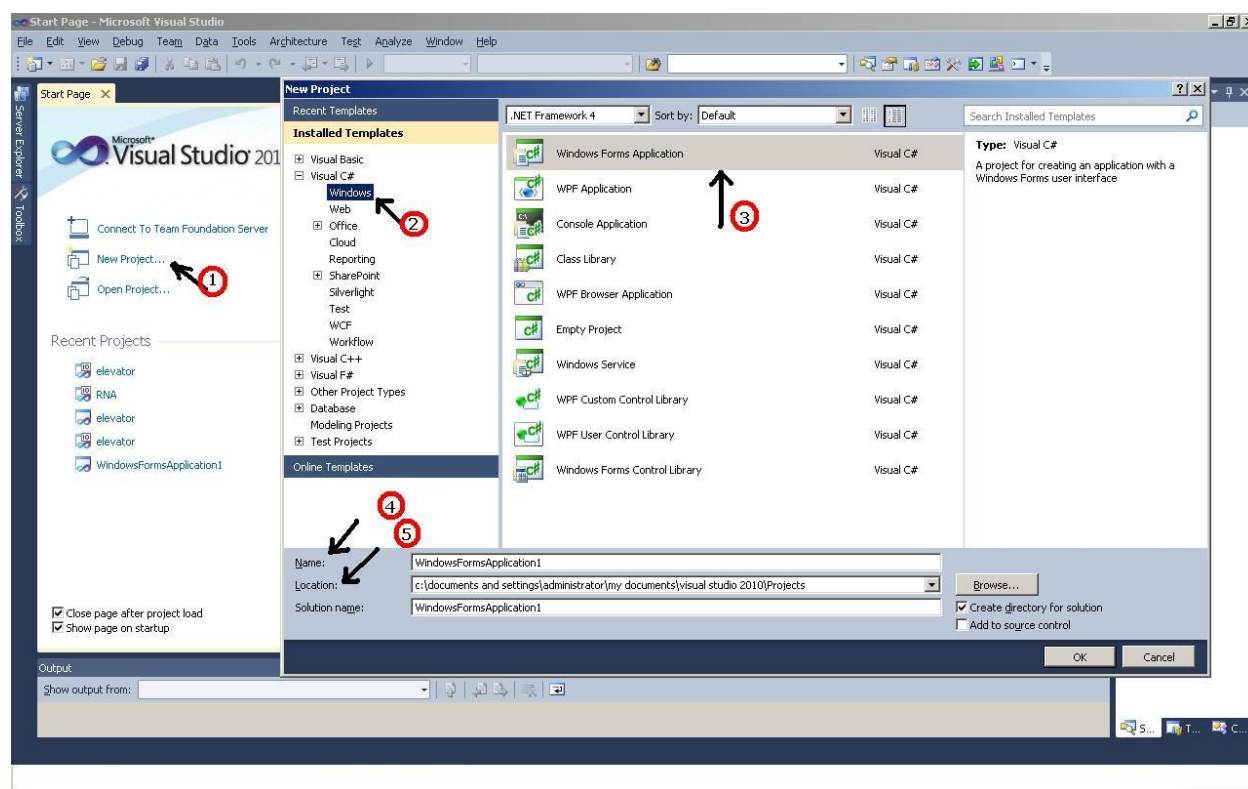


نحوه ارتباط یک برنامه Microsoft Visual Studio با میکروکنترلر

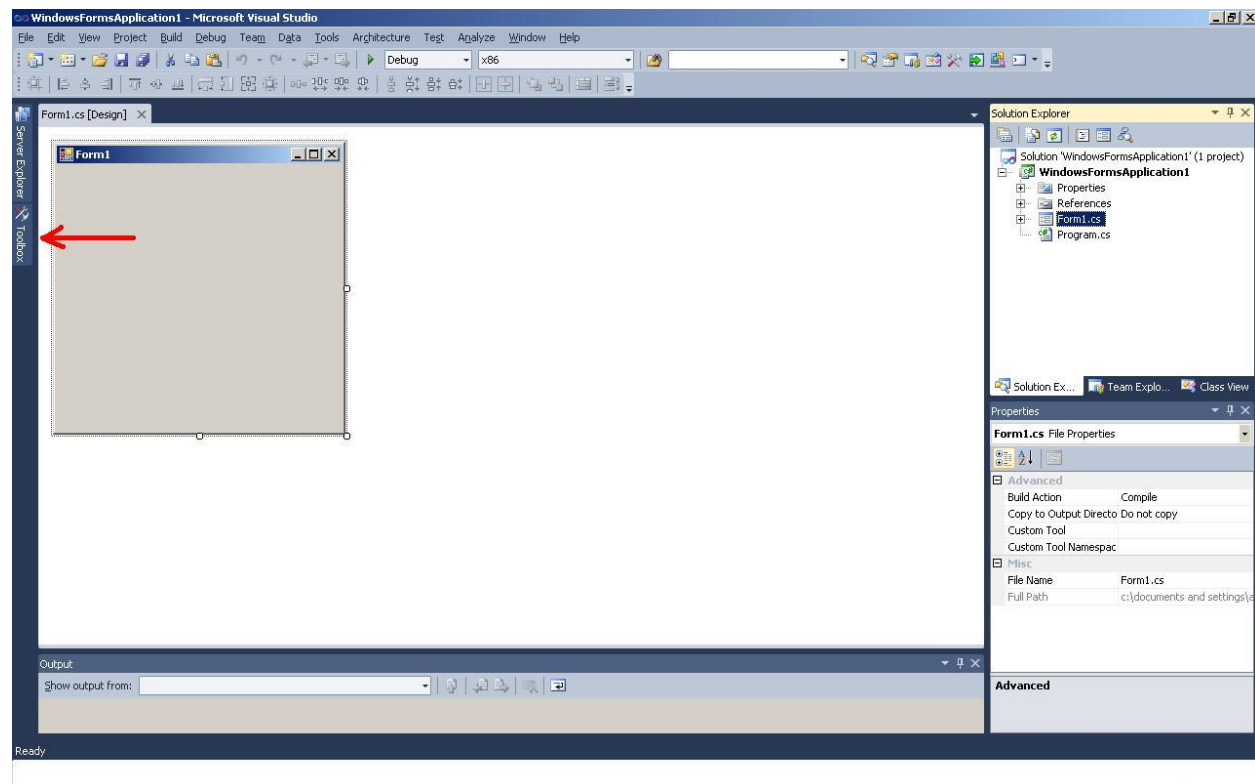
به منظور اینکه داده هایی مابین میکروکنترلر و یک محیط نرم افزاری رد و بدل شوند می توان از Microsoft Visual Studio استفاده نمود. به طور خاص تر می توانیم از زبان برنامه نویسی C# در این محیط نرم افزاری استفاده نماییم. نحوه ایجاد یک پروژه C# در Visual Studio 2010 و یک برنامه ساده که برای ارتباط با میکروکنترلر باید نوشته شود، در ادامه بیان می شود.

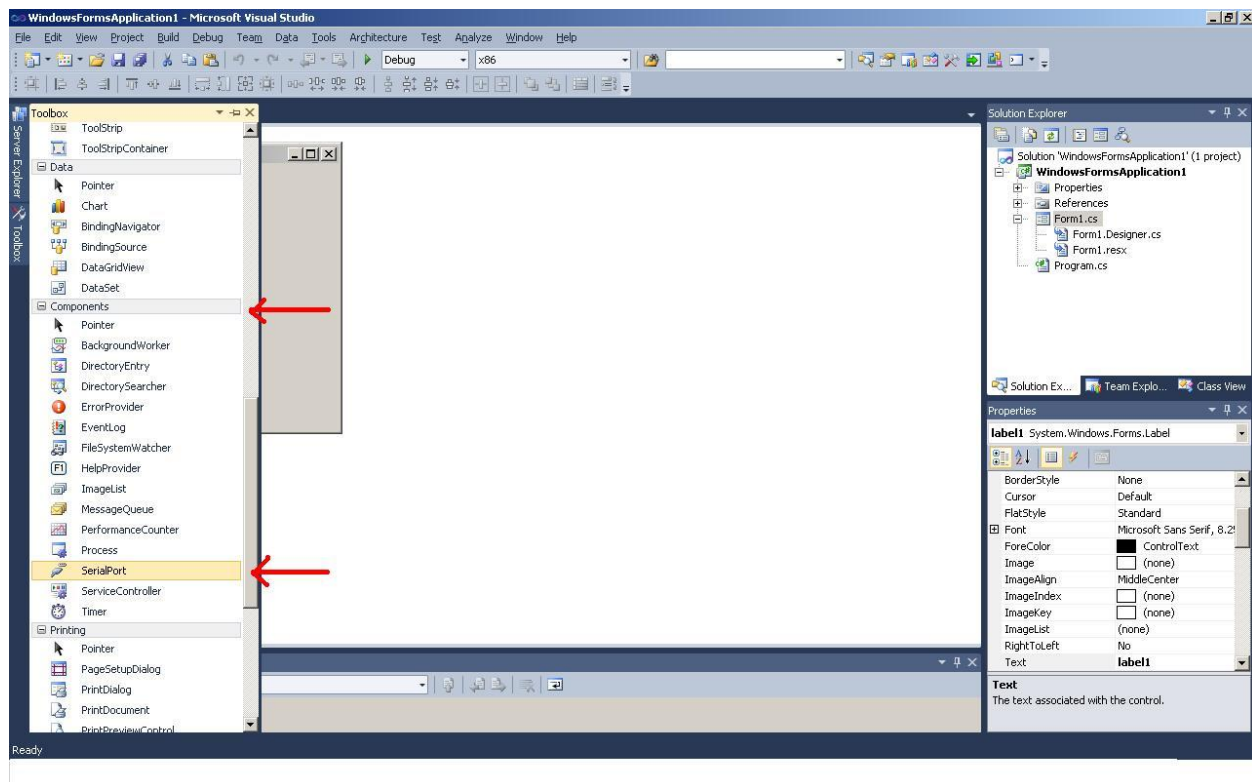
برای برقراری ارتباط همزمان یا غیر همزمان مابین واحد پردازشگر مرکزی و دستگاه های جانبی از واسط ارسال و دریافت سریال همزمان-غیرهمزمان USART استفاده می شود. برای این که برنامه نرم افزاری بتواند با میکرو ارتباط برقرار کند باید امکانی داشته باشد که بتواند با پورت سریال ارتباط برقرار کند. به این منظور از یک component پورت سریال در برنامه استفاده می کنیم.

Visual Studio 2010 را اجرا کنید. در لبه Start Page (همان پنجره آغازین)، New Project... را انتخاب کرده و در پنجره باز شده، زبان برنامه نویسی را روی Visual C# قرار داده و نوع Windows را انتخاب کنید و از بین option های موجود، Windows Forms Application را انتخاب کنید، در آخر نیز نام پروژه و مکان ذخیره سازی آن را مشخص کنید. مراحل ذکر شده، در شکل پایین مشخص است.

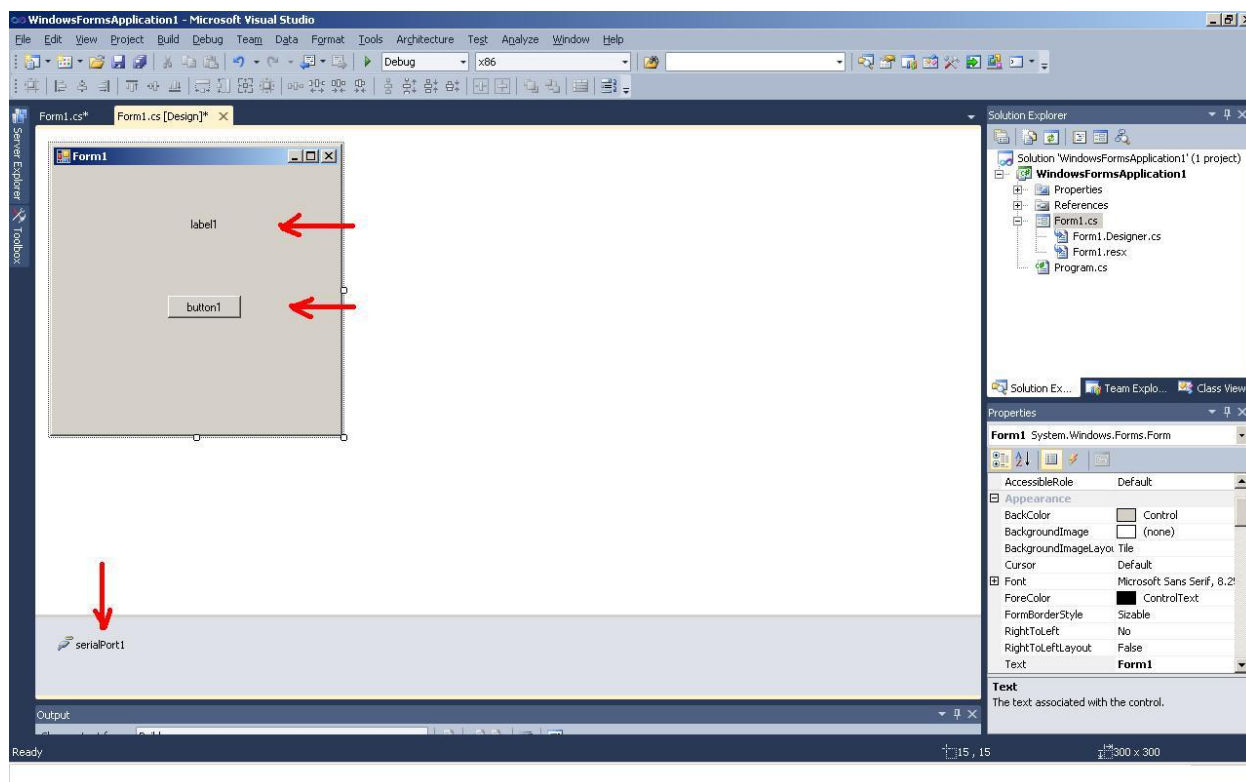


پس از زدن کلید OK پروژه، ایجاد می شود و محیطی مانند شکل زیر را مشاهده می کنید. در سمت راست این محیط، یک Solution Explorer مشاهده می شود که یک لیست از فایل های تولید شده توسط Visual Studio در طول ایجاد پروژه است. Form1.cs را انتخاب کرده و Form1 را مشاهده کنید. این همان فرمی است که قرار است با اجرای برنامه، ظاهر شود. به این فرم که در واقع یک Windows Forms است می توانید component های دلخواه را اضافه نمایید. با توجه به فلش موجود در شکل پایین، نشانگر موس را روی لبه Toolbox قرار داده تا یک لیست از tool های موجود، به نمایش درآیند. در شکل بعد، این لیست را مشاهده می کنید.





از قسمت Common Controls یک Label و یک Button و از قسمت Components یک SerialPort را drag کرده و در Form1، drop نمایید.



حال روی فایل Form1.cs در Solution Explorer کلیک راست کنید و View Code را انتخاب نمایید تا یک پنجره جدید ظاهر شود.

پنجره جدید باز شده با نام Form1.cs می باشد که برنامه مورد نظر را باید در این قسمت بنویسیم.

کد نوشته شده در CodeVision. اطلاعات قابل نمایش خود را از طریق تابع `putchar()` روی پورت سریال قرار می دهد و کد نوشته شده در Visual Studio نیز از طریق تابع `ReadChar(char)` که توسط object پورت سریال فراخوانی می شود، این اطلاعات را که یک کاراکتر می باشد می خواند و آن را روی یک label می نویسد. همچنین BaudRate پورت سریال در این برنامه باید با BaudRate تنظیم شده در تنظیمات مربوط به Usart در CodeVision یکی باشد که این کار نیز توسط property BaudRate پورت سریال تنظیم می شود. (`serialPort1.BaudRate = 110`)

انتقال اطلاعات از میکرو به کامپیوتر

در برنامه نوشته شده زیر، هر وقت button موجود روی فرم فشرده شود، کاراکتری که توسط میکرو به کامپیوتر فرستاده شده است خوانده شده و روی label روی فرم نمایش داده می شود.

```

using ...

namespace WindowsFormsApplication1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            this.Show();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            serialPort1.Open();
            if (serialPort1.IsOpen)
            {
                char a;
                serialPort1.BaudRate = 110;
                a = (char)serialPort1.ReadChar();
                label1.Text = a.ToString();
            }
            else label1.Text = "Serial Port is Closed";
            serialPort1.Close();
        }
    }
}

```

انتقال اطلاعات از کامپیوتر به میکرو

در برنامه نوشته شده زیر، هر وقت button موجود روی فرم فشرده شود، کاراکتری توسط کامپیوتر به میکرو فرستاده می شود و میکرو توسط متد `getchar()`، کاراکتر ارسالی را دریافت می کند.

```

using ...

namespace WindowsFormsApplication1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            this.Show();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            serialPort1.Open();
            if (serialPort1.IsOpen)
            {
                char[] buffer = new char[1];
                buffer[0] = 'a';
                serialPort1.BaudRate = 110;
                serialPort1.Write(buffer, 0, 1);
            }
            else label1.Text = "Serial Port is Closed";
            serialPort1.Close();
        }
    }
}

```