



## Parameterized Screens using Alias Variables

Part	Version	Revision	Date	Status
en	5.1.1183.1	001	2021-04-28	Released

## Content

<b>Introduction</b>	<b>2</b>
<b>Procedure</b>	<b>2</b>
1. Create Parameter Files . . . . .	2
2. Create Parameterized Screen . . . . .	5
3. Create Screen Commands . . . . .	6
<b>Disclaimer</b>	<b>10</b>

---

## Introduction

This document explains how to use Screen Parameters in an HMI project. A single parameterized screen may replace many individual screens that display similar information. This saves development time, leaves fewer screens to manage, and saves memory with fewer project resources. In these situations, only one screen may be created and its elements can be linked to “alias” variables. These “dummy” alias variables are linked to actual Real Time DB variables via Parameter File resources.

## Procedure

This section describes the procedure for defining Parameter Files and creating and calling a parameterized screen.

### 1. Create Parameter Files

To add a new parameter file to the project, right-click on *Parameters* in the project resources and select *Add a new Parameter File*.

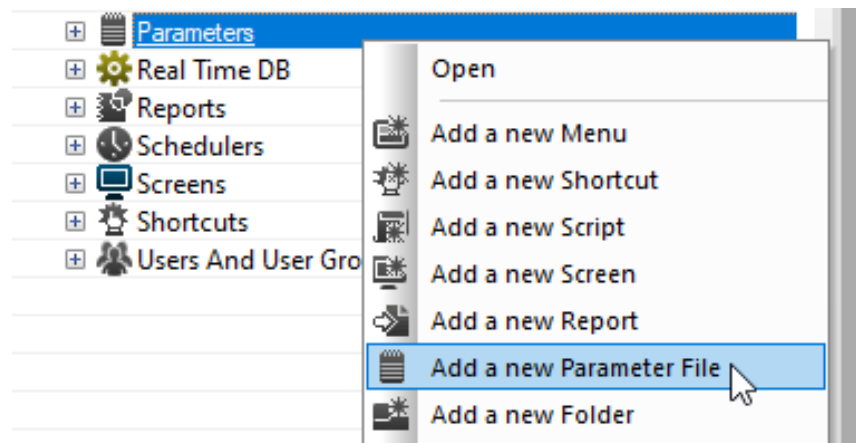


Fig. 1: Add Parameter File

An empty parameter file will appear. To add a new entry, right-click on an empty row and select *New Alias*. Define a descriptive name.

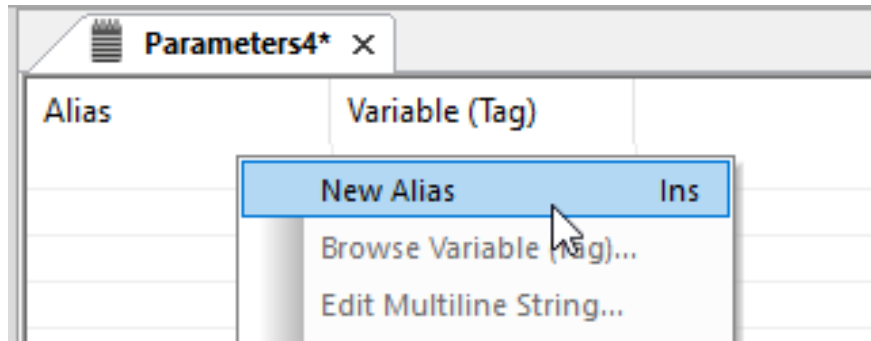


Fig. 2: Add New Alias

Right-click in the *Variable (Tag)* column and select *Browse Variable (Tag)...*. Locate the variable that will be assigned to the alias, select it, then select *OK*.

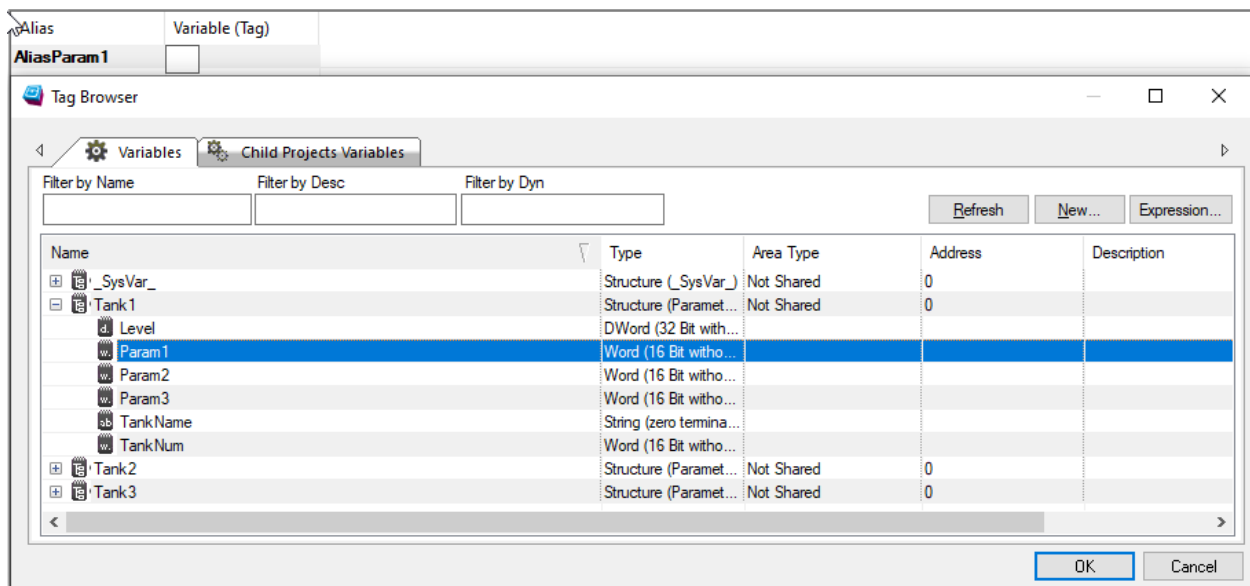


Fig. 3: Browse for Variable

Alias variables inherit the data type of their linked Real Time DB variables. In this example, a parameterized screen will be used to display and edit settings for three tanks.

# FAQ COMBIVIS Studio HMI

Parameters1* x		Parameters2 x		Parameters3 x	
Alias	Variable (Tag)	Alias	Variable (Tag)	Alias	Variable (Tag)
AliasParam1	Tank1:Param1	AliasParam1	Tank2:Param1	AliasParam1	Tank3:Param1
AliasParam2	Tank1:Param2	AliasParam2	Tank2:Param2	AliasParam2	Tank3:Param2
AliasParam3	Tank1:Param3	AliasParam3	Tank2:Param3	AliasParam3	Tank3:Param3
AliasTankNum	Tank1:TankNum	AliasTankNum	Tank2:TankNum	AliasTankNum	Tank3:TankNum
AliasLevel	Tank1:Level	AliasTankName	Tank2:TankName	AliasTankName	Tank3:TankName
AliasTankName	Tank1:TankName	AliasLevel	Tank2:Level	AliasLevel	Tank3:Level

Fig. 4: Sample Parameter files

Each of these tanks have three parameters that will be modified in the screen, a tank level, a tank name, and a tank number. A “Tank” structure was created to organize the parameterized variables in this example, but this is not necessary.

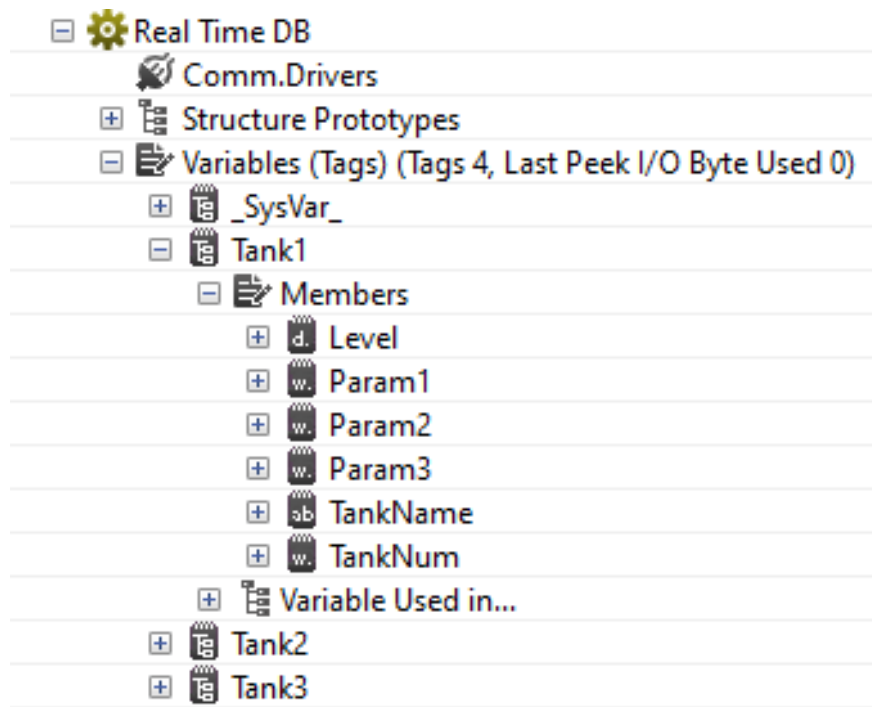


Fig. 5: Tank Structures in Real Time DB

## 2. Create Parameterized Screen

Add a new screen to the project that the Parameter Files will control. Disable the *Keep always in memory* option in the *Style* properties of the screen. Parameterized screens load their parameter file when the screen loads. Therefore, failure to disable the *Keep always in memory* option will lock the first parameter file that the screen is called with rather than change it with each call of the screen according to the command. Also, set the *Close Screen Delay* option in the *Background* properties of the screen to 0. This will keep the screen from being retained in memory with its parameter file once it is closed.

Add elements to the screen. Parameterized screens may be used to display and write to parameter file variables as well as include the same elements and perform the same functions as any normal screen. Instead of linking Real Time DB variables to the parameterized elements, however, enter the alias name for the variable that is defined in the parameter file. Shown below is an editbox element that will edit "AliasParam1" in the parameterized screen.

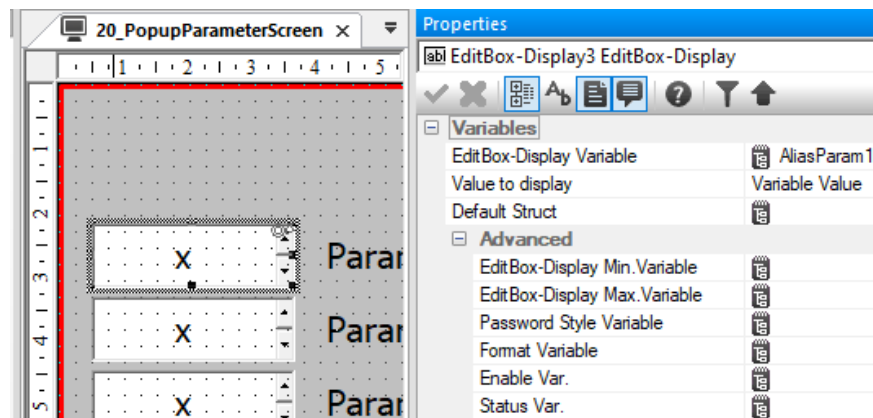


Fig. 6: Link Editbox Element to Alias

Depending on the Parameter File that the screen was loaded with, this editbox element will edit "Tank1:Param1", "Tank2:Param1", or "Tank3:Param1".

In the example parameterized screen shown below, the three parameters for the tank are editable via editbox elements, the tank level is displayed with an animated symbol from the Symbol Libraries, the tank name is displayed at the top of the screen, and the dynamic text-edge color is file by the tank number.

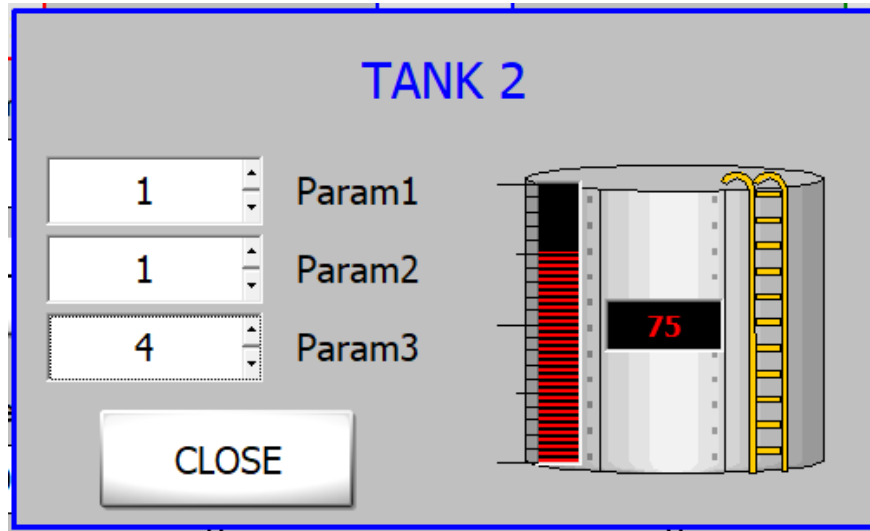


Fig. 7: Sample Parameterized Screen

As with any other screen, remember to include a way to navigate out of the screen such as a *Close and Return Back Screen Command* linked to a pushbutton.

### 3. Create Screen Commands

To load a parameterized screen, a screen command must be defined. This may be linked to an event, activated by a pushbutton, etc. This example used three buttons with click commands to open the parameterized screen with each of the three parameter files.

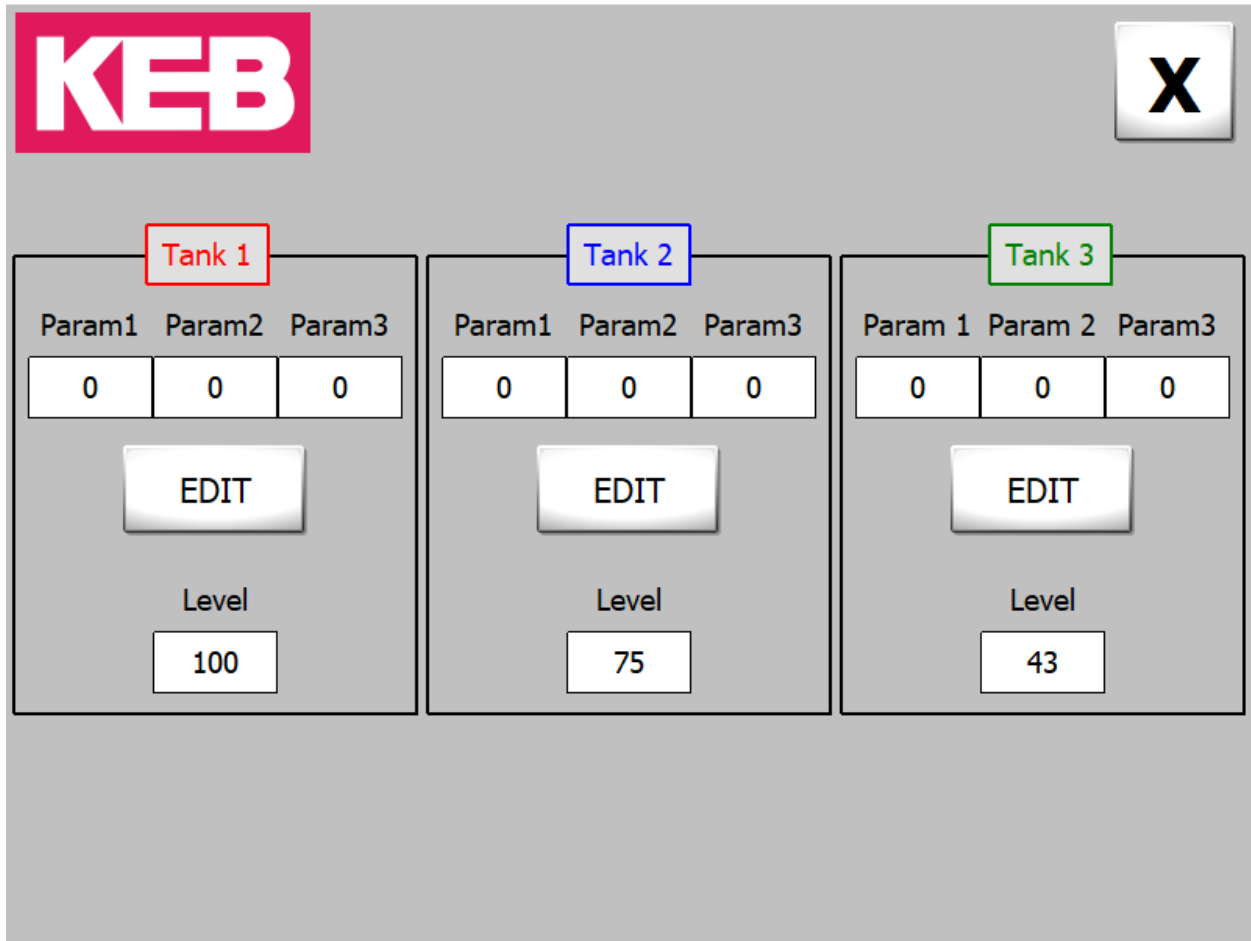


Fig. 8: Sample Main Screen

Each of the EDIT buttons in this example project triggers a different screen command for the Parameter File that corresponds to its tank. The three parameter values and tank level are displayed here as for reference.

From the *Command List* screen, select *New Command...* then select the *Screen* option in the *Command Type* window.

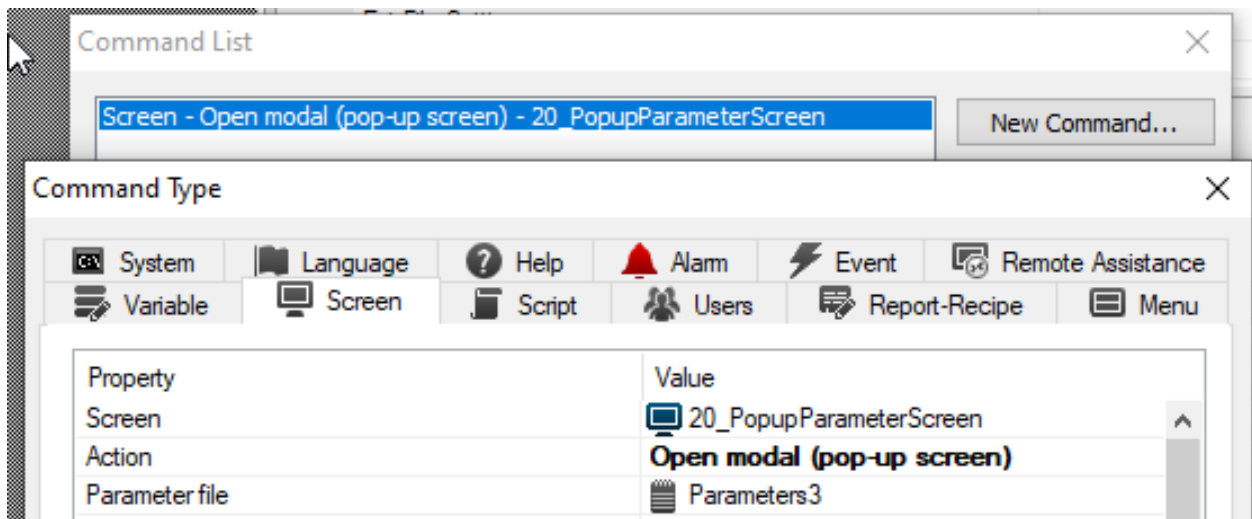


Fig. 9: Open Parameterized Screen Command

Browse for the parameterized screen in the *Screen* field with the "...". Select it, then select *OK*.

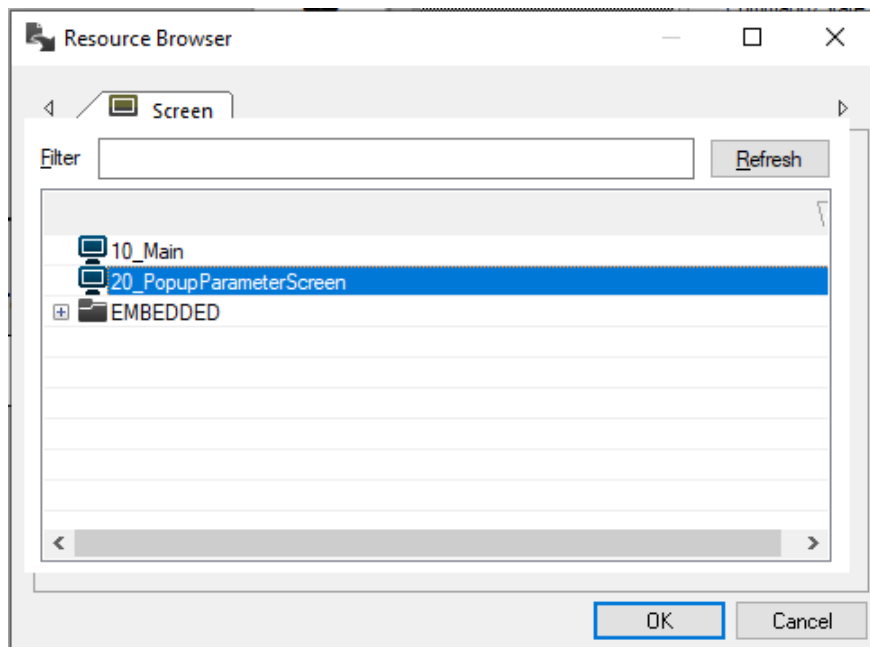


Fig. 10: Browse for Parameterized Screen

Select the screen action. This example uses a popup parameter screen, therefore the *Open modal (pop-up screen)* is selected.



## FAQ COMBIVIS Studio HMI



Select the appropriate *Parameter file* by browsing with the “...” icon. Select it, then select *OK*.

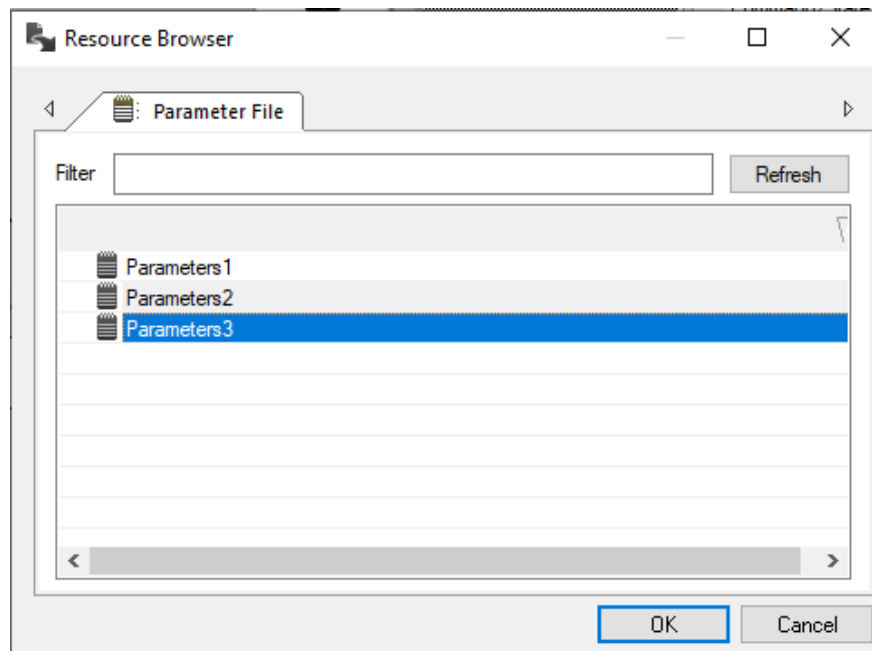


Fig. 11: Select Parameter File

## Disclaimer

KEB America, Inc. reserves the right to change/adapt specifications and technical data without prior notification. The safety and warning reference specified in this manual is not exhaustive. Although the manual and the information contained in it is made with care, KEB does not accept responsibility for misprint or other errors or resulting damages. The marks and product names are trademarks or registered trademarks of the respective title owners.

The information contained in the technical documentation, as well as any user-specific advice in verbal or in written form are made to the best of our knowledge and information about the application. However, they are considered for information only without responsibility. This also applies to any violation of industrial property rights of a third-party.

Inspection of our units in view of their suitability for the intended use must be done generally by the user. Inspections are particularly necessary, if changes are executed, which serve for the further development or adaptation of our products to the applications (hardware, software or download lists). Inspections must be repeated completely, even if only parts of hardware, software or download lists are modified.

**Application and use of our units in the target products is outside of our control and therefore lies exclusively in the area of responsibility of the user.**

### Americas:

KEB America, Inc.  
5100 Valley Industrial Blvd South  
Shakopee, MN 55379, USA  
(+1) 952-224-1400  
[info@kebamerica.com](mailto:info@kebamerica.com)

### Headquarters:

KEB Automation KG  
Suedstrasse 38  
D - 32683 Barntrup, Germany  
(+49) 5263 401-0  
[info@keb.de](mailto:info@keb.de)