

## Overview

The `KDU_MENU.LIB` library provides the programmer with a set of functions that allow an efficient and easy-to-use menu system for use by the operator. The programmer can list a set of action items for the operator to choose from. Menus can be nested to call other menus or perform a desired task.

All the menu functions are nonblocking.

## Display

The LCD will show the title of the menu, with a border, and display two menu items at a time:



Using the keypad, the operator can move the highlight with the **UP/DOWN** arrow keys. Press the **ENTER** key to select the highlighted item.

## Keypad Operation

The LCD/keypad module has a 7-key keypad. The keys have the following response:

- LEFT ARROW**—not used.
- UP ARROW**—moves selection up 1 menu item.
- DOWN ARROW**—moves selection down 1 menu item.
- RIGHT ARROW**—not used.
- MINUS KEY**—moves selection down 2 menu items.
- PLUS KEY**—moves selection up 2 menu items.
- ENTER KEY**—selects the item highlighted.

## Menu Function Listing

```
int kduMenuInit(kduMenu *menu, char **menu_options,  
char* title);
```

Initialization function for a particular menu. Each menu will have its own **kduMenuInit** call.

### PARAMETERS

**menu** is a pointer to the **kduMenu** structure assigned.

**menu\_options** is a pointer to the list of menu items associated with the menu.

**title** is the title that will be displayed when the menu is displayed.

### RETURN VALUE

1 when completed.

```
int kduDisplayMenu (kduMenu *mPtr, int *state, int  
x, int y);
```

This function displays the menu pointed to by **mPtr**.

### PARAMETERS

**mPtr** is a pointer to the **kduMenu** structure to display.

**state** is a pointer to the state of the menu to display

**MENU\_INIT** will display the menu with the first item highlighted.

**MENU\_REFRESH** will display the menu with the last item selected highlighted.

**x** is the *x* coordinate of the starting horizontal pixel.

**y** is the *y* coordinate of the starting vertical pixel.

### RETURN VALUE

0 if waiting for an item to be selected, or the item number selected when the **ENTER** key is pressed.

```
void kduMenuClear(kduMenu *mPtr);
```

Clears the menu pointed to by **mPtr** from the LCD.

### PARAMETER

**mPtr** is a pointer to the **kduMenu** structure assigned.

### RETURN VALUE

None.

```
int kduMenuBorder(kduMenu *mPtr);
```

Places a border around the menu. This is normally done by `kduDisplayMenu`.

#### PARAMETER

`mPtr` is a pointer to the `kduMenu` structure assigned.

#### RETURN VALUE

1 when completed.

## Data-Entry Functions

The menu library contains a set of data-entry functions that enable an operator to enter data such as parameters, time and date, IP addresses, phone numbers, passwords, etc. Data-entry functions use the keypad differently than the menu system, so please look at the appropriate data-entry functions for their keypad instructions. See the sample program `KDU_DATA_ENTRY.C` for a better understanding of their use.

```
int kduEnterLong(long *Value, char *Heading,
int digits);
```

Allows for data entry of long integer values.



#### PARAMETERS

`Value` is a pointer to location of data to be stored.

`Heading` is a pointer to the heading to be displayed.

`digits` is the maximum number of digits to allow for data entry.

#### RETURN VALUE

1 when completed, 0 if pending.

#### KEYPAD INSTRUCTIONS

**LEFT ARROW**—moves selection 1 to the left.

**UP ARROW**—increment value selected by 1.

**DOWN ARROW**—decrement value selected by 1.

**RIGHT ARROW**—moves selection 1 to the right.

**MINUS KEY**—no action.

**PLUS KEY**—no action.

**ENTER KEY**—Select the value entered, and exit the entry screen.

```
int kduEnterFloat(float *Value, char *Heading,  
int digits);
```

Allows for data entry of float values.



#### PARAMETERS

**Value** is a pointer to the location of data to be stored.

**Heading** is a pointer to the heading to be displayed.

**digits** is the maximum number of digits to allow for data entry.

#### RETURN VALUE

1 when completed, 0 if pending.

#### KEYPAD INSTRUCTIONS:

**LEFT ARROW**—moves selection 1 to the left.

**UP ARROW**—increment value selected by 1.

**DOWN ARROW**—decrement value selected by 1.

**RIGHT ARROW**—moves selection 1 to the right.

**MINUS KEY**—no action.

**PLUS KEY**—no action.

**ENTER KEY**—Select the value entered, and exit the entry screen.

```
int kduEnterString(char *String, int digits);
```

Allows for data entry of string variables.



#### PARAMETERS

**String** is a pointer to location of data to be stored in.

**digits** is the maximum number of digits to allow for data entry.

#### RETURN VALUE

1 when completed, 0 if pending.

#### KEYPAD INSTRUCTIONS

**LEFT ARROW**—moves selection 1 to left.

**UP ARROW**—moves selection 1 up.

**DOWN ARROW**—moves selection 1 down.

**RIGHT ARROW**—moves selection 1 to right.

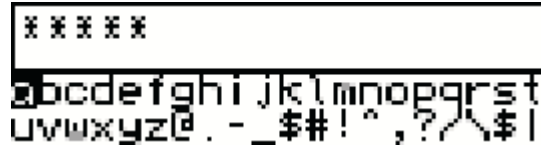
**MINUS KEY**—delete the last item select.

**PLUS KEY**—accepts the item selected.

**ENTER KEY**—select the value entered, only asterisks will be displayed.

```
int kduEnterPassword(char *String, int digits);
```

Allows for data entry of passwords. As a string item is selected, an asterisk will appear instead of the character selected.



```
*****
abcdefghijklmnopqrst
vwxyz@.-_!^,?/\$|
```

#### PARAMETERS

**String** is a pointer to location of data to be stored in.

**digits** is the maximum number of digits to allow for data entry.

#### RETURN VALUE

1 when completed, 0 if pending.

#### KEYPAD INSTRUCTIONS

**LEFT ARROW**—moves selection 1 to left.

**UP ARROW**—moves selection 1 up.

**DOWN ARROW**—moves selection 1 down.

**RIGHT ARROW**—moves selection 1 to right.

**MINUS KEY**—delete the last item select.

**PLUS KEY**—accepts the item selected.

**ENTER KEY**—select the value entered, only asterisks will be displayed.

```
int kduEnterIpAddress(long *Address, char *Heading);
```

Allows for data entry of IP addresses.



#### PARAMETERS

**Address** is a pointer to location of data to be stored in.

**Heading** is a pointer to the heading to display in the heading box.

#### RETURN VALUE

1 when completed, 0 if pending.

#### KEYPAD INSTRUCTIONS

**LEFT ARROW**—moves selection 1 to the left.

**UP ARROW**—increment value selected by 1.

**DOWN ARROW**—decrement value selected by 1.

**RIGHT ARROW**—moves selection 1 to the right.


**MINUS KEY**—no action.

**PLUS KEY**—no action.

**ENTER KEY**—Select the value entered, and exit the entry screen.

```
int kduEnterPhoneNumber(char *PhoneString,  
char *Heading);
```

Allows for data entry of 11-digit phone numbers.



```
Enter Phone #  
1-(530)-757-373█
```

#### PARAMETERS

**Address** is a pointer to location of data to be stored in.

**Heading** is a pointer to the heading to display in the heading box.

#### RETURN VALUE

1 when completed, 0 if pending.

#### KEYPAD INSTRUCTIONS

**LEFT ARROW**—moves selection 1 to the left.

**UP ARROW**—increment value selected by 1.

**DOWN ARROW**—decrement value selected by 1.

**RIGHT ARROW**—moves selection 1 to the right.

**MINUS KEY**—no action.

**PLUS KEY**—no action.

**ENTER KEY**—Select the value entered, and exit the entry screen.

```
int kduEnterTimeDate(struct tm *Time,  
    char *Heading);
```

Allows for data entry of time and date.



#### PARAMETERS

**Time** is a pointer to location of time/date data to be stored in.

**Heading** is a pointer to the heading to display in the heading box.

#### RETURN VALUE

1 when completed, 0 if pending.

#### KEYPAD INSTRUCTIONS

**LEFT ARROW**—moves selection 1 to the left.

**UP ARROW**—increment value selected by 1.

**DOWN ARROW**—decrement value selected by 1.

**RIGHT ARROW**—moves selection 1 to the right.

**MINUS KEY**—no action.

**PLUS KEY**—no action.

**ENTER KEY**—Select the value entered, and exit the entry screen.

#### Z-World, Inc.

2900 Spafford Street  
Davis, California 95616-6800  
USA

Telephone: (530) 757-3737  
Fax: (530) 757-3792

[www.zworld.com](http://www.zworld.com)

#### Rabbit Semiconductor

2932 Spafford Street  
Davis, California 95616-6800  
USA

Telephone: (530) 757-8400  
Fax: (530) 757-8402

[www.rabbitsemiconductor.com](http://www.rabbitsemiconductor.com)