Smarter solutions for industrial efficiency

automationmagazine.co.uk

# Automation

November 2021



## Fantastic Christmas offer from QDUKI and Raptor Photonics

*more on page 35*

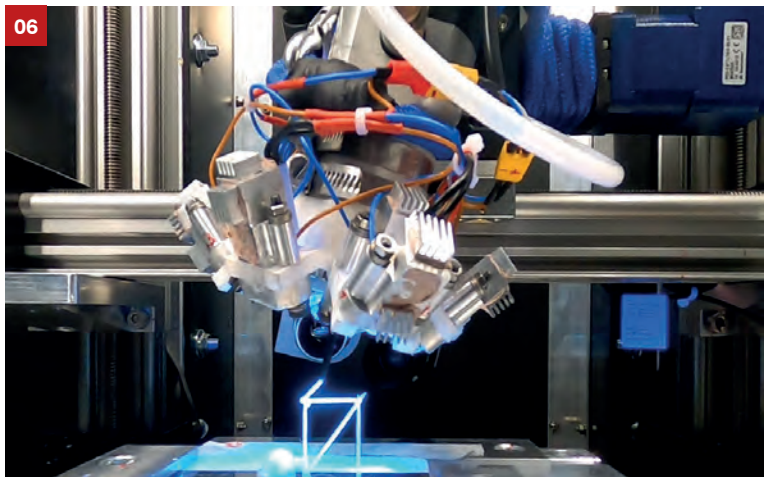
**INDUSTRY FOCUS**
**Food & Beverage / Oil & Gas/ Automated Warehousing**

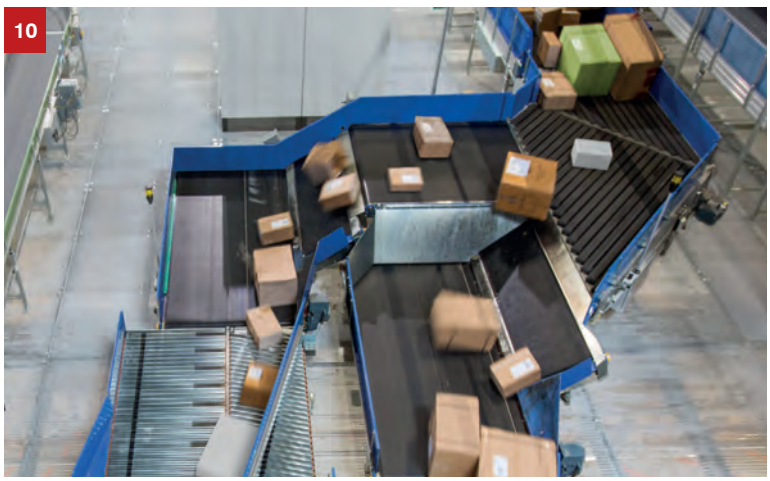


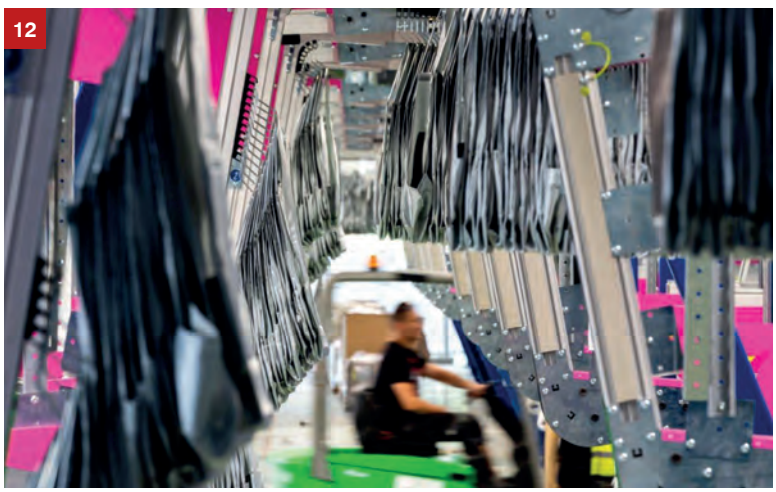
Topics in this issue:

- Robotics
- Sensors & Sensing Systems
- Coding, Marking & Labelling
- AI & VR

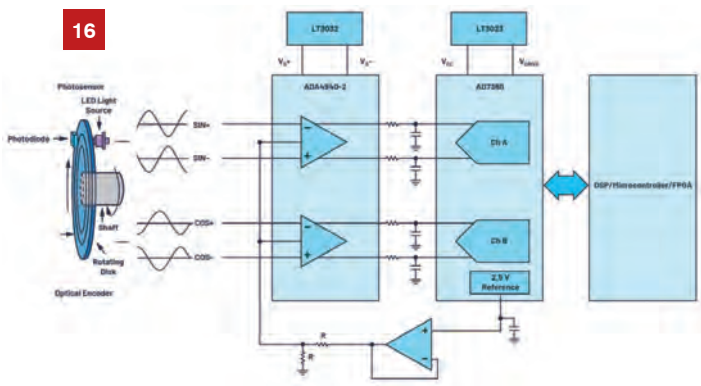
# Contents



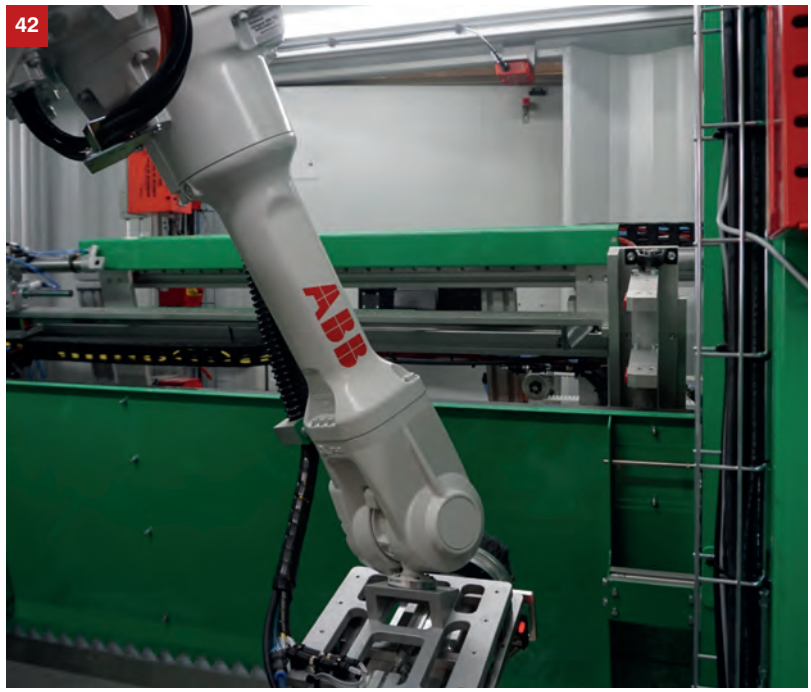



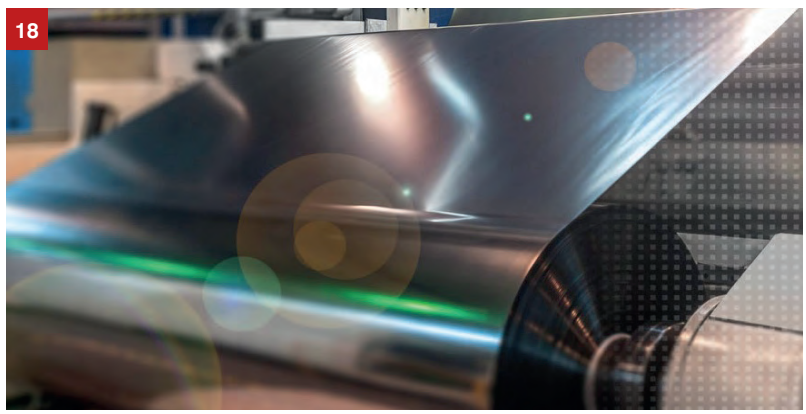


# Contents




# Is your facility connected? You are then at risk

IoT-enabled devices used in industrial applications are vulnerable to many forms of cyberattack. Michael Barrett, Managing Director of Nexus Industrial Memory, outlines some ways in which devices might be attacked and suggests how to make them more secure

[Image: Lalit Kumar for Unsplash]

Security is a growing concern for the IoT and its industrial counterpart, the IIoT. It's worth mentioning that those attacking IIoT have different objectives to those attacking IoT. For instance, IIoT data theft is more likely to be about acquiring industrially-sensitive information, and ransomware attacks will be out to disable equipment and disrupt processes. Other attacks aim to cause damage by compromising safety-critical systems.

If a device is connected to the Internet, it is exposed to cyberattacks, and if it still uses the default username and password with which it left the factory, it may as well have no security at all. For example, Mirai, a self-propagating botnet (a.k.a., "zombie") attacks poorly-protected systems using telnet to find devices that are still using their default username and password.

If these devices are used in multiple locations around the world – to report performance and diagnostic data back to a single server, for example – they can be instructed (from a command-and-control, CnC, centre) to perform a distributed denial of service (DDoS) attack (Figure 1).

## Varied attacks

In many cases though, it is not necessary to rely on the user failing to change a default password. There's another way in: through data the device is expecting to receive. For instance, a common form of attack on IoT-enabled devices, and for which the programs are written in a low-level language like C, is through a forced memory-buffer overflow.

The spearhead of the attack is to write data to memory reserved for runtime activities that is larger than that the device expects to receive. The excess data overflows into other memory space and overwrites machine code that governs the system's behaviour. If the overflow data is something like a new return address, a different part of the program will execute next. This might be a legitimate function, such as the restoration of factory settings (including default passwords), or the hacker can simply set a new password. Either way, the hacker has access to the system. However, the legitimate user might wonder why they no longer have access.

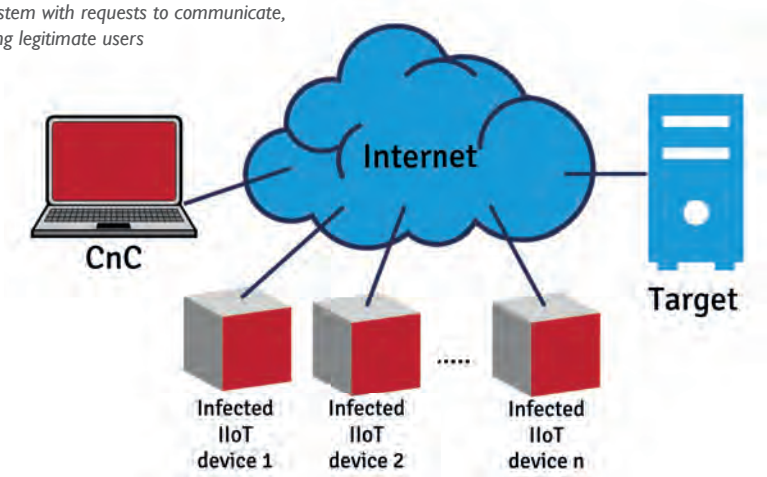
A more severe memory buffer overflow attack sees the introduction of shellcode to give the device new behaviour. This might be to reveal the password set by the legitimate user of the system, or it could be to reveal the password the device uses to get on to the network and communicate with other devices. The legitimate user may never know the device has been compromised. As for how the hacker knows where to write the overflow data and what it should be, if the device is an off-the-shelf product, it can be reverse-engineered to establish its memory map.

## Protection

One way of protecting against memory-buffer overflow attacks is not to program in a low-level language. Instead, use C# or Java, for example. Also, if there's room for an operating system use it since it will offer memory management. If not, dedicated memory management unit (MMU) chips can be used. Either way, certain areas of the device's memory need to be protected (declared as Read Only, for example) during runtime.

An IIoT device can also be attacked through its application program interface (API). Many devices use a representational state transfer (REST)-based API called

Figure 1: The attacker uses the CnC to instruct infected devices to inundate the IP address of a target system with requests to communicate, thus denying legitimate users




RESTful. Such APIs are popular because they do not require much bandwidth, can be crafted in Python or JavaScript, and they use HTTP to communicate with the cloud. This means data can be created, updated, read or deleted.

The use of any REST-based API presents certain cyber challenges. These can be addressed through better authentication for access control, blocking certain payloads (size and/or type that aren't expected) and access from unknown IP addresses and domains. If the IIoT-based device is just one of few that is part of a well-conceived IT and operating technology (OT) system, none of these solutions should prove too difficult to implement.

## Secure by design

Traditionally, security considerations have always come late during product development, sometimes as late as prototyping. This must change; security should be considered when specifying a device's requirements. Its intended length of service in the field and importance of what it does and the data it handles will govern the measures to take.

Let's assume high security is required and consider, for a moment, end-to-end communications. An important element is the device's identity. How trustworthy is it? Is static data encrypted on the device? Should/can data in transit be encrypted to thwart 'man-in-the-middle' interceptions?

Thankfully, the OEMs of microcontrollers are producing some great ICs that are very much geared for high-security IoT operation. Take Microchip Technology's CryptoAuthentication family of devices, for example. These devices can work alongside the microcontroller or microprocessor within an IoT-enabled systems. IC features include a unique and non-changeable 72-bit serial number (set by Microchip), a 512-bit one-time programmable (OTP) zone, a random number generator and a SHA-256 hash algorithm for data encryption. They also include APIs for storing, retrieving and manipulating X.509 certificates for Transport Layer Security (TLS) integration.

In our example of a server communicating with multiple IIoT-enabled devices, the end-to-end communications link can be made far more secure by virtue of unique IDs in the field and encrypted transferred data.

## IIoT in Automation

The IIoT is very much part of Industry 4.0 and M2M communication. It has brought great benefits. However, IIoT is challenging the Purdue Model, shown in Figure 2, which reflects the hierarchy of IT and OT systems elements, and comprises six layers:

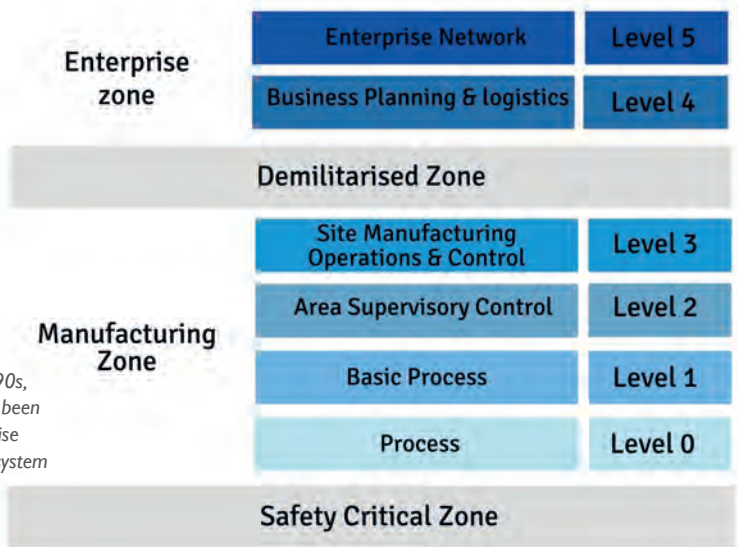
- Level 5 = corporate network systems.
- Level 4 = IT systems for business logistics (includes databases and servers).
- Level 3 = systems for site-wide monitoring and control.
- Level 2 = control systems such as HMIs and SCADA software.
- Level 1 = basic control devices such a programmable logic controllers.
- Level 0 = sensors, actuators, motors and pumps, and so on.

The purpose of the Purdue architecture is to assure safe control, noting that safety and security go hand in hand. Within the enterprise zone it has historically been only the enterprise network that has had access to the Internet and the outside world. Malware hitting IT equipment at levels 5 or 4 should not be able to affect anything at level 3 or lower because of the firewalled demilitarised zone.

Today, many OT devices in the manufacturing zone are now IoT enabled. Smart sensors and controllers, along with edge-processing systems, are connected to the Internet. Data no longer flows between the Purdue Model levels.

There are mixed views in the industry whether the Purdue Model needs to be replaced or enhanced in light of the increased use of IIoT within the manufacturing zone. As stressed earlier, a risk analysis must be performed on any IIoT device relative to its level in the Purdue Model and whether it is playing a role in monitoring, controlling or both.

Figure 2: Since the 1990s, the Purdue Model has been a standard for enterprise and industrial control system networks




CONTACT:
Nexus Industrial Memory
www.nexusindustrialmemory.com