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

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

modifications to scan matrix example uCApps

Hardware

DOUT wiring

DIN wiring

**note: schematics not finished!*

Software

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;
...
```

Appendum

To avoid flickering leds when pushing a button the MIOS button debouncing should be turned off!!

Due to the setup of the SRIO driver the debounce algoritm also delays the DOUT chain when a DIN event is being debounced. So when a button is pushed the led update frequency is reduced, the higher the debounce value, the lower the update frequency.

According to this post TK will fix this in a future MIOS release:

> This is something what I'm planning to solve in one of the next MIOS versions - currently the same

> SR scanning routine is used for DIN and DOUT registers, which means, >when the DINs are temporary disabled due to the cheap debouncing method, the DOUT registers won't be updated.

> The solution is to add a second scan routine which only services the DOUTs so long the debouncing delay is active.

In the meantime THIS ISSUE IS SOLVED WITH MIOS1.9c!!!

Workaround

To turn the debouncing off, set

```
#define DIN_DEBOUNCE_VALUE    0
```

back to DSEQ32