

TOUCH CONTROLLER

CUWIN

UI Component

사용설명서 Version :200808A

COMFILE
TECHNOLOGY

컴파일 테크놀로지 주식회사
www.comfile.co.kr

본 COMPONENT 는 CUWIN 시리즈에서만 사용 가능합니다.

(타사 WINCE 제품에서는 동작하지 않습니다.)

CUWIN 고객에 한하여 무료로 배포합니다.

본사 영업부에 요청하시면 됩니다. [02-711-2592]

등록상표

WINDOWS 는 Microsoft Corporation 의 등록상표입니다.

CUWIN 은 Comfile Technology 의 등록상표입니다.

기타 다른 상표는 해당회사의 등록상표입니다.

알림

본 설명서의 내용은 사전 통보 없이 변경될 수 있습니다. 본 제품의 기능은 성능 개선을 위하여 사전 통보 없이 변경될 수 있습니다. 본 제품을 이 자료에서 설명한 용도 외에서 사용할 경우, 폐사에서는 어떠한 책임도 지지 않으므로 주의하시기 바랍니다. 본 제품은 컴파일 테크놀로지의 고유 기술을 사용하여 개발된 제품으로 저작권법에 의한 보호를 받고 있습니다. 따라서 본 제품 (제품에 대한 아이디어 및 설명서 및 기타 포함)의 어떠한 부분도 사전에 폐사와의 문서 동의 없이 복사되거나 변경, 재 생산할 수 없으며 또한 다른 언어로도 번역될 수 없습니다.

주의사항

본 UI COMPONENT 는 CUWIN 사용자를 위하여 제작된 것입니다. CUWIN 고객은 무료로 이용할 수 있습니다만, 타사의 WINCE 제품에서는 동작하지 않습니다. 만약 타사 WINCE 제품에 본 프로그램을 무단 복제 및 무단 사용시에는 민,형사상의 법적 제재를 받을 수 있음을 알려드립니다. 본 COMPONENT 는 컴파일 테크놀로지의 지적 재산임을 밝힙니다.

본 UI COMPONENT 는 고객 여러분의 프로그램 제작을 돕기 위해서 라이브러리 형태로 제공하는 것이며, 완성된 형태의 독립 실행 파일이 아닙니다. 따라서 본 프로그램사용으로 인한 어떠한 피해 및 손해에 대하여, 폐사에서는 책임을 지지 않으므로 주의하시기 바랍니다. 본 책자의 내용 또한 저작권법에 의한 보호를 받고 있습니다. 따라서 본 책자의 (아이디어 및 설명서 및 기타 포함)의 어떠한 부분도 사전에 폐사와의 문서 동의 없이 복사되거나 변경, 재 생산할 수 없으며, 또한 다른 언어로도 번역될 수 없습니다.

인쇄 내역

2008년 08월 배포함





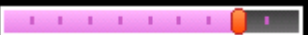


Copyright 1996,2008 컴파일 테크놀로지(주)

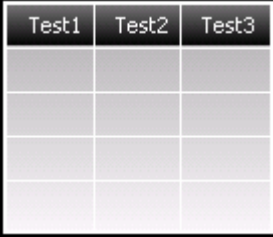
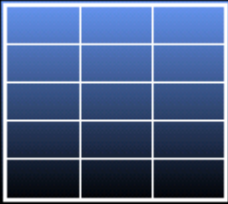


CUWIN UI Component 소개

CUWIN UI Component는 CUWIN 제품군에서 사용이 가능한 그래픽 컨트롤 모음입니다. 이 컨트롤 들은 .Net Compact Framework환경에서 .Net Component형식으로 제공이 되어 VB.Net, C#등의 개발 언어에서 사용이 가능합니다.

Visual Studio 툴에서 기본적으로 제공되는 버튼이나, 라벨 등의 UI 컨트롤의 경우 WINCE 환경에선 그래픽이 단조로워서, 좀 더 화려한 그래픽을 구현하려면, 직접 그래픽 디자인을 해야 하고 이에 대한 코딩을 별도로 해야 하는 번거로움이 있습니다. 그래서 이런 수고를 조금이라도 덜고자 CUWIN에서 사용 가능한 UI Component 모음을 제작하였습니다.

UI Component의 구성은 다음과 같습니다.

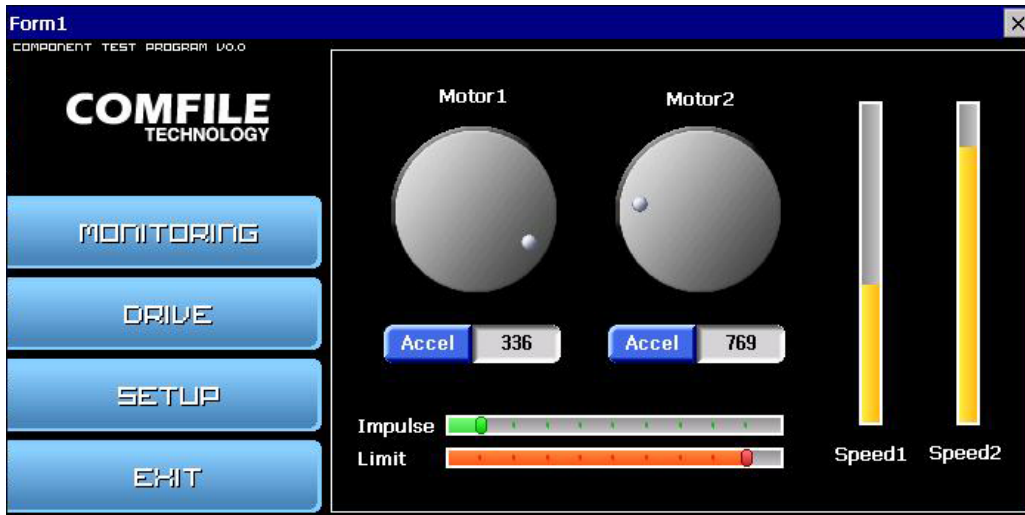
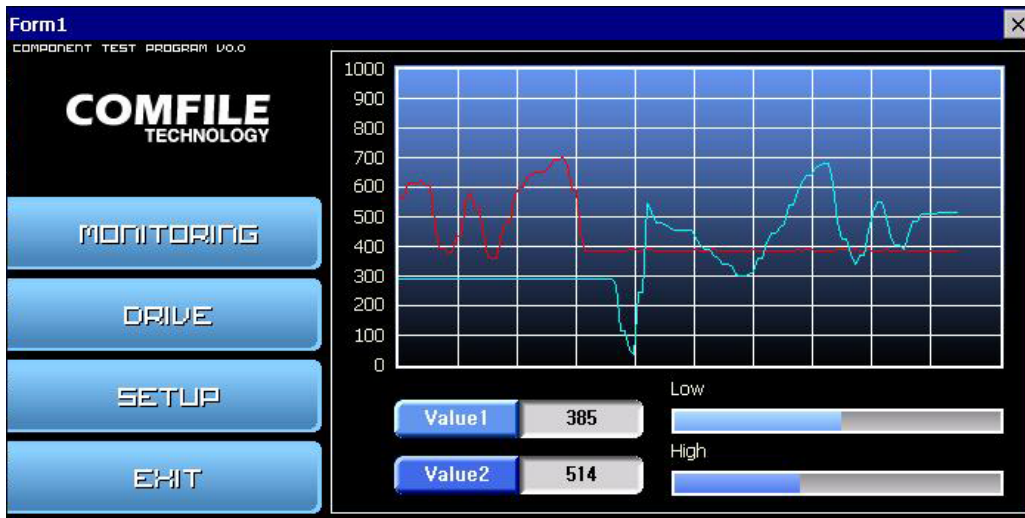
	RoundGradientButtonControl
	LabelControl
	RoundButtonControl
	ProgressBarControl
	SliderBarControl
	ClockControl
	CalendarControl

	<p>TableControl</p>
	<p>TrendControl</p>
	<p>CheckBoxControl</p>
	<p>WhirlControl</p>

총 11개 컨트롤로 구성되어 있으며 컨트롤은 차후 지속적으로 업데이트 될 예정입니다.

각 컨트롤에 대한 설명은 주요한 속성과 이벤트, 멤버 메소드를 위주로 설명하도록 하겠습니다.

UI Component 예제 화면



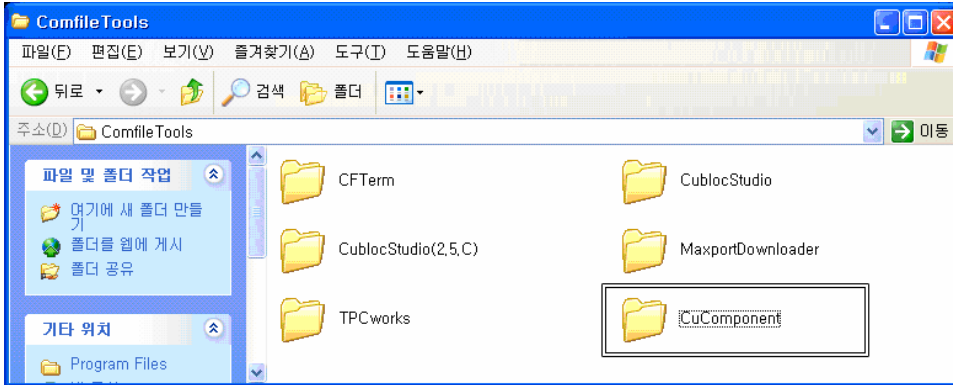
Row	Accel	Impulse	Limit
5	250	50	400
100	30	200	
150	30	200	
180	30	200	
200	50	400	
220	50	400	

위 화면은 컴포넌트를 이용하여 제작한 프로그램으로 압축파일에 함께 들어있습니다.

설치 방법

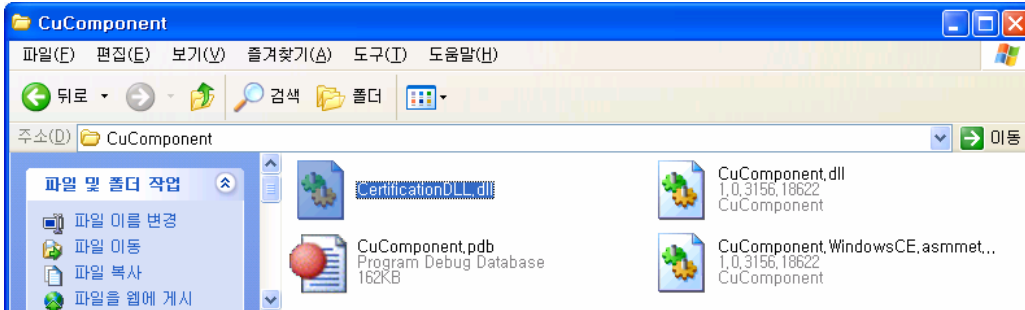
Component를 설치하는 방법은 기존 ModproControl 설치 방법과 동일합니다.

우선 현재 문서와 함께 압축되어 있는 CuComponent 폴더를 원하는 위치로 옮기시기 바랍니다.

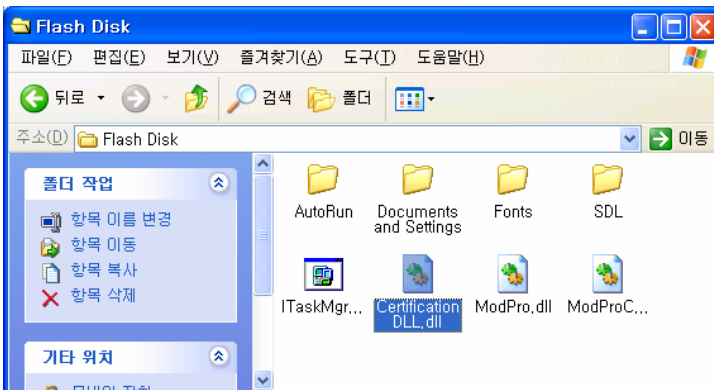


예> C:\WProgram files\ComfileTools\CuComponent

폴더 내용을 보면

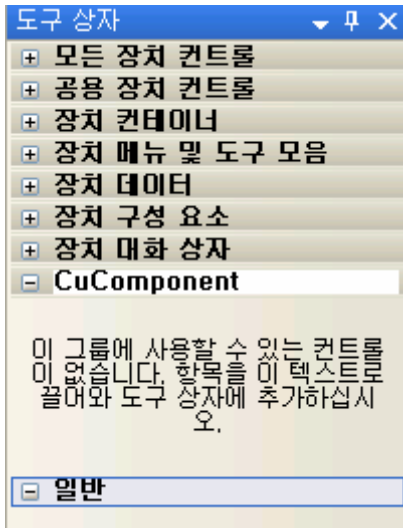


위와 같이 4개 파일이 존재 합니다. 이 파일 중 CertificationDLL.dll 파일을 CUWIN의 Flash disk 폴더로 이동시켜 주십시오.



Visual Studio를 실행하시고 새 프로젝트를 하나 생성합니다.

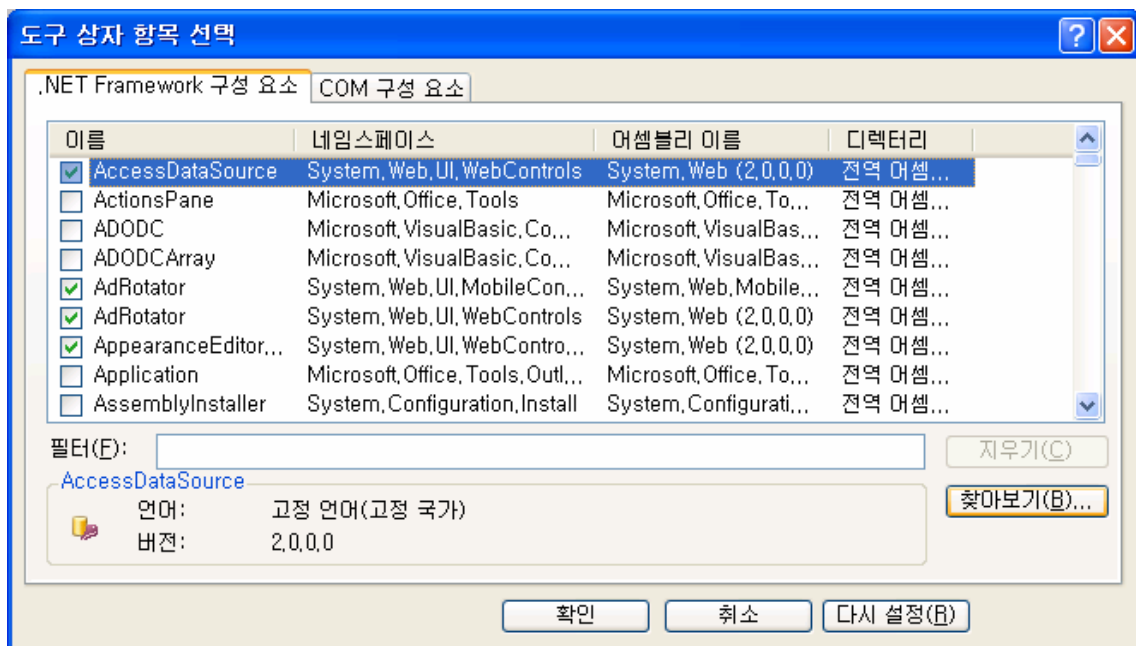
생성 후, 도구상자에서 마우스 우측버튼을 클릭하고 탭 추가 항목을 선택하시면, 새로운 탭이 하나 생성됩니다. 여기에 원하시는 탭 이름을 입력하십시오



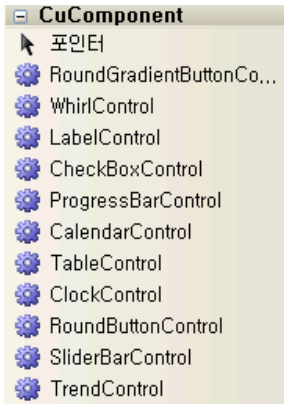
예> CuComponent

새로 생성된 탭에서 마우스 오른쪽버튼을 클릭하시면, 항목선택이라는 메뉴가 나오게 됩니다.

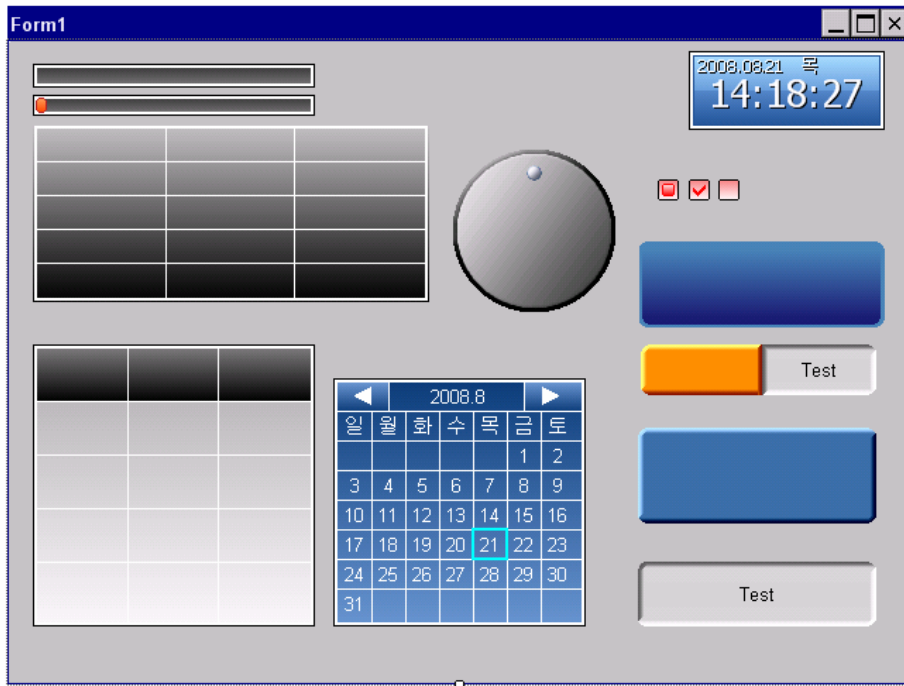
이를 클릭해주십시오.



항목선택을 클릭하면 위와 같은 창이 하나 뜨게 됩니다. 여기서 찾아보기를 선택하여 처음에 옮겼던 CuComponent 폴더의 CuComponent.dll파일을 선택하십시오.



정상적으로 컨트롤이 등록되었다면, 위와 같은 컨트롤들이 등록되어 있을 것입니다. 이제 컨트롤 들을 폼 위로 끌어 놓아 크기를 조정하여 배치하여 주시기 바랍니다.

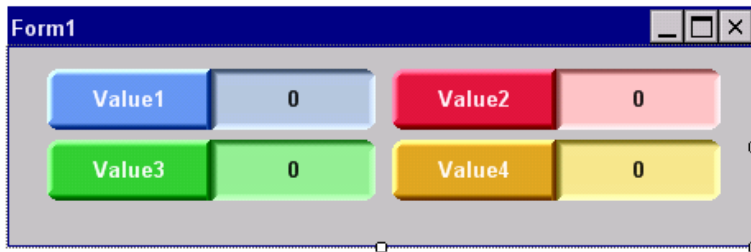
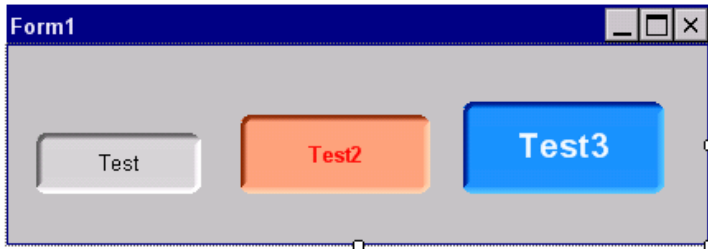


폼 위에 올려놓은 컨트롤은 일부 컨트롤을 제외하고 모두 동일한 크기로 생성이 될 것입니다. 하지만 크기 조절이 가능하므로, 원하는 크기로 변경하시기 바랍니다. 단, WhirlControl의 경우 크기 조절이 불가능합니다.

위의 화면처럼 컨트롤이 보인다면 정상적으로 설치가 된 것입니다.

만약 컨트롤이 정상적으로 보이지 않다면, 폴더에 CuComponent.WindowsCE.asmmeta.dll 파일이 있는지 확인하여 주시기 바랍니다.

LabelControl



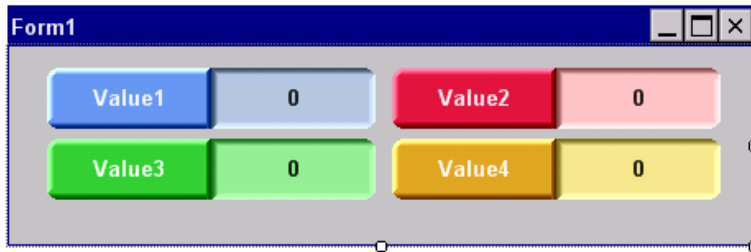
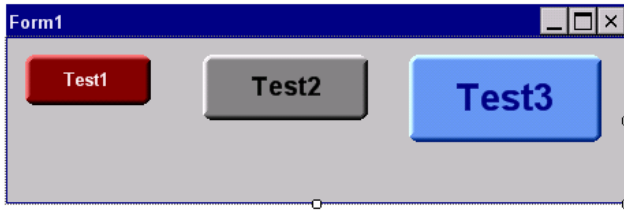
LabelControl의 경우 Text를 디스플레이를 할 수 있는 컨트롤입니다. Visual Studio에서 제공하는 Label과 동일한 역할을 합니다.

주요 속성

속성명	설명
LabelColor	· 컨트롤의 색을 지정합니다.
BackColor	· 컨트롤의 배경색을 지정합니다.
Font	· 컨트롤의 사용될 폰트를 지정합니다.
LeftRectMode	· 컨트롤의 왼편의 모서리를 각진 형태로 사용할 지 둥근 형태로 사용할지를 지정합니다.
TextColor	· 컨트롤의 글자색을 지정합니다.
TextValue	· 컨트롤에서 표시될 Text를 지정합니다.

※위 2번째 그림과 같이 LeftRectMode 속성을 이용하면 RoundButtonControl과 붙여 좀더 자연스러운 그래픽을 만들 수 있습니다.

RoundButtonControl



RoundButtonControl은 모서리가 둥근 형태의 버튼입니다.

주요 속성

속성명	설명
ForeColor	· 컨트롤의 색을 지정합니다.
ClickForeColor	· 컨트롤이 눌러졌을 때 색을 지정합니다.
BackColor	· 컨트롤의 배경색을 지정합니다.
Font	· 컨트롤의 사용될 폰트를 지정합니다.
RightRectMode	· 컨트롤의 오른쪽의 모서리를 각진 형태로 사용할 지 둥근 형태로 사용할지를 지정합니다.
TextColor	· 컨트롤의 글자색을 지정합니다.
TextValue	· 컨트롤에서 표시될 Text 를 지정합니다.

※위 2번째 그림과 같이 RightRectMode 속성을 이용하면, LabelControl과 붙여 좀더 자연스러운 그래픽을 만들 수 있습니다.

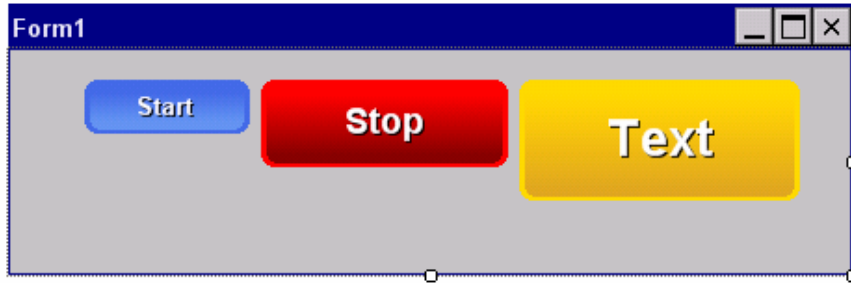
이벤트

Click: 컨트롤의 클릭 이벤트입니다.

MouseDown: 컨트롤이 눌러졌을 때 이벤트입니다.

MouseUp: 컨트롤이 떼어졌을 때 이벤트입니다.

RoundGradientButtonControl



RoundGradientButtonControl은 Gradation효과가 들어간 모서리가 둥근 형태의 버튼입니다.

주요 속성

속성명	설명
StartColor	· 버튼의 시작 색을 지정합니다.
EndColor	· 버튼의 끝 색을 지정합니다.
ClickStartColor	· 버튼 클릭 시 시작 색을 지정합니다.
ClickEndColor	· 버튼 클릭 시 끝 색을 지정합니다.
BackColor	· 컨트롤의 배경색을 지정합니다.
TextColor	· 컨트롤의 글자색을 지정합니다.
TextValue	· 컨트롤에서 표시될 Text를 지정합니다.
Font	· 컨트롤의 사용될 폰트를 지정합니다.

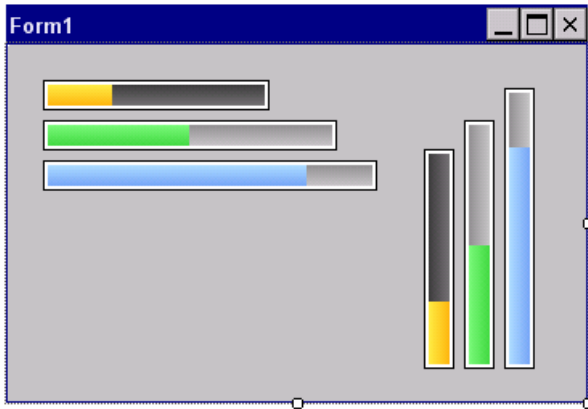
이벤트

Click: 컨트롤의 클릭 이벤트입니다.

MouseDown: 컨트롤이 눌러졌을 때 이벤트입니다.

MouseUp: 컨트롤이 떼어졌을 때 이벤트입니다.

ProgressBarControl

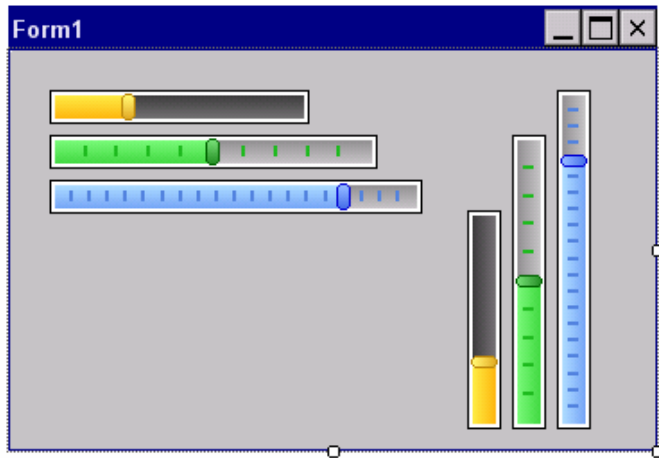


ProgressBarControl은 Max치에 대한 현재 Value값을 게이지 형태로 디스플레이 하는 컨트롤입니다.

주요 속성

속성명	설명
EmptyColor	· 값이 비어있을 때의 색을 지정합니다.
FillColor	· 값이 채워져 있을 때의 색을 지정합니다.
LineColor	· 테두리의 색을 지정합니다.
MaxValue	· 수치의 최대치를 지정합니다.
Value	· 현재 컨트롤의 수치값을 지정합니다.
ViewMode	· 디스플레이 형태를 수직 혹은 수평 형태로 지정합니다.

SliderBarControl



SliderBarControl은 Max치에 대한 현재 Value값을 디스플레이하고 이를 변경할 수 있는 컨트롤입니다.

주요 속성

속성명	설명
EmptyColor	· 값이 비어있을 때의 색을 지정합니다.
FillColor	· 값이 채워져 있을 때의 색을 지정합니다.
LineColor	· 테두리의 색을 지정합니다.
MaxValue	· 수치의 최대치를 지정합니다.
Value	· 현재 컨트롤의 수치값을 지정합니다.
ViewMode	· 디스플레이 형태를 수직 혹은 수평 형태로 지정합니다.
PointFillColor	· 값을 표시하는 Point 커서의 색을 지정합니다.
PointLineColor	· 값을 표시하는 Point 커서의 테두리 색을 지정합니다.
Divide	· 컨트롤을 지정한 수치로 등분합니다. · 속성에 입력할 수 있는 값은 MaxValue값의 대한 소수 형태이며 그 이외 값을 입력할 시 가장 가까운 소수를 찾아 값이 지정됩니다. 최대 20등분까지 가능합니다. · 값이 1인 경우 등분 없이 컨트롤을 사용하게 됩니다.

이벤트

Click: 컨트롤의 클릭 이벤트입니다.

ClockControl



ClockControl은 시간을 디스플레이 하는 컨트롤 입니다.

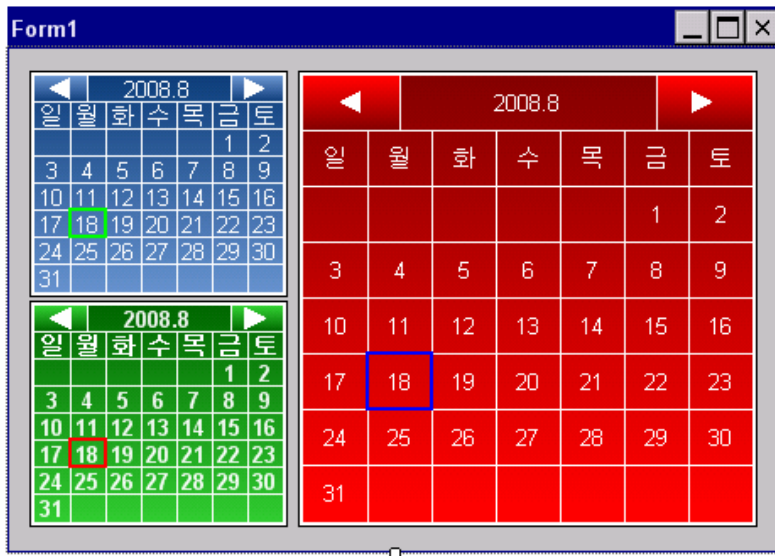
주요 속성

속성명	설명
DisplayLanguage	· 요일을 표시할 언어를 지정합니다.
UpStartColor	· 컨트롤 상단의 시작 색을 지정합니다.
UpEndColor	· 컨트롤 상단의 끝을 지정합니다.
DownStartColor	· 컨트롤 하단의 시작 색을 지정합니다.
DownEndColor	· 컨트롤 하단의 시작 색을 지정합니다.
LineColor	· 컨트롤의 테두리 색을 지정합니다.
TextColor	· 컨트롤에서 사용하는 텍스트의 색을 지정합니다.
DateFont	· 날짜를 표시하는 폰트를 지정합니다.
TimeFont	· 시간을 표시하는 폰트를 지정합니다.

주요 메소드

메소드 명	설명
Start ()	시간 디스플레이를 시작합니다.
Stop ()	시간 디스플레이를 중지합니다.

CalendarControl



CalendarControl은 달력을 디스플레이 할 수 있는 컨트롤 입니다. 상단 화살표를 눌러 표시 되는 달을 변경할 수 도 있고 특정일에 메모를 삽입할 수 있습니다. 날짜를 클릭할 시 상, 하, 좌, 우, 각 대각선 방향까지 총 9칸만큼의 크기의 메모를 디스플레이 할 수 있는 영역 이 나타납니다.

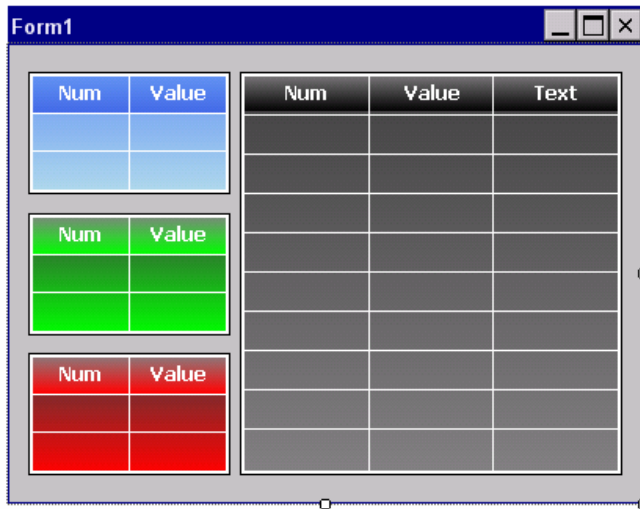
주요 속성

속성명	설명
DisplayLanguage	· 컨트롤의 시작 색을 지정합니다.
StartColor	· 컨트롤의 끝 색을 지정합니다.
EndColor	· 메모가 삽입된 날을 표시할 테두리 색을 지정합니다.
MemoColor	· 오늘 날짜를 표시할 테두리 색을 지정합니다.
TodayLineColor	· 글자색을 지정합니다.
Font	· 컨트롤에서 사용할 폰트를 지정합니다.

주요 메소드

메소드 명	설명	
AddMemo (string Date, string Data)	지정한 날짜에 지정한 메모를 등록합니다	
	인자	설명
	Date	메모 날짜 (Format:"2008.8.20"),
	Data	메모 내용

TableControl



TableControl은 지정한 행, 열 만큼의 테이블을 사용할 수 있는 컨트롤 입니다.

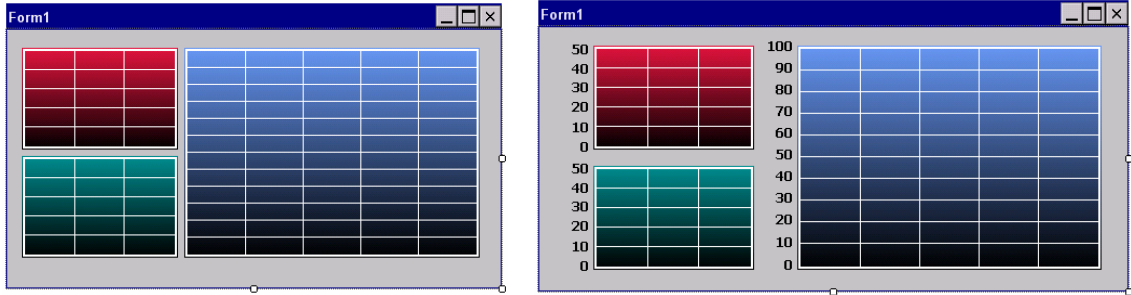
주요 속성

속성명	설명
Rows	· 테이블의 행수를 지정합니다.
Cols	· 테이블의 열수를 지정합니다.
ColumnStartColor	· 열 제목 부분의 시작색을 지정합니다.
ColumnEndColor	· 열 제목 부분의 끝색을 지정합니다.
ColumnFont	· 열 제목 부분에서 사용할 폰트를 지정합니다.
ColumnStrings	· 열 제목, string 배열 형태를 지정합니다.
ColumnTextColor	· 열 제목 부분의 글자색을 지정합니다.
LineColor	· 컨트롤의 테두리색을 지정합니다.
TableStartColor	· 데이터 영역의 시작색을 지정합니다.
TableEndColor	· 데이터 영역의 끝색을 지정합니다.
TextColor	· 데이터 영역의 글자색을 지정합니다.
Font	· 컨트롤에서 사용할 폰트를 지정합니다.

주요 메소드

메소드 명	설명	
SetValue (int Row, int Col, string val)	데이터를 지정한 셀에 입력합니다.	
	인자	설명
	Row	메모 날짜 (Format:"2008.8.20"),
	Col	메모 내용
SetValues (string[,] values)	Val	입력할 데이터
	테이블 전체 데이터를 입력합니다.	
	인자	설명
	Values	테이블 데이터, 2차원 string 배열

TrendControl



TrendControl은 입력되는 값을 실시간 그래프 형태로 표시하는 컨트롤입니다. 기본적으로 좌측의 그림처럼 눈금에 대한 값을 표시하지 않습니다. 만약 우측 그림처럼 값을 표기하고 싶으시다면, label를 배치하여 표시해 주시기 바랍니다.

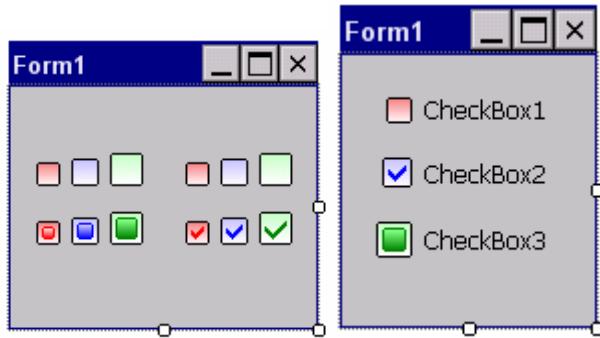
속성

속성명	설명
Rows	· 테이블의 행수를 지정합니다.
Cols	· 테이블의 열수를 지정합니다.
BackColor	· 컨트롤의 색을 지정합니다.
LineColor	· 컨트롤의 테두리색을 지정합니다.
MaximumValue	· 값의 최대치를 지정합니다.
RefreshInterval	· 값을 읽어 화면의 뿌리는 타임을 지정합니다.
ShiftGapWidth	· 이전 값과 현재 값 사이 간격을 지정합니다.
ValueCount	· 화면에 표시할 값의 개수를 지정합니다.

주요 메소드

메소드 명	설명	
SetValue (ArrayList data)	컨트롤의 표시할 데이터를 입력합니다	
	인자	설명
	ArrayList	입력데이터, ArrayList방식
Start ()	컨트롤 표시를 시작합니다.	
Stop ()	컨트롤 표시를 중지합니다.	

CheckBoxControl



CheckBoxControl은 좌측 그림과 같이 두 가지의 형태의 체크박스를 지원합니다.

CheckBoxControl은 컨트롤 자체에서 Text를 지원하지 않고 있어, 우측 그림처럼 CheckBoxControl과 Label을 같이 배치하여 사용하셔야 합니다.

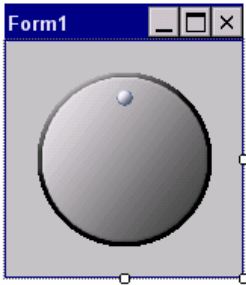
속성

속성명	설명
BackColor	· 컨트롤의 배경색을 지정합니다.
CheckStartColor	· Rect 모드: 체크 표시의 시작색을 지정합니다. · Check 모드: 체크 표시 색을 지정합니다.
CheckEndColor	· Rect 모드: 체크 표시의 끝색을 지정합니다.
EmptyColor	· 체크 되지 않았을 때 색을 지정합니다.
LineColor	· 테두리 색을 지정합니다.
ControlSize	· 컨트롤 크기를 지정합니다. 크기는 Small, Medium, Large 세 가지입니다.
ViewMode	· 체크 형태를 지정합니다. 모드에는 Rect, Check 두 가지가 있습니다.
Checked	· 컨트롤이 체크 상태를 지정합니다.

이벤트

Click: 컨트롤의 클릭 이벤트입니다.

WhirlControl



WhirlControl은 다이얼 버튼과 같은 형식으로 제작되었습니다. WhirlControl에서 다루는 값은 360도를 기준으로 한 각도 값으로, 이 값을 특정 값으로 변경하는 것은 간단한 비례공식을 이용하여 사용할 수 있습니다.

예> 1000을 Max값으로 하였을 때 현재의 각도가 나타내는 값, 현재 각도:180

$$360:180 = 1000:x$$

$$360*x = 1000*180$$

$$x = 180000/360$$

$$x = 500$$

WhirlControl은 크기 수정이 불가능 합니다.

속성

속성명	설명
Angle	· 현재 멈춰있는 포인트의 각도입니다.
ChangeAngle	· 현재 이동중인 포인트의 각도입니다.
BackColor	· 컨트롤의 배경색을 지정합니다.
TimerInterval	· 포인트가 이동할 때의 Interval을 지정합니다.

이벤트

ChangedValue: 포인트가 이동이 완료한 후 호출되는 이벤트입니다.

ChangingValue: 포인트가 이동 중일 때 호출되는 이벤트입니다.

예제

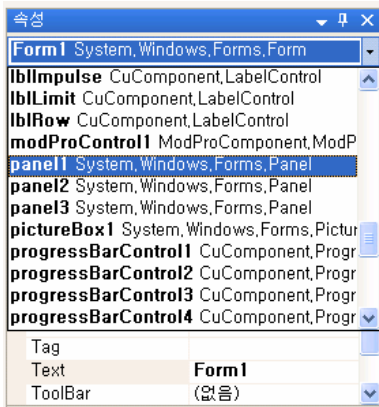
예제 프로그램은 ZIP 파일 안에 라이브러리와 함께 들어있습니다.

예제 프로그램은 CuComponent(UI Component)와 ModproControl을 이용하여 제작하였습니다. 코드에 대한 자세한 내용은 실제 예제 프로그램의 코드에 주석으로 처리 해주었습니다. 본 문서에서는 예제 프로그램에서 유의할 사항에 대해서만 기술하였습니다.

- Panel을 이용한 페이지 교체

페이지 교체는 동일한 위치의 동일한 크기의 Panel 컨트롤 3개를 교체하여 보여주는 방식을 사용하였습니다.

Panel을 디자인 타임에서 교체하는 방법은



속성창의 컨트롤 리스트의 Panel을 선택하면 아래와 같이 폼 디자인 화면의 Panel이 선택이 됩니다.



선택된 Panel에서, 좌측상단의 컨트롤 이동 아이콘 → 우측버튼을 클릭하여 “맨 앞으로 가져오기” 메뉴를 선택하여 주시면 컨트롤이 보이게 됩니다.

- CUBLOC과의 통신

예제 프로그래밍에 ModproControl을 이용하여 Cubloc과 통신을 하게 됩니다.

아래는 CUBLOC쪽 소스입니다.

```
Const Device = CB280
```

```
Opencom 1,9600,3,100,100 ' RS 232 통신 규격 설정
```

```
Set Modbus 1,1
```

```
Set Ladder On
```

```
Ramclear
```

```
Dim nADValue As Integer
```

```
Dim nADValue2 As Integer
```

```
Do
```

```
    nADValue=Adin(0)
```

```
    nADValue2 =Adin(1)
```

```
    If nADValue > 1000 Then
```

```
        nADValue=1000
```

```
    End If
```

```
    If nADValue2 > 1000 Then
```

```
        nADValue2=1000
```

```
    End If
```

```
    _D(0)=nADValue
```

```
    _D(1)=nADValue2
```

```
Loop
```

주의사항

현재 CuComponent의 경우, 컨트롤 자체에서 그래픽 작업을 하기 때문에, 지나치게 많은 컨트롤을 사용시 메모리 사용과 속도에 제약이 생길 수 있습니다.

항상 CuComponent의 컨트롤을 이용하여 프로그램 하실 때에는 메모리와 속도에 유의하여 프로그램 하시기 바랍니다.

CuComponent는 저희 CUWIN 제품군에서만 동작하도록 되어있습니다. CUWIN이외 제품군에서 사용이 불가하오니 이에 유의 하여 주시기 바랍니다.

앞으로도 지속적으로 CuComponent를 업데이트 할 예정입니다.

사용하시면서 버그를 발견하시거나, 개선점, 새로운 컨트롤에 대한 내용을 저희 홈페이지의 커뮤니티에 올려주세요!