

Table of Contents

증상	1
발생 시점	1
원인	1
Solution (ComilDE)	1
Solution (C++)	1
Solution (C#)	2

프로그램 느려짐, 타이밍 문제 발생

× PC 교체, Windows 재설치 이후 Application에서 딜레이가 발생하는 경우에 대한 안내 페이지입니다.

증상

- Application 의 변경 없이 프로그램이 느려지거나 타이밍 문제 발생
- Serial이 통신의 비정상 동작
- 동일 사양의 양산 설비에서 설비에 따라 증상이 나오거나 나오지 않음

발생 시점

- PC 교체 후
- Windows 재설치 및 버전 변경 후

원인

- Windows의 **Timer Resolution**이 기본값으로 설정되어 있음
- Windows의 Timer 는 15.625ms 의 Resolution을 기본값으로 가지며,
- 이 경우, 1ms Sleep 명령에도 15ms 또는 16ms 의 Sleep을 가짐
- Sleep Time이 길어짐에 따라 상기 문제 발생
- Visual Studio 등의 프로그램 설치에 따라 Timer Resolution이 1ms로 바뀔 수 있음

Solution (ComiIDE)

- **Shell**을 이용한 **TimerResolution** 확인, 변경 참조

Soultion (C++)

```
// timeBeginPeriod 사용
// 1ms 까지 설정가능
// 0.5ms Resolution 필요 시 NtSetTimerResolution 사용

#define TARGET_RESOLUTION 1 // 1-millisecond target resolution

TIMECAPS tc;
UINT wTimerRes;
```

```
if (timeGetDevCaps(&tc, sizeof(TIMECAPS)) != TIMERR_NOERROR)
{
    // Error; application can't continue.
}

wTimerRes = min(max(tc.wPeriodMin, TARGET_RESOLUTION), tc.wPeriodMax);
timeBeginPeriod(wTimerRes);

//      do your stuff here at approx. 1 ms timer resolution
timeEndPeriod(wTimerRes);
```

Solution (C#)

```
[DllImport("ntdll.dll", SetLastError = true)]
private static extern NtStatus NtSetTimerResolution(
    uint DesiredResolution,
    bool SetResolution,
    ref uint CurrentResolution);

public static float SetTimerResolution(uint
timerResolutionIn100nsUnits, bool doSet = true)
{
    // 입출력값의 단위는 100ns
    // 1ms로 변경 시 SetTimerResolution(10000);
    // Enum NtStatus 는 별도 정의, 불필요 시 int로 대체

    uint currentRes_100ns = 0;
    var result = NtSetTimerResolution(timerResolutionIn100nsUnits,
doSet, ref currentRes_100ns);
    return currentRes_100ns;
}
```

From:

<http://comizoa.co.kr/info/> - -

Permanent link:

http://comizoa.co.kr/info/doku.php?id=faq:downloadandsetup:common:09_timing&rev=1623042182

Last update: **2024/07/08 18:22**