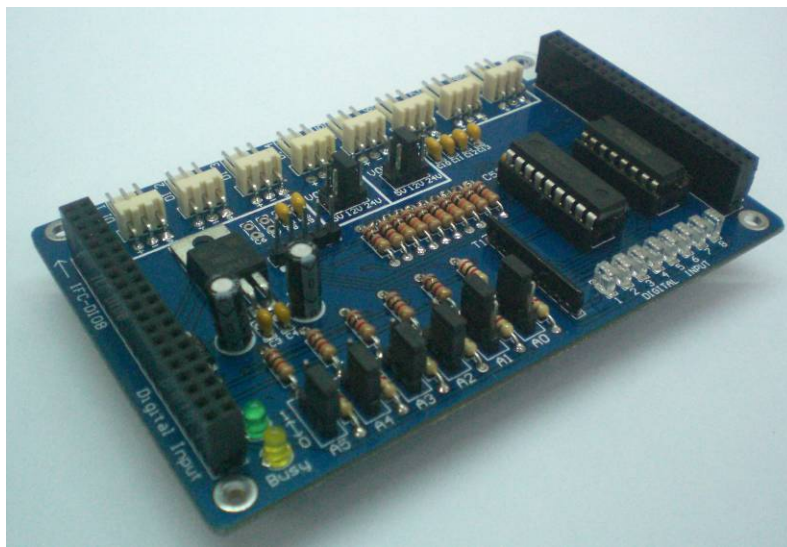




## **IFC-DI08**

### **Interface Free Controller**

### **Digital Input**



## **Card Library Functions for Visual C#**

## **Express and Visual Basic Express**

**V1.0**

**Apr 2009**

Information contained in this publication regarding device applications and the like is intended through suggestion only and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. No representation or warranty is given and no liability is assumed by Cytron Technologies Incorporated with respect to the accuracy or use of such information, or infringement of patents or other intellectual property rights arising from such use or otherwise. Use of Cytron Technologies's products as critical components in life support systems is not authorized except with express written approval by Cytron Technologies. No licenses are conveyed, implicitly or otherwise, under any intellectual property rights.

### Function Prototype for Digital Input (DI08)

This document explains the function prototype for controlling IFC-DI08 using PC through IFC-CI00. User may also use ‘object browser’ under Microsoft Visual C# to view the summary, parameter and return value description of IFC-DI08 function prototype. User need to add reference ‘ifc\_ci.dll’ and ‘ifc\_di.dll’ for IFC-CI00 and IFC-DI08 card in order to control/communicate IFC-DI08 using PC. Please note that before user start the programming, user need to initialize the ‘ifc.ifc\_ci’ and ‘ifc.ifc\_di’ in order to use the functions to control IFC-DI08. Example of creating a ‘ifc.ifc\_ci’ class called ‘ifc1’ and ‘ifc.ifc\_di’ class called di1:

```
static ifc.ifc_ci ifc1 = new ifc.ifc_ci(74);
ifc.ifc_di di1 = new ifc.ifc_di(ifc1, 4);
```

For ‘ifc.ifc\_ci’ class, user need to specified the COM Port that is connected to IFC-CI00 and for ‘ifc.ifc\_di’ class, user need to specified the IFC-CI00 in use and also the address for IFC-DI08. Please make sure that the address must be unique and different with other IFC card in the IFC system.

Function Prototype	Example	Summary	Parameter Description	Return Value
<b>void di_c1clr()</b>	<b>di1.di_c1clr()</b>	To clear Counter 1 value.		
<b>void di_c1con(bool on_off)</b>	<b>di1.di_c1con(true)</b>	To configure Counter 1 on IFC-DI08.	<i>on_off</i> : True to turn ON Counter 1 and false to turn OFF Counter 1. (bool)	
<b>void di_c1con(int on_off)</b>	<b>di1.di_c1con(1)</b>	To configure Counter 1 on IFC-DI08.	<i>on_off</i> : 1 to turn ON Counter 1 and 0 to turn OFF Counter 1. (int)	
<b>void di_c1con(byte on_off)</b>	<b>di1.di_c1con(0)</b>	To configure Counter 1 on IFC-DI08.	<i>on_off</i> : 1 to turn ON Counter 1 and 0 to turn OFF Counter 1. (byte)	
<b>di_c1val()</b>	<b>di1.di_c1val()</b>	To read Counter 1 value.		Counter 1 value in 16-bit. (int)

<b>void di_c2clr()</b>	<b>di1.di_c2clr()</b>	To clear Counter 2 value.		
<b>void di_c2con(<a href="#">bool</a> on_off)</b>	<b>di1.di_c2con(<a href="#">true</a>)</b>	To configure Counter 2 on IFC-DI08.	<i>on_off</i> : True to turn ON Counter 2 and false to turn OFF Counter 2. (bool)	
<b>void di_c2con(<a href="#">int</a> on_off)</b>	<b>di1.di_c2con(<a href="#">1</a>)</b>	To configure Counter 2 on IFC-DI08.	<i>on_off</i> : 1 to turn ON Counter 2 and 0 to turn OFF Counter 2. (int)	
<b>void di_c2con(<a href="#">byte</a> on_off)</b>	<b>di1.di_c2con(<a href="#">0</a>)</b>	To configure Counter 2 on IFC-DI08.	<i>on_off</i> : 1 to turn ON Counter 2 and 0 to turn OFF Counter 2. (byte)	
<b>di_c2val()</b>	<b>di1.di_c2val()</b>	To read Counter 2 value.		Counter 2 value in 16-bit. (int)
<b>di_read(<a href="#">int</a> selection)</b>	<b>di1.di_read(<a href="#">8</a>)</b>	To read the digital input of a selected port.	<i>selection</i> : Selected input port to read, in range of 1 to 8. (int)	Return the digital input status, true if the digital input status is cleared. (bool)
<b>di_read(<a href="#">byte</a> selection)</b>	<b>di1.di_read(<a href="#">5</a>)</b>	To read the digital input of a selected port.	<i>selection</i> : Selected input port to read, in range of 1 to 8. (byte)	Return the digital input status, true if the digital input status is cleared. (bool)
<b>di_read_all()</b>	<b>di1.di_read_all()</b>	To read the all digital input of IFC-DI08.		Return the digital input status in one byte. Bit 0 represent DI_1... and bit 7 represent DI_8. Representative bit clear if the digital input status is cleared. (byte)

<b>ifc_di</b> ( <a href="#">ifc.ifc_ci</a> ifc_ci, <a href="#">int</a> address)	<b>ifc.ifc_di</b> ( <a href="#">ifc1</a> , <a href="#">5</a> )	Initializes a new instance of the ifc.ifc_di class using the specified ifc.ifc_ci and address for IFC-DI08.	<i>ifc_ci</i> : ifc.ifc_ci in use. <i>address</i> : Address for IFC-AI08, in range of 0 to 63. (int)	
<b>ifc_di</b> ( <a href="#">ifc.ifc_ci</a> ifc_ci, <a href="#">byte</a> address)	<b>ifc.ifc_di</b> ( <a href="#">ifc1</a> , <a href="#">10</a> )	Initializes a new instance of the ifc.ifc_di class using the specified ifc.ifc_ci and address for IFC-DI08.	<i>ifc_ci</i> : ifc.ifc_ci in use. <i>address</i> : Address for IFC-AI08, in range of 0 to 63. (byte)	

**Table 1 Function Prototype for DI08 card**

*Prepared by*  
***Cytron Technologies Sdn. Bhd.***  
19, Jalan Kebudayaan 1A,  
Taman Universiti,  
81300 Skudai,  
Johor, Malaysia.

*Tel:* +607-521 3178

*Fax:* +607-521 1861

*URL:* [www.cytron.com.my](http://www.cytron.com.my)

*Email:* [support@cytron.com.my](mailto:support@cytron.com.my)  
[sales@cytron.com.my](mailto:sales@cytron.com.my)