SLIDERBAR TECHNICAL REFERENCE

Motorized keyboard slider with an advanced controller

Features

- Motorized fader composed of a linear potentiometer and a belt and pulley system to move the fader
- Open hardware and open source
- USB Powered (USB Mini B connector)
- Desktop configuration GUI
- Programmable plugins
- USB CDC Virtual Serial Port for plug-n-play on any PC

Hardware specifications

- USB Mini B connector (power & communications)
- STM32F072C8T6 MCU
- 5-pin ST-Link V2 programming interface
- 5V to 3.3V regulator (MIC5205-33)
- 5V to 10V boost converter (MT3608) with trimmable potentiometer
- TB6612FNG motor driver (max 1.2 A)
- Pin headers for debug, motor out, and potentiometer in

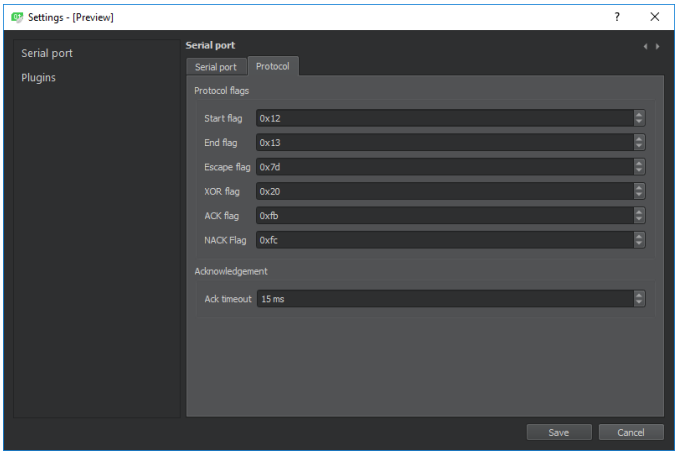

Figure 1. Configuration window

Figure 2. Motorized fader

Software specifications

- Desktop side:
  - Written in C++ with Qt for the GUI
  - IDE: Qt Creator
  - Base application allows user to manage settings, connect, load plugins and set activators for each plugin
  - Base application does nothing, the plugins parse the slider's input
  - Each plugin receives the SliderBar's position and can use the app's API to execute actions
- Embedded side:
  - Written in C++ with STM32CubeHal
  - IDE: Visual Studio Code with PlatformIO
  - Lightweight communication protocol.

Table of contents

1 Introduction

1.1 Description

1.1.1 General description

The SliderBar is a keyboard slider that mounts to or sits near your keyboard. It is based on a motorized fader as originally used in Audio mixer tables (See Figure 3).

Figure 3. 52-4261B Motor Fader Module

2 Hardware

2.1 General description

The USB connector provides power and communications to the MCU and motor.

The host (PC) controls the SliderBar via a lightweight USB communications protocol.

The STM32F072C8T6 MCU processes the commands and controls the motor accordingly through the TB6612FNG motor driver.

Position is read using the potentiometer provided in the fader.

The hardware consists of different modules:

- Microcontroller
- USB Mini B connector
- Programming interface
- 3.3V regulator
- 10V boost converter
- Motor driver
- Pin headers for debug, motor out, and potentiometer in

2.2 Modules description

2.2.1 Microcontroller

The microcontroller used is the STM32F072C8T6.

It was selected because it is cheap, low power, supports crystal-less USB 2.0 FS and is quite powerful enough.

2.2.2 USB Mini B connector

It is stronger than a Micro B connector.

2.2.3 Programming interface

Programming is done using an ST-Link V2 through the Serial Wire Debug (SWD) interface. There are 5 exposed pins: SWDIO, GND, SWCLK, V_{PP}, NRST (See Figure 4. Debug header). SWCLK is the clock of the SWD, SWDIO is the input output of the SWD.
V_{PP} is the programming voltage. Connect it to the 5V of the ST-Link V2.
Note: Do no exceed the limits of 3.5V and 8V (See Table 1. General operating conditions).

2.2.4 3.3V regulator

2.2.5 10V boost converter

2.2.6 Motor driver

2.2.7 Pin headers

2.2.7.1 Debug

2.2.7.2 Motor out

2.2.7.3 Potentiometer in

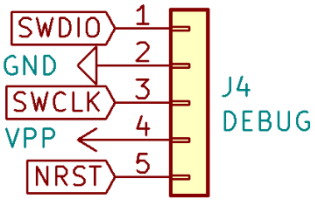

Figure 4. Debug header

3 Software

4 Electrical characteristics

4.1 Typical values

Unless otherwise specified, typical data are based on $T_A = 25$ °C, $V_{DD} = 5$V.

4.2 Operating conditions

4.2.1 General operating conditions

Table 1. General operating conditions

Symbol	Parameter	Conditions	Min	Typical	Max	Unit
V_{PP}	Programming supply voltage	-	3.5	5	8	V
V_{BUS}	USB supply voltage	-	4.75	5	5.25	V

4.2.2 Supply current characteristics

The current consumption is a function of several parameters and factors such as microcontroller state (run, sleep) and motor state (full speed, stop).

The current consumption is measured using a cheap USB volt/amp meter. See Figure 5.

Figure 5. USB Voltmeter

Table 2. Supply current characteristics

Symbol	Parameter	Conditions	Min	Typical	Max	Unit
I_{idle}	Idle current at V_{BUS}	-	8	10	20	mA
$I_{average}$	Average current at V_{BUS}	-	8	100	500	mA

5 Revision history