Warptrix

Projektverwalter: Nick

Beschreibung

Das Projekt WarpTrix besteht aus zwei großen LED-Panels die ehemalig zu einer Bahnhofsanzeige gehörten. Beide Panels bestehen aus 20 kleineren Modulen, welche jeweils in 2 Reihen zu zehnt angeordnet sind. Jedes Modul besteht aus 16×24 LEDs. 24 zeilen zu 16 Spalten. Die einzelnen Module sind 1:8 gemultiplext, besitzen somit 8 Zeilentreiber und 6 8-Bit Shiftregister. In einem Panel sind jeweils 10 Module in einer Reihe zu einer Daisychain verbunden. Die Shiftreister werden mittels SPI mit einem STM32F7 Discovery Board angesteuert, wobei zwei Reihen an Modulen mit zwei SPIs parallel angesteuert werden. Es können entweder beide Panels zusammen betrieben werden um eine Auflösung von 160px*96px zu erreichen oder einzeln mit einer Auflösung von 160px*48px.

Git-repo mit Code für den STM32 sowie ausführliche Dokumentation zur Ansteuerung auf Hardwareebene folgt später.

Aktuell hängt eins der beiden Panel mit einem weißen Diffusor in der E-Werkstatt mit einer Fernseh-halterung über der Werkzeugwand.

Der STM32F7 hat die IP 10.0.0.68 bzw. den Hostname warptrix.warpzone und kann mit TCP auf dem Port 1236 angesprochen werden. Mittels softpwm lassen sich auf der Matrix bis zu 16 Graustufen (oder eher Gelbstufen :D) + schwarz Anzeigen.

Wie beuge ich die Warptrix meinem Willen?

Die Warptrix hat die IP 10.0.0.68 und den Hostname warptrix.warpzone. Angesprochen wird die Warptrix per TCP auf den port 1236. Es ist nur eine TCP Verbindung gleichzeitig möglich. Werden 5-10 Sekunden keine neuen Daten gesendet, gibts einen Timeout.

Es gibt zwei Befehle die gesendet werden können:

mode{framerate}{subframes}{use_both_panels}
wobei framerate, subframe und use_both_panels jeweils ein Byte sind. Der mode Befehl wird zu Beginn der Übertragung geschickt und legt die maximale Framerate, die Anzahl der Graustufen und die Konfiguration der Panel fest.

Der Parameter use_both_panels muss aktuell noch auf 1 gesetzt werden, sonst wird kein Bild angezeigt. Will fix later :P

Die maximale Framerate bestimmt die Framerate die maximal erreicht wird. Ist dein Rechner zu langsam, die Übertragung zu schlecht oder die subframe Anzahl zu hoch, kann die echte Framerate auch geringer sein.

Der Parameter subframes legt fest, in wieviele kleinere Frames eine ganzer Frame zerlegt wird. Wird subframes mit 10 angegeben, so wird ein Frame in 10 subframes zerlegt und nacheinander angezeigt. Es werden dabei von Subframe zu Subframe die LEDs ihrem Graustufenwert angeschaltet, wodurch eine art PWM erzeugt wird. Desto mehr Subframes, desto mehr Graustufen können somit erreicht werden, wodurch jedoch die insgesammte Framerate sinkt. Mit subframes = 10 ist die maximal erreichbare Framerate somit ein zehntel derer die mit subframes = 1 möglich ist. Ab subframes = 14 beginnt das Panel sichtbar zu flakern.

start{number_of_bytes}{frame}
number_of_bytes ist die Anzahl an Bytes die der Frame hat. frame ist das eigentliche Bild das angezeigt werden soll.

Jeder Pixel kann 16 Graustufen, wird also mit 4 Bit bzw. einem Nibble angesteuert. Ein Byte (bestehend aus high- und low-Nibble) steuert somit 2 Pixel an, die in x-Richtung nebeneinander liegen. Da das Panel 160*48 Pixel hat, werden somit um alle Pixel anzusteuern 80*48 Bytes benötigt, also X-Auflösung/2. Das low-Nibble steht dabei für den linken, das high-Nibble für den rechten Pixel. In Hexadezimal ausgedrückt: 0x2A gibt dem linken Pixel den Graustufenwert 2, dem rechten Pixel den Graustufenwert 10.

Wie füge zwei Pixel in ein Byte zusammen?

pixelAB = 0x00 # pixelAB ist ein uint_8
pixelA = 0x2 # linker Pixel mit Grauwert 2
pixelB = 0xA # rechter Pixel mit Grauwert 10
pixelAB = pixelA | (pixelB « 4) # Beide pixel zusammen = 0x2A

Beispielcode

```
import socket
import sys

TCP_IP = sys.argv[1]
TCP_PORT = 1236

subframes = 12 # number of greyscale values possible other than black.
X_SIZE = 80 # half number of colums (160), as one byte (high-nibble and low-
nibble) controlls two leds
Y_SIZE = 48 # number of rows


mode = bytearray()
mode.extend('mode'.encode('latin-1'))
mode.append(20) #framerate
mode.append(subframes) #subframe nr
mode.append(1) #use both panels, or not.  NOTE: KEEP THIS VALUE 1 UNTIL
FURTHER NOTICE!
```

```
data_size =  X_SIZE*Y_SIZE

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect((TCP_IP, TCP_PORT))

s.send(mode)

for k in range(1000000):
    data1 = bytearray()
    data1.extend('start'.encode('latin-1'))
    data1.append(data_size >> 8)
    data1.append(data_size & 0xff)
    _data_size =  X_SIZE*Y_SIZE
    _data1 = [0 for x in range(data_size)]
    for j in range(Y_SIZE):
            for i in range(0,X_SIZE,8):
                _data1[j*X_SIZE+i+0] = 0x01
                _data1[j*X_SIZE+i+1] = 0x23
                _data1[j*X_SIZE+i+2] = 0x45
                _data1[j*X_SIZE+i+3] = 0x67
                _data1[j*X_SIZE+i+4] = 0x89
                _data1[j*X_SIZE+i+5] = 0xAB
                _data1[j*X_SIZE+i+6] = 0xCD
                _data1[j*X_SIZE+i+7] = 0xEF
    [data1.append(a) for a in _data1]
    s.send(data1)
    print("Frame {0} send.".format(k))
exit()
s.close()
```

From:
http://wiki.warpzone.ms/ - **warpzone**

Permanent link:
http://wiki.warpzone.ms/orga:dokumentationsteam_warptrixx-projektdoku?rev=1560440343

Last update: **13.06.2019**