



# Life With *Devices*

黄季冬

Microsoft MVP

Device Application Development

# Agenda

A World of Services & Devices

Device Oriented Programming

.Net Micro Framework

Q & A

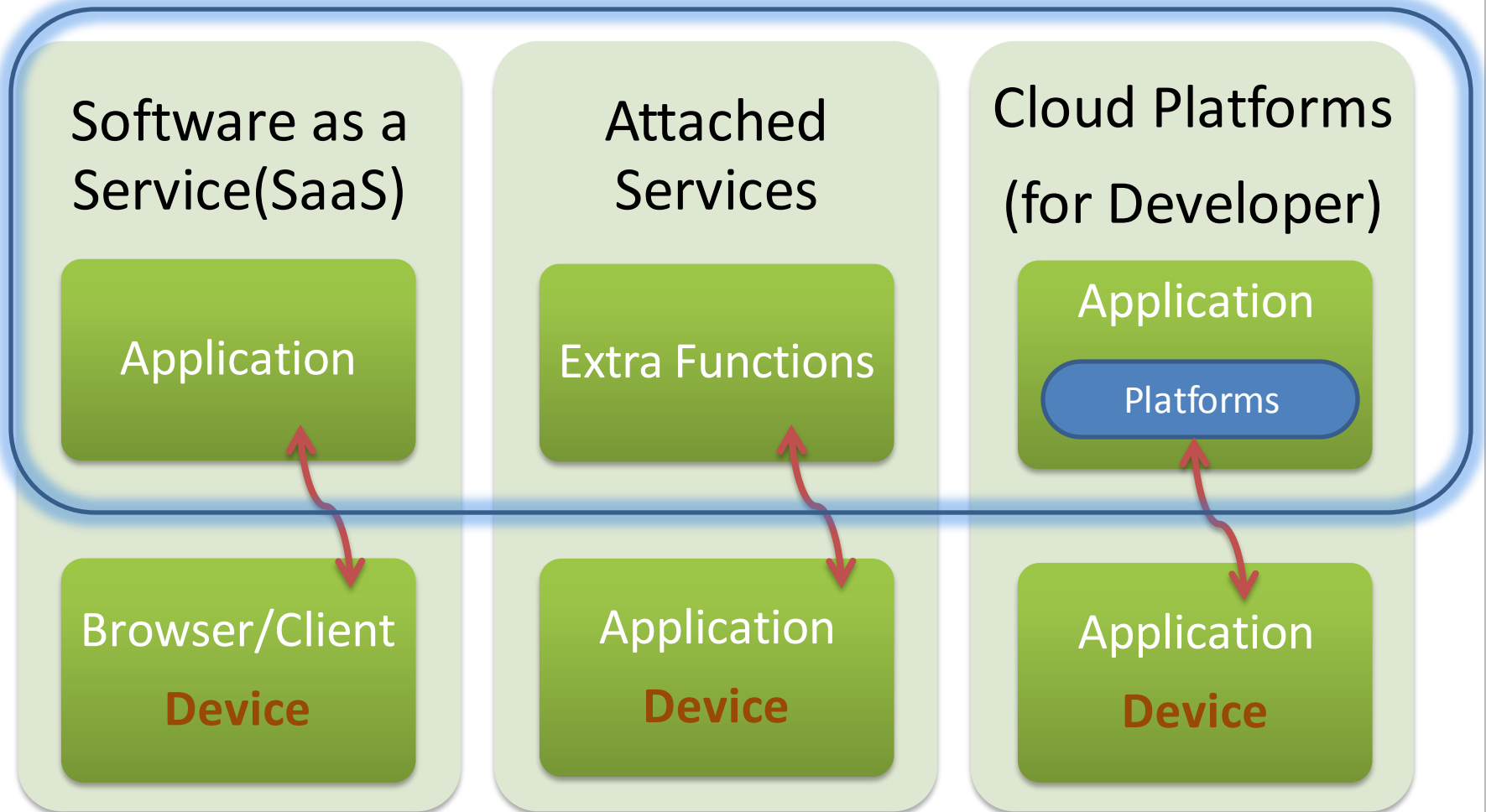
# A World of Services

闹铃，餐饮，公共交通，移动通信

供电供水，网络，广告，媒体



# Cloud is not all



**How many devices in the video clip?**



# I'm a Device



I'm Smart



I'm  
Personal



I'm  
Everywhere

# Agenda

A World of Services & Devices

Device Oriented Programming

.Net Micro Framework

Q & A

# Development on Device(Old Ways)

The screenshot displays the Metrowerks CodeWarrior IDE interface. On the left, the Project Workspace shows a tree view of the project structure, including folders like Startup, Drivers, GLCD, CDF73, and Common\_Source, along with various source files. The main window shows the linker table for the project 'Generic\_Attitude-9S12.C'. The linker table has columns for File, Code, and Data, and lists various components of the build, such as Debugger Cmd Files, Libraries, Linker Map, and User Modules. The code size for the User Modules is 112K, and the data size is 7K. The linker table also shows the size of individual source files, such as dtc01.c (15858 bytes code, 100 bytes data) and dtc02.c (11419 bytes code, 0 bytes data).

File	Code	Data
Debugger Cmd Files	0	0
Debugger Project File	0	0
Doc	0	0
Generated Code	2K	533
Libraries	18K	1K
Linker Map	0	0
Prm	0	0
readme.txt	n/a	n/a
Sources	157	1
Startup Code	69	24
User Headers	0	0
User Modules	112K	7K
alerts.c	1472	11
BUSSIM.C	110	2
CustomCodePage.c	0	256
DiagModule.c	1114	0
DisplayMenu.c	1078	958
DisplayScreens.c	5239	1080
dtc01.c	15858	100
dtc02.c	11419	0
dtcctrl.c	960	0

```
bre:
case 1
PSS:
PSS:
bre:
case 1
PSS:
PSS:
bre:
case 1
PSS:
PSS:
bre:
defau.
bre:
}
}
#endif
// init_ua:
// init_UA:
// init_ad:
// initPer:
#ifdef TRUC
q Attitude
```



# Serial IO in C

```
//global timer variable for ms timer
unsigned long g_timerMS = 0;
//timer interrupt
__irq void timerISR(void)
```

```
{
    T1IF //function to put a character to the serial port
    int putchar (int ch)
    {
        g_ti // special case for line feed
        // send a CR and then the line feed below
        VICV // a
        if (ch == '\n')
        {
            // make sure the UART buffer is empty
            while (!(U1LSR & 0x20))
            {
                //function to get a character from the serial port
                int getchar (unsigned long msTimeout)
                {
                    // variable to;
                    // wait for a character to be available
                    while (!(U1LSR & 0x01))
                    {
                        //if timeout has expired re
                        if(g_timerMS - lastTime >
                    }
                }
            }
            // return the character received
            return (U1RBR);
        }
    }
}
```

```
//serial message to send
const int g_msg = {'h', 'e', 'l', 'l', 'o'};
const int g_msgLen = 5;

void main(void)
{
    int i;
    int input[5];

    T1TCR=2;
    //Initialize Timer 1 for 1 ms clock
    T1PR = 5999; // number of clock ticks before incrementing the TC
    T1MR0 = 10; // value for TC to match to generate an interrupt
    T1MCR = 3; // Interrupt and Reset on MRO
    T1TCR = 1; // Start Timer
    T1EMR = 0; // External Match register set.

    VICVectAddr0 = (unsigned long) timerISR; // set interrupt vector
    VICVectCntl0 = 0x20 | 5; // use it for Timer1 Interrupt
    VICIntEnable |= 0x00000020; // Enable Timer1 Interrupt

    //send the message on the serial port
    for (i = 0; i < g_msgLen; i++)
        putchar(g_msg[i]);

    //wait for 5 bytes of data for up to 10 seconds
    for (i = 0; i < 5; i++)
    {
        //receive a character, waiting at most 10 seconds
        input[i] = getchar(10000);

        //if we timed out, exit loop
        if (input[i] == -1)
            break;
    }
}
```

# Small device background

Inflection point in shift from 8/16 bit to 32 bit processors

Time to market 1-2 years -> huge market risk

Software development is 80-90% of that effort

Proliferation of low power wireless protocols

Market demands for more complex products -> increasing time to market

Desperate need for productivity increases

- Movement to standardized OSs and higher level languages

Accelerating Innovation

- For small, simple devices by bringing modern computing paradigms from the desktop

# Device abstract from I/O Interface

Interrupt can be a TYPE

Serial Port Device

SPI Device

I<sup>2</sup>C Device

Any Peripheral Device

# We Want Serial IO in OO way

```
class serial
{
    // static variable for the serial port object
    private static SerialPort portSerial;

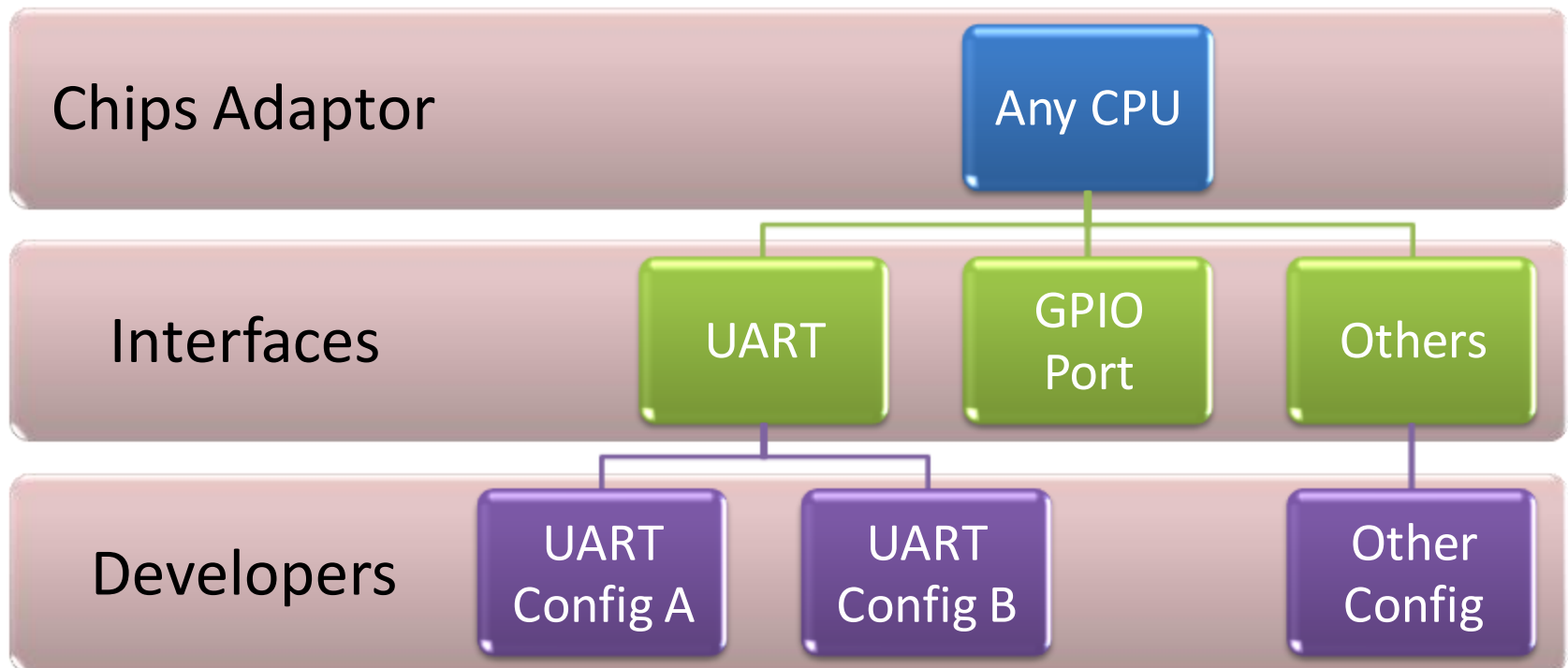
    // main entry point
    public static void Main()
    {
        // create the serial port object
        portSerial = new SerialPort(new
        SerialPort.Configuration(SerialPort.Serial.COM1,
        SerialPort.BaudRate.Baud9600, false));

        // create a set of bytes 'hello'
        byte[] bytes = { (byte)'h', (byte)'e', (byte)'l', (byte)'l', (byte)'o' };

        // write the bytes out the serial port
        portSerial.Write(bytes, 0, bytes.Length);

        // wait for 5 bytes of data for up to 10 seconds
        portSerial.Read(bytes, 0, 5, 10000);
    }
}
```

# A Simple DOP Model



# Agenda

A World of Services & Devices

Device Oriented Programming

.Net Micro Framework

Q & A

# .NET Micro Framework

## Small is Beautiful

### Version 3 launched!

Boosts secure connectivity,  
adds native code interoperability

### increased productivity ↗

Visual Studio 2008  
Visual C# Express



Touch  
Gesture

### more hardware choices

Interoperability  
with Native Code,  
64K RAM



### secure connectivity

SSL  
USB  
Wi-Fi



# Overview of .NET Micro Framework

- A merging of .NET and embedded technologies built specifically for the lower end of the 32 bit processor space
- Complement to the existing Microsoft embedded offerings
- Value proposition: Bringing the modern computing models found on the desktop (.NET, DPWS,...) to the embedded space to address the increasing connectedness and complexity of new device scenarios

NITE 国家信息技术紧缺人才培养工程  
National Information Technology Education Project  
国家信息技术紧缺人才培养工程系列丛书

## .Net Micro Framework 嵌入式开发入门与典型实例

信息产业部软件与集成电路促进中心 编著

Best Practice .Net Micro Framework





# 32bit, Small Bootable Runtime

## .NET tailored to deeply embedded devices

- Purpose built from the ground up
- Managed Drivers
- WPF
- Different compatibility model

## Porting to target hardware

- Simpler platform/simpler port
- 1-2 man months

## Resource constrained

- Low end 32 bit w/o MMU, w/o external RAM, less power consumption – lower BOM

## Complete Visual Studio Integration

- On board debugging
- Extensible Emulation on the PC

## New Paradigm for Small, Connected Device Development

Managed Code for applications on smart, connected devices

Fully integrated with Microsoft® Visual Studio®

Full-featured debugging on device

Familiar tools reduce total cost of development

Increased productivity and faster time to market

# User Interface / Shell

Object Model based on Windows Presentation Foundation (WPF)

Input event routing

Layout system

- Content sizing
- Text flow
- Rich support for nested controls

Bitmap fonts

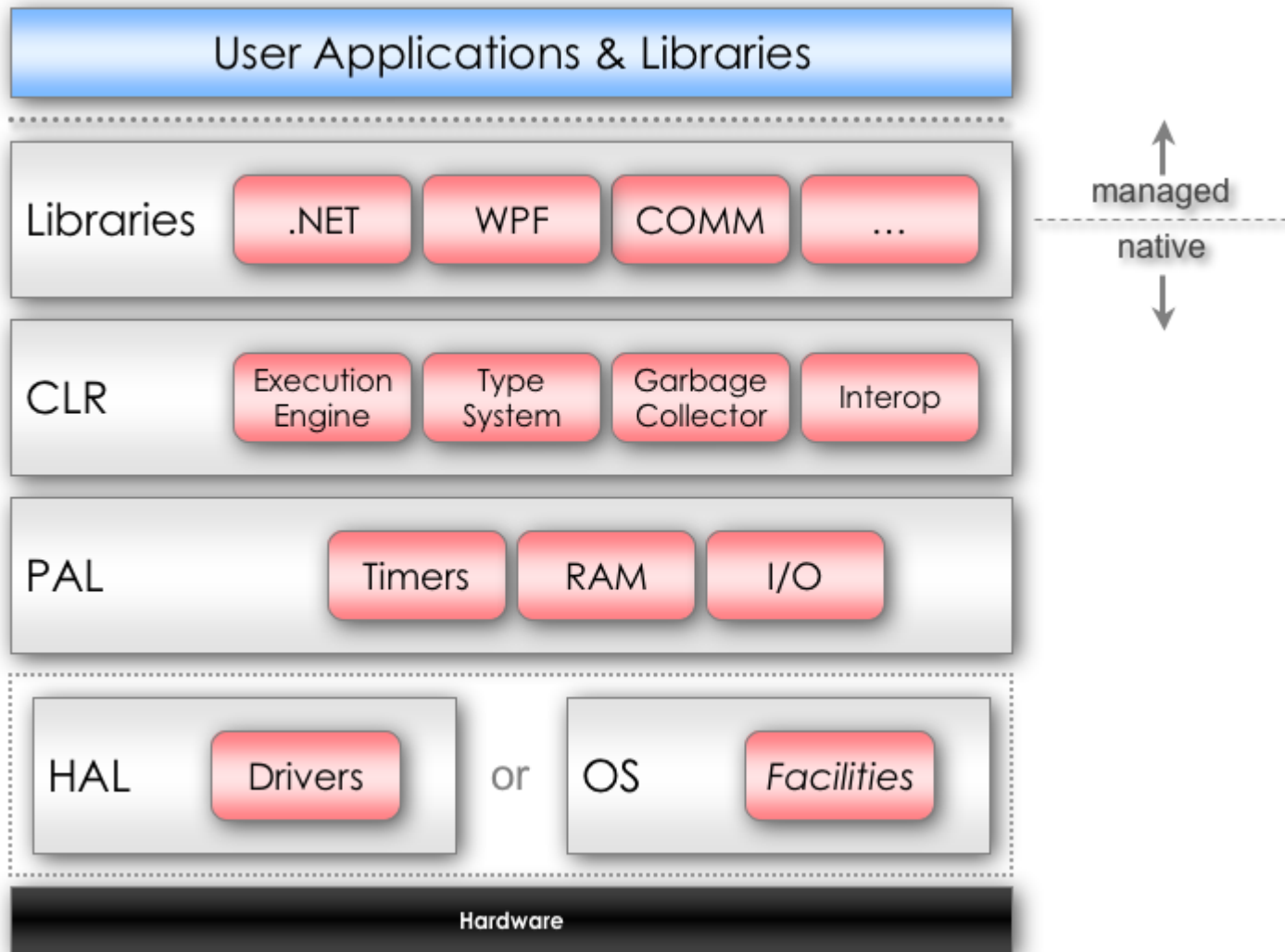
Images

Pens, brushes, colors

Touch/Gesture



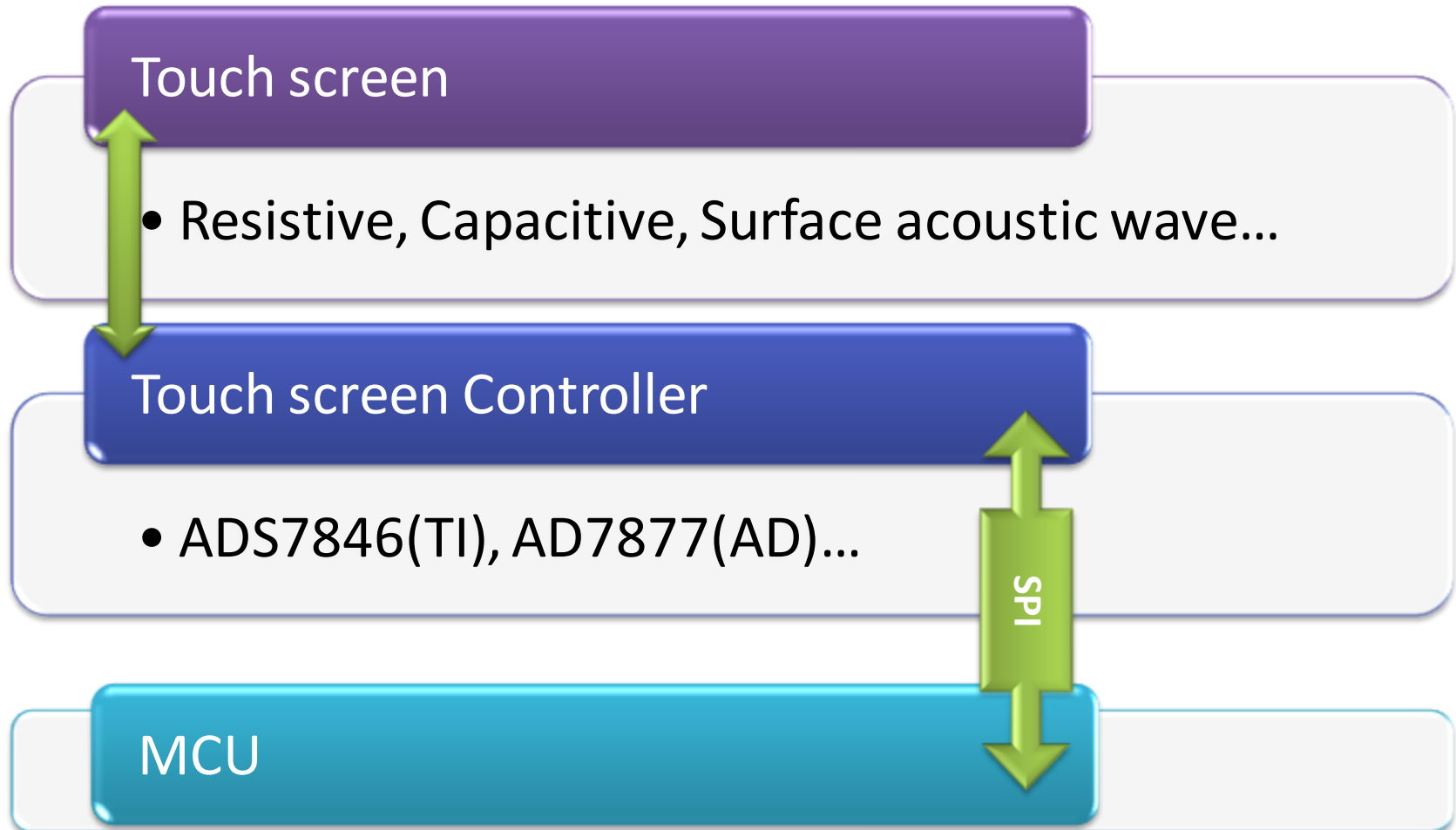
# Architecture



# New Features in Version 3.0

- ADI Libraries
  - Enabling Blackfin Family
- Native Code Interoperability
  - Users Extending the .NET OM
- USB Device
  - Managed code access to USB
- Touch/Gesture support
  - Fully integrated with WPF infrastructure
- **File System**
  - In addition to EWR, FAT32, SD Card compatibility
- DPWS Code Generation
  - WSDL or C# -> Client and Server stubs
- SSL support
  - Extending the previously released TCP/IP Stack
- Footprint reductions
  - **256kFlash, 64K RAM**
- 'Solutions Wizard'
  - Auto-generate build files
- GCC compatibility on ARM
- WiFi integration
- Visual Studio 2008 support
- More accessible Porting Kit

# Touch screen in .Net Micro Framework



# New Development Hardware

- Analog Devices International
  - Blackfin EZ-Lite
- GHI Electronics
  - GHI CANExtra Box
  - USBizi chip
  - USBizi Development Kit
  - Embedded Master Module
  - Embedded Master Dev Kit
- Emtrion
  - HiCO.ARM9 Starter Kit
- Device Solutions
  - Tahoe II



# Resources

## Links

- <http://microsoft.com/netmf>
- <http://fox23.cnblogs.com> (My blog)

## Books

- 《.Net Micro Framework 嵌入式开发入门与典型实例》
- [Embedded Programming with the .Net Micro Framework](#)
- [Expert .Net Micro Framework](#)



THANKS!

[hjd.click@gmail.com](mailto:hjd.click@gmail.com)

Questions?