AID-016-12841-1
Communication Protocol
For
LabVIEW to DSCM
AT
University of Surrey

Job Title:	Communication Protocol for LabVIEW to DSCM	
Document Ref:	AID-016-128541-1.doc	
Client:		
Client Reference:		
Document History:		
Date	Details	Issue
December 2019	Issue	1

AERODYNAMIC TEST EQUIPMENT LTD
Crown Technical Centre, Burwash Road,
Heathfield, East Sussex, TN21 8QZ, UK
Tel: +44(0)1435 865245
Fax: +44(0)1435 865588
email: *info@ate-aerotech.co.uk*

Table of Contents

1.0 INTRODUCTION TO CLIENT COMMUNICATION FACILITIES

The communications consists of the following:-
SurreyBalDas.VI which is only a demonstration unit for proving the protocol. This needs to be set up in a directory on the LabVIEW machine. The LabVIEW version we are using is 8.6 but as I understand you should be able to go back with minor alterations but are not able to run newer versions then the version you have at the university. So if you are going to try and run our demo version then it might be good to set up the same directory for the DLL to save changing all the library paths in the Call Library Function Nodes. This is currently set to ***C:\AerotechSurreyBalDas\BALDAS_If.dll***. Also the BALDAS can be sited in the same directory or on a separate machine. In the LabVIEW block diagram the Ip address is set to 127.0.0.1. This can be changed as you wish to run it on another PC.

The files you need :-

If a separate PC for the balance
BALDAS.EXE
Balance. Mat
To connect to the DSCM the Ip port address should be set 192.168.7.36

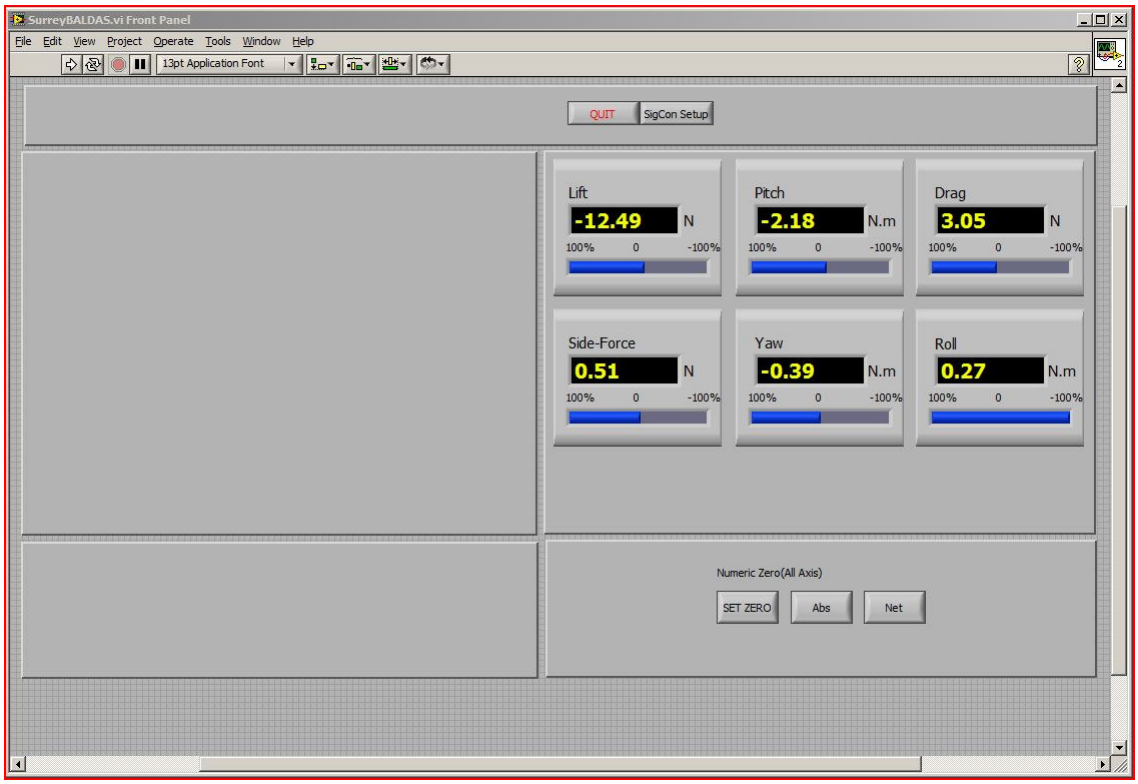
LabVIEW machine
BALDAS_If.dll.
SurreyBalDas.VI (For testing)

The files you need :-

If the same PC for the balance and LabVIEW
BALDAS.EXE
Balance. Mat
BALDAS_If.dll.
SurreyBalDas.VI (For testing)

To connect to the DSCM the Ip port address should be set 192.168.7.36

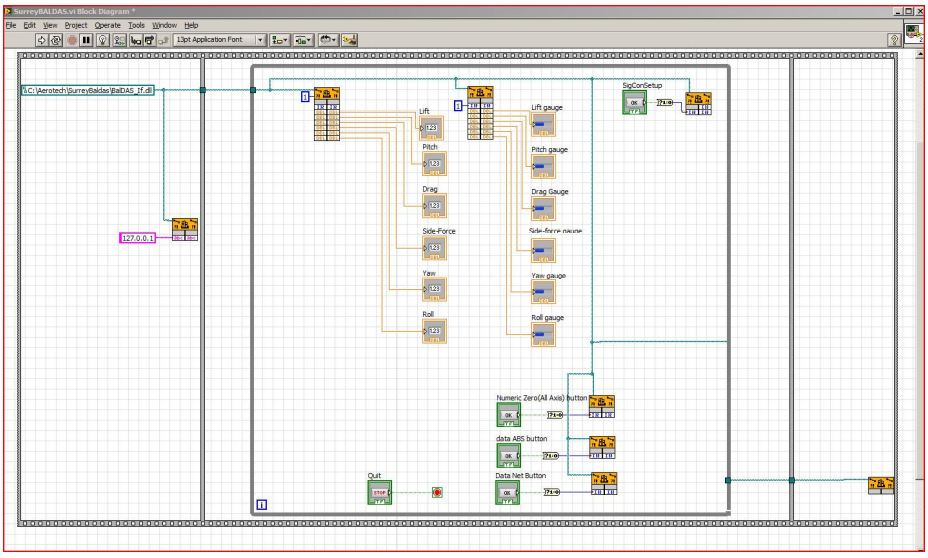
2.0 LabVIEW TEST SETUP

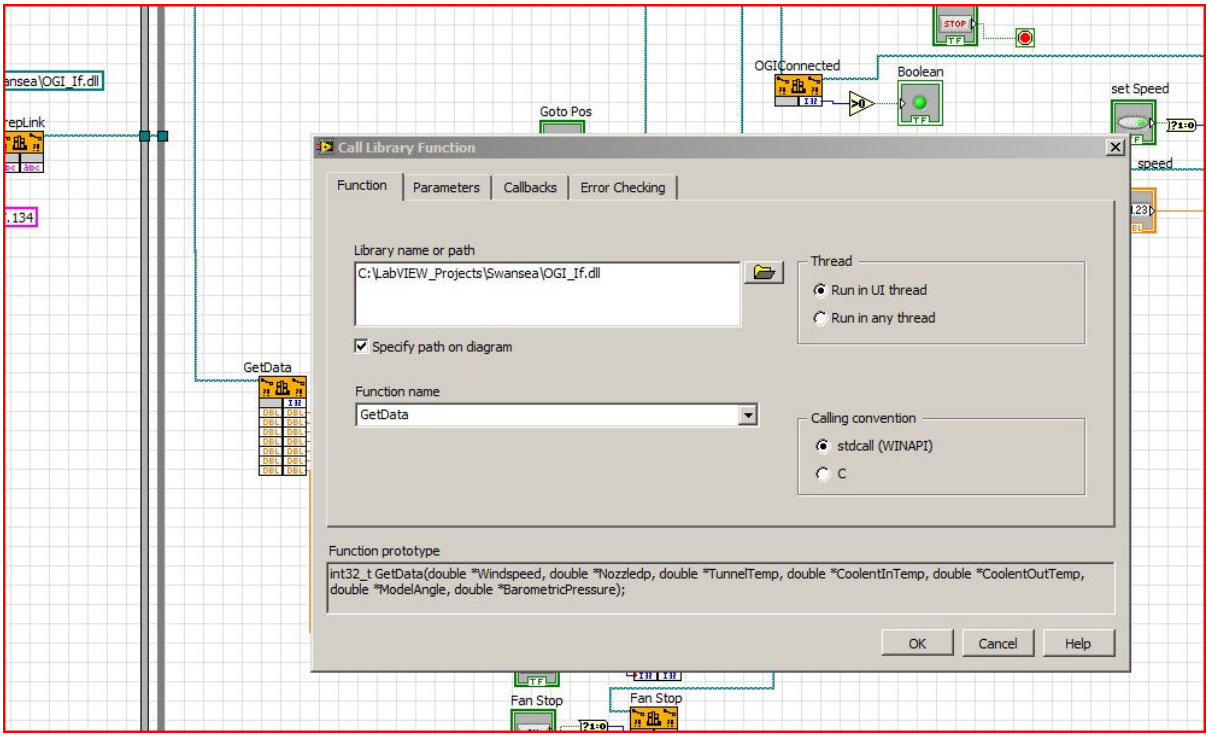


LabVIEW Front Screen

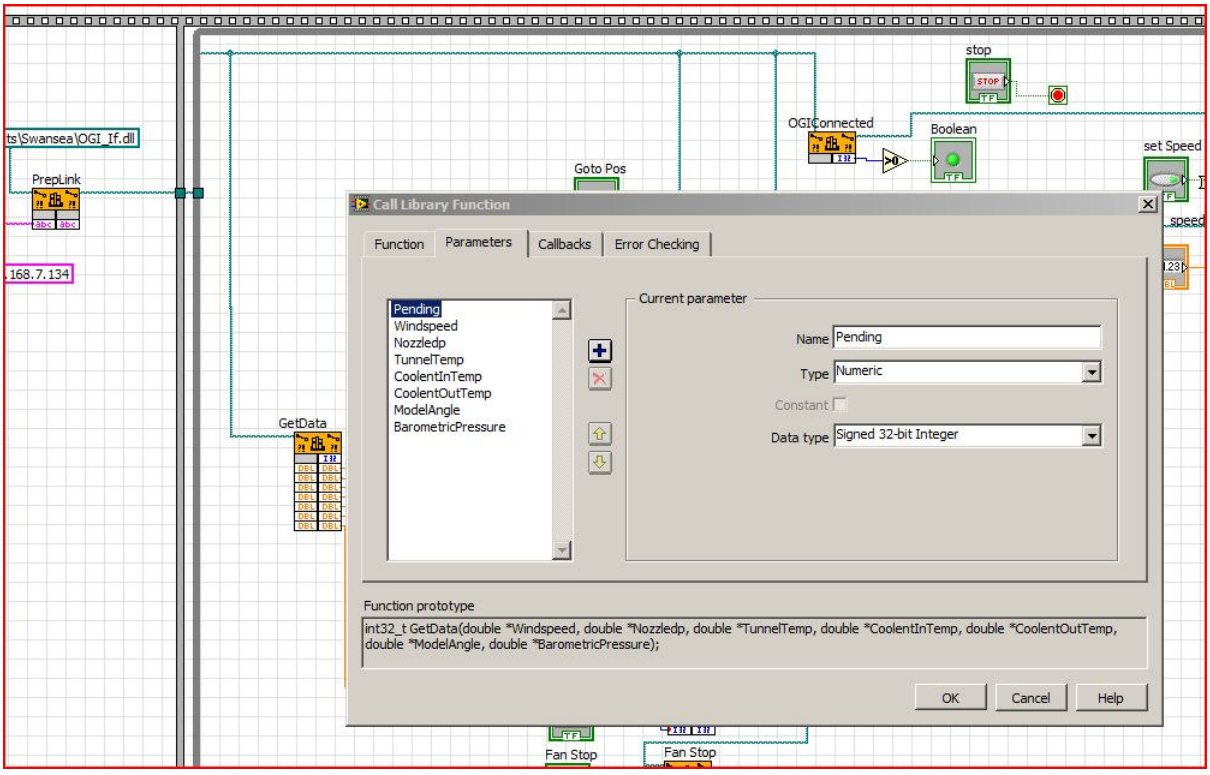
The BALDAS needs to be running first before the LabVIEW programs. On connection the LabVIEW will start steaming data.

You can see the Communication set up on the diagram. All the dLL functions use the Call Library Function Node These are set up to collect a Dll call or collect data.





Once you have created the Node and set the path you can then select the function name form the DLL.



Once you have completed this you can set up the Data Parameters and data type.

3.0 DLL FUNCTIONS

Function	Name	Type	Data Type	Pass
Prep Link	Return type	void		
	IPAddress	string	C string pointer	
GetAxisLoads	Enable	Numeric	Signed 32-bit int	Value
	Lift	Numeric	8-byte Double	Pointer to value
	Pitch	Numeric	8-byte Double	Pointer to value
	Drag	Numeric	8-byte Double	Pointer to value
	Side	Numeric	8-byte Double	Pointer to value
	Yaw	Numeric	8-byte Double	Pointer to value
	Roll	Numeric	8-byte Double	Pointer to value
GetAxisPercs	Enable	Numeric	Signed 32-bit int	Value
	Lift	Numeric	Unsigned 32-bit int	Pointer to value
	Pitch	Numeric	Unsigned 32-bit int	Pointer to value
	Drag	Numeric	Unsigned 32-bit int	Pointer to value
	Side	Numeric	Unsigned 32-bit int	Pointer to value
	Yaw	Numeric	Unsigned 32-bit int	Pointer to value
	Roll	Numeric	Unsigned 32-bit int	Pointer to value
USERSigConSetup	ack	Numeric	Signed 32-bit int	
	Enable	Numeric	Signed 32-bit int	Value
Set Zero	Return type	Void		
	Enable	Numeric	Signed 32-bit int	Value
Select absolute	Return type	Void		
	Enable	Numeric	Signed 32-bit int	Value
Select Net	Return type	Void		
	Enable	Numeric	Signed 32-bit int	Value

CloseLink	Return type	Void		
	Enable	Numeric	Signed 32-bit int	Value