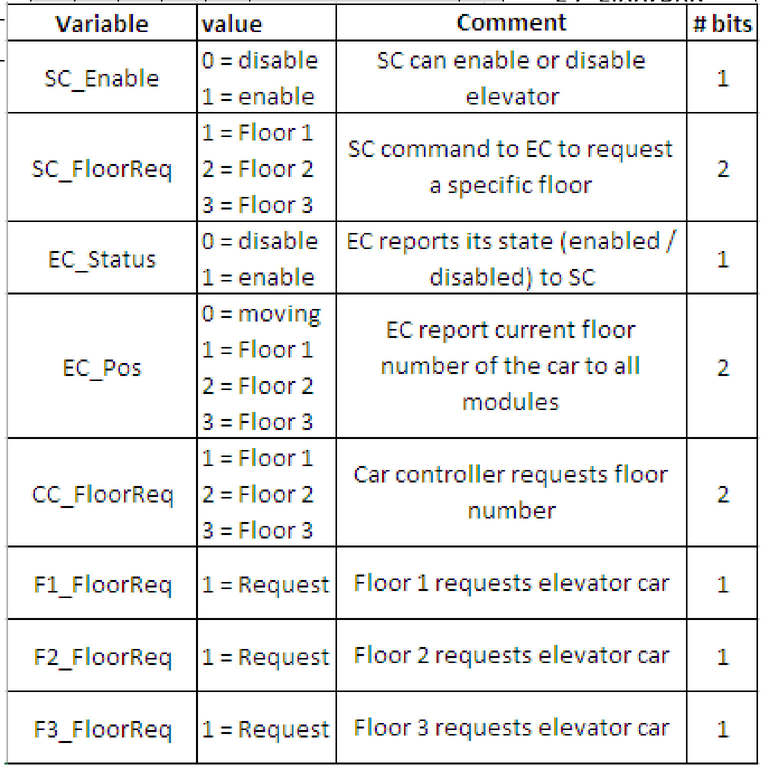
Theory of CAN

Common CAN Protocol Message Layout

CAN ID (Hex)	Transmitter	Recipient(s)	DLC	Byte 0							
				7	6	5	4	3	2	1	0
0x100	SC	EC	1						SC_Enable	SC_FloorReq	
0x101	EC	ALL	1						EC_Status	EC_Pos	
0x200	CC	SC	1							CC_FloorReq	
0x201	F1	SC	1								F1_FloorReq
0x202	F2	SC	1								F2_FloorReq
0x203	F3	SC	1								

Legend

SC	*Supervisory Controller*
EC	*Elevator Controller*
CC	*Car Controller*
F1	*Floor 1 Controller*
F2	*Floor 2 Controller*
F3	*Floor 3 Controller*



Variable	value	Comment	# bits
SC_Enable	0 = disable 1 = enable	SC can enable or disable elevator	1
SC_FloorReq	1 = Floor 1 2 = Floor 2 3 = Floor 3	SC command to EC to request a specific floor	2
EC_Status	0 = disable 1 = enable	EC reports its state (enabled / disabled) to SC	1
EC_Pos	0 = moving 1 = Floor 1 2 = Floor 2 3 = Floor 3	EC report current floor number of the car to all modules	2
CC_FloorReq	1 = Floor 1 2 = Floor 2 3 = Floor 3	Car controller requests floor number	2
F1_FloorReq	1 = Request	Floor 1 requests elevator car	1
F2_FloorReq	1 = Request	Floor 2 requests elevator car	1
F3_FloorReq	1 = Request	Floor 3 requests elevator car	1

Explanation of CAN Theory

CAN Protocol Message Layout

Each CAN message is defined by:

- **CAN ID** (identifies the sender type)
- **Transmitter** (who sends it)
- **Recipient(s)** (who reads it)
- **DLC** (Data Length Code – always 1 byte in this doc)
- **Byte 0 Data** (content of the message)

The specific meaning of bits inside Byte 0 (like how many bits for floor number vs enable state) is not detailed but likely follows a bitfield or enum.

Message Breakdown

CAN ID	From	To	DLC	Byte 0 Content
0x100	SC	EC	1	SC_Enable, SC_FloorReq
0x101	EC	ALL	1	EC_Status, EC_Pos
0x200	CC	SC	1	CC_FloorReq
0x201	F1	SC	1	F1_FloorReq
0x202	F2	SC	1	F2_FloorReq
0x203	F3	SC	1	F3_FloorReq

🔄 System Operation Summary

1. A **user presses a floor button** (either in the car or on a floor panel).
2. The respective controller (CC or Fx) sends a **request message** to SC.
3. The SC updates its **state machine** and sends movement commands to EC.
4. EC moves the elevator and sends back **position/status updates** on CAN.
5. All controllers can read this to update LEDs or display current floor.