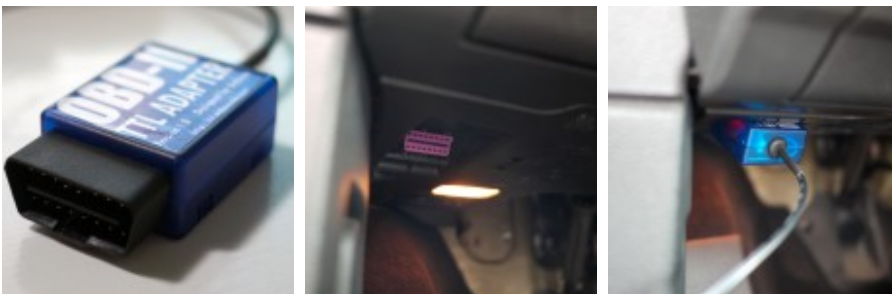


OBD-II UART Adapter for Arduino

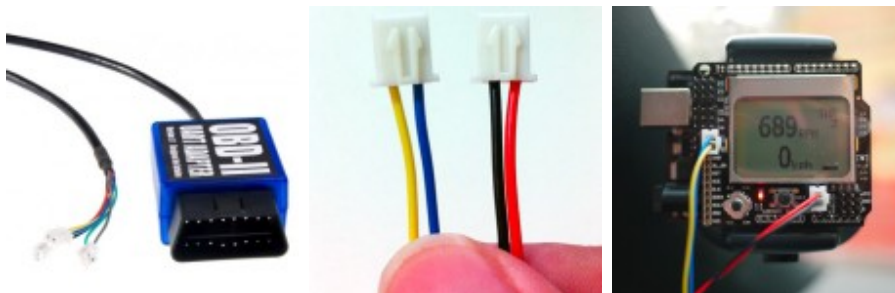


Manual:

The adapter stays plugged into the OBD port usually located under the steering column or slightly to the left of it.



A cable comes out from the adapter and splits into 2 connectors (3 connectors for model B) at the end, which are power connector (VCC/GND), OBD-II data connector (Rx/Tx) and additional I2C sensor connector (SDA/SCL) for model B. They can be connected to Arduino with onboard breakout pins or breakout shield. Your Arduino device will look tidy in car with only one connected cable.



Power Connector:

- Red: VCC (connecting to Arduino's VCC)
- Black: GND (connecting to Arduino's GND)

OBD-II Data Connector:

- White: Tx (connecting to Arduino's UART Rx / D0)
- Green: Rx (connecting to Arduino's UART Tx / D1)

Sensor Connector: (model B only)

- Grey: SDA
- Brown/White: SCL

Arduino Sample Code:

A dedicated Arduino library is developed and maintained, providing a set of easy-to-use APIs to retrieve realtime data from a vehicle.

Here is an example code of a simplest engine RPM indicator, which uses the pin 13 LED (built in every Arduino board) to indicate whether the engine is above 5000rpm.

```
#include <OBD.h>

COBD obd;

void setup()

{

    // we'll use the debug LED as output

    pinMode(13, OUTPUT);

    // start serial communication at the default adapter baudrate

    Serial.begin(OBD_SERIAL_BAUDRATE);

    // initiate OBD-II connection until success

    while (!obd.Init());

}
```

```
void loop()

{

    int value;

    // save engine RPM in variable 'value', return true on success

    if (obd.ReadSensor(PID_RPM, value)) {

        // light on LED on Arduino board when the RPM exceeds 5000

        digitalWrite(13, value > 5000 ? HIGH : LOW);

    }

}
```