HOW LONG DID IT TAKE?

George Gwilt

Early this afternoon the postman delivered somemail. This included two interesting items. They were the latest editions of Quanta and QL Today. I opened the Quanta magazine and saw inside page 3 an imposing black bordered rectangle. This contained the information that the COPY DATE for JUNE/JULY 2011 was 5th July 2011. A glance at my watch told me that today's date was 4th July. Blimey! This does not give me mch time for thinking up, writing, correting, re-correcting and sending off an article. Since time is, as they say, of the essence I thought I might give a short explanation of how I myself work out how long programs take to run.

It is usual for programmers ot use DATE for this purpose. DATE is the SuperBASIC function returning the number of seconds that have elapsed since the beginning of 1961. To time a program, all you need to do is record the value of DATE at the start and subtract this from its value when the program ends. However, this gives the time in no more finely divided time interval than seconds. Furthermore, it can give an answer up to a second out either way. Thus if the program starts just after DATE has been altered and ends just before an alteration to DATE the caluclated time will be nearly one second below the actual elapsed time.

A more accurate measurement can be obtained by using the computer's internal timing device. The QL has an interrupt each 1/50th or 1/60th of a second depending on its location. In the UK it is 1/50th of a second that is used.

Let us call that fraction of a second, which is either 1/5th or 1/60th, atick. If we were able to count the number of ticks that had occurred while a program is running we could tell the elapsed time at least 50 times as accurately as we can by using DATE.

The Timer

I use such a timer.

My timer, ROUT, consists of just two instructions.

```
ROUT  ADDQ.L     #1,$DC(A6)       ;Add 1 to the counter

      RTS
```

The routine, ROUT, has to be linked into the polled list so that it will be called at each tick. When the code is called by the operating system A6 points to the system variables. You will see, then, that I have used the relative address $DC for the counter. This is because the four long words at the relative address $D0 are designated as SYS_FPU and are earmarked for programs allowing use of a Floating Point Unit (FPU) where one exists. For example both Q40 and Q60 have an FPU. Since I was the person who wrote the programs for the FPU I know that, so fra at any rate, only the first three long words are used for that purpose. So I know that the long word at relative address $DC in system variables is available for use as a timer.

Using The Timer

To time a process I set

```
tme = PEEK_L($280DC)
```

at the start and

```
PRINT (PEEK_L($280DC)-tme)/50
```

at the end.

Linking The Timer

In order to set the timer going it has to be linked into the polled list. The code to do that is:

```
      LEA      LNK,A0           ;address of link
      LEA      ROUT,A1          ;address of routine to be linked
      MOVE.L   A1,4(A0)         ;Set address of ROUT in the link
      MOVEQ    #MT_LPOLL,D0     ;Link…
      TRAP     #1               ;…it in
      MOVEQ    #0,D0            ;Return…
      RTS                       ;…to BASIC
LNK   DS.L  2                   ;Link to the next routine
ROUT  ADDQ.L   #1,$DC(A6)       ;Add 1 to the counter
```

All you have to do is assemble this code and LRESPR the result.

Finally

I would be interested in comments on this timing device. Indeed I have a few myself which I may divulge at a later date.