Author: Noah Batcher

Date: 7/22/2025

Goal:

Design a gateway chip to interact with real CAN protocol([SAE](https://en.wikipedia.org/wiki/SAE_International) standard J1979) for OBD-II systems. Gateway should be able to interact with CAN networks to perform all 10 basic and required services/modes. For now, services 01, 02, 03, 04, 07,and 0A are being implemented.

A screenshot of a computer

AI-generated content may be incorrect.

<https://en.wikipedia.org/wiki/OBD-II_PIDs#Service_03>

Above is the table of 10 required PIDS and the link to the wiki where all the information on CAN was pulled from.

CAN Frame types:

SF = single frame response

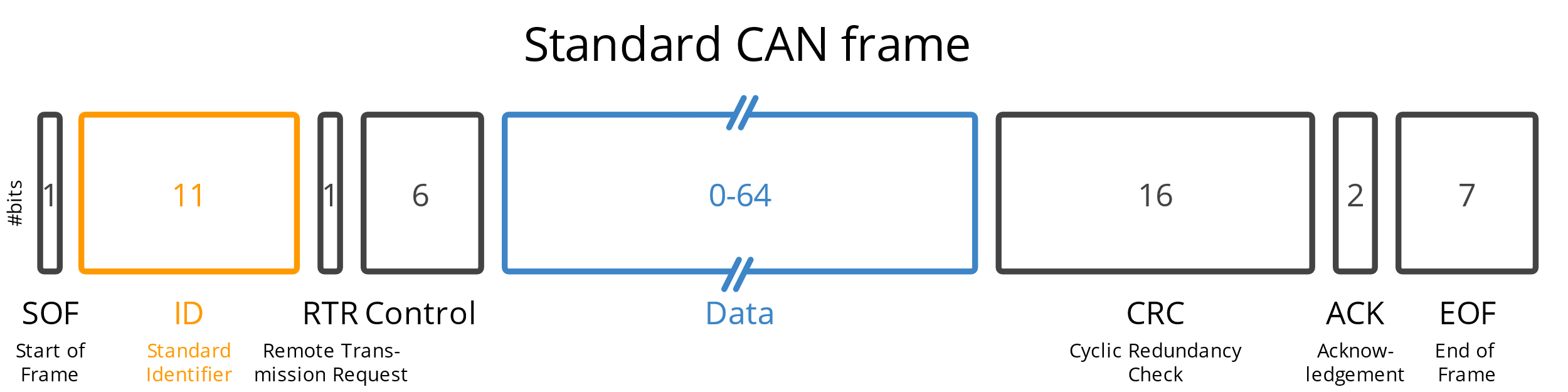
FF = First Frame of Multi Frame response

CF = Consecutive Frame response

FC = Flow Control Frame

CAN Frame:

CAN frames have two main parts that are important for communication with ECUs. The first is the ID which is used to filter to the correct ECU. There is an ID the diagnostic tool will use to ask each ECU individually (0x7E0 is the ID for requesting from the ECM) and a general one that will go to all ECUs (0x7DF). The other section that is important for talking to ECU’s is the Data section. This section is always 8 bytes and will contain requests and responses respectively.

<https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.csselectronics.com%2Fpages%2Fcan-bus-simple-intro-tutorial&psig=AOvVaw02toCuPe16PCtGw36W9Tjc&ust=1753301556145000&source=images&cd=vfe&opi=89978449&ved=0CBYQjRxqFwoTCNiFq7Kj0Y4DFQAAAAAdAAAAABAp>

CAN Request:

To make a CAN request set the ID to either broadcast on all ECUs or better select the specific ECU ID to broadcast a request too. Then in the data section have the first byte mark how many bytes are in the message only counting mode bytes and data bytes. For a service 3 request it would look like (hex: 01,03,00,00,00,00,00,00). For services 1,2, and 9 they will need three total bytes, one for the number of mode and data bytes (2), one for the mode byte, and one for the PID request (service 1 hex: 02,01,05,00,00,00,00,00). A mode byte is where the request mode request or response will be. The PID is a Proportional-Integral-Derivative (Vehicle speed, engine coolant temp, fuel system status, etc.).

CAN Response:

A can response will come in two different forms (single frame SF or multi frame which will have first frames FF and consecutive frames CF). Below shows how the frames would be used for service 3 (stored DTCs). For SF it will count the total number of bytes 0-8 in the message then have a mode byte with the rest being data. The FF will have the value 0x10 to show it is a first frame then the number of bytes followed by the mode byte. The rest will be data, but for DTCs they come it sets of two bytes and cannot be split across frames so it will have one byte padding. CF has a the first byte as 0x20 + N where N is the number consecutive frame it is. Then The rest is data and for service 3 it will have one padding byte, so it doesn’t split a DTC across frames. CF does not contain mode bytes.

SF hex: 04, 43,XX,XX,00,00,00,00

FF hex: 10,12,43,XX,XX,XX,XX,00

CF hex: 2N,XX,XX,XX,XX,XX,XX,00

DTC packing:

A DTC will be packed into two bytes so that the first two bits of the first byte represent the letter, the next two being the first number, then rest of the byte being the next number, and then the last two numbers will be half of the second byte.

A white text with black text

AI-generated content may be incorrect.