

### Wdh.: Farbwerte einstellen

- Knallrot (1111 1000 0000 0000) → 0xF800, im 8-Bit Modus 0xE0
- Knallgrün (0000 0111 1110 0000) → 0x07E0, im 8-Bit Modus 0x1C
- Knallblau (0000 0000 0001 1111) → 0x1F, im 8-Bit Modus 0x03
- Schwarz → 0x0
- Weiß → 0xFFFF, im 8-Bit Modus 0xFF

### Wdh.: Kommando Sequenz zur Definition eines Ausgabefensters

1. Einleitungssequenz mit Kommando  
0xEF, 0x08 (bzw. 0xEF08, wird in 8 Bit Werte aufgeteilt)
2. Setzen der Ausgaberrichtung  
Hochformat: 0x1800, gedreht (180°): 0x1803  
Querformat: 0x1805, gedreht (180°): 0x1806
3. Festlegen der  $x_1$ -Koordinate des ersten Punktes  
0x12 < $x_1$ >
4. Festlegen der  $x_2$ -Koordinate des letzten Punktes  
0x15 < $x_2$ >
5. Festlegen der  $y_1$ -Koordinate des ersten Punktes  
0x13 < $y_1$ >
6. Festlegen der  $y_2$ -Koordinate des letzten Punktes  
0x16 < $y_2$ >

### Beispiel zur Displayprogrammierung

Ein Quadrat aus 18 x 18 Pixeln soll auf dem Display in den vier möglichen Ausgaberrichtungen dargestellt werden. Das Muster ist dabei 6 Zeilen blau, 6 Zeilen rot und 6 Zeilen grün. Der Abstand vom Displayrand soll 50 Pixel links (51. Pixel als 1. Pixel des Quadrats) und 25 Pixel oben betragen.

- a) Hochformat

Berechnung der Koordinaten:

$$x_1 = 50 \rightarrow 0x32$$

$$x_2 = x_1 + 18 - 1 = 67 \rightarrow 0x43$$

$$y_1 = 25 \rightarrow 0x19$$

$$y_2 = y_1 + 18 - 1 = 42 \rightarrow 0x2A$$

zu übertragende Kommandosequenz:

0xEF08, 0x1800, 0x1232, 0x1543, 0x1319, 0x162A

Hinweise:

- zum Überschreiben muss das Display komplett überschrieben werden
- Hintergrundfarbe muss vorher erstellt werden
- Farbwerte werden durch Schleifen übergeben (Bildaten z.B. in Array)
- Orientierung der Pixel hat keinen Einfluss auf die tatsächlichen Nummern der Pixel

b) Hochformat 180° gedreht

Berechnung der Koordinaten:

$$x_1 = x_{\max} - x_1' = 131 - 50 = 81 \rightarrow 0x51$$

$$x_2 = x_1 - (18 - 1) = 64 \rightarrow 0x40$$

$$y_1 = y_{\max} - y_1' = 175 - 25 = 150 \rightarrow 0x96$$

$$y_2 = y_1 - (18 - 1) = 133 \rightarrow 0x85$$

zu übertragende Kommandosequenz:

0xEF08, 0x1803, 0x1251, 0x1540, 0x1396, 0x1685

c) Querformat

Berechnung der Koordinaten:

$$x_1 = x_{\max} - y_1' = 131 - 25 = 106 \rightarrow 0x6A$$

$$x_2 = x_1 - (18 - 1) = 89 \rightarrow 0x59$$

$$y_1 = x_1' = 50 \rightarrow 0x32$$

$$y_2 = y_1 + (18 - 1) = 67 \rightarrow 0x43$$

zu übertragende Kommandosequenz:

0xEF08, 0x1805, 0x1264, 0x1559, 0x1332, 0x1643

d) Querformat 180° gedreht

Berechnung der Koordinaten:

$$x_1 = y_1' = 25 \rightarrow 0x19$$

$$x_2 = x_1 + (18 - 1) = 42 \rightarrow 0x2A$$

$$y_1 = y_{\max} - x_1' = 175 - 50 = 125 \rightarrow 0x7D$$

$$y_2 = y_1 - (18 - 1) = 108 \rightarrow 0x6C$$

zu übertragende Kommando­sequenz:  
0xEF08, 0x1806, 0x1219, 0x152A, 0x137D, 0x166C

### Übungsaufgabe

Ein Fensterausschnitt aus 40 x 20 Pixeln soll vom linken Rand 30 Pixel und vom oberen Rand 10 Pixel entfernt beginnen. Bestimmen Sie die Kommando­sequenz, die zum Display übertragen werden muss für alle Orientierungen.

a) Hochformat

zu übertragende Kommando­sequenz:  
0xEF08, 0x1800, 0x121E, 0x1545, 0x130A, 0x161D

b) Hochformat 180° gedreht

zu übertragende Kommando­sequenz:  
0xEF08, 0x1800, 0x1265, 0x153E, 0x13A5, 0x1692

c) Querformat

zu übertragende Kommando­sequenz:  
0xEF08, 0x1800, 0x1279, 0x1566, 0x131E, 0x1645

d) Hochformat 180° gedreht

zu übertragende Kommando­sequenz:  
0xEF08, 0x1800, 0x120A, 0x151D, 0x1391, 0x166A

### Einfache Möglichkeiten zur Komprimierung von Grafiken

a) Komprimierung per Farbtabelle

Ein zu übertragendes Bild bestehe aus 64 Farben. Die Werte der Farben werden in der Farbtabelle abgelegt.

const unsigned int Farbtabelle[64] = {0xF13E, 0x2CAB, ...} → 2<sup>6</sup> Farbwerte (6 Bit)  
Index enthält 6 Bit Farbwert

Vollbild: 132 x 176 Pixel x 6 Bit = 139332 Bit / 8 / 1024 = 17 kByte

b) Komprimierung bei Farbwiederholungen

Einfache Speicherung: 0x00, 0x00, 0x00, 0x00, 0x1C

Komprimierung überträgt den sich wiederholenden Farbwert doppelt, danach folgt die Zahl der Wiederholungen (ohne die bereits gesendete Doppelsequenz). Der Folgewert enthält den Farbwert für das erste Pixel nach der Wiederholung.

Komprimierung: 0x00, 0x00, 0x02, 0x1C

Praxisbeispiel: 276-mal schwarz übertragen

Sequenz: 0x00, 0x00, 0xFF, 0x00, 0x00, 0x11