OBELISK
SYSTEMS

COMMUNITY CUBESAT STANDARDS PREVIEW

A DRAFT DOCUMENT OF THE PROPOSED CUBESAT MECHANICAL, ELECTRICAL AND COMMUNICATIONS STANDARD BY OBELISK SYSTEMS

NOT FOR PUBLIC RELEASE

HTTP://OBELISKSYSTEMS.COM

INFO@OBELISKSYSTEMS.COM

Contents

INTRODUCTION:

For many years, the Cubesat community has remained largely fragmented in the design and coordination of the finer details of the Cubesat standards. The only official standard that exists to date is the Cubesat Design Specification (CDS), which in 2000 paved the way for Cubesats to revolutionise access to space. We're now well into 2016 with hundreds of Cubesat launches that have carried payloads enabling new communications systems, propulsion techniques and hard science. An issue that has plagued almost any Cubesat is that they are developed in their own way, based on accessible heritage, custom-tailored from vendors or a combination of both. This means that most Cubesats after over a decade are yet to share common interconnectivity, and the ones that do are because of some de-facto standards that emerged out of necessity.

Obelisk Systems has built on the work of Arthur Scholz's (cubesat.de) attempt to consolidate the bus and combined it with the latest in Cubesat bus trends to create finalised electric and mechanical designs. The purpose of the documentation is to encourage the community to adopt this unified, absolute standard to allow for cross-compatibility of payloads. We don't want the community to be fragmented any more.

The Cubesat Printed Circuit Boards (PCBs) are what make Cubesats possible, with each PCB encapsulating a core function of the satellite (e.g. communications subsystems, on-board computer, power system, etc.), or parts of it. As each module is stacked on top of the other, the assembly of the Cubesat is fast, reliable and allows for many universities, institutions and suppliers *to build functional hardware together.*

CONFORMANCE GUIDELINES:

The proposed specification covers three aspects of board design: mechanical, electrical and software interfacing. Conformance is only met for any of the three classes when ALSO meeting conformance to the Cal-Poly Cubesat Design Specification. These "conformance levels" should be implemented to allow for maximum interoperability of hardware between different companies and institutions.

It is intended to provide three classes of conformance to this specification as follows:

Class A: Mechanical conformance.

Refers to devices that are conforming to the mechanical interface specifications outlined in section 1.1. For now, simply adhering to the external rails for the chassis (Cal-Poly Cubesat Design Spec) will be enough for preliminary conformance.

Class B: Mechanical & electrical conformance.

Refers to devices that are conforming to Class A and to the electrical interface specifications as outlined in Section 1.2.

Class C: Mechanical, Electrical and Software interfacing.

Refers to the devices that are conforming to Class A and B and to the software interface specifications as outlined in Section 1.3.

1. SPECIFICATIONS:

1.1 Mechanical:

The Mechanical specification has been derived from the well-known PC/104 form factor, which was originally modified by Pumpkin for the Cubesat Kit. The larger majority of Cubesats utilise this mechanical standard due to its robustness, versatility and simplicity. **Mechanical standardisation sees the largest reduction in cost for launches as the launch pods for Cubesats (P-POD) are fixed in design for 10cm x 10cm structures.**

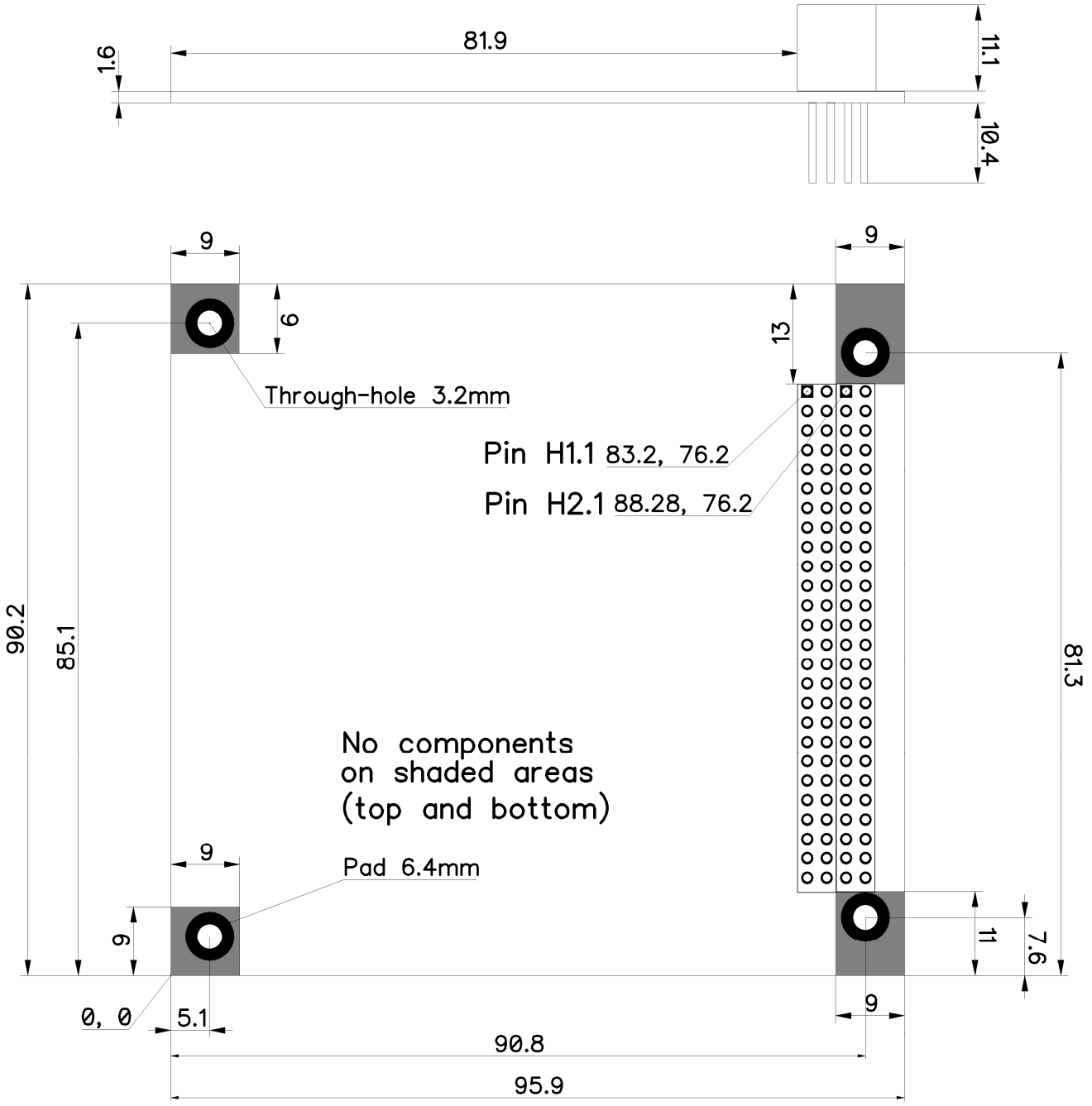
The specification drawing is provided in Illustration 1. **All dimensions are in millimetres**, and conformance is only complete a PCB fulfils all requirements listed in Table 1.

These specifications are to accompany the existing Cubesat Design Specification and deviation from either specification is not permitted. Any deviation is to be documented and stated in conformance paperwork.

Table 1: Mechanical Requirements

ID	Requirement	Justification
REQ-M-01	The size and location of the mounting holes must be adhered to.	Conforming with PC/104
REQ-M-02	No components shall be placed on the shaded areas around the mounting holes (neither top nor bottom).	Mounting shall not be obstructed.
REQ-M-03	The location of the connector must be adhered (using the pin-hole locations as reference).	Without this, interconnectivity deviates and becomes impossible.
REQ-M-04	The board shape must not extrude over the given dimensions. But it is allowed to cut away portions of the board, provided that the integrity of the board is not affected.	Board must fit within Cubesat dimensions for deployment.
REQ-M-05	The components on the top side shall not extrude more than 11mm. Otherwise it must be explicitly declared, when stating the conformance.	Components must not exceed dimensions of Cubesat standard.
REQ-M-06	The components on the bottom side shall not extrude more than 3mm, otherwise it must be explicitly declared when stating conformance.	To avoid obstruction when stacking boards.
REQ-M-07	The board thickness shall be 1.6mm or less.	Standard PCB Thickness.
REQ-M-08	The connector shall be a SAMTEC ESQ-126-39-G-D stackthrough or ESQ-126-37-G-D non-stackthrough or compatible.	Connector used by the Cubesat Design Specification and CubesatKit.

Illustration 1: Cubesat mechanical dimensions.



All dimensions in mm unless otherwise specified.

1.2 Electrical

This section provides specification to each pin on the Cubesat PCB primary connector. Each pin has a corresponding designator, description and documentation of function. In order to meet electrical conformance, each PCB (board) must comply with the requirements of Table 2.

Table 2 Electrical Requirements:

ID	Requirement	Justification
REQ-E-01	The designation and usage of the pins as defined in Table 3 must be adhered to. This means that one shall not put any other signal on the pins other than what is specified.	Electrical incompatibilities will cause malfunction of Cubesat systems.
REQ-E-02	A board may be the source of or sink to any number of specified pins, as long as no conflicts arise in the overall system.	When implemented correctly, there will not be interference.

Table 3: Cubesat Primary Header Electrical Specification:

CUBESAT LEFT HEADER

CANL	**Pin 1**	**Pin 2**	D0
CANH	**Pin 3**	**Pin 4**	D1
DCLK	**Pin 5**	**Pin 6**	D2
-DBUSY	**Pin 7**	**Pin 8**	D3
-DW	**Pin 9**	**Pin 10**	D4
-DR	**Pin 11**	**Pin 12**	D5
USR	**Pin 13**	**Pin 14**	D6
USR	**Pin 15**	**Pin 16**	D7
TX1	**Pin 17**	**Pin 18**	RX1
TX2	**Pin 19**	**Pin 20**	RX2
SCK0	**Pin 21**	**Pin 22**	SDIO
SDO0	**Pin 23**	**Pin 24**	-CS
-FAULT	**Pin 25**	**Pin 26**	VREF0
-READY	**Pin 27**	**Pin 28**	VREF1
-RESET	**Pin 29**	**Pin 30**	VREF2
USR	**Pin 31**	**Pin 32**	CHRG
USR	**Pin 33**	**Pin 34**	USR
USR	**Pin 35**	**Pin 36**	USR
USR	**Pin 37**	**Pin 38**	USR
TX0	**Pin 39**	**Pin 40**	RX0
SDA_SYS	**Pin 41**	**Pin 42**	VBU
SCL_SYS	**Pin 43**	**Pin 44**	SDA1
VCCS	**Pin 45**	**Pin 46**	SCL1
VCC0	**Pin 47**	**Pin 48**	VCC1
VCC2	**Pin 49**	**Pin 50**	VCC3
VCC4	**Pin 51**	**Pin 52**	VCC5

CUBESAT RIGHT HEADER

EPS0	**Pin 1**	**Pin 2**	EPS1
EPS2	**Pin 3**	**Pin 4**	EPS3
EPS4	**Pin 5**	**Pin 6**	EPS5
EPS6	**Pin 7**	**Pin 8**	EPS7
COM0	**Pin 9**	**Pin 10**	COM1
COM2	**Pin 11**	**Pin 12**	COM3
COM4	**Pin 13**	**Pin 14**	COM5
COM6	**Pin 15**	**Pin 16**	COM7
ADCS0	**Pin 17**	**Pin 18**	ADCS1
ADCS2	**Pin 19**	**Pin 20**	ADCS3
ADCS4	**Pin 21**	**Pin 22**	ADCS5
ADCS6	**Pin 23**	**Pin 24**	ADCS7
+5V	**Pin 25**	**Pin 26**	+5V
+3.3V	**Pin 27**	**Pin 28**	+3.3V
GND	**Pin 29**	**Pin 30**	GND
AGND	**Pin 31**	**Pin 32**	GND
S0	**Pin 33**	**Pin 34**	S0
S1	**Pin 35**	**Pin 36**	S1
S2	**Pin 37**	**Pin 38**	S2
S3	**Pin 39**	**Pin 40**	S3
S4	**Pin 41**	**Pin 42**	S4
S5	**Pin 43**	**Pin 44**	S5
VBATT	**Pin 45**	**Pin 46**	VBATT
PAY0	**Pin 47**	**Pin 48**	PAY1
PAY2	**Pin 49**	**Pin 50**	PAY3
PAY4	**Pin 51**	**Pin 52**	PAY5

Table 4: Detailed Description of the Cubesat electrical layout.

ID	Pin(s)	Description + Characteristics
Control and Monitoring Buses		
SDA_SYS	H1.41	Data line of primary I2C bus.
SCL_SYS	H1.43	Clock line of primary I2C bus.
SDA1	H1.44	Data line of secondary I2C bus.
SCL1	H1.46	Clock line of secondary I2C bus.
CANL	H1.1	CAN- signal of CAN bus.
CANH	H1.3	CAN+ signal of CAN bus.
Data Buses (Parallel)		
DCLK	H1.5	Data bus clock signal. Data byte is valid at rising edge.
-DBUSY	H1.7	Data bus busy flag. Active low.
-DW	H1.9	Data bus write enable. Active low.
-DR	H1.11	Data bus read enable. Active low.
D[0-7]	H1.[2, 4, 6, 8, 10, 12, 14, 16]	Data bits. D0 is LSB. The data bus is can be used to transport large data amounts from payload to CDHS, and vice-versa.
Data Buses (Serial)		
TX0	H1.39	TXD of UART0 for data to be transmitted via communication system.
RX0	H1.40	RXD of UART0 for data received via communication system.
TX1	H1.17	TXD of UART1
RX1	H1.18	RXD of UART1
TX2	H1.19	TXD of UART1
RX2	H1.20	RXD of UART1
SCK0	H1.21	SPI Clock
SDI0	H1.22	SPI Master Data In
SDO0	H1.23	SPI Master Data Out
-CS	H1.24	Slave Select Signal. Active Low
Power		
VBU	H1.42	Backup power supply (+3.3V, max. 0.5 Watt). Shall be active even when spacecraft is powered off.
CHRG	H1.32	Input for charging the spacecraft batteries. Input is +5 V +/- 10%. Typically implemented with a USB interface.
VCCS	H1.45	Switchable 'special' power supply line. Proposed 12V.
VCC0	H1.47	Switchable power supply line. 3.3V
VCC1	H1.48	Switchable power supply line. 5V
VCC2	H1.49	Switchable power supply line. 3.3V
VCC3	H1.50	Switchable power supply line. 5V

VCC4	H1.51	Switchable power supply line. 3.3V
VCC5	H1.52	Switchable power supply line. 5V
+5V	H2.[25,26]	5 Volt permanent power line. (For drivers and heavy loads.)
+3.3V	H2.[27, 28]	3.3 Volt permanent power line. (For microcontroller and periphery.)
GND	H2.[29, 30, 32]	Ground.
AGND	H2.31	Analogue (Analog) Ground.
VBATT	H2.[45,46]	Battery power permanent power line.
Reference Voltages		
VREF0	H1.26	Voltage reference 0. Usually 1.8 V.
VREF1	H1.28	Voltage reference 1. Usually 2.5 V.
VREF2	H1.30	Voltage reference 2. Usually 3.3 V.
Interrupts		
-FAULT	H1.25	Indicates occurrence of an (hardware) error occurred. Pulsed, active low.
-RESET	H1.29	Triggers system reset. Systems listening on this line shall reset themselves when triggered.
-READY	H1.27	Indicates that an activity has completed. Pulsed, active low.
Electrical Payload System Pins		
EPS0-EPS7	H2.[1-8]	Electrical Power System general input/output pins. Use for voltage monitoring, watchdog signals, etc.
Communications System Pins		
COM0-COM7	H2.[9-16]	Telecommunications and signalling general input/output pins. Use for additional monitoring/configuration of telecoms systems.
Attitude Determination and Control Systems		
ADCS0-ADCS7	H2.[17-24]	General input/output pins for satellite stabilisation, control and advanced systems.
Payload-Specific Pins		
PAY0-PAY5	H2[47-52]	General input/output pins for control of payloads connected to Cubesat.
RBF and Deployment Switches		
S0	H2.[33,34]	Connect to RBF Switch normally closed (NC) terminal.
S1	H2.[35,36]	Connect to Deployment Switch normally closed (NC) terminal.
S2	H2.[37,38]	Connect to RBF Switch normally open (NO) terminal.
S3	H2.[39,40]	Connect to Deployment Switch normally open (NO) terminal.
S4	H2.[41,42]	Connect to RBF Switch common (COM) terminal.
S5	H2.[43,44]	Connect to Deployment Switch common (COM) terminal.

2 GROUPING JUSTIFICATION:

2.1 Control and Monitoring Buses:

These are available to all other systems. Generally, a centralised architecture is used with the CDHS being the master and all other subsystems as slaves. We want to use this to allow for the CDHS to packetize and delegate absolutely everything on the Cubesat.

There are two (2) primary I2C data buses that are common to the whole system, and allow a lot of traffic to pass through this. These lines should be preserved for signalling/control only, not data.

2.2 Data Buses:

This serves as a high-speed data bus for exchange between systems. Again, a centralised architecture is normally used, with CDHS controlling all traffic.

This is the largest part of the standard that is up for discussion, as the parallel data bus might not be desirable for some applications.

This is offset by providing three (3) Serial connections, one (1) I2C primary data connection and an SPI bus.

2.3 Power:

These pins are for the output from the EPS. Only the CHRG pin is for external input for charging batteries or the EPS (Typically a USB +5V rail). Most power pins are already specified by the PC/104 standard, so there's little to no deviation from original spec here.

2.4 Reference Voltages:

Stable voltage references to be normally supplied from EPS. These aren't mandated as necessary, however provision allows for more accurate calibration of on-board monitoring for telemetry systems to function adequately.

2.5 Interrupts:

Reserved for asynchronous notifications. There's a significant amount of coded messages that can be utilised with asynchronous comms for flags, resets and more. Having dedicated interrupt lines allows for simplification and standardisation of core errors.

2.6 Subsystem-specific Buses:

These pins are used to route subsystem-specific signals to the CDHS or to external, such as JTAG for in-system programming.

For best use of this standard, these pins should be logic-only pins that connect directly to the on-board microcontroller, or passed through a multiplexer to preserve MCU IO.

These logic pins are responsible for triggering different states or providing feedback of device status other than the main data bus.

2.7 Switches:

For implementing the RBF and Deployment Switch logic + Control.

2.8 User Configurable Pins

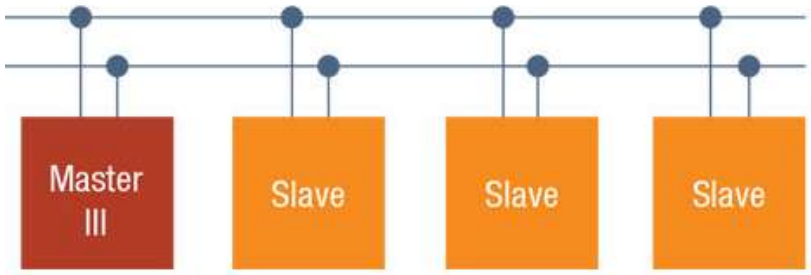
These pins are labelled USR, and are typically available for either power or logic: They should NOT be used for high-frequency switching.

3. SOFTWARE INTERFACING

The software interface should be kept as simple as possible. It is expected that the I2C bus will be the primary system bus for inter-system interfacing, and this is common practice in the industry. A centralised architecture has been used, with CDHS acting as the master and all other systems as slaves. The master can read/write the values of registers of the slave devices for control and monitoring.

Each slave is therefore considered as a state-machine, which provides a number of registers that can be written to and/or written from. In this way, the master can fully configure and control the slave as well as monitoring its status and reading its telemetry. The architecture depicted in Illustration 2 demonstrates this function. External components usually interface directly with the pertinent subsystem board via dedicated interfaces (UART, JTAG, etc.), or optionally may be attached directly to the system bus as well.

Illustration 2:



To conform to the standard, each board must comply with the requirements listed in table 5.

Table 5: Software interfacing requirements

ID	Requirement	Justification
REQ-S-01	The I2C bus shall be used as the main system bus for control and monitoring of the (sub)systems.	
REQ-S-02	Individual components may be connected to the main system bus or be directly connected to the relevant (sub)system board.	
REQ-S-03	There shall be only one master on the main system bus, namely the CHDS. An exception to this rule is for external EGSE, which may also be configured as a master.	
REQ-S-04	The slave boards shall provide registers that can be accessed by read/write operations on the main system bus for controlling and monitoring of itself.	

The two basic functionalities of the software interface are the reading from and writing to device registers. For this, each device (which can be anything from a subsystem to component) defines and maintains a number of internal registers (i.e. a continuous byte array in memory) to which another device can write to or read from.

Register Configuration:

The device registers are categorized into:

- **Main:** General registers and command input;
- **Settings:** Registers related to device settings and configuration;
- **Flags:** Registers that hold bit flags;
- **Values:** Registers that hold reporting data (such as measurement data)

Writing and reading of the device registers allows for control and monitoring of the device. The layout of the register is specific for each device and defined by the developer.

The registers shall be 8 bit in size. Larger values are stored by combining individual registers. For example, a 16 bit integer is stored in two registers, one for the high and one for the low byte.

The concept of operation is that a (master) device does a write operation to a (slave) device in order to issue a command to it or to change the configuration of the (slave) device. Further, the master issues a read operation to the (slave) device in order to read the device's settings, flags, or reporting data.

The following paragraphs describe how to carry out the read and write functions between two devices. The device initiating the actions is the master and the other device is the slave, for the duration of the operations.

Write Operation:

The master initiates a write operation by issuing a **START** condition followed by the slave address and a **WRITE** bit (0). When the master receives an **ACKNOWLEDGE** from the device it can transmit the 8-bit register address. The slave will respond with a second **ACKNOWLEDGE**. The master then transmits the 8-bit value that shall be loaded into this register. The slave will respond with a third **ACKNOWLEDGE**. The write operation is then completed when the master issues a **STOP** condition or a second **START** condition.

S	Device Address	W	A	Register Address	A	Value	A	P

Read Operation:

The master initializes a read operation with a **START** condition followed by the slave address and a **WRITE** bit (0). When the master receives an **ACKNOWLEDGE** from the slave it can transmit the 8-bit register address. The device will respond with a second **ACKNOWLEDGE**.

The master must then issue a new **START** condition followed by the slave address and a READ bit (1). The slave will respond with an **ACKNOWLEDGE** followed by the value of the register. The master must issue a **NOT ACKNOWLEDGE** followed by a **STOP** condition to terminate the operation.

S	Device Address	W	A	Register Address	A	S	Device Address	R	A	Value	NA	P

Bit Operations:

The read and write operations operate on the basis of single bytes. In order to set or clear individual bits from the internal register without changing the other bits of that register, the register must be first read out, then be subjected to **AND/OR** operation for clearing/setting a single or multiple bits, and finally the result must be written back to the register.

<REGISTER NAME>							
Bit 7 MSB	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0 LSB

PLEASE CONTACT ANDREAS@OBELISKSYSTEMS.COM TO PROVIDE FEEDBACK ON THIS STANDARD. WE WISH TO MAKE IT AVAILABLE TO AS MANY AS POSSIBLE OVER THE COMING WEEKS.

HTTP://OBELISKSYSTEMS.COM

HTTP://FB.COM/OBELISKSYSTEMS

HTTP://TWITTER.COM/OBELISKSYS