LET THE HW GUIDE YOU

Datasheet Provides Addresses

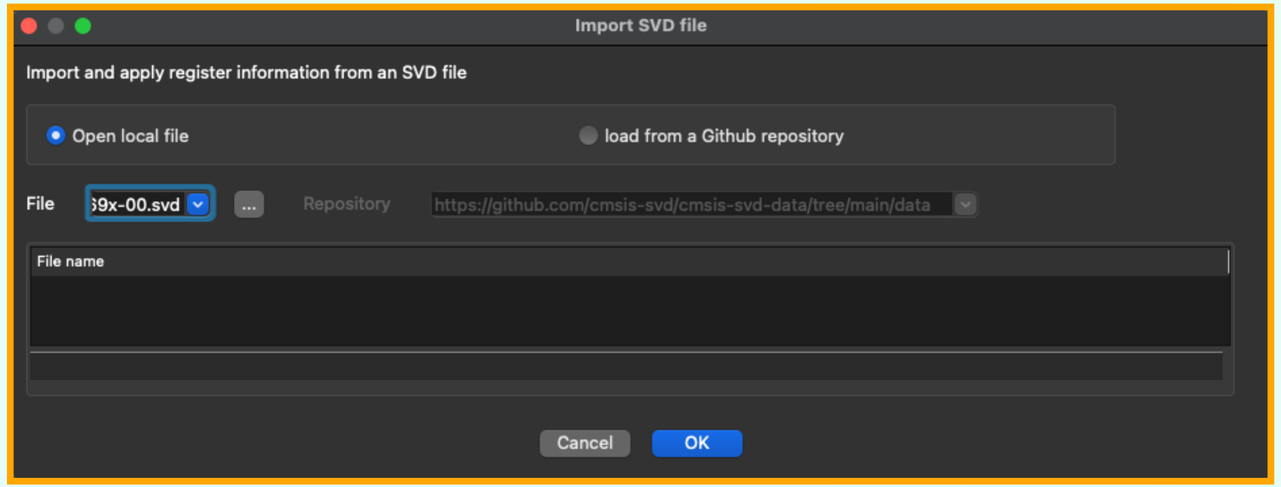
IDA Segments

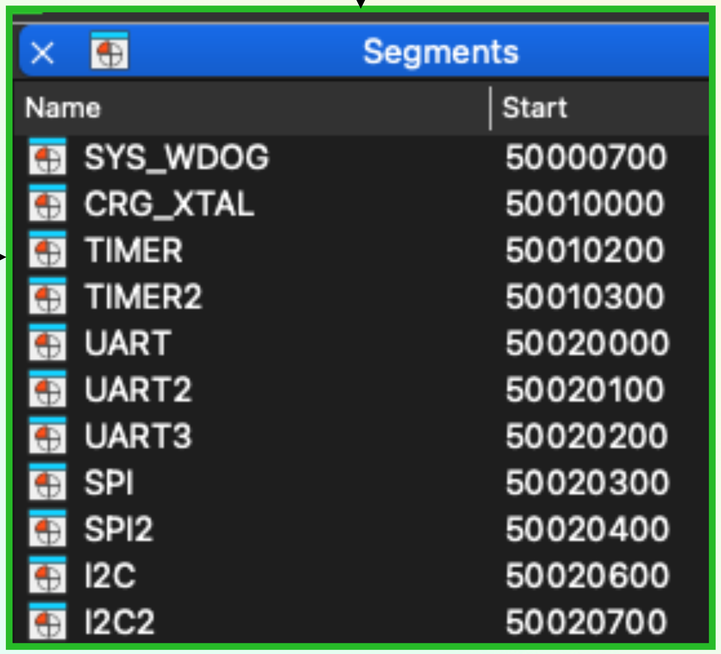
- **Create manually/IDAPython**
- **SVD Loader (Edit>Plugins>SVD File Management)**
  - **SDK_10.0.12.146.1/config/embsys/Dialog_Semiconductor/DA1469x-00.xml**

Resource	Start Address	End Address	Size (kB)
WDOG	50000700	50000800	0,25
Reserved			
XTAL32M_C	50010000	50010200	0,5
TIMER	50010200	50010300	0,25
TIMER2	50010300	50010400	0,25
MAC_TIM	50010400	50010500	0,25
Reserved			
UART	50020000	50020100	0,25
UAR2	50020100	50020200	0,25
UART3	50020200	50020300	0,25
SPI	50020300	50020400	0,25
SPI2	50020400	50020500	0,25

```
<?xml version="1.0" encoding="ascii"?>
<!-- File naming: Dialog_DA1469x.svd -->
<!--
  Copyright (C) 2019-2021 Dialog Semiconductor.
  This computer program includes Confidential, Proprietary Information
  of Dialog Semiconductor. All Rights Reserved.

  Generated by cmsis-svd (version 0.2), d.d. August 26, 2021 - 08:12:40
 -->
<device xmlns:xs="http://www.w3.org/2001/XMLSchema-instance" schemaVersion="1.3" xs:noNamespaceSchemaLocation="CMSIS-SVD.xsd">
  <vendor>Dialog</vendor>                          <!-- device vendor name -->
  <vendorID>ARM</vendorID>                           <!-- device vendor short name -->
  <name>DA1469x</name>                             <!-- name of part-->
  <series>DA1469x</series>                           <!-- device series the device belongs to -->
  <version>1.2</version>                           <!-- version of this description, adding CMSIS-SVD 1.1 tags -->
  <description>690</description>
  <licenseText><!-- this license text will appear in header file. \n forces line breaks -->
```





BACKWARDS IS FORWARD

Register access defines functionality

```
UART:5002000C UART_UART_LCR_REG % 4      ; DATA XREF: sub_264C+14↑r
UART:5002000C                            ; sub_264C+1A↑w ...
UART:5002000C                            ; RW: Line Control Register
```

```
1  int __fastcall sub_264C(int result)
2  {
3    int v1; // r3
4
5    UART_UART_SRR_REG = 7;
6    v1 = UART_UART_LCR_REG;
7    UART_UART_LCR_REG = v1 | 0x80;
8    UART_UART_DLF_REG = (unsigned __int8)result;
9    UART_UART_RBR_THR_DLL_REG = BYTE1(result);
10   UART_UART_IER_DLH_REG = BYTE2(result);
11   UART_UART_LCR_REG = 3;
12   UART_UART_IIR_FCR_REG = 7;
13   UART_UART_IER_DLH_REG = BYTE2(result) & 0xFE;
14   return result;
15 }
```

→

```
1  int __fastcall UART_reset_and_configure_uart(int result)
2  {
3    int v1; // r3
4
5    UART_UART_SRR_REG = 7;                          // reset UART
6    v1 = UART_UART_LCR_REG;
7    UART_UART_LCR_REG = v1 | 0x80;                  // set Divisor Latch Access Bit, required to set
8                                                    // buad rate via DLL/DLH reg
9    UART_UART_DLF_REG = result;                     // set the rate divisor fractional part - 0x1106
10   UART_UART_RBR_THR_DLL_REG = BYTE1(result);      // set the low byte of divisor
11   UART_UART_IER_DLH_REG = BYTE2(result);          // set the hibyte of the divisor
12   UART_UART_LCR_REG = 3;                          // datalen select b11 is 8bits
13   UART_UART_IIR_FCR_REG = 7;                      // setup read interrupt register
14   UART_UART_IER_DLH_REG = BYTE2(result) & 0xFE;
15   return result;
16 }
```

Function Folders

Functions

Function name	Segment	Start	Length
sub_258C	Code	0000258C	00000056
sub_65F4	Code	000065F4	0000005A
sub_BE2	Code	00000BE2	0000005C
sub_264C	Code	0000264C	0000005C
sub_1F46	Code	00001F46	0000005C
sub_2052	Code	00002052	00000064
sub_26AC			
sub_417A			
sub_2A9E			
sub_D78			
sub_452C			

Create folder with items...
Reset natural order
Rename

→

Functions

Function name	Segment	Start	Length	Locals
_				
UART				
sub_2720	Code	00002720	0000003A	00000010
sub_258C	Code	0000258C	00000056	00000010
sub_264C	Code	0000264C	0000005C	00000010
sub_26AC	Code	000026AC	0000006A	00000018
NMI_handler	Code	000024D0	00000002	
SVCall_handler	Code	000024D2	00000002	