# 테스트 환경

- Host Application 은 **STM32CubeIDE : V1.13.2** 에서 컴파일 하였습니다.
- Toolkit Version : V2.8.0.1
- LFW Version : V5.4.0.1

# DeviceNet Slave 구현시 주요 참고 소스

본 예제는 단순 설정을 위주로 작성된 예제입니다. 사용자가 원하는 DeviceNet Object를 추가하기 위해서는 추가 작업이 필요하며 본 예제에는 포함되어 있지 않습니다.

- ***App_IODataHandler*** 함수에서 Configuration이 정상적으로 진행이 된 후 DeviceNet을 통한 Process Data들을 처리하게 됩니다.
- 이 데이타를 이용하여 사용자 데이터로 사용합니다. (모터 데이타 등등...)
- Explicit Message는 App_AllChnnels_PacketHandler 함수를 통하여 처리할 수 있습니다.

참고 파일은 다음 경로에서 확인할 수 있습니다.

\F7_netX_DNSV5_SimpleConfig\Hil_DemoApp\Sources\App_DemoApplication.c

```
int App_CifXApplicationDemo(char *szDeviceName)
{
...

  /* Download the configuration */
  if (CIFX_NO_ERROR != (lRet = App_AllChannels_Configure(&g_tAppData)))
Protocol_StartConfiguration
  {
    PRINTF("Error: A channel failed to configure: 0x%08X" NEWLINE, (unsigned
int)lRet);
    goto error_exit;
  }

  /** now the bus is running */
  while(g_tAppData.fRunning && lRet == CIFX_NO_ERROR)
  {
    /** check and process incoming packets */

    App_IODataHandler(&g_tAppData);
    lRet = App_AllChannels_PacketHandler(&g_tAppData);

    OS_Sleep(1);
  }

...
  return lRet;
}
```

- ***AppDNS_SetConfiguration*** 함수는 DeviceNet Slave의 Parameter들을 설정하는 함수입니다.
- 이 함수에서 VendorID, DeviceType, Product Code, Product Name, Input&Output Size등 DeviceNet 속성과 관련된 파라미터들을 정의합니다.
- 관련 내용은 DeviceNet Protocol Manual에서 확인할 수 있습니다.
- 해당 내용은 EDS의 파일과 동일하게 정의되어야 합니다.

참고 파일은 다음 경로에서 확인할 수 있습니다.

\F7_netX_DNSV5_SimpleConfig\Hil_DemoAppDNS\Sources\AppDNS_DemoApplicationFunctions.c

```
uint32_t AppDNS_SetConfiguration(APP_DATA_T* ptAppData)
{
  uint32_t ulRet = CIFX_NO_ERROR;
  DNS_PACKET_SET_CONFIGURATION_REQ_T* ptReq = AppDNS_GlobalPacket_Init(ptAppData);

  DNS_CONFIGURATION_V1_T *ptCfg = &ptReq->tData.unCfg.tV1;

  /* Set the packet command, length and DNS configuration version */
  ptReq->tHead.ulCmd        = DNS_CMD_SET_CONFIGURATION_REQ;
  ptReq->tHead.ulLen        = DNS_SET_CONFIGURATION_V1_REQ_SIZE;
  ptReq->tHead.ulSta        = 0;
  ptReq->tData.ulVersion    = DNS_CONFIGURATION_V1;

  /* Set the slave related parameters */
  memset(ptCfg, 0x00, sizeof(DNS_CONFIGURATION_V1_T));
  ptCfg->ulSystemFlags      = DNS_SYS_FLG_MANUAL_START;
  ptCfg->ulWdgTime          = 0;

  /* Set network parameter */
  ptCfg->ulNodeId = g_tAppDnsData.bNodeIdValue;
  ptCfg->ulBaudrate = g_tAppDnsData.bBaudRateValue;

  ptCfg->ulConfigFlags      = 0;
  ptCfg->ulObjectFlags      = 0;

  /* Set Identity Data */
  ptCfg->usVendorId         = VENDOR_ID;
  ptCfg->usDeviceType       = DEVICE_TYPE_COMMUNICATION_ADAPTER;
  ptCfg->usProductCode      = PRODUCT_CODE;
  ptCfg->bMajorRev          = MAJOR_REVISION;
  ptCfg->bMinorRev          = MINOR_REVISION;
  ptCfg->bProductNameLen    = sizeof(abProductName)-1;
  memcpy(&ptCfg->abProductName[0], &abProductName[0], ptCfg->bProductNameLen);

  /* Set Process Data */
  ptCfg->ulProduceAsInstance  = DNS_DEMO_PRODUCING_ASSEMBLY_INSTANCE;
  ptCfg->ulProduceAsSize      = DNS_DEMO_PRODUCING_ASSEMBLY_INSTANCE_SIZE;
  ptCfg->ulConsumeAsInstance  = DNS_DEMO_CONSUMING_ASSEMBLY_INSTANCE;
  ptCfg->ulConsumeAsSize      = DNS_DEMO_CONSUMING_ASSEMBLY_INSTANCE_SIZE;
```

```
  ulRet = AppDNS_GlobalPacket_SendReceive(ptAppData);
  return ulRet;
}
```

# 참고 자료

- 메뉴얼 : https://hilscher.atlassian.net/wiki/spaces/DL/pages/77734767/DeviceNet+Slave+V4+V5
- 예제 소스 : https://hilscher.atlassian.net/wiki/spaces/NXLFWHST/pages/101286754/DeviceNet+Slave+Examples