```c
#include <avr/io.h>

#include "internerADC.h"
#include "globalextern.h"


void ADC_Init(void)
{

  ADMUX = (1<<REFS0);                                 // die Versorgungsspannung
    AVcc als Referenz wählen
  // ADMUX = (1<<REFS1) | (1<<REFS0);                 // oder interne 2.5V
    Referenzspannung als Referenz für den ADC wählen:

  ADCSRA = (1<<ADEN)+(1<<ADPS2)+(1<<ADPS1);           // Frequenzvorteiler = 64
                                                      // ADC aktivieren

  /* nach Aktivieren des ADC wird ein "Dummy-Readout" empfohlen, man liest
     also einen Wert und verwirft diesen, um den ADC "warmlaufen zu lassen" */

  ADCSRA |= (1<<ADSC);                                // eine ADC-Wandlung
  while (ADCSRA & (1<<ADSC) ) {                       // auf Abschluss der
    Konvertierung warten
  }

  (void) ADCW;                                        // ADCW muss einmal gelesen
     werden, sonst wird Ergebnis der nächsten Wandlung nicht übernommen
}


uint16_t ADC_Read( uint8_t channel )                  // ADC Einzelmessung
{
    ADMUX = (ADMUX & ~(0x1F)) | (channel & 0x1F);     // Kanal waehlen, ohne
      andere Bits zu beeinflußen
    ADCSRA |= (1<<ADSC);                              // eine Wandlung
      "single conversion"
    while (ADCSRA & (1<<ADSC) ) {                     // auf Abschluss der
      Konvertierung warten
    }
    return ADCW;                                      // ADC auslesen und
      zurückgeben
}


uint16_t ADC_Read_Avg( uint8_t channel, uint8_t nsamples )  /* ADC Mehrfachmessung
  mit Mittelwertbbildung */
{
    uint32_t sum = 0;                                 /* beachte:
      Wertebereich der Summenvariablen */

    for (uint8_t i = 0; i < nsamples; ++i ) {
        sum += ADC_Read( channel );
    }
```

```
    return (uint16_t)( sum / nsamples );
}
```