Betreff: Re: [sdcc-devel] Arduino port for STM8 and contributions to SDCC
Von: Benedikt Freisen <b.freisen@gmx.net>
Datum: 19.11.18, 11:19
An: Georg Icking-Konert <georg@cream-tea.de>
Kopie (CC): Michael Mayer <michael-mayer@gmx.de>

Kurz gesagt: Ich hoffe, dass ich noch in dieser Woche das C-basierte sduino mit LLVM+SDCC ans Laufen kriegen werde. Dann muss ich im nächsten Schritt sduino nach C++ zurück portieren. Was dann folgt, ist das Ausbügeln von Ecken und Kanten, die sich in dem ganzen Prozess ergeben, was eventuell nötige Verbesserungen aller beteiligten Werkzeuge beinhalten kann.
Das ganze Ding am Ende als Board-Manager-Paket auf einen Server zu legen, ist geplant, ich kann da aber nicht versprechen, dass das noch in diesem Jahr geschehen wird.

Gruß
Benedikt

Am 19.11.18 um 10:44 schrieb Georg Icking-Konert:

> hallo Benedikt,
>
> bei diesem Namen und der Mail-Adresse gehe ich davon aus, dass Du Deutsch sprichst...!? But if you don't speak German, just drop me a note...
>
> Für mich sind Deine Erläuterungen leider nur böhmische Dörfer, da ich nur SDCC-User bin. Die Devel-Mailingliste hatte ich nur benutzt um Dich zu erreichen, sorry! Ich kenne mich zwar etwas mit HW-naher embedded Programmierung aus (konkret STM8), aber leider gar nicht mit den dafür benötigten Compilern ☹ Ich stehe immer nur staunend davor und wundere mich dass es funktioniert... 😉
>
> Kannst Du mir daher bitte nochmal kurz den Status für den Laien zusammenfassen? Konkret: siehst Du eine Chance in absehbarer Zeit einen Installer von LLVM+SDCC hinzubekommen, um den STM8 per Arduino IDE in C++ zu programmieren. Und da demnächst Weihnachten ist, vielleicht sogarper Arduino Boardmanager...?
>
> Sorry Dich zu nerven, aber falls Du das in absehbarer Zeit hinkriegst, würde ich aktuell eher an einem anderen Projekt arbeiten als jetzt Zeit in sduino zu stecken und das später in C++ zu portieren. Wäre toll wenn es klappt, da sich die Anpassung von Arduino-Libs dann auf die unterste HW-Ebene beschränken würde (wo ich mich besser auskenne) - und man die ganzen Klimmzüge in C nicht mehr bräuchte!
>
> Die noch einen schönen Tag! 🙂
>
> Gruß,
> Georg Icking-Konert (aka gicking)
>
>
> Am 17.11.18 um 13:06 schrieb Benedikt Freisen:
>
>> Hi!
>>
>> It's still work-in-progress. In more detail, the current status of my Arduino port for STM8 based on LLVM+SDCC and sduino is as follows:
>>
>> - I am still working on improvements to the SDCC compiler. As you may have noticed, the C99-style intermingling of declarations and code has made it into the 3.8.0 release. Struct assignment support has been merged shortly thereafter. Sduino can already profit from that if used with an up-to-date SDCC.
>>
>> - I've added support for my new "sduino MB" board in Arduino Mega form factor to sduino. This might actually be a good opportunity to submit a pull request for that.
>>
>> - There is a somewhat working wrapper for the LLVM+SDCC tool chain that invokes Clang, llvm-cbe and SDCC with appropriate arguments. It still needs a lot of polishing, though.
>>
>> - Right now, I am integrating support for the interrupt attribute into Clang and llvm-cbe, which is necessary for functioning interrupt service routines.

> > Future steps will include:
> >
> > - Making the C-based sduino library compilable with the wrapped LLVM+SDCC tool chain. This involves porting the platform.txt file.
> >
> > - Porting everything to C++. This should boil down to reintegrating the STM8 code from sduino into the Arduino code.
> >
> > - Support for the new STM8-based Nucleo boards with Arduino headers.
> >
> > - Further SDCC improvements. Most importantly support for structs as function arguments and return values.
> >
> > Well, for now, this is it.
> >
> > Regards
> > Benedikt
> >
> > Am 17.11.18 um 07:12 schrieb Georg Icking-Konert:
> >
> > > Hi Benedikt,
> > >
> > > I just stumbled across your below post in this mailing list (https://sourceforge.net/p/sdcc/mailman/message/36341959/). You describe a project to combine LLVM+SDCC to (amongst others) facilitate C++ for the sduino project (https://github.com/tenbaht/sduino) --> support STM8 for the Arduino IDE via SDCC
> > >
> > > As I sometimes contribute (a tiny bit) to sduino, I am highly interested in this! Because SDCC support for C++ would render all the current tricks obsolete and would limit porting Arduino libs to STM8 (and actually all supported controllers) to the lowest HW layer. Great!
> > >
> > > Could you please report about your current status and your release plan? Any chance us humble users can get this anytime soon? Thanks a lot in advance!
> > >
> > > Regards,
> > > Georg Icking-Konert
> > >
> > > > On 13 Jun 2018, Benedikt Freisen wrote:
> > > >
> > > > Good afternoon!
> > > >
> > > > My name is Benedikt Freisen, I am a masters student in computer science and first appeared on this mailing list in February this year in the context of memcpy improvements and wiki recovery.
> > > >
> > > > For a while now I've been planning to create a proper Arduino port for my sduino UNO board (1) and will now be able to do so for credit as part of my studies, which will allow me to dedicate significantly more time to the project than would otherwise be possible.
> > > >
> > > > The core of this project will be a C++-based back-port of Michael Mayer's C-based Sduino Project (2) combined with a user friendly GCC-like wrapper for the LLVM+SDCC tool chain (3), along with improvements that are necessary to make things work or that are otherwise desirable.
> > > >
> > > > This work will therefore also include some improvements to SDCC. To facilitate their inclusion in a future SDCC release (e.g. 3.9.0) I would like to share these early on.
> > > >
> > > > Some of the issues I've already looked into over the course of the past three weeks are:
> > > >
> > > > - C99 declarations after statements:
> > > >   Implemented using a minimally invasive approach that I'd like to call "implicit blocks" -- You'll shortly get a preliminary patch
> > > > - Syntactic support for K&R function definitions:
> > > >   Implemented via parser-level rewriting, i.e. with the semantics of a fully prototyped ANSI-C version -- still useful for Dhrystone
> > > > - AST-level rewriting of struct assignments to memcpy invocations:
> > > >   Works if memcpy is already known but still fails to properly synthesize a memcpy prototype if that's not the case
> > > > - Support for Universal Character Names in identifiers:
> > > >   I'll most likely postpone this one

> > > > Because some of the bullet points above where merely a way to get in touch with the code base and have relatively little to do with my needs, the list might grow over the course of the next couple of months, depending on what problems I run into.
> > > >
> > > > Regards
> > > > Benedikt
> > > >
> > > > 1. https://github.com/roybaer/sduino_UNO
> > > > 2. https://github.com/tenbaht/sduino
> > > > 3. http://www.colecovision.eu/llvm+sdcc/
> > >
> > > ____
> > > sdcc-devel mailing list
> > > sdcc-devel@lists.sourceforge.net
> > > https://lists.sourceforge.net/lists/listinfo/sdcc-devel