

This page will contain the information about the combined lcd/button matrix

right now I'm using a modified version of the sm_simple C example but this code will be rewritten to make it more coherent

modifications to scan matrix example: in main.c:

```
...
//second shiftregister drives the leds
#define LEDOUT 1
...
void LM_SetRow(){
    MIOS_DOUT_SRSet(LEDOUT,ledtest[sm_col]);
}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
/
// This function is called by MIOS before the shift register are loaded
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
/
void SR_Service_Prepare(void) __wparam
{
    // call the Scan Matrix Driver
    SM_PrepareCol();
    // call the Led Matrix Driver
    LM_SetRow();
}
...
```

in sm_simple.asm:

```
...
    global    _sm_button_column
    global    _sm_button_row
    global    _sm_button_value
    global    _sm_col

    ;; import lables
    extern    _SM_NotifyToggle

; =====

accessram      udata          ; (no access ram required, these variables can
be located anywhere)

_sm_button_column    res    1    ; exported to C, therefore an "_" has been
added
_sm_button_row       res    1
_sm_button_value     res    1
```

```
_sm_col          res 1

...

SM_PrepareCol
    ;; select next DOUT register

    ;; (current column + 1) & 0x07
    SET_BSR      sm_selected_column
    incf         sm_selected_column, W, BANKED    ; (* see note below)
    andlw        0x07
    ;_sm_col is used by LM_SetRow()
    movwf        _sm_col
    call         MIOS_HLP_GetBitANDMask          ; (inverted 1 of 8 code)

...
```

and finally in sm_simple.h:

```
...
extern unsigned char sm_button_value;
extern unsigned char sm_col;
...
```

From:

<http://wiki.midibox.org/> - **MIDIbox**

Permanent link:

http://wiki.midibox.org/doku.php?id=dseq32_matrix&rev=1152138061

Last update: **2006/10/15 09:35**

