

Mit der FC-Hardware PPM decodieren :

Hier werden 6 Kanäle gelesen (4 x Steuerung plus 2 Schaltkanäle).

Mit diesem Teil (der dann in einer Klasse „PPM“ als Methode „PPMinit“ aufgerufen wird) steht die Initialisierung der Ports auf der FC und des Interruptverhaltens :

```
DDRD &= ~(1<<PORTD6);    //in für ppm-signal
PORTD |= (1<<PORTD6);
TCCR1A = 0;
TCCR1B = 195;
TCCR1C = 0;
TIMSK1 = 32;
```

Die eigentliche ISR (Interrupt Service Routine) steht am Ende des Hauptprogramms, außerhalb der 200hz-Routine, die nötigen Variablen sind in main definiert :

```
ISR(TIMER1_CAPT_vect)
{
    wert = ICR1;                //wenn ein Interrupt erkannt, wird der Zählerwert übernommen
    TCNT1H=0;                  //und der Zähler wieder Null gesetzt
    TCNT1L=0;
    if (wert < 300)             //störung bei Zählwert zu klein
    {
        kanalcounter = 0;      //dann alles von vorne !
        valid = 0;
    }
    if (wert > 2000)            //sync-impuls falls wert groß genug
    {
        kanalcounter = 0;      //kanal rücksetzen
        valid = 1;              //ab jetzt gilt's, die kanäle werden nacheinander gemessen
    }
    else                         //gültiger impuls (kein sync, keine störung)
    {
        if (valid==1)
        {
            kanal[kanalcounter] = wert;    //kanal(x) wird mit zählwert gesetzt
            kanalcounter ++;
            if (kanalcounter > 6)           //bis zum 6. Kanal, dann wieder von vorne
            {
                kanalcounter =0;
                valid=0;
            }
        }
    }
}
```

Funktionserklärung für den benutzten Timer 1 des ATmega-Prozessors :

http://www.mikrocontroller.net/articles/AVR-GCC-Tutorial/Die_Timer_und_Zähler_des_AVR#8-Bit_Timer.2FCounter