

SylixOS 实时性能测试报告

SylixOS 测试报告

TR0010010001 V1.00

Date: 2014/09/15

测试报告

类别	内容
关键词	SylixOS 示波器 Matlab 中断响应能力 任务切换时间
摘 要	使用示波器和 Matlab 测试反映 SylixOS 实时性能的两个关键参数：中断响应能力和任务切换时间

修订历史

版本	日期	原因
V1.00	2014/09/15	创建文档

目 录

1. 测试目的.....	1
2. 测试工具.....	1
3. 测试硬件平台.....	2
4. 操作系统.....	2
5. 软件编译参数.....	2
6. 中断响应能力测试方法.....	3
6.1 测试方案设计.....	3
6.2 代码设计.....	4
6.3 示波器设置.....	5
7. 中断响应能力测试结果.....	6
8. 任务切换时间测试方法.....	8
8.1 测试方案设计.....	8
8.2 代码设计.....	9
8.3 示波器设置.....	11
9. 任务切换时间测试结果.....	12

1. 测试目的

测试反映 SylixOS 实时性能的两个关键参数：中断响应能力和任务切换时间。

2. 测试工具

测试工具使用北京普源精电科技有限公司设计、生产和发行销售的 DS1104Z 型数字示波器，如图 2.1。



图 2.1 DS1104Z 型数字示波器

产品特性如下：

1. 模拟通道带宽：100MHz/70MHz
2. 4 个模拟通道，16 个数字通道（MSO）
3. 实时采样率达 1GSa/s
4. 标配存储深度达 12Mpts，选配达 24Mpts
5. 波形捕获率达 30,000 个波形每秒
6. 多达 6 万帧的硬件实时波形不间断录制和回放功能（选配）
7. 独创的 UltraVision 技术
8. 丰富的触发和总线的解码功能
9. 低底噪声，垂直档位 1mV/div~10V/div
10. 内置 25MHz 双通道函数/任意波发生器（MSO/DS1000Z-S）
11. 丰富的接口：USB Host&Device、LAN(LXI Core Device 2011)、AUX、USB-GPIB（可选）
12. 新颖精巧的工业设计，便捷的操作
13. 7 英寸 WVGA（800X480）TFT 宽屏，多级波形灰度显示

3. 测试硬件平台

测试硬件平台使用广州友善之臂电子有限公司设计、生产和发行销售的 mini210s 型开发板，如图 3.1。

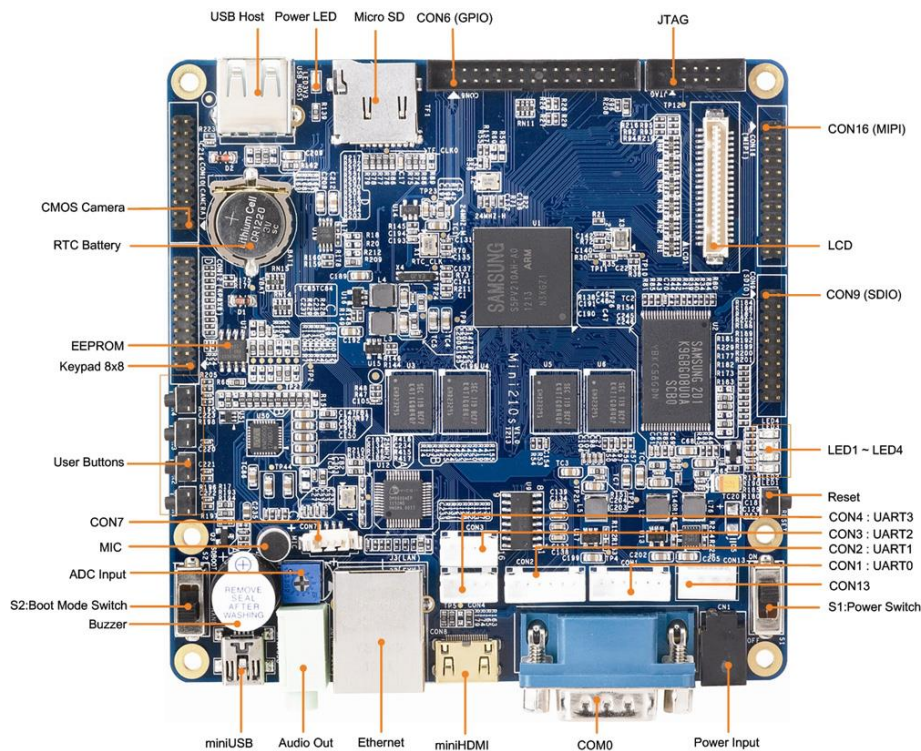


图 3.1 mini210 开发板

关键硬件参数如下：

CPU: Samsung 公司的 S5PV210 芯片（基于 ARM Cortex-A8，运行主频为 1GHz，带有 32KB 的 D-Cache 和 32KB 的 I-Cache 及 512KB 的 L2-Cache）

内存: 512MB DDR2（32bit 数据总线，单通道，@200MHz）

FLASH 存储: 512M SLC NAND Flash

4. 操作系统

测试硬件平台运行的操作系统为 SylixOS，libsylixos 的版本为 1.0.0-rc43，bspmart210 的版本为 1.00rc。

libsylixos 关闭了对 MONITOR 和 SMP 支持。

5. 软件编译参数

编译 libsylixos 和 bpsmart210 使用的工具链为 arm-sylixos-toolchain，其中 c 编译器 arm-sylixos-eabi-gcc 的版本为 4.8.4。

编译 libsylixos 和 bspsmart210 时，编译参数中的 cpu 参数为“-mcpu=cortex-a8”，优化参数为“-O2 -Os”，即使用 O2 等级优化并且优化代码体积。

6. 中断响应能力测试方法

6.1 测试方案设计

由于 DS1104Z 型数字示波器带有 1KHz 方波输出，将 S5PV210 的 GPH0(1) GPIO 管脚配置为外部中断模式，即成为 EINT1 管脚，触发方式配置为上升沿触发。将 DS1104Z 型数字示波器的 1KHz 方波输出连接到 EINT1 管脚，SylixOS 将每秒响应外部中断 EINT1 1000 次；同时将 1KHz 方波输出连接到 DS1104Z 型数字示波器的模拟输入通道 CH1。

将 S5PV210 的 GPJ3 (7) GPIO 管脚配置为输出模式，将 GPJ3 (7) 管脚连接到 DS1104Z 型数字示波器的模拟输入通道 CH2。

将 DS1104Z 型数字示波器的方波信号地连接到 mini210s 开发板的数字地 DGND。

硬件连接如图 6.1。

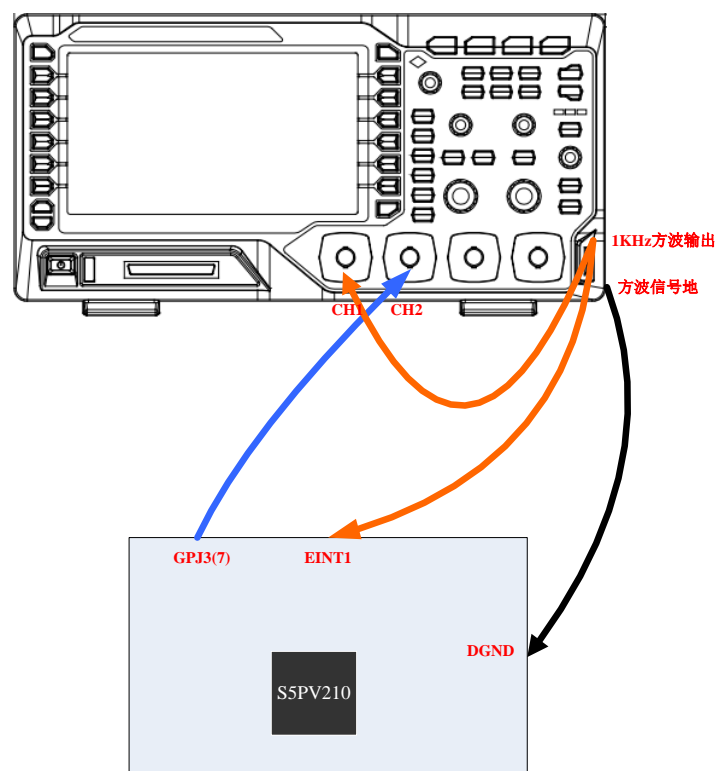


图 6.1 硬件连接

编写 halIdleHook 函数，halIdleHook 函数的功能是将 GPJ3 (7) 管脚电平置为低电平，将 halIdleHook 函数添加为空闲线程 t_idle 的钩子函数。

而在 EINT1 的中断服务函数里将 GPJ3 (7) 管脚电平置为高电平。由于 cmos 电路，电压高于 $0.7 \times V_{cc}$ 即被识别为高电平，根据 S5PV210 芯片数据手册得知， $V_{cc}=3.3V$ ，从 EINT1 接收到上升沿信号（电平高于 $0.7 \times V_{cc}=0.7 \times 3.3=2.31V$ 时）到 GPJ3 (7) 管脚变为高电平的时间即为中断响应时间。

6.2 代码设计

bspInit.c 增加的代码如程序清单 6.1 所示。

程序清单 6.1 bspInit.c 增加代码

```
#define SMART210_GPJ37_GPIO          S5PV210_GPJ3(7)
#define SMART210_EINT1_GPIO          S5PV210_GPH0(1)

/*
 * 空闲线程 t_idle 的勾子函数
 */
static VOID halIdleHook (VOID)
{
    *(volatile UINT32 *)S5PV210_GPJ3DAT = 0x00;
}

/*
 * 初始化空闲线程 t_idle 的勾子函数
 */
static VOID halIdleInit (VOID)
{
    API_SystemHookAdd(halIdleHook,
                      LW_OPTION_THREAD_IDLE_HOOK);
}

/*
 * EINT1 中断服务函数
 */
static irqreturn_t SylixOSRealTimeTestEint1Isr (VOID)
{
    irqreturn_t iRet;

    *(volatile UINT32 *)S5PV210_GPJ3DAT = 0xff;

    iRet = API_GpioSvrIrq(SMART210_EINT1_GPIO);
    if (iRet == LW_IRQ_HANDLED) {
        API_GpioClearIrq(SMART210_EINT1_GPIO);
    }

    return (iRet);
}

/*
 * SylixOS 实时测试初始化
 */
```

```
static INT SylixOSRealTimeTestInit (VOID)
{
    INT iRet;
    INT iIrq;

    API_GpioRequestOne(SMART210_GPJ37_GPIO, LW_GPIOF_OUT_INIT_LOW, "gpj37");

    iRet = API_GpioRequestOne(SMART210_EINT1_GPIO, LW_GPIOF_OUT_INIT_LOW, "eint1");
    if (iRet != ERROR_NONE) {
        return (PX_ERROR);
    }

    iIrq = API_GpioSetupIrq(SMART210_EINT1_GPIO, LW_FALSE, 1);
    if (iIrq < 0) {
        return (PX_ERROR);
    }

    API_InterVectorConnect((ULONG)iIrq,
                           (PINT_SVR_ROUTINE)SylixOSRealTimeTestEint1Isr,
                           LW_NULL, "eint1_isr");

    API_InterVectorEnable((ULONG)iIrq);

    return (ERROR_NONE);
}
```

6.3 示波器设置

将示波器的 CH1 与 CH2 通道设置如下：

1. 耦合：直流
2. 带宽限制：关闭
3. 探头：10X
4. 反相：关闭
5. 幅度档位：粗调
6. 单位：**【V】**

将示波器的触发设置如下：

1. 触发类型：边沿触发
2. 信源：CH1
3. 边沿类型：上升沿
4. 触发方式：自动
5. 触发电压：2.2V（略低于 2.31V）

将示波器的光标设置如下：

1. 模式：自动
2. 自动项目：项目 1（即延迟 1→2 μ s）

将示波器的测量设置如下：

1. 信源：CH1
2. 频率计：关闭
3. 清除测量：只保留测试项 1
4. 全部测量：关闭
5. 全部测量信源：CH1 与 CH2
6. 统计：打开
7. 统计选择：极值
8. 设置：类型为门限、上限为 95%、中限为 94%、下限为 10%
9. 测量范围：屏幕区域
10. 测量历史：关闭，显示方式为图形

注：探头补偿正确后，示波器的 1KHz 方波输出连接到开发板的 EINT1 管脚后，电压最大值的平均值为 2.4V，中限为 $2.31V/2.4V=0.9625=96\%$ ，但由于上限最高能设置为 95%，中限最高能设置为 94%，故中限设置为 94%，由于不是目标的 96%，所以测量的延迟 1→2 μ s 时间将比实际的稍长。

7. 中断响应能力测试结果

由于 SylixOS 将每秒响应外部中断 EINT1 1000 次，20 分钟将响应 120 万次，使用示波器测量这 120 万次中断的最小响应时间、平均响应时间、最大响应时间，结果如图 7.1 所示。

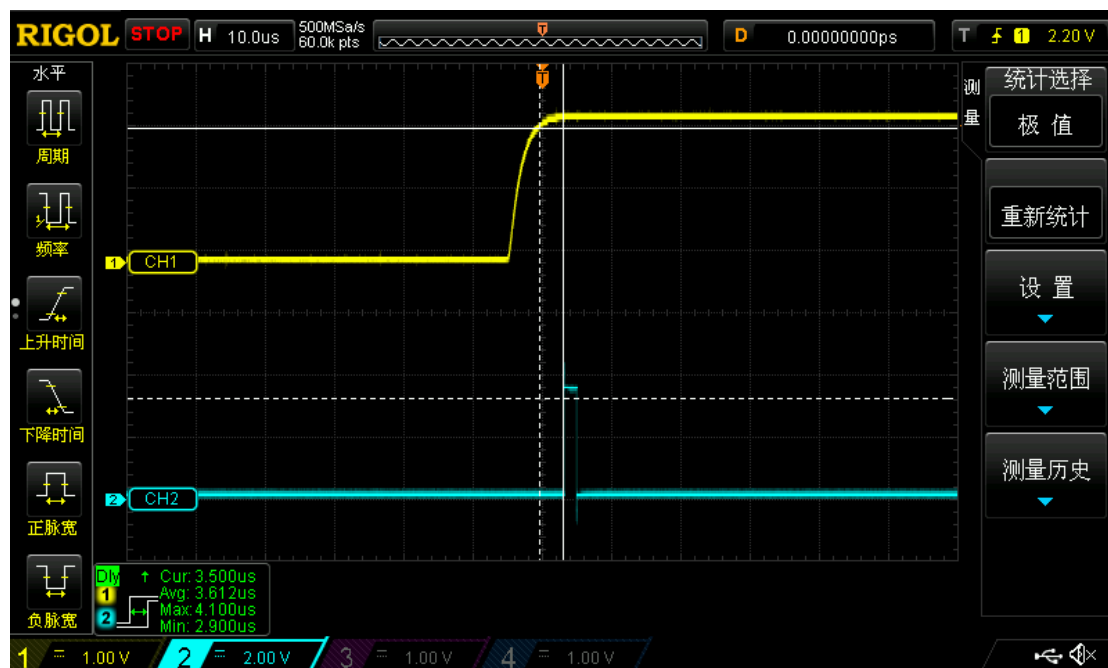


图 7.1 中断响应时间测量结果

总结如表 7.1。

表 7.1 中断响应时间

120 万次中断	时间
最小响应时间	2.900us
平均响应时间	3.612us
最大响应时间	4.100us

同时使用如程序清单 7.1 的 matlab 程序从中抓取 1 万个数值。

程序清单 7.1 matlab 程序

```
MSO1000Z = visa('ni','USB0::0x1AB1::0x04CE::DS1ZA154902817::INSTR');

MSO1000Z.InputBufferSize = 2048;

fopen(MSO1000Z);

N = 10000;

delay{1} = 1;
delay{N} = 6;

for i=2:N-1
    fprintf(MSO1000Z, 'MEASure:STATistic:ITEM? CURRent,RDElay,CHANnel1,CHANnel2' );
    value = fgets(MSO1000Z);
    delay{i} = str2double(value) * 1000000;
end

fclose(MSO1000Z);
delete(MSO1000Z);
clear MSO1000Z;

new_delay = cell2mat(delay);

x = sort(new_delay);
y = normpdf(x, mean(x), 0.1);

subplot(211);

plot(x, y);

xlabel('SylixOS 响应时间(us);
```

生成的正态分布图如图 7.2。

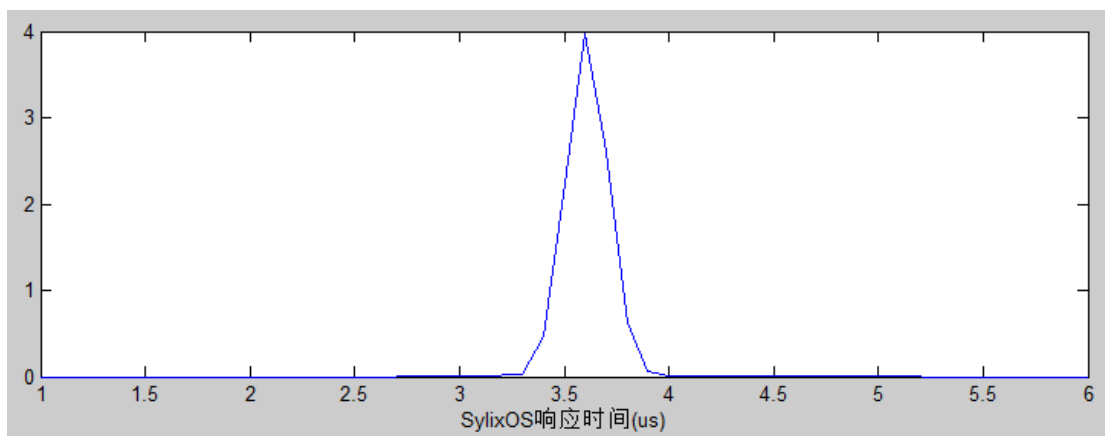


图 7.2 中断响应时间正态分布图

8. 任务切换时间测试方法

8.1 测试方案设计

将 S5PV210 的 GPJ3 (7) GPIO 管脚配置为输出模式，将 GPJ3 (7) 管脚连接到 DS1104Z 型数字示波器的模拟输入通道 CH1。

硬件连接如图 8.1。

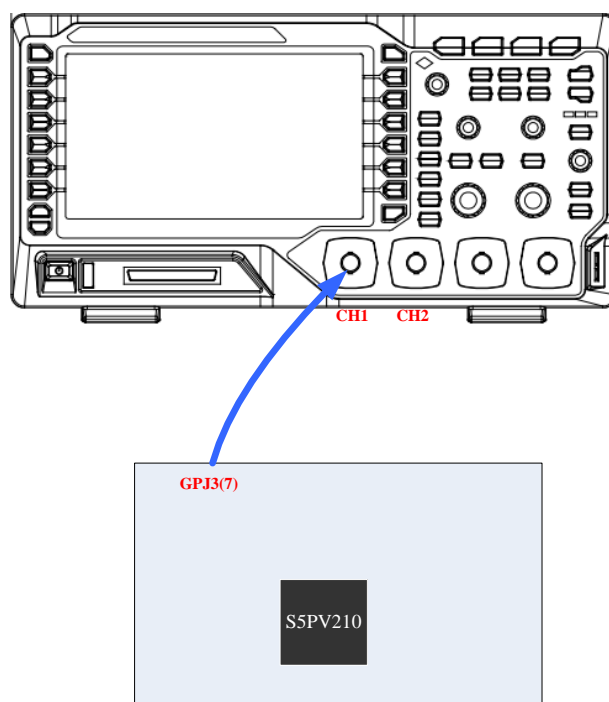


图 8.1 硬件连接

SylixOS 线程上下文切换的函数是 archTaskCtxSwitch 函数，在 archTaskCtxSwitch 进入时，将 GPJ3 (7) 管脚置为高电平；在 archTaskCtxSwitch 退出时，将 GPJ3 (7) 管脚置为低电平；GPJ3 (7) 管脚为高电平的时间将是 archTaskCtxSwitch 函数的执行消耗时间，即示波器的 CH1 通道的正脉宽。

修改 t_led 的线程函数 smart210LedThread, smart210LedThread 函数的功能是不断休眠 1 个 tick, 由于 SylixOS 的时钟频率是 100Hz, 所以 SylixOS 将每秒进行任务切换至少 100 次。

8.2 代码设计

bspInit.c 增加的代码如程序清单 8.1 所示。

程序清单 8.1 bspInit.c 增加代码

```
#define SMART210_GPJ37_GPIO          S5PV210_GPJ3(7)

/*
 * SylixOS 实时测试初始化
 */
static INT  SylixOSRealTimeTestInit (VOID)
{
    INT  iRet;

    iRet = API_GpioRequestOne(SMART210_GPJ37_GPIO, LW_GPIOF_OUT_INIT_LOW, "gpj37");
    if (iRet != ERROR_NONE) {
        return  (PX_ERROR);
    }

    return  (ERROR_NONE);
}

static VOID  smart210LedThread (VOID)
{
    while (1) {
        API_TimeSleep(1);
    }
}
```

修改后的 archTaskCtxSwitch 函数如程序清单 8.2 所示。

程序清单 8.2 archTaskCtxSwitch 函数

```
FUNC_DEF(archTaskCtxSwitch)

/* 将 GPJ3 (7) 管脚置为高电平 */
ldr r1, =0xE02002A4
mov r2, #0xFF
str r2, [r1]

STMFD  SP!, {LR}                                /* 任务返回 PC 值
入栈      */
```

```

STMFD    SP!, {LR}
STMFD    SP!, {R0-R12}                ;/* Push registers
*/
MRS      R4 , CPSR                    ;/* CPSR 入栈
*/
STMFD    SP!, {R4}

MOV      R4 , R0                      ;/* stack current = SP
*/
LDR      R5 , [R4]
STR      SP , [R5]

STMFD    SP!, {R0}                    ;/* 当前 CPU 信息
入栈      */
LDR      R1 , =_SchedSwp              ;/* _SchedSwp();
*/
MOV      LR , PC
BX       R1
LDMFD    SP!, {R0}                    ;/* 当前 CPU 信息
出栈      */

MOV      R4 , R0                      ;/* SP = stack current
*/
LDR      R5 , [R4]
LDR      SP , [R5]

LDMFD    SP!, {R4}                    ;/* CPSR 出栈
*/
MSR      SPSR_cxsf, R4

;/* 将 GPI3 (7) 管脚置为低电平 */
ldr r1, =0xE02002A4
mov r2, #0x00
str r2, [r1]

LDMFD    SP!, {R0-R12, LR, PC}^       ;/* 包括 PC 的所有
寄存器出栈, */

;/* 同时更新 CPSR
*/
FUNC_END()

```

8.3 示波器设置

将示波器的 CH1 通道设置如下：

1. 耦合：直流
2. 带宽限制：关闭
3. 探头：10X
4. 反相：关闭
5. 幅度档位：粗调
6. 单位：【V】

将示波器的触发设置如下：

1. 触发类型：边沿触发
2. 信源：CH1
3. 边沿类型：上升沿
4. 触发方式：自动
5. 触发电压：2.2V（略低于 2.31V）

将示波器的光标设置如下：

1. 模式：自动
2. 自动项目：项目 1（即正脉宽）

将示波器的测量设置如下：

1. 信源：CH1
2. 频率计：关闭
3. 清除测量：只保留测试项 1
4. 全部测量：关闭
5. 全部测量信源：CH1
6. 统计：打开
7. 统计选择：极值
8. 设置：类型为门限、上限为 90%、中限为 50%、下限为 10%
9. 测量范围：屏幕区域
10. 测量历史：关闭，显示方式为图形

9. 任务切换时间测试结果

SylixOS 将每秒进行任务切换至少 100 次，20 分钟将响应 12 万次，使用示波器测量这 12 万次任务切换的最小时间、平均时间、最大时间，结果如图 9.1 所示。

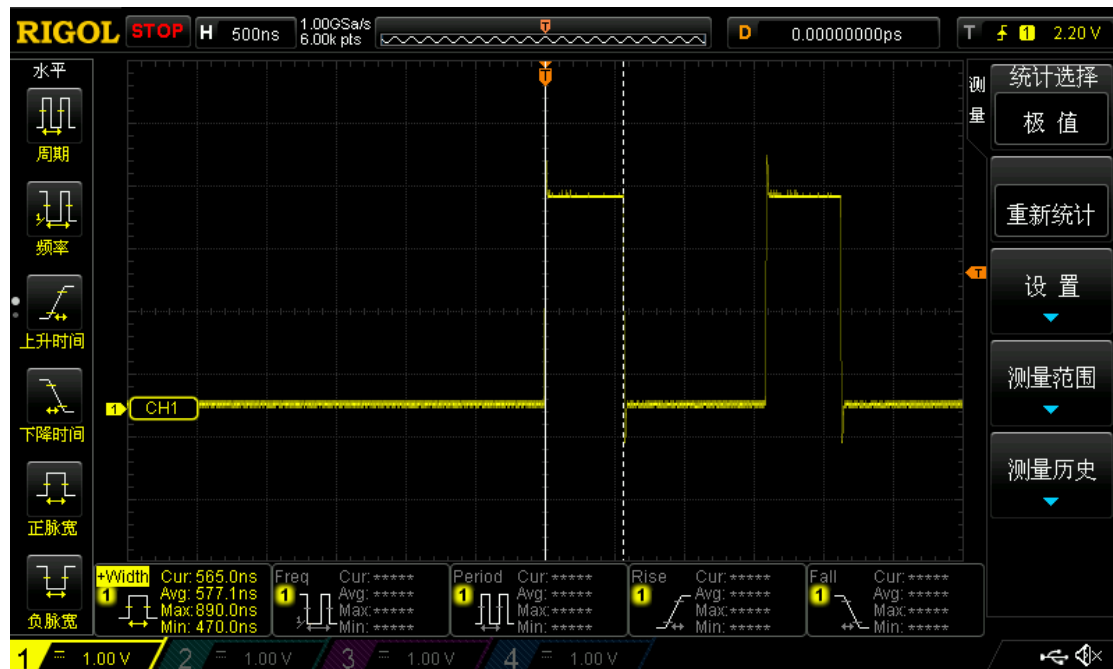


图 9.1 任务切换时间测量结果

总结如表 9.1。

表 9.1 任务切换时间

12 万次任务切换	时间
最小任务切换时间	470ns
平均任务切换时间	577.1ns
最大任务切换时间	890ns

同时使用如程序清单 9.1 所示的 matlab 程序从中抓取 1 万个数值：

程序清单 9.1 matlab 程序

```
MSO1000Z = visa('ni','USB0::0x1AB1::0x04CE::DS1ZA154902817::INSTR');

MSO1000Z.InputBufferSize = 2048;

fopen(MSO1000Z);

N = 10000;

delay{1} = 200;
```

```

delay{N} = 1000;

for i=2:N-1
    fprintf(MSO1000Z, 'MEASure:STATistic:ITEM? CURRent,PWIDth,CHANnel1');
    value = fgets(MSO1000Z);
    delay{i} = str2double(value) * 1000000000;
end

fclose(MSO1000Z);
delete(MSO1000Z);
clear MSO1000Z;

new_delay = cell2mat(delay);

x = sort(new_delay);
y = normpdf(x, mean(x), 0.1);

subplot(211);

plot(x, y);

xlabel('SylixOS 任务切换时间(ns)');
    
```

生成的正态分布图如图 9.2 所示。

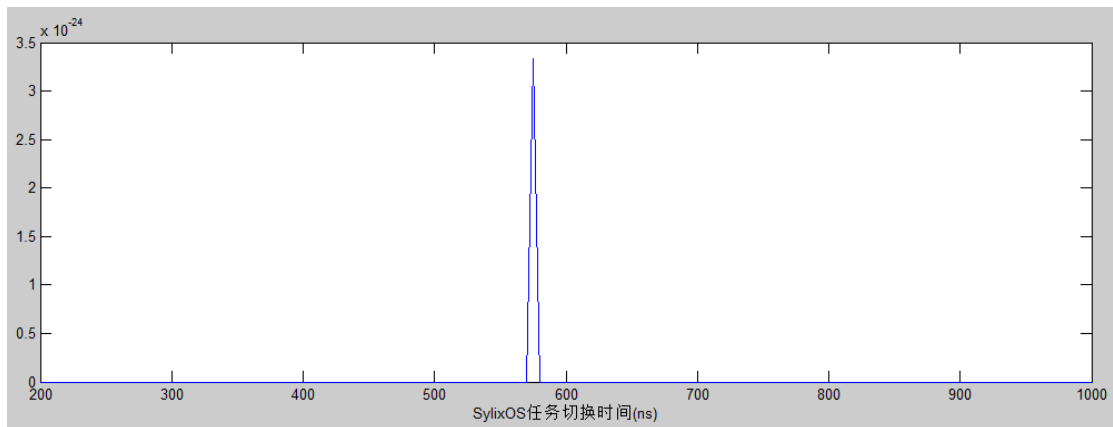


图 9.2 任务切换时间正态分布图