Работа программы:

Определяем константу F\_CPU, представляющую частоту процессора. В данном случае она установлена равной 8 МГц.

Далее определяем константу USART\_BAUD, представляющую скорость передачи данных в бодах для связи USART.

После определяем константы TIMER1\_INTERVAL и TIMER2\_INTERVAL, представляющие интервалы для TIMER1 и TIMER2 в миллисекундах.

Потом определяем временный регистр (R16) для общего использования в программе.

Следующие строки определяют константы TIMER1\_STR и TIMER2\_STR.

Далее устанавливаем вектор сброса на main. Выполнение программы начинается с main.

Потом задаем векторы для TIMER1 и TIMER2 с соответствующими лейблами.

После устанавливаем вектор для регистра данных USART пустым с лейблом usart\_tx\_empty\_isr.

Далее инициализируем указатель стека, загружая младший и старший байты RAMEND в TEMP регистр и затем сохраняя их в младших (SPL) и старших (SPH) байтах регистра указателя стека.

Потом инициализируем USART путем вычисления и установки значений регистра скорости передачи данных в бодах, включения передатчика и приемника и установки формата кадра данных.

Далее инициализируем TIMER1 и TIMER2 для сброса таймера в режиме сравнения (CTC) с предварительным значением 8, устанавливаем значение сравнения соответствия для желаемого интервала и включаем TIMER1 и TIMER2 для сравнения соответствия прерывания.

Потом устанавливаем флаг включения глобального прерывания.

После создаем бесконечный цикл, отмеченный меткой main\_loop.

Дальше оформляем процедуру обслуживания прерываний (ISR) для TIMER1 и TIMER2. Она загружает младший байт TIMER1\_STR или TIMER2\_STR во временный регистр, вызывает подпрограмму usart\_send\_string и возвращается из прерывания.

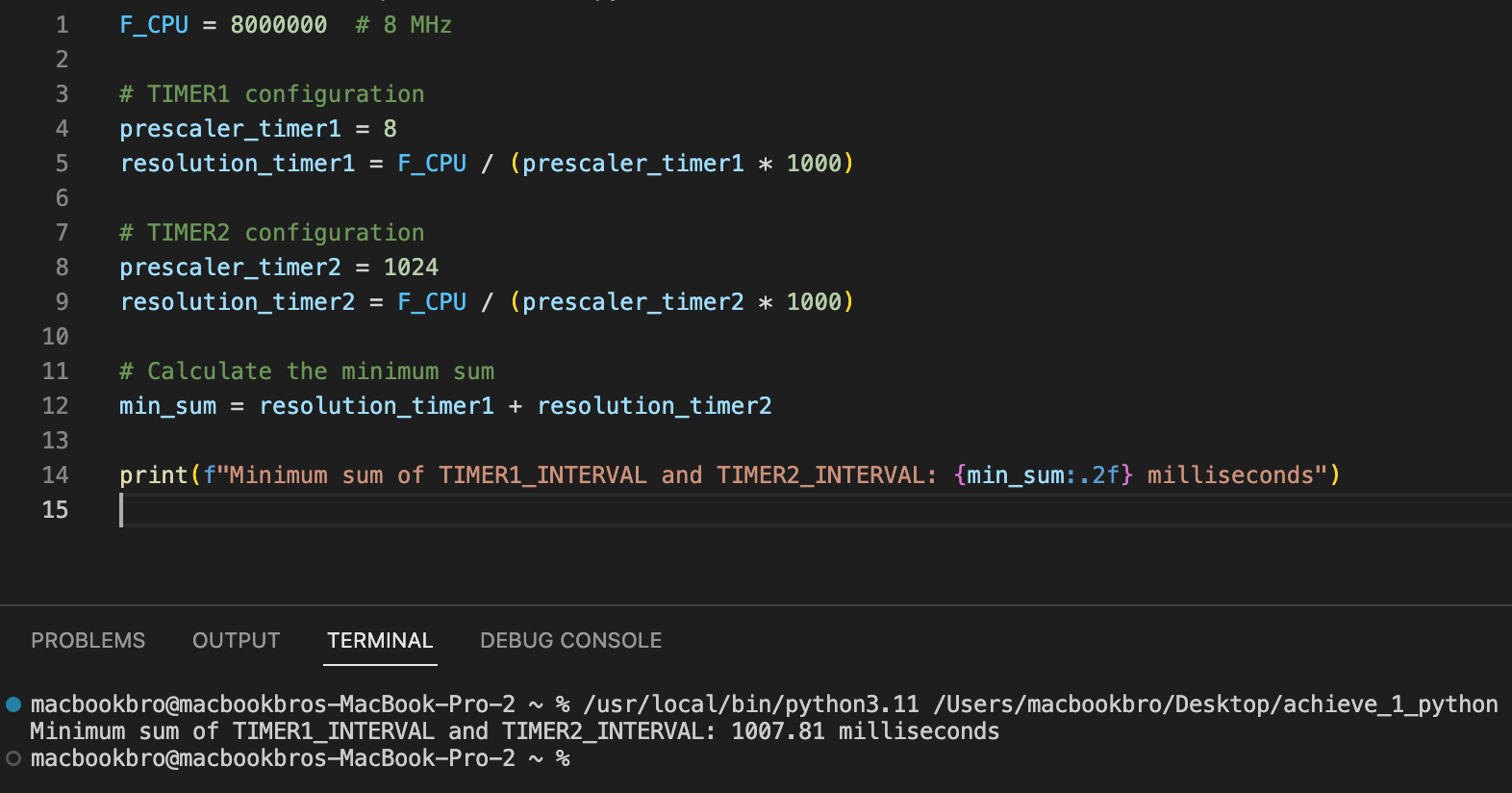
После загружаем адрес строки USART в Z-регистр.

Следующий цикл считывает символы из строки USART, на которую указывает Z, и отправляет их в подпрограмму usart\_send\_char до тех пор, пока не будет обнаружен null terminator.

Дальнейший цикл ожидает установки флага USART Data Register Empty (UDRE), указывающего на то, что регистр готов к приему новых данных.

Последняя часть отправляет символ из регистра TEMP в регистр данных USART (UDR) и возвращает его из подпрограммы.

Находим минимальную сумму значений TIMER1\_INTERVAL и TIMER2\_INTERVAL, при которых программа работает корректно:



Ответ: 1007,81 миллисекунд.