

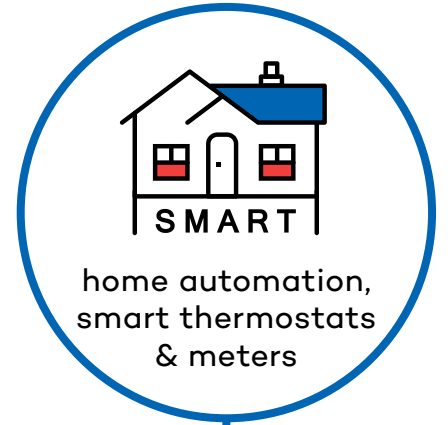
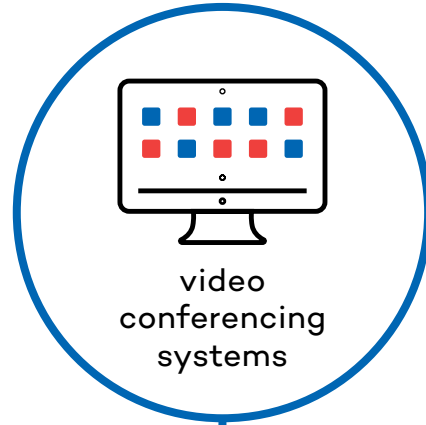
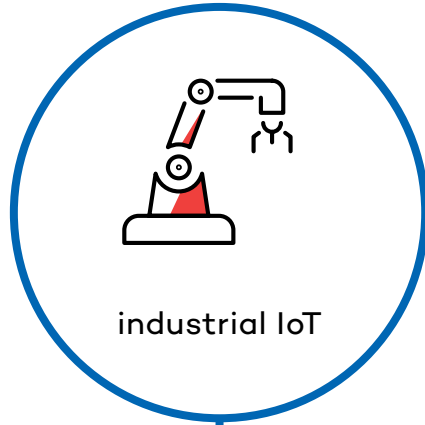


**embedded  
wi-fi  
development  
best  
practices**

# 1 common development challenges

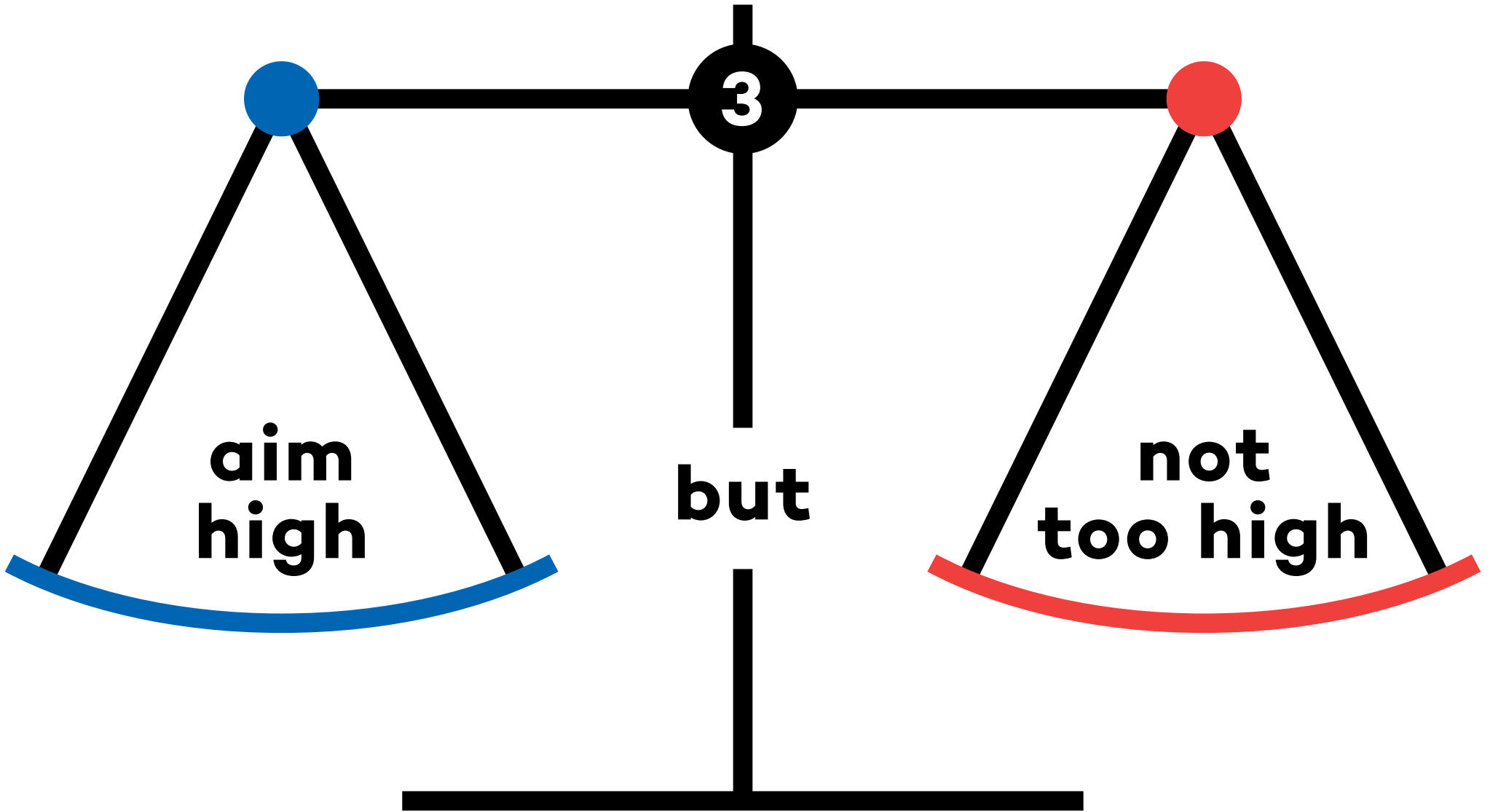
Embedded systems are complex, with software development and debugging often the most critical steps in system development.

Wi-Fi adds to the complexity. To help you, here are a few best practices to overcome common development challenges and deliver projects on time.



## 2. example use cases





Balance tradeoffs between high-reaching specs and proven technology. Good engineering practice requires that you identify the real requirements.



4  
understand  
that PCs  
and embedded  
ecosystems  
have different  
firmware and  
software  
requirements



**a**

Consider using a real-time operating system (RTOS).

**b**

Consider code already verified on embedded platforms.

**c**

Cater for the need for embedded devices to be robust. Use an automated test architecture for a rock solid product.



L  
010  
0110  
010011  
00101010  
1001100  
01101  
110  
0

# 5 make smart build-it or buy-it decisions

\$  
¥£  
£€\$¥  
€\$¥£€\$  
\$¥£€\$¥£€  
£€\$¥£€  
¥£€\$¥  
€\$¥  
\$

Time to completion and level of internal expertise  
may dictate the best choice.

Does building your own  
fit your project timeline?  
Or should you focus  
your efforts on application  
logic and end user  
requirements?

The silicon vendor's  
driver and open source  
drivers for Linux may  
not meet the requirements  
of the RTOS and limited  
resource architectures.

Look for ways to acquire the  
expertise of Wi-Fi specialists.  
Experts may not only speed  
up the initial development  
but assist to resolve bugs and  
remedy vulnerabilities.

**6 trust**

**that**

**failure**

**is a**

**stepping**

**stone**

**to success**

Build a proof of concept (POC), test early, test hard, fail, fix, and repeat.

Use debug tools and logs provided by ClariFi essential.



Tip: Wi-Fi debugging requires experience and good tools. If you don't yet have the experience, consider working with experts in the field.

**7. manage  
the co-existence  
of wi-fi**

Control from a single point  
in the same framework,  
the same application and  
the same API structure.

**with**

**Bluetooth  
and/or Bluetooth  
Low Energy**

Manage  
bandwidth  
actively.



8



**optimize  
data transfer &  
power management**



Find the optimal speed  
versus resource consumption  
for your project.

Monitor the dynamic behavior  
of the target platform to gain  
high visibility of system  
performance without  
interference.





9

# future-proof your software development life cycle

Use an abstraction layer over your RTOS, hardware interface and debugging architecture.

Use the same APIs, stacks, and applications to promote reusability when changing the target platform.<sup>1</sup>

Plan how to follow the ever-evolving Wi-Fi standards: Wi-Fi-4, Wi-Fi-5, Wi-Fi-6, and/or Wi-Fi-6e.

<sup>1</sup> Combination of MCU/MPU, RTOS, and wireless chipset.



**clarinox**

**deliver innovative, trusted wireless connectivity and  
get to market faster with Clarinox solutions.**



# about clarinox technologies pty ltd

Clarinox Technologies Pty Ltd provides cost-effective and innovative Bluetooth and Wi-Fi solutions for embedded systems. We passionately deliver flexible and robust wireless protocol software for embedded systems developers.

Contact Clarinox today to discuss your next Wi-Fi integration project.

[www.clarinox.com](http://www.clarinox.com)

