Mikroprocesory i mikrokontrolery Laboratorium nr 5	Temat: Obsługa portu szeregowego
Grupa: 21b	Michał Lechowicz

Cel ćwiczenia:

Celem ćwiczenia było zapoznanie się z działaniem i obsługą **portu szeregowego** w procesorze **Intel 8051**. Port szeregowy otwiera nam możliwości komunikacji ze światem zewnętrznym jako jeden z interfejsów komunikacyjnych. Podobnie jak przy innych możliwościach procesora **Intel 8051,** obsługa portu szeregowego sprowadza się w dużym uproszczeniu do używania kilku rejestrów, a także przerwań.

Zadanie na ocenę 5.0:

1. Opis zadania:

W zadaniu należało napisać program, który wykorzystując system przerwań wpisywałby do szesnastobajtowego bufora cyklicznego, umiejscowionego w pamięci RAM mikrokontrolera, dane przychodzące przez **UART**.

2. Universal Asynchronous Receiver / Transmitter (UART)

Za pomocą portu szeregowego (wykorzystując standard UART) możemy wysyłać dane z i do kontrolera **ZL2MCS51**. Odbywa się to dwoma parami przewodów **RX** (P3.0) i **TX** (P3.1). Odpowiednio **RX** służy do odbioru, a **TX** do nadawania. Dla ustawionego trybu pracy 1 komunikacja odbywa się w trybie asynchronicznym. Standard UART zapewnia nam komunikację dwukierunkową po dwóch osobnych przewodach.

3. Kod programu:

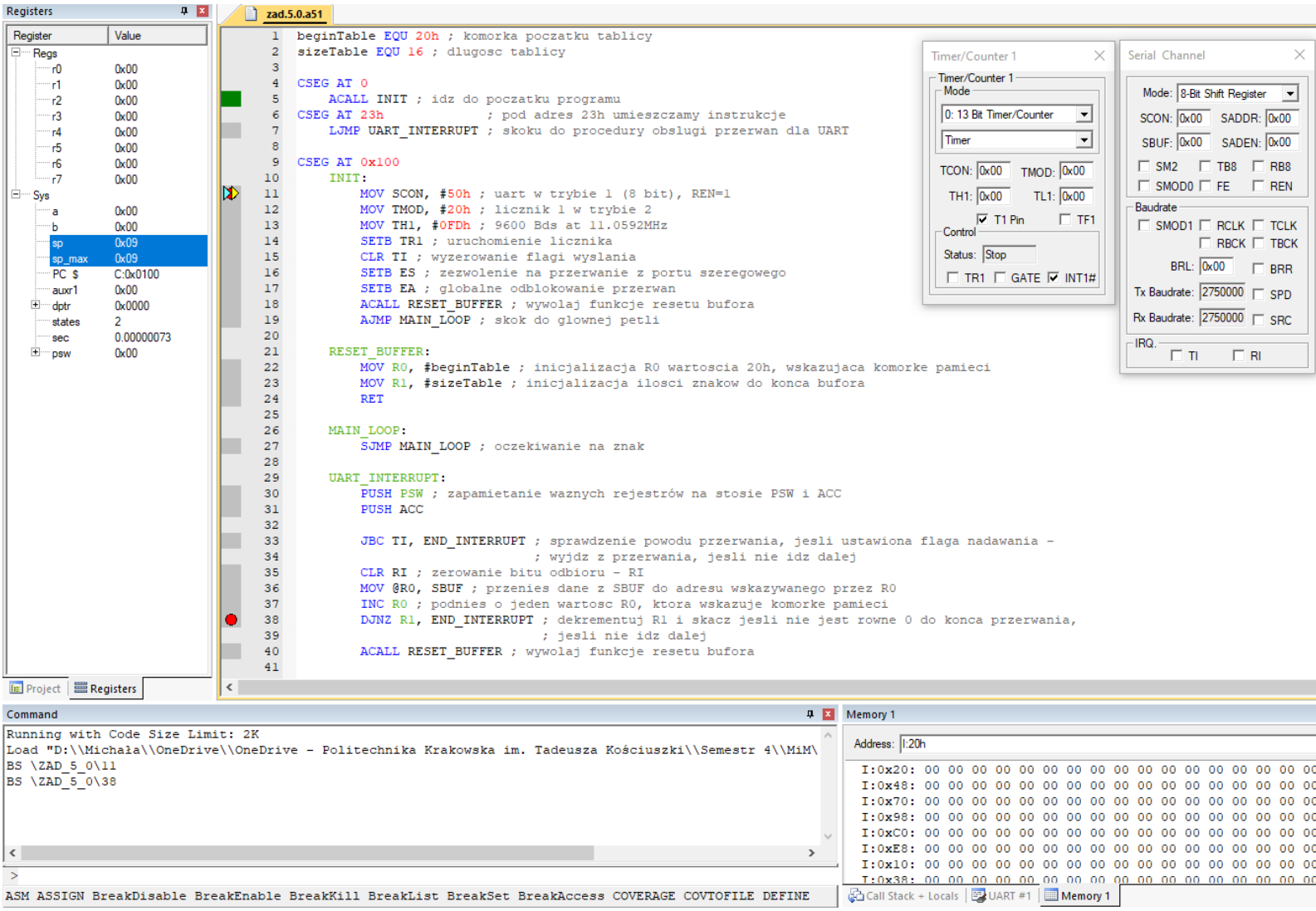
```
beginTable EQU 20h ; komorka poczatku tablicy
sizeTable EQU 16 ; dlugosc tablicy

CSEG AT 0
        ACALL INIT ; idz do poczatku programu
CSEG AT 23h         ; pod adres 23h umieszczamy instrukcje
        LJMP UART_INTERRUPT       ; skoku do procedury obslugi przerwan dla UART

CSEG AT 0x100
        INIT:
                MOV SCON, #50h ; uart w trybie 1 (8 bit), REN=1
                MOV TMOD, #20h ; licznik 1 w trybie 2
                MOV TH1, #0FDh ; 9600 Bds at 11.0592MHz
                SETB TR1 ; uruchomienie licznika
                CLR TI ; wyzerowanie flagi wyslania
                SETB ES ; zezwolenie na przerwanie z portu szeregowego
                SETB EA ; globalne odblokowanie przerwan
                ACALL RESET_BUFFER ; wywolaj funkcje resetu bufora
                AJMP MAIN_LOOP ; skok do glownej petli

        RESET_BUFFER:
                MOV R0, #beginTable ; inicjalizacja R0 wartoscia 20h, wskazujaca komorke pamieci
                MOV R1, #sizeTable ; inicjalizacja ilosci znakow do konca bufora
                RET

        MAIN_LOOP:
                SJMP MAIN_LOOP ; oczekiwanie na znak
```

```
        UART_INTERRUPT:
                PUSH PSW ; zapamietanie waznych rejestrów na stosie PSW i ACC
                PUSH ACC

                JBC TI, END_INTERRUPT ; sprawdzenie powodu przerwania, jesli ustawiona flaga nadawania
                                                                ; wyjdz z przerwania, jesli nie idz dalej
                CLR RI ; zerowanie bitu odbioru - RI
                MOV @R0, SBUF ; przenies dane z SBUF do adresu wskazywanego przez R0
                INC R0 ; podnies o jeden wartosc R0, ktora wskazuje komorke pamieci
                DJNZ R1, END_INTERRUPT ; dekrementuj R1 i skacz jesli nie jest rowne 0 do konca przerwania,
                                                                ; jesli jest rowne idz dalej
                ACALL RESET_BUFFER ; wywolaj funkcje resetu bufora

                END_INTERRUPT: ; wyslij, ze stosu, do ACC i PSW wartosci sprzed przerwania
                POP ACC
                POP PSW
                RETI
END
```

Kod programu

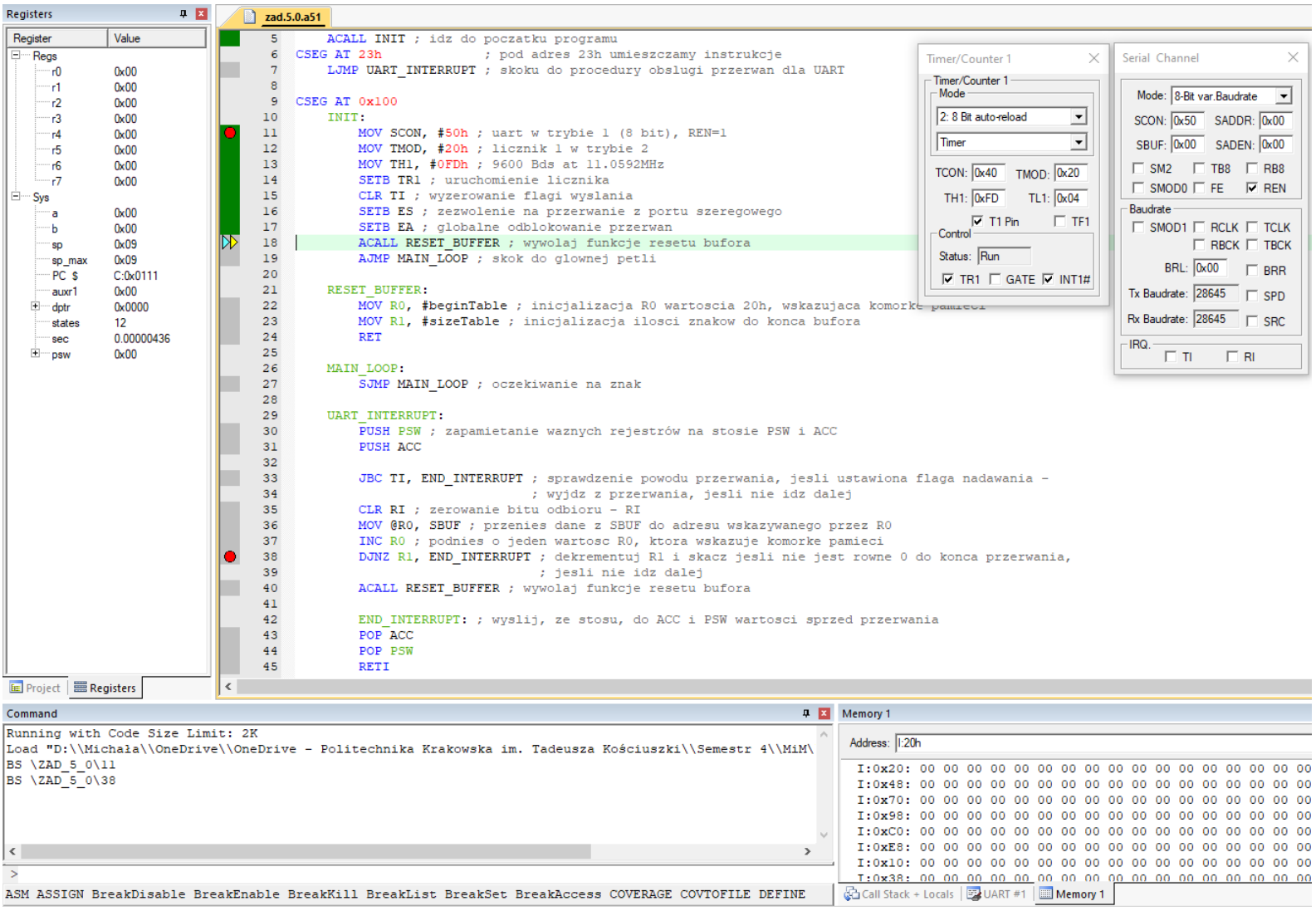
4. Opis programu:

1. Ustawienie obsługi przerwania pod adresem 23h;



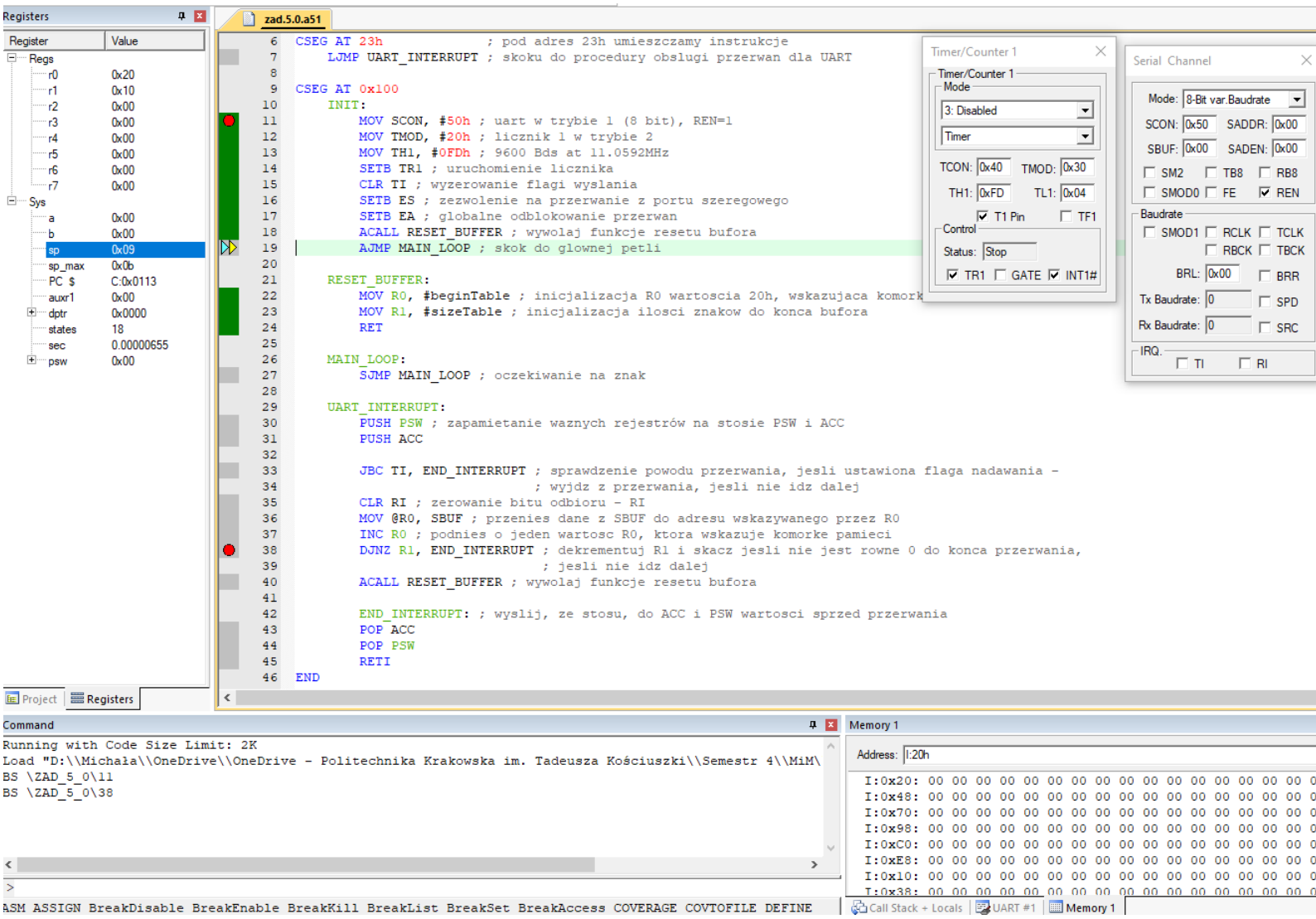
Rys. 1 Ustawienie procedury przerwania

2. Inicjalizacja rejestru SCON (dla UART), wektora przerwań (linie 16 i 17) oraz licznika T1 (linie 12 do 14);



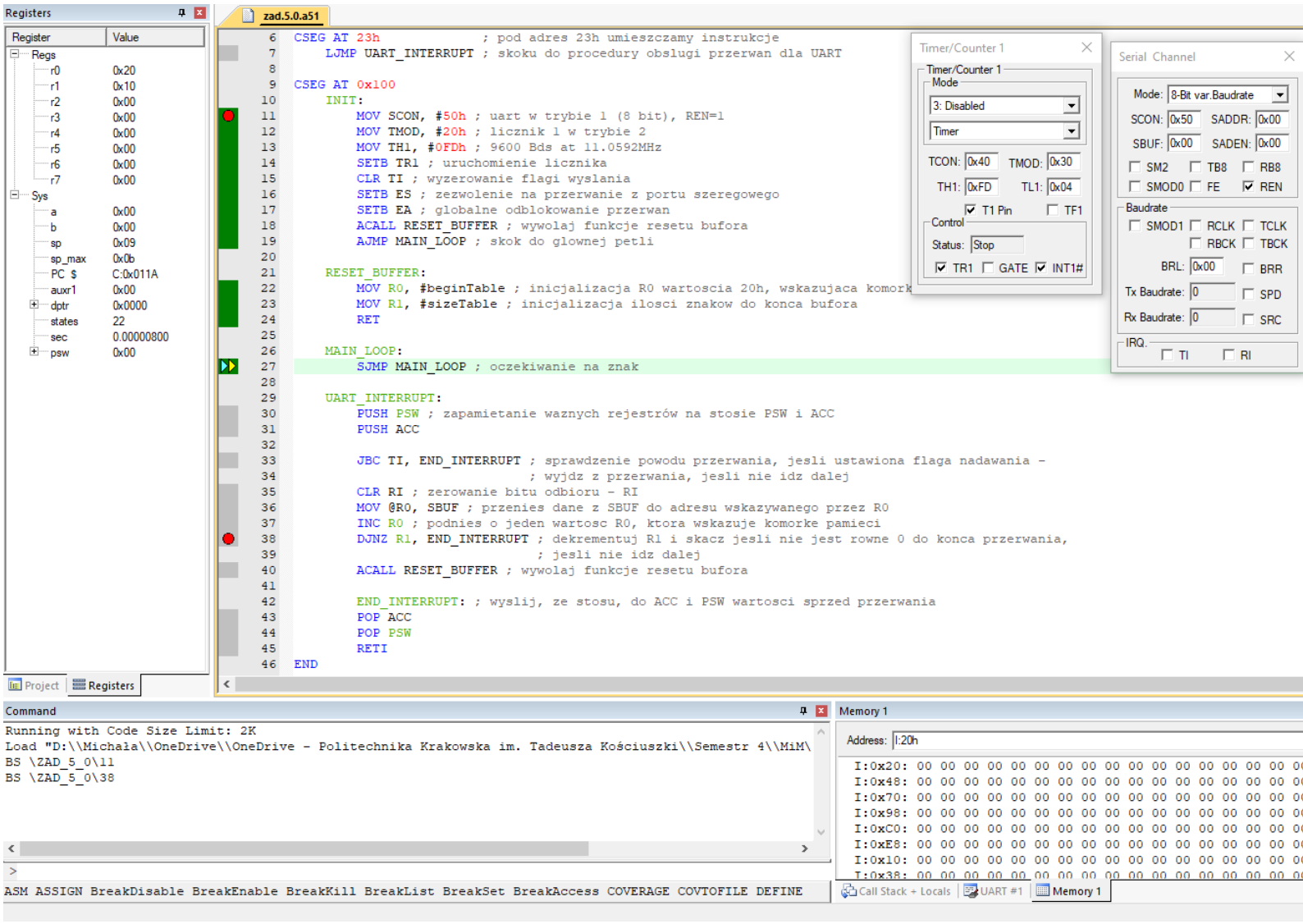
Rys. 2. Inicjalizacja programu

3. Inicjalizacja rejestrów R0 (wskazującego komórkę pamięci, która jest początkiem bufora cyklicznego) i R1 (wskazującego ilość znaków do przepełnienia się bufora);



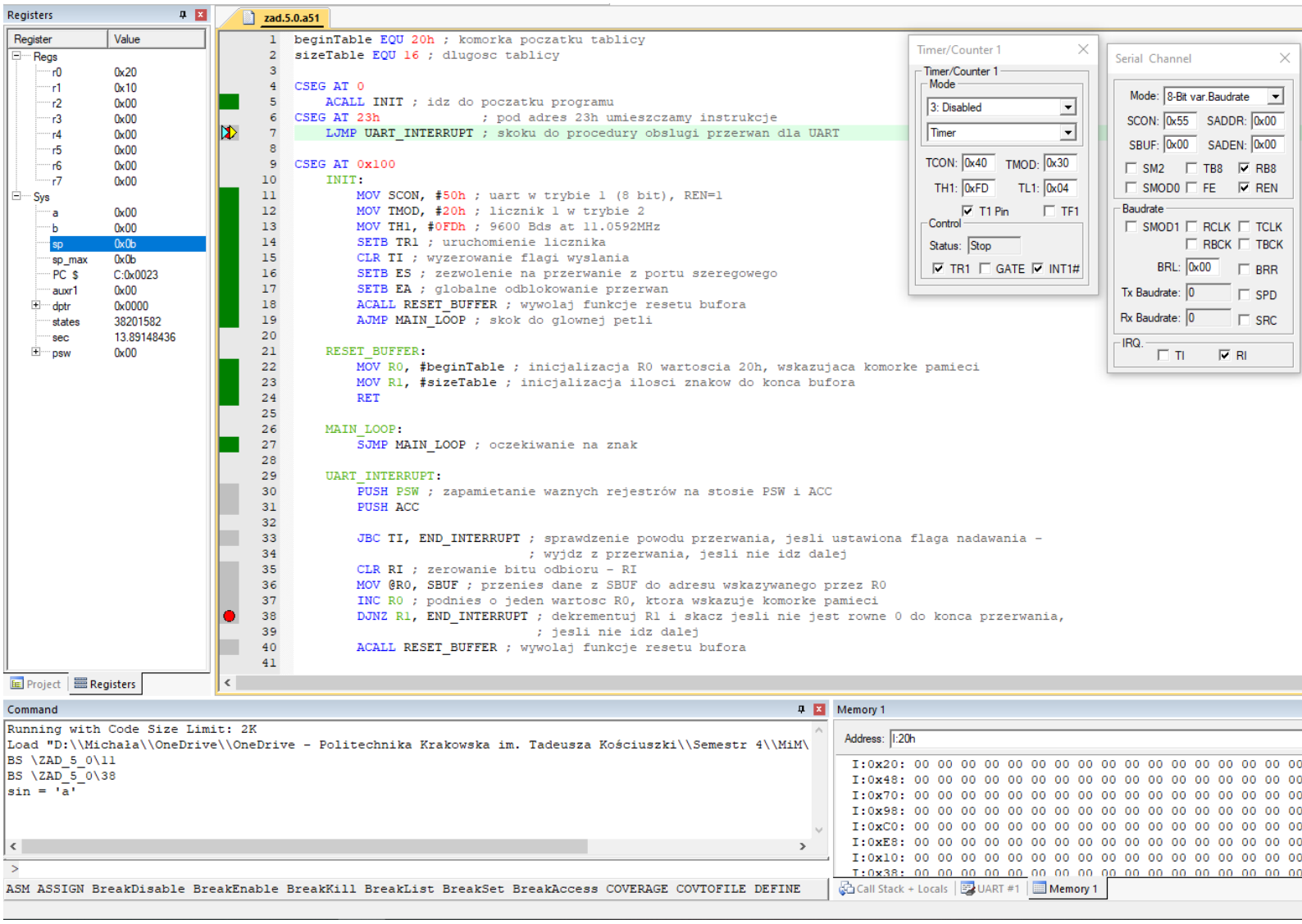
Rys. 3. Ustawienie rejestrów

4. Następnie program wchodzi do głównej pętli, gdzie oczekuje na przerwanie ze strony portu szeregowego (UART);



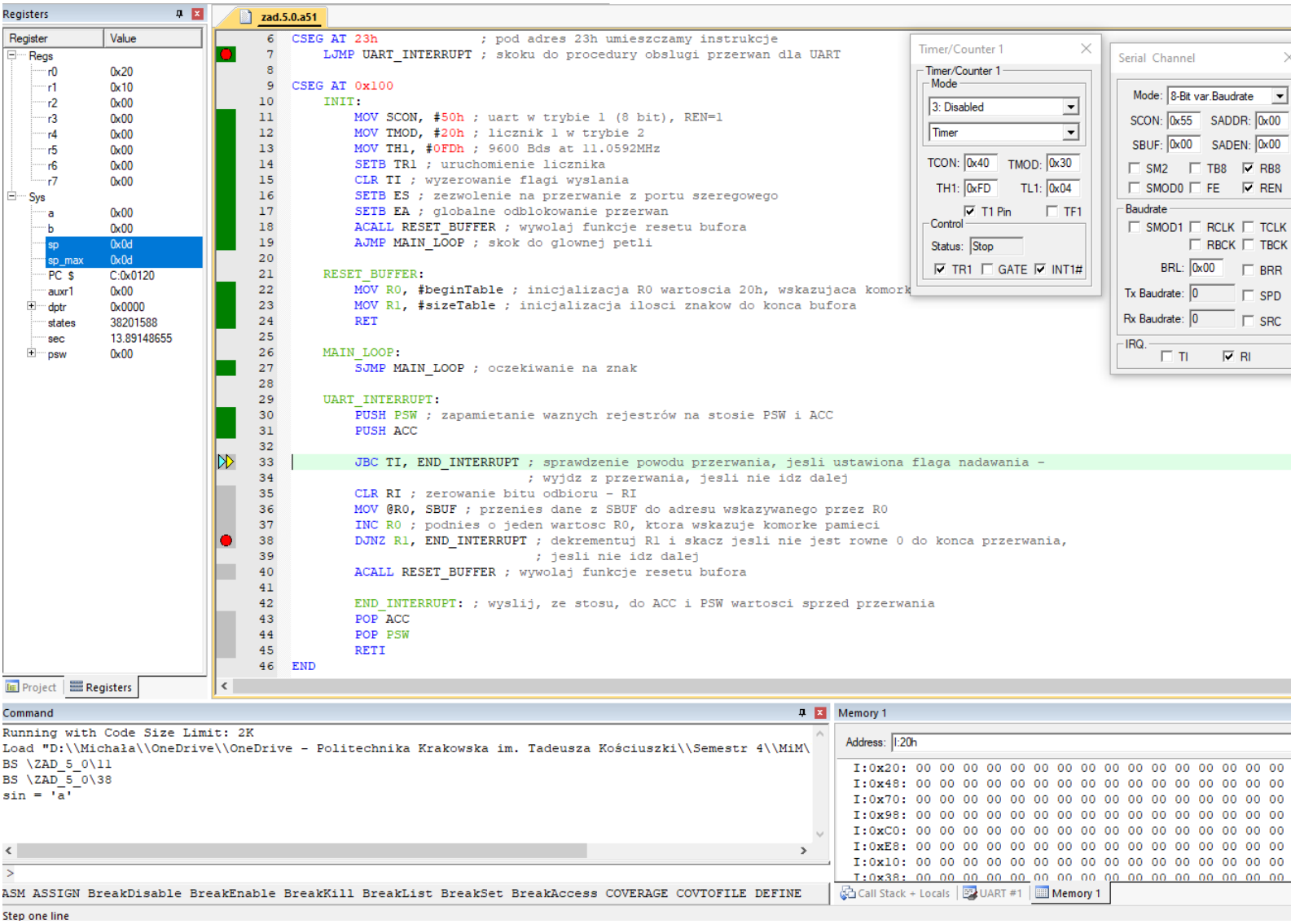
Rys. 4. Oczekiwanie na znak od użytkownika

5. Gdy użytkownik wyśle znak przez UART, następuje aktywacja flagi RI oraz wejście do obsługi przerwania;



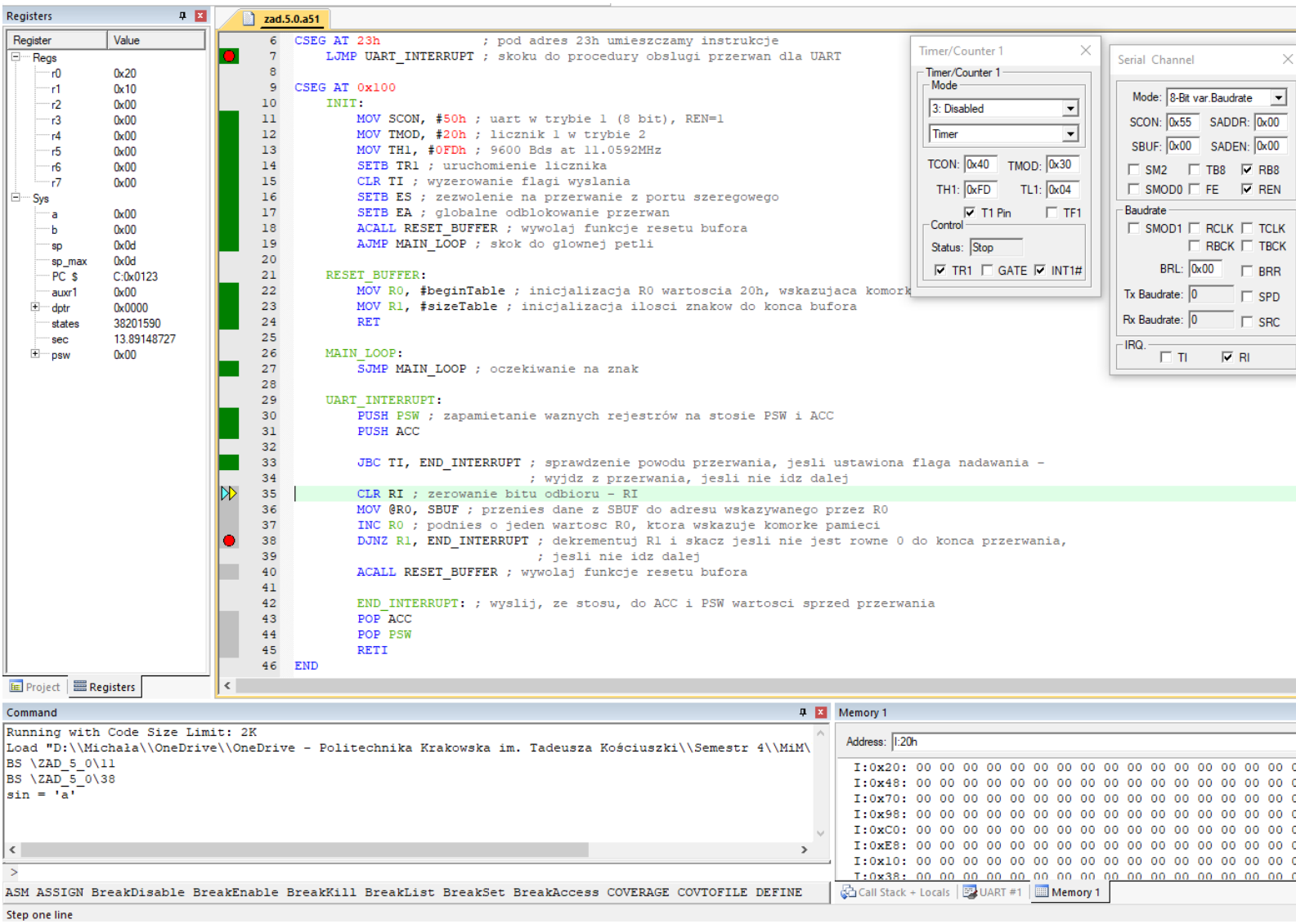
Rys. 5. Wejście do obsługi przerwania

6. W obsłudze przerwania, rejestr PSW oraz Akumulator są przenoszone na stos;



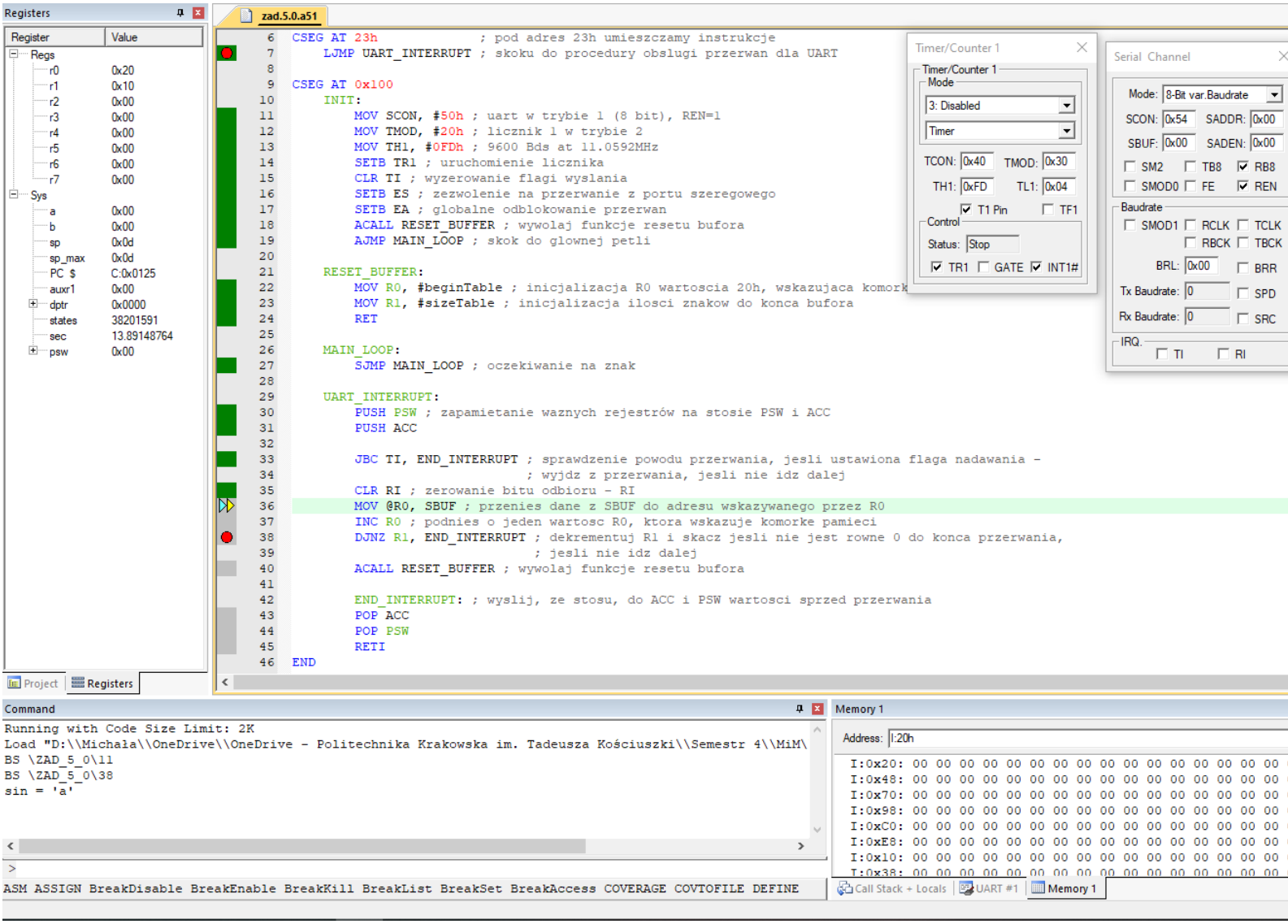
Rys. 6. Zapamiętanie rejestrów PSW i ACC na stosie

7. Następnie, jeśli przerwanie wystąpiło z powodu aktywacji flagi TI, program idzie do procedury zakończenia przerwania, a jeśli nie, idzie dalej;



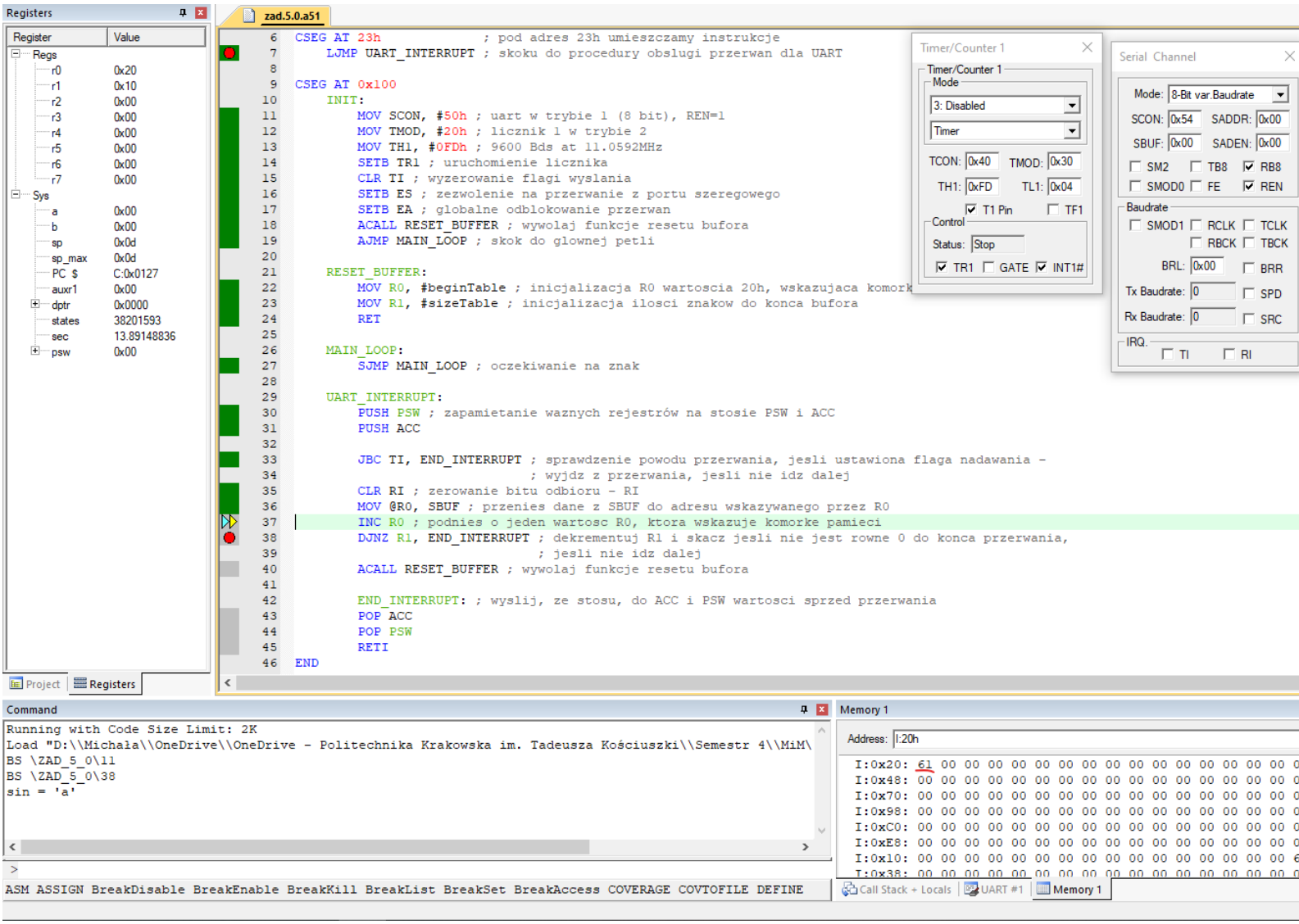
Rys. 7. Sprawdzenie powodu przerwania

8. Następuję zerowanie bitu odbioru RI;



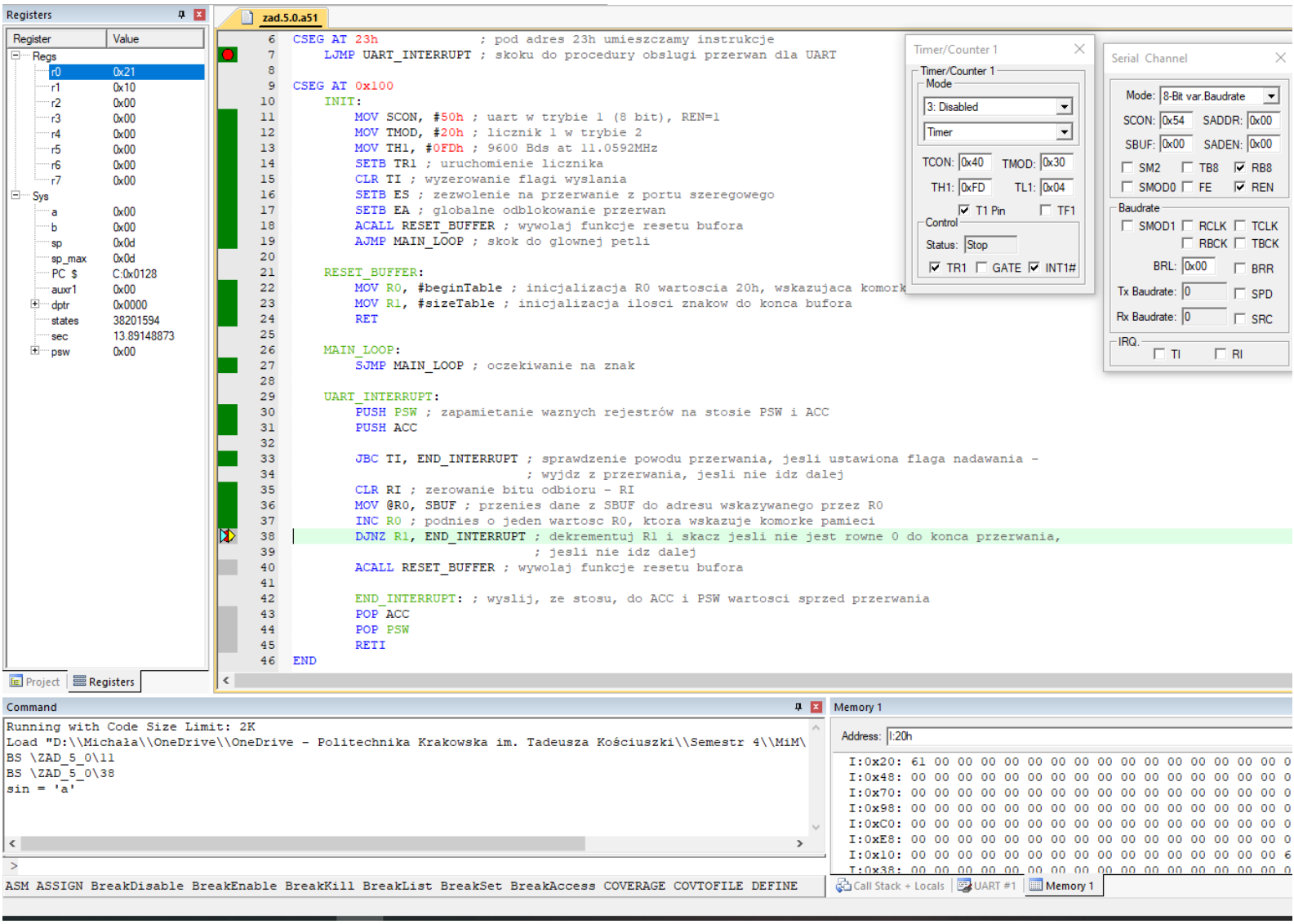
Rys. 8. Zerowanie bitu RI

9. Dalej następuje przenoszenie danych z rejestru SBUF, czyli znaku wprowadzonego przez użytkownika, do adresu wskazywanego przez rejestr R0;



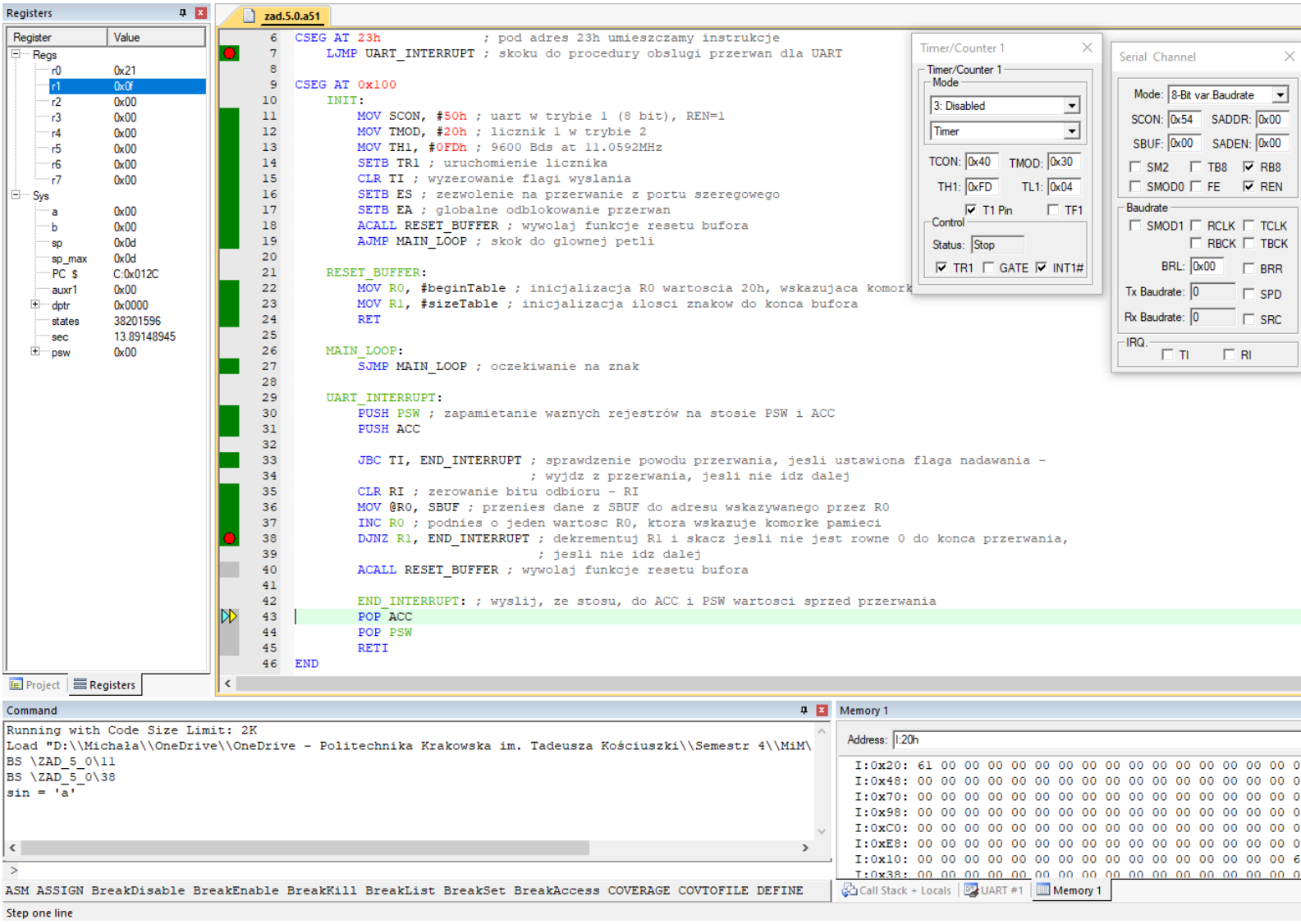
Rys. 9. Przenoszenie znaku z SBUF do pamięci RAM

10. W następnym kroku program inkrementuje R0, by wskazywał kolejną komórkę bufora cyklicznego (w pamięci RAM);



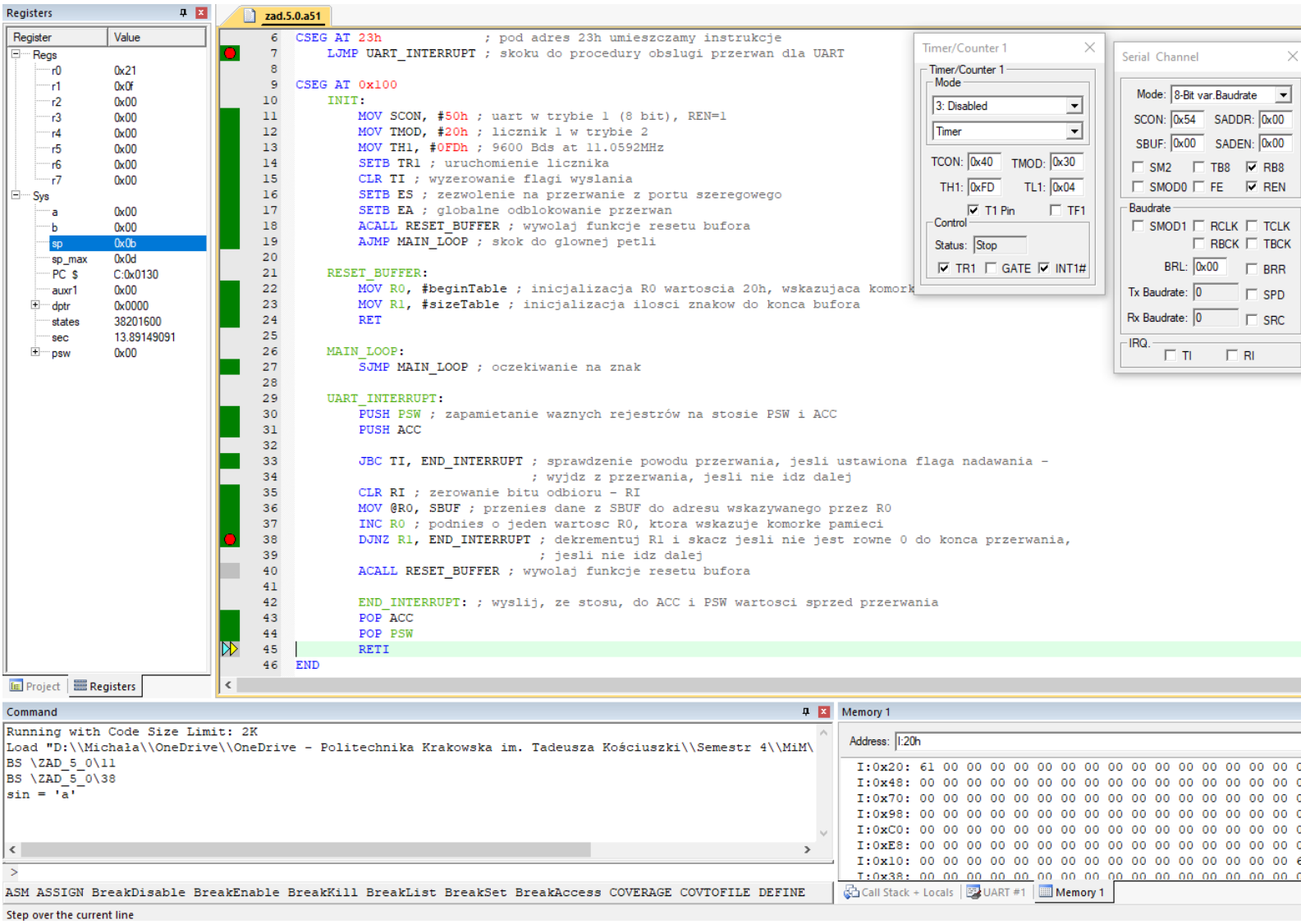
Rys. 10. Inkrementacja R0

11. Dalej algorytm dekrementuje rejestr R1 oraz sprawdza, czy jest równy zero, jeśli tak, oznacza to, że osiągnęliśmy koniec bufora i wywoływana jest funkcja resetująca rejestry R0 i R1 (opis w punkcie 13). Jeśli nie, program idzie do obsługi zakończenia przerwania;



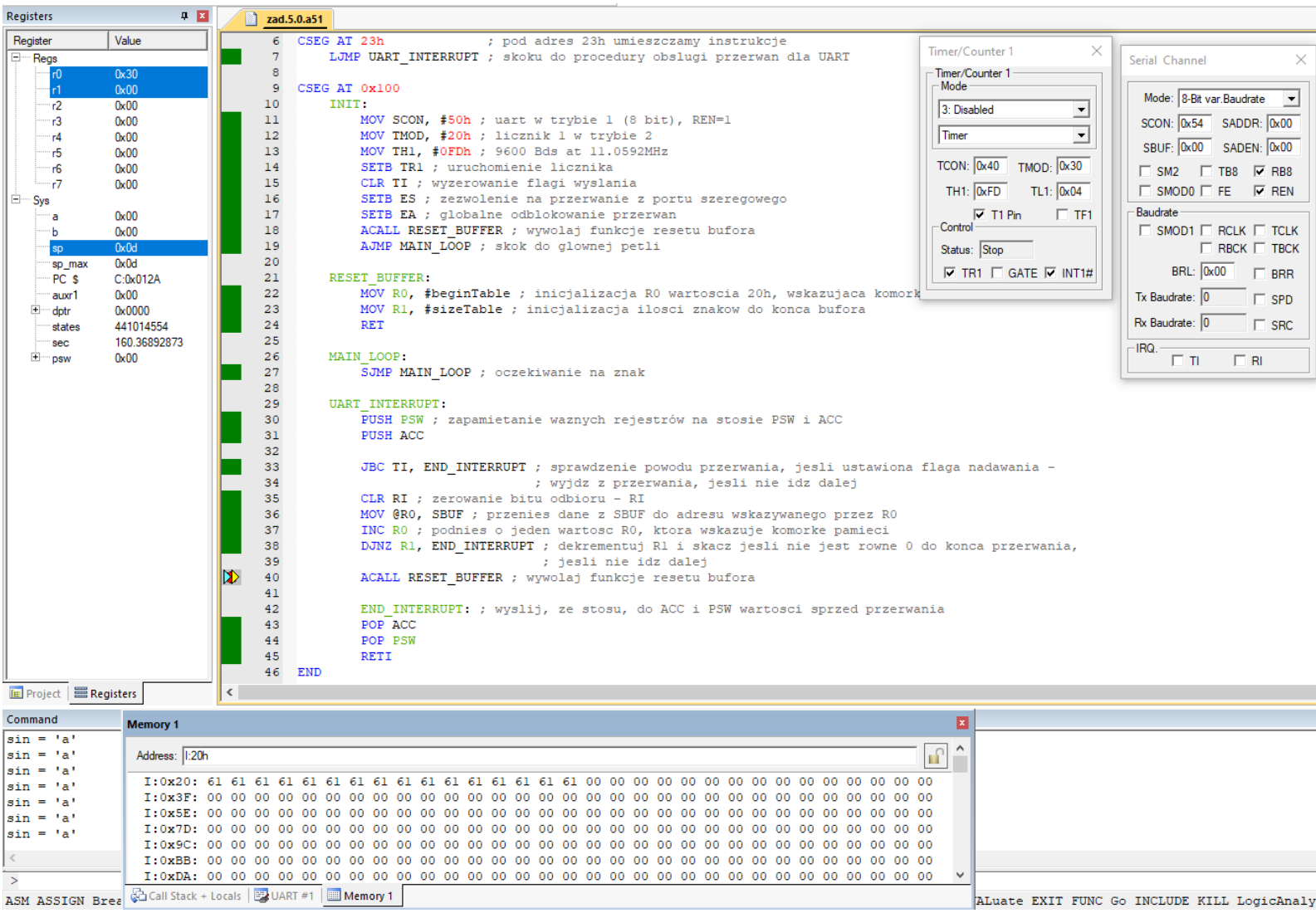
Rys. 11. Sprawdzenie warunku końca bufora cyklicznego z dekrementacją rejestru R1

12. W trakcie procedury zakończenia przerwania, program ściąga ze stosu wartości rejestrów PSW i ACC sprzed rozpoczęcia przerwania oraz wychodzi z niego.

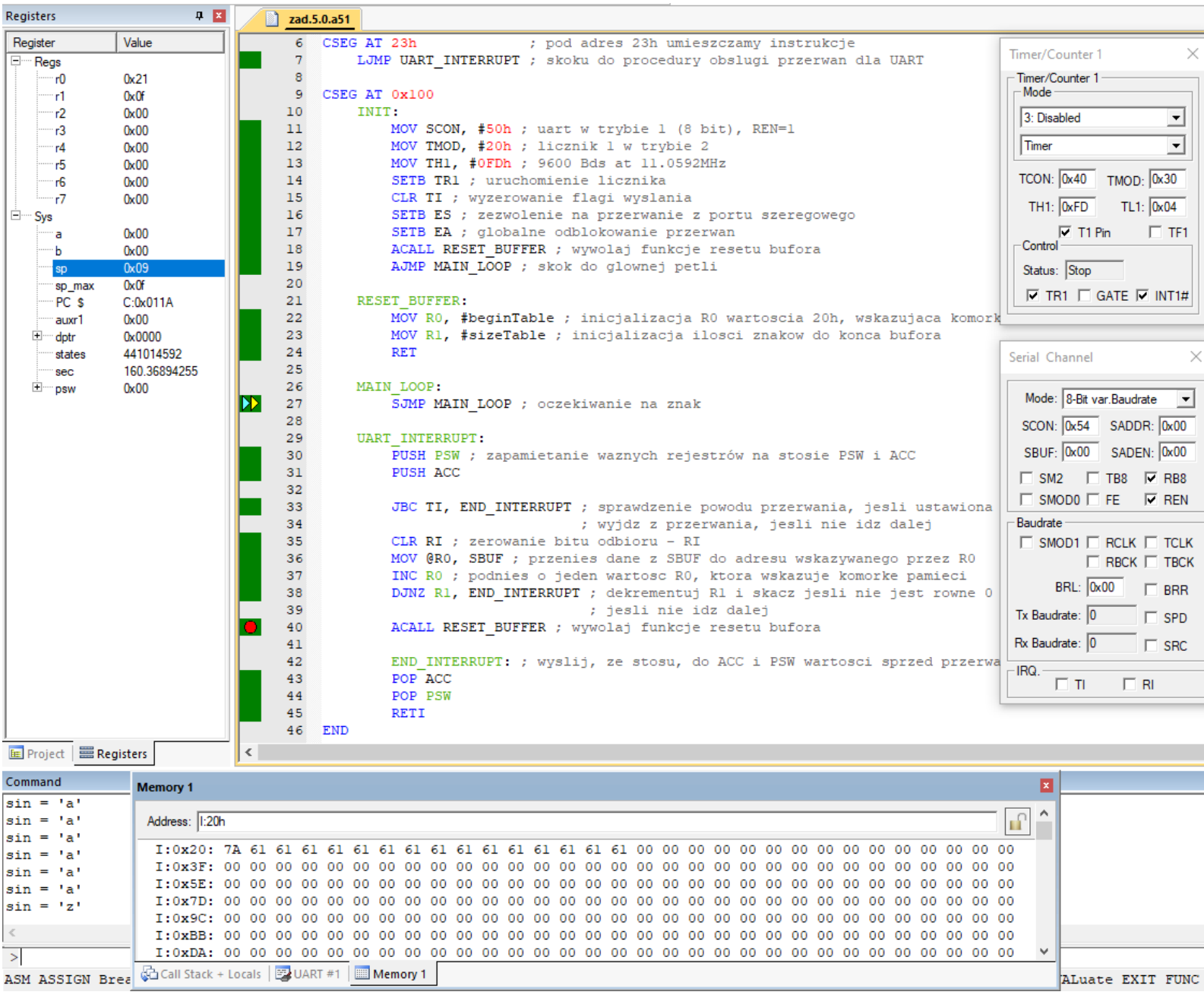


Rys. 12. Przywrócenie wartości rejestrów ACC i PSW ze stosu do rejestrów

13. Jeśli w trakcie działania programu bufor cykliczny zapełni się, wskaźnik na rejestr R0 zostanie zresetowany do wartości domyślnej (początek bufora), a do rejestru R1 zostanie załadowana długość bufora cyklicznego.



Rys. 13. Zapełnienie bufora cyklicznego



Rys. 14. Powrót na początek bufora cyklicznego, po zapełnieniu

4. Podsumowanie i wnioski:

Komunikacja z urządzeniami zewnętrznymi nie jest skomplikowana dla badanego kontrolera. Dzięki wsparciu dla standardu UART, możliwe jest sparowanie mikrokontrolera z szeroką gamą urządzeń peryferyjnych.