

vi customization

A few simple additions to make vi much more pleasant to use for C programming

From this...

```
#define F_CPU 16000000UL //16Mhz clock
#include <avr/io.h>
#include <util/delay.h>

/*****
//                               debounce_switch
// Adapted from Ganssel's "Guide to Debouncing"
// Checks the state of pushbutton S1 It shifts in ones till the button is pushed.
// Function returns a 1 only once per debounced button push so a debounce and toggle
// function can be implemented at the same time. Expects active low pushbutton on
// Port D bit zero. Debounce time is determined by external loop delay times 12.
*****/
int8_t debounce_switch() {
    static uint16_t state = 0; //holds present state
    state = (state << 1) | (! bit_is_clear(PIND, 0)) | 0xE000;
    if (state == 0xF000) return 1;
    return 0;
}

/*****/
```

To this

```
10
11 #define F_CPU 16000000UL //16Mhz clock
12 #include <avr/io.h>
13 #include <util/delay.h>
14
15 /*****
16 //          debounce_switch
17 // Adapted from Ganssel's "Guide to Debouncing"
18 // Checks the state of pushbutton S1 It shifts in ones till the button is pushed.
19 // Function returns a 1 only once per debounced button push so a debounce and toggle
20 // function can be implemented at the same time. Expects active low pushbutton on
21 // Port D bit zero. Debounce time is determined by external loop delay times 12.
22 *****/
23 int8_t debounce_switch() {
24     static uint16_t state = 0; //holds present state
25     state = (state << 1) | (! bit_is_clear(PIND, 0)) | 0xE000;
26     if (state == 0xF000) return 1;
27     return 0;
28 }
29
```

Or even this

```
File Edit Tools Syntax Buffers Window Help
35 #include <avr/io.h>
36 #include <util/delay.h>
37 #include <avr/interrupt.h>
38
39 uint8_t segment_data[5];
40
41 //*****
42 //                               chk_buttons
43 //*****
44
45 uint8_t chk_buttons(uint8_t button) {
46
47     static uint16_t state[8] = {0,0,0,0,0,0,0,0}; //holds present state of each pin
48     state[button] = (state[button] << 1) | (! bit_is_clear(PINA, button)) | 0xE000;
49     if (state[button] == 0xF000) return 1;
50     return 0;
51
52     //ISR timer to check button status
53 }
54
55 ISR(TIMERO_OVF_vect) {
56     button_flag = 1;
57 }
58 }
```

38,0-1 9%

My .vimrc file

File Edit View Terminal Tabs Help

```
1 :filetype on
2 :autocmd FileType c,cpp :set cindent
3 syntax on
4 :set number
5 colorscheme torte
```

".vimrc" 5L, 90C

1,1

All

.vimrc options explained

- `:filetype on`

Detects the type of file being edited.

- `:autocmd FileType c, cpp :set cindent`

Automatic C indenting in C or C++

- `syntax on`

Syntax highlighting

.vimrc options explained

- :set number

Displays line numbers

- colorscheme torte

The particular color scheme I like

Some other common schemes are: blue, delek, evening, murphy, slate, darkblue, desert, koehler, pablo, torte, default, elflord, morning, peachpuff, shine, and zellner.

Additional resources

- `/usr/share/vim/vim71/colors`
- <http://www.vim.org/html/doc/options.html>
- <http://people.cis.ksu.edu/~bhoward/vi/>