UnivIO Base Specification

First Published:	2021-11-24
Publish Site:	github.com/univio
Original Author:	Viktor Nagy
License Type	MIT (Open Source)
Actual Version, date:	1.0, 2021-11-24

Basics

The UnivIO is a simple, universal peer to peer communication system. Main purpose to create simple but flexible PC peripherials with non time critical data exchange.

The UnivIO device is accessed by 16-bit addresses:

- Addresses 0x0000 - 0x7FFF are objects, where every object has a predefined size and data type.
- Addresses 0x8000 - 0xFFFF behave like byte addressed linear memory. This allows to handle multiple IO in one request.

Common (Mandatory) Device Objects

Addr. (hex)	Type	R/W	Function
Device Generic Information (Mandatory)			
0000	u32	R	Comm Check, fix 0x66CCAA55
0001	u32	R	Maximal data length: 32 - 4096. Usually limited by the device internal storage.
0002	u32	R	Linear memory size (beginning at 0x8000) May be segmented, thus it should contain the maximal addressable size
0003	str[32]	R	Device FW ID Useful for pre-configuration check.
0004	u32	R	Device FW version
Device Generic Setup (Mandatory)			
0010	u8	RW	Configuration Status and Control Read and Write: 0 = CONFIG mode: outputs should be safe, configuration objects are writeable 1 = RUN mode: outputs activated, configuration objects are read only Write only: 2 = restart device (keeps the actual CONFIG/RUN status) Writing value 1 saves the configuration into the internal non-volatile storage. When the device starts loads the configuration and goes into RUN mode then. Writing value 0 also saves the run-mode so when the device restart stays in CONFIG mode. Special write values: Beware, that too frequent configuration save might wear out the Flash memory used for storage in the devices. Generic device in the manufactured state or without valid configuration must be CONFIG mode.
0011	str[32]	RW	Device ID Used as the USB Device ID too. Recommended to assign a unique ID for different configurations and / or different pieces.
0012	u16	RW	USB Vendor ID
0013	u16	RW	USB Product ID
0014	str[32]	RW	Serial Number

Serial protocol

The UNIVIO communication protocol is a request / response binary protocol. The smallest unit is 8-bit = byte. The primary target is UART and virtual UART using USB CDC (VCP).
Multi-byte integers are always transferred in LSB (little-endian) format.

The request and responses are protected with an 8-bit CRC. If the received CRC in the request does not correspond to the calculated CRC the device does not executes and does not answer to the request.

Request Format

Bytes	Segment ID	Description
1	SYNC	sync byte, always = 0x55
1	RW, MLEN, LEN	Read/Write and length information bit0: RW: 0 = read, 1=write bit1: reserved bit2..3: MLEN: metada length, 0=0, 1=2 byte, 2=4, 3=8 bit4..7: LEN: read or write length, if LEN = 15 then EXTLEN follows
0 | 2	EXTLEN	16-bit extended length, present only if LEN = 15
2	ADDRESS	16-bit address
0 - 8	METADATA	Optional metadata,present only if MLEN > 0 Can be useful for extra request parameters
0 - 4096	WDATA	Write data. Present only if RW=1, length determined by LEN or EXTLEN (when LEN = 15).
1	CRC	The CRC calculated for all the previous bytes (including the SYNC byte). The CRC8 with generator polinom is 0x07.

(the response format is described on the next page)

Response Format

Bytes	Segment ID	Description
1	SYNC	sync byte, always = 0x55
1	RW, ERR, MLEN, LEN	Read/Write, Error and length information bit0: RW: 0 = read response, 1 = write response bit1: ERR, 0 = no error code, 1 = 16 bit error code at the RDATA bit2..3: MLEN: metada length, 0=0, 1=2 byte, 2=4, 3=8 bit4..7: LEN: read or write length, if LEN = 15 then EXTLEN follows
0 | 2	EXTLEN	16-bit extended length, present only if LEN = 15
2	ADDRESS	16-bit address (same as in the request)
0 - 8	METADATA	Optional metadata, present only if MLEN > 0.
0 | 2	ECODE	16-bit error code, present only if ERR = 1
0 - 4096	RDATA	Read data. Present only if RW=0 (read), and ERR = 0. Length determined by LEN or EXTLEN (when LEN = 15).
1	CRC	The CRC calculated for all the previous bytes (including the SYNC byte). The CRC8 with generator polinom is 0x07.