Program Your Reality with dinam-mite

Dawud Gordon, Martin Alexander Neumann, and Michael Beigl

Karlsruhe Institute of Technology, TecO,
Karlsruhe, Germany, Germany
{firstname.lastname}@kit.edu
http://www.teco.kit.edu/

Abstract. This demonstration will present a self-contained wireless sensor network development environment. The entire toolchain required for development is served by each sensor node, including the IDE, libraries, code, data and visualization. Conference visitors will be able to program applications using the web browser on their own laptops or mobile phones, and see for themselves how much this integration simplifies the process of WSN application development.

Key words: Wireless Sensor Networks, BASIC, Demonstration, Dinam, Prototyping, Programmable Reality

1 The dinam-mite and a Programmable Reality

One of the goals of research in pervasive computing is to enable a proactive environment which adapts to the wishes of the user without explicit commands. Research in context recognition and prediction will hopefully one day allow this "proactive reality" be realized, but until then, truly proactive environments remain unattainable. As a step between the modern state of the art and a proactive reality, we see a "programmable reality" where the user is able to define the behavior of his environment using their own logic, effectively pushing his or her intelligence into the environment.

Currently the largest obstacles remain the amount of development time and complex toolchain necessary in order to create such applications [1],[2]. Furthermore, the technological systems which are required in order to implement these functionalities introduce a level of complexity in terms of hardware, software, libraries, development environments, etc. which is almost impossible for non-experts to overcome.

Here we will present a tool which we have developed to enable a programmable reality. The tool itself is called a *dinam-mite*, which uses the *dinam* [3] concept to completely integrate the IDE, code, logic, data, analysis and visualization into a single sensor node. The self-contained development application allows users to easily implement their own logic and embed this into their own environment using a web browser, even on a mobile phone, without requiring in-depth technical

Pervasive 2011, June 12 - 15, San Francisco, CA, USA

knowledge of the system. The ease of use of the system allows rapid development of applications, even in minutes for beginners [3].

This enables post-hoc creation of logical applications to specify the behavior of otherwise unintelligent objects, creating a tool for rapid application development, including but not limited to prototyping, temporary and even one-time use programs. In this demonstration, we will introduce the newly developed, fully functional *dinam-mite* (see Fig. 1) and allow users to develop applications using the nodes and their own mobile phones.

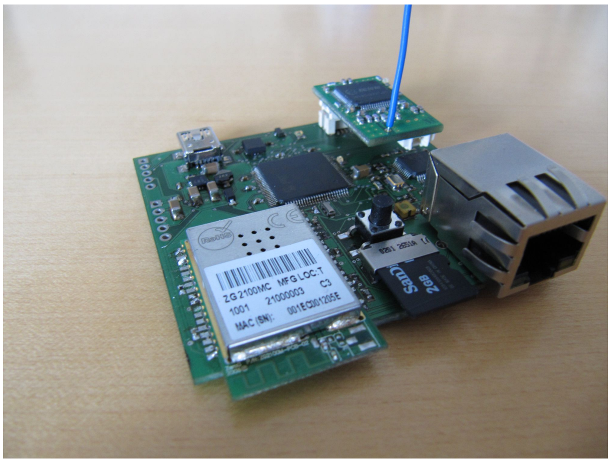


Fig. 1. The *dinam-mite* WiFi Sensor Node

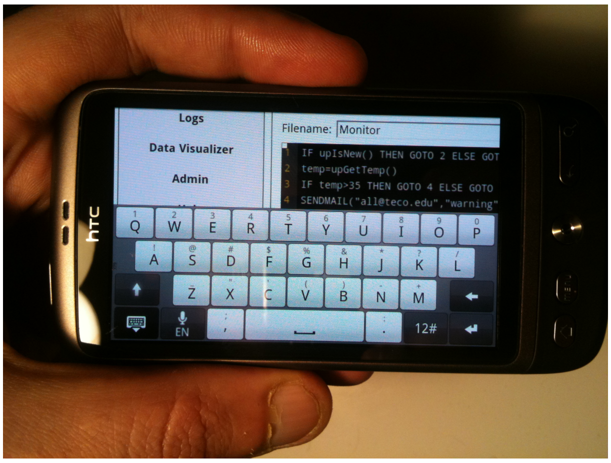
2 The dinam Platform

The *dinam* platform has been designed to run multiple BASIC[5] scripts (programs) and a BASIC console in parallel. The platform concept is implemented by the *dinam-mite sensor node* that hosts a multi-process runtime environment for single-threaded BASIC programs. Conceptually, each BASIC program implements a self-contained logic, as for example the kitchen monitoring application presented in section 3. Based on this, responsibilities can be intuitively separated, as individual tasks are implemented in separate programs and do not have to be intermixed in a single larger program *(separation of concerns)*.

The *dinam-mite* node is based on a single-processor hardware. As a consequence, all programs (scripts and console) share the same processing unit and are executed concurrently in an interleaving fashion instead of true parallelism. The processes are scheduled in a static execution order with preemption. The BASIC run time environment contains a run time library (RTL) which can be used by any running program. The library offers BASIC language-specific operations, e.g. string manipulation or type casting, embedded systems commands such as access to mass storage and LED control, and a set of archetype-based functions for WSN operations [1] such as packet exchange, polling-based access to sensors and TCP/IP-based communication. Based on this functionality, sensor values can be retrieved, processed and logged to permanent storage (SD card) or sent to a remote system via TCP or UDP. A BASIC script may for example process some sensor values and afterwards send an e-mail, post a message on Twitter or store an entry in a database using HTTP PUT.

The *dinam-mite*'s software stack is based on a TCP/IP stack and a FAT32 driver on top of the FreeRTOS[4] operating system. The BASIC run time environment consists of an interpreter, a scheduler and managers for scripts and a console. Clients are provided access to a webserver that hosts a console front-end and an Integrated Development Environment (IDE) front-end. The console front-end offers interactive command line programming and the IDE front-end allows users to program and manage BASIC scripts. The node's hardware consists of an 80 MHz PIC32 micro-controller (MCU) with 128 kB of main memory (RAM). Furthermore, an Ethernet and WiFi PHY and MAC are part of the hardware platform which provide ad-hoc or infrastructure-based connections. Ethernet and WiFi have been chosen, as they are already ubiquitous communication technologies for end-user devices and offer connectivity at nearly zero configuration overhead. End-users can quickly establish a connection to the dinam-mite and immediately start programming their reality, for example using a mobile phone, a tablet or a notebook. The *dinam* concept and further details on the software and hardware implementation are covered in [3].

3 The *dinam-mite* Experience



Filename: Monitor

```
1 IF upIsNew() THEN GOTO 2 ELSE GOT
2 temp=upGetTemp()
3 IF temp>35 THEN GOTO 4 ELSE GOTO
4 SENDMAIL("all@teco.edu","warning"
```

Fig. 2. The *dinam-mite* IDE Accessed from Android

The only tool necessary to define logical applications and behavior on the dinam-mite is a web-enabled device with a browser. Opening the browser and navigating to the NetBIOS name defined on the dinam-mite node opens the welcome page displaying configuration information and navigational links to pages for code development, data visualization, a debug console and log administration. Clicking on the IDE link opens a page with a JavaScript based development environment, where the user may write BASIC code directly. Saving and restarting the node loads the program into memory which then runs in the background. Clicking on the visualizer links allows the user to view system logs, sensor data, and custom log files generated by BASIC logic scripts.

By accessing the IDE served by the dinam-mite on their own mobile phones or laptops (see Fig. 2), conference visitors will be able to create their own intelligent objects. This can be done by attaching the nodes to otherwise inanimate objects, or by wearing the nodes in a pocket, and accessing it over the WiFi network. An example of such post-hoc logic can be found in the code below, which is programmed to remind our lab via email if the oven is left on after normal working hours.

```
1  temp=GetTemp()
2  time=Time()
3  IF time < 080000 THEN GOTO 5 ELSE GOTO 4
4  IF time > 190000 THEN GOTO 5 ELSE GOTO 7
5  IF temp > 35 THEN GOTO 6 ELSE GOTO 7
6 SendMail("alerts@teco.edu", "Warning", "The oven is on!")
7 GOTO 1
```

The authors would like to acknowledge joint funding by the European Commission under the FP7 ICT projects “CHOSeN” (Project No. 224327, FP7-ICT-2007-2) and “TIMBUS” (Project No. 269940, FP7-ICT-2009-6).

References

1. Lan S. Bai, Robert P. Dick, and Peter A. Dinda. Archetype-based design: Sensor network programming for application experts, not just programming experts. In *IPSN ’09: Proceedings of the 2009 International Conference on Information Processing in Sensor Networks*, pages 85–96, Washington, DC, USA, 2009. IEEE Computer Society.
2. David Bannach, Kai Kunze, Jens Weppner, and Paul Lukowicz. Integrated tool chain for recording and handling large, multimodal context recognition data sets. In *Proceedings of the 12th ACM international conference adjunct papers on Ubiquitous computing*, Ubicomp ’10, pages 357–358, New York, NY, USA, 2010. ACM.
3. Dawud Gordon, Michael Beigl, and Martin Alexander Neumann. dinam: A wireless sensor network concept and platform for rapid development. In *Proceedings of the Seventh Internation Conference on Networked Sensing Systems*, pages 57 – 60, Kassel, Germany, 2010. IEEE.
4. Rich Goyete. An analysis and description of the inner working of the freertos kernel. 2007.
5. Thomas E. Kurtz. Basic. *History of programming languages I*, pages 515–537, 1981.