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 (slot) 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.

**To do**: when block sensor is active, this card should be dedicated to block sensor. No i/o from/to ethernet.

## Analog/Out card

**To do**: when block sensor is active, this card should be dedicated to block sensor. No i/o from/to ethernet.

## Block sensor

Block-sensor depends on A0308Manager for outputting analog value.

BlockSensor uses analog and digital card. The first devices that he finds!

## 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 away from Program.cs
4. Maybe it is better **not** to use Activator.CreateInstance() For device managers, you earn one and loose two. Something like DeviceManagerFactory might be more efficient.
5. Config2: Use GetSetupEntryForDevice instead of getting setup entry by index.

## Converters

**To do 1**: Dedicated converter for each card looks overwhelming. There should be Analog-Convertor which unites 308 and 201 cards, and Digital-Convertor with unites 403 and 470 cards. All others (serials) might be handled be the device manager itself.

**To do 2**: Use

public interface **IConvert2<T>**

{

T DownstreamConvert(byte[] messagePayload);

byte[] UpstreamConvert(T dt);

string DeviceName { get; }

}

Instead of IConvert,

**To do 2**: It is too heavy to enforce a convert to know about the setup of the program. Just give the needed parameters! (min voltage, max voltage, etc…)