Editor: Albrecht Schmidt ■ University of Stuttgart ■ albrecht@computer.org

Ingredients for a New Wave of Ubicomp Products

Thomas Kubitza, Norman Pohl, Tilman Dingler, Stefan Schneegaß, Christian Weichel, and Albrecht Schmidt

Over the last few years, many new embedded computing platforms have emerged, lowering the hurdle for creating ubiquitous computing devices. For instructors, as well as for hobbyists and researchers, the Arduino platform has been widely used for tinkering and prototyping. However, here we highlight some of the newer platforms, communication technologies, sensors, actuators, and development tools, which are opening up new opportunities for ubiquitous computing.

ONE SIZE DOESN'T FIT ALL

To create a smart object or new computing device, the developer must consider processing, communication, storage capabilities, and interaction with the physical world. Nearly all platforms follow a modular approach to support this. Modularity is a key to combat complexity, minimizing the number of required variants while still letting developers customize the devices.

However, modularity doesn't come for free—developers must physically build their custom setup, and components must be connected on a mechanical and electrical level. Yet the custom-made and potentially unique systems should be still easy to program, and appropriate libraries should be available. Furthermore, looking at the variety of requirements—ranging from size and form factor, to energy and processing needs, to robustness and cost constraints—it's apparent that even the most modular platform can't satisfy all requirements. So, it's important to choose the "right" platform and technology.

VARIOUS EMBEDDED COMPUTING PLATFORMS

The following examples highlight how the platform selection can massively ease the development task. The Arduino platform's success has revealed the importance of not just considering the core hardware but also paying attention to opportunities to connect external components (such as sensors and actuators) as well as the development environment and supported tools. The following review isn't comprehensive, but it highlights current trends in platform use.

MSP430 and Launchpad

The MSP430 is Texas Instruments' ultralow power and low-cost platform, where chips are available below US$1, even for small quantities. Creating custom devices is fairly simple, because these 16 bit microcontroller units (MCUs) have an internal clock and require only a resistor as external component.

For prototyping and for hobbyists, Texas Instruments provides the MSP430 Launchpad (www.ti.com/launchpad) as a cheap ($10) but small and powerful evaluation and prototyping board. The platform supports a range of development tools. For nonprogrammers, Modkit (www.modk.it/alpha) aims to provide a graphical programming environment similar to Scratch. The Energia project (https://github.com/energia/Energia) offers an Arduino-style development environment (see Figure 1), while Texas Instruments' Code Composer Studio supports professional developers and complex developments.

mbed

NXP's mbed platform takes a radically different approach from traditional embedded systems development (http://mbed.org). By providing a Web-based IDE and compiler, all code is written in a browser-based IDE for C++ and supported by an open source library. The code is compiled "in the cloud," and the resulting binary file can be downloaded via the browser. For deployment, the code just needs to be written in the mbed boards storage section (which is similar to copying files to a USB stick), and from there the code is written into the MCU's memory for execution. A development board for this ARM-based system (96 MHz ARM Cortex-M3, 32 Kbytes of RAM, 512 Kbytes of Flash) is approximately $50. Furthermore, the browser-based development environment supports social coding. For more details, see the "Cloud Coding for Embedded Systems" sidebar.

Raspberry PI and Beaglebone

The Raspberry Pi platform has probably received the most attention over the last

INNOVATIONS IN UBICOMP PRODUCTS

CLOUD CODING FOR EMBEDDED DEVICES

Traditionally embedded software development was bound to expensive hardware and heavyweight integrated development environments (IDEs), and the assumption was that only highly trained engineers would program embedded platforms. However, a rising interest in "physical computing" has given way to more accessible and cheaper hardware and more lightweight and simpler IDEs.

Cloud-based coding takes this one step further. The IDE is delivered in the browser, and the compiler runs in the cloud, providing true cross-platform development and a simple way to share code and social programming (see Figure A). Cloud-based programming also comes with challenges. In particular, installing the compiled code on the target device is technically more difficult to automate, because the hardware isn't directly connected to the machine running the compiler. For similar reasons, debugging and step-by-step code execution are also difficult.

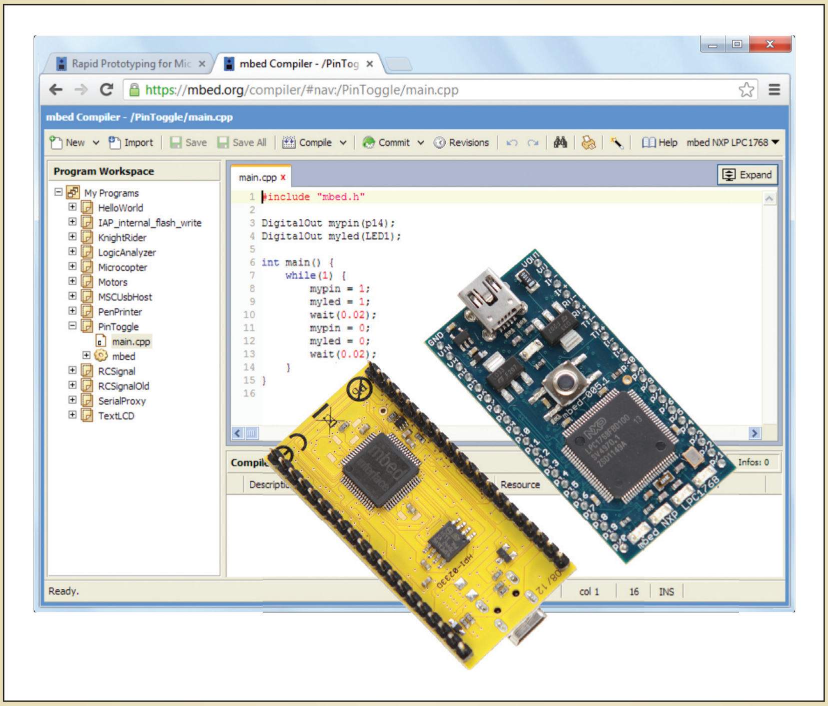

Figure A. The mbeds browser-based integrated development environment for C++ programming (background) and the mbed device. The IDE already integrates a version-control system as well as provides the means for easily searching, importing, and exporting code snippets and libraries.

year (www.raspberrypi.org). Its price (approximately $25) is the most impressive feature of this credit-card sized, fully featured computer, running a 700 MHz ARM processor with 512 Mbytes of RAM. The platform comes with a high-definition multimedia interface (HDMI), 2xUSB, an Ethernet, an S-Video connector, an SD-card reader, and general-purpose input output (GPIO) pins for extension with sensors and actuators. Different extension modules, such as a five-megapixel camera and relays, are available.

Various open source operating systems are available as SD card images, ranging from Raspbian (Debian clone) to Android and XBMC, a fully featured media center. Many popular UNIX programming languages (C, python, nodeJS, and so on) can be used for developing and controlling GPIO pins. Wireless connectivity can be easily added—for example, by plugging in USB Wi-Fi and Bluetooth adapters.

Introduced as "Raspberry Pi with turbo," the BeagleBone (http://beagleboard.org) is another credit-card-sized computer that has been released recently. It comes with a 1 GHz Cortex A8 processor, 512 Mbytes of RAM, 2 Gbytes of onboard flash storage, and most other connectors known from the RaspBerry Pi.

.NET Gadgeteer

A modular but solderless hardware design is at the core of the .NET Gadgeteer (www.netmf.com/gadgeteer). It provides a high-level programming environment with the Microsoft .NET Micro Framework, and software can be written in C# or Visual Basic using high-level APIs.[1] Different modules can be added and removed without soldering, providing a means for interaction with the physical world as well as for communication.

The .NET Gadgeteer aims to help create tangible objects and smart artifacts and supports not only hardware and software development but also the creation of 3D objects that embed computing hardware. Designing enclosures using 3D printing and laser cutting is supported by providing 3D models of all available components as well as plugins for CAD software (such as Solidworks).

CONNECTED TO THE CLOUD

The Internet of Things is quickly becoming a reality, and interactive artifacts are becoming a part of the Web. The following two devices come in the form factor of an SD card but have very different functions.

Electric Imp

Electric Imp (http://electricimp.com) uses the SD card form factor for convenience, because it's easy and cheap to get connectors for SD cards; otherwise, it has little in common with a conventional memory card. Electric Imp is a Wi-Fi enabled development platform that currently costs less than $30. The Wi-Fi connection is configured over the optical channel—a light sensor in the card receives the information that a dedicated smartphone application provides, literally flashing the configuration data using the display as a light source.

Once a Wi-Fi connection is established, the card automatically connects itself to a cloud service pulling the latest code version. The connection between all devices is managed within a Web-based IDE. A simple GUI is used to connect all devices, whereas the device-dependent code is developed using the Squirrel language. Additionally, an HTTP interface is available. The platform provides a high abstraction level and offers a quick way to connect several devices via the Internet using a small form factor. The card offers six programmable I/O pins that can also be configured for Inter Integrated Circuit Communications (I^2C) and the Serial-Peripheral interface (SPI). Breakout boards for a quick start are available (see Figure 2).

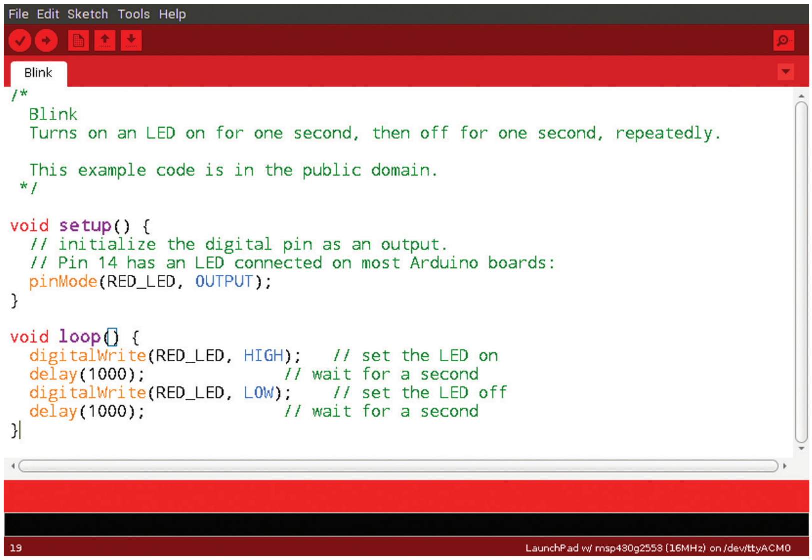

Figure 1. The Energia integrated development environment (IDE) provides a similar development experience to the Arduino IDE. It is based on the Wiring and Arduino framework.

Eye-Fi SD Card

The Eye-FI SD card (www.eye.fi) looks and acts like a conventional SD card but also offers a seamless Wi-Fi connection to the Internet. Originally designed for linking cameras to the cloud, this SD card offers a simple way to upload content without additional programming. The host platform can write to the file system, and the data will be synced with a folder in the cloud. Unfortunately, the firmware only lets you transfer image and video files. However, short data can be easily transferred by writing to a tiny image's Exchangeable Image File Format (EXIF) metadata fields (such as the *Extension Description* field), and long data can be encoded into the image body itself.

The card must be set up to connect to a Wi-Fi access point (for example, in a home Wi-Fi network or through the Wi-Fi access point on a smartphone). Additionally, it must be connected to a folder (such as an FTP account), where the data is uploaded.

Figure 2. Electric Imp's Web-based editor (background), the SD card form factor main board (center), and the Sparkfun breakout board (right). Written Squirrel code is automatically compiled in the cloud and the binaries are deployed seamlessly to the device over the Internet.

INTEGRATING PROCESSING AND COMMUNICATION

Many devices, including smartphones and tablet computers, support Bluetooth. The latest version of the Bluetooth specification (BT 4.0) includes new features that support low-energy operation and fast connections. This will enable many applications, ranging from sensor networks to interactive

INNOVATIONS IN UBICOMP PRODUCTS

objects. Manufacturers, such as Nordic Semiconductor and Texas Instruments, have started to integrate communication modules with processors in a single package, creating a low-cost (less than $5), low-power platform. Examples are the Nordic nRF51822, which combines a Bluetooth LE radio with a Cortex M0 processor, or the Texas Instruments CC2540 system-on-chip (SoC), which combines Bluetooth LE with an 8051 processor.

In addition to the evaluation boards, there are community activities to make these technologies more accessible for prototyping and for hobbyists. One example project is the RFduino (www.kickstarter.com/projects/1608192864/rfduino-iphone-bluetooth-40-arduino-compatible-boa).

SENSING AND ACTUATION

A variety of modules for interacting with the environment and users have become available in recent years. Many of the online shops list more than 100 sensor and actuator modules. There are many different types of connectors and buses, such as the mikroBus (www.mikroe.com/click), where already more than 50 different modules are available. Here we present three examples—a display, an actuator, and a sensor—that show current trends.

Displays

Conventional displays are power hungry. In contrast, E-Ink displays are energy efficient but typically monocolor, and they have a slow refresh rate. Somewhere in between are memory displays. Sharp offers a high-contrast, high-resolution display that can hold pixel states using little power while offering high refresh rates (www.sharpmemorylcd.com). Such displays can thus be used for animations without consuming too much power, offering always-on operation in small devices, such as watches.

Actuators

Besides the common availability of servo and linear motors, shape-memory alloy actuators (SMAs) are becoming increasingly popular. SMAs are smart materials that "remember" their initial shape. When heated up, they will be deformed to a shape that is defined in the production process, so metals can be bent or twisted on actuation. Miga Motor has produced an SMA actuator that provides an output force of 4.5 pounds with a 0.325-inch stroke (www.migamotors.com).

Sensors

Hundreds of sensors are available for a variety of platforms. Conventional sensors—such as pressure, temperature, humidity, barometric, light, color, motion, acceleration, gas, and pulse—are increasingly becoming available as smart sensors (those that bring their own processors). Another trend is to integrate different sensing modalities into a single chip. An example is the InvenSense MPU-9150 (www.invensense.com/mems/gyro/mpu9150.html), which includes a triple-axis accelerometer, gyroscope, and magnetometer and integrates software filters.

Looking at these many new developments, it seems apparent that a new hardware wave is eminent. Over the last few years, many researchers have moved to phones as their ubicomp prototyping platform in combination with interface boards (such as IOIO; https://github.com/ytai/ioio/wiki). Using phones limits the size and type of devices that researchers can build. It will be interesting to see which ideas and prototypes researchers and inventors develop, given the many opportunities with these new computing platforms and technologies.

ACKNOWLEDGMENT

This work is partly funded by the European Project meSch (http://mesch-project.eu; grant agreement no 600851).

REFERENCE

1. S. Hodges et al., "A New Era for Ubicomp Development," *IEEE Pervasive Computing*, vol. 11, no. 1, 2012, pp. 5–9; doi: 10.1109/MPRV.2012.1.

Thomas Kubitza is a PhD student in the human-computer interaction group at the University of Stuttgart. Contact him at thomas.kubitza@vis.uni-stuttgart.de.

Norman Pohl is a PhD student in the joint program of the Media University and the University of Stuttgart. Contact him at norman.pohl@vis.uni-stuttgart.de.

Tilman Dingler is a PhD student in the human-computer interaction group at the University of Stuttgart. Contact him at tilman.dingler@vis.uni-stuttgart.de.

Stefan Schneegaß is a PhD student in the human-computer interaction group at the University of Stuttgart. Contact him at stefan.schneegass@vis.uni-stuttgart.de.

Christian Weichel is a PhD student in the School of Computing and Communications at Lancaster University. Contact him at c.weichel@lancaster.ac.uk.

Albrecht Schmidt is a professor of human-computer interaction at the University of Stuttgart. Contact him at albrecht.schmidt@vis.uni-stuttgart.de.