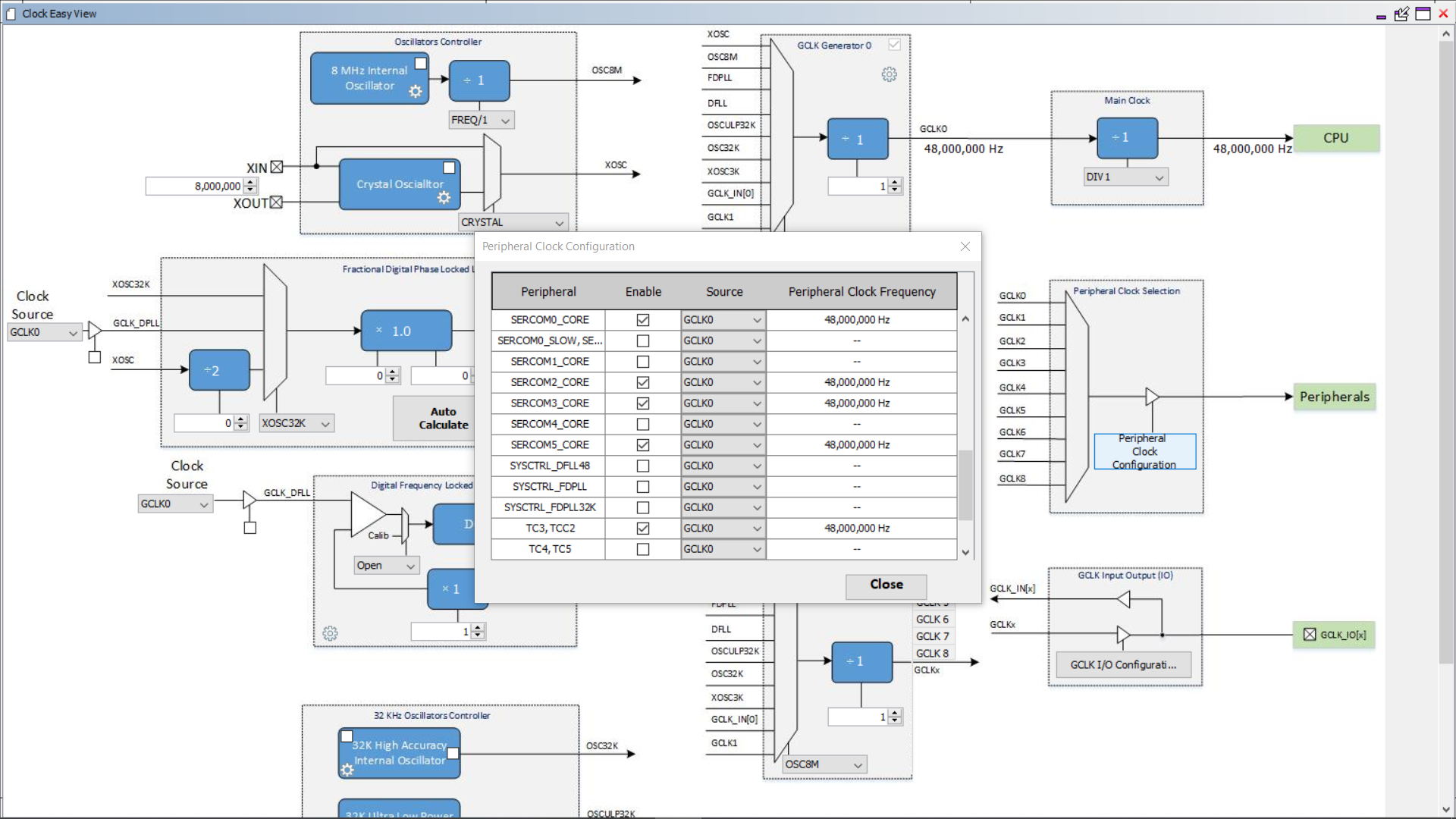
**SPI**

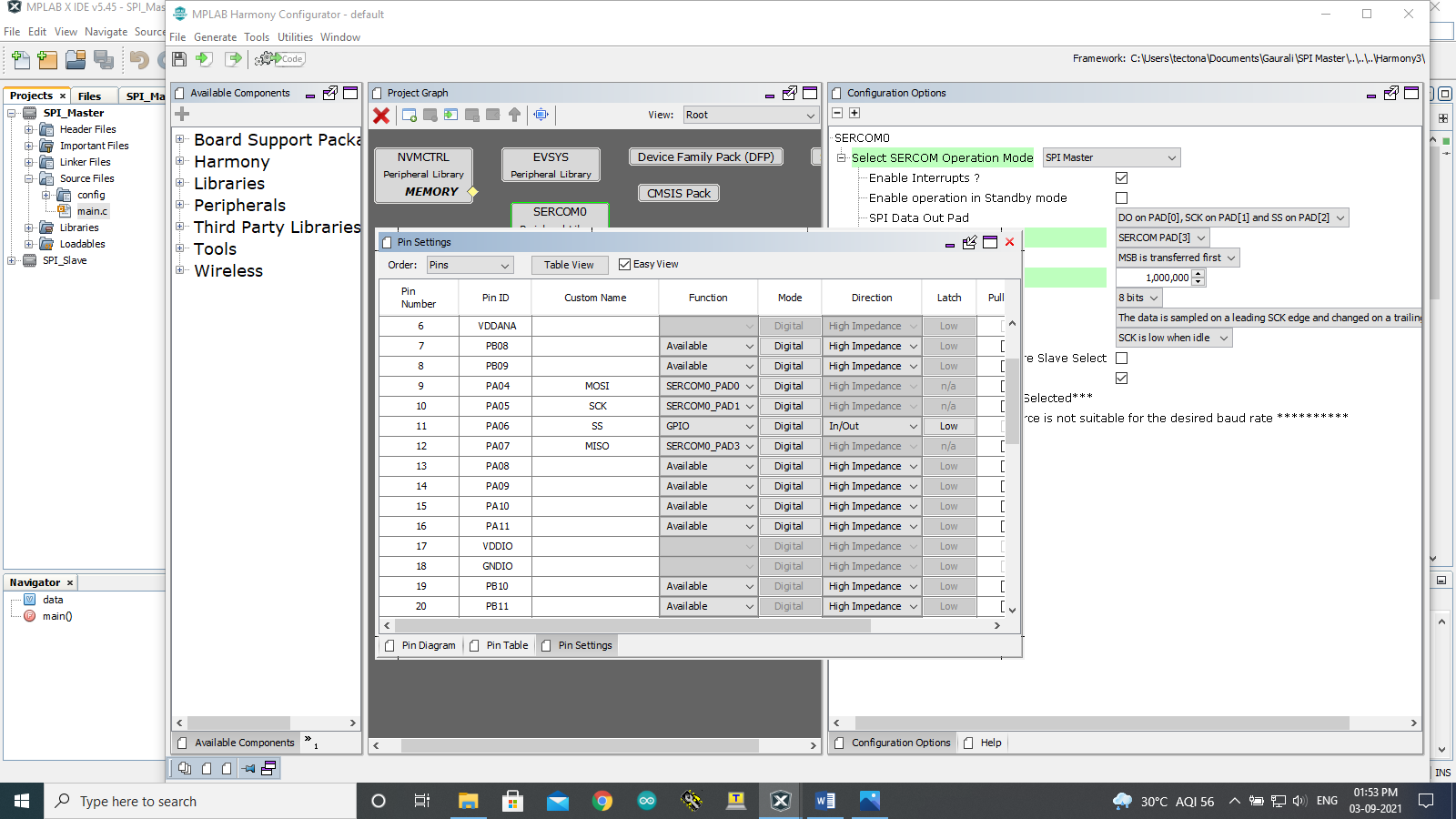
Step 1: Create a MPLAB Project and Open MHC (Refer to Creating First project).

Step 2: Clock Configuration:

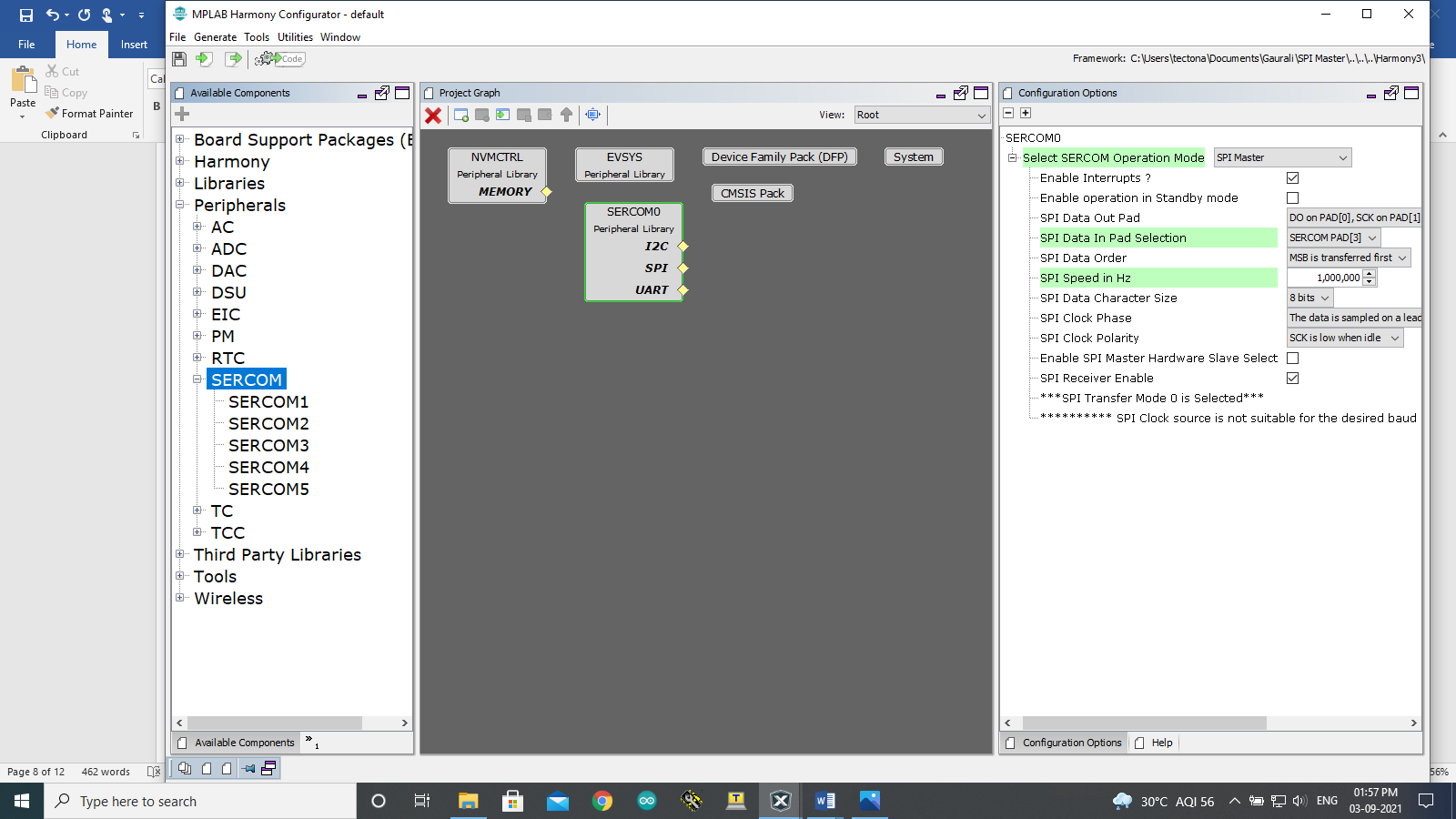


**SPI Master:**

Step 3: Set PA04 as MOSI, PA05 as SCK, PA06 as SS and PA07 as MISO.



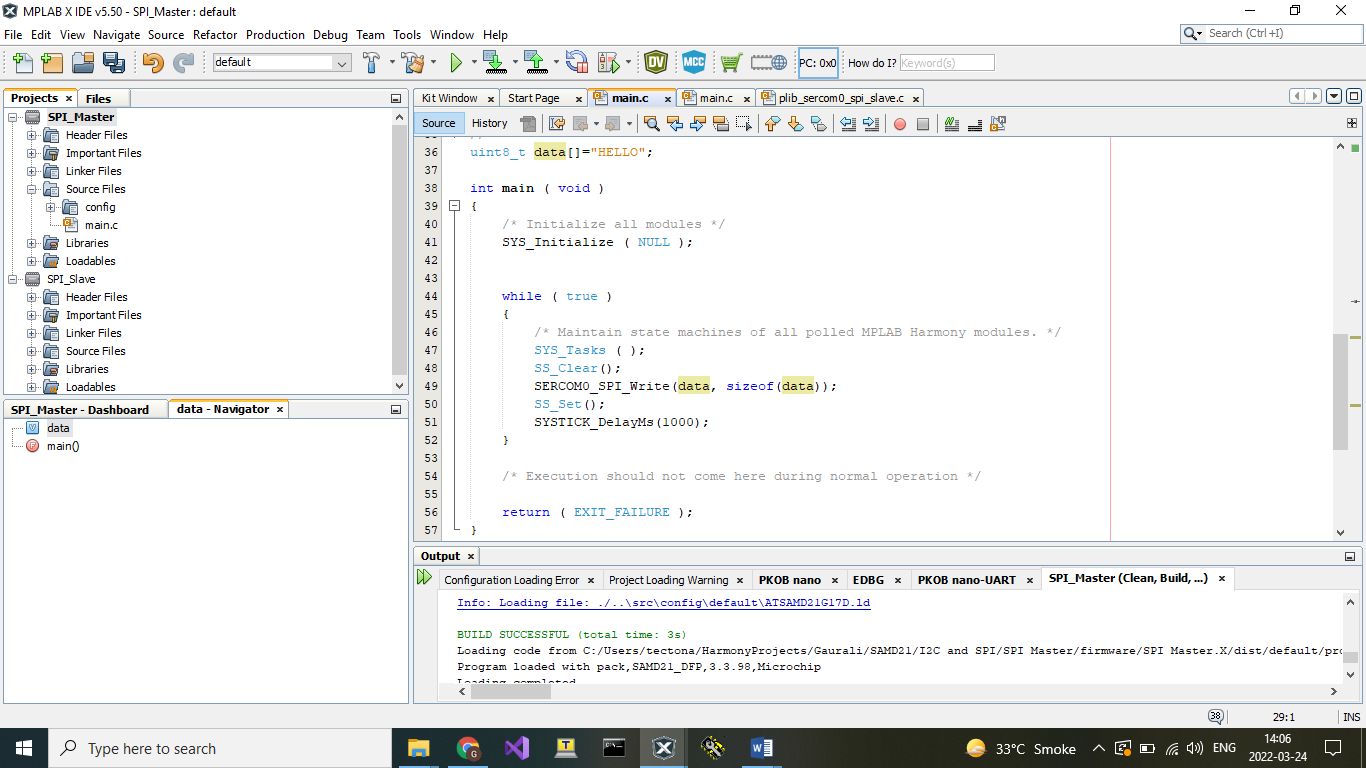
Step 4: Click on Peripherals and select the SERCOM0 and set the configurations as shown below.



Step 5: Click on Generate Code.

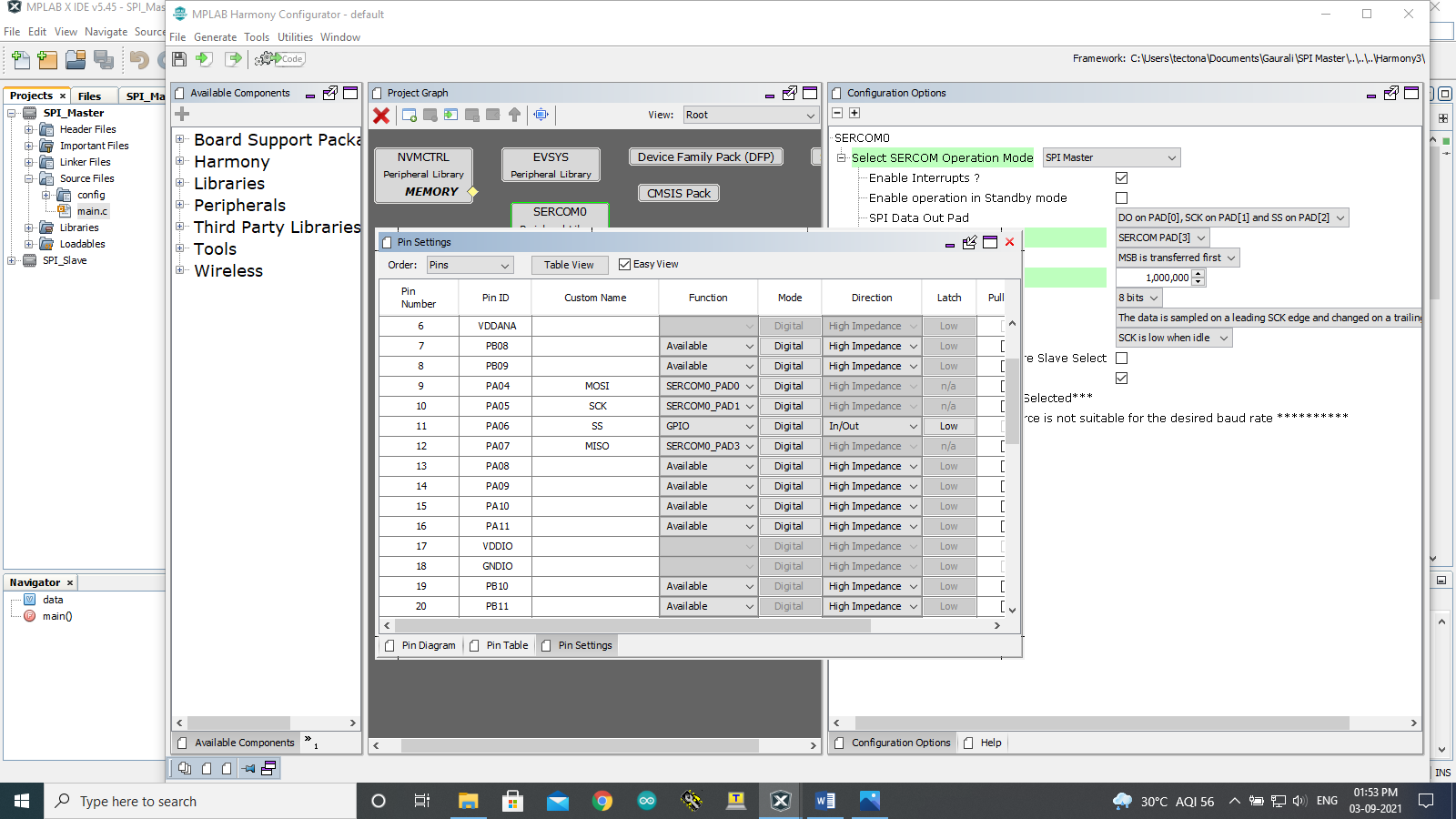
You should be able to see MHC generated files under Project->Source Files->config->default->peripheral->sercom0->plib\_spi\_master.c

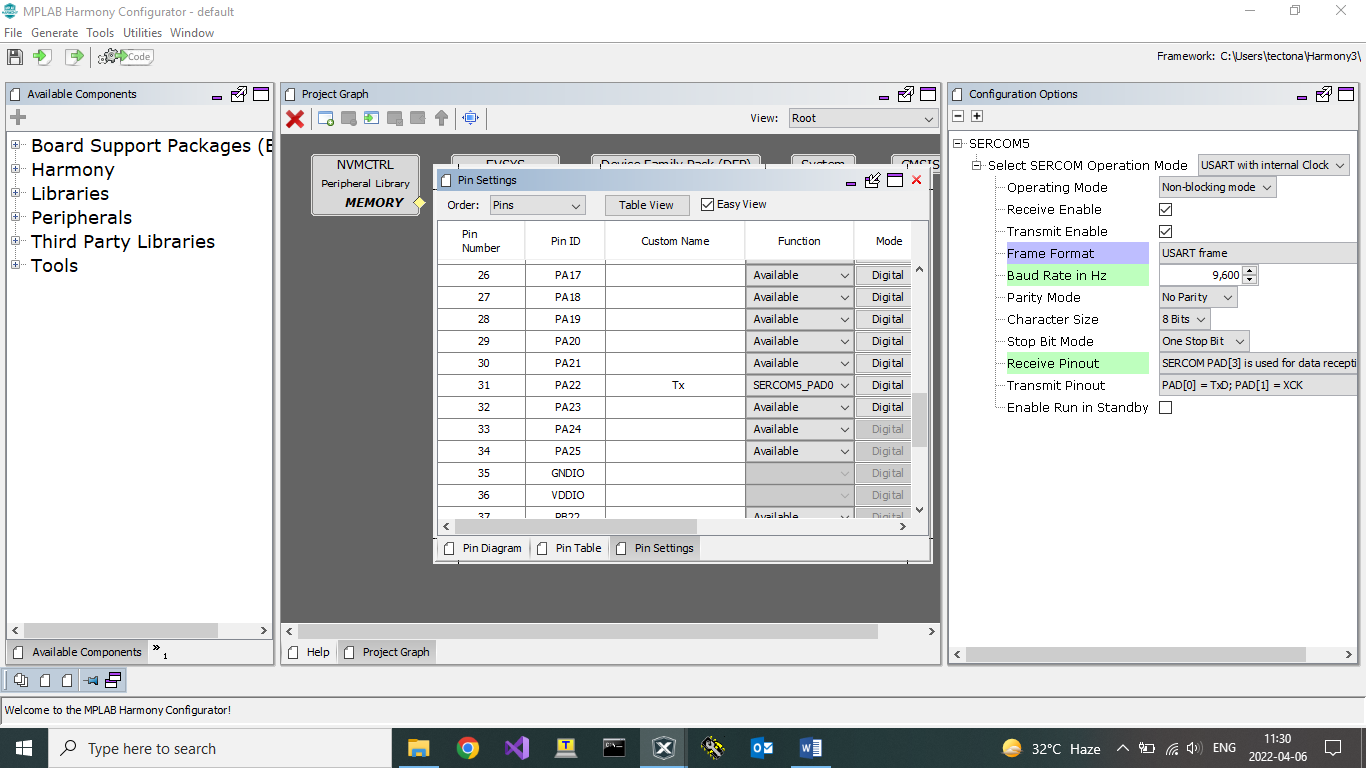
Master code:



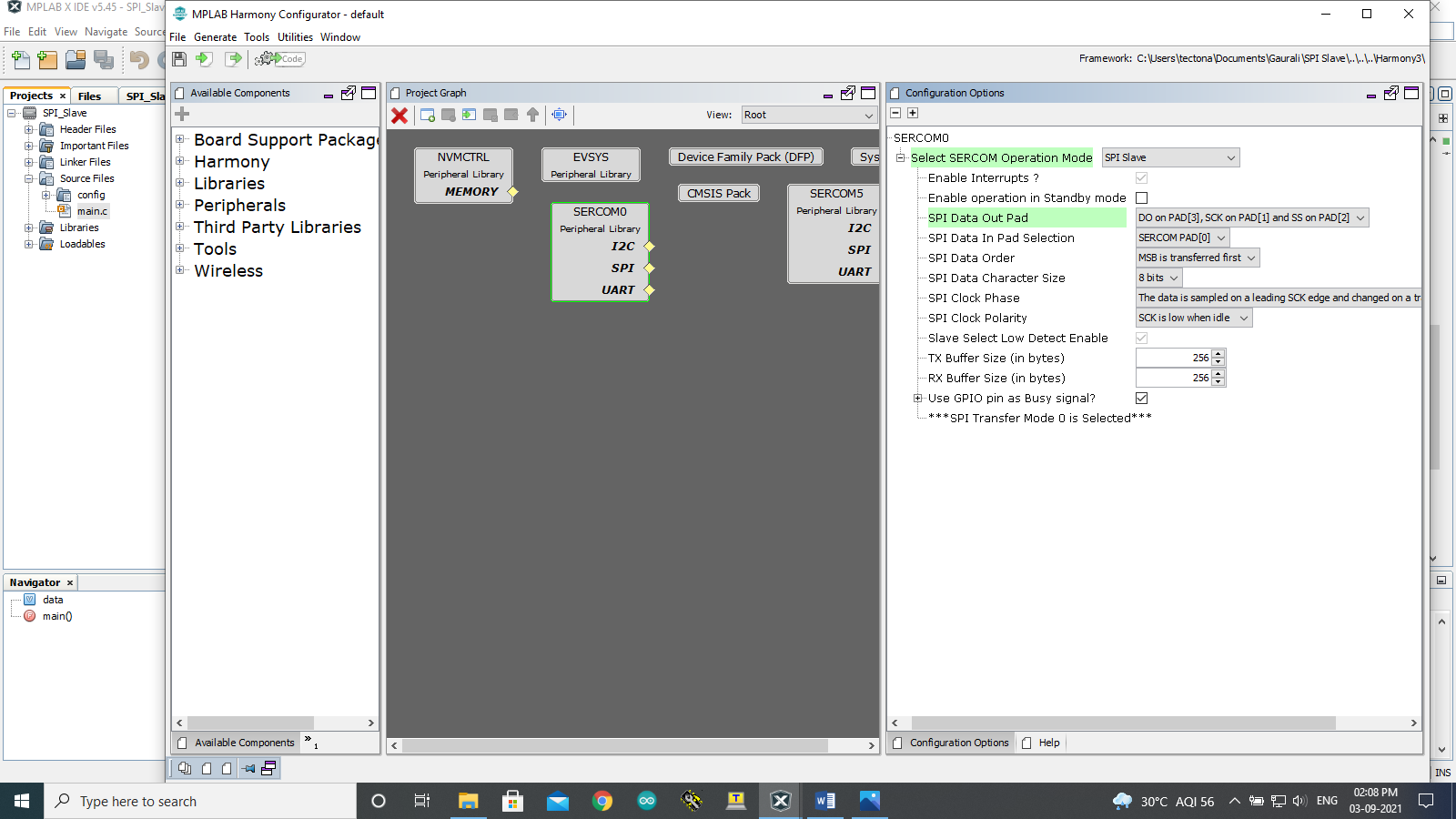
**SPI Slave:**

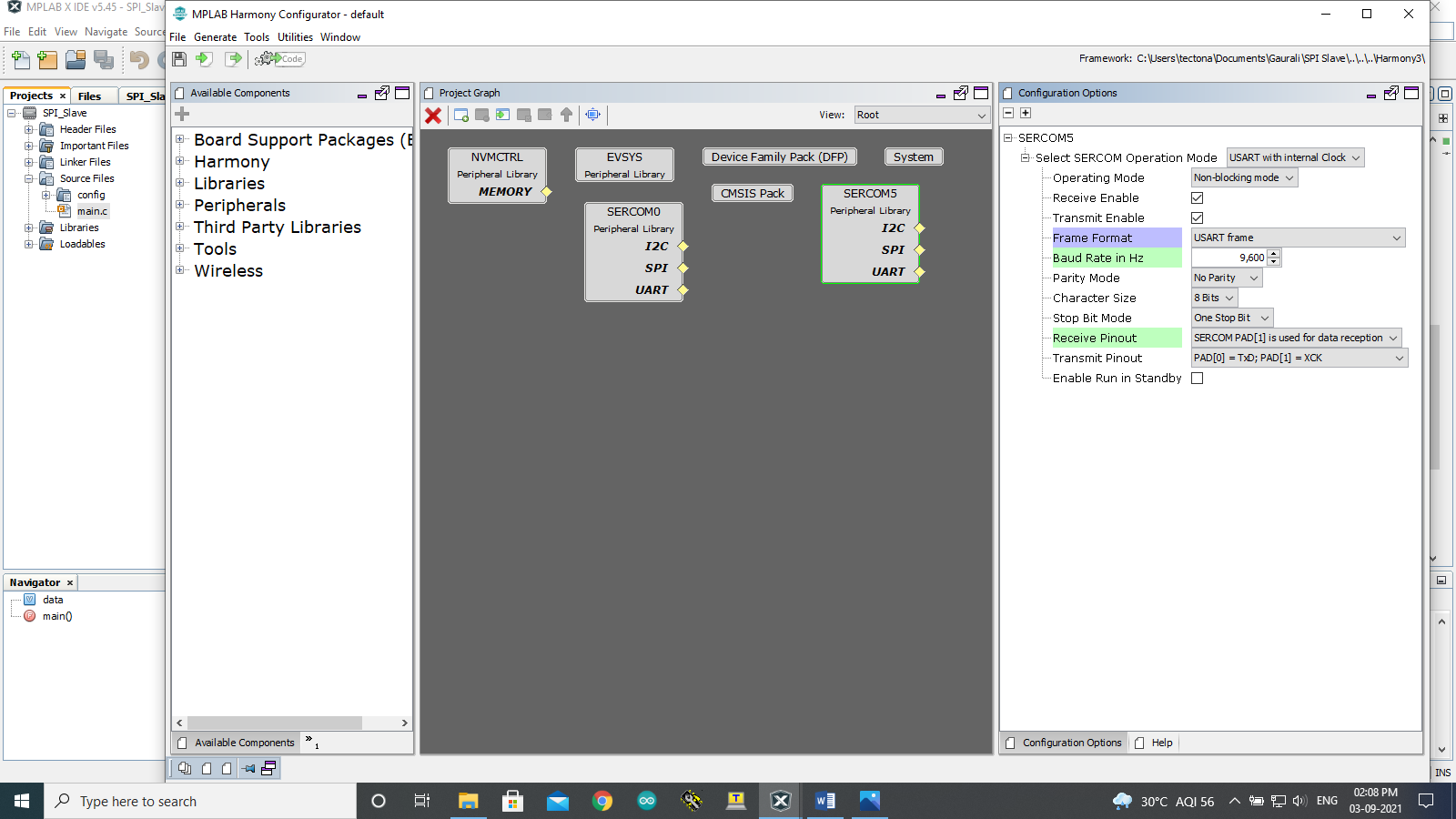
Step 3: Set PA04 as MOSI, PA05 as SCK, PA06 as SS, PA07 as MISO and PA22 as Tx.





Step 4: Click on Peripherals and select SERCOM0 and SERCOM5. Configure them as shown below.





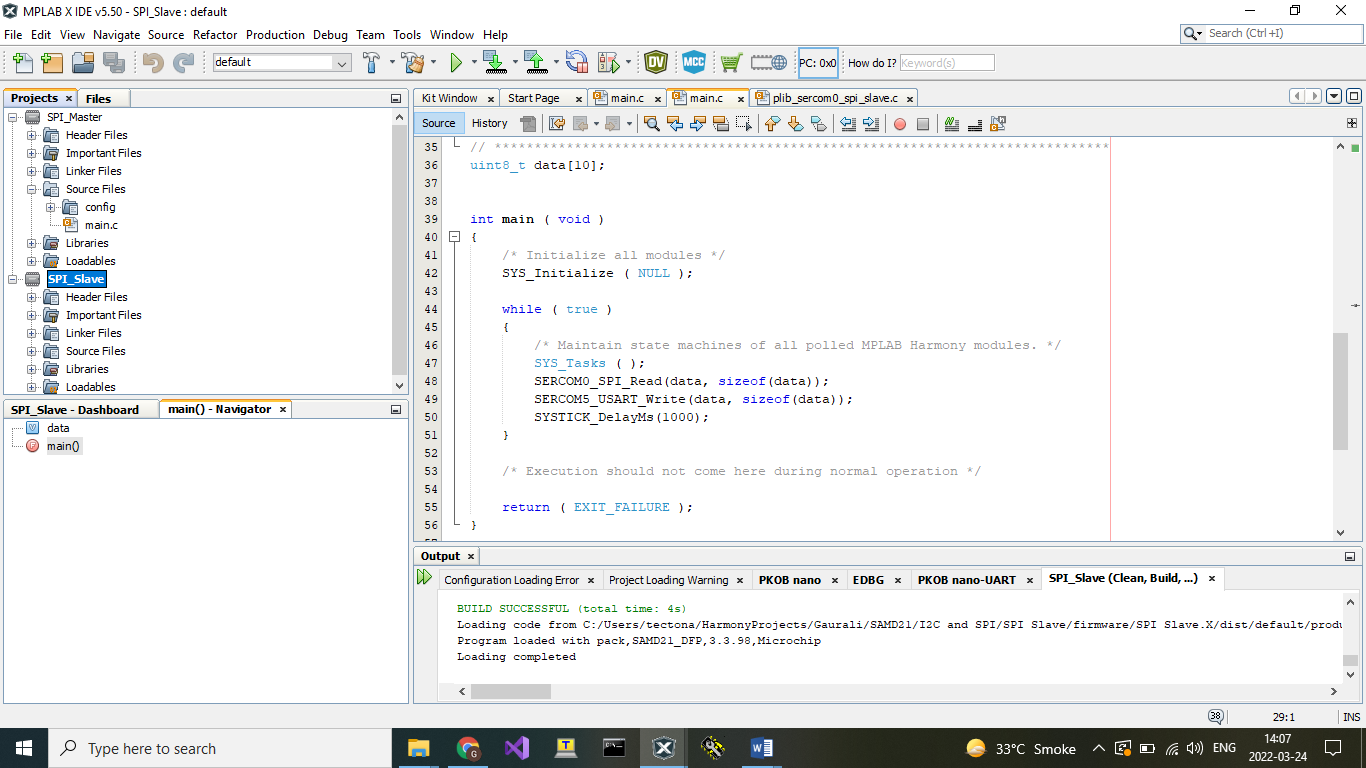
Step 5: Click on Generate Code.

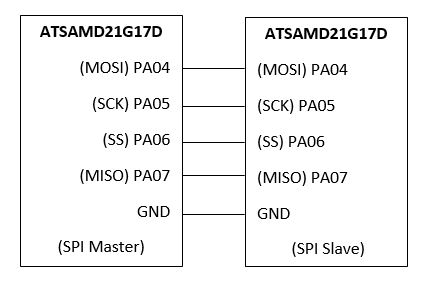
You should be able to see MHC generated files under

Project->Source Files->config->default->peripheral->sercom2->plib\_spi\_slave.c

Project->Source Files->config->default->peripheral->usart- >plib\_sercom5\_usart.c.

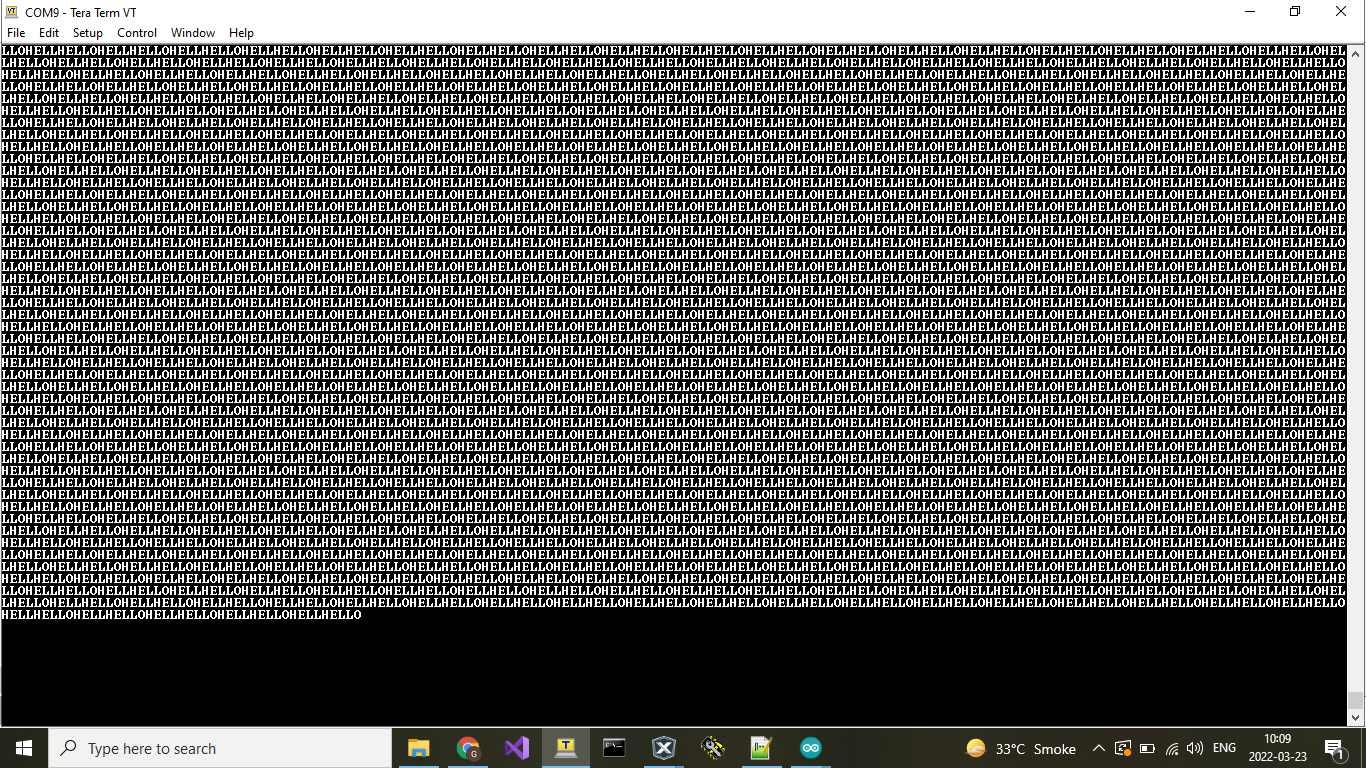
Slave code:





You can see the output using TeraTerm on the COM of Slave board.

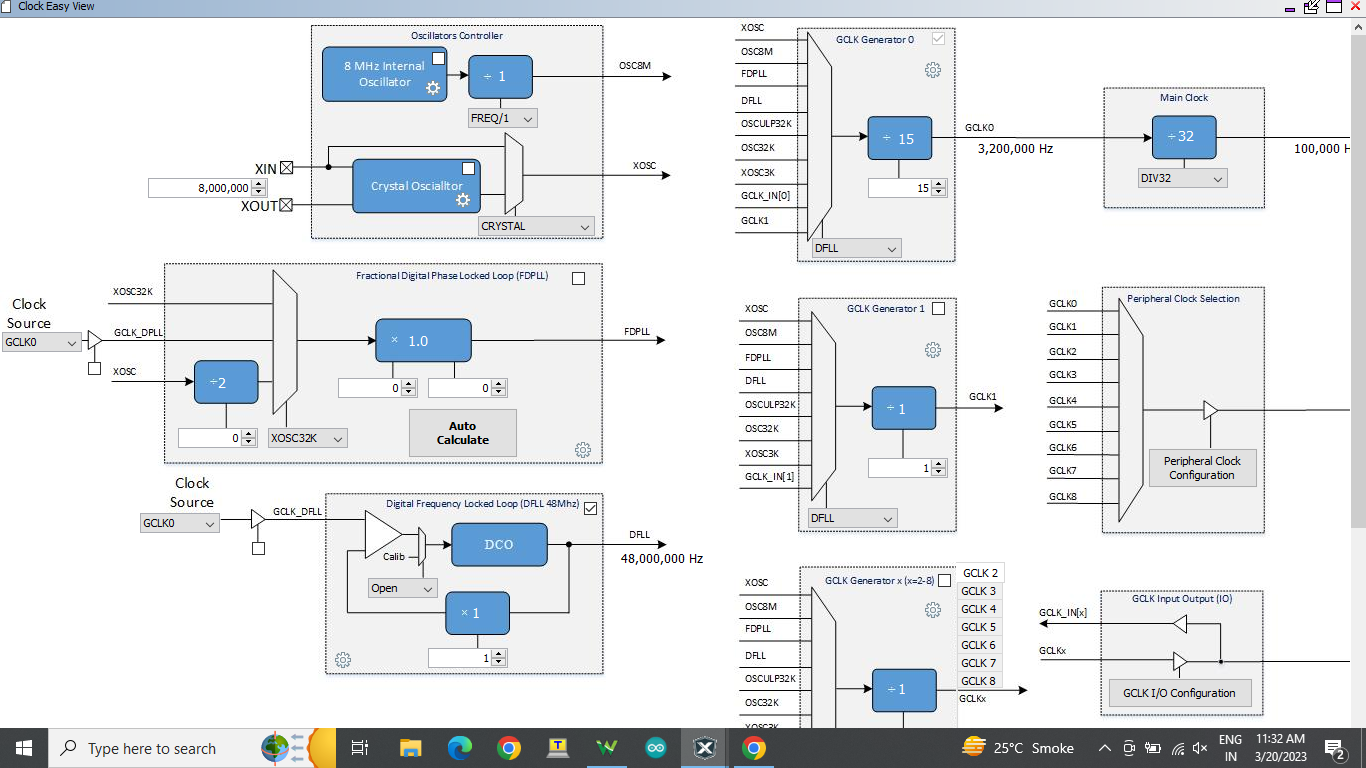
Output:



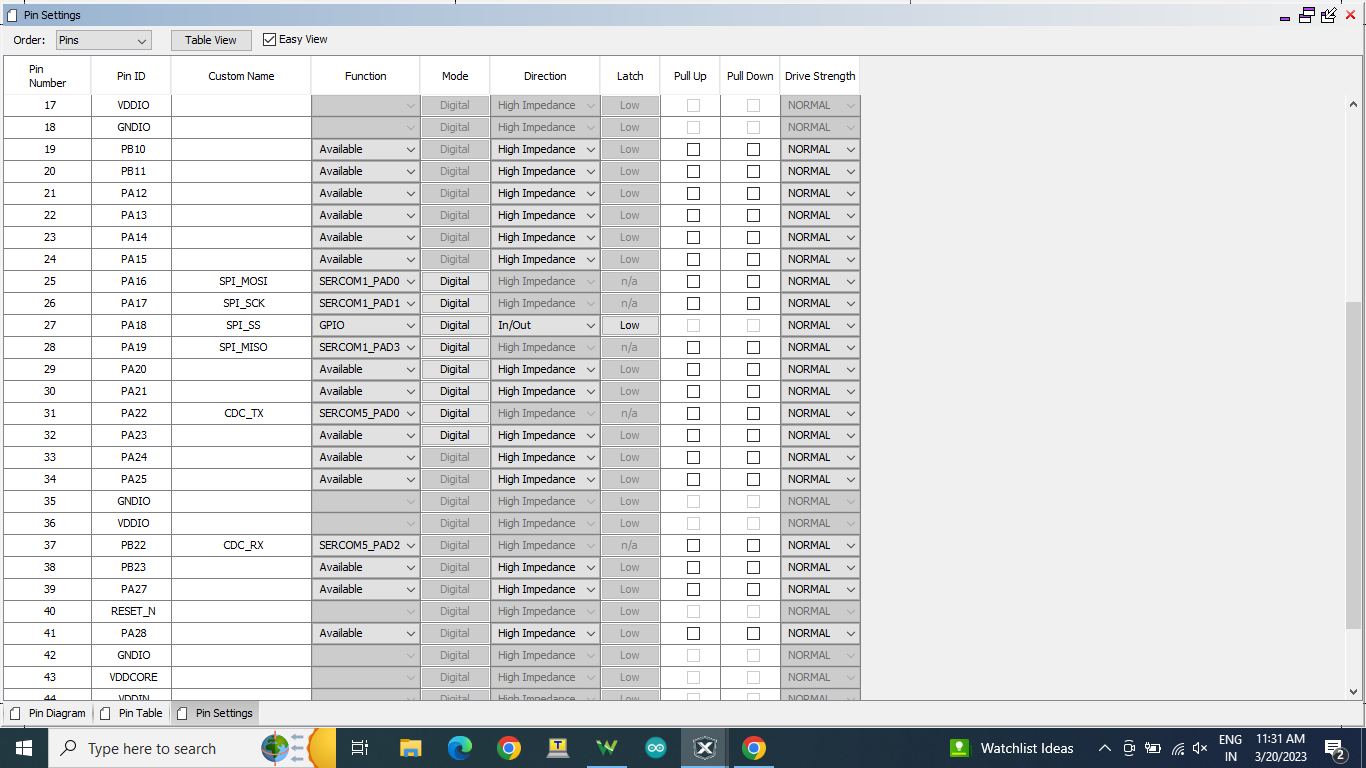
**SPI DMA – Master:**

Step 1: Create a MPLAB Project and Open MHC (Refer to Creating First project).

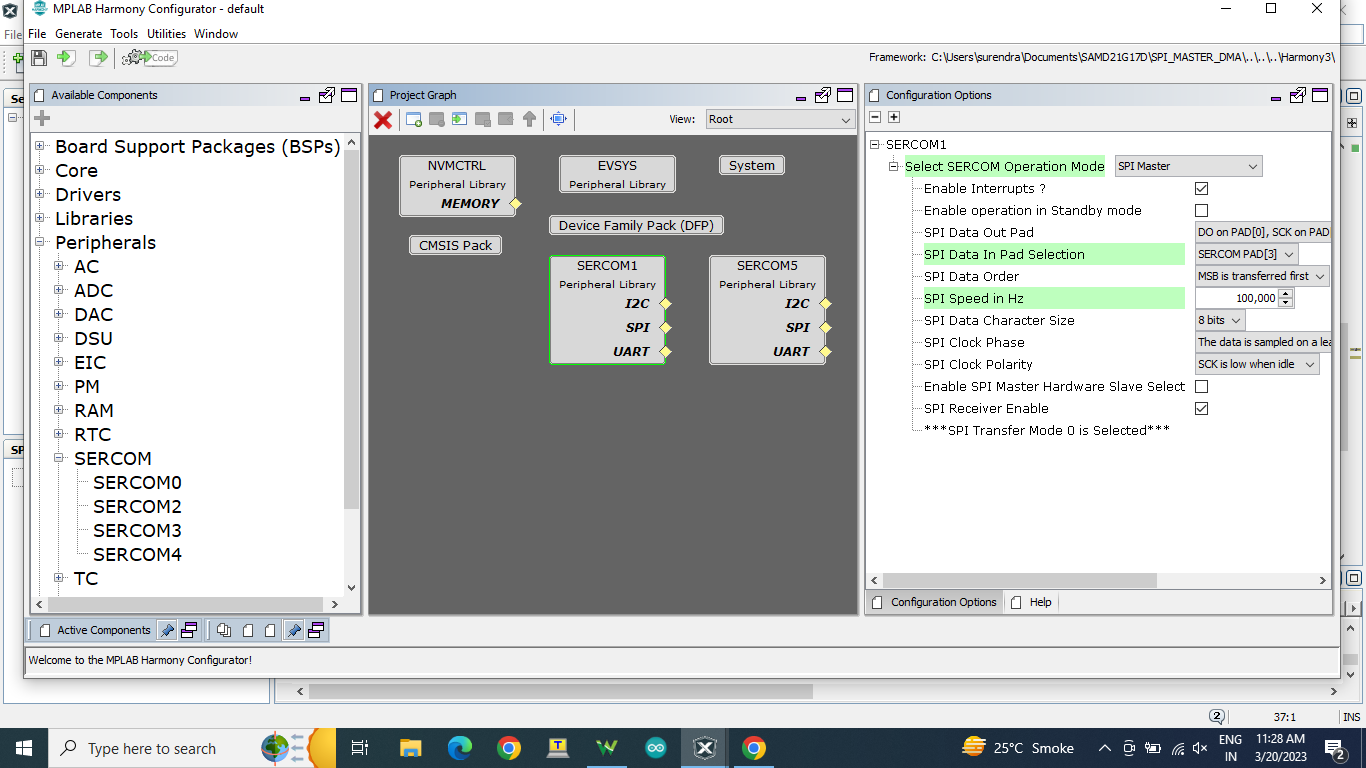
Step 2: Clock Configuration:



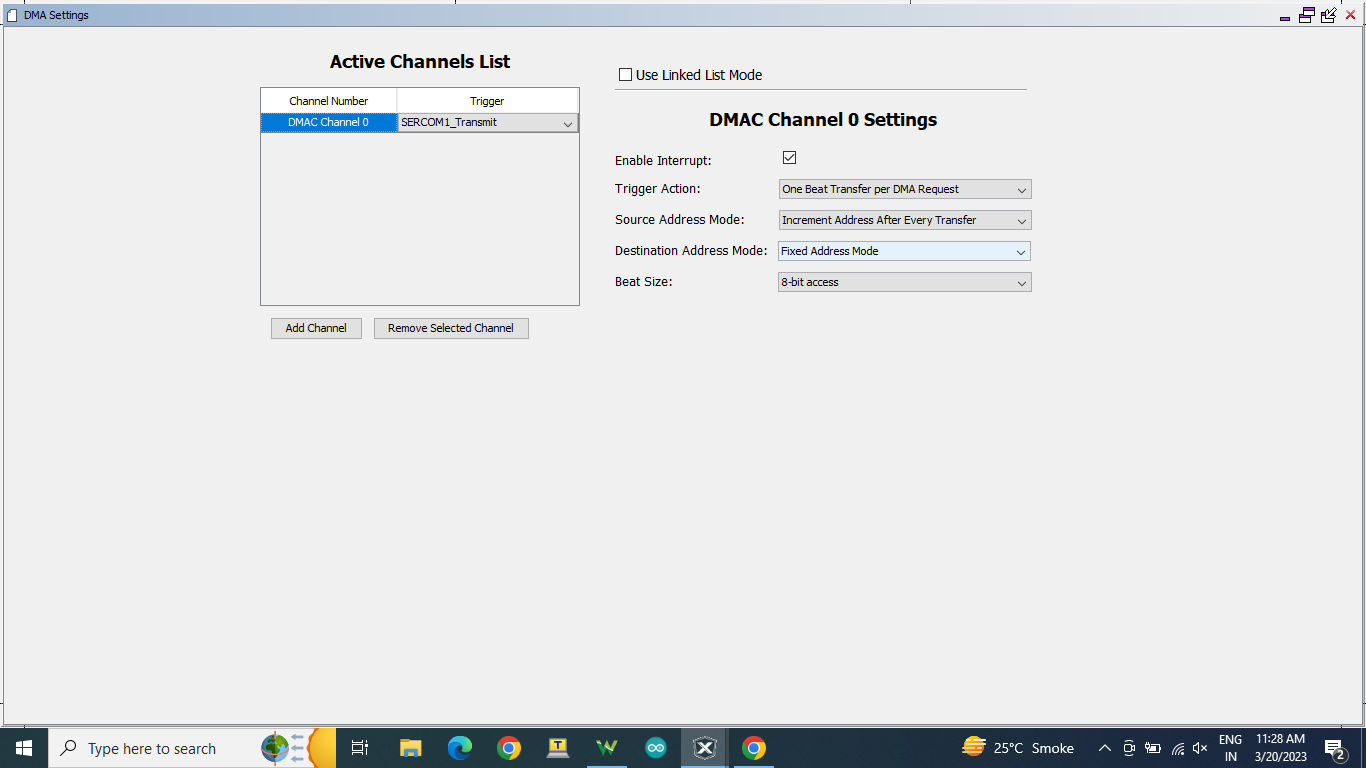
Step 3: Set PA16 as MOSI, PA17 as SCK, PA18 as SS and PA19 as MISO. PA22 as CDC TX and PB22 as CDC RX.



Step 4: Click on Peripherals and select the SERCOM1 and SERCOM5 and set the configurations as shown below.



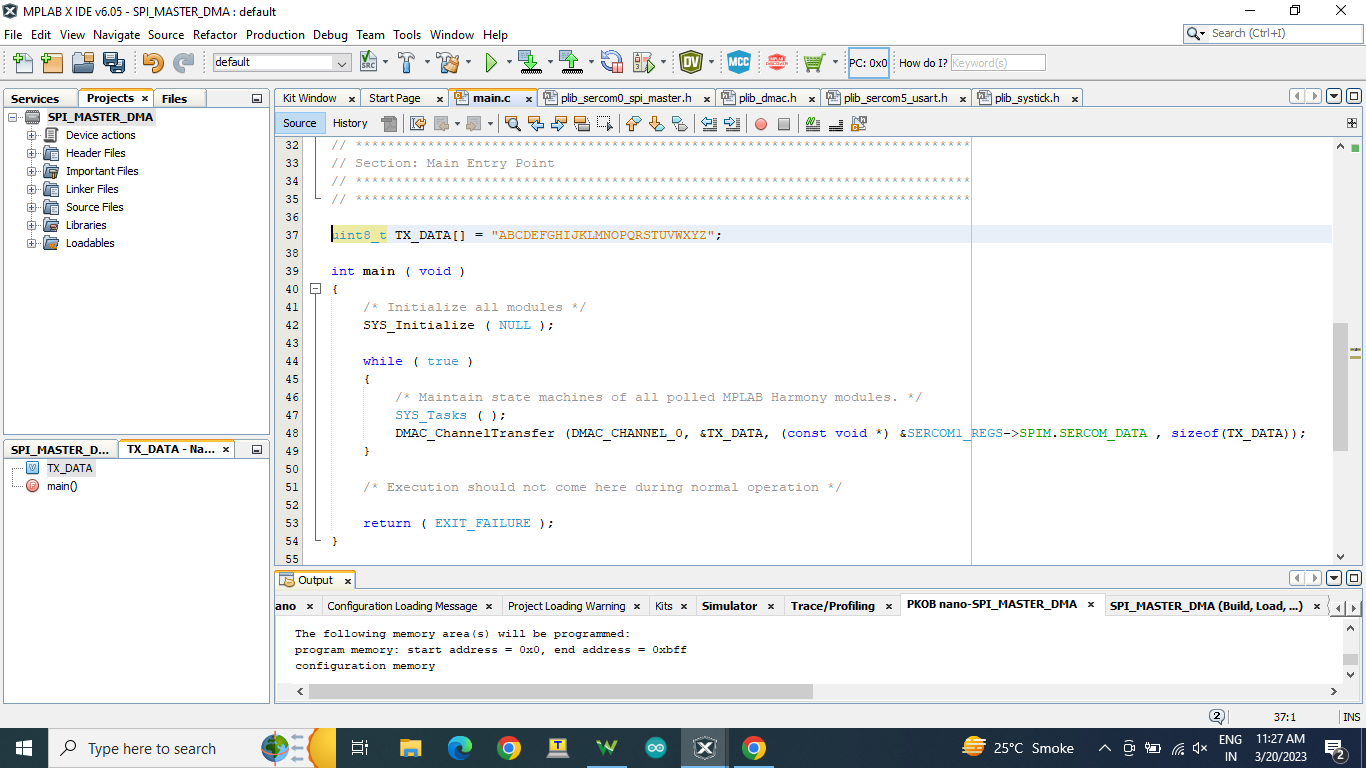
Step 5 : Set DMA configuration as per shown below.



Step 6: Click on Generate Code.

You should be able to see MHC generated files under Project->Source Files->config->default->peripheral->sercom0->plib\_spi\_master.c is used for function and main.c is used for code.

Master code:



**Master Sample Code:**

uint8\_t TX\_DATA[] = "ABCDEFGHIJKLMNOPQRSTUVWXYZ\n";

int main ( void )

{

/\* Initialize all modules \*/

SYS\_Initialize ( NULL );

while ( true )

{

/\* Maintain state machines of all polled MPLAB Harmony modules. \*/

SYS\_Tasks ( );

DMAC\_ChannelTransfer (DMAC\_CHANNEL\_0, &TX\_DATA, (const void \*) &SERCOM1\_REGS->SPIM.SERCOM\_DATA , sizeof(TX\_DATA));

}

/\* Execution should not come here during normal operation \*/

return ( EXIT\_FAILURE );

}

**OUTPUT :**

Master will send data to Slave as shown below.

