HTML:

```
if (settings.chargertype == ChevyVolt)
  {
    msg.id  = 0x30E;
    msg.len = 1;
    msg.buf[0] = 0x02; //only HV charging , 0x03 hv and 12V charging
    Can0.write(msg);

    msg.id  = 0x304;
    msg.len = 4;
    msg.buf[0] = 0x40; //fixed
    if ((chargecurrent * 2) > 255)
    {
      msg.buf[1] = 255;
    }
    else
    {
      msg.buf[1] = (chargecurrent * 2);
    }
      msg.buf[2] = highByte(uint16_t((settings.ChargeVsetpoint * settings.Scells ) *
2));
      msg.buf[3] = lowByte(uint16_t((settings.ChargeVsetpoint * settings.Scells ) *
2));
    Can0.write(msg);
  }

  if (settings.chargertype == Coda)
  {
    msg.id  = 0x050;
    msg.len = 8;
    msg.buf[0] = 0x00;
    msg.buf[1] = 0xDC;
      msg.buf[2] = highByte(uint16_t((settings.ChargeVsetpoint * settings.Scells ) *
10));
      msg.buf[3] = lowByte(uint16_t((settings.ChargeVsetpoint * settings.Scells ) *
10));

    msg.buf[4] = 0x00;
    if ((settings.ChargeVsetpoint * settings.Scells)*chargecurrent < 3300)
    {
      msg.buf[5] = highByte(uint16_t(((settings.ChargeVsetpoint * settings.Scells) *
chargecurrent) / 240));
      msg.buf[6] = highByte(uint16_t(((settings.ChargeVsetpoint * settings.Scells) *
chargecurrent) / 240));
    }
    else //15 A AC limit
    {
      msg.buf[5] = 0x00;
      msg.buf[6] = 0x96;
    }
    msg.buf[7] = 0x01; //HV charging
    Can0.write(msg);
  }
}
```

Well i couldnt get the Volt heater to work yet i admit. I now have a SW CAN shield for Arduino Due, but i have put this on hold in interest of getting the car out. Later on i will work on heater and A/C compressor some more.

I think your problem may be in nature of charger communication.

1. every 30ms you have to send heartbeat msg with ID 030E
data field itself is 03, (charge HV and 12V battery) only a single byte. If you miss that, charger wont function

2. every 500ms you then send command msg ID 0304
data field is 4 bytes 40 C8 03 34 to command 10A up to 410Vdc

Charger has to be connected to 12V battery (not just PSU) as well as HV battery. Then when you send command msg, you hear relay inside click and charger starts in relation of power. If you command 410Vdc and battery is close it will reduce amps to charge at 3kW.

Also you can command say 8A@395V charge with data field 40 A0 03 16.

Also beware! There are TWO CAN lines on Volt charger. Maybe for pairing? If you are sending on the wrong channel (pins 10 and 11)charger will ignore you. See my blog for correct pinout.

JB Note: One talks at 500kbs to the vehicle, one talks at 125kbs to the battery - should be that arrangement anyway.

Mark,

i'll try to capture some data with parallel DashDAQ and CANUSB/CANHACKER running. I hope to decode some new CANIDs these days...

This are the known IDs so far:

0C9 Byte 5 Accelerator 0 (0%) to 254 (100%)
0F1 Byte 2 Brake 0 (0%) to Unknown (254?) Typical pressure on brake pedal generates about 30.
135 Byte 1 Drive Position 0=Park, 1=Neutral, 2=Drive/L, 3=Reverse
1A1 Byte 8 Accelerator 0 (0%) to 254 (100%)
1C3 Byte 8 Accelerator 0 (0%) to 254 (100%)
1EF Bytes 3-4 Gas Engine RPM RPM
1F5 Byte 4 Shift Position PRNDL 1=Park, 2=Reverse, N=Neutral, D=Drive, L=Low
206 Bytes 1-2 Battery SOC .250kWh Units possibly .244kWh
32A Bytes 1-4 GPS Latitude Milliarcseconds
32A Bytes 5-8 GPS Longitude Milliarcseconds
3E9 Bytes 1-2 Speed 1/100 MPH 55MPH would be 5500 (0x157c)

regards,

Johannes

Yeah..... DashDAQ + CANUSB makes it !

Here is some data from the charger:

5EC Byte 3 Charging Current units 0.2 A (ex. 14.2 A = 0x47)
5EC Byte 4 Charging Voltage units 2 V (ex. 222V = 0x6F)

....going on hacking.

Johannes

there it goes..

5EC Byte 5 Outside Air Temperature in °C unit 0.5°C offset +40 (exp.: 0x60 = 8.0 °C)
5EC Byte 6 Outside Air Temperature filtered in °C unit 0.5°C offset +40

;-)

If the first byte in 0BC has bit 8 set (leftmost), the car is on; if bit 5 is set, the car is off.

For charging, I can only tell indirectly (e.g. by monitoring the battery SOC in 206, or the battery kW usage in 3e3 bytes 4-5).

Speed is in the first two bytes of 3E9, as MPH*100 (so hex of 0102 would be decimal 258, or a very slow 2.58MPH).

4C1 byte 5 has the outside temperature, in degrees Fahrenheit with an offset of 50 (so 0x76 or 118 decimal would be 68 degrees F).

I'd like to see if I am I reading this correctly before I start poking around on my canbus.
Is the first part saying it is sending a message to

mode 49
ID 547
device 7E4
(for config)

looking for data to display
mode 49,
ID 563 to 7E4
looking fro response
from device 5EC?

Is 43h = current 70=Volt? Any idea what are the units?

Also why is this example with OAT so different from the previous OAT example you gave?