# Important variables

date2[]: feeding time

date3[]:duration in ms

blt[]:belt number

# important loop

loadtable(): load feeding info table

loadtable1(): load tank status monitor panel(table)

# important interface components

Textbox: store data in to feeding information arrays and mysql database(transfer station)

button1,3: start/stop button of manual control

button2:switch mode button

button7:update feeding info(store feeding info into arrays)

Button8:clear all data and create new feeding info table

button9:file button

Button17:export feeding data to excel

Button18:import data from excel

button19:setting button

timer1:check whether system time now equal to store feeding time

timer2,3,4: count the feeding duration for feeding the tank 1,2,3, interval is 1s

timer5,6:control the progress bar1 of manual control

timer7,8,9: perform steps and control the filling time of progress bar 2,3,4(auto)

timer10,11,12:assist the control of progress bar 2,3,4

timer13:display system time now on label3

Timer14:clear all previous data stored in the VS, fill in the new data from excel.

comboBox1:select belt number(manual)

comboBox2:select remote number(manual)

label3:display system time now

lable19: manual feeding duration

# important methods

//Create Excel

Microsoft.Office.Interop.Excel.Application excel = new Microsoft.Office.Interop.Excel.Application();

excel.Application.Workbooks.Add(true);

//fill data(i:row number;j:cells number)

for (int i = 0; i < gridView.RowCount - 1; i++)

{

for (int j = 0; j < gridView.ColumnCount; j++)

{

if (gridView[j, i].ValueType == typeof(string))

{

excel.Cells[i + 2, j + 1] = "'" + gridView.Rows[i].Cells[j].Value.ToString();

excel.Cells[i+2,j+1].Style.HorizontalAlignment = Microsoft.Office.Interop.Excel.XlHAlign.xlHAlignCenter;

excel.Columns.AutoFit();

}

else

{

excel.Cells[i + 2, j + 1] = gridView.Rows[i].Cells[j].Value.ToString();

excel.Cells[i + 2, j + 1].Style.HorizontalAlignment = Microsoft.Office.Interop.Excel.XlHAlign.xlHAlignCenter;

excel.Columns.AutoFit();

}

}

[GetBytes(String)](http://msdn.microsoft.com/en-us/library/ds4kkd55(v=vs.110).aspx)

When overridden in a derived class, encodes all the characters in the specified string into a sequence of bytes.

receivePoint = new IPEndPoint(IPAddress.Parse("10.0.1.41"), port);

define the terminal of the data , parse = analysis or catch useful info