SOFTWARE AL SICURO CON UNA PENNA USB

IN QUESTO ARTICOLO MOSTREREMO COME UTILIZZARE UNA PARTICOLARE CHIAVETTA USB, DOTATA DI MICROCONTROLLORE E FIRMWARE PERSONALIZZABILE, PER METTERE AL SICURO LE APPLICAZIONI. MAI PIÙ COPIE ILLEGALI E MANOMISSIONI DI ALCUNA SORTA

In diverse circostante si rende necessario proteggere il proprio software dalla creazione di copie. La chiave USB proposta in questo articolo, consente proprio di venire incontro a questa esigenza, evitando che vengano effettuate copie non autorizzate delle applicazioni

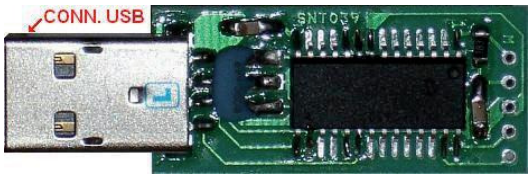

Fig. 1: La chiavetta USB priva di scocca

All'interno del software da proteggere si dovrà inserire una procedura che periodicamente controllerà la presenza della chiave USB. Quest'ultima contiene anche un contatore, in questo modo si può, ad esempio, avviare il proprio software in modalità completa per un determinato numero di riavvii; superato questo, il software entrerà in modalità demo. Si tratta solo di un esempio su come utilizzare il dispositivo. Inviare e ricevere dati dalla chiavetta USB è molto semplice, in quanto, al momento dell'inserimento del dispositivo, questo sarà riconosciuto come una porta seriale (virtuale) RS232. All'interno del proprio software occorrerà semplicemente aprire una connessione seriale e inviare e ricevere i dati secondo un protocollo abbastanza semplice, senza doversi districare tra API e librerie di terze parti, dedite al controllo delle porte USB. In definitiva, quello che viene chiesto al programmatore, è l'utilizzo del componente *Serial Port*, contenuto all'interno dell'ambiente Visual Studio o in altri linguaggi di sviluppo.
Trascinando il controllo *SerialPort* all'interno del proprio form, si avrà immediatamente disponibile l'istanza di una classe *SerialPort* pronta per essere utilizzata adoperando i seguenti due metodi:
SerialPort1.Write e *SerialPort1.Read*
Attraverso queste due funzioni è possibile inviare ricevere dati da e verso la chiavetta USB.
Prima però di poter utilizzare le funzioni *read* e *write* sarà necessario specificare il numero della porta RS232 virtuale attraverso la funzione: *SerialPort1.Portname = "COM5"*. Per ottenere la lista di tutte le porte seriali presenti sul computer si può utilizzare il comando *SerialPort.GetPortNames*. Quest'ultimo restituisce un array di stringhe contenente tutte le porte disponibili. Nel seguente esempio di codice, scritto in Visual Basic .NET, vediamo come aprire la porta seriale:

```
Dim s As String = "COM1"
For Each s In SerialPort.GetPortNames()
PortaSerialeScelta = s
Next s
SerialPort1.PortName = s
SerialPort1.BaudRate = 115200
SerialPort1.Open() ' Apro la porta seriale scelta
SerialPort1.Write("w") ' invio il comando "w" mi
                       aspetto come risposta il numero di riavvii
System.Threading.Thread.Sleep(100)
Dim DatoRicevuto As String
DatoRicevuto = SerialPort1.Read()  ' La chiavetta
                                   risponde con il numero di riavvii
```

In allegato al Cd-Rom trovate un'applicazione completa che visualizza un form con una label verde "*Chiave USB presente*", oppure una label rossa "*Chiave USB assente*"
Nell'esempio si è creato un thread separato per la lettura dei dati dalla seriale.

INSTALLAZIONE DELLA CHIAVETTA USB

La prima volta che si collega la chiavetta USB al computer viene rilevato un nuovo hardware
Occorre quindi indicare il driver del dispositivo. In realtà si tratta di un semplice file di testo *.inf*, le cui ultime 4 righe di testo sono modificabili e permettono di personalizzare la stringa visualizzata al momento dell'installazione. Il driver è disponi-

Conoscenze richieste
Visual Basic .NET

Software
Visual Studio 2005/2008

Impegno

Tempo di realizzazione

bile nel supporto Cd-Rom che accompagna la rivista. Nel caso usiate un sistema operativo Linux non viene richiesto alcun driver.

HARDWARE E FIRMWARE DEL DISPOSITIVO

La scheda elettronica è composta da pochissimi componenti. Il microcontrollore della Microchip PIC18F2455, un connettore USB, un connettore per ICP (*In Circuit Programming*) cioè permette di programmare il microcontrollore anche dopo che è stato saldato sulla scheda. La scheda ha anche un bootloader che permette di riprogrammare il PIC senza la necessità di un programmatore. L'alimentazione (5 Volt stabilizzati) viene prelevata direttamente dal bus USB che permette di erogare fino a 500mA, più che sufficienti per i pochi componenti presenti sulla scheda.
Il microcontrollore adottato, integra u driver USB. La casa produttrice, la Microchip, inoltre, mette a disposizione numerose applicazioni che facilitano lo sviluppo del firmware della scheda. L'upload di un firmware personalizzato non richiede nessun programmatore di sorta, infatti, la scheda è dotata di un apposito bootloader. L'applicazione che interagisce con lo stesso è la MPLAB, prelevabile gratuitamente dal sito della Microchip (www.microchip.com)

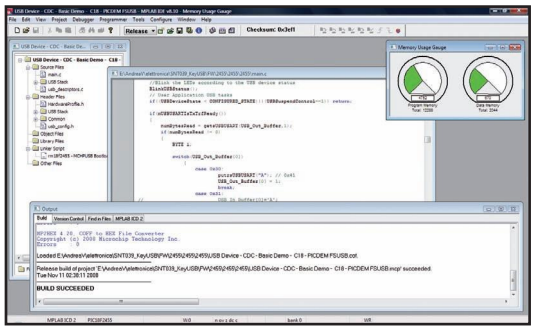

Fig. 2: L'IDE MPLAB V.8.10

Oltre all'installazione dell'MPLAB, occorre anche installare il compilatore C, anche questo disponibile gratuitamente in una versione con piccole limitazioni. Dopo aver installato l'ambiente di sviluppo (MPLAB) e il compilatore C, è possibile aprire il progetto (file *.mcw*). Il progetto contiene tutta la parte di gestione USB che non commenteremo, mentre, invece, l'unico file che possiamo modificare è il file *main.c* nel quale si fa una get dal bus USB e in base al dato ricevuto si esegue un determinato comando:

```
if(mUSBUSARTIsTxTrfReady())
{
numBytesRead = getsUSBUSART(USB_Out_Buffer,1);
if(numBytesRead != 0)
{ BYTE i;
        switch(USB_Out_Buffer[0])
        {
        case 0x30: // ricevo il dato zero dalla
                                seriale RS232 virtuale
                                putrsUSBUSART("A"); //
                rispondo inviando il carattere 'A' = 0x41
                                USB_Out_Buffer[0] =1;
                                break;
                                ........
```

Per inviare e ricevere i dati dalla chiavetta, oltre a poter usare il programmino in Visual Basic .NET (allegato al Cd-Rom) è possibile adoperare una qualunque applicazione terminale in grado di connettersi a una porta COM, per esempio HyperTerminal, già presente in Windows XP e Windows 2000. Non è presente in Windows Vista. Per quest'ultimo sistema operativo vi consigliamo il download del software *UTF-8 TeraTerm Pro*. All'interno della EEPROM del micro controller è anche possibile immagazzinare fino a 256 byte di dati, utili, ad esempio, per effettuare lo store di una password.

CONCLUSIONI

L'articolo proposto vuole essere un esempio come si possa semplicemente interfacciare il computer con dispositivi esterni. Abbiamo avuto modo di apprezzare sia le componenti hardware del progetto, sia la parte puramente software. Si ringrazia la Microchip per tutto il materiale fornito e per aver ridotto il time to market nello sviluppo di questo tipo di dispositivi grazie a tutto il materiale gratuito disponibile sul sito internet. Nei prossimi numeri, presenteremo un altro progetto elettronico: una scheda di input/output con relè e ingressi digitali, pilotabili via ethernet e quindi anche da una pagina Web.

Andrea Santagati

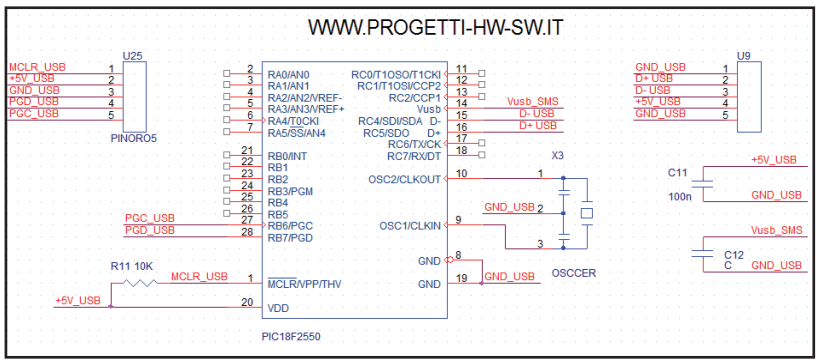

Fig. 3: Schema elettronico della chiavetta "proteggi software"

NOTA

ACQUISTO DELLA CHIAVETTA

Il dispositivo mostrato in queste pagina e acquistabile sul sito web:

http://www.Progetti-HW-SW.it

Il prezzo è di euro 19,00. Oltre a personalizzare il firmware della chiavetta, è anche possibile "brandizzare" il package inserendo, ad esempio, il logo della propria azienda

L'AUTORE

Andrea Santagati è un ingegnere elettronico che svolge attività di elettronica, non sono solo per lavoro, ma anche per hobby. L'autore è a disposizione per chiarimenti e per ricevere suggerimenti all'indirizzo di posta elettronica andrea.santagati@gmail.com