Ejemplo thingsBoard paso a paso – para profundizar ver : <https://blog.thingsboard.io/2017/01/esp8266-gpio-control-over-mqtt-using.html> y

<https://www.youtube.com/watch?v=JCW_hShAp7I&t=2s>

este ejemplo muestra cómo mandar telemetría de un sensor DHT11 hacia el servidor y como abrir y cerrar una puerta con un servo y enviar los grados que gira otro motor servo.

## Crear un dispositivo

Crear un dispositivo (poner tipo “default)

Guardar token de acceso

Configurar datos dispositivo y red en código de ejemplo

//#define WIFI\_AP "SSID RED"

//#define WIFI\_PASSWORD "PASSWORD RED"

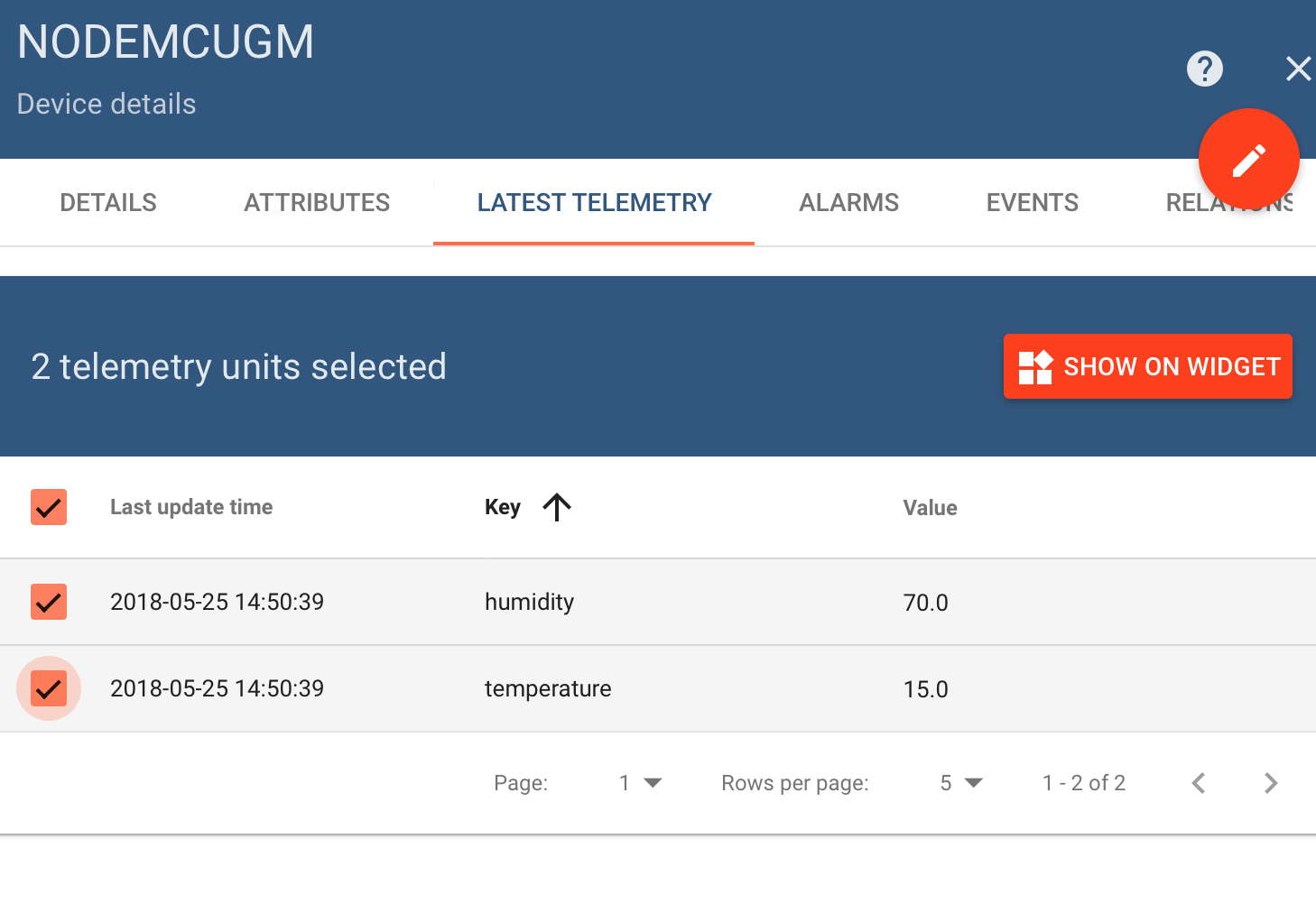
//#define NODE\_ID "NOMBRE DISPOSITIVO"

//#define NODE\_TOKEN "TOKEN DISPOSITIVO"

Donde TOKEN DISPOSITIVO es copiado del dispositivo

Subir el Código al nodeMCU y ejecutarlo

Verificar que se reciban datos de los sensores en el tab LATEST TELEMETRY del panel del dispositivo.



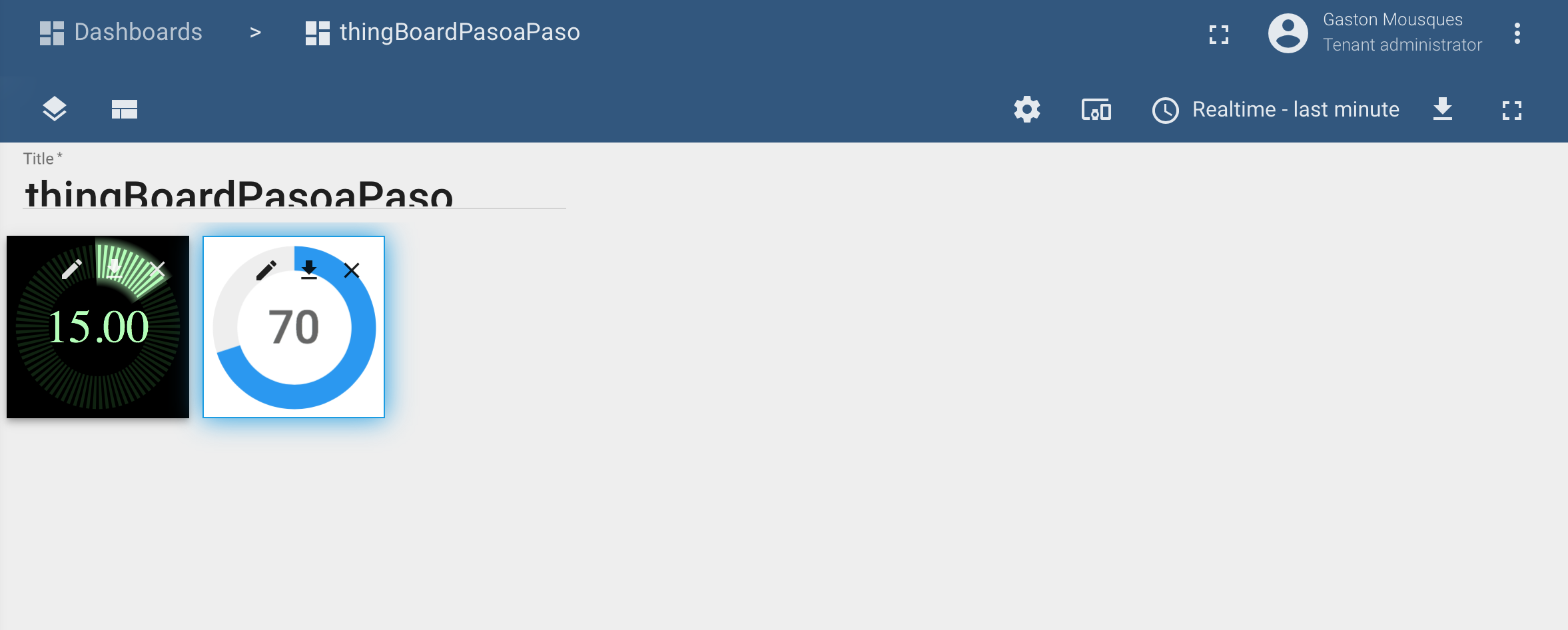
## Crear el dashboard y los widgets

Agregar los widgets a un Dashboard (si no existe crearlo)

La forma fácil de hacerlo es ir al dispositivo a la hoja de LatestTelemetry y seleccionar de la lista los datos que se reciben. Luego seleccionar Show on widget y seleccionar el dashboars

Ir al Dashboard y ajustar los widgets poniendo títulos y otras configuraciones

Se puede agregar nuevos widgets al dashboard



Controlando una puerta

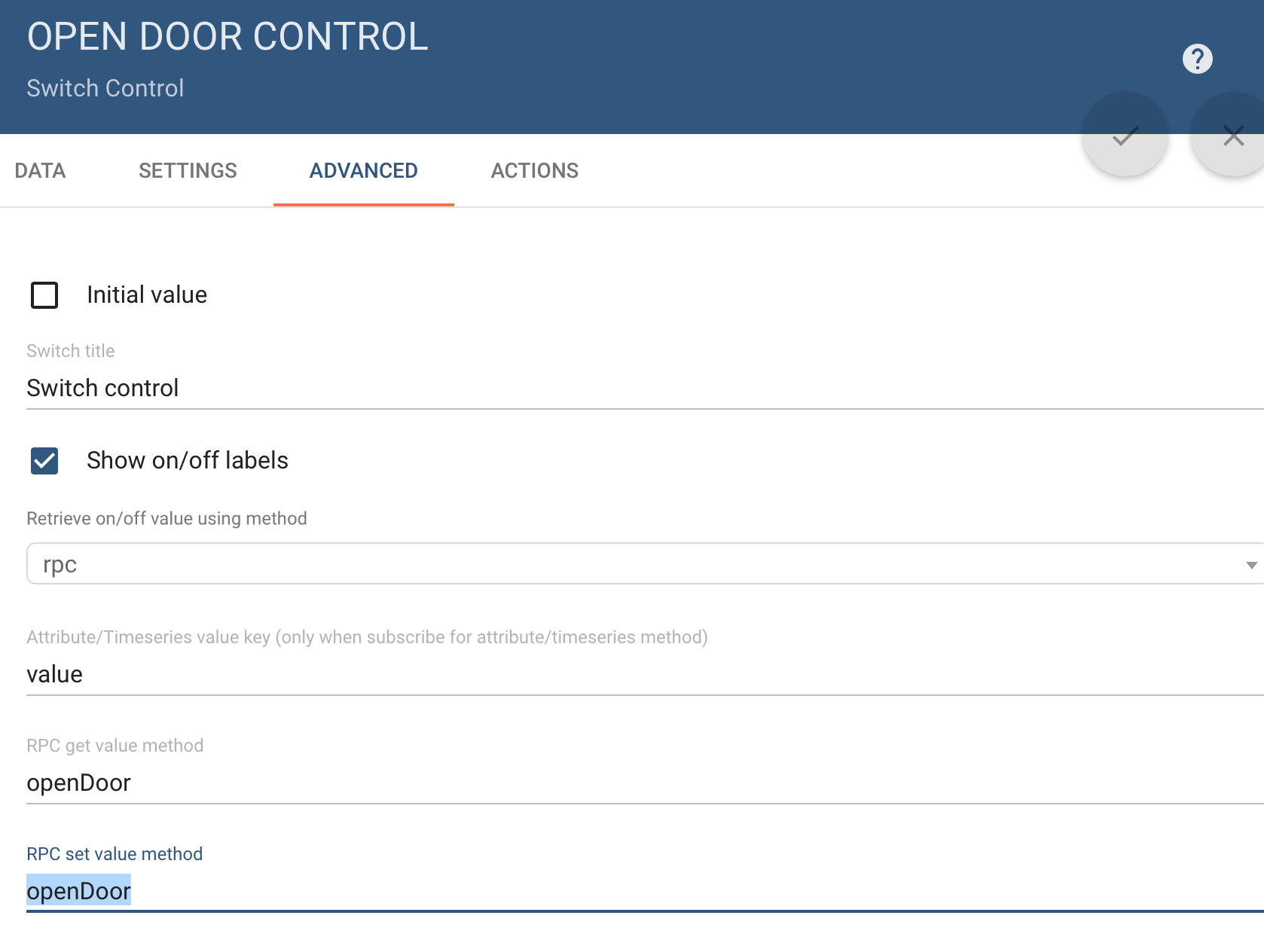
Agregar un widget de control al Dashboard



En el target device poner el dispositivo

En la configuración del Switch en la hoja Advanced

Agregar el nombre del método para RPC setvalue con el comando openDoor que es lo que se enviar al nodo (dispositivo)



## Intercambiar mensajes entre el servidor y el nodo

Para intercambiar información entre el nodo y el servidor se definen en el sketch los siguientes tópicos

/\*definir topicos.

\* telemetry - para enviar datos de los sensores

\* request - para recibir una solicitud y enviar datos

\* attributes - para recibir comandos en baes a atributtos shared definidos en el dispositivo

\*/

char telemetryTopic[] = "v1/devices/me/telemetry";

char requestTopic[] = "v1/devices/me/rpc/request/+"; //RPC - El Servidor usa este topico para enviar rquests, cliente response

char attributesTopic[] = "v1/devices/me/attributes"; // Permite recibir o enviar mensajes dependindo de atributos compartidos

### Enviando telemetría

En el loop se encuentra el método getAndSendData(); que envía la información del sensor DHT11 mediante el tópico **telemetryTopic**

## Recibiendo comandos

En el sketch de Arduino en la función callback (on\_message) se agrega el método que se desea llamar para el método que viene en el tópico.

Primero se parcea el nombre del método que vine en el mensaje.

En función del nombre del método se invoca el método correspondiente en el skectch, En este caso **openDoor** y **rotateMotorValue** (ver código debajo). Luego para se invoca el método que envía la respuesta utilizando el tópico

/\*

\* Este callback se llaman cuando se utilizan widgets de control que envian mensajes por el topico requestTopic

\* en la función de reconnect se realiza la suscripción al topico de request

\*/

void on\_message(const char\* topic, byte\* payload, unsigned int length)

{

……..

…….

// Obtener el nombre del método invocado, esto lo envia el switch de la puerta y el knob del motor que están en el dashboard

String methodName = String((const char\*)data["method"]);

Serial.print("Nombre metodo:");

Serial.println(methodName);

//responder según el método

**if (methodName.equals("openDoor")) {**

bool action = data["params"];

**String doorStatus = openDoor(action);**

// responder al servidor con el estado de la puerta

**replyDoorRequest(doorStatus, topic);**

}

**else if (methodName.equals("rotateMotorValue")) {**

String gradosTemp = (data["params"]);

int grados = gradosTemp.toInt();

// se llama al motor para que gire los grados del parametro

moverMotor(grados);

replyMotorRequest(grados, topic);

}

}

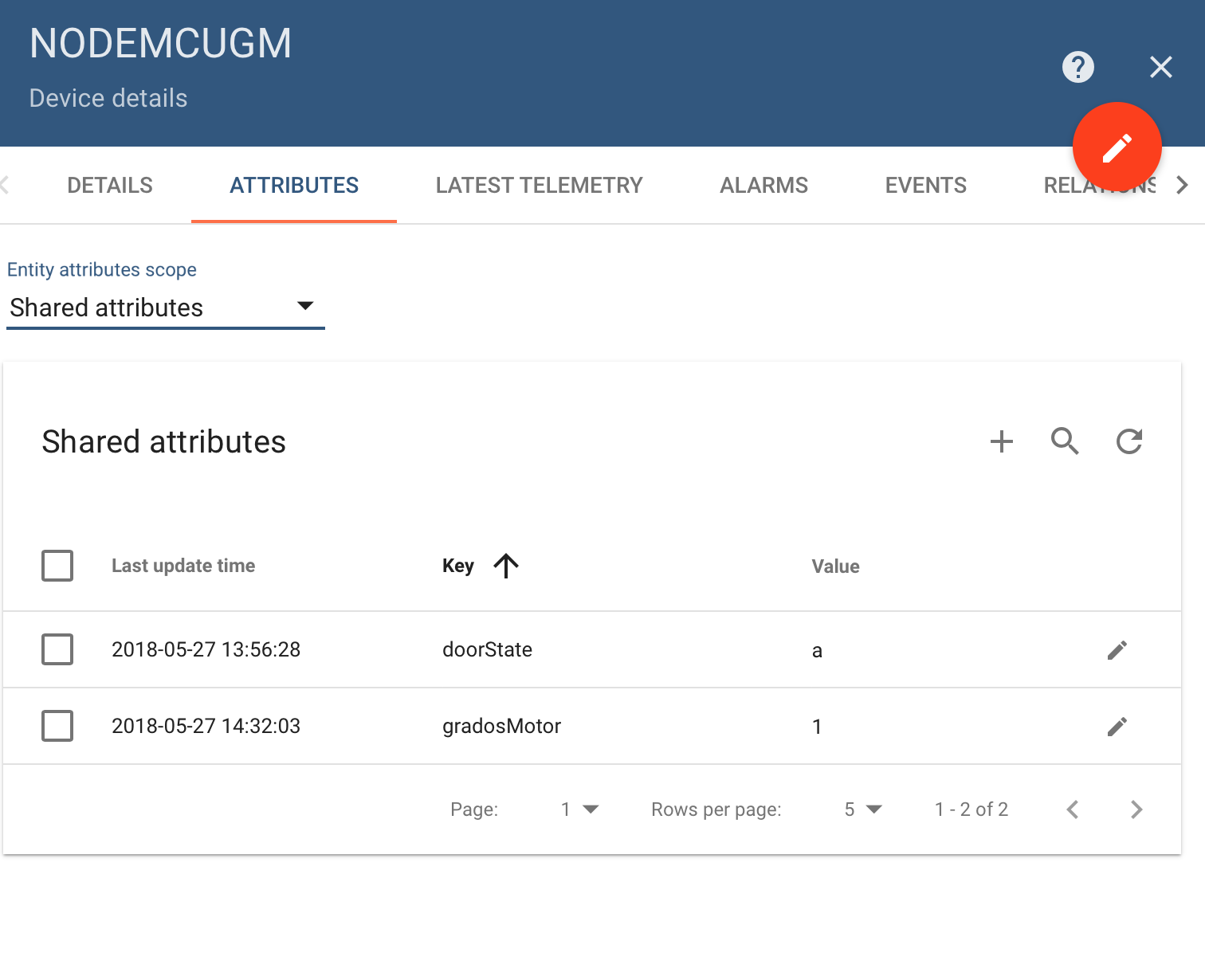
Repetir lo mismo luego de agregar otro widget de control, por ejemplo, en este caso un Knob para el motor y dos tarjetas en las cuales se muestra información enviada por los métodos de respuesta del sketch.



## Cómo intercambiar información con el servidor

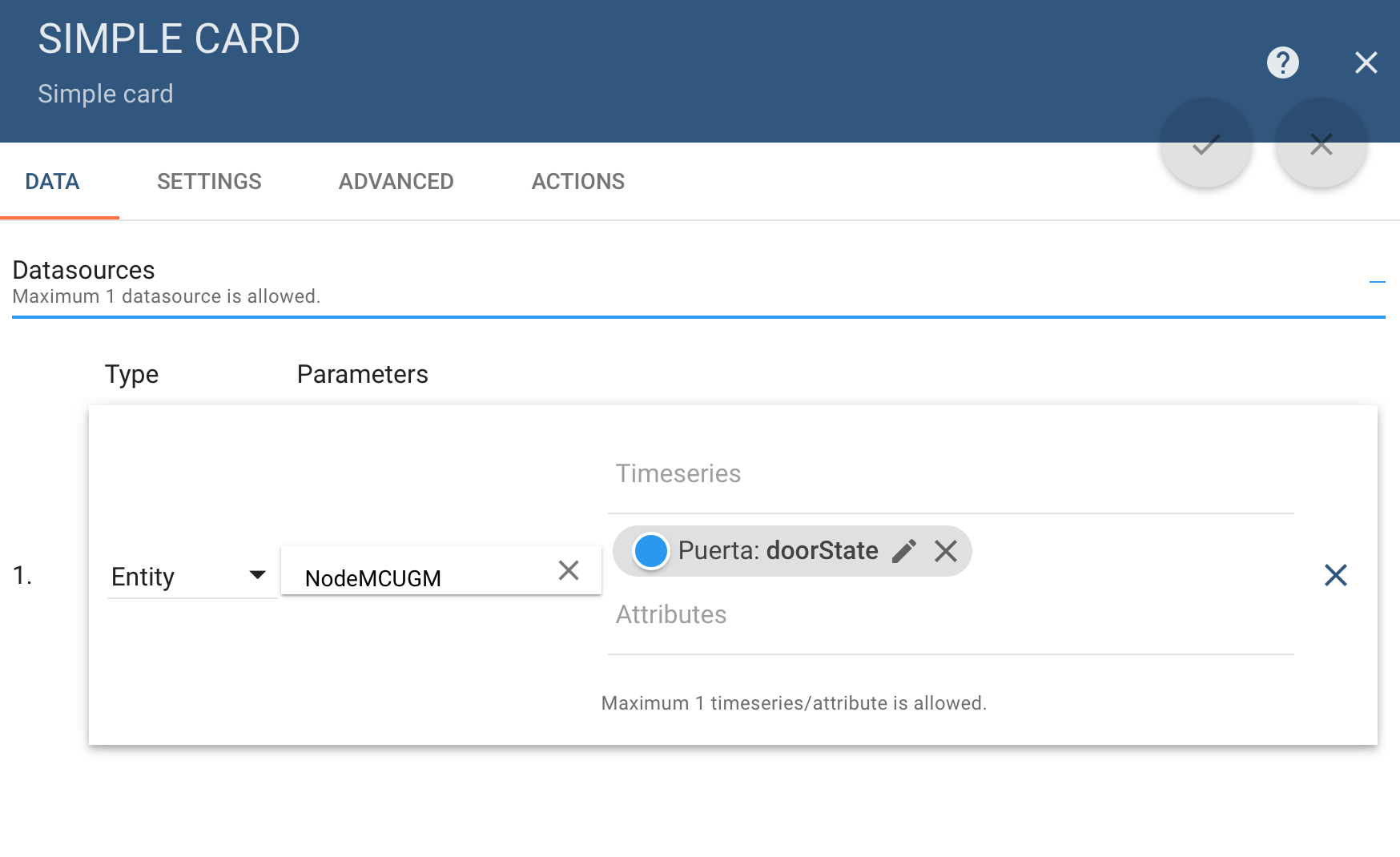
Para poder intercambiar información es necesario definir en el dispositivo Shared Attributes.

En este caso se definieron una para recibir el estado de la puerta y otro para recibir los grados que giro el motor.



Ambos atributos se asociaron a tarjetas en el dashboard. Por ejemplo en el caso de la puerta se asoció el atributo a una tarjeta que refleja los cambios al valor del mismo realizadas por el nodo.

Para hacerlo se configura los datos de la widget Card seleccionado atributos donde aparecerán los definidos en **Shared Attributes** del dispositivo



En este ejemplo cuando el callback envía una respuesta se actualiza el valor de las tarjetas