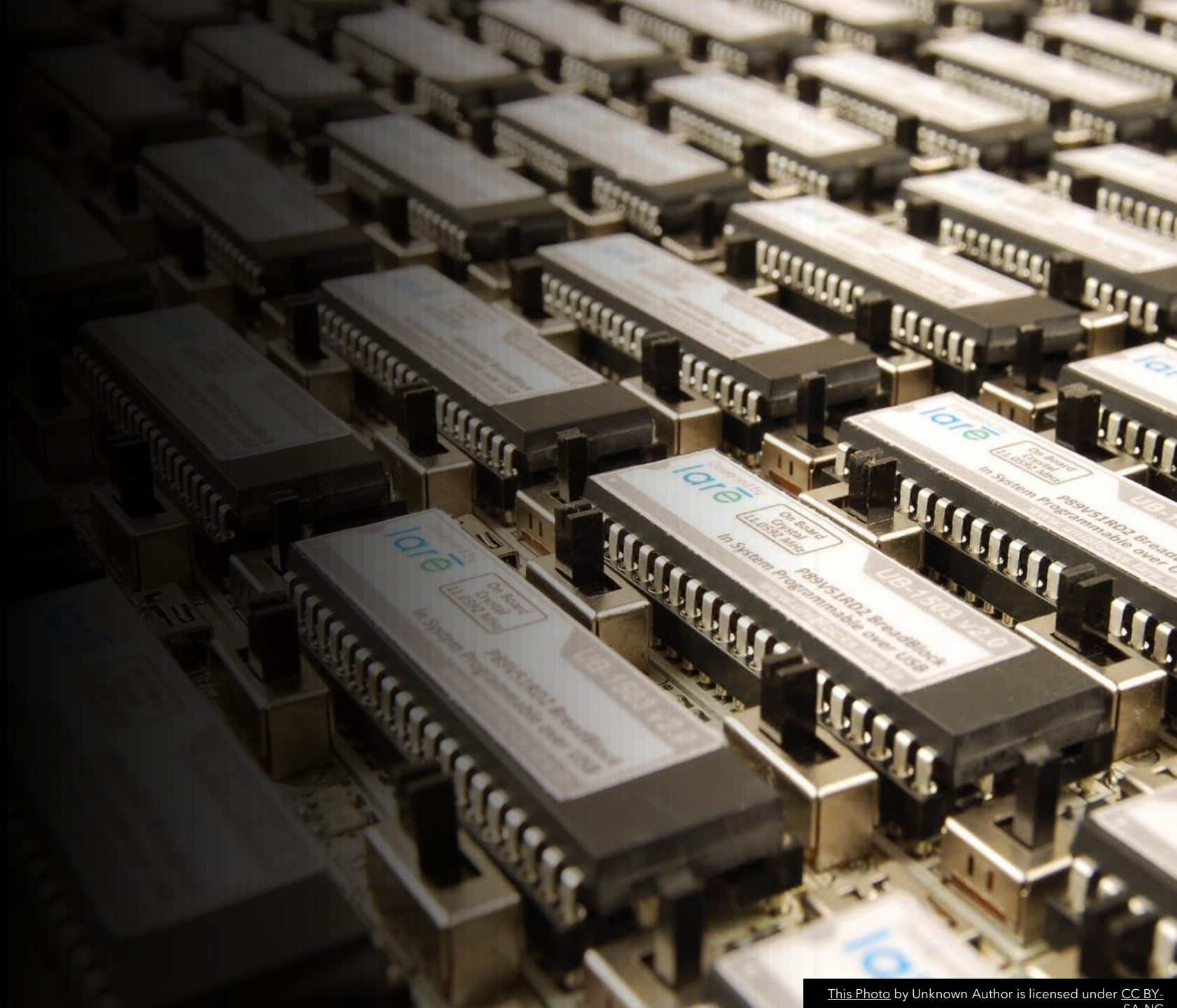
TinyML Frameworks & Tools

Tensorflow Lite Micro

Clinton Oduor
Machine Learning Engineer,
Amini

Embedded devices Compute constraints

- Low RAM
- Low Flash Memory
- Low Clock speed
- Doesn’t run on an OS

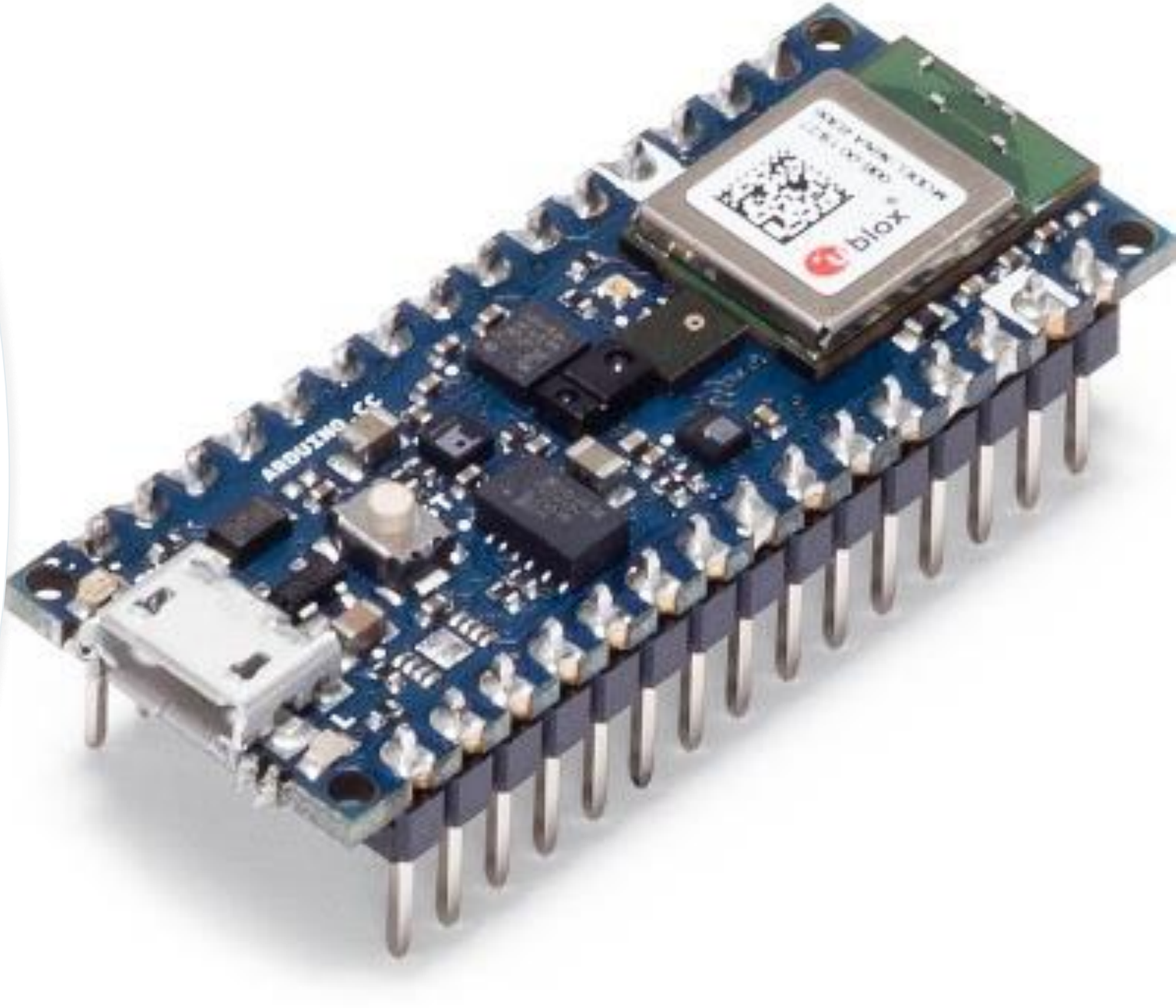


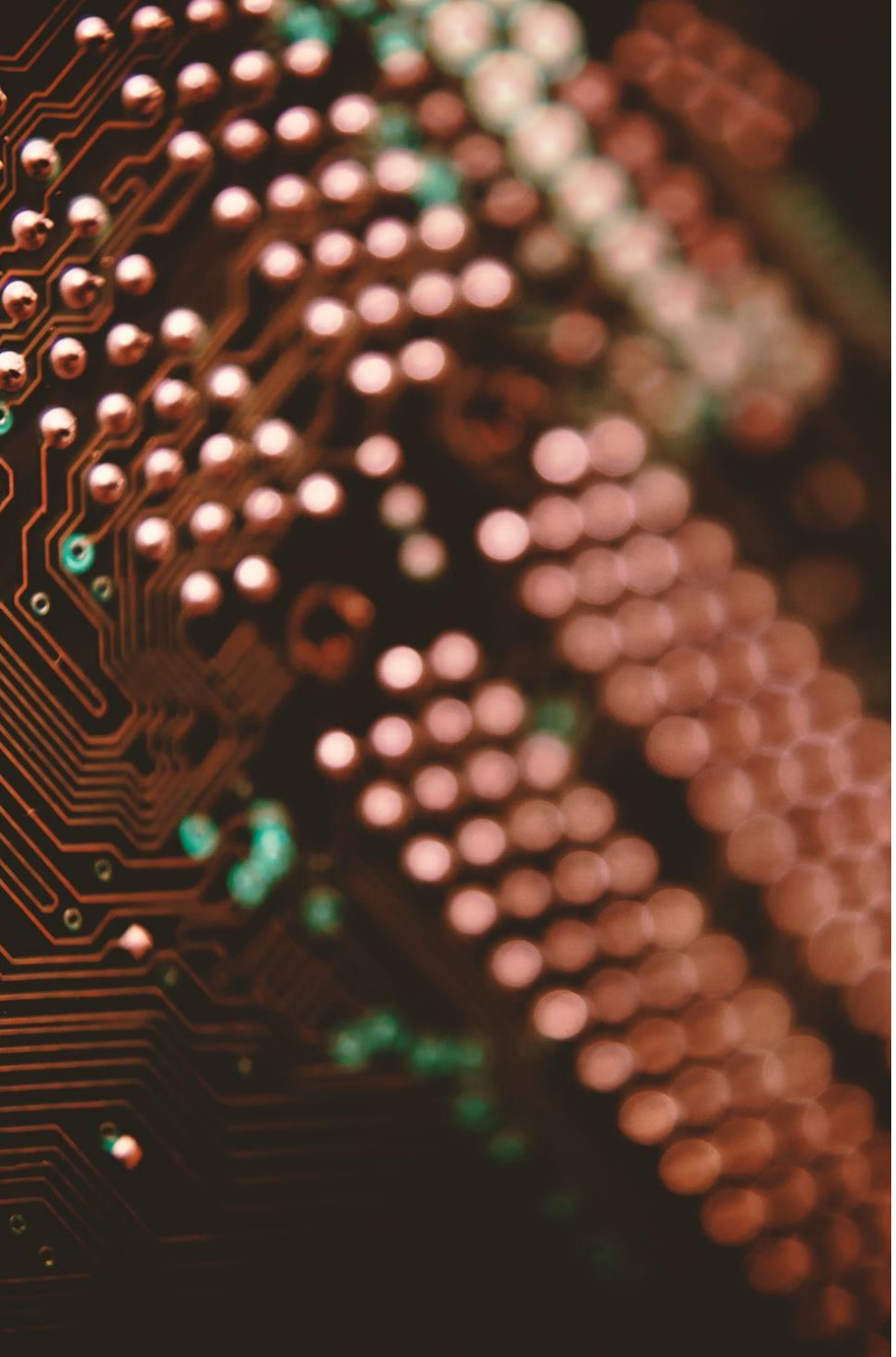
Tinyml Applications Development Workflow

Key Features of TensorFlow Lite Micro

Core runtime fits in just 16KB on an Arm Cortex M3.

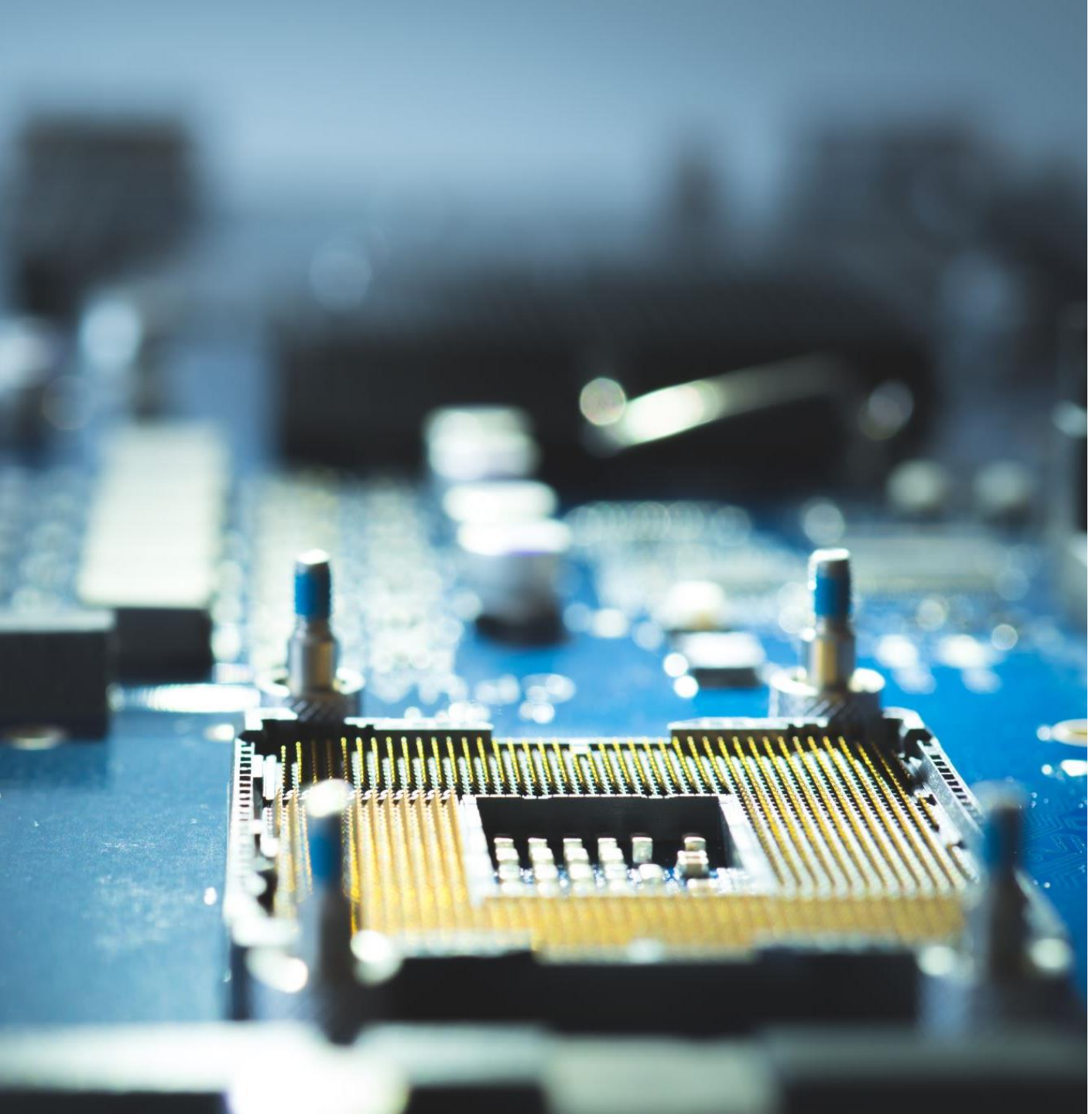
Designed for extremely constrained environments, allowing ML models to run on devices with limited memory.





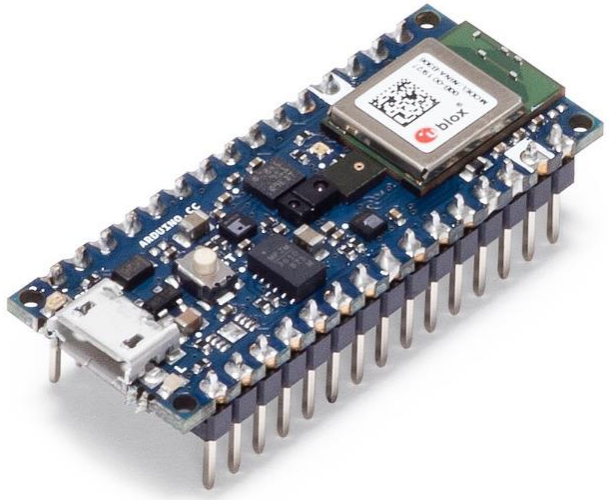
Static Memory Allocation

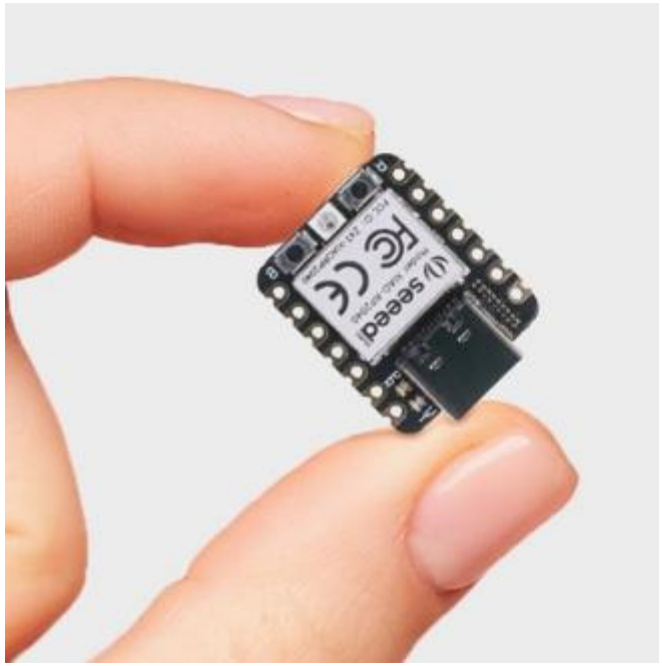
- No dynamic memory allocation required.
- This improves the predictability and reliability of applications, a key factor in many real-world systems.

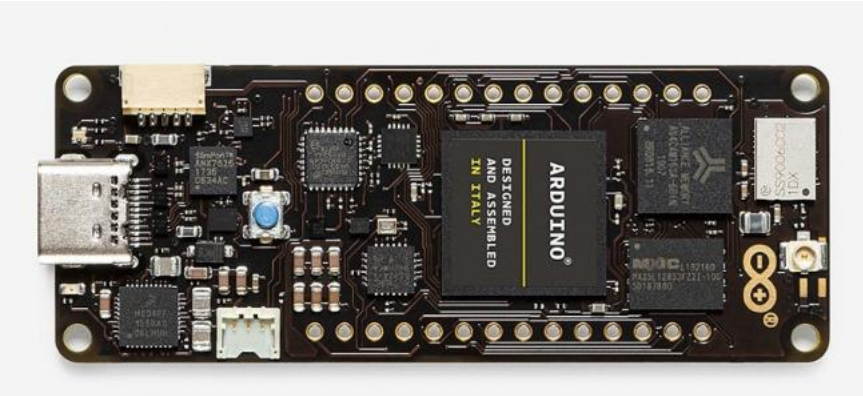


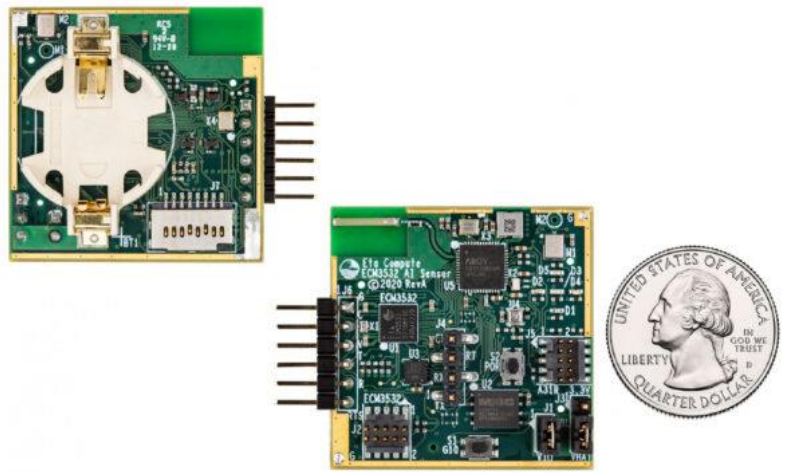
Broad Platform Support

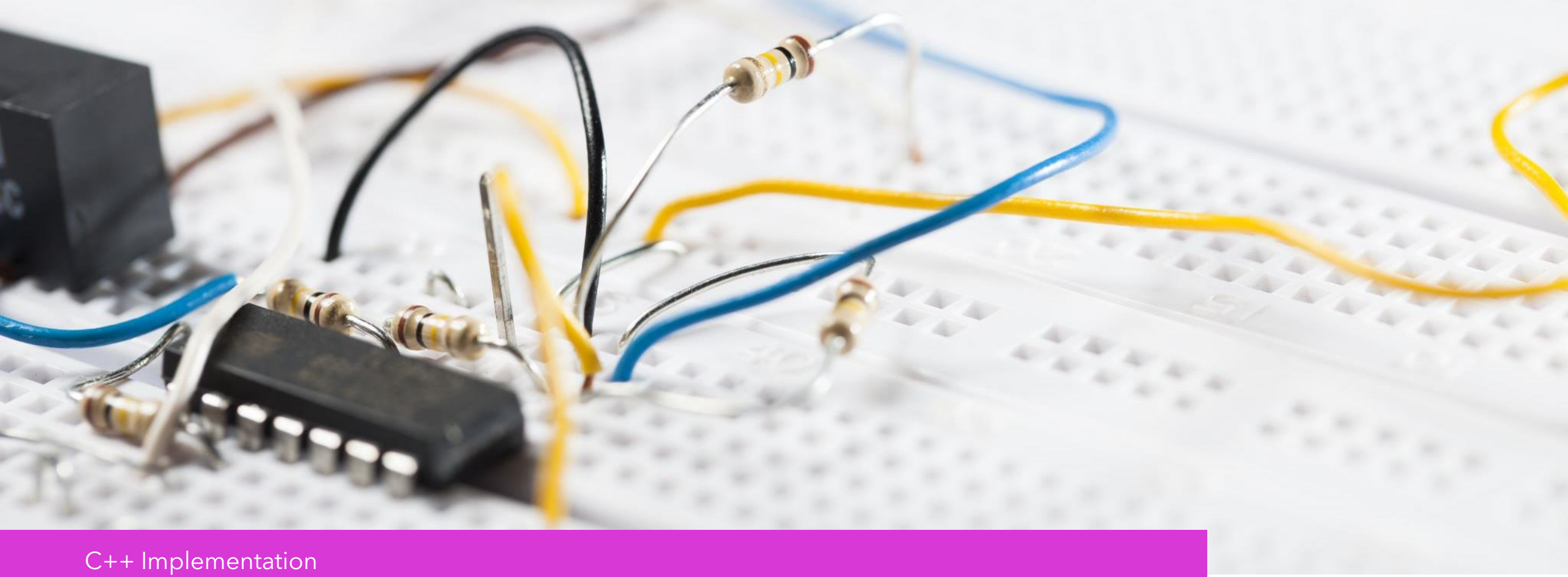
- Compatible with a variety of 32-bit microcontrollers.
- This flexibility enables TensorFlow Lite Micro to be deployed across a wide range of devices, from household appliances to industrial sensors.


ARDUINO.CC
blox


seeed


ARDUINO
DESIGNED AND ASSEMBLED IN ITALY


ECM3532
LIBERTY
QUARTER DOLLAR

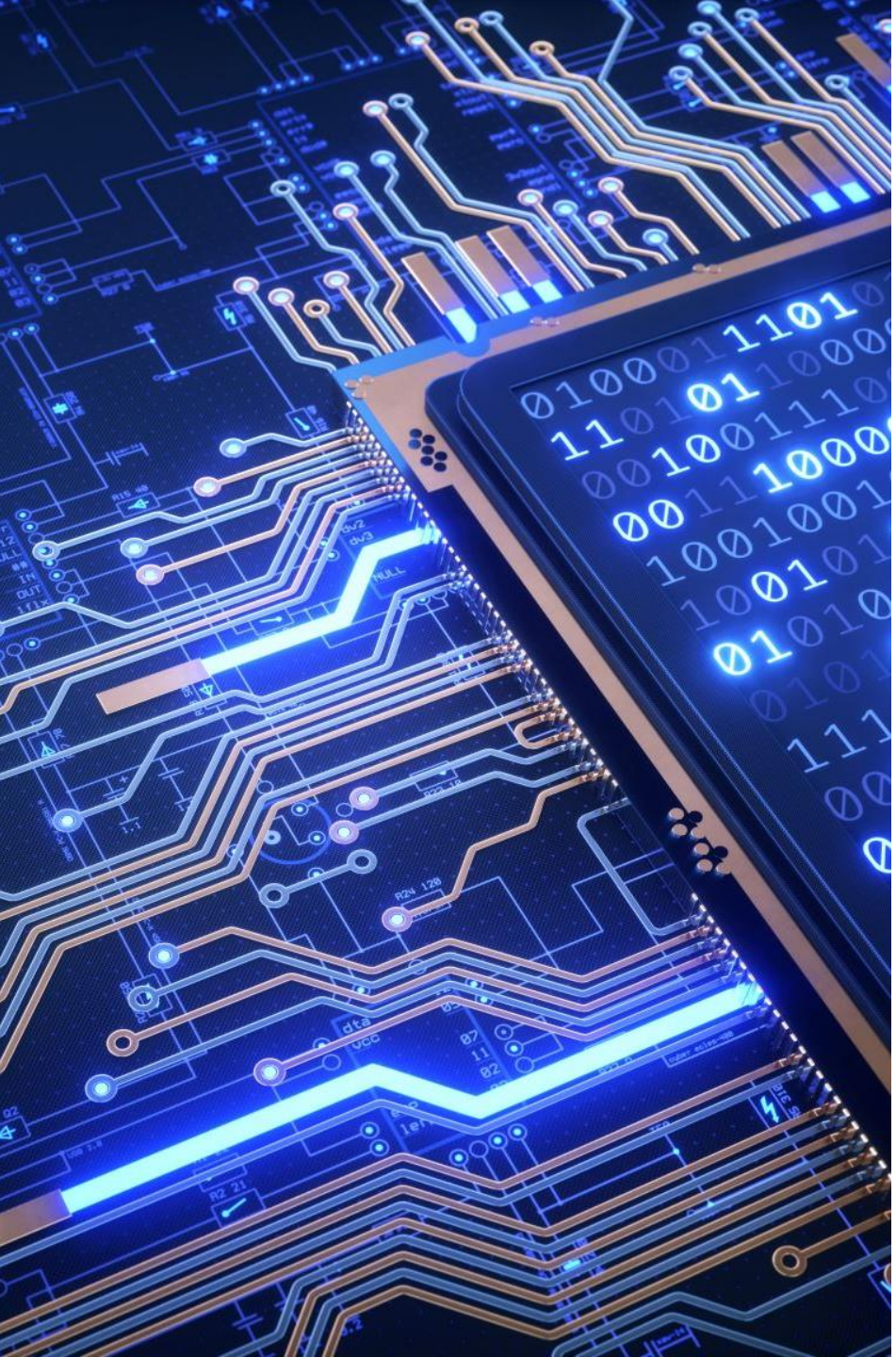


C++ Implementation

- C++ allows for low-level access to hardware and high performance, making it easier to integrate with existing C++ projects.

Designed with a minimalistic interpreter focused on inference.

- The interpreter is optimized for low memory usage and fast execution, making it ideal for real-time applications



Role of the Interpreter in TensorFlow Lite Micro

- Inference Execution
- Operation Kernels
- Memory efficiency

Pre-built Models

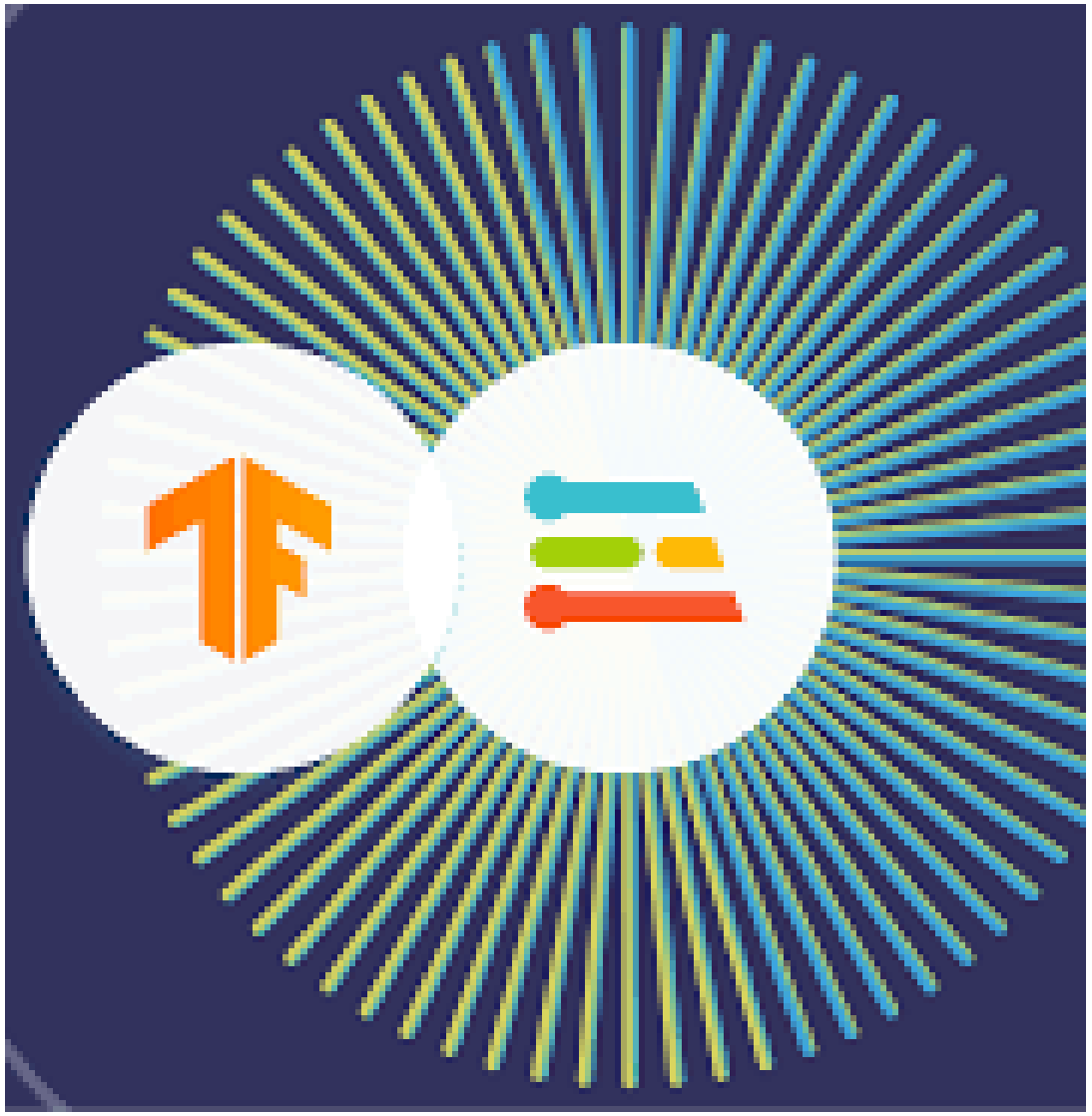
- Availability of models for common tasks.
- Pre-built models for tasks like speech recognition and person detection are available, reducing the time and expertise required to develop solutions

Operational without an underlying operating system.

- This reduces the complexity of the system and eases integration into bare-metal or custom OS solutions.

Community Support

Tools & Platforms



Edge Impulse X Tensorflow Lite Micro