A3xx Rudder Trim LCD

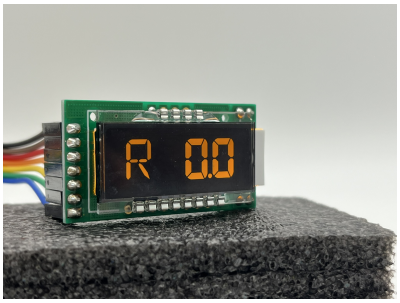
Kav Simulations A3xx Rudder Trim LCD

Thanks for your purchase of our bespoke Airbus A3xx Rudder Trim LCD. This document outlines the details of interacting with the component using our library.

The library consists of a '`.cpp`' and a '`.h`' file which gives you access to functions to display the required information on the display. This document is designed to talk you through how to use and call the provided functions.

If you wish to use the device 'plug-and-play' with MobiFlight, then this document may go into more detail than you need. But, if you want to understand a bit more about how the display works, then it's probably worth a read!

Ok, let's get started!



Installation

1. Copy the provided .cpp and .h files into your Arduino project folder.

2. Our library depends on the 'HT1621.h' library to communicate with the HT1621 LCD driver. You need to include the 'HT1621.h' and 'HT1621.cpp' files in your project as well. The files can be found on our GitHub repo for our custom firmware.

Usage

Initialisation

First, you need to connect the device as described in the Interface Configuration section below, and then initialise the LCD object and attach it to the correct pins. **If you're not using MobiFlight, you need to remove the '`#include "commandmessenger.h"`' line from the libraries if it is there. It has been removed from the demo code already.**

```
#include "KAV_A3XX_RUDDER_LCD.h"

KAV_A3XX_RUDDER_LCD lcd(CS_PIN, CLK_PIN, DATA_PIN);

void setup() {
  lcd.attach(CS_PIN, CLK_PIN, DATA_PIN);
}
```

Replace 'CS_PIN', 'CLK_PIN', and 'DATA_PIN' with the appropriate pin numbers for your setup.

We first create the Rudder object with the required pins. Then, we call the attach function which sets up the device using the `HT1621` library with the required settings, then puts the device into it's start state.

Display Functions

These functions are used to control the appearance of the LCD.

```
void setLeft(bool enabled);
void setRight(bool enabled);
void setDot(bool enabled);
void setValueInt(uint16_t state);
void showLeftValueInt(uint16_t value);
void showRightValueInt(uint16_t value);
```

```
// Display Functions Example:
lcd.setLeft(true);
lcd.setRight(false);
lcd.setDot(true);
lcd.setValueInt(2);
lcd.showLeftValueInt(20);
lcd.showRightValueInt(5);
```

Command Handling

The library also includes a function to handle MobiFlight commands. You can pass a string containing a command and it will be executed.

```
void handleMobiFlightCmd(char *cmd);
```

Example:

```
char command[] = "setLeft=1";
lcd.handleMobiFlightCmd(command);
```

Complete Example

This is just an example showing how functions are called and used. When you first plug the device in from new, it might take a few moments for the display to initialise.

```
#include "KAV_A3XX_RUDDER_LCD.h"

KAV_A3XX_RUDDER_LCD rudder(CS_PIN, CLK_PIN, DATA_PIN);

void setup() {
  rudder.attach(CS_PIN, CLK_PIN, DATA_PIN)
}

void loop() {
  // Demo State 1
  rudder.setLeft(1);
  delay(4000);

  // Demo State 2
  rudder.setRight(1);
  delay(4000);

  // Demo State 3
  rudder.showLeftValueInt(5);
  delay(4000);

  // Clear
  rudder.clearLCD();
}
```

API Reference

'void attach(byte CS, byte CLK, byte DATA)'

Attaches the LCD object to the specified pins and initialises the display. Parameters:

- 'CS': Chip Select pin
- 'CLK': Clock pin
- 'DATA': Data pin

'void detach()'

Detaches the LCD object and disables the display.

'void refreshLCD(uint8_t address)'

Refreshes the LCD display at the specified address. Parameters:

- 'address': The address to refresh on the display.

'void clearLCD()'

Clears the entire LCD display.

'void displayDigit(uint8_t address, uint8_t digit)'

Displays a digit at the specified address on the LCD. Parameters:

- 'address': The address to display the digit on the LCD.
- 'digit': The digit to display (0-9, 10 = dash, 11 = 't', 12 = 'd' and 13 = blank).

The addresses are predefined at the top of the file as 'DIGIT_ONE', 'DIGIT_TWO', 'DIGIT_THREE' and 'DIGIT_FOUR' for easier use.

Hardware Level Specification

Our device comes pre-soldered on a PCB with a driver chip, and everything is set up in our library above to make it all work. You shouldn't need anything more than the above. If you do, please contact us and let us know and we will provide you with the information you need.

Interface Configuration

- Pin 1: CS (Serial Shift Pulse Input)
- Pin 2: CLK (Serial Shift Pulse Input)
- Pin 3: DI (Serial Data Input)
- Pin 4: VSS (Power Ground)
- Pin 5: VDD: (Power Anode)
- Pin 6: BLA (Backlight Anode)
- Pin 7: BLK: (Backlight Ground)

Operating Data

- Operating Voltage: 4.8V 5.2V
- Operating Current: <300uA (5.0V)
- Operating Temperature: -10°C +60°C
- Storage Temperature: -20°C +70°C

Final Words

Thank you for taking the time to read this document, and we hope that you found it useful. If you do still have any questions, then please contact us at info@jak-kav.co.uk.