HOW-TO

How-to Set Up Keyboards with Encoders in ADU

Document version: 1.0
Firmware: 120.0 or later
Published on: 05 May 2025

ECUMASTER

1. Description

This guide explains how to configure CANbus keyboards with rotary encoders in the Ecumaster ADU. From firmware version 120.0, two types of such keyboards are supported:

- **5x3MT Ecumaster Keyboard** – equipped with 13 standard buttons and 2 rotary encoders.
- **RacePad** – equipped with 8 standard buttons and 4 encoders. Mounted on a steering wheel.

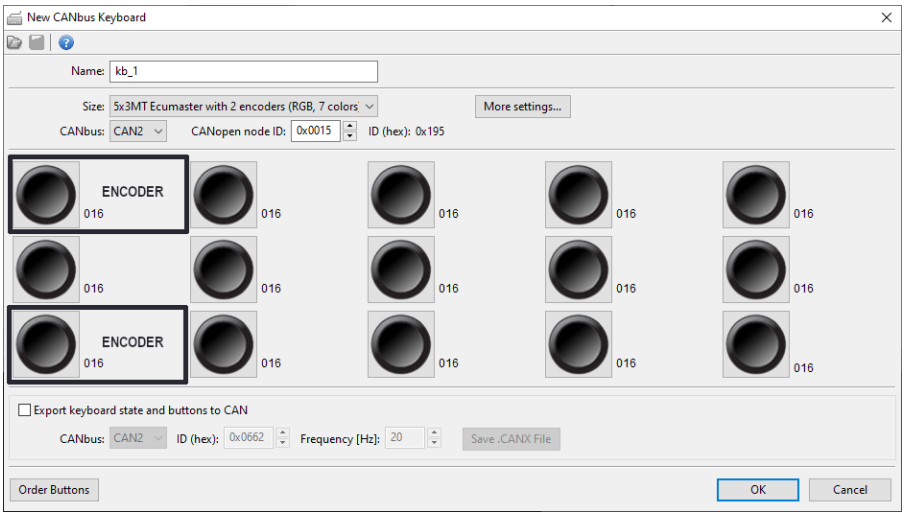
This document focuses mainly on the configuration options for the encoders. For more general information on CAN bus keyboard support, see the ADU user manual https://www.ecumaster.com/files/ADU/adu_manual_en.pdf.

Basic Configuration

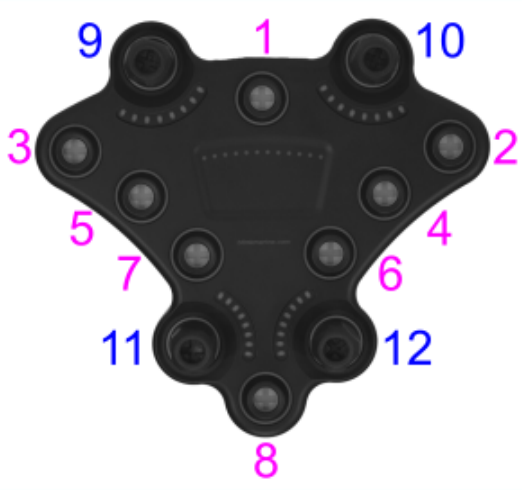
To configure any keyboard, click ***Add*** in the ***Project Tree*** and select ***CANbus Keyboard*** from the list, or click the keyboard icon in the toolbar.
Then, select the appropriate keyboard from the ***Size*** dropdown menu:

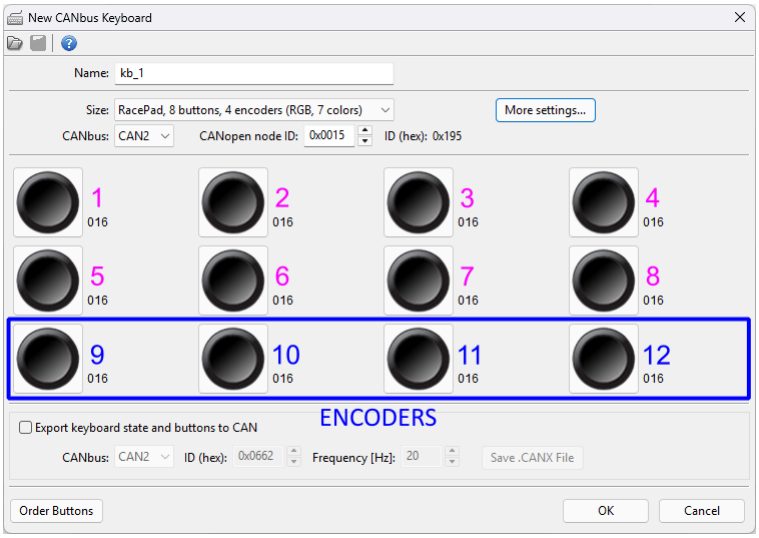
- 5x3MT Ecumaster with 2 encoders (RGB, 7 colors)
- RacePad, 8 buttons, 4 encoders (RGB, 7 colors)

5x3MT Ecumaster Keyboard configuration window:



RacePad configuration window:





Keyboard-Specific Settings

5x3MT Ecumaster Keyboard

Surrounding each encoder are 16 LEDs that visually represent the current state of the encoder. The default starting LED is at 9 o'clock, numbered clockwise.

To change this, go to ***More settings*** and adjust the ***Start offset for encoder LEDs*** from 0 to 15, where 0 corresponds to the 9 o'clock position.

RacePad

Each encoder features 8 arc-shaped LEDs. Additionally, a central LED bar with 12 LEDs is positioned in the middle of the device.

RacePad has several dedicated settings available in ***More settings***:

Parameter	Description
Invert LEDs encoder #1–4	Reverses the direction in which the LEDs light up for each encoder.
Invert encoder direction #1–4	Reverses the rotation logic. By default, turning clockwise increases value.
Enable central LEDs	Enables the center 12-LED strip.

Parameter	Description
Bi-directional	Enables symmetric (centered) visualization, e.g., for G-force or time gain/loss channels. In default mode, LEDs progressively light up from left to right. In bi-directional mode, LEDs light up outward from the center to the left or right, depending on the channel value.
Central LEDs channel	Assigns a channel to control the center LED strip, based on the channel's value.
Channel min value	Lower bound of the channel value range.
Channel max value	Upper bound of the channel value range.
Channel central value	Midpoint value used for symmetric LED visualization.

Encoder Operation Modes

Knowing the three types of encoder operations is essential for getting the best configuration:

1. ***encoder***

   Rotating the encoder adjusts the associated channel's value. Unlike a rotary switch, you have the flexibility to configure the encoder's range of values.

   - *First state* - the lowest encoder state
   - *Last state* - the highest encoder state
   - *Default state* - After startup, the encoder will adopt its default state unless the *Autosaved channels* feature is used. (Refer to the how-to guide document for detailed instructions on using *Autosaved channels* https://www.ecumaster.com/files/ADU/HowTo/How_to_Configure_Autosaved_Channels_in_ADU.pdf)

2. ***page changer***

   Turning the encoder clockwise moves to the next page, and turning counter-clockwise goes back to the previous page. In this mode, the LEDs illuminate in a fan-like pattern, offering a visual indication of the operating mode.

   - *Wrap pages* decide if the encoder should jump from the last page to the first and vice versa.

3. ***parameter controller*** and ***parameter selector***

   You can use the keyboard encoder to control multiple settings. This is possible by setting it to *parameter controller* mode. For each setting you want to manage, select a button and set

its mode to *parameter selector* mode. These selector buttons function like radio buttons, allowing you to choose one active parameter at a time that is modified by the knob.

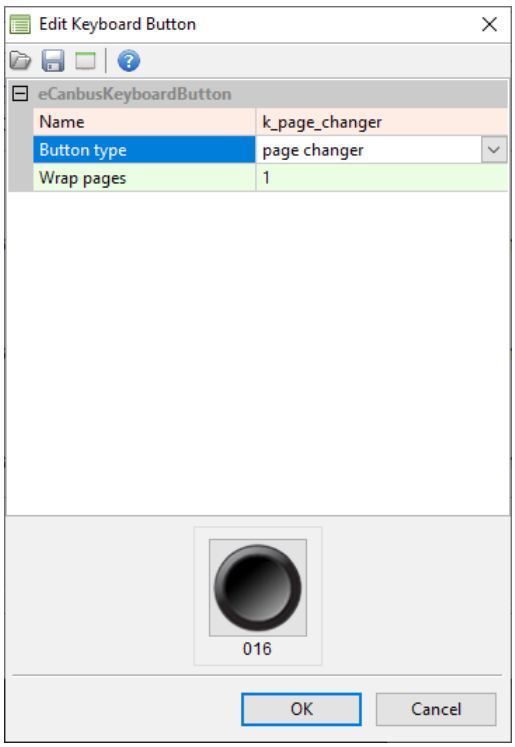
Each *parameter selector* comes with its own range and state, updating the *parameter controller* when pressed.

a. If there are no button presses defined as a *parameter selector*, *parameter controller* remains neutral and rotation has no effect.
b. Pressing the button defined as a *parameter selector*, conveys the parameter information to the encoder. Encoder can change the state of the *parameter selector*.
c. After pressing the currently selected *parameter selector* button again or pressing the *parameter controller*, the encoder returns to the neutral behaviour. (Pressing any other button (not defined as a *parameter selector*) will not cause the encoder to lose control of the *parameter selector*.)

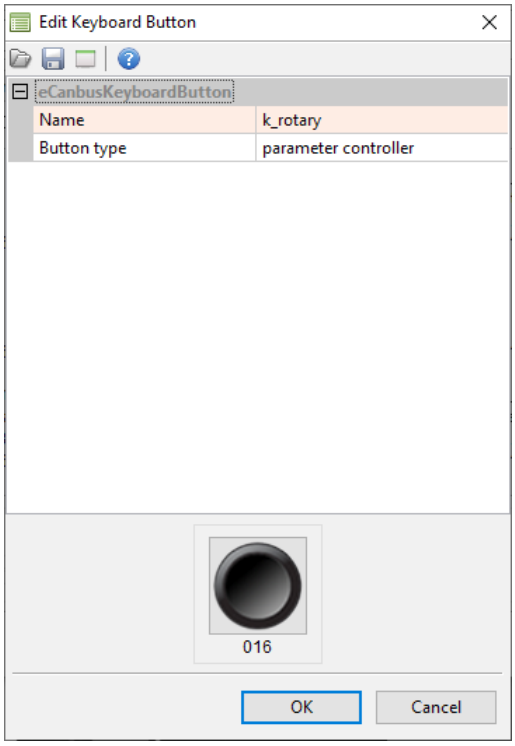
2. Example

We're configuring the 5x3MT to finely adjust four parameters and quickly navigate pages with the two rotary encoders. In this example we'll use the top encoder for pages and the bottom one to control parameters.

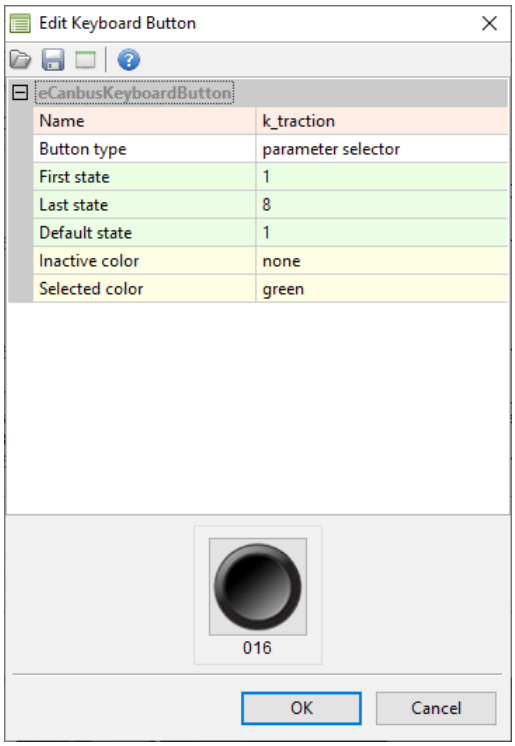
Let's start with the pages. In the keyboard settings, select the top-left box for the rotary encoder. Set it as a '*page changer*' and determine whether pages should wrap around.



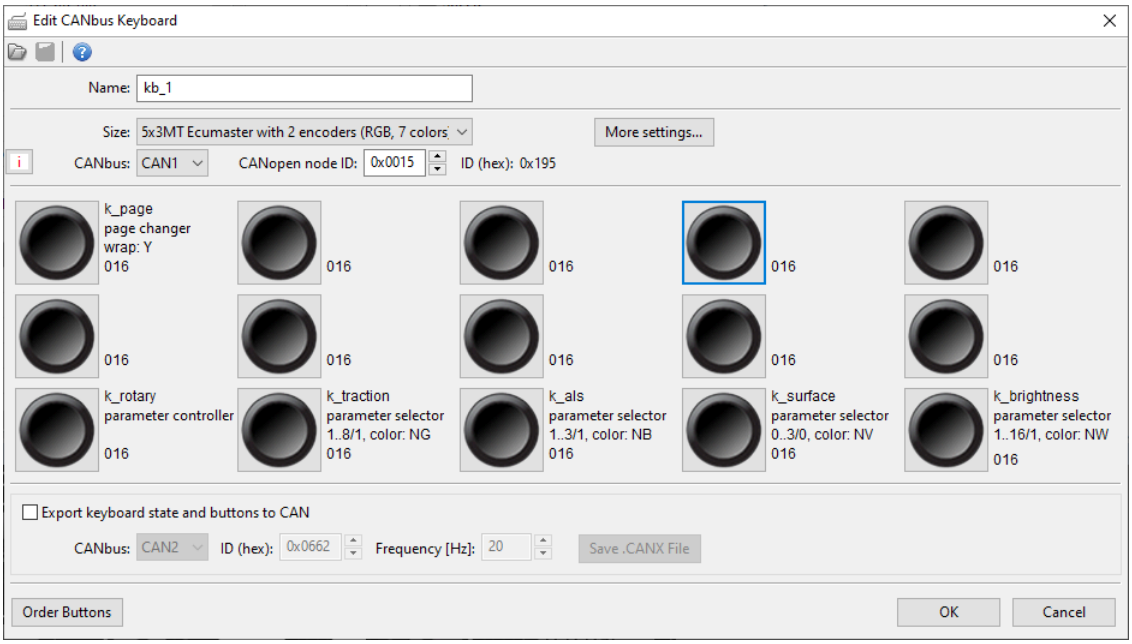
Moving on to our second goal, we aim to use the other encoder for controlling four parameters. Begin by selecting the bottom-left box and designate it as a '*parameter controller*'.



Now proceed to configure each parameter. For each of the four, choose one button, change its type to '*parameter selector*' and set the first, last, and default states to adjust their value range.



The final setup of our example keyboard looks as follows:



When the driver wishes to adjust a setting, they can press the corresponding button. The rotary encoder will activate, enabling them to change the parameter by turning it.

3. Document history

Version	Data	Changes
1.0	2025.05.05	Initial release