

```

/* Blinkl_An_Aus_Taster121114.c
 * Created: 12.11.2014 11:21:40
 * Author: Uhli
 * LED auf C5, Taster auf B0
 * Verwendung der Entprellroutine durch Flankenerkennung nach µCnet
 */
#define F_CPU 8000000UL // 8 M Hz interner Oszi; Takt Def. immer als 1.!
#include <avr/io.h> // Standard-Bibliotheken, dann eigene Bibliotheken
#include <util/delay.h>

// #include "G:\Mikroprogs\6.2\Eig_header\Entprell-header.h"
// Entprellheader Taster
# include "G:\Mikroprogs\6.2\Eig_header\LED_blinken.h"

// #####
// Entprell-header; Ziel ist einbindung als header:h #####
#define TASTERPORT PORTC
#define TASTERBIT PINC5
char rw; // rw ist Rueckgabewert
char taster(void)
{
    static unsigned char zustand; // Taster hat 4 zustaende
    rw = 0;

    if(zustand == 0 && !(TASTERPORT & (1<<TASTERBIT)))
    {zustand = 1; // Taster WIRD GEDRUECKT (steigende Flanke)
    rw = 1; // hier ist rw das einzige mal 1 !
    }

    else if (zustand == 1 && !(TASTERPORT & (1<<TASTERBIT)))
    {zustand = 2; // Taster GEHALTEN
    rw = 0;
    }

    else if (zustand == 2 && (TASTERPORT & (1<<TASTERBIT)))
    {zustand = 3; // Taster WIRD LOSGELASSEN (fallende Flanke)
    rw = 0;
    }

    else if (zustand == 3 && (TASTERPORT & (1<<TASTERBIT)))
    {zustand = 0; //Taster NICHT BETAETIGT
    rw = 0;
    }

    return rw;
} // Ende Entprell-header ##### Ende Entprell-header #####
// #####
//
// main ##### main ##### main ##### main ##### main #####
int main(void)
{
    DDRC |= _BV(PC5); // LED ;setzen, also Ausgang _BV entspr. mein. wiss. 1<<
    DDRB &= ~(1<<PB0); // Taster; nicht setzen, also Eingang
    PORTB |= _BV(PB0); // Pull Up Widerstand
    // (nicht noetig wenn hardwaremaessig realisiert !)
    int Tastendruckzaehler=0; // 0=LED aus, 1=Ledblinken(), 2=LED ein
    //
    while(1) // Endlosschleife des Hauptprogrammes #####
    {
        taster();
        //
        if(rw==1) // Tastendruckzaehler=0
        {PORTC|=(1<<PC5); Tastendruckzaehler++;} // neg. Logik -> + schaltet, LED aus
        //
        if((rw==1)&&(Tastendruckzaehler==1)) // blink blink blink
        {Ledblinken(); Tastendruckzaehler++;}
        //
        if((rw==1)&&(Tastendruckzaehler==2)) // leucht leucht leucht
        {PORTC&=~(1<<PC5); Tastendruckzaehler=0;} // neg. Logik -> - schaltet, LED ein
        //
    } // Ende while (1) #####
    //
    return 0;
} // Ende main #####

```

```

/*static unsigned char Zustand;
char ET = 0; // ET ist entprellter Tastendruck
char Entprellung(void) // der Fktn werden keine Para uebergeben
{

```

DAS GELB UNTERLEGTE
IST ALLES AUSKOMMENTIERT !!

```

    if(Zustand == 0 &&!((1<<TASTERPORT) & (1<<TASTERBIT) ))
    {Zustand = 1; // Taster wird gedruickt (steigende Flanke)
    ET = 1;
    }

    else if (Zustand == 1 &&!((1<<TASTERPORT) & (1<<TASTERBIT) ))
    {Zustand = 2; // Taster wird gehalten
    ET = 0;
    }

    else if ( ((Zustand == 1)|| (Zustand == 2)) &&!(1<<TASTERPORT)&(1<<TASTERBIT) )
    {Zustand = 2; // Taster wird gehalten
    ET = 0;
    }

    else if (Zustand == 2 && ((1<<TASTERPORT)&(1<<TASTERBIT) ))
    {Zustand = 3; //Taster wird losgelassen (fallende Flanke)
    ET = 0;
    }

    else if (Zustand == 3 && ((1<<TASTERPORT)&(1<<TASTERBIT) ))
    {Zustand = 0; //Taster losgelassen
    ET = 0;
    }

return ET; // das Prog gibt ET zurueck
}
// Ende Entprellroutine #####
//#####*/

/* // Andere Entprellmethode #####
int Modus = 0; // Modus=1->blinken und Modus=0->kein_blinken
while(1) // Aufruf von Ledblinken und Tasterabfrage mit delays
{ if ( bit_is_clear(PINB,0) && (Modus==0) )
    {Ledblinken();
    Modus=1;
    }

    if ( bit_is_set(PINB,0) && (Modus==1) )
    {Ledblinken();
    }

    if ( bit_is_clear(PINB,0) && (Modus==1) )
    {Modus=0;
    delay_ms(250);
    }
} // Ende while(1) #####
#####*/

```