Acquisition and Relaying of Data from a Floating Wireless Sensor Node using an Unmanned Aerial Vehicle

Remus Barbatei, Amund Skavhaug, Tor A. Johansen

Abstract—This paper describes a technological development project in charge of collecting and relaying data using UAVs (Unmanned Aerial Vehicles) from one or more small, low power, modular sensor nodes that can be placed in remote areas or floating on the water surface. The overall characteristics and architecture of the proposed sensor nodes and system around them are described. Field tests for one custom-built hardware prototype were carried out and the results are presented and discussed. The tests included: a proof-of-concept sensor node in water, an X-8 UAV for gathering and relaying the data and a base station connected to a computer used as an endpoint for the sensor data. The system was created using a combination of standardized and custom tailored communication protocols, firmware and application software. The experiments concluded with data successfully being relayed from the remote sensor floating in water to a UAV and further on to a base station that can be located at tens of kilometers away from the UAV.

Index Terms—Unmanned Aerial Vehicles; Payloads; Unmanned Aviation; Wireless Sensor Networks; Oceanography; Embedded System Design; Communication Relays; Floating Sensors; Modular Sensors; Sensor Fusion; Communication Protocols

I. INTRODUCTION

Oceanography, marine environmental monitoring and offshore operations have been around for many years. The application range for these kinds of systems varies from research and observation in fisheries and aquaculture, oil spill response, iceberg tracking to marine biology research and environmental studies. There is a potential to move away from the use of large ships towards small and autonomous unmanned vehicles systems. In many cases, the usage of UAVs can reduce time and facilitate communication acting as a relay node for networks of underwater, surface or aerial sensor nodes, thus collectively contributing to remote sensing, in-situ sensing and data acquisition [1], [2]. This research is motivated by the use of UAVs with increased autonomy, large data storage and high bandwidth radio communication capability [3], [4], [5].

Buoy sensor networks nodes operating at the ocean surface [6], [7] are suitable for many monitoring applications, but they also face some challenges: size, cost and, in some cases, limited communication capabilities. Particularly, there may be poor conditions for transmission of radio signals over large distances to other assets that are at the ocean surface, or at low altitude onshore. This is mainly due to electromagnetic waves propagation related phenomena at microwave frequencies. Microwave and UHF frequency signals require Line-of-Sight (LOS) which, on longer distances, is disturbed by the Earth curvature and affected by first Fresnel Zone requirements [8]. Moreover, signal reflections and changes in pressure, humidity and temperature at the Atmospheric Boundary Layer (ABL) can directly influence waves characteristic and propagation paths which effect in ‘fading’. Considering water environment, it has been proven that sea state - both wave height and frequency - has direct influence on fading effect [9]. In addition, the endurance of autonomous buoys and other floating sensor nodes depend on low power consumption (i.e. low radio transmission power) and their design usually prevents the use of large or elevated antennas.

A first alternative to LOS data transmission
would be to use elevated antenna relays that greatly improve the radio communication channel. But these require a suitable platform to provide elevation. Mountains, tall masts and aerostats may provide the necessary means onshore or on a larger mother-ship, but for data nodes far from the shoreline, another type of elevated platform is needed near their location.

Yet another option for communication is the use of a satellite link. An example of a successful implementation of satellite communication in oceanic observations is Argo [10], [7]. It is a global array composed of approximately 3000 profiling sensor nodes that are free-drifting in the oceans of the world. They are not only floating, but can submerge down to 2000m under the ocean surface, allowing for continual monitoring of salinity, temperature and the velocity of the upper ocean. The data they gather is sent via satellite to a centralized database which is publicly available. This kind of sensor network can also be part of bigger projects and experiments such as [11] or [12]. These projects are carried out by international organizations and governments with the purpose of having a global ocean observing network and prediction system.

Another solution for deep ocean monitoring is the second-generation Deep-Ocean Assessment and Reporting of Tsunamis system know as DART II [13]. Tsunami data from the DART system can be combined with seismic data ingested into a forecast model to generate accurate tsunami forecasts for coastal areas. The system is composed of two physical components: a ‘tsunameter’ on the ocean floor and a surface buoy with Iridium satellite network telecommunications capability. The DART II systems have bi-directional communication links and are thus able to send and receive data from the Tsunami Warning Center and similar sensors via the Internet.

Even though for Argo and DART II satellite communication is appropriate, for systems with small footprint, in some cases, coverage may be limited [14]. Furthermore, price-performance ratio for Iridium satellite communication does not facilitate its integration into any system, especially in systems with low-power, size constrains or large amount of data transfer requirements.

In order to address the matters previously described and, at the same time, try to resolve communication, cost, and small footprint issues that current solutions have, we propose a solution built around a custom made sensor node and its interaction with UAVs. The main contribution of the present paper is a complete design of a floating sensor node in terms of hardware, software and mechanical prototype. It is custom made to host various sensing modules as well as a digital radio communication link towards a UAV payload that implements data storage, real-time communication relaying, and delay-tolerant networking using DUNE and the IMC (Inter-Module Communication) protocol towards a central data hub with operators interfaces based on NEPTUS. IMC, DUNE and NEPTUS are part of the software toolchain developed for inter-operable networked aerial, surface and underwater vehicles by University of Porto, [15]. The overall system is composed of one or several units, UAVs and a base station for collecting the data and interacting with the devices as represented in Figure 1.

Fig. 1: General Overview of System Components and Interaction

This paper focuses on describing and presenting a new system that could, potentially, resolve some of the issues that current solutions have.

**Information presented in this document:**

- Proposed solution and main characteristics
II. PROPOSED SOLUTION AND DESIGN

A. Sensor node Physical Case Construction

The proposed shape for the sensor node is cylindrical, like most buoys due to their directional stability in water. Issues such as the ability of being deployed in water easily, the size and availability of the battery and electronics that can be fitted inside have been taken into account for a shape and size for the prototype. The sensor node size and weight are small enough so that it could be dropped from a small UAV in water. It is made out of PVC plastic and it is composed of 3 modules and a combo puck antenna[16].

Figure 2 shows a vertical section of two sensor node modules. Each of these sections has a rubber O-ring at the bottom that acts as a water sealant. These are also fastened together with screws placed above the O-rings so that there would be no holes drilled inside the compartment containing the electronics (blue color in the Figure 2).

B. Hardware Design and Architecture of the sensor node

The proposed sensor nodes are designed to work without constant coordination from a master device, they are modular, low-power and capable of housing various small-sized sensors including: GPS positioning capability, wireless communication and a non-volatile storage medium. The current prototype weight is a bit over 400 grams the height is approximately 27cm and the cross section is 5.5cm. Most of its case is submerged in water but the top (which contains the antennas) is above the water level. Each section has a specific functionality. Such sections can be added or removed from the sensor node, depending on the desired application in which the sensor will be used. Fig. 3 shows the built prototype and its custom hardware. It is composed of 3 sections and an antenna.

The core of the hardware lies in a 16 bit microcontroller: ATXMEGA192C3 [17]. The microcontroller operates at 32MHz and its current consumption does not exceed 15mA. The speed of the device can be dynamically lowered or increased at runtime and this enables flexibility in terms of power consumption. It possesses SPI, I2C, and UART which are used in the current system. Modules included in the biut prototype: micro SD card (interfaced using the SPI interface), GPS module (connected via a UART link), temperature sensor and a 9 axis Inertial Measurement Unit. Both IMU and the temperature sensor are connected to the same I2C bus. The IMU can be used to measure wave data (such as amplitude and period). This usually requires some processing that may be too power consuming for microcontroller. In this regard, signal processing that relies on simple numeric algorithms can be used [18], [19]. Each sensor node communicates with UAVs to send realtime or previously saved monitored data or for receiving commands and configuration parameters. These parameters can be anything from specific measurement time intervals patterns or software modifications that change the functionality of the node. This data must be transmitted fast, robust and over a relatively large distance. For this reason an RF wireless communication module is used: the Radiocrafts-RF1280 module. It uses a central frequency of 868MHz which translates into an approximate range of about 1-2km and it has a data rate of 4.8Kbits/s. It also has a relatively low current consumption: 21mA for receiving and 28mA for transmitting the data. The module uses a UART interface to connect to the microcontroller.

With modularity being an important characteristic, it is recommended that all sensors connected to the device should be interfaced by either an I2C or an RS-485 interface. Each new added sensor...
will be a new device on one of these interfaces and modifications will only be carried out on the software part, substantially reducing the workload required for adapting the hardware for new needs.

Figure 4 shows a block diagram representing various components and their interconnections in the proposed hardware solution. A minimum configuration is recommended for all sensor nodes: non-volatile data storage (micro SD card), wireless communication and a GPS module. The rest of the sensors can be chosen according to the monitoring requirements. For example, some nodes can be specially designed to monitor oil spills while others can be designed to monitor sea life or sea water properties such as the level of chlorophyll, nutrients, dissolved oxygen and pH.

Information regarding power consumption can be seen in Table 1. The most important components of the device and their power consumption are listed. All the measurements are made in the lab on the built prototype. The numbers represent the maximum values for consumption. In a normal scenario of operation, those values are only seen when the sensor wakes up and samples data with all of its sensors. During the rest of its operating cycle it is in sleep mode and the power consumption is decreased by several orders of magnitude.

<table>
<thead>
<tr>
<th>Component</th>
<th>Mode of Operation</th>
<th>Power (mW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPS module</td>
<td>Searching for satellites(^1)</td>
<td>290.4</td>
</tr>
<tr>
<td>GPS module</td>
<td>Tracking(^2)</td>
<td>217.8</td>
</tr>
<tr>
<td>IMU sensor</td>
<td>Normal operation</td>
<td>13.2</td>
</tr>
<tr>
<td>Radio module</td>
<td>Receiving(^3)</td>
<td>85.8</td>
</tr>
<tr>
<td>All components(^4)</td>
<td>Continuous sampling</td>
<td>726</td>
</tr>
</tbody>
</table>

**TABLE I: Power Consumption**

\(^1\) with connected active antenna, \(^2\) with connected active antenna and lock on 6 satellites \(^3\) also including consumption of the level shifter \(^4\) normal sampling and transmitting, no SD card writes, GPS locked, voltage regulator in use

**Battery life estimation:** The battery capacity of the prototype is 4000mAh. Based on the power consumption table, the device in full sampling mode would have about 220mA current consumption (with VDD at 3.3V) and in sleep mode ap-
proximatively 5mA\(^1\). The battery life for a scenario in which the device samples data\(^2\) for 5 minutes every hour is about 1 week. But not all applications require this kind of sampling frequency. For example monitoring during the night might not be needed all the time and wireless communication can be turned on only in specific pre-defined periods in which the devices expects a UAV flying around to collect data. If no radio communication is used and the devices only samples data 3 times every day, the battery life would be extended to 27 days. These calculations provide only a general estimate for the built prototype hardware and may vary for other scenarios and sensor configurations.

C. Software Architecture of the sensor node

The software running on the embedded device is a state machine. Figure 5 represents the main system states and their transition. The system has an initial state to which it cannot switch back to once it has been completed except for reset. This initial state is mainly responsible for the starting up of the system, loading of the drivers, initializing the filesystem, configuring the microcontroller frequencies, allocating the necessary memory for the peripherals used and performing an initial self-check and calibration of the sensors. Another state is the monitoring state. This is how the device will behave most of the time. During this state it samples data from the sensors and stores it for later transmission or sends it in realtime to any surrounding UAV. The third state in which the device can be is the command and control. The device transits from the monitoring state to command and control state only when the UAV issues a specific command. If the device enters the command and control state it waits for commands from the UAV or a timeout, before it can resume normal sensor sampling. The commands can be simple identification requests, or reconfiguring the device parameters (change of sampling rates for the sensors, deactivating certain sensors, etc.). The device can go into sleep mode either at specific pre-programmed time intervals or when issued by a command coming from a UAV.

Figure 6 shows the dynamic workflow of the software on the sensor node, starting from the initial state and proceeding to the monitoring state. Each of these steps are described below:

- **Configure MCU internals.** It is the first set of operations carried out by the microcontroller. It configures the clock systems, interrupt routines and allocates the necessary data memory. At this point an interrupt routine is started. This routine is dedicated to handling

---

\(^1\)including the voltage regulator loss and current running through some passive components connected to the microcontroller in sleep mode

\(^2\)with all sensors and wireless communication turned on
the communication with the UAV. It runs concurrently with the rest of the states and it is the means by which the device can abort its normal flow and enter into the Command and Control state.

- **Configure peripherals.** In this step the microcontroller loads the drivers for its base core devices attached to it. Most sensors will not be without a GPS module, a micro SD card for storage. The rest of the sensors connected to the main board are also started in this step.

- **Sensor self-check and calibration.** All the connected sensors to the device have their initial run and sample their first few reference values and test measurements.

- **Sample sensors and save/transmit data.** Here all functional and active the sensors that are connected to the device are sampled and their data is stored on the storage medium or sent to the UAV.

- **Enter sleep mode.** After the sensors have been sampled the device enters sleep mode and wakes up after a certain pre-defined time.

---

**III. DATA ACQUISITION AND RELAYING WITH UNMANNED AERIAL VEHICLE**

The software part of the system is split into 3 main categories:

1. The software that runs on the sensor node, responsible for interfacing the sensors measuring the data and for UAV radio communication.

2. The software framework used by the UAV to communicate with the sensor node - DUNE.

3. end-user application and interface: NEPTUS used to monitor the sensor(s)

Viewed from a higher level an overall architecture of the system is presented in Figure 7. DUNE software is responsible for relaying the data to the user interface. It can run on almost any machine that is capable of running Linux v2.6 or above. This means it can be installed on embedded computers which can be mobile and powered by batteries, thus achieving a high degree of freedom in the overall system. In the current scenario the framework is running on a Pandaboard which has been fitted inside an X8 UAV. The
Pandaboard is connected via a UTP cable to a Rocket M5 5GHz AirMax wireless radio device and with the Radiocraft-RF1280 Radio module using the serial port. A DUNE task has been programmed to read the incoming data from the serial port (radio link with the sensor node), parse it and pack it into IMC messages packages and to send these packages to a ground station running NEPTUS via a 5 GHz link. The protocol used for the wireless radio link between the UAV and the sensor node is custom built based on the IMC protocol: non-relevant data packets present in IMC such as timestamp have been removed in order to minimize unnecessary traffic over the low power radio link. IMC messages are sent through the network as UDP packets.

One possibility in which a UAV can collect data from a sensor node is as follows: A sensor node can be preprogrammed to constantly send the sampled data at a predefined time interval while the UAV flying above the sensor is constantly reading for incoming data. Alternatively, the UAV can wake-up the sensor node when it comes in range with it and issue a read transaction. To conserve as much power as possible the UAV is the one that continually polls for signals coming from the sensor node. The unit sends basic ping messages at configurable time interval. When any of these messages are received by the UAV it issues a wake-up command to the sensor and that in turn starts sending data to the UAV.

IV. Tests and Results

Tests have been performed in order to check the feasibility of the entire system as described in Fig. 7.

Test scenario and setup: Hopavagen, Agdenes, Norway was selected for the test location. The sensor node has been placed next to the shore in shallow waters, (about 1m deep) anchored to the ground using rope. The UAV was loitering above it to collect its data and to forward it to a base station. Figure 8 shows the UAV and its payload. In this test scenario the sensor node has been programed to sample data at a fixed timed interval, send it to the UAV and log it to the SD card as well. No sleep modes were used. The data sent by the sensor node is composed of temperature read and its GPS coordinates. The DUNE task running on the Pandaboard performs the following:

- reads the data coming from the sensor node and logs it in its own not-volatile memory
- sends an acknowledge message back to the sensor node
- forwards the received sensor data via the 5GHz link to the base station.

The base station is a computer running NEPTUS software which is also connected to a ROCKET

Fig. 7: Communication flow between the sensor node, the UAV running DUNE and a computer running NEPTUS

Fig. 8: X8 UAV payload
The flight time was approximately 12 minutes. The maximum distance at which the UAV was situated from the sensor node was 264m (measured using Google maps) and the maximum altitude measured during tests was approximately 100m. Within this distance, for the entire flight time there was a valid data link between the sensor node, the UAV and the base station. Previous experiments carried out by the author[20] have shown that the Radiocraft modules can communicate further than 600m, meaning that this test scenario was well within the previously tested limitations of the radio device. Figure 10 shows the generated flight trajectory logged by the UAV. During the test 344 data packages (IMC Messages) representing sensor readings were sent from the sensor node to the UAV. Out of these, 3 were dropped as having a CRC fail. The size of an IMC message ranges between a few tens to hundreds of bytes [21]. Two packages (GPS fix and temperature) were sent every 1200 milliseconds from the sensor node to the UAV. Precisely measuring the link speed during the flight test was not the main purpose of the experiment, but some estimates can be presented: The link between the UAV and the base station was done by interconnecting two ROCKET M5 AirMax Base Stations. Since they were at a maximum distance of 264m during the flight, the data rate would be in the range of tens of mbps ³ [22]. The maximum data rate for the radio device used to make the communication link between the sensor node and the UAV is 4.8KB/s [23].

Figure 9 shows NETPUS command interface showing the sensor node on the map, its coordinates and its temperature readings. Further technical details and more in depth test results for this project can be found at [20]. A video of the flight tests (including weather conditions and experiment setup) is available in: [24].

Even though the devices are 5GHz capable, the networking interface with the PC/embedded computer is done using a 10/100 BASE-TX Ethernet Interface, limiting the whole system to 100mbps

## V. Conclusions

With regard to other similar solutions, some key differences between what has already been made in this field and what was described above are:

- The sensor nodes are modular and can be easily adapted to new environments, both from software and hardware point of view.
- The ecosystem around the sensor nodes uses open source software that is compatible with most operating systems and has very flexible hardware requirements [15].
- Because of their adaptability, the sensor nodes are meant to be used together and interconnected with other types UAS and AUVs and even other sensor types.

The results of the tests suggest that the designed system can be used in the monitoring applications listed in the introduction. Some particular conclusions following the tests are:

- The tested range of the radio devices and antennas used in the sensor node proved to be sufficient, allowing for a good sensor node to UAV communication.
- The proposed physical case design proved to float in the water, be directionally stable and water-tight.
- The designed PCB and firmware in the embedded device have properly met the application requirements
- DUNE software framework was found to be suitable for this type data acquisition scenarios
- NEPTUS command interface can be used with ease for displaying sensor readings for such application types.

The research and work that has been carried out so far and the prototype built lead us to conclude that the approach and the idea for the project so far is valid. The DUNE and NEPTUS software used is open source, making it free, easy to expand, upgrade and debug, having a community of developers that support it.

## VI. Acknowledgments

This work was supported by the Research Council of Norway through the Centers of Excellence funding scheme, Project number 223254 - Centre
Fig. 9: Screen capture of Neptus showing the sensor node being placed in the water.

Fig. 10: Flight trajectory of the UAV during the test (generated with Google Earth)
for Autonomous Marine Operations and Systems (AMOS). The authors are grateful for the assistance provided by the team at NTNU, in particular Frederik Leira, Kristian Klausen, Jo Arve Alfredsen, John Olav Horrigmo, Per Inge Snildal, Terje Haugen, Lars Semb and Torkel Hansen, and the team at University of Porto, in particular João Fortuna, Filipe Ferreira and João Sousa.

REFERENCES


