We needed to write a header file to be used in Vee to work with the DLL file from Pico for the PT-104 temperature measurement device.

Dan and I took various files, and started with getting one example in a header file to work with the imported dll.

Here is the one line from the header file, and what I learned from each piece.

long \_\_stdcall UsbPt104OpenUnit(short \* handle, char \* serial);

long : this defines the data type output of the routine. Vee uses this to build an output on the box named “Ret Value”

\_\_stdcall : Without this, the routine will fail. It tells how to use the stack, and how to put on and off the data on the stack??

UsbPt104OpenUnit : The name of the routine inside the dll.

short \* handle : short is the data type of the output, and \* denotes that it will be an input of the same name as the output ‘handle’, and \* says the input handle is a pointer to a data location. In this case in Vee, there will be an input of name ‘handle’ and you need to deliver an int32 (that’s a short) into the input. It can be any int32 value, like a 0 (zero\_ and the routine will use that values address as a pointer to load the value in and return it on the output ‘handle’.

char \* serial : input type is char (text), and \* is the pointer. There is an input and an putput named ‘serial’. So in Vee after getting this Call box (get it from ‘compiled functions’) just deliver a text string. This particular string needs to be a valid serial number of the Pico unit for this routine to search for. If it can’t find it, an error 3 will be deliverd on the ‘Ret Value’ output. There is a Pico\_status.h file that can be used to translate the 3 into a human readable text string with an error message.

So we learned that:

 Char = text

 Short = int16

 Long = int32

 Float = real32 (I think)

 “\*” asterick means this is a pointer to a location, and in Vee that means an input will be generated, so you pass a variable of that type into that input pin. There will also be an output named the same name and have the same type.

 Void = When used as the beginning of a line, it says there will be no output (‘Ret Value’).

Files:

Usbpt104pg.en.pdf : Contains the syntax details for data types etc that are used in the routine.

USBPT104.dll : the dll file provided by Pico.

UsbPt104Api.h : Looks like a .h file used when compiling for C?

Pt104\_working.vh : our Vee .h file that works.

Pt104.vh : an older version of a .h file that worked with a Vee example (usb\_tc08.vee) from Microsoft XP and Vee 5 days.

picoStatus.h : the error messages file.

Molbloc – Resistance & Impedance.vee : Dan’s version of Vee he uses with our new .vh file to get Pico working.

Pico PT-104.vee Kirk’s simplified version of the program just using the dll to show how the .vh file works.