USBizi User Manual

Rev. 2.03 Date: June 5, 2009 User Manual

Document Information

Information	Description
Abstract	This document covers complete information about USBizi - *pronounced USB easy* -, specifications, tutorials, and references.
related documents	Embedded Master Manual

GHI Electronics

Table of Contents

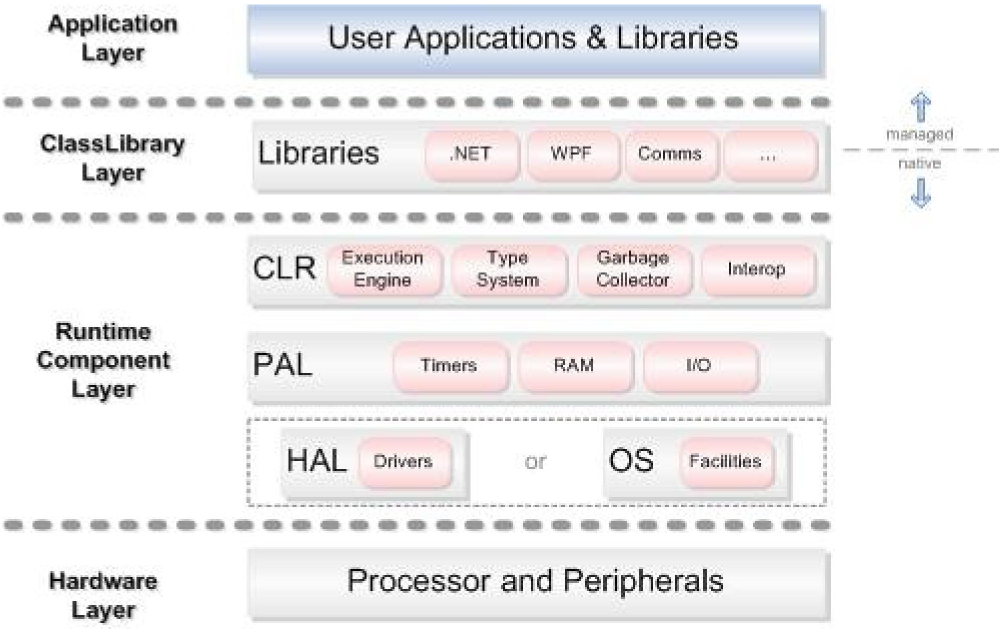
1. Introduction

1.1. What is The Microsoft .NET Micro Framework

The Microsoft .NET Micro Framework combines the reliability and efficiency of managed code with the premier development tools of Microsoft Visual Studio to deliver exceptional productivity for developing embedded applications on small devices.

The .NET Micro Framework brings a rich, managed-code environment to smaller, less expensive, and more resource-constrained devices. Requiring only a few hundred kilobytes of RAM and an inexpensive processor, the .NET Micro Framework was built from the ground up to let you build applications using familiar Visual Studio development tools.

With .NET Micro Framework SDK, you can develop your embedded solutions in C# using a subset of the .NET libraries focused on embedded applications. Your development environment is Visual Studio, where you can take advantage of its powerful editing, object browsing, project management, and debugging capabilities. These capabilities are available when using the .NET Micro Framework SDK’s extensible device emulation system or on real hardware.



1.2. What is USBizi

USBizi chip *-pronounced as USB easy-* is a headless .NET Micro Framework system based on LPC2388\LPC2387 chipset. It is basically a subset of Embedded Master which provides many of the great features Embedded Master offers, with everything implemented on a **single chip**. This is the smallest and lowest cost Micro Framework device that still implements unique features like USB host.

Its limitations include reduced Flash and RAM size than Embedded Master and no TCP/IP nor LCD/Graphics libraries are included. However, an LCD can be connected and controlled using some of the many GPIOs available or SPI.

Comparison between Embedded Master and USBizi

	Embedded Master	USBizi
RAM	Over 8000 KB	96 KB
FLASH	Over 4500 KB	512 KB
User RAM	Over 6000 KB	About 40 KB
User Flash	Over 3000 KB	About 150 KB
Native Graphics	Yes	No
Native Ethernet	Yes	No
GHI Native Library	Yes	Yes

Developers can use .NET Micro Framework and write managed code (C#) without having to deal with much complexity. USBizi is fully licensed to use with .NET Micro Framework.
In addition to .NET Micro Framework standard features, Embedded Master Module has many exclusive additional features that are not supported in other .NET Micro Framework platforms, such as USB Hosting, Analog inputs, Analog outputs, PWM, CAN and more.

For a complete reference on USBizi features and functions, please refer to Embedded Master User Manual.

USBizi Packages

USBizi chipset is available in two packages LQFP100/144. LQFP100 Chipset is identical to the LQFP144 version except it doesn't contain a USB host. USB device is till available and work exactly the same way.

1.3. Example applications:

- Data logger

- MP3 Player
- Measurement tool or Tester
- Robotics
- GPS navigation
- Medical instrumentation
- Industrial automation devices
- Other small devices that require an efficient, low-resource-consuming Microsoft .NET client

1.4. Key features

Software

USBizi has most of Embedded Master's library features except Native TCP/IP and Graphics Libraries that requires more FLASH and RAM resources that could not be needed in USBizi application.

However, user can employ the various GPIOs and serial interfaces to control external devices to add those features:

- ✔ Graphics Support: using an LCD with SPI-based graphics accelerator, such as SSD1339 LCD controller.
  http://www.sparkfun.com/commerce/product_info.php?products_id=763
- ✔ Ethernet and TCP/IP Support: Using a Hardwired TCP/IP Embedded Ethernet Controller, such as WIZnet W5100 chip.

Feature	Supported
.NET Micro Framework V3.0	√
Hibernate - Low Power Mode	√
GPIO	√
Native Graphics Library	Not Supported, However using an LCD with SPI based graphics accelerator, such as SSD1339 LCD controller , could easily add this feature to USBizi
Native TCP/IP Library	Not Supported, However Using a Hardwired TCP/IP Embedded Ethernet Controller, such as WIZnet W5100 chip, could easily add this feature to USBizi
COM1	√
COM2	√
COM3	√
COM4	√
SPI1 8-bit or 16-bit	√
SPI2 8-bit or 16-bit	√
I2C	√

Feature	Supported
1-wire	√
PWM	√
FAT File System	√
SD/MMC/SDHC	√
ADC	√
DAC	√
CAN	√
USB Host	√
USB Host Hub	√
USB Host Mass Storage	√
USBH HID (Mouse, Keyboard, Joystick)	√
USB Host Printer	√
USB Host CDC	√
USB Host to Serial FTDI	√
USB Host to Serial Prolific	√
USB Host to Serial SiLabs	√
USB Sierra C885, 3G Modem	√
USB Host Low Level Access	√
USB Device (Client)	√
USB Device Mouse Direct Support	√
USB Device CDC (VCOM) Direct Support	√
Simple Pin Capture	√
Native Register Access	√
Native functions to Set debug interface	√
Battery RAM	√
User controlled WatchDog	√
Managed Application Protection	√

Hardware

- 72Mhz ARM Processor
- 96 KB RAM
- 512 KB FLASH
- Embedded USB host/device
- 44 GPIO

- 35 Interrupt Inputs
- 2 SPI
- I2C
- 4 UART
- 2 CAN Channels
- 8 10-bit Analog Inputs
- 10-bit Analog Output
- 4-bit SD Memory card interface
- 6 PWM
- 100 mA everything enabled
- 200uA Hibernate Mode
- -40ºC to +85ºC Operational
- RoHS Lead Free

2. Pin-Out And Description

The LPC2387/8 72Mhz ARM7 32-bit processor is the core of USBizi. The processor has a wide range of peripherals that adds a lot of functions and features to USBizi such as PWM, GPIO, LCD Controller, USB HC ...etc.

Most signals on USBizi are multiplexed to offer more than one function for every pin. It is up to the developer to select which one of the functions to use. GHI drivers and .NET Micro Framework does some checking to make sure the user is not trying to use two functions on the same pin. The developer should still understand what functions are multiplexed so there is no conflict. For example, analog channel 3 (ADC3) and the analog output (AOUT) are on the same pin. Either function can be used but not both of them simultaneously.

- MAC and EMC pins cannot be accessed directly, however, they can be used using the ***Register*** class provided by GHI native library.
- Digital I/O pins are named Exx, where xx is an assigned number.

2.1. USBizi^{TM} LQFP 144 Pin-out table

USBizi LQFP144 chipset is based on LPC2388 from NXP. This chip is used in USBizi Development System.

Pin #	Description	Pin #	Description	Pin #	Description	Pin #	Description
1	TDO	37	USBD-2	73	P2.12/SD_DAT2	109	P0.9/MOSI1/E6*
2	N/A	38	N/A	74	N/A	110	N/A
3	TDI	39	N/A	75	P2.11/SD_DAT1	111	P0.8/MISO1/E8*
4	TMS	40	VBUS	76	BL#/P2.10/E0*	112	P0.7/SCK1/E10*
5	TRST#	41	VDD1	77	VDD3	113	P0.6/SSEL1/E12*
6	VCC	42	P0.29/USBD+1	78		114	VDD5
7	TCK	43	P0.30/USBD-1	79	VSS5	115	P0.5/TD2/E14*
8	P0.26/ADC3/AOUT /E22*	44	VSS2	80	P0.22/SD_DAT0	116	P0.4/RD2/E16*
9	N/A	45	N/A	81	N/A	117	VSS7
10	P0.25/ADC2/E24*	46	P1.18/PWM1/E35	82	P0.21/MCIPWR	118	P4.28/TXD3/E13
11	P0.24/ADC1/E26*	47	P1.19/USB_PWR_FN/E34	83	P0.20/SD_CMD	119	VSS8
12	N/A	48	P0.14/USB_CON2	84	N/A	120	P4.29/RXD3/E17
13	P0.23/ADC0/E28*	49	P1.20/PWM2/E25	85	P0.19/SD_CLK	121	VDD(DC-DC-3V)_3
14	VDDA	50	P1.21/PWM3/E23	86	P0.18/MOSI0/E41*	122	N/A
15	VSSA	51	P1.22/UAN_PWR_RD	87	P0.17/MISO0/E40*	123	N/A

Pin #	Description	Pin #	Description	Pin #	Description	Pin #	Description
16	N/A	52	N/A	88	N/A	124	N/A
17	VREFA	53	N/A	89	P0.15/SCK0/E42*	125	N/A
18	VDD(DC-DC-3V)_1	54	N/A	90	P0.16/SSEL0/E43*	126	N/A
19	N/A	55	N/A	91	N/A	127	N/A
20	RSTOUT#	56	N/A	92	P2.9/E37*	128	N/A
21	N/A	57	N/A	93	P2.8/E36*	129	N/A
22	VSS1	58	N/A	94		130	N/A
23	RTCX1	59	VSS3	95	P2.7/RTS1/E11*	131	N/A
24	RESET#	60	VDD(DC-DC-3V)_2	96	P2.6/E9*	132	N/A
25	RTCX2	61	P1.27/USB_PWR_OC/E21	97	P2.5/PWM6/E7*	133	N/A
26	ALARM	62	VDD2	98	N/A	134	N/A
27	VBAT	63	N/A	99	P2.4/PWM5/E2*	135	N/A
28	P1.31/ADC5/E32	64	N/A	100	P2.3/PWM4/MODE/E4*	136	N/A
29	P0.12/ADC6/E30*	65	VSS4	101	N/A	137	N/A
30	P1.30/ADC4/E29	66	P0.0/RD1/E19*	102	VDD4	138	VDD6
31	XTAL1	67	P0.1/TD1/E15*	103	VSS6	139	VSS9
32	P0.13/ADC7/E27*	68	N/A	104	N/A	140	N/A
33	XTAL2	69	P0.10/SDA2/E39*	105	P2.2/CTS1/E1*	141	P0.2/TXD0//E18*
34	P0.28/SCL0/E31* (open drain)	70	P0.11/SCL2/E38*	106	P2.1/RXD1/E3*	142	P0.3/RXD0/E20*
35	P0.27/SDA0E33* (open drain)	71	N/A	107	P2.0/TXD1/E5*	143	RTCK
36	USBD+2	72	P2.13/SD_DAT3	108	N/A	144	N/A

* Interrupt capable input.

2.2. USBizi^{TM} LQFP 100 Pin-out table

USBizi LQFP100 chipset is based on LPC2387 from NXP.

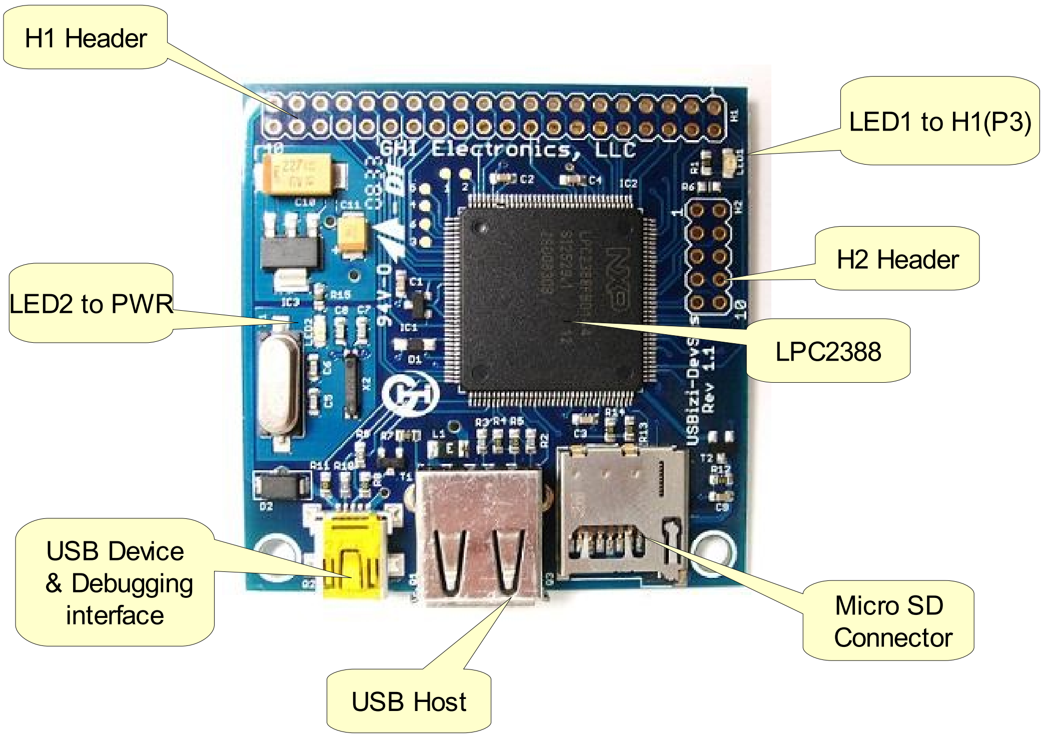
Pin #	Description	Pin #	Description	Pin #	Description	Pin #	Description
1	TDO	26	N/A	51	P2.12/SD_DAT2	76	P0.9/MOSI1/E6*
2	TDI	27	N/A	52	P2.11/SD_DAT1	77	P0.8/MISO1/E8*
3	TMS	28	VDD1	53	BL#/P2.10/E0*	78	P0.7/SCK1/E10*
4	TRST#	29	P0.29/USBD+1	54	VDD2	79	P0.6/SSEL1/E12*
5	TCK	30	P0.30/USBD-1	55	VSS4	80	P0.5/TD2/E14*
6	P0.26/ADC3/AOUT /E22*	31	VSS2	56	P0.22/SD_DAT0	81	P0.4/RD2/E16*
7	P0.25/ADC2/E24*	32	P1.18/PWM1/E35	57	P0.21/MCIPWR	82	P4.28/TXD3/E13
8	P0.24/ADC1/E26*	33	P1.19/USB_PWR_ FN/E34	58	P0.20/SD_CMD	83	VSS7
9	P0.23/ADC0/E28*	34	P1.20/PWM2/E25	59	P0.19/SD_CLK	84	VDD(DC-DC-3V)_3
10	VDDA	35	P1.21/PWM3/E23	60	P0.18/MOSI0/E41*	85	P4.29/RXD3//E17
11	VSSA	36	P1.22/UAN_PWR_ RD	61	P0.17/MISO0/E40*	86	N/A
12	VREFA	37	N/A	62	P0.15/SCK0/E42*	87	N/A
13	VDD(DC-DC-3V)_1	38	N/A	63	P0.16/SSEL0/E43*	88	N/A
14	RSTOUT#	39	N/A	64	P2.9/USB_CONN	89	N/A
15	VSS1	40	N/A	65	P2.8/E36*	90	N/A
16	RTCX1	41	VSS3	66	P2.7/RTS1/E11*	91	N/A
17	RESET#	42	VDD(DC-DC-3V)_2	67	P2.6/E9*	92	N/A
18	RTCX2	43	P1.27/USB_PWR_ OC/E21	68	P2.5/PWM6/E7*	93	N/A
19	VBAT	44	N/A	69	P2.4/PWM5/E2*	94	N/A
20	P1.31/ADC5/E32	45	N/A	70	P2.3/PWM4/MODE /E4*	95	N/A
21	P1.30/ADC4/VBUS/ E29	46	P0.0/RD1/E19*	71	VDD3	96	VDD4
22	XTAL1	47	P0.1/TD1/E15*	72	VSS5	97	VSS6
23	XTAL2	48	P0.10/TXD2/E39*	73	P2.2/CTS1/E1*	98	P0.2/TXD0/E18*
24	P0.28/SCL0/E31* (open drain)	49	P0.11/RXD2/E38*	74	P2.1/RXD1/E3*	99	P0.3/RXD0/E20*
25	P0.27/SDA0E33* (open drain)	50	P2.13/SD_DAT3	75	P2.0/TXD1/E5*	100	RTCK

* Interrupt capable input.
Note: E27,E30 and E36 are not supported in 100 package.

3. USBizi Development System

It is a (2.2x2.2 inches) development system based on USBizi chipset that exposes all peripherals. This Development System is very easy to use and can be powered through USB. This board can be used as an OEM board that could be suitable for many product Ideas.

The following gives a brief description of USBizi Development System components



3.1. USBizi^{TM} LQFP144 -LPC2388-

The LPC2387/8 72Mhz ARM7 32-bit processor is the core of USBizi. The processor has a wide range of peripherals that adds a lot of functions and features to USBizi such as PWM, GPIO, LCD Controller, USB HC ...etc.

3.2. H1 Header

Exposes many USBizi Pins to the user

H1 Pin	H1 Name	LPC2387/8 Pin	LPC2387/8 Name	USBizi C#
1	BL#	76	P2.10	E0 *
2	CTS1	105	P2.2	E1*
3	PWM5 (LED1)	99	P2.4	E2*
4	RXD1	106	P2.1	E3*
5	PWM4/MODE	100	P2.3	E4*
6	TXD1	107	P2.0	E5*
7	MOSI1	109	P0.9	E6*
8	PWM6	97	P2.5	E7*
9	MISO1	111	P0.8	E8*
10	P2.6	96	P2.6	E9*
11	SCK1	112	P0.7	E10*
12	RTS1	95	P2.7	E11*
13	SSEL1	113	P0.6	E12*
14	TXD3	118	P4.28	E13
15	TD2	115	P0.5	E14*
16	TD1	67	P0.1	E15*
17	RD2	116	P0.4	E16*
18	RXD3	122	P4.29	E17
19	TXD0	141	P0.2	E18*
20	RD1	66	P0.0	E19*
21	RDX0	142	P0.3	E20*
22	USB_PWR_OC	61	P1.27	E21

H1 Pin	H1 Name	LPC2387/8 Pin	LPC2387/8 Name	USBizi C#
23	ADC3 AOUT	8	P0.26	E22*
24	PWM3	50	P1.21	E23
25	ADC2	10	P0.25	E24*
26	PWM2	49	P1.20	E25
27	ADC1	11	P0.24	E26*
28	ADC7	32	P0.13	E27*
29	ADC0	13	P0.23	E28*
30	ADC4	30	P1.30	E29
31	RESET#	24	RESET#	N/A
32	ADC6	29	P0.12	E30*
33	SCL0 (open drain)	34	P0.28	E31*
34	ADC5	28	P1.31	E32
35	SDA0 (open drain)	35	P0.27	E33*
36	VBAT	27	VBAT	N/A
37	GND	N/A	GND	N/A
38	USB_PWR_EN	47	P1.19	E34
39	+5Volt	N/A	N/A	N/A
40	PWM1	46	P1.18	E35

3.3. H2 Header

Compatible with UEXT header used by olimex modules, MP3 decoder for example.

H2 Pin	H2 Name	LPC2387/8 Pin	LPC2387/8 Name	USBizi C#
1	VCC	N/A	N/A	N/A
2	GND	N/A	N/A	N/A
3	TXD2	93	P2.8	E36*
4	RXD2	92	P2.9	E37*
5	GPIO	70	P0.11	E38*

H2 Pin	H2 Name	LPC2387/8 Pin	LPC2387/8 Name	USBizi C#
6	GPIO	69	P0.10	E39*
7	MISO0	87	P0.17	E40*
8	MOSI0	86	P0.18	E41*
9	SCK0	89	P0.15	E42*
10	SSEL0	90	P0.16	E43*

3.4. Micro SD Connector

Micro SD cards are internally identical to full size SD cards. Developers can use the card to store billions of bytes.

3.5. USB Device

This connector is connected to the internal USB device peripheral of USBizi. This is also used for debugging and deployment of applications. This port also can be used to power up USBizi.

3.6. USB Host

USBizi USB Host connector. This connector supplies 5V to the connected USB device through a 250mA resettable fuse.

4. Communicating With USBizi

By default, USBizi communication is done through USB (BootLoader and Firmware), but the user can select communication over UART. This is done by using the pin **P2.3/PWM4/MODE.** This pin is pulled up high internally and sampled at power up, in bootloader and firmware. If the pin is high (default), USB is used. When connecting this pin to ground, UART communications will be enabled on UART0 (COM1) and USB will be stopped.

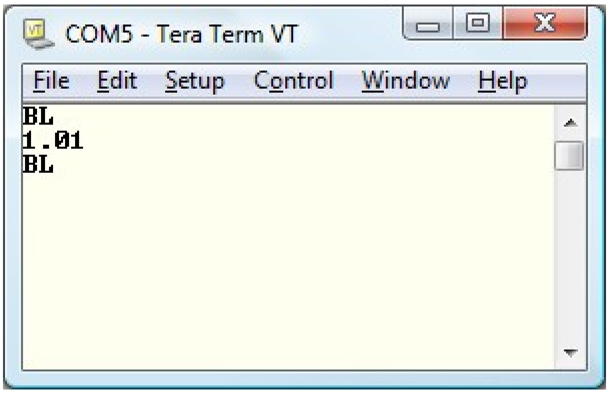
The pin can still be used as GPIO or PWM after power up.

4.1. Boot Loader Interface

The boot loader interface is needed to update USBizi firmware or to erase your application and download the firmware again. This might be needed if an application caused Micro Framework TinyCLR to become unresponsive. The boot loader is active if it detected a corrupted firmware or is activated by the user. For details on using the Boot Loader, see USBizi Boot Loader section.

When connecting USBizi in boot loader mode to your PC (no firmware loaded for example) for the first time, Windows will ask for the USB drivers. The driver is included with the SDK, it has the name ***"USBizi - Bootloader -Interface.inf"***. Simply point Windows to the file and it should install the driver. This driver works on XP and Vista.

When USB cable is plugged in, a virtual COM port will be installed on your PC to communicate with the boot loader. Look for the new installed COM port, open a terminal window and you can start using Boot Loader commands. Note that some errors can occur if you have opened a COM port and then reseted USBizi, the open COM port will no longer work. The user must close the COM port first, reset USBizi and then open it again.



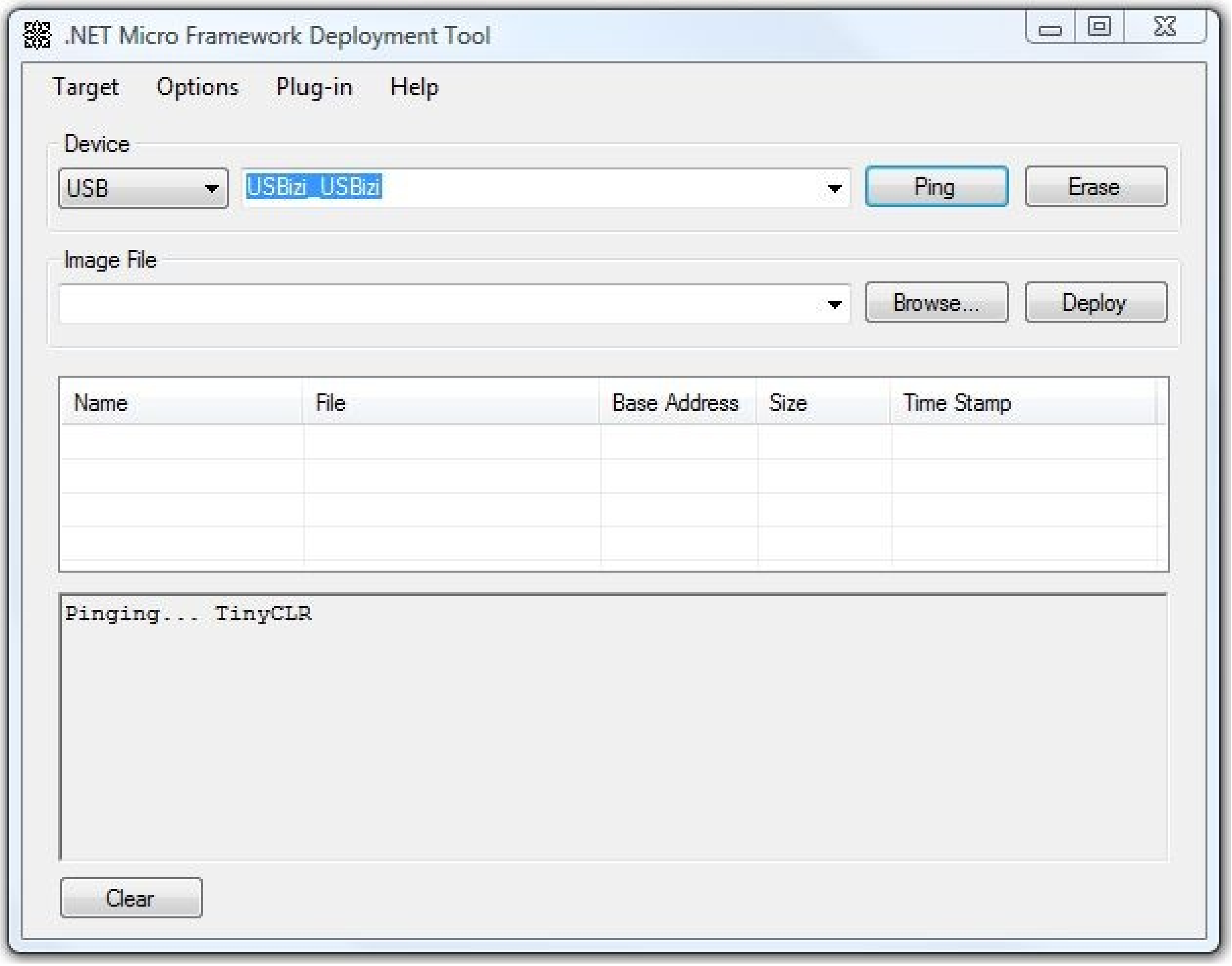
```
BL
1.01
BL
```

When using UART, you can open the COM port at 115200 baudrate, 8 data bits, no parity, 1 stop bit and no handshaking. In UART case, use UART0 (COM1) pins on USBizi.

4.2. Firmware Interface

When connecting USBizi, in firmware mode, to your PC for the first time, it will need a USB driver. The driver is included with the USBizi downloads, it has the name ***"GHIMFInterafce.inf"***. Simply point Windows to the file and it should install the driver.

This interface is used to communicate with Micro Framework and deploy applications. When driver is loaded properly, you can ping USBizi using MFDeploy.



Device: USB — USBizi_USBizi

```
Pinging... TinyCLR
```

When using UART, you can select **Serial** instead of **USB** for **Device** in MFDeploy or in Visual Studio when deploying the application.

5. USBizi Boot Loader

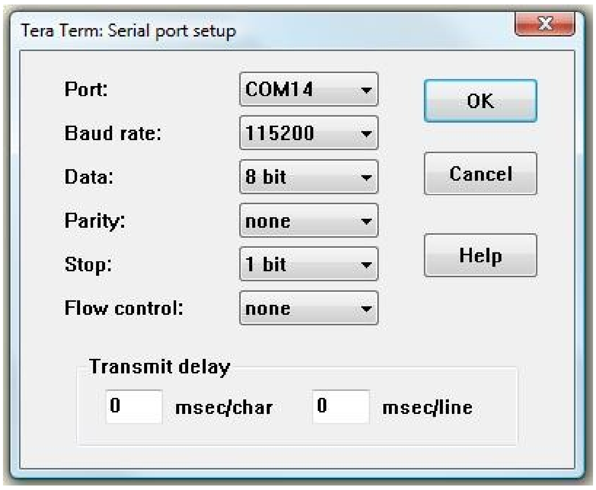
USBizi Boot Loader is a software developed by GHI and is included on all USBizi modules. It is used to update the firmware (including .NET Micro framework libraries) of USBizi.

Note that TinyBooter (Microsoft's Boot Loader) is not available on USBizi, only on Embedded Master. This will not limit the user application.

5.1. Using The Boot Loader

The Boot Loader can be enabled by using the **BL#** pin or through a special command on UART0. **BL#** is pulled high (internally) and is sampled on power up. If the pin is high (default), the firmware is verified and executed; otherwise, if the pin is low (connected to ground) then boot loader mode will be entered in either USB or Serial mode, depending on P2.3/PWM4/MODE pin.

Enabling Boot Loader through UART



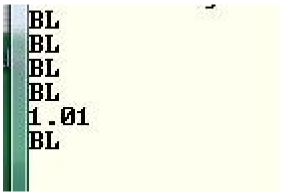
Tera Term: Serial port setup

Port: COM14
Baud rate: 115200
Data: 8 bit
Parity: none
Stop: 1 bit
Flow control: none

Transmit delay
0 msec/char 0 msec/line

1. The Boot Loader uses UART0 (COM1) at 115200, 8N1. Connect UART0 through an RS232 circuit to your PC. At the PC, you can have a Terminal program connected to the appropriate COM port (COM1 on USBizi has nothing to do with COM port number on PC).
2. It is recommended to use the **BL#** pin to enter the BootLoader. There is another option: in the PC terminal window keep sending “%” character continuously and quickly. Note the quotes are used here just to isolate the character.
3. While doing step 2, reset USBizi.

4. After pressing reset while sending “%”, and with under a second, the terminal should show BL and will keep sending BL for every character entered.
5. If at this point you do not see BL then do not proceed. Go back and check the steps again.
6. Enter V (upper case) and you will see back the GHI loader version number.



```
BL
BL
BL
BL
1.01
BL
```

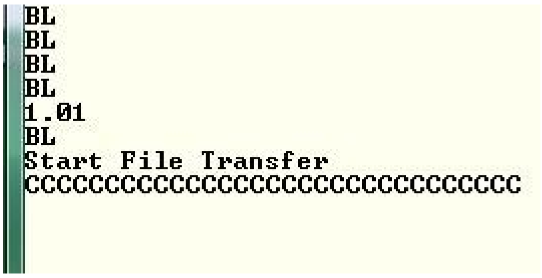
The Boot Loader supports the following commands

Command	Description
V	Get GHI Loader version number
E	Erase all memory
X	Load new USBizi Firmware
N	Display serial number
R	Run Firmware
L	Load managed application
G	Read managed application
D	Delete managed application
P	Disable reading managed application

5.2. Firmware Update

USBizi Firmware

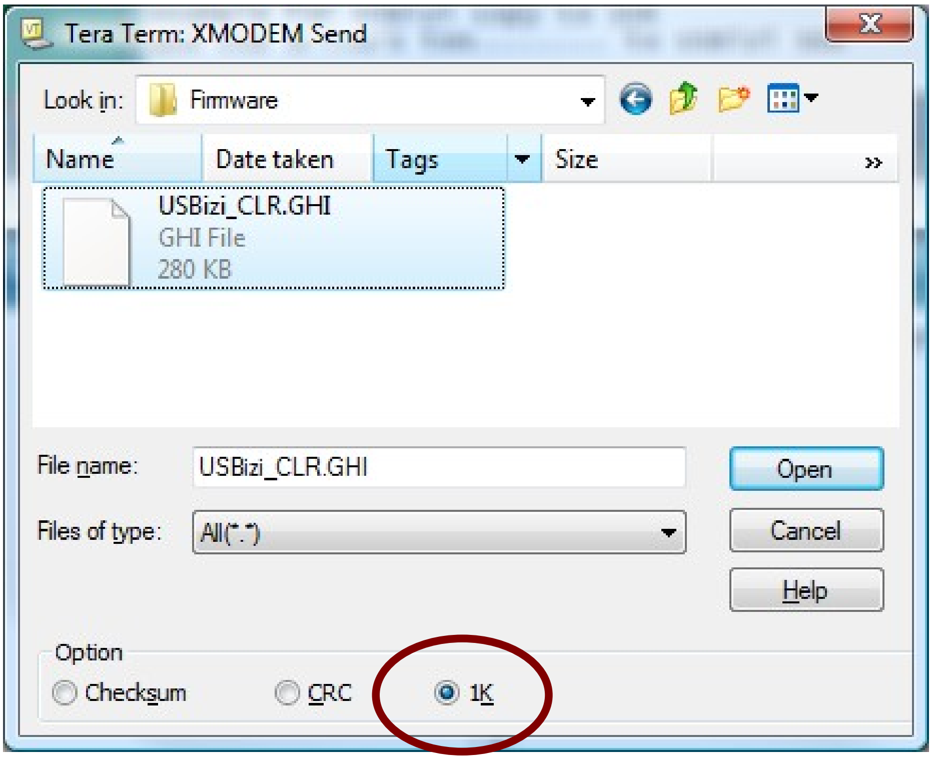
Loading new Firmware is simple but it requires a terminal that supports XMODEM file transfer. XMODEM has many versions, **GHI Loader requires 1K transfers** with 16-bit CRC error checking. Once the X command is entered, GHI Loader will start sending back character C continuously. This C is an indicator for XMODEM that a device is waiting for data. After you see character C coming on the terminal window, you can now select XMODEM transfer and point the software to **USBizi_CLR.GHI** file. Updating the firmware takes very few seconds to load and when loading is done and the file is valid, the new firmware is executed automatically.



```
BL
BL
BL
BL
1.01
BL
Start File Transfer
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

HyperTerminal software comes with Windows and it supports XMODEM 1K transfers but Windows Vista does not come with any terminal software. A free terminal software that is tested to work with GHI Loader is "TeraTerm Pro" which is available with USBizi downloads on GHI website.

MFDeploy cannot be used to update firmware on USBizi.



Tera Term: XMODEM Send

Look in: Firmware

USBizi_CLR.GHI
GHI File
280 KB

File name: USBizi_CLR.GHI

Files of type: All(*.*)

Option

○ Checksum ○ CRC ● 1K

Managed Application

Managed application update from the Bootloader is added to Bootloader version 1.06 and above. This is similar to updating USBizi firmware in the previous section, but instead it updates the managed application. Also, the managed application can be protected from reading. This is useful if you need to protect your managed application against copying, tampering or disassembling. Although extensive testing is done on this feature, GHI cannot guarantee and is not responsible for the possibility of hacking or bypassing protection.

Here are the expected steps in developing an application for USBizi:

1. Develop the application and deploy using Visual Studio.
2. When the application development is done, the user will read the application using the BootLoader **G** command. This is using XMODEM 1K Receive mode.
3. In production, the application is simply loaded using the **L** command and XMODEM 1K Send mode.
4. If read protection is needed, the user can use the **P** command.

Note: All USBizi products will start shipping with Bootloader 1.06 or above. In case you received older Bootloader installed, it cannot be upgraded. Please contact GHI if you need to get USBizi products with the new Bootloader installed.

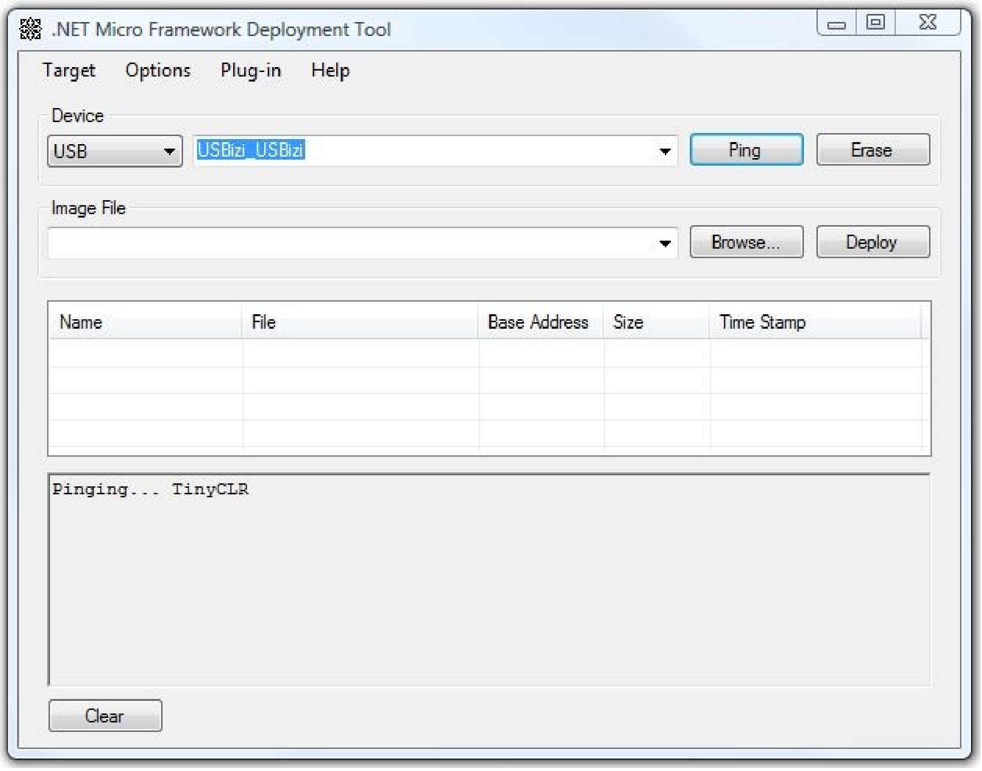
6. Getting Started With USBizi

In this section, we will deploy a simple program to USBizi.

First connect USBizi using the USB Device connector to your PC which will provide power for the device. It will ask you for drivers. Point it to ***"GHIMFInterafce.inf".***

Once installed, let's ping the device using MFDeploy to make sure the device is responsive. MFDeploy is a tool provided by Microsoft to Erase and Deploy new Firmware. Note that on USBizi, the firmware (TinyCLR) is updated using the GHI Boot Loader not MFDeploy. MFDeploy is also included with Microsoft .NET Micro Framework 3.0 SDK Tools.

Make sure USBizi is connected and run MFDeploy. Under "Device" Select USB. Note the USBizi device *"USBizi_USBizi"* and click Ping. USBizi which is running .Net Micro Framework will respond with "TinyCLR".



Device: USB | USBizi_USBizi

Image File:

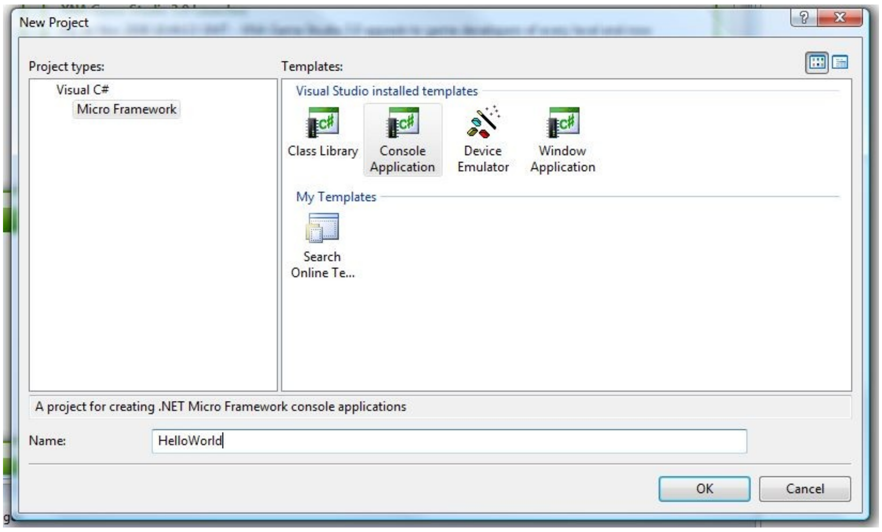
```
Pinging... TinyCLR
```

This assures that USBizi is running and responding properly.

Loading “Hello World!”

This final step requires Visual Studio 2008 or Visual C# 2008 Express Edition. It also requires .NET Micro Framework SDK 3.0.

Start a new console project from .NET Micro Framework template



New Project

Project types: Visual C# > Micro Framework

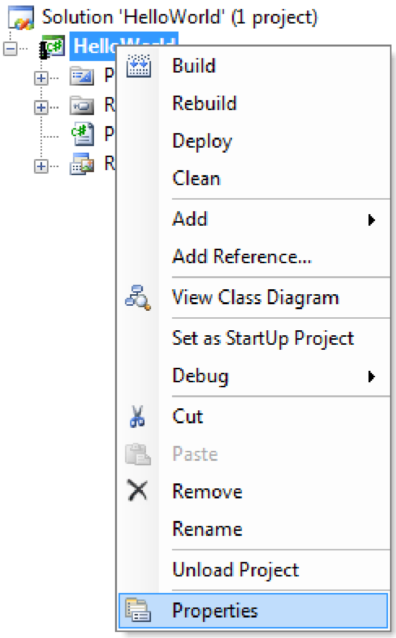
Templates: Visual Studio installed templates: Console Application

A project for creating .NET Micro Framework console applications

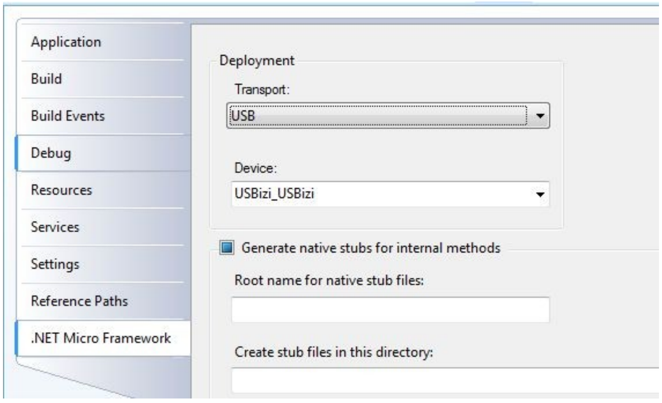
Name: HelloWorld

We now need to set the USB device port so Visual Studio will know where to send the managed application.

Right click on the project and select “properties” as showing below:



From the “Micro Framework” tab, select USB for Transport.



Deployment

Transport: USB

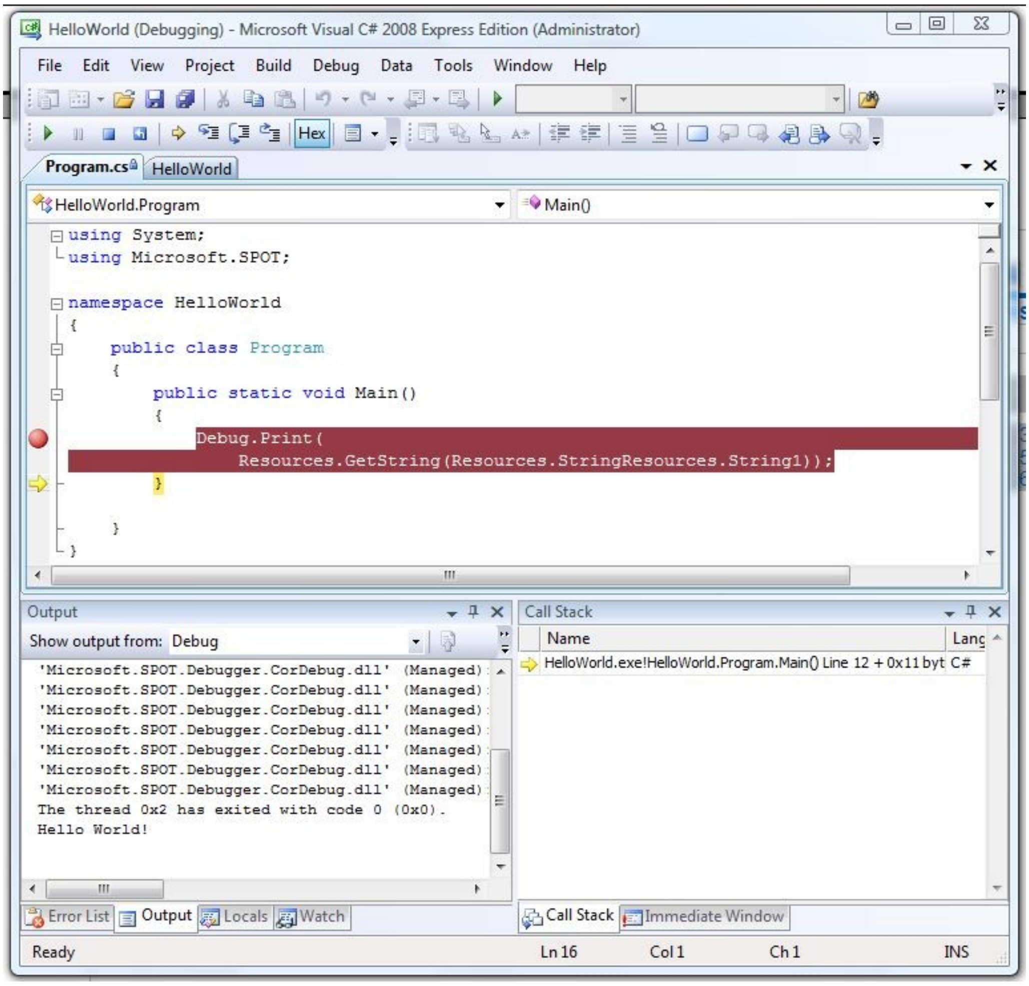
Device: USBizi_USBizi

☐ Generate native stubs for internal methods

Root name for native stub files:

Create stub files in this directory:

Now, press F5 key and the program will load and run on the device. In this case, we have set a breakpoint in the code so we can stop it and show you the output window.


HelloWorld (Debugging) - Microsoft Visual C# 2008 Express Edition (Administrator)
File Edit View Project Build Debug Data Tools Window Help
Hex
Program.cs
HelloWorld
HelloWorld.Program
Main()
Error List
Output
Locals
Watch
Call Stack
Immediate Window
Ready
Ln 16
Col 1
Ch 1
INS

```
using System;
using Microsoft.SPOT;

namespace HelloWorld
{
    public class Program
    {
        public static void Main()
        {
            Debug.Print(
                Resources.GetString(Resources.StringResources.String1));
        }

    }
}
```

Output

Show output from: Debug

```
'Microsoft.SPOT.Debugger.CorDebug.dll' (Managed)
'Microsoft.SPOT.Debugger.CorDebug.dll' (Managed)
'Microsoft.SPOT.Debugger.CorDebug.dll' (Managed)
'Microsoft.SPOT.Debugger.CorDebug.dll' (Managed)
'Microsoft.SPOT.Debugger.CorDebug.dll' (Managed)
'Microsoft.SPOT.Debugger.CorDebug.dll' (Managed)
'Microsoft.SPOT.Debugger.CorDebug.dll' (Managed)
The thread 0x2 has exited with code 0 (0x0).
Hello World!
```

Call Stack

Name	Lang
HelloWorld.exe!HelloWorld.Program.Main() Line 12 + 0x11 byt	C#

7. Further Information

Since USBizi is a subset of Embedded Master, please refer to Embedded Master Manual for complete details on features and functions.

Licensing

USBizi is fully licensed. The module price covers the commercial use of USBizi with .Net Micro Framework.

Disclaimer

IN NO EVENT SHALL GHI ELECTRONICS, LLC. OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT IABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE AND/OR HARDWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. SPECIFICATONS ARE SUBJECT TO CHANGE WITHOUT ANY NOTICE. GHI ELECTRONICS, LLC LINE OF PRODUCTS ARE NOT DESIGNED FOR LIFE SUPPORT APPLICATIONS.