OK** |
| **remark** | |  |
| **demo** | | s8 bcdbuf[16] = {0}; MfSdkUtilInt2Bcd(123,bcdbuf, 2);  APP\_TRACE\_BUFF\_TIP(bcdbuf, 2, "MfSdkUtilInt2Bcd1"); memset(bcdbuf,0,sizeof(bcdbuf)); MfSdkUtilInt2Bcd(123,bcdbuf, 4);  APP\_TRACE\_BUFF\_TIP(bcdbuf, 4, "MfSdkUtilInt2Bcd4"); Result:  MfSdkUtilInt2Bcd1 01 23  MfSdkUtilInt2Bcd4  00 00 01 23 |

## MfSdkUtilBcd2Int

| **Prototype** | | **LIB\_EXPORT s32 MfSdkUtilBcd2Int( s8 \*BcdBuf, u32 \*IntData,s32 BcdLen)** |
| --- | --- | --- |
| **Function** | | **BCD code to int type** |
| **Params** | **in** | **BcdLen** BCD code data length |
| **BcdBuf** BCD data to be converted |



|  | **out** | **IntData** Int data after conversion |
| --- | --- | --- |
| **return** | | **Others Failed, Ref. MfSdkUtilRet\_E** |
| **MFSDK\_UTIL\_RET**  **Success**  **\_OK** |
| **remark** | |  |
| **demo** | | u32 IntData = 0;  char \*BcdBuf = "\x12\x23"; int ret = -1;  ret = MfSdkUtilBcd2Int(BcdBuf, &IntData , 2); APP\_TRACE("MfSdkUtilBcd2Int:%d\r\n",IntData); Result:  MfSdkUtilBcd2Int:1223 |

## MfSdkUtilGenLrc

| **Prototype** | | **LIB\_EXPORT u8 MfSdkUtilGenLrc(s8 \*Data, s32 DataLen)** |
| --- | --- | --- |
| **Function** | | **Calculates LRC and generates LRC parity bits (bitwise exclusive or)** |
| **Params** | **in** | **Data** Data to be calculated for LRC check bits |
| **DataLen** Data length |
| **return** | | **LRC** The LRC check value generated by calculation |
| **remark** | |  |
| **demo** | | char\* Data = “xxxxxxxxxx”; unsigned char LRC;  LRC = MfSdkUtilGenLrc(Data, strlen(Data)); |

## MfSdkUtilDes

| **Prototype** | | **LIB\_EXPORT s32 MfSdkUtilDes(u8 bDesType, u8 bDesMode, s8**  **\*Key, s8 \*InData, s32 DataLen, s8 \*OutData)** |
| --- | --- | --- |
| **Function** | | **DES encryption and decryption to DES data encryption and decryption, or 3DES encryption and decryption** |
| **Params** | **in** | DES encryption and decryption algorithm: 0 denotes DES encryption,  **bDesType** 1 denotes DES decryption,   1. denotes 3DES encryption(16 bytes key), 2. denotes 3DES decryption(16 bytes key), |



|  |  | 1. denotes 3DES encryption(24 bytes key), 2. denotes 3DES decryption(24 bytes key). |
| --- | --- | --- |
| **bDesMode** 0 ECB 1 CBC |
| Used for encryption and decryption of  **Key**  transmission keys, must be 8 times |
| Input data to be encrypted or decrypted must  **InData**  be 8 times bytes |
| **DataLen** The length of input data |
| **out** | The key after encryption and decryption must  **OutData**  be 8 bytes |
| **return** | | **Others Failed, Ref. MfSdkUtilRet\_E** |
| **MFSDK\_UTIL\_RET\_**  **Success**  **OK** |
| **remark** | |  |
| **demo** | |  |

## MfSdkUtilBeep

| **Prototype** | | **LIB\_EXPORT void MfSdkUtilBeep(s32 num)** |
| --- | --- | --- |
| **Function** | | **Buzzing success indicates buzzing when successful, non blocking** |
| **Params** | **in** | **num** Buzz times |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | | MfSdkUtilBeep(100); |

## MfSdkUtilBuzzerSound

| **Prototype** | | **LIB\_EXPORT void MfSdkUtilBuzzerSound(s32 nMillisecond)** |
| --- | --- | --- |
| **Function** | | **Set the buzzer ringing time** |
| **Params** | **in** | **nMillisecond** ringing time(Unit millisecond) |
| **out** | **Nothing** |
| **return** | | **Nothing** |



| **remark** |  |
| --- | --- |
| **demo** | MfSdkUtilBuzzerSound(100); |

## MfSdkUtilGetRand

| **Prototype** | | **LIB\_EXPORT s32 MfSdkUtilGetRand(s32 len, u8\* pBuf)** |
| --- | --- | --- |
| **Function** | | **Generating random numbers** |
| **Params** | **in** | **len** random number len |
| **out** | **pBuf** random number string |
| **return** | | **Others Failed, Ref. MfSdkUtilRet\_E** |
| **MFSDK\_UTIL\_RET**  **Success**  **\_OK** |
| **remark** | |  |
| **demo** | | unsigned char randomNumberStr[6] = {0}; int ret = -1;  ret = MfSdkUtilGetRand(5, randomNumberStr ); |

## MfSdkUtilSHA1

| **Prototype** | | **LIB\_EXPORT s32 MfSdkUtilSHA1(const u8\* psSrc, u32 nSrcLen, u8\* psDst)** |
| --- | --- | --- |
| **Function** | | **SHA1 calculation** |
| **Params** | **in** | **psSrc** Source buffer |
| **nSrcLen** Source length |
| **out** | **psDst** Result of SHA1 |
| **return** | | **Others Failed, Ref. MfSdkUtilRet\_E** |
| **MFSDK\_UTIL\_RET\_**  **Success**  **OK** |
| **remark** | |  |
| **demo** | | unsigned char\* psSrc = "demotest"; unsigned char psDst[20+1] = {0}; int ret = -1;  ret = MfSdkUtilSHA1((const unsigned char\*)psSrc, strlen(psSrc), psDst); |



## MfSdkUtilGeneCodePic

| **Prototype** | | **LIB\_EXPORT s32 MfSdkUtilGeneCodePic(s8 \* chData, s32 iLen, MfSdkUtilQrInfo\_T \*qrparam , s8 \* bitmap)** |
| --- | --- | --- |
| **Function** | | **QR code generation** |
| **Params** | **in** | **chData** QR code data |
| **iLen** Data length |
| **qrparam** QR code parameters |
| **out** | **bitmap** Generated two-dimensional code dot matrix data |
| **return** | | **Others Failed, Ref. MfSdkUtilRet\_E** |
| **>0 Successfully generated QR code width** |
| **remark** | |  |
| **demo** | | #define QR\_HEIGHT 80  MfSdkUtilQrInfo\_T qrparam;  char \*bitmap = MALLOC(QR\_HEIGHT\*QR\_HEIGHT/8); char \*qrData = "[http://en.morefun-et.com](http://en.morefun-et.com/)";  int length = strlen(qrData); int width = 0;  int Qr\_width = MfSdkGuiGetWidth(); qrparam.nVersion = 0; //auto qrparam.nLevel = 0;  qrparam.moduleWidth = 1;  while(qrparam.moudleWidth > 0 && (width <= 0 || width > Qr\_heith))  {  memset(bitmap , 0 , QR\_HEIGHT\*QR\_HEIGHT/8);  width = MfSdkUtilGeneCodePic(data , length, &qrparam, bitmap); qrparam.moudleWidth --;  } |

## MfSdkUtilLed

| **Prototype** | **LIB\_EXPORT void MfSdkUtilLed(s32 num, s32 type)** |
| --- | --- |
| **Function** | **LED light control** |



| **Params** | **in** | **num** LED light number  (0 red, 1 blue, 2 yellow, 3 green) |
| --- | --- | --- |
| **type** LED light switch(0 close, 1 open) |
| **out** | **Nothing** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | | MfSdkUtilLed(1,1); // open blue led |

## MfSdkUtilStr2Longlong

| **Prototype** | | **LIB\_EXPORT long long MfSdkUtilStr2Longlong(const s8 \*amt)** |
| --- | --- | --- |
| **Function** | | **String to long long** |
| **Params** | **in** | **amt** amount string |
| **out** | **Nothing** |
| **return** | | **long long** amount in long long type |
| **remark** | |  |
| **demo** | | char \*amt = "1533352";  long long amt = MfSdkUtilStr2Longlong((const char\*)amt); |

## MfSdkUtilMd5File

| **Prototype** | | **LIB\_EXPORT s32 MfSdkUtilMd5File(const s8\* pszFilePath, u8\* pOutMD5)** |
| --- | --- | --- |
| **Function** | | **File MD5 value calculation** |
| **Params** | **in** | **pszFilePath** File Path |
| **out** | **pOutMD5** MD5 result |
| **return** | | **MFSDK\_FALSE** file not found |
| **MFSDK\_TRUE** success |
| **remark** | |  |
| **demo** | | #define FILE\_TEST "data\\test.txt" unsigned char \*pOutMD5 = NULL;  int ret = MfSdkUtilMd5File(FILE\_TEST, pOutMD5); |



## MfSdkUtilUtf8str2Astr

| **Prototype** | | **LIB\_EXPORT s32 MfSdkUtilUtf8str2Astr(char \*utfstr, s32 utfchars, u8\*astr,s32 buffsize)** |
| --- | --- | --- |
| **Function** | | **Utf-8 string to ASCII string** |
| **Params** | **in** | **utfstr** utf-8 string |
| **utfchars** utf-8 string length |
| **buffsize** astr buffer size |
| **out** | **astr** out ascii string |
| **return** | | **Others Failed, Ref. MfSdkUtilRet\_E** |
| **>0 ascii length** |
| **remark** | |  |
| **demo** | | char AscBuf[50] = {0}; char \*utfstr= “test”;  int ret = -1;  ret = MfSdkUtilUtf8str2Astr(utfstr, strlen(utfstr), AscBuf, sizeof(AscBuf)); |

## MfSdkUtilUnicodeToUtf8

| **Prototype** | | **LIB\_EXPORT s32 MfSdkUtilUnicodeToUtf8(u16 codepoint, s8**  **\*outstr)** |
| --- | --- | --- |
| **Function** | | **Unicode data transfer to utf-8 code** |
| **Params** | **in** | **codepoint** Unicode data to be converted |
| **out** | **outstr** utf-8 data after conversion |
| **return** | | **Others Failed, Ref. MfSdkUtilRet\_E** |
| **>0 The length of outstr** |
| **remark** | |  |
| **demo** | | char utfstr[50] = {0}; char \*unicode= “test”; int utflen= -1;  utflen = MfSdkUtilUnicodeToUtf8((unsigned short)unicode, utfstr); |



## MfSdkUtilUtf8ToUnicode

| **Prototype** | | **LIB\_EXPORT s32 MfSdkUtilUtf8ToUnicode(u8 \*instr, u8 \*outstr)** |
| --- | --- | --- |
| **Function** | | **utf-8 code data transfer to Unicode** |
| **Params** | **in** | **instr** utf-8 data to be converted |
| **out** | **outstr** Unicode data after conversion |
| **return** | | **Others Failed, Ref. MfSdkUtilRet\_E** |
| **>0 The length of outstr** |
| **remark** | |  |
| **demo** | | char unicodestr[50] = {0}; char \*utfstr= “test”;  int unilen= -1;  unilen = MfSdkUtilUtf8ToUnicode(utfstr, unicodestr); |

## MfSdkUtilTlvGetDataByTag

| **Prototype** | | **LIB\_EXPORT s32 MfSdkUtilTlvGetDataByTag(u8\*tagName,u8**  **\*pInBuf,s32 inBufLength,u8 \*pValue, s32 valueLength)** |
| --- | --- | --- |
| **Function** | | **Get TLV data by tag** |
| **Params** | **in** | **tagName** tag name eg. "\x9F\x02" |
| **pInBuf** tlv bytes stream |
| **inBufLength** tlv bytes stream length |
| **valueLength** pValue buffer size |
| **out** | **pValue** tagName value |
| **return** | | **Others Failed, Ref. MfSdkUtilRet\_E** |
| **>0 pValue length** |
| **remark** | |  |
| **demo** | |  |



## MfSdkUtilCtrlBeep

| **Prototype** | | **LIB\_EXPORT void MfSdkUtilCtrlBeep(int ms, int hz)** |
| --- | --- | --- |
| **Function** | | **Set the buzzer ringing time and Hz** |
| **Params** | **in** | **ms** beep time ms |
| **hz** beep hz |
| **out** | **Nothing** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | | //750Hz ringing 500ms  MfSdkUtilCtrlBeep(500,750); |

## MfSdkUtilGenRsa

| **Prototype** | | **LIB\_EXPORT s32 MfSdkUtilGenRsa(s32 keyBits,u8 \*privPem, s32 privPemLength,u8 \*pubPem, s32 pubPemLength,s32 timeoutMs)** |
| --- | --- | --- |
| **Function** | | **Generate RSA public and private keys. The exponent is fixed at 65537** |
| **Params** | **in** | **keyBits** RSA key bits |
| **privPemLength** privPem buffer size |
| **pubPemLength** pubPem buffer size |
| **timeoutMs** timeout ms |
| **out** | **privPem** private key, format is pem |
| **pubPem** public key format is pem |
| **return** | | **Others Failed, Ref. MfSdkUtilRet\_E** |
| **0 Success** |
| **remark** | |  |
| **demo** | |  |

## MfSdkUtilGenRsaKey

**LIB\_EXPORT s32 MfSdkUtilGenRsaKey(u8 index,s32 keyBits,s32 timeoutMs)**

****

| **Function** | | **Generate RSA key** |
| --- | --- | --- |
| **Params** | **in** | **index** index |
| **keyBits** 1024 - 4096 bits |
| **timeoutMs** timeout ms |
| **out** | **Nothing** |
| **return** | | **0 Success** |
| **Other Failed, Ref. MfSdkUtilRet\_E** |
| **remark** | |  |
| **demo** | |  |

## MfSdkUtilRsaSign

| **Prototype** | | **LIB\_EXPORT s32 MfSdkUtilRsaSign(u8**  **index ,MfSdkUtilMDType\_E mdAlg, u8\* hash , s32 hashLength,u8\* signData,s32 signDataLength)** |
| --- | --- | --- |
| **Function** | | **Generate RSA signature** |
| **Params** | **in** | **index** |
| **mdAlg** Ref. **MfSdkUtilMDType\_E** |
| **hash** |
| **hashLength** |
| **out** | **signData** |
| **signDataLength** |
| **return** | | **Others Failed, Ref. MfSdkUtilRet\_E** |
| **>0 Success** |
| **remark** | |  |
| **demo** | |  |

* 1. MfSdkUtilRsaGetPublicKey

| **Prototype** | | **LIB\_EXPORT s32 MfSdkUtilRsaGetPublicKey(u8 index , u8**  **\*pubPem, s32 pubPemLength)** |
| --- | --- | --- |
| **Function** | | **Get RSA public key** |
| **Params** | **in** | **index** |



|  |  | **pubPemLength** |
| --- | --- | --- |
| **out** | **pubPem** key string |
| **return** | | **0 Success** |
| **Others Failed, Ref. MfSdkUtilRet\_E** |
| **remark** | |  |
| **demo** | |  |

* 1. MfSdkUtilGenRsaDelKey

| **Prototype** | | **LIB\_EXPORT s32 MfSdkUtilGenRsaDelKey(u8 index)** |
| --- | --- | --- |
| **Function** | | **Delete index RSA key** |
| **Params** | **in** | **index** |
| **out** | **Nothing** |
| **return** | | **MFSDK\_UTIL\_RET\_**  **Success**  **OK** |
| **Others Failed, Ref. MfSdkUtilRet\_E** |
| **remark** | |  |
| **demo** | |  |

* 1. MfSdkUtilGenRsaDelAllKey

| **Prototype** | | **LIB\_EXPORT s32 MfSdkUtilGenRsaDelAllKey(void)** |
| --- | --- | --- |
| **Function** | | **Delete all RSA key** |
| **Params** | **in** | **Nothing** |
| **out** | **Nothing** |
| **return** | | **MFSDK\_UTIL\_RET\_**  **Success**  **OK** |
| **Others Failed, Ref. MfSdkUtilRet\_E** |
| **remark** | |  |
| **demo** | |  |



* 1. MfSdkUtilRsaRand

| **Prototype** | | **LIB\_EXPORT s32 MfSdkUtilRsaRand(void\* para, u8\* pbuf, u32 len)** |
| --- | --- | --- |
| **Function** | | **For mbedtls rand** |
| **Params** | **in** | **para** |
| **pbuf** |
| **len** |
| **out** | **Nothing** |
| **return** | | **Others Failed, Ref. MfSdkUtilRet\_E** |
| **0 Success** |
| **remark** | |  |
| **demo** | |  |

* 1. MfSdkUtilCompress

| **Prototype** | | **LIB\_EXPORT s32 MfSdkUtilCompress (u8\*dest, u32 \*destLen, const u8\* source, u32 sourceLen)** |
| --- | --- | --- |
| **Function** | | **Compresses the source buffer into the destination buffer. The level parameter has the same meaning as in deflateInit. sourceLen is the byte length of the source buffer. Upon entry, destLen is the total size of the destination buffer, which must be at least 0.1% larger than sourceLen plus 12 bytes. Upon exit,**  **destLen is the actual size of the compressed buffer.** |
| **Params** | **in** | **destLen** |
| **source** |
| **sourceLen** |
| **out** | **destLen** |
| **dest** |
| **return** | | **Others Failed, Ref. MfSdkUtilRet\_E** |
| **MFSDK\_UTIL\_RET\_**  **Success**  **OK** |
| **remark** | |  |
| **demo** | |  |



* 1. MfSdkUtilCompressBound

| **Prototype** | | **LIB\_EXPORT u32 MfSdkUtilCompressBound (u32 sourceLen)** |
| --- | --- | --- |
| **Function** | | **get compress bound** |
| **Params** | **in** | **sourceLen** |
| **out** | **Nothing** |
| **return** | | **compress bound**  **value** |
| **remark** | |  |
| **demo** | |  |

* 1. MfSdkUtilSave2Zip

| **Prototype** | | **LIB\_EXPORT s32 MfSdkUtilSave2Zip(const char \*zip\_filename, const char \*file\_name\_in\_zip, const u8 \*data, u32 data\_len)** |
| --- | --- | --- |
| **Function** | | **save data to zip file** |
| **Params** | **in** | **zip\_filename** zip file name |
| **file\_name\_in\_zip** compress data file name |
| **data** compress data |
| **data\_len** compress data length |
| **out** | **Nothing** |
| **return** | | **Others Failed, Ref. MfSdkUtilRet\_E** |
| **MFSDK\_UTIL\_RET\_**  **Success**  **OK** |
| **remark** | |  |
| **demo** | |  |

# RFID module

## Module description

This module mainly includes NFC APIs.



## Module structure declaration

None.

## Constant declarations

typedef enum

{

MFSDK\_NFC\_RET\_MUTILCARD = MFSDK\_ICC\_RET\_MUTILCARD,

// more than one card

MFSDK\_NFC\_RET\_NOCARD = MFSDK\_ICC\_RET\_NOCARD, // dont

find card

MFSDK\_NFC\_RET\_FAILED = MFSDK\_ICC\_RET\_FAILED, //failed MFSDK\_NFC\_RET\_BOUNDS = MFSDK\_ICC\_RET\_BOUNDS, //Array

out-of-bounds

MFSDK\_NFC\_RET\_PARM\_ERROR =

MFSDK\_ICC\_RET\_PARM\_ERROR, //check param MFSDK\_NFC\_RET\_OK = 0,

}MfSdkNfcRet\_E;

typedef enum

{

MFSDK\_NFC\_LED\_BLUE = 0, MFSDK\_NFC\_LED\_YELLOW, MFSDK\_NFC\_LED\_GREEN, MFSDK\_NFC\_LED\_RED,

}MfSdkNfcLed\_E;

typedef enum

{

MFSDK\_NFC\_LED\_OFF = 0,

MFSDK\_NFC\_LED\_ON = 1,

}MfSdkNfcLedSwitch\_E;

## MfSdkNfcInit

| **Prototype** | | **LIB\_EXPORT s32 MfSdkNfcInit(void)** |
| --- | --- | --- |
| **Function** | | **Initialization of the NFC module.** |
| **Params** | **in** | **None** |
| **out** | **None** |



| **return** | **Others Failed, Ref. MfSdkNfcRet\_E** |
| --- | --- |
| **MFSDK\_NFC\_RET\_**  **Success**  **OK** |
| **remark** |  |
| **demo** |  |

## MfSdkNfcApdu

| **Prototype** | | **LIB\_EXPORT s32 MfSdkNfcApdu(u8 \*tbuf, u16 tlen, u8 \*\*rbuf, u16**  **\*rlen)** |
| --- | --- | --- |
| **Function** | | **NFC APDU exchange.** |
| **Params** | **in** | **tbuf** apdu request command |
| **tlen** apdu request command length |
| **out** | **rbuf** eg. unsigned char \*pTemp &pTemp |
| **rlen** pdu response length point |
| **return** | | **MFSDK\_NFC\_RET\_FAIL**  **Failed, Ref. MfSdkNfcRet\_E**  **ED** |
| **MFSDK\_NFC\_RET\_PAR**  **Invalid parameters.**  **M\_ERROR** |
| **MFSDK\_NFC\_ETR\_OK Success** |
| **remark** | |  |
| **demo** | | s32 iRet = MFSDK\_NFC\_RET\_FAILED;  u8 \*pRApdu = NULL; u16 rApduLength = 0; u8 cApud[] =  {  0x00,0xA4,0x04,0x00,0x0E,0x32,0x50,0x41,  0x59,0x2E,0x53,0x59,0x53,0x2E,0x44,0x44,  0x46,0x30,0x31,0x00  };  iRet = MfSdkNfcApdu(cApud,sizeof(cApud),&pRApdu , &rApduLength); if(MFSDK\_NFC\_RET\_OK == iRet)  {  //Success TODO  }  else |



{

//Fail

}

## MfSdkNfcClose

| **Prototype** | | **LIB\_EXPORT s32 MfSdkNfcClose(void)** |
| --- | --- | --- |
| **Function** | | **Turn off the NFC module.** |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **Others Failed, Ref. MfSdkNfcRet\_E** |
| **MFSDK\_NFC\_RET\_OK Success** |
| **remark** | |  |
| **demo** | |  |

## MfSdkNfcCtlsComm

| **Prototype** | | **LIB\_EXPORT s32 MfSdkNfcCtlsComm(s32 iCardType,MfSdkIccApdu\_T \*Apdu)** |
| --- | --- | --- |
| **Function** | | **NFC Ctls command** |
| **Params** | **in** | **iCardType** |
| **Apdu** |
| **out** | **None** |
| **return** | | **Others Failed, Ref. MfSdkNfcRet\_E** |
| **MFSDK\_NFC\_RET\_OK Success** |
| **remark** | |  |
| **demo** | |  |

## MfSdkNfcCtlsPowerDown

| **Prototype** | **LIB\_EXPORT s32 MfSdkNfcCtlsPowerDown(void)** |
| --- | --- |



| **Function** | | **NFC Ctls Power Down** |
| --- | --- | --- |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **Others Failed, Ref. MfSdkNfcRet\_E** |
| **MFSDK\_NFC\_RET\_OK Success** |
| **remark** | |  |
| **demo** | |  |

## MfSdkNfcCtlsPowerUpAndSeek

| **Prototype** | | **LIB\_EXPORT s32 MfSdkNfcCtlsPowerUpAndSeek(int iCardType, char \*psUID)** |
| --- | --- | --- |
| **Function** | | **NFC Ctls Power Up and seek** |
| **Params** | **in** | **iCardType** |
| **psUID** |
| **out** | **None** |
| **return** | | **Others Failed, Ref. MfSdkNfcRet\_E** |
| **MFSDK\_NFC\_RET\_OK Success** |
| **remark** | |  |
| **demo** | |  |

## MfSdkNfcDetect

| **Prototype** | | **LIB\_EXPORT s32 MfSdkNfcDetect(void)** |
| --- | --- | --- |
| **Function** | | **Check the user whether tap card .** |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **Others Failed, Ref. MfSdkNfcRet\_E** |
| **MFSDK\_NFC\_RET\_NOC**  **No card**  **ARD** |
| **MFSDK\_NFC\_RET\_MUTI**  **multiple card**  **LCARD** |



|  | **MFSDK\_NFC\_RET\_OK Success** |
| --- | --- |
| **remark** |  |
| **demo** |  |

## MfSdkNfcApduStream

| **Prototype** | | **LIB\_EXPORT s32 MfSdkNfcApduStream(u8 \*pC\_Apdu, s32 clen, u8 \*pR\_Apdu, s32 rlen)** |
| --- | --- | --- |
| **Function** | | Bytes stream apdu command |
| **Params** | **in** | **pC\_Apdu** CAPDU bytes stream |
| **clen** CAPDU bytes stream length |
| **rlen** recv buffer size |
| **out** | **pR\_Apdu** RAPDU recv buffer |
| **return** | | **Others Failed, Ref. MfSdkNfcRet\_E** |
| **> 0 R\_Apdu length** |
| **MFSDK\_ICC\_RET\_PARM\_E**  **params error**  **RROR** |
| **MFSDK\_ICC\_RET\_BOUNDS rlen < card response length** |
| **remark** | |  |
| **demo** | |  |

## MfSdkNfcGetUid

| **Prototype** | | **LIB\_EXPORT s32 MfSdkNfcGetUid(u8\* uid)** |
| --- | --- | --- |
| **Function** | | Get card UID |
| **Params** | **in** | **uid** |
| **out** | **None** |
| **return** | | **Others Failed, Ref. MfSdkNfcRet\_E** |
| **> 0 uid length** |
| **remark** | |  |
| **demo** | |  |



## MfSdkNfcIsProbe

| **Prototype** | | **LIB\_EXPORT s32 MfSdkNfcIsProbe(void)** |
| --- | --- | --- |
| **Function** | | Check whether the NFC card is detected |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **Others Failed, Ref. MfSdkNfcRet\_E** |
| **1 Success** |
| **remark** | |  |
| **demo** | |  |

## MfSdkNfcM1Atqa

| **Prototype** | | **LIB\_EXPORT s32 MfSdkNfcM1Atqa(void)** |
| --- | --- | --- |
| **Function** | |  |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **Others Failed, Ref. MfSdkNfcRet\_E** |
| **MFSDK\_NFC\_RET\_**  **Success**  **OK** |
| **remark** | |  |
| **demo** | |  |

## MfSdkNfcM1Auth

| **Prototype** | | **LIB\_EXPORT s32 MfSdkNfcM1Auth(s32 cmd, s32 block)** |
| --- | --- | --- |
| **Function** | |  |
| **Params** | **in** | **cmd** |



|  |  | **block** |
| --- | --- | --- |
| **out** | **None** |
| **return** | | **Others Failed, Ref. MfSdkNfcRet\_E** |
| **MFSDK\_NFC\_RET\_**  **Success**  **OK** |
| **remark** | |  |
| **demo** | |  |

## MfSdkNfcM1Close

| **Prototype** | | **LIB\_EXPORT s32 MfSdkNfcM1Close(void)** |
| --- | --- | --- |
| **Function** | |  |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **Others Failed, Ref. MfSdkNfcRet\_E** |
| **MFSDK\_NFC\_RET\_**  **Success**  **OK** |
| **remark** | |  |
| **demo** | |  |

## MfSdkNfcM1Decrement

| **Prototype** | | **LIB\_EXPORT s32 MfSdkNfcM1Decrement(s32 block, s32 operand)** |
| --- | --- | --- |
| **Function** | |  |
| **Params** | **in** | **block** |
| **operand** |
| **out** | **None** |
| **return** | | **Others Failed, Ref. MfSdkNfcRet\_E** |
| **MFSDK\_NFC\_RET\_**  **Success**  **OK** |
| **remark** | |  |
| **demo** | |  |



## MfSdkNfcM1Increment

| **Prototype** | | **LIB\_EXPORT s32 MfSdkNfcM1Increment(s32 block, s32 operand)** |
| --- | --- | --- |
| **Function** | |  |
| **Params** | **in** | **block** |
| **operand** |
| **out** | **None** |
| **return** | | **Others Failed, Ref. MfSdkNfcRet\_E** |
| **MFSDK\_NFC\_RET\_**  **Success**  **OK** |
| **remark** | |  |
| **demo** | |  |

## MfSdkNfcM1Open

| **Prototype** | | **LIB\_EXPORT s32 MfSdkNfcM1Open(void)** |
| --- | --- | --- |
| **Function** | | M1 open |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **Others Failed, Ref. MfSdkNfcRet\_E** |
| **MFSDK\_NFC\_RET\_**  **Success**  **OK** |
| **remark** | |  |
| **demo** | |  |

## MfSdkNfcM1Read

**Prototype LIB\_EXPORT s32 MfSdkNfcM1Read(s32 block, u8\* buf, s32 \*len)**

****

| **Function** | |  |
| --- | --- | --- |
| **Params** | **in** | **block** |
| **len** |
| **out** | **buf** |
| **return** | | **Others Failed, Ref. MfSdkNfcRet\_E** |
| **MFSDK\_NFC\_RET\_**  **Success**  **OK** |
| **remark** | |  |
| **demo** | |  |

## MfSdkNfcM1Restore

| **Prototype** | | **LIB\_EXPORT s32 MfSdkNfcM1Restore(s32 block)** |
| --- | --- | --- |
| **Function** | |  |
| **Params** | **in** | **block** |
| **out** | **None** |
| **return** | | **Others Failed, Ref. MfSdkNfcRet\_E** |
| **MFSDK\_NFC\_RET\_**  **Success**  **OK** |
| **remark** | |  |
| **demo** | |  |

## MfSdkNfcM1SetKey

| **Prototype** | | **LIB\_EXPORT s32 MfSdkNfcM1SetKey(u8 \*key)** |
| --- | --- | --- |
| **Function** | |  |
| **Params** | **in** | **key** |
| **out** | **None** |
| **return** | | **Others Failed, Ref. MfSdkNfcRet\_E** |
| **MFSDK\_NFC\_RET\_**  **Success**  **OK** |



| **remark** |  |
| --- | --- |
| **demo** |  |

## MfSdkNfcM1Transfer

| **Prototype** | | **LIB\_EXPORT s32 MfSdkNfcM1Transfer(s32 block)** |
| --- | --- | --- |
| **Function** | |  |
| **Params** | **in** | **block** |
| **out** | **None** |
| **return** | | **Others Failed, Ref. MfSdkNfcRet\_E** |
| **MFSDK\_NFC\_RET\_**  **Success**  **OK** |
| **remark** | |  |
| **demo** | |  |

## MfSdkNfcM1Uid

| **Prototype** | | **LIB\_EXPORT s32 MfSdkNfcM1Uid(u8 \*uid)** |
| --- | --- | --- |
| **Function** | |  |
| **Params** | **in** | **uid** |
| **out** | **None** |
| **return** | | **Others Failed, Ref. MfSdkNfcRet\_E** |
| **MFSDK\_NFC\_RET\_**  **Success**  **OK** |
| **remark** | |  |
| **demo** | |  |

## MfSdkNfcM1Write

**Prototype LIB\_EXPORT s32 MfSdkNfcM1Write(s32 block, u8 \*buf, s32 len)**

****

| **Function** | |  |
| --- | --- | --- |
| **Params** | **in** | **block** |
| **buf** |
| **len** |
| **out** | **None** |
| **return** | | **Others Failed, Ref. MfSdkNfcRet\_E** |
| **MFSDK\_NFC\_RET\_**  **Success**  **OK** |
| **remark** | |  |
| **demo** | |  |

## MfSdkNfcOpen

| **Prototype** | | **LIB\_EXPORT s32 MfSdkNfcOpen(void)** |
| --- | --- | --- |
| **Function** | | NFC open |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **Others Failed, Ref. MfSdkNfcRet\_E** |
| **MFSDK\_NFC\_RET\_**  **Success**  **OK** |
| **remark** | |  |
| **demo** | |  |

## MfSdkNfcReset

| **Prototype** | | **LIB\_EXPORT s32 MfSdkNfcReset(u8\* pAtr, s32 iAtrLength)** |
| --- | --- | --- |
| **Function** | | NFC Reset |
| **Params** | **in** | **pAtr** |
| **iAtrLength** |
| **out** | **None** |
| **return** | | **Others Failed, Ref. MfSdkNfcRet\_E** |



|  | **MFSDK\_NFC\_RET\_**  **Success**  **OK** |
| --- | --- |
| **remark** |  |
| **demo** |  |

## MfSdkNfcTagEmulateInitSetData

| **Prototype** | | **s32 MfSdkNfcTagEmulateInitSetData(s8\* data, s32 inDataLen, s8\* code)** |
| --- | --- | --- |
| **Function** | | ntag set data initialization |
| **Params** | **in** | **data** data |
| **inDataLen** data len |
| **code** data encoding format(example "utf-8") |
| **out** | **None** |
| **return** | | **Others Failed, Ref. MfSdkNfcRet\_E** |
| **MFSDK\_NFC\_RET\_**  **Success**  **OK** |
| **remark** | |  |
| **demo** | |  |

## MfSdkNfcTagEmulateInitSetUrl

| **Prototype** | | **s32 MfSdkNfcTagEmulateInitSetUrl(u8\* url)** |
| --- | --- | --- |
| **Function** | | ntag set url initialization |
| **Params** | **in** | **url** set url |
| **out** | **None** |
| **return** | | **Others Failed, Ref. MfSdkNfcRet\_E** |
| **MFSDK\_NFC\_RET\_**  **Success**  **OK** |
| **remark** | |  |
| **demo** | |  |



## MfSdkNfcTagEmulateProcess

| **Prototype** | | **void MfSdkNfcTagEmulateProcess(void)** |
| --- | --- | --- |
| **Function** | | ntag execution processing |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **None** |
| **remark** | |  |
| **demo** | |  |

## MfSdkNfcTagEmulateDeinit

| **Prototype** | | **s32 MfSdkNfcTagEmulateDeinit(void)** |
| --- | --- | --- |
| **Function** | | ntag deinitialization |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **MFSDK\_NFC\_RET\_**  **Success**  **OK** |
| **remark** | |  |
| **demo** | |  |

## MfSdkNfcLed

| **Prototype** | | **LIB\_EXPORT void MfSdkNfcLed(MfSdkNfcLed\_E num, MfSdkNfcLedSwitch\_E type)** |
| --- | --- | --- |
| **Function** | | led control |
| **Params** | **in** | **num** led color, Ref. **MfSdkNfcLed\_E** |
| **type** switch type, Ref. **MfSdkNfcLedSwitch\_E** |



|  | **out** | **None** |
| --- | --- | --- |
| **return** | | **None** |
| **remark** | |  |
| **demo** | |  |

## MfSdkNfcIsAroundFrontLcd

| **Prototype** | | **LIB\_EXPORT MFSDKBOOL MfSdkNfcIsAroundFrontLcd(void)** |
| --- | --- | --- |
| **Function** | | Check whether the NFC antenna is around the LCD |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **MFSDK\_TRUE nfc antenna around lcd** |
| **remark** | |  |
| **demo** | |  |

# Magstripe module

## Module description

This module mainly includes Magstripe Card APIs.

## Module structure declaration

None.

## Constant declarations

typedef enum

{

MFSDK\_MAG\_RET\_FAILED = -3, //failed MFSDK\_MAG\_RET\_BOUNDS = -2, //Array out-of-bounds MFSDK\_MAG\_RET\_PARM\_ERROR = -1, //check param MFSDK\_MAG\_RET\_OK = 0,

}MfSdkMagRet\_E;



## MfSdkMagStripeDetect

| **Prototype** | | **LIB\_EXPORT s32 MfSdkMagStripeDetect(MfSdkMagTraceInfo\_T**  **\*trackinfo)** |
| --- | --- | --- |
| **Function** | | Magstripe Card Detect |
| **Params** | **in** | **trackinfo** |
| **out** | **None** |
| **return** | | **Others Failed, Ref. MfSdkMagRet\_E** |
| **MFSDK\_MAG\_RET**  **Success**  **\_OK** |
| **remark** | |  |
| **demo** | |  |

## MfSdkMagtekClose

| **Prototype** | | **LIB\_EXPORT void MfSdkMagtekClose(void)** |
| --- | --- | --- |
| **Function** | |  |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **None** |
| **remark** | |  |
| **demo** | |  |

## MfSdkMagTekFlush

| **Prototype** | | **LIB\_EXPORT s32 MfSdkMagTekFlush(void)** |
| --- | --- | --- |
| **Function** | |  |
| **Params** | **in** | **None** |
| **out** | **None** |



| **return** | **Others Failed, Ref. MfSdkMagRet\_E** |
| --- | --- |
| **MFSDK\_MAG\_RET**  **Success**  **\_OK** |
| **remark** |  |
| **demo** |  |

## MfSdkMagtekOpen

| **Prototype** | | **LIB\_EXPORT void MfSdkMagtekOpen(void)** |
| --- | --- | --- |
| **Function** | |  |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **None** |
| **remark** | |  |
| **demo** | |  |

# ICC module

## Module description

This module mainly includes ICC APIs.

## Module structure declaration

None.

## Constant declarations

typedef enum

{

MFSDK\_ICC\_SLOT\_MIN = 0,

MFSDK\_ICC\_SLOT\_NFC = MFSDK\_ICC\_SLOT\_MIN, //rfid

MFSDK\_ICC\_SLOT\_NFC\_EXTERN , //rfid extern MFSDK\_ICC\_SLOT\_ICC, //contact ic card slot MFSDK\_ICC\_SLOT\_ICC\_EXTERN, //contact ic card slot extern MFSDK\_ICC\_SLOT\_PSAM1,



MFSDK\_ICC\_SLOT\_PSAM2, MFSDK\_ICC\_SLOT\_PSAM3, MFSDK\_ICC\_SLOT\_PSAM4, MFSDK\_ICC\_SLOT\_PSAM5, MFSDK\_ICC\_SLOT\_PSAM6, MFSDK\_ICC\_SLOT\_MAX

}MfSdkIccSlot\_E;

typedef enum

{

MFSDK\_ICC\_RET\_MUTILCARD = -5, // more than one card MFSDK\_ICC\_RET\_NOCARD = -4, // dont find card MFSDK\_ICC\_RET\_FAILED = -3, //failed MFSDK\_ICC\_RET\_BOUNDS = -2, //Array out-of-bounds MFSDK\_ICC\_RET\_PARM\_ERROR = -1, //check param MFSDK\_ICC\_RET\_OK = 0,

}MfSdkIccRet\_E;

typedef enum

{

MFSDK\_ICC\_TYPE\_CPUCARD = 0,

}MfSdkIccType\_E;

## MfSdkIccClose

| **Prototype** | | **LIB\_EXPORT s32 MfSdkIccClose(MfSdkIccSlot\_E iSlotType)** |
| --- | --- | --- |
| **Function** | |  |
| **Params** | **in** | **iSlotType** Ref. **MfSdkIccSlot\_E** |
| **out** | **None** |
| **return** | | **MFSDK\_ICC\_RET\_**  **Success**  **OK** |
| **Other Fail** |
| **remark** | |  |
| **demo** | |  |



## MfSdkIccComm

| **Prototype** | | **LIB\_EXPORT s32 MfSdkIccComm(s32 iCardType,MfSdkIccSlot\_E iSlotType, MfSdkIccApdu\_T \*Apdu)** |
| --- | --- | --- |
| **Function** | |  |
| **Params** | **in** | **iCardType** Ref. **MfSdkIccType\_E** |
| **iSlotType** Ref. **MfSdkIccSlot\_E** |
| **Apdu** |
| **out** | **None** |
| **return** | | **Ref. MfSdkIccRet\_E** |
| **remark** | |  |
| **demo** | |  |

## MfSdkIccGetCardATR

| **Prototype** | | **LIB\_EXPORT s32 MfSdkIccGetCardATR(s32 iCardType, MfSdkIccSlot\_E iSlotType, u8 \*psATR, s32\*pnATRLen)** |
| --- | --- | --- |
| **Function** | |  |
| **Params** | **in** | **iCardType** Ref. **MfSdkIccType\_E** |
| **iSlotType** Ref. **MfSdkIccSlot\_E** |
| **psATR** |
| **pnATRLen** |
| **out** | **None** |
| **return** | | **Ref. MfSdkIccRet\_E** |
| **remark** | |  |
| **demo** | |  |

## MfSdkIccGetCardStatus

**LIB\_EXPORT s32 MfSdkIccGetCardStatus(MfSdkIccSlot\_E iSlotType)**

****

| **Function** | |  |
| --- | --- | --- |
| **Params** | **in** | **iSlotType** Ref. **MfSdkIccSlot\_E** |
| **out** | **None** |
| **return** | | **Ref. MfSdkIccRet\_E** |
| **remark** | |  |
| **demo** | |  |

## MfSdkIccGetModuleVer

| **Prototype** | | **LIB\_EXPORT s32 MfSdkIccGetModuleVer(u8 \*pszVer)** |
| --- | --- | --- |
| **Function** | | get module version |
| **Params** | **in** | **None** |
| **out** | **pszVer** Icc Module Version |
| **return** | | **Ref. MfSdkIccRet\_E** |
| **remark** | |  |
| **demo** | |  |

## MfSdkIccInsertDetect

| **Prototype** | | **LIB\_EXPORT s32 MfSdkIccInsertDetect(void)** |
| --- | --- | --- |
| **Function** | |  |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **Ref. MfSdkIccRet\_E** |
| **remark** | |  |
| **demo** | |  |



## MfSdkIccApdu

| **Prototype** | | **LIB\_EXPORT s32 MfSdkIccApdu(MfSdkIccSlot\_E iSlotType, u8\* pC\_Apdu, s32 cLength,u8\* pR\_Apdu,s32 rLength)** |
| --- | --- | --- |
| **Function** | | apdu bytes stream |
| **Params** | **in** | **iSlotType** Ref. **MfSdkIccSlot\_E** |
| **pC\_Apdu** bytes stream |
| **cLength** |
| **rLength** |
| **out** | **pR\_Apdu** |
| **return** | | **> 0 R\_Apdu length** |
| **MFSDK\_ICC\_RET\_PARM\_**  **params error**  **ERROR** |
| **MFSDK\_ICC\_RET\_BOUND**  **rLength < card response length**  **S** |
| **remark** | |  |
| **demo** | |  |

## MfSdkIccOpen

| **Prototype** | | **LIB\_EXPORT s32 MfSdkIccOpen(MfSdkIccSlot\_E iSlotType)** |
| --- | --- | --- |
| **Function** | |  |
| **Params** | **in** | **iSlotType** Ref. **MfSdkIccSlot\_E** |
| **out** | **None** |
| **return** | | **Ref. MfSdkIccRet\_E** |
| **remark** | |  |
| **demo** | |  |

## MfSdkIccPowerOff

**LIB\_EXPORT s32 MfSdkIccPowerOff(s32 iCardType, MfSdkIccSlot\_E iSlotType)**

****

| **Function** | |  |
| --- | --- | --- |
| **Params** | **in** | **iCardType** Ref. **MfSdkIccType\_E** |
| **iSlotType** Ref. **MfSdkIccSlot\_E** |
| **out** | **None** |
| **return** | | **Ref. MfSdkIccRet\_E** |
| **remark** | |  |
| **demo** | |  |

## MfSdkIccPowerUp

| **Prototype** | | **LIB\_EXPORT s32 MfSdkIccPowerUp(s32 iCardType, MfSdkIccSlot\_E iSlotType ,u8 \*atrstr,s32 buflen)** |
| --- | --- | --- |
| **Function** | |  |
| **Params** | **in** | **iCardType** Ref. **MfSdkIccType\_E** |
| **iSlotType** Ref. **MfSdkIccSlot\_E** |
| **buflen** |
| **out** | **atrstr** |
| **return** | | **Ref. MfSdkIccRet\_E** |
| **remark** | |  |
| **demo** | |  |

## MfSdkIccPowerOn

| **Prototype** | | **LIB\_EXPORT s32 MfSdkIccPowerOn(s32 iCardType, MfSdkIccSlot\_E iSlotType)** |
| --- | --- | --- |
| **Function** | |  |
| **Params** | **in** | **iCardType** Ref. **MfSdkIccType\_E** |
| **iSlotType** Ref. **MfSdkIccSlot\_E** |
| **out** | **None** |
| **return** | | **Ref. MfSdkIccRet\_E** |
| **remark** | |  |
| **demo** | |  |

# GUI module

## Module description

This module mainly includes UI APIs.

## Module structure declaration

//MfSdkGuiInputPage Parameters typedef struct

{

s8\* title; //[in] Title displayed in the middle of the first line s8\* msgPrompt; //[in] Prompt information

s8\* str; //[out] input data string

s32 disp\_mode; //[in] 0:digit input, 1:character input, 2:password input s32 disp\_line; //[in] the number of rows Prompt displayed

s32 input\_line; //[in] the number of rows inputs displayed

s32 disp\_pattern; //[in] Prompt display position, 0 left alignment, 1 center; 2 right alignment

s32 input\_pattern; //[in] inputs display position, 0 left alignment, 1 center; 2 right alignment

s32 min; //[in] minimum input length s32 max; //[in] maximum input length

s32 timeout; //[in] waiting for input timeout time (seconds)

void (\*extra\_paint\_func)(void \*extra\_param); //[in] extra painting function on input page

void\* extra\_param; //[in] parameter of extra\_paint\_func

}MfSdkGuinputPageParam\_T;

## Constant declarations



## MfSdkGuiLedAmount

| **Prototype** | | **LIB\_EXPORT void MfSdkGuiLedAmount(char \*msg)** |
| --- | --- | --- |
| **Function** | | Set Amount display |
| **Params** | **in** | **msg** text amount eg. 0.00/1.00 |
| **out** | **None** |



| **return** | **Nothing** |
| --- | --- |
| **remark** |  |
| **demo** | MfSdkGuiLedAmount("0.00"); //show amount on code break screen |

## MfSdkGuiLedCounter

| **Prototype** | | **LIB\_EXPORT void MfSdkGuiLedCounter(char \*msg)** |
| --- | --- | --- |
| **Function** | | Show Counter display |
| **Params** | **in** | **msg** 3 bytes ascii digit eg. "0"-"999" |
| **out** | **None** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | | MfSdkGuiLedCounter(“0”); |

## MfSdkGuiLedTime

| **Prototype** | | **LIB\_EXPORT void MfSdkGuiLedTime(char \*msg)** |
| --- | --- | --- |
| **Function** | |  |
| **Params** | **in** | **msg** HHMM eg. "09:00" |
| **out** | **None** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | | MfSdkGuiLedTime(“09:00”); |

## MfSdkGuiLedDigitShow

| **Prototype** | **LIB\_EXPORT void MfSdkGuiLedDigitShow(char \*digit\_str)** |
| --- | --- |
| **Function** | Segment code |



| **Params** | **in** | **digit\_str** |
| --- | --- | --- |
| **out** | **None** |
| **return** | | **MFSDK\_RET\_OK Success** |
| **remark** | |  |
| **demo** | |  |

## MfSdkGuiBeginBatchPaint

| **Prototype** | | **LIB\_EXPORT void MfSdkGuiBeginBatchPaint()** |
| --- | --- | --- |
| **Function** | | Batch refresh starts |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | | MfSdkGuiBeginBatchPaint(); MfSdkGuiClearDc();  //painting APIs  MfSdkGuiEndBatchPaint(); |

## MfSdkGuiEndBatchPaint

| **Prototype** | | **LIB\_EXPORT void MfSdkGuiEndBatchPaint()** |
| --- | --- | --- |
| **Function** | | End of batch refresh |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | | MfSdkGuiBeginBatchPaint(); MfSdkGuiClearDc();  //painting APIs  MfSdkGuiEndBatchPaint(); |



## MfSdkGuiSetColor

| **Prototype** | | **LIB\_EXPORT void MfSdkGuiSetColor(s32 color)** |
| --- | --- | --- |
| **Function** | | Set the foreground color |
| **Params** | **in** | **color** the foreground color |
| **out** | **None** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | |  |

## MfSdkGuiGetColor

| **Prototype** | | **LIB\_EXPORT s32 MfSdkGuiGetColor()** |
| --- | --- | --- |
| **Function** | | Get the foreground color |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **>= 0 success, the foreground color** |
| **Other failure** |
| **remark** | |  |
| **demo** | | s32 foreground\_color = MfSdkGuiGetColor(); |

## MfSdkGuiSetBgColor

| **Prototype** | | **LIB\_EXPORT void MfSdkGuiSetBgColor(s32 color)** |
| --- | --- | --- |
| **Function** | | Set the background color |
| **Params** | **in** | **color** the background color |
| **out** | **None** |
| **return** | | **Nothing** |



| **remark** |  |
| --- | --- |
| **demo** |  |

## MfSdkGuiGetBgColor

| **Prototype** | | **LIB\_EXPORT s32 MfSdkGuiGetBgColor()** |
| --- | --- | --- |
| **Function** | | Get the background color |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **>= 0 success, the background color** |
| **Other failure** |
| **remark** | |  |
| **demo** | | s32 background\_color = MfSdkGuiGetBgColor(); |

## MfSdkGuiSetFullScreen

| **Prototype** | | **LIB\_EXPORT void MfSdkGuiSetFullScreen(s32 full)** |
| --- | --- | --- |
| **Function** | | Set to full screen display |
| **Params** | **in** | **full** is full screen display(1 - Yes, 0 - No) |
| **out** | **None** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | | MfSdkGuiSetFullScreen(1); //set full screen |

## MfSdkGuiPixel

| **Prototype** | **LIB\_EXPORT s32 MfSdkGuiPixel(s32 x, s32 y)** |
| --- | --- |
| **Function** | Draw a point |



| **Params** | **in** | **x** | horizontal coordinate |
| --- | --- | --- | --- |
| **y** | vertical coordinate |
| **out** | **None** | |
| **return** | | **0** | **success** |
| **Other** | **failure** |
| **remark** | |  | |
| **demo** | | s32 ret;  MfSdkGuiBeginBatchPaint(); MfSdkGuiClearDc();  ret = MfSdkGuiPixel(5 , 5); //draw a point at (5, 5)  MfSdkGuiEndBatchPaint(); | |

## MfSdkGuiLineTo

| **Prototype** | | **LIB\_EXPORT s32 MfSdkGuiLineTo(s32 x, s32 y)** | |
| --- | --- | --- | --- |
| **Function** | | Draw a line | |
| **Params** | **in** | **x** | horizontal coordinate |
| **y** | vertical coordinate |
| **out** | **None** | |
| **return** | | **0** | **success** |
| **Other** | **failure** |
| **remark** | |  | |
| **demo** | |  | |

## MfSdkGuiBarRc

| **Prototype** | | **LIB\_EXPORT void MfSdkGuiBarRc(s32 left, s32 top, s32 right, s32 bottom)** | |
| --- | --- | --- | --- |
| **Function** | | Gui filled area | |
| **Params** | **in** | **left** | left border |
| **top** | upper boundary |
| **right** | right border |
| **bottom** | lower boundary |



|  | **out** | **None** |
| --- | --- | --- |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | | MfSdkGuiBeginBatchPaint(); MfSdkGuiClearDc(); MfSdkGuiBarRc(5, 5, 10, 10);  MfSdkGuiEndBatchPaint(); |

## MfSdkGuiSetBarColor

| **Prototype** | | **LIB\_EXPORT void MfSdkGuiSetBarColor(s32 color)** |
| --- | --- | --- |
| **Function** | | Set the fill color |
| **Params** | **in** | **color** fill color(format 0x00RRGGBB) |
| **out** | **None** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | |  |

## MfSdkGuiGetBarColor

| **Prototype** | | **LIB\_EXPORT s32 MfSdkGuiGetBarColor()** |
| --- | --- | --- |
| **Function** | | Get the fill color |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **>=0 success, the fill color** |
| **Other failure** |
| **remark** | |  |
| **demo** | | s32 barColor = MfSdkGuiGetBarColor(); |



## MfSdkGuiSetFont

| **Prototype** | | **LIB\_EXPORT void MfSdkGuiSetFont(s32 font)** |
| --- | --- | --- |
| **Function** | | Set display font |
| **Params** | **in** | **font** font(0: 12 lattice, 1: 16 lattice) |
| **out** | **None** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | | MfSdkGuiSetFont(0); //set to 12 lattice font |

## MfSdkGuiGetFont

| **Prototype** | | **LIB\_EXPORT s32 MfSdkGuiGetFont()** |
| --- | --- | --- |
| **Function** | | Get display font |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **>=0 success, the display font** |
| **Other failure** |
| **remark** | |  |
| **demo** | | s32 font = MfSdkGuiGetFont(); |

## MfSdkGuiSetTextColor

| **Prototype** | | **LIB\_EXPORT void MfSdkGuiSetTextColor(s32 color)** |
| --- | --- | --- |
| **Function** | | Set text color |
| **Params** | **in** | **color** text color |
| **out** | **None** |
| **return** | | **Nothing** |



| **remark** |  |
| --- | --- |
| **demo** |  |

## MfSdkGuiGetTextColor

| **Prototype** | | **LIB\_EXPORT s32 MfSdkGuiGetTextColor()** |
| --- | --- | --- |
| **Function** | | Get text color |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **>=0 success, the text color** |
| **Other failure** |
| **remark** | |  |
| **demo** | | s32 textColor = MfSdkGuiGetTextColor(); |

## MfSdkGuiSetTextBgColor

| **Prototype** | | **LIB\_EXPORT void MfSdkGuiSetTextBgColor(s32 color)** |
| --- | --- | --- |
| **Function** | | Set the text background color |
| **Params** | **in** | **color** text background color |
| **out** | **None** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | |  |

## MfSdkGuiGetTextBgColor

| **Prototype** | **LIB\_EXPORT s32 MfSdkGuiGetTextBgColor()** |
| --- | --- |
| **Function** | Get the text background color |



| **Params** | **in** | **None** |
| --- | --- | --- |
| **out** | **None** |
| **return** | | **>=0 success, the text background color** |
| **Other failure** |
| **remark** | |  |
| **demo** | | s32 textBgColor = MfSdkGuiGetTextBgColor(); |

## MfSdkGuiClearDc

| **Prototype** | | **LIB\_EXPORT void MfSdkGuiClearDc()** |
| --- | --- | --- |
| **Function** | | Clear screen display |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | | MfSdkGuiBeginBatchPaint(); MfSdkGuiClearDc();  //painting APIs  MfSdkGuiEndBatchPaint(); |

## MfSdkGuiSetTextZoom

| **Prototype** | | **LIB\_EXPORT void MfSdkGuiSetTextZoom(s32 size)** |
| --- | --- | --- |
| **Function** | | Set text magnification |
| **Params** | **in** | **size** multiple, the default value is 2 |
| **out** | **None** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | | MfSdkGuiSetTextZoom(4); |



## MfSdkGuiGetTextZoom

| **Prototype** | | **LIB\_EXPORT s32 MfSdkGuiGetTextZoom()** |
| --- | --- | --- |
| **Function** | | Get text magnification |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **>=0 success, the text magnification** |
| **Other failure** |
| **remark** | |  |
| **demo** | | s32 textZoom = MfSdkGuiGetTextZoom(); |

## MfSdkGuiSetPixel

| **Prototype** | | **LIB\_EXPORT s32 MfSdkGuiSetPixel(s32 x, s32 y, s32 color)** |
| --- | --- | --- |
| **Function** | | Draw a point with color |
| **Params** | **in** | **x** horizontal coordinate |
| **y** vertical coordinate |
| **color** point color |
| **out** | **None** |
| **return** | | **0 success** |
| **Other failure** |
| **remark** | |  |
| **demo** | |  |

## MfSdkGuiGetPixel

| **Prototype** | **LIB\_EXPORT s32 MfSdkGuiGetPixel(s32 x, s32 y)** |
| --- | --- |
| **Function** | Get the color of the point on the screen |



| **Params** | **in** | **x** | horizontal coordinate |
| --- | --- | --- | --- |
| **y** | vertical coordinate |
| **out** | **None** | |
| **return** | | **>=0** | **success, point color** |
| **Other** | **failure** |
| **remark** | |  | |
| **demo** | | s32 pixelColor = MfSdkGuiGetPixel(5, 5); //get the color of point (5, 5) | |

## MfSdkGuiTextOut

| **Prototype** | | **LIB\_EXPORT s32 MfSdkGuiTextOut(s32 x, s32 y, char \* text)** | |
| --- | --- | --- | --- |
| **Function** | | Display text on the screen ,Show only English | |
| **Params** | **in** | **x** | horizontal coordinate |
| **y** | vertical coordinate |
| **text** | text content |
| **out** | **None** | |
| **return** | | **0** | **success** |
| **Other** | **failure** |
| **remark** | |  | |
| **demo** | | s32 ret = -1; MfSdkGuiBeginBatchPaint(); MfSdkGuiClearDc();  ret = MfSdkGuiTextOut(5, 5, “morefun”);  MfSdkGuiEndBatchPaint(); | |

## MfSdkGuiGetTextWidth

| **Prototype** | | **LIB\_EXPORT s32 MfSdkGuiGetTextWidth(char \*text)** |
| --- | --- | --- |
| **Function** | | Get the display width of the text |
| **Params** | **in** | **text** text content |
| **out** | **None** |
| **return** | | **>=0 success, the text width** |



|  | **Other failure** |
| --- | --- |
| **remark** |  |
| **demo** | s32 textWidth = MfSdkGuiGetTextWidth(“morefun”); |

## MfSdkGuiGetTextHeight

| **Prototype** | | **LIB\_EXPORT s32 MfSdkGuiGetTextHeight(char \*text)** |
| --- | --- | --- |
| **Function** | | Get the display height of the text |
| **Params** | **in** | **text** text content |
| **out** | **None** |
| **return** | | **>=0 success, the text height** |
| **Other failure** |
| **remark** | |  |
| **demo** | | s32 textHeight = MfSdkGuiGetTextHeight(“morefun”); |

## MfSdkGuiCline

| **Prototype** | | **LIB\_EXPORT void MfSdkGuiCline(s32 x1, s32 y1, s32 x2, s32 y2, s32 color)** |
| --- | --- | --- |
| **Function** | | Draw a line |
| **Params** | **in** | **x1** point1 X coordinate |
| **x2** point2 X coordinate |
| **y1** point1 Y coordinate |
| **y2** point2 Y coordinate |
| **color** color of the line |
| **out** | **None** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | |  |



## MfSdkGuiGetWidth

| **Prototype** | | **LIB\_EXPORT s32 MfSdkGuiGetWidth(void)** |
| --- | --- | --- |
| **Function** | | Get screen width |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **>=0 success, the screen width** |
| **Other failure** |
| **remark** | |  |
| **demo** | | s32 width = MfSdkGuiGetWidth(); |

## MfSdkGuiGetHeight

| **Prototype** | | **LIB\_EXPORT s32 MfSdkGuiGetHeight(void)** |
| --- | --- | --- |
| **Function** | | Get screen height |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **>=0 success, the screen height** |
| **Other failure** |
| **remark** | |  |
| **demo** | | s32 height = MfSdkGuiGetHeight(); |

## MfSdkGuiPageOpPaint

| **Prototype** | | **LIB\_EXPORT void MfSdkGuiPageOpPaint(char \* left\_str, char \* right\_str)** |
| --- | --- | --- |
| **Function** | | Bottom options painting |
| **Params** | **in** | **left\_str** The character displayed in the lower left corner |
| **right\_str** The character displayed in the lower right corner |



|  | **out** | **None** |
| --- | --- | --- |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | | MfSdkGuiBeginBatchPaint(); MfSdkGuiClearDc(); MfSdkGuiPageOpPaint(“Cancel”, “OK”);  MfSdkGuiEndBatchPaint(); |

## MfSdkGuiImeSetMode

| **Prototype** | | **LIB\_EXPORT s32 MfSdkGuiImeSetMode(s32 def\_mode, s32 allow\_mode, s32 password)** |
| --- | --- | --- |
| **Function** | | Set input method parameters |
| **Params** | **in** | **def\_mode** Default input method |
| **allow\_mode** Support input method |
| **password** enter password |
| **out** | **None** |
| **return** | | **0 success** |
| **Other failure** |
| **remark** | |  |
| **demo** | |  |

## MfSdkGuiImeStartInput

| **Prototype** | | **LIB\_EXPORT s32 MfSdkGuiImeStartInput(char \* buffer, s32 max, char \* help)** |
| --- | --- | --- |
| **Function** | | Open the input method page |
| **Params** | **in** | **buffer** Input buffer |
| **max** Maximum input character |
| **help** Enter page title |
| **out** | **None** |
| **return** | | **0 success, the text length** |
| **Other failure** |
| **remark** | | **ONLY for H9G,H9L** |



**demo**

## MfSdkGuiMainMenuFuncAdd

| **Prototype** | | **LIB\_EXPORT s32 MfSdkGuiMainMenuFuncAdd(void \* pfunc)** |
| --- | --- | --- |
| **Function** | | Add menu handler |
| **Params** | **in** | **pfunc** menu handler |
| **out** | **None** |
| **return** | | **0 success** |
| **Other failure** |
| **remark** | |  |
| **demo** | |  |

## MfSdkGuiMainMenuFuncDel

| **Prototype** | | **LIB\_EXPORT s32 MfSdkGuiMainMenuFuncDel(void \* pfunc)** |
| --- | --- | --- |
| **Function** | | Del menu handler |
| **Params** | **in** | **pfunc** menu handler |
| **out** | **None** |
| **return** | | **0 success** |
| **Other failure** |
| **remark** | |  |
| **demo** | |  |

## MfSdkGuiMainMenuShow

| **Prototype** | **LIB\_EXPORT void MfSdkGuiMainMenuShow(char \*id , s32 timeover)** |
| --- | --- |
| **Function** | Display menu |



| **Params** | **in** | **id** menu id |
| --- | --- | --- |
| **timeover** overtime time |
| **out** | **None** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | |  |

## MfSdkGuiPostMessage

| **Prototype** | | **LIB\_EXPORT u32 MfSdkGuiPostMessage(u32 msg\_id, u32 wparam, u32 lparam)** | |
| --- | --- | --- | --- |
| **Function** | | Send a message | |
| **Params** | **in** | **msg\_id** | message id |
| **wparam** | parameter 1 |
| **lparam** | parameter 2 |
| **out** | **None** | |
| **return** | | **0** | **success** |
| **Other** | **failure** |
| **remark** | |  | |
| **demo** | | MfSdkGuiPostMessage(MFSDK\_GUI\_GUIPAINT, 0, 0); | |

## MfSdkGuiGetMessage

| **Prototype** | | **LIB\_EXPORT u32 MfSdkGuiGetMessage( MfSdkGuiMsg\_T \* pmsg , s32 timeover)** | |
| --- | --- | --- | --- |
| **Function** | | Recv a message | |
| **Params** | **in** | **pmsg** | message structure |
| **timeover** | overtime time |
| **out** | **None** | |
| **return** | | **0** | **success** |
| **Other** | **failure** |
| **remark** | |  | |
| **demo** | | MfSdkGuiMsg\_T pmsg;  if(MfSdkGuiGetMessage(&pmsg, 100) == 0) | |



{

//TODO

}

## MfSdkGuiProcDefaultMsg

| **Prototype** | | **LIB\_EXPORT s32 MfSdkGuiProcDefaultMsg( MfSdkGuiMsg\_T \* pmsg )** |
| --- | --- | --- |
| **Function** | | Let the system process the default message |
| **Params** | **in** | **pmsg** message structure |
| **out** | **None** |
| **return** | | **0 success** |
| **Other failure** |
| **remark** | |  |
| **demo** | |  |

## MfSdkGuiMessageBoxShow

| **Prototype** | | **LIB\_EXPORT s32 MfSdkGuiMessageBoxShow(char \*title, char**  **\*msg , char\* pszLeftOp, char\* pszRightOp , s32 timeover)** | |
| --- | --- | --- | --- |
| **Function** | | Display dialog | |
| **Params** | **in** | **title** | message title |
| **msg** | message content |
| **pszLeftOp** | bottom left corner |
| **pszRightOp** | tip in the lower right corner |
| **timeover** | overtime time |
| **out** | **None** | |
| **return** | | **1** | **Confirm return** |
| **2** | **Cancel back** |
| **3** | **Timeout** |
| **Other** | **failure** |
| **remark** | |  | |
| **demo** | | int ret = -1;  ret = MfSdkGuiMessageBoxShow(“Tips”, “Success” , “Cancel”, “Confirm” , “30 \* 1000”); | |



## MfSdkGuiLoadBmpEx

| **Prototype** | | **LIB\_EXPORT char \*MfSdkGuiLoadBmpEx(char \* filename, int**  **\*width, int \*height, int \* color)** | |
| --- | --- | --- | --- |
| **Function** | | Load bmp into memory | |
| **Params** | **in** | **filename** | image name |
| **out** | **width** | image width |
| **height** | image height |
| **color** | image color |
| **return** | | **!=0** | **success,Image content array, which needs to**  **be released after use** |
| **0** | **failure** |
| **remark** | |  | |
| **demo** | | #define LOGOBMP “data\\logo.bmp”  char \*pbmp = NULL; int logowidth = 0;  int logoheight = 0; int logocolor = 0;  pbmp = MfSdkGuiLoadBmpEx(LOGOBMP, &logowidth, &logoheight, &logocolor); | |

## MfSdkGuiLoadBmp

| **Prototype** | | **LIB\_EXPORT char \*MfSdkGuiLoadBmp(char \* filename, s32**  **\*width, s32 \*height)** | |
| --- | --- | --- | --- |
| **Function** | | Load bmp into memory | |
| **Params** | **in** | **filename** | image name |
| **out** | **width** | image width |
| **height** | image height |
| **return** | | **!=0** | **success,Image content array, which needs to**  **be released after use** |
| **0** | **failure** |
| **remark** | |  | |

#define LOGOBMP “data\\logo.bmp”

**demo**

char \*pbmp = NULL; int logowidth = 0;

int logoheight = 0;

pbmp = MfSdkGuiLoadBmp(LOGOBMP, &logowidth, &logoheight);//bmp only black and white

## MfSdkGuiBmpFree

| **Prototype** | | **LIB\_EXPORT void MfSdkGuiBmpFree(char \* pbmp)** |
| --- | --- | --- |
| **Function** | | Free memory of bmp |
| **Params** | **in** | **pbmp** image content array |
| **out** | **None** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | | MfSdkGuiBmpFree(pbmp); |

## MfSdkGuiOutBits

| **Prototype** | | **LIB\_EXPORT void MfSdkGuiOutBits(s32 x, s32 y, unsigned char**  **\*pbits, s32 width , s32 height, s32 mode)** | |
| --- | --- | --- | --- |
| **Function** | | Display image | |
| **Params** | **in** | **x** | horizontal coordinate |
| **y** | vertical coordinate |
| **pbits** | image data |
| **width** | image width |
| **height** | image height |
| **mode** | display mode(0 - Positive, 1 - Reverse) |
| **out** | **None** | |
| **return** | | **Nothing** | |
| **remark** | |  | |
| **demo** | | #define LOGOBMP “data\\logo.bmp” | |



s8 \*pbmp = NULL; s32 logowidth = 0; s32 logoheight = 0;

pbmp = MfSdkGuiLoadBmp(LOGOBMP, &logowidth, &logoheight); if(pbmp)

{

s32 logotop = (MfSdkGuiGetHeight() - logoheight)/2; s32 logoleft = (MfSdkGuiGetWidth() - logowidth)/2; MfSdkGuiBeginBatchPaint();

MfSdkGuiClearDc();

MfSdkGuiOutBits(logoleft, logotop , pbmp, logowidth , logoheight, 0); //show logo at the center of the screen

MfSdkGuiEndBatchPaint();

}

## MfSdkGuiOutBitsEx

| **Prototype** | | **LIB\_EXPORT void MfSdkGuiOutBitsEx(s32 x, s32 y, unsigned char \*pbits, s32 width , s32 height, s32 mode , s32 color)** | |
| --- | --- | --- | --- |
| **Function** | | Display image | |
| **Params** | **in** | **x** | horizontal coordinate |
| **y** | vertical coordinate |
| **pbits** | image data |
| **width** | image width |
| **height** | image height |
| **mode** | display mode(0 - Positive, 1 - Reverse) |
| **color** | bit color of the picture(1,4,24) |
| **out** | **None** | |
| **return** | | **Nothing** | |
| **remark** | |  | |
| **demo** | | #define LOGOBMP “data\\logo.bmp”  s8 \*pbmp = NULL; s32 logowidth = 0; s32 logoheight = 0; s32 logocolor = 0; | |



pbmp = MfSdkGuiLoadBmpEx(LOGOBMP, &logowidth, &logoheight, &logocolor); if(pbmp)

{

s32 logotop = (MfSdkGuiGetHeight() - logoheight)/2; s32 logoleft = (MfSdkGuiGetWidth() - logowidth)/2; MfSdkGuiBeginBatchPaint();

MfSdkGuiClearDc();

MfSdkGuiOutBitsEx(logoleft, logotop , pbmp, logowidth , logoheight, 0, logocolor); //show logo at the center of the screen MfSdkGuiEndBatchPaint();

}

## MfSdkGuiOutBitsZoom

| **Prototype** | | **LIB\_EXPORT void MfSdkGuiOutBitsZoom(s32 x, s32 y, unsigned char \*pbits, s32 width , s32 height, s32 mode , s32 zoom)** | |
| --- | --- | --- | --- |
| **Function** | | Display image | |
| **Params** | **in** | **x** | horizontal coordinate |
| **y** | vertical coordinate |
| **pbits** | image data |
| **width** | image width |
| **height** | image height |
| **mode** | display mode(0 - Positive, 1 - Reverse) |
| **zoom** | amplification factor |
| **out** | **None** | |
| **return** | | **Nothing** | |
| **remark** | |  | |
| **demo** | |  | |

## MfSdkGuiTextWidthEx

| **Prototype** | | **LIB\_EXPORT s32 MfSdkGuiTextWidthEx(char \* str)** |
| --- | --- | --- |
| **Function** | | get text width |
| **Params** | **in** | **str** text content |



|  | **out** | **None** |
| --- | --- | --- |
| **return** | | **>=0 success, the text width** |
| **Other failure** |
| **remark** | |  |
| **demo** | |  |

## MfSdkGuiTextOutEx

| **Prototype** | | **LIB\_EXPORT s32 MfSdkGuiTextOutEx(s32 x, s32 y,char \* str)** |
| --- | --- | --- |
| **Function** | | Display text on the screen ,Show different languages |
| **Params** | **in** | **x** horizontal coordinate |
| **y** vertical coordinate |
| **str** text content |
| **out** | **None** |
| **return** | | **0 success** |
| **Other failure** |
| **remark** | |  |
| **demo** | | s32 ret = -1; MfSdkGuiBeginBatchPaint(); MfSdkGuiClearDc();  ret = MfSdkGuiTextOutEx(5, 5, “morefun”);  MfSdkGuiEndBatchPaint(); |

## MfSdkGuiSetTextStyle

| **Prototype** | | **LIB\_EXPORT void MfSdkGuiSetTextStyle(s32 textStyle)** |
| --- | --- | --- |
| **Function** | | Setting Text Style |
| **Params** | **in** | **textStyle** text style(0 - opaque, 1 - transparent) |
| **out** | **None** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | | MfSdkGuiSetTextStyle(0); |



## MfSdkGuiSelectPageEx

| **Prototype** | | **LIB\_EXPORT s32 MfSdkGuiSelectPageEx(char \*title , char**  **\*items[],s32 itemscount,s32 timeover, s32 select)** |
| --- | --- | --- |
| **Function** | | select page |
| **Params** | **in** | **title** the title of the select page |
| **items** menu items |
| **itemscount** number of menu items |
| **timeover** menu timeout |
| **select** default menu item |
| **out** | **None** |
| **return** | | **>=0 success, index in items of chosen item** |
| **Other failure** |
| **remark** | |  |
| **demo** | |  |

## MfSdkGuiSelectPageExT

| **Prototype** | | **LIB\_EXPORT s32 MfSdkGuiSelectPageExT(char \*title ,char items[][255],s32 itemscount,s32 timeover, s32 select)** |
| --- | --- | --- |
| **Function** | | select page |
| **Params** | **in** | **title** the title of the select page |
| **items** menu items |
| **itemscount** number of menu items |
| **timeover** menu timeout |
| **select** default menu item |
| **out** | **None** |
| **return** | | **>=0 success, index in items of chosen item** |
| **Other failure** |
| **remark** | |  |
| **demo** | |  |



## MfSdkGuiSelectPageCb

| **Prototype** | | **LIB\_EXPORT s32 MfSdkGuiSelectPageCb(char \*title , char**  **\*items[],int itemscount,int timeover,int select,char callback\_key,int (\*callbackfunc)(const void \*items[],const int itemscount))** |
| --- | --- | --- |
| **Function** | | select page with key trigger callback function |
| **Params** | **in** | **title** the title of the select page |
| **items** menu items |
| **itemscount** number of menu items |
| **timeover** menu timeout |
| **select** default menu item |
| the key used to trigger the callback (recommend  **callback\_key**  KEY\_0 or KEY\_BACKSPACE) |
| **callbackfunc** callback function |
| **out** | **None** |
| **return** | | **>=0 success, index in items of chosen item** |
| **Other failure** |
| **remark** | |  |
| **demo** | |  |

## MfSdkGuiTitleColorBackground

| **Prototype** | | **LIB\_EXPORT void MfSdkGuiTitleColorBackground(s32 color)** |
| --- | --- | --- |
| **Function** | | set the background color of title |
| **Params** | **in** | **color** the background color of title |
| **out** | **None** |
| **return** | | **Nothing** |
| **remark** | | **only for H9G, H9L** |
| **demo** | |  |



## MfSdkGuiTitleColorForeground

| **Prototype** | | **LIB\_EXPORT void MfSdkGuiTitleColorForeground(s32 color)** |
| --- | --- | --- |
| **Function** | | set the foreground color of title |
| **Params** | **in** | **color** the foreground color of title |
| **out** | **None** |
| **return** | | **Nothing** |
| **remark** | | **only for H9G, H9L** |
| **demo** | |  |

## MfSdkGuiTitleFont

| **Prototype** | | **LIB\_EXPORT void MfSdkGuiTitleFont(int font)** |
| --- | --- | --- |
| **Function** | | Set title font |
| **Params** | **in** | **font** font(0 - 12 lattice, 1 - 16 lattice) |
| **out** | **None** |
| **return** | | **Nothing** |
| **remark** | | **only for H9G, H9L** |
| **demo** | |  |

## MfSdkGuiMenuHightlineColor

| **Prototype** | | **LIB\_EXPORT void MfSdkGuiMenuHightlineColor(s32 color)** |
| --- | --- | --- |
| **Function** | | set the color of menu hightline |
| **Params** | **in** | **color** the color of menu hightline |
| **out** | **None** |
| **return** | | **Nothing** |
| **remark** | | **only for H9G, H9L** |



**demo**

## MfSdkGuiTextOutLineCenter

| **Prototype** | | **LIB\_EXPORT void MfSdkGuiTextOutLineCenter(char \*pMsg , s32 top)** |
| --- | --- | --- |
| **Function** | | Display text in the line center on the screen,Show only English |
| **Params** | **in** | **pMsg** text content |
| **top** top coordinate |
| **out** | **None** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | | MfSdkGuiBeginBatchPaint(); MfSdkGuiClearDc(); //clean screen  //show “Morefun” at the center of line 5 on screen MfSdkGuiTextOutLineCenter(“Morefun”, MFSDK\_GUI\_LINE\_TOP(5));  MfSdkGuiBeginEndPaint(); |

## MfSdkGuiTextOutLineRight

| **Prototype** | | **LIB\_EXPORT void MfSdkGuiTextOutLineRight(char \*pMsg , s32 top)** |
| --- | --- | --- |
| **Function** | | Display text in the line right on the screen,Show only English |
| **Params** | **in** | **pMsg** text content |
| **top** top coordinate |
| **out** | **None** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | | MfSdkGuiBeginBatchPaint(); MfSdkGuiClearDc(); //clean screen  //show “Morefun” at the right of line 5 on screen MfSdkGuiTextOutLineRight(“Morefun”, MFSDK\_GUI\_LINE\_TOP(5));  MfSdkGuiBeginEndPaint(); |



## MfSdkGuiTextOutLineLeft

| **Prototype** | | **LIB\_EXPORT void MfSdkGuiTextOutLineLeft(char \*pMsg , s32 top)** |
| --- | --- | --- |
| **Function** | | Display text in the line left on the screen,Show only English |
| **Params** | **in** | **pMsg** text content |
| **top** top coordinate |
| **out** | **None** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | | MfSdkGuiBeginBatchPaint(); MfSdkGuiClearDc(); //clean screen  //show “Morefun” at the left of line 5 on screen MfSdkGuiTextOutLineLeft(“Morefun”, MFSDK\_GUI\_LINE\_TOP(5));  MfSdkGuiBeginEndPaint(); |

## MfSdkGuiTextOutWinCenter

| **Prototype** | | **LIB\_EXPORT void MfSdkGuiTextOutWinCenter(char \*pmsg)** |
| --- | --- | --- |
| **Function** | | Display text in the middle of the screen,Show only English |
| **Params** | **in** | **pmsg** text content |
| **out** | **None** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | | MfSdkGuiBeginBatchPaint(); MfSdkGuiClearDc(); //clean screen  MfSdkGuiTextOutWinCenter(“Morefun”); //show “Morefun” at the center of the screen  MfSdkGuiBeginEndPaint(); |



## MfSdkGuiClearRect

| **Prototype** | | **LIB\_EXPORT void MfSdkGuiClearRect(s32 left, s32 top, s32 right, s32 bottom, s32 color)** |
| --- | --- | --- |
| **Function** | | Refresh the specified area |
| **Params** | **in** | **left** left border |
| **top** upper boundary |
| **right** right border |
| **bottom** lower boundary |
| **color** refresh with specified color |
| **out** | **None** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | |  |

## MfSdkGuiMessageBoxShowEx

| **Prototype** | | **LIB\_EXPORT s32 MfSdkGuiMessageBoxShowEx(char \*title, char**  **\*msg , char\* pszLeftOp, char\* pszRightOp , s32 timeover,s32 flag,unsigned char \* keylist, s32 \* presskey)** |
| --- | --- | --- |
| **Function** | | Display dialog |
| **Params** | **in** | **title** message title |
| **msg** message content |
| **pszLeftOp** bottom left corner |
| **pszRightOp** tip in the lower right corner |
| **timeover** overtime time |
| **flag** show timeout on title right side(1 - Yes, 0 - No) |
| **support** key value eg:{key0,key1,keyok} |
| **out** | **presskey** return press key value |
| **return** | | **1 Confirm return** |
| **2 Cancel back** |
| **3 Timeout** |
| **Other Failure** |
| **remark** | |  |



**demo**

## MfSdkGuiSetTitle

| **Prototype** | | **LIB\_EXPORT void MfSdkGuiSetTitle(char \*title)** |
| --- | --- | --- |
| **Function** | | Set title |
| **Params** | **in** | **title** title string |
| **out** | **None** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | |  |

## MfSdkGuiTextOutHighlight

| **Prototype** | | **LIB\_EXPORT s32 MfSdkGuiTextOutHighlight(s32 x, s32 y, s32 y2,char \* text)** |
| --- | --- | --- |
| **Function** | | Highlight a line of text |
| **Params** | **in** | **x** the starting x coordinate of the highlight |
| **y** the starting y coordinate of the highlight |
| **y2** the ending y coordinate of the highlight |
| **text** the text want to highlight |
| **out** | **None** |
| **return** | | **0 success** |
| **Other failure** |
| **remark** | |  |
| **demo** | |  |

## MfSdkGuiRectHighlight

| **Prototype** | **LIB\_EXPORT s32 MfSdkGuiRectHighlight(int x, int x2, int y, int y2)** |
| --- | --- |



| **Function** | | Highlight an area | |
| --- | --- | --- | --- |
| **Params** | **in** | **x** | the starting x coordinate of the highlight |
| **x2** | the ending x coordinate of the highlight |
| **y** | the starting y coordinate of the highlight |
| **y2** | the ending y coordinate of the highlight |
| **out** | **None** | |
| **return** | | **0** | **success** |
| **Other** | **failure** |
| **remark** | |  | |
| **demo** | |  | |

## MfSdkGuiDefaultMsgFuncAdd

| **Prototype** | | **LIB\_EXPORT s32 MfSdkGuiDefaultMsgFuncAdd(void \* pfunc)** |
| --- | --- | --- |
| **Function** | | Add default message processing |
| **Params** | **in** | **pfunc** message processing function |
| **out** | **None** |
| **return** | | **0 success** |
| **Other failure** |
| **remark** | |  |
| **demo** | |  |

## MfSdkGuiMainMenuItemAdd

| **Prototype** | | **LIB\_EXPORT s32**  **MfSdkGuiMainMenuItemAdd(MfSdkGuiMenuItemDef\_T \* menu\_item)** |
| --- | --- | --- |
| **Function** | | Add menu item |
| **Params** | **in** | **menu\_item** menu item |
| **out** | **None** |
| **return** | | **0 success** |
| **Other failure** |



| **remark** |  |
| --- | --- |
| **demo** |  |

## MfSdkGuiMainMenuItemDel

| **Prototype** | | **LIB\_EXPORT s32 MfSdkGuiMainMenuItemDel(char \*name ,char**  **\*id)** |
| --- | --- | --- |
| **Function** | | Delete menu item |
| **Params** | **in** | **name** the value of st\_gui\_menu\_item\_def->name |
| **id** the value of st\_gui\_menu\_item\_def->id |
| **out** | **None** |
| **return** | | **0 success** |
| **Other failure** |
| **remark** | |  |
| **demo** | |  |

## MfSdkGuiKeyGetEvent

| **Prototype** | | **LIB\_EXPORT s32 MfSdkGuiKeyGetEvent(MfSdkGuiKeyEvent\_T**  **\*key\_event)** |
| --- | --- | --- |
| **Function** | | get event |
| **Params** | **in** | **None** |
| **out** | **key\_event** event structure |
| **return** | | **0 success** |
| **Other failure** |
| **remark** | |  |
| **demo** | |  |

## MfSdkGuiWifiSetPage

**Prototype LIB\_EXPORT s32 MfSdkGuiWifiSetPage()**

****

| **Function** | |  |
| --- | --- | --- |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **0 success** |
| **remark** | |  |
| **demo** | |  |

## MfSdkGuiSetProcs

| **Prototype** | | **LIB\_EXPORT void MfSdkGuiSetProcs( MfSdkGuiSignProcs\_T**  **\*procs )** |
| --- | --- | --- |
| **Function** | | Set sign processing function |
| **Params** | **in** | **procs** the sign processing function |
| **out** | **None** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | |  |

## MfSdkGuiSignExist

| **Prototype** | | **LIB\_EXPORT int MfSdkGuiSignExist( const char \*index )** |
| --- | --- | --- |
| **Function** | | Determine whether it has been signed |
| **Params** | **in** | **index** Signature index |
| **out** | **None** |
| **return** | | **0 success** |
| **other failure** |
| **remark** | |  |
| **demo** | |  |



## MfSdkGuiSignProc

| **Prototype** | | **LIB\_EXPORT int MfSdkGuiSignProc( const char \*index, const char \*date, const char \*refno, int timeout )** |
| --- | --- | --- |
| **Function** | | Signature processing |
| **Params** | **in** | **index** Signature index |
| **date** Device date |
| **refno** Reference No |
| **timeout** timeout time |
| **out** | **None** |
| **return** | | **0 success** |
| **Other failure** |
| **remark** | | **API: gui\_sign\_proc and gui\_sign\_proc2 merge to MfSdkGuiSignProc** |
| **demo** | |  |

## MfSdkGuiSignPrint

| **Prototype** | | **LIB\_EXPORT const char \*MfSdkGuiSignExist( const char \*index )** |
| --- | --- | --- |
| **Function** | | Get sign print string |
| **Params** | **in** | **index** Signature index |
| **out** | **None** |
| **return** | | **!=0 success, the print string** |
| **0 failure** |
| **remark** | |  |
| **demo** | |  |

## MfSdkGuiSignEncode

**LIB\_EXPORT int MfSdkGuiSignEncode( const char \*index, char**

**\*\*jbigencode )**

****

| **Function** | | JBIG encoding interface |
| --- | --- | --- |
| **Params** | **in** | **index** Signature index |
| **out** | **jbigencode** JBIG encoded signature data |
| **return** | | **!=0 success, the print string** |
| **0 failure** |
| **remark** | |  |
| **demo** | |  |

## MfSdkGuiSignEncodeFree

| **Prototype** | | **LIB\_EXPORT int MfSdkGuiSignEncodeFree( char \*jbigencode )** |
| --- | --- | --- |
| **Function** | | JBIG results released |
| **Params** | **in** | **jbigencode** the JBIG result |
| **out** | **None** |
| **return** | | **0 success** |
| **Other failure** |
| **remark** | |  |
| **demo** | |  |

## MfSdkGuiSignClean

| **Prototype** | | **LIB\_EXPORT int MfSdkGuiSignClean()** |
| --- | --- | --- |
| **Function** | | Clean up signature data |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **0 success** |
| **Other failure** |
| **remark** | |  |
| **demo** | |  |



## MfSdkGuiSignSavetofile

| **Prototype** | | **LIB\_EXPORT int MfSdkGuiSignSavetofile(char \*filename, const char \*index)** |
| --- | --- | --- |
| **Function** | | Save signature to file |
| **Params** | **in** | **filename** the filename of Signature |
| **index** Signature index |
| **out** | **None** |
| **return** | | **0 success** |
| **Other failure** |
| **remark** | |  |
| **demo** | |  |

## MfSdkGuiSetPowerfullColor

| **Prototype** | | **LIB\_EXPORT void MfSdkGuiSetPowerfullColor(unsigned int iColor)** |
| --- | --- | --- |
| **Function** | | set powerfull color |
| **Params** | **in** | 0xFF0000(red)/ 0x00FF00(green) /  **iColor**  0x0000FF(blue) |
| **out** | **None** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | |  |

## MfSdkGuiStateFuncAdd

| **Prototype** | | **LIB\_EXPORT int MfSdkGuiStateFuncAdd(void \* pfunc)** |
| --- | --- | --- |
| **Function** | |  |
| **Params** | **in** | **pfunc** |



|  | **out** | **None** |
| --- | --- | --- |
| **return** | | **0 success** |
| **remark** | |  |
| **demo** | |  |

## MfSdkGuiCleartState

| **Prototype** | | **LIB\_EXPORT void MfSdkGuiCleartState(int left , int width)** |
| --- | --- | --- |
| **Function** | |  |
| **Params** | **in** | **left** |
| **width** |
| **out** | **None** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | |  |

## MfSdkGuiOutStateBitsColor

| **Prototype** | | **LIB\_EXPORT void MfSdkGuiOutStateBitsColor(int x, int y, unsigned char \*pbits, int width , int height, int mode , int fc ,int bc)** |
| --- | --- | --- |
| **Function** | |  |
| **Params** | **in** | **x** |
| **y** |
| **pbits** |
| **width** |
| **height** |
| **mode** |
| **fc** |
| **bc** |
| **out** | **None** |
| **return** | | **Nothing** |
| **remark** | |  |



**demo**

## MfSdkGuiGetDefineColor

| **Prototype** | | **LIB\_EXPORT int MfSdkGuiGetDefineColor(int index)** |
| --- | --- | --- |
| **Function** | |  |
| **Params** | **in** | **index** |
| **out** | **None** |
| **return** | |  |
| **remark** | |  |
| **demo** | |  |

## MfSdkGuiSetCurlSocketIcon

| **Prototype** | | **LIB\_EXPORT void MfSdkGuiSetCurlSocketIcon(char\* recv\_icon,char\* send\_icon,int x)** |
| --- | --- | --- |
| **Function** | | set curl socket status icon |
| **Params** | **in** | **recv\_icon** recv icon buf |
| **send\_icon** send icon buf |
| **x** X-coordinate offset |
| **out** | **None** |
| **return** | | **Nothing** |
| **remark** | | **icon buf only 30\*30 1-bit bmp** |
| **demo** | |  |

## MfSdkGuiRefreshMod

| **Prototype** | **LIB\_EXPORT void MfSdkGuiRefreshMod(int mod)** |
| --- | --- |
| **Function** | Set the screen refresh mode,disable by default |

| **Params** | **in** | 0:Open the refresh task(It is recommended to set  **mod**  when the code has a large number of pixels |
| --- | --- | --- |
| **out** | **None** |
| **return** | |  |
| **remark** | | **Called once after the sys\_init** |
| **demo** | |  |

# Printer module

## Module description

This module mainly includes printer APIs.

## Module structure declaration

None

## Constant declarations

typedef enum

{

back

MFSDK\_PRT\_SUCCESS = 0, //Success MFSDK\_PRT\_FAIL = -1, //Printer unknown fault MFSDK\_PRT\_DEV\_FAIL = -2, //Printer device failure

MFSDK\_PRT\_OUTOF\_PAPER = -3, //The printer is out of paper MFSDK\_PRT\_LOSE\_COMMAND = -4, //Print handle not obtained MFSDK\_PRT\_FILE\_FAIL = -5, //Fail to open the file MFSDK\_PRT\_HANDLE\_BACK = -6, //Split machine handle is not put

MFSDK\_PRT\_CACHE\_ERR = -7, //Save cache failed MFSDK\_PRT\_PARM\_ERROR = -8,//Parameter error

}MfSdkPrtRet\_E;

typedef enum

{

MFSDK\_PRT\_PATTERN\_LEFT = 0, //left

MFSDK\_PRT\_PATTERN\_CENTER = 1, //center MFSDK\_PRT\_PATTERN\_RIGHT = 2, //right

}MfSdkPrtPattern\_E;



## MfSdkPrtInit

| **Prototype** | | **LIB\_EXPORT s32 MfSdkPrtInit(void)** |
| --- | --- | --- |
| **Function** | | **Printer initialize, check the printer status (if it is out of paper).** |
| **Params** | **in** | **Nothing** |
| **out** | **Nothing** |
| **return** | | **For details, see MfSdkPrtRet\_E** |
| **remark** | |  |
| **demo** | | S32 ret = MfSdkPrtInit();  if(ret == MFSDK\_PRT\_SUCCESS)  {  //TODO  } |

## MfSdkPrtStr

| **Prototype** | | **LIB\_EXPORT s32 MfSdkPrtStr( s8 \*str, s32 linegap, MFSDKBOOL newline)** |
| --- | --- | --- |
| **Function** | | **String printing with automatic line break function, support \r\n newline** |
| **Params** | **in** | **s8 \*str** Need to print string information |
| Line spacing: unit pixels, 0 is the default value  **s32 linegap**  (for Pin printing use) |
| **MFSDKBOOL** 0 Does not support line breaks;1 support \r\n  **newline** newline |
| **out** | **Nothing** |
| **return** | | **For details, see MfSdkPrtRet\_E** |
| **remark** | | **Save the string to the printer cache and call Print Start**  **(Print\_Start) to print** |
| **demo** | | S32 ret = UPrint\_Str("AMOUNT", 1, 1); if(ret == MFSDK\_PRT\_SUCCESS)  {  //TODO  } |



## MfSdkPrtBitMap

| **Prototype** | | **LIB\_EXPORT s32 MfSdkPrtBitMap( s8 \*BmpFile, MfSdkPrtPattern\_E pattern)** |
| --- | --- | --- |
| **Function** | | **print image (XXX.bmp)** |
| **Params** | **in** | **s8 \*BmpFile** Image file name (XXX.bmp) |
| **MfSdkPrtPattern\_E** pattern alignment (see 'MfSdkPrtPattern\_E'  **pattern** enum) |
| **out** | **Nothing** |
| **return** | | **For details, see MfSdkPrtRet\_E** |
| **remark** | |  |
| **demo** | | S32 ret =MfSdkPrtBitMap("data//img\_mof.bmp", MFSDK\_PRT\_PATTERN\_CENTER);  if(ret == MFSDK\_PRT\_SUCCESS)  {  //TODO  } |

## MfSdkPrtStart

| **Prototype** | | **LIB\_EXPORT s32 MfSdkPrtStart(void)** |
| --- | --- | --- |
| **Function** | | **Start printing** |
| **Params** | **in** | **Nothing** |
| **out** | **Nothing** |
| **return** | | **For details, see MfSdkPrtRet\_E** |
| **remark** | | **Thermal printing: no more blank lines (processed by the**  **application layer) are required after printing is completed;** |
| **demo** | | S32 ret = MfSdkPrtStart();  if(ret == MFSDK\_PRT\_SUCCESS)  {  //TODO  } |



## MfSdkPrtStrBold

| **Prototype** | | **LIB\_EXPORT s32 MfSdkPrtStrBold( s8 \*pszStr, MfSdkPrtPattern\_E pattern, s32 linegap, MFSDKBOOL newline)** |
| --- | --- | --- |
| **Function** | | **String printing with automatic line break function, support \r\n newline** |
| **Params** | **in** | **s8 \*pszStr** Need to print string information |
| **MfSdkPrtPattern\_E** alignment (see 'MfSdkPrtPattern\_E' enum)  **pattern** |
| Line spacing, unit pixels, 0 is the default value  **s32 linegap**  (for Pin printing use) |
| **MFSDKBOOL** 0 Does not support line breaks;1 support \r\n  **newline** newline |
| **out** | **Nothing** |
| **return** | | **For details, see MfSdkPrtRet\_E** |
| **remark** | |  |
| **demo** | | S32 ret = MfSdkPrtStrBold(“MFSDK”,MFSDK\_PRT\_PATTERN\_LEFT,10,1); if(ret == MFSDK\_PRT\_SUCCESS)  {  //TODO  } |

## MfSdkPrtFeed

| **Prototype** | | **LIB\_EXPORT s32 MfSdkPrtFeed(s32 linegap)** |
| --- | --- | --- |
| **Function** | | **Printer paper** |
| **Params** | **in** | **s32 linegap** Paper length (pixels) |
| **out** | **Nothing** |
| **return** | | **For details, see MfSdkPrtRet\_E** |
| **remark** | |  |
| **demo** | | S32 ret = MfSdkPrtFeed(36);  if(ret == MFSDK\_PRT\_SUCCESS)  {  //TODO  } |



## MfSdkPrtMatrixCode

| **Prototype** | | **LIB\_EXPORT s32 MfSdkPrtMatrixCode(const s8 \*psMatrixCode, s32 nLen, s8 cSize, MfSdkPrtPattern\_E pattern)** |
| --- | --- | --- |
| **Function** | | **Print QR code,Convert incoming data to QR code and print** |
| **Params** | **in** | **const s8** QR code data  **\*psMatrixCode** |
| **s32 nLen** QR code data length |
| **s8 cSize** QR code size, 0-small, 1-medium, 2-large |
| **MfSdkPrtPattern\_E** alignment (see 'MfSdkPrtPattern\_E' enum)  **pattern** |
| **out** | **Nothing** |
| **return** | | **For details, see MfSdkPrtRet\_E** |
| **remark** | |  |
| **demo** | | S32 ret = MfSdkPrtMatrixCode(“MFSDK”,6,1,MFSDK\_PRT\_PATTERN\_CENTER); if(ret == MFSDK\_PRT\_SUCCESS)  {  //TODO  } |

## MfSdkPrtSetFontEN

| **Prototype** | | **LIB\_EXPORT s32 MfSdkPrtSetFontEN(s32 size, s32 zoomW, s32 zoomH)** |
| --- | --- | --- |
| **Function** | | **Set print font(English)** |
| **Params** | **in** | **s32 size** et print English font size(0--8) |
| **s32 zoomW** Set the horizontal magnification of English(1--5) |
| **s32 zoomH** Set the vertical magnification of English(1--5) |
| **out** | **Nothing** |
| **return** | | **For details, see MfSdkPrtRet\_E** |
| **remark** | |  |
| **demo** | | S32 ret = MfSdkPrtSetFontEN(5,4,4);  if(ret == MFSDK\_PRT\_SUCCESS) |



{

//TODO

}

## MfSdkPrtSetFontCN

| **Prototype** | | **LIB\_EXPORT s32 MfSdkPrtSetFontCN(s32 size, s32 zoomW, s32 zoomH)** |
| --- | --- | --- |
| **Function** | | **Set print font(Chinese)** |
| **Params** | **in** | **s32 size** Set print other font size(0--1) |
| **s32 zoomW** Set the horizontal magnification of English(1--5) |
| **s32 zoomH** Set the vertical magnification of English(1--5) |
| **out** | **Nothing** |
| **return** | | **For details, see MfSdkPrtRet\_E** |
| **remark** | |  |
| **demo** | | S32 ret = MfSdkPrtSetFontCN(1,4,4); if(ret == MFSDK\_PRT\_SUCCESS)  {  //TODO  } |

## MfSdkPrtSetDensity

| **Prototype** | | **LIB\_EXPORT s32 MfSdkPrtSetDensity(s32 val)** |
| --- | --- | --- |
| **Function** | | **Set print density** |
| **Params** | **in** | **s32 val** Set print density (1--5, 3 is normal) |
| **out** | **Nothing** |
| **return** | | **For details, see MfSdkPrtRet\_E** |
| **remark** | |  |
| **demo** | | S32 ret = MfSdkPrtSetDensity(3); if(ret == MFSDK\_PRT\_SUCCESS)  {  //TODO  } |



## MfSdkPrtSetAlign

| **Prototype** | | **LIB\_EXPORT void MfSdkPrtSetAlign(MfSdkPrtPattern\_E pattern)** |
| --- | --- | --- |
| **Function** | | **String printing with automatic line break function, support \r\n newline** |
| **Params** | **in** | **pattern** alignment (see 'MfSdkPrtPattern\_E' enum) |
| **out** | **Nothing** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | | MfSdkPrtInit(MFSDK\_PRT\_PATTERN\_CENTER); |

## MfSdkPrtSetLineAlign

| **Prototype** | | **LIB\_EXPORT void MfSdkPrtSetLineAlign(MfSdkPrtPattern\_E pattern)** |
| --- | --- | --- |
| **Function** | | **Set the line print buffer align format,will not affect next line** |
| **Params** | **in** | **MfSdkPrtPattern\_E** alignment (see 'MfSdkPrtPattern\_E' enum)  **pattern** |
| **out** | **Nothing** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | | MfSdkPrtSetLineAlign(MFSDK\_PRT\_PATTERN\_CENTER); |

## MfSdkPrtStrLine

| **Prototype** | | **LIB\_EXPORT void MfSdkPrtStrLine( s8 \*sLeft, s8 \*sRight, s32 nLinegap)** |
| --- | --- | --- |
| **Function** | | **String printing with automatic line feed function, support \r\n newline** |
| **Params** | **in** | **Nothing** |
| **out** | **Nothing** |
| **return** | | **Nothing** |



| **remark** |  |
| --- | --- |
| **demo** | MfSdkPrtStrLine(“amt”,”123”,5); |

## MfSdkPrtSetLineSpace

| **Prototype** | | **LIB\_EXPORT void MfSdkPrtSetLineSpace(s32 nLinegap)** |
| --- | --- | --- |
| **Function** | | **set the line space of follow print buffer** |
| **Params** | **in** | **nLinegap** line space value |
| **out** | **Nothing** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | |  |

## MfSdkPrtSetFontMode

| **Prototype** | | **LIB\_EXPORT void MfSdkPrtSetFontMode(s32 mod)** |
| --- | --- | --- |
| **Function** | | **Set the printing mode for English letters and numbers** |
| **Params** | **in** | 1 Print English letters and numbers characters  **mod** using font files; 0 or no setting, the default print uses the code font |
| **out** | **Nothing** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | |  |

## MfSdkPrtCheckNopaper

| **Prototype** | **LIB\_EXPORT s32 MfSdkPrtCheckNopaper()** |
| --- | --- |
| **Function** | **Check if the device is out of paper** |



| **Params** | **in** | **Nothing** |
| --- | --- | --- |
| **out** | **Nothing** |
| **return** | | **Ref. MfSdkPrtRet\_E** |
| **remark** | |  |
| **demo** | |  |

## MfSdkPrtSetArabicNumOrder

| **Prototype** | | **LIB\_EXPORT s32 MfSdkPrtSetArabicNumOrder(s32 order)** |
| --- | --- | --- |
| **Function** | | **Set the printing mode for English letters and numbers** |
| **Params** | **in** | **order** 1:Set up Arabic digital reverse printing |
| **out** | **Nothing** |
| **return** | | **Ref. MfSdkPrtRet\_E** |
| **remark** | |  |
| **demo** | |  |

# Pinpad module

## Module description

This module mainly includes pin entry device.

## Module structure declaration

typedef struct

{

u8 mode; // mode 2,open encrypted pin(The backbutton deletes a pin); 3,open encrypted pin(The backbutton deletes all pin);

u8 min;//Minimum pin length u8 max;//Maximum pin length u32 timeoverMs;//timeover(ms)

u8 isBypass;//Whether to support bypass

} MfSdkPedPinModCfg\_T;

typedef struct

{

s8 appVer[10]; s8 pinpadSn[32]; s8 model[10];

}MfSdkPedPinpadInfo\_T;

## Constant declarations

typedef enum

{

MFSDK\_PED\_NOT\_KEY = -22, // No injection key MFSDK\_PED\_PARM\_ERROR = -21, // Parameter error MFSDK\_PED\_PINPAD\_INSUFFICIENT\_SPACE = -16,//The pinpad

space is insufficient

MFSDK\_PED\_PINPAD\_NO\_PUBLIC\_KEY\_SIGN = -15,//The pinpad upgrade lacks the public key signature

MFSDK\_PED\_PINPAD\_REPEATED\_SET\_MODE = -12, //Repeated

setting mode

MFSDK\_PED\_PINPAD\_UPDATEFILE\_TOO\_BIG = -11, // update file

too big

MFSDK\_PED\_UPDATEFILE\_FORMAT\_ERR = -10, // update file

format error

MFSDK\_PED\_UPDATE\_COMPLETE\_RECONNECT\_FAILED = -9, //

failed to reconnect to the pinpad after the update is complete MFSDK\_PED\_MD5\_FAILED = -8, // pinpad app md5 authentication

failed

MFSDK\_PED\_WRITE\_FAILED = -7, // pinpad failed to write files MFSDK\_PED\_OPEN\_FAILED = -6, // app to be updated failed to open MFSDK\_PED\_NO\_APP\_UPDATE = -5, // no apps to update(mo

data/pinpad.bin)

MFSDK\_PED\_PINPAD\_TIMEOUT = -2, // External Pinpad communication timed out

MFSDK\_PED\_OTHER = -1,

MFSDK\_PED\_SUCCESS = 0, // Success

}MfSdkPedRet\_E; typedef enum

{

MFSDK\_PED\_ENCRYPT = 0, // Encryption mode MFSDK\_PED\_DECRYPT = 1, // Decryption mode

} MfSdkPedMod\_E;

typedef enum

{

MFSDK\_PED\_DES\_ECB = 0, // des mode ECB MFSDK\_PED\_DES\_CBC = 1, // des mode CBC



} MfSdkPedDesMod\_E;

typedef enum

{

MFSDK\_PED\_PINPAD\_INPUT\_PIN = 0, MFSDK\_PED\_PINPAD\_PRESS\_OK, MFSDK\_PED\_PINPAD\_PRESS\_ESC,

}MfSdkPedPinpadStatus\_E;

#define MFSDK\_MKSK\_MAINKEY\_TYPE 0x00

#define MFSDK\_MKSK\_PINENC\_TYPE 0x01 #define MFSDK\_MKSK\_MACENC\_TYPE 0x02 #define MFSDK\_MKSK\_MAGDEC\_TYPE 0x03 #define MFSDK\_MKSK\_TRANSKEY\_TYPE 0x04

#define MFSDK\_SEC\_FIXED\_FIELD 0x00 #define MFSDK\_SEC\_MKSK\_FIELD 0x01 #define MFSDK\_SEC\_DUKPT\_FIELD 0x02

#define MFSDK\_SEC\_MAC\_UPAY\_FORMAT 0x01 #define MFSDK\_SEC\_MAC\_X99\_FORMAT 0x02 #define MFSDK\_SEC\_MAC\_X919\_FORMAT 0x03 #define MFSDK\_SEC\_MAC\_XOR\_FORMAT 0x04

#define MFSDK\_SEC\_PIN\_FORMAT0 0x00 #define MFSDK\_SEC\_PIN\_FORMAT1 0x01 #define MFSDK\_SEC\_PIN\_FORMAT2 0x02 #define MFSDK\_SEC\_PIN\_FORMAT3 0x03 #define MFSDK\_SEC\_PIN\_FORMAT4 0x04

#define MFSDK\_DUKPT\_DES\_KEY\_PIN 0x00 #define MFSDK\_DUKPT\_DES\_KEY\_MAC1 0x01 #define MFSDK\_DUKPT\_DES\_KEY\_MAC2 0x02 #define MFSDK\_DUKPT\_DES\_KEY\_DATA1 0x03 #define MFSDK\_DUKPT\_DES\_KEY\_DATA2 0x04 #define MFSDK\_DUKPT\_DES\_KEY\_PEK 0x05

typedef enum

{

MFSDK\_PED\_TR31\_TYPE\_IPEK = 0, MFSDK\_PED\_TR31\_TYPE\_BDK,

}MfSdkPedTr31KeyType\_E;



typedef enum

{

MFSDK\_PED\_PINPAD\_UPDATE\_START = 0x01, MFSDK\_PED\_PINPAD\_UPDATEING, MFSDK\_PED\_PINPAD\_UPDATE\_END, MFSDK\_PED\_PINPAD\_RESTORE\_CONNECTION,

}MfSdkPedPinpadUpdateStatus\_E;

## MfSdkPedSetKeySize

| **Prototype** | | **LIB\_EXPORT s32 MfSdkPedSetKeySize(s32 size)** |
| --- | --- | --- |
| **Function** | | **Set the key length** |
| **Params** | **in** | **size** 16 or 24 |
| **out** | **nothing** |
| **return** | | **For details, see MfSdkPedRet\_E.** |
| **remark** | |  |
| **demo** | | S32 ret = MfSdkPedSetKeySize(16); if(ret == MFSDK\_PED\_SUCCESS)  {  //TODO  } |

## MfSdkPedGetKeySize

| **Prototype** | | **LIB\_EXPORT s32 MfSdkPedGetKeySize(void)** |
| --- | --- | --- |
| **Function** | | **Get the key length** |
| **Params** | **in** | **nothing** |
| **out** | **nothing** |
| **return** | | **key length** |
| **remark** | |  |
| **demo** | | S32 keyLength = MfSdkPedGetKeySize(); |



## MfSdkPedDukptLoadKey

| **Prototype** | | **LIB\_EXPORT s32 MfSdkPedDukptLoadKey(s32 mode, s32 type, s32 gid, u8 \*init\_ksn, u8 \*init\_key, u8 \*kvc)** |
| --- | --- | --- |
| **Function** | | **initialize the dukpt key use IPEK ciphertext** |
| **Params** | **in** | encryption method of initial key(0 - plaintext, 1 -  **s32 mode**  tmk encryption, 2 - kek encryption) |
| **s32 type** initial key type(0 - ipek, 1 - bdk) |
| **s32 gid** key grouping,,max 100 group (0 - 99) |
| **u8 \*init\_ksn** initial ksn |
| **u8 \*init\_key** Initial key |
| **out** | **u8 \*kvc** key kvc(Key plaintext encryption 8 0x00) |
| **return** | | **For details, see MfSdkPedRet\_E.** |
| **remark** | |  |
| **demo** | | u8 kvc[8] = { 0 };  u8 ksn[10] = "\xFF\xFF\x01\x11\x7C\x70\xCA\x00\x00\x00"; u8 ipek[16] =  "\xC1\xD0\xF8\xFB\x49\x58\x67\x0D\xBA\x40\xAB\x1F\x37\x52\xEF\x0D"; S32 ret = MfSdkPedDukptLoadKey(0,0,0, ksn, ipek, kvc);  if(ret == MFSDK\_PED\_SUCCESS)  {  //TODO  } |

## MfSdkPedDukpt3desRun

| **Prototype** | | **LIB\_EXPORT s32 MfSdkPedDukpt3desRun(MfSdkPedMod\_E mode, s8 \*inData, s32 size, s8 \*outData, MfSdkPedDesMod\_E**  **desMode, s32 keyTpye)** |
| --- | --- | --- |
| **Function** | | **use the previously obtained key 3des operation** |
| **Params** | **in** | **MfSdkPedMod\_E** mode(Ref. enum'MfSdkPedMod\_E' )  **mode** |
| **s8 \*inData** raw data |
| **s32 size** data length (8-byte multiple) |
| **MfSdkPedDesMod\_** des mode(see 'MfSdkPedDesMod\_E' enum)  **E desMode** |



|  |  | key type(0 - MFSDK\_DUKPT\_DES\_KEY\_PIN, 1 - MFSDK\_DUKPT\_DES\_KEY\_MAC1,  **s32 keyTpye** 2 - MFSDK\_DUKPT\_DES\_KEY\_MAC2,   1. - MFSDK\_DUKPT\_DES\_KEY\_DATA1, 2. - MFSDK\_DUKPT\_DES\_KEY\_DATA2) |
| --- | --- | --- |
| **out** | **s8 \*outData** out data |
| **return** | | **For details, see MfSdkPedRet\_E.** |
| **remark** | |  |
| **demo** | | u8 ksn[20] = { 0 };  u8 inData[] = "morefun"; u8 ounData[20] = { 0 };  S32 ret = MfSdkPedDukptGetKsn(0, ksn); if(ret == MFSDK\_PED\_SUCCESS)  {  ret = MfSdkPedDukpt3desRun(MFSDK\_PED\_ENCRYPT, inData, 8, ounData, MFSDK\_PED\_DES\_ECB, MFSDK\_DUKPT\_DES\_KEY\_DATA1);  if(ret == MFSDK\_PED\_SUCCESS)  {  //TODO  }  } |

## MfSdkPedDukptGetKsn

| **Prototype** | | **LIB\_EXPORT s32 MfSdkPedDukptGetKsn(u32 gid, u8 \*ksn)** |
| --- | --- | --- |
| **Function** | | **get key corresponds to ksn** |
| **Params** | **in** | **u32 gid** gid key grouping ,max 100 group (0 - 99) |
| **out** | **u8 \*ksn** key corresponds to ksn |
| **return** | | **For details, see MfSdkPedRet\_E.** |
| **remark** | |  |
| **demo** | | u8 ksn[20] = { 0 };  S32 ret = MfSdkPedDukptGetKsn(0, ksn); if(ret == MFSDK\_PED\_SUCCESS)  {  //TODO  } |



## MfSdkPedMkSkSavePlaintextKey

| **Prototype** | | **LIB\_EXPORT s32 MfSdkPedMkSkSavePlaintextKey(s32 type, u32 gid, u8 \*key, u8 \*kvc)** |
| --- | --- | --- |
| **Function** | | **get key corresponds to ksn** |
| **Params** | **in** | type key type(   1. - MFSDK\_MKSK\_MAINKEY\_TYPE, 2. - MFSDK\_MKSK\_PINENC\_TYPE,   **s32 type**   1. - MFSDK\_MKSK\_MACENC\_TYPE, 2. - MFSDK\_MKSK\_MAGDEC\_TYPE, 3. - MFSDK\_MKSK\_TRANSKEY\_TYPE) |
| **u32 gid** key grouping ,max 100 group (0 - 99) |
| **u8 \*key** key key plaintext |
| **out** | **u8 \*kvc** key kvc(key plaintext encryption 8 0x00) |
| **return** | | **For details, see MfSdkPedRet\_E.** |
| **remark** | |  |
| **demo** | | u8 kvc[8] = { 0 }; u8 key[16] =  "\xC1\xD0\xF8\xFB\x49\x58\x67\x0D\xBA\x40\xAB\x1F\x37\x52\xEF\x0D"; S32 ret =  MfSdkPedMkSkSavePlaintextKey(MFSDK\_MKSK\_MAINKEY\_TYPE,0,key, kvc); if(ret == MFSDK\_PED\_SUCCESS)  {  //TODO  } |

## MfSdkPedMkSkSaveEncryptedKey

| **Prototype** | | **LIB\_EXPORT s32 MfSdkPedMkSkSaveEncryptedKey(s32 type, u32 gid, u8 \*key, MfSdkPedDesMod\_E desMode, u8 \*kvc)** |
| --- | --- | --- |
| **Function** | | **save key ciphertext** |
| **Params** | **in** | type key type(   1. - MFSDK\_MKSK\_MAINKEY\_TYPE,   **s32 type**   1. - MFSDK\_MKSK\_PINENC\_TYPE, 2. - MFSDK\_MKSK\_MACENC\_TYPE, |



|  |  | 1. - MFSDK\_MKSK\_MAGDEC\_TYPE, 2. - MFSDK\_MKSK\_TRANSKEY\_TYPE) |
| --- | --- | --- |
| **u32 gid** key grouping ,max 100 group (0 - 99) |
| **MfSdkPedDesMod\_** des mode(see 'MfSdkPedDesMod\_E' enum)  **E desMode** |
| **u8 \*key** key ciphertext |
| **out** | **u8 \*kvc** key kvc(key plaintext encryption 8 0x00) |
| **return** | | **For details, see MfSdkPedRet\_E.** |
| **remark** | |  |
| **demo** | | u8 kvc[8] = { 0 }; u8 key[16] =  "\xC1\xD0\xF8\xFB\x49\x58\x67\x0D\xBA\x40\xAB\x1F\x37\x52\xEF\x0D"; S32 ret =  MfSdkPedMkSkSavePlaintextKey(MFSDK\_MKSK\_MAINKEY\_TYPE,0,key, kvc); if(ret == MFSDK\_PED\_SUCCESS)  {  //TODO  } |

## MfSdkPedMkSkGetKcv

| **Prototype** | | **LIB\_EXPORT s32 MfSdkPedMkSkGetKcv(s32 keyIndex, s32 keyType, u8 \*OutKcv)** |
| --- | --- | --- |
| **Function** | | **get mksk kcv** |
| **Params** | **in** | **s32 keyIndex** keyIndex key index(0-99) |
| (0 - MFSDK\_MKSK\_MAINKEY\_TYPE,  1 - MFSDK\_MKSK\_PINENC\_TYPE,  **s32 keyType** 2 - MFSDK\_MKSK\_MACENC\_TYPE,  3 - MFSDK\_MKSK\_MAGDEC\_TYPE,  4 - MFSDK\_MKSK\_TRANSKEY\_TYPE) |
| **out** | **u8 \*OutKcv** key kvc(key plaintext encryption 8 0x00) |
| **return** | | **For details, see MfSdkPedRet\_E.** |
| **remark** | |  |
| **demo** | | u8 kcv[10] = { 0 };  s32 ret = MfSdkPedMkSkGetKcv(0,MFSDK\_MKSK\_MAINKEY\_TYPE,kcv); |



## MfSdkPedMkSk3desRun

| **Prototype** | | **LIB\_EXPORT s32 MfSdkPedMkSk3desRun(s32 type, s32 gid, MfSdkPedMod\_E mode, u8 \*ind, s32 size, u8 iv[8], u8 \*outd,**  **MfSdkPedDesMod\_E desMode)** |
| --- | --- | --- |
| **Function** | | **use key 3des operation** |
| **Params** | **in** | (0 - MFSDK\_MKSK\_MAINKEY\_TYPE,  1 - MFSDK\_MKSK\_PINENC\_TYPE,  **s32 type** 2 - MFSDK\_MKSK\_MACENC\_TYPE,  3 - MFSDK\_MKSK\_MAGDEC\_TYPE,  4 - MFSDK\_MKSK\_TRANSKEY\_TYPE) |
| **s32 gid** key grouping,max 100 group (0 - 99) |
| **MfSdkPedMod\_E** mode(see 'MfSdkPedMod\_E' enum)  **mode** |
| **u8 \*ind** raw data |
| **s32 size** data length (8-byte multiple) |
| **u8 iv[8]** initialization vector |
| **MfSdkPedDesMod\_** des mode(see 'MfSdkPedDesMod\_E' enum)  **E desMode** |
| **out** | **u8 \*outd** calculation results |
| **return** | | **For details, see MfSdkPedRet\_E.** |
| **remark** | |  |
| **demo** | | u8 ksn[20] = { 0 };  u8 inData[] = "morefun"; u8 ounData[20] = { 0 }; u8 iv[8] = { 0 };  S32 ret = 0;  ret = MfSdkPedMkSk3desRun(MFSDK\_MKSK\_MAINKEY\_TYPE, 0,MFSDK\_PED\_ENCRYPT,inData,8,iv,ounData,MFSDK\_PED\_DES\_ECB); if(ret == MFSDK\_PED\_SUCCESS)  {  //TODO}  } |

## MfSdkPedMacProc

**LIB\_EXPORT s32 MfSdkPedMacProc(s32 fid, s32 gid, s32 format, u8 \*data, s32 len, u8 \*mac, s32 keyTpye, s32 keySize)**

****

| **Function** | | **computing mac** |
| --- | --- | --- |
| **Params** | **in** | field id (1 - MFSDK\_SEC\_MKSK\_FIELD, 2 -  **s32 fid**  MFSDK\_SEC\_DUKPT\_FIELD) |
| **s32 gid** key grouping ,max 100 group (0 - 99) |
| mac format (  1 - MFSDK\_SEC\_MAC\_UPAY\_FORMAT,  **s32 format** 2 - MFSDK\_SEC\_MAC\_X99\_FORMAT,  3 - MFSDK\_SEC\_MAC\_X919\_FORMAT,  4 - MFSDK\_SEC\_MAC\_XOR\_FORMAT) |
| **u8 \*data** mac source data |
| **s32 len** data length |
| key type(  **s32 keyTpye** 1 - MFSDK\_DUKPT\_DES\_KEY\_MAC1,  2 - MFSDK\_DUKPT\_DES\_KEY\_MAC2) |
| **s32 keySize** key size |
| **out** | **u8 \*mac** mac |
| **return** | | **For details, see MfSdkPedRet\_E.** |
| **remark** | |  |
| **demo** | | U8 data[10] = “\x12\x34\x56”; U8 mac[30]={0};  S32 ret = MfSdkPedMacProc(MFSDK\_SEC\_DUKPT\_FIELD,0,MFSDK\_SEC\_MAC\_X919\_FORMAT,data,8, mac,MFSDK\_DUKPT\_DES\_KEY\_MAC1,16);  if(ret == MFSDK\_PED\_SUCCESS)  {  //TODO}  } |

## MfSdkPedEncryptPinProc

| **Prototype** | | **LIB\_EXPORT s32 MfSdkPedEncryptPinProc(s32 fid, s32 format, s32 gid, u8 \*pan, u8 \*pinBlock, u8 \*pin)** |
| --- | --- | --- |
| **Function** | | **read pin ciphertext from the security keyboard** |
| **Params** | **in** | field id (1 - MFSDK\_SEC\_MKSK\_FIELD, 2 -  **s32 fid**  MFSDK\_SEC\_DUKPT\_FIELD) |
| pin format (  **s32 format**  0 - MFSDK\_SEC\_PIN\_FORMAT0, |



|  |  | 1. - MFSDK\_SEC\_PIN\_FORMAT1, 2. - MFSDK\_SEC\_PIN\_FORMAT2, 3. - MFSDK\_SEC\_PIN\_FORMAT3, 4. - MFSDK\_SEC\_PIN\_FORMAT4) |
| --- | --- | --- |
| **s32 gid** key grouping ,max 100 group (0 - 99) |
| **u8 \*pan** card number |
| **u8 \*pin** pin plaintext |
| **out** | **u8 \*pinBlock** pin Block |
| **return** | | **For details, see MfSdkPedRet\_E.** |
| **remark** | |  |
| **demo** | | U8 pan[10] = “1234123488888888”;  U8 pinblock[30]={0}; S32 ret =  MfSdkPedEncryptPinProc(MFSDK\_SEC\_DUKPT\_FIELD,MFSDK\_SEC\_PIN\_FORMAT0,0,pan,pin block,”1234”);  if(ret == MFSDK\_PED\_SUCCESS)  {  //TODO}  } |

## MfSdkPedSetPinModeCfgInit

| **Prototype** | | **LIB\_EXPORT void MfSdkPedSetPinModeCfgInit(void)** |
| --- | --- | --- |
| **Function** | | **Pin mode config init**  **pin is plaintext and 4-6 , it does not support bypass MfSdkPedPinModCfg\_T flag is 0** |
| **Params** | **in** | **Nothing** |
| **out** | **Nothing** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | |  |

## MfSdkPedSetPinModeCfg

**LIB\_EXPORT void MfSdkPedSetPinModeCfg(MfSdkPedPinModCfg\_T cfg)**

****

| **Function** | | **Set enable pin input mode config** |
| --- | --- | --- |
| **Params** | **in** | **MfSdkPedPinModC** For details, see MfSdkPedPinModCfg\_T  **fg\_T cfg** |
| **out** | **Nothing** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | | MfSdkPedPinModCfg\_T pinCfg = { 0 }; pinCfg.mode = 2;  pinCfg.min = 4;  pinCfg.max = 6;  pinCfg.timeoverMs = 45000;  pinCfg.isBypass = 1; MfSdkPedSetPinModeCfg(pinCfg); |

## MfSdkPedEncryptPinProcEx

| **Prototype** | | **LIB\_EXPORT s32 MfSdkPedEncryptPinProcEx(u8\* pan, u8\* pinBlock)** |
| --- | --- | --- |
| **Function** | | **Calculated pinblock value,MfSdkPedSetPinModeCfg needs to be called first.** |
| **Params** | **in** | **pan** Card PAN |
| **out** | **pinBlock** |
| **return** | | **For details, see MfSdkPedRet\_E.** |
| **remark** | |  |
| **demo** | | MfSdkPedPinModCfg\_T pinCfg = { 0 }; pinCfg.mode = 2;  pinCfg.min = 4;  pinCfg.max = 6;  pinCfg.timeoverMs = 45000;  pinCfg.isBypass = 1; MfSdkPedSetPinModeCfg(pinCfg); u8\* pan = “1234567890123456”;  u8 pinBlock[8] = {0};  s32 nRet = MfSdkPedEncryptPinProcEx(pan, pinBlock); |



## MfSdkPedGetPinModeCfg

| **Prototype** | | **LIB\_EXPORT MfSdkPedPinModCfg\_T**  **\*MfSdkPedGetPinModeCfg(void)** |
| --- | --- | --- |
| **Function** | | **Get pin input mode config** |
| **Params** | **in** | **nothing** |
| **out** | **nothing** |
| **return** | | **MfSdkPedPinModCfg\_T point, pin input mode config** |
| **remark** | |  |
| **demo** | | MfSdkPedPinModCfg\_T pinCfg = { 0 };  &pinCfg = MfSdkPedGetPinModeCfg(); |

## MfSdkPedGetPinModeStatus

| **Prototype** | | **LIB\_EXPORT s32 MfSdkPedGetPinModeStatus(s32 \*length)** |
| --- | --- | --- |
| **Function** | | **Get pin mode status** |
| **Params** | **in** | **Nothing** |
| **out** | **s32 \*length** pin length |
| **return** | | **0 Still input** |
| **1 Press the enter(OK) button** |
| **2 Press the exit button** |
| **3 timeover** |
| **remark** | |  |
| **demo** | | S32 len=0;  S32 ret = MfSdkPedGetPinModeStatus(&len); If(ret == 1)  {  //user press the enter(OK) button  } |



## MfSdkPedSaveRsaPriKey

| **Prototype** | | **LIB\_EXPORT s32 MfSdkPedSaveRsaPriKey(s32 index, s32 length, u8 \*p, u8 \*q)** |
| --- | --- | --- |
| **Function** | | **save the private key to the security module** |
| **Params** | **in** | **s32 index** index key index (0 - 9) |
| **s32 length** rsa byte size(128/256) |
| **u8 \*p** private key P component |
| **u8 \*q** private key Q component |
| **out** | **Nothing** |
| **return** | | **For details, see MfSdkPedRet\_E.** |
| **remark** | |  |
| **demo** | | s32 ret = MfSdkPedSaveRsaPriKey(0,128,”10”,”20”);  if(ret == MFSDK\_PED\_SUCCESS)  {  //TODO}  } |

## MfSdkPedSaveRsaPukKey

| **Prototype** | | **LIB\_EXPORT s32 MfSdkPedSaveRsaPukKey(s32 index, s32 length, u8 \*n)** |
| --- | --- | --- |
| **Function** | | **save the public key to the security module** |
| **Params** | **in** | **s32 index** index key index (0 - 9) |
| **s32 length** length rsa byte size(128/256) |
| **u8 \*n** public key N component |
| **out** | **Nothing** |
| **return** | | **For details, see MfSdkPedRet\_E.** |
| **remark** | |  |
| **demo** | | s32 ret = MfSdkPedSaveRsaPukKey(0,128,”10”);  if(ret == MFSDK\_PED\_SUCCESS)  {  //TODO}  } |



## MfSdkPedRsaBlock

| **Prototype** | | **LIB\_EXPORT s32 MfSdkPedRsaBlock(s32 index, s8 \*inD, s8**  **\*outD, s32 length)** |
| --- | --- | --- |
| **Function** | | **RSA block calculation** |
| **Params** | **in** | **s32 index** index key index (0 - 9) |
| **s8 \*inD** in data |
| **s32 length** rsa key byte size(128/256) |
| **out** | **s8 \*outD** out data |
| **return** | | **For details, see MfSdkPedRet\_E.** |
| **remark** | |  |
| **demo** | | s8 out[20]={0};  s32 ret = MfSdkPedRsaBlock(0,”1234”,out,128);  if(ret == MFSDK\_PED\_SUCCESS)  {  //TODO}  } |

## MfSdkPedDeleteKey

| **Prototype** | | **LIB\_EXPORT s32 MfSdkPedDeleteKey(s32 keyType, s32 keyIndex)** |
| --- | --- | --- |
| **Function** | | **elete index key** |
| **Params** | **in** | **s32 keyType** 0:MKSK,1:DUKPT |
| **s32 keyType** key index(0-99) |
| **out** | **Nothing** |
| **return** | | **For details, see MfSdkPedRet\_E.** |
| **remark** | |  |
| **demo** | | s8 out[20]={0};  s32 ret = MfSdkPedDeleteKey(0,0);  if(ret == MFSDK\_PED\_SUCCESS)  {  //TODO}  } |



## MfSdkPedTr31GetData

| **Prototype** | | **LIB\_EXPORT s32 MfSdkPedTr31GetData(u32 type, u8 \*mk, s32 nKeySize, u8 \*data, s32 size, s8 \*wk)** |
| --- | --- | --- |
| **Function** | | **save key plaintext** |
| **Params** | **in** | **u32 type** default 0 |
| **u8 \*mk** key for decrypt the TR31 string |
| **s32 nKeySize** mk size |
| **u8 \*data** TR31 string |
| **s32 size** TR31 data size |
| **out** | **s8 \*wk** Decrypted key |
| **return** | | **For details, see MfSdkPedRet\_E.** |
| **remark** | |  |
| **demo** | |  |

## MfSdkPedRkiDukptLoad

| **Prototype** | | **LIB\_EXPORT s32 MfSdkPedRkiDukptLoad(const s8 \*keyData, s32 len)** |
| --- | --- | --- |
| **Function** | | **load the rki key use dukpt** |
| **Params** | **in** | **const s8 \*keyData** key data |
| **s32 len** key length |
| **out** | **s8 \*wk** Decrypted key |
| **return** | | **otherr Failed** |
| **0 Success** |
| **remark** | |  |
| **demo** | |  |

## MfSdkPedRkiCheckAppKey

**Prototype LIB\_EXPORT s32 MfSdkPedRkiCheckAppKey( u8 \*sHashValue)**

****

| **Function** | | **verify the app hash** |
| --- | --- | --- |
| **Params** | **in** | **sHashValue** sha256 value of app KEY |
| **out** | **Nothing** |
| **return** | | **1, Yes; 0, No** |
| **remark** | |  |
| **demo** | |  |

## MfSdkPedSetPinpad

| **Prototype** | | **LIB\_EXPORT s32 MfSdkPedSetPinpad(s32 mod)** |
| --- | --- | --- |
| **Function** | | **set input pinpad** |
| **Params** | **in** | **s32 mod** 1:external Pinpad, 0:device's Pinpad |
| **out** | **Nothing** |
| **return** | | **For details, see MfSdkPedRet\_E.** |
| **remark** | |  |
| **demo** | | MfSdkPedSetPinpad(1); |

## MfSdkPedGetPinpad

| **Prototype** | | **LIB\_EXPORT s32 MfSdkPedGetPinpad(void)** |
| --- | --- | --- |
| **Function** | | **get input pinpad** |
| **Params** | **in** | **Nothing** |
| **out** | **Nothing** |
| **return** | | **1 external Pinpad** |
| **0 device's Pinpad** |
| **remark** | |  |
| **demo** | | s32 ret = MfSdkPedGetPinpad(); If(ret == 1)  {  //external pinpad |

## MfSdkPedSetPedUi

| **Prototype** | | **LIB\_EXPORT s32 MfSdkPedSetPedUi(u8 \*buff, u32 len)** |
| --- | --- | --- |
| **Function** | | **Send page buf to external pinpad** |
| **Params** | **in** | **u8 \*buff** page buff |
| **u32 len** buff len |
| **out** | **Nothing** |
| **return** | | **For details, see MfSdkPedRet\_E** |
| **remark** | |  |
| **demo** | | u8 buf[153600]={0};  s32 ret = MfSdkPedSetPedUi(buf,153600); If(ret == MFSDK\_PED\_SUCCESS )  {  //TODO  } |

## MfSdkPedCheckCardOnce

| **Prototype** | | **LIB\_EXPORT s32 MfSdkPedCheckCardOnce(void)** |
| --- | --- | --- |
| **Function** | | **Use the external pinpad to check card** |
| **Params** | **in** | **Nothing** |
| **out** | **Nothing** |
| **return** | | **0 no card** |
| **1 icc card** |
| **2 rfid cad** |
| **other fail** |
| **remark** | |  |
| **demo** | | s32 ret = MfSdkPedCheckCardOnce(); if(ret == 1)  {  //ICC card |

## MfSdkPedIsWithPinpad

| **Prototype** | | **LIB\_EXPORT s32 MfSdkPedIsWithPinpad(s32 timeoutMs)** |
| --- | --- | --- |
| **Function** | | **Whether the terminal has external pinpad.** |
| **Params** | **in** | **s32 timeoutMs** timeout ms |
| **out** | **Nothing** |
| **return** | | **1 Connect the external pinpad** |
| **0 External pinpad is not connected** |
| **remark** | |  |
| **demo** | | s32 ret = MfSdkPedIsWithPinpad(3000); if(ret == 1)  {  //Connect the external pinpad  } |

## MfSdkPedExPinpadUpdate

| **Prototype** | | **s32 MfSdkPedExPinpadUpdate(MfSdkPedPinpadUpdateCb cb)** |
| --- | --- | --- |
| **Function** | | **Update the external pinpad app.The pinpad app path must be in data/pinpad.bin** |
| **Params** | **in** | **MfSdkPedPinpadU** Set the callback for the update process(500ms  **pdateCb cb** call once) |
| **out** | **Nothing** |
| **return** | | **For details, see MfSdkPedRet\_E** |
| **remark** | |  |
| **demo** | | static s32 AppPedPinpadUpdateCb(MfSdkPedPinpadUpdateStatus\_E status, s32 total, s32 updatedSize, u32 time)  {  return 0;  }  s32 ret = MfSdkPedExPinpadUpdate(AppPedPinpadUpdateCb); If(ret == MFSDK\_PED\_SUCCESS ) |



{

//TODO

}

## MfSdkPedGetExPinpadVersion

| **Prototype** | | **s32 MfSdkPedGetExPinpadVersion(s8 \*buf, s32 bufSize)** |
| --- | --- | --- |
| **Function** | | **Get the version of the external pinpad** |
| **Params** | **in** | **bufSize** Version buff size |
| **out** | **buf** Version buff |
| **return** | | **For details, see MfSdkPedRet\_E** |
| **remark** | |  |
| **demo** | | s8 buf[20] = {0};  s32 ret = MfSdkPedGetExPinpadVersion(buf,20); If(ret == MFSDK\_PED\_SUCCESS )  {  //TODO  } |

## MfSdkPedGetExPinpadInfo

| **Prototype** | | **LIB\_EXPORT s32**  **MfSdkPedGetExPinpadInfo(MfSdkPedPinpadInfo\_T \*info)** |
| --- | --- | --- |
| **Function** | | **Get the info of the external pinpad** |
| **Params** | **in** | **nothing** |
| **out** | **info** pinpad info |
| **return** | | **For details, see MfSdkPedRet\_E** |
| **remark** | |  |
| **demo** | | MfSdkPedPinpadInfo\_T pinfo = {0};  s32 ret = MfSdkPedPinpadInfo\_T(&pinfo ); If(ret == MFSDK\_PED\_SUCCESS )  {  //TODO  } |



## MfSdkPedSetPedPinMod

| **Prototype** | | **LIB\_EXPORT s32 MfSdkPedSetPedPinMod(s32 min, s32 max)** |
| --- | --- | --- |
| **Function** | | **Set the external pinpad pin** |
| **Params** | **in** | **s32 min** pin min len |
| **s32 max** pin max len |
| **out** | **Nothing** |
| **return** | | **For details, see MfSdkPedRet\_E.** |
| **remark** | |  |
| **demo** | | s32 ret = MfSdkPedSetPedPinMod(4,6);  if(ret == MFSDK\_PED\_SUCCESS)  {  //TODO}  } |

## MfSdkPedGetPin

| **Prototype** | | **LIB\_EXPORT s32 MfSdkPedGetPin(s32 \*len, s8 \*pinBuf)** |
| --- | --- | --- |
| **Function** | | **Gets the pin of the external pinpad** |
| **Params** | **in** | **Nothing** |
| **out** | **s32 \*len** Enter the pin length |
| **s8 \*pinBuf** pin |
| **return** | | **For details, see MfSdkPedPinpadStatus\_E** |
| **remark** | |  |
| **demo** | | s8 pinBuf[10]={0}; s32 pinLen=0; s32 ret=0;  ret = MfSdkPedGetPin(&pinLen,pinBuf);  if(ret == MFSDK\_PED\_PINPAD\_PRESS\_OK)  {  //pinpad press OK  } |



## MfSdkPedTR31LoadKBPK

| **Prototype** | | **s32 MfSdkPedTR31LoadKBPK(s32 gid, u8 \*kbpk, u32 kbpkLen)** |
| --- | --- | --- |
| **Function** | | **Load TR-31 clear-text KBPK** |
| **Params** | **in** | **gid** key index 0-99 |
| **kbpk** kbpk value |
| **kbpkLen** kbpk length |
| **out** | **Nothing** |
| **return** | | **For details, see MfSdkPedPinpadStatus\_E** |
| **remark** | |  |
| **demo** | | s32 ret=0; u8 kbpk[] =  {  0x89,0xE8,0x8C,0xF7,0x93,0x14,0x44,0xF3,  0x34,0xBD,0x75,0x47,0xFC,0x3F,0x38,0x0C  };  ret = MfSdkPedTR31LoadKBPK(0, kbpk, sizeof(kbpk)); if(ret == MFSDK\_PED\_SUCCESS)  {  //load success  } |

## MfSdkPedDukptLoadEncryptedKeyByTR31

| **Prototype** | | **s32**  **MfSdkPedDukptLoadEncryptedKeyByTR31(MfSdkPedTr31KeyTy pe\_E type, s32 gid, u8 \*keyblock, u32 keyblockLen)** |
| --- | --- | --- |
| **Function** | | **Load cipher-text dukpt bdk/ipek by TR-31** |
| **Params** | **in** | **type** (0 - ipek, 1 - bdk) |
| **gid** key index 0-99 |
| **keyblock** keyblock value |
| **keyblockLen** keyblock length |
| **out** | **Nothing** |
| **return** | | **For details, see MfSdkPedPinpadStatus\_E** |



| **remark** |  |
| --- | --- |
|  | s32 ret=0; |
|  | u8\* keyblock = |
|  | "B0080B1TX00N0000D94B387D9FACD303A6AEE1DF9CECB7F8FA3D314DB43F7C0EDCBC62 |
|  | E94412D427"; |
| **demo** | ret = MfSdkPedDukptLoadEncryptedKeyByTR31(MFSDK\_PED\_TR31\_TYPE\_IPEK, 1, keyblock,  strlen(keyblock)); |
|  | if(ret == MFSDK\_PED\_SUCCESS) |
|  | { |
|  | //load success |
|  | } |

## MfSdkPedDukptLoadEncryptedKeyByTR31A

| **Prototype** | | **s32 MfSdkPedDukptLoadEncryptedKeyByTR31A(MfSdkPedTr31KeyT ype\_E type, s32 gid, u8 \*keyblock, u32 keyblockLen, u8**  **\*pKsn ,s32 ksnLength,u8 \*kvc ,s32 kvcLenth)** |
| --- | --- | --- |
| **Function** | | **Load cipher-text dukpt bdk/ipek by TR-31** |
| **Params** | **in** | **type** (0 - ipek, 1 - bdk) |
| **gid** key index 0-99 |
| **keyblock** keyblock value |
| **keyblockLen** keyblock length |
| **pKsn** |
| **ksnLength** |
| **kvcLenth** |
| **out** | **kvc** |
| **return** | | **For details, see MfSdkPedPinpadStatus\_E** |
| **remark** | | **Note pKsn != NULL && ksnLength > 0 will load pKsn** |
| **demo** | |  |

## MfSdkPedMkskSaveEncryptedKeyByTR31

| **Prototype** | **s32 MfSdkPedMkskSaveEncryptedKeyByTR31(s32 type, u32 gid, u8 \*keyblock, u32 keyblockLen ,u8 \*kvc ,s32 kvcLenth)** |
| --- | --- |
| **Function** | **Load cipher-text mk/wk by TR-31** |



| **Params** | **in** | **type** | MFSDK\_MKSK\_MAINKEY\_TYPE MFSDK\_MKSK\_PINENC\_TYPE MFSDK\_MKSK\_MACENC\_TYPE MFSDK\_MKSK\_MAGDEC\_TYPE  MFSDK\_MKSK\_TRANSKEY\_TYPE |
| --- | --- | --- | --- |
| **gid** | key index 0-99 |
| **keyblock** | keyblock value |
| **keyblockLen** | keyblock length |
| **kvcLenth** | kvc length |
| **out** | **kvc** | kvc buff |
| **return** | | **For details, see MfSdkPedPinpadStatus\_E** | |
| **remark** | |  | |
| **demo** | | s32 ret=0;  u8 kvc[32] = { 0 };  u8\* keyblock = "B0080B1TX00N0000D94B387D9FACD303A6AEE1DF9CECB7F8FA3D314DB43F7C0EDCBC62 E94412D427";  ret = MfSdkPedMkskSaveEncryptedKeyByTR31(MFSDK\_MKSK\_PINENC\_TYPE, 1, keyblock, strlen(keyblock), kvc, sizeof(kvc));  if(ret == MFSDK\_PED\_SUCCESS)  {  //load success  } | |

## MfSdkPedSetTransKey

| **Prototype** | | **s32 MfSdkPedSetTransKey(s32 gid, s8 \*key)** |
| --- | --- | --- |
| **Function** | | **Save the mksk trans key** |
| **Params** | **in** | **gid** max 100 group (0 ~ 99) |
| **key** trans key |
| **out** | **nothing** |
| **return** | | **For details, see MfSdkPedRet\_E** |
| **remark** | |  |
| **demo** | |  |



## MfSdkPedGetTransKeyKvc

| **Prototype** | | **s32 MfSdkPedGetTransKeyKvc(s32 gid, s8\* kvcBuf, s32 kvcBufLen)** |
| --- | --- | --- |
| **Function** | | **Save the mksk trans key** |
| **Params** | **in** | **gid** max 100 group (0 ~ 99) |
| **kvcBufLen** kvc buffer length |
| **out** | **kvcBuf** kvc(Length 4 bytes) |
| **return** | | **For details, see MfSdkPedRet\_E** |
| **remark** | |  |
| **demo** | |  |

## MfSdkPedSaveEncryptedMKByTransKey

| **Prototype** | | **s32 MfSdkPedSaveEncryptedMKByTransKey(u32 gid, u8\* key, MfSdkPedDesMod\_E mode, u8\* kvc)** |
| --- | --- | --- |
| **Function** | | **Save the encrypted main key by trans key** |
| **Params** | **in** | **gid** max 100 group (0 ~ 99) |
| **key** encrypted main key by trans key |
| **mode** des mode(see 'MfSdkPedDesMod\_E' enum) |
| **out** | **kvc** kvc(Length 4 bytes) |
| **return** | | **For details, see MfSdkPedRet\_E** |
| **remark** | |  |
| **demo** | |  |

## MfSdkPedDeleteTransKey

| **Prototype** | | **s32 MfSdkPedDeleteTransKey(s32 gid)** |
| --- | --- | --- |
| **Function** | | **Delete the mksk trans key** |
| **Params** | **in** | **gid** max 100 group (0 ~ 99) |



|  | **out** | **nothing** |
| --- | --- | --- |
| **return** | | **For details, see MfSdkPedRet\_E** |
| **remark** | |  |
| **demo** | |  |

## MfSdkPedRebootCmd

| **Prototype** | | **s32 MfSdkPedRebootCmd()** |
| --- | --- | --- |
| **Function** | | **Reboot the external pinpad** |
| **Params** | **in** | **nothing** |
| **out** | **nothing** |
| **return** | | **For details, see MfSdkPedRet\_E** |
| **remark** | |  |
| **demo** | |  |

## MfSdkPedGotoMainPage

| **Prototype** | | **s32 MfSdkPedGotoMainPage()** |
| --- | --- | --- |
| **Function** | | **External pinpad Back to the main page** |
| **Params** | **in** | **nothing** |
| **out** | **nothing** |
| **return** | | **For details, see MfSdkPedRet\_E** |
| **remark** | |  |
| **demo** | |  |

## MfSdkPedAesSavePlaintextKey

**s32 MfSdkPedAesSavePlaintextKey(s32 type, s32 gid, const u8 \* key,s32 length, u8\*kvc)**

****

| **Function** | | **Save the aes key** |
| --- | --- | --- |
| **Params** | **in** | MFSDK\_MKSK\_AES\_KEY\_TMK -  **type**  MFSDK\_MKSK\_AES\_KEY\_TMK2\_MAG |
| **gid** 0 - 49 |
| **key** keyblock value |
| **length** keyblock length |
| **out** | **kvc** |
| **return** | | **For details, see MfSdkPedRet\_E.** |
| **remark** | |  |
| **demo** | |  |

## MfSdkPedTr31LoadAesKey

| **Prototype** | | **s32 MfSdkPedTr31LoadAesKey(s32 gid, s32 keyType,const u8\* data, s32 size, u8 checkvalue[4])** |
| --- | --- | --- |
| **Function** | | **Three-level key system** |
| **Params** | **in** | **gid** 0 - 49 |
| MFSDK\_MKSK\_AES\_KEY\_TMK -  **keyType**  MFSDK\_MKSK\_AES\_KEY\_TMK2\_MAG |
| **data** |
| **size** |
| **out** | **checkvalue** |
| **return** | | **For details, see MfSdkPedRet\_E.** |
| **remark** | |  |
| **demo** | |  |

## MfSdkPedMkSkAesRun

| **Prototype** | | **s32 MfSdkPedMkSkAesRun(s32 type, s32 gid, MfSdkPedMod\_E mode, u8 \*ind, s32 size, u8 iv[16], u8 \*outd, MfSdkPedDesMod\_E**  **desMode)** |
| --- | --- | --- |
| **Function** | | **use key aes operation** |
| **Params** | **in** | **type** key type(0x08 - |



|  |  | MFSDK\_MKSK\_AES\_KEY\_PINENC, 0x09 - MFSDK\_MKSK\_AES\_KEY\_MACENC, 0x0A - MFSDK\_MKSK\_AES\_KEY\_MAGENC ...  MFSDK\_MKSK\_AES\_KEY\_TMK2\_MAG) |
| --- | --- | --- |
| **gid** key grouping,max 50 group (0 - 49) |
| **mode** mode(see 'MfSdkPedMod\_E' enum) |
| **ind** raw data |
| **size** data length (16-bytes multiple) |
| **iv** initialization vector |
| **desMode** des mode(see 'MfSdkPedDesMod\_E' enum) |
| **out** | **outd** calculation results |
| **return** | | **For details, see MfSdkPedRet\_E.** |
| **remark** | |  |
| **demo** | |  |

# 10Communication module

## Module description

This module mainly includes communication APIs.

## Module structure declaration

typedef struct

{

| char | ip[20]; /\*IP | address\*/ |
| --- | --- | --- |
| char | gateway[30]; | /\*Local gateway\*/ |
| char | mask[30]; | /\*Local subnet mask\*/ |
| char | dns[30]; | /\*DNS\*/ |



## Constant declarations





typedef enum

{

MFSDK\_COMM\_SOCKET\_INDEX\_MIN = 0, MFSDK\_COMM\_SOCKET\_INDEX\_0 =

MFSDK\_COMM\_SOCKET\_INDEX\_MIN, MFSDK\_COMM\_SOCKET\_INDEX\_1, MFSDK\_COMM\_SOCKET\_INDEX\_2, MFSDK\_COMM\_SOCKET\_INDEX\_3, MFSDK\_COMM\_SOCKET\_INDEX\_4, MFSDK\_COMM\_SOCKET\_INDEX\_MAX

}MfSdkCommSocketIndex\_E;

typedef enum

{

MFSDK\_COMM\_NET\_ONLY\_WIRELESS = 0, // only wireless 0 MFSDK\_COMM\_NET\_ONLY\_WIFI, // only wifi MFSDK\_COMM\_NET\_ONLY\_WIRELESS\_1, // only

wireless 1 (Device not supported)

MFSDK\_COMM\_NET\_ONLY\_ETHERNET, // ETHERNET

MFSDK\_COMM\_NET\_FIRST\_WIRELESS, // wireless

priority

MFSDK\_COMM\_NET\_FIRST\_WIFI, // wifi priority

}MfSdkCommNetSelect\_E;

typedef MfSdkCommNetSelect\_E MfSdkCommMode\_E;

typedef enum

{

MFSDK\_COMM\_UART\_COM0 = 0, MFSDK\_COMM\_UART\_COM1, MFSDK\_COMM\_UART\_COM2, MFSDK\_COMM\_UART\_COM3, MFSDK\_COMM\_UART\_COM4, MFSDK\_COMM\_UART\_COM5, MFSDK\_COMM\_UART\_COM6, MFSDK\_COMM\_UART\_COM7, MFSDK\_COMM\_UART\_COM8, MFSDK\_COMM\_UART\_COM9, MFSDK\_COMM\_UART\_COM10, MFSDK\_COMM\_UART\_COM11, MFSDK\_COMM\_UART\_COM12, MFSDK\_COMM\_UART\_COM13, MFSDK\_COMM\_UART\_COM14,



MFSDK\_COMM\_UART\_COM15, MFSDK\_COMM\_UART\_COM16, MFSDK\_COMM\_UART\_COM17, MFSDK\_COMM\_UART\_COM18, MFSDK\_COMM\_UART\_COM19, MFSDK\_COMM\_UART\_COM20, MFSDK\_COMM\_UART\_COM21, MFSDK\_COMM\_UART\_COM22, MFSDK\_COMM\_UART\_COM23, MFSDK\_COMM\_UART\_COM24, MFSDK\_COMM\_UART\_COM25, MFSDK\_COMM\_UART\_COM26, MFSDK\_COMM\_UART\_COM27, MFSDK\_COMM\_UART\_COM28, MFSDK\_COMM\_UART\_COM29, MFSDK\_COMM\_UART\_COM30,

}MfSdkCommUart\_E;

typedef enum

{

MFSDK\_COMM\_RET\_FAILED = -3, //failed MFSDK\_COMM\_RET\_BOUNDS = -2, //Array out-of-bounds MFSDK\_COMM\_RET\_PARM\_ERROR = -1, //check param MFSDK\_COMM\_RET\_OK = 0,

}MfSdkCommRet\_E;

typedef enum

{

MFSDK\_COMM\_ATCCPIN\_FAIL = 0, //Failure MFSDK\_COMM\_ATCCPIN\_LOCKED, //locked MFSDK\_COMM\_ATCCPIN\_NORMAL, //noraml MFSDK\_COMM\_ATCCPIN\_PUK //Puk

}MfSdkCommAtcCpin\_E;

typedef enum

{

MFSDK\_COMM\_WIFI\_STATE\_CONNECT\_PROCESS = -999,

MFSDK\_COMM\_WIFI\_STATE\_DISCONNECT\_PROCESS = -998,

MFSDK\_COMM\_WIFI\_STATE\_CONNECT\_START = -997,

MFSDK\_COMM\_WIFI\_STATE\_NOT\_EXIST = -2,

MFSDK\_COMM\_WIFI\_STATE\_PWD\_ERROR = -1,

MFSDK\_COMM\_WIFI\_STATE\_DISCONNECT = 0,

MFSDK\_COMM\_WIFI\_STATE\_CONNECT = 1,



}MfSdkCommWifiState\_E;

typedef enum

{

MFSDK\_COMM\_MOBILE\_NO\_SIGNAL = 0,

MFSDK\_COMM\_MOBILE\_1G, //The 1st Generation Mobile Communication System

MFSDK\_COMM\_MOBILE\_2G, //The 2nd Generation Mobile Communication System

MFSDK\_COMM\_MOBILE\_3G, //The 3rd Generation Mobile Communication System

MFSDK\_COMM\_MOBILE\_4G, //The 4th Generation Mobile Communication System

MFSDK\_COMM\_MOBILE\_5G,//RFU The 5th Generation Mobile Communication System

MFSDK\_COMM\_MOBILE\_6G,//RFU The 6th Generation Mobile Communication System

}MfSdkCommMobileGen\_E;

typedef enum

{

MFSDK\_COMM\_IPV4 = 1, //only ipv4 MFSDK\_COMM\_IPV6, //only ipv6 MFSDK\_COMM\_IPV4\_IPV6, // both ipv4 and ipv6

}MfSdkCommIpVer\_E;

typedef enum

{

MFSDK\_COMM\_ATC\_OFF = 0, // turn off MFSDK\_COMM\_ATC\_ON, //turn on

}MfSdkCommAtcPowerState\_E;

typedef enum

{

MFSDK\_COMM\_SIM\_PROC\_PPP\_NETWORK\_OK = 0, MFSDK\_COMM\_SIM\_PROC\_NETWORK\_CONNECTED, MFSDK\_COMM\_SIM\_PROC\_NETWORK\_REGISTER\_NETOWK, MFSDK\_COMM\_SIM\_PROC\_SIM\_NORMAL, MFSDK\_COMM\_SIM\_PROC\_SWITCHING\_SIM, MFSDK\_COMM\_SIM\_PROC\_NO\_SIM\_CARD,

}MfSdkCommSimProcStatus\_E;



typedef enum

{

MFSDK\_COMM\_IPTYPE\_V4, // Get IP address type IPv4 MFSDK\_COMM\_IPTYPE\_V6, // Get IP address type IPv6

}MfSdkCommIpType\_E;

## MfSdkCommLinkState

| **Prototype** | | **LIB\_EXPORT MFSDKBOOL MfSdkCommLinkState(void)** |
| --- | --- | --- |
| **Function** | | **Get net link state for both GPRS and Wifi** |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **1 Linked** |
| **0 Disconnect** |
| **remark** | |  |
| **demo** | | MFSDKBOOL linkState = MfSdkCommLinkState(); |

## MfSdkCommGetNetMode

| **Prototype** | | **LIB\_EXPORT MfSdkCommMode\_E MfSdkCommGetNetMode(void)** |
| --- | --- | --- |
| **Function** | | **Get Comm mode, Ref.MfSdkCommMode\_E** |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **NetMode Ref.MfSdkCommMode\_E** |
| **remark** | |  |
| **demo** | | s32 netMode = MfSdkCommGetNetMode(); |

## MfSdkCommAtcCell

| **Prototype** | **LIB\_EXPORT s32 MfSdkCommAtcCell(s32 index)** |
| --- | --- |



| **Function** | | **get 1st net registered cell** |
| --- | --- | --- |
| **Params** | **in** | **index** |
| **out** | **None** |
| **return** | | **Cell ID** |
| **remark** | |  |
| **demo** | |  |

## MfSdkCommAtcCellInfor

| **Prototype** | | **LIB\_EXPORT s32 MfSdkCommAtcCellInfor(s32 \*cid, s32 size, s32**  **\*num)** |
| --- | --- | --- |
| **Function** | | **get net registered cell** |
| **Params** | **in** | **cid** buffer |
| **size** buffer size |
| **out** | **num** |
| **return** | | **MFSDK\_COMM\_RE**  **Success**  **T\_OK** |
| **Other Fail** |
| **remark** | |  |
| **demo** | |  |

## MfSdkCommAtcCpin

| **Prototype** | | **LIB\_EXPORT MfSdkCommAtcCpin\_E MfSdkCommAtcCpin(s32 index)** |
| --- | --- | --- |
| **Function** | | **get net registered cell** |
| **Params** | **in** | **index** 0~1 |
| **out** | **None** |
| **return** | | **MFSDK\_COMM\_ATCCPIN\_FAI**  **Fail**  **L** |
| **MFSDK\_COMM\_ATCCPIN\_LO**  **Locked**  **CKED** |
| **MFSDK\_COMM\_ATCCPIN\_NO Normal** |



|  | **RMAL** |
| --- | --- |
| **MFSDK\_COMM\_ATCCPIN\_PU**  **Puk**  **K** |
| **remark** |  |
| **demo** |  |

## MfSdkCommAtcGetLocalIp

| **Prototype** | | **LIB\_EXPORT s32 MfSdkCommAtcGetLocalIp(s8\* ip)** |
| --- | --- | --- |
| **Function** | | **Get local ip** |
| **Params** | **in** | **None** |
| **out** | **ip** |
| **return** | | **MFSDK\_COMM\_RE**  **Success**  **T\_OK** |
| **Other Fail** |
| **remark** | |  |
| **demo** | | s8 ip[32] = {0};  s32 ret = MfSdkCommAtcGetLocalIp(ip); |

## MfSdkCommAtcIccid

| **Prototype** | | **LIB\_EXPORT const s8\* MfSdkCommAtcIccid(s32 index)** |
| --- | --- | --- |
| **Function** | | **Get index Integrate circuit card identity value** |
| **Params** | **in** | **index** 0~1 |
| **out** | **None** |
| **return** | | **Icc ID** |
| **remark** | |  |
| **demo** | |  |



## MfSdkCommAtcImei

| **Prototype** | | **LIB\_EXPORT const s8\* MfSdkCommAtcImei(s32 index)** |
| --- | --- | --- |
| **Function** | | **Get Module imei** |
| **Params** | **in** | **index** 0~1 |
| **out** | **None** |
| **return** | | **imei value** |
| **remark** | |  |
| **demo** | |  |

## MfSdkCommAtcImsi

| **Prototype** | | **LIB\_EXPORT const s8\* MfSdkCommAtcImsi(s32 index)** |
| --- | --- | --- |
| **Function** | | **Get Module imsi** |
| **Params** | **in** | **index** 0~1 |
| **out** | **None** |
| **return** | | **imsi value** |
| **remark** | |  |
| **demo** | |  |

## MfSdkCommAtcLac

| **Prototype** | | **LIB\_EXPORT s32 MfSdkCommAtcLac(s32 index)** |
| --- | --- | --- |
| **Function** | | **Get net registered lac** |
| **Params** | **in** | **index** GPRS/4G index 0-1 |
| **out** | **None** |
| **return** | | **lac value** |
| **remark** | |  |



**demo**

## MfSdkCommAtcGetMcc

| **Prototype** | | **LIB\_EXPORT s32 MfSdkCommAtcGetMcc(s32 index)** |
| --- | --- | --- |
| **Function** | | **Get gprs MCC** |
| **Params** | **in** | **index** use 0 |
| **out** | **None** |
| **return** | | **lac value** |
| **remark** | |  |
| **demo** | | s32 mcc = MfSdkCommAtcGetMcc(0); |

## MfSdkCommAtcGetMnc

| **Prototype** | | **LIB\_EXPORT s32 MfSdkCommAtcGetMnc(s32 index)** |
| --- | --- | --- |
| **Function** | | **Get gprs MNC** |
| **Params** | **in** | **index** use 0 |
| **out** | **None** |
| **return** | | **lac value** |
| **remark** | |  |
| **demo** | | s32 mnc = MfSdkCommAtcGetMnc(0); |

## MfSdkCommAtcLacInfor

| **Prototype** | | **LIB\_EXPORT s32 MfSdkCommAtcLacInfor(s32 \*lac, s32 size, s32**  **\*num)** |
| --- | --- | --- |
| **Function** | | **Get net registered lac** |
| **Params** | **in** | **lac** buffer |
| **size** buffer size |
| **out** | **num** lac num |



| **return** | **MFSDK\_COMM\_RE**  **Success**  **T\_OK** |
| --- | --- |
| **Other Fail** |
| **remark** |  |
| **demo** |  |

## MfSdkCommAtcSignal

| **Prototype** | | **LIB\_EXPORT s32 MfSdkCommAtcSignal(s32 index)** |
| --- | --- | --- |
| **Function** | | **Get Module signal** |
| **Params** | **in** | **index** GPRS/4G index 0-1 |
| **out** | **None** |
| **return** | | **0 Unregistered** |
| **1 - 4 Signal, e.g. 1:1 grid signal** |
| **remark** | |  |
| **demo** | |  |

## MfSdkCommGetAtcGeneration

| **Prototype** | | **LIB\_EXPORT MfSdkCommMobileGen\_E MfSdkCommGetAtcGeneration(s32 index)** |
| --- | --- | --- |
| **Function** | | **Get the atc generation** |
| **Params** | **in** | **index** sim index 0-1 (Single sim card select 0) |
| **out** | **None** |
| **return** | | **Ref. MfSdkCommMobileGen\_E** |
| **remark** | |  |
| **demo** | |  |

## MfSdkCommGetAtcPower

**LIB\_EXPORT MfSdkCommAtcPowerState\_E MfSdkCommGetAtcPower(void)**

****

| **Function** | | **Get the atc power** |
| --- | --- | --- |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **Ref. MfSdkCommAtcPowerState\_E** |
| **remark** | |  |
| **demo** | |  |

## MfSdkCommAtcSendCmd

| **Prototype** | | **LIB\_EXPORT s32 MfSdkCommAtcSendCmd(s8 \* cmd,s8 \* RecvData,s32 RecvSize,s32 timeout,s32 count)** |
| --- | --- | --- |
| **Function** | | **Send and recv AT commend (4G module)** |
| **Params** | **in** | **cmd** AT cmd |
| **RecvSize** size of recv buff |
| **timeout** Timeout unit: ms |
| **count** resend count |
| **out** | **RecvData** recv buff |
| **return** | | **0 success** |
| **other fail** |
| **remark** | |  |
| **demo** | |  |

## MfSdkCommGetNetSelect

| **Prototype** | | **LIB\_EXPORT MfSdkCommNetSelect\_E MfSdkCommGetNetSelect(void)** |
| --- | --- | --- |
| **Function** | | **Get mode of wifi or gprs or the Priority** |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **0 WIRELESS** |
| **1 WIFI** |
| **Other Ref.MfSdkCommNetSelect\_E** |



| **remark** |  |
| --- | --- |
| **demo** |  |

## MfSdkCommGetOperateId

| **Prototype** | | **LIB\_EXPORT s32 MfSdkCommGetOperateId(void)** |
| --- | --- | --- |
| **Function** | | **Read the id of the working sim card(Only devices with two sim cards are supported)** |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **0 SIM 1** |
| **1 SIM 2** |
| **Other Fail** |
| **remark** | |  |
| **demo** | |  |

## MfSdkCommGetWifiPower

| **Prototype** | | **LIB\_EXPORT s32 MfSdkCommGetWifiPower(void)** |
| --- | --- | --- |
| **Function** | | **Get the WiFi power** |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **0 Power OFF** |
| **1 Power ON** |
| **remark** | |  |
| **demo** | |  |

## MfSdkCommGsmGetSignal

**Prototype s32 MfSdkCommGsmGetSignal(void)**

****

| **Function** | | **Get gsm signal** |
| --- | --- | --- |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | |  |
| **remark** | |  |
| **demo** | |  |

## MfSdkCommHttpDownload

| **Prototype** | | **LIB\_EXPORT s32 MfSdkCommHttpDownload(s8 \*url, s8\*fullpathfilename, s32 iscontinue, s32 nRetry)** |
| --- | --- | --- |
| **Function** | | **HTTP download** |
| **Params** | **in** | **url** url address |
| **fullpathfilename** The full name of the downloaded file |
| **iscontinue** 1-sequel 0-download again |
| **nRetry** retry count |
| **out** | **None** |
| **return** | | **MFSDK\_COMM\_RE**  **Success**  **T\_OK** |
| **Other Fail** |
| **remark** | |  |
| **demo** | |  |

## MfSdkCommMbedtlsInit

| **Prototype** | | **LIB\_EXPORT s32 MfSdkCommMbedtlsInit(s32 useSession, s32 logLevel)** |
| --- | --- | --- |
| **Function** | | **if you use mbedtls ,need init mbedtls** |
| **Params** | **in** | **useSession** |
| **logLevel** |
| **out** | **None** |
| **return** | | **MFSDK\_COMM\_RE**  **Success**  **T\_OK** |



|  | **Other Ref. MfSdkCommRet\_E** |
| --- | --- |
| **remark** |  |
| **demo** |  |

## MfSdkCommNetLink

| **Prototype** | | **LIB\_EXPORT s32 MfSdkCommNetLink(void \* pfunc, char \* apn, s32 timeover)** |
| --- | --- | --- |
| **Function** | | Connect Network |
| **Params** | **in** | **pfunc** Callback |
| **apn** GPRS apn |
| **timeover** Connection timeout |
| **out** | **None** |
| **return** | | **MFSDK\_COMM\_RE**  **Success**  **T\_OK** |
| **Other Fail** |
| **remark** | |  |
| **demo** | |  |

## MfSdkCommNetLinkWithUserInfo

| **Prototype** | | **LIB\_EXPORT s32 MfSdkCommNetLinkWithUserInfo( char \* apn, char \*user, char \*pwd)** |
| --- | --- | --- |
| **Function** | | Connect Network |
| **Params** | **in** | **apn** GPRS apn |
| **user** GPRS apn user id |
| **pwd** GPRS apn user password |
| **out** | **None** |
| **return** | | **MFSDK\_COMM\_RE**  **Success**  **T\_OK** |
| **Other Fail** |
| **remark** | |  |
| **demo** | |  |



## MfSdkCommNetUnlink

| **Prototype** | | **LIB\_EXPORT s32 MfSdkCommNetUnlink(void)** |
| --- | --- | --- |
| **Function** | | **Disconnect from the network** |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **MFSDK\_COMM\_RE**  **Success**  **T\_OK** |
| **Other Fail** |
| **remark** | |  |
| **demo** | |  |

## MfSdkCommSetApMode

| **Prototype** | | **LIB\_EXPORT s32 MfSdkCommSetApMode(MfSdkCommIpVer\_E mode)** |
| --- | --- | --- |
| **Function** | | **set IPV version, should be called in main.c before network initial** |
| **Params** | **in** | **mode** 1:only IPV4; 2:only IPV6; 3:IPV4&IPV6 |
| **out** | **None** |
| **return** | | **MFSDK\_COMM\_RE**  **Success**  **T\_OK** |
| **Other Ref. MfSdkCommRet\_E** |
| **remark** | |  |
| **demo** | |  |

## MfSdkCommSetApnList

| **Prototype** | **LIB\_EXPORT s32**  **MfSdkCommSetApnList(MfSdkCommApnInfo\_T apn\_list[], s32 ncount)** |
| --- | --- |
| **Function** |  |



| **Params** | **in** | **apn\_list[]** APN operator corresponding list |
| --- | --- | --- |
| **ncount** APN count |
| **out** | **None** |
| **return** | | **MFSDK\_COMM\_RE**  **Success**  **T\_OK** |
| **Other Ref. MfSdkCommRet\_E** |
| **remark** | |  |
| **demo** | |  |

## MfSdkCommSetApnListMcc

| **Prototype** | | **LIB\_EXPORT s32**  **MfSdkCommSetApnListMcc(MfSdkCommApinfoMcc\_T\*apnlistMc c, s32 count)** |
| --- | --- | --- |
| **Function** | | Set APN MCC list. |
| **Params** | **in** | **apn\_list[]** APN operator corresponding list |
| **ncount** APN count |
| **out** | **None** |
| **return** | | **MFSDK\_COMM\_RE**  **Success**  **T\_OK** |
| **Other Ref. MfSdkCommRet\_E** |
| **remark** | |  |
| **demo** | |  |

## MfSdkCommSetAtcPower

| **Prototype** | | **s32 MfSdkCommSetAtcPower(MfSdkCommAtcPowerState\_E flag)** |
| --- | --- | --- |
| **Function** | | Set the atc power |
| **Params** | **in** | 1. - Turn off atc,   **flag**   1. - Turn on atc |
| **out** | **None** |
| **return** | | **MFSDK\_COMM\_RE**  **Success**  **T\_OK** |
| **Other Ref. MfSdkCommRet\_E** |



| **remark** |  |
| --- | --- |
| **demo** |  |

## MfSdkCommSetInitApn

| **Prototype** | | **LIB\_EXPORT s32 MfSdkCommSetInitApn(s8 \*apn, s8 \*user, s8**  **\*pwd)** |
| --- | --- | --- |
| **Function** | | set gprs apn |
| **Params** | **in** | **apn** GPRS APN |
| **user** GPRS USER |
| **pwd** GRPS PASSWORD |
| **out** | **None** |
| **return** | | **MFSDK\_COMM\_RE**  **Success**  **T\_OK** |
| **Other Ref. MfSdkCommRet\_E** |
| **remark** | |  |
| **demo** | |  |

## MfSdkCommSetNetMode

| **Prototype** | | **LIB\_EXPORT s32 MfSdkCommSetNetMode(s32 mode)** |
| --- | --- | --- |
| **Function** | | Switch network switching mode |
| **Params** | **in** | **mode** Ref. MfSdkCommNetSelect\_E |
| **out** | **None** |
| **return** | | **MFSDK\_COMM\_RE**  **Success**  **T\_OK** |
| **Other Ref. MfSdkCommRet\_E** |
| **remark** | |  |
| **demo** | |  |



## MfSdkCommSetNetSelect

| **Prototype** | | **LIB\_EXPORT s32**  **MfSdkCommSetNetSelect(MfSdkCommNetSelect\_E index)** |
| --- | --- | --- |
| **Function** | |  |
| **Params** | **in** | **index** Ref. MfSdkCommNetSelect\_E |
| **out** | **None** |
| **return** | | **MFSDK\_COMM\_RE**  **Success**  **T\_OK** |
| **Other Ref. MfSdkCommRet\_E** |
| **remark** | |  |
| **demo** | |  |

## MfSdkCommSetWifiName

| **Prototype** | | **LIB\_EXPORT s32 MfSdkCommSetWifiName(s32 mode)** |
| --- | --- | --- |
| **Function** | | set wifi name, (WiFi version V5.1.2 and above support modify the wifi name) |
| **Params** | **in** | **mode** mode = 0: KM\_AP, default: KM\_(SN) |
| **out** | **None** |
| **return** | | **MFSDK\_COMM\_RE**  **Success**  **T\_OK** |
| **Other Ref. MfSdkCommRet\_E** |
| **remark** | |  |
| **demo** | |  |

## MfSdkCommSocketClose

| **Prototype** | **LIB\_EXPORT s32**  **MfSdkCommSocketClose(MfSdkCommSocketIndex\_E index)** |
| --- | --- |
| **Function** | Disconnect the server |



| **Params** | **in** | **index** Sock index (0-4) |
| --- | --- | --- |
| **out** | **None** |
| **return** | | **MFSDK\_COMM\_RE**  **Success**  **T\_OK** |
| **Other Ref. MfSdkCommRet\_E** |
| **remark** | |  |
| **demo** | |  |

## MfSdkCommSocketConnect

| **Prototype** | | **LIB\_EXPORT s32**  **MfSdkCommSocketConnect(MfSdkCommSocketIndex\_E index, char \* ip, s32 port, s32 timeout, void \*func)** | |
| --- | --- | --- | --- |
| **Function** | | Connect to the server | |
| **Params** | **in** | **index** | Sock index (0-4) |
| **ip** | IP address |
| **port** | Port number |
| **timeout** | time-out period unit:ms |
| **func** | callback function |
| **out** | **None** | |
| **return** | | **>=0** | **Success** |
| **<0** | **Fail** |
| **remark** | |  | |
| **demo** | |  | |

## MfSdkCommSocketConnectPriority

| **Prototype** | | **LIB\_EXPORT s32 MfSdkCommSocketConnectPriority(s32 mode)** |
| --- | --- | --- |
| **Function** | | Set connect priority |
| **Params** | **in** | mode=0(priority ipv4),  **mode**  mode=1(priority ipv6) |
| **out** | **None** |
| **return** | | **MFSDK\_COMM\_RE**  **Success**  **T\_OK** |



|  | **Other Ref. MfSdkCommRet\_E** |
| --- | --- |
| **remark** |  |
| **demo** |  |

## MfSdkCommSocketCreate

| **Prototype** | | **LIB\_EXPORT s32**  **MfSdkCommSocketCreate(MfSdkCommSocketIndex\_E index)** |
| --- | --- | --- |
| **Function** | | Create socket |
| **Params** | **in** | **index** Sock index (0-4) |
| **out** | **None** |
| **return** | | **>=0 Success** |
| **<0 Fail** |
| **remark** | |  |
| **demo** | |  |

## MfSdkCommSocketFifoResize

| **Prototype** | | **LIB\_EXPORT s32 MfSdkCommSocketFifoResize(s32 index, s32 size)** |
| --- | --- | --- |
| **Function** | |  |
| **Params** | **in** | **index** Sock index (0-4) |
| **size** |
| **out** | **None** |
| **return** | | **>=0 Success** |
| **<0 Fail** |
| **remark** | |  |
| **demo** | |  |

## MfSdkCommSocketRecv

**LIB\_EXPORT s32**

**MfSdkCommSocketRecv(MfSdkCommSocketIndex\_E index,**

|  | | **unsigned char \* buff, s32 len, u32 timeover)** |
| --- | --- | --- |
| **Function** | | Receive data |
| **Params** | **in** | **index** Sock index (0-4)  **timeover** Receive timeout |
| **out** | **buff** Receive buffer |
| **len** Receive length |
| **return** | | **>0 Receive length** |
| **0 Timeout** |
| **-1 Network disconnection** |
| **remark** | |  |
| **demo** | |  |

## MfSdkCommSocketSend

| **Prototype** | | **LIB\_EXPORT s32**  **MfSdkCommSocketSend(MfSdkCommSocketIndex\_E index, u8\* buff, s32 size)** |
| --- | --- | --- |
| **Function** | | Send data |
| **Params** | **in** | **index** Sock index (0-4) |
| **buff** Send buff |
| **len** Send length |
| **out** | **None** |
| **return** | | **>=0 Success** |
| **<0 Fail** |
| **remark** | |  |
| **demo** | |  |

## MfSdkCommSslAuthMode



### Prototype

**LIB\_EXPORT s32**

**MfSdkCommSslAuthMode(MfSdkCommSocketIndex\_E index, s32 authMode)**

****

| **Function** | | Set the certificate verification mode. |
| --- | --- | --- |
| **Params** | **in** | **index** Sock index (0-4) |
| Ref.  MBEDTLS\_SSL\_VERIFY\_NONE  **authMode** MBEDTLS\_SSL\_VERIFY\_OPTIONAL MBEDTLS\_SSL\_VERIFY\_REQUIRED MBEDTLS\_SSL\_VERIFY\_UNSET |
| **out** | **None** |
| **return** | | **MFSDK\_COMM\_RE**  **Success**  **T\_OK** |
| **Other Ref. MfSdkCommRet\_E** |
| **remark** | |  |
| **demo** | | s32 sock = MfSdkCommSocketCreate(MFSDK\_COMM\_SOCKET\_INDEX\_1); MfSdkCommSslAuthMode(sock,MBEDTLS\_SSL\_VERIFY\_OPTIONAL); MfSdkCommSslInit(sock, "xxxx\\ca.pem", "xxxx\\cli.crt", "xxxx\\pri.key", 1); s32 ret = MfSdkCommSslConnect(sock,  "a2d911mzqj2e50-ats.iot.ap-south-1.amazonaws.com", 8883,NULL);  if(ret != MFSDK\_COMM\_RET\_OK) {MfSdkCommSocketClose(sock); } |

## MfSdkCommSslClose

| **Prototype** | | **LIB\_EXPORT s32**  **MfSdkCommSslClose(MfSdkCommSocketIndex\_E index)** |
| --- | --- | --- |
| **Function** | | Close handle |
| **Params** | **in** | **index** Sock index (0-4) |
| **out** | **None** |
| **return** | | **MFSDK\_COMM\_RE**  **Success**  **T\_OK** |
| **Other Ref. MfSdkCommRet\_E** |
| **remark** | |  |
| **demo** | | s32 sock = MfSdkCommSocketCreate(MFSDK\_COMM\_SOCKET\_INDEX\_1); MfSdkCommSslAuthMode(sock,MBEDTLS\_SSL\_VERIFY\_OPTIONAL); MfSdkCommSslInit(sock, "xxxx\\ca.pem", "xxxx\\cli.crt", "xxxx\\pri.key", 1); s32 ret = MfSdkCommSslConnect(sock,  "a2d911mzqj2e50-ats.iot.ap-south-1.amazonaws.com", 8883,NULL);  if(ret != MFSDK\_COMM\_RET\_OK) {MfSdkCommSocketClose(sock); } |



## MfSdkCommSslConnect

| **Prototype** | | **LIB\_EXPORT s32**  **MfSdkCommSslConnect(MfSdkCommSocketIndex\_E index, char**  **\* ip, s32 port, void \*func)** |
| --- | --- | --- |
| **Function** | | Connect to the server |
| **Params** | **in** | **index** Sock index (0-4) |
| **ip** IP address |
| **port** Port number |
| callback function  **func**  int (\*connect\_server\_func)(void); |
| **out** | **None** |
| **return** | | **MFSDK\_COMM\_RET\_**  **Success**  **OK** |
| **MFSDK\_COMM\_RET\_F**  **Fail**  **AILED** |
| **remark** | |  |
| **demo** | | s32 sock = MfSdkCommSocketCreate(MFSDK\_COMM\_SOCKET\_INDEX\_1); MfSdkCommSslAuthMode(sock,MBEDTLS\_SSL\_VERIFY\_OPTIONAL); MfSdkCommSslInit(sock, "xxxx\\ca.pem", "xxxx\\cli.crt", "xxxx\\pri.key", 1); s32 ret = MfSdkCommSslConnect(sock,  "a2d911mzqj2e50-ats.iot.ap-south-1.amazonaws.com", 8883,NULL);  if(ret != MFSDK\_COMM\_RET\_OK) {MfSdkCommSocketClose(sock); } |

## MfSdkCommSslInit

| **Prototype** | | **LIB\_EXPORT s32 MfSdkCommSslInit(MfSdkCommSocketIndex\_E index, char \* cacert, char \* clientcert, char \* clientkey, s32 level)** |
| --- | --- | --- |
| **Function** | |  |
| **Params** | **in** | **index** Sock index (0-4) |
| **cacert** Cacert path |
| **clientcert** Clientcert path |
| **clientkey** Clientkey path |
| **level** Log level default value is 1 |
| **out** | **None** |
| **return** | | **MFSDK\_COMM\_RET\_ Success** |



|  | **OK** |
| --- | --- |
| **MFSDK\_COMM\_RET\_F**  **Fail**  **AILED** |
| **remark** |  |
| **demo** | s32 sock = MfSdkCommSocketCreate(MFSDK\_COMM\_SOCKET\_INDEX\_1); MfSdkCommSslAuthMode(sock,MBEDTLS\_SSL\_VERIFY\_OPTIONAL); MfSdkCommSslInit(sock, "xxxx\\ca.pem", "xxxx\\cli.crt", "xxxx\\pri.key", 1); s32 ret = MfSdkCommSslConnect(sock,  "a2d911mzqj2e50-ats.iot.ap-south-1.amazonaws.com", 8883,NULL);  if(ret != MFSDK\_COMM\_RET\_OK) {MfSdkCommSocketClose(sock); } |

## MfSdkCommSslSetHostname

| **Prototype** | | **LIB\_EXPORT s32**  **MfSdkCommSslSetHostname(MfSdkCommSocketIndex\_E index , const char \* cn)** |
| --- | --- | --- |
| **Function** | | Set ssl host name |
| **Params** | **in** | **index** Sock index (1-4) |
| **out** | **cn** Host name |
| **return** | | **MFSDK\_COMM\_RE**  **success**  **T\_OK** |
| **MFSDK\_COMM\_RE**  **failed**  **T\_FAILED** |
| **remark** | |  |
| **demo** | |  |

## MfSdkCommSslMbedtls

| **Prototype** | | **LIB\_EXPORT s32 MfSdkCommSslMbedtls(s32 mbedtls)** |
| --- | --- | --- |
| **Function** | | Whether to use mbedtls. |
| **Params** | **in** | **mbedtls** 1 - enable 0-disable |
| **out** | **None** |
| **return** | | **MFSDK\_COMM\_RE**  **success**  **T\_OK** |



| **remark** |  |
| --- | --- |
| **demo** | MfSdkCommSslMbedtls(1); |

## MfSdkCommSslRecv

| **Prototype** | | **LIB\_EXPORT s32**  **MfSdkCommSslRecv(MfSdkCommSocketIndex\_E index, char \* pdata, s32 size)** |
| --- | --- | --- |
| **Function** | | Use new API MfSdkCommSocketRecv pls. |
| **Params** | **in** | **index** Sock index (0-4) |
| **size** |
| **out** | **pdata** recv data |
| **return** | | **>=0 Success** |
| **<0 Fail** |
| **remark** | |  |
| **demo** | |  |

## MfSdkCommSslSend

| **Prototype** | | **LIB\_EXPORT s32**  **MfSdkCommSslSend(MfSdkCommSocketIndex\_E index, char \* pdata, s32 size)** |
| --- | --- | --- |
| **Function** | | Use new API MfSdkCommSocketSend pls. |
| **Params** | **in** | **index** Sock index (0-4) |
| **size** send data size |
| **pdata** send data |
| **out** |  |
| **return** | |  |
| **remark** | |  |
| **demo** | |  |



## MfSdkCommSslSend2

| **Prototype** | | **LIB\_EXPORT s32 MfSdkCommSslSend2(s32 index, char \* pdata, s32 size)** |
| --- | --- | --- |
| **Function** | | Use new API MfSdkCommSocketSend pls. |
| **Params** | **in** | **index** Sock index (0-4) |
| **size** send data size |
| **pdata** send data |
| **out** |  |
| **return** | |  |
| **remark** | |  |
| **demo** | |  |

## MfSdkCommUartClear

| **Prototype** | | **LIB\_EXPORT s32 MfSdkCommUartClear(MfSdkCommUart\_E nCom)** |
| --- | --- | --- |
| **Function** | | Clear uart fifo buff |
| **Params** | **in** | **nCom** For details, see MfSdkCommUart\_E. |
| **out** | **None** |
| **return** | | **MFSDK\_COMM\_RE**  success  **T\_OK** |
| **other** fail |
| **remark** | |  |
| **demo** | | if(MfSdkCommUartClear(MFSDK\_COMM\_UART\_COM10)==MFSDK\_COMM\_RE T\_OK)  {  //clear succ  } |



## MfSdkCommUartClose

| **Prototype** | | **LIB\_EXPORT s32 MfSdkCommUartClose(MfSdkCommUart\_E nCom)** |
| --- | --- | --- |
| **Function** | | close uart port |
| **Params** | **in** | **nCom** For details, see MfSdkCommUart\_E. |
| **out** | **None** |
| **return** | | **MFSDK\_COMM\_RE**  success  **T\_OK** |
| **other** fail |
| **remark** | |  |
| **demo** | | if(MfSdkCommUartClose(MFSDK\_COMM\_UART\_COM10)==MFSDK\_COMM\_R ET\_OK)  {  //close succ  } |

## MfSdkCommUartGetRxBufLength

| **Prototype** | | **LIB\_EXPORT s32**  **MfSdkCommUartGetRxBufLength(MfSdkCommUart\_E nCom)** |
| --- | --- | --- |
| **Function** | | Get receive buff len |
| **Params** | **in** | **nCom** For details, see MfSdkCommUart\_E. |
| **out** | **None** |
| **return** | | **>=0** receive buff len |
| **remark** | |  |
| **demo** | | if(MfSdkCommUartGetRxBufLength(MFSDK\_COMM\_UART\_COM10) > 0)  {  //TODO RECV  } |



## MfSdkCommUartOpen

| **Prototype** | | **LIB\_EXPORT s32 MfSdkCommUartOpen(MfSdkCommUart\_E nCom, u32 nBaud,u32 nData,u32 nStop,u32 nParity)** |
| --- | --- | --- |
| **Function** | | open uart port |
| **Params** | **in** | **nCom** For details, see MfSdkCommUart\_E. |
| **nBaud** Baud rate |
| **nData** Data bits |
| **nStop** Stop bits |
| **nParity** Parity |
| **out** | **None** |
| **return** | | **MFSDK\_COMM\_RE**  success  **T\_OK** |
| **remark** | |  |
| **demo** | | s32 ret = MfSdkCommUartOpen(MFSDK\_COMM\_UART\_COM10, 115200, 8, 0,  0);  if(ret == MFSDK\_COMM\_RET\_OK)  {  //TODO  } |

## MfSdkCommUartRecv

| **Prototype** | | **LIB\_EXPORT s32 MfSdkCommUartRecv(MfSdkCommUart\_E nCom, u8\* pBuffer, u32 nLength, s32 timeout)** |
| --- | --- | --- |
| **Function** | | receive uart port data |
| **Params** | **in** | **nCom** For details, see MfSdkCommUart\_E. |
| **nLength** recv length |
| **timeout** time-out period |
| **out** | **pBuffer** recv buffer |
| **return** | | **>=0** receive data len |
| **remark** | |  |
| **demo** | | u8 recvBuf[512] = {0}; if(MfSdkCommUartRecv(MFSDK\_COMM\_UART\_COM10,recvBuf,512,10000)>0)  { |



//receive succ

}

## MfSdkCommUartSend

| **Prototype** | | **LIB\_EXPORT s32 MfSdkCommUartSend(MfSdkCommUart\_E nCom, u8\* pBuffer, u32 nLength)** |
| --- | --- | --- |
| **Function** | | send uart port data |
| **Params** | **in** | **nCom** For details, see MfSdkCommUart\_E. |
| **pBuffer** send buffer |
| **nLength** send buffer length |
| **out** | **None** |
| **return** | | **>=0 send data len** |
| **remark** | |  |
| **demo** | | u8 sendBuf[512] =”12345678”;  if(MfSdkCommUartSend(MFSDK\_COMM\_UART\_COM10,sendBuf,8)==8)  {  //send succ  } |

## MfSdkCommUartSetupComm

| **Prototype** | | **LIB\_EXPORT s32**  **MfSdkCommUartSetupComm(MfSdkCommUart\_E nCom , u8\* buffer , s32 size)** |
| --- | --- | --- |
| **Function** | | set uart comm buffer size.Generally do not set |
| **Params** | **in** | **nCom** For details, see MfSdkCommUart\_E. |
| **buffer** buffer uart comm buffer handler |
| **size** buffer size |
| **out** | **None** |
| **return** | | **MFSDK\_COMM\_RE**  **success**  **T\_OK** |
| **remark** | |  |
| **demo** | | u8 buf[512] ={4096};  if(MfSdkCommUartSetupComm(MFSDK\_COMM\_UART\_COM10,buf,4096)==MF SDK\_COMM\_RET\_OK) |



{

//TODO

}

## MfSdkCommWifiClearListApNum

| **Prototype** | | **LIB\_EXPORT s32 MfSdkCommWifiClearListApNum(void)** |
| --- | --- | --- |
| **Function** | | Clear wifi ap list |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **MFSDK\_COMM\_RE**  **success**  **T\_OK** |
| **remark** | |  |
| **demo** | |  |

## MfSdkCommWifiGetApMac

| **Prototype** | | **LIB\_EXPORT s8 \*MfSdkCommWifiGetApMac(void)** |
| --- | --- | --- |
| **Function** | | Get Wifi AP MAC |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **AP Mac** |
| **remark** | |  |
| **demo** | |  |

## MfSdkCommWifiGetChannel

| **Prototype** | **LIB\_EXPORT s32 MfSdkCommWifiGetChannel(void)** |
| --- | --- |
| **Function** | Get wifi channel |



| **Params** | **in** | **None** |
| --- | --- | --- |
| **out** | **None** |
| **return** | | **wifi channel** |
| **remark** | |  |
| **demo** | |  |

## MfSdkCommWifiGetLinkState

| **Prototype** | | **LIB\_EXPORT MfSdkCommWifiState\_E MfSdkCommWifiGetLinkState(void)** |
| --- | --- | --- |
| **Function** | | Get connection status |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **Ref. MfSdkCommWifiLinkState\_E** |
| **remark** | |  |
| **demo** | |  |

## MfSdkCommWifiGetLocalip

| **Prototype** | | **s8\* MfSdkCommWifiGetLocalip(void)** |
| --- | --- | --- |
| **Function** | | Get Local IP |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **wifi ip** |
| **remark** | |  |
| **demo** | |  |

## MfSdkCommWifiGetLocalMac

**Prototype LIB\_EXPORT s8\* MfSdkCommWifiGetLocalMac(void)**

****

| **Function** | | Get Wifi Mac |
| --- | --- | --- |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **wifi mac** |
| **remark** | |  |
| **demo** | |  |

## MfSdkCommWifiGetRssi

| **Prototype** | | **LIB\_EXPORT s32 MfSdkCommWifiGetRssi(void)** |
| --- | --- | --- |
| **Function** | | Get Wifi Rssi |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **wifi rssi** |
| **remark** | |  |
| **demo** | |  |

## MfSdkCommWifiGetSignal

| **Prototype** | | **LIB\_EXPORT s32 MfSdkCommWifiGetSignal(void)** |
| --- | --- | --- |
| **Function** | | Get Wifi Signal |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **wifi signal** |
| **remark** | |  |
| **demo** | |  |



## MfSdkCommWifiGetSsid

| **Prototype** | | **LIB\_EXPORT s8\* MfSdkCommWifiGetSsid(void)** |
| --- | --- | --- |
| **Function** | | Get Wifi Ssid |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **wifi ssid** |
| **remark** | |  |
| **demo** | |  |

## MfSdkCommWifiLinkAp

| **Prototype** | | **LIB\_EXPORT s32**  **MfSdkCommWifiLinkAp(MfSdkCommWifiApList\_T \*apList, s8**  **\*pwd)** |
| --- | --- | --- |
| **Function** | | Connect to WiFi |
| **Params** | **in** | **apList** MfSdkCommWifiApList\_T item |
| **pwd** wifi password |
| **out** | **None** |
| **return** | | **MFSDK\_COMM\_RE**  **success**  **T\_OK** |
| **other failed** |
| **remark** | |  |
| **demo** | |  |

## MfSdkCommWifiListAp

| **Prototype** | | **LIB\_EXPORT s32**  **MfSdkCommWifiListAp(MfSdkCommWifiApList\_T \*apList)** |
| --- | --- | --- |
| **Function** | | Get the router list |
| **Params** | **in** | **None** |



|  | **out** | Router list data, The ap\_list space is allocated by  **apList** the caller with an array size of 10/30(Soundbox 10,traditional POS 30) |
| --- | --- | --- |
| **return** | | **Number of routers** |
| **remark** | |  |
| **demo** | |  |

## MfSdkCommWifiListApQuit

| **Prototype** | | **LIB\_EXPORT s32 MfSdkCommWifiListApQuit(void)** |
| --- | --- | --- |
| **Function** | | quit the router list |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **MFSDK\_COMM\_RE**  **success**  **T\_OK** |
| **remark** | |  |
| **demo** | |  |

## MfSDkCommWifiListNetWork

| **Prototype** | | **LIB\_EXPORT s32**  **MfSDkCommWifiListNetWork(MfSdkCommWifiApList\_T \*apList)** |
| --- | --- | --- |
| **Function** | | Get the router save ap list |
| **Params** | **in** | **None** |
| **out** | Router list data, The ap\_list space is allocated by  **apList** the caller with an array size of 10/30(Soundbox 10,traditional POS 30) |
| **return** | |  |
| **remark** | |  |
| **demo** | |  |



## MfSdkCommWifiRemoveNetWorkAp

| **Prototype** | | **LIB\_EXPORT s32 MfSdkCommWifiRemoveNetWorkAp(s8\* ssid)** |
| --- | --- | --- |
| **Function** | | Remove Wifi Ap |
| **Params** | **in** | **ssid** Wifi Ap |
| **out** | **None** |
| **return** | | **MFSDK\_COMM\_RE**  **success**  **T\_OK** |
| **remark** | |  |
| **demo** | |  |

## MfSdkCommWifiSetPower

| **Prototype** | | **LIB\_EXPORT s32 MfSdkCommWifiSetPower(s32 flag)** | |
| --- | --- | --- | --- |
| **Function** | | Set the WiFi power | |
| **Params** | **in** | **flag** 0 - Turn off wifi power; 1 - Turn on wifi power | |
| **out** | **None** | |
| **return** | | **MFSDK\_COMM\_RE**  **Success**  **T\_OK** | |
| **other** | **failed** |
| **remark** | |  | |

## MfSdkCommWifiSetScan

| **Prototype** | | **LIB\_EXPORT s32 MfSdkCommWifiSetScan(s32 flag)** |
| --- | --- | --- |
| **Function** | | Start scan wifi |
| **Params** | **in** | 0:stop scan 1:auto scan one time 2:scan  **flag**  immediately one time |
| **out** | **None** |



| **return** | **MFSDK\_COMM\_RE**  **success**  **T\_OK** | |
| --- | --- | --- |
| **other** | **failed** |
| **remark** |  | |

## MfSdkCommWifiUnlinkAp

| **Prototype** | | **LIB\_EXPORT s32 MfSdkCommWifiUnlinkAp(s8\* ssid)** | |
| --- | --- | --- | --- |
| **Function** | | Unlink wifi ap | |
| **Params** | **in** | **ssid** wifi ap | |
| **out** | **None** | |
| **return** | | **MFSDK\_COMM\_RE**  **success**  **T\_OK** | |
| **other** | **failed** |
| **remark** | |  | |

## MfSdkCommSet4gMode

| **Prototype** | | **LIB\_EXPORT s32 MfSdkCommSet4gMode(void)** |
| --- | --- | --- |
| **Function** | | set net to 4G mode |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **MFSDK\_COMM\_RE**  **Success**  **T\_OK** |
| **remark** | |  |
| **demo** | |  |

## MfSdkCommWifiCheckState

**Prototype LIB\_EXPORT s32 MfSdkCommWifiCheckState(void)**

****

| **Function** | | check wifi state |
| --- | --- | --- |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **1 Success** |
| **Other Fail** |
| **remark** | |  |
| **demo** | |  |

## MfSdkCommWifiStartConfig

| **Prototype** | | **LIB\_EXPORT s32 MfSdkCommWifiStartConfig(void)** |
| --- | --- | --- |
| **Function** | | start config wifi |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **MFSDK\_COMM\_RE**  **Success**  **T\_OK** |
| **remark** | |  |
| **demo** | |  |

## MfSdkCommUartInit

| **Prototype** | | **LIB\_EXPORT void MfSdkCommUartInit(void)** |
| --- | --- | --- |
| **Function** | | init device uart |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | |  |
| **remark** | |  |
| **demo** | |  |



## MfSdkCommLanCableCheck

| **Prototype** | | **LIB\_EXPORT s32 MfSdkCommLanCableCheck(void)** |
| --- | --- | --- |
| **Function** | | Whether the device is connected to a network cable |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **0 unconnected** |
| **1 connected** |
| **remark** | |  |
| **demo** | |  |

## MfSdkCommLanSetDHCP

| **Prototype** | | **LIB\_EXPORT s32 MfSdkCommLanSetDHCP(int mod)** |
| --- | --- | --- |
| **Function** | | Set whether to enable DHCP |
| **Params** | **in** | 1 enable,  **mod**  0 disenable |
| **out** | **None** |
| **return** | | **MFSDK\_COMM\_RE**  **Success**  **T\_OK** |
| **Other Ref. MfSdkCommRet\_E** |
| **remark** | |  |
| **demo** | |  |

## MfSdkCommLanIsDHCP

| **Prototype** | **LIB\_EXPORT s32 MfSdkCommLanIsDHCP()** |
| --- | --- |
| **Function** | Whether to enable DHCP |



| **Params** | **in** | **None** |
| --- | --- | --- |
| **out** | **None** |
| **return** | | **0 disenable** |
| **1 enable** |
| **Other Ref. MfSdkCommRet\_E** |
| **remark** | |  |
| **demo** | |  |

## MfSdkCommLanCfg

| **Prototype** | | **LIB\_EXPORT void MfSdkCommLanCfg(const MfSdkCommLanCfg\_T \*lanCfg)** |
| --- | --- | --- |
| **Function** | | Configuring Ethernet |
| **Params** | **in** | **lanCfg** Configuring Ethernet |
| **out** | **None** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | |  |

## MfSdkCommGetSwitchSimProcStatus

| **Prototype** | | **LIB\_EXPORT MfSdkCommSimProcStatus\_E MfSdkCommGetSwitchSimProcStatus()** |
| --- | --- | --- |
| **Function** | | get the status of the sim card switchover |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **For details, see MfSdkCommSimProcStatus\_E** |
| **remark** | |  |
| **demo** | |  |



## MfSdkCommSetOperateIdTask

| **Prototype** | | **LIB\_EXPORT s32 MfSdkCommSetOperateIdTask(s32 val)** |
| --- | --- | --- |
| **Function** | | set the id of the working sim card(nonblocking)(Only devices with two sim cards are supported) |
| **Params** | **in** | 0: sim 1,  **value**  1: sim 2 |
| **out** | **None** |
| **return** | | **>=0 Success** |
| **Other Fail** |
| **remark** | |  |
| **demo** | |  |

## MfSdkCommWifiPageInitMode

| **Prototype** | | **LIB\_EXPORT void MfSdkCommWifiPageInitMode(void\* state,int mode)** |
| --- | --- | --- |
| **Function** | | init wifi logo |
| **Params** | **in** | **state** lv\_obj\_t\* |
| **mode** logo color 0-black ,1-white |
| **out** | **None** |
| **return** | | **None** |
| **remark** | |  |
| **demo** | |  |

## MfSdkCommWifiPageInit

| **Prototype** | | **LIB\_EXPORT void MfSdkCommWifiPageInit(void\* obj, int align, int x\_ofs, int y\_ofs)** |
| --- | --- | --- |
| **Function** | | init wifi page. |
| **Params** | **in** | **obj** lv\_obj\_t\* |



|  |  | **align** ref. lv\_align\_t |
| --- | --- | --- |
| **x\_ofs** x offset |
| **y\_ofs** y offset |
| **out** | **None** |
| **return** | | **None** |
| **remark** | |  |
| **demo** | |  |

## MfSdkCommAtcPageInit

| **Prototype** | | **LIB\_EXPORT void MfSdkCommAtcPageInit(void\* obj, int align, int x\_ofs, int y\_ofs)** |
| --- | --- | --- |
| **Function** | | init gprs page |
| **Params** | **in** | **obj** lv\_obj\_t\* |
| **align** ref. lv\_align\_t |
| **x\_ofs** x offset |
| **y\_ofs** y offset |
| **out** | **None** |
| **return** | | **None** |
| **remark** | |  |
| **demo** | |  |

## MfSdkCommAtcPageInitMode

| **Prototype** | | **LIB\_EXPORT void MfSdkCommAtcPageInitMode(void\* state,int mode)** |
| --- | --- | --- |
| **Function** | | init GSM logo |
| **Params** | **in** | **state** lv\_obj\_t\* |
| **mode** logo color 0-black ,1-white |
| **out** | **None** |
| **return** | | **None** |
| **remark** | |  |
| **demo** | |  |



## MfSdkCommLanEnable

| **Prototype** | | **LIB\_EXPORT s32 MfSdkCommLanEnable();** |
| --- | --- | --- |
| **Function** | | enable Ethernet |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **For details, see MfSdkCommRet\_E** |
| **remark** | |  |
| **demo** | |  |

## MfSdkCommLanDisenable

| **Prototype** | | **LIB\_EXPORT s32 MfSdkCommLanEnable();** |
| --- | --- | --- |
| **Function** | | disenable Ethernet |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **For details, see MfSdkCommRet\_E** |
| **remark** | |  |
| **demo** | |  |

## MfSdkCommAtcGetIp

| **Prototype** | | **LIB\_EXPORT s32 MfSdkCommAtcGetIp(MfSdkCommIpType\_E ipType, s8\* ip ,s32 length)** |
| --- | --- | --- |
| **Function** | | Get local IP, default IPv4 |
| **Params** | **in** | **ipType** IPv4/IPv6 |
| **length** ip buffer size |
| **out** | **ip** ip buffer |



| **return** | **> 0 ip length** |
| --- | --- |
| **other failed** |
| **remark** |  |
| **demo** |  |

## MfSdkCommLanGetEthernetPower

| **Prototype** | | **LIB\_EXPORT s32 MfSdkCommLanGetEthernetPower(void);** |
| --- | --- | --- |
| **Function** | | Get Ethernet Power |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **1 Turn on** |
| **0 Turn off** |
| **remark** | |  |
| **demo** | |  |

## MfSdkCommLanSetEthernetPower

| **Prototype** | | **LIB\_EXPORT s32 MfSdkCommLanSetEthernetPower(s32 flag);** |
| --- | --- | --- |
| **Function** | | Set Ethernet Power |
| **Params** | **in** | **flag** 0 - Turn off , 1 - Turn on |
| **out** | **None** |
| **return** | | **For details, see MfSdkCommRet\_E** |
| **remark** | |  |
| **demo** | |  |

## MfSdkCommLanChipExist

**Prototype LIB\_EXPORT s32 MfSdkCommLanChipExist()**

****

| **Function** | | Check if the device has a WiFi chip |
| --- | --- | --- |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **1 Have WiFi chip** |
| **0 No WiFi chip** |
| **remark** | |  |
| **demo** | |  |

## MfSdkCommWifiChipExist

| **Prototype** | | **LIB\_EXPORT s32 MfSdkCommWifiChipExist()** |
| --- | --- | --- |
| **Function** | | Check if the device has a Ethernet chip |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **1 Have Ethernet chip** |
| **0 No Ethernet chip** |
| **remark** | |  |
| **demo** | |  |

## MfSdkCommAtcSetNetMode

| **Prototype** | | **LIB\_EXPORT void MfSdkCommAtcSetNetMode(s32 mode)** |
| --- | --- | --- |
| **Function** | | Set network mode |
| **Params** | **in** | 0 = "AUTO";1 = "only 2G";2 = "only 3G"; 3 =  **mode**  "only 4G";4 = "2G\_3G";5 = "2G\_3G\_4G"; |
| **out** | **None** |
| **return** | | **None** |
| **remark** | |  |
| **demo** | |  |



## MfSdkCommAtcSendAtCmd

| **Prototype** | | **LIB\_EXPORT s32 MfSdkCommAtcSendAtCmd(s8\* cmd, s32 timeover, s8\* retstr, s32 len, s32 count)** |
| --- | --- | --- |
| **Function** | | Send AT command |
| **Params** | **in** | **cmd** at cmd |
| **timeover** timeover(ms) |
| **len** AT cmd len |
| **count** retry count |
| **out** | **retstr** Returned data |
| **return** | | **0 Success** |
| **Other Failure** |
| **remark** | |  |
| **demo** | |  |

## MfSdkCommLanGetIpAndMask

| **Prototype** | | **LIB\_EXPORT s32 MfSdkCommLanGetIpAndMask(s8\* ip, s8\* Mask);** |
| --- | --- | --- |
| **Function** | | Get Ethernet IP and mask |
| **Params** | **out** | **ip** ip info |
| **Mask** mask info |
| **in** | **None** |
| **return** | | **0 Success** |
| **Other Failure** |
| **remark** | |  |
| **demo** | |  |

## MfSdkCommLanGetGateway

**Prototype LIB\_EXPORT s32 MfSdkCommLanGetGateway(s8\* gateway)**

****

| **Function** | | Gateway Gateway info |
| --- | --- | --- |
| **Params** | **in** | **None** |
| **out** | **Gateway** Gateway info |
| **return** | | **0 Success** |
| **Other Failure** |
| **remark** | |  |
| **demo** | |  |

## MfSdkCommLanGetDns

| **Prototype** | | **LIB\_EXPORT s32 MfSdkCommLanGetDns(s8 dnsArr[][64], s32 dnsArrCount);** |
| --- | --- | --- |
| **Function** | | Get Ethernet DNS |
| **Params** | **in** | **dnsArrCount** dns array count |
| **out** | **dnsArr** dns info |
| **return** | | **>0 Success,count of DNS** |
| **Other Failure** |
| **remark** | |  |

## MfSdkCommLanGetMac

| **Prototype** | | **LIB\_EXPORT s32 MfSdkCommLanGetMac(s8\* mac);** |
| --- | --- | --- |
| **Function** | | Get Ethernet mac |
| **Params** | **in** | **None** |
| **out** | **mac** mac info |
| **return** | | **0 Success** |
| **Other Failure** |
| **remark** | |  |
| **demo** | |  |



## MfSdkCommConfigReset

| **Prototype** | | **LIB\_EXPORT void MfSdkCommConfigReset(void);** |
| --- | --- | --- |
| **Function** | | Reply to network connection status |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **None** |
| **remark** | |  |
| **demo** | |  |

## MfSdkCommWifiStopConfig

| **Prototype** | | **LIB\_EXPORT void MfSdkCommWifiStopConfig(void);** |
| --- | --- | --- |
| **Function** | | WIFI stop config network |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **None** |
| **remark** | |  |
| **demo** | |  |

## MfSdkCommWifiRestart

| **Prototype** | | **LIB\_EXPORT void MfSdkCommWifiRestart(void)** |
| --- | --- | --- |
| **Function** | | Restart the WiFi module only for ET389 |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **None** |
| **remark** | |  |



**demo**

## MfSdkCommSetPingAddr

| **Prototype** | | **LIB\_EXPORT s32 MfSdkCommSetPingAddr(u8\* addr);** |
| --- | --- | --- |
| **Function** | | Set ping address |
| **Params** | **in** | **addr** address.example:"[www.baidu.com](http://www.baidu.com/)" |
| **out** | **None** |
| **return** | | **0 Success** |
| **Other Failure** |
| **remark** | |  |
| **demo** | |  |

## MFSdkCommSocketGetModelType

| **Prototype** | | **LIB\_EXPORT int MFSdkCommSocketGetModelType(int socket\_index);** |
| --- | --- | --- |
| **Function** | | Get current model type |
| **Params** | **in** | **socket\_index** socket index |
| **out** | **None** |
| **return** | | **0 WIRELESS** |
| **1 WIFI** |
| **remark** | |  |
| **demo** | |  |

# EMV module

## Module description

This module mainly includes EMV setting.



## Module structure declaration

typedef struct

{

u8 cTranType;// 00 sale 09 cashback

u8 szAmt [12+1];

u8 szOtherAmt[12+1];

u8 cReadCardMode;// refer to MfSdkCardMode . Can be combined , 0x07 is all

u8 cPinInput; // u8 cPinMinLen;

u8 cPinMaxLen; u8 cKeyPid;//

SEC\_DUKPT\_FIELD,SEC\_MKSK\_FIELD,SEC\_FIXED\_FIELD

u8 cDukptPinIndex; u8 cDukptDataIndex;

u8 cDesMode; //DES\_MODE\_ECB,DES\_MODE\_CBC u8 cMKSKPinIndex;

u8 cPinFormat; //refer to SEC\_PIN\_FORMAX in libapi\_security.h u8 cBypassPin;

}sdk\_read\_card\_in;

typedef struct

{

u8 cCardType; u8 szPan[22];

u8 szPinBlock[16+1]; u8 szPinKSN[10+1];

u8 szDataKSN[10+1];

u8 szTrack2[SDK\_TRACK\_MAX\_LENTH]; u8 cTrack2Len;

u8 szTrack3[SDK\_TRACK\_MAX\_LENTH]; u8 cTrack3Len;

u8 cEmvClsMode;

u8 szServiceCode[3+1]; u8 cClsCvmMethod; u16 nErrCode;

}sdk\_read\_card\_out;

ypedef struct {

u8 szRid[5]; //tag 9f06

u8 cCapkIndex; //tag 9f22



u8 szCapkExpire[4]; //tag df05

u8 cCapkHashFlag; //tag df06

u8 cCapkFlag; //tag df07

u8 szCapkMod[248]; //tag df02

u8 cCapkModLen; //szCAPKMod len u8 szCapkExponent[3]; //tag df04

u8 cCapkExpLen; //szCAPKExponent len u8 szCapkCheckSum3[20]; //tag df03

u8 cCheckSumLen; //szCAPKCheckSum3 len

}MfSdkEmvCapkInfo\_T;

typedef st\_read\_card\_in MfSdkEmvReadCardIn\_T; typedef st\_read\_card\_out MfSdkEmvReadCardOut\_T; typedef card\_magtek\_track\_info MfSdkEmvTrackInfo\_T; typedef AID\_STRU MfSdkEmvAidSt\_T;

## Constant declarations



| #define MODE\_API\_PBOC | 0x01 | | ///< pboc |
| --- | --- | --- | --- |
| #define MODE\_API\_VCPS | 0x02 | | ///< vcps |
| #define MODE\_API\_MSD | 0x04 | | ///< msd |
| #define MODE\_API\_AMEX | 0x40 | | ///< AMEX |
| #define MODE\_API\_M\_CHIP | 0x21 | | ///< MasterCard |
| CHIP mode  #define MODE\_API\_M\_STRIPE 0x22 ///< MasterCard MagStripe mode  #define MODE\_API\_R\_LEGACY 0x23 ///< RUPAY LEGACY mode  #define MODE\_API\_R\_NON\_LEGACY 0x24 ///< RUPAY NON LEGACY mode  #define MODE\_API\_EMV\_MODE 0x25 ///< EMV\_MODE  (for AMEX) | | | |
| #define MODE\_API\_MS\_MODE (for AMEX)  #define MODE\_API\_MOBILE\_MODE mobile\_MODE (for amex) | | 0x26  0x27 | ///< MS\_MODE  ///< |
| // application TLV TAG  #define APP\_TAG\_DF8A01\_AIDOTHERTLV  "\xDF\x8A\x01" // aid other tlv | | | |



#define APP\_TAG\_DF8A02\_FORCEONLINE

"\xDF\x81\x02" //0:no forced online; 1:forced online(TVR) 2:foced online (request ARQC in case of TC)

#define DEF\_TAG\_AID\_MCHIP\_OTHER\_PARAM "\xDF\x84\x06" //AID other Contact

#define DEF\_TAG\_AID\_RF\_OTHER\_PARAM "\xDF\x84\x07" //AID other contactless

#define DEF\_TAG\_TAP\_MODE "\xDF\x84\x08"

//tap mode (1-long tap 2-second tap)

#define DEF\_TAG\_CONTACTLESS\_REFUN\_CONFIG

"\xDF\x84\x0A" // Contactless Refund Config #define APP\_TAG\_DF810C\_KERNELID "\xDF\x81\x0C" // Kernel ID

#define SDK\_TRACK\_MAX\_LENTH 144

enum

{

SDK\_QPBOC\_VER=0, SDK\_VISA\_VER, SDK\_MASTER\_VER, SDK\_AMEX\_VER, SDK\_DPAS\_VER, SDK\_RUPAY\_VER,

}SDK\_KERNEL\_VER;

enum

{

SDK\_KERNELID\_MASTER = 0x02, SDK\_KERNEID\_VISA, SDK\_KERNELID\_AMEX, SDK\_KERNELID\_JCB, SDK\_KERNELID\_DISCOVER, SDK\_KERNELID\_UNIONPAY, SDK\_KERNELID\_SIBS, SDK\_KERNELID\_PURE,

}SDK\_KERNEL\_ID;

enum

{

MSG\_SELECT\_APP, MSG\_ONLINE\_PIN, MSG\_OFFLINE\_PIN, MSG\_OFFLINE\_PIN\_RETRY\_COUNT,



MFSDK\_EMV\_MSG\_DISPLAY\_CARD\_NO, // display card no. MFSDK\_EMV\_MSG\_SET\_PUREAMT\_BEFOREGPO,

}CALLBACK\_MSG;

enum

{

MFSDK\_EMV\_CARD\_RET\_QUIT, ///< quit MFSDK\_EMV\_CARD\_RET\_INPUT,///< input MFSDK\_EMV\_CARD\_MAGTEK, ///< magstripe card MFSDK\_EMV\_CARD\_ICC, ///< ICC MFSDK\_EMV\_CARD\_RFID, ///< RF

}MfSdkEmvReadCardMode\_E;

enum

{

MFSDK\_EMV\_CARD\_PROC\_RET\_TC = 0, ///< TC (offine approve)

MFSDK\_EMV\_CARD\_PROC\_RET\_ARQC = 1, ///< ARQC(online

require)

MFSDK\_EMV\_CARD\_PROC\_RET\_AAC = -11, ///< AAC(offline

declined)

MFSDK\_EMV\_CARD\_PROC\_RET\_TERM = - 2, ///< Terminate MFSDK\_EMV\_CARD\_PROC\_RET\_CANCEL = -3, // User cancel MFSDK\_EMV\_CARD\_PROC\_RET\_TIMEOUT = -4, //Timeout

MFSDK\_EMV\_CARD\_PROC\_RET\_FORCEIC = -5, //Transaction

force IC

MFSDK\_EMV\_CARD\_PROC\_RET\_OTHER = -6, //Contactless

turn to other interface

MFSDK\_EMV\_CARD\_PROC\_RET\_FALLBACK = -7, // Fallback MFSDK\_EMV\_CARD\_PROC\_RET\_SEEPHONE = -8, // See

Phone and Retry

MFSDK\_EMV\_CARD\_PROC\_RET\_APPBLOCK = -9, //app block MFSDK\_EMV\_CARD\_PROC\_RET\_CARDBLOCK = -10, //Card block MFSDK\_EMV\_CARD\_PROC\_RET\_NOAPP = -12, //card

without any application available MFSDK\_EMV\_CARD\_PROC\_RET\_ERR\_POWERUP = -13, //card not

responding or chip not present MFSDK\_EMV\_CARD\_PROC\_RET\_INVDATA = -14, //card

behaves correctly but has invalid or inconsistent data.

MFSDK\_EMV\_CARD\_PROC\_RET\_APPNAUT = -15, //card

returned error 6985

MFSDK\_EMV\_CARD\_PROC\_RET\_CAPK\_NOFOUND = -16, //card

CAPK not found



MFSDK\_EMV\_CARD\_PROC\_RET\_APDU\_NORETURN = -17,

} MfSdkEmvCardProcRet\_E; enum {

MFSDK\_READ\_CARD\_MODE\_MAG = 0x01, ///< magstripe MFSDK\_READ\_CARD\_MODE\_IC = 0x02, ///< ic MFSDK\_READ\_CARD\_MODE\_RF = 0x04 ///< rf

}MfSdkCardMode;

typedef int(\*MfSdkEmvSelecAppDisplay)(MfSdkEmvAidSt\_T \*stICCAID,int nAidNum,int \*select\_choic);

typedef int (\*MfSdkEmvInputOffPin)(char \*psCardNo,char \*psAmt,char cMsgType,char \*psPin);

typedef int (\*MfSdkEmvInputOnlinePin)(char \*psCardNo,char \*psAmt,char

\*psPin);

typedef void (\*MfSdkReadPage)(void);

typedef int (\*MfSdkRuPay2ndPage)(void);

typedef int (\*MfSdkDpas2ndPage)(void);

typedef int (\*MfSdkPreprocessSetOtherDataCb)(u8\*pOutData ,s32 iOutData);

## MfSdkEmvKernelInit

| **Prototype** | | **LIB\_EXPORT void MfSdkEmvKernelInit(void)** |
| --- | --- | --- |
| **Function** | | **Emv kernel init** |
| **Param s** | **in** | **Nothing** |
| **out** | **Nothing** |
| **return** | | **< 0 Failed** |
| **0 Success** |
| **remark** | |  |
| **demo** | | **MfSdkEmvKernelInit()** |



## MfSdkEmvTerminalConfigInit

| **Prototype** | | **LIB\_EXPORT s32 MfSdkEmvTerminalConfigInit(u8\***  **ConfigBuff,s32 BuffLen)** |
| --- | --- | --- |
| **Function** | | **Set emv terminal configuration** |
| **Param s** | **in** | **ConfigBuff:** Terminal parameters(all in tlv format)  **BuffLen** Length of configBuff |
| **out** | **Nothing** |
| **return** | | **< 0 Failed** |
| **0 Success** |
| **remark** | |  |
| **demo** | |  |

## MfSdkEmvCardLoop

| **Prototype** | | **LIB\_EXPORT s32 MfSdkEmvCardLoop(s32 cardMode,s32**  **timeout)** | |
| --- | --- | --- | --- |
| **Function** | | **Loop detection card** | |
| **Param s** | **in** | **CardMode**  **timeout** | 0x01:mag, 0x02:ICC,0x04:RFIC,0x07:All  Unit is seconds |
| **out** | **Nothing** | |
| **return** | | **> 0** | **Ref.** MfSdkCardMode |
| **0** | No card |
| **-3** | It meams timeout. |
| **-4** | Multiple cards |
| **remark** | |  | |
| **demo** | | s32 ret = MfSdkEmvCardLoop(0x01 | 0x02 | 0x04,60); | |



if(MFSDK\_READ\_CARD\_MODE\_MAG == ret){

//TODO magstripe

}

else if(MFSDK\_READ\_CARD\_MODE\_IC == ret){

//TODO contact

}

else if(MFSDK\_READ\_CARD\_MODE\_RF == ret){

//TODO contactless

}

## MfSdkEmvCardProc

| **Prototype** | | **LIB\_EXPORT s32 MfSdkEmvCardProc(s32**  **ret,sdk\_read\_card\_in st\_in,u8\* inBuff,s32 inLen,sdk\_read\_card\_out \* st\_out)** |
| --- | --- | --- |
| **Function** | | **Card reading process** |
| **Param s** | **in** | **ret** cardType: return value of "MfSdkEmvCardLoop"  **st\_in inbuff**  **inLen** |
| **out** | **st\_out** |
| **return** | |  |
| **remark** | |  |
| **demo** | |  |

## MfSdkEmvGetCard

| **Prototype** | **LIB\_EXPORT s32 MfSdkEmvGetCard(s32 ret,u8\* inBuff,u8\*outBuff);** |
| --- | --- |



| **Function** | | **Only for BC** |
| --- | --- | --- |
| **Params** | **in** | **ret** |
| **inBuff** |
| **out** | **outBuff** |
| **return** | | **None** |
| **remark** | |  |
| **demo** | |  |

## MfSdkEmvGoOnChip

| **Prototype** | | **LIB\_EXPORT s32 MfSdkEmvGoOnChip(s32 ret,u8\* inBuff,u8\*outBuff);** |
| --- | --- | --- |
| **Function** | | **Only for BC** |
| **Params** | **in** | **ret** |
| **inBuff** |
| **out** | **outBuff** |
| **return** | | **None** |
| **remark** | |  |
| **demo** | |  |

## MfSdkEmvSetAid

| **Prototype** | | **LIB\_EXPORT s32 MfSdkEmvSetAid(u8\* AidBuff,s32 BuffLen)** |
| --- | --- | --- |
| **Function** | | Add one aid |
| **Param s** | **in** | **AidBuff** TLV hex format  **BuffLen** |
| **out** | **st\_out** |
| **return** | | **< 0 Failed** |
| **0 Success** |
| **remark** | |  |
| **demo** | |  |



## MfSdkEmvDeleteOneAid

| **Prototype** | | **LIB\_EXPORT s32 MfSdkEmvDeleteOneAid(u8\* Aid,u8**  **aidLength)** |
| --- | --- | --- |
| **Function** | | delete one aid |
| **Param s** | **in** | **Aid** Aid buff  **aidLength** |
| **out** | **None** |
| **return** | | **< 0 Failed** |
| **0 Success** |
| **remark** | |  |
| **demo** | |  |

## MfSdkEmvClearAllAid

| **Prototype** | | **LIB\_EXPORT void MfSdkEmvClearAllAid(void);** |
| --- | --- | --- |
| **Function** | | delete all aid |
| **Param s** | **in** | **None** |
| **out** | **None** |
| **return** | | **< 0 Failed** |
| **0 Success** |
| **remark** | |  |
| **demo** | |  |



## MfSdkEmvGetAidNum

| **Prototype** | | **LIB\_EXPORT s32 MfSdkEmvGetAidNum(void);** |
| --- | --- | --- |
| **Function** | | The total number of aids |
| **Param s** | **in** | **None** |
| **out** | **None** |
| **return** | | **>0 aid total** |
| **Other Fail** |
| **remark** | |  |
| **demo** | |  |

## MfSdkEmvSetCapk

| **Prototype** | | **LIB\_EXPORT s32 MfSdkEmvSetCapk(u8 \* CapkBuff,s32**  **BuffLen);** |
| --- | --- | --- |
| **Function** | | add one capk |
| **Param s** | **in** | **CapkBuff** CapkBuff TLV format |
| **out** | **None** |
| **return** | | **0 success** |
| **Other Fail** |
| **remark** | |  |
| **demo** | |  |

## MfSdkEmvDeleteAllCapk

| **Prototype** | | **LIB\_EXPORT s32 MfSdkEmvDeleteAllCapk(void);** |
| --- | --- | --- |
| **Function** | | delete all capk |
| **Param s** | **in** | **None** |
| **out** | **None** |



| **return** | **0 success** |
| --- | --- |
| **Other Fail** |
| **remark** |  |
| **demo** |  |

## MfSdkEmvGetCapkNum

| **Prototype** | | **LIB\_EXPORT s32 MfSdkEmvGetCapkNum(void);** |
| --- | --- | --- |
| **Function** | | The total number of capks |
| **Param s** | **in** | **None** |
| **out** | **None** |
| **return** | | **>=0 capk total** |
| **Other Fail** |
| **remark** | |  |
| **demo** | |  |

## MfSdkEmvGetCapkByIndex

| **Prototype** | | **LIB\_EXPORT s32**  **MfSdkEmvGetCapkByIndex(MfSdkEmvCapkInfo\_T\* stCapk, s32 nRecNum);** |
| --- | --- | --- |
| **Function** | | get capk info by index |
| **Param s** | **in** | **nRecNum** index |
| **out** | **stCapk** capk info |
| **return** | | **=0 success** |
| **Other Fail** |
| **remark** | |  |
| **demo** | |  |



## MfSdkEmvGetDataByTag

| **Prototype** | | **LIB\_EXPORT s32 MfSdkEmvGetDataByTag(u8\* Tag,u8\***  **Src,s32 ScrLen,u8\* OutBuff,s32\* OutLen);** |
| --- | --- | --- |
| **Function** | | Get tag data from TLV string(hex format) |
| **Param s** | **in** | **Tag** Tag |
| **Src** Src |
| **ScrLen** Scr Len |
| **out** | **OutBuff** Out Buff |
| **OutLen** Out Buff Len |
| **return** | | **=0 success** |
| **Other Fail** |
| **remark** | |  |
| **demo** | |  |

## MfSdkEmvGetKernelData

| **Prototype** | | **LIB\_EXPORT s32 MfSdkEmvGetKernelData(u8\* Tag,s32\***  **OutValueLen,u8\* OutValue)** |
| --- | --- | --- |
| **Function** | | Get emv tag from kernel |
| **Param s** | **in** | **Tag** |
| **out** | **OutValueLen**  **OutValue** |
| **return** | | **< 0 Failed** |
| **0 Success** |
| **remark** | |  |
| **demo** | |  |



## MfSdkEmvGetDataByTag

| **Prototype** | | **LIB\_EXPORT s32 MfSdkEmvGetDataByTag(u8\* Tag,u8\***  **Src,s32 ScrLen,u8\* OutBuff,s32\* OutLen)** |
| --- | --- | --- |
| **Function** | | Get tag data from TLV string(hex format) |
| **Param s** | **in** | **Tag**  **Src ScrLen** |
| **out** | **OutBuff**  **OutLen** |
| **return** | | **< 0 Failed** |
| **0 Success** |
| **remark** | |  |
| **demo** | |  |

## MfSdkEmvPackTLVData

| **Prototype** | | **LIB\_EXPORT s32 MfSdkEmvPackTLVData(u8\* Tag,u8\***  **TagValue,s32 ValueLen,u8\* OutBuff,s32\* OutLen)** |
| --- | --- | --- |
| **Function** | | Packed TLV string to an array |
| **Param s** | **in** | **Tag**  **TagValue ValueLen** |
| **out** | **OutBuff**  **OutLen** |
| **return** | | **< 0 Failed** |
| **0 Success** |
| **remark** | |  |
| **demo** | |  |



## MfSdkEmvSetKernelData

| **Prototype** | | **LIB\_EXPORT s32 MfSdkEmvSetKernelData(u8\* Tag,u8\***  **SetData,s32 DataLen,s32 bOverRide)** |
| --- | --- | --- |
| **Function** | | Get tag data from TLV string(hex format) |
| **Param s** | **in** | **Tag SetData DataLen**  **bOverRide** YES or No |
| **out** |  |
| **return** | | **< 0 Failed** |
| **0 Success** |
| **remark** | |  |
| **demo** | |  |

## MfSdkEmvSetDRL

| **Prototype** | | **LIB\_EXPORT s32 MfSdkEmvSetDRL(TERMDRL stDRL)** |
| --- | --- | --- |
| **Function** | | Set visa and amex DRL config |
| **Param s** | **in** | **stDRL** |
| **out** | **None** |
| **return** | | **< 0 Failed** |
| **0 Success** |
| **remark** | |  |
| **demo** | |  |



## MfSdkEmvClearDRLFile

| **Prototype** | | **LIB\_EXPORT s32 MfSdkEmvClearDRLFile()** |
| --- | --- | --- |
| **Function** | | clear all DRL |
| **Param s** | **in** | **None** |
| **out** | **None** |
| **return** | | **< 0 Failed** |
| **0 Success** |
| **remark** | |  |
| **demo** | |  |

## MfSdkEmvMatchErrCode

| **Prototype** | | **LIB\_EXPORT s32 MfSdkEmvMatchErrCode(u32 iErrcode)** |
| --- | --- | --- |
| **Function** | | Match error code |
| **Param s** | **in** | **iErrcode** |
| **out** | **None** |
| **return** | | **< 0 Failed** |
| **0 Success** |
| **remark** | |  |
| **demo** | |  |



## MfSdkEmvOnlineRespPack

| **Prototype** | | **LIB\_EXPORT s32 MfSdkEmvOnlineRespPack(s32**  **nOnlineRes,s8\* sField39,u8\* sField55,u8\* EmvTag,u8\* PackValue,s32\* packLen)** |
| --- | --- | --- |
| **Function** | | Process of emv online resp proc and pack tlv data |
| **Param s** | **in** | **nOnlineRes** 0--online success(Any return in the server is success) -1--online fail  **sField39**  **sField55 EmvTag** |
| **out** | **PackValue**  **packLen** |
| **return** | | **< 0 Failed** |
| **0 Success** |
| **remark** | |  |
| **demo** | |  |

## MfSdkEmvCardFree

| **Prototype** | | **LIB\_EXPORT s32 MfSdkEmvCardFree(s32 Type)** |
| --- | --- | --- |
| **Function** | | Process of emv online resp proc and pack tlv data |
| **Param s** | **in** | **Type** 2:free icc card data; 3:free rfic card data |
| **out** | **None** |
| **return** | | **< 0 Failed** |
| **0 Success** |
| **remark** | |  |
| **demo** | |  |



## MfSdkEmvAddCardBlackList

| **Prototype** | | **LIB\_EXPORT s32 MfSdkEmvAddCardBlackList(BlackCard**  **stBlackCard,s32 cCoverFlag)** |
| --- | --- | --- |
| **Function** | | Add one card black list |
| **Param s** | **in** | **stBlackCard**  **cCoverFlag** |
| **out** | **None** |
| **return** | | **< 0 Failed** |
| **0 Success** |
| **remark** | |  |
| **demo** | |  |

## MfSdkEmvDelCardBlackList

| **Prototype** | | **LIB\_EXPORT s32 MfSdkEmvDelCardBlackList(BlackCard**  **stBlackCard)** |
| --- | --- | --- |
| **Function** | | detlete one card black list |
| **Param s** | **in** | **stBlackCard** |
| **out** | **None** |
| **return** | | **< 0 Failed** |
| **0 Success** |
| **remark** | |  |
| **demo** | |  |

## MfSdkEmvGetVersion

**Prototype LIB\_EXPORT s32 MfSdkEmvGetVersion(s8 \*Version)**

****

| **Function** | | Get emv version |
| --- | --- | --- |
| **Param s** | **in** | **Version** Version string |
| **out** | **None** |
| **return** | | **< 0 Failed** |
| **0 Success** |
| **remark** | |  |
| **demo** | |  |

## MfSdkEmvGetEntryVersion

| **Prototype** | | **LIB\_EXPORT s32 MfSdkEmvGetEntryVersion(s8\* Version)** |
| --- | --- | --- |
| **Function** | | Get emv entry version |
| **Param s** | **in** | **Version** Version string |
| **out** | **None** |
| **return** | | **< 0 Failed** |
| **0 Success** |
| **remark** | |  |
| **demo** | |  |

## MfSdkEmvGetContactlessVersion

| **Prototype** | | **LIB\_EXPORT s32 MfSdkEmvGetContactlessVersion(s32**  **KernelType,s8\* Version)** |
| --- | --- | --- |
| **Function** | | Get emv contactless version |
| **Param s** | **in** | **Version** Version string  **KernelType:** refer to SDK\_KERNEL\_VER |
| **out** | **None** |
| **return** | | **< 0 Failed** |
| **0 Success** |



| **remark** |  |
| --- | --- |
| **demo** |  |

## MfSdkEmvSetCallBackFunction

| **Prototype** | | **LIB\_EXPORT void MfSdkEmvSetCallBackFunction(s32 (\*callback)(s32 MsgType,u8\* Indata,s32 InLen,u8\***  **OutData,s32\* Outlen))** |
| --- | --- | --- |
| **Function** | | Set emv callback function |
| **Param s** | **in** | **callback** callback func  **MsgType** refer to CALLBACK\_MSG  **Indata: InLen** |
| **out** | **OutData**  **Outlen** |
| **return** | | **< 0 Failed** |
| **0 Success** |
| **remark** | |  |
| **demo** | |  |

## MfSdkEmvReadCardPage

| **Prototype** | | **lv\_obj\_t\*MfSdkEmvReadCardPage(lv\_obj\_t**  **\*parent,void\*pfunc,MfSdkEmvReadCardIn\_T \*pCardIn, MfSdkEmvReadCardOut\_T \*pCardOut);** | |
| --- | --- | --- | --- |
| **Function** | | create read card | page |
| **Params** | **in** | **parent** | lvgl parent obj |
| **pCardIn** | Card In parameter |
| **out** | **pfunc** | callback |



|  |  | **pCardOut** Card out parameter |
| --- | --- | --- |
| **return** | | **lv\_obj\_t\* lvgl page obj** |
| **remark** | |  |
| **demo** | |  |

## MfSdkEmvSetSelectAppCallback

| **Prototype** | | **void**  **MfSdkEmvSetSelectAppCallback(MfSdkEmvSelecAppDisplay pFunc);** |
| --- | --- | --- |
| **Function** | | Mutil App select |
| **Params** | **in** | **pFunc** callback |
| **out** | **None** |
| **return** | | **None** |
| **remark** | |  |
| **demo** | |  |

## MfSdkEmvOfflinePinCallback

| **Prototype** | | **void MfSdkEmvOfflinePinCallback(MfSdkEmvInputOffPin pFunc);** |
| --- | --- | --- |
| **Function** | | offline pin callback |
| **Params** | **in** | **pFunc** callback |
| **out** | **None** |
| **return** | | **None** |
| **remark** | |  |
| **demo** | |  |

## MfSdkEmvOnlinePinCallback

**Prototype void MfSdkEmvOnlinePinCallback(MfSdkEmvInputOnlinePin**

****

| **Function** | | online pin callback |
| --- | --- | --- |
| **Params** | **in** | **pFunc** callback |
| **out** | **None** |
| **return** | | **None** |
| **remark** | |  |
| **demo** | |  |

## MfSdkEmvSetReadPageCallback

| **Prototype** | | **void MfSdkEmvSetReadPageCallback(MfSdkReadPage pFunc);** |
| --- | --- | --- |
| **Function** | | display read card processing |
| **Params** | **in** | **pFunc** callback |
| **out** | **None** |
| **return** | | **None** |
| **remark** | |  |
| **demo** | |  |

## MfSdkEmvSetRuPay2ndTapCallback

| **Prototype** | | **void MfSdkEmvSetRuPay2ndTapCallback(MfSdkRuPay2ndPage pFunc);** |
| --- | --- | --- |
| **Function** | | second tap UI |
| **Params** | **in** | **pFunc** callback |
| **out** | **None** |
| **return** | | **None** |
| **remark** | |  |
| **demo** | |  |



## MfSdkEmvSetDpas2ndTapCallback

| **Prototype** | | **void MfSdkEmvSetDpas2ndTapCallback(MfSdkDpas2ndPage pFunc);** |
| --- | --- | --- |
| **Function** | | second tap UI |
| **Params** | **in** | **pFunc** callback |
| **out** | **None** |
| **return** | | **None** |
| **remark** | |  |
| **demo** | |  |

## MfSdkEmvGetPageWin

| **Prototype** | | **lv\_obj\_t\* MfSdkEmvGetPageWin(void);** |
| --- | --- | --- |
| **Function** | | Get EMV lvgl page handler |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **lv\_obj\_t\*** EMV lvgl page handler |
| **remark** | |  |
| **demo** | |  |

## MfSdkEmvSetPreprocessOtherDataCallback

| **Prototype** | | **void**  **MfSdkEmvSetPreprocessOtherDataCallback(MfSdkPreprocessSe tOtherDataCb pFuncCallback);** |
| --- | --- | --- |
| **Function** | | set contactless other data callback |
| **Params** | **in** | **pFunc** callback |
| **out** | **None** |
| **return** | | **None** |
| **remark** | |  |



**demo**

## MfSdkEmvGetPageWinTip

| **Prototype** | | **lv\_obj\_t\* MfSdkEmvGetPageWinTip(void);** |
| --- | --- | --- |
| **Function** | | set contactless other data callback |
| **Params** | **in** | **pFunc** callback |
| **out** | **None** |
| **return** | | **None** |
| **remark** | |  |
| **demo** | |  |

## MfSdkEmvGetProcInfo

| **Prototype** | | **s32 MfSdkEmvGetProcInfo(s8\* outBuf, s32 bufLen);** |
| --- | --- | --- |
| **Function** | | Get read card information |
| **Params** | **in** | **bufLen** buff len. |
| **out** | **outBuf** buff(>4096) ,ASCII format |
| **return** | | **None** |
| **remark** | |  |
| **demo** | |  |

## MfSdkEmvCallbackEventSetAmtBeforeGpo

| **Prototype** | | **void MfSdkEmvCallbackEventSetAmtBeforeGpo(void);** |
| --- | --- | --- |
| **Function** | | set amount before gpo |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **None** |



| **remark** |  |
| --- | --- |
| **demo** |  |

## MfSdkEmvCallbackEventGetAmtBeforeGpo

| **Prototype** | | **s32 MfSdkEmvCallbackEventGetAmtBeforeGpo(void);** |
| --- | --- | --- |
| **Function** | | get amount before gpo |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **0x00** |
| **0x01** |
| **remark** | |  |
| **demo** | |  |

## MfSdkEmvCallbackEventClear

| **Prototype** | | **void MfSdkEmvCallbackEventClear(void);** |
| --- | --- | --- |
| **Function** | | clear emv callback event |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **None** |
| **remark** | |  |
| **demo** | |  |

## MfSdkEmvCallbackEventClear

| **Prototype** | **char\* MfSdkEmvGetEmvL2Version(void);** |
| --- | --- |
| **Function** | Get EMV(contact) kernel L2 cert version |



| **Params** | **in** | **None** |
| --- | --- | --- |
| **out** | **None** |
| **return** | | **!=NULL EMV(contact) kernel L2 cert version** |
| **remark** | |  |
| **demo** | |  |

## MfSdkEmvGetEmvL2KernelName

| **Prototype** | | **char\* MfSdkEmvGetEmvL2Version(void);** |
| --- | --- | --- |
| **Function** | | Get EMV(contact) kernel L2 cert kernel name |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **!=NULL EMV(contact) kernel L2 cert version** |
| **remark** | |  |
| **demo** | |  |

## MfSdkEmvGetAidsInit

| **Prototype** | | **void \*MfSdkEmvGetAidsInit(void);** |
| --- | --- | --- |
| **Function** | | aid params init ref. MfSdkEmvGetAidsFree |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **!=NULL aids pointer** |
| **remark** | |  |
| **demo** | |  |

## MfSdkEmvGetAid

**s32 MfSdkEmvGetAid(void \*pTerminalApps, s32 index , u8**

**\*outAidsTlv,s32 outAidsTlvLength);**

****

| **Function** | | Get Aid tlv |
| --- | --- | --- |
| **Params** | **in** | **pTerminalApps aids pointer** |
| **aid index index must less than**  **index**  **MfSdkEmvGetAidNum()** |
| **outAidsTlvLength outAidsTlv size min value 1024** |
| **out** | **None aid tlv length** |
| **return** | | **> =0 aids pointer** |
| **other ref. MfSdkRet\_E** |
| **remark** | |  |
| **demo** | |  |

## MfSdkEmvGetAidsFree

| **Prototype** | | **void MfSdkEmvGetAidsFree(void \*pTerminalApps);** |
| --- | --- | --- |
| **Function** | | free aids param pointer |
| **Params** | **in** | **pTerminalApps aids pointer** |
| **out** | **None** |
| **return** | | **None** |
| **remark** | |  |
| **demo** | |  |

## MfSdkEmvSetTlv2Kernel

| **Prototype** | | **s32 MfSdkEmvSetTlv2Kernel(u8\* pBerTlvData , s32 length);** |
| --- | --- | --- |
| **Function** | | set Kernel emv tlv |
| **Params** | **in** | **pBerTlvData** tlv data |
| **length** tlv data length |
| **out** | **None** |
| **return** | | **None** |
| **remark** | |  |



**demo**

# System module

## Module description

This module mainly includes system APIs.

## Module structure declaration



## Constant declarations

| #define MFSDK\_SYS\_DEVICE\_MF960 | CPU\_EC8521 |  |
| --- | --- | --- |
| ///<mf960 |  |  |
| #define MFSDK\_SYS\_DEVICE\_H9N | CPU\_EC25 | ///<h9n |



#define MFSDK\_SYS\_DEVICE\_H9L CPU\_V133 ///<h9l #define MFSDK\_SYS\_DEVICE\_H9PRO CPU\_X2600\_H9Pro

///<h9pro

#define MFSDK\_SYS\_DEVICE\_960B CPU\_X2600\_MF960

///<960b

typedef enum

{

MFSDK\_SYS\_RET\_FAILED = -3, //failed MFSDK\_SYS\_RET\_BOUNDS = -2, //Array out-of-bounds MFSDK\_SYS\_RET\_PARM\_ERROR = -1, //check param MFSDK\_SYS\_RET\_OK = 0,

}MfSdkSysRet\_E;

typedef enum

{

MFSDK\_SYS\_LCD\_TYPE\_GET\_FILE = -1, ///<get file MFSDK\_SYS\_LCD\_TYPE\_480\_800 = 0, ///<mf960

MFSDK\_SYS\_LCD\_TYPE\_320\_240, ///<h9u/h9l/h9n MFSDK\_SYS\_LCD\_TYPE\_240\_240 = 11, MFSDK\_SYS\_LCD\_TYPE\_320\_480, MFSDK\_SYS\_LCD\_TYPE\_128\_32,

}MfSdkSysLcdType\_E;

typedef enum

{

MFSDK\_SYS\_POWER\_SHUTDOWN, MFSDK\_SYS\_POWER\_LOW, MFSDK\_SYS\_POWER\_CHARGE, MFSDK\_SYS\_POWER\_OUT, MFSDK\_SYS\_NET\_CONFIG\_MODE, MFSDK\_SYS\_NET\_GPRS\_MODE, MFSDK\_SYS\_NET\_WIFI\_MODE, MFSDK\_SYS\_WIFI\_AIRKISS\_CONFIG, MFSDK\_SYS\_WIFI\_AP\_CONFIG, MFSDK\_SYS\_WIFI\_CONFIG\_SUCCESS, MFSDK\_SYS\_WIFI\_CONFIG\_FAIL,

MFSDK\_SYS\_TMS\_UPDATE\_START,//Tms Update Start MFSDK\_SYS\_TMS\_UPDATE\_UNZIP,//Tms Update Unzip MFSDK\_SYS\_TMS\_UPDATE\_SUCC,//Tms Update Succ MFSDK\_SYS\_TMS\_UPDATE\_FAIL,//Tms Update Fail MFSDK\_SYS\_TMS\_SIGN\_SUCC,//Tms Sign Succ

Pause

MFSDK\_SYS\_TMS\_SIGN\_FAIL,//Tms Sign Fail MFSDK\_SYS\_TMS\_DOWN\_FINISH\_SUCC,//Tms Down Finish Succ MFSDK\_SYS\_TMS\_DOWN\_FINISH\_FAIL,//Tms Down Finish Fail MFSDK\_SYS\_TMS\_DOWN\_FINISH\_PAUSE,//Tms Down Finish

MFSDK\_SYS\_TMS\_DOWN\_FINISH\_CONTINUE,//Tms Finish

Continue

MFSDK\_SYS\_TMS\_DOWN\_FINISH\_STOP,//Tms FinishStop MFSDK\_SYS\_LOW\_BATTERY\_SHUTDOWN,//Low Battery Shutdown MFSDK\_SYS\_RTC\_TIME\_SET,//RTC SET TIME

MFSDK\_SYS\_POWER\_FULL,//Battery full

}MfSdkSysState\_E;

typedef enum

{

MFSDK\_SYS\_DEV\_MODEL\_UNKNOWN = -1,

MFSDK\_SYS\_DEV\_MODEL\_M60A = 0, MFSDK\_SYS\_DEV\_MODEL\_M60B, MFSDK\_SYS\_DEV\_MODEL\_M90L, MFSDK\_SYS\_DEV\_MODEL\_M60C, MFSDK\_SYS\_DEV\_MODEL\_MP60, MFSDK\_SYS\_DEV\_MODEL\_70N, MFSDK\_SYS\_DEV\_MODEL\_70N2, MFSDK\_SYS\_DEV\_MODEL\_M90, MFSDK\_SYS\_DEV\_MODEL\_H9, MFSDK\_SYS\_DEV\_MODEL\_H9A, MFSDK\_SYS\_DEV\_MODEL\_H9L, MFSDK\_SYS\_DEV\_MODEL\_H9N, MFSDK\_SYS\_DEV\_MODEL\_H9U, MFSDK\_SYS\_DEV\_MODEL\_H9PRO, MFSDK\_SYS\_DEV\_MODEL\_ET389, MFSDK\_SYS\_DEV\_MODEL\_ET389PRO, MFSDK\_SYS\_DEV\_MODEL\_MF919, MFSDK\_SYS\_DEV\_MODEL\_MF960, MFSDK\_SYS\_DEV\_MODEL\_MF960B, MFSDK\_SYS\_DEV\_MODEL\_70N3, MFSDK\_SYS\_DEV\_MODEL\_SR600MINI, MFSDK\_SYS\_DEV\_MODEL\_MP70A30, MFSDK\_SYS\_DEV\_MODEL\_MP70N5, MFSDK\_SYS\_DEV\_MODEL\_MP70A20, MFSDK\_SYS\_DEV\_MODEL\_MP70MIS, MFSDK\_SYS\_DEV\_MODEL\_MP70A5,



MFSDK\_SYS\_DEV\_MODEL\_H9PROA30, MFSDK\_SYS\_DEV\_MODEL\_MP70A6, MFSDK\_SYS\_DEV\_MODEL\_Z990,

}MfSdkSysDevType\_E;

typedef enum

## MfSdkSysGetDevModel

| **Prototype** | | **LIB\_EXPORT s32 MfSdkSysGetDevModel(void)** |
| --- | --- | --- |
| **Function** | | Obtain the terminal model. |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **Dev Model Ref. MfSdkSysDevType\_E** |
| **remark** | |  |
| **demo** | |  |

## MfSdkSysGetDevModelName

| **Prototype** | | **LIB\_EXPORT char\* MfSdkSysGetDevModelName(void)** |
| --- | --- | --- |
| **Function** | | Obtain the terminal name. |
| **Params** | **in** | **None** |
| **out** | **None** |



| **return** | **device model name** |
| --- | --- |
| **remark** |  |
| **demo** | ET389Pro:  char \*modelName = MfSdkSysGetDevModelName(); APP\_TRACE("modelName:%s\r\n",modelName); Result:  modelName:ET389Pro |

## MfSdkSysDevIs

| **Prototype** | | **LIB\_EXPORT MFSDKBOOL MfSdkSysDevIs(s32 model)** |
| --- | --- | --- |
| **Function** | | Check whether the terminal is a model. |
| **Params** | **in** | **model** Ref. MfSdkSysDevType\_E |
| **out** | **None** |
| **return** | | **MFSDK\_TRUE yes** |
| **MFSDK\_FALSE no** |
| **remark** | |  |
| **demo** | | APP\_TRACE("MFSDK\_SYS\_DEV\_MODEL\_ET389 :%s\r\n",MfSdkSysDevIs(MFSDK\_SYS\_DEV  \_MODEL\_ET389)); |

## MfSdkSysGetHardwareVer

| **Prototype** | | **LIB\_EXPORT s32 MfSdkSysGetHardwareVer(void)** |
| --- | --- | --- |
| **Function** | | Get hardware version. |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **hardware version 300/400/500** |
| **remark** | |  |
| **demo** | |  |



## MfSdkSysGetTime

| **Prototype** | | **LIB\_EXPORT s32 MfSdkSysGetTime(MfSdkSysTime\_T \*pTime)** |
| --- | --- | --- |
| **Function** | | Get terminal date and time. |
| **Params** | **in** | **\*pTime MfSdkSysTime\_T point.** |
| **out** | **None** |
| **return** | | **MFSDK\_SYS\_RE**  **Success**  **T\_OK** |
| **MFSDK\_SYS\_RE**  **fail**  **T\_PARM\_ERROR** |
| **remark** | |  |
| **demo** | | MfSdkSysTime\_T stDateTime; memset(&stDateTime,0,sizeof(MfSdkSysTime\_T)); if(MfSdkSysGetTime(&stDateTime) == MFSDK\_SYS\_RET\_OK)  {  //TODO  } |

## MfSdkSysAppIsLock

| **Prototype** | | **LIB\_EXPORT s32 MfSdkSysAppIsLock()** |
| --- | --- | --- |
| **Function** | | Determine whether the app is locked. For example, when TMS is running, the application is locked. |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **1 lock** |
| **0 unlock** |
| **remark** | |  |
| **demo** | | if(MfSdkSysAppIsLock() == 1)  {  //TODO  } |



## MfSdkSysAuxlcdGetBrightness

| **Prototype** | | **LIB\_EXPORT s32 MfSdkSysAuxlcdGetBrightness()** |
| --- | --- | --- |
| **Function** | | LCD get brightness |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **light The range is between 10 and 255** |
| **remark** | |  |
| **demo** | |  |

## MfSdkSysAuxlcdSetBrightness

| **Prototype** | | **LIB\_EXPORT void MfSdkSysAuxlcdSetBrightness(s32 light)** |
| --- | --- | --- |
| **Function** | | LCD set brightness |
| **Params** | **in** | **light** The range is between 10 and 255 |
| **out** | **None** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | |  |

## MfSdkSysSubAuxlcdGetBrightness

| **Prototype** | | **s32 MfSdkSysSubAuxlcdGetBrightness();** |
| --- | --- | --- |
| **Function** | | Sub LCD get brightness |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **>=10** The range is between 10 and 255 |
| **remark** | |  |



**demo**

## MfSdkSysSubAuxlcdSetBrightness

| **Prototype** | | **void MfSdkSysAuxlcdSetBrightness(s32 light);** |
| --- | --- | --- |
| **Function** | | Sub LCD set brightness |
| **Params** | **in** | **light** light The range is between 10 and 255 |
| **out** | **None** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | |  |

## MfSdkSysBatterCharge

| **Prototype** | | **LIB\_EXPORT s32 MfSdkSysBatterCharge(void)** |
| --- | --- | --- |
| **Function** | | Charging state |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **MFSDK\_SYS\_RET\_**  **charging**  **OK** |
| **Other not charging** |
| **remark** | |  |
| **demo** | |  |

## MfSdkSysGetBatterStatus

| **Prototype** | | **LIB\_EXPORT s32 MfSdkSysGetBatterStatus(MfSdkBatterAttr\_T\* batterystatus)** |
| --- | --- | --- |
| **Function** | | Battery state |
| **Params** | **in** | **None** |



|  | **out** | **batterystatus** battery status |
| --- | --- | --- |
| **return** | | **1 charging** |
| **0 not charging** |
| **-1 fail** |
| **remark** | |  |
| **demo** | |  |

## MfSdkSysBuzzerSound

| **Prototype** | | **LIB\_EXPORT void MfSdkSysBuzzerSound(s32 nMillisecond)** |
| --- | --- | --- |
| **Function** | | Buzzer |
| **Params** | **in** | **nMillisecond** time |
| **out** | **None** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | |  |

## MfSdkSysCheckKey

| **Prototype** | | **LIB\_EXPORT s32 MfSdkSysCheckKey(void)** |
| --- | --- | --- |
| **Function** | | Scan button, non blocking |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **KEY\_VALUE key pressed** |
| **0 No key** |
| **-1 Fail** |
| **remark** | |  |
| **demo** | |  |



## MfSdkSysCheckTick

| **Prototype** | | **LIB\_EXPORT s32 MfSdkSysCheckTick(s32 tick, s32 timeover)** |
| --- | --- | --- |
| **Function** | | Check for timeout |
| **Params** | **in** | **tick** Starting time |
| **timeover** Timeout period |
| **out** | **None** |
| **return** | | **1 timeout** |
| **0 within the effective time** |
| **remark** | |  |
| **demo** | |  |

## MfSdkSysClrKey

| **Prototype** | | **LIB\_EXPORT void MfSdkSysClrKey(void)** |
| --- | --- | --- |
| **Function** | | Not implemented |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | |  |

## MfSdkSysConfig

| **Prototype** | | **LIB\_EXPORT void MfSdkSysConfig(void)** |
| --- | --- | --- |
| **Function** | | The manufacturer sets the individual parameter settings and invokes the individual settings function of each manufacturer. |
| **Params** | **in** | **None** |
| **out** | **None** |



| **return** | **Nothing** |
| --- | --- |
| **remark** |  |
| **demo** |  |

## MfSdkSysCurlInit

| **Prototype** | | **LIB\_EXPORT s32 MfSdkSysCurlInit(s32 Argc, s8\*\* Argv, s8\* AppName)** |
| --- | --- | --- |
| **Function** | | Initialization of system and initialization of application layer |
| **Params** | **in** | **Argc** Reusing main function parameter Argc |
| **Argv** Reusing main function parameter Argv |
| **AppName** Application name |
| **out** | **None** |
| **return** | | **For the first time since the program was**  **1**  **updated** |
| **2 Program is not run for the first time** |
| **Other Fail** |
| **remark** | | **Not to be processed without access. Inside the function calls the private API of the manufacturer.The API returns to the first operation only when the program is newly installed,and the**  **update is not the first operation.** |
| **demo** | |  |

## MfSdkSysDelay

| **Prototype** | | **LIB\_EXPORT void MfSdkSysDelay(u32 uiMs)** |
| --- | --- | --- |
| **Function** | | Delay blocking |
| **Params** | **in** | **uiMs** Delay time unit MS |
| **out** | **None** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | |  |



## MfSdkSysEnergySetTime

| **Prototype** | | **LIB\_EXPORT s32 MfSdkSysEnergySetTime(s32 nTime)** |
| --- | --- | --- |
| **Function** | | Energy-saving time |
| **Params** | **in** | **nTime** unit second |
| **out** | **None** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | |  |

## MfSdkSysEnergyTime

| **Prototype** | | **LIB\_EXPORT s32 MfSdkSysEnergyTime()** |
| --- | --- | --- |
| **Function** | | Get Energy-saving time |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **Energy-saving time unit:second** |
| **remark** | |  |
| **demo** | |  |

## MfSdkSysEraseSecureArea

| **Prototype** | | **LIB\_EXPORT s32 MfSdkSysEraseSecureArea(u32 addr)** |
| --- | --- | --- |
| **Function** | | Erase safe area data, 4K at a time. |
| **Params** | **in** | **addr** 0 / 4\*1024 |
| **out** | **None** |
| **return** | | **MFSDK\_SYS\_RE**  **Success**  **T\_OK** |



|  | **Other Fail** |
| --- | --- |
| **remark** |  |
| **demo** | int iRet = Sys\_EraseSecureArea(0);  iRet = Sys\_EraseSecureArea(4\*1024); |

## MfSdkSysFileSetPath

| **Prototype** | | **LIB\_EXPORT void MfSdkSysFileSetPath(s32 mode)** |
| --- | --- | --- |
| **Function** | | Set the storage mode |
| **Params** | **in** | Storage location,  **mode**  Reference enum FILELOCATION |
| **out** | **None** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | |  |

## MfSdkSysGetBatterLevel

| **Prototype** | | **LIB\_EXPORT s32 MfSdkSysGetBatterLevel(void)** |
| --- | --- | --- |
| **Function** | | Get power supply |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **0-5 Battery level** |
| **-1 Fail** |
| **remark** | |  |
| **demo** | |  |

## MfSdkSysGetDateTime

**Prototype LIB\_EXPORT s32 MfSdkSysGetDateTime(u8\* DateTime)**

****

| **Function** | | Get SysTime |
| --- | --- | --- |
| **Params** | **in** | **None** |
| **out** | **DateTime** "YYYYMMDDHHMMSS" 14 bytes |
| **return** | | **MFSDK\_SYS\_RET\_**  **Success**  **OK** |
| **Other Fail** |
| **remark** | |  |
| **demo** | |  |

## MfSdkSysGetIsLcd

| **Prototype** | | **LIB\_EXPORT s32 MfSdkSysGetIsLcd(void)** |
| --- | --- | --- |
| **Function** | | Check whether 389 has a front display |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **1 have front display** |
| **-1 have no front display** |
| **remark** | |  |
| **demo** | |  |

## MfSdkSysGetLcdType

| **Prototype** | | **LIB\_EXPORT s32 MfSdkSysGetLcdType()** |
| --- | --- | --- |
| **Function** | | Get the screen LCD type |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **LCD type Ref. MfSdkSysLcdType\_E** |
| **remark** | |  |
| **demo** | |  |



## MfSdkSysGetPsn

| **Prototype** | | **LIB\_EXPORT void MfSdkSysGetPsn(s8\* psn)** |
| --- | --- | --- |
| **Function** | |  |
| **Params** | **in** | **None** |
| **out** | **psn** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | |  |

## MfSdkSysGetTerminalInfo

| **Prototype** | | **LIB\_EXPORT s32**  **MfSdkSysGetTerminalInfo(MfSdkSysTermInfo\_T\* terminal)** |
| --- | --- | --- |
| **Function** | | Get terminal information |
| **Params** | **in** | **None** |
| **out** | **terminal** Reference TERMINALINFO |
| **return** | | **MFSDK\_SYS\_RET\_**  **success**  **OK** |
| **Other fail** |
| **remark** | |  |
| **demo** | |  |

## MfSdkSysGetTermSn

| **Prototype** | | **LIB\_EXPORT s32 MfSdkSysGetTermSn(s8\* Sn)** |
| --- | --- | --- |
| **Function** | | Get terminal SN |
| **Params** | **in** | **None** |
| **out** | **sn** Reference TERMINALINFO |



| **return** | **MFSDK\_SYS\_RET\_**  **success**  **OK** |
| --- | --- |
| **Other fail** |
| **remark** | **no safe please use API MfSdkSysGetTerminalSn** |
| **demo** |  |

## MfSdkSysGetTerminalSn

| **Prototype** | | **LIB\_EXPORT const s8\* MfSdkSysGetTerminalSn(s8\*pData,s32 length)** |
| --- | --- | --- |
| **Function** | | Get terminal SN |
| **Params** | **in** | **length** |
| **out** | **pData** Reference TERMINALINFO |
| **return** | | **None** |
| **remark** | |  |
| **demo** | |  |

## MfSdkSysGetTick

| **Prototype** | | **LIB\_EXPORT u32 MfSdkSysGetTick()** |
| --- | --- | --- |
| **Function** | | Get the current time |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **tick Get time tick unit ms** |
| **remark** | |  |
| **demo** | |  |

## MfSdkSysGetTickDiff

**Prototype LIB\_EXPORT s32 MfSdkSysGetTickDiff(s32 tick1)**

****

| **Function** | | Calculate time difference |
| --- | --- | --- |
| **Params** | **in** | **tick1** start tick |
| **out** | **None** |
| **return** | | **tick Time difference** |
| **remark** | |  |
| **demo** | |  |

## MfSdkSysGetTimeStamp

| **Prototype** | | **LIB\_EXPORT time\_t MfSdkSysGetTimeStamp()** |
| --- | --- | --- |
| **Function** | | get time stamp |
| **Params** | **in** | **tick1** start tick |
| **out** | **None** |
| **return** | | **time\_t lt time stamp** |
| **remark** | |  |
| **demo** | |  |

## MfSdkSysGuiGroupGetObj

| **Prototype** | | **LIB\_EXPORT void\* MfSdkSysGuiGroupGetObj()** |
| --- | --- | --- |
| **Function** | |  |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **lvgl object** |
| **remark** | |  |
| **demo** | |  |



## MfSdkSysHttpDownload

| **Prototype** | | **LIB\_EXPORT s32 MfSdkSysHttpDownload(s32 sock\_index, const s8\* url, const s8\* fullpathfilename)** |
| --- | --- | --- |
| **Function** | | http download file |
| **Params** | **in** | socket used by the device Ref.  **sock\_index**  MfSdkCommSocketIndex\_E |
| **url** Download link |
| **fullpathfilename** Save path |
| **out** | **None** |
| **return** | | **MFSDK\_SYS\_RET\_**  **success**  **OK** |
| **other fail** |
| **remark** | |  |
| **demo** | |  |

## MfSdkSysHttpDownloadUseAgent

| **Prototype** | | **LIB\_EXPORT void MfSdkSysHttpDownloadUseAgent(s32 useAgent)** |
| --- | --- | --- |
| **Function** | | Http download message configuration |
| **Params** | **in** | UseAgent Whether to use User-Agent for http  **useAgent**  download(0 - 1) |
| **out** | **None** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | |  |

## MfSdkSysInit

**Prototype LIB\_EXPORT s32 MfSdkSysInit(s32 Argc, s8\*\* Argv, s8\***

****

| **Function** | | Initialization of system and initialization of application layer Nothing. (For Traditional POS) |
| --- | --- | --- |
| **Params** | **in** | **Argc** Reusing main function parameter Argc |
| **Argv** Reusing main function parameter Argv |
| **AppName** Application name |
| **out** | **None** |
| **return** | | **<0 fail** |
| **for the first time since the program was**  **1**  **updated.** |
| **2 Program is not run for the first time.** |
| **remark** | | Not to be processed without access.Inside the function calls the private API of the manufacturer.The API returns to the first operation only when the program is newly installed,and the update is not the first  operation. |
| **demo** | |  |

## MfSdkSysLcdCalibration

| **Prototype** | | **LIB\_EXPORT void MfSdkSysLcdCalibration()** |
| --- | --- | --- |
| **Function** | | Screen Calibration |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | |  |

## MfSdkSysLogoInit

| **Prototype** | | **LIB\_EXPORT s32 MfSdkSysLogoInit(s8\* logoPath)** |
| --- | --- | --- |
| **Function** | | logo init |
| **Params** | **in** | **logoPath** Path of the logo.bin file |
| **out** | **None** |



| **return** | **MFSDK\_SYS\_RET\_**  **success**  **OK** |
| --- | --- |
| **Other fail** |
| **remark** |  |
| **demo** |  |

## MfSdkSysLogoInitA

| **Prototype** | | **s32 MfSdkSysLogoInitA(s8\* logoPath,MFSDKBOOL bDelete);** |
| --- | --- | --- |
| **Function** | | logo init |
| **Params** | **in**  **in** | **logoPath** logo file Path |
| Whether to delete logo file MFSDK\_TRUE-yes,  **bDelete**  MFSDK\_FALSE-no |
| **out** | **None** |
| **return** | | **MFSDK\_SYS\_RET\_**  **success**  **OK** |
| **Other fail** |
| **remark** | |  |
| **demo** | |  |

## MfSdkSysModelType

| **Prototype** | | **LIB\_EXPORT s32 MfSdkSysModelType()** |
| --- | --- | --- |
| **Function** | |  |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **0 wireless** |
| **1 MODEM** |
| **remark** | |  |
| **demo** | |  |



## MfSdkSysNetSetDatacallType

| **Prototype** | | **LIB\_EXPORT void MfSdkSysNetSetDatacallType()** |
| --- | --- | --- |
| **Function** | | Set other network injection processes |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **Nothing** |
| **remark** | | Called before Sys\_Start.Resolved the jiociot sim card network injection  failure |
| **demo** | |  |

## MfSdkSysPrintAdd

| **Prototype** | | **LIB\_EXPORT void MfSdkSysPrintAdd(const s8\* pbuff)** |
| --- | --- | --- |
| **Function** | | print buff add |
| **Params** | **in** | **pbuff** print buff |
| **out** | **None** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | |  |

## MfSdkSysReadFlashData

| **Prototype** | | **LIB\_EXPORT void MfSdkSysReadFlashData(s8\* pdata, s32 size)** |
| --- | --- | --- |
| **Function** | | Read flash data |
| **Params** | **in** | **size** data size |
| **out** | **pdata** read Data content |
| **return** | | **Nothing** |



| **remark** |  |
| --- | --- |
| **demo** |  |

## MfSdkSysReadSecureArea

| **Prototype** | | **LIB\_EXPORT s32 MfSdkSysReadSecureArea(u32 addr, u8\* pData, s32 iDataLength)** |
| --- | --- | --- |
| **Function** | | Read secure area data. |
| **Params** | **in** | **addr** Read data start offset 0-8191 |
| **iDataLength** |
| **out** | **pData** |
| **return** | | **>=0 read data length** |
| **< 0 fail** |
| **remark** | |  |
| **demo** | | s32 iRet = 0;  u8 mBuffer[1024] = {0};  iRet = Sys\_ReadSecureArea(0, mBuffer, sizeof(mBuffer)); |

## MfSdkSysReboot

| **Prototype** | | **LIB\_EXPORT s32 MfSdkSysReboot(void)** |
| --- | --- | --- |
| **Function** | | Terminal reboot |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **MFSDK\_SYS\_RET\_**  **success**  **OK** |
| **-1 fail** |
| **remark** | |  |
| **demo** | |  |



## MfSdkSysRfidEmulateConfig

| **Prototype** | | **LIB\_EXPORT s32 MfSdkSysRfidEmulateConfig(u8\* cardinfo, s32 infolen, u8(\*cardblock)[4], u32 block\_cap)** |
| --- | --- | --- |
| **Function** | | ntag analog card config |
| **Params** | **in** | **cardinfo** Pointer to memory card information |
| **infolen** The length of card information |
| A two-dimensional array pointer for storing card  **cardblock**  block data |
| **block\_cap** The number of blocks that the card can support |
| **out** | **None** |
| **return** | | **MFSDK\_SYS\_RET\_**  **success**  **OK** |
| **other fail** |
| **remark** | |  |
| **demo** | |  |

## MfSdkSysRfidEmulateDeinit

| **Prototype** | | **LIB\_EXPORT s32 MfSdkSysRfidEmulateDeinit(void)** |
| --- | --- | --- |
| **Function** | | ntag deinitialization of analog cards |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **MFSDK\_SYS\_RET\_**  **success**  **OK** |
| **Other fail** |
| **remark** | |  |
| **demo** | |  |



## MfSdkSysRfidEmulateInit

| **Prototype** | | **LIB\_EXPORT s32 MfSdkSysRfidEmulateInit(void)** |
| --- | --- | --- |
| **Function** | | ntag analog card initialization |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **MFSDK\_SYS\_RET\_**  **success**  **OK** |
| **Other fail** |
| **remark** | |  |
| **demo** | |  |

## MfSdkSysRfidEmulateProcess

| **Prototype** | | **LIB\_EXPORT void MfSdkSysRfidEmulateProcess(void)** |
| --- | --- | --- |
| **Function** | | ntag analog card execution processing |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | |  |

## MfSdkSysRun

| **Prototype** | | **LIB\_EXPORT s32 MfSdkSysRun()** |
| --- | --- | --- |
| **Function** | | system running |
| **Params** | **in** | **None** |
| **out** | **None** |



| **return** | **Nothing** |
| --- | --- |
| **remark** |  |
| **demo** |  |

## MfSdkSysSetDateTime

| **Prototype** | | **LIB\_EXPORT s32 MfSdkSysSetDateTime(u8\* DateTime)** |
| --- | --- | --- |
| **Function** | | Set SysTime |
| **Params** | **in** | **DateTime** "YYYYMMDDHHMMSS" 14 bytes |
| **out** | **None** |
| **return** | | **-2 Parameter Error** |
| **-1 Fail** |
| **0 Success** |
| **remark** | | **Internal judgment of time format in API** |
| **demo** | |  |

## MfSdkSysSetLanguage

| **Prototype** | | **LIB\_EXPORT void MfSdkSysSetLanguage(s32 nLanguageType)** |
| --- | --- | --- |
| **Function** | | Set device language, must set the right font priority |
| **Params** | **in** | 0 - Chinese,  **nLanguageType** 1 - English,  2 - Persian |
| **out** | **None** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | |  |



## MfSdkSysSetLogData

| **Prototype** | | **LIB\_EXPORT void MfSdkSysSetLogData(s8\* buff, s32 len)** |
| --- | --- | --- |
| **Function** | | Log set |
| **Params** | **in** | **buff** |
| **len** |
| **out** | **None** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | |  |

## MfSdkSysSetScrBackLight

| **Prototype** | | **LIB\_EXPORT s32 MfSdkSysSetScrBackLight(byte mode)** |
| --- | --- | --- |
| **Function** | | Set screen backlight mode |
| **Params** | **in** | Backlight mode:   1. - turn off backlight,   **mode**   1. - system automatically controls, 2. - backlight is always bright. |
| **out** | **None** |
| **return** | | **MFSDK\_SYS\_RET\_**  **success**  **OK** |
| **Other fail** |
| **remark** | | Mode=1 indicates that the backlight is automatically controlled by the  system, with a default of 1. |
| **demo** | |  |

## MfSdkSysSleep

**Prototype LIB\_EXPORT s32 MfSdkSysSleep(u32 Time)**

****

| **Function** | | This API is a delay feature. |
| --- | --- | --- |
| **Params** | **in** | **Time** Entering dormancy time, unit milliseconds |
| **out** | **None** |
| **return** | | **MFSDK\_SYS\_RET\_**  **success**  **OK** |
| **Other fail** |
| **remark** | |  |
| **demo** | |  |

## MfSdkSysStart

| **Prototype** | | **LIB\_EXPORT void MfSdkSysStart(const s8\* app\_ver, MfSdkSysAppStartCallback apprun)** |
| --- | --- | --- |
| **Function** | | Check for timeout |
| **Params** | **in** | **app\_ver** Application version |
| **apprun** Application task |
| **out** | **None** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | |  |

## MfSdkSysTaskAppSet

| **Prototype** | | **LIB\_EXPORT void MfSdkSysTaskAppSet(void\* pfunc)** |
| --- | --- | --- |
| **Function** | | Open Task |
| **Params** | **in** | **pfunc** Processing function of task |
| **out** | **None** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | |  |



## MfSdkSysTaskCreate

| **Prototype** | | **LIB\_EXPORT s32 MfSdkSysTaskCreate(void (\*pfun)(void\* param), s32 prio, void\* param, s32 task\_size)** |
| --- | --- | --- |
| **Function** | | Create system task |
| **Params** | **in** | **pfun** Processing function of task  **prio** Task priority  **param** Task param  **task\_size** Stack size |
| **out** | **None** |
| **return** | | **MFSDK\_SYS\_RET\_**  **success**  **OK** |
| **other fail** |
| **remark** | |  |
| **demo** | |  |

## MfSdkSysTimerCheck

| **Prototype** | | **LIB\_EXPORT s32 MfSdkSysTimerCheck(s32 iHandle)** |
| --- | --- | --- |
| **Function** | | Check whether the timing time is up to |
| **Params** | **in** | **iHandle** Timer |
| **out** | **None** |
| **return** | | **success,Remaining time, unit milliseconds (0**  **>=0**  **indicates timed time)** |
| **Other fail** |
| **remark** | |  |
| **demo** | |  |

## MfSdkSysTimerClose

**Prototype LIB\_EXPORT s32 MfSdkSysTimerClose(s32 iHandle)**

****

| **Function** | | Off Timer |
| --- | --- | --- |
| **Params** | **in** | **iHandle** Timer |
| **out** | **None** |
| **return** | | **MFSDK\_SYS\_RET\_**  **success**  **OK** |
| **Other fail** |
| **remark** | |  |
| **demo** | |  |

## MfSdkSysTimerCreate

| **Prototype** | | **LIB\_EXPORT u32 MfSdkSysTimerCreate(void\* pfunc, void\* pParam, u32 nPeriod, u32 nMode, u32\* pnErrorCode)** |
| --- | --- | --- |
| **Function** | | Create a timer |
| **Params** | **in** | **pfunc** callback func |
| **pParam** params |
| **nPeriod** period |
| **nMode** mode |
| **pnErrorCode** error code |
| **out** | **None** |
| **return** | | **>0 success, timer handle** |
| **0 fail** |
| **remark** | |  |
| **demo** | |  |

## MfSdkSysTimerEnable

| **Prototype** | | **LIB\_EXPORT u32 MfSdkSysTimerEnable(u32 nTimerNo)** |
| --- | --- | --- |
| **Function** | | Enable timer |
| **Params** | **in** | **nTimerNo** timer handle |
| **out** | **None** |
| **return** | | **MFSDK\_SYS\_RET\_ success** |



|  | **OK** |
| --- | --- |
| **Other fail** |
| **remark** |  |
| **demo** |  |

## MfSdkSysTimerDelete

| **Prototype** | | **u32 MfSdkSysTimerDelete(u32 nTimerNo)** |
| --- | --- | --- |
| **Function** | | Delete timer |
| **Params** | **in** | **nTimerNo** timer handle |
| **out** | **None** |
| **return** | | **MFSDK\_SYS\_RET\_**  **success**  **OK** |
| **Other fail** |
| **remark** | |  |
| **demo** | |  |

## MfSdkSysTimerOpen

| **Prototype** | | **LIB\_EXPORT u32 MfSdkSysTimerOpen(u32 TimerMs)** |
| --- | --- | --- |
| **Function** | | Open timer and set timer time |
| **Params** | **in** | **TimerMs** Timer time |
| **out** | **None** |
| **return** | | **>=0 success,return timer handle** |
| **Other fail** |
| **remark** | |  |
| **demo** | |  |



## MfSdkSysTtsSystemSetFunc

| **Prototype** | | **LIB\_EXPORT void MfSdkSysTtsSystemSetFunc(MfSdkSysPlayCallback playproc)** |
| --- | --- | --- |
| **Function** | | system tts callback eg. batter low, wifi mode etc. |
| **Params** | **in** | **playproc** |
| **out** | **None** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | |  |

## MfSdkSysUnzipFileFunc

| **Prototype** | | **LIB\_EXPORT s32 MfSdkSysUnzipFileFunc(s8\* szFileName, s8\* folder, MFSDKBOOL del)** |
| --- | --- | --- |
| **Function** | | Unzip packet |
| **Params** | **in** | **szFileName** Path of the compressed package |
| **folder** Unzip to the specified folder |
| MFSDK\_TRUE:Delete the zip package after the  decompression is complete;  **del**  MFSDK\_FALSE:The zip package is not deleted  after the decompression is successful |
| **out** | **None** |
| **return** | | **MFSDK\_SYS\_RET\_**  **success**  **OK** |
| **other fail** |
| **remark** | |  |
| **demo** | | MfSdkSysUnzipFileFunc("exdata\\images.zip","exdata",MFSDK\_TRUE); |

## MfSdkSysVersion

**Prototype LIB\_EXPORT s32 MfSdkSysVersion()**

****

| **Function** | | get system version and trace out to log tool |
| --- | --- | --- |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **system version** |
| **remark** | |  |
| **demo** | |  |

## MfSdkSysWriteFlashData

| **Prototype** | | **LIB\_EXPORT void MfSdkSysWriteFlashData(s8\* pdata, s32 size)** |
| --- | --- | --- |
| **Function** | |  |
| **Params** | **in** | **pdata** write data stream |
| **size** write data stream length |
| **out** | **None** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | |  |

## MfSdkSysWriteSecureArea

| **Prototype** | | **LIB\_EXPORT s32 MfSdkSysWriteSecureArea(u32 addr, u8\* pData, s32 iDataLength)** |
| --- | --- | --- |
| **Function** | |  |
| **Params** | **in** | **addr** 0--8191 |
| **pData** write data stream |
| **iDataLength** write data stream length |
| **out** | **None** |
| **return** | | **MFSDK\_SYS\_RET\_**  **success**  **OK** |
| **Other fail** |
| **remark** | |  |



**demo** iRet = Sys\_WriteSecureArea(0, (u8\*)"12345678", 8);

## MfSdkSysZipUpdate

| **Prototype** | | **LIB\_EXPORT void MfSdkSysZipUpdate()** |
| --- | --- | --- |
| **Function** | | TMS update(Unpack zip file and upgrade, then reboot) |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | |  |

## MfSdkSysDriverLibInit

| **Prototype** | | **LIB\_EXPORT void MfSdkSysDriverLibInit()** |
| --- | --- | --- |
| **Function** | | Initialization driver |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | |  |

## MfSdkSysTaskAppInit

| **Prototype** | | **LIB\_EXPORT s32 MfSdkSysTaskAppInit(void)** |
| --- | --- | --- |
| **Function** | | if SoundBox support NFC should init |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **MFSDK\_SYS\_RET\_ success** |



|  | **OK** |
| --- | --- |
| **Other fail** |
| **remark** |  |
| **demo** |  |

## MfSdkSysPubDriverInit

| **Prototype** | | **LIB\_EXPORT s32 MfSdkSysPubDriverInit(void)** |
| --- | --- | --- |
| **Function** | | Init Driver |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **MFSDK\_SYS\_RET\_**  **success**  **OK** |
| **Other fail** |
| **remark** | |  |
| **demo** | |  |

## MfSdkSysFontInit

| **Prototype** | | **LIB\_EXPORT s32 MfSdkSysFontInit(void)** |
| --- | --- | --- |
| **Function** | | Init font library |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **MFSDK\_SYS\_RET\_**  **success**  **OK** |
| **Other fail** |
| **remark** | |  |
| **demo** | |  |



## MfSdkSysConsoleSwitch

| **Prototype** | | **void MfSdkSysConsoleSwitch(MfSdkSysConsole\_E switchE);** |
| --- | --- | --- |
| **Function** | | set device port |
| **Params** | **in** | **switchE** refer to MfSdkSysConsole\_E |
| **out** | **None** |
| **return** | | **None** |
| **remark** | |  |
| **demo** | |  |

## MfSdkSysThreadMutexInit

| **Prototype** | | **s32 MfSdkSysThreadMutexInit(MfSdkSysThreadMutex \*mutex);** |
| --- | --- | --- |
| **Function** | | init mutex |
| **Params** | **in** | **None** |
| **out** | **mutex** MfSdkSysThreadMutex handle |
| **return** | | **ref. MfSdkSysRet\_E** |
| **remark** | |  |
| **demo** | |  |

## MfSdkSysThreadMutexLock

| **Prototype** | | **s32 MfSdkSysThreadMutexLock(MfSdkSysThreadMutex**  **\*mutex ,void\* param);** |
| --- | --- | --- |
| **Function** | | mutex lock |
| **Params** | **in** | **mutex** MfSdkSysThreadMutex handle |
| **param** NULL |
| **out** | **None** |
| **return** | | **ref. MfSdkSysRet\_E** |



| **remark** |  |
| --- | --- |
| **demo** |  |

## MfSdkSysThreadMutexUnlock

| **Prototype** | | **s32 MfSdkSysThreadMutexUnlock(MfSdkSysThreadMutex**  **\*mutex);** |
| --- | --- | --- |
| **Function** | | mutex unlock |
| **Params** | **in** | **mutex** MfSdkSysThreadMutex handle |
| **out** | **None** |
| **return** | | **ref. MfSdkSysRet\_E** |
| **remark** | |  |
| **demo** | |  |

## MfSdkSysThreadMutexDestroy

| **Prototype** | | **s32 MfSdkSysThreadMutexDestroy(MfSdkSysThreadMutex**  **\*mutex);** |
| --- | --- | --- |
| **Function** | | Destroy mutex |
| **Params** | **in** | **mutex** MfSdkSysThreadMutex handle |
| **out** | **None** |
| **return** | | **ref. MfSdkSysRet\_E** |
| **remark** | |  |
| **demo** | |  |

## MfSdkSysGetSegmentLcdDisplayMaxLength

| **Prototype** | | **s32 MfSdkSysGetSegmentLcdDisplayMaxLength(void);** | | |
| --- | --- | --- | --- | --- |
| **Function** | | Get segment lcd | max | length of display 6/7/9 |
| **Params** | **in** | **mutex** | MfSdkSysThreadMutex handle | |



|  | **out** | **None** |
| --- | --- | --- |
| **return** | | **>0** Get segment lcd max length of display 6/7/9 |
|  | | **other** failed |
| **remark** | |  |
| **demo** | |  |

## MfSdkSysIsExternalFlash

| **Prototype** | | **s32 MfSdkSysGetSegmentLcdDisplayMaxLength(void);** |
| --- | --- | --- |
| **Function** | | Check whether there is an external flash. |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **MFSDK\_FALSE** not external Flash |
| **MFSDK\_TRUE** external flash |
| **remark** | |  |
| **demo** | |  |

## MfSdkSysSetForceSleepTime

| **Prototype** | | **s32 MfSdkSysSetForceSleepTime(s32 timeS);** |
| --- | --- | --- |
| **Function** | | Set the time for the device to enter the forced sleep mode after the screen is broken |
| **Params** | **in** | **timeS** time. unit seconds |
| **out** | **None** |
| **return** | | ref. MfSdkSysRet\_E |
| **remark** | |  |
| **demo** | |  |

## MfSdkSysSetSleepToPoweroff

**Prototype s32 MfSdkSysSetSleepToPoweroff(s32 timeS);**

****

| **Function** | | Set the shutdown time after the device enters sleep mode |
| --- | --- | --- |
| **Params** | **in** | **timeS** time. unit seconds |
| **out** | **None** |
| **return** | | ref. MfSdkSysRet\_E |
| **remark** | |  |
| **demo** | |  |

## MfSdkSysGetHeapInformation

| **Prototype** | | **void MfSdkSysGetHeapInformation(MfSdkSysHeap\_T**  **\*pHeapInfor);** |
| --- | --- | --- |
| **Function** | | get heap info |
| **Params** | **in** | **None** |
| **out** | **pHeapInfor** heap info |
| **return** | | **None** |
| **remark** | |  |
| **demo** | |  |

## MfSdkSysSetRestartEnabledOnce

| **Prototype** | | **s32 MfSdkSysSetRestartEnabledOnce();** |
| --- | --- | --- |
| **Function** | | not restart the app after it crashes, use crash=reenter |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **ref. MfSdkSysRet\_E** |
| **remark** | |  |
| **demo** | |  |

# File System module

## Module description

This module mainly includes File system APIs.

## Module structure declaration

typedef struct

{

s32 d\_ino; ///< inode number, file system implementation can use it for any purpose

u8 d\_type; ///< 8-file,other- dir type of file char d\_name[256]; ///< file name

}MfSdkDirent\_T;

## Constant declarations





#define MFSDK\_FS\_WRONLY (FILE\_WRONLY) ///< Write only, there is no return error

#define MFSDK\_FS\_RDWR (FILE\_RDWR) ///< Read and write, there is no return error

#define MFSDK\_FS\_APPEND (FILE\_APPEND) ///< Append, create if it does not exist, pointer to the end of the file when it exists

#define MFSDK\_FS\_CREAT (FILE\_CREAT) ///< Create, delete rebuild when it already exists

typedef enum

{

MFSDK\_FS\_FLAG\_WRITE, // read and write MFSDK\_FS\_FLAG\_CREAT, // if file does not exist create

it ,if file exist will be truncated to length 0.

MFSDK\_FS\_FLAG\_READ,// only read

}MfSdkFsFlag\_E; typedef enum

{

MFSDK\_FS\_MODE\_READ, // only read MFSDK\_FS\_MODE\_WRITE, // read and write

}MfSdkFsMode\_E;

## MfSdkFsSetPath

| **Prototype** | | **LIB\_EXPORT s32 MfSdkFsSetPath(const s8 \*pPath)** |
| --- | --- | --- |
| **Function** | | set file path |
| **Params** | **in** | **pPath** file path |
| **out** | **None** |
| **return** | | **MFSDK\_FS\_RET\_O**  **success**  **K** |
| **remark** | |  |
| **demo** | |  |
| **demo** | |  |



## MfSdkFsCheckPath

| **Prototype** | | **LIB\_EXPORT s32 MfSdkFsCheckPath(const s8 \*FileName)** |
| --- | --- | --- |
| **Function** | | check if file exist |
| **Params** | **in** | **FileName** file name |
| **out** | **None** |
| **return** | | **MFSDK\_FS\_RET\_OK The file exist** |
| **MFSDK\_FS\_RET\_NO**  **The file does not exist**  **EXIST** |
| **remark** | |  |
| **demo** | |  |

## MfSdkFsClean

| **Prototype** | | **LIB\_EXPORT s32 MfSdkFsClean()** |
| --- | --- | --- |
| **Function** | | delete all files |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **MFSDK\_FS\_RET\_O**  **success**  **K** |
| **remark** | |  |
| **demo** | |  |

## MfSdkFsClose

| **Prototype** | | **LIB\_EXPORT s32 MfSdkFsClose(FILE\_HANDLE handle)** |
| --- | --- | --- |
| **Function** | | check if file exist |
| **Params** | **in** | **handle** file handler |



|  | **out** | **None** |
| --- | --- | --- |
| **return** | | **MFSDK\_FS\_RET\_O**  **success**  **K** |
| **Other fail,Ref.MfSdkFsRet\_E** |
| **remark** | |  |
| **demo** | |  |

## MfSdkFsDelete

| **Prototype** | | **LIB\_EXPORT s32 MfSdkFsDelete(FILE\_HANDLE handle, u32 size)** |
| --- | --- | --- |
| **Function** | | File truncation |
| **Params** | **in** | **handle** file handler |
| **size** truncate size |
| **out** | **None** |
| **return** | | **MFSDK\_FS\_RET\_OK Success** |
| **Other Fail** |
| **remark** | |  |
| **demo** | |  |

## MfSdkFsGetFreeSpace

| **Prototype** | | **LIB\_EXPORT long MfSdkFsGetFreeSpace(const s8 \*drive)** |
| --- | --- | --- |
| **Function** | | File system remaining space |
| **Params** | **in** | **drive** default "" |
| **out** | **None** |
| **return** | | **>0 success, the remaining space unit:KB** |
| **-1 fail** |
| **remark** | |  |
| **demo** | | MfSdkFsGetFreeSpace(""); |



## MfSdkFsGetTotalSpace

| **Prototype** | | **LIB\_EXPORT long MfSdkFsGetTotalSpace(const s8 \*drive)** |
| --- | --- | --- |
| **Function** | | File system total space |
| **Params** | **in** | **drive** default "" |
| **out** | **None** |
| **return** | | **>0 success, the remaining space unit:KB** |
| **-1 fail** |
| **remark** | |  |
| **demo** | | MfSdkFsGetTotalSpace(""); |

## MfSdkFsGetModuleVer

| **Prototype** | | **LIB\_EXPORT s32 MfSdkFsGetModuleVer(char \*pszVer)** |
| --- | --- | --- |
| **Function** | | Get File module version |
| **Params** | **in** | **None** |
| **out** | **pszVer** |
| **return** | | **MFSDK\_FS\_RET\_O**  **success**  **K** |
| **Other fail** |
| **remark** | |  |
| **demo** | |  |

## MfSdkFsLseek

| **Prototype** | | **LIB\_EXPORT long MfSdkFsLseek(FILE\_HANDLE handle, long offset, s32 origin)** |
| --- | --- | --- |
| **Function** | | Locating file pointer |
| **Params** | **in** | **handle** file handler |



|  |  | **offset** |
| --- | --- | --- |
| **origin** start position |
| **out** | **None** |
| **return** | | **>=0 success, offset position** |
| **Other fail** |
| **remark** | |  |
| **demo** | |  |

## MfSdkFsMkdir

| **Prototype** | | **LIB\_EXPORT s32 MfSdkFsMkdir(s8\* dirName)** |
| --- | --- | --- |
| **Function** | | Creating directories does not support recursive creation. |
| **Params** | **in** | **dirName** |
| **out** | **None** |
| **return** | | **MFSDK\_FS\_RET\_O**  **success**  **K** |
| **Other fail** |
| **remark** | |  |
| **demo** | | if(MfSdkFsMkdir(“edxata\\testdir”) == MFSDK\_FS\_RET\_OK)  {  //create a folder of testdir  } |

## MfSdkFsOpen

| **Prototype** | | **LIB\_EXPORT s32 MfSdkFsOpen(const s8\* name, s32 flag, s32 mode)** |
| --- | --- | --- |
| **Function** | | file open |
| **Params** | **in** | **name** File Name End with NULL |
| **flag** Open the file flag Ref.MfSdkFsFlag\_E |
| Open the file mode and refer to the  **mode**  MfSdkFsMode\_E definition. |
| **out** | **None** |
| **return** | | **>=0 success, file handler** |



|  | **Other fail** |
| --- | --- |
| **remark** |  |
| **demo** | #include <stdio.h> #include <string.h>  #include "libapi\_xpos/inc/mfsdk\_log.h" #include "libapi\_xpos/inc/mfsdk\_fs.h"  #define FS\_TRACE(...) MfSdkLogLevel("app", MFSDK\_LOG\_LEVEL\_TRACE, VA\_ARGS );  #define FS\_TRACE\_BUFF(a, b) MfSdkLogHexBuff("app", MFSDK\_LOG\_LEVEL\_TRACE, a, b);  void TestFsCreateWriteRead(void)  {  s32 ret = MFSDK\_RET\_FAILED;  ret = MfSdkFsCheckPath((const s8 \*)TEST\_FS\_FILE); FS\_TRACE("MfSdkFsCheckPath:%d\r\n", ret);  if(ret == MFSDK\_FS\_RET\_NOEXIST)  { // create new file  ret = MfSdkFsOpen((const s8 \*)TEST\_FS\_FILE, MFSDK\_FS\_FLAG\_CREAT, MFSDK\_FS\_MODE\_WRITE);  FS\_TRACE("MfSdkFsOpen:%d\r\n", ret); if(ret >= 0)  {  MfSdkFsWrite(ret, "Hello World!", strlen("Hello World!")); MfSdkFsClose(ret);  }  }  else  {  ret = MfSdkFsOpen((const s8 \*)TEST\_FS\_FILE, MFSDK\_FS\_FLAG\_WRITE, MFSDK\_FS\_MODE\_WRITE); |
|  | if(ret >= 0)  {  char buffer[1024] = {0}; s32 length = 0; |
|  | length = MfSdkFsLseek(ret, 0, MFSDK\_FS\_SEEK\_SET); FS\_TRACE("MfSdkFsLseek:%d\r\n", length);  length = MfSdkFsRead(ret, buffer, sizeof(buffer)); FS\_TRACE("MfSdkFsRead:%d\r\n", length); |
|  | if(length) { FS\_TRACE("MfSdkFsRead buffer:%s\r\n", buffer); } length = MfSdkFsLseek(ret, 5, MFSDK\_FS\_SEEK\_SET); |



FS\_TRACE("MfSdkFsLseek:%d\r\n", length); memset(buffer, 0, sizeof(buffer));

length = MfSdkFsRead(ret, buffer, sizeof(buffer)); FS\_TRACE("MfSdkFsRead:%d\r\n", length);

if(length) { FS\_TRACE("MfSdkFsRead buffer:%s\r\n", buffer); } length = MfSdkFsLseek(ret, 0, FILE\_SEEK\_CUR); FS\_TRACE("FILE\_SEEK\_CUR MfSdkFsLseek:%d\r\n", length); memset(buffer, 0, sizeof(buffer));

length = MfSdkFsRead(ret, buffer, sizeof(buffer)); FS\_TRACE("MfSdkFsRead:%d\r\n", length);

if(length) { FS\_TRACE("MfSdkFsRead buffer:%s\r\n", buffer); } length = MfSdkFsLseek(ret, 0, FILE\_SEEK\_END); FS\_TRACE("FILE\_SEEK\_END MfSdkFsLseek:%d\r\n", length); memset(buffer, 0, sizeof(buffer));

length = MfSdkFsRead(ret, buffer, sizeof(buffer)); FS\_TRACE("MfSdkFsRead:%d\r\n", length);

if(length) { FS\_TRACE("MfSdkFsRead FILE\_SEEK\_CUR buffer:%s\r\n", buffer); }

MfSdkFsClose(ret);

}

}

}

## MfSdkFsPathClean

| **Prototype** | | **LIB\_EXPORT s32 MfSdkFsPathClean(s8\* path)** |
| --- | --- | --- |
| **Function** | |  |
| **Params** | **in** | **path** |
| **out** | **None** |
| **return** | | **MFSDK\_FS\_RET\_O**  **success**  **K** |
| **Other fail** |
| **remark** | |  |
| **demo** | |  |



## MfSdkFsRead

| **Prototype** | | **LIB\_EXPORT s32 MfSdkFsRead(FILE\_HANDLE handle, char**  **\*buffer, s32 size)** |
| --- | --- | --- |
| **Function** | | file read |
| **Params** | **in** | **handle** file handler |
| **size** read size |
| **out** | **buffer** read buffer |
| **return** | | **>0 success, the number of bytes actually read** |
| **Other fail** |
| **remark** | |  |
| **demo** | |  |

## MfSdkFsReadLine

| **Prototype** | | **LIB\_EXPORT s32 MfSdkFsReadLine(FILE\_HANDLE pFile, s8**  **\*pLineBuff,u32 LineBuffSize)** |
| --- | --- | --- |
| **Function** | | Read a line of text and support. |
| **Params** | **in** | **pFile** file handler |
| **size** Line Buff Size |
| **out** | **pLineBuff** Text data read |
| **return** | | **MFSDK\_FS\_RET\_O**  **success**  **K** |
| **Other fail** |
| **remark** | | **For text files, read one row of data from the current location and**  **jump to the next line.** |
| **demo** | |  |

## MfSdkFsReadProfileInt

**LIB\_EXPORT s32 MfSdkFsReadProfileInt(const s8\* section, const s8\* key, s32 default\_value, const s8\* file)**

****

| **Function** | | Read int value in initialization file. |
| --- | --- | --- |
| **Params** | **in** | **section** name of the section containing the key name |
| **key** name of the key pairs to value |
| **default\_value** default value of result |
| **file** path of the initialization file |
| **out** | **None** |
| **return** | | **profile int value,if read fail, return default value** |
| **remark** | |  |
| **demo** | |  |

## MfSdkFsReadProfileString

| **Prototype** | | **LIB\_EXPORT s32 MfSdkFsReadProfileString(const s8\* section, const s8\* key, s8\* value, s32 size, const s8\* default\_value, const**  **s8\* file)** |
| --- | --- | --- |
| **Function** | | Read string in initialization file. |
| **Params** | **in** | **section** name of the section containing the key name |
| **key** name of the key pairs to value |
| **size** size of result's buffer |
| **default\_value** default value of result |
| **file** path of the initialization file |
| **out** | pointer to the buffer that receives the retrieved  **value**  string |
| **return** | | **1 read success** |
| **0 read fail** |
| **remark** | |  |
| **demo** | |  |

## MfSdkFsUnlink

| **Prototype** | **LIB\_EXPORT s32 MfSdkFsUnlink(const s8 \* filename)** |
| --- | --- |
| **Function** | Delete file |



| **Params** | **in** | **filename** delete file name |
| --- | --- | --- |
| **out** | **None** |
| **return** | | **MFSDK\_FS\_RET\_O**  **success**  **K** |
| **Other fail,Ref. MfSdkFsRet\_E** |
| **remark** | |  |
| **demo** | |  |

## MfSdkFsWrite

| **Prototype** | | **LIB\_EXPORT s32 MfSdkFsWrite(FILE\_HANDLE handle, char**  **\*buffer, s32 size)** | |
| --- | --- | --- | --- |
| **Function** | | File writing | |
| **Params** | **in** | **handle** | File handle to be written |
| **buffer** | Data to be written |
| **size** | data sizeto be written |
| **out** | **None** | |
| **return** | | **Success.The number of bytes actually**  **>0**  **written.** | |
| **Other** | **fail,Ref. MfSdkFsRet\_E** |
| **remark** | |  | |
| **demo** | |  | |

## MfSdkFsWriteSync

| **Prototype** | | **s32 MfSdkFsWriteSync(FILE\_HANDLE handle, char \*buffer, s32 size, MFSDKBOOL bSync);** |
| --- | --- | --- |
| **Function** | | This includes writing through or flushing a disk cache if present |
| **Params** | **in** | **handle** File handle to be written |
| **buffer** Data to be written |
| **size** data sizeto be written |
| **out** | **None** |
| **return** | | **Success.The number of bytes actually**  **>0**  **written.** |



|  | **Other fail,Ref. MfSdkFsRet\_E** |
| --- | --- |
| **remark** |  |
| **demo** |  |

## MfSdkFsWriteBlockByName

| **Prototype** | | **LIB\_EXPORT s32 MfSdkFsWriteBlockByName(s8 \* file\_name, s32 write\_start, s8 \* write\_buff, s32 write\_size)** |
| --- | --- | --- |
| **Function** | | write one block data |
| **Params** | **in** | **file\_name** File handle to be written |
| **write\_start** Data to be written |
| **write\_buff** data sizeto be written |
| **write\_size** write data count |
| **out** | **None** |
| **return** | | **Success.The number of bytes actually**  **>0**  **written.** |
| **Other fail,Ref. MfSdkFsRet\_E** |
| **remark** | |  |
| **demo** | |  |

## MfSdkFsWriteProfileInt

| **Prototype** | | **LIB\_EXPORT s32 MfSdkFsWriteProfileInt(const s8\* section, const s8\* key, s32 value, const s8\* file)** |
| --- | --- | --- |
| **Function** | | write a profile int to a ini file |
| **Params** | **in** | name of the section,can't be NULL and empty  **section**  string |
| name of the key pairs to value, can't be NULL  **key**  and empty string |
| **value** profile int value |
| **file** file path of ini file |
| **out** | **None** |
| **return** | | **1 Success** |
| **0 fail** |



| **remark** |  |
| --- | --- |
| **demo** |  |

## MfSdkFsWriteProfileString

| **Prototype** | | **LIB\_EXPORT s32 MfSdkFsWriteProfileString(const s8\* section, const s8\* key, const s8\* value, const s8\* file)** |
| --- | --- | --- |
| **Function** | | write a profile string to a ini file |
| **Params** | **in** | name of the section,can't be NULL and empty  **section**  string |
| name of the key pairs to value, can't be NULL  **key**  and empty string |
| **value** profile string value |
| **file** file path of ini file |
| **out** | **None** |
| **return** | | **1 Success** |
| **0 fail** |
| **remark** | |  |
| **demo** | |  |

## MfSdkFsRenamePath

| **Prototype** | | **LIB\_EXPORT s32 MfSdkFsRenamePath(const char \*oldName, const char \*newName)** |
| --- | --- | --- |
| **Function** | | renames a file, moving it between directories if required |
| **Params** | **in** | **oldName** |
| **newName** |
| **out** | **None** |
| **return** | | **MFSDK\_FS\_RET\_O**  **Success**  **K** |
| **MFSDK\_FS\_RET\_P**  **Input params NULL**  **ARM\_ERROR** |
| **Other Failed** |
| **remark** | |  |



**demo** MfSdkFsRenamePath("testdir3\\test.txt", "testdir4\\test.txt");

## MfSdkFsClearFile

| **Prototype** | | **s32 MfSdkFsClearFile(cchar \*FileName);** |
| --- | --- | --- |
| **Function** | | truncate to 0 |
| **Params** | **in** | **FileName** File Name |
| **out** | **None** |
| **return** | | **MFSDK\_FS\_RET\_O**  **Success**  **K** |
| **MFSDK\_FS\_RET\_P**  **Input params NULL**  **ARM\_ERROR** |
| **Other Failed** |
| **remark** | |  |
| **demo** | |  |

## MfSdkFsDelDirFiles

| **Prototype** | | **s32 MfSdkFsDelDirFiles(const char\* path);** |
| --- | --- | --- |
| **Function** | | Delete the files in the directory |
| **Params** | **in** | **path** path |
| **out** | **None** |
| **return** | | **Ref. MfSdkFsRet\_E** |
| **remark** | |  |
| **demo** | |  |

## MfSdkFsRenameA

| **Prototype** | **s32 MfSdkFsRenameA(const char \*oldpath, const char**  **\*newpath);** |
| --- | --- |



| **Function** | | File or directory rename. |
| --- | --- | --- |
| **Params** | **in** | **oldpath** old path |
| **out** | **newpath** new path |
| **return** | | **Ref. MfSdkFsRet\_E** |
| **remark** | |  |
| **demo** | |  |

## MfSdkFsGetFileLength

| **Prototype** | | **LIB\_EXPORT s32 MfSdkFsGetFileLength(const char \*fileName)** |
| --- | --- | --- |
| **Function** | | Get file length. |
| **Params** | **in** | **fileName** File name |
| **out** | **None** |
| **return** | | **< 0 Failed** |
| **>=0 File length** |
| **remark** | |  |
| **demo** | |  |

## MfSdkFsRmdir

| **Prototype** | | **LIB\_EXPORT s32 MfSdkFsRmDir(const char \*path)** |
| --- | --- | --- |
| **Function** | | **Remove dir** |
| **Params** | **in** | **path** Dir path |
| **out** | **None** |
| **return** | | **MFSDK\_FS\_RET\_O**  **success**  **K** |
| **Other fail** |
| **remark** | |  |
| **demo** | | **if(MfSdkFsRmDir(“exdata\\testdir”) == MFSDK\_FS\_RET\_OK)**  **{** |



**// testdir will be removed**

**}**

## MfSdkFsOpenDir

| **Prototype** | | **LIB\_EXPORT void \*MfSdkFsOpenDir(const char \*path)** |
| --- | --- | --- |
| **Function** | | **Open dir** |
| **Params** | **in** | **path** Dir path |
| **out** | **None** |
| **return** | | **Dir pointer success** |
| **NULL fail** |
| **remark** | |  |
| **demo** | | **void \*pdp = MfSdkFsOpenDir(dirname); if(pdp != NULL)**  **{**  **MfSdkDirent\_T \*ent = NULL;**  **while((ent = MfSdkFsReadDir(pdp)) != NULL)**  **{**  **FS\_TRACE("ql\_readdir ent->d\_type=%d ,ent->d\_name: %s ,ent->d\_ino:%d\r\n", ent->d\_type, ent->d\_name, ent->d\_ino);**  **}**  **MfSdkFsCloseDir(pdp);**  **}**  **else**  **{**  **FS\_TRACE("folder %s does not exist\r\n",dirname);**  **}** |

## MfSdkFsReadDir

| **Prototype** | | **LIB\_EXPORT MfSdkDirent\_T\* MfSdkFsReadDir(void \*dp)** |
| --- | --- | --- |
| **Function** | | **Read dir** |
| **Params** | **in** | **dp** Dir pointer |



|  | **out** | **None** |
| --- | --- | --- |
| **return** | | **pointer success** |
| **NULL fail** |
| **remark** | |  |
| **demo** | | **void \*pdp = MfSdkFsOpenDir(dirname); if(pdp != NULL)**  **{**  **MfSdkDirent\_T \*ent = NULL;**  **while((ent = MfSdkFsReadDir(pdp)) != NULL)**  **{**  **FS\_TRACE("ql\_readdir ent->d\_type=%d ,ent->d\_name: %s ,ent->d\_ino:%d\r\n", ent->d\_type, ent->d\_name, ent->d\_ino);**  **}**  **MfSdkFsCloseDir(pdp);**  **}**  **else**  **{**  **FS\_TRACE("folder %s does not exist\r\n",dirname);**  **}** |

## MfSdkFsCloseDir

| **Prototype** | | **LIB\_EXPORT s32 MfSdkFsCloseDir(void \*dp)** |
| --- | --- | --- |
| **Function** | | **Close dir** |
| **Params** | **in** | **path** Dir path |
| **out** | **None** |
| **return** | | **Dir pointer success** |
| **NULL fail** |
| **remark** | |  |
| **demo** | | **void \*pdp = MfSdkFsOpenDir(dirname); if(pdp != NULL)**  **{**  **MfSdkDirent\_T \*ent = NULL;**  **while((ent = MfSdkFsReadDir(pdp)) != NULL)**  **{**  **FS\_TRACE("ql\_readdir ent->d\_type=%d ,ent->d\_name: %s ,ent->d\_ino:%d\r\n", ent->d\_type, ent->d\_name, ent->d\_ino);** |



**}**

**MfSdkFsCloseDir(pdp);**

**}**

**else**

**{**

**FS\_TRACE("folder %s does not exist\r\n",dirname);**

**}**

# 14FIFO module

## Module description

FIFO stands for "First In, First Out," and it refers to a method of organizing and manipulating data structures, particularly in computing and queue management. In a FIFO system, the first item that is added is the first one to be removed. It operates on the principle that the first element added to a collection is the first one to be removed.

## Module structure declaration



## Constant declarations





## MfSdkFifoCreate

| **Prototype** | | **LIB\_EXPORT MFSDKBOOL MfSdkFifoCreate(MfSdkFifoData\_T**  **\* pstFifo , s32 nSize )** |
| --- | --- | --- |
| **Function** | | Create fifo |
| **Params** | **in** | **pstFifo** MfSdkFifoData\_T |
| **nSize** FIFO size |
| **out** | **Nothing** |
| **return** | | **MFSDK\_TRUE Success** |
| **MFSDK\_FALSE Failed** |
| **remark** | |  |
| **demo** | | MfSdkFifoData\_T playFifo; memset(&playFifo,0,sizeof(MfSdkFifoData\_T)); MFSDKBOOL b = MfSdkFifiCreate(&playFifo,4096); if(b)  {  //TODO success  }  else  {  //TODO failed  } |

## MfSdkFifoGet

| **Prototype** | | **LIB\_EXPORT s32 MfSdkFifoGet(MfSdkFifoData\_T\* pstFifo,u8 \* pData , s32 nLen)** | |
| --- | --- | --- | --- |
| **Function** | | Get FIFO data. | |
| **Params** | **in** | **pstFifo** | MfSdkFifoData\_T |
| **nLen** | pData buffer max size |
| **out** | **pData** | Get FIFO data |
| **return** | | **> 0** | **Get FIFO data length** |
| **other** | **Failed** |
| **remark** | |  | |
| **demo** | |  | |



## MfSdkFifoInit

| **Prototype** | | **LIB\_EXPORT MFSDKBOOL MfSdkFifoInit(MfSdkFifoData\_T\* pstFifo)** |
| --- | --- | --- |
| **Function** | | Init FIFO |
| **Params** | **in** | **pstFifo** MfSdkFifoData\_T |
| **out** | **Nothing** |
| **return** | | **MFSDK\_TRUE Success** |
| **MFSDK\_FALSE Failed** |
| **remark** | |  |
| **demo** | | MfSdkFifoData\_T playFifo; memset(&playFifo,0,sizeof(MfSdkFifoData\_T)); MFSDKBOOL b = MfSdkFifiCreate(&playFifo,4096); if(b)  {  //TODO success MfSdkFifoInit(&playFifo);  }  else  {  //TODO failed  } |

## MfSdkFifoIsEmpty

| **Prototype** | | **LIB\_EXPORT MFSDKBOOL MfSdkFifoIsEmpty(MfSdkFifoData\_T\* pstFifo)** |
| --- | --- | --- |
| **Function** | | Check whether the queue is empty. |
| **Params** | **in** | **pstFifo** MfSdkFifoData\_T |
| **out** | **Nothing** |
| **return** | | **MFSDK\_TRUE Empty** |
| **MFSDK\_FALSE Not empty** |



| **remark** |  |
| --- | --- |
| **demo** | MfSdkFifoData\_T playFifo; if(MfSdkFifoIsEmpty(&pstFifo))  {  //TODO Empty  } |

## MfSdkFifoIsFull

| **Prototype** | | **LIB\_EXPORT MFSDKBOOL MfSdkFifoIsFull(MfSdkFifoData\_T\* pstFifo)** |
| --- | --- | --- |
| **Function** | | Check whether the queue is full. |
| **Params** | **in** | **pstFifo** MfSdkFifoData\_T |
| **out** | **Nothing** |
| **return** | | **MFSDK\_TRUE Full** |
| **MFSDK\_FALSE Not full** |
| **remark** | |  |
| **demo** | | MfSdkFifoData\_T playFifo; if(MfSdkFifoIsFull(&pstFifo))  {  //TODO Full  } |

## MfSdkFifoPut

| **Prototype** | | **LIB\_EXPORT s32 MfSdkFifoPut(MfSdkFifoData\_T\* pstFifo, u8 \* pData , s32 nLen)** |
| --- | --- | --- |
| **Function** | | Insert data to FIFO. |
| **Params** | **in** | **pstFifo** MfSdkFifoData\_T |
| **pData** Insert data buffer |
| **nLen** Insert data buffer length |
| **out** | **Nothing** |
| **return** | | **> 0 Actual data length** |
| **< 0 Failed** |



| **remark** |  |
| --- | --- |
| **demo** | MfSdkFifoData\_T playFifo; char data[128] = {0}; Strcpy(data,"volmax.mp3");  s32 ilen = MfSdkFifoPut(&playFifo,data ,strlen(data)); If(ilen != strlen(data))  {  //TODO failed  } |

## MfSdkFifoResize

| **Prototype** | | **LIB\_EXPORT MFSDKBOOL MfSdkFifoResize(MfSdkFifoData\_T**  **\* pstFifo , s32 nSize )** |
| --- | --- | --- |
| **Function** | | Change the FIFO size. |
| **Params** | **in** | **pstFifo** MfSdkFifoData\_T |
| **nSize** FIFO size |
| **out** | **Nothing** |
| **return** | | **MFSDK\_TRUE Success** |
| **MFSDK\_FALSE Failed** |
| **remark** | |  |
| **demo** | | MfSdkFifoData\_T playFifo; memset(&playFifo,0,sizeof(MfSdkFifoData\_T)); MFSDKBOOL b = MfSdkFifoResize(&playFifo,4096); if(b)  {  //TODO success  }  else  {  //TODO failed  } |

## MfSdkGetFifoNum

**Prototype LIB\_EXPORT s32 MfSdkGetFifoNum(MfSdkFifoData\_T \***

****

| **Function** | | Get the number of elements in the FIFO. |
| --- | --- | --- |
| **Params** | **in** | **pstFifo** MfSdkFifoData\_T |
| **out** | **Nothing** |
| **return** | | **> 0 The number of elements in the FIFO** |
| **MFSDK\_FIFO\_PARM\_E**  **Param invalid**  **RROR** |
| **remark** | |  |
| **demo** | | MfSdkFifoData\_T playFifo;  s32 fifo\_len = MfSdkGetFifoNum(&playFifo); |

## MfSdkGetGetFifoSize

| **Prototype** | | **LIB\_EXPORT s32 MfSdkGetGetFifoSize(MfSdkFifoData\_T \* pstFifo)** |
| --- | --- | --- |
| **Function** | | Get fifo size. |
| **Params** | **in** | **pstFifo** MfSdkFifoData\_T |
| **out** | **Nothing** |
| **return** | | **> 0 FIFO size** |
| **MFSDK\_FIFO\_PARM\_E**  **Param invalid**  **RROR** |
| **remark** | |  |
| **demo** | | MfSdkFifoData\_T playFifo;  s32 fifoSize = MfSdkGetGetFifoSize(&playFifo); |

# Audio module

## Module description

This module mainly includes Audio APIs.

## Module structure declaration

typedef enum

{

MFSDK\_AUD\_PLAY\_UNIT\_MIN = 0, MFSDK\_AUD\_PLAY\_UNIT\_TEN = MFSDK\_AUD\_PLAY\_UNIT\_MIN, MFSDK\_AUD\_PLAY\_UNIT\_HUNDRED,



MFSDK\_AUD\_PLAY\_UNIT\_THOUSAND, MFSDK\_AUD\_PLAY\_UNIT\_TEN\_THOUSAND, MFSDK\_AUD\_PLAY\_UNIT\_YUAN, MFSDK\_AUD\_PLAY\_UNIT\_MAX

}MfSdkAudPlayUnit\_T;

## Constant declarations

typedef enum

{

MFSDK\_AUD\_PAY\_MIN = 0, MFSDK\_AUD\_PAY\_PROCESSING = MFSDK\_AUD\_PAY\_MIN, MFSDK\_AUD\_PAY\_FAIL,

MFSDK\_AUD\_PAY\_SUCCESS, MFSDK\_AUD\_PAY\_SCAN\_SUCCESS, MFSDK\_AUD\_PAY\_VOID\_SUCCESS, MFSDK\_AUD\_PAY\_MAX,

}MfSdkAudPayResult\_E;

typedef enum

{

MFSDK\_AUD\_PLAY\_NUM\_MIN = 0, MFSDK\_AUD\_PLAY\_NUM\_ZERO = MFSDK\_AUD\_PLAY\_NUM\_MIN, MFSDK\_AUD\_PLAY\_NUM\_ONE,

MFSDK\_AUD\_PLAY\_NUM\_TWO, MFSDK\_AUD\_PLAY\_NUM\_THREE, MFSDK\_AUD\_PLAY\_NUM\_FOUR, MFSDK\_AUD\_PLAY\_NUM\_FIVE, MFSDK\_AUD\_PLAY\_NUM\_SIX, MFSDK\_AUD\_PLAY\_NUM\_SEVEN, MFSDK\_AUD\_PLAY\_NUM\_EIGHT, MFSDK\_AUD\_PLAY\_NUM\_NINE, MFSDK\_AUD\_PLAY\_NUM\_DOT, MFSDK\_AUD\_PLAY\_NUM\_MAX

}MfSdkAudPlayNum\_E;

typedef enum

{

MFSDK\_AUD\_PAY\_TYPE\_MIN = 0, MFSDK\_AUD\_PAY\_TYPE\_WECHAT =

MFSDK\_AUD\_PAY\_TYPE\_MIN, MFSDK\_AUD\_PAY\_TYPE\_ALIPAY, MFSDK\_AUD\_PAY\_TYPE\_UNIONPAY, MFSDK\_AUD\_PAY\_TYPE\_CARD,



MFSDK\_AUD\_PAY\_TYPE\_SCAN = 7, MFSDK\_AUD\_PAY\_TYPE\_MAX

}MfSdkAudPlayPayType\_E;

typedef enum

{

MFSDK\_AUD\_RET\_FAILED = -3, //failed MFSDK\_AUD\_RET\_BOUNDS = -2, //Array out-of-bounds MFSDK\_AUD\_RET\_PARM\_ERROR = -1, //check param MFSDK\_AUD\_RET\_OK = 0,

}MfSdkAudRet\_E;

## MfSdkAudPlayVoice

| **Prototype** | | **LIB\_EXPORT void MfSdkAudPlayVoice(s8 \*msg)** |
| --- | --- | --- |
| **Function** | | Play Audio file. |
| **Params** | **in** | **msg** |
| **out** | **Nothing** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | |  |

## MfSdkAudPlayAmt

| **Prototype** | | **LIB\_EXPORT s32 MfSdkAudPlayAmt(s32 amount)** |
| --- | --- | --- |
| **Function** | | Play Amt audio. |
| **Params** | **in** | **amount** |
| **out** | **Nothing** |
| **return** | | **Ref. MfSdkAudRet\_E** |
| **remark** | |  |
| **demo** | |  |



## MfSdkAudPlayBatteryLevel

| **Prototype** | | **LIB\_EXPORT s32 MfSdkAudPlayBatteryLevel(void)** |
| --- | --- | --- |
| **Function** | | Play battery level. |
| **Params** | **in** | **Nothing** |
| **out** | **Nothing** |
| **return** | | **Ref. MfSdkAudRet\_E** |
| **remark** | |  |
| **demo** | |  |

## MfSdkAudPlayFile

| **Prototype** | | **LIB\_EXPORT s32 MfSdkAudPlayFile(const s8 \* folder, const s8**  **\* sndfile)** |
| --- | --- | --- |
| **Function** | | Play audio file. |
| **Params** | **in** | **Nothing** |
| **out** | **Nothing** |
| **return** | | **Ref. MfSdkAudRet\_E** |
| **remark** | | **pub\_tts\_play and pub\_tts\_playOpt** |
| **demo** | |  |

## MfSdkAudPlay

| **Prototype** | | **LIB\_EXPORT s32 MfSdkAudPlay(const s8 \* sndfile)** |
| --- | --- | --- |
| **Function** | | Play audio file. |
| **Params** | **in** | **sndfile** |
| **out** | **Nothing** |
| **return** | | **Ref. MfSdkAudRet\_E** |
| **remark** | |  |



**demo**

## MfSdkAudPlayNum

| **Prototype** | | **LIB\_EXPORT s32 MfSdkAudPlayNum(MfSdkAudPlayNum\_E index)** |
| --- | --- | --- |
| **Function** | |  |
| **Params** | **in** | **index** |
| **out** | **Nothing** |
| **return** | | **Ref. MfSdkAudRet\_E** |
| **remark** | |  |
| **demo** | |  |

## MfSdkAudPlayNumber

| **Prototype** | | **LIB\_EXPORT s32 MfSdkAudPlayNumber(s32 num)** |
| --- | --- | --- |
| **Function** | |  |
| **Params** | **in** | **num** |
| **out** | **Nothing** |
| **return** | | **Ref. MfSdkAudRet\_E** |
| **remark** | |  |
| **demo** | |  |

## MfSdkAudPlayPayResult

| **Prototype** | | **LIB\_EXPORT s32**  **MfSdkAudPlayPayResult(MfSdkAudPayResult\_E index)** |
| --- | --- | --- |
| **Function** | |  |
| **Params** | **in** | **index** |
| **out** | **Nothing** |



| **return** | **Ref. MfSdkAudRet\_E** |
| --- | --- |
| **remark** |  |
| **demo** |  |

## MfSdkAudPlayPayType

| **Prototype** | | **LIB\_EXPORT s32**  **MfSdkAudPlayPayType(MfSdkAudPlayPayType\_E payType)** |
| --- | --- | --- |
| **Function** | |  |
| **Params** | **in** | **index** |
| **out** | **Nothing** |
| **return** | | **Ref. MfSdkAudRet\_E** |
| **remark** | |  |
| **demo** | |  |

## MfSdkAudPlayUnit

| **Prototype** | | **LIB\_EXPORT s32 MfSdkAudPlayUnit(MfSdkAudPlayUnit\_T index)** |
| --- | --- | --- |
| **Function** | |  |
| **Params** | **in** | **index** Ref. MfSdkAudPlayUnit\_T |
| **out** | **Nothing** |
| **return** | | **Ref. MfSdkAudRet\_E** |
| **remark** | |  |
| **demo** | |  |

## MfSdkAudTtsState

| **Prototype** | **LIB\_EXPORT MFSDKBOOL MfSdkAudTtsState(void)** |
| --- | --- |
| **Function** |  |



| **Params** | **in** | **Nothing** |
| --- | --- | --- |
| **out** | **Nothing** |
| **return** | | **MFSDK\_TRUE success** |
| **MFSDK\_FALSE fail** |
| **remark** | |  |
| **demo** | | if(MfSdkAudTtsState())  {  //busy  }  else{  //TODO  } |

## MfSdkAudBatchBegin

| **Prototype** | | **LIB\_EXPORT void MfSdkAudBatchBegin(void)** |
| --- | --- | --- |
| **Function** | |  |
| **Params** | **in** | **Nothing** |
| **out** | **Nothing** |
| **return** | | **Nothing** |
| **remark** | | **Deprecated** |
| **demo** | |  |

## MfSdkAudBathcEnd

| **Prototype** | | **LIB\_EXPORT void MfSdkAudBathcEnd(void)** |
| --- | --- | --- |
| **Function** | |  |
| **Params** | **in** | **Nothing** |
| **out** | **Nothing** |
| **return** | | **Nothing** |
| **remark** | | **Deprecated** |
| **demo** | |  |



## MfSdkAudClear

| **Prototype** | | **LIB\_EXPORT void MfSdkAudClear(void)** |
| --- | --- | --- |
| **Function** | |  |
| **Params** | **in** | **Nothing** |
| **out** | **Nothing** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | |  |

## MfSdkAudTtsPlay

| **Prototype** | | **LIB\_EXPORT s32 MfSdkAudTtsPlay(s8 \*msg)** |
| --- | --- | --- |
| **Function** | | TTS |
| **Params** | **in** | **msg** Text Message |
| **out** | **Nothing** |
| **return** | | **Ref. MfSdkAudRet\_E** |
| **remark** | |  |
| **demo** | | **MfSdkAudTtsPlay(“exdata\\welc.mp3”);** |

## MfSdkAudSetVolume

| **Prototype** | | **LIB\_EXPORT void MfSdkAudSetVolume(s32 val)** |
| --- | --- | --- |
| **Function** | | set audio volume |
| **Params** | **in** | **val** |
| **out** | **Nothing** |
| **return** | | **Nothing** |
| **remark** | |  |



**demo**

## MfSdkAudSetVolumeRunning

| **Prototype** | | **LIB\_EXPORT void MfSdkAudSetVolumeRunning(s32 val)** |
| --- | --- | --- |
| **Function** | | Set the volume when playing voice |
| **Params** | **in** | **val** |
| **out** | **Nothing** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | |  |

## MfSdkAudGetVolume

| **Prototype** | | **LIB\_EXPORT s32 MfSdkAudGetVolume()** |
| --- | --- | --- |
| **Function** | | Get the volume size |
| **Params** | **in** | **Nothing** |
| **out** | **Nothing** |
| **return** | | **Volume** |
| **remark** | |  |
| **demo** | |  |

## MfSdkAudSetSpeed

| **Prototype** | | **LIB\_EXPORT void MfSdkAudSetSpeed(s32 val)** |
| --- | --- | --- |
| **Function** | | Set audio play speed |
| **Params** | **in** | **val** |
| **out** | **Nothing** |



| **return** | **Nothing** |
| --- | --- |
| **remark** | **unimplemented** |
| **demo** |  |

# KeyBoard module

## Module description

This module mainly includes APIs related to KeyBoard.

## Module structure declaration

None.

## Constant declarations

typedef enum

{

MFSDK\_KB\_RET\_TIMEOUT = -4, //time over MFSDK\_KB\_RET\_FAILED = -3, //failed MFSDK\_KB\_RET\_BOUNDS = -2, //Array out-of-bounds MFSDK\_KB\_RET\_PARM\_ERROR = -1, //check param MFSDK\_KB\_RET\_OK = 0,

}MfSdkKbRet\_E;

typedef enum

{

MFSDK\_VK\_INVALID = -1,

MFSDK\_VK\_0 = 0x30, MFSDK\_VK\_1 = 0x31, MFSDK\_VK\_2 = 0x32, MFSDK\_VK\_3 = 0x33, MFSDK\_VK\_4 = 0x34, MFSDK\_VK\_5 = 0x35, MFSDK\_VK\_6 = 0x36, MFSDK\_VK\_7 = 0x37, MFSDK\_VK\_8 = 0x38, MFSDK\_VK\_9 = 0x39,

MFSDK\_VK\_CTL = 100, MFSDK\_VK\_UP , MFSDK\_VK\_DOWN, MFSDK\_VK\_LEFT,



MFSDK\_VK\_RIGHT,

MFSDK\_VK\_F1, MFSDK\_VK\_F2, MFSDK\_VK\_BACKSPACE, MFSDK\_VK\_ESC, MFSDK\_VK\_ENTER,

MFSDK\_VK\_XING, // start MFSDK\_VK\_JING, //Hash

}MfSdkKbKeyCode\_E;

## MfSdkKbKeySetParam

| **Prototype** | | **LIB\_EXPORT void MfSdkKbKeySetParam(int key, int short\_press\_time, int long\_press\_time, int short\_press\_delay)** |
| --- | --- | --- |
| **Function** | |  |
| **Params** | **in** | **key** |
| **short\_press\_time** |
| **long\_press\_time** |
| **short\_press\_delay** |
| **out** | **Nothing** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | |  |

## MfSdkKbGetKeySound

| **Prototype** | | **LIB\_EXPORT s32 MfSdkKbGetKeySound()** |
| --- | --- | --- |
| **Function** | |  |
| **Params** | **in** | **Nothing** |
| **out** | **Nothing** |
| **return** | | **0 close** |
| **1 open** |
| **remark** | |  |



**demo**

## MfSdkKbSetKeySound

| **Prototype** | | **LIB\_EXPORT void MfSdkKbSetKeySound(s32 nOpen)** |
| --- | --- | --- |
| **Function** | |  |
| **Params** | **in** | **nOpen** 1 open; 0 close |
| **out** | **Nothing** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | |  |

## MfSdkKbWaitKey

| **Prototype** | | **LIB\_EXPORT s32 MfSdkKbWaitKey(s32 TimeOut)** |
| --- | --- | --- |
| **Function** | | Waiting button, waits for the button within the set time, while the Nothing button waits for the timeout. |
| **Params** | **in** | Waiting timeout time (seconds), 0 means  **TimeOut**  blocking |
| **out** | **Nothing** |
| **return** | | **>=0 key value** |
| **Other fail** |
| **remark** | |  |
| **demo** | |  |

# Lcd module

## Module description

This module mainly includes APIs related to LCD.

## Module structure declaration

None.



## Constant declarations

typedef enum

{

MFSDK\_LCD\_OFF = 0, //turn off backlight MFSDK\_LCD\_AUTO = 1,// system automatically controls MFSDK\_LCD\_ON = 2, //backlight is always bright

}MfSdkLcdStatus\_E;

## MfSdkLcdBackLight

| **Prototype** | | **LIB\_EXPORT s32 MfSdkLcdBackLight(s32 status)** |
| --- | --- | --- |
| **Function** | |  |
| **Params** | **in** | **status** Ref. MfSdkLcdStatus\_E |



|  |  | MFSDK\_LCD\_OFF-OFF,MFSDK\_LCD\_O  N-ON |
| --- | --- | --- |
| **out** | **Nothing** |
| **return** | | **Ref. MfSdkLcdRet\_E** |
| **remark** | |  |
| **demo** | |  |

## MfSdkLcdSegmentBackLight

| **Prototype** | | **LIB\_EXPORT s32 MfSdkLcdSegmentBackLight(s32 status)** |
| --- | --- | --- |
| **Function** | | Segment code LCD |
| **Params** | **in** | Ref. MfSdkLcdStatus\_E  **status** MFSDK\_LCD\_OFF-OFF,MFSDK\_LCD\_O N-ON |
| **out** | **Nothing** |
| **return** | | **Ref. MfSdkLcdRet\_E** |
| **remark** | |  |
| **demo** | |  |

## MfSdkLcdGetSubProbe

| **Prototype** | | **LIB\_EXPORT s32 MfSdkLcdGetSubProbe()** |
| --- | --- | --- |
| **Function** | |  |
| **Params** | **in** | **Nothing** |
| **out** | **Nothing** |
| **return** | | **Ref. MfSdkLcdRet\_E** |
| **remark** | | **unimplemented** |
| **demo** | |  |



## MfSdkLcdSetIndex

| **Prototype** | | **LIB\_EXPORT void MfSdkLcdSetIndex(s32 index)** |
| --- | --- | --- |
| **Function** | |  |
| **Params** | **in** | **index** |
| **out** | **Nothing** |
| **return** | | **Nothing** |
| **remark** | | **unimplemented** |
| **demo** | |  |

## MfSdkLcdGetPowerDownTime

| **Prototype** | | **LIB\_EXPORT s32 MfSdkLcdGetPowerDownTime()** |
| --- | --- | --- |
| **Function** | | Get the shutdown time |
| **Params** | **in** | **Nothing** |
| **out** | **Nothing** |
| **return** | | **Shutdown time** |
| **remark** | |  |
| **demo** | |  |

## MfSdkLcdSetPowerDownTime

| **Prototype** | | **LIB\_EXPORT void MfSdkLcdSetPowerDownTime(s32 ntime)** |
| --- | --- | --- |
| **Function** | | Set the shutdown time |
| **Params** | **in** | **ntime** Shutdown time unit: second |
| **out** | **Nothing** |
| **return** | | **Nothing** |
| **remark** | |  |



**demo**

## MfSdkLcdGetBackLightTime

| **Prototype** | | **LIB\_EXPORT s32 MfSdkLcdGetBackLightTime()** |
| --- | --- | --- |
| **Function** | | Get BackLight time |
| **Params** | **in** | **Nothing** |
| **out** | **Nothing** |
| **return** | | **Backlight time** |
| **remark** | |  |
| **demo** | |  |

## MfSdkLcdSetBackLightTime

| **Prototype** | | **LIB\_EXPORT void MfSdkLcdSetBackLightTime(s32 ntime)** |
| --- | --- | --- |
| **Function** | | Set backlight time |
| **Params** | **in** | **ntime** Backlight time |
| **out** | **Nothing** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | |  |

## MfSdkLcdGetFrontBackLightTime

| **Prototype** | | **s32 MfSdkLcdGetFrontBackLightTime();** |
| --- | --- | --- |
| **Function** | | Get Front Screen Backlight time (for MP70A6) |
| **Params** | **in** | **None** |
| **out** | **None** |



| **return** | **Backlight time** |
| --- | --- |
| **remark** |  |
| **demo** |  |

## MfSdkLcdSetFrontBackLightTime

| **Prototype** | | **void MfSdkLcdSetFrontBackLightTime(s32 ntime);** |
| --- | --- | --- |
| **Function** | | Set Front Screen Backlight time (for MP70A6) |
| **Params** | **in** | Backlight time,max value 99999999  **ntime**  unit:second |
| **out** | **None** |
| **return** | | **None** |
| **remark** | |  |
| **demo** | |  |

## MfSdkLcdGetRearBackLightTime

| **Prototype** | | **s32 MfSdkLcdGetRearBackLightTime();** |
| --- | --- | --- |
| **Function** | | Get Rear Screen Backlight time (for MP70A6) |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **Backlight time** |
| **remark** | |  |
| **demo** | |  |

## MfSdkLcdSetRearBackLightTime

**Prototype void MfSdkLcdSetRearBackLightTime(s32 ntime);**

****

| **Function** | | Set Rear Screen Backlight time (for MP70A6) |
| --- | --- | --- |
| **Params** | **in** | Backlight time,max value 99999999  **ntime**  unit:second |
| **out** | **None** |
| **return** | | **None** |
| **remark** | |  |
| **demo** | |  |

## MfSdkLcdBrightnessLevelSettings

| **Prototype** | | **s32 MfSdkLcdBrightnessLevelSettings(MfSdkLcdBrighnesstLevel\_E**  **level);** |
| --- | --- | --- |
| **Function** | | Screen Brightness Level Settings |
| **Params** | **in** | **level** MfSdkLcdBrighnesstLevel\_E level 0~5 |
| **out** | **None** |
| **return** | | **Refer to MfSdkLcdRet\_E** |
| **remark** | |  |
| **demo** | |  |

## MfSdkLcdAutoFlush

| **Prototype** | | **s32 MfSdkLcdAutoFlush(MFSDKBOOL bValue);** |
| --- | --- | --- |
| **Function** | | Whether to auto Refresh the LCD |
| **Params** | **in** | true-enble flush lcd,false-disable flush  **bValue**  lcd |
| **out** | **None** |
| **return** | | **Refer to MfSdkLcdRet\_E** |
| **remark** | |  |
| **demo** | |  |



## MfSdkLcdSetNormalDirection

| **Prototype** | | **s32 MfSdkLcdSetNormalDirection();** |
| --- | --- | --- |
| **Function** | | Return to normal display |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **Refer to MfSdkLcdRet\_E** |
| **remark** | |  |
| **demo** | |  |

## MfSdkLcdArrowDisplay

| **Prototype** | | **void MfSdkLcdArrowDisplay(MfSdkLcdArrow\_E index, MFSDKBOOL enable);** |
| --- | --- | --- |
| **Function** | | Status Bar Display Up/Down Arrow.only support SR600mini |
| **Params** | **in** | **index** 0:up,1:down |
| **enable** 1:on,0:off |
| **out** | **None** |
| **return** | | **None** |
| **remark** | |  |
| **demo** | |  |

## MfSdkLcdBacklightIsBright

| **Prototype** | | **MFSDKBOOL MfSdkLcdBacklightIsBright();** |
| --- | --- | --- |
| **Function** | | Check the screen backlight status |
| **Params** | **in** | **None** |
| **out** | **None** |
| **return** | | **1** on |



|  | **0** off |
| --- | --- |
| **remark** |  |
| **demo** |  |

# Log module

## Module description

This module mainly includes APIs about log output.

## Module structure declaration

None.

## Constant declarations

//close log output

#define MFSDK\_LOG\_CLOSE\_OUT (0)

//The logs will be output from the USB port #define MFSDK\_LOG\_USB\_OUT (1)

//The logs will be output from the wifi port #define MFSDK\_LOG\_WIFI\_OUT (2)

typedef enum

{

MFSDK\_LOG\_LEVEL\_TRACE = 0, //trace MFSDK\_LOG\_LEVEL\_DEBUG, //debug MFSDK\_LOG\_LEVEL\_INFO, //info MFSDK\_LOG\_LEVEL\_WARN, //warning MFSDK\_LOG\_LEVEL\_ERROR,//error MFSDK\_LOG\_LEVEL\_FATAL, MFSDK\_LOG\_LEVEL\_FILE,

}MfSdkLogLevel\_E;

## MfSdkLogSoundSet

| **Prototype** | **LIB\_EXPORT s32 MfSdkLogSoundSet(s32 val)** |
| --- | --- |
| **Function** | Set the log output mode |



| **Params** | **in** | MFSDK\_LOG\_USB\_OUT/MFSDK\_LOG  **val**  \_WIFI\_OUT | |
| --- | --- | --- | --- |
| **out** | **Nothing** | |
| **return** | | **0** | success |
| **remark** | |  | |
| **demo** | |  | |

## MfSdkLog

| **Prototype** | | **LIB\_EXPORT void MfSdkLog(const char \*pTag,const char**  **\*fmt,...)** |
| --- | --- | --- |
| **Function** | | module is "MFSDKLOG" |
| **Params** | **in** | **pTag** |
| **fmt** |
| **...** |
| **out** | **Nothing** |
| **return** | | **Nothing** |
| **remark** | | **default level:MFSDK\_LOG\_LEVEL\_TRACE** |
| **demo** | |  |

## MfSdkLogTip

| **Prototype** | | **LIB\_EXPORT void MfSdkLogTip(const char\* module,**  **MfSdkLogLevel\_E level, const void\* Buffer, s32 nSize, char\* tip, s32 breakline)** |
| --- | --- | --- |
| **Function** | |  |
| **Params** | **in** | **module** |
| **level** |
| **Buffer** |
| **nSize** |
| **tip** |
| **breakline** |
| **out** | **Nothing** |
| **return** | | **Nothing** |



| **remark** |  |
| --- | --- |
| **demo** |  |

## MfSdkLogHexBuff

| **Prototype** | | **LIB\_EXPORT void MfSdkLogHexBuff(const char\* module, MfSdkLogLevel\_E level,s8\*Buffer, s32 nSize)** |
| --- | --- | --- |
| **Function** | |  |
| **Params** | **in** | **module** |
| **level** |
| **Buffer** |
| **nSize** |
| **out** | **Nothing** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | |  |

## MfSdkLogLevel

| **Prototype** | | **LIB\_EXPORT void MfSdkLogLevel(const char\* module, MfSdkLogLevel\_E level, const char\* fmt, ...)** |
| --- | --- | --- |
| **Function** | |  |
| **Params** | **in** | **module** |
| **level** |
| **fmt** |
| **...** |
| **out** | **Nothing** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | |  |



## MfSdkLogOutputSwitch

| **Prototype** | | **void MfSdkLogOutputSwitch(MFSDKBOOL enable);** |
| --- | --- | --- |
| **Function** | | default enable |
| **Params** | **in** | MFSDK\_TRUE: enable output of logs ,  **val** MFSDK\_FALSE: disenable output of logs |
| **out** | **Nothing** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | |  |

# Power module

## Module description

This module mainly includes APIs about power, battery and backlight time set.

## Module structure declaration

None.

## Constant declarations

typedef enum

{

MFSDK\_POWER\_RET\_UNSUPPORT = -4, // dont support MFSDK\_POWER\_RET\_FAILED = -3, //failed MFSDK\_POWER\_RET\_BOUNDS = -2, //Array out-of-bounds MFSDK\_POWER\_RET\_PARM\_ERROR = -1, //check param MFSDK\_POWER\_RET\_OK = 0, //success

}MfSdkPowerRet\_E;

typedef enum{ MFSDK\_POWER\_POWER\_MANAGER\_REBOOT, MFSDK\_POWER\_POWER\_MANAGER\_SHUTDOWN, MFSDK\_POWER\_MANAGER\_ENTER\_PAGE = 4,



}MfSdkPowerManagerType\_E;

typedef enum { MFSDK\_POWER\_SUPPLY\_CAPACITY\_LEVEL\_UNKNOWN = 0, MFSDK\_POWER\_SUPPLY\_CAPACITY\_LEVEL\_CRITICAL, MFSDK\_POWER\_SUPPLY\_CAPACITY\_LEVEL\_LOW, MFSDK\_POWER\_SUPPLY\_CAPACITY\_LEVEL\_NORMAL, MFSDK\_POWER\_SUPPLY\_CAPACITY\_LEVEL\_HIGH, MFSDK\_POWER\_SUPPLY\_CAPACITY\_LEVEL\_FULL,

}MfSdkPowerCapacityLevel\_E;

typedef enum

{

MFSDK\_POWER\_LOW\_BATTERY\_REMID = 0, MFSDK\_POWER\_LOW\_BATTERY\_SHUTDOWN,

}

MfSdkPowerPageParm\_E;

typedef enum

{

MFSDK\_POWER\_ACONLINE\_SLEEP\_DN = 0, // Unable to sleep when connected to external power

MFSDK\_POWER\_ACONLINE\_SLEEP\_EN = 1, // Enable to sleep when connected to external power

}MfSdkPowerSleep\_E;

typedef s32 (\*MfSdkPowerResumeProcCb)(s32 ret);

typedef s32 (\*MfSdkPowerManagerCb)(MfSdkPowerManagerType\_E type);

typedef void(\*MfSdkPowerPageCbFunc)(int);

## MfSdkPowerResumeProc

| **Prototype** | | **LIB\_EXPORT s32**  **MfSdkPowerResumeProc(MfSdkPowerResumeProcCb pFuncCb)** |
| --- | --- | --- |
| **Function** | | set resume callback |
| **Params** | **in** | **pFuncCb** |
| **out** | **Nothing** |
| **return** | | **Ref. MfSdkPowerRet\_E** |
| **remark** | |  |



**demo**

## MfSdkPowerManagerSetFunc

| **Prototype** | | **void MfSdkPowerManagerSetFunc(MfSdkPowerManagerCb cb);** |
| --- | --- | --- |
| **Function** | | set power manager callback |
| **Params** | **in** | **cb** power manager callback |
| **out** | **Nothing** |
| **return** | | **None** |
| **remark** | |  |
| **demo** | |  |

## MfSdkPowerLockApp

| **Prototype** | | **LIB\_EXPORT s32 MfSdkPowerLockApp(char \*sfun)** |
| --- | --- | --- |
| **Function** | |  |
| **Params** | **in** | **sfun** |
| **out** | **Nothing** |
| **return** | | **Ref. MfSdkPowerRet\_E** |
| **remark** | |  |
| **demo** | |  |

## MfSdkPowerUnlockApp

| **Prototype** | | **LIB\_EXPORT void MfSdkPowerUnlockApp(void)** |
| --- | --- | --- |
| **Function** | |  |
| **Params** | **in** | **Nothing** |
| **out** | **Nothing** |



| **return** | **Nothing** |
| --- | --- |
| **remark** |  |
| **demo** |  |

## MfSdkPowerTaskInit

| **Prototype** | | **LIB\_EXPORT s32 MfSdkPowerTaskInit(s32 taskid)** |
| --- | --- | --- |
| **Function** | |  |
| **Params** | **in** | **taskid** |
| **out** | **Nothing** |
| **return** | | **Ref. MfSdkPowerRet\_E** |
| **remark** | |  |
| **demo** | |  |

## MfSdkPowerTaskSuspend

| **Prototype** | | **LIB\_EXPORT s32 MfSdkPowerTaskSuspend(s32 taskid , s32 ms)** |
| --- | --- | --- |
| **Function** | | Sleep is possible within xx ms |
| **Params** | **in** | **taskid** Task ID |
| **ms** ms (unit:ms) |
| **out** | **Nothing** |
| **return** | | **Ref. MfSdkPowerRet\_E** |
| **remark** | |  |
| **demo** | |  |

## MfSdkPowerReset

**Prototype LIB\_EXPORT s32 MfSdkPowerReset(void)**

****

| **Function** | | Restart device. |
| --- | --- | --- |
| **Params** | **in** |  |
| **out** |  |
| **return** | | **Ref. MfSdkPowerRet\_E** |
| **remark** | |  |
| **demo** | |  |

## MfSdkPowerPageCb

| **Prototype** | | **LIB\_EXPORT s32**  **MfSdkPowerPageCb(MfSdkPowerPageCbFunc fun)** |
| --- | --- | --- |
| **Function** | | The device start-up callback. |
| **Params** | **in** | **fun** |
| **out** | **Nothing** |
| **return** | | **Ref. MfSdkPowerRet\_E** |
| **remark** | |  |
| **demo** | |  |

## MfSdkPowerPageInit

| **Prototype** | | **LIB\_EXPORT s32 MfSdkPowerPageInit(void\* state, lv\_align\_t align, lv\_coord\_t x\_ofs, lv\_coord\_t y\_ofs)** |
| --- | --- | --- |
| **Function** | |  |
| **Params** | **in** | **state** |
| **align** |
| **x\_ofs** |
| **y\_ofs** |
| **out** | **Nothing** |
| **return** | | **Ref. MfSdkPowerRet\_E** |
| **remark** | |  |
| **demo** | |  |



## MfSdkPowerSetTime

| **Prototype** | | **LIB\_EXPORT void MfSdkPowerSetTime(s32 time\_num)** |
| --- | --- | --- |
| **Function** | | Set the sleep mode time |
| **Params** | **in** | **time\_num** Sleep mode time(unit: second) |
| **out** | **Nothing** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | |  |

## MfSdkPowerOff

| **Prototype** | | **LIB\_EXPORT void MfSdkPowerOff(void)** |
| --- | --- | --- |
| **Function** | | Power off |
| **Params** | **in** | **Nothing** |
| **out** | **Nothing** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | |  |

## MfSdkPowerKeySetLight

| **Prototype** | | **LIB\_EXPORT void MfSdkPowerKeySetLight()** |
| --- | --- | --- |
| **Function** | |  |
| **Params** | **in** | **Nothing** |
| **out** | **Nothing** |
| **return** | | **Nothing** |
| **remark** | |  |



**demo**

## MfSdkPowerGetBatteryPercentage

| **Prototype** | | **LIB\_EXPORT s32 MfSdkPowerGetBatteryPercentage()** |
| --- | --- | --- |
| **Function** | | Get the battery percentage. |
| **Params** | **in** | **Nothing** |
| **out** | **Nothing** |
| **return** | | **0-100 Battery Percentage** |
| **MFSDK\_POWER\_RET\_F**  **fail**  **AILED** |
| **remark** | |  |
| **demo** | |  |

## MfSdkPowerSetBacklightTime

| **Prototype** | | **LIB\_EXPORT s32 MfSdkPowerSetBacklightTime(s32 seconds)** |
| --- | --- | --- |
| **Function** | | set turn off backlight time |
| **Params** | **in** | **seconds** |
| **out** | **Nothing** |
| **return** | | **MFSDK\_POWER\_RET\_**  **success**  **OK** |
| **Other fail,Ref.MfSdkPowerRet\_E** |
| **remark** | |  |
| **demo** | |  |

## MfSdkPowerSupertimeReset

| **Prototype** | **LIB\_EXPORT s32 MfSdkPowerSupertimeReset(void)** |
| --- | --- |



| **Function** | | reset backlight time |
| --- | --- | --- |
| **Params** | **in** | **Nothing** |
| **out** | **Nothing** |
| **return** | | **MFSDK\_POWER\_RET\_**  **success**  **OK** |
| **Other fail,Ref.MfSdkPowerRet\_E** |
| **remark** | |  |
| **demo** | |  |

## MfSdkPowerSwitchResetTick

| **Prototype** | | **s32 MfSdkPowerSwitchResetTick(void);** |
| --- | --- | --- |
| **Function** | | Reset the sleep timer |
| **Params** | **in** | **Nothing** |
| **out** | **Nothing** |
|  | | **Ref.MfSdkPowerRet\_E** |
| **remark** | |  |
| **demo** | |  |

## MfSdkPowerSleepSwitch

| **Prototype** | | **void MfSdkPowerSleepSwitch(MfSdkPowerSleep\_E switchE);** |
| --- | --- | --- |
| **Function** | | set sleep |
| **Params** | **in** | MFSDK\_POWER\_ACONLINE\_SLEEP\_  EN:enable,  **switchE**  MFSDK\_POWER\_ACONLINE\_SLEEP\_D  N:disable |
| **out** | **Nothing** |
|  | | **Nothing** |
| **remark** | |  |



**demo**

# QR module

## Module description

This module mainly includes APIs about QR & Scanner operations.

## Module structure declaration



## Constant declarations



## MfSdkQrDecode

| **Prototype** | | **LIB\_EXPORT s32 MfSdkQrDecode(s8\* code, s32 size)** |
| --- | --- | --- |
| **Function** | | Get scan data |
| **Params** | **in** | **size** scan size |
| **out** | **code** scan data |
| **return** | | **MFSDK\_QR\_RET\_FAILED not found camera** |
| **MFSDK\_QR\_RET\_DECODE\_F**  **decode failed**  **AILED** |
| **> 0 decode data length** |
| **remark** | |  |
| **demo** | |  |



## MfSdkQrScannerClose

| **Prototype** | | **LIB\_EXPORT s32 MfSdkQrScannerClose(void)** |
| --- | --- | --- |
| **Function** | | Close scanner |
| **Params** | **in** | **Nothing** |
| **out** | **Nothing** |
| **return** | | **MFSDK\_QR\_RET\_OK success** |
| **Other fail** |
| **remark** | |  |
| **demo** | |  |

## MfSdkQrScannerGetImg

| **Prototype** | | **LIB\_EXPORT s8\* MfSdkQrScannerGetImg(void)** |
| --- | --- | --- |
| **Function** | | Get scan image |
| **Params** | **in** | **Nothing** |
| **out** | **Nothing** |
| **return** | | **scan image** |
| **remark** | | **(MF960: 480\*640)** |
| **demo** | |  |

## MfSdkQrScannerOpen

| **Prototype** | | **LIB\_EXPORT s32 MfSdkQrScannerOpen(void)** |
| --- | --- | --- |
| **Function** | | Open scanner |
| **Params** | **in** | **Nothing** |
| **out** | **Nothing** |
| **return** | | **MFSDK\_QR\_RET\_OK success** |



|  | **Other fail** |
| --- | --- |
| **remark** |  |
| **demo** |  |

## MfSdkQrSetScanBoxPosition

| **Prototype** | | **void MfSdkQrSetScanBoxPosition(s32 x, s32 y);** |
| --- | --- | --- |
| **Function** | | Set scaner preview position |
| **Params** | **in** | **x** x offset |
| **y** y offset |
| **out** | **Nothing** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | |  |

## MfSdkQrScannerStart

| **Prototype** | | **LIB\_EXPORT s32 MfSdkQrScannerStart(void)** |
| --- | --- | --- |
| **Function** | | Start scanner |
| **Params** | **in** | **Nothing** |
| **out** | **Nothing** |
| **return** | | **MFSDK\_QR\_RET\_OK success** |
| **Other fail** |
| **remark** | |  |
| **demo** | |  |

## MfSdkQrScannerStop

**Prototype LIB\_EXPORT s32 MfSdkQrScannerStop(void)**

****

| **Function** | | Stop scanner |
| --- | --- | --- |
| **Params** | **in** | **Nothing** |
| **out** | **Nothing** |
| **return** | | **MFSDK\_QR\_RET\_OK success** |
| **Other fail** |
| **remark** | |  |
| **demo** | |  |

## MfSdkQrScannerSetPreview

| **Prototype** | | **LIB\_EXPORT void MfSdkQrScannerSetPreview(int value)** |
| --- | --- | --- |
| **Function** | | Set scaner preview |
| **Params** | **in** | **value** 1:open 0:close |
| **out** | **Nothing** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | |  |

# TMS module

## Module description

This module mainly includes APIs about TMS operations.

## Module structure declaration

None.

## Constant declarations

typedef enum

{

MFSDK\_TMS\_RET\_FAILED = -3, //failed MFSDK\_TMS\_RET\_BOUNDS = -2, //Array out-of-bounds MFSDK\_TMS\_RET\_PARM\_ERROR = -1, //check param



MFSDK\_TMS\_RET\_OK = 0, //success

}MfSdkTmsRet\_E;

typedef enum

{

MFSDK\_TMS\_UPGRADE\_DATA = 0, //0 upgrade data only MFSDK\_TMS\_UPGRADE\_APP, //1 upgrade application

}MfSdkTmsUpdateType\_E;

typedef enum

{

MFSDK\_TMS\_APP\_NOT\_BUSY = 0, MFSDK\_TMS\_APP\_STATE\_BUSY,

}MfSdkTmsAppState\_E;

typedef enum

{

MFSDK\_TMS\_OTA\_LOCAL\_FLASH = 0, //download OTA package

in Flash

MFSDK\_TMS\_OTA\_LOCAL\_RAM, //download OTA package

in RAM

}MfSdkTmsOtaLocal\_E;

typedef enum

{

MFSDK\_TMS\_ACTION\_DOWNLOADING = 0, //tms downloading MFSDK\_TMS\_ACTION\_UPDATEING, //Prepare to upgrade the

downloaded program

MFSDK\_TMS\_ACTION\_CALLBACK, //App add tms download progress callback

}MfSdkTmsAction\_E;

enum {

MFSDK\_TMS\_WAIT\_TASK = 0, // wait task MFSDK\_TMS\_WAIT\_NETWORK\_LINK, // wait network link MFSDK\_TMS\_NETWORK\_ANOMALY, // network anomaly MFSDK\_TMS\_CONNETING, // conneting MFSDK\_TMS\_CONNET\_FAILED, // connect failed MFSDK\_TMS\_REQUEST, // request MFSDK\_TMS\_RECEIVING, // receiving

MFSDK\_TMS\_RECEIVED, // received MFSDK\_TMS\_LOGON\_FAILURE, // logon faliure MFSDK\_TMS\_ILLEGAL\_EQUIPMENT, // illegal equipment



MFSDK\_TMS\_FAILURE\_SEND, // failure send

MFSDK\_TMS\_FAILURE\_RECV, //failure recv

MFSDK\_TMS\_DOWNLOAD, // dowmload

MFSDK\_TMS\_DOWNLOADING, // downloading MFSDK\_TMS\_DOWNLOAD\_LIMIT, // download limit MFSDK\_TMS\_ERROR\_FORMAT, // error format MFSDK\_TMS\_MD5\_ERROR\_TRY, // md5 error try MFSDK\_TMS\_DOWNLOAD\_COMPLETED, // download completed MFSDK\_TMS\_DOWNLOAD\_CANCEL, // download cancel MFSDK\_TMS\_VALIDATION\_FAILURE, // validation failure MFSDK\_TMS\_VERIFICATION\_SIGNATURE, //verification signature MFSDK\_TMS\_NONEED\_TO\_UPDATE, // no need to update MFSDK\_TMS\_NEW\_VERSION, // new version

MFSDK\_TMS\_UPDATING, // updating

MFSDK\_TMS\_WIFI\_UPGRADE, // wifi upgrade MFSDK\_TMS\_UPDATE\_SUCCESS, // update success MFSDK\_TMS\_UPDATE\_FAILED, // update failed MFSDK\_TMS\_REBOOTING, // rebooting

}MfSdkTmsResult\_E;

/\*\*

* @brief
* @param[in] int current value
* @param[in] int total value

\*/

typedef void (\*MfSdkTmsProgressCb)(int, int);

/\*\*

* @brief
* @param[in] int result type

\*/

typedef void (\*MfSdkTmsResultCb)(int);

/\*\*

* @brief
* @param[in] int result type
* @return 1 : continue

\*/

typedef s32 (\*MfSdkTmsActionCb)(MfSdkTmsAction\_E type);

## MfSdkTmsSetProgressCallback

| **Prototype** | **LIB\_EXPORT void MfSdkTmsSetProgressCallback(MfSdkTmsProgressCb**  **pFunCallback)** |
| --- | --- |



| **Function** | | Set TMS progress callback function |
| --- | --- | --- |
| **Params** | **in** | **Nothing** |
| **out** | **Nothing** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | |  |

## MfSdkTmsSetResultCallback

| **Prototype** | | **void MfSdkTmsSetResultCallback(MfSdkTmsResultCb pFunCallback);** |
| --- | --- | --- |
| **Function** | | tms result callback |
| **Params** | **in** | **pFunCallback** Callback |
| **out** | **Nothing** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | |  |

## MfSdkTmsHeartBeat

| **Prototype** | | **LIB\_EXPORT s32 MfSdkTmsHeartBeat()** |
| --- | --- | --- |
| **Function** | | TMS heartBeat |
| **Params** | **in** | **Nothing** |
| **out** | **Nothing** |
| **return** | | **MFSDK\_TMS\_RET\_OK success** |
| **Other fail** |
| **remark** | |  |
| **demo** | |  |



## MfSdkTmsUpdate

| **Prototype** | | **LIB\_EXPORT void MfSdkTmsUpdate(void)** |
| --- | --- | --- |
| **Function** | | TMS Update |
| **Params** | **in** | **Nothing** |
| **out** | **Nothing** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | |  |

## MfSdkTmsUpdateFile

| **Prototype** | | **LIB\_EXPORT s32 MfSdkTmsUpdateFile(const s8\* file, MfSdkTmsUpdateType\_E flag)** |
| --- | --- | --- |
| **Function** | |  |
| **Params** | **in** | **file** |
| **flag** |
| **out** | **Nothing** |
| **return** | |  |
| **remark** | |  |
| **demo** | |  |

## MfSdkTmsAppBusy

| **Prototype** | | **LIB\_EXPORT void MfSdkTmsAppBusy(MfSdkTmsAppState\_E nBusyState)** |
| --- | --- | --- |
| **Function** | | Set TMS state |
| **Params** | **in** | **nBusyState** |
| **out** | **Nothing** |
| **return** | | **Nothing** |



| **remark** |  |
| --- | --- |
| **demo** |  |

## MfSdkTmsCheckTimeDisable

| **Prototype** | | **LIB\_EXPORT void MfSdkTmsCheckTimeDisable()** |
| --- | --- | --- |
| **Function** | | Disable check time from tms server.Only for traditional POS. |
| **Params** | **in** | **nBusyState** |
| **out** | **Nothing** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | |  |

## MfSdkTmsGetMsg

| **Prototype** | | **LIB\_EXPORT s8 \*MfSdkTmsGetMsg()** |
| --- | --- | --- |
| **Function** | | Gets the current tms message |
| **Params** | **in** | **Nothing** |
| **out** | **Nothing** |
| **return** | | **tms msg string** |
| **remark** | |  |
| **demo** | |  |

## MfSdkTmsGetResult

| **Prototype** | | **LIB\_EXPORT s8 \*MfSdkTmsGetResult()** |
| --- | --- | --- |
| **Function** | | Get tms result information |
| **Params** | **in** | **Nothing** |



|  | **out** | **Nothing** |
| --- | --- | --- |
| **return** | | **tms result string** |
| **remark** | |  |
| **demo** | |  |

## MfSdkTmsSetSig

| **Prototype** | | **LIB\_EXPORT void MfSdkTmsSetSig(s32 flag);** |
| --- | --- | --- |
| **Function** | | Whether the tms upgrade verifies the signature.(For traditional POS,Soundbox not applicable) |
| **Params** | **in** | **s32 flag** 0-enable ,1-disable |
| **out** | **Nothing** |
| **return** | | **none** |
| **remark** | |  |
| **demo** | | **MfSdkTmsSetSig(0)/MfSdkTmsSetSig(1);** |

## MfSdkTmsUpdateOta

| **Prototype** | | **s32 MfSdkTmsUpdateOta(char \*filename,char \*flag);** |
| --- | --- | --- |
| **Function** | | update OTA |
| **Params** | **in** | download file path and name eg.  **filename**  "exdata\\down.tmp" |
| **flag** fix string "ota.bin" |
| **out** | **Nothing** |
| **return** | | **MFSDK\_TMS\_RET\_OK success** |
| **other fail** |
| **remark** | |  |
| **demo** | |  |



## MfSdkTmsSetConnectRetryCnt

| **Prototype** | | **void MfSdkTmsSetConnectRetryCnt(u32 count);** |
| --- | --- | --- |
| **Function** | | Set TMS connect retry count |
| **Params** | **in** | **none** |
| **out** | **none** |
| **return** | | **none** |
| **remark** | |  |
| **demo** | |  |

## MfSdkTmsSetErrorRetryCnt

| **Prototype** | | **void MfSdkTmsSetErrorRetryCnt(u32 count);** |
| --- | --- | --- |
| **Function** | | Set TMS error retry count |
| **Params** | **in** | **none** |
| **out** | **none** |
| **return** | | **none** |
| **remark** | |  |
| **demo** | |  |

## MfSdkTmsEnable

| **Prototype** | | **void MfSdkTmsEnable(MFSDKBOOL b);** |
| --- | --- | --- |
| **Function** | | Whether to enable the TMS service. Default support for TMS. |
| **Params** | **in** | MFSDK\_TRUE-enable ,  **b**  MFSDK\_FALSE-disable |
| **out** | **none** |
| **return** | | **none** |



| **remark** |  |
| --- | --- |
| **demo** | //if disable TMS  MfSdkTmsEnable(MFSDK\_FALSE); |

## MfSdkTmsSetActionCallback

| **Prototype** | | **s32 MfSdkTmsSetActionCallback(MfSdkTmsActionCb pFunCallback);** |
| --- | --- | --- |
| **Function** | | Set TMS action progress callback function |
| **Params** | **in** | **pFunCallback** Callback |
| **out** | **none** |
| **return** | | **For details, see**  **MfSdkTmsResult\_E** |
| **remark** | |  |
| **demo** | |  |

## MfSdkTmsEnableSyncTime

| **Prototype** | | **void MfSdkTmsEnableSyncTime(MFSDKBOOL b);** |
| --- | --- | --- |
| **Function** | | Whether to synchronize TMS time (enabled by default).Only for Soundbox. |
| **Params** | **in** | MFSDK\_TRUE - enable  **b**  MFSDK\_FALSE - disable |
| **out** | **none** |
| **return** | |  |
| **remark** | |  |
| **demo** | |  |



# LVGL module

## Module description

This module mainly includes APIs about LVGL functions.

## Module structure declaration

None.

## Constant declarations

typedef enum

{

MFSDK\_LVGL\_OPER\_SUCC = 0,

MFSDK\_LVGL\_OPER\_FAIL = -1,

MFSDK\_LVGL\_OPER\_CANCEL = -2,

MFSDK\_LVGL\_OPER\_TIMEOUT = -3,

MFSDK\_LVGL\_OPER\_BACKSPACE = -4,

} MfSdkLvglOper\_E;

typedef enum

{

MFSDK\_LVGL\_LARGE\_FONT = 2,

MFSDK\_LVGL\_NORMAL\_FONT = 1,

MFSDK\_LVGL\_SMALL\_FONT = 0,

} MfSdkLvglFont\_E;

typedef enum

{

MFSDK\_LVGL\_ALIGN\_RIGHT = 2,

MFSDK\_LVGL\_ALIGN\_CENTER = 1,

MFSDK\_LVGL\_ALIGN\_LEFT = 0,

} MfSdkLvglAlign\_E;

typedef enum

{

MFSDK\_LVGL\_INPUT\_MODE\_NUM = 0,

MFSDK\_LVGL\_INPUT\_MODE\_TEXT = 1,

MFSDK\_LVGL\_INPUT\_MODE\_IP = 2,

MFSDK\_LVGL\_INPUT\_MODE\_PWD = 3,

MFSDK\_LVGL\_INPUT\_MODE\_AMOUNT = 4,

} MfSdkLvglInput\_E;

typedef enum

{

MFSDK\_LVGL\_INPUTTEXT\_IMAGE = 0,

}MfSdkLvglImageType\_E;

typedef enum {

MFSDK\_LVGL\_SB\_320\_240\_BLACK = 0, //320\*240 MFSDK\_LVGL\_SB\_320\_240\_WHITE = 1, //320\*240 MFSDK\_LVGL\_SB\_480\_800\_BLACK = 4, //480\*800 MFSDK\_LVGL\_SB\_480\_800\_WHITE = 5, //480\*800

}MfSdkLvglStatusBar\_E;

typedef enum

{



OVER,

OVER,

MFSDK\_LVGL\_NETMENU\_EXIT\_TIMEOUT = -1,

MFSDK\_LVGL\_NETMENU\_EXIT\_CANCEL = 0, MFSDK\_LVGL\_NETMENU\_NETINFO\_EXIT, MFSDK\_LVGL\_NETMENU\_NETSET1\_EXIT\_TIMEOVER, MFSDK\_LVGL\_NETMENU\_NETSET1\_EXIT, MFSDK\_LVGL\_NETMENU\_NET4G\_ALREADY\_EXIT, MFSDK\_LVGL\_NETMENU\_NET4G\_FAILED\_NO\_SIM, MFSDK\_LVGL\_NETMENU\_NET4G\_FAILED\_EXIT\_TIMEOVER, MFSDK\_LVGL\_NETMENU\_NET4G\_SUCCEED\_EXIT, MFSDK\_LVGL\_NETMENU\_NETWIFI\_FAILED\_EXIT\_TIMEOVER, MFSDK\_LVGL\_NETMENU\_NETWIFI\_SUCCEED\_EXIT, MFSDK\_LVGL\_NETMENU\_NETADDWIFI\_EXIT\_TIMEOVER, MFSDK\_LVGL\_NETMENU\_NETADDWIFI\_EXIT, MFSDK\_LVGL\_NETMENU\_NETWIFISAVE\_EXIT\_TIMEOVER, MFSDK\_LVGL\_NETMENU\_NETWIFISAVE\_EXIT, MFSDK\_LVGL\_NETMENU\_NETWIFI\_SWITCH\_FAILED\_EXIT\_TIME

MFSDK\_LVGL\_NETMENU\_NETWIFI\_SWITCH\_FAILED\_EXIT, MFSDK\_LVGL\_NETMENU\_NETWIFISCANING\_EXIT\_TIMEOVER, MFSDK\_LVGL\_NETMENU\_NETWIFISCANING\_EXIT, MFSDK\_LVGL\_NETMENU\_NETWIFILIST\_EXIT\_TIMEOVER, MFSDK\_LVGL\_NETMENU\_NETWIFILIST\_EXIT, MFSDK\_LVGL\_NETMENU\_NETWIFIINPUTPASSWORD\_EXIT\_TIME



MFSDK\_LVGL\_NETMENU\_NETWIFIINPUTPASSWORD\_EXIT, MFSDK\_LVGL\_NETMENU\_NETWIFICONNECT\_SUCCEED\_EXIT, MFSDK\_LVGL\_NETMENU\_NETWIFICONNECT\_FAILED\_EXIT,

}MfSdkLvglNetMenuExitValue\_E;

## MfSdkLvglInit

| **Prototype** | | **LIB\_EXPORT s32 MfSdkLvglInit(void)** |
| --- | --- | --- |
| **Function** | | lvgl module initialization |
| **Params** | **in** | **Nothing** |
| **out** | **Nothing** |
| **return** | | **0 success** |
| **Other fail** |
| **remark** | |  |
| **demo** | |  |

## MfSdkLvglGetPageBody

| **Prototype** | | **LIB\_EXPORT lv\_obj\_t\* MfSdkLvglGetPageBody()** |
| --- | --- | --- |
| **Function** | | Get the modal lvgl page parent object |
| **Params** | **in** | **Nothing** |
| **out** | **Nothing** |
| **return** | | **0 fail** |
| **Other success,lvgl page parent object** |
| **remark** | |  |
| **demo** | |  |

## MfSdkLvglExit

| **Prototype** | **LIB\_EXPORT s32 MfSdkLvglExit(void)** |
| --- | --- |



| **Function** | | lvgl module exit |
| --- | --- | --- |
| **Params** | **in** | **Nothing** |
| **out** | **Nothing** |
| **return** | | **0 success** |
| **Other fail** |
| **remark** | |  |
| **demo** | |  |

## MfSdkLvglCls

| **Prototype** | | **LIB\_EXPORT void MfSdkLvglCls(void)** |
| --- | --- | --- |
| **Function** | | clear screen |
| **Params** | **in** | **Nothing** |
| **out** | **Nothing** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | |  |

## MfSdkLvglClsEnableStatusBar

| **Prototype** | | **LIB\_EXPORT void MfSdkLvglClsEnableStatusBar(s32 status)** |
| --- | --- | --- |
| **Function** | | enable status bar |
| **Params** | **in** | **status** 0 - off, 1 - on |
| **out** | **Nothing** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | |  |



## MfSdkLvglEnableDefaultStatusBarIcons

| **Prototype** | | **LIB\_EXPORT void MfSdkLvglEnableDefaultStatusBarIcons(s32 status)** |
| --- | --- | --- |
| **Function** | | enable default status bar icons |
| **Params** | **in** | **status** 0 - off, 1 - on |
| **out** | **Nothing** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | |  |

## MfSdkLvglClearLine

| **Prototype** | | **LIB\_EXPORT void MfSdkLvglClearLine(s32 lineNo)** |
| --- | --- | --- |
| **Function** | | clear specified line |
| **Params** | **in** | **lineNo** line number(starting from 0) |
| **out** | **Nothing** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | |  |

## MfSdkLvglGetStatusBarHeight

| **Prototype** | | **LIB\_EXPORT s32 MfSdkLvglGetStatusBarHeight()** |
| --- | --- | --- |
| **Function** | | get status bar height |
| **Params** | **in** | **Nothing** |
| **out** | **Nothing** |
| **return** | | **height** |
| **remark** | |  |



**demo**

## MfSdkLvglCreateButtons

| **Prototype** | | **LIB\_EXPORT s32**  **MfSdkLvglCreateButtons(MfSdkLvglButtons\_T items[], s32 size)** |
| --- | --- | --- |
| **Function** | | create buttons |
| **Params** | **in** | create buttons parameter.If items->id < 0 is  **items**  not clickable |
| **size** items size |
| **out** | **Nothing** |
| **return** | | **height** |
| **remark** | |  |
| **demo** | |  |

## MfSdkLvglClearAll

| **Prototype** | | **LIB\_EXPORT void MfSdkLvglClearAll(void)** |
| --- | --- | --- |
| **Function** | | clear screen without line 0 |
| **Params** | **in** | **Nothing** |
| **out** | **Nothing** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | |  |

## MfSdkLvglWaitKey

| **Prototype** | **LIB\_EXPORT s32 MfSdkLvglWaitKey(s32 timeout)** |
| --- | --- |
| **Function** | waiting for key pressed |



| **Params** | **in** | **timeout** timeout(second) |
| --- | --- | --- |
| **out** | **Nothing** |
| **return** | | **>=0 success, key value or button id** |
| **Other fail** |
| **remark** | |  |
| **demo** | |  |

## MfSdkLvglWaitKeyMs

| **Prototype** | | **LIB\_EXPORT s32 MfSdkLvglWaitKeyMs(s32 timeoutMs)** |
| --- | --- | --- |
| **Function** | | waiting for key pressed |
| **Params** | **in** | **timeout** timeout(ms) |
| **out** | **Nothing** |
| **return** | | **>=0 success, key value or button id** |
| **Other fail** |
| **remark** | |  |
| **demo** | |  |

## MfSdkLvglClrKeyFlag

| **Prototype** | | **LIB\_EXPORT void MfSdkLvglClrKeyFlag()** |
| --- | --- | --- |
| **Function** | | Clean the key flag bit before getting the key loop |
| **Params** | **in** | **Nothing** |
| **out** | **Nothing** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | |  |



## MfSdkLvglCheckKey

| **Prototype** | | **LIB\_EXPORT MFSDKBOOL MfSdkLvglCheckKey(void)** |
| --- | --- | --- |
| **Function** | | check if key pressed |
| **Params** | **in** | **Nothing** |
| **out** | **Nothing** |
| **return** | | **1 key pressed** |
| **0 no key pressed** |
| **remark** | |  |
| **demo** | |  |

## MfSdkLvglGetKey

| **Prototype** | | **LIB\_EXPORT s32 MfSdkLvglGetKey(void)** |
| --- | --- | --- |
| **Function** | | get key value |
| **Params** | **in** | **Nothing** |
| **out** | **Nothing** |
| **return** | | **>=0 Ref. MfSdkKbKeyCode\_E** |
| **Other fail** |
| **remark** | |  |
| **demo** | |  |

## MfSdkLvglDispTextCoord

| **Prototype** | | **LIB\_EXPORT s32 MfSdkLvglDispTextCoord(s32 x, s32 y, s32 font, s8\* text)** |
| --- | --- | --- |
| **Function** | | disp text by coordinate |
| **Params** | **in** | **x** horizontal coordinate |
| **y** vertical coordinate |



|  |  | **font** 0 - small,1 - normal,2 - big |
| --- | --- | --- |
| **text** text |
| **out** | **Nothing** |
| **return** | | **0 success** |
| **Other fail** |
| **remark** | |  |
| **demo** | |  |

## MfSdkLvglDispTextLine

| **Prototype** | | **LIB\_EXPORT s32 MfSdkLvglDispTextLine(s32 lineNo, s32 font, const s8\* text, s32 align, u8 reverse)** |
| --- | --- | --- |
| **Function** | | disp text by line number |
| **Params** | **in** | **lineNo** line number(starting from 0) |
| **font** font(0 - small,1 - normal,2 - big) |
| **text** text |
| **align** align type(0 - left, 1 - center, 2 - right) |
| **reverse** reverse mode(1 - yes, 0 - no) |
| **out** | **Nothing** |
| **return** | | **0 success** |
| **Other fail** |
| **remark** | |  |
| **demo** | |  |

## MfSdkLvglDispMenuText

| **Prototype** | | **LIB\_EXPORT s32 MfSdkLvglDispMenuText(s32 font, s8\* title, MfSdkLvglMenuText\_T menus[], s32 size, s32 select, s32 timeout)** |
| --- | --- | --- |
| **Function** | | disp text menus |
| **Params** | **in** | **font** font(0 - small,1 - normal,2 - big) |
| **title** title |
| **menus** menu list |
| **size** list size |
| **select** default menu id |



|  |  | **timeout** timeout(second) |
| --- | --- | --- |
| **out** | **Nothing** |
| **return** | | **>=0 success, menu id** |
| **Other fail** |
| **remark** | |  |
| **demo** | |  |

## MfSdkLvglDispMenuIcon

| **Prototype** | | **LIB\_EXPORT s32 MfSdkLvglDispMenuIcon(s32 font, s8\* title, MfSdkLvglMenuIcon\_T menus[], s32 size, s32 timeout)** |
| --- | --- | --- |
| **Function** | | disp image menus |
| **Params** | **in** | **font** 0 - small,1 - normal,2 - big |
| **title** title |
| **menus** menu list |
| **size** list size |
| **timeout** timeout(second) |
| **out** | **Nothing** |
| **return** | | **>=0 success, menu id** |
| **Other fail** |
| **remark** | |  |
| **demo** | |  |

## MfSdkLvglDispButton

| **Prototype** | | **LIB\_EXPORT s32 MfSdkLvglDispButton(s32 lineNo, s32 font, MfSdkButtonIcon\_T buttons[], s32 size, s32 timeout)** |
| --- | --- | --- |
| **Function** | | disp buttons |
| **Params** | **in** | **lineNo** line number(starting from 0) |
| **font** font(0 - small,1 - normal,2 - big) |
| **buttons** button list |
| **size** list size |
| **timeout** timeout(second) |
| **out** | **Nothing** |



| **return** | **>=0 success, button id** |
| --- | --- |
| **Other fail** |
| **remark** |  |
| **demo** |  |

## MfSdkLvglMessageBox

| **Prototype** | | **LIB\_EXPORT s32 MfSdkLvglMessageBox(s32 width, s32 height, s8\* msg, s8\* btnLeft, s8\* btnRight, s32 timeout)** |
| --- | --- | --- |
| **Function** | | disp text menus |
| **Params** | **in** | **width** width |
| **height** height |
| **msg** message |
| **btnLeft** left button text |
| **btnRight** right button text |
| **timeout** timeout(second) |
| **out** | **Nothing** |
| **return** | | **>=0 success, button id (0 - left, 1 - right)** |
| **Other fail** |
| **remark** | |  |
| **demo** | |  |

## MfSdkLvglDispList

| **Prototype** | | **LIB\_EXPORT s32 MfSdkLvglDispList(s32 font, s8\* title, s8\* items[], s32 size, s32 timeout)** |
| --- | --- | --- |
| **Function** | | disp text list |
| **Params** | **in** | **font** font(0 - small,1 - normal,2 - big) |
| **title** title |
| **items** text list |
| **size** list size |
| **timeout** timeout(second) |
| **out** | **Nothing** |
| **return** | | **0 success** |



|  | **Other fail** |
| --- | --- |
| **remark** |  |
| **demo** |  |

## MfSdkLvglDispRollpage

| **Prototype** | | **LIB\_EXPORT s32 MfSdkLvglDispRollpage(s32 font, s8\* title, s8\* buf, s32 timeout)** |
| --- | --- | --- |
| **Function** | | disp text rollpage |
| **Params** | **in** | **font** font(0 - small,1 - normal,2 - big) |
| **title** title |
| **buf** text buffer |
| **timeout** timeout(second) |
| **out** | **Nothing** |
| **return** | | **0 success** |
| **Other fail** |
| **remark** | |  |
| **demo** | |  |

## MfSdkLvglInputText

| **Prototype** | | **LIB\_EXPORT s32 MfSdkLvglInputText(s32 font, s8\* title, s8\* msg, s8\* text, s32 mode, s32 min, s32 max, s32 timeout)** |
| --- | --- | --- |
| **Function** | | input text, show input text on a line |
| **Params** | **in** | **font** font(0 - small,1 - normal,2 - big) |
| **title** title |
| **msg** message |
| 0 - number, 1 - text,  **mode**  2 - IP, 3 - password, 4 - amount |
| **min** mininum length |
| **max** maxinum length |
| **timeout** timeout(second) |
| **out** | **text** input text |
| **return** | | **0 success** |



|  | **Other fail** |
| --- | --- |
| **remark** |  |
| **demo** |  |

## MfSdkLvglInputTextEx

| **Prototype** | | **LIB\_EXPORT s32 MfSdkLvglInputTextEx(s32 font, s8\* title, s8\* msg, s8\* text, s32 mode, s32 min, s32 max, s32 timeout)** |
| --- | --- | --- |
| **Function** | | input text, show input text in a input box |
| **Params** | **in** | **font** font(0 - small,1 - normal,2 - big) |
| **title** title |
| **msg** message |
| 0 - number, 1 - text,  **mode**  2 - IP, 3 - password, 4 - amount |
| **min** mininum length |
| **max** maxinum length |
| **timeout** timeout(second) |
| **out** | **text** input text |
| **return** | | **0 success** |
| **Other fail** |
| **remark** | |  |
| **demo** | |  |

## MfSdkLvglDispBar

| **Prototype** | | **LIB\_EXPORT s32 MfSdkLvglDispBar(s32 width, s32 height, s32 min, s32 max, void\*\* bar)** |
| --- | --- | --- |
| **Function** | | disp progress bar |
| **Params** | **in** | **width** width |
| **height** height |
| **min** mininum value |
| **max** maxinum value |
| **out** | **bar** progress bar handle |
| **return** | | **0 success** |



|  | **Other fail** |
| --- | --- |
| **remark** |  |
| **demo** |  |

## MfSdkLvglUpdateBar

| **Prototype** | | **LIB\_EXPORT s32 MfSdkLvglUpdateBar(void\* bar, s32 value)** |
| --- | --- | --- |
| **Function** | | update progress bar |
| **Params** | **in** | **bar** progress bar handle |
| **value** current value |
| **out** | **Nothing** |
| **return** | | **0 success** |
| **Other fail** |
| **remark** | |  |
| **demo** | |  |

## MfSdkLvglDispLed

| **Prototype** | | **LIB\_EXPORT s32 MfSdkLvglDispLed(s32 x, s32 y, s32 width, s32 height)** |
| --- | --- | --- |
| **Function** | | disp progress bar |
| **Params** | **in** | **x** horizontal coordinate |
| **y** vertical coordinate |
| **width** width |
| **height** height |
| **out** | **Nothing** |
| **return** | | **0 success** |
| **Other fail** |
| **remark** | |  |
| **demo** | |  |



## MfSdkLvglUpdateLed

| **Prototype** | | **LIB\_EXPORT s32 MfSdkLvglUpdateLed(s32 index, s32 state)** |
| --- | --- | --- |
| **Function** | | update led state |
| **Params** | **in** | led index (0 - red, 1 - blue,  **index**  2 - green, 3 - yellow) |
| **state** ( 0 - off, 1 - on) |
| **out** | **Nothing** |
| **return** | | **0 success** |
| **Other fail** |
| **remark** | |  |
| **demo** | |  |

## MfSdkLvglShowQrcode

| **Prototype** | | **LIB\_EXPORT s32 MfSdkLvglShowQrcode(s32 font, s8\* title, s8\* qrcode, s32 width, s32 align)** |
| --- | --- | --- |
| **Function** | | show qrcode |
| **Params** | **in** | **font** font(0 - small,1 - normal,2 - big) |
| **title** title |
| **qrcode** qrcode text |
| **width** width |
| **align** align type(0 - left, 1 - center, 2 - right) |
| **out** | **Nothing** |
| **return** | | **0 success** |
| **Other fail** |
| **remark** | |  |
| **demo** | |  |



## MfSdkLvglShowQrcodeEx

| **Prototype** | | **LIB\_EXPORT s32 MfSdkLvglShowQrcodeEx(s8\* qrcode, s32 qrcodeLen, s32 width, s32 align, s32 y)** |
| --- | --- | --- |
| **Function** | | show qrcode |
| **Params** | **in** | **qrcode** qrcode data |
| **qrcodeLen** qrcode data length |
| **width** width |
| **align** align type(0 - left, 1 - center, 2 - right) |
| **y** Y offset |
| **out** | **Nothing** |
| **return** | | **0 success** |
| **Other fail** |
| **remark** | |  |
| **demo** | |  |

## MfSdkLvglShowImage

| **Prototype** | | **LIB\_EXPORT s32 MfSdkLvglShowImage(s32 x, s32 y, s8\* fileName, s8\* text, s32 id)** |
| --- | --- | --- |
| **Function** | | show image button |
| **Params** | **in** | **x** horizontal coordinate |
| **y** vertical coordinate |
| **fileName** image file name |
| **text** button text |
| button id( >=0 image with click event, <0  **id**  no click events) |
| **out** | **Nothing** |
| **return** | | **0 success** |
| **Other fail** |
| **remark** | |  |
| **demo** | |  |



## MfSdkLvglShowImageEx

| **Prototype** | | **LIB\_EXPORT s32 MfSdkLvglShowImage(s32 x, s32 y, s32 align, s8\* fileName, s8\* text, s32 id)** |
| --- | --- | --- |
| **Function** | | show image button |
| **Params** | **in** | **x** horizontal coordinate |
| **y** vertical coordinate |
| **align** align type(0 - left, 1 - center, 2 - right) |
| **fileName** image file name |
| **text** button text |
| button id( >=0 image with click event, <0  **id**  no click events) |
| **out** | **Nothing** |
| **return** | | **0 success** |
| **Other fail** |
| **remark** | |  |
| **demo** | |  |

## MfSdkLvglShowImagebuff

| **Prototype** | | **LIB\_EXPORT s32 MfSdkLvglShowImagebuff(s32 x, s32 y, s32 w, s32 h, s8\* buf, s8\* text, s32 id)** |
| --- | --- | --- |
| **Function** | | show image buff |
| **Params** | **in** | **x** horizontal coordinate |
| **y** vertical coordinate |
| **w** width |
| **h** height |
| **buf** image buffer |
| **text** button text |
| button id( >=0 image with click event, <0  **id**  no click events) |
| **out** | **Nothing** |
| **return** | | **0 success** |
| **Other fail** |



| **remark** |  |
| --- | --- |
| **demo** |  |

## MfSdkLvglShowScreenCanves

| **Prototype** | | **LIB\_EXPORT s32 MfSdkLvglShowScreenCanves(s8\* buf)** |
| --- | --- | --- |
| **Function** | | show except for the status bar the screen buff |
| **Params** | **in** | **buf** Screen buff(rgba 32-bit) |
| **out** | **Nothing** |
| **return** | | **0 success** |
| **Other fail** |
| **remark** | | **MF960 with status bar,screen size is 480 \* (800-30),buff size is 480**  **\* (800-30) \* 4;MF960 without status bar,screen size is 480 \* 800,buff size is 480 \* 800 \* 4** |
| **demo** | |  |

## MfSdkLvglDrawLine

| **Prototype** | | **LIB\_EXPORT s32 MfSdkLvglDrawLine(s32 x, s32 y, s32 w, s32**  **h)** |
| --- | --- | --- |
| **Function** | | draw a line |
| **Params** | **in** | **x** horizontal coordinate |
| **y** vertical coordinate |
| **w** width |
| **h** height |
| **out** | **Nothing** |
| **return** | | **0 success** |
| **Other fail** |
| **remark** | |  |
| **demo** | |  |



## MfSdkLvglDrawBox

| **Prototype** | | **LIB\_EXPORT s32 MfSdkLvglDrawBox(s32 x, s32 y, s32 w, s32**  **h)** |
| --- | --- | --- |
| **Function** | | draw a box |
| **Params** | **in** | **x** horizontal coordinate |
| **y** vertical coordinate |
| **w** width |
| **h** height |
| **out** | **Nothing** |
| **return** | | **0 success** |
| **Other fail** |
| **remark** | |  |
| **demo** | |  |

## MfSdkLvglEsign

| **Prototype** | | **LIB\_EXPORT s32 MfSdkLvglEsign(s32 font, s8\* title, s8\* factor, u8 \*\* imgData, s32\* dataSize, s32 timeout)** |
| --- | --- | --- |
| **Function** | | electronic signature |
| **Params** | **in** | **font** font(0 - small,1 - normal,2 - big) |
| **title** title |
| **factor** factor |
| **timeout** timeout(second) |
| **out** | **imgData** image data |
| **dataSize** image data size |
| **return** | | **0 success** |
| **Other fail** |
| **remark** | |  |
| **demo** | |  |



## MfSdkLvglScan

| **Prototype** | | **LIB\_EXPORT s32 MfSdkLvglScan(s32 font, s8\* title, s8\* data, s32 datasize, s32 timeout)** |
| --- | --- | --- |
| **Function** | | Scan the QR code, barcode |
| **Params** | **in** | **font** font(0 - small,1 - normal,2 - big) |
| **title** title |
| **dataSize** code data size |
| **timeout** timeout(second) |
| **out** | **data** code data |
| **return** | | **0 success** |
| **Other fail** |
| **remark** | |  |
| **demo** | |  |

## MfSdkLvglDispDialog

| **Prototype** | | **LIB\_EXPORT s32 MfSdkLvglDispDialog(s32 font, s8\* title, s8\* msg, s32 timeout, u8 isCountDown, void\*\* handle)** |
| --- | --- | --- |
| **Function** | | display dialog |
| **Params** | **in** | **font** font(0 - small,1 - normal,2 - big) |
| **title** title |
| **msg** message |
| **timeout** timeout(second) |
| **isCountDown** countdown mode (1 - Yes, 0 - No) |
| **out** | **handle** dialog handle |
| **return** | | **0 success** |
| **Other fail** |
| **remark** | |  |
| **demo** | |  |



## MfSdkLvglUpdateDialog

| **Prototype** | | **LIB\_EXPORT s32 MfSdkLvglUpdateDialog(void\* handle, s8\* title, s8\* msg, s32 timeout, u8 isCountDown)** |
| --- | --- | --- |
| **Function** | | update dialog |
| **Params** | **in** | **handle** dialog handle |
| **title** title |
| **msg** message |
| **timeout** timeout(second) |
| **isCountDown** countdown mode (1 - Yes, 0 - No) |
| **out** | **Nothing** |
| **return** | | **0 success** |
| **Other fail** |
| **remark** | |  |
| **demo** | |  |

## MfSdkLvglCloseDialog

| **Prototype** | | **LIB\_EXPORT s32 MfSdkLvglCloseDialog(void\* handle)** |
| --- | --- | --- |
| **Function** | | close dialog |
| **Params** | **in** | **handle** dialog handle |
| **out** | **Nothing** |
| **return** | | **0 success** |
| **Other fail** |
| **remark** | |  |
| **demo** | |  |

## MfSdkLvglDispRollpageEx

**LIB\_EXPORT s32 MfSdkLvglDispRollpageEx(s32 font, s8\* title, s8\* buf, s32 timeout)**

| **Function** | | display text rollpage |
| --- | --- | --- |
| **Params** | **in** | **font** font(0 - small,1 - normal,2 - big) |
| **title** title |
| **buf** text buffer |
| **timeout** timeout(second) |
| **out** | **Nothing** |
| **return** | | **0 success** |
| **Other fail** |
| **remark** | |  |
| **demo** | |  |

## MfSdkLvglDispMultipleChoice

| **Prototype** | | **LIB\_EXPORT s32 MfSdkLvglDispMultipleChoice(s32 font,const**  **s8\* title, MfSdkLvglMenuText\_T\* choiseArry, s32 choiseSize, s32 select, s32 timeout)** |
| --- | --- | --- |
| **Function** | | display multiple choice |
| **Params** | **in** | **font** font(0 - small,1 - normal,2 - big) |
| **title** title |
| **choiseArry** list |
| **choiseSize** list size |
| **select** default list id |
| **timeout** timeout(second) |
| **out** | **Nothing** |
| **return** | | **>=0 success, list id** |
| **Other fail** |
| **remark** | |  |
| **demo** | |  |

## MfSdkLvglDispSetSlider



### Prototype

**LIB\_EXPORT s32 MfSdkLvglDispSetSlider(s8\* title, s32 width, s32 height, s32 minLevel, s32 maxLevel, s32 startValue, s32 timeout)**

****

| **Function** | | set slider |
| --- | --- | --- |
| **Params** | **in** | **title** title |
| **width** slider width |
| **height** slider height |
| **minLevel** minimum slider value |
| **maxLevel** maximum slider value |
| **startValue** startValue slider initial value |
| **timeout** timeout(second) |
| **out** | **Nothing** |
| **return** | | **0 success** |
| **Other fail** |
| **remark** | |  |
| **demo** | |  |

## MfSdkLvglDispSetResult

| **Prototype** | | **LIB\_EXPORT s32 MfSdkLvglDispSetResult(s32 result, s8\* msg, s32 timeout)** |
| --- | --- | --- |
| **Function** | | display set result |
| **Params** | **in** | **result** result |
| **msg** message |
| **timeout** timeout(second) |
| **out** | **Nothing** |
| **return** | | **0 success** |
| **Other fail** |
| **remark** | |  |
| **demo** | |  |

## MfSdkLvglInputPin

**LIB\_EXPORT s32 MfSdkLvglInputPin(s8\* title, s8\* msg, s8\* amount, s8\* cardID, s8\* pin, s32 timeout)**

****

| **Function** | | update dialog |
| --- | --- | --- |
| **Params** | **in** | **title** title |
| **msg** message |
| **amount** amount |
| **cardID** cardID |
| **timeout** timeout(second) |
| **out** | **pin** input pin |
| **return** | | **0 success** |
| **Other fail** |
| **remark** | |  |
| **demo** | |  |

## MfSdkLvglDispAnimation

| **Prototype** | | **LIB\_EXPORT s32 MfSdkLvglDispAnimation(s32 imgSize, s8\*\* imgName, s8\* msg, s32 interval, void\*\* handle)** |
| --- | --- | --- |
| **Function** | | display animation |
| **Params** | **in** | **imgSize** images size |
| **imgName** images name |
| **msg** message |
| **interval** image display interval |
| **out** | **handle** animation handle |
| **return** | | **0 success** |
| **Other fail** |
| **remark** | |  |
| **demo** | |  |

## MfSdkLvglCloseAnimation

| **Prototype** | **LIB\_EXPORT s32 MfSdkLvglCloseAnimation(void\* handle)** |
| --- | --- |
| **Function** | close animation |



| **Params** | **in** | **handle** animation handle |
| --- | --- | --- |
| **out** | **Nothing** |
| **return** | | **0 success** |
| **Other fail** |
| **remark** | |  |
| **demo** | |  |

## MfSdkLvglCreateStatusBarTask

| **Prototype** | | **LIB\_EXPORT void MfSdkLvglCreateStatusBarTask(MfSdkStatusBarCb task\_cb)** |
| --- | --- | --- |
| **Function** | | set callback function for status bar task(The loop call callback once a second) |
| **Params** | **in** | callback function for status bar  **task\_cb**  task(loop) |
| **out** | **Nothing** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | |  |

## MfSdkLvglSetStatusBarIcon

| **Prototype** | | **LIB\_EXPORT s32 MfSdkLvglSetStatusBarIcon(s32 x, s8\* iconname, s32 group)** | |
| --- | --- | --- | --- |
| **Function** | | display status bar icon | |
| **Params** | **in** | **x** | horizontal coordinate |
| **iconname** | icon name |
| **group** | icon group(max 5 groups, 0,1,2,3,4) |
| **out** | **Nothing** | |
| **return** | | **0** | **success** |
| **Other** | **fail** |
| **remark** | | **This API can only be in MfSdkLvglCreateStatusBarTask callback**  **function calls** | |
| **demo** | |  | |



## MfSdkLvglQuickSetStatusBarIcon

| **Prototype** | | **LIB\_EXPORT s32 MfSdkLvglQuickSetStatusBarIcon(s32 x, s8\* iconname, s32 group)** | |
| --- | --- | --- | --- |
| **Function** | | Quick display status bar icon | |
| **Params** | **in** | **x** | horizontal coordinate |
| **iconname** | icon name |
| **group** | icon group(max 5 groups, 0,1,2,3,4) |
| **out** | **Nothing** | |
| **return** | | **0** | **success** |
| **Other** | **fail** |
| **remark** | | **This API is not in MfSdkLvglCreateStatusBarTask callback**  **function calls** | |
| **demo** | |  | |

## MfSdkLvglSetStatusBar

| **Prototype** | | **LIB\_EXPORT void MfSdkLvglSetStatusBar(s32 height, lv\_color\_t bgColor)** |
| --- | --- | --- |
| **Function** | | set status bar config |
| **Params** | **in** | **height** status bar height |
| **bgColor** status bar background color |
| **out** | **Nothing** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | |  |

## MfSdkLvglGetUiBuff

**Prototype LIB\_EXPORT void MfSdkLvglGetUiBuff(void (\*getUI)(u8 \*buf,**

****

| **Function** | | Get the LVGL refresh screen buff |
| --- | --- | --- |
| **Params** | **in** | The callback function for refreshing the  **getUI**  screen |
| **out** | **Nothing** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | |  |

## MfSdkLvglLoadPng

| **Prototype** | | **LIB\_EXPORT s8 MfSdkLvglLoadPng(const s8 \*filename)** |
| --- | --- | --- |
| **Function** | | Load png image |
| **Params** | **in** | **filename** png image path |
| **out** | **Nothing** |
| **return** | | **Ref.MfSdkLvglOper\_E** |
| **remark** | |  |
| **demo** | |  |

## MfSdkLvglFreePng

| **Prototype** | | **LIB\_EXPORT s8 MfSdkLvglFreePng(const s8 \*filename)** |
| --- | --- | --- |
| **Function** | | Load png image |
| **Params** | **in** | **filename** png image path |
| **out** | **Nothing** |
| **return** | | **Ref.MfSdkLvglOper\_E** |
| **remark** | |  |
| **demo** | |  |



## MfSdkLvglGuiInit

| **Prototype** | | **LIB\_EXPORT void MfSdkLvglGuiInit(void)** |
| --- | --- | --- |
| **Function** | | init LVGL group |
| **Params** | **in** | **Nothing** |
| **out** | **Nothing** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | |  |

## MfSdkLvglGroupSet

| **Prototype** | | **LIB\_EXPORT void MfSdkLvglGroupSet(lv\_obj\_t\* obj)** |
| --- | --- | --- |
| **Function** | | obj add to group and set focus obj |
| **Params** | **in** | **obj** |
| **out** | **Nothing** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | |  |

# Memory module

## Module description

This module mainly includes APIs on memory operations.

## Module structure declaration

None.

## Constant declarations

None.



* 1. MfSdkMemFree

| **Prototype** | | **LIB\_EXPORT void MfSdkMemFree(void \*ptr)** |
| --- | --- | --- |
| **Function** | | ref. standard free |
| **Params** | **in** | **ptr** |
| **out** | **Nothing** |
| **return** | | **Nothing** |
| **remark** | |  |
| **demo** | |  |

* 1. MfSdkMemMalloc

| **Prototype** | | **LIB\_EXPORT void \*MfSdkMemMalloc(s32 size)** |
| --- | --- | --- |
| **Function** | | Ref. standard malloc |
| **Params** | **in** | **size** |
| **out** | **Nothing** |
| **return** | |  |
| **remark** | |  |
| **demo** | |  |

* 1. MfSdkMemRealloc

| **Prototype** | | **LIB\_EXPORT void \*MfSdkMemRealloc(void \*ptr, s32 size)** |
| --- | --- | --- |
| **Function** | | Ref. standard realloc |
| **Params** | **in** | **ptr** |
| **size** |
| **out** | **Nothing** |
| **return** | |  |



| **remark** |  |
| --- | --- |
| **demo** |  |

* 1. MfSdkMemCalloc

| **Prototype** | | **LIB\_EXPORT void \*MfSdkMemCalloc(s32 nitems, s32 size)** |
| --- | --- | --- |
| **Function** | |  |
| **Params** | **in** | **nitems** |
| **size** |
| **out** | **Nothing** |
| **return** | |  |
| **remark** | |  |
| **demo** | |  |

# Base module

## Module description

Base device.

## Module structure declaration

None.

## Constant declarations

None.

* 1. MfSdkBaseCheck

| **Prototype** | | **s32 MfSdkBaseCheck()** |
| --- | --- | --- |
| **Function** | | Whether the device uses a base |
| **Params** | **in** | **Nothing** |
| **out** | **Nothing** |



| **return** | **0 Not base** |
| --- | --- |
| **1 have base** |
| **remark** |  |
| **demo** | s32 ret = 0;  ret = MfSdkBaseCheck(); if(ret==1)  {  //have base  } |