

# **THE K8055 CLASS**

**By Vanderhaegen Bart**

## INTRODUCTION

This manual will show you how to use the K8055 class for your projects. First you need to add the K8055 class to your project (in Visual Basic Express 2008: **Menu Project > Add Existing Item**). Select the K8055 class.

Now the class is added to your project, you can create an Object from it. Below you can see my recommended code to set all the channels of the card.

```
Public Class frmMain

    ' Declare the instance variable
    Private card As K8055

    Private Sub frmMain_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        ' Create a new Object of the K8055 class. card will refer to it.
        card = New K8055()

        ' Create the connection
        card.connect(SK5:=True, SK6:=True)

        ' Check if the card is connected, then set all digital outputs.
        If card.isConnected() Then
            card.setAllDigitalOutputs()
        End If
    End Sub
End Class
```

The K8055 is accessible through the card object. You have several functions, methods and properties that can be called from this card object. This manual gives you an overview of all available methods.

## PUBLIC PROPERTIES

```
Public Property timerEnabled() As Boolean
Public Property timerInterval() As Integer
Public ReadOnly Property isConnected() As Boolean
Public Property cacheEnabled() As Integer
```

- **TimerEnabled:** this property enables or disables the internal Timer which controls the input channels, and raises the Events. The class won't generate Events when the `timerEnabled` is set to `False`.
- **TimerInterval:** The interval of the internal Timer, expressed in milliseconds. The default value is 100 ms.
- **isConnected** will return `True` when there is a connection with a K8055 card.
- **cacheEnabled:** this property enabled/disables

## DIGITAL OUTPUTS

```
Public Sub setDigitalOutput(ByVal channel As Integer)
Public Sub clearDigitalOutput(ByVal channel As Integer)
Public Function getDigitalOutput(ByVal channel As Integer)
Public Sub clearAllDigitalOutputs()
Public Sub setAllDigitalOutputs()
```

These functions will control the Digital Outputs. You can choose to set all digital outputs, or specify a `channel` parameter when calling the functions.

When setting or clearing one of the outputs, its state is saved to a variable in the K8055 class. You can retrieve one of the states by calling the `getDigitalOutput()` function. This function doesn't read the state from the K8055 (the card doesn't support that), but from the variable that was saved earlier.

You should be aware there might be situations where the `getDigitalOutput()` doesn't reflect the actual state of the K8055 output (For example: in the case of a hardware error).

## DIGITAL INPUTS

```
Public Function getDigitalInput(ByVal channel As Integer) As Boolean  
Public Function getAllDigitalInput() As Integer
```

These functions will read the digital inputs from the card. You can either read all digital inputs, or only a selected channel.

*ATTENTION! These functions are influenced by the 'caching' option.*

*The K8055 class reads the states of the analog/digital inputs at a fixed time interval. The interval of the timer is determined by the value of the `timerInterval()` property, which was described earlier in this manual.*

*Each time the K8055 class reads the values, they are stored into a cache. If caching is enabled, the digital input functions will read the state of the inputs from the cache, and not from the K8055 card. If you choose the value of the `timerInterval()` wisely (100 ms), this shouldn't be a problem.*

*If you don't like the Caching-feature, you can just turn it off by setting the `cacheEnabled()` property to false. The caching is also disabled if you turn off the build-in Timer of the class by setting the `timerEnabled()` property False.*

## COUNTERS

```
Public Function readCounter_(ByVal CounterNr As Integer) As Integer  
Public Sub resetCounter_(ByVal CounterNr As Integer)  
Public Sub setCounterDebounceTime_(ByVal CounterNr As Integer, ByVal debounceTime As Integer)
```

These functions are controlling the counters on the K8055 card. They work exactly the same as in the original k8055d.dll.

## CONNECTION

```
Public Function connect(ByVal cardAddress As Integer) As Boolean
Public Function connect(ByVal SK5 As Boolean, ByVal SK6 As Boolean) As
Public Sub disconnect()
Public Function getVersion() As Integer
```

The connection functions will make the connection to the K8055 card. You can create an Object of the K8055 class, but it won't be connected to the K8055 card unless you call one of these functions. You either have to specify the card address, or the jumper settings (SK5 and SK6).

*ATTENTION: You can create multiple objects of the K8055 class, but only one can make an actual connection to the K8055 card. An Exception is generated when you try to connect a second K8055 object.*

Each time you create a connection to the K8055 card, **all outputs are cleared**. I had to do this because otherwise my class can't determine the correct states of the outputs.

## ANALOG OUTPUTS

```
Public Sub setAllAnalogOutputsToMax()
Public Sub clearAllAnalogOutputs()
Public Sub setAnalogOutputToMax(ByVal channel As Integer)
Public Sub clearAnalogOutput(ByVal channel As Integer)
Public Sub setAnalogOutputToValue(ByVal channel As Integer, ByVal value As Integer)
Public Sub setAllAnalogOutputsToValue(ByVal value1 As Integer, ByVal value2 As Integer)
Public Function getAnalogOutput(ByVal channel As Integer) As Integer
```

These functions are controlling the analog outputs. You can set them to any value between 0 and 255.

Just like with the digital outputs, the analog output value is also written to a variable in the class. You can retrieve a value by calling the `getAnalogOutput()` function.

## ANALOG INPUTS

```
Public Function ReadAnalogInput(ByVal channel As Integer)
Public Sub readAllAnalogInputs(ByRef data1 As Integer, ByRef data2 As Integer)
```

These functions can read the value of the analog input channels.

*ATTENTION! These functions are influenced by the 'caching' option. See the section 'Digital Inputs' to read the description of the caching feature.*

## CONSTRUCTORS

```
Public Sub New()
Public Sub New(ByVal enableEvents As Boolean)
Public Sub New(ByVal enableEvents As Boolean, ByVal interval As Integer)
```

The class constructors are called when you create a new instance of the class. They specify the options you can pass to the new Object. This is an example code to create a new Object without specifying parameters:

```
Dim card As K8055
card = New K8055()
```

This is a second example code to create a new Object, while specifying the enableEvents and Interval options:

```
Dim card As K8055
card = New K8055(True,100)
```

## EVENTS

The K8055 class supports three types of Events. An Event is raised when something happens at the K8055 card.

```
Public Event AnalogInputChanged(ByVal channel As Integer, ByVal Value As Integer)
```

This Event is raised when one of the Analog Input channels was changed. The Event will specify the changed channel, and the new value of that channel. If more than one channel changes at the same time, the `AnalogInputChanged` is fired two times.

```
Public Event digitalInputChanged(ByVal channel As Integer, ByVal Value As Boolean)
```

This Event is raised when one of the Digital Input channels was changed. The Event will specify the changed channel, and the new value of that channel. If more than one channel changes at the same time, the `digitalInputChanged` is fired multiple times.

```
Public Event readCycleCompleted()
```

This Event is raised when the Internal Timer of my class has completed one cycle. You will probably never need it. I have added it for debugging purposes.

## USING EVENTS

You might be wondering how you can use events on my K8055 class. This is how you can do it: First you need to make an instance variable (a variable in your class that is defined outside any sub/function, and therefore it will become accessible from any sub/function in your program). You need to specify the `WithEvents` option while declaring it. Example code:

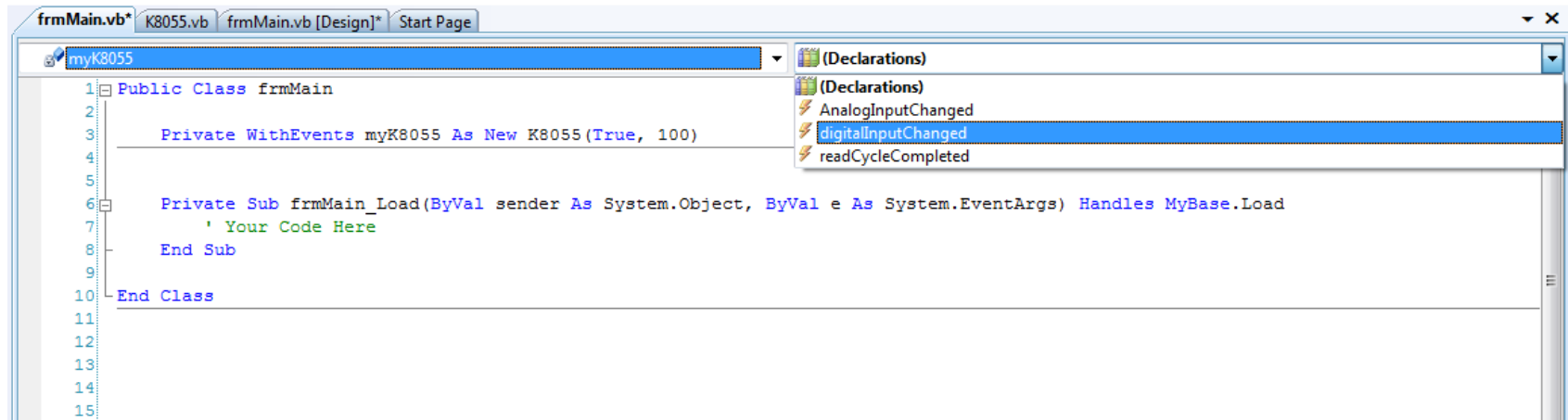
```
Public Class frmMain

    Private WithEvents myK8055 As New K8055(True, 100)

    Private Sub frmMain_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        ' Your Code Here
    End Sub

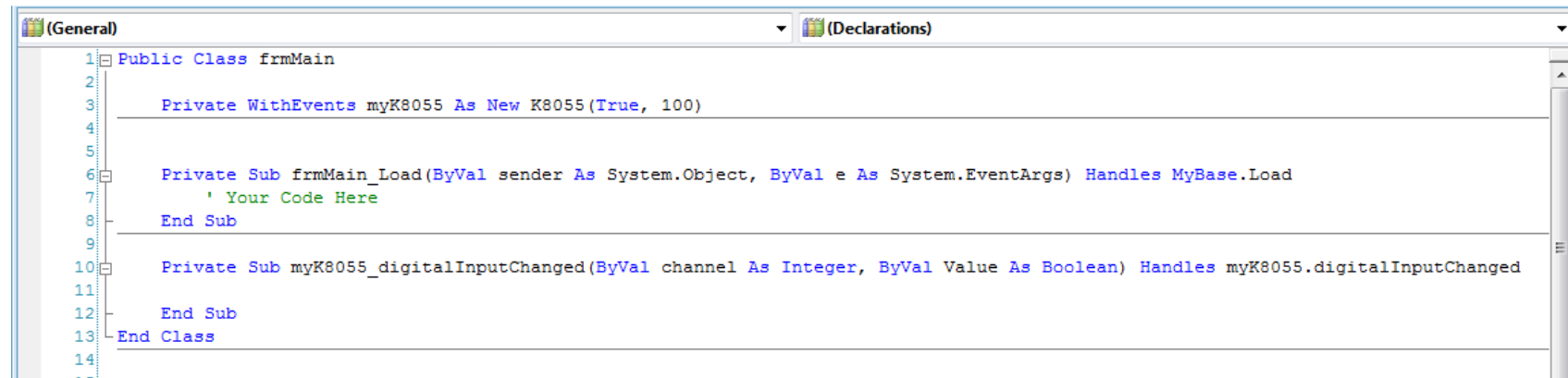
End Class
```

Once you do so, your `myK8055` object will appear in the list with class names (left). At the right you now can choose between the three possible events. You just have to click one of these events, and the needed Sub will be added to your code automatically.





The needed code was added to the code:



```
1 Public Class frmMain
2
3     Private WithEvents myK8055 As New K8055(True, 100)
4
5
6     Private Sub frmMain_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
7         ' Your Code Here
8     End Sub
9
10    Private Sub myK8055_digitalInputChanged(ByVal channel As Integer, ByVal Value As Boolean) Handles myK8055.digitalInputChanged
11
12    End Sub
13 End Class
14
```