TEAleaf-USB™

USB Authentication Key for Software Licensing and Revenue Protection

Summary

TEAleaf-USB is a low cost authentication system which may be used to verify that a software product is not an unauthorized pirate copy. The host PC uses a simple but robust algorithm to verify that a TEAleaf-USB device is present, using a 128-bit encryption key.

A complete set of BOM and Gerber blueprints is available off-the-shelf manufacture of ultra low cost *TEAleaf-USB* hardware security keys.

TEAleaf-USB uses the Human Interface Device (HID) USB profile. It does not require USB drivers and is immediately plug-and-play compatible with present and future Windows, Linux and Mac operating systems.

TEAleaf-USB is available in 28-pin DIL and 20-pin SSOP packages. They require only a few discrete components.

Applications

- Computer software copy protection and licensing
- Pay-per-use hardware protection
- Random number generation

Security

- Extended Tiny Encryption Algorithm (xTEA)
- 128 bit security key
- Uncrackable random-hashed USB communications during authentication
- Quantum limited true random number generator

Features

- True USB 2.0 HID plug and play - No drivers required
- Ultra low cost single chip solution
- Low speed USB, can use a low cost resonator
- Security key, product name, manufacturer name, serial number, GUID configurable from USB interface, minimizing development costs
- No Vendor ID / Product ID registration required
- Authentication success indication
- All-Systems-Go indication
- Tx / Rx indication
- Low power indication
- 5 digital, analog, interrupt virtual I/O
- 122-byte EEPROM

Firmware Factory Ltd
2 Marshall St, 3rd Floor
London W1F 9BB, UK
sales@firmwarefactory.com
support@firmwarefactory.com

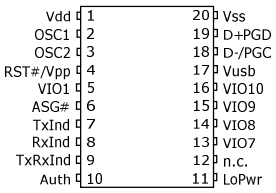


Figure 1a - SSOP pinout

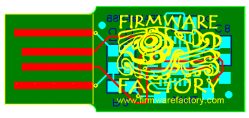


Figure 1b (Shown actual size)
BOM & Gerber blueprints available for off-the-shelf manufacture.
Complete units also available.

Table 1. Device Pinout

DIL	*SSOP*	*Name*	*Description*
20	*1*	*Vdd*	Power positive input
9	*2*	*OSC1*	Oscillator output
10	*3*	*OSC2*	Oscillator input
1	*4*	*RST#* *Vpp*	Reset input (active low) TEAclipper Vpp
5	*5*	*VIO1*	Software configured virtual I/O
6	*6*	*ASG#*	All-Systems-Go output
7	*7*	*TxInd*	Tx Indication output
23	*8*	*RxInd*	Rx Indication output
24	*9*	*TxRxInd*	Tx/Rx Indication output
18	*10*	*Auth*	Authentication Success output
11	*11*	*LoPwr*	Low Power Indication output
12	*13*	*VIO7*	Software configured virtual I/O
13	*14*	*VIO8*	Software configured virtual I/O
22	*15*	*VIO9*	Software configured virtual I/O
21	*16*	*VIO10*	Software configured virtual I/O
14	*17*	*Vusb*	USB supply filter
15	*18*	*D-*	USB data -
27		*PGC*	TEAclipper PGC
16	*19*	*D+*	USB data+
28		*PGD*	TEAclipper PGD
8,19	*20*	*Vss*	Power ground reference

Active low
Unassigned pins should be grounded
SOIC packages have same pinout as DIL

Firmware Factory USB Product Family

- USB-232 asynchronous serial interface
- TEAleaf-USB security and authentication dongle
- expandIO-USB I/O expander
- USB-SPI synchronous serial interface
- USB-I2C synchronous serial interface
- USB-TakeOff managed power take-off, wakeup, standby and charge controller
- USB-DAQ data logger
- USB-Config configuration and diagnostic module
- WattLogic electricity consumption monitor
- AnniLogic anniversary reminder module

Electrical Specifications

Table 2. Electrical Specifications

Operating voltage, 18F2450	4.2V – 5.5V*
Operating voltage, 18F14K50	3.3V – 5.5V*
Typical/max supply current, Vdd = 5.0	10mA / 21mA
Operating Temperature	-40°C to +85°C

Basic Operation

To the PC ('host'), TEAleaf-USB looks like a Human Interface Device (HID) with which it may exchange information using simple commands. The commands can be used to authenticate the presence of the TEAleaf-USB chip, and to provide auxiliary functions.

Dedicated Pin Functions

The pin functions are shown in table 1 and are described in detail below. Note that the output pins are in a tri-state condition until ~20µs after power-on.

Vss, Vdd, Vusb

Vss is the power supply ground reference. Vdd should be connected to a regulated supply, for example the USB bus power. Vusb should be connected, via a 470nF capacitor, to Vss. See for example C8 in figure 2.

OSC1, OSC2

OSC1 and OSC2 should be connected to a 12MHz parallel cut crystal circuit with 22pF capacitors. It may be replaced with a 12MHz resonator with 1.5% total tolerance, e.g. Murata CSTCE12M0G55-R0.

Vpp, PGC, PCD

TEAclipper programming pins. Refer to the Delivery and Programming section for details. Note that the Vpp pin may be subject to voltages as high as 12V during programming.

Reset

The pin is an active low reset input. It should be connected via a 22k resistor to Vdd.

All-Systems-Go Indication

Active low output that indicates when the TEAleaf-USB is configured and not suspended, and so up power may be drawn from the bus if required. If it is low, up to 500mA may be drawn if the Low Power output is low, or 100mA if the Low Power output is high. If it is high, no more than 100µA may be drawn.

Tx Indication

Output for connecting to a transmit indication LED. It outputs high for approximately 100ms when data has been transmitted to the host.

Rx Indication

Output for connecting to a receive indication LED. It outputs high for approximately 100ms when data has been received from the host.

Tx / Rx Indication

Output for connecting to a transmit / receive indication LED. It outputs high for approximately 100ms when data has been transmitted to or received from the host.

Low Power Indication

Output which is high when the device must draw no more than 100mA from the bus, rather than the maximum power it has been configured for.

Authentication Success Indication

Auth outputs high if the last command / response exchange was successful, or low otherwise. This may be used in pay-per-use hardware applications.

VIO Pins

The VIO pin functions can be controlled using the Set Pin command. On power-up, they are digital inputs. The available functions are given below.

Digital Input

Digital Input is a general purpose input. Its state can be read using the Get Pin command. This setting is available on any VIO pin.

Digital Output

Digital Output is a general purpose output. Its state can be set using the Set Pin command and read using the Get Pin command. On power-up and reset, it will initialize to the inactive state. This setting is available on any VIO pin except VIO0.

Interrupt

Interrupt is an input whose state can be read using the Get Pin command. When it transitions from the inactive state to the active state, it will generate am Interrupt response. Interrupts must be on VIO9 or VIO10. If this pin is connected to a switch it should be de-bounced to avoid generating multiple Interrupt responses.

Analog Input

Analog Input is an analog input whose voltage can be read using the Get Analog command. This setting is available on VIO1 only on 28-pin devices and VIO10 only on 20-pin devices.

Device Fuses

Fuses are non-volatile settings you may select to customize your device. For information on how to modify them, refer to the device configuration section.

Write Lock

Once the write lock bit is set, all commands that change the device strings and fuses will have no effect. Unless otherwise configured, the default is unlocked.

Max Bus Power

You can use draw power from the USB port if required. The maximum power required by your product is specified by the Max Bus Power fuse. It allows the host to balance its power budget, and is subject to certain limitations:

1. No device may consume more than its Max Bus Power specification at any time, and never more than 500mA.
2. No device may consume more than 100mA unless the Low Power output pin indicates it permitted to do so.
3. If the or the All-Systems-Go output goes low, the host is in sleep mode and the product may draw no

more than 100µA from the bus (not including the power consumed by the TEAleaf-USB chip).

Unless otherwise configured, the default value is 100mA.

Remote Wakeup

A device may be configured to implement remote wakeup. In this mode, an interrupt pin may be used to wake up the host. This feature requires remote wakeup to be supported but the generic HID driver on the host, and so performance is not guaranteed.

Custom VID / PID

Personalized Vendor and Product IDs are not required. However, you may customize them if you wish. Unless otherwise configured, the default Vendor ID is 0x0B40, and the default Product ID 0x011E for the 28-pin device and 0x011F for the 20-pin device.

Device Strings

Device strings are non-volatile Unicode strings stored by the TEAleaf-USB and which may be read by the host PC and all its applications. For information on how to modify them, refer to the customization section.

Product Name

The manufacturer name is a Unicode string of up to 61 characters plus zero terminator. The host application can read this data using a Get Feature request for string 1. The host PC commonly displays this string while it is installing the default HID driver when it is first inserted. Unless otherwise configured, the default value is "TEAleaf-USB".

Manufacturer Name

The manufacturer name is a Unicode string of up to 61 characters plus zero terminator. The host application can read this data using a Get Feature request for string 2. The host PC commonly displays this string while it is installing the default HID driver when it is first inserted. Unless otherwise configured, the default value is "Firmware Factory Ltd".

Serial Number

The Serial Number data is a Unicode string of up to 61 characters plus zero terminator. The host application can read this data using a Get Feature request for string 3. The Serial Number is a unique string which you can use to differentiate one physical device from other devices with the same TEAleaf-USB Vendor ID / Product ID / Product GUID combination. Unless otherwise configured, the default value is a unique value.

Product GUID

The product GUID is a Unicode string of up to 61 characters plus zero terminator. The host application can read this data using a Get Feature request for string 4. The product GUID is a string which you can use to make a hardware security key specific to a particular software application. It is used to differentiate it from other security keys with the TEAleaf-USB Vendor ID / Product ID combination. It should be the same for all products of the same type. Unless otherwise configured, the default value is "No GUID".

Config (EEPROM) String

The configuration data is a Unicode string of up to 61 (122 bytes) characters plus zero terminator. You can use it as you wish to store configuration data on the product which the host software can access. The host application can read this data using a Get Feature request for string 5. Unless otherwise configured, the default value is "No Config".

True Random Number Generator

The true random number generator exploits the fact that the RC circuit in the watchdog timer is subject to substantial drift with temperature, operating voltage, age and quantum effects. Each random bit is generated by running a calibrated 12MHz timer for 4ms, as measured by the watchdog timer. At the end of this period, the least 3 significant bits of the timer are essentially random, being sensitive to 0.002% variations in the watchdog timer period.

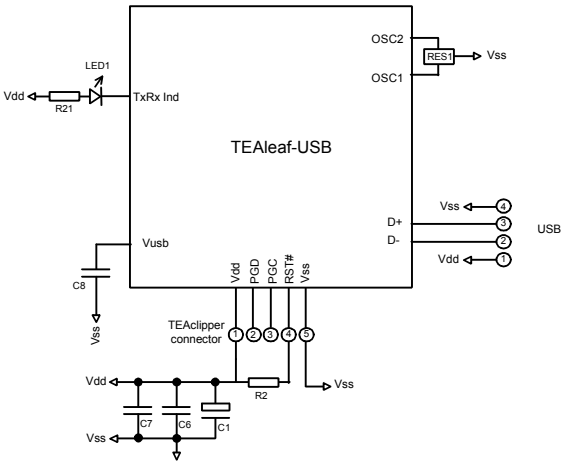
192 such random bits are generated. These are then fed into a high-avalanche polynomial ring to derive a 192-bit random number seed. The entire process takes 250ms during power-up.

Application Circuits

The following circuits are typical implementations of the TEAleaf-USB. Suggested component values are shown in table 3.

Table 3. Suggested component values	
Label	***Component***
R1, R2	*22k resistor*
R6	*1k resistor*
R21	*470R resistor*
LED1x	*Light emitting diode*
C7	*1uF capacitor*
C4, C6, C7	*100nF capacitor*
C8	*470nF capacitor*
RES1	*12MHz resonator 0.25% tolerance*
T1	*P-channel Mosfet, e.g. NDS352P*

Figure 2



Device Evaluation, Hardware Security Keys

Figure 2 is the suggested circuit when initially evaluating USB-232, and for implementing basic hardware security keys.

The pushbutton circuits allow you to apply logic 1 or logic 0 to input pins. The LED circuits allow you to view the state of output pins. As shown they are active low. For commercial applications these circuits need only be provided if required.

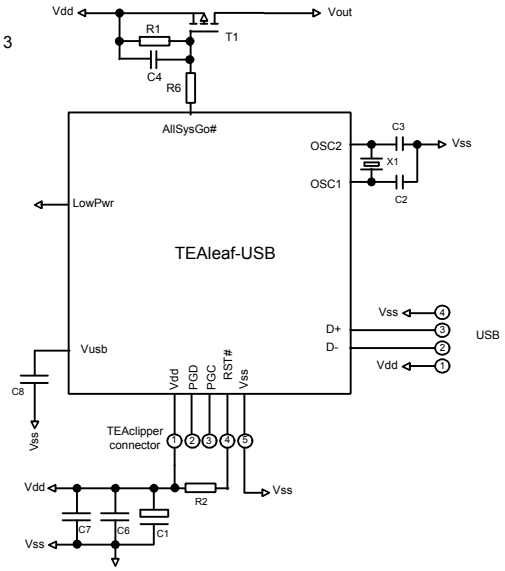
Oscillator X1/C2/C3 may be replaced by a low-cost resonator, provided its frequency tolerance is greater than 1.5%. C1 and C6 should be placed close to the USB connector. C7 should be placed near the Vss and Vdd pins of the TEAleaf-USB and is required only if it would be some distance from C6. C8 is a filter capacitor for an internal regulator and is required.

The TEAclipper connector is for in-circuit programming of devices where the firmware has been purchased from HexWax. It is recommended to allow firmware updates, even if the firmware is initially supplied pre-programmed.

Power Take-Off

Figure 3 is the suggested circuit when additionally using the USB as a source of power for other purposes. C4 / R6 provide a slow switch-on to prevent inrush current exceeding USB power limitations.

Figure 3



The AllSysGo# output is used to switch on the power when the host permits power to be drawn. It will stay on unless the computer goes into a sleep state. The Low Pwr output indicates how much power the device may draw from the USB bus. R1 ensures that T1 does not switch on while TEAleaf-USB is initializing.

Pay-Per-Use Circuit

The device can be used to control unlock hardware in pay-per-use applications. In this scenario, software on the PC host establishes that usage has been granted (e.g. by a web server) before entering the authentication process.

The Auth output can then be used to unlock the device. This would be relatively easy to 'hotwire' electrically, so contact us if you are interested in this feature being made more secure.

Power considerations

Initially, the device will request the full power it is configured for. If it is not granted this power level within three seconds, it performs a soft detach and requests a maximum of 100mA. This allows devices to operate in a low power mode when connected to an unpowered hub. This reduced power mode is indicated by the Low Pwr Indication pin.

When designing self powered circuits, ensure power is never be fed into the Vdd USB line. Design note AN1149 from Microchip Technology, in the development kit, discusses designs for recharging batteries using USB bus power.

If the device is electromagnetically noisy, a ferrite bead is recommended on the USB Vdd supply in order to suppress any transmission of noise to the rest of the USB network.

USB Connectors

Common USB connector and cable configurations are shown in figure 4 and table 4. The shield on the connector should be left unconnected. The ID pin on the mini connector permits the distinction of A and B plugs. The micro connector pin-out is the same as the mini connector.

Figure 4 Common USB pin-outs for male connectors

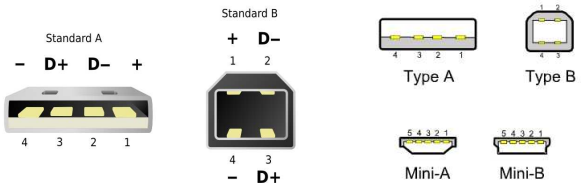


Table 4. USB Connection Key				
Pin		*Name*	*Cable color*	*Description*
Std	*Mini*			
1	1	Vcc	Red	+5V (can dip to 4.08V)
2	2	D–	White	Data –
3	3	D+	Green	Data +
–	4	ID	–	Type A: Connect to ground Type B: Not connected
4	5	Gnd	Black	Signal ground

For ultra-low cost products, it is possible to form a USB Type-A plug direct from a circuit board as shown in figure 5. This connector is only suitable for a number of insertions (~50 before cleaning is required). It is unshielded and recommended only for 'dongle' type products with no cables attached.

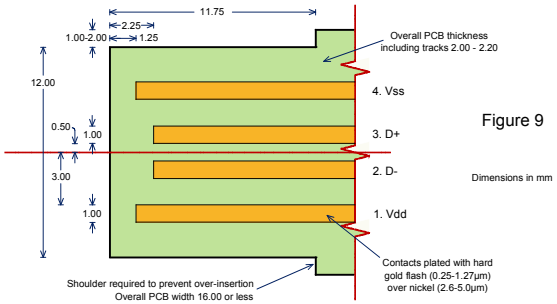


Figure 9

For further dimensional information, refer to figure 6-7 of the USB 2.0 Specification, in the development kit.

Host-Side Interfacing

TEAleaf-USB uses the Human Interface Device (HID) USB interface. It has the advantages that no device drivers are required, and that a host application can easily locate the TEAleaf.

All exchanges of data ('reports') between the host and the TEAleaf-USB are 8 bytes in length, regardless of how many bytes of meaningful data are actually transferred. In HID terms, all transfers are 10ms interrupt reports of 8 bytes, to and from output ID 0 on EP1.

The host software has two perform two tasks. First it has to locate the device. Then it has to communicate with it. To locate the device, enumerate all devices with Vendor ID 0x0B40 and Product ID 0x011E (28-pin devices) or 0x011F (20-pin devices). Then use a Get Feature request for the string 4, the Product GUID. If this matches the product GUID you configured for the device, you have located it.

Once you have located the device, you need to open a file to communicate with it. You can then send data and receive data as 8-byte reports.

Sample source code for Windows and a Windows dynamic link library (DLL) are provided in the development kit. For a detailed description, please refer to the comments embedded in the source code and the Visual Basic example in the Excel spreadsheet. Sample source code for Mac OS and Linux is in preparation.

Commands

The first byte ('identifier') of the 8-byte report in either direction identifies the remaining contents ('payload'). If the command does not require all 8 bytes, then the contents of the rest are ignored.

With the exception of the Authenticate and Random commands, if a response to the command is required, the response will have the same identifier as the command to which it is responding. The Interrupt response has no command associated with it and it may be received by the host at any time.

Note: Accidentally sending a command in the range 0x80-0x8F can modify settings that may permanently disable the device. During product development, it is recommended that you work with a device that has been write locked using HIDconfig.exe. Devices intended for production should always be write locked.

Authenticate

The identifier AUTH (0xA7) is used to initiate the authentication process. The TEAleaf-USB will reply with a four-byte random value in the first 4 bytes. To this the host should append its own four-byte random value in the second 4 bytes and then encrypt using the xTEA algorithm given below. The resulting 8-byte value should be sent to the TEAleaf. The TEAleaf will decrypt the data to determine the host's random value and to verify the random value it sent.

If the value is not correct, TEAleaf will respond with eight zero bytes. If it is correct, it replaces its random number with another random value, encrypts and sends the result to the host. The host decrypts the result to verify the random value it sent to the TEAleaf-USB. If the random value is correct, authentication is complete.

Example:

(Key is the factory default FFEEDDCCBBAA99887766554433221100)

A7 00 00 00 00 00 00 00	Auth
13 16 3A 03 00 00 00 00	Random # from TEAleaf-USB
13 16 3A 03 9B 67 2B 99	Host adds random #
44 92 D0 06 8C F5 90 F2	Host encrypts, sends to TEAleaf-USB
13 16 3A 03 9B 67 2B 99	TEAleaf-USB decrypts, verifies random#
20 2D 6A 18 9B 67 2B 99	TEAleaf-USB adds new random #
45 60 6C 84 BF 86 EE C2	TEAleaf-USB encrypts, sends to host
20 2D 6A 18 9B 67 2B 99	Host decrypts and verifies random #

Random

The identifier RANDOM (0xA8) is used to obtain a 4-byte random number from TEAleaf.

Example:

A8	Auth
2B 73 CE 89 00 00 00 00	Random # response from TEAleaf

Get Pin

The identifier GETPIN (0x90) retrieves the digital value of a VIO pin. The command payload has one byte, which indicates the pin, as shown in table 3. The response payload has two bytes, as shown in table 3.

Example:

90 18	Command – Get VIO8 pin
90 18 01	Response – Pin is active

Table 3. Get Pin Payload Structure

Pin	*Payload byte 1*	*Payload byte 2**
VIO1	*0x11*	00 = Low, 01 = High
VIO7	*0x17*	
VIO8	*0x18*	
VIO9	*0x19*	
VIO10	*0x1A*	

*Response only

Get Analog

The identifier GETANALOG (0x96) retrieves the voltage of the analog pin. The command has not payload. The response payload has two bytes, representing a number from 0x0000 to 0x03FF, which indicates the voltage relative to Vdd.

Example:

96	Command – Get Analog
96 02 36	Response – V = Vdd * (0x236/0x3FF)

Setup Pin

The identifier SETUPPIN (0x97) configures a VIO pin. The command payload has two bytes, which indicate the pin and the desired configuration, as shown in table 4.

Example:

97 18 01	Command – Set VIO8 pin high

*Table 4. Get Pin / Setup Pin Payload Structure**		
Pin	***Payload byte 1***	***Payload byte 2***
VIO1	*0x11*	0x00 = Output low 0x01 = Output high 0x02 = Digital or analog input 0x05 = Low-to-high interrupt 0x06 = High-to-low interrupt
VIO7	*0x17*	
VIO8	*0x18*	
VIO9	*0x19*	
VIO10	*0x1A*	

Interrupt

The Interrupt response is an unprompted message from the device that an interrupt input transitioned from the inactive to the active state.

It consists of the identifier INTERRUPT (0x95) and one payload byte, which is 0x09 if the interrupt occurred on pin VIO9, or 0x0A if the interrupt occurred on pin VIO10. It will not be sent while an Authenticate procedure is in progress.

Get Firmware ID

The identifier GETFWID (0x94) retrieves a zero-terminated ASCII text string identifying the firmware and its version number. It will probably need to do so over several response packets.

Example:

94	Command – Get Firmware ID
94 54 45 41 6C 65 61 66	“TEAleaf”
94 2D 55 53 42 20 30 31	“-USB 01”
94 2E 30 30 20 28 32 34	“.00 (24”
94 35 30 29 00 91 D5 7E	“50)”

xTEA Algorithm

The xTEA algorithm is a robust Feistel network proposed by Needham & Wheeler. For details refer to *RM Needham and DJ Wheeler, TEA extensions, Technical report, Computer Laboratory, University of Cambridge, October 1997.*

The host-side algorithm is presented below as C code:

```
// unsigned long integers are 32-bit
// unsigned char integers are 8-bit
// Arg pVal is a 2-long array containing your
//   randomly generated challenge
//   pVal[0] is challenge bits C31-C0
//   pVal[1] is challenge bits C63-C0
// Arg pKey is a 4-long array containing key
//   pKey[0] is the least significant 32 bits
//   pKey[1] is the next least significant 32 bits
//   pKey[2] is the next most significant 32 bits
//   pKey[3] is the most significant 32 bits
void Encr(unsigned long *pVal, unsigned long * pKey)
{
  unsigned long sum = 0;
  unsigned long delta = 0x9E3779B9;
  unsigned char i;

  for (i=0; i<32; i++)
  {
    unsigned sa = sum & 0x03;
    unsigned long Key2;
    Key2 = pKey[sa];
    pVal[0] += (( (pVal[1] << 4) ^ (pVal[1] >> 5)) +
                  pVal[1] ) ^ (sum + Key2);
    sum += delta;
    sa =  (sum>>11) & 0x03;
    Key2 = pKey[sa];
    pVal[1] += (( (pVal[0] << 4) ^ (pVal[0] >> 5)) +
                  pVal[0] ) ^ (sum + Key2);
  }
}

// On exit the TEAleaf's response must match pVal
//   pVal[0] is response bits R31-R0
//   pVal[1] is response bits R47-R32
//   ( ignoring 16 highest bits of pVal[1] )
```

The files to *TEAleafHost.c* and *TEAleafHost.hex* in the development kit contain a complete code example for host applications.

The application *HIDconfig.exe* in the development kit pack can generate data for verifying implementations of the algorithm.

Security

The xTEA algorithm has a very high avalanche effect and is extremely robust against plaintext and related-key differential attacks. Data is hashed with random number values generated by both sides. The non-reversibility of the random hashing step renders the key uncrackable as far as is known.

Care should be taken that the host-side algorithm executable code does not expose the key. It is strongly recommended that you make unique modifications to the example source code *TealeafHost.c*, for example by obfuscating the key and inserting dummy calculations.

Customization

The product can be customized in one of three ways:

1. Using the *HIDconfig.exe* application in the development kit. This application makes it very easy to copy the configuration from an existing product to a new product and is suitable for in-factory use. (It cannot be used if you have changed the Vendor ID and / Product ID.)

   The 128-bit security key can be set using *HIDconfig.exe*, but not read. To make new settings permanent, use the *Write Lock* feature.

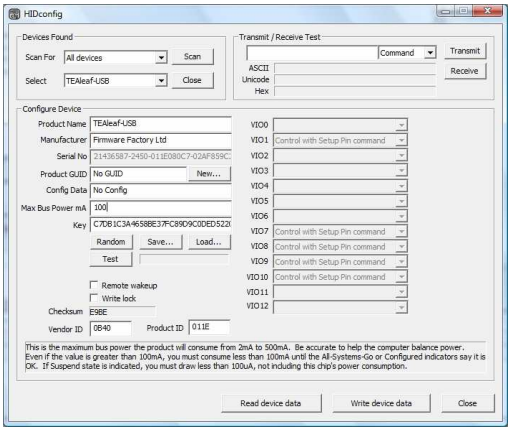


Fig 6. Hidconfig.exe application

2. By requesting the custom settings to be supplied pre-programmed when buying pre-programmed chips (5K units minimum).

3. Using customization commands. Documentation on these commands is available on request.

Design Blueprint

A design blueprint comprised of schematic diagram, bill of materials, PCB Gerbers and component placement is available. (A signed nondisclosure agreement is required.)

The design is intended for step-and-repeat on PCB panels. To facilitate in-circuit programming, PGC and PGD are connected to D+ and D- USB pins respectively. Access to *Vpp* is provided via a copper pad which can be connected to using a crocodile clip.

Individual circuits are V-scored. After breaking apart, the non-routed section should be labeled and then dipped in varnish to add robustness and electrical isolation. This enclosure-less design works well with the low profile components specified; LEDs have not been used as they would be liable to break off.

The gold plating on the USB connector should last for 50 insertions; before cleaning of the contacts will be required.

Evaluation Board

TEAleaf-USB may be evaluated with the Firmware Factory USB Products Eval Board (figure 7). The components which should be fitted are shown in table 6.

The printed circuit board integrates a USB plug which may be plugged into a USB extension cable.

When connected as described, the LEDs will light for All-Systems-Go, Tx, Rx, and Authenticated conditions. The *HIDconfig.exe* application can be used to discover and test the evaluation device.

Table 6. Evaluation Board bill of materials	
Label	***Component***
U2	*TEAleaf-USB-DIL*
D2	*Wire link*
C4	*100nF capacitor*
C7	*10uF capacitor*
C8	*470nF capacitor*
X1	*12MHz parallel cut crystal*
C2, C3	*22pF capacitor*
R13-15, R18	*470R resistor – adjust for LED brightness*
R2	*22k resistor*
LED1-3, LED5	*Light emitting diode 5mm*
W1	*Wire link from ASG# (pin 6) to LED5*
W2	*Wire link from Tx (pin 7) to LED3*
W3	*Wire link from Rx (pin 23) to LED2*
W4	*Wire link from Auth (pin 18) to LED1*

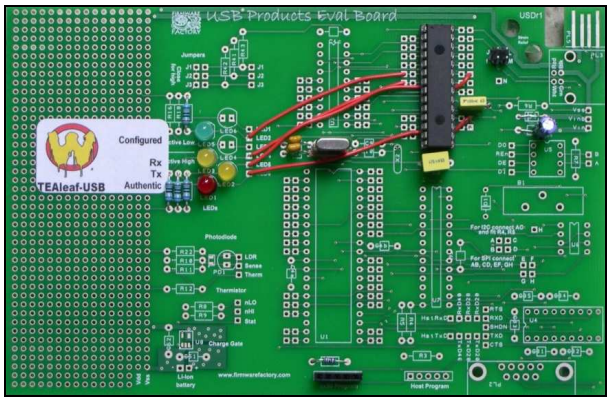


Fig 7. USB Products Eval Board

Firmware Delivery

TEAleaf-USB is available pre-programmed in 28-pin DIL and 20-pin SSOP packages. TEAleaf-USB-SS (SSOP package) may be supplied with an ID label, or it may be identified with a white mark on the package. It is also available as a fully assembled device according to the design blueprint.

Programming TEAleaf-USB

If practical, a TEAclipper programming socket should be added to the circuit board in order to facilitate in-circuit firmware updates.

TEAleaf-USB may be programmed in-circuit provided the programming signals *PGC*, *PGD* and *Vpp* are protected against contention. In particular, note that the *Vpp* line is subject to a voltage of up to 13V during programming. Nothing else should be connected to this input except a 22k pull-up resistor.

Since the programming time is fast, no programming socket is required for the TEAclipper. It may be leaned against five plate-through holes as described in figure 8. In-circuit programming connections of some form should always be provided, even if the device is supplied pre-loaded, in order to facilitate firmware upgrades.

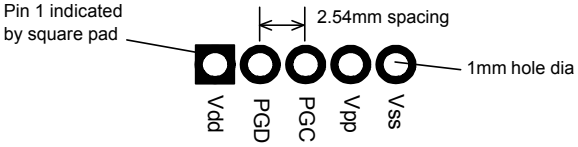


Figure 8. Recommended plate-through connector design

Development Kit

A firmware development kit is available for download from *www.hexwax.com* containing the following files:

- Base controller data sheets (© Microchip Technology Inc)
- *USB 2.0 Specification* (© HP / Intel / Lucent / Microsoft / NEC / Philips 2000)
- *HIDconfig.exe*, an application which allows you to customize TEAleaf-USB devices via the USB port. It is designed for low labor in-factory use and also serves to test the USB circuit.
- *AN1149 Designing a Li-Ion charger system…* for design examples on charging batteries from USB power (© Microchip Technology Inc 2006)
- *usb-win.c* and *usb-win.h*, sample HID code for Windows. Additionally the files *setupapi.h*, *hidsdi.h*, *hidpi.h*, *setupapi.lib* and *hid.lib* are provided, which must be included in the application.
- *FwFhid.dll* dynamic link library and Visual Basic example *FwFhidDLLExample.xls*.

In addition the following can be supplied in return for a signed nondisclosure agreement.

- *TLUrx BOM.xls* blueprint bill of materials.
- *TLUrx Gerber.zip* blueprint PCB Gerbers.
- *TLUrx BOM.pdf* schematic and component placement diagram.

Firmware Factory Ltd
2 Marshall St, 3rd Floor
London W1F 9BB, UK
sales@firmwarefactory.com
support@firmwarefactory.com