# Real-time Water Level Monitoring Using Low-Power Wireless Sensor Network

Nguyen Minh Phuong, Manuel Schappacher
Prof. Dr. –Ing. Axel Sikora
Laboratory Embedded Systems and Communication
Elektronics, University of Applied Science Offenburg
Badstrasse 24, 77652 Offenburg, Germany
Email: nphuong@stud.hs-offenburg.de
{manuel.schappacher, axel.sikora}@hs-offenburg.de

Zahoor Ahmah, Prof. Dr. Abubakr Muhammad
Laboratory for Cyber Physical Networks and Systems
School of Science and Engineering,
Lahore University of Management Sciences
Opposite Sector U, DHA, Lahore 54791, Pakistan
Email: {zahoor.ahmad, abubakr}@lums.edu.pk

***Abstract*—Environmental Monitoring is an attractive application field for Wireless Sensor Network (WSN). Water Level Monitoring helps to increase the efficiency of water distribution and management. In Pakistan, the world's largest irrigation system covers 90.000 km of channels which needs to be monitored and managed on different levels. Especially the sensor systems for the small distribution channels need to be low energy and low cost. The distribution presents a technical solution for a communication system which is developed in a research project being co-funded by German Academic Exchange Service (DAAD). The communication module is based on IEEE-802.15.4 transceivers which are enhanced through Wake-On-Radio (WOR) to combine low-energy and real-time behavior. On higher layers, IPv6 (6LoWPAN) and corresponding routing protocols like Routing Protocol for Low power and Lossy Networks (RPL) can extend range of the network. The data are stored in a database and can be viewed online via a web interface. Of course, also automatic data analysis can be performed**

***Keywords— Water Level Monitoring, Wireless Sensor Network, 6LoWPAN, IEEE-802.15.4, Wake-on-Radio***

## I. INTRODUCTION

Pakistan is an agricultural country with irrigated agriculture occupying 74 percent of total cultivated area of 22 mio hectares. Nearly 97 percent of available surface water is diverted for irrigated agriculture, contributing to more than 90 percent of agricultural production [1].

The irrigated agriculture of Pakistan mainly depends on Indus River System and its tributaries. The need for improvement in efficiency and productivity of the irrigation water has become one of key issues for the irrigation as well as agriculture sector. Various studies have showed that the state owned irrigation systems had not been performing well and deteriorating day by day, especially in developing countries due to financial, managerial, and socio-political factors [11]. Therefore there is a great need for efficient management of surface water as a mean to enhance food security, reduce poverty, and adapt to uncertainties owing to climate changes.

Wireless sensor network is an appropriate choice for such long-duration and large-scale environmental monitoring system, providing sensor measurements at high temporal and spatial resolution. The project aims at understanding of water parameters of the irrigation system, such as level, temperature, and velocity, in response to challenges faced by Pakistan due to its urgent demand of efficient of equal water distribution. Based on sensor design provided from other projects at Lahore University of Management Sciences (LUMS, which has made several field deployments in Punjab province), an extension is developed and tested in a joint effort from the authors' teams. This collaborative research and development project is co-funded by German Academic Exchange Service (DAAD) since 2013.

This work presents an IEEE-802.15.4 based ultralow power wireless sensor network solution supported by Wake-On-Radio (WOR) technology to the water metering system in Pakistan.

The contribution is structured as follows. The overall measurement solution is shortly presented in ch. II whereas the current GSM/GPRS-based solution is discussed with it characteristics in ch. III. Based on this analysis, some design possibilities are considered in ch. IV. Subsequently the hardware design is introduced in ch. V, followed by the description of the software architecture in ch. VI. Experimental measurements are detailed in ch. VII.

## II. MEASUREMENT SOLUTION

A low-power water discharge measurement has been developed in [2] [8] by LUMS which is suitable for wide-scale deployments in irrigation canal networks of the Indus river basin. Sensing of water levels is achieved by ultrasonic technology [8] and reported it to be an effective solution for sensing water levels at distributary and branch canals (flows in the range of 10-1000 cusecs). Canal flows are computed from water levels using a hydraulic formula (the so-called rating curve) which is obtained from an infrequent calibration process at the point of measurement. The solution in [2] [8] employs IP67-standard weather resistant Maxbotix MB7380 [9] and uses GPRS for communication. 17 such units have been deployed in the Bahawalnagar area in Hakra branch canal command and data has been recorded for over a year, see Fig 1). Backend database and website clients have also been developed for dissemination of data to farmers and irrigation engineers. Around 6,000 critical sites have been marked by Punjab Irrigation Department for such instrumentation.

Fig. 1. Photographs of smart water meter installations in Bahawalnagar on distributary canals. The meter is mounted inside a concrete stilling well [2][8]

LUMS team is also developing an ultrasonic level sensor based measurement solution for small watercourses, i.e., outlets from distributary canals (< 5 cusecs) from distributary canals. The measurement hydraulics for such small flows is very different from relatively large flows of distributaries. In a test deployment on LUMS campus [10] (See Fig. 2), a long-throat steel flume has been inserted in an open channel.

Fig. 2. Smart water meter installation at LUMS campus on a steel flume before being inserted in a drainage channel [10]

Water level is simultaneously measured by ultrasonic ranging (in mm) at two points along the flume's length. The levels are wirelessly transmitted to a computer server where water levels are converted into instantaneous flow (in cusecs) using a hydraulic equation. Measuring such small flows at remote sites is extremely challenging due to rapid sedimentation, calibration difficulties, unreliable power sources and remote communication. To understand the scale of such instrumentation, each of the 6,000+ sites marked for instrumentation at the distributary level can be further instrumented to measure the 10-100 outlets per channel. A similar work has been reported in [12].

## III. LEGACY M2M SOLUTION

The currently installed solution is based on GSM/GPRS services. Sensor devices wake up every 30 minutes, sensing for 2 seconds before measurements are relayed to the database. Due to the long sleeping periods, this solution is quite power-efficient. According to [2] the system is expected to last for 691 days while running on a pair of Lithium Thionyl Chloride batteries 3.6V, 14000mAH. However, for really large-scale scenarios with flexible topologies requirement and in-network processing operations such as aggregation and actuation, this approach comes with the following caveats.

GSM/GPRS-based design only supports at locations with GSM/GPRS coverage, and therefore does not support ad hoc network.GSM/GPRS-based design only supports waking up a node passively at particular time interval, and therefore downlink and multicast communication cannot be supported.

The current installed solution is available online at [3].

## IV. DESIGN CONSIDERATIONS

### A. Requirements

The following requirements need to be taken into account:

- A large scale installation with several thousands of measurement nodes shall be potentially possible.
- Hierarchical network design shall be possible.
- Standardized communication shall be used.
- The frequency band shall be selectable; however, in the first generation the 43470° MHz frequency shall be used.
- Real-time communication shall be supported both uplink and downlink. Real-time communication shall mean reaction times in the range of 1 to 5 seconds.
- A single hop communication shall have a range of ~1 km.
- Multi-hop (routed) communication shall be supported.
- The average energy consumption of a communication node shall be in the range of 500 mAh p.a.
- Flexible stack with regard to different transceiver compatibility.
- The cost shall be as low as possible. For prototyping, standard Commercial Off-the-Shelf (COTS) modules shall be available.

### B. Technology Selection

Recent advances in wireless communication had led to a multitude of wireless sensor network technologies, such as Bluetooth Low Energy (BLE), EnOcean Radio Protocol (ERP), ZigBee, Z-Wave, 6LoWPAN, and many proprietary solutions. During a technology scouting phase amongst the standardized solutions, 6LoWPAN has been selected the candidate with the optimum parameters meeting the requirements.

## *C. Wireless sensor network architecture*

Due to the large spread of the network and the range of a single hop communication of 1km most of the devices must be able to route messages. This becomes possible thanks to the 6LoWPAN technology, which is based on IEEE-802.15.4 standard, as the network layer. However, in order to meet the low-power requirements, the lower layer needs to be extended by Wake-On-Radio technology.

The standard IEEE-802.15.4 defines two types of devices: a full function device (FFD) and a reduced function device (RFD). An FFD can directly communicate with other FFDs and RFDs, and operate in three modes acting either as personal area network (PAN) coordinator, coordinator, or a device. Meanwhile, an RFD, in contrast, can talk only to another FFD, and therefore is suitable for extremely simple applications.

The most important difference between FFDs and RFDs in this scenario is reachability. FFDs must be ready to receive data all the time whereas RFDs have idle moments and can poll data from their associated coordinators once they are in active states. Therefore routers must be configured as FFD.

From the application point of view, information of water level, temperature, and velocity from static nodes will be relayed to a base station before carried over the Internet to remote server, as shown in Fig. 3.

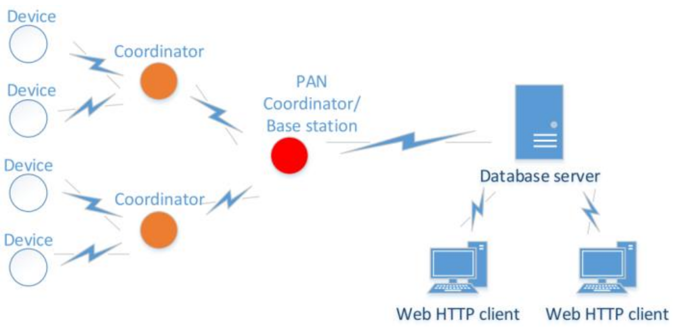


Fig. 3. Water metering system architecture

## *D. Hardware Setup*

The project's experimental systems are based on Texas Instrument's MSP430 microcontroller communicating with a radio frequency board CC1120 over SPI bus.

The MSP430 family is featured with extensive low-power modes, a powerful 16-bit RISC CPU, 16-bit registers, three 16-bit timers, high performance 12-bit analog-to-digital converter. The digitally controlled oscillator allows the device to wake up from low-power modes to active mode in 3.5µs [4].

The CC1120 device is a fully-integrated single-chip radio transceiver designed for high performance at very low power and low voltage operation in cost-effective wireless system. The device totally supports unlicensed Industrial, Scientific, and Medical (ISM) frequency band with operation frequency ranges of 164-192° MHz, 274-320° MHz, 410-480° MHz, and 820-960°MHz. The CC1120 comes with low current consumption during operation, and adds low power modes [5]:

- RX: 2mA in RX sniff mode
- RX: 17mA peak current in low-power mode
- RX: 22mA peak current in high performance mode
- TX: 45mA at +14dBm.

Beside extensive hardware support provision for packet handling, data buffering, burst transmission, link quality indication, and clear channel assessment, the CC1120 also supports Wake-On-Radio technology that help source node to wake up destination node using preamble signal before data transmission. This mechanism enables the radio to periodically wake up from sleep mode, listening for incoming packets with minimal CPU interaction. If a valid packet is not received during receiving mode, the radio immediately returns to sleep, helping to conserve power.

## *E. Low Power Extensions*

First stage to reduce power consumption within the project is usage of Wake-On-Radio functionality, so called RX Sniff Mode. This novel hardware-based feature, which is enabled by the TI Performance Line WaveMatch technology [6], allows a fully transparent data communication by autonomously duty cycling the receiver while waiting for a packet. Therefore WoR-powered devices are able to receive packets all the time.

Enhanced Wake-On-Radio (eWOR), which uses a flexible integrated sleep timer, enables automatic receiver polling with no intervention from the microcontroller [5]. When the radio frequency transceiver enters receiving (RX) mode, it listens for preamble signal. If a valid preamble is not received it then immediately goes to sleep mode as illustrated in Fig 2. The sleep timeout and duty cycle can be configured to make a trade-off between network latency and power consumption.

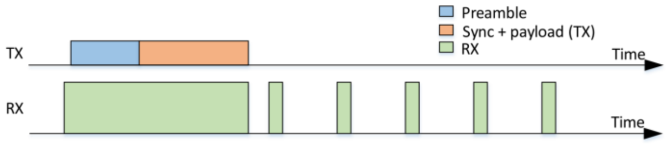


Fig. 4. Wake On Radio timeout

# V. PROTOCOL AND SOFTWARE STACK

## *A. Protocol stack*

The IEEE-802.15.4 protocol supports both beacon enabled and non-beacon enabled networks.

The formal mechanism supports synchronization in data transmission by dint of superframes that are configured and maintained by coordinators. The superframe contains guaranteed time slots, each of which can be assigned to a specific device in the network, preventing media access contention. The beacon-enabled network, therefore, enables devices to consume less energy since the receivers can be powered off during parts of the superframe.

As opposed to beacon enable network, the latter features less complexity in implementation and configuration. In this network, no beacon frames are transmitted except for scan operation. Source nodes attempt data transmission as long as the channel is idle and receiving nodes must be listening all the time. By virtue of Wake-On-Radio technology, the receivers can stay in sleep mode and only be waked up if a valid preamble signal is received, and therefore maximize power conservation with acceptable price of software complexity.

The complete IEEE-802.15.4 frame structure is presented in Fig. 5. The physical packet includes many fields, starting with the variable length preamble for synchronization, 8-bit start of frame delimiter (SPD) equal to "11100101" indicating the end of synchronization header and the start of the packet data, followed by 8-bit physical header and physical service data unit (PSDU) that has a variable length (0 to 127 bytes). The MAC frame structure is designed to be robust on a noisy channel at low cost in complexity. A MAC frame consists of three fields, namely MAC header (MHR), MAC service data unit (MSDU), and MAC footer (MFR).

TABLE I. SOME OF IMPLEMENTED PRIMITIVES IN THE SOFTWARE STACK

| Primitives | Description |
|---|---|
| PHY_PdDataReq() | Request the transfer of data to another device |
| PHY_PdDataIn() | Indicates the reception of data from another device |
| PHY_PlmeGetReq() | Request information about a given PHY PAN information base (PIB) attribute |
| PHY_PlmeSetReq() | Write the given value to the indicated PHY PIB attribute |
| PHY_PlmeCcaReq() | Request information about Clear Channel Assessment (CCA) |
| PHY_PlmeEdReq() | Request information about Energy Detection (ED) |
| MAC_mlmeSetReq() | Request information about a given MAC PIB attribute |
| MAC_mlmeGetReq() | Write the given value to the indicated MAC PIB attribute |
| MAC_mlmeScanReq() | Request a channel scan over a given list of channels |
| MAC_mlmeScanCnf() | Reports the result of the channel scan request |
| MAC_mlmeAssocReq() | Request an association with a coordinator |
| MAC_mlmeAssocCnf() | Reports the result of the association request |
| MAC_mlmeAssocInd() | Indicates reception of an association request |
| MAC_mlmeAssocRsp() | Response to an association indication primitive. |
| MAC_mlmeStartReq() | Initiate a new PAN |
| MAC_mcpsDataReq() | Request the transfer of data to another device |
| MAC_mcpsDataCnf() | Report the result of the data request |
| MAC_mcpsDataInd() | Indicates reception of a data frame |

Some of the protocol primitives that were developed in the project are briefly introduced in Table I.

Apart from PHY PIB attributes defined in the IEEE-802.15.4 standard, the software stack also introduces some extensions as an adaptation to WOR functionality (physical sniff mode attribute) and variable length preamble. With regard to the symbol length preamble, it does not only support 31 and 127 [7], but also 191 [5].

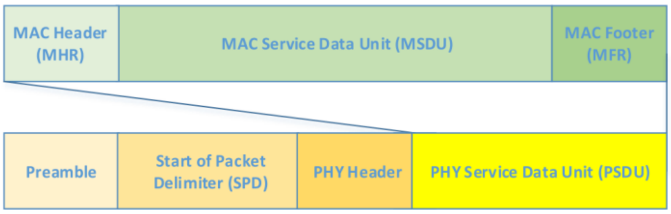


Fig. 5. Wake On Radio timeout

## B. Software stack

Our ultimate goal of the wireless sensor network is to provide domain scientists with the core technology which is reliable, to operate unattended, and easy to be configured and deployed. For these reasons, one simplification in software stack design and implementation is to deploy non-beacon enabled network. The software stack is demonstrated in Fig. 6, including core IEEE-802.15.4 implementation, hardware abstraction layer, target module, utility module, and a demo application layer illustrating how to work with the stack API.

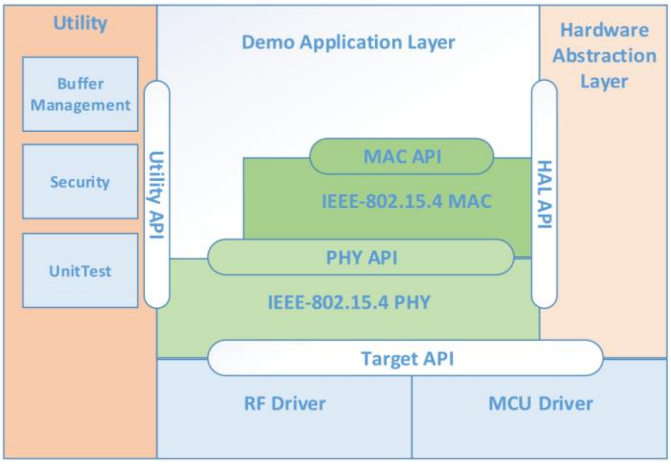


Fig. 6. Software stack architecture

## C. Network setup

An overview of necessary steps in setting up an IEEE-802.15.4-based non-beacon enabled network using a star topology is demonstrated in Fig 7.

Firstly the software stack should be initialized. Next a device must be selected to be the PAN coordinator of the network to be created. After issuing MAC-RESET-REQUEST primitive, the PAN coordinator starts performing scan operation. Based on results from the scan operation, the PAN coordinator then chooses PAN identifier for the network to be setup and select a short address for its own. After that it creates a network, joining other devices. Finally nodes that operate on the created network are able to exchange data with others using either short addresses which are provided by their associated coordinator or their long extended addresses.

Fig. 7.

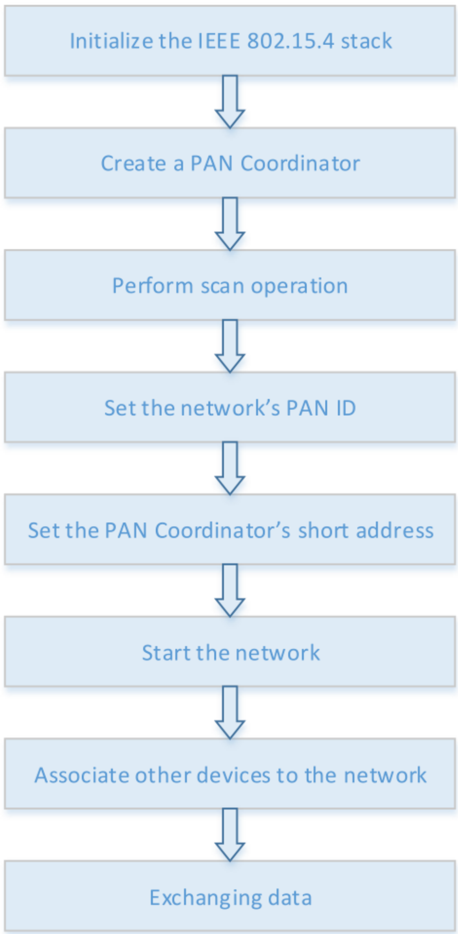


Fig. 8. Non-beacon enabled network setup process

## VI. MEASUREMENT

### A. *Measurement setup*

The measurements were performed using an oscilloscope which measures voltage dropped on Shunt resisters, as seen in Fig. 8, and eventually current flowing through the device is calculated by Ohm's law:

$$I = U / R \quad (1)$$

Fig. 9.

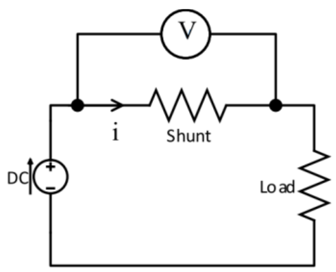


Fig. 10. Measurement setup.

On the development board some measurement points are designed to easily perform current consumption measurements on different segments. Description of those measurement points and according shunt resister values are shown in TABLE II. Also other timing parameters can be measured with the aid of the oscilloscope.

TABLE II. SHUNT RESISTER VALUES AND MEASURE POINT DESCRIPTION

| **Measurement points** | **Description** | **Shunt values** | **I_max (U_mx)** |
|---|---|---|---|
| I_TOT | Total current consumption | 4 Ω | 30 mA (100mV) |
| I_MCU | MCU current consumption | 112,5 Ω | 1mA (100mV) |
| I_RF | Radio frequency current consumption | 4 Ω | 30mA 100mV |

### B. *Measurement scenarios*

To measure the energy consumption of the IEEE-802.15.4 protocol stack, a simple demo application has been written, setting up a star-topology based non-beacon enabled network with one PAN Coordinator and one device. Besides what is necessary to perform wireless data transmission over IEEE-802.15.4 protocol, both devices to be measured do not contain any other hardware which could consume energy and bias the results. The measurement scenario represents a realistic example of wireless sensor nodes where the amount of involved hardware and energy consumption is tried to be kept at minimum.

In our scenario, end device does not query the coordinator whether it has pending data. Instead, the end device just transmit data with and without acknowledgment required after performing scan operation, association requesting operation, and starting operation.

TABLE III. MEASUREMENT CONFIGURATIONS

| **Parameters** | **Description** | **Values** |
|---|---|---|
| Preamble length | Length of the preamble used for packet transmission. The preamble length depends on WOR timeout and vice versa (e.g. longer timeout requires longer preamble length) | 31 or 191 bit |
| Deviation | FSK frequency deviation | 25 kHz |
| Rx-filter bandwidth | andwidth of the receiving filter | 100 kHz |
| Data rate | Data rate of the packets to be transmitted or received | 20 or 50 kbps |
| Data length | Length of the frame to send or receive | 100 bytes |
| Tramsmission Power | Power used to transmit packets | 15 dBm |
| macAckWait-Duration | Maximum number of symbols to wait for an acknowledgment frame to arrive following a transmitted data frame | 500 symbols |
| macMaxBE | Maximum value of backoff exponent (BE) in the CSMA-CA algorithm | 5 |
| macMinBE | Minimum value of backoff exponent in the CSMA-CA algorithm | 3 |
| macMaxCSMA Backoffs | Maximum number of backoffs the CSMA-CA algorithm will attempt before indicating a channel access | 4 |

| Parameters | Description | Values |
|---|---|---|
| | failure | |

Several parameters of the protocol stack can be modified to assess their impacts on the energy consumption. They are listed in Table III. Among them we are especially interested in the impacts of two parameters, namely preamble length and data rate, on sleeping period of the radio frequency transceiver. This is because the transceiver's sleeping time is propositional to the preamble length and inversely proportional to the data rate according to [6]. Therefore they are main factors determining energy consumption of the radio frequency device which is also the most power consumption of a wireless sensor node.

### C. Values to be measured

In order to assess performance of the software stack in energy consumption as well as estimate lifetime of a sensor node, these following values are to be measured.

In the experimental measurement current consumption of the radio frequency transceiver, microcontroller as well as of the whole development board are to be measured.

TABLE IV. OBJECTS UNDER MEASUREMENT

| Objects | Description |
|---|---|
| $T_{timeout}$ | Wake-On-Radio period |
| $T_{sleep}$ | Sleeing duration in idle state |
| $E_{sniff}$ | Energy consumption during sniffing |
| $P_{sleep}$ | Power consumption in sleeping state |
| $E_{1h}$ | Total energy consumption of the development board in one hour |
| $T_{tx}$ | Packet transmission duration |
| $E_{tx}$ | Energy needed for a packet transmission |
| $T_{rx}$ | Packtet reception duration |
| $E_{rx}$ | Energy needed for a packet reception |
| $T_{txACK}$ | ACK tranmission duration |
| $E_{txACK}$ | Energy needed for transmitting an ACK |
| $T_{rxACK}$ | ACK reception duration |
| $E_{rxACK}$ | Energy needed for reiving an ACK |

## VII. RESULTS

### A. Energy consumption in idle mode with WOR enabled

In the first scenario energy consumption is measured for a node in idle state with WOR enabled.

As can be easily seen in Table IV, the current consumption of the transceiver using 31 bit preamble length is three times as much as that while using 191 bit long preamble at the same data rate of 20kbps due to the shorter possible sleep intervals $T_{timeout}$. This results in 2.3-times lower total energy consumption in the latter case. Due to the fact that the wireless sensor node spends as much time as possible in idle state, the longer the preamble length the less energy it consumes.

As opposed to preamble length, data rate has inverse influence on energy consumption of the transceiver since higher data rate leads to shorter possible sleep intervals $T_{timeout}$, See TABLE IV. With the same 191 bit length preamble, the energy consumption of the transceiver operating at 50kbps is as approximately 2 time as much as that 20kbps.

TABLE V. AVERAGE CURRENT CONSUMPTION IN IDLE STATE WITH WOR ENABLED

| Configurations | | Current consumption | | |
|---|---|---|---|---|
| *Data rate [kbps]* | *Preamble length [bits]* | *I_MCU [mA]* | *I_RF [mA]* | *I_TOT [mA]* |
| 50 | 31 | 0,970 | 8,470 | 9,870 |
| 20 | 31 | 0,983 | 5,375 | 6,428 |
| 50 | 191 | 0,973 | 4,155 | 5,215 |
| 20 | 191 | 0,993 | 1,753 | 2,760 |

As a result, a combination of 191 bit length preamble and data rate of 20kbps offer the lowest energy consumption compared to other configurations, as shown in TABLE V.

A typical sniffing behavior in WOR mode is illustrated in Fig. 9, where the node wakes up, move to receiving mode, and then immediately goes back to sleep when no valid preamble signal is received.

Interestingly, energy consumption during sniff period and power consumption over sleeping time see similarities across different configurations. The total energy spent on idle state over a given amount of time is, therefore, proportional to the sleeping duration.

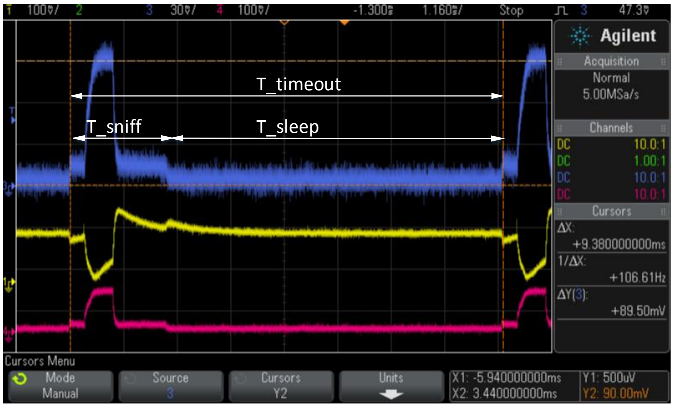


Fig. 11. Radio frequency transceiver using 191 bit length preamble and data rate of 20kbps in sniff mode

TABLE VI. ENERGY CONSUMPTION IN IDLE STATE FOR DIFFERENT CONFIGURATIONS

| Configurations | | | Measurement results | | | |
|---|---|---|---|---|---|---|
| **Data rate** | **Preamble length** | $T_{timeout}$ | $T_{sleep}$ | $E_{sniff}$ | $P_{sleep}$ | $E_{1h}$ |
| 50kbps | 31bit | 2,37ms* | 0,54ms | 0,532µJ | 3,43mW | 84,1J |
| 20kbps | 31bit | 2,98ms | 1,14ms | 0,527µJ | 3,44mW | 69,6J |
| 50kbps | 191bit | 3,76ms | 1,92ms | 0,545µJ | 3,41mW | 58,8J |
| 20kbps | 191bit | 9,38ms | 7,28ms | 0,547µJ | 3,40mW | 30,0J |

[a] * Minimum possible timeout from the current implementation

### B. Energy consumption during packet transmission

As seen in Fig. 10, the node first attempts to transmit the packet, going immediately to idle state before starting to receive the corresponding ACK.

Fig. 12.

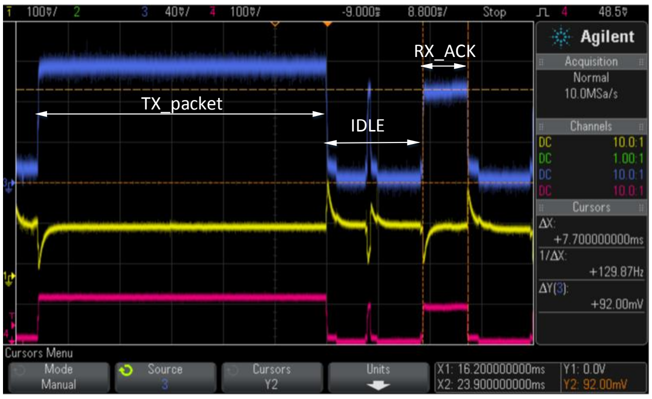


Fig. 13. Data transmission with ACK required

The TABLE VI shows that the time needed for a packet transmission is inversely proportional to data rate. The opposite is true for energy consumption during the transmission period. In addition, longer preamble caused longer transmission time. For example, it took 8.2 ms more to transmit the same packet using 191 bit length preamble than using 31 bit length preamble at a data rate of 20 kbps. Therefore energy consumed in the former case was approximately 17 % larger than in the latter case.

Unlike in the packet transmission, the energy spent in ACK reception sees slight differences. The reason for this is that the ACK frame comes with a length of 7 bytes, which is much shorter than length of the packet to transmit.

TABLE VII. ENERGY CONSUMPTION DURING A PACKET TRANSMISSION

| Configurations | | Measurement results | | | |
|---|---|---|---|---|---|
| ***Data rate*** | ***Preamble length*** | $T_{tx}$ | $E_{tx}$ | $T_{rxACK}$ | $E_{rxACK}$ |
| 50kbps | 31bit | 16,55ms | 1,38mJ | 2,6ms | 0,173mJ |
| 20kbps | 31bit | 41,1ms | 3,39mJ | 5,0ms | 0,345mJ |
| 50kbps | 191bit | 19,6ms | 1,65mJ | 6,4ms | 0,424mJ |
| 20kbps | 191bit | 49,4ms | 4,06mJ | 7,7ms | 0,515mJ |

*C.*

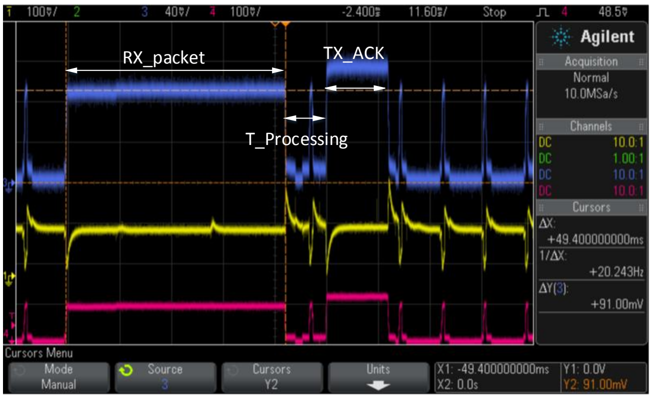


Fig. 14. Data reception with ACK required

### D. Energy consumption during a packet reception

As seen in Fig 11, the node starts processing the received frame before replying with an ACK.

TABLE VIII. ENERGY CONSUMPTION DURING A PACKET RECEPTION

| Configurations | | Measurement results | | | |
|---|---|---|---|---|---|
| ***Data rate*** | ***Preamble length*** | $T_{rx}$ | $E_{rx}$ | $T_{txACK}$ | $E_{txACK}$ |
| 50kbps | 31bit | 16,5ms | 1,12mJ | 2,5ms | 0,201mJ |
| 20kbps | 31bit | 41,2ms | 2,76mJ | 5,9ms | 0,491mJ |
| 50kbps | 191bit | 19,3ms | 1,31mJ | 5,6ms | 0,470mJ |
| 20kbps | 191bit | 49,4ms | 3,27mJ | 7,7ms | 0,623mJ |

### E. Sensor node lifetime estimation

It is assumed that a wireless sensor node operates under traffic load of 200 packet transmissions and receptions in one hour. The packets to transmit and to receive at one node are of 100 byte length. In addition, packets that require retransmission are considered as part of traffic load. Therefore the energy consumption in one hour is calculated as follow.

$$E_{total} = traffic * [E_{tx} + E_{rxACK} + E_{rx} + E_{txACK}] + \\ +N * [E_{sniff} + T_{sleep} * P_{sleep}] \quad (2)$$

Where $\mathbf{E_{total}}$ is total energy consumption of the development board; and N is number of times the node goes to idle state. N is a function of traffic load and WOR timeout, and calculated as follow.

$$N = \frac{1h - traffic * [T_{tx} + T_{rxACK} + T_{rx} + T_{txACK}]}{T_{timeout}} \quad (3)$$

The estimated lifetimes of a sensor node in different configurations are shown in TABLE VIII. The maximum lifetime is about 483.96 days, which is approximately 30% smaller than in the original approach. The new solution, however, provides real-time water monitoring ability in unit of millisecond and, more importantly, this is just the result in first stage of saving energy in the radio frequency transceiver.

TABLE IX. SENSOR NODE LIFETIME ESTIMATION FOR DIFFERENT CONFIGURATIONS WITH TRAFFIC RATE OF 200 PACKETS PER HOUR

| Configurations | | | Lifetime |
|---|---|---|---|
| ***Data rate*** | ***Preamble length*** | $T_{timeout}$ | |
| 50kbps | 31bit | 2,37ms | 180,32 days |
| 20kbps | 31bit | 2,98ms | 219,41 days |
| 50kbps | 191bit | 3,76ms | 257,19 days |
| 20kbps | 191bit | 9,38ms | 483,96 days |

In order to solve the higher energy consumption a modified WOR scheme that brings longer sleeping duration is presented. The modified WOR mechanism configured the sleeping period to be odd multiple of preamble transmission time, for example 3 times as illustrated in Fig. 12.

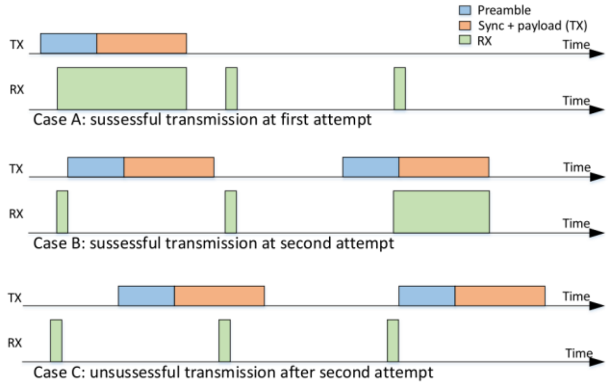


Fig. 15. Modified WOR mechanism for data transmission scenarios

As a result, with same values for data rate and preamble length, configurations using longer sleeping duration offer longer lifetime than those which stay less in sleep mode, as shown in Fig. 13.

Equally important, the estimated lifetime of a sensor node falls significantly with higher traffic load. For example, a node that operates at data rate of 20 kbps and uses 191 bit length preamble can run for approximately 1300 days under traffic rate of 100 packets per hour. However its lifetime will be about 500 days shorter when the traffic load goes up to 1000 packets per hour.

Even though these numbers still seem to be high in comparison with other ultra-low energy approaches, it should be taken into account that the given approach allows fast real-time capabilities, whereas legacy sleeping devices might not be accessible, as they are sleeping deeply for hours or even days.

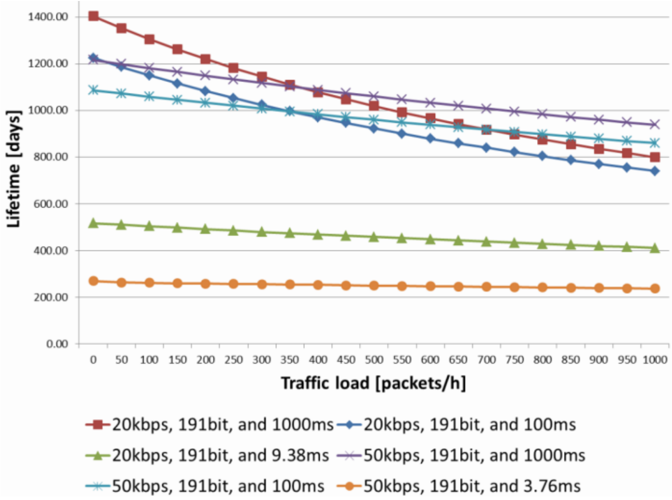


Fig. 16. Lifetime versus traffic load for different configurations

## VIII. CONCLUSION

This paper has shown our first attempt to develop a scalable communication backbone for deploy the low-power real-time water monitoring system, suitable for deployment in Pakistan's large irrigation network in Pakistan. The software stack works properly according to the IEEE-802.15.4 specification. Measurement results have addressed the major factors on total energy consumption, namely preamble length and data rate. By maximizing preamble length and operating at reasonable low data rate, the energy consumption efficiency is significantly improved.

In addition, the paper has introduced a modified WOR scheme which is only advantageous when sending occur at large intervals. Furthermore it can cost at network latency in case packets fail to be transmitted at the first or second attempt. Nevertheless the new scheme is quite promising in terms of energy saving and deserves furthermore studies.

In environmental monitoring systems, where objects to be observed such as water level, temperature, and velocity are unlikely to change over the time period of milliseconds, it is reasonable to increase the response time unit to second or even minute and the system is still considered real-time. With this in mind, further more sophisticated software-based energy saving schemes can be implemented in the next step in order to deeply reduce energy consumption without losing real-time functionality of the monitoring system. The results of further work will be presented in other article.

## ACKNOWLEDGMENT

This work was partially supported by German Academic Exchange Service (DAAD) in the 2013/2014 program "German-Pakistani Research Collaborations". The authors are grateful for this support.

## REFERENCES

[1] Muhammad Arif Raza, Muhammad Ashfaq, and Ifran Ahmad Baig, "Institutional reforms in irrigation sectors of Punjab (Pakistan) and their

impact on sugarcane productivity", Journal of Agricultural Research 47, pp 63-72, 2009

[2] Zahoor Ahmad, Ehsan U. Asad, Abubakr Muhammad, Waqas Ahmad, and Arif Anwar, "Development of a Low-Power Smart Water Meter for Discharges in Indus Basin Irrigation Network", 2013.

[3] http://iwmi-lhe-vm02.cloudapp.net/login.aspx

[4] Texas Instrument, "MSP430F5438A Datasheet".

[5] Texas Instrument, "CC1120 High-Performance RF Transceiver for Narrowband Systems", SWRS112G, June.

[6] Texas Instrument, "CC112x/CC120x RX Sniff Mode", SWRA428A, October 2013.

[7] IEEE, "IEEE Standard for Local and metropolitan area networks – Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)", ISBN 978-0-7381-6684-1 STDPD97126, 2011

[8] Ahmad, Z., Muhammad, A." Low Power Hydrometry for Open Channel Flows" (IECON'2014), The 40th Annual Conference of the IEEE Industrial Electronics Society, Dallas, TX – USA.

[9] Maxbotix; MB7380 Outdoor Ultrasonic Range Finders. http://www.maxbotix.com, 2012.

[10] Riaz W., Muhammad A., 'Electronic Discharge Measurement in Small Outlets Using a Long Throat Flume". LUMS Technical Report. 2014.

[11] Briscoe, J., Qamar, U.: Pakistan's Water Economy Running Dry. Oxford Univ. Pr. (2009).

[12] Yang S., Liang G., Zhao W., Xie D, Huang Z., "Design of Low-power Consumption Ultrasonic Open Channel Flow Meter", (ICEMI 2009), The 9th Int'l Conf. on Electronic Measurement & Instruments.