A new way to create electronic devices

Nicolas Villar
Microsoft Research Cambridge, UK
What is .NET Gadgeteer?

• .NET Gadgeteer is a new toolkit for quickly constructing, programming and shaping new small computing devices (gadgets)

• Takes you from concept to working device quickly and easily
Driving principle behind .NET Gadgeteer

- Low threshold
  - Simple gadgets should be very simple to build

- High ceiling
  - It should also be possible to build sophisticated and complex devices
The three key components of .NET Gadgeteer

Modular Hardware
The three key components of .NET Gadgeteer

- Modular Hardware
- Object-Oriented Programming

```csharp
void ProgramStarted()
{
    // Initialize GTM.Modules and ev
    myButton = new GTM.Button(GTM.Bu
    myLed = new GTM.MulticolorLED(GT

    myButton.
    // Do one
    Debug.Pri
}
```
The three key components of .NET Gadgeteer

Modular Hardware

Object-Oriented Programming

Digital Design and Rapid Manufacture
Some background

• We originally built Gadgeteer as a tool for ourselves (in Microsoft Research) to make it easier to prototype new kinds of devices

• We believe the ability to prototype effectively is key to successful innovation
Some background

- Gadgeteer has proven to be of interest to other researchers – but also hobbyists and educators

- With the help of colleagues from all across Microsoft, we are working on getting Gadgeteer out of the lab and into the hands of others
Some background

• Nicolas Villar, James Scott, Steve Hodges
  Microsoft Research Cambridge

• Kerry Hammil
  Microsoft Research Redmond

• Colin Miller
  Developer Division, Microsoft Corporation

• Scarlet Schwiderski-Grosche, Stewart Tansley
  Microsoft Research Connections

• The Garage @ Microsoft
First key component of .NET Gadgeteer

Modular Hardware
Quick example: Building a digital camera
Some existing hardware modules
Mainboard: Core processing unit
User interface modules

- USB Host
- Rotary Encoder
- D-Pad
- Button
- Multicolor LED
- OLED Display
- 480x272 Touch screen (not shown)
Sensor modules

- Ultrasonic Ranger
- Passive IR
- Temp. & Humidity
- Color Sensor
- Accelerometer (not shown)
Multimedia modules

Audio

Camera
Networking modules

- ZigBee
- 802.15.4
- WiFi
- Ethernet
- UART
- WiFi
- Ethernet
Storage modules
Actuator modules

Motor & Servo Control

Relays – low and high voltage (not shown)
Power supply modules

- USB Power
- External DC Power (not shown)
- Rechargeable Battery (not shown)
Extensibility modules
Second key component of .NET Gadgeteer

```csharp
void ProgramStarted()
{
    // Initialize GTM.Modules and event handling
    myButton = new GTM.Button(GTM.Button.Event);  // ButtonPressed
    myLed = new GTM.MulticolorLED(GTM.Led.Event);  // ButtonReleased

    // Do something
    Debug.Print("Hello, World!");
}
```
Software Development Libraries

- Gadgeteer uses the **Microsoft .NET Micro Framework (NETMF)**, which provides a simple and powerful way to write software for small devices.

- Software is developed and debugged in **Visual Studio**, and code is in managed, object-oriented **C#**.

- The **Gadgeteer SDK** provides classes encapsulating functionality for individual hardware modules as well as other utility functions.
.NET Micro Framework

- C# Application and User Libraries
- Class Libraries (Display, Networking, I/O, File System...)
- Runtime Component Layer (Hardware Abstraction + CLR)
- Hardware
Other .NET Micro Framework Devices

FEZ (GHI Electronics)

Netduino (Secret Labs)
Gadgeteer SDK

C# Application and User Libraries

Gadgeteer Library

Modules:
- Audio
- Camera
- WiFi

Interfaces:
- I2C
- PWM
- SPI

Sockets and Pins

Program, Timers, Utilities

Class Libraries

Runtime Component Layer (Hardware Abstraction + CLR)

Hardware
Development support from Visual Studio tools

```csharp
public partial class Program
{

    // Define GTM.Modules here.
    GTM.

    void
    {
        AnalogInput
        Audio
        Breakout
        // Button
        Camera
        DaisyLinkModule
        Deprecated
        DigitalIO
        Display_128x128

    }

}
```

```csharp
void ProgramStarted()
{
    // Initialize GTM.Modules and event handlers here.
    myButton = new GTM.Button(GTM.Button.CompatibleSocket.D);
    myLed = new GTM.MulticolorLED(GTM.MulticolorLED.CompatibleSocket.L);

    myButton.
    // Do one
    Debug.Print
}
```
Third key component of .NET Gadgeteer
Cardboard prototyping
Cardboard prototyping
Digital design and rapid manufacture
Falling cost and increasing availability of 3D printers
Integration of .NET Gadgeteer with 3D Modeling Tools

Reference Case Templates

(“Jigsaw Box” Template)
Integration of .NET Gadgeteer with 3D Modeling Tools

Adding digital models of Gadgeteer modules
Integration of .NET Gadgeteer with 3D Modeling Tools

Automatically generating mounting features
Inte\ration of .NET Gadgeteer with 3D Modeling Tools

Automatically generating cutouts
Making a handheld gaming device in less than 24 hours
Hardware configuration

5 minutes to assemble
public class Piece
{
    public Point[] positions;
    public Point displacement;
    public Color color;

    public Piece(Point[] positions, Point displacement, Color color)
    {
        this.positions = positions;
        this.displacement = displacement;
        this.color = color;
    }

    public void Rotate(bool clockwise)
    {
        for (int i = 0; i < positions.Length; i++)
        {
            Point oldpos = positions[i];
            positions[i].x = clockwise ? -oldpos.y : oldpos.y;
            positions[i].y = clockwise ? oldpos.x : -oldpos.x;
        }
    }

    public Piece Clone()
    {
        Piece clone = new Piece((Point[])positions.Clone(), new Point(displacement.
        return clone;
    }
}
Case design

3 hours to design a custom case
3D printing

6 hours to 3D print case
Final assembly

30 minutes to assemble
A fully functional device in less than 24 hours
Next steps: Getting .NET Gadgeteer out of the lab

- .NET Gadgeteer software, hardware design and design guidelines released as open source project: [http://gadgeteer.codeplex.com/](http://gadgeteer.codeplex.com/)

Next steps: Getting .NET Gadgeteer out of the lab

- Working with a number of hardware manufacturers who will build, distribute and sell the hardware modules
- Initial availability expected end of July
- Started kit priced around $250
- More modules to become available from different manufacturers during the rest of the year
More information

Please get in touch if you are interested in using .NET Gadgeteer for research or teaching

gadgeteer@microsoft.com