Teeny 4.0 OBDII CAN-Bus ECU Simulator

USER GUIDE V1.0

Product name	Teensy 4.0 OBDII CAN-Bus ECU Simulator with Enclosure
Model number	TEENSY4.0-OBDII-SIM-EN
Manufacturer	SK Pang Electronics Ltd

Contents

Table of Contents

1. Introduction

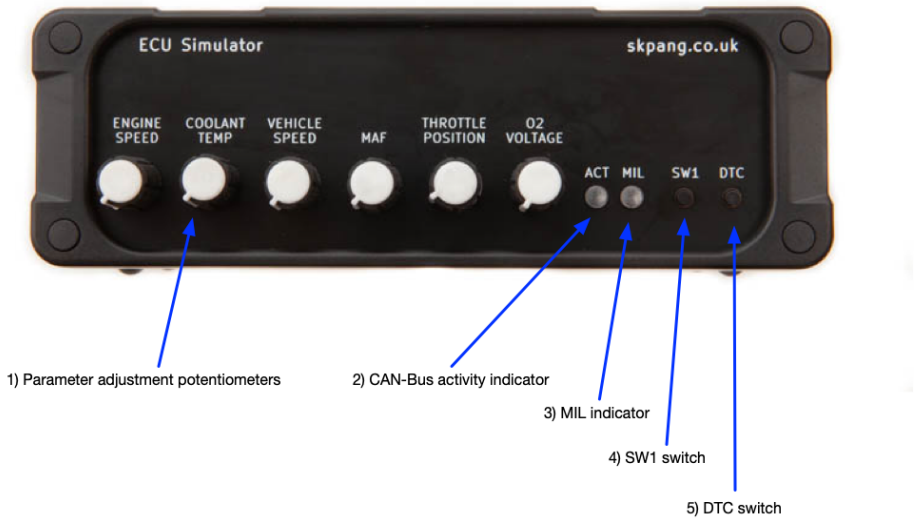
This is a CAN-Bus OBDII ECU simulator using the Teensy 4.0 module (included). Useful for testing OBDII interface and writing diagnostic software. ECU PIDs parameters adjustable via potentiometers.

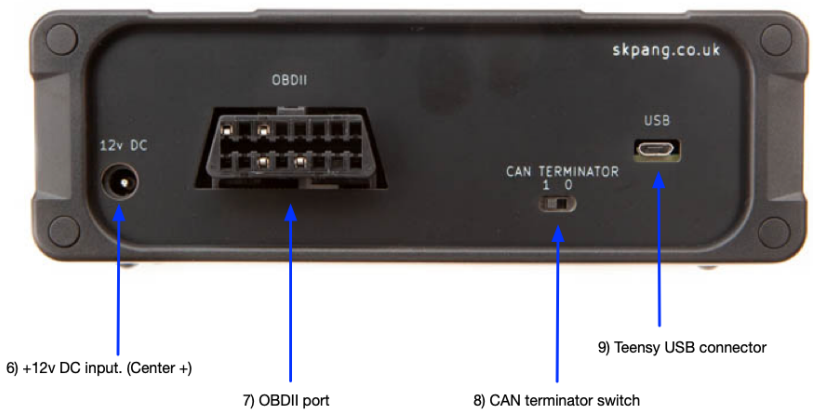
This unit requires a 12v supply. A 12v adapter is included with UK, US, EU, AU plug.

1.1. Features

- Teensy 4.0 module (installed)
- Teensy pre-programmed with OBDII ECU simulator firmware
- 500kb/s CAN speed
- Open source firmware
- Female OBDII socket with 12v supply to interface
- 6x potentiometers for PID adjustment
- 12v supply. External 12v PSU included with UK, US, EU AU plug
- Firmware Features :
    - SAE standard J1979. PIDs partially implemented. Mode 01, 02, 03
    - Adjustable PID parameters via potentiometers
    - Engine RPM
    - Throttle position
    - Vehicle speed
    - Coolant temperature
    - MAF airflow sensor
    - O2 sensor voltages
    - Setting and clearing Diagnostic Trouble Codes (DTC)

Open source firmware, other PIDs can be added.





2. Usage

1.2. Power supply

The simulator is supplied with a 12v PSU with plug for US, UK, EU, AU. Swap out plug to suit your country.

Plug the PSU into the mains and switch on. Connect the cable to the +12v DC input (6) of the simulator.

1.3. CAN Terminator

Ensure the CAN terminator switch (8) is set to 1 (enabled).

1.4. Connect Up

Connect your OBDII device to the OBDII port (7). Start your application to read the OBDII parameters.

1.5. Parameter Adjust

Use the Parameter adjustment potentiometers (1) to adjust the parameter value. You should see the ACT indicator (2) flashes when a request is being made.

1.6. MIL and DTC

Press the DTC switch (5) this will set the MIL indictor on (3). The MIL indicator can be cleared by sending a clear message in your application.

1.7. User Button

SW1 switch (4) can be used for your own function.

3. Modifying the firmware

You can modify the firmware to suit your own needs.

The Arduino IDE would need to be installed first:

https://www.arduino.cc/en/software

Then the Teensyduino add-on:

https://www.pjrc.com/teensy/teensyduino.html

The Teensy sketch:

https://github.com/skpang/Teensy40_OBDII_simulator

A list of OBDII PIDs

https://en.wikipedia.org/wiki/OBD-II_PIDs