

# EasyUSB Datasheet

Preliminary Rev 0.3 – Nov/2010

## 1. Description

The EasyUSB is an embedded Universal Serial Bus interface with solid state memory and external asynchronous serial interface, designed to facilitate the file sharing and direct communication between host and device applications, therefore improving the end-user experience, reducing development costs and time to market.



On the host side, in almost any computer or portable device, no driver or further installation procedure is needed as the USB Mass Storage Class implemented is widely supported by almost any operating system.

On the device, the file system is accessible through a classical Universal Asynchronous Serial Interface. Eventually the software can be distributed inside the embedded memory, rendering it really plug-n-play and usable everywhere.

***Furthermore host computer and embedded controller can directly communicate over a dedicated serial link, built using an innovative technique<sup>1</sup>, without implementing the Communication Device Class. In practice it can benefit from both interfaces, mass storage and communication, using only the simple and widespread Mass Storage Class, eliminating the need of problematic composite peripherals or expensive custom driver development and maintenance.***

## 2. Specifications

USB MSC Specification 2.0 compliant  
File system: FAT16  
Memory size: 2MB to 16MB  
Typical Input Voltage: 5.0V  
Input Voltage Range: 3.0V to 5.2V  
Logic Signal: TTL/CMOS voltage levels  
Baud rate: 9600, 19200, 38400, 57600, 115200, 921600 bps  
Temperature: 0°C to 70°C  
Package: DIP8 style

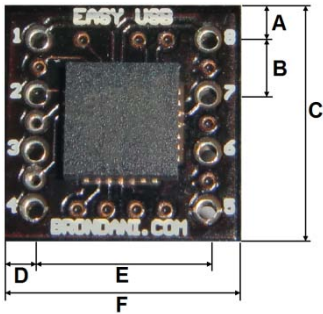
---

<sup>1</sup> Patent Application UD2010A000038

### 3. Dimensions

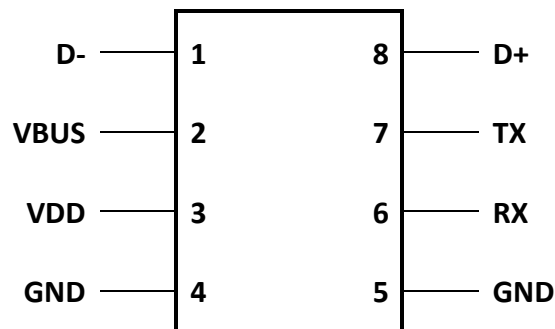
Part Number: EASYUSB-M01 (2MB)

The module fits a standard DIP8 socket:



Dimension	mm	inch
A	1.78	0.070
B	2.54	0.100
C	11.18	0.440
D	1.78	0.070
E	7.62	0.300
F	11.18	0.440
Thickness	2.54	0.100

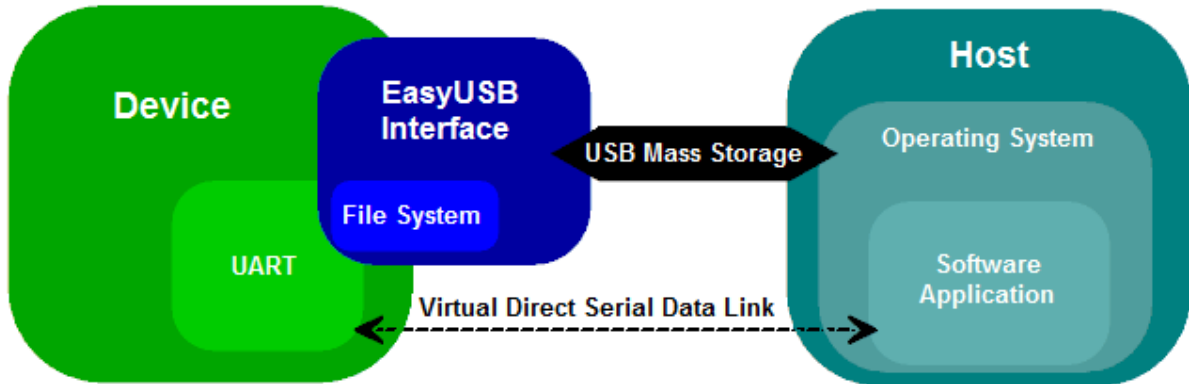
### 4. Pinout Description



Pin	Name	Type	Description
1	D-	I/O	USB Data Minus
2	VBUS	Input	VBUS Sense input
3	VDD	PWR	5V Power supply
4	GND	PWR	Ground supply
5	GND	PWR	Ground supply
6	RX	Input	UART Receive
7	TX	Output	UART Transmit
8	D+	I/O	USB Data Plus

## 5. Functional Description

With the EasyUSB Interface is easier and faster to add USB connectivity to an embedded system. Using a common UART port, the device can access the file system to store and share files and also communicate in real time with software applications.



### 5.1. Host

The host recognizes the EasyUSB interface as a standard mass storage device when it's plugged in. No further user operation is needed.

The UART interface is virtually direct accessed by the host through the library functions, detailed on the EasyUSB API documentation. External formatting is not supported.

A software application that demonstrates all the library functions is available.

### 5.2. Device

The EasyUSB is easily controlled by embedded systems through the standard UART interface.

When the VBUS is high (USB host is plugged) it outputs the message "Serial Bridge Mode" and acts as a bridge between the UART signals and the library read/write functions.

When VBUS is low (USB host is unplugged), it outputs the message "File System Mode" and the embedded processor can access the file system using the commands detailed in the next section.

## 6. Commands

The commands can be transmitted through the UART interface when the File System Mode is active. After the command prompt ">" (0x3E), the byte code corresponding to the desired command must be issued followed by the respective parameters.

If the byte <ESC> (0x1B) is sent, the command execution is canceled and it returns to the command prompt.

### 6.1. Read File

Description: Reads the content of a file stored on the file system root. If the file does not exist, the message "File Not Found" is returned.

Byte code: "r" (0x72)

Parameters: String "filename.ext" where:

"filename" = name of the file with a length from 1 to 8 characters.

"ext" = extension of the file with a length of 3 characters.

Return data: "Read File: filename.ext" <CR><LF>  
<read data bytes>

### 6.2. Write File

Description: Appends up to 32 bytes to a file. If the file does not exist, it is created. If the length is out of range, the message "Invalid Length" is returned.

Byte code: "w" (0x77)

Parameters: String "filename.ext" "nn" <data bytes> where:

"filename" = name of the file with a length from 1 to 8 characters.

"ext" = extension of the file with a length of 3 characters.

"nn" = length of the data to be written from "00" to "32".

<data bytes> = data to be written

Return data: "Write File: filename.ext" <CR><LF>  
"Length: nn" <CR><LF>  
<written data bytes>

### 6.3. Delete File

Description: Deletes a file. If the file doesn't exist, the message "File Not Found" is returned.

Byte code: "d" (0x64)

Parameters: String "filename.ext" where:

"filename" = name of the file with a length from 1 to 8 characters.

"ext" = extension of the file with a length of 3 characters.

Return data: "Delete File: filename.ext" <CR><LF>

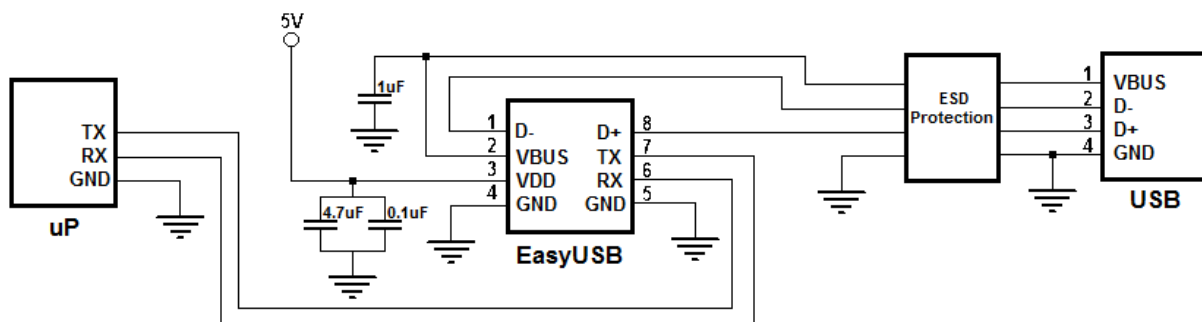
### 6.4. Format

Description: Formats the memory.  
 Byte code: "f" (0x66)  
 Parameters: "y" to confirm, "n" to cancel  
 Return data: "Format Disk? (y/n)" <CR><LF>

### 6.5. Erase

Description: Erase and formats the memory.  
 Byte code: "e" (0x65)  
 Parameters: "y" to confirm, "n" to cancel  
 Return data: "Erase Disk? (y/n)" <CR><LF>

## 7. Typical Application Circuit



## Contacts

This is a preliminary release of the EasyUSB project.

For technical questions, software, documentation and for information about producing, distributing, licensing, please contact the developer:

[easyusb@brondani.com](mailto:easyusb@brondani.com)

### Disclaimer

The information in this document is believed to be accurate in all respects at the time of publication but is subject to change without notice. The developer assumes no responsibility for errors and omissions, and disclaims responsibility for any consequences resulting from the use of information included herein. Additionally, the developer assumes no responsibility for the functioning of not described features or parameters. The developer reserves the right to make changes without further notice. The developer makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does the developer assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. Unless specifically provided otherwise, the products are not suitable for, and shall not be used in, automotive applications and are not intended, authorized, or warranted for use as components in applications intended to support or sustain life. No freedom to use patents or other intellectual property rights is implied by the publication of this document.