



---

# Interpretationspapier zur **Verordnung (EU) 2023 / 1230 über Maschinen**

<b>Kapitel</b>	I
<b>Artikel</b>	3, Ziffer 35
<b>Anhang</b>	IV
<b>Abschnitt</b>	m)
<b>Dokument</b>	IntPa-04.01
<b>Datum</b>	07. Aug. 2025
<b>Version</b>	1.0

## **Artikel 3, Ziffer 35**

„**Quellcode**“ bezeichnet die derzeit installierte Version der Software eines in den Anwendungsbereich dieser Verordnung fallenden Produkts, die in einer Programmiersprache so geschrieben ist, dass sie für den Menschen eindeutig und verständlich ist.

## **Anhang IV Technische Dokumentation, Teil A**

Die technischen Unterlagen enthalten zumindest folgende Elemente:

m) den **Quellcode** oder die Programmierlogik der Schaltung der sicherheitsrelevanten Software zum Nachweis der Konformität der Maschine oder des dazugehörigen Produkts mit dieser Verordnung auf begründeten Antrag einer zuständigen nationalen Behörde, falls dies für die Überprüfung der Einhaltung der grundlegenden Sicherheits- und Gesundheitsschutzanforderungen nach Anhang III durch diese Behörden erforderlich ist.



Weitere Interpretationspapier auf  
<https://www.nsbiv.ch/verordnung-eu-2023-1230>



## 1 Ziel und Zweck

In der Verordnung (EU) 2023/1230 über Maschinen (EU-MaschV) werden einige neue Anforderungen an die Wirtschaftsakteure gestellt. Derzeit besteht weder ein Leitfaden zur Anwendung der neuen EU-MaschV noch sind zu allen Anforderungen harmonisierte Normen (Stand der Technik) verfügbar.

Deshalb stellt die NSBIV AG, Zertifizierungsstelle *SIBE Schweiz*, den Wirtschaftsakteuren Interpretationspapiere zur Verfügung, die nach heutigem Stand von Wissen und Technik erstellt, laufend an die technische Entwicklung und die Erfahrungen aus dem Feld angepasst werden.

Die Interpretationspapiere haben keinen gesetzlichen Charakter, können aber als Stand der Technik Papiere verwendet werden, bis harmonisierte Normen oder ein Leitfaden die Anforderungen konkretisieren.

## 2 Erläuterung der Anforderung

Der **Begriff Quellcode** (en: Sourcecode) bezeichnet den in einer Programmiersprache geschriebenen Text, der die Anweisungen für ein lauffähiges Programm enthält.

Unter Ziffer 35 des Artikel 3 ist ein Quellcode auf der Stufe der Software zu verstehen, welche von einem **Anwender**, der die Programmiersprache beherrscht, **einfach gelesen** werden kann.

Ein kompiliertes File (Binary-File), welches direkt auf die Steuerung geladen wird ist KEIN Quellcode, da die Maschinensprache nicht durch den Menschen zu lesen ist.

Mit dem Quellcode ist, die auf der Maschine **aktuell installierte Version** gemeint. Im Umkehrschluss heisst dies, dass veraltete oder sich noch in der Entwicklung befindende Versionen im Sinne der Verordnung (EU) 2023/1230 über Maschinen nicht als Quellcode bezeichnet werden.

Als Quellcode gilt **auch eine Programmierlogik**, welche insbesondere bei einer Sicherheits-Steuerung zur Anwendung kommt. Diese sind in der Anwendung oft nicht auf Text basierend, sondern als „graphische“ Programmierung in Form von Funktionsbausteindiagrammen ausgeführt.

Für die Einhaltung der grundlegenden Sicherheits- und Gesundheitsschutzanforderungen betrifft der Quellcode in dem meisten Fällen nur den Quellcode der Sicherheits-Steuerung oder die Parametrisierung von Antriebseinheiten (siehe Kapitel Beispiele).

Der Quellcode muss in begründeten Fällen den nationalen Behörden als Nachweis, für die bei Inverkehrbringen programmierten Sicherheitsfunktion, zur Verfügung gestellt werden können.

## 3 Stand der Technik

### 3.1 Harmonisierten Normen

**EN ISO 13849-1/2:** Sicherheit von Maschinen – Sicherheitsbezogene Teile von Steuerungen  
In dieser Norm ist eine ausführliche Beschreibung der Anforderungen an die sicherheitsbezogene Steuerungssoftware und deren Entwicklung nach V-Modell zu finden.

**EN 61131-3:** Speicherprogrammierbare Steuerungen - Teil 3: Programmiersprachen  
Diese Norm beinhaltet Guidelines bezüglich der Programmiersprachen von Steuerungen

## 4 Interpretation nach *SIBE Schweiz*

### 4.1 Anforderungen an die Wirtschaftsakteure

Unter Ziffer 35 des Artikels 3 ist ein Quellcode die Stufe der Software, welche von einem Anwender, der die Programmiersprache beherrscht, einfach gelesen werden kann.

Ein kompiliertes File (Binary-File) welches direkt auf die Steuerung geladen wird ist KEIN Quellcode, da die Maschinensprache nicht durch den Menschen zu lesen ist.



## 4.2 Mögliche Umsetzung

Ein Quellcode ohne entsprechende Dokumentation ist für den Anwender nicht einfach lesbar, da viel Interpretationsspielraum besteht. Folglich beinhaltet der Quellcode auch praxisübliche Dokumentationsformen wie Flussdiagramme, Zustandsmaschinen oder die Codebeschreibungen.

Diese Dokumentation kann je nach Möglichkeit des Herstellers innerhalb des Codes oder in einem separaten Dokument ausgeführt sein. Oft bieten Hersteller von Sicherheitssteuerungen an, diese Dokumentation automatisch zu generieren.

Der Quellcode muss über den gesamten Lebenszyklus der Maschine zur Verfügung stehen. Wenn der Quellcode im Laufe einer Serienproduktion ändert, ist diese im Sinne der Rückverfolgbarkeit zu dokumentieren. Konkret heisst dies, die Seriennummer der Maschine und der Quellcode haben eine eindeutige Beziehung zueinander.

Bezüglich der Lesbarkeit ist zu beachten, dass die Lebensdauer der Maschine bis zu 20 Jahren betragen kann. Folglich muss der Quellcode auch nach dieser Zeit noch lesbar sein. Für die Dokumentation sind Standard-Dateiformate zu verwenden.

Es empfiehlt sich, einen Rechner mit Betriebssystem und Software Tools für eine spätere Reproduzierbarkeit des Quellcode aufzubewahren. Die Systemeinstellungen der Tools, wie beispielsweise des Compilers, sind zu dokumentieren. So ist sichergestellt, dass die Sicherheitsfunktionen auf Verlangen erneut validiert werden können.

Zu den technischen Unterlagen, welche dem Betreiber der Maschine ausgehändigt werden, gehören auch Funktionsbausteindiagramme, welche gleichwertig wie Stromlaufpläne zu betrachten sind. Der Betreiber muss in der Lage sein, die Sicherheitsfunktionen anhand der technischen Unterlagen nachvollziehen zu können, wie beispielsweise für Reparaturzwecke oder das Ersetzen defekter Steuerungen.

## 5 Beispiele

### 5.1 Arten von «Codierungen»

Flussdiagramme	Zustandsmaschinen	Codebeschreibungen (Text)
<pre> graph TD     Start([START]) --&gt; In1[/Eingabe Zahl 1/]     In1 --&gt; In2[/Eingabe Zahl 2/]     In2 --&gt; Summe[Summe = Zahl1 + Zahl2]     Summe --&gt; Out{Ausgabe?}     Out -- Ja --&gt; Print[Print Summe]     Out -- Nein --&gt; End([ENDE])     Print --&gt; End         </pre>	<pre> stateDiagram-v2     [*] --&gt; START     START --&gt; Zahl1_erhalten : Eingabe Zahl1     Zahl1_erhalten --&gt; Summe : Eingabe Zahl2     Summe --&gt; Ausgabe_Status : Ausgabe Anforderung     Ausgabe_Status --&gt; ENDE : Summe ausgeben         </pre>	<pre> 1 #include &lt;stdio.h&gt; 2 3 int main() { 4     int zahl1, zahl2, summe; 5 6     // Eingabe der ersten Zahl 7     printf("Gib die erste Zahl ein: "); 8     scanf("%d", &amp;zahl1); 9 10    // Eingabe der zweiten Zahl 11    printf("Gib die zweite Zahl ein: "); 12    scanf("%d", &amp;zahl2); 13 14    // Berechnung der Summe 15    summe = zahl1 + zahl2;         </pre>

## 5.2 Sicherheitsbezogene Embedded- Software (SRESW)



```

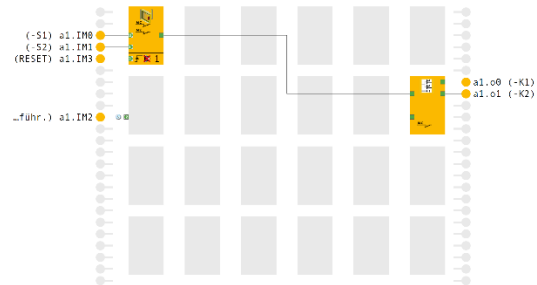
1 #include <stdio.h>
2
3 int main() {
4     int zahl1, zahl2, summe;
5
6     // Eingabe der ersten Zahl
7     printf("Gib die erste Zahl ein: ");
8     scanf("%d", &zahl1);
9
10    // Eingabe der zweiten Zahl
11    printf("Gib die zweite Zahl ein: ");
12    scanf("%d", &zahl2);

```

Quelle: ACD-Antriebstechnik

Diese wird hauptsächlich von Herstellern von Sicherheitsbauteilen verwendet. Die Programmierung erfolgt durch ausgebildete Softwareentwickler oder Ingenieure. Als Plattform dienen Leiterplatten mit Mikroprozessoren, welche mit höheren Programmiersprachen wie C oder C# programmiert werden.

## 5.3 Sicherheitsbezogene Anwendungssoftware (SRASW)

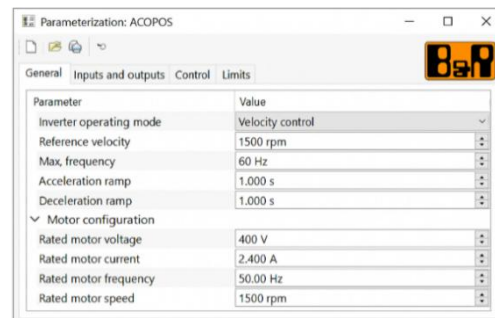


Quelle: Pilz GmbH & Co. KG

Die Anwendersoftware wird in einer Programmierumgebung erstellt, die vom Hersteller des Sicherheitsbauteil zu Verfügung gestellt wird.

Die Programmierung kann je nach Hersteller in verschiedenen Programmierformen wie Kontaktplan, strukturierter Text, Ablaufsprache, Funktionsbausteindiagramme und Anweisungsliste erfolgen. Somit sind Steuerungsprogrammierer ohne Kenntnisse einer höheren Programmiersprache in der Lage, die Sicherheitsfunktionen in der Steuerung zu implementieren.

## 5.4 Softwarebasierende Parametrisierung (PARAM)



Quelle: B&R Industrial Automation

Wie bei der sicherheitsbezogene Anwendungssoftware, wobei die Programmierung ausschliesslich über die Parametrisierung (ohne Logik) erfolgt. Typische Beispiele sind Antriebsmodule mit integrierten Sicherheitsfunktionen wie:

- Sicherer Momentabschaltung (STO, Safe Torque Off)
- Sicherer Stopp der Kategorie 1 (SS1, Safe Stopp 1)
- Sicher begrenzte Geschwindigkeit (SLS, Safe Limited Speed)