Przemysław ZIELONY, Krzysztof PIOTROWSKI
IHP – Leibniz-Institut für innovative Mikroelektronik, Im Technologiepark 25
15236 Frankfurt (Oder), Germany

MODULAR DEVELOPMENT OF DATA-CENTRIC APPLICATIONS FOR EDGE DEVICES

With the dynamic development of the sensor network technology, there is a growing need for effective software design that is scalable and flexible. This work presents a modular structure based on tinyDSM, enabling the connection of various adapters and services. This approach streamlines the development process by enabling developers to leverage previously implemented modules, thereby increasing the speed and efficiency of subsequent WSN implementations. The adapters ensure communication between tinyDSM and microcontroller peripherals or external hardware blocks, such as sensors, and facilitate the integration of new modules. The emphasis on edge computing enables data processing at the node level, reducing network traffic and increasing resource efficiency.

Modułowy rozwój aplikacji zorientowanych na dane dla urządzeń brzegowych

Wraz z dynamicznym rozwojem technologii sieci sensorowych, rośnie potrzeba metod efektywnego projektowania oprogramowania, które będą zarówno skalowalne, jak i elastyczne. W niniejszej pracy zaprezentowano konstrukcję modułową opartą o tinyDSM, umożliwiającą dołączanie różnych adapterów i serwisów. Takie podejście usprawnia proces programowania, umożliwiając programistom wykorzystanie wcześniej zaimplementowanych modułów, zwiększając w ten sposób szybkość i efektywność kolejnych wdrożeń WSN. Adaptery zapewniają komunikację pomiędzy tinyDSM a peryferiami mikrokontrolera czy zewnętrznymi blokami sprzetowymi, takimi jak czujniki, a także ułatwiają integrację nowych modułów. Nacisk na przetwarzanie brzegowe umożliwia przetwarzanie danych na poziomie węzła, redukując ruch w sieci i zwiększając efektywność wykorzystania zasobów.

1. INTRODUCTION

With the dynamic evolution of the Internet of Things (IoT) and the growing need for efficient data management, focusing on creating data-centric applications in sensor networks becomes highly significant. Faced with resource constraints such as computational power, memory, and energy, due to the characteristics of edge devices, a modular approach to application development becomes indispensable. Modularity allows for constructing applications by combining smaller, self-contained modules, which not only facilitates the development process, but also ensures flexibility and scalability of the target system.

In the context of wireless sensor networks (WSNs), where the data is collected, processed, and transmitted in real-time, the challenge lies in ensuring efficient data management while optimizing system performance. Modular development of data applications enables the design of systems that are more responsive to changing operating conditions. Additionally, in the proposed approach, a methodology for effective design and implementation of data applications for edge devices will be presented.

2. RELATED WORK

Middleware for wireless sensor networks must be designed as lightweight software to help developers implement new applications. However, due to the diverse range of microcontroller units (MCUs) with different capabilities and limitations, it is challenging to achieve universal adaptation across all MCUs.

There is no one-size-fits-all middleware solution for Wireless Sensor Networks. Typically, developers must prioritize specific aspects depending on their application requirements. These aspects may include high-frequency data sampling, energy efficiency, node mobility, time synchronization, data replication, node provisioning, node localization, power management policies, and Quality of

Service considerations [1]. Different middleware solutions offer varying levels and approaches to QoS, and different applications may require different QoS levels.

Various approaches have been adopted in designing middleware for WSN, each offering distinct capabilities and exhibiting its own set of limitations.

3. TINYDSM

The tinyDSM [2] is a data-oriented and modular middleware for wireless sensor network devices and embedded devices. It allows to define local variables that can store user application's data in a structured way and provides the so called Data Interface API for software components to store and access data in the middleware. The shared data area is implemented using tuple spaces with data shared as distinct variables and in its initial approach tinyDSM is supporting communication based on localized mesh topology for intensive data replication. By that, it provides a distributed data storage and allows to define the data replication area with a configurable level of reliability. Its core is implemented in a generic way employing an adaptation layer allowing it to be ported on almost any MCU software platform. The instance of tinyDSM is configured with the list of variables to be used in the given application and defines a tailor-cut and optimum implementation.

The application implemented based on tinyDSM consists of simple modules (called services and adapters) with functionality limited to data accessing and processing, what allows better testing and code reuse. Adapters translate other interfaces to the Data Interface, allowing exchanging data with external systems or hardware blocks. With these functional blocks it is, for instance, possible to implement software drivers supporting sensors or actuators – adapters control the specific external interfaces and import or export the specific data (variables). On the other hand, the services focus solely on processing the data available in the tinyDSM middleware and generate new data.

4. THE MODULAR APPLICATION DEVELOPMENT APPROACH

Our approach for application development is thus based on the configurable data storage and exchange middleware tinyDSM at the bottom, with plug and play data processing modules (services) and data transformation modules (adapters) on top of the middleware and configurable networking protocol managing the data exchange between sensor nodes and gateway nodes (see Fig. 1).

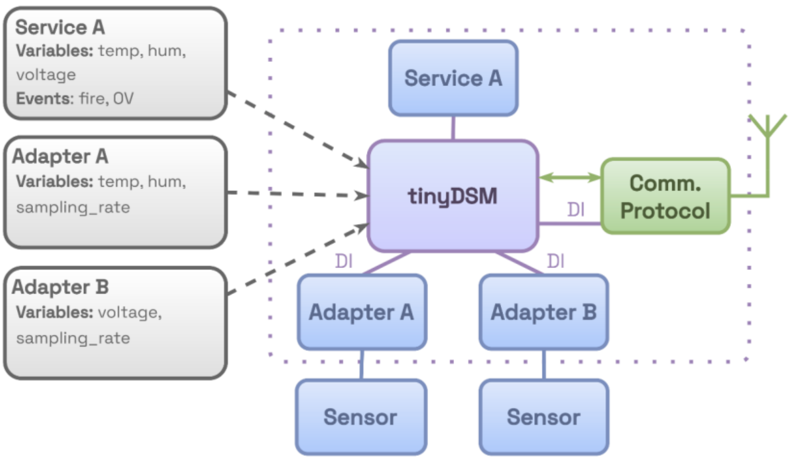

Fig. 1. The tinyDSM WSN node application concept
Rys. 1. Koncept aplikacji na węźle WSN na basie tinyDSM

The central point of the proposed architecture is the tinyDSM middleware that supports the modular approach allowing for the attachment of various adapters and services. This approach brings significant advantages, especially in the context of successive deployments of wireless sensor network applications, where the speed of creating programs using the same or similar adapters becomes a crucial factor. Instead of spending time rewriting code, developers can leverage pre-written and tested modules. Adapting software to WSN hardware would rely on configuring ready-made software blocks or making little changes to the program code.

Initially, such software blocks in the form of adapters or services would have to be implemented from scratch. However, over time, with successive applications of this solution, more and more modules will become available.

The core tinyDSM module is customized with its application-specific configuration and the adaptation layer allows it to be deployed on most microcontrollers available on the market. The adaptation layer also defines the settings related to the initial processor clock parameters and available memory. This makes the solution universal for designing WSNs with various applications and adaptable to the hardware constraints.

Data driven applications are defined by data flows between data sources and data sinks. These can be represented by different physical components, like sensors or actuators. Handling external functionality and peripherals on any microcontroller can be realized using adapters. Within the proposed architecture, these relations are solely considered as data flows and specialized adapter modules can be created or applied to shape the endpoints related to the path of the data flows in the applications. All the different data sources can be read out using their specific drivers and the data representing the measurements is then stored in the measurement variables within the tinyDSM middleware (see Fig. 2)

Adapters are a crucial element, acting as an intermediary layer in communication between tinyDSM and peripherals or sensors. In most applications, each sensor is typically managed by a microcontroller using a driver provided by the device manufacturer or developed by WSN developers. Due to differences in the operation and the extensive functionality of sensor drivers, they cannot be universally integrated with microcontroller operation. Adapters enable the customization of these drivers to work with tinyDSM middleware, ensuring interoperability and flexibility in handling various devices, by making the interaction data-centric.

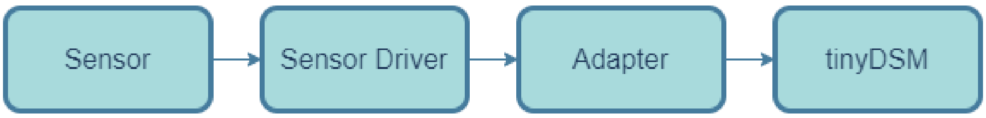

Fig. 2. Sensordata flow on a device based on tinyDSM
Rys. 2. Przepływ danych z sensora na urządzeniu opartym o tinyDSM

As the measurement data becomes available in the middleware it can be further processed. This task is done by the services, i.e., they read the data from the middleware and store the results of their data processing back. Combining the adapters and services it is possible to shape the complete data path required by the specific application.

The main functionality of the services and adapters is defined at compile time, but it can be also influenced by, so called, configuration variables that offer the possibility of modifying the parameters of the functional modules during program execution. An example here is for instance the variable controlling the sample rate of the given sensor. This approach further extends the application development flexibility and supports the adaptation to changing conditions.

Additionally, the approach proposes the virtual sensors, implemented by services capable of aggregating data from a number of physical sensors, and combining them into a single value. For instance, integrating data from sensors measuring temperature, air humidity, air pressure, and PM particles, it becomes feasible to compute the virtual Indoor Air Quality measurement. This approach offers a versatile and efficient means of monitoring diversity of parameters, providing valuable insights for applications including building automation, health management, and environmental monitoring.

Thus, the set of measurement and configuration data related to the application defines the set of variables required to be handled by the middleware and defines its configuration and final instantiation.

In the context of communication protocols in wireless sensor networks, the tinyDSM approach facilitates the integration with a diversity of wireless network protocols through appropriate API. This makes expanding network functionality by adding new protocols much simpler and more efficient. Within the realm of WSN, there exists a diverse array of nodes or edge devices, each equipped with its own radio module(s). These modules are designed to support a variety of communication protocols, depending on the specific requirements of different applications and environments. This flexible and modular approach allows to apply the most suitable platform while maintaining data interoperability.

Support for additional radio modules on network nodes offers the possibility of using them to monitor nodes or remotely update software, increasing flexibility and efficiency of network management and maintenance. All software modules, like adapters and services for the tinyDSM middleware make use of variables. The standard configuration of the device is done at compile time, but it is also possible to change the configuration at run time using a service that provides connection via the Bluetooth low-energy (BLE) module, allowing a direct connection to a mobile device such as a smartphone. Such a connection facilitates on-site node configuration, reading data from sensors and remote software updates. This solution would also be helpful for identifying errors and debugging within the deployment.

CONCLUSIONS

Our approach focuses on a configurable data storage and exchange middleware – tinyDSM – extended by plug-and-play data processing and transformation components (adapters and services). This architecture offers significant advantages, particularly in Wireless Sensor Network (WSN) deployments, where rapid program creation and adaptation are critical. By utilizing pre-written and tested modules, developers can expedite application development without the need for extensive code rewriting. The modularity of the architecture allows for easy customization and adaptation to diverse hardware and application requirements, enhancing universality in WSN design.

In today's WSN applications, optimal use of network resources is crucial for efficient data handling. Traditionally, in many middleware applications, raw data from sensors is sent to central devices, where it is processed at higher layers. However, new approaches are emphasizing data processing already at the edge devices, significantly reducing the amount of data sent over the network. For these new approaches, processing services can be implemented at the lowest layer – at the edge nodes. These services are capable of analyzing sensor data, detecting anomalies, discarding incorrect data and sending only relevant information. In this way, the amount of data sent over the network is significantly reduced, which contributes to the efficient use of network resources, while implementing the desired data flows.

On top of the proposed approach many high-level functionalities can be implemented. For instance, to manage energy on WSN nodes we propose a special energy management service that implements a distributed monitoring of energy consumption and allows adjusting the system operation depending on energy availability.

REFERENCES

1. M.M. Wang, J.N. Cao, J. Li, K.D. Sajal, "Middleware for Wireless Sensor Networks: A Survey." J. Comput. Sci. Technol. 2008

2. K. Piotrowski, P. Langendoerfer, and S. Peter, "tinyDSM: A highly reliable cooperative data storage for Wireless Sensor Networks," in Collaborative Technologies and Systems, International Symposium on, 2009, pp. 225-232, doi: 10.1109/CTS.2009.5067485