```
#include "allpic.h"
#pragma config = 0b000110000100          // hex-datei: fuses des 12F629

#define _LED            0                // led-ausgang
#define _FAST           1                // integrator-steuerung 8.2k/10uF
#define _SLOW           2                // integrator-steuerung 1Meg/10uF
#define _INC            3                // relaiskontakt Normal Close
#define _INO            4                // relaiskontakt Normal Open
#define _DDA            5                // analog-ausgabe

static uns16 TMR1 @ (uns16)&TMR1L;

//********************** macros und prototypen *********************************

#define CLOSE   1                        // kontaktabfragen
#define OPEN    0
#define RISE    1                        // relaistrom
#define SINK    0

#define REL_WAIT(in,close) {             /* kontaktabfrage */                   \
        #if (close)                                                             \
          while(GPIO.in);                /* warte auf kontaktschließung */      \
        #else                                                                   \
          while(!GPIO.in);               /* warte auf kontaktöffnung */         \
        #endif                                                                  \
}

#define DDA_RESOL     16                 // 8 oder 16
#define DDA_LEN             16           // 8 oder 16
#define DDA_US              50           // +16us * 65536 = laufzeit dda_out 16 bit
                                         // +12us * 256 = laufzeit dda_out 8 bit

//#define genAdd(r,a) { W = (a); if(Carry) W = incsz(a); (r) += W; }
#define genSub(r,a) { W = (a); if(Borrow) W = incsz(a); (r) -= W; }

#define DDA_OUT(v) {                                                            \
    #if DDA_RESOL == 8                                                          \
        uns8 err = 0;                    /* 8 bit-analogausgabe */              \
    #else                                                                       \
        uns16 err = 0;                   /* 16 bit-analogausgabe */             \
    #endif                                                                      \
    #if DDA_LEN == 8                                                            \
        uns8 len = 0;                    /* 256 bits */                         \
    #else                                                                       \
        uns16 len = 0;                   /* 65536 bits */                       \
    #endif                                                                      \
        do {                                                                    \
          GPIO._DDA = FALSE;                                                    \
          err.low8 -= (v).low8;                                                 \
         #if DDA_RESOL == 16                                                    \
          genSub(err.high8,(v).high8);   /* 16 bit */                           \
         #endif                                                                 \
          if(Borrow) GPIO._DDA = TRUE;                                          \
          INDF._DDA = FALSE;             /* kurzer high oder low-impuls */      \
          USEC(DDA_US);                                                         \
          INDF._DDA = TRUE;                                                     \
    #if DDA_LEN == 8                                                            \
        } while(++len);                  /* 8bit länge */                       \
    #else                                                                       \
        } while((++(len.low8))||(++(len.high8)));    /* 16-bit länge */         \
    #endif                                                                      \
}
```

```
#define INIT {                                                          \
        RP0 = TRUE;                     /* spezialregister auswählen */ \
        OPTION = 0;                     /* TMR0: 1:1, pullup ein (20kOhm) */ \
        WPU = _BV(_INC) | _BV(_INO); /* _INC: externer pullup */        \
        RP0 = FALSE;                    /* normale register und ram */  \
        CMCON = 0x07;                   /* komparator aus */            \
        GPIO = 0;                       /* alle portbits löschen */     \
        FSR = (&TRISIO);                /* tristate-steuerung per INDF */ \
        INDF = ~(_BV(_LED) | _BV(_DDA));/* led und dda ausgang auf low */ \
}

#define MEASURE {                                                       \
        TMR1 = 0; TMR1IF = FALSE;    /* zeitmessung vorbereiten */      \
        GPIO._SLOW = SINK;           /* spulenstrom sinkt langsam */    \
        INDF._SLOW = FALSE;          /* strom einschalten */            \
        REL_WAIT(_INO,OPEN);         /* warte bis INO öffnet */         \
        INDF._SLOW = TRUE;           /* strom halten */                 \
        TMR1ON = TRUE;               /* messung: timer starten */       \
        REL_WAIT(_INC,CLOSE);        /* schwung bis INC schließt */     \
        TMR1ON = FALSE;              /* timer stoppen */                \
}

#define REL_RESET {                                                     \
        GPIO._FAST = RISE;           /* spulenstrom steigt schnell */   \
        INDF._FAST = FALSE;          /* strom einschalten */            \
        REL_WAIT(_INO,CLOSE);        /* warte bis INO schließt */       \
        INDF._FAST = TRUE;           /* strom halten */                 \
}

#define DA_OUT {                                                        \
        GPIO._LED = TRUE;            /* lampe an */                     \
        if(TMR1IF) {                 /* wenn überlauf... */             \
          GPIO._LED = FALSE;         /* lampe aus */                    \
          TMR1 = 0;                  /* timer löschen */                \
        }                                                               \
        DDA_OUT(TMR1);               /* analogausgabe */                \
}

#define ONLY_PROTOTYPES
#include "softtime.h"

//********************** programm *********************************************
#pragma resetVector -                // programm ab 0x000 ohne goto main

void main(void) {                   // ab hier funktionen und prozeduren
        INIT;                        // hardware konfiguration
        FOREVER {                    // ewige schleife
          MEASURE;                   // messung
          REL_RESET;                 // relais scharf machen
          DA_OUT;                    // analogwert ausgeben
        }
}

// #include "softtime.h"

/* ENDE */
```