PS2 Maus Emulator

Zuerst noch etwas grundsätzliches zum Keyboard: Jedes Keyboard sendet beim drücken (MAKE) einer Taste und beim loslassen einer Taste (Break)
jeweils einen code. Diese heissen scan codes. Heisst die Taste Shift einmal kurz gedrückt sendet den code &H12, &HF0, &H12. Hier sieht man schön das man manche Codes zerlegen muss, nur Shift drücken nutzt ja wie in Zeile 28 nichts. Also zerlegen in MAKE: &H12, dann den Buchstaben t senden, dann das BREAK für shift.
Der scancode muss an anfang die Menge der zu Übertragengen codes angeben, im fall von CTRL_ALT_entf sind es also 11
Hier eine auflistung der Scancodes, bitte beachten das diese für ein Englisches Tastaturlayout sind, z.b. kommt dann bei eingestelltem Duetschen Layout statt einem z ein Y auf dem Computer raus.
computer-engineering.org/ps2keyboard/scancodes2.html
Hier einmal als Beispiel:

Quellcode

```
1.  $regfile = "m328pdef.dat" 'Der ATMEGA328P wird genutzt
2.  $crystal = 8000000 'Es werden die internen 8MHz genutzt
3.  $hwstack = 150
4.  $swstack = 150
5.  $framesize = 150
6.  $lib "at_emulator.lib"
11. Config Atemu = Int1 , Data = Pind.3 , Clock = Pind.7
12. 'PS2 Tastatur sim config - Datapin MUSS ein INT pin sein, 2,2K pullups an Data und Clock leitung notwendig
13. Enable Interrupts
15. do
16. wait 5
17. 'schreibt nach 5 Sekunden= Test
18. Sendscankbd Taste_shift_up 'Shift make code senden
19. waitms 300
20. Sendscankbd Taste_t 'T senden
21. waitms 300
22. Sendscankbd Taste_shift_down 'Shift BREAK code senden
23. waitms 300
24. Sendscankbd Taste_e
25. waitms 300
26. Sendscankbd Taste_s
27. waitms 300
28. Sendscankbd Taste_shift 'Hier wird weiter klein geschrieben da der BREAK direkt nach dem MAKE code gesendet
    wurde
29. waitms 300
30. Sendscankbd Taste_t
31. waitms 300
32. loop
35. ' *** Scancodes ***
36. Taste_e:
37. Data 3 , &H24 , &HF0 , &H24
39. Taste_s:
40. Data 3 , &H1B , &HF0 , &H1B
42. Taste_t:
43. Data 3 , &H2C , &HF0 , &H2C
45. Taste_shift:
46. Data 3 , &H12, &HF0, &H12
48. Taste_shift_up:
49. Data 1 , &H12
51. Taste_shift_down:
52. Data 2 , &HF0, &H12
53. Taste_ctrl_alt_entf:
```

55. Data 11 , &H14 , &H11 , &HE0 , &H71 , &HE0 , &HF0 , &H71 , &HF0 , &H11 , &HF0 , &H14

Alles anzeigen

Aber es geht natürlich noch mehr, wir können auch die Maus mitbenutzen, angegeben wird die Verschiebung in Pixel:
Die Funktioniert mit dem Befehl PSMOUSEXY oder einzelne Pixel mit scancodes:

Quellcode

```
$regfile = "m328pdef.dat" 'Der ATMEGA328P wird genutzt
$crystal = 8000000 'Es werden die internen 8MHz genutzt
$hwstack = 150
$swstack = 150
$framesize = 150
Dim A as Byte
Enable Interrupts
$lib "at_emulator.lib"
Config Atemu = Int1 , Data = Pind.3 , Clock = Pind.7
'PS2 Tastatur sim config - Datapin MUSS ein INT pin sein, 2,2K pullups an Data und Clock leitung notwendig
CONFIG PS2EMU= int0 , DATA = PIND.2, CLOCK=pind.6
'PS2 Maus sim config - Datapin MUSS ein INT pin sein, 2,2K pullups an Data und Clock leitung notwendig
do
wait 5
'schreibt nach 5 Sekunden= Test
Sendscankbd Taste_shift_up 'Shift make code senden
waitms 300
Sendscankbd Taste_t 'T senden
waitms 300
Sendscankbd Taste_shift_down 'Shift BREAK code senden
waitms 300
Sendscankbd Taste_e
waitms 300
Sendscankbd Taste_s
waitms 300
Sendscankbd Taste_shift 'Hier wird weiter klein geschrieben da der BREAK direkt nach dem MAKE code gesendet wurde
waitms 300
Sendscankbd Taste_t
waitms 300
wait 5
PS2MOUSEXY 30 , 0, 0 '30 pixel nach rechts
waitms 500
for A = 1 to 30 'jetzt mal als Scancode, 30x ein pixel nach oben
Sendscan Mouseup
Waitms 10
next A
waitms 500
PS2MOUSEXY -30 , 0, 0 '30 pixel nach links
waitms 500
PS2MOUSEXY 0 , -30, 0 '30 pixel nach unten
```

```
waitms 500
sendscan R_close 'Maustasten funktionieren auch als Scancode, rechte Taste gedrückt
waitms 500
sendscan R_open 'Maustasten funktionieren auch als Scancode, rechte Taste losgelassen
waitms 500
PS2MOUSEXY 0 , 0, 1 'linke Maustaste gedrückt
waitms 500
PS2MOUSEXY 0 , 0, 0 'linke Maustaste losgelassent
waitms 500
loop
' *** Scancodes ***
Taste_e:
Data 3 , &H24 , &HF0 , &H24
Taste_s:
Data 3 , &H1B , &HF0 , &H1B
Taste_t:
Data 3 , &H2C , &HF0 , &H2C
Taste_shift:
Data 3 , &H12, &HF0, &H12
Taste_shift_up:
Data 1 , &H12
Taste_shift_down:
Data 2 , &HF0, &H12
Taste_ctrl_alt_entf:
Data 11 , &H14 , &H11 , &HE0 , &H71 , &HE0 , &HF0 , &H71 , &HF0 , &H11 , &HF0 , &H14
Mouseup:
Data 3 , &H08 , &H00 , &H01
R_open:
Data 3 , &H08,&H00,&H00
R_close:
Data 3 , &H0a,&H00,&H00
```

Alles anzeigen