CEC Continuing Education Center

DesignNews

Machine Learning in Microcontrollers

# DAY 1 : AI and ML for Microcontrollers

Sponsored by

Digi-Key ELECTRONICS

BENINGO EMBEDDED GROUP

informa markets

## Webinar Logistics

- Turn on your system sound to hear the streaming presentation.
- If you have technical problems, click “Help” or submit a question asking for assistance.
- Participate in ‘Group Chat’ by maximizing the chat widget in your dock.

## THE SPEAKER

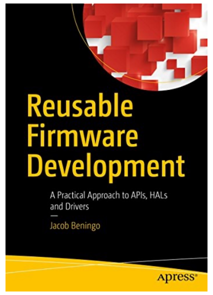
Jacob Beningo

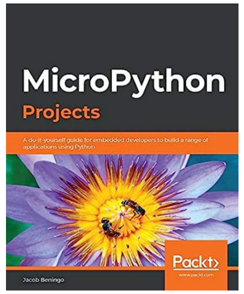
Visit 'Lecturer Profile'

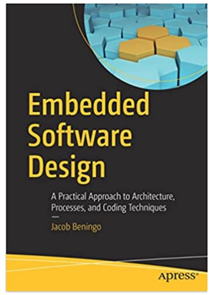
## Beningo Embedded Group - President

Focus: Embedded Software Consulting and Training

Specializes in creating and promoting embedded software **excellence** in businesses around the world.


Reusable Firmware Development
A Practical Approach to APIs, HALs and Drivers
Jacob Beningo
Apress




Embedded Software Design
A Practical Approach to Architecture, Processes, and Coding Techniques
Jacob Beningo
Apress

Blogs for:
- DesignNews.com
- Embedded.com
- EmbeddedRelated.com
- MLRelated.com

Visit www.beningo.com to learn more …

## Course Sessions

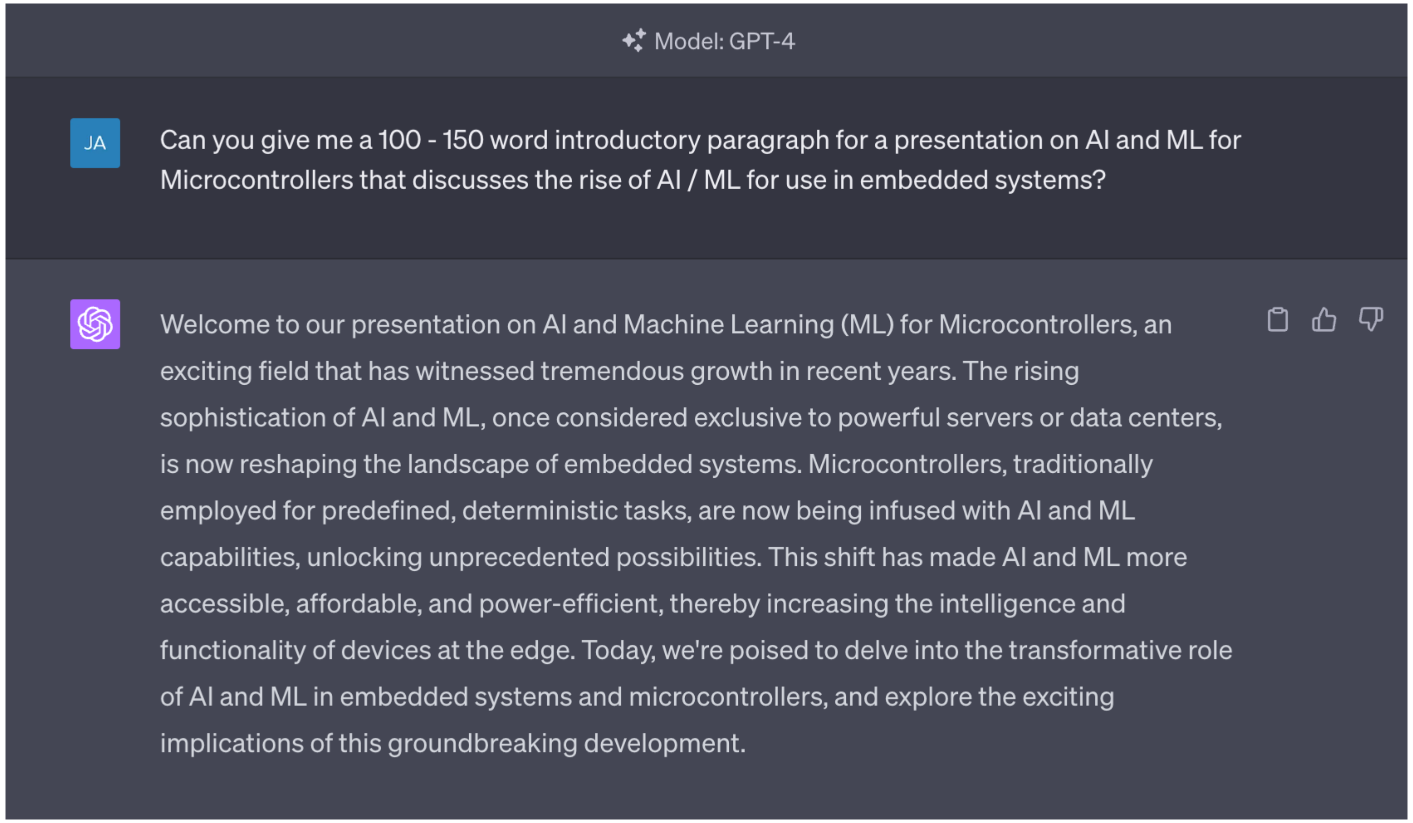
- **AI and ML for Microcontrollers**
- Writing Embedded Software with ChatGPT and Open.AI
- Tools for Machine Learning in Microcontrollers
- Training a Model for the STM32
- Deploying Machine Learning Models

Sponsored By

1

## The Rise of AI/ML

## The Rise of AI / ML

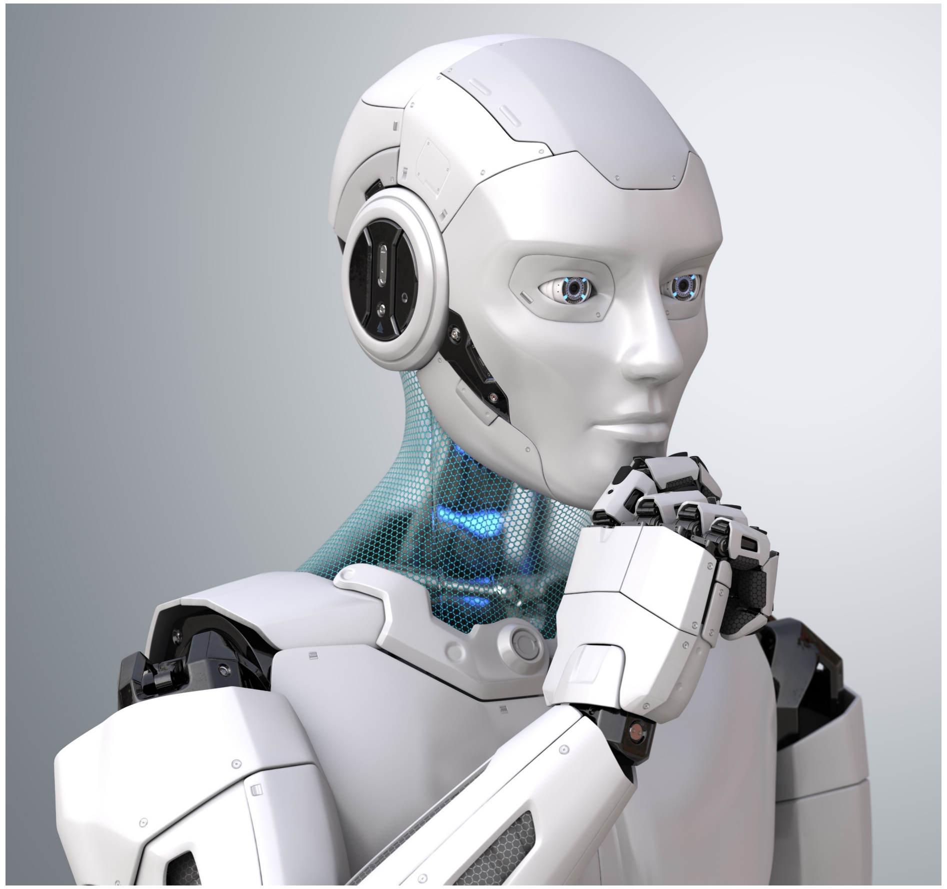

Model: GPT-4

JA Can you give me a 100 - 150 word introductory paragraph for a presentation on AI and ML for Microcontrollers that discusses the rise of AI / ML for use in embedded systems?

Welcome to our presentation on AI and Machine Learning (ML) for Microcontrollers, an exciting field that has witnessed tremendous growth in recent years. The rising sophistication of AI and ML, once considered exclusive to powerful servers or data centers, is now reshaping the landscape of embedded systems. Microcontrollers, traditionally employed for predefined, deterministic tasks, are now being infused with AI and ML capabilities, unlocking unprecedented possibilities. This shift has made AI and ML more accessible, affordable, and power-efficient, thereby increasing the intelligence and functionality of devices at the edge. Today, we're poised to delve into the transformative role of AI and ML in embedded systems and microcontrollers, and explore the exciting implications of this groundbreaking development.

## What is artificial intelligence?

"Artificial intelligence is the simulation of human intelligence processes by machines, especially computer systems. Specific applications of AI include expert systems, natural language processing, speech recognition and machine vision."



## What is machine learning?

"Machine learning is a field of computer science that often uses statistical techniques  to give computers the ability to 'learn' with data, without being explicitly programmed"

- Wikipedia

## The Rise of AI / ML

- Artificial Intelligence
  - Cloud-based
  - Generative AI
    - Write code
    - Debug code
    - Code suggestions
    - Datasheet to code converter
    - etc

- Machine Learning
  - On-chip
  - Resource constrained
  - Wide range of applications
    - Sensors
    - Classifications,
    - etc

Are you currently using AI / ML in or to develop your embedded applications?
- Yes, AI to develop and ML in my application
- Yes, AI to develop only, no ML
- Yes, no AI, but using ML in my application
- No, but within the next 6 months
- No, the distant future

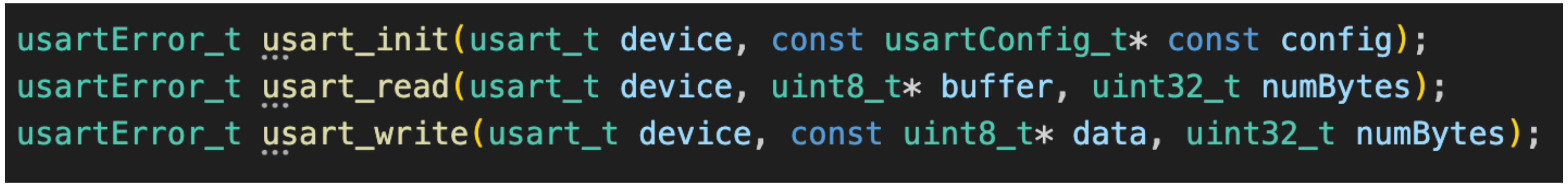
CEC Continuing Education Center

BENINGO EMBEDDED GROUP

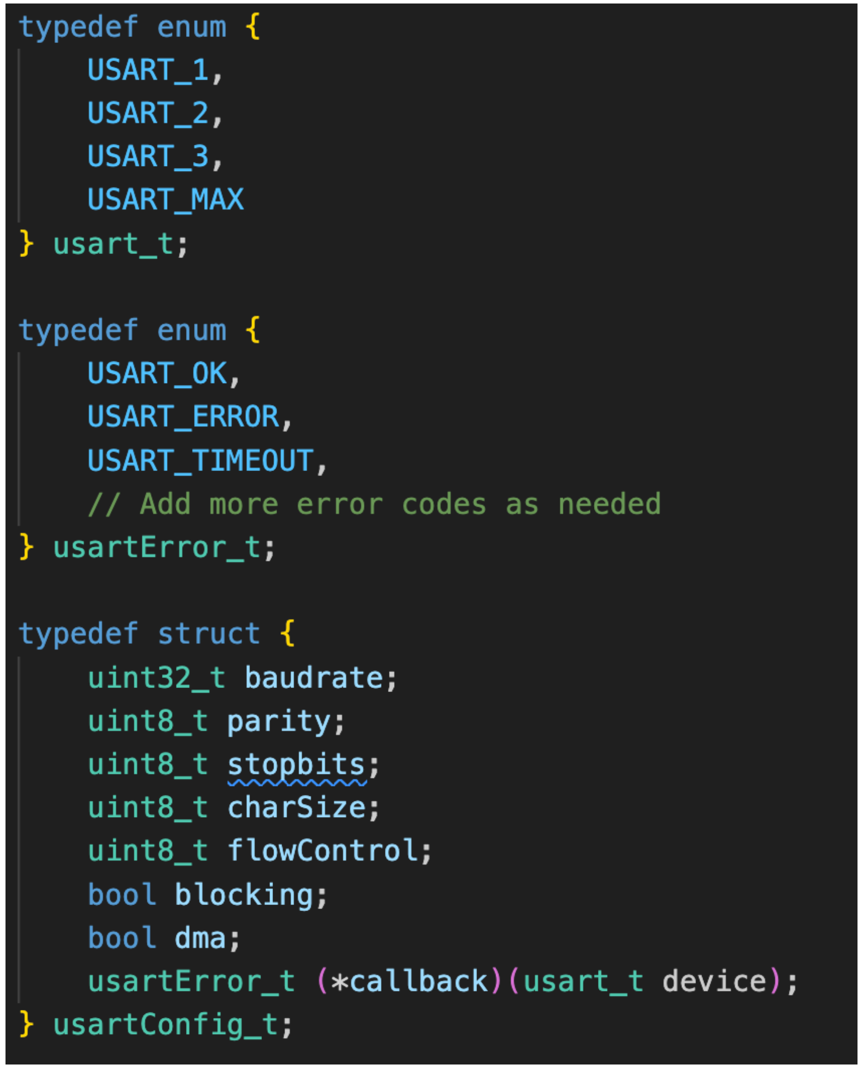
Sponsored By Digi-Key CORPORATION

## 2 AI Use Cases for Embedded

## AI Use Cases for Embedded #1 – Code Generation



```c
usartError_t usart_init(usart_t device, const usartConfig_t* const config);
usartError_t usart_read(usart_t device, uint8_t* buffer, uint32_t numBytes);
usartError_t usart_write(usart_t device, const uint8_t* data, uint32_t numBytes);
```



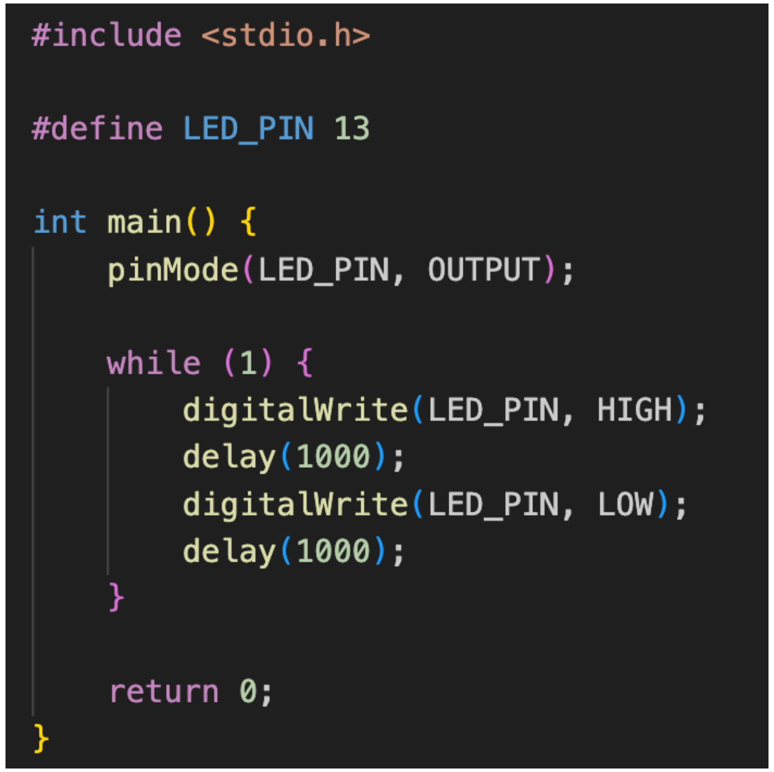
```c
typedef enum {
    USART_1,
    USART_2,
    USART_3,
    USART_MAX
} usart_t;

typedef enum {
    USART_OK,
    USART_ERROR,
    USART_TIMEOUT,
    // Add more error codes as needed
} usartError_t;

typedef struct {
    uint32_t baudrate;
    uint8_t parity;
    uint8_t stopbits;
    uint8_t charSize;
    uint8_t flowControl;
    bool blocking;
    bool dma;
    usartError_t (*callback)(usart_t device);
} usartConfig_t;
```

CEC Continuing Education Center

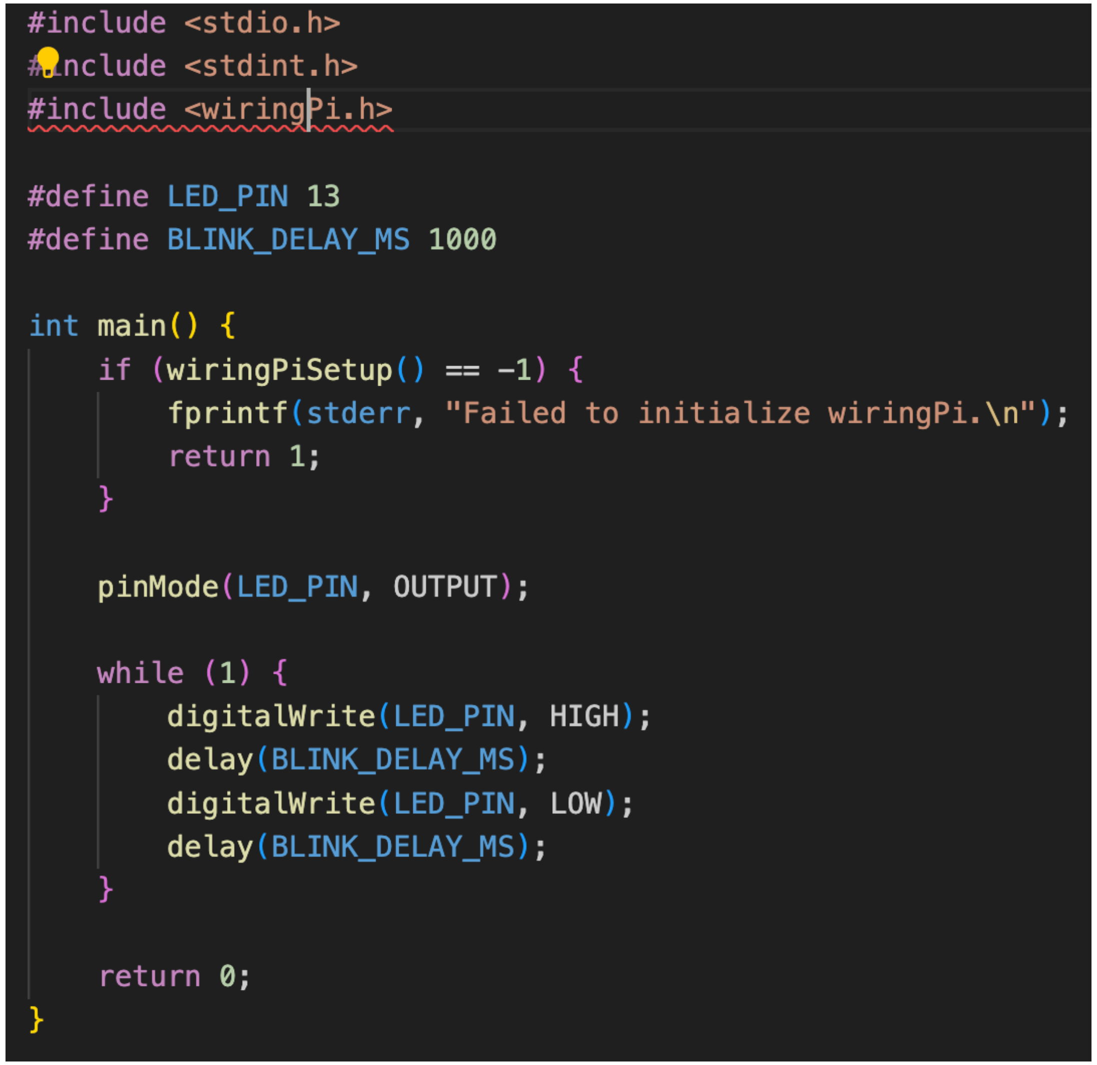
BENINGO EMBEDDED GROUP

Sponsored By Digi-Key Corporation

## AI Use Cases for Embedded #2 – Code Reviews



```c
#include <stdio.h>

#define LED_PIN 13

int main() {
    pinMode(LED_PIN, OUTPUT);

    while (1) {
        digitalWrite(LED_PIN, HIGH);
        delay(1000);
        digitalWrite(LED_PIN, LOW);
        delay(1000);
    }

    return 0;
}
```



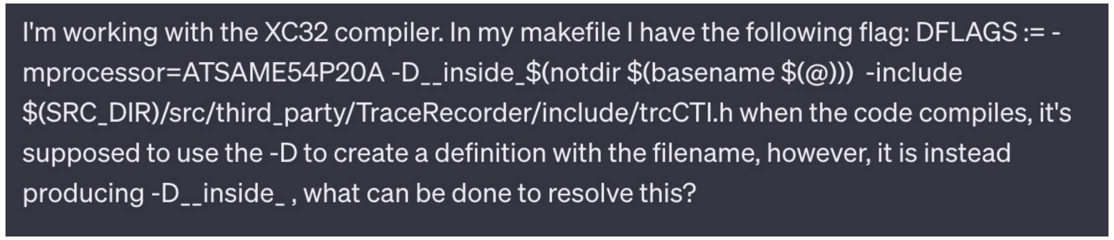
```c
#include <stdio.h>
#include <stdint.h>
#include <wiringPi.h>

#define LED_PIN 13
#define BLINK_DELAY_MS 1000

int main() {
    if (wiringPiSetup() == -1) {
        fprintf(stderr, "Failed to initialize wiringPi.\n");
        return 1;
    }

    pinMode(LED_PIN, OUTPUT);

    while (1) {
        digitalWrite(LED_PIN, HIGH);
        delay(BLINK_DELAY_MS);
        digitalWrite(LED_PIN, LOW);
        delay(BLINK_DELAY_MS);
    }

    return 0;
}
```

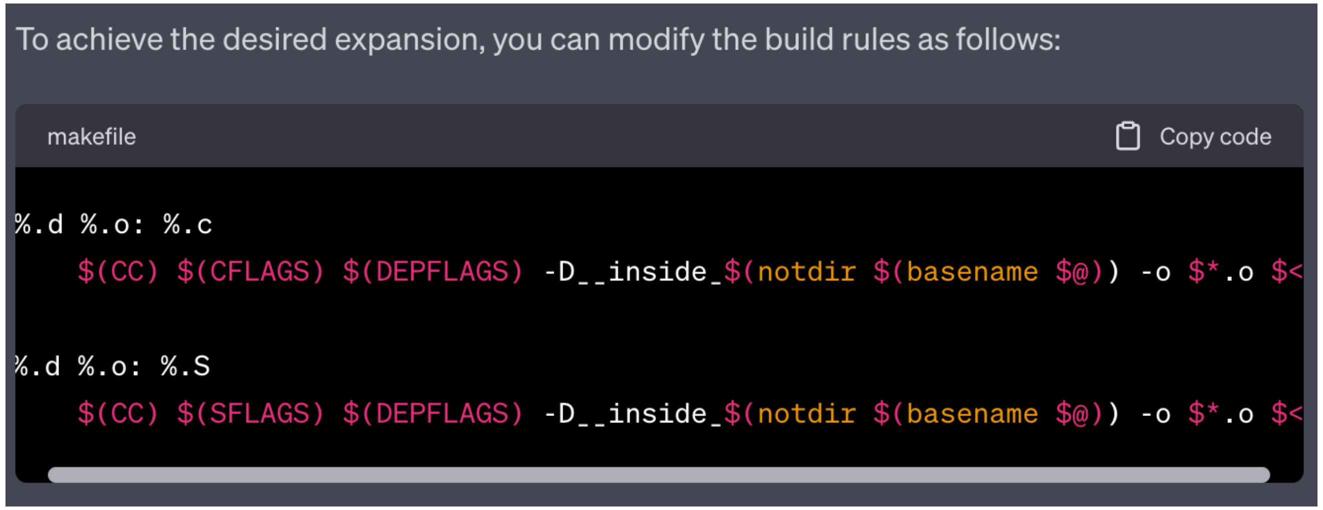
CEC Continuing Education Center

BENINGO EMBEDDED GROUP

Sponsored By

# AI Use Cases for Embedded #3 – Code Debugging


I'm working with the XC32 compiler. In my makefile I have the following flag: DFLAGS := -mprocessor=ATSAME54P20A -D__inside_$(notdir $(basename $(@)))  -include $(SRC_DIR)/src/third_party/TraceRecorder/include/trcCTI.h when the code compiles, it's supposed to use the -D to create a definition with the filename, however, it is instead producing -D__inside_ , what can be done to resolve this?


To achieve the desired expansion, you can modify the build rules as follows:

makefile

Copy code

```makefile
%.d %.o: %.c
	$(CC) $(CFLAGS) $(DEPFLAGS) -D__inside_$(notdir $(basename $@)) -o $*.o $<

%.d %.o: %.S
	$(CC) $(SFLAGS) $(DEPFLAGS) -D__inside_$(notdir $(basename $@)) -o $*.o $<
```

What application are you most interested in using AI for?
- Code Generation
- Code Reviews
- Debugging
- Other

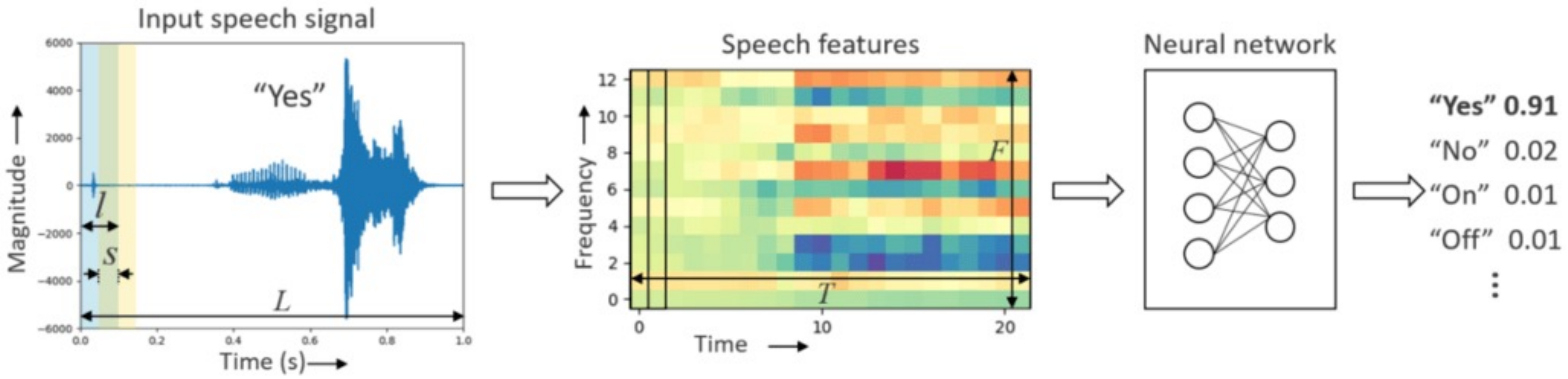
CEC Continuing Education Center

BENINGO EMBEDDED GROUP

Sponsored By Digi-Key CORPORATION

## 3 ML Use Cases

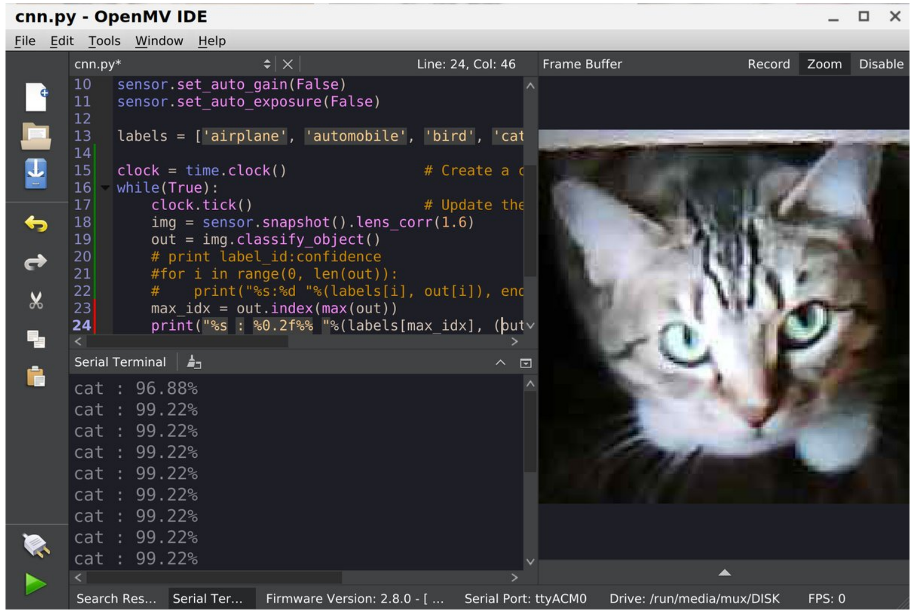
## ML Use Cases #1 – Keyword Spotting



Source: Arm

## AI / ML Use Cases #2– Image Recognition

OpenMV Cam with a Cortex-M7


cnn.py - OpenMV IDE
File Edit Tools Window Help
cnn.py*
Line: 24, Col: 46
Frame Buffer
Record Zoom Disable
Serial Terminal
Search Res... Serial Ter... Firmware Version: 2.8.0 - [ ... Serial Port: ttyACM0 Drive: /run/media/mux/DISK FPS: 0

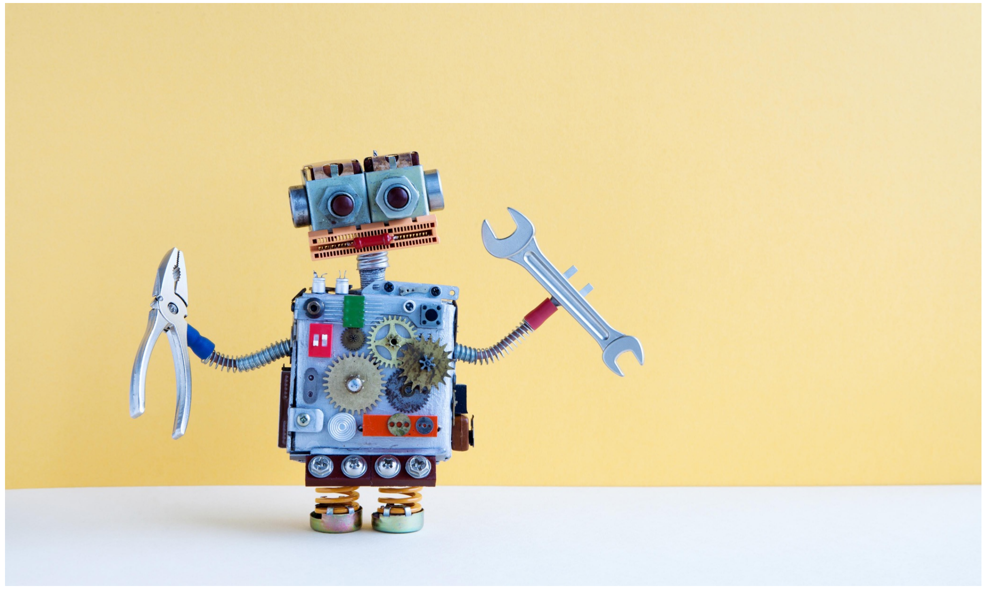
```
sensor.set_auto_gain(False)
sensor.set_auto_exposure(False)

labels = ['airplane', 'automobile', 'bird', 'cat

clock = time.clock()                # Create a c
while(True):
    clock.tick()                    # Update the
    img = sensor.snapshot().lens_corr(1.6)
    out = img.classify_object()
    # print label_id:confidence
    #for i in range(0, len(out)):
    #    print("%s:%d "%(labels[i], out[i]), enc
    max_idx = out.index(max(out))
    print("%s : %0.2f%% "%(labels[max_idx], (out
```

```
cat : 96.88%
cat : 99.22%
cat : 99.22%
cat : 99.22%
cat : 99.22%
cat : 99.22%
cat : 99.22%
cat : 99.22%
cat : 99.22%
```

Video : https://www.youtube.com/watch?v=PdWi_fvY9Og

## AI / ML Use Cases #X – Choose your own adventure!

- Gesture classification
- Anomaly detection
- Analog meter reader
- Guidance and Control (GNC)
- Game AI
- Package detection
- (a plethora of applications)



What application are you most interested in using ML for?
- Keyword spotting
- Image classification
- Predictive maintenance
- Other

Sponsored By

4

## Going Further

## AI and ML Resources

- Jacob's AI Blogs
- Jacob's CEC courses
- Jacob's ML Blogs

- Embedded Bytes Newsletter
  - http://bit.ly/1BAHYXm

www.beningo.com

CEC Continuing Education Center

DesignNews

# Thank You

Sponsored by

Digi-Key ELECTRONICS

BENINGO EMBEDDED GROUP

informa markets