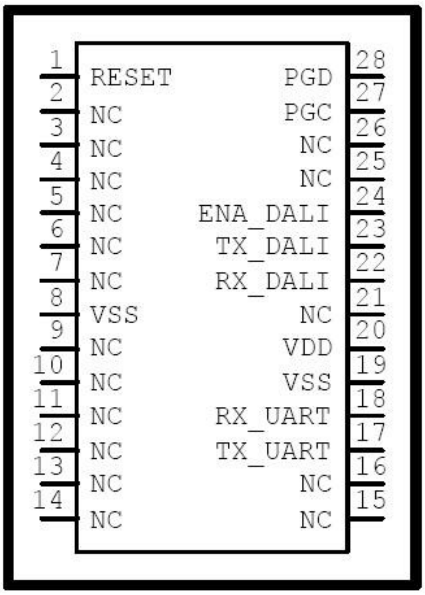
DALI BRIDGE

- Interface between Dali bus and uart bus
- Internal Clock and Reset
- Dali bus overload protection
- Dali master and/or Dali data logger function
- "Dalichip button" handle (under development)
- Single supply
- 28pin ssop
- Available demo kit
- Release 1.29



The Dali Bridge chip is able to make an easy interface between a serial uart ttl bus and a Dali bus. With a few command it's possible:

- drive a Dali network from microcontroller or pc
- have the log of the data in the network
- both previous function

Pins function:

Pin Number	*Pin Name*	*Pin Type*	*Level(*)*	*Pin Function*
1	RESET	IN	ST	Chip reset, active low
2 3 4 5 6 7 9 10 11 12 13 14 15 16 21 25 26	NC	OUT	CMOS	Unconnected pins
8 19	VSS	POWER	---	Ground
17	TX_UART	OUT	CMOS	Uart data ttl output
18	RX_UART	IN	ST	Uart data ttl input
20	VDD	POWER	---	Power supply
22	RX_DALI	IN	TTL	Dali rx data
23	TX_DALI	OUT	CMOS	Dali tx data
24	ENA_DALI	OUT	CMOS	Dali power enable
27	PGC	IN	ST	Program pin
28	PGD	BIDIR	ST/CMOS	Program pin

(*) ST schmitt trigger input with CMOS level
CMOS compatible CMOS input or output
TTL compatible input

Serial Command

The serial commands, to send to the board by serial bus RS232 TTL, must have a simple protocol. Each packet in the protocol has three bytes, some of them are not relevant, and it depends of the command send in the packet.

The next part of the document has the information about the protocol command, but about the meaning of Dali command it's necessary to consult the Dali manual. The complete Dali specification doesn't concern this manual.

Serial protocol

9600 bps or 19200bps, No parity, 8bit data, 1 stop bit.

Packet structure

The entire packet, transmitted or received in the bridge, have three bytes:

- the first (BYTE1) identify the command of the Bridge protocol packet
- the others two bytes (BYTE2 + BYTE3) are the data of the Dali packet (the data in the Dali bus), or they can be data to set bridge parameters.

Bridge internal Commands

Bridge Control Word

BYTE1 = 04h = Bridge parameter setting

- PC transmission: 04h,data_H,00h

  data_H.bit0 = enable logger Dali bus function (1 enable, 0 disable)

  data_H.bit1 = baud rate selection (1 19200, 0 9600). When you change the baudrate the answer at the command will be at the old speed. At start up baud rate is 9600.

  data_H.bit2 = enable handle "button dalichip" (1 enable, 0 disable) (under development)

  data_H.bit3 to data_L.bit7 unused

  To have compatibility with previous release, the value of these flags are not stored in eeprom: they will be 0 each time the bridge switch on.

- Bridge reply: FFh,00h,00h

The answer will be in 1ms.

Bridge software release

BYTE1 = 05h = Bridge software release

- PC transmission: 05h,00h,00h
- Bridge reply: FEh,release,00h

The answer will be in 1ms.

Bridge reset program

BYTE1 = 06h = Bridge reset program

- PC transmission: 06h,00h,00h
- Bridge reply: FFh,00h,00h

The answer will be in 1ms.

After the answer it's necessary to wait at least 300ms before that the Bridge IC it's running right.

Bridge Status Word

BYTE1 = 07h = Bridge report Status

- PC transmission: 07h,00h,00h
- Bridge reply: FEh,data_H,00h
  data_H.bit0 = status logger Dali bus function (1 enabled, 0 disabled)
  data_H.bit1 = error last Dali reception (1 error present)
  data_H.bit2 = overload Dali bus (1 overload)
  data_H.bit3 = “button dalichip” status (1 enabled, 0 disabled)
  data_H.bit4 to data_H.bit7 unused

The answer will be in 1ms.

The logger Dali function enable the transmission, on the uart bus, of all the Dali data bus, sent from others units (slave or master), but not from this unit (the pc/microcontroller already knows these values). It enables the monitor of the Dali network.

IMPORTANT: logger is run time function, there is not memory buffer on the bridge IC. This means that the uart transmission of log data will be synchronous with dali signals, and not with pc request: it's not master/slave (pc/bridge) like the other commands.

Store first packet

BYTE1 = 08h = send sequence command first packet

- PC transmission: 08h,data_H,data_L
- Bridge reply: FFh,00h,00h

It stores the data_H and data_L values in the bridge; these values are the first data packet sent in the Dali bus by the command “send sequence” (see later).

The answer will be in 1ms.

Store second packet

BYTE1 = 09h = send sequence command second packet

- PC transmission: 09h,data_H,data_L
- Bridge reply: FFh,00h,00h

It stores the data_H and data_L values in the bridge; these values are the second data packet sent in the Dali bus by the command “send sequence” (see later).

The answer will be in 1ms.

Store third packet

BYTE1 = 0Ah = send sequence command third packet

- PC transmission: 0Ah,data_H,data_L
- Bridge reply: FFh,00h,00h

It stores the data_H and data_L values in the bridge; these values are the third data packet sent in the Dali bus by the command “send sequence” (see later).

The answer will be in 1ms.

EEprom Write Values

BYTE1 = 20h = Write the EEprom value at the address given.

- PC transmission: 22h,data_H,data_L
  data_H is the eeprom address to read, from 0 to FFh
  data_L id the value to write in EEprom
- Bridge reply: FEh,eeprom_value,00h

The answer will be in 10ms at the end of write.

EEprom Read Values

BYTE1 = 21h = read EEprom value at the address given.

- PC transmission: 22h,data_H,0x00
  data_H is the eeprom address to read, from 0 to FFh.
- Bridge reply: FEh,eeprom_value,00h

The answer will be in 1ms.

Dalichip button (under development)

Dalichip.com extended Dali function to have the possibility to connect switches on the dali bus. With the Bridge IC, and the PC based software *DaliChipSW* it's possible to have Dali standard communication and implement a network of button (maximum 127).

To do this the Bridge must have two info:

- info1: it needs to know for each button what is the slave to drive (short, group or broadcast address): this means 127 values of slave address, one for each switches
- info2: then it needs to know, if a button drive a group, the short address of one element of the group: this means 16 values of slave address, one for each group

It stores these values in EEprom, it content will be:

- address [00h..7Fh] there are values about info1
- address [80h..8Fh] there are values about info2
- address [90h..FDh] unused
- address [FEh] command 20h, adjust bus timing**,** Half period
- address [FFh] command 20h, adjust bus timing, Duty_cycle

The info1 and info2 values should be make according to the standard addressing format of Dali specification for short, group or broadcast addressing. According to Dali Standard the MASK value FFh is reserved to indicate when there is not slave to drive.

Adjust bus timing

Writing at the two last EEprom address it's possible to change the half period and the duty cycle of the wave sent on the Dali bus. In this way it's possible to correct the waveform to satisfy the waveform Dali specification, and adjust the hardware problems.

The half period value has a base unit of 8µs: typical period of Dali signal is 832µs => 832µs/2 = 416µs = 52*8µs => half_period = 52.

The duty cycle value is a percentage, and the number is between 0 and 100.

We suggest using this command only to calibrate the bridge (the develop kit boards are already calibrated).

Bridge rx/tx Command

In the next command the (BYTE2 + BYTE3) are Dali data: BYTE2 is Dali command and BYTE3 is Dali data. Meaning of these two bytes is function of the Dali command to implement. The same is for BYTE1, it identifies command for the Bridge: “single tx”, “double tx”, “tx and rx” and "packet tx".

Send single packet

BYTE1 = 10h = send single packet

- PC transmission: 10h,data_H,data_L
- Bridge reply: FFh,00h,00h

It sends data_H e data_L in the Dali bus.

The answer will be in 20ms, after the transmission in the Dali bus.

Send double packet
BYTE1 = 11h = send double packet

- PC transmission: 11h,data_H,data_L
- Bridge reply: FFh,00h,00h

It sends data_H and data_L in the Dali bus two times: second tx will in the 100ms after the first tx (according to Dali specification).
The answer will be in 100ms, after the second transmission in the Dali bus.

Send and receive packet
BYTE1 = 12h = send and receive packet

- PC transmission: 12h,data_H,data_L
- Bridge reply (if at least one slave reply): FEh,slave_data,00h
- Bridge reply (if not slave reply): 00h,00h,00h

It sends data_H e data_L in the Dali bus and waits the answer from one slave.
The answer will be in 30ms, or immediately after the slave answer.

Send sequence packet
BYTE1 = 13h = send sequence packet

- PC transmission: 13h,00h,00h
- Bridge reply: FFh,00h,00h

It sends on the Dali bus the sequence of packet stored in the bridge by the commands "Store first packet", "Store second packet" and "Store third packet". The three packets will be send at least in 100ms. This command is useful in the test phase of the network slaves.
The answer will be in 100ms, after the three transmissions on the Dali bus.

Answer to master (PC or microcontroller)
The first byte of the Bridge answer is important.
If the answer packet comes because there was a request packet from the master (master-slave model), the values of the first byte of the answer can be:

- 00h = no rx from slaves (used in "send receive packet program")
- FFh = the second and third bytes (after FFh) are not relevant
- FEh = only the second byte (after FEh) is relevant, not the third
- FDh = both the bytes (after FDh) are relevant

If the logger function is enabled, then the Bridge sends each packet of the Dali network. In this case the master reception is fully asynchronous and doesn't depend from master request. (not master-slave model). In this situation the values of the first byte can be:

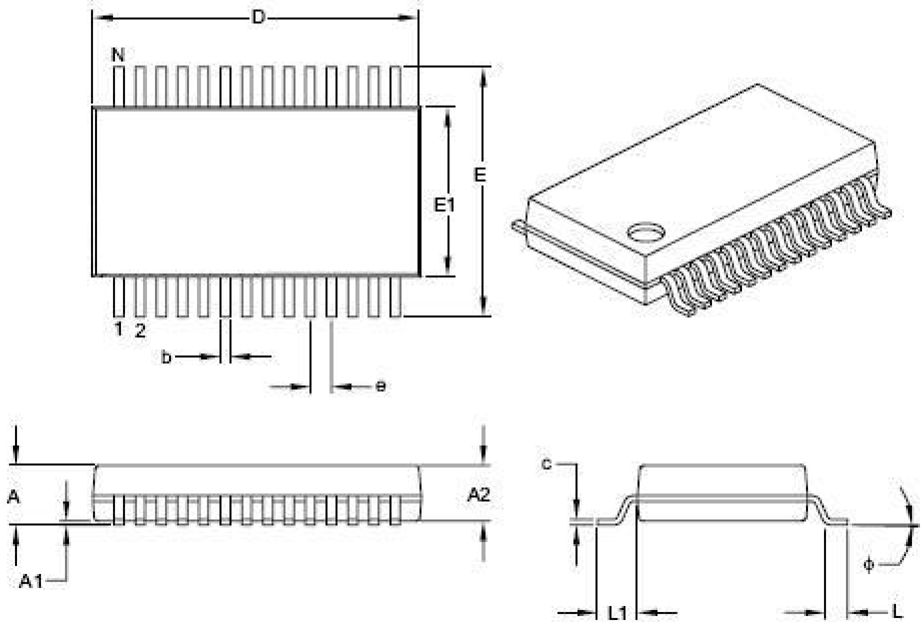
- EEh = only the second byte (after EEh) is a valid data, not the third. It happens when the Bridge receives a packet message from a slave device.
- EDh = both bytes (after EDh) are valid data. It happens when the Bridge receives a packet message from another Master device (Dali bus is typically single master model, this means that this bridge is used like logger function, not master function).
- ECh = both bytes (after ECh) are valid data. It happens when the Bridge receives a packet message from another Master device, the message is equal to the previous and there is a few than 100ms between these two messages. This means that the two packets come from a “double send” command. Like the previous Dali bus is typically single master model, this means that this bridge is used like logger function, not master function

To have the circuit example of connection check the demo kit manual.

Electrical specifications:

Characteristics		***Min***	***Tip***	***Max***	***Units***
Power Supply (VDD)		3	---	5.5	V
Oscillator Frequency		---	4	---	MHz
V_{IL}	ST (VDD>4.5V)	VSS	---	0.8	V
	ST (VDD<4.5V)	VSS	---	0.15VDD	V
	ST (RESET pin)	VSS	---	0.2VDD	V
	CMOS	VSS	---	0.2VDD	V
V_{IH}	ST (VDD>4.5V)	2.0	---	VDD	V
	ST (VDD<4.5V)	0.25VDD+0.8	---	VDD	V
	ST (RESET pin)	0.8VDD	---	VDD	V
	CMOS	0.8VDD	---	VDD	V
V_{OL}		VSS	---	0.6	V
V_{OH}		VDD-0.7	---	VDD	V
Operating temperature		-40	---	+85	°C

Package dimension:



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N	28		
Pitch	e	0.65 BSC		
Overall Height	A	–	–	2.00
Molded Package Thickness	A2	1.65	1.75	1.85
Standoff	A1	0.05	–	–
Overall Width	E	7.40	7.80	8.20
Molded Package Width	E1	5.00	5.30	5.60
Overall Length	D	9.90	10.20	10.50
Foot Length	L	0.55	0.75	0.95
Footprint	L1	1.25 REF		
Lead Thickness	c	0.09	–	0.25
Foot Angle	ϕ	0°	4°	8°
Lead Width	b	0.22	–	0.38

: