Confirm
Smart Manufacturing

Confirm
Smart Manufacturing

OTA-TinyML: Over the Air Deployment of TinyML Models and Execution on IoT Devices

Bharath Sudharsan, John G. Breslin, Mehreen Tahir, Muhammad Intizar Ali, Omer Rana, Schahram Dustdar, and Rajiv Ranjan

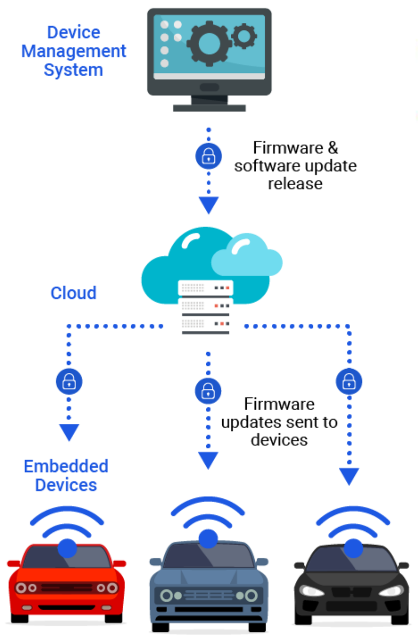
A World Leading SFI Research Centre

Science Foundation Ireland For what's next

University of Limerick Ollscoil Luimnigh · Tyndall National Institute · UCC University College Cork, Ireland Coláiste na hOllscoile Corcaigh · NUI Maynooth Ollscoil na hÉireann Má Nuad · OÉ Gaillimh NUI Galway · Cork Institute of Technology Institiúid Teicneolaíochta Chorcaí · AIT · LIT

Introduction

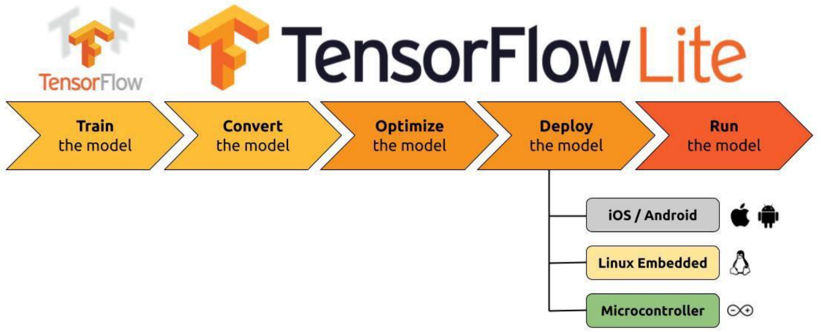
Confirm Smart Manufacturing


Device Management System
Firmware & software update release
Cloud
Firmware updates sent to devices
Embedded Devices

Firmware and software updates download new code to the embedded devices

OTA updates

https://www.rambus.com/blogs/ota-updates-explained/



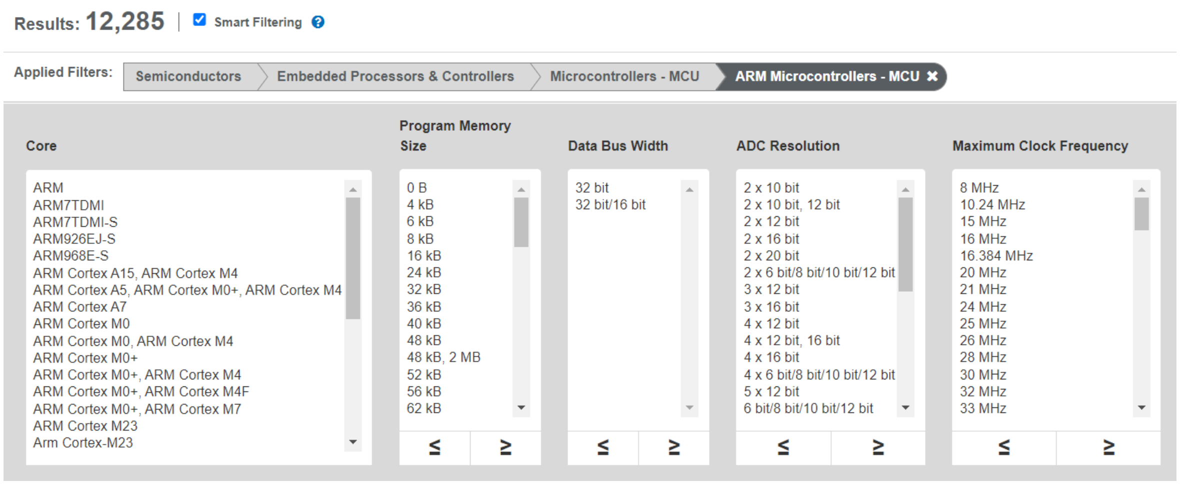
ESP32-CAM	Sipeed MAIX Bit	M5StickV AI Camera	OpenMV Cam H7
CPU: Dual core, 32-bit, 240 MHz, 4 MB Flash, 520 KB SRAM	CPU: RISC-V Dual Core, 64-bit, 400 Mhz, 16 MB Flash, 8 MB SRAM	CPU: RISC-V, 64-bit, 400 MHz, Dual FPU, 16 MB Flash, 8 MB SRAM	CPU: ARM Cortex-M7, 32-bit, 480 MHz, 2 MB Flash, 1 MB SRAM
Camera: 2 MP	Camera: 2 MP	Camera: VGA	Camera: 5 MP
Price: ≈ $ 8	Price: ≈ $ 21	Price: ≈ $ 30	Price: ≈ $ 65

Workflow of deploying ML models on IoT devices

- OTA updates - remotely, efficiently, and reliably update field devices
- IoT devices memory only sufficient for routine device functionalities
- Poorly executed OTA
  - ✓ Result in bricked devices
  - ✓ Customer inconvenience
  - ✓ Reputational damage
- OTA-TinyML - end-to-end remote fetching, storing, and execution of TinyML models on IoT devices

Challenges

Confirm
Smart Manufacturing



Results: 12,285 | ☑ Smart Filtering

Applied Filters: Semiconductors > Embedded Processors & Controllers > Microcontrollers - MCU > ARM Microcontrollers - MCU ✖

Core
ARM
ARM7TDMI
ARM7TDMI-S
ARM926EJ-S
ARM968E-S
ARM Cortex A15, ARM Cortex M4
ARM Cortex A5, ARM Cortex M0+, ARM Cortex M4
ARM Cortex A7
ARM Cortex M0
ARM Cortex M0, ARM Cortex M4
ARM Cortex M0+
ARM Cortex M0+, ARM Cortex M4
ARM Cortex M0+, ARM Cortex M4F
ARM Cortex M0+, ARM Cortex M7
ARM Cortex M23
Arm Cortex-M23

Program Memory Size
0 B
4 kB
6 kB
8 kB
16 kB
24 kB
32 kB
36 kB
40 kB
48 kB
48 kB, 2 MB
52 kB
56 kB
62 kB

≤ ≥

Data Bus Width
32 bit
32 bit/16 bit

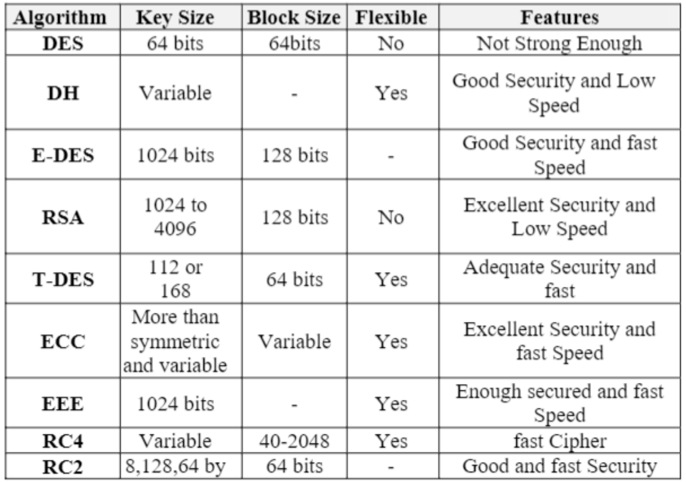
≤ ≥

ADC Resolution
2 x 10 bit
2 x 10 bit, 12 bit
2 x 12 bit
2 x 16 bit
2 x 20 bit
2 x 6 bit/8 bit/10 bit/12 bit
3 x 12 bit
3 x 16 bit
4 x 12 bit
4 x 12 bit, 16 bit
4 x 16 bit
4 x 6 bit/8 bit/10 bit/12 bit
5 x 12 bit
6 bit/8 bit/10 bit/12 bit

≤ ≥

Maximum Clock Frequency
8 MHz
10.24 MHz
15 MHz
16 MHz
16.384 MHz
20 MHz
21 MHz
24 MHz
25 MHz
26 MHz
28 MHz
30 MHz
32 MHz
33 MHz

≤ ≥



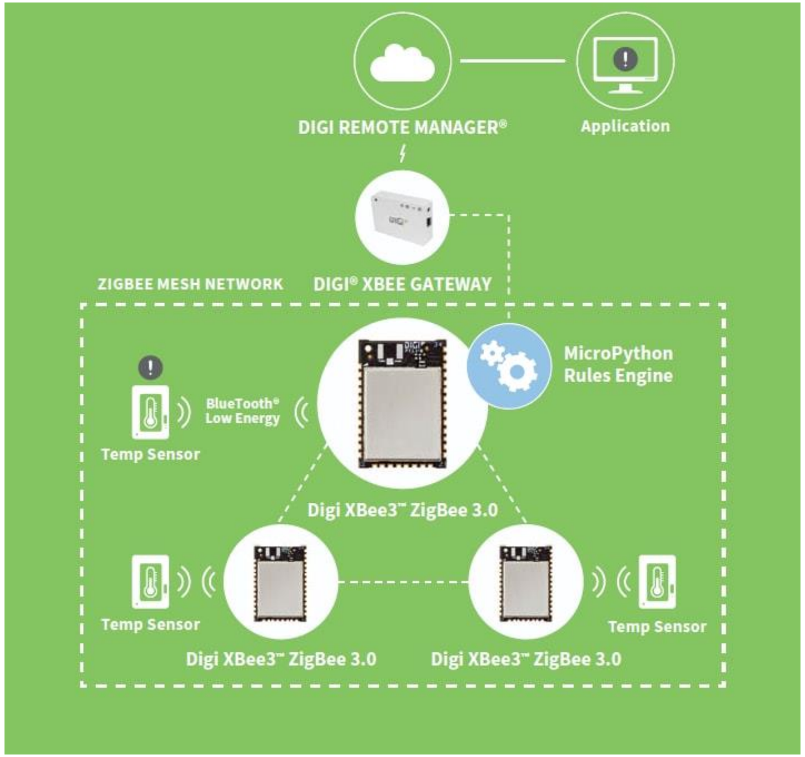
Algorithm	Key Size	Block Size	Flexible	Features
DES	64 bits	64bits	No	Not Strong Enough
DH	Variable	-	Yes	Good Security and Low Speed
E-DES	1024 bits	128 bits	-	Good Security and fast Speed
RSA	1024 to 4096	128 bits	No	Excellent Security and Low Speed
T-DES	112 or 168	64 bits	Yes	Adequate Security and fast
ECC	More than symmetric and variable	Variable	Yes	Excellent Security and fast Speed
EEE	1024 bits	-	Yes	Enough secured and fast Speed
RC4	Variable	40-2048	Yes	fast Cipher
RC2	8,128,64 by	64 bits	-	Good and fast Security

- **IoT Hardware Level**
  - ✓ Intrinsically heterogeneous specifications
  - ✓ Resource-friendliness of OTA updates
  - ✓ Exploiting chipset level flaws

- **Cryptography Level**
  - ✓ Engineering efforts - optimization, cross-platform porting
  - ✓ Public-key cryptography smaller key size than RSA
  - ✓ HACL, Mbed TLS, TweetNaCl, NaCl, WolfSSL, Monocypher

Challenges

Confirm Smart Manufacturing



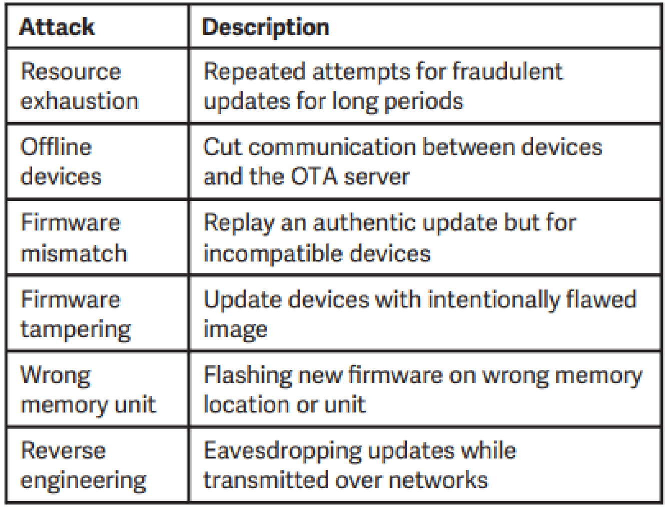
Example IoT architecture - challenges at various levels
https://www.mouser.ie/new/digi-international/digi-xbee3-modules/

- **Network and Transport Level**
  - ✓ Reverse engineering attacks
  - ✓ Devices fleet - Digi XBee sub-1 GHz networks or LR-WPAN
  - ✓ Bandwidth constraints or wireless interference
  - ✓ Transmission errors or corrupt the OTA files
- **Remote Device Management and OS Level**
  - ✓ Vertical IoT integrations - sensors to business level
  - ✓ LwM2M, CWMP - third-party management
  - ✓ High-privilege access to massive networks of devices
  - ✓ FreeRTOS, mC/OS, RIOT, Mbed OS, Zephyr, Tock

Directions for Handling Challenges

Confirm Smart Manufacturing

TABLE 1. Sample list to evaluate OTA update mechanisms.



Attack	Description
Resource exhaustion	Repeated attempts for fraudulent updates for long periods
Offline devices	Cut communication between devices and the OTA server
Firmware mismatch	Replay an authentic update but for incompatible devices
Firmware tampering	Update devices with intentionally flawed image
Wrong memory unit	Flashing new firmware on wrong memory location or unit
Reverse engineering	Eavesdropping updates while transmitted over networks

- Batching Devices and Update Roll-Out Phases
- Remote and Physical Security
- Minimizing Unexpected Saturation and Downtimes
- End-to-End Integrity Protection
- Failure Recovery
- Evaluation Metrics

Memory Unit and FS Rudiments

- **EEPROM File System (EEFS)** - Most commonly considered nonvolatile memory unit to add to an embedded system
  - ✓ EEFS serves purpose of FS abstraction for EEPROM, RAM, PROM memories while being efficient, lightweight, and reliable
- **On-Chip Flash** - Data persist across resets or power failures - network credentials, API keys - can use Preferences.h library
  - ✓ Higher capacity flash chip can be soldered on the board outside the processor chip and interfaced via a protocol
- **SPI Flash File System (SPIFFS)** - Internal FS that can store files in the NOR flash chip
  - ✓ Low RAM usage; supports Posix-like API that can accept open, close, read, write; compatible with any NOR flash
- **SD Card File System (SDFS)** - IoT apps can move through directories, create and read/write files
  - ✓ Compared to other memory units this contains numerous open-source contributions and supports FAT16, FAT32
- **Embedded Multimedia Card (eMMC)** - Comparatively more expensive and come with benefits
  - ✓ Higher reliability due to a lower likelihood of physical corruption and faster IoT application interaction with memory

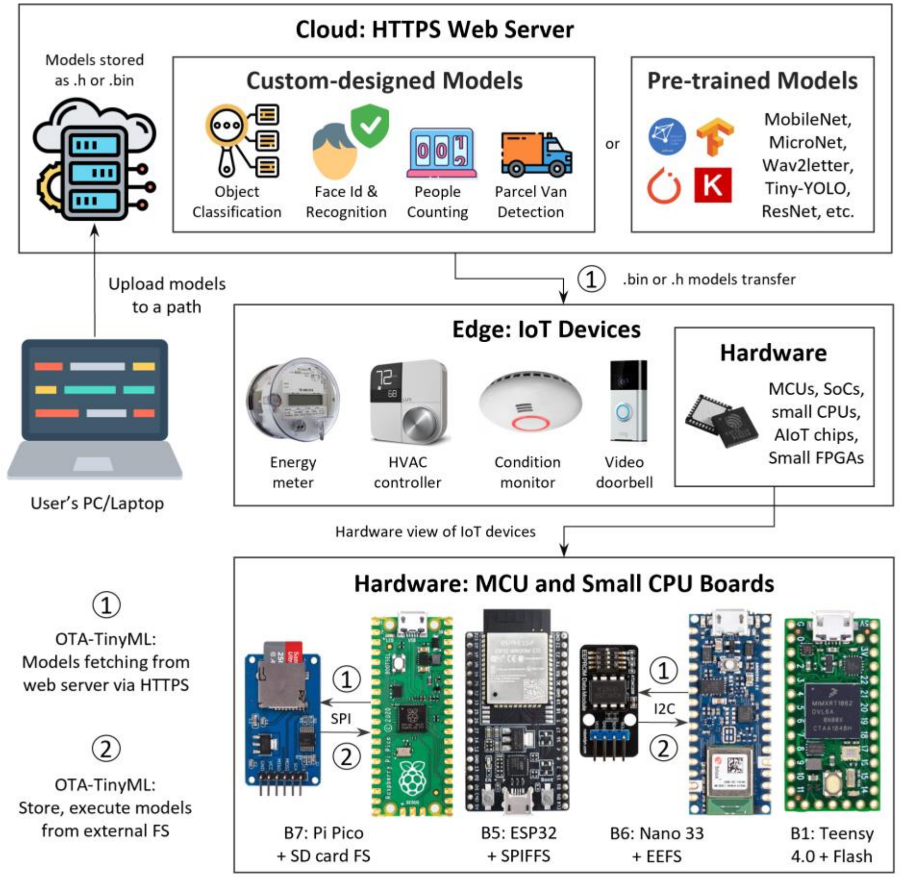
Memory Unit Selection Guidelines

Confirm Smart Manufacturing

- **Interplay among Cost, Space, and Speed**
  - ✓ EEPROMs are cheaper than SD cards. e.g., a 64 KB costs $1.50
  - ✓ EEPROM bit rate ranges from 100 to 1000 Kbits/s. SD card is faster but more susceptible to noise
  - ✓ When no-cost restrictions FRAM can be considered - write cycles < 50 ns and more radiation tolerant
- **Energy and Memory**
  - ✓ FRAM just needs 1.5 V to operate. SD card needs 3.3 V. EEPROMs from 1.8 to 5 V
  - ✓ Both EEPROMs and SD cards have a limited number of write/erase cycles
  - ✓ For FRAM it is upwards of one trillion and is the highest
- **Reliability**
  - ✓ SD cards have risk of sudden failure/data loss - voltage transients, elevated temperatures, shocks, dust
  - ✓ In high-vibration environment SD cards can shake out of its slot. Soldered EEPROM or eMMC better options

OTA-TinyML Design

Confirm Smart Manufacturing



- Operational flow of OTA-TinyML:
- Part 1: Fetching TinyML Models via HTTPS
  - ✓ Upon demand, fetches ML model files from the cloud server on the edge devices
- Part 2: Store, Execute TinyML Models from Memory Units
  - ✓ Enable storage of fetched files
  - ✓ In internal memory or external files systems
  - ✓ Loading and model execution

Part 1: Fetching Models Via HTTPS

- Enable IoT devices to download ML models from the internet
- TinyML models need to be stored in the web server
  - ✓ **.bin** (model as a compressed binary file) or as **.h** (model as a C array)
  - ✓ Generate from trained models using API **TFLiteConverter.from_keras_model()**
- Download using target server address along with the directory/path of the ML models
  - ✓ Initially establish a connection to the server via **Ethernet or WiFi**
  - ✓ Then download the target from the HTTPS URL using **http.get()** method of HTTPClient object
- Passes model files to OTA-TinyML Part 2 for storage on the available memory unit of the edge device

Part 2: Store & Execute Models

- Enables storing of multiple ML models (fetched from a web server) on any memory unit of choice:
  - **Internal.** On-chip flash and SPI flash file system (SPIFFS)
  - **External.** EEPROM file system (EEFS) and SD card file system (SDFS)
- Model file is imported using #include model_name.h on which the TF Lite Micro interpreter is run to obtain predictions
- We show it is possible to load an ML model from a file instead of a C array. Interpreters works identical in both cases:
  - ✓ *Whether the model is declared as an array from the beginning*
  - ✓ *Or loaded as an array from somewhere else*
- Using this finding/concept, OTA-TinyML:
  - ✓ Initially reads the ML model in **model_name.bin** format stored in any memory unit into a byte array
  - ✓ It allocates memory for the read model using the **malloc()** function
  - ✓ Copies model **byte-by-byte** from .bin file to the MCU SRAM memory using which the interpreter produces predictions

OTA-TinyML Testing

Confirm
Smart Manufacturing

TABLE 2. Popular MCU boards (top) and optimized INT8 models (bottom) used for OTA-TinyML testing.

Name	Processor	Flash (MB)	SRAM	Clock (MHz)
B1: Teensy 4.0	Cortex-M7	2	1 MB	600
B2: STM32 Nucleo H7	Cortex-M7	2	1 MB	480
B3: Arduino Portenta	Cortex-M7+M4	16	8 MB	480
B4: Feather M4 Express	Cortex-M4	2	192 KB	120
B5: Generic ESP32	Xtensa LX6	4	520 KB	240
B6: Arduino Nano 33	Cortex-M4	1	256 KB	64
B7: Raspberry Pi Pico	Cortex-M0+	16	264 KB	133

Task: Model Name	Score	.tflite (KB)	.h (KB)
Recognize Gestures: MagicWand	0.67 (Acc)	19	118
Visual Wake Words: MicroNet S-L	0.76-0.82 (Acc)	273-529	1689-3267
Speech Recognition: Wav2letter	0.0783 (LER)	22600	143421
Keyword Spotting: DNN S-L	0.82-0.86 (Acc)	82-491	508-3029
Keyword Spotting: CNN S-L	0.91-0.92 (Acc)	75-492	436-3029
Keyword Spotting: MicroSpeech	0.62 (Acc)	18	112
Image Classification: MobileNet v2	0.69 (Acc)	3927	24215
Anomaly Detection: MicroNet S-L	0.95-0.96 (AUC)	246-452	1523-2794

- **Code.** OTA-TinyML C++ implementation .ino file flashed on B1-B7 - provided opensource:
- https://github.com/bharathsudharsan/OTA-TinyML
- **Part 1 Testing**. Fetch models from 112 KB (MicroSpeech) to 143421 KB (Wav2letter)
  - ✓ Diverse hardware spec/manufacturer
  - ✓ Successfully fetching without stalling
- **Part 2 Testing**. Four types of memory units to test onboard model storing & loading
  - ✓ Successfully stored, loaded, executed models
  - ✓ From internal memory (Flash, SPIFFS)
  - ✓ From external FS (SDFS, EEFS)

Conclusion

Confirm Smart Manufacturing

- Investigated research and engineering challenges when practicing OTA machine learning involving heterogeneous IoT devices
- OTA-TinyML was introduced to remotely repurpose (load and run the model on demand) their devices on the fly
- End-to-end fetching, storage, and execution of many ML models on a single memory-constrained device
- $3 ESP32 chip with only 4 MB flash can dynamically fetch many models from web server then execute any model upon demand

Future Work

Confirm
Smart Manufacturing

- To address challenges and bring OTA-TinyML closer to the standard OTA mechanisms of Android, edge gateways, etc.
    - ✓ Adding failure recovery mechanism
    - ✓ Adding end-to-end integrity protection
- Investigate cryptography algorithms for TinyML and IoT devices in the OTA updates context
- Investigate ML model execution and OTA updates on shredded OS such as FreeRTOS, mC/OS, RIOT, Mbed OS, Zephyr, Tock