

The recreated...

sinclair ZX Spectrum



ZX Spectrum Keyboard Technical Document (prerelease 23/01/2015).

The ZX Spectrum Keyboard has two layers, the Game layer and the QWERTY layer.
(Do QWERTY layer later).

1. The Game layer.

This uses a non-standard protocol to communicate, whereby it uses a different keycode to indicate that the key has been released.

For example, if you press the 1 key.

Instead of the usual:

Keydown 1.

Then you release the 1 key:

Keyup 1.

With the Game layer:

You press the 1 key:

Keydown a.

Keyup a.

You release the 1 key:

Keydown b.

Keyup b.

- There are 40 keys overall, each with a different keydown and keyup code.
- A couple of the keys are affected by the language encoding UK/US/etc. of the operating system. (M and Symbol Shift).
- There is no auto-repeat.
- Caps Lock has no effect on this layer.

Example Code:

```
zxkeyboard.c
zxkeyboard.h
WinMain.cpp
resource.h
resource.rc
```

A Microsoft Visual Studio 2015 project file is included (if you're using an older version of Visual Studio, it's just an empty Win32 project with the 5 files included).

The portable files are zxkeyboard.c and zxkeyboard.h.

```
int ZXKeyboard_ASCIIKeyPress(STRUCT ZXKEYBOARD *zxkb, ZXWCHAR key); // Call this to report a
keypress to the routines.
int *ZXKeyboard_GetMap(STRUCT ZXKEYBOARD *zxkb); // Returns array of
keypresses (1 = pressed, 0 = unpressed).
ZXWCHAR *ZXKeyboard_GetMapVisual(STRUCT ZXKEYBOARD *zxkb); // Get string
representation of current keyboard state (for demo purposes).
void ZXKeyboard_Reset(STRUCT ZXKEYBOARD *zxkb); // Reset internal
structures.
void ZXKeyboard_Init(STRUCT ZXKEYBOARD *zxkb, int iZXKBMode); // Initialise keyboard
routines.
```

Basically...

```
#include "zxkeyboard.h"
struct ZXKEYBOARD gZXKB;
```

... then initialise with one of these...

```
ZXKeyboard_Init(&gZXKB, eZXKEYBOARD_MODE_APPLE);
ZXKeyboard_Init(&gZXKB, eZXKEYBOARD_MODE_PC_UK);
ZXKeyboard_Init(&gZXKB, eZXKEYBOARD_MODE_PC_US);
ZXKeyboard_Init(&gZXKB, eZXKEYBOARD_MODE_ANDROID_US);
```

Report a keypress with:

```
ZXKeyboard_ASCIIKeyPress(&gZXKB, wCharacterCode);
```

Read the keymap like this:

```
int *pKeyMap = ZXKeyboard_GetMap(&gZXKB);
```

"pKeyMap" points to an array of 40 ints, which are set to 1 if pressed or 0 if not pressed. You can access each key by using one of the enumerated offsets into the array, for example:

```
if(pKeyMap[eZXKEYBOARD_CAPS_SHIFT])
{
    // Caps Shift is currently pressed.
}
```

(All of the enumerated keys are listed in zxkeyboard.h)

And whenever you want to reset the keymap:

```
ZXKeyboard_Reset(&gZXKB);
```

Key up/down list

Windows and Android key columns are only filled if the keycodes are different for them.

| ZX Spectrum | Apple/IOS Down | Apple/IOS Up | Windows/ Android US Down | Windows/ Android US Up | Windows/ Android UK Down | Windows/ Android UK Up | Key down names | Key up names |
|----------------|-------------------|-----------------|--------------------------------|------------------------------|--------------------------------|------------------------------|-------------------|-----------------|
| I | a | b | | | | | | |
| 2 | c | d | | | | | | |
| 3 | e | f | | | | | | |
| 4 | g | h | | | | | | |
| 5 | i | j | | | | | | |
| 6 | k | l | | | | | | |
| 7 | m | n | | | | | | |
| 8 | o | p | | | | | | |
| 9 | q | r | | | | | | |
| 0 | s | t | | | | | | |
| Q | u | v | | | | | | |
| W | w | x | | | | | | |
| E | y | z | | | | | | |
| R | A | B | | | | | | |
| T | C | D | | | | | | |
| Y | E | F | | | | | | |
| U | G | H | | | | | | |
| I | I | J | | | | | | |
| O | K | L | | | | | | |
| P | M | N | | | | | | |
| A | O | P | | | | | | |
| S | Q | R | | | | | | |
| D | S | T | | | | | | |
| F | U | V | | | | | | |
| G | W | X | | | | | | |
| H | Y | Z | | | | | | |
| J | 0 | 1 | | | | | | |
| K | 2 | 3 | | | | | | |
| L | 4 | 5 | | | | | | |
| ENTER | 6 | 7 | | | | | | |
| CAPS | 8 | 9 | | | | | | |
| Z | ` | TAB | | | | | Grave | Tab |
| X | - | "=" | | | | | Minus | Equal |
| C | [|] | | | | | L Square. | R Square. |
| V | ; | ' | | | | | Semicolon | Apostr. |
| B | , | . | | | | | Comma | Dot |
| N | + | / | | | | | Plus | Slash |
| M | ~ | ! | ~ | ! | ~ | ! | Not/Tilde | Exclam. |
| SYMBOL | @ | \$ | @ | \$ | " | \$ | At/Quote | Dollar |
| SPACE | % | ^ | | | | | Percent | Circumf. |