Simply Automated, Inc.

UPB
Input/ Output Module
(UMI-32)
Firmware Specification

V 4.6

6/10/05

Revision History

Spec. Rev.	Date	Firmware Rev.	Description
3.0	11/08/04	2.0.6	UMI-32 spec – 1st version in the standard format. Bug changes.
4.0	12/10/04	2.0.7	Reflect the memory map changes due to the TCL redefinition. Add information on factory default.
4.1	12/13/04	2.0.7	Corrected the problems from KH. Made the channel # 0 based. Corrected the text in section 10.1
4.2	12/21/04	2.0.8	Removed archaic language, new memory map
4.3	3/7/05	2.0.8	Correct errors, never released
4.4	3/16/05	2.0.8	Updated the specification to agree with the new UPB System Description.
4.5	5/2/05	2.0.8	Changed wording on output loads
4.6	6/10/05	3.0.0	Major changes to add the buffered inputs, ZAP feature, test mode, and utilized the remaining Core space.

Table of Contents

1. Scope 1

2. Introduction 1

2.1. UPB Hardware 1

2.1.1. Channel Outputs 1

2.1.2. Channel Inputs 1

2.1.3. UPB Receiver 2

2.1.4. UPB Transmitter 2

2.1.5. Single Pushbutton 2

2.1.6. Unit LED Indicator 2

2.1.7. Channel LED Operation 2

2.2. UPB Software 3

2.2.1. Software Architecture 3

2.2.2. Core Software 3

2.2.3. Application (APP) Software 3

2.2.4. UPB System Model 3

3. Modes of Operation 4

3.1. Normal Mode 4

3.2. Test Mode 4

3.3. ZAP Mode 4

3.4. Setup Mode 4

3.4.1. Entering Setup Mode 4

3.4.2. Exiting Setup Mode 5

3.5. Factory Default Mode 5

3.5.1. Entering Factory Default Mode 5

3.5.2. Exiting Factory Default Mode 5

4. UPB Setup Registers 6

4.1. The UPBID 6

4.2. The Configuration Registers 8

4.3. The Scratch-Pad Registers 9

5. Unit Status LED Operation 9

5.1. The LED Options Register 9

5.2. LED Modes 10

5.2.1. Diagnostic LED Mode 10

5.2.2. Normal LED mode 10

5.2.3. Special Mode LED Indications 10

5.2.4. Firmware Version LED Indication 11

6. UPB Message Receiving 11

6.1. Receive Components 11

6.2. Receiving UPB Link Packets 12

6.2.1. Activating/Deactivating Receive Components 12

6.3. Receiving the Core Command Message Set 12
6.4. Receiving the Device Control Command Set 13
6.4.1. “Activate” Command 13
6.4.2. “Deactivate” Command 13
6.4.3. “Goto” Command 14
6.4.4. Direct “Goto” Command 14
6.4.5. Link “Goto” Command 14
6.4.6. “Report Status” Command 14
7. UPB Message Transmitting 16
7.1. Transmit Components 16
7.2. UPB Transmit Control Register 16
7.2.1. Link Packet Enable 17
7.2.2. Acknowledge Message Request 17
7.2.3. ID Pulse Request 18
7.2.4. ACK Pulse Request 18
7.2.5. Transmission Count 18
7.2.6. Transmission Component Link (TCL) Tables 18
8. Power-Up Operation 19
9. Appendix - Memory Map 19
9.1. Memory Map - UPBID 19
9.2. Receive Component Link Table Channel 1 21
9.3. Receive Component Link Tables, Channel 2 23
9.4. Transmit Component Link Tables 24
9.5. Reserved 25
9.6. Unit Options 25
9.7. Scratch Pad 25
10. References 26
11. Glossary 26

1. Scope

This document is meant to provide enough information to allow developers to create software to interface with the UMI-32.

2. Introduction

The Input/Output device (UMI-32) is capable of controlling two low voltage loads. The UMI-32 can be employed to control its two outputs by Universal Powerline Bus™ (UPB) Commands. The UMI-32 contains a bi-color LED to indicate status, modes, and events; two LEDs to indicate the status of the outputs and three LEDs to reflect the state of the inputs. The UMI-32 contains a single pushbutton for user input. The input events are buffered to avoid loss up to ten deep.

2.1. UPB Hardware

2.1.1. Channel Outputs

The UMI-32 has two outputs with a maximum rating of 60 V (AC/DC) @ 400mA. The outputs are non-dimming meaning they are either OFF or ON.

2.1.1.1. Zap

ZAP is enabled by a flag in the DeviceOptions control register (see memory map). In this mode if the user sends a command to close an output relay. The relay will be closed for 1 second and then opened again automatically.

2.1.2. Channel Inputs

The UMI-32 has three (3) channel inputs. These inputs are qualified in software. Input 1 is qualified for 150ms and the other two for 64 ms. The messages, driven by these events, are stored in a FIFO 10 buffers deep. If the buffer is overwhelmed it will proceed to overwrite the oldest events in the buffer.

Input	Functions	Pins
1	4-5V AC/DC sense or phone line (ring detect)	
2	4-5V AC/DC sense or contact closure	
3	4-5V AC/DC sense, 200 mA sense (doorbell), or contact closure, or line level audio presence.	

Table 1 – Description of Hardware Inputs

2.1.3. UPB Receiver

The UMI-32 has UPB Receiver Logic capable of receiving UPB Communication Packets from the powerline. The Core software will accept, decode and validate messages from the powerline bus.

2.1.4. UPB Transmitter

The UMI-32 has UPB Transmitter Logic capable of transmitting UPB Communication Packets onto the powerline. The Core accepts messages from the application software and places them onto the powerline bus at the appropriate times. The Core does not have any buffering. It can take up to a second before it completes the transmission due to the number of repeats.

2.1.5. Single Pushbutton

The UMI-32 has a momentary pushbutton for user input to enter Setup mode, and reset factory defaults.

2.1.6. Unit LED Indicator

The UMI-32 has a bi-color LED indicator used to indicate its current status and mode of operation.

2.1.7. Channel LED Operation

These LEDs are hardwired to input signal processing circuits and indicate the status of each output/input channel.

2.1.7.1. Output LEDs

There is a green LED wired to each output channel that reflects the state of the closure, open being off and closed being green.

2.1.7.2. Input LEDs

There is a green LED wired to each input channel that reflects the state of the line; the LED is turned on when it detects a closure.

2.2. UPB Software

2.2.1. Software Architecture

The UMI-32 is split into Core and Application software. The Core software is provided by Simply Automated and implements many of the low level hardware interfacing functions (especially with the powerline). The Application software gives the device its personality. It implements the functions above the basic communications on the powerline.

2.2.2. Core Software

The Core provides a Framework for the Application code to run under; receives messages, validates messages, and executes certain commands or passes the others to the Application code; an Application Interface is provided for use by the Application code. These features are described in the UPB System Description Document (http://simply-automated.com/tech_specs/). In order to avoid writing this term out each time, it will be referred to as UPBSD.

2.2.3. Application (APP) Software

The APP software is an event loop that continually checks for changes on the three inputs and commands from the powerline. Beside this it can be interrupt by the unit pushbutton.

2.2.4. UPB System Model

The system model is a description of the unit's operational variables that are contained in EEPROM. These variables can be changed while in Setup mode (see below) by a user. The model has 3 parts UPBID registers, Configuration registers, and Scratch Pad registers.

- UPBID registers - contain the important unit variables i.e. password, address, etc.
- Configuration registers – contain the Receive Component Link tables, Transmit Component Link tables, Option variables, and control bits.
- Scratch Pad registers – contain a set of counters that the Core will use to keep track of events.

Note: the size of the UPBID and Configuration sections added together is defined at compile time and is available from the unit via the "Get Signature" command

(see UPBSD). This allows management devices to upload only the parts of memory that are being used.

3. Modes of Operation

The UMI-32 is capable of being put into different modes of operation by a pushbutton sequence: Normal, Test, Setup and Factory Default Mode. Other modes such as ZAP are configured by writing to EEPROM.

3.1. Normal Mode

The UMI-32 usually operates in the Normal Mode. The Normal Mode is the UMI-32's default mode of operation. While in the Normal Mode the UMI-32 performs all of its normal operations except that Setup Register Write Protection is enabled.

3.2. Test Mode

Test mode is entered by holding the pushbutton. The LED turns red when the mode has changed. Then press the pushbutton once and the outputs will change to the inverse of their present state. At the same time the LED will turn to orange. To exit Test mode press and hold the pushbutton until the outputs change to their original state (observe the status LEDs).

3.3. ZAP Mode

When an output is closed in ZAP mode it will timeout in one second and then open the relay. The configuration byte for this mode is stored in EEPROM and once configured will persist across a power cycle.

3.4. Setup Mode

In Setup mode the UMI-32 can be accessed remotely by a management device or enter Factory Default mode (by using the pushbutton). The Setup Mode is a special mode of operation that every UPB device that conforms to the UPB System Model must have. The UPBSD document describes the Setup Mode in more detail. The UMI-32 can enter Setup Mode by two different methods. One is by receiving a valid "Start Setup Mode" command message over the powerline as described in the UPBSD. The second method is instigated by manually pressing the pushbutton on the case.

3.4.1. Entering Setup Mode

The UMI-32 enters Setup Mode when the pushbutton is pressed exactly 5 times. This change of state is indicated by blinking the LED as defined in section 5.5.

- Setup mode – 5 times on pushbutton

3.4.2. Exiting Setup Mode

The UMI-32 exits Setup Mode and enters the Normal Mode, when the pushbutton is pressed 1 time. When the UMI-32 exits the Setup Mode it will indicate so by changing the LED mode of operation as defined in section 5.2.2.

- Exit Setup Mode – 1 time on pushbutton

3.5. Factory Default Mode

The Factory Default Mode is a special mode of operation that, when entered, sets the Setup Registers to their Factory Default values as defined in the appendix. While in the Factory Default Mode, the LED indicator shall indicate this mode of operation (see “Special Mode LED Indications”).

3.5.1. Entering Factory Default Mode

Once in the Setup Mode, the UMI-32 shall exit Setup Mode and enter the Factory Default Mode when the pushbutton is pressed for exactly 10 times. When the UMI-32 enters the Factory Default Mode it will indicate so by blinking it’s LED as defined in section 5.5. In factory mode it will reinitialize the Setup Registers marked in the appendix. **Warning**, not all registers will be reset. For example, the room names will remain the same.

- Setup mode – press 5 times on pushbutton
- Factory default mode – press 10 times on pushbutton

3.5.2. Exiting Factory Default Mode

Once in the Factory Default Mode, the UMI-32 shall exit Factory Defaults Mode and enter the Normal Mode when the pushbutton is pressed 1 time. When the UMI-32 exits the Setup Mode it will indicate so by stopping the blinking of its LED as defined in section 5.5.

- Exit Factory Default mode – press 1 time on pushbutton

4. UPB Setup Registers

Like all UPB devices that conform to the UPB System Model, the UMI-32 has a set of non-volatile 8-bit registers known as UPB Setup Registers. The UMI-32 has a total of 256 UPB Setup Registers. These registers are used to define and configure how the UMI-32 will operate as well as to store other important information as described herein. The UMI-32 allows read/write access to its UPB Setup Registers via special UPB Messages communicated on the powerline. The UMI-32's Setup Registers are partitioned into three main groups the UPBID, the Configuration Registers, and the Scratch-Pad Registers as described below.

4.1. The UPBID

The UPBID is a set of 64 non-volatile registers that contains information that uniquely identifies the individual UPB device. The UMI-32 implements the UPBID in the first 64 Setup Registers. Table 2 below describes the Setup Registers that make up the UPBID. The table describes each register's use as well as its factory default value. The UPBSD document contains more detailed information about the UPBID.

Setup Register Field Name	Reg. Num.	Factory Default	Description
Network ID (NID)	0x00	255 (0xFF)	Unique identifier (1 – 255) for the UPB Network that this device communicates on.
Unit ID (UID)	0x01	40 (0x28)	Unique identifier (1 – 255) for this UPB device. Use Product ID for initial value.
Network Password (NPW)	0x02 – 0x03	4660 (0x1234)	Password designed to keep unauthorized users from modifying the Setup Registers of this device.
UPB Options (UPBOP)	0x04	00 (0x00)	Identifies UPB Options that are enabled for this device.
UPB Version (UPBVER)	0x05	01 (0x01)	Identifies the version of the UPB specification this device conforms to.
Manufacturer ID (MID)	0x06 – 0x07	00 (0x0000) OEM	Unique identifier of the manufacturer of this UPB device.
Product ID (PID)	0x08 – 0x09	40(0x28) UMI-32	The manufacturer's unique product identifier for this UPB device.

Setup Register Field Name	Reg. Num.	Factory Default	Description
Firmware Version (FWVER)	0x0A – 0x0B	Depends on the current F/W	Identifies the version of firmware in this device.
Serial Number (SERNUM)	0x0C – 0x0F	Set by the manufacturer	The manufacturer's unique serial number for this UPB device.
Network Name (NNAME)	0x10 – 0x1F	"New Network Name"	A human readable (ASCII) name for the UPB Network that this device communicates on.
Room Name (RNAME)	0x20 – 0x2F	"New Room Name "	A human readable (ASCII) name for the Room that this UPB device is installed in.
Device Name (DNAME)	0x30 – 0x3F	"New UMI-32 "	A human readable (ASCII) name for this UPB device.

Table 2 - The UMI-32's UPBID

4.2. The Configuration Registers

The Configuration Registers are a set of non-volatile registers that configure how a device will operate. The UMI-32 implements the Configuration Registers in the next 128 Setup Registers. The definition of the Configuration Registers is application dependent. See the appendix for a definition and location of Configuration Registers for the UMI-32 application. Note: the use of transmit and receive below is in reference to this unit. So when the unit receives a command it will refer to the Receive Component Link (RCL) table and when it detects an input state change it will transmit a command in the Transmit Component Link table.

Setup Register Field Name	Description
Receive Component Link Table Channel One	Set of command buffers that control the outputs. These can be initialized by the user or disabled. See appendix for description
Receive Component Link Table Channel Two	Set of command buffers that control the outputs. These can be initialized by the user or disabled. See appendix for description
Transmit Component Link Table Channel One	Set of command buffers that control the response to an input change. These can be initialized by the user or disabled. See appendix for description
Transmit Component Link Table Channel Two	Set of command buffers that control the response to an input change. These can be initialized by the user or disabled. See appendix for description
Transmit Component Link Table Channel Three	Set of command buffers that control the response to an input change. These can be initialized by the user or disabled. See appendix for description
Transmit Options register	Setup transmit message repeats.
Device Options Register	Setup the ZAP mode.
Unused	Reserved for future use

Table 3 - Configuration Registers

4.3. The Scratch-Pad Registers

The Scratch-Pad Registers are a set of non-volatile registers that the application can use for any purpose, with the exception of the registers defined below. The table describes each register's location and use. Note: there are no factory default values for these registers.

Setup Register Field Name	Reg. Num.	Description
Setup Mode Counter	0xFA	Count of number of times this device went into Setup Mode.
WERR Counter	0xFB	Count of number of times this device had an EEPROM Write Error.
POR Counter	0xFC	Count of number of times this device had a Power-On Reset.
BOR Counter	0xFD	Count of number of times this device had a Brown-Out Reset.
WDT Counter	0xFE	Count of number of times this device had a Watchdog Timer Reset.
MCLR Counter	0xFF	Count of number of times this device had a Master Clear Reset.

Table 4 - The UMI-32 Scratch-Pad Registers

5. Unit Status LED Operation

The UMI-32 has a single bi-color Light Emitting Diode (LED) that it uses to indicate its current status. The Status LED is configured by the settings in the LED Options Register (as defined below). The LED can be set to black (off), green, red or orange (alternating green and red). The color and state (blinking, flash, etc.).

5.1. The LED Options Register

The UMI-32 has an 8-bit LED Options Register. The unit constantly checks the register and updates the LED operating mode.

Bit	Name	Description
7	LED Mode	0 = Normal LED enabled 1 = Diagnostic LED enabled
4-6	Unused	Unused
3 - 2	Unused	Unused
1 - 0	Unused	Unused

Table 5 - The LED Options Register

5.2. LED Modes

There are two LED modes of operating. The LED has a color and a state (Flash or blinking).

5.2.1. Diagnostic LED Mode

Another name may have been verbose mode. In this mode the unit reflects many of the events that occur by the unit LED. This is the default LED mode the unit in which the unit ships. When the Diagnostic LED Mode is enabled, the UMI-32 shall turn the Status LED solid orange. Whenever the UMI-32 transmits a UPB message, it flashes the status LED red. Whenever the UMI-32 receives a valid UPB message, it flashes the status LED green. Whenever the UMI-32 receives an invalid UPB message, it shall indicate so by flashing the status LED black (off).

- Normal – solid orange
- Transmit message – flash red
- Receive valid UPB message – flash green
- Receive invalid message – flash black

5.2.2. Normal LED mode

When the Normal LED Mode (maybe a better name is quiet mode) is enabled the UMI-32 shall turn the Status LED off (black). Whenever the UMI-32 transmits a UPB message, it shall not give an indication. Whenever the UMI-32 receives a valid UPB message, it shall not give an indication. Whenever the UMI-32 receives an invalid UPB message, it shall not give an indication.

- Normal – solid black

5.2.3. Special Mode LED Indications

The UMI-32 has two special operating modes that it can be put into: Setup Mode and Factory Default Mode. When the UMI-32 is in Setup Mode it shall indicate so by blinking its status LED alternately between green and black (off). When the UMI-32 is in Factory Default Mode it shall indicate so by blinking its status LED

alternately between red and black (off). When the UMI-32 is in Normal Mode it shall indicate so by stopping the blinking and turning its status LED to the proper condition for the LED mode it is in (either solid orange or black).

- Setup mode – blinking green/black
- Test mode – Solid red
- Factory Default mode – blinking red/black
- Diagnostic mode – solid orange or black for Normal mode
- Zap mode – no change

5.2.4. Firmware Version LED Indication

When power is first applied, the UMI-32 shall use the status LED to indicate the version of the firmware it is running. It shall do this by flashing the LED four times. Each flash will either be red or green. The four flashes shall be such as to indicate the binary value that matches the least significant digit of the firmware version. A flash of red shall indicate a binary "zero" and a flash of green shall indicate a binary "one". As an example, if the firmware version is 4.15, then the LED will indicate the binary value for 5, which is 0-1-0-1. The UMI-32 will therefore flash its LED four times upon power-up: red-green-red-green.

- Version displayed on power up
- 4 digits
- Red = 0
- Green = 1

6. UPB Message Receiving

The UMI-32 shall be capable of receiving UPB messages from the powerline.

6.1. Receive Components

The UMI-32 uses the concept of Receive Components (as described in the UPBSD document) to configure its UPB Link Packet receiving behavior. The UMI-32 shall have sixteen 3-byte Receive Components for each of its two output channels implemented in non-volatile Configuration Registers as shown in the appendix. All Receive Components (referred to as Presets) are associated with the channel output state. Each Receive Component shall have an associated Link ID byte that is used when receiving UPB Link Packets. Each Receive Component shall also have an associated byte for holding an output state (and the 3rd byte is reserved for future use and to maintain uniformity with the Simply Automated data structure) for use in processing the "Activate" and "Deactivate" commands (see section 0 for details). The output state shall be interpreted as a 001 to be a relay closure and a 000 to be a relay open.

- Two channels
- 16 components for each channel
- 3 bytes to each component
- LID/State

Associated Channel	Component	LID	State	Reserved
1	1	Valid LID	Preset State	
1	2	Valid LID	Preset State	
1	x	Valid LID	Preset State	
2	1	Valid LID	Preset State	
2	2	Valid LID	Preset State	
2	x	Valid LID	Preset State	

Table 6 – Receive Component Link Table Structure (x=16)

6.2. Receiving UPB Link Packets

Whenever the UMI-32 channel receives a UPB Link Packet it will attempt to match its Destination ID to one of the valid Link IDs of its sixteen Receive Components. If a match is not found then that Link Packet is not for this UMI-32 channel and it **shall** be ignored. If a match is found then the UMI-32 **shall** accept the Link Packet for further processing. The particular Receive Component that had the Link ID match is "linked" to this Link Packet. Note, once a component is matched the search will stop. Therefore, it is not possible to have multiple copies of Link ID in a RCL table.

6.2.1. Activating/Deactivating Receive Components

The UMI-32 shall handle the special UPB Link Packet commands "Activate" and "Deactivate". When the UMI-32 accepts the "Activate" command it shall set its channel's output to values corresponding to the "linked" Receive Component preset, this shall be a zero or a one, opening or closing the relay. When the UMI-32 accepts the "Deactivate" command it shall set its channel's output to an open. A direct "Deactivate" command is not valid.

- Activate command will cause unit to follow instructions in RCL.
- Deactivate command sets the selected output channel to open.

6.3. Receiving the Core Command Message Set

The UMI-32 shall be capable of handling received UPB Messages from the UPB Core Command Message Set as described in the UPB System Description document

6.4. Receiving the Device Control Command Set

Besides handling the UPB Core Commands, the UMI-32 shall also handle the following set of UPB Commands from the UPB Device Control Command Set (see Table 7).

MDID (Hex)	Command Name	Command Description
0x20	Activate	Commands the UMI-32 to “activate” its linked Receive Component’s (Preset) output state.
0x21	Deactivate	Commands the UMI-32 to “deactivate” its linked Receive Component’s (Preset) output state.
0x22	Goto	Commands the UMI-32 to set the output to the specified output state
0x30	Report Status	Commands the UMI-32 to send back a Device State Report containing the current state of the inputs.
0x86	Device State Report	Report of the UMI-32 input and output interface’s state

Table 7 - UMI-32 UPB Device Control Commands

6.4.1. “Activate” Command

When the UMI-32 receives and accepts an “Activate” Command message (MDID = 0x20) in a UPB Link Packet it shall check it’s receive component tables for a valid preset, a preset that matches the LID of the “Activate” message. The state of this preset shall be used to configure the output. The output configured will correspond to the table where the selected preset resides, table 1 for channel 1 and table 2 for channel 2. Note, a direct “Activate” command is not valid.

MDID
0x20

Table 8 – Command Structure of Activate Command

6.4.2. “Deactivate” Command

When the UMI-32 receives and accepts a “Deactivate Link” Command message (MDID = 0x21) in a UPB Link Packet it shall set its channel (depends on the LID) output to open. Note, a direct deactivate is not valid.

MDID
0x21

Table 9 – Command Structure of Deactivate Command

6.4.3. "Goto" Command

6.4.4. Direct "Goto" Command

When the UMI-32 receives and accepts a "Goto" Command message (MDID = 0x22) in a UPB Direct Packet it shall set its channel output to the state specified in the command. The LL field indicates the state desired. The RR field is not used for the UMI-32 (see Lamp module). The CC field indicates the affected channel.

MDID	Level	Rate	Channel
0x22	LL	RR	CC
LL	0	Open	
	X	Close	
RR	-	-	
CC	0x00	Channel1	
	0x01	Channel2	

Table 10 – Command Structure of Direct Goto Command

6.4.5. Link "Goto" Command

When the UMI-32 receives and accepts a "Goto" Command message (MDID = 0x22) in a UPB Link Packet it **shall** change the output state for that LID to the state specified in the message, overriding the preset. See above for definitions of LL and RR.

MDID	Level	Rate
0x22	LL	RR

Table 11 – Command Structure of Linked Goto Command

6.4.6. "Report Status" Command

When the UMI-32 receives and accepts a "Report Status" Command message (MDID = 0x30) in a UPB Direct Packet it **shall** build and transmit a Device State Report message. Note that this Device State Report message is also generated in response to a change in the state of the input channels. The same action

occurs if the message is of a link or a direct type. A report status message that is triggered by a state change will be sent once, but if a report is requested it will use the value in the TXCTRL register.

Request:

MDID
0x30

Table 12 – Command Structure of Report Status Command

MDID	Inputs	Outputs
0x86	IN	OUT

Table 13 – Command Structure of Device State Report

IN bits	Input State
Bit0	Input Channel #1
Bit1	Input Channel #2
Bit2	Input Channel #3
Bit3	Unused
Bit4	Unused
Bit5	Unused
Bit6	Unused
Bit7	Unused

Table 10 - IN bit definitions

OUT bits	Output State
Bit0	Output #1
Bit1	Output #2
Bit2	Unused
Bit3	Unused
Bit4	Unused
Bit5	Unused
Bit6	Unused
Bit7	Unused

Table 13 - OUT bit definitions

7. UPB Message Transmitting

The UMI-32 shall be capable of transmitting UPB messages on the powerline.

7.1. Transmit Components

The UMI-32 uses the concept of Transmit Components (as described in the UPB System Description document) to configure its UPB Link Packet receiving behavior. The UMI-32 **shall** have two 4-byte Components for each of its three input channels implemented in non-volatile Configuration Registers. All Transmit Components (referred to as presets) are associated with the channel input state. Each Transmit Component **shall** have an associated Link ID byte that is used when transmitting UPB Link Packets. Each transmit Component **shall** also have an associated message buffer for holding a command to be sent upon an event on the channel input. This transmission is qualified by the event types allowed for that channel. A channel can be pre-specified for only closed-to-open events, only open-to-closed events, any events or no events. This is indicated by having a legal LID in the component and a command in the component buffer. If the component does not have a valid LID no command will be sent. For example in the table below is a representation of the memory structure used in the Transmit Component Table (TCL). If an Open-Close event is recognized on the channel 1 input, then Component 1 would be searched for a valid LID and command. If these are not found then no further action is needed. To turn off a component a 0xff is placed in the LID.

Associated Channel	Component	LID	Msg Buffer	Comments
Channel 1	Component 1	LID	Msg Buffer 1.1	Open-Close events
Channel 1	Component 2	LID	Msg Buffer 1.2	Close-Open Events
Channel 2	Component 1	LID	Msg Buffer 2.1	Open-Close events
Channel 2	Component 2	LID	Msg Buffer 2.2	Close-Open Events
Channel 3	Component 1	LID	Msg Buffer 3.1	Open-Close events
Channel 3	Component 2	LID	Msg Buffer 3.2	Close-Open Events

Table 14 - Transmit Component Table (TCL)

7.2. UPB Transmit Control Register

The UMI-32 **shall** have an 8-bit UPB Transmit Control Register implemented in the non-volatile Setup Registers that allows for the customization of its

transmissions. Figure 1 shows the UPB Transmit Control Register and explains the meanings of its various bits and fields.

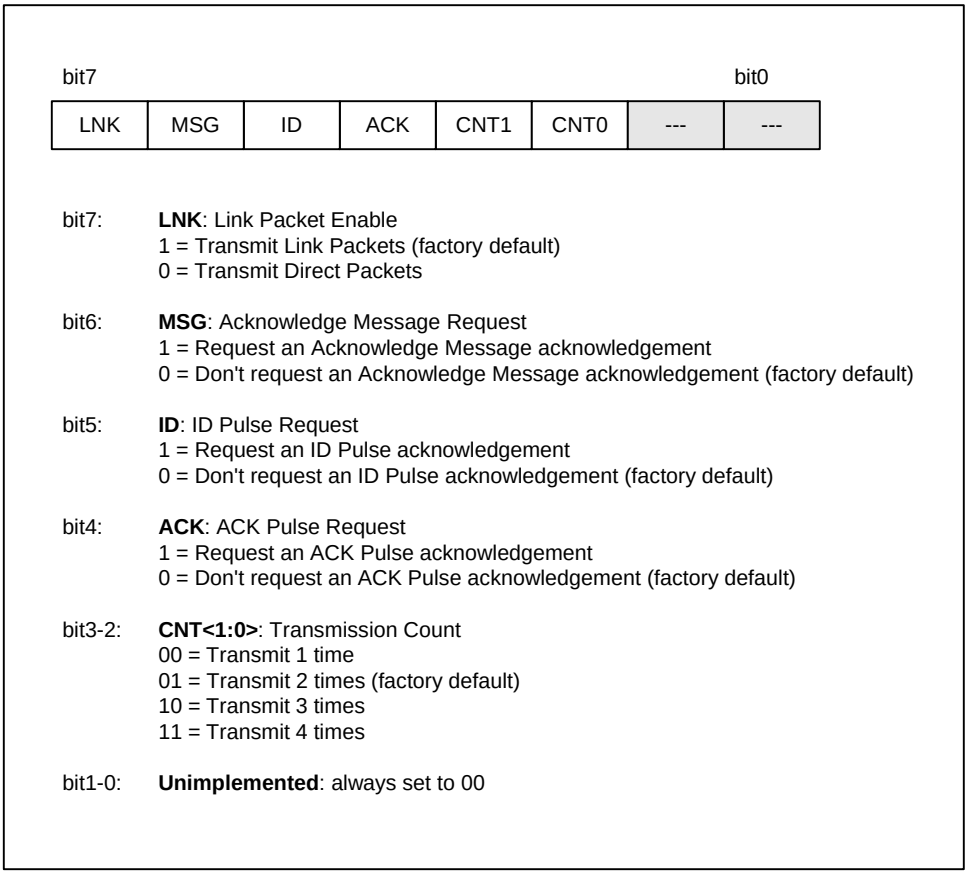

Figure 1 - The UPB Transmit Control Register

7.2.1. Link Packet Enable

Bit number seven (7) of the UPB Transmit Control Register is used to configure whether transmissions are sent in Link Packets or Direct Packets. If this bit is set to '0' the UMI-32 **shall** build and transmit all transmissions with the LNK-bit cleared in its Control Word (Direct Packet). If this bit is set to '1' the UMI-32 **shall** build and transmit all transmissions with the LNK-bit set in its Control Word (Link Packet). This is superseded in situations that require a specific transmission type.

7.2.2. Acknowledge Message Request

Bit number (6) of the UPB Transmit Control Register is used to configure whether an Acknowledge Message is requested. If this bit is set to '1' the UMI-32 **shall** build and transmit all transmissions with the MSG-bit set in its Control Word.

7.2.3. ID Pulse Request

Bit #5 of the UPB Transmit Control Register is used to configure whether an ID Pulse is requested from the receiver of any transmissions. If this bit is set to '1' the UMI-32 **shall** build and transmit all transmissions with the ID-bit set in its Control Word.

7.2.4. ACK Pulse Request

Bit #4 of the UPB Transmit Control Register is used to configure whether an ACK Pulse is requested from the receiver. If this bit is set to '1' the UMI-32 **shall** build and transmit all transmissions with the ACK-bit set in its Control Word.

7.2.5. Transmission Count

Bit #3 and #2 of the UPB Transmit Control Register are used to configure how many times to transmit a message in a row for each transmission event. If this field is set to 00 the UMI-32 **shall** transmit any transmission one time only. If this field is set to 01 or 10 or 11 the UMI-32 **shall** transmit any transmission two times or three times or four times respectively. The default is 2 repeats.

7.2.6. Transmission Component Link (TCL) Tables

There are three input channels and each has a TCL table associated with it. These tables have two components. Each component is a buffer where a UPB command can be stored. The first buffer in each table is executed when a "closed" event occurs on that channel input. The 2nd buffer command is executed for an "open" event. The buffer is four bytes in length. If a byte isn't used it is initialized to 0xff. The message buffer is defined as follows:

LID	MDID	MSG1	MSG2
LID	MDID	XX	XX

Table 15 – TCL Structure

LID – Valid LID as described in the Simply Automated system document.
MDID - Valid LID as described in the Simply Automated system document
XX - Supporting arguments

7.2.6.1. Default TCL Commands

By default the unit will send out an "Activate" command for a "close" event and a "Deactivate" command for an "open" event. This is true for each input channel. These values will be reinitialized in to the TCL table upon a factory default.

8. Power-Up Operation

Upon power-up the UMI-32 will enter Normal Mode and will display its current firmware version by blinking its LED four times as described in section 5.2.4. It will then read the saved output state for each channel from non-volatile memory and set each output to the correct state (note: the factory default is “open” for all channels). The inputs shall be sampled and used to initialize the state machine. The LED will be initialized in accordance with the LED mode that has been preset.

9. Appendix - Memory Map

Note: the * concatenated to the address, in the following tables, indicates the address data is reset during a factory default operation.

9.1. Memory Map - UPBID

Address	Name	Default	Comment
0*	Network ID	0xff (255)	Network ID
1*	Unit_ID	0x28 (40)	Unit ID set to Product ID
2*	Password_H	0x12	Password
3*	Password_L	0x34	
4*	UPB Options	0x00	UPB Options
5*	UPB Versions	0x01	UPB Version
6*	MANID_H	0x00	OEM
7*	MANID_L	0x00	OEM
8*	PRODID_H	0x00	UMI-32 Product ID
9*	PRODID_L	0x28 (40)	
A*	FIRMVER_H	0x00	Firmware version
B*	FIRMVER_L	0x00	
C*	SERNUM_1	0x00	Serial number
D*	SERNUM_2	0x00	
E*	SERNUM_3	0x00	
F*	SERNUM_4	0x00	
10	Network Name	“N”	Network Name
11	Network Name	“e”	
12	Network Name	“t”	
13	Network Name	“w”	
14	Network Name	“o”	
15	Network Name	“r”	
16	Network Name	“k”	
17	Network Name	“ ”	

18	Network Name	“1”	
19	Network Name		
1A	Network Name		
1B	Network Name		
1C	Network Name		
1D	Network Name		
1E	Network Name		
1F	Network Name		
20	Room Name	“N”	Room Name
21	Room Name	“e”	
22	Room Name	“w”	
23	Room Name	“ “	
24	Room Name	“R”	
25	Room Name	“o”	
26	Room Name	“o”	
27	Room Name	“m”	
28	Room Name	“ “	
29	Room Name	“N”	
2A	Room Name	“a”	
2B	Room Name	“m”	
2C	Room Name	“e”	
2D	Room Name		
2E	Room Name		
2F	Room Name		
30	Device Name	“N”	Device Name
31	Device Name	“e”	
32	Device Name	“w”	
33	Device Name	“ “	
34	Device Name	“C”	
35	Device Name	“M”	
36	Device Name	“0”	
37	Device Name	“1”	
38	Device Name		
39	Device Name		
3A	Device Name		
3B	Device Name		
3C	Device Name		
3D	Device Name		
3E	Device Name		
3F	Device Name		

Table 16 – UPBID Memory Map

9.2. Receive Component Link Table Channel 1

Location	Name	Default	Definition	Channel
40*	Component 1	0xC4 (196)	LID	RCL Channel1
41*	Component 1	0x01 (01)	Close	RCL Channel1
42*	Component 1	0xFF (0255)	Unused	RCL Channel1
43*	Component 2	0xC5 (197)	LID	RCL Channel1
44*	Component 2	0x00 (00)	Open	RCL Channel1
45*	Component 2	0xFF (0255)	Unused	RCL Channel1
46	Component 3	0xFF (255)	Unused	RCL Channel1
47	Component 3	0xFF (255)	Unused	RCL Channel1
48	Component 3	0xFF (0255)	Unused	RCL Channel1
49	Component 4	0xFF (255)	Unused	RCL Channel1
4A	Component 4	0xFF (255)	Unused	RCL Channel1
4B	Component 4	0xFF (0255)	Unused	RCL Channel1
4C	Component 5	0xFF (255)	Unused	RCL Channel1
4D	Component 5	0xFF (255)	Unused	RCL Channel1
4E	Component 5	0xFF (0255)	Unused	RCL Channel1
4F	Component 6	0xFF (255)	Unused	RCL Channel1
50	Component 6	0xFF (255)	Unused	RCL Channel1
51	Component 6	0xFF (0255)	Unused	RCL Channel1
52	Component 7	0xFF (255)	Unused	RCL Channel1
53	Component 7	0xFF (255)	Unused	RCL Channel1
54	Component 7	0xFF (0255)	Unused	RCL Channel1
55	Component 8	0xFF (255)	Unused	RCL Channel1
56	Component 8	0xFF (255)	Unused	RCL Channel1
57	Component 8	0xFF (0255)	Unused	RCL Channel1
58	Component 9	0xFF (255)	Unused	RCL Channel1
59	Component 9	0xFF (255)	Unused	RCL Channel1
5A	Component 9	0xFF (0255)	Unused	RCL Channel1
5B	Component10	0xFF (255)	Unused	RCL Channel1
5C	Component10	0xFF (255)	Unused	RCL Channel1
5D	Component10	0xFF (0255)	Unused	RCL Channel1
5E	Component11	0xFF (255)	Unused	RCL Channel1
5F	Component11	0xFF (255)	Unused	RCL Channel1
60	Component11	0xFF (0255)	Unused	RCL Channel1
61	Component12	0xFF (255)	Unused	RCL Channel1
62	Component12	0xFF (255)	Unused	RCL Channel1
63	Component12	0xFF (0255)	Unused	RCL Channel1
64	Component13	0xFF (255)	Unused	RCL Channel1

65	Component13	0xFF (255)	Unused	RCL Channel1
66	Component13	0xFF (0255)	Unused	RCL Channel1
67	Component14	0xFF (255)	Unused	RCL Channel1
68	Component14	0xFF (255)	Unused	RCL Channel1
69	Component14	0xFF (0255)	Unused	RCL Channel1
6A	Component15	0xFF (255)	Unused	RCL Channel1
6B	Component15	0xFF (255)	Unused	RCL Channel1
6C	Component15	0xFF (0255)	Unused	RCL Channel1
6D	Component16	0xFF (255)	Unused	RCL Channel1
6E	Component16	0xFF (255)	Unused	RCL Channel1
6F	Component16	0xFF (0255)	Unused	RCL Channel1

Table 17 - Memory Map, Receive Component Link Tables, Channel 1

9.3. Receive Component Link Tables, Channel 2

Address	Name	Default	Definition	Channel
70*	Component 1	0xC7 (198)	LID	RCL Channel2
71*	Component 1	0x01 (01)	Close	RCL Channel2
72*	Component 1	0xFF (0255)	Unused	RCL Channel2
73*	Component 2	0xC8 (199)	LID	RCL Channel2
74*	Component 2	0x00 (00)	Open	RCL Channel2
75*	Component 2	0xFF (0255)	Unused	RCL Channel2
76	Component 3	0xFF (255)	Unused	RCL Channel2
77	Component 3	0xFF (255)	Unused	RCL Channel2
78	Component 3	0xFF (0255)	Unused	RCL Channel2
79	Component 4	0xFF (255)	Unused	RCL Channel2
7A	Component 4	0xFF (255)	Unused	RCL Channel2
7B	Component 4	0xFF (0255)	Unused	RCL Channel2
7C	Component 5	0xFF (255)	Unused	RCL Channel2
7D	Component 5	0xFF (255)	Unused	RCL Channel2
7E	Component 5	0xFF (0255)	Unused	RCL Channel2
7F	Component 6	0xFF (255)	Unused	RCL Channel2
80	Component 6	0xFF (255)	Unused	RCL Channel2
81	Component 6	0xFF (0255)	Unused	RCL Channel2
82	Component 7	0xFF (255)	Unused	RCL Channel2
83	Component 7	0xFF (255)	Unused	RCL Channel2
84	Component 7	0xFF (0255)	Unused	RCL Channel2
85	Component 8	0xFF (255)	Unused	RCL Channel2
86	Component 8	0xFF (255)	Unused	RCL Channel2
87	Component 8	0xFF (0255)	Unused	RCL Channel2
88	Component 9	0xFF (255)	Unused	RCL Channel2
89	Component 9	0xFF (255)	Unused	RCL Channel2
8A	Component 9	0xFF (0255)	Unused	RCL Channel2
8B	Component10	0xFF (255)	Unused	RCL Channel2
8C	Component10	0xFF (255)	Unused	RCL Channel2
8D	Component10	0xFF (0255)	Unused	RCL Channel2
8E	Component11	0xFF (255)	Unused	RCL Channel2
8F	Component11	0xFF (255)	Unused	RCL Channel2
90	Component11	0xFF (0255)	Unused	RCL Channel2
91	Component12	0xFF (255)	Unused	RCL Channel2
92	Component12	0xFF (255)	Unused	RCL Channel2
93	Component12	0xFF (0255)	Unused	RCL Channel2
94	Component13	0xFF (255)	Unused	RCL Channel2
95	Component13	0xFF (255)	Unused	RCL Channel2
96	Component13	0xFF (0255)	Unused	RCL Channel2
97	Component14	0xFF (255)	Unused	RCL Channel2
98	Component14	0xFF (255)	Unused	RCL Channel2

99	Component14	0xFF (0255)	Unused	RCL Channel2
9A	Component15	0xFF (255)	Unused	RCL Channel2
9B	Component15	0xFF (255)	Unused	RCL Channel2
9C	Component15	0xFF (0255)	Unused	RCL Channel2
9D	Component16	0xFF (255)	Unused	RCL Channel2
9E	Component16	0xFF (255)	Unused	RCL Channel2
9F	Component16	0xFF (0255)	Unused	RCL Channel2

Table 18 - Memory Map, Receive Component Link Tables, Channel 2

9.4. Transmit Component Link Tables

Address	Component	Default Value	Definition	Channel
A0*	Component1	0xBE (190)	LID	TCL Channel1
A1*	Component1	0x20 (32)	MDID	TCL Channel1
A2*	Component1	0xFF (255)	MSG Byte	TCL Channel1
A3*	Component1	0xFF (255)	MSG Byte	TCL Channel1
A4*	Component2	0xBF (191)	LID	TCL Channel1
A5*	Component2	0x21 (33)	MDID	TCL Channel1
A6*	Component2	0xFF (255)	MSG Byte	TCL Channel1
A7*	Component2	0xFF (255)	MSG Byte	TCL Channel1
A8*	Component1	0xC0 (192)	LID	TCL Channel2
A9*	Component1	0x20 (32)	MDID	TCL Channel2
AA*	Component1	0xFF (255)	MSG Byte	TCL Channel2
AB*	Component1	0xFF (255)	MSG Byte	TCL Channel2
AC*	Component2	0xC1 (192)	LID	TCL Channel2
AD*	Component2	0x21 (33)	MDID	TCL Channel2
AE*	Component2	0xFF (255)	MSG Byte	TCL Channel2
AF*	Component2	0xFF (255)	MSG Byte	TCL Channel2
B0*	Component1	0xC2 (193)	LID	TCL Channel3
B1*	Component1	0x20 (32)	MDID	TCL Channel3
B2*	Component1	0xFF (255)	MSG Byte	TCL Channel3
B3*	Component1	0xFF (255)	MSG Byte	TCL Channel3
B4*	Component2	0xC3 (194)	LID	TCL Channel3
B5*	Component2	0x20 (32)	MDID	TCL Channel3
B6*	Component2	0xFF (255)	MSG Byte	TCL Channel3
B7*	Component2	0xFF (255)	MSG Byte	TCL Channel3

Table 19 – Memory Map, Transmit Component Link Table, channels 1:3

9.5. Reserved

Address	Definition	Default
B8-BF	Reserved	FF (255)
C4-EF	Reserved	FF (255)

Table 20 – Memory Map, Reserved EEPROM Space

9.6. Unit Options

Address	Definition	Default	Comments
C0*	TXCTRL	0x84 (132)	Link message, 2 repeats
C1*	LED Options	0x80 (138)	Diagnostic state
C2*	Relay State	0x00 (00)	Both channels open
C3*	Device Options	0x00 (00)	Reserved

Table 21 – Unit Options

9.7. Scratch Pad

Address	Default	Definition
F0	0xFF (255)	Reserved
F1	0xFF (255)	Reserved
F2	0xFF (255)	Reserved
F3	0xFF (255)	Reserved
F4	0xFF (255)	Reserved
F5	0xFF (255)	Reserved
F6	0xFF (255)	Reserved
F7	0xFF (255)	Reserved
F8	0xFF (255)	Reserved
F9	0xFF (255)	Reserved
FA	0x00 (00)	EEPROM Setup Mode Counter
FB	0x00 (00)	EEPROM Write Error Counter
FC	0x00 (00)	Power-On-Reset Counter
FD	0x00 (00)	Brown-Out-Reset Counter
FE	0x00 (00)	WDT-Reset Counter
FF	0x00 (00)	MCR Counter

Table 22 – Memory Map, Scratch Pad Area

10. References

Document	Version	Date	Owner
The UPB System Description (UPBSD)	1.2	1/4/05	Simply Automated
UMI-32 Schematic			

Table 23 – References

11. Glossary

Term	Definition	Comments
UPB	Universal Power Bus technology	
UPBSD	The Simply Automated UPB System Description	
Management Devices	An electronic device that programs UPB devices with the configuration. Examples are the Web Mountain Server or the HCA UpStart tool.	
UpStart tool	A management device provided by HCA.	
Web Mountain Server	A home networking Linux-based controller that is capable of handling the UPB protocol.	
Configuration registers	EEPROM memory locations that define the operational variables for the device.	
UPBID registers	EEPROM memory locations that define the identification and address for the unit.	
Scratch Pad registers	EEPROM memory locations that hold counters used by the Core software.	
Register	EEPROM memory location	
EEPROM		
Watch Dog Timer (WDT)		
Framework		
Operational Variables		
Configuration		
Light Emitting Diode (LED)		
Setup Registers		
Transmit		

Receive		
Transmit Component link Table		
Receive Component link Table		
OEM		

Table 24 - Glossary