Volume XXIII 2020
ISSUE no.1
MBNA Publishing House Constanta 2020

Scientific Bulletin of Naval Academy

SBNA PAPER • OPEN ACCESS

Cloud controlled USV over 5G network infrastructures

To cite this article: Sorin R. Sintea and Valentina M. Pomazan, Scientific Bulletin of Naval Academy, Vol. XXIII 2020, pg.277-281.

ISSN: 2392-8956; ISSN-L: 1454-864X

doi: 10.21279/1454-864X-20-I1-039

Cloud Controlled USV over 5G Network infrastructures

Sorin R. Sintea[1], **Valentina M. Pomazan**[2]
[1]Maritime University of Constanţa, 104, Mircea cel Batran blvd, 900663, România

[2]Ovidius University of Constanţa, 124 Mamaia, 900527 Constanţa, România

[1]sinteasorin@yahoo.com, [2]vpomazan@yahoo.com

Abstract. The communication devices can transport information from data controller to a long-distance central communication point. Industry 4.0 advances the concept of mobile devices that must be controlled and monitored from long distance. This paper represents a common effort to adapt the new ITC expectations and possibilities for existing equipment. This work proposes a highway communication infrastructure which ensures communication between mobile devices and fixed 5G networks infrastructure. The solution given addresses to the control and monitoring of an USV (unnamed surface vehicle) from Cloud, using IoT services. The scope is to reliably connect the USV real-time communication through IoT services, over 5G Mobile network solution. All data are to be send in Cloud, where will be installed the USV monitoring and control application, detailed by its modules, components and their role.

1. Introduction

The need for real time communications conducted the new ITC technologies into a new paradigm. Sensors capturing process data, are consolidated under digital prototypes and report for big analytics. The data format, its integrity and timing are essential for further analysis and networking, as well for creating input packages for reliable feedback and machine learning modules. A lot of new devices are developed to be controlled and to communicate using new cloud computing technology. Different types of infrastructures are offered for resilient both ways communication, and these technologies are aligned under one name: Internet of Things (IoT). IoT groups new technologies that make easy the communication between a plethora of devices and cloud, collecting meaningful data for mining and analysis. IoT can be used bidirectional, to receive data in cloud from a device and to send commands to a device from cloud.

Based on these concepts, this work reports an attempt to modernize the control of a new unnamed surface test vehicle (USV). In this case, our project refers to a remote-controlled drone, using radio waves.

2. System Architecture

The basic architecture of the proposed system consists of three components: USV, 5G infrastructure communication for IoT services and cloud IoT Server (installed on shore), as described in figure 1.

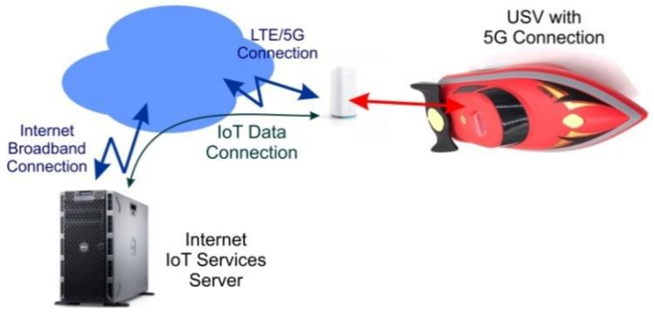

Figure 1

The basic design starts with an USV powered with two brushed drives. The USV is self-powered with electric accumulators and has a central process unit, dedicated for IoT connection and for USV autonomous control. In order to be able to ensure autonomous USV functioning, it was installed a GPS receiver, to measure the position in real-time. The USV must communicate with IoT services server, using 5G network communication infrastructure and, for this purpose, a 5G modem has been installed on aboard. The Internet IoT services server is installed in the cloud of the IoT services provider.

2.1. USV Hardware Implementation

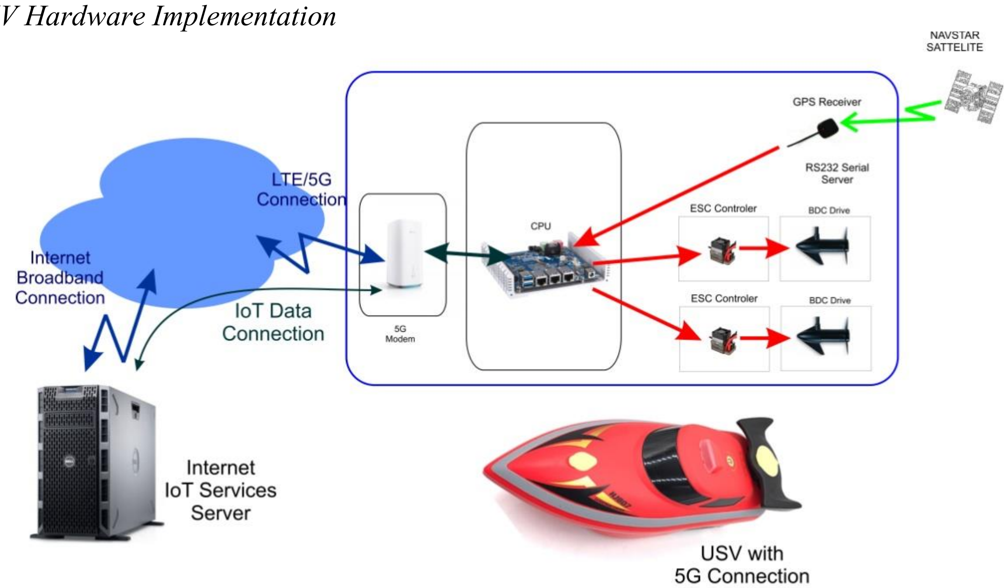

Figure 2

For this project (figure 2), two 12V DC electrical drives were used to power the USV. Each drive is controlled using ESC controller (electronic speed controller) – RNG 12V 320A. Each controller is powered from a local battery source and its control is ensured by local CPU (central processor unit) [1]. The CPU is a QBoat Sunny platform, specially designed for IoT applications by QNAP [2]. One important function of the CPU module is to control the USV after a preprogrammed route. The route is saved in the CPU memory. To be able to track the USV route, the CPU is connected with a GPS receiver (model BU-353S4 produced by GlobalSat WorldCom Corporation), which sends to CPU the USV position, course and speed in real time. Using all this data the CPU will control the drives speed and sense, based on tracking algorithm.

2.2. USV IoT Data Connection
The USV relates to IoT services over Internet, using a 5G router (model Huawei 5G CPE Pro Router produced by Huawei Technologies Co. Ltd) [3]. This router connects the CPU with 5G Data network (via data communication provider network). IoT services server are in the cloud of IoT service provider. For this application Microsoft's Azure IoT Hub services were used. All these services are located on the Microsoft cloud.

2.3. IoT Data Structure
The IoT services are organized on structured dynamic database. IoT service stores the preprogrammed track of USV and also the real-time position of USV. The data algorithm, running on cloud machine, analyzes the preprogrammed track and the real-time track of USV. The designer monitors through IoT services other USV parameters (status of batteries, temperature) or data monitored by the on board equipment (meteorological data, local video images)

3. System Logical Architecture

3.1. USV Software Architecture
The USV software architecture (figure 3) is created on two basic modules:
- "USV Track Control" – this module is responsible with USV control based on current position and a preprogrammed track, saved in CPU memory,
- "USV IoT Comm" – is the module responsible with data transfer with Microsoft's Azure IoT services server.

The "USV IoT Comm" module can communicate with Microsoft Azure IoT server using Azure IoT Library. To be able to use this library, Microsoft .Net 4 with .Net Core CLI was installed and we enhancing with .Net library the Microsoft.Azure.Devices.Client [4].

USV IoT Comm module can retrieve data from a specialized module "USV Data". The "USV Data" module can get data from different type of sensors installed on the USV (battery voltage level, current consumed, meteorological data, GPS and many more).

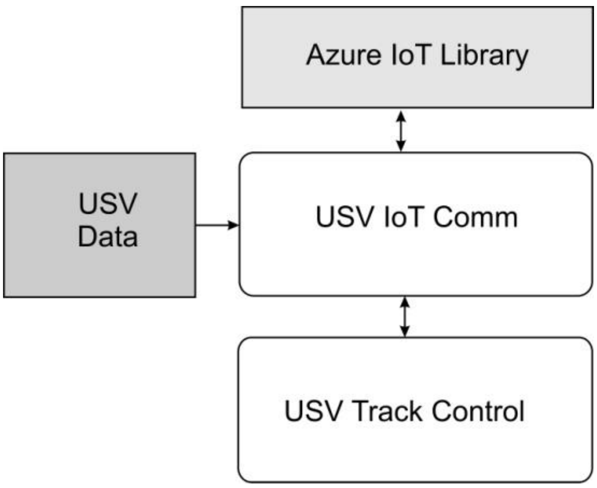

Figure 3

3.2. USV Track Control

The “USV Track Control” main function is to maintain the course and speed of the USV in according with a preprogrammed track saved in CPU memory.

The software module receives the data stream from GPS receiver through “GPS Data” module and stores this data into the CPU memory, as real-time data streams (figure 4). All data readied are stored into memory, into organized data structure location. In table 1, we present the GPS data stored into CPU memory. All data received are sent to the next module, “GPS, real-time clock, position, speed and course” module, which gets the data from streams (according to the table 1) and converts them into corresponding data types, interpretable by the program code.

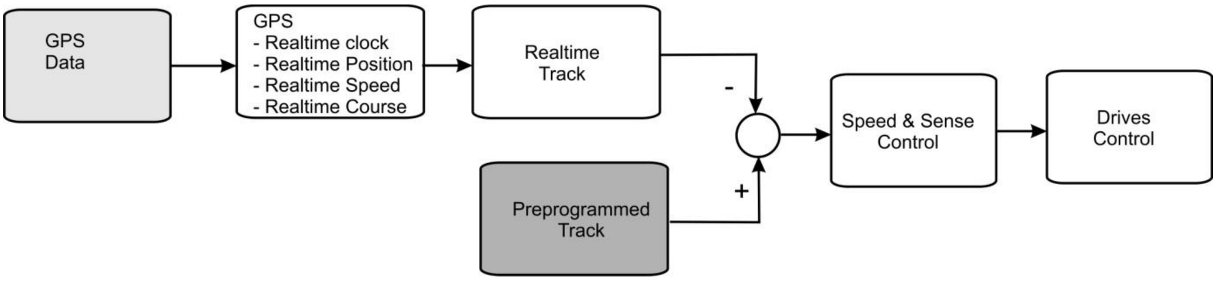

Figure 4

Table 1. GPS Data locations

Variable Name	Value (String data type)
Time	GPS absolute time and date
Location	GPS absolute location (latitude, longitude, altitude)
SOG	GPS absolute speed over ground
Course	GPS course

Next module, “Realtime Track” (figure 4), calculates the track parameters of USV movement. The data calculated by the module are used in next module, which is a track error comparator. The track error comparator makes the subtraction between the preprogrammed track and the real-time track and returns the current track correction.

The current track correction information is used to command the “Speed and Sense Control” module. This module calculates the USV speed and sense, according with track correction information. Speed and sense information calculated for current GPS location, are sent to a “Drives Control” module. “Drive Control” module (figure 4) calculates the values needed to control the USV drives.

3.3. USV IoT Comm

This module is designed to communicate data through Azure IoT services, between the USV and Azure Cloud IoT Services Server. This module will refresh the preprogrammed track information from local USV with information stored on the cloud server.

Data stored in USV IoT Comm module are structured as in table 2.

Table 2. USV IoT Data

Data Field Name	Data Type	Data Security
RT_Time	Date	ReadOnly
RT_Location	Location (long, long, long)	ReadOnly
RT_Speed	int	ReadOnly
RT_Course	int	ReadOnly
RT_Track_Data	{ int points; { position(long, long, long), time(Data)} }	ReadOnly
PRES_Track_Data	{ int points; { position(long, long, long), time(Data)} }	ReadWrite

Data presented in table 2 represents:

- RT_Time – is the real-time of USV's clock
- RT_Location – is the real-time of USV's location (latitude, longitude, altitude)
- RT_Speed – is the real-time of USV's speed
- RT_Course – is the real-time of USV's course
- RT_TrackData – is the real-time of USV's track points. This track data is a collection of track points (location and time stamp) measured starting with started point of USV.
- PRES_TrackData – are the presets USV's track points. This track data is a collection of track points (location and time stamp) presets to be achieved by USV.

4. Conclusions

The new technology applied to the USV can improve the versatility of a data collecting mobile intelligent device, like a water borne drone. This paper offers a solution to connect the data IoT library with an USV drone. Thus, the enhanced drone can be long distance controlled, using Internet technology. Real time monitoring for both position and speed are achieved, and any gaps in communication timeline can be either recovered or computed by trajectory data integration and analysis.

References

[1] Dobref V., Popa I., Popov P., Sintea S., Study for functional model of autonomous surface naval platforms, Navy Sectorial Program for Research and Innovation Proiect no.151/2016, "Mircea cel Batran" Naval Academy, Constanta, Romania 2016.

[2] *** https://www.qnap.com/en/product/qboat%20sunny

[3] *** https://consumer.huawei.com/en/routers/5g-cpe-pro/

[4] *** https://docs.microsoft.com/en-us/dotnet/api/overview/azure/iot?view=azure-dotnet