

# Design of IIoT device to parse data directly to scada systems using LoRa physical layer

Robin Maharaj, Vipin Balyan\*  
and Mohammed Tariq Kahn

Cape Peninsula University of  
Technology, Bellville, Symphony  
way, Cape Town, SA.

\*E-mail: vipin.balyan@rediffmail.com

This paper was edited by  
Subhas Chandra Mukhopadhyay.

Received for publication  
July 27, 2021.

## Abstract

With the advent of the 4th Industrial revolution and the need to improve productivity levels, there is an increasing need for data analytics. Techniques should be pursued to extend the range of sensors both to Local systems and to remote systems. In this paper, the proof of concept is implemented on an STM32F4 development board to realise the performance of two wireless technologies to extend the range of sensor connectivity to a local process control system. The technologies investigated are physical layer LoRa nodes, nrf2401l (a 2.4 GHz radio module), 4 to 20 mA Converters together with protocol wrappers and interfaces to enhance the data value chain in a legacy process control system.

## Keywords

LoRa nodes, NRF2401L at 2.4GHz radio module, Protocol wrappers, 4 to 20 mA converters.

## Introduction

The Internet of things (IoT) made it possible for things to talk or communicate, which brought the virtual worlds into the physical together. In recent times, IoT devices have grown exponentially and will grow at a higher rate in the next decade (Zhou et al., 2016). Industry 4.0 with IoT applications, abbreviated as IIoT, is moving towards automating industry using IoT. A method is proposed in Pagnon (2017) for automation in a factory, from ordering online to shipping the product without any human interference. The work in Langmann and Rojas-Pena (2016) suggests using PLC as a component for Industry 4.0 features in the production domain. In Chen et al. (2017), a hierarchical architecture is proposed using new developments for making smart factories. Due to the innovative functionality of cyber-physical systems, the work in Shrouf et al. (2014) shows the significance of IoT and services provided in our day-to-day life. The work in Guo et al. (2017) proposes a smart factory system that can meet Industry 4.0. The study in Shrouf et al. (2014) provides an overview of IoT based smart factories. Along with it, the mechanism to

increase production system energy efficiency with energy management is introduced.

In Bassi (2017), for Industry 4.0, details on how Laser Processing can be used as direct park marking in smart industries has been explained. Authors in Astarloa et al. (2016) designed a programmable System-on-Chip implemented with an intelligent IP-gateway used to bring new technologies to the IIoT manufacturing sector. The authors in Wollschlaeger et al. (2017) review different technologies like wireless etc, and their effect on the industry. A case study has been done in Yan et al. (2017) on the impact of IIoT on Industrial automation, considering their present state. The authors in Tao et al. (2014), using an example concerning energy management, described the architecture of IIoT.

The 4th Industrial Revolution (4IR) space race is reaching epic proportions, especially in an industrial automation space. Process control engineering is by far not excluded from the innovations and drivers of this revolution, and it is where a lot of research and innovation is taking place to bring expensive propriety systems in line with the 4IR space. There is also an inherent need for supervisory control and

data acquisition (SCADA) systems to integrate with maintenance management systems, enterprise resource planning (ERP) systems, and manufacture execution systems (MES) to present an effective, low-cost data model with improved Mean Time to Repair (MTTR), reduced production cost, etc. This indicates the requirement of integrating capability in legacy systems or the way data are made available to them for adaptation to new systems and requires a minimum cost.

The 4IR is all about data, data consumers, data drivers, data cost, data latency, data integration, data decision making, and the list goes on. The process engineering data are not new; it is the source of data and thus one of the critical data drivers in 4IR.

## Major problems emanating in using IIoT systems

There are essentially two disparate technology systems that seem to be converging in the 4IR automation world. One is the legacy process control systems comprising PLC's human-machine interface (HMI) and SCADA. The other is relatively new IIoT systems, which are mainly microcontroller-based process systems. The architecture of these systems is different, yet on the higher-level data consumer level, they work similarly (Bassi, 2017). The older legacy systems lend themselves to expensive propriety-based solutions. In contrast, the new microcontroller systems offer incredibly cheaper open architecture and more widely used protocol systems, resulting in considerable cost savings. However, one system does not negate the other, and both systems sometimes need to be implemented in the new data consumer-driven world. There is a growing need for systems to augment each other. This means that there is a need for research to get low-level and field-level systems to parse data seamlessly with high-level systems offering greater interoperability and good latency.

To address these issues, the work in this paper:

1. Looks at different field-level devices and interfaces with a communication medium to enhance the seamless interoperability of data between all levels of the data value chain.
2. Increase the sensor range between the process control system and sensor using wireless technologies.
3. Provides a solution for an existing Waste Water treatment plant with a legacy process control system with an existing SCADA and HMI system with a SQL database. This design will get

additional data back to the data warehouse and to a High-level system like ERP/MES system or web service.

## Architecture

A typical modern-day process control system comprises a PLC system with an OPC server, a SCADA system, HMI, and sometimes a SQL database server, PLC or PAC with field devices, actuators, and instruments connected to it. All time-series data acquired from the time series database residing in the SCADA historian can be also be stored in SQL format in its Local SQL database (Calderón Godoy and Pérez, 2018). These data can be made available to high-level applications or web applications.

Figure 1 represents an architecture in which a measuring device was not installed as part of the initial process control system design. One reason could be there is no capacity to include it in the process control system, or the cost of physically adding the section to the process control system is too expensive due to the long distance between the Process Control System and the physical instrument. These devices are situated hundreds of meters away from the process controller or SCADA server on many occasions, like in a wastewater treatment plant.

An inexpensive wireless solution needs to bring this field device data to the SCADA system or a remote MES/ERP system, and this research investigates two possible solutions. To add this component to the system. There is an option to use one of the two wireless IIoT options, namely:

- a. the nRF24L01 2.4 GHz radio module.
- b. LoRa to LoRa node implementing the physical layer.

Both these options will need an IP bridge or some protocol wrapper to get these data back into the SCADA system, which can also be further parsed to the OPC-UA server for further transportation or any other application server for that matter. At the source of the instrument, there would also be a need to convert the 4 to 20 mA signal or 0 to 10 Volt to a data variable which can be parsed to a "LoRa or nRF2401" format.

## 4 to 20 mA to the data interface

Most industrial process control systems like PLCs or PAC connect field devices like sensors and actuators via 4 to 20 mA current loops directly into the process control system.

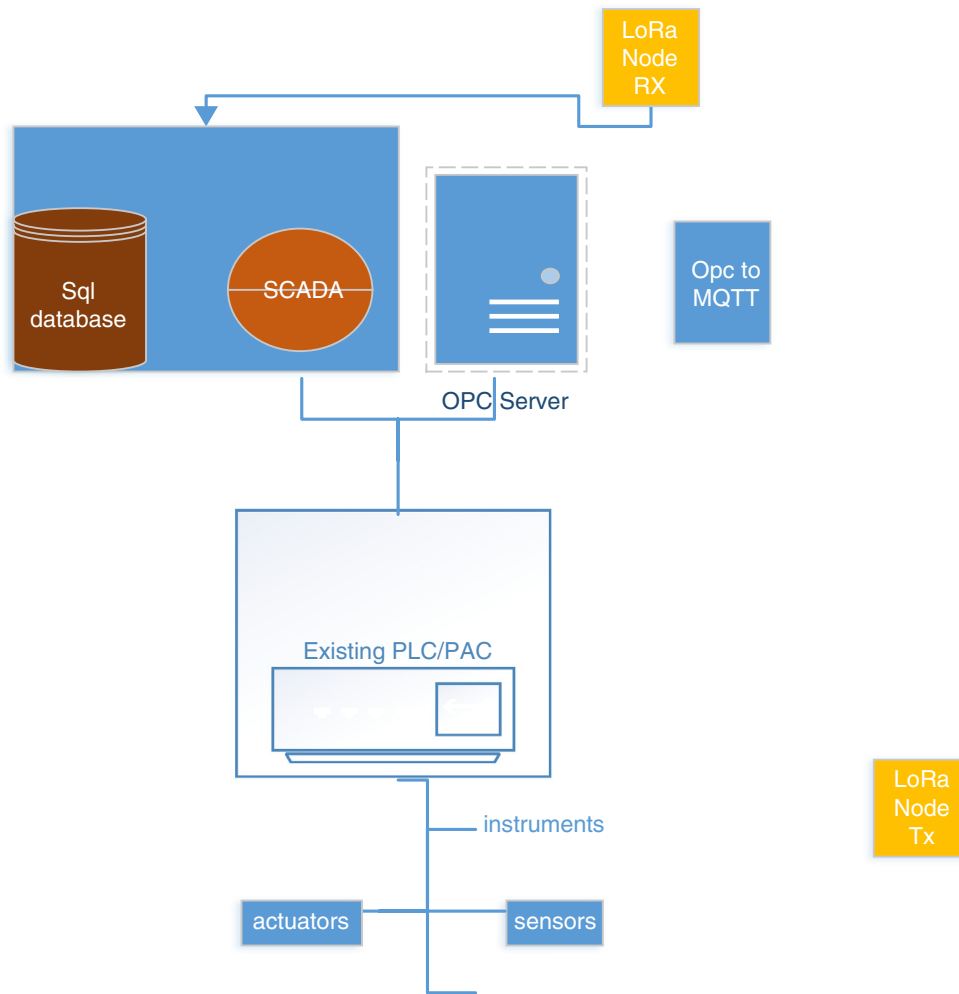


Figure 1: Typical modern-day solution with added IoT LoRa device.

The 4 to 20 mA is a very rugged and robust way of processing signals in the industrial environment and has been found to stand the test of time in the worst industrial conditions. Due to this, it is so much popular in the industry today. The 4 to 20 mA current loop is one of the most common sensor signalling standards (Azhari and Kaabi, 2001).

Current loops are ideal for signal transmission because of their inherent insensitivity to electrical noise. In a 4 to 20 mA current loop, all signalling current flows through all components; the same current flows even if the wire terminations are not perfect. All components in the loop drop voltage due to the signalling current flowing through them. The signalling current is not affected by these voltage drops as long as the power supply voltage is greater than the sum of the voltage drops around the loop at the maximum signalling current of 20 mA.

The operation of the 4 to 20 mA loop is straightforward: a sensor's output voltage is first converted to a proportional current, with 4 mA representing the sensor's zero-level output, and 20 mA representing the sensor's full-scale output (Surachman et al., 2011).

Converting/translating the 4 to 20 mA current loop to a voltage signal is necessary as in the IoT space most universal IoT controllers are microcontroller-based systems either capable of processing 0 to 5 V or 0 to 3.3 V by its ADC controller peripheral. This design implemented a mikroE 4 to 20 mA data interface connected to the microcontroller via the micro's serial peripheral interface (SPI). The modules that form this 4 to 20 mA interface are the INA196 current shunt monitor, MCP3201 12-bit ADC, and TPS61041 DC/DC boost converter. It receives the output current (4–20 mA) from the transmitter and converts it into a voltage (0.4–2 V). Then through the

ADC sends the signal to the built-in microcontroller. This built-in microcontroller then presents these data via an SPI interface to other devices that require these data.

## LoRa to LoRa node implementing the physical layer

### The LoRa transmitter

The LoRa transmitter, which consists of the LoRa radio module, 4 to 20 mA interface, and STM32f4 development board, essentially converts the 4 to 20 mA signal into a variable which is then parsed to a buffer in the microcontroller. These data are parsed from this buffer to the payload of the LoRa radio module to be transmitted over the air. The LoRa radio module is interfaced to the microcontroller via SPI as seen in Figure 2.

### The LoRa receiver

The LoRa receiver, which consists of the LoRa radio module, IP bridge (ENC28j60), and STM32f4 development board, decodes the data payload from the LoRa modulated signal and parses these data to the microcontroller. These data are then parsed to a buffer in the microcontroller interfaced to the LoRa radio module via SPI. These data are then translated to an IP packet by the ENC28j60 module. This module is connected to a packet manager application on the local network.

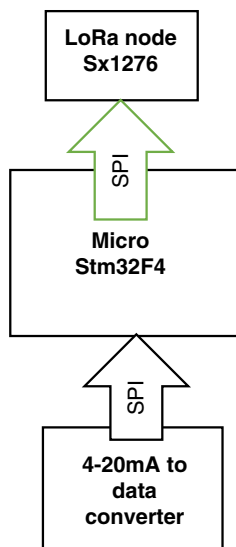


Figure 2: LoRa transmitter module.

## Phase 1: Development phase using STM Nucleo board with LoRa radio

The first phase of the design, which implements the proof of concept where the purpose was to test the LoRa Physical layer. Dummy data were transferred from the  $T_x$  buffer of the microcontroller (STM Nucleo board) and parsed to the LoRa radio module for transmission. The received LoRa CSS signal is received by the LoRa module, parsing the data into the  $R_x$  buffer in the microcontroller (STM32f4 discovery board). These data are visualised by Putty software via UART of the microcontroller.

## Phase 2: implementation of design using the B-L072Z development board

In phase 2 of the development, the solution is implemented on a different development board (B-L072Z running STM32L072CZY6TR MCU), including a built-in LoRa radio module (SX1276) already interfaced with the microcontroller, which made working with the development board more convenient and less cumbersome. Only the 4 to 20 mA *mikroe* boards needed to be interfaced to the board in case of the transmitter node and the ENC28J60 Ethernet module had to be interfaced to the receiver node.

## The NRF24L01 module

The nRF24L01 transceiver module operates in a 2.4 GHz worldwide ISM Band. This module uses GFSK modulation for data transmission with data transfer rates of either 250kbps, 1Mbps, or 2Mbps. The module supports a programmable output power of 0dbm to -18dbm. There are also different variations of these modules available in the market today. The differences of these modules lie in the with/without power amplification option, which can be programmed via the SPI interface, the other variation is also with/without low noise amplifier (LNA), and the available antenna is attached to the module, which also has a significant effect on the range of the module. The module is rated to consume 12mA at 0 dBm, making it ideal for low-power applications (Nordic, 2008). The transceiver also easily interfaces with microcontrollers using an SPI interface at a data rate of up to 10Mbps. It should also be noted that varying the data rate also impacts the range of this module. During the implementation of this research, the range was a determining factor, and therefore, 250 kbps were used.

The nRF24L01 was easily implemented using STM32F4 microcontroller devices and the interfacing

between the nRF24L01 radio module and the micro was accomplished using the SPI serial interface (Langmann and Rojas-Pena, 2016).

## Differences between LoRa WAN and LoRa PHY

It is imperative to understand the differences between LoRa Phy over LoRa WAN. In LoRa WAN, the end node communicates with a gateway where the end node is already registered using its device EUI, app EUI to differentiate itself from the end nodes and authenticate itself by the gateway and the app server either via OOTA or APB methods (Orange, 2016).

With LoRa Phy, the LoRa end node communicates directly on one-to-one basis with another end node without any authentication by either node. It is merely a LoRa payload transfer from one node to the other using the LoRa radio like RM95, SX1276, SX1278, etc.

## LoRa PHY

The LoRa®-protocol defines the link layer (Shrouf et al., 2014), describing how the data are modulated into the transmitted signal. The protocol is proprietary, and not all information is freely available. Nevertheless, it is enough to know that due to the Chirp Spread Spectrum (CSS), the power consumption does not depend on the content of the data. Therefore, the energy required for transmission only depends on the so-called time to air (ToA) and the output power. Time on air refers to the time it takes a signal to travel from the transmitter to the receiver.

LoRa uses orthogonal spreading factors. This allows the network to preserve the battery life of connected end nodes by optimising an individual end node's power level and data rate. This enhances LoRa footprint in the LPWAN technology when compared to its counterparts.

An end device located close to a gateway should transmit data at a low spreading factor since very little link budget is needed. However, an end device located several miles from a gateway will need to transmit with a much higher spreading factor. This relates to LoRa adaptive features. LoRa is a long-range radio at fairly low bit rates. LoRa is also very low bandwidth and consumes very little power, which makes it ideal for battery-powered operations.

LoRa operates in the license-free band. Either 433, 868, or 915 MHz LoRa has a substantial link budget. LoRa PHY is the best-effort transmission and, analogous to UDP, doesn't care if it gets there. LoRa WAN works on the layer on top of LoRa PHY knows

that the data get there like TCP. LoRa WAN also adds encryption, whereas LoRa node/LoRa PHY has no encryption. The payload of the standardised packet is fixed to 10 bytes, being close to the proposed 12 bytes maximum of TTN. For a package sent via the LPWAN, the payload increases by at least 13 bytes to a total length of 23 bytes.

## LoRa packet format

The basic LoRa packet comprises three elements: the preamble, an optional header, and payload. There are essentially two types of LoRa packet transmission: explicit header mode and implicit header mode (Koon, 2020). The functional difference between these two modes is the presence or absence of the header, which contains information regarding the payload length, the coding rate, and the presence of 16 bits' cyclic redundancy check. In the implicit mode where the header is absent, the header information must be manually configured on both sides of the radio link.

The robustness of the physical layer can be attributed to one significant parameter of the LoRa Physical Layer, i.e., forward error correction (FEC). FEC is where redundant bits are added to the transmitted data. These redundant bits help restore the data when it gets corrupted by interference. The higher the number of error correction bits added, the easier it is to correct the data. However, by doing this, the payload inherently increases, which increases the power consumption in the LPWAN transmission, adversely affecting battery life. The physical layer/radio interface provides an excellent conduit for low-power signal communications where a low data rate, low power, and long distant range is necessary (Bäumker et al., 2019).

The interfacing of the LoRa radio module to the microcontroller is easily interfaced to the microcontroller with the SPI interface as seen in Figure 3. In the modern era, rapid development tools like the ST CUBEmax make this interface quite simple to the seasoned low-level C-programmer. There are also many open source libraries available to adapt and to use to interface the LoRa radio with the microcontroller.

The node-to-to-node implementation, on the other hand, does not use a gateway and data are transmitted directly to a LoRa node which is essentially a transceiver. There are many off-the-shelf implementations of the LoRa PHY, but this paper implements using the SX1276 LoRa radio module, open-source libraries for the SX1278 module are also adaptable to use with SX1276. When implementing the code for the microcontroller to set up and



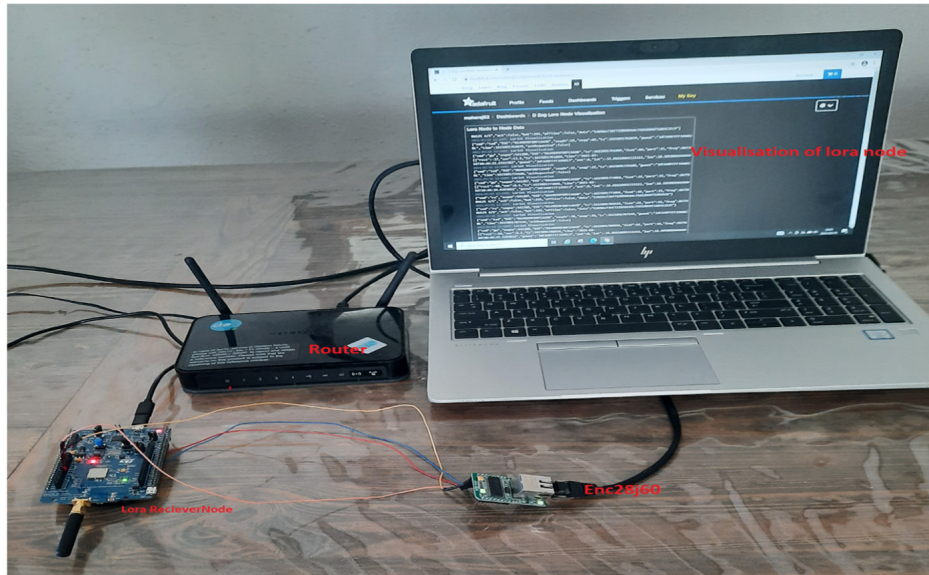


Figure 3: Phase 2 receiver module with Enc28j60 module (LoRa with Ip wrapper).

implement the LoRa node, careful consideration must be given to comply with the guidelines/rules of the LoRa alliance. The interface between the LoRa radio module and the microcontroller is SPI (Qfix, 2012).

## IP wrapper/bridge

On the receiving end of the LoRa device, the data would then be parsed to the SCADA via a LoRa to IP wrapper. The LoRa to IP wrapper/bridge is merely a software bridge from LoRa PHY format to IP parsing data in the required format for the SCADA device driver to handle (Qfix, 2012). The serialisation of the data from LoRa to IP would be carried out by the microcontroller, which in this case is the STM32 device interfaced with an ENC28J60 Ethernet module. The output of this Ethernet module is UDP.

The bridge is simply a high-performance device to connect two dissimilar or similar networks. In this research, our specific need is to connect/translate the format of the LoRa message to the IP format for passing the acquired data to the SCADA system. Once the data are presented to the SCADA system with the familiar format that the SCADA device driver can recognise, it can be further deserialised/translated by the SCADA device driver to be processed by the SCADA system (Tel, 2012). The device driver of the SCADA system (Ignition in this research) will connect to one or more ports with a given IP address and bring in any data as a value of a tag. In this research, we will parse the data as UDP

data, and thus the device driver of the ignition SCADA system will only listen/read the data from the LoRa device via the IP/LoRa bridge shown in Figure 4.

Both LoRa and Ethernet have different frame formats and practice other MAC mechanisms and routing algorithms. Because we are using the LoRa node-to-node implementation (LoRa physical layer and not LORAWAN communication, there are no routing algorithms used as opposed to if we had to implement LORAWAN to IP bridge. The embedded code needs to take care of CRC issues, arbitration,

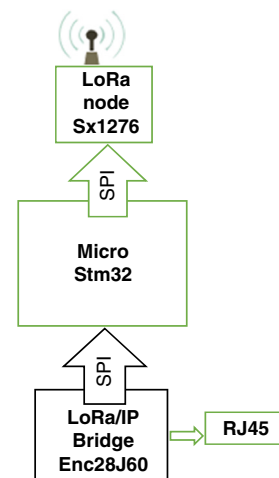


Figure 4: LoRa node receiver with Ip/LoRa bridge.

handling of errors, etc. As well as compliance with the LoRa specifications (D. Resources and D. Features, 2015).

The LoRa transceiver receives the LoRa format data from the transmitting LoRa device node. Both the LoRa radio modules SX1276 are interfaced to the microcontroller via SPI peripheral interface (D. Resources and D. Features, 2015). The low-level drivers/libraries move data either from the microcontroller buffer to the radio module in the case of the transmitter or from the radio module to the microcontroller buffer in the case of the receiver radio module. The data that are parsed to the microcontroller radio buffer is then encapsulated/serialised into an IP/UDP format by the LoRa to IP bridge.

The queued datagrams received from the LoRa receiver are presented as an IP/UDP frame, using the LoRa device as the unique network address and a dedicated SCADA port number. This, however, is not a direct translation from LoRa data buffer to IP. Still, initially, the data get serialised from the LoRa module's buffer of the LoRa module by the microcontroller and parses this serialised data to the Tx buffer of the ENC28J60 controller. The ENC28J60 controller (Figure 5 shows the physical circuit of the Ethernet controller) is interfaced to the microcontroller via the SPI. When data are available in the buffer of the microcontroller, the microcontroller initiates (via an interrupt routine) the transfer of these data via SPI commands to the TX buffer of the ENC28J60 controller (see Figure 6). The controller then adds the MAC and PHY components to the data to implement an IP data frame. There are numerous low-level

functions written to carry out this implementation and can be summarised as follows.

1. Initialisation of the ENC28J60.
2. Assigning a MAC address to the controller.
3. Writing to the  $T_x$  buffer.
4. Sending packets.

## Testing and implementing the IP bridge

Before implementing the IP wrapper part of the design process, dummy data need to be sent to the SCADA to understand how it would resolve the data and to be able to write serialisation routines for the IP/LoRa bridge for implementation. This is done to get an indication of how to serialise the data at the LoRa node. To achieve this, we sent the data via Java-based socket application program and sent data in the delimited format that the bridge would implement parsing the delimited data and a dedicated port number that we configure the SCADA agent server to accept UDP data via this port number.

During the testing of the wrapper, we found a lot of error-prone packets arriving at the SCADA system. There was a need to implement a packet manager or a buffer of sorts to manage the packets arriving at the SCADA system. To accomplish this, it was necessary to instantiate a packet manager and adapt to the LoRa/IP bridge. The packets leaving the bridge were configured to be destined for an assigned port number. Before passing the UDP packets to

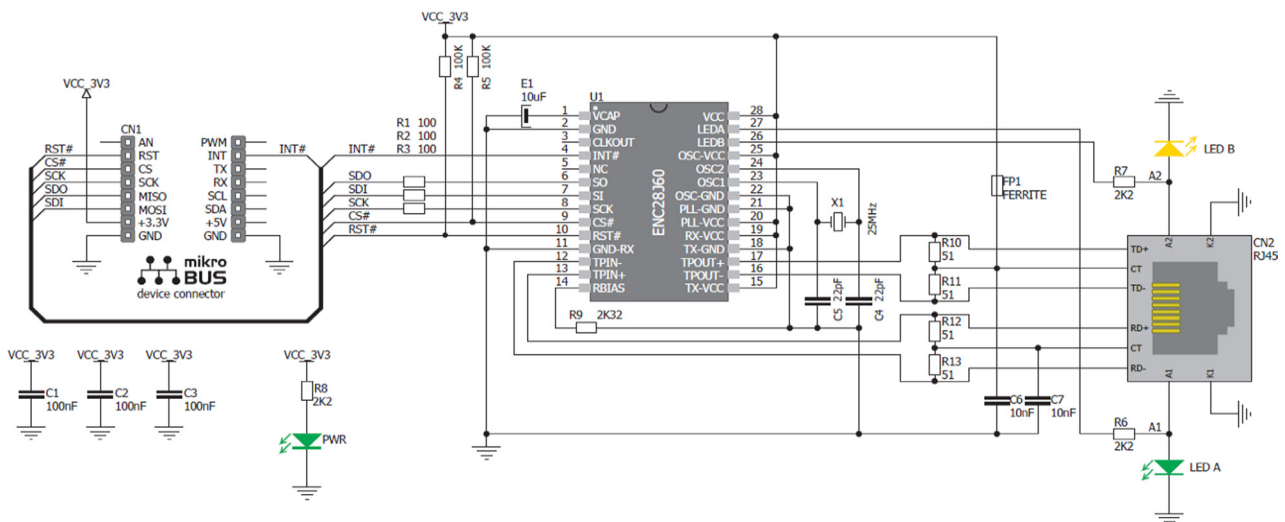


Figure 5: Enc28j60 interface is used to connect loRa receivers to scada servers.

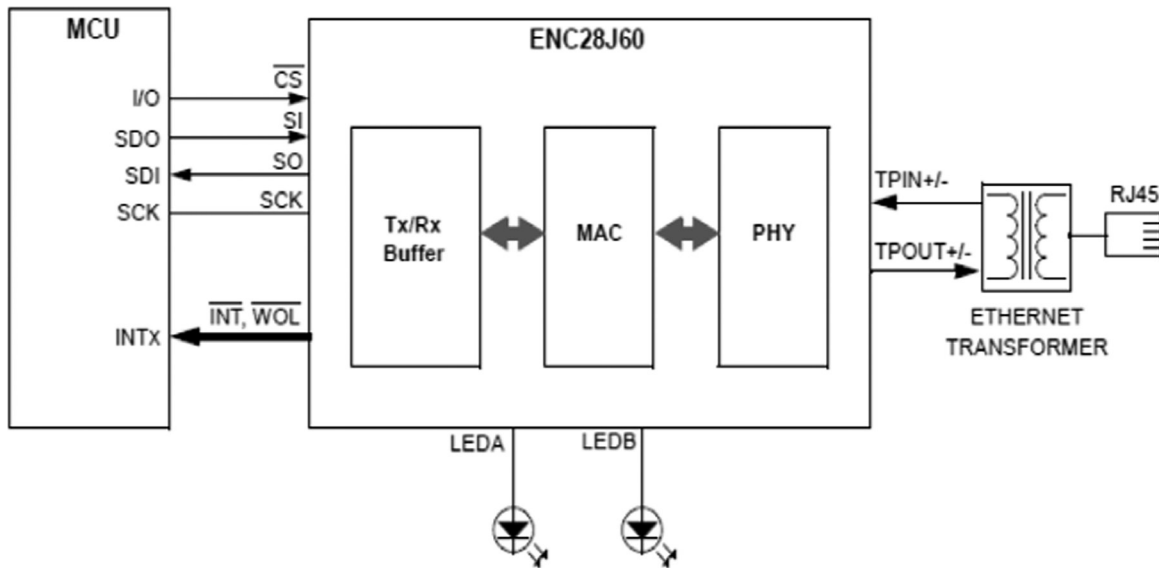


Figure 6: Device architecture of the Enc28j60.

the SCADA, it is necessary to have implemented a packet manager/buffer. The purpose of this packet manager/buffer is to parse only packets in a certain format to the SCADA while discarding any non-compliant packets. This packet manager takes care of processes that UDP doesn't handle. It also serves as a serialisation routine that formats the data in the format that the SCADA driver can handle/process. Figure 7 is the capture of packets after the packet manager.

## Comparing the NRF24L01 and the LoRa PHY node

By researching different wireless platforms to transmit process data from plants, the two technologies were realised as the most appropriate and could provide a viable solution for getting data from standalone industrial instruments back into the processing system or SCADA system for local visualisation of data or remote visualisation and/or input into MES or ERP systems.

There were other technologies researched to make an informed decision to implement an appropriate wireless solution to transmit the acquired data from the field device to the SCADA system, but two wireless technologies that stood out in terms of robustness, bandwidth was NRF24L01 and LoRa node-to-to-node communication (LoRa PHY).

After researching and physically implementing the two technologies from an embedded perspective,

interfacing both the LoRa and the NRF24L01 modules to an STM24F4 development board, it was discovered:

1. The NRF24L01 has far better data rate and bandwidth than the LoRa node, but it is significantly poorer in SNR and link budget and the distance it can cover for end-to-end wireless data transmission.
2. The NRF24L01 module is significantly inferior concerning range as it could only achieve a range of up to 120 meters without degradation of data. In contrast, LoRa PHY achieved a range greater than 700 meters.
3. LoRa, on the other hand, is very flexible in sending different data rates depending on the surrounding transmission conditions. It also has a built-in data rate adapter built on to LoRa physical layer. (This relates to the spreading factor and FEC setting). LoRa also shows considerable superiority in terms of the SNR compared to the NRF24L01 radio.
4. The NRF24L01 module has superiority in data rate, but the NRF is significantly poorer in the range when compared to the LoRa module.
5. This significant superiority in SNR robustness of LoRa can be attributed to the propriety signal transmission Chirp Spread Spectrum (CSS) that LoRa PHY uses.

Once received from the wireless interface, data need to be converted by the IP bridge to parse





```

OutBoundDgm 2020-09-02 UDP 46400.txt - Notepad
File Edit Format View Help
2020-09-02 00:00:04|2020-09-01 23:59:56|2020-09-01 23:59:56|ADROITSERVER|IPUDP|10.243.74.85 46400|189|10.243.74.85 46400|DATA|CCTPS|41|
2137303031303130323330373130397836322352303835414C414335333133323735312B33460D0A|
2020-09-02 00:00:04|2020-09-01 23:59:57|2020-09-01 23:59:57|ADROITSERVER|IPUDP|10.243.75.8 46400|190|10.243.75.8 46400|DATA|CCTPS|44|
213230303930313233333832373037337836382352323632414C444339313034303038453030312B37340D0A|
2020-09-02 00:00:04|2020-09-01 23:59:57|2020-09-01 23:59:57|ADROITSERVER|IPUDP|10.243.75.84 46400|191|10.243.75.84 46400|DATA|CCTPS|167|
213230303930313233333633363030367836342352333338414C444339313138313431323030302B30440D0A21323030393031323333363337303636783633235233338414C414335333431303538302
33633373034367836312352333338414C414335333234323045422B33410D0A21323030393031323333363337303636783633235233338414C414335333431303538302
2020-09-02 00:00:04|2020-09-01 23:59:58|2020-09-01 23:59:58|ADROITSERVER|IPUDP|10.243.75.84 46400|192|10.243.75.84 46400|DATA|CCTPS|164|
213230303930313233333633373039317836422352333338414C414335333433303931342B33320D0A21323030393031323333363337313036783634235233338414C41
73132367836362352333338414C414335333532314442352B33430D0A21323030393031323333363337313437783631235233338414C414335343031313135382B33320
2020-09-02 00:00:04|2020-09-01 23:59:59|2020-09-01 23:59:59|ADROITSERVER|IPUDP|10.243.74.92 46400|193|10.243.74.92 46400|DATA|CCTPS|41|
213230303930313137313434333435397836462352303932414C41433533303331436352B34410D0A|
2020-09-02 00:00:04|2020-09-02 00:00:00|2020-09-02 00:00:00|ADROITSERVER|IPUDP|10.243.75.112 46400|194|10.243.75.112 46400|DATA|CCTPS|41|
213230303930313233333530313531317836302352333636414C414335333031334444442B34340D0A|
2020-09-02 00:00:04|2020-09-02 00:00:00|2020-09-02 00:00:00|ADROITSERVER|IPUDP|10.243.74.87 46400|195|10.243.74.87 46400|DATA|CCTPS|82|
21323030393031323334333036343236783632352303837414C414335333033313445352B34380D0A213230303930313233343330363532367836322352303837414C41
2020-09-02 00:00:04|2020-09-02 00:00:01|2020-09-02 00:00:01|ADROITSERVER|IPUDP|10.243.76.57 46400|196|10.243.76.57 46400|DATA|CCTPS|44|
213730303130313133323930383930377836312352353636414C444339313034303038453030312B37370D0A|
2020-09-02 00:00:04|2020-09-02 00:00:01|2020-09-02 00:00:01|ADROITSERVER|IPUDP|10.243.75.31 46400|197|10.243.75.31 46400|DATA|CCTPS|44|
213230303930313233333831313732397836352352323835414C444339313034303041453030312B30340D0A|
2020-09-02 00:00:04|2020-09-02 00:00:01|2020-09-02 00:00:01|ADROITSERVER|IPUDP|10.243.75.112 46400|198|10.243.75.112 46400|DATA|CCTPS|41|
213230303930313233333530333131317836362352333636414C414335333031344342372B33310D0A|
2020-09-02 00:00:04|2020-09-02 00:00:01|2020-09-02 00:00:01|ADROITSERVER|IPUDP|10.243.74.87 46400|199|10.243.74.87 46400|DATA|CCTPS|41|
213230303930313233343330373232367836342352303837414C414335333033323945452B33360D0A|
2020-09-02 00:00:04|2020-09-02 00:00:02|2020-09-02 00:00:02|ADROITSERVER|IPUDP|10.243.75.127 46400|200|10.243.75.127 46400|DATA|CCTPS|41|
213230303930313233333434333735307836302352333831414C414335333032333137452B34390D0A|
2020-09-02 00:00:04|2020-09-02 00:00:02|2020-09-02 00:00:02|ADROITSERVER|IPUDP|10.243.74.77 46400|201|10.243.74.77 46400|DATA|CCTPS|44|
213230303930313233343331363530377836302352303737414C444339313034303141383030312B37370D0A|
2020-09-02 00:00:12|2020-09-02 00:00:04|2020-09-02 00:00:04|ADROITSERVER|IPUDP|10.243.74.87 46400|202|10.243.74.87 46400|DATA|CCTPS|123|

```

Figure 7: Snapshot of outbound packets from packet manager to scada.

data to the SCADA device driver/dll. LoRa node has significant bandwidth restriction but can be overcome or augmented by techniques like bit encoding to optimise data transmission. LoRa node/LoRa PHY can be likened to UDP transmission and LoRa WAN to TCP. This means some form of CRC needs to be implemented to ensure proper data integrity if this is at all a priority, especially when using just LoRa Phy.

The SCADA system can also store this time series database (historian), which can then be stored in a SQL database that can be used by high-level systems like an MES/ ERP system. Now that these data are part of the SCADA data architecture, the SCADA system can also parse these data to the OPC server. The OPC UA server makes these data available for remote ERP/MES clients. These data can be further converted to a lightweight protocol like message queuing telemetry transport (MQTT) in the publisher/ subscriber module for a cloud-based web solution.

The primary objective of an open platform communications Unified Architecture (OPC UA) server is to expose information that clients can use to manage an underlying real-time process. OPC UA is a non-agnostic architecture that is employed for seamless data interoperability solutions. This OPC UA technology of exposing data to high-level systems is fast gaining ground.

Information or data collected describes the process's state and behaviour, and the OPC server can transfer these data in both directions. Both the wireless options will need an IP bridge to get these data back into the SCADA system, which can also be passed to the OPC UA server for further transportation (Calderón Godoy and Pérez, 2018). At the source of the instrument, there would be a need to convert the 4 to 20 mA signal to serial data format.

The LoRa to IP bridge can also have more than one level of implementation to achieve data portability. The

data acquired by the LoRa device or the NRF24L01 module needs to be parsed to the SCADA system. For seamless portability in line with 4IR specifications, it is important to parse the data in a format like IP or any other common format. Depending on the level of integration, either a low-level implementation using the ENC28J60 or the W5500 which is an alternative implementation of this converter/wrapper. Many SCADA systems do have driver libraries to parse data from other vendors using their device drivers. In this research, the Ignition SCADA system integrated the data acquired from the LoRa node.

The principle of operation of this LoRa to IP bridge is pretty straightforward. The ENC28J60 controller has an industry-standard SPI interface that can be easily interfaced with a modern microcontroller. The  $T_x/R_x$  buffer is the data that need to be parsed from the microcontroller, these are the data that need to be transformed to an IP frame.

### Data value chain from LoRa node $T_x$ To Ignition Scada server

Measurements recorded by the LoRa Node TX are partitioned into small elements—called datagrams.

The field device datagrams received by the device driver of the SCADA system are used by the SCADA engine can be visualised by the SCADA system (Figure 8). This translation of the data presented to the SCADA system is translated by the device driver of the SCADA system using the custom UDP device driver.

- Each datagram is forwarded to the LoRa Node  $R_x$  via a LoRa physical layer wireless link.
- The LoRa Node  $R_x$  forwards the datagrams to a packet manager (Data switch) using the packet data format.
- The datagrams are received by a data switch on the local data server network.
- The data switch pushes the datagrams to the Ignition Master server, also located on the local data server network.

The data transfer process through the network is explained by discussing each link below.

### 4 to 20 Ma interface and LoRa $T_x$ Link

Data transfer between 4 and 20 mA and the LoRa  $T_x$  is over an SPI peripheral of the microcontroller.

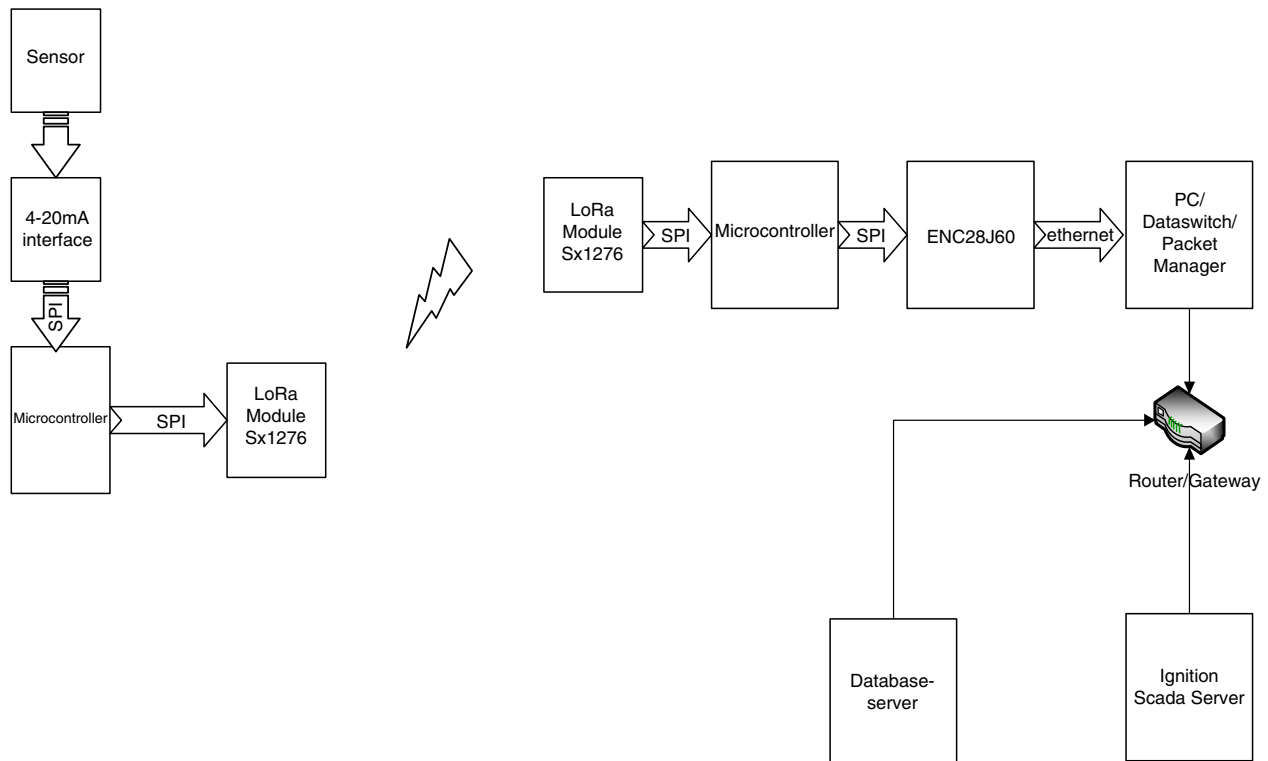


Figure 8: Complete data value chain.

A timeout is used to determine the start and end of each frame (datagram). The LoRa  $T_x$  buffer has a queue that can store up to four datagrams (to be forwarded to the LoRa Rx Node). There is no handshaking between the 4 and 20 mA interface's buffer and the LoRa Tx Node to control the flow of datagrams. The latter will be silently discarded if the interface transmits more datagrams than the LoRa TX can transmit.

### LoRa node $T_x$ –LoRa node, $R_x$ wireless communication

The data between the LoRa Node  $T_x$  and LoRa Node  $R_x$  are transmitted using the propriety CSS. Each of the respective nodes is configured to serve either as the LoRa Transmitter module or the LoRa receiver module. The LoRa modules are the interface to the microcontroller via SPI. There are many open-source libraries for the interface between specific microcontrollers and LoRa radio modules. The libraries used in this research with STM32F4 micro and LoRa SX1278 radio module.

### LoRa node RX/data switch link

The queued datagrams received from the LoRa Receiver Module are encapsulated in an IP/UDP frame, using the unique network address and a dedicated (SCADA) port number. The LoRa Node Rx has the interface ENC28J60 controller that is responsible for wrapping the LoRa data format into UDP format. The encapsulated datagram is forwarded to a Gateway (listener) on the Data Switch (Packet Manger). The data switch will strip the encapsulation and check the LoRa Node  $R_x$  datagram for conformity:

- a. Start of frame
- b. End of frame
- c. LoRa Node  $T_x$  identifier

Any inconsistency will be treated as a malformed datagram and silently discarded. The IP addresses of the LoRa radio Module(receiver) on the SCADA network are static. The data switch gateway will maintain the relationship between the LoRa Radio Module (Transmitter) identifier and the wireless network IP address.

### Data switch—ignition master link

The Data Switch (packet manager) is resident to the same network, thus a private IP range as the SCADA

Server will identify each LoRa Rx device using the same address convention used by the LoRa device to identify itself. The data switch packet manager is an application running on PC buffering the data acquired from different LoRa modules. However, in this research, the buffered is a straightforward buffer for transmission delay check. This identifier is in Switch Terminology, called the Subscriber Identifier (SID). The parsing of data to the Ignition SCADA system will be made using the UDP/TCP driver.

The TCP Client installed on the Ignition Master Server will transfer the datagrams received to the Ignition server for processing and handling. The datagram which the packet manager sends will be delivered to the Ignition Master Server. The packet transport mechanism will be completely transparent to the server, i.e., as if the packet manager had been directly connected to the server. The results of the data analysis will be accessible to ignition slave stations (terminals).

Figure 9 represents the system's architecture, including the IoT devices connected via an LTE link with a remote data centre. The Ignition SCADA system can also receive MQTT data via the MQTT broker installed on the ignition SCADA in the form of the MQTT transmission module. Therefore, that any topic that the Ignition server is configured to listen to will be translated/'heard'.

## Conclusion

Presently, there are many challenges for system integrators to include new data points into existing industrial process systems with minimal cost. It can be seen from this research that it is very plausible and achievable if the data can be parsed in some IP frame format. The challenges lie in managing this acquired data without compromising the security of the process control system and not compromising the deterministic nature of the protocols used in a process control system. The LoRa physical layer in its original form/framework offers very little interoperability regarding the visualisation of the data. This paper implements an option that makes LoRa PHY integrate-able into High-level systems or MES. The mixing of different systems and protocols to meet the system's specific requirements sometimes needs integration of both automation protocols with IT transport layer protocols are becoming a new norm in the IIoT world (Felser et al., 2019). Additionally, with the advent of MQTT, more options are fast becoming available, especially wrt parsing data to higher-level applications.

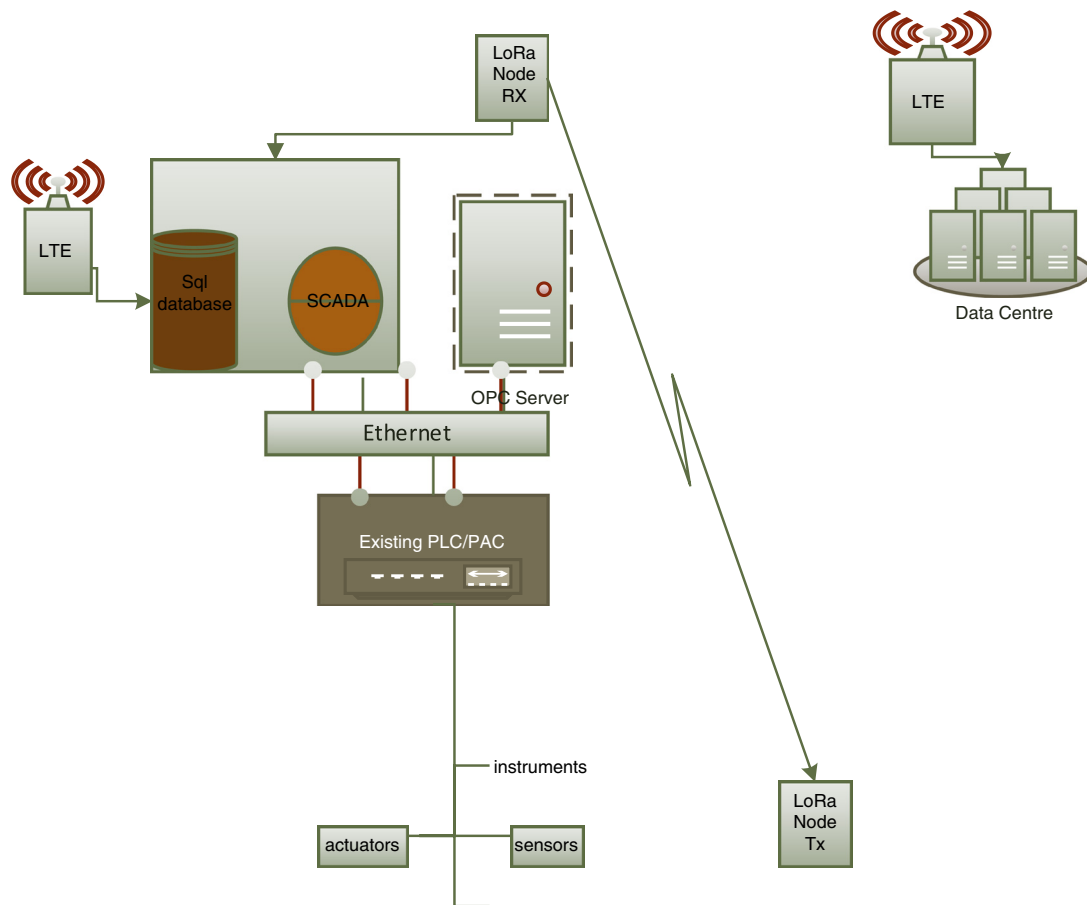


Figure 9: Architecture of a system with Added IIoT device using LoRa Phy.

## Conflicts of Interest

The authors declare that there is no conflict of interest.

## Funding statement

Research is funded by CPUT (publications only when accepted).

## Data availability

Data wherever required are referenced in the paper and ongoing research cannot be made available at this stage.

## Literature Cited

Astarloa, A., Bidarte, U., Jimenez, J., Zuloaga, A. and Lazaro, J. 2016. Intelligent gateway for industry 4.0-compliant production. *IECON Proceedings (Industrial Electronics Conference)*, doi: 10.1109/IECON.2016.7793890.

Azhari, S. J. and Kaabi, H. 2001. 4-20 Ma current loop transmitters. *International Journal of Engineering* 14(1): 35–40.

Bassi, L. 2017. Industry 4.0: Hope, hype or revolution? *RTSI 2017—IEEE 3rd International Forum on Research and Technologies for Society and Industry, Conference Proceedings*, doi: 10.1109/RTSI.2017.8065927.

Bäumker, E., Miguel Garcia, A. and Woias, P. 2019. Minimizing power consumption of LoRa® and LoRaWAN for low-power wireless sensor nodes. *Journal of Physics: Conference Series* 1407(1), doi: 10.1088/1742-6596/1407/1/012092.

Calderón Godoy, A. J. and Pérez, I. G. 2018. Integration of sensor and actuator networks and the SCADA system to promote the migration of the legacy flexible manufacturing system towards the industry 4.0 concept. *Journal of Sensor and Actuator Networks* 7(2), doi: 10.3390/jsan7020023.

Chen, B., Wan, J., Shu, L., Li, P., Mukherjee, M. and Yin, B. 2017. Smart factory of industry 4.0: key technologies, application case, and challenges. *IEEE Access*, doi: 10.1109/ACCESS.2017.2783682.



- D. Resources and D. Features 2015. Compact CAN-to-ethernet converter using 32-Bit ARM® Cortex™-M4F MCU TI designs, Available at: [https://esd.eu/en/products/ethercan-2?mtm\\_campaign=google\\_ads&mtm\\_kwd=ethercan2](https://esd.eu/en/products/ethercan-2?mtm_campaign=google_ads&mtm_kwd=ethercan2).
- Felser, M., Rentschler, M. and Kleineberg, O. 2019. Coexistence standardization of operation technology and information technology. *Proc. IEEE* 107(6), doi: 10.1109/JPROC.2019.2901314.
- Guo, Z., Zhang, K., Xin, H., Bi, M., He, H. and Hu, W. 2017. An optical access network framework for smart factory in the industry 4.0 era supporting massive machine connections. *ICOON 2017—16th International Conference on Optical Communications and Networks*, doi: 10.1109/ICOON.2017.8121481.
- Koon, J. 2020. LoRaWAN empowers very low-power, wireless applications. Vol. 1.1. Tech Idea Research.
- Langmann, R. and Rojas-Pena, L. F. 2016. A PLC as an industry 4.0 component. *Proceedings of 2016 13th International Conference on Remote Engineering and Virtual Instrumentation, REV 2016*, doi: 10.1109/REV.2016.7444433.
- NORDIC 2008. nRF24L01 + Single Chip 2.4GHz transceiver product specification v1.0. *Build. Res* no. 1: 78.
- Orange 2016. LoRa Device developer guide. 42; [Online]. Available at: <https://partner.orange.com/wp-content/uploads/2016/04/LoRa-Device-Developer-Guide-Orange.pdf>.
- Pagnon, W. 2017. The 4th industrial revolution—a smart factory implementation guide. *International Journal of Robotics and Automation*, doi: 10.15226/2473-3032/2/2/00123.
- Qfix 2012. User manual user manual 3304:1–148; no. January.
- Shrouf, F., Ordieres, J. and Miragliotta, G. 2014. Smart factories in Industry 4.0: A review of the concept and of energy management approached in production based on the Internet of Things paradigm. *IEEE International Conference on Industrial Engineering and Engineering Management*, doi: 10.1109/IEEM.2014.7058728.
- Surachman, A., Suhendi, A., Budiman, M., Abdullah, M. and Khairurrijal, K. 2011. A simple microcontroller-based 4-20 ma current loop receiver for sensors with current transmitters. *Journal of Otomasi Kontrol dan Instrumentasi* 2(1): 15.
- Tao, F., Zuo, Y., Da Xu, L. and Zhang, L. 2014. IoT-Based intelligent perception and access of manufacturing resource toward cloud manufacturing. *IEEE Transactions on Industrial Informatics*, doi: 10.1109/TII.2014.2306397.
- Tel, G. 2012. *Communication Protocols. Introduction to Distributed Algorithms*. Cambridge University Press, pp. 74–102, doi: 10.1017/cbo9781139168724.004.
- Wollschlaeger, M., Sauter, T. and Jasperneite, J. 2017. The future of industrial communication: automation networks in the era of the internet of things and industry 4.0. *IEEE Industrial Electronics Magazine*, doi: 10.1109/MIE.2017.2649104.
- Yan, J., Meng, Y., Lu, L. and Li, L. 2017. Industrial big data in an industry 4.0 environment: challenges, schemes, and applications for predictive maintenance. *IEEE Access*, doi: 10.1109/ACCESS.2017.2765544.
- Zhou, K., Liu, T. and Zhou, L. 2016. Industry 4.0: Towards future industrial opportunities and challenges. *2015 12th International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2015*, doi: 10.1109/FSKD.2015.7382284.