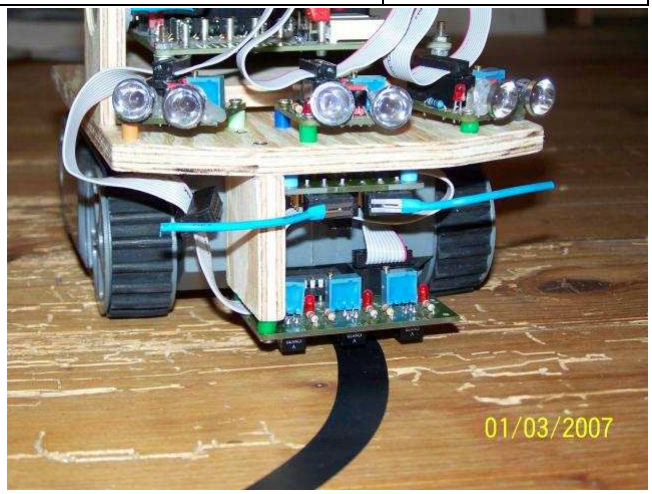
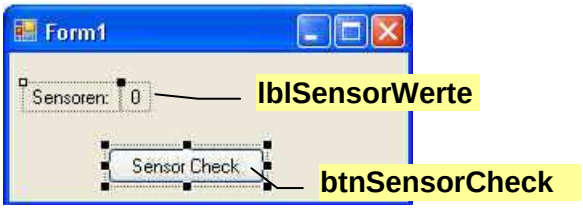
Richtig interessant wird es natürlich erst, wenn der Roboter seine Sensoren verwendet, um bestimmte Aufgaben selbständig zu erledigen. Im Bild werden die Reflextaster verwendet, um den Roboter auf einer schwarzen Linie fahren zu lassen.



Zum Test der Sensorabfrage erstellen Sie ein neues Projekt aus der Roboter-Vorlage.

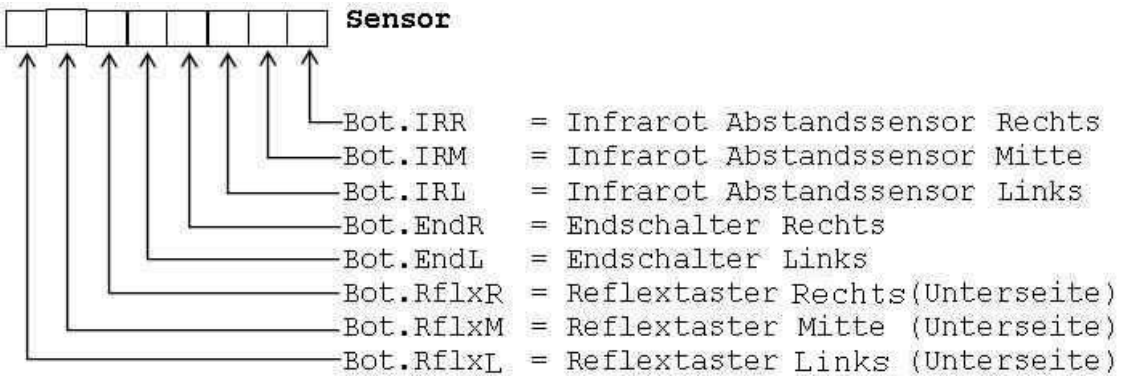
Fügen Sie zwei Label- und ein Button-Steuerelement wie abgebildet ein.



Programmieren Sie die abgebildete Click-Ereignisprozedur für den Button:

```
private void btnSensorCheck_Click(object sender, EventArgs e)
{
    Sensor = robo.readSensors();
    lblSensorWerte.Text = Convert.ToString(Sensor);
}
```

Ein Programmlauf ergibt etwa folgendes Ergebnis: Sensoren: IRM, EndR, EndL
Das bedeutet, die Sensoren Infrarot-Abstandssensor Mitte, Endschalter Links und Endschalter Rechts sind aktiviert.
Die Variable **Sensor** ist vom Enum-Typ Sens (Alias Bot) und besteht genaugenommen aus einem Byte, also 8 Bit. Dabei gilt folgende Reihenfolge der Bit inherhalb von Sensor:



Die Abfrage einzelner Sensoren kann über logische Verknüpfungen realisiert werden, die immer einen **boolschen Wahrheitswert**, also WAHR (TRUE) oder FALSCH (FALSE), ergeben müssen.
Als Beispiel soll der Roboter den linken Endschalter abfragen und bei Betätigung den Roboter stoppen:

```
if ((Sensor & Bot.EndL) == Bot.EndL)
{
    robo.motor(Mot.stopp, 0);
}
```

Innerhalb der Klammer der **if-Anweisung** wird der boolsche Wahrheitswert gebildet.

Die Wirkung wird am Beispiel des betätigten Endschalter Links (zusätzlich ist noch RflxL betätigt) gezeigt:

Der aktueller Wert der Variablen Sensor 00110000
wird UND-verknüpft mit Bot.EndL & 00010000
Das Ergebnis wird mit Bot.EndL = 00010000 == Bot.EndL
verglichen. Das Ergebnis der Abfrage ist hier 1, d.h., der Roboter wird gestoppt.

Um auf den rechten Endschalter ebenfalls reagieren zu können, kann auf die gleiche Weise vorgegangen werden, bzw. beide Abfragen können **ODER**-verknüpft werden. In C# lautet die Formulierung dann so:

```
if ((Sensor & Bot.EndL) == Bot.EndL || (Sensor & Bot.EndR) == Bot.EndR)
```

Endschalter Links — ODER — Endschalter Rechts

Eine etwas kürzere Möglichkeit ist:

```
if ((Sensor & (Bot.EndL | Bot.EndR)) != 0)
    robo.motor(Mot.stopp,0 );
```

Machen Sie sich die Funktion der Abfrage mit Hilfe der folgenden Operatorenliste klar!
C# kennt folgende Operatoren für das Bitweise (logische)-Verknüpfen von Byte-Werten, sowie das Verknüpfen von boolschen Wahrheitswerten:

Verknüpfung	Bitweise	Bool
UND	&	&&
ODER	|	||
Nicht	!	
XOR	^	

Für die Bildung des boolschen Wahrheitswertes existieren die relationalen Operatoren:

Test	Operator	Test	Operator
Gleichheit	==	**Größer gleich**	>=
Ungleichheit	!=	**Kleiner gleich**	<=
Größer als	>	**Kleiner als**	<

Die Klammern in den Wahrheitsausdrücken sind immer dann nötig, wenn die gewünschte Reihenfolge der Auswertung von der vorgegebenen Reihenfolge der Auswertung abweicht. In der abgebildeten Tabelle hat sinkt der Vorrang von oben nach unten und von Rechts nach Links:

Relational und Typtest	<, >, <=, >=,
Gleichheit	==, !=
Logisch, in Operatorrangfolge	&, ^, |
Bedingt, in Operatorrangfolge	&&, ||, ?: