

***LaurTec***

# **LaurTec PIC Bootloader**

**Autore :** *Mauro Laurenti*

**ID:** PJ11005-IT

## INFORMATIVA

Come prescritto dall'art. 1, comma 1, della legge 21 maggio 2004 n.128, l'autore avvisa di aver assolto, per la seguente opera dell'ingegno, a tutti gli obblighi della legge 22 Aprile del 1941 n. 633, sulla tutela del diritto d'autore.

Tutti i diritti di questa opera sono riservati. Ogni riproduzione ed ogni altra forma di diffusione al pubblico dell'opera, o parte di essa, senza un'autorizzazione scritta dell'autore, rappresenta una violazione della legge che tutela il diritto d'autore, in particolare non ne è consentito un utilizzo per trarne profitto.

La mancata osservanza della legge 22 Aprile del 1941 n. 633 è perseguibile con la reclusione o sanzione pecuniaria, come descritto al Titolo III, Capo III, Sezione II.

A norma dell'art. 70 è comunque consentito, per scopi di critica o discussione, il riassunto e la citazione, accompagnati dalla menzione del titolo dell'opera e dal nome dell'autore.

## AVVERTENZE

I progetti presentati non hanno la marcatura CE, quindi non possono essere utilizzati per scopi commerciali nella Comunità Economica Europea.

Chiunque decida di far uso delle nozioni riportate nella seguente opera o decida di realizzare i circuiti proposti, è tenuto pertanto a prestare la massima attenzione in osservanza alle normative in vigore sulla sicurezza.

L'autore declina ogni responsabilità per eventuali danni causati a persone, animali o cose derivante dall'utilizzo diretto o indiretto del materiale, dei dispositivi o del software presentati nella seguente opera.

Si fa inoltre presente che quanto riportato viene fornito così com'è, a solo scopo didattico e formativo, senza garanzia alcuna della sua correttezza.

L'autore ringrazia anticipatamente per la segnalazione di ogni errore.

Tutti i marchi citati in quest'opera sono dei rispettivi proprietari.

**Indice**

<b>Introduzione</b> .....	4
<b>Il progetto LaurTec PIC Bootloader</b> .....	5
I Bootloader.....	6
I Progetti ed Esempi.....	7
<b>Aggiornare un sistema Embedded</b> .....	8
Passo 1.....	8
Passo 2.....	8
Passo 3.....	9
<b>Appendice</b> .....	13
PID e VID.....	13
<b>Bibliografia</b> .....	15
<b>History</b> .....	16

## Introduzione

I sistemi embedded moderni, sebbene permettano di svolgere compiti più o meno complessi e possano sembrare soluzioni complete ed affidabili, spesso richiedono di qualche ritocco nel Software al fine di migliorare le funzioni o correggere eventuali bug.

Un sistema che possa quindi essere aggiornato in maniera pratica è certamente un'ottima caratteristica e alcune volte un prerequisito.

Al fine di supportare un aggiornamento pratico e veloce anche i sistemi presentati sul sito LaurTec supportano spesso l'interfaccia USB per mezzo della quale si ha un canale di comunicazione con un PC Host. Il protocollo USB rappresenta un'opzione ideale per aggiornare un sistema embedded senza richiedere che questo debba essere aperto.

Molti sistemi di sviluppo hanno spesso un connettore di programmazione che permette ad un programmatore di essere collegato direttamente alla scheda, ma quando il progetto risulta completato è spesso contenuto in una scatola di montaggio per cui il connettore di programmazione non è più accessibile. Come detto l'interfaccia USB permette di risolvere questo problema ma al tempo stesso, quando non usata per la programmazione, l'interfaccia può essere utilizzata per una comunicazione con il PC per il trasferimento di dati di altra natura.

Sebbene l'interfaccia USB rappresenti una possibile soluzione per aggiornare un sistema embedded, sono necessari altri due componenti:

- Un Bootloader.
- Un'applicazione dal lato USB Host.

Il Bootloader rappresenta un'applicazione da caricare nel microcontrollore mentre l'applicazione dal lato USB Host rappresenta un'applicazione, programma, che dal lato PC permette di riconoscere il sistema embedded nel quale è presente il Bootloader, per mezzo del quale caricare un nuovo programma o aggiornamento. Il riconoscimento di un dispositivo USB avviene per mezzo di una combinazione di numeri, noti come PID (Product ID) e VID (Vendor ID). Questa combinazione è unica in ogni dispositivo o classe di dispositivi forniti da un produttore di sistemi Hardware. Come descritto in appendice, i sistemi Hardware LaurTec sono caratterizzati da diverse combinazioni PID e VID a seconda del tipo di applicazione e classe USB utilizzata. In particolare il Bootloader USB ha una combinazione PID VID unica, in maniera da poter identificare quando il sistema embedded è in modalità Bootloader. Lo stesso sistema embedded, non in modalità Bootloader, si presenta ad un USB Host con un differente PID e VID.

Quale soluzione completa al fine di avere un unico punto di riferimento per i sistemi embedded LaurTec è stato realizzato il progetto LaurTec PIC Bootloader che rappresenta una raccolta di diversi progetti e file che permettono di aggiornare i sistemi embedded LaurTec per mezzo di una sola applicazione.



### Nota

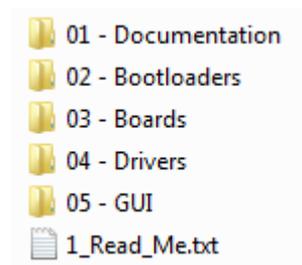
I progetti presentati sotto il nome LaurTec PIC Bootloader, rappresentano una derivazione del lavoro Microchip, per cui possono essere utilizzati solo in sistemi basati su microcontrollori Microchip. Per maggiori dettagli far riferimento all'applicazione `USB_Bootloader_2.13` menù Help → About.

## Il progetto LaurTec PIC Bootloader

Il progetto LaurTec PIC Bootloader è contenuto nella directory:

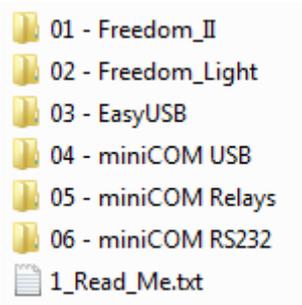
| LaurTec\_PIC\_Bootloader\_x.x.x

Le x.x.x della parte finale ne identificano la versione. La versione cambia al variare del contenuto delle directory che, come vedremo a breve, oltre all'applicazione per aggiornare i sistemi embedded LaurTec, contiene anche vari file d'impostazione ed esempi per varie schede di sviluppo. La directory principale è strutturata nel seguente modo:



In particolare l'applicazione LaurTec PIC Bootloader (USB\_Bootloader\_2.13) che rappresenta l'interfaccia grafica per mezzo della quale è possibile aggiornare un sistema embedded con interfaccia USB è contenuta nella directory GUI.

La directory Boards contiene al suo interno diverse directory con il relativo nome delle schede supportate nella versione corrente del progetto. Nella versione corrente si hanno le seguenti cartelle:



La cartella Drivers contiene i drivers richiesti nei vari esempi. In particolare sono presenti i file .inf necessari per gli esempi basati su classe CDC.

I file di esempio sono precompilati, ovvero in formato Intel HEX, con la caratteristica di avere un code offset pari a 0x1000, ovvero possono essere caricati direttamente nel sistema embedded nel quale è presente un Bootloader presente nella relativa cartella Bootloaders. La cartella Bootloaders contiene al suo interno vari Bootloader a seconda del microcontrollore utilizzato.



### Nota

Un'applicazione per microcontrollori che deve essere programmata all'interno di un sistema con Bootloader non deve essere programmata a partire dall'indirizzo 0x0000 ma dall'indirizzo 0x1000.

## I Bootloader

Come visto, nella directory Bootloaders sono contenuti diversi Bootloader. Il Bootloader altro non è che un'applicazione che deve essere caricata nel PIC prima dell'applicazione finale. Per programmare il Bootloader è richiesto un programmatore, mentre per l'applicazione finale è possibile effettuare la programmazione direttamente per mezzo del connettore USB e l'applicazione LaurTec PIC Bootloader. Il nome del Bootloader segue il seguente formato:

X\_Y\_Bootloader\_Type\_Z

dove:

X : Nome del PIC nel quale è possibile caricare il rispettivo Bootloader.

Y : Classe del Bootloader (HID, CDC)

Z : Tipo del Bootloader

In particolare per un modello di PIC possono essere presenti diversi Bootloader\_Type. Il tipo del Bootloader identifica la sequenza di avvio del Bootloader, ovvero la sequenza per permettere al microcontrollore di passare dalla normale esecuzione del programma alla modalità Bootloader. Un esempio di nome è : PIC18F14K50\_HID\_Bootloader\_Type\_A

Attualmente sono presenti i seguenti Bootloader :

### **PIC18F14K50\_HID\_Bootloader\_Type\_A**

Per avviare la modalità Bootloader è necessario premere il tasto RESET, staccare il cavo USB e riattaccarlo, tenendo premuto il tasto RESET. Se il cavo USB non è ancora collegato basta premere il tasto RESET prima di collegare il cavo USB. Una volta collegato il cavo è possibile rilasciare il tasto Reset.

### **PIC18F4550\_HID\_Bootloader\_Type\_A**

Per avviare la modalità Bootloader è necessario premere il tasto collegato al pin RB4 (normalmente chiamato S1 o BT1). Premere successivamente il tasto RESET tenendo premuto il tasto sul pin RB4 e rilasciare il RESET mantenendo ancora premuto il tasto sul pin RB4. Successivamente è possibile rilasciare il tasto sul pin RB4.

La modalità Bootloader è riconosciuta dall'applicazione LaurTec PIC Bootloader e viene segnalata sulla finestra principale (si veda il paragrafo successivo).

La modalità di attivazione dei vari Bootloader è descritta nel file 1\_Read\_Me contenuto nella directory Bootloaders. Si ricorda che un Bootloader non attivo non influenza in alcun modo l'applicazione principale ovvero non richiede risorse dal sistema.



### **Nota**

I Bootloader sono unici per ogni PIC, per cui caricare un Bootloader in un PIC errato può creare comportamenti anomali al sistema.

## I Progetti ed Esempi

Il progetto LaurTec PIC Bootloader, oltre a fornire l'interfaccia grafica per caricare e programmare un programma in un sistema embedded, fornisce anche numerosi esempi precompilati che possono essere caricati direttamente nelle varie schede di sviluppo. Ogni scheda di sviluppo e relativi esempi sono raggruppati in una directory il cui nome riflette il nome della scheda stessa.

La descrizione degli esempi è fornita nelle schede tecniche dei relativi sistemi di sviluppo. I codici sorgenti di esempio, per i quali il progetto LaurTec PIC Bootloader fornisce solo il codice HEX precompilato, possono essere scaricati direttamente alla pagina web associata alle relative schede di sviluppo.

Il nome dei file associati agli esempi presenti nel progetto LaurTec PIC Bootloader, segue un semplice standard al fine di poter estrarre le informazioni principali direttamente dal file, mentre i codici sorgenti hanno in generale un nome differente.

In particolare gli esempi sono numerati. L'esempio 00 rappresenta l'esempio di test che può essere caricato sulla scheda per poterne testare le funzioni principali ovvero Test Suite.

Per la scheda miniCOM USB si ha per esempio:

- 00-Type-A-PIC18F14K50-miniCOM\_USB\_Test\_Suite\_v-1.0

Dopo il numero che identifica l'esempio, è specificato il tipo di Bootloader necessario per poter caricare l'esempio. In particolare quello che interessa in questo caso è la dimensione del Bootloader e le Configuration Word. Bootloader differenti potrebbero avere dimensioni differenti e far uso di Configuration Word non compatibili con l'applicazione.

Il tipo del Bootloader è seguito dal nome del PIC per il quale è compilato il progetto. Successivamente è presente una breve descrizione dell'esempio e la versione della stesso.

Per la scheda miniCOM USB gli esempi sono:

- 00-Type-A-PIC18F14K50-miniCOM\_USB\_Test\_Suite\_v-1.0
- 01-Type-A-PIC18F14K50-miniCOM\_Relays\_CDC\_Class\_v-1.0
- 02-Type-A-PIC18F14K50-miniCOM\_Relays\_HID\_Class\_v-1.0
- 03-Type-A-PIC18F14K50-miniCOM\_USB\_CDC\_Class\_Input\_Output\_v-1.0

Si noti come la descrizione dell'esempio contenga anche la classe USB usata dall'applicazione stessa.

## Aggiornare un sistema Embedded

L'applicazione LaurTec PIC Bootloader permette di aggiornare il Firmware all'interno di sistemi embedded che possiedono una porta USB e abbiano un Bootloader compatibile con l'applicazione stessa. Alcuni KIT forniti da LaurTec sono forniti con microcontrollore già programmato con il Bootloader per cui il Passo 2 non è necessario e si può procedere direttamente seguendo il Passo 1 e 3. In generale però è necessario seguire i seguenti Passi:

### Passo 1

Procurarsi l'ultima versione di LaurTec PIC Bootloader. Il progetto contiene oltre alla GUI per programmare le schede anche i Bootloader ed esempi.

Questa fase va periodicamente eseguita al fine di avere sempre l'ultima versione degli esempi

### Passo 2

Installare il bootloader necessario per la propria scheda, a seconda del PIC montato. In particolare nella directory LaurTec\_PIC\_Bootloader → Bootloaders sono presenti i seguenti Bootloader:

- PIC18F14K50\_HID\_Bootloader\_Type\_A
- PIC18F4550\_HID\_Bootloader\_Type\_A

Si ricorda che dal nome del file è possibile sapere:

- Il modello del PIC per il quale è stato compilato.
- La tipologia del Bootloader HID, CDC.
- Il Tipo del Bootloader.

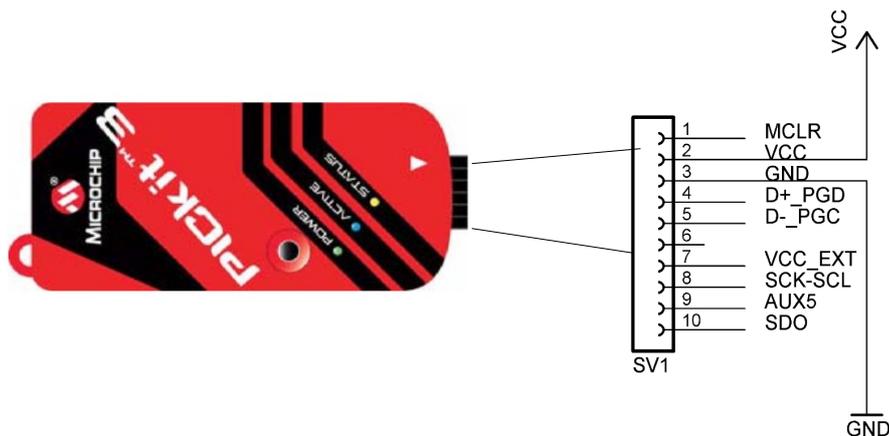
In particolare per uno stesso PIC potrebbero essere presenti diversi tipi. La differenza è la sequenza di avvio del Bootloader. La sequenza di avvio del Bootloader è descritta nel file 1\_Read\_Me.txt

Per installare il Bootloader è necessario un programmatore (si consiglia un PICKIT 2 o PICKIT 3). Questa programmazione deve essere fatta solo una volta, visto che il Bootloader permette poi di programmare la scheda direttamente per mezzo della connessione USB. Per le seguenti schede è possibile collegare il programmatore direttamente al connettore di programmazione dedicato:

- Freedom II
- Freedom Light
- EasyUSB

per la scheda miniCOM USB bisogna collegare il programmatore al connettore SV1 come mostrato in Figura 1, facendo uso di uno strip di prolunga maschio (personalmente ho

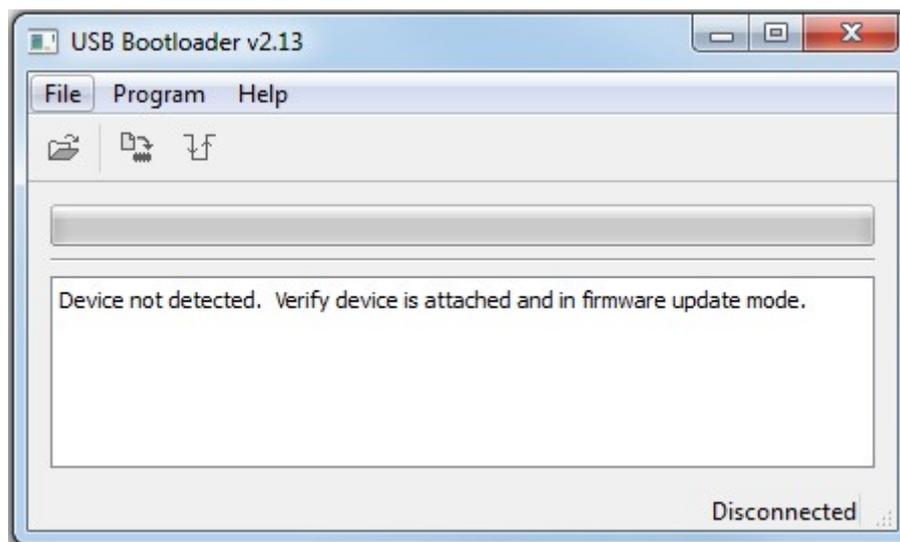
saldato due strip a 90 gradi per avere un collegamento del programmatore orizzontale).



**Figura 1:** Connessione del programmatore PICKIT 3 alla scheda miniCOM USB.

### Passo 3

Il Passo 3 presume che il microcontrollore da programmare possieda un Bootloader LaurTec. Avviare l'applicazione LaurTec PIC Bootloader come riportato in Figura 2.



**Figura 2:** Avvio dell'applicazione LaurTec PIC Bootloader.

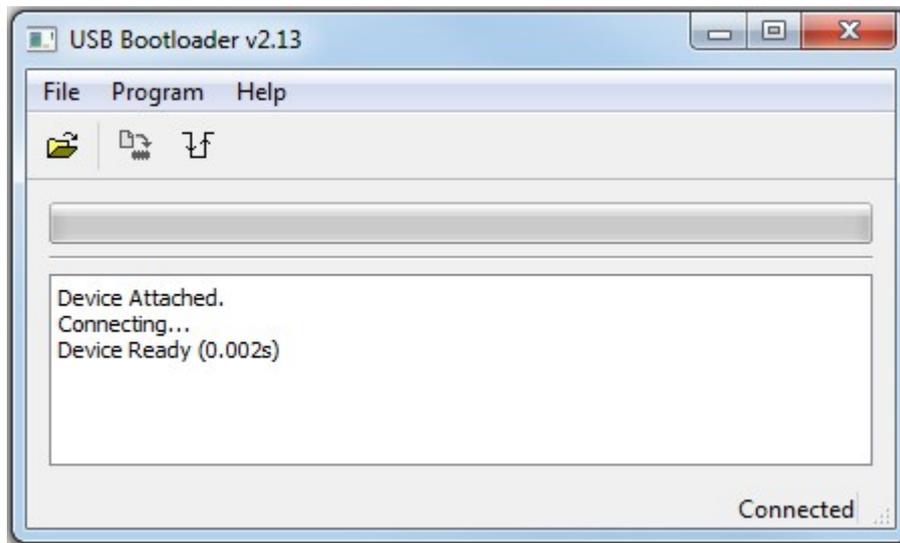
L'applicazione mostrerà che non è stata trovata nessuna scheda. A questo punto collegare la scheda che si vuole programmare al connettore USB ed avviarla in modalità Bootloader.



#### Nota

Per la scheda miniCOM USB l'avvio del bootloader avviene tenendo premuto il tasto S1 (RESET) prima di collegare il cavo USB.

Quando la scheda esegue il Bootloader, viene riconosciuta dall'Applicazione LaurTec PIC Bootloader, come mostrato in Figura 3.



**Figura 3:** Riconoscimento della scheda da parte dell'applicazione LaurTec PIC Bootloader.

Una volta riconosciuta la scheda, è possibile caricare un qualunque esempio compatibilmente con la scheda stessa. Gli esempi possono essere caricati per mezzo del menù:

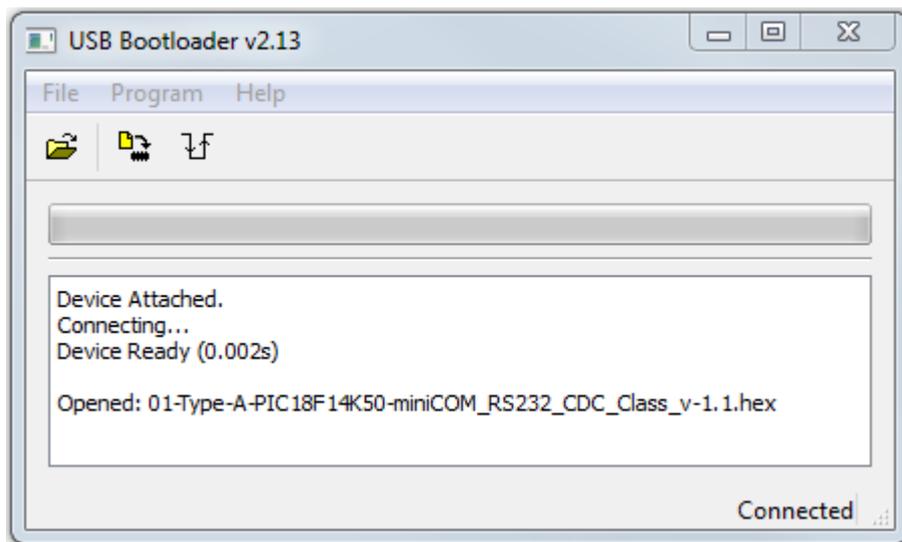
| File → Import Firmware Image



### Nota

I file di esempio forniti con il progetto LaurTec PIC Bootloader sono precompilati per le rispettive schede ed hanno un code offset pari a 0x1000. Questi esempi possono essere caricati solo per mezzo del Bootloader. Caricarli per mezzo del programmatore potrebbe causare comportamenti anomali.

Per sviluppare dei propri esempi è bene partire dai codici sorgenti forniti con le rispettive schede. Una volta selezionato il file vengono mostrati i dettagli sulla finestra principale come mostrato in Figura 4.

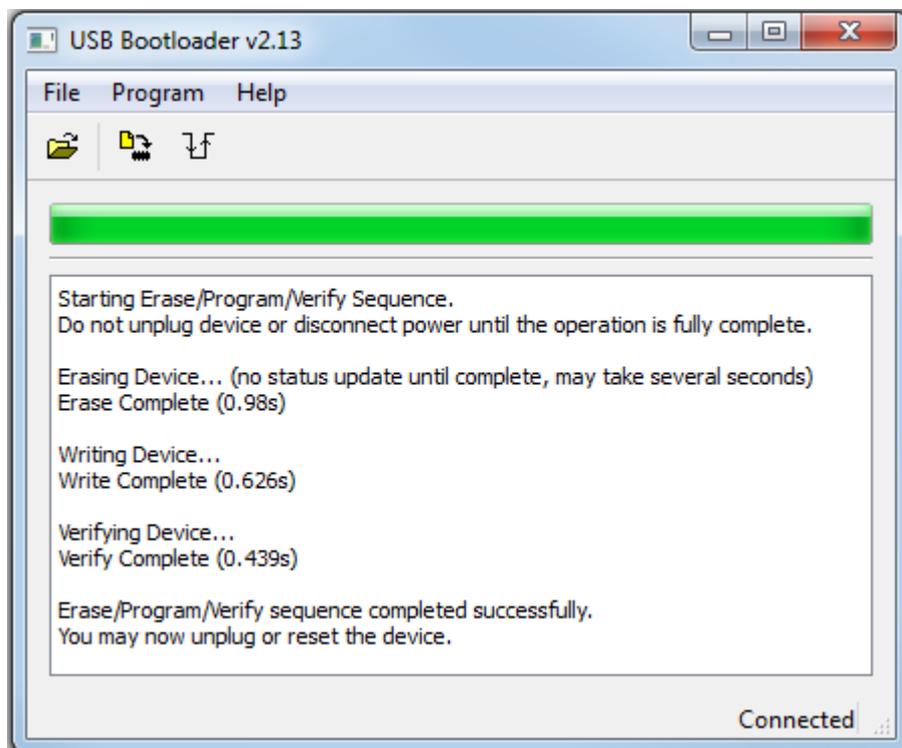


**Figura 4:** *Dettagli del file caricato.*

Per caricare il programma di esempio nel PIC basta avviare la funzione del menù:

| Program → Erase / Program / Verify Device

Ogni fase della programmazione è mostrata in dettaglio come in Figura 5.



**Figura 5:** *Dettagli della fase di programmazione.*

In alternativa si può premere il tasto:



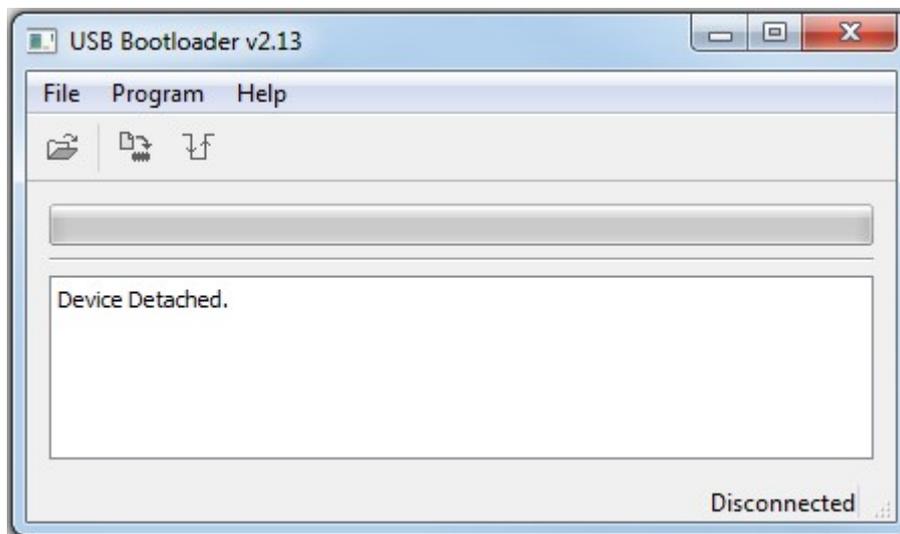
Per avviare il codice di esempio è necessario staccare e riattaccare il cavo USB o resettare la scheda per mezzo del tasto:



o per mezzo del menu:

| Program → Reset Device

Una volta resettata la scheda, il sistema non è più in fase di Bootloader per cui l'applicazione LaurTec PIC Bootloader mostrerà il messaggio di Figura 6, ovvero di scheda rimossa.



**Figura 6:** Rimozione della connessione.

## Appendice

### PID e VID

Gli esempi Software LaurTec sono forniti con una o più combinazioni PID-VID uniche, fornite dalla Microchip ed assegnate a LaurTec, per cui non possono essere utilizzate in applicazioni commerciali riadattando gli esempi proposti per i Progetti LaurTec. Un uso personale o di studio è comunque possibile senza problemi mentre un'applicazione commerciale necessita una richiesta di PID -VID al consorzio USB o alla Microchip, per mezzo della procedura di sub licensing.

A partire dall'ottobre 2013 LaurTec utilizza il seguente standard:

#### PIC18 Bootloader

- PID = 0xFC5D
- VID = 0x04D8

#### Classe CDC

- PID = 0xF5B9
- VID = 0x04D8

#### Classe HID

- PID = 0xF750
- VID = 0x04D8

L'utilizzo di PID-VID differenti permette di discriminare il Bootloader e relativi driver necessari, senza creare conflitti di driver tra classi differenti.

Qualora si fossero eseguiti esempi USB dal sito LaurTec compilati prima dell'ottobre 2013 potrebbe essere necessario disinstallare il vecchio driver. Questo discende dal fatto che il sistema operativo abbina ad ogni driver anche la combinazione PID-VID. I vecchi esempi, dal momento che ero in possesso solo di una combinazione PID-VID fanno uso sempre della stessa combinazione PID-VID indistintamente dalla classe usata CDC o HID. Il nuovo standard risolve questo possibile disagio.

## Indice Alfabetico

<b>I</b>		<b>H</b>	
1_Read_Me.txt.....	8	HEX.....	7
<b>B</b>		<b>I</b>	
Boards.....	5	Import Firmware Image.....	10
Bootloader.....	4	Intel HEX.....	5
Bootloader USB.....	4	<b>M</b>	
Bootloaders.....	5	Microchip.....	4, 13
<b>C</b>		miniCOM USB.....	8
classe CDC.....	5	<b>P</b>	
Classe CDC.....	13	PIC18 Bootloader.....	13
Classe HID.....	13	PICKIT 2.....	8
classe USB.....	4	PICKIT 3.....	8
code offset.....	5, 10	PID.....	4, 13
Configuration Word.....	7	Product ID.....	4
<b>D</b>		Program.....	11
Drivers.....	5	<b>S</b>	
<b>E</b>		sequenza di avvio.....	6
EasyUSB .....	8	sub licensing.....	13
Erase / Program / Verify Device.....	11	<b>T</b>	
<b>F</b>		Test Suite.....	7
file .inf.....	5	<b>U</b>	
Freedom II.....	8	USB Host.....	4
Freedom Light.....	8	<b>V</b>	
<b>G</b>		Vendor ID.....	4
GUI.....	8	VID.....	4, 13

## Bibliografia

[1] [www.LaurTec.it](http://www.LaurTec.it): sito ufficiale della scheda FREEDOM II dove è possibile scaricare utili tutorial come ad esempio “XC8 step by step” e le librerie utilizzate negli esempi riportati in questo articolo.

## History

Data	Versione	Autore	Revisione	Descrizione Cambiamento
26.10.13	1.0	Mauro Laurenti	Mauro Laurenti	Versione Originale.
10.07.16	2.0	Mauro Laurenti	Mauro Laurenti	Aggiornato il formato del documento. Aggiornato il software e relativa spiegazione.

## Indice Alfabetico

<b>7</b>		PIC184550.....	28
7805.....	13	PJ7010.....	11
<b>8</b>		Point to Point.....	19
8100-TDGGSM.....	5	PWR.....	7
<b>A</b>		PWRKEY.....	5, 21, 32
Arduino.....	5, 8	<b>Q</b>	
ASCII.....	15, 26	quad band.....	6
<b>B</b>		<b>R</b>	
buffer.....	28	RCSTA.....	28
<b>C</b>		RS232.....	11
CMOS.....	11	RS232 Terminal.....	11
comandi AT Hayes.....	15	RX.....	10
<b>F</b>		<b>S</b>	
Freedom II.....	11	Short Message Service.....	19
Freedom II.....	14	SIM.....	11
<b>G</b>		SIM340.....	11
GND.....	10	SIM340DZ.....	6
GSM.....	10	SIM900.....	8, 11
<b>I</b>		SIMCOM.....	11
IDC.....	7	SMS.....	19
<b>L</b>		SMS/CB.....	19
LM2596-ADJ.....	13	SMS/PP.....	19
<b>M</b>		<b>T</b>	
MAX232.....	11, 14, 21	Text Mode.....	19
MOD-GSM.....	5	Time Out.....	38
modem Hayes.....	15	TTL.....	11
MPLABX.....	30	TX.....	10
<b>O</b>		<b>U</b>	
Olimex.....	5	UEXT.....	7
Overrun.....	28	Unicode.....	19
<b>P</b>		USART.....	28
PDU Mode.....	19	<b>X</b>	
PIC-GSM-PC.....	11	XC8.....	30
		XC8 Step by Step.....	4