Drivers and API Conversion from C to C++ using Existing Software

Intent of this project is to convert existing tool generated C platform software into C++ platform without doing modifications in existing software.

It contains strategies to convert current tool generated C software drivers into C++ drivers with API.

I have taken reference from Infineon DAVE generated software and modified it according to me and then converted it into C++.

C++ gives us much better modularity and flexibility for API development which is why I opted to do this task.

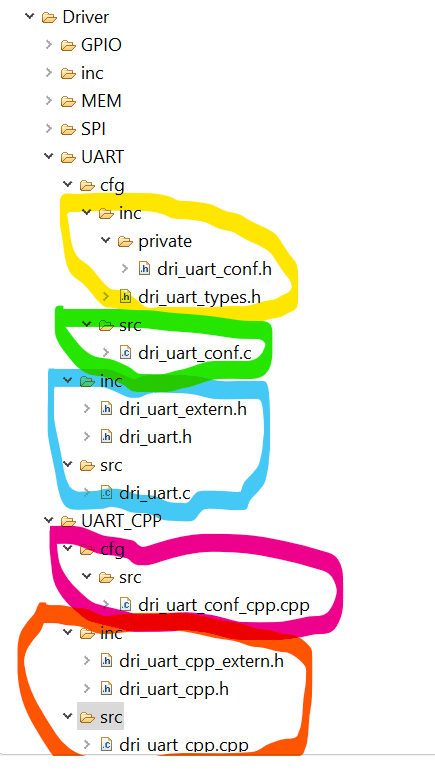
I personally think that APIs and Application should be developed in C++ because of its advantages over C, but memory space for Microcontrollers and similar platforms becomes a critical aspect to do so.

Optimization techniques needs to be considered while converting code, otherwise code size and code processing might become a major task to be dealt with!

Author: Hemant Sharma

I will consider UART serial driver as an example.

Directories and files generated are as follows:



Tool Generated sources:

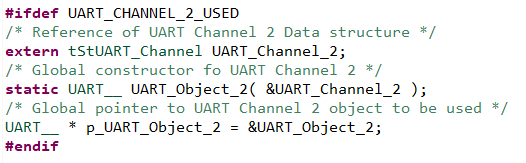
* Yellow portion contains configuration headers and data types for UART Driver and API. Data type header contains HAL headers and library reference for HW interface.
* Green portion have the data structure used for different channels of UART which contains channel parameters and properties.
* Blue portion contains the API and channel references for global usage in system for C interface.

Customised source according to Tool generated source:

* Pink portion have the reference of data structure for channels to be used in C++ interface defined in Green portion. Two more variables are used for C++ class object and its pointer to be used for global usage.
* Orange portion contains API and channel object pointers for global usage in system for C++ interface.

By this all the configuration and data types are same as of C interface, just addition of some variables and class has been introduced in C++ interface.

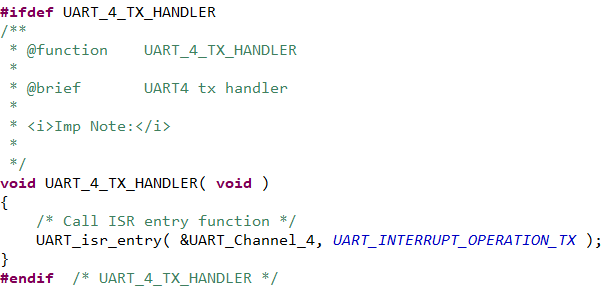
Now what have been done is, without creating any further data structure for channel properties in C++, I have taken reference of data structure defined in C configuration file and created an object of channel with its pointer.

 *dri\_uart\_conf\_cpp.cpp*

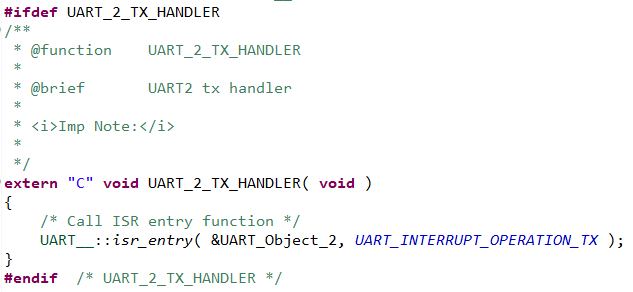
C++ interface variables: UART\_Channel\_2 is already defined in C configuration file[*dri\_uart\_conf.c*]. There is an object of that channel in which channel handle is passed as an argument and a pointer to that class object.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

ISR entries in C and C++ interface

 *dri\_uart\_conf.c*

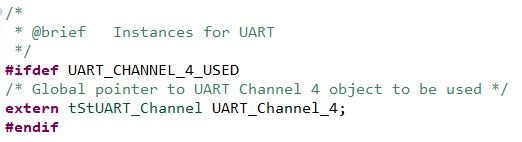
C interface

 *dri\_uart\_conf\_cpp.cpp*

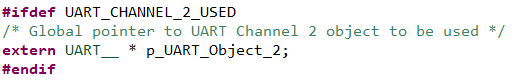
C++ interface (extern used in C++ files because of C++ compilation because of which Exception Handler Entry/Veneer[from startup.s] will not be able to call ISR handler for interrupts)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

External references of Handles and Object pointers are as follows:

 *dri\_uart\_extern.h*

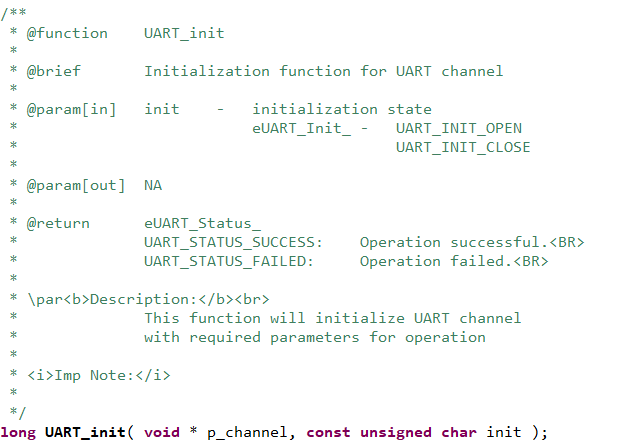
C interface Global handles

 *dri\_uart\_cpp\_extern.h*

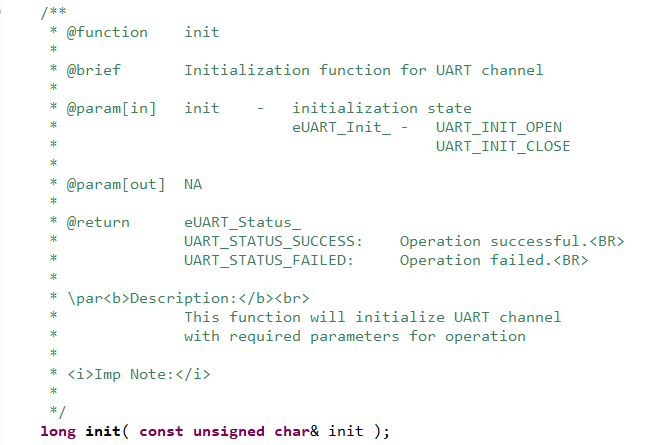
C++ interface Global object pointers

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

APIs structure for interfaces:

*dri\_uart.h*

C interface

*dri\_uart\_cpp.h*

C++ interface (part of class now)

Major difference here is, now we do not need any handle reference and everything can be done by using class object pointer or class reference.

Every function is same in definition (just need to bind everything within class) which can be compared by looking in code.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Now we will talk about cost of converting C into C++ by looking at analysis using Static Memory Analyzer.

Functions and Objects data without Optimization:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Type | Adr [Hex] | Size [B] | Region |
| \_ZN6UART\_\_4initERKh | FUNC | 0x0800E2E9 | 460 | FLASH 1 (cached) |
| UART\_init | FUNC | 0x0800EE99 | 438 | FLASH 1 (cached) |
| \_ZN6UART\_\_12rx\_isr\_entryE | FUNC | 0x0800E869 | 380 | FLASH 1 (cached) |
| UART\_rx\_isr\_entry | FUNC | 0x0800F3F9 | 368 | FLASH 1 (cached) |
| \_ZN6UART\_\_12tx\_isr\_entryE | FUNC | 0x0800E6A1 | 308 | FLASH 1 (cached) |
| UART\_tx\_isr\_entry | FUNC | 0x0800F23D | 294 | FLASH 1 (cached) |
| \_ZN6UART\_\_8transmitEPKhRK | FUNC | 0x0800E4B5 | 196 | FLASH 1 (cached) |
| UART\_transmit | FUNC | 0x0800F051 | 194 | FLASH 1 (cached) |
| UART\_rx\_isr\_direct | FUNC | 0x0800F365 | 148 | FLASH 1 (cached) |
| \_ZN6UART\_\_13rx\_isr\_direct | FUNC | 0x0800E7D5 | 148 | FLASH 1 (cached) |
| \_ZNK6UART\_\_11get\_channelE | FUNC | 0x0800E579 | 140 | FLASH 1 (cached) |
| UART\_get\_channel | FUNC | 0x0800F115 | 132 | FLASH 1 (cached) |
| \_Z25UART\_callback\_handler | FUNC | 0x08010BD1 | 124 | FLASH 1 (cached) |
| \_Z41\_\_static\_initializati | FUNC | 0x0800EAE5 | 92 | FLASH 1 (cached) |
| \_ZNK6UART\_\_19reconfigure\_ | FUNC | 0x0800E649 | 86 | FLASH 1 (cached) |
| calc\_fifo | FUNC | 0x0800F199 | 82 | FLASH 1 (cached) |
| \_Z41\_\_static\_initializati | FUNC | 0x08010391 | 76 | FLASH 1 (cached) |
| \_ZN6UART\_\_9calc\_fifoERA2\_ | FUNC | 0x0800E605 | 68 | FLASH 1 (cached) |
| UART\_callback\_handler | FUNC | 0x08010B7D | 60 | FLASH 1 (cached) |
| \_ZN6UART\_\_C2EPv | FUNC | 0x0800E9E5 | 52 | FLASH 1 (cached) |
| \_ZN6UART\_\_C1Epv | FUNC | 0x0800E9E5 | 52 | FLASH 1 (cached) |
| UART\_isr\_entry | FUNC | 0x0800F569 | 48 | FLASH 1 (cached) |
| \_ZN6UART\_\_9isr\_entryEPS\_R | FUNC | 0x0800EA3D | 40 | FLASH 1 (cached) |
| \_ZN6UART\_\_D2Ev | FUNC | 0x0800EA19 | 34 | FLASH 1 (cached) |
| \_ZN6UART\_\_D1Ev | FUNC | 0x0800EA19 | 34 | FLASH 1 (cached) |
| UART\_Channel\_3\_Configurat | OBJECT | 0x20009F70 | 32 | DSRAM 1 (system) |
| UART\_Channel\_2\_Configurat | OBJECT | 0x20009F40 | 32 | DSRAM 1 (system) |
| UART\_Channel\_3\_Control | OBJECT | 0x20009BD0 | 28 | DSRAM 1 (system) |
| UART\_Channel\_4\_Control | OBJECT | 0x20009BEC | 28 | DSRAM 1 (system) |
| UART\_Channel\_2\_Control | OBJECT | 0x20009BB4 | 28 | DSRAM 1 (system) |
| \_Znaj | FUNC | 0x08010441 | 24 | FLASH 1 (cached) |
| \_Znwj | FUNC | 0x08010769 | 24 | FLASH 1 (cached) |
| \_ZdlPv | FUNC | 0x08010219 | 20 | FLASH 1 (cached) |
| \_ZdaPv | FUNC | 0x08010459 | 20 | FLASH 1 (cached) |
| \_GLOBAL\_\_sub\_I\_p\_UART\_Obj | FUNC | 0x0800EB41 | 16 | FLASH 1 (cached) |
| \_GLOBAL\_\_sub\_D\_p\_UART\_Obj | FUNC | 0x0800EB51 | 16 | FLASH 1 (cached) |
| UART\_Channel\_4 | OBJECT | 0x20009FC4 | 12 | DSRAM 1 (system) |
| UART\_Channel\_2 | OBJECT | 0x20009F64 | 12 | DSRAM 1 (system) |
| UART\_Channel\_3 | OBJECT | 0x20009F94 | 12 | DSRAM 1 (system) |
| \_ZL13UART\_Object\_2 | OBJECT | 0x20009BAC | 4 | DSRAM 1 (system) |
| \_ZL13UART\_Object\_3 | OBJECT | 0x20009BB0 | 4 | DSRAM 1 (system) |
| UART\_Channel\_3\_Callbacks | OBJECT | 0x20009F90 | 4 | DSRAM 1 (system) |
| p\_UART\_Object\_3 | OBJECT | 0x20009F3C | 4 | DSRAM 1 (system) |
| p\_UART\_Object\_2 | OBJECT | 0x20009F38 | 4 | DSRAM 1 (system) |
| UART\_Channel\_4\_Callbacks | OBJECT | 0x20009FC0 | 4 | DSRAM 1 (system) |
| UART\_Channel\_2\_Callbacks | OBJECT | 0x20009F60 | 4 | DSRAM 1 (system) |

Functions and Objects data with Optimization:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Type | Adr [Hex] | Size [B] | Region |
| UART\_init | FUNC | 0x080091DD | 504 | FLASH 1 (cached) |
| UART\_isr\_entry | FUNC | 0x08009451 | 488 | FLASH 1 (cached) |
| \_ZN6UART\_\_4initERKh | FUNC | 0x08008CE5 | 484 | FLASH 1 (cached) |
| \_ZN6UART\_\_12rx\_isr\_entryE | FUNC | 0x08008FDD | 246 | FLASH 1 (cached) |
| \_ZN6UART\_\_12tx\_isr\_entryE | FUNC | 0x08008EC9 | 176 | FLASH 1 (cached) |
| UART\_transmit | FUNC | 0x080093D5 | 124 | FLASH 1 (cached) |
| \_ZN6UART\_\_8transmitEPKhRK | FUNC | 0x08008BC9 | 120 | FLASH 1 (cached) |
| \_ZN6UART\_\_13rx\_isr\_direct | FUNC | 0x08008F79 | 100 | FLASH 1 (cached) |
| \_ZNK6UART\_\_11get\_channelE | FUNC | 0x08008C41 | 88 | FLASH 1 (cached) |
| UART\_get\_channel | FUNC | 0x0800916D | 80 | FLASH 1 (cached) |
| \_Z25UART\_callback\_handler | FUNC | 0x0800A0F1 | 64 | FLASH 1 (cached) |
| \_GLOBAL\_\_sub\_I\_p\_UART\_Obj | FUNC | 0x08009115 | 44 | FLASH 1 (cached) |
| \_GLOBAL\_\_sub\_D\_p\_UART\_Obj | FUNC | 0x08009141 | 44 | FLASH 1 (cached) |
| \_ZN6UART\_\_9calc\_fifoERA2\_ | FUNC | 0x08008C99 | 40 | FLASH 1 (cached) |
| \_ZNK6UART\_\_19reconfigure\_ | FUNC | 0x08008CC1 | 36 | FLASH 1 (cached) |
| UART\_Channel\_3\_Configurat | OBJECT | 0x20009F28 | 32 | DSRAM 1 (system) |
| UART\_Channel\_4\_Configurat | OBJECT | 0x20009F08 | 32 | DSRAM 1 (system) |
| UART\_Channel\_2\_Configurat | OBJECT | 0x20009F78 | 32 | DSRAM 1 (system) |
| UART\_Channel\_3\_Control | OBJECT | 0x20009BB0 | 28 | DSRAM 1 (system) |
| UART\_Channel\_4\_Control | OBJECT | 0x20009BCC | 28 | DSRAM 1 (system) |
| UART\_Channel\_2\_Control | OBJECT | 0x20009BE8 | 28 | DSRAM 1 (system) |
| UART\_callback\_handler | FUNC | 0x0800A0D5 | 24 | FLASH 1 (cached) |
| UART\_Channel\_4 | OBJECT | 0x20009F60 | 12 | DSRAM 1 (system) |
| UART\_Channel\_2 | OBJECT | 0x20009F48 | 12 | DSRAM 1 (system) |
| UART\_Channel\_3 | OBJECT | 0x20009F54 | 12 | DSRAM 1 (system) |
| \_ZL13UART\_Object\_2 | OBJECT | 0x20009BA8 | 4 | DSRAM 1 (system) |
| \_ZL13UART\_Object\_3 | OBJECT | 0x20009BAC | 4 | DSRAM 1 (system) |
| UART\_Channel\_3\_Callbacks | OBJECT | 0x20009F74 | 4 | DSRAM 1 (system) |
| p\_UART\_Object\_3 | OBJECT | 0x20009F04 | 4 | DSRAM 1 (system) |
| p\_UART\_Object\_2 | OBJECT | 0x20009F00 | 4 | DSRAM 1 (system) |
| UART\_Channel\_4\_Callbacks | OBJECT | 0x20009F6C | 4 | DSRAM 1 (system) |
| UART\_Channel\_2\_Callbacks | OBJECT | 0x20009F70 | 4 | DSRAM 1 (system) |

Excel File for Comparison Analysis

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

There might me multiple ways to achieve conversion from C to C++, you can use the above strategies for your design on any platform.

Happy to Share and Help 😊

END