Performing Periodic Tasks

One way to install a periodic **Time Manager** task is to have the task reactivate itself. Because the task record is already inserted into the **Time Manager** task queue, the task can simply call **PrimeTime** to do this. To call **PrimeTime**, however, the task needs to know the address of the corresponding task record. In the revised and extended Time Managers, the task record's address is placed into register A1 when the task is called. The following program illustrates how the task can reactivate itself by retrieving the address in register A1 and passing that address to **PrimeTime**.

```
// Defining a periodic Time Manager task

// Assuming inclusion of <MacHeaders>

typedef struct {            // Time Manager information record
    TMTask    atmTask;   // original and revised TMtask record
    long      tmRefCon;  // space to pass address of A5 world
} TMInfo;

typedef TMInfo *TMInfoPtr;

pascal TMInfoPtr GetTMInfo (void)
    = 0x2E89;                   // MOVE.L A1,(SP)

pascal void MyTask (void);

pascal void MyTask ()               // for revised and extended TMs
{
    TMInfoPtr recPtr;
    long      myDelay;      // delay value

    recPtr = GetTMInfo();         // first get your own address
    myDelay = 2000;           // no. of milliseconds to delay

    // do something in here

    PrimeTime((QElemPtr) recPtr, myDelay);
}
```

Note: The technique illustrated in this program cannot be used with the original Time Manager because that version of the **Time Manager** does not pass the address of the task record in register A1.