Uei-Bridge application

Design, decisions, detailed specification.

## ICD

Ethernet => Device, starts with {aah, 55h).

Device => Ethernet, starts with {55h, aah).

## Device table

List<List<PerDeviceObjects>> \_deviceObjectsTable;

Entry per device.

Entry index must suite the location of each device in the cube (as read by Device.GetIndex())

## Names

Device name is determines by as Uei api: GetDeviceName()

Block sensor device name is **“BlockSensor”**

## Digital card DIO403

48 bits. Might be configured as input or output

Read/Write is done with an array of Int16. The 8 higher bits are ignored.

Currently, lower 24 bits are considered output, and upper 24 bits are considered input. This is ‘very’ hard coded. Must at least set global constant.

## Block sensor

When block-sensor is active, the analog-out card (ao308) is dedicated to block-sensor. This means that downstream message that are aimed to the ao308 shall be rejected. (Might disable the udp-reader).

Block-sensor depends on A0308Manager, while DIO403/input must send its output to blocksensor.

## ToDo

1. Unit test on EthMessage
2. DIO403, replace bits. The input bits should be in lower order. (for the sake of block sensor)
3. Move device managers creation from Program.cs
4. Maybe it is better not to use Activator.CreateInstance(). You earn one and loose two. Something like DeviceManagerFactory might be more efficient.
5. BlockSensor device should reside on the first cube which contains analog and digital cards. At this stage only cube 0 is considered.
6. Use GetSetupEntryForDevice instead of getting entry by index.

## Convertes

Todo: Use

public interface IConvert2

{

object DownstreamConvert(byte[] messagePayload);

byte[] UpstreamConvert(object dt);

string DeviceName { get; }

}

Instead of IConvert,

Todo: AO308Convert and AI201Converter are both analog. Should be united.

Todo: There is unused code. 470 card does not need upstream converter.

*In second thought, not sure that set of converters is needed. Only analog in/out does a meaningful work. All others can be done in the device manager itself.*