machdyne / **musli** Public

Müsli USB Pmod-compatible module

View license

9 stars 1 fork

Star Watch

Code Issues 1 Pull requests Actions Projects Security Insights

main

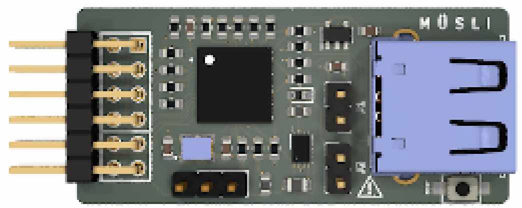
inc add host_musli_cdc ... on Feb 21 13

View code

Müsli USB Pmod™ compatible module

Overview

Müsli is a 12-pin Pmod™ compatible module with a RP2040 MCU and a USB type A port. Müsli can be configured to act as a USB host or USB device. In host mode the 5V USB VBUS power is supplied by a boost converter.



This repo contains documentation, schematics and example host/device firmware.

Find more information on the Müsli product page.

Modes

Careful attention should be paid to jumpers J1 and J2 to avoid unintended consequences, please read the *warnings* below.

J1	J2	Mode	Power
OPEN	OPEN	PMOD USB DEVICE	VSYS from PMOD. VBUS disconnected.
OPEN	SHORT	PMOD USB HOST	VSYS from PMOD. VBUS from VSYS.
SHORT	OPEN	USB STAND-ALONE DEVICE	VSYS from VBUS.

Warnings

- Do not plug the module into a PMOD socket when in STAND-ALONE DEVICE mode.
- Do not plug the USB cable into a computer when in USB HOST mode.

Linux Device Setup

On Linux you may need to configure udev rules to allow access to the device, for example by adding the following lines to your udev rules:

```
SUBSYSTEM=="usb", GROUP="plugdev", ATTR{idVendor}=="2e8a",
ATTRS{idProduct}=="0003"
SUBSYSTEM=="usb", GROUP="plugdev", ATTR{idVendor}=="2e8a",
ATTRS{idProduct}=="1025"
```

Firmware

This repo contains example firmware based on the Raspberry Pi Pico USB examples.

The host firmware example host_musli_kbd sends USB keyboard keypresses as ascii characters over the UART. It also acts as an SPI master and sends PS2 scancodes over the SPI pins. See the Zucker SOC for example Verilog code that acts as an SPI slave to receive the scancodes.

The device firmware example dev_musli implements a Vendor Specific Device that provides commands for bitbanging GPIO and performing SPI master data transfers. This firmware is used by ldprog to program Lone Dynamics FPGA boards.

To build the firmware you will need to install the Raspberry Pi Pico SDK.

Set the `PICO_SDK_PATH` environment variable to your SDK path.

```
$ git clone https://github.com/machdyne/musli
$ cd musli/firmware/dev_musli
$ mkdir build
$ cd build
$ cmake ..
$ make
```

Note: If your Müsli has a non-Winbond flash chip, you may need to build the firmware using the following commands:

```
$ cmake .. -DPICO_DEFAULT_BOOT_STAGE2_FILE=/your-pico-sdk-directory/pico-sdk/src/rp2_common/boot_stage2/boot2_generic_03h.S
$ make
```

This will create an ELF file that can be programmed via SWD as well as a UF2 file that can be copied to the USB Mass Storage Device when boot mode is selected (by holding the Müsli BOOT button during power-up).

Resources

- RP2040 Datasheet (PDF)
- Getting started with Raspberry Pi Pico (PDF)
- Raspberry Pi Pico Examples
- TinyUSB Examples

Pinouts

12-pin PMOD Header

```
1  2  3  4  5  6
7  8  9  10 11 12
```

Pin	Signal	Device Firmware	Host Firmware
1	GPIO0	UART0 TX	UART0 TX
2	GPIO1	UART0 RX	UART0 RX
3	GPIO2	GPIO	
5	GND		
6	PWR3V3		
7	GPIO8	SPI1 RX	
8	GPIO9	SS	SS
9	GPIO10	SPI1 SCK	SPI1 SCK
10	GPIO11	SPI1 TX	SPI1 TX
11	GND		
12	PWR3V3		

The GPIO numbers correspond with RP2040 GPIO numbers.

3-pin SWD Header

Pin	Signal
1	SWDIO
2	SWCLK
3	GND

Pin 1 is closest to the PMOD header.

Releases

No releases published

Packages

No packages published

Languages

● **C** 85.7% ● **CMake** 14.3%