# Exercises

1. Examine the reference manual for your board to determine the answers to the following questions.
   1. Which pins can be used for UART?

UART1

PA8:USART1\_CK

PA9,PB6:USRT1\_TX

PA10,PB7:USART1\_RX

PA11:USART1\_CTS

PA12:USART1\_RTS

UART2

PA0,PD3:USART2\_CTS

PA1,PD4:USART2\_RTS

PA2:USART2\_TX

PA3:USART2\_RX

PA4:USART2\_CK

UART3

PB10,PC10,PD8:USART3\_TX

PB11,PC11,PD9:USART3\_RX

PB12,PC12,PD10:USART3\_CK

PB13,PD11:USART3\_CTS

PD12:USART3\_RTS

* 1. Which pins can be used for I2C?

I2C1

PB5:I2C1\_SMBA

PB6,PB8:I2C1\_SCL

PB7,PB9:I2C1\_SDA

I2C2

PB10:I2C2\_SCL

PB11:I2C2\_SDA

PB12:I2C2\_SMBA

* 1. Which pins can be used for SPI?

SPI1

PA4,PA15:SPI1\_NSS

PA5,PB3:SPI1\_SCK

PA6,PB4:SPI1\_MISO

PA7,PB5:SPI1\_MOSI

SPI2

PB9,PB12:SPI2\_NSS

PB10,PB13:SPI2\_SCK

PB14,PC2:SPI2\_MISO

PB15,PC3:SPI2\_MOSI

1. Show the register settings needed to configure UART2 to transmit and UART3 to receive at 9600 baud, eight data bits (LSB first), one stop bit and no parity. Enable an interrupt to indicate that the module has accepted data.

Set Baud rate:

Baud rate =

USARTDIV=42M/(16\*9600)=273.4375

DIV\_Fraction=16\*0d0.4375=0d07=0x7

DIV\_Mantissa=mantissa(0d273.4375)=0d273=0x111

Then, USART2\_BRR and USART3\_BRR=0x1117

Set UART2(TX):

1. Enable clock and power output to relative GPIO(PD5) pins and UART modules

RCC->AHB1ENR|=RCC\_AHB1ENR\_GPIODEN;

RCC->APB1ENR|=RCC\_APB1ENR\_USART2EN;

1. Choose Alternate function mode for GPIO as UART(AF7) and change the configuration to appropriate

GPIOD->MODER|=GPIO\_MODER\_MODER5\_1;

GPIOD->OSPEEDR|=GPIO\_OSPEEDER\_OSPEEDR5\_1;

GPIOD->AFR[0]|=7<<20;//Set the AF to AF7(USART1~3);

1. Set the UART module as TX

USART2->CR1|=USART\_CR1\_UE| USART\_CR1\_TE;

Set UART3(RX):

It is the same as UART2 except for USART2->CR1|= USART\_CR1\_UE| USART\_CR1\_RE;

1. Show the register settings needed to configure I2C as a master with 7 data bits (MSB first).

Enable the clock and power output to relevant GPIO pins and I2C module (e.g. I2C1)

RCC->AHB1ENR|=RCC\_AHB1ENR\_GPIOBEN;

RCC->APB1ENR|=RCC\_APB1ENR\_I2C1EN;

Configure the GPIO settings as appropriate (e.g. PB8 & PB9)

GPIOB->MODER|=GPIO\_MODER\_MODER8\_1|GPIO\_MODER\_MODER9\_1;

GPIOB->OSPEEDR|=GPIO\_OSPEEDER\_OSPEEDR8\_1|GPIO\_OSPEEDER\_OSPEEDR9\_1;

GPIOB->OTYPER|=GPIO\_OTYPER\_OT\_8|GPIO\_OTYPER\_OT\_9;

GPIOB->PUPDR|=GPIO\_PUPDR\_PUPDR8\_0|GPIO\_PUPDR\_PUPDR9\_0;

Initialize I2C

Set FREQ bit in CR2;

Set TRISE register according to PCLK and FREQ;

Set CCR bit in CCR register;

Configure relative settings in CR1 register (e.g. perform error checking and enable ACK)

It will also be used during transmitting and receiving data.

Input the information about the slave address in OAR1 register (Set to 7-bit)