

Variablen – Datentypen

int – Ganzzahl (100)
double – Kommazahl (1.0)
string – Zeichenkette ("Wert")
bool – Wahrheitswert (**true** oder **false**)

Variablen – Daten Speichern

```
int sensor_1_wert = 1;           // Danach steht 1 in der Variablen "sensor_1_wert"
double rechnung = 1 + sensor_1_wert * 2 / 3; // Rechnen
if (sensor_1_wert < 500) { }      // Variable mit Wert vergleichen (>, >=, ==, !=, <, <=)
string sensor_wert_text = sensor_1_wert.ToString(); // Zahl zur Zeichenkette
```

Solution 'DimLED' (1 project)

DimLED

Properties

References

Microsoft.CSharp

Phidget21.NET

System.Runtime.InteropServices

Form1.cs

Form1.Designer.cs

Form1.resx

Program.cs

Projektname

Phidget dll – wichtig, um die Phidgets zu benutzen

Die Oberfläche & das Programm
Doppelklicken auf die Oberfläche öffnet das Programm

Das eigentliche Programm

InterfaceKit

```
InterfaceKit ifKit;           // die Variable ifKit erzeugen
ifKit = new InterfaceKit();    // Ein InterfaceKit-Objekt erzeugen
ifKit.open();                 // Das InterfaceKit benutzen
ifKit.waitForAttachment();     // Warten bis angeschlossen
int sensor_links_wert = ifKit.sensors[0]; // Sensor 0, links auslesen (rechts ist 7)
ifKit.outputs[0] = sensor_links_wert > 500; // die LED 0 anschalten, wenn wert > 500
```

Servo

```
Servo servo = new Servo();    // Servo-Objekt erzeugen
servo.open();
ifKit.waitForAttachment();
servo.servos[0].Position = 15.00; // Servo auf minimale Position setzen (max 232.00)
```

Kontrollfluss

```
if (ifKit.Attached)           // WENN das InterfaceKit angeschlossen ist,
{                               // dann
    label1.Text = "angeschlossen"; // Setze den Text von label1 auf "angeschlossen"
}

while (ifKit.sensors[0] > 300) // SOLANGE der Wert von Sensor 1 größer als 300 ist,
{                               // tue folgendes:
    Thread.Sleep(100);         // Warte 100 Millisekunden
    ifKit.outputs[0] = !ifKit.outputs[0]; // Schalte Ausgang 0 um
}

for (int i = 15; i < 232; i = i + 1) // Wiederhole bestimmte Anzahl oft (15 bis 232)
{                                   // mit i = 15, i = 16, i = 17, ..., i = 231, i = 232, tue für jedes i:
    Thread.Sleep(10);          // Warte 10 Millisekunden
    servo.servos[0].Position = i; // Setze Servomotorposition auf i
}
```