Abstract The past decade has seen significant advancement in the field of mobile devices. Various smart devices such as cellular phones, tablets and PDAs have become universal tools in our everyday lives. Their versatility is based on their computing power, portability and their integration with other devices and services such as the World Wide Web. However, these smart devices have an even wider usability spectrum. They can also be used for wireless industrial measurements using existing sensors. The wireless connectivity of existing industrial sensors is achieved by equipping them with a Bluetooth module, which digitizes the data and passes it to any Bluetooth capable smart device for further processing, evaluation and logging. This paper describes the specially designed Bluetooth platform for wireless measurements all the way from the basic concept, through hardware, firmware and software implementation, to the sample tests and measurements.

Keywords Sensors, Wireless, Bluetooth, Data Acquisition, Condition Monitoring

1. Introduction

With the progress of industrial modernization a large number of sensors and indicators need to be monitored in order to achieve the optimal production efficiency and product quality. Data is gathered either automatically over wired data acquisition systems [1] or in modern factories by wireless using wireless sensor networks and/or mobile robots for data collection [2-4].

When installing new sensors or troubleshooting existing sensors, engineers often need special measuring tools, such as oscilloscopes or more sophisticated sensor signal conditioning equipment, in order to check the correct operation of the industrial sensors. Such measuring equipment is often expensive and inflexible and can be used only for a specific type or brand of industrial sensors [5]. Moreover, most measuring tools have wired connection to the sensors; meaning each time the measurement must be carried out, the cables must be connected [6]. Each measuring tool has its own user-interface, which is often awkward and tool-specific. Such a measuring tool is used, for example, in the field of industrial signal conditioning of hydraulic systems and is shown in Figure 1. [7]

In today’s most sophisticated automation systems, such as manipulators, robots and mobile robots, there are numerous sensors, which are responsible for their correct operation. The causes of faults in such systems are hard
to detect and isolate, especially if the error appears only during operation. Therefore, advanced algorithms are being developed for FDI (Fault Detection and Isolation) [8, 9]. Such algorithms often provide the operator with general error descriptions, for example ‘clamp error’ and do not provide the exact fault location within the subsystem. Maintenance personnel often use trial and error approaches to find out which component has failed. Such methods can be time consuming.

2. Related works

Wireless sensors have been extensively studied in literature over the last two decades. Most of the generally available sensors use one of the standard wireless communication protocols – Bluetooth (IEEE 802.15.1), ZigBee (IEEE 802.15.4) or Wi-Fi (IEEE 802.11) [10, 11]. There are numerous specific wireless sensor solutions available for different application areas. Most wireless sensor solutions have a built in transducer, signal conditioner and wireless radio. For examples see [12-16]. There are few solutions either for converting existing sensors to wireless, in such a way that we can still use them when wired as before [17, 18], or for making wireless data available to smart devices.

2.1 Brief overview of the most commonly used wireless communication standards

As mentioned before the most commonly used wireless communication protocols in the field of wireless sensor technology are Bluetooth, ZigBee and Wi-Fi. Each of them offers different advantages and also disadvantages. A comparison of the communication protocols is shown in Table 1.

| Standard | Bluetooth     | ZigBee      | Wi-Fi       |
|----------|---------------|-------------|-------------|
| Application focus [19] | cable replacement | monitoring and control | web, email, video |
| Success metrics [19] | cost, convenience | reliability, power, cost | speed, flexibility |
| IEEE spec. | ISM 2.4 GHz | ISM 2.4 GHz | ISM 2.4, 5 GHz |
| Frequency band | 3 with EDR | 0.25 | 54 |
| Max data rate [Mib/s] | 0.72 | 10 - class 2 | 100 |
| Nominal range [m] | 10 - class 2 | 100 - class 1 | 100 |
| Number of RF channels | 79 | 161 (2.4 GHz) | 14 (2.4 GHz) |
| Basic cell | piconet | star | BSS |
| Max cell nodes | 8 | >65000 | 2007 |
| Encryption | E0 | AES | AES |
| Protection | 16-bit CRC | 16-bit CRC | 32-bit CRC |
| Power consumption [mW] | RX: 84.6 | 81 | 709.5 |
| TX | 102.6 | 74.1 | 722.7 |
| Nominal energy consumption [mJ/Mb] | RX | 117.5 | 324 | 13.14 |
| TX | 142.5 | 296.4 | 13.38 |

Table 1. Comparison of the wireless communication protocols [20]

2.1.1 Wi-Fi

Wi-Fi uses a CSMA/CA (carrier sense multiple access with collision avoidance) access scheme, meaning that every Wi-Fi device should listen to the medium before transmitting. Transmission is allowed only if the medium is sensed to be idle. Wi-Fi networks operate over one of the 14 22 MHz wide channels defined for the 2.4 GHz ISM (industrial scientific and medical) band, meaning that no more than three networks can be simultaneously operated within the same area without interference [21].
2.1.2 ZigBee

Similarly to Wi-Fi, ZigBee devices use a CSMA/CA channel access algorithm and a 2.4 GHz ISM band. The difference between the two is that ZigBee’s channels are a lot narrower (just 2 MHz) and therefore do not overlap, meaning that 16 ZigBee networks can coexist within the same area. [21]

2.1.3 Bluetooth

Bluetooth is designed for cable replacement and connection-oriented services with low power consumption and short range operations. The output power and range depend on the device class (see Table 2).

| Device Class | Range   |
|--------------|---------|
| Class 1      | 100 m   |
| Class 2      | 10 m    |
| Class 3      | 1 m     |

Table 2. Bluetooth device classes and theoretical range

Bluetooth uses 79 1 MHz wide channels within the 2402-2480 MHz frequency range. To reduce interference FHSS (frequency hopping spread spectrum) and AFH (adaptive frequency hopping) – since v1.2 – are used to spread the signal across all 79 channels. More Bluetooth networks can coexist within the same area by employing different hopping patterns. The Bluetooth channel access scheme is based on TDD (time division duplex). The master may start transmitting at even numbered slots and the slave at odd.

The low channel width and FHSS technique are fairly ineffective in contrast to Wi-Fi interference (one Wi-Fi channel pollutes 22 Bluetooth channels), resulting in a drastic Bluetooth signal degradation of up to 68%. [21]

3. Bluetooth platform for wireless measurements

Most modern smart devices have Bluetooth and Wi-Fi built in and therefore no additional hardware is required for these two wireless protocols. This fact rules out ZigBee, which would need a special, device specific hardware or a Bluetooth or Wi-Fi to ZigBee converter. Such a converter would increase the complexity and latency of the connection and probably introduce some compatibility issues. Although Bluetooth is slower and has a smaller number of cell nodes compared to Wi-Fi, Bluetooth has been chosen as the appropriate protocol for wireless measurements. This is due to a practical reason. If a user wanted to measure sensor signals and browse the internet or log data to a remote server at the same time, the smart device would have to switch between a router with internet access and an AD-HOC wireless sensor network, which is unacceptable in such applications.

3.1 Basic concept

The basic idea of the platform is to add wireless access to existing industrial sensors so that data can be read wirelessly when needed, with no effect on the existing sensor connection. Figure 2 shows the basic idea of using the platform: 1. disconnect connected sensor, 2. add connector module and 3. connect sensor back via the connector module.

![Figure 2](image)

DIN EN 175301-803 connectors

Type A

Type B

Type C

M12x1 4 pole

M12x1 5 pole

Figure 3. More commonly used industrial sensor connectors

The connector module has a built in EEPROM, which stores all the sensor properties including name, description, measured quantity range, output signal range (e.g., 4 - 20 mA), etc. All the properties are defined on the first data acquisition. The main purpose of the connector module, along with sensor properties storage, is to enable hot-swap for the main module, meaning that the main module can be added or removed whilst the sensor is operating. Moreover, such a configuration is ideal for factories with a lot of sensors that need to be accessed wirelessly only when performing maintenance on a specific machine. On such a machine each sensor has...
its own connector module and the main module is added only to those sensors where wireless access is needed (Figure 4). This minimizes the cost by reducing the number of main modules needed.

Figure 4. Sensors with connector module installed – one with and one without the main module

If data from a specific sensor needs to be measured, the main module is connected to the connector module. Now wireless access to the sensor is available and data can be read using a smart device with built in Bluetooth and the appropriate application installed.

3.2 Hardware configuration

As described before there are two parts: the connector module and the main module.

3.2.1 The connector module

The connector module consists of three connectors; first one is used to connect the connector module to the sensor. Then the second connector of the opposite gender is used to connect the connector module with the signal cable. The third one is a DE-15 connector used for connection with the main module. Besides the two connectors, the connector module has a built in electronic signal routing between the three connectors and an EEPROM chip with FC (Inter-Integrated Circuit) bus. The schematic outline of the connector module is shown in Figure 5.

Figure 5. Schematic outline of the connector module

There are several different types of connector modules: analogue connector modules, RS232 connector modules, digital input/output (DIO) connector modules, etc. The signal routing circuit differs from connector to connector. The signal routing circuit links the signal lines to the appropriate pin on the main module and contains a current sense resistor (analogue connector module) or RS232 to UART voltage level shifter (RS232 connector module) or transistors (DIO connector module). The connector module type is chosen depending on the sensor type and connector.

3.2.2 The main module

Configuration of the main module is a bit more complicated than the connector module. Its block diagram is shown in Figure 6. The main parts of the main module are the DC/DC converter (for efficient supply of power), the microcontroller unit (MCU), the Bluetooth module and the ADC (Analogue-Digital Converter).

Figure 6. Main module layout and block diagram

The heart of the main module is a MCU. The MCU is a Microchip dsPIC, running at 3.3 V and 20 MHz. The clock is provided by a crystal (XTAL) oscillator. The MCU features peripheral pin select technology, built in hardware ADCs, DACs (Digital-Analogue Converter), UART (Universal Asynchronous Receiver/Transmitter, FC and SPI (Serial Peripheral Interface). The UART is used for communication with the Bluetooth module (red) and FC (green) for communication with EEPROM and an external 16 bit ADC. 3 digital pins are used to drive LED indicators, which enable visualization of the main module status.
The most important part is the Bluetooth module. A Roving Networks RN-41 Class 1 Bluetooth module was chosen for this application. This is mostly because it is capable of connecting with all the smart devices available on the market, including Apple devices, which require an external authentication coprocessor. Moreover, the RN-41 automatically handles communication with an authentication coprocessor on a separate FC bus (orange). RN-41 is connected to the MCU via the UART configured to a baud rate of 115200 Bd/s with RTS/CTS (Request to Send/ Clear to Send) flow control, no parity and 1 stop bit (N=10 Bd/s). This yields a maximum theoretical transfer rate of 11.25 kibiB/s (eq. 1). This corresponds to a sample rate of 7.56 k/s when using 16 bit (2 byte) AD conversion (eq. 2).

\[
R_b = \frac{B}{N}, \quad \text{where} \]

\[
R_b = \text{...data rate \left[ B/\text{s} \right]} \\
B = \text{...baud rate \left[ B/\text{d} \right]} \\
N = \text{...number of bauds required to transmit one byte\left[ B/\text{B} \right]} \\
\]

\[
f_s = \frac{R_b}{N}; \quad \text{where} \\
f_s = \text{...sample rate \left[ S/\text{s} \right]} \\
R_b = \text{...data rate \left[ B/\text{s} \right]} \\
N = \text{...number of bytes in one sample\left[ B/\text{S} \right]} \\
\]

The external ADC offers accurate 16 bit wide conversion with data rates of up to 7 k/s, with built in averaging including voltage, current and power calculation. The gain error of the ADC is less than 0.1% and the ADC does not need any external voltage references. Moreover, the ADC is factory calibrated and therefore provides accurate voltage and current readings.

A DC/DC converter was preferred over a linear voltage regulator mainly due to better efficiency and wider input range. The power losses on the ideal linear voltage regulator (eq. 3) would be bigger than 1 W when being powered from 24 V line and transmitting (50 mA).

\[
P_{\text{loss}} = (U_{\text{in}} - U_{\text{out}}) \cdot I_{\text{TX}}; \quad \text{where} \\
P_{\text{loss}} = \text{...power loss \left[ \text{W} \right]} \\
U_{\text{in}} = \text{...input voltage \left[ \text{V} \right]} \\
U_{\text{out}} = \text{...output voltage \left[ \text{V} \right]} \\
I_{\text{TX}} = \text{...current consumption while transmitting \left[ \text{A} \right]} \\
\]

3.3 Firmware

Custom firmware has been developed for the platform. The firmware is written in C programming language using Microchip’s Integrated Development Environment (IDE) called MplabX. The Microchip C30 compiler was used to compile the code and circuit debugger ICD3 was used to debug it. Firmware that is running on the MCU consists of initialization procedures for all the peripheral devices, an UART command parser/handler and a data acquisition timer. Both the UART command parser and the data acquisition timer are interrupt-driven, which reduces the number of checks that should be carried-out at each iteration of the main loop. A simplified flowchart of the firmware program is shown in Figure 7, where synchronous tasks are presented in blue and asynchronous (interrupt driven tasks) in orange. Interrupt Service Routine (ISR) gets called at the specified hardware event.

![Figure 7. Firmware program flowchart](image-url)

3.3.1 Communication protocol

A special protocol has been developed, which is used for communication between the main module and the software on the smart device. The protocol is a master/slave type, where the smart device acts as a master and the main module as a slave. This means that the main module may transmit only when the master wants it to. Available commands are divided into three groups: GET commands, SET commands and ACTION commands. Using the GET commands the software requests particular data from the main module, for example, the name of the sensor. SET commands are used to send data to the main module. ACTION commands are special commands that, for example, initiate data acquisition with a specific data rate.

The protocol uses the Bluetooth Serial Port Profile (SPP), which is based on the Bluetooth Core Protocol and RFCOMM protocol. Bluetooth Core Protocol is divided into four parts: The first part is Baseband Protocol (BP), which is used to control the Bluetooth radio. Link Management Protocol (LMP) is used to establish the link between Bluetooth devices. The third and one of the more important parts is the Logical Link Control and Adaptation protocol (L2CAP), which provides
segmentation and the reassembly of on-air packets. The last part of the Bluetooth Core Protocol is the Service Discovery Protocol (SDP), which allows a device to discover services supported by other devices. Services are identified by a Universally Unique Identifier (UUID). In our case SDP discovers whether the SPP service is available. The RFCOMM protocol is used for serial cable emulation. Figure 8 shows the described protocol stack used in our Bluetooth measurement platform [22].

![Bluetooth platform protocol stack](image)

**Figure 8.** The Bluetooth platform protocol stack

### 3.3.2 Communication example

As soon as the main module and the smart device are connected the communication begins with a HELLO command sent by the smart device, which is always replied to with hello, which means that the main module is ready and the connector module is already configured properly. After that some information about the sensor and the connector module is read from the EEPROM of the connector module with the GCT (Get Connector Type), GNAME (Get NAME) and GDSC (Get DeSCription). Later on more values specific to each connector type are read. For an analogue connector module: GUNIT (Get UNIT), GLSB (Get Least Significant Bit), GK (Get slope (K)), GN (Get y-intercept (N)) and GST (Get Sample Time). All those values are required for correct calculation of the measured value. If we want to change the sample time (ST), the ST variable can be set using a SST command. As soon as the module is set to our needs, the measurement may be initiated by sending the START command. After the START command the module replies with 2 bytes (16 bits) of data each x ms as defined in ST variable. The measured value is calculated using slope and y-intercept values (which differ with sensor range) and the least significant bit (which differs with voltage or current measurement) (eq. 4).

\[
V = \text{LSB} \cdot \text{K} \cdot V_{\text{raw}} + N \quad \text{where}
\]

\[
V \quad \text{value} \quad \text{[unit]}
\]

\[
\text{LSB} \quad \text{least significant bit} \quad \text{[mA/\text{bit}]} \quad \text{or} \quad \text{[V/\text{bit}]}
\]

\[
V_{\text{raw}} \quad \text{raw value} \quad \text{[bit]}
\]

\[
K \quad \text{slope} \quad \text{[unit]} \quad \text{or} \quad \text{[unit/V]}
\]

\[
N \quad \text{y intercept} \quad \text{[unit]}
\]

(4)

The measurement is stopped using the STOP command. The communication example described above is shown in Figure 9.

**Figure 9.** Communication example between the Smart Device (SD) and the Main Module (MM)

### 3.4 Software

Good software is one that makes the hardware useful. Along with the basic functionality, the graphical user interface is the most important aspect of user satisfaction. The user interface should therefore be transparent, easy to navigate and use pleasing graphics. The whole concept of the user interface must be very intuitive.[23]

The used software is a custom made program that implements the described communication protocol. It is written in Java using Xcode IDE provided by Apple Inc. The Xcode features programming, compiling, debugging and the simulation of the code and has a lot of built in objects and libraries, which simplify the development process.

Used software consists of the following segments:

- **Settings.**
- **Cockpit (shown in Figure 10) - the area that shows all the data. Users can freely organize the cockpit and determine how the data will appear.**
- **Event Manager - a place where all previous events (measurements) can be accessed. Measurements are organized by client, project and events.**
• Calculator - calculate all main industrial quantities (electrical, hydraulic, pneumatic ...).
• Reporter – report generation tool.

4. Test results

The range and throughput of the Bluetooth measurement platform was tested in order to prove its usability for industrial measurements and maintenance.

4.1 Range measurements

Range measurements were performed using the main module and a 3G version of Apple iPad 2. The maximum range is a range where iPad can still connect to the main module to send and receive data (the module responds properly to START and STOP commands). The delays and jitter were not measured. Range measurement results are presented in Table 3.

| Measurement | Range     |
|-------------|-----------|
| 1           | through window with LoS + Wi-Fi | 40 m |
| 2           | laboratory environment through 3 brick walls | 28 m |
| 3           | laboratory environment through one reinforced concrete wall and one brick wall | 10 m |
| 4           | industrial environment through one reinforced concrete wall and one brick wall | 10 m |

Table 3. Range measurement results

Measurement 1 was taken in open space with line of sight (LoS) of the device through a window. Measurements 2 and 3 were taken in laboratory environments through different obstacles. A laboratory environment means that there are computers operating nearby and the area is polluted with at least one Wi-Fi computer network. Last of all an industrial environment measurement (4) was performed through the same walls as in measurement 3, with an 18 kW motor running on a frequency inverter between the main module and the iPad. Measurement 4 showed no range decrease compared to the office environment.

4.2 Throughput measurements

Throughput measurements were performed using the main module and a 3G version of Apple iPad 2. The measurement examined whether all the data had arrived from the main module to the iPad. The main module sampled the voltage signal with 16 bit accuracy and sent data over Bluetooth to the iPad. Four different sample rates were chosen for the test. Various obstacles had almost no effect on the throughput. The test was performed using the same commands as shown in Figure 9. The only difference was the sample rate, which was different for each measurement.

Table 4 shows the results of the throughput measurements. It was noticed that on sample rates greater than 500 S/s data loss was detected. For some reason the CTS signal of RN-41 had become high and disabled the transmission, which caused the UART TX FIFO buffer in the MCU to overflow. This might have happened because the RN-41 was unable to transmit all the data in time or the iPad was unable to receive all the data due to a bug in the benchmarking application.

| Sample rate | Raw serial data rate | Result |
|-------------|----------------------|--------|
| 100 S/s     | 200 B/s              | Ok     |
| 500 S/s     | 1000 B/s             | Ok     |
| 1 kS/s      | 1.95 kB/s            | Loss   |
| 5 kS/s      | 9.77 kB/s            | Loss   |

Table 4. Throughput measurement results

5. Conclusion

The main objective of this work was to develop a user friendly platform for wireless measurements based on Bluetooth wireless protocol that can be used with existing sensors. Such a system has many advantages, especially when maintaining and troubleshooting existing industrial machinery including robots. For most measurement systems the jitter, delays and communication breakdown are not a problem, as long as no data loss occurs and all the data can be read by the smart device. On the other hand, in safety critical or closed loop control applications, the delays or communication breakdowns would cause a problem and therefore such wireless systems cannot replace wired ones.

The measurement results have proved that the developed Bluetooth platform can be used for reliable industrial measurements with sample rates of up to 500 S/s. Therefore the platform could be a replacement for sophisticated sensor signal conditioning equipment in the near future, replacing the brand-specific equipment with the universal measurement system for all industrial sensors.
Some useful applications using the developed Bluetooth platform in robotics are briefly described and referenced below:

- Such a platform can enable existing sensors to acquire data in remote places with no database access. Acquired data is collected using mobile robots in specific intervals [24, 25].
- Enables robot interaction with sensors in close proximity. For example on nearby machines or other instruments. This is a need for maintenance robots, whether teleoperated, under supervisory control, or autonomous, especially if they are working in hazardous environments [26].
- Enables fast and simple robotic tool change – no connection/disconnection of sensor cables [27].
- Provides a wireless interface for all sensors in the field of factory automation including robots, resulting in simplified installation, reduced engineering efforts and eliminated cable wear and tear factors [28].

6. Further work

Future research plans are to perform a detailed measurement of the platform’s performance within an industrial environment, including latency and jitter measurements [29]. If the measurements were to provide give satisfactory results, the time synchronizations of different channels would be implemented [30]. Moreover, the platform would firstly be tested using controlling actuators and later for closed loop control. If the platform is used for control, jitter, communication breakdown and security issues would have to be taken into account. For example, what to do if there is no data or there is large jitter from a closed control loop sensor; should the loop be stopped or continue running using the last received value? A lot of effort will be put into such intelligent fault detection and isolation systems in the future. Moreover, any existing security holes should be found and fixed to prevent unauthorized access via the main module to control loop properties and actuators. To ease the firmware update procedure, a bootloader on the MCU is planned. A bootloader would enable ‘over the air’ firmware updating on the main modules.

More connectors are also planned – for strain gauges, piezoelectric and CANbus sensors. Moreover, we want to add the possibility of powering the sensor and main module from a battery or solar cell, which would enable using any sensor as an autonomous wireless sensor.

7. Acknowledgement

Operation part financed by the European Union, European Social Fund. Operation implemented in the framework of the Operational Programme for Human Resources Development for the Period 2007-2013, Priority axis 1: Promoting entrepreneurship and adaptability. Main type of activity 1.1.: Experts and researchers for competitive enterprises.

8. References

[1] Li Y, Rong J, Huang S (2011) Technology study of new industry data collection and monitoring system. 2011 International Conference on Electronic and Mechanical Engineering and Information Technology (EMEIT). pp. 4709-4712.
[2] Freeman J, Surendran S (2011) Remote monitoring of indoor environment using mobile robot based wireless sensor network. 2011 6th International Conference on Computer Science & Education (ICCSE). pp. 1080-1084.
[3] Tekdas O, Karnad N, Isler V (2011) Energy-Efficient Data Collection from Wireless Nodes Using Mobile Robots. In Pradalier C, Siegwart R, Hirzinger G editors. Robotics Research. Heidelberg: Springer Berlin. pp. 697-712.
[4] Di Francesco M, Das S, Anastasi G (2011) Data Collection in Wireless Sensor Networks with Mobile Elements: A Survey. ACM Trans. Sen. Netw. vol. 8(7): 1550-4859. pp. 1-31.
[5] Parker Hannifin Corporation (2010). Available: http://www.parker.com/literature/Quick%20Coupelin g/Bulletin%203800%20SM-EZ%20-10.pdf. Accessed 2012 Jun 5.
[6] HYDAC ELECTRONIC GMBH. Available: http://www.hydac.com/fileadmin/pdb/pdf/PRO0000 0000000000001806410011.pdf. Accessed 2012 Jun 5.
[7] Les K (2011) Razvoj novega orodja za diagnostiranje hidravličnih sistemov. Fluidna Tehnika 2011 (FT2011). pp. 257-264.
[8] Hsiao T, Weng M-C (2012) A hierarchical multiple-model approach for detection and isolation of robotic actuator faults. Robotics and Autonomous Systems vol. 60(2): 0921-8890. pp. 154-166.
[9] Mehranbod N, Sorough M, Panjapornpon C (2005) A method of sensor fault detection and identification. Journal of Process Control vol. 15(3): 0959-1524. pp. 321-339.
[10] Egea-Lopez E, Martinez-Sala A, Vales-Alonso J, García-Haro J, Malgos-Sanahuja J (2005) Wireless communications deployment in industry: a review of issues, options and technologies. Computers in Industry vol. 56(1): 0166-3615. pp. 29-53.
[11] Yick J, Mukherjee B, Ghosal D (2008) Wireless sensor network survey. Computer Networks vol. 52(12): 1389-1286. pp. 2292-2330.
[12] Martos J, Montero Á, Torres J, Sorret J, Martínez G, García-Olencia R (2011) Novel Wireless Sensor System for Dynamic Characterization of Borehole Heat Exchangers. Sensors vol. 11(7). pp. 7082-7094.
[13] Kameoka T, Kondo H, Isozaki M, Nakamura T, Kitamura H, Kimura J (2011) Environmental Measurement for indoor plant factory with micro-miniature size wireless sensor. 2011 Proceedings of SICE Annual Conference (SICE). pp. 2811-2816.

[14] Lin C, Lei Y (2011) Design of Car Tire Pressure Monitoring System Based on ARM and Wireless Sensor. Computer Measurement & Control vol. 19(2): 1671-4598. pp. 283-285.

[15] Vražić M, Gašparac I, Kovačić M (2012) Measurement Systems for Electrical Machine Monitoring. In Haq Z editor. Applied Measurement Systems. Rijeka: InTech.

[16] Bi X, Zhu M (2011) An intelligent actuator controller for wireless sensor/actuator networks. 2011 IEEE Symposium on Industrial Electronics and Applications (ISIEA). pp. 42-46.

[17] Marex Y, Schrag D, Bloj R (2011) Low Power Design for Wireless Instrumentation. SENSOR+TEST Conferences 2011. pp. 439-444.

[18] Li B, Zou J (2011) The design of oil drilling wireless data acquisition system. 20011 International Conference on Electrical and Control Engineering. pp. 1810-1813.

[19] Lu K-Y (2011) A plug-and-play data gathering system using ZigBee-based sensor network. Computers in Industry vol. 62(7): 0166-3615. pp. 719-728.

[20] Lee J-S, Su Y-W, Shen C-C (2007) A Comparative Study of Wireless Protocols: Bluetooth, UWB, ZigBee, and Wi-Fi. IECON 2007. 33rd Annual Conference of the IEEE Industrial Electronics Society. pp. 46-51.

[21] Garroppo R, Gazzarrini L, Giordano S, Tavanti L (2011) Experimental assessment of the coexistence of Wi-Fi, ZigBee, and Bluetooth devices. 2011 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM). pp. 1-9.

[22] Liu J, Sun G, Zhao D, Yao X, Zhang Y (2012) MCU-Controlling Based Bluetooth Data Transferring System. Procedia Engineering, vol. 29: 1877-7058. pp. 2109-2115.

[23] Browne K, Anand C (2012) An empirical evaluation of user interfaces for a mobile video game. Entertainment Computing vol. 1(3): 1875-9521. pp. 1-10.

[24] Tekdas O, Iser V (2009) Using mobile robots to harvest data from sensor fields. Wireless Communications, IEEE vol. 16(1): 1536-1284. pp. 22-28.

[25] Jea D, Somasundara A, Srivastava M (2005) Multiple Controlled Mobile Elements (Data Mules) for Data Collection in Sensor Networks. In Prasanna V, Iyengar S, Spirakis P, Welsh M editors. Distributed Computing in Sensor Systems. Heidelberg: Springer Berlin. pp. 244-257.

[26] Parker L, Draper J (2007) Robotics applications in maintenance and repair. In Nof S editor. Handbook of Industrial Robotics: John Wiley & Sons, Inc. pp. 1023-1036.

[27] Gyimothy D, Toth A (2011) Experimental evaluation of a novel automatic service robot tool changer. Advanced Intelligent Mechatronics (AIM), 2011 IEEE/ASME International Conference on. pp. 1046-1051.

[28] Kjellsson J, Vallestad A E, Steigmann R, Dzung D (2009) Integration of a Wireless I/O Interface for PROFIBUS and PROFINET for Factory Automation. Industrial Electronics, IEEE Transactions on vol. 56(10): 0278-0046. pp. 4279-4287.

[29] Yu K, Gidlund M, Åkerberg R, Björkman M (2011) Reliable and Low Latency Transmission in Industrial Wireless Sensor Networks. Procedia Computer Science, vol. 5: 1877-0509. pp. 866-873.

[30] Ranganathan P, Nygard K (2010) Time Synchronization in Wireless Sensor Networks: A Survey. International Journal of UbiComp (IJU) vol. 1(2): 0976-2213. pp. 92-102.