

# BASCOM WS2812 Knightrider

## rainbow\_ws2812\_Knightrider

```

1. '-----
2. '                                     rainbow_ws2812_Knightrider.bas
3. '-----
4. $Regfile = "m88pdef.dat"
5. $Crystal = 8000000
6. $hwstack = 40
7. $swstack = 16
8. $framesize = 32
9.
10. Config RAINBOW=1, RB0_LEN=8, RB0_PORT=PORTB,rb0_pin=0
11. '                                     ^ connected to pin
12. '                                     ^----- connected to
13. '                                     ^----- 8 leds on stripe
14. '                                     ^----- 1 channel
15.
16.
17. 'Global Color-variables
18. Dim Color(3) as Byte
19. R alias Color(_base) : G alias Color(_base + 1) : B alias Color(_base +
20. 2)
21. 'CONST
22. const numLeds=8
23.
24. '---- [MAIN]-----
25. Dim n as Byte
26.
27. RB_SelectChannel 0           ' select first channel
28. R = 50 : G = 0 : B = 100    ' define a color
29. RB_SetColor 0 , color(1)    ' update leds
30. RB_Send
31.
32. Do
33.   For n = 1 to Numleds-1
34.     rb_Shiftright 0 , Numleds 'shift to the right all leds except the
35.       Waitms 100
36.       RB_Send

```

```
37. Next
38. For n = 1 to Numleds-1
39.   rb_Shiftleft 0 , Numleds   'shift to the left all leds except the
   last one
40.   Waitms 100
41.   RB_Send
42. Next
43. waitms 500                 'wait a bit
44. Loop
```

## rainbow\_ws2812\_KnightriderDual-RGBW

```
1. '-----
   '-----
2. '                               rainbow_ws2812_KnightriderDual-RGBW.bas
3. '-----
   '-----
4. $Regfile = "m88pdef.dat"
5. $Crystal = 8000000
6. $hwstack = 40
7. $swstack = 16
8. $framesize = 32
9.
10. Config RAINBOW = 1 , rgb = 4 , RB0_LEN = 8 , RB0_PORT = PORTB , rb0_pin
   = 0
11. '                               ^-- using rgbW leds #### MUST BE FIRST PARAMETER
   when defined ###
12. '                               ^ connected to pin
   0
13. '                               ^----- connected to
   portB
14. '                               ^----- 8 leds on stripe
15. '                               ^----- 1 channel
16.
17.
18. 'Global Color-variables
19. Dim Color(4) as Byte
20. R alias Color(_base) : G alias Color(_base + 1) : B alias Color(_base +
   2) : W alias color(_base + 3)
21.
22. 'CONST
23. const numLeds = 8
24.
25. '---- [MAIN]-----
   '-----
26. Dim n as Byte
```

```

27.
28. RB_SelectChannel 0 ' select
    first channel
29. R = 50 : G = 0 : B = 100 : w = 10 ' define a
    color
30. RB_SetColor 0 , color(_base) ' update led
    on the left
31. RB_SetColor numleds - 1 , color(_base) ' update led
    on the right
32. RB_Send
33. waitms 2000
34.
35. Do
36.   For n = 1 to Numleds / 2 - 1
37.     rb_Shiftright 0 , Numleds / 2 'shift to the
        right
38.     rb_Shiftleft Numleds / 2 , Numleds / 2 'shift to the
        left all leds except the last one
39.     Waitms 1000
40.     RB_Send
41.   Next
42.   For n = 1 to Numleds/2 - 1
43.     rb_Shiftleft 0 , Numleds / 2 'shift to the
        left all leds except the last one
44.     rb_Shiftright Numleds / 2 , Numleds / 2 'shift to the
        right
45.     Waitms 1000
46.     RB_Send
47.   Next
48.   waitms 500 'wait a bit
49. Loop

```

## ATTiny13

1. '( 1312-WS2812-T13, Test für max. 12 RGB-Led
2. BENUTZUNG AUF EIGENE GEFAHR
- 3.
4. Achtung : Stromquelle Vor Oder Gleichzeitig Mit Der Datenleitung Anlegen!
5. Zur Sicherheit wird ein 10k in Reihe mit der Datenleitung Din geschaltet.
6. Das Rgbfeld wird mit Daten gefüllt,
- 7.
8.  $T0H=250-550 + T0L=700-1000 / T1H=650-950 + T1L=300-600$
9. bedingte Kompilierung für das Maschinensprache-Sub:
10. MHz= 8 8,0MHz -> 500ns+1000ns/ 875ns+500ns
11. 9,6MHz -> 417ns+ 833ns/ 729ns+417ns
12. MHz=12 12 MHz -> 500ns+1000ns/ 916ns+500ns

```
13.          16 MHz -> 375ns+ 750ns/ 688ns+375ns
14. #####
#####
15. ')
16. Const Mhz = 8           ' 8-10,7MHz 4+ 8/ 7+4
   Cyclen
17. 'Const Mhz = 12        '12-16,9MHz 6+12/11+6
   Cyclen
18.
19. '$PROG &HFF,&H7A,&HFF,&H00' generated. Take care that the chip supports
   all fuse bytes.
20. $regfile = "attiny13.dat"
21. $crystal = 9600000
22. $hwstack = 8
23. $swstack = 4
24. $framesize = 8
25.
26. Open "comb.4:9600,8,n,1,inverted" For Input As #2           'B.4 <- com-
   Daten
27. Ddrb = 1           'B.0 = &18 , 0
28.
29. Declare Sub Send           'asm-Subroutine
30. Dim Rgbfeld(36) As Byte           'max.12Led -> r0
31. Dim Bytmax As Byte : Bytmax = 36           '3*rgb-led -> r17
32. Dim Inbyte As Byte           'vom com-Port
33. Dim Index As Byte           'wie der Name sagt
34.
35. Do           'Main-Loop
36. 'aus Dta1 lesen
37. '  Restore Dta1 : For Index = 1 To Bytmax : Read Rgbfeld(index) : Next
   Index
38. 'aus com-Port lesen
39. '  Inbyte = Inkey(#2) : If Inbyte > 0 Then : Rgbfeld(index) = Inbyte :
   If Index >= Bytmax Then Index = 1 Else Incr Index : End If
40. 'Zufallsmuser ausgeben
41.   For Index = 1 To Bytmax : Rgbfeld(index) = Rnd(15) : Next Index :
   Send : Wait 1
42. 'Ampel
43. '  Restore Dta2 : For Index = 1 To Bytmax : Read Rgbfeld(index) : Next
   Index : Send : Wait 5           'grün
44. '  Restore Dta3 : For Index = 1 To Bytmax : Read Rgbfeld(index) : Next
   Index : Send : Wait 2           'gelb
45. '  Restore Dta4 : For Index = 1 To Bytmax : Read Rgbfeld(index) : Next
   Index : Send : Wait 5           'rot
46. '  Restore Dta5 : For Index = 1 To Bytmax : Read Rgbfeld(index) : Next
   Index : Send : Wait 2           'rotgelb
47. Loop
48. '#####
#####
```

```

49. Sub Send
50.   push r17                '2
51.   push r16                '2
52.   push r0                 '2
53.   Loadadr Bytmax , X     'Bytmax holen
54.   ld r17,x                '1
55.   Loadadr Rgbfeld(1) , X 'Feldadr holen
56. Nextbyte:
57.   ld r0,x+                '2
58.   ldi r16,8               '1
59. Nextbit:
60. '   T0H=250-550 + T0L=700-1000 / T1H=650-950 + T1L=300-600
61. '       62,5-137,5      87,5-125      92,9-135,7      75-150
62. '4+8/7+4 -----
63. #if Mhz = 8                '8 MHz-10,76 MHz
64.   Sbi $18 , 0              '2 <- 2 2
65.   lsl r0                   '1 1 1
66.   brcs Bit1                '1/2 1 2
67. Bit0:
68.   cbi $18 , 0              '2 <- 2
69. Bit1:
70.   Nop                       '1 1 1 *
   ggf. für M8/8M entfernen
71.   dec r16                   '1 1 1
72.   cbi $18 , 0              '2 <- 2 2
73.   brne Nextbit            '1/2 2 2
74. #endif
75. '   41,7-91,7  58,3-83,3  59,1-86,4  50-100
76. '6+12/11+6 -----
77. #if Mhz <> 8                '12 MHz-16,92 MHz
78.   Sbi $18 , 0              '2 <- 2 2
79.   nop                       '1 1 1
80.   Nop                       '1 1 1
81.   lsl r0                   '1 1 1
82.   brcs Bit1                '1/2 1 2
83. Bit0:
84.   cbi $18 , 0              '2 <- 2
85. Bit1:
86.   Nop                       '1 1 1
87.   Nop                       '1 1 1
88.   Nop                       '1 1 1
89.   dec r16                   '1 1 1
90.   cbi $18 , 0              '2 <- 2 2
91.   Nop                       '1 1 1
92.   nop                       '1 1 1
93.   brne Nextbit            '1/2 2 2
94. #endif
95. '-----
   -----
96. Waitus 5
97.   dec r17                   '1
    
```

```
98.    cpi r17,0                '1
99.    brne Nextbyte          '1/2
100.  ' ---
101.    pop r0                 '2
102.    pop r16               '2
103.    pop r17               '2
104.  End Sub
105.  '#####
    #####
106.  Dta1:
107.  '    grün    rot    blau
108.  Data &H0F , &H00 , &H00    'grün
109.  Data &H00 , &H0F , &H00    'rot
110.  Data &H00 , &H00 , &H0F    'blau
111.  Data &H0F , &H0F , &H00    'gelb
112.  Data &H0F , &H00 , &H0F    'cyan
113.  Data &H00 , &H0F , &H0F    'magenta
114.  Data &H0F , &H0F , &H0F    'weiss
115.  Data &H00 , &H00 , &H00    'schwarz
116.  Data &H0F , &H00 , &H00    'grün
117.  Data &H00 , &H0F , &H00    'rot
118.  Data &H00 , &H00 , &H0F    'blau
119.  Data &H0F , &H0F , &H00    'gelb
120.  Data &H0F , &H00 , &H0F    'cyan
121.  Data &H00 , &H0F , &H0F    'magenta
122.  Data &H0F , &H0F , &H0F    'weiss
123.  Data &H00 , &H00 , &H00    'schwarz
124.
125.  Dta2:
126.  '    grün    rot    blau
127.  Data &H0F , &H00 , &H00    'grün
128.  Data &H00 , &H00 , &H00
129.  Data &H00 , &H00 , &H00
130.  Data &H00 , &H00 , &H00
131.  Data &H00 , &H00 , &H00
132.  Data &H00 , &H0F , &H00    'rot
133.  Data &H00 , &H00 , &H00
134.  Data &H00 , &H00 , &H00
135.  Data &H00 , &H00 , &H00
136.  Data &H00 , &H0F , &H00    'rot
137.
138.  Dta3:
139.  '    grün    rot    blau
140.  Data &H00 , &H00 , &H00
141.  Data &H0F , &H0F , &H00    'gelb
142.  Data &H00 , &H00 , &H00
143.  Data &H00 , &H00 , &H00
144.  Data &H00 , &H00 , &H00
145.  Data &H00 , &H0F , &H00    'rot
```

```
146. Data &H00 , &H00 , &H00
147. Data &H00 , &H00 , &H00
148. Data &H0F , &H0F , &H00      'gelb
149. Data &H00 , &H0F , &H00      'rot
150.
151. Dta4:
152. '    grün    rot    blau
153. Data &H00 , &H00 , &H00
154. Data &H00 , &H00 , &H00
155. Data &H00 , &H0F , &H00      'rot
156. Data &H00 , &H00 , &H00
157. Data &H0F , &H00 , &H00      'grün
158. Data &H00 , &H00 , &H00
159. Data &H00 , &H00 , &H00
160. Data &H0F , &H00 , &H00      'grün
161. Data &H00 , &H00 , &H00
162. Data &H00 , &H00 , &H00
163.
164. Dta5:
165. '    grün    rot    blau
166. Data &H00 , &H00 , &H00
167. Data &H0F , &H0F , &H00      'gelb
168. Data &H00 , &H0F , &H00      'rot
169. Data &H00 , &H00 , &H00
170. Data &H00 , &H00 , &H00
171. Data &H00 , &H0F , &H00      'rot
172. Data &H00 , &H00 , &H00
173. Data &H00 , &H00 , &H00
174. Data &H0F , &H0F , &H00      'gelb
175. Data &H00 , &H00 , &H00
176.
177. ' ENDE
```

From: <https://wiki.modellbahn-anlage.de/> - Wiki der Modellbahn-Anlage.de

Permanent link: <https://wiki.modellbahn-anlage.de/elektronik/bascom/bascom-ws2812-knigh-rider>

Last update: 20.10.2024 00:05

