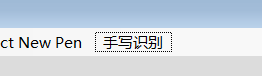
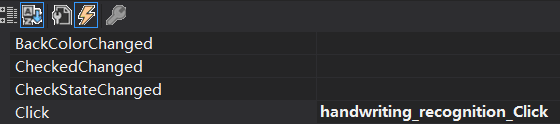
1. 实现步骤：

1、在FormMain窗体

### 1.1、添加手写识别菜单按键



### 1.2、添加点击事件handwriting\_recognition\_Click



### 1.3、定义全局变量gHandwriting\_recognition\_Form

代码：

public static Handwriting\_recognition\_Form gHandwriting\_recognition\_Form;

### 1.4、在handwriting\_recognition\_Click()函数中添加代码代码

代码：

int i, j;

int k = 0;

float[][] x = new float[stroke\_c[gCurPageID]][];

float[][] y = new float[stroke\_c[gCurPageID]][];

for (i = 0; i < stroke\_c[gCurPageID]; i++) //遍历笔画数

{

x[i] = new float[stroke\_ep[gCurPageID][i] - stroke\_sp[gCurPageID][i] + 1];

y[i] = new float[stroke\_ep[gCurPageID][i] - stroke\_sp[gCurPageID][i] + 1];

k = 0;

for (j = stroke\_sp[gCurPageID][i]; j <= stroke\_ep[gCurPageID][i]; j++) //遍历每一笔的点数

{

x[i][k] = buf1\_x[gCurPageID][j];

y[i][k++] = buf1\_y[gCurPageID][j];

}

}

gHandwriting\_recognition\_Form = new Handwriting\_recognition\_Form(x, y);

gHandwriting\_recognition\_Form.ShowDialog();

2、在Handwriting\_recognition\_Form窗体

### 2.1、添加设置textbox控件

拖动添加：



属性：





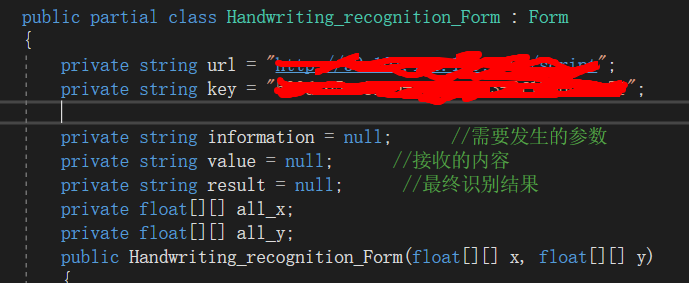








### 2.2、在Handwriting\_recognition\_Form中定义全局变量



代码：

private string url = "\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

private string key = "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

private string information = null; //需要发生的参数

private string value = null; //接收的内容

private string result = null; //最终识别结果

private float[][] all\_x;

private float[][] all\_y;

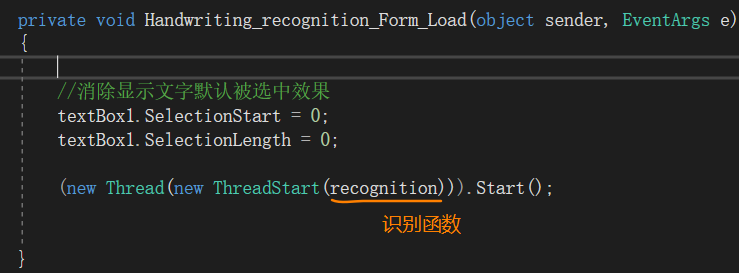
### 2.3、在Handwriting\_recognition\_Form()函数中添加代码

代码：

all\_x = x;

all\_y = y;

### 2.5、在Handwriting\_recognition\_Form\_Load ()函数中添加代码



代码：

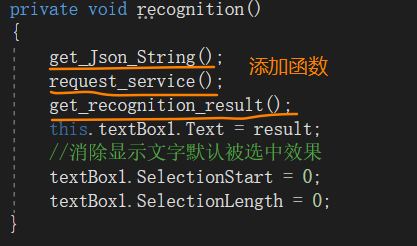
//消除显示文字默认被选中效果

textBox1.SelectionStart = 0;

textBox1.SelectionLength = 0;

(new Thread(new ThreadStart(recognition))).Start();

### 2.6、添加recognition()识别函数



代码：

private void recognition()

{

get\_Json\_String();

request\_service();

get\_recognition\_result();

this.textBox1.Text = result;

//消除显示文字默认被选中效果

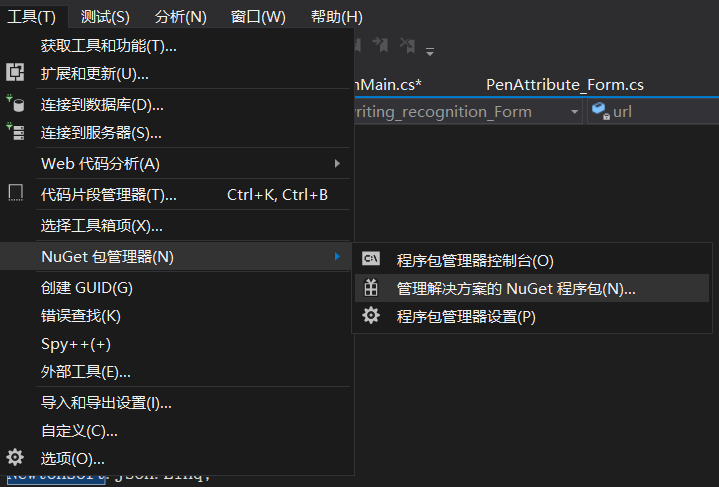
textBox1.SelectionStart = 0;

textBox1.SelectionLength = 0;

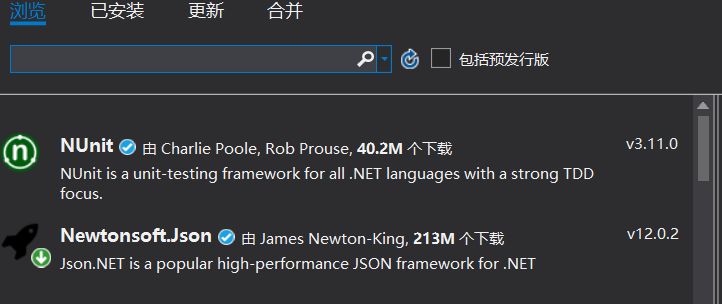
}

### 2.7、添加Newtonsoft.Json包

打开NuGet



搜索Newtonsoft.Json并安装



### 2.8、添加get\_Json\_String()Json数据生成函数

代码：

private void get\_Json\_String()

{

JObject josn = new JObject();

try

{

josn.Add("viewSizeHeight", 1527);

josn.Add("viewSizeWidth", 1085);

josn.Add("applicationKey", key);

josn.Add("scriptType", "Text");

josn.Add("languages", "zh\_CN");

JArray jsonArray = new JArray();

if (all\_x != null && all\_x.Count() > 0)

{

JObject jo\_Point = null;

string x = "";

string y = "";

for (int i = 0; i < all\_x.Count(); i++)

{

for (int j = 0; j < all\_x[i].Count(); j++)

{

if (j == 0)

{

jo\_Point = new JObject();

x += all\_x[i][j];

y += all\_y[i][j];

}

else

{

x += "," + all\_x[i][j];

y += "," + all\_y[i][j];

}

}

jo\_Point.Add("x", x);

jo\_Point.Add("y", y);

jsonArray.Add(jo\_Point);

x = "";

y = "";

}

}

josn.Add("penData", jsonArray);

}

catch(Exception ex)

{

}

information = josn.ToString();

}

### 2.9、添加request\_service ()请求服务函数

代码：

private void request\_service()

{

if (information != null && !information.Equals(""))

{

try

{

System.Net.HttpWebRequest request;

//创建连接

request = (System.Net.HttpWebRequest)WebRequest.Create(new Uri(url));

//无缓存接收

HttpRequestCachePolicy noCachePolicy = new HttpRequestCachePolicy(HttpRequestCacheLevel.NoCacheNoStore);

request.Method = "POST";

request.Accept = "\*/\*";

request.ContentType = "application/json;charset=UTF-8";

request.CachePolicy = noCachePolicy;

//不需要保持长时间连接

request.KeepAlive = false;

byte[] bt = Encoding.UTF8.GetBytes(information);

request.ContentLength = bt.Length;

Stream postStream = request.GetRequestStream();

postStream.Write(bt, 0, bt.Length);

postStream.Dispose();

System.Net.HttpWebResponse response = (System.Net.HttpWebResponse)request.GetResponse();

StreamReader read = new StreamReader(response.GetResponseStream(), Encoding.UTF8);

value = read.ReadToEnd();

read.Close();

request.Abort();

response.Close();

}

catch(Exception ex)

{

}

}

}

### 2.10、添加get\_recognition\_result ()获取识别结果

代码：

private void get\_recognition\_result()

{

result = "";

if(value!=null&&!value.Trim().Equals(""))

{

try

{

JObject json\_object = JObject.Parse(value);

result = json\_object.GetValue("data").ToString();

//Textbox不识别"\n"换行符

result = result.Replace("\n", "\r\n");

}

catch(Exception ex)

{

}

}

}

二、参考资料：

HTTP的C#和java对照使用：

https://blog.csdn.net/Mavolas/article/details/84592497

三、改进建议：

因sdk对笔坐标数据接收处理问题，服务器会将一些笔写下的点识别成字母“i”；后期可以适当进行排版优化。