Custom USB-DIO-32 Firmware / DLL Task for David Berenda @ SRI International

New Function in AIOUSB.DLL 8.x:

```
__declspec(dllimport) unsigned long DIO_CSA_DoSync(unsigned long
DeviceIndex, double * BaseRateHz, double * DurAms, double * DurBms,
double * DurCms);
```

This function will take over operation of the USB-DIO-32, leaving no spare digital I/O for customer use.

The function will assert B and C as outputs, D as input, and perform the Camera / Light "Sync" function as configured with the parameters.

DeviceIndex: normal usage. Typically just pass the constant diOnly (0xFFFFFFFD)
BaseRateHz: A pointer to a double, Hz.
Typically you'll pass "60.0", but it needs to be in a variable so you can pass the pointer. See note 1.

DurAms, DurBms, DurCms: also pointers to doubles.
These are the number of milliseconds you want the three outputs to be "active". You would pass "1.0L" or "2.0L" to get a 1 millsecond "active" pulse, or 2 ms pulse, respectively. See note 1 and 2.

Note 1. The current revision of the custom firmware will allow you to achieve any frequency that is an integral divisor of a 3KHz internal clock tick. Therefore, if you ask for 60.0Hz, you will get 60.0 Hz, to the accuracy of the onboard crystal oscillator. On the other hand, if you ask for a frequency that does not result in an integral divisor, the Windows DLL will instead select the nearest frequency that does: The DLL modifies the parameter you pass to indicate the frequency the hardware will *actually* achieve. This is why these parameters are pointers.

Note 2. You can specify any number of milliseconds that will result in an integer number of 3KHz internal timer ticks up to 255 ticks. This results in pulse durations selectable from 0.3ms to 85ms. The operation of the outputs is NOT DEFINED if your durations are longer than the BaseRateHz can accommodate.

A New .SPT (firmware) file will be created that supports this new function:

It will generate a pulse output of duration "DurAms" at the frequency selected by "BaseRateHz". This is signal "A".

At the same moment "A" asserts itself the first time, a pulse of duration "DurBms" will be asserted. This is "B".

At the moment "A" asserts itself the second time, a pulse of duration "DurCms" will be asserted. This is "C".

Signal B and C continue generating pulses on every A activation, alternating: The B signal activates on even pulses of A; the C signal activates on odd pulses of A.

Pinout:

Port A - output	Port B - output	Port C - input	Port D - reserved
Signals A, B, C, and others	Inverted Port A signals	See pinout table	Reserved for factory use

Pin	Port	Function
47	A0	Signal "A" (Camera Sync) high going pulse
45	A1	Signal "B" (LED 1) high pulse
43	A2	Signal "C" (LED 2) high pulse
41	A3	Signal "A" (Camera Sync) high going pulse
39	A4	"Active" High if operation is running
37	A5	reserved
35	A6	reserved
33	A7	reserved
31	B0	Signal "A" (Camera Sync) low pulse
29	B1	Signal "B" (LED 1) low pulse
27	B2	Signal "C" (LED 2) low pulse
25	B3	Signal "A" (Camera Sync) low pulse
23	B4	"Active". Low if operation is running
21	B5	reserved
19	B6	reserved
17	B7	reserved

Pin	Port	Function
15	C0	If all pins are not high, or if all pins are not low, process will be PAUSED
13	C1	
11	C2	
9	C3	
7	C4	
5	C5	
3	C6	
1	C7	
46	D0	reserved
42	D1	reserved
38	D2	reserved
34	D3	reserved
30	D4	reserved
26	D5	reserved
22	D6	reserved
20	D7	reserved

All other pins retain the functions documented in the USB-DIO-32 Manual.