Using hardware keys with S60 and Maemo Platform

Archived: This article is archived because it is not considered relevant for third-party developers creating commercial solutions today. If you think this article is still relevant, let us know by adding the template {{ReviewForRemovalFromArchive|user=~~~~|*write your reason here*}}.

The article is believed to be still valid for the original topic scope.

Comparison

Both S60 Platform and Maemo Platform devices have a number of hardware keys for developers to use. The number of keys available varies between different devices. Both platforms have devices with full QWERTY keyboard. For example the E90 Communicator and the Nokia N810 Internet Tablet.

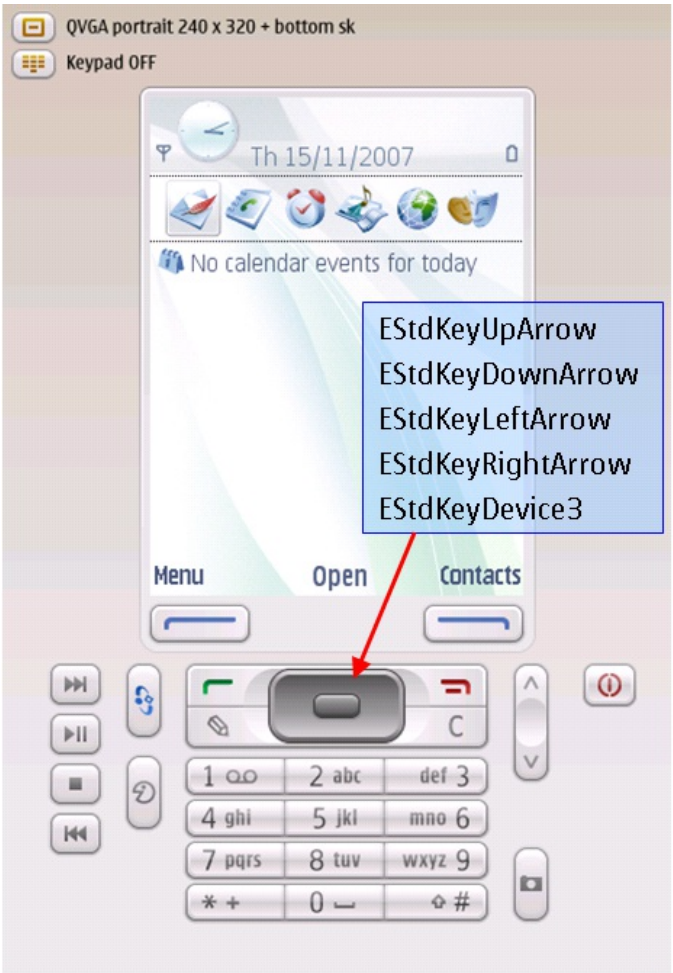
Nokia E90 Communicator

Nokia N810 Internet Tablet

Comparing S60 and Maemo Platforms

S60 Platform

(C) Copyright Nokia 2013. All rights reserved.



\epoc32\include\e32keys.h

```
enum TStdScanCode
 {
...
 EStdKeyLeftArrow=0x0e,                /**< Scan code for Left arrow key.*/
 EStdKeyRightArrow=0x0f,               /**< Scan code for Right arrow key.*/
 EStdKeyUpArrow=0x10,                  /**< Scan code for Up arrow key.*/
 EStdKeyDownArrow=0x11,                /**< Scan code for Down arrow key.*/
...
 };
```

```
// ----------------------------------------------------
// CAknExEditorAppUi::HandleKeyEventL
// Handles key events.
//
// ----------------------------------------------------
//
TKeyResponse CUTFViewerAppUi::HandleKeyEventL(
    const TKeyEvent& aKeyEvent,TEventCode aType )
    {
    if ( aType == EEventKey)
     {
     switch ( aKeyEvent.iScanCode )
     {
        case EStdKeyUpArrow:
...
     }
    return EKeyWasNotConsumed;
    }
```

Maemo Platform

(C) Copyright Nokia 2013. All rights reserved.

```
typedef struct _AppData AppData;

struct _AppData
{
    HildonProgram *program; /* handle to application */
    HildonWindow *window; /* handle to app's window */
    osso_context_t *osso; /* handle to osso */
};
```

```
/* Creates and initialises a main_view */
MainView* interface_main_view_new( AppData *data )
{
    /* Zero memory with g_new0 */
    MainView* result = g_new0( MainView, 1 );
    ...
    /* Add hardware button listener to application */
    g_signal_connect(G_OBJECT(result->data->window),
        "key_press_event", G_CALLBACK(callback_key_press), result);
    ...
}
```

```
/* Callback for hardware keys */
gboolean callback_key_press(GtkWidget * widget, GdkEventKey * event, gpointer data)
{
    switch (event->keyval) {

    case HILDON_HARDKEY_UP:
        callback_up( NULL, data );
        return TRUE;

    case HILDON_HARDKEY_DOWN:
        callback_down( NULL, data );
        return TRUE;

    case HILDON_HARDKEY_LEFT:
        callback_left( NULL, data );
        return TRUE;

    case HILDON_HARDKEY_RIGHT:
        callback_right( NULL, data );
        return TRUE;

    case HILDON_HARDKEY_SELECT:
        callback_reset( NULL, data );
        return TRUE;

    case HILDON_HARDKEY_FULLSCREEN:
        callback_fullscreen( NULL, data );
        return TRUE;

    case HILDON_HARDKEY_INCREASE:
        return FALSE;

    case HILDON_HARDKEY_DECREASE :
        return FALSE;

    case HILDON_HARDKEY_ESC :
        return FALSE;
```


(C) Copyright Nokia 2013. All rights reserved.

```
    }
    return FALSE;
}
```


(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.