

usburnGUI

Graphische Benutzeroberfläche für usburn von Sprut

3. Oktober 2012

Inhaltsverzeichnis

1 Überblick	3
1.1 Systemvoraussetzungen	3
2 Oberfläche	4
2.1 Burning	4
2.1.1 Source HEX-File	4
2.1.2 PIC - Typ	4
2.1.3 PIC - Actions	5
2.1.4 Konsolenausgabe	5
2.2 Settings	6
2.2.1 Path to usburn	6
2.2.2 Sync GUI-DB with usburn	6
2.2.3 Calibrate Brenner8	6
2.2.4 Update Brenner8 Firmware	7
2.2.5 Display HEX file after reading	7
2.2.6 Owner and group for downloaded HEX files	7
2.3 Info	7
2.4 HEX Ansicht	7
3 Sonstiges	8
3.1 3.1 Konfigurationsdateien	8
3.2 Quellcodes	9
4 Quellen und Hinweise	9

1 Überblick

Das Programm usburnGUI dient als Benutzeroberfläche für usburn von Sprut (www.sprut.de). Es soll das Brennen von PIC-Prozessoren mittels Spruts Brenner8 erleichtern.

Da es bereits eine komfortable Software für Windows gibt, ist usburnGUI ausschließlich für Linux gedacht.

1.1 Systemvoraussetzungen

Linux OS (i386):



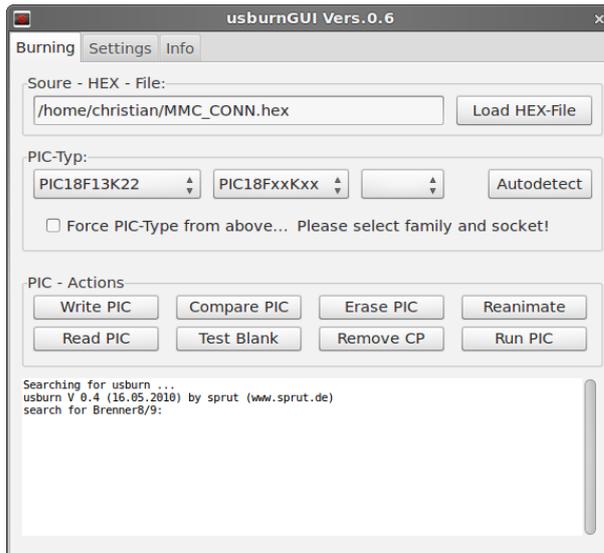
- ▷ Kernel 2.2, oder höher
- ▷ 32 MB Arbeitsspeicher
- ▷ 5 MB freier Festplattenspeicher
- ▷ GTK2 (meist bereits enthalten)
- ▷ Gnome Desktop ; (Entwicklung erfolgt in Gnome, andere Desktopumgebungen sollten aber ebenfalls funktionieren =)

Der Programm wurde komplett in PureBasic, einer Multi-Plattform Entwicklungsumgebung, geschrieben. Meine Quellcodes liegen im Unterverzeichnis „sources“ bei.

2 Oberfläche

Die Oberfläche teilt sich in vier Kategorien, die als Panel angeordnet sind. Für die eigentliche Arbeit ist aber nur der erste Rider Namens „Burning“ relevant. Vor dem ersten Gebrauch empfiehlt es sich allerdings, einen Blick auf Settings <2.2> zu werfen!

2.1 Burning



2.1.1 Source HEX-File

... erklärt sich eigentlich von selbst: Hier wird die Quelldatei zum Brennen gewählt.

2.1.2 PIC - Typ

In dieser Sektion kann der zu brennende PIC festgelegt werden. Dabei sind zuerst die Familie und Sockel zu wählen - anschließend kann per „Autodetect“, oder auch manuell ein Typ gewählt werden.



Autodetect funktioniert nur korrekt, wenn Sockel und Familie richtig gewählt wurden!



Ist keine Auswahl vorhanden, muss die Datenbank von usburnGUI aktualisiert werden. Dabei ist wie unter <2.2.2> beschrieben, vorzugehen.



„Force PIC Typ“ übergibt auch die Bezeichnung des PICs an usburn. Das ist normalerweise nicht notwendig und nur bei „Reanimate“ zu benutzen.

2.1.3 PIC - Actions

Alle Funktionen zur eigentlichen Arbeit mit den Controllern sind hier zu erreichen. Für genauere Informationen zu den jeweiligen Aktionen, empfiehlt es sich die Dokumentation von usburn zu studieren.

Write PIC Brennt das angegebene HEX-File in den PIC.

Read_PIC Liest den PIC in die zuvor per Dialog gewählte Zieldatei.

Compare PIC Vergleicht den PIC mit der unter Source-HEX- file angegebenen Quelldatei.

Test Blank Prüft, ob der PIC Daten enthält.

Erase PIC Löscht den PIC.

Remove CP Entfernt die Code-Protection und löscht den PIC.

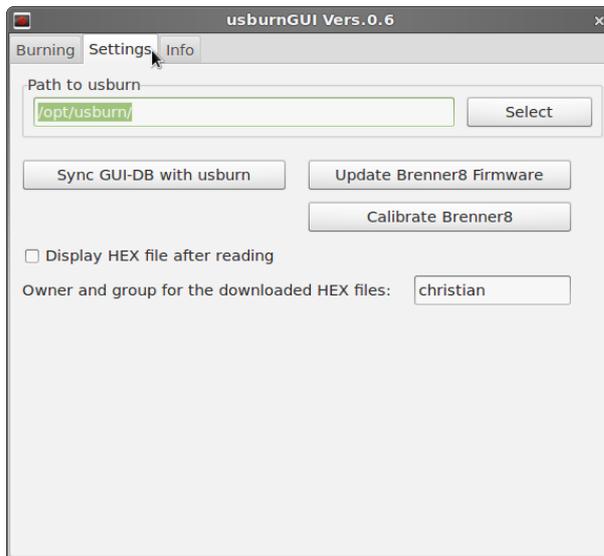
Reanimate Reanimiert den PIC, falls er nicht mehr angesprochen werden kann.

Run PIC Startet die per ICSP angeschlossene Applikation.

2.1.4 Konsolenausgabe

In diesem Bereich wird die originale Konsolenausgabe von usburn wiedergespiegelt. Das ermöglicht eine Überwachung des Prozesses und eine bessere Fehlersuche.

2.2 Settings



2.2.1 Path to usburn

In dieser Kategorie wird auf das Konsolen-Tool von Sprut verwiesen. Mit „Select“ kann der Programmpfad komfortabel gewählt werden.



Von Hand eingegebene Pfade werden erst nach einem erneuten Programmstart übernommen!

2.2.2 Sync GUI-DB with usburn

Mit dieser Funktion übernimmt usburnGUI alle usburn bekannten PICs in seine eigene Datenbank. („picDB.ini“)

2.2.3 Calibrate Brenner8

Diese Funktion öffnet ein Terminal, in welchem usburn, mit entsprechenden Parameter, gestartet wird. Als Terminalprogramm wird „xterm“ verwendet, weil es auf allen (mir bekannten) Linux-Distributionen vorhanden ist. Falls nicht, kann der Brenner (da ich davon ausgehe, dass das nicht so oft passieren wird) auch normal mit usburn kalibriert werden.

2.2.4 Update Brenner8 Firmware

Auch hier wird xterm benutzt, um usburn auszuführen. Falls xterm nicht installiert ist, kann man usburn auch einfach auf der shell starten.

2.2.5 Display HEX file after reading

Ist diese Option aktiviert, wird das vom PIC geladene HEX file nicht nur im Programm zur Ansicht geladen, sondern auch gleich angezeigt.

2.2.6 Owner and group for downloaded HEX files

Falls root-Rechte erforderlich sind um den Brenner anzusteuern, kann usburnGUI auch als root gestartet werden. Das hat allerdings zur Folge, dass auch Dateien, die von usburnGUI erstellt werden, im Besitz von „root“ sind und somit als normaler User weder gelöscht, noch verändert werden dürfen. Als Workaround für dieses Problem, kann usburnGUI den Besitzer und die Gruppe für diese Dateien setzen. In diesem Fall werden Besitzer und Gruppe einfach auf den, in diesem Feld eingetragenen Namen gesetzt. Wird dieses Feld leer gelassen, bekommt die Datei automatisch die Rechte der ausführenden Person.

2.3 Info

Hier werden allerlei, mehr oder weniger unnütze, Informationen angezeigt.

2.4 HEX Ansicht

Diese Rider beinhalten zum Brennen geladene, sowie vom PIC geladene HEX-Files. Ist ein solcher Rider aktiv, kann er einfach mittels „Esc“ geschlossen werden.

3 Sonstiges

3.1 3.1 Konfigurationsdateien

picDB.ini

Diese Datei enthält die eigene Datenbank aller PIC Typen. Sie kann wie unter <2.2.2> beschrieben, mit usburn synchronisiert werden.

Aufbau:

- ▷ Familientyp in eckigen Klammern
- ▷ bis zur folgenden Klammer: Alle Typen dieser Familie
- ▷ Typenangabe: <Typ> = <Pinzahl>

Beispiel:

```
[ PIC10 ]
PIC10F200 = 6
PIC10F202 = 6
```

```
[ PIC12 ]
PIC12F508 = 8
PIC12F510 = 8
```

```
[ PIC16 ]
PIC16F54 = 18
PIC16F57 = 28
```

usburnGUI.ini Hier werden alle Einstellungen für usburnGUI gespeichert.

Von Interesse ist hier eigentlich nur die Zeile: „Font = Monospace,8“. Sie definiert die Standardschrift in der Konsole, sowie bei der HEX-Ansicht. Schriftstandard ist Monospace, die Größe ist acht. Sollten hier Einstellungen vorgenommen werden, empfiehlt sich ausdrücklich eine Schriftart, bei der alle Buchstaben die gleiche Breite haben!

3.2 Quellcodes

Die Quellcodes zu diesem Projekt befinden sich unter „...sources/“ im Programmverzeichnis.

Mein Tool wurde mit PureBasic (www.purebasic.de), einer Basic-Plattform für Win, Linux und Mac, erstellt. Die Entwicklung erfolgt unter Ubuntu-Studio. Kompatibilität zu anderen Distributionen sollte ebenfalls vorhanden sein.

4 Quellen und Hinweise

www.sprut.de	Viele Projekte, jede Menge Infos, der Entwickler vom Brenner8/9 und usburn, sowie ein guter Einstieg in die Mikrocontrollerprogrammierung. Danke an dieser Stelle!!!
www.purebasic.de	Eine (vor allem für Einsteiger) leicht zu erlernende Programmiersprache, die einen guten Funktionsumfang, sowie eine gute Unterstützung verschiedener Systeme mit sich bringt.
http://piklab.sourceforge.net	Piklab eine Entwicklungsumgebung für PIC. Eigentlich für KDE gedacht, passt sie sich auch in Gnome ganz gut ein.

Für Bug - Reports und Hinweise bin ich dankbar: hobby.programmierer@gmx.de