

Hitex Tech Tips

Enlighten Your Development

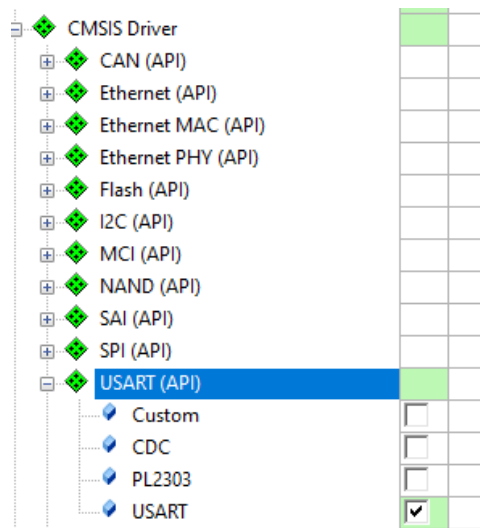
Setting up a CMSIS-Driver in Keil Microvision

By Trevor Martin

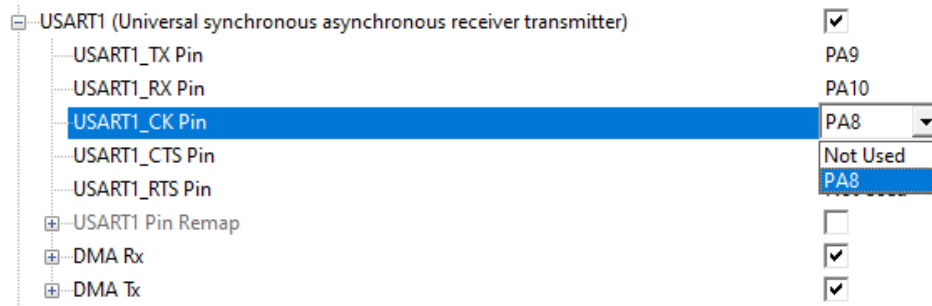
Setting up a CMSIS-Driver

The Common Microcontroller Software Interface Standard (CMSIS) defines a standardised API for a wide range of microcontroller communication peripherals. Using these drivers to access your microcontroller peripherals is a great way to make your code portable and reusable.

In Microvision, open the Runtime Environment Manager and select CMSIS Driver, then select USART.



This will add the CMSIS driver to your project along with a header file called RTE_Device.h. This is a templated file that allows you to configure the USART pins and additional features.



Add the CMSIS Driver header file to your project.

```
#include "Driver_USART.h"
```

Each available instance of the USART driver is declared in the CMSIS-Driver file so we can now select the driver for each USART.

```
extern ARM_DRIVER_USART Driver_USART1;
static ARM_DRIVER_USART * USARTdrv = &Driver_USART1;
```

Next, we can initialize the driver and install a callback which is triggered by the interrupt function.

```
USARTdrv->Initialize(myUSART_callback);
```

Now we can power up and configure the USART.

```
USARTdrv->PowerControl(ARM_POWER_FULL);
USARTdrv->Control(ARM_USART_MODE_ASYNCHRONOUS |
                ARM_USART_DATA_BITS_8 |
                ARM_USART_PARITY_NONE |
                ARM_USART_STOP_BITS_1 |
                ARM_USART_FLOW_CONTROL_NONE, 9600);
USARTdrv->Control(ARM_USART_CONTROL_TX, 1);
USARTdrv->Control(ARM_USART_CONTROL_RX, 1);
```

Once configured we have simple send and receive functions.

```
USARTdrv->Send("\nPress Enter to receive a message", 33);
osThreadFlagsWait(0x01, osFlagsWaitAny, osWaitForever);
```

```
USARTdrv->Receive(&cmd, 1); /* Get byte from UART */
osThreadFlagsWait(0x01, osFlagsWaitAny, osWaitForever);
```

At the end of the send and receive functions we are waiting on an RTOS event flag to signal the end of transmit and receive complete interrupt. These flags are set in the interrupt callback function.

```
void myUSART_callback(uint32_t event)
{
    event &= ARM_USART_EVENT_RECEIVE_COMPLETE | ARM_USART_EVENT_SEND_COMPLETE ;

    switch (event)
    {
        case ARM_USART_EVENT_RECEIVE_COMPLETE:
            case ARM_USART_EVENT_SEND_COMPLETE:
            case ARM_USART_EVENT_TX_COMPLETE:
                /* Success: Wakeup Thread */
                osThreadFlagsSet(tid_Thread, 0x01);
            break;
    }
}
```

Rinse and Repeat

Each of the CMSIS drivers follows a similar API and are provided as standard support for many different Microcontrollers. Once you are familiar with the CMSIS-Driver API it is quick and easy to bring up initial support for many key device peripherals.

Further Information

For more information visit our website: www.hitex.co.uk or get in touch: info@hitex.co.uk. You can also connect with us: [LinkedIn](#)