NAME OF THE PROJECT	THE DON LIGHT 2.0			**PROJECT**	32002100100
BRIEFING REFERENCE	**32032100101**	**REVISION**	01.00.00	**DATE OF REVISION**	2021.11.10
CUSTOMER	GENERAL	**DATE OF CREATION**	2021.11.10		
RESPONSIBLE IN CUSTOMER	DENNES SMALLEGANGE				
RESPONSIBLE IN LUXAE	JOSE MANUEL GARZA				

1. SCOPE OF THIS DOCUMENT

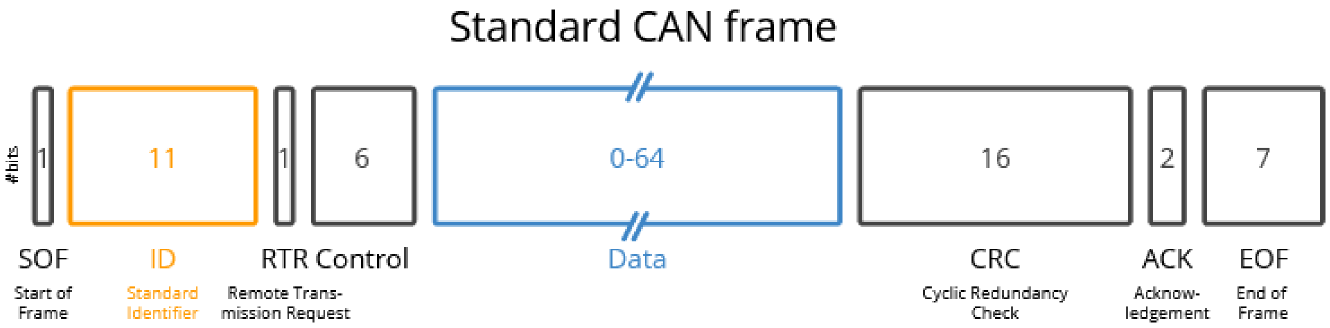
This document explains how to connect and control any configuration of sets of the DONLIGHT 2.0, using a third party CANBUS controller.

A minimum knowledge of serial communications is considered essential to take advantage of this document.

2. CANBUS SPECIFICATION

The specs of this can bus system are:

- STANDARD 11BIT ID
- CAN 2.0B COMPLIANT
- BAUD RATE = 125kbps
- ISO11898-2 PHY
- 8 BYTES DATA LENGHT



3. CONSIDERATIONS

The use of a breakout box excludes the use of any other CANBUS controller, except a polaris/vigilink controller.

To integrate a breakout box together with another third party's controller, please refer to Marelko.

From now on, we will only refer to the ID and DATA fields of the CANBUS message.

11
0-64
ID
Standard Identifier
Data

We use hexadecimal (0xaaa) and binary (aaaaab) notation when we talk about values of the can id and data bytes.

A function will be considered active if its corresponding bit (inside DATA) is set to 1. As soon as that bit is set to 0, the function will be considered not active. That means that the control of the different functions of the system is carried out by level (as if it were a switch), and not by edge (as if it were a pushbutton).

BIT VALUE OF THE FUNCTION ON THE CANBUS MESSAGE	STATUS OF THE FUNCTION
1b	ON
0b	OFF

To be able to use a function, the color corresponding to that function needs to be present in the installed sets. We will not be able to activate the warning lights rear red if the red LEDs are not installed in the sets.

4. ASSIGNED CAN IDs AND MESSAGE TIMING

List of standard can message IDs, needed to the correct behaviour of the complete system

CAN MESSAGE TYPE	CAN MESSAGE ID	NOTES
SYNCH	0x100	**Sync each 500miliseconds**
LIGHT CONTROL	0x219	**Function control**

5. PREDEFINED COMMUNICATION MESSAGES

5.1. SYNCH MESSAGE

The controller must send a synchronization message every 500ms this is used as the entire system light pattern clock.

5.2. LIGHT CONTROL MESSAGE

This message should be sent only to notify a change in the INPUTS status, no need to send the same message containing the same status twice.

The Function control message shows the status of all inputs, the entire system will react instantaneously to new changes here, the DONLIGHT module configuration will define how it reacts to the received Control CANBUS message.

As an example, if a Control message with the front emergency light is sent, a front configured module will trigger its lights but a rear configured module wont.

6. CONTROL MESSAGES

As discussed previously, the control message will have this format:

ID	BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5	BYTE 6	BYTE 7

Going into the detail of the function of each bit, we can see the following:

11 bit CAN ID

0x219 - BECAUSE IT IS A LUXAE'S CONTROL MESSAGE

BYTE 0

X	X	X	X	X	X	X	X
7	6	5	4	3	2	1	0
0x80	0x40	0x20	0x10	0x08	0x04	0x02	0x01

BYTE 1

WHITE AB	WHITE DC	WHITE HG	AUX	REAR BLUE	FRONT BLUE	REAR AMBER	FRONT AMBER
7	6	5	4	3	2	1	0
0x80	0x40	0x20	0x10	0x08	0x04	0x02	0x01

BYTE 2

REAR RED	FRONT RED	CRUISE AM	CRUISE BL	NIGHT MODE	RIGHT TURN	LEFT TURN	WHITE FE
7	6	5	4	3	2	1	0
0x80	0x40	0x20	0x10	0x08	0x04	0x02	0x01

BYTE 3

X	RESERVED	STROBE CD	STROBE HG	STROBE FE	STROBE GFED	STROBE AB	STROBE HABC
7	6	5	4	3	2	1	0
0x80	0x40	0x20	0x10	0x08	0x04	0x02	0x01

BYTE 4

X	X	X	X	X	X	X	X
7	6	5	4	3	2	1	0
0x80	0x40	0x20	0x10	0x08	0x04	0x02	0x01

BYTE 5

X	X	X	X	X	X	X	X
7	6	5	4	3	2	1	0
0x80	0x40	0x20	0x10	0x08	0x04	0x02	0x01

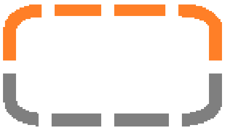
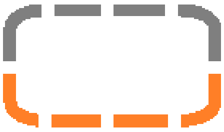
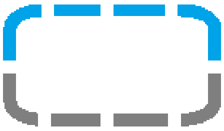
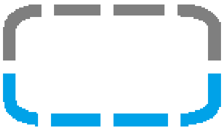
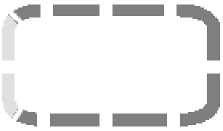
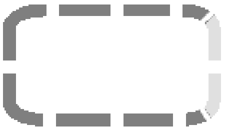
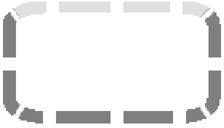
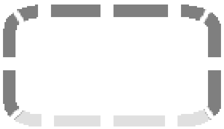
BYTE 7

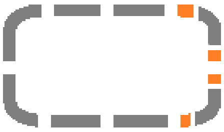
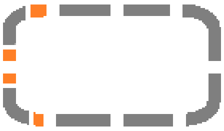
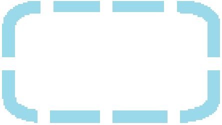
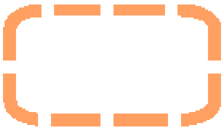
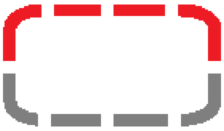
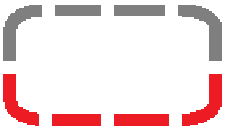
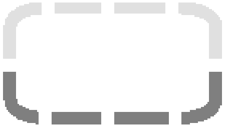
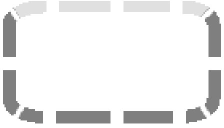
X	X	X	X	X	X	X	X
7	6	5	4	3	2	1	0
0x80	0x40	0x20	0x10	0x08	0x04	0x02	0x01

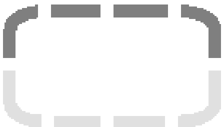
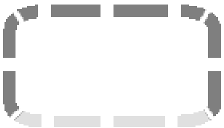
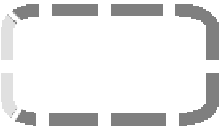
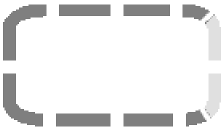
BYTE 8

X	X	X	X	X	X	X	X
7	6	5	4	3	2	1	0
0x80	0x40	0x20	0x10	0x08	0x04	0x02	0x01

NOW WE ARE GOING TO EXPLAIN WITH MORE DETAIL THE FUNCTION OF EACH BIT IN THE NEXT TABLE:

FUNCTION	DESCRIPTION	LOCATION OF CONTROL BIT IN CAN MESSAGE (MASK)	LABEL ON THE CONTROL MESSAGE
1	Warning Lights Front amber 	BYTE 1 Bit0: xxxx xxx1	FRONT AMBER
2	Warning Lights Rear Amber 	BYTE 1 Bit1: xxxx xx1x	REAR AMBER
3	Warning Lights Front Blue 	BYTE 1 Bit2: xxxx x1xx	FRONT BLUE
4	Warning Lights Rear Blue 	BYTE 1 Bit3: xxxx 1xxx	REAR BLUE
5	5A AUX output (activates green aux lights)	BYTE 1 Bit4: xxx1 xxxx	AUX
6	White scene lights on H and G (blue and amber go off if these are on) 	BYTE 1 Bit5: xx1x xxxx	WHITE HG
7	White scene lights on D and C (blue and amber go off if these are on) 	BYTE 1 Bit6: x1xx xxxx	WHITE DC
8	White scene lights on A and B (blue and amber go off if these are on) 	BYTE 1 Bit7: 1xxx xxxx	WHITE AB
9	White scene lights on F and E (blue and amber go off if these are on) 	BYTE 2 Bit0: xxxx xxx1	WHITE FE

10	Left Turn (domes 1 and 6 on left modules) illumination follows status of the input feed, it should be alternating with the car. If a warning light is flashing (Amber or blue) the warning light will go off. Please keep in mind it there needs to be a delay. 	BYTE 2 Bit1: xxxx xx1x	LEFT TURN
11	Right Turn (domes 1 and 6 on right modules) illumination follows status of the input feed, it should be alternating with the car. If a warning light is flashing (Amber or blue) the warning light will go off. Please keep in mind it there needs to be a delay. 	BYTE 2 Bit2: xxxx x1xx	RIGHT TURN
12	Dim for blue /amber / red (day/night)	BYTE 2 Bit3: xxxx 1xxx	NIGHT MODE
13	Cruise lights blue 	BYTE 2 Bit4: xxx1 xxxx	CRUISE BL
14	Cruise lights amber 	BYTE 2 Bit5: xx1x xxxx	CRUISE AM
15	Warning Lights Front red 	BYTE 2 Bit6: x1xx xxxx	FRONT RED
16	Warning Lights Rear red 	BYTE 2 Bit7: 1xxx xxxx	REAR RED
17	Flashing white high H, A, B, C (Stroboscope effect) 	BYTE 3 Bit0: xxxx xxx1	STROB HABC
18	Flashing white high A and B (Stroboscope effect) 	BYTE 3 Bit1: xxxx xx1x	STROBE AB

19	Flashing white high G, F, E, D (Stroboscope effect) 	BYTE 3 Bit2: xxxx x1xx	STROBE GFED
20	Flashing white high F and E (Stroboscope effect) 	BYTE 3 Bit3: xxxx 1xxx	STROBE FE
21	Flashing white high H and G (Stroboscope effect) 	BYTE 3 Bit3: xxx1 xxxx	STROBE HG
22	Flashing white high C and D (Stroboscope effect) 	BYTE 3 Bit3: xx1x xxxx	STROBE CD
23	Reserved	BYTE 4 Bit3: x1xx xxxx	RESERVED
where x – don’t care and 1 – function activation bit (state 1 for enable)			

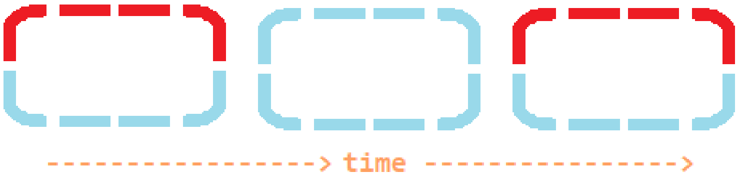
6. FUNCTION PRIORITIES

THERE IS DEFINED A PRIORITY IN THE FUNCTIONS AS INDICATED BELOW:

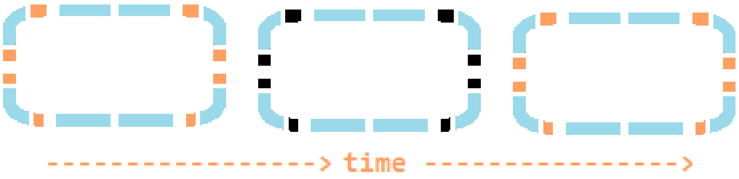
1. MAXIMUM PRIORITY: TURN/HAZARD LIGHTS
2. SCENE LIGHTS
3. EMERGENCY/WARNING
4. CRUISE
5. MINIMUM AUXILIAR OUTPUT

How DONLIGHT2.0 manages concurrent patterns:

- Functions with higher priority will override the lower ones.
  he functions with lower priority will still be working as long as the LEDs are not being used by a higher priority function.
  For example, Cruise lights blue+ emergency front red would look like:



- Special functions like turning/hazard lights will **RESERVE** the LED when the pattern reaches an OFF state.
  For example, Cruise lights blue+ both turning sinals (hazard) would look like:


time