## Task 4: Connect to STM32-Based SPI Device

* *(a) Introduction and High-Level Description*
  + In this task, students need to use SPI to communicate. A STaTS device, Nucleo-F303RE board, is provided as a peripheral in this task. The DISCO board is the controller for SPI.
  + Since the firmware is given, only the program on the DISCO board need to be written by student.
  + There are multiple small tasks in this task, the following is the method to fulfill each task:
    - Send terminal characters from DISCO to peripheral: send each input character to peripheral unless ESC for calling a menu.
    - Receive terminal characters from peripheral device: a function in the menu open the receive mode, each time 1 character can be received.
    - Read firmware version upon start up. This function is called after initialization, receive data from the peripheral and print it on the terminal
    - Read temperature: a function in the menu, open receive temperature mode. Print the temperature data on the terminal when reading is ready.
    - Clear peripheral terminal: function in the send data to certain register to clear the terminal screen.
    - Change Device ID: a function in the menu. Get the new ID from keyboard and send it to peripheral
  + Wire connection:

Table 2: Task 4 schematic

|  |  |  |
| --- | --- | --- |
|  | Connected to | Connected to |
| SPI\_MISO | D11 DISCO board | D11 peripheral |
| SPI\_SIMO | D12 DISCO board | D12 peripheral |
| SPI\_CLK | D13 DISCO board | D13 peripheral |
| SPI\_CS | D10 DISCO board | A2 peripheral |

* *(b) Low Level Description*
  + All the data using in this task is from the data sheet, STaTS device Datasheet.
  + The basic idea of SPI is set the CS pin to reset, set a delay of 10us, then transmit to locate the required bit. Set a delay, transmit or receive data from the peripheral. Set a delay and set the CS pin to set status.
  + For each transmission requires a delay.
  + The GPIO setting for the pin to operate SPI is important Pin D11, 12, and 13 are for SPI\_SIMO, SPI\_MISO, and SPI\_CLK, these pins need to be set to alternate function with push-pull. Pin D10, SPI\_CS, need to be set to output push-pull mode.
  + The SPI transfer setting need to set correctly to transform data. Polarity low, phase 2 edge, NSS soft, baud pre-scaler 128, data size 8 bit, first bit most significant bit.
  + Here student use the HAL\_GPIO\_WritePin to change the status of CS pin, the part is the same as it used in LAB 1.
  + HAL\_SPI\_Transmit, HAL\_SPI\_Receive, and HAL\_SPI\_TransmitReceive[citation] are used for transmit or receiving data. The first parameter is the SPI configuration, the second is the bit of register is the data to transfer, it can be a register number or the data to be sent to the peripheral
  + For sending and receiving character, register 5, CH\_BUF, is used. Sending data uses receive, receiving data use TransmitRecevie function by sending 0x00 prevent the peripheral receive data. Then receive the 3 bit data from it. To check if data is in the peripheral data, the keep checking status register bit 6 and 5 until one of them is not 0. Then request the data in buffer and print it in the terminal.
  + For the firmware, register 7 and 8 are used.
  + For the temperature, first send data to control register bit 1 to request a new temperature read. Then keep checking status bit 3 until it is 1 when the temperature ready. Then ready the register 3 and 4 to get the 12 bit temperature number. The Celsius degree of temperature need to be changed:

Temperature = 357.6 – 0.187 \* value

* + To change the device ID of the peripheral, read register 9 DEVID to get the old ID. First set bit 7 of control register to 1 to enable the device ID change. With this step, the controller will be able to change ID. Get the ID from keyboard, change the input char to into by minus it to “0”, otherwise the device ID will be AscII number. Send the data to the register 9 DEVID.
* *(c) Results and Analysis:*
  + The program worked as expected
  + The delay is significant in SPI. Without delays, the reading and sending may not work properly.
  + The SPI connection may be influenced by the wire connection, the SPI may not work if the wire had a bad connection. So the SPI have chance not to work.