必应

当然可以。下面是一个简单的例子，演示如何使用 __cpu_time__ 宏来计算CPU使用率：

```
#include <stdio.h>
#include <stdint.h>

// 假设有一个函数 get_system_ticks() 返回当前系统时间
int64_t get_system_ticks() {
    // 返回当前系统时间
    return 0;
}

#define SAFE_NAME(__NAME) CONNECT3(__,__NAME,__LINE__)
#define CONNECT3(__A, __B, __C) __CONNECT3(__A, __B, __C)
#define __CONNECT3(__A, __B, __C) __A##__B##__C

#define start_task_cycle_counter(...) start_cycle_counter()

int64_t g_lLastTimeStamp = 0;

void start_cycle_counter(void)
{
 g_lLastTimeStamp = get_system_ticks();
}

#define stop_task_cycle_counter(...) stop_cycle_counter()

int64_t stop_cycle_counter(void)
{
 int64_t lTemp = (get_system_ticks() - g_lLastTimeStamp);

 return lTemp;
}

#define __cpu_time__(__CNT, ...) \
 static int64_t SAFE_NAME(s_lTimestamp) = 0, SAFE_NAME(s_lTotal) = 0; \
 static uint32_t s_wLoopCounter = (__CNT); \
 using(float __usage__ = 0, ({ \
 if (0 == s_wLoopCounter) { \
 __usage__ = (float)((double)SAFE_NAME(s_lTotal) \
 / (double)( get_system_ticks() \
 - SAFE_NAME(s_lTimestamp))); \
 __usage__ *= 100.0f; \
 SAFE_NAME(s_lTimestamp) = 0; \
 SAFE_NAME(s_lTotal) = 0; \
 if (__PLOOC_VA_NUM_ARGS(__VA_ARGS__) == 0) { \
 printf("CPU Usage %3.2f%%\r\n", (double)__usage__); \
 } else { \
 __VA_ARGS__ \
 } \
 } \
 if (0 == SAFE_NAME(s_lTimestamp)) { \
 SAFE_NAME(s_lTimestamp) = get_system_ticks(); \
 s_wLoopCounter = (__CNT); \
 } \
 start_task_cycle_counter();}), \
 ({SAFE_NAME(s_lTotal) += stop_task_cycle_counter(); \
 s_wLoopCounter--;}))

int main() {
    for (int i = 0; i < 100; i++) {
        __cpu_time__(10,
            // 这里是你想要测量CPU使用率的代码
            // 比如执行一些计算
            int x = i * i;
            int y = x * x;
        )
    }
```

```
    return 0;
}
```

在这个例子中，我们在 `main` 函数中的循环中调用了 `__cpu_time__` 宏。我们传入了两个参数：第一个参数 `10` 表示每10次调用后计算并打印出CPU使用率；第二个参数是我们想要测量CPU使用率的代码，即执行一些计算。

当我们运行这个程序时，它会在每10次循环后计算并打印出CPU使用率。

注意，这个例子中省略了一些细节，例如 `get_system_ticks` 函数的实现和 `g_nOffset` 变量的定义。如果你想要运行这个例子，你需要根据你自己的系统实现这些细节。