Electro test UI

1

Generated by Doxygen 1.8.11

Contents

Chapter 1

File Index

1.1 File List

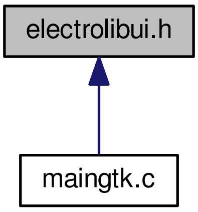
Here is a list of all documented files with brief descriptions:

Chapter 2

File Documentation

2.1 electrolibui.h File Reference

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



Functions

- void printDebugText (const char ∗text,...)

  Central function for printing debug information.
- void addInputSection (char ∗text, GtkGrid ∗grid, void(∗entryChangedFuncPtr)(GtkEditable ∗editable, gpointer), int col, int row, int ∗index)

  Add an input section with label and text input.
- void addInputSections (int numberOfComponents, int oldNumberOfComponents, GtkGrid ∗mainGrid, void(∗entryChangedFuncPtr)(GtkEditable ∗editable, gpointer))

  Adds a set of input sections, typicaly when it's needed to present a list of values or similar.
- void addOutputLabel (GtkGrid ∗grid, char ∗text, int col, int row, int width)

  Adds an output label that is used to communicate information to the end user.
- void addCheckbox (char ∗text, GtkGrid ∗grid, void(∗entryChangedFuncPtr)(GtkCheckButton ∗editable, gpointer), int col, int row)

  Adds an check box with an including text that describes what the check box does.
- void addDropDownList (int numberOfElements, GtkGrid ∗grid, void(∗entryChangedFuncPtr)(GtkComboBox ∗editable, gpointer), int col, int row)

Adds a drop down list generating input items up to the numberOfElements parameter.

- GtkGrid ∗ createWindowAndMainGrid (int argc, char ∗argv[ ], void(∗destroyFuncPtr)(GtkWidget ∗widget, gpointer), void(∗deleteEventFuncPtr)(GtkWidget ∗widget, GdkEvent ∗event, gpointer data))

  Creates a window and a corresponding grid, the grid is returned for further ui element creation.
- void addButton (char ∗text, GtkGrid ∗grid, void(∗buttonFuncPtr)(GtkButton ∗button, gpointer), int col, int row, int width)

  Adds a button with corresponding text.

2.1.1 Function Documentation

2.1.1.1 void addButton (char ∗ *text*, GtkGrid ∗ *grid*, void(∗)(GtkButton ∗button, gpointer) *buttonFuncPtr*, int *col*, int *row*, int *width*)

Adds a button with corresponding text.

Add a button with the corresponding text, click event and position in the grid

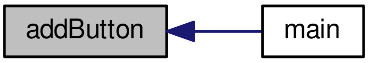
Parameters

text	The text on the button
grid	The grid where the button should be generated
entryChangedFucPtr	The function that handle the event when a user clicks the button
col	The column position where the output label should be generated
row	The row number where the output label should be generated
width	How many columns the button should span

Returns

void

Here is the caller graph for this function:



2.1.1.2 void addCheckbox (char ∗ *text*, GtkGrid ∗ *grid*, void(∗)(GtkCheckButton ∗editable, gpointer) *entryChangedFuncPtr*, int *col*, int *row*)

Adds an check box with an including text that describes what the check box does.

This function generates a check box and a descriptive text that says what the check box does

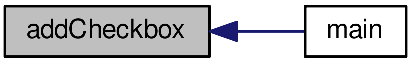
Parameters

text	The explanation of what the check box does
grid	The grid where the output section should be generated
entryChangedFucPtr	The function that handle the event when a user checks or un-checks the check box
col	The column position where the output label should be generated
row	The row number where the output label should be generated

Returns

void

Here is the caller graph for this function:



2.1.1.3 void addDropDownList (int *numberOfElements,* GtkGrid ∗ *grid,* void(∗)(GtkComboBox ∗editable, gpointer) *entryChangedFuncPtr,* int *col,* int *row*)

Adds a drop down list generating input items up to the numberOfElements parameter.

Generates a drop down list with integer numbers as elements. Items are generated from 1 to numberOfElements

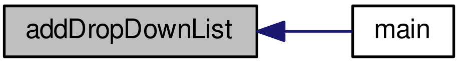
Parameters

grid	The grid where the output section should be generated
entryChangedFucPtr	The function that handle the event when a user selects an item in the list
col	The column position where the output label should be generated
row	The row number where the output label should be generated

Returns

void

Here is the caller graph for this function:



2.1.1.4 void addInputSection (char ∗ *text,* GtkGrid ∗ *grid,* void(∗)(GtkEditable ∗editable, gpointer) *entryChangedFuncPtr,* int *col,* int *row,* int ∗ *index*)

Add an input section with label and text input.

Generates a pair of a label and text input. The label prints out what to fill into the text box

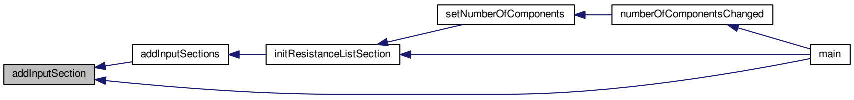
Parameters

text	The text to use in the label that describes what to fill into the text input field
grid	A GtkGrid object that will be the parent grid of the input section
entyChangedFuncPtr	The event function that should handle every time a user types in a character in the text box.
col	The column number where the input section should be placed
row	The row number where the input section should be placed
index	The id for the input section, this can be used for identification in the event function provided in the entryFunctionPtr

Returns

void

Here is the caller graph for this function:



2.1.1.5 void addInputSections (int *numberOfComponents,* int *oldNumberOfComponents,* GtkGrid ∗ *mainGrid,* void(∗)(GtkEditable ∗editable, gpointer) *entryChangedFuncPtr*)

Adds a set of input sections, typicaly when it's needed to present a list of values or similar.

Generates a set of input section, typically when it's needed to bind an array of values to ui input

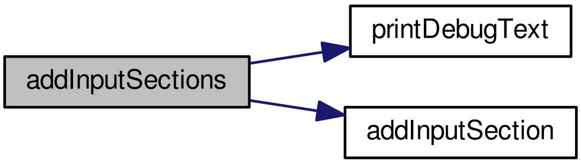
Parameters

numberOfComponents	The new number of components to be used
oldNumberOfComponents	The number of components used since before, this was the value set before it was changed to numberOfComponents
mainGrid	The grid to attach the input sections to
entryChangedFuncPtr	The function that should handle the input changes. The index is the identifier to be used in the event function to determine what array position to update etc.

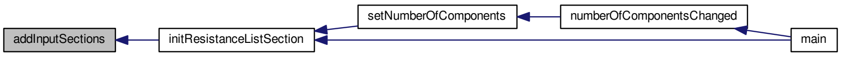
Returns

void

Here is the call graph for this function:



Here is the caller graph for this function:



2.1.1.6 void addOutputLabel (GtkGrid ∗ *grid,* char ∗ *text,* int *col,* int *row,* int *width*)

Adds an output label that is used to communicate information to the end user.

This function generates an output label that displays the text with bold text

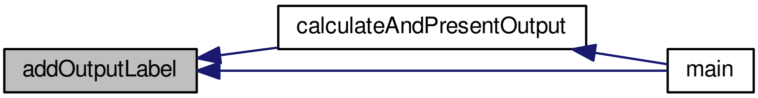
Parameters

grid	The grid where the output section should be generated
text	The text to print out
col	The column position where the output label should be generated
row	The row number where the output label should be generated
width	How many columns the output section should span

Returns

void

Here is the caller graph for this function:



2.1.1.7 GtkGrid∗ createWindowAndMainGrid (int *argc,* char ∗ *argv[ ],* void(∗)(GtkWidget ∗widget, gpointer) *destroyFuncPtr,* void(∗)(GtkWidget ∗widget, GdkEvent ∗event, gpointer data) *deleteEventFuncPtr*)

Creates a window and a corresponding grid, the grid is returned for further ui element creation.

Generates a window and a grid, the grid is the boilerplate for further ui elements to be added. This is the bootstrap method to wire up the main GTK functionality

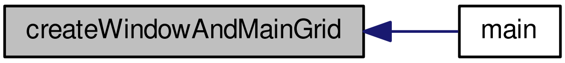
Parameters

argc	The number of arguments, this should be taken from the command arguments in main(...)
argv[ ]	The arguments, this should be taken from the command arguments in main(...)
destroyFuncPtr	The function that handle the kills the application
deleteEventFuncPtr	The function that is called when the window is closed

Returns

GtkGrid∗ Returns the grid that should be used when adding new ui elements.

Here is the caller graph for this function:



2.1.1.8 void printDebugText (const char ∗ *text,* ...)

Central function for printing debug information.

Central function for printing debug information

Parameters

text	The text to print
...	Send in any number of paramters, similar as formating for printf

Returns

void

Central function for printing debug information.

A wrapper around printf that simplifies printing out test result text in different colours.

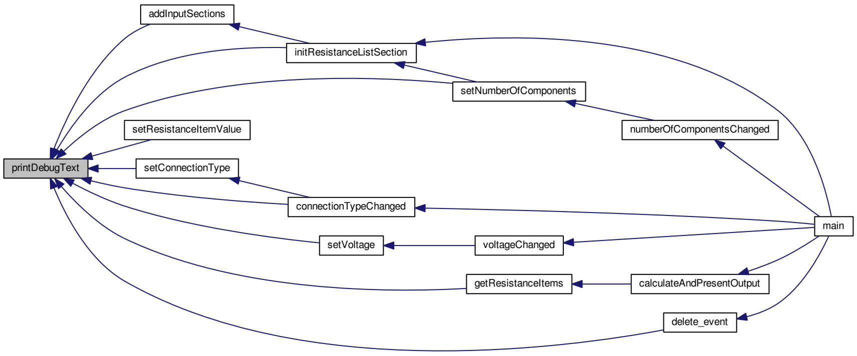
Parameters

text	The text message to print out
args	The arguments for the text output, it's a va_list so provide it in the same way as for a printf call

Returns

void

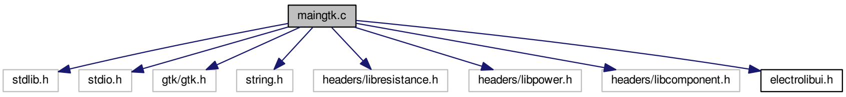
Here is the caller graph for this function:



2.2 maingtk.c File Reference

```
#include <stdlib.h>
#include <stdio.h>
#include <gtk/gtk.h>
#include <string.h>
#include "headers/libresistance.h"
#include "headers/libpower.h"
#include "headers/libcomponent.h"
#include "electrolibui.h"
```

Include dependency graph for maingtk.c:



Functions

- static void initResistanceListSection (int currentNumberOfComponents, int oldNumberOfComponents, char ∗∗∗resistanceList, GtkGrid ∗mainGrid, void(∗entryChangedFuncPtr)(GtkEditable ∗editable, gpointer))

  Initializes the resistance array and wires it up with the UI components.
- static void setResistanceItemValue (int index, char ∗value)

  Sets the resistance value for an index.
- static void **resistanceListItemChanged** (GtkEditable ∗editable, gpointer user_data)
- static void setConnectionType (char connectionType)

  Sets the connection type.
- static char getConnectionType ()

  Gets the connection type.
- static void setVoltage (char ∗value)

  Sets voltage value.
- static float getVoltage ()

  Gets the voltage.
- static int getNumberOfComponents ()

  Gets the number of components.
- static void setNumberOfComponents (char ∗value)

  Sets the number of components.
- static float ∗ getResistanceItems ()

  Gets the resistance items as a float array.
- static void voltageChanged (GtkEditable ∗editable, gpointer user_data)

  Event method that is fired when voltage is changed.
- static void connectionTypeChanged (GtkCheckButton ∗checkButton, gpointer user_data)

  Event method that is fired when connection type is changed.
- static void numberOfComponentsChanged (GtkComboBox ∗comboBox, gpointer user_data)

  Event method that is fired when number of components are changed.
- static gboolean delete_event (GtkWidget ∗widget, GdkEvent ∗event, gpointer data)

  Event method that is fired when window is closed.
- static void destroy (GtkWidget ∗widget, gpointer data)

  Event method that is fired when the application is being closed.
- static void calculateAndPresentOutput ()

  Event method that is fired when it's time to calculate the output of the input.
- int main (int argc, char ∗argv[ ])

  Starts the whole application and calls the electrolibui to present the UI in GTK.

Variables

- static GtkGrid ∗ **_mainGrid**
- static char ∗ **_voltage** = "0"
- static char **_connectionType** = 'S'
- static char ∗ **_numberOfComponents** = "1"
- static char ∗∗ **_resistanceValueList** = NULL

2.2.1 Function Documentation

2.2.1.1 static void calculateAndPresentOutput ( ) [static]

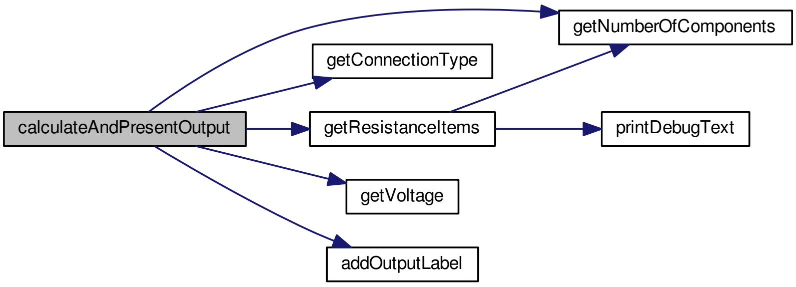
Event method that is fired when it's time to calculate the output of the input.

Event method that fires when the button is clicked. This calculates the result of the electrolib functionality.

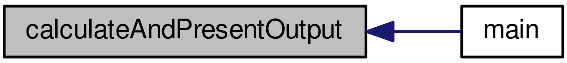
Returns

void

Here is the call graph for this function:



Here is the caller graph for this function:



2.2.1.2 static void connectionTypeChanged (GtkCheckButton ∗ *checkButton,* gpointer *user_data*) [static]

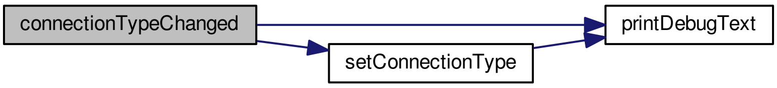
Event method that is fired when connection type is changed.

Event method that fires when connection type checkbox is changed. It updates the model data with the set↵ ConnectionType method.

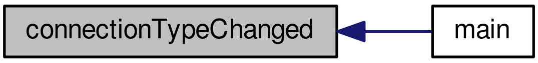
Returns

void

Here is the call graph for this function:



Here is the caller graph for this function:



2.2.1.3 static gboolean delete_event (GtkWidget ∗ *widget,* GdkEvent ∗ *event,* gpointer *data*) [static]

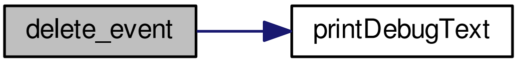
Event method that is fired when window is closed.

Event method that fires when the window is closed. It returns FALSE if it should move on to run destroy, e.g. close the application.

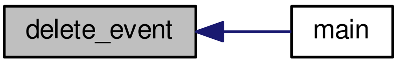
Returns

void

Here is the call graph for this function:



Here is the caller graph for this function:



2.2.1.4 static void destroy (GtkWidget ∗ *widget,* gpointer *data*) [static]

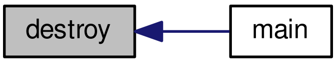
Event method that is fired when the application is being closed.

Event method that fires when the application is about to be closed.

Returns

void

Here is the caller graph for this function:



2.2.1.5 static char getConnectionType ( ) [static]

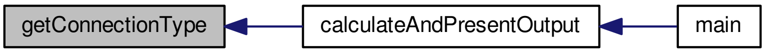
Gets the connection type.

Gets the connection type

Returns

char

Here is the caller graph for this function:



2.2.1.6 static int getNumberOfComponents ( ) [static]

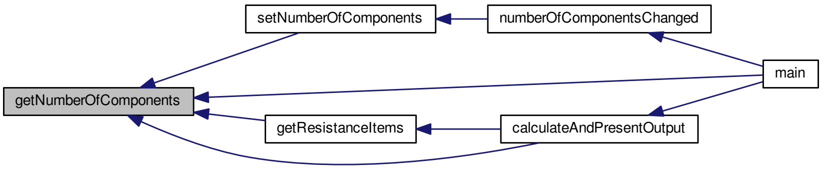
Gets the number of components.

Gets the number of components

Returns

int

Here is the caller graph for this function:



2.2.1.7 static float∗ getResistanceItems ( ) [static]

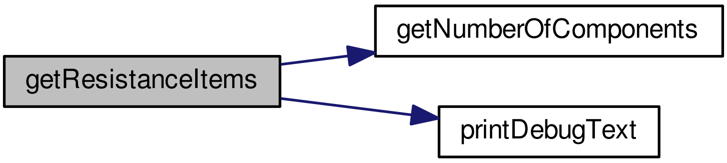
Gets the resistance items as a float array.

Get the current array of resistance values as float. This function converts to float on the fly from a char pointer array.

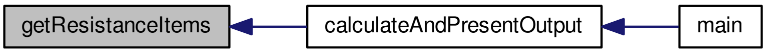
Returns

float∗

Here is the call graph for this function:



Here is the caller graph for this function:



2.2.1.8 static float getVoltage ( ) [static]

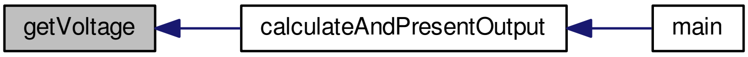
Gets the voltage.

Gets the voltage. The voltage is stored as char∗ and converted on the fly by this method

Returns

float

Here is the caller graph for this function:



2.2.1.9 static void initResistanceListSection (int *currentNumberOfComponents*, int *oldNumberOfComponents*, char ∗∗∗ *resistanceList*, GtkGrid ∗ *mainGrid*, void(∗)(GtkEditable ∗editable, gpointer) *entryChangedFuncPtr*) [static]

Initializes the resistance array and wires it up with the UI components.

Initializes the resistance array and wires it up with the UI components. This function also handles the change when different number of components are selected.

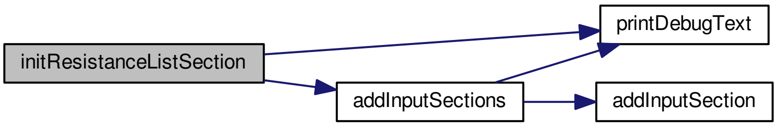
Parameters

currentNumberOfComponents	The number of components selected
oldNumberOfComponents	The number of components selected before
resistanceList	The array with the resistance value pointers. The function handles all memory allocation and releasing of memory.
mainGrid	The grid where this should be added
entryChangedFuncPtr	The function to handle the event when a resistance value is changed

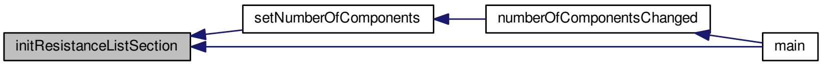
Returns

void

Here is the call graph for this function:



Here is the caller graph for this function:



2.2.1.10 int main (int *argc,* char ∗ *argv[ ]*)

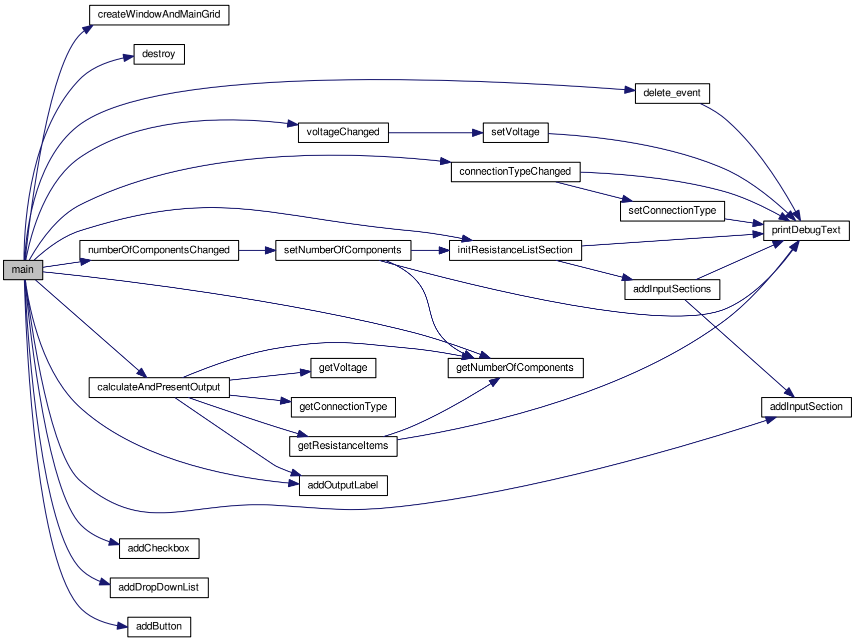
Starts the whole application and calls the electrolibui to present the UI in GTK.

Starts the whole application and calls the electrolibui to present the UI in GTK

Returns

void

Here is the call graph for this function:



2.2.1.11 static void numberOfComponentsChanged (GtkComboBox ∗ *comboBox,* gpointer *user_data*) [static]

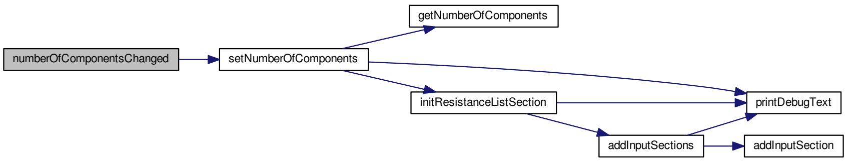
Event method that is fired when number of components are changed.

Event method that fires when the number of component drop down list is changed. It updates the model data with the setNumberOfComponents method.

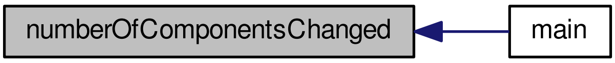
Returns

void

Here is the call graph for this function:



Here is the caller graph for this function:



2.2.1.12 static void setConnectionType (char *connectionType*) [static]

Sets the connection type.

Sets the connection type, can be either S or P

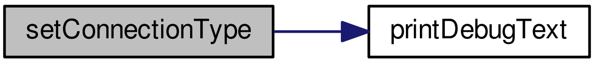
Parameters

connectionType	Can either be S or P

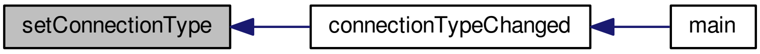
Returns

void

Here is the call graph for this function:



Here is the caller graph for this function:



2.2.1.13 static void setNumberOfComponents (char * *value*) [static]

Sets the number of components.

Sets the number of components

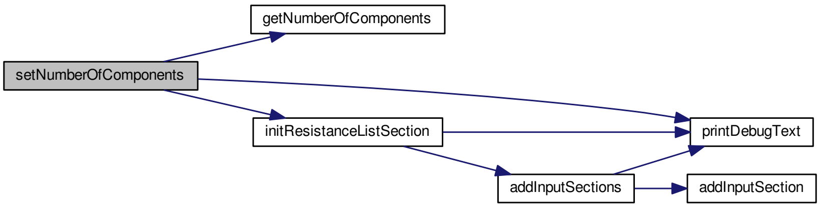
Parameters

value	The number of components

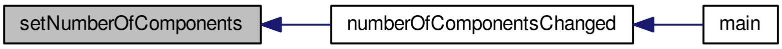
Returns

void

Here is the call graph for this function:



Here is the caller graph for this function:



2.2.1.14 static void setResistanceItemValue (int *index*, char ∗ *value*) [static]

Sets the resistance value for an index.

Sets the resistance value for an index

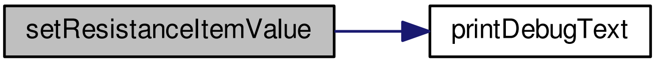
Parameters

index	The index to update
value	The value to set

Returns

void

Here is the call graph for this function:



2.2.1.15 static void setVoltage (char ∗ *value*) [static]

Sets voltage value.

Sets the value of the voltage

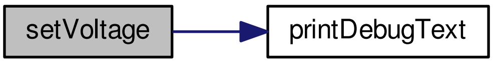
Parameters

value	The value of the voltage

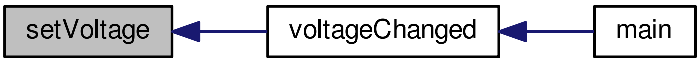
Returns

void

Here is the call graph for this function:



Here is the caller graph for this function:



2.2.1.16 static void voltageChanged (GtkEditable ∗ *editable,* gpointer *user_data*) [static]

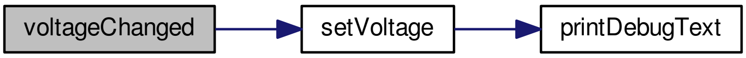
Event method that is fired when voltage is changed.

Event method that fires when voltage input textbox is changed. It updates the model data with the setVoltage method.

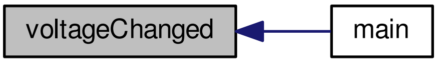
Returns

void

Here is the call graph for this function:



Here is the caller graph for this function:



Index