



基于 C#语言的 GTS 控制卡

初级入门手册
(点位运动示例)

目录

一、如何识别控制卡和端子板型号.....	3
1、包装盒.....	3
2、控制卡和端子板标签.....	4
二、安装控制卡和驱动.....	6
1、安装控制卡和驱动.....	6
三、编程.....	8
1、打开 VS2010	8
2、点击新建项目	9
3、导入库文件.....	10
4、添加头文件.....	11
5、配置 MCT2008.....	12
6、生成 MCT2008 配置文件	13
7、导入 MCT2008 文件	14
8、添加控件.....	16
9、更改控件属性.....	16
10、进入代码编辑页面.....	18
11、调试运行.....	22

一、如何识别控制卡，端子板型号

方法 1、包装盒,如图 1



图 1

方法 2、控制卡和端子板标签，如图 2，如图 3



图 2



图 3

注：GT2-800-ACC2-VB-G-A 具体型号说明请参考用户手册

二、安装控制卡和驱动

1、打开主机后盖，将卡插到 PCI 插槽上，然后右击我的电脑，点击属性，打开设备管理器，会看到 PCI 设备上有个感叹号，然后右击，弹出更新驱动程序软件，如图 4，如图 5，如图 6，如图 7，如图 8 所示

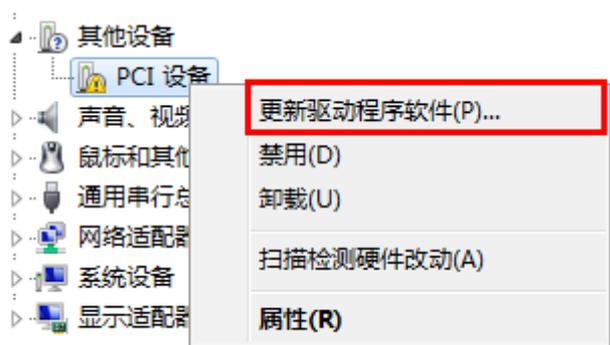


图 4

您想如何搜索驱动程序软件？

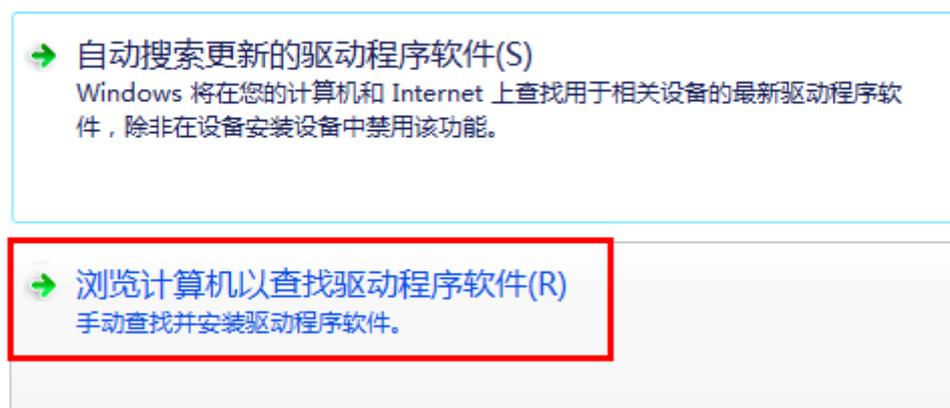


图 5

浏览计算机上的驱动程序文件

在以下位置搜索驱动程序软件:

包括子文件夹(I)

- ➔ 从计算机的设备驱动程序列表中选择(L)
此列表将显示与该设备兼容的已安装的驱动程序软件，以及与该设备处于同一类别下的所有驱动程序软件。

图 6

Windows 已经成功地更新驱动程序文件

Windows 已经完成安装此设备的驱动程序软件:



图 7

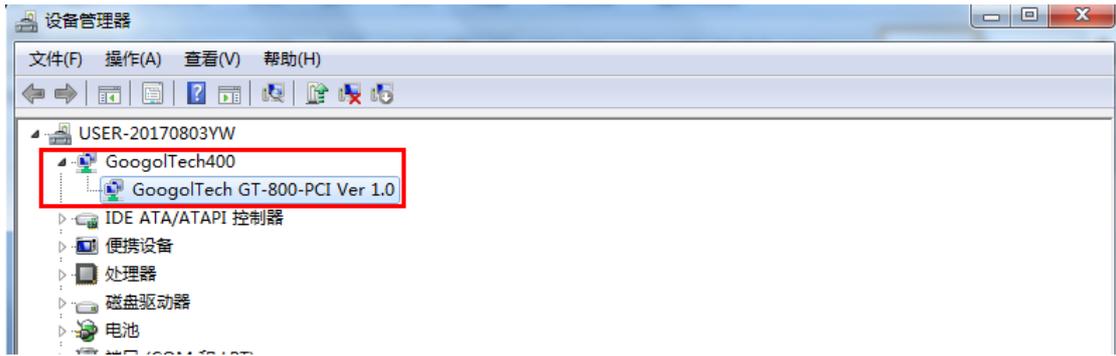


图 8

注：GTS400 和 GTS800 控制卡安装完驱动以后，都会显示 GoogolTech GT-800-PCI Ver 1.0

三、驱动安装完以后，开始编程。

1、首先打开 VS2010，如图 9 所示。

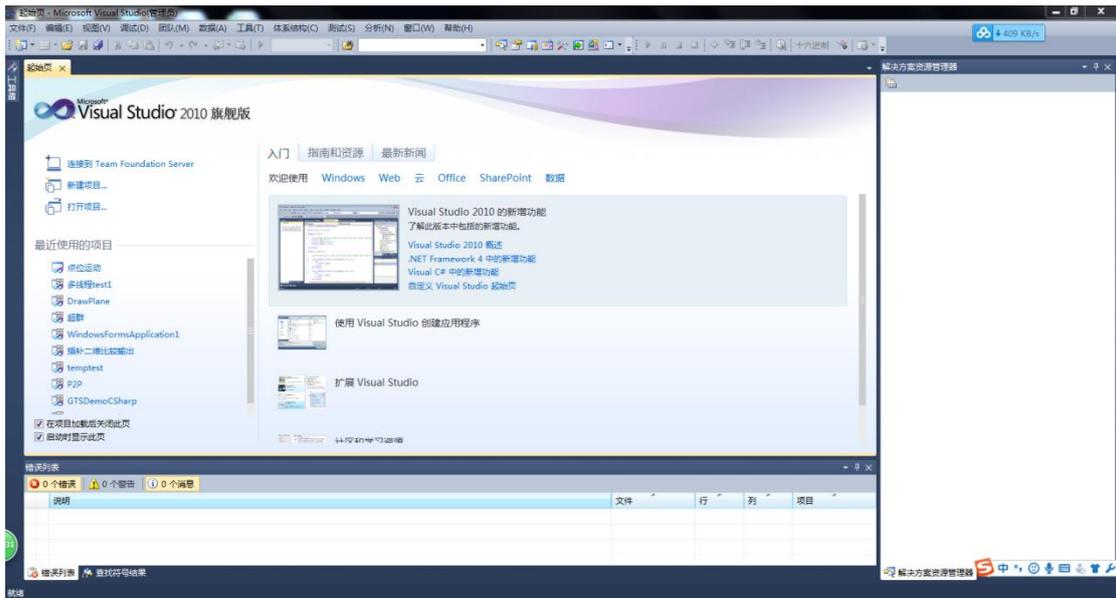


图 9

2、然后点击新建项目，选择 **Visual C#**，然后选择创建 **Windows** 窗体应用程序，将名称改为点位运动，然后点击确定，如图 10，如图 11。

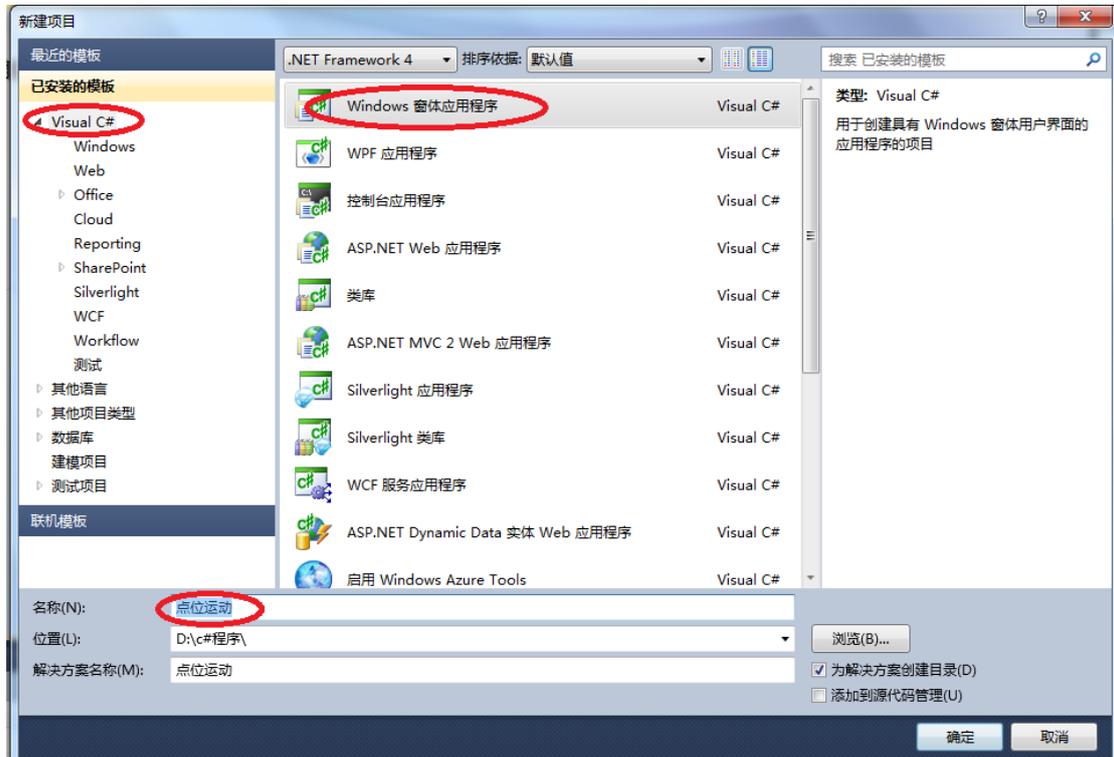


图 10

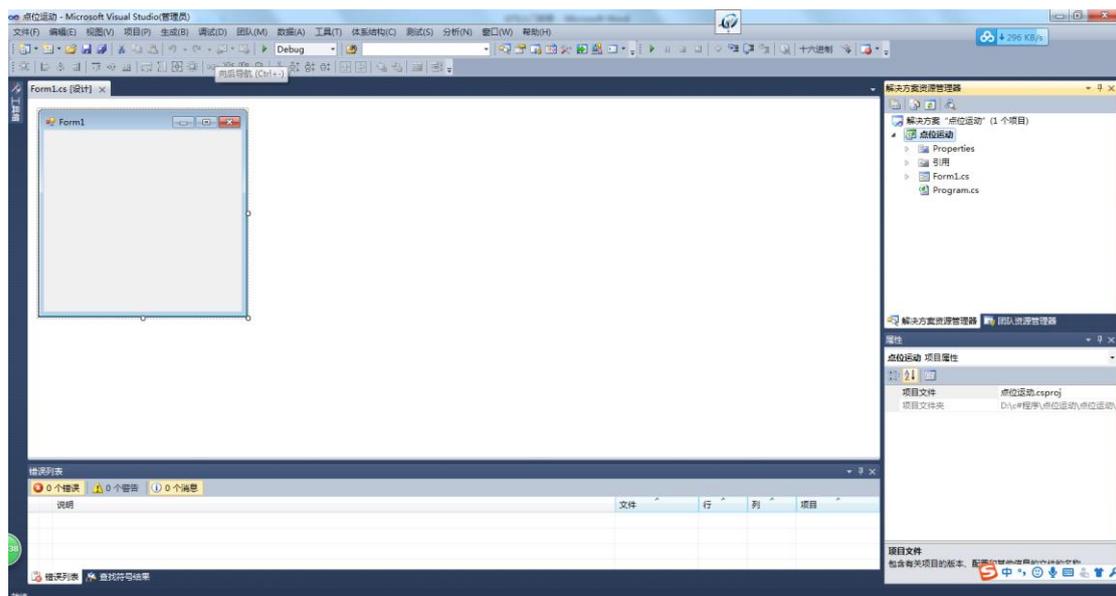


图 11

3、找到 gts 卡的 C#库文件(D:\资料\固高\VB 版\dll\C#)，
先将 C#文件夹下的 gts.dll 复制到刚才创建的 VS2010 C#工
程文件的最底层 Debug 文件夹 (D:\c#程序\点位运动\点位
运动\bin\Debug) 下，如图 12

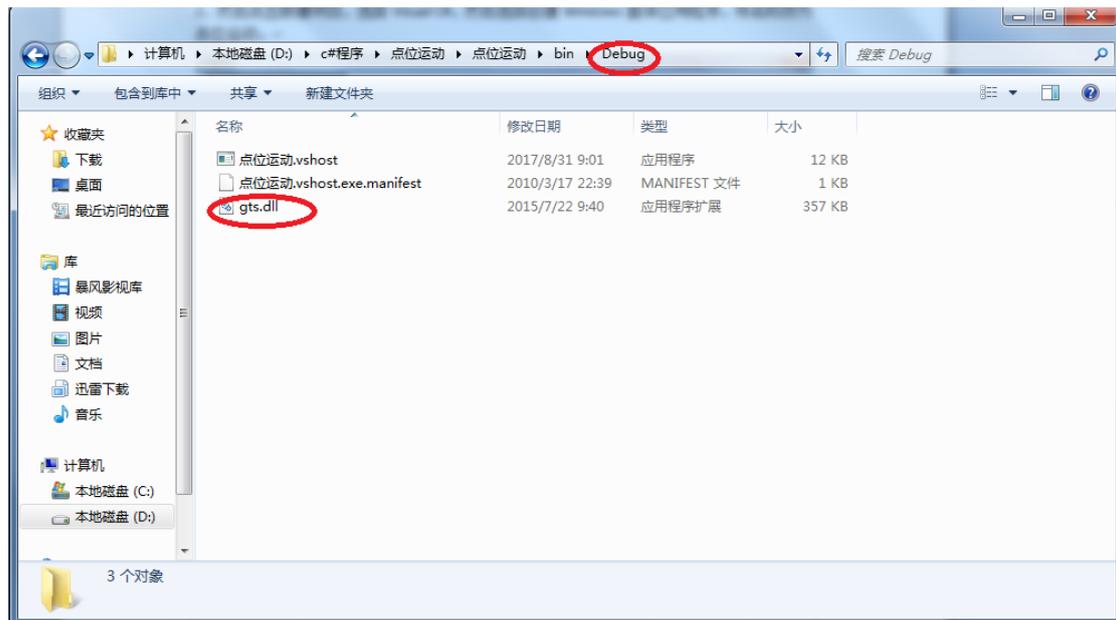


图 12

4、然后再将 `gts.cs` 复制到(D:\c#程序\点位运动\点位运动) 文件夹下，如图 13

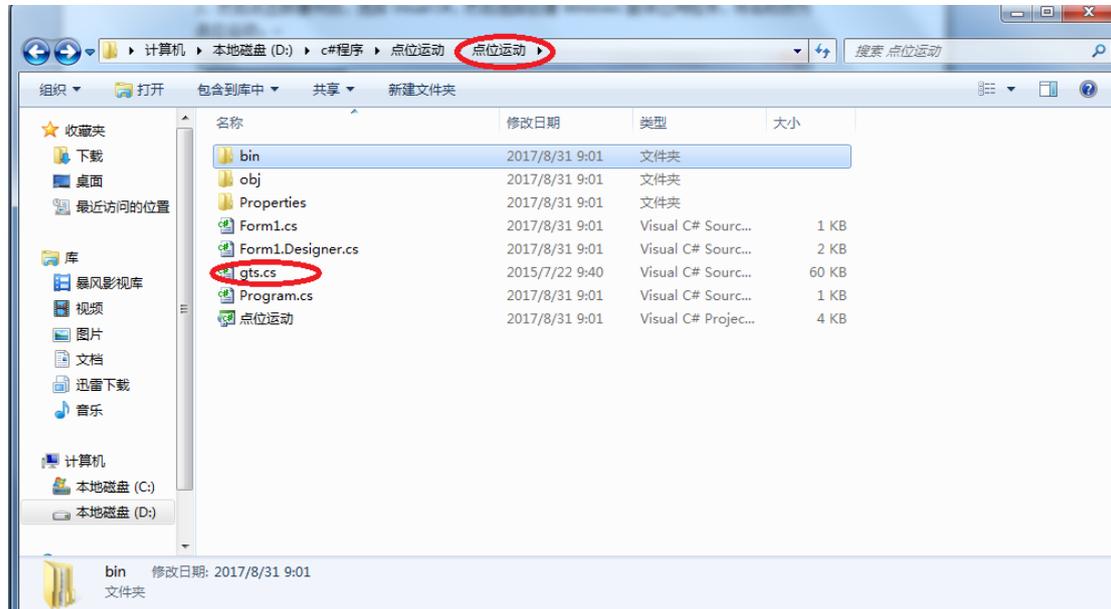


图 13

5、打开 mct2008，选择工具→控制器配置，正、负限位选择 none，如图 14



图 14

6、然后点击控制→写入控制器，然后点击文件→写入到文件，将文件名改成 GTS800，保存到桌面，如图 15

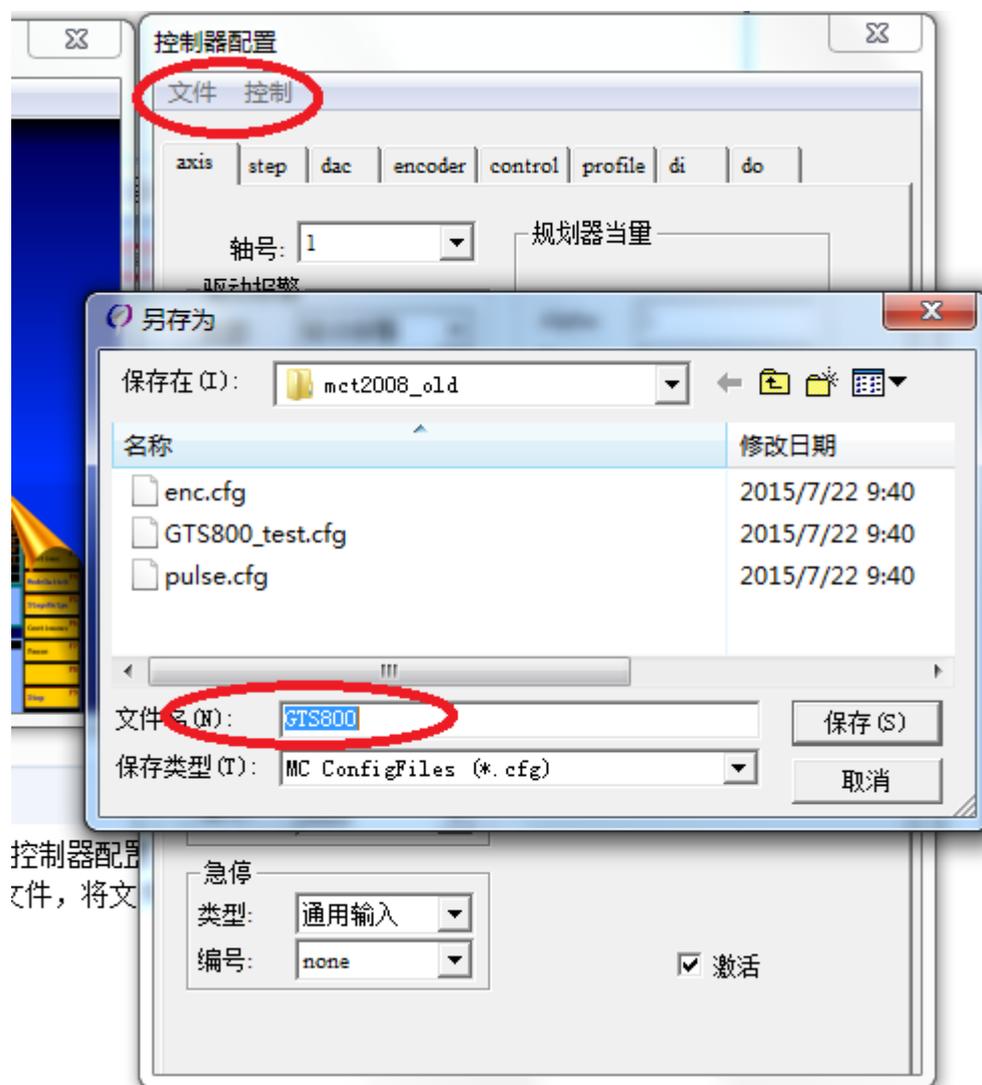


图 15

7、把桌面生成的 GTS800 文件复制到（D:\c#程序\点位运动\点位运动\bin\Debug）文件下，如图 16

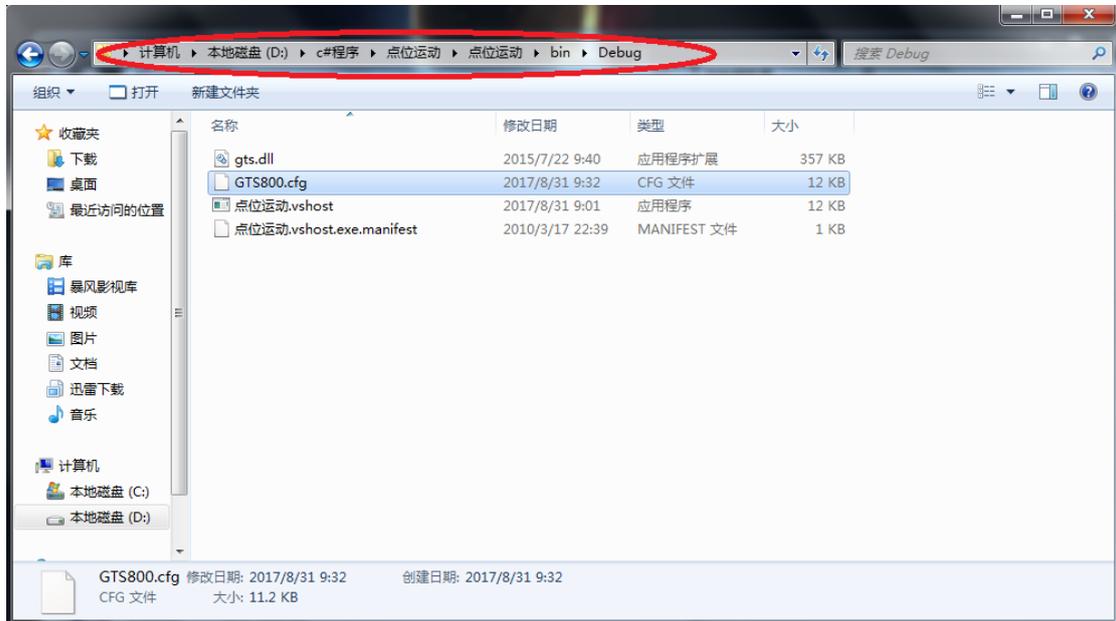


图 16

8、打开刚才创建的 vs2010 程序，在右侧的解决方案管理器中右击点位运动→添加→现有项，然后选择 gts.cs，再点击添加，如图 17，如图 18

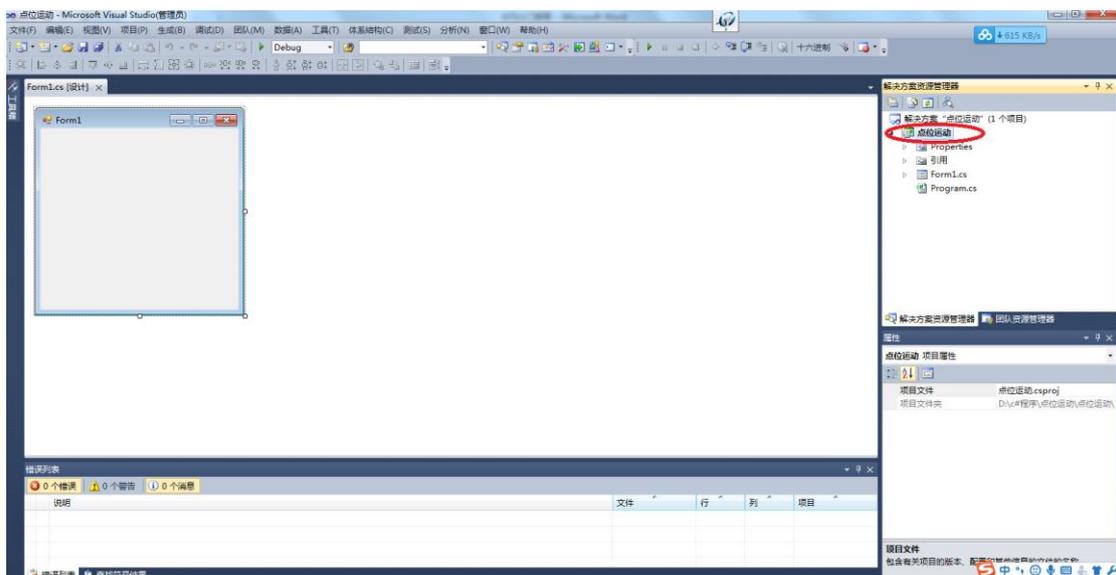


图 17

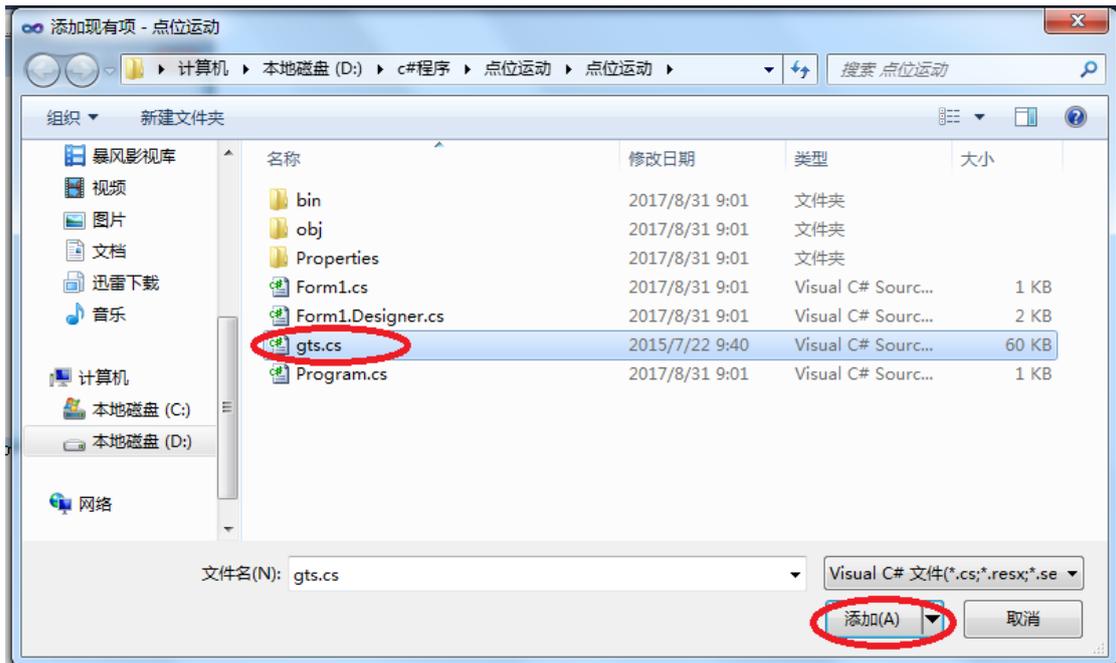


图 18

9、添加完以后，右侧解决方案管理器中会显示刚才添加的 `gts.cs` 文件，如图 19

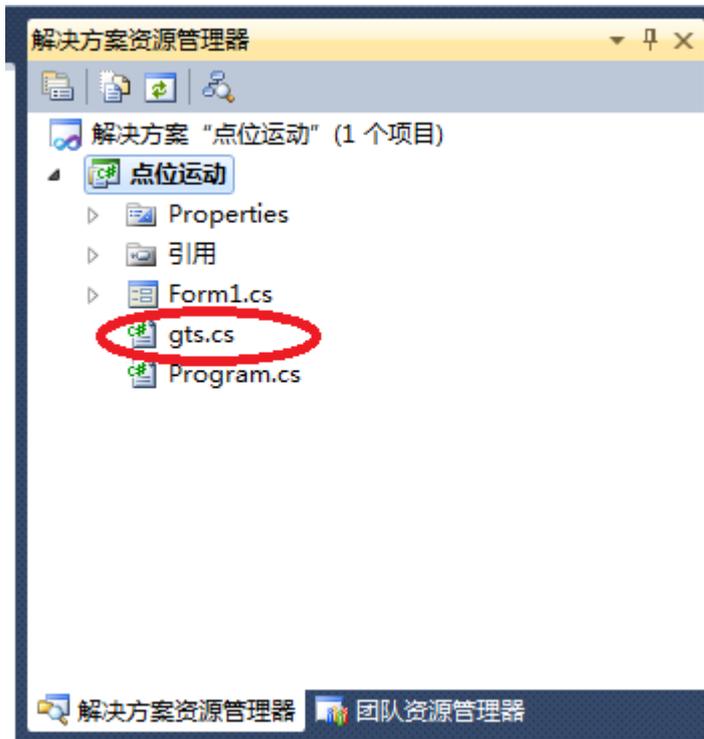


图 19

10、下一步开始编程，在左侧工具箱中选中 **button** 控件，然后添加到 **form1** 窗体中，如图 20

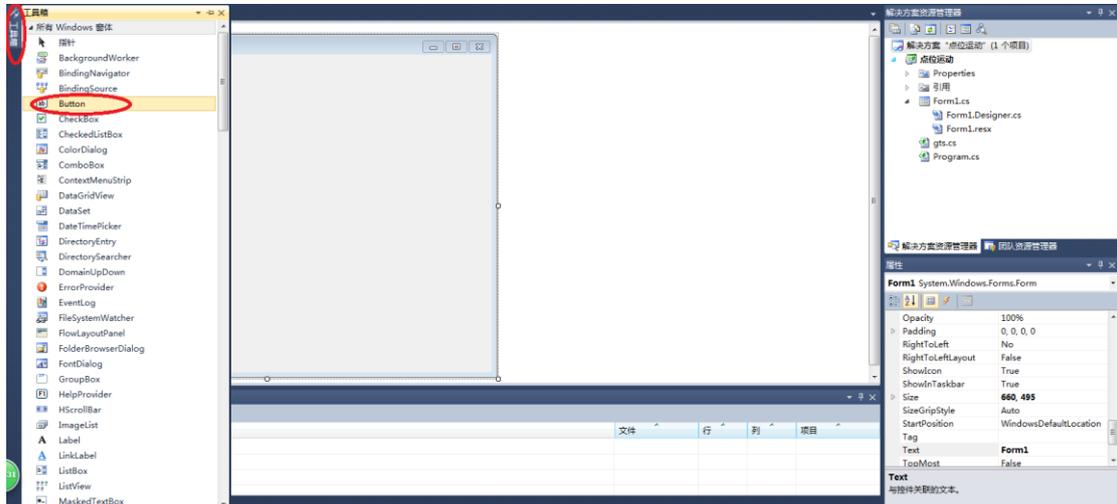


图 20

11、然后选中 **button1**，在右下角的属性窗口中，选择 **text** 属性，将其改为“初始化”，如图 21

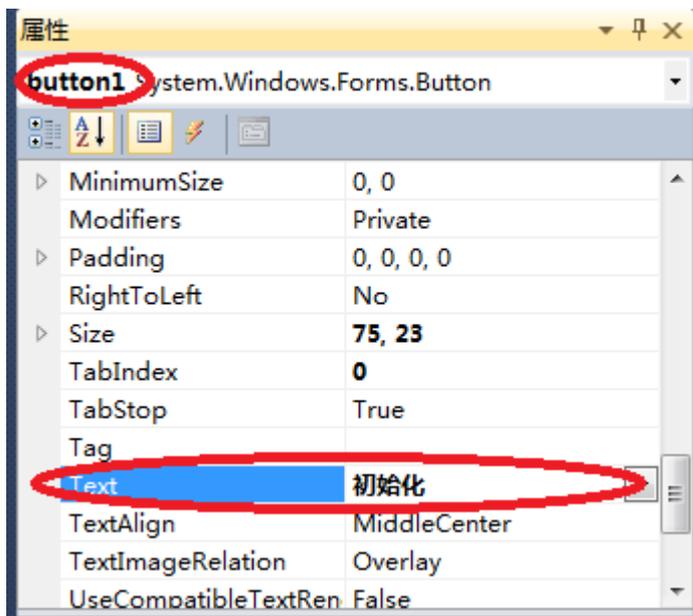


图 21

12、然后在工具箱中选择 label, textbox, timer 控件放到 form1 中，并将其 text 属性改成图中所示名称，如图 22



图 22

13、然后双击初始化按钮，会自动进入代码编辑页面，如图 23

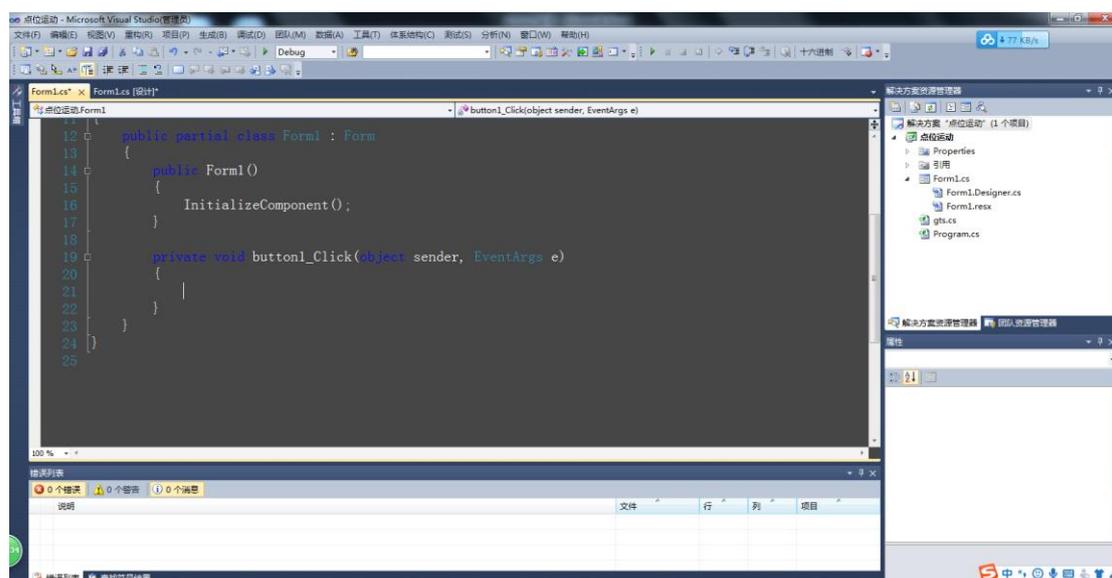


图 23

14、然后依次双击所有按钮以及 timer1，如图 24

```
11 |  
12 | public partial class Form1 : Form  
13 | {  
14 |     public Form1()  
15 |     {  
16 |         InitializeComponent();  
17 |     }  
18 |  
19 |     private void button1_Click(object sender, EventArgs e)  
20 |     {  
21 |  
22 |     }  
23 |  
24 |     private void button2_Click(object sender, EventArgs e)  
25 |     {  
26 |  
27 |     }  
28 |  
29 |     private void button3_Click(object sender, EventArgs e)  
30 |     {  
31 |  
32 |     }  
33 |  
34 |     private void button4_Click(object sender, EventArgs e)  
35 |     {  
36 |  
37 |     }  
38 |  
39 |     private void button5_Click(object sender, EventArgs e)  
40 |     {  
41 |  
42 |     }  
43 |  
44 |     private void button6_Click(object sender, EventArgs e)  
45 |     {  
46 |  
47 |     }  
48 |  
49 |     private void timer1_Tick(object sender, EventArgs e)  
50 |     {
```

图 24

先定义全局变量，如图 25

```

public Form1()
{
    InitializeComponent();
}
bool flag=false;
short rtn;
int pos, vel;
uint clk;
double prfpos, prfvel, encpos, encvel;
private void button1_Click(object sender, EventArgs e)

```

图 25

在 private void button1_click(object sender,EventArgs e)

```

{
}

```

中输入以下代码，如图 26

```

private void button1_Click(object sender, EventArgs e)
{
    rtn = gts.mc.GT_Open(0, 1); //打开控制卡
    rtn = gts.mc.GT_Reset(); //控制卡复位
    rtn = gts.mc.GT_LoadConfig("GTS800.cfg"); //加载配置文件
    rtn = gts.mc.GT_ClrSts(1, 8); //清除状态
    timer1.Enabled = true; //开启定时器
}

```

图 26

timer1.start();

在 private void button2_click(object sender,EventArgs e)

```

{
}

```

中输入以下代码，如图 27

```

private void button2_Click(object sender, EventArgs e)
{
    rtn = gts.mc.GT_ClrSts(1, 8); //清除状态
}

```

图 27

在 private void button3_click(object sender,EventArgs e)

```

{
}

```

中输入以下代码，如图 28

```
private void button3_Click(object sender, EventArgs e)
{
    if (flag == false) //判断伺服标志位
    {
        gts.mc.GT_AxisOn(1); //开伺服
        button3.Text = "伺服开启";
        flag = !flag;
    }
    else if (flag == true)
    {
        gts.mc.GT_AxisOff(1);
        button3.Text = "伺服关闭"; //关伺服
        flag = !flag;
    }
}
```

图 28

在 private void button4_click(object sender,EventArgs e)
{
}

中输入以下代码，如图 29

```
private void button4_Click(object sender, EventArgs e)
{
    gts.mc.GT_ZeroPos(1, 8); //位置清零
}
```

图 29

在 private void button5_click(object sender,EventArgs e)
{
}

中输入以下代码，如图 30

```

private void button5_Click(object sender, EventArgs e)
{
    rtn = gts.mc.GT_PrflTrap(1); //设置当前轴为点位运动模式
    gts.mc.TTrapPrm trapprm; //定义点位运动结构体变量
    gts.mc.GT_GetTrapPrm(1, out trapprm);
    trapprm.acc = 0.1; //设置加速度
    trapprm.dec = 0.1; //设置减速度
    trapprm.smoothTime = 1; //设置平滑时间
    gts.mc.GT_SetTrapPrm(1, ref trapprm); //设置点位运动参数
    pos += Convert.ToInt32(textBox1.Text);
    vel = Convert.ToInt32(textBox2.Text);
    gts.mc.GT_SetPos(1, pos); //设置点位距离
    gts.mc.GT_SetVel(1, vel); //设置点位速度
    gts.mc.GT_Update(1); //开启点位运动
}

```

图 30

在 private void button6_click(object sender,EventArgs e)

```

{
}

```

中输入以下代码，如图 31

```

private void button6_Click(object sender, EventArgs e)
{
    gts.mc.GT_Stop(1, 0); //停止运动
}

```

图 31

在 private void timer_Tick(object sender,EventArgs e)

```

{
}

```

中输入以下代码，如图 32

```

private void timer1_Tick(object sender, EventArgs e)
{
    gts.mc.GT_GetPrfPos(1, out prfpos, 1, out clk); //读取规划位置
    gts.mc.GT_GetEncPos(1, out encpos, 1, out clk); //读取实际位置
    textBox3.Text = prfpos.ToString();
    textBox5.Text = encpos.ToString();
    gts.mc.GT_GetPrfVel(1, out prfvel, 1, out clk); //读取规划速度
    gts.mc.GT_GetEncVel(1, out encvel, 1, out clk); //读取实际速度
    textBox4.Text = prfvel.ToString();
    textBox6.Text = encvel.ToString();
}

```

图 32

15、检查代码没有错误后，开始调试运行按 F5，或者点击
如图所示按钮，如图 33

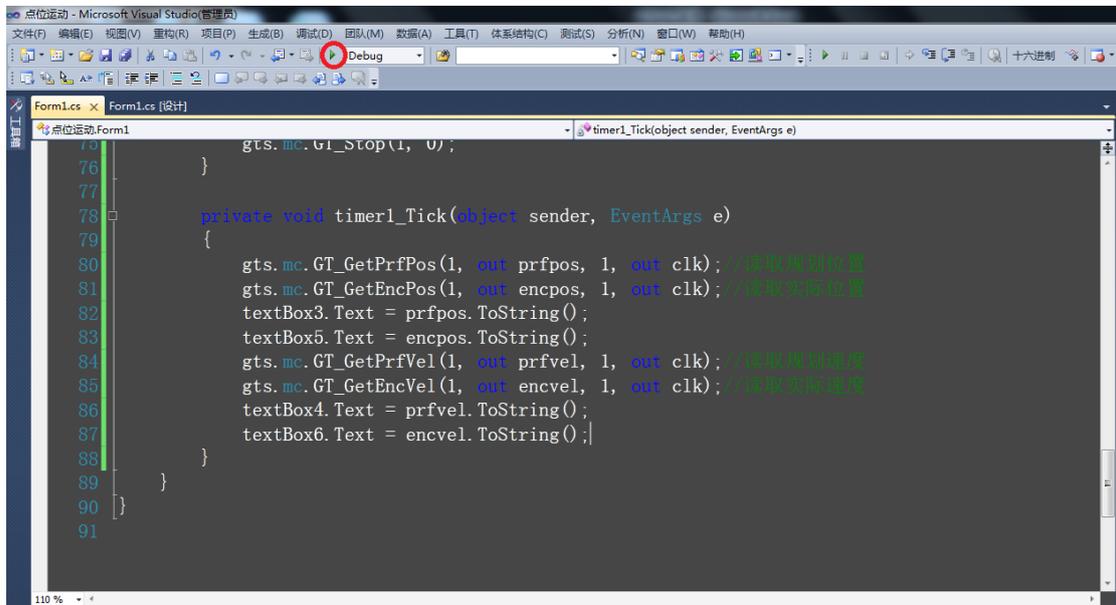


图 33

16、输入点位距离和速度，然后点击开始运动按钮，如图 34



图 34

17、运动过程中，规划位置和实际位置会实时变化，点击停止按钮可停止点位运动。点击 X 号可以关闭整个调试程序。