

An SDK For Ultrasonic Imaging on a Windows Mobile Cell Phone

David M. Zar

Washington University in St. Louis

4 February 2009

dzar@wustl.edu

Why? (Motivation)

- USB ultrasound probes have been around since 2006 and are ideally suited for portable and low-cost applications:
 - » First responder
 - » Starting IVs and central lines
 - » Urology (Bladder volume/catheter)
 - » Primary care in underserved/developing areas
 - » Veterinary uses
- Carrying around a laptop is cumbersome and expensive.
- Cell phones are ubiquitous and easy to use.
- Cell phones have a *network* built in for image transmission and retrieval.

Who are We?

- William D. Richard
Associate Professor, Washington University in St. Louis
- David M. Zar
Research Associate, Washington University in St. Louis
- Z&R Technologies, LLC develops fully integrated USB ultrasound probes:
 - » Company has license from the University and continues development of the probes.
 - » All of the electronics are in the probe, itself.
 - » The probe, a USB cable and your imaging platform is all you need.
 - » No external power or hardware is required.
 - » Technology has been licensed by multiple manufacturers and commercial computer/laptop-based systems have been on the market for over three years.
 - » The probes are FDA approved and have received CE marking approval.
- Working under a Microsoft Research grant, *Cell Phone as a Platform for Healthcare*, we modified probes to work with a smart phone and developed the SDK presented, here.

Cell Phones

- The most significant requirement for the cell phone is that it has a **host** USB port.
- Windows Mobile 5/6 is supported.
- Even a 260 MHz processor works!



+



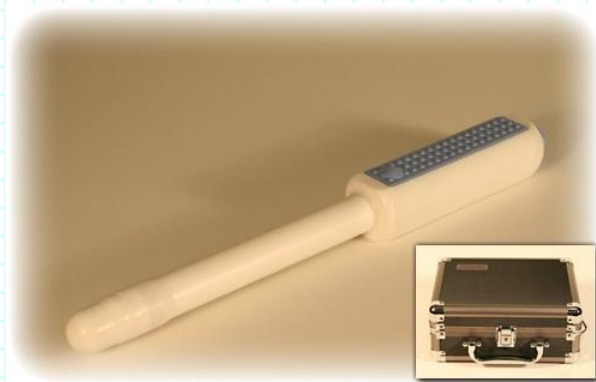
USB Ultrasound Probes



5.0 MHz GP Probe



12 MHz Ophthalmic Probe

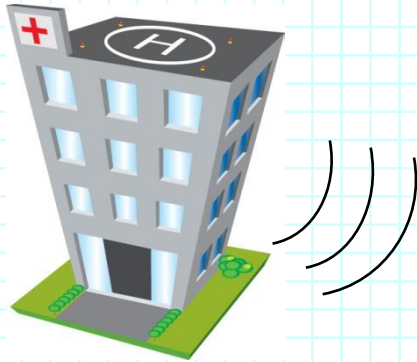


7.5 MHz Endocavity Probe

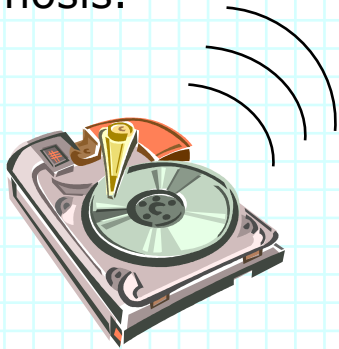


7.5 MHz Vascular Access Probe

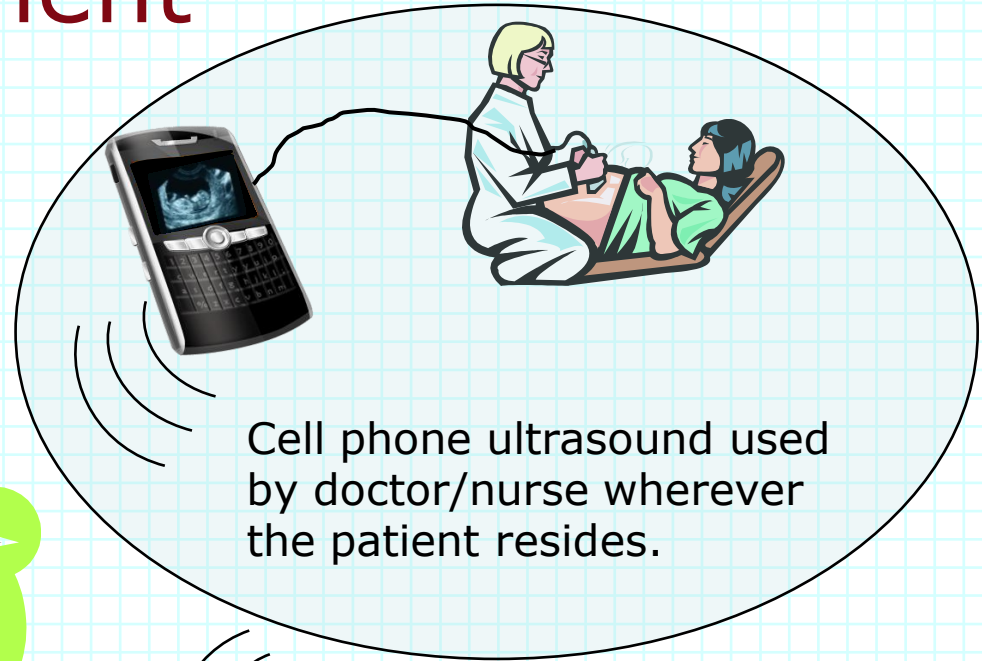
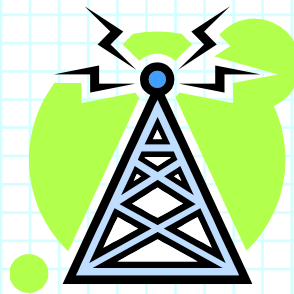
Imaging Environment



Send image and patient data to hospital for diagnosis.



Store data anywhere (HealthVault, GoogleHealth, etc.)



Cell phone ultrasound used by doctor/nurse wherever the patient resides.



Remotely view data or retrieve it from databases.

SDK Features

- The SDK is based on functionality already available in commercial systems using the USB probes:
 - » Complete control of the probe is available via a provided DLL
 - Detect attachment/detachment of probes
 - Initialize and configure the probe
 - Configure cine buffers
 - Start/stop probe data acquisition
 - » A-mode and B-mode functionality is provided through a second DLL
 - Setup scan conversion parameters
 - Image display
 - Optional image overlay
 - Configure TGC (Time Gain Correction)
 - Utility functions

SDK Development

- The SDK works with Microsoft Visual Studio 2008
 - » Requires the Windows Mobile 6 SDK (also includes WM5 support).
- A fully-functioning laptop/desktop-based Windows XP/Vista application is included with probes for testing and determining optimal imaging settings.
- A fully-functioning Windows Mobile sample application in C# with full source code is included.
- Cost: FREE (under BSD-style permissive license).
- Probes are available for a nominal cost to developers.

SDK Examples

■ Initialize the Probe and Display

```
UInt32 probeExistStatus = UsbProbeDll.usbFindProbes(pChar, hImageWindow);

Display = BmodeDll.bmInitializeDisplay(winXSize * winYSize, 0);
BmodeDll.InitializeDIB(ref bmi, winXSize, winYSize);
BmodeDll.bmSetDisplayOffset(0, 0);

// initialize user probe
hProbe = UsbProbeDll.usbUserProbeHandle();
UsbProbeDll.usbInitializeProbe(hProbe, winYSize, 1540, 0x07);

if (!UsbProbeDll.usbHardwareDetected())
{
    UsbProbeDll.usbSelectProbe(hProbe, 0);
} else {
    UsbProbeDll.usbInitializeProbes(winYSize, 1540);
    UsbProbeDll.usbSelectProbe((System.IntPtr)null, 0);
    UsbProbeDll.usbProbeHandle(0, ref hProbe);
}

ProbeID = UsbProbeDll.usbProbeID(hProbe);
```

SDK Examples

■ Setup probe imaging parameters

```
UsbProbeDll.usbSetWindowDepth(hProbe, winYSize);  
UsbProbeDll.usbSetVelocity(hProbe, 1540.0f);  
UsbProbeDll.usbSet30MHzClock(hProbe);  
// and set the clock divider for 1: ~ 5cm @ 30MHz and 1540 m/s velocity  
UsbProbeDll.usbSetClockDivider(hProbe, 4);  
UsbProbeDll.usbSetProbeFrequency(6.0f);  
UsbProbeDll.usbSetUnidirectionalMode();  
UsbProbeDll.usbTurnOnSync();  
UsbProbeDll.usbSetPulseVoltage(100.0f);
```

SDK Examples

■ Setup image display parameters

```
if (UsbProbeDll.usbSetCineBuffers(totalNumFrames) != totalNumFrames)
{
    MessageBox.Show(@"Could not allocate Cine buffers.", "ERROR");
    return;
}
UsbProbeDll.usbClearCineBuffers();

ptCenter.x = winXSize / 2;
ptCenter.y = winYSize / 2;
BmodeDll.bmSetDisplayZoom(1.0f, 0);

if (BmodeDll.bmCalculateDisplay(winXSize, winYSize,
    ptCenter, hProbe, (uint) winXSize, Rotation,
    BmodeDll.bmCD_REFERENCE_LEFT, 0) == BmodeDll.ERROR)
{
    MessageBox.Show(@"CalculateDisplay ERROR: Bad Display Value");
}
```

SDK Examples

■ Image Loop

```
private void ImageThread()
{
    while (true)
    {
        if (UsbProbeDll.usbWaitFrame() == 1)
        {
            FramesReceived++;
            bsptr = UsbProbeDll.usbCurrentCineFrame();

            BmodeDll.bmDrawImage(hImageWindow, bsptr, bm, Interpolate,
                                Average, (System.IntPtr)null, BmodeDll.bmDI_DRAW, 0, true);
        }
    }
}
```

Futures

- Ultra-low-cost probes
 - » There is a development project underway to bring the probe cost down to <\$400.
- Collaborative Research
 - » Invited to Microsoft Research TechFest 2009
 - » Microsoft Research White Paper
 - » Application specific probes
- Open Source USB Probe Software
 - » A group at Washington University in St. Louis has begun a semester-log project to produce an open source imaging suite.
- A website for these developments is at
<http://ultrasound.engineering.wustl.edu>

SDK DEMO

- A complete, simple imaging application is provided in the SDK (with source code).
- Features:
 - » Load/save cine image loops
 - » Load/save images
 - » Detect and configure probes
 - » Live imaging
 - Image contrast and intensity control
 - TGC control