|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 |
| 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 |
| 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 | 31 | 30 |
| 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 |
| 59 | 58 | 57 | 56 | 55 | 54 | 53 | 52 | 51 | 50 |
| 60 | 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 |
| 79 | 78 | 77 | 76 | 75 | 74 | 73 | 72 | 71 | 70 |
| 80 | 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 |
| 99 | 98 | 97 | 96 | 95 | 94 | 93 | 92 | 91 | 90 |

**IR Sensor Layout = LED Analog Layout (leds[x] , IRWert[x])**

**(leds[x]=…) x∈ [0,99]**

**If (IRWert[x] == …)** **x∈ [0,99]**

*int IRWert[]* wird standardmäßig zu beginn jedes loop() Durchlaufes aktualisiert.

**LED XY Layout ( leds[XY[x,y]], IRWert[XY[x,y]])**

Die Funktion int XY(x,y) gibt einen integer zurück, welcher die Analoge Position des Feldes (x,y) Beschreibt

**(leds[XY[x,y]]=…)**

**If (IRWert[XY[x,y]] == …)** **x,y∈ [0,9]**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 |
| 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 |
| 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 | 31 | 30 |
| 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 |
| 59 | 58 | 57 | 56 | 55 | 54 | 53 | 52 | 51 | 50 |
| 60 | 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 |
| 79 | 78 | 77 | 76 | 75 | 74 | 73 | 72 | 71 | 70 |
| 80 | 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 |
| 99 | 98 | 97 | 96 | 95 | 94 | 93 | 92 | 91 | 90 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0,0 | 1,0 | 2,0 | 3,0 | 4,0 | 5,0 | 6,0 | 7,0 | 8,0 | 9,0 |
| 0,1 |  |  |  |  |  |  |  |  |  |
| 0,2 |  |  |  |  |  |  |  |  |  |
| 0,3 |  |  |  |  |  |  |  |  |  |
| 0,4 |  |  |  |  |  |  |  |  |  |
| 0,5 |  |  |  |  |  |  |  |  |  |
| 0,6 |  |  |  |  |  |  |  |  |  |
| 0,7 |  |  |  |  |  |  |  |  |  |
| 0,8 |  |  |  |  |  |  |  |  |  |
| 0,9 |  |  |  |  |  |  |  |  |  |

**Beispiel zur Nutzung des Tisches (Analog):**

*void startAnimation(){*

#Setze alle LEDs auf Rot

#Setze Wert der LED p

#Aktualisiere Bild auf Tisch mit Array leds[]

*for (int p=0;p< NUM\_LEDS;p++){*

*leds[p] = CRGB::Red;*

*}*

*FastLED.show();*

*}*

**Beispiel 2 zur Nutzung des Tisches mit XY:**

# setze LED an den Koordinaten 3,4 auf Rot

# setze LED an den Koordinaten 7,4 auf rgb Werte (r,g,b ∈ [0,255])

# int brightness ∈ [0,255]

#Aktualisiere Bild auf Tisch mit Array leds[]

*leds[XY(3,4)] = CRGB::Red;*

*leds[XY(2,7)] = CRGB( r, g, b);*

*FastLED.setBrightness(brightness);*

*FastLED.show();*

**Beispiel 3; Nutzung mit Inputs:**

*void test(){* #Feld leuchtet wenn Feld abgedeckt

*for ( int a = 0 ; a <= 100; a++ ) {*

*if(IRWert[a] > grenzwert){*

*int grenzwert* ist die Variable, welche zu Programmbeginn festgelegt wird und für den wert, ab dem ein Input erkannt werden soll steht.

*leds[a] = CRGB::Red;*

*IRTimer[a] = 5;*

*}else if(IRTimer[a] > 0){*

*leds[a] = CRGB::Red;*

*IRTimer[a] -= 1;*

*}else{*

*leds[a] = CRGB::Black;*

*}*

*}*

FastLED.show();

}

*(Hinweis: IRTimer ist hier ein int array welches die aktive Zeit des Feldes verlängert um ungewünschtes Flackern zu vermeiden. Dies ist für einfache Anwendungen und Input Prüfung in der Regel nicht notwendig und kann daher ignoriert werden.)*