Matrixanzeige im Multiplexbetrieb

Technologieschema:

PC\_0

Mikrocontroller

PC\_1

PC\_2

PC\_3

PC\_4

PC\_5

PC\_6

PC\_7

L293D

PB\_0

PB\_1

PB\_2

PB\_3

PB\_4

PB\_5

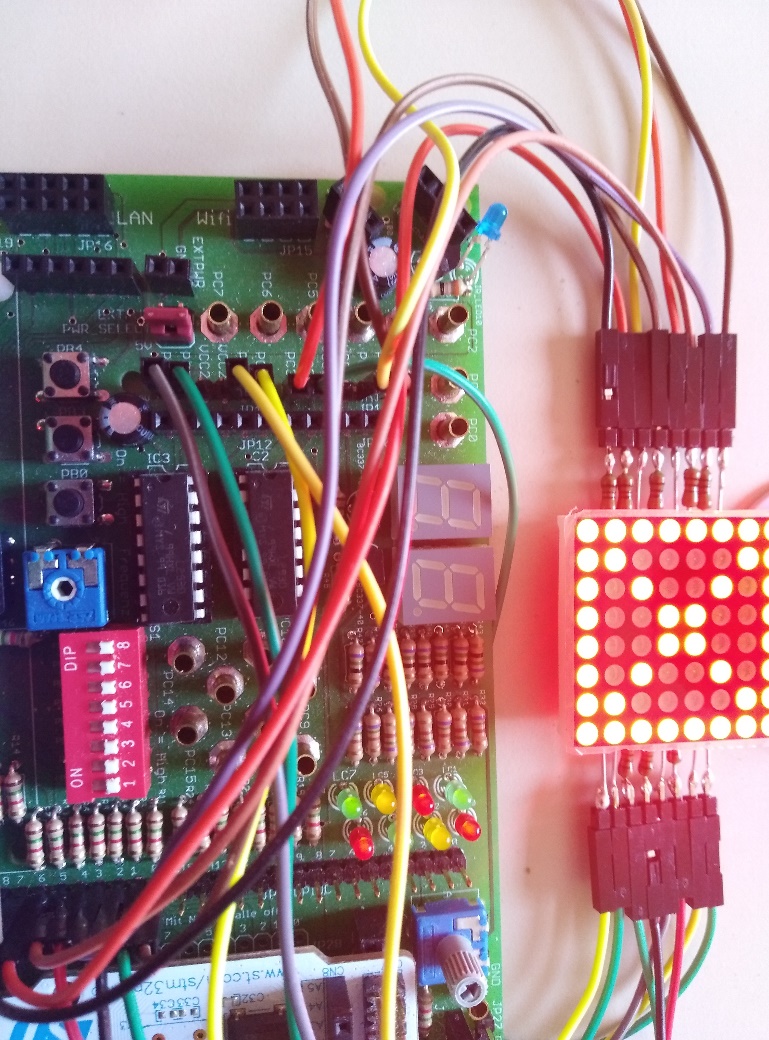
PB\_6

PB\_7

8x100Ω

**3MM / 8×8 RED DOT MATRIX**  
**A-1088BS**

Pin-Nr am Display



Brücke

PortCmit

L293D

PortB

100Ω

Ansteuerung:

PC\_0

Mikrocontroller

1=PC\_1

PC\_2

PC\_3

PC\_4

PC\_5

PC\_6

PC\_7

L293D

PB\_0

PB\_1

PB\_2=0

PB\_3

PB\_4

PB\_5

PB\_6

PB\_7

8x100Ω

**3MM / 8×8 RED DOT MATRIX**  
**A-1088BS**

Um eine LED ein zu schalten muss an der betreffenden Zeile eine 1 und an der betreffenden Spalte eine 0 ausgegeben werden. Beispiel: PC\_1=1 und PB\_2=0 bewirkt, dass in der 2. Zeile und in der 3. Spalte die LED leuchtet.

Aufgaben: Die Aufgaben sollen unter Verwendung eines Tickers gelöst werden.

1. Lauflicht in Zeile 1
2. Ein Balken soll von oben nach unten wiederholt durchlaufen
   1. Anzeigedauer pro Balken 1s
   2. Anzeigedauer pro Balken 1ms

Welcher optische Eindruck entsteht bei der Anzeigedauer 1ms

1. Muster ausgeben: Legen Sie ein array int muster[8] an, in dem Sie zeilenweise die LEDs angeben die leuchten sollen. Vervollständigen Sie zunächst die Tabelle.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Zeile 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0xFF |
| Zeile 2 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0xC3 |
| Zeile 3 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0xE7 |
| Zeile 4 |  |  |  |  |  |  |  |  |  |
| Zeile 5 |  |  |  |  |  |  |  |  |  |
| Zeile 6 |  |  |  |  |  |  |  |  |  |
| Zeile 7 |  |  |  |  |  |  |  |  |  |
| Zeile 8 |  |  |  |  |  |  |  |  |  |

Der optische Eindruck dieser Figur entsteht, indem Sie, wie bei den Balken in 2.b. die Anzeigedauer für eine Zeile auf 1ms setzen.

1. Musterwechsel alle 0,5s

Erweitern Sie das Programm aus 3. um einen weiteren Ticker, der alle 0,2s ein neues Muster auswählt. Die Muster können in einem 2-dimensionalen array int muster[5][8] abgebildet werden:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Muster | Zeile i=0 | Zeile i=1 | Zeile i=2 | Zeile i=3 | Zeile i=4 | Zeile i=5 | Zeile i=6 | Zeile i=7 |
| j=0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| j=1 | 0 | 0 | 0 | 0x18 | 0x18 | 0 | 0 | 0 |
| j=2 | 0 | 0 | 0x24 | 0x18 | 0x18 | 0x24 | 0 | 0 |
| j=3 | 0 | 0x42 | 0x24 | 0x18 | 0x18 | 0x24 | 0x42 | 0 |
| j=4 | 0xFF | 0xC3 | 0xA5 | 0x99 | 0x99 | 0xA5 | 0xC3 | 0xFF |

Lösung:

PortOut reihen(PortC,0xFF);

PortOut spalten(PortB,0xFF);

Ticker ticker;

Ticker ticker2;

int muster[5][8]={{0,0,0,0,0,0,0,0},

{0,0, 0x0, 0x18, 0x18, 0, 0, 0},

{0,0, 0x24, 0x18, 0x18, 0x24, 0, 0},

{0,0x42, 0x24, 0x18, 0x18, 0x24, 0x42, 0},

{0xFF,0xC3, 0xA5, 0x99, 0x99, 0xA5, 0xC3, 0xFF}};

int i=0,j=0;

void anzeigen()

{

i=(i+1)%8;

reihen=1<<i;

spalten=~muster[j][i];

}

void naechstesMuster()

{

j=(j+1)%5;

}

int main()

{

ticker2.attach(&naechstesMuster,0.2);

ticker.attach(&anzeigen,0.001);

while (true) {

}

}