Best Practices for Designing Real-Time Embedded Systems

DAY 2 : Designing a Hardware-less System

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 'Attendee Chat' by maximizing the chat widget in your dock.

Course Sessions

- System Level Design Philosophy
- Designing a Hardware-less System
- It's All About the Data
- Testing Your Way to Design Success
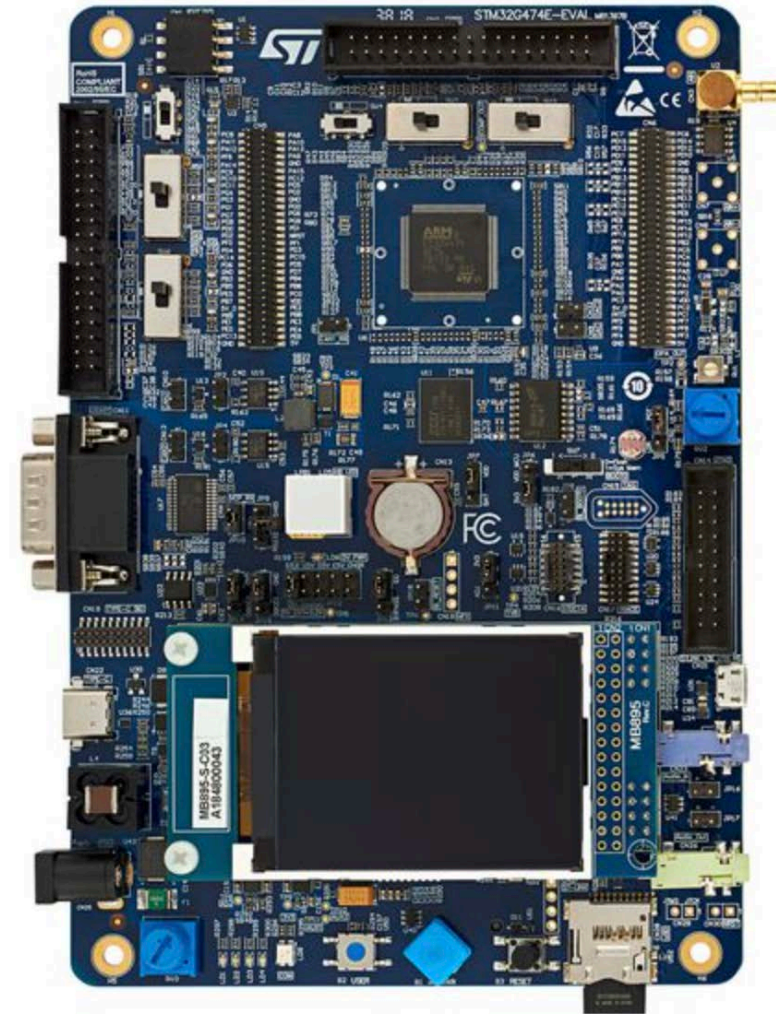- The Best Practices Lightning Round

1 Traditional Embedded Software Dev

Embedded software development is where the bits and bytes meet the hardware.

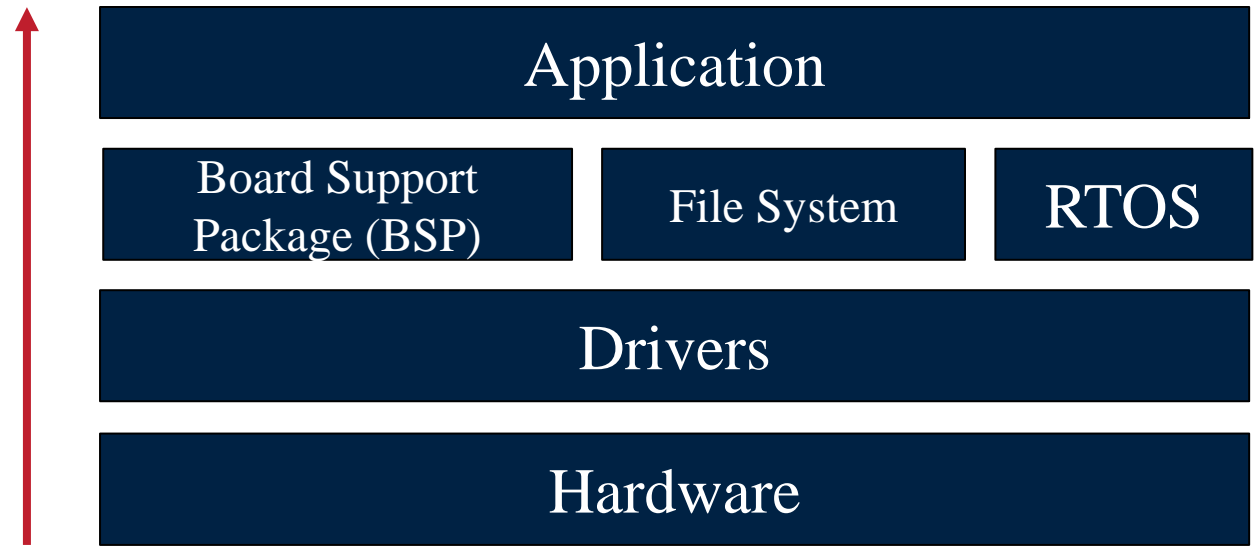# Traditional Embedded Software Dev

Hardware Centric Development

- Microcontroller / Processor Focused
- Drivers and Peripherals
- Memory Maps
- External sensor chips
- Development boards
- Shields

Traditional Embedded Software Dev

Bottom-Up Approach

- Review MCU Datasheet
- Get a dev board
- Write Drivers
- Integrate Middleware
- Develop application modules
- Test the system

Application		
Board Support Package (BSP)	File System	RTOS
Drivers		
Hardware		

Traditional Embedded Software Dev

Software Development Model

- Application is written on the hardware
- Traditional IDE / Debug environment
- Little to no simulation or modeling
- Testing is often spot checked
- Generally can be very successful

When you start developing software what is the first software layer you start to develop for?

- Drivers
- RTOS
- Middleware
- Application
- Other

2 Hardware-less Embedded Software

Modern Embedded software development abstracts the hardware and focuses on application feature development.

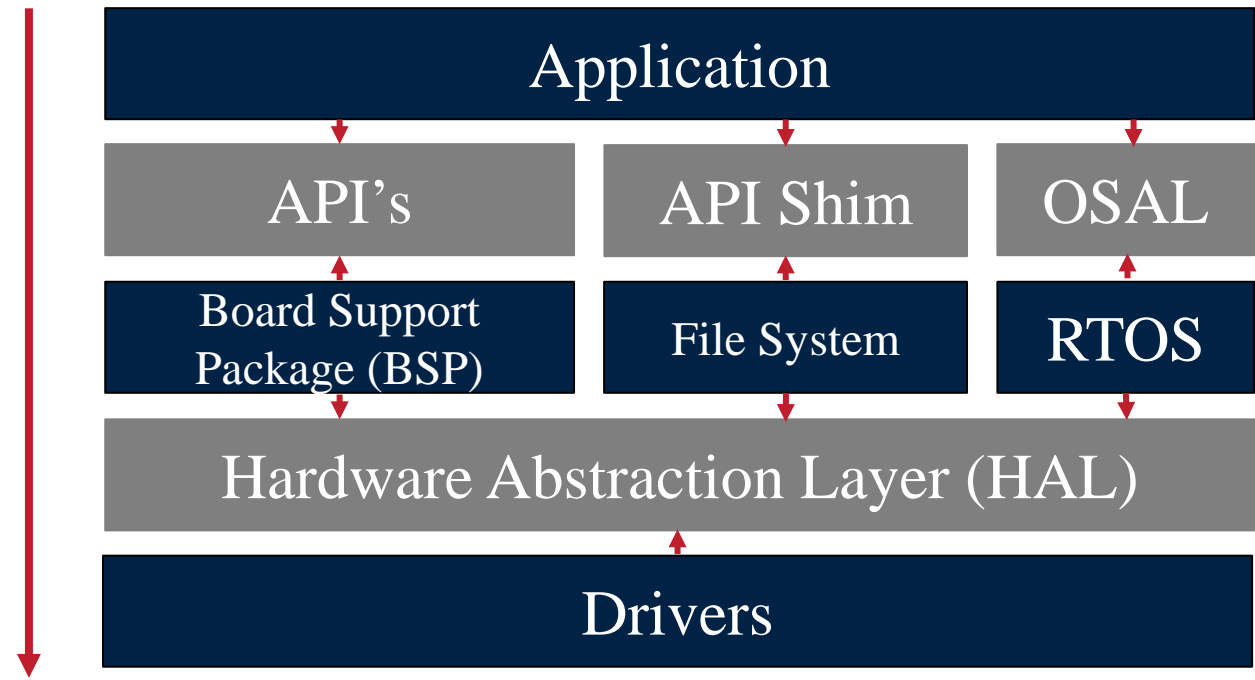# Hardware-less Embedded Software

Application Centric Development

- Feature Focused
- Hardware Independent
- Data Driven
- Leverages Abstractions
- Highly Testable
- Utilizes Automation

Hardware-less Embedded Software

Top-Down Approach
- Feature Breakdown
- Model and simulate features
- Off-target development
- Test with a harness
- Integrate on final hardware
- Hardware critical tested early

Application		
API's	API Shim	OSAL
Board Support Package (BSP)	File System	RTOS
Hardware Abstraction Layer (HAL)		
Drivers		

Hardware-less Embedded Software

Software Development Model

- Application is written off target
- Unit testing harnesses for debug
- Simulation and modeling to prove design
- Application never "touches" hardware
- Integrated with target hardware
- Can highly automate regression and deployment

Which software development model is more appealing to you?

- Traditional Embedded Software Dev
- Hardware-less Embedded Software Dev
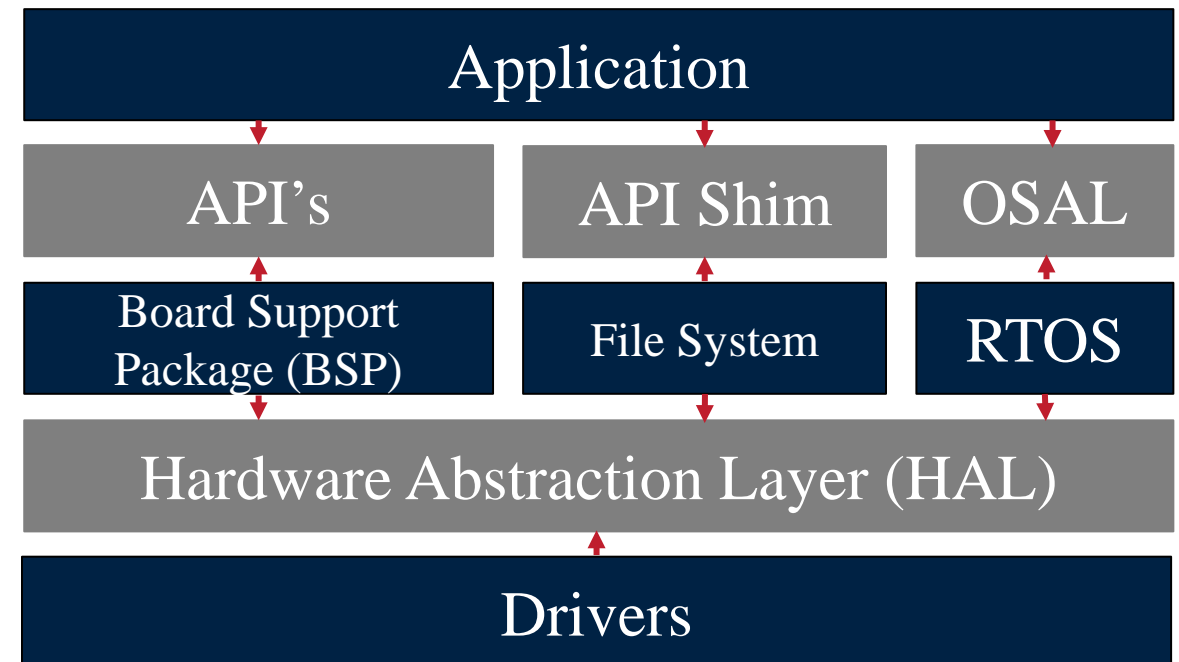- Undecided
- Other

3 Simulation and Software Architecture

Modern embedded software focuses on simulation, abstraction and testing as much as possible with the hardware "out-of-loop".

BENINGO EMBEDDED GROUP

Simulation and Software Architecture

Software Architecture

- Describes major components
- High-level
- Shows how components interact
- Hardware independent
- Multiple views to explore and define the software behavior

Application		
API's	API Shim	OSAL
Board Support Package (BSP)	File System	RTOS
Hardware Abstraction Layer (HAL)		
Drivers		

Simulation and Software Architecture

Simulation

- Test application in a PC environment
- Run more iterations in shorter time
- Experiment and prove assumptions
- Avoid compile, deploy and debug time
- Auto-generate code
- Operates on abstraction w/o hardware

Simulation and Software Architecture

Operate on data not hardware!

```
void Valve_Set(uint8_t const Valve, State_t const State)
{
    if(State == OPEN)
    {
        HAL_GPIO_Write(Valve, GPIO_SET);
    }
    else
    {
        HAL_GPIO_Write(Valve, GPIO_RESET);
    }
}
```

```
void Valve_Set(Valve_t * Valve, State_t const State)
{
    if(State == OPEN)
    {
        *Valve.State = OPEN;
    }
    else
    {
        *Valve.State = CLOSE;
    }
}
```

Simulation and Software Architecture

Operate on data not hardware!

```
uint8_t Valve = 3;

Valve_Set(Valve, OPEN);
```

```
Valve_t Valve = {3, CLOSE};

Valve_Set(&Valve, OPEN);

// Valve not open yet! Just the data state is updated!
Valve_HardwareUpdate(Valve);

// Now the hardware is updated!
```

How do you currently write your software?

- Operating on data
- Operating on hardware
- Other

Thank you for attending

Please consider the resources below:
- www.beningo.com
  - Blog, White Papers, Courses
  - Embedded Bytes Newsletter
    - http://bit.ly/1BAHYXm

From www.beningo.com under
   - Blog > CEC – Best Practices for Real-Time Embedded Systems

Thank You

Sponsored by