Reference Manual

Generated by Doxygen 1.7.1

Sat Sep 21 2013 10:02:11

Contents

Chapter 1

Module Index

1.1 Modules

Here is a list of all modules:

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

Chapter 3

Module Documentation

3.1 LCD KPD Module Header.

Functions

- int lk_open (void)

  This API Function is used to open LCD and Keypad devices. Note: This function must be called before any LCD and Keypad functions.

- void lk_dispclr (void)

  This API Function clears all pixels on LCD. Note: This function must be called explicitly after lk_open to clear LCD.

- void lk_dispfill (void)

  This API Function fills all pixels on LCD.

- int lk_displineclr (unsigned char line_no)

  This API Function Clears pixels of the Specified line on LCD.

- int lk_disptext (unsigned char line_no, unsigned char column, unsigned char ∗data, unsigned char font)

  This API Function displays the string of data at specified line and column in a given font.

- void lk_close ()

  This API Function is used to close device.

- int lk_dispbmp (unsigned char xstartcord, unsigned char ystartcord, unsigned char xendcord, unsigned char yendcord, unsigned char ∗bmp)

  This API Function displays the BMP Image on LCD at user given coordinates. Windows tool provided to convert bmp to hex.

- int lk_dispbox (unsigned char line_no, unsigned char cmnd)

  This API Function Displays Box at Specified Line.

- int lk_getalpha (unsigned char line_no, unsigned char start_pos, unsigned char ∗data, unsigned char max_cnt, unsigned char count, unsigned char keytype)

This API function is used to pass a buffer with a Null string or a non-null string and the subsequent alpha numeric characters are filled from keypad entry by user. Entire String is displayed in LCD. Alpha Key is pressed to toggle among numeric and various alphabets for a single key.

- int lk_getnumeric (unsigned char line_no, unsigned char start_pos, unsigned char ∗data, unsigned char max_cnt, unsigned char count)

  This API function is used to pass a buffer with a Null string or a non-null string and the subsequent Numeric characters are filled from keypad entry by user. Entire String is displayed in LCD.

- int lk_getpassword (unsigned char ∗ptr, unsigned char line_no, unsigned char max_cnt)

  This API Function is used to get the password from the keypad. Digits entered will be displayed as '∗' on the display and fills the target buffer with entered digits.

- int lk_getamount (unsigned char line_no, unsigned char start_pos, unsigned char ∗ptr, unsigned char max_digits, unsigned char decimal_pos)

  This API function is used to get amount from user.

- int lk_dispbutton (unsigned char ∗button1, unsigned char ∗button2, unsigned char ∗button3, unsigned char ∗button4)

  This API Function Display the Four Buttons at 5th line of LCD.

- int lk_disphlight (unsigned char line_no)

  This API function is used to highlight a given row of text on LCD.

- int lk_dispfont (unsigned char ∗user_font, unsigned char font_width)

  This API Function which type of Font to be Displayed on LCD.

- int lk_bkl_timeout (int bkltime)

  This API function sets statically LCD backlight timeout. The default timeout is 4 sec. If you want the change the backlight timeout permanently you have to call this function once in your application.

- int lk_dispbklight (int val)

  This API function shows lcd backlight intensity.

- int lk_dispkpbklight (int val) int lk_lcdintensity(unsigned char value)

  This API function shows keypad backlight intensity.

- int lk_gsmsignal (unsigned long gsm_signal)

  This function is to display signal strength of GSM modem which is connected to ttymxc0. By giving 'at+csq' command to modem, we can get the GSM signal strength. By giving the GSM signal strength value to this function we can update the signal strength bars on top of the LCD.

- unsigned char lk_getkey (void)

  This API Function is used to get key that is pressed. It will wait until key is pressed and returns the key value.

- unsigned char lk_getkey_wait (void)

  This function returns key pressed at that particular instant, if there is no key event returns 0xff. This function is similar to lk_getkey, only difference is it returns 0xff if there is no key pressed and it won waits for a key.

- int lk_buzzer_time ()

  This API Function gives buzzer Beep delay of 20ms time.

- int lk_buzzer (int count)

  This API Function gives buzzer Beep count time.

- int lk_underline (unsigned char line_no, unsigned char col, unsigned char length, unsigned char font_width)

  This API Function underlines the text On LCD.

- int lk_getpinpadid (char ∗machineid)

  This API function is used to get the PIN PAD ID. (Pre Requirements: This id should be set at u-boot prompt before call this API like below VISIONTEK # setenv machineid 1234567890 VISIONTEK # saveenv).

- int lk_gethwid (char ∗hwid)

  This API function is used to get the Hardware ID. (Pre Requirements: This id should be set at u-boot prompt before call this API like below VISIONTEK # setenv hwid 87654321 VISIONTEK # saveenv).

3.1.1 Function Documentation

3.1.1.1 int lk_bkl_timeout (int *bkltime*)

This API function sets statically LCD backlight timeout. The default timeout is 4 sec. If you want the change the backlight timeout permanently you have to call this function once in your application.

Parameters

bkltime Description: back light timeout in seconds and the range is 4 sec to 30 sec. Type: Input Range: 4 - 30

Returns

0 on Success. -1 on Failure. -2 on Argument errors.

3.1.1.2 int lk_buzzer (int *count*)

This API Function gives buzzer Beep count time.

Parameters

count Description: Time count to get buzzer Type: Input Range: 0 - 10

Returns

0 on Success. -1 on Failure. Limits are like below if count <=0 it will give one beep. if count > 10 it gives maximum of 10 Beeps

3.1.1.3 int lk_buzzer_time ( )

This API Function gives buzzer Beep delay of 20ms time.

Returns

0 on Success. -1 on Failure.

3.1.1.4 void lk_close ( )

This API Function is used to close device.

Returns

: None.

3.1.1.5 int lk_dispbklight (int *val*)

This API function shows lcd backlight intensity.

Parameters

value value (0-3) default value is 3.

Returns

0 on Success. -1 on Failure.

3.1.1.6 int lk_dispbmp (unsigned char *xstartcord,* unsigned char *ystartcord,* unsigned char *xendcord,* unsigned char *yendcord,* unsigned char $*$ *bmp*)

This API Function displays the BMP Image on LCD at user given coordinates. Windows tool provided to convert bmp to hex.

Parameters

xstartcord Description: Start of x co-ordinate Type: Input Range: 0 - 127

xendcord Description: end of x co-ordinate Type: Input Range: upto 127

ystartcord Description: Start of y co-ordinate Type: Input Range: 0 - 63

yendcord Description: end of y co-ordinate Type: Input Range: upto 63

bmp Description: Displays BMP hex data Type: Input Range: (y) 64 x 128 (x)

Returns

0 on SUCCESS. -1 on Error. -2 if there is arguments error.

3.1.1.7 int lk_dispbox (unsigned char *line_no,* unsigned char *cmnd*)

This API Function Displays Box at Specified Line.

Parameters

line_no Description: line number to display the box on the LCD. Type: Input Range: 0 - 5

cmnd Description: cmnd should be 0x00 or 0x01. Type: Input Range: 0 - 1 '0' for box '1' for no box Note: display text in the box after drawing box (i.e. don't draw box after write text)

Returns

0 on Success. -1 on Error. -2 if There is arguments error.

3.1.1.8 int lk_dispbutton (unsigned char ∗ *button1,* unsigned char ∗ *button2,* unsigned char ∗ *button3,* unsigned char ∗ *button4*)

This API Function Display the Four Buttons at 5th line of LCD.

Parameters

button1,button2,button3,button4 Description: Buffers to display 4 characters of each button Type: Input Range: 4 characers only including space character

Returns

0 on Success. -1 on Error.

3.1.1.9 void lk_dispclr (void)

This API Function clears all pixels on LCD. Note: This function must be called explicitly after lk_open to clear LCD.

Returns

0 on Success. -1 on Failure.

3.1.1.10 void lk_dispfill (void)

This API Function fills all pixels on LCD.

Returns

None.

3.1.1.11 int lk_dispfont (unsigned char ∗ *user_font,* unsigned char *font_width*)

This API Function which type of Font to be Displayed on LCD.

Parameters

∗***user_font*** Description: which font table to be displayed of 6x8 character size. Type: Input Range: 255 characters font table

font_width Description: width of the given Font. Type: Input Range: 6

Returns

0 on Success -1 on Error Note: This function should call immediately after lk_open()

3.1.1.12 int lk_disphlight (unsigned char *line_no*)

This API function is used to highlight a given row of text on LCD.

Parameters

line_no Description: line number to display the text on the LCD. Type: Input Range: 0 - 5

Returns

0 on Success. -1 on Failure. -2 on Argument Errors.

3.1.1.13 int lk_dispkpbklight (int *val*)

This API function shows keypad backlight intensity.

Parameters

value value (0-3) default value is 3.

Returns

0 on Success. -1 on Failure. This API shows LCD Intensity.

Parameters

value Description: (0-63) Default value is 24. Type: Input Range: 0 - 63

Returns

: 0 on Success. -1 on Failure. -2 on Argument Errors.

3.1.1.14 int lk_displineclr (unsigned char *line_no*)

This API Function Clears pixels of the Specified line on LCD.

Parameters

line_no Description: line number to clears the LCD. Type: Input Range: 0 - 5

Returns

0 on Success. -1 on Error. -2 if there is argument error.

3.1.1.15 int lk_disptext (unsigned char *line_no,* unsigned char *column,* unsigned char ∗ *data,* unsigned char *font*)

This API Function displays the string of data at specified line and column in a given font.

Parameters

line_no Description: line number to display the txt on the LCD. Type: Input Range: Depends on font 0-5 - font 0; 0-4 - font 1; 0-5 - font 2; 0-4 - font 3;

column Description: column number to Display the txt. Type: Input Range: 0 - 20 - Font 0 Font1; 0 - 10 - Font 2; 0 - 18 - Font 3

data Description: Buffer to display text data on LCD. Type: Input Range: Data Length range minimum is 1 character and maximum 124 characters,

font Description: type of font. Type: Input Range: 0 - 3 '0' 6x8 then each line displays 21 characters and lines range is 0-5 '1' 6x16 then each line displays 21 characters and possible 4 lines (because height is doubled) '2' 12x8 then each line displays 10 characters and lines range is 0-5(because width is doubled) '3' 12x16 then each line displays 10 characters and possible 4 lines (both doubled)

Returns

Number of characters Displayed on LCD -1 on Failure. -2 if there is arguments error.

3.1.1.16 int lk_getalpha (unsigned char *line_no*, unsigned char *start_pos*, unsigned char ∗ *data*, unsigned char *max_cnt*, unsigned char *count*, unsigned char *keytype*)

This API function is used to pass a buffer with a Null string or a non-null string and the subsequent alpha numeric characters are filled from keypad entry by user. Entire String is displayed in LCD. Alpha Key is pressed to toggle among numeric and various alphabets for a single key.

Parameters

line_no Description: line number to display edit keys on the LCD. Type: Input Range: 0 - 5

start_pos Description: Specifies the column position to display or to take input. Type: Input Range: 0 - 20

data Description: Target buffer to be filled with edit keys data. Type: Input / Output Range: only 21 characters are visible if starts at 0

max_cnt Description: Number of alphabets to be taken from the keypad.(maximum 21 alphabets are visible at once on display if starts at 0) Type: Input Range: 21 characters are visible (Limit depends on buffer size taken)

count Description: This field signifies the length of a non-null string in the buffer. In the case of a Null String it is zero. Both start_pos and count act as a base and offset for the input from keypad. Type: Input Range: 0 - 20

keytype Description: edit key type (should be 0) Type: Input Range: 0 for shift key (Future expansion)

Returns

Number of characters entered 0 pressed Enter key without data. -1 Cancel key Pressed. -2 if there is argument error.

3.1.1.17 int lk_getamount (unsigned char *line_no*, unsigned char *start_pos*, unsigned char ∗ *ptr*, unsigned char *max_digits*, unsigned char *decimal_pos*)

This API function is used to get amount from user.

Parameters

line_no Description: line number to display on the LCD. Type: Input Range: 0 - 5

start_pos Description: Specifies the column position to take input. Type: Input Range: 0 - 20

ptr Description: Target buffer to be filled with the amount. Type: Input Range: 21 characters ('∗'s) are visible (Limit depends on buffer size taken)

max_digits Description: Maximum number of numeric characters to be taken from the keypad. 21 is the maximum numeric characters are visible at once on display if we selected Start position as 0. Type: Input Range: 21 characters ('∗'s) are visible (Limit depends on buffer size taken)

decimal_pos Description: Number of decimal positions required after decimal point + 1.Suppose we want 2 decimal positions we have give decimal position as 3. Range is 1 to 4(we can obtain max 3 decimal positions). Type: Input Range: 1 - 4

Returns

Number of characters entered 0 pressed Enter key without data. -1 Cancel key Pressed. -2 if there is argument error

3.1.1.18 int lk_gethwid (char ∗ *hwid*)

This API function is used to get the Hardware ID. (Pre Requirements: This id should be set at u-boot prompt before call this API like below VISIONTEK # setenv hwid 87654321 VISIONTEK # saveenv).

Parameters

hwid Description: Read Hardware ID of 8 digits from u-boot environment variables. Type: Output Range: 8 digits (bytes)

Returns

0 on Success. -1 on Failure.

3.1.1.19 unsigned char lk_getkey (void)

This API Function is used to get key that is pressed. It will wait until key is pressed and returns the key value.

Returns

Numeric keys or 255

3.1.1.20 unsigned char lk_getkey_wait (void)

This function returns key pressed at that particular instant, if there is no key event returns 0xff. This function is similar to lk_getkey, only difference is it returns 0xff if there is no key pressed and it won waits for a key.

Returns

Corresponding key value or 0xff.

3.1.1.21 int lk_getnumeric (unsigned char *line_no,* unsigned char *start_pos,* unsigned char ∗ *data,* unsigned char *max_cnt,* unsigned char *count*)

This API function is used to pass a buffer with a Null string or a non-null string and the subsequent Numeric characters are filled from keypad entry by user. Entire String is displayed in LCD.

Parameters

line_no Description: line number to display edit keys on the LCD. Type: Input Range: 0 - 5

start_pos Description: Specifies the column position to display or to take input. Type: Input Range: 0 - 20

data Description: Target buffer to be filled with edit keys data. Type: Input / Output Range: only 21 characters are visible if starts at 0

max_cnt Description: Number of numerics to be taken from the keypad (maximum 21 alphabets are visible at once on display if starts at 0) Type: Input Range: 21 characters are visible (Limit depends on buffer size taken)

count Description: This field signifies the length of a non-null string in the buffer. In the case of a Null String it is zero. Both start_pos and count act as a base and offset for the input from keypad. Type: Input Range: 0 - 20

Returns

Number of characters entered 0 pressed Enter key without data. -1 Cancel key Pressed. -2 if there is argument error.

NOTE: 21 character input not displayed no LCD.

3.1.1.22 int lk_getpassword (unsigned char ∗ *ptr,* unsigned char *line_no,* unsigned char *max_cnt*)

This API Function is used to get the password from the keypad. Digits entered will be displayed as '∗' on the display and fills the target buffer with entered digits.

Parameters

ptr Description: buffer to be filled with the password entered through keyboard. Type: Input Range: 21 characters ('∗'s) are visible (Limit depends on buffer size taken)

line_no Description: line number to display on the LCD. Type: Input Range: 0 - 5

max_cnt Description: Maximum Number of digits to be taken from the keypad. Type: Input Range: 21 characters ('∗'s) are visible (Limit depends on buffer size taken)

Returns

Number of digits entered on Success. 0 if user pressed enter key without data. -1 if Cancel key is pressed. -2 on the argument errors.

3.1.1.23 int lk_getpinpadid (char ∗ *machineid*)

This API function is used to get the PIN PAD ID. (Pre Requirements: This id should be set at u-boot prompt before call this API like below VISIONTEK # setenv machineid 1234567890 VISIONTEK # saveenv).

Parameters

machineid Description: Read machine ID of 10 digits from u-boot environment variables. Type: Output Range: 10 digits (bytes)

Returns

Return Values 0 on SUCCESS. -1 on Failure.

3.1.1.24 int lk_gsmsignal (unsigned long *gsm_signal*)

This function is to display signal strength of GSM modem which is connected to ttymxc0. By giving 'at+csq' command to modem, we can get the GSM signal strength. By giving the GSM signal strength value to this function we can update the signal strength bars on top of the LCD.

Parameters

gsm_signal Description: GSM modem signal strength value. Type: Input Range: 0 - 31 (if wit at+csq receive 99 then pass 0 to show no signal on LCD) Greater than 31 or less than 1 - No bars Greater than 25 - 4bars Greater than 15 - 3bars Greater than 10 - 2bars Greater than 1 - 1bar

Returns

0 on Success. -1 on Failure.

3.1.1.25 int lk_open (void)

This API Function is used to open LCD and Keypad devices. Note: This function must be called before any LCD and Keypad functions.

Returns

0 on Success. -1 on failure.

3.1.1.26 int lk_underline (unsigned char *line_no,* unsigned char *col,* unsigned char *length,* unsigned char *font_width*)

This API Function underlines the text On LCD.

Parameters

line_no Description: line number to display the text on the LCD. Type: Input Range: 0 - 5

col Description: Column number to starting underline. Type: Input Range: 0 -20

length,: Description: Number of characters to underline. Type: Input Range: max 21 characters (depends on col)

font_width Description: Font width is 6. Type: Input Range: 6

Returns

0 on success. -1 on Error. -2 on Arguments Error.

3.2 RTC Module Header.

Functions

- int lk_getrtc (struct tm ∗)
  This API Function gets the time from the RTC chip or from the OS if already set.
- int lk_setrtc (struct tm ∗)
  This API function sets the time to the RTC chip as well as for OS.

3.2.1 Function Documentation

3.2.1.1 int lk_getrtc (struct tm ∗)

This API Function gets the time from the RTC chip or from the OS if already set.

Parameters

∗ A pointer to tm structure. (time.h)The API call populates this structure

Returns

0 on success. -1 on failure.

3.2.1.2 int lk_setrtc (struct tm ∗)

This API function sets the time to the RTC chip as well as for OS.

Parameters

∗ A pointer to tm structure. The API call populates this structure.

Returns

0 on success. -1 on failure.

3.3 Printer Module Header.

Functions

- int prn_open ()

  This API Function is used to open the printer device . Note: This function must be called before any printer function.

- int prn_close ()

  This API Function is used to close the printer device .

- int prn_write_text (unsigned char ∗text2, int len, int font)

  This API Function is used to prints the text .

- int prn_write_bmp (unsigned char ∗bitmap, long len)

  This API Function is used to print bmp data .

- int prn_paper_feed (int scanlines)

  This API Function is used to advance the paper .

- int prn_paperstatus (void)

  This API Function is used printer papar status .

3.3.1 Function Documentation

3.3.1.1 int prn_close ( )

This API Function is used to close the printer device .

Returns

1 if the device closed successfully -1 if the device is not close.

3.3.1.2 int prn_open ( )

This API Function is used to open the printer device . Note: This function must be called before any printer function.

Returns

1 on Success. 2 if the device is already opend -1 on failure.

3.3.1.3 int prn_paper_feed (int *scanlines*)

This API Function is used to advance the paper .

Parameters

scanlines,: Description: scanlines is to be passed as parameter (1 text line is equal to 17 scan lines). Type: INPUT

Returns

0 no. of characters written on success -1 if error occurs. -2 if the length is more . -3 NO Paper . -4 Low Battery. -5 Max temp . -6 No Lines. -7 WRITE_ERROR.

3.3.1.4 int prn_paperstatus (void)

This API Function is used printer papar status .

Returns

0 on Success. 3 on Error.

3.3.1.5 int prn_write_bmp (unsigned char ∗ *bitmap,* long *len*)

This API Function is used to print bmp data .

Parameters

bitmap,: Description: pointer to the buffer from which the data to be printed.the image is 384x480 pixels type: INPUT

len,: Description: length of the text. type: INPUT

Returns

0 no. of characters written on success -1 if error occurs. -2 if the length is more . -3 NO Paper . -4 Low Battery. -5 Max temp . -6 No Lines. -7 WRITE_ERROR.

3.3.1.6 int prn_write_text (unsigned char ∗ *text2*, int *len*, int *font*)

This API Function is used to prints the text .

Parameters

text2,: Description: pointer to the buffer from which the text to be printed. Type: INPUT

len,: Description: length of the text. Type: INPUT

font,: Description: font of the the text presently there are two fonts are supported.(1 and 2 fonts). Type: INPUT

Returns

0 on Success. -1 if the device not opened . -2 if the length is more . -3 NO Paper . -4 Low Battery. -5 Max temp . -6 No Lines. -7 WRITE_ERROR.

3.4 Barcode Module Header.

Functions

- int bar_code_open (void)

  This API Function is used to open barcode device. Note: This function must be called before any barcode function.

- int bar_code_read (unsigned char ∗rxbuf, int len)

  This API Function reads the barcode.

- int bar_code_close (void)

  This API Function closes the barcode.

3.4.1 Function Documentation

3.4.1.1 int bar_code_close (void)

This API Function closes the barcode.

Returns

0 on Success. -1 on Error.

3.4.1.2 int bar_code_open (void)

This API Function is used to open barcode device. Note: This function must be called before any barcode function.

Returns

0 on Success. -1 on failure.

3.4.1.3 int bar_code_read (unsigned char ∗ *rxbuf,* int *len*)

This API Function reads the barcode.

Parameters

rxbuf,:	Description: buffer used to store barcode data. type: output Range: 0 - 256
len,:	Description:len means size of data to be read (For future expansion. Now, it will return on enter) Returns data size read. type: input Range: 0 - 256

Returns

>0 on Success. -1 on Error.

3.5 RFID Module Header.

Functions

- int tama_open (void)

  This API Function is used to open RFID device. Note: This function must be called before any rfid function.

- void tama_close (void)

  This API Function is used to close rfid device. Note: This function must be called before any rfid function.

- int tama_detectTarget (unsigned char Tg_Nos, unsigned char ∗Tg_Detected, unsigned char ∗Tg_-ID)

  This API Function is used to detect the targets of RFID Card reader.

- int tama_authenticate_mifare (unsigned char Tg_No, unsigned char Blk_Addr, unsigned char index, unsigned char Key_Type)

  This API Function authenticates a Tag's Block for writing/reading with the said key.

- int tama_authenticate_mifare_key (unsigned char Tg_No, unsigned char Blk_Addr, unsigned char ∗Key, unsigned char Key_Type)

  This API Function authenticates a Tag's Block for writing/reading with the said key.

- int tama_halt_mifare (unsigned char Tg_LID)

  This API Function halts (Deselects) a tag for access.

- int tama_read_target (unsigned char Blk_Addr, unsigned char ∗Data_In)

  This API Function reads the block data and stores it into passed address.

- int tama_write_target (unsigned char Blk_Addr, unsigned char ∗Data_Out, int Dlen)

  This API Function writes the data sent to the said block.

- int tama_rffieldon (void)

  This API Function is used to power ON RF field. Note: This function must be called after tama_open function.

- int tama_rffieldoff (void)

  This API Function is used to power OFF RF field. Note: This function must be called after tama_open function.

3.5.1 Function Documentation

3.5.1.1 int tama_authenticate_mifare (unsigned char *Tg_No*, unsigned char *Blk_Addr*, unsigned char *index*, unsigned char *Key_Type*)

This API Function authenticates a Tag's Block for writing/reading with the said key.

Parameters

Tg_No Description: Tag Number to authenticate(0 - Card 1, 1 - Card 2). Type: INPUT

Blk_Addr Description: Block address Type: INPUT

index Description:key index Type: INPUT

Key_Type Description:Key Type (Key A - 0x0A /Key B - 0x0B) Type: INPUT

Returns

0 on success. < 1 on failure.

3.5.1.2 int tama_authenticate_mifare_key (unsigned char *Tg_No*, unsigned char *Blk_Addr*, unsigned char ∗ *Key*, unsigned char *Key_Type*)

This API Function authenticates a Tag's Block for writing/reading with the said key.

Parameters

Tg_No Description: Tag Number to authenticate(1- Card 1, 2 - Card 2). Type: INPUT

Blk_Addr Description: Block address Type: INPUT

∗***Key*** Description:key Type: INPUT

Key_Type Description:Key type (Key A - 0x0A /Key B - 0x0B). Type: INPUT

Returns

0 on success. < 1 on failure.

3.5.1.3 void tama_close (void)

This API Function is used to close rfid device. Note: This function must be called before any rfid function.

3.5.1.4 int tama_detectTarget (unsigned char *Tg_Nos,* unsigned char $*$ *Tg_Detected,* unsigned char $*$ *Tg_ID*)

This API Function is used to detect the targets of RFID Card reader.

Parameters

Tg_Nos Description: Number of targets to detect(Max. 2). Type: INPUT Range: Max. 2

Tg_Detected Description:Address of memory to store number of Tags detected. Type: INPUT

Tg_ID,: Description:Address of memory to store Tag ID. Type: INPUT

Returns

Number of Tags read/initialized is stored in the address passed to Tg_Detected.The read Tag is stored in location pointed by Tag ID pointer. 0 on Success. -1 on Failure.

3.5.1.5 int tama_halt_mifare (unsigned char *Tg_LID*)

This API Function halts (Deselects) a tag for access.

Parameters

Tg_LID Description: Tag's logical ID. Type: INPUT

Returns

0 on success. < 1 on failure.

3.5.1.6 int tama_open (void)

This API Function is used to open RFID device. Note: This function must be called before any rfid function.

Returns

0 opened file descriptor on success. < 1 on failure.

3.5.1.7 int tama_read_target (unsigned char *Blk_Addr,* unsigned char $*$ *Data_In*)

This API Function reads the block data and stores it into passed address.

Parameters

Blk_Addr Description: Block address to read. Type: INPUT

Data_In Description:pointer to the data in buffer. Type: INPUT

Returns

0 on success. < 1 on failure.

3.5.1.8 int tama_rffieldoff (void)

This API Function is used to power OFF RF field. Note: This function must be called after tama_open function.

Returns

0 opened file descriptor on success. $<$ 1 on failure.

3.5.1.9 int tama_rffieldon (void)

This API Function is used to power ON RF field. Note: This function must be called after tama_open function.

Returns

0 opened file descriptor on success. $<$ 1 on failure.

3.5.1.10 int tama_write_target (unsigned char *Blk_Addr,* unsigned char $*$ *Data_Out,* int *Dlen*)

This API Function writes the data sent to the said block.

Parameters

Blk_Addr Description: Block address to write. Type: INPUT

Data_Out Description:pointer to the data . Type: INPUT

Dlen Description:length of data to write . Type: INPUT

Returns

0 on success. $<$ 1 on failure.

3.6 Magswipe Device Module Header .

Functions

- int mscr_open (void)

  This API Function is used to open magswipe device. Note: This function must be called before any magswipe function.

- int mscr_pwr_enable (void)

  This API Function is used to power ON magswipe device.

- int mscr_pwr_disable (void)

  This API Function is used to power OFF magswipe device.

- int mscr_read (char $*$track_1_data, char $*$track_2_data, char $*$track_3_data)

  This API function reads the data of the both tracks from the card reader .

- int mscr_flush (void)

This API Function clears the card reader device buffer returns SUCCESS on success, returns MAG_SW_-ERR if the device not opened.

- int mscr_getstate (void)

  This API Function is used to check wether the device is having valid data (both track1 and track2 data).

- int mscr_close (void)

  This API Function is to close magnetic swipe device.

3.6.1 Function Documentation

3.6.1.1 int mscr_close (void)

This API Function is to close magnetic swipe device.

Returns

1 on Success. MAG_SW_ERR (-1) on failure.

3.6.1.2 int mscr_flush (void)

This API Function clears the card reader device buffer returns SUCCESS on success, returns MAG_SW_-ERR if the device not opened.

Returns

1 on Success. MAG_SW_ERR (-1) on Failure

3.6.1.3 int mscr_getstate (void)

This API Function is used to check wether the device is having valid data (both track1 and track2 data).

Returns

1 on Success. MAG_SW_ERR (-1) if there is no valid data.

3.6.1.4 int mscr_open (void)

This API Function is used to open magswipe device. Note: This function must be called before any magswipe function.

Returns

1 on Success. MAG_SW_ERR (-1) on Failure

3.6.1.5 int mscr_pwr_disable (void)

This API Function is used to power OFF magswipe device.

Returns

1 on Success. -1 on Failure

3.6.1.6 int mscr_pwr_enable (void)

This API Function is used to power ON magswipe device.

Returns

1 on Success. -1 on Failure

3.6.1.7 int mscr_read (char ∗ *track_1_data,* char ∗ *track_2_data,* char ∗ *track_3_data*)

This API function reads the data of the both tracks from the card reader .

Parameters

track_1_data,track_2_data,track_3_data,:

Description: Pointers to the buffers where tracks data to be copied are to be passed as arguments, track1 and track2 are charecter pointer data types and the memory allocated to them should be length of the expected data+1(suggested 256 bytes each) Type: input Range: 256 bytes

Returns

’1’ if only track1 data is valid and copied data into track_1_data buffer ’2’ if only track2 data is valid and copied data into track_2_data buffer ’3’ if only track2 data is valid and copied data into track_3_-data buffer ’4’ if track 1 & 2 are valid and copied data into track_1_data buffer & track_2_data buffer ’5’ if track 2 & 3 are valid and copied data into track_2_data buffer & track_3_data buffer ’6’ if track 1 & 3 are valid and copied data into track_1_data buffer & track_3_data buffer ’7’ if track 1, 2 & 3 are valid and copied into all three buffers, MAG_SW_ERR (-1) if there is no valid data of all three tracks or device not opened.

3.7 SAM And IFD Module Header.

Functions

- int SAM_Open (void)

  This function checks for the existence of the Sam card reader device and if found prepares it for further communication. Before attempting any operat ∗ ions related to Sam card reader device this function should be called.

- int SAM_Close ()

  This function closes the SAM device. After finishing all the operations related to SAM card reader this function should get called.

- int SAM_GetAtr (unsigned int ∗length, unsigned char ∗buffer)

  This API reads ATR from the card.

- int SAM_SendCommand (unsigned char ∗CAPDU_Data, unsigned int CLen, unsigned char ∗RAPDU_Data, int ∗RLen)

  This API is used to send all the EMV complaint CAPDUs to the card. The API call sends the command contained in CAPDU Data buffer and gives the response and ∗its length out through RAPDU Data and RLen variables respectively.

- int SAM_ColdReset (int selection)

  This API is to power up the card,after this call Vcc is applied as selected during card select API. After this call only we can do any command operations o ∗n card.

- int SAM_Select_S1 (int volt)

  This API is to select the SAM slot one. And send return value to SAM_ColdReset API.

- int SAM_Select_S2 (int volt)

  This API is to select the SAM slot two. And send return value to SAM_ColdReset API.

- int SAM_Poweroff (void)

  This API is used to Power off SAM module.

3.7.1 Function Documentation

3.7.1.1 int SAM_Close ( )

This function closes the SAM device. After finishing all the operations related to SAM card reader this function should get called.

Returns

’0’ on Success. ’-1’ on Error

3.7.1.2 int SAM_ColdReset (int *selection*)

This API is to power up the card,after this call Vcc is applied as selected during card select API. After this call only we can do any command operations o ∗n card.

Parameters

selection Description: 1- card1 5v 2- card1 3v 3- card1 1.8v 4- card2 5v 5- card2 3v 6- card3 1.8v type:input

Returns

’0’ on Success ’-1’ on Failure

3.7.1.3 int SAM_GetAtr (unsigned int ∗ *length,* unsigned char ∗ *buffer*)

This API reads ATR from the card.

Parameters

∗***length,:*** Description: length of ATR from card in bytes. type: output
∗***buffer*** Description: pointer to buffer which holds ATR response from the card . type: output

Returns

'0' on Success '-1' on Error 'if unable to receive ATR this happens only if card is not present in slot'.

3.7.1.4 int SAM_Open (void)

This function checks for the existence of the Sam card reader device and if found prepares it for further communication. Before attempting any operat ∗ ions related to Sam card reader device this function should be called.

Returns

'0' on Success. '-1' on Error

3.7.1.5 int SAM_Poweroff (void)

This API is used to Power off SAM module.

Returns

-1 Failure 0 Success

3.7.1.6 int SAM_Select_S1 (int *volt*)

This API is to select the SAM slot one. And send return value to SAM_ColdReset API.

Parameters

volt Description: 1 - 5v 2 - 3v 3 - 1.8v

Returns

10 for 5v 11 for 3v 12 for 1.8v

3.7.1.7 int SAM_Select_S2 (int *volt*)

This API is to select the SAM slot two. And send return value to SAM_ColdReset API.

Parameters

volt Description: 1 - 5v 2 - 3v 3 - 1.8v

Returns

7 for 5v 8 for 3v 9 for 1.8v

3.7.1.8 int SAM_SendCommand (unsigned char ∗ *CAPDU_Data*, unsigned int *CLen*, unsigned char ∗ *RAPDU_Data*, int ∗ *RLen*)

This API is used to send all the EMV complaint CAPDUs to the card. The API call sends the command contained in CAPDU Data buffer and gives the response and ∗its length out through RAPDU Data and RLen variables respectively.

Parameters

∗***CAPDU_Data***	Description:Buffer holding the command APDU in the standard format type:input
CLen	Description:length of the CAPDU Data buffer type:input
∗***RAPDU_Data***	Description:Buffer to hold the response APDU in the standard format type:output
RLen	Description:variable to hold response length. type:output

Returns

’0’ on Success ’-1’ on Failure ’communication error occurred and the device is closed’.

3.8 USB-HOST Module Header.

Functions

- int usb_pwr_enable (void)

  This API Function is used to power on usb-host device.

- int usb_pwr_disable (void)

  This API Function is used to power off usb-host device.

3.8.1 Function Documentation

3.8.1.1 int usb_pwr_disable (void)

This API Function is used to power off usb-host device.

Returns

0 on Success. -1 on failure.

3.8.1.2 int usb_pwr_enable (void)

This API Function is used to power on usb-host device.

Returns

0 on Success. -1 on failure.

3.9 Audio Module Header.

Functions

- int audio_pwr_enable (void)

    This API Function is used to power on audio device.

- int audio_pwr_disable (void)

    This API Function is used to power off audio device.

3.9.1 Function Documentation

3.9.1.1 int audio_pwr_disable (void)

This API Function is used to power off audio device.

Returns

0 on Success. -1 on failure.

3.9.1.2 int audio_pwr_enable (void)

This API Function is used to power on audio device.

Returns

0 on Success. -1 on failure.

3.10 Wi-fi Module Header.

Functions

- int wifi_enable (void)

    This API Function is used to enable wifi functionality.

- int wifi_disable (void)

    This API function is used to disable wifi functionality.

- int wifi_rfpower_on (void)

    This API function is used to power on the RF section of wifi.

- int wifi_rfpower_off (void)

    This API function is used to power off the RF section of wifi.

3.10.1 Detailed Description

* * *

3.10.2 Function Documentation

3.10.2.1 int wifi_disable (void)

This API function is used to disable wifi functionality.

Returns

0 On Success. -1 On Failure.

3.10.2.2 int wifi_enable (void)

This API Function is used to enable wifi functionality.

Returns

0 on Success. -1 on failure.

3.10.2.3 int wifi_rfpower_off (void)

This API function is used to power off the RF section of wifi.

Returns

0 on Success. -1 on Failure.

3.10.2.4 int wifi_rfpower_on (void)

This API function is used to power on the RF section of wifi.

Returns

0 on Success. -1 on Failure. -2 on wifi is not enable.

Chapter 4

File Documentation

4.1 GL14pos.h File Reference

Functions

- int lk_open (void)

  This API Function is used to open LCD and Keypad devices. Note: This function must be called before any LCD and Keypad functions.

- void lk_dispclr (void)

  This API Function clears all pixels on LCD. Note: This function must be called explicitly after lk_open to clear LCD.

- void lk_dispfill (void)

  This API Function fills all pixels on LCD.

- int lk_displineclr (unsigned char line_no)

  This API Function Clears pixels of the Specified line on LCD.

- int lk_disptext (unsigned char line_no, unsigned char column, unsigned char ∗data, unsigned char font)

  This API Function displays the string of data at specified line and column in a given font.

- void lk_close ()

  This API Function is used to close device.

- int lk_dispbmp (unsigned char xstartcord, unsigned char ystartcord, unsigned char xendcord, unsigned char yendcord, unsigned char ∗bmp)

  This API Function displays the BMP Image on LCD at user given coordinates. Windows tool provided to convert bmp to hex.

- int lk_dispbox (unsigned char line_no, unsigned char cmnd)

  This API Function Displays Box at Specified Line.

- int lk_getalpha (unsigned char line_no, unsigned char start_pos, unsigned char ∗data, unsigned char max_cnt, unsigned char count, unsigned char keytype)

This API function is used to pass a buffer with a Null string or a non-null string and the subsequent alpha numeric characters are filled from keypad entry by user. Entire String is displayed in LCD. Alpha Key is pressed to toggle among numeric and various alphabets for a single key.

- int lk_getnumeric (unsigned char line_no, unsigned char start_pos, unsigned char ∗data, unsigned char max_cnt, unsigned char count)

  This API function is used to pass a buffer with a Null string or a non-null string and the subsequent Numeric characters are filled from keypad entry by user. Entire String is displayed in LCD.

- int lk_getpassword (unsigned char ∗ptr, unsigned char line_no, unsigned char max_cnt)

  This API Function is used to get the password from the keypad. Digits entered will be displayed as '∗' on the display and fills the target buffer with entered digits.

- int lk_getamount (unsigned char line_no, unsigned char start_pos, unsigned char ∗ptr, unsigned char max_digits, unsigned char decimal_pos)

  This API function is used to get amount from user.

- int lk_dispbutton (unsigned char ∗button1, unsigned char ∗button2, unsigned char ∗button3, unsigned char ∗button4)

  This API Function Display the Four Buttons at 5th line of LCD.

- int lk_disphlight (unsigned char line_no)

  This API function is used to highlight a given row of text on LCD.

- int lk_dispfont (unsigned char ∗user_font, unsigned char font_width)

  This API Function which type of Font to be Displayed on LCD.

- int lk_bkl_timeout (int bkltime)

  This API function sets statically LCD backlight timeout. The default timeout is 4 sec. If you want the change the backlight timeout permanently you have to call this function once in your application.

- int lk_dispbklight (int val)

  This API function shows lcd backlight intensity.

- int lk_dispkpbklight (int val) int lk_lcdintensity(unsigned char value)

  This API function shows keypad backlight intensity.

- int lk_gsmsignal (unsigned long gsm_signal)

  This function is to display signal strength of GSM modem which is connected to ttymxc0. By giving 'at+csq' command to modem, we can get the GSM signal strength. By giving the GSM signal strength value to this function we can update the signal strength bars on top of the LCD.

- unsigned char lk_getkey (void)

  This API Function is used to get key that is pressed. It will wait until key is pressed and returns the key value.

- unsigned char lk_getkey_wait (void)

  This function returns key pressed at that particular instant, if there is no key event returns 0xff. This function is similar to lk_getkey, only difference is it returns 0xff if there is no key pressed and it won waits for a key.

- int lk_buzzer_time ()

  This API Function gives buzzer Beep delay of 20ms time.

- int lk_buzzer (int count)

  This API Function gives buzzer Beep count time.

- int lk_underline (unsigned char line_no, unsigned char col, unsigned char length, unsigned char font_width)

  This API Function underlines the text On LCD.

- int lk_getpinpadid (char ∗machineid)

  This API function is used to get the PIN PAD ID. (Pre Requirements: This id should be set at u-boot prompt before call this API like below VISIONTEK # setenv machineid 1234567890 VISIONTEK # saveenv).

- int lk_gethwid (char ∗hwid)

  This API function is used to get the Hardware ID. (Pre Requirements: This id should be set at u-boot prompt before call this API like below VISIONTEK # setenv hwid 87654321 VISIONTEK # saveenv).

- int lk_getrtc (struct tm ∗)

  This API Function gets the time from the RTC chip or from the OS if already set.

- int lk_setrtc (struct tm ∗)

  This API function sets the time to the RTC chip as well as for OS.

- int prn_open ()

  This API Function is used to open the printer device . Note: This function must be called before any printer function.

- int prn_close ()

  This API Function is used to close the printer device .

- int prn_write_text (unsigned char ∗text2, int len, int font)

  This API Function is used to prints the text .

- int prn_write_bmp (unsigned char ∗bitmap, long len)

  This API Function is used to print bmp data .

- int prn_paper_feed (int scanlines)

  This API Function is used to advance the paper .

- int prn_paperstatus (void)

  This API Function is used printer papar status .

- int bar_code_open (void)

  This API Function is used to open barcode device. Note: This function must be called before any barcode function.

- int bar_code_read (unsigned char ∗rxbuf, int len)

  This API Function reads the barcode.

- int bar_code_close (void)

  This API Function closes the barcode.

- int tama_open (void)

  This API Function is used to open RFID device. Note: This function must be called before any rfid function.

- void tama_close (void)

  This API Function is used to close rfid device. Note: This function must be called before any rfid function.

- int tama_detectTarget (unsigned char Tg_Nos, unsigned char ∗Tg_Detected, unsigned char ∗Tg_-ID)

  This API Function is used to detect the targets of RFID Card reader.

- int tama_authenticate_mifare (unsigned char Tg_No, unsigned char Blk_Addr, unsigned char index, unsigned char Key_Type)

  This API Function authenticates a Tag's Block for writing/reading with the said key.

- int tama_authenticate_mifare_key (unsigned char Tg_No, unsigned char Blk_Addr, unsigned char ∗Key, unsigned char Key_Type)

  This API Function authenticates a Tag's Block for writing/reading with the said key.

- int tama_halt_mifare (unsigned char Tg_LID)

  This API Function halts (Deselects) a tag for access.

- int tama_read_target (unsigned char Blk_Addr, unsigned char ∗Data_In)

  This API Function reads the block data and stores it into passed address.

- int tama_write_target (unsigned char Blk_Addr, unsigned char ∗Data_Out, int Dlen)

  This API Function writes the data sent to the said block.

- int tama_rffieldon (void)

  This API Function is used to power ON RF field. Note: This function must be called after tama_open function.

- int tama_rffieldoff (void)

  This API Function is used to power OFF RF field. Note: This function must be called after tama_open function.

- int mscr_open (void)

  This API Function is used to open magswipe device. Note: This function must be called before any magswipe function.

- int mscr_pwr_enable (void)

  This API Function is used to power ON magswipe device.

- int mscr_pwr_disable (void)

  This API Function is used to power OFF magswipe device.

- int mscr_read (char ∗track_1_data, char ∗track_2_data, char ∗track_3_data)

  This API function reads the data of the both tracks from the card reader .

- int mscr_flush (void)

  This API Function clears the card reader device buffer returns SUCCESS on success, returns MAG_SW_-ERR if the device not opened.

- int mscr_getstate (void)

  This API Function is used to check wether the device is having valid data (both track1 and track2 data).

- int mscr_close (void)

  This API Function is to close magnetic swipe device.

- int SAM_Open (void)

  This function checks for the existence of the Sam card reader device and if found prepares it for further communication. Before attempting any operat ∗ ions related to Sam card reader device this function should be called.

- int SAM_Close ()

  This function closes the SAM device. After finishing all the operations related to SAM card reader this function should get called.

- int SAM_GetAtr (unsigned int ∗length, unsigned char ∗buffer)

  This API reads ATR from the card.

- int SAM_SendCommand (unsigned char ∗CAPDU_Data, unsigned int CLen, unsigned char ∗RAPDU_Data, int ∗RLen)

  This API is used to send all the EMV complaint CAPDUs to the card. The API call sends the command contained in CAPDU Data buffer and gives the response and ∗its length out through RAPDU Data and RLen variables respectively.

- int SAM_ColdReset (int selection)

  This API is to power up the card,after this call Vcc is applied as selected during card select API. After this call only we can do any command operations o ∗n card.

- int SAM_Select_S1 (int volt)

  This API is to select the SAM slot one. And send return value to SAM_ColdReset API.

- int SAM_Select_S2 (int volt)

  This API is to select the SAM slot two. And send return value to SAM_ColdReset API.

- int SAM_Poweroff (void)

  This API is used to Power off SAM module.

- int usb_pwr_enable (void)

  This API Function is used to power on usb-host device.

- int usb_pwr_disable (void)

  This API Function is used to power off usb-host device.

- int audio_pwr_enable (void)

  This API Function is used to power on audio device.

- int audio_pwr_disable (void)

  This API Function is used to power off audio device.

- int wifi_enable (void)

This API Function is used to enable wifi functionality.

- int wifi_disable (void)

  This API function is used to disable wifi functionality.

- int wifi_rfpower_on (void)

  This API function is used to power on the RF section of wifi.

- int wifi_rfpower_off (void)

  This API function is used to power off the RF section of wifi.

4.1.1 Detailed Description

GL14POS APIs and Definitions

Company Name: Linkwell Telesystems Pvt. Ltd (visiontek.co.in)

API Description: GL14POS LCD,KPD,RTC,Printer,Barcode,RFID,Magswipe,USB host,Audio,wi-Fi Device and SAM and IFD APIs.

Index