**User manual:**

**Library Name:**

p\_kiosk\_sdl

**Global Variables** (declared in header file):

const int SCREEN\_WIDTH

Purpose: global variable for screen width

const int SCREEN\_HEIGHT

Purpose: global variable for screen height

const int WINDOW\_WIDTH

Purpose: global variable for window width

const int WINDOW\_HEIGHT

Purpose: global variable for window height

const int PRINT\_SCREEN\_WIDTH

Purpose: global variable for print screen width

const int PRINT\_SCREEN\_HEIGHT

Purpose: global variable for print screen height

**Structs:**

p\_sdl\_data {

SDL\_Window\* window (Pointer to SDL window)

SDL\_Surface\* screen\_surface (Pointer to screen surface)

SDL\_Surface\* keypad\_surface (Pointer to keypad surface)

SDL\_Surface\* print\_screen\_surface (Pointer to receipt printer surface)

SDL\_Renderer\* renderer (Pointer to renderer struct handling rendering)

SDL\_Texture \*screen\_texture (texture of screen image)

SDL\_Texture \*keypad\_texture (texture of keypad image)

SDL\_Texture \*print\_screen\_texture (texture of receipt printer image)

int mouse\_cursor\_x (The x location of the mouse cursor)

int mouse\_cursor\_y (The y location of the mouse cursor)

sdl\_color color (The current color)

TTF\_Font \*text\_font (pointer to font)

int font\_size (size of font)

int print\_font\_size (size of receipt printer font)

int text\_line\_size (current line size of text rendered to screen)

int print\_text\_line\_size (current line size of text rendered to screen)

SDL\_Rect text\_space (position and area occupied by text)

SDL\_Rect print\_text\_space (position and area occupied by text)

SDL\_Rect keypad\_dest (position and area occupied by text)

SDL\_Rect screen\_dest (position and area occupied by text)

SDL\_Rect print\_screen\_dest (position and area occupied by text)

int text\_cursor\_x (x coordinate of text cursor position)

int text\_cursor\_y (y coordinate of text cursor position)

int print\_screen\_cursor\_x (x coordinate of receipt printer position)

int print\_screen\_cursor\_y (y coordinate of receipt printer position)

File \*receipt\_print (output file for receipt printer)

SDL\_Event \*mouse\_event (sdl event)

bool mouse\_down (checks if mouse is clicked)

}

**Functions:**

**Name:** p\_sdl\_new(void)

**Purpose:** Creates all necessary SDL items, (windows, surfaces, etc.), renders the base backgrounds and items, (buttons, etc.), and returns a struct.

**Returns:** A pointer to struct p\_sdl\_data\*

**Name:** p\_sdl\_close(p\_sdl\_data\*)

**Purpose:** Closes all necessary SDL items (windows, surfaces, etc.).

**Returns:** Integer status

**Name:** p\_sdl\_get\_mouse\_click(p\_sdl\_data\*)

**Purpose:** Returns the last selected value, if there is any, returns -1 if nothing is available.

**Returns:** -1 if nothing, binary value relative to area clicked

**Name:** p\_sdl\_clear\_screen(p\_sdl\_data\*)

**Purpose:** Clears the kiosk screen.

**Returns:** Integer status, 0 if fine

**Name:** p\_sdl\_reset\_values(p\_sdl\_data\*)

**Purpose:** Resets the Kiosk SDL data variables to their default values (values set in the init function)

**Returns:** Int status 0 if success, else error

**Name:** p\_sdl\_render\_string(p\_sdl\_data\*, char[ ] string)

**Purpose:** Renders the entire string beginning at the current cursor location. If the string will go off screen, nothing is displayed, and an error status is returned. If kiosk pointer to color is NULL/unset use {0,0,0} as default color.

**Returns:** Integer status, 0 if fine

**Name:** p\_sdl\_render\_char(p\_sdl\_data\*, char character)

**Purpose:** Renders the a character at the current cursor location, if the cursor will go off screen, nothing is displayed and an error status is returned.

**Returns:** Integer status, 0 if fine

**Name:** p\_sdl\_set\_cursor\_x(p\_sdl\_data\*, int x)

**Purpose:** Sets the cursor x location to value if valid, otherwise an error status is returned (1).

**Returns:** Integer status, 0 if valid, 1 if not

**Name:** p\_sdl\_set\_cursor\_y(p\_sdl\_data\*, int y)

**Purpose:** Sets the cursor y location to value if valid, otherwise an error status is returned (1).

**Returns:** Integer status, 0 if valid, 1 if not

**Name:** p\_sdl\_set\_color(p\_sdl\_data\*, int color)

**Purpose:** Sets the current color for text rendering and drawing.

**Returns:** Integer status, 0 if valid, 1 if not

**Name:** p\_sdl\_get\_text\_cursor\_x

**Purpose:** Gets current location of the text cursor on the X-axis.

**Returns:** Current X location of text cursor

**Name:** p\_sdl\_get\_text\_cursor\_y

**Purpose:** Gets current location of the text cursor on the Y-axis.

**Returns:** Current Y location of the text cursor

**Name:** p\_sdl\_get\_color

**Purpose:** Gets the current color.

**Returns:** RGB Color value, -1 if invalid

**Draw functions:**

**Name:** p\_sdl\_draw\_line(p\_sdl\_data\*, int start\_x, int start\_y, int end\_x, int end\_y)

**Purpose:** Draws a line from the the start x and y to the end x and y arguments, returns 1 for invalid inputs (off screen, etc.).

**Returns:** Integer status, 0 if okay, 1 if not

**Name:** p\_sdl\_draw\_rectangle(p\_sdl\_data\*, int x, int y, int height, int width, int dofill (0 or 1))

**Purpose:** Draws a rectangle onto the screen. Returns 1 for invalid inputs. 0 is unfilled, 1 is filled.

**Returns:** Integer status, 0 if okay, 1 if not

**Name:** p\_sdl\_draw\_pixel(p\_sdl\_data\*, int x, int y)

**Purpose:** Draws a dot at the defined x and y location arguments

**Returns:** Integer status, 0 if okay, 1 if not

**Name:** p\_sdl\_draw\_circle(p\_sdl\_data\*, int x, int y, int radius, int dofill (0 or 1))

**Purpose:** Draws a circle at the defined location arguments, returns 1 for invalid inputs. 0 is unfilled, 1 is filled.

**Returns:** Integer status, 0 if okay, 1 if not

**Receipt printer functions:**

**Name**: p\_sdl\_set\_receipt\_cursor\_x (p\_sdl\_data\*, int x)

**Purpose**: Sets the receipt cursor x location to value if valid, otherwise an error status is returned (1).

**Returns**: Integer status, 0 if valid, 1 if not

**Name**: p\_sdl\_set\_receipt\_cursor\_y (p\_sdl\_data\*, int y)

**Purpose:** Sets the receipt cursor x location to value if valid, otherwise an error status is returned (1).

**Returns:** Integer status, 0 if valid, 1 if not

**Name:** p\_sdl\_get\_receipt\_cursor\_x(p\_sdl\_data\*)

**Purpose:** Gets the receipt cursor x location.

**Returns:** Integer status

**Name:** p\_sdl\_get\_receipt\_cursor\_y(p\_sdl\_data\*)

**Purpose:** Gets the receipt cursor y location.

**Returns:** Integer status

**Name:** p\_sdl\_receipt\_render\_char(p\_sdl\_data\*, char character)

**Purpose:** Renders a character at the current receipt location, if the cursor will go off screen, nothing is displayed and an error status is returned

**Returns:** Integer status, 0 if fine

**Name:** int p\_sdl\_receipt\_printer\_new\_line(p\_sdl\_data\*)

**Purpose:** Clears current line and resets the X and Y position to original position

**Returns:** Integer status, 0 if fine

**Simple Graphics U.0 Functions:**

**Name:** p\_sdl\_draw\_screen(p\_sdl\_data\*)

**Purpose:** Determine if the mouse is in the screen area first. Then draws pixels while the user holds the mouse button down.

**Returns:** 0 if the mouse event is equal to quit, 1 if the function is successful.

|  |  |  |
| --- | --- | --- |
| p\_sdl\_data.color int value | Color | RGB value |
| 0 | Black | 0, 0, 0 |
| 1 | Blue | 0, 0, 255 |
| 2 | Green | 0, 255, 0 |
| 3 | Red | 255, 0, 0 |
| 4 | Cyan | 0, 255, 255 |
| 5 | Magenta | 255, 0, 255 |
| 6 | Yellow | 255, 255, 0 |
| 7 | Deep Sky Blue | 0, 191, 255 |
| 8 | Deep Pink | 255, 20, 147 |
| 9 | Medium Spring Green | 0, 250, 154 |
| 10 | Purple | 128, 0, 128 |