

Jetzt schaut die Umbelegungsanweisung in der Sprache **AHK** so aus :

```
SCOD::Send A
+SCOD::Send B
^SCOD::Send C
+^SCOD::Send D
!SCOD::Send E
!+SCOD::Send F
!^SCOD::Send G
!+^SCOD::Send H
#SCOD::Send I
#+SCOD::Send J
#^SCOD::Send K
#+^SCOD::Send L
#!SCOD::Send M
#!+SCOD::Send N
#!^SCOD::Send P
<^>!SCOD::Send Q
<^>!+SCOD::Send R
```

Man sieht, wie viele verschiedene Buchstaben man ein und derselben Taste mit verschiedenen Kombinationen der Umschaltetasten zuweisen könnte. Das ist natürlich nur eine Übung, niemand wird das ausnutzen wollen.

Man könnte, etwa als Mathematiker oder als Austro-Griechen auf die Idee kommen, die Tastatur auch mit dem griechischen Alphabet zu belegen. Dazu kann man prinzipiell die Taste **[AltGr]** verwenden, würde die Taste **[AltGr]** nicht einige Zeichen erzeugen, nämlich @€ und µ. Meine Lösung schaut so aus:

### symbols.ahk

Das griechische Alphabet braucht kein Q, daher kann das Zeichen @ bleiben, wo es ist. Das µ steht ohnehin an der Stelle, an der es auch im griechischen Alphabet stehen würde, es kann daher ebenfalls bleiben.

Man verschiebt also das € in die Zahlenreihe, (**symbols.ahk**) damit sind alle Alpha-Zeichen frei von einer Doppelbelegung.

Die **[AltGr]**-Taste bietet sich für die unbelegten Zeichen in der Zahlenreihe an.

```
RA1t & 1::Send {U+00b9} ;Eins hochgestellt
RA1t & 5::Send {U+20ac} ;Euro
```

Es ist mir aufgefallen, dass das jüngste Zeichen, „das große scharfe s“, manchmal nicht mit **[AltGr]** **[⊕]** **[ß]** gesendet werden kann. Die letzte Zeile dieser Symboldefinitionen **symbols.ahk** generiert dieses Zeichen.

Die Umlaut-Tasten bekommen die Zusatzfunktion, dass sie mit der **AltGr**-Taste die **Html**-Ersatzcodes generieren.

Hinweis **RA1t** entspricht **<^>!**

### greek.ahk

Man ordnet jedem Alpha-Zeichen das entsprechende griechische Zeichen zu und aktiviert den griechischen Kleinbuchstaben mit **[AltGr]** und den griechischen Großbuchstaben mit **[⊕]** **[AltGr]**.

Um jetzt zum Beispiel die Taste **[A]** mit dem Alpha-Zeichen zu belegen, benötigt man zwei **AHK**-Zeilen:

```
RA1t & a::Send(0x03b1) ; Alpha klein α
!RA1t & a::Send(0x0391) ; Alpha groß Α
```

Und so weiter für alle anderen griechischen Zeichen.

Wenn man daher beabsichtigt, die Tastatur so umzubelegen, dass jede Taste gemeinsam mit **[AltGr]** und **[⊕]** **[AltGr]** eine weitere Bedeutung bekommt, dann lohnt es sich, dafür eine Funktion zu schreiben und damit eine Taste in einer Zeile abzarbeiten. Die Funktion nennen wir **ChkShift**.



### symbols.ahk

```
;Zeichen
<^>!1::Send {U+00b9} ;¹
+<^>!1::Send {U+00a9} ;©
+<^>!2::Send {U+00ae} ;®
+<^>!3::Send {U+00b7} ;·
<^>!4::Send {U+20ac} ;€
+<^>!4::Send {U+2022} ;·
<^>!5::Send {U+2030} ;‰
+<^>!5::Send {U+2122} ;™
<^>!6::Send {U+263a} ;☺
+<^>!6::Send {U+263b} ;☹
+<^>!8::Send {U+00ab} ;«
+<^>!9::Send {U+00bb} ;»
+<^>!SC56::Send {U+00b1} ;±
<^>!SC2b::Send {U+2264} ;≤
+<^>!SC2b::Send {U+2265} ;≥
+<^>!SC1b::Send {U+2248} ;≈
+<^>!SC0c::Send {U+1E9E} ;ß

<^>!SC28::Send &auml ;
+<^>!SC28::Send &Auml ;
<^>!SC27::Send &ouml ;
+<^>!SC27::Send &Ouml ;
<^>!SC1a::Send &uuml ;
+<^>!SC1a::Send &Uuml ;
+<^>!SC1b::Send &szlig ;
```

### grafik.ahk

```
;Rahmenzeichen
RA1t & i::ChkShift(0x250c, 0x2554) ;┌┐
RA1t & o::ChkShift(0x252c, 0x2566) ;└┘
RA1t & p::ChkShift(0x2510, 0x2557) ;┌┐
RA1t & j::ChkShift(0x251c, 0x2560) ;└┘
RA1t & k::ChkShift(0x253c, 0x256c) ;└┘
RA1t & l::ChkShift(0x2524, 0x2563) ;└┘
RA1t & n::ChkShift(0x2514, 0x255A) ;└┘
RA1t & m::ChkShift(0x2534, 0x2569) ;└┘
RA1t & ,::ChkShift(0x2518, 0x255D) ;└┘
RA1t & .::ChkShift(0x2500, 0x2550) ;=
RA1t & -::ChkShift(0x2502, 0x2551) ;||
;Pfeile
RA1t & t::ChkShift(0x2196, 0x21d6) ;↖
RA1t & z::ChkShift(0x2191, 0x21d1) ;↑
RA1t & u::ChkShift(0x2197, 0x21d7) ;↗
RA1t & f::ChkShift(0x2190, 0x21d0) ;←
RA1t & g::ChkShift(0x2194, 0x21d4) ;↔
RA1t & h::ChkShift(0x2192, 0x21d2) ;⇒
RA1t & c::ChkShift(0x2199, 0x21d9) ;↘
RA1t & v::ChkShift(0x2193, 0x21d3) ;↓
RA1t & b::ChkShift(0x2198, 0x21d8) ;↙
```

```
ChkShift(ShiftUpCode, ShiftDownCode)
{
    ShiftUpCode := Chr(ShiftUpCode)
    ShiftDownCode := Chr(ShiftDownCode)
    GetKeyState, state, Shift
    if state = D
        Send %ShiftDownCode%
    else
        Send %ShiftUpCode%
}
```

### greek.ahk

```
;griechisches Alphabet
RA1t & a::ChkShift(0x3b1, 0x0391) ;Alpha
RA1t & b::ChkShift(0x3b2, 0x0392) ;Beta
RA1t & c::ChkShift(0x3c8, 0x03a8) ;Psi
RA1t & d::ChkShift(0x3b4, 0x0394) ;Delta
RA1t & e::ChkShift(0x3b5, 0x0395) ;Epsilon
RA1t & f::ChkShift(0x3c6, 0x03a6) ;Phi
RA1t & g::ChkShift(0x3b3, 0x0393) ;Gamma
RA1t & h::ChkShift(0x3b7, 0x0397) ;Eta
RA1t & i::ChkShift(0x3b9, 0x0399) ;Iota
RA1t & j::ChkShift(0x3be, 0x039e) ;Xi
RA1t & k::ChkShift(0x3ba, 0x039a) ;Kappa
RA1t & l::ChkShift(0x3bb, 0x039b) ;Lambda
RA1t & m::ChkShift(0x3bc, 0x039c) ;Mu
RA1t & n::ChkShift(0x3bd, 0x039d) ;Nu
RA1t & o::ChkShift(0x3bf, 0x039f) ;Omicron
RA1t & p::ChkShift(0x3c0, 0x03a0) ;Pi
RA1t & r::ChkShift(0x3c1, 0x03a1) ;Rho
RA1t & s::ChkShift(0x3c3, 0x03a3) ;Sigma
RA1t & t::ChkShift(0x3c4, 0x03a4) ;Tau
RA1t & u::ChkShift(0x3b8, 0x0398) ;Theta
RA1t & v::ChkShift(0x3c9, 0x03a9) ;Omega
RA1t & w::ChkShift(0x3c2, 0x03a2) ;Sigma1
RA1t & x::ChkShift(0x3c7, 0x03a7) ;Chi
RA1t & y::ChkShift(0x3c5, 0x03a5) ;Upsilon
RA1t & z::ChkShift(0x3b6, 0x0396) ;Zeta
ChkShift(ShiftUpCode, ShiftDownCode)
{
    ShiftUpCode := Chr(ShiftUpCode)
    ShiftDownCode := Chr(ShiftDownCode)
    GetKeyState, state, Shift
    if state = D
        Send %ShiftDownCode%
    else
        Send %ShiftUpCode%
}
```