By calculating the delay time in system units instead of in seconds, the program avoids having to convert system time to seconds each loop cycle.

**Type Aliases**

C++ has two ways to establish a new name as an alias for a type. One is to use the preprocessor:

```
#define BYTE char // preprocessor replaces BYTE with char
```

The preprocessor then replaces all occurrences of `BYTE` with `char` when you compile a program, thus making `BYTE` an alias for `char`.

The second method is to use the C++ (and C) keyword `typedef` to create an alias. For example, to make `byte` an alias for `char`, do this:

```
typedef char byte; // makes byte an alias for char
```

Here's the general form:

```
typedef typeName aliasName;
```

In other words, if you want `aliasName` to be an alias for a particular type, declare `aliasName` as if it were a variable of that type and then prefix the declaration with the `typedef` keyword. For example, to make `byte_pointer` an alias for pointer-to-char, declare `byte_pointer` as a pointer-to-char and then stick `typedef` in front:

```
typedef char * byte_pointer; // pointer to char type
```

You could try something similar with `#define`, but that won't work if you declare a list of variables. For example, consider the following:

```
#define FLOAT_POINTER float *
FLOAT_POINTER pa, pb;
```