

MQX RTCS Memory Configuration Parameters Introduction

RTCS uses some global variables when an application creates it. All the variables have default values. If you want to change the values, the application must do so before it creates RTCS.

There are various configuration parameters that allow to fine tune RTCS memory usage. Next I will introduce the parameters, and I hope it will help beginners in configuration the RTCS correctly

1 _RTCSPCB / _RTCS_msgpool / _RTCS_socket_part

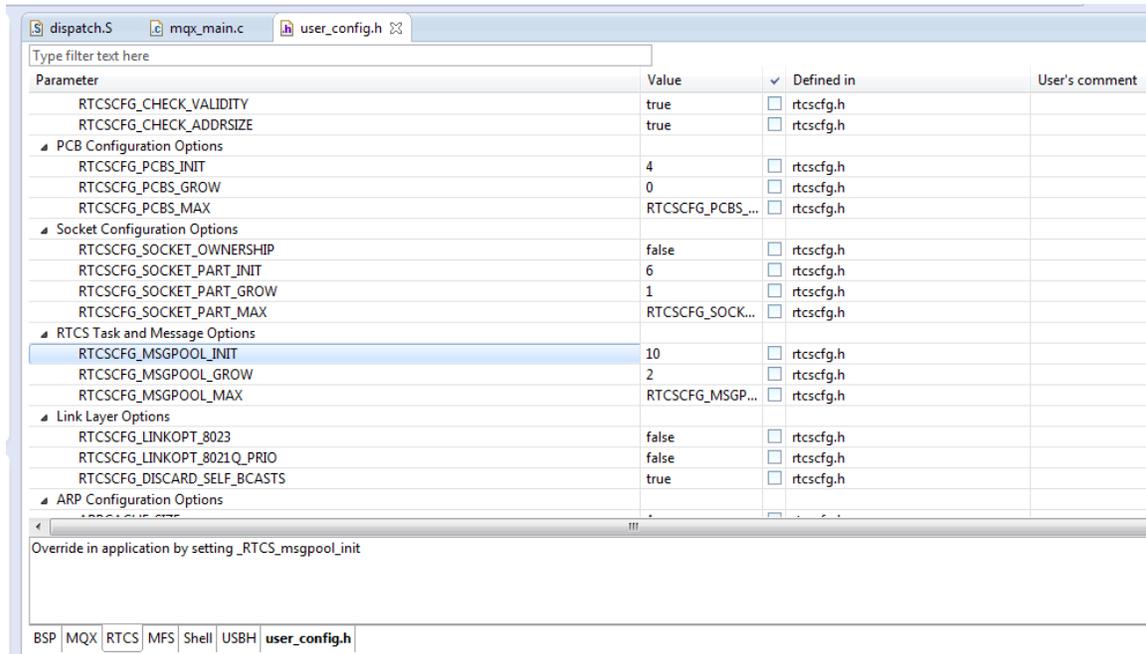
RTCSPCB represents the pool of Packet Control Blocks - the number of packets you want RTCS to handle at any given time. RTCSPCB has influence on losing packets.

RTCS_msgpool represents the pool of RTCS messages, used to encapsulate application requests to RTCS – the number of simultaneous application requests to RTCS.

RTCS_Sockets represents the pool of sockets – the number of sockets in existence at any point in time.

“_init” is pre-allocated count, allocation happens when RTCS starts. During runtime, when RTCS needs more of these resources, it allocates more by “_grow” chunks until the “_max” is consumed, which is the effective limit for RTCS resources. These parameters must be called before _RTCS_start

By default these parameters are defined as RTCSCFG_PCBS_INIT, RTCSCFG_PCBS_GROW, RTCS_PCBS_MAX, RTCSCFG_MSGPOOL_INIT, RTCSCFG_MSGPOOL_GROW, RTCSCFG_PCBS_MAX, RTCSCFG_SOCKET_PART_INIT, RTCSCFG_SOCKET_PART_GROW, RTCSCFG_SOCKET_PART_MAX. In application we can override these defaults by _init, _grow, _max. We also can override these in user_config.h.



How to configure these parameters? We don't have a formal method how to configure these for a specific application. We can start with the defaults and then spent some time increasing these parameters, then we can use RTCS TAD window to see, what RTCS resources are out. Then increase the one a bit.

Example: the configuration in web_hvac demo

```

_RTCSPCB_init = 4;

_RTCSPCB_grow = 2;

_RTCSPCB_max = 20;

_RTCS_msgpool_init = 4;

_RTCS_msgpool_grow = 2;

_RTCS_msgpool_max = 20;

_RTCS_socket_part_init = 4;

_RTCS_socket_part_grow = 2;

_RTCS_socket_part_max = 20;

RTCS_Create();

```

2 Tx Window Size and Rx Window Size

Each socket requires to allocation Tx and Rx window size for a listening socket and then for each connected client. The default tx window size and rx window size are 4380 bytes. With TAD, we can see that every new TCP socket connection needs extra $500+4392 \times 2+148=9432\text{B}$ as follows:

MQX -> Lightweight Memory Block Summary

Size (Decimal) Owner Type

500 0x10001 TCP Control Block;RTCS/TCP

4392 0x10001 TCP Tx Window;RTCS/TCP

4392 0x10001 TCP Rx Window;RTCS/TCP

148 0x10001 TCP Send Clock;RTCS/TCP

(TCP/IP Task id is 0x10001)

If we call `setsockopt()` to reduce these two buffers, for example,

```
uint_32 value;  
...  
value = 1024;  
setsockopt(sock, SOL_TCP, OPT_TBSIZE, &value, sizeof(value));  
setsockopt(sock, SOL_TCP, OPT_RBSIZE, &value, sizeof(value));
```

The memory usage to open a new TCP socket will drop as follows:

MQX -> Lightweight Memory Block Summary

Size (Decimal) Owner Type

500 -> 500 0x10001 TCP Control Block;RTCS/TCP

4392 -> 1108 0x10001 TCP Tx Window;RTCS/TCP

4392 -> 1108 0x10001 TCP Rx Window;RTCS/TCP

148 -> 84 0x10001 TCP Send Clock;RTCS/TCP

So we can see that, every new socket connection needs extra $500+1108 \times 2+84=2800$