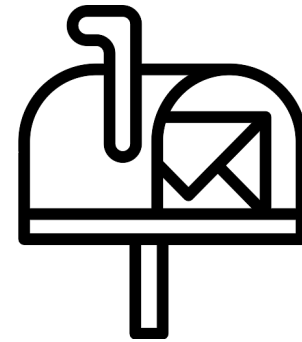


# RTOS: Mailbox example



# Mailbox example for uC/OS-II

```
#include <stdio.h>
#include "includes.h"

/* Definition of Task Stacks */
#define TASK_STACKSIZE 2048
OS_STK task1_stk[TASK_STACKSIZE];
OS_STK task2_stk[TASK_STACKSIZE];
OS_STK task3_stk[TASK_STACKSIZE];

/* Definition of Task Priorities */
#define TASK1_PRIORITY 6
#define TASK2_PRIORITY 7
#define TASK3_PRIORITY 8

//Semaphore to protect jtag uart
OS_EVENT *shared_jtag_sem;

//Message mailbox OS_EVENT structure
OS_EVENT *MSG_box;
```

```
int main(void)
{

printf("=====\n");
printf(" Starting mailbox example\n");
printf("=====\n");

//Create semaphore to protect jtag uart
shared_jtag_sem = OSSemCreate(1);
//Create an empty mailbox
msg_box = OSMBboxCreate((void*)NULL);

//Create the various tasks
OSTaskCreateExt(task1,
                NULL,
                &task1_stk[TASK_STACKSIZE-1],
                TASK1_PRIORITY,
                TASK1_PRIORITY,
                &task1_stk[0],
                TASK_STACKSIZE,
                NULL,
                OS_TASK_OPT_STK_CHK | OS_TASK_OPT_STK_CLR
                );

OSTaskCreateExt(task2,
                NULL,
                &task2_stk[TASK_STACKSIZE-1],
                TASK2_PRIORITY,
                TASK2_PRIORITY,
                &task2_stk[0],
                TASK_STACKSIZE,
                NULL,
                OS_TASK_OPT_STK_CHK | OS_TASK_OPT_STK_CLR
                );

OSTaskCreateExt(task3,
                NULL,
                &task3_stk[TASK_STACKSIZE-1],
                TASK3_PRIORITY,
                TASK3_PRIORITY,
                &task3_stk[0],
                TASK_STACKSIZE,
                NULL,
                OS_TASK_OPT_STK_CHK | OS_TASK_OPT_STK_CLR
                );

//Start multitasking under uc0sii
OSStart();

return 0;
}
```

# Mailbox example for uC/OS-II

```
void task1(void* pdata)
{
    INT8U error_code = OS_NO_ERR;
    int t1;

    while (1)
    {
        t1 = OSTimeGet();
        OSSemPend(shared_jtag_sem,0,&error_code);
        printf("Task1 sending message: %d ms\n",t1);
        OSSemPost(shared_jtag_sem);
        //Post the message with broadcast to all pending tasks
        error_code = OSMboxPostOpt(MSG_box,(void *)&t1,OS_POST_OPT_BROADCAST);
        OSTimeDlyHMSM(0, 0, 1, 0);
    }
}
```

Sending task

```
void task2(void* pdata)
{
    INT8U error_code = OS_NO_ERR;
    int t1;
    int *msg_rx;

    while (1)
    {
        //Pend messages sent from task1
        msg_rx = (int*)OSMboxPend(MSG_box,0,&error_code);
        t1 = OSTimeGet();

        OSSemPend(shared_jtag_sem,0,&error_code);
        printf("Task2 received message: %d ms (at %d ms)\n",*msg_rx ,t1);
        OSSemPost(shared_jtag_sem);
    }
}
```

Receiving task

(Similar for task3)

Running application with broadcast

```
Task1 sending message: 1071 ms
Task2 received message: 1071 ms (at 1076 ms)
Task3 received message: 1071 ms (at 1079 ms)
Task1 sending message: 2076 ms
Task2 received message: 2076 ms (at 2081 ms)
Task3 received message: 2076 ms (at 2083 ms)
```

Running application w/o broadcast

```
Task1 sending message: 1097 ms
Task2 received message: 1097 ms (at 1100 ms)
Task1 sending message: 2100 ms
Task2 received message: 2100 ms (at 2104 ms)
```

```
error_code = OSMboxPost(MSG_box,(void *)&t1);
```

# Mailbox example for uC/OS-II

- OSMboxPend returns a pointer to the message sent through the mailbox
- If that messages is updated before the receiving task has processed the message, the data will be overwritten
- Solution: create a local copy of the received message

```
Task1 sending message: 171 ms
Task2 received message: 171 ms (at 174 ms)
Task1 sending message: 274 ms
Task2 received message: 274 ms (at 277 ms)
Task3 received message: 274 ms (at 176 ms)
```

```
void task3(void* pdata)
{
    INT8U error_code = OS_NO_ERR;
    int t1;
    int *msg_rx;
    int msg_local;
    while (1)
    {
        msg_rx = (int*)OSMboxPend(MSG_box,0,&error_code);
        msg_local = *msg_rx;
        t1 = OSTimeGet();
        usleep(110000);
        OSSemPend(shared_jtag_sem,0,&error_code);
        printf("Task3 received message: %d ms (at %d ms)\n",msg_local ,t1);
        OSSemPost(shared_jtag_sem);
    }
}
```

```
Task1 sending message: 170 ms
Task2 received message: 170 ms (at 173 ms)
Task1 sending message: 273 ms
Task2 received message: 273 ms (at 276 ms)
Task3 received message: 170 ms (at 175 ms)
```