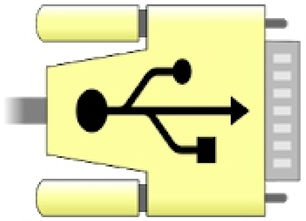
Preparation

1 >> Download “Serial USB Terminal” via your default App Store



Serial USB Terminal

Kai Morich Tools

3+

Offers in-app purchases

This app is compatible with all of your devices.

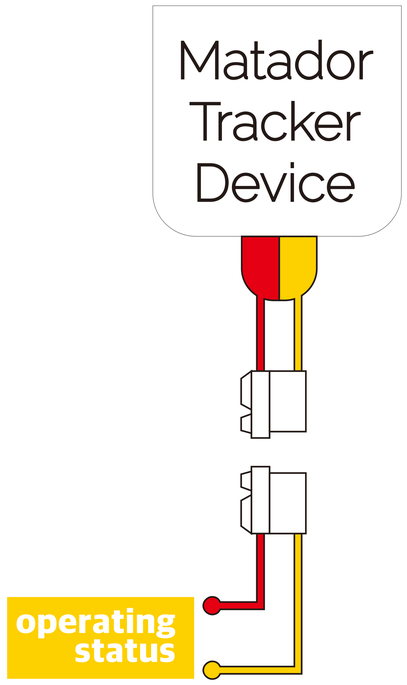
s.id/serialmonitor

2>> Prepare a OTG Connector and micro USB Cable

3>> Connect Device to your phone with OTG

Operating Status

>> Hardware Assembly <<



>> Digital Mode Settings <<

command :

```
setDigitalOperatingStatus(<address>,<mode>,<threshold>)
```

example :

```
setDigitalOperatingStatus(0,1,500)
```

>> Default Config <<

```
mode: 0
```

```
threshold : 400
```

```
address : operating status is in 0
```

>> info <<

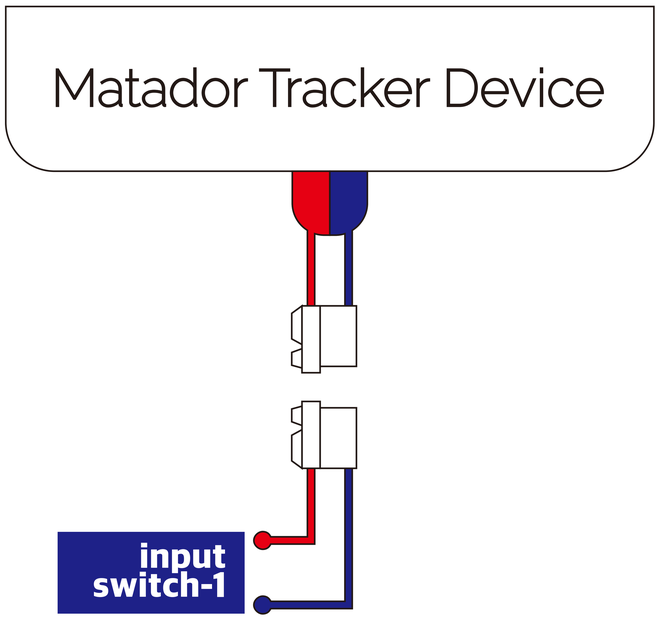
```
mode : input sensor has 2 mode, first is normal logic second is
reverse logic. For generally, input sensor works using threshold
to decide output logic. At the normal logic, while the input
sensor below the threshold, output logic will be set 0
or LOW, and set 1 or HIGH while above it. For case of
reverse logic. Its totally reverse of normal logic, the output
value will be set 1 if input sensor value below the threshold
and set 0 while above it.
```

```
threshold : value limit where the output changes(range is 0-1023)
```

Input Sw - 1

```
address : this part, value must be `1`
```

>> Hardware Assembly <<



>> Digital Mode Settings <<

command :

```
setDigitalOperatingStatus(<address>,<mode>,<threshold>)
```

example :

```
setDigitalOperatingStatus(1,1,500)
```

>> Default Config <<

```
mode: 0
```

```
threshold : 400
```

>> info <<

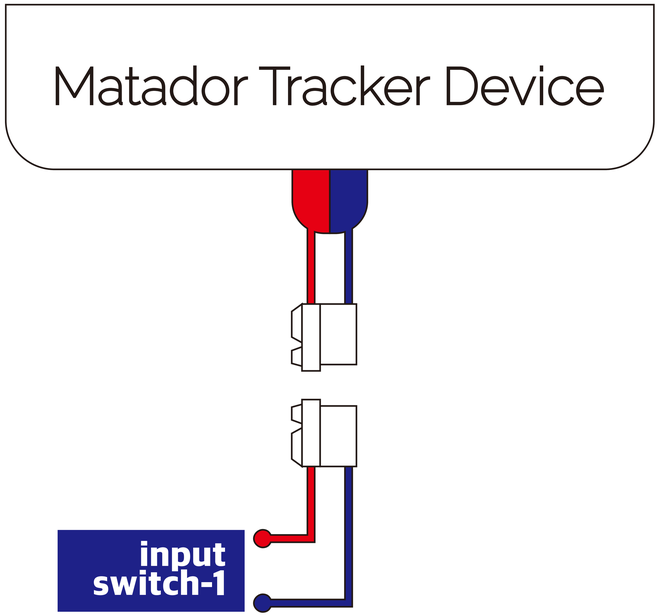
```
mode : input sensor has 2 mode, first is normal logic second is
reverse logic. For generally, input sensor works using threshold
to decide output logic. At the normal logic, while the input
sensor below the threshold, output logic will be set 0
or LOW, and set 1 or HIGH while above it. For case of
reverse logic. Its totally reverse of normal logic, the output
value will be set 1 if input sensor value below the threshold
and set 0 while above it.
```

```
threshold : value limit where the output changes(range is 0-1023)
```

Input Sw - 1

address : this part, value must be `1`

>> Hardware Assembly <<



>> Analog Mode Settings <<

command :

setAnalogOperatingStatus(<address>,<min>,<max>)

example :

setAnalogOperatingStatus(1,1,1023)

>> Default Config <<

min: 0

max : 1023

>> info <<

input sensor value divided as minimum and maximum value. At the system minimum value of input sensor is represent as a 0 in a percent(%). And maximum value of switch is declared as 100 in a percent(%).

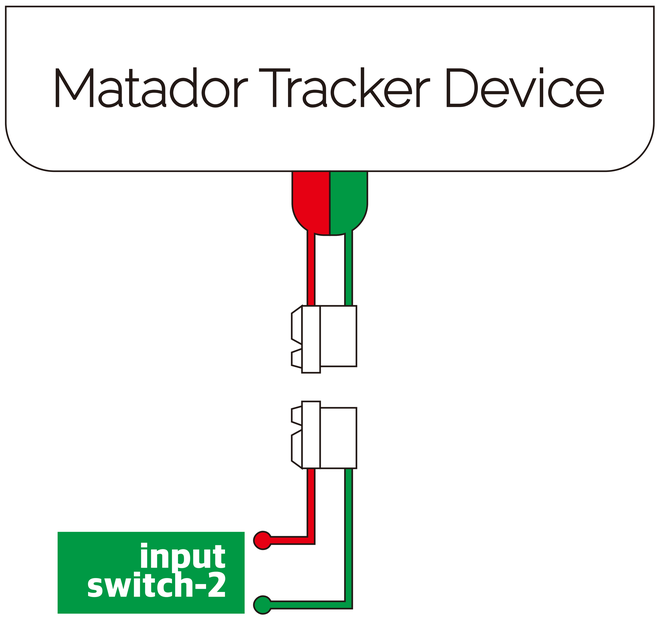
min : the lower bound of the value’s current range.

max : the upper bound of the value’s current range.

Input Sw - 2

```
address : this part, value must be `2`
```

>> Hardware Assembly <<



>> Digital Mode Settings <<

command :

```
setDigitalOperatingStatus(<address>,<mode>,<threshold>)
```

example :

```
setDigitalOperatingStatus(2,1,500)
```

>> Default Config <<

```
mode: 0
```

```
threshold : 400
```

>> info <<

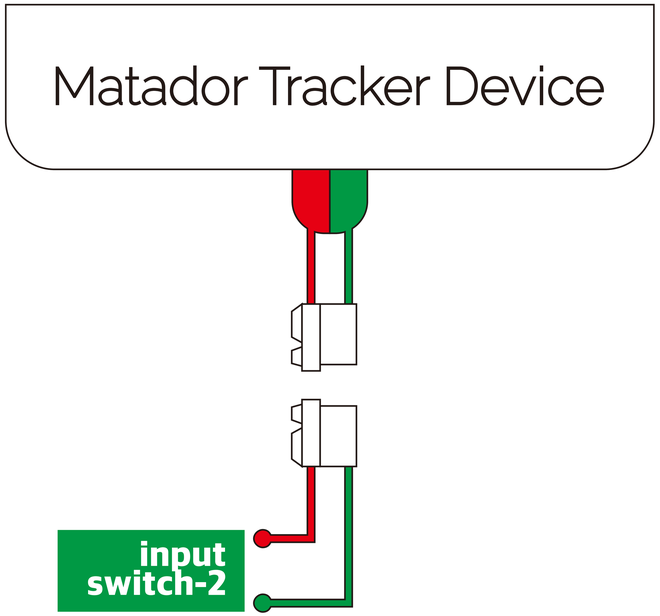
```
mode : input sensor has 2 mode, first is normal logic second is
reverse logic. For generally, input sensor works using threshold
to decide output logic. At the normal logic, while the input
sensor below the threshold, output logic will be set 0
or LOW, and set 1 or HIGH while above it. For case of
reverse logic. Its totally reverse of normal logic, the output
value will be set 1 if input sensor value below the threshold
and set 0 while above it.
```

```
threshold : value limit where the output changes(range is 0-1023)
```

Input Sw - 2

address : this part, value must be `2`

>> Hardware Assembly <<



>> Analog Mode Settings <<

command :

setAnalogOperatingStatus(<address>,<min>,<max>)

example :

setAnalogOperatingStatus(2,1,1023)

>> Default Config <<

min: 0

max : 1023

>> info <<

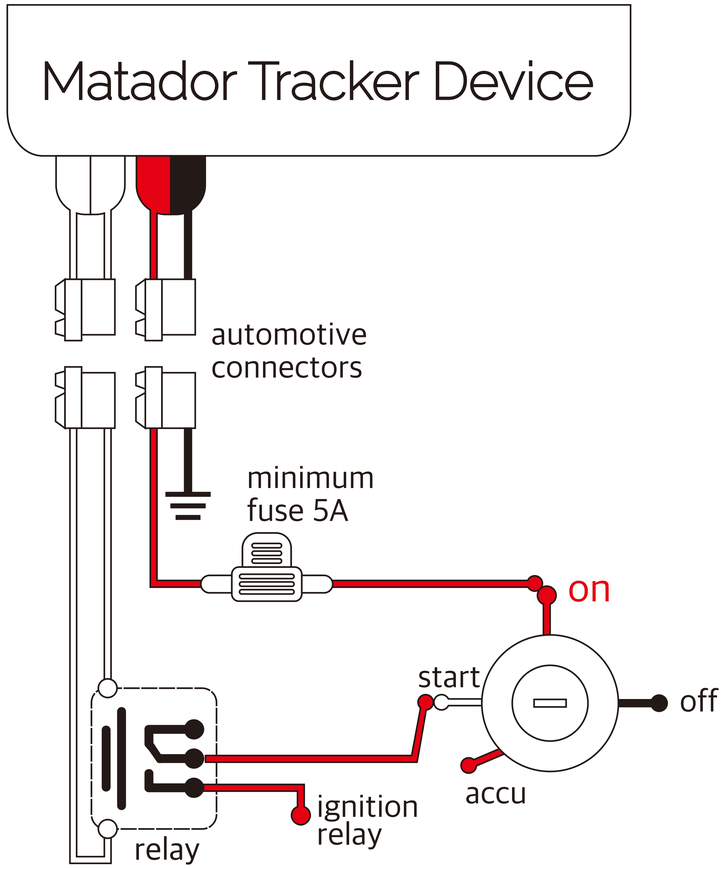
input sensor value divided as minimum and maximum value. At the system minimum value of input sensor is represent as a 0 in a percent(%). And maximum value of switch is declared as 100 in a percent(%).

min : the lower bound of the value’s current range.

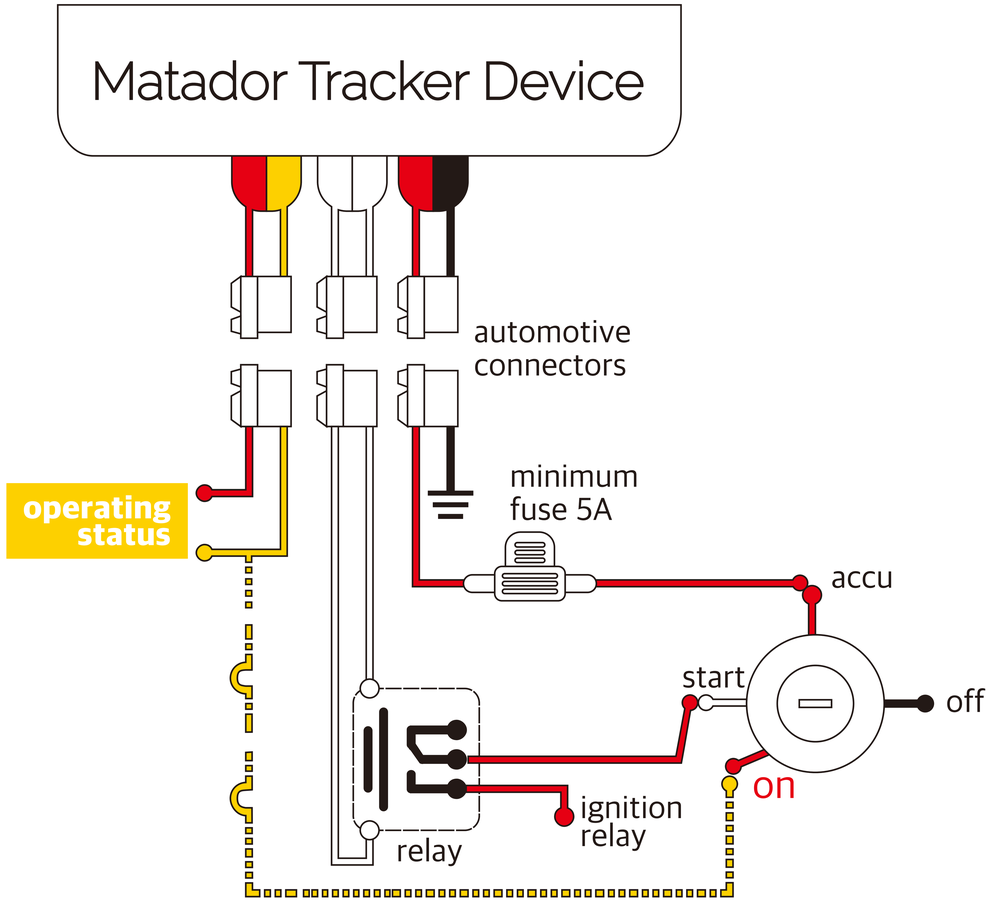
max : the upper bound of the value’s current range.

Relay

>> Hardware Assembly <<



>> Uncontinuous Mode <<



>> Continuous Mode <<

Relay

>> info <<

```
mode : relay has 2 mode, first is normal close logic,
and second is normally open logic.
```

>> Mode Settings <<

command :

```
setModeRelay(<String, "nc" or "no">)
```

example for normally close logic :

```
setModeRelay(nc)
```

example for normally open logic :

```
setModeRelay(no)
```

>> Default Config <<

```
mode: nc
```