Datei Taster

```
unsigned short lfsr = 0xACE1u;
unsigned bit;

static unsigned int count = 0;

void Taster()
{
  P2IES = BIT0+BIT5;   // BIT0, BIT5;
  P2IE = BIT0+BIT5;    //Interrupt enable für die Taster;
}

#pragma vector=PORT2_VECTOR
__interrupt void P2_ISR(void)
{

 if (P2IFG & BIT5)
  {
 //  bit = 0;
   P1OUT = 0xFF;
   P2IFG &=~BIT5;

   // Zufahlszahlgenerator , einfach LFSR
   bit  = ((lfsr >> 0) ^ (lfsr >> 2) ^ (lfsr >> 3) ^ (lfsr >> 5) ) & 1;
   lfsr =  (lfsr >> 1) | (bit << 15);

   if ( (lfsr > 0) && (lfsr < 16000))
     count = 30000;
   else if ( (lfsr >= 16000) && (lfsr < 32000))
     count = 40000;
   else if ( (lfsr >= 32000) && (lfsr < 48000))
     count = 50000;
   else if ( (lfsr >= 45000) && (lfsr < 65535))
     count = 60000;

 TimerA_init(count);
  }

  else if (P2IFG & BIT0)
  {
    P2IFG &=~ BIT0;
    P1OUT = 0x00;
  }

}
```

Datei TimerA

```
void TimerA_init ()
{

  TACTL = TASSEL_1 + TAIE;  // SMCLK, Interrupt Enable

  CCTL0 =  CCIE;      //
  CCR0 = 8192;
  TACTL |= MC_1;          // Timer im Up Mode


}

// wenn CCR0 erreicht wird
#pragma vector=TIMERA0_VECTOR
__interrupt void Timer_A0 (void)
  {
    P1OUT = 0x00;  //   LED an
    CCTL0 &=  ~CCIFG;
    TACTL &= ~TAIFG;

 }
```

Datei main

```
#include "msp430x22x2.h"
#include "Timer_A.c"
#include "Taster.c"
#include "BCM.c"

int main( void )
{
  WDTCTL = WDTPW + WDTHOLD; // Stop watchdog timer to prevent time out reset
  P1DIR = 0xFF;
  P1OUT = 0x00;
 // P1OUT = 0xFF;
  BCM();
 // Taster();
  _BIS_SR(LPM0_bits + GIE);
//  __enable_interrupt();
//  __low_power_mode_0();
  while (1) {
  if (P2SEL + BIT0)
    P1OUT = 0x0F;
  }
}
```