

الف)

از آنجا که در سؤال اشاره نشده از عملیات با سرعت معمولی استفاده شود یا سرعت مضاعف، ما یکبار محاسبات را برای محاسبات با سرعت معمولی انجام داده ایم و برای سرعت مضاعف تنها به اعداد نهایی بسنده کرده ایم چون روند محاسبات در این دو دقیقا عین هم میباشد.

عملیات با سرعت معمولی:

در حالتی که بیت U2X در ثبات UCSRA برابر 0 باشد، ماژول USART در حالت عملیات با سرعت معمولی کار خواهد کرد. در این حالت داریم:

$$UBRR + 1 = f_{OSC} / (16 * BAUD_RATE)$$

از آنجا که $BAUD_RATE = 300$ و همچنین $f_{OSC} = 1\text{Mhz} = 10^6$ با جایگذاری در رابطه بالا مقدار $UBRR + 1$ برابر 208.33 بدست خواهد آمد که از آنجا مقدار $UBRR$ برابر 207.33 محاسبه می شود که با رند کردن به $UBRR = 207$ میرسیم. این رند کردن باعث خطا می شود چون اگر $UBRR = 207$ را در رابطه بالا قرار دهیم مقدار $BAUD_RATE$ با فرض $f_{OSC} = 10^6$ برابر 300.48 محاسبه خواهد شد. حال از رابطه زیر برای بدست آوردن درصد خطا استفاده می کنیم:

$$\text{Error}[\%] = ((BAUD_RATE_{\text{Closest_Match}} / BAUD_RATE) - 1) * 100$$

$$BAUD_RATE_{\text{Closest_Match}} = 300.48$$

$$BAUD_RATE = 300$$

$$\Rightarrow \text{Error} = 0.2 \%$$

عملیات با سرعت مضاعف:

در این حالت $U2X = 1$ و نیز رابطه به شکل زیر تغییر می کند:

$$UBRR + 1 = f_{OSC} / (8 * BAUD_RATE)$$

این بار $UBRR + 1 = 416.67$ و مقدار $UBRR$ با رند کردن برابر 416 بدست می آید. مطابق مقدار رند شده $UBRR$ مجدداً $BAUD_RATE$ محاسبه می شود که برابر 299.76 بدست می آید.

$$\text{Error}[\%] = ((BAUD_RATE_{\text{Closest_Match}} / BAUD_RATE) - 1) * 100$$

$$BAUD_RATE_{\text{Closest_Match}} = 299.76$$

$$BAUD_RATE = 300$$

$$\Rightarrow \text{Error} = -0.07 \%$$

با توجه به این که این حالت خطای کمتری نسبت به حالت با سرعت معمولی دارد، در پیاده سازی اسمبلی از این حالت استفاده کرده ایم.

علت خطا، این است که در رابطه گفته شده، چه در حالت سرعت معمولی و چه در حالت سرعت مضاعف، صورت کسر بر مخرج بخشپذیر نمی شود و ما به ناچار باید UBRR را رند کنیم. این رند کردن موجب می شود خطایی غیر صفر داشته باشیم که اگر رند کردن به پایین باشد درصد خطا مثبت و اگر رند کردن به بالا باشد درصد خطا منفی به دست می آید.

در حالتی که $f_{osc} = 1.8432\text{Mhz} = 1843200\text{hz}$ این مشکل رخ نمیدهد و با تقسیم صورت بر مخرج به اعدادی صحیح می رسم و نیازی به رند کردن نیست. پس درصد خطا در این حالت برابر 0 خواهد بود. هم در حالت سرعت معمولی و هم در حالت سرعت مضاعف.

ثبات های کنترلی مربوط به USART در زیرروال USART_Init برنامه ریزی شده اند. تصویر کد زیرروال مذکور در پایین آورده شده است:

```
115 USART_Init:
116
117 ;Set operational mode: asynchronous, double mode
118 ;double mode has been selected due to having less error
119 ;than the normal mode
120 ldi r16, (1 << U2X)
121 ;ldi r16, (0 << U2X)
122 out UCSRA, r16
123
124 ;Enable receiver and transmitter
125 ;Set frame format: 9 bit characters, 1 stop bit, enabled, even parity
126 ldi r16, (1 << RXEN) | (1 << TXEN) | (1 << UCSZ2)
127 ;andi r16, ~(1 << UCSZ2)
128 out UCSRB, r16
129
130 ldi r16, ((1 << URSEL) | (1 << UCSZ0) | (1 << UCSZ1) | (1 << UPM1))
131 andi r16, ((~(1 << USBS)) & ~(1 << UPM0)) & ~(1 << UMSEL)))
132 out UCSRC, r16
133
134 ;Set BAUD RATE for double mode
135 ;UBRR = 416 = 256 + 151
136 ldi r16, 1;256
137 out UBRRH, r16
138 ldi r16, 160
139 out UBRRH, r16
140 clr r16
141 ret
```

(ب)

کد اسمبلی در فایل جدا ضمیمه شده است.