

Re: Source code for CAN Implementations

Source: <http://coding.derkeiler.com/Archive/General/comp.arch.embedded/2007-07/msg00483.html>

- *From:* Urs Beeli <usenet@xx>
 - *Date:* Wed, 18 Jul 2007 12:10:56 +0000 (UTC)
-

On Wed, 18 Jul 2007 12:00:18 GMT dathome wrote:

"Urs Beeli" <usenet@xx> wrote in message news:slrnf9rv41.jre.usenet@xxxxxxxxxxxxxxxxxxxxxxxx

On Mon, 16 Jul 2007 20:45:57 GMT dathome wrote:

"raj" <maddukuri.rajesh@xxxxxxxx> wrote in message news:1184104563.300014.48500@xx

Dear All

I Kindly request the Source code links for
CAN Implementation as I am
very curious about insights.

Unfortunately I am not aware of any CAN implementation that are open
source
or in the public domain.

In my former job we have been using CAN on both an infineon C167/C164
and
on
a PPC5200. On the former we wrote our own CAN and CANopen library, on
the
latter we bought the Vector CAN/CANopen stack and had them adapt their
CAN
driver for our hardware.

Hi Raj,

This depends on how much detail you need! I've written my
own small
drivers
for C164 and have also used drivers from Vector. The Vector
ones are

Re: Source code for CAN Implementations

pretty
daunting to get started with and it's unlikely you'll be able to
get
hold of
them anyway unless you know someone ;-).

I doubt you would get them without dishing out some serious money!
However,
despite being daunting, they are a fine piece of software, I was very much
impressed with them (both their performance and their source code). Also,
their people are competent and helpful.

I've got to disagree somewhat with the comments regarding how good the
Vector drivers were with particular reference to the coding standards used.
They were littered with continue and break statements as well as having
bunches of code that was compiled dependant on #defines. Not at all MISRA-C
compliant despite being for vehicle platforms! Additionally the footprint
was way bigger than necessary occupying about 8-9k on a 16bit micro.

Fair enough. As we were using CANopen in a non security critical environment
MISRA-C and SIL2 were not something we cared about greatly :)

What I tried to say was that I found the code well enough written to easily
follow it, understand it, even make some modifications to it at a later
stage where we needed some proprietary extensions. Compared to some other
third party libraries I have seen in my time, that was a positive surprise.

They obviously have very good sales teams to have got all the major car
manufacturers to use it.

It might be a lack of competition. At the time we were looking for
CAN/CANopen implementations under QNX and there was only one other company
besides Vector that even had a product. It might be different for other
OSes, though.

To be fair though the support was good

Exactly my experience. Also, they were very patient with our management
being very difficult about signing the contract. Had I been in their
position I would have told my company to go stuff ourselves :)

Re: Source code for CAN Implementations

The execution time of the various tasks also seemed pretty good as well which suggests that a large amount of the code was never actually used.

Yepp. My impression was, that their code base is fairly large in order to cover all options of the CANopen standard, very flexible and highly configurable but would only include the code that you actually activated using the proper defines.

Cheers

/urs

--

"Change is inevitable, except from a vending machine."

-- Urs Beeli, <usenet@xx>

.