Subject: Porting upp Core to ESPE32/FreeRTOS
Posted by kohait00 on Sat, 24 Feb 2024 10:50:03 GMT
View Forum Message <> Reply to Message

Hi guys

I think it's time for upp to put a foot in the embedded / IoT world..

Having spent some time in ESP32 / FreeRTOS, I consider it quite possible to have upp console appocations be able to run as tasks inside the respective environment.

Why esp32 / FREERTOS: They represent both currently very popular choices that can be found in other environments too.

This means of corse porting Core to FreeRTOS.

Do we have a sort of porting guide or collection of typical construction/adaption sites?

I remember

config.h
Defs.h
Mt
LocalProcess
App.h

And of course

Core.h

The changes would very likely be

* processor adaption (endianness, types,..)
* OS adaption (task, locks, memory management,..)
* replacing tools or mapping to existing counterparts

Subject: Re: Porting upp Core to ESPE32/FreeRTOS
Posted by Xemuth on Mon, 04 Mar 2024 09:48:47 GMT
View Forum Message <> Reply to Message

Hello Kohait00,

As an esp32 idf framework user having u++ core with this framework would be kind of strange except if you are separate logic that use only Upp core from logic that interact with the chip using

esp idf framework.

Moreover the size of memory in esp32 is limited -> May upp core is too heavy to fit properly in several esp32 (Maybe) ?

Can't remember if Upp Core use RTTI but it can be a problem you have to think about (If I remember rtti on esp32 is disable by default)
Idem for exception.

I won't lie, when I started working with esp idf framework I wish I had Upp Core. Nowaday if Core was available without huge cost on esp32 binary size I would certainly use it. :d

PS: Upp core is maybe highly tied to platform it run on (i.e linux/windows) for several reason/ features OS provide to Upp Core. Porting it to esp would irremediably result in a shrinked core or a huge amount of works to adapt everythings.

Subject: Re: Porting upp Core to ESPE32/FreeRTOS
Posted by mirek on Sat, 09 Mar 2024 12:20:38 GMT
View Forum Message <> Reply to Message

U++ as designed is unlikely to work with less than 4MB of heap memory.

It would be fun to do something for embedded, it could even use some U++ concepts, but I am afraid it would have to start mostly from the scratch.
