

HELLIGKEITREGELUNG

Diese Thematik scheint alleine schon ein Projekt für sich zu sein. So gibt es nach meinen Recherchen vier verschiedene Möglichkeiten diese Aufgabenstellung zu lösen:

- Die naheliegende Lösung wäre an sich diese hier: Es gibt in der UTFT.h den Befehl [LCD.setBrightness\(Wert\)](#); mit dem man die Helligkeit der Hintergrund-Beleuchtung dimmen kann. Das funktioniert aber nicht so ohne weiteres. Warum auch immer ?! Da gibt es eine Vielzahl von anderen Interessierten, die genau dieser Sache auch versuchen nachzugehen – und nicht recht weiterkommen. Denn offenbar ist nämlich zuvor noch nötig den TFT-Anzeige-Chip noch via Software umzuprogrammieren. Wenn man dort die Register-Adresse 0xBE anspricht, besteht dort die Möglichkeit die Grundeinstellung dafür zu aktivieren (enable) , dass eben eine Einstellung per Software möglich ist. → Ist mir aber noch nicht gelungen – wie vielen anderen Interessierten in den Foren zuvor auch nicht.
- Die zweite Möglichkeit ist die, dass der Chip und unsere TFT-Platine in der Lage sein sollen, die Funktion DBC anzuwenden. Damit ist eine ‚dynamische Helligkeitsregelung, abhängig von der Umgebungs-Helligkeit‘ gemeint. Bedarf dann aber auch vermutlich wieder der Umprogrammierung des TFT-Chips. Es gibt dazu eine Anleitung auf Seite 24 des PDF-Dokuments [SSD1961_2_3_Application_note_v1.7.pdf](#)
- Die dritte Möglichkeit ist die, Jumper auf der Rückseite zu nutzen. Ich sehe keine Jumper-Anschlüsse ! Allerdings ist es tatsächlich so, dass auf der Rückseite der Platine ein ausgewiesener Bereich ‚BrightnessControl‘ vorhanden ist. Und dort gibt es offenbar drei Möglichkeiten, wobei die erste Möglichkeit ‚LED_A‘ mit einem SMD-Widerstand belegt ist. Da könnte noch Potential für den Interessierten drin sein. Dazu sind dann aber die Unterlagen zum Display aus dem Internet zu studieren !!!
- Dann kommen wir zu der von mir favorisierten Methode, die ich von Beginn an verfolgt hatte. Dabei ist es so, dass ich zwar nicht die Hintergrundbeleuchtung steuere, sondern die Farben heller und dunkler setze. So ist beispielhaft der Wert (255,255,255) gleich einem grell-weiß. Und (0,0,0) ist total schwarz. Damit habe ich dann auch die gewünschte Display-Helligkeit.
Das lässt sich nun in der abschließenden Version mittels eines Poti's einstellen. Kleiner Schönheitsfehler: Die eigentliche Farbgebung des ‚BackLight‘ erfolgt im Setup-Teil – und wird damit im Laufe des Programms nicht mehr abgefragt. Im Gegenteil zu den Hintergrundfarben der einzelnen Zeichen. Die lassen sich auch ‚fließend‘ während des Betriebes ändern. – Ist dann aber nicht schlimm: Man kann ja die Uhr schlicht nochmals Aus- und Ein schalten.
Eine andere Möglichkeit wäre die: Über einen Mehrstufen-Schiebeschalter den Analog-Pin (A0) im 1. Fall gegen Vcc zu legen, im 2. Fall über einen 4,7kΩ-Widerstand mit Vcc verbinden, und im 3. Fall direkt mit GND zu verbinden. Das bringt drei Sensor-Werte, die dann mittels „IF - ELSE IF – ELSE“ Funktion mit unterschiedlichen Farb- und Helligkeitseinstellungen belegt werden können. So z.B. auch mit Grundtönen in orange und grün und blau (der Wert (135,95,0) ist ein schönes dunkles orange). DA werden dann zu Beginn des Programm-Durchlaufes ‚alle‘ Werte gleich gesetzt.
Um zunächst sogar den Schalter zu sparen, lässt sich das mit dem Widerstand auch problemlos auf dem kleinen Zusatz-Steckbrett realisieren. Je nach dem wo ich auf der Platine das A0-Kabel anschließe (Vcc, GND, oder Widerstand), bekomme ich nach einem Neustart (oder Reset) der Uhr (Aus-Ein) eine meine zuvor eingestellten Farben.
Diese Variation werde ich für mich beim späteren Gehäuse-Einbau nutzen. Dort werde ich dann einen kleinen 3-Stufen-Schiebeschalter dafür einsetzen. Das scheint mir am meisten ‚praxisgerecht‘ zu sein. Und selbst der Widerstand ließe sich sogar noch sparen. In dem Fall müsste ersatzweise die auf dem Shield zur Verfügung stehende zweite Spannung von 3,3V abgegriffen – und auf den ‚A0-Pin‘ geschaltet werden. Somit hätte ich dann die drei Zustände 5V, 3,3V und GND.
Aktuell, in der Version GPS-Uhr.ino, ist es aber noch das Poti. Auch bei der Version mit dem Widerstand ändert sich an dem Programm, oder der Verkabelung nichts (!).