

## 13.6 Kontextmenü – TrayIcon

Die Komponente `gb.gui.trayicon` mit den beiden Klassen `TrayIcon` und `TrayIcons` wird im Kapitel 19.2 beschrieben. Im diesem Kapitel geht es darum, einem Tray-Icon ein Kontext-Menü zuzuweisen, das in der Task-Leiste über dem Tray-Icon mit der rechten Maustaste aufgerufen werden kann.

Wenn Sie ein Gambas-Programm im Hintergrund laufen lassen wollen, dann bietet sich für den Zugriff auf das Programm über ein Icon in der Task-Leiste die Klasse `TrayIcon` an. Das hat den Vorteil, dass Sie einerseits stets Zugriff auf die Programmoberfläche haben oder alternativ ausgewählte Teil-Programme in einem dem Tray-Icon zugewiesenen Kontext-Menü aufrufen können. Das Menü wird in vor-gestellten Projekt in einem Modul deklariert. Sie können es aber auch mit dem Menü-Editor anlegen, dessen Beschreibung Sie im Kapitel 13.2 Menü-Editor nachlesen können.

Da nicht jeder Leser über ein USB-Interface zur Temperatur-Messung verfügt und die Klasse `TrayIcon` vor Allem für Programme interessant ist, die lange Zeit versteckt arbeiten und ihre Anwesenheit nur durch ein Icon in der Taskleiste signalisieren, wird Ihnen ein Simulationsprojekt vorgestellt. Die Temperaturwerte werden zufällig in einem vorgegebenen Temperaturbereich erzeugt und angezeigt. Außerdem werden die Temperaturwerte in einer Log-Datei protokolliert:

```
MESS-PROTOKOLL
DATUM: 14. Februar 2018
-----
> 09:47:56 | T = 22,3 °C
> 09:47:57 | T = 22,1 °C
> 09:47:58 | T = 21,8 °C
> 09:47:59 | T = 21,0 °C
> 09:48:00 | T = 21,5 °C
> 09:48:01 | T = 20,3 °C
> 09:48:02 | T = 19,9 °C
```

Der Programmteil, der sich dem Kontext-Menü für das Tray-Icon widmet, wurde in einem Modul gespeichert. In diesem Modul werden

- zuerst ein Menü deklariert,
- dann diesem Menü drei weitere Menüs als Untermenüs zugewiesen und
- abschließend die Aktionen beschrieben, die beim Aufruf der Menüs im Kontext-Menü ausgelöst werden sollen.

Der Quelltext zum Modul `MTI` (Modul `TrayIcon`) birgt keine Überraschungen:

```
[1] ' Gambas module file
[2]
[3] Public M10MainMenu As Menu
[4] Public M11ProgramHideShow As Menu
[5] Public M12ShowData As Menu
[6] Public M13CloseExperiment As Menu
[7]
[8] Public Sub CreateMenu()
[9]
[10] ' True -> The menu is NOT displayed in the main program!
[11] M10MainMenu = New Menu(FMain, True) As "Menu"
[12] M10MainMenu.Text = ("Caption")
[13]
[14] ' This property is required for the assignment as pop-up menu of the tray icon!
[15] M10MainMenu.Name = "M10MainMenu"
[16]
[17] M11ProgramHideShow = New Menu(M10MainMenu) As "M11ProgramHideShowShow"
[18] M11ProgramHideShow.Text = ("Hide program")
[19] M11ProgramHideShow.Picture = Picture["icon:/16/down"]
[20]
[21] M12ShowData = New Menu(M10MainMenu) As "M12ViewData"
[22] M12ShowData.Text = ("Display data")
[23] M12ShowData.Picture = Picture["icon:/16/view-icon"]
[24]
[25] M13CloseExperiment = New Menu(M10MainMenu) As "M13CloseExperiment"
[26] M13CloseExperiment.Text = ("Close experiment")
[27] M13CloseExperiment.Picture = Picture["icon:/16/quit"]
[28]
[29] End
[30]
[31] Public Sub M11ProgramHideShowShow_Click()
[32] If M11ProgramHideShow.Show Then
[33] FMain.Visible = False
```

```
[34]     M11ProgramHideShow.Text = ("Show program")
[35]     M11ProgramHideShow.Picture = Picture["icon:/16/up"]
[36]     Else
[37]         FMain.Visible = True
[38]         M11ProgramHideShow.Text = ("Hide program")
[39]         M11ProgramHideShow.Picture = Picture["icon:/16/down"]
[40]     Endif
[41] End
[42]
[43] Public Sub M12ViewData_Click()
[44]     Message.Info("Measurement data are displayed ...")
[45] End ' M12ViewData
[46]
[47] Public Sub M13CloseExperiment_Click()
[48]     FMain.Close()
[49] End
```

Das vollständige Projekt zur Simulation einer Langzeit-Temperaturmessung finden Sie im Projektarchiv `long.time.experiment.tar.gz` im Downloadbereich. Den Quelltext für das Hauptprogramm finden Sie auch am Ende des Kapitels.



Abbildung 13.6.1: GUI des Simulationsprojektes

Hinweise:

- In der System-Menüleiste (Task-Leiste) finden Sie nach dem Programmstart das Tray-Icon , das in der IDE von Gambas im Bildeditor entworfen wurde.
- Mit einem Klick der linken Maustaste auf das Tray-Icon können Sie eine Programmhilfe aufrufen. Diese Hilfe ist im vorliegenden Projekt nicht vollständig formuliert.
- Ein Klick mit der rechten Maustaste über dem Tray-Icon öffnet ein Kontextmenü mit 3 Menüs für ausgewählte Programm-Aktionen:



Abbildung 13.6.2: Inhalt Kontext-Menü für das Tray-Icon

- Die gespeicherten Messdaten werden in diesem Projekt nicht ausgewertet und angezeigt. Dafür wird nur ein Hinweis in einer Message-Box ausgegeben, der darauf verweist, dass Sie das an dieser Stelle einfügen können.

Im folgenden Quelltext sind wichtige Abschnitte farbig hervorgehoben:

```
' Gambas class file

Public timerDataLog As Timer
Public timerReadData As Timer
Public timerSimulation As Timer
Public fTemperature As Float
```

```

Public hTrayIcon As TrayIcon

Public Sub Form_Open()

    FMain.Resizable = False
    FMain.Height = 160
    expRX_TX.Hidden = True
    expRX_TX.Animated = True

    SetRS232Parameters()

    timerSimulation = New Timer As "timerSimulation"
    ' Generate a random temperature value from the specified interval every 100 ms
    timerSimulation.Delay = 100

    timerReadData = New Timer As "timerReadData"
    ' The current temperature value is read out every 1000 ms
    timerReadData.Delay = 1000

    timerDataLog = New Timer As "timerDataLog"
    ' Time interval of data storage in a log file
    timerDataLog.Delay = 3000 * 1

    SetLEDColor(picStatus, "red")
    btnExperimentStop.Enabled = False

    hTrayIcon = New TrayIcon As "hTrayIcon"
    hTrayIcon.Icon = Picture["Symbols/experiment.png"]
    hTrayIcon.Tooltip = ("Long time experiment")
    hTrayIcon.Visible = False
    MTI.CreateMenu()
    ' The menu declared in the module MTI.module is assigned to the tray icon as a context menu.
    hTrayIcon.PopupMenu = MTI.M10MainMenu.Name

End

Public Sub btnExperimentStart_Click()

    GetRS232Parameters()

    If Exist(Application.Path & "rs232.log") Then
        If Message.Question("Should the last measurement log be deleted?", ("Yes"), ("No")) = 1 Then
            Try Kill Application.Path & "rs232.log"
            AddTextToFile("Log of measurement data") & gb.NewLine)
            AddTextToFile("Date") & ": " & Format(Now, "dd. mmmm yyyy")
            AddTextToFile("-----")
        Endif
    Else
        AddTextToFile("Log of measurement data")
        AddTextToFile("Date") & ": " & Format(Now, "dd. mmmm yyyy")
        AddTextToFile("-----")
    Endif

    btnExperimentStart.Enabled = False
    btnExperimentStop.Enabled = True
    SetLEDColor(picStatus, "green")
    lblTemperaturAnzeige.Text = "--- °C"

    timerSimulation.Start()
    timerSimulation.Trigger()
    timerReadData.Start()
    timerReadData.Trigger()
    timerDataLog.Start()
    timerDataLog.Trigger()

    expRX_TX.Hidden = True

    MTI.M11ProgramHideShow.Text = ("Show measuring program")
    MTI.M11ProgramHideShow.Picture = Picture["icon:/16/up"]

    FMain.Hide()
    hTrayIcon.Show()

End

Public Sub btnExperimentStop_Click()

    timerSimulation.Stop()
    timerReadData.Stop()
    timerDataLog.Stop()
    btnExperimentStart.Enabled = True
    btnExperimentStop.Enabled = False
    SetLEDColor(picStatus, "red")

```

```

lblTemperaturAnzeige.Text = "--- °C"
End

Public Sub TimerSimulation_Timer()
    Dim aRange As Float[] = [19.0, 22.0]

    Randomize
    fTemperature = Round(Rnd(aRange[0], aRange[1]), -1)
End

Public Sub timerReadData_Timer()
    lblTemperaturAnzeige.Text = Str(fTemperature) & " °C"
End

Public Sub timerDataLog_Timer()
    AddTextToFile("> " & Format(Now, "hh:nn:ss") & " | " & "T = " & Str(fTemperature) & " °C")
End

Public Sub expRX_TX_Hide()
    FMain.Height = 160
    expRX_TX.Hidden = True
End

Public Sub expRX_TX_Show()
    FMain.Height = 430
    expRX_TX.Hidden = False
End

Public Sub AddTextToFile(Text As String)

    Dim hFile As File
    Dim FilePath As String

    FilePath = Application.Path & "rs232.log"

    Try hFile = Open FilePath For Append
    If Error Then
        Message.Error("File error")
        Return
    Endif
    Print #hFile, Text
    Close #hFile
End

Public Sub hTrayIcon_Click()
    Message.Info("I am the little program help...")
End

Public Sub hTrayIcon_MiddleClick()
    FMain.Close()
End

Public Sub hTrayIcon_Scroll(Delta As Float, Orientation As Integer)
    If Orientation = TrayIcon.Vertical Then Print "VERTICAL"
    Print Delta
End

Private Sub SetRS232Parameters()

    Dim aDataFlow As New String[]

    aDataFlow.Add("None")
    aDataFlow.Add("XON/XOFF")
    aDataFlow.Add("RFR/CTS")
    aDataFlow.Add("RFR/CTS + XON/XOFF")
    cmbFlow.List = aDataFlow

    cmbRS232Port.List = [("/dev/ttyUSB0.virtual")]
    cmbSpeed.List = ["4800", "9600", "2400"]
    cmbParity.Add("None")
    cmbParity.Add("Even")
    cmbParity.Add("Odd")
    cmbDataBits.List = ["8", "7", "6", "5"]
    cmbStopBits.List = ["1", "2"]
End

Private Sub GetRS232Parameters()

    RS232.PortName = cmbRS232Port.Text
    ' Setting the transmission parameters

```

```
RS232.Speed = cmbSpeed.Text
RS232.Parity = cmbParity.Index
RS232.DataBits = cmbDataBits.Text
RS232.StopBits = cmbStopBits.Text
RS232.FlowControl = cmbFlow.Index

End

Private Sub SetLEDColor(picBox As PictureBox, sLEDColor As String)
    picBox.Picture = Picture[ "LED/led_" & sLEDColor & ".svg" ]
End

Public Sub Form_Close()

    If Message.Question("Finish this experiment?", ("Yes"), ("No")) = 1 Then
        hTrayIcon.Hide()
        FMain.Close()
    Endif

End
```