A1

(Part a) The baud rate is given as:

$$\text{Baud rate} = 9600\,\text{bps}$$

The oversampling factor is **16x**. Therefore, the sampling frequency of the UART receiver can be calculated as:

$$\text{Sampling Frequency} = \text{Baud rate} \times \text{Oversampling Factor}$$

$$\text{Sampling Frequency} = 9600 \times 16 = 153,600\,\text{Hz}$$

(Part b) Each byte consists of 1 start bit, 1 stop bit. So, the total number of bits per frame is:

$$\text{Total bits} = \text{Start bit} + \text{Data bits} + \text{Stop bit} = 1 + 8 + 1 = 10\ \text{bits}$$

Time to transmit one bit (t_b):

$$t_b = \frac{1}{B} = \frac{1}{9600} \approx 104.17\,\mu\text{s}$$

Time to transmit one byte (t_{byte}):

$$t_{\text{byte}} = \text{Total bits} \times t_b = 10 \times 104.17\,\mu\text{s} = 1041.7\,\mu\text{s} = 1.0417\,\text{ms}$$

(Part c) For this system,

$$\text{Drift per bit} = 0.02 \times 16 = 0.32\,\text{samples per bit}$$

Over the course of 10 bits (1 start bit, 8 data bits, 1 stop bit), the total drift is:

$$\text{Total drift} = 0.32 \times 10 = 3.2\,\text{samples}$$

3-4 incorrect samples could occur due to this clock mismatch. However, since the receiver samples each bit 16 times, the majority of the samples (12-13) will still be correct, even with this drift.

A2

(Part a) If oversampling is disabled, the baud rate divider is calculated as:

$$\text{Baud rate divider} = \frac{4\ \text{MHz}}{115200\ \text{bps}} \approx 34.722$$

The decimal part is `UCBRx`, which is the whole register of `UCAxBRW` = HEX(34) = 0x0022.

For `UCAxBRW`:

$$\texttt{UCAxBRW} = \text{0x0022}$$

As the fractional part is 0.722, then based on the fraction to decimal mapping (S25), for 0.772, `UCBRSx` = 0xBB.

Since oversampling is disabled, then `UCBRFx` = 0x0 and `UCOS16` = 0.

For `UCAxMCTLW`:

$$\texttt{UCAxMCTLW} = \text{0xBB00}$$

With 8 data bits, no parity, lsb first, 2 stop bits, and SMCLK as the clock source, then following is for `UCAxCTLW0`:

$$\texttt{UCAxCTLW0}\ \text{(at operational)} = 0b0000\ 1000\ 1000\ 0000 = \text{0x0880}$$

$$\texttt{UCAxCTLW0}\ \text{(at reset)} = 0b0000\ 1000\ 1000\ 0001 = \text{0x0881}$$

(Part b) When oversampling (with the factor of 16) is enabled, `UCBRFx`, `UCOS16`, and `UCBRx` will be updated based on the overlapping factor:

$$\text{Baud rate divider w/ oversampling} = \frac{4\ \text{MHz}}{115200\ \text{bps} \times 16} \approx 2.170$$

The decimal part is `UCBRx`, which is the whole register of `UCAxBRW` = HEX(2) = 0x0002.

For `UCAxBRW`:

$$\texttt{UCAxBRW} = \text{0x0002}$$

`UCBRSx` value will be unchanged, so `UCBRSx` = 0xBB.

This time the oversampling is enabled, then `UCOS16` = 1, and `UCBRFx` = $[0.17 \times 16]$ = 0x2.

For `UCAxMCTLW`:

$$\texttt{UCAxMCTLW} = \text{0xBB21}$$

The `UCAxCTLW0` register requires no change compared to the case we had no oversampling:

$$\texttt{UCAxCTLW0} \text{ (at operational)} = 0b0000\ 1000\ 1000\ 0000 = \text{0x0880}$$

$$\texttt{UCAxCTLW0} \text{ (at reset)} = 0b0000\ 1000\ 1000\ 0001 = \text{0x0881}$$

A3

(Part a) For the maximum baud rate, set the divider `UCAxBRW` to its minimum value of 1:

$$\text{Baud Rate}_{\text{max}} = \frac{8,000,000}{1} = 8,000,000 \text{ bps}$$

(Part b) When oversampling is enabled, the effective clock frequency is:

$$\text{Clock}_{\text{effective}} = \frac{\text{SMCLK}}{16} = \frac{8,000,000}{16} = 500,000 \text{ Hz}$$

To achieve the minimum baud rate, use the maximum divider value `UCAxBRW` = 65,535:

$$\text{Baud Rate}_{\text{min}} = \frac{500,000}{65,535} \approx 7.629 \text{ bps}$$

A4

(Part a) The baud rate is typically set using the formula:

$$\text{Baud Rate} = \frac{\text{System Clock}}{\text{Baud Rate Divider}}$$

First, we find the baud rate divider using the nominal system clock:

$$\text{Baud Rate Divider} = \frac{\text{System Clock}}{\text{Desired Baud Rate}} = \frac{16,000,000\,\text{Hz}}{115,200\,\text{bps}} = 138.8889$$

Since the divider must be an integer, it will be set to 139.

Now, calculate the actual baud rate with the defective clock (15.6 MHz):

$$\text{Actual Baud Rate} = \frac{\text{Actual System Clock}}{\text{Baud Rate Divider}} = \frac{15,600,000\,\text{Hz}}{139} \approx 112230.22\,\text{bps}$$

(Part b)

$$\text{Percentage Difference} = \left| \frac{\text{Sensor Baud Rate} - \text{Microcontroller Baud Rate}}{\text{Sensor Baud Rate}} \right| \times 100\%$$

$$\text{Percentage Difference} = \left| \frac{115,200 - 112,230.22}{115,200} \right| \times 100\% \approx 2.576\%$$

(Part c)

Since the percentage difference (approximately 2.576%) exceeds the acceptable ±2% threshold, reliable communication **cannot** be guaranteed.

(Part d) First, we calculate the bit time for both devices:

– **Sensor's bit time:**

$$T_{\text{sensor}} = \frac{1}{115,200} \approx 8.6806\,\mu\text{s}$$

– **Microcontroller's bit time:**

$$T_{\text{micro}} = \frac{1}{112,230.22} \approx 8.9065\,\mu\text{s}$$

Difference in bit time per bit:

$$\Delta T = T_{\text{micro}} - T_{\text{sensor}} \approx 8.9065\,\mu\text{s} - 8.6806\,\mu\text{s} \approx 0.2259\,\mu\text{s}$$

Half a bit period of the sensor:

$$\frac{T_{\text{sensor}}}{2} = \frac{8.6806\,\mu\text{s}}{2} \approx 4.3403\,\mu\text{s}$$

Number of bits until cumulative error equals half a bit period:

$$\text{Number of bits} = \frac{\frac{T_{\text{sensor}}}{2}}{\Delta T} = \frac{4.3403\,\mu\text{s}}{0.2259\,\mu\text{s}} \approx 19.22 \text{ bits}$$

Since UART frames are typically 10 bits long, the cumulative error would reach half a bit period after approximately 19 bits, which is almost two frames.

A5

(Part a)

Using the formula:

$$f_{SCL} = \frac{f_{PCLK}}{2 \times (PRESCALER + 1) \times (DIVIDER + 1)}$$

Given:

$$f_{SCL} = 400\,\text{kHz} = 400,000\,\text{Hz}, \quad f_{PCLK} = 8\,\text{MHz} = 8,000,000\,\text{Hz}$$

Rearranging:

$$(PRESCALER + 1) \times (DIVIDER + 1) = \frac{f_{PCLK}}{2 \times f_{SCL}} = \frac{8,000,000}{2 \times 400,000} = 10$$

The smallest possible values are:

$$PRESCALER = 0 \quad (\text{so } PRESCALER + 1 = 1), \quad DIVIDER = 9 \quad (\text{so } DIVIDER + 1 = 10)$$

(Part b)

$$f_{SCL} = \frac{8,000,000}{2 \times (0 + 1) \times (9 + 1)} = 400,000\,\text{Hz}$$

Error: 0% (no error).