

# **Titan Router** V6 Firmware

Sending data to ThingsBoard platform

www.webdyn.com

## Scenario Details

TITAN routers have all the typical functionalities of 4G/3G/2G routers, as well as a series of added features that make them one of the most feature-packed routers on the market.

One of the added features is the datalogger, where the TITAN router can store a number of types of records in its non-volatile memory in JSON format. These records can come from MODBUS readings, SERIAL data captures via the RS232 / RS485 ports, or GPS positions, etc. These JSON-type records are stored in the TITAN router's internal non-volatile memory and can subsequently be sent to remote platforms via protocols such as HTTP, HTTPS, MQTT, MQTTS, FTP and FTPS.

As mentioned, the TITAN router stores the JSON registers in its internal memory in a proprietary format by default. This can sometimes be a problem when communicating with platforms that expect to receive information in a certain format (i.e. a format other than JSON, the one used by the TITAN router).

The "JSON Transformer Function Script" enables the TITAN router to format any JSON object before it is stored in the internal memory. This means JSON objects can be converted to the appropriate format for each platform. You don't really need to convert the data into a JSON format, since it can be converted into any String format.

In this application note, we will guide you through an entire example of how send to data to the wellknown ThingsBoard platform (https://www.thingsboard.cloud/), which requires the sent JSONs to be in a special format.

In this particular application note, we will assume that 2 registers are to be read from 2 PLCs with Modbus communications connected to a Webdyn - EasyTunnel via their RS485 port



The main aim is for the "Webdyn - EasyTunnel" device to read the Modbus registers with addresses 40000 and 40001 of PLC1, and registers 40000 and 40001 of PLC2 every minute. In both PLCs, register 40000 corresponds to the measured temperature, and register 40001 to the humidity level. The readings taken must be stored in the internal non-volatile memory of the "Webdyn - EasyTunnel" (in its datalogger), which must send the read data to the DEXMA platform whenever possible (coverage, IP connectivity, etc.). Communication with the PLCs is carried out via an RS485 bus with a 9600,8,N,1 configuration

## 1. WAN mobile configuration.

The "Webdyn - EasyTunnel" must communicate with the ThingsBoard platform via 4G/3G/2G communications, so the "Mobile- Basic Settings" section must be configured correctly according to the SIM card used.

Basic Settings	Mobile WAN	Enabled (IP active)	~	Enable Wireless WAN interface
eep Online	Sim Mode	SIM1	~	Sim selection
hernet Basic Settings				
sic Settings	SIM1 APN:	movistar es	1	SIM Card 1 APN
wall				
horized IPs	SIM1 Username:	MOVISTAR		SIM Card 1 username
l Settings	SIM1 Password:			SIM Card 1 password
rial Port1-RS232 rial Port2-RS485	SIM1 Pin:		]	SIM Card 1 PIN
SL Certificates	SIM1 Auth:	None	~	SIM card 1 authentication
eneric Serial Device emperature Sensor C102 Meter PS Receiver	SIM2 Username: SIM2 Password: SIM2 Pin:		] ] ]	SIM Card 2 username SIM Card 2 password SIM Card 2 PIN
jins eneric	SIM2 Auth:	Auto	~	SIM card 2 authentication
Command	Network selection:	Auto (4G/3G/2G)	~	Network selection
nDns rate DynDns s control iodic Autoreset	DNS selection:	Get DNS from Operator	~	
e Servers	DNS1:	8.8.8.8	1	Preferred DNS1
note Console	DNS2:	8.8.4.4	1	Preferred DNS2
	DN32.	0.0.4.4		FIGICITED DITUSE

## 2. Configuring the RS485 serial port

The two PLCs will connect to 9600,8,N,1 via the RS485 serial port, so the "Serial Settings- Serial Port2-RS485" section must be configured by setting the parameters as shown below.

*	Mobile o Status	Serial Gateway	Com2 Settings	
	Basic Settings	Baudrate:	9600	<ul> <li>Baudrate of serial port</li> </ul>
	• Keep Online	Data bits:	8	<ul> <li>Number of data bit</li> </ul>
*	Ethernet • Basic Settings	Parity:	none	✓ Parity
	Firewall	Stop bits:	1	✓ Number of stop bits
1	• Authorized IPs	Timeout ms:	50	msec without serial data before sending (default: 50)
* C	Serial Port1-R5232 • Serial Port2-R5485 • SEL Cerunicates External Devices • Logger configuration • ModBus Devices • Generic Serial Device • Temperature Sensor • IEC102 Meter			Ex.: <mtxtunnel>AT</mtxtunnel> Ex.: <mtxtunnelr>AT</mtxtunnelr> Only TCP Server and TCP Client functions or Nothing. 2G (CSD) network required.
*	o GPS Receiver Plugins o Generic Other	O Function: Serial - 1 TCP Local Port: Temporal client RS232 Temporal client	P Gateway (TCP Server) 20011	Listening TCP Port (1 65535) Check if you need a temporal TCP Client when data is present at serial port. DDHHMM. Example: XX2200 starts a temporal
	AT Command     DynDns     Private DynDns	Wakeup Temporal client time:	60	client every day at 22:00 Seconds for temporal client
	• Sms control • Periodic Autoreset	Temporal client Random	0	Seconds. Random time for temporal client Wakeup
	• Time Servers • Remote Console	SSL/TLS enabled		SSL/TLS Enabled (SSL Certs needed)

## 3. Logger configuration

The next step is to configure the internal datalogger of the device. Go to the "External Devices- Logger configuration" menu. The configuration should be similar to the one shown in the screenshot below:

	and a state of the		
Basic Settings	ID:	ID0001	Optional. Device identification
• Keep Online	Send mode:	FIFO	<ul> <li>Send mode (normally FIFO)</li> </ul>
Ethernet	Time format:	unix (vvvv-mm-ddTHH:mm:se	<ul> <li>Time format used in timestamp logger data</li> </ul>
<ul> <li>Basic Settings</li> </ul>			
Firewall	Use script:		Check for customized json using 'Json Transformer Script' in Script section.
• Authorized IPs	Use array:		Check if you want to send more than one JSON per transmition.
Serial Settings	Check date:		Save data in Logger only if date has been se (check Time Servers)
• Serial Port1-RS232 • Serial Port2-RS485			
• SSL Certificates	Communication mod	le: WEB PLATFORM (HTTP	REST)
External Devices	Enabled:		Communication mode HTTP enabled
Logger configuration     ModBus Devices	Mode:	HTTP GET (JSON)	<ul> <li>Method of sending data</li> </ul>
Generic Serial Device	Custom parameters:		Optional. Ex: &a=1&b=2 only for "HTTP

You must select the MQTT communication mode at the bottom of the same display and enter v1/devices/ me/telemetry in the telemetry delivery topic field.

External Devices	Enabled:		Communication mode HTTP enabled
Logger configuration     ModBus Devices	Mode:	HTTP GET (JSON)	<ul> <li>Method of sending data</li> </ul>
Generic Serial Device     Temperature Sensor     IEC102 Meter	Custom parameters:		Optional. Ex: &a=1&b=2 only for "HTTP GET/PUT (PARAMETERS)" modes Optional, Custom header1, For example:
• GPS Receiver	Custom header1:		Content-type;application/json Optional. Custom header2. For example:
Other	Custom header2:		IDENTITY_KEY;YOUR_KEY
Other     Other     Other     Other	Custom header3:		Optional. Custom header3.
<ul><li>DynDns</li><li>Private DynDns</li></ul>	Server:		Destination URL. Example: www.mydomain.com/set.asp?data=
Sms control     Periodic Autoreset	Server Username:		Optional. Blank if no server authentication required
Time Servers     Remote Console	Server Password:		Optional. Blank if no server authentication required
Snmp     Tacacs+     Matt	Communication mod	le: FTP SERVER	
• Http / Https	Enabled:		Communication mode FTP enabled
<ul> <li>User Permissions</li> <li>Passwords Web UI</li> </ul>	FTP prot.:	FTP	FTP / FTPS protocol
• CA Certificates • Email Config	FTP Server:		Destination FTP Server. Example: ftp.mydomain.com
<ul> <li>ModBus Slave</li> <li>Titan Scripts</li> </ul>	FTP port:	21	FTP server port. Default 21
Connectivity tools     Digital I/O	FTP Path:		FTP path. Example: /dev/plcs/
Custom Skin	FTP Username:		FTP Username
Led Config     Syslog	FTP Password:		FTP Password
Backup / Factory     Firmware Upgrade	FTP File Period:	day	<ul> <li>FTP File Period (one file every minute, hour, day)</li> </ul>
• Reboot • Logout	Communication mod	le: MQTT	
	Enabled:		Communication mode MQTT enabled
	MQTT Topic	v1/devices/me/telemetry	MQTT Topic. Example: [IMEI]/logger
	<u> </u>		Note: Other>MQTT menu must be configured

The following parameters are particularly important:

• "Time format": select the "unix" time format.

• "**Use script**": you must check this box, because activating it will run the script that you will integrate later. This script will also convert the standard JSON created by the device, with the data, into a JSON with the required format supported by the ThingsBoard platform.

• "**MQTT Enabled**": The logger must use the MQTT delivery method to send data to the ThingsBoard platform, so this box needs to be checked.

• **"MQTT Topic"**. Indicates the telemetry delivery topic – "v1/devices/me/telemetry" – the topic required by the ThingsBoard platform.

### 4. Configuring the Modbus section

In this configuration section ("External Devices- Modbus Devices"), you will configure the Modbus readings to be performed on the device on the 2 PLCs.

*	Mobile • Status	External Dev	ices	ModBus I	RTU / TO	CP					
	Basic Settings     Keep Online	Enabled:	ľ					lbus Devices			
*	• Basic Settings	Serial Port: Logger:		Serial Port 2		~	Select the o	connected se Iger must be		if need	ed
*	Firewall • Authorized IPs	SAVE CONFIG		VIEW LOG		_	Please, con option	figure logge	r before	using th	ils
*	Serial Settings • Serial Port1-RS232 • Serial Port2-RS485 • SSL Certificates	SAVE CONFIG		VIEW LOG							
	• SSL Certificates	Dev. name / ID	Addr.	Command	Start @	Nun	n word/bit	Reg Type	Period		
*	• Logger configuration	PLC1	1	0x04	40000		2	WORD	1	Del	Test
1	ModBus Devices     Generic Serial Device	PLC2	2	0x04	40000		2	WORD	1	Del	Test
	• Temperature Sensor • IEC102 Meter • GPS Receiver	Device name / ID	:	PLC2			Insert the o	levice name	or ID	1. 640	
*	Plugins	Address:		2			Modbus RT	U address or	IP:port	address	13
	• Generic	Command:		0x04		*	Modbus rea	d command			
*	Other	Start:		40000			Address of	the first regi	ister		
	AT Command     DynDns	Number Words / E	Bits:	2			Words for o 0x01/0x02	command Ox	03/0x04	. Bits fo	r
	Private DynDns     Sms control	Reg Type:		WORD		~	Type of reg	isters for co	mmand	0x03/0x	04
	Periodic Autoreset     Time Servers	Period:		1		~	Read period	d (minutes)			

As can be seen in the previous screenshot, the Modbus service must be enabled by activating the "Enabled" checkbox. You must select the "Serial Port 2" serial port, since the reading will be made via the RS485 port. The "Logger" box must also be checked, since the Modbus registers read from the PLCs must be stored in the internal datalogger of the Titan-based device.

Two new devices must also be created, which we have called PLC1 and PLC2 in this example. You must indicate the Modbus address of each of the PLCs in the "address" field. PLC1 shall have Modbus address 1, and PLC2 shall have Modbus address 2. The Modbus command you will use in this example to read registers 40000 and 40001 will be 0x04, so you must select 0x04 in the "Command" field. The first register to be read in both PLCs is register 40000, so enter 40000 in the "Start" field. The aim is to read 2 Modbus registers from each PLC (40000 and 40001), so enter the value "2" in the "Number Words"

field. Select "WORD" for the register type ("Reg Type") and 1 for the "Period", because you will want to receive readings from the Modbus registers every minute.

## 5. Configuring the SCRIPT to convert the standard formatted JSON generated by the Titan-based device to the format we want to use to communicate with the ThingsBoard platform.

The standard JSON format generated by the Titan-based device, with the data read, and which, in its standard form (not converted), will be sent to a platform, will take the following format:

#### For PLC1:

{"IMEI":"869101054286683","TYPE":"MODB","TS":"2022-11-11T12:17:00Z","ID":"PLC1","A":"1","ST": "40000","N":"2","V":[225,62],"P":"ID0001"}

#### For PLC2:

{"IMEI":"869101054286683","TYPE":"MODB","TS":"2022-11-11T12:17:01Z","ID":"PLC2","A":"2","ST": "40000","N":"2","V":[241,71],"P":"ID0001"}

But to make things easier, you will ideally want to send the data to the ThingsBoard platform as follows:

#### For PLC1:

{"TS1":"2022-11-11T12:17:00Z", "TEMP1":22.5,"HUM1":62}

#### For PLC2:

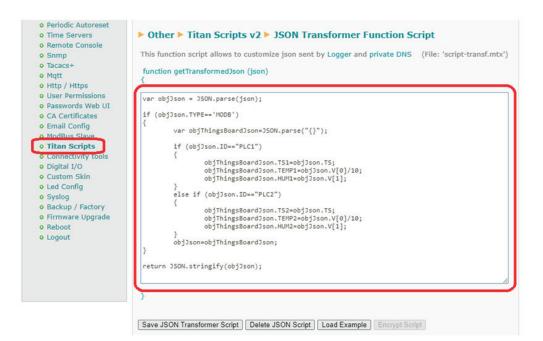
{"TS2":"2022-11-11T12:17:00Z", "TEMP2":24.1,"HUM2":71}

In other words, the aim is to assign the temperature of each PLC (further dividing each value by 10 to obtain the temperature in degrees) to the JSON fields "TEMP1" and "TEMP2". The fields "HUM1" and "HUM2" shall be assigned the humidity values obtained from the Modbus reading.

You must therefore create a short SCRIPT to convert the data from the standard format to that required for sending data to ThingsBoard. Remember that this conversion script will be run because you previously checked the "Use script" box in the "External Devices- Logger Configuration" section.

<ul><li>Firewall</li><li>Authorized IPs</li></ul>	Use script: Use array:		Check for customized json using Json Transformer Script <sup>4</sup> in Script section. Check if you want to send more than one JSON per transmition.
<ul> <li>Serial Settings</li> <li>Serial Port1-RS232</li> <li>Serial Port2-RS485</li> <li>SSL Certificates</li> </ul>	Check date:	mode: WEB PLATFORM (H	Save data in Logger only if date has been set (check Time Servers)
External Devices     O Logger configuration     O ModBus Devices	Enabled: Mode:	HTTP GET (JSON)	Communication mode HTTP enabled <ul> <li>Method of sending data</li> </ul>

To enter the code for the data conversion, go to the "Other- Titan Scripts" menu. The conversion script must be entered in the "JSON Transformer Function Script" section.



The SCRIPT will have the following code.

//Create a JSON-type object variable, because the "json" variable that receives the

//function as an argument is a "String"-type variable.

var objJson = JSON.parse(json);

//If it is indicated in the original JSON received that it is a Modbus data JSON ...

```
if (objJson.TYPE=='MODB')
```

{

//Create a new JSON object where you will store the converted data
//transformados

```
var objDexmaJson=JSON.parse("{}");
//If the data corresponds to PLC1 readings ....
if (objJson.ID=="PLC1")
{
       //Add the date of the data.
       objThingsBoardJson.TS1=objJson.TS;
       // Add the temperature reading, divided by 10, in the TEMP1 field.
       objThingsBoardJson.TEMP1=objJson.V[0]/10;
       // Add the humidity reading in the HUM1 field.
       objThingsBoardJson.HUM1=objJson.V[1];
}
//If the data corresponds to PLC2 readings ....
if (objJson.ID=="PLC2")
{
       //Add the date of the data.
       objThingsBoardJson.TS2=objJson.TS;
       // Add the temperature reading, divided by 10, in the TEMP2 field.
       objThingsBoardJson.TEMP2=objJson.V[0]/10;
       // Add the humidity reading in the HUM2 field.
       objThingsBoardJson.HUM2=objJson.V[1];
}
//Then, copy the converted JSON to return it as a result
// of the conversion function.
```

objJson=objThingsBoardJson;

}

## 6. Configuring the MQTT section and ThingsBoard platform

You must also correctly configure the "Other - Mqtt" section of the Titan-based device so that it can connect to the ThingsBoard platform. But first, you must go to the web platform to add the "Webdyn - EasyTunnel" device and obtain the mqtt authentication attributes.

Once you have accessed your http://thingsboard.cloud account, go to the "Device groups" section. Click on the "+" icon to create a new group.

Ø Intelligent Router X Ø Thin ← → C @ thingsboard.cloud/device0	gsBoard Cloud   Device grov X +						Q	<ul><li>✓</li><li>✓</li></ul>	-	
	😡 Device groups			Current subscription Thi Status Tria	ngsBoard Cloud Maker	0	0	lose Galle Tenant adr Add entity o		tor 1
Customers hierarchy	Device groups							+	c	Q
😫 User groups 🛛 🗸								-		
🟦 Customer groups 🗸 🗸	Created time 🕹	Name	Description		Public					
🖬 Asset groups 🗸 🗸	2023-01-26 15:47:12	All					<u>e</u> •	< *	1	ii.
el Device groups										
(a) AI										
🕯 Profiles 🗸 🗸										
Entity view groups 🗸 🗸										
🖥 Edge groups 🛛 🗸										
🕅 Edge management 🗸 🗸										
Widgets Library										
Dashboard groups 🗸 🗸										
OTA updates										
Scheduler										
White Labeling 🗸 🗸										
Version control				1	ems per page: 10 🗢	1-1of		< <	5	31
du Audit Loos										~

For example, you can enter the group name "Webdyn". Then, click on the "Add" button.

<ul> <li>⊘ Intelligent Router</li> <li>X</li> <li>Y</li> <li>Y<!--</th--><th>gsBoard Cloud   Device grov × +</th><th></th><th></th><th></th><th><ul> <li>✓ - σ ×</li> <li>◎、 ☆ ★ □ ● ;</li> </ul></th></li></ul>	gsBoard Cloud   Device grov × +				<ul> <li>✓ - σ ×</li> <li>◎、 ☆ ★ □ ● ;</li> </ul>
ThingsBoard	Co Device groups			ThingsBoard Cloud Maker	Jose Gallego Tenant administrator
Customers hierarchy User groups	Device groups	Add entity group	e ×	Public	+ C Q
Customer groups     Asset groups     Control Device groups	2023-01-26 15:47:12 All	Entity group details     Name *     Webdyn	Share entity group Optional		<b>8</b> < ∧ ∕ ≣
Co① All  Profiles ✓ Entity view groups ✓		Description			
Edge groups V			_		
Widgets Library Dashboard groups			Next: Share entity group		
Scheduler			Cancel Add		
Version control     Audit Logs				Items per page: 10 💌	1-1of1  < < >)

Once the "Webdyn" group has been created, enter the group created (by clicking on the line corresponding to that group) and then create a new device by clicking again on the "+" icon.

🕲 Intelligent Router 🛛 🗙 💋	hingsBoard Cloud   Device gro.: × +					~ - σ ×
← → C @ thingsboard.cloud/devi	ceGroups/69be1580-a07b-11ed-bf3d-5d40b1136f31				Q, y	9 🖈 🇯 🖬 🐌 i
ThingsBoard	🗔 Device groups 🔉 🗔 W	ebdyn	Current subsorip Sta	ion ThingsBoard Cloud Maker tus Trial ends on the Feb 25, 2023	C 😫 Jos	e Gallego ant administrator
	1					1.5
- Customers hierarchy	Webdyn: Devices 🧳				+	1 C Q
😌 User groups 🛛 🗸 🗸						5 I
📇 Customer groups 🗸 🗸	Created time 🕹	Name	Device profile	Label		
🗈 Asset groups 🗸 🗸						
LoD Device groups						
[0] AI						
Coll Webdyn						
🖆 Profiles 🗸 🗸						
Entity view groups 🗸 🗸			No devices found			
🛃 Edge groups 🗸 🗸			NO GENCES IOUNA			
👚 Edge management 🛛 🗸						
🔡 Widgets Library						
Dashboard groups 🗸						
OTA updates						
Scheduler						
🏲 White Labeling 🛛 🗸				Items per page: 10 -	1-0of0 [<	
Version control					IS	a 6 6

You can enter "Webdyn-Easy-Tunnel-1" as the device name. Then, you must click on the "Add" button.

	Throppharel Courd (brive ges: x + + + + + + + + + + + + + + + + + +	v - o x Q ± x <b>x ⊡ Ø</b> i
		Jose Gallego Tenant administrator
ruies     Customers hierarchy	Add new device 2 ×	+ <u>∗</u> c Q
User groups	Optional	
Customer groups     Asset groups	Webyer, Fasy Timpel 1	
Coll Device groups	Label	
CoD Webdyn	Select existing device profile      default      x	
Profiles	Create new device profile	
Edge groups		
Edge management     Widgets Library	Description	
Dashboard groups		
OTA updates Scheduler	Next: Credentials	
T White Labeling	Cancel Add Imma per page 10 • 1 - 0 o	fo:  < < > >
Version control		

Next, click on the line corresponding to the newly created device and click on the "Copy device ID" and "Copy access token" buttons. You will need to note down these values, as they will need to be integrated into the "Other - MQTT" section of the Titan-based device.

	loard Cloud   Device grov. X +	ν - σ x
	epu#BetSB0-a078-11ed-8580-56808113803 តែជិ Device groups > តែជិ Webdyn	Current subscription ThingsBoard Cloud Maker Status Tritil ends on the Fe225,2223
Customers hierarchy User groups	Webdyn: Devices 🧳	Webdyn-Easy-Tunnel-1
21 Customer groups 🗸	Created time 🕹 Name	Contails Attributes Latest telemetry Alarms Events Relations Audit Logs
Asset groups	2023-01-30 09:57:36 Webdyn Easy-Ten	Open details page         Manage credentials         Delete device           Copy device Id         Copy access token         Copy access token
CaO All CaO Webdyn		Name Webdyn-Easy-Tunnel-1
Profiles 🗸		Device profile default
🛃 Edge groups 🗸 🗸		Label
Widgets Library		Assigned firmware
OTA updates		Assigned software
<ul> <li>♦ Scheduler</li> <li>♥ White Labeling</li> </ul>		Is gateway Description
Version control		

Now, in the "Other-Mqtt" section of the Titan-based device, you must enable the "MQTT Client" service so that it can connect to the ThingsBoard platform. Enter the value "tcp://thingsboard.cloud:1883" in the "MQTT Broker" field, the "access token" in the "Mqtt Username" field and the "device ID" in the "MQTT ID" field, with all these fields copied from the previous step. Then, press the "SAVE Config" button and restart the Titan-based device to apply the new configuration.

o Status			
• Basic Settings	Enabled:		Enable MQTT broker service
• Keep Online	TCP Port:	1883	Listening port (for example 1883)
e Basic Settings	Anonymous User:		Allow anonymous user (no user / password is needed)
Firewall	User:		Username (used if anonymous is not selected
o Authorized IPs	Password:		Password (used if anonymous is not selected)
• Serial Port1-RS232 • Serial Port2-RS485 • SSL Certificates	► Other ► MQTT	Client	
External Devices			
<ul> <li>Logger configuration</li> <li>ModBus Devices</li> </ul>	Enabled:		Enable MQTT client
<ul> <li>Generic Serial Device</li> <li>Temperature Sensor</li> <li>IEC102 Meter</li> <li>GPS Receiver</li> </ul>	MQTT Broker	tcp://thingsboard.cloud:1883	Destination MQTT Broker. Examples: tcp://test.mosquitto.org:1883 ssl://test.mosquitto.org:8883 (certificate needed) ssl://test.mosquitto.org:8884 (certificates needed)
Other	MQTT Username	2X5PYTCx80KYNo1ftqwP	MQTT Username (blank if not used)
• AT Command • DynDns	MQTT Password		MQTT Password (blank if not used)
Private DynDns     Sms control	MQTT ID	2474d9e0-a07c-11ed-a8d5-€	Device identification
Periodic Autoreset     Time Servers	MQTT Qos	1	MQTT Quality Of Service (0 2)
• Remote Console	MQTT Keepalive	60	Seconds for keepalive (30 3600)
• Snmp • Tacacs+	MQTT Persistence		Data persistence
Mqtt     Http://			
User Permissions	MOTT AT Topic		This topic will be subscribed for receiving AT

After restarting the Titan-based device and waiting a few seconds for the IP connection, as long as everything is in order, the "Other - MQTT" section will display the status of the mqtt connection as follows:

<ul> <li>Mqtt</li> <li>Http / Https</li> <li>User Permissions</li> <li>Passwords Web UI</li> <li>CA Certificates</li> <li>Email Config</li> <li>ModBus Slave</li> <li>Titan Scripts</li> <li>Connectivity tools</li> <li>Digital 1/0</li> <li>Custom Skin</li> <li>Led Config</li> <li>Syslog</li> <li>Backup / Factory</li> <li>Firmware Upgrade</li> <li>Reboot</li> <li>Logout</li> </ul>	MQTT AT Topic MQTT AT Resp Topic MQTT AT Topic 2 MQTT AT Resp Topic 2 MQTT AT Resp Topic 3 MQTT Script Topic 1 MQTT Script Topic 2 SAVE CONFIG	v1/devices/me/rpc/request/+	This topic will be subscribed for receiving AT Commands (usefull for individual device) This topic will be used for publishing the AT Command Responses of AT Topic This topic will be subscribed for receiving AT Commands (usefull for groups) This topic will be used for publishing the AT Commands (usefull for all devices) This topic will be subscribed for receiving AT Commands (usefull for all devices) This topic will be used for publishing the AT Command Responses of AT Topic 3 When data is received in this topic the 'Topic Script' will be executed. When data is received in this topic the 'Topic Script' will be executed.
	Other > MQTT Clie Internet status: MQTT connection status: REFRESH	online Connected	Checked every MQTT Keepalive period

At this point, you cannot tell whether the ThingsBoard platform is receiving the data from the Modbus registers corresponding to the temperature and humidity readings from both PLCs via Modbus. To check this, click again on the line corresponding to the device, then click on the "Latest telemetry" section. In this particular section of the ThingsBoard platform, if the data sent by the Titan-based device is being received correctly, it should appear as shown in the screenshot below:

	psRoad Coud (Device pro. x + roups/69be1580-a07b-11ed-b13d-5d40b113601		v – o x Q & A 🛪 🖬 🐼 🗄
ThingsBoard	តេរី Device groups 🔸 🗔 Webdyn	Current subscription Thin Status Tribl	aniloard Cloud Maker Jose Gallego : Canart administrator :
Customers hierarchy	Webdyn: Devices 🎤	Webdyn-Easy-Tunnel-1	• ×
User groups	Created time 🕹 Name	Contact declaration     C	Alarms Events Relations Audit Logs
🖿 Asset groups 🗸 🗸	2023-01-30 09:57:36 Webdyn-Easy-Tunn		
Go⊡ Device groups ∧		Latest telemetry	Q
Cell Webdyn		Last update time Key 🕈	Value
Profiles 🗸		2023-01-30 10:19:03 HUM1	62
🛋 Edge groups 🗸 🗸		2023-01-30 10:19:03 TEMP1	22.5
<ul> <li>Edge management</li> <li>Widgets Library</li> </ul>		2023-01-30 10:19:04 TEMP2	24.1
Dashboard groups V		2023-01-30 10:19:03 TS1	2023-01-30T09:19:01Z
OTA updates     Schotzking		2023-01-30 10:19:04 TS2	2023-01-30T09:19:01Z
<ul> <li>③ Scheduler</li> <li>➡ White Labeling ✓</li> </ul>			ms per page: 10 • 1 - 6 of 6   < < > >
Version control			un hei halfer in + i - a nig   ζ ζ > >

## 7. Building a Dashboard in ThingsBoard to visualise temperature and humidity data.

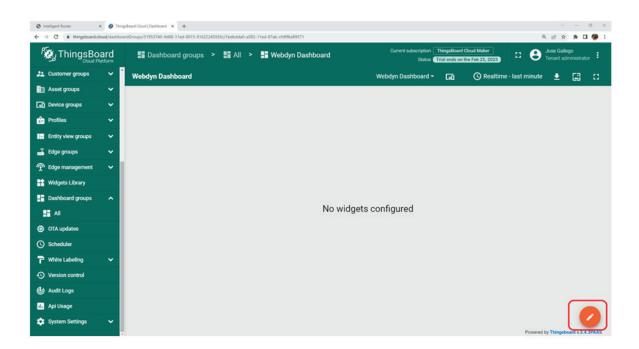
To create a dashboard to graphically represent the data, click on the "DashBoard groups - All" section, and then on the "+" icon, to create a new Dashboard.

		dGroups/51953740-9d88-11ed-8015-91622243936c		ion ThingsBoard Cloud Maker			* * 1		4
ThingsBoa	form	Dashboard groups > All		tus Trial ends on the Feb 25, 2023		B Jose Tena	Gallego Int admin		
Customer groups	× 1								
Asset groups	*	All: Dashboards 🧨				+	±	C	9
Device groups	•	Created time	Title						
Profiles	*								
Entity view groups	*								
Edge groups	*								
Edge management	<b>~</b>								
Widgets Library									
Dashboard groups	^								
All			No dashboards found						
OTA updates									
) Scheduler									
White Labeling	×								
) Version control									
Audit Logs									
Api Usage									
System Settings	<b>v</b>			Items per page: 10 -	1-0 of 0	K	<	>	>

You can enter "Webdyn Dashboard" as the name of the dashboard and click on the "Add" button.

Carlos Contractor Contractor	TringsBoard Cloud   Deshboard × +	22243936c		✓ - 0 ×
	Dashboard groups	> 🏭 All	Current subscription Thin Status Citist	exBoard Cloud Maker Jose Gallego ends on the Feb 25 (2023)
21 Customer groups 🗸	• All: Dashboards	Add Dashboard	0 ×	+ ± C Q
🗈 Asset groups 🗸 🗸	All: Dashboards	Add Dashboard	• ^	+ ± C 4
□ Device groups ✓	Created time 🕹	Title *		
🖆 Profiles 🗸 🗸		Webdyn Dashboard		
📰 Entity view groups 🗸 🗸		Description		
🝶 Edge groups 🗸 🗸				
👚 Edge management 🗸 🗸		Mobile application setting	ngs	
Widgets Library		Dashboard image		
Dashboard groups				
II AI		No image selected ×	Drag and drop an image or Browse file	
OTA updates				
Scheduler		Maximum upload file size: 512.0	KB	
T White Labeling V		Hide dashboard in m	obile application	
Version control				
() Audit Logs		Dashboard order in mob		
1. Api Usage			Cancel	
🗱 System Settings 🗸 🗸			te	ms per page: 10 ▼ 1 - 0 of 0  < < > >

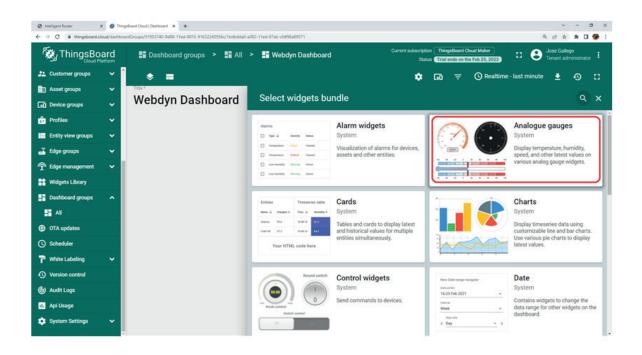
Next, click on the line corresponding to the newly created dashboard, then on the edit icon at the bottom right.



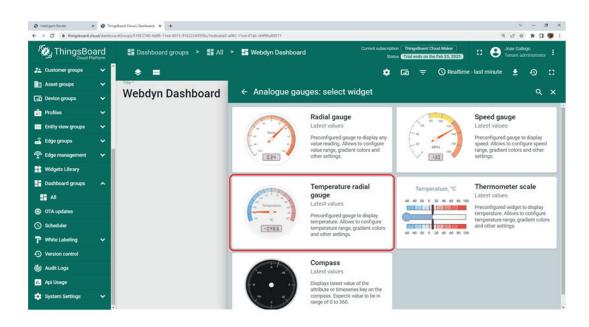
Click again on the "Add new widget" button to add a new widget to the dashboard.

		after Coul (Defound x + - 0 X
		Correct subscription for upset of the subscription for the subscrine for the subscription for the subscription for the subscriptio
2 Customer groups	~	📚 🚍 🌼 🛱 🎝 🎝 🔅
Asset groups	~	
Col Device groups	~	Webdyn Dashboard
Profiles	~	
Entity view groups	~	
🛃 Edge groups	~	
😤 Edge management	~	
Widgets Library		
Dashboard groups	^	(
E AI		Add new widget
OTA updates		
Scheduler		
P White Labeling	~	
Version control		
🕲 Audit Logs		
ili Api Usage		
🔅 System Settings	~	Powere by Thingshard v3.43PAS

Select "Analogue gauges" as the gauge type.



A "Temperature Radial gauge" will then be added.



Once added, click on the "Create a new one" link in the "Temperature radial gauge" properties area of the "Data" section to create a new Alias referring to the Titan-based "Webdyn - EasyTunnel" device.

	ThingsBoard Cloud   Dashb											~		0 ×
-				936c/1bdbdda0-a082-11ed-87			Current subscription Thin	- Boord Class	ditteler			L Le ☆ Jose Galler		<b>9</b> i
ThingsBoar	d 🔡 Das			E All > E 1	Webdyn Das	hboard	Status Trial			• •		Tenant adn		
22 Customer groups	~ 📔 🌒											±		0
Asset groups	×	Add Wid	get: Tem	perature radial ga	auge			?	×					
Device groups	~ Wel	Dat	a	Settings	Advance	ed Actions			- î					
Profiles	<b>~</b>								18					
Entity view groups	~	Dataso Maximun	urces n 1 datasource i	s allowed.				^	- 18					
🛃 Edge groups	~		Туре	Parameters					- 18					
Edge management	~			Entity alias *										
Widgets Library		= 1.	Entity	• No aliases f	ound.	Latest data keys	98749940	— ×						
Dashboard groups	^			Create a new		Entity timeseries/attributes are in Maximum 1 timeseries/attribute								
E All						-								
OTA updates		Deste es												
C Scheduler		Data se	ttings					~	- 11					
T White Labeling	~								- 1					
Version control									- 8					
(b) Audit Logs							Car	cel 1	• Add					
🚯 Api Usage		-			_		Cu							
🔅 System Settings	~											Č	e	×
											Powered t	y Thingsbo	ard v.3.4.3	SPAAS

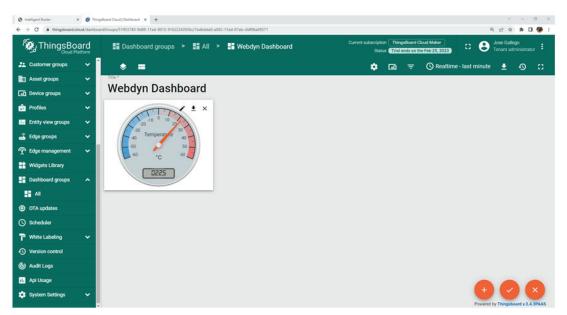
	ard	E Dashboard groups >	All > ## Webdyn Dashboard Current Subscription ThingsBoard Gloud Maker Status (1111:contemport 129:39:0023)	C S Jose Gallego E Tenant administrator
2. Customer groups	~			
Asset groups	· •	Add Widget: Tempera	nture radial gauge 🧿 🗙	
Device groups	~	We Data	in the second	
Profiles	~	( reserves )	Add alias X	
Entity view groups	×	Datasources Maximum 1 datasource is allow	Avias name * Resolve as multiple entities	
🝶 Edge groups	~	Туре	Alias Webdyn-easy-tunnel	
🙊 Edge management	~		Filter type * Device type	
😫 Widgets Library		= 1. Entity	Device type * 1	
Dashboard groups	^		default ×	
EE AI				
OTA updates			Device name expression	
Scheduler		Data settings	· · · · · · · · · · · · · · · · · · ·	
T White Labeling	~		Cancel	
S Version control				
() Audit Logs			Cancel Add	
🔝 Api Usage			VIII VIII POU	
🗱 System Settings	× .			Powered by Thingsboard v 3.4.3PAAS

Create the Alias to be added to the device and click the "Add" button.

After creating the Alias, you must select the "TEMP1" field from the drop-down menu next to it. The dropdown fields will be displayed automatically if the connection to the Titan-based device is established and the modem has already sent data. You must then click on the "Add" button to finish.

Ø Intelligent Router	× 0 Thi	ngsBoard Cloud   Dr	shboard X -	r i								~		a ×
← → C @ thingsboard.c	loud/dashbo	oardGroups/5195	3740-9d88-11ed	-8015-916222431	36c/1bdbdda0-a082-11ed-87a	b-cfdf96a89571						0, 12 ☆	* 0	1 C
ThingsBo	ard	II D			22 Ali > 22 W	/ebdyn Dashb			n ThingsBoard Clou s Trial ends on the F		• <b>8</b>	Jose Galle Tenant adr		
22 Customer groups	~		Add Wid	lget: Temp	perature radial gau	ıge			0	×	me - last minute	• <b>±</b>	Ð	0
Asset groups	~ )	We			1999 - 1999 - 1999 - 1999 - 1999 - 1999 - 1999 - 1999 - 1999 - 1999 - 1999 - 1999 - 1999 - 1999 - 1999 - 1999 - 1999 - 1999 - 1999 - 1999 - 1999 - 1999 - 1999 - 1999 - 1999 - 1999 - 1999 - 1999 - 1999 - 1999 - 1999 - 1999 -				1997 (A. 1997)					
Device groups	~	WC	Dat	a	Settings	Advanced	Actions			- 18				
Profiles	~		Dataso							18				
Entity view groups	~			n 1 datasource i	s allowed.				^	- 88				
🛃 Edge groups	~			Туре	Parameters					. 8				
2 Edge management	~				Entity alias * Alias Webdyn-	easy-tunne v	= 🔵 🛩 TEMP	1: TEMP1 🖍 🗙						
Widgets Library			= 1.	Entity	*		+Latest data key		×	- 88				
Dashboard groups	^				Filter		HUM1	owed.		- 88				
E All				-			₩ HUM2	UN DA.						
OTA updates							₩ TEMP1							
() Scheduler			Data se	ttings			₩ TEMP2		~	- 10				
T White Labeling	~						₩ TS1							
S Version control							≁ TS2							
🕑 Audit Logs								<u>.</u>	Cancel	dd				
ala Api Usage			_	_	_	_	_	_						
🕸 System Settings	~										t t	d by Thingsbo	76	2

At this point, the widget will be added to the dashboard and should display the current temperature value read by PLC1: 22.5°C in the example scenario.



To add other widgets containing the temperature value of PLC2 and the humidity values of PLC1 and PLC2, simply repeat the latest steps, selecting the corresponding TEMP2, HUM1 and HUM2 fields for each.

## 8. Direct communication between the ThingsBoard platform and the Titan-based "Webdyn-Easy-Tunnel" device for status reading and remote configuration changes.

To send commands from the ThingsBoard platform to the Titan-based device, you must first add the appropriate MQTT topics. To do so, add the "v1/devices/me/rpc/request/+" topic in the "MQTT Script Topic 1" field of the "Other  $\Box$  Mqtt" section. This will internally forward any data received by the Titan-based device in the "v1/devices/me/rpc/request/xxxxx" topic to the "MQTT Topic Function Script". Please note that this functionality (the ability to add the "+" character in the topic to subscribe to any topic starting with v1/devices/me/rpc/request/xxxxx, where xxxxx can be anything ) is supported in firmware version 6.17 and later versions. After the configuration change, you must restart the Titan-based device.

This is required, because the ThingsBoard platform will send commands intended for the Titan-based device to the "v1/devices/me/rpc/request/xxxxx" topic, where xxxxx will represent a number that will increase with each command sent. The Titan-based device shall send the response to the "v1/devices/me/rpc/response/xxxxx topic.

o Tacacet o Mqtt	MQTT Persistence		Data persistence
<ul> <li>Http / Https</li> <li>User Permissions</li> <li>Passwords Web UI</li> </ul>	MQTT AT Topic		This topic will be subscribed for receiving AT Commands (usefull for individual device)
• CA Certificates	MQTT AT Resp Topic		This topic will be used for publishing the AT Command Responses of AT Topic
• Email Config • ModBus Slave	MQTT AT Topic 2		This topic will be subscribed for receiving AT Commands (usefull for groups)
• Titan Scripts	MQTT AT Resp Topic 2		This topic will be used for publishing the AT
<ul> <li>Connectivity tools</li> <li>Digital I/O</li> <li>Custom Skin</li> </ul>	MQTT AT Topic 3		Command Responses of AT Topic 2 This topic will be subscribed for receiving AT Commands (usefull for all devices)
• Led Config	MQTT AT Resp Topic 3		This topic will be used for publishing the AT Command Responses of AT Topic 3
<ul> <li>Syslog</li> <li>Backup / Factory</li> </ul>			communa responses of Ar Tople 5
• Firmware Upgrade • Reboot	MQTT Script Topic 1	v1/devices/me/rpc/request/+	When data is received in this topic the 'Topic Script' will be executed.
• Logout	MQTT Script Topic 2		When data is received in this topic the 'Topic Script' will be executed.
			Script will be executed.
	SAVE CONFIG		

Now, add the following script in the "Other - MQTT" section:

<pre>var objJson = JSON.parse(stringData); function replaceSpecialChars(data) {</pre>	
<pre>{     data=data.replace(/(\r\n \n \r)/gm, " ");;</pre>	
return data;	
3	
<pre>mtx.println("topic:" + topic);</pre>	
<pre>mtx.println("stringData:" + stringData);</pre>	
<pre>//Si el JSON es de tipo MODB (Modbus) construiremos el nuevo : if (objJson.method=='command')</pre>	SON de resultado
	•
3	
1	
Save TOPIC Script Delete TOPIC Script Load Example Encrypt Scr	

Here is the complete and fully explained script code.

//Create a JSON object with the data received from ThingsBoard
var objJson = JSON.parse(stringData);

//Function to be used to replace special characters such as ENTER, etc. with blank spaces. function replaceSpecialChars(data)

```
{
```

data=data.replace(/(/(\r\n | \n | \r)/gm, " ");; return data;

}

//The topic and data received by the platform are printed through the debug output
mtx.println("topic:" + topic);
mtx.println("stringData:" + stringData);

//If the ID "command" is present in the "method" field of the json received from the platform
if (objJson.method=='command')

{

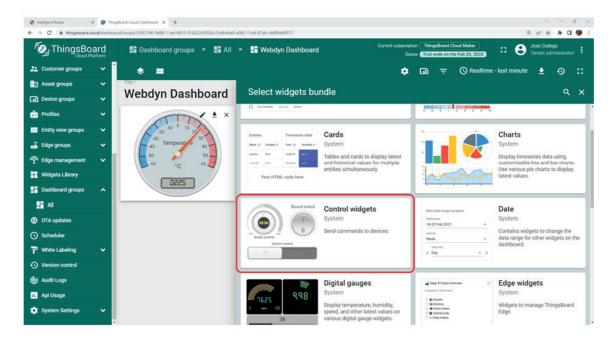
//If there is a command to execute in the "params" field of the received JSON
if (objJson.params!=null)

{

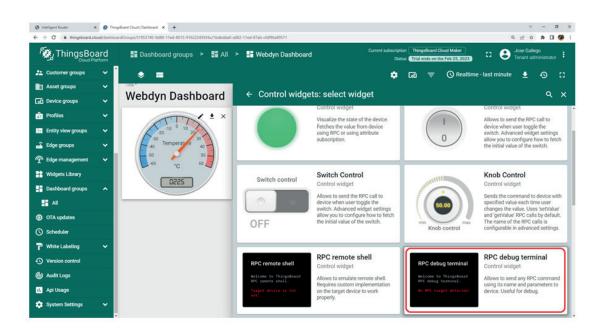
```
//Execute AT command
var res=mtx.atSend(objJson.params,2000);
//Create the response topic, replacing "request" (text) with "response"
var topicResponse=topic.replace("request", "response");
//Send a JSON with the response to the executed command
var answer="{"method":command",{"params":command"] + replaceSpecialChars(res)
var r=mtx.mqttSend(answer,topicResponse,1);
}
```

}

Another widget must be added to the ThingsBoard platform, in this case, by clicking on the "Control widgets" type.



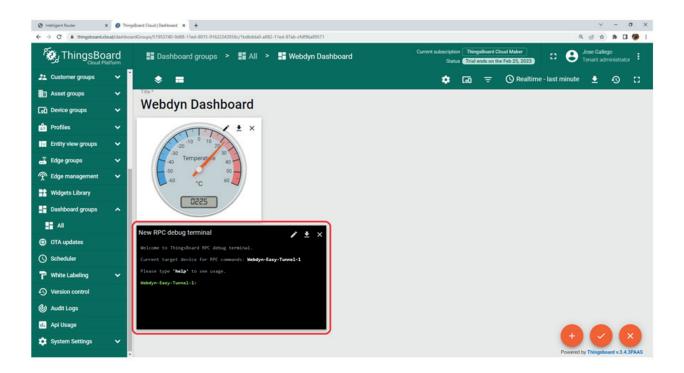
And, in the "Control widgets" section, you must select the "RPC debug Terminal" widgets type:



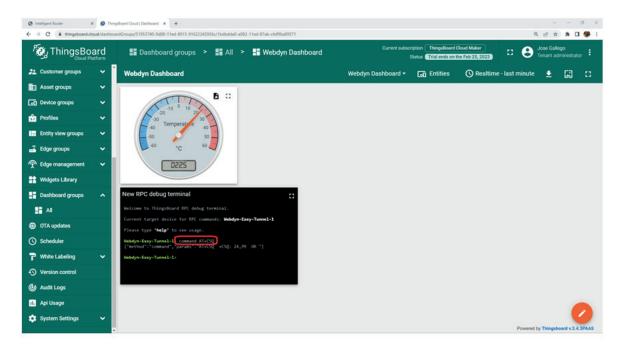
In the "Entity Alias" field, you must select the previously created alias, "Alias Webdyn-Easy-Tunnel", then click on the "Add" button > "V" icon.

	-		board × +	a082-11ed-87ab-cfd95a89571				0,	· -	0 X
ThingsBo	ard	E Da	shboard groups ゝ 🏭 All	> 📲 Webdyn Dashboard		Current subscription ThingsBoard Cloue Status Trial ends on the F		□ <b>8</b> 10	se Gallego nant administ	ator E
22 Customer groups	ř		Add Widget: RPC debug ter	rminal		0		- last minute	± ∙0	8
Asset groups	× ×	We	Data Setting	gs Advanced	Actions		i			
Profiles	~					^				
Entity view groups	~	1	Target device			~				
Edge groups Edge management	ž		Entity alias Alias Webdyn-easy-tunnel							
Widgets Library	Ť	1								
Dashboard groups	~		Data settings			Ý	- 10			
All		-								
OTA updates										
() Scheduler										
P White Labeling	~									
Version control										
(b) Audit Logs						Cancel	dd			
1 Api Usage										
System Settings	~							Powered by	Ŭ,	4.3PAAS

At this point, the widget that allows you to send AT commands to the Titan-based device remotely will appear.



For example, to send any command, you simply need to type the text "command", followed by a blank space, and enter the command to be executed. For example, if you wanted to execute the AT+CSQ command to find out about coverage remotely, you would simply need to follow the steps in the following screenshot:



Any questions?

Please direct your enquiries to iotsupport@mtxm2m.com