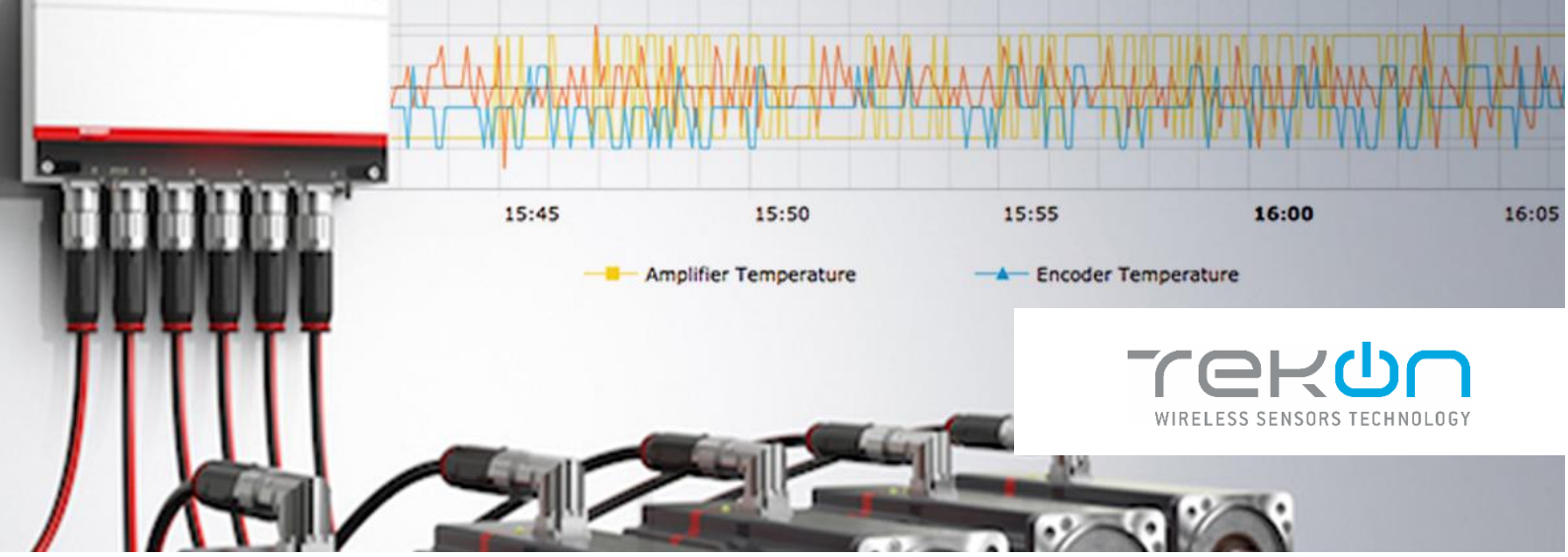

BLOG

INTEGRATION OF TEKON IoT PLATFORM WITH BECKHOFF EMBEDDED PC

The standardization of communications is beneficial for all those looking for the evolution and speed of processes.

With the arrival of the Industry 4.0 concept, the systems that migrate information are increasingly broad and allow connectivity between devices and platforms with the use of protocols that have been standardizing data processing and opening technological frontiers imposed by the limitation of hardware and/or software.

The standardization of communications is beneficial for all those looking for the evolution and speed of processes. The Beckhoff brand is one of the leaders in the automation systems market. TI and automation technologies continue to converge and communications and data to and from the cloud are increasingly implemented in industrial automation projects. Monitoring the industrial process is one of the purposes intended for the Tekon IoT Platform.

The connectivity offered by Beckhoff's solutions, allowed to structure a connection method between your equipment and the Tekon Electronics platform. Communication is implemented using the MQTT messaging protocol, which is based on a publisher / subscriber model, with technical characteristics that are evident in minimizing the occupation of the network and resources to ensure, in a reliable way, the delivery of messages.

Figure 1 - Integration architecture with MQTT in Tekon IoT Platform

The FB_IotMqttClient function block has in its structure a set of inputs, outputs and methods required for the connection and communication with the MQTT broker.

Figure 2 - FB_IotMqttClient function

In the program block, is declared a variable that will be an instance of the function block.

```
fbMqttClient: FB_IotMqttClient;
```

The connection to the Tekon IoT Platform MQTT broker is made through a session that requires credentials, so that the session is validated, and the message can be delivered. The inputs of the FB_IotMqttClient function block are called to make the connection. This validation is reinforced with the use of the DigiCert Global Root CA digital certificate, model PEM. This link has access to the implemented digital certificate.

```
bSetParameter := FALSE;
fbMqttClient.sHostName:= 'TekonIoTPlatfornInstanceLink';  //cloud url
fbMqttClient.nHostPort:= 8883;  // MQTT broker port
fbMqttClient.sClientId:= 'CX-35D594';  //Device name
fbMqttClient.sTopicPrefix:= '';
fbMqttClient.sUserName:= 'UsernameMQTTCredentials';  //Tekon IoT Platform credential
fbMqttClient.sUserPassword:= sPassword;  //Tekon IoT Platform credential (declared in variable)
fbMqttClient.stTLS.sCA := 'C:\Certificates\digicertCA.cer';  //digital certificate location
fbMqttClient.ipMessageQueue := fbMessageQueue;
fbMqttClient.stTLS.bNoServerCertCheck := bAux;
```

The message, in this context, called payload, to be sent is built in a JSON format. The data is organized in a support structure that is later transformed into the JSON format. Using the FB_JsonSaxWriter function block, the payload is built, and the data is organized into the message that will be sent to the Tekon IoT Platform MQTT broker.

```
fbJson.ResetDocument();
fbJson.StartObject();
fbjson.Addkey('timestampMilliseconds');
fbjson.AddUlint(uliUnixTime);
fbJson.Addkey('datasourceApiKey');
fbJson.AddString(datasourceApiKey);
fbJson.Addkey('userApikey');
fbJson.AddString(userApikey);
fbJson.Addkey('variable1');
fbJson.AddBool(Enable);
fbJson.Addkey('variable2');
fbJson.AddBool(Error);
fbJson.Addkey('variable3');
fbJson.AddUdint(ErrorId);
fbJson.Addkey('variable4');
fbJson.AddReal(ActPos);
fbJson.Addkey('variable5');
fbJson.AddReal(ModuloActPos);
fbJson.Addkey('variable6');
fbJson.AddReal(ActVel);
fbJson.Addkey('variable7');
fbJson.AddReal(AmplifierTemperature);
fbJson.Addkey('variable8');
fbJson.AddReal();
fbJson.Addkey('variable9');
fbJson.AddReal(DcLinkVoltage);
fbJson.Addkey('variable10');
fbJson.AddReal(AuxiliaryVoltage);
fbJson.EndObject();
sJsonDoc := fbJson.GetDocument();   //if string has 255 characters
fbJson.CopyDocument(sJsonDoc2, SIZEOF(sJsonDoc2));   //if string has more than 255 characters
```

When the connection to the MQTT broker is established, the MQTT client publishes the message with a topic. Topic validation is done by the broker and the message is sent to the Tekon IoT Platform.

```
IF fbMqttClient.bConnected THEN
        IF TonPublish1.Q OR bPublish THEN   // publish a new payload
                //sPayloadPub := FbJsonTekon_0.sJsonDoc;
                fbMqttClient.Publish(     sTopic:= sTopicPub, pPayload:= ADR(sPayloadPub),
nPayloadSize:= LEN2(ADR(sPayloadPub)), eQoS:= TcIotMqttQos.AtMostOnceDelivery, bRetain:= FALSE, bQueue:= FALSE );
                IF fbMqttClient.bError THEN
                        // add your error logging here
                        hrErrorOccurred := fbMqttClient.hrErrorCode;
                        udiNumberPublishError := udiNumberPublishError + 1;
                ELSE
                        udiNumberPublish := udiNumberPublish + 1;
                END_IF
                bPublish := FALSE;
        END_IF
END_IF
```

In the Tekon IoT Platform instance, a “Generic” datasource was configured, with variables featured to correspond to the type of data they will associate - Bool, Int, Float, etc.

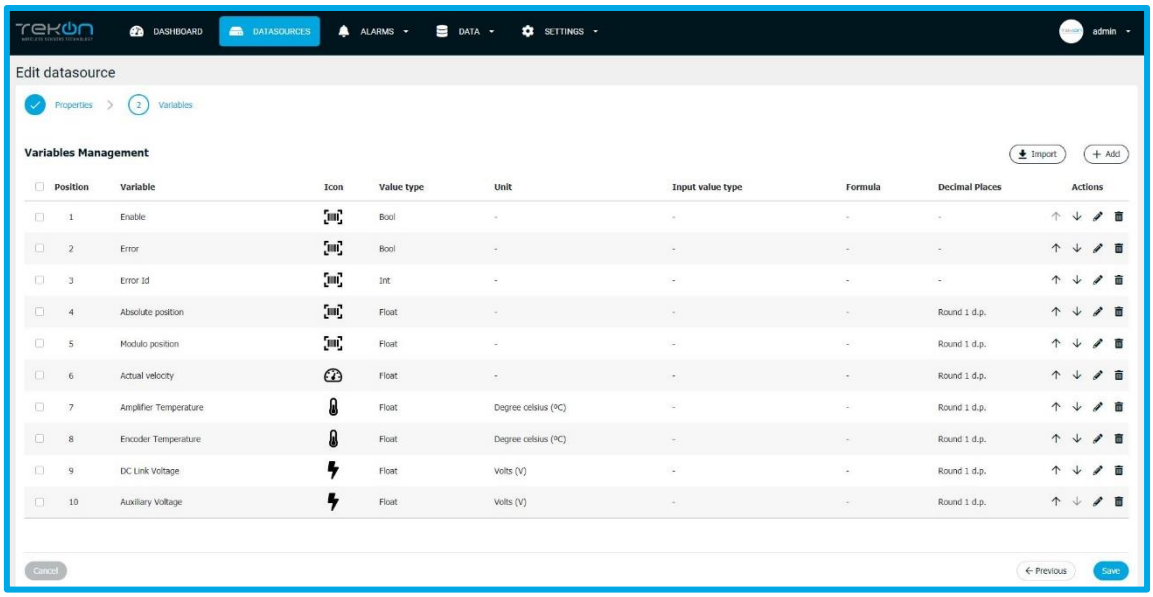

Figure 3 - Datasource variables page

Cloud data storage and the several existing generalist protocols have streamlined communication between different platforms, offering solutions for rapid implementation, with low allocation of resources for the task. The Tekon IoT Platform aims to be an easy-to-use interface, with added value enhanced by the visualization, in real time, of industrial process data.

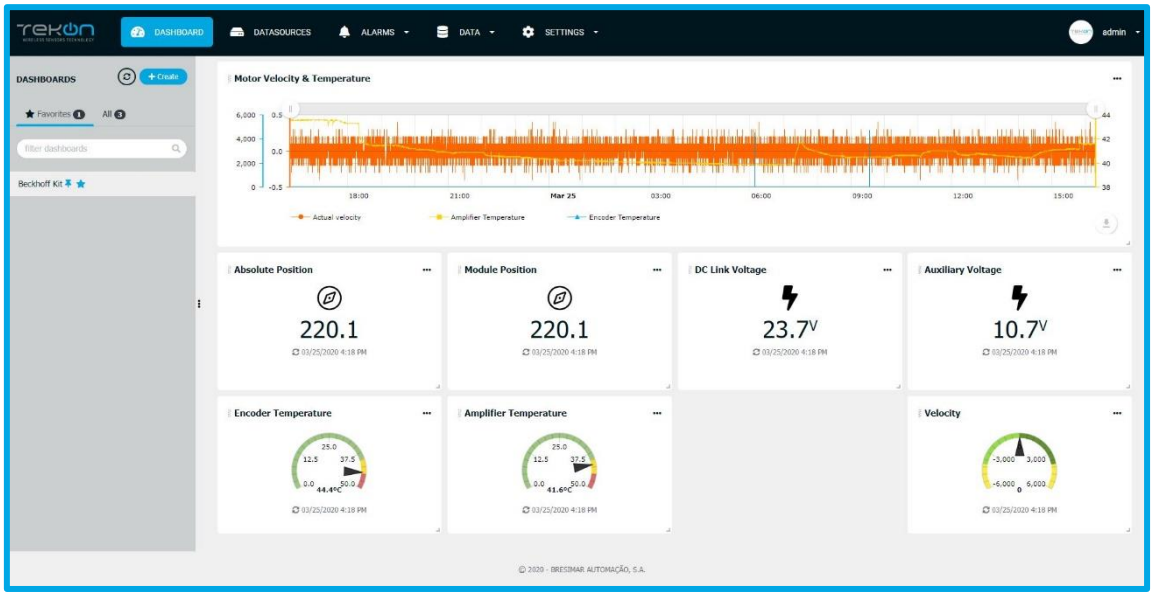

Figure 4 – Real time dashboard with data from Beckhoff equipment

If you need help with the integration of your Beckhoff equipment, via MQTT, on the Tekon IoT Platform, we are available to help you. Contact us via our technical support email.