Lab Cortex-M4

CAN Communication
Before you begin

- Connect CAN cable to the CAN connector of the board
- Correct polarity
STM32CubeMX

- Open STM32CubeMX and click New Project
Board Selector

- Select Board Selector Tab
- Select Discovery and STM32F4
- Double click STM32F407VGTx MCU
Select Settings from Project menu
- Project Name: Can
- Select Code Generator Tab
- Select Add necessary library files as reference in the toolchain project configuration file
- Click OK
Pinout Selection

- CAN1: Master Mode
- I2C1: I2C
- I2C3: I2C
- USART2: Asynchronous
- USART3: Asynchronous
Clock Configuration

- Select Clock Configuration
- Type 168 in HCLK and Enter
- Check 42MHz for APB1 peripheral clocks
- Select Configuration Tab
- Click CAN1 and select Parameter Settings
- Change Time Quanta to 2 Times and 3 Times
- Change Prescaler to 7
- Click OK
CAN Interrupt Setting

- Select NVIC Settings
- Check CAN1 RX0 interrupts
- Click OK
- Save from File menu
- Generate Code from Project menu
- Do not click Open Project
- Open folder or Close
Open Visual Studio 2015

- Start New Project
- Select VisualGDB and Embedded Project Wizard
- Check the location and change if necessary
- Change name to Can
- Click OK
- Import a project built with command-line-tools
- Generate a makefile to build the project
- **Uncheck** Copy imported files to project directory
- Click Next
- Type STM32F407VG and double click the device
- Or select the device and click Next
- Directory for imported source D:\work\stm32cube\Can

New Embedded Project

**Import Source**

- **Import preserving directory structure**
  VisualGDB will create virtual subfolders inside "source files" and "header files" in Project Explorer.

- **Show all source files together**
  All source files from all subdirectories will be put together in "Source files" view in Project Explorer.

- **Do not import source files to Visual Studio project**
  The created project will not link to any source files. Building and debugging the project will be possible, however, the source files will not be shown in Project Explorer and won't be parsed by IntelliSense. You will be able to add the files to the project manually using the "Add existing files" command.
- Check debugging interface and click Finish
Select VisualGDB Project Properties
- Uncheck STM32F4 Default init File
- Click Apply
- Set Include directories
- Click OK
Build Solution
- Double click main.c in Solution Explorer
- Add the following code in main.c

```c
int main(void)
{
    /* USER CODE BEGIN 1 */
    /* USER CODE END 1 */
    /* MCU Configuration */
    /* Reset of all peripherals, Initializes the Flash interface and the System
     HAL_Init();
    */
}
```
/* USER CODE BEGIN 2 */
CAN_FilterConfTypeDef sFilterConfig;
static CanTxMsgTypeDef TxMessage;
static CanRxMsgTypeDef RxMessage;
/* Configure the CAN Filter */
sFilterConfig.FilterNumber = 0;
sFilterConfig.FilterMode = CAN_FILTERMODE_IDMASK;
sFilterConfig.FilterScale = CAN_FILTERSCALE_32BIT;
sFilterConfig.FilterIdHigh = 0x0000;
sFilterConfig.FilterIdLow = 0x0000;
sFilterConfig.FilterMaskIdHigh = 0x0000;
sFilterConfig.FilterMaskIdLow = 0x0000;
sFilterConfig.FilterFIFOAssignment = 0;
sFilterConfig.FilterActivation = ENABLE;
sFilterConfig.BankNumber = 14;

if (HAL_CAN_ConfigFilter(&hcan1, &sFilterConfig) != HAL_OK)
{
    /* Filter configuration Error */
    Error_Handler();
}
hcan1.pTxMsg = &TxMessage;
hcan1.pRxMsg = &RxMessage;
hcan1.pTxMsg->StdId = 0x001;
hcan1.pTxMsg->RTR = CAN_RTR_DATA;
hcan1.pTxMsg->IDE = CAN_ID_STD;
hcan1.pTxMsg->DLC = 8;

/* Test CAN data transmission */
hcan1.pTxMsg->Data[0] = 0x12;
hcan1.pTxMsg->Data[1] = 0x34;
if (HAL_CAN_Transmit(&hcan1, 100) != HAL_OK)
{
  /* Transmission Error */
  Error_Handler();
}
if (HAL_CAN_Receive_IT(&hcan1, CAN_FIFO0) != HAL_OK)
{
  /* Reception Error */
  Error_Handler();
}
/* USER CODE END 2 */
/* USER CODE BEGIN 4 */
void HAL_CAN_RxCpltCallback(CAN_HandleTypeDef* CanHandle)
{
    if (CanHandle->pRxMsg->StdId == 0x002)
    {
        hcan1.pTxMsg->Data[0] = hcan1.pRxMsg->Data[0];
        hcan1.pTxMsg->Data[1] = hcan1.pRxMsg->Data[1];
        if (HAL_CAN_Transmit(&hcan1, 100) != HAL_OK)
        {
            /* Transmition Error */
            Error_Handler();
        }
    }
    /* Receive */
    if (HAL_CAN_Receive_IT(CanHandle, CAN_FIFO0) != HAL_OK)
    {
        /* Reception Error */
        Error_Handler();
    }
}
/* USER CODE END 4 */
Get ready PCANView and run the program
PCANView

- Right click to create a New CAN message
- Manual transmission: 0ms Period
Double click the message to transmit CAN data from PC to the board
Design Project

- STM32F407 실습 보드의 CAN 통신을 이용하여 엘리베이터 시뮬레이터를 제어하는 프로그램을 작성한다.
- Rhapsody를 이용하여 작성한 프로그램과 동일한 기능을 구현한다.
- 보고서의 결론(Discussion)에는 UML을 이용한 프로그래밍 방법과 전통적인 프로그램 언어를 이용한 방법을 비교하고 각각의 장단점에 대해서 본인의 생각을 기술한다.