CANSee

2019.01.12

Technical Documentation

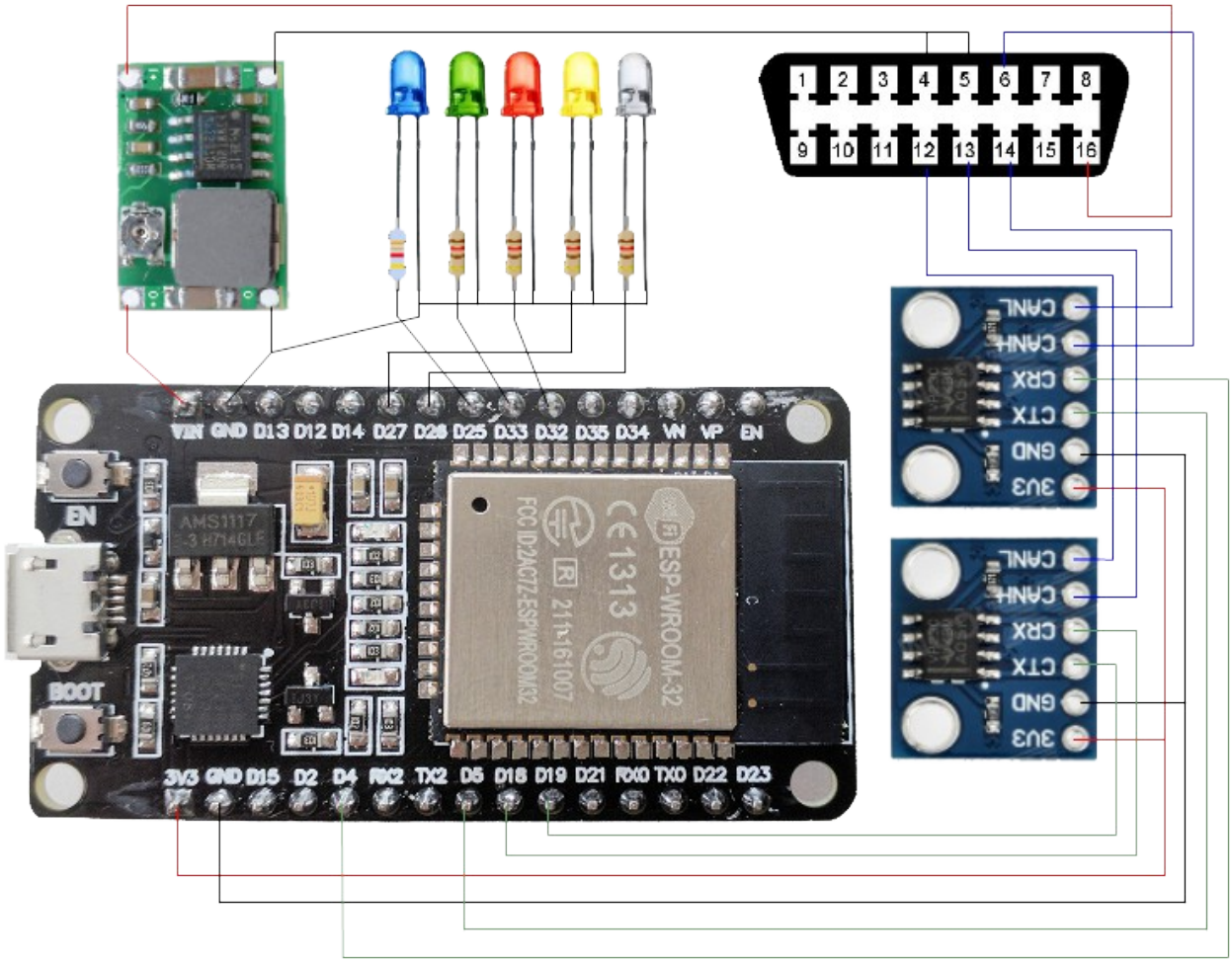
Table of Contents

A) Hardware

1) List of components

- 1x ODB case with connector
  https://www.aliexpress.com/item/CASE-ELM327-OBD2-Connector-J1962m-Plug-with-Enclosure-16pin-Male-Connector/32827624441.html
- 1x DOIT ESP32 DEVKIT or similar
  https://www.aliexpress.com/item/ESP32S-ESP-32S-ESP32-ESP-32-CP2102-Wireless-WiFi-Bluetooth-Development-Board-Micro-USB-Dual-Core/32842731763.html
- 1x Mini DC-DC converter
  https://www.aliexpress.com/item/10Pcs-Mini-DC-DC-12-24V-To-5V-3A-Step-Down-Power-Supply-1-8V-2/32848669016.html
- 2x SN65HVD230 CAN bus transceiver
  https://www.aliexpress.com/item/SN65HVD230-CAN-Bus-Transceiver-Communication-Module-Thermal-Protection-Slope-Control-for-Arduino/32851150816.html
- 5x 3mm LED (blue, green, red, yellow, white)
- 4x 120 Ω resistor
- 1x 3.3 kΩ resistor

2) Connecting things together



The DC-DC converter must be adjusted to an output of 5V. **Don't mix up CAN connections on the ODB2 connector, as you may break your car's bus system if improperly connected!**

3) Pin usage

Pin	Connection
4	RX of main CAN bus
5	TX of main CAN bus
18	RX of multimedia CAN bus
19	TX of multimedia CAN bus
25	blue LED
26	white LED
27	yellow LED
32	red LED
33	green LED

B) Software

1) The repository

The repository of the CANSee code if located on Gitlab

https://gitlab.com/jeroenmeijer/cansee

2) Configuration settings

%

3) Serial configuration commands

<table>
<tr><th>Command</th><th>Explanation</th></tr>
<tr><td>a</td><td>Show all buffered free CAN_frames</td></tr>
<tr><td>z</td><td>Reboot (needed for all configuration changes except debug mode)</td></tr>
<tr><td>r</td><td>Reset configuration to default and reboot.</td></tr>
<tr><td>gxxx</td><td>Get free frame with id xxx</td></tr>
<tr><td>ixxx,yyyy</td><td>Get ISO-TP frame with id xxx, PID yyyy</td></tr>
<tr><td>nxxx,config</td><td>set EEPROM configuration
xxx is determined by this table
<table>
<tr><td>100</td><td>Set the flags configuration as hex as ssbbwwlldd [01010200ff]
ss: Serial over USB
00 for off, any other for on
bb: Bluetooth
00 for off, any other for on
ww: WiFi
00 for off, 01 for Station mode, 02 for Soft Access Point mode.
For station mode a DHCP server is assumed.
ll: Use leds
00 for off, ff for on
dd: Debug
00 for off, ff for on. A bit pattern can be used and added
01: show CANbus receive free CAN_frame_std
02: show CANbus receive ISO-TP CAN_frame_std
04: show Commands issued
08: show free frame Commands
10: show ISO-TP Commands</td></tr>
<tr><td>200</td><td>Set the Bluetooth name [CANSee]</td></tr>
<tr><td>201</td><td>Set the Bluetooth pin [1234, not implemented by the API!!]</td></tr>
<tr><td>300</td><td>Set the WiFi ssid for Soft Access Point mode [CANSee]</td></tr>
<tr><td>301</td><td>Set the WiFi password for Soft Access Point mode [CANSeeMe]</td></tr>
<tr><td>400</td><td>Set the WiFi ssid for Station mode [Home]</td></tr>
<tr><td>401</td><td>Set the WiFi password for Station mode [Password]</td></tr>
<tr><td>500</td><td>Set CANbus parameter for CAN0 in hex as sprxtx
sp: speed in 25 kbps increments. [0x14 for can0, 0x0a for can1]
rx: Rx pin of transceiver [4 for can0, 0x12 for can1]
tx: Tx pin of transceiver [5 for can0, 0x13 for can1]
Note that this is a very advanced setting and selecting pins that are internally used on your board (notably 6-11) will crash the board. Also pins 34 and up are input only.</td></tr>
<tr><td>501</td><td>Set CANbus parameter for CAN1 in hex as sprxtx (see above).</td></tr>
</table>
</td></tr>
</table>

		Usage of the semi-secondary CAN bus by switching the single controller to another pin pair is not supported and basically fails spectacularly.