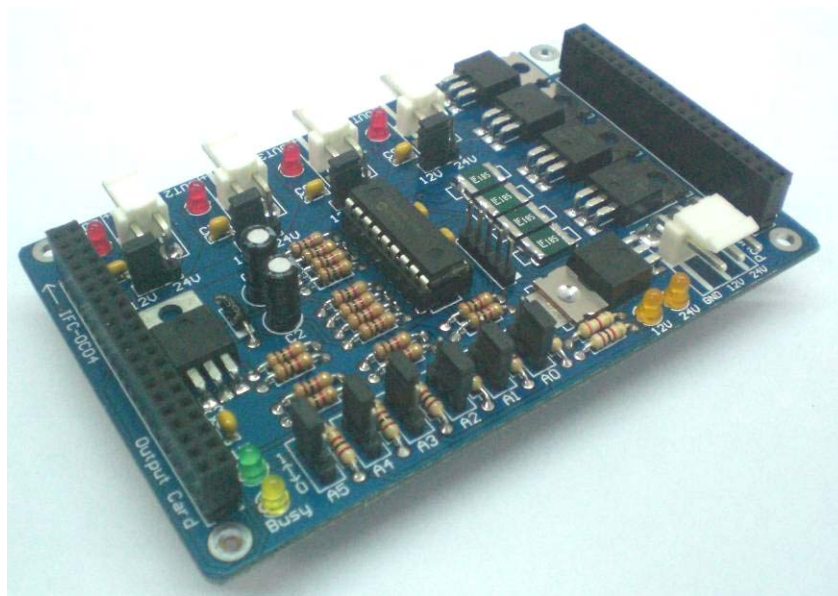




IFC-OC04

Interface Free Controller Output Card



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contained in this publication regarding device applications and the like is intended through suggestion only and may be superseded by updates.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 Output Card (OC04)

This document explains the function prototype for controlling IFC-OC04 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-OC04 function prototype. User need to add reference ‘ifc_ci.dll’ and ‘ifc_oc.dll’ for IFC-CI00 and IFC-OC04 card in order to control/communicate IFC-OC04 using PC. Please note that before user start the programming, user need to initialize the ‘ifc.ifc_ci’ and ‘ifc.ifc_oc’ in order to use the functions to control IFC-OC04. Example of creating a ‘ifc.ifc_ci’ class called ‘ifc1’ and ‘ifc.ifc_oc’ class called oc1:

```
static ifc.ifc_ci ifc1 = new ifc.ifc_ci(74);
ifc.ifc_oc oc1 = new ifc.ifc_oc(ifc1,3);
```

For ‘ifc.ifc_ci’ class, user need to specified the COM Port that is connected to IFC-CI00 and for ‘ifc.ifc_oc’ class, user need to specified the IFC-CI00 in use and also the address for IFC-OC04. 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
ifc_oc(ifc.ifc_ci ifc_ci, byte address)	ifc.ifc_oc(ifc1, 3)	Initializes a new instance of the ifc.ifc_oc class using the specified ifc.ifc_ci and address for IFC-OC04.	<i>ifc_ci</i> : ifc.ifc_ci in use. <i>address</i> : Address for IFC-OC04, in range of 0 to 63. (byte)
ifc_oc(ifc.ifc_ci ifc_ci, int address)	ifc.ifc_oc(ifc1, 3)	Initializes a new instance of the ifc.ifc_oc class using the specified ifc.ifc_ci and address for IFC-OC04.	<i>ifc_ci</i> : ifc.ifc_ci in use. <i>address</i> : Address for IFC-OC04, in range of 0 to 63. (int)
void oc_out(byte data)	oc1.oc_out(4)	To configure all output.	<i>data</i> : Output value in one byte. Bit 0 represent OUT1, bit 1 represent OUT2, bit 2 represent OUT3, and bit 3 represent OUT4. (byte)
void oc_out(int data)	oc1.oc_out(8)	To configure all output.	<i>data</i> : Output value in one integer. Bit 0 represent OUT1, bit 1 represent OUT2, bit 2 represent OUT3, and bit 3 represent OUT4. (int)

void oc_out1 (byte data)	oc1.oc_out1 (1)	To configure output condition for OUT1.	<i>data</i> : 1 to activate OUT1 and 0 to deactivate OUT1. (byte)
void oc_out1 (int data)	oc1.oc_out1 (1)	To configure output condition for OUT1.	<i>data</i> : 1 to activate OUT1 and 0 to deactivate OUT1. (int)
void oc_out1 (bool data)	oc1.oc_out1 (false)	To configure output condition for OUT1.	<i>data</i> : True to activate OUT1 and false to deactivate OUT1. (bool)
void oc_out2 (byte data)	oc1.oc_out2 (1)	To configure output condition for OUT2.	<i>data</i> : 1 to activate OUT2 and 0 to deactivate OUT2. (byte)
void oc_out2 (int data)	oc1.oc_out2 (1)	To configure output condition for OUT2.	<i>data</i> : 1 to activate OUT2 and 0 to deactivate OUT2. (int)
void oc_out2 (bool data)	oc1.oc_out2 (false)	To configure output condition for OUT2.	<i>data</i> : True to activate OUT2 and false to deactivate OUT2. (bool)
void oc_out3 (byte data)	oc1.oc_out3 (0)	To configure output condition for OUT3.	<i>data</i> : 1 to activate OUT3 and 0 to deactivate OUT3. (byte)
void oc_out3 (int data)	oc1.oc_out3 (0)	To configure output condition for OUT3.	<i>data</i> : 1 to activate OUT3 and 0 to deactivate OUT3. (int)
void oc_out3 (bool data)	oc1.oc_out3 (true)	To configure output condition for OUT3.	<i>data</i> : True to activate OUT3 and false to deactivate OUT3. (bool)
void oc_out4 (byte data)	oc1.oc_out4 (0)	To configure output condition for OUT4.	<i>data</i> : 1 to activate OUT4 and 0 to deactivate OUT4. (byte)

void oc_out4(int data)	oc1.oc_out4(1)	To configure output condition for OUT4.	<i>data:</i> 1 to activate OUT4 and 0 to deactivate OUT4. (int)
void oc_out4(bool data)	oc1.oc_out4(true)	To configure output condition for OUT4.	<i>data:</i> True to activate OUT4 and false to deactivate OUT4. (bool)
void oc_pwm(byte data)	oc1.oc_pwm(180)	To configure PWM output on OUT1.	<i>data:</i> PWM value in range of 0 to 255. (byte)
void oc_pwm(int data)	oc1.oc_pwm(255)	To configure PWM output on OUT1.	<i>data:</i> PWM value in range of 0 to 255. (int)

Table 1 Function Prototype for OC04 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

Email: support@cytron.com.my
sales@cytron.com.my