D-Bridge: A Platform for Developing Low-Cost WSN Product Solutions

Dawud Gordon
Distributed and Ubiquitous Computing
Technical University of Braunschweig,
Germany
Email: gordon@ibr.cs.tu-bs.de

Michael Beigl
Distributed and Ubiquitous Computing
Technical University of Braunschweig,
Germany
Email: beigl@ibr.cs.tu-bs.de

***Abstract*—Today the construction of wireless sensing systems requires the use of rather high-priced pieces of hardware and the assembly of these hardware parts (wireless transmission core, sensor module, gateway device), as well as the implementation of the according software (sensor mote software, gateway software, end user software). This makes overall development and the thereby resulting product costly in terms of both time and money. With the D-Bridge concept that we present in this paper we will show an approach that is much cheaper than any other solution in terms of hardware and development costs. This D-Bridge is a combination of a gateway between the wireless sensor network and the IP network which also includes a Web-based application server at the wireless sensor network appliance's location.**

I. Introduction and Related Work

Existing wireless sensor network solutions often target complex problems using wireless sensor networks. This is done by adapting powerful systems of existing wireless sensor devices and infrastructures which provide open programming interfaces. Examples of such device families are the Motes [5] and BTnode platforms [4].

Developers of wireless networked sensing system applications have to develop multiple system components: The application at the wireless sensor node, the gateway application to connect the wireless sensor network and the network containing the device running the end-user application, and that end-user application itself. The basic components e.g. the sensor node hardware, gateway hardware, the sensor node software etc., are offered as independent, open modules by hardware vendors. This allows the construction of very powerful applications which are best suited for a complex application task. One example producer is Crossbow (www.xbow.com), that offers 6 types of different wireless modules, 5 different types of gateway solutions and 9 types of sensor modules.

The approach for wireless networked sensing system applications that we want to follow is going in a different direction: our goal is to minimize development and installation time to enable the rapid implementation and installation of small wireless sensing system appliances. To our knowledge, there is currently no other system available in literature or on the market that follows a similar approach. In order to accomplish the minimization of development and installation effort we will implement the following steps:

First, we will employ a wireless sensor node that can be re-configured but not re-programmed. This avoids one development step and also reduces overall complexity, thus increasing reliability in practice. To this end we use our tiny uPart [2] wireless sensor node. This sensor node periodically reads its sensors and send the results to a gateway. Period and other parameters can be adjusted by a configuration interface.

Second, we avoid using a client application that requires installation at an end-user computer. Instead, we use a Web-based application. This approach also lowers complexity because in this way we ensure that our application is independent of the operating system on the client computer.

Third, we focus all application functionality on one central device, the D-Bridge. This devices contains the gateway between sensor network and IP based networks, but also the application logic and presentation.

Fourth, we support the development of applications by providing an interface to the wireless sensor network that allows the creation of a small networked sensing systems application within a few minutes.

In the next sections we will explain this process in detail, as well as give an example of the development of an application in order to show the simplicity of the process.

II. System Components

The uPart is an extremely low power sensor node which operates by taking sensory readings at set intervals and transmitting this information via a RF communication module. The nodes are quite efficient, being able to survive many months on a single button-cell battery. They are outfitted with light, temperature and vibration sensors and are small and very low cost. They are not capable of receiving data via RF communication, which improves their energy consumption, but requires an additional heterogeneous network member acting as a bridge to form a fully functional network. The bridge allows the user to interface with the wireless sensor network and analyze the resulting data output of the sensor nodes, usually in conjunction with an application which presents the data in a meaningful manner. The D-Bridge is such a device which creates a connection between a uPart wireless sensor network and an Ethernet network.

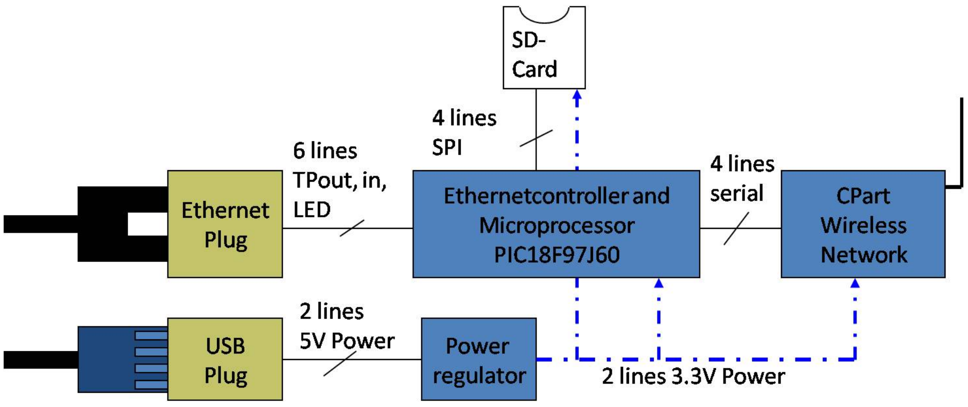

Fig. 1. A Schematic of the D-Bridge

The D-Bridge implements a micro embedded web server, making a website available to all clients in the network. The bridge consists of a PIC18F97J60 processor from Microchip controlling an Ethernet jack as well as an interface with a Cpart for communication with the uPart network. The processor also supports a SD card slot for additional data storage. The device is powered by a USB adapter which delivers 5V of power, converted to 3.3V by a regulator as shown in figure 1.

The firmware is a combination of a modified version of the TCPStack software from Microchip which implements a TCP/IP stack as well as a basic web server and DHCP client. This enables the D-Bridge to host a website on the network and to automatically configure its IP address from a DHCP server. A FAT32 file system is implemented on the SD card using code purchased from www.embedded-code.com, which allows the system to access the SD card using functions similar to those used for file stream operations in C. Communication with the wireless sensor network occurs over a Cpart programmed with the firmware for the USBBridge.

III. Framework

The goal of the D-Bridge platform is to reduce development time for simple and cost-effective applications. The user interface is embedded within the D-Bridge itself in the form of a website and is accessible via HTTP, through which the user can interact with the wireless sensor network. The system is constructed such that the uPart nodes in the sensor network periodically communicate with the D-Bridge over RF using a simple protocol [1]. The bridge reacts to each packet from the uPart nodes in a way specified by the the application, and also facilitates communication with the user over the network as shown in figure 2. Communication on the network side between the D-Bridge and the client can be configured to occur directly ad-hoc, over the local area network, or via the WWW, depending on what the current application calls for.

The HTTP protocol is specifically appropriate for the user bridge communication because it is well tested and almost universally implemented. It is also machine independent which eliminates portability issues, and its intrinsic request and response architecture allows systems on both sides of the network bridge to function with minimal interdependence. As a further plus, one can assume that all clients using this product have had previous experience with the protocol through interaction with the WWW.

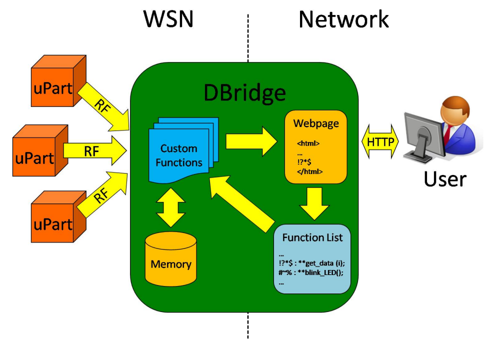

Fig. 2. System Architecture

The D-Bridge provides a platform to create an application based on HTML and provide access to this application to users on the network. The website is composed of static HTML with a proprietary method of creating dynamic content. Each .html file of the website is preprocessed by the web server at request-time which searches for specific identifier strings. Each string is linked to a function in the program memory of the processor and, when a string is found, the function specified by that string is executed. These functions can return data which replaces the special strings embedded within the code, or it can be used to execute a command and carry out a specific operation. The resulting bridge offers the developer a platform with which to quickly and easily create applications. The proprietary method was selected over the implementation of a preexisting web development language in order to maintain memory and processor usage within the limits of the PIC18F97J60 because of its efficient nature and low cost.

The website is stored on an external SD card which can also be used as data storage for items such as system or packet logs, configurations or context information. All sensory data packets

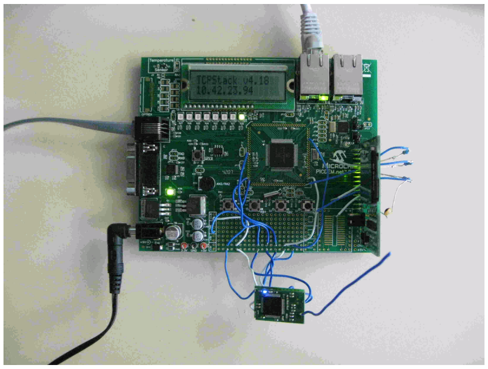

Fig. 3. The D-Bridge Prototype

from the sensor network are transfered to the application and can be retransmitted on the network via TCP/UDP, logged on the SD card, analyzed, discarded, or handled by any number of custom operations which can be created with minimal effort. A prototype of the D-Bridge has already been integrated into a uPart network and can be seen in figure 3.

In order to demonstrate the simplicity of product creation using this platform, the steps required to build a product will be described. This example describes an application to display the temperature history of n different rooms, each monitored by a uPart. The components of the application and their interaction can be seen in figure 2. The first step is to create a function to log the temperature values from all packets received from the uParts. This is accomplished by creating a file on the SD card using the functions provided by the FAT32 system, and writing the data to that file, sorted by sensor node and chronological order. The next step is to write a function `get_data()` in C which takes one parameter, i, extracts the last 10 readings for the i^{th} node and returns these in a comma delimited string. An identifier for this function is then entered into the table of dynamic function identifiers with the information that the function takes one parameter: i. This table creates a dynamic link between the character string in the HTML text and the C function in program memory.

The application itself consists of a HTML web page containing a table with one column and n rows, where every row contains the results of the dynamic function, or the last 10 temperature sensor readings for the i^{th} node. This is accomplished by replacing the text in each cell with the identifier for the C function as it appears in the identifier table, with the parameter i, for i = 1 to n. The result is a dynamic link between the website and the data stored on the SD card. When the user requests the web page over HTTP, the system searches for any identifiers embedded in the HTML and upon finding one, a pointer to that function is retrieved from the function list. The function `get_data()` is executed for each node, accessing the SD card each time, and returns the resulting history string to the website. The new website containing the dynamically generated content is then uploaded to the user over HTTP, allowing him or her to view the content

TABLE I
D-BRIDGE DATA TRANSFER RATES

Type	TCP kB/s	UDP kB/s
Output	30.2	140
Throughput	15	n/a

IV. Outcome

Embedding the application within the bridge has several positive effects on the system as a whole. The fact that the application is no longer dependent on the user's hardware, operating system type and version reduces time spent making applications portable. A product developed on this platform can be delivered with the application preinstalled, reducing installation effort for the user. Both of these factors reduce the total cost of ownership of the system, improving attractiveness for the user.

The current design of the D-Bridge consists of only a few low cost components: the Ethernet plug (from Pulse Inc., includes the operation LEDs and the transformer, 2€), the processor (includes CPU, Ethernet controller, dual-ported buffer RAM, periphery esp. SPI, 3,60€), the USB plug (0,20€), SD-Card slot (0,65€), oscillator (0,46€), voltage regulator (0,20€) and about 10 resistors and capacitors (1€). This leads to a component cost of less than 10 with PCB, and a possible sales price after production of about 20-25€ (assumed production quantity: 50 pieces, without RF part). Together with the price of about 20€ per sensor node, a framework kit can start as low as 60€ for the hardware. The low cost of this platform allows developers to create solutions for problems which previously could not justify the expensive development of a wireless sensor network solution.

Analysis of the data transfer rates of the D-Bridge show that the system has the power necessary for small to medium applications. Table I shows the measured transfer rates, where output is the rate at which the system can transfer data to the network from RAM, and throughput is the rate at which the system can upload data from a file located on the SD card to the network. The latter, which should also represent throughput from the wireless sensor network, was tested using HTTP GET which explains why there is no result for UDP. The rates show that the system has a high enough throughput to support a moderately complex website.

V. Conclusion

The on-board application development environment reduces time and effort required for product creation, as well as installation. The affordability of the uParts and of the D-Bridge create a sensor network which is elegantly simple and inexpensive. The affordable nature of the hardware, as well as the effort saved by the embedded application provides developers with a platform to create extremely affordable WSN solutions to solve problems of low to moderate complexity. These products will then be able to fill a niche previously left empty by the preventatively expensive development costs of wireless sensor network solutions.

VI. Future work

The current D-Bridge is still a prototype, and not in a final design stage. We are currently working on a small integrated solution of about 30x40mm.

References

[1] M. Beigl, A. Krohn, T. Zimmer, C. Decker, and P. Robinson, *AwareCon: Situation Aware Context Communication*, 5th International Conference on Ubiquitous Computing (Ubicomp) 2003

[2] M. Beigl, A. Krohn, T. Riedel, T. Zimmer, C. Decker, M. Isomura *The uPart experience: Building a wireless sensor network*, IEEE/ACM Conference on Information Processing in Sensor Networks (IPSN), 2006

[3] C. Decker, M. Berchtold, L. W. F. Chaves, M. Beigl, D. Roehr, T. Riedel, M. Beuster, T. Herzog, D. Herzig *Cost-Benefit Model for Smart Items in the Supply Chain* Conference on The Internet of Things, Zurich, March 2008

[4] J. Beutel, O. Kasten, F. Mattern, K. Rmer, F. Siegemund, L. Thiele *Protoyping Wireless Sensor Network Applications with BTnodes* 1st European Workshop on Wireless Sensor Networks, January 2004

[5] J. Polastre, R. Szewcyk, A. Mainwaring, D. Culler, J. Anderson *Analysis of Wireless Sensor Networks for Habitat Monitoring* ACM International Workshop on Wireless Sensor Networks and Applications, September 2002