

Quick Start Guide



IP Modules & Carriers

Windows® 10

Version 1.0 October, 2019

Contents

Introduction.....	3
Driver Installation.....	3
Step 1: Go to the Device Manager	3
Step 2: Prepare for Installation	3
Step 3: Install Carrier driver/s	3
Step 4: Install the Module driver/s	4
Step 5: Installation complete	4
User Application	4
Menu options	4
(C/S) Continous/Stop	5
(F/O) Forever/Once	5
(L/N) Limit/Normal	5
(D/T) Debug/Test.....	5
(H) Help	5
(E) Exit.....	5
Running tests	5
Using Breakpoints	5
Adding tests.....	6
Step 1: Open Solution in Visual Studio	6
Step 2: Add a Source File	6
Step 3: Define the Function	6
Step 4: Declare the Function	6
Step 5: Add test to menu	6
Step 6: Build	6
IP GENERIC.....	7
FAQ	8
Why am I getting an error message when installing the driver?.....	8
Why am I getting errors saying "IOCTL_____ failed: ##"?	8
Why won't the test provided by DE work for me?.....	8
Can I use DE drivers for an IP module that was not provided by DE?.....	8
What SW can be used as reference?.....	8



Introduction

This quick start guide is intended to help you get started with the use of the Dynamic Engineering (DE) IndustryPack (IP) carriers and modules. It gives a brief description of driver installation, the use of the user applications, and an explanation about the IP Generic driver/user application. Answers to a few FAQs can be found at the end.

Manuals for the hardware and software for all carriers and modules can be found on the DE website¹. It should be noted that DE modules can be used with any IndustryPack compliant carrier. The included software driver packages are intended to be used with Dynamic Engineering carrier hardware and supporting driver. The module driver will not install if the carrier is not one provided by DE.

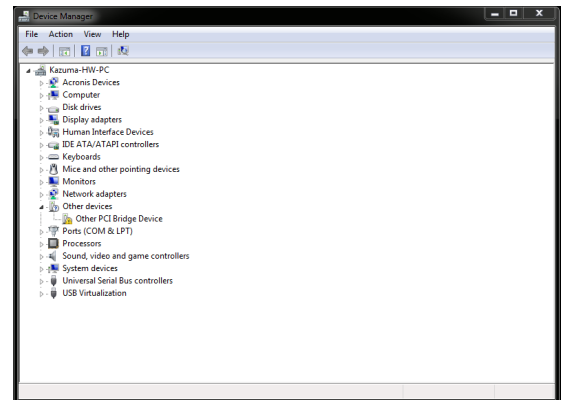
Driver Installation

Step 1: Open the Device Manager

The device manager can be found by typing 'device manager' in the search box found on the left side of the task bar. By default, the task bar can be found on the bottom of the Windows 10 screen.

Step 2: Prepare for Installation

If no other IP carrier or module has ever been installed it should look like the example to the right. If the device is not found it may be necessary to Scan for Hardware changes. If a driver is already installed it is recommended that the driver/s be uninstalled. This is done by right clicking the device and selecting uninstall. Make sure to click the "Delete the driver software for this device" check box before confirming the Device uninstall.



Step 3: Install Carrier driver/s

Right click on the "Other PCI Bridge Device". Click on Update Driver Software and navigate to the location of the IP Carrier drivers. Click Next and Install. A message saying that the software has successfully installed will appear. Windows will immediately try to install the driver software for the modules. Allow the search to finish before continuing.

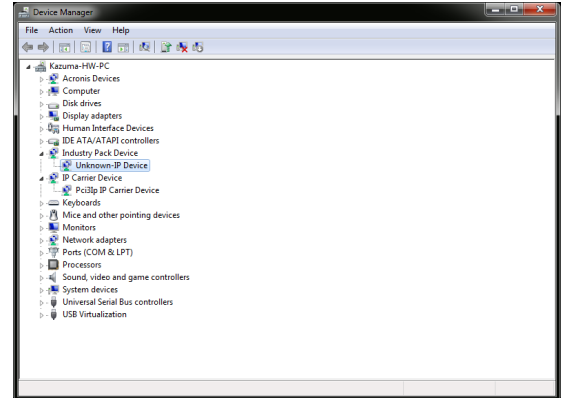
¹ <http://www.dyneng.com/industryack.html>

Note: Clicking on Properties -> Details -> Hardware IDs will show the ID of the selected hardware.

Step 4: Install the Module driver/s

If the Carrier driver was successfully installed a new device/s should appear in “Other devices”. The new device/s should show the name of the carrier and the slot in which the hardware is installed. Follow the same procedure as the one in the previous step to install the module driver/s.

Note: Once installed the device manager will show the name of the installed driver. If a driver does not exist for the hardware the device manager will install the IP Generic driver and will show “Unknown-IP Device”.

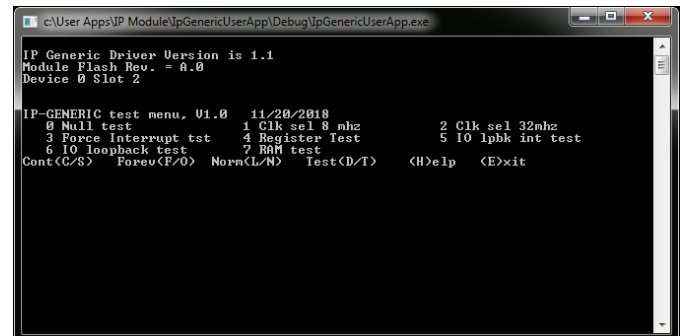


Step 5: Installation complete

Once installation is complete the name of the IP carrier and module will appear in the device manager. In this example the PCI3IP and the IP Generic drivers are installed.

User Application

The UserApp is a stand-alone code set with a simple, and powerful menu plus a series of “tests” that can be run on the installed hardware. Each of the tests execute calls to the driver, pass parameters and structures, and get results back. With the sequence of calls demonstrated, the functions of the hardware are utilized for loop-back testing. The software is used for manufacturing test at DE. Below is a quick run through of some of the important features of the User App.



Menu options

At the bottom of the menu you will see 6 sets of letter/s inside parentheses. To use/switch between different modes simply press the letter shown in the parentheses then enter. For example to exit the UserApp press ‘e’ then hit enter. Below is a description of the various modes.



(C/S) Continuous/Stop

This defaults to Continuous mode. When set to Stop mode the test will stop on the failure flag. When set to Continuous mode the test will continually run even after a failure. When in Continuous mode, press the space bar to stop test. The UserApp will complete the test it is currently running then stop.

(F/O) Forever/Once

This defaults to Forever mode. When set to Once mode, the test will run once and stop. When set to Forever mode it will run until the space bar is pressed.

(L/N) Limit/Normal

This defaults to Normal mode. To use the limit mode feature type "L" followed by a number. When in Limit mode the test will run for the number of loops entered.

(D/T) Debug/Test

This defaults to Test mode. When in Debug mode each time a failure occurs the error message will be printed to screen. When in Test mode the error message will only be printed if the error is different from the previous one.

(H) Help

In the current version of the UserApp this feature is not used.

(E) Exit

Press 'E' then enter to exit the UserApp

Running tests

To run a test, simply press the number of the test then enter. The test will run according to the menu options selected, as described above. Note that some of the tests require a loop back fixture in order to pass.

Using Breakpoints

Breakpoints are a debugging tool that can pause the debugger execution. This can be used to step through lines of code when debugging. The Microsoft website² has a useful guide that can help getting started with breakpoints.

² <https://docs.microsoft.com/en-us/visualstudio/debugger/using-breakpoints?view=vs-2017>

Adding tests

A useful feature of the UserApp is the ability to add tests. Below are step by step instructions to adding an “EmptyTest” test.

Step 1: Open Solution in Visual Studio

Start Microsoft Visual Studio. Click File -> Open -> Project/Solution. Navigate then double click on the IP Modules ____.sln file. In the example to the right we used the IpGenericUserApp.sln.

Step 2: Add a Source File

In the Solution Explorer in Visual Studio right click the Source Files. Navigate to Add then to New Item. Under name type ‘EmptyTest.c’. Copy the includes and the HANDLE from a different file.

Step 3: Define the Function

In the file you created define the function as shown in the example to the right.

Step 4: Declare the Function

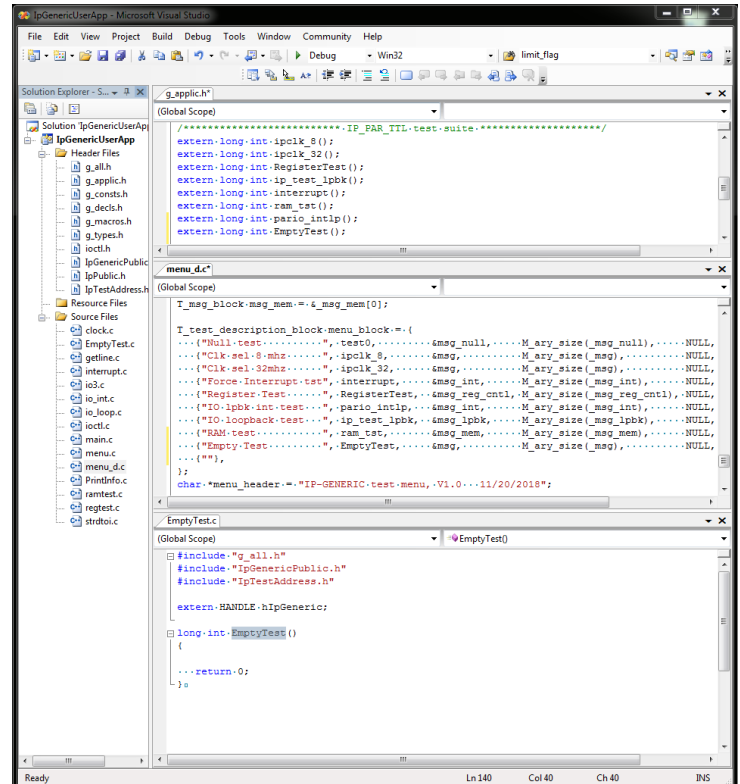
Declare the function as shown in the g_applic.h file in the example.

Step 5: Add test to menu

Add the test to the menu_block in the menu_d.c file as shown in the example. In this example the default error message block is used by this can be changed when needed.

Step 6: Build

Build then open the executable. You will notice the Empty Test has been added to the test menu.



```
g_applic.h
.....IP_PAR_TTL.test.suite.....//
extern long int ipclk_8();
extern long int ipclk_32();
extern long int RegisterTest();
extern long int ip_test_ipbk();
extern long int interrupt();
extern long int ram_test();
extern long int pario_intip();
extern long int EmptyTest();

menu_d.c
T_msg_block msg_mem = { &msg_mem[0] };
T_test_description_block menu_block = {
    { "Null-test", test0, &msg_null, M_ary_size(msg_null), NULL,
    { "Clk-sel-8-mhz", ipclk_8, &msg, M_ary_size(msg), NULL,
    { "Clk-sel-32mhz", ipclk_32, &msg, M_ary_size(msg), NULL,
    { "Force-Interrupt-test", interrupt, &msg_int, M_ary_size(msg_int), NULL,
    { "Register-Test", RegisterTest, &msg_reg_ctrl, M_ary_size(msg_reg_ctrl), NULL,
    { "IO-ipbk-int-test", ip_test_ipbk, &msg_int, M_ary_size(msg_int), NULL,
    { "IO-loopback-test", ip_test_ipbk, &msg_ipbk, M_ary_size(msg_ipbk), NULL,
    { "RAM-test", ram_test, &msg_mem, M_ary_size(msg_mem), NULL,
    { "Empty-Test", EmptyTest, &msg, M_ary_size(msg), NULL,
    { "" },
};
char *menu_header = "IP-GENERIC-test-menu, V1.0...11/20/2018";

EmptyTest.c
#include "g_all.h"
#include "IpGenericPublic.h"
#include "IpTestAddress.h"

extern HANDLE hIpGeneric;

long int EmptyTest()
{
    return 0;
}
```

IP GENERIC

DE manufactures and provides drivers for a number of IP carriers and modules. The full list of them can be found on our website¹. When a driver is installed for an IP module where the hardware ID is recognized, the UserApp for that specific module can be used to provide a running start. The UserApp will show examples of how the IO Controls (IOCTLS) are used. The IOCTLS provided will give the user all the necessary controls for executing calls to the driver.

When the hardware ID is not recognized while installing the driver the IP Generic driver will be installed. The address map of the hardware will be unknown. Instead of using specific IOCTLS, the IOCTLS called GetData and SetData are used. These two IOCTL require a structure as input. The structures include, among other things, the local address offset from the base address. Using these two IOCTLS any address within the IP can be accessed.

The IpGenericUserApp was written using the IpTest hardware. This UserApp demonstrates how all the functionality of the IpTest driver/UserApp can be replicated by the IpGeneric driver/UserApp. You will notice in the IpTestAddress.h file, the address offsets for the IPTest are defined. Creating a similar file with the address offset for the IP module of interest will be necessary when using this UserApp for a different module.



FAQ

Why am I getting an error message when installing the driver?

This is most likely due to an outdated driver. Make sure to get the most up to date driver from DE then follow the instructions above for driver installation.

Why am I getting errors saying “IOCTL_____ failed: ##”?

These IOCTL error can occur in the software provided by DE if the UserApp and the driver are not both up to date. If this occurs please uninstall both carrier and module drivers and try again.

An IOCTL error can occur in software that is not provided by DE. In this case the problem is harder to diagnose. There are a multitude of mistakes that can cause this. The System Error Codes³ can help narrow down the issue, but sometimes the description is vague. The easiest solution may be to copy and paste the IOCTL from a part of the software that is working. Make sure that all data types that are passed in or out are of the right type.

Why won't the test provided by DE work for me?

Some of the tests we provide need a lookback fixture to be installed in order to pass. The hardware manual shows the pinout for the loopback fixtures.

Can I use DE drivers for an IP module that was not provided by DE?

Yes, the IP Generic driver will be installed. Refer to the IP GENERIC section above for more details.

What SW can be used as reference?

Each driver has a UserApp associated with it. This UserApp is the one used at DE for manufacturing test. It can be used to provide a running to writing software.

³ <https://docs.microsoft.com/en-us/windows/desktop/debug/system-error-codes--0-499->