Subject: IDL and C-
Posted by rsmith1 on Thu, 20 May 2004 01:44:25 GMT
View Forum Message <> Reply to Message

Hello-

I am attending a small undergraduate university at I am trying to I am trying to interface a CCD with a computer through an NI IMAQ card in IDL.
The only examples they give are in C or through the use of labview. I
need to be able to snap images from the card inside of IDL. The way I am currently going about the problem is to use the code they have given
inside of C and then reference it using the IDL call_external function.
The C code is compiling fine now and the IDL code seems to be referencing the C code, however there are some variable declaration issues. Here is my code in IDL:

```
Pro TestDLL2

image = bytarr(640,480)
help, image
image = call_external("C:\11ryan\temp\Debug\testDLL.dll","testDLL ")
help, image
END
```

This references the following C code:

```
#include <stdio.h>
#include <windows.h>



#include ".\resource.h"
#include "IDL_export.h"
#define _NIWIN

#include "nitypes.h"
#include "niimaq.h"

#include "sample.h"
#include "acqfuncs.h"

extern "C" __declspec(dllexport) char* testDLL(void)
{
```

```
  INTERFACE_ID interfaceID;
  SESSION_ID sessionID;
  Int8* buffer = NULL;
  IMG_ERR error;
  uInt32  top, left, height, width, rowBytes;


  //open an interface and start a session
  error = imgInterfaceOpen("img0", &interfaceID);
  error = imgSessionOpen(interfaceID, &sessionID);

  //pass a pointer to a NULL pointer and the driver will allocate
  //a buffer of the appropriate size for you.
  error = imgSnap(sessionID, &buffer);

  //get the image dimensions.  These default dimensions depend on the
type
  //of camera that is currently configured
  error = imgSessionGetROI(sessionID, &top, &left, &height, &width);
  error = imgGetAttribute(sessionID, IMG_ATTR_ROWBYTES, &rowBytes);




  //process function here


  //close this interface and free all resources associated with it,
  //such as the buffer that was allocated by the driver during
imgSnap


return buffer;
  }
```

The c code was given in a manual from NI, however I left out the final portion where they close the buffer and I made some changes in order to
get it to compile as well as pass information through to IDL. As it stands now, I have declared an array inside of IDL the size of my image. When I use the call_external function IDL changes the variable type on the fly and returns a long int. I think that this is a pointer
to the memory address of the start of the buffer. I am unsure as to

how to take this information and obtain the image from it inside of
IDL. Any help you can offer is appreciated. Thanks in advance-

-Ryan

Subject: Re: IDL and C-
Posted by rsmith1 on Mon, 24 May 2004 20:02:26 GMT
View Forum Message <> Reply to Message

JD-

Thanks a bunch for the reply. I tried plugging in the code you gave
me and I got a variable redifinition error due to the fact that buffer
is defined earlier as follows:
 Int8* buffer = NULL;
I then took out the char *buffer and tried the looping statment to
fill the array only to have it compile but crash IDL when the
call_external is executed. Any other thoughts? Thanks again for your
help-

-Ryan

JD Smith <jdsmith@as.arizona.edu> wrote in message
news:<pan.2004.05.20.21.49.28.290847@as.arizona.edu>...
> On Thu, 20 May 2004 13:50:43 -0700, Ryan Smith wrote:
>
>> M. Katz-
>>
>> Thanks a bunch for your response. I tried the code and ran into more
>> issues. Upon using the call_external it simply converts the image
>> variable back to a long since IDL does its variable type declarations
>> on the fly. I then tried the following code:
>> image = ptr_new(bytarr(call_external("C:\Users\11ryan\CIDLFinal\testDLL\Debug\testDLL.dll
","testDLL")))
>> window, xsize=640, ysize=480 ;--- open a window for display
>> help, image
>> tvscl, *image
>>
>> And the help, image says that it is a pointer, but when trying to
>> display it i get an error saying TVSCL: Width and Height must be less
>> than 32000. It looks as if it is trying to take the value and use it
>> as a dimension instead. Any more advice on what I could try? thanks
>> again for all the help-
>>
>> -Ryan
>>
>> MKatz843@onebox.com (M. Katz) wrote in message

news:<4a097d6a.0405192255.693cd62b@posting.google.com>...
>>> This is just a guess, but you might try the following.
>>>
>>>   ;--- declare image as a pointer to an array of byte type
>>> image = ptr_new(bytarr(640,480))
>>> image = call_external("C:\11ryan\temp\Debug\testDLL.dll","testDLL ")
>>> window, xsize=640, ysize=480 ;--- open a window for display
>>> tvscl, *image ;--- scale and display the contents of the image pointer
>>>
>>> after the call_external, you might also issue
>>> print, image
>>> If it returns something like this <PtrHeapVar1> then it's certainly a pointer.
>>> M. Katz
>
> You can't just return a raw character pointer from C and expect IDL to
> convert it into an IDL array variable. The traditional way to do this
> is first make an array in IDL, pass it by reference to the function
> via call_external, and copy the camera data over to it before
> returning. Something like:
>
> image=bytarr(1024,1024)
> ret = call_external("C:\11ryan\temp\Debug\testDLL.dll","testDLL ",image)
>
> and in the C code:
>
> int _blah _blah newtestDLL(int argc, void *argv[]) {
>   char *buffer,*out;
>   int i;
>   /* Grab buffer from the camera */
>   ...
>   /* Copy to output array */
>   out=(char *)argv[0]; /* This points to the IDL image variable's data */
>   for(i=0;i<1024*1024;i++) out[i]=buffer[i];
>   return 1;
> }
>
> Note that IDL pointers and C pointers are completely different beasts
> which share almost nothing in common (IDL's could more properly have
> been called "references").
>
> JD

Subject: Re: IDL and C-
Posted by JD Smith on Tue, 25 May 2004 01:17:49 GMT
View Forum Message <> Reply to Message

On Mon, 24 May 2004 13:02:26 -0700, Ryan Smith wrote:

> JD-
>
> Thanks a bunch for the reply. I tried plugging in the code you gave
> me and I got a variable redifinition error due to the fact that buffer
> is defined earlier as follows:
> Int8* buffer = NULL;
> I then took out the char *buffer and tried the looping statment to
> fill the array only to have it compile but crash IDL when the
> call_external is executed. Any other thoughts? Thanks again for your
> help-

Did `buffer' actually get filled up by camera data and did the loop
stay within the bounds of buffer and the output passed in from IDL?
If you don't know the size of buffer a priori, you can split it into
two calls: one which gets the size, and another which initializes the
variable in IDL and gets it filled up.

If you need really high speed, you might consider looking into IDL
shared memory implementation: see SHMMAP. You could then map a piece
of memory in both IDL and in C, and just dump the camera data directly
into it, at which point it would be directly visible in IDL. This
minimizes the number of times the data is copied, and would maximize
frame rate.

JD
