

---

---

## Anhang C

### Funktionen des BORIS-Parsers

Hier werden alle Funktionen des BORIS-Parsers aufgeführt und ihre Entsprechungen in C erläutert, sofern diese nicht in der Headerdatei *math.h* definiert worden sind. Da der BORIS-Parser nur unter Verwendung von Fließpunktzahlen zu C-Code generiert werden kann, betreffen die hier aufgeführten Funktionen lediglich die Fließpunktarithmetik (mit Ausnahme der round-Funktion, die für beide Zahlentypen zur Verfügung steht und daher in einen separaten Funktionscodeabschnitt ausgelagert wurde). Im Funktionscodeabschnitt */\*Definitions for floatingpoint\*/* werden Makrodefinitionen getroffen und der Abschnitt für die round-Funktion eingebunden:

```
/*Definitions for floatingpoint*/
#include <math.h>
#include <stdlib.h>
#define _int(a) ((SVartyp)(long)(a))
#define frac(a) ((a)-(SVartyp)(long)(a))
#define sign(a) ((a<0)?-1:(a>0)?1:0)
#define step(a) ((a<0)?0:1)
#define __Pow(a,b) pow(a,b)
#define sqr(a) __Pow(a,2)
#define deg(a) (a*180/PI)
#define rad(a) (a*PI/180)
#InsertBlock /*round*/
/*END*/
```

*Makrodefinitionen zur Anpassung entsprechender BORIS-Parser-Funktionen*

Diese werden entsprechend folgender Tabelle eingesetzt:

Funktion im BORIS-Parser	Eingesetztes Makro bzw. Funktion
int	_int
frac	frac
sign	sign
step	step
round	round
^	__Pow
sqr	sqr
deg	deg
rad	rad

Der Funktionscodeabschnitt der round-Funktion sieht folgendermaßen aus:

```

/*round*/
SVartyp round(const SVartyp a)
{
    SVartyp zw=a;
    char vz=0;
    if (zw<0){vz=1; zw=-zw;}
#ifdef FLOATINGPOINT
    zw=((zw&FracMask)>(FracMask>>1)) ? (zw&IntMask)+One : zw &
IntMask;
#else
    zw=(frac(zw)>=0.5)? ceil(zw): floor(zw);
#endif
    if (vz)zw=-zw;
    return zw;
}

```

```
/*end*/
```

Die `_Pow` Makrodefinition ist möglicherweise geringfügig different zur BORIS-Potenzfunktion (compilerabhängig). Soll exakte Gleichheit erzielt werden, so muß die Makrodefinition innerhalb des Funktionsabschnittes durch die untenstehende Funktion ersetzt werden. Eventuell muß `<errno.h>` eingefügt werden!

```
MEM_ATTRIBUTE SVartyp __Pow(SVartyp x,SVartyp y)
{ SVartyp zw;
  if (x!=0){
    zw=log(fabs(x));
    zw*=y;
    zw=exp(zw);
    if(x<0)
      if (zw!=0) zw=1/zw;
      else errno=EDOM;
    return zw;
  }else return 0;
}
```

*Potenzfunktion der entsprechenden BORIS -Parser-Funktionen*