Name	DID? Y/N	EID CY/N	CAN Flags?	Simple Node Ms	Wire Hex Nibble 1 (3 bits) Priority Group (2 bits)	Wire Hex Nibble 2 (5 bits) Type (4 bits)	Wire Hex Nibble 3 DID/EID/Flags (3 bits)	Wire Hex Total 16 bits hex	CAN Var Total 15 bits hex	Format 3 bits	Type Byte 8 bits	Flag bit A 0 default 1 bit	Flag bit B 1 default 1 bit	Flag bit C 0 default 1 bit	Flag bit D 1 default 1 bit	CAN Content
Base Messages																
Initialization Complete	N	N			0	8	0	3080	108F	1	0x08					Full Source Node ID
Verify Node ID Number	Y	N		Y	0	10	4	30A4	6ddd	6XXX	6 dest NIDa					MTI byte 0x0A
Verify Node ID Number	N	N		Y	0	10	0	30A0	10AF	1	0	0x0A				
Verified Node ID Number	N	N			0	11	0	30B0	10BF	1	1	0x0B				Full Source Node ID
Protocol Support Inquiry	Y	N		Y	1	14	4	32E4	6ddd	6XXX	6	0x2E				MTI byte 0x2E
Protocol Support Reply	Y	N		Y	1	15	4	32F4	6ddd	6XXX	6	0x2F				MTI byte 0x2F, protocol flags
Optional Interaction Rejected	Y	N			0	12	4	30C4	6ddd	6XXX	6 dest NIDa					MTI byte 0x0C, MTI, error, optional information
Terminate Due to Error	Y	N			0	13	4	30D4	6ddd	6XXX	6 dest NIDa					MTI byte 0x0D, MTI, error, optional information
Event Exchange Messages																
Identify Consumers	N	Y		Y	1	4	2	3242	124F	1	0	0x24				EventID (no room for DestID!)
Consumer Identify Range	N	Y			1	5	2	3252	125F	1	1	0x25				EventID w mask (no room for DestID!)
Consumer Identified	N	Y	Y		1	6	3	3263	126f	0	1	0x26		valid	uncertain	EventID (no room for DestID!)
Identify Producers	N	Y		Y	1	8	2	3282	128F	1	0	0x28				EventID (no room for DestID!)
Producer Identify Range	N	Y			1	9	2	3292	129F	1	1	0x29				EventID w mask (no room for DestID!)
Producer Identified	N	Y	Y		1	10	3	32A3	12Af	0	1	0x2A		valid	uncertain	EventID (no room for DestID!)
Identify Events	Y	N		Y	1	11	2	32B2	6ddd	6	6	dest NIDa				MTI byte 0x2B
Identify Events	N	N		Y	1	11	0	32B0	12BF	1	0	0x2B				
Learn Event	N	Y		Y	1	12	2	32C2	12CF	1	0	0x2C				EventID
Producer/Consumer Event Report	N	Y		Y	1	13	2	32D2	12DF	1	0	0x2D				EventID
Datagram Messages																
Datagram (General)	Y	N		Y	2	0	4	3404	4/5ddd	4/5XXX	4,5	dest NIDa				Data (0-8 bytes)
Datagram Received OK	Y	N		Y	2	12	4	34C4	6ddd	6XXX	6	dest NIDa				MTI byte
Datagram Rejected	Y	N		Y	2	13	4	34D4	6ddd	6XXX	6	dest NIDa				MTI byte, error code
Stream Messages																
Stream Initiate Request	Y	N			2	14	4	34E4	6ddd	6XXX	6	dest NIDa				MTI byte, buffer size (2 bytes), Source Stream ID (1 byte), reserved byte, flags (tagged=0x80)
Stream Initiate Reply	Y	N			2	15	4	34F4	6ddd	6XXX	6	dest NIDa				MTI byte 0x4B,buffer size (2 bytes), Source Stream ID (1 byte), Dest Stream ID, flags (tagged=0x80; error info)
Stream Data Send	Y	N			3	9	4	3694	7ddd	7XXX	7	dest NIDa				(stream IDs inferred on CAN); 8 bytes data
Stream Data Proceed	Y	N			3	10	4	36A4	7ddd	7XXX	6	dest NIDa				MTI byte, Stream IDs (2 bytes)
Stream Data Complete	Y	N			3	11	4	36B4	7ddd	7XXX	6	dest NIDa				MTI byte, Stream IDs (2 bytes); optional length (4 bytes)

0 gets more priority — coding: 1=carries EID, 2=carries DID — d=dest NIDa, f=flags — 0=simple MTI, 1=complex MTI

Full value must be checked! — 4=DestID datagram, 5=DestID datagram last segment, 6=DestID non-Stream, 7=DestID stream data — If flags not specified, send and check 1 bits

Places these appear in code:
prototypes/Arduino/libraries/OpenLCB/OpenLcbCan.h
prototypes/CBUS-PIC/canlib/frametypes.c