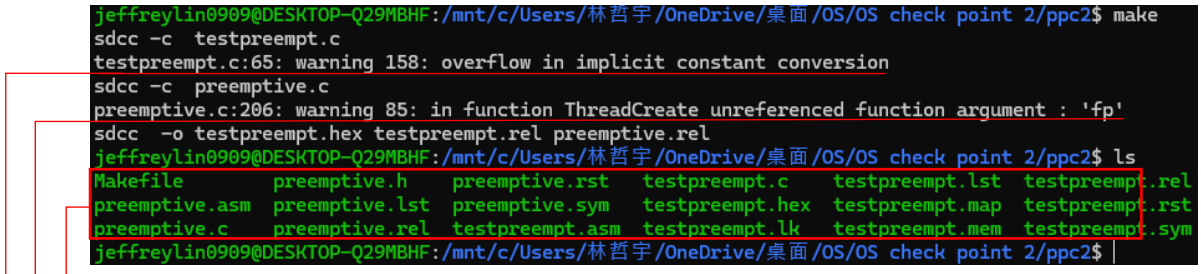
Programming Project Checkpoint 2

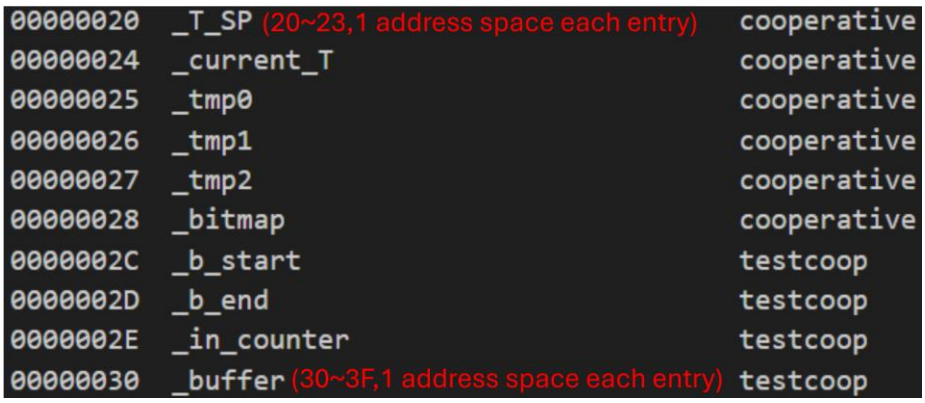
111062311 林哲宇

1. Screenshot for compilation:



- The warning message is because of setting TH1 to negative number (-6 for UART baud rate), and it's totally safe.
- The warning message is because of using DPH/DPL instead of identifier of parameter, and it's totally safe.
- Files generated after compilation (including .hex and .map).

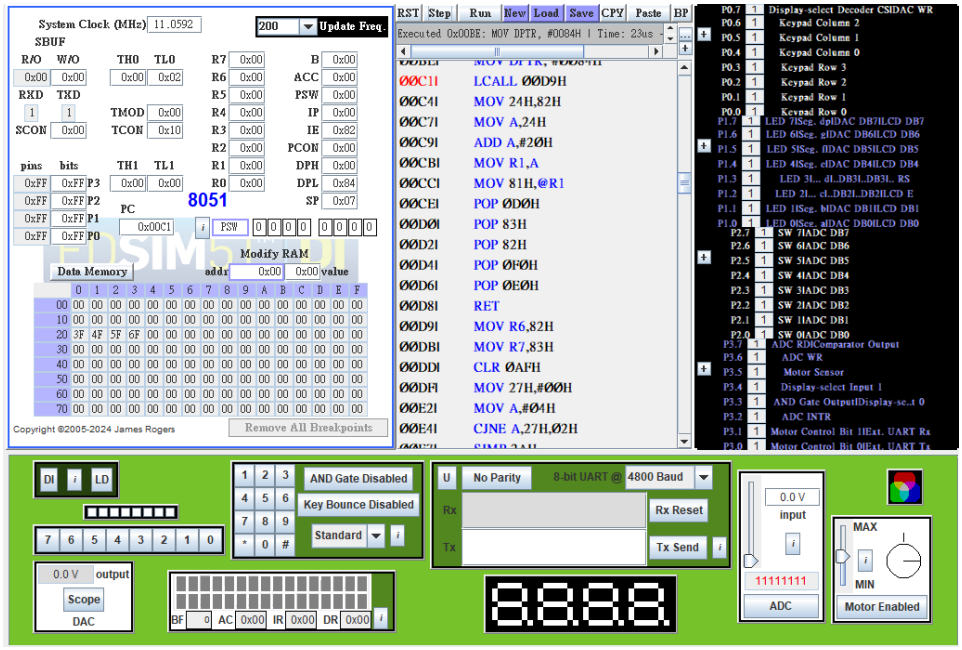
Note: for better understanding for following explanation, here's the variable address map of my code:



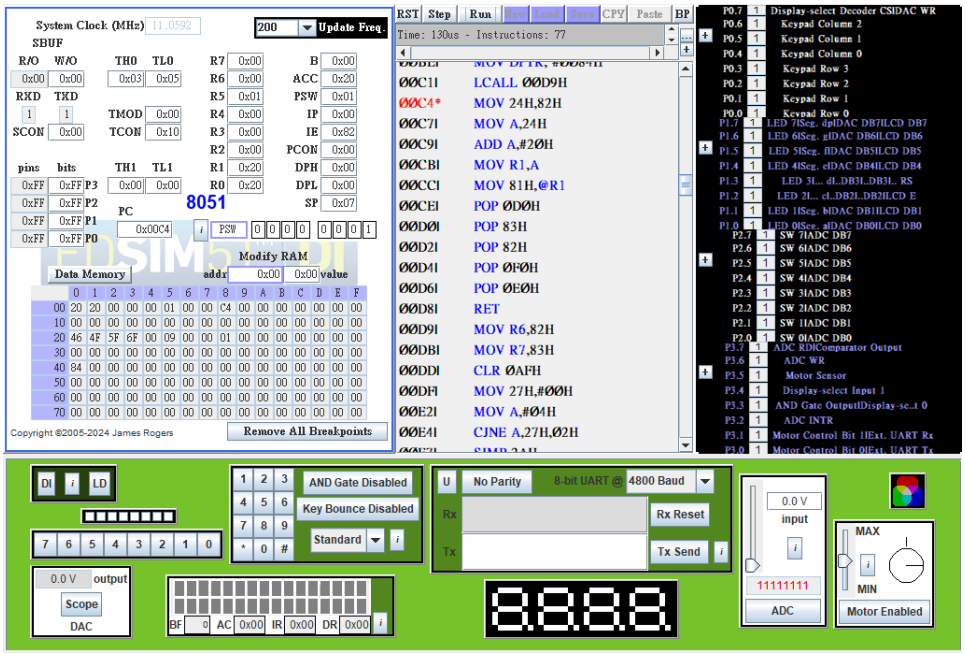
And here's the function address map:

2. Screenshots and explanation:

Before **`ThreadCreate(main)`**:



After **`ThreadCreate(main)`**:

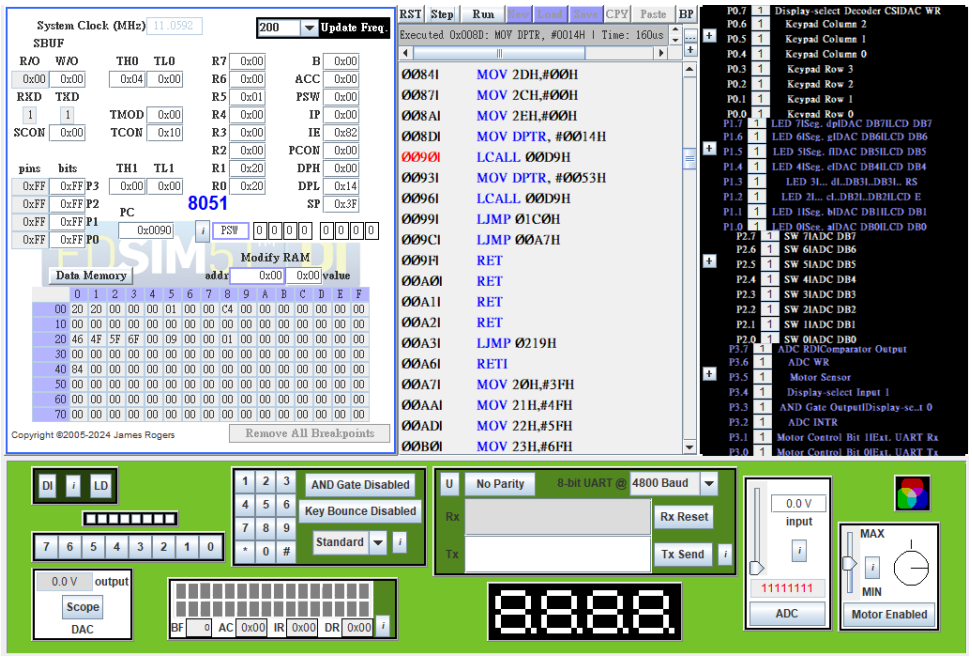


We can see that after **`ThreadCreate(main)`**, we can see:

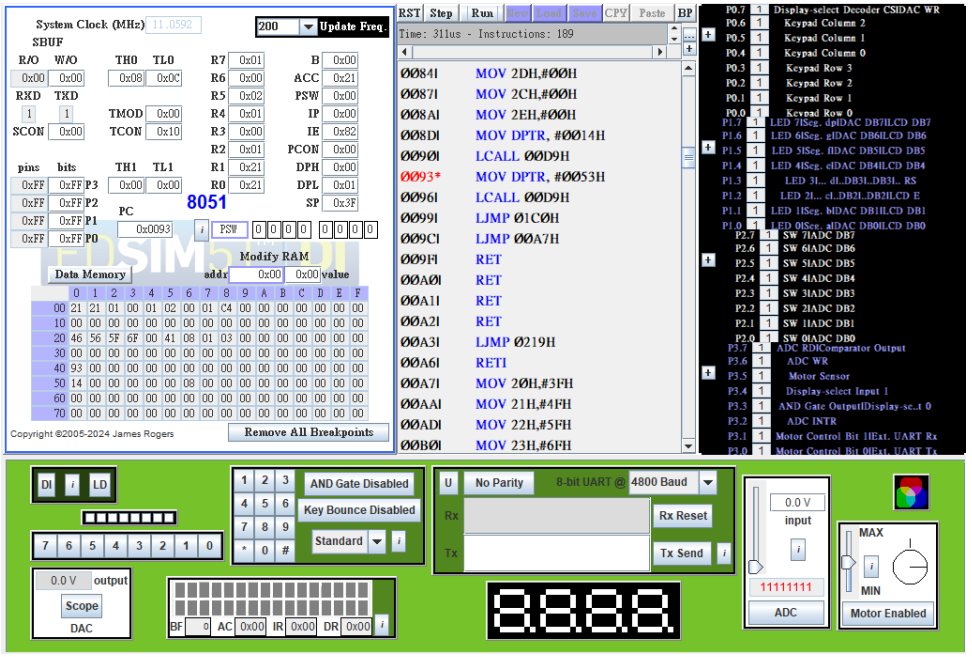
- **SP of thread #0 (address: 0x20) is changed to 46**
- **the bottom of stack of thread #0 (address: 0x40~0x41) is changed to 0084, which is the address of main function**
- **thread bitmap (address: 0x28) is also changed from 0000 to 0001**

meaning main function is assigned to thread #0.

Before **`ThreadCreate(Producer)`**:



After **`ThreadCreate(Producer)`**:

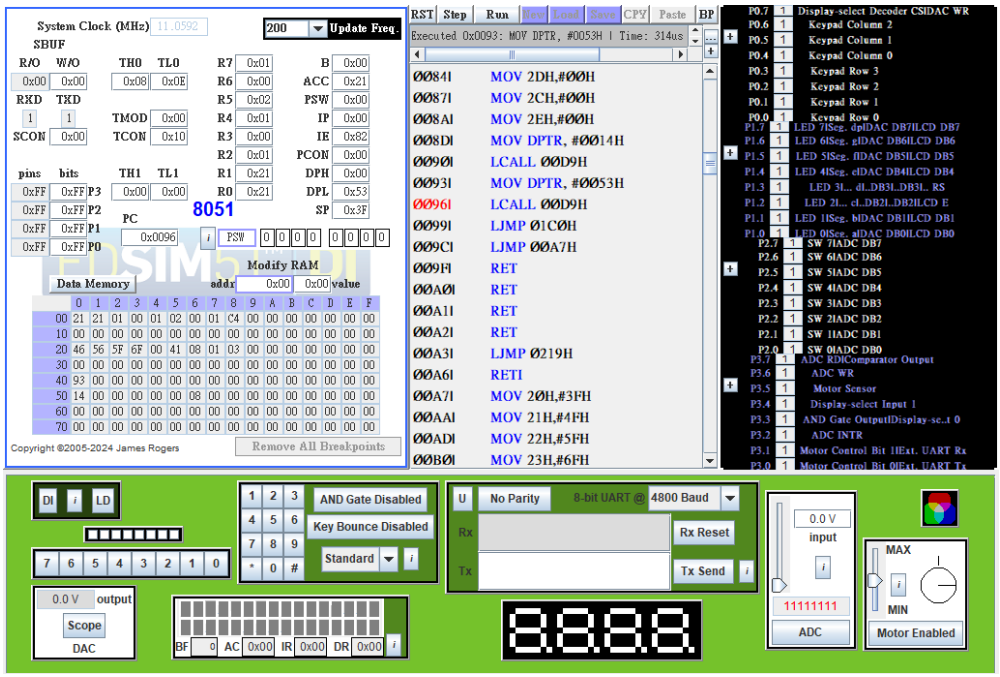


We can see that after **`ThreadCreate(Producer)`**, we can see:

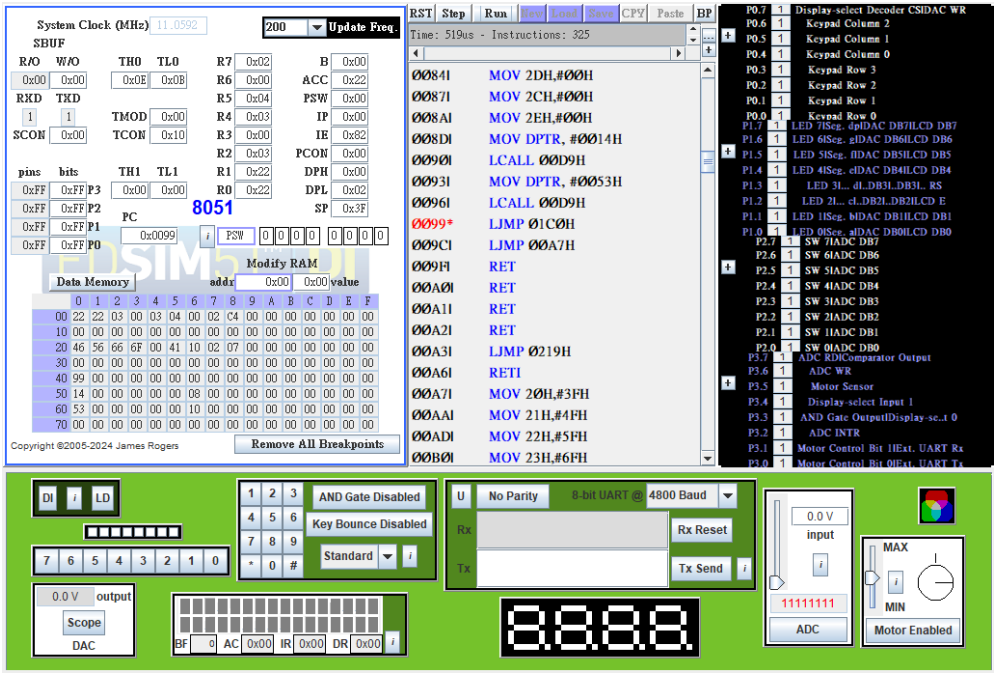
- **SP of thread #1 (address: 0x21) is changed to 56**
- **the bottom of stack of thread #1 (address: 0x50~0x51) is changed to 0014, which is the address of Producer function**
- **thread bitmap (address: 0x28) is also changed from 0001 to 0011**

meaning Producer function is assigned to thread #1.

Before **`ThreadCreate(Consumer)`**:



After **`ThreadCreate(Consumer)`**:

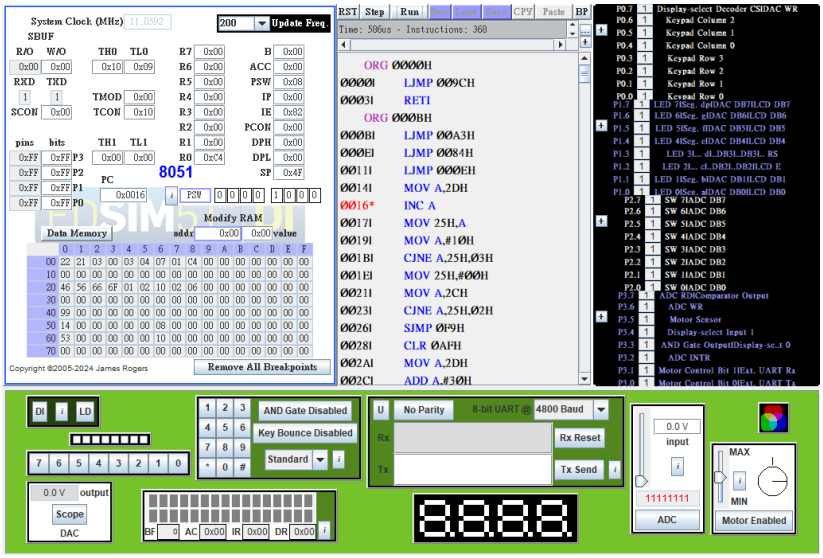


We can see that after **`ThreadCreate(Consumer)`**, we can see:

- **SP of thread #2 (address: 0x22) is changed to 66**
- **the bottom of stack of thread #2 (address: 0x60~61) is changed to 0053, which is the address of Consumer function**
- **thread bitmap (address: 0x28) is also changed from 0011 to 0111**

meaning Consumer function is assigned to thread #2.

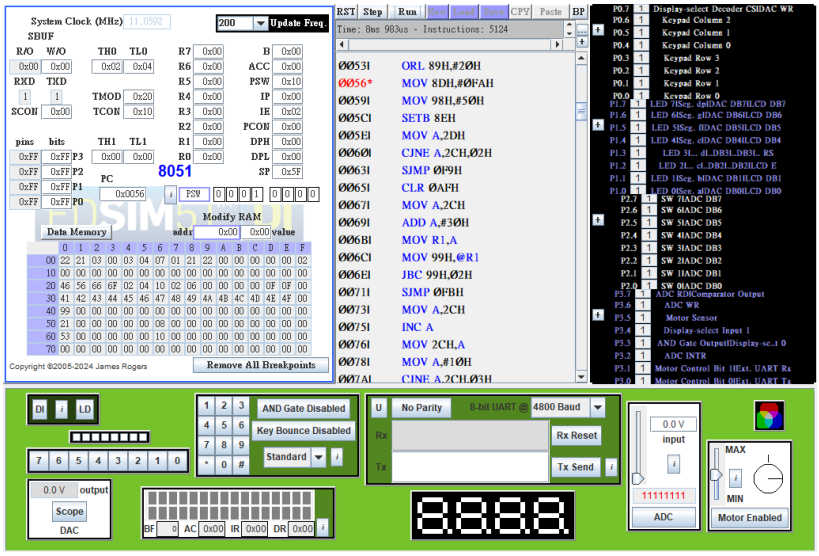
One screenshot when the Producer is running:



How do I know:

- Precise: since PC is in the code section of the function (0x14 ≤PC<0x53).
- Roughly (ignore thread management function): Current thread number (address: 0x24) is 1.

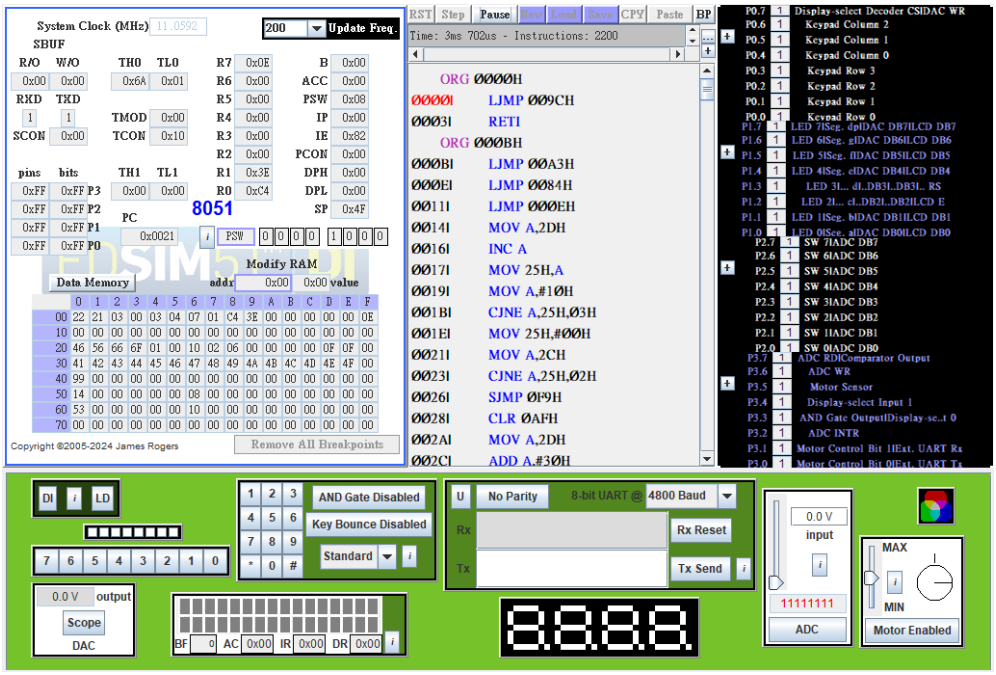
One screenshot when the Consumer is running:



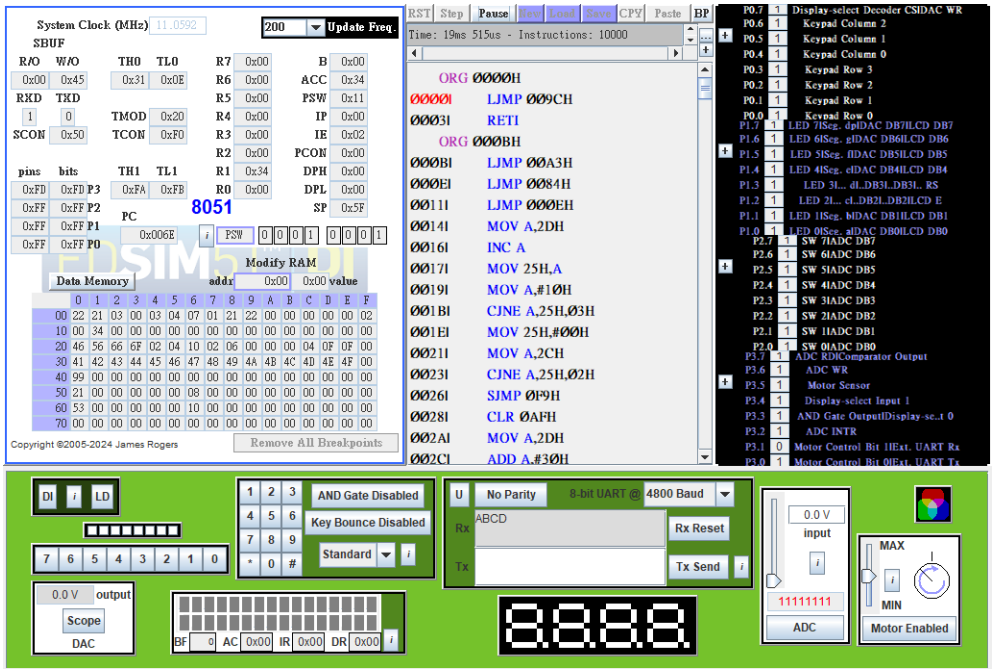
How do I know:

- Precise: since PC is in the code section of the function (0x53 ≤PC<0x84).
- Roughly (ignore thread management function): Current thread number (address: 0x24) is 2.

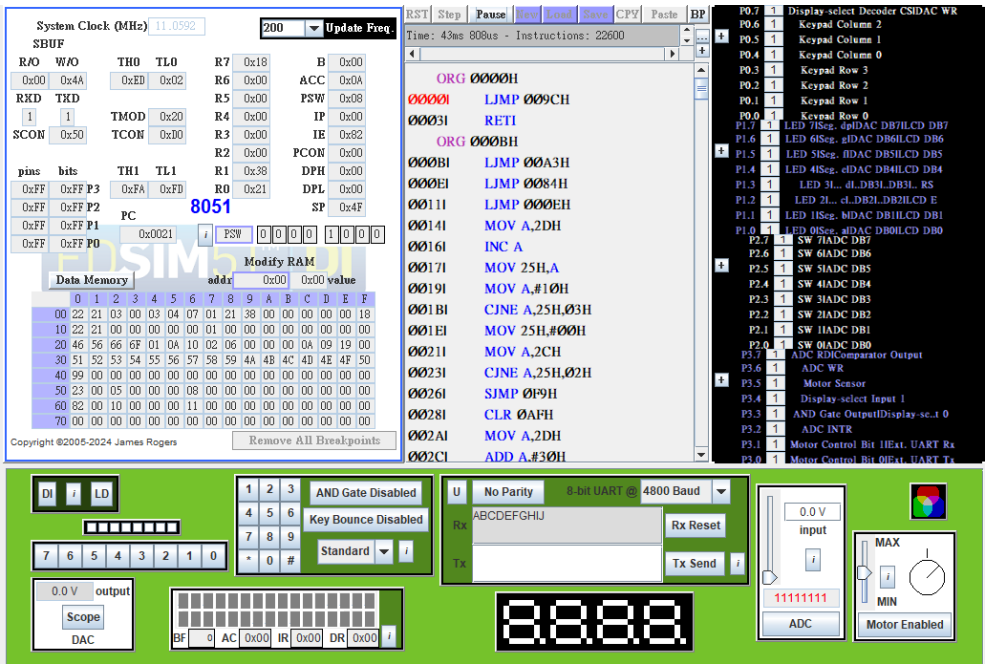
Every time {TH0, TL0} reach 0xFFFF, current thread number (address: 0x24) changes, means that interrupt (to switch current thread) is triggered on a regular basis.



Initially, current thread number is 1 (producer running).



{TH0, TL0} reach 0xFFFF, current thread number switch to 2 (consumer running).



{TH0, TL0} reach 0xFFFF, current thread number switch to 1 (producer running).