MacUSBmicroExample Documentation

This software is licensed under the terms of the GNU Public License v3 as outlined in the LICENSE-gpl3.txt located at the root of the source distribution.

# Description of the Application

The application will search for U4x1 devices that are attached and show each of them on a list on the left. The view on the right shows details about the device including the status of each bit on each of the two ports. You can set the state of each pin so it can be used as input, output or 1-Wire (for 1-Wire temperature sensors).

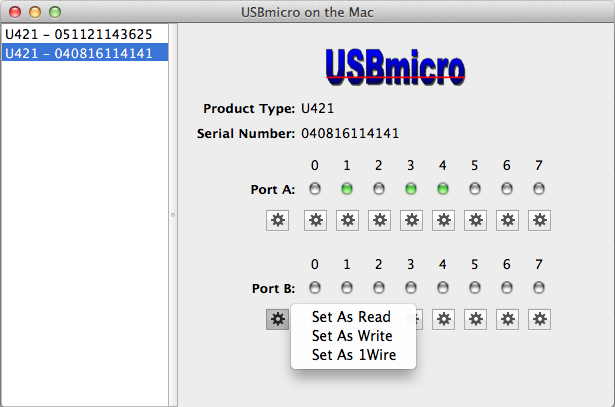
This program is to be used only as a sample for how to integrate with a U4x1 device and there is no warranty of any kind and no guarantee that it will work right. So you should make sure you thoroughly test any code that you use in your own programs. There is no formal support of any kind, but if you have questions, go to <http://www.homedomination.com/contact_us.php>, select the support checkbox, be sure to put MacUSBmicroExample in the subject line somewhere, and leave a detailed message.

# Running the Program

Make sure you connect your U4x1 devices before you start the program. When it starts, you’ll see the following window with the list of devices on the left, and info about the selected device on the right.



Each pin can be set to input, output or 1-Wire by clicking the gear button and choosing the desired option. The gear button beneath the port is used to set all the pins for that port.



Here’s what it looks like after setting all the pins on port B to Write, and clicking on pins 1 and 4:



If you have a temperature sensor attached to a data pin, you can set the pin to 1-Wire to get a continuous temperature sensor update. The supported sensor types are DS1822, DS18B20 and DS18S20. This picture shows pin 5 with a temperature sensor attached, and the pin set to 1-Wire.



# Source Code Highlights

## MainWindowController.m

This is the main UI controller that handles actions from the main window.

## USBmicroDevices.m

This auto detects the devices on the system, holds the list of devices, and monitors the devices.

## USBmicroDevice.m

Contains the product type and serial number for the device, and holds a USBmicroDevice object for each port. It also has a selector and object that is used to notify of changes. It has functions to monitor the device, write the port value, or read the port value.

## USBmicroDeviceIO.m

This object determines the actual command data that is used to send data to the device and receive data back from the device, and it initiates that communication.

## hid.c

This does the actual work of communicating with the U4x1 device.

## USBmicroPort.m

Contains the name of the port and an array of USBmicroPins. It has functions to monitor pins and check for changes to pins. It also has a selector and object that is used to notify of changes.

## USBmicroPin.m

This contains the status of the pin, has a selector and object that is used to notify of changes, and it also has an instance of TemperatureDS18x in case the pin is set to 1-Wire and needs to read temperature data.

## TemperatureDS18x.m

This is used to request an update from the temperature sensor and translate the resulting data into an actual temperature.