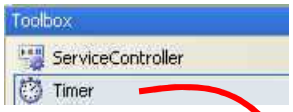
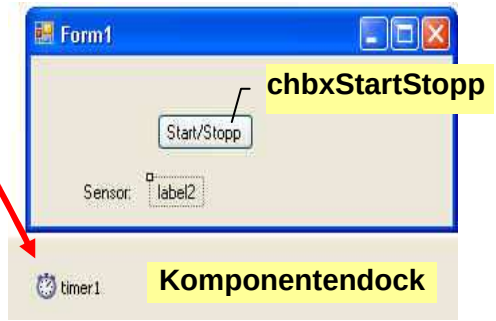
Um auf Hindernisse schnell reagieren zu können, müssen die Sensoren in kurzen Zeitintervallen abgefragt werden. Dazu wird ein **Timer-Steuerelement** verwendet. Erstellen Sie zunächst ein neues Projekt und fügen Sie Label- und Checkbox-Steuerelemente, wie abgebildet, in die Form ein. Um das **Checkbox**-Steuerelement als Schaltfläche darzustellen, stellen Sie dessen **Appearance**-Eigenschaft auf Button!

Das Timer-Steuerelement wird aus der Toolbox auf das Formular gezogen.



Es erscheint dann unterhalb des Forms im **Komponentendock**:



Es werden noch zwei Prozeduren benötigt, um einerseits den Timer mit "Start/Stopp" zu starten und andererseits bei jedem Timeraufruf (Tick) die Sensoren abzufragen und deren Wert anzuzeigen:

```
private void chbxStartStopp_CheckedChanged(object sender, EventArgs e)
{
    int speed = Convert.ToInt16(numSpeed.Value);

    if (chbxStartStopp.Checked)  // ist der "Haken" gesezt??
    {
        timer1.Interval = 100;   // Timer-Tick-Ereignis alle 100ms
        timer1.Start();
    }
    else
    {
        timer1.Stop();
    }
}

private void timer1_Tick(object sender, EventArgs e)
{
    Sensor = robo.readSensors();
    lblSensor.Text = Convert.ToString(Sensor);
}
```

Aufgaben:

- Der Roboter soll Geradeausfahren, solange der mittlere Abstandssensor (IRM) kein Hindernis erkennt.

- Fügen Sie einen "Not-Stopp" mit den Endschaltern ein (Siehe oben!).

- Ergänzen Sie eine Geschwindigkeitssteuerung!

- Programmieren Sie einen einfachen Ausweichroboter! Ergänzen Sie die Timer-Tick-Prozedur durch die abgebildete Switch-Abfrage!

Machen Sie sich die Funktion der Switch-Anweisung klar!

Was passiert, wenn die erste **break**-Anweisung weggelassen wird?

```
switch (Sensor)
{
    case Bot.IRL:
        robo.motor(Mot.rechts, speed); break;
    case Bot.IRM:
        robo.motor(Mot.stopp, 0); break;
    case Bot.IRR:
        robo.motor(Mot.links, speed); break;
    default:
        robo.motor(Mot.vor, speed); break;
}
```

- Einige Sensor-Kombinationen sind noch nicht codiert. Erweitern Sie die Steuerung durch Logische ODER-Verknüpfung der Case-Bedingungen. Bsp.: Die Bedingung **Bot.IRL | Bot.IRM** frägt den Wert 00000110 der Sensor-Variablen ab! Überlegen Sie, wie der Roboter "klug" reagieren soll!

- Um den Roboter bei Hindernissen "sensorlos" zurückfahren zu lassen, kann man in den entsprechenden case-Zweig eine Wartezeit einprogrammieren:

```
case Bot.IRL|Bot.IRM|Bot.IRR:
    robo.motor(Mot.rechtsrueck, speed); // Nach hinten-rechts ausweichen!
    Thread.Sleep(1000);        // 1s warten
    break;
```

- Lassen Sie den Summer vor dem Rückfahren 3 x 0,5s beepen, mit jeweils 0,5s Pause!

- Überlegen Sie sich weitere Verbesserungen!