

Ένας μικροελεγκτής 8-bit όπως ο PIC 16F877 της MICROCHIP που είναι πολύ διαδεδομένος στα σχολικά εργαστήρια έχει πολύ λιγη ενσωματωμένη μνήμη RAM 96 Bytes καταχωρητές γενικού σκοπού και δεν έχει επίσης στοίβα (stack) για χρήση από συναρτήσεις, αυτό έχει σαν συνέπεια να είναι απαγορευτική η χρήση ενός Λειτουργικού Συστήματος Πραγματικού Χρόνου (σημαφόροι, μηνύματα, ουρές μηνυμάτων, καθυστερήσεις, διεργασίες με προτεραιότητες κτλ) για την ανάπτυξη εφαρμογών. Την λύση σε αυτό το πρόβλημα μας δίνει το Salvo RTOS μαζί με την χρήση του μεταγλωτιστή PICC της HI-TECH. ΣΕ ΛΙΓΟ ΚΑΙΡΟ ΘΑ ΑΝΑΡΤΗΘΟΥΝ ΠΛΗΡΩΣ ΟΙ ΣΧΕΤΙΚΕΣ ΠΛΗΡΟΦΟΡΙΕΣ.

[Salvo an1](#) , [Salvo an4](#) , [Salvo an17](#) , [Salvo and PICC Compiler](#) , [Ένα πολύ καλό και αναλυτικό παράδειγμα εφαρμογής !!!](#)

Ένα παράδειγμα κώδικα και μεταγλώττισης είναι το παρακάτω:

```
/******
```

```
Copyright (C) 1995-2002 Pumpkin, Inc. and its  
Licensor(s). Freely distributable.
```

```
$Source: C:\RCS\d\salvo\tut\tu6\main.c,v $  
$Author: aek $  
$Revision: 3.10 $  
$Date: 2003-07-08 23:15:13-07 $
```

Source for tutorial programs.

```
*****/
```

```
#include "hdrpiccd.h"  
#include "main.h"  
#include "salvo.h"
```

```
#define TASK_COUNT_P    OSTCBP(1) /* task #1    */
#define TASK_SHOW_P    OSTCBP(2) /* "" #2    */
#define TASK_BLINK_P   OSTCBP(3) /* "" #3    */
#define PRIO_COUNT     12    /* task priorities*/
#define PRIO_SHOW      10    /* ""    */
#define PRIO_BLINK     2     /* ""    */
#define MSG_UPDATE_PORT_P OSECBP(1) /* semaphore #1  */
```

```
unsigned int counter;
```

```
char CODE_B = 'B';
char CODE_C = 'C';
```

```
_OSLabel(TaskCount1)
_OSLabel(TaskShow1)
_OSLabel(TaskBlink1)
_OSLabel(TaskBlink2)
```

```
void TaskCount( void ) // ΔΙΕΡΓΑΣΙΑ ΜΕΤΡΗΤΗ
```

```
{
    counter = 0;

    for (;;) {
        counter++;

        if ( !(counter & 0x01FF) )
            OSSignalMsg(MSG_UPDATE_PORT_P, (OStypeMsgP) &CODE_C); //ΣΤΕΙΛΕ
            ΣΗΜΑΤΟΦΟΡΕΑ

        OS_Yield(TaskCount1); // ΑΛΛΑΞΕ ΔΙΕΡΓΑΣΙΑ
    }
}
```

```
void TaskShow( void ) // ΔΙΕΡΓΑΣΙΑ ΕΜΦΑΝΙΣΗΣ
{
    OStypeMsgP msgP;

    InitPORT();

    for (;;) {
        OS_WaitMsg(MSG_UPDATE_PORT_P, &msgP, OSNO_TIMEOUT, TaskShow1); //ΠΕΡΙ
        ΜΕΝΕ ΣΗΜΑΤΟΦΟΡΕΑ

        if ( *(char *)msgP == CODE_C ) {
            PORT = (PORT & ~0xFE) | ((counter >> 8) & 0xFE);
        }
        else
            PORT ^= 0x01;
    }
}
```

```
void TaskBlink( void ) // ΔΙΕΡΓΑΣΙΑ ΑΝΑΨΕ-ΣΒΗΣΕ
{
    OStypeErr err;

    for (;;) {
        OS_Delay(50, TaskBlink1); //ΚΑΘΥΣΤΕΡΗΣΗ 50ms

        err = OSSignalMsg(MSG_UPDATE_PORT_P, (OStypeMsgP) &CODE_B);

        if ( err == OSERR_EVENT_FULL ) {
            OS_SetPrio(PRIO_SHOW+1, TaskBlink2); //ΑΛΛΑΞΕ ΤΗΝ ΠΡΟΤΕΡΑΙΟΤΗΤΑ
            OSSignalMsg(MSG_UPDATE_PORT_P, (OStypeMsgP) &CODE_B);
            OSSetPrio(PRIO_BLINK);
        }
    }
}
```

```
void main( void )
{
    Init();

    OSInit(); //ΑΡΧΙΚΟΠΟΙΗΣΗ ΤΟΥ ΛΕΙΤΟΥΡΓΙΚΟΥ ΣΥΣΤΗΜΑΤΟΣ

    //ΔΗΜΙΟΥΡΓΙΑ ΤΡΙΩΝ ΔΙΕΡΓΑΣΙΩΝ ΜΕ ΠΡΟΤΕΡΑΙΟΤΗΤΕΣ
    OSCreateTask(TaskCount, TASK_COUNT_P, PRIO_COUNT);
    OSCreateTask(TaskShow, TASK_SHOW_P, PRIO_SHOW);
    OSCreateTask(TaskBlink, TASK_BLINK_P, PRIO_BLINK);

    OSCreateMsg(MSG_UPDATE_PORT_P, (OSTypeMsgP) 0); //ΔΗΜΙΟΥΡΓΙΑ ΣΗΜΑΦΟΡΟΥ

    OSEi();

    for (;;)
        OSSched(); //ΕΝΑΡΞΗ ΛΕΙΤΟΥΡΓΙΑΣ ΤΟΥ ΧΡΟΝΟΔΡΟΜΟΛΟΓΗΤΗ
}
```

Memory Usage Map:

Program ROM \$0000 - \$0088 \$0089 (137) words
Program ROM \$04DA - \$07FF \$0326 (806) words
\$03AF (943) words total Program ROM

Bank 0 RAM \$0020 - \$0037 \$0018 (24) bytes
Bank 0 RAM \$0070 - \$0072 \$0003 (3) bytes
\$001B (27) bytes total Bank 0 RAM

Bank 1 RAM \$00A0 - \$00B7 \$0018 (24) bytes total Bank 1 RAM

Program statistics:

Total ROM used 943 words (11.5%)
Total RAM used 51 bytes (13.9%) Μόνο !!!!

Build completed successfully.