A novel self-configuring ethernet-based smart sensor application for remote condition monitoring

Syed Adnan Yusuf [1] , Siddharth Shetty [1] , Nigel Wilkinson [1] , David J Brown [2]

[1] STS Defence Ltd
Mumby Road
Gosport, United Kingdom
{adnan.yusuf, n.wilkinson, siddharth.shetty}@sts-defence.com

[2] Institute of Industrial Research
University of Portsmouth
Portsmouth, United Kingdom
d.j.brown@port.ac.uk

Abstract— **The research proposes a smart data collection and monitoring system where large amounts of data are to be captured from multiple sensors in remote environments. Based on the status of the broadcasted data, the underlying model filters the vast amounts of data and sends only critical information to the monitoring station. The system comprise of a three-tier embedded and soft-computing architecture that integrates an array of smart sensors built in a smart sensor module (SSM), centralized controlling unit (CCU) application over a solid-state XP system stationed at a remote operating vehicle (ROV) and a monitoring front-end to present critical information at the ground-based monitor (GBM) server.**

The proposed methodology develops a SSM based solution where each SSM should be able to plug onto an existing LAN and automatically identify itself to the centralized controller. The network contains an array of SSMs where each SSM receives the configuration setup (from the CCU), starts collecting data and updates the data-rate based on the impending situation. The whole framework operates without any human intervention or additional setup thereby making the system highly extensible. The SSM is prototyped on a KEIL evaluation board (with an ARM Cortex M3 microcontroller).

The software tier operates over the raw data received from the SSM that is evaluated against a set of range and domain criteria defined for any discrepancies in any of the sensor outputs. In case of an emergency, the CCU sends an emergency signal back to the SSD to alter the data rate. Based on the impending situation, the CCU broadcasts the data to the user located at the GBM where the situation is dealt-with accordingly. The system's novelty lies in its ability to optimize data broadcast rate for intermittent/expensive data links and its ability to be a lightweight middle-tier application sending only the critical information to a remote station.

Keywords— Condition monitoring; Embedded Monitoring; Machine Learning; Client Server Computing;

I. INTRODUCTION

Monitoring and control of vehicles in remote areas such as open-seas and hazardous environments is one of the hottest areas of research. With the advent of high-speed internet technologies and advanced telecommunication techniques along with publicly available communication hardware such as the smart phones, it is now possible to monitor condition of remotely located equipment and hardware via standard TCP/IP protocols. Though, with the current advances in the GPRS and 4G protocols, internet access has become available at remote locations, continuous and robust access to internet in remote areas, particularly open-seas still remains an expensive and intermittently available commodity. Though companies such as Inmarsat and Thuraya aim to provide global access to internet and telephony services, their connection services and devices are expensive and the counterpart cheaper versions regularly require re-calibration with the satellites to ensure continuous internet access.

Conventionally, sensor outputs used to relay information remotely are depicted by electronic devices that output a standard analogue value with in a particular range (e.g.0-5 V, 4-20mA). With a continued drop in the cost of microcontrollers, a new generation of sensors termed "smart sensors" is now growing which can be utilized in mass-producing devices [1]. These sensors also exhibit an embedded network capability in them which allows easier control and monitoring of data being collected from the sensors. Conventionally, sensor networks have been implemented using application specific buses such as CAN, ProfiNet and LonWorks. However, Ethernet is fast replacing these widely used mid-level field bus systems [2] The main advantage of using Ethernet as a networking standard is because they it is cheaper and faster than most proprietary networks [3]. Ethernet enabled devices solve the problem of remote communication. Moreover, problems encountered in asset tracking, diagnostics, monitoring and data retrieval from sensors can be accomplished from remote centralized location [4]. However, a robust communication access to remotely monitor an environment is still not feasible. Well-known internet providers do provide global internet coverage but the modules used are expensive and require continuous calibration to keep a continuous connection. Moreover, Ethernet is not the preferred choice for control and monitoring of systems with hard real time requirements. Though this is currently an intensively researched area, Ethernet networks can only be used in systems with soft real time deadlines such as non-time critical data logging for trend analysis.

Based on these limitations, this paper presents a novel, multi-tier framework to facilitate efficient communication of critical sensor data from remote vessels or UAVs to ground-based locations via a standardized variable-mode burst-type data transfer via a standard file-transfer protocol. The paper thus presents the design and implementation of a remote condition monitoring and control system based on ARM microcontroller-based sensors. The detailed framework described in this paper is as follows. Section II presents a review of existing state-of-the-art in condition monitoring and control along with its limitations and possible improvements. Section III presents a discussion into the framework implemented along with the architectures and protocols used to collect, collate, model and broadcast data from remotely operating vehicles (ROVs) to various ground-based monitor (GBM) locations. Section 3 presents the underlying methodology used to train an AI model based on regular time-based behavior of all the sensors presents within the architecture which is then used as a basis to parse the data necessary and required to be broadcasted via the internet. Section 4 presents an analysis of the outcomes and Section 5 concludes with future directions and prospects of this work.

II. OVERVIEW

A. Condition monitoring and data broadcast in ROVs

Condition monitoring of ROVs and high-risk environments and devices is regarded as one of the foremost areas of research in a wide range of real-world applications. The term monitoring in industrial domain is generally regarded as a process used to prevent catastrophic failure of critical machinery or processes with objectives ranging from repair time and cost reduction to revenue loss and maintenance cost saving [1]. Recently, in the backdrop of safety, security and surveillance applications, the term has further extended to processes where the techniques are used to monitor equipments or even behaviors via real-time, image-based condition monitoring.

B. Industrial backdrop of TCP/IP data acquisition and analysis

Condition monitoring of machinery and processes over standard TCP/IP domain have increasingly being used in systems lately to improve the overall monitoring process. The systems generally monitored as such are complex and contain networks of multiple sensors connected in specific topologies to propagate status of the underlying systems. Generally regarded as multi-sensor data fusion, the underlying phenomenon of collating and processing information from these multiple sensors was initially investigated by Hall and Llinas [2] who presented a multi-level approach to integrate information. The data thus obtained from various devices was thus based on its importance generally on the basis of an expertly tuned rule/knowledge base. Yen and Lin [3] utilized a wavelet packet feature extraction methodology to monitor and classify vibration signals. Self-configuration of various network sensor topologies were initially investigated by Cerpa and Astrin [4] who reported a linear increase in energy saving in case of node failures which is a critical issue in hazardous environments. A detailed review of condition monitoring and fault diagnosis in electrical machinery was reported by Nandi *et al.* [5] who reported various techniques based on axial flux-based measurements, vibration analysis, transient current assessment and voltage monitoring.

In the industrial backdrop, remote maintenance and condition monitoring systems related research is done by a wide range of organizations for safety as well as security purposes with scopes extending to background subtraction [6], erosion detection [7], wind turbines assessment[8], industrial machine monitoring [9], transport monitoring [10] and water quality assessment [11]. However, the predominant bottleneck in internet-based condition/situational monitoring still remains with the processing of raw information to extract the most suitable information to be broadcasted to the remote locations. In the presence of video-based sensors, this requirement is further bolstered due to the size of each of the video frames being processed by the processing server. GPRS-based technologies were recently used by [12] to remotely monitor a pre-fabricated substation. [13] designed and implemented a remote online machine condition monitoring system based on a distributed sensor setup. The technique developed a predictive maintenance system to monitor a wide array of machines on the basis of features extracted to train an artificial neural network (ANN)-based system. However, the systems mentioned above have their own short-comings in terms of seamless transmission of data in remote locations, number of sensor integrated as well as their ability to handle large amount of data.

III. A MULTI-TIER DATA BROADCASTING FRAMEWORK

The system uses standard TCP-IP sockets to broadcast data to a centralized hub where the data is evaluated against a trained neural model to assess its viability to be broadcasted via a standard internet medium to a remote ground-based station. The packet network provides two types of accesses depending upon the type of application it is used in. For open-seas usage, it is aimed to use a standard Inmarsat broadband access whereas it uses standard mobile IP for land-based internet access. The proposed framework is adaptable to a wide range of industrial, video security and medical remote monitoring applications such as CCTV monitoring, machine status assessment, factory automation and remote (IP-based) control systems such as emergent system shut-downs or resets.

As discussed earlier, there are potential problems in the system architecture and functionality of currently available Ethernet enabled smart sensors and their networks. Typically, manual configuration (sample time update and setup) of individual sensors in a large network can be time consuming and prone to human errors. Usually these smart sensors are developed around a 16 bit microcontroller with limited memory resources in the region of around of RAM [14] . Most microcontroller based Ethernet enabled sensors store the main application source code, the TCP/IP

The project is sponsored by the Knowledge Transfer Partnerships/Technology Strategy Board, UK. The project is a joint venture between STS Defence Ltd and University of Portsmouth

stack and the web pages on the on-chip ROM [15]. The ROM memory on chip is limited and this would mean the need for an external ROM chip. Also when the web pages need to be updated, the microcontroller in the sensor would need to be re-programmed. Internet connectivity is also intermittent especially in remote locations. Consequently, there might be instances when the internet connection maybe down for a few hours where thousands of sensor readings might be taken. The data upload through the web server might not happen in this case and cause data loss.

Based on the hardware and software limitations discussed about, the methodology presented in this paper presents a multi-sensor network architecture operating over a standard Ethernet topology to gather data from an array of sensors. The smart sensor and central controller based prototype solution attempts to resolve the drawbacks encountered in traditional Ethernet based sensor network solutions. The smart sensor units (SSUs) developed for this framework is designed to have a self-identification and auto-updating capability of sample rates. The microcontroller contains only application specific code and the web pages are hosted on the GBU side. The data is captured from the sensors and sent to a centralized controller unit (CCU) via a standard TCP/IP Ethernet connection (See Figure 1). The CCU is a low cost embedded computer and can handle multiple sensor connections through a hub. The SSU is prototyped on the microcontroller board with the Arm Cortex M3 microcontroller. The board is used since it can be easily run on 16-bit microcontroller. The sensor utilized in this work comprise of three primary tiers with each having its own unique objectives. The main network architecture is shown in Figure 1.

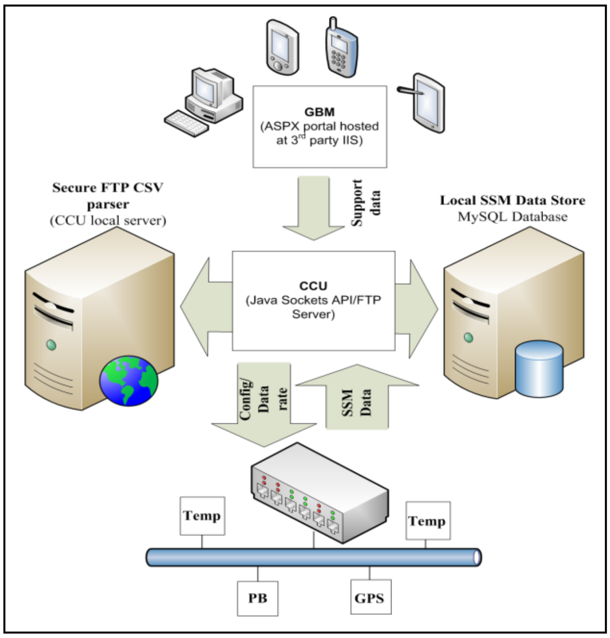

Figure 1. Architectural layout of the remote monitoring framework

A. A self-configurable sensor module

The network utilized in the proposed framework contains an array of sensors with the architecture shown in Figure 2. The STM32F207 series microcontroller (based on ARM Cortex M3) was chosen as the microcontroller for the smart sensor. The Ethernet stacks being used were carefully selected in a way that the application would be able to run on a microcontroller with lesser on-board RAM & ROM. The Ethernet stack being used was the LWIP stack since it was open source and it is actively being used in the industry. The focus of the LWIP TCP/IP implementation is to reduce RAM use while still having a full scale TCP/IP stack. This makes LWIP suitable for use in embedded systems[13] [16].

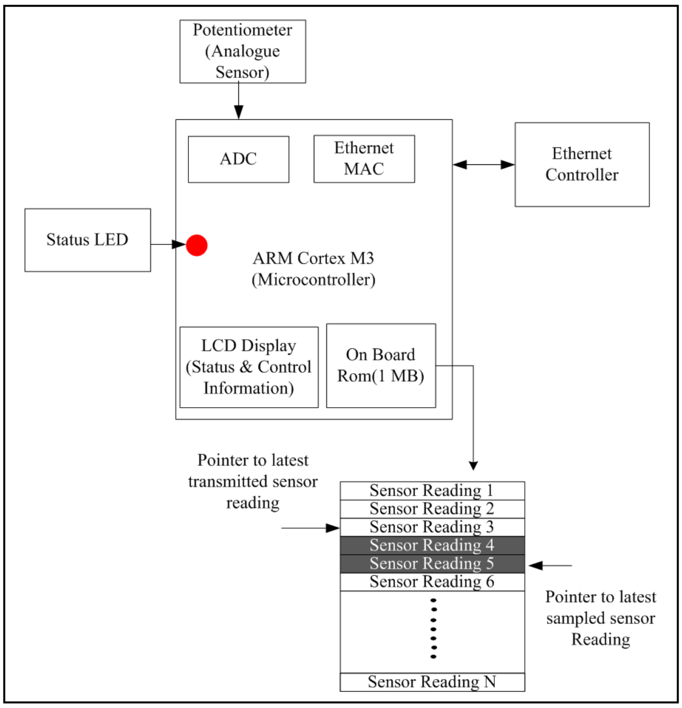

Figure 2. Layout of a smart sensor module (SSM)

The raw callback API without any operating system was used to call functions from the TCP/IP stack. The smart senor was built to in co-operate the following features

- Fully integrated unit that could be plugged onto an existing Local Area Network via the Ethernet port.
- Self-Powered via a POE adapter: Since sensors are often located at remote areas, the solution developed by us will include a Power over Ethernet (POE) adapter which will supply the required power to the SSUs.
- Self-Identification: Once connected to the LAN, the SSU establishes a TCP/IP connection with the server (CCU) and identifies itself to the CCU automatically. This is possible by assigning every sensor with a unique static IP which the CCU has stored on its MySQL database. For e.g.: Sensors with IP addresses between 192.168.1.0 and 192.168.1.10 are saved as temperature sensors.

- Self-Configuration: The server (CCU) then sends the configuration parameters such as sample rates, data transmit rates for different conditions such as normal operating mode, alarm mode, emergency mode during which the sample rate is higher as the data broadcasted during abnormal situations is deemed critical.
- Data storage and transmission algorithm: The data collected is stored on the on-chip ROM (or external ROM chip) in a cyclic array from where data is transmitted as shown in Figure 2.

As soon as the smart sensor is plugged onto the LAN it gets powered via a POE splitter drawing the Power from the Ethernet switch.

B. *A multi-tier software architecture for real-time ROV monitoring*

It must be noted that the data being broadcasted to the GBU predominantly depends upon the type of situation the ROV is currently operating in. The CCU uses a standard file transfer protocol to upload critical data which is then parsed by the GBU to display critical reports to a remote user. In order to enable the CCU to decide which data could be sent via TCP/IP, it was imperative to perform a statistical assessment of the data to differentiate between normal and critical data. The main objective here was to reduce the amount of garbage sent over an expensive and intermittent medium. For instance, a temperature sensor broadcasting engine room temperature to the GBU when the room temperature goes beyond a normal range of operation can be initiated via the CCU if the value goes outside the temperature range stated in the configuration table.

1) Smart Centralized Controller Unit (CCU) based TCP/IP data parser

The main objective of the centralize CCU is to accept data from a network of SSUs, assess it and then broadcast the most crucial values to the remotely located GBUs. The underlying notion is to minimize storage and transfer of irrelevant information. The CCU acts as a central hub that decides if raw data inbound from a set of SSUs is actually eligible to be broadcasted as CSV FTP file to the GBU. The CCU comprise of an embedded Intel Atom-based 1.2 GHz processor operating over embedded XP with 1GB of RAM. The CCU software was developed on J2SE platform which operates as a terminate-and-stay (TSR) resident program to ensure recovery in case of accidental restarts. The embedded computer is a solid-state IP6 device with no moveable parts and a hard sink to dissipate heat. Once started, the system acts as a TCP/IP WinSock server that opens a unique port that is known to all the SSU devices being connected to the network (See Figure 3).

Once listening, any configured SSM device can attempt a connection with the CCU via a standard Socket connection. Once the connection is attempted, the CCU broadcasts two core messages to the attempting SSM in order to advice an initial data rate via a configuration table and system status type. The status type at the beginning of a connection is usually LIVE mode. Once an emergency situation is trigged via any of the SSU or a manual over-ride such as a panic button, a distress signal is sent to the CCU.

With regards to the retrieval of the distress signal from the SSM, the CCU performs the following core tasks in the strictest order:

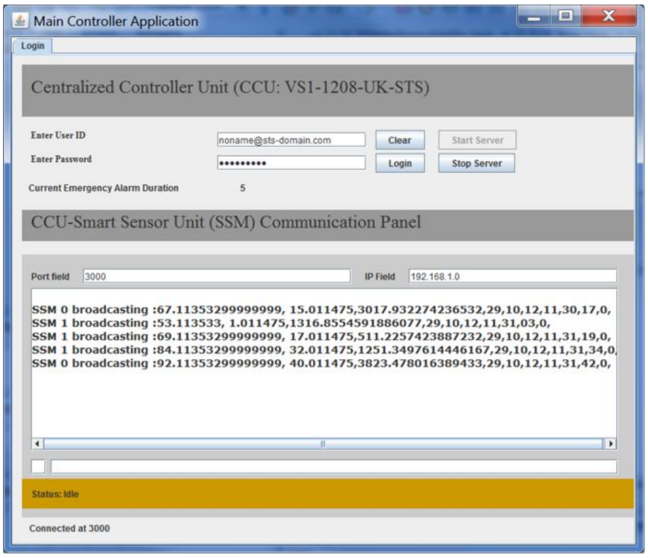

Figure 3. Architecture of a CCU interface connected to two SSM devices (SSM 0 and SSM 1). The device ids can be seen in the real-time relayed and showed in the CCU-SSM Communication Panel

Based upon the severity of signal obtained, broadcasts a short emergency command via FTP to the GBU

- Requests a data transfer rate update from the SSU side which is normally an increased rate
- Broadcasts the new data rate to the remote, GBU portal
- Awaits any further changes in the emergency mode from the SSU side

The detailed handshaking mechanism for the whole process is shown in Figure 4. The system is primarily distributed at two locations – the ROV side and the ground-based GBU side. The ROV side comprise of the array of SSUs and a single CCU device. The initial connection is initiated between the SSM and the CCU when a normal TCP/IP connection is established within and no remote communication takes place. In case of a sensor-based or manually triggered emergency, the data is parsed by the CCU and an increased data rate is requested from the respective CCU side. At this point the CCU also attempts a “server write” connection with the GBU server to FTP the recently updated CSV file containing the emergency notification. The GBM parses the file based on its modification timestamp and communicates the recent GPS location and other sensor values to the main web portal. (See Figure 5 and Figure 6)

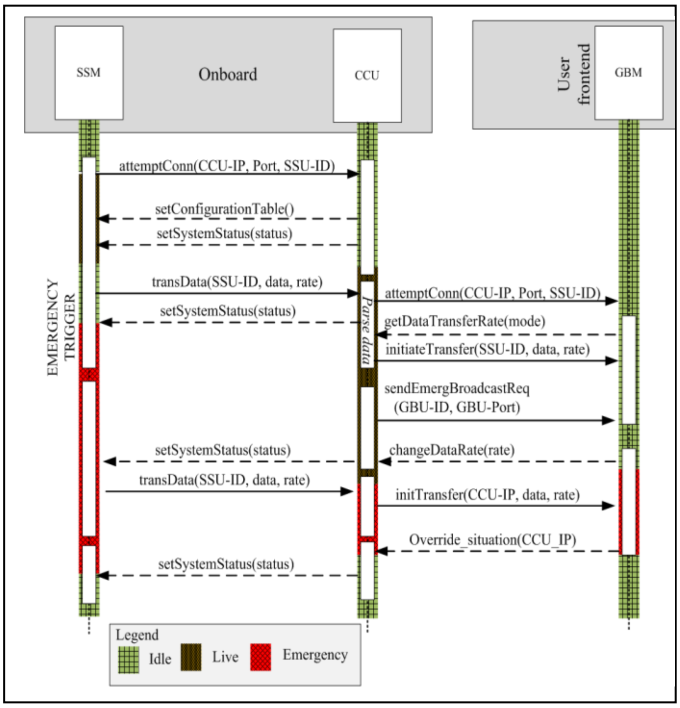

Figure 4. Emergency broadcast communication architecture between the ROV systems (SSMs/CCU) and the GBM

2) Server-side AJAX-based GUI for real-time distress signal monitoring

The server-side GBU is a hosted web server with its support modules written in AJAX, BingMaps and Google Charts API to display a real-time display of data from a set of vessel, vehicular or UAV CCUs. A demonstration of BingMap-based mapping of longitude and latitude coordinates of a simulated UAV flight are shown in Fig *. The flight was simulated from the city of Southampton to an assumed base station in the North Pole due to the current unavailability of long-range ground, air or maritime-based vessels to evaluate the current system.

Moreover, based on the type of emergency, the increased data rate was also broadcasted over a Charting engine written in Google Charts API. The scripting page written in C# evaluated the time and ID-stamped CSV file obtained via FTP from the CCU was parsed and four values that included GPS location coordinates (longitude, latitude) and temperature. The Chart control demonstrating a time-based variation in temperature values is shown in Figure 5.

IV. Conclusion and Future Directions

The paper presented the design and development of an array of Ethernet-based smart sensor modules (SSMs) capable of sampling, regulating and broadcasting data depending upon the impending situation at a remote vessel/vehicle (ROV). The data is stored at a ROV-based local central server (CCU) which is parsed and uploaded to the remote ground-based user/station (GBM). At the SSM side, the framework was implemented via an ARM Cortex M3 microcontroller which is responsible for the TCP/IP-based data relay to the CCU. The CCU forms the core of this framework which is responsible for the analysis of data values broadcasted by the SSMs. Based on a configuration table containing permissible normal data values set by the experts, the CCU decides if a set of values are actually within the normal operating range of a particular SSM. If an anomaly occurs, the CCU initiates a change in data rate with the respective SSM and uploads the data values to the GBM via a secure FTP connection. The GBM-side implementation is based upon a third-party .Net/SQL Server 2008 server setup that periodically checks for all the ROV uploaded files for any updates and thus updates the user front-end with any ROV-related information.

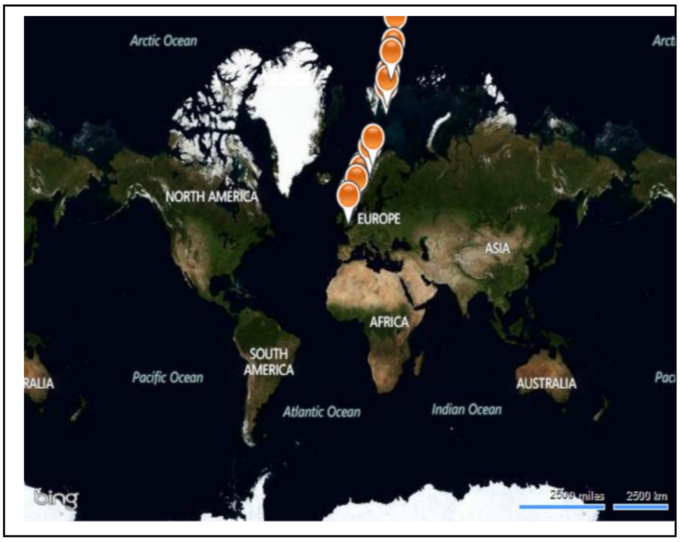

Figure 5. GPS locations of a simulated UAV flight path obtained via a TCP/IP simulator demonstrating the longitude and latitude coordinates from Southampton, UK to the North Pole

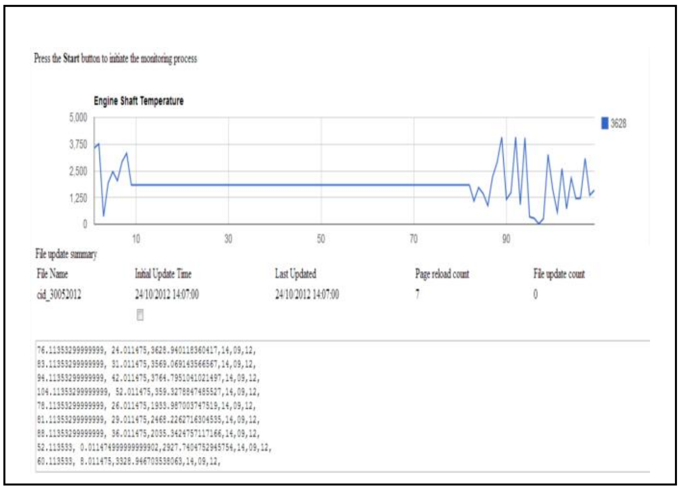

Figure 6. Mapping of parsed, time-stamped data obtained via CSV FTP from the CCU

As an ongoing venture, the underlying rationale of this work is to integrate an AI-based condition-monitoring model on the CCU side. Based on this AI module, the CCU will assess the data on the basis of a supervised machine learning technique instead of hardcoded configuration table values. Development of such a system will introduce a novel technique utilizing a burst-type AI methodology to monitor remotely operating vehicles/vessels and industrial machinery with minimal manual intervention and a high level of self-configuration.

REFERENCES

[1] Wang, K., *Intelligent condition monitoring and diagnosis systems : a computational intelligence approach*. Frontiers in artificial intelligence and applications. 2003, Amsterdam ; Oxford: IOS Press. x, 122 p.

[2] Hall, D.L. and J. Llinas, *An introduction to multisensor data fusion.* Proceedings of the Ieee, 1997. **85**(1).

[3] Yen, G.G. and K.C. Lin, *Wavelet packet feature extraction for vibration monitoring.* Ieee Transactions on Industrial Electronics, 2000. **47**(3).

[4] Cerpa, A. and D. Estrin, *ASCENT: Adaptive self-configuring sEnsor network topologies.* Computer Communication Review, 2002. **32**(1).

[5] Nandi, S., H.A. Toliyat, and X.D. Li, *Condition monitoring and fault diagnosis of electrical motors - A review.* Ieee Transactions on Energy Conversion, 2005. **20**(4).

[6] Yusuf, A., Syed, Shetty, S., Wilkinson, N., Brown, D., J. A Modified Codebook-based Background Subtraction Technique to improve Activity Classification in Highly Variable Environments. in International Conference on Industrial Engineering. 2012. Dubai, UAE: World Academy of Science, Engineering and Technology.

[7] King, C., et al., The application of remote-sensing data to monitoring and modelling of soil erosion. Catena, 2005. **62**(2-3).

[8] Garcia Marquez, F.P., et al., Condition monitoring of wind turbines: Techniques and methods. Renewable Energy, 2012. **46**.

[9] Wang, W., P.W. Tse, and J. Lee, *Remote machine maintenance system through Internet and mobile communication.* International Journal of Advanced Manufacturing Technology, 2007. **31**(7-8).

[10] Errath, R.A., J.J. Dominguez, and I. Ieee, *Remote drive condition monitoring.* 1999 Ieee-Ias/Pca Cement Industry Technical Conference - Conference Record, 1999.

[11] Glasgow, H.B., et al., Real-time remote monitoring of water quality: a review of current applications, and advancements in sensor, telemetry, and computing technologies. Journal of Experimental Marine Biology and Ecology, 2004. **300**(1-2).

[12] Xiaorong, Y., L. Shuguang, and P. Tong, Remote Monitoring System of Prefabricated Substation Based on GPRS, in Computer, Mechatronics, Control and Electronic Engineering (CMCE), 2010 International Conference on2010, IEEE. p. 236 - 239.

[13] Pan, M.-C., P.-C. Li, and Y.-R. Cheng, *Remote online machine condition monitoring system.* Measurement, 2008. **41**(8).

[14] Drumea, A. and P. Svasta. Designing low cost embedded systems with ethernet connectivity. in Design and Technology in Electronic Packaging (SIITME), 2011 IEEE 17th International Symposium for. 2011.

[15] Wani, S.A. and R.P. Chaudhari. Ethernet enabled digital I/O control in embedded systems. in Computing, Electronics and Electrical Technologies (ICCEET), 2012 International Conference on. 2012.