

GIDE/USB-Interface für K1520-Bus

Prinzipiell aufgebaut nach <http://www.gaby.de/gide/> nur der RTC wurde nicht mit übernommen und die benötigten Signale werden am Bus und nicht an der CPU abgegriffen.

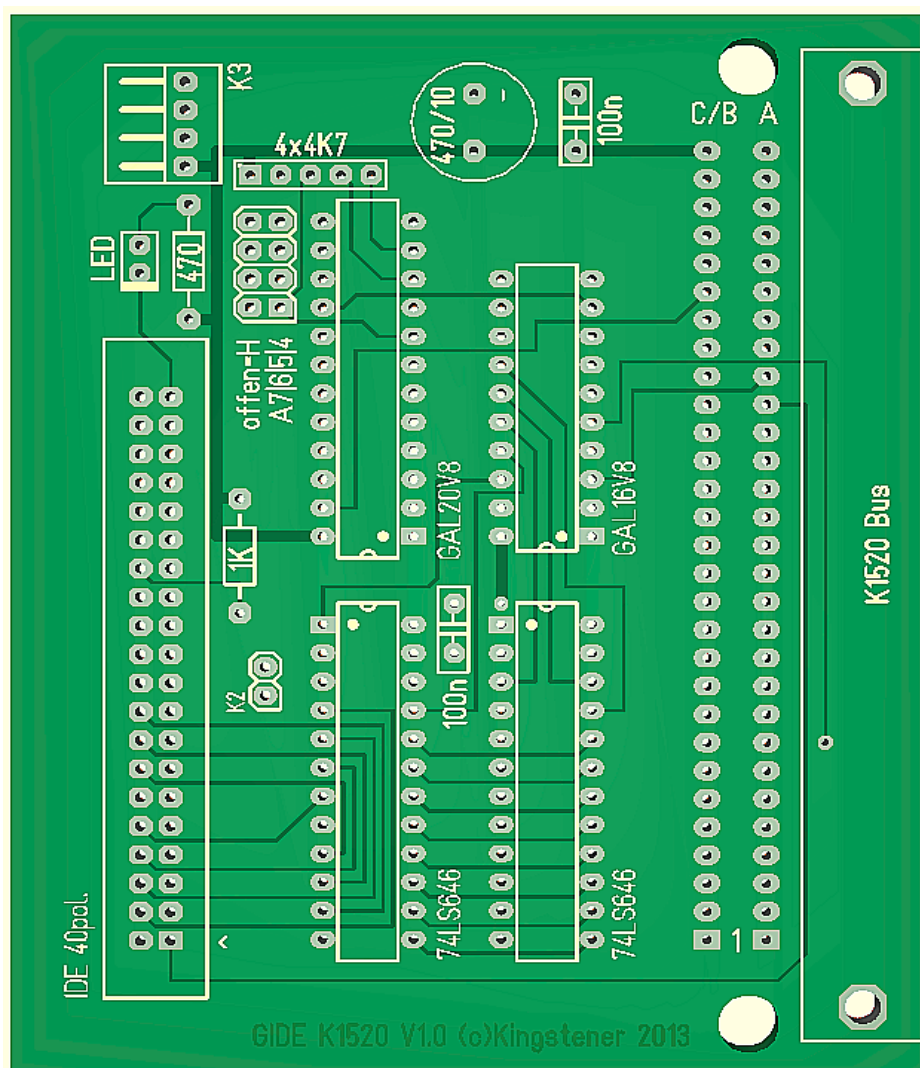
Das GIDEBIOS aus o.g. Quelle kann in das benutzte Betriebssystem integriert oder als nachladbarer Treiber verarbeitet werden.

Testweise wurde im BIOS des ROBOTRON - CP/A das GIDEBIOS in die originale BIOSHD.MAC und BIOSHDC.MAC implementiert (Geräteadresse D0h), allerdings adressiert das derzeitig da enthaltene BDOS nur 8MB. Da keine Unterverzeichnisse unterstützt werden ist mehr aber auch nicht unbedingt so sinnvoll, denn bei durchschnittlich 300-400 Dateien im Verzeichnis (theoretisch max. 1024) ist es so schon unübersichtlich genug, auch wenn man mit der umständlichen USER – Zuweisung etwas strukturieren kann. Alternativ könnte man evtl. mehrere Laufwerke einrichten. Das ist immer eine Frage der Notwendigkeit und hat eben entsprechenden Platzbedarf, was dann zu Lasten des freien TPA geht, der Treiber hier (als Anhang) ist ein Kompromiss dazu.



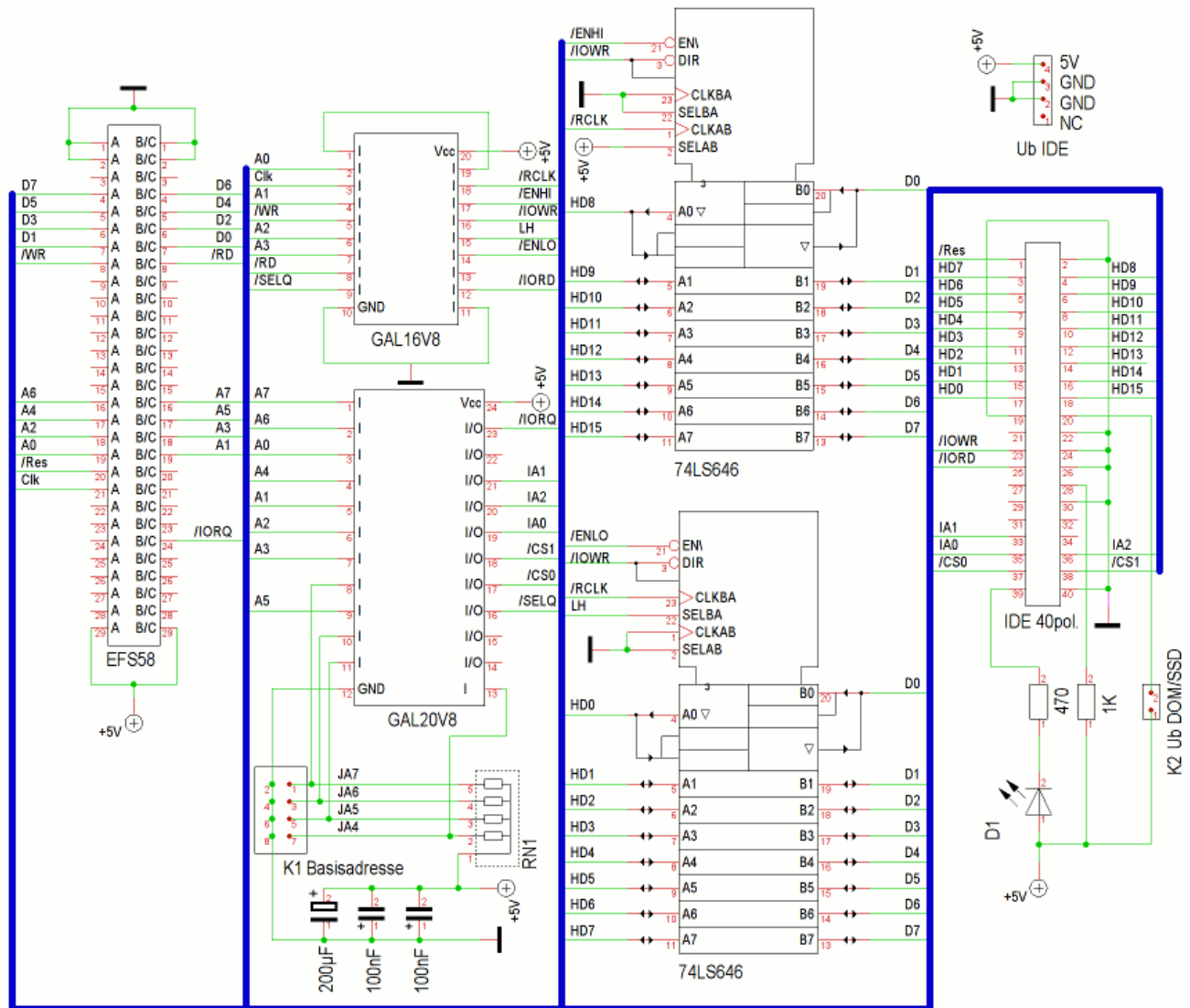
Das CCP wird (nur im implementierten Treiber!) auf die Festplatte ausgelagert (bringt reichlich 2KB mehr TPA), mit swap a: h: können die Laufwerke getauscht werden und danach kann die Startdisk entfernt werden, der Warmstart erfolgt dann von Festplatte.

GIDE- Platine/Bestückung



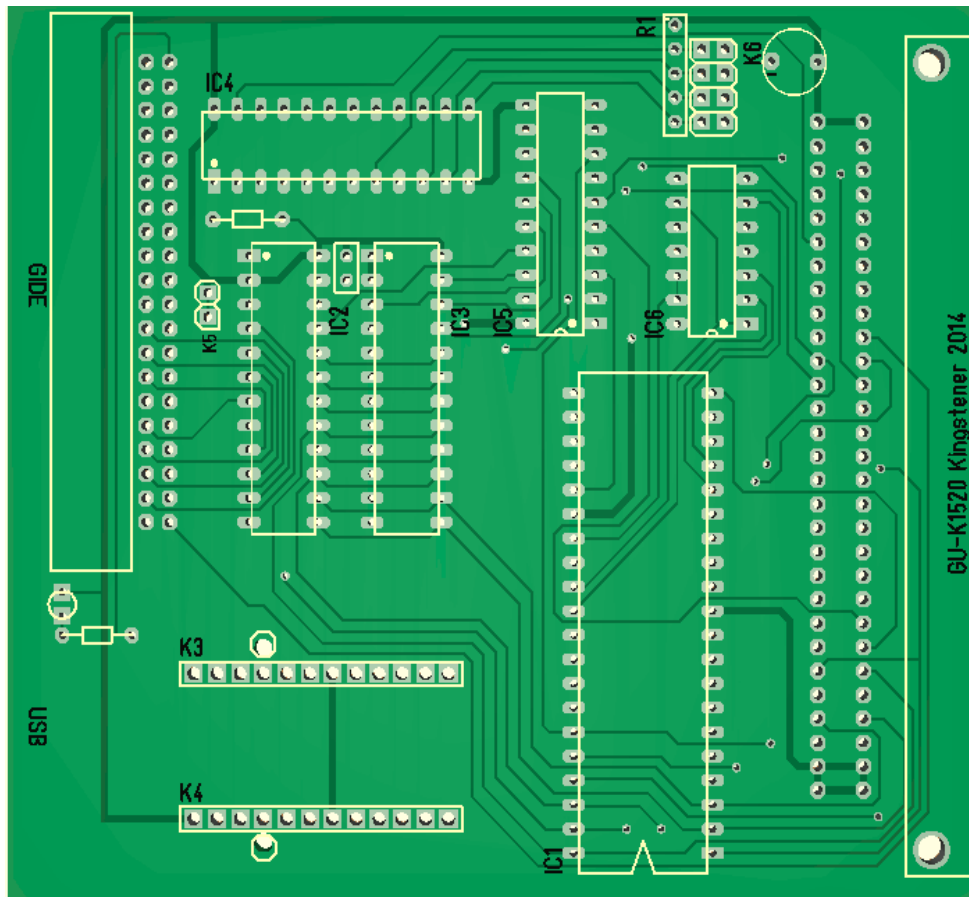
- K1 - Jumper zur Adressierung in Schritten zu 10h
 - von oben(RN GIDE, Platinenrand GUK) nach unten A4-A7
 - offen=1 geschlossen=0
- K2 - Jumper für 5V an Pin 20 IDE zur Spannungsversorgung bei DOM/SSD
 ACHTUNG!!Bei anderen Medien Jumper unbedingt offen lassen!!

Schaltplan GIDE - K1520

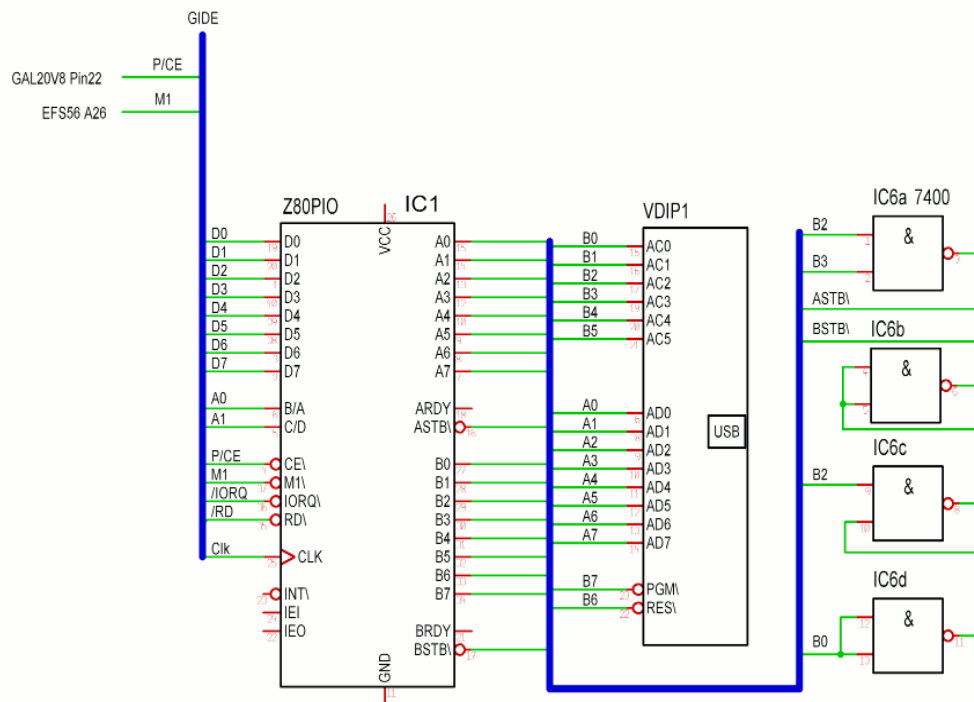


Eine Weiterentwicklung der Platine enthält zusätzlich einen USB-Port für externen Massenspeicher bzw. schnellen Datenaustausch. Die Adressierung erfolgt mit über den vorhandenen GAL20V8 und belegt die vier freien unteren Adressen des GIDE-Adressbereiches (USB X0-X3H und GIDE X6-XFH). Die Utools dafür gibt es unter <http://www.kc85.info>. In deren USBINC.Z80 werden also immer die PIO - Adressen X0h-X3h (X= am K6 eingestellte Kartenadresse) eingetragen und der GIDE Treiber wird ebenfalls mit X0h als Grundadresse kompiliert.

GU-K1520 – Platine



Plan der Erweiterung



Materialliste und Kalkulation (Stand 12/2014)

GU-K1520

1	GS-KO 14P	IC-Sockel, 14-polig, mit Abblockkondensator	0,65 €	0,65 €	Reichelt
1	GS 24P	IC-Sockel, 24-polig	0,31 €	0,31 €	Reichelt
1	GS-KO 20P	IC-Sockel, 20-polig, mit Abblockkondensator	0,95 €	0,95 €	Reichelt
1	GS-KO 24P-S	IC-Sockel, 24-polig, schmal mit Abblockkondensator	0,99 €	0,99 €	Reichelt
2	GS 24P-S	IC-Sockel, 24-polig, schmal	0,40 €	0,80 €	Reichelt
1	GS 40P	IC-Sockel, 40-polig	0,62 €	0,62 €	Reichelt
1	VF 470/16 K-F	Elektrolytkondensator 470µF /16V	0,33 €	0,33 €	Reichelt
2	X7R-2,5 100N	Vielschicht-Keramikkondensator 100N (RM2,5oder1206)	0,05 €	0,10 €	Reichelt
1	SIL 5-4 4,7K	Widerstands-Netzwerk, 4Wid./5Pins, 4,7 K-Ohm	0,06 €	0,06 €	Reichelt
2	SMD 1/4W 1,0K	SMD-Chip-Widerstand, Bauform 1206, 1,0 K-Ohm	0,10 €	0,20 €	Reichelt
1	WSL 40W	Wannenstecker, 40-polig, gewinkelt	0,30 €	0,30 €	Reichelt
1	MPE 087-2-008	Stiftleisten 2,54 mm, 2X04, gerade	0,19 €	0,19 €	Reichelt
1	JUMPER 2,54 SW	Kurzschlussbrücke, schwarz, RM 2,54	0,05 €	0,05 €	Reichelt
1	LED B-3	Led-Kunststoffblock mit 3mm LED	0,23 €	0,23 €	Reichelt
1	LS00	LS - Low Power Schottky, 74LS00 , DIL-14	0,79 €	0,79 €	Reichelt
1	Platine	bei Abnahme 10Stk.	19,85 €	19,85 €	PCBP

Summe GU-K1520 ohne obsoleten Bauteilen 26,42 €

1	GAL16V8		ca.	1,50 €	obsolet
1	GAL20V8		ca.	2,00 €	obsolet
2	SN74LS646NT		ca.	4,30 €	obsolet
1	Z80 PIO (UB885D)		ca.	3,00 €	obsolet
1	EFS58	Stecker EFS58 A/B oder A/C gewinkelt	ca.	2,50 €	obsolet

ca. Summe obsoleter Bauteile, je nach Anbieter 13,30 €

Zuzüglich 1x USB-Modul VDIP1 , V2DIP1-48 oder V2DIP1-32 (für die V2DIP werden noch 2x12pol. Präzisionsverbinder einreihig RM2,54 benötigt oder die Module eben direkt einlöten)

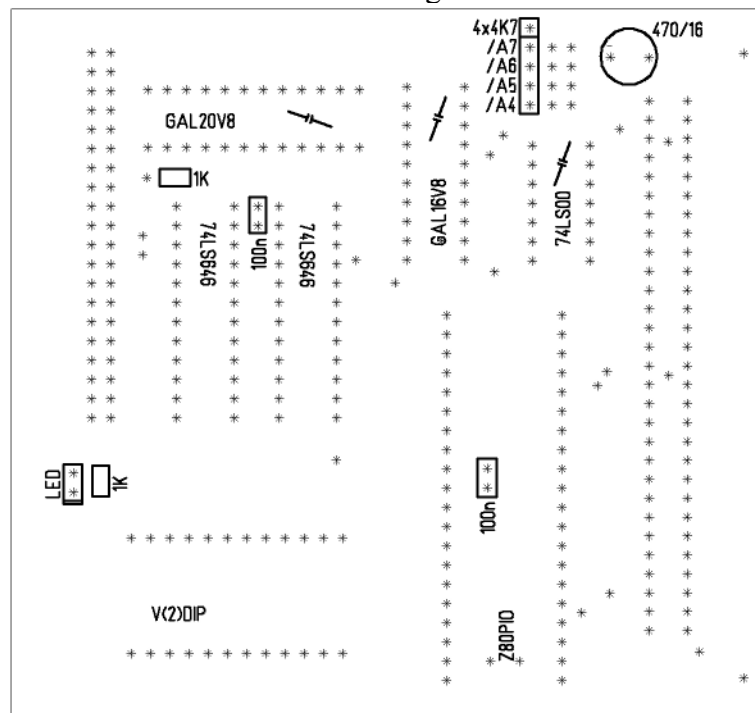
Das USB-Modul aus der „Eigenproduktion“ wird normal in den IC-Sockel gesteckt, durch die unten liegende Buchse ist die Bauhöhe für Steckkarten noch etwas optimaler (Bild unten).

Die Platine wurde für den Einsatz in meinen K1520 Systemen konzipiert, die Größe orientierte sich ursprünglich aber auch an den KC87 Modulen. Bei Einsatz in Kleinsystemen (halbe Bauhöhe wie z.B. NANOS) gibt es zum Teil etwas Platzprobleme am Widerstandsnetzwerk und IDE Stecker im Bereich der oberen Führungsschiene !

GU-K1520V6



Bestückung V.6



Bei dem Aufdruck ist es leider zu etwas unterschiedlichen Ausführungen gekommen. Für die korrekte Adressierung der GU-K1520V6 laut Aufdruck /A4-/A7 muss der GAL20V8 mit der GUK_V6a_USB.JED programmiert werden und für das Layout beginnend bei /A7 (siehe Bild) an der Außenkante, mit der GUK_V6_USB.JED !

Der GAL16V8 bleibt inhaltlich wie in den originalen Quellen aus dem Link ganz oben.

Für einen Einsatz am Z1013-128/KC87 dient die Datei GUK_V6_USB_DC.JED, da dort die Adressen etwas Querbeet liegen wird die Adresse DCh für USB zusätzlich zur Verfügung gestellt (außer bei D0h als Kartenadresse).

Hinweise zur Benutzung des GIDE:

Bei dem ersten Einsatz eines Mediums am GIDE sollte dieses formatiert werden, der nachladbare Treiber enthält dafür eine Schnellformatierung welche mit der Option /F aufgerufen und mit J (Großbuchstabe!) bestätigt wird.

Alternativ kann auch das Programm DUF05(<http://www.cpm.z80.de/binary.html#utilities>), Menüpunkt 4 und anschließend der Befehl `era *.*` benutzt werden.

Zur Verwendung des im CP/A integrierten Treibers sollte das Medium beim ersten Einsatz leer sein, andernfalls wird die erste Datei mit dem CCP überschrieben !

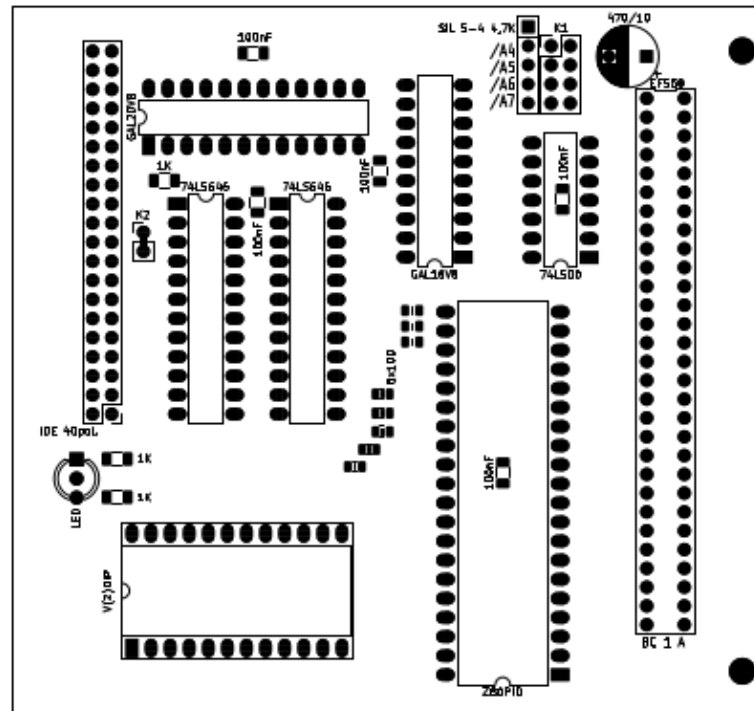
Hinweise zur Benutzung USB:

Der Status der Speichermedien sollten beim an-/abstecken via Befehl USB entsprechend aktualisiert werden. Doku zu den USB Tools findet man unter der o.g. Quelle auf KC85.info.

Treten beim kopieren von anderen Medien auf den USB-Datenträger Lesefehler auf, wird der Vorgang meist abgebrochen und dabei bleibt die zu schreibende Datei geöffnet. Man sollte dann unbedingt mit USB CLF die Datei schließen und dann mit USB DLF Filename das Fragment löschen um ein eventuelles zerstören der Dateistruktur des USB-Datenträgers zu vermeiden.

Nachtrag(2020/2021):

GU-K1520V8



Mit der GU-K1520V8, bin ich nun endlich mal den Problemen nachgegangen, welche die GU-K1520V7 (**kein Produkt von mir!**) z.B. im PC1715 bereitet. Die V7 ist im Prinzip nur eine V6, bei der die IC Sockel mit integrierten Abblockkondensator durch normale Sockel und extern angeordneten SMD-Kondensatoren ersetzt wurden. Das hat sich tatsächlich negativ auf das Verhältnis Nutz- zu Störpegel ausgewirkt und zieht damit Fehler in der Datenübertragung des GIDE-Interface nach sich.

Bei der V8 wurde die Möglichkeit geschaffen Serienwiderstände in den Datenleitungen vom GIDE einzuschleifen, welche diese Probleme kompensieren sollen. Damit ist wieder eine fehlerfreie Datenübertragung möglich.

Nebenbei wurde noch der Einsatz einer Duo-LED ermöglicht um bei Verwendung eines V2DIP zusätzlich auch den Status des USB-Mediums signalisieren zu können.

Der GAL20V8 wird mit der GUK_V8_USB.JED bzw. GUK_V8_USB_DC.JED und der GAL16V8 mit gidel1.jed wie in den originalen Quellen aus dem Link ganz oben programmiert.

Anmerkung:

Die Datei dient(e) eigentlich nur der eigenen privaten Dokumentation der GU-K1520 und erhebt keinen Anspruch auf Vollständigkeit und sachliche Richtigkeit. Der Status „Publik“ wurde nur aus „Nettigkeit“ gesetzt und sollte keinesfalls eine Aufforderung sein, dies als Basis für kommerzielle Nutzung zu verwenden! Die private Nachnutzung auf eigene Gefahr sei gestattet, wobei die Angabe der Quelle bei Vervielfältigung (auch in Auszügen) selbstverständlich sein sollte!

@Kingstener 2020