RAE-SAN

ray-san@ray-san.com
www.ray-san.com

Modern Electronics For Vintage Motorcycles

V1.02 OBD_Bridge

Installation Manual

Rae-San

19/12/2019

Rae-San OBD_Bridge

Congratulations on your purchase of a new Dash setup for your motorcycle.

Your Kit should be as depicted in the pictures below-

The kit will contain a number of items

- The OBD_Bridge Assembly
- 2 x 8 way pluggable connectors for inputs
- 1 x 2 way pluggable connector for power
- 1 USB to microUSB cable for programming/setup activities.

Additionally, depending on the model ordered there may also be some of

- Piezo Pressure transducer module
- Speedo pickup hall sensor adaptor
- Capacitive Fuel Sensor module

The System

The Rae-San OBD_Bridge is designed to allow the monitoring and logging of your motorcycles(or car) important parameters over a Bluetooth connection using standad off the shelf software such as Torque Pro and RealDash

To achieve this, it monitors and measures a set of configurable inputs, processes them and makes them available over the Bluetooth connection, by masquerading as an ECU with a ELM327 OBDII interface connected.

In short – measures the inputs and looks like a normal OBD port with a reader connected. This allows the standard off the shelf software to talk and retrieve the values as it desires.

The idea is simple – in order to make it work across a whole range of bikes (or other vehicles) and handle all the different sensor input ranges and types, as well as provide all the input protection – is a little more complicated.

The Rae-San OBD_Bridge provides:

Features

- Measurement of Speedo, Engine Rpm, Fuel sensor, Voltage, Temperature.
- Support for Fuel Measurement from Injector Pulse – Once off calibration.
- Measurement of an additional quantity (oil pressure with optional sensor)
- Software adjustable gauge and variable scaling for fuel, temperature, oil pressure speed and tacho
- Non linear gauge scaling adjustment.
- Support for 4 complete different configuration sets (profiles)
- Largely interactive configuration process that provides live display of values on gauge.
- Configuration mode accessible without jumpers or opening case.
- Diagnostic output available over USB link when not if configuration mode.

- Provides OBD II Data over a Bluetooth link
  - Uses the ELM 327 protocol to look like a ELM-327 OBD interface
  - Works with Torque Software, RealDash and others
- Measures and Senses Inputs, converts and sends data.
- Input Ranges and sensor compensation.
- Configuration via Terminal on USB port.
- Software upgradable .
- Configuration in EEPROM.
- Small Size
- Pluggable connectors
- Able to be plugged in parallel with existing instruments

- Bias Generators for supplying sensors when used standalone.
- Programmable ranges with 2 breakpoint curve fitting.
- Increasing or decreasing sensors.
- 6 analog ranged inputs (nominally as below)
  - Voltage, Fuel Level, Coolant Temp, Oil Temp, Oil Pressure, Throttle Position, O2 Sense.
- 5 state inputs (nominally)
  - Fuel Low, Neutral, High Beam, Temp High, Fan On.
- Tacho + Speedo inputs.
- Odometer and tripmeter support in kmh / mph.
- Adjustable wheel circumference.

- Adjustable tacho pulse per rpm.
- Voltmeter calibration to allow for voltage drops.
- Adjustments for tank and reserve capacity

- Inputs can be reassigned if needed.
- Sample Rate 10Hz for each input.
- Selectable Filtering on Analog inputs –
    - RAW, Avg 1,3,6s, Peak hold HIGH or LOW – 3, 6 s decay.
    - Used for pulsed or PWM outputs.
- Inputs are filtered and clamped – protection.
- Field Upgradable firmware.
- 4 stored profiles of configuration settings – move back and forth between 4 bike types with one setting change.
- Profiles come preconfigured for popular bikes.

Possible Uses

- Bluetooth connected logging
- Log all the data to file for analysis later
- With TPS and O2 sensor – fuel map can be measured for injection / carb adjustment
- Log engine / performance data using the phone accelerometers.
- Bluetooth connected Dash –
- Use your phone / device as a customisable dash.

Configuration –BLUETOOTH Link establishment.

The Rae-SAN OBD_Bridge comes preconfigured with a blue tooth name of **RAE-SAN_OBD_BRIDGE**.

Your device should be set to scan for Bluetooth connections and when the Rae-San is found, can be paired. On some systems it may appear as a raw Bluetooth Address until after paring.

Pairing password is **1234**.

Once the device is paired the next step is to configure your OBD software. The example here will use Torque as this is available for many platforms. Torque Pro is necessary to be able to use custom PIDS which is required to get full access to the measurements. Torque Lite will allow access to the standard PIDS but not the custom ones.

The Torque software needs to be configured to read from the Bluetooth device paired above.

The Basic Torque Pro configuration is detailed here – but more advanced configuration is beyond the scope of this manual – please refer to the Torque Pro documentation.

An example profile is available on the Rae-San website under the support section – it will include all displays in the standard configuration and includes the custom PIDS and formulae definitions and should be a good starting point to define your own custom setup.

Torque pro images / setup

Configuration –setup

There is provided a cut out in the OBD_Bridge body to allow for access to the usb connector after assembly of the instrument.

In order to enter configuration mode – a serial terminal session **of 8,N,1 @ 57600** Baud should be opened to the OBD_Brigde within **10s econds from powerup**. If more than 10 seconds has elapsed before the connection is opened the serial terminal will receiver the diagnostic output from the OBD_Bridge – this will show the current values and when the Bluetooth connection is established to the OBD software – it will show traces of the pids received and sent.

Requirements

- ANSI serial terminal software – I use Putty generally – free and works well
- USB to MINI-USB cable
- 12V power supply for testing motor movement. – or on the bike.
- CH340 / Arduino Drivers if not recognised by your PC.

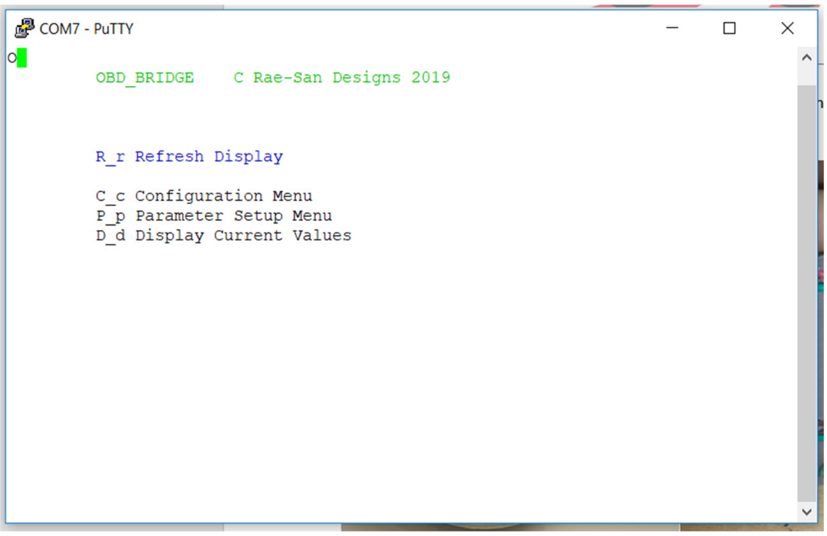
Settings

- Communication is 8 bits, No Parity, 1 Stop bit (8N1) and **NO FLOW CONTROL** (Off).
- The port will depend on your computer – so plug it in and check what port appears.
- **Configuration mode is entered by holding down button A and turning on the power – the Gauge reads the button states at power on and enters into configuration mode.**

Open a serial terminal session to the meter port -

You should be presented with the following screen – if you are greeted with a flashing "o" in the top left corner – hit the R key to "refresh" this will re-send the screen.

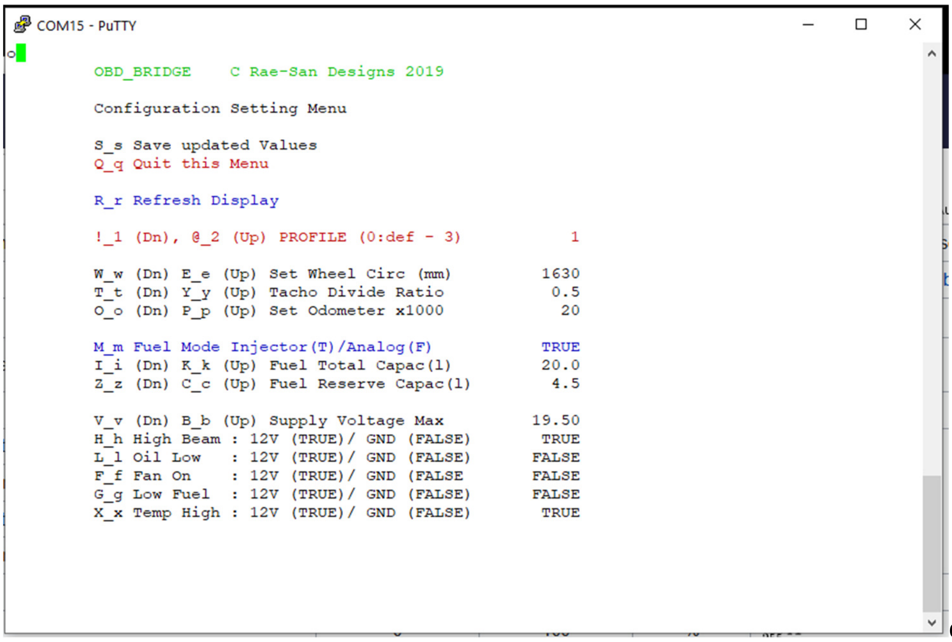
You should then see –



The menus are text driven and the Keys to use are at the start of each line – eg

- R or r will refresh the display.
- C or c will enter the Configuration Menu
- P or p will enter the Parameter Menu.

Press “c” to enter the configuration menu.



qp

Once Again

- R or r for refresh
- Q or q to quit this menu back to the previous – note this does not save any changes
- S or s to save any changes to the EEPROM memory

Now the specific items –

The Key fields are shown as X_x, Y_y : the shifted version of the key is used for a big step, the lowercase for a single step up or down. The Shifted step is normally 10x the single step.

- **!_1 or @_2** : Select / change the profile number that is being edited and used. The profile that was selected when last saved is what becomes the default power up profile.
  - This allows for 4 completely different Gauge setups to be stored in the gauge in the case that you might move them between bike models or wish to experiment.
  - This relates to the values on the Parameters screen as well as those on the configuration screen.
- **W_w or E_e :** Increment or decrement the Wheel Circumference.
  - Note – this is in **mm** as this gives more accuracy in an easier to adjust format.
  - Note that this represents the ground distance travelled for one complete pulse from the speed sensor. This can widely vary from 1 pulse per revolution of the wheel – to many pulses per revolution. See the appendix for known or example values.
- **T_t or Y_y :** Increment or decrement the tacho pulse to RPM divider. This sets the ratio of pulses to the engine RPM. Usually this will be **1.0** for a wasted spark system. Values 0.5 to 3.0 are available
  - Wasted Spark – coil sense – 1.0
  - Non wasted spark – single coil sense- 0.5
  - Camshaft RPM sensor – 0.5 (as at ½ engine RPM)
- **O_o or P_p:** Set Odometer reading: Set the starting Odometer reading to the nearest 1000 km or miles.
  - Note this can only be set the first time – once the bike is operated the increased value is stored and any changes made in the configuration will be ignored.
  - Note – there is a reset procedure to override the odometer and allow it to be reset if you change the meter between bikes.

    - Set the odometer x1000 to 100.
    - Save the setting.
    - Power off the meter.
    - Power the meter while holding down Button A to enter configuration mode.
    - Connect the serial terminal and set the odometer X1000 to 0
    - Save the setting and Power Down.
    - Power the meter while holding down Button A to enter configuration mode.
    - Connect the serial terminal.
    - Configure to the value desired and save.

- **M_m :** Fuel Injector or Analog mode. Injector (TRUE). Analog (FALSE).

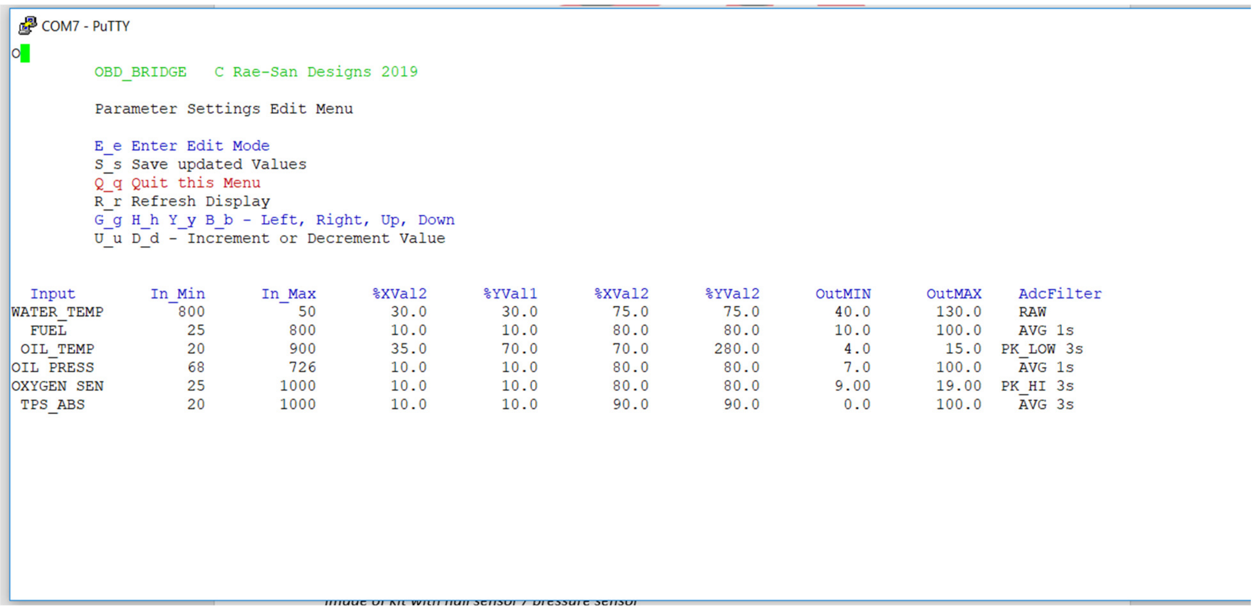
- **I_i or K_k:** Fuel Tank Total Capacity – sets of 0.1 l
- **Z_z or C_c:** Fuel Tank Reserve Capacity – steps of 0.1 l - sets warning level on Gauge

- **H_h :** High Beam Polarity Sense. Active High (TRUE). ACTIVE LOW (FALSE).
  Normally High beam on when +12 **-> TRUE.**
- **L_l :** Low Oil Pressure Polarity Sense. Active High (TRUE). ACTIVE LOW (FALSE).
  Normally GND for Oil pressure low **-> FALSE**.
- **F_f :** Fan On Polarity Sense. Active High (TRUE). ACTIVE LOW (FALSE).
  Thermo Switch goes LOW to turn fan on **-> FALSE**.
- **G_g :** Low Fuel / Neutral Polarity Sense. Active High (TRUE). ACTIVE LOW (FALSE).
  Normally GND for NEUTRAL **- > FALSE.**
- **X_x :** High Temp Polarity Sense. Active High (TRUE). ACTIVE LOW (FALSE).
  Dependant on setup.

- **V_v or B_b :** Set the voltage divider value. This is normally about 19.5 volts to give the correct reading – it is a calibration that allows for variation in the internal devices and allows for compensation of the bikes wiring losses. The eaiest way to set it – is to measure the supply voltage with a multimeter , have the Gauge set to display a voltage (via the display selection earlier) and then adjust the value so as to display the correct voltage. Calibration done.

Parameters – Setup

Next from the top level menu – press the P button to enter the parameter configuration page.



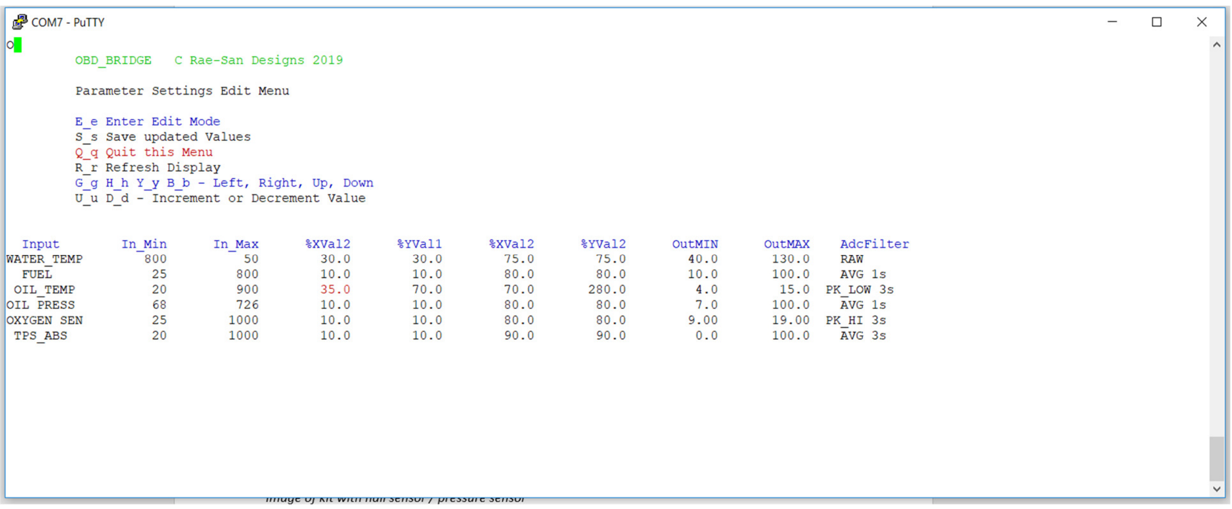
Once again there are the standard keys

- S and s for saving the changes
- Q and q to exit the menu back to the top level
- R and r to refresh the display

Now there are some extra keys –

- E or e E – Enter edit mode – this allows changing the values in the table.
    - The value currently selected for edit is shown in **RED.**
- G or g (move Left), H or h (move Right), Y or y(move Up), B or b(move Down). These allow moving around the selection of the value to be edited.
- Note the key layout is like a arrow keypad but with letters.
- **U**(+10) **u**(+1) or **D**(-10) or **d**(-1) to increment or decrement the selected values.

Once you enter edit mode the screen should appear as below.



Here you can see that the %XVal2 of the OIL_TEMP entry is selected.

Once changes are made S should be used to save them to the EEPROM. This will exit from Edit mode and the RED selection will disappear.

Parameters – meanings

The previous section dealt with the navigation of setting the parameter values – but now we need to look at what the parameters mean and control.

There are 6 sets of parameters for Coolant Temperature, Fuel measurement, Oil Temperature, Oil Pressure, Oxygen Sense, Throttle Position Sense.

These parameters are used to adjust for the differences in sensors and importantly for the differences in gauge faces – and adjust for any non-linear scales.

Input

This is just the input type and is not changeable.

WATER_TEMP:
FUEL:
OIL_TEMP:
OIL_PRESSURE:
OXYGEN_SENSE:
TPS_ABS:

In_Min

Adc input value (0 – 1023) at the **minimum** for the measured quantity. Note for a a decreasing sensor such as temperature or often fuel – this will be higher than the max value below – the MIN and MAX refer to the quantity being measured.

In_Max

Adc input value (0 – 1023) at the **maximum** for the measured quantity. Not for a a decreasing sensor such as temperature or often fuel – this will be lower than the max value above – the MIN and MAX refer to the quantity being measured.

OUTMin

These values are for the Output MIN. These values need to relate to the real quantity measured or the PID being sent.

WATER_TEMP: Typically output in degrees - 40 to 130 typical.
FUEL: Output value is a percetnate – 0 to 100%
OIL_TEMP: Output in Degrees – typically 40 to 150 Degrees.
OIL_PRESSURE: Output value – defn say in KPa but PSI is also possible dependant upon the custom PID definition I use PSI – 7- 100.
OXYGEN_SENSE: Output value could be lambda – but I use AFR as makes the custom PID easier to define – 9.0 – 19.0. Will deped what optional O2 sensor is used and how its output is defined
TPS_ABS: Depends on the input from the TPS. Usually would be set up as a resistive sensor with an range of near 0 to near 5V – output mapped 0 – 100% usually.

OUTMax

These values are for the Output MAX These values need to relate to the real quantity measured or the PID being sent.

As per the above ranges. Max values

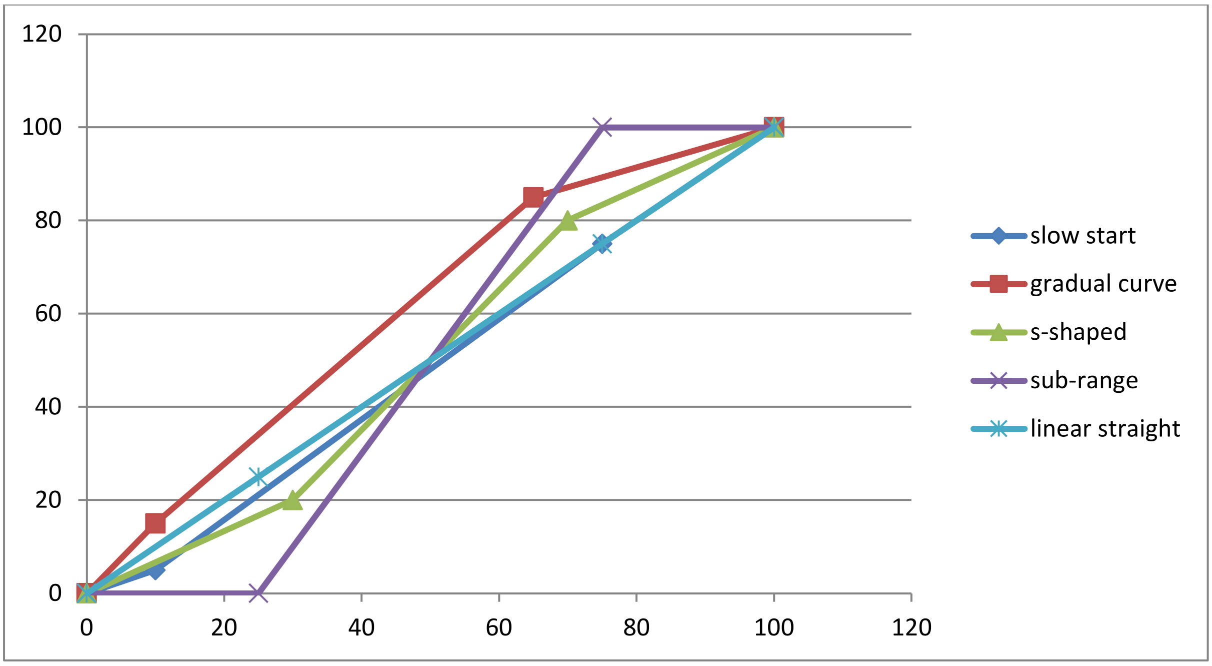
%X1,%YVal1; %X2,%YVal2

These parameters control the linearity of the scale between minimum and maximum. They map the input reading % to an output reading % that is displayed on the meter scale.

The start point is always 0%,0% and the final point is always 100%,100%. In between are 2 movable points that control the curve used. The chart below shows a few examples. The % values use the defined In_Min and In_Max to set the 0% to 100% range.

Examples:

	%x1	**%y1**	**%x2**	**%y2**
slow start	10	5	75	75
gradual curve	10	15	65	85
s-shaped	30	20	70	80
sub-range	25	0	75	100
linear straight	25	25	75	75



Generally the curves will be linear or a slow start type of curve as often the speedo or tacho has a region at the bottom where it doesn't respond – eg tacho below 500rpm and speedo below 10km/h.

By setting the points appropriately the electronics will map to the markings to correctly display the value.

The sub range curve is useful to map a sensor - fuel or temperature that has a narrower range of variation to the full meter range.

Known configurations can be found in the appendix.

As a starting point – just setting the max angle and maximum values and leaving the defaults for the rest should be quite close to correct.

When setting these points – the gauge needle or the grap[hic meter will display the Y (output %) value as it is adjusted.

AdcFilter

This is a setting that controls how the measured value is processed. For the most part the sensors will be set to RAW or FILTERS 1s or FILTER 3. The Settings and behaviours are.

RAW : No filtering or averaging is used – the latest measurement goes straight to the displays.
AVG 1s : The input value is averaged over a rolling 1s (10 samples)and the result goes to the displays.
AVG 3s : The input value is averaged over a rolling 3s (30 samples)and the result goes to the displays.
AVG 6s : The input value is averaged over a rolling 6s (60 samples)and the result goes to the displays.
PK_LOW_3s : The samples are passed through a fast attack slow decay filter to capture the lowest value hold it with a slow decay: This is used to capture a pulsed sensor value where the minimum value needs to be measured - such as a temperature sensor.
PK_LOW_6s : Same as above but with a 6s decay time.
PK_HIGH_3s : The samples are passed through a fast attack slow decay filter to capture the higest value hold it with a slow decay: This is used to capture a pulsed sensor value where the maximum value needs to be measured .
PK_HIGH_6s : Same as above but with a 6s decay time.

The value can be set to any of the above types as appropriate. The Average 1s should be suitable for most analogue sensors. The other options are used when dealing with pulsed sensors or PWM outputs from ECUs.

Example Calculation of Settings

Fuel Sensor Example

Ok - lets look at a bit trickier example. Fuel sensors are notoriously inaccurate - as the tank has a wierd shape usually - We can actually use the corrections to make the fuel gauge read more accurately.

Theres no needle to worry about for the fuel and temperature sensors so the angles don’t mean anything - but the correction values do.

So lets assume we take the tank off the bike when its empty - and then hook a multimeter to the sensor to measure the resistance.

We then start with an empty tank - :

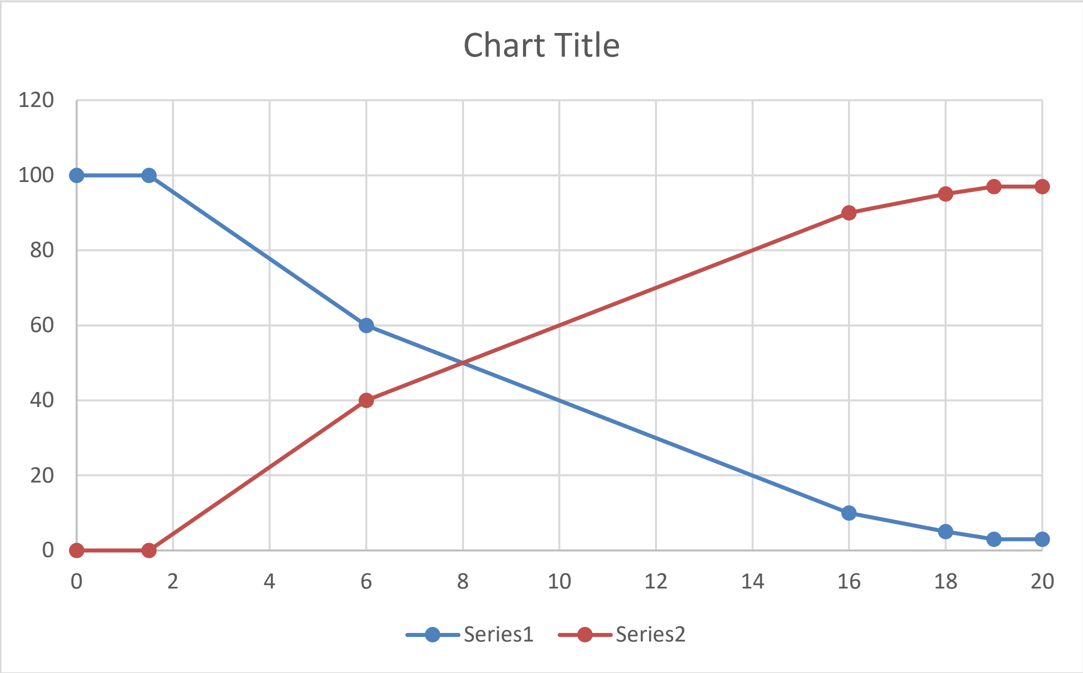
- with volume empty - we measure the resistance as 100 ohm
- The reading starts to change once we’ve added 1.5l
- When the level gets mostly over the tunnel in the tank - its say 6l and 60ohm
- When its getting near full and the tank starts to narrow - its 16l and 10 ohms
- When we get to 18l the resistance and is 5 ohms
- When it gets to 19l the resistance stops changing and is 3 ohms - until full at 20l

The table below shows the measumentd and the converted percentages

Volume	**Ohms**	**100-ohms**	**%(100-ohm)**	**%full**
0	100	0	0	0
1.5	100	0	0	7.5
6	60	40	40	30
16	10	90	90	80
18	5	95	95	90
19	3	97	97	95
20	3	97	97	100

Below shows the measurements and the 100 ohms (max R) - value which represents the fuel value.

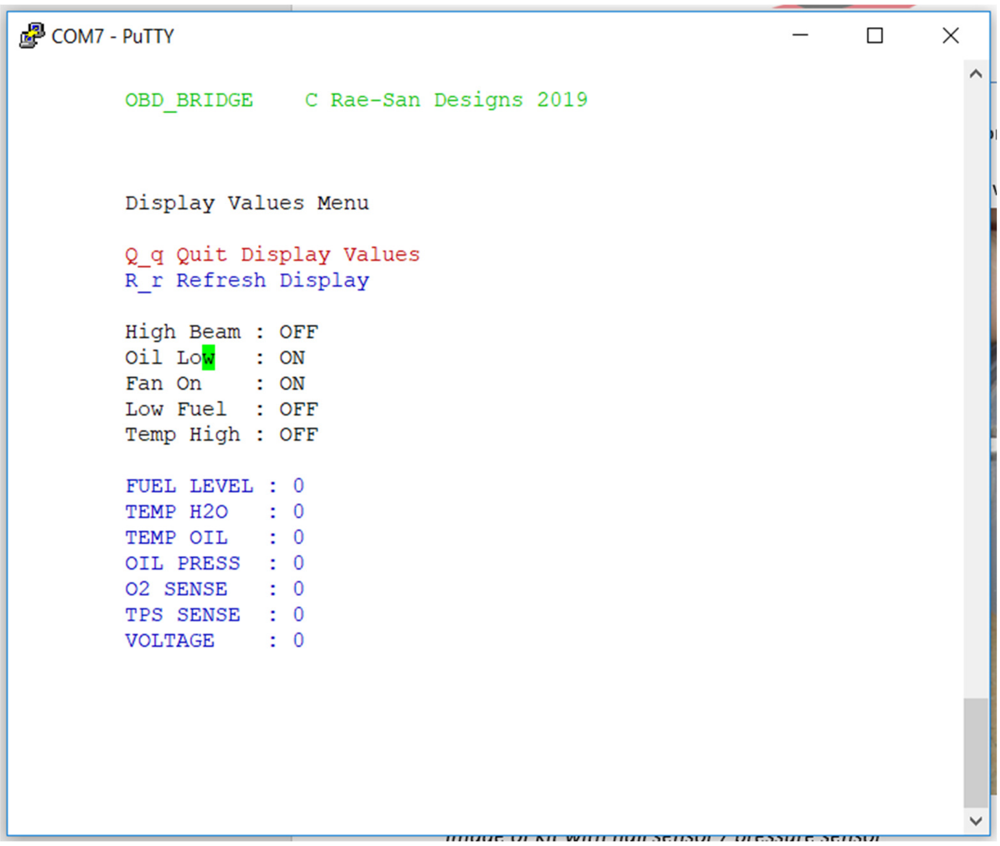
As can be seen its not perfectly straight. We can remedy some of this . we cant do anything about the less than 1.5 l or over 18 litres (empty and full) as the sensor doesn't respond in these regions, but we can straighten out the rest.



Temp Sensor Example

Display Values

This selection displays the current measured values and states. This allows you to check the current settings while you vary the inputs on the bike.



OBD Bridge PIDs

The ODB_Bridge uses standard PIDS (messages) to the software where possible but some of the quantities are not in the standard PID set, so some custom PIDS are used. All used data PIDS are listed in the table below. This should give you enough information to set up any custom quantities required.

PID (hex)	PID (Dec)	Data bytes returned	Description	Min value	Max value	Units	Formula[a]
05	5	1	Engine coolant temperature	-40	215	°C	A-40
0C	12	2	Engine RPM	0	16,383.75	rpm	(256A+B)/4
0D	13	1	Vehicle speed	0	255	km/h	A
11	17	1	Throttle position	0	100	%	(100/256)A
2F	47	1	Fuel Tank Level Input	0	100	%	(100/256)A
42	66	2	Control Module Voltage	0	21	V	(256A + B)/100
5C	92	1	Engine oil temperature	-40	210	°C	A-40
9C	156	2	O2 sensor Data	9.0	19.0	AFR	(256A + B) / 100
A6	166	4	Odometer	0	429,496,729	km	(D+256C+256^2B+256^3A)/10 Km. In 0.1km units
E5		2	Fule Usage Rate	0		l/hr	(256A +B)/20
B0	176	2	Oil Pressure	0	200	psi	(256A +B) / 10
B1	177	1	High Beam	0	1	Boolean	0 Off , 1 On
B2	178	1	Oil Warn	0	1	Boolean	0 Off , 1 On
B3	179	1	Fan On	0	1	Boolean	0 Off , 1 On
B4	180	1	Low Fuel	0	1	Boolean	0 Off , 1 On
B5	181	1	Over Temp	0	1	Boolean	0 Off , 1 On
B6	182	2	Distance To Empty	0	65536	Km	(256A +B)/10
B7	183	2	Avg Fuel Economy	0	65536	l/100km	(256A +B)/10

The custom PID definitions in the setup I use are as follows.
As the output ranges defined for the sensors affects the value reported and hence the PID values sent, these are shown in the other columns of the table to show the interdependence.

Standard PIDS

Standard Pids – already defined	
Custom Pids – need to be defined	

PID	Quantity	OutMin	OutMax	unit	Internal range	Suggested Formula	Min	Max
0105	Coolant TEMP	40.0	130.0	Deg C	0-3000	A - 40	-40	215
012F	FUEL	10.0	100.0	%	0-1000	(100 *A)/255	0	100
015C	OIL TEMP	40.0	150.0	Deg C	0-3000	A - 40	40	150
01B0	OIL PRESSURE	7.0	100.0	PSI	0-2000	Int16(a:b)/10 (a*256+b)/10	0.0	110.0
0111	TPS	0.0	100.0	%	0-1000	(100 *A)/255	0	100
019C	O2 SENSE	9.0	19.0	A/F	900-1900	Int16(a:b)/100 (a*256+b)/100	8.0	20.0
0142	Voltage	0.0	21.0	V	0..21000	Int16(a:b)/1000	0.0	21.0
010D	SPEED	0.0	255	Km/h	0-255	Int8(a)	0.0	255.0
010C	TACHO	0.0	16384	RPM	0-1000 (eg)	Int16(a:b)/4	0	16000
015E	Engine Fuel Rate	0.0	65536	l/hr	0-65.536	Int16(a:b)/20	0	65.5
01B3	Cooling Fan On	0	1	-	0..1	A	0.0	1.0
01B1	High Beam	0	1	-	0..1	A	0.0	1.0
01B4	Low Fuel Warn	0	1	-	0..1	A	0.0	1.0
01A6	Odometer	0.0	100000000	10cm	Int32	Int32(a:b:c:d)/100	0.0	100000.0

01B2	Oil Pressure Warn	0	1	-	0..1	A	0.0	1.0
01B5	Over Temp	0	1	-	0..1	A	0.0	1.0
01B6	Distance To Empty	0	65536	km	6553.6	Int16(a:b)/10	0.0	6553.6
01B7	Avg Fuel Economy	0	65536	l/100km	3276.8	Int16(a:b)/10	0.0	3276.8

Wiring the OBD_Bridge.

Shown below is a diagrammatic view from the rear of the gauge to indicate the wires and their colours.
Typcial colours indicated.

2 PIN Power Connector (TOP)

Wire Position	**Wire**	**Wire Function**
RIGHT SIDE	POWER - RED	+12V power in – switched power from the bike
LEFT SIDE	POWER_GRND -GREEN	GROUND connection – POWER GROUND for gauge and backlights.

ANALOGUE INPUT CONNECTOR (LEFT SIDE)

Connector Position	**Wire Colour**	**Wire Function**
Not Connected	POWER - RED	+12V power in – switched power from the bike
Fuel Sense Input	YELLOW / WHITE	
Coolant Temp	GREEN / BLUE	
Oil Temp	Not Standard	
Oil Pressure	Not Standard	
Oxygen Sense	Not Standard	
Throttle Posn Sense	Not Standard	
GROUND	GREEN	

STATE INPUT CONNECTOR (RIGHT SIDE)

Connector Position	**Wire Colour**	**Wire Function**
TACHO Input	YELLOW	+12V power in – switched power from the bike
SPEEDO Input	WHITE / BLACK ??	
High Beam Sense	BLUE	High Beam Indication
Oil Low Pressure Sense	BLIE/RED BLACK / BROWN	Oil Warning Lamp indication
Fan On Sense	varies	Can be used for other function.
Fuel Low Sense / Neutral sense	LIGHT GREEN/RED	Fuel Low Sense not often present if reserve position on tank – Can be used for Neutral or other indicator FUNCTIONS AS FUEL PROGRAMMING SWITCH IN INJECTOR MODE
Temp High Sense	Varies if present	Can be used for Neutral or Other

GROUND		

Jumper Settings

The jumpers are used to select whether the ODB_Bridge is to provide a bias current of 33mA or not.
If the gauges and sensors are fed by external instruments then biasing of the sensors is not required.
If the OBD_Bridge is used without any existing gauges then it is likely the biasing is needed if there is no other power going to the sensors.

If No biasing is required the jumpers should all be in the off position as this will mimimise the current drawn by the OBD_Bridge. If the ODB_Bridge is set to biasing there will be a current draw of 6x33mA or Approximately 0.2 of an Amp more than if the biasing is all off.

If no Bias is needed then all jumpers should be removed to mimimize current draw. If one sensor or more needs biasing then all jumpers should be fitted in either the BIAS or SHUNT TO GROUND position as appropriate.

Bias Positions

Not Fitted

BIAS to Pin

BIAS SHUNT TO GROUND.

Jumper	
1	Bias selection for Fuel Sensor
2	Bias selection for Coolant Temp Sensor
3	Bias selection for Oil Temp Sensor
4	Bias selection for Oil Pressure Sensor
5	Bias selection for O2 Sensor
6	Bias selection for Fuel Throttle Position Sensor

Standard Bike Sensors

Fuel - Analog

Most older bikes, if equipped with a fuel sensor, use a 100 ohm wirewound reisistor with a float on an arm. Depending on the shape of the tank – they can give a reasonable estimation of the fuel in the tank – usually the rider learns things like “the gauge reads 1/3 full when the tank is ½ full, and I need to fill up at a certain level.

For these types of sender – Averaging of 1s is fine and the levels can be easily set with an empty and full fuel tanks. Intermediate points can be easily set if desired – to give a more accurate tracking of the actual fuel level using the correction curve. Usually 4-10Ohms Full, 95-100 Ohms Empty.

For Fuel injected bikes with an ECU – they can use a resistive sensor as above, or they may have some other style of sensor and then output a processed signal to drive a fuel gauge. In these cases the slow average of 6s is most likely the best choice.

The VF750 Magna (and some others most likely) uses a scheme where the measurement is only pulsed every few seconds – and then it returns to a 12V reading – in this case it is necessary to capture the low point of the pulse and save this as the value – so in this case the PK_LOW_6S is the option of choice.

If using the bias from the OBD-Bridge then a 1s Average value is fine as there is not pulsing to deal with.

level	EMPTY	1/3	2/3	Full
Resistance	97	68	37	8
Comment	Empty meter reading	1/3 full	2/3 fiull	FULL
Voltage when biased by OBD_Bridge	3.2V	2.25v	1.22	0.26v
Voltage counts biased by OBD_Bridge	512	360	195	41
Voltage biased by Meter				
Counts Biased by meter				

Fuel – Injection

If set for the fuel injection measurement mode – the system measures the fuel flow from the total injector opening time. This is largely a cilbrate and forget system apart from needing to be reset when the tank is refilled – as it needs to then count the usage from a full tank.

The single button input for the Low Fuel alarm is used to control the fuel calibration.

- Holding the button **>2.5s** and **<10s,** in normal mode – resets to a full tank but does not recalibrate.
- Holding the button in > 10 seconds – enters Calibration mode
    - In calibration mode the fuel amount is shown on the tacho display at 100 RPM / litre – so 1650r rpm => 16.5 litres
- In calibration mode – a short press **(<2.5s**) increments the amount by 0.1 litre
- In calibration mode – a long press **(> 2.5s , < 10s**) increments by 1.0 litre
- When max (25 litre is reached a short press will reset to minimum)
- Holding the Button in **> 10** secs will then exit calibration mode, calculating the calibration and storing the values.

So the simplest Calibration procedure is

- Fill the tank to full
- Press button for **>2.5 < 10s** to reset to zero the counters.
- Ride the bike normally until you hit reserve or say 200km
- Fill the tank and note the amount of fuel put in the tank.

- With power on and dash connected but engine not running – hold in the button **> 10s**
- The Tach display will show the default fill amount of Capacity – reserve litres as detailed above.
- Adjust the amount displayed to match the amount of fuel added to the tank – eg 14.2 l => 1420 RPM.
- Hold the button **> 10s** calculate the calibration and store the values
- Tach will return to zero to indicate exiting Calibration mode
- Done.

Normal fill procedure –

- Fill tank.
- Power on, hold button **>2.5s < 10s**
- Fuel indication will reset to full.
- Done

Coolant Temp

For most bikes this is a thermistor with a negative temperature coefficient. That is, as the temperature increases the resistance decreases. This is very similar to the Fuel sensor above – except that the maximum resistance becomes higher.

Usually the resistance values are listed in the service manual for the bike and these form a good basis for the measurement values.

A typical example from a honda sabre manual is

Temp	60 Deg C	85 Deg C	110 Deg C	120 Deg C
Resisatance	104 Ohm	44 Ohm	20.3 Ohm	16.1 Ohm
Comment	Beginning– low meter reading	Normal Operating point	Upper Limit normal operatin – Start Overheat	Max Overheat point
Voltage when biased by OBD_Bridge	3.4v	1.45v	0.67v	0.53v
Voltage counts biased by OBD_Bridge	544	232	107	91
Voltage biased by Meter				
Counts Biased by meter				

This provides a first guess as to the values to be used for most sensors.

Speedo

The speedo input measures pulses from a hall sensor or other pulse sensor (usually on the front wheel). The pulse amplitude needs to be between at least 0 and 5V, 0 and 12V is fine as the input is protected. The system increment the disatance travelled and as a result the calculated speed on each pulse.

Tacho

The Tacho input is designed to work with 12C -> Ov (or vices versa) pulses consistent with the trigger pulses used to drive 12V ignition coils. The system measures the interval between pulses – so dwell of the pulse does not affect the reading.

For a CDI system – either a 12V compatible output is needed or a CDI tacho adaptor to convert the CDI pulse to a 12V compatible pulse should be used.

Oil Pressure

On most bikes the Oil pressure indication is a simple switch that opens when the oil pressure reaches a certain threshold – usually between 4 and 7 psi on most bikes. This is interfaced most directly to the oil pressure state sensor line.

This should be set to – Polarity Normal (true).

Some bikes (notably some Yamahas – have a low oil level light rather than a low oil pressure (FZR)) but the sensing is the same.

If your bike has an oil pressure sender that actually indicates pressure rather than being a simple switch then this will normally operate like a resistive sensor between 0 and 100 ohms – If it is connected to an existing oil pressure display – you will need to work out the values by measurement.

If it is biased from the OBD_Bridge – then it should be a reading of approximately 5.9 adc counts per ohm of resistance.

The other option is to fit an electronic Piezo sensor – See the Rae-San adaptor.

Tail/ Head Light

Some bikes are equipped with sensors that report on the status of the headlight or taillight bulbs. Usually these are a 12V / OV signal that can be simply interfaces on the state monitoring inputs if desired.

SideStand

Some bikes are equipped with an interlock on the sidestand . Usually these are a 12V / OV signal that can be simply interfaces on the state monitoring inputs if desired.

Throttle Position Sensor

Many of the bikes with fuel injection use a Throttle position sensor to feed the ECU. These ae usually a resistor based sensor with an output voltage between 0 and 5V that feeds into the ECU. These should be able to be directly interfaced to the OBD_Bridge.

For older non-fuel injected bikes – it is possible to rig up a throttle position sensor to input position to the OBD_bridge. This is only really worth the effort if an O2 meeter is also fitted so that the fuelling can be analysed in detail.

This can be used by the OBD_Bridge

Rae-San Additional Sensors

Oil Pressure Sensor

The Rae-San Oil pressure sensor is a piezo (solid state) pressure sensor with an adaptor board to provide 5V power to the sensor. It also provides a switched output that replicates the original pressure switch that drives the normal dash light. This means the new sensor can be fitted in place of the original sensor with no loss in functionality as the existing dash light will work as before – but you also get an accurate oil pressure reading in PSI (or kPa should you desire)

Speedo Sensor

Generally any hall / magnetic based sensor setup will work fine – whether it be from a push bike, any of the kits or from some other meter. I may make a kit available if there is enough demand, but there are plenty of options already out there.

Capacitive Fuel Sensor

Under development – this is a kit sensor that senses the petrol level by measuring the capacitance of two elements – a reference element and a main element. The two elements are incorporated into the one assembly. The ratio of the capacitance of the two elements is a function of the height of the petrol. Some smarts are involvd to compensate for temperature and in particular the ethanol in the fuel. Currently in bench test.

Other Aftermarket Sensors

Oxygen / AFR meter Sensors

I have a TechEdge 2Y DIY WbO2 sensore – other sensors can be made to work as well. Most WbO2 sensors have a programmable range output so 0-5V is quite easily set up. The OBD_Bridge can alternatively be set up to match the range output by the Wb02 sensor as required.

Default Configuration

When shipped the gauge will have data loaded in the 4 profiles

Configuration

Type	Profile #	Wheel Circ	Tach Div	Odo	Volt Max	High Beam	Oil Low	Fan On	Low Fuel	Temp High
Value	0	470	1.0	000000	19.5					
	1									
	2									
	3									

Parameters

Profile 0

Type	In_Min	In_Max	%Xval1	%YVal1	%Xval2	%YVal2	OutMin	OutMax	AdcFilter
WATER TEMP									
FUEL LEVEL									
OIL TEMP									
OIL PRESSURE									
O2 SENSE									
TPS_ABS									

Profile 1

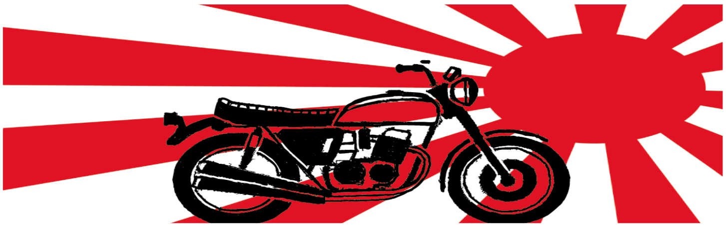
Type	In_Min	In_Max	%Xval1	%YVal1	%Xval2	%YVal2	OutMin	OutMax	AdcFilter
WATER TEMP									
FUEL LEVEL									
OIL TEMP									
OIL PRESSURE									
O2 SENSE									
TPS_ABS									

Profile 2

Type	In_Min	In_Max	%Xval1	%YVal1	%Xval2	%YVal2	OutMin	OutMax	AdcFilter
WATER TEMP									
FUEL LEVEL									
OIL TEMP									
OIL PRESSURE									
O2 SENSE									
TPS_ABS									

Profile 3

Type	In_Min	In_Max	%Xval1	%YVal1	%Xval2	%YVal2	OutMin	OutMax	AdcFilter
WATER TEMP									
FUEL LEVEL									
OIL TEMP									
OIL PRESSURE									
O2 SENSE									
TPS_ABS									



MOTORCYCLE PRODUCT DISCLAIMER

Motorcycling is an inherently dangerous activity, which may result in personal injury and / or death. While Rae-San (and/or Entities Trading as Rae-San) products are designed to offer superior riding comfort and performance, no product can offer complete protection from injury or damage to individuals and property in case of fall, loss of control or otherwise. You should wear appropriate protective equipment, obey all traffic laws, regulations and use good judgment. Never take alcohol or drugs (legal or illegal) before riding. Always inspect your motorcycle before riding, and read your owners manual. You should be familiar with the wide range of foreseeable hazards of motorcycling and decide whether to assume the risks inherent in such activity, which include damage to property, serious injury to you or others and death.

RAE-SAN (AND/OR ENTITIES TRADING AS RAE-SAN) DISCLAIMS ANY RESPONSIBILITY FOR INJURIES OR DAMAGES INCURED WHILE USING ANY OF ITS PRODUCTS. RAE-SAN (AND/OR ENTITIES TRADING AS RAE-SAN) MAKES NO GUARANTEES OR REPRESENTATIONS, EXPRESSED OR IMPLIED, REGARDING MATERIALS AND WORKMANSHIP OR THE FITNESS OF ITS PRODUCTS FOR ANY PARTICULAR PURPOSE, FURTHER RAE-SAN (AND/OR ENTITIES TRADING AS RAE-SAN) MAKES NO GUARANTEES OR REPRESENTATIONS, EXPRESSED OR IMPLIED, REGARDING THE EXTENT TO WHICH ITS PRODUCTS PROTECT INDIVIDUALS OR PROPERTY FROM INJURY OR DEATH OR DAMAGE. RAE-SAN (AND/OR ENTITIES TRADING AS RAE-SAN) PRODUCTS MUST BE INSTALLED AND REPAIRED BY A TRAINED PROFESSIONAL. RAE-SAN (AND/OR ENTITIES TRADING AS RAE-SAN) DISCLAIMS ANY RESPONSIBILITY FOR INJURIES, DEATH OR DAMAGES DUE TO IMPROPER INSTALLATION OF ITS PRODUCTS. ANY ACTUAL ATTORNEY FEES FOR ANY UNSUCCESSFUL CLAIM AGAINST SELLER TO BE PAID BY PLAINTIFF/COUNTER-CLAIMENT.

Modification of a motorcycle by fitting the devices obtained from Rae-San (and/or entities trading as Rae-San) may void manufacturer’s warranty, state and federal laws and render insurance void. It is the responsibility of the motorcycle owner to be familiar with these laws. Fitting of any product from Rae-San (and/or entities trading as Rae-San) constitutes acceptance of the responsibility and consequences of any legal, insurance, or warranty breach by the motorcycle owner. Furthermore fitting of any device constitutes acceptance and agreement to indemnify Rae-San (and/or entities trading as Rae-San) of all damages and liabilities that may arise in the event of accident, misfortune or failure of a device.

12 MONTH LIMITED WARRANTY AND LIABILITY DISCLAIMER

Rae-San (and/or Entities Trading as Rae-San) warrants this product free from defects in material or workmanship for a period of 12 months from the date of purchase, except as noted. The product warranty does not cover damage caused by misuse or abuse; accident; the attachment of any unauthorized accessory; alteration to the product, improper installation, exposure to harsh weather and/or chemicals, lack of proper maintenance, lubrication, or any other conditions that are beyond the control of Rae-San (and/or Entities Trading as Rae-San). Rae-San (and/or Entities Trading as Rae-San) shall not be responsible for any type of incidental, consequential, or special damages. All implied warranties, including but not limited to those warranties of fitness and merchant accountability, are limited in the total duration to 12 months from the original purchase date. If the warranty occurs within 30 days from the date of purchase, please contact the dealer the product was purchased from. Over 30 days from the date of purchase, please contact Rae-San (and/or Entities Trading as Rae-San) direct at Rae-San@Rae-San.com. Customer must contact Rae-San (and/or Entities Trading as Rae-San) prior to sending warranty product. Claims for lost shipment, damaged shipments, or other problems regarding freight must be made directly to the responsible carrier. Warranty is solely through Rae-San (and/or Entities Trading as Rae-San). Service to our products by anyone other than Rae-San (and/or Entities Trading as Rae-San) authorized installation centers voids warranty. Rae-San (and/or Entities Trading as Rae-San) disclaims any and all liability for consequential or incidental damages. Rae-San (and/or Entities Trading as Rae-San) will not be held responsible for personal damage, bodily harm or any other legal matters.