Lab Session - 8

Using Mailbox in uCOS - II

Name : Durvesh Naresh Patil
PRN : 2019BTEEN00035
Batch : EN-1
Subject : RTOS.

- **Programme for mailbox**

```
// programme to illustrate mailbox

#include "config.h"
#include "stdlib.h"
#include <stdio.h>

#define TaskStkLength 64

OS_STK Task0Stk[TaskStkLength];
OS_STK Task1Stk[TaskStkLength];

void Task0(void *pdata);
void Task1(void *pdata);

// create pointer to the mailbox

OS_EVENT *MyMailBox;

// declare variable to store error

uint8 err;
```

```
int main (void)
{
    LED-init();
    TargetInit();
    OsInit();

    // create actual mailbox with no message

    MyMailBox = osMboxCreate((void*)0);

    OSTaskCreate(Task0,(void*)0,&Task0Stk0[
                                TaskStkLength-1],6);
    OSTaskCreate(Task1,(void*)0,&Task1Stk[
                                TaskStkLength-1],7);

    OsStart();
    return 0;
}

// task0

void Task0 (void *pdata)
{
    unsigned int c;
    pdata = pdata;

    while(1)
    {
        c = 3;
        OsMboxPost(MyMailBox,&c); // sending data
        LED-on(0);
        OSTimeDly(1);
        LED-off(0);
        OSTimeDly(20);
    }
}
```

```
// Task1 definition

void Task1 ( void *pdata)
{
    int i;
    unsigned int* ptr;
    unsigned int cl;
    pdata = pdata;

    while (1)
    {
        cl = 0;
        ptr = OSMboxPend (myMailBox, 0, &err);
        cl = *ptr;

        for ( i=0; i< cl; i++)
        {
            LED_on(1);
            OSTimeDly(1);

            LED-off(1);
            OSTime Dly (1);
        }

        OsTime Dly (4);
    }
}
```

* Inter-task Communication *

It is the communication betⁿ two or more tasks.

Every task has its own while loop then why it need intertask communication.

It does its own dedicated work.

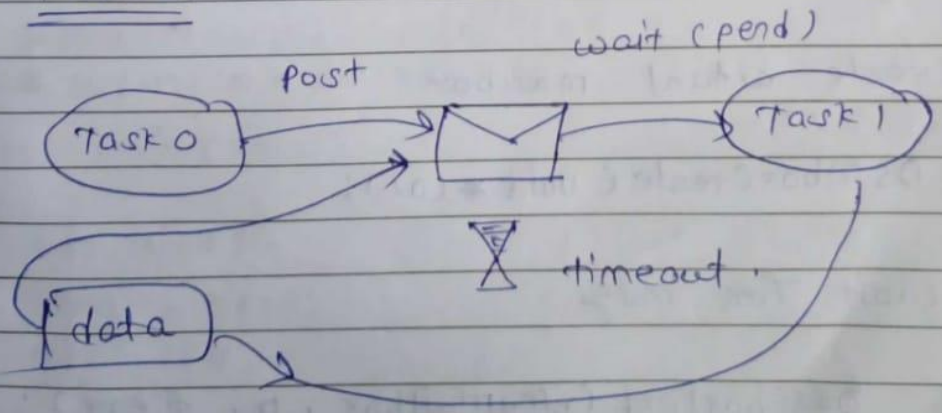
However, we need to pass data from 1 task to another task. i.e. communicate with other task..

for eg. If a task is reading parameter pressure & converting it in digital (ADC) then it may read data, processing, final value has to be shared with other task

Intertask communication can be implemented in two ways :

1. mailbox (mechanism to pass message from one task to other task)
2. Queue

1. Mailbox :



Q. what can be the type of message ?
⇒ Message can be of any type.

Q. what can be amt. of data that can be pass ?
⇒ Any depends on the system.

To implement this, os allows us to pass message in the pointer form, since datatype is also not known we will be using void pointer.

This is the reason why mailbox hold pointer to the data

Steps to implement mailbox

Q How many mailboxes can be in a system
⇒ Any number of mailboxes depending on logic of product

① Create Pointer to a mailbox

```
OS_EVENT * pMyMailbox;
```

② Create actual mailbox

```
pMyMailbox = OSMboxCreate (void *(0));
```

③ wait for msg

```
OSMboxPend (pMyMailbox, 0, &err);
```

0 → infinite wait.
timeout

④ ~~Post message~~
OSMboxPost

③ A7 create data variable

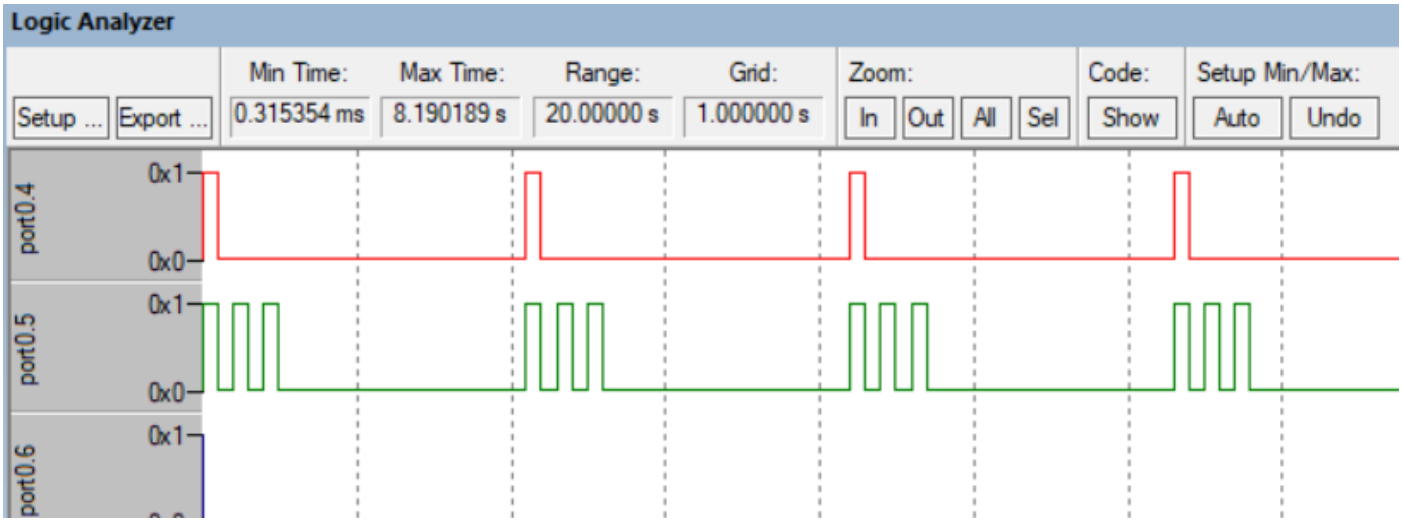
unsigned char pressure;

④ Post message

OSMboxPost (, &pressure);

ptr to which mailbox msg is deposited

Observation:



Conclusion :

(i) using mailbox we can send data from one task to another task.

(ii) We cannot send multiple data through the mailbox

(iii) We can send only one message using the mailbox