



Application Note

Version: 1.3
Datum: 11.01.2024
Autor: Achim Döbler



Inhaltsverzeichnis

1	Einleitung	2
2	Design-In	4
2.1	Integration als Smart Display	4
2.2	Integration als zentrale Steuerung der Applikation	5
3	Software	7
3.1	Entwurf der grafischen Benutzerschnittstelle	7
3.2	Integration der GUI in KEIL MDK-ARM	9
3.2.1	Konfiguration des Profils	9
3.2.2	Auswahl des Displays	9
4	Hardware	11
4.1	Spannungsversorgung	11
4.2	Pinbelegung	12
4.2.1	Programmierschnittstelle	12
4.2.2	Buchsenleisten	12
4.3	Schaltplan	16
5	Technische Daten	21
5.1	Komponenten	21
5.2	Elektrische Eigenschaften	21
5.3	Mechanische Eigenschaften	21
A	Abkürzungsverzeichnis	23

1 Einleitung

Bei einem Großteil der industriellen Applikationen stellt immer mehr die Benutzerschnittstelle ein wesentliches Verkaufsmerkmal dar. Insbesondere durch die Verfügbarkeit moderner Smartphones inspiriert, erwarten Kunden heute häufig auch selbstverständlich im industriellen Bereich ansprechende grafische Benutzerschnittstellen sowie eine einfache und intuitive Toucheingabe. Die Kernkompetenz der Hersteller wiederum liegt jedoch überwiegend direkt im Umfeld der jeweiligen Applikation, z.B. in der Regelungstechnik, Messtechnik, Medizintechnik, Leistungselektronik, und nicht im Bereich der GUI-Programmierung. Dieser Umstand hat zur Folge, dass viele, z.T. sehr leistungsfähige und am Markt renommierte Applikationen nach wie vor mit „veralteten“ Benutzerschnittstellen, bestehend aus primitiven monochrom-grafischen Displays und Schaltern, angeboten werden. Hersteller solcher Applikationen laufen nun Gefahr, trotz ihrer exzellenten Kernkompetenz, von anderen Anbietern, deren Geräte über eine ansprechende Benutzerschnittstelle verfügen, vom Markt verdrängt zu werden. Gleichzeitig werden Hersteller, die dem Trend einer modernen GUI folgen, vor enorme Schwierigkeiten gestellt. Die Entwicklung einer modernen Grafikschnittstelle erfordert sowohl im Bereich der Hardware als auch in dem der Software erhebliche zeitliche sowie finanzielle Ressourcen. Um diesem Problem zu begegnen und die benötigten Ressourcen auf ein Minimum zu reduzieren, hat ACTRON das **LIZARD**-Board entwickelt (Abb. 1). Hierbei handelt es sich um eine leistungsfähige, hochintegrierte und flexible Plattform zur direkten Ansteuerung von modernen Aktiv-Matrix-TFT-LCDs der **ACT I³**-Serie. Das **LIZARD**-Board besitzt hierdurch wesentliche Vorteile im Vergleich zu anderen Lösungen:

- ✓ Kompakte Abmessungen durch Verwendung von BGA-Technik: 55 mm x 70 mm
- ✓ Einfache Spannungsversorgung von 5V bis 38V¹
- ✓ Mikrocontrollerbasierte STM32-Plattform, kein Betriebssystem notwendig (z.B. Linux)
- ✓ MCU: STM32F469 (skalierbar)
- ✓ RAM: 8MB SDRAM (skalierbar)
- ✓ Speicher: 64MB QSPI-Flash, EEPROM, SD-Karte (skalierbar)
- ✓ Display: 3.5“ / 4.3“ / 5“ / 7“ aus **ACT I³**-Serie mit PCAP-Multitouch
- ✓ USB-OTG, USB-Host und USB-Slave-Unterstützung
- ✓ 43 GPIOs: Digitale und analoge Ein- und Ausgänge an Board-to-Board-Connector
- ✓ Industrielle Schnittstellen enthalten: u.a. SPI, CAN, UART, LIN, LAN, I²C, I²S
- ✓ Langzeitverfügbarkeit, kein proprietäres Design: Schaltpläne offen, keine Spezial-ICs
- ✓ Freies BSP für Embedded Wizard Grafikbibliothek verfügbar

¹Der zulässige Spannungsbereich hängt von der jeweiligen Konfiguration ab.

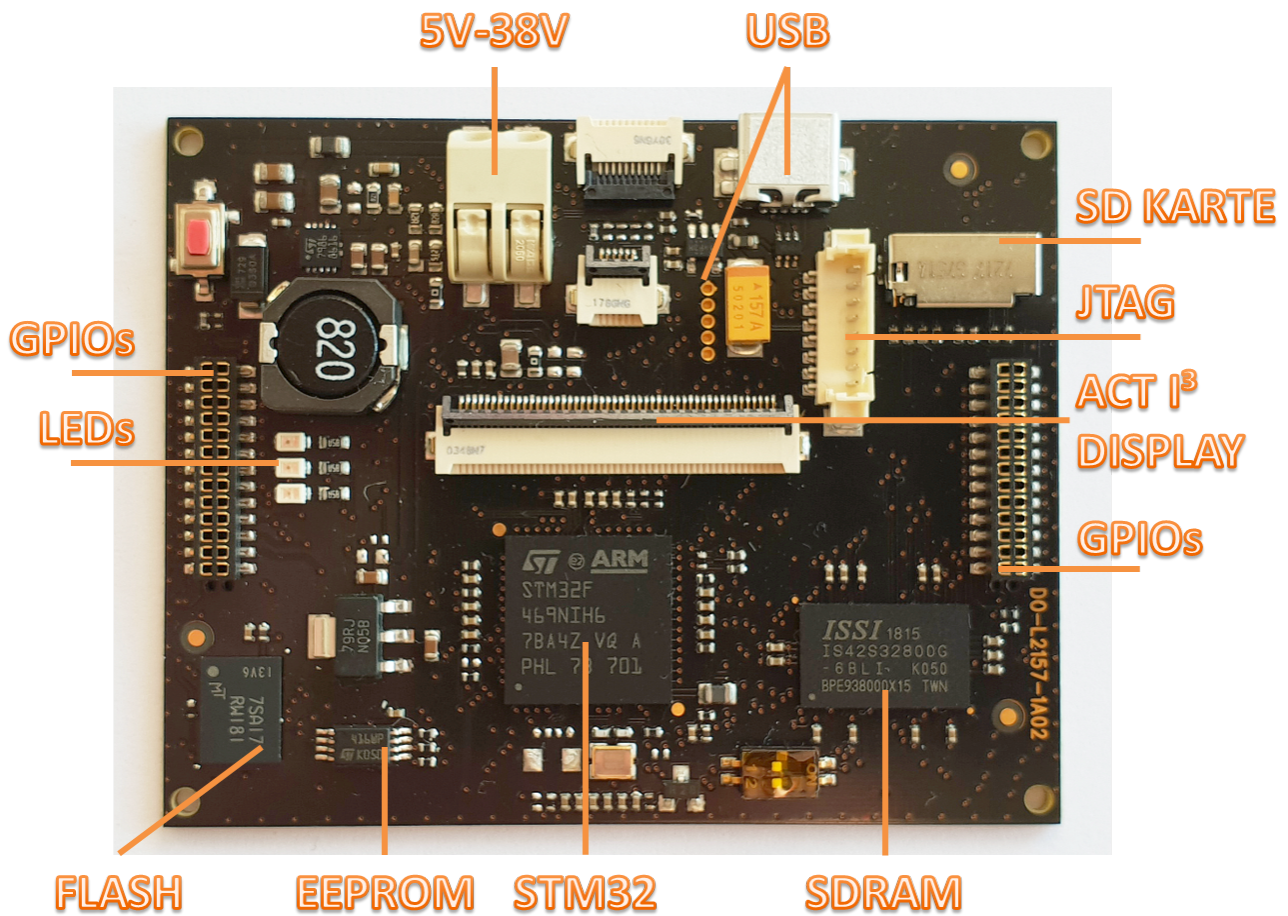


Abbildung 1: LIZARD-Board

2 Design-In

Der kompakte und flexible Aufbau des LIZARD-Boards ermöglicht es, diese leistungsfähige Plattform auf zwei verschiedene Arten in eine Applikation zu integrieren. In der Folge werden beide Varianten vorgestellt.

2.1 Integration als Smart Display

Eine Variante besteht darin, das LIZARD-Board direkt auf die Rückseite des Displays zu kleben (Abb. 2). Das LIZARD-Board wurde gezielt so entworfen, dass dies problemlos bei allen Displays der **ACT I³**-Serie möglich ist. Aus dem Display wird somit ein sog. *Smart Display*, d.h., das Display stellt die gesamte dezentrale grafische Benutzerschnittstelle bereit. Zur Datenübertragung (z.B. Benutzereingaben) zwischen LIZARD-Board und der restlichen Applikation kann eine einfache Schnittstelle wie beispielsweise SPI, I²C, UART oder CAN verwendet werden.

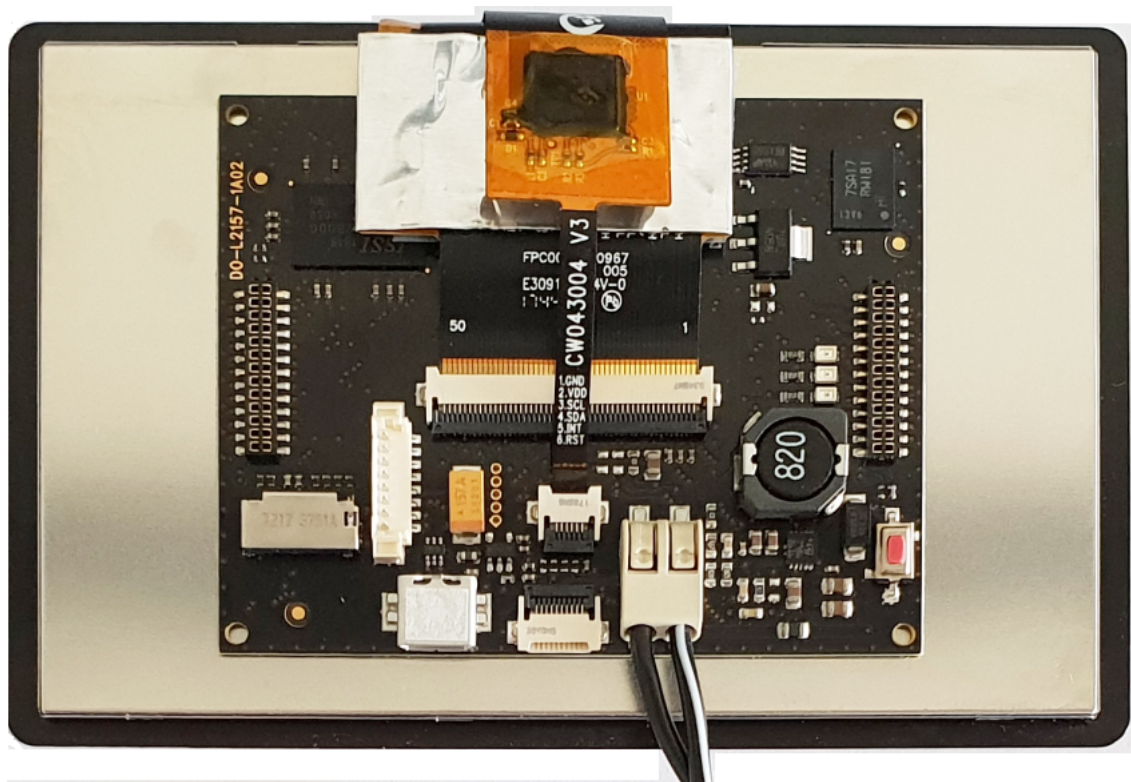


Abbildung 2: LIZARD-Board auf einem 4.3“ **ACT I³**-Display

Der Kunde kann hierbei frei entscheiden, welche Schnittstellen er verwenden möchte, er muss lediglich ein entsprechendes Kabel für die GPIO-Buchsenleisten des LIZARD-Boards konfektionieren. Benötigt der Bus einen zusätzlichen Transceiver-Baustein (z.B. CAN, RS232), so kann er auch eine kleine Adapterplatine entwickeln, welche dann von oben auf das LIZARD-Board gesteckt wird und die Verbindung zur Applikation herstellt.

Diese Integrationsvariante eignet sich ideal für bestehende Applikationen, die durch das LIZARD-Bord ein Facelift in Form eines modernen GUIs erhalten sollen. Die bisherige Hard- und Software kann hier weitestgehend unverändert bestehen bleiben. Das LIZARD-Board ist aufgrund des frei programmierbaren STM32 in der Lage, das bisherige, zu ersetzende Eingabekonzept zu emulieren und gleichzeitig zu erweitern. Auf diese Weise gestaltet sich eine Integration in ein bestehendes System denkbar einfach. Als zweite Zielgruppe für eine Smart-Display-Integration eignen sich Anwendungen, bei denen die eigentliche Funktion über hochspezialisierte Baugruppen, wie z.B. FPGAs oder Applikationsprozessoren, realisiert wird und bei denen der STM32 daher nicht direkt die Kernaufgaben übernehmen kann.

Vorteile:

- Ideal geeignet, um bereits bestehende Applikationen mit einem Facelift aufzuwerten
- Anbindung zur Applikation kann über eine einfache serielle Schnittstelle realisiert werden
- Entwurf der grafischen Benutzerschnittstelle kann vollkommen eigenständig erfolgen
- Minimaler Integrationsaufwand

Nachteile:

- Nicht alle Ressourcen des LIZARD-Boards werden genutzt
- Kundenapplikation benötigt einen weiteren Prozessor bzw. Mikrocontroller

2.2 Integration als zentrale Steuerung der Applikation

Das LIZARD-Board basiert auf einem leistungsfähigen STM32F469-Mikrocontroller. Ein wesentliches Merkmal dieser Mikrocontrollerfamilie ist der enthaltene SDRAM-Controller, die parallele DPI-Schnittstelle zur Ansteuerung von hochauflösenden AM-TFT-LCDs sowie der DMA-basierte Hardwarebeschleuniger *Chrom-ART Accelerator*. Diese Peripheriekomponenten machen den STM32 somit zu einer idealen Plattform für anspruchsvolle grafische Benutzerschnittstellen. Neben diesen eher grafikspezifischen Funktionen besitzt der STM32 jedoch auch äußerst performante Peripheriekomponenten aus anderen Bereichen. Hierzu zählen u.a. die unterschiedlichsten Schnittstellen aus dem Industriebereich, wie z.B. CAN oder Ethernet, sowie digitale und analoge Spezialfunktionen aus der Mess-, Regelungs-, Audio- und Sicherheitstechnik. Des Weiteren sind umfangreiche Funktionen zur Ansteuerung von MOSFETs enthalten, wodurch der STM32 auch hervorragend für Anwendungen aus der Leistungselektronik (z.B. Motoransteuerungen, Frequenzumrichter) geeignet ist. Vor diesem Hintergrund bietet es sich an, das LIZARD-Board nicht nur für die GUI, sondern als zentrale Steuerung für die gesamte Applikation zu verwenden. Aus diesem Grund besitzt das LIZARD-Board zwei zweireihige Buchsenleisten im 1,27mm Rastermaß, über die die Plattform auf eine kundenspezifische Trägerplatine aufgesteckt werden kann. Die Buchsenleisten können hierbei sowohl von oben als auch von unten kontaktiert werden, wodurch dem Kunden bei seiner Entwicklungsarbeit ein Maximum an Freiheitsgraden bereitgestellt wird. Die Trägerplatine stellt die Schnittstelle zur eigentlichen Applikation her und steuert die dort vorhandenen Sensoren bzw. Aktoren. Während das LIZARD-Board aus einem 6-Lagen-Multilayer-PCB und verschiedenen BGA-basierten ICs besteht, wird diese Komplexität für die Trägerplatine nicht benötigt, sodass die Entwicklung entsprechend vereinfacht wird.

Vorteile:

- Kostenersparnis, da kein zusätzlicher Prozessor bzw. Mikrocontroller für die Applikation benötigt wird
- Kunde kann alle Ressourcen des LIZARD-Boards nutzen und von der hohen Integrationsdichte profitieren
- Entwicklungsarbeit wird vereinfacht, da der Entwurf der Mikrocontrollerumgebung entfällt

Nachteile:

- Grafische Benutzerschnittstelle und Applikation laufen auf derselben Hardware und müssen gemeinsam entwickelt werden

Abbildung 3 zeigt ein Beispiel einer solchen kundenspezifischen Trägerplatine. Neben digitalen und analogen Ein- und Ausgängen wurden hier 4 Halbbrücken aus MOSFETs, eine Hall-Effekt-basierte Strommessung, ein Quadratur-Decoder-Interface für Drehimpulsgeber, eine CAN-Schnittstelle sowie 4 Relaisausgänge realisiert. Abbildung 4 zeigt die Trägerplatine mit aufgestecktem LIZARD-Board.

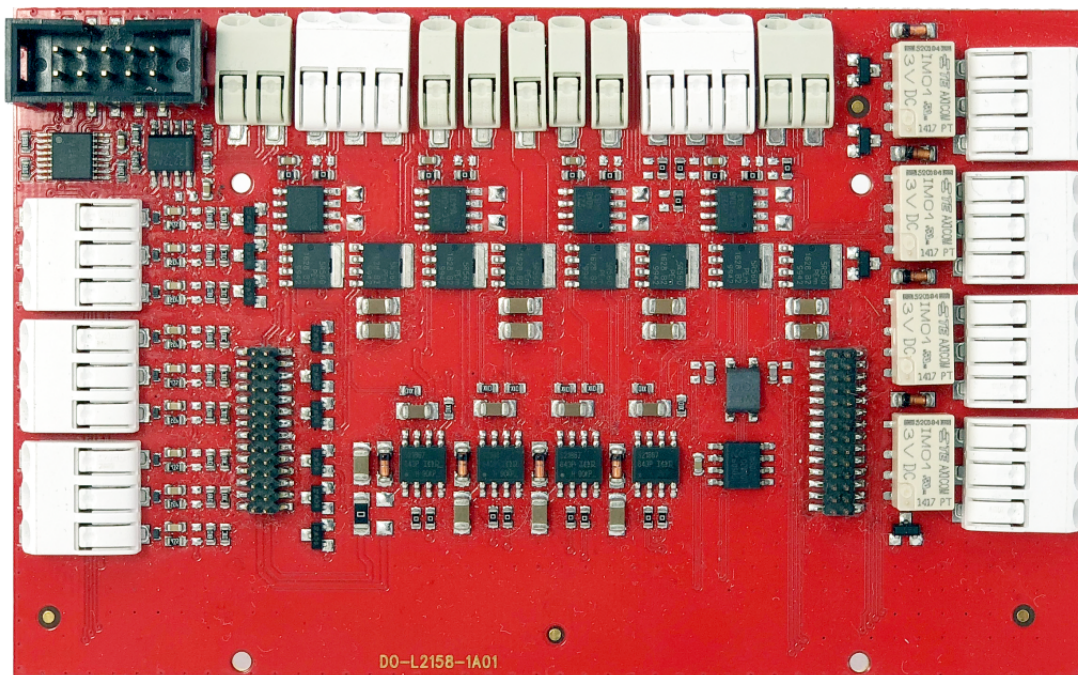


Abbildung 3: Beispiel einer kundenspezifischen Trägerplatine

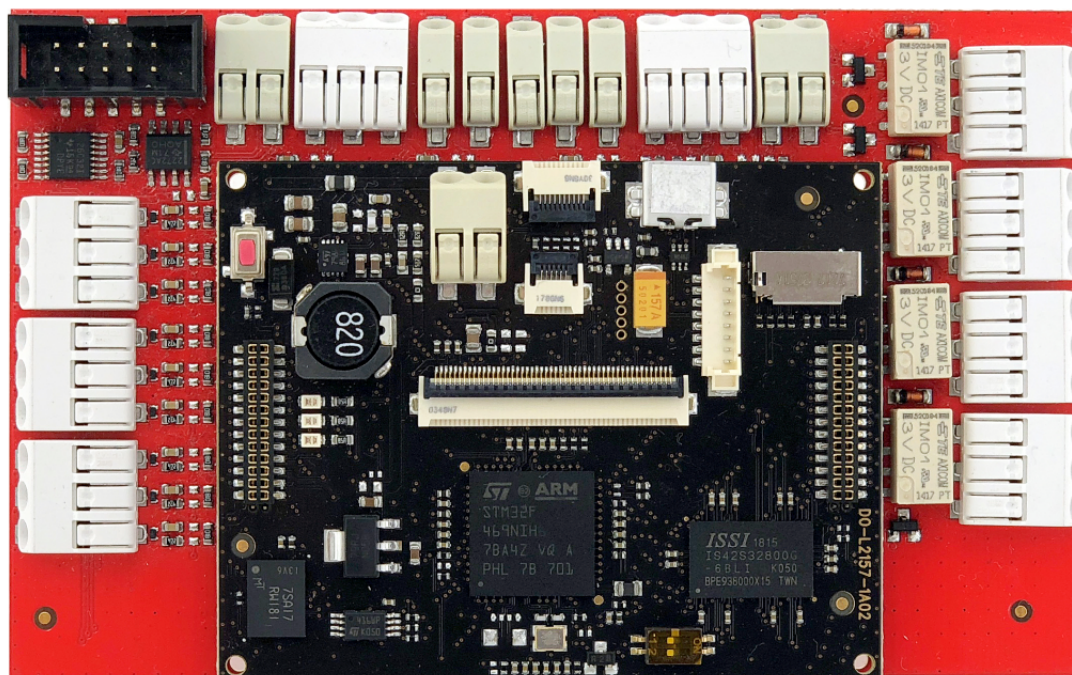


Abbildung 4: LIZARD-Board auf einer kundenspezifischen Trägerplatine

3 Software

Da das LIZARD-Board auf einem STM32 basiert, ist es grundsätzlich für alle gängigen Toolchains, wie z.B. *Atollic TrueSTUDIO*, *IAR Embedded Workbench*, *KEIL MDK-ARM* (μ Vision) oder *GCC*, geeignet, die diesen Mikrocontroller unterstützen. Auch im Bereich der Grafikbibliothek kann eine Vielzahl von GUIs der unterschiedlichsten Hersteller verwendet werden. Hierzu zählen u.a. *Segger emWin*, *STMicroelectronics TouchGFX* oder *TARA Systems Embedded Wizard*. Um dem Benutzer den Design-In-Prozess der LIZARD-Platine in seine Applikation zu erleichtern, stellt ACTRON ein frei verfügbares BSP für die weitverbreitete Toolchain *KEIL MDK-ARM* sowie die äußerst leistungsfähige Grafikbibliothek *TARA Systems Embedded Wizard* zur Verfügung. In der Folge wird exemplarisch ein Beispiel skizziert, um die benötigten Integrationsschritte zu erläutern.

3.1 Entwurf der grafischen Benutzerschnittstelle

Zunächst erstellt der Benutzer, über das Programm *Embedded Wizard Studio*, eine auf seine Anforderungen angepasste grafische Benutzerschnittstelle. Diese enthält u.a. die benötigten Grafiken und Texte in unterschiedlichen Sprachen, Tasten, Menüs, Diagramme und Animationen (Abb. 5). Anschließend kann die GUI direkt am PC simuliert werden (Abb. 6). Das LIZARD-Board wird hierzu noch nicht benötigt.

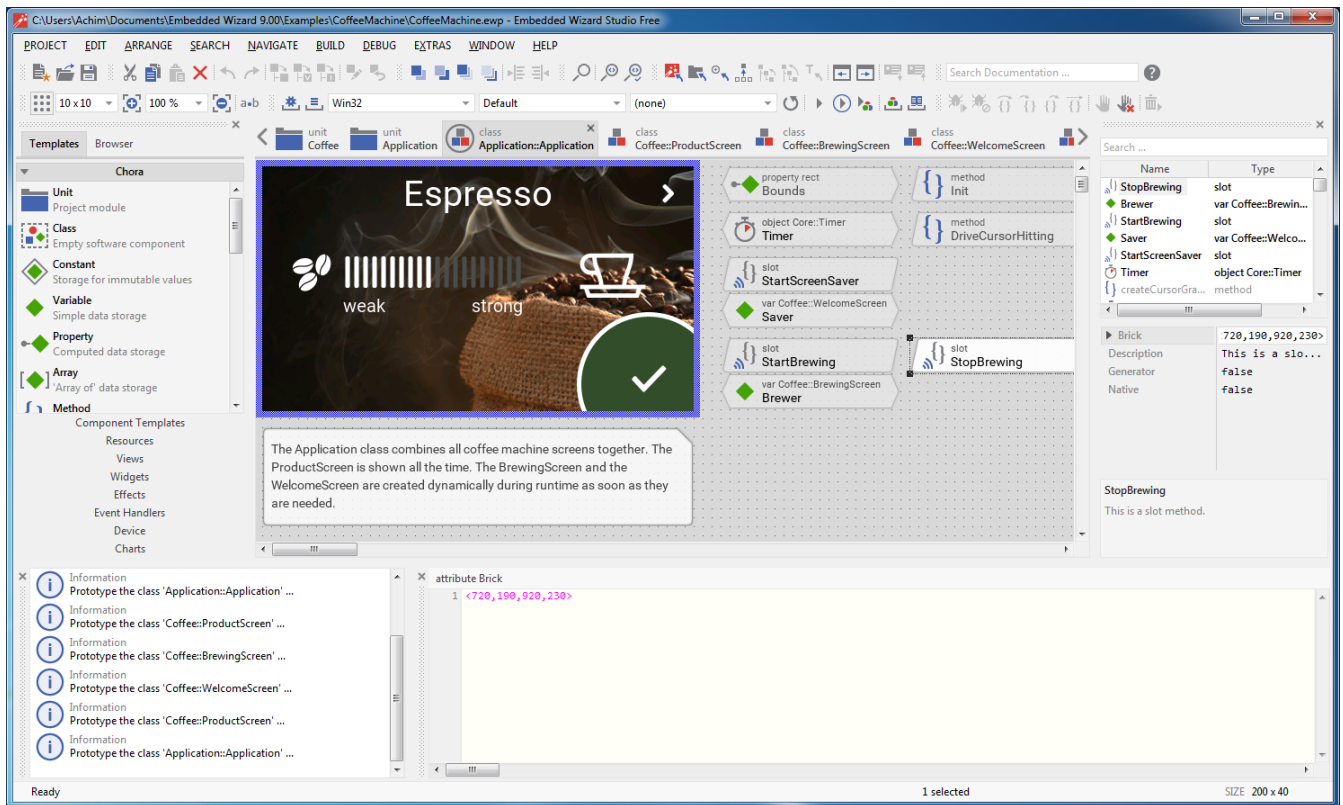


Abbildung 5: Entwurf der grafischen Benutzerschnittstelle

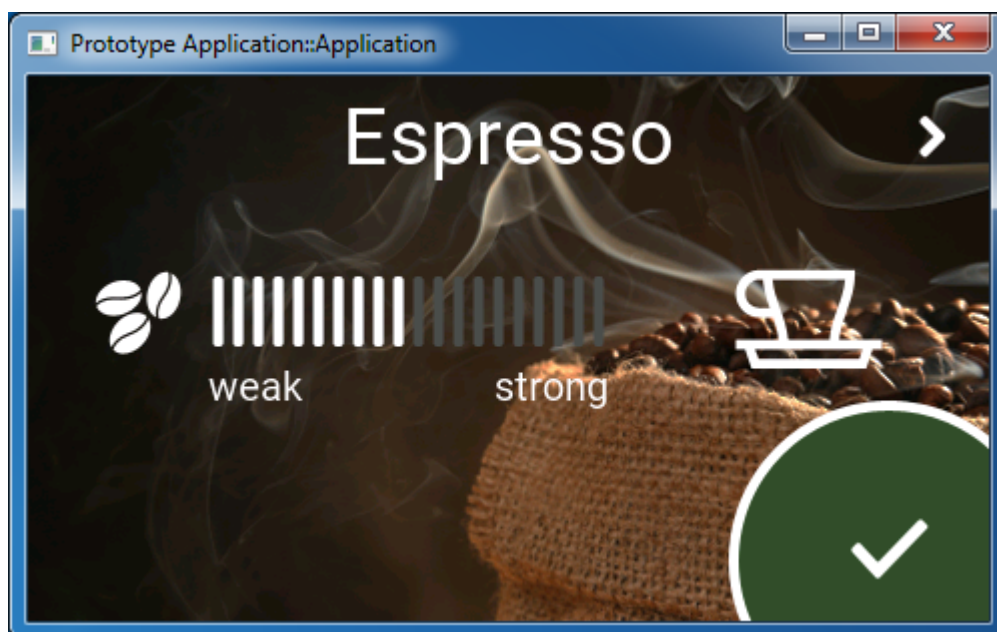


Abbildung 6: Simulation der GUI auf dem PC

3.2 Integration der GUI in KEIL MDK-ARM

Um die im *Embedded Wizard Studio* erstellte grafische Benutzerschnittstelle in die Mikrocontroller-Entwicklungsumgebung, in diesem Beispiel *KEIL MDK-ARM*, zu integrieren, muss eine Verbindung zwischen den beiden Programmen hergestellt werden. Diese ist denkbar einfach und wird in den folgenden Schritten beschrieben.

3.2.1 Konfiguration des Profils

Zunächst wird im *Embedded Wizard Studio* das Profil ausgewählt und dort in der Property-Liste folgende Werte zugeordnet.

Bezeichnung	Wert
PlatformPackage	STM.STM32.RGB565
OutputDirectory	Pfad zu dem Ordner GeneratedCode des BSPs
PostProcess	Pfad zu der Datei MDK-ARM_ew_post_process.cmd des BSPs

Tabelle 1: Konfiguration des Profils

Sind alle Werte eingetragen, so kann das Profil mit **F8** erzeugt werden.

3.2.2 Auswahl des Displays

Anschließend muss in der BSP-Datei **main.c** das verwendete Display ausgewählt werden. Dies kann direkt in dem geöffneten Projekt in *KEIL MDK-ARM* erfolgen (Abb. 7). Danach sollte sich das Projekt fehlerfrei kompilieren und auf das LIZARD-Board übertragen lassen (Abb. 8).

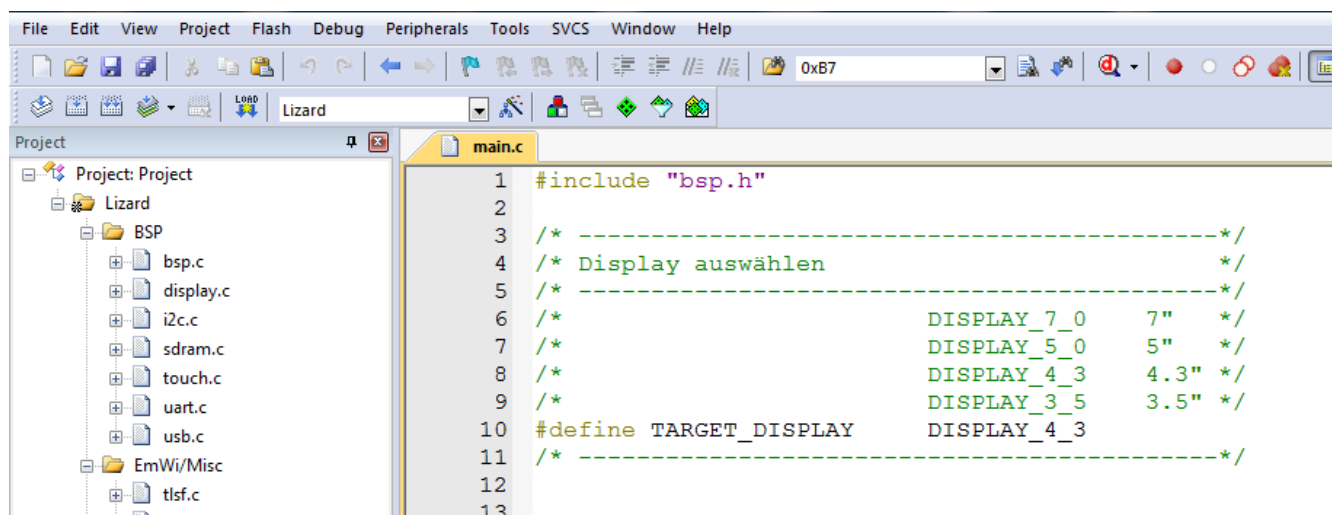


Abbildung 7: Auswahl des Displays in KEIL MDK-ARM



Abbildung 8: LIZARD-Board mit fertiger Benutzerschnittstelle

4 Hardware

4.1 Spannungsversorgung

Das LIZARD-Board kann je nach Anwendung entweder über die Leiterplattenklemmen (Abb. 9), USB oder die Buchsenleisten versorgt werden. Details zu den unterschiedlichen Konfigurationen sowie elektrischen Parametern sind in Kapitel 5.2 beschrieben.

ACHTUNG

Der zulässige Spannungsbereich des LIZARD-Boards kann je nach bestellter Bestückungsvariante variieren. Bitte betreiben Sie die Plattform erst dann mit Spannungen über 15V, wenn Sie die hierfür benötigte Konfiguration sichergestellt haben. Das LIZARD-Board besitzt keinen Verpolungsschutz.

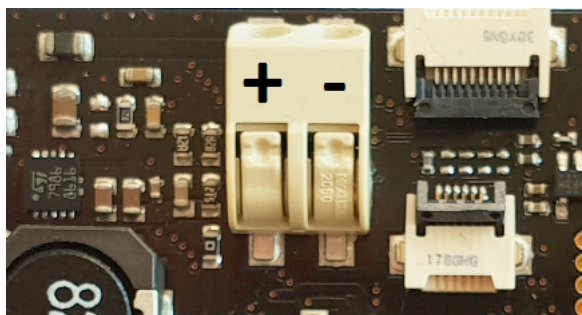


Abbildung 9: Leiterplattenklemmen

4.2 Pinbelegung

4.2.1 Programmierschnittstelle

Das LIZARD-Board kann über die Programmierschnittstelle (Abb. 10) entweder über SWD oder JTAG programmiert werden. Die Pinbelegung sowie die Anbindung an gängige Programmieradapter kann aus Tabelle 2 entnommen werden. Ein passendes Programmierkabel für das LIZARD-Board ist bei ACTRON erhältlich.

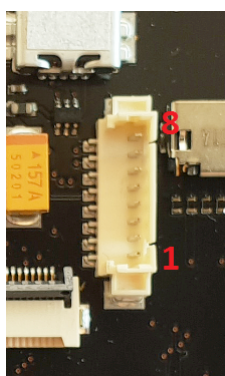


Abbildung 10: Programmierschnittstelle des LIZARD-Boards

Pin	Signal	20-poliger JTAG-Stecker	6-poliger SWD-Stecker
1	3V3	1	1
2	GND	4	3
3	TMS/SWDIO	7	4
4	TCK/SWCLK	9	2
5	$\overline{\text{RESET}}$	15	5
6	TDI	5	-
7	TDO/SWO	13	6
8	TRST	3	-

Tabelle 2: Pinbelegung der Programmierschnittstelle des LIZARD-Boards

4.2.2 Buchsenleisten

Alle Pins des STM32, die nicht für interne Funktionen des LIZARD-Boards benötigt werden, sind über Buchsenleisten² herausgeführt (Tabelle 3 und 4). Bei der Entwicklung dieser Plattform wurde besonders darauf geachtet, möglichst viele Pins mit Spezialfunktionen herauszuführen. Hierzu zählen neben unterschiedlichen digitalen Schnittstellen des STM32 auch analoge Ein- und Ausgänge und besondere Spezialfunktionen wie beispielsweise eine Batterieversorgung oder komplementäre PWM-Ausgänge mit einstellbarer Totzeit zur Ansteuerung von MOSFETs. Eine Übersicht über die verfügbaren Ports ist in Tabelle 5 dargestellt.

²Es wird empfohlen, Stiftleisten mit einer Mindeststiftlänge von 3.8 mm zu verwenden

Pin	Name	Pinbelegung / Spezialfunktion(en)	Analogfunktion(en)
1	GND	Masse	-
2	3V3	Versorgungsspannung	-
3	PA8	MCO1, T1_C1, USART1_CK	-
4	PC6	T3_C1, T8_C1, USART6_TX	-
5	PB13	T1_C1N, SPI2_SCK	-
6	PA7	T1_C1N, T3_C2, T8_C1N, SPI1_MOSI, T14_C1	ADC12I7
7	PA9	T1_C2, SPI2_SCK, USART1_TX	-
8	PC7	T3_C2, T8_C2, USART6_RX	-
9	PB14	T1_C2N, T8_C2N, T12_C1, SPI2_MISO, T12_C1, USB-	-
10	PB0	T1_C2N, T3_C3, T8_C2N	ADC12I8
11	PA10	T1_C3, USART1_RX	-
12	PC8	T3_C3, T8_C3, USART6_CK	-
13	PB1	T1_C3N, T3_C4, T8_C3N	ADC12I9
14	PB15	T1_C3N, T8_C3N, SPI2_MOSI, T12_C2, USB+	-
15	PG12	SPI6_MISO, USART6_RTS	-
16	PD5	USART2_TX	-
17	PG13	SPI6_SCK, USART6_CTS	-
18	PD6	USART2_RX	-
19	PG14	SPI6_MOSI, USART6_TX	-
20	PF6	T10_C1, UART7_RX	ADC3I4
21	PB11	T2_C4	-
22	PC5	-	ADC12I15
23	PA6	T3_C1, SPI1_MISO, T13_C1	ADC12I6
24	PA3	T2_C4, T5_C4, T9_C2, USART2_RX	ADC123I3
25	5V0	-	-
26	GND	Masse	-

Tabelle 3: Pinbelegung der Buchsenleiste X13

Pin	Name	Pinbelegung / Spezialfunktion(en)	Analogfunktion(en)
1	GND	Masse	-
2	3V3	Versorgungsspannung	-
3	PA4	SPI1_NSS, SPI3_NSS, USART2_CK	ADC12I4, DAC1
4	PA5	T2_C1, T8_C1N, SPI1_SCK	ADC12I5, DAC2
5	PA0	T2_C1, T5_C1, USART2_CTS, UART4_TX	ADC123I0
6	PC0	-	ADC123I10
7	PA1	T2_C2, T5_C2, USART2_RTS, UART4_RX	ADC123I1
8	PC1	SPI2_MOSI	ADC123I11
9	PA2	T2_C3, T5_C3, T9_C1, USART2_TX	ADC123I2
10	PC2	SPI2_MISO	ADC123I12
11	PG9	USART6_RX	-
12	PC3	SPI2_MOSI	ADC123I13
13	PG10	-	-
14	PC4	-	ADC12I14
15	PG11	-	-
16	PH6	T12_C1	-
17	PG3	-	-
18	PH7	-	-
19	PB8	T4_C3, T10_C1, I2C1_SCL, CAN1_RX	-
20	PE5	T9_C1	-
21	PB9	T4_C4, T11_C1, I2C1_SDA, SPI2_NSS, CAN1_TX	-
22	PE6	T9_C2	-
23	VBAT	Batterieversorgung	-
24	PF7	T11_C1, UART7_TX	ADC3I5
25	VPP	-	-
26	GND	Masse	-

Tabelle 4: Pinbelegung der Buchsenleiste X19

PORT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A						•	•	•	•	•	•	•	•	•	•	•
B	•	•	•		•		•	•							•	•
C								•	•	•	•	•	•	•	•	•
D										•	•					
E										•	•					
F									•	•						
G		•	•	•	•	•	•						•			
H									•	•						
I																
J																
K																

Tabelle 5: Portübersicht der verfügbaren GPIOs

Schnittstelle	Verfügbare Module	Anzahl gesamt
SPI	1, 2, 6	3
QSPI	-	0
RMI	1	1
CAN	1	1
UART	1, 2, 4, 6, 7	5
USB	FS	1
LIN	1, 2, 4, 6, 7	5
IRDA	1, 2, 4, 6, 7	5
I ² C	1	1
I ² S	2	1

Tabelle 6: An den Buchsenleisten zur Verfügung stehende Schnittstellen

4.3 Schaltplan

Der Schaltplan des LIZARD-Boards ist in den Abbildungen 11 bis 14 dargestellt. Der zugehörige Positionsdruck kann aus Abbildung 15 entnommen werden.

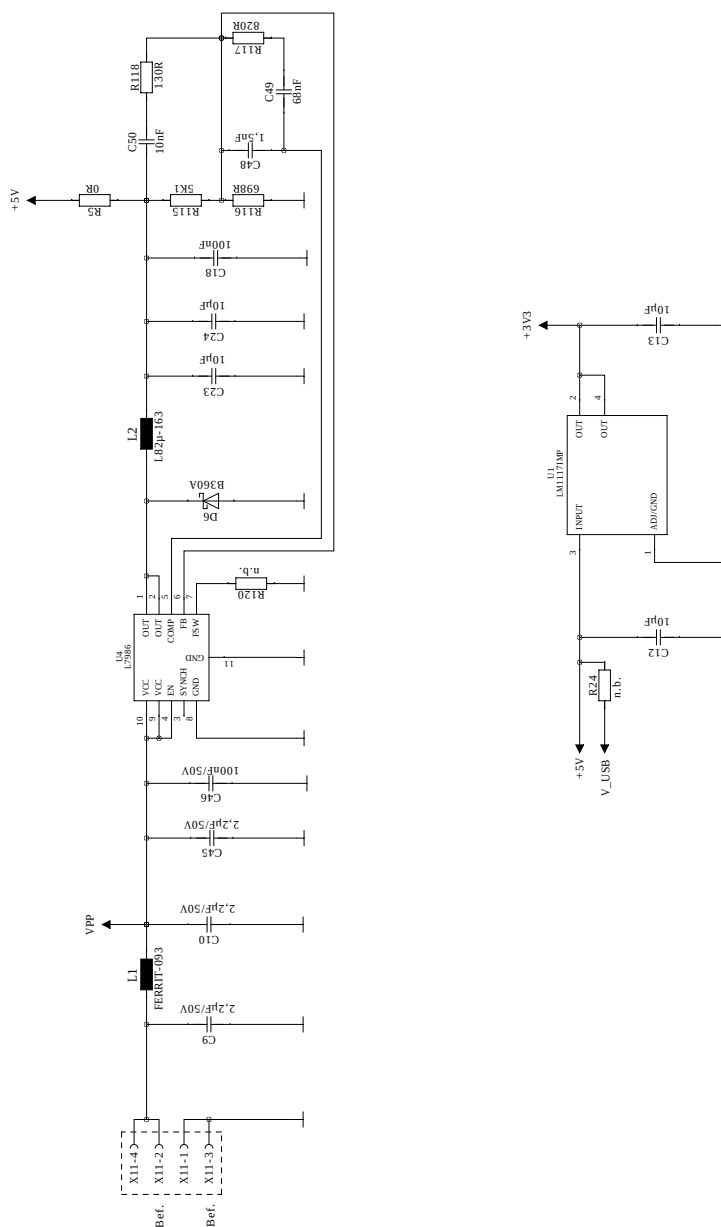


Abbildung 11: Schaltplan: Spannungsversorgung des LIZARD-Boards



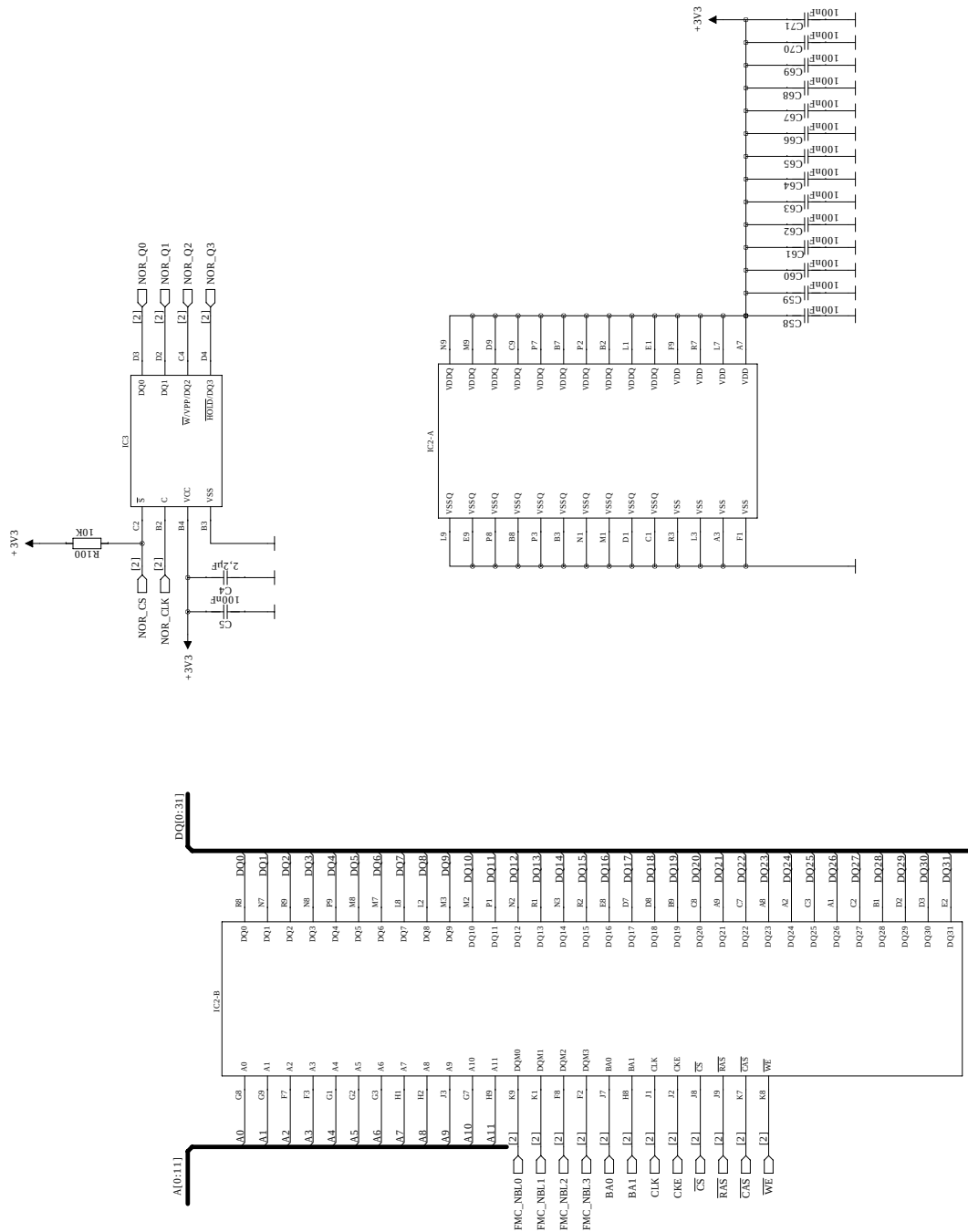


Abbildung 13: Schaltplan: SDRAM und NOR-Flash



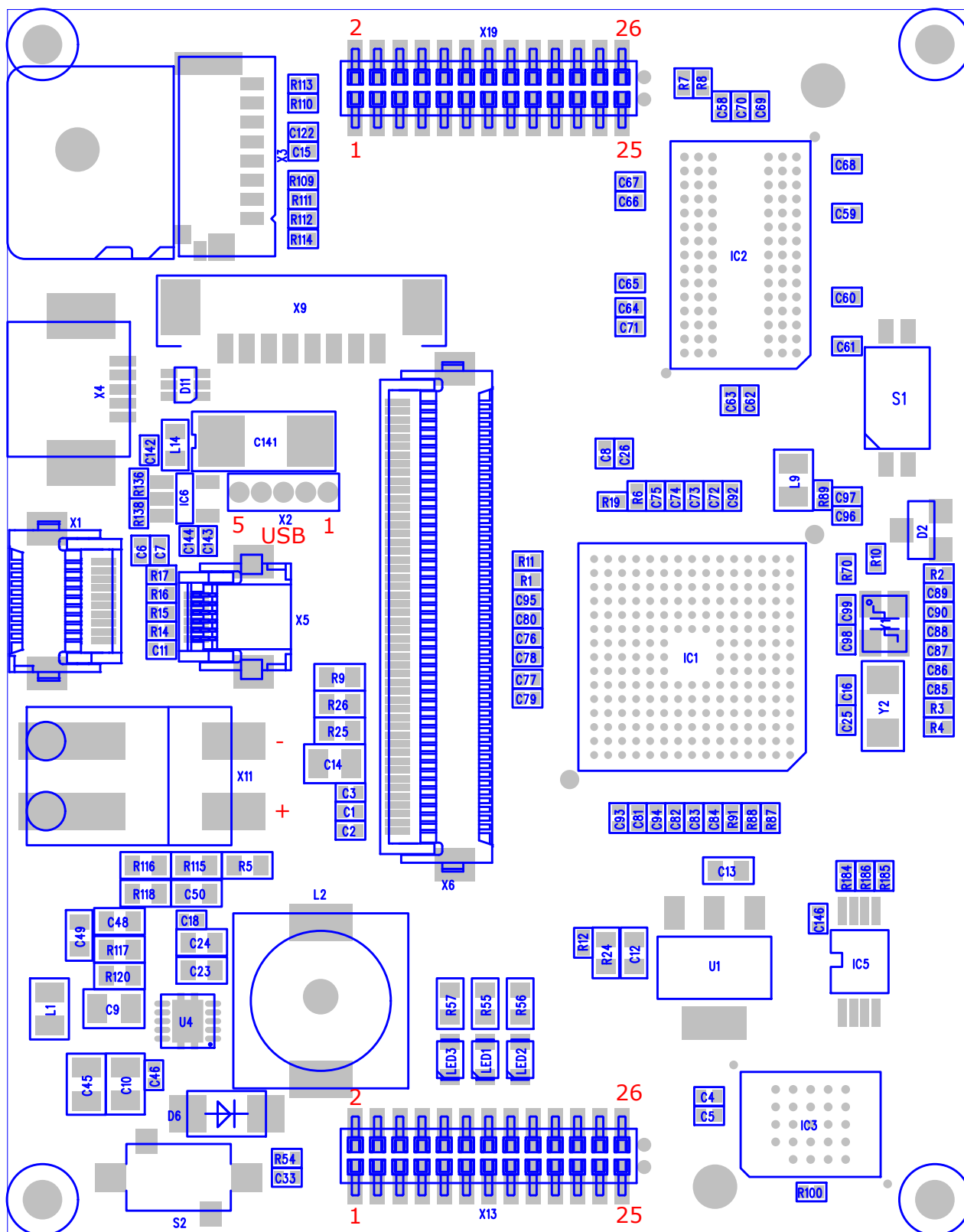


Abbildung 15: Positionsdruck des LIZARD-Boards

5 Technische Daten

5.1 Komponenten

Bezeichnung	Typ	Eigenschaften	Hersteller
Mikrocontroller	STM32F469NIH6 (default)	32-Bit ARM Cortex M4, 180MHz	STMicroelectronics
RAM	IS42S32200L-6BLI	32-Bit, 8MB SDRAM	ISSI
Flash	MT25QL512ABB8E12	64MB QSPI NOR-Flash	Micron
EEPROM	M24C16-WDW6TP	2KB EEPROM	STMicroelectronics

Tabelle 7: Wichtige Komponenten des LIZARD-Boards

5.2 Elektrische Eigenschaften

Symbol	Parameter	Bedingung	MIN	TYP	MAX	Einheit
V_{PP}	Versorgungsspannung ^a	R25 geschlossen, R9 offen (default)	5	12	15	V
V_{PP}	Versorgungsspannung	R25 offen, R9 geschlossen	5	24	38	V
T_A	Umgebungstemperatur	-	-20	25	70	°C

Tabelle 8: Elektrische Eigenschaften

^aDie Eingangsspannung wird direkt zur Versorgung des **ACT I³**-Displays verwendet. Daher ist der Spannungsbereich hier auf $5V \leq V_{PP} \leq 15V$ begrenzt.

5.3 Mechanische Eigenschaften

Symbol	Parameter	Bedingung	MIN	TYP	MAX	Einheit
L	Länge	-	69.8	70.0	70.2	mm
B	Breite	-	54.8	55.0	55.2	mm
H	Höhe	-	-	6.5	-	mm

Tabelle 9: Mechanische Eigenschaften

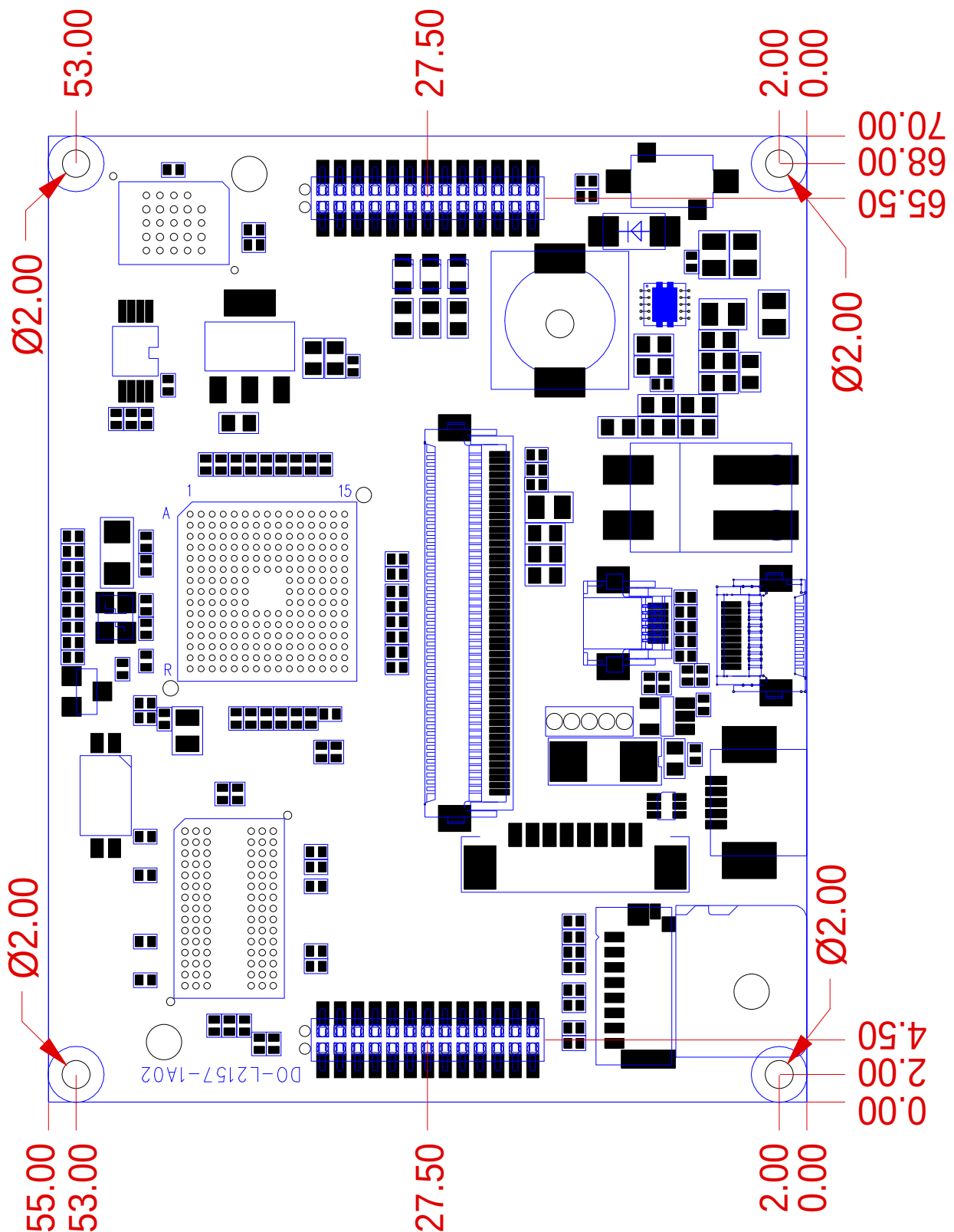


Abbildung 16: Abmessungen des LIZARD-Boards

A Abkürzungsverzeichnis

A	
AM	active-matrix
B	
BGA	ball grid array
BSP	board support package
C	
CAN	controller area network
D	
DMA	direct memory access
DPI	display pixel interface
E	
EEPROM	electrically erasable programmable read-only memory
F	
FPGA	field programmable gate array
G	
GUI	graphical user interface
GPIO	general-purpose input/output
I	
IDE	integrated development environment
I²C, IIC	inter-integrated circuit
I²S	inter-IC sound
J	
JTAG	joint test action group
L	
LAN	local area network
LCD	liquid crystal display
LIN	local interconnect network
M	
MCU	microcontroller unit
MOSFET	metal-oxide-semiconductor field-effect transistor
O	
OTG	on the go
P	
PCAP	projected capacitive
Q	
QSPI	quad serial peripheral interface
R	
RAM	random access memory
ROM	read only memory
S	
SPI	serial peripheral interface

SDRAM synchronous dynamic random access memory

T

TFT thin film transistor

U

UART universal asynchronous receiver transmitter

USB universal serial bus

Abbildungsverzeichnis

1	LIZARD-Board	3
2	LIZARD-Board auf einem 4.3“ ACT I³ -Display	4
3	Beispiel einer kundenspezifischen Trägerplatine	6
4	LIZARD-Board auf einer kundenspezifischen Trägerplatine	7
5	Entwurf der grafischen Benutzerschnittstelle	8
6	Simulation der GUI auf dem PC	8
7	Auswahl des Displays in KEIL MDK-ARM	9
8	LIZARD-Board mit fertiger Benutzerschnittstelle	10
9	Leiterplattenklemmen	11
10	Programmierschnittstelle des LIZARD-Boards	12
11	Schaltplan: Spannungsversorgung des LIZARD-Boards	16
12	Schaltplan: STM32	17
13	Schaltplan: SDRAM und NOR-Flash	18
14	Schaltplan: Display, EEPROM, LEDs, SD-Karte, USB, GPIOs und JTAG	19
15	Positionsdruck des LIZARD-Boards	20
16	Abmessungen des LIZARD-Boards	22

Tabellenverzeichnis

1	Konfiguration des Profils	9
2	Pinbelegung der Programmierschnittstelle des LIZARD-Boards	12
3	Pinbelegung der Buchsenleiste X13	13
4	Pinbelegung der Buchsenleiste X19	14
5	Portübersicht der verfügbaren GPIOs	15
6	An den Buchsenleisten zur Verfügung stehende Schnittstellen	15
7	Wichtige Komponenten des LIZARD-Boards	21
8	Elektrische Eigenschaften	21
9	Mechanische Eigenschaften	21

Version	Datum	Beschreibung
v1.3	11.01.2024	▪ Pin 23 in Tabelle 3 korrigiert. ▪ Hinweis zur Mindeststiftlänge ergänzt.
v1.2	28.10.2018	▪ Korrektur einiger Fehler.
v1.1	15.10.2018	▪ Korrektur einiger Fehler. ▪ Durchmesser der Bohrungen ergänzt. ▪ Pinbelegung der USB-Stiftleiste ergänzt.
v1.0	07.09.2018	▪ Erste Veröffentlichung. ▪ Dokumentation erstellt mit $\text{\LaTeX}$.