

## ARDUINO - Hardwareerweiterungen fürs Schalten und "Lese

[Software](#) -> [ARDUINO - Hardwareerweiterungen fürs Schalten und "Lese](#)

gaulois

#1/8 Verfasst am: 03 Okt 2018 9:54 Titel: ARDUINO - Hardwareerweiterungen fürs Schalten und "Lese

Hallo zusammen,  
der Arduino kann zwar ein- und ausgeben, er ist aber für kleine Stromstärken angelegt.

Egal, was die Steuerung übernehmen soll, ist immer auch zu überlegen, welche Hardware-Erweiterungen man benötigt, wo es die gibt, was sie können und was sie nicht können.

Das ist eigentlich auch einen eigenen thread wert, denn das ist unabhängig vom Inhalt des Steuerungssystems.

Der Übersicht halber schlage ich daher vor, dass wir diese Dinge hier diskutieren. Dazu gehören nicht nur Relaiskarten, Optokoppler und ähnliches, sondern auch LED-Matrizen, Bildschirme und Touchscreens, Entfernungsmesser, Hall-Sensoren, Reed-Kontakte und andere externe Geräte, auf die ich zurzeit nicht komme.

gaulois

#2/8 Verfasst am: 03 Okt 2018 10:21 Titel: Ansteuerung eines Bildschirms

... ich fang mal inhaltlich mit einem Thema an, das ich kürzlich hatte, das ist die Ansteuerung einer großen Reklametafel (1,8" Bildschirm) zur Darstellung von Bildern (Text geht natürlich auch). Da habe ich folgenden Code gewählt, um 5 aufeinanderfolgende Bilder darzustellen:

```
// include the necessary libraries
#include <SPI>
#include <SD>
#include <TFT> // Arduino LCD library
// #include <string> // brauchen wir nicht
// #include <stdio> // brauchen wir nicht
// #include <sstream> // brauchen wir nicht

// pin definition for the Uno and nano
#define sd_cs 4
#define lcd_cs 10
#define dc 9
#define rst 8

// pin definition for the Leonardo // brauchen wir nicht für den nano oder uno
// #define sd_cs 8
// #define lcd_cs 7
// #define dc 0
// #define rst 1

TFT TFTscreen = TFT(lcd_cs, dc, rst);

// const int resetPin = 5; // brauchen wir nur für externen Reset-Knopf
const int x = 0;
const int y = 0;
const int defaultDelay = 1500;
// const int AnzahlBilder = 5; // brauchen wir erst, wenn Dateinamen berechnet werden können
// int Lauf; // Zähler für die Anzahl Durchläufe, zur Kontrolle, wann das Programm "abstürzt"
// boolean ersterFehler; // Variable zur Feststellung, ob vorher schon ein Fehler vorlag

// this variable represents the image to be drawn on screen
PImage BildDatei;
```

```

void setup() {
// pinMode(resetPin, OUTPUT); // wird benötigt für externen Reset-Knopf
// digitalWrite(resetPin, HIGH);

// Lauf = 1;
// ersterFehler = true;

// initialize the serial port: it will be used to
// print some diagnostic info
Serial.begin(9600);
while (!Serial) {
// wait for serial port to connect. Needed for native USB port only
}

// clear the GLCD screen before starting
// initialize and clear the GLCD screen
TFTscreen.begin();
TFTscreen.background(31, 31, 31);
// try to access the SD card. If that fails (e.g.
// no card present), the setup process will stop.
Serial.print(F("Initializing SD card..."));
if (!SD.begin(sd_cs)) {
Serial.println(F("failed!"));
return;
}
Serial.println(F("OK!"));
}

void(* resetFunc) (void) = 0; //declare reset function @ address 0

void loop() {
Serial.println(F("drawing image"));
// char* BildName;
//
// for (int i = 1; i < AnzahlBilder + 1; i++){
//// BildName = sprintf("Bild", i, ".bmp");
//// BildName << "Bild" << i << ".bmp";
//
// BildName = "Bild1.bmp";
//
// BildLaden(BildName, x, y, defaultDelay);
// TFTscreen.println (F(BildName));
// }

if (BildLaden("Bild1.bmp", x, y, defaultDelay)){
if (BildLaden("Bild2.bmp", x, y, defaultDelay)){
if (BildLaden("Bild3.bmp", x, y, defaultDelay)){
if (BildLaden("Bild4.bmp", x, y, defaultDelay)){
if (BildLaden("Bild5.bmp", x, y, defaultDelay)){
}
}
}
}
}
}
// Lauf = Lauf + 1;
}

boolean BildLaden(char* inBild, int x, int y, int inDelay) {
BildDatei = TFTscreen.loadImage(inBild);
boolean BildValide = (BildDatei.isValid());
if (BildValide) {
TFTscreen.image(BildDatei, x, y);
}
else {
// if (ersterFehler) {
// ersterFehler = false;

```

```
// TFTscreen.background(0, 0, 0);
// Serial.println(F("error while loading arduino.bmp"));
// TFTscreen.println();
// TFTscreen.print(F("Fehler in Durchlauf "));
// TFTscreen.println(Lauf);
// TFTscreen.print(F("bei BILD: "));
// TFTscreen.println(inBild);

// digitalWrite(resetPin, LOW);
// resetFunc(); //call reset
// }
// }
// delay(inDelay);
// return BildValide;
}
```

Da sich der Bildschirm immer mal wieder "aufgehängt" hat und keinen Bildwechsel mehr vollzog, habe ich in dem Code auch eine Reset-Funktion. Berechnung der Bildnamen "Bild1", "Bild2", ... "BildX" ist mir leider nicht gelungen, dafür hatte ich eine Konstante, die die maximale Zahl der Bilder aufnimmt vorgesehen und wollte dann den Code so schreiben, dass der Dateiname zusammengesetzt wird aus dem Wort "Bild" und einer fortlaufenden Nummer. Hat nicht geklappt. Wenn ich das raus habe, melde ich mich, wenn es einer von Euch weiß, bitte ich um Info!

Der Code mag auch dazu verhelfen, sich mit verschiedenen Programmieretechniken in C zu beschäftigen, sodass man den Arduino strukturiert mit möglichst wenig Ressourcenaufwand arbeiten lässt.

gaulois

#3/8 Verfasst am: 03 Okt 2018 12:59 Titel:

... ein wichtiger Baustein für das Schalten mit dem Arduino sind Treiber, die größere Stromstärken vertragen als der Arduino selbst. Sonst kann man nur LEDs ansteuern, aber keine Motoren etc.

Der ULN2803A scheint der richtige Baustein für so was zu sein:

<http://www.datasheetcafe.com/uln2803a-datasheet-pdf-stmicro/>

kompakt und leistungsfähig.

joe

#4/8 Verfasst am: 03 Okt 2018 23:01 Titel:

Hallo Gerd,  
ich hätte mir gewünscht, du hättest ein Video mit dem 1,8" Oled Bildschirm eingestellt. So ist der Code nur graue Theorie, die ich in Ermangelung eines OLED-Displays nicht nachvollziehen kann.  
Für den Arduino gibt es Shields, z.B. für Servos (Weichenantriebe)  
<https://www.ebay.de/itm/PCA9685-16-CH-12-bit-PWM-Servo-Shield-Motor-Driver-I2C-Module-Robot-For-Arduino/182063251574?hash=item2a63d0c076:g:LboAAOSwRJxZf-XZ>  
Suchbegriff: 16ch servo shield (Ich habe den Suchbegriff extra angegeben, da Ebay nach Beendigung des Angebotes alternative oder völlig sinnbefreite Suchvorschläge anbietet)  
Motor shields mit L293 oder 298 (je nach Stromverbrauch des Motors) Suchbegriff: arduino motor shield

gaulois

#5/8 Verfasst am: 04 Okt 2018 20:49 Titel:

Hallo Jörg,  
da hast Du sicher Recht. Ich hatte in meinem Anlagenthread nur ein Foto gepostet, mehr nicht. Ich musste aber den Screen leider auch wieder zu Völkner zurückschicken und vorher war nicht genug Zeit, sich um ein Video zu kümmern.

Heute ist ein Neuer gekommen, mal sehen, ob der besser ist, und mal sehen, ob ich mit der bescheidenen Videokamera, die ich habe (feste Brennweite), etwas vorzeigbares zaubern kann, denn ich bin mir nicht sicher, ob die Bildwiederholfrequenz vom Screen zu Bildaufnahmefrequenz der Kamera passt.

Ich nehme die Anregung aber gerne auf und versuche's.

gaulois

#6/8 Verfasst am: 04 Okt 2018 22:21 Titel:

... so, der neue Bildschirm hatte ein kleines Problem mit dem SD-Slot (mechanisch), aber jetzt läuft er streifenfrei und ein wirklich grobes Testvideo ohne Text und Erklärungen gibt es hier:

<http://youtu.be/RENOqy2EIXs>

bzw.

<http://www.youtube.com/watch?v=RENOqy2EIXs&feature=youtu.be>

Zu sehen ist halt der kleine Bildschirm, wie er nacheinander verschiedene Bilder lädt (man kann dem Laden der Bilder zusehen, auch wenn der Bildschirm den SD-Slot auf dem Buckel hat, braucht er trotzdem einige Zeit dafür), was für eine Reklametafel ja so durchaus sein könnte. Dann habe ich das Teil mal gedreht, sodass man (bei der schlechten Beleuchtung 😊) mehr erahnen als sehen kann, wie wenig man für diese Anwendung braucht. Das ist ein Effekt, der auf der Anlage sicherlich gut rüberkommt, wenn man den Schirm irgendwo einbaut und dann da Bilder ablaufen lässt.

Für Filme ist er aber natürlich nicht geeignet, viel zu langsam...

gaulois

#7/8 Verfasst am: 04 Okt 2018 22:22 Titel:

wusste ich's doch, dass die Forensoftware den ersten Link nicht mag und verändert. Diesen bitte ignorieren und den zweiten nehmen, der läuft...

joe

#8/8 Verfasst am: 05 Okt 2018 6:16 Titel:

Hallo Gerd,  
das sieht klasse aus.