Step-By-Step: Starting a C#.NET Data Acquisition solution using the Universal Library from scratch

Open C#.NET, and start a new Windows Application

Add a reference to MccDaq

- Go to your Solution Explorer window
- Expand the References folder (if it is not already expanded)
- Right mouse click on References folder, and select “Add Reference…”
- The Add Reference dialog box appears
- Scroll down to MccDaq, click on it, click on the Select button at the right
- Then click on the ‘OK’ button at the bottom of the dialog box

Add the namespace to the project

- Go to your Solution Explorer window
- click on Form1.vb then press F7 to activate the code window
- Locate the line “using System.Data;”
- Just after that add “using MccDaq;”

Add Project variables

- Locate the line “publicclass Form1 : System.Windows.Forms.Form”
- After the opening parenthesis add (paste in) the following code:

```
//Here's where we declare our variables for the project

private MccDaq.MccBoard DaqBoard;

   private MccDaq.ErrorInfo ULStat;

   private MccDaq.Range Range;
```

Add the Initialization to Form1 procedure

- Locate“public Form1()” procedure
- Locate “InitializeComponent();”
- Just after that add the following

```
// Initiate error handling

// activating error handling will trap errors like

// bad channel numbers and non-configured conditions.

// Parameters:

//   MccDaq.ErrorReporting.PrintAll :all warnings and errors

//             encountered will be printed

//   MccDaq.ErrorHandling.StopAll   :if an error is encountered,

//             the program will stop
```

```
ULStat =
MccDaq.MccService.ErrHandling(MccDaq.ErrorReporting.PrintAll,
MccDaq.ErrorHandling.StopAll);
```

Add a new instance of a board object

- Go to your Solution Explorer window
- click on Form1.vb then press Shift + F7 to activate the Form Designer
- Double click on the form to activate the “Form1_Load”event
- Just after that add

```
int BoardNum;


BoardNum = 1; //board number assigned by InstaCal

DaqBoard = new MccDaq.MccBoard(BoardNum);
```

That's it. From here you can add code and procedures to things such as:

Please see the Universal Library function reference guide or C:\MCC\Unilib.chm for more information.

For example, you could add a command button and 2 labels to the form, then paste in the follow to the code window:

```
privatevoid button1_Click(object sender, System.EventArgs e)

{

    float EngUnits;
```

```
        System.UInt16 DataValue;

        int Chan;



        //  Collect the data by calling AIn memeber function of MccBoard
object

        //   Parameters:

        //     Chan       :the input channel number

        //     Range      :the Range for the board.

        //     DataValue  :the name for the value collected

        Range = Range.Bip10Volts;     //  select Bip10Volts (member of
Range enumeration)

        Chan = 0;                           //  set input channel



        ULStat = DaqBoard.AIn( Chan, Range, out DataValue);

        if (ULStat.Value == MccDaq.ErrorInfo.ErrorCode.BadRange)

        {

            MessageBox.Show( "Change the Range argument to one
supported by this board.", "Unsupported Range",
MessageBoxButtons.OK);

            Application.Exit();
```

```
        }


        //  Convert raw data to Volts by calling ToEngUnits (member
function of MccBoard class)

        ULStat = DaqBoard.ToEngUnits( Range, DataValue, out EngUnits);



        label1.Text = DataValue.ToString();              //  print the counts

        label2.Text = EngUnits.ToString("F4") + " Volts"; //  print the
voltage

}
```

Note: This will only work for MCC hardware that has an A/D converter with a Bipolar 10 volt range.

Measurement Computing Data Acquisition Knowledgebase
http://kb.mccdaq.com/KnowledgebaseArticle50140.aspx