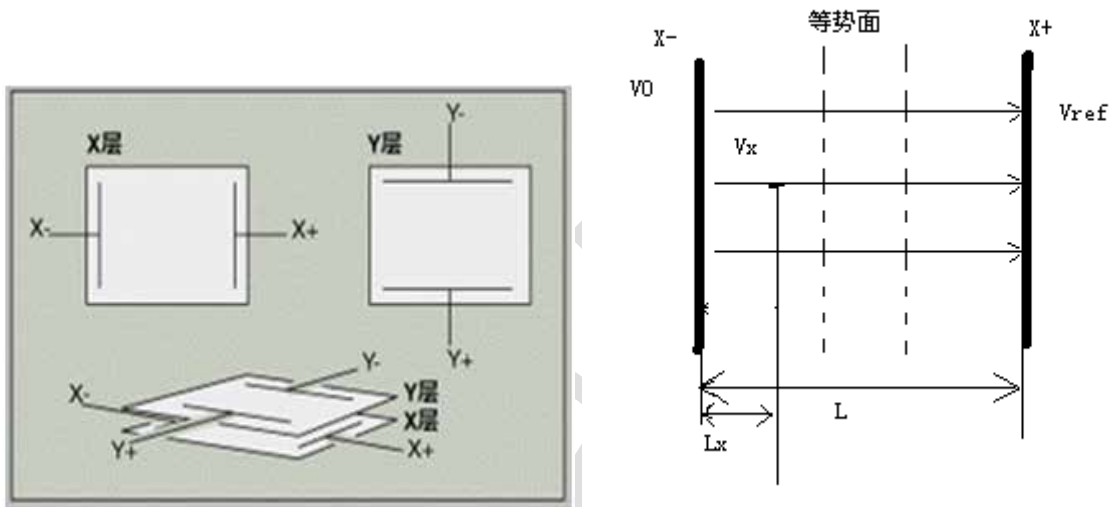


电阻触屏因其使用简单,成本较低被广泛应用在消费市场,医疗,工业和汽车市场.其中,4 线电阻触屏更是因为成本的原因,在市场中最为常见.本文将就 4 线电阻触屏与 [Xinnova](#) 32 位 XN12L 系列 MCU 连接的一个低成本低功耗方案进行讨论.

## 1. 电阻触屏原理



电阻触屏原理示意图

测量 X 坐标时：

- 1) 在 X+, X-两电极加上一个电压  $V_{ref}$  , Y+接一个高阻抗的 ADC。
- 2) 两电极间的电场呈均匀分布,方向为 X+到 X-。
- 3) 手触摸时,两个导电层在触摸点接触,触摸点 X 层的电位被导至 Y 层所接的 ADC,得到电压  $V_x$ 。
- 4) 通过  $L_x/L = V_x/V_{ref}$  ,即可得到 x 点的坐标。

Y 轴的坐标可同理将 Y+, Y-接上电压  $V_{ref}$  ,然后 X+电极接高阻抗 ADC 得到。

## 2. 传统电阻触屏方案

## 低成本低功耗 4 线电阻触屏 MCU 实现方案

在大多数应用中,一般通过外加专用的电阻触屏 IC.该 IC 一方面检测 X+,X-,Y+,Y-因用户触摸引起的电压的变化,另一方面通过 I2C/SPI 接口把检测到的 XY 坐标传输给系统 MCU.该方案主要的问题是单独的触屏 IC 增加了系统成本.把对电阻触屏 X+,X-,Y+,Y-的检测集成到系统 MCU 无疑是最好的选择.

由于在检测 X/Y 坐标位置需要用到 ADC 转换,并且要提供相应的信号源,一般的 MCU 控制比较麻烦,但幸运的是,Xinnova MCU 的引脚具备多重模拟/数字复用功能,可以轻而易举的解决这个问题.

### 3. [Xinnova](#) XN12L 系列 MCU 简介

XN12L 系列是基于 ARM M0 内核的通用 MCU。该系列可以涵盖从低端到高端各种 MCU 应用,具有高性能,低成本,代码加密可靠等特点,是取代 8 位机 16 位理想的产品。与其它 MCU 相比, XN12L 系列指令精简,内含用于增强运算的 xDSP,主频更是可高达 100MHz,外设丰富实用,支持在线调试,在目前 MCU 市场上表现非凡。主要特点有:

- 高达 100MHz ARM Cortex M0 CPU
- 高达 88KB 用户 Flash 和 16KB SRAM
- xDSP 用于增强 MCU 运算功能

#### 32 位单周期除法器

CORDIC 运算器

CRC 校验

- 多种时钟系统供用户选择

1%精度的内部晶振

支持外部时钟和晶振

内部 PLL

支持实时时钟(RTC)

- 多达 3 个独立的 ADC 转换器更适合系统高速采样需求

12 位, 1MHz 采样率

多达 12 路 ADC 通道

- 2 个模拟比较器

- 10 位 DAC, 1MHz 转换率
- 4 个增强型系统定时/计数器, 支持正交编码信号
- 集成的片上温度传感器
- 支持各种通讯接口

4 个带波特率自动检测和 IrDA 功能 UART

1 个 SPI

1 个 Quad SPI (支持 Flash 4 IOs 数据传输)

1 个 TWS (I2C 兼容)

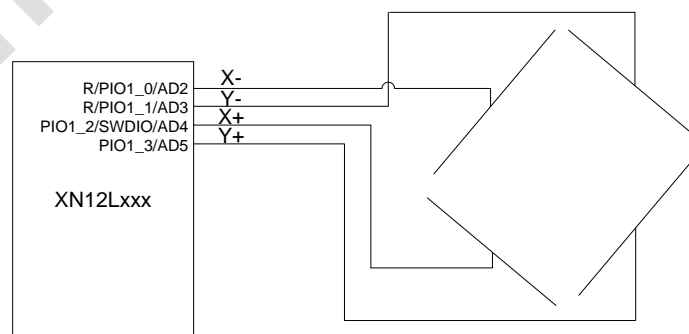
- 支持内存, 外设间的 DMA 大容量数据传输
- 支持故障诊断恢复功能 (WDT/BOD)
- 支持睡眠, 深度睡眠和掉电三种低功耗模式
- 数据和程序的高可靠和保密性能

2 个 128 位密码的分区加密和保护技术, 确保片内数据安全和防知识产权的克隆

加密模式下的应用二次开发, 更好知识产权回报

- 单电源供电 (3.3v)

## 4. 硬件连接



MCU 与 4 线电阻触屏连接示意图

# 低成本低功耗 4 线电阻触屏 MCU 实现方案

在该方案中,Y+,X+,Y-,X-直接与 MCU 引脚相连,无需任何外部器件。

## 5. 软件实现方法

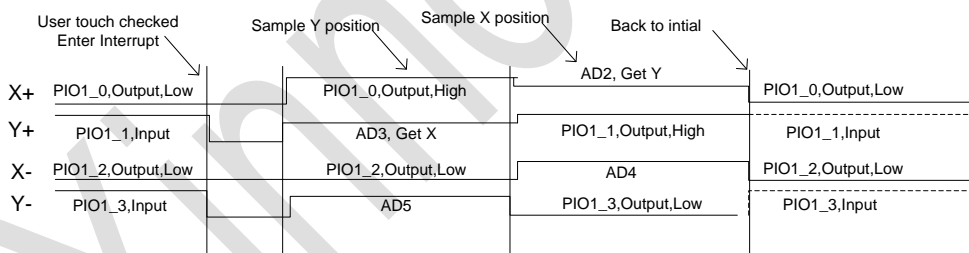
在该应用中,重点考量以下几个方面:

- a) 低功耗考虑
- b) CPU 时间占用考虑
- c) AD 检测点考虑
- d) 反应时间/灵敏度
- e) 位置校对

考虑到 CPU 时间占用和功耗,CPU 不要时时刻刻对 Y+,X+,Y-,X-进行检测。只有在用户触屏时才会做出反应。因此,我们考虑使用中断来对用户触屏进行检测。在平常,X+,X-设置为 GPIO 输出低电平,而 Y+,Y-为 GPIO 带内部上拉输入,并允许下降沿中断。无触摸时,CPU 没有任何动作且没有任何功耗在电阻触屏。当用户触屏时,Y+,Y-变为低并触发中断。在该中断处理程序中,我们需把 X+端口置为高电平,同时把 Y+,Y-设为 ADC 输入。在检测到 Y+的电压值(该电压值对应触点 X 轴位置)后,把 Y+,Y-设置为 GPIO 输出,且 Y+为高,Y-为低,X+,X-设为 ADC 输入并对 X+进行检测。检测到的电压值对应触点 Y 轴位置。在检测结束后,恢复 X+,X-设置为 GPIO 输出低电平,而 Y+,Y-为 GPIO 带内部上拉输入,并允许下降沿中断。

注意,在对 X,Y 进行位置 ADC 检测时,由于存在高低电平转换,需注意 ADC 检测必须在电压信号稳定后进行,所以相应的延迟是必须的。

另外,用户在使用电阻触屏时,需加入位置校对。



检测方法示意图

## 程序例程

本例程用到下列 [Xinnova](#) MCU 库函数, 请参看 [Xinnova](#) 网页和论坛。

Xn12lxxx.h: device description file

Xn\_adc.h: ADC h file

Xn\_adc.c: ADC driver file

Xn\_uart.h: UART h file

Xn\_UART: UART driver file

```
/******
```

```
// PIO1_0 (AD2) -> X-
```

```
// PIO1_1 (AD3) -> Y-
```

```
// PIO1_2 (AD4) -> X+
```

```
// PIO1_3 (AD5) -> Y+
```

```
**
```

```
*****/
```

```
uint32_t x,y;
```

```
void SetTouchIO(void)
```

```
{
```

```
//init Touch
```

```
XN_IOCON->PIO1_0 = 0x91; //x-
```

```
XN_IOCON->PIO1_1 = 0x91; //y-
```

```
XN_IOCON->PIO1_2 = 0x90; //x+
```

```
XN_IOCON->PIO1_3 = 0x90; //y+
```

```
XN_GPIO1->DIR = 0x05; //set x-,x+ as output
```

```
XN_GPIO1->OUT = 0; //x-,x+ set to low
```

```
delays(100);
```

```
XN_GPIO1->IBE = 0x02; //沿触发中断
```

## 低成本低功耗 4 线电阻触屏 MCU 实现方案

```
//XN_GPIO1->IC = 0x02; //清除中断
```

```
//XN_GPIO1->IE = 0x02;
```

```
//NVIC_EnableIRQ(EINT1_IRQn);
```

```
}
```

```
uint32_t TouchPressed(void)
```

```
{
```

```
//verify if y is low
```

```
if ((XN_GPIO1->PIN & 0x2)==0)
```

```
    return 1;
```

```
else
```

```
    return 0;
```

```
}
```

```
void ReadXY(uint32_t * x,uint32_t* y)
```

```
{
```

```
uint32_t temp;
```

```
*x=0;
```

```
*y=0;
```

```
//Read x
```

```
SetADCPin(AD3PIN|AD5PIN);
```

```
XN_IOCON->PIO1_0 = 0x91; //x-
```

```
XN_IOCON->PIO1_2 = 0x90; //x+
```

```
XN_GPIO1->MASK = 0xFFFFFFFF;
```

```
XN_GPIO1->DIR = 0x05;

XN_GPIO1->OUT = 0x04; //X+ 3.3v;X- GND

//SetADCPin(AD3PIN|AD5PIN);

ADCInit(XN_ADC0,10000,TRIGGERMODE,START_BY_SOFTWAER, 0); //?K conversion rate and software trigger

ADCSeIChannel(XN_ADC0,((AD5<<DR0SEL)|(AD5<<DR1SEL)|(AD5<<DR2SEL)|(AD5<<DR3SEL)|(AD5<<DR4SEL)|(AD
5<<DR5SEL)|(AD5<<DR6SEL)|(AD5<<DR7SEL)));

delayms(100);

ADCSoftwareTrigger(XN_ADC0);

do
{
    temp=ADCValue(XN_ADC0,DR7);
}while(temp==0xFFFFFFFF);

*x =temp;

//Read y
SetADCPin(AD2|AD4);
XN_IOCON->PIO1_1 = 0x91; //y-
XN_IOCON->PIO1_3 = 0x90; //y+

XN_GPIO1->MASK = 0xFFFFFFFF0;

XN_GPIO1->DIR = 0x0A;

XN_GPIO1->OUT = 0x02; //y+ 3.3v;y- GND

//SetADCPin(AD2|AD4);
```

## 低成本低功耗 4 线电阻触屏 MCU 实现方案

```
ADCSetChannel(XN_ADC0,(AD2<<DR0SEL)|(AD2<<DR1SEL)|(AD2<<DR2SEL)|(AD2<<DR3SEL)|(AD2<<DR4SEL)|(AD2<<DR5SEL)|(AD2<<DR6SEL)|(AD2<<DR7SEL));
```

```
delayms(100);
```

```
ADCSoftwareTrigger(XN_ADC0);
```

```
do
```

```
{
```

```
temp=ADCValue(XN_ADC0,DR7);
```

```
}while(temp==0xFFFFFFFF);
```

```
*y =temp;
```

```
}
```

```
int main(void)
```

```
{
```

```
//disable watchdog
```

```
XN_WDT->MOD = 0;
```

```
//init Touch
```

```
SetTouchIO();
```

```
XN_GPIO1->IC = 0x02; //清除中断
```

```
XN_GPIO1->IE = 0x02; //enable INT
```

```
NVIC_EnableIRQ(EINT1_IRQn);
```

```
while (1)
```

```
{
```

```
// Loop forever
```



```
}  
  
}  
  
void EINT1_IRQHandler(void)  
  
{  
  
    NVIC_DisableIRQ(EINT1_IRQn);  
  
    XN_GPIO1->IE = 0x0; //屏蔽 GPIO1 IO 中断  
  
    if (TouchPressed())  
  
    {  
  
        ReadXY(&touch_x,&touch_y);  
  
        SetTouchIO();  
  
        if (TouchPressed())  
  
        {  
  
            touch_xy_new=1;  
            UART0PutHex (touch_x);  
            UART0Send(", ",1);  
            UART0PutHex (touch_y);  
            UART0Send(" ",4);  
  
        }  
  
    }  
  
}  
  
XN_GPIO1->IC = 0x02; //清除中断
```

## 低成本低功耗 4 线电阻触屏 MCU 实现方案

---

```
XN_GPIO1->IE = 0x02; //enable INT
```

```
NVIC_EnableIRQ(EINT1_IRQn);
```

```
return;
```

```
}
```

### 总结

[Xinnova](#) MCU 引脚复用,配置灵活。在应对 4 线电阻触屏应用中占用较少资源,使用简单。该方法在 EBike 显示屏被采用,表现稳定。